# The GitHub History of the Scala Language

## Github

<https://github.com/Alinda-SH/Stroomdiagram-Scala-uitleg-functies>

Klik op de link om het repository te openen.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Ga naar <> Code.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Download ZIP.

## Project Tasks

df = dataframe variable

### 1. Scala's real-world project repository data

*# import pandas*

Import pandas as pd

<https://www.datacourses.com/how-to-installing-importing-pandas-in-python-1324/>

*# loading csv file as dataframe*

df = pd.read\_csv(‘path/to/csv-file.csv’)

<https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_csv/>

### 2. Preparing and cleaning the data

*# append a dataframe to another dataframe*

df.append(other\_df)

<https://www.geeksforgeeks.org/python-pandas-dataframe-append/>

*# convert the date for the object*

pd.to\_datetime(df[‘column’], format=’format’)

<https://stackoverflow.com/questions/26763344/convert-pandas-column-to-datetime>

format='%Y-%m-%d’

### 3. Merging the DataFrames

*# merge dataframes*

df.merge(other\_df)

<https://realpython.com/pandas-merge-join-and-concat/> c

### 4. Is the project still actively maintained?

# *create a new column that will store the month*

df[‘name\_new\_column’] = df[‘column’].dt.month

<https://datagy.io/create-new-columns-in-pandas/>

*# create a new column that will store the year*

df[‘name\_new\_column’] = df[‘column’].dt.year

<https://stackoverflow.com/questions/30405413/python-pandas-extract-year-from-datetime-dfyear-dfdate-year-is-not-wo>

*# Group by the month and year and count*

df.groupby([‘month’, ‘year’]).count()

<https://stackoverflow.com/questions/38792122/how-to-group-and-count-rows-by-month-and-year-using-pandas>

*# plot the results*

df.plot(kind=’bar’, figsize = (12,4))

<https://www.geeksforgeeks.org/how-to-plot-a-dataframe-using-pandas/>

### 5. Is there camaraderie in the project?

*# import matplotlib*

Import matplotlib as plt

<https://www.tutorialspoint.com/how-to-import-matplotlib-in-python>

*# group by*

df.groupby(‘column’).count()

see earlier link

*# plot the histogram*

df.hist(bins=10)

<https://www.w3resource.com/pandas/dataframe/dataframe-hist.php>

### 6. What files were changed in the last ten pull requests?

*# identify the last 10*

df.sort\_values(‘column’, ascending=False)[:10]

<https://www.w3schools.com/python/pandas/ref_df_sort_values.asp>

<https://www.w3schools.com/python/numpy/numpy_array_slicing.asp>

*# join the two data sets*

pd.merge(df, other\_df, on=’column’)

see earlier link

*# identify the unique files*

Set(df[‘column’].unique))

<https://stackoverflow.com/questions/47933213/list-unique-values-in-a-pandas-dataframe>

*# print the results*

print(df)

<https://realpython.com/python-print/>

### 7. Who made the most pull requests to a given file?

*# file interested in*

file = ‘string’

*# identify the information*

df[df[‘column’] == var]

<https://stackoverflow.com/questions/17071871/how-do-i-select-rows-from-a-dataframe-based-on-column-values>

*# count the number of changes*

df.merge(df, other\_df[[‘column’, ‘column’]], on=’column’)

df.groupby(‘column’).count()

see earlier link

*# print top 3*

print(list(df.nlargest(3, ‘column’).index))

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.nlargest.html>

### 8. Who made the last ten pull requests on a given file?

file = 'string’

*# Select the pull requests that changed the target file*

df[df[‘column’] == file]

see earlier link

*# Merge the obtained results with the pulls DataFrame*

pd.merge(df,other\_df, on='column')

see earlier link

*# Find the users of the last 10 most recent pull requests*

set(df.nlargest(10, ‘column’)[‘column’])

see earlier link

*# Printing the results*

df

### 9. The pull requests of two special developers

%matplotlib inline

*# The developers we are interested in*

authors = ['xeno-by', 'soc']

*# Get all the developers' pull requests*

by\_author = pulls[pulls['user'].isin(authors)]

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isin.html>

*# Count the number of pull requests submitted each year*

counts = by\_author.groupby([by\_author['user'], by\_author['date'].dt.year]).agg({'pid': 'count'}).reset\_index()

<https://www.geeksforgeeks.org/python-pandas-series-agg/>

<https://www.geeksforgeeks.org/python-pandas-dataframe-reset_index/>

*# Convert the table to a wide format*

counts\_wide = counts.pivot\_table(index='date', columns='user', values='pid', fill\_value=0)

<https://datagy.io/python-pivot-tables/>

*# Plot the results*

counts\_wide.plot(kind='bar')

see earlier link

### 10. Visualizing the contributions of each developer

authors = ['xeno-by', 'soc']

file = 'src/compiler/scala/reflect/reify/phases/Calculate.scala'

*# Merge DataFrames and select the pull requests by the author*

by\_author = data[data['user'].isin(authors)]

see earlier link

*# Select the pull requests that affect the file*

by\_file = by\_author[by\_author['file'] == file]

see earlier link

*# Group and count the number of PRs done by each user each year*

grouped = by\_file.groupby(['user', by\_file['date'].dt.year]).count()['pid'].reset\_index()

see earlier link

*# Transform the data into a wide format*

by\_file\_wide = grouped.pivot\_table(index='date', columns='user', values='pid', fill\_value=0)

see earlier link

*# Plot the results*

by\_file\_wide.plot(kind='bar')

see earlier link