



ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στα
ΠΟΛΥΠΛΟΚΑ ΣΥΣΤΗΜΑΤΑ και ΔΙΚΤΥΑ

ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΤΜΗΜΑ ΒΙΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ

ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ



ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δίκτυα Γνώσης και Εφαρμογές Μεθόδων Στατιστικής σε Διασυνδεδεμένα Δεδομένα Κινηματογράφου

Νικόλαος Α. Λιβανός

ΕΠΙΒΛΕΠΩΝ: Δρ. Χαράλαμπος Μπράτσας

Τμήμα Μαθηματικών Α.Π.Θ.

Υπό την εποπτεία: Ιωάννη Αντωνίου

Καθηγητή Σ.Θ.Ε. Α.Π.Θ.

Θεσσαλονίκη, Νοέμβριος 2016



ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στα
ΠΟΛΥΠΛΟΚΑ ΣΥΣΤΗΜΑΤΑ και ΔΙΚΤΥΑ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΤΜΗΜΑ ΒΙΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ
ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ



Δίκτυα Γνώσης και Εφαρμογές Μεθόδων Στατιστικής σε Διασυνδεδεμένα Δεδομένα Κινηματογράφου

Νικόλαος Α. Λιβανός

ΕΠΙΒΛΕΠΩΝ: Δρ. Χαράλαμπος Μπράτσας

Τμήμα Μαθηματικών Α.Π.Θ.

Υπό την εποπτεία: Ιωάννη Αντωνίου

Καθηγητή Σ.Θ.Ε. Α.Π.Θ.

Εγκρίθηκε από την Τριμελή Εξεταστική Επιτροπή την

.....
Χ. Μπράτσας
ΕΕΔΙΠ Σ.Θ.Ε. Α.Π.Θ.

.....
Ι. Αντωνίου
Καθηγητής Σ.Θ.Ε. Α.Π.Θ.

.....
Π. Μπαμίδης
Επίκουρος καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Νοέμβριος 2016

.....
Νικόλαος Α. Λιβανός
Πτυχιούχος Μαθηματικών Α.Π.Θ.

Copyright © Νικόλαος Α. Λιβανός, 2016
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευτεί ότι εκφράζουν τις επίσημες θέσεις του Α.Π.Θ.

Στον πατέρα μου

Περίληψη

Στην εργασία αυτή λαμβάνονται μαζικά διασυνδεδεμένα δεδομένα κινηματογράφου από τον Παγκόσμιο Ιστό και περιγράφεται η διαδικασία εξόρυξης, καθαρισμού, επεξεργασίας και τέλος, δημιουργίας μιας εφαρμογής οπτικοποίησης των αποτελεσμάτων που αυτά προσφέρουν. Τα δεδομένα αυτά υπάρχουν ήδη μεμονωμένα στο πλαίσιο πληροφοριών κάθε σελίδας κινηματογραφικής ταινίας στη Wikipedia και μπορεί κάθε χρήστης να τα ανακτήσει, απλά μπαίνοντας στις αντίστοιχες σελίδες της. Για τη εξόρυξη και την εννοιολογική τους επεξεργασία όμως, χρησιμοποιήθηκε η τεχνολογία των διασυνδεδεμένων δεδομένων. Τα εννοιολογικά δεδομένα που ανακτήθηκαν χρειάστηκαν περαιτέρω καθαρισμό. Ο λόγος είναι ότι προέρχονται από την Wikipedia, και συνεπώς έχουν δημιουργηθεί από πολλούς διαφορετικούς χρήστες, χωρίς κανόνες διατύπωσης. Η εννοιολογική αναπαράσταση των δεδομένων αυτών στην DBpedia κληρονομεί τα σφάλματα των πρωτογενών δεδομένων της Wikipedia. Η περιγραφική λογική στην οντολογία της DBpedia είναι σε στάδιο ανάπτυξης έτσι ώστε να μπορεί να εντοπίσει τέτοια σφάλματα και να τα διορθώνει αυτόματα. Στην εργασία αυτή τα σφάλματα εντοπίστηκαν μέσω διαφορετικών τεχνολογιών και διορθώθηκαν. Στη συνέχεια τα δεδομένα αναλύθηκαν μέσω δικτύων, στατιστικής επεξεργασίας και ανάλυσης χρονοσειρών. Τα αποτελέσματα οπτικοποιούνται στο χρήστη μέσω μιας διαδραστικής εφαρμογής, δίνοντας του την ευελιξία να τα προβάλλει με πολλούς διαφορετικούς τρόπους. Η μέθοδος που παρουσιάζεται μπορεί να χρησιμοποιηθεί σαν οδηγός για την αποσφαλμάτωση της Wikipedia, δίνοντας έμπνευση για δημιουργία αλγορίθμων με αυτό το σκοπό. Ο κινηματογράφος είναι μία από τις πολλές κατηγορίες διασυνδεδεμένων δεδομένων όπου μπορεί να εφαρμοσθεί η μεθοδολογία και η ανάλυση που προτείνουμε και συνεπώς η εργασία αυτή μπορεί να αποτελέσει οδηγό για περαιτέρω έρευνα σε άλλες κατηγορίες ανοιχτών διασυνδεδεμένων δεδομένων.

Διασυνδεδεμένα Δεδομένα, Δίκτυο Γνώσης, Στατιστική, Κινηματογράφος

Abstract

In this paper, we describe the methodology and the results of the statistical analysis, data mining, graph analysis and network analytics of cinematography data retrieved from the Linked Open Data cloud. At first many issues arise mainly concerning the quality of the data retrieved. Wikipedia is a crowd sourced website and, as a result, the entries are phrased or worded differently by each user and this creates errors in DBpedia's entries. Descriptive logic in Linked Open Data is under construction as of this moment and these errors cannot be spotted and corrected automatically yet. Therefore, it was necessary to clean the dataset with different technologies. Afterwards the data was analyzed and the results were visualized through an interactive application giving the user the opportunity to view them from many aspects. The methods described can be used as a guide to correct DBpedia, offering insight for the creation of algorithms with that purpose. This paper is a study that demonstrates the benefits of using Linked Open Data in Data Science and its methodology can be reused in many other multivariate Linked Open Data subjects.

Linked Data, Knowledge Network, Statistics, Cinematography, Data Analysis

Πίνακας Περιεχομένων

Περίληψη	7
Abstract	9
Πίνακας Περιεχομένων	11
Εισαγωγή.....	13
Introduction.....	17
Ευχαριστίες.....	21
1. ΔΕΔΟΜΕΝΑ ΚΙΝΗΜΑΤΟΓΡΑΦΟΥ ΚΑΙ ΑΝΟΙΧΤΑ ΔΙΑΣΥΝΔΕΔΕΜΕΝΑ ΔΕΔΟΜΕΝΑ	23
1.1 Διασυνδεδεμένα Δεδομένα	23
1.2 Δεδομένα Κινηματογράφου.....	25
2. ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΕΞΟΡΥΞΗΣ, ΑΝΑΛΥΣΗΣ ΚΑΙ ΟΠΤΙΚΟΠΟΙΗΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	33
2.1 Εξόρυξη Δεδομένων.....	33
2.2 Καθαρισμός Δεδομένων.....	38
2.3 Ανάλυση Δεδομένων.....	42
2.4 Οπτικοποίηση Δεδομένων και Αποτελεσμάτων	46
3. ΕΞΟΡΥΞΗ ΚΑΙ ΚΑΘΑΡΙΣΜΟΣ ΤΩΝ ΚΙΝΗΜΑΤΟΓΡΑΦΙΚΩΝ ΔΕΔΟΜΕΝΩΝ.....	55
3.1 Εξόρυξη των κινηματογραφικών δεδομένων από την DBpedia	55
3.2 Καθαρισμός των κινηματογραφικών δεδομένων.....	62
4. ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΚΙΝΗΜΑΤΟΓΡΑΦΙΚΩΝ ΔΕΔΟΜΕΝΩΝ.....	71
4.1 Συνεργασίες εταιριών διανομής	73
4.1.1 Υπολογισμός πλήθους συνεργασιών	73
4.1.2 Ανάλυση των συνεργασιών	74
4.2 Αναζήτηση συσχετίσεων στα οικονομικά δεδομένα	77
4.2.1 Δημιουργία χρονοσειρών	77
4.2.2 Ανάλυση χρονοσειρών	78
5. ΟΠΤΙΚΟΠΟΙΗΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	82
5.1 Προσπέλαση δεδομένων από το χρήστη	82

5.2 Οπτικοποίηση μέσω δικτύων	83
5.3 Οπτικοποίηση μέσω γραφημάτων	85
5.4 Παρουσίαση των χρονοσειρών.....	87
5.5 Παρουσίαση των συσχετίσεων.....	89
Συνεισφορά - Συμπεράσματα.....	92
Μελλοντική δουλειά - Επεκτάσεις	95
Βιβλιογραφία	97
Σύνδεσμοι.....	98
Ευρετήριο Εικόνων.....	99

Εισαγωγή

Η παρούσα εργασία ασχολείται αρχικά με τη δημιουργία ενός δικτύου γνώσης από ανοιχτά διασυνδεδεμένα δεδομένα κινηματογράφου και στη συνέχεια με την ανάλυση του δικτύου αυτού και γενικά της πληροφορίας που αντλείται από τα δεδομένα αυτά. Τα δεδομένα υπάρχουν μεμονωμένα στα infoboxes των σελίδων της Wikipedia και για την άντληση τους χρησιμοποιήθηκαν ερωτήματα SPARQL στο endpoint της DBpedia [1]. Η τελική παρουσίαση των αποτελεσμάτων έγινε μέσω μιας εφαρμογής κατασκευασμένης μέσω της βιβλιοθήκης shinydashboard της R.

Τα διασυνδεδεμένα δεδομένα αποτελούν μια σχετικά πρόσφατη τεχνολογία και συνδέονται άρρηκτα με το σημασιολογικό Ιστό. Η χρήση τους υποστηρίζεται όλο και περισσότερο από κυβερνήσεις και την επιστημονική κοινότητα και οι ποσότητες των δεδομένων που περιέχουν ολοένα και αυξάνεται. Σταδιακά θα αποτελέσουν αναπόσπαστο κομμάτι του Διαδικτύου και σκόπος της εργασίας αυτής είναι να δώσει μια ιδέα στον αναγνώστη για τις πολλές δυνατότητες που του προσφέρουν τα δεδομένα όταν είναι διασυνδεδεμένα.

Τα δεδομένα που αντλούνται και αναλύονται αφορούν δημοσιευμένες ταινίες των 16 κορυφαίων εταιριών διανομής την 15ετία 2000 - 2014. Συγκεκριμένα επιλέχθηκαν οι κατηγορίες τίτλος ταινίας, εταιρία διανομής, έσοδα, έξοδα και έτος κυκλοφορίας. Τα δεδομένα αυτά και η ανάλυση τους είναι χρήσιμα από πολλές πλευρές, κυρίως για τις ίδιες τις εταιρίες αφού μπορούν να εντοπίσουν αίτιες στα προβλήματα τους, να κάνουν προβλέψεις και όχι μόνο. Φυσικά όταν μια τέτοια πληροφορία είναι διαθέσιμη και στο ευρύ κοινό εξασφαλίζει τη διαφάνεια των οικονομικών στοιχείων των εταιριών ή πιο απλά δίνει μια ιδέα στο θεατή για τις επόμενες ταινίες που θα παρακολουθήσει. Επίσης, η εργασία αυτή μπορεί να αποτελέσει έναυσμα για την επιστημονική κοινότητα για περαιτέρω αναλύσεις πάνω στο αντικείμενο αυτό, ή παρόμοιες αναλύσεις σε διαφορετικές κατηγορίες, όπως η μουσική, η λογοτεχνία, η τηλεόραση και ο αθλητισμός.[9]

Δεν είναι λίγες οι έρευνες που έχουν γίνει για τις εταιρίες διανομής και αφορούν είτε οικονομικά στοιχεία, είτε διαφάνεια σε ό,τι αφορά τις μη εξουσιοδοτημένες κυκλοφορίες και την πειρατεία ή ακόμη πιο απλά για να δημιουργήσουν ένα πιο εύκολο και φιλικό προς το χρήστη τρόπο να έχει πρόσβαση σε τέτοιου είδους δεδομένα. [6]

Σκοπός λοιπόν της εργασίας αυτής είναι αρχικά να δώσει στον κάθε χρήστη μια ιδέα για το τι είδους έρευνα μπορεί να συντάξει χρησιμοποιώντας τα διασυνδεδεμένα δεδομένα, να του επισημάνει σημαντικές δυσκολίες και προβλήματα που ενδεχομένως θα συναντήσει σε αντίστοιχη εργασία, να προτείνει λύσεις στις δυσκολίες αυτές και προφανώς να του παρουσιάσει την πληροφορία που δημιουργείται με τον πρακτικότερο δυνατό τρόπο.

Στο 1ο κεφάλαιο γίνεται μια ειδική αναφορά στα διασυνδεδεμένα δεδομένα καθώς και σε κάποιες ενδεικτικές πηγές κινηματογραφικών δεδομένων στο Ιστό. Εξηγείται η λογική των RDF δεδομένων καθώς και όλων των τεχνολογιών που παίρνουν μέρος στη δημιουργία των δεδομένων αυτών. Οι πηγές παρουσιάζονται κλιμακωτά ξεκινώντας από αυτές που δεν έχουν καμία σχέση με τα διασυνδεδεμένα δεδομένα, προχωρώντας σε αυτές που κάνουν προσπάθειες προς αυτή την κατεύθυνση και καταλήγοντας σε αυτές που όντως τα δεδομένα τους είναι σε μια συντριπτική πλειοψηφία διασυνδεδεμένα.

Στο 2ο κεφάλαιο εξηγούνται μερικές λειτουργίες εργαλείων και λογισμικών που χρησιμοποιήθηκαν για την εξόρυξη, τον καθαρισμό, την επεξεργασία και την οπτικοποίηση των δεδομένων. Στόχος του είναι να βοηθήσει τον αναγνώστη να καταλάβει τι εφόδια θα χρειαστεί σε περίπτωση που θελήσει ο ίδιος να κάνει αντίστοιχη έρευνα σε δεδομένα κάθε περιεχομένου.

Αρχικά γίνεται μια ειδική αναφορά στη γλώσσα SPARQL, εξηγείται η λογική της και επισυνάπτεται ένα παράδειγμα το οποίο εξηγείται και σχηματικά, αλλά και ανά σειρά κώδικα. Με τον τρόπο αυτό δίνεται η δυνατότητα στον αναγνώστη να συντάξει αντίστοιχα ερωτήματα στη γλώσσα αυτή και επομένως να δημιουργήσει τις δικές του βάσεις δεδομένων.

Στη συνέχεια περιγράφονται οι βασικές λειτουργίες του Open Refine [7], ενός εργαλείου απαραίτητου για την εργασία αυτή, μιας και τα δεδομένα της Wikipedia είναι μεμονωμένα, έχουν διαφορετικές διατυπώσεις και δεν είναι μηχαναγνώσιμα. Στο κομμάτι αυτό εξηγούνται

όλες οι λειτουργίες του Open Refine που χρησιμοποιήθηκαν στην εργασία αυτή καθώς και αρκετές ακόμη, οι οποίες αν και δε χρειάστηκε να εφαρμοσθούν στα συγκεκριμένα δεδομένα, είναι ιδιαίτερα χρήσιμες γενικότερα.

Μια ειδική αναφορά γίνεται στο SPSS καθώς προσέφερε πιο εύκολες και γρήγορες λύσεις σε ένα μικρό κομμάτι της εργασίας. Περιγράφεται το περιβάλλον του, καθώς και επιγραμματικά οι βασικότερες από τις λειτουργίες που προσφέρει.

Στο τέλος του κεφαλαίου αυτού περιγράφεται η R και η εξελιγμένη της έκδοση R Studio. Γίνεται μία σύντομη αναφορά στο περιβάλλον της και στον τρόπο λειτουργίας της και στη συνέχεια εξηγούνται αναλυτικά οι δύο βασικότερες βιβλιοθήκες εντολών της που χρησιμοποιήθηκαν για την εργασία αυτή. Στο πρώτο στάδιο αναλύεται η λειτουργία των βιβλιοθηκών shiny και shinydashboard μέσω των οποίων έγινε η οπτικοποίηση των αποτελεσμάτων, καθώς και πολλές βασικές ενέργειες που μπορεί να κάνει κάθε χρήστης για να φτιάξει τη δική του εφαρμογή. Στη συνέχεια εξηγούνται οι βασικότερες εντολές της βιβλιοθήκης igraph, με την οποία σχεδιάστηκαν όλοι οι γράφοι που εμφανίζονται στην τελική εφαρμογή.

Στο 3ο κεφάλαιο περιγράφεται αρχικά η διαδικασία άντλησης των κινηματογραφικών δεδομένων από την DBpedia. Επισυνάπτεται και εξηγείται ο κώδικας SPARQL που χρησιμοποιήθηκε και παρουσιάζεται η αρχική μορφή των δεδομένων. Εντοπίζονται ασυνέχειες και θέτονται στόχοι για τα επόμενα βήματα.

Στη συνέχεια αναλύεται η διαδικασία μεταφοράς των πρωτογενών δεδομένων στο Open Refine. Ο καθαρισμός όλης της βάσης δεδομένων περιγράφεται βήμα βήμα παρουσιάζοντας τη διαδικασία με τις αντίστοιχες εικόνες και παραδείγματα. Στο ίδιο κεφάλαιο περιγράφεται ο εμπλουτισμός και μερικές μικροεπεξεργασίες της βάσης δεδομένων από το SPSS.

Στο 4ο κεφάλαιο εξηγείται αρχικά η μεταφορά των δεδομένων στην R καθώς και η δομή που τους δόθηκε για ευκολότερη προσπέλαση στον κώδικα της εφαρμογής. Στη συνέχεια δημιουργήθηκε μια ξεχωριστή ενότητα για κάθε καρτέλα της εφαρμογής. Σε κάθε ενότητα περιγράφονται αναλυτικά τα βήματα και ο τρόπος σκέψης για να υλοποιηθεί η αντίστοιχη καρτέλα.

Συνολικά δημιουργήθηκαν 6 καρτέλες. Στην καρτέλα Data ο χρήστης μπορεί να προβάλλει και να κατεβάσει όποιο κομμάτι της βάσης δεδομένων θέλει. Στις καρτέλες Network, Bar Chart και Pie Chart παρουσιάζονται στοιχεία συνεργασίας μεταξύ των εταιριών διανομής με κριτήριο το πλήθος των ταινιών που κυκλοφόρησαν μαζί. Τέλος στις καρτέλες Time Series και Statistics παρουσιάζονται τα οικονομικά δεδομένα κάθε εταιρίας διανομής ανά έτος και αναζητούνται συσχετίσεις μεταξύ των χρονοσειρών που προκύπτουν μέσω διαφόρων στατιστικών μέτρων. (Cross Correlation, Mutual Information, Granger Causality). Τα αποτελέσματα που προκύπτουν ελέγχονται για τη στατιστική σημαντικότητα τους και κατόπιν χρησιμοποιούνται ως "πρώτη ύλη" για την κατασκευή γράφων στους οποίους ο χρήστης μπορεί εύκολα να πλοηγηθεί στις καρτέλες που αναφέρθηκαν.

Η εργασία αυτή καθώς και η εφαρμογή που δημιουργήθηκε παρουσιάστηκαν στο SEMANTiCS, το Σεπτέμβριο του 2016 στη Λειψία. Δόθηκε ιδιαίτερο ενδιαφέρον στις ατέλειες που εντοπίστηκαν στις εγγραφές της DBpedia καθώς και στη διαδικασία καθαρισμού των δεδομένων. Επιπλέον, ο κώδικας της εφαρμογής οπτικοποίησης υπήρξε αντικείμενο συζήτησης από πολλούς συμμετέχοντες, ως παράδειγμα των δυνατοτήτων της R στον τομέα αυτό.

Introduction

The motion picture industry is one of the largest markets worldwide, grossing a total of 38.3 billion US dollars only in 2015. For an amount of this magnitude, it is quite tempting to analyze data containing information about the financial situation of the distribution companies. However, obtaining such a large dataset can be a bottleneck for the average researcher. This thesis serves as an example on how Linked Open Data can present a solution to this problem.

The subject of this thesis is, firstly, to create a knowledge network from open linked cinematography data and secondly to analyze the above network and generally the information that is extracted from it. The data is already available for any user to see in the infobox of every movie page in Wikipedia, however it is isolated. As a result, SPARQL queries were used at the endpoint of DBpedia [1] in order to massively download it. The final presentation of the results was made through an application created for this purpose using the library "shinydashboard" from R.

The technology of linked data is recent and highly affiliated with the Semantic Web. They essentially are a new method for publishing structured data on the Web and creating links between them in order to form useful information. They are based on technologies such as URIs and the HTTP protocol and they are implemented with RDF technology, whereby the relational data are mapped into triplets, in a form suitable to connect with each other. This way, all data published on the Web as linked data can form a single global database. For queries on this type of data, an appropriate language has been created, named SPARQL.

The data that was retrieved and analyzed contain information about published movies of the top movie distributors through the 15 year time period of 2000 - 2014. Specifically the categories chosen were: movie title, distribution company, budget, box office and year of release. This dataset and its analyzation is useful from many aspects, mainly for the

distribution companies themselves, since through similar researches they can spot causes to their financial problems, make predictions and many more.

Additionally, when information like this is made public, it can be used by anyone on the Web to check the clear financial situation of each company, or for an even simpler purpose, such as giving the user an idea of what movie should he watch next, based on his or her preferences.

Finally, this thesis can be used by the research community as a guide to create many similar works, not necessarily having to do with movie data, but in other categories such as music, literature, television, or sports.

Many researches have been conducted for the same subject, and their purpose was to either create financial studies on the income and outcome of the distribution companies, or a study on piracy and its effects in the movie industry, or simply to create a friendly way for every user to have access to this kind of data. [6]

So, the purpose of this thesis is initially to give every internet user the idea of what he can accomplish by using linked data, explain many problems and difficulties that he may encounter along his way, suggest solutions on the mentioned difficulties and, of course, present the information that is created through the most practical way available.

The first chapter of the thesis provides general information on linked data, along with many resources of movie data anyone can find on the Web. The logic of the RDF data is explained, along with every other tool taking place on its creation. The resources are presented on a scale, from the ones that have no use of linked data, to the ones that partly use them, and conclude on those whose data is in its overwhelming majority linked.

On the second chapter is explained every tool and software used through the creation of this thesis, from the start to the finish. Firstly, the SPARQL language is mentioned, its logic is analyzed, and an example is attached and explained through a directed shape, and through every code line. This way, the reader gains the skill to conduct similar queries on this language, thus creating his or her own databases.

Subsequently, the basic functions of Open Refine [7] are described, a tool that was really necessary for this thesis, since the data on the Wikipedia pages were isolated, had different wordings than others, and therefore were not machine-readable. In this part every function from Open Refine that was used is explained, along with many others, that although were not used in this thesis, are especially useful in general.

A special reference is made to SPSS, since it offered many easy and fast solutions on a small part of this thesis. Its environment is described, and many of its basic functions are mentioned.

At the end of this chapter the language R is described along with its improved version, R Studio. A short reference is made concerning its environment and its way of functioning. Afterwards the two libraries that were most used in this thesis are explained. Firstly the function of the libraries shiny and shinydashboard are explained, along with many actions a user can make, in order to create his or her own application. Finally, the basic functions of the library igraph are described, the library used to draw every network presented in the final application.

The third chapter firstly describes the whole procedure of the data mining through DBpedia. The SPARQL code that was used is attached and explained, along with the data as it firstly occurred from the referred query. The procedure of importing the JSON files containing the dataset to Open Refine is described afterwards. The cleaning of the data is explained step by step, showing the corresponding images and examples. On this chapter is described the enrichment and the micro editing applied with the SPSS as well.

The fourth and final chapter initially describes the importation of data onto R along with the structure it was given in order to easily access it in the application's code. Afterwards, a small chapter containing details for every tab of the final application was created. Each of these chapters contains in detail the steps and the train of thought followed to create the corresponding tab.

The final application consists of 6 tabs in total. At the "Data" tab the user can access any point of the database he or she wants. At the tabs "Network", "Bar Chart" and "Pie Chart" are

presented results of cooperations between the distribution companies taking into consideration the movies that they released together. Finally, at the tabs "Time Series" and "Statistics" are presented the financial data of every distribution company per year and correlations between the time series that occur are searched with many statistical tools (Cross Correlation, Mutual Information, Granger Causality). The results are checked for their statistical significance and afterwards are used as "raw material" to create graphs that the user can easily navigate through the tabs mentioned.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κύριο Χαράλαμπο Μπράτσα, επιβλέποντα καθηγητή της εργασίας για τη βοήθεια και την καθοδήγηση του κατά τη διάρκεια εκπόνησης της εργασίας, καθώς και για τη γενικότερη συνεργασία μας.

Επιπλέον, θα ήθελα να ευχαριστήσω τον πρόεδρο του μεταπτυχιακού κύριο Ιωάννη Αντωνίου και γενικότερα όσους συνέβαλαν και συνεχίζουν να συμβάλλουν στη λειτουργία του ΔΠΜΣ: Πολύπλοκα Συστήματα και Δίκτυα, για τους ορίζοντες που άνοιξαν σε εμένα και σε κάθε φοιτητή του προγράμματος και για το έναυσμα που μας έδωσαν να ασχοληθούμε με ένα τόσο σύγχρονο και επίκαιρο κομμάτι της έρευνας.

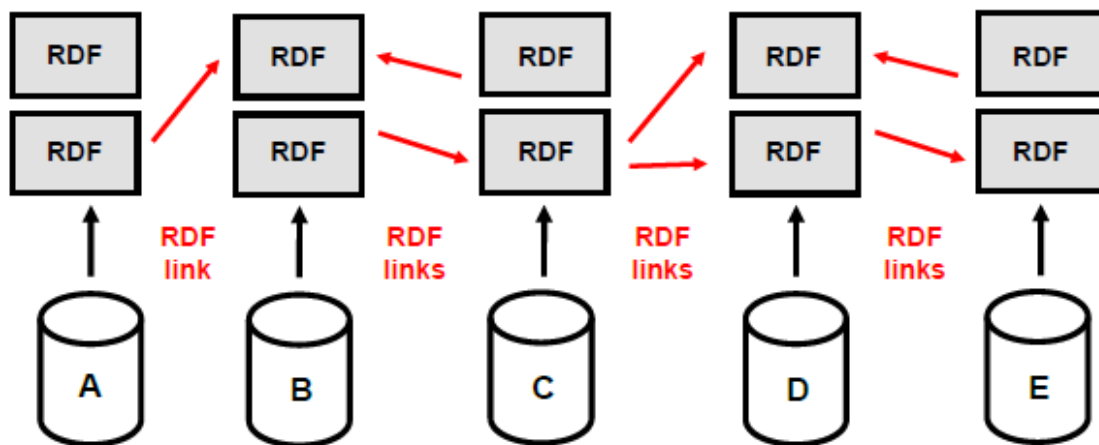
1. ΔΕΔΟΜΕΝΑ ΚΙΝΗΜΑΤΟΓΡΑΦΟΥ ΚΑΙ ΑΝΟΙΧΤΑ ΔΙΑΣΥΝΔΕΔΕΜΕΝΑ ΔΕΔΟΜΕΝΑ

1.1 Διασυνδεδεμένα Δεδομένα

Το ίντερνετ είναι ένα μέσο με το οποίο ο ανθρώπινος πολιτισμός ανταλλάσσει τεράστιο όγκο πληροφοριών καθημερινά. Παρόλο που η ήδη υπάρχουσα δόμηση έχει βοηθήσει στην ανάπτυξη επιστημών, τεχνολογιών και γενικά της γνώσης, ο όγκος πληροφορίας που παράγει η σύγχρονη κοινωνία μας τείνει να υπερκαλύψει την ικανότητα ανάλυσης της, δημιουργώντας την ανάγκη για εύρεση νέων τρόπων επεξεργασίας της.

Ο όρος *linked data* (διασυνδεδεμένα δεδομένα) περιγράφει μία μέθοδο δημοσιοποίησης δομημένων δεδομένων ώστε να είναι μεταξύ τους αλληλένδετα και κατά συνέπεια να γίνουν πιο χρήσιμα. Η μέθοδος αυτή στηρίζεται σε δυο θεμελιώδεις τεχνολογίες του Ιστού, το HTTP και τα URIs (Uniform Resource Identifiers), αλλά αντί να τις χρησιμοποιεί για να εξυπηρετεί ιστοσελίδες για τους ανθρώπινους αναγνώστες, τις επεκτείνει ώστε να ανταλλάσσουν πληροφορίες με τρόπο που να μπορούν να διαβαστούν αυτόματα από τους υπολογιστές. Αυτό επιτρέπει δεδομένα από διαφορετικές πηγές να συνδέονται και να μπορούν να αναζητηθούν. Ο όρος επινοήθηκε από τον ίδιο τον Tim Berners Lee το 2006 από μία σημείωση σε μια συζήτηση γύρω από το σχεδιασμό του Σημασιολογικού Ιστού.

Τα URIs και το HTTP υλοποιούνται σε μια τεχνολογία πολύ σημαντική στον Ιστό Δεδομένων, το Resource Description Framework (RDF). Αυτό είναι που χρησιμοποιείται για τη δημοσίευση δεδομένων στο διαδίκτυο. Πρακτικά, από τα δεδομένα URI οι μηχανές λαμβάνουν το RDF ενώ οι άνθρωποι την HTML. Έτσι, δεδομένα από διαφορετικές πηγές είναι συνδεδεμένα μεταξύ τους.



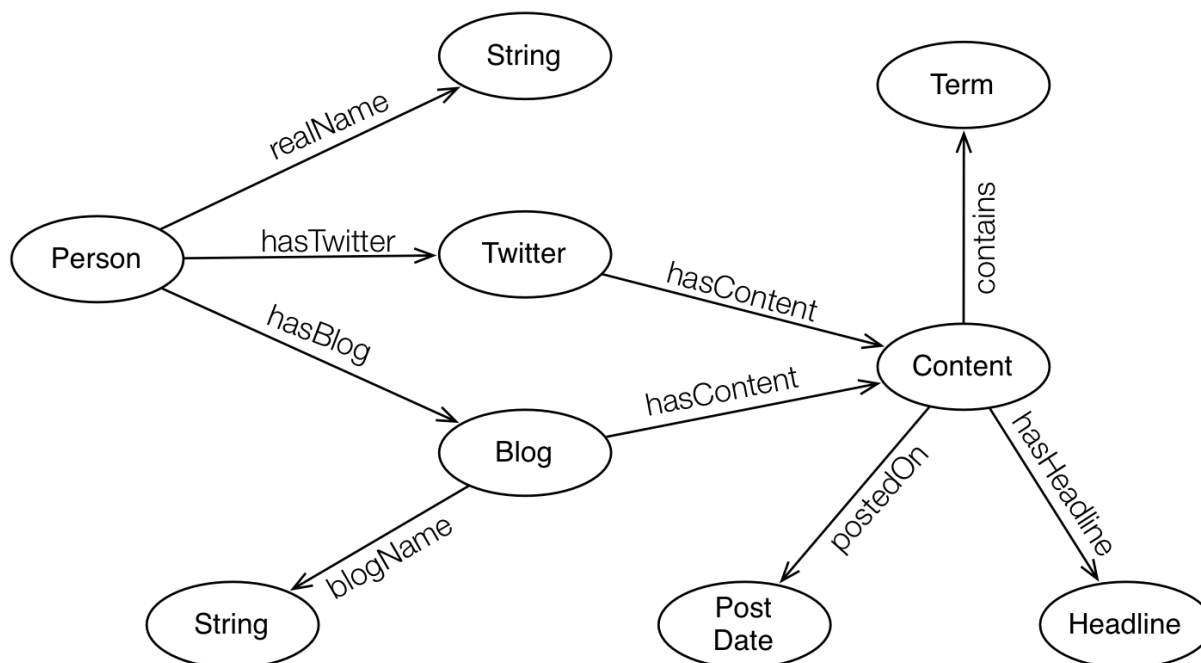
Εικόνα 1: Παράδειγμα σύνδεσης RDF δεδομένων από διαφορετικές βάσεις δεδομένων.

Όλα αυτά δε θα είχαν νόημα, εάν ο χρήστης δεν ήταν ικανός να επωφεληθεί άμεσα από αυτή την διαθέσιμη διασυνδεδεμένη πληροφορία. Για το σκοπό αυτό τα RDF δεδομένα μοντελοποιούνται με τρόπο εφάμιλλο με αυτόν των λογικών επιχειρημάτων. Συγκεκριμένα σε μορφή τριπλέτας που ακολουθείται η ιδέα "Υποκείμενο, Κατηγορούμενο, Αντικείμενο". Για την πραγματοποίηση ερωτημάτων, με σκοπό την προσπέλαση και κατ'επέκταση την επεξεργασία των δεδομένων, έχει δημιουργηθεί μια ειδική γλώσσα, η SPARQL.



Εικόνα 2: "Υποκείμενο, Κατηγορούμενο, Αντικείμενο": Το μοντέλο των RDF δεδομένων.

Συνδέοντας έννοιες με αυτόν τον τρόπο, προκύπτει ένα δίκτυο, το οποίο ολόένα και αυξάνεται σε μέγεθος κόμβων και ακμών, όσο φυσικά πληθαίνουν και τα διασυνδεδεμένα δεδομένα. Οι κόμβοι παριστάνουν τις συνδεόμενες έννοιες "Υποκείμενο" και "Αντικείμενο", ενώ οι ακμές την ιδιότητα "Κατηγορούμενο" που τις συνδέει. Αυτή η τυπική αναπαράσταση γνώσης ως ένα σύνολο εννοιών, σχέσεων και ιδιοτήτων, ονομάζεται "Οντολογία".



Εικόνα 3: Παράδειγμα διασυνδεδεμένων δεδομένων με τις αντίστοιχες τριπλέτες καταχωρημένες.

Παρόμοια λογική με αυτή των RDF δεδομένων, ακολουθεί και η γλώσσα SPARQL για την πραγματοποίηση ερωτημάτων. Στη δική μας περίπτωση θα ασχοληθούμε μόνο με το endpoint της DBpedia στο οποίο γράφονται ερωτήματα αποκλειστικά για τους χρήστες που θέλουν να "κατεβάσουν" μαζικά δεδομένα από τις σελίδες της Wikipedia. Ο κώδικας SPARQL βέβαια έχει την ίδια δομή και για τα υπόλοιπα endpoint.

1.2 Δεδομένα Κινηματογράφου

Ανάμεσα στις πολλές κατηγορίες δεδομένων που θα μπορούσε να επιλέξει κανείς μέσα από τις σελίδες της Wikipedia, για την εργασία αυτή επιλέχθηκαν δεδομένα που αφορούν κινηματογραφικές ταινίες και τους διανομείς τους. Αφορμή στάθηκε ο παρακάτω πίνακας που είναι αναρτημένος στον Ιστό και βρίσκεται στο σύνδεσμο:

<http://www.the-numbers.com/market/distributors>

Market Share for Each Distributor 1995-2016

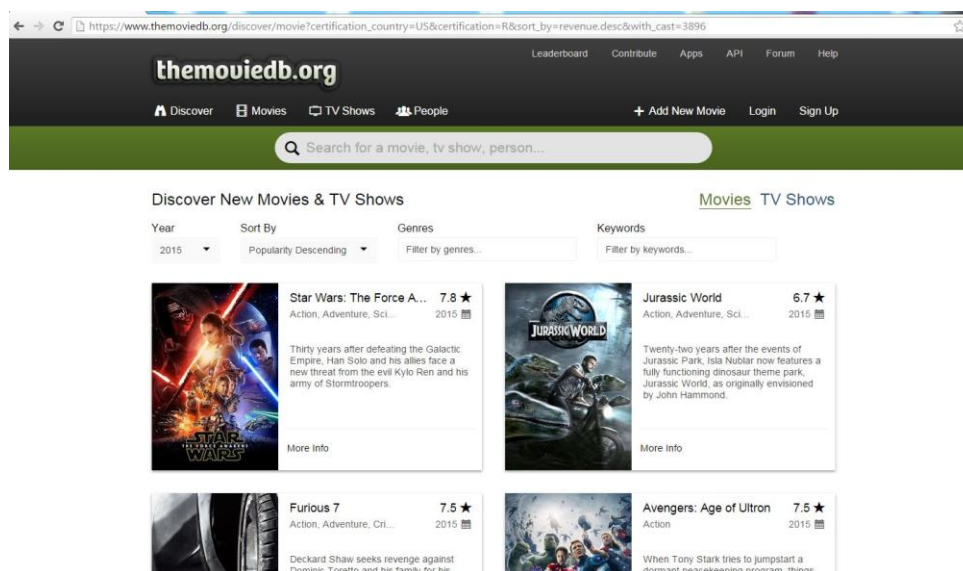
Rank	Distributor	Movies	Total Box Office	Tickets	Share
1	Warner Bros.	651	\$28,258,171,882	4,322,285,489	15.00%
2	Walt Disney	514	\$27,482,526,091	4,308,426,467	14.58%
3	Sony Pictures	620	\$23,916,506,873	3,731,087,578	12.69%
4	20th Century Fox	448	\$21,798,173,661	3,338,654,032	11.57%
5	Paramount Pictures	419	\$21,492,172,633	3,341,670,498	11.41%
6	Universal	415	\$21,201,624,702	3,240,516,936	11.25%
7	Lionsgate	317	\$6,764,679,906	901,157,601	3.59%
8	New Line	205	\$6,189,969,171	1,115,521,469	3.28%
9	Dreamworks SKG	75	\$4,083,905,398	724,398,160	2.17%
10	Miramax	382	\$3,836,565,824	714,036,742	2.04%
11	MGM	230	\$3,506,881,845	641,403,988	1.86%
12	Fox Searchlight	192	\$2,264,666,133	327,537,238	1.20%
13	Weinstein Co.	160	\$2,064,702,028	263,713,257	1.10%
14	Summit Entertainment	40	\$1,667,017,440	217,203,670	0.88%
15	Focus Features	135	\$1,630,613,559	226,919,191	0.87%
16	Sony Pictures Classics	438	\$1,052,989,603	163,914,206	0.56%
17	Miramax/Dimension	28	\$1,035,673,087	177,909,462	0.55%
18	Relativity	43	\$912,600,240	113,187,630	0.48%
19	Open Road	34	\$616,888,068	75,996,911	0.33%
20	Newmarket Films	26	\$450,931,979	72,910,768	0.24%
21	IFC Films	347	\$399,661,737	63,035,258	0.21%
22	FilmDistrict	16	\$382,933,030	47,522,112	0.20%
23	CBS Films	23	\$382,636,677	47,480,446	0.20%
24	Artisan	60	\$381,652,504	72,288,687	0.20%
25	Paramount Vantage	87	\$376,578,437	55,384,292	0.20%
26	Overture Films	23	\$342,696,132	45,769,788	0.18%
27	USA Films	28	\$296,282,431	53,244,929	0.16%
28	Gramercy	59	\$293,135,835	64,394,511	0.16%
29	Weinstein/Dimension	12	\$292,964,782	40,767,850	0.16%
30	IMAX Films	85	\$289,840,199	48,442,583	0.15%

Εικόνα 4: Πίνακας με τους κορυφαίους παραγωγούς ταινιών τα τελευταία 20 χρόνια.

Στην παραπάνω σελίδα μπορεί κανείς να βρει αρκετά δεδομένα, πρακτικά για όλους τους διανομείς ταινιών στον κόσμο. Επιπλέον, στους συνδέσμους της παρέχονται πολλές αναλυτικότερες πληροφορίες για κάθε χρόνο, αλλά και για κάθε ταινία, όπως συνολικό box office, εισιτήρια, budget κ.α.

Στη σελίδα themoviedb.org μπορεί κανείς να βρει δεδομένα για κάθε ταινία, οργανωμένα σε ένα πολύ φιλικό περιβάλλον για το χρήστη. Περιέχονται μεταξύ άλλων βαθμολογία χρηστών, συντελεστές παραγωγής, ηθοποιοί, οικονομικά δεδομένα, χρονολογίες και πολλά άλλα. Εξαιρετικό ενδιαφέρον όμως παρουσιάζει το γεγονός ότι τα δεδομένα είναι δομημένα με τέτοιο τρόπο, ώστε να δίνουν στο χρήστη πολύ ευέλικτα φίλτρα αναζήτησης. Ενώ σε μια

τυχαία σελίδα με κινηματογραφικά δεδομένα, ο χρήστης μπορεί να κάνει μια απλή αναζήτηση σε κάποιον τίτλο ή συντελεστή και η μηχανή αναζήτησης να του επιστρέψει εγγραφές που περιέχουν "αυστηρά" τους χαρακτήρες που χρησιμοποίησε, στη σελίδα αυτή μπορεί μέσω του widget "discover", να πληκτρολογήσει στο url κατάλληλο κώδικα με αποτέλεσμα η αναζήτηση του να είναι πιο "έξυπνη". Για παράδειγμα προσθέτοντας https://www.themoviedb.org/discover/movie?certification_country=US&certification=R&sort_by=revenue.desc&with_cast=3896 στο τέλος του url θα επιστραφούν στο χρήστη αποτελέσματα από τις ταινίες του Liam Neeson, που είχαν τα υψηλότερα κέρδη και είχαν δείκτη καταλληλότητας R.



Εικόνα 5: Παράδειγμα "έξυπνης" αναζήτησης στο themoviedb.org

Ενώ τα δεδομένα στο themoviedb.org είναι άρτια οργανωμένα και επιτρέπουν στο χρήστη να τα προσπελάσει με πολλές μεθόδους, δεν του δίνουν τη δυνατότητα να τα κατεβάσει μαζικά στον υπολογιστή του για να κάνει την ανάλυση που αυτός επιθυμεί. Το γεγονός αυτό, περιορίζει αρκετά όποιον θέλει να ασχοληθεί με μία "big data" ανάλυση πάνω στον κινηματογράφο.

Η διασημότερη, ίσως, ιστοσελίδα με δεδομένα κινηματογράφου και όχι μόνο είναι η Internet Movie Database (IMDb). Τα δεδομένα παρουσιάζονται με πολλές λεπτομέρειες και έχουν μεγαλύτερη ποικιλία από κάθε άλλη παρόμοια σελίδα στον ιστό. Μεταξύ άλλων η

πληροφορία που προσφέρουν περιλαμβάνει παραγωγούς, αναλυτική λίστα ηθοποιών, οικονομικά δεδομένα με γεωγραφική ταξινόμηση, βαθμολογίες από επίσημες πηγές αλλά και από τους ίδιους τους χρήστες της σελίδας και πολλά ακόμη.



Εικόνα 6: Παράδειγμα συνδέσμου ταινίας στο IMDb.

Σε ένα βαθμό, τη βάση δεδομένων του IMDb μπορεί κανείς να τη βρει και με τη μορφή διασυνδεδεμένων δεδομένων στο σύνδεσμο LinkedMDb.org. Συγκεκριμένα, τα ανοιχτά δεδομένα που προσφέρονται στο χρήστη περιλαμβάνουν τις κατηγορίες:

actor

content_rating

cinematographer

content_rating_system

country	film_film_company_relationship
director	film_film_distributor_relationship
dubbing_performance	film_format
editor	film_genre
film	film_job
film_art_director	film_location
film_awards_ceremony	film_production_designer
film_casting_director	film_regional_release_date
film_character	film_screening_venue
film_collection	film_series
film_company	film_set_designer
film_costume_designer	film_story_contributor
film_crew_gig	film_subject
film_crewmember	film_theorist
film_critic	interlink
film_cut	linkage_run
film_distribution_medium	music_contributor
film_distributor	performance
film_featured_song	personal_film_appearance
film_festival	personal_film_appearance_type
film_festival_event	producer
film_festival_focus	production_company
film_festival_sponsor	special_film_performance_type
film_festival_sponsorship	writer

Οι κατηγορίες αυτές εκ πρώτης όψεως μοιάζουν πολλές σε πλήθος και θα φανταζόταν κανείς ότι είναι αρκετές για κάποιον που θέλει να ξεκινήσει μια ανάλυση πάνω στα δεδομένα αυτά. Κάτι τέτοιο δεν ισχύει όμως. Λείπουν στοιχειώδεις κατηγορίες όπως το budget και το box office της κάθε ταινίας. Θα έλεγε κανείς ότι λείπει σχεδόν κάθε κατηγορία που ο τύπος μεταβλητής της θα ήταν αριθμητικός. Παρ' όλ' αυτά είναι μια αξιόλογη προσπάθεια, που αν

με τον καιρό εμπλουτίζεται, θα δοθεί η δυνατότητα στους χρήστες να παράγουν βάσεις δεδομένων που θα δώσουν ακριβή και ενδιαφέροντα αποτελέσματα.

SPARQL Explorer for <http://www.linkedmdb.org/sparql>

SPARQL:

```
SELECT * WHERE {
...
}
```

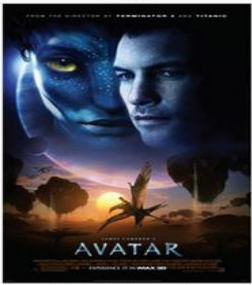
Results:

Must be application/x-www-form-urlencoded

Εικόνα 7: Η αρχική σελίδα του endpoint του LinkedMDB.

Η σελίδα που περιέχει πληροφορίες όχι μόνο για κινηματογράφο, αλλά για τα πάντα σχεδόν, είναι η γνωστή σε όλους Wikipedia. Η προσπάθεια σύνδεσης των δεδομένων της ξεκίνησε το 2007 μέσω του προγράμματος DBpedia και μέχρι τη στιγμή που γράφεται η εργασία αυτή, αποτελείται από 3 δισεκατομμύρια RDF τριπλέτες, εκ των οποίων περίπου οι 580 εκατομμύρια αφορούν τα άρθρα της αγγλόγλωσσας Wikipedia και οι υπόλοιπες αφορούν τις εκδόσεις των υπόλοιπων γλωσσών. Αξιοσημείωτο είναι ότι από τις 4.58 εκατομμύρια εκχωρήσεις, οι 4.22 εκατομμύρια έχουν ήδη καταταχθεί σε μία ενιαία οντολογία και οι 87 χιλιάδες από αυτές, αφορούν κινηματογραφικές ταινίες.

Ο χρήστης μπορεί να πάρει μια ιδέα για τα διασυνδεδεμένα δεδομένα που του προσφέρονται απλά βλέποντας το infobox στο πάνω δεξιά μέρος κάθε σελίδας της Wikipedia. Συγκεκριμένα για τις κινηματογραφικές ταινίες τα δεδομένα που σίγουρα δίνονται είναι σκηνοθέτης, παραγωγός, σεναριογράφος, πρωταγωνιστές, εταιρίες παραγωγής και διανομής, ημερομηνία κυκλοφορίας, διάρκεια, γλώσσα, budget και box office. Μια ποικιλία ικανοποιητικά μεγάλη να δώσει επιλογές σε επίδοξους αναλυτές κινηματογράφου και όχι μόνο.



Directed by	James Cameron
Produced by	James Cameron Jon Landau
Written by	James Cameron
Starring	Sam Worthington Zoe Saldana Stephen Lang Michelle Rodriguez Sigourney Weaver
Music by	James Horner
Cinematography	Mauro Fiore
Edited by	James Cameron John Refoua Stephen E. Rivkin
Production company	Lightstorm Entertainment Dune Entertainment Ingenious Film Partners
Distributed by	20th Century Fox
Release dates	December 10, 2009 (London premiere) December 17, 2009 (United Kingdom) December 18, 2009 (United States)
Running time	161 minutes ^[1]
Country	United States United Kingdom ^[2]
Language	English
Budget	\$237 million ^[3] \$9 million+ (re-release) ^[4]
Box office	\$2.788 billion ^{[5][6]}

Εικόνα 8: Παράδειγμα από infobox σελίδας της Wikipedia

Για να δει αναλυτικά ο χρήστης τα διασυνδεδεμένα δεδομένα που του προσφέρονται μέσω της DBpedia, αλλά και τις ιδιότητες των οντολογιών με τις οποίες συνδέονται, αρκεί να μπει στην αντίστοιχη σελίδα της. Θα έλεγε κανείς ότι για κάθε σελίδα της Wikipedia της οποίας τα δεδομένα είναι διασυνδεδεμένα, υπάρχει και μία αντίστοιχη σελίδα της DBpedia στην οποία μπορεί να δει κανείς ό,τι στοιχείο χρειάζεται. Ενας απλός τρόπος για να βρεθεί στη σελίδα αυτή, είναι να πληκτρολογήσει στο φυλλομετρητή του: dbpedia.org/page/X, όπου X, το τελευταίο κομμάτι της διεύθυνσης της αντίστοιχης σελίδας στη Wikipedia. Για παράδειγμα από το σύνδεσμο [https://en.wikipedia.org/wiki/Avatar_\(2009_film\)](https://en.wikipedia.org/wiki/Avatar_(2009_film)), προσθέτοντας το τελευταίο κομμάτι του συνδέσμου όπως περιγράφηκε παραπάνω ώστε να προκύψει [dbpedia.org/page/Avatar_\(2009_film\)](http://dbpedia.org/page/Avatar_(2009_film)). Η σελίδα που προκύπτει φαίνεται παρακάτω.

About: Avatar (2009 film)An Entity of Type : [work](#), from Named Graph : <http://dbpedia.org>, within Data Space : [dbpedia.org](#)

Avatar (marketed as James Cameron's Avatar) is a 2009 American epic science fiction film directed, written, co-produced, and co-edited by James Cameron, and starring Sam Worthington, Zoe Saldana, Stephen Lang, Michelle Rodriguez, Giovanni Ribisi, and Sigourney Weaver. The film is set in the mid-22nd century, when humans are mining a room-temperature superconductor called unobtainium on Pandora, a lush habitable moon of a gas giant in the Alpha Centauri star system.

Property	Value
dbo:Work/runtime	161.0 170.0 178.0
dbo:abstract	Avatar (marketed as James Cameron's Avatar) is a 2009 American epic science fiction film directed, written, co-produced, and co-edited by James Cameron, and starring Sam Worthington, Zoe Saldana, Stephen Lang, Michelle Rodriguez, Giovanni Ribisi, and Sigourney Weaver. The film is set in the mid-22nd century, when humans are mining a room-temperature superconductor called unobtainium on Pandora, a lush habitable moon of a gas giant in the Alpha Centauri star system. The expansion of the mining colony threatens the continued existence of a local tribe of Na'vi – a humanoid species indigenous to Pandora. The film's title refers to a genetically engineered Na'vi body with the mind of a remotely located human, and is used to interact with the natives of Pandora. Development of Avatar began in 1994, when Cameron wrote an 80-page treatment for the film. Filming was supposed to take place after the completion of Cameron's 1997 film Titanic, for a planned release in 1999, but according to Cameron, the necessary technology was not yet available to achieve his vision of the film. Work on the language of the film's extraterrestrial beings began in summer 2005, and Cameron began developing the screenplay and fictional universe in early 2006. Avatar was officially budgeted at \$237 million. Other estimates put the cost between \$280 million and \$310 million for production and at \$150 million for promotion. The film made extensive use of cutting edge motion capture filming techniques, and was released for traditional viewing, 3D viewing (using the RealD 3D, Dolby 3D, XpanD 3D, and IMAX 3D formats), and for "4D" experiences in select South Korean theaters. The stereoscopic filmmaking was touted as a breakthrough in cinematic technology. Avatar premiered in London on December 10, 2009, and was internationally released on December 16 and in the United States and Canada on December 18, to positive critical reviews, with critics highly praising its groundbreaking visual effects. During its theatrical run, the film broke several box office records and became the highest-grossing film of all time, as well as in the United States and Canada, surpassing Titanic, which had held those records for twelve years (and was also directed by Cameron). It also became the first film to gross more than \$2 billion. Avatar was nominated for nine Academy Awards, including Best Picture and Best Director, and won three, for Best Art Direction, Best Cinematography and Best Visual Effects. The film's home media release went on to break opening sales records and became the top-selling Blu-ray at the time. Following the film's success, Cameron signed with 20th Century Fox to produce three sequels, making Avatar the first of a planned tetralogy. The three sequels, all directed and co-written by Cameron, will be released each year starting from December 2017 to 2019.
dbo:budget	2.37E8 9000000.0
dbo:cinematography	dbr:Mauro_Fiore
dbo:director	dbr:James_Cameron
dbo:distributor	dbr:20th_Century_Fox
dbo:editing	- dbr:John_Refoua - dbr:Stephen_E_Rivkin
dbo:gross	2.787965087E9
dbo:musicComposer	dbr:James_Horner
dbo:producer	dbr:Jon_Landau_(film_producer)
dbo:runtime	9660.000000 (xsd:double) 10200.000000 (xsd:double) 10680.000000 (xsd:double)

Εικόνα 9: Παράδειγμα σελίδας της DBpedia.

Στην αριστερή στήλη μπορεί κανείς να δει όλες τις ιδιότητες που προσφέρονται για την ταινία αυτή, ενώ στη δεξιά τις τιμές που είναι καταχωρημένες στις ιδιότητες αυτές. Οι ιδιότητες αυτές χρησιμοποιούνται ως το predicate (κατηγορούμενο) στις τριπλέτες που προαναφέρθηκαν. Το υλικό των σελίδων αυτών θα χρησιμοποιηθεί στο επόμενο βήμα για την άντληση των δεδομένων.

Στον Ιστό υπάρχουν δεκάδες αν όχι εκατοντάδες σελίδες με ανοιχτό κινηματογραφικό περιεχόμενο. Είναι η δυνατότητα που μερικές από αυτές προσφέρουν στο χρήστη να προσπελάσει τα δεδομένα τους με έξυπνα φίλτρα ή ακόμα καλύτερα να τα κατεβάσει στον υπολογιστή του μαζικά που τις κάνει να ξεχωρίζουν από άλλες. Αν κάποιος θέλει να δει συγκεκριμένα στοιχεία μιας μόνο ταινίας είναι πλέον πολύ εύκολο. Για να μπορέσει να κάνει σύγκριση και μελέτη με άλλες αντίστοιχες ταινίες όμως, χρειάζεται να ψάξει παραπάνω, στις σελίδες που προαναφέρθηκαν μιας και η επιλογή αυτή είναι λιγότερο διαθέσιμη.

2. ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΕΞΟΡΥΞΗΣ, ΑΝΑΛΥΣΗΣ ΚΑΙ ΟΠΤΙΚΟΠΟΙΗΣΗΣ ΔΕΔΟΜΕΝΩΝ

Στο κεφάλαιο αυτό θα περιγραφούν λειτουργίες εργαλείων και λογισμικών

- Εξόρυξης (SPARQL)
- Καθαρισμού (Open Refine, SPSS)
- Ανάλυσης (SPSS, R-RStudio)
- Οπτικοποίησης (igraph, shiny, shinydashboard)

δεδομένων, δίνοντας μεγαλύτερο βάρος σε αυτά που όντως χρησιμοποιήθηκαν στην εργασία αυτή.

2.1 Εξόρυξη Δεδομένων

Η SPARQL (SPARQL Protocol and RDF Query Language) αποτελεί μια από τις σημαντικότερες τεχνολογίες του Semantic Web και είναι μια γλώσσα που χρησιμοποιείται για την πραγματοποίηση ερωτημάτων σε RDF γράφους, καθώς επίσης και για την ανάκτηση και το χειρισμό των δεδομένων που αυτοί περιέχουν. Τα ερωτήματα αυτά πραγματοποιούνται σε κάποιο SPARQL endpoint, το οποίο είναι μια υπηρεσία που δέχεται SPARQL ερωτήματα και επιστρέφει τα αποτελέσματα.

Η SPARQL μπορεί να πραγματοποιήσει τέσσερα διαφορετικά είδη ερωτημάτων:

- SELECT: για την εξαγωγή πρωτογενών δεδομένων από το endpoint, με τα αποτελέσματα είναι σε μορφή πίνακα
- CONSTRUCT: τα αποτελέσματα μετατρέπονται σε RDF
- ASK: τα αποτελέσματα είναι της μορφής true/false
- DESCRIBE: επιστρέφει έναν RDF γράφο που περιγράφει τις πηγές που βρέθηκαν

Σε κάθε ένα από αυτά τα ερωτήματα μπορεί να υπάρχει η WHERE για να περιορίσει τα αποτελέσματα, βάσει των patterns αναζήτησης που περιέχει. Οι μεταβλητές σε ένα ερώτημα έχουν μπροστά το χαρακτήρα ? (ή \$), ενώ συνηθίζεται να γίνεται ορισμός prefixes και να χρησιμοποιούνται αυτά μέσα στο ερώτημα, αντί για ολόκληρο το namespace της οντολογίας που αντιπροσωπεύουν

Ενα οποιοδήποτε SPARQL ερώτημα αποτελείται από τα παρακάτω βασικά στάδια:

1. **Δηλώσεις προθεμάτων:** Ορίζονται συντομεύσεις για τα URIs.
2. **Ορισμός βάσης δεδομένων:** Ορίζεται πάνω σε ποια δεδομένα θα γίνει το ερώτημα.
3. **Ορισμός αποτελεσμάτων:** Ορίζονται οι πληροφορίες που θα επιστραφούν από το ερώτημα.
4. **Κορμός ερωτήματος:** Το βασικότερο κομμάτι που ορίζεται τι ακριβώς θέλει να ρωτήσει ο χρήστης από την παραπάνω βάση δεδομένων.
5. **Τροποποιητές ερωτήματος:** Εντολές αποκοπής, ταξινόμησης και γενικά τακτοποίησης των αποτελεσμάτων.

Μια γενική μορφή ερωτήματος βασισμένη στα παραπάνω βήματα είναι:

Δηλώσεις προθεμάτων

```
PREFIX foo: <http://example.com/resources/>
...
```

Ορισμός βάσης δεδομένων

```
FROM ...
```

Ορισμός αποτελεσμάτων

```
SELECT ...
```

Κορμός ερωτήματος

```
WHERE {
    ...
}
```

Τροποποιητές ερωτήματος

```
ORDER BY ...
```

Η εντολή PREFIX συνοδεύεται από συνήθως τρία ή τέσσερα γράμματα και ένα URI δημιουργεί μια συντόμευση για το URI αυτό. Ο χρήστης δηλαδή όταν θέλει να το προσπελάσει χρησιμοποιεί μόνο τα γράμματα που ορίζει και όχι ολόκληρο το σύνδεσμο.

Η εντολή FROM συνοδεύεται πάντα από ένα URI που περιέχει τη βάση δεδομένων πάνω στην οποία γίνεται το ερώτημα. Ειδικά για το endpoint της DBpedia, μπορεί να παραλειφθεί, μιας και η βάση δεδομένων είναι αποκλειστικά η Wikipedia.

Η εντολή SELECT συνοδεύεται από τις μεταβλητές που θα περιέχει ο πίνακας αποτελεσμάτων. Αξίζει να σημειωθεί πως με την εντολή SELECT DISTINCT απαλείφονται οι διπλοεγγραφές που ενδεχομένως προκύπτουν.

Η εντολή WHERE ακολουθείται από άγκιστρα που περιέχουν όλες τις συνδέσεις των ορισμένων μεταβλητών με τα δεδομένα του συνόλου δεδομένων που επιθυμεί ο χρήστης. Η βασικότερη δομή είναι απλά η σύνδεση δύο μεταβλητών με μια ιδιότητα της οντολογίας, αλλά σε πολυπλοκότερα ερωτήματα μπορούν να χρησιμοποιηθούν και παραπάνω από μία ιδιότητες. Ο χρήστης μπορεί επίσης στο στάδιο αυτό να περιορίσει τα αποτελέσματα του με την εντολή FILTER.

Στο τελευταίο στάδιο γίνεται η τακτοποίηση των αποτελεσμάτων με διάφορες εντολές. Από τις βασικότερες είναι η ORDER BY που συνοδεύεται από μία ή περισσότερες εκ των ορισμένων μεταβλητών σύμφωνα με τις οποίες θα γίνει η ταξινόμηση. Για αύξουσα σειρά χρησιμοποιείται ταυτόχρονα η ASC ενώ για φθίνουσα η DESC (ascending - descending). Πολύ βασική επίσης είναι η εντολή LIMIT που συνοδεύεται απλά από έναν αριθμό και περιορίζει το πλήθος των εμφανιζόμενων αποτελεσμάτων στον αριθμό αυτό.



Εικόνα 10: Η αρχική σελίδα του endpoint της DBpedia.

Τα αποτελέσματα επιστρέφονται στο χρήστη σε διάφορες μορφές όπως XML, JSON, CSV/TSV, RDF, HTML. Ειδικά το endpoint της DBpedia προσφέρει τα αποτελέσματα μόνο σε JSON και XML μορφή καθώς και σε μορφή πίνακα μόνο για ανάγνωση με την επιλογή "Browse".

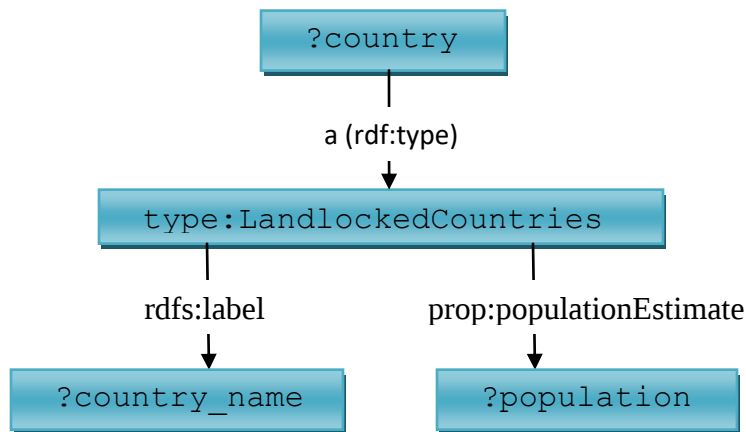
Παρακάτω εξηγείται ένα απλό παράδειγμα ερωτήματος SPARQL που επιστρέφει όλες τις χώρες που δε βρέχονται από θάλασσα και έχουν πληθυσμό μεγαλύτερο των 15000000:

```
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
    ?country a type:LandlockedCountries ;
        rdfs:label ?country_name ;
        prop:populationEstimate ?population .
    FILTER (?population > 15000000 &&
        langMatches(lang(?country_name), "EN")) .
} ORDER BY DESC(?population)
```

Στις δύο πρώτες γραμμές του ερωτήματος ορίζονται οι συντομεύσεις με τις εντολές PREFIX που αναφέρθηκαν.

Στην εντολή `SELECT` ζητούνται δύο μεταβλητές με όνοματα `country_name` και `population`.

Στον κορμό του ερωτήματος μέσα στην εντολή `WHERE` ορίζεται η μεταβλητή `country` ως μια εκχώρηση `type:LandlockedCountries`, συνδεδεμένη με την ιδιότητα `a` που είναι συντόμευση για την κοινή ιδιότητα `rdf:type`. Οι εκχωρήσεις `type:LandlockedCountries` έχουν την ιδιότητα `rdfs:label` όπου εκχωρείται το όνομα τους και αυτό είναι που ο χρήστης εκχωρεί στη μεταβλητή `country_name`. Επιπλέον στην ιδιότητα `prop:populationEstimate` καταχωρείται ο πληθυσμός της χώρας και ο χρήστης χρησιμοποιεί την ιδιότητα αυτή για να τον εκχωρήσει στη μεταβλητή του με όνομα `population`. Το παρακάτω διάγραμμα ροής περιγράφει τις τριπλέτες που χρησιμοποιήθηκαν αναλυτικά.



Στην εντολή `FILTER` περιορίστηκαν οι εκχωρήσεις της μεταβλητής `?population` σε αυτές που η τιμή τους είναι μεγαλύτερη των 15000000 καθώς και περιορίζει τα αποτελέσματα της μεταβλητής `country_name` σε αυτά που έχουν αγγλική γλώσσα, για να μην υπάρξουν μεταφρασμένες διπλοεγγραφές. Αυτό πετυχαίνεται με την εντολή `langMatches`. Οι δύο αυτές συνθήκες ικανοποιούνται ταυτόχρονα συνδέοντας τις με τη λογική έκφραση `"&&"`.

Τέλος στην περιοχή ταξινόμησης ζητείται μέσω της `ORDER BY DESC` να εμφανιστούν τα αποτελέσματα σε φθίνουσα σειρά βάσει του πληθυσμού τους.

Τα αποτελέσματα εμφανίζονται με την επιλογή `"Browse"` ως εξής:

SPARQL Explorer for <http://dbpedia.org/sparql>

SPARQL:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdfs-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
  ?country a type:LandlockedCountries ;
    rdfs:label ?country_name ;
    prop:populationEstimate ?population .
  FILTER (?population > 15000000 &&
    langMatches(lang(?country_name), "EN")) .
} ORDER BY DESC(?population)

```

Results:

SPARQL results:

country_name	population
"Ethiopia"@en	87952991
"Afghanistan"@en	31822848
"Uzbekistan"@en	30185000
"Burkina Faso"@en	17322796
"Malawi"@en	16407000

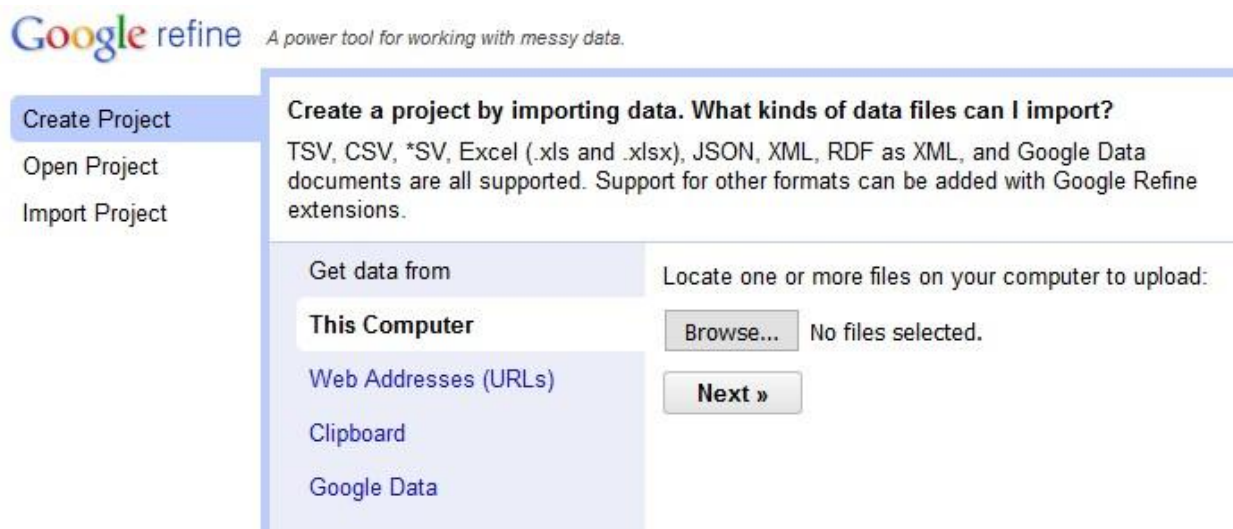
Εικόνα 11: Αποτελέσματα του παραδείγματος SPARQL

2.2 Καθαρισμός Δεδομένων

Το Open Refine είναι ένα δυναμικό εργαλείο για όποιον δουλεύει με ακατάστατα δεδομένα. Ειδικά ενδείκνυται για να:

- Ανιχνεύει και να καθαρίζει ασυνέχειες
- Μετατρέπει δεδομένα από τη μία δομή ή μορφή σε άλλη
- Συνδέει μεταβλητές μέσα από τα δεδομένα του χρήστη σε βάσεις δεδομένων.

Τα αρχεία που υποστηρίζει είναι της μορφής TSV, CSV, *SV, Excel (.xls και .xlsx), JSON, XML, RDF as XML, και Google Data.



Εικόνα 12: Η αρχική σελίδα του Open Refine.

Αφού ο χρήστης περάσει το αρχείο στο Open Refine, τα δεδομένα του εμφανίζονται με τη μορφή υπολογιστικού φύλλου. Οι επιλογές του στο σημείο αυτό είναι πάρα πολλές, εδώ θα περιγραφούν οι βασικότερες.

All	results - bindings - __anonymous__ - movie - v	results - bindings - __anonymous__ - dist - value	results - bindin	results - bindings - __anonymous__
1.	http://dbpedia.org/resource/The_Road_to_El_Dorado	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	95000000	76432727
2.	http://dbpedia.org/resource/The_Road_to_El_Dorado	http://dbpedia.org/resource/Category:DreamWorks_Animation_animated_films	95000000	76432727
3.	http://dbpedia.org/resource/Road_Trip_(film)	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	16000000	119754278
4.	http://dbpedia.org/resource/An_Everlasting_Piece	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	14000000	75228
5.	http://dbpedia.org/resource/The_Contender_(2000_film)	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	20000000	22361811
6.	http://dbpedia.org/resource/The_Legend_of_Bagger_Vance	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	60000000	39459427
7.	http://dbpedia.org/resource/Almost_Famous	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	60000000	47383689
8.	http://dbpedia.org/resource/Gladiator_(2000_film)	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	103000000	457600000
9.	http://dbpedia.org/resource/The_Road_to_El_Dorado	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	95000000	76432727
10.	http://dbpedia.org/resource/The_Road_to_El_Dorado	http://dbpedia.org/resource/Category:DreamWorks_Animation_animated_films	95000000	76432727
11.	http://dbpedia.org/resource/Small_Time_Crooks	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	18000000	29934477
12.	http://dbpedia.org/resource/Cast_Away	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	90000000	429632142
13.	http://dbpedia.org/resource/What_Lies_Beneath	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	100000000	291420351
14.	http://dbpedia.org/resource/Chicken_Run	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	45000000	224834564
15.	http://dbpedia.org/resource/Chicken_Run	http://dbpedia.org/resource/Category:DreamWorks_Animation_animated_films	45000000	224834564
16.	http://dbpedia.org/resource/Meet_the_Parents	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	55000000	330444045
17.	http://dbpedia.org/resource/What_Lies_Beneath	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	100000000	291420351
18.	http://dbpedia.org/resource/Monsters_vs_Allies	http://dbpedia.org/resource/Category:DreamWorks_Pictures_films	175000000	381509870

Εικόνα 13: Δεδομένα σε μορφή υπολογιστικού φύλλου στο Open Refine.

Οι περισσότερες επιλογές του χρήστη εμφανίζονται όταν αυτός ανοίξει το μενού επιλογών κάθε στήλης. Του δίνεται η δυνατότητα να κάνει ταξινόμηση σύμφωνα με όποια στήλη θέλει, να προσθέσει και να αφαιρέσει στήλες, να κάνει αναζήτηση εγγραφών, να φιλτράρει τα αποτελέσματα, ακόμα και να συνδέσει τα δεδομένα του με τα ήδη υπάρχοντα στο διαδίκτυο και κατά συνέπεια να τα εμπλουτίσει με καινούρια στοιχεία.

Από τις πιο απλές εντολές είναι η sort. Μόλις την επιλέξει ο χρήστης έχει τη δυνατότητα να ταξινομήσει τα δεδομένα του σε αύξουσα ή φθίνουσα αριθμητική ή αλφαβητική σειρά. Τυχόν κενές ή εσφαλμένες εγγραφές τοποθετούνται στην αρχή ή στο τέλος των δεδομένων, πάλι μετά από επιλογή του χρήστη.

Επίσης απλή εντολή είναι η text filter. Εδώ εμφανίζεται ένα παράθυρο στα αριστερά της εφαρμογής, στο οποίο ο χρήστης μπορεί να πληκτρολογήσει όποιους χαρακτήρες θέλει, και το Open Refine να του εμφανίσει μόνο τις εγγραφές που περιέχουν τους χαρακτήρες αυτούς.

Στην κατηγορία view δίνονται στο χρήστη 4 επιλογές. Να καταργήσει τη στήλη αυτή, να καταργήσει όλες τις στήλες εκτός από αυτή, να καταργήσει όλες τις στήλες στα αριστερά ή στα δεξιά της στήλης αυτής. Σε συνδυασμό με την ταξινόμηση στηλών που προσφέρεται στην επιλογή All, δημιουργείται ένα πολύ ισχυρό εργαλείο μαζικού καθαρισμού άχρηστης πληροφορίας.

Λίγο πολύ τις παραπάνω επιλογές μπορεί κανείς να τις βρει σε οποιαδήποτε εφαρμογή υπολογιστικού φύλλου. Αυτές που ακολουθούν όμως είναι που δίνουν στο Open Refine την προστιθέμενη αξία που κάνει τη διαφορά.

Στην κατηγορία facet ο χρήστης μπορεί μαζικά στα δεδομένα του να χωρίσει σε κατηγορίες όλες τις εγγραφές ανάλογα με το πόσες φορές εμφανίζεται η καθεμία. Τις κατηγορίες αυτές μπορεί στη συνέχεια να τις συγχωνεύσει σε μία ή να τις επεξεργαστεί μεμονωμένα. Στην πιο συνηθισμένη επιλογή, το text facet, αυτόματα εμφανίζεται ένα παράθυρο στα αριστερά της εφαρμογής με όλες τις διαφορετικές εγγραφές που υπάρχουν στη στήλη που επιλέχθηκε. Οι εγγραφές αυτές συνοδεύονται και από έναν αριθμό ο οποίος δηλώνει πόσες φορές εμφανίζεται η συγκεκριμένη εγγραφή σε όλη τη στήλη.

Facet / Filter

Undo / Redo

395 records

Extensions: Freebase

Refresh

Reset All

Remove All

results - bindings - __anonymous__ - change

movie - value

112 choices Sort by: name count Cluster

http://dbpedia.org/resource/A_Beautiful_Mind_(film) 2

http://dbpedia.org/resource/A_Thousand_Words_(film) 2

http://dbpedia.org/resource/A_I_Artificial_Intelligence 2

http://dbpedia.org/resource/Almost_Famous 2

http://dbpedia.org/resource/An_Everlasting_Piece 1

http://dbpedia.org/resource/...

All

results - bindings - __anonymous__ - movie - v

results - bindings - __anonymous__ - dist - value

results - bindin

results - bindin

results - bindings - __anonymous__

1. Facet

2. Text filter

3. Edit cells

4. Edit column

5. Transpose

6. Sort...

7. View

8. Reconcile

9. http://dbpedia.org/resource/Gladiator_(2000_film)

10. http://dbpedia.org/resource/The_Road_to_El_Dorado

11. http://dbpedia.org/resource/Small_Time_Crooks

12. http://dbpedia.org/resource/Cast_Away

13. http://dbpedia.org/resource/What_Lies_Beneath

14. http://dbpedia.org/resource/Chicken_Run

15. http://dbpedia.org/resource/Chicken_Run

16. http://dbpedia.org/resource/Meet_the_Parents

17. http://dbpedia.org/resource/What_Lies_Beneath

Text facet

Numeric facet

Timeline facet

Scatterplot facet

Custom text facet...

Custom numeric facet...

Customized facets

Almost_Famous

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_films

http://dbpedia.org/resource/Category:DreamWorks_Pictures_fil

Εικόνα 14: Η επιλογή Text facet

Μία τέτοια δυνατότητα είναι πολύ χρήσιμη, καθώς πολλές εγγραφές μπορεί να έχουν το ίδιο περιεχόμενο, αλλά να είναι διαφορετικά διατυπωμένες και κατά συνέπεια ο υπολογιστής να τις αντιμετωπίζει σαν διαφορετικές. Για παράδειγμα αν κάποιος θέλει να δει την εγγραφή Dolce & Gabbana, σε βάση δεδομένων μεγάλου όγκου θα τη βρει με πολλές διαφορετικές διατυπώσεις, όπως dolce & gabbana, dolce and gabbana, D & G, D and G και πολλές άλλες, ακόμα και χωρίς τέλεια ορθογραφία. Μέσω του Open Refine μπορεί να τις εντοπίσει όλες και να τις επεξεργαστεί μαζικά, ώστε να είναι όλες με τον ίδιο τρόπο εκχωρημένες. Από εκεί και πέρα μπορεί να λειτουργήσει και σαν φίλτρο, αφού απλά επιλέγοντας την κατηγορία που θέλει, ο χρήστης μπορεί να εμφανίσει μόνο τις σειρές που περιέχουν την εγγραφή που επέλεξε και να την απομονώσει από τις άλλες.

Πολλές ασυνέχειες μπορεί να είναι πολύ μικρού μεγέθους όπως η προσθήκη ενός κενού στο τέλος της εγγραφής, η διαφορά πεζών κεφαλαίων και άλλα. Μέσω της επιλογής Edit cells→ Transform ή Edit cells→ Common transforms ο χρήστης μπορεί να επεξεργαστεί μαζικά τέτοιου είδους ατέλειες. Του δίνεται η δυνατότητα μεταξύ άλλων να μετατρέψει μια ολόκληρη στήλη σε κεφαλαία ή σε πεζά, να σβήσει συνεχόμενα κενά ανάμεσα σε λέξεις ή κενά στην αρχή και στο τέλος των εγγραφών, ή να μετατρέψει τα δεδομένα του από μια κατηγορία μεταβλητής σε άλλη.

Εντυπωσιακά αποτελέσματα προσφέρει το Open Refine σε συνδυασμό με τα διασυνδεδεμένα δεδομένα και τη σελίδα Freebase. Με τις επιλογές Add column by fetching URLs, Add columns from Freebase και Reconcile ο χρήστης μπορεί να εμπλουτίσει τα δεδομένα του μόνο με μερικά κλικ. Ας πάρουμε, για παράδειγμα, ότι ο χρήστης έχει φτιάξει έναν πίνακα στο Excel με τις αγαπημένες του ταινίες και τον έχει φορτώσει στο Open Refine. Με την επιλογή Reconcile, "σκανάρονται" οι εκχωρήσεις του και γίνεται ταυτοποίηση για τον τύπο τους, στην προκειμένη περίπτωση "Film". Οι εγγραφές του χρήστη αντιστοιχίζονται με τις αντίστοιχες σελίδες ταινιών στη Freebase και πλέον έχουν τη μορφή συνδέσμου που προορισμό έχει την εκάστοτε σελίδα. Στη συνέχεια ο χρήστης με την επιλογή Add columns from Freebase μπορεί να προσθέσει στήλες στα δεδομένα του οι οποίες να περιέχουν όποια ιδιότητα ταινιών αυτός επιθυμεί, για παράδειγμα σκηνοθέτη, σεναριογράφος, εταιρία παραγωγής και διανομής, ημερομηνία κυκλοφορίας και πολλά άλλα. Η Freebase είναι μόνο μία από τις πολλές σελίδες που προσφέρουν την υπηρεσία Reconcile. Η DBpedia είναι ένα ακόμα παράδειγμα.

2.3 Ανάλυση Δεδομένων

Το SPSS είναι ένα από τα πιο διαδεδομένα λογισμικά που υπάρχουν για εφαρμογές στατιστικής και όχι μόνο. Οι λειτουργίες που προσφέρει στο χρήστη μπορούν από μόνες τους να γεμίσουν ολόκληρα βιβλία. Για το λόγο αυτό η αναφορά του στην εργασία αυτή θα είναι πολύ περιορισμένη σε σχέση με το συνολικό πλήθος των δυνατοτήτων που προσφέρει. Η βασική μορφή αρχείου SPSS είναι η sav, αλλά υποστηρίζει και όλες σχεδόν τις υπόλοιπες μορφές αρχείων υπολογιστικών φύλλων.

Το SPSS αποτελείται από δύο βασικές καρτέλες, το Data view και το Variable view. Στην πρώτη, ο χρήστης μπορεί να δει τα δεδομένα του σε μορφή πίνακα, με τις μεταβλητές του εκχωρημένες κατά στήλες. Πάνω στα δεδομένα της καρτέλας αυτής, ο χρήστης εφαρμόζει όλες τις μεθόδους που του προσφέρονται και θα έλεγε κανείς ότι είναι η βασικότερη καρτέλα από τις δύο. Αυτό όμως καθόλου δε μειώνει τη χρησιμότητα της "Variable view", καθώς εκεί είναι που ο χρήστης θα ορίσει τον τύπο των μεταβλητών του (αριθμητικές, χαρακτήρες) και άλλες λεπτομέρειες όπως τα ονόματα των τους, το πλήθος των δεκαδικών ψηφίων των τιμών τους, το πλάτος των χαρακτήρων τους και πολλά ακόμη.

	Name	Type	Width	Decimals	Label	Values	Missing	Columns	Align	Measure	Role
1	Movie	String	102	0		None	None	40	Left	Nominal	Input
2	Distributor	String	16	0		None	None	11	Left	Nominal	Input
3	Budget	Numeric	8	0		None	None	8	Right	Scale	Input
4	BoxOffice	Numeric	8	0		None	None	8	Right	Scale	Input
5	Year	Numeric	8	0		None	None	8	Right	Scale	Input
6	Difference	Numeric	8	0		None	None	8	Right	Scale	Input
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											

Εικόνα 15: Η Variable View του SPSS.

Το παραπάνω παράθυρο (Data Editor) είναι μόνο το ένα από τα τρία που χρησιμοποιεί το SPSS. Τα άλλα δύο είναι τα Syntax Editor και Output Viewer. Στο πρώτο, ο χρήστης μπορεί να πληκτρολογήσει εντολές για να ανοίξει και να αποθηκεύσει αρχεία, να επεξεργαστεί δεδομένα και να παράγει αποτελέσματα. Στο Output Viewer εμφανίζονται όλα τα αποτελέσματα (πίνακες, στατιστικοί έλεγχοι, γραφήματα και άλλα) που ο χρήστης επέλεξε να τρέξει για τα δεδομένα του.

Στο επάνω μέρος του παραθύρου Data Editor βρίσκει κανείς όλες τις δυνατότητες που του προσφέρονται, ανά κατηγορία. Οι πιο βασικές καρτέλες είναι οι:

- Data: Ο χρήστης προετοιμάζει τα δεδομένα του για επεξεργασία, με διάφορες ταξινομήσεις και καθαρισμούς.
- Transform: Ο χρήστης χρησιμοποιεί τα ήδη υπάρχουσες μεταβλητές του για να παράγει νέες με απλούς και πολύπλοκους τρόπους, όπως για παράδειγμα αριθμητικές πράξεις και συναρτήσεις.

- **Analyze:** Στην καρτέλα αυτή προσφέρεται στο χρήστη μια ποικιλία στατιστικών ελέγχων, παλινδρομήσεων και μοντέλων για να εφαρμόσει στα δεδομένα του. Τα αποτελέσματα εμφανίζονται στο Output Viewer.

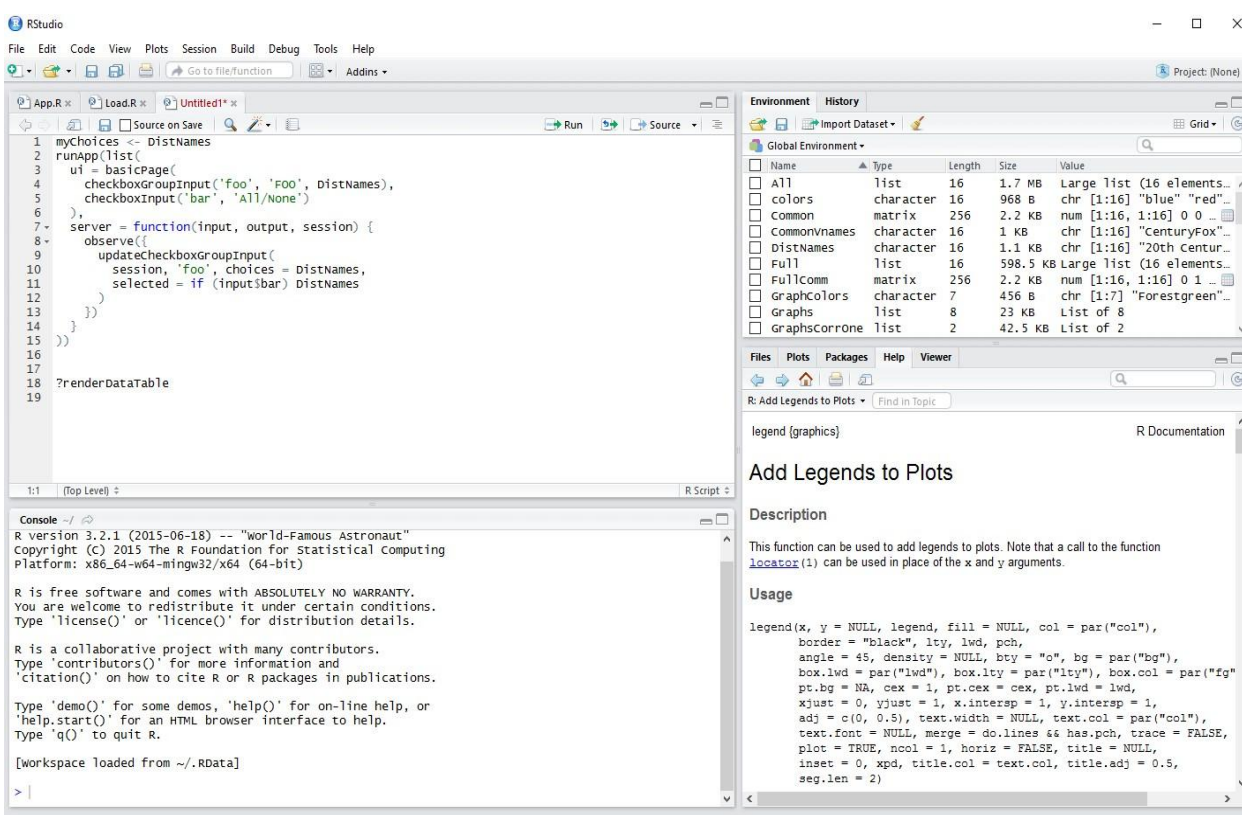
- **Graphs:** Εδώ ο χρήστης μπορεί να σχεδιάσει αυτόματα γραφήματα για τα δεδομένα του. Οι επιλογές του είναι πολλές και τα σχέδια μπορεί να είναι και τρισδιάστατα. Τα αποτελέσματα εμφανίζονται επίσης στο Output Viewer από όπου ο χρήστης μπορεί να τα επεξεργαστεί παραπάνω.

Γενικά, το SPSS είναι ένα εργαλείο που συνεχώς εξελίσσεται και οι δυνατότητες που προσφέρει ολοένα και αυξάνονται. Περιέχει συνήθως πολύ περισσότερα από όσα χρειάζεται κάποιος για να επεξεργαστεί και να εμπλουτίσει τα δεδομένα του, να επιλέξει από μια μεγάλη ποικιλία στατιστικών ελέγχων και να τα παρουσιάσει με τρόπο όμορφο και φιλικό στο κοινό όπου αυτός θέλει.

Μία από τις πιο διαδεδομένες γλώσσες προγραμματισμού για ανάλυση δεδομένων των τελευταίων χρόνων είναι η R. Στην απλή έκδοση της εμφανίζεται στο χρήστη μόνο το παράθυρο εκτέλεσης εντολών, ενώ ο ίδιος μπορεί να ανοίξει ένα ή περισσότερα παράθυρα ακόμη με scripts εντολών που έχει αποθηκευμένα στον υπολογιστή του. Τα διαγράμματα και οι γράφοι, προβάλλονται σε ξεχωριστά παράθυρα που εμφανίζονται με την εκτέλεση της εκάστοτε εντολής.

Η πιο εξελιγμένη έκδοση της R, που είναι και αυτή που χρησιμοποιήθηκε στη συγκεκριμένη εργασία, είναι η R Studio. Προϋποθέτει ο χρήστης να έχει ήδη την απλή έκδοση εγκατεστημένη στον υπολογιστή του. Προσφέρει στο χρήστη ένα πιο “γεμάτο” και φιλικό περιβάλλον, με τέσσερα βασικά παράθυρα. Το ένα φυσικά είναι το παράθυρο εκτέλεσης που ο χρήστης μπορεί να τρέξει μεμονωμένες εντολές. Τα scripts του αυτή τη φορά δεν ανοίγουν το καθένα σε ξεχωριστό παράθυρο, αλλά συγχωνεύονται όλα σε ένα, δίνοντας τη δυνατότητα στο χρήστη να πλοηγηθεί σε αυτά με καρτέλες, δημιουργώντας μια καινούρια καρτέλα για κάθε script που ανοίγει, με το αντίστοιχο όνομα. Τα διαγράμματα και οι γράφοι εμφανίζονται αυτή τη φορά σε παράθυρο εντός του περιβάλλοντος του προγράμματος στο οποίο προσφέρονται στο χρήστη μερικές ακόμα δυνατότητες, όπως το να κατεβάσει άμεσα πακέτα εντολών, τα παράθυρα βοήθειας και να πλοηγηθεί στους φακέλους του υπολογιστή

του για να ανοίξει όποιο αρχείο θέλει εύκολα. Τέλος, υπάρχει ένα ξεχωριστό παράθυρο με όλες τις εκχωρήσεις που έχει κάνει ο χρήστης και τον τύπο τους (αριθμός, πίνακας, λίστα, και άλλα). Ο χρήστης μπορεί επιπλέον να φορτώσει μεταβλητές από προηγούμενη δουλειά του για να τις επεξεργαστεί περαιτέρω, επιλογή που τον γλυτώνει από πολύ κόπο μιας και δε θα χρειάζεται κάθε φορά που ανοίγει τη δουλειά που έχει κάνει, να ξανατρέχει όλες τις εντολές του από την αρχή.



Εικόνα 16: Το περιβάλλον της R Studio.

Παρόλο που η R από μόνη της έχει έναν ικανοποιητικό αριθμό εντολών, τη συντριπτική πλειοψηφία των συναρτήσεων της μπορεί να τη βρει κανείς στα πακέτα (packages) εντολών που υπάρχουν στον Ιστό. Τα περισσότερα από αυτά τα πακέτα έχουν κατασκευαστεί από τους ίδιους τους χρήστες και έχουν διαμοιραστεί στους υπόλοιπους, γεγονός που έχει δημιουργήσει μια πολύ ισχυρή κοινότητα προγραμματιστών R που ολοένα αυξάνεται σε πλήθος και

ατόμων, και υλικού. Για να εγκαταστήσει ένα πακέτο το μόνο που χρειάζεται είναι το όνομα του. Για παράδειγμα, για να προσθέσει το πακέτο με όνομα `example`, αρκεί να τρέξει την εντολή: `install.packages("example")`. Απαιτείται φυσικά να έχει πρόσβαση στο διαδίκτυο για να ληφθούν τα απαραίτητα αρχεία, κάτι που γίνεται αυτόματα. Μέτα την πρώτη εγκατάσταση τους, και κάθε φορά που ο χρήστης θέλει να χρησιμοποιήσει τις εντολές του πακέτου αυτού, τρέχει απλά την εντολή `library(example)` που φορτώνει τις συναρτήσεις από τον υπολογιστή του χρήστη χωρίς να τις ξανακατεβάξει από τον Ιστό.

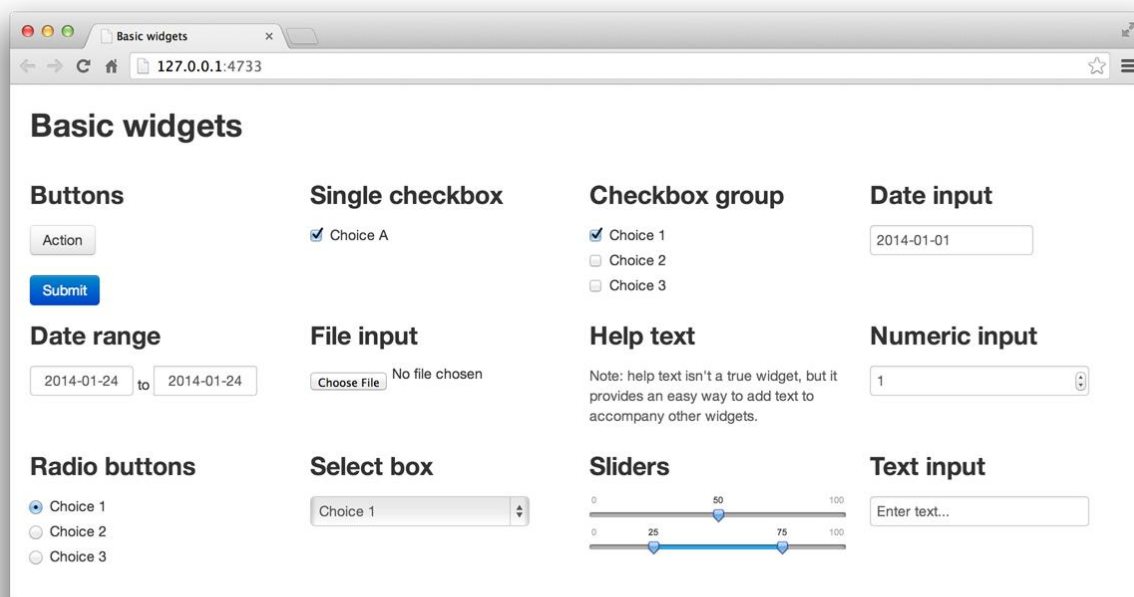
2.4 Οπτικοποίηση Δεδομένων και Αποτελεσμάτων

Η βιβλιοθήκη "shiny" της R δίνει τη δυνατότητα στο χρήστη να δημιουργεί δικές του εφαρμογές οπτικοποίησης δεδομένων από το μηδέν, καθώς και την ευελιξία να πλοηγηθεί σε αυτές, με πολύ φιλικό τρόπο.

Για να το πετύχει αυτό χρειάζεται δύο βασικές συναρτήσεις, τη `shinyServer()` και τη `shinyUI()` (User Interface). Οι δύο αυτές διαδικασίες συνδυάζονται με την εξής λογική. Στη `shinyUI()` ορίζεται το περιβάλλον της εφαρμογής, δηλαδή ότι θα προβάλλεται στο χρήστη κάθε στιγμή που πλοηγείται σε αυτή. Συνήθως περιλαμβάνει widgets για επεξεργασία ή φιλτράρισμα, γραφήματα και γράφους γενικότερα. Η διάταξη τους καθορίζεται επίσης εδώ. Σε κάθε widget ή plot καταχωρείται επίσης μία ετικέτα. Στη συνάρτηση `shinyServer()` συνδέονται οι ετικέτες αυτές με τα δεδομένα που έχει εκχωρήσει ο χρήστης έτσι ώστε όποια επιλογή στα widgets και να κάνει ο χρήστης, να πάρει το αντίστοιχο αποτέλεσμα.

Για παράδειγμα, έστω ότι έχουμε δώσει την ετικέτα "slider" σε ένα widget μπάρας κύλισης. Η τιμή που θέλει ο χρήστης μεταβάλλεται ανάλογα με το πόσο αριστερά ή δεξιά την κινήσει. Σε κάθε σημείο της όμως είναι ένας σταθερός αριθμός. Ο αριθμός αυτός χρησιμοποιείται στη `shinyServer()` μέσω της εντολής `input$slider` για να ξεχωρίσει το γράφημα, πίνακα, γράφο ή οτιδήποτε άλλο θέλει να συνδέσει ο χρήστης στην τιμή αυτή. Για να το πετύχει αυτό, χρησιμοποιεί την εντολή `output$X`, όπου `X` μια νέα ετικέτα που θέλει να δώσει στο αποτέλεσμα του. Τέλος, για να εμφανιστεί στην εφαρμογή το αποτέλεσμα του, τρέχει στη `shinyUI()` μια εντολή τύπου `YOutput("X",...)`, όπου `Y` ο τύπος μεταβλητής του `X` (table για πίνακες, plot για γραφήματα και γράφους, και άλλα).

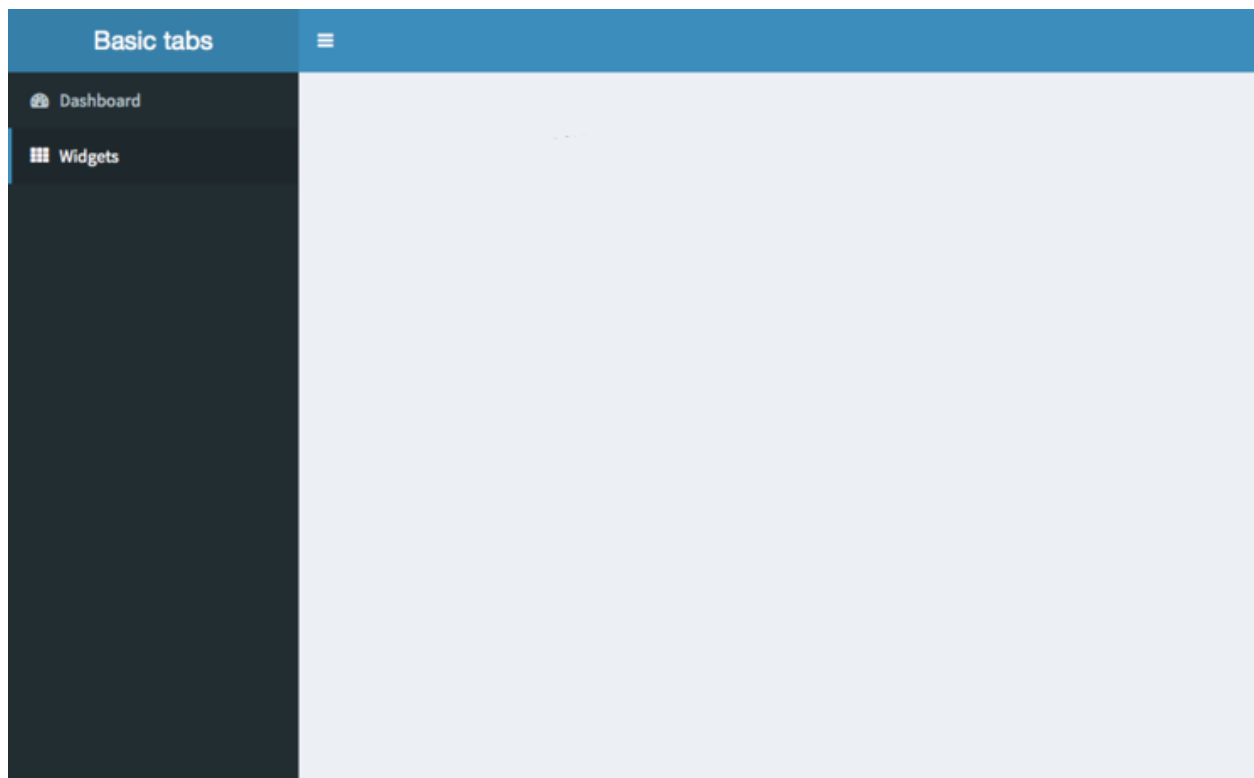
Τα συνηθέστερα widgets που μπορεί να προσθέσει κανείς στην εφαρμογή του, φαίνονται παρακάτω



Εικόνα 17: Τα βασικά widgets της Shiny.

Η παραπάνω εικόνα εξυπηρετεί και σαν παράδειγμα απλής εφαρμογής της Shiny. Τα διάφορα widgets είναι τοποθετημένα σε τέσσερις στήλες. Παρατηρούμε το άσπρο φόντο, τον τίτλο της εφαρμογής (Basic widgets), και τον τίτλο σε κάθε ένα από τα widgets ξεχωριστά. Όλα αυτά έχουν οριστεί στη shinyUI() και εφόσον οι εκχωρήσεις στα widgets αυτά δεν έχουν κάποιο αποτέλεσμα στην εφαρμογή, η shinyServer() θα μπορούσε να είναι κενή.

Ενα σκαλί παραπάνω από τη Shiny, είναι η βιβλιοθήκη Shinydashboard. Προσφέρει στο χρήστη ένα πολύ πιο φιλικό αρχικό περιβάλλον για την εφαρμογή του, καθώς και την επιλογή να προσθέσει καρτέλες, ακόμα και καρτέλες μέσα στις καρτέλες του για να κατηγοριοποιήσει με αυτόν τον τρόπο τα δεδομένα που θέλει να παρουσιάσει με έναν πιο οργανωμένο τρόπο.



Εικόνα 18: Κενή εφαρμογή με δύο καρτέλες της shinydashboard.

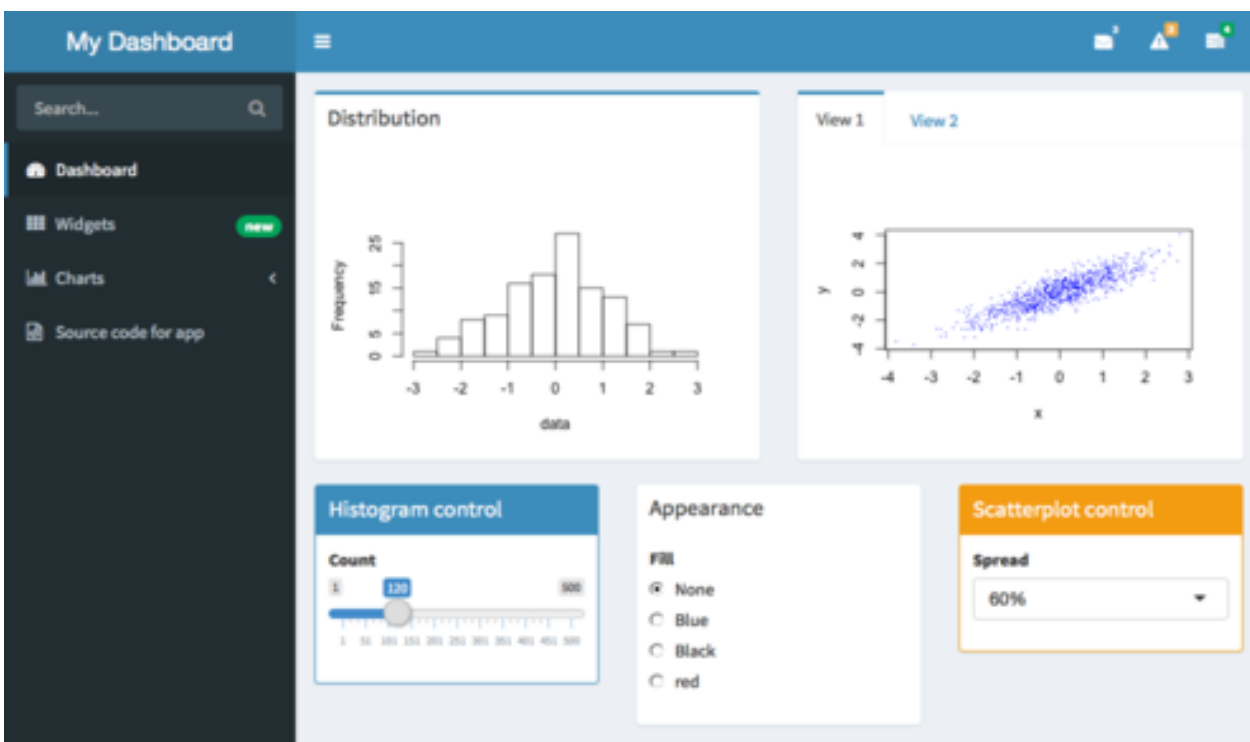
Όπως στη Shiny, έτσι και στη Shinydashboard μια εφαρμογή αποτελείται από δύο functions. Μόνο που στην προκειμένη περίπτωση η συνάρτηση για το user interface είναι η `dashboardPage()` ενώ για το server είναι μια απλή `function(input,output)`. Λογικό, μιας και το μόνο που αλλάζει σε σχέση με τη Shiny, είναι το περιβάλλον της εφαρμογής και όχι ο τρόπος με τον οποίο θα υπολογίζονται τα αποτελέσματα. Ο τίτλος της εφαρμογής στο πάνω αριστερά κομμάτι της ορίζεται στην `dashboardPage()` από τη συνάρτηση `dashboardHeader()` ενώ τα περιεχόμενα στη στήλη στα αριστερά, από τη συνάρτηση `dashboardSidebar()`. Από εκεί ο χρήστης μπορεί να πλοηγηθεί στις διάφορες καρτέλες της εφαρμογής. Τα ονόματα, εικονίδια και τα περιεχόμενα κάθε καρτέλας ορίζονται με τη συνάρτηση `sidebarMenu()` μέσα στην οποία ο χρήστης δημιουργεί μια καινούρια καρτέλα κάθε φορά που χρησιμοποιεί τη συνάρτηση `menuItem()`.

Μετά τις `dashboardHeader()` και `dashboardSidebar()`, η τρίτη, βασικότερη και τελευταία συνάρτηση της `dashboardPage()` είναι η `dashboardBody()`. Εδώ ξεχωρίζει το τι θα μπει σε

κάθε καρτέλα, αρχικά ανοίγοντας τη συνάρτηση `tabItems()` και μέσα σ'αυτή ανοίγοντας μια συνάρτηση `tabItem()` για κάθε καρτέλα. Η πρώτη παράμετρος σε κάθε `tabItem()` είναι η `tabName` και χρησιμοποιείται για να ορίσει σε ποια καρτέλα από αυτές που εκχωρήθηκαν στην `sidebarMenu()` θα αναφέρονται οι συναρτήσεις που θα ακολουθήσουν.

Τα widgets, τα plots και ό,τι άλλο προσθέσει ο χρήστης τοποθετούνται πρώτα από επάνω αριστερά προς τα κάτω και μετά προς τα δεξιά της εφαρμογής, εκτός αν οριστεί διαφορετικά. Τρεις πολύ βασικές εντολές για να το επεξεργαστεί κανείς αυτό, είναι η `row()`, η `column()` και η `box()`. Με τις `row()` και `column()` ο χρήστης μπορεί να χωρίσει σε γραμμές και στήλες την εφαρμογή του. Αν λοιπόν για παράδειγμα θέλει ένα στοιχείο να μη φαίνεται κάτω από ένα άλλο, αλλά δίπλα του, τότε το μόνο που έχει να κάνει είναι να ορίσει 2 στήλες με την `column()` και να τοποθετήσει το κάθε αντικείμενο σε μια στήλη ξεχωριστά, ή να χρησιμοποιήσει μια γραμμή με τη `row()` και να τα τοποθετήσει το ένα μετά το άλλο.. Το πλάτος της γραμμής ή στήλης ορίζεται εντός της συνάρτησης, με την παράμετρο "width". Με την εντολή `box()` προστίθεται ένα πλαίσιο στην εφαρμογή που μπορεί να περιέχει ένα ή περισσότερα αντικείμενα της, δίνοντας έτσι ένα καλύτερο αποτέλεσμα οπτικά, αλλά και τοποθετώντας αντικείμενα με παρόμοιες ιδιότητες σε ομάδες. Το πλάτος και το ύψος του κάθε πλαισίου ορίζεται από τις `width` και `height` αντίστοιχα, ενώ προσφέρεται η δυνατότητα για αλλαγή του χρώματος του, επιλογή ελαχιστοποίησης, και πολλά ακόμη.

Ενα παράδειγμα εφαρμογής με 4 καρτέλες, 2 γραμμές και 5 πλαίσια φαίνεται στην παρακάτω εικόνα. Σε κάθε πλαίσιο έχει τοποθετηθεί είτε ένα plot είτε ένα widget μαζί με τον αντίστοιχο τίτλο.



Εικόνα 19: Παράδειγμα εφαρμογής με 4 καρτέλες, 2 γραμμές και 5 boxes.

Πολλά από τα αποτελέσματα στην εργασία αυτή παρουσιάστηκαν μέσω γράφων και για το λόγο αυτό, χρησιμοποιήθηκε εκτενώς η βιβλιοθήκη `igraph` της R. Πρακτικά για να σχεδιαστεί ένας γράφος χρειάζονται δύο εντολές. Η πρώτη είναι για να τον δημιουργήσει σαν αντικείμενο `graph` της R και η δεύτερη για να τον σχεδιάσει. Η δημιουργία μπορεί να γίνει με πολλούς τρόπους, ανάλογα τον τύπο των πρωτογενών δεδομένων που έχουμε. Το συνηθέστερο είναι να έχουμε έναν πίνακα γειτνίασης με γραμμές ή στήλες όσες και οι κόμβοι και στοιχεία του πίνακα το βάρος των ακμών ανάμεσα στους κόμβους αυτούς. Τα μηδενικά στοιχεία δηλώνουν ότι δεν υπάρχει ακμή στον αντίστοιχο συνδυασμό. Η συνάρτηση που χρησιμοποιείται για το σκοπό αυτό είναι η `graph.adjacency` ή `graph_from_adjacency_matrix` και παίρνει σαν input προφανώς τον παραπάνω πίνακα, καθώς και αρκετές παραμέτρους για τον γράφο που θα προκύψει, όπως βάρος ακμών (`weighted=...`), στοιχεία της διαγωνίου (`diag=...`), κατευθυνόμενο γράφο (`directed=...`) και άλλα.

Ο χρήστης μπορεί να δημιουργήσει γράφους και με πολλούς ακόμα τρόπους μέσω της `igraph`, ακόμη κι αν δεν έχει τον πίνακα γειτνίασης. Οι υπόλοιπες εντολές για τη δημιουργία γράφων με διαφορετικά αρχικά δεδομένα είναι οι παρακάτω:

- `graph_from_adj_list`
- `graph_from_atlas`
- `graph_from_edgelist`
- `graph_from_graphdb`
- `graph_from_graphnel`
- `graph_from_incidence_matrix`
- `graph_from_isomorphism_class`
- `graph_from_lcf`
- `graph_from_literal`

Αφού δημιουργηθεί ο γράφος ο χρήστης μπορεί να τον σχεδιάσει με την εντολή `plot.igraph()`. Μέσα στη συνάρτηση ορίζονται οι ιδιότητες του σχεδίου, με τις βασικότερες να είναι χρώμα ακμών και κόμβων, πάχος ακμών και κόμβων, το layout του γράφου, οι ετικέτες που προαιρετικά υπάρχουν στους κόμβους και στις ακμές και μπορεί να δηλώνουν το όνομα, την αξία, ή το βάρος αντίστοιχα και πάρα πολλά ακόμη. Αξίζει να σημειωθεί πως το ο γράφος εμφανίζεται σαν εικόνα και κατά συνέπεια δεν είναι δυνατή η περαιτέρω επεξεργασία του μετά τη σχεδίαση του. Εάν λοιπόν ο χρήστης θέλει να μετακινήσει έναν κόμβο με το δείκτη του ποντικιού του αυτό δεν είναι εφικτό με την `plot.igraph()`. Μπορεί όμως να το πετύχει με την `tkplot()`, δαπανώντας φυσικά πολύ περισσότερο χώρο και resources του υπολογιστή του. Δυστυχώς η `tkplot()` ανοίγει το γράφο σε καινούριο παράθυρο και για το λόγο αυτό δεν είναι συμβατή με τις εφαρμογές της `shinydashboard`.

Το παρακάτω παράδειγμα σχεδιάζει το γράφο του πίνακα με τυχαία στοιχεία 0, 1 ή 2:

$$mat = \begin{pmatrix} 0 & 2 & 0 & 2 & 1 \\ 2 & 1 & 2 & 0 & 2 \\ 2 & 0 & 1 & 2 & 2 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \end{pmatrix}$$

Ο πίνακας δημιουργήθηκε με την εντολή `sample()`. Πρώτα πρέπει να δημιουργηθεί ο γράφος:

```
Graph<-graph.adjacency(mat,
                        mode= "directed",
                        weighted=TRUE,
                        diag=FALSE)
```

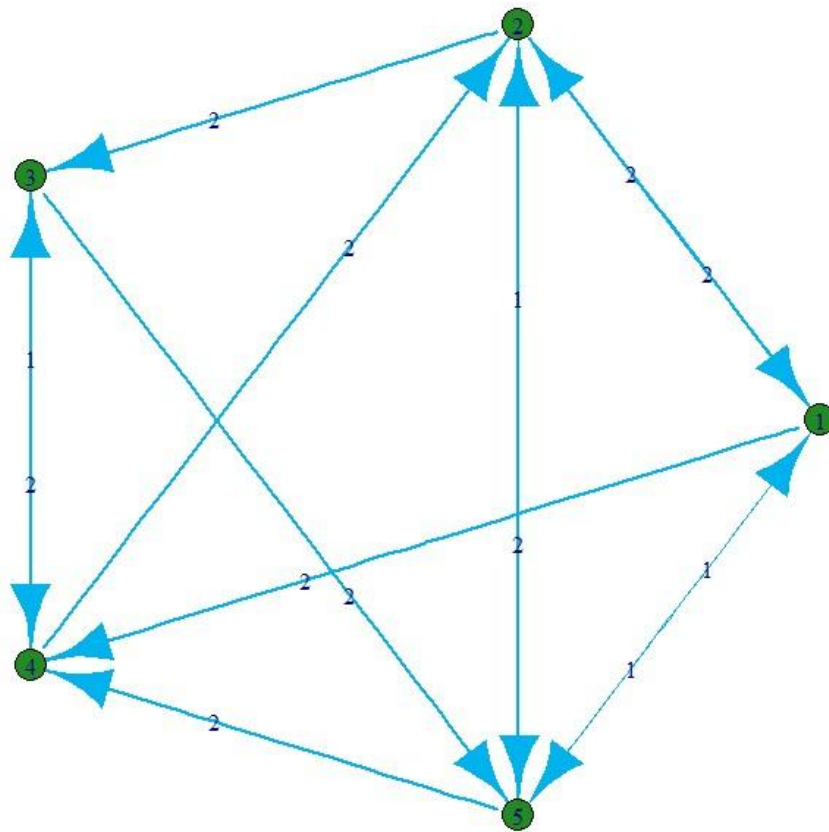
Η ιδιότητα `mode` έχει την τιμή `"directed"` και συνεπώς ο γράφος θα είναι κατευθυνόμενος. Σε αντίθετη περίπτωση θα χρησιμοποιούταν η `"undirected"`. Η λογική ιδιότητα `weight` που στην προκειμένη περίπτωση είναι `TRUE` δηλώνει πως οι τιμές του πίνακα θα χρησιμοποιηθούν σαν βάρη στις ακμές του γράφου. Στην περίπτωση `FALSE`, οι τιμές του πίνακα δε θα είχαν κανένα νόημα, με την εξαίρεση ότι τα μηδενικά στοιχεία θα εξακολουθούσαν να δείχνουν την έλλειψη σύνδεσης στους αντίστοιχους κόμβους. Τέλος, η ιδιότητα `diag`, επίσης λογική, δηλώνει αν μας ενδιαφέρουν οι συνδέσεις από οποιονδήποτε κόμβο στον εαυτό του. Πρακτικά, μηδενίζει τα στοιχεία της κύριας διαγωνίου του πίνακα. Εδώ δίνεται η τιμή `FALSE`. Ο γράφος εκχωρείται στη μεταβλητή με όνομα `"Graph"`.

```
plot.igraph(Graph,
             layout=layout.circle,
             vertex.color="forestgreen",
             vertex.size=8,
             edge.color="Deepskyblue2",
             edge.width=E(Graph)$weight,
             edge.label=E(Graph)$weight)
```

Η παραπάνω εντολή σχεδιάζει το γράφο `"Graph"`. Παρατηρούμε ότι πέρα από το όνομα του γράφου, έχουμε 6 ιδιότητες. Η `layout` αφορά το στήσιμο που θα έχουν οι κόμβοι του δικτύου στο σχεδιασμό (εδώ χρησιμοποιείται η `layout.circle`, που τους τοποθετεί σε κύκλο). Άλλα γνωστά `layouts` είναι το `layout.fruchterman.reingold`, `layout.sphere`, και `layout.kamada.kawai`. Οι δύο επόμενες, αφορούν τους κόμβους και συγκεκριμένα η `vertex.color` το χρώμα τους και η `vertex.size` το μέγεθος τους. Εδώ επιλέχθηκε το `"forestgreen"` για χρώμα και το 8 για το μέγεθος. Τέλος, οι τρεις τελευταίες αφορούν τις ακμές και συγκεκριμένα η `edge.color` το χρώμα τους, η `edge.width` το πάχος τους και η `edge.label` την ετικέτα που ενδεχομένως ο χρήστης θέλει να έχουν πάνω. Η τιμές που επιλέχθηκαν είναι για το χρώμα το `"Deepskyblue2"`, ενώ για το πάχος και την ετικέτα η `E(Graph)$weight` στην

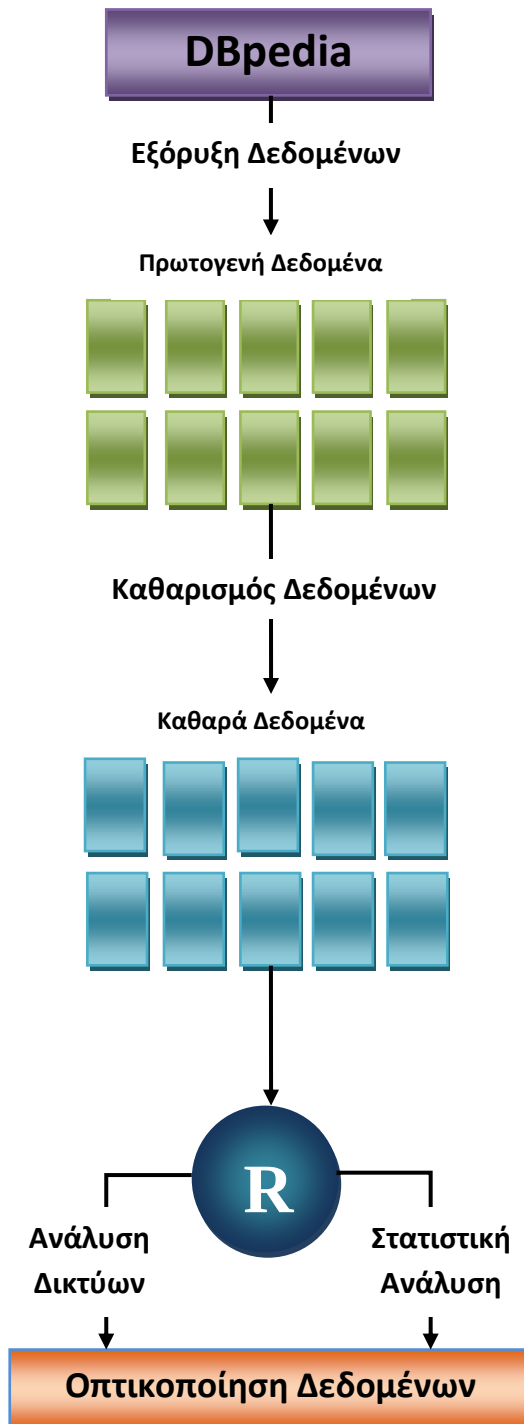
οποία είναι εκχωρημένα τα βάρη κάθε ακμής. Αρα όσο πιο μεγάλο το βάρος, τόσο πιο παχιά η ακμή, και η ετικέτα θα το δείχνει αυτό. Σημειώνεται ότι το βάρος είναι στην ουσία οι τιμές του πίνακα γειτνίασης `mat` από τον οποίο ορίστηκε ο γράφος εξ'αρχής.

Τρέχοντας την εντολή, παίρνουμε το παρακάτω δίκτυο:



Εικόνα 20: Παράδειγμα γράφου, σχεδιασμένου μέσω της `igraph`.

Η διαδικασία που ακολουθήθηκε σε όλη την εργασία περιγράφεται στο παρακάτω διάγραμμα ροής:



3. ΕΞΟΡΥΞΗ ΚΑΙ ΚΑΘΑΡΙΣΜΟΣ ΤΩΝ ΚΙΝΗΜΑΤΟΓΡΑΦΙΚΩΝ ΔΕΔΟΜΕΝΩΝ

3.1 Εξόρυξη των κινηματογραφικών δεδομένων από την DBpedia

Όπως προαναφέρθηκε, το σύνολο της πληροφορίας με το οποίο ασχολήθηκε η εργασία αυτή προήλθε από τη DBpedia. Αρχικά, έπρεπε να αποφασιστεί η κατηγορία δεδομένων και σε δεύτερο χρόνο οι λεπτομέρειες πάνω στην κατηγορία αυτή. Ο κινηματογράφος είναι μια πολύ δελεαστική πρόταση, μιας και υπάρχει τεράστιος όγκος δεδομένων από ταινίες, πολλές συσχετίσεις μεταξύ των εταιριών, των ετών παραγωγής, των ηθοποιών, της οικονομικής επιτυχίας κάθε ταινίας και πολλά ακόμη. Επιπλέον η Wikipedia είναι γνωστό πως έχει ένα άρθρο που αφορά κάθε ταινία, με τις πληροφορίες της σε διασυνδεδεμένα δεδομένα στο εκάστοτε infobox. Η ιδέα είναι αρχικά να δούμε ποιες εταιρίες έχουν τις περισσότερες ή τις λιγότερες συνεργασίες και στη συνέχεια να πάμε ένα σκαλί παραπάνω, εφαρμόζοντας στατιστικές μεθόδους στις οικονομικές απολαβές κάθε εταιρίας ανά έτος. Για το λόγο αυτό, επιλέχθηκε να αντληθούν δεδομένα σε πίνακες με στήλες:

- Τίτλος ταινίας
- Εταιρία διανομής
- Κόστος παραγωγής
- Εσοδα
- Έτος κυκλοφορίας

Αρχικά έπρεπε να εντοπίσουμε τις ιδιότητες που συνδέουν τα δεδομένα των παραπάνω στηλών. Ολόκληρη η οντολογία της DBpedia βρίσκεται στο σύνδεσμο <http://mappings.dbpedia.org/server/ontology/classes/>. Από εκεί βλέπουμε ότι οι ταινίες είναι εκχωρημένες στην κατηγορία "Film". Αυτό που τώρα χρειάζεται είναι να ελέγξουμε μια οποιαδήποτε σελίδα ταινίας στην DBpedia για να δούμε με ποιες ιδιότητες είναι συνδεδεμένες

οι στήλες που θέλουμε. Πλοηγούμαστε λοιπόν με τον τρόπο που προαναφέρθηκε και δημιουργούμε τον παρακάτω πίνακα.

Πίνακας 1: Πίνακας κατηγοριών δεδομένων με τις αντίστοιχες ιδιότητες τους.

Κατηγορία	Ιδιότητα
Τίτλος ταινίας	a (rdf:type) dbpedia0:Film
Εταιρία διανομής	dbp:distributor
Κόστος παραγωγής	dbp:budget
Εσοδα	dbp:gross
Ετος κυκλοφορίας	dct:subject

Αφού πλοηγηθεί κανείς σε μερικές σελίδες ταινιών της DBpedia παρατηρεί ότι η ημερομηνία κυκλοφορίας δεν έχει καταχωρηθεί στην πλειοψηφία αυτών. Σε αυτές που έχει καταχωρηθεί, χρησιμοποιείται η βολική ιδιότητα `dbo:releaseDate`. Ακόμα και να υπήρχε σε όλες τις ταινίες, ο τρόπος σύνταξης της ημερομηνίας έχει πάρα πολλές διαφορές από σελίδα σε σελίδα. Μερικά παραδείγματα είναι: HH-MM-EEEE, EEEE-MM-HH, EE-MM-HH, EE/MM/HH, όπου E, M και H τα ψηφία του έτους, του μήνα και της ημέρας αντίστοιχα. Το πλήθος των συνδυασμών συμβόλων και διαδοχής των ψηφίων είναι τεράστιο, κάτι που θα καθιστούσε τον καθαρισμό των δεδομένων εξαιρετικά δύσκολο και χρονοβόρο. Για το χρονολογικό διαχωρισμό των δεδομένων λοιπόν, χρησιμοποιήθηκε η `dct:subject`, η οποία υπάρχει για κάθε ταινία και περιέχει όλες της κατηγορίες στις οποίες ανήκει η ταινία αυτή. Για παράδειγμα για την ταινία Avatar, η `dct:subject` περιλαμβάνει τις κατηγορίες:

- `dbc:Films_shot_in_New_Zealand`
- `dbc:2000s_science_fiction_films`
- `dbc:Rotoscoped_films`
- `dbc:20th_Century_Fox_films`

- dbc:American_epic_films
- dbc:American_films
- dbc:American_science_fiction_action_films
- dbc:Artificial_uterus_in_fiction
- dbc:Best_Film_Empire_Award_winners
- dbc:English-language_films
- dbc:Environmental_films
- dbc:Epic_films
- dbc:Fictional-language_films
- dbc:Film_scores_by_James_Horner
- dbc:Films_about_rebellions
- dbc:Films_about_telepresence
- dbc:Films_directed_by_James_Cameron
- dbc:Films_shot_in_California
- dbc:Films_that_won_the_Best_Visual_Effects_Academy_Award
- dbc:Films_whose_art_director_won_the_Best_Art_Direction_Academy_Award
- dbc:Films_whose_cinematographer_won_the_Best_Cinematography_Academy_Award
- dbc:Performance_capture_in_film
- dbc:Science_fiction_war_films
- dbc:Screenplays_by_James_Cameron
- dbc:Space_adventure_films
- dbc:2009_films
- dbc:Avatar_(2009_film)
- dbc:Films_about_technology
- dbc:Dune_Entertainment_films
- dbc:2000s_action_films
- dbc:Billion-dollar_grossing_films
- dbc:Films_set_in_the_22nd_century
- dbc:Films_shot_in_Hawaii
- dbc:Films_using_computer-generated_imagery
- dbc:IMAX_films
- dbc:Lightstorm_Entertainment_films
- dbc:Planetary_romances
- dbc:Rebellions_in_fiction
- dbc:2000s_3D_films
- dbc:Military_science_fiction_films
- dbc:Films_about_cloning

Κάθε ταινία που ικανοποιεί μια από τις παραπάνω συνθήκες προστίθεται στην αντίστοιχη κατηγορία. Έτσι, κάθε ταινία έχει τη δική της λίστα με ιδιότητες `dc:subject`. Από όλες τις κατηγορίες αυτές εμάς μας ενδιαφέρει μόνο αυτή στην οποία αναφέρεται το έτος κυκλοφορίας και συγκεκριμένα η `dbc:2009_films`. Η ετικέτα αυτή υπάρχει σε κάθε σελίδα ταινίας και το μόνο που διαφέρει είναι προφανώς το `dbc:20XX_films`, ανάλογα με το έτος. Πάνω σε αυτή τη λεπτομέρεια θα σταθούμε για να την ξεχωρίσουμε από τις υπόλοιπες με την εντολή `FILTER`

της SPARQL. Συνδυάζοντας όλα αυτά, είμαστε έτοιμοι για να γράψουμε το SPARQL ερώτημα.

```
PREFIX db0: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?movie ?dist ?budg ?boxoffice ?date
WHERE {
  ?movie a db0:Film ;
        dbp:distributor ?dist ;
        dbp:gross ?boxoffice ;
        dbp:budget ?budg ;
        dct:subject ?date

  FILTER (regex(?dist, 'Disney')) .
  FILTER (regex(?date, '20')) .
}

ORDER BY ?date
```

Στα προθέματα προσθέτουμε την db0 για να συντομεύσουμε την οντολογία της DBpedia, που θα χρησιμοποιηθεί για να προσδιορίσουμε την έννοια Film. Όλα τα υπόλοιπα προθέματα είναι ήδη ορισμένα από τη σελίδα του endpoint και τρέχουν σε κάθε ερώτημα.

Με την εντολή SELECT ζητάμε αποτελέσματα με πέντε μεταβλητές, τις οποίες ονομάζουμε ?movie, ?dist, ?budg, ?boxoffice και ?date. Η λογική μετά έχει ως εξής:

Ζητούμε μια έννοια Film της οποίας την ετικέτα εκχωρούμε στη μεταβλητή ?movie;

Της οποίας την ετικέτα της ιδιότητας dbp:distributor εκχωρούμε στη μεταβλητή ?dist;

Της οποίας την ετικέτα της ιδιότητας dbp:gross εκχωρούμε στη μεταβλητή ?boxoffice;

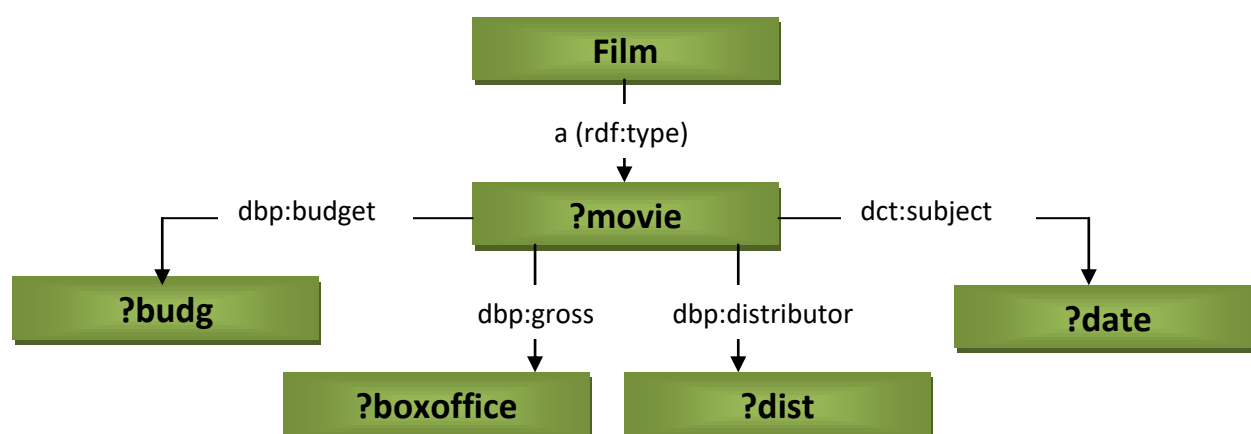
Της οποίας την ετικέτα της ιδιότητας dbp:budget εκχωρούμε στη μεταβλητή ?budg;

Της οποίας τις κατηγορίες της ιδιότητας dct:subject εκχωρούμε στη μεταβλητή ?date

Δυστυχώς το endpoint της DBpedia μπορεί να επιστρέψει μέχρι 10000 αποτελέσματα, γεγονός που κατέστησε απαραίτητο να τρέξουμε το ίδιο ερώτημα ξεχωριστά για κάθε εταιρία διανομής. Αυτό επιτεύχθηκε με το πρώτο φίλτρο, που στο συγκεκριμένο παράδειγμα, απαιτεί

η ετικέτα που εκχωρήθηκε στη μεταβλητή `?dist`, να περιέχει τη λέξη Disney. Τέτοιου είδους φίλτρα, δηλαδή για εύρεση συγκεκριμένων χαρακτήρων στις εγγραφές απαιτούν και τη συνάρτηση `regex()`, όπως φαίνεται στο παράδειγμα. Το τελευταίο φίλτρο ξεχωρίζει από όλες τις κατηγορίες της `dct:subject`, αυτές που περιέχουν το "20", για να ξεχωρίσουν έτσι οι "20XX_films" που αναφέρθηκαν παραπάνω, όποιο και να είναι το έτος κυκλοφορίας.

Τέλος, τα αποτελέσματα ταξινομούνται με βάση τη στήλη `?date`.



Οι εταιρίες διανομής και οι λέξεις που χρησιμοποιήθηκαν σαν φίλτρα στα ερωτήματα φαίνονται αλφαβητικά στον παρακάτω πίνακα.

Πίνακας 2: Οι εταιρίες διανομής και τα αντίστοιχα φίλτρα που χρησιμοποιήθηκαν στα ερωτήματα.

Εταιρία διανομής	Φίλτρο
20th Century Fox	"Fox"
Columbia Pictures	"Columbia"
DreamWorks	"Dreamworks"
Focus Features	"Focus_Features"
Fox Searchlight	"Fox"
Lionsgate	"Lionsgate"

Metro Goldwyn Mayer	"Goldwyn_Mayer"
Miramax	"Miramax"
New Line Cinema	"New_Line_Cinema"
Paramount Pictures	"Paramount"
Relativity Media	"Relativity"
Sony Pictures	"Sony"
Summit Entertainment	"Summit"
Universal Pictures	"Universal"
Walt Disney Pictures	"Disney"
Warner Bros.	"Warner"
The Weinstein Company	"Weinstein"

Αρχικά επιλέχθηκε η επιλογή Browse από το endpoint της DBpedia για να πάρουμε μια πρώτη ιδέα για τη μορφή των δεδομένων. Με την πρώτη ματιά βλέπουμε αρκετές ασυνέχειες:

- Διαφορετικές διατυπώσεις (Warner_Bros, Warner_Bros_Pictures)
- Αριθμητικές ασυνέχειες (10million, 10000000, 10E6)
- Διπλοεγγραφές
- Ορθογραφικά λάθη

Παρατηρήθηκε καλή κατάσταση στη στήλη monie, με τους τίτλους των ταινιών να εμφανίζονται με τα URI τους που οδηγούν στην αντίστοιχη σελίδα της DBpedia με τις ιδιότητες τους. Στη στήλη dist οι εταιρίες εμφανίζονται με αρκετές διαφορετικές διατυπώσεις, πολλές φορές και για την ίδια ταινία, γεγονός που προκαλεί αρκετές διπλοεγγραφές. Γενικά όμως όλες οι περιπτώσεις είναι περιορισμένες σε πλήθος, οπότε ο καθαρισμός τους θα είναι

σχετικά εύκολος. Οι δύο στήλες `budget` και `boxoffice` με τα αριθμητικά δεδομένα είναι αυτές που παρουσιάζουν τις μεγαλύτερες διαφορές μιας και πολλές εγγραφές αφορούν διαφορετικά νομίσματα, είναι τυπωμένες με όλα τα ψηφία τους ή με λιγότερα από αυτά (10.000.000 ή 10 million), περιέχουν σύμβολα νομισμάτων, ακόμα και επεξηγηματικό κείμενο με λεπτομέρειες για τα έσοδα και τα έξοδα. Για μια μελέτη ακριβείας αυτές οι στήλες είναι οι πιο σημαντικές, επομένως ο καθαρισμός τους θα είναι πιο δύσκολος και χρονοβόρος. Τέλος η στήλη `date` περιέχει πολλές περιττές στήλες μιας και το φίλτρο "20" που χρησιμοποιήθηκε δεν είναι αρκετά αυστηρό και οι δυο αυτοί χαρακτήρες θα μπορούσαν να εμφανίζονται σε πολλές κατηγορίες της `dict:subject`. Ετσι, πέρα από τις επιθυμητές `20XX_films` εγγραφές, περνάνε εγγραφές όπως `20XX_animated_films`, `20XX_science_fiction_films` και άλλες οι οποίες κυρίως αφορούν την ίδια ταινία και ως αποτέλεσμα η ταινία αυτή να εμφανίζεται πολλαπλές φορές στον κάθε πίνακα.

Μια εικόνα από ένα δείγμα του ερωτήματος για τη Walt Disney Pictures φαίνεται παρακάτω.

movie	dist	budget	boxoffice	date
20,000_Leagues_Under_the_Sea_(1954_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"5000000.0" ↗ dbpedia:datatype/usDollar	"2.82E7" ↗ dbpedia:datatype/usDollar	Category:20,000_Leagues_Under_the_Sea_films ↗
High_Fidelity_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"3.0E7" ↗ dbpedia:datatype/usDollar	"4.712629E7" ↗ dbpedia:datatype/usDollar	Category:2000_compilation_albums ↗
Dinosaur_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.27E8" ↗ dbpedia:datatype/usDollar	"3.49E8" ↗ dbpedia:datatype/usDollar	Category:2000_computer-animated_films ↗
Gone_in_60_Seconds_(2000_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"9.0E7" ↗ dbpedia:datatype/usDollar	"2.3720229E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Duets_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.0E7" ↗ dbpedia:datatype/usDollar	"4.739023.0" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Keeping_the_Faith ↗	Walt_Disney_Studios_Motion_Pictures ↗	"2.9E7" ↗ dbpedia:datatype/usDollar	"6.9945183E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
High_Fidelity_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"3.0E7" ↗ dbpedia:datatype/usDollar	"4.712629E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Shanghai_Noon ↗	Walt_Disney_Studios_Motion_Pictures ↗	"6.5E7" ↗ dbpedia:datatype/usDollar	"9.9274467E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Mission_to_Mars ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.0E8" ↗ dbpedia:datatype/usDollar	"1.10983407E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Coyote_Ugly_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"4.5E7" ↗ dbpedia:datatype/usDollar	"1.13916474E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
The_Crew_(2000_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"2.3E7" ↗ dbpedia:datatype/usDollar	"1.3019253E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Dinosaur_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.27E8" ↗ dbpedia:datatype/usDollar	"3.49E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Disney's_The_Kids ↗	Walt_Disney_Studios_Motion_Pictures ↗	"6.5E7" ↗ dbpedia:datatype/usDollar	"1.1031758E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Unbreakable_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"7.5E7" ↗ dbpedia:datatype/usDollar	"2.48118121E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Down_to_You ↗	The_Walt_Disney_Studios_(division) ↗	"3.5E7" ↗ dbpedia:datatype/usDollar	"2.4419914E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Love's_Labour's_Lost_(2000_film) ↗	The_Walt_Disney_Studios_(division) ↗	"1.3E7" ↗ dbpedia:datatype/usDollar	"299792.0" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Remember_the_Titans ↗	Walt_Disney_Studios_Motion_Pictures ↗	"3.0E7" ↗ dbpedia:datatype/usDollar	"1.36706083E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
The_Emperor's_New_Groove ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.0E8" ↗ dbpedia:datatype/usDollar	"1.693E8" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
The_Tigger_Movie ↗	Walt_Disney_Pictures ↗	"3.0E7" ↗ dbpedia:datatype/usDollar	"9.61598E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
The_Tigger_Movie ↗	Walt_Disney_Studios_Motion_Pictures ↗	"3.0E7" ↗ dbpedia:datatype/usDollar	"9.61598E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Gun_Shy_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.4E7" ↗ dbpedia:datatype/usDollar	"3221280.0" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
Nine_Queens ↗	Walt_Disney_Studios_Motion_Pictures ↗	"\$1,300,000" ↗ en	"1.241388E7" ↗ dbpedia:datatype/usDollar	Category:2000_films ↗
High_Fidelity_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"3.0E7" ↗ dbpedia:datatype/usDollar	"4.712629E7" ↗ dbpedia:datatype/usDollar	Category:2000_soundtracks ↗
Bolt_(2008_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.5E8" ↗ dbpedia:datatype/usDollar	"3.1E8" ↗ dbpedia:datatype/usDollar	Category:2000s_3D_films ↗
Up_(2009_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.7E8" ↗ dbpedia:datatype/usDollar	"7.313E8" ↗ dbpedia:datatype/usDollar	Category:2000s_3D_films ↗
Hannah_Montana_and_Miley_Cyrus_Best_of_Both_Worlds_Concert ↗	Walt_Disney_Studios_Motion_Pictures ↗	"7000000.0" ↗ dbpedia:datatype/usDollar	"7.06E7" ↗ dbpedia:datatype/usDollar	Category:2000s_3D_films ↗
G-Force_(film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.5E8" ↗ dbpedia:datatype/usDollar	"2.92E8" ↗ dbpedia:datatype/usDollar	Category:2000s_3D_films ↗
Chicken_Little_(2005_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.5E8" ↗ dbpedia:datatype/usDollar	"3.144E8" ↗ dbpedia:datatype/usDollar	Category:2000s_3D_films ↗
A_Christmas_Carol_(2009_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"\$175-200 million" ↗ en	"3.253E8" ↗ dbpedia:datatype/usDollar	Category:2000s_3D_films ↗
Atlantis_The_Lost_Empire ↗	Walt_Disney_Studios_Motion_Pictures ↗	"\$90-120 million" ↗ enSince the estimated budget has a range, the officially reported budget of \$100 million cited by The New York Times from Disney executives is used within this article's prose for clarity" ↗ en	"1.861E8" ↗ dbpedia:datatype/usDollar	Category:2000s_American_animated_films ↗
Fantasia_2000 ↗	Walt_Disney_Studios_Motion_Pictures ↗	"\$0.0E7" ↗ dbpedia:datatype/usDollar	"\$0.0E7" ↗ dbpedia:datatype/usDollar	Category:2000s_American_animated_films ↗
Bolt_(2008_film) ↗	Walt_Disney_Studios_Motion_Pictures ↗	"1.5E8" ↗ dbpedia:datatype/usDollar	"3.1E8" ↗ dbpedia:datatype/usDollar	Category:2000s_American_animated_films ↗

Εικόνα 21: Κομμάτι αποτελεσμάτων στο *endpoint* της *DBpedia*.

Το ερώτημα εκτελέστηκε για κάθε εταιρία διανομής ξεχωριστά, και οι 17 πίνακες αποθηκεύτηκαν σε μορφή json.

3.2 Καθαρισμός των κινηματογραφικών δεδομένων

Όπως φαίνεται και από την προεπισκόπηση των πινάκων των δεδομένων, οι εγγραφές έχουν μεγάλη ποικιλία διατυπώσεων και ανομοιομορφία. Η ιδέα του καθαρισμού είναι να εντοπιστούν όλες οι διαφορετικές εγγραφές και να επεξεργαστούν ώστε να είναι ίδια διατυπωμένες με τις υπόλοιπες. Με αυτόν τον τρόπο θα είναι εφικτή η σύγκριση τους και η εξαγωγή αξιόλογων αποτελεσμάτων. Φυσικά ο όσο το δυνατόν μαζικός καθαρισμός αυτών των ασυνεχειών είναι πρωταρχικός στόχος για οικονομία και κόπου και χρόνου. Για το σκοπό αυτό υπάρχουν πολλά λογισμικά και εφαρμογές. Στην εργασία αυτή επιλέχθηκε το Open Refine.

Στόχος είναι να καταλήξουμε με υπολογιστικά φύλλα που θα έχουν στήλες με ομοιόμορφες εγγραφές που αφορούν:

- Τίτλο ταινίας
- Εταιρία διανομής
- Έξοδα
- Έσοδα
- Έτος κυκλοφορίας
- Διαφορά εσόδων - εξόδων

Αυτό επιτεύχθηκε με τη χρήση κυρίως του Open Refine, και σε ένα μικρό ποσοστό του SPSS.

Τα βήματα που ακολουθήθηκαν από το ανέβασμα των πρωτογενών δεδομένων μέχρι την εξαγωγή της τελικής "καθαρής" βάσης δεδομένων θα περιγραφούν αναλυτικά χρησιμοποιώντας για παράδειγμα τον καθαρισμό του αρχείου δεδομένων της Walt Disney.

Ο καθαρισμός σε κάθε ένα από τα 16 αρχεία αρχικά μέσω του Open Refine έγινε συνολικά σε 3 στάδια:

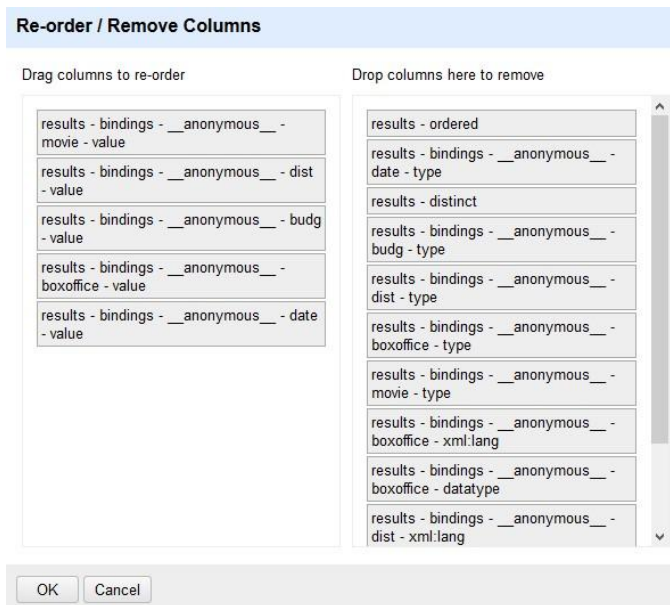
1. Διαγραφή και ταξινόμηση στηλών.
2. Καθαρισμός εγγραφών κειμένου.
3. Καθαρισμός αριθμητικών εγγραφών.

Αφού περάσουμε ως JSON το κάθε αρχείο στο Open Refine παίρνουμε ένα αποτέλεσμα της παρακάτω μορφής:

▼ All	▼ results - order	▼ results - distinct	▼ results - binding	▼ results - bindings - __anonymous__ - dat	▼ results - binding	▼ results - binding	▼ results - binding	▼ result
1.	true	false	uri	http://dbpedia.org/resource/Category:20_000_Leagues_Under_the_Sea_films	typed-literal	5000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_compilation_albums	typed-literal	30000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_computer-animated_films	typed-literal	12750000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	90000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	16000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	29000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	30000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	55000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	100000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	45000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	23000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	12750000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	65000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	75000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	35000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	13000000	http://dbpedia.org/datatype/usDollar	
			uri	http://dbpedia.org/resource/Category:2000_films	typed-literal	30000000	http://dbpedia.org/datatype/usDollar	

Εικόνα 22: Αρχική μορφή των δεδομένων.

Στην παραπάνω εικόνα, παρατηρούμε ότι εμφανίζονται πολύ περισσότερες στήλες από τις πέντε που εμείς αρχικά θέλουμε. Αυτό οφείλεται στην αποκωδικοποίηση του json αρχείου από το Open Refine, σε συνδυασμό με τα περιττά σύμβολα που υπήρχαν στην αρχή ή το τέλος πολλών εγγραφών στα αρχικά δεδομένα από τη DBpedia. Όλες αυτές οι περιττές στήλες πρέπει να διαγραφούν μιας και δεν περιέχουν ουσιώδη πληροφορία. Επιπλέον οι στήλες που θέλουμε να παραμείνουν πρέπει να μουν στη σωστή σειρά. Αυτό πετυχαίνεται με την Re-order/ Remove Columns και αποτελεί το πρώτο στάδιο καθαρισμού των δεδομένων μας.



Εικόνα 23: Διαγραφή και ταξινόμηση των στηλών.

Μετά από τη διαδικασία αυτή το υπολογιστικό φύλλο που προκύπτει είναι σαφώς πιο κοντά στο επιθυμητό.

▼ results - bindings - __anonymous__ - movie - value	▼ results - bindings - __anonymous__ - dist - value	▼ results - bindings - __anonymous__ - budg - value	▼ results - bindings - __anonymous__ - boxoffice - value	▼ results - bindings - __anonymous__ - date - value
http://dbpedia.org/resource/20,000_Leagues_Under_the_Sea_(1954_film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	5000000	28200000	http://dbpedia.org/resource/Category:20,000_Leagues_Under_the_Sea
http://dbpedia.org/resource/High_Fidelity_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	30000000	47126295	http://dbpedia.org/resource/Category:2000_compilation_albums
http://dbpedia.org/resource/Dinosaur_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	127500000	349800000	http://dbpedia.org/resource/Category:2000_computer-animated_films
http://dbpedia.org/resource/Gone_in_60_Seconds_(2000_film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	90000000	237202299	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Duets_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	16000000	4739023	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Keeping_the_Faith	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	29000000	59945183	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/High_Fidelity_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	30000000	47126295	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Shanghai_Noon	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	55000000	99274467	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Mission_to_Mars	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	100000000	110983407	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Coyote_Ugly_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	45000000	113916474	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/The_Crew_(2000_film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	23000000	13019253	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Dinosaur_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	127500000	349800000	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Disney's_The_Kid	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	65000000	110317580	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Unbreakable_(film)	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	75000000	248118121	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Down_to_You	http://dbpedia.org/resource/The_Walt_Disney_Studios_(division)	35000000	24419914	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Love's_Labour's_Lost_(2000_film)	http://dbpedia.org/resource/The_Walt_Disney_Studios_(division)	13000000	299792	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/Remember_the_Titans	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	30000000	13670683	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/The_Emperor's_New_Groove	http://dbpedia.org/resource/Walt_Disney_Studios_Motion_Pictures	100000000	169300000	http://dbpedia.org/resource/Category:2000_film_soundtracks
http://dbpedia.org/resource/The_Tigger_Movie	http://dbpedia.org/resource/Walt_Disney_Pictures	30000000	96159800	http://dbpedia.org/resource/Category:2000_film_soundtracks

Εικόνα 24: Πρώτο στάδιο καθαρισμού.

Η πρώτη στήλη θα παραμείνει ως έχει μέχρι το τέλος του καθαρισμού. Στα δύο επόμενα στάδια θα γίνει εκτενής χρήση του text facet για τον εντοπισμό μικροδιαφορών στις διατυπώσεις και την εύκολη διόρθωση αυτών.

Στο 2ο στάδιο καθαρισμού θα επεξεργαστούμε τις στήλες dist και year ώστε να γίνουν ομοιόμορφες.

Αρχικά στη στήλη dist το text facet στο συγκεκριμένο παράδειγμα βγάζει 9 διαφορετικές διατυπώσεις για την εταιρία διανομής. Όλες όμως εκφράζουν την ίδια εταιρία, εδώ τη Walt Disney. Επιλέγουμε μία προς μία τις διατυπώσεις αυτές και τις διορθώνουμε με αυτές τις δυο λέξεις. Το Open Refine διορθώνει αυτόματα κάθε εγγραφή κάθε κατηγορίας και έτσι, πολύ γρήγορα η στήλη dist καθαρίζει και γίνεται ομοιόμορφη.

Στη στήλη year θέλουμε αυστηρά να αναγράφονται μόνο τα τέσσερα ψηφία που περιγράφουν το έτος κυκλοφορίας. Από τις κατηγορίες της dct:subject στην DBpedia έχουμε αντλήσει όλες όσες περιέχουν τον αριθμό 20 σε οποιοδήποτε σημείο στην ετικέτα τους, με σκοπό να απομονώσουμε αυτές που έχουν τη μορφή 20XX_films. Η απομόνωση αυτή γίνεται σε αυτό το στάδιο, πάλι με την text facet. Συγκεκριμένα εντοπίζουμε μέσα στα φίλτρα τις ετικέτες http://dbpedia.org/resource/Category:20XX_films και τις αντικαθιστούμε μια προς μια με τον αντίστοιχο αριθμό 20XX. Όλες οι εγγραφές που δεν είναι της μορφής αυτής δε μας ενδιαφέρουν και στην συντριπτική πλειοψηφία τους αποτελούν διπλοεγγραφές μιας ταινίας για την οποία ήδη έχουμε καθαρίσει τη στήλη date. Για να ξεχωρίσουν, εφαρμόζουμε στη στήλη date την επιλογή sort. Έτσι όλες οι εγγραφές που θέλουμε να κρατήσουμε μεταφέρονται στις πάνω γραμμές και μάλιστα με χρονολογική σειρά, ενώ οι περιττές εγγραφές που θέλουμε να διαγράψουμε εμφανίζονται στις πιο κάτω γραμμές και από εκεί αφαιρούνται εύκολα.

Στο στάδιο αυτό ένα κομμάτι του πίνακα ταξινομημένου φαίνεται παρακάτω:

All	results - bindings - __anonymous__ - movie - value	results - bindin	results - bindin	results - bindin	results - bindin
240.	http://dbpedia.org/resource/Bubble_Boy_(film)	Walt Disney	13000000	5007898	2001
241.	http://dbpedia.org/resource/Monsters,_Inc.	Walt Disney	115000000	562800000	2001
242.	http://dbpedia.org/resource/Out_Cold_(2001_film)	Walt Disney	24000000	13906394	2001
243.	http://dbpedia.org/resource/Recess:_School's_Out	Walt Disney	23000000	44400000	2001
244.	http://dbpedia.org/resource/The_Other_Side_of_Heaven	Walt Disney	7000000	4720371	2001
245.	http://dbpedia.org/resource/Behind_the_Sun_(film)	Walt Disney	4000000	RS2.063.956	2001
246.	http://dbpedia.org/resource/The_Princess_Diaries_(film)	Walt Disney	26000000	165335153	2001
250.	http://dbpedia.org/resource/Signs_(film)	Walt Disney	72000000	408247917	2002
251.	http://dbpedia.org/resource/Sorority_Boys	Walt Disney	12000000	12517488	2002
252.	http://dbpedia.org/resource/Moonlight_Mile_(film)	Walt Disney	21000000	10011050	2002
253.	http://dbpedia.org/resource/Sweet_Home_Alabama_(film)	Walt Disney	38000000	180622424	2002
254.	http://dbpedia.org/resource/The_Hot_Chick	Walt Disney	34000000	54639553	2002
255.	http://dbpedia.org/resource/Snow_Dogs	Walt Disney	33000000	115035090	2002
256.	http://dbpedia.org/resource/Bad_Company_(2002_film)	Walt Disney	70000000	65977295	2002
257.	http://dbpedia.org/resource/Abandon_(film)	Walt Disney	25000000	12302219	2002
258.	http://dbpedia.org/resource/Tuck_Everlasting_(2002_film)	Walt Disney	15000000	19344615	2002
259.	http://dbpedia.org/resource/The_Count_of_Monte_Cristo_(2002_film)	Walt Disney	35000000	75395048	2002
260.	http://dbpedia.org/resource/Return_to_Never_Land	Walt Disney	20000000	109862682	2002
261.	http://dbpedia.org/resource/The_Country_Bears	Walt Disney	35000000	18012097	2002
262.	http://dbpedia.org/resource/The_Rookie_(2002_film)	Walt Disney	22000000	80000000	2002
263.	http://dbpedia.org/resource/City_of_God_(2002_film)	Walt Disney	33000000	30641770	2002
264.	http://dbpedia.org/resource/Lilo_&_Stitch	Walt Disney	80000000	273100000	2002
265.	http://dbpedia.org/resource/Reign_of_Fire_(film)	Walt Disney	USD\$60,000,000	USD\$82,150,183	2002
266.	http://dbpedia.org/resource/The_Santa_Clause_2	Walt Disney	65000000	172855065	2002
267.	http://dbpedia.org/resource/Treasure_Planet	Walt Disney	140000000	109600000	2002

Εικόνα 25: Δεύτερο στάδιο καθαρισμού.

Απομένουν οι δυο στήλες που αφορούν το budget και το box office κάθε ταινίας. Η διαδικασία που ακολουθήθηκε είναι η ίδια και για τις δυο, μιας και τα προβλήματα ανομοιομορφίας που εμφανίστηκαν ήταν τα ίδια. Πολύ βοηθητική είναι και πάλι η text facet μιας και διάσπαρτες υπάρχουν αρκετές εγγραφές που χρειάζονται καθαρισμό, χωρίς κάποια συγκεκριμένη λογική για το πως εμφανίζονται. Οι αριθμητικές τιμές με πράσινο χρώμα είναι στην πλειοψηφία τους "καθαρές". Αυτός μάλιστα είναι και ένας τρόπος για να ξεχωρίσουν οπτικά οι εγγραφές που απαιτούν επεξεργασία. Όποια κελιά δεν έχουν πράσινους χαρακτήρες, χρειάζονται επεξεργασία.

Αρχικά πολύ συνηθισμένο είναι να υπάρχει δίπλα από τον αριθμό το σύμβολο του νομίσματος το οποίο εκφράζει. Μάλιστα σε αρκετές περιπτώσεις το νόμισμα αυτό δεν είναι το αμερικάνικο δολλάριο, γεγονός που δυσκολεύει περισσότερο την κατάσταση. Με το text facet είναι εύκολο να εντοπιστούν αυτές οι εγγραφές. Στην περίπτωση που το σύμβολο είναι USD ή

\$ τότε απλά αφαιρείται και ο αριθμός αφήνεται ως έχει. Σε άλλη περίπτωση, οι τιμές μετατρέπονται μία προς μία μέσω της google στην αντίστοιχη ισότιμη τους σε αμερικάνικα δολλάρια, με τις εν έτει 2015 ισχύουσες ισοτιμίες. Λογικό είναι να υπάρχει μια απόκλιση μιας και οι ισοτιμίες αλλάζουν από λεπτό σε λεπτό, αλλά μπροστά στον όγκο των δεδομένων, η απόκλιση αυτή θεωρείται αμελητέα.

Οι υπόλοιπες μετατροπές δεν απαιτούν τόσο δύσκολη εργασία, όσο χρονοβόρα. Πρέπει να εντοπιστούν και να διορθωθούν μία προς μία διατυπώσεις που περιέγραφαν τη χρηματική ποσότητα με λόγια και όχι με αριθμούς (10 million αντί για 10,000,000) καθώς και διατυπώσεις που ήθελαν να δώσουν επιπλέον πληροφορίες για την προέλευση ή τη διαδικασία των εσόδων και εξόδων(π.χ. Χ δολλάρια από Ευρώπη, Υ δολλάρια από Αμερική). Στη σελίδα της Wikipedia τέτοιες λεπτομέρειες δίνουν μια καινούρια πληροφορία στο χρήστη, οπότε έχουν θετικό αποτέλεσμα. Στην παρούσα εργασία όμως και φυσικά σε αντίστοιχες δουλειές πάνω στα δεδομένα αυτά, αποτέλεσαν εμπόδιο.

Τέλος για μεγαλύτερη σιγουριά εφαρμόστηκε η text facet και για τη στήλη με τους τίτλους ταινιών. Ελέγχθηκε κάθε ταινία ξεχωριστά ώστε όλα τα στοιχεία που την αφορούν να μην έχουν λάθος διατυπώσεις. Η συντριπτική πλειοψηφία ταινιών ήταν ήδη όπως θα έπρεπε, δεν έλειπαν όμως και περιπτώσεις που η επέμβαση αυτή δεν ήταν μάταιη, καθώς τελειοποιήθηκαν και τα λιγοστά προβλήματα που απέμεναν με πιο συνηθισμένο αυτό της διπλοεγγραφής.

Όπως γίνεται κατανοητό, το τρίτο και τελευταίο στάδιο καθαρισμού είναι και το πιο απαιτητικό, μιας και οι αριθμητικές εγγραφές έχουν μεγαλύτερη πιθανότητα να εμφανίσουν ατέλειες και ανομοιομορφίες στις διατυπώσεις τους.

▼ results - bindings - __anonymous__ - movie - value	▼ results - bindin	▼ results - bindin	▼ results - bindin	▼ results - bindi
http://dbpedia.org/resource/Gone_in_60_Seconds_(2000_film)	Walt Disney	90000000	237202299	2000
http://dbpedia.org/resource/Duets_(film)	Walt Disney	16000000	4739023	2000
http://dbpedia.org/resource/Keeping_the_Faith	Walt Disney	29000000	59945183	2000
http://dbpedia.org/resource/High_Fidelity_(film)	Walt Disney	30000000	47126295	2000
http://dbpedia.org/resource/Shanghai_Noon	Walt Disney	55000000	99274467	2000
http://dbpedia.org/resource/Mission_to_Mars	Walt Disney	100000000	110983407	2000
http://dbpedia.org/resource/Coyote_Ugly_(film)	Walt Disney	45000000	113916474	2000
http://dbpedia.org/resource/The_Crew_(2000_film)	Walt Disney	23000000	13019253	2000
http://dbpedia.org/resource/Dinosaur_(film)	Walt Disney	127500000	349800000	2000
http://dbpedia.org/resource/Disney's_The_Kid	Walt Disney	65000000	110317580	2000
http://dbpedia.org/resource/Unbreakable_(film)	Walt Disney	75000000	248118121	2000
http://dbpedia.org/resource/Down_to_You	Walt Disney	35000000	24419914	2000
http://dbpedia.org/resource/Love's_Labour's_Lost_(2000_film)	Walt Disney	13000000	299792	2000
http://dbpedia.org/resource/Remember_the_Titans	Walt Disney	30000000	136706683	2000
http://dbpedia.org/resource/The_Emperor's_New_Groove	Walt Disney	100000000	169300000	2000
http://dbpedia.org/resource/The_Tigger_Movie	Walt Disney	30000000	96159800	2000
http://dbpedia.org/resource/Gun_Shy_(film)	Walt Disney	14000000	3221280	2000
http://dbpedia.org/resource/Nine_Queens	Walt Disney	1300000	12413888	2000
http://dbpedia.org/resource/Pearl_Harbor_(film)	Walt Disney	140000000	449220945	2001
http://dbpedia.org/resource/Just_Visiting_(film)	Walt Disney	35000000	16176732	2001
http://dbpedia.org/resource/Crazy/Beautiful	Walt Disney	13000000	19937988	2001
http://dbpedia.org/resource/The_Royal_Tenenbaums	Walt Disney	21000000	71441250	2001
http://dbpedia.org/resource/Atlantis:_The_Lost_Empire	Walt Disney	105000000	186100000	2001
http://dbpedia.org/resource/Corky_Romano	Walt Disney	11000000	23900000	2001
http://dbpedia.org/resource/Bubble_Boy_(film)	Walt Disney	13000000	5007898	2001

Εικόνα 26: Τρίτο στάδιο καθαρισμού.

Τα τρία αυτά στάδια εφαρμόστηκαν μια φορά για κάθε json αρχείο, δηλαδή σύνολο 16 φορές. Τα υπολογιστικά φύλλα μετά τον καθαρισμό τους αποθηκεύτηκαν σε μορφή csv. Ήδη από το μέγεθος του κάθε αρχείου στο δίσκο, γίνεται εμφανής η διαφορά σε όγκο πληροφορίας που έχει η κάθε εταιρία διανομής.

Στη συνέχεια επιλέχθηκε το SPSS για να γίνουν μερικές ακόμα μικρές αλλά απαραίτητες τροποποιήσεις. Εδώ το SPSS χρησιμοποιείται όχι για τις δυνατότητες που προσφέρει στη στατιστική ανάλυση, αλλά για την ευκολία που προσφέρει το περιβάλλον του ως εφαρμογή υπολογιστικού φύλλου.

Αρχικά, μεταβήκαμε στη Variable View και μετονομάσαμε τις μεταβλητές σε Movie, Distributor, Budget, BoxOffice και Year αντίστοιχα. Αυτό θα μπορούσε βέβαια να γίνει και από το Open Refine, αλλά μέσω του SPSS έγινε πιο γρήγορα σε κάθε πίνακα.

Επίσης στη Variable View τροποποιήσαμε την παράμετρο width σε 102, 16, 8, 8 και 8 αντίστοιχα για κάθε μεταβλητή σε κάθε πίνακα. Αυτό έγινε διότι ίδιες εγγραφές με διαφορετικό πλήθος χαρακτήρων, θα αντιμετωπίζονταν σαν διαφορετικές από τον υπολογιστή στη συνέχεια, και αυτό θα έφερνε σημαντικά σφάλματα στις συγκρίσεις που θα ακολουθούσαν.

Από τα έσοδα και τα έξοδα κάθε ταινίας, δημιουργήθηκε μια καινούρια μεταβλητή με όνομα Difference και width 8, που περιγράφει τα καθαρά κέρδη σε κάθε εγγραφή. Υπολογίστηκε προφανώς ως εξής:

$$\text{Difference} = \text{BoxOffice} - \text{Budget} \quad (1)$$

Φυσικά σε αρκετές περιπτώσεις οι ταινίες δεν είχαν τα αναμενόμενα κέρδη και ως αποτέλεσμα τα έσοδα τους δεν κάλυψαν ούτε το κόστος παραγωγής τους. Στις περιπτώσεις αυτές η Difference παίρνει αρνητικές τιμές, που είναι και το επιθυμητό για τη συνέχεια.

Στη συνέχεια οι δύο πίνακες της Sony Pictures και της Columbia ενσωματώθηκαν σε έναν, μιας και η Columbia Pictures είναι θυγατρική της Sony Pictures από το 1989. Επιπλέον διαχωρίστηκαν από τον πίνακα της 20th Century Fox οι εγγραφές της Fox Searchlight γιατί, αν και θυγατρική εταιρία, έχει μια σημαντικά ανεξάρτητη πορεία από την ανώτερη της, κάτι που φαίνεται παρακάτω. Κατά συνέπεια το πλήθος των πινάκων παρέμεινε ίδιο στους 16.

Ολοι οι πίνακες ταξινομήθηκαν πρώτα με βάση το έτος κυκλοφορίας και μετά αλφαβητικά ανάλογα τον τίτλο της κάθε ταινίας. Στη συνέχεια διαχωρίστηκαν χρονολογικά έτσι ώστε να υπάρχει ένα αρχείο για κάθε έτος για κάθε εταιρία διανομής. Επιπλέον, με διαφορετικό διαχωρισμό, δημιουργήθηκαν αρχεία που περιέχουν τις ταινίες όλων των εταιριών σε ένα. Τέλος, αποθηκεύτηκε ένα αρχείο με όλες τις ταινίες, όλων των εταιριών διανομής, κάθε έτους. Ο τελικός αριθμός των αρχείων υπολογιστικών φύλλων έφτασε τα 272.

	Movie	Distributor	Budget	BoxOffice	Year	Difference
1	http://dbpedia.org/resource/Behind_Enemy_Lines_(2001_film)	20th Century Fox	40000000	91753222	2001	51753222
2	http://dbpedia.org/resource/Black_Knight_(film)	20th Century Fox	50000000	39976235	2001	-10023765
3	http://dbpedia.org/resource/Don't_Say_a_Word	20th Century Fox	50000000	100020092	2001	50020092
4	http://dbpedia.org/resource/Dr._Dolittle_2	20th Century Fox	71000000	176104344	2001	105104344
5	http://dbpedia.org/resource/Freddy_Got_Fingered	20th Century Fox	14000000	14300000	2001	300000
6	http://dbpedia.org/resource/Glitter_(film)	20th Century Fox	22000000	5271666	2001	-16728334
7	http://dbpedia.org/resource/Joe_Somebody	20th Century Fox	38000000	24516772	2001	-13483228
8	http://dbpedia.org/resource/Joy_Ride_(2001_film)	20th Century Fox	23000000	36642838	2001	13642838
9	http://dbpedia.org/resource/Kiss_of_the_Dragon	20th Century Fox	25000000	64400000	2001	39400000
10	http://dbpedia.org/resource/Monkeybone	20th Century Fox	75000000	7622365	2001	-67377635
11	http://dbpedia.org/resource/Moulin_Rouge!	20th Century Fox	52000000	179213434	2001	127213434
12	http://dbpedia.org/resource/Planet_of_the_Apes_(2001_film)	20th Century Fox	100000000	362211740	2001	262211740
13	http://dbpedia.org/resource/Say_It_Isn't_So_(film)	20th Century Fox	25000000	12320393	2001	-12679607
14	http://dbpedia.org/resource/Shallow_Hal	20th Century Fox	40000000	141069860	2001	101069860
15	http://dbpedia.org/resource/Someone_Like_You_(film)	20th Century Fox	23000000	38689940	2001	15689940
16	http://dbpedia.org/resource/Y_Tu_Mam%C3%A1_Tambi%C3%A9n	20th Century Fox	5000000	33616692	2001	28616692

Εικόνα 27: Τελική μορφή αρχείου της βάσης δεδομένων.

Σημειώνεται ότι μερικές εταιρίες διανομής είτε δεν είχαν καμία παραγωγή για ένα ολόκληρο έτος, είτε ιδρύθηκαν μετά το 2000 και προφανώς δεν υπάρχουν εγγραφές για αυτές μέχρι το έτος ίδρυσης τους. Στις περιπτώσεις αυτές, αποθηκεύτηκε ένα κενό αρχείο `san` με όνομα το αντίστοιχο έτος, το οποίο περιέχει το μήνυμα "No entries for this year" στο κελί (1,1).

4. ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ ΚΙΝΗΜΑΤΟΓΡΑΦΙΚΩΝ ΔΕΔΟΜΕΝΩΝ

Οι στόχοι που τέθηκαν στο στάδιο αυτό είναι οι παρακάτω:

- Υπολογισμός και ανάλυση των συνεργασιών ανάμεσα στις εταιρίες διανομής
- Αναζήτηση συσχετίσεων μεταξύ των εταιριών αυτών σε ό,τι αφορά τα οικονομικά δεδομένα τους

Αρχικά, στόχος είναι να μεταφέρουμε τα αρχεία υπολογιστικών φύλλων στην R και μάλιστα με τρόπο όσο το δυνατόν μαζικότερο μιας και το πλήθος τους είναι μεγάλο και αν ανέβουν μεμονωμένα, η διαδικασία θα είναι πολύ χρονοβόρα. Για το λόγο αυτό χρησιμοποιήθηκαν επαναληπτικές εντολές της `read.spss()` της βιβλιοθήκης `foreign`.

Πολύ σημαντικό κομμάτι για την ευκολότερη ανάλυση στα επόμενα στάδια είναι η οργάνωση των δεδομένων. Αρχικά δημιουργήθηκαν 17 λίστες υπολογιστικών φύλλων, 16 για κάθε εταιρία διανομής και μία για τα συνολικά δεδομένα. Κάθε λίστα αποτελείται από 16 πίνακες, 15 για κάθε έτος ξεχωριστά και μία για όλα τα έτη συνολικά.

Οι 17 παραπάνω λίστες εκχωρούνται όλες σε μια νέα λίστα με το όνομα `All` για ακόμα μεγαλύτερη ευκολία προσπέλασης οποιουδήποτε σημείου της βάσης δεδομένων. Τα δεδομένα για κάθε εταιρία διανομής και για κάθε χρονιά τυπώνονται γράφοντας `All[[i]][[j]]`, όπου *i* και *j* οι παράμετροι όπως φαίνονται στον παρακάτω πίνακα.

Πίνακας 3: Οι παράμετροι για κάθε εταιρία διανομής για κάθε έτος.

i	Εταιρία Διανομής	j	Έτος
1	20th Century Fox	1	2000
2	Walt Disney	2	2001
3	DreamWorks	3	2002
4	Focus Features	4	2003
5	Fox Searchlight	5	2004
6	Lionsgate	6	2005
7	Metro Goldwyn Mayer	7	2006
8	Miramax	8	2007
9	New Line Cinema	9	2008
10	Paramount Pictures	10	2009
11	Relativity Media	11	2010
12	Sony Pictures	12	2011
13	Summit Entertainment	13	2012
14	Universal Pictures	14	2013
15	Warner Bros.	15	2014
16	The Weinstein Company	16	Όλα τα έτη
17	Όλες οι εταιρίες	-	-

4.1 Συνεργασίες εταιριών διανομής

4.1.1 Υπολογισμός πλήθους συνεργασιών

Ενα αρκετά μεγάλο ποσοστό ταινιών δεν διανεμήθηκε μόνο από μια εταιρία, αλλά συνεργατικά από δύο ή μερικές φορές και τρεις διαφορετικές εταιρίες. Επιχειρήσαμε λοιπόν να εντοπίσουμε τα ζευγάρια εταιριών που συνεργάζονται πιο συχνά καθώς και τις εταιρίες που διανέμουν συχνότερα ταινίες μόνες τους. Για το σκοπό αυτό δημιουργήθηκε ένας κενός πίνακας 16x16 στην R που ονομάσαμε `Common`. Οπου `Common[i,j]` θα εκχωρηθεί το συνολικό πλήθος κοινών ταινιών ανάμεσα στις εταιρίες i και j .

Σημειώνεται ότι για να προσπελαστεί ολόκληρος ο όγκος των δεδομένων μέσω της λίστας `All` χρειάστηκαν 5 επαναληπτικές εντολές η μία μέσα στην άλλη για να επεξεργάζονται όλους τους απαραίτητους δείκτες. Πρακτικά συγκρίνονται οι εγγραφές της πρώτης στήλης κάθε ζευγαριού πινάκων εταιριών διανομής ανά έτος και αν αυτές είναι όμοιες προστίθεται 1 στο αντίστοιχο στοιχείο του πίνακα `Common`. Οι επαναληπτικές χρησιμοποιήθηκαν για να τρέχουν το έτος, την πρώτη και δεύτερη εταιρία διανομής καθώς και τις γραμμές του κάθε πίνακα της πρώτης και δεύτερης εταιρίας διανομής.

Για να προστεθεί 1 στο στοιχείο του πίνακα i πρέπει να ικανοποιούνται δύο συνθήκες. Η μία είναι και η προφανής, που ελέγχει αν οι δύο συγκρινόμενες εγγραφές είναι ίδιες. Η δεύτερη προστίθεται αναγκαστικά για τις περιπτώσεις που οι εταιρίες δεν έχουν διανείμει καμία ταινία για ένα ολόκληρο έτος. Στον καθαρισμό, οι περιπτώσεις αυτές αντικαταστάθηκαν από έναν πίνακα με ένα μόνο κελί που περιέχει το κείμενο "no entries for this year". Το πρόγραμμα θα τις αντιμετώπιζε σαν ίδιες και θα συνέχιζε να προσθέτει στον πίνακα `Common` χωρίς στην πραγματικότητα να υπάρχουν όμοιες ταινίες. Για το λόγο αυτό προστέθηκε η συνθήκη κάθε εγγραφή να είναι διαφορετική από την "no entries for this year" πριν ελεγχθεί για ομοιότητα. Προφανώς ο πίνακας `Common` είναι συμμετρικός και για οικονομία χρόνου υπολογίστηκαν μόνο τα άνω τριγωνικά στοιχεία του.

Μαθηματικά, αν $t_v, v=1,2,\dots$ το σύνολο όλων των ταινιών, και `All[i]`, `All[j]` οι πίνακες ταινιών των i και j εταιριών διανομής, το στοιχείο `Common(i,j)`, $i,j=1,2,\dots,16$ του πίνακα που θέλουμε, υπολογίζεται από τον παρακάτω τύπο:

$$Common(i, j) = Common(i, j) + 1, \text{αν } \tau_v \in All[i] \text{ και } \tau_v \in All[j],$$

$$\text{όπου } \tau_v \neq \text{"no entries for this year"} \quad (2)$$

Ο πίνακας που προέκυψε αφού προστέθηκε με τον ανάστροφο του φαίνεται παρακάτω:

	20th Century Fox	Walt Disney	DreamWorks	Focus Features	Fox Searchlight	Lionsgate	Metro Goldwyn Mayer	Miramax	New Line Cinema	Paramount	Relativity Media	Sony Pictures	Summit Entertainment	Universal	Warner Bros.	Weinstein
20th Century Fox	0	1	11	0	0	1	12	2	2	2	1	7	4	3	3	5
Walt Disney	1	0	10	1	0	1	0	10	0	2	0	1	3	2	3	1
DreamWorks	11	10	0	0	0	0	0	0	0	34	1	5	0	8	6	0
Focus Features	0	1	0	0	0	1	0	2	2	1	7	3	0	25	3	0
Fox Searchlight	0	0	0	0	0	0	0	1	0	1	0	0	1	1	2	0
Lionsgate	1	1	0	1	0	0	0	1	0	3	6	3	18	7	2	4
Metro Goldwyn Mayer	12	0	0	0	0	0	0	1	3	6	3	17	1	4	3	15
Miramax	2	10	0	2	1	1	1	0	0	4	0	4	1	8	3	6
New Line Cinema	2	0	0	2	0	0	3	0	0	1	1	2	0	1	24	0
Paramount	2	2	34	1	1	3	6	4	1	0	4	5	2	3	10	4
Relativity Media	1	0	1	7	0	6	3	0	1	4	0	36	0	64	1	3
Sony Pictures	7	1	5	3	0	3	17	4	2	5	36	0	0	13	15	2
Summit Entertainment	4	3	0	0	1	18	1	1	0	2	0	0	0	8	10	1
Universal	3	2	8	25	1	7	4	8	1	3	64	13	8	0	4	3
Warner Bros.	3	3	6	3	2	2	3	3	24	10	1	15	10	4	0	1
Weinstein	5	1	0	0	0	4	15	6	0	4	3	2	1	3	1	0

Εικόνα 28: Ο πίνακας *Common*.

4.1.2 Ανάλυση των συνεργασιών

Τα παραπάνω αποτελέσματα μπορεί να περιγράφουν το πλήθος σχέσεων ανάμεσα στις διάφορες εταιρίες διανομής, αλλά δεν δείχνουν εμφανώς το συνολικό πλήθος ταινιών που διένειμε η κάθε εταιρία. Το αναμενόμενο είναι ότι ανάλογα το μέγεθος της εταιρίας και κατά συνέπεια το πλήθος ταινιών που φέρουν το όνομα της, θα αυξάνονταν και οι συνεργασίες στις οποίες συμμετέχει, όπως και οι ταινίες στις οποίες είναι το μόνο όνομα στη διανομή. Όπως θα φανεί παρακάτω, αυτό απέχει πολύ από την πραγματικότητα.

Αρχικά χρειαζόμαστε το συνολικό πλήθος ταινιών που διένειμε κάθε εταιρία στη 15ετία που εξετάζουμε. Αυτό υπολογίζεται εύκολα, καθώς το μόνο που χρειάζεται είναι να εντοπίσουμε το πλήθος γραμμών του πίνακα που περιέχει όλες τις εγγραφές για κάθε εταιρία. Στην R το πλήθος γραμμών δίνεται από τη συνάρτηση `nrow()` και στην προκειμένη περίπτωση

για την i εταιρία διανομής ο πίνακας αυτός έχει εκχωρηθεί στη λίστα All στο σημείο All[[i]][[16]].

Επιπλέον, χρειαζόμαστε το συνολικό πλήθος συνεργασιών που είχε η κάθε εταιρία με όλες τις υπόλοιπες. Για να το υπολογίσουμε αυτό χρησιμοποιούμε τον πίνακα Common που κατασκευάσαμε παραπάνω, και με μια επαναληπτική εντολή προσθέτουμε όλα τα στοιχεία κάθε γραμμής (ή κάθε στήλης αφού ο πίνακας είναι συμμετρικός).

Τέλος, υπολογίζουμε το ποσοστό του πλήθους των ταινιών που διανεμήθηκαν από την κάθε εταιρία σε συνέργασία με άλλες, προς το σύνολο των ταινιών που διανεμήθηκαν από την εταιρία αυτή με ακρίβεια 2 δεκαδικών ψηφίων. Η διαφορά του κάθε αριθμού από τη μονάδα περιγράφει το ποσοστό ταινιών που διανεμήθηκαν αποκλειστικά από την αντίστοιχη εταιρία προς το συνολικό πλήθος ταινιών της.

Στον παρακάτω πίνακα (Πίνακας 4) φαίνονται αναλυτικά οι τιμές που προκύπτουν από την παραπάνω ανάλυση.

Πίνακας 4: Στοιχεία συνεργασιών ανάμεσα στις εταιρίες διανομής.

Εταιρία διανομής	Συνολικό πλήθος ταινιών	Συνολικό πλήθος συνεργασιών	Ποσοστό συνεργασιών	Ποσοστό αποκλειστικών κυκλοφοριών
20th Century Fox	241	54	22,41%	77,59%
Walt Disney	190	35	18,42%	81,58%
DreamWorks	111	75	67,57%	32,43%
Focus Features	80	45	56,25%	43,75%
Fox Searchlight	66	6	9,09%	90,91%
Lionsgate	185	47	25,41%	74,59%
Metro Goldwyn Mayer	101	65	64,36%	35,64%
Miramax	102	43	42,16%	57,84%
New Line Cinema	122	36	29,51%	70,49%
Paramount Pictures	234	82	35,04%	64,96%
Relativity Media	147	127	86,39%	13,61%
Sony Pictures	320	113	35,31%	64,69%
Summit Entertainment	75	49	65,33%	34,67%
Universal Pictures	276	154	55,8%	44,2%
Warner Bros.	341	90	26,39%	73,61%
The Weinstein Company	101	45	44,55%	55,45%

4.2 Αναζήτηση συσχετίσεων στα οικονομικά δεδομένα

Κάθε εταιρία διανέμει ένα συγκεκριμένο αριθμό ταινιών κάθε χρόνο. Αυτός ο αριθμός πολλές φορές διαφέρει από χρονιά σε χρονιά και ακόμη περισσότερο από εταιρία σε εταιρία. Υπάρχουν εταιρίες με 20 ή 25 ταινίες σε ένα έτος, και εταιρίες με πλήρη αδράνεια και καμία διανομή στο ίδιο χρονικό διάστημα. Αυτό φυσικά έχει αντίκτυπο στα αντίστοιχα έσοδα και τα έξοδα του έτους αυτού. Παρ' όλα αυτά αν θεωρήσουμε τη βιομηχανία διανομής ταινιών ως μια μικρογραφία του χρηματιστηρίου, τότε έχει νόημα να ελέγξουμε συσχετίσεις στις χρονοσειρές που προκύπτουν, όταν τα συνολικά κέρδη κάθε εταιρίας ανά έτος συγκριθούν με τα αντίστοιχα των υπόλοιπων εταιριών.

Θα περιγραφεί η διαδικασία:

- Δημιουργίας των χρονοσειρών.
- Αναζήτησης συσχετίσεων με διάφορα στατιστικά μέτρα.

4.2.1 Δημιουργία χρονοσειρών

Αρχικά θα πρέπει να δημιουργηθεί ο πίνακας των χρονοσειρών με τα κέρδη των εταιριών ανά έτος. Για να το πετύχουμε αυτό χρησιμοποιούμε τη στήλη Difference από κάθε πίνακα, που υπολογίστηκε στο SPSS. Στη συνέχεια προστίθενται όλες οι εγγραφές ανα έτος της στήλης αυτής και κατόπιν εκχωρούνται σε έναν πίνακα 16 διανομέων επί 15 έτη που ονομάζουμε Profit. Ειδικά για τις περιπτώσεις που κάποια εταιρία δεν κυκλοφόρησε καμία ταινία σε κάποιο έτος, στο αντίστοιχο στοιχείο του πίνακα Profit εκχωρούμε 0.

	20th Century Fox	Walt Disney	DreamWorks	Focus Features	Fox Searchlight	Lionsgate	Metro Goldwyn Mayer	Miramax	New Line Cinema	Paramount	Relativity Media	Sony Pictures	Summit Entertainment	Universal	Warner Bros.	Weinstein
1	1204480810	958163459	1393332739	69853509	32055161	65796312	-15403090	74829054	220811437	864023434	0	901208189	0	1548214612	204983975	0
2	674729593	1031874768	866613743	-13896183	30995647	96268730	373185404	791237805	1059446097	537447509	0	382766568	168088341	1812264268	1613129123	0
3	1467805410	1050014496	988386966	156030050	95396402	49222402	383427757	881662949	1311587275	600241875	0	1534763836	0	995876134	826918598	0
4	835879712	2080547783	205838385	165876072	121165	89753381	165383164	557890555	1491527778	395783987	0	1500215131	16050575	1442818775	1311903693	0
5	1892064007	676504763	1926306310	152642197	207749193	446256175	43950697	881094624	269337837	513630519	0	1427869724	0	1186938139	2109051063	-9147936
6	185593978	1519874309	1546249592	389720404	29657855	402292961	288885941	168806357	319758414	821210144	0	810987557	446405398	1125584340	1535695812	219850107
7	2250583577	1665217891	503696163	15743631	230034097	276024049	6041133284	177374766	243871958	1227486298	705627465	2499964710	218651117	889003749	958025539	162550612
8	1344406551	2222828003	1849190508	216174729	260255967	308521254	427649483	228970287	496517104	2467954650	575475697	1187094542	137625868	1347224798	2604603249	317336452
9	2175488983	1408733423	701949747	299723779	447786454	420653053	598884124	27916563	679779161	2244942840	2235919173	1651367428	770732914	1729235578	2272673123	354403796
10	4454851383	1402700000	965050747	69094265	92848567	366647232	115297374	0	581658705	1524871117	1205146452	2211617870	801595926	962810090	2244429470	-3379746
11	1080351447	2458729575	1171171201	183642643	40086752	622862957	28572262	48806737	455302940	2436945318	1361968358	1274551536	1290868058	1173626849	2758811942	542609192
12	1746540269	1690228601	1555912409	103879794	316515470	33061179	232470189	158000000	393932281	3473004063	1814038157	1895758901	879020161	1501145503	2509363440	365993793
13	1811018781	1851489844	952565551	150522252	28238338	1703763182	1936981051	0	231111373	1490036621	1259962985	3164590127	912959762	2602424979	1879267879	334375338
14	1992228619	2752200000	615575350	79007317	204388135	1309431366	1212198238	0	1369100177	1690529199	1309709411	1364052593	715912031	3409036298	3312410899	328232946
15	435589621	2013300000	1012000000	137983299	195900000	1045959384	1277854610	-25600000	1374254610	2314646826	289556826	1380009889	356679526	1908718299	3245410166	409823083

Εικόνα 29: Ο πίνακας Profit.

Μαθηματικά, ο πίνακας Profit υπολογίζεται με τον παρακάτω τύπο:

$$Profit(i,j) = \sum_{k=1}^n All[i][j] \cdot Difference[k] \quad (3)$$

$$i = 1,2, \dots, 16 \quad j = 1,2, \dots, 15$$

4.2.2 Ανάλυση χρονοσειρών

Για την ανάλυση των παραπάνω χρονοσειρών χρησιμοποιήθηκαν 4 στατιστικές μέθοδοι:

- Cross Correlation (Zero latency)
- Cross Correlation (One latency)
- Mutual Information
- Granger Causality

4.2.2.1 Cross Correlation

Αρχικά ελέγξαμε τυχόν γραμμικές συσχετίσεις που υπάρχουν, όχι μόνο στον ίδιο χρόνο, αλλά και με ένα έτος διαφορά. Για να το πετύχουμε αυτό χρησιμοποιήσαμε συνάρτηση cross correlation της MATLAB. Η ίδια εργασία θα μπορούσε να γίνει και στην R, όμως η MATLAB περιέχει πιο εύκολες στη χρήση εντολές για στατιστικό έλεγχο των αποτελεσμάτων, καθώς και για τον ένα χρόνο καθυστέρηση που επιθυμούμε να υπολογίσουμε. Η συναρτήσεις που χρησιμοποιήθηκαν είναι η crosscorr() για τον υπολογισμό του cross correlation για καθυστέρηση 0 και 1, ενώ για τον στατιστικό έλεγχο χρησιμοποιήθηκε η corrcoef στη μορφή της [R,P] = corrcoef() ώστε σαν αποτέλεσμα να επιστραφεί και ο πίνακας P ο οποίος περιέχει τις p-value τιμές στατιστικής σημαντικότητας.

Προέκυψαν συνολικά 4 πίνακες, οι 2 με τις τιμές της συσχέτισης για τις δύο καθυστερήσεις και οι άλλοι 2 με τις αντίστοιχες τιμές της p-value, οι οποίοι περάστηκαν στην R για περαιτέρω επεξεργασία. Οι 2 πρώτοι πίνακες θα χρησιμοποιηθούν ως πίνακες γειτνίασης για να σχεδιαστούν γράφοι που θα μπορεί ο χρήστης να δει και να επεξεργαστεί στην εφαρμογή. Στους πίνακες p-value που προφανώς περιέχονται τιμές από το 0 ως το 1, ορίζεται ως κατώφλι το 95% (0.95) και οι υπόλοιπες διαγράφονται. Με τη βοήθεια των

πινάκων αυτών διαγράφουμε από τους πρώτους πίνακες όσα στοιχεία έχουν ίδιες συντεταγμένες με τα μηδενικά στοιχεία στους p-value πίνακες. Ετσι δημιουργούνται 2 καινούριοι πίνακες συσχετίσεων, που περιέχουν μόνο τις στατιστικά σημαντικές τιμές.

Ο υπολογισμός του Cross Correlation έγινε με τους παρακάτω τύπους:

$$r_{y_1 y_2}(k) = \frac{c_{y_1 y_2}(k)}{s_{y_1 y_2}}; k = 0, \pm 1, \pm 2, \dots \quad (4)$$

όπου

$$c_{y_1 y_2}(k) = \begin{cases} \frac{1}{T} \sum_{t=1}^{T=k} (y_{1t} - \bar{y}_1)(y_{2,t+k} - \bar{y}_2); k = 0, 1, 2, \dots \\ \frac{1}{T} \sum_{t=1}^{T=k} (y_{2t} - \bar{y}_2)(y_{1,t+k} - \bar{y}_1); k = 0, -1, -2, \dots \end{cases} \quad (5)$$

όπου $\bar{y}_1$ και $\bar{y}_2$ οι μέση τιμή της κάθε χρονοσειράς.

4.2.2.2 Mutual Information

Το εργαλείο που επιλέχθηκε για την αναζήτηση μη γραμμικών συσχετίσεων στις παραπάνω χρονοσειρές ήταν η αμοιβαία πληροφορία. Από τις πολλές εντολές που υπάρχουν σε διάφορες βιβλιοθήκες της R επιλέχθηκε η `makemim()` της βιβλιοθήκης `c3net`. Αυτό γιατί η εντολή `sigtestp()` επιστρέφει ένα περιβάλλον, έστω `sigMI`, του οποίου το στοιχείο `sigMI$Inew` επιστρέφει τον παραπάνω πίνακα της αμοιβαίας πληροφορίας με μηδενισμένες τις μη στατιστικά σημαντικές τιμές. Ετσι ο εντοπισμός αυτών των τιμών είναι πολύ ευκολότερος.

Ο υπολογισμός της αμοιβαίας πληροφορίας [3] από τις παραπάνω εντολές, γίνεται μέσω του τύπου:

$$\hat{I}(X:Y) = \frac{1}{N} \sum_{i=1}^N \log \frac{\hat{f}(x_i y_i)}{\hat{f}(x_i) \hat{f}(y_i)} \quad (6)$$

4.2.2.3 Granger Causality

Το τελευταίο στατιστικό μέτρο που χρησιμοποιήθηκε είναι η αιτιότητα Granger. Μέσω αυτού έγινε προσπάθεια να εντοπιστούν μονόδρομες συσχετίσεις στις παραπάνω χρονοσειρές, καθώς μια εταιρία διανομής θα μπορούσε να ασκεί επιρροή σε μια άλλη, χωρίς απαραίτητα να συμβαίνει το αντίστροφο. Το πλήθος των εντολών για υπολογισμό αιτιότητας Granger στην R ήταν σαφώς πιο περιορισμένο σε σχέση με τα προηγούμενα στατιστικά μέτρα. Τελικά επιλέχθηκε η εντολή `granger.test` της βιβλιοθήκης `MSBVAR`.

Η παραπάνω εντολή δέχεται σαν `input` τον πίνακα των χρονοσειρών, καθώς και έναν ακέραιο, ο οποίος δηλώνει την καθυστέρηση με την οποία θέλει ο χρήστης να υπολογίσει τη συσχέτιση. Εδώ χρησιμοποιήθηκε ο πίνακας `Profit` και καθυστέρηση 1, δηλαδή πως επηρεάζει η απόδοση μιας εταιρίας, την απόδοση κάθε άλλης την επόμενη χρονιά.

Η εντολή επιστρέφει ένα διάνυσμα αντί για πίνακα με τις επιθυμητές τιμές. Για την καλύτερη κατανόηση των συσχετίσεων αλλά και την ευκολότερη περαιτέρω ανάλυση κρίθηκε απαραίτητη η μετατροπή αυτού του διανύσματος σε πίνακα. Η μετατροπή αυτή έγινε αρχικά “με το χέρι” για μεγαλύτερη σιγουριά ότι όλα τα στοιχεία θα πάνε ακριβώς στις θέσεις του πίνακα που πρέπει και στη συνέχεια συντάχθηκε και κατάλληλο κομμάτι κώδικα το οποίο επιστρέφει αυτόματα τον επιθυμητό πίνακα και μπορεί να χρησιμοποιηθεί εκ νέου σε κάθε ανάλυση.

```
grdat<-granger.test(Profit, 1)
z=0
for (i in 1:ncol(Profit)) {
  for (j in 1:ncol(Profit)) {
    if (i==j) {
      grang[i,j]<-0
    } else {
      z=z+1
      grang[i,j]<-grdat[z,1]
    }
  }
}
```

Ο έλεγχος στατιστικής σημαντικότητας έγινε μέσω 1000 ανακατατάξεων του πίνακα χρονοσειρών χειροκίνητα αυτή τη φορά, μιας και δεν υπήρχε αντίστοιχη βοηθητική εντολή.

Μετά τον υπολογισμό και την ταξινόμηση των 1000 αντίστοιχων πινάκων Granger οι τιμές που ήταν μεγαλύτερες της 950 θεωρήθηκαν στατιστικά σημαντικές.

Ο υπολογισμός της αιτιότητας Granger [4] από την παραπάνω εντολή, γίνεται μέσω του τύπου:

$$X(t) = \sum_{T=1}^L A_T X(t-T) + c(t) \quad (7)$$

5. ΟΠΤΙΚΟΠΟΙΗΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Όλα τα παραπάνω αποτελέσματα παρουσιάζονται σε μια εφαρμογή που δημιουργήθηκε με τις εντολές της βιβλιοθήκης "shinydashboard". Η εφαρμογή αυτή αποτελείται από 6 καρτέλες:

- Network
- Bar Chart
- Pie Chart
- Data
- Time Series
- Statistics

Η τελική εφαρμογή είναι διαθέσιμη στον Ιστό¹

Ο κώδικας αλλά και το σύνολο των δεδομένων που χρειάζεται για να λειτουργήσει είναι ανεβασμένα στο github²

5.1 Προσπέλαση δεδομένων από το χρήστη

Αρχικά, ήταν απαραίτητο να δημιουργηθεί μια καρτέλα στην οποία ο χρήστης θα μπορεί να προσπελάσει όποιον από τους 272 πίνακες της βάσης δεδομένων θέλει. Έτσι, δημιουργήθηκε η καρτέλα Data. Αρχικά τοποθετήθηκαν δυο widgets με τη selectInput(). Με το ένα ο χρήστης μπορεί να επιλέγει την εταιρία διανομής που θέλει, ενώ με το άλλο το έτος. Πρακτικά, με το ένα να χειρίζεται τη μεταβλητή i του παραπάνω πίνακα και με το άλλο την j.

¹ <http://linkedanalytics.okfn.gr:3838/sample-apps/movies/>

² <https://github.com/NLivanos/Cinema-Linked-Data>

Movie	Distributor	Budget	BoxOffice	Year	Difference
1096 http://dbpedia.org/resource/%C3%86on_Flux_(film)	Paramount Pictures	62000000	52304001	2005	-9695999
2435 http://dbpedia.org/resource/10,000_BC_(film)	Warner Bros	105000000	269784201	2008	164784201
477 http://dbpedia.org/resource/127_Hours	Fox Searchlight	18000000	60700000	2010	42700000
2477 http://dbpedia.org/resource/127_Hours	Warner Bros	18000000	60700000	2010	42700000
1546 http://dbpedia.org/resource/12_(2007_film)	Sony Pictures	2500000	7500000	2007	5000000
493 http://dbpedia.org/resource/12_Years_a_Slave_(film)	Fox Searchlight	22000000	187700000	2013	165700000
1765 http://dbpedia.org/resource/12_Years_a_Slave_(film)	Summit Entertainment	22000000	187700000	2013	165700000
2022 http://dbpedia.org/resource/12_Years_a_Slave_(film)	Universal	22000000	187700000	2013	165700000
1468 http://dbpedia.org/resource/13_Going_on_30	Columbia Pictures	37000000	96455697	2004	59455697
746 http://dbpedia.org/resource/1408_(film)	Metro Goldwyn Mayer	25000000	131998242	2007	106998242

Εικόνα 30: Η καρτέλα Data της τελικής εφαρμογής.

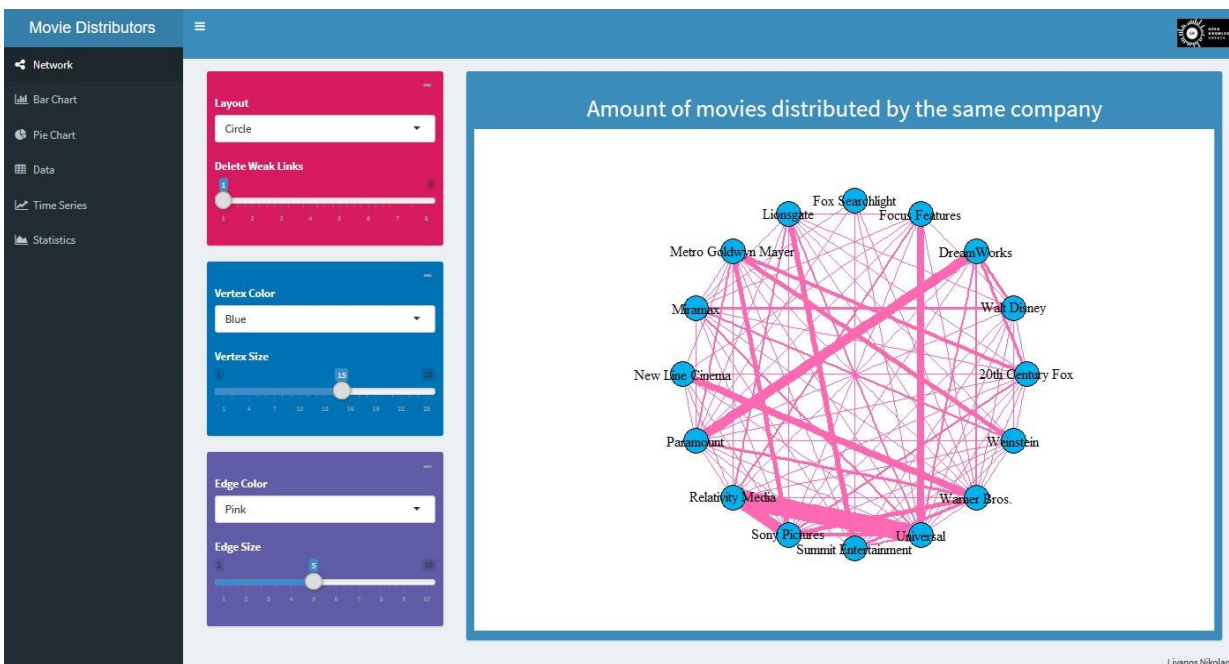
Για την εμφάνιση του κάθε πίνακα με την παραπάνω μορφή, χρησιμοποιήθηκαν οι εντολές της βιβλιοθήκης "DT" που ειδικεύεται ακριβώς στο κομμάτι αυτό, δηλαδή στον όμορφο και πρακτικό σχεδιασμό dataframe σε εφαρμογές οπτικοποίησης. Συγκεκριμένα, στην `dashboardPage()` χρησιμοποιήθηκε η `DT::dataTableOutput()` και στη `server` η `DT::renderDataTable()`. Μόνο με τις παραπάνω εντολές, δημιουργείται ένα περιβάλλον σχεδιασμού πίνακα, το οποίο περιέχει αριθμημένες τις εγγραφές, τη δυνατότητα φθίνουσας ή αύξουσας ταξινόμησης σύμφωνα με όποια στήλη επιθυμεί ο χρήστης, επίλογη για πλήθος εγγραφών που εμφανίζονται σε 10, 25, 50 ή 100, διαχωρισμός των εγγραφών σε σελίδες, αλλά και αναζήτηση εγγραφών στον πίνακα. Τέλος, προστέθηκε μια επιλογή Download, με την οποία ο χρήστης μπορεί να κατεβάσει στον υπολογιστή του τα δεδομένα που εμφανίζονται στην οθόνη του, σε μορφή .csv.

5.2 Οπτικοποίηση μέσω δικτύων

Για την οπτικοποίηση του δικτύου συνεργασιών κατασκευάστηκε ένας γράφος του οποίου ο πίνακας γειτνίασης είναι ο Common. Τα ονόματα στους κόμβους θα περιγράφουν τους διανομείς ενώ το πάχος των ακμών θα είναι ανάλογο του πλήθους των συνεργασιών που είχαν μέσα στη 15ετία.

Για την οπτικοποίηση του γράφου στην εφαρμογή χρησιμοποιήθηκε η `plotOutput()`. Το δίκτυο που προκύπτει είναι σχεδόν πλήρες, οπότε για την εξαγωγή αξιολογών συμπερασμάτων, διαγράφηκαν σταδιακά οι πιο αδύναμες ακμές. Τα στάδια διαγραφής ακμών είναι 8 σε πλήθος και για το καθένα σχεδιάστηκε και ένα ξεχωριστό δίκτυο. Τα 8 αυτά δίκτυα αποθηκεύτηκαν σε μία λίστα με όνομα "Graphs" και στην εφαρμογή προστέθηκε ένα widget μπάρας κύλισης με 8 θέσεις και με όνομα "Delete weak links". Όσο πιο δεξιά τοποθετεί ο χρήστης την επιλογή στη μπάρα αυτή, τόσο πιο "αυστηρό" είναι το φίλτρο που διαγράφει τις ακμές.

Για την καλύτερη επεξεργασία της οπτικοποίησης προστέθηκαν 4 ακόμη widget με τα οποία ο χρήστης μπορεί να επεξεργαστεί το μέγεθος και το χρώμα των κόμβων και των ακμών. Συγκεκριμένα του δίνεται η δυνατότητα να επιλέξει ανάμεσα σε 7 χρώματα και για τους κόμβους και για τις ακμές, μέσω 2 widget της εντολής `selectInput()` καθώς και να επεξεργαστεί το μέγεθος τους μέσω 2 μπαρών κύλισης με 25 θέσεις για τους κόμβους και 10 θέσεις για τις ακμές.

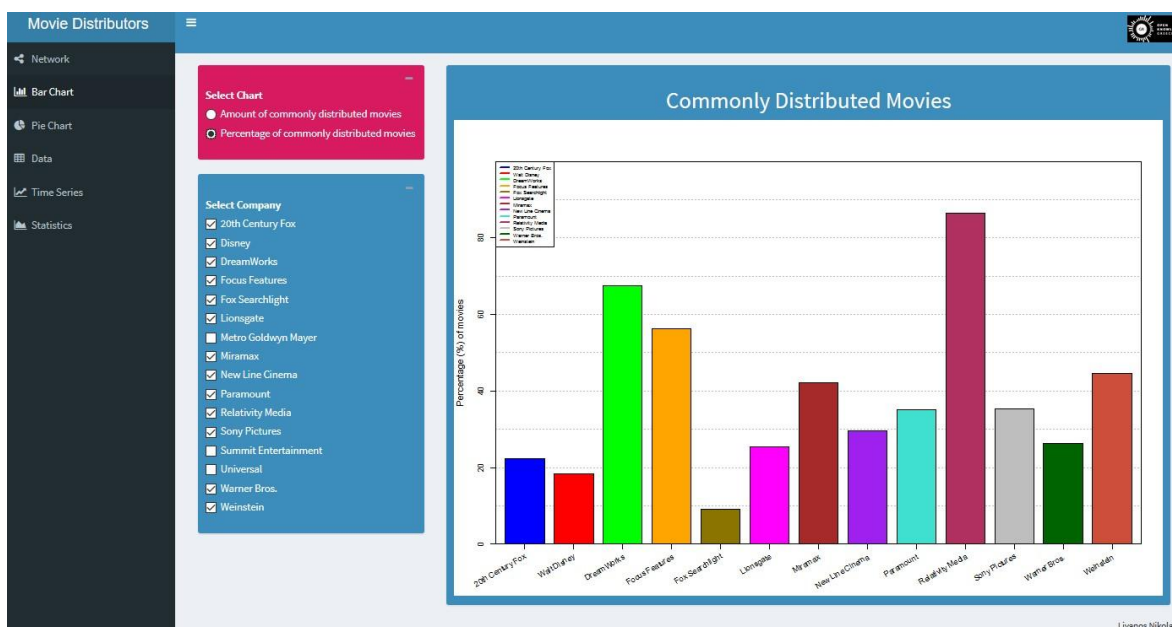


Εικόνα 31: Η καρτέλα Network.

Τέλος, δίνεται η δυνατότητα στο χρήστη να επιλέξει το layout του σχεδιασμένου γράφου, ανάμεσα στα 4 πιο δημοφιλή: Circle, Fruchterman-Reingold, Sphere και Kamada-Kawai. Ο υπολογιστής δέχεται το input του χρήστη και στη συνέχεια στη συνάρτηση `server()` μέσω της εντολής `observe()` και τη χρήση μίας `if` σχεδιάζει το ανάλογο δίκτυο. Τα widgets ταξινομήθηκαν σε 3 boxes με διαφορετικά χρώματα ανάλογα με τη χρήση τους. Στο πρώτο box τοποθετήθηκαν αυτά που επεξεργάζονται το layout και τη διαγραφή ακμών, στο δεύτερο το χρώμα και το μέγεθος των κόμβων και στο τρίτο το χρώμα και το μέγεθος των ακμών. Τα κουτιά έχουν και την επιλογή `collapsible = TRUE`, που προσθέτει πάνω δεξιά την επιλογή για ελαχιστοποίηση, εφ'όσον ο χρήστης δε σκοπεύει να χρησιμοποιήσει τα αντίστοιχα widgets.

5.3 Οπτικοποίηση μέσω γραφημάτων

Τα δεδομένα του πίνακα 4 οπτικοποιούνται στις καρτέλες Bar Chart και Pie Chart. Ο χρήστης μπορεί να σχεδιάσει ένα ραβδόγραμμα ή ένα κυκλικό διάγραμμα με αρχικά δεδομένα οποιαδήποτε στήλη από τον πίνακα επιθυμεί.

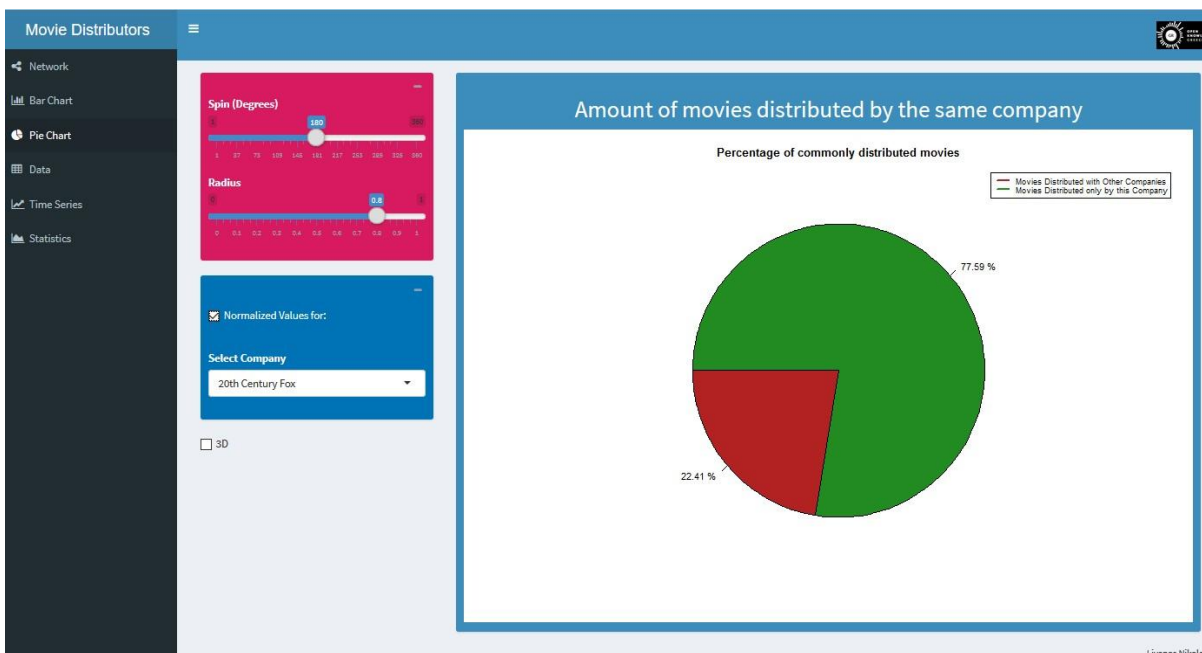


Εικόνα 32: Η καρτέλα Bar Chart.

Ειδικά στην καρτέλα Bar Chart, δίνεται στο χρήστη η δυνατότητα να σχεδιάσει κάθε ραβδόγραμμα μόνο με τις εταιρίες διανομής που αυτός θέλει να επιλέξει, απλά τσεκάροντας τα σωστά κουτάκια στο αντίστοιχο widget. Όλα τα ραβδογράμματα σχεδιάζονται με τη συνάρτηση `barp()` της βιβλιοθήκης `plotrix`.

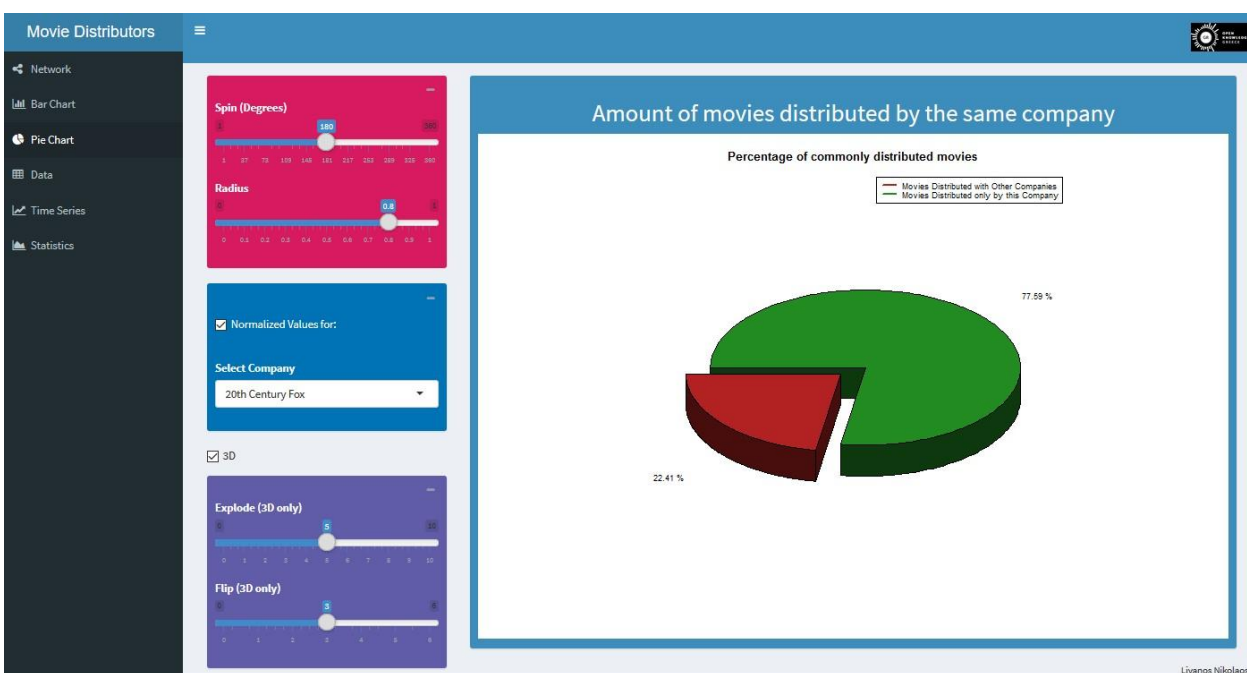
Για να γίνει εμφανής η σημασία κάθε τιμής του πίνακα 4 σχεδιάστηκε ένα κυκλικό γράφημα για κάθε εταιρία διανομής, 16 στο σύνολο, τα οποία αποτελούνται από δύο μόνο τομείς, σύμφωνα με τα ζευγάρια τιμών της `normsum`. Έτσι ο χρήστης μπορεί να δει για κάθε εταιρία διανομής και το πλήθος ταινιών στο οποίο συμμετείχε συνεργατικά με άλλες, αλλά και το ποσοστό της τιμής αυτής σε σχέση με το συνολικό πλήθος ταινιών που ορίστηκε ως διανομέας.

Ο χρήστης μπορεί μέσω της εφαρμογής να επεξεργαστεί την ακτίνα και την περιστροφή της κάθε "πίτας" με δύο widgets μπάρας κύλισης. Επιπλέον, μπορεί να εναλλάσσεται μεταξύ των γραφημάτων που περιγράφουν το πλήθος ταινιών και αυτών που περιγράφουν ποσοστό απλά επιλέγοντας στο κουτάκι με την ετικετα "Normalized values for:" και την αντίστοιχη εταιρία διανομής



Εικόνα 33: Η καρτέλα Pie Chart με κανονικοποιημένες τιμές.

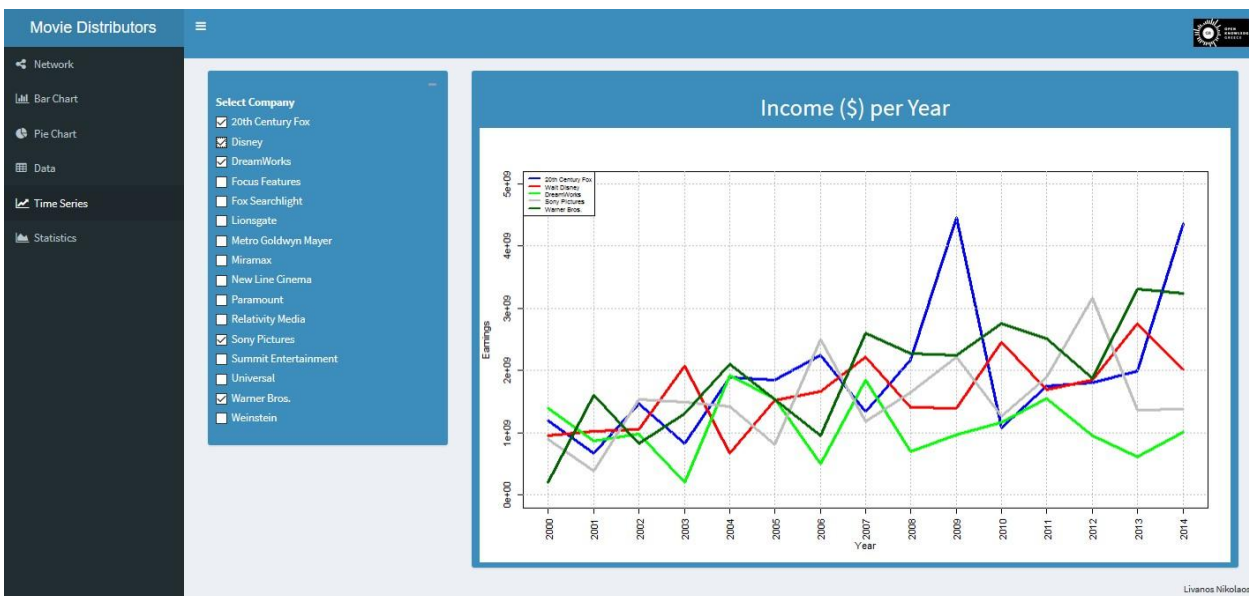
Για πιο εντυπωσιακά αποτελέσματα οπτικοποίησης χρησιμοποιήθηκαν οι εντολές της βιβλιοθήκης plotrix. Συγκεκριμένα, με την εντολή pie3D() δίνεται η δυνατότητα στο χρήστη να σχεδιάσει όλα τα παραπάνω κυκλικά διαγράμματα σε τρισδιάστατο σχέδιο. Για να το πετύχει αυτό επιλέγει το κουτάκι με ετικέτα 3D. Παράλληλα, εμφανίζεται ένα επιπλέον box με δύο widgets που αφορούν την επεξεργασία μόνο των τρισδιάστατων γραφημάτων, και συγκεκριμένα το πρώτο ξεχωρίζει και απομακρύνει τους τομείς από το κέντρο του κύκλου και το δεύτερο στρέφει όλο το γράφημα ως προς την τρίτη διάσταση.



Εικόνα 34: Η καρτέλα Pie Chart με ενεργή την επιλογή 3D.

5.4 Παρουσίαση των χρονοσειρών

Στόχος της καρτέλας "Time Series" είναι να προβάλλει την πληροφορία του πίνακα Profit με τον καλύτερο δυνατό τρόπο. Επιλέχθηκε ένα διάγραμμα "γραμμής" στο οποίο ο χρήστης έχει τη δυνατότητα να επιλέξει πόσες και ποιες εταιρίες θέλει να εμφανίσει.



Εικόνα 35: Η καρτέλα Time Series.

Στον κώδικα της `dashboardPage()` το μόνο που χρειαζόταν ήταν να τοποθετηθούν τα 2 πάνελ με στη σωστή θέση, και φυσικά το παραπάνω widget με το συνδεδεμένο plot. Ο κώδικας της `server()` είναι λίγο πιο πολύπλοκος, μιας και χρειάζεται να εντοπίζει τις επιλογές του χρήστη, όσες και όποιες και να είναι, και να σχεδιάζει τις αντίστοιχες γραμμές.

Στην R για να προσθέσει κάποιος γραμμές σε ένα ήδη σχεδιασμένο plot χρησιμοποιεί την εντολή `lines()`. Οπότε διακρίνονται οι 3 παρακάτω περιπτώσεις:

1) Ο χρήστης δεν έχει καμία επιλογή ενεργή: Στην περίπτωση αυτή το λογικότερο θα ήταν να μην εμφανίζεται τίποτα. Για καλύτερη οπτικοποίηση όμως επιλέχθηκε να εμφανίζεται ένα κενό plot με τους ίδιους άξονες και πλέγμα με τις επόμενες περιπτώσεις. Αυτό επιτεύχθηκε ορίζοντας ως συνθήκη το μήκος του διανύσματος των επιλογών να είναι μηδέν.

2) Ο χρήστης έχει μόνο μία επιλογή ενεργή: Εδώ είναι και η πιο απλή περίπτωση μιας και με βάση την επιλογή του χρήστη, το πρόγραμμα ζητάει την αντίστοιχη στήλη του πίνακα Profit και σχεδιάζει τη γραμμή που πρέπει, καθώς και το υπόμνημα.

3) Ο χρήστης έχει παραπάνω από μία επιλογές ενεργές: Στην περίπτωση αυτή θα πρέπει να εντοπιστεί το πλήθος των επιλογών του χρήστη, κάτι που γίνεται δυνατό μέσω του μήκους του διανύσματος επιλογών του. Στη συνέχεια τρέχει το plot όπως στην περίπτωση 2 για τη μία

επιλογή ενεργή για το πρώτο στοιχείο του διανύσματος. Μετά αρκεί μια επαναληπτική βημάτων (μήκους - 1) η οποία να περιέχει την εντολή `lines()`, ώστε να σχεδιάζονται οι επιπλέον γραμμές στο ίδιο γράφημα η μία πάνω στην άλλη. Το υπόμνημα ανανεώνεται με αντίστοιχο τρόπο.

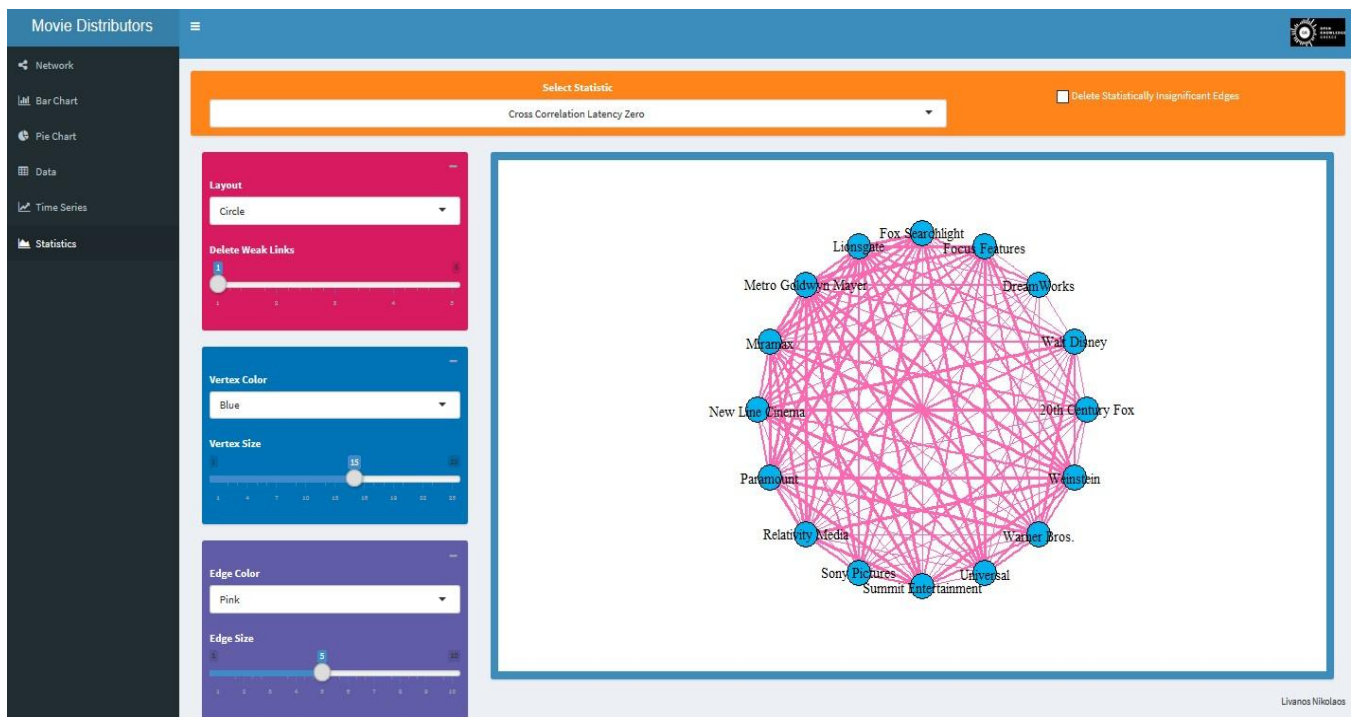
Οι άξονες ορίστηκαν με βάση την ελάχιστη και τη μέγιστη τιμή του πίνακα Profit και το πλέγμα περιέχει οριζόντιες γραμμές σε απόσταση 1000000000 και κάθετες ανά έτος.

5.5 Παρουσίαση των συσχετίσεων

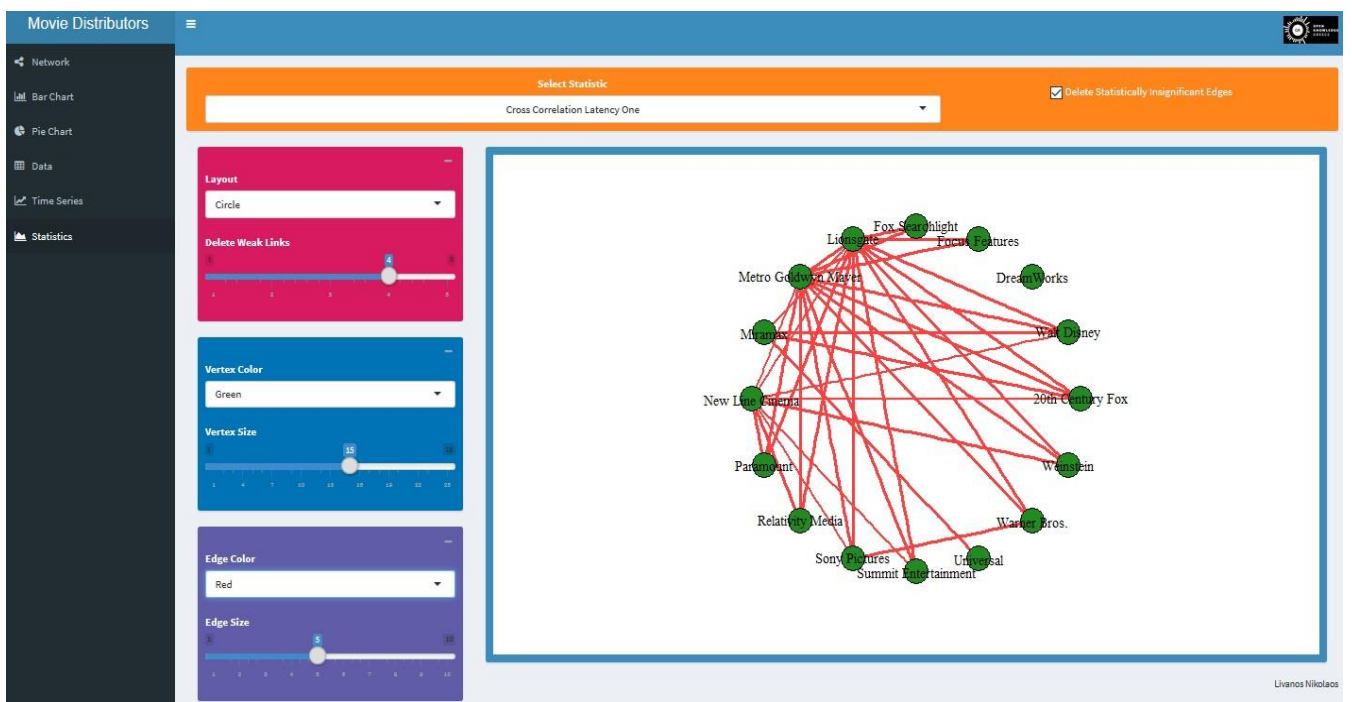
Τα αποτελέσματα από τον υπολογισμό των στατιστικών μέτρων έδωσαν ένα σύνολο πινάκων, 8 στο σύνολο. Οι 4 αφορούν τις τιμές άμεσα υπολογισμένες από τις αντίστοιχες εντολές, ενώ οι υπόλοιποι ακριβώς τους ίδιους πίνακες, αλλά με μηδενισμένα τα μη στατιστικά σημαντικά στοιχεία.

Χρησιμοποιώντας τους πίνακες αυτούς, σχεδιάστηκαν γράφοι όπως στην καρτέλα "Network" οι οποίοι παρουσιάζονται στην καρτέλα "Statistics" της εφαρμογής. Ο χρήστης αρχικά επιλέγει το στατιστικό μέτρο που θέλει να προβάλλει, και στη συνέχεια εάν θέλει να περιέχονται ή όχι η μη στατιστικά σημαντικές τιμές.

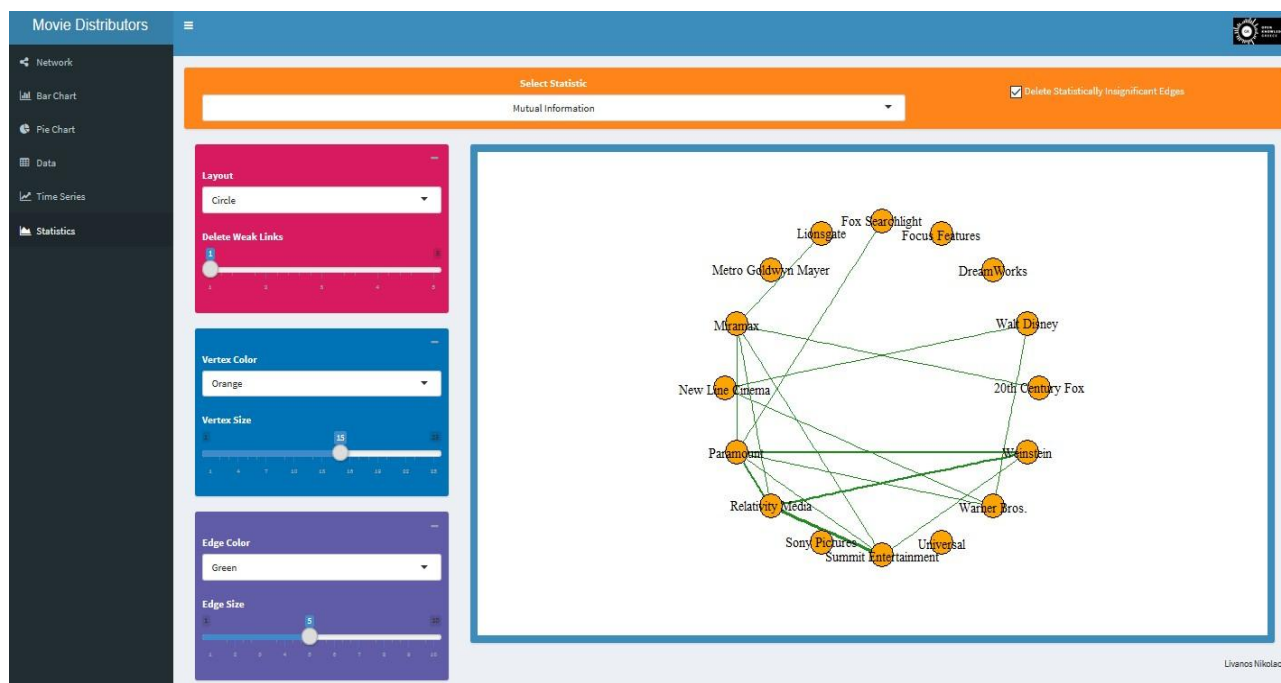
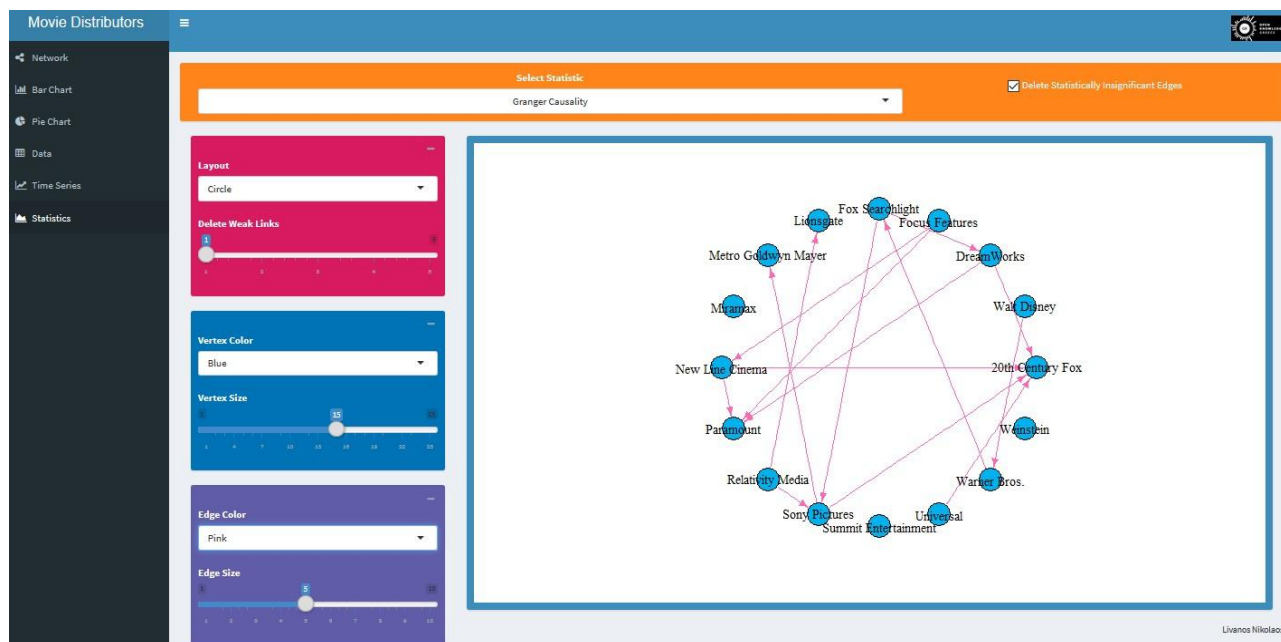
Επιπλέον του δίνεται η δυνατότητα να επεξεργαστεί το σχεδιασμένο δίκτυο με όλα τα widgets που υπάρχουν και στην καρτέλα "Network". Ειδικά η σταδιακή διαγραφή αδύναμων ακμών εδώ είναι αναγκαία, αφού ακόμα και στις ελάχιστες συσχετίσεις, σχεδιάζονται ακμές πολύ μικρού πάχους, που αν δε διαγραφούν, δυσκολεύουν το χρήστη να βγάλει τα συμπεράσματα που θέλει.



Εικόνα 36: Η καρτέλα Statistics με το Cross Correlation μηδενικής καθυστέρησης.



Εικόνα 37: Στατιστικά σημαντικές τιμές Cross Correlation με καθυστέρηση 1 στο 4ο επίπεδο διαγραφής ακμών.

Εικόνα 38: Στατιστικά σημαντικές τιμές *Mutual Information*.Εικόνα 39: Στατιστικά σημαντικές τιμές αιτιότητας *Granger*.

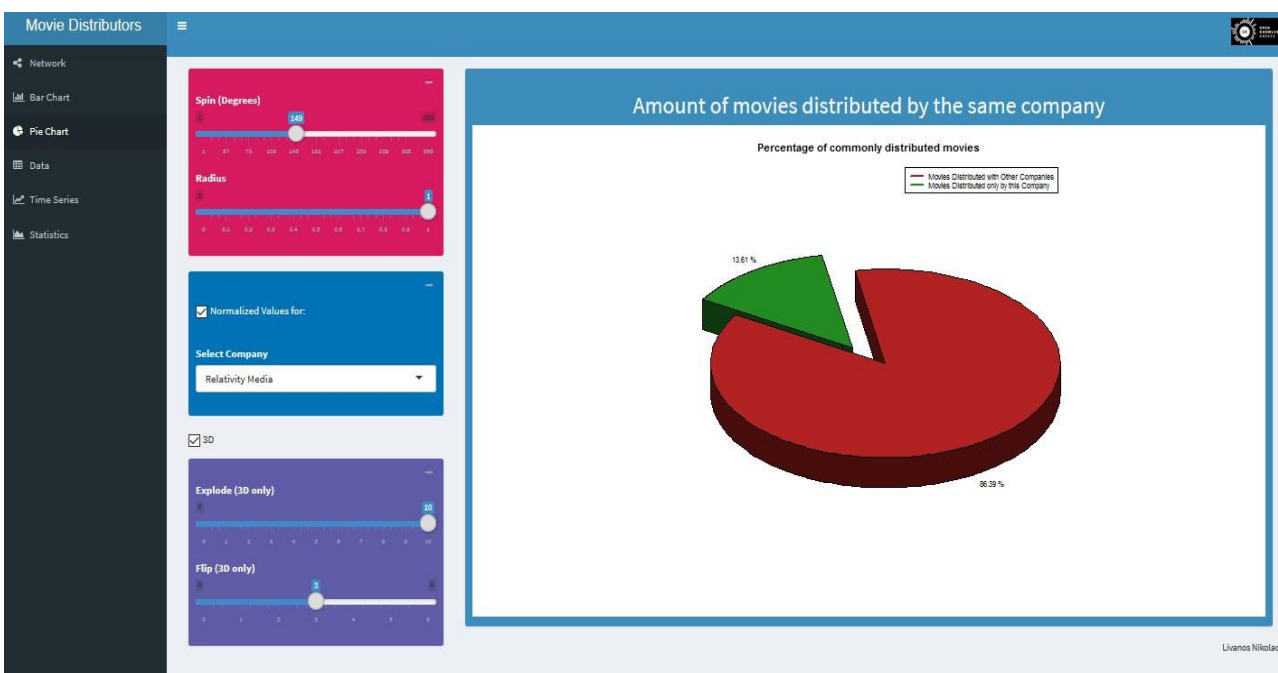
Συνεισφορά - Συμπεράσματα

Στην παρούσα εργασία έγινε προσπάθεια να αναδειχθούν οι δυνατότητές που προσφέρονται σε κάθε χρήστη του διαδικτύου από τα ανοιχτά διασυνδεδεμένα δεδομένα. Συγκεκριμένα για τη βάση δεδομένων της DBpedia η καινοτομία αυτή είναι σε αρκετά πρώιμο στάδιο, καθώς οι εγγραφές της Wikipedia όλα αυτά τα χρόνια επεξεργάζονταν από πολλούς διαφορετικούς χρήστες και κατά συνέπεια δεν έχουν συνοχή. Παρ'όλα αυτά υπάρχουν πολλά εργαλεία καθαρισμού στον Ιστό με τα οποία ο κάθε χρήστης μπορεί να δημιουργήσει μια αξιόλογη βάση δεδομένων και να τη μελετήσει.

Ιδανικά η εργασία αυτή θα μπορούσε να εξυπηρετήσει σαν οδηγός για να διορθωθεί το περιεχόμενο της DBpedia τουλάχιστον σε σημασιολογικό επίπεδο. Τα προβλήματα που συναντάει κάποιος που χρησιμοποιεί τέτοιου είδους δεδομένα έχουν καταγραφεί και χωριστεί σε κατηγορίες. Επομένως καθίσταται ευκολότερη και η επίλυση τους με αλγορίθμους που θα μπορούν να ελέγχουν και να εντοπίζουν τις εγγραφές με ασυνέχειες ίδιου τύπου με αυτές που περιγράφηκαν και στη συνέχεια να τις διορθώνουν αυτόματα. Ερχόμαστε έτσι ένα βήμα πιο κοντά στην περιγραφική λογική των διασυνδεδεμένων δεδομένων.

Το πρώτο στάδιο της μελέτης έδειξε ότι οι εταιρίες συνεργάζονται σε ένα πολύ καλό ποσοστό με άλλες για να διανείμουν τις ταινίες τους. Θα περίμενε κανείς ότι το πλήθος των συνεργασιών που είχε κάθε εταιρία θα είναι ανάλογο του πλήθους των συνολικών ταινιών στις οποίες αναφέρεται. Κάτι τέτοιο όμως δεν ισχύει. Γενικά το ποσοστό ταινιών που διανεμήθηκαν συνεργαστικά από παραπάνω από μία εταιρίες κυμαίνεται στο 30 - 60% για κάθε εταιρία με μερικές εξαιρέσεις. Η μεγαλύτερη εξαίρεση όμως αποτελεί η Relativity Media της οποία το συντριπτικό 86,39% των ταινιών της διανεμήθηκε συνεργατικά με άλλες εταιρίες. Επιπλέον διαθέτει τις ακμές με τα 2 μεγαλύτερα βάρη στο δίκτυο της καρτέλας Network, 64 με τη Universal και 36 με τη Sony Pictures, ενώ το όνομα της αναφέρεται σε λιγότερες ταινίες από κάθε άλλη εταιρία. Ακολουθεί το 34 που συνδέει την Paramount με την DreamWorks, μια γνωστή συνεργασία λόγω των πολλών ταινιών κινουμένων σχεδίων που διανέμουν μαζί.

Η εργασία αυτή αναφέρεται σε δεδομένα από το 2000 έως το 2014. Θα μπορούσε κάποιος να θεωρήσει τυχαίο το φαινόμενο της Relativity Media. Όμως, στις 30 Ιουλίου του 2015 η εταιρία κύρηξε πτώχευση λόγω αδυναμίας πληρωμών των δανείων της. Βέβαια, στις 18 Μαρτίου του 2016 κατάφερε να βγει από την κατάσταση χρεοκοπίας, χωρίς αυτό να αναιρεί το παραπάνω γεγονός. Η χρεοκοπία αυτή θα μπορούσε να ερμηνευτεί λόγω συνεχόμενων αποτυχιών στις τελευταίες ταινίες της. Αλλά θα μπορούσε επίσης να ερμηνευτεί ως αποτέλεσμα των πάρα πολλών συνεργασιών της με τις υπόλοιπες εταιρίες.



Εικόνα 40: Η περίπτωση της Relativity Media.

Τα αποτελέσματα της ανάλυσης των χρονοσειρών έδειξαν ότι υπάρχουν αρκετές γραμμικές συσχετίσεις εκ των οποίων πολλές είναι και στατιστικά σημαντικές. Αυτό ισχύει και για καθυστέρηση 0 και για 1 με διαφορετικές συνδέσεις σε κάθε περίπτωση. Αντίθετα ο υπολογισμός της αμοιβαίας πληροφορίας έδειξε ελάχιστες στατιστικά σημαντικές μη γραμμικές συσχετίσεις, χωρίς βέβαια αυτό να σημαίνει ότι δεν υπήρχαν καθόλου. Αντίστοιχα αποτελέσματα έδωσε και η αιτιότητα Granger για τη μονόδρομη συσχέτιση με αρκετές στατιστικά σημαντικές τιμές, αλλά μικρές σε τιμή, ώστε να θεωρηθεί ότι κάποια εταιρία επηρεάζει σε μεγάλο βαθμό κάποια άλλη σε καθυστέρηση 1.

Αξίζει να σημειωθεί ότι τα αρχικά δεδομένα περιείχαν το συνολικό budget και box office της κάθε ταινίας. Στις περιπτώσεις όμως που η ταινία κυκλοφόρησε από περισσότερες από μία εταιρίες δεν αναγράφεται το εκάστοτε μερίδιο εσόδων και εξόδων που διέθεσε ή έλαβε η καθεμία. Αυτό δημιουργεί ατέλειες στα οικονομικά δεδομένα και κατ'επέκταση στην ανάλυση των χρονοσειρών. Ιδανικά, θα υπολογιζόταν ο πίνακας χρονοσειρών για κάθε εταιρία με βάση μόνο τα δικά της μερίδια και όχι τα συνολικά έσοδα-έξοδα της κάθε ταινίας της. Δεδομένα όμως με τόσο λεπτομερείς πληροφορίες δεν είναι διαθέσιμα συνήθως ούτε για ανάγνωση, πόσο μάλλον διασυνδεδεμένα για μαζική εξόρυξη.

Η μεθοδολογία και τα αποτελέσματα της εργασίας αυτής παρουσιάστηκαν στο SEMANTiCS το Σεπτέμβριο του 2016 στη Λειψία.

Μελλοντική δουλειά - Επεκτάσεις

Η κατηγορία δεδομένων που επιλέχθηκε, δηλαδή ο συνδυασμός των ταινιών με τις εταιρίες διανομής τους είναι μόνο μία από τις πολλές που μπορεί ο καθένας να ασχοληθεί και να διεξάγει παρόμοια έρευνα. Αν θέλει να παραμείνει στην κατηγορία του κινηματογράφου θα μπορούσε να συνδυάσει τις συνεργασίες των εταιριών διανομής με τις οικονομικά δεδομένα και να εντοπίσει έτσι τις πιο κερδοφόρες από αυτές. Μπορεί έτσι μια αντίστοιχη εργασία να λειτουργήσει ως συμβουλευτική για τις εταιρίες αυτές.

Επιπλέον, εκτός της κατηγορίας του κινηματογράφου, ανάλογα δεδομένα μπορούν να αντληθούν και να επεξεργαστούν και για τη μουσική, τη λογοτεχνία, ή ακόμη και την τηλεόραση. Οι αντίστοιχες εγγραφές των εταιριών διανομής θα μπορούσαν να είναι οι δισκογραφικές εταιρίες, οι εκδοτικοί οίκοι ή οι τηλεοπτικοί σταθμοί αντίστοιχα.

Η εργασία αυτή μπορεί να δώσει μια κατεύθυνση σε επίδοξους προγραμματιστές ώστε να βελτιωθεί η ποιότητα δεδομένων που μπορεί κάποιος να αντλήσει από το "σύννεφο" των διασυνδεδεμένων δεδομένων. Σε όλη την τεχνολογία της DBpedia μπορούν να προστεθούν αλγόριθμοι που να ελέγχουν την ομοιομορφία των εγγραφών ως προς τις κατηγορίες σφαλμάτων που εντοπίστηκαν στην εργασία αυτή. Σε δεύτερο χρόνο μπορούν οι αλγόριθμοι αυτοί να διορθώνουν αυτόματα τέτοιου είδους σφάλματα, όπου αυτά εμφανίζονται.

Επιπλέον, μια καλή ιδέα θα ήταν να δημιουργηθεί ένα ενιαίο πακέτο εντολών για την R το οποίο θα ειδικεύεται σε τέτοιου είδους στατιστικές αναλύσεις, μιας και οι εντολές που χρησιμοποιήθηκαν βρέθηκαν από πολλά διαφορετικά. Έτσι μια ανάλυση αυτής της μορφής θα ήταν λιγότερο χρονοβόρα αλλά και τα αποτελέσματα της θα ήταν πιο διασταυρωμένα αφού θα προέρχονταν από την ίδια πηγή.

Η γλώσσα SPARQL προϋποθέτει την κατανόηση των RDF τριπλετών, αναζήτηση για τις διάφορες ιδιότητες στις αντίστοιχες οντολογίες και ακόμη και τότε, το συντακτικό της είναι πολύ διαφορετικό από αυτό των υπολοίπων γλωσσών προγραμματισμού. Αυτό την καθιστά δυσπρόσιτη στο μέσο χρήστη, με αποτέλεσμα τα ανοιχτά διασυνδεδεμένα δεδομένα στις διάφορες τοποθεσίες του Ιστού στα οποία δίνει πρόσβαση να μην είναι τελικά τόσο "ανοιχτά"

λόγω αυτού. Οι εφαρμογές που δημιουργούνται σε εργασίες όπως η παρούσα, προσφέρουν μόνο ένα κομμάτι της διαθέσιμης πληροφορίας στο χρήστη και φυσικά τον περιορίζουν στο κομμάτι αυτό. Το θετικό όμως με τη SPARQL είναι πως η δομή της είναι αρκετά "αυστηρή", γεγονός που θα μπορούσε να τυποποιηθεί και ενδεχομένως να συντάσσεται αυτόματα από τον υπολογιστή. Το όραμα λοιπόν είναι μια πλατφόρμα, στην οποία ο χρήστης με διάφορα widget να εμφανίζει στην οθόνη του κάθε ιδιότητα ολόκληρης της οντολογίας, να επιλέγει τις κατηγορίες που θέλει, και τα δεδομένα να έρχονται αυτόματα στον υπολογιστή του, καθαρισμένα από ασυνέχειες. Το ερώτημα SPARQL να συντάσσεται αυτόματα βάσει των επιλογών του χρήστη, και ο καθαρισμός να έχει γίνει με τον τρόπο που περιγράφηκε παραπάνω. Κάτι τέτοιο θα έδινε πρόσβαση στα δεδομένα αυτά σε μεγαλύτερο κομμάτι του πληθυσμού επιταχύνοντας και εμπλουτίζοντας την έρευνα.

Βιβλιογραφία

- [1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A crystallization point for the Web of Data. *Journal of Web Semantics*, 7(3):154-165, 2009.
- [2] A. S. Dadzie and M. Rowe. Approaches to visualising Linked Data: A survey, 2011.
- [3] S. Gao, G. V. Steeg, and A. Galstyan. Estimating mutual information by local gaussian approximation. *CoRR*, abs/1508.00536, 2015.
- [4] C. W. J. Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3):424-438, 1969.
- [5] H. Paulheim. Generating possible interpretations for statistics from linked open data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7295 LNCS, pages 560-574, 2012.
- [6] S. Sparviero. The business strategy of hollywood's most powerful distributors: An empirical analysis. *Observatorio*, 7(4):45-62, 2013.
- [7] M. Verlic. LODGrefine - LOD-enabled Open Refine in action. In *CEUR Workshop Proceedings*, volume 932, pages 31-37, 2012.
- [8] D. Wienand and H. Paulheim. The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings, chapter Detecting Incorrect Numerical Data in DBpedia, pages 504-518. Springer International Publishing, Cham, 2014.
- [9] P. M. Philippides, C. Bratsas, A. Veglis, E. Chondrokostas, D. Tsigkari, I. Antoniou. Creating and using sports linked data: Applications and analytics. *CEUR Workshop Proceedings*, 1481:38-41, September 2015.

Σύνδεσμοι

<https://cran.r-project.org/web/packages/bc3net/bc3net.pdf>

<https://cran.r-project.org/web/packages/c3net/c3net.pdf>

<https://cran.r-project.org/web/packages/circlize/circlize.pdf>

<https://cran.r-project.org/web/packages/DT/DT.pdf>

<https://cran.r-project.org/web/packages/entropy/entropy.pdf>

<https://cran.r-project.org/web/packages/foreign/foreign.pdf>

<https://cran.r-project.org/web/packages/igraph/igraph.pdf>

<https://cran.r-project.org/web/packages/infotheo/infotheo.pdf>

<https://cran.r-project.org/web/packages/MSBVAR/MSBVAR.pdf>

<https://cran.r-project.org/web/packages/plotrix/plotrix.pdf>

<https://cran.r-project.org/web/packages/shiny/shiny.pdf>

<https://cran.r-project.org/web/packages/shinydashboard/shinydashboard.pdf>

Ευρετήριο Εικόνων

<i>Εικόνα 1: Παράδειγμα σύνδεσης RDF δεδομένων από διαφορετικές βάσεις δεδομένων.</i>	24
<i>Εικόνα 2: "Υποκείμενο, Κατηγορούμενο, Αντικείμενο": Το μοντέλο των RDF δεδομένων.</i>	24
<i>Εικόνα 3: Παράδειγμα διασυνδεδεμένων δεδομένων με τις αντίστοιχες τριπλέτες καταχωρημένες.</i>	25
<i>Εικόνα 4: Πίνακας με τους κορυφαίους παραγωγούς ταινιών τα τελευταία 20 χρόνια.</i>	26
<i>Εικόνα 5: Παράδειγμα "έξυπνης" αναζήτησης στο themoviedb.org</i>	27
<i>Εικόνα 6: Παράδειγμα συνδέσμου ταινίας στο IMDb.</i>	28
<i>Εικόνα 7: Η αρχική σελίδα του endpoint του LinkedMDB.</i>	30
<i>Εικόνα 8: Παράδειγμα από infobox σελίδας της Wikipedia</i>	31
<i>Εικόνα 9: Παράδειγμα σελίδας της DBpedia</i>	32
<i>Εικόνα 10: Η αρχική σελίδα του endpoint της DBpedia</i>	36
<i>Εικόνα 11: Αποτελέσματα του παραδείγματος SPARQL</i>	38
<i>Εικόνα 12: Η αρχική σελίδα του Open Refine.</i>	39
<i>Εικόνα 13: Δεδομένα σε μορφή υπολογιστικού φύλλου στο Open Refine.</i>	39
<i>Εικόνα 14: Η επιλογή Text facet</i>	41
<i>Εικόνα 15: Η Variable View του SPSS</i>	43
<i>Εικόνα 16: Το περιβάλλον της R Studio.</i>	45
<i>Εικόνα 17: Τα βασικά widgets της Shiny</i>	47
<i>Εικόνα 18: Κενή εφαρμογή με δύο καρτέλες της shinydashboard.</i>	48
<i>Εικόνα 19: Παράδειγμα εφαρμογής με 4 καρτέλες, 2 γραμμές και 5 boxes.</i>	50
<i>Εικόνα 20: Παράδειγμα γράφου, σχεδιασμένου μέσω της igraph.</i>	53
<i>Εικόνα 21: Κομμάτι αποτελεσμάτων στο endpoint της DBpedia</i>	61
<i>Εικόνα 22: Αρχική μορφή των δεδομένων.</i>	63
<i>Εικόνα 23: Διαγραφή και ταξινόμηση των στηλών.</i>	64
<i>Εικόνα 24: Πρώτο στάδιο καθαρισμού.</i>	64
<i>Εικόνα 25: Δεύτερο στάδιο καθαρισμού.</i>	66
<i>Εικόνα 26: Τρίτο στάδιο καθαρισμού.</i>	68
<i>Εικόνα 27: Τελική μορφή αρχείου της βάσης δεδομένων.</i>	70
<i>Εικόνα 28: Ο πίνακας Common.</i>	74
<i>Εικόνα 29: Ο πίνακας Profit.</i>	77
<i>Εικόνα 30: Η καρτέλα Data της τελικής εφαρμογής.</i>	83
<i>Εικόνα 31: Η καρτέλα Network.</i>	84
<i>Εικόνα 32: Η καρτέλα Bar Chart.</i>	85
<i>Εικόνα 33: Η καρτέλα Pie Chart με κανονικοποιημένες τιμές.</i>	86
<i>Εικόνα 34: Η καρτέλα Pie Chart με ενεργή την επιλογή 3D.</i>	87
<i>Εικόνα 35: Η καρτέλα Time Series.</i>	88

<i>Εικόνα 36: Η καρτέλα Statistics με το Cross Correlation μηδενικής καθυστέρησης.</i>	<i>90</i>
<i>Εικόνα 37: Στατιστικά σημαντικές τιμές Cross Correlation με καθυστέρηση 1 στο 4ο επίπεδο διαγραφής ακμών.</i>	<i>90</i>
<i>Εικόνα 38: Στατιστικά σημαντικές τιμές Mutual Information.</i>	<i>91</i>
<i>Εικόνα 39: Στατιστικά σημαντικές τιμές αιτιότητας Granger.</i>	<i>91</i>
<i>Εικόνα 40: Η περίπτωση της Relativity Media.</i>	<i>93</i>