Description of Course Project

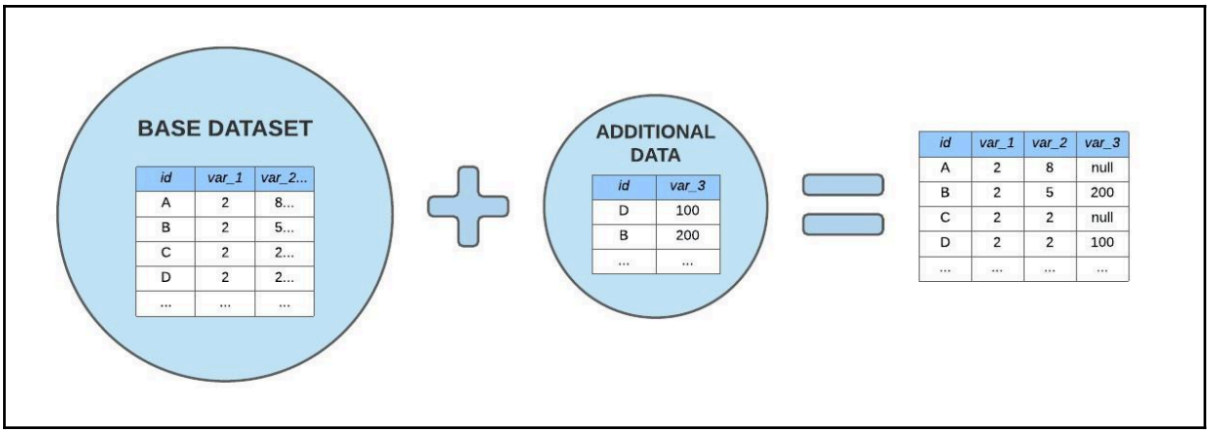
The course project accounts for **50%** of the total course grade. It will be done in groups of 5 students. You will be given the chance to select your own groups on MyCourses.

In the course project, you will use the tools covered in the lectures to analyze data. The project will be handed in as a **Python Jupyter notebook** (IPYNB-file) that includes all of the code used and a detailed written report of the analysis (as Markdown cells). The code (**programming component**) and the reporting (**project report**) will be graded separately, but they should be freely interspersed in the notebook.

The data you use must come in two separate parts (from different sources):

1. A **base dataset** (larger) as a CSV or Excel file
2. Some **additional data** (can be smaller) either manually written in the code or as a file

Your task is to combine these two groups of semi-related data into a single whole and analyze their relationship. The combining must be done as part of the code in your notebook (i.e. the additional data should not be in your base dataset file, it should come from an entirely different source).



You can find the base dataset yourself OR choose one of the predefined datasets (see below). You must find the additional data yourself. The selected data and a short topic plan must be sent in for approval prior to working on the project (a submission box will be in MyCourses on the Course Project -section).

Once your topic plan is approved, your group can get started on the Project Notebook, which is the main focus of the course project. Once the Notebook is submitted, all groups will give short presentations during one of the course lectures. In addition, students must individually conduct a short peer review of a different group's project (Notebook and presentation).

Grading

The project will be primarily graded via a checklist that is provided to you in advance (see below), and secondarily based on the overall quality of analysis and reporting. This means that your overall grade may be reduced, even if the submission superficially meets the requirements of the checklist. The bullet points in the checklist correspond to project requirements, and the final grade will be based on the percentage of these requirements met. Note that requirements can also be met only partially, in

which case a fraction of the points can be given. After the Project Notebook is submitted and your group has done its presentation, your task is to peer-review the project of another group (using the checklist).

Submissions

Topic plan:

Prior to working on the Python Jupyter notebook, your group should send a short topic plan for approval. The topic plan should cover the following:

(1) what specific base dataset and additional data is used (with links),
(2) what is the (presumed) connection between the base dataset and the additional data,
(3) what kinds of visualizations will you expect to use as part of your analysis.

You should carefully read through the "requirements for data" and the overall project, so that you understand what kinds of data to choose. As long as it answers the above three questions, the topic plan needs to be only about **100 words** long.

Project notebook & presentation slides:

The final submission **must be a zip file containing at least the project notebook, main dataset, additional dataset and presentation slides**. The folder structure for the zip file must be done correctly such that the jupyter notebook can be run without any modifications or extra steps, i.e. it finds and reads the data.

The project notebook is the bulk of the project workload, and your group can start working on it as soon as their topic plan is approved. The project notebook is submitted (in the zip file) as a Python Jupyter notebook file (IPYNB-file), and it should include snippets of code interspersed with a written report of the analysis. All data processing and calculations should be in the notebook itself (prior processing of the data before loading it into the notebook is not allowed). The full requirements of the project notebook are listed in the grading checklist. An easy way to work on a Jupyter notebook together is to use the Google Colaboratory, but other methods (such as Git) are allowed, as well.

The notebook should roughly follow the outline of **(1)** loading the data, **(2)** cleaning the data prior to combination (e.g. removing rows with null values), **(3)** combining the data, **(4)** analyzing the data and **(5)** visualizing and reporting the results. The (relatively simple) analysis should include both calculations of basic statistical properties (mean, standard deviation, etc.) and appropriate visualizations of the data (e.g. line charts, histograms, scatter plots with regression, etc.). It is crucial that your visualizations fit the relationship you are analyzing; for example, for time series data, a line chart is generally more appropriate than a pie chart.

Project presentation:

All groups will give a short (approx. 5 minutes) presentation on their project during one of the course lectures (see schedule). The presentation also affects the group's grade, and it should roughly follow along with the project report (i.e. the written portion of the notebook). The presentation's requirements are outlined in the checklist, alongside the requirements of the report.

Peer reviews (individual work):

After projects are submitted, students must *individually* peer-review the project of one other group (using the checklist). Students will be randomly sent the notebook of another group, and each student must fill out a MyCourses form and check off all of the requirements that the other group's project met. If a requirement was met only partially, fractional points (between 0 and 1) can also be granted.

Schedule (*subject to change*)

04.03.2024: Deadline for selecting groups

11.03.2024: Deadline for sending dataset & topic plan to course staff for approval

02.04.2024: Deadline for submitting the project notebook at 11:00 am in the morning. Projects submitted up to 5 days late will have their grade reduced by 1. Anything later than this will result in failing the course.

02.04.2024: Short presentations of course projects (during course lecture - depending on the number of groups there can be a second presentation slot as agreed upon in the lecture)

11.04.2024: Deadline for peer reviews

Predefined base datasets (*OPTIONAL*)

The following are two optional datasets to use as your project's "base dataset" (if you don't want to find one yourself). Note that these datasets are panel data, i.e. they contain both cross-sectional (different passengers) and time series (different dates) data. Keep this in mind when analyzing them.

1. **Predefined option #1: Google Play Store Reviews**
   Link: https://www.kaggle.com/datasets/prakharrathi25/google-play-store-reviews

2. **Predefined option #2: Netflix daily top 10**
   Link: https://www.kaggle.com/datasets/prasertk/netflix-daily-top-10-in-us

3. **Predefined option #3: Health Nutrition and Population Statistics**
   Link: https://datacatalog.worldbank.org/search/dataset/0037652
   (check "Bulk download file (CSV)" or "Bulk download file (Excel)")
   This is a rich data set with a more complicated structure than the other ones so data preparation might take more time but the questions will be more interesting/rewarding to answer.

Requirements for data

All of the following requirements **must be met for a passing grade**; topic plans that do not meet these requirements will not be approved. Kaggle is a good source for CC-licensed datasets.

Requirements for the base dataset:

- Dataset is legally available online, for free, from a trustworthy organization
    - e.g. Creative Commons licensed data
- Dataset is available as a CSV or Excel file
- Dataset has a minimum of 50 non-null rows (ideally much more)
- Dataset has at least two meaningful columns, at least one of which is quantitative
    - i.e. dataset is (at least) two-dimensional
- Dataset is properly cited in the report using an Aalto-approved reference style

Requirements for the additional data:

- Additional data is able to be merged together with the base dataset:
    - The additional data shares key values with the base dataset (e.g. country names)
    - There exists a plausible relationship between the (quantitative or qualitative) information in the additional data and the quantitative information in the dataset
- Additional data does not need to be from a CSV file, but can be. It can also be manually written in the code (as a list/dataframe/dictionary/etc.).
- In general, the additional data does not need to be taken as seriously as the base dataset. The additional data can be gathered from less-serious sources like news articles, blog posts, etc. Still, you are not allowed to simply make up the additional data.

Requirements for data (*once in groups*)

1. Find a dataset that meets the requirements (or select one of the predefined ones).
2. Find additional data that can be meaningfully merged with the base dataset (the additional data can be quite simple). Note that the additional data also has some requirements.
3. Write a short topic plan: What are the sources of data (include links), how might they be connected, and what visualizations will you use? Topic plans should be sent to course staff for approval (see schedule).
4. Create a Jupyter notebook that you can edit together. This is easiest to do with the Google Colaboratory, but it can also be accomplished via Git or other means.
5. Import the needed libraries and read the base dataset into the notebook.
6. Combine the additional data and the dataset within the notebook.
7. Clean the data with the notebook.
8. Calculate basic statistical properties and create appropriate visualizations of your data in the notebook.
9. Add written descriptions, clarifications, and other required project report content into the notebook as Markdown (not Python code) cells.
10. Prepare a short presentation from the finished notebook.

Getting started with Jupyter Notebooks

Jupyter Notebook files are documents created in the Jupyter Notebook App. They contain both code (in this case Python) and formatted text elements (Markdown) divided into cells that can be run independently. The Jupyter Notebook App runs through a web browser interface, and it can either be launched locally or accessed through the internet. You can run the app locally by downloading and

running Jupyter in the Anaconda Distribution. You can also write Jupyter Notebooks online, through the Aalto JupyterHub or through the Google Colaboratory. Google Colab can be useful for the course project, as it allows for the shared editing of Notebook files (similar to how sharing works on Google Docs).

When editing your Notebook, you have two modes: Edit mode (accessed by pressing Enter with a cell selected) and Command mode (accessed by pressing Esc with a cell selected). Edit mode is for changing the contents of the cell, while Command mode is used for keyboard shortcuts. The full list of commands can be found through *Help → Keyboard shortcuts* (Esc+H) on a local notebook, [Left menu] → *Commands* (Ctrl+Shift+C) on the Aalto JupyterHub or *Tools → Command* palette (Ctrl+Shift+P) on Google Colab.

The Notebook cells can be run independently (Ctrl+Enter), so bear in mind that this affects the order in which changes in cells take effect. To run all of the cells in order, an option named Run all can be found from the top drop-down menu (the exact location varies by platform). Here is a full guide for getting started with Jupyter Notebooks . Here is a guide for formatting your text cells with Markdown . Note that the text cells must be "run" (Ctrl+Enter) in order for the formatting to show.

Here is a full guide for getting started with Jupyter Notebooks.

Here is a guide for formatting your text cells with Markdown. Note that the text cells must be "run" (Ctrl+Enter) in order for the formatting to show.

Project Checklist (total 100p + 8 bonus points):

Programming component (40 points):

Basic requirements (4 points):

- ➢ The code is easy to comprehend and meets all of the following requirements:
  - The program is clearly structured.
  - The program's variables are named clearly, concisely and meaningfully.
  - The program includes a sufficient amount of explanation both using Python comments and Markdown.

Reading base dataset (4 points):

- ➢ The program reads its unmodified base dataset from CSV/XLS file(s) and meaningfully utilizes Numpy and Pandas to convert the data into a usable form.
  - Additional data can be provided as part of the code or read from another file.

Combining and cleaning data (8 points):

- ➢ The program inserts the additional data into the base dataset (Pandas DataFrame) as new columns.
  - e.g. for Climate data by country, a "latitude" column could be inserted
- ➢ The program drops any irrelevant columns and incomplete/null data points prior to analysis, if applicable. If the data is already sufficiently clean, its scope should be otherwise narrowed down to a smaller subset of rows.

Calculations (8 points):

- ➢ The program calculates and presents some of the basic statistical properties of the combined data, including mean and standard deviation values (where relevant).
    - These statistical properties should not be simply taken from the data without thought; they should be values that are relevant to the wider analysis.
- ➢ The program follows all logical conventions of statistics and processes data correctly.
    - e.g. combining the percentage rates (of some metric) for two differently-sized groups is not done by directly summing or averaging these two percentage rates → because these two groups are of different size, the resulting combined rate should be a *weighted* average.

Results & presentation (16 points):

- ➢ Data presentation is clear and concise, meeting all of the following requirements:
    - Visualizations/chart(s) with multiple data series have labels on those data series (using a legend, for example).
    - All visualizations/chart(s) have descriptive axis labels.
    - All visualizations/chart(s) have appropriate x and y axis tick label values. (See screenshots for examples).
    - None of the visualizations/chart(s) have overlapping or unreadable text.
    - Visualization/chart type is always appropriate to the data.
        - e.g. a line chart for time series data (and not a bar chart, for example)
    - The program creates at least three visualizations to explain different aspects of the analyzed relationship.
    - The program uses at least two different kinds of visualizations (all visualizations cannot be the same chart type).

Project report (36 points):

Prerequisites (4 points):

- ➢ The report meets all of the following basic requirements:
    - The report is interspersed with the code using the Jupyter notebook Markdown.
    - The report follows a logical structure, such as by following along with the code cells and giving conclusions at the end.
    - The report does not have a significant amount of spelling or grammar errors.

Meta-information (4 points):

- ➢ The report clearly outlines the responsibilities and tasks done by each group member.

Description of data (4 points):

- ➢ The report describes the nature of the used data and defines the relevant columns. All of the following information is given:
    - What are the columns used in analysis, and what type of data do they contain?
    - How does the data need to be cleaned for analysis (if at all)? If no cleaning is needed, how will you narrow down the range of rows analyzed?

Description of functions and charts (8 points):

- ➢ For each function and chart used, the report and the presentation explain why the group chose to use them specifically (and not a function or chart of another type).
    - e.g. for a line graph, why did the group choose a line graph over a scatter plot?
- ➢ For each chart, the report and the presentation state the trend/relationship or other information that can be directly gleaned from it.

Conclusions (12 points):

- ➢ The report describes at least three clear and accurate findings from the analysis. At least one of these findings describes the relationship between the additional data and the relevant base dataset data.
- ➢ The report suggests some possible reasons as to why the relationship(s) exist (or do not exist) in the data.
- ➢ The report reflects on the things the group member(s) learned as a result of this project (what was learned from the data and what was learned in terms of programming).

Sources (see the Aalto guidelines for academic sourcing) (4 points):

- ➢ The report properly sources the data, libraries and other outside information used in the project.

Presentation (max 24 points):

Prerequisites (4 points):

- ➢ The presentation meets all of the following basic requirements:
  - The presentation is roughly in the agreed upon time range
  - The presentation has a visual component that supplements the verbal explanation (e.g. Powerpoint or RISE slideshow).
  - Each team members contribution to the project is clearly outlined during the presentation

Narrative (4 points):

- ➢ The presentation has a clear narrative
  - E.g. it follows a question and is not just a time-line description of the project
  - The used illustrations are fitting to the narrative and are not just a representation of the code and the graphs.
  - It includes the following parts: short introduction, insights and conclusions

Introduction (4 points):

- ➢ The introduction is appropriate in length and detail. Covering:
  - A description of the data sets, including what are the columns used in the analysis, and what type of data do they contain?
  - Describes the aim of the project, which question one wants to answer with the analysis

Insights (4 points):

- ➢ Share insights into the project and the group work process
  - Describe an interesting part of the project, e.g. a challenge and its solution, a fun mistake and its effects, a problem in the collaboration of the group and its resolution, a finding on the side that was interesting.

Conclusions (4 points):

- ➢ The presentation ends with clear conclusions and includes
  - A description of the main finding of the project
  - Describes the answer to the original question or why it was not found
  - The questions to the project are answered in a meaningful way

Presentation material (4 points):

- ➢ The presentation material
  - Fits the presentation in length and details
  - Is easy to read (does not contain many spelling mistakes/clutter on the slides)
  - Helps understand the presentation

Bonus points (max 8 bonus points):

Data processing (4 points):

- ➢ To simplify the program (as a substitute for a self-made function, for instance), the program meaningfully utilizes an additional library that is not covered by the course.

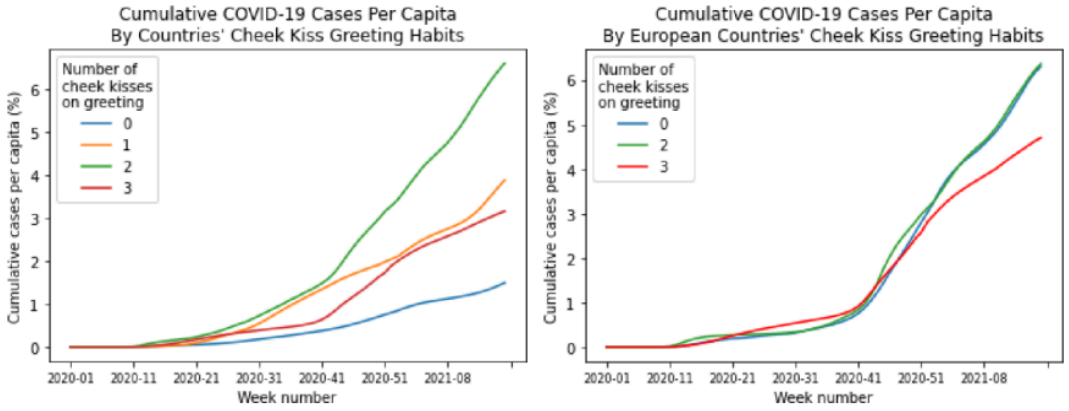
Creative & Thorough work (4 points):

- ➢ The project shows exceptional effort and dedication and goes beyond the guidelines. This can be a creative and interesting topic, very relevant analysis or a fun presentation!

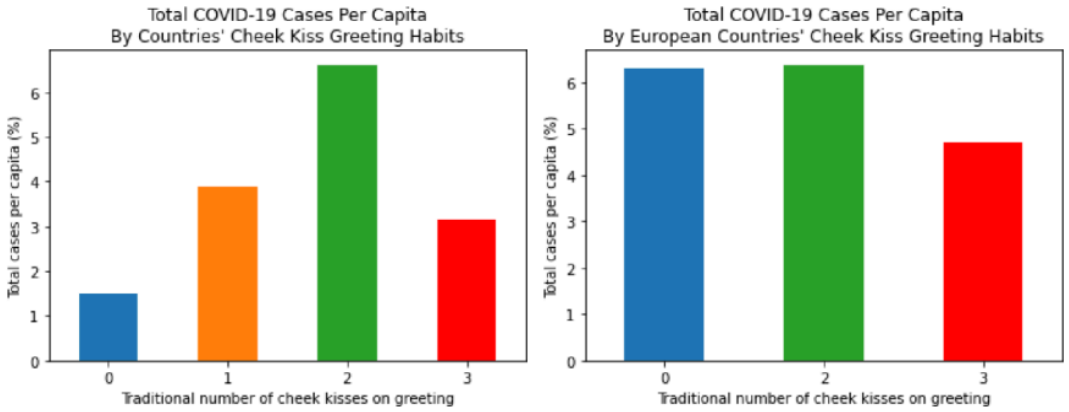
Example data visualizations

Finally, here are some more examples of the kinds of visualizations you can create when doing this project. In this case, the base dataset was Weekly COVID-19 Cases by Country (source: https://opendata.ecdc.europa.eu/covid19/nationalcasedeath/) and the additional data was the amount of cheek kisses* that are traditionally given on greeting in various countries. After some initial processing, a column "kiss_count" was inserted into the DataFrame, with values based on the given row's "country" value. These kiss counts were then used to group data together and create the following plots.

Visualization #1: Cumulative line chart (both globally and filtered to European countries)



Visualization #2: Bar chart (both globally and filtered to European countries)

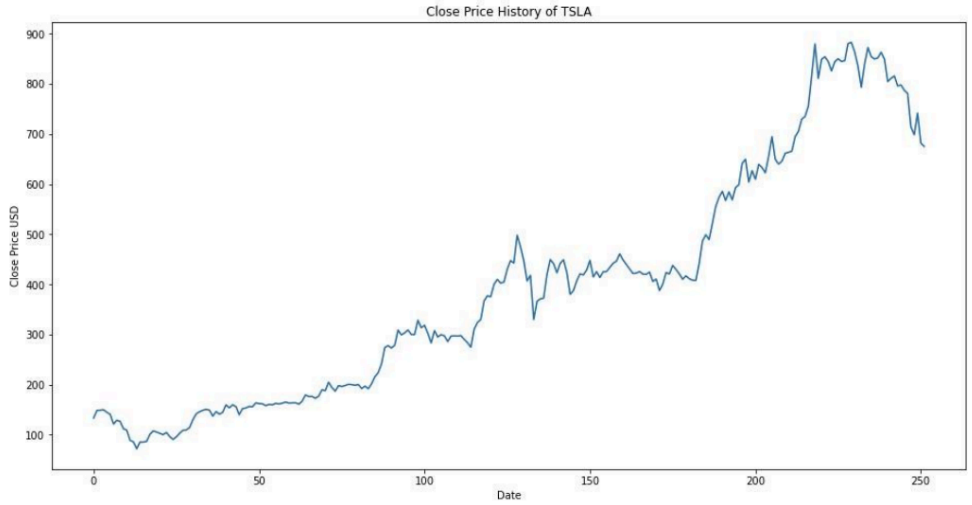


* Note: the creator of these charts is not seriously implying that the number of traditional cheek kisses on greeting is a significant factor in COVID-19 spread. In reality, these greeting habits are correlated with actually significant geographical factors, namely that these countries are mostly in Europe and South America, where COVID-19 spread has been disproportionately large.

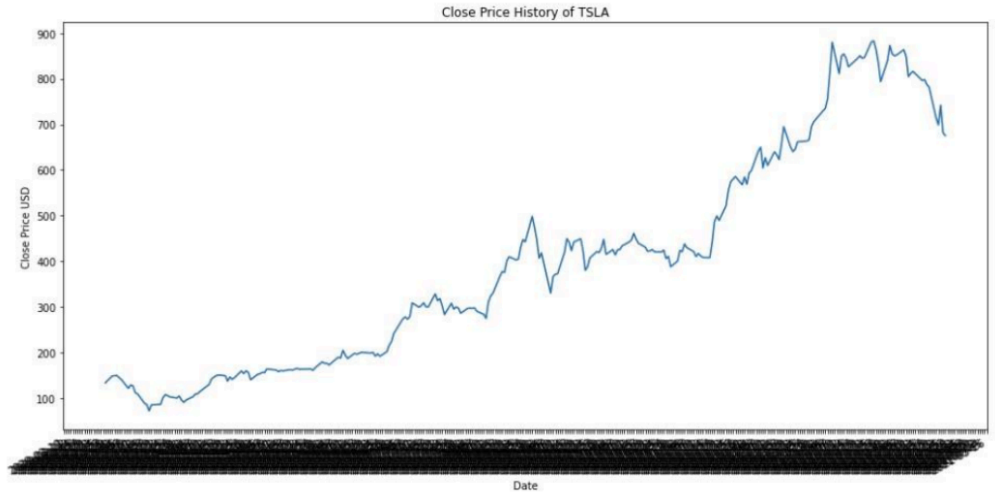
Examples of common mistakes in data visualizations

The following are example screenshots of data visualizations (line charts, in this case) used to demonstrate some common mistakes. Pay attention to these errors when making your own charts.

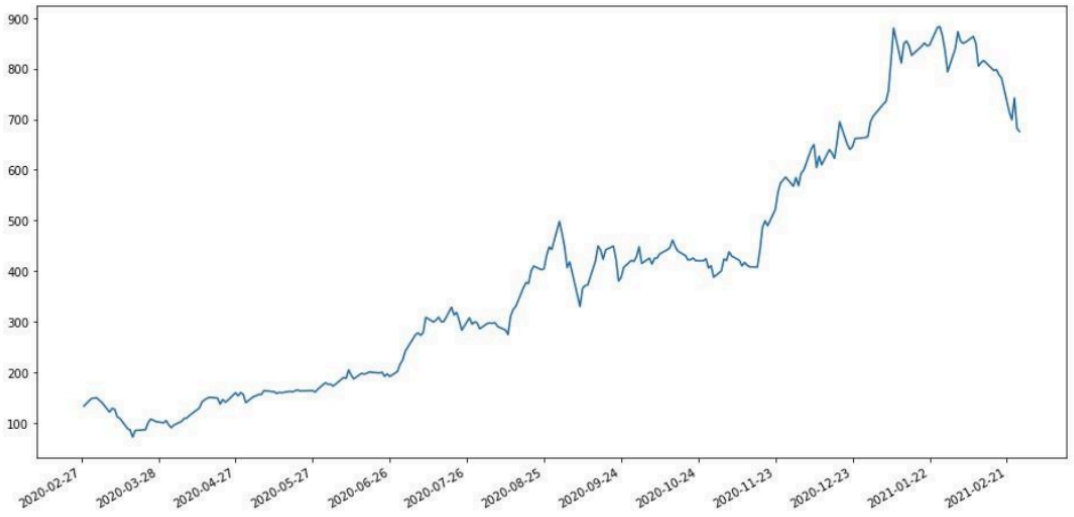
As seen below, the x-axis is not in the correct format. The axis label should be changed to "Number of Days since DD.MM.YYYY" or the tick marks should contain dates.



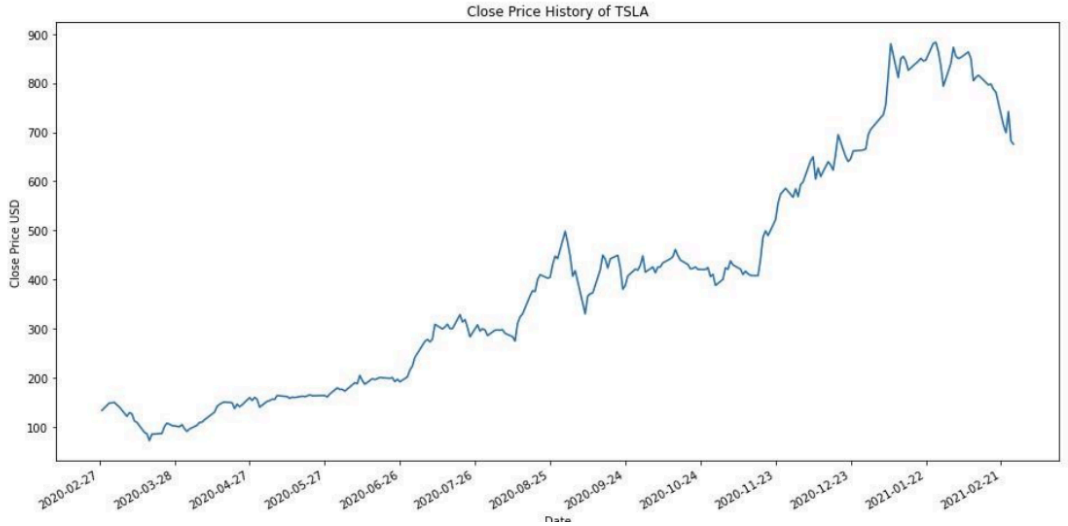
Even though the plot below has the appropriate x and y axes, the ticks on the x-axis are unreadable since the dates are all clogged up/overlapping with each other.



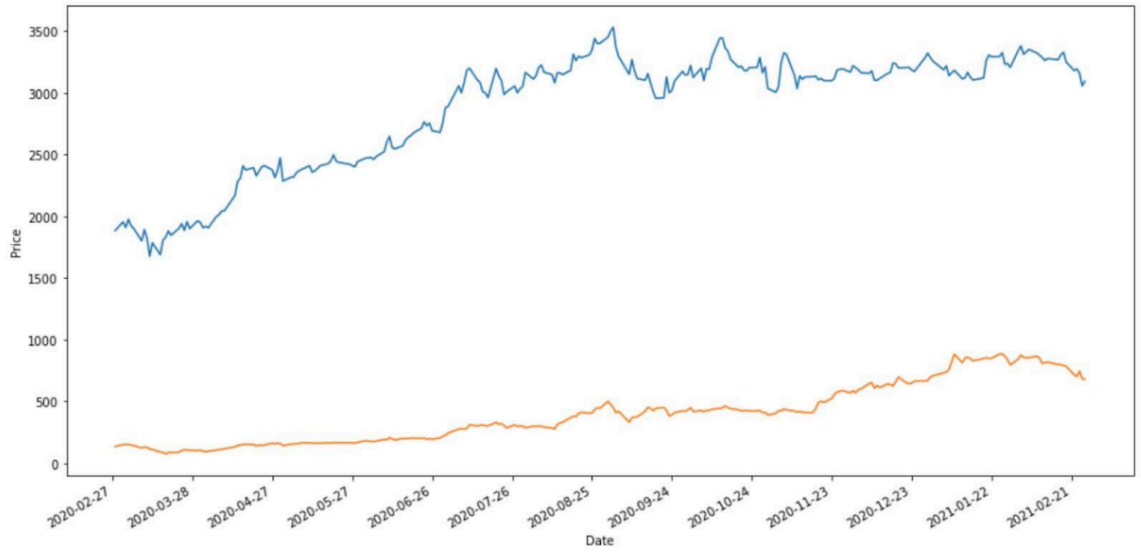
Remember to label your x and y axes and title your plots. The plot below is an example of a plot without a main title or axis titles.



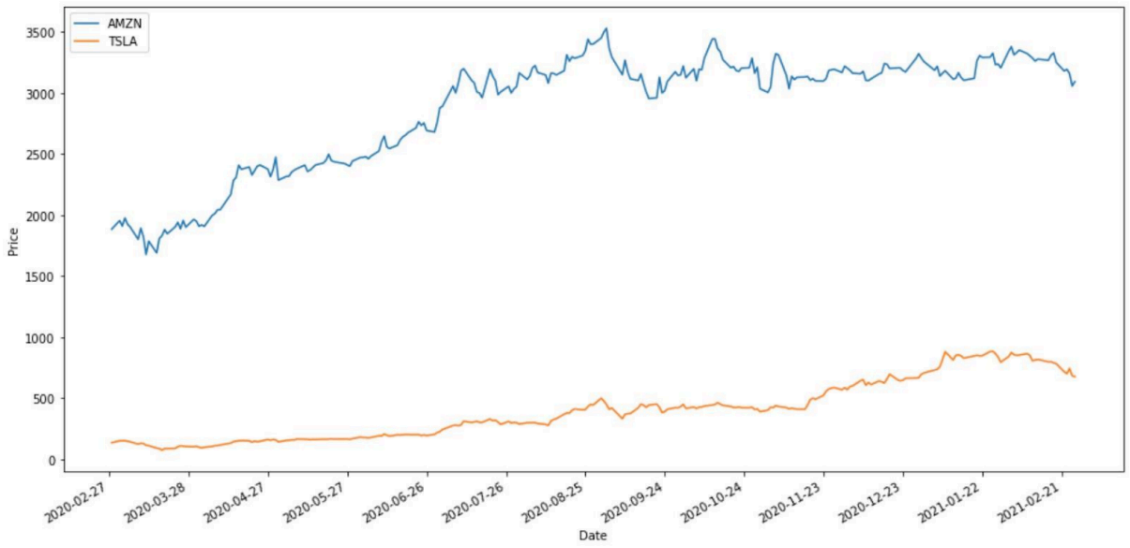
Appropriate Plot:



When visualizing/analyzing more than 1 data series on the same plot, it is important to label them. This can be done using (for example) a legend. The plot below does not indicate what the lines represent.



Appropriate Plot:



Close Price History of AMZN and TSLA