**QUIZ**

**Comprehensive Python Quiz: Manipulation, Visualization, EDA & Practice Prompts**

**1. Data Manipulation using Python**

**Q1.** What will be the output of the code below?

import pandas as pd

s = pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])

print(s[['a', 'c']])

* **A)**
* a 10
* c 30
* dtype: int64
* **B)**
* 10
* 30
* **C)** ['a', 'c']
* **D)** Error

**Answer:** A  
**Explanation:** Label-based indexing with a list returns those entries.

**Q2.** What does df.drop(['A', 'B'], axis=1) do?

* **A)** Drops rows ‘A’ and ‘B’
* **B)** Drops columns ‘A’ and ‘B’
* **C)** Selects only columns ‘A’ and ‘B’
* **D)** Raises an error

**Answer:** B  
**Explanation:** axis=1 refers to columns.

**Q3.** What is returned by df.iloc[0:2, 1:3]?

* **A)** First two rows and first two columns
* **B)** Rows 0 and 1, columns 1 and 2
* **C)** First row and first column
* **D)** Rows 1 and 2, columns 0 and 1

**Answer:** B  
**Explanation:** .iloc uses integer-based slicing (end-exclusive).

**Q4.** Which method replaces missing values in a DataFrame?

* **A)** .fillna()
* **B)** .replace()
* **C)** .dropna()
* **D)** .isna()

**Answer:** A  
**Explanation:** .fillna() fills NaNs with specified values.

**Q5.** In NumPy, what is the result of np.array([1,2,3]) + np.array([4,5,6])?

* **A)** [5,7,9]
* **B)** [14]
* **C)** [1,2,3,4,5,6]
* **D)** Error

**Answer:** A  
**Explanation:** NumPy performs element-wise addition.

**Q6.** What does df.mean(axis=0) compute?

* **A)** Row‑wise mean
* **B)** Column‑wise mean
* **C)** Mean of all values
* **D)** Nothing

**Answer:** B  
**Explanation:** axis=0 aggregates down columns.

**Q7.** Which is **not** a valid way to filter rows where A > 5?

* **A)** df[df['A'] > 5]
* **B)** df.query('A > 5')
* **C)** df.loc[df['A'] > 5]
* **D)** df.where(df['A'] < 5)

**Answer:** D  
**Explanation:** .where() keeps True values and sets others to NaN.

**Q8.** What is the shape of this array?

import numpy as np

a = np.array([[1,2,3],[4,5,6]])

print(a.shape)

* **A)** (2, 3)
* **B)** (3, 2)
* **C)** (6,)
* **D)** (2,)

**Answer:** A  
**Explanation:** Two rows, three columns.

**Q9.** To reset a DataFrame’s index permanently, which should you use?

* **A)** df.reset\_index(drop=True)
* **B)** df.reset\_index(inplace=False)
* **C)** df.reset\_index(inplace=True)
* **D)** df.index.reset()

**Answer:** C  
**Explanation:** inplace=True modifies the original DataFrame.

**Q10.** What does df['A'].apply(lambda x: x\*2) do?

* **A)** Doubles each value in column A
* **B)** Squares each value in A
* **C)** Drops column A
* **D)** Replaces NaNs

**Answer:** A  
**Explanation:** .apply() applies a function element‑wise.

**Q11.** If you concatenate two DataFrames with different columns along rows (axis=0), what happens?

* **A)** Only common columns are kept
* **B)** Union of columns is formed, missing entries become NaN
* **C)** Error is raised
* **D)** Rows are appended ignoring columns

**Answer:** B  
**Explanation:** Pandas unions columns and fills absent ones with NaN.

**Q12.** What is the output of np.arange(3, 10, 3)?

* **A)** [3, 6, 9]
* **B)** [4, 7, 10]
* **C)** [3, 6, 9, 12]
* **D)** [3, 10]

**Answer:** A  
**Explanation:** Start at 3, step of 3, stop before 10.

**Q13.** What does df.set\_index('A') do?

* **A)** Deletes column A
* **B)** Makes column A the new index
* **C)** Sorts by A
* **D)** Renames A

**Answer:** B  
**Explanation:** Moves A into the DataFrame’s index.

**Q14.** Which function stacks multiple DataFrames vertically (row‑wise)?

* **A)** merge()
* **B)** concat(axis=0)
* **C)** join()
* **D)** pivot()

**Answer:** B  
**Explanation:** concat with axis=0 appends rows.

**Q15.** When applied on a Series, what does .isin([1,2,3]) return?

* **A)** Checks if any value matches 1,2,3 anywhere
* **B)** Boolean Series indicating membership per element
* **C)** Series with values 1,2,3 removed
* **D)** Error

**Answer:** B  
**Explanation:** It yields True/False for each element.

**2. Data Visualization using Python**

**Q1.** What does plt.show() do?

* **A)** Saves the plot as a file
* **B)** Displays the plot window
* **C)** Closes all figures
* **D)** Draws a histogram

**Answer:** B  
**Explanation:** Renders and displays the figure(s).

**Q2.** What is the default plot type for df.plot() (Pandas)?

* **A)** Bar plot
* **B)** Scatter plot
* **C)** Line plot
* **D)** Pie chart

**Answer:** C  
**Explanation:** It defaults to a line plot.

**Q3.** In plt.plot(), which parameter sets the line color to red?

* **A)** color='red'
* **B)** marker='red'
* **C)** c='blue'
* **D)** color='green'

**Answer:** A  
**Explanation:** The color argument specifies line color.

**Q4.** What does sns.pairplot() do?

* **A)** Plots univariate histograms
* **B)** Scatter plots for every pair of numeric features
* **C)** Time-series plots
* **D)** Pie charts for each column

**Answer:** B  
**Explanation:** It visualizes pairwise relationships and distributions.

**Q5.** In Matplotlib, what does plt.subplot(2,1,1) specify?

* **A)** 2 rows, 1 column grid, first subplot
* **B)** 2 plots in a single row
* **C)** Single plot figure
* **D)** Error

**Answer:** A  
**Explanation:** Subplot grid of 2×1; place on position 1.

**Q6.** Which library creates interactive plots by default?

* **A)** Matplotlib
* **B)** Seaborn
* **C)** Plotly
* **D)** Pandas

**Answer:** C  
**Explanation:** Plotly’s figures are interactive out‑of‑the‑box.

**Q7.** What does sns.heatmap() visualize?

* **A)** Only missing data
* **B)** A correlation matrix or any 2D grid of values
* **C)** Bar charts
* **D)** Scatter plots

**Answer:** B  
**Explanation:** It color-codes values in a matrix.

**Q8.** What does plt.hist(data, bins=5) do?

* **A)** Plots 5 separate bar charts
* **B)** Draws a histogram with 5 bins
* **C)** Randomly splits data into 5 groups
* **D)** None of the above

**Answer:** B  
**Explanation:** bins=5 specifies number of intervals.

**Q9.** In Plotly, which function creates a box plot?

* **A)** go.Box()
* **B)** px.histogram()
* **C)** go.Pie()
* **D)** px.line()

**Answer:** A  
**Explanation:** go.Box() generates box-and-whisker plots.

**Q10.** Which command adds a title to a Matplotlib figure?

* **A)** plt.name()
* **B)** plt.header()
* **C)** plt.title()
* **D)** plt.caption()

**Answer:** C  
**Explanation:** Use plt.title("Your Title").

**Q11.** In Seaborn’s sns.scatterplot(), what does the hue parameter do?

* **A)** Adds a 3D effect
* **B)** Changes the x‑axis values
* **C)** Colors points by category
* **D)** Changes marker size

**Answer:** C  
**Explanation:** hue maps a categorical variable to color.

**Q12.** How do you save a Matplotlib figure as a PNG file?

* **A)** plt.save('plot.png')
* **B)** plt.savefig('plot.png')
* **C)** plt.store('plot.png')
* **D)** plt.write('plot.png')

**Answer:** B  
**Explanation:** savefig() writes the current figure to disk.

**Q13.** What does sns.countplot() display?

* **A)** Bar chart of value counts for a categorical variable
* **B)** Line plot of continuous data
* **C)** Scatter plot of two numeric columns
* **D)** Box plot for numerical data

**Answer:** A  
**Explanation:** Shows frequency of each category.

**Q14.** Which function produces a violin plot in Seaborn?

* **A)** sns.violinplot()
* **B)** sns.boxplot()
* **C)** sns.catplot()
* **D)** sns.lmplot()

**Answer:** A  
**Explanation:** Violin plots combine KDE and boxplot features.

**Q15.** What does plt.figure(figsize=(10,5)) do?

* **A)** Sets figure size to 10×5 pixels
* **B)** Sets figure size to 10×5 inches
* **C)** Sets DPI level
* **D)** Causes an error

**Answer:** B  
**Explanation:** figsize is measured in inches.

**3. Exploratory Data Analysis (EDA) using Python**

**Q1.** Which method returns a quick statistical summary of a DataFrame?

* **A)** df.head()
* **B)** df.describe()
* **C)** df.info()
* **D)** df.mean()

**Answer:** B  
**Explanation:** Shows count, mean, std, min, max, quartiles.

**Q2.** What does df.info() provide that df.describe() does not?

* **A)** Histograms
* **B)** Correlations
* **C)** Column data types and non-null counts
* **D)** Standard deviation

**Answer:** C  
**Explanation:** info() reveals data types and counts of non-null entries.

**Q3.** Which method returns the number of unique values per column?

* **A)** df.nunique()
* **B)** df.count()
* **C)** df.value\_counts()
* **D)** df.unique()

**Answer:** A  
**Explanation:** .nunique() counts distinct entries in each column.

**Q4.** Which visualization helps identify multicollinearity?

* **A)** df.head()
* **B)** sns.heatmap(df.corr())
* **C)** df.isnull()
* **D)** df.groupby()

**Answer:** B  
**Explanation:** A correlation heatmap shows pairwise correlations.

**Q5.** What does df.value\_counts() return when used on a Series?

* **A)** Total number of rows
* **B)** Unique values with their frequencies
* **C)** A plot
* **D)** A histogram

**Answer:** B  
**Explanation:** Returns a Series with frequency counts.

**Q6.** What is the purpose of df.isnull().sum()?

* **A)** Replaces nulls with zero
* **B)** Drops missing values
* **C)** Counts missing values per column
* **D)** Finds duplicates

**Answer:** C  
**Explanation:** Quickly identifies where data is missing.

**Q7.** What does .corr() compute in pandas?

* **A)** Pairwise string similarity
* **B)** Pearson correlation between numeric columns
* **C)** Rank correlation only
* **D)** Covariance matrix

**Answer:** B  
**Explanation:** Default correlation method is Pearson.

**Q8.** Which Seaborn function visualizes distributions per category?

* **A)** sns.lineplot()
* **B)** sns.boxplot()
* **C)** sns.regplot()
* **D)** sns.heatmap()

**Answer:** B  
**Explanation:** Boxplots show quartiles, medians, and outliers.

**Q9.** What is the output of df.duplicated().sum()?

* **A)** A boolean array
* **B)** Number of duplicate rows
* **C)** A DataFrame
* **D)** Indices of duplicates

**Answer:** B  
**Explanation:** Counts rows flagged as duplicates.

**Q10.** Which method replaces missing values with the column mean?

* **A)** df.fillna(df.mean())
* **B)** df.replace('NaN', df.mean())
* **C)** df.dropna()
* **D)** df.mean().fillna()

**Answer:** A  
**Explanation:** Common strategy for numeric imputation.

**Q11.** When is sns.pairplot(df) most useful?

* **A)** Only with categorical variables
* **B)** Inspecting relationships among numeric features
* **C)** Time-series analysis
* **D)** Simple bar charts

**Answer:** B  
**Explanation:** Shows pairwise scatterplots and histograms.

**Q12.** What does .mode() return for a Series?

* **A)** Mean
* **B)** Median
* **C)** Most frequent value(s)
* **D)** Standard deviation

**Answer:** C  
**Explanation:** Mode is the value(s) with highest frequency.

**Q13.** What is a good practice **before** computing .corr()?

* **A)** Remove categorical columns
* **B)** Fill or drop missing values
* **C)** Sort the DataFrame
* **D)** Drop columns with few unique values

**Answer:** B  
**Explanation:** Correlation cannot be calculated on NaNs.

**Q14.** What does df.groupby('col').mean() accomplish?

* **A)** Sorts by 'col'
* **B)** Filters rows
* **C)** Computes group‑wise means
* **D)** Normalizes data

**Answer:** C  
**Explanation:** Aggregates numeric columns by group.

**Q15.** Which method identifies skewness in a numeric column?

* **A)** df['col'].skew()
* **B)** df['col'].std()
* **C)** df['col'].var()
* **D)** df['col'].sum()

**Answer:** A  
**Explanation:** Skewness measures asymmetry of distribution.

**4. EDA Practice Prompts**

1. **Sales Data Deep Dive**
   * Dataset: sales.csv (StoreID, Date, ProductCategory, UnitsSold, Revenue)
   * Tasks: Find top store by avg revenue; plot monthly trends; identify underperforming categories.
2. **Missing Data Patterns**
   * Dataset: Titanic (Age, Cabin, etc.)
   * Tasks: Identify and count missing values; visualize missingness; propose imputation strategies.
3. **Airbnb Listings Analysis**
   * Dataset: listings.csv (price, room\_type, neighbourhood, number\_of\_reviews, …)
   * Tasks: Analyze price distribution by room type; detect and handle price outliers; explore price vs. reviews.
4. **Student Performance Evaluation**
   * Dataset: students.csv (Gender, Math\_Score, Reading\_Score, Writing\_Score)
   * Tasks: Compute score correlations; compare performance by gender; check for score skewness.
5. **Movie Ratings Exploration**
   * Dataset: movies.csv (Genre, Rating, Votes, Gross, Budget)
   * Tasks: Determine top genres by avg rating; investigate budget vs. gross revenue; check multicollinearity.
6. **COVID‑19 Dataset Analysis**
   * Dataset: covid\_data.csv (Country, TotalCases, Deaths, Recovered)
   * Tasks: Calculate death/recovery rates by country; identify outliers; correlate cases vs. deaths.
7. **Bank Customer Churn**
   * Dataset: bank\_customers.csv (Age, Balance, IsActive, Exited)
   * Tasks: Identify churn factors; visualize age/activity distributions for churned vs. retained; handle imbalances.
8. **E‑commerce User Behavior**
   * Dataset: clickstream.csv (SessionID, TimeSpent, PagesVisited, Purchase)
   * Tasks: Compare time spent by purchasers vs. non‑purchasers; analyze pages visited vs. purchase rate; plot purchase rate by session length.
9. **Housing Market EDA**
   * Dataset: housing.csv (Price, Area, Bedrooms, Location)
   * Tasks: Identify locations with highest avg prices; detect price outliers via boxplots; correlate features with price.
10. **Environmental Data Analysis**
    * Dataset: air\_quality.csv (City, PM2.5, PM10, NO2, Temperature, Date)
    * Tasks: Explore seasonal pollution patterns; find city with cleanest air; analyze temperature vs. PM2.5 relationship.

Feel free to copy, distribute, or adjust as needed!