# Importing pandas

**import** pandas **as** pd

importeerd het data packed pandas als pd

# Loading in the data

pulls\_one = pd.read\_csv('datasets/pulls\_2011-2013.csv')

pulls\_two = pd.read\_csv('datasets/pulls\_2014-2018.csv')

pull\_files = pd.read\_csv('datasets/pull\_files.csv')

importeerd drie data sets pull\_files.csv, pulls\_2014-2018.csv, pulls\_2011-2013.csv onder de namen pulls one, pulls two en pull files.

# Append pulls\_one to pulls\_two

pulls = pulls\_two.append(pulls\_one, ignore\_index=True)

# Convert the date for the pulls object

pulls['date'] = pd.to\_datetime(pulls['date'], utc=True)

de twee datasets worden geimporteerd in de verkeerde format om makkelijk met python te werken. Deze moet dus eerst worden omgezet in pythons vorm van datetime

# Merge the two DataFrames

data = pulls.merge(pull\_files, on='pid')

merges de data op basis van de pid collom in de twee data frames

%matplotlib inline

# Create a column that will store the month

data['month'] = data['date'].dt.month

# Create a column that will store the year

data['year'] = data['date'].dt.year

# Group by month\_year and count the pull requests

counts = data.groupby(['year', 'month'])['pid'].count()

# Plot the results

counts.plot(kind='bar', figsize = (12,4))

maakt een collom voor de maand en een kollom voor het jaar van de pull request. Op deze data sets wordt een counts toegepast en dit houd in dat het aantal pull requests wordt bekeken voor overeenkomende maanden en jaren. Hier wordt een histogram geplot voor deze data.

# Required for matplotlib

%matplotlib inline

# Group by the submitter

by\_user = data.groupby('user').agg({'pid': 'count'})

# Plot the histogram

by\_user.hist()

een histogram wordt geplot waar in wordt gezien hoeveel users hoeveel pull requests doen.

# Identify the last 10 pull requests

last\_10 = pulls.sort\_values(by = 'date').tail(10)

last\_10

# Join the two data sets

joined\_pr = pull\_files.merge(last\_10, on='pid')

# Identify the unique files

files = set(joined\_pr['file'])

# Print the results

Files

Deze code identificeerd de laatste tien pull requests van twee data sets. En print het resultaat hiervan.

# This is the file we are interested in:

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

# Identify the pull requests that changed the file

file\_pr = data[data['file'] == file]

# Count the number of changes made by each developer

author\_counts = file\_pr.groupby('user').count()

# Print the top 3 developers

author\_counts.nlargest(3, 'file')

dit telt het aantal veranderingen gemaakt door elke verschillende develiper.

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

# Select the pull requests that changed the target file

file\_pr = pull\_files[pull\_files['file'] == file]

# Merge the obtained results with the pulls DataFrame

joined\_pr = pulls.merge(file\_pr, on='pid')

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

users\_last\_10 = set(joined\_pr.nlargest(10, 'date')['user'])

# Printing the results

users\_last\_10

deze code print de laatste tien gebruikers die het bestand hebben aangepast.

%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)]

# 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()

# Convert the table to a wide format

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

# Plot the results

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

dit is een table waar in gezien kan worden hoeveel pull request er zijn gedaan per jaar van de auteurs xeno-by en soc.

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)]

# Select the pull requests that affect the file

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

# 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()

# Transform the data into a wide format

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

# Plot the results

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

in deze grafiek kan gezien worden hoeveel van de pull requests hebben geleid tot een aanpassing en in welk jaar deze aanpassing heeft plaats gevonden