Rules

- **Internet and Books are ALLOWED**
- **Name your file as following:** `StudentID_ChineseName/EnglishName_quiz#` **Ex:** `123456789_安瓦_quiz1` **or** `123456789_Anvar_quiz1`
- **Extension of your file or your file type should be** `.py`**.**
- **If MOSS (Measure of Software Similarity) detects that any two files have more than 50% similarity, both students will get 0 for this quiz**
- **If you submit the code within an hour you will get the full score. Otherwise, you will have 24 hours to finish it and you will get 80% of your final score.**
- **Follow the guidelines for functions definition.**
  - **Name your function wrong and get 0**
  - **Return the wrong values and get 0**
- **All of the problems are connected to each other and each subsequent problem will depend on the data you have created before, so you won't be able to skip problems this time.**

In the beginning of your file

- Import pandas using its usual alias.

- Import matplotlib.pyplot using its usual alias.

In [ ]:

```
# Here is the template for your functions
# Change N according to the number of the problem
def problem_N():
    # your code here
    return
```

Quiz 2 Pandas Data Manipulation

Stefano asked you to investigate if the movies were getting shorter over the time. He thinks that they are and is very upset about it. This quiz is for you to determine if he is correct.

Problem 1. Loading data into a dictionary (10 points)

Stefano found some data on the internet, and told you to explore it.

Use provided data to create a dictionary.

- **Create a list of years from 2011 to 2020 and a list durations of the average movie lengths (103, 101, 99, 100, 100, 95, 95, 96, 93, and 90) in minutes.**
- **Create a dictionary with the keys "years" and "durations" and the values set to your lists years and durations.**
- **Return the dictionary**

- **Function should return a dictionary**

In [ ]:

```
# Example
def problem_1():
    # your code here
    return # your variable here
```

Problem 2. Creating a DataFrame from a dictionary (10 points)

- **Create a DataFrame using your dictionary you created in the problem 1.**
- **Return the entire DataFrame.**

- **Pass your dictionary from problem 1 as an argument**
- **Function should return a DataFrame**

Problem 3. Visualization of the data (10 points)

Take a look at the data to see if Stefano's assumptions were correct.

- **Create a line plot of the data with the years column of your DataFrame on the x-axis, and the durations column on the y-axis.**
- **Add the title "Movie Durations 2011-2020" to your plot.**
- **Define the figure size as 12 by 9**
- **Return the x-axis data and y-axis data in this order**

- **Pass your dictionary from problem 2 as an argument**
- **Function should return two Series.**
- **Function should have a flag argument that will:**
  - **hide the plot if set to `False` and only return the values.**
  - **plot the graph if set to `True` and will return the values.**

(DO NOT SAVE AND ATTACH YOUR PLOTS TO SUBMISSION! THIS WILL BE CHECKED AUTOMATICALLY!)

Problem 4. Loading the data from a CSV. (10 points)

After seeing the results, it looks like there is something to the idea that movie lengths have decreased over the past ten years! But having only Stefano's aggregations, you're limited in the further explorations you can perform. There are a few questions about this trend that you are currently unable to answer, including:

- What does this trend look like over a longer period of time?

- Is this explainable by something like the genre of entertainment?

You decide to ask Stefano for the original CSV he used to conduct his analysis. He says that you are stupid but gives you the data. You now have access to the CSV file.

- **Create another DataFrame, this time using the CSV file we provided for this quiz,**
- **Return the DataFrame.**

- **Pass the CSV file's filename as an argument**
- **Function should return a DataFrame**

Problem 5. Filtering for movies! (10 points)

You now have the data. By exemining the first five rows of the new DataFrame, you notice a column `type`. Scanning the column, it's clear there are also TV shows in the dataset. You also realize that `duration` column you might have planned to use seems to represent different values depending on whether the row is a movie or a show (perhaps the number of minutes versus the number of seasons)?

It is time to put your subsetting skills to work.

- **Subset the DataFrame such that only rows where the type is a "Movie" are preserved.**
- **Subset the freshly created DataFrame to preserve only the columns title, country, genre, release_year, and duration.**
- **Return the DataFrame from the previous step**

- **Pass the DataFrame from Problem 4 as an argument**
- **Function should return a DataFrame**

Problem 6. Creating a scatter plot (10 points)

You've read in the raw data, selected rows of movies, and have limited you DataFrame to columns of interest. Try visualizing the data again to inspect the data over a longer range of time.

This time, you are no longer working with aggregates provided by Stefano, but instead with individual movies. So, line plot is no longer a good choice for visualization. Try a scatter plot instead.

- **Using the DataFrame created in Problem 5, create a scatter plot, placing the release_year on the x-axis and the movie duration on the y-axis.**
- **Add a title to your plot: "Movie Duration by Year of Release".**
- **Return the x-axis data and y-axis data in this order**

- **Pass the DataFrame as an argument to your function**
- **Function should return two Series.**
- **Function should have a flag argument that will:**
  - **hide the plot if set to `False` and only return the values.**
  - **plot the graph if set to `True` and will return the values.**

(DO NOT SAVE AND ATTACH YOUR PLOTS TO SUBMISSION! THIS WILL BE CHECKED AUTOMATICALLY!)

Problem 7. Dig deeper (10 points)

This is much more informative than the simple plot we created when Stefano first gave you some data. You can also see that, while newer movies are overrepresented, many short movies have been released in the past two decades.

Upon further inspection, you realize that something else is going on. Some of these films are under an hour long. Filter you DataFrame for movies with a duration under 60 minutes and look at the genres. This might give you some insight into what is dragging down the average.

- Subset DataFrame created in Problem 5 to create a new DataFrame containing only movies that have a duration fewer than 60 minutes.
- Return the first 20 rows.

- Pass the DataFrame from Problem 5 as an argument
- Function should return a DataFrame

Problem 8. Marking non-feature films (10 points)

It looks like many of the films that are under 60 minutes fall into genres such as "Children & Family movies", "Stand-Up Comedy", and "Documentaries". This is a logical result, as these types of films are probably often shorter than 90 minute Hollywood blockbuster.

You could eliminate these rows from your DataFrame and plot the values again. But another interesting way to explore the effect of these genres on your data would be to plot them, but mark them with a different color.

- Initialize an empty list called colors to store different color values.
- Use a for loop to iterate through the rows of the DataFrame created in Problem 5 and append colors to your colors list based on the following conditions:
  - If the genre is Children related, append "red".
  - If the genre is Documentaries related, append "blue".
  - If the genre is Stand-Up related, append "green".
  - If the genre is any other genre, append "black".
- Return the values of your colors list.

- Pass the DataFrame from Problem 5 as an argument
- Function should return a list

Problem 9. Plotting with color (10 points)

- Using the data contained in DataFrame created in Problem 5, plot the same scatter plot as you did in Problem 6, but with a few modifications:
- Color the points on the scatter plot using your colors list you defined in the previous step.
- Add a title "Movie duration by year of release", an x-axis label "Release year", and a y-axis label "Duration (min)".
- Return the x-axis data, y-axis data and colors in this order

- Pass the DataFrame from Problem 5 and List from Problem 8 as you arguments
- Function should return two Series and a list.
- Function should have a flag argument that will:
  - hide the plot if set to `False` and only return the values.
  - plot the graph if set to `True` and will return the values.

(DO NOT SAVE AND ATTACH YOUR PLOTS TO SUBMISSION! THIS WILL BE CHECKED AUTOMATICALLY!)

In [ ]:

```
# Use the following snippet of code in your function for visualization purposes only.
plt.style.use('fivethirtyeight')
fig = plt.figure(figsize=(12,8))
```

Problem 10. Your conclusion (10 points)

Well, as you probably suspected, non-typical genres such as children's movies and documentaries are all clustered around the bottom half of the plot. But we can't know for certain until we perform additional analyses.

- **Provide your answer to the question "Are we certain that movies are getting shorter?" in the form of a string.**
- **Make it a simple "Yes" or "No"**

- **Function should return a string**