Q1. Assumptions of ANOVA and violations

ANOVA requires the following assumptions:

1. **Independence of observations**
   - Observations are independent of each other
   - **Violation example:** Measuring the same subject multiple times in one-way ANOVA
2. **Normality**
   - Data in each group is approximately normally distributed
   - **Violation example:** Strongly skewed data or heavy outliers
3. **Homogeneity of variances (equal variances)**
   - Variances across groups are equal
   - **Violation example:** One group has much larger variance than others

🔴 Violations can lead to **inflated Type I errors** or **loss of power**.
✔ Remedies: data transformation, Welch's ANOVA, or non-parametric tests.

Q2. Types of ANOVA

1. **One-way ANOVA**
   - One independent variable (factor)
   - Example: Comparing test scores across 3 teaching methods
2. **Two-way ANOVA**
   - Two independent variables

    - Example: Teaching method × gender

3. **Repeated Measures ANOVA**
    - Same subjects measured multiple times
    - Example: Blood pressure before, during, and after treatment

---

Q3. Partitioning of variance in ANOVA

Total variation in data is divided as:

SST=SSB (Between)+SSW (Within)\text{SST} = \text{SSB (Between)} + \text{SSW (Within)}SST=SSB (Between)+SSW (Within)

- **Between-group variance:** due to treatment/factor
- **Within-group variance:** due to random error

📌 Important because the **F-statistic** is based on this ratio:

F=Between varianceWithin varianceF = \frac{\text{Between variance}}{\text{Within variance}}F=Within varianceBetween variance

---

Q4. Calculating SST, SSB, SSW in one-way ANOVA (Python)

```
import numpy as np

groups = {
    'A': np.array([5, 6, 7]),
    'B': np.array([8, 9, 6]),
    'C': np.array([4, 5, 3])
}

all_data = np.concatenate(list(groups.values()))
grand_mean = np.mean(all_data)
```

```
# SST
SST = np.sum((all_data - grand_mean)**2)

# SSB
SSB = sum(len(v) * (np.mean(v) - grand_mean)**2 for v in
groups.values())

# SSW
SSW = sum(np.sum((v - np.mean(v))**2) for v in groups.values())

SST, SSB, SSW
```

Q5. Two-way ANOVA main & interaction effects (Python)

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols

df = pd.DataFrame({
    'score': [80,82,78,85,88,90,75,77,74,79],
    'method': ['A','A','A','B','B','B','A','A','B','B'],
    'gender': ['M','F','M','F','M','F','F','M','F','M']
})

model = ols('score ~ C(method) * C(gender)', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
anova_table
```

Q6. Interpretation of F = 5.23, p = 0.02

- p-value < 0.05 → **Reject null hypothesis**
- There is a **statistically significant difference** between group means

- ANOVA does **not** tell which groups differ → post-hoc test required

---

Q7. Missing data in repeated measures ANOVA

Common methods:

1. **Listwise deletion** – remove subjects with missing data
   ❌ Reduces sample size
2. **Mean/Last observation imputation**
   ❌ Underestimates variability
3. **Mixed-effects models (preferred)**
   ✔ Handles missing data appropriately

Q8. Post-hoc tests after ANOVA

Test	When to use
Tukey HSD	Equal variances, all pairwise comparisons
Bonferroni	Conservative, many comparisons
Scheffé	Unequal sample sizes
Dunnett	Compare groups to a control

📌 Example: ANOVA shows diet differences → use Tukey HSD to find which diets differ.

Q9. One-way ANOVA: Diet weight loss (Python)

```
from scipy.stats import f_oneway

diet_A = [5, 6, 7, 8]
diet_B = [3, 4, 5, 4]
```

```
diet_C = [6, 7, 8, 9]

F, p = f_oneway(diet_A, diet_B, diet_C)
F, p
```

Interpretation

- If $p < 0.05$ → Significant difference in mean weight loss among diets

---

Q10. Two-way ANOVA: Software × Experience (Python)

```
df = pd.DataFrame({
    'time': [30,28,35,40,38,42,25,27,29,34,36,33],
    'software': ['A','A','A','B','B','B','C','C','C','A','B','C'],
    'experience': ['N','E','N','E','N','E','N','E','N','E','N','E']
})

model = ols('time ~ C(software) * C(experience)', data=df).fit()
sm.stats.anova_lm(model, typ=2)
```

Interpretation

- Main effect: software or experience affects time
- Interaction: software effectiveness depends on experience

Q11. Two-sample t-test & post-hoc

```
from scipy.stats import ttest_ind

control = [70,72,68,75,71]
experimental = [78,80,82,79,81]

t, p = ttest_ind(control, experimental)
```

```
t, p
```

- If $p < 0.05$ → significant difference
- Since only two groups → **no post-hoc needed**
  (Post-hoc is for ≥3 groups)

Q12. Repeated Measures ANOVA: Retail store sales

```
from statsmodels.stats.anova import AnovaRM

df = pd.DataFrame({
    'day': list(range(1,31))*3,
    'store': ['A']*30 + ['B']*30 + ['C']*30,
    'sales': np.random.normal(100, 10, 90),
    'subject': list(range(1,31))*3
})

anova = AnovaRM(df, 'sales', 'subject', within=['store'])
anova.fit()
```

If significant:

- Conduct **pairwise comparisons** with Bonferroni correction

---