06.11.25

OPENCAMPUS.sh

Introduction to Data Science and machine Learning

VERSIONING WITH GIT (PART 1) AND DATA PREPARATION

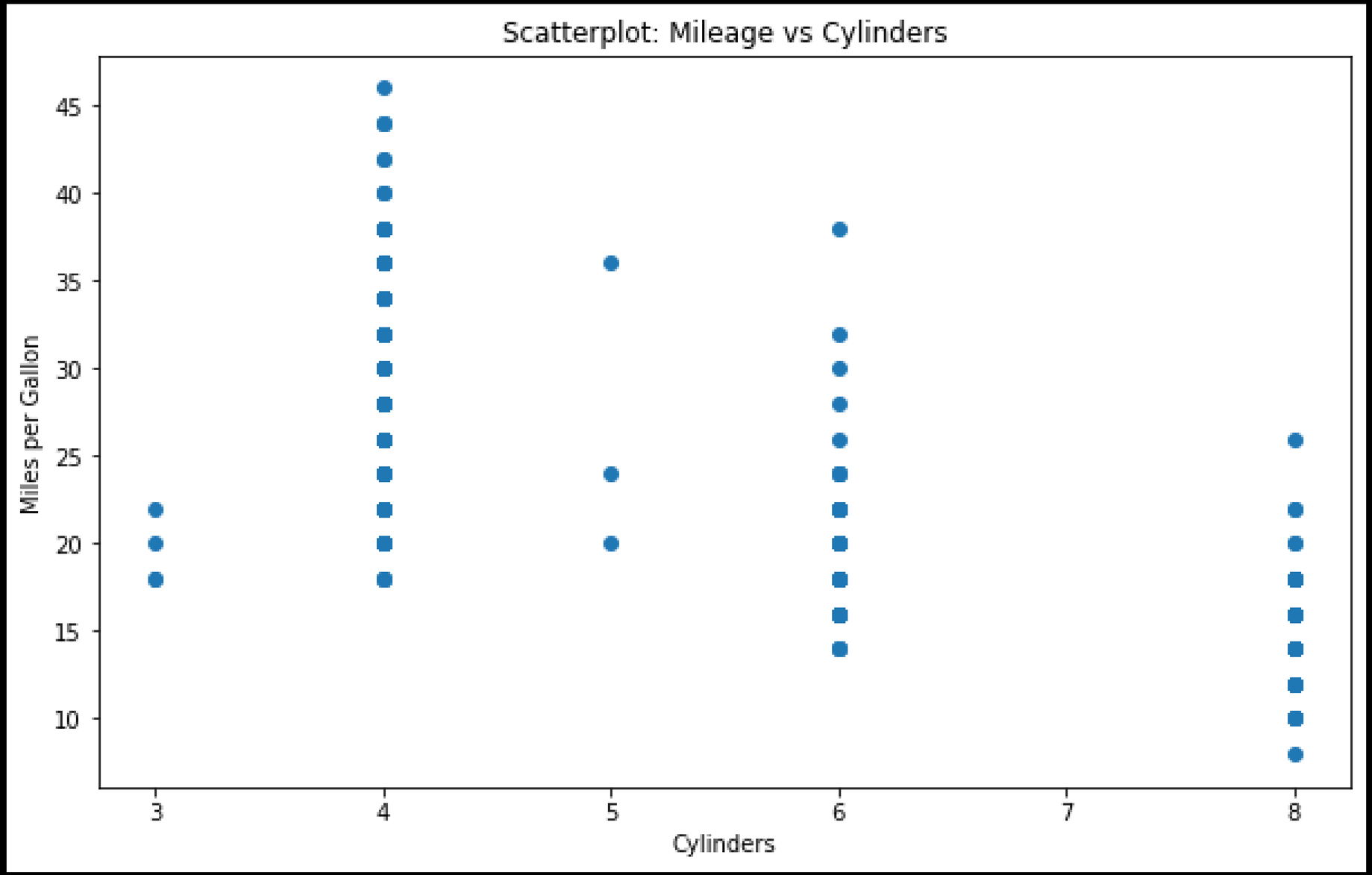
- Team Formation
- Getting to Know Each Other and Discussion of the Tasks for today
- Additional Examples of Diagrams
- Statistical Significance
- Introduction to Version Control with Git
- Introduction to Data Preparation

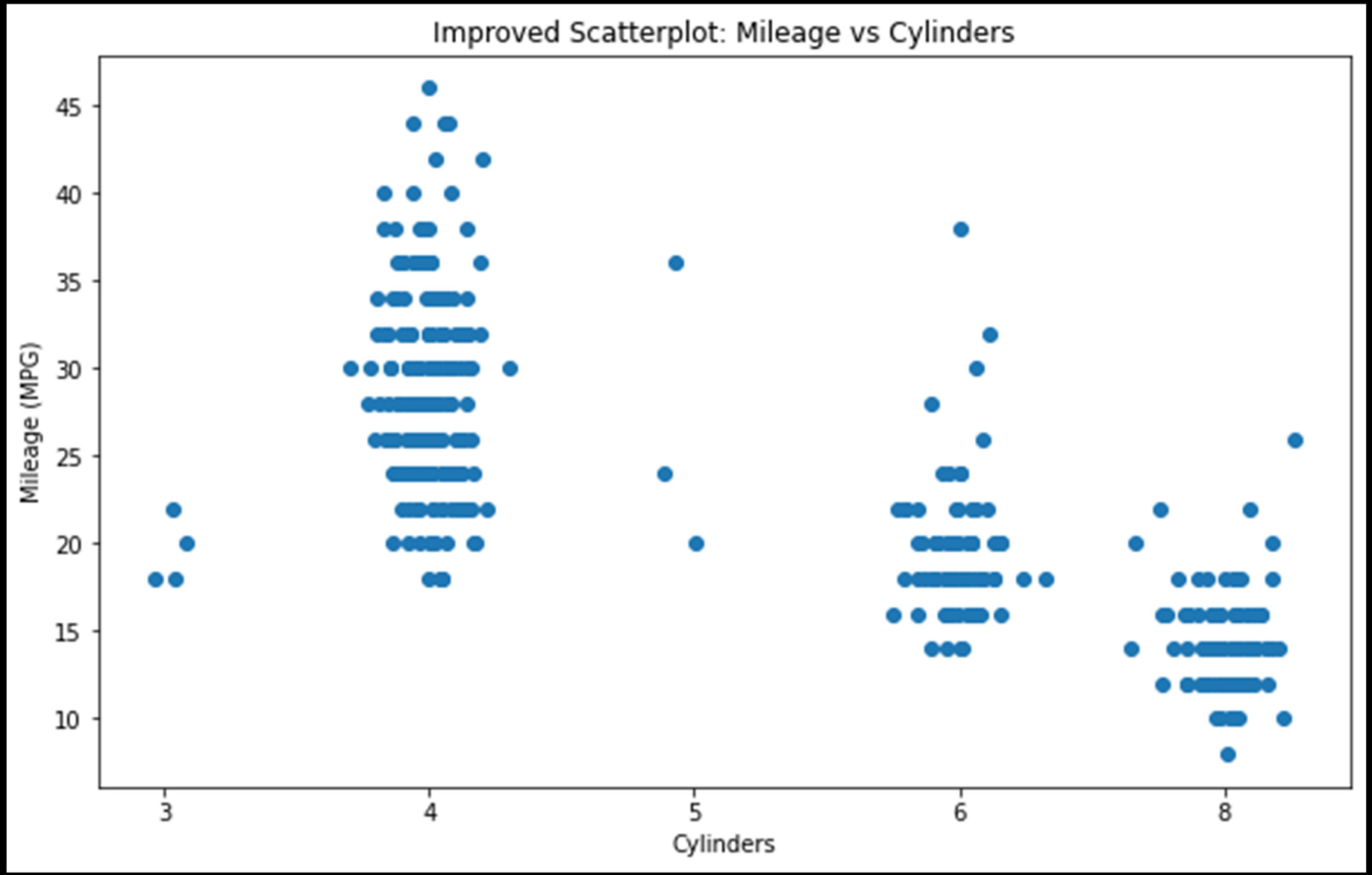
TEAM FORMATION

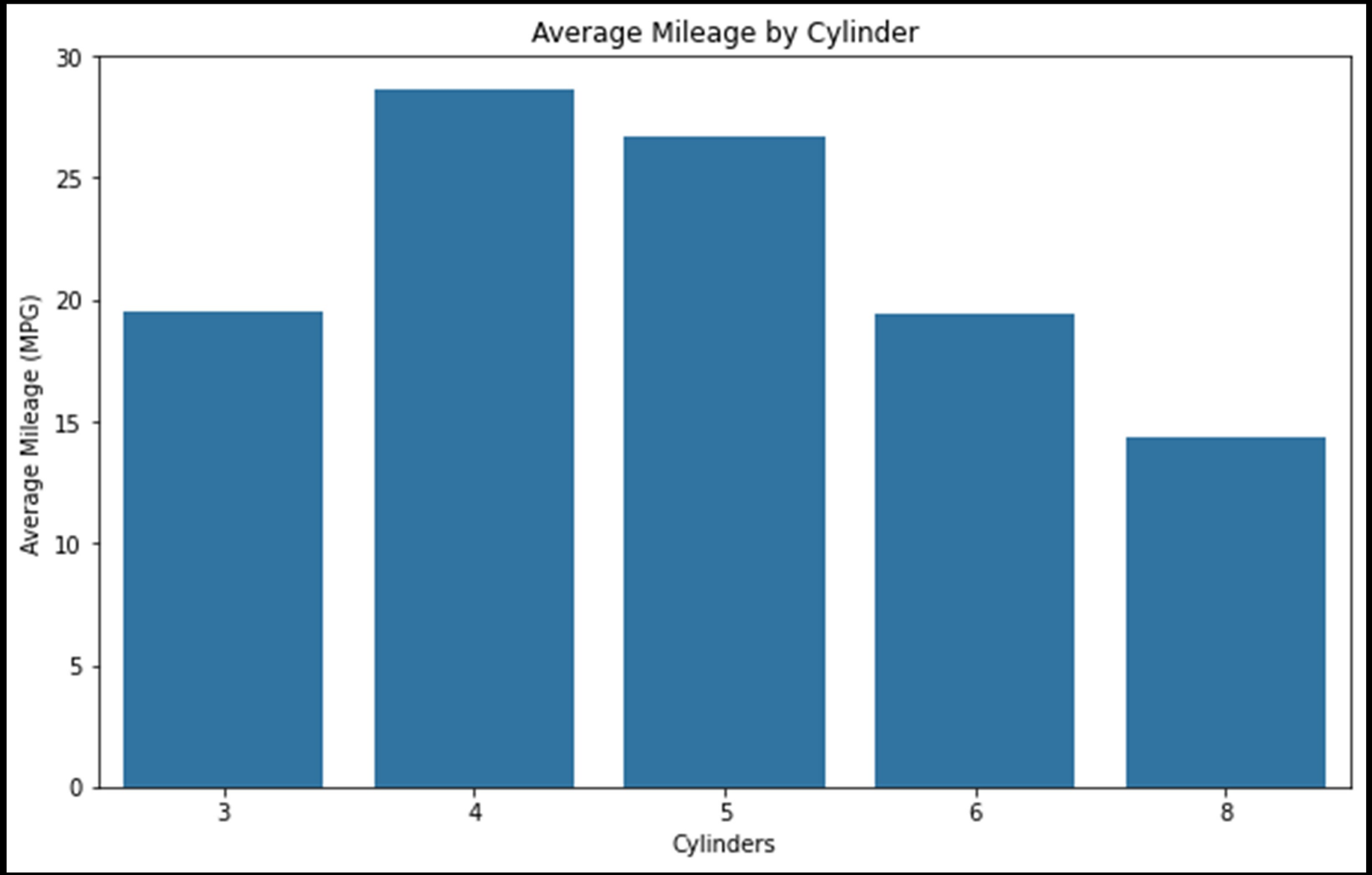
BREAKOUT

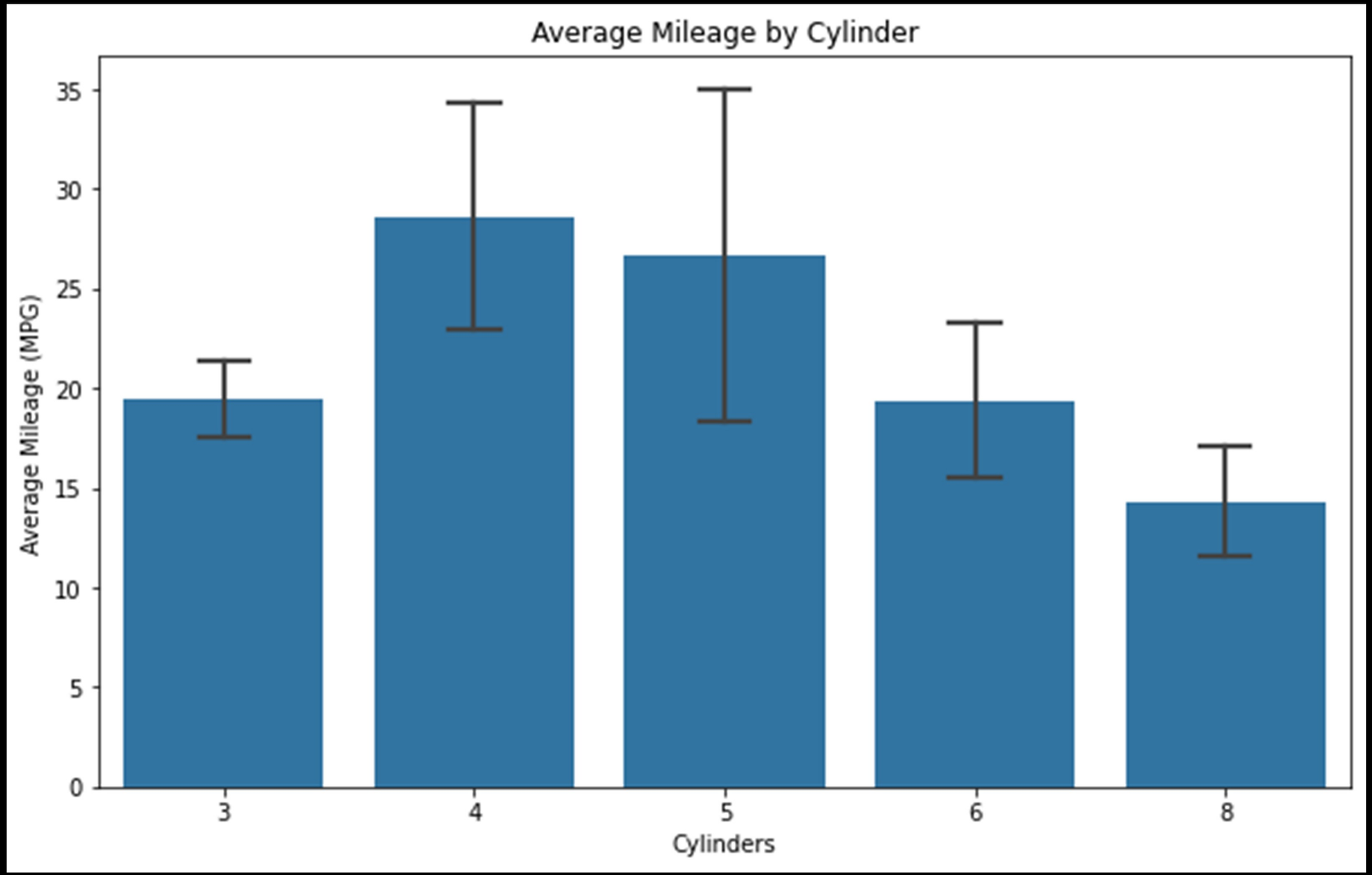
- Introduction Round
- Create a team channel in Mattermost that includes the number
  - of your Breakout room – if you are online
  - your assigned team number – if you are in presence
- Discussion of Exercise Tasks

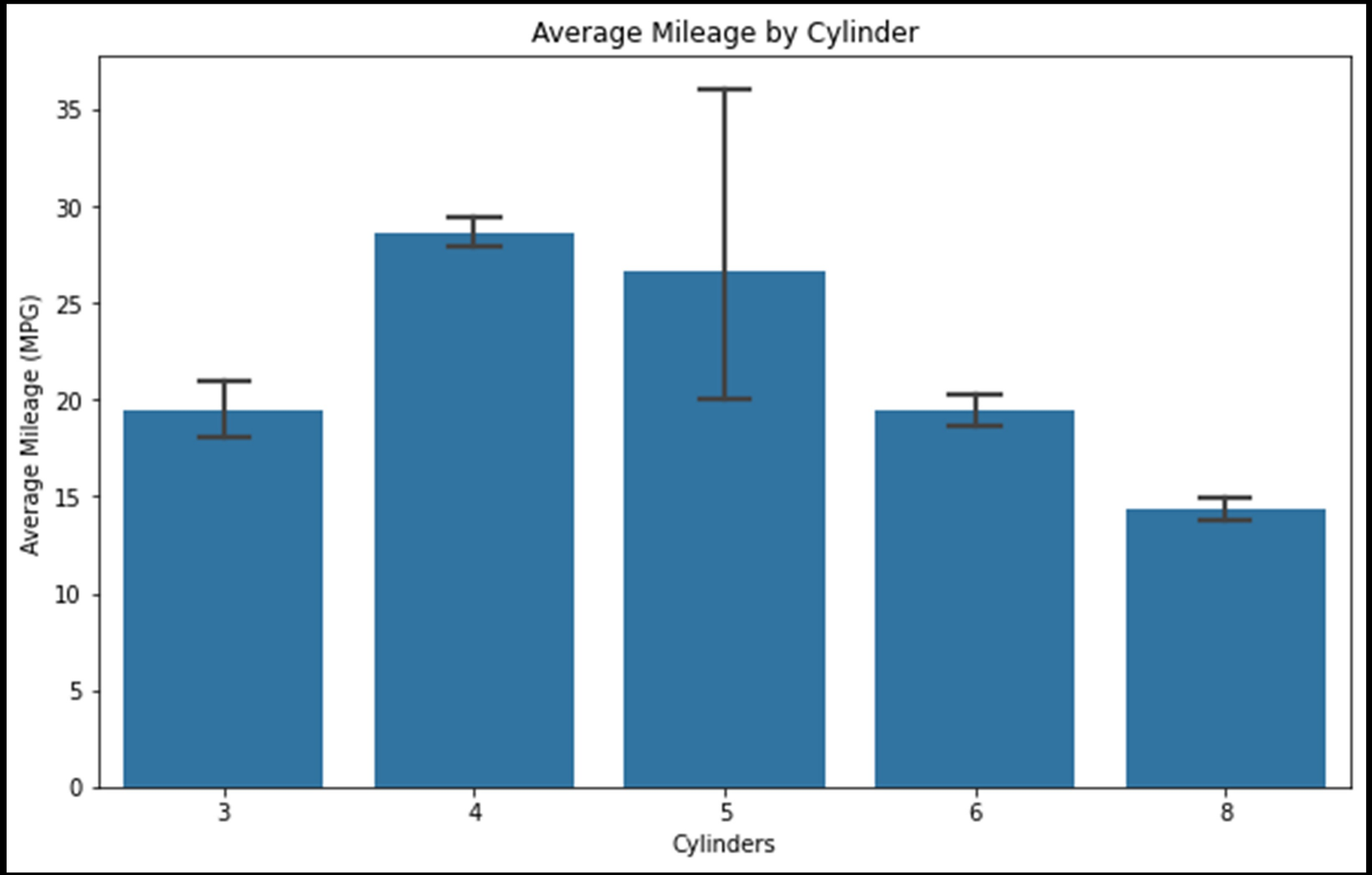
ADDITIONAL EXAMPLES OF DIAGRAMS

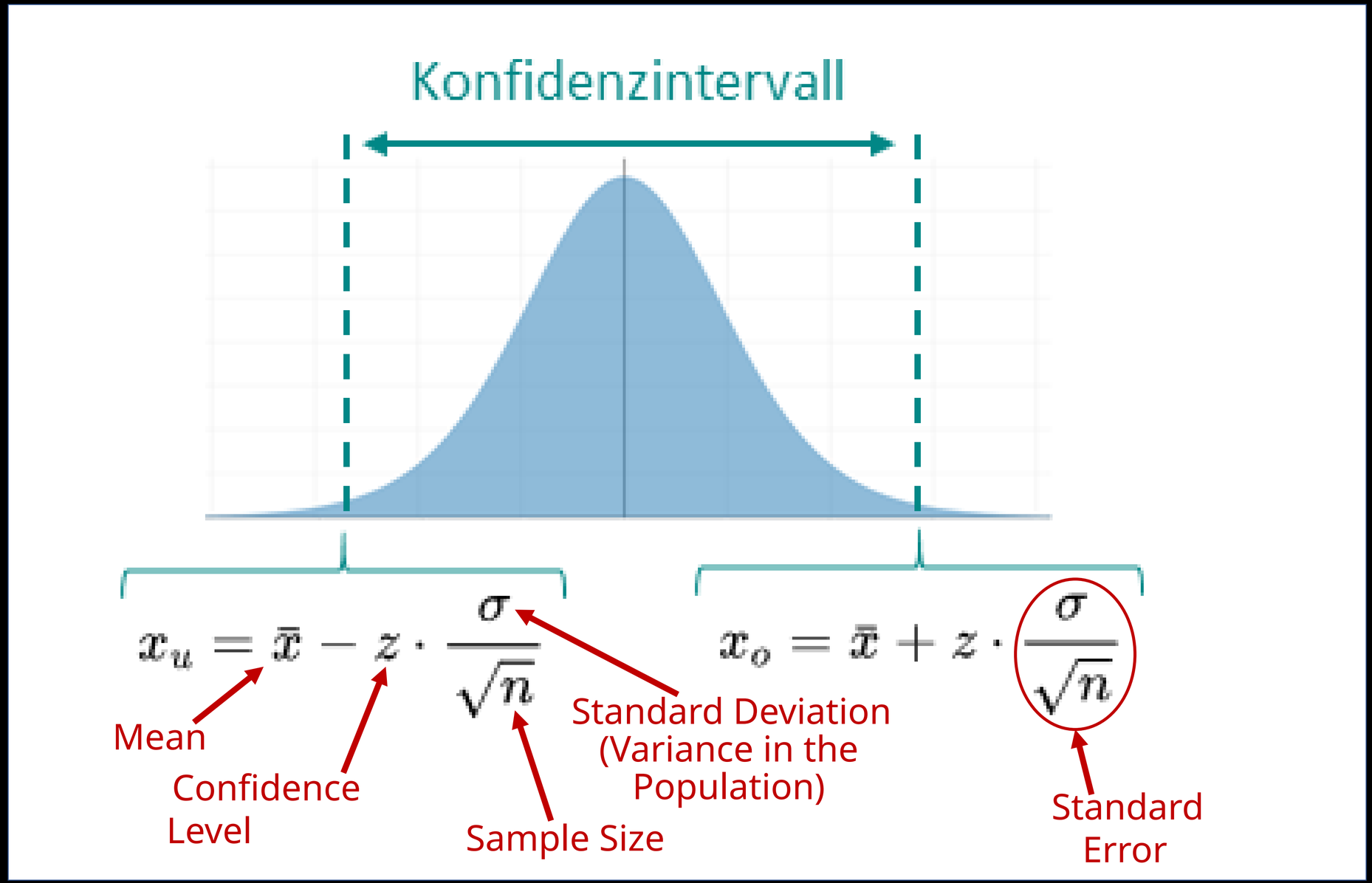

Scatterplot: Mileage vs Cylinders
Miles per Gallon
45
40
35
30
25
20
15
10
3
4
5
6
7
8
Cylinders


Improved Scatterplot: Mileage vs Cylinders
Mileage (MPG)
45
40
35
30
25
20
15
10
3
4
5
6
8
Cylinders


Average Mileage by Cylinder
Average Mileage (MPG)
30
25
20
15
10
5
0
3
4
5
6
8
Cylinders


Average Mileage by Cylinder
Average Mileage (MPG)
0
5
10
15
20
25
30
35
3
4
5
6
8
Cylinders


Average Mileage by Cylinder
Average Mileage (MPG)
0
5
10
15
20
25
30
35
3
4
5
6
8
Cylinders



Datatab e.U. (n.d.). Konfidenzintervall. Retrieved November 8, 2021, from Datatab Statistik Rechner website: https://datatab.de/tutorial/konfidenzintervall

STANDARD DEVIATION VS. STANDARD ERROR

Standard Deviation:

- Information about the spread of observations within a group

→ „Description“

Standard Error:

- Information about the error of the mean of observations for a group, *when considering it as a representative sample for a larger population*.

→ “Prediction”

SIGNIFICANCE TESTS (1)

- How do you know if a mean is "statistically significant" above (or below) a defined value?
  - If you know with (typically) 95% certainty that the mean is greater than x.
  - If the 95% confidence interval lies below x.

Relevance of Model Parameters:

If a model parameter is not significantly above or below zero, it is not relevant for the model.

SIGNIFICANCE TESTS (2)

- How do you know that two means differ "statistically significantly" from each other?

  - If the 95% confidence intervals of both means do not overlap.
  - If the difference between the means is with 95% certainty different from 0.

T-TEST

```
> t.test(july$Temp, may$Temp)

        Welch Two Sample t-test

data:  july$Temp and may$Temp
t = 12.616, df = 50.552, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 15.43351 21.27617
sample estimates:
mean of x mean of y
 83.90323  65.54839
```

SIGNIFICANCE VERSUS RELEVANCE

- **Properties can differ statistically significantly, but the difference may have no practical relevance (no "effect").**

Consideration of effect size according to Cohen:
(Size of the mean difference compared to the standard deviation)

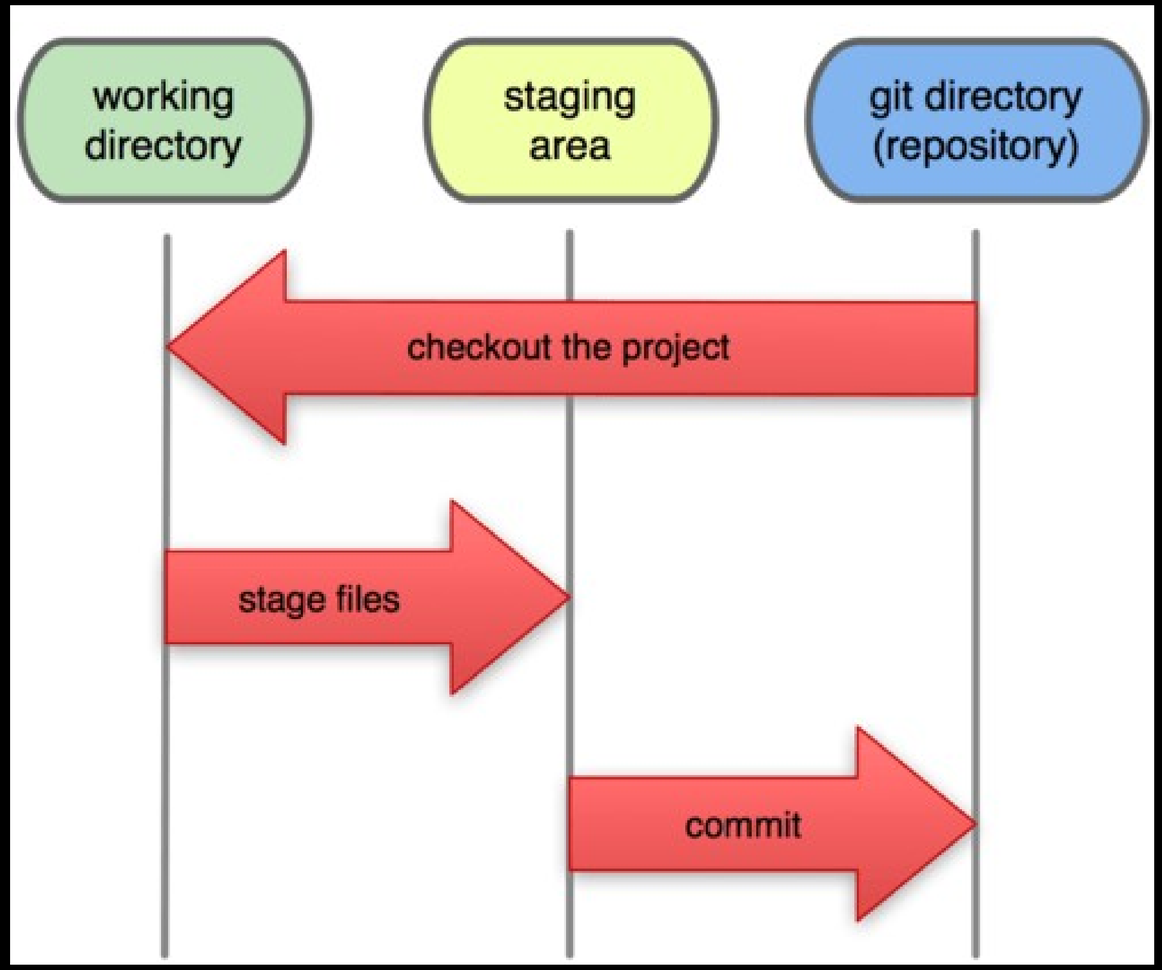
$$d = \frac{\mu_1 - \mu_2}{\sigma}$$

Simplified case of equal sample size and standard deviation

$\lvert d \rvert > 0{,}2$	**small effect**
$\lvert d \rvert > 0{,}5$	**medium effckt**
$\lvert d \rvert > 0{,}8$	**großer Effekt**

VERSION CONTROL WITH GIT

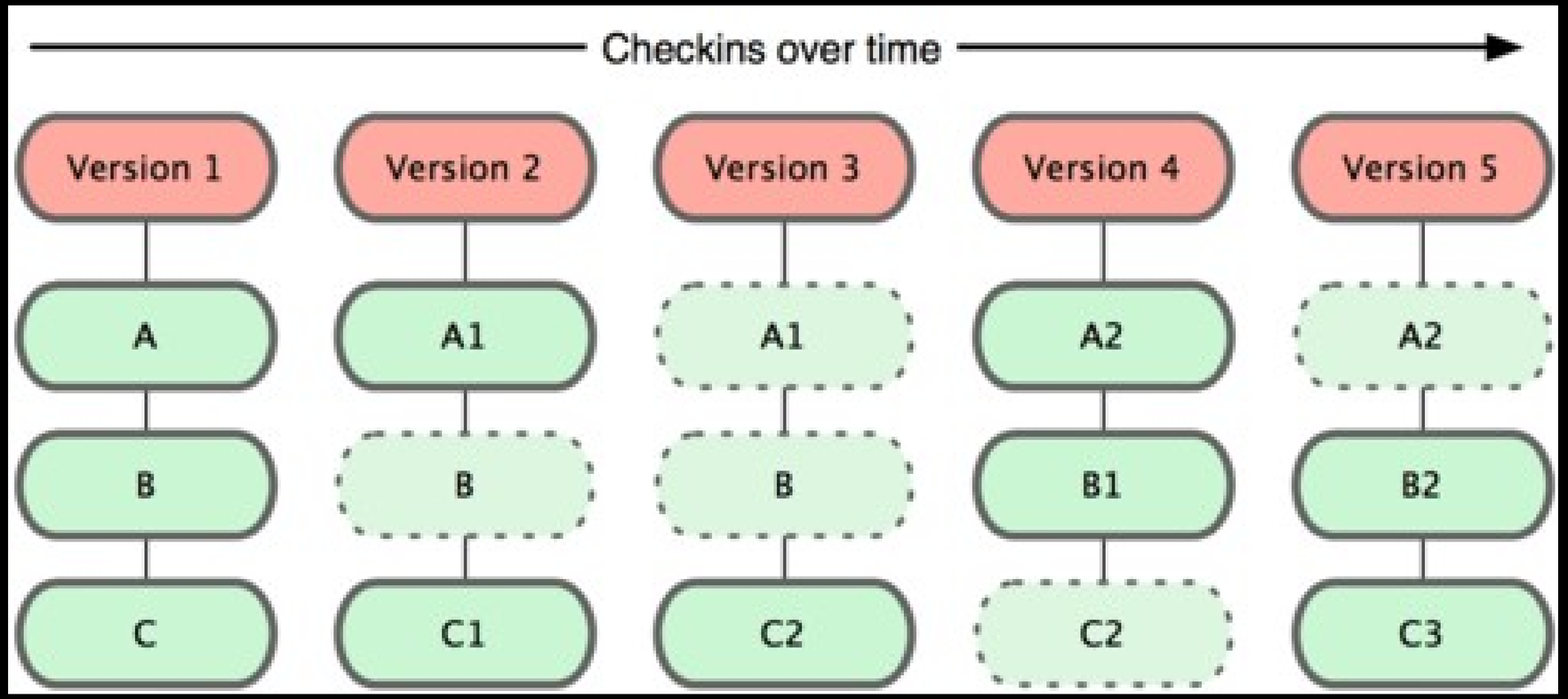
LOCAL VERSION CONTROL WITH GIT



A file can have the following states:

- Ignored (excluded from version control via .gitignore)
- Untracked (not yet under version control)
- Modified (changed compared to the last versioned file)
- Staged (marked for inclusion in the next version)
- Committed (under version control)

LOCAL VERSION CONTROL WITH GIT



- All versions are stored in a specially designated directory called a “repository.”
- Each version contains all files of the project, but internally only the actual changes of each version are stored.

USAGE IN VS CODE

LOCAL GIT CONFIGURATION

Before using Git for the first time, you must define (once) in whose name the repositories of the installed Git will be managed.

- In the "Terminal", execute the following two lines with your GitHub username and your email address (if not already set):

  ```
  git config --global user.name "your_username“
  git config --global user.email your_email@example.com
  ```

- To verify, you can enter the following command:
  ```
  git config --list
  ```

BREAKOUT

1) “Stage” one or more files that you want to version, then “commit” them.
2) Execute a “commit” to create a new project version including the newly staged files.
3) Take a look at the history of your repository.

VERSIONING WITH GIT

Part 1: Local Version Control

Part 2: Synchronizing Local Version Control with a Remote Repository and Working in a Team

DATA PREPARATION FOR MACHINE LEARNING

- **Data Collection**
  Collecting raw data from various sources such as databases, files, APIs, or web scraping.
- **Cleaning**
  Handling missing data, removing duplicates, and correcting errors.
- **Data Exploration / Analysis**
  Understanding the distribution of variables, identifying outliers and relationships—for example, using statistical metrics, correlation matrices, or visualization tools.
- **Feature-Engineering**
  Transforming raw data into features that better represent the underlying problem for models (e.g., creating new variables from existing data), encoding categorical variables, and handling missing values.
- **Data Splitting**
  Dividing the data into training, validation, and test sets to ensure that the model generalizes well to new, unseen data..

JOIN-METHODEN FÜR PANDAS DATAFRAMES

- **Inner Join**
  Returns only the rows with matching keys in both DataFrames.
  `df1.merge(df2, on='key', how='inner')`

- **Outer Join**
  Returns all rows from both DataFrames. Non-matching keys will have NaN for the missing values from the other DataFrame.
  `df1.merge(df2, on='key', how='outer')`

- **Left Join (and Right Join correspondingly)**
  Returns all rows from the right DataFrame, and the matching rows from the left DataFrame. Missing matches are filled with NaN.
  `df1.merge(df2, on='key', how='left’)`

INITIAL DATA INSPECTION

- `df.head()`, `df.tail()`:
  Shows the first/last rows.
- `df.sample()`:
  Shows a random selection of rows.
- `df.shape()`:
  Returns the dimensions of the DataFrame.
- `df.info()`:
  Summary of the DataFrame, including data types.

DESCRIPTIVE STATISTICS

- `df.describe()`:
  Statistical summary of the numerical columns.

- `df.isnull()`:
  Checks for NaN values.

VISUALIZATIONS

- Scatterplots
- Bar Charts
- Histograms

LEARNING RESOURCES

- Complete this course on DataCamp about merging DataFrames.
- Watch this video on Git in VS Code.
- For an introduction to the possibilities of regular expressions, check out this this 11-minute video.

TASKS

- Watch this video (3 minutes) and designate a person in the team who will create a team repository as shown there.
- Watch this video (2 minutes) to create a GitHub Codespace based on your team repository.
- Import the data from `umsatzdaten_gekuerzt.csv`, `kiwo.csv`, and `wetter.csv` and merge them into a single Pandas DataFrame.
- Meet with your team and assign responsibilities for creating descriptive statistics and visualizations for specific variables.
- Create descriptive statistics and visualizations for the respective variables.