UNIT 2

Data Prep & Exploration Checklist

This is a non-exhaustive checklist that can help you structure your analysis on Jupyter Notebook.

Step 1: Import & Observe

☐ Import necessary libraries	`import pandas as pd` `import numpy as np` `import matplotlib.pyplot as plt` `import seaborn as sns`
☐ Set notebook options	`# show all dataframe columns` `pd.set_option('display.max_columns', None)` `# set matplotlib global settings eg. figsize` `plt.rcParams['figure.figsize'] = (8.0, 6.0)`
☐ Import your data (set delimiter and index column if applicable)	`# import data` `df_full = pd.read_csv('data.csv', sep=';', index_col=0)` `# check the first few rows of your dataframe` `df_full.head()`
☐ Check the dimensions of your dataframe	`df_full.shape`
☐ Checkpoint #1: Make a copy of your dataset and remove any unecessary columns (you might come back to this later as you start analysing the columns)	`df = df_full.drop(columns=['B', 'C']).copy()`
☐ Get a first overview of your dataset (look out for data types, missing values, column names and row range)	`df.info()`

Step 2: Format & Validate

☐ Check the output of `df.info()`. Does each column have the data type you would expect? If not, reformat column	`# some useful functions (click for more on string functions)` `df['col'].str.split()` `df['col'].str.replace()` `df['col'].astype()`
☐ Format date columns	`# click to read more on how to specify date formats` `pd.to_datetime(df['date_col'], format='%Y%m%d')`
☐ Check for suspicious values / outliers (Eg. what is the count? is the max value too far away from 75th percentile? Are there any negative values where there shouldn't be?)	`df.describe()`
☐ Create subsets for categorical / numerical variables to make your analysis easier	`# categorical variables` `df_cat = df.select_dtypes(include='object').copy()` `# numerical variables` `df_num = df.select_dtypes(include=['int','float']).copy()`
☐ Check how many unique values are in each categorical column	`# count number of unique values per column` `df_cat.nunique()`
☐ For variables with a small number of unique values, check value counts for consistency (eg. ideally you don't want a gender column to have values "M", "F", "Male", "Female")	`# check actual count` `df["categorical_col"].value_counts()` `# check proportion` `df["categorical_col"].value_counts(normalize=True) * 100`

Step 3: Check for duplication

☐ Check for duplicated rows	`# check how many rows are duplicated` `df.duplicated().sum()` `# observe duplicated rows (understand why duplication occurs)` `df[df.duplicated(keep=False)]`
☐ Check for duplicated columns (sometimes columns with different names might have the same values)	`df.T.duplicated()` `# if data is too large you can just check the first few rows` `df.head(100).T.duplicated()`
☐ Check that ID column does not contain duplicates (and if it does, understand why)	`df['ID'].nunique() / len(df)`
☐ Decide on duplication handling strategy (keep vs remove)	`# drop duplicates and keep the first occurence` `df.drop_duplicates(keep='First')`

Step 4: Check for missing values

☐	Check for missing data	`# to check number of missing values` `df.isna().sum()` `# to check proportion of missing values` `df.isna().mean() * 100`
☐	Visualize missing data - Check if missing values coincide across columns - Sort by date and plot, do missing values occure before/after a certain date?	`sns.heatmap(df.isnull(),` `            yticklabels=False,` `            cbar=False)`
☐	Decide on missing data handling strategy (imputation vs deletion)	`# to drop missing values` `df.dropna()` `# to impute missing values eg. using mean` `df.filna(df['co'l.mean()])`
☐	Checkpoint #2: Reset index, make a copy of your final clean dataframe and save it as a csv	`# create a copy to use for the rest of your analysis` `df_final = df.reset_index(drop=True).copy()` `# save a csv so that you can quickly import it next time` `df_final.to_csv('data_final.csv')`

Step 5: Visualize data

☐	Univariate visualizations	`# categorical: which category has the most/least observations?` `sns.countplot(df, x="categorical_col")` `# numerical: what distribution do you observe?` `sns.histplot(df, x="numerical_col")` `sns.boxplot(df, x="numerical_col")`
☐	Multivariate visualizations	`# create scatterplots for all numerical variable combinations` `sns.pairplot(df_num)` `# visualize correlations between numerical columns` `sns.heatmap(df_num.corr())` `# check how distribution varies by different categories` `sns.violinplot(data=df, x="cat1", y="num", hue="cat2")` `sns.boxplot(data=df, x="cat1", y="num", hue="cat2")`
☐	Save your visualizations	`# make sure to give your file a meaningful name` `plt.savefig('sales_by_quarter.png', dpi=200)`

Step 6: Clean up notebook

- ☐ Remove any redundant / unused cells (or put them in an appendix)
- ☐ Make sure your notebook is well structured with clear sections defined by headings and subheadings
- ☐ Add a notebook introduction/executive summary at the beginning
- ☐ Make sure you have communicated your insights clearly (don't leave any plots or findings without explanations)
- ☐ Finally, reset your kernel and check that your notebook runs from top to bottom with no errors