

# Assignment 1

Prof: Pierre Collin-Dufresne

TA: Florian Perusset

## Problem 1

In this exercise, you will learn how to get data from the WRDS database which we will use more extensively later during the course. After having created an account at <https://wrds-web.wharton.upenn.edu/wrds/>, you have to install the packages *wrds* and *psycpg2* by starting the anaconda prompt window and typing

```
pip install wrds
```

and

```
pip install psycpg2
```

, respectively. If you are using WRDS and Python for the first time, you will have to run the following code (replacing “joe” with your WRDS username):

```
import wrds
```

```
db = wrds.Connection(wrds username="joe")
```

```
db.create pgpass file()
```

 You will be asked to enter your login details while running these lines. After that, you have created a pgpass file on your computer and will not be asked to enter your login details after running `db = wrds.Connection(wrds username='joe')`. You can now start working with the data.

(a) Run the following code:

```
import wrds
```

```
db = wrds.Connection(wrds username="joe")
```

(b) Download daily closing stock returns<sup>1</sup> for the period starting on January 1<sup>st</sup>, 2001 and ending on December 31<sup>st</sup>, 2023 of the following companies: Apple, Goldman Sachs, Microsoft,

---

<sup>1</sup>Prices in the CRSP database do not contain dividends and are not adjusted for stock splits etc. In order to get meaningful data on returns you have to download the variable denoted by “ret.”

Procter and Gamble, and General Electric. (*Hint: Use the command: `aapl=db.raw sql("select date, ret from crsp.dsf where permco in (7) and date>='2001-01-01' and date<='2023-12-31'"`) to get the data for Apple (which has the CRSP's company number 7). GS, MSFT, PG and GE have the permco's 35048, 8048, 21446 and 20792, respectively. You can find details on how to access data from WRDS at <https://wrds-www.wharton.upenn.edu/pages/support/programming-wrds/programming-python/querying-wrds-data-python/>.)*

- (c) From the time-series of daily returns (variable “ret”), reconstruct the time-series of “adjusted” daily prices.
- (d) Save the data panel to a .csv file using the command  
`DataFrame.to_csv()`.
- (e) We can now read the csv file using the command  
`pd.read_csv()`.

## Problem 2

- (a) Using the historical data of daily prices from Problem 1, generate a time-series of daily simple-returns, a time-series of weekly simple-returns, and a time-series of monthly simple-returns. (*Hint: If  $S$  is a **pandas** series, then the method `S.resample(rule).how` might be useful here.*)
- (b) For each frequency, compute summary statistics. (*Hint: If  $S$  is a **pandas** series, then the method `S.describe()` might be useful here.*)
- (c) For each frequency, compute the annualized mean and standard deviation of simple-returns, and print the result on the screen.
- (d) For each frequency, plot the time-series of the annualized mean and std deviation of simple-returns estimated using a rolling one-year window. (*Hint: If  $S$  is a **pandas** series, then the method `S.rolling` might be useful here.*)
- (e) How does the historical variation in the annualized mean and standard deviation estimates compare across sampling frequencies?
- (f) How do the variation in the mean and standard deviation estimates compare across sampling frequencies?
- (g) How do the estimates change as we go through the COVID Crisis window? Do you

observe other crises through estimations?