Nitya CloudTech
Dream.Achieve.Succeed

Pandas EDA (Exploratory Data Analysis) Cheatsheet

Data Loading

pd.read_csv(path): Reads a CSV file
pd.read_excel(path, sheet_name="Sheet1"): Reads an Excel file
pd.read_sql(query, Connection_Object): Reads SQL table
pd.read_json(path): Reads a JSON file
pd.read_html(url): Reads tables from an HTML page
pd.read_parquet(path): Reads a Parquet file
df.to_csv("output.csv", index=False): Saves DataFrame to a CSV file
df.to_excel("output.xlsx", index=False): Saves DataFrame to an Excel file
df.to_json("output.json"): Saves DataFrame to a JSON file
df.to_parquet("output.parquet"): Saves DataFrame to a Parquet file

🔎 Data Overview

df.head(n): Displays first n rows (default 5)
df.tail(n): Displays last n rows (default 5)
df.shape: Returns (rows, columns)
df.info(): Displays column data types & memory usage
df.columns: Lists all column names
df.index: Displays index range
df.dtypes: Shows data types of each column
df.describe(): Summary statistics for numerical columns
df.describe(include="all"): Summary statistics for all columns

🔍 Checking Missing Values

Nitya CloudTech
Dream.Achieve.Succeed

df.isnull().sum(): Counts missing values in each column

df.isna().sum(): Same as isnull()

df[df.isnull().any(axis=1)]: Displays rows with missing values

df.dropna(): Removes rows with missing values

df.fillna(value): Replaces missing values with a specified value

df.fillna(df.median()): Fills missing values with median

df.interpolate(): Performs linear interpolation to fill NaN

📊 Checking Duplicates

df.duplicated(): Returns a Boolean Series for duplicate rows

df[df.duplicated()]: Displays duplicate rows

df.drop_duplicates(): Removes duplicate rows

Summary Statistics

Nitya CloudTech
Dream.Achieve.Succeed

df.mean(): Mean of numerical columns
df.median(): Median of numerical columns
df.mode(): Mode of numerical columns
df.std(): Standard deviation of numerical columns
df.var(): Variance of numerical columns
df.min(): Minimum value of each column
df.max(): Maximum value of each column
df.count(): Count of non-null values per column
df.nunique(): Number of unique values per column

📊 Value Counts & Distributions

df["column"].value_counts(): Counts occurrences of each unique value
df["column"].value_counts(normalize=True): Normalized value counts (percentage)
df["column"].unique(): Lists unique values
df["column"].nunique(): Number of unique values

📊 Correlation & Covariance

df.corr(): Correlation matrix (Pearson by default)
df.corr(method="kendall"): Kendall correlation
df.corr(method="spearman"): Spearman correlation
df.cov(): Covariance matrix

📊 Grouping & Aggregation

Nitya CloudTech
Dream.Achieve.Succeed

df.groupby("column")["value"].mean(): Groups by column and gets mean

df.groupby("column")["value"].agg(["sum", "count", "mean"]): Aggregates multiple stats

df.pivot_table(values="sales", index="category", aggfunc="sum"): Pivot table

📈 Data Visualization (Quick Plots)

df.hist(figsize=(10, 5)): Histogram for numerical columns

df.boxplot(figsize=(10, 5)): Box plot for outlier detection

df["column"].plot(kind="hist"): Histogram for a single column

df["column"].plot(kind="box"): Box plot for a single column

df.plot(kind="scatter", x="col1", y="col2"): Scatter plot

🧹 Data Cleaning & Transformation

df["column"].str.lower(): Converts text to lowercase

df["column"].str.upper(): Converts text to uppercase

df["column"].str.strip(): Removes leading/trailing spaces

df["column"].str.replace("old", "new"): Replaces text

df["column"].astype("int"): Converts column to integer type

df["column"] = pd.to_datetime(df["column"]): Converts column to datetime

DateTime Analysis

df["date_column"].dt.year: Extracts year
df["date_column"].dt.month: Extracts month
df["date_column"].dt.day: Extracts day
df["date_column"].dt.weekday: Extracts weekday

Data Filtering & Selection

df.loc[condition]: Filters data based on a condition
df.query('condition'): Filters data using a query string
df.iloc[start:end]: Selects rows by position (inclusive start, exclusive end)
df[df["column"] > value]: Filters rows where column values are greater than a specified value
df[df["column"].isin([value1, value2]): Filters rows where the column matches any of the specified values

🔍 Handling Outliers Using Z-Score

Nitya CloudTech
Dream.Achieve.Succeed

from scipy import stats
z_scores = stats.zscore(df["column"]): Computes Z-scores for a column
df = df[(z_scores < 3) & (z_scores > -3)]: Filters out outliers with Z-scores above 3 or below -3

📊 Pivot Tables & Cross Tabulation

pd.pivot_table(df, values='value', index='row_group', columns='column_group', aggfunc='sum'): Creates pivot table with aggregation
pd.crosstab(df['column1'], df['column2'], margins=True): Creates a cross-tabulation of two columns (with margins)
df.pivot_table(values="value", index="category", aggfunc=["sum", "mean", "std"]): Multiple aggregation functions in a pivot table

Nitya CloudTech

Dream.Achieve.Succeed

FOR CAREER GUIDANCE, CHECK OUT OUR PAGE

www.nityacloudtech.com

Follow Us on Linkedin:

Aditya Chandak

Free SQL Interview Questions

Digital Product | 92 Sales

Helpful | Pyspark | Practical

Aditya Chandak offers valuable and practical insights, particularly in Pyspark and Data Engineering, helping greatly with interview preparation.

AI-generated based on testimonials

Are you preparing for SQL interviews? Don't miss this **FREE collection of SQL Interview Questions**, carefully curated to cover **real-world scenarios, advanced concepts, and tricky queries**.

🎯 **What's Inside?**

- ✅ Questions for beginners to advanced professionals.
- ✅ Scenario-based problems to test your skills.
- ✅ Focus on SQL optimization, joins, and query building.

💡 **Perfect for candidates aiming for top tech roles!**

Grab it now and give yourself the edge in your next SQL interview!

Don't take it from me

Hear what others have to say

Very helpful

Reyansh Srivastava
Dec 2024

I high
knowled

Nitya CloudTech
Dream.Achieve.Succeed

Free SQL Interview Preparation:

https://topmate.io/nitya_cloudtech/1403841

Data Analyst Certification:

https://nityacloudtech.com/pages/courses/NCT_Courses

Data Engineer Certification:

https://nityacloudtech.com/pages/courses/NCT_Courses

Artificial Intelligence Certification:

https://nityacloudtech.com/pages/courses/NCT_Courses

Register for Free AI Workshop:

https://nityacloudtech.com/pages/placement_training/AI_MLMasterClass

Nitya CloudTech

Dream.Achieve.Succeed