**Scala opdracht**

Dit document is ook gepubliceerd als een publieke repositary op github, zie hier de link:

<https://github.com/0967179/Stroomdiagram-Scala-uitleg-functies>

Uitleg hoe je je eigen github data kunt vinden en downloaden:  
- Open je gewenste repository

- Klik op het gewenste bestand

- Klik op view raw (hierdoor word de data gedownload en kan je het eventueel opslaan)

Het script verdeeld in 10 stappen:

1. # Importing pandas

import pandas -> <https://www.statology.org/import-pandas-as-pd>

# Loading in the data   
pd.read\_csv -> <https://www.analyticsvidhya.com/blog/2022/08/different-ways-of-loading-data-using-python/>

1. # Append pulls\_one to pulls\_two

.append -> <https://www.simplilearn.com/tutorials/python-tutorial/append-in-python>

# Convert the date for the pulls object

Pd.to\_datetime() -> <https://www.geeksforgeeks.org/python-pandas-to_datetime/>

1. # Merge the two DataFrames

Pd.merge -> <https://towardsdatascience.com/why-and-how-to-use-merge-with-pandas-in-python-548600f7e738>

1. %matplotlib inline -> <https://www.scaler.com/topics/matplotlib/matplotlib-inline/>

# Create a column that will store the month

.dt -> <https://stackoverflow.com/questions/68732057/assign-new-columns-to-datatable>

# Create a column that will store the year

.dt-> <https://stackoverflow.com/questions/68732057/assign-new-columns-to-datatable>

# Group by the month and year and count the pull requests

.groupby -> <https://towardsdatascience.com/all-about-pandas-groupby-explained-with-25-examples-494e04a8ef56>

# Plot the results

.plot -> <https://www.w3schools.com/python/matplotlib_plotting.asp#>

1. # Required for matplotlib

%matplotlib inline -> <https://www.scaler.com/topics/matplotlib/matplotlib-inline/>

# Group by the submitter

.groupby -> <https://towardsdatascience.com/all-about-pandas-groupby-explained-with-25-examples-494e04a8ef56>

# Plot the histogram

.plot -> <https://www.w3schools.com/python/matplotlib_plotting.asp#>

1. # Identify the last 10 pull requests

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

# Join the two data sets

.merge -> <https://www.w3schools.com/python/pandas/ref_df_merge.asp#:~>:

# Identify the unique files

.unique -> <https://www.javatpoint.com/numpy-unique#:~>:

# Print the results

Print() -> <https://www.programiz.com/python-programming/methods/built-in/print>

1. # Identify the commits that changed the file

== -> [https://learn.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-  
equal?view=msvc-170](https://learn.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-equal?view=msvc-170)

# Count the number of changes made by each developer

.groupby -> <https://towardsdatascience.com/all-about-pandas-groupby-explained-with-25-examples-494e04a8ef56>

.count() -> <https://www.simplilearn.com/tutorials/python-tutorial/count-in-python#>

# Print the top 3 developers

Print() -> <https://www.programiz.com/python-programming/methods/built-in/print>

1. # Select the pull requests that changed the target file

== -> [https://learn.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-  
equal?view=msvc-170](https://learn.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-equal?view=msvc-170)

# Merge the obtained results with the pulls DataFrame

.merge -> <https://www.w3schools.com/python/pandas/ref_df_merge.asp#:~>:

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

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

# Printing the results

Print() -> <https://www.programiz.com/python-programming/methods/built-in/print>

1. %matplotlib inline -> <https://www.scaler.com/topics/matplotlib/matplotlib-inline/>

# Get all the developers' pull requests

.isin -> <https://www.w3schools.com/python/pandas/ref_df_isin.asp>

# Count the number of pull requests submitted each year

.groupby -> <https://towardsdatascience.com/all-about-pandas-groupby-explained-with-25-examples-494e04a8ef56>

.isin-> <https://www.w3schools.com/python/pandas/ref_df_isin.asp#>

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

.agg -> <https://www.w3schools.com/python/pandas/ref_df_agg.asp>

.reset\_index() -> <https://www.w3schools.com/python/pandas/ref_df_reset_index.asp>

# Convert the table to a wide format

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

# Plot the results

.plot -> <https://www.w3schools.com/python/matplotlib_plotting.asp#>

Ax.set\_title -> <https://www.geeksforgeeks.org/matplotlib-axes-axes-set_title-in-python/>

Ax.set\_xlabel -> <https://www.geeksforgeeks.org/matplotlib-axes-axes-set_xlabel-in-python/>

Ax.set\_ylabel -> <https://www.geeksforgeeks.org/matplotlib-axes-axes-set_ylabel-in-python/>

1. # Select the pull requests submitted by the authors, from the `data` DataFrame

.isin -> <https://www.w3schools.com/python/pandas/ref_df_isin.asp>

# Select the pull requests that affect the file

== -> [https://learn.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-  
equal?view=msvc-170](https://learn.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-equal?view=msvc-170)

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

.groupby -> <https://towardsdatascience.com/all-about-pandas-groupby-explained-with-25-examples-494e04a8ef56>

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

.count -> <https://www.simplilearn.com/tutorials/python-tutorial/count-in-python#>

.reset\_index() -> <https://www.w3schools.com/python/pandas/ref_df_reset_index.asp>

# Transform the data into a wide format

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

# Plot the results

.plot -> <https://www.w3schools.com/python/matplotlib_plotting.asp#>