Wine and Obesity Data Analysis

Adrian T. and Olivia K.

2025-12-09

Table of contents

1. Dataset Overview

1.1 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

1.1.1 Summary Statistics

Table 2: Summary Statistics for Quantitative Variables in the Wine Quality Dataset

Variable	Mean	Median	Mode	Range	Variance	Std. Dev.	IQR
Fixed Acidity	7.2153	7.00	6.8	12.1	1.6807	1.2964	1.3
Volatile Acidity	0.3397	0.29	0.28	1.50	0.0271	0.1646	0.17
Citric Acid	0.3186	0.31	0.30	1.66	0.0211	0.1453	0.14
Residual Sugar	5.4432	3.00	2.0	65.2	22.6367	4.7578	6.3
Chlorides	0.0560	0.047	0.044	0.602	0.0012	0.0350	0.027
Free Sulfur Dioxide	30.5253	29.00	29.0	288.0	315.0412	17.7494	24.0
Total Sulfur Dioxide	115.7446	118.00	111.0	434.0	3194.720	56.5219	79.0
Density	0.9947	0.9949	0.9972	0.0519	0.0000	0.0030	0.0047
pH	3.2185	3.21	3.16	1.29	0.0259	0.1608	0.21
Sulphates	0.5313	0.51	0.50	1.78	0.0221	0.1488	0.17
Alcohol	10.4918	10.30	9.5	6.90	1.4226	1.1927	1.8

Table 2 summarizes the central tendency and variability of the quantitative variables int eh Wine Quality dataset. Alcohol content has a mean of approximately 10.49% and a standard deviation of 1.19. THis indicates a moderate variation across wine samples. Several of the other properties such as pH values and density, show little to no variability. In contrast, residual sugar and sulfur dioxide concentrations show larger ranges and higher variability. This reflects differences in wine styles and fermentation processes. Overall, these summary statistics show that although some chemical properties stay stable across wines, others largely vary and may contribute to differences like perceived quality.

Table 3: Distribution of Wine Quality Scores

Quality Score	Frequency	Proportion
3	30	0.0046
4	216	0.0332
5	2138	0.3291
6	2836	0.4365
7	1079	0.1661
8	193	0.0297
9	5	0.0008

The distribution of wine quality scores is shown in Table 3. A majority of the wines received a score of 5 or 6. Scores of 3-4 and 8-9 are extremely rare. The concentration around mid-range scores suggests that the dataset consists of average-quality wines, with fewer extreme cases.

1.1.2 Sampling Distribution of Density

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

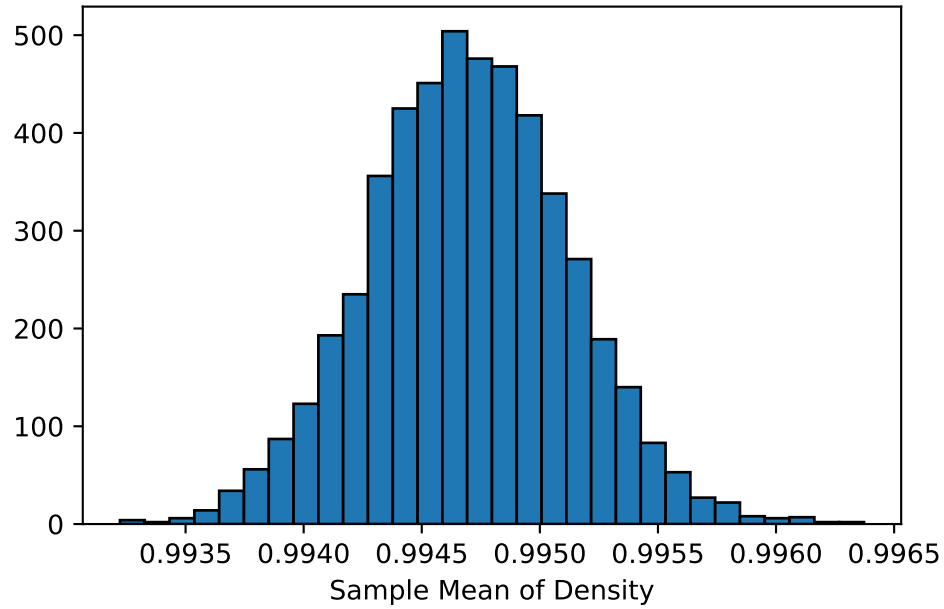

Figure 1: Sampling Distribution of Sample Mean (Density)

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.

1.2 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 4: 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	-

1.2.1 Summary Statistics

Table 5: Summary Statistics for Quantitative Variables in the Obesity Dataset

Variable	Mean	Median	Mode	Range	Variance	Std. Dev.	IQR
Age (years)	24.3126	22.7779	18.0	47.0	40.2713	6.346	6.0528
Height (m)	1.7017	1.7005	1.7	0.53	0.0087	0.0933	0.1385
Weight (kg)	86.5861	83.0	80.0	134.0	685.9775	26.1912	41.9573
FCVC	2.419	2.3855	3.0	2.0	0.2851	0.5339	1.0
NCP	2.6856	3.0	3.0	3.0	0.6053	0.778	0.3413
CH2O	2.008	2.0	2.0	2.0	0.3757	0.613	0.8926
FAF	1.0103	1.0	0.0	3.0	0.7235	0.8506	1.5422
TUE	0.6579	0.6254	0.0	2.0	0.3708	0.6089	1.0

Table 5 shows the summary stastics for quantitative variables in the Obesity database. The sample consited of primarily young adults. The average participant was aged at 24 years, with a moderate spread. Weight displays substantial variability and has a standard deviation exceeding 26 kg. In contrast, height shows much less variability.

Table 6: Gender Distribution in the Obesity Dataset

Gender	Frequency	Proportion
Female	1043	0.4941
Male	1068	0.5059

Table 7: Family History of Overweight

Family History	Frequency	Proportion
No	385	0.1824
Yes	1726	0.8176

Table 8: Frequent High-Calorie Food Consumption (FAVC)

FAVC	Frequency	Proportion
No	245	0.1161
Yes	1866	0.8839

Table 9: Snacking Between Meals (CAEC)

Snacking Frequency	Frequency	Proportion
Always	53	0.0251
Frequently	242	0.1146
Sometimes	1765	0.8361
No	51	0.0242

Table 10: Smoking Status

Smoking	Frequency	Proportion
No	2067	0.9792
Yes	44	0.0208

Table 11: Calorie Monitoring (SCC)

Calorie Monitoring	Frequency	Proportion
No	2015	0.9545
Yes	96	0.0455

Table 12: Alcohol Consumption Frequency (CALC)

Alcohol Consumption	Frequency	Proportion
Always	1	0.0005
Frequently	70	0.0332
Sometimes	1401	0.6637
No	639	0.3027

Table 13: Mode of Transportation (MTRANS)

Transportation Mode	Frequency	Proportion
Automobile	457	0.2165
Bike	7	0.0033
Motorbike	11	0.0052
Public Transportation	1580	0.7485
Walking	56	0.0265

Table 14: Obesity Classification (NObeyesdad)

Obesity Category	Frequency	Proportion
Insufficient Weight	272	0.1288
Normal Weight	287	0.1360
Obesity Type I	351	0.1663
Obesity Type II	297	0.1407
Obesity Type III	324	0.1535
Overweight Level I	290	0.1374
Overweight Level II	290	0.1374

Table 6 through Table 14 summarize the distributions of categorical variables in the Obesity dataset. The sample is approximately evenly split by gender. A large majority of participants report a family history of overweight and frequent consumption of high-calorie foods, which shows that genetic and dietary risk factors are common in this population. Snacking between meals is also prevalent, with most individuals reporting that they snack at least sometimes. Smoking and calorie monitoring behaviors are relatively uncommon, suggesting that these factors are unlikely to play a major role in weight variation within this sample. Public transportation is the dominant mode of transportation, while active transportation methods such as walking and biking are rare. Finally, the obesity classification variable is fairly evenly distributed across normal weight, overweight, and obesity categories, which makes the dataset well suited for comparision analyses across weight groups.

1.2.2 Sampling Distribution of Weight

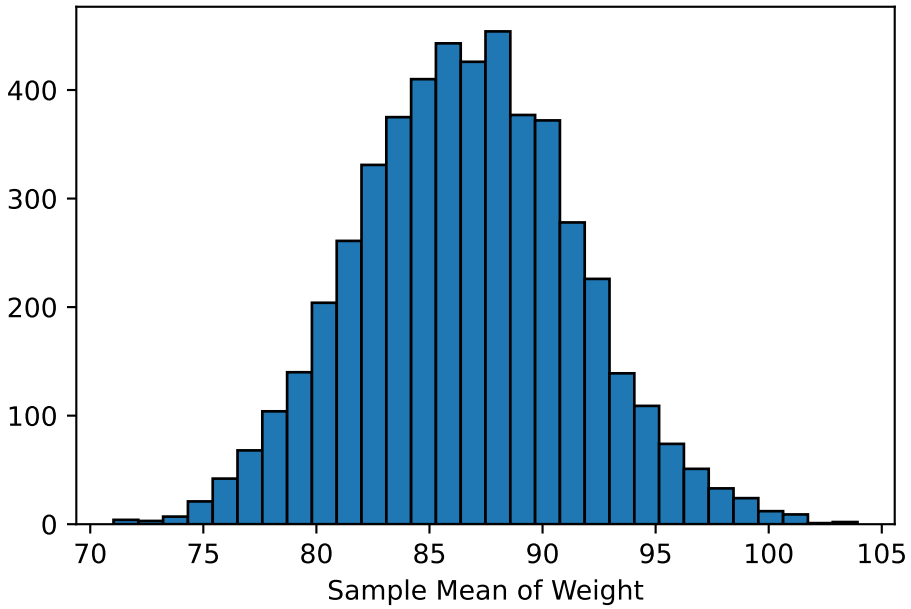

Figure 2: Sampling Distribution of Sample Mean (Weight)

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. The right tail has some samples as high as 104 kg, and the left tail has some samples as low as 70 kg. There is no noticeable skewness and heavy tails.

---

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

The validity of the one-sample t-test relies on several key assumptions:

1. **Independence:**
   Each observation represents a separate wine sample, and there is no indication that the measurements are related. Independence is satisfied since the wines were independently sampled.

2. **Normality:**
   Although the distribution of alcohol content is not perfectly normal, the sample size is very large, n = 6497. Thus, by the Central Limit Theorem, the sampling distribution of the sample mean is approximately normal, which satisfies the normality requirement.

Thus, the assumptions necessary for the one-sample t-test are reasonably satisfied.

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 α , 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

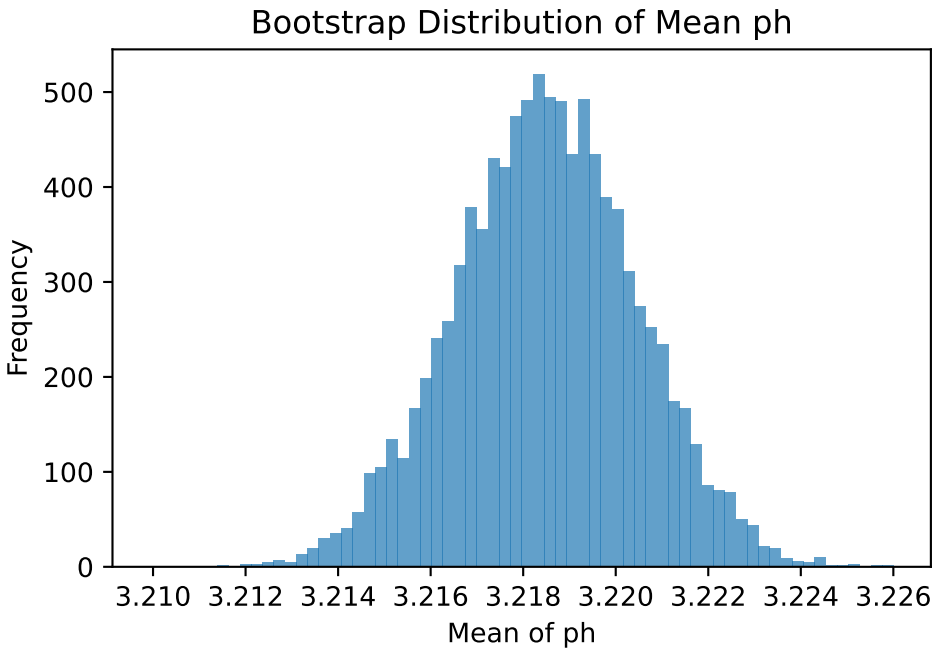

Figure 3: Bootstrap Distribution of Wine pH

Figure 3 is the Bootstrap Distribution of Wine pH. The distribution appears approximately normal centered around a mean pH of 3.22. The frequency near the center are roughly 500.

3.2 QQ Plot for Bootstrap

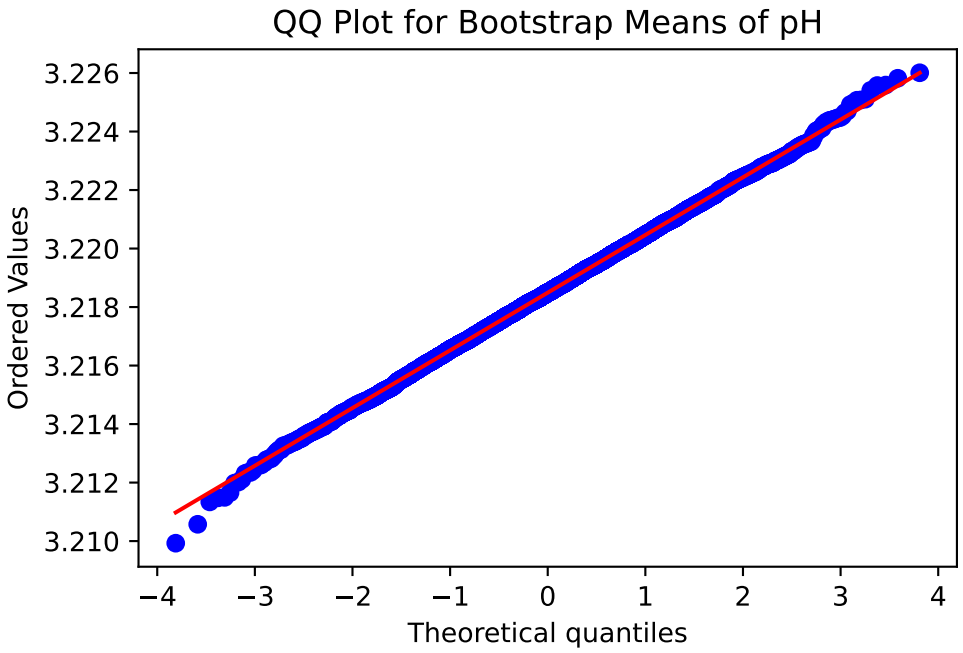

Figure 4: QQ Plot of Bootstrap Means (pH)

Figure 4 is the QQ plot of the Bootstrap Means. The data points lie entirely on the normal line, indicating that the distribution is approximatly normal.

3.3 Bootstrap Confidence Interval

```
The bootstrap test CI was calculated to be (3.2146358704017235, 3.222370363244574)
```

Conclusion: Using an α 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

The goal of this analysis is to determine whether the mean alcohol content differs across wine quality levels.

The null hypothesis states that the mean alcohol content is equal across all quality categories:

$$H_0 : \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = \mu_8 = \mu_9$$

The alternative hypothesis states that at least one quality level has a different mean alcohol content:

$$H_1 : \text{At least one mean differs}$$

The one-way ANOVA F-test yielded a test statistic of

$$F = 320.59.$$

At a significance level of $\alpha = 0.05$, the rejection region is defined as

$$F > 2.10.$$

Since the observed F-statistic greatly exceeds the critical value and the associated p-value is less than 0.0001, the null hypothesis is rejected.

4.2 ANOVA Table

Table 15: ANOVA Table for Alcohol Content by Wine Quality Level

Source of Variation	Sum of Squares	Degrees of Freedom	F-Statistic	p-value
Wine Quality	2112.73	6	320.59	< 0.0001
Residual Error	7128.23	6490	—	—
Total	**9240.96**	**6496**		

Table Table 15 summarizes the results of the one-way ANOVA for alcohol content across wine quality levels.

4.3 AVOVA Assumptions Check

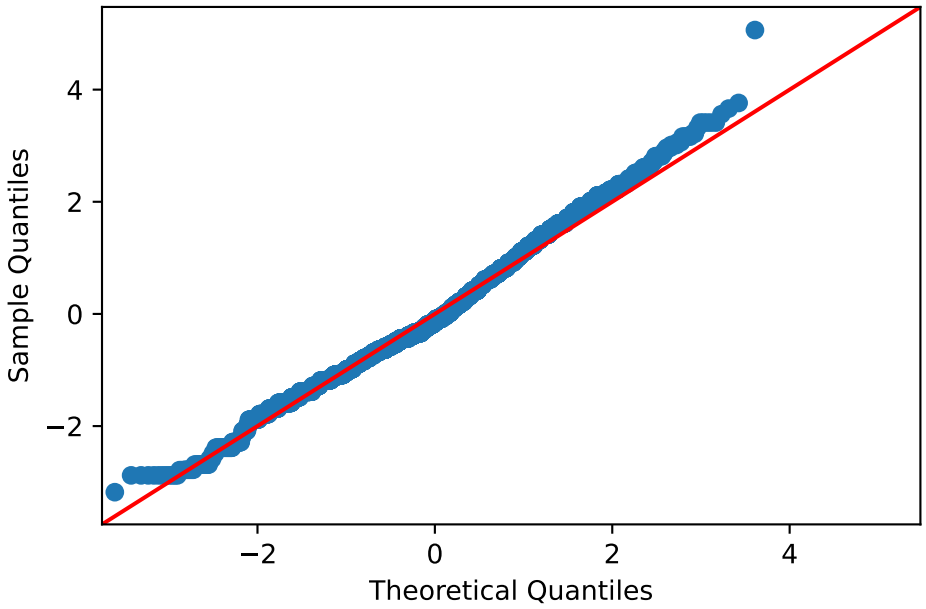

Figure 5: QQ plot of residuals from the one-way ANOVA model for alcohol content

Figure 5 assesses the normality assumption for the ANOVA by displaying the QQ plot of residuals. Since the residuals lie approximately along the reference line with only minor deviations, the normality assumption for the ANOVA residuals is reasonably satisfied.

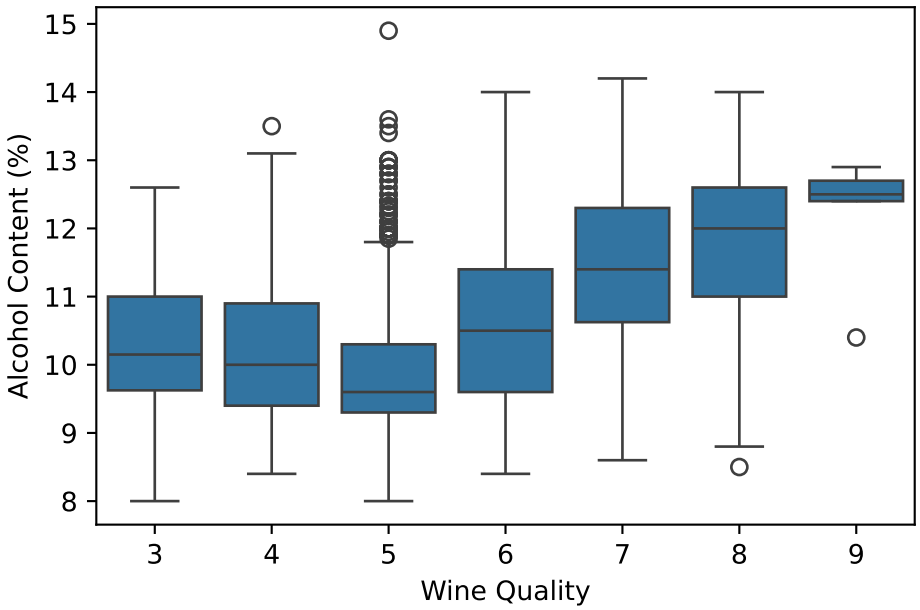

Figure 6: Boxplots of alcohol content across wine quality levels

Figure 6 checks that all groups have a roughly equal variance by comparing the spread of alcohol content across wine quality groups. The boxplots show similar spread and variability, which shows that the equal variance assumption is not violated.

Overall, the figures indicate that the assumptions required for the one-way ANOVA are satisfied.

4.4 Conclusion

Based on the results of the one-way ANOVA, the null hypothesis of equal mean alcohol content across wine quality levels in rejected at 5% significance level. This supplies strong evidence that alcohol content differs among at at least some wine quality categories. Since ANOVA doesn't identify which specific quality levels differ, the next step is to conduct comparision procedures like Tukey's HSD test, to determine where the differences occur.

---

5. Multiple Comparisons (Obesity Dataset)

5.1 ANOVA Model and Assumptions

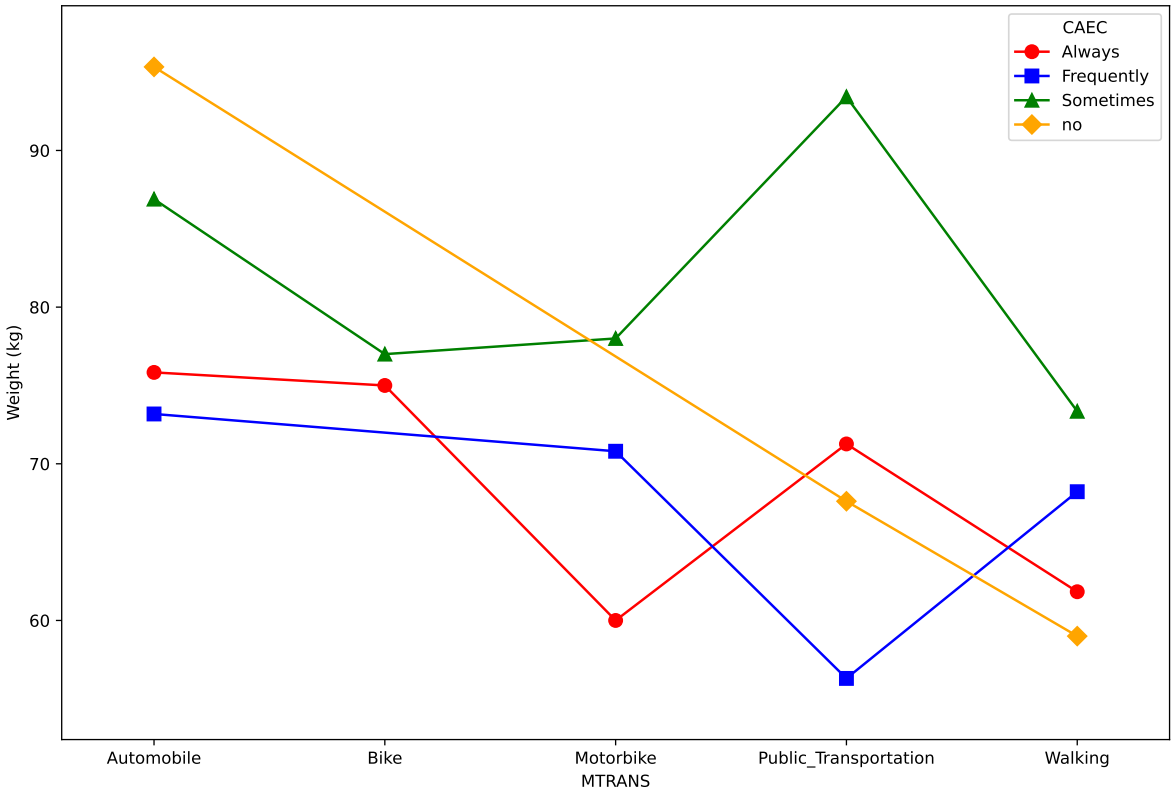

Figure 7: Two-way ANOVA with interaction: Weight ~ MTRANS * CAEC

- **Interpretation:** Based on Figure 7, 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:

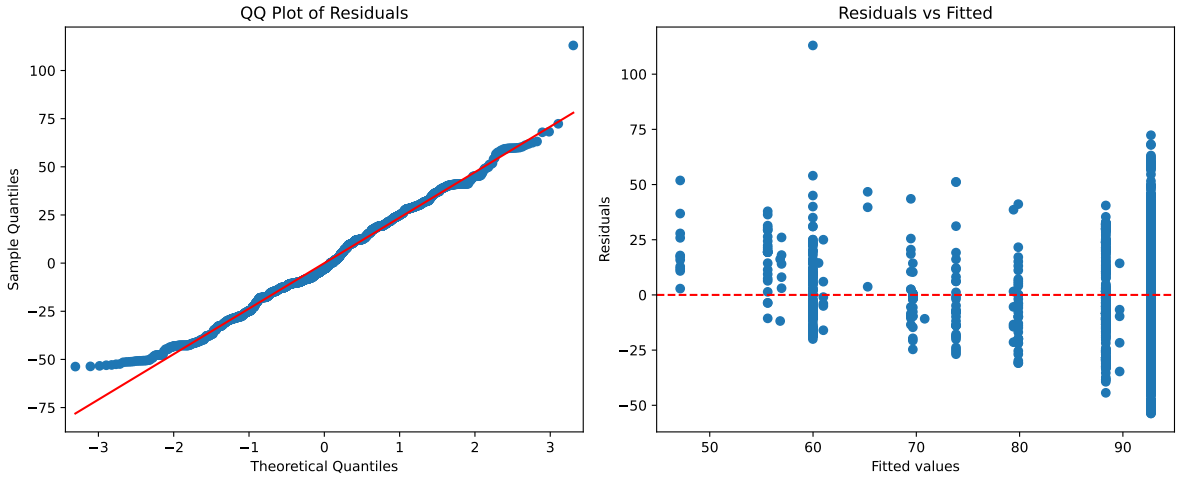

Figure 8: QQ Plot and Residual Plot

- The left plot in Figure 8 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 8 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 16: ANOVA results for MTRANS and CAEC factors

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

Interpretation: Table 16 shows the ANOVA results of the two way effect of MTRANS and CAEC on mean weight. In each factor, the F-statistic is greater than the critical value, indicating that at least one group within each factor has a mean weight that is significantly different from another group.

5.4 Pairwise Comparisons

Table 17: 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 17, 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 18: 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

Table 18: Tukey HSD Pairwise Comparison for CAEC

	CAEC A	CAEC B	meandiff	p-adj	lower	upper	reject
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: From Table 18 shows the results of Tukey's HSD test on pairwise differences across the different levels of CAEC in mean Weight. Almost all pairs presented significant differences.

The largest difference was between individuals who snacked between meals 'frequently' and 'sometimes', with the mean difference being 32.47. Additionally, individuals who snack 'never'differ greatly from ones who snacked 'frequently' and 'sometimes'.

Overall, these results suggest that the frequency of snacking between meals is strongly associated with differences in mean weight, with the exception of the 'always' vs. 'no' comparison, which did not produce a significant difference.

---

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.
data.features
y2 = estimation_of_obesity_levels_based_on_eating_habits_and_physical_condition.
data.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():
    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
    )
    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)

    # 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

    df = model.model.data.frame.copy()

    df['MTRANS_str'] = df['MTRANS'].astype(str).str.replace('_', ' ')
    df['CAEC_str'] = df['CAEC'].astype(str).str.replace('_', ' ')

    # --- 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
    )

    mtrans_df = mtrans_df.rename(columns={'group1': 'Transportion A', 'group2':
'Transportion B'})


    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]
    )
    caec_df = caec_df.rename(columns={'group1': 'CAEC A', 'group2': 'CAEC B'})

    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():
    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()
```