TECHACADEMY

Data Science Project Guide: Netflix

TechAcademy e.V.

Wintersemester 2022/23

Contents

1 Welcome!

Let us start with an overview of what is waiting for you in the next weeks and in this project guide!

If you are a beginner,in the first few chapters you will be introduced to the basics of the `R` and `Python` tracks respectively and you will find helpful explanations to questions you might have in the beginning of your coding journey. There will be a quick introduction to the Data Science track so that you can get started with the project quickly. So let's get started with the basics! You will work on your project in small groups of fellow students. This not only helps you get the project done faster, it also helps make your results even better. Our experience shows: Contrasting university backgrounds and different opinions and ideas will produce the best results. Besides, it is of course more fun to work on a project together than to code alone! The groups can consist of a maximum of four members and you can choose your teammates independently. We explicitly encourage you to collaborate with students from different departments. When submitting your final project, it is important to note: for a certificate, each person must submit the project individually. However, this can be identical within your group. You can get more information at our "Introduction to Coding" Workshop on November 9, 2022 or at our first Coding Meetup on November 23, 2022. This Netflix case study and the associated project guide was developed and written entirely from scratch by TechAcademy's Data Science team. Jessica Weigel and Isabel Schnorr developed the project in `R`, while Lea Karoza, Moritz Schwerdt and Rocky Auer developed it in `Python`.

2 What's Data Science and How Do I Do It?

2.1 What's R and Datacamp Courses?

R is a programming language that was developed by statisticians in the early 90s for use in the calculation and visualization of statistical applications. R is now one of the most widely used programming languages in the field of data science. Code in R does not have to be compiled, but can be used interactively and dynamically. This makes it possible to quickly gain basic knowledge about existing data and to display it graphically. R offers much more than just programming, but also a complete system for solving statistical problems. A large number of packages and interfaces are available, with which the functionality can be expanded and integration into other applications is made possible.

2.1.1 RStudio

Install R and RStudio locally

To get started you should install R and RStudio locally on your computer. Before we try to put in word how to do so: Here's a DataCamp tutorial on how to do that. If you struggle, we will help you with installing at the "Introduction to Coding" Workshop or at our first Coding Meetup. Until then, focus on learning the "hard skills" of programming with the courses on DataCamp. That brings us to your curriculum in the next section!

2.1.2 DataCamp courses and Curriculum

The following list shows the required DataCamp courses for the Data Science with R Track at TechAcademy. As a beginner, please stick to the courses of the "beginner" program. Ambitious beginners can, of course, take the advanced courses afterward. However, it would be best if you worked through the courses in the order we listed them.

The same applies to the advanced courses. Here, too, you should finish the specified courses in the given order. Since it can, of course, happen that you have already mastered the topics of an advanced course, you can replace some courses. If you are convinced that the course does not add value to you, feel free to replace it with one of the courses in the "Exchange Pool" (see list below).

To receive the certificate, both beginners and advanced learners must complete at least 6 courses of the curriculum (6/7 courses). After completing the curriculum and the project's requirements, you will receive your TechAcademy certificate!

Data Science in R Fundamentals (Beginner)

1. Introduction to R (4h)
2. Intermediate R (6h)
3. Data Manipulation with dplyr (4h)
4. Cleaning Data in R (4h)
5. Introduction to Data Visualization with ggplot2 (4h)
6. Exploratory Data Analysis in R (4h)
7. Reporting with R Markdown (4h)

Data Science in R (Advanced)

1. Intermediate R (6h)
2. Data Manipulation with dplyr (4h)
3. Cleaning Data in R (4h)
4. Introduction to Data Visualization with ggplot2 (4h)
5. Text Mining with Bag-of-Words in R (4h)
6. Writing Efficient R Code (4h)
7. Reporting with R Markdown (4h)

Data Science in R (Advanced) – Exchange Pool

- Intermediate Data Visualization with ggplot2
- Interactive Maps with leaflet in R (4h)
- Interactive Data Visualization with plotly in R (4h)
- Multiple and Logistic Regression in R (4h)
- Machine Learning in Tidyverse (5h)
- Introduction to Writing Functions in R (4h)

2.1.3 Helpful Links

- RStudio Cheat Sheets
- RMarkdown Explanation (to document your analyses)
- StackOverflow (forum for all kinds of coding questions)
- CrossValidated (Statistics and Data Science forum)

2.2 What's Python and Datacamp Courses?

Python is a dynamic programming language. You can execute the code in the interpreter, so you do not have to compile the code first. This feature makes Python very easy and quick to use. The excellent usability, easy readability, and simple structuring were and still are core ideas in developing this programming language.

You can use Python to program according to any paradigm, whereby structured and object-oriented programming is most straightforward due to the structure of the language. Still, functional or aspect-oriented programming is also possible. These options give users significant freedom to design projects the way they want and great space to write code that is difficult to understand and confusing. For this reason, programmers developed specific standards based on the so-called Python Enhancement Proposals (PEP) over the decades.

2.2.1 Google Colab, Anaconda and Jupyter

Normally before you can use Python, it must be installed on the computer. In order to make it easier to use Python right out of the box, we use Google Colab as an online Notebook. Using Google Colab will be proficient enough for the TechAcademy DataScience Course. If you want to expand your view, you can circle back to this Introduction and decide Python and Jupyter on your machine to play around with.

Python is already installed on Linux and Unix systems (such as macOS), but often it is an older version. Since there are differences in the handling of Python version 2 – which is not longer supported anymore – and version 3, we decided to work with version 3.6 or higher. One of the easiest ways to get both Python and most of the best known programming libraries is to install Anaconda. There are detailed explanations for the installation on all operating systems on the website of the provider. With Anaconda installed, all you have to do is open the Anaconda Navigator and you're ready to go. There are two ways to get started: Spyder or Jupyter. Spyder is the integrated development environment (IDE) for Python and offers all possibilities from syntax highlighting to debugging (links to tutorials below). The other option is to use Jupyter or Jupyter notebooks. It is an internet technology based interface for executing commands. The big advantage of this is that you can quickly write short code pieces and try them out interactively without writing an entire executable program. Now you can get started! If you have not worked with Jupyter before, we recommend that you complete the course on DataCamp first. There you will get to know many tips and tricks that will make your workflow with Jupyter much easier.

In order to make your work and, above all, the collaboration easier, we are working with the Google Colab platform that contains a Jupyter environment with the necessary libraries. You can then import all the data necessary for the project with Google Drive. We will introduce this environment during our first Coding Meetup. Until then, focus on learning the "hard skills" of programming with your courses on DataCamp. This brings us to your curriculum in the next section.

2.2.2 DataCamp courses and Curriculum

The following list shows the required DataCamp courses for the Data Science with `Python` Track at TechAcademy. As a beginner, please stick to the courses of the "beginner" program. Ambitious beginners can, of course, take the advanced courses afterward. However, it would be best if you worked through the courses in the order we listed them.

The same applies to the advanced courses. Here, too, you should finish the specified courses in the given order. Since it can, of course, happen that you have already mastered the topics of an advanced course, you can replace some courses. If you are convinced that the course does not add value to you, feel free to replace it with one of the courses in the "Exchange Pool" (see list below).

To receive the certificate, both beginners and advanced learners must complete at least 6 courses of the curriculum (6/7 courses). After completing the curriculum and the project's requirements, you will receive your TechAcademy certificate! To receive the certificate, both beginners and advanced learners must complete at least 6 courses of the curriculum (6/7 courses).

Data Science with Python Fundamentals (Beginner)

1. Introduction to Data Science in Python (4h)
2. Intermediate Python (4h)
3. Data Manipulation with pandas (4h)
4. Cleaning Data in Python (4h)
5. Introduction to Data Visualization with Matplotlib (4h)
6. Exploratory Data Analysis in Python (4h)
7. Python for Data Science Toolbox (Part 1) (3h)

Data Science with Python (Advanced)

1. Intermediate Python (4h)
2. Data Manipulation with pandas (4h)
3. Cleaning Data in Python (4h)
4. Introduction to Data Visualization with Matplotlib (4h)
5. Python Data Science Toolbox (Part 2) (4h)
6. Feature Engineering for NLP in Python (4h)
7. Writing Efficient Code with pandas (4h)

Data Science with Python (Advanced) - Exchange Pool

- Interactive Data Visualization with Bokeh (4h)
- Data Visualization with Seaborn (4h)
- Introduction to Data Visualization with Plotly in Python 4h)
- Intermediate Regression with Statsmodels in Python (4h)
- Machine Learning in Scikit-learn 4h)
- Writing Efficient Python Code (4h)

2.2.3 Helpful Links

Official Tutorials/Documentation:

- https://docs.python.org/3/tutorial/index.html
- https://jupyter.org/documentation

Further Explanations:

- https://pythonprogramming.net/
- https://automatetheboringstuff.com/
- https://www.reddit.com/r/learnpython
- https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook

2.3 Your Data Science Project

2.3.1 Prelude

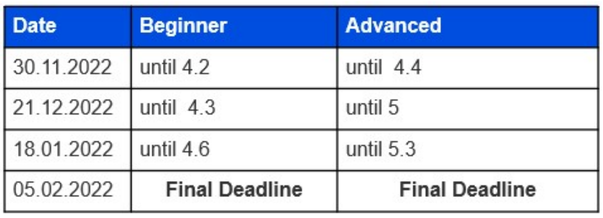
Now that you have learned the theoretical foundation of Data Science in the DataCamp courses, you can put your skills into practice. We have put together a project for you based on real data sets. You can read about the details of this project in the following chapters of this project guide. If you still feel a little unsure about how to start off, there will be a Coding Introduction on November 9, 2022, where we will give you a general sense on how to start off with `Python` or `R`. Of course, we will also describe the project. We will discuss everything you need to know during the first Coding Meetup, which will take place on November 23, 2022. After that, your work on the project will officially begin. You can find the exact project tasks together with further explanations and hints in the following chapters.

2.3.2 Coding Meetups

To give you a little overview on the dates of our Meetups, you can find all the dates in one go here!

- **02.11.2022**, Kick-Off Event
- **09.11.2022**, Introduction to Coding Workshop
- **23.11.2022**, Coding MeetUp 1
- **14.12.2022**, Coding MeetUp 2
- **11.01.2023**, Coding MeetUp 3
- **05.02.2023**, Coding MeetUp 4

To receive the certificate, it is essential that you hand in your project **at least 3 times** during the semester. The following dates are due dates:



Date	Beginner	Advanced
30.11.2022	until 4.2	until 4.4
21.12.2022	until 4.3	until 5
18.01.2022	until 4.6	until 5.3
05.02.2022	**Final Deadline**	**Final Deadline**

2.3.3 Final Deadline

For both your DataCamp courses and the project files hand-in the deadline is **05.02.2023, 23:59**

3 Introduction to Your Project

3.1 Purpose of the Project Guide

This document will guide you through the different steps of your project and will provide you with valuable hints along the way. However, it is not a detailed step-by-step manual because we felt like you needed to develop the skills of coming up with your way of solving different tasks. This method is a great way to apply the knowledge and tools you have acquired through DataCamp. Since data science concepts are independent of specific programming languages, we will describe the general approach in a text chunk. Having understood the bigger picture and starting with the tasks, you will find language-specific tips and tricks in visually separated boxes (`R` -Track : blue-bordered boxes, `Python` - Track : yellow-bordered boxes). Questions might come up, or you might not know how to solve a task right away—but don't worry—this is just part of coding. In those cases, you could ask your fellow team member for help. If they are not able to help, and in the unlikely case that even Google can't help you, the TechAcademy mentors will help you via Slack or directly during the coding meetups. At the end of the project guide, you will find an overview of all tasks that have to be completed, depending on your track (beginner/advanced). You can use this list to check which tasks you still need to complete or which assignments are relevant for your track.

3.2 What is this Project About?

This semester, we will have a look at Netflix data! More precisely, we are first analyzing a very detailed general Netflix data set, compiled on kaggle, after which you will get the chance to look at your own Netflix data. You will find all kinds of information in the data sets - valuable and useless ones. Are you already curious to see for yourself? In analogy to the typical Data Science workflow, we split this project into two parts.

3.3 Exploratory Data Analysis – Getting to Know the Data Set

This first part of the project is structured in a way that lets you get to know the data thoroughly by completing the given tasks one after the other. As a beginner, you can stop after this part because you will have fulfilled the necessary coding requirements for the certificate. However, if this first part inspires you to learn more, we encourage you to also work on the second part. If you get stuck, **Google** and StackOverflow are amazing problem solver (besides the mentors, of course).

3.4 Content-Based Recommendation System

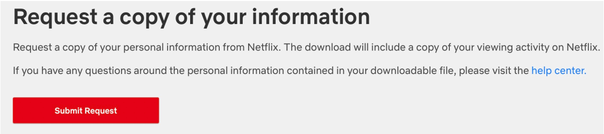
This part is mainly for the advanced TechAcademy participants. If you are a beginner and you were able to complete the first part without too many difficulties, we highly recommend trying to do the second part as well.

4 Exploratory Data Analysis

Before you can dive into the data, you need to do two things. First of all, you need to request your personal data from Netflix, which can take a few days. Therefore, you should ask for your data as early as possible. (If you do not have access to a Netflix account, contact your mentor, and we will provide you with some data)

Download your data

1. Follow this link to request your data, and sign in with your username and password.
2. After that, click on the red box Submit Request.

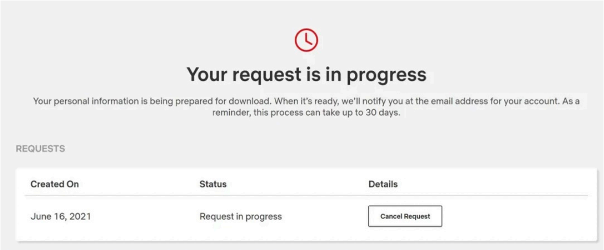


Request a copy of your information

Request a copy of your personal information from Netflix. The download will include a copy of your viewing activity on Netflix.

If you have any questions around the personal information contained in your downloadable file, please visit the help center.

3. Now you need to verify your request via Mail.
4. Once you confirm the request, you will be forwarded to the website and have to enter your password again. If your request is in progress you might have to wait a few days until the data is available to download.



Your request is in progress

Your personal information is being prepared for download. When it's ready, we'll notify you at the email address for your account. As a reminder, this process can take up to 30 days.

REQUESTS

Created On	Status	Details
June 16, 2021	Request in progress	Cancel Request

5. When you receive the mail "Your download is ready," you can download your data by clicking on the red box Get Information. Don't wait too long since you only have 7 days to download it from your account!
6. After confirming your password, you can download your data as a netflix-report.zip file. Unzip the file, open the folder CONTENT_INTERACTION and look for the ViewingActivity.csv file.

However, you can already start with the first exercises of this chapter with data we provide. To start right away, you need to set up your programming environment. This will be the place where the magic happens.

We work with R and RStudio locally which requires a setup. If you haven't installed it yet here is a DataCamp tutorial. If you struggle, we will help you with installing at the "Introduction to Coding" Workshop or at our first Coding Meetup

We recommend using Google Colab for this project since it requires no particular setup, stores its notebooks to your Google Drive, and makes it easy for you to share them with your team members.
As an alternative to Google Colab, you might want to install Jupyter Notebook locally using the Anaconda distribution. We will give you a more detailed step-by-step demo during the “Introduction to Coding Workshop” or our first Coding Meetup.

4.1 Getting started

4.1.1 Discovering the Data

The first big step is importing a general Netflix data set into your coding environment:

- Netflix

Once done, let’s start by looking at the Netflix data set.

- Have a look at the columns of the data set and their “values”
- Do you see any missing values or data entries that are different from the other entries?
- What are the data formats (e.g. data types)?
- Look at the date_added columns. Do you see any critical aspects of the data?

Write down a couple of sentences to these questions, the goal is to show (your readers) what you are seeing. Also, comment on any errors or irregularities which you notice and that could be an issue later on in the project.

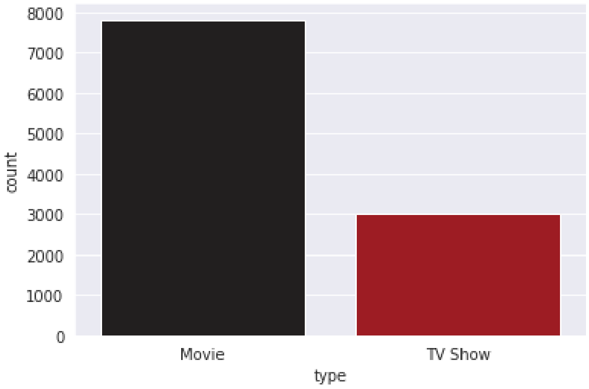
You can import data files by, for examples, using the command netflix_general <- read_csv(“Netflix.csv”). However, it has to be the correct path. Use head() and glimpse() or class()(on a specific colum) to get an overview. Lastly, get a quick summary of the data using {base} R’s summary() or {psych}’s describe() function. To find more ways to generate quick data summaries in R, check this blog post from Adam Medcalf. If you need some additional, more general information on how to import data and different data types, check out this cheat sheet.

You can feed the links to the respective data files above to a method of the pandas package (you might want to specify the index column). Check out the resulting pd.DataFrame instance with the head() method and the dtpyes attribute. You can also dig into a specific column with describe(). Here’s an additional pandas cheat sheet for you to reference

4.1.2 Give some overall statements

Now that you have imported the data set, let’s have a deeper look. Since you already got a feeling by now, it would be interesting to indicate some outstanding features (based on the uncleaned and untransfomated dataset)!

- What's the longest movie (not TV show) included in the dataset? (Look very closely if your result is valid)
- Which country released the most content (movies and tv shows)?
- How many movies and tv shows are included? Try to do a simple plot here so you can see the distribution. If you cannot plot it just now, come back later to finish this.



You could use {dplyr}'s select, filter, count and arrange function to compute the desired outcome to answer the question. If you haven't heard of the {dplyr} package yet, take the respective DataCamp course asap! For the plot, look at {ggplot} and then call geom_bar(). {ggplot2}'s syntax is perhaps a bit harder to learn at the start, but it gives you more plotting benefits in the long run. If you haven't heard of the {ggplot2} package yet, take the respective DataCamp course.

In order to manipulate big amounts of data, Pandas implements different functions to manipulate entries in the Dataframe. Useful functions are: select, filter, value_counts and max. If some of those functions seem new to you, don't scare away from them just now. When it comes to packages and their implementations, there are usually some easy to understand examples that can be found on their wiki page. Keep in mind that the country data is saved as a list, pandas offers the explode function that helps us with that. In order to display graphs, we recommend using seaborn as a package. Seaborn offers a wide variety of different graphs options, it is important that seaborn just translates your input into MatplotLib commands. So if you feel unsatisfied with the customisation options in seaborn, you can always build your own graph using MatplotLib. To display the amount of movies and tv-shows a seaborn.countplot can be used.

4.2 Data Cleaning and Useful Transformations

4.2.1 Date Formatting

From exploring the data in the previous tasks you might have noticed that the time and dates are not in a "date format". In order to fix this, convert the "date_added" and "release_year" column into date format.

You can use the lubridate package to transform the column into a date format. To make things easier, check out the lubridate cheat sheet. Converting the release year might take two steps.

pd.to_datetime() is what I would look at for example. A corresponding DataCamp resource is section 4 in Working with dates and times in Python. Also, the Data Manipulation with pandas course is of great help for the following exercises

4.2.2 More details of the longest movie

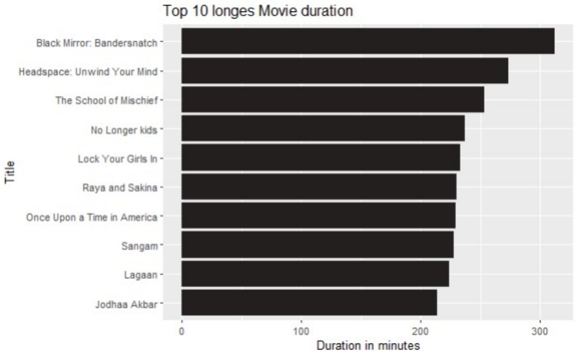
So far, you have converted the column into a more appropriate format for further investigations. Can you also fix the duration of the movies into a numeric format? Can you tell now what the longest movie is? On the concept of distribution, you should also try to compute the mean and standard deviation of the movie duration in minutes.

First off, you might want to create a separate data frame for the movies so the tv shows do not bother you further. Use the filter call here. Next you might have recognized the "min" appendix. This can be deleted with the "gsub" call. With the help of mutate you can convert the column into numeric values and arrange the movies descending. For the mean and standard deviation simply use {base} R's mean() function to compute a mean of any column in R. Similarly, R comes with the sd() function to calculate the standard deviation of a column. You can combine these two functions, e.g., in a {dplyr}'s summarise() verb.

First off, you might want to create a separate data frame for the movies so the tv shows do not bother you further. Try to find a useful criteria in the data that separates series and movies in order to filter out just the movies. To get the numeric value out of the string either a separation by space or match by regex will work. Make sure to convert the string text to a more suitable number type such as integer. Pandas already helps when finding out the standard deviation and mean with the implemented functions std() and mean().

4.2.3 A histogram of the top 10 longest movies duration

Visualizing data is essential to facilitate perception and to understand information: Create a graph to visualize the top 10 longest movies. We focus on creating a histogram in the tips section since it's perhaps the most common approach for plotting. You can, however, choose a different chart type; just make sure that the information you want to display is clear and correct. However, as in school, always add axis labels when possible.

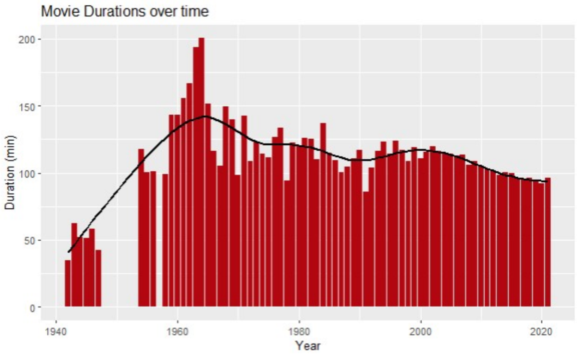


If your data is saved as a dataframe you can perceive with the data from the previous task. Use {ggplot2} to create a histogram. Integrate top_n() before the ggplot() call and select geom_col() to achieve a histogram presentation.

How exactly you do this is up to you. An easy approach might be to sort the dataframe with movie durations in descending order and select only the first ten columns for a barplot. Look into pandas.nlargest() for more information.

4.2.4 Visualizing movie durations over time

While we are at the topic of movie durations, it might also be interesting to see how the average movie length evolved: Choose and plot the chart you think is most appropriate for displaying this type of information. After plotting, please comment and interpret the graph: Were there any significant increases/decreases in movie length over time? If so, what could be the reason?



Again, it's ggplot's turn. Instead of preparing the summarise() and mean() functions combination you can also directly integrate the adjustments in the geom_bar call with (stat = "summary", fun = "mean").

In order to summarize the mean movie length, look into the functions groupby() to group all movies by year and then find the average length mean(). Make sure that every year in the dataframe is present. In order to plot two plots on one axis, declare the axis as one plot and use the function axis.twinx().

4.3 Your personal data

4.3.1 Load your data

You will now use the data set which you requested from Netflix. In the Netflix folder you will find the document of interest: ViewingActivity. Load this in your environment and inspect it as you did before with the Netflix Dataset. If you can't request your data, ask your mentor; they will provide you with an alternative data set.

4.3.2 Clean and transform dataset

As you might have noticed, Netflix recorded every time you clicked on a movie even if you didn't watch it. Check which column indicates those with a specific value. To avoid a bias in the following analysis, delete the respective rows. You can further drop columns that seem unnecessary or don't give any information and change column names if you wish. Additionally, convert the column Duration in minutes (round them) and extract the viewing day and viewing time from "Start time" into two separate columns (e.g., "date" and "time").

mutate and filter are your best friends in this cleaning process.

The most useful course for this task is probably Cleaning Data in Python. Use datetime functions, drop columns, filter by time.

4.3.3 Merging datasets: primitive approach

As a data scientist, you'll often find yourself working with data sets from different data sources referencing the same object. For example, you might have the movie names in one file and the respective genre in a separate file. It would make more sense to just merge the two data sets into one. Indeed, this is the case with our data. Your Netflix data does not provide information on genre, actor, or director, while the general Netflix data set does. So, to make life easier for the upcoming tasks, you'll now need to merge both data sets by the title name.

There are several things kind of "wrong" with the merged data set. What is it and why? Tipp: You need to prepare the dataset in a way that the title, session and episode will be split into a column of title, session and episode separately in order for the two data frames to join each other properly.

Calls such as separate(), ifelse() and rename() are suitable here. Merging can be done in several ways. You could, e.g., use {dplyr}'s left_join() to combine the two data sets based on the "left sided" dataset. Which one should be the "left-sided"?

The merge function allows for different types of merges. Try coming up with the most logical one in this scenario!

4.3.4 Merging datasets: advanced approach

You might have noticed that merging the data set is a little challenging. This is because Netflix sometimes displays title, session and episode in a remixed way and also because your Netflix data might be in German whereas the general Netflix Data is in English. We fixed this issue externally and you can now send your data to Moritz Schwerdt, who will provide you with a respective general Netflix dataset. Now try merging again!