Frankfurt School
of Finance & Management
German Excellence. Global Relevance.

Introduction to Data Analytics
Serie 1.

2022 WS
Prof. Lucas Böttcher

Exercise 1. *Working with pandas.*

In this exercise, we will learn some `pandas` basics that will be useful for your data science course projects.

Hint. In the listings below, we assume that `numpy` *and* `pandas` *have been imported as* `np` *and* `pd`*, respectively.*

(a) Create a pandas series from each of the items below (a list, numpy array, and dictionary).

```
a_list = list("abcdefg")
numpy_array = np.arange(1, 10)
dictionary = {"A":0, "B":1, "C":2, "D":3, "E":5}
```

(b) Combine the following to series into one dataframe.

```
ser1 = pd.Series(list('abcedfghijklmnopqrstuvwxyz'))
ser2 = pd.Series(np.arange(26))
```

(c) Compute the minimum, 25th percentile, median, 75th percentile, and maximum of the following series.

```
state = np.random.RandomState(100)
ser = pd.Series(state.normal(10, 5, 25))
```

(d) Calculate the frequency counts of each unique value in the following series.

```
ser = pd.Series(np.take(list('abcdefgh'), \
np.random.randint(8, size=30)))
```

(e) Reshape the series `ser` into a dataframe with 7 rows and 5 columns.

```
ser = pd.Series(np.random.randint(1, 10, 35))
```

(f) Import the Boston housing dataset, but while importing change the "medv" (median house value) column so that values < 25 becomes "Low" and > 25 becomes "High". What is the proportion of "High" and "Low" values in the dataset?

Hint. The Bostong housing dataset can be downloaded here: https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html

Exercise 2. *Setting up Git.*

You can decide if you want to use Gitlab or GitHub for your course projects. My recommendation is that you use Gitlab because it offers many features for free that are not directly integrated into GitHub. I will primarily focus on Gitlab in the exercises below. (There have also been a few heated discussions on collaborations between GitHub and certain government agencies: https://github.blog/2019-10-09-github-and-us-government-developers/.)

(a) Open gitlab.com and create an account. If you already have an account, login into your existing one. After logging in, create a new project. Choose a project name, a description, and whether you want it to be private or publicly visible.

(b) Install git on your local machine (see https://docs.gitlab.com/ee/topics/git/how_to_install_git/ for further details). Configure your user name and email ID.

```
git config --global user.name "test.user"
git config --global user.email test.user@fs.de
```

(c) Create your first repository "tutorial".

```
mkdir tutorial
cd tutorial
```

(d) Initialize the repository using the following command.

```
git init
```

You will notice that something called the "main" (or "master") appears on the screen. Whenever a git repository is created for the first time, it creates a main (or master) branch.

(e) Now create an empty text file that you name README.md.

Hint. You can also add repository description in your readme file using the git markdown (see https://about.gitlab.com/handbook/markdown-guide/.)

(f) Add the new readme file to the staging area using the following command.

```
git add README.md
```

(g) Now add a corresponding commit comment.

```
git commit -m "added README.md"
```

(h) Go to your GitLab and copy the git remote origin command. It will look similar to `git remote add origin https://gitlab.com/.../tutorial.git`.

(i) Now use the remote and push commands to update your repository.

Hint. You can also directly clone your repository and avoid using `remote` *commands.*

```
git remote -v
git push -u origin main
```

(j*) You may also work through the following tutorials if you are interested in further details: https://docs.gitlab.com/ee/tutorials/.