HY360: Δεύτερο Μέρος Συνθετικής Εργασίας

Τμήμα 2 | Εξερεύνηση του GitHub Dataset μέσω του Colaboratory (30 μονάδες)

Υποδείξεις - διαβάστε τις πολύ προσεκτικά! - :

- Σιγουρευτείτε ότι διαβάσατε καλά τις οδηγίες σε κάθε κελί και κατανοήσατε τι υλοποιεί πριν το εκτελέσετε.
- Να θυμάστε ότι έχετε τη δυνατότητα να μεταφορτώνετε το αρχικό "σημειωματάριο" όποτε το χρειάζεστε.
- Μπορείτε να δημιουργείτε νέα κελιά για να τα χρησιμοποιείτε σε ελέγχους, εκσφαλμάτωση, εξερεύνηση κλπ. Μάλιστα προτείνουμε να το κάνετε! **Βεβαιωθείτε εντούτοις ότι η τελική απάντηση σε κάθε ερώτηση βρίσκεται στο *δικό της κελί* και *προσδιορίζεται ρητά***.
- Το Colaboratory δεν σας ειδοποιεί για τα bytes που θα καταναλώσει η εκτέλεση των SQL ερωτημάτων σας. **Σιγουρευτείτε ότι ελέγχετε την κατανάλωση μέσω της διεπαφής (UI) του BigQuery πριν εκτελέσετε τα ερωτήματά σας στο Colaboratory!**
- Ακολουθείστε τις οδηγίες υποβολής.

Μέλη της Ομάδας Εργασίας:

Παραθέστε τα ονοματεπώνυμα και τους ΑΜ των μελών της ομάδας στην ακόλουθη λίστα:

Μάριος Κωνσταντίνος Κωνσταντάκης, 3219.

Ιωακείμ Ορφέας Νικολουδάκης, 3682.

Ρυθμίσεις για το BigQuery και τις σχετικές εξαρτήσεις

Εκτελέστε τα δύο ακόλουθα κελιά (shift + enter) προκειμένου να πιστοποιήσετε την εργασία σας και να φορτώσετε τις απαιτούμενες βιβλιοθήκες.

Προσέξτε ότι θα χρειαστεί να συμπληρώσετε τη μεταβλητή `project_id` στο πρώτο κελί με το Google Cloud project ID που έχετε δημιουργήσει για τις ανάγκες της εργασίας σας. Για να δείτε το project ID μεταβείτε στη σελίδα https://console.cloud.google.com/cloud-resource-manager.

```
# Εκτελέστε αυτό το κελί προκειμένου να πιστοποιήσετε την εργασία σας στο BigQuery.
from google.colab import auth
auth.authenticate_user()
project_id = 'groovy-analyst-227015'
```

```
# Βιβλιοθήκες που θα χρειαστείτε
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline
plt.style.use('seaborn-whitegrid')
```

Σχετικά

Το BigQuery διαθέτει ένα τεράστιο σύνολο δεδομένων (dataset) με αρχεία και στατιστικά από το GitHub που περιέχουν πληροφορία σχετική με αποθετήρια (repositories), δεσμεύσεις (commits) και περιεχόμενα αρχείων. Στο τμήμα αυτό της εργασίας σας θα εντρυφήσουμε σε αυτό το σύνολο δεδομένων. Μην ανησυχείτε εάν δεν είσαστε εξοικειωμένοι με τα Gits και το GitHub -- θα εξηγηθεί επαρκώς ότι χρειάζεστε για να ολοκληρώσετε αυτό το τμήμα εργασίας.

Σημειώσεις

Το σύνολο δεδομένων του GitHub που έχει αποθηκευτεί στο BigQuery είναι τεράστιο. *Μια και μόνη επερώτηση μόνο στον πίνακα "contents" (που έχει μέγεθος 2.16TB!) μπορεί να καταναλώσει τη δωρεάν μηνιαία χρήση του 1TB που παρέχεται σε όλους τους χρήστες και ακόμη περισσότερο.*

Για να γίνει περισσότερο διαχειρίσιμο αυτό το τμήμα της εργασίας έχουμε φτιάξει ένα υποσύνολο των πρωτότυπων δεδομένων. Διατηρήσαμε σχεδόν ολόκληρη την πληροφορία από τους πρωτότυπους πίνακες, αλλά επιλέξαμε να περιοριστούμε στα 500,000 αποθετήρια (repositories) του GitHub στα οποία έγιναν οι περισσότερες προσπελάσεις ανάμεσα στον Ιανουάριο του 2016 και τον Οκτώβριο του 2018. Οι πίνακες με τους οποίους θα εργαστούμε βρίσκονται εδώ.

Εντούτοις, για να μπορέσετε να τους προσπελάσετε και να θέσετε ερωτήματα, απατείται να γίνει εκ μέρους σας έγκαιρα μια ενέργεια: θα πρέπει να κατανοήσετε τι χρειάζεται να ρυθμιστεί και το αργότερο μέχρι 11/1/2019 να έχετε επικοινωνήσει στο email: tsatsaki@csd.uoc.gr ζητώντας να γίνει η κατάλληλη ρύθμιση. Επαναλαμβάνουμε: θα πρέπει να κατανοήσετε σε τι αφορά η ρύθμιση και να υποβάλλετε τα απαραίτητα στοιχεία για να γίνει.

Αφού εξασφαλίσετε τη δυνατότητα προσπέλασης, περιηγηθείτε στα σχήματα των πινάκων για να κατανοήσετε τα δεδομένα που περιέχουν. Να σημειωθεί ότι και σε αυτό το σύνολο δεδομένων υπάρχουν πίνακες πολύ μεγάλοι (ο πίνακας contents έχει μέγεθος μεγαλύτερο από 500GB), γι' αυτό θα πρέπει να προσέχετε πώς τους χρησιμοποιείτε. Ελέγξτε τη χρέωση για τα ερωτήματα που θα θέσετε, **πριν τα θέσετε,** στη διεπαφή του BigQuery.

Ένας πολύ σύντομος οδηγός στο GitHub

Εάν δεν είσαστε εξοικειωμένοι με τα Git και το GitHub, ακολουθούν αδρές επεξηγήσεις των βασικών εννοιών που πλαισιώνουν αυτό το τμήμα της εργασίας:

- *GitHub*: Το GitHub είναι πάροχος υπηρεσίας ελέγχου εκδόσεων αρχείων, που επιτρέπει (μεταξύ άλλων) τη συνεργατική υλοποίηση και παρακολούθηση πηγαίου κώδικα, με αρκετά αποτελεσματικό τρόπο.
- *commit*: Ως commit (αναθεώρηση) θεωρείται η αλλαγή που εφαρμόζεται σε ένα σύνολο αρχείων. Έτσι, εάν κάνετε αλλαγές σε ένα σύνολο αρχείων που είναι σε μια κατάσταση Α, μετά από *commit* το σύνολο θα βρίσκεται σε μια νέα κατάσταση Β. Ένα commit χαρακτηρίζεται από ένα "μίγμα" (*hash ή SHA*) πληροφοριών που σχετίζονται με την αναθεώρηση (τον συντάκτη του commit, ποιος στην πραγματικότητα έκανε [εφάρμοσε] τις αλλαγές στα αρχεία, σε τι αφορούν οι αλλαγές, κλπ.)
- *parent commit*: Το commit που έγινε πριν από το τρέχον commit.
- *repo*: Αποθετήριο ονομάζεται μια αφηρημένη συλλογή (κάτι σαν φάκελος) αρχείων μαζί με ένα ιστορικό αναθεωρήσεων (commits) αυτών. Εάν το GitHub username σας είναι "foo" και φτιάξετε ένα αποθετήριο (repo) με όνομα "data-rocks", το απόλυτο όνομά του θα είναι "foo/data-rocks". Μπορείτε να σκέφτεστε την ιστορία των αποθετηρίων σε σχέση με τις αναθεωρήσεις τους (commits). Πχ το "foo/data-rocks" μπορεί να πέρασε από ένα σύνολο "καταστάσεων" A->B->C->D, στο οποίο κάθε αλλαγή κατάστασης (A->B, B->C, C->D) σχετίζεται με μια αναθεώρηση (commit).
- *branch*: Προκειμένου να παρακολουθήσουν διαφορετικά ιστορικά αναθεωρήσεων, τα αποθετήρια του GitHub μπορεί να έχουν διακλαδώσεις. Η 'κύρια' διακλάδωση ενός αποθετήριου ονομάζεται 'master' branch. Έστω ότι στο "foo/data-rocks" έχουμε την ιστορία αναθεωρήσεων A->B->C->D στην κύρια διακλάδωση. Εάν κάποιος αποφασίσει να προσθέσει ένα νέο χαρακτηριστικό στο "foo/data-rocks", μπορεί να δημιουργήσει μια διακλάδωση με όνομα"cool-new-feature" που ξεχωρίζει από την κύρια διακλάδωση. Όλος ο κώδικας της κύριας διακλάδωσης θα βρίσκεται στην καινούργια, αλλά ο κώδικας που θα προστεθεί στην καινούργια δεν θα υπάρχει στην κύρια διακλάδωση (από την οποία προέκυψε η καινούργια διακλάδωση).
- *ref*: Για το σκοπό αυτού του τμήματος της εργασίας, μπορείτε να θεωρείτε το πεδίο 'ref' του πίνακα "αρχείων" ως αυτό που αναφέρεται στη διακλάδωση στην οποία "κατοικεί" το αρχείο σ' ένα αποθετήριο σε μια δεδομένη στιγμή.

Για το τμήμα αυτό της εργασίας δεν χρειάζεται να γνωρίζετε με λεπτομέρεια τα ακόλουθα:

- Δέντρα αναθεωρήσεων (commit trees)
- Το γνώρισμα κωδικοποίησης (encoding attribute) του πίνακα αναθεωρήσεων

Για περισσότερες πληροφορίες μπορείτε να δείτε εδώ και εδώ.

Ενότητα 1 | Γνωριμία με τα δεδομένα του GitHub

Κατανόηση των πινάκων του GitHub

Γνωρίζουμε ότι τα διαγράμματα οντοτήτων-συσχετίσεων είναι μία αναπαράσταση της δομής μιας ΒΔ (συνόλου πινάκων) με μη τεχνικό τρόπο και με όλη την απαραίτητη πληροφορία για τη ΒΔ. Όπως θα φανεί, με τα διαγράμματα οντοτήτων-συσχετίσεων θα εξοικειωθούμε με τους πίνακες του συνόλου δεδομένων GitHub, πριν ακόμη αναλύσουμε τα δεδομένα τους.

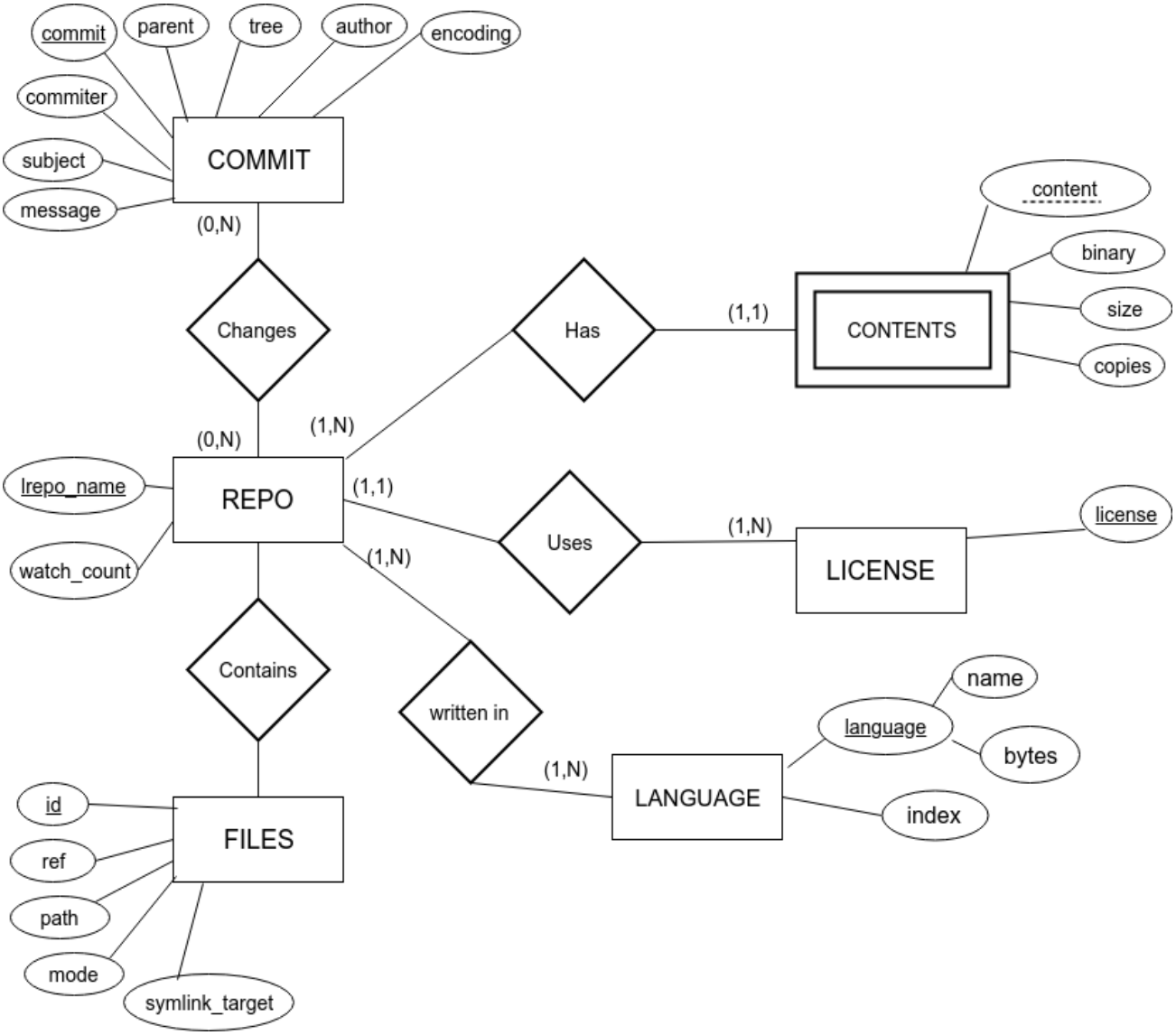
Ερώτηση 1: Πίνακες CS360 GitHub --> Διάγραμμα οντοτήτων-συσχετίσεων (4 μονάδες)

Δημιουργήστε ένα διάγραμμα οντοτήτων-συσχετίσεων για τα δεδομένα που περιέχονται στους πίνακες `cs360nt:project_part_2_2` που βρίσκονται εδώ. Αγνοήστε τον πίνακα `github_repo_readme_contents_cs360` (θα τον χρησιμοποιήσετε σε επόμενα ερωτήματα).

Σημειώσεις

- Είναι πιθανό να μην είναι δυνατή η απευθείας "μετάφραση" των πινάκων του CS360 GitHub Repo σε διάγραμμα οντοτήτων-συσχετίσεων με τον τρόπο που γνωρίζετε από το μάθημα. Σημαντικό μέρος αυτής της ερώτησης είναι η ανάλυση των πινάκων, η σκέψη και ο προσδιορισμός των σχέσεων μεταξύ των αντικειμένων που περιέχουν και η δημιουργία εν τέλει ενός εύλογου διαγράμματος οντοτήτων-συσχετίσεων βασισμένου στην ανάλυση που θα κάνετε.
- Θεωρήστε τα γνωρίσματα "author" και "committer" που έχουν τύπο εγγραφής (record) ως μοναδιαία (με απλό τύπο). Είναι σημαντικό ότι δε χρειάζεται να συμπεριλάβετε τα `committer.name`, `committer.email`, κλπ στα διαγράμματά σας. Να σημειωθεί ότι το γνώρισμα "language" έχει τύπο *array*, γεγονός που πρέπει να ληφθεί υπόψη στη σχεδίαση των διαγραμμάτων σας.
- Τα διαγράμματά σας πρέπει να είναι αρχεία εικόνας που θα σχεδιάσετε με όποιον τρόπο θέλετε (κατάλληλο λογισμικό ή με το χέρι), **αρκεί να είναι ευανάγνωστα**. Θα τα συμπεριλάβετε στο σημειωματάριό σας ως εξής:

    1. Προσθέστε το αρχείο εικόνας στο Google Drive σας
    2. Δημιουργήστε ένα κοινόχρηστο URL για το αρχείο σας, και σημειώστε το πεδίο "ID" από το URL που θα δημιουργηθεί Το URL θα έχει τη μορφή "`https://drive.google.com/open?id=<some ID>`"
    3. Προσθέστε την ακόλουθη επισήμανση (markup) στο κατάλληλο κελί του σημειωματαρίου σας ! `[]` `(https://drive.google.com/uc?export=view&id=<your image ID>)`
    4. Εκτελέστε (run) τον κώδικα στο κελί σας.



Ερώτηση 2: Εξηγήστε το διάγραμμα οντοτήτων-συσχετίσεων που φτιάξατε (2 μονάδες)

Σε μια μικρή παράγραφο εξηγήστε το διάγραμμά σας. Θα πρέπει να καλύψετε τουλάχιστον τα ακόλουθα:

- ποιες είναι οι οντότητες,
- ποιες οι μεταξύ τους σχέσεις (αναφέρετε εάν πρόκειται για 1-N, N-1, κλπ.),
- ποια είναι τα κλειδιά σε καθεμιά οντότητα.

Πρέπει επίσης να εξηγήσετε σύντομα με ποιο τρόπο καθορίσατε τη συνολική δομή του διαγράμματός σας.

Οντότητες: Commit, Repo, Contents(ασθενής οντότητα της οντότητας Repo), Files, License, Language

Σχέσεις:

```
Commit Changes Repo (N-N)
Repo Has Contents (N-1)
Repo Contains Files (N-1)
Repo written_in Language (N-N)
Repo uses License (1-N)
```

Κλειδιά: Commit : κλειδί = commit Repo: κλειδί = lrepo_name Contents: κλειδί = content License: κλειδί = license language: κλειδί = language

---

Ερώτηση 3: Μεταφράστε το διάγραμμά σας στο αντίστοιχο σχεσιακό σχήμα (3 μονάδες)

Δώστε το σχεσιακό σχήμα που αντιστοιχεί στο διάγραμμα που σχεδιάσατε στην προηγούμενη ερώτηση. Αυτό, θα πρέπει να διαφέρει από το σχήμα σύμφωνα με το οποίο φτιάχτηκαν οι πίνακες του συνόλου δεδομένων CS360 GitHub Repo.

Σιγουρευτείτε ότι έχετε καθορίσει στο σχήμα σας:

1. το ***όνομα*** κάθε γνωρίσματος (μην αναφερθείτε σε τύπους),
2. το ***κλειδί κάθε πίνακα***,
3. ***ξένα κλειδιά σε κάθε πίνακα*** και σε ποιους πίνακες αναφέρονται.

Commit

commit(κλειδί)	tree	parent	author	commiter	encoding	subject	message	lrepo_name(ξένο κλειδί)

Repo

lrepo_name(κλειδί)	watch_count	license(ξένο κλειδί)

Contents

content(κλειδί)	size	binary	copies	lrepo_name(ξένο κλειδί)

Files

id(κλειδί)	ref	path	mode	symlink_target	lrepo_name(ξένο κλειδί)

License

license(κλειδί)

Language

language(κλειδί)

Repo languages

lrepo_name	language

Ερώτηση 4: Ανάλυση (2 points)

Έχετε πλέον στη διάθεσή σας δύο σχήματα: αυτό που φτιάξατε στην ερώτηση 3 και αυτό που είχαν οι πίνακες όπως σας τους δώσαμε.

Σε μια και μόνη παράγραφο (μέχρι 100 λέξεις), συγκρίνετέ τα. Ποιο θεωρείτε καλύτερο;

Δεν υπάρχει μοναδική σωστή απάντηση. Τα σχήματα των ΒΔ θα πρέπει να καλύπτουν επαρκώς και τις εφαρμογές οι οποίες θα χρησιμοποιήσουν τις ΒΔ.

Εισάγετε εδώ τη συγκριτική σας ανάλυση

Ενότητα 2 | Οπτικοποίηση του Git!

Πριν ξεκινήσετε ...

Τώρα που έχετε κατανοήσει το σύνολο δεδομένων με το οποίο θα ασχοληθείτε, θα συνεχίσετε με την ανάλυση ορισμένων από τις ιδιότητές του. Για τις απαιτούμενες οπτικοποιήσεις μπορείτε να χρησιμοποιήσετε οποιαδήποτε βιβλιοθήκη γραφικών αναπαραστάσεων θέλετε. Προτείνουμε κάποια από τις:

- seaborn (https://seaborn.pydata.org/tutorial.html)
- matplotlib (https://matplotlib.org/3.0.0/tutorials/index.html)
- altair (https://altair-viz.github.io/)
- pandas (https://pandas.pydata.org/pandas-docs/stable/visualization.html)

- **σημειώστε ότι**: μπορείτε, εάν θέλετε, να σχεδιάζετε μέσα από ένα Pandas DataFrame .

Χρήση του BigQuery στο Collab

Τα σημειωματάρια στο Jupyter (στα οποία βασίζονται τα σημειωματάρια του Collab) χρησιμοποιούν τη ιδέα της "μαγείας". Εάν γράψετε την ακόλουθη γραμμή στην κορυφή ενός κελιού με 'Κώδικα' :

```
%%bigquery --project $project_id variable # this is the key line
SELECT ....
FROM ...
```

το "%%" μετατρέπει το κελί σε κελί SQL. Ο πίνακας που παράγεται από το ερώτημα αποθηκεύεται στη μεταβλητή `variable`. Στη συνέχεια μπορείτε να χρησιμοποιήσετε τη μεταβλητή `variable` στη βιβλιοθήκη οπτικοποίησης που θα χρησιμοποιήσετε για να δημιουργήσετε γραφικές αναπαραστάσεις!

Εκτελέστε τα δύο ακόλουθα κελιά για να πάρετε μια ιδέα του τι γίνεται στην πράξη.

```
%%bigquery --project groovy-analyst-227015 example

SELECT lrepo_name, watch_count
FROM `cs360nt.project_part_2_2.github_repos_cs360`
ORDER BY watch_count DESC
LIMIT 10;
```

	lrepo_name	watch_count
0	freecodecamp/freecodecamp	291503
1	vuejs/vue	119634
2	tensorflow/tensorflow	107721
3	facebook/react	92644
4	sindresorhus/awesome	73781
5	getify/you-dont-know-js	71632
6	kamranahmedse/developer-roadmap	59674
7	microsoft/vscode	57390
8	airbnb/javascript	55436
9	twbs/bootstrap	52244

```
example.head()
```

	lrepo_name	watch_count
0	freecodecamp/freecodecamp	291503
1	vuejs/vue	119634
2	tensorflow/tensorflow	107721
3	facebook/react	92644
4	sindresorhus/awesome	73781

Ερώτηση 5: Κατανομές τιμών για διάφορα πεδία (γνωρίσματα) (9 μονάδες)

Ας βρέξουμε τα πόδια μας στα δεδομένα του συνόλου που μελετούμε δημιουργώντας τις ακόλουθες γραφικές παραστάσεις:

1. Κατανομή αδειών (licences) στα διάφορα αποθετήρια (repos)

2. Κατανομή γλωσσών (languages) στα διάφορα αποθετήρια (repos)
3. Κατανομή μεγέθους αρχείων (file sizes) στα διάφορα αποθετήρια (repos)
4. Κατανομή πλήθους αρχείων (files) που περιλαμβάνονται στα διάφορα αποθετήρια (repos)
5. Αριθμός των αναθεωρήσεων (commits) κατά συντάκτη (author) και αναθεωρητή (committer) στα διάφορα αποθετήρια (repos)

Λάβετε υπόψιν ότι δεν θα πάρετε όλες τις μονάδες εάν τα διαγράμματά σας δεν είναι καλά φτιαγμένα (πχ δυσανάγνωστα).

Συμβουλές

- Ορισμένα διαγράμματα θα χρειαστεί να έχουν τουλάχιστον ένα άξονά τους σε λογαριθμική κλίμακα (log-scaled) για να είναι ευανάγνωστα
- Για δημιουργία ευανάγνωστων διαγραμμάτων μπορείτε να χρησιμοποιήσετε pandas.DataFrame.sample. Δείγμα μεγέθους μεταξύ 1,000 και 10,000 θα δώσει περισσότερο ευανάγνωστα διαγράμματα.

Να θυμάστε:

- Να είσαστε προσεκτικοί με τα ερωτήματά σας! Μην εκτελείτε SELECT * τυφλά σε κάποιο πίνακα στο παρόν σημειωματάριο του Colaboratory, καθώς δεν θα λάβετε προειδοποίηση του μεγέθους των δεδομένων που θα καταναλωθούν για το ερώτημά σας. Δοκιμάστε πρώτα το ερώτημά σας στο BigQuery UI καθώς εκεί έχετε τις απαιτούμενες προειδοποιήσεις -- ακόμη καλύτερα βάλτε και όρια στα ερωτήματά σας με βάση όσα έχουμε ήδη πει.
- Μην ξεχνάτε να χρησιμοποιείτε το υποσύνολο δεδομένων cs360nt:project_part_2_2 που βρίσκονται εδώ.

a) Κατανομή αδειών (1 μονάδα)

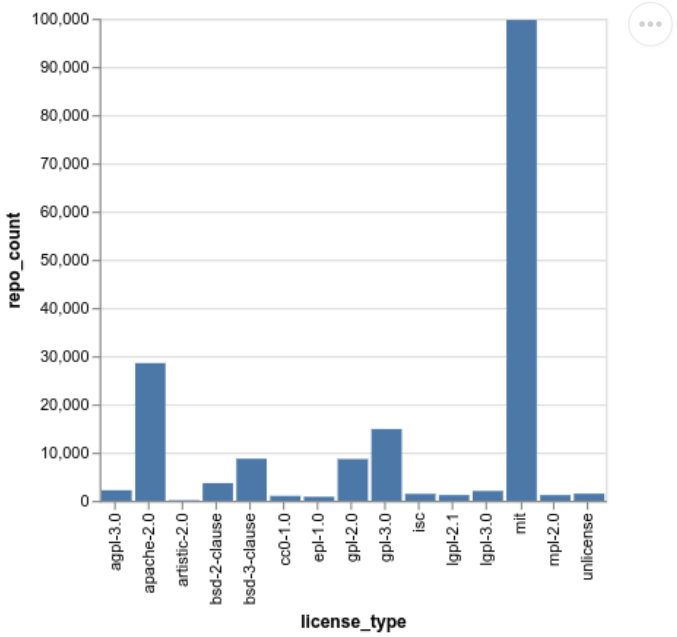
(x-άξονας: είδος άδειας (license type), y-άξονας: # αποθετηρίων (repos) που περιέχουν αυτή την άδεια)

```
%%bigquery --project groovy-analyst-227015 q5a

SELECT license AS license_type, COUNT(lrepo_name) AS repo_count FROM `groovy-analyst-227015.project_part
GROUP BY license_type
ORDER BY repo_count

# YOUR QUERY HERE
```

	license_type	repo_count
0	artistic-2.0	153
1	epl-1.0	852
2	cc0-1.0	1030
3	mpl-2.0	1197
4	lgpl-2.1	1200
5	isc	1461
6	unlicense	1498
7	lgpl-3.0	2057
8	agpl-3.0	2204
9	bsd-2-clause	3692
10	gpl-2.0	8673
11	bsd-3-clause	8748
12	gpl-3.0	14906
13	apache-2.0	28578
14	mit	99730

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
import numpy as np
import altair as alt

alt.Chart(q5a).mark_bar().encode(
alt.X("license_type"),
y = 'repo_count')
```



b) Κατανομή γλωσσών (1 μονάδα)

(x-άξονας: γλώσσα προγραμματισμού (programming language), y-άξονας: # αποθετηρίων (repos) που περιέχουν τουλάχιστον ένα αρχείο σε αυτή τη γλώσσα)

Για να είναι το γράφημα ευανάγνωστο, κρατήστε τις 20 επικρατέστερες γλώσσες.

Συμβουλή: https://cloud.google.com/bigquery/docs/reference/standard-sql/arrays

```
%%bigquery --project groovy-analyst-227015 q5b
#LIMIT 20
# YOUR QUERY HERE


# YOUR PLOT CODE HERE
```

c) Κατανομή μεγέθους αρχείων (1 μονάδα)

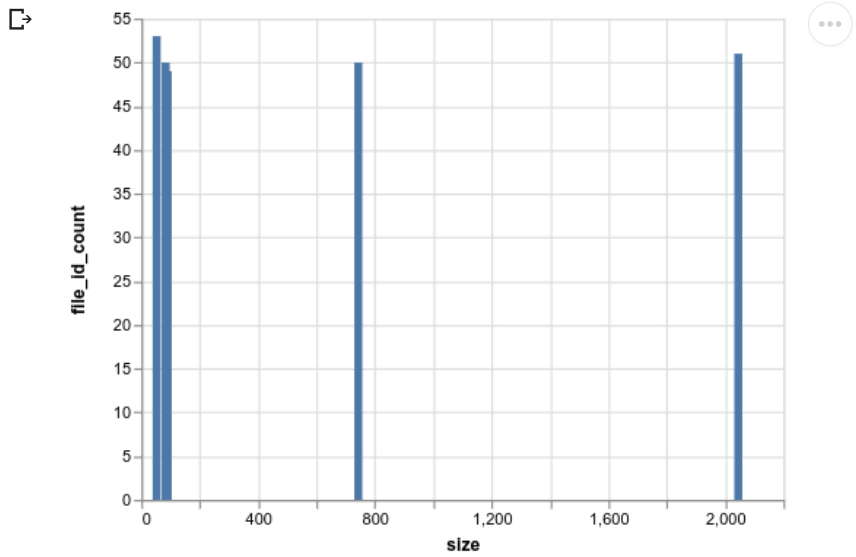
(x-άξονας: μέγεθος αρχείου, y-άξονας: # αρχείων με αυτό το μέγεθος)

```
%%bigquery --project groovy-analyst-227015 q5c
SELECT DISTINCT size,COUNT(DISTINCT id) AS file_id_count FROM `groovy-analyst-227015.project_part_2_2.g
GROUP BY size
ORDER BY file_id_count DESC

#most probably it's that
# YOUR QUERY HERE
```

	size	file_id_count
0	52	53
1	2045	51
2	83	50
3	743	50
4	90	49
5	64	49
6	67	49
7	85	49
8	1724	48
9	66	48
10	55	47
11	1398	47
12	1344	47
13	1224	47
14	1731	47
15	72	47
16	1096	46
17	1220	46
18	68	46
19	78	46
20	968	46
21	54	46
22	95	46
23	1754	46
24	158	46
25	102	45
26	1607	45
27	1563	45
28	805	45
29	976	45
...	...	...
23706	142169	1
23707	33739	1
23708	17095	1
23709	12272	1
23710	19704	1
23711	266614	1
23712	7933	1

23713	1002484	1
23714	21589	1
23715	73170	1
23716	27672	1
23717	11423	1
23718	140788	1
23719	18904	1
23720	55634	1
23721	49514	1
23722	15058	1
23723	14830	1
23724	22801	1
23725	13432	1
23726	86444	1
23727	27670	1
23728	43037	1

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt

alt.Chart(q5c.head()).mark_bar().encode(
alt.X("size:Q"),
y = 'file_id_count:Q')
```



d) Κατανομή αρχείων που σχετίζονται με ένα αποθετήριο (repo) (1 μονάδα)

(x-άξονας: # αρχείων που σχετίζονται με ένα αποθετήριο (repo) , y-άξονας: # αποθετηρίων (repos) που σχετίζονται με τέτοιο πλήθος αρχείων)

```
%%bigquery --project groovy-analyst-227015 q5d


SELECT DISTINCT COUNT(DISTINCT id) AS file_counter, COUNT(DISTINCT lrepo_name) AS repo_counter
```

```
FROM `groovy-analyst-227015.project_part_2_2.github_repo_files_cs360`

# YOUR QUERY HERE
```

	file_counter	repo_counter
0	36843674	175930

```
# YOUR PLOT CODE HERE
```

e) Πλήθος αναθεωρήσεων (commits) κατά συντάκτη (author) και αναθεωρητή (committer) (3 μονάδες)

(x-άξονας: # commits, y-άξονας: # authors/committers με τόσα commits)

Σημείωση: στο εν λόγω διάγραμμα, σχεδιάστε τις καμπύλες για τους συντάκτες (authors) και τους αναθεωρητές (committers) δίπλα - δίπλα για σύγκριση.

```
%%bigquery --project groovy-analyst-227015 q5e_authors
SELECT DISTINCT COUNT(commit) AS commit_count,author.name
FROM `groovy-analyst-227015.project_part_2_2.github_repo_commits_cs360`
GROUP BY author.name
ORDER BY commit_count DESC


# YOUR QUERY HERE
```

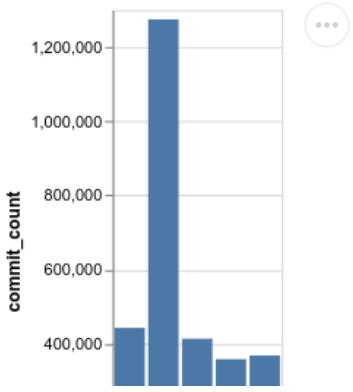
	commit_count	name
0	1274820	Linus Torvalds
1	442631	David S. Miller
2	413276	Linux Build Service Account
3	368108	Takashi Iwai
4	358038	Mark Brown
5	340289	Al Viro
6	324783	Ingo Molnar
7	286911	time
8	259775	Wladimir J. van der Laan
9	258874	Russell King
10	250224	Tejun Heo
11	243956	Greg Kroah-Hartman
12	243740	Mauro Carvalho Chehab
13	238860	H Hartley Sweeten
14	236075	Johannes Berg
15	218118	Thomas Gleixner
16	206329	Jenkins
17	198585	Paul Mundt
18	191148	Felix Fietkau
19	180242	Hans Verkuil
20	173538	Bartlomiej Zolnierkiewicz
21	167974	Christoph Hellwig
22	166043	Joe Perches
23	160787	Arnd Bergmann
24	156048	Adrian Bunk
25	155999	Eric Dumazet
26	153894	Ralf Baechle
27	151870	Alan Cox
28	149298	Trond Myklebust
29	146889	Jeff Garzik
...	...	...
869241	1	eedwardsdisco
869242	1	themonks
869243	1	Théo B.
869244	1	Andras Nagy
869245	1	Graham McBain

```
%%bigquery --project groovy-analyst-227015 q5e_committers
SELECT DISTINCT COUNT(commit) AS commit_count,committer.name
FROM `groovy-analyst-227015.project_part_2_2.github_repo_commits_cs360`
```

```
GROUP BY committer.name
ORDER BY commit_count DESC


# YOUR QUERY HERE
```

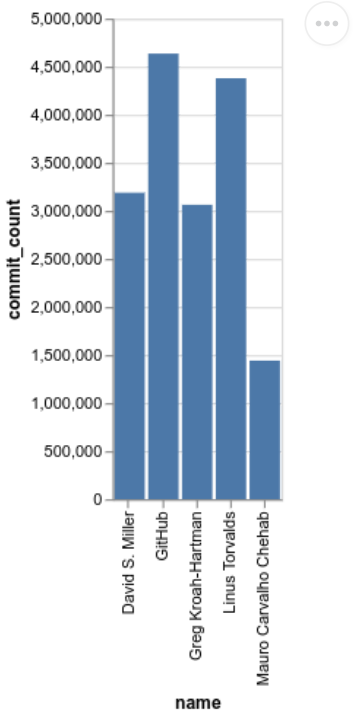
	commit_count	name
0	4636241	GitHub
1	4379958	Linus Torvalds
2	3189260	David S. Miller
3	3063483	Greg Kroah-Hartman
4	1442327	Mauro Carvalho Chehab
5	1297063	John W. Linville
6	1244491	Ingo Molnar
7	792079	Mark Brown
8	669314	Jeff Garzik
9	635043	James Bottomley
10	602614	Russell King
11	506808	Takashi Iwai
12	403276	Ralf Baechle
13	378826	Dave Airlie
14	378416	Gerrit - the friendly Code Review server
15	362202	Paul Mundt
16	343324	Gerrit Code Review
17	332856	Paul Mackerras
18	327930	Len Brown
19	295745	Al Viro
20	293980	Trond Myklebust
21	292357	Daniel Vetter
22	288833	Avi Kivity
23	286911	tiwai

```
# YOUR PLOT CODE HERE - AUTHORS
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt


alt.Chart(q5e_authors.head()).mark_bar().encode(
alt.X("name:N"),
y = 'commit_count:Q')
```




```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt
```

```
alt.Chart(q5e_committers.head()).mark_bar().encode(
alt.X("name:N"),
y = 'commit_count:Q')
```



f) Σε μια παράγραφο (με λιγότερες από 100 λέξεις), περιγράψτε και αναλύστε τα διαγράμματα που δημιουργήσατε. Ποιες ενδιαφέρουσες τάσεις παρατηρείτε στα δεδομένα; Προέκυψε κάτι που δεν ήταν αναμενόμενο; (2 μονάδες)

Εισάγετε εδώ την ανάλυση των διαγραμμάτων σας

Ποια τα χαρακτηριστικά ενός καλού αποθετηρίου (repo)?

Με δεδομένο ότι έχουμε ενδιαφέροντα δεδομένα στη διάθεσή μας, ας προσπαθήσουμε να απαντήσουμε το ερώτημα: ποια τα χαρακτηριστικά ενός καλού αποθετηρίου (repo) του GitHub; Για το σκοπό μας "καλό" θεωρείται ένα αποθετήριο με μεγάλο αριθμό "παρατηρητών", δηλαδή ανθρώπων που παρακολουθούν το αποθετήριο για ενδεχόμενες αλλαγές.

Για αρχή, ας εξετάσουμε εάν κάποια από τα γνωρίσματα *που μόλις διερευνήσαμε* μας δίνουν καλές απαντήσεις.

Ερώτηση 6: Ας χρησιμοποιήσουμε τα αποτελέσματα της προηγούμενης δουλειάς μας (10 μονάδες)

Φτιάξτε γραφικές παραστάσεις για τα ακόλουθα χαρακτηριστικά ενός αποθετηρίου (repo) σε σχέση με τον αριθμό παρατηρητών (watch count) του αποθετηρίου :

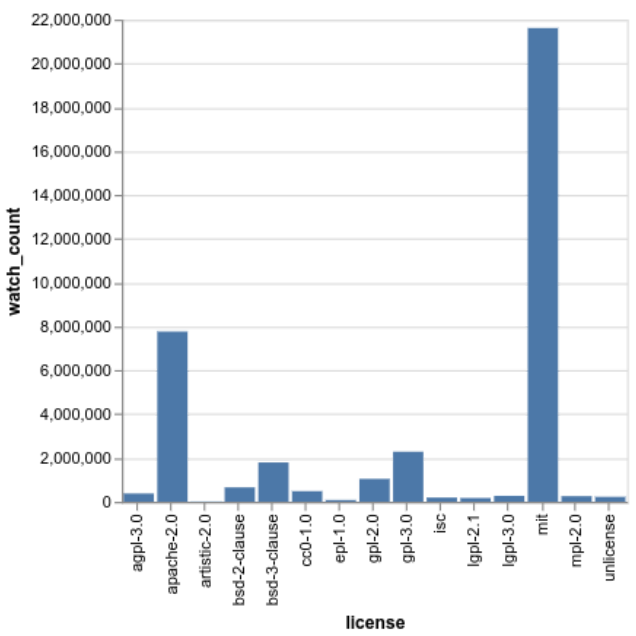
1. Τύπος άδειας
2. Γλώσσες (προγραμματισμού)
3. Μέσο μέγεθος αρχείου στο αποθετήριο
4. Πλήθος αρχείων αποθετηρίου
5. Αριθμός ισχυρών αναθεωρητών ή συντακτών του αποθετηρίου ("power" committers / authors)

a) Τύπος άδειας (1 μονάδα)

```
%%bigquery --project groovy-analyst-227015 q6a

#SELECT DISTINCT license,watch_count FROM `groovy-analyst-227015.project_part_2_2.github_repo_licenses_
#`groovy-analyst-227015.project_part_2_2.github_repos_cs360` repos
#WHERE licenses.lrepo_name=repos.lrepo_name

SELECT license, sum(watch_count) AS watch_count
FROM `groovy-analyst-227015.project_part_2_2.github_repos_cs360` repos
JOIN `groovy-analyst-227015.project_part_2_2.github_repo_licenses_cs360` licenses
ON (repos.lrepo_name = licenses.lrepo_name)
GROUP BY license
ORDER BY watch_count DESC

# YOUR QUERY HERE
```

	license	watch_count
0	mit	21620520
1	apache-2.0	7771999
2	gpl-3.0	2288839
3	bsd-3-clause	1802562
4	gpl-2.0	1049076
5	bsd-2-clause	661800
6	cc0-1.0	489873
7	agpl-3.0	383098
8	lgpl-3.0	274496
9	mpl-2.0	261096
10	unlicense	230085
11	isc	195880
12	lgpl-2.1	176409
13	epl-1.0	87085
14	artistic-2.0	16932

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt

alt.Chart(q6a).mark_bar().encode(
alt.X("license:N"),
y = 'watch_count:Q')
```



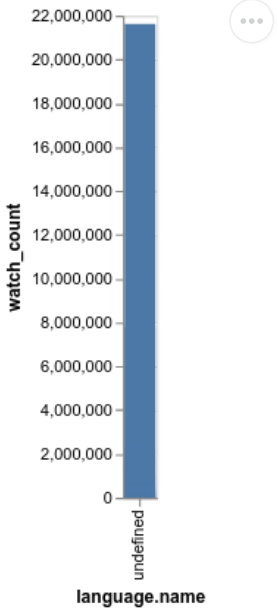
b) Γλώσσες (προγραμματισμού) (1 μονάδα)

```
%%bigquery --project groovy-analyst-227015 q6b

SELECT ANY_VALUE(language) language, sum(watch_count) as watch_count
FROM `groovy-analyst-227015.project_part_2_2.github_repo_languages_cs360` languages
JOIN `groovy-analyst-227015.project_part_2_2.github_repos_cs360` repos
ON (languages.lrepo_name = repos.lrepo_name)
GROUP BY TO_JSON_STRING(language)
ORDER BY watch_count DESC
```

	language	watch_count
0	[]	1566823
1	[{'name': 'CSS', 'bytes': 190263}, {'name': 'H...	291503
2	[{'name': 'CSS', 'bytes': 11301}, {'name': 'HT...	119634
3	[{'name': 'Assembly', 'bytes': 3325}, {'name':...	107721
4	[{'name': 'C', 'bytes': 5225}, {'name': 'C++',...	92644
5	[{'name': 'Batchfile', 'bytes': 5838}, {'name'...	57390
6	[{'name': 'JavaScript', 'bytes': 64350}]	55436
7	[{'name': 'JavaScript', 'bytes': 202}]	53316
8	[{'name': 'CSS', 'bytes': 416604}, {'name': 'H...	52244
9	[{'name': 'C++', 'bytes': 1570996}, {'name': '...	44918
10	[{'name': 'C', 'bytes': 3225}, {'name': 'C++',...	44297
11	[{'name': 'AppleScript', 'bytes': 2214}, {'nam...	44224
12	[{'name': 'CSS', 'bytes': 393}, {'name': 'HTML...	39298
13	[{'name': 'CSS', 'bytes': 329252}, {'name': 'D...	38594
14	[{'name': 'CSS', 'bytes': 3601}, {'name': 'Cof...	37658
15	[{'name': 'Assembly', 'bytes': 1772}, {'name':...	37575
16	[{'name': 'JavaScript', 'bytes': 2740}]	35231
17	[{'name': 'CSS', 'bytes': 428360}, {'name': 'H...	34285
18	[{'name': 'Batchfile', 'bytes': 232}, {'name':...	34155
19	[{'name': 'Assembly', 'bytes': 102873}, {'name...	34068
20	[{'name': 'CSS', 'bytes': 229974}, {'name': 'J...	33493
21	[{'name': 'CSS', 'bytes': 304674}, {'name': 'S...	32649
22	[{'name': 'C', 'bytes': 2840}, {'name': 'Docke...	32563
23	[{'name': 'CSS', 'bytes': 5808}, {'name': 'Ghe...	32458
24	[{'name': 'CSS', 'bytes': 48802}, {'name': 'HT...	32438
25	[{'name': 'AngelScript', 'bytes': 4300}, {'nam...	31298
26	[{'name': 'Batchfile', 'bytes': 1985}, {'name'...	31244
27	[{'name': 'CSS', 'bytes': 128}, {'name': 'Java...	31113
28	[{'name': 'CSS', 'bytes': 24820}, {'name': 'Ja...	30939
29	[{'name': 'Assembly', 'bytes': 28453}, {'name'...	29852
...	...	...
167541	[{'name': 'Scala', 'bytes': 45208}, {'name': '...	12
167542	[{'name': 'C', 'bytes': 51667}, {'name': 'C++'...	12
167543	[{'name': 'Java', 'bytes': 44866}]	12
167544	[{'name': 'Python', 'bytes': 5734}]	12
167545	[{'name': 'Python', 'bytes': 16562}]	12
167546	[{'name': 'CSS', 'bytes': 1911}, {'name': 'Jav...	12
167547	[{'name': 'Java', 'bytes': 421334}]	12

167548	[{'name': 'PHP', 'bytes': 24543}, {'name': 'Sh...	12
167549	[{'name': 'JavaScript', 'bytes': 4585}, {'name...	12
167550	[{'name': 'Go', 'bytes': 41107}, {'name': 'Mak...	12

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt

alt.Chart(q6a).mark_bar().encode(
alt.X("language.name:N"),
y = 'watch_count:Q')
```



167568	[{'name': 'JavaScript', 'bytes': 5503}]	12

c) Μέσο μέγεθος αρχείου στο αποθετήριο (1 μονάδα)

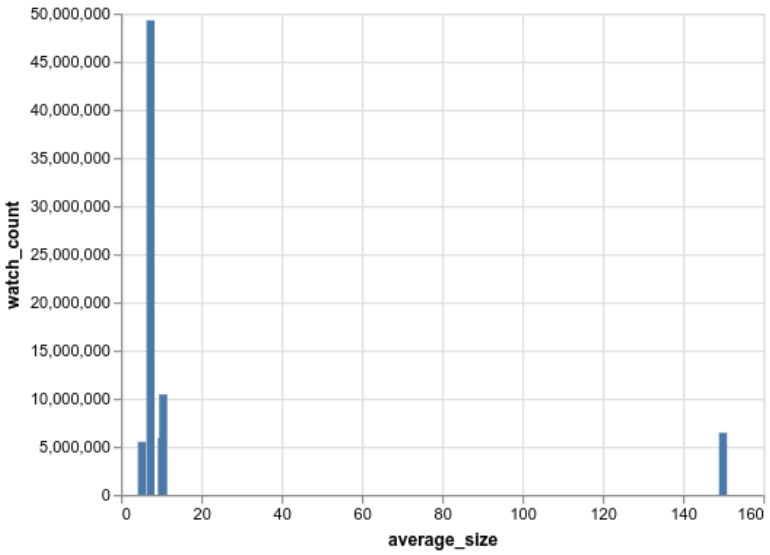
Σημείωση: Για την ερώτηση αυτή μπορείτε να χρησιμοποιήσετε τον πίνακα `github_repo_readme_contents_cs360` αντί του πίνακα με όλο το περιεχόμενο.

```
%%bigquery --project groovy-analyst-227015  q6c
SELECT DISTINCT sum(watch_count) AS watch_count, AVG(size) AS average_size
FROM `groovy-analyst-227015.project_part_2_2.github_repos_cs360` repos
JOIN (SELECT lrepo_name, size
      FROM  `groovy-analyst-227015.project_part_2_2.github_repo_readme_contents_cs360` readmefiles
      JOIN  `groovy-analyst-227015.project_part_2_2.github_repo_files_cs360` files
      ON(readmefiles.id = files.id))files2
ON(repos.lrepo_name = files2.lrepo_name)
GROUP BY repos.lrepo_name
ORDER BY watch_count DESC
```

	watch_count	average_size
0	49297680	7.342241
1	10436250	10.496047
2	6463260	149.950000
3	5929176	10.166667
4	5520936	5.187997
5	5346900	4.682051
6	4857882	620.046512
7	3743618	21.412371
8	3478788	31.805213
9	3248553	2286.189315
10	2967488	8.179020
11	2930670	1613.688889
12	2299766	7.140026
13	2182948	225.618421
14	2117376	64.535156
15	1871428	50.054795
16	1757400	8.217241
17	1566400	118.548864
18	1505712	0.498183
19	1481924	160.368421
20	1457515	49747.200000
21	1432580	16.852410
22	1396788	3116.853659
23	1386743	182.885463
24	1167144	216.507576
25	1132830	13.306189
26	1101516	7.530100
27	1075100	3.574365
28	950232	347.735294
29	944463	9.683060
...	...	...
161194	12	646.000000
161195	12	612.000000
161196	12	1539.000000
161197	12	8264.000000
161198	12	651.000000
161199	12	107.000000
161200	12	1463.000000

161201	12	792.000000
161202	12	500.000000
161203	12	2125.000000
161204	12	2527.000000
161205	12	390.000000
161206	12	4501.000000
161207	12	4867.000000
161208	12	1798.000000
161209	12	1249.000000
161210	12	1786.000000
161211	12	139.000000
161212	12	1605.000000
161213	12	1689.000000

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt

alt.Chart(q6c.head()).mark_bar().encode(
alt.X("average_size:Q"),
y = 'watch_count:Q')
```



d) Πλήθος αρχείων ενός αποθετηρίου (1 μονάδα)

```
%%bigquery --project groovy-analyst-227015 q6d

SELECT  COUNT(id) AS File_count, COUNT(`groovy-analyst-227015.project_part_2_2.github_repo_files_cs360`
AS repo_counter
FROM `groovy-analyst-227015.project_part_2_2.github_repo_files_cs360`,
`groovy-analyst-227015.project_part_2_2.github_repos_cs360`
WHERE `groovy-analyst-227015.project_part_2_2.github_repo_files_cs360`.lrepo_name=`groovy-analyst-22701
```

	File_count	repo_counter
0	68777075	68777075

```
# YOUR PLOT CODE HERE
```

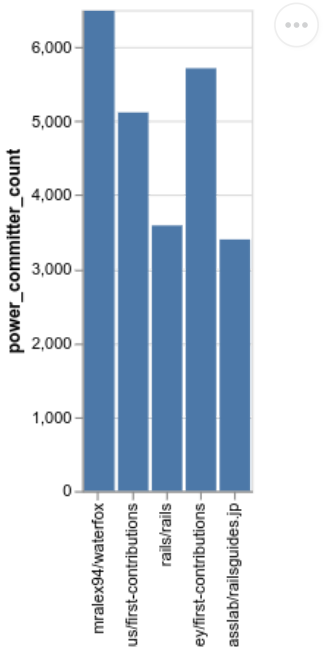
e) Αριθμός ισχυρών αναθεωρητών ή συντακτών ενός αποθετηρίου ("power" committers / authors) (3 μονάδες)

Ορισμός: "ισχυρός" αναθεωρητής ή συντάκτης είναι ένας λογαριασμός (account) μέσω του οποίου έχουν γίνει **τουλάχιστον 1,000 αναθεωρήσεις** (commits) σε αναθεωρήσεις ή συντάξεις (commit/author).

```
%%bigquery --project groovy-analyst-227015  q6e_power_committers
SELECT DISTINCT  lrepo_name, COUNT(DISTINCT committer.name) AS power_committer_count
FROM `groovy-analyst-227015.project_part_2_2.github_repo_commits_cs360`
GROUP BY lrepo_name
HAVING COUNT(commit)>=1000
ORDER BY Power_committer_count DESC

# YOUR QUERY HERE
```

	lrepo_name	power_committer_count
0	mralex94/waterfox	6491
1	roshanjossey/first-contributions	5713
2	multunus/first-contributions	5116
3	rails/rails	3589
4	yasslab/railsguides.jp	3400
5	slavomirvojacek/adbrain-typescript-definitions	2699
6	saltstack/salt	2616
7	docker/docker.github.io	2162
8	kubernetes/kubernetes	2065
9	vmware/kubernetes	2062
10	moby/moby	1993
11	docker/docker	1993
12	caskroom/homebrew-cask	1991
13	chromiumwebapps/chromium	1975
14	jetbrains/swot	1947
15	gitlabhq/gitlabhq	1885
16	coreos/docker	1847
17	resin-io/docker	1836
18	openstack/stackalytics	1731
19	coreos/kubernetes	1664
20	t-zuehlsdorff/gitlabhq	1599
21	ansible/ansible	1589
22	containers/storage	1539
23	projectatomic/docker	1531
24	htve/gitlabforchinese	1512
25	xelabs/tokudb	1442
26	microsoft/docker	1432
27	openstack/nova	1428
28	laravel/framework	1419
29	percona/percona-xtrabackup	1400

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt

alt.Chart(q6e_power_committers.head()).mark_bar().encode(
alt.X("lrepo_name:N"),
y = 'power_committer_count:Q')
```




```
%%bigquery --project groovy-analyst-227015  q6e_power_authors

SELECT DISTINCT  lrepo_name, COUNT(DISTINCT author.name) AS power_author_count
FROM `groovy-analyst-227015.project_part_2_2.github_repo_commits_cs360`
GROUP BY lrepo_name
HAVING COUNT(commit)>=1000
ORDER BY Power_author_count DESC

# YOUR QUERY HERE
```

	lrepo_name	power_author_count
0	google/capsicum-linux	16725
1	minipli/linux-grsec	16723
2	ljalves/linux_media	16359
3	linux-scraping/linux-grsecurity	16243
4	florentrevest/linux-sunxi-cedrus	16120
5	fail0verflow/ps4-linux	16119
6	ipts-linux-org/ipts-linux-new	16075
7	svenkatr/linux	15777
8	helio-x20/linux	15595
9	linusw/linux-bfq	15561
10	segment-routing/sr-ipv6	15533
11	libos-nuse/net-next-nuse	15526
12	parallella/parallella-linux	15374
13	pali/linux-n900	15237
14	adafruit/adafruit-raspberrypi-linux	14967
15	altramayor/xia-for-linux	14962
16	jidetechnology/remixos-kernel	14923
17	96boards/linux	14730
18	xobs/novena-linux	14715
19	patrykk/linux-udoo	14714
20	lukier/linux-hi3518	14688
21	google/ktsan	14112
22	nextthingco/chip-linux	14046
23	alucard24/alucard-kernel-lg-g5	14020
24	01org/edison-linux	13789
25	softroce/rxe-dev	13688
26	tkkg1994/superkernel	13079
27	cirruslogic/rpi-linux	12999
28	01org/igvtg-kernel	12938
29	penberg/linux-kvm	12934
...	...	...
13417	bozimmerman/coffeemud	1
13418	radfordneal/pqr	1
13419	robrix/ui-effects	1
13420	creasty/dotfiles	1
13421	evanhahn/dotfiles	1
13422	hackerfoo/poprc	1
13423	michaelmior/nose	1

13424	trevordmiller/settings	1

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import altair as alt


alt.Chart(q6e_power_authors.head()).mark_bar().encode(
x = 'lrepo_name:N',
y = 'power_authorr_count:Q')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-9-85b7b8d76646> in <module>()
      6
      7
----> 8 alt.Chart(q6e_power_authors.head()).mark_bar().encode(
      9 x = 'lrepo_name:N',
     10 y = 'power_authorr_count:Q')

NameError: name 'q6e_power_authors' is not defined
```

SEARCH STACK OVERFLOW

f) Από όσα μελετήσαμε, υπάρχουν γνωρίσματα και ποια είναι αυτά που έχουν τη μεγαλύτερη συσχέτιση με τα αποθετήρια (repos) με υψηλό αριθμό παρατηρητών; Είναι λογικοφανής η απάντησή σας ή μοιάζει να αντιβαίνει τη διαίσθησή σας; Δώστε την απάντησή σας σε μια παράγραφο, όχι μεγαλύτερη των 200 λέξεων. Αναφερθείτε στις γραφικές παραστάσεις που φτιάξατε. (3 μονάδες)

13442	jpalardy/dotfiles	1

Εισάγετε εδώ την απάντησή σας

13444	google/forensd	1
13445	fractalblocks/fractal	1