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:

- Καραθανάσης Δημητρης 3547
- Συλλιγάρδος Εμμανουήλ 3849

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 = "ancient-tractor-226920"
```

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

```
import altair as alt
import pandas as pd

%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.

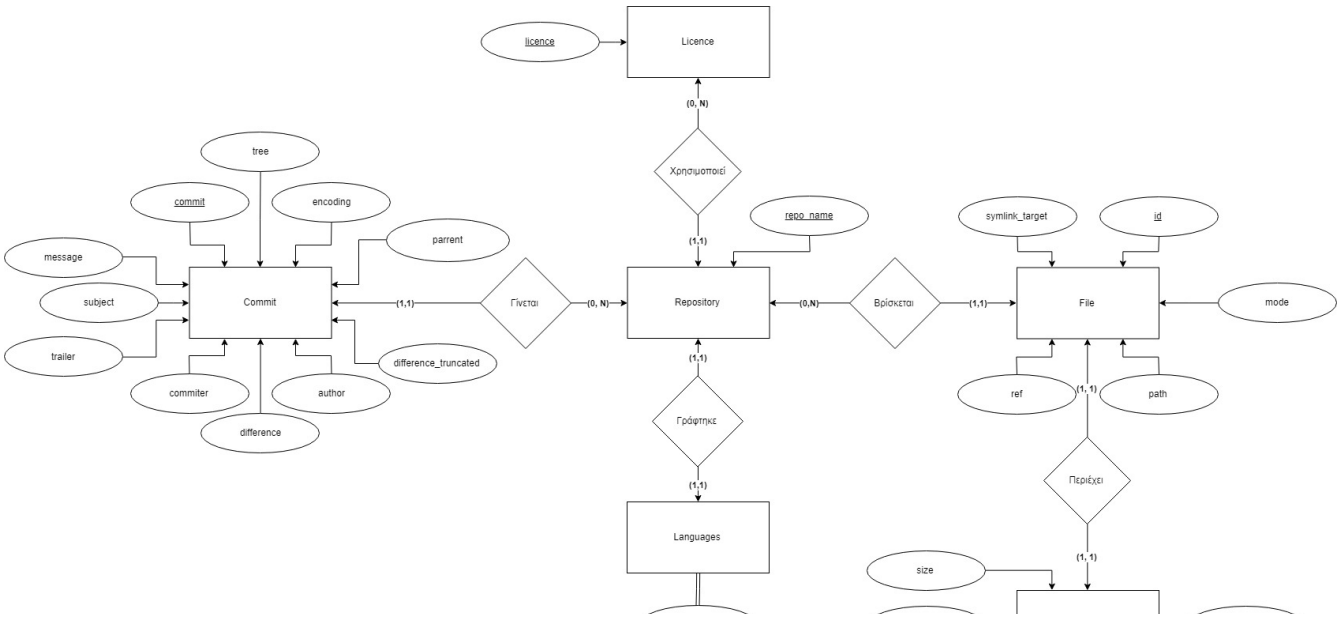
Ερώτηση 1: Πίνακες CS360 GitHub --> Διάγραμμα οντοτήτων-συσχετίσεων (4 μονάδες)

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 l `![](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.

Οι οντότητες στο διάγραμμα μας είναι:

- Commit
- Licence
- File
- Languages
- Content

Οι σχέσεις είναι:

- Commit --(1, 1)--> Γίνεται <--(0, N)-- Repository
- Licence --(0, N)--> Χρησιμοποιεί <--(1, 1)-- Repository
- File --(1, 1)--> Βρίσκεται <--(0, N)-- Repository
- Languages --(1, 1)--> Γράφτηκε <--(1, 1)-- Repository
- Content --(1, 1)--> Περιέχει <--(1, 1)-- File

Τα κλειδιά κάθε οντότητας είναι:

- Commit : commit
- Licence : repo_name
- File : id
- Languages : repo_name
- Content : id

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.

- Repository

repo_name	license

- Commit

commit	repo_name	message	subject	trailer	commiter	difference	author	difference_truncated	parrent	encoding	ti

- License

license

- File

id	repo_name	mode	path	ref	symlink_target

- Languages

name	repo_name	bytes

- Content

id	id_file	repo_name	copies	binary	content	size

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.

Παρατηρώντας, αρχικά τα δύο σχήματα παρατηρούμε ότι οι διαφορές δεν είναι μεγάλες. Συγκεκριμένα, στο δικό μας σχήμα έχουμε επιπλέον ένα πίνακα τον Repository. Ο πίνακας αυτός περιέχει τα ονόματα όλων των repositories του Github και το licence που χρησιμοποιεί κάθε repository. Συνεπώς, στο σχήμα μας, ο πίνακας licence περιέχει μόνο τόσα entries όσα και τα διαφορετικά licences που υπάρχουν και όχι 1 entry για κάθε repo. Ακόμα στον πίνακα Content έχουμε προσθέσει το id_file και το repo_name στο οποίο ανήκει αυτό το content. Έτσι μπορούμε να βρούμε που το κάθε content άμεσα, χωρίς join, να βρούμε σε ποιο file και σε ποιο repo ανήκει. Τελικά, οι διαφορές των δύο σχημάτων είναι λίγες, παρόλα αυτά, θεωρούμε καλύτερο το δικό μας αφού είναι πιο ξεκάθαρο και γλιτώνει μερικά join στα queries.

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 $project_id example

SELECT lrepo_name, watch_count
```

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

```
example.head()
```

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 $project_id q5a

SELECT COUNT(repo_name) AS Nr_of_repos, license
```

```
FROM `bigquery-public-data.github_repos.licenses`
GROUP BY license
```

	Nr_of_repos	license
0	8696	artistic-2.0
1	17814	isc
2	1710301	mit
3	27044	cc0-1.0
4	24425	epl-1.0
5	345720	gpl-2.0
6	343693	gpl-3.0
7	18929	mpl-2.0
8	41513	agpl-3.0
9	22994	lgpl-2.1
10	40069	lgpl-3.0
11	46766	unlicense
12	493471	apache-2.0
13	56062	bsd-2-clause
14	153409	bsd-3-clause

```
alt.Chart(q5a, height = 600, width = 600).mark_bar().encode(
    x="license",
    y="Nr_of_repos"
)
```

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 $project_id q5b

SELECT language, COUNT(repo_name)
FROM
(SELECT language[OFFSET(0)].name AS language, repo_name
FROM `bigquery-public-data.github_repos.languages`
UNION DISTINCT
SELECT language[OFFSET(1)].name AS language, repo_name
FROM `bigquery-public-data.github_repos.languages`
UNION DISTINCT
SELECT language[OFFSET(2)].name AS language, repo_name
FROM `bigquery-public-data.github_repos.languages`
UNION DISTINCT
SELECT language[OFFSET(3)].name AS language, repo_name
FROM `bigquery-public-data.github_repos.languages`)
GROUP BY language


# YOUR PLOT CODE HERE
```

c) Κατανομή μεγέθους αρχείων (1 μονάδα)

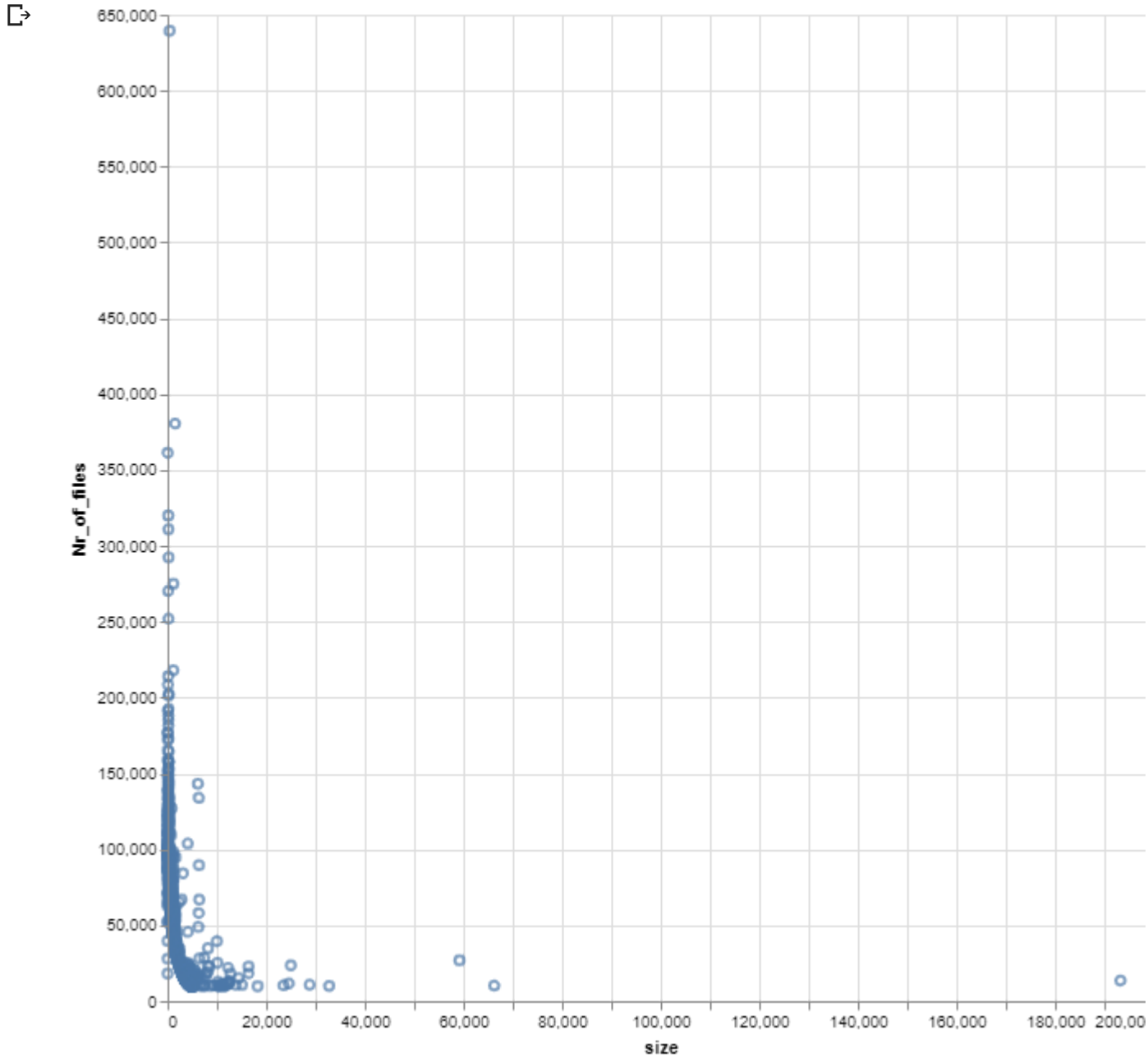
(x-άξονας: μέγεθος αρχείου, y-άξονας: # αρχείων με αυτό το μέγεθος)

```
%%bigquery --project $project_id q5c

SELECT size, COUNT(id) AS Nr_of_files
FROM
(SELECT id, size
FROM `bigquery-public-data.github_repos.contents`)
GROUP BY size
ORDER BY Nr_of_files DESC LIMIT 4999
```

	size	Nr_of_files
0	399	639528
1	1507	380646
2	32	361356
3	129	320041

```
alt.Chart(q5c,height=600,width=600).mark_point().encode(
      x='size',
      y='Nr_of_files',
)
```



28	212	157538

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

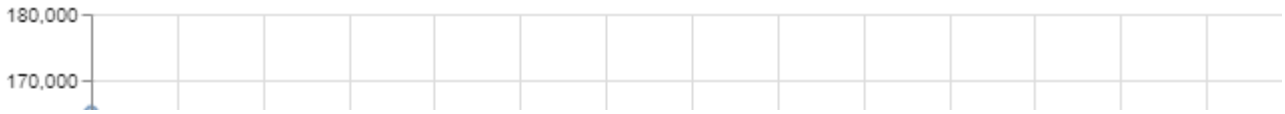
(x-άξονας: # αρχείων που σχετίζονται με ένα αποθετήριο (repo) , y-άξονας: # αποθετηρίων (repos) που σχετίζονται με τέτοιο πλήθος αρχείων)

```
%%bigquery --project $project_id q5d

SELECT COUNT(repo_name) AS Nr_of_repos, Nr_of_files
FROM
(SELECT repo_name, COUNT(id) AS Nr_of_files
FROM
(SELECT repo_name, id
FROM `bigquery-public-data.github_repos.files`)
GROUP BY repo_name)
GROUP BY Nr_of_files
ORDER BY Nr_of_repos DESC LIMIT 4999
```

	Nr_of_repos	Nr_of_files
0	165215	3
1	108295	2
2	107982	4
3	103796	5
4	98051	6
5	91019	7
6	87761	8
7	81600	9

```
alt.Chart(q5d,height=600,width=600).mark_point().encode(
      x='Nr_of_files',
      y='Nr_of_repos',
)
```



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 $project_id q5e_authors

# YOUR QUERY HERE
```

```
%%bigquery --project $project_id q5e_committers

# YOUR QUERY HERE
```

```
# YOUR PLOT CODE HERE - AUTHORS
```

```
# YOUR PLOT CODE HERE - COMMITTERS
```

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 $project_id q6a
# YOUR QUERY HERE


# YOUR PLOT CODE HERE
```

b) Γλώσσες (προγραμματισμού) (1 μονάδα)

```
%%bigquery --project $project_id q6b
# YOUR QUERY HERE


# YOUR PLOT CODE HERE
```

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 $project_id q6c
# YOUR QUERY HERE


# YOUR PLOT CODE HERE
```

d) Πλήθος αρχείων ενός αποθετηρίου (1 μονάδα)

```
%%bigquery --project $project_id q6d
# YOUR QUERY HERE


# 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

```
%%bigquery --project $project_id q6e_power_committers

# YOUR QUERY HERE


# YOUR PLOT CODE HERE


%%bigquery --project $project_id q6e_power_authors

# YOUR QUERY HERE


# YOUR PLOT CODE HERE
```

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)

Εισάγετε εδώ την απάντησή σας
