

1 PARC

Data Wrangling I – Report

1. Import Required Libraries

python

CopyEdit

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

We start by importing essential Python libraries:

- **NumPy** and **Pandas** for numerical and data manipulation,
- **Matplotlib** and **Seaborn** for visualizations.

2. Locate and Describe Dataset

Dataset: Australian Weather Data

Source: Kaggle - Rain in Australia

Description:

This dataset contains daily weather observations from numerous Australian weather stations. It includes 24 variables like temperature, humidity, wind direction/speed, and rainfall recorded over 10 years. The goal is to predict whether it will rain tomorrow.

3. Load Dataset into Pandas DataFrame

python

CopyEdit

```
df = pd.read_csv('datasets/weatherAUS.csv')
```

```
print(df.head())
```

We use `pd.read_csv()` to load the dataset. The first few (`head()`) and last few (`tail()`) rows confirm successful loading and give a snapshot of the data.

4. Data Preprocessing

a) Initial Exploration

python

CopyEdit

```
print(df.info())
```

```
print(df.describe(include='all'))
```

- **Dimensions:** (142193, 24) — 142K rows and 24 columns.
- Used `.info()` to check data types and null values.
- Used `.describe()` to obtain statistics for numeric and object columns.

b) Missing Values Check

python

CopyEdit

```
print(df.isnull().sum())
```

This shows the count of missing values per column. For example, 'Sunshine' has 67,816 missing values.

5. Data Formatting and Normalization

a) Variable Types

python

CopyEdit

```
print(df.dtypes)
```

- Variables are categorized into object, float64, etc.
- Categorical features: Location, WindGustDir, WindDir9am, RainToday, RainTomorrow.
- Continuous features: MinTemp, MaxTemp, etc.

b) Convert 'Date' Column to Datetime

python

CopyEdit

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

Converted the 'Date' column to datetime format for time-based analysis.

c) Handle Missing Categorical Data

python

CopyEdit

```
categorical_cols = df.select_dtypes(include='object').columns
```

```
for col in categorical_cols:
```

```
    df[col] = df[col].fillna(df[col].mode()[0])
```

Used **mode** to fill missing values in categorical columns.

d) Handle Missing Numeric Data

python

CopyEdit

```
numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
```

```
for col in numeric_cols:
```

```
    df[col] = df[col].fillna(df[col].mean())
```

Used **mean** to fill missing values in numeric columns.

6. Convert Categorical to Quantitative Variables

python

CopyEdit

```
df = pd.get_dummies(df, columns=['RainToday', 'RainTomorrow', 'WindGustDir',  
'WindDir9am', 'WindDir3pm', 'Location'], drop_first=True)
```

- Transformed all **categorical features** into **one-hot encoded** numeric columns using `pd.get_dummies()`.
- `drop_first=True` avoids multicollinearity by removing the first category.

Final Verification

python

CopyEdit

```
print(df.info())
```

Now all columns are either numerical or datetime. There are no missing values, and all features are properly formatted for modeling or further analysis.

Would you like help with visualizations or model-building next?

4o

You said:

explain fillna and dropna

ChatGPT said:

Sure! Here's a clear explanation of fillna() and dropna() in pandas:

◆ **fillna(): Fill Missing Values**

The fillna() function is used to **replace missing (NaN) values** in a DataFrame or Series with a specified value or strategy.

✅ **Common Usage:**

python

CopyEdit

```
df['column'].fillna(0)          # Replaces NaN with 0
df.fillna(method='ffill')      # Forward fill: replaces NaN with previous value
df.fillna(method='bfill')      # Backward fill: replaces NaN with next value
df.fillna(df.mean())           # Fills with mean (only works on numeric columns)
```

📄 **Example:**

python

CopyEdit

```
df['Temperature'].fillna(df['Temperature'].mean(), inplace=True)
```

This replaces all NaN values in the Temperature column with the column's **mean**.

◆ **dropna(): Remove Missing Values**

The dropna() function is used to **remove rows or columns** that contain missing (NaN) values.

✅ **Common Usage:**

python

CopyEdit

```
df.dropna()                    # Drops all rows with any NaN
df.dropna(axis=1)              # Drops columns with any NaN
df.dropna(how='all')           # Drops rows where all elements are NaN
df.dropna(thresh=3)            # Keeps rows with at least 3 non-NaN values
```

Example:

python

CopyEdit

```
df.dropna(subset=['Rainfall'], inplace=True)
```

This drops only the rows where the Rainfall column has a NaN value.

Summary:

Function Purpose

Use When

fillna() Fill missing values with a value You want to keep the data but complete it

dropna() Remove rows/columns with NaNs You want to discard incomplete data

Would you like an example comparing fillna() vs dropna() on a toy dataset?

4o

You said:

explain ategorical variables quantitative variables

ChatGPT said:

Sure! Here's a simple and clear explanation of **categorical** and **quantitative variables**:

◆ **Categorical Variables (Qualitative Variables)**

These are variables that represent **categories or labels**. They describe **qualities** or **groups**, **not numbers** (even if they look like numbers).

✅ **Examples:**

- Gender: Male, Female
- City: Pune, Mumbai, Delhi
- Grade: A, B, C
- Marital Status: Single, Married, Divorced
- Blood Group: A+, B-, O+

Even if you use numbers like 1 for Male and 0 for Female, they are still **categorical** because the number doesn't mean quantity.

◆ **Quantitative Variables (Numerical Variables)**

These are variables that represent **numerical values** and **can be measured**. They describe **quantities**.

✅ **Examples:**

- Age: 21, 35
- Height: 170 cm, 5.9 ft
- Salary: ₹50,000
- Temperature: 22.5°C
- Number of children: 2

These can be further divided into:

- **Discrete:** Countable values (e.g., number of siblings)
- **Continuous:** Any value within a range (e.g., weight, temperature)

Summary Table:

Type	Description	Examples
Categorical	Represents categories or labels	Gender, City, Blood Group
Quantitative	Represents measurable numerical data	Age, Salary, Temperature

Do you want help identifying variable types in a dataset you're working on?

4o

You said:

give me question and answers of this practical that the external ask me [hard mode]
because external is very strict {40 question}

ChatGPT said:

Updated saved memory

Here are 40 **hard-mode questions and answers** that a strict external examiner might ask you during a practical exam involving **pandas**, **categorical and quantitative variables**, and **missing data handling (fillna and dropna)**:

📚 **Categorical & Quantitative Variables**

1. **Q: What distinguishes a categorical variable from a quantitative one?**
A: A categorical variable represents types or groups (e.g., gender, color), while a quantitative variable represents numeric values that quantify something (e.g., age, income).

2. **Q: Can a numeric column be a categorical variable? Give an example.**
A: Yes. For example, "zip code" is numeric but represents categories, not quantities.
 3. **Q: How would you convert a column to a categorical variable in pandas?**
A: `df['col'] = df['col'].astype('category')`
 4. **Q: What is the use of .cat.codes in pandas?**
A: It assigns integer codes to each category, useful for machine learning models.
 5. **Q: Which visualization is best for comparing categorical and quantitative variables?**
A: Boxplot or violin plot.
 6. **Q: What statistical summary is meaningful for a quantitative variable but not for a categorical one?**
A: Mean and standard deviation.
 7. **Q: How do you check the number of unique categories in a column?**
A: `df['col'].nunique()`
 8. **Q: How would you handle too many unique categories in a feature?**
A: Combine infrequent categories into an "Other" group.
 9. **Q: What is one-hot encoding and when is it used?**
A: It's the process of converting categorical variables into binary columns. Used before feeding data to ML models.
 10. **Q: Which function can convert categorical variables using one-hot encoding in pandas?**
A: `pd.get_dummies(df['category_col'])`
-

Missing Data (fillna & dropna)

11. **Q: What does fillna() do in pandas?**
A: It replaces missing (NaN) values with a specified value or method.
12. **Q: What does dropna() do?**
A: It removes rows or columns with missing values.
13. **Q: How do you fill missing values in a column with the mean?**
A: `df['col'].fillna(df['col'].mean(), inplace=True)`
14. **Q: How do you drop rows where all values are NaN?**
A: `df.dropna(how='all')`
15. **Q: How would you forward fill missing values in a column?**
A: `df['col'].fillna(method='ffill')`
16. **Q: Which parameter in dropna() lets you drop columns instead of rows?**
A: `axis=1`

17. **Q: Why might fillna(method='bfill') be risky in time series data?**
A: It uses future data to fill current gaps, which can introduce data leakage.
18. **Q: How do you fill missing values differently for different columns?**
A: Use a dictionary: df.fillna({'col1': 0, 'col2': 'unknown'})
19. **Q: What does the limit parameter in fillna() do?**
A: Limits the number of consecutive NaNs to fill.
20. **Q: How would you identify columns with more than 30% missing values?**
A:

python

CopyEdit

```
missing_percent = df.isnull().mean()  
cols = missing_percent[missing_percent > 0.3].index
```

Data Analysis & Edge Cases

21. **Q: How do you treat missing values in a categorical column?**
A: Fill with mode or "Unknown".
22. **Q: What happens if you use fillna() without inplace=True?**
A: The original DataFrame remains unchanged.
23. **Q: Can fillna() fill different methods per column?**
A: Not directly; apply methods per column separately.
24. **Q: What's the difference between .interpolate() and .fillna()?**
A: interpolate() fills missing values using linear or time-based interpolation, not static values.
25. **Q: How do you check the total number of missing values?**
A: df.isnull().sum().sum()
26. **Q: Which method helps fill missing numeric data while preserving distribution?**
A: Imputation using sklearn's SimpleImputer with strategy='mean' or strategy='median'.
27. **Q: What is meant by data leakage in handling missing values?**
A: Using future or test data to fill missing values in training data.
28. **Q: Is it always correct to drop rows with missing values? Why or why not?**
A: No; it can cause loss of information, especially if too many rows are dropped.
29. **Q: Which pandas method gives a full report of missing values?**
A:

python

CopyEdit

`df.isnull().sum()`

30. Q: What is the impact of filling missing data in categorical variables on model performance?

A: It can bias the model if filled incorrectly or distort category distributions.

Advanced Thinking & Application

31. Q: How would you handle a dataset where both categorical and quantitative columns have missing values?

A: Use separate strategies: fill categorical with mode, numeric with mean/median.

32. Q: How does one-hot encoding handle NaN values?

A: It creates a separate column for NaN if `dummy_na=True` is set in `get_dummies()`.

33. Q: Explain how missing values can affect correlation calculations.

A: NaNs are ignored, possibly distorting the true correlation structure.

34. Q: Can we perform group-based imputation in pandas?

A: Yes, using `groupby`:

python

CopyEdit

```
df['col'] = df.groupby('group_col')['col'].transform(lambda x: x.fillna(x.mean()))
```

35. Q: How would you visualize missing data in a dataset?

A: Use libraries like `missingno` or `seaborn`'s heatmap with `df.isnull()`.

36. Q: How does dropping missing data affect statistical validity?

A: It can bias results if data is not missing completely at random (MCAR).

37. Q: What does MCAR, MAR, and MNAR mean?

A: Types of missing data:

- MCAR: Missing Completely At Random
 - MAR: Missing At Random (depends on other variables)
 - MNAR: Missing Not At Random (depends on the missing value itself)
- 38. Q: Why might replacing missing numeric data with zero be problematic?**
- A:** It may change the meaning of the data and introduce bias.
- 39. Q: How do you verify if imputation has skewed your data?**
- A:** Compare distribution before and after imputation using histograms or summary stats.
- 40. Q: What would be your approach to handle missing data in a real-world machine learning pipeline?**
- A:**
- Analyze pattern of missingness

- Drop columns with excessive NaNs
- Impute appropriately (mean, mode, model-based)
- Cross-validate model to check if imputation introduces bias