# Wine and Obesity Data Analysis

Adrian T. and Olivia K.

2025-12-09

## Table of contents

## 1. Dataset Overview

### A. Wine Quality Dataset

- **Collection Year:** Dataset was collected in 2009.
- **Study Type:** The study is observational. Each wine was independently sampled.
- **Description:** Physiochemical properties for red and white Portuguese "Vinho Verde" wines.

- **Size:** 6497 rows, 12 columns

- **Variables:** `fixed_acidity`, `volatile_acidity`, `citric_acid`, `residual_sugar`, `chlorides`, `free_sulfur_dioxide`, `total_sulfur_dioxide`, `density`, `ph`, `sulphates`, `alcohol`, `quality`, `color`

Table 1: Dataset 1 Variables

| Variable Name | Type | Description |
|---|---|---|
| fixed_acidity | Continuous | |
| volatile_acidity | Continuous | |
| citric_acid | Continuous | |
| residual_sugar | Continuous | |
| chlorides | Continuous | |
| free_sulfur_dioxide | Continuous | |
| total_sulfur_dioxide | Continuous | |
| density | Continuous | |
| pH | Continuous | |
| sulphates | Continuous | |
| alcohol | Continuous | |
| quality | Integer | score between 0 and 10 |
| color | Categorical | red or white |

### Sampling Distribution of Density

The following histogram shows the sampling distribution of the **sample mean of the `density` variable**.

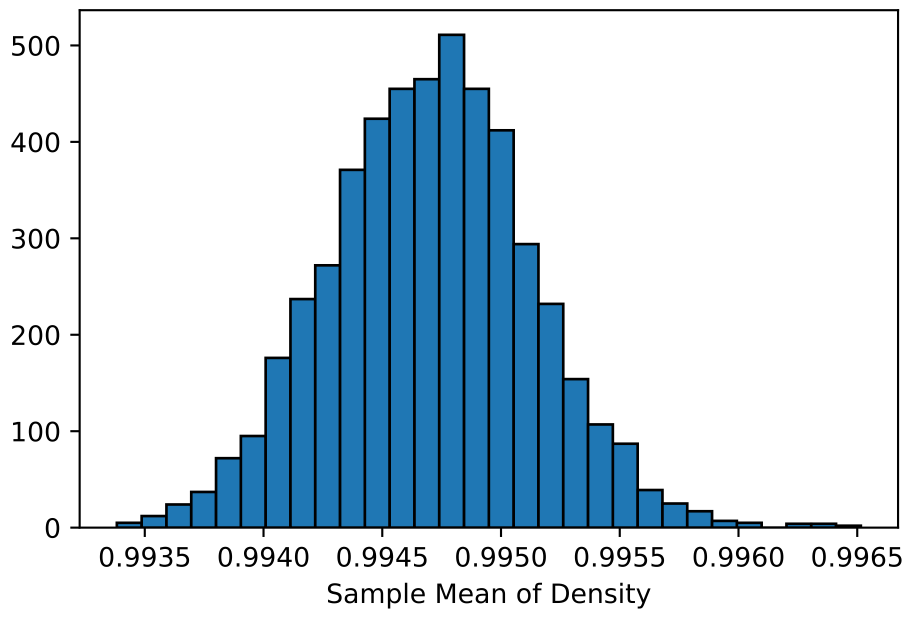

Figure 1: Sampling Distribution of Sample Mean (Density)

Figure 1 shows the sampling distribution of the sample mean of wine densities. There is a heavy clustering with similar frequencies from 0.9945 to 0.995 g/ml. The right tail appears thinner than the left tail but with small frequencies. The shape of the sampling distribution is approximately normal with little to no skew.

## B. Obesity Dataset

- **Collection Year:** The dataset was collected in 2019.
- **Study Type:** The study is observational, each person was measured independently.
- **Description:** Estimation of obesity levels in individuals from Mexico, Peru, and Colombia, based on eating habits and physical condition.

- **Size:** 2111 rows, 16 columns

- **Variables:** `Gender`, `Age`, `Height`, `Weight`, `Family_history_with_overweight`, `FAVC`, `FCVC`, `NCP`, `CAEC`, `SMOKE`, `CH2O`, `SCC`, `FAF`, `TUE`, `CALC`, `MTRANS`, `NObeyesdad`

Table 2: Dataset 2 Variables

| Variable Name | Type | Description | Units |
|---|---|---|---|
| Gender | Categorical | Biological sex | - |
| Age | Continuous | Age in years | Years |
| Height | Continuous | Height | Meters |
| Weight | Continuous | Weight | Kg |
| Family_history_with_overweight | Binary | Family history of overweight | - |
| FAVC | Binary | Frequent high-calorie food intake | - |
| FCVC | Integer | Frequency of vegetable consumption | - |
| NCP | Continuous | Number of main meals per day | Count |
| CAEC | Categorical | Snacking between meals | - |
| SMOKE | Binary | Smokes or not | - |
| CH20 | Binary | Monitors calorie intake | - |
| SCC | Continuous | Physical activity frequency | Hours/Week |
| FAF | Continuous | Daily screen/technology use | Hours |
| TUE | Continuous | Daily electronics use | Hours |
| CALC | Categorical | Alcohol consumption | - |
| MTRANS | Categorical | Mode of transportation | - |
| NObeyesdad | Categorical | Obesity classification | - |

### Sampling Distribution of Weight

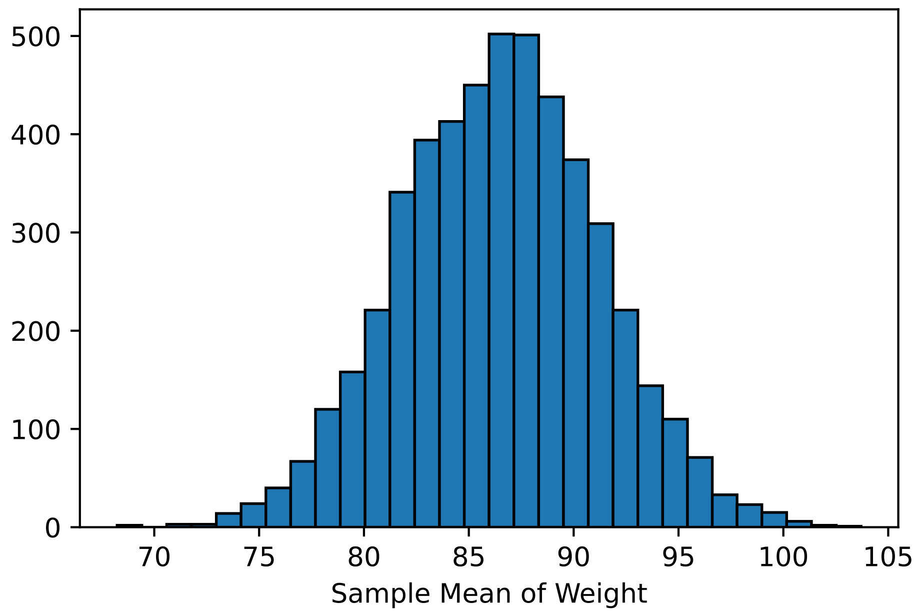

Figure 2: Sampling Distribution of Sample Mean (Weight)

Figure 2 shows the sampling distribution of the sample mean of individual weights. The histogram is approximately symmetric and bell-shaped, resembling the normal distribution. The sample clusters around 87 and 88 Kg, with a slightly larger frequency on the left of the center than the right. There is no noticeable skewness.

---

## 2. One-Sample T-Test: Wine Alcohol Content

### 2.1 Research Question and Hypothesis

**Research Question:** Is the average alcohol content of red and white wine greater than 10.5%?

**Hypotheses:** The null hypothesis states that the population mean alcohol content is equal to 10.5%, while the alternate hypothesis claims the population mean alcohol content of red and white wine is greater than 10.5%.

## 2.2 Assumptions

## 2.3 Test

**Results:**
In the one-sample t-test, the sample mean was 10.492, with a corresponding t-statistic of $-0.5541$. At a significance level of $= 0.05$, the critical t-value for a one-tailed test was 1.6451. The resulting one-tailed p-value was 0.7102. Since the test statistic does not exceed the critical value and the p-value is greater than , there is insufficient evidence to reject the null hypothesis. Lastly, the 95% confidence interval for the population mean was (10.4628, 10.5208).

## 2.4 Conclusion:

Since the critical t-value was greater than $\alpha$ , we fail to reject the null hypothesis. There is insufficient evidence that the mean alcohol content exceeds 10.5%.

---

## 3. Bootstrap Approach: Wine pH

### 3.1 Bootstrap Distribution

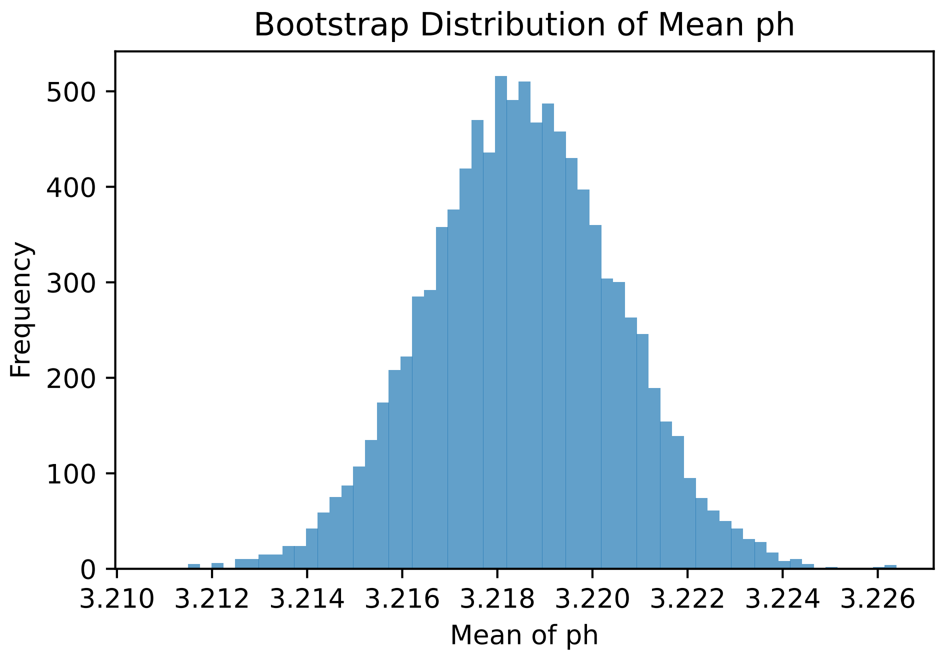

Figure 3: Bootstrap Distribution of Wine pH

## 3.2 QQ Plot for Bootstrap

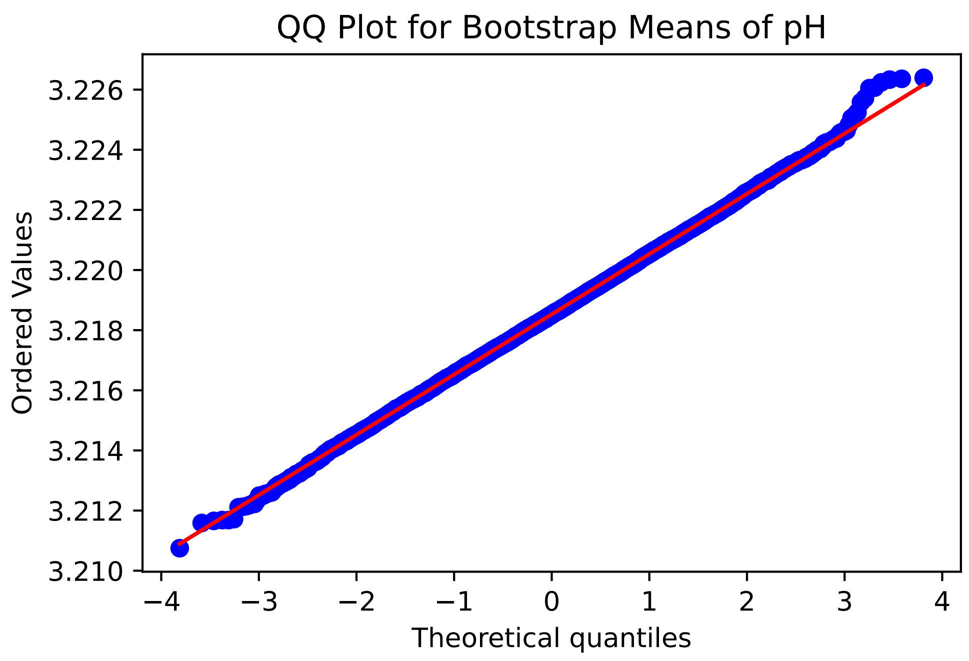

Figure 4: QQ Plot of Bootstrap Means (pH)

## 3.3 Bootstrap Confidence Interval

```
The bootstrap test CI was calculated to be (3.2146051639218105, 3.2224704863783282)
```

**Conclusion:** Using an $\alpha$ of 0.05, We are 95% confident that the true population mean pH lies between 3.215 and 3.222.

## 4. Analysis of Variance (Wine Dataset)

### 4.1 F Test

### 4.2 ANOVA Table

### 4.3 AVOVA Assumptions Check

### 4.4 Conclusion

---

## 5. Multiple Comparisons (Obesity Dataset)

### 5.1 ANOVA Model and Assumptions

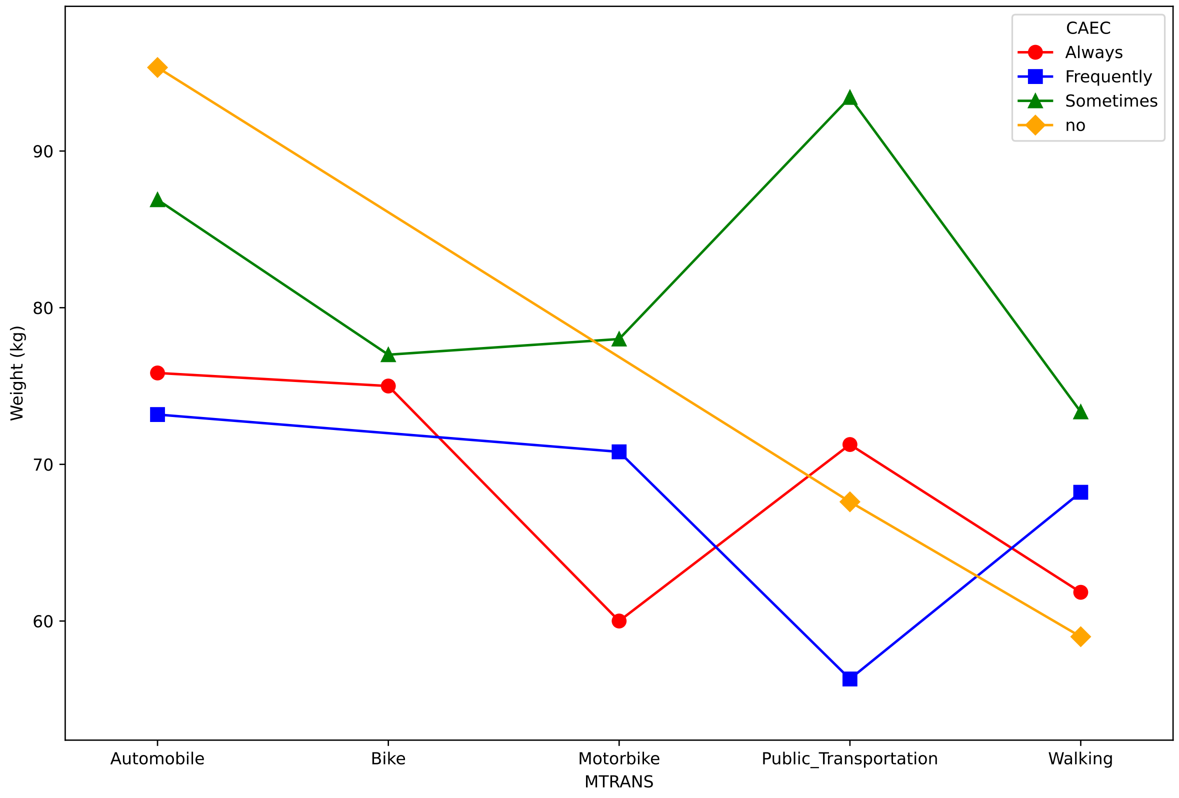

Figure 5: Two-way ANOVA with interaction: Weight ~ MTRANS * CAEC

- **Interpretation:** Based on Figure 5, Weight differs across the different MTRANS (Method of Transportation) categories. The lines for different CAEC (Snacking between Meals) overlap, suggesting a relationship exists between MTRANS and CAEC levels.

## 5.2 QQ Plot and Residual Analysis:

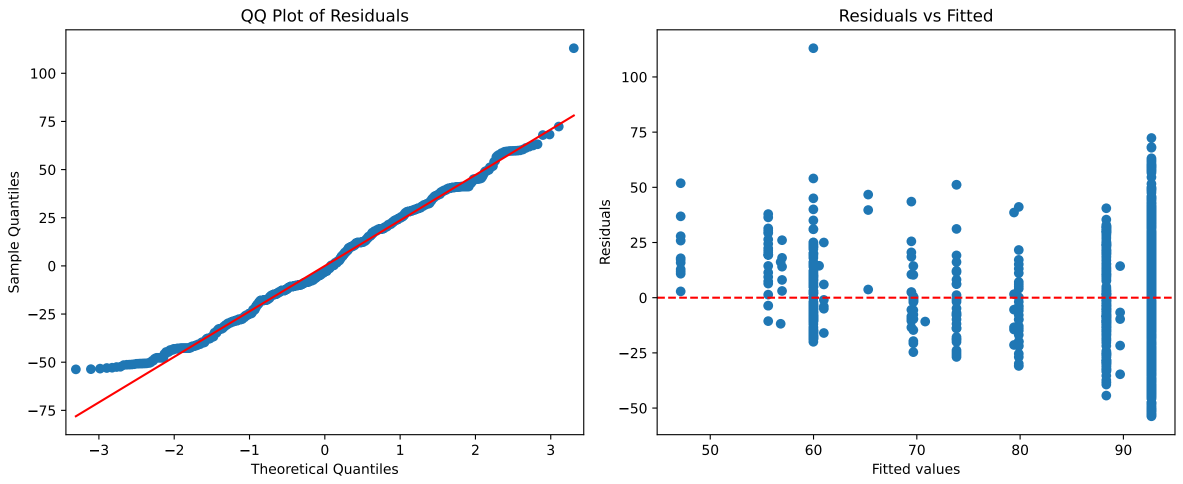

Figure 6: QQ Plot and Residual Plot

- The left plot in Figure 6 is the QQ plot, which shows approximate normality as the data falls around the line. There is a single datapoint on the far right tail that is deviated from the normal line, indicating a potential outlier. On the left end of the normal line, the datapoints appear to have a curvature away from the normal.
- The right plot in Figure 6 shows the corresponding residual plot. The datapoints shows no clear pattern, however there is a singular point that has an abnormally high residual around (60, 120), which is most likely the outlier that was observed in the QQ plot.
- **Conclusion:** Overall, the ANOVA assumptions of normality, independence, and equal variance are satisfied.

## 5.3 F-Tests for Factors

Table 3: ANOVA results for MTRANS and CAEC factors

| Factor | F-statistic | F-critical | p-value | Conclusion |
|---|---|---|---|---|
| MTRANS | 6.85 | 2.376 | $1.78 \times 10^{-5}$ | Reject H : At least one group mean differs |
| CAEC | 149.69 | 2.609 | $1.11 \times 10^{-16}$ | Reject H : At least one group mean differs |

**Interpretation:** Both MTRANS and CAEC significantly affect Weight; differences between group means are unlikely due to chance.

## 5.4 Pairwise Comparisons

Table 4: Tukey HSD Pairwise Comparison for MTRANS

| | Transportion A | Transportion B | meandiff | p-adj | lower | upper | reject |
|---|---|---|---|---|---|---|---|
| 0 | Automobile | Bike | -9.1933 | 0.8866 | -36.2772 | 17.8906 | False |
| 1 | Automobile | Motorbike | -12.8167 | 0.4894 | -34.5151 | 8.8817 | False |
| 2 | Automobile | Public Transportation | 1.5791 | 0.7845 | -2.1981 | 5.3563 | False |
| 3 | Automobile | Walking | -15.3115 | 0.0003 | -25.3800 | -5.2430 | True |
| 4 | Bike | Motorbike | -3.6234 | 0.9985 | -38.0069 | 30.7601 | False |
| 5 | Bike | Public Transportation | 10.7724 | 0.8109 | -16.1659 | 37.7107 | False |
| 6 | Bike | Walking | -6.1182 | 0.9772 | -34.6275 | 22.3911 | False |
| 7 | Motorbike | Public Transportation | 14.3958 | 0.3584 | -7.1206 | 35.9122 | False |
| 8 | Motorbike | Walking | -2.4948 | 0.9984 | -25.9482 | 20.9586 | False |
| 9 | Public Transportation | Walking | -16.8906 | 0.0000 | -26.5606 | -7.2206 | True |

- **Transportion A, Transportion B:** The two transportation categories being compared.
- **meandiff**: Difference in group means (B minus A).
- **p-adj**: P-value adjusted for multiple comparisons to control the family-wise error rate.
- **lower, upper**: Lower and upper bounds of the 95% confidence interval for the mean difference.
- **reject**: Indicates whether the null hypothesis of equal means is rejected at $\alpha = 0.05$.

**Conclusion:** From Table 4, the Tukey HSD pairwise comparisons revealed a few significant differences between pairs of transportation methods. There were differences between Automobile vs. Walking and also Public Transportation vs. Walking. The mean difference in Weight were 15.3 and 16.9 respectivly. These results suggest that walking is assosiated with a lower average weight.

Table 5: Tukey HSD Pairwise Comparison for CAEC

| | CAEC A | CAEC B | meandiff | p-adj | lower | upper | reject |
|---|---|---|---|---|---|---|---|
| 0 | Always | Frequently | -12.2049 | 0.0041 | -21.4825 | -2.9273 | True |
| 1 | Always | Sometimes | 20.2698 | 0.0000 | 11.7416 | 28.7980 | True |
| 2 | Always | no | -2.1881 | 0.9659 | -14.1876 | 9.8115 | False |
| 3 | Frequently | Sometimes | 32.4747 | 0.0000 | 28.2813 | 36.6681 | True |
| 4 | Frequently | no | 10.0168 | 0.0322 | 0.5911 | 19.4425 | True |
| 5 | Sometimes | no | -22.4579 | 0.0000 | -31.1469 | -13.7688 | True |

- **CAEC A, CAEC B:** The two frequencies of snacking categories being compared.
- **meandiff**: Difference in group means (B minus A).
- **p-adj**: P-value adjusted for multiple comparisons to control the family-wise error rate.
- **lower, upper**: Lower and upper bounds of the 95% confidence interval for the mean difference.
- **reject**: Indicates whether the null hypothesis of equal means is rejected at $\alpha = 0.05$.

**Conclusion:**

---

## 6. References

Cortez, Paulo, et al. “Wine Quality.” UCI Machine Learning Repository, 2009, https://doi.org/10.24432/C56S3T.

“Estimation of Obesity Levels Based On Eating Habits and Physical Condition .” UCI Machine Learning Repository, 2019, https://doi.org/10.24432/C5H31Z.

—‘

## 7. Code Appendix

```
from ucimlrepo import fetch_ucirepo
import pandas as pd
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.graphics.factorplots import interaction_plot
import statsmodels.api as sm
import statsmodels.formula.api as smf
import seaborn as sns

# Fetch the Wine Quality dataset
wine_quality = fetch_ucirepo(id=186)

# Features and targets as pandas DataFrames
X = wine_quality.data.features
y = wine_quality.data.targets

estimation_of_obesity_levels_based_on_eating_habits_and_physical_condition = (
    fetch_ucirepo(id=544)
)
X2 = (
    estimation_of_obesity_levels_based_on_eating_habits_and_physical_condition.d
ata.features
)
y2 = (
    estimation_of_obesity_levels_based_on_eating_habits_and_physical_condition.d
ata.targets
)


# --------------------------------
# Sampling Distributions (Both Datasets)
# --------------------------------
def plot_histogram_q1():
    data = X["density"].values

    sample_size = 50
    num_samples = 5000

    sample_means = []
    for i in range(num_samples):
        sample = np.random.choice(data, size=sample_size, replace=True)
        sample_means.append(np.mean(sample))

    plt.hist(sample_means, bins=30, edgecolor="black")
    plt.xlabel("Sample Mean of Density")
    # plt.title("Sampling Distribution of Sample Mean (Density)")
    plt.show()


def plot_histogram_q2():
    data = X2["Weight"].values
    sample_size = 30
    num_samples = 5000
    sample_means = []
    for i in range(num_samples):
        sample = np.random.choice(data, size=sample_size, replace=True)
        sample_means.append(np.mean(sample))
    plt.hist(sample_means, bins=30, edgecolor="black")
    plt.xlabel("Sample Mean of Weight")
    # plt.title("Sampling Distribution of Sample Mean (Weight)")
    plt.show()


# -------------------------------
# Bootstrap (Wine Quality Dataset)
# -------------------------------
def bootstrap():
    # wine_quality = fetch_ucirepo(id=186)
    X = wine_quality.data.features
    df = X["pH"].values

    iter = 10000
    runs = np.zeros(iter)

    for i in range(iter):
        sample = np.random.choice(df, size=len(df), replace=True)
        runs[i] = np.mean(sample)
    return runs


def plot_bootstrap(runs):
    plt.hist(x=runs, bins="auto", alpha=0.7)
    plt.xlabel("Mean of ph")
    plt.ylabel("Frequency")
    plt.title("Bootstrap Distribution of Mean ph")
    plt.show()


def bootstrap_qq(vals):
    stats.probplot(vals, dist="norm", plot=plt)
    plt.title("QQ Plot for Bootstrap Means of pH")
    plt.show()


# --------------------------------
# ANOVA and Interaction Model (Obesity Dataset)
# --------------------------------
def mult_comparisons():
    # fetch dataset
    # estimation_of_obesity_levels_based_on_eating_habits_and_physical_condition
= fetch_ucirepo(id=544)

    # data (as pandas dataframes)
    # X = estimation_of_obesity_levels_based_on_eating_habits_and_physical_condi
tion.data.features
    # y = estimation_of_obesity_levels_based_on_eating_habits_and_physical_condi
tion.data.targets

    # metadata
    # print(estimation_of_obesity_levels_based_on_eating_habits_and_physical_con
dition.metadata)

    # variable information
    # print(estimation_of_obesity_levels_based_on_eating_habits_and_physical_con
dition.variables)

    df = pd.concat([X2, y2], axis=1)
    df.columns = list(X2.columns) + list(y2.columns)

    df["MTRANS_str"] = df["MTRANS"].astype(str)
    df["CAEC_str"] = df["CAEC"].astype(str)

    # Interaction plot
    fig, ax = plt.subplots(figsize=(12, 8))
    interaction_plot(
        df["MTRANS"],
        df["CAEC"],
        df["Weight"],
        colors=["red", "blue", "green", "orange"],
        markers=["o", "s", "^", "D"],
        ms=8,
        ax=ax,  # pass axes to avoid auto (a)/(b)
    )
    ax.set_xlabel("MTRANS")
    ax.set_ylabel("Weight (kg)")
    # ax.set_title('Interaction Plot: Weight by MTRANS and CAEC')

    plt.show()

    model = smf.ols("Weight ~ C(MTRANS) + C(CAEC)", data=df).fit()

    return model


def plot_resid_fitted(model):
    residuals = model.resid
    fitted = model.fittedvalues
    # Create plots
    fig, ax = plt.subplots(1, 2, figsize=(12, 5))

    # 1. Q-Q plot of residuals
    sm.qqplot(residuals, line="s", ax=ax[0])
    ax[0].set_title("QQ Plot of Residuals")

    # 2. Residuals vs Fitted plot
    ax[1].scatter(fitted, residuals)
    ax[1].axhline(0, color="red", linestyle="--")
    ax[1].set_xlabel("Fitted values")
    ax[1].set_ylabel("Residuals")
    ax[1].set_title("Residuals vs Fitted")

    plt.tight_layout()
    plt.show()


def hypothesis_test(model):
    anova_table = sm.stats.anova_lm(model, typ=2)
    # 4. Hypothesis test for equality of means
    # Calculate MSA and MSB
    msa = anova_table["sum_sq"]["C(MTRANS)"] / anova_table["df"]["C(MTRANS)"]
    msb = anova_table["sum_sq"]["C(CAEC)"] / anova_table["df"]["C(CAEC)"]
    mse = anova_table["sum_sq"]["Residual"] / anova_table["df"]["Residual"]

    # Calculate F-statistics
    f_stat_a = msa / mse
    f_stat_b = msb / mse
    print(f"F-statistic for MTRANS: {f_stat_a}")
    print(f"F-statistic for CAEC: {f_stat_b}")

    # Critical F-value
    alpha = 0.05
    f_crit_a = stats.f.ppf(
        1 - alpha, anova_table["df"]["C(MTRANS)"], anova_table["df"]["Residual"]
    )
    f_crit_b = stats.f.ppf(
        1 - alpha, anova_table["df"]["C(CAEC)"], anova_table["df"]["Residual"]
    )
    print(f"Critical F-value for MTRANS: {f_crit_a}")
    print(f"Critical F-value for CAEC: {f_crit_b}")

    # p-values
    p_value_a = 1 - stats.f.cdf(
        f_stat_a, anova_table["df"]["C(MTRANS)"], anova_table["df"]["Residual"]
    )
    p_value_b = 1 - stats.f.cdf(
        f_stat_b, anova_table["df"]["C(CAEC)"], anova_table["df"]["Residual"]
    )
    print(f"P-value for MTRANS: {p_value_a}")
    print(f"P-value for CAEC: {p_value_b}")

    # Conclusion
    if f_stat_a > f_crit_a:
        print(
            "Reject null hypothesis for MTRANS: At least one group mean is
different."
        )
    else:
        print(
            "Fail to reject null hypothesis for MTRANS: No significant
difference between group means."
```

```
        )
    if f_stat_b > f_crit_b:
        print("Reject null hypothesis for CAEC: At least one group mean is
different.")
    else:
        print(
            "Fail to reject null hypothesis for CAEC: No significant difference
between group means."
        )

    return anova_table


def multiple_comparisons(model):
    from statsmodels.stats.multicomp import pairwise_tukeyhsd
    import pandas as pd

    # Extract dataframe used in the model
    df = model.model.data.frame.copy()

    # Convert factors to string for Tukey labels
    df["MTRANS_str"] = df["MTRANS"].astype(str)
    df["CAEC_str"] = df["CAEC"].astype(str)

    # --- Tukey for MTRANS ---
    tukey_mtrans = pairwise_tukeyhsd(
        endog=df["Weight"], groups=df["MTRANS_str"], alpha=0.05
    )

    # Convert Tukey result to DataFrame
    mtrans_df = pd.DataFrame(
        data=tukey_mtrans._results_table.data[1:],  # skip header row
        columns=tukey_mtrans._results_table.data[0],  # use header row
    )

    # --- Tukey for CAEC ---
    tukey_caec = pairwise_tukeyhsd(
        endog=df["Weight"], groups=df["CAEC_str"], alpha=0.05
    )

    caec_df = pd.DataFrame(
        data=tukey_caec._results_table.data[1:],
        columns=tukey_caec._results_table.data[0],
    )
    return mtrans_df, caec_df


# -------------------------------
# Summary Statistics (Both Datasets)
# -------------------------------
def quantitative_summary(df, title):
    number_df = df.select_dtypes(include="number")

    # compute summary statistics
    summary = pd.DataFrame(
        {
            # Measures of Central Tendency
            "Mean": number_df.mean(),  # find means
            "Median": number_df.median(),  # find medians
            "Mode": number_df.mode().iloc[0],  # find modes
            # Measures of Variability
            "Range": number_df.max() - number_df.min(),  # find ranges
            "Variance": number_df.var(),  # find variances
            "Standard Deviation": number_df.std(),  # find standard deviations
            "IQR": number_df.quantile(0.75)
            - number_df.quantile(0.25),  # find interquartile ranges
        }
    )

    # create figure and axes
    figure, axes = plt.subplots(figsize=(14, 6))
    axes.axis("off")  # hide plot axes

    # create summary table
    summary_table = plt.table(
        cellText=summary.round(4).values,  # round 4 decimal places
        rowLabels=summary.index,  # label rows
        colLabels=summary.columns,  # label columns
        loc="center",  # center table
        cellLoc="center",  # center text
    )

    # size table
    summary_table.scale(1, 2)

    # add title
    plt.title(title)
    plt.show()


def category_summary(df, title, wine_flag=False):
    # For wine quality dataset
    if wine_flag:
        category_columns = df.columns

    # For obesity dataset
    else:
        category_columns = df.select_dtypes(exclude="number").columns

    # categorical variable summary
    for col in category_columns:
        categorical_summary = pd.DataFrame(
            {
                "Quality Score": df[col]
                .value_counts()
                .sort_index()
                .index,  # find quality scores
                "Frequency": df[col].value_counts().sort_index(),  # find
frequencies
                "Proportion": df[col]
                .value_counts(normalize=True)
                .sort_index(),  # find proportions
            }
        )

        # create figure and axes
        figure, axes = plt.subplots(figsize=(14, 6))
        axes.axis("off")  # hide plot axes

        # create summary table
        summary_table = plt.table(
            cellText=categorical_summary.round(4).values,  # round 4 decimal
places
            colLabels=categorical_summary.columns,  # label columns
            loc="center",  # center table
            cellLoc="center",  # center text
        )

        # size table
        summary_table.scale(1, 3)

        # add title
        plt.title(f"{title}: {col}")
        plt.show()


# --------------------------------
# One Sample Test (Wine Quality Dataset)
# --------------------------------
def alcohol_t_test():
    # get variable of interest
    alcohol_content = X["alcohol"].values

    # Hypothesized population mean
    pop_mean = 10.5

    # Sample statistics
    n = len(alcohol_content)
    df = n - 1
    sample_mean = np.mean(alcohol_content)
    sample_std = np.std(alcohol_content, ddof=1)

    # Compute t-statistic manually
    t_statistic = (sample_mean - pop_mean) / (sample_std / np.sqrt(n))

    # Find t-critical value for one-tailed test
    alpha = 0.05
    t_crit = stats.t.ppf(1 - alpha, df)

    # Print results
    print(f"Sample mean: {sample_mean:.4f}")
    print(f"T-statistic: {t_statistic:.4f}")
    print(f"T-critical (one-tailed, alpha={alpha}): {t_crit:.4f}")

    # Decision based on t-critical
    if t_statistic > t_crit:
        print("Reject the null hypothesis based on t-critical value.")
    else:
        print("Fail to reject the null hypothesis based on t-critical value.")

    # Also show p-value for reference
    t_stat, p_value_2tail = stats.ttest_1samp(a=alcohol_content,
popmean=pop_mean)

    # find one-tailed p-value
    if t_statistic > 0:
        p_value_1tail = p_value_2tail / 2
    else:
        p_value_1tail = 1 - (p_value_2tail / 2)

    print(f"\nOne-Tailed P-value: {p_value_1tail:.4f}")

    if p_value_1tail < alpha:
        print("Reject the null hypothesis based on p-value.")
    else:
        print("Fail to reject the null hypothesis based on p-value.")

    # find 95% confidence interval
    margin_error = (stats.t.ppf(1 - 0.05 / 2, df)) * (sample_std / np.sqrt(n))
    low_interval = sample_mean - margin_error
    high_interval = sample_mean + margin_error

    print()
    print(f"95% Confidence Interval:\n({low_interval:.4f},
{high_interval:.4f})")


# --------------------------------
# ANOVA (Wine Quality Dataset)
# --------------------------------
def anova_analysis():
    # combine X and Y dataframes
    combined_data = X.join(y)

    # Run ANOVA test
    anova_model = smf.ols("alcohol ~ C(quality)", data=combined_data).fit()
    anova_table = sm.stats.anova_lm(anova_model, typ=2)

    # print table
    print(anova_table)

    # create figure and axes
    figure, axes = plt.subplots(figsize=(14, 6))
    axes.axis("off")  # hide plot axes
    # create summary table
    anova_table_output = plt.table(
        cellText=anova_table.round(4).values,  # round 4 decimal places
        rowLabels=["Quality", "Residual Error"],  # label rows
        colLabels=[
            "Sum of Squares",
            "Degrees of Freedom",
            "F-Statistic",
            "P-Value",
        ],  # label columns
        loc="center",  # center table
        cellLoc="center",  # center text
    )

    # size table
    anova_table_output.scale(1, 2)

    # add title
    plt.title("ANOVA Table: Alcohol Content by Wine Quality")
    plt.show()

    # find rejection region
    alpha = 0.05
    df1 = 6
    df2 = 6490

    f_critical_value = stats.f.ppf(1 - alpha, df1, df2)
    print(f"F-Critical Value: {f_critical_value}")

    print(f"\nReject H0 if F > {f_critical_value:.4f}")

    # residual check with QQ-plot
    sm.qqplot(anova_model.resid, line="45")
    plt.title("QQ-Plot of ANOVA Residuals")
    plt.show()

    # equal variance check with boxplots
    sns.boxplot(x="quality", y="alcohol", data=combined_data)
    plt.title("Alcohol Content Through Alcohol Quality Levels")
    plt.show()


# --------------------------------
# Main Execution (Both Datasets)
# -------------------------------
if __name__ == "__main__":

    # Sampling Distributions
    plot_histogram_q1()
    plot_histogram_q2()

    # Bootstrap
    runs = bootstrap()
    plot_bootstrap(runs)
    bootstrap_qq(runs)

    # Summary Statistics
    # Wine Quality
    quantitative_summary(X, "Wine Quality - Quantitative Summary")
    category_summary(y, "Wine Quality - Quality Score", wine_flag=True)

    # Obesity Dataset
    quantitative_summary(X2, "Obesity Dataset - Quantitative Summary")
    category_summary(X2, "Obesity Dataset - Categorical Variable")
    category_summary(y2, "Obesity Dataset - Target Variable")

    # Hypothesis Tests
    alcohol_t_test()

    # ANOVA and Interaction Model
    model = mult_comparisons()
    plot_resid_fitted(model)
    hypothesis_test(model)
    multiple_comparisons(model)
    pass
```