Welcome to the CoGrammar Data Cleaning

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

Data Science Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: [Questions]

Data Science Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support

- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting

- We would love your **feedback** on lectures: Feedback on Lectures

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Data Cleaning

April 2024

Learning objectives

- Describe techniques for **handling missing data** and when each is appropriate to use.
- Demonstrate how to identify and remove duplicate records in a dataset using Pandas.
- Explain the importance of **consistent data formatting and apply methods to standardise data.**
- Define **outliers** and discuss strategies for detecting and handling them appropriately based on the data context.

Handling Missing Data

- Missing data refers to the **absence of values in one or more variables in a dataset.**

- Identifying missing values:
  - Look for **null, NaN, or empty cells in the dataset**.
  - Use functions like **isnull() or isna()** in Pandas to detect missing values.

```
data.isnull().sum()
```

```
Name      0
Age       1
Salary    0
City      0
dtype: int64
```

Handling Missing Data

- Techniques for dealing with missing data:
  - **Deletion:** Remove records with missing values (only suitable if missing data is minimal and random).

```
data_deleted = data.dropna()
```

Handling Missing Data

- Techniques for dealing with missing data:
  - **Imputation:** Fill in missing values with estimated or calculated values.
    - **Simple imputation methods:** Mean, median, or mode imputation.

```
data_imputed_mean = data.fillna(data["Age"].mean())
```

    - **Advanced imputation methods:** K-Nearest Neighbors (KNN), Multiple Imputation by Chained Equations (MICE).

```
data_imputed_knn = imputer.fit_transform(data[['Age']])
```

Imputation Considerations

- The choice of imputation method depends on the **nature of the missing data** and the analysis requirements.

CoGrammar

Missing Completely at Random (MCAR)

- The probability of a value being missing is the **same for all cases** and does not depend on any other variables in the dataset.

- Example: In a survey, a participant accidentally skips a question. The missingness is unrelated to the participant's characteristics or other responses.

CoGrammar

Missing at Random (MAR)

- The probability of a value being missing **depends on other observed variables in the dataset but not on the missing values themselves.**

- Example: In a medical study, **younger participants are more likely to miss a follow-up appointment**. The missingness is r**elated to the observed variable "age" but not to the unobserved health outcomes**.

CoGrammar

Missing Not at Random (MNAR)

- The probability of a value being missing **depends on the missing values themselves,** even after accounting for other observed variables.

- Example: In an income survey, **high-income individuals are more likely to refuse to report their income**. The missingness is related to the unobserved income level itself.

Importance of Missing Data Mechanisms

- Understanding the type of missingness is crucial for selecting appropriate handling techniques.

- **MCAR:** Simple methods like deletion or mean imputation may be suitable.
- **MAR:** More advanced methods like multiple imputation can be used.
- **MNAR:** Requires careful consideration and modeling of the missingness mechanism.

Determining the Missing Data Mechanism

- ❖ Assess the relationship between missingness and other variables in the dataset.
- ❖ Consider domain knowledge and the data collection process.
- ❖ Conduct statistical tests to examine patterns of missingness.
- ❖ Be cautious and transparent about assumptions made regarding the missing data mechanism.

CoGrammar

Dealing with Duplicates

- ❖ Duplicate records are multiple instances of the same data point in a dataset.

4	David	40.0	80000	London
5	David	40.0	80000	London

- ❖ Identifying duplicates:
  - ➢ Use functions like **duplicated()** in Pandas to identify duplicate records.
    Specify the subset of columns to consider when identifying duplicates.

Dealing with Duplicates

- Strategies for handling duplicates:
  - **Removing duplicates:** Drop duplicate records from the dataset using drop_duplicates().
  - **Merging duplicates:** Combine duplicate records into a single record by aggregating or selecting relevant information.

```
data_deduplicated = data.drop_duplicates()
```

CoGrammar

Data Formatting and Standardisation

- ❖ Consistent data formatting is essential for **accurate analysis** and compatibility with different tools and algorithms.

- ❖ Common formatting issues:
  - ➢ **Date and time formats:** Ensure consistent representation (e.g., YYYY-MM-DD, HH:MM:SS).
  - ➢ **Text case inconsistencies:** Convert text to a consistent case (lowercase or uppercase).
  - ➢ **Inconsistent value representations:** Standardise values (e.g., "Yes"/"No" vs. "Y"/"N").

CoGrammar

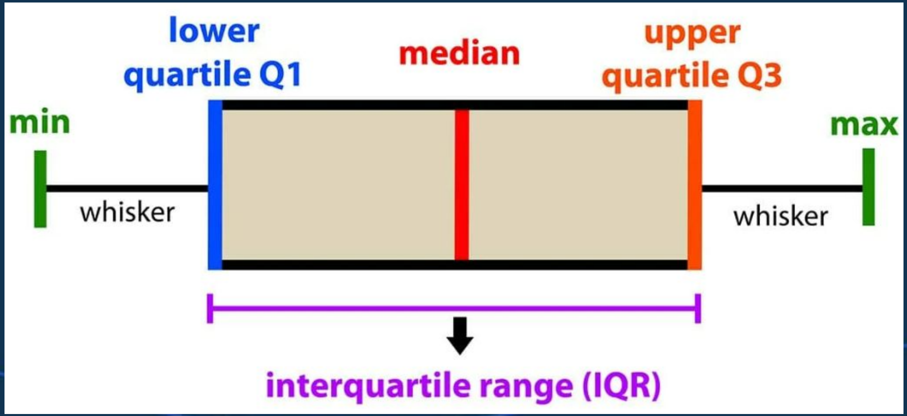
Handling Outliers

❖ Outliers are data points that significantly deviate from the rest of the data distribution.

CoGrammar

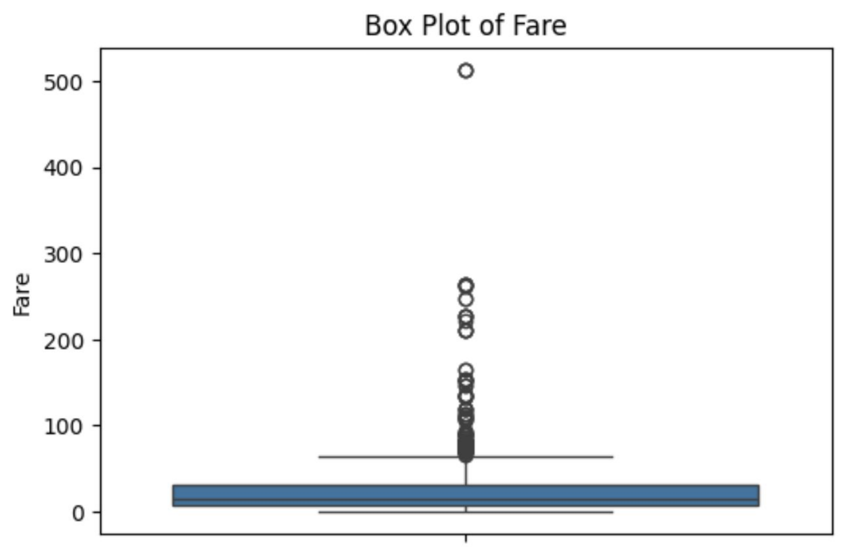
Handling Outliers

- Identifying outliers:
  - **Visual inspection:** Use plots like box plots, scatter plots, or histograms to identify extreme values.



Handling Outliers

Using a box plot to identify outliers.



CoGrammar

Handling Outliers

- Strategies for handling outliers:
  - **Removal:** Remove outliers from the dataset if they are erroneous or irrelevant to the analysis.
  - **Transformation:** Apply mathematical transformations (e.g., logarithmic, square root) to reduce the impact of outliers.
  - **Winsorisation:** Replace extreme values with the nearest non-outlier values.

```
data_winsorized['Salary'] = stats.mstats.winsorize(data_winsorized['Salary'], limits=0.2)
```

- Consider the context and domain knowledge when deciding how to handle outliers.

CoGrammar

Which of the following is NOT a common data quality issue?

A. Missing values
B. Duplicates
C. Inconsistent formatting
D. Small sample size

Which of the following is NOT a common data quality issue?

A. Missing values

B. Duplicates

C. Inconsistent formatting

D. Small sample size

CoGrammar

Which technique is suitable for handling missing data only if the amount is minimal and missing at random?

A. Mean imputation

B. Deletion

C. K-Nearest Neighbours imputation

D. Multiple Imputation by Chained Equations

CoGrammar

Which technique is suitable for handling missing data only if the amount is minimal and missing at random?

A. Mean imputation

B. Deletion

C. K-Nearest Neighbours imputation

D. Multiple Imputation by Chained Equations

CoGrammar

In Pandas, which function can be used to identify duplicate records in a dataset?

A. find_duplicates()

B. duplicated()

C. is_duplicate()

D. has_duplicates()

CoGrammar

In Pandas, which function can be used to identify duplicate records in a dataset?

A. find_duplicates()

B. duplicated()

C. is_duplicate()

D. has_duplicates()

Which of the following is a technique for standardising inconsistent text case?

A. astype()

B. to_datetime()

C. upper() or lower()

D. strip()

Which of the following is a technique for standardising inconsistent text case?

A. astype()

B. to_datetime()

C. upper() or lower()

D. strip()

Which strategy replaces outlier values with the nearest non-outlier values?

A. Removal
B. Transformation
C. Winsorisation
D. Standardisation

Which strategy replaces outlier values with the nearest non-outlier values?

A. Removal

B. Transformation

C. Winsorisation

D. Standardisation

Conclusions

CoGrammar

Conclusions

- Recap of key points:
  - Data cleaning is an essential step in the data science pipeline.
  - It involves **handling missing data, duplicates, formatting issues, outliers, and data validation.**
  - Various techniques and tools are available for effective data cleaning.

Conclusions

- Importance of iterative data cleaning:
  - Data cleaning is an iterative process that may require multiple rounds.
  - Continuously assess and refine the cleaned data based on analysis results and feedback.

Further Learning

KDNuggets - Learn Data Cleaning and Preprocessing for Data Science with This Free eBook

Kaggle - Short Data Cleaning Course

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar