**[1. Scala's real-world project repository data](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit" \l "1-scalas-real-world-project-repository-data)**

# Importing pandas

**import** pandas **as** 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')

[**2. Preparing and cleaning the data**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#2-preparing-and-cleaning-the-data)

# 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**)

[**3. Merging the DataFrames**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#3-merging-the-dataframes)

# Merge the two DataFrames

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

[**4. Is the project still actively maintained?**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#4-is-the-project-still-actively-maintained)

%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

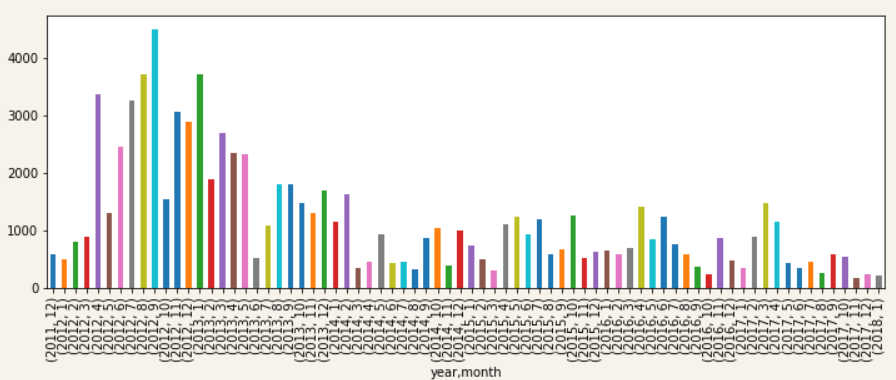
counts = data.groupby(['year', 'month'])['pid'].count()

​

# Plot the results

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

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f55b5ec75f8>



[**5. Is there camaraderie in the project?**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#5-is-there-camaraderie-in-the-project)

# Required for matplotlib

%matplotlib inline

​

# Group by the submitter

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

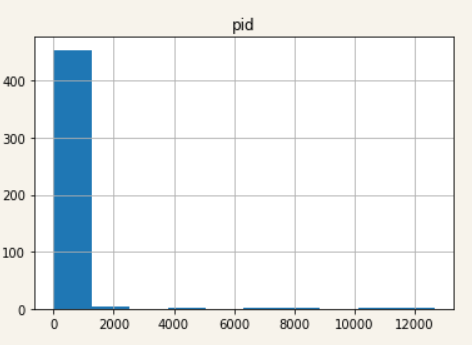
​

# Plot the histogram

by\_user.hist()

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f55b651a358>]],

dtype=object)



[**6. What files were changed in the last ten pull requests?**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#6-what-files-were-changed-in-the-last-ten-pull-requests)

# 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

[**7. Who made the most pull requests to a given file?**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#7-who-made-the-most-pull-requests-to-a-given-file)

# 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]

​

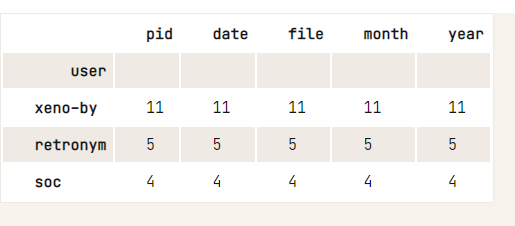
# Count the number of changes made by each developer

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

​

# Print the top 3 developers

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



[**8. Who made the last ten pull requests on a given file?**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#8-who-made-the-last-ten-pull-requests-on-a-given-file)

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

{'bjornregnell', 'retronym', 'soc', 'starblood', 'xeno-by', 'zuvizudar'}

[**9. The pull requests of two special developers**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#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)]

​

# 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

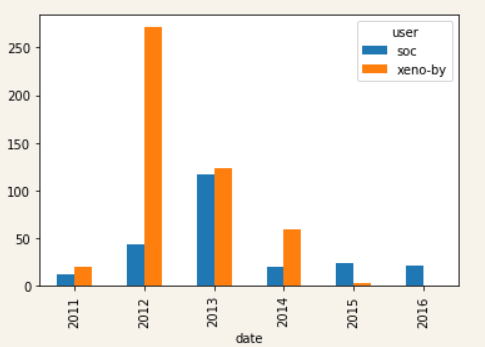
counts\_wide = counts.pivot\_table(index='date', columns='user', values='pid', fill\_value=0)

​

# Plot the results

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

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f55d0cd2470>



[**10. Visualizing the contributions of each developer**](https://app.datacamp.com/workspace/w/34c17ec9-c660-4d08-a8fe-1d99662b302d/edit#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)]

​

# 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')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f55d0d1c7f0>

