Part 2 — Workshop 4: Grouping and aggregation

TECH2: Introduction to Programming, Data, and Information Technology

Richard Foltyn
NHH Norwegian School of Economics

October 17, 2025

Exercise 1: House price levels and dispersion

For this exercise, we're using data on around 3,000 observations of house prices and house characteristics from Ames, a small city in Iowa.

1. Load the Ames housing data set from `ames_houses.csv` located in the `data/` folder.

   To familiarize yourself with the data, report the columns present in the DataFrame and then restrict the data to the columns `SalePrice` and `Neighborhood`.

   Check that there are no observations with missing values in the final data set.

2. Compute the average house price (column `SalePrice`) by neighborhood (column `Neighborhood`). List the three most expensive neighborhoods, for example by using `sort_values()`.

   Create a bar chart showing the average sale price for all neighborhoods in descending order.

   Hint: You can create the bar chart by either using pandas's `DataFrame.plot.bar()` or Matplotlib's `bar()`.

3. You want to quantify the price dispersion in each neighborhood. To this end, compute the standard deviation by neighborhood using `std()`. Which are the three neighborhoods with the most dispersed prices?

   Create a scatter plot with the average house price on the x-axes and the standard deviation on the y-axis. Is there a relationship between the average house prices and their dispersion within neighborhood?

   Hint: You can create the scatter plot by either using pandas's `DataFrame.plot.scatter()` or Matplotlib's `scatter()`.

Exercise 2: Determinants of house prices

For this exercise, we're using data on around 3,000 observations of house prices and house characteristics from Ames, a small city in Iowa, to understand how house prices vary with selected house characteristics.

1. Load the Ames housing data set from `ames_houses.csv` located in the `data/` folder, and keep only the columns `SalePrice`, `LotArea`, `YearBuilt`, and `Bedrooms`.

   Restrict your data set to houses with one or more bedrooms and a lot area of at least 100m^2.

2. Compute the average year in which a house was built (using the column `YearBuilt`). Create a new column `New` which takes on the value of 1 if the house was built after the average year of construction (*"new"*), and 0 otherwise (*"old"*).

   What is the average year of construction within these two categories?

3. Create a new column `Rooms` which categorizes the number of `Bedrooms` into three groups: 1, 2, and 3 or more. You can create these categories using boolean indexing, `np.where()`, pandas's `where()`, or some other way.
4. Compute the mean `SalePrice` within each group formed by `New` and `Rooms` (for a total of 6 different categories) using `groupby()`.

   What is the average price difference between an old house with 1 bedroom and a new house with 3+ bedrooms?
5. Create a figure with two subplots arranged in two columns. The left column should contain a bar chart showing the average house price by number of bed rooms (1, 2, 3+) for *old* houses, whereas the right column should show the corresponding bars for *new* houses.

Exercise 3: Inflation and unemployment in the US

In this exercise, you'll be working with selected macroeconomic variables for the United States reported at monthly frequency obtained from FRED. The data set starts in 1948 and contains observations for a total of 864 months.

1. Load the data from the file `FRED_monthly.csv` located in the `data/FRED` folder. Print the first 10 observations to get an idea how the data looks like.

   Keep only the columns `Year`, `Month`, `CPI`, and `UNRATE`. Moreover, perform this analysis only on observations prior to 1970 and drop the rest.
2. The column `CPI` stores the consumer price index for the US. You may be more familiar with the concept of inflation, which is the percent change of the CPI relative to the previous period. Create a new column `Inflation` which contains the *annual* inflation *in percent* relative to the same month in the previous year by applying `pct_change()` to the column `CPI`.

   Hints:

   - Since this is monthly data, you need to pass the arguments `periods=12` to `pct_change()` to get annual percent changes.
   - You need to multiply the values returned by `pct_change()` by 100 to get percent values.
3. Compute the average unemployment rate (column `UNRATE`) over the whole sample period. Create a new column `UNRATE_HIGH` that contains an indicator whenever the unemployment rate is above its average value (*"high unemployment period"*).
   - How many observations fall into the high- and the low-unemployment periods?
   - What is the average unemployment rate in the high- and low-unemployment periods?
4. Compute the average inflation rate for high- and low-unemployment periods. Is there any difference?
5. Use `resample()` to aggregate the inflation data to annual frequency and compute the average inflation within each calendar year.
   - Report the three years with the highest inflation rates.
   - Create a plot that shows the average annual inflation over the sample period.

   Hint: Use the resampling rule `'YE'` when calling `resample()`.