TASK 1

MUHAMMAD MOIZ FAROOQUI

Objective 1:

Get a list of the last 1 Million repos from Github, load data into a dataset, and run an exploratory data analysis
source: [List public repositories](#).

Highlight any differences and trends in the dataset by user attributes, programming languages, and other attributes.

Note: you need to fetch **at least** 600,000 repos for us to consider this adequately completed.

Deliverable:

1. Share the codebase in a notebook or python file, the dataset, any assets created, and instructions to run it.
2. Would be assessed on:

a. Quality, testing, and reliability of the data pipeline
   - **Bonus:** Loading data into a cloud data warehouse (GCP, Azure, AWS, etc.)

b. Visualization of results
   - Focus on identifying and analyzing distinct audiences in the dataset (you can use descriptive or predictive approaches).

Fetching Github repositories.

Around 850000 repositories were fetched initially for the analysis.

In [1]:
```
import requests
import pandas as pd
import tqdm
import subprocess
```

In [2]:
```
respositories_url = "https://api.github.com/repositories?since={}"
GH_AUTH_TOKEN = "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"
PAGE_NO=30
```

In [ ]:
```
response_list =[]
required_repos= 1000000
per_page_repos = 100
iterations = int(required_repos/per_page_repos)
print(f"Running {iterations} iterations.....")
repo_id=1800000
for i in tqdm.tqdm(range(iterations)):
    try:
        response = requests.get(respositories_url.format(repo_id), headers={"Authorization":f"Bearer {GH_AUTH_TOKEN}"}).json()
        repo_id = response[-1]['id']
        response_list.extend(response)
        repositoy_df = pd.json_normalize(response_list)
        repositoy_df.to_csv("respositories.tsv", sep='\t')
    except Exception as e:
        print("Exception",e)
        pass
```

```
Running 10000 iterations.....
  7%|▉▉▉▉                                                    | 687/10000 [55:51<22:44:16,  8.79s/it]
```

In [5]:
```
qq = pd.read_csv(r'respositories.tsv',sep ="\t", on_bad_lines='skip')
qq.head(5)
```

Out[5]:

	Unnamed: 0	id	node_id	name	full_name	private	html_url	description	fork
0	0	2037156	MDEwOlJlcG9zaXRvcnkyMDM3MTU2	Roar-Engine-API-for-Unity	digimonk/Roar-Engine-API-for-Unity	False	https://github.com/digimonk/Roar-Engine-API-fo...	Unity C# wrapper for Roar Engine 2.0 with inbu...	False
1	1	2037157	MDEwOlJlcG9zaXRvcnkyMDM3MTU3	core	rockharbor/core	False	https://github.com/rockharbor/core	In-depth church member and event management so...	False
2	2	2037158	MDEwOlJlcG9zaXRvcnkyMDM3MTU4	xen-api-libs-debian	jonludlam/xen-api-libs-debian	False	https://github.com/jonludlam/xen-api-libs-debian	debian packaging files for xen-api-libs	False
3	3	2037159	MDEwOlJlcG9zaXRvcnkyMDM3MTU5	HaitiReporter	digidem/HaitiReporter	False	https://github.com/digidem/HaitiReporter	Database System for reporting gender-based vio...	False
4	4	2037160	MDEwOlJlcG9zaXRvcnkyMDM3MTYw	gayt	gaytjeanluc/gayt	False	https://github.com/gaytjeanluc/gayt	NaN	False

Analysis on 850000 Repositories:

1. Most of the Repositories are owned individually by the user itself the others are owned by the organizations.
   - User 775102
   - Organization 74892

2. The Forked and Un-Forked repositories are as follows:
   - Forked 521690
   - Organization 328304

Additional analysis was conducted following the retrieval of additional repository data using Repository URLs. This process involved fetching further details about the respective repositories.

The program was provided with Repository URLs to retrieve specific details, The repositories were fetched sequentially therefore does not any ways depicts the trends of the data fetched earlier of about 850,000 Repositories. Kindly take this in consideration.

From a pool of 849,994 repositories, additional details were obtained for around 29,000 repositories to facilitate a more in-depth analysis.

In [19]:
```
url = "{}"
GH_AUTH_TOKEN = "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"
PAGE_NO=30
```

In [ ]:
```
response_list =[]
for i,url in enumerate(tqdm.tqdm(df['url'].tolist())):
    try:
        response = requests.get(url, headers={"Authorization":f"Bearer {GH_AUTH_TOKEN}"}).json()
        repo_id = response['id']
        response_list.append(response)
        if i%1000==0:
            repositoy_df = pd.json_normalize(response_list)
            repositoy_df.to_csv("url_repos.tsv", sep='\t')
    except Exception as e:
        print("Exception",e)
        pass
```

In [237]:
```
df = pd.read_table(r'url_repos.tsv', sep='\t', on_bad_lines='skip',encoding='unicode_escape')
```

In [238]:
```
dt = pd.read_table(r'C:\Users\moizf\Desktop\url_repos.tsv', sep='\t', on_bad_lines='skip',encoding='unicode_escape')
```

In [239]:
```
test=pd.concat([df,dt],axis=0)
```

In [242]:
```
test.head(5)
```

Out[242]:

	Unnamed: 0	id	node_id	name	full_name	private	html_url	description	fork	
0	0	1	MDEwOlJlcG9zaXRvcnkx	grit	mojombo/grit	False	https://github.com/mojombo/grit	**Grit is no longer maintained. Check out libg...	False	https://api.github
1	1	26	MDEwOlJlcG9zaXRvcnkyNg==	merb-core	wycats/merb-core	False	https://github.com/wycats/merb-core	Merb Core: All you need. None you don't.	False	https://api.github.com
2	2	27	MDEwOlJlcG9zaXRvcnkyNw==	rubinius	rubinius/rubinius	False	https://github.com/rubinius/rubinius	The Rubinius Language Platform	False	https://api.github.co

Following results can be drawn from above analysis.

1. The most Popular Repository is Rails-Ruby on Rails with 5432 Stars.

2. The Top 5 languages of the repositories are as Follows.
   - Ruby 11105
   - JavaScript 2271
   - Python 1311
   - C 902
   - PHP 668

3. Most of the Repositories don’t have any topics defined.

In [1]:
```
import requests
import pandas as pd
import tqdm
import subprocess
```

In [2]:
```
respositories_url = "https://api.github.com/repositories?since={}"
GH_AUTH_TOKEN = "github_pat_11A6IYZXA0DWXvzOziEs3V_fwHWzvgABMt8DEHzfciAb8NPHo1X30x0
PAGE_NO=30
```

In [ ]:
```
response_list =[]
required_repos= 1000000
per_page_repos = 100
iterations = int(required_repos/per_page_repos)
print(f"Running {iterations} iterations.....")
repo_id=1800000
for i in tqdm.tqdm(range(iterations)):
    try:
        response = requests.get(respositories_url.format(repo_id), headers={"Author
        repo_id = response[-1]['id']
        response_list.extend(response)
        repositoy_df = pd.json_normalize(response_list)
        repositoy_df.to_csv("respositories.tsv", sep='\t')
    except Exception as e:
        print("Exception",e)
        pass
```

```
Running 10000 iterations.....
  7%|▊                                                                              |
687/10000 [55:51<22:44:16,  8.79s/it]
```

In [4]:
```
repo_id
```

Out[4]:
```
2037154
```

In [ ]:
```
response_list =[]
required_repos= 1000000
per_page_repos = 100
iterations = int(required_repos/per_page_repos)
print(f"Running {iterations} iterations.....")
repo_id=2037154
for i in tqdm.tqdm(range(iterations)):
    try:
        response = requests.get(respositories_url.format(repo_id), headers={"Author
        repo_id = response[-1]['id']
        response_list.extend(response)
        if i%100==0:
            repositoy_df = pd.json_normalize(response_list)
            repositoy_df.to_csv("respositories.tsv", sep='\t')
    except Exception as e:
        print("Exception",e)
        pass
```

```
Running 10000 iterations.....
 33%|████████████████████████                                                   | 334
8/10000 [1:02:06<18:41:29, 10.12s/it]
Exception HTTPSConnectionPool(host='api.github.com', port=443): Max retries exceed
ed with url: /repositories?since=2960792 (Caused by SSLError(SSLEOFError(8, '[SSL:
UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:100
6)')))
```

In [126…
```
import requests
import pandas as pd
import tqdm
import subprocess
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('default')
```

In [5]:
```
qq = pd.read_csv(r'respositories.tsv',sep ="\t", on_bad_lines='skip')
qq.head(5)
```

Out[5]:

	Unnamed: 0	id	node_id	name	full_name	priv
0	0	2037156	MDEwOlJlcG9zaXRvcnkyMDM3MTU2	Roar-Engine-API-for-Unity	digimonk/Roar-Engine-API-for-Unity	Fa
1	1	2037157	MDEwOlJlcG9zaXRvcnkyMDM3MTU3	core	rockharbor/core	Fa
2	2	2037158	MDEwOlJlcG9zaXRvcnkyMDM3MTU4	xen-api-libs-debian	jonludlam/xen-api-libs-debian	Fa
3	3	2037159	MDEwOlJlcG9zaXRvcnkyMDM3MTU5	HaitiReporter	digidem/HaitiReporter	Fa
4	4	2037160	MDEwOlJlcG9zaXRvcnkyMDM3MTYw	gayt	gaytjeanluc/gayt	Fa

5 rows × 64 columns

In [6]:
```
# Checking for null values
qq.isnull().sum()
```

Out[6]:
```
Unnamed: 0                  0
id                          0
node_id                     0
name                        6
full_name                   0
                           ..
owner.repos_url             0
owner.events_url            0
owner.received_events_url   0
owner.type                  0
owner.site_admin            0
Length: 64, dtype: int64
```

In [7]:
```
# Dropping Null values
qq.dropna(subset=['name'], inplace=True)
```

Lets Rename Some of the columns for better understanding of the the data

In [8]:
```
qq.rename(columns={'owner.following_url ':'ofollowing_url','owner.gists_url':'ogist
                   'owner.subscriptions_url':'osubscriptions_url','owner.organizati
                   'owner.repos_url':'orepos_url','owner.events_url':'oevents_url',
                   'owner.site_admin':'osite_admin','owner.login': 'ologin','owner.
                   'owner.node_id':'onode_id','owner.url':'ourl','owner.followers_u
                    },inplace=True)
```

In [9]:
```
qq.head(9)
```

Out[9]:

	Unnamed: 0	id	node_id	name	full_name	priv
0	0	2037156	MDEwOlJlcG9zaXRvcnkyMDM3MTU2	Roar-Engine-API-for-Unity	digimonk/Roar-Engine-API-for-Unity	Fa
1	1	2037157	MDEwOlJlcG9zaXRvcnkyMDM3MTU3	core	rockharbor/core	Fa
2	2	2037158	MDEwOlJlcG9zaXRvcnkyMDM3MTU4	xen-api-libs-debian	jonludlam/xen-api-libs-debian	Fa
3	3	2037159	MDEwOlJlcG9zaXRvcnkyMDM3MTU5	HaitiReporter	digidem/HaitiReporter	Fa
4	4	2037160	MDEwOlJlcG9zaXRvcnkyMDM3MTYw	gayt	gaytjeanluc/gayt	Fa
5	5	2037162	MDEwOlJlcG9zaXRvcnkyMDM3MTYy	platform-agent	leithaus/platform-agent	Fa
6	6	2037164	MDEwOlJlcG9zaXRvcnkyMDM3MTY0	level-theme	jamespidgeon/level-theme	Fa
7	7	2037167	MDEwOlJlcG9zaXRvcnkyMDM3MTY3	Poker	videoplaza/Poker	Fa
8	8	2037171	MDEwOlJlcG9zaXRvcnkyMDM3MTcx	premailer	Bertrand/premailer	Fa

9 rows × 64 columns

In [10]:
```
qq.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 849994 entries, 0 to 849999
Data columns (total 64 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Unnamed: 0            849994 non-null  int64  
 1   id                    849994 non-null  int64  
 2   node_id               849994 non-null  object 
 3   name                  849994 non-null  object 
 4   full_name             849994 non-null  object 
 5   private               849994 non-null  bool   
 6   html_url              849994 non-null  object 
 7   description           679965 non-null  object 
 8   fork                  849994 non-null  bool   
 9   url                   849994 non-null  object 
 10  forks_url             849994 non-null  object 
 11  keys_url              849994 non-null  object 
 12  collaborators_url     849994 non-null  object 
 13  teams_url             849994 non-null  object 
 14  hooks_url             849994 non-null  object 
 15  issue_events_url      849994 non-null  object 
 16  events_url            849994 non-null  object 
 17  assignees_url         849994 non-null  object 
 18  branches_url          849994 non-null  object 
 19  tags_url              849994 non-null  object 
 20  blobs_url             849994 non-null  object 
 21  git_tags_url          849994 non-null  object 
 22  git_refs_url          849994 non-null  object 
 23  trees_url             849994 non-null  object 
 24  statuses_url          849994 non-null  object 
 25  languages_url         849994 non-null  object 
 26  stargazers_url        849994 non-null  object 
 27  contributors_url      849994 non-null  object 
 28  subscribers_url       849994 non-null  object 
 29  subscription_url      849994 non-null  object 
 30  commits_url           849994 non-null  object 
 31  git_commits_url       849994 non-null  object 
 32  comments_url          849994 non-null  object 
 33  issue_comment_url     849994 non-null  object 
 34  contents_url          849994 non-null  object 
 35  compare_url           849994 non-null  object 
 36  merges_url            849994 non-null  object 
 37  archive_url           849994 non-null  object 
 38  downloads_url         849994 non-null  object 
 39  issues_url            849994 non-null  object 
 40  pulls_url             849994 non-null  object 
 41  milestones_url        849994 non-null  object 
 42  notifications_url     849994 non-null  object 
 43  labels_url            849994 non-null  object 
 44  releases_url          849994 non-null  object 
 45  deployments_url       849994 non-null  object 
 46  ologin                849994 non-null  object 
 47  oid                   849994 non-null  int64  
 48  onode_id              849994 non-null  object 
 49  owner.avatar_url      849994 non-null  object 
 50  owner.gravatar_id     0 non-null       float64
 51  ourl                  849994 non-null  object 
 52  owner.html_url        849994 non-null  object 
 53  ofollowers_url        849994 non-null  object 
 54  owner.following_url   849994 non-null  object 
 55  ogists_url            849994 non-null  object 
 56  ostarred_url          849994 non-null  object 
 57  osubscriptions_url    849994 non-null  object 
 58  oorganizations_url    849994 non-null  object 
```

```
 59  orepos_url            849994 non-null  object
 60  oevents_url           849994 non-null  object
 61  oreceived_events_url  849994 non-null  object
 62  otype                 849994 non-null  object
 63  osite_admin           849994 non-null  bool
dtypes: bool(3), float64(1), int64(3), object(57)
memory usage: 404.5+ MB
```

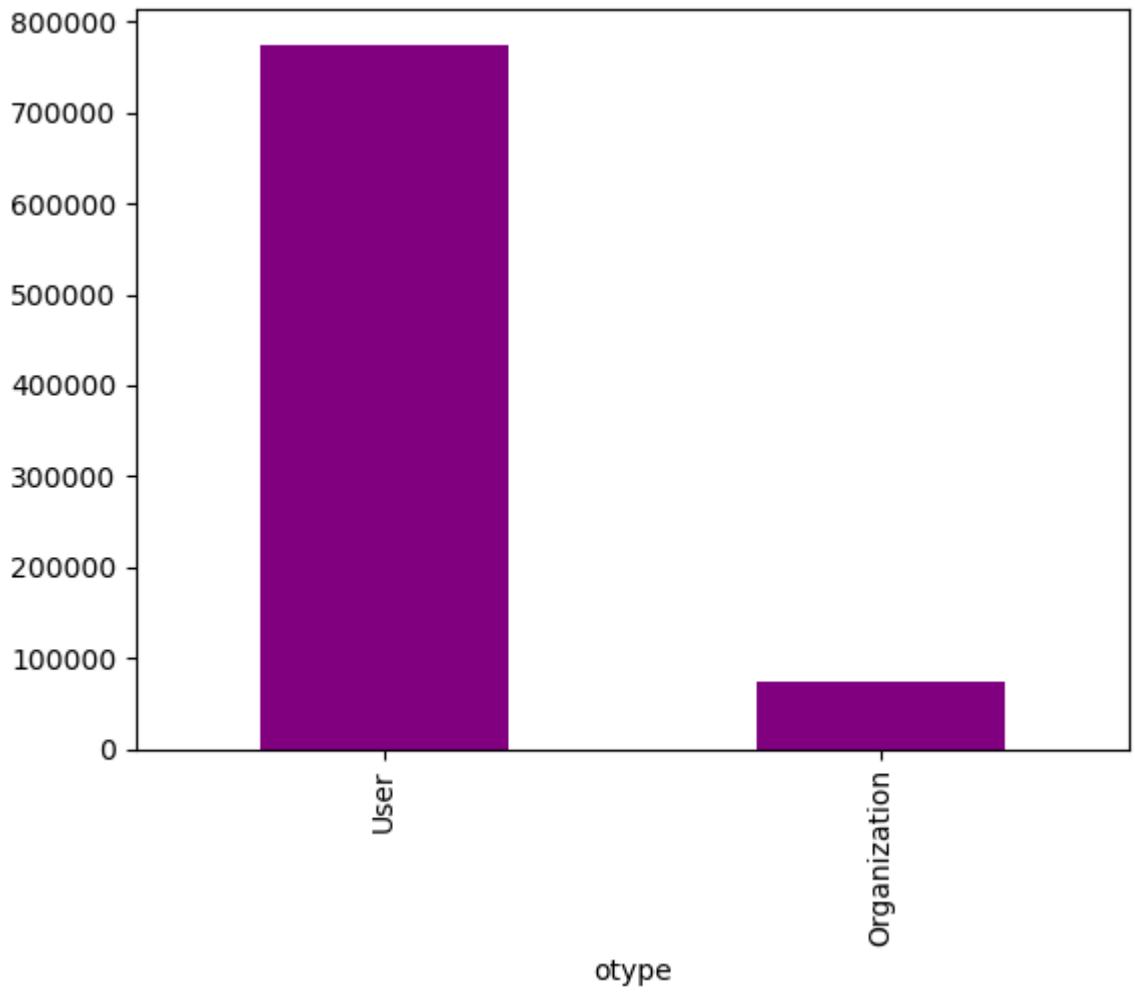
Lets Check how many repositories are owned by a single user and how many are owned by the Organization

In [11]:
```
qq.otype.value_counts()
```

Out[11]:
```
otype
User            775102
Organization     74892
Name: count, dtype: int64
```

In [12]:
```
qq['otype'].value_counts().plot(kind='bar', color='purple')
```

Out[12]:
```
<Axes: xlabel='otype'>
```



we can conclude by the above graph that most of the repositories present at github are owned by the users individually . The ratio of User owned to organization owned repositories is almost 10:1.

Lets Check whether the repositories are forked or not

For Understanding: A fork is a new repository that shares code and visibility settings with the original “upstream” repository.Forking is used when you want to contribute to a project without affecting the original project

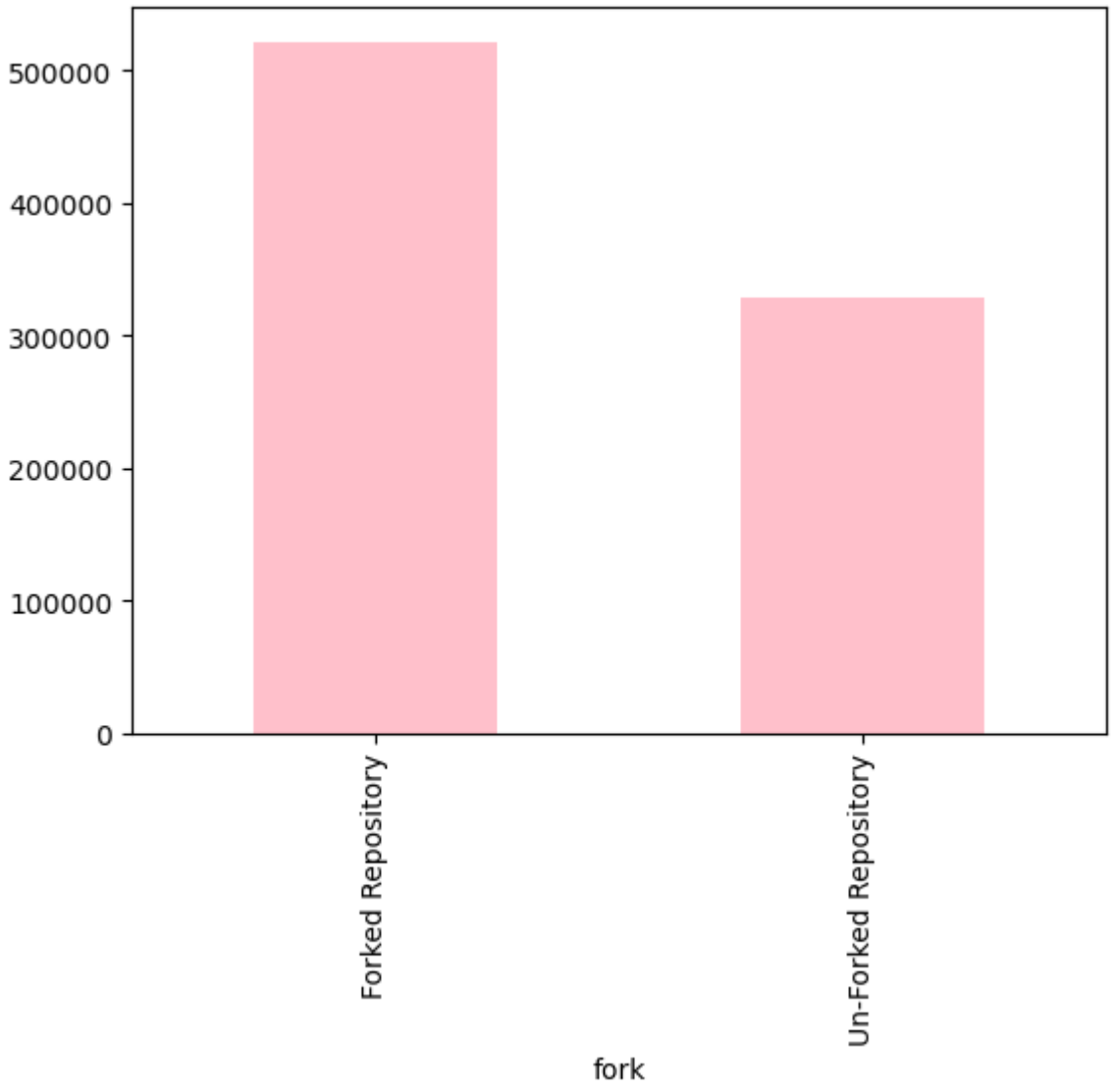
In [13]:
```
qq.fork.value_counts()
```

Out[13]:
```
fork
False    521690
True     328304
Name: count, dtype: int64
```

In [14]:
```
plot=qq['fork'].value_counts().plot(kind='bar',color='pink')

plot.set_xticklabels( ('Forked Repository', 'Un-Forked Repository') )
```

Out[14]:
```
[Text(0, 0, 'Forked Repository'), Text(1, 0, 'Un-Forked Repository')]
```



Additional analysis was conducted following the retrieval of additional repository data through the use of Repository URLs. This process involved fetching further details about the respective repositories.

The program was provided with Repository URLs to retrieve specific details, The repositories were fetched sequentially therfefore doesnot any ways depicts the trends of the data fetched earlier of about 850,000 Repositories . Kindly keep this i consideration

From a pool of 849,994 repositories, additional details were obtained for around 29,000 repositories to facilitate a more in-depth analysis.

In [15]:
```
df = pd.read_excel(r'C:\Users\moizf\Desktop\test.xlsx')
```

In [18]:
```
df
```

Out[18]:

	url
0	https://api.github.com/repos/mojombo/grit
1	https://api.github.com/repos/wycats/merb-core
2	https://api.github.com/repos/rubinius/rubinius
3	https://api.github.com/repos/mojombo/god
4	https://api.github.com/repos/vanpelt/jsawesome
...	...
194995	https://api.github.com/repos/barsoomcore/dponi...
194996	https://api.github.com/repos/jmacgreg/docs
194997	https://api.github.com/repos/bonefish/active_s...
194998	https://api.github.com/repos/jgagner/datastruc...
194999	https://api.github.com/repos/dbpatterson/happs...

195000 rows × 1 columns

In [19]:
```
url = "{}"
GH_AUTH_TOKEN = "github_pat_11A6IYZXA0DWXvzOziEs3V_fwHWzvgABMt8DEHzfciAb8NPHo1X30x0
PAGE_NO=30
```

In [ ]:
```
response_list =[]
for i,url in enumerate(tqdm.tqdm(df['url'].tolist())):
    try:
        response = requests.get(url, headers={"Authorization":f"Bearer {GH_AUTH_TOK
        repo_id = response['id']
        response_list.append(response)
        if i%1000==0:
            repositoy_df = pd.json_normalize(response_list)
            repositoy_df.to_csv("url_repos.tsv", sep='\t')
    except Exception as e:
        print("Exception",e)
        pass
```

```
  0%|                                                                        | 1
79/195000 [03:21<56:45:48,  1.05s/it]
Exception 'id'
  0%|                                                                        | 3
78/195000 [07:01<58:39:15,  1.08s/it]
Exception 'id'
  0%|                                                                        | 5
78/195000 [10:36<39:23:15,  1.37it/s]
```

```
 13%|████████                                                          | 25037/
195000 [7:21:03<184:47:42,  3.91s/it]
Exception HTTPSConnectionPool(host='api.github.com', port=443): Max retries exceed
ed with url: /repos/ktheory/rightscale_api_wrapper (Caused by ConnectTimeoutError
(<urllib3.connection.HTTPSConnection object at 0x000001B5A91BC450>, 'Connection to
api.github.com timed out. (connect timeout=None)'))
```

In [237…
```
df = pd.read_table(r'url_repos.tsv', sep='\t', on_bad_lines='skip',encoding='unicod
```

In [238…
```
dt = pd.read_table(r'C:\Users\moizf\Desktop\url_repos.tsv', sep='\t', on_bad_lines=
```

In [239…
```
test=pd.concat([df,dt],axis=0)
```

In [242…
```
test.head(5)
```

Out[242]:

	Unnamed: 0	id	node_id	name	full_name	private	
0	0	1	MDEwOlJlcG9zaXRvcnkx	grit	mojombo/grit	False	https://(
1	1	26	MDEwOlJlcG9zaXRvcnkyNg==	merb-core	wycats/merb-core	False	https://githu
2	2	27	MDEwOlJlcG9zaXRvcnkyNw==	rubinius	rubinius/rubinius	False	https://gitl
3	3	28	MDEwOlJlcG9zaXRvcnkyOA==	god	mojombo/god	False	https://g
4	4	29	MDEwOlJlcG9zaXRvcnkyOQ==	jsawesome	vanpelt/jsawesome	False	https://githu

5 rows × 329 columns

In [194…
```
test.columns
```

Out[194]:
```
Index(['Unnamed: 0', 'id', 'node_id', 'name', 'full_name', 'private',
       'html_url', 'description', 'fork', 'url',
       ...
       'source.is_template', 'source.web_commit_signoff_required',
       'source.topics', 'source.visibility', 'source.forks',
       'source.open_issues', 'source.watchers', 'source.default_branch',
       'parent.license', 'source.license'],
      dtype='object', length=329)
```

In [195…
```
# check for duplicate values in id column
duplicate_values = test['id'].duplicated()
duplicate_values
```

Out[195]:
```
0       False
1       False
2       False
3       False
4       False
        ...
7348     True
7349     True
7350     True
7351     True
7352     True
Name: id, Length: 29276, dtype: bool
```

In [196…
```
test = test.drop_duplicates(subset=['id'], keep='first')
```

In [197…
```
# check for duplicate values in id column
duplicate_values = test['id'].duplicated()
duplicate_values
```

Out[197]:
```
0       False
1       False
2       False
3       False
4       False
        ...
4093    False
4094    False
4095    False
5060    False
5633    False
Name: id, Length: 23974, dtype: bool
```

Most Popular Repositories wrt Stars:

In [198…
```
test['stargazers_count'] = pd.to_numeric(test['stargazers_count'],errors='coerce')
```

In [199…
```
test[["stargazers_count"]].describe(include="all")
```

Out[199]:

	stargazers_count
count	23971.000000
mean	52.078595
std	691.419449
min	0.000000
25%	2.000000
50%	3.000000
75%	8.000000
max	54314.000000

The Top 5 Repositories based on the no of stars.

In [200…
```
starTop=test.sort_values(by = "stargazers_count", ascending = False).head(5)
starTop
```

Out[200]:

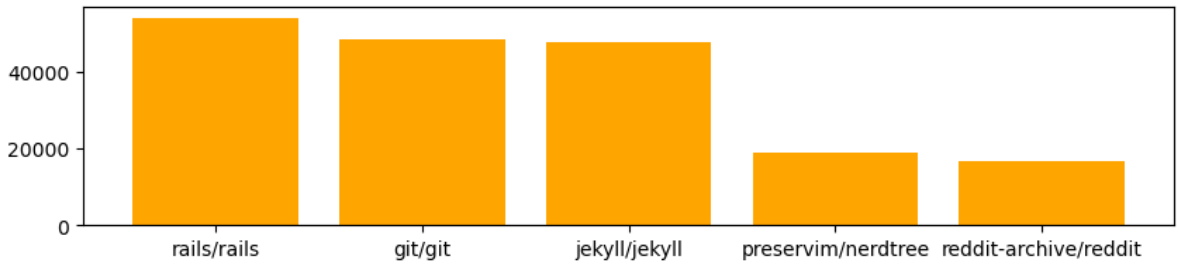
	Unnamed: 0	id	node_id	name	full_name	private
1641	1641	8514	MDEwOlJlcG9zaXRvcnk4NTE0	rails	rails/rails	False
8125	8125	36502	MDEwOlJlcG9zaXRvcnkzNjUwMg==	git	git/git	False
16415	16415	65252	MDEwOlJlcG9zaXRvcnk2NTI1Mg==	jekyll	jekyll/jekyll	False
663	663	3282	MDEwOlJlcG9zaXRvcnkzMjgy	nerdtree	preservim/nerdtree	False
5763	5763	26554	MDEwOlJlcG9zaXRvcnkyNjU1NA==	reddit	reddit-archive/reddit	False

5 rows × 329 columns

In [219…

```
y = starTop.stargazers_count
x = starTop.full_name

plt.figure(figsize=(10,2))
plt.bar(x,y,color='orange')
plt.show()
```



The Last 5 Repositories based on the no of stars.

In [204…

```
starBottom=test.sort_values(by = "stargazers_count", ascending = False).tail(5)
starBottom
```

Out[204]:

	Unnamed: 0	id	node_id
1031	1031	5111	MDEwOlJlcG9zaXRvcnk1MTEx
12196	12196	50609	MDEwOlJlcG9zaXRvcnk1MDYwOQ==
2082	2082	11415	MDEwOlJlcG9zaXRvcnkxMTQxNQ==
2083	kom veleu	NaN	NaN
2084	iliatu u Zagrebu.	False	https://api.github.com/repos/captblanket/h2tx

5 rows × 329 columns

In [213…
```
maxCount = test['stargazers_count'].max()
maxCount
```

Out[213]:
```
54314.0
```

In [214…
```
minCount = test['stargazers_count'].min()
minCount
```

Out[214]:
```
0.0
```

Most popular repo- 54314 stars | Least popular repo- 0 stars

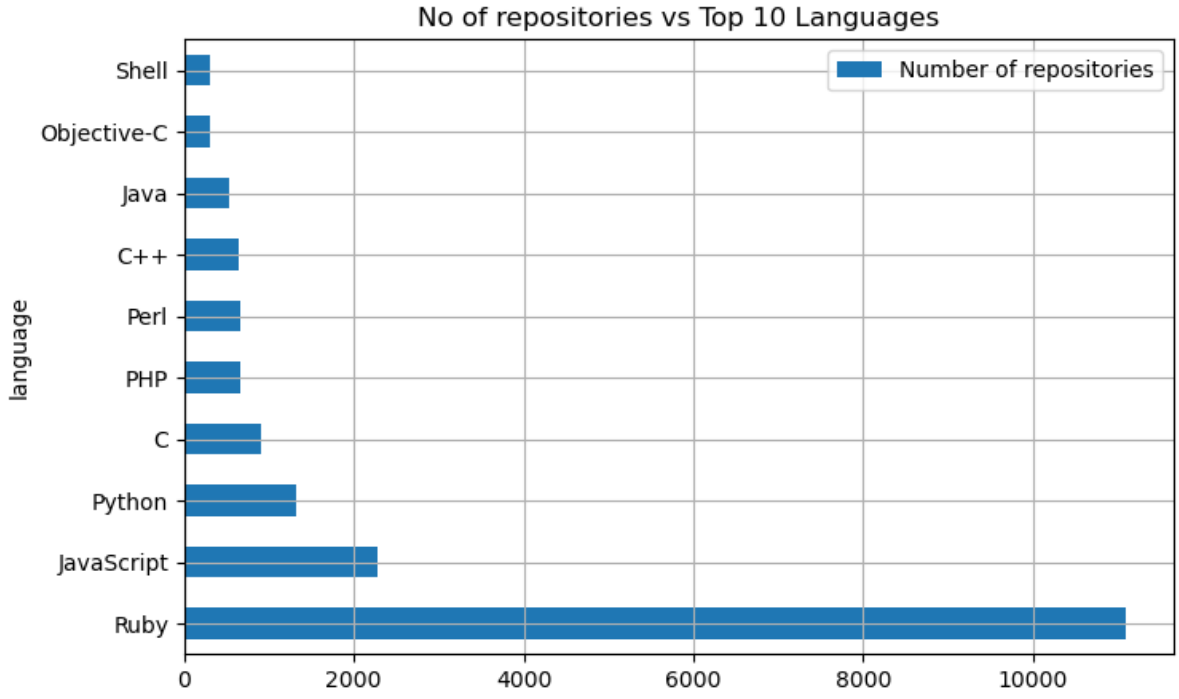
Most Popular Languages:

In [215…
```
lang=test.language.value_counts()
lang.head(5)
```

Out[215]:
```
language
Ruby          11105
JavaScript     2271
Python         1311
C               902
PHP             668
Name: count, dtype: int64
```

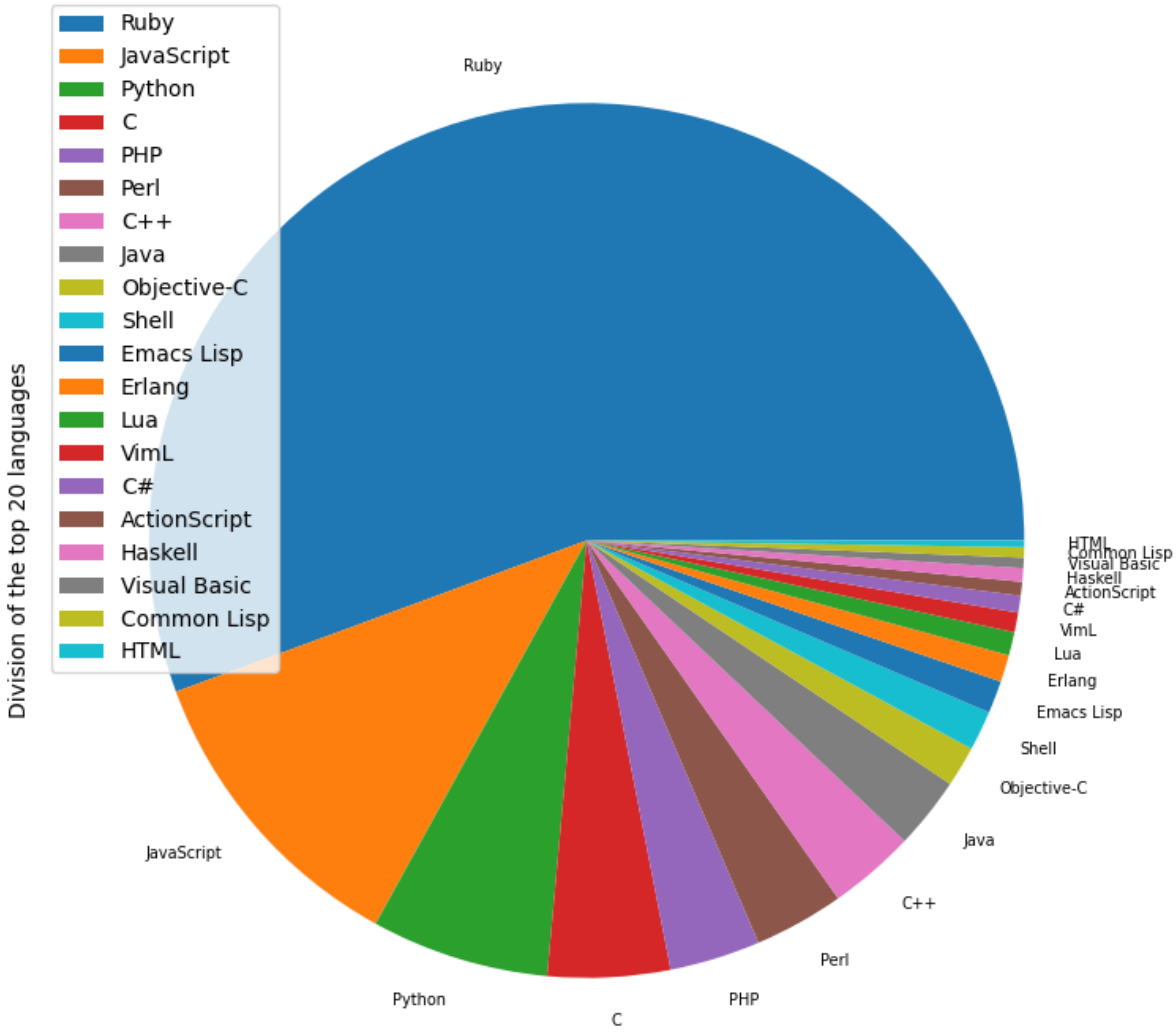
In [220…
```
plt.figure()
lang[:10].plot(kind='barh',figsize=(8,5),grid=True, label='Number of repositories',
```

Out[220]:
```
<Axes: title={'center': 'No of repositories vs Top 10 Languages'}, ylabel='languag
e'>
```



In [224…
```
lang[:20].plot.pie(label="Division of the top 20 languages",fontsize=7,figsize=(9,9
```

Out[224]:
```
<Axes: ylabel='Division of the top 20 languages'>
```



Most Popular Topics of repositories:

In [228…
```
top=test['topics'].value_counts()
top.head(16)
```

Out[228]:
```
topics
[]                                                        23659
['perl', 'perl-module']                                       5
['perl', 'perl-module', 'static-analysis']                    4
['ruby']                                                      4
['hacktoberfest']                                             3
['grails', 'groovy', 'presentation', 'slides']                3
['openbsd', 'patching']                                       2
['rails-plugin', 's3']                                        2
['google-appengine', 'javascript', 'python']                  2
['dotfiles', 'rbenv', 'tmux', 'vimfiles']                     2
['cran', 'r', 'yaml']                                         2
['clojure', 'http', 'ring', 'routing']                        2
['jquery', 'jquery-plugin', 'profiler']                       2
['database', 'dbase', 'dbf', 'foxpro', 'ruby', 'xbase']       2
['postgresql', 'rails', 'ruby']                               2
['api-wrapper', 'tvdb']                                       2
Name: count, dtype: int64
```

In [229…
```
test[['topics']].replace('[]', np.nan)
```

Out[229]:

	topics
0	NaN
1	NaN
2	['programming-languages', 'rubinius', 'virtual...
3	NaN
4	NaN
...	...
4093	NaN
4094	NaN
4095	NaN
5060	NaN
5633	NaN

23974 rows × 1 columns

In [230…
```
nonull_test = test[['topics']].dropna()
tags_list = nonull_test['topics'].str.split(',')
```

In [231…
```
from wordcloud import WordCloud
```

In [232…
```
initial = nonull_test['topics'].str.split(',')
a = []
for item in initial:
       a = a+item
wc_text = ' '.join(a)

%matplotlib inline
wordcloud = WordCloud(background_color='White',width=1600, height=800).generate(wc_
plt.figure(figsize=(25,10), facecolor='k')
plt.imshow(wordcloud, interpolation='bilinear')
plt.tight_layout(pad=0)
plt.axis("off")
```

Out[232]:
```
(-0.5, 1599.5, 799.5, -0.5)
```

In [185…
```
nonull_test['CountTopics']=0
for i in range(0,489,1):
    nonull_test['CountTopics'].iloc[i] = len(list(nonull_test['topics'].iloc[i].spl
```

In [186…
```
nonull_test['CountTopics'].corr(test['stargazers_count'])
```

Out[186]:
```
0.06183617306408601
```

Conclusion

Following results can be drawn from above analysis

a. The most Popular Repository is Rails-Ruby on Rails with 5432 Stars.

b. The Top 5 languages of the repositories are as Follows
1. Ruby 11105
2. JavaScript 2271
3. Python 1311
4. C 902
5. PHP 668

c. Most of the Repositories dont have any topics defined.

Analysis on 850000 Repositories

d. Most of the Repositories are owned individuallly by the user itself the others are owned by the organizations.
1. User 775102
2. Organization 74892

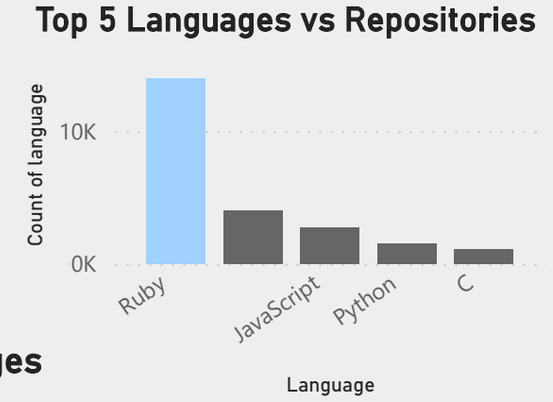
e. The Forked and Un-Forked repositories are as follows
1. Forked 521690
2. Organization 328304

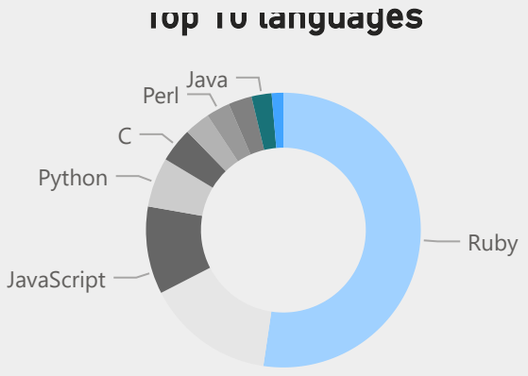
In [ ]:

In [ ]:

GITHUB ANALYSIS FOR 29370 REPOSITORIES

The Top 5 Languages on which the repositories are based on:
The graph shows that the Ruby has the highest repositories based on.



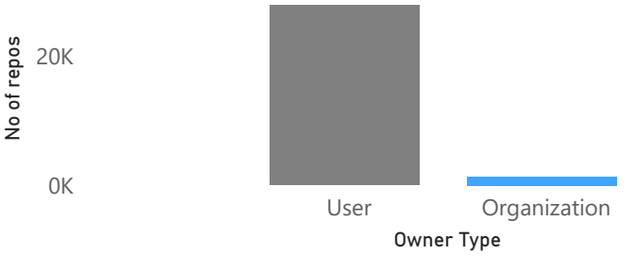


The Division of top 10 languages is shown in the donut chart.

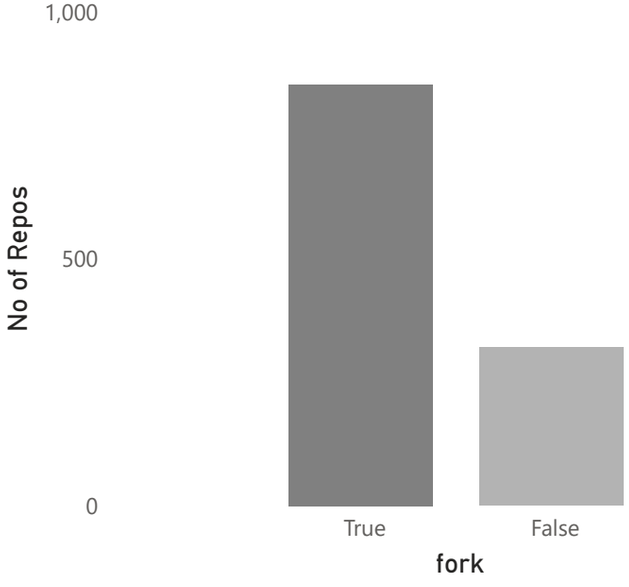
29.37K
Total repositories in Dataset

11.31K
Total Licensed Repos

The bar chart indicates that most of the repositories are owned by the users while some are owned by organizations



The Chart indicates that whether the specific repository is forked or not.Most of the repositories present in the dataset are forked



Most Popular Repository

full_name	stargazers_count
preservim/nerdtree	19075
jekyll/jekyll	47850
git/git	48618
git/git	48627
rails/rails	54314

Ruby on Rails

rubyonrails.org

ruby rails html activerecord framework mvc activejob

Readme
MIT license
Code of conduct
Security policy
Activity
Custom properties
54.3k stars
2.3k watching
21.6k forks