Data Science and Visualization, S2026

Working with data in pandas

2026-02-24

Exercise 1: Libraries

1.1 Use the empty code cell below to install the library pandas. Then comment out or delete the installation code before using the cell to import pandas. If you need help, go to the pandas installation guide and see lecture slides for the syntax used to import pandas

```
# Install pandas,
# comment out the installation code,
# and then import pandas
```

1.2 Check out the websites of the libraries listed below. Find out what the purpose of the libraries are and how to install them. When you have installed them, comment out the code that you used for this and rerun the cell to clear verbose output from your notebook

- Polars: Fast data wrangling library written for effective parallelism
- Matplotlib: Comprehensive library for data visualization
- NetworkX: Library for creation, manipulation, and analysis of complex networks

```
# Install Polars, Matplotlib, and NetworkX
```

Exercise 2: Series and DataFrames

The code cell below creates a list of Danish population counts in number of people from 2000 to 2024. The numbers are taken from https://www.statbank.dk/statbank5a/default.asp?w=1536

```
# Create a list of Danish population counts from 2000 to 2024
population_count = [
```

```
    5330020, 5349212, 5368354, 5383507, 5397640,
    5411405, 5427459, 5447084, 5475791, 5511451,
    5534738, 5560628, 5580516, 5602628, 5627235,
    5659715, 5707251, 5748769, 5781190, 5806081,
    5822763, 5840045, 5873420, 5932654, 5961249
]
```

Use the empty code cell below to do the following:

1. Create a new Series containing the years from 2000 to 2024. Assign the Series to the variable `year`
2. Create a new DataFrame that has the Series `year` and the list `population_count` as its columns. Assign the DataFrame to the variable `DK_POP_00_24`
3. Output `DK_POP_00_24`

```
# Create a Series with the years from 2000 to 2024

# Create a new DataFrame with year and population count

# Output DK_GDP_80_24
```

Excercise 3: Loading

In this exercise, you will use the pandas library to explore a classic dataset of passengers on the Titanic. The dataset contains the following columns:

- **PassengerId**: Unique identifier
- **Survived**: Survival status (0, 1)
- **Pclass**: Ticket class (1, 2, 3)
- **Name**: Passenger name
- **Sex**: Gender (male, female)
- **Age**: Age in years
- **SibSp**: Number of siblings/spouses aboard
- **Parch**: Number of parents/children aboard
- **Ticket**: Ticket number
- **Fare**: Passenger fare in GBP
- **Cabin**: Cabin number
- **Embarked**: Port of embarkation

Download the file *titanic.csv* from Moodle and save it in the same folder as this notebook. Then use the empty code cell below to read it as DataFrame object and assign it to the variable `df`. Finally, find use a DataFrame method to find out how many rows and columns the data has and print the results

```
# Load the titanic data

# How many rows and columns does df have?
```

Exercise 4: Selecting and filtering

Use the `loc` operator and Boolean indexing to answer the questions listed below. Print your answers to the questions

1. In which cabin did the passenger with the name Dodge, Master. Washington sleep?
2. What was the last name of the oldest passenger on the Titanic?
3. How many passengers embarked the Titanic in Cherbourg?
4. How many passengers aged 60 years or older survived the sinking of the Titanic?
5. How many men under the age of 30 bought a ticket for 50 GBP or more?
6. What was the last name of the youngest woman with a ticket price below 10 GBP to survive the sinking of the Titanic?

```
# In which cabin did the passenger
# with the name Dodge, Master. Washington sleep?

# What was the last name of the oldest passenger on the Titanic?

# How many passengers embarked the Titanic in Cherbourg?

# How many passengers aged 60 years or older survived
# the sinking of the Titanic?

# How many men under the age of 30 bought a ticket for 50 GBP or more?

# What was the last name of the youngest woman
# with a ticket price below 10 GBP to survive
# the sinking of the Titanic?
```

Exercise 5: Cleaning

5.1 Use the empty code cell below to do the following:

1. Find out how many missing values there are on each of the columns in `df`. Print the Series containing the results
2. Replace missing values on the column Embarked with 'Cherbourg'

```
# Compute and output the number of missing values on all columns in df

# Replace missing values on the column Embarked with "Cherbourg"
```

5.2 Use the empty code cell below to do the following:

1. Find out how many duplicated rows there are in `df`. Print the result
2. Remove the duplicated rows while keeping the first copy of each

```
# Compute and output the number of duplicated rows in the df

# Remove duplicates in df while keeping the first copy
```

Exercise 6: Creating and transforming

6.1 Use the empty code cell below to do the following:

1. Create a new column in `df` that contains the number of family members aboard the Titanic for each passenger. The columns SibSp and Parch contain the information needed for this. Label the column FamMen
2. Create a new column in `df` that contains the last name of each passenger. You will need to use the `.map()` method, a built-in string method and define a function to do this. Label the column LastName
3. Create a new column in `df` that contains the deck where each passenger's cabin was located. The cabin number starts with a letter that identifies the deck where the cabin was located. You will to use the `.map()` method and define a function that includes a conditional statement to do this. Label the column Deck

```
# Create a new column containing the number of family members aboard

# Create a new column containing last name

# Create a new column containing deck
```

6.2 Use the empty code cell below to do the following

1. Create a new column in `df` that categorizes passengers by the number of family members they were travelling with. Define categories for passengers travelling alone, passengers travelling with 1-2 family members, and passengers travelling with 3 or more family members. The column FamMem that you created under exercise 6.1 contains the information needed for this. You will need to define a function and use the `.map()` method to do this. Label the new column FamSize
2. Based on the categorical column FamSize that you just created, create 3 new dummy variables for travelling alone, travelling with a small family, and travelling with a large family. Then use the method `.join()` to merge the dummies back onto `df`
3. Use the the function `pd.qcut()` to create a new column in `df` that groups passengers into bins defined by quartiles on the age variable. Label the column AgeQuart

```
# Create a new column containing family size

# Create 3 new dummy variables for family size
# and merge them back on to df

# Create a new column containing age bins defined by quartiles
```

Exercise 7: Merging and concatenating

This exercise will use the DataFrame `DK_GDP_00_24` that you created in exercise 2. The code cell below creates a list of Danish GDP in Danish Kroner from 2000 to 2024. The numbers are taken from https://www.statbank.dk/statbank5a/default.asp?w=1536

```
# Create a list of Danish GDP from 2000 to 2024
GDP = [
    1767252e6, 1783295e6, 1800910e6, 1815208e6, 1872596e6,
    1933235e6, 2000184e6, 2014425e6, 2025536e6, 1927603e6,
    1981158e6, 1988226e6, 1994714e6, 2033469e6, 2076855e6,
    2128263e6, 2195314e6, 2263289e6, 2295506e6, 2334244e6,
    2326592e6, 2501738e6, 2561083e6, 2470753e6, 2535701e6
]
```

Use the empty code cell below to do the following:

1. Create a new DataFrame that has the Series `year` and the list `GDP` as its columns. Assign the DataFrame to the variable `DK_GDP_00_24`. You created the Series `year` in exercise 2
2. Use the function `pd.merge()` to merge the DataFrame `DK_GDP_00_24` to the DataFrame `DK_POP_00_24` on the Year column. You you created the DataFrame `DK_GDP_00_24` in exercise 2. Assign the resulting DataFrame to the variable `DK_POP_GDP_00_24`
3. Download the file *DK_POP_GDP_80_99.csv* from Moodle and save it in the same folder as this notebook. Then read it as DataFrame object and assign it to the variable `DK_POP_GDP_80_99`
4. Stack the DataFrames `DK_POP_GDP_80_99` and `DK_POP_GDP_00_24` using the function `pd.concat()`. Assign the resulting DataFrame to the variable `DK_POP_GDP_80_24`
5. Create a new column in `DK_POP_GDP_80_24` containing Danish GDP per capita for the years 1980-2024. Use the formula $PCAP = GDP/POP$ where PCAP denotes GDP per capita and POP denotes population count
6. Output `DK_POP_GDP_80_24`

```
# Create a new DataFrame with year and GDP

# Merge the DataFrames DK_GDP_00_24 and DK_POP_00_24

# Load DK_POP_GDP_80_99

# Stack the DataFrames DK_POP_GDP_80_99 and DK_POP_GDP_00_24

# Create a new column containing GDP per capita

# Output DK_POP_GDP_80_24
```

Exercise 8: Summarizing

In this exercise we will return to the Titanic data and the DataFrame `df`. We will use the original variables and the variables that you created in exercise 6 to compute summary statistics. As you compute the summary statistics, think about what information they provide about the passengers at the Titanic

8.1 Use the empty code cells below to do the following:

1. Create and ouput a table of summary statistics for the variables Age, Fare, and FamMem
2. Create and output a table of frequencies and relative frequencies of the categories defined by the column Pclass
3. Create and ouput a table of frequencies and relative frequencies of the categories defined by the column Survived

```
# Output a table of summary statistics for Age, Fare, and FamMem
```

```
# Output a table of frequencies and relative frequencies for Pclass
```

```
# Output a table of frequencies and relative frequencies
# of categories defined by Survived
```

8.2 Use the empty code cells below to do the following:

1. Create a table displaying the correlation between Survived and the variables Pclass, Age, Fare, and FamMem. Start by creating a list containing the correlation coefficients. Then create a DataFrame with a single column and 4 rows from this list. Then assign a list containing the string 'Correlation with Survived' as the column label and a list containing the strings 'Pclass', 'Age', 'Fare', and 'FamMem' as the row labels. Outputting this DataFrame will give you a table displaying the correlations
2. Create a contingency table displaying relative frequencies of the categories defined by the column Survived across groups defined by the column Pclass

```
# Output a table displaying table displaying correlations with Survived
```

```
# Output a contigency table displaying of relative frequencies
# of categories defined by Survived across groups defined by Pclass
```

Exercise 9: split-apply-combine

9.1 In the empty code cell below, use the `.groupby()` and `.agg()` methods to create a table displaying number of observations, mean, standard deviation, mininum and maximum of Fare across groups defined by the column Embarked. Assign the labels 'N', 'Mean', 'SD', 'Min', and 'Max' to the columns of the table

```
# Create a table displaying summary statistics
# for Fare across groups defined by Embarked
```

9.2 In the empty code cell below, use the `.groupby()` and `.agg()` methods to create a table displaying number of observations, the frequency of passengers who died and survived, respectively, and the percentage of passengers who survived across groups defined by the columns Pclass and FamSize. Assign the labels 'N', 'Survived', 'Died', and 'Percent survived' to the columns of the table You will need to define at least two custom functions and pass them to the `agg()` method to do this. Think about what information the table provides about which passengers survived the sinking of the Titanic

```
# Create a table summarizing survival status
# across groups defined by Pclass and FamSize
```