Part 2 — Workshop 3

TECH2: Introduction to Programming, Data, and Information Technology

Richard Foltyn
Norwegian School of Economics (NHH)

October 18, 2024

See GitHub repository for notebooks and data:

https://github.com/richardfoltyn/TECH2-H24

Contents

1 Exercise: House price levels and dispersion

For this exercise, we're using data on around 1,500 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.
2. Restrict the data to the columns `SalePrice` and `Neighborhood`.
3. Check that there are no observations with missing values in this data.
4. Compute the average house price (column `SalePrice`) by neighborhood (column `Neighborhood`). List the three most expensive neighborhoods, for example by using `sort_values()`.
5. You are interested 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?
6. An alternative measure of dispersion is the ratio of the 90th and 10th percentile of the house price distribution. Use the `quantile()` method to compute the P90 and P10 statistics by neighborhood, compute their ratio and print the three neighborhoods with the largest dispersion.

   Hint: The `quantile()` function takes *quantiles* as arguments, i.e., instead of the 90th percentile you need to specify the quantile as 0.9.

2 Exercise: Determinants of house prices

For this exercise, we're using data on around 1,500 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.
2. Restrict the data to the columns `SalePrice`, `LotArea` and `Bedrooms`.
3. Restrict your data set to houses with one or more bedrooms and a lot area of at least 100m².
4. Compute the average lot area. Create a new column `LargeLot` which takes on the value of 1 if the lot area is above the average (*"large"*), and 0 otherwise (*"small"*).

   What is the average lot area within these two categories?
5. 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.
6. Compute the mean `SalePrice` within each group formed by `LargeLot` and `Rooms` (for a total of 6 different categories) using `groupby()`.
7. Compute and report the average price difference between 1 and 2 bedrooms for a house with a small lot area.
8. Compute and report the average price difference between a small and a large lot for a house with 2 bedrooms.

3 Exercise: 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/` folder. Print the first 10 observations to get an idea how the data looks like.
2. Keep only the columns `Year`, `Month`, `CPI`, and `UNRATE`. Moreover, perform this analysis only on observations prior to 1970 and drop the rest.
3. Since pandas has great support for time series data, we want to create an index based on observation dates.

   - To this end, use `to_datetime()` to convert the `Year` and `Month` columns into a date.

     Hint: `to_datetime()` requires information on Year/Month/Day, so you need to create a `Day` column first and assign it a value of 1. You can then call `to_datetime()` with the argument `df[['Year', 'Month', 'Day']]` to create the corresponding date.
   - Store the date information in the column `Date`. Delete the columns `Year`, `Month` and `Day` once you are done as these are no longer needed.
   - Set the `Date` column as the index for the `DataFrame` using `set_index()`.
4. 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.

5. 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?

6. Compute the average inflation rate for high- and low-unemployment periods. Is there any difference?

7. Use `resample()` to aggregate the inflation data to annual frequency and compute the average inflation within each calendar year.

   Which are the three years with the highest inflation rates in the sample?

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