**Q1: Assumptions of ANOVA and examples of violations:**

Assumptions of ANOVA:

1. **Independence:** Observations within each group are independent of each other.
2. **Normality:** The residuals (errors) are normally distributed within each group.
3. **Homogeneity of variances:** The variance of the residuals is the same for all groups.

Examples of violations:

1. **Independence:** Violated when observations within groups are not independent, such as repeated measures or nested designs where individuals contribute to multiple groups.
2. **Normality:** Violated when the residuals are not normally distributed, which can occur with skewed or heavily-tailed distributions.
3. **Homogeneity of variances:** Violated when the variance of the residuals is not consistent across groups, which can occur with unequal sample sizes or unequal variances.

**Q2: Types of ANOVA and their usage:**

1. **One-way ANOVA:** Used when comparing the means of three or more independent groups on a single dependent variable.
2. **Two-way ANOVA:** Used when there are two independent variables (factors) and their interaction effect on a dependent variable needs to be examined.
3. **Repeated measures ANOVA:** Used when the same subjects are measured at multiple time points or under multiple conditions.

**Q3: Partitioning of variance in ANOVA:** Partitioning of variance involves decomposing the total variance observed in the data into different components attributable to different sources or factors. Understanding this concept is important as it helps to determine the relative contribution of each factor (e.g., treatment effects, error) to the overall variance observed in the dependent variable.

**Q4: Calculation of SST, SSE, and SSR in one-way ANOVA using Python:** To calculate these sums of squares, you'll need the group means, overall mean, and individual data points. Here's a Python code snippet using numpy:

pythonCopy code

import numpy as np # Assuming data is stored in a dictionary where keys are group names and values are lists of observations data = {'A': [data\_points\_group\_A], 'B': [data\_points\_group\_B], 'C': [data\_points\_group\_C]} # Calculate overall mean overall\_mean = np.mean(np.concatenate(list(data.values()))) # Calculate SST SST = sum([(x - overall\_mean)\*\*2 for group\_data in data.values() for x in group\_data]) # Calculate SSE SSE = sum([(x - np.mean(group\_data))\*\*2 for group\_data in data.values() for x in group\_data]) # Calculate SSR SSR = SST - SSE

**Q5: Calculation of main and interaction effects in two-way ANOVA using Python:** Main and interaction effects can be calculated using statistical libraries like statsmodels or scipy in Python. Here's a simplified example using statsmodels:

import statsmodels.api as sm from statsmodels.formula.api import ols # Assuming data is in a pandas DataFrame with columns: 'dependent\_variable', 'independent\_variable1', 'independent\_variable2' model = ols('dependent\_variable ~ independent\_variable1 \* independent\_variable2', data=data).fit() anova\_table = sm.stats.anova\_lm(model, typ=2) # Main effects and interaction effects can be extracted from the 'anova\_table'

**Q6: Interpretation of one-way ANOVA results:**

Given an F-statistic of 5.23 and a p-value of 0.02 from a one-way ANOVA:

* The F-statistic tests the null hypothesis that the group means are equal against the alternative hypothesis that at least one group mean is different.
* Since the p-value (0.02) is less than the significance level (α = 0.05), we reject the null hypothesis.
* This indicates that there is sufficient evidence to suggest that at least one group mean is different from the others.
* However, it does not specify which group(s) are different from each other.

**Q7: Handling missing data in repeated measures ANOVA:**

* One common approach is to use techniques like mean imputation or linear interpolation to replace missing values with estimates based on available data.
* Another approach is to use statistical methods for handling missing data, such as multiple imputation or maximum likelihood estimation.
* The consequences of using different methods include potential bias in parameter estimates, underestimation of standard errors, and reduced power to detect effects.

**Q8: Common post-hoc tests after ANOVA:**

1. **Tukey's Honestly Significant Difference (HSD):** Used to identify which specific groups differ significantly from each other. It controls the familywise error rate.
2. **Bonferroni correction:** Adjusts the significance level for multiple comparisons to avoid Type I errors.
3. **Sidak correction:** Similar to Bonferroni correction but often more powerful.
4. **Duncan's multiple range test:** Compares all possible pairs of group means to identify homogeneous subsets.

Example scenario: Suppose an ANOVA comparing the effectiveness of four different teaching methods shows a significant difference. A post-hoc test like Tukey's HSD can be used to determine which pairs of teaching methods are significantly different from each other.

**Q9: One-way ANOVA in Python to compare mean weight loss of three diets:**

pythonCopy code

import numpy as np from scipy.stats import f\_oneway # Sample data (mean weight loss for each diet) diet\_A = [data\_points\_diet\_A] # Replace data\_points\_diet\_A with actual data diet\_B = [data\_points\_diet\_B] # Replace data\_points\_diet\_B with actual data diet\_C = [data\_points\_diet\_C] # Replace data\_points\_diet\_C with actual data # Perform one-way ANOVA f\_statistic, p\_value = f\_oneway(diet\_A, diet\_B, diet\_C) # Report results print("F-statistic:", f\_statistic) print("p-value:", p\_value) # Interpretation if p\_value < 0.05: print("There is a significant difference in mean weight loss between the three diets.") else: print("There is no significant difference in mean weight loss between the three diets.")

**Q10: Two-way ANOVA in Python to compare software programs and experience levels:**

pythonCopy code

import pandas as pd from statsmodels.formula.api import ols from statsmodels.stats.anova import anova\_lm # Assuming data is in a DataFrame with columns: 'time', 'software', 'experience' data = pd.read\_csv('data.csv') # Replace 'data.csv' with actual file path # Perform two-way ANOVA model = ols('time ~ C(software) \* C(experience)', data=data).fit() anova\_table = anova\_lm(model, typ=2) # Report results print(anova\_table)

Interpret the results from the ANOVA table to determine if there are significant main effects or interaction effects between software programs and experience levels.

**Q11: Two-sample t-test in Python to compare test scores between control and experimental groups:**

pythonCopy code

from scipy.stats import ttest\_ind # Sample data (test scores for control and experimental groups) control\_group = [test\_scores\_control] # Replace test\_scores\_control with actual data experimental\_group = [test\_scores\_experimental] # Replace test\_scores\_experimental with actual data # Perform two-sample t-test t\_statistic, p\_value = ttest\_ind(control\_group, experimental\_group) # Report results print("T-statistic:", t\_statistic) print("p-value:", p\_value) # Interpretation if p\_value < 0.05: print("There is a significant difference in test scores between the control and experimental groups.") else: print("There is no significant difference in test scores between the control and experimental groups.")

**Q12: Repeated measures ANOVA in Python to compare daily sales of three retail stores:**

import pandas as pd from statsmodels.stats.anova import AnovaRM # Assuming data is in a DataFrame with columns: 'day', 'store', 'sales' data = pd.read\_csv('sales\_data.csv') # Replace 'sales\_data.csv' with actual file path # Perform repeated measures ANOVA rm\_anova = AnovaRM(data, 'sales', 'day', within=['store']) results = rm\_anova.fit() # Report results print(results.summary())

Interpret the results from the summary to determine if there are significant differences in sales between the three stores over the 30 days. If the results are significant, follow up with a post-hoc test to determine which store(s) differ significantly from each other.