DA-AG-005 — Statistics Basics

Assignment Solutions

Question 1: What is the difference between descriptive statistics and inferential statistics? Explain with examples.

Descriptive statistics summarize and organize data (e.g., mean, median, mode, variance, charts). For example, computing the average test score of 50 students in a class.

Inferential statistics use sample data to make conclusions about a population (e.g., hypothesis tests, confidence intervals). For example, using a sample of 50 students to estimate the average score of all students in a school.

Question 2: What is sampling in statistics? Explain the differences between random and stratified sampling.

Sampling is selecting a subset of a population to draw conclusions.

- **Random Sampling:** Every member has equal chance (e.g., picking lottery tickets).
- **Stratified Sampling:** Population divided into subgroups (strata) and samples taken proportionally, ensuring representation (e.g., sampling students from each grade).

Question 3: Define mean, median, and mode. Explain why these measures of central tendency are important.

- **Mean:** Arithmetic average; sensitive to outliers.
- **Median:** Middle value; robust to skew/outliers.
- **Mode:** Most frequent value; useful for categorical data.

These help identify the 'center' of data and are essential for summarizing distributions.

Question 4: Explain skewness and kurtosis. What does a positive skew imply about the data?

- **Skewness:** Measures asymmetry.

Positive skew ⇒ right tail longer (mean > median > mode).
Negative skew ⇒ left tail longer.

- **Kurtosis:** Measures tail heaviness.

High kurtosis ⇒ more extreme outliers; low kurtosis ⇒ lighter tails.

Question 5: Python program to compute mean, median, and mode of a list.

```
numbers = [12, 15, 12, 18, 19, 12, 20, 22, 19, 19, 24, 24, 24, 26, 28]

from collections import Counter

mean_val = sum(numbers)/len(numbers)

sorted_vals = sorted(numbers)
```

```
n = len(sorted_vals)
median_val = sorted_vals[n//2] if n%2==1 else (sorted_vals[n//2-1]+sorted_vals[n//2])/2

counts = Counter(numbers)
max_freq = max(counts.values())
modes = [k for k,v in counts.items() if v==max_freq]

print("Mean:", round(mean_val,2))
print("Median:", median_val)
print("Mode(s):", modes, "with frequency", max_freq)
```

```
Mean: 19.6
Median: 19
Mode(s): [12, 19, 24] with frequency 3
```

Question 6: Compute covariance and correlation coefficient between two datasets.

```
import numpy as np
x = np.array([10,20,30,40,50], dtype=float)
y = np.array([15,25,35,45,60], dtype=float)

cov = np.cov(x,y,ddof=1)[0,1]
corr = np.corrcoef(x,y)[0,1]

print("Covariance:", round(cov,2))
print("Correlation:", round(corr,3))
```

```
Covariance: 275.0
Correlation: 0.996
```

Question 7: Boxplot for data and identification of outliers.

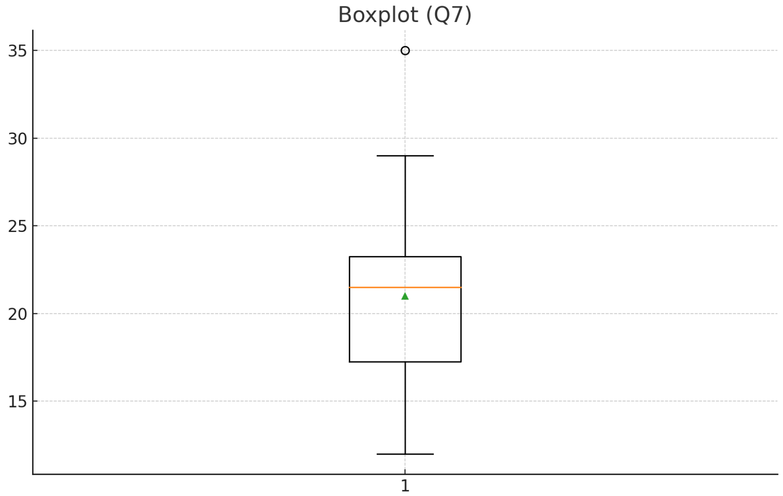
```
import numpy as np, matplotlib.pyplot as plt
data = [12,14,14,15,18,19,19,21,22,22,23,23,24,26,29,35]

q1, q2, q3 = np.percentile(data,[25,50,75])
iqr = q3-q1
lower, upper = q1-1.5*iqr, q3+1.5*iqr
outliers = [x for x in data if x<lower or x>upper]

plt.boxplot(data, showmeans=True)
plt.title("Boxplot (Q7)")
plt.savefig("/mnt/data/q7_boxplot_better.png")
plt.close()

print("Q1:", q1, "Median:", q2, "Q3:", q3)
print("IQR:", iqr)
print("Outliers:", outliers)
```

```
Q1: 17.25 Median: 21.5 Q3: 23.25
IQR: 6.0
Outliers: [35]
```



Question 8: Relationship between advertising spend and daily sales — correlation.

Covariance shows direction of relationship, while correlation standardizes it (between -1 and 1). High positive correlation indicates sales rise with advertising spend.

```
import numpy as np
ad_spend = np.array([200,250,300,400,500], dtype=float)
sales = np.array([2200,2450,2750,3200,4000], dtype=float)
corr = np.corrcoef(ad_spend, sales)[0,1]
print("Correlation:", round(corr,3))
```

```
Correlation: 0.994
```

Question 9: Distribution of customer satisfaction scores — recommended statistics and histogram.

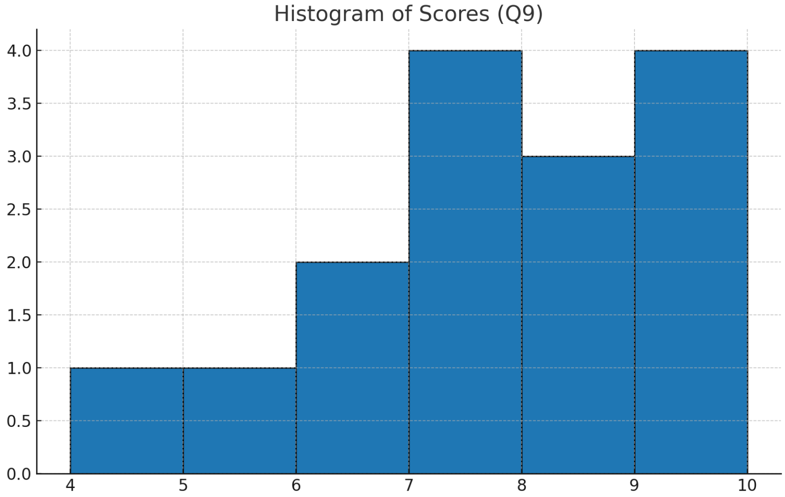
A histogram shows distribution shape. Summary statistics (mean, median, std, min, max, mode) provide insights into central tendency and spread.

```
import numpy as np, matplotlib.pyplot as plt
scores = [7,8,5,9,6,7,8,9,10,4,7,6,9,8,7]
arr = np.array(scores)

print("Mean:", round(arr.mean(),2))
print("Median:", np.median(arr))
print("Std Dev:", round(arr.std(ddof=1),2))
print("Min:", arr.min(), "Max:", arr.max())

plt.hist(arr, bins=6, edgecolor='black')
plt.title("Histogram of Scores (Q9)")
plt.savefig("/mnt/data/q9_hist_better.png")
plt.close()
```

```
Mean: 7.33
Median: 7.0
Std Dev: 1.63
Min: 4 Max: 10
```



Histogram of Scores (Q9)