



MODULE 07 - Piscine Python for Data Science

Pandas, SQL and Data Visualization

Summary: Today we will help you with data visualization in Matplotlib, Seaborn, and Plotly.

Contents

I	Foreword	2
II	Instructions	4
III	Specific instructions of the day	5
IV	Mandatory part	
	Exercise 00 : Line chart	6
V	Exercise 01 : Line chart with styles	8
VI	Exercise 02 : Bar	10
VII	Exercise 03 : Bar charts	12
VIII	Exercise 04 : Histogram	14
IX	Exercise 05 : Boxplot	16
X	Exercise 06 : Scatter Matrix	18
XI	Bonus part	
	Exercise 07 : Heatmap	20
XII	Exercise 08 : Seaborn	22
XIII	Exercise 09 : Plotly	24

Chapter I

Foreword

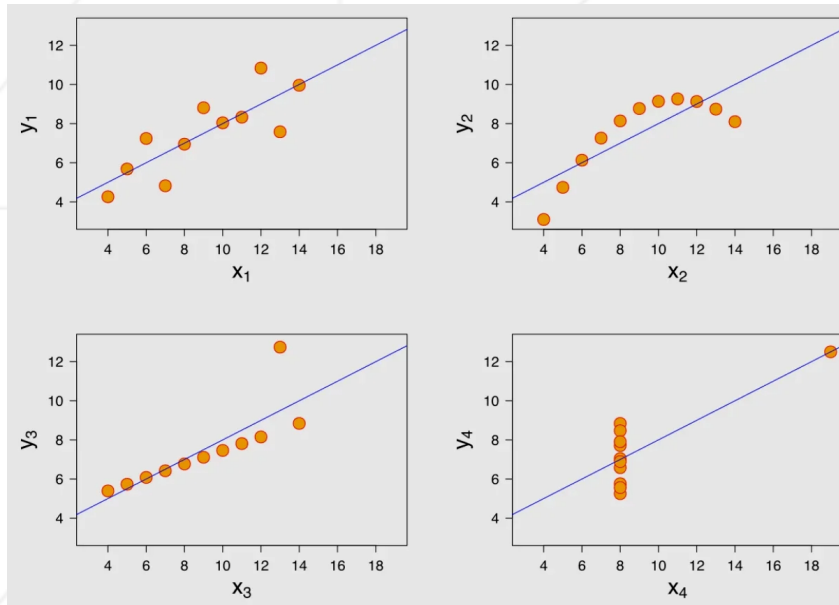
Visualization is good for two reasons. The First, it is very useful when you need to communicate your results to somebody: your employer, colleagues, customers, etc. Second, it is useful building for a better understanding of the data. Here is an example that shows why it is important.

Try to imagine or use a piece of paper to graph the distribution of two variables with the following characteristics.

Property	Value
Mean of x	9
Variance of x	11
Mean of y	7.50
Variance of y	4.125
Correlation between x and y	0.816
Linear regression line	$y = 3.00 + 0.5x$
Coefficient of determination	0.67

Do you think that there is only one way to place the dots?

No, there are several. It is called [Anscombe's quartet](#).



All of the graphs above have the same characteristics. Can you believe it?

Looking at the characteristics alone can be misleading. Use graphs to understand the data better.

Chapter II

Instructions

- Use this page as your only reference. Do not listen to any rumors or speculations about how to prepare your solution.
- Here and further on we use Python 3 as the only correct version of Python.
- The solutions for python exercises (d01, d02, d03) must have the following block in the end: `if __name__ == '__main__':`
- Pay attention to the permissions of your files and directories.
- To be assessed your solution must be in your GIT repository.
- Your solutions will be evaluated by your piscine peers.
- You should not leave any additional files in your directory other than those explicitly specified in the subject. It is recommended that you modify your .gitignore to avoid any accidents.
- When you need to get precise output in your programs, it is forbidden to display a precalculated output instead of performing the exercise correctly.
- Have a question? Ask your neighbor on the right. If that fails, try your neighbor on the left.
- Your reference material: peers / Internet / Google.
- You can ask questions in Slack.
- Read the examples carefully. They may require things that are not otherwise specified in the subject.
- And may the Force be with you!

Chapter III


Specific instructions of the day

- Use Jupyter Notebook to work with your code
- No imports are allowed, except those explicitly mentioned in the section “Authorized functions” of the title block of each exercise
- You can use any built-in function, as long as it is not prohibited within the exercise
- Save and load all the required data in the subfolder data/

Chapter IV

Mandatory part

Exercise 00 : Line chart

	Exercise 00
Line chart	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <i>00_line_chart.ipynb</i>	
Allowed functions : <code>import pandas as pd, import sqlite3</code>	

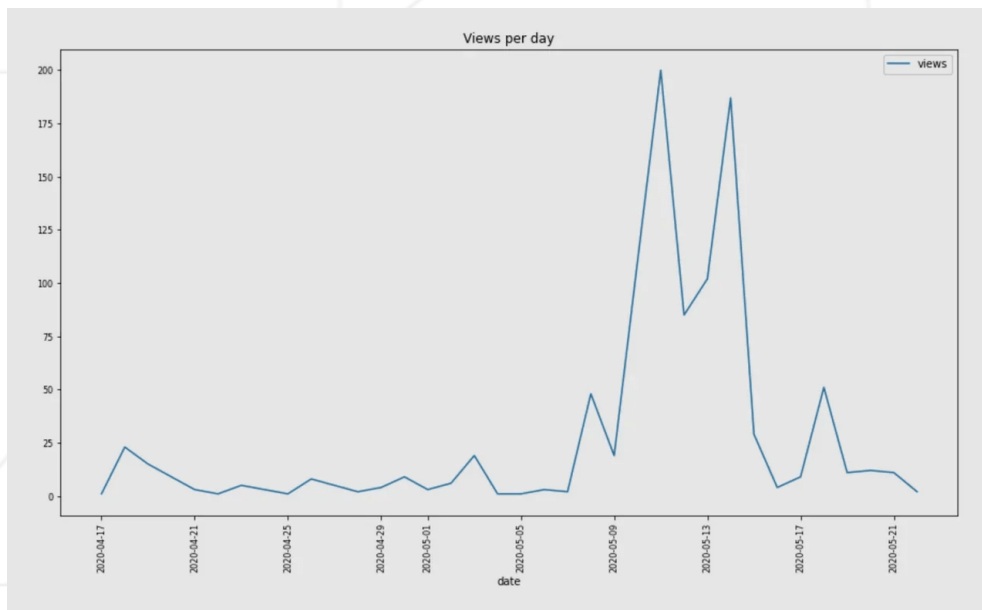
Today, you will work with the same datasets that you used on the previous day. We will try to understand the data about how the students of the educational company behave. You will use Pandas and SQL again to sharpen your skills and use various libraries for data visualization in Python: Matplotlib, Seaborn, and Plotly.

As usual, let us start with something simple. If you have not drawn a graph in Python, it is time to do it for the first time.

Remember how we analyzed the newsfeed page? Did you wonder how often the page was visited in time?


- make a connection to [the database](#) (it is the same as the previous day)
- run a query that gets the datetime from the pageviews table, selecting only the users and not the admins
- using Pandas, create a new dataframe where the visits are counted and grouped by date
- using Pandas method `.plot()`, create a graph
 - the size of the font should be 8
 - the size of the figure is (15,8)
 - the graph must have the title Views per day

- notice the rotation of xticks on the graph below
- close the connection to the database



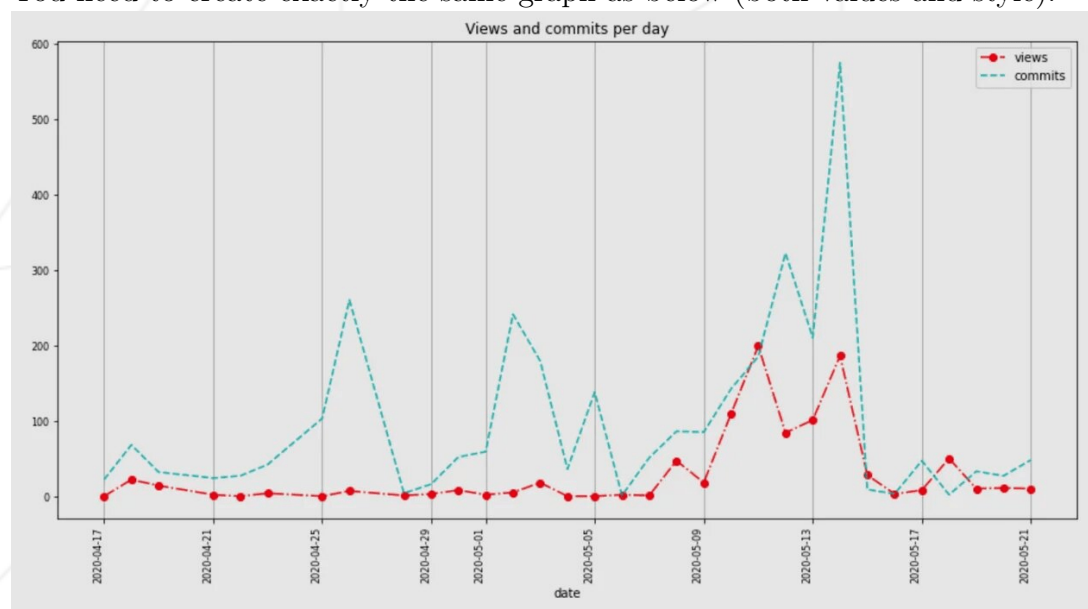
Chapter V

Exercise 01 : Line chart with styles

	Exercise 01
Line chart with styles	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <i>01_line_chart_styles.ipynb</i>	
Allowed functions : <code>import pandas as pd, import sqlite3</code>	

Cool! Remember that we have the data about the commits? Wouldn't it be cool to draw both of the metrics in time on the same graph? What if we will see some patterns?

You need to create exactly the same graph as below (both values and style):




- analyze only the users and not the admins
- analyze only the dates when there were both views and checker commits
- use size of the font should be 8
- the size of the figure is (15,8)

- at the end of your Jupyter Notebook create a markdown cell and insert the question: “How many times was the number of views larger than 150?” Insert: “The answer is ____”. Put the number in the text instead of the underline.

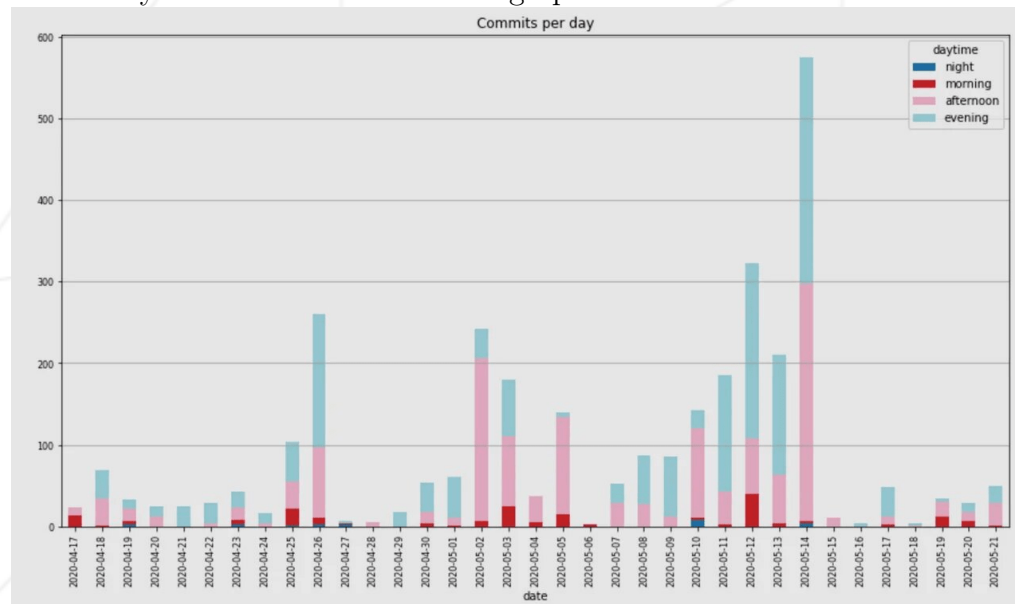
Chapter VI

Exercise 02 : Bar

	Exercise 02
Bar	
Turn-in directory : <i>ex02/</i>	
Files to turn in : 02_bar_chart.ipynb	
Allowed functions : <code>import pandas as pd, import sqlite3</code>	

We have another question for you to answer: when do our users usually commit the labs: in the night, morning, afternoon, or evening? And how has it changed over time?

Do what you need to do to create a graph like this:




- analyze only the users and not the admins
- the fontsize and the figsize are still the same
- night is from 0:00:00 to 03:59:59, morning is from 04:00:00 to 09:59:59, afternoon is from 10:00:00 to 16:59:59, evening is from 17:00:00 to 23:59:59

- choose a palette that you really enjoy, you do not have to replicate it from the graph above
- at the end of your Jupyter Notebook, create a markdown cell and insert the questions:
 - “When do our users usually commit the labs: in the night, morning, afternoon, or evening?”, the answer is the two most common periods.
 - Which day has:
 - * the most number of commits
 - * and at the same time, the number of commits in the evening is higher than in the afternoon?

The answer is the date of that day.

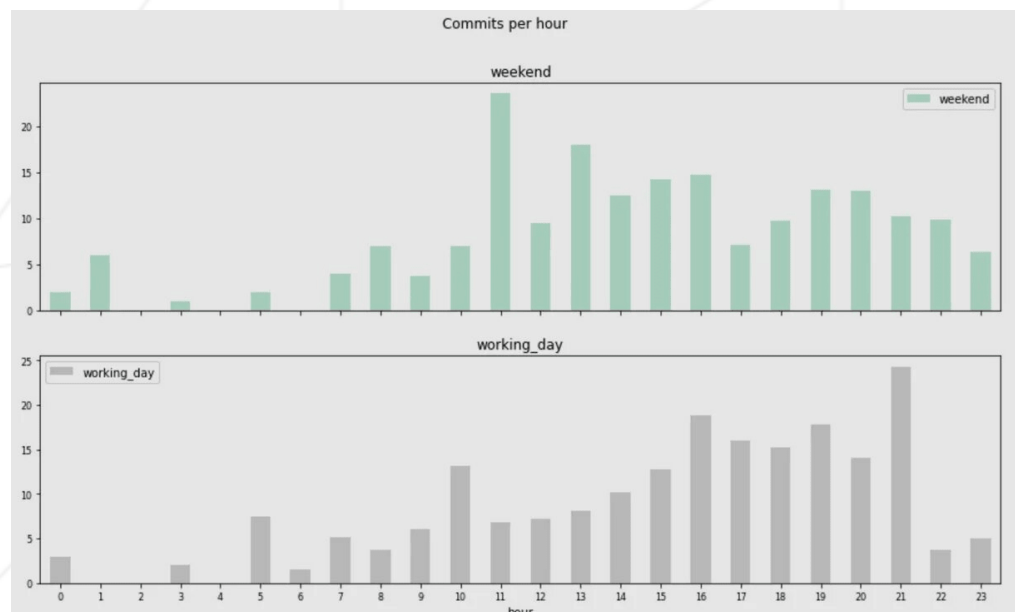
Chapter VII

Exercise 03 : Bar charts

	Exercise 03
Bar charts	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <i>03_bar_charts.ipynb</i>	
Allowed functions : <code>import pandas as pd, import sqlite3</code>	

What if the average number of commits is different when it is a working day or weekend?

Do what you need to do to create a graph like this:




- analyze only the users and not the admins
- the fontsize and the figsize remain the same
- for each hour, calculate the average number of commits on working days and on weekends (if there were no commits in an hour, do not use it to calculate the

average) use these values for your graph, for example: Mon, 17-18: 5 commits, Tue, 17-18: 6 commits, Wed, 17-18: 7 commits

- choose a palette that you really enjoy, you do not have to replicate it from the graph above
- at the end of your Jupyter Notebook, create a markdown cell and insert the question
 - “Is the dynamic different on working days and weekends?”, for the answer include the hour when the number of commits is the largest during working days and the hour when it is the largest during the weekend.

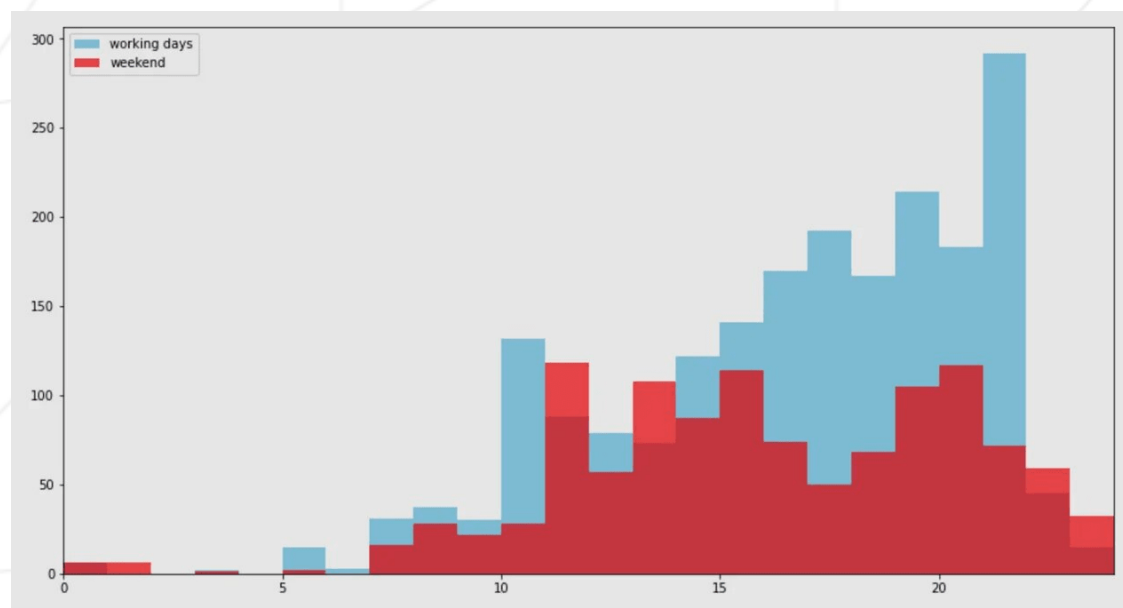
Chapter VIII

Exercise 04 : Histogram

	Exercise 04
Histogram	
Turn-in directory : <i>ex04/</i>	
Files to turn in : 04_histogram.ipynb	
Allowed functions : <code>import pandas as pd, import sqlite3, import matplotlib.pyplot as plt</code>	

In the previous exercise, you had to draw a distribution grouping the values using Pandas. Wouldn't it be nice if we could draw it in a more automatic way? Well, we can. But we have to use another type of visualization – histograms. This time, we will not use the averages. We will use the absolute numbers of commits and will compare them during working days and weekends.

Do what you need to do to create a graph like this:




- analyze only the users and not the admins

- create two lists of values (for working days and for weekends) for the histogram input
- the figsize is still the same, you can choose the fontsize as well as the color palette
- use a level of transparency for the histogram in front equal to 0.7
- at the end of your Jupyter Notebook, create a markdown cell and insert the question: “Are there hours when the total number of commits was higher on weekends than on working days?” In your answer, put the top-4 examples.

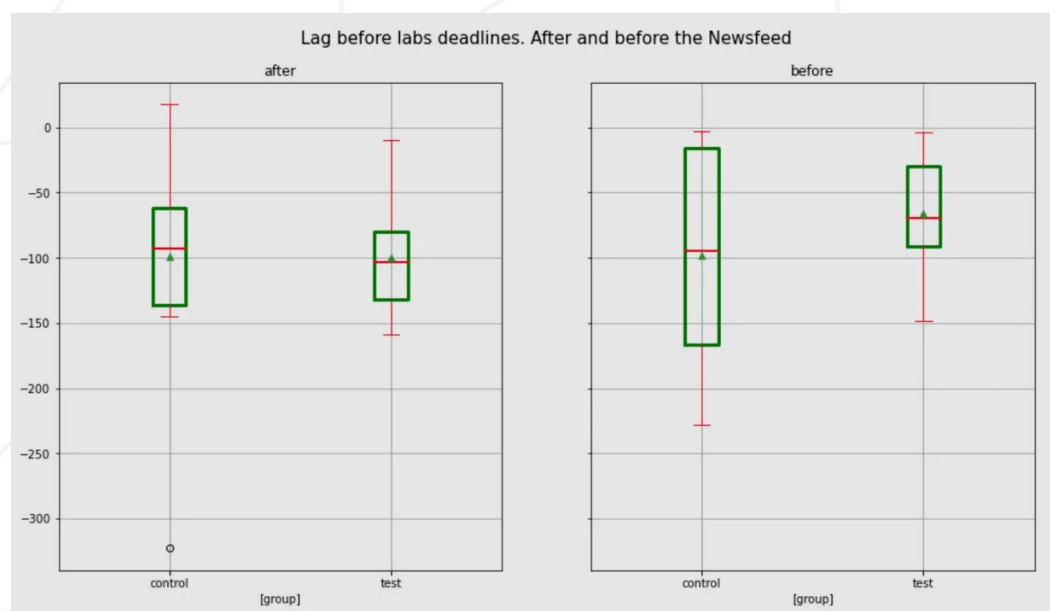
Chapter IX

Exercise 05 : Boxplot

	Exercise 05
Boxplot	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <i>05_boxplot.ipynb</i>	
Allowed functions : <code>import pandas as pd, import sqlite3, import matplotlib.pyplot as plt</code>	

Remember how we tried to figure out if the newsfeed affected the behavior of the test and control users? Last time, we just calculated the average values. But do we know something about the variances? What if it changed too? What if we had some outliers? To answer those questions it may be handy to draw a boxplot.

Do what you need to do to create a graph like this:




- use the data from [the file](#), read it to a dataframe and make any modification that you may find useful to solve the task

- the figsize is still the same, you can choose whatever fontsize you like
- the color palette should be the same as in the example
- the fontsize of the title is 15
- the width of the box lines is 3, the width of the median lines is 2
- at the end of your Jupyter Notebook, create a markdown cell and insert the question:
“What was the IQR of the control group before the newsfeed?” In your answer, put the approximate value that you can get just by looking at the graph, round it to the nearest 10

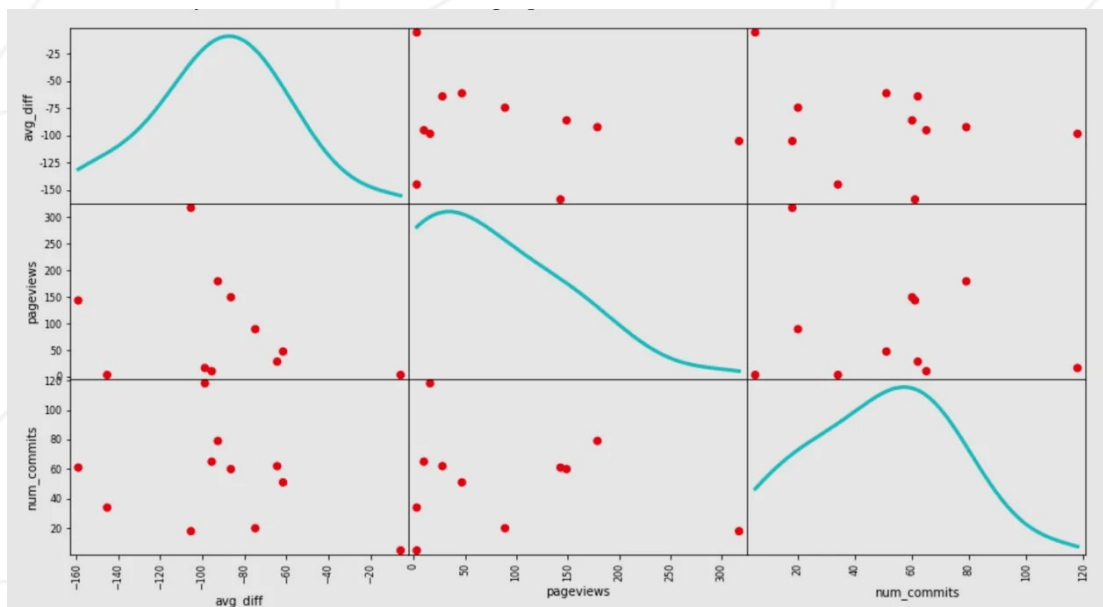
Chapter X

Exercise 06 : Scatter Matrix

	Exercise 06
Scatter Matrix	
Turn-in directory : <i>ex06/</i>	
Files to turn in : 06_scatter_matrix.ipynb	
Allowed functions : <code>import pandas as pd, import sqlite3, from pandas.plotting import scatter_matrix</code>	

Remember how we tried to find out if there was a correlation between the number of visits to the Newsfeed and the average difference between the first commit and the lab deadline? The problem is that the correlation coefficient shows whether there is a linear relationship between the two variables. But what if it is not linear? How can we see that? That's right – by drawing graphs!

Do what you need to do to create a graph like this:




- create a dataframe where each user of the test group has the average difference, number of pageviews and number of commits

- do not take project1 into account for calculations of the average difference and the number of commits
- take the number of commits from the checker table
- the figsize is still the same, you can choose whatever fontsize you like as well as the color palette
- the size of the dots should be 200
- the width of the lines of the diagonal graphs (kde) should be 3
- at the end of your Jupyter Notebook, create a markdown cell and insert the questions:
 - “Can we say that if a user has a low number of pageviews then they likely have a low number of commits?” The answer: yes or no.
 - “Can we say that if a user has a low number of pageviews then they likely have a small average difference between the first commit and the lab deadline?” The answer: yes or no.
 - “Can we say that there are many users with a low number of commits and a few with a high number of commits”? The answer: yes or no.
 - “Can we say that there are many users with a small average difference and a few with a large average difference”? The answer: yes or no.

Chapter XI

Bonus part

Exercise 07 : Heatmap

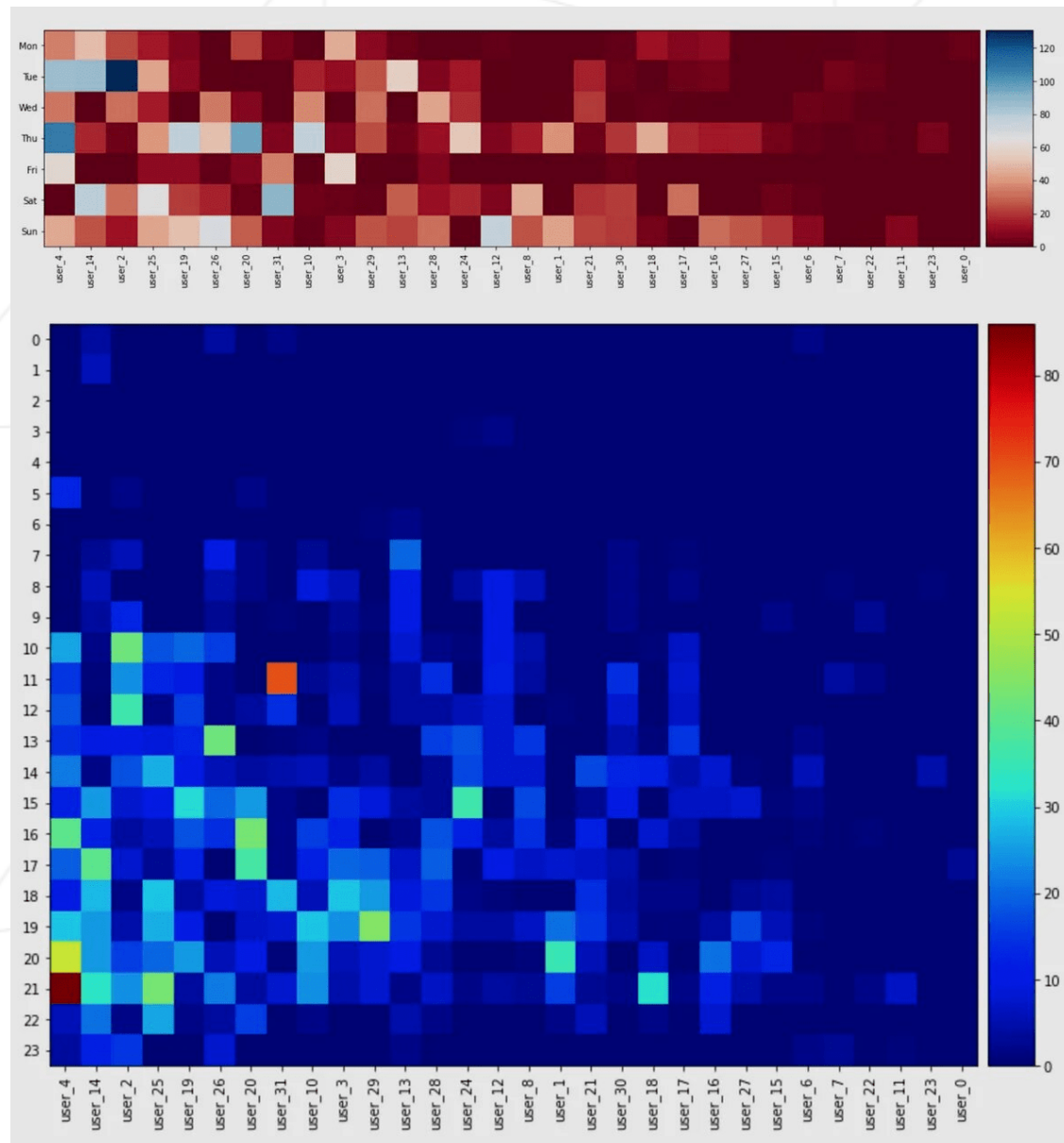
	Exercise 07
Heatmap	
Turn-in directory : <i>ex07/</i>	
Files to turn in : <i>07_heatmap.ipynb</i>	
Allowed functions : <code>import pandas as pd, import sqlite3, import matplotlib.pyplot as plt, from mpl_toolkits.axes_grid1 import make_axes_locatable</code>	

Several exercises back, we wanted to see if there are different patterns for users during working days and weekends. In this exercise, let us find out if there are different patterns for users between different weekdays and between different hours.

- analyze only the users and not the admins
- you can choose the color palette that you like for both of the graphs that you will need to draw in this exercise
- use the table checker for your query
- use absolute values of the commits, not the averages
- sort the dataframes by the total number of commits made by a user
- at the end of your Jupyter Notebook create a markdown cell and insert the questions (answer them looking only at the graphs):
 - “Which user has the most commits on Tue?” The answer: user_.*.
 - “Which user has the most commits on Thu?” The answer: user_.*.


- “On which weekday do the users not like making a lot of commits?” The answer, for example: Mon.
- “Which user at which hour made the largest number of commits?” The answer, for example: user_1, 15.

Do what you need to do to create two graphs like this:



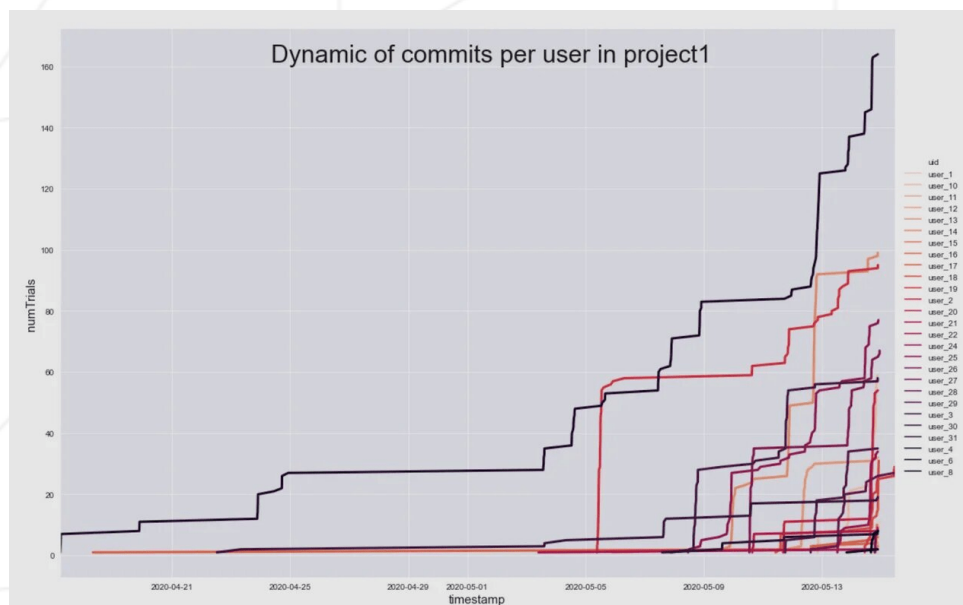
Chapter XII

Exercise 08 : Seaborn

	Exercise 08
Seaborn	
Turn-in directory : <i>ex08/</i>	
Files to turn in : 08_seaborn.ipynb	
Allowed functions : <code>import pandas as pd</code> , <code>import sqlite3</code> , <code>import matplotlib.pyplot as plt</code> , <code>import seaborn as sns</code>	

Ok, sometimes in the previous exercises we ignored project1 in our calculations. The project was a competition. It had longer deadlines and much more commits than ordinary labs had. Let us see the dynamic of commits in this project per user. This time we will use another library for data visualization in Python – Seaborn. In general, it is much easier to create something beautiful in that library.

Do what you need to do to create a graph like this:




- analyze only the users and not the admins
- take into account only logs from the table checker where the status is ready

- you can choose the palette that you enjoy
- the linewidth should be 3
- the background of the graph is gray
- the height should be 10, and the width should 1.5x in relation to the height
- the fontsize of the title should be 30
- the fontsize of the axes labels is 15
- at the end of your Jupyter Notebook create a markdown cell and insert the questions (answer them looking only at the graphs):
 - “Which user was the leader in the number of commits almost all of the time?”
The answer: user_.*.
 - “Which user was the leader for only a short period of time?” The answer:
user_.*.

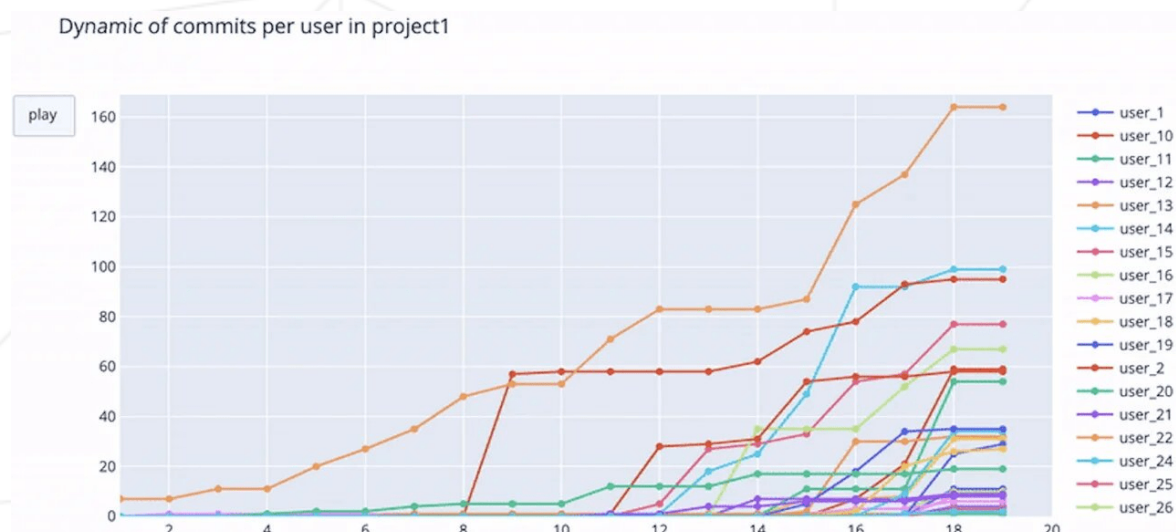
Chapter XIII

Exercise 09 : Plotly

	Exercise 09
	Plotly
	Turn-in directory : <i>ex09/</i>
	Files to turn in : 09_plotly.ipynb
	Allowed functions : <code>import pandas as pd</code> , <code>import sqlite3</code> , <code>import plotly.graph_objects as go</code> , <code>import numpy as np</code>

- Matplotlib and Seaborn are really powerful libraries and you can use them for most of the tasks that you may have related to DataViz. But they do not offer you the functionality of creating interactive charts and animations. And Plotly can help you with that. In this exercise, you will need to create almost the same graph as in the previous exercise but in an animation.

Do what you need to do to create a graph like this:





Video can be found in attachments as (Line_race_plotly.mov)

It is not an easy task, and it is hard to find good and clear tutorials, so use this [link](#) as a reference.