Project: Advanced Statistics

Problem 1

A physiotherapist with a male football team is interested in studying the relationship between foot injuries and the positions at which the players play from the data collected.

	Striker	Forward	Attacking Midfielder	Winger	**Total**
Players Injured	45	56	24	20	**145**
Players Not Injured	32	38	11	9	**90**
Total	**77**	**94**	**35**	**29**	**235**

Based on the above data, answer the following questions:

1.1 What is the probability that a randomly chosen player would suffer an injury?

In [410…

```
Total_Players_Injured = 145
Total = 235

P_Suffered_Injury = Total_Players_Injured / Total

print(
    "The probability that a randomly chosen player would suffer an injury is:",
    round(P_Suffered_Injury, 4)
)
```

```
The probability that a randomly chosen player would suffer an injury is: 0.617
```

1.2 What is the probability that a player is a forward or a winger?

In [411…

```
P_Forward = 94 / 235
P_Winger = 29 / 235

P_Forward_Winger = P_Forward + P_Winger

print(
    "The probability that a player is a forward or a winger is:",
    round(P_Forward_Winger, 4)
)
```

```
The probability that a player is a forward or a winger is: 0.5234
```

1.3 What is the probability that a randomly chosen player plays in a striker position and has a foot injury?

In [412...

```
P_Striker_Foot_Injury = 45 / 235

print(
    "The probability that a randomly chosen player plays in a striker position and
    round(P_Striker_Foot_Injury, 4)
)
```

```
The probability that a randomly chosen player plays in a striker position and has a
foot injury is: 0.1915
```

1.4 What is the probability that a randomly chosen injured player is a striker?

In [413...

```
P_Injured_Striker = 45 / 145

print(
    "The probability that a randomly chosen injured player is a striker is:",
    round(P_Injured_Striker, 4)
)
```

```
The probability that a randomly chosen injured player is a striker is: 0.3103
```

Problem 2

The breaking strength of gunny bags used for packaging cement is normally distributed with a mean of 5 kg per sq. centimeter and a standard deviation of 1.5 kg per sq. centimeter. The quality team of the cement company wants to know the following about the packaging material to better understand wastage or pilferage within the supply chain.

Answer the questions below based on the given information:

To answer the given questions, we will use the normal distribution formulas:

cdf (cumulative density function) = cdf(x, mu(μ), sigma(σ))
pdf (probability density function) = pdf(x, mu(μ), sigma(σ))

Where,

x = observed value
mu(μ) = mean
sigma(σ) = standard deviation

In [414...

```
# Libraries to help with reading and manipulating data
import numpy as np
import pandas as pd

# Libraries to help with data visualization
import matplotlib.pyplot as plt
```

```
import seaborn as sns
%matplotlib inline

# Library to help with statistical analysis
import scipy.stats as stats
from scipy.stats import norm
```

2.1 What proportion of the gunny bags have a breaking strength of less than 3.17 kg per sq cm?

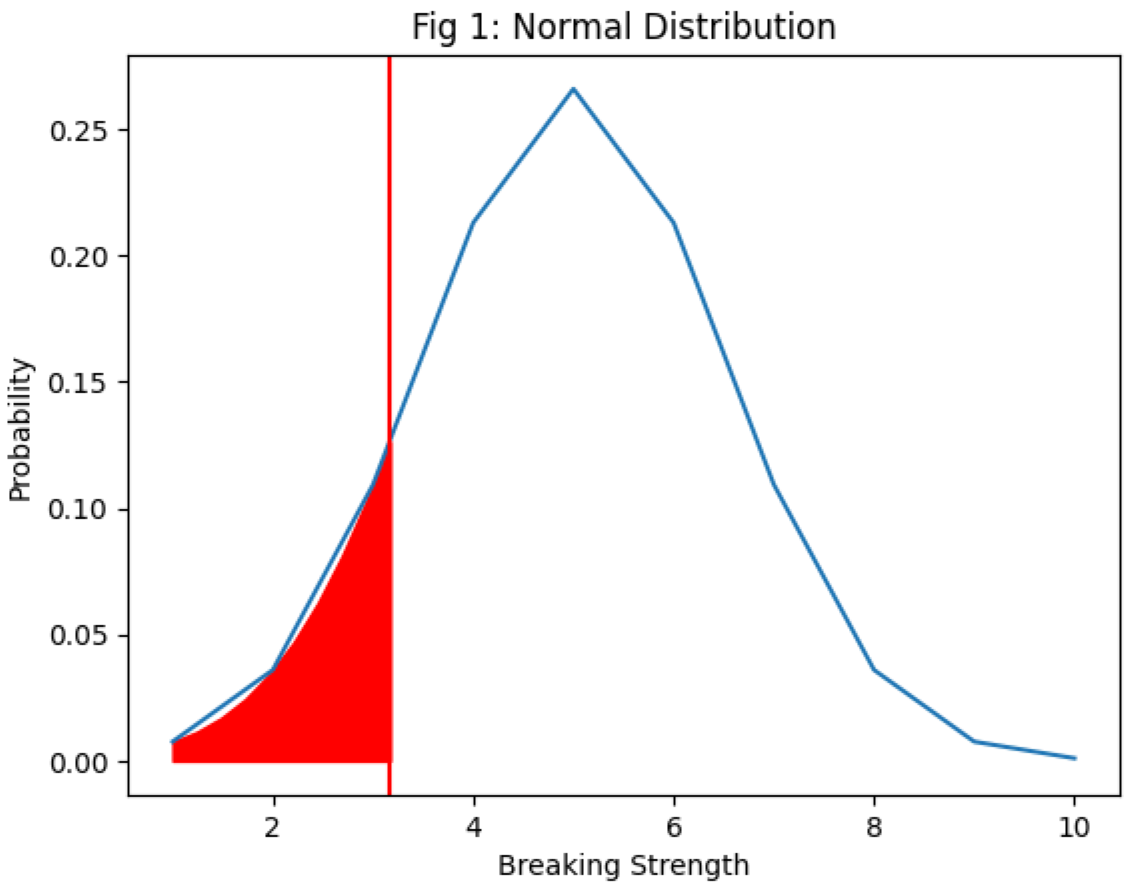
In [415…

```
mu = 5
sigma = 1.5

# find the cumulative probability

prob_less_than_3_17 = norm.cdf(3.17, mu, sigma)

print(
    "The probability that proportion of the gunny bags have a breaking strength of
    round(prob_less_than_3_17, 4)
)
```

```
The probability that proportion of the gunny bags have a breaking strength of less t
han 3.17 kg per sq cm is: 0.1112
```

In [416…

```
# plot the pdf of breaking strength using norm.pdf()

density = pd.DataFrame() # create an empty DataFrame

density["x"] = np.linspace(
    1, 10, 10
) # create an array of 10 numbers in between the 1 and 10 breaking strength range a

density["pdf"] = norm.pdf(density["x"], 5, 1.5) # calculate the pdf() of the create

plt.plot(density["x"], density["pdf"]) # plot the pdf of the normal distribution

plt.axvline(x=3.17, c="r") # draw a red vertical line at x = 3.17
x1 = np.linspace(density["x"].min(), 3.17, 10) # create an array of 10 numbers betw
plt.fill_between(x1, norm.pdf(x1, 5, 1.5), color="r") # fill the specified region w
plt.xlabel("Breaking Strength") # set the x-axis label
plt.ylabel("Probability") # set the y-axis label
plt.title("Fig 1: Normal Distribution") # set the title
plt.show()
```



2.2 What proportion of the gunny bags have a breaking strength of at least 3.6 kg per sq cm.?

In [417…

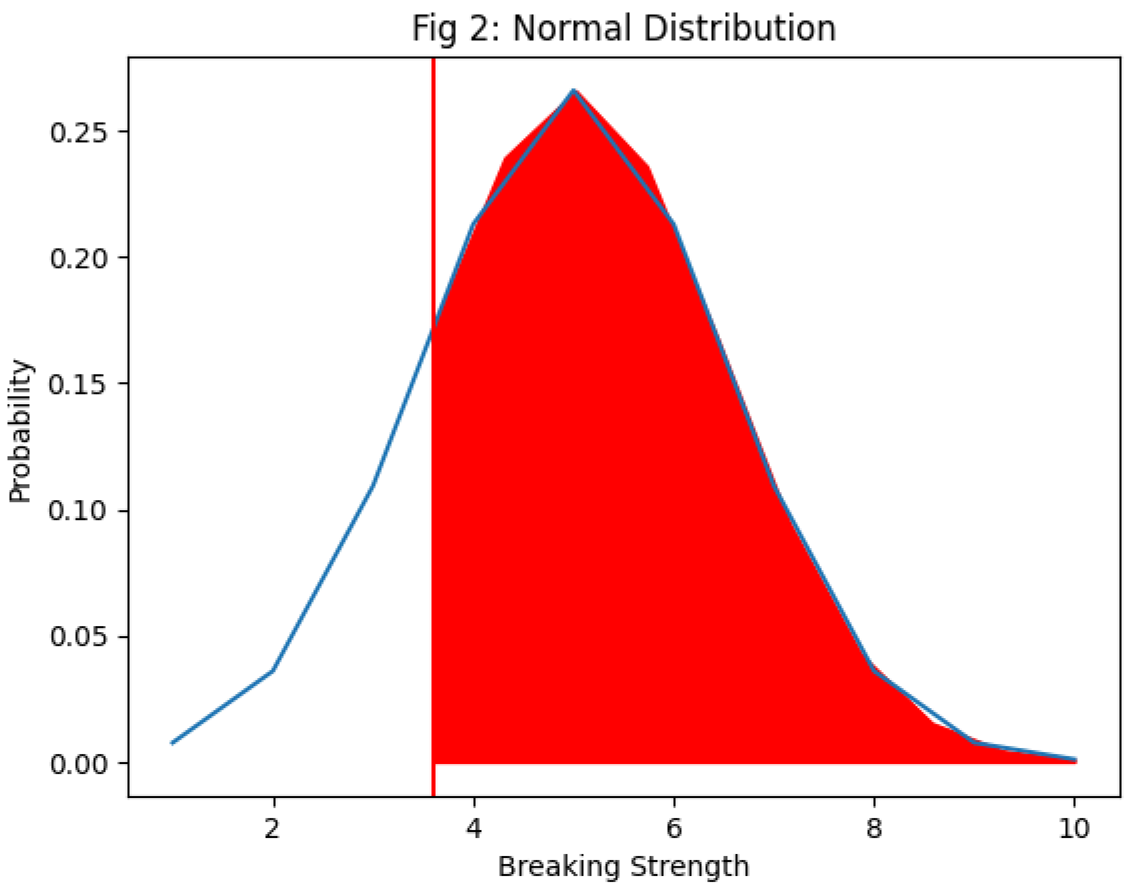
```
mu = 5
sigma = 1.5

# find the cumulative probability

prob_more_than_3_6 = 1 - norm.cdf(3.6, mu, sigma)
print(
    "The probability that proportion of the gunny bags have a breaking strength of
    round(prob_more_than_3_6, 4)
)
```

```
The probability that proportion of the gunny bags have a breaking strength of at lea
st 3.6 kg per sq cm is: 0.8247
```

In [418…

```
# plot the pdf of breaking strength using norm.pdf()

density = pd.DataFrame() # create an empty DataFrame

density["x"] = np.linspace(
    1, 10, 10
) # create an array of 10 numbers in between the 1 and 10 breaking strength range a

density["pdf"] = norm.pdf(density["x"], 5, 1.5) # calculate the pdf() of the create

plt.plot(density["x"], density["pdf"]) # plot the pdf of the normal distribution
```

```
plt.axvline(x=3.6, c="r") # draw a red vertical line at x = 3.6
x1 = np.linspace(3.6, density["x"].max(), 10) # create an array of 10 numbers betwe
plt.fill_between(x1, norm.pdf(x1, 5, 1.5), color="r") # fill the specified region w
plt.xlabel("Breaking Strength") # set the x-axis label
plt.ylabel("Probability") # set the y-axis label
plt.title("Fig 2: Normal Distribution") # set the title
plt.show()
```



2.3 What proportion of the gunny bags have a breaking strength between 5 and 5.5 kg per sq cm.?

In [419…

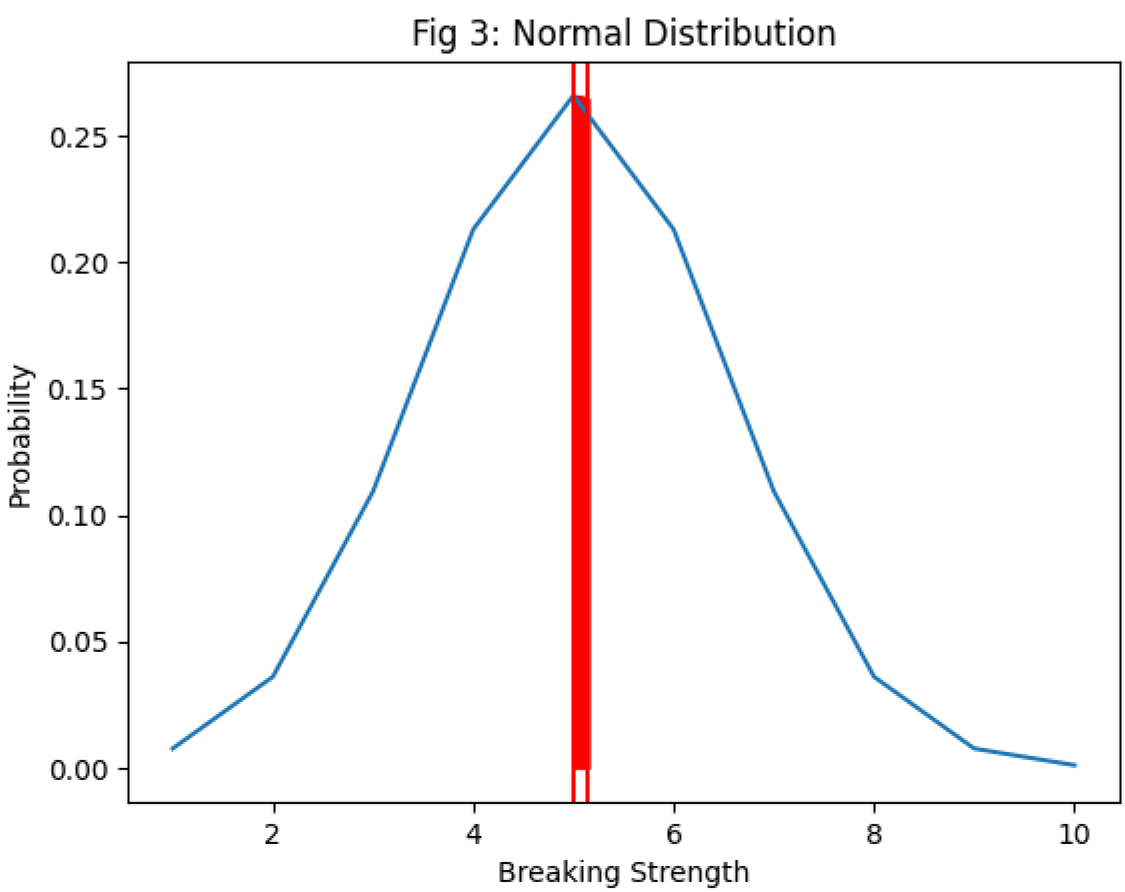
```
mu = 5
sigma = 1.5

# find the cumulative probability

prob_bet_5_a_5_5 = stats.norm.cdf(5.5,mu,sigma) - stats.norm.cdf(5,mu,sigma)
print(
    "The probability that proportion of the gunny bags have a breaking strength bet
    round(prob_bet_5_a_5_5, 4)
)
```

```
The probability that proportion of the gunny bags have a breaking strength between 5
and 5.5 kg per sq cm is: 0.1306
```

In [420…

```
# plot the pdf of breaking strength using norm.pdf()
```

```
density = pd.DataFrame() # create an empty DataFrame

density["x"] = np.linspace(
    1, 10, 10
) # create an array of 10 numbers in between the 1 and 10 breaking strength range a

density["pdf"] = norm.pdf(density["x"], 5, 1.5) # calculate the pdf() of the create

plt.plot(density["x"], density["pdf"]) # plot the pdf of the normal distribution

plt.axvline(x=5, c="r") # draw a red vertical line at x = 5
plt.axvline(x=5.15, c="r") # draw a red vertical line at x = 5.15

x1 = np.linspace(5, 5.15, 10) # create an array of 10 numbers between 5 and 5.15
plt.fill_between(x1, norm.pdf(x1, 5, 1.5), color="r") # fill the specified region w
plt.xlabel("Breaking Strength") # set the x-axis label
plt.ylabel("Probability") # set the y-axis label
plt.title("Fig 3: Normal Distribution") # set the title
plt.show()
```



2.4 What proportion of the gunny bags have a breaking strength NOT between 3 and 7.5 kg per sq cm.?

In [421…

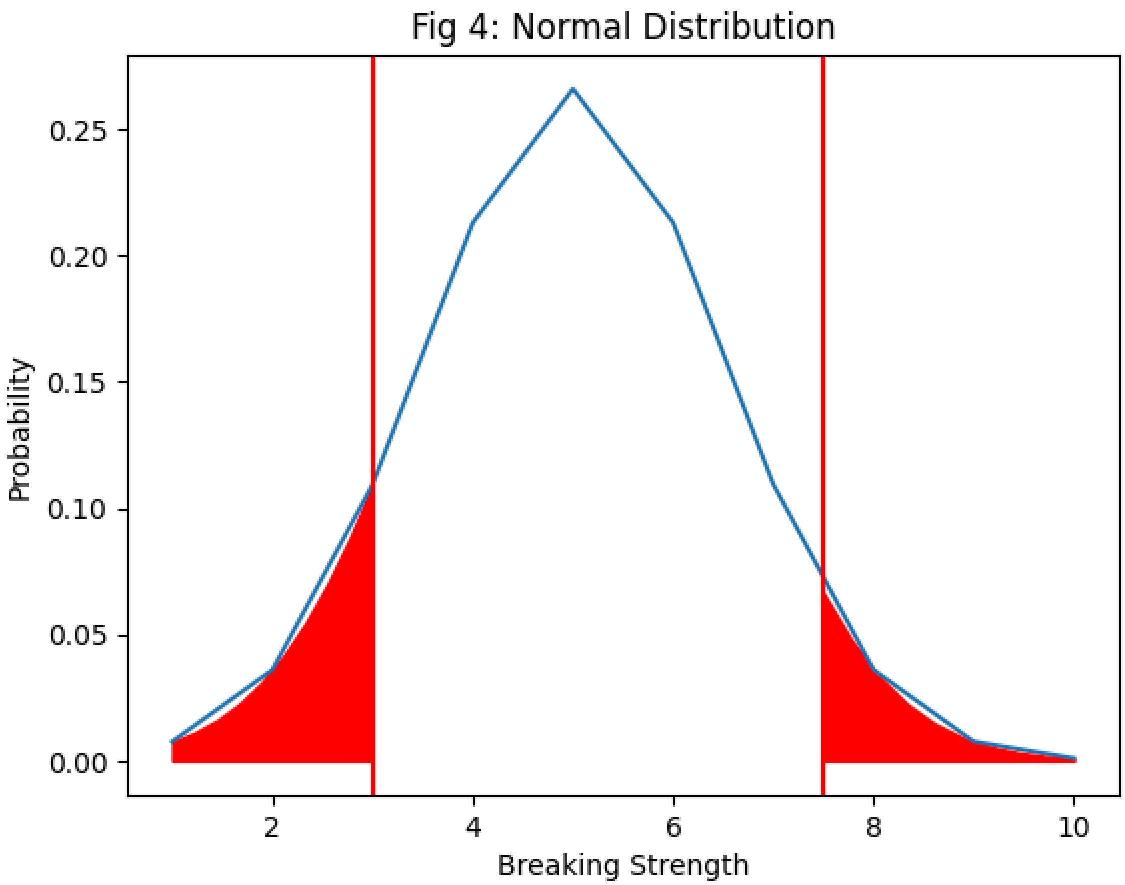
```
mu = 5
sigma = 1.5

# find the cumulative probability
```

```
prob_not_bet_3_a_7_5 = 1 - (stats.norm.cdf(7.5,mu,sigma) - stats.norm.cdf(3,mu,sigm
print(
    "The probability that proportion of the gunny bags have a breaking strength NOT
    round(prob_not_bet_3_a_7_5, 4)
)
```

```
The probability that proportion of the gunny bags have a breaking strength NOT betwe
en 3 and 7.5 kg per sq cm is: 0.139
```

In [422…

```
# plot the pdf of breaking strength using norm.pdf()

density = pd.DataFrame() # create an empty DataFrame

density["x"] = np.linspace(
    1, 10, 10
) # create an array of 10 numbers in between the 1 and 10 breaking strength range a

density["pdf"] = norm.pdf(density["x"], 5, 1.5) # calculate the pdf() of the create

plt.plot(density["x"], density["pdf"]) # plot the pdf of the normal distribution

plt.axvline(x=3, c="r") # draw a red vertical line at x = 3
plt.axvline(x=7.5, c="r") # draw a red vertical line at x = 7.5

plt.axvline(x=3, c="r") # draw a red vertical line at x = 3.17
x1 = np.linspace(density["x"].min(), 3, 10) # create an array of 10 numbers between
plt.fill_between(x1, norm.pdf(x1, 5, 1.5), color="r") # fill the specified region w

plt.axvline(x=7.5, c="r") # draw a red vertical line at x = 3.6
x2 = np.linspace(7.5, density["x"].max(), 10) # create an array of 10 numbers betwe
plt.fill_between(x2, norm.pdf(x2, 5, 1.5), color="r") # fill the specified region w

plt.xlabel("Breaking Strength") # set the x-axis label
plt.ylabel("Probability") # set the y-axis label
plt.title("Fig 4: Normal Distribution") # set the title
plt.show()
```



Problem 3

Zingaro stone printing is a company that specializes in printing images or patterns on polished or unpolished stones. However, for the optimum level of printing of the image, the stone surface has to have a Brinell's hardness index of at least 150. Recently, Zingaro has received a batch of polished and unpolished stones from its clients. Use the data provided to answer the following (assuming a 5% significance level):

In [423…

```
from scipy.stats import ttest_ind # Library to help with statistical analysis
```

In [424…

```
# Load dataset

df_z = pd.read_csv('Zingaro_Company.csv')

df_z.head() # returns first 5 rows
```

Out[424...

	Unpolished	Treated and Polished
0	164.481713	133.209393
1	154.307045	138.482771
2	129.861048	159.665201
3	159.096184	145.663528
4	135.256748	136.789227

In [425...

```
df_z.shape # view the shape of the dataset
```

Out[425...

```
(75, 2)
```

In [426...

```
# check the data types of the columns in the dataset

df_z.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75 entries, 0 to 74
Data columns (total 2 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unpolished            75 non-null     float64
 1   Treated and Polished  75 non-null     float64
dtypes: float64(2)
memory usage: 1.3 KB
```

All columns are float.

There are no missing values in the dataset.

In [427...

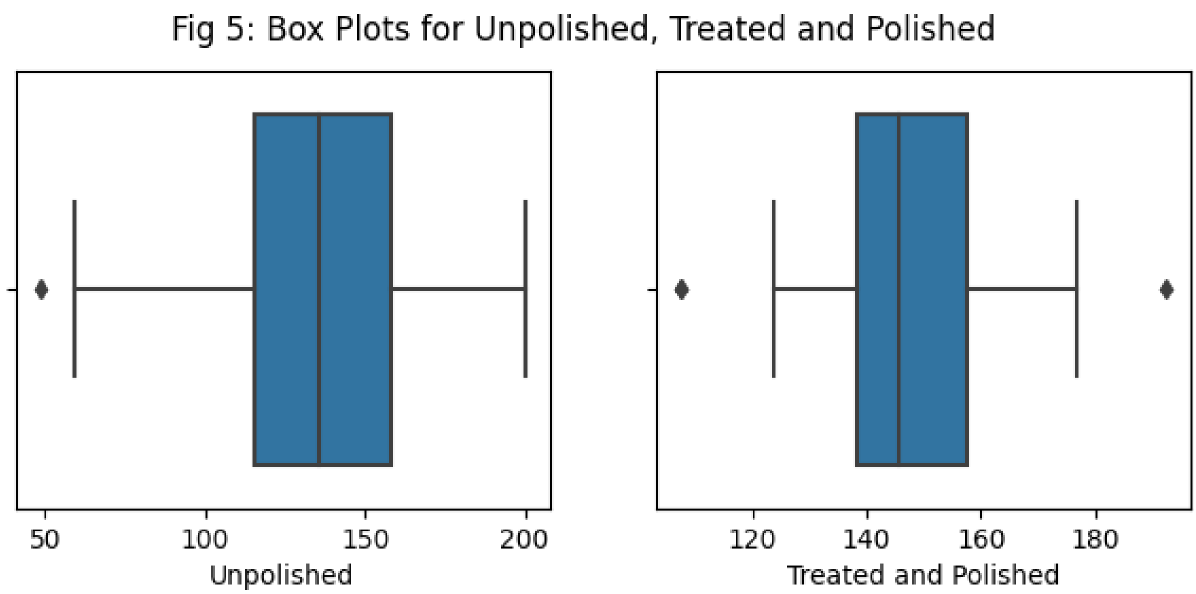
```
df_z.describe().T # statistical summary of the dataset
```

Out[427...

	count	mean	std	min	25%	50%	75%
Unpolished	75.0	134.110527	33.041804	48.406838	115.329753	135.597121	158.215098
Treated and Polished	75.0	147.788117	15.587355	107.524167	138.268300	145.721322	157.373318

In [428...

```
# Box Plots for Unpolished, Treated and Polished

fig, ax = plt.subplots(1,2, figsize=(8,3))

sns.boxplot(data=df_z, x='Unpolished', ax=ax[0])
ax[1]=sns.boxplot(data=df_z, x='Treated and Polished')
ax[0].set(xlabel = 'Unpolished')
ax[1].set(xlabel = 'Treated and Polished')
fig.suptitle('Fig 5: Box Plots for Unpolished, Treated and Polished')
plt.show()
```

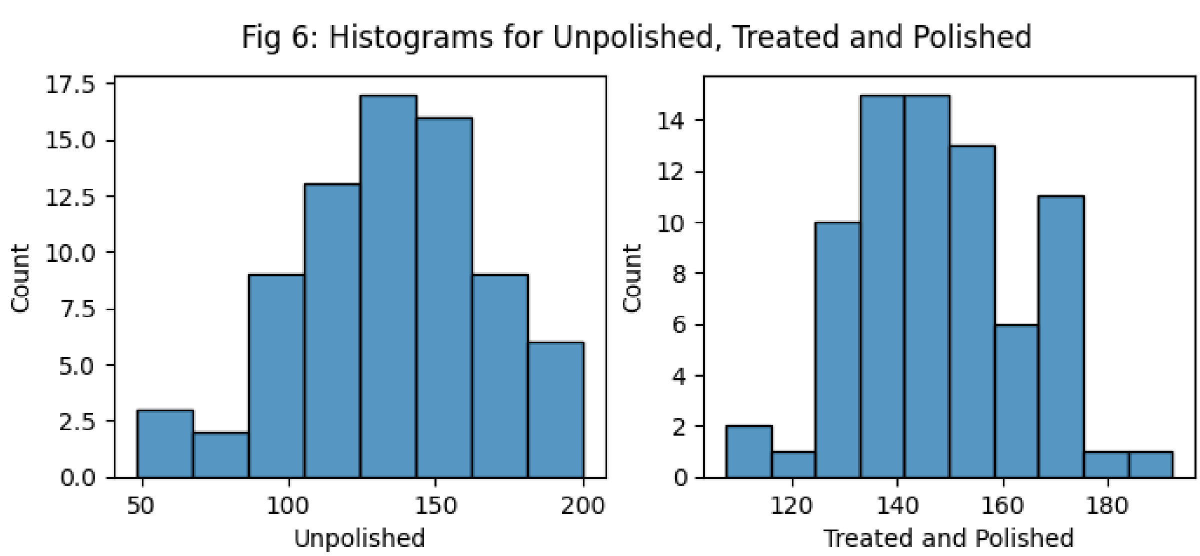


In Fig 5, we can infer:

1. The mean of the two samples is not equal.
2. There are few outliers in both the data so we need to do further statistical test to compares two averages (means) which will give further information if these two means are statistically different from each other or not.

In [429…

```
# Histograms for Unpolished, Treated and Polished

fig, ax = plt.subplots(1,2, figsize=(8,3))

sns.histplot(data=df_z, x='Unpolished', ax=ax[0])
ax[1]=sns.histplot(data=df_z, x='Treated and Polished')
ax[0].set(xlabel = 'Unpolished', ylabel = 'Count')
ax[1].set(xlabel = 'Treated and Polished', ylabel = 'Count')
fig.suptitle('Fig 6: Histograms for Unpolished, Treated and Polished')
plt.show()
```



In Fig 6, Histogram is depicting that data from each of the 2 groups following a normal distribution.

null (H_0) and alternative hypotheses (H_a):

> H_0: μA = μB (the Brinell's harness index of polished and unpolished stones are equal)
> H_a: μA ≠ μB (the Brinell's harness index of polished and unpolished stones are not equal)

Significance level:

$\alpha = 0.05$

Test statistic:

Population standard deviation is not known.
The observations in one sample should be independent of the observations in the other sample.
Data from each of the 2 groups following a normal distribution.

Therefore, we will use T Test to compute p-value.

In [430…

```
# Calculate the p value of test statistic t

t_stat, p_value = ttest_ind(df_z['Unpolished'], df_z['Treated and Polished'])

t_stat, p_value
```

Out[430…

```
(-3.2422320501414053, 0.0014655150194628353)
```

According to the T Score to P Value Calculator, the p-value associated with t = -3.2422320501414053 and degrees of freedom = n1+n2-2 = 75+75-2 = 148 is 0.001465515019462831.

Two-sample t-test p-value= 0.001465515019462831

Conclusion:

p-value < α = 0.05

We reject the null hypothesis in favour of alternative hypothesis.
We conclude that the mean of Brinell's harness index of the polished & unpolished stones are not same.

3.1 Zingaro has reason to believe that the unpolished stones may not be suitable for printing. Do you think Zingaro is justified in thinking so?

Based on the above t – test it is concluded that Brinell's harness index of the polished & unpolished stones are not same. Therefore, Zingaro has enough reason to believe now that the unpolished stones may not be suitable for printing.

3.2 Is the mean hardness of the polished and unpolished stones the same?

Based on the above t – test it is concluded that mean hardness of the polished and unpolished stones is not same.

Problem 4

Dental implant data: The hardness of metal implants in dental cavities depends on multiple factors, such as the method of implant, the temperature at which the metal is treated, the alloy used as well as the dentists who may favor one method above another and may work better in his/her favorite method. The response is the variable of interest.

In [431…

```
# Load dataset

df_d = pd.read_excel('Dental+Hardness+data.xlsx')

df_d.head() # returns first 5 rows
```

Out[431…

	Dentist	Method	Alloy	Temp	Response
0	1	1	1	1500	813
1	1	1	1	1600	792
2	1	1	1	1700	792
3	1	1	2	1500	907
4	1	1	2	1600	792

In [432…

```
df_d.shape # view the shape of the dataset
```

Out[432…

```
(90, 5)
```

In [433…

```
# check the data types of the columns in the dataset

df_d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Dentist   90 non-null     int64
 1   Method    90 non-null     int64
 2   Alloy     90 non-null     int64
 3   Temp      90 non-null     int64
 4   Response  90 non-null     int64
dtypes: int64(5)
memory usage: 3.6 KB
```

All columns are numerical.

There are no missing values in the dataset.

In [434…

```
# Changing Dentist, Method and Alloy data types to Categorial

df_d['Dentist']= pd.Categorical(df_d['Dentist'])
df_d['Method']= pd.Categorical(df_d['Method'])
df_d['Alloy']= pd.Categorical(df_d['Alloy'])
```

In [435…

```
# check the data types of the columns in the dataset

df_d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Dentist   90 non-null     category
 1   Method    90 non-null     category
 2   Alloy     90 non-null     category
 3   Temp      90 non-null     int64
 4   Response  90 non-null     int64
dtypes: category(3), int64(2)
memory usage: 2.3 KB
```

The `Dentist` , `Method` and `Alloy` columns are categorical while `Temp` and `Response` columns are numerical.

There are no missing values in the dataset.

In [436…

```
df_d.describe(include='all').T # statistical summary of the dataset
```

Out[436…

	count	unique	top	freq	mean	std	min	25%	50%	7!
Dentist	90.0	5.0	1.0	18.0	NaN	NaN	NaN	NaN	NaN	N
Method	90.0	3.0	1.0	30.0	NaN	NaN	NaN	NaN	NaN	N
Alloy	90.0	2.0	1.0	45.0	NaN	NaN	NaN	NaN	NaN	N
Temp	90.0	NaN	NaN	NaN	1600.000000	82.107083	1500.0	1500.0	1600.0	170(
Response	90.0	NaN	NaN	NaN	741.777778	145.767845	289.0	698.0	767.0	82

In [437…

```
df_d.Dentist.value_counts() # value counts of Dentists
```

Out[437…

```
Dentist
1    18
2    18
3    18
4    18
5    18
Name: count, dtype: int64
```

In [438…

```
df_d.Method.value_counts() # value counts of Methods
```

Out[438…

```
Method
1    30
2    30
3    30
Name: count, dtype: int64
```

In [439…

```
df_d.Alloy.value_counts() # value counts of Alloy Types
```

Out[439…

```
Alloy
1    45
2    45
Name: count, dtype: int64
```

There are five Dentists: 1, 2, 3, 4 and 5.

There are three Methods: 1, 2 and 3.

There are two Alloy Types: 1 and 2.

Sample is equally divided among Dentists, Methods and Alloy Types.

In [440…

```
df_d_a1 = df_d.loc[df_d['Alloy'] == 1] # Alloy Type 1
```

In [441…

```
df_d_a1.shape # view the shape of the dataset
```

Out[441…

```
(45, 5)
```

In [442…

```
df_d_a1.head() # returns first 5 rows
```

Out[442...

	Dentist	Method	Alloy	Temp	Response
0	1	1	1	1500	813
1	1	1	1	1600	792
2	1	1	1	1700	792
6	1	2	1	1500	782
7	1	2	1	1600	698

In [443...

```
df_d_a1.describe(include='all').T # statistical summary of the dataset
```

Out[443...

	count	unique	top	freq	mean	std	min	25%	50%	7!
Dentist	45.0	5.0	1.0	9.0	NaN	NaN	NaN	NaN	NaN	N
Method	45.0	3.0	1.0	15.0	NaN	NaN	NaN	NaN	NaN	N
Alloy	45.0	1.0	1.0	45.0	NaN	NaN	NaN	NaN	NaN	N
Temp	45.0	NaN	NaN	NaN	1600.000000	82.572282	1500.0	1500.0	1600.0	170
Response	45.0	NaN	NaN	NaN	707.488889	121.194551	289.0	681.0	743.0	78.

In [444...

```
df_d_a2 = df_d.loc[df_d['Alloy'] == 2] # Alloy Type 2
```

In [445...

```
df_d_a2.shape # view the shape of the dataset
```

Out[445...

```
(45, 5)
```

In [446...

```
df_d_a2.head() # returns first 5 rows
```

Out[446...

	Dentist	Method	Alloy	Temp	Response
3	1	1	2	1500	907
4	1	1	2	1600	792
5	1	1	2	1700	835
9	1	2	2	1500	1115
10	1	2	2	1600	835

In [447...

```
df_d_a2.describe(include='all').T # statistical summary of the dataset
```

Out[447…

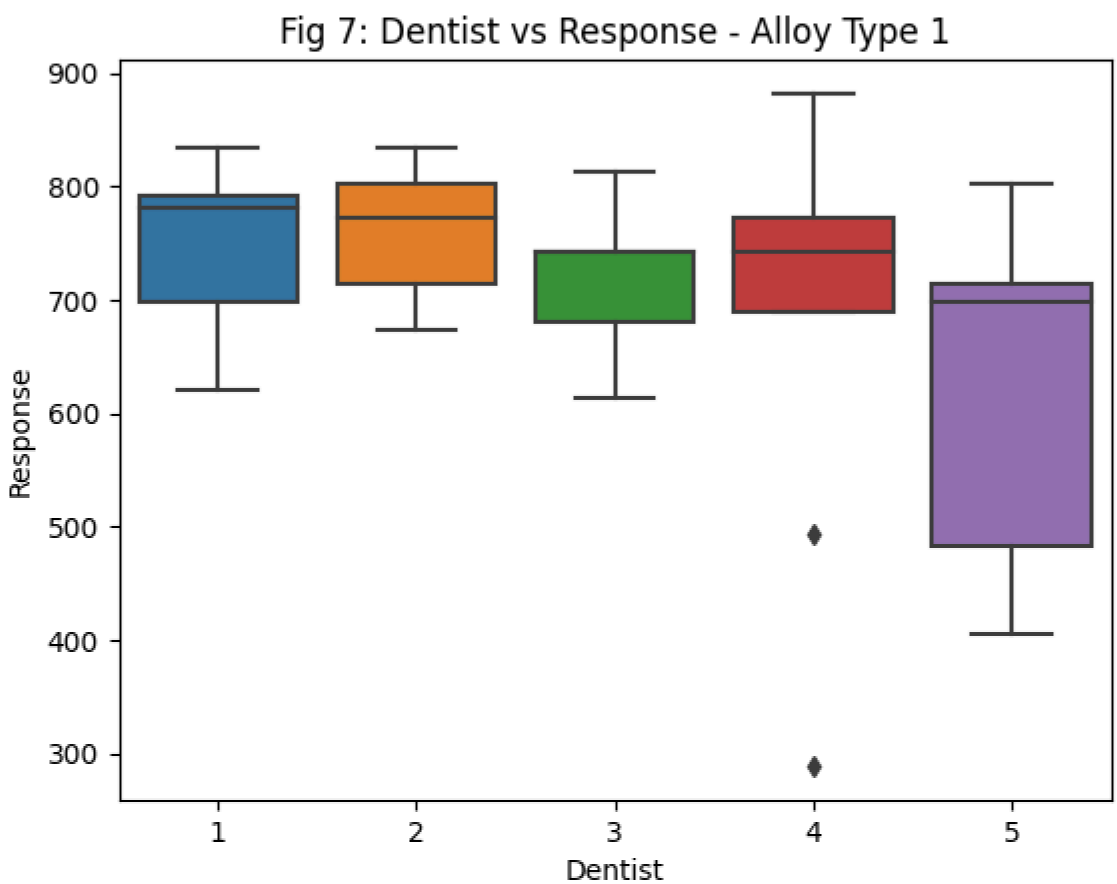
	count	unique	top	freq	mean	std	min	25%	50%	7!
Dentist	45.0	5.0	1.0	9.0	NaN	NaN	NaN	NaN	NaN	N
Method	45.0	3.0	1.0	15.0	NaN	NaN	NaN	NaN	NaN	N
Alloy	45.0	1.0	2.0	45.0	NaN	NaN	NaN	NaN	NaN	N
Temp	45.0	NaN	NaN	NaN	1600.000000	82.572282	1500.0	1500.0	1600.0	170
Response	45.0	NaN	NaN	NaN	776.066667	160.892595	312.0	715.0	824.0	85

4.1 How does the hardness of implants vary depending on dentists?

Visualize data

In [448…

```
# visual analysis of the Response for the the Dentists (Alloy Type 1)

sns.boxplot(x="Dentist", y="Response", data = df_d_a1)
plt.title("Fig 7: Dentist vs Response - Alloy Type 1")
plt.show()
```

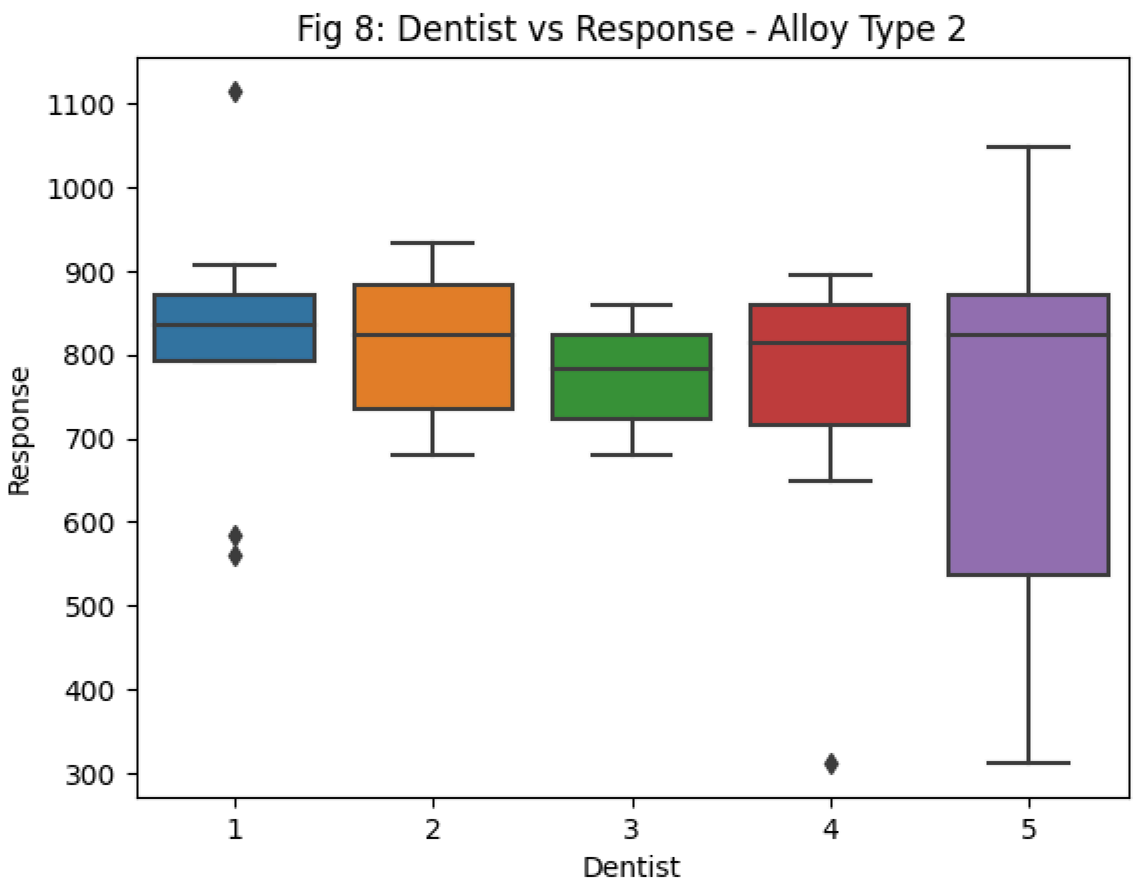


- The distribution of Response seems to differ among the Dentists.
- Dentist 4 seems to impact the highest Response.

- The median Response seems to be very close for the Dentist 1 and 2, Dentist 3 and 4 but the variation is higher in the Response by Dentist 5 in comparison to other Dentists.

In [449…

```
# visual analysis of the Response for the the Dentists (Alloy Type 2)

sns.boxplot(x="Dentist", y="Response", data = df_d_a2)
plt.title("Fig 8: Dentist vs Response - Alloy Type 2")
plt.show()
```



- The distribution of Response seems to differ among the Dentists.
- Dentist 1 seems to impact the highest Response.
- The median Response seems to be very close for the all Dentists but the variation is higher in the Response by Dentist 5 in comparison to other Dentists.

null and alternative hypotheses for the two types of alloys:

For Alloy Type 1:

Null Hypothesis (H_0): There is no difference among the dentists on the implant hardness.
Alternative Hypothesis (H_a): There is a difference among the dentists on the implant hardness.

For Alloy Type 2:

Null Hypothesis (H_0): There is no difference among the dentists on the implant hardness.
Alternative Hypothesis (H_a): There is a difference among the dentists on the implant hardness.

This is a problem, concerning five population means. One-way ANOVA is an appropriate test here provided normality and equality of variance assumptions are verified.

The dependent variable (the variable of interest) is on a continuous scale.
For testing of normality, Shapiro-Wilk's test is applied to the response variable.
For equality of variance, Levene test is applied to the response variable.

Significance level:

α = 0.05

Shapiro-Wilk's test

We will test the null hypothesis

> H_0 : Response follow a normal distribution

against the alternative hypothesis

> H_a : Response do not not follow a normal distribution

In [450…

```
# Assumption 1: Normality (Alloy Type 1)

# Use the shapiro function for the scipy.stats library for this test

# find the p-value

w, p_value = stats.shapiro(df_d_a1['Response'])

print('The p-value is', p_value)
```

```
The p-value is 1.1945070582441986e-05
```

In [451…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 1.1945070582441986e-05 is less than the level of significance, we rej
ect the null hypothesis.
```

In [452…

```
# Assumption 1: Normality (Alloy Type 2)

# Use the shapiro function for the scipy.stats library for this test

# find the p-value

w, p_value = stats.shapiro(df_d_a2['Response'])
```

```
print('The p-value is', p_value)
```

```
The p-value is 0.00040293222991749644
```

In [453…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we 
```

```
As the p-value 0.00040293222991749644 is less than the level of significance, we rej
ect the null hypothesis.
```

Levene’s test

We will test the null hypothesis

> H_0: All the population variances are equal

against the alternative hypothesis

> H_a: At least one variance is different from the rest

In [454…

```
#Assumption 2: Homogeneity of Variance (Alloy Type 1)

# use levene function from scipy.stats library for this test

# find the p-value

statistic, p_value = stats.levene(df_d_a1[df_d_a1['Dentist']==1]['Response'],
                                   df_d_a1[df_d_a1['Dentist']==2]['Response'],
                                   df_d_a1[df_d_a1['Dentist']==3]['Response'],
                                   df_d_a1[df_d_a1['Dentist']==4]['Response'],
                                   df_d_a1[df_d_a1['Dentist']==5]['Response']
                                         )
print('The p-value is', p_value)
```

```
The p-value is 0.2565537418543793
```

In [455…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we 
```

```
As the p-value 0.2565537418543793 is greater than the level of significance, we fail
to reject the null hypothesis.
```

In [456…

```
#Assumption 2: Homogeneity of Variance (Alloy Type 2)

# use levene function from scipy.stats library for this test

# find the p-value

statistic, p_value = stats.levene(df_d_a2[df_d_a2['Dentist']==1]['Response'],
                                   df_d_a2[df_d_a2['Dentist']==2]['Response'],
                                   df_d_a2[df_d_a2['Dentist']==3]['Response'],
```

```
                                df_d_a2[df_d_a2['Dentist']==4]['Response'],
                                df_d_a2[df_d_a2['Dentist']==5]['Response']
                                     )
print('The p-value is', p_value)
```

```
The p-value is 0.23686777576324947
```

In [457…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 0.23686777576324947 is greater than the level of significance, we fai
l to reject the null hypothesis.
```

We will use the `f_oneway()` function to perform a one-way ANOVA test. The `f_oneway()` function takes the sample observations from the different groups and returns the test statistic and the p-value for the test.

The sample observations are the values of Response with respect to the Dentists.

In [458…

```
from scipy.stats import f_oneway # import the required function
from statsmodels.stats.multicomp import pairwise_tukeyhsd # For pairwise tukey test
```

In [459…

```
# For Alloy Type 1

# create separate variables to store the Response with respect to the Dentists

Dentist_1 = df_d_a1[df_d_a1['Dentist']==1]['Response']
Dentist_2 = df_d_a1[df_d_a1['Dentist']==2]['Response']
Dentist_3 = df_d_a1[df_d_a1['Dentist']==3]['Response']
Dentist_4 = df_d_a1[df_d_a1['Dentist']==4]['Response']
Dentist_5 = df_d_a1[df_d_a1['Dentist']==5]['Response']
```

In [460…

```
# find the p-value
test_stat, p_value = f_oneway(Dentist_1, Dentist_2, Dentist_3, Dentist_4, Dentist_5
print('The p-value is', p_value)
```

```
The p-value is 0.11656712140267618
```

In [461…

```
# print the conclusion based on p-value
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 0.11656712140267618 is greater than the level of significance, we fai
l to reject the null hypothesis.
```

In [462…

```
# For Alloy Type 2

# create separate variables to store the Response with respect to the Dentists

Dentist_1 = df_d_a2[df_d_a2['Dentist']==1]['Response']
Dentist_2 = df_d_a2[df_d_a2['Dentist']==2]['Response']
Dentist_3 = df_d_a2[df_d_a2['Dentist']==3]['Response']
```

```
Dentist_4 = df_d_a2[df_d_a2['Dentist']==4]['Response']
Dentist_5 = df_d_a2[df_d_a2['Dentist']==5]['Response']
```

In [463…

```
# find the p-value
test_stat, p_value = f_oneway(Dentist_1, Dentist_2, Dentist_3, Dentist_4, Dentist_5
print('The p-value is', p_value)
```

```
The p-value is 0.7180309510793431
```

In [464…

```
# print the conclusion based on p-value
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 0.7180309510793431 is greater than the level of significance, we fail
to reject the null hypothesis.
```

Since the p-value is more than the level of significance (5%) for both Alloy Types, we fail to reject the null hypothesis. Hence, we have enough statistical evidence to say that there is no difference among the dentists on the implant hardness for both Alloy Types.

In [465…

```
# pairwise tukey test between Response and Dentist (Alloy Type 1)

tukey = pairwise_tukeyhsd(endog=df_d_a1['Response'],
                          groups=df_d_a1['Dentist'],
                          alpha=0.05)
print(tukey)
```

```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
========================================================
group1 group2  meandiff p-adj    lower     upper   reject
--------------------------------------------------------
     1      2   11.3333 0.9996 -145.0423  167.709  False
     1      3  -32.3333 0.9757  -188.709 124.0423  False
     1      4  -68.7778 0.7189 -225.1535  87.5979  False
     1      5 -122.2222 0.1889 -278.5979  34.1535  False
     2      3  -43.6667 0.9298 -200.0423  112.709  False
     2      4  -80.1111 0.5916 -236.4868  76.2646  False
     2      5 -133.5556 0.1258 -289.9312  22.8201  False
     3      4  -36.4444 0.9626 -192.8201 119.9312  False
     3      5  -89.8889 0.4805 -246.2646  66.4868  False
     4      5  -53.4444 0.8643 -209.8201 102.9312  False
--------------------------------------------------------
```

In [466…

```
# pairwise tukey test between Response and Dentist (Alloy Type 2)

tukey = pairwise_tukeyhsd(endog=df_d_a2['Response'],
                          groups=df_d_a2['Dentist'],
                          alpha=0.05)
print(tukey)
```

```
  Multiple Comparison of Means - Tukey HSD, FWER=0.05
=======================================================
group1 group2 meandiff p-adj   lower    upper   reject
-------------------------------------------------------
     1      2  -4.1111    1.0 -225.5687 217.3465  False
     1      3 -36.5556 0.9895 -258.0131  184.902  False
     1      4    -70.0 0.8941 -291.4576 151.4576  False
     1      5 -90.1111 0.7724 -311.5687 131.3465  False
     2      3 -32.4444 0.9933  -253.902 189.0131  False
     2      4 -65.8889 0.9132 -287.3465 155.5687  False
     2      5    -86.0 0.8008 -307.4576 135.4576  False
     3      4 -33.4444 0.9925  -254.902 188.0131  False
     3      5 -53.5556 0.9574 -275.0131  167.902  False
     4      5 -20.1111  0.999 -241.5687 201.3465  False
-------------------------------------------------------
```

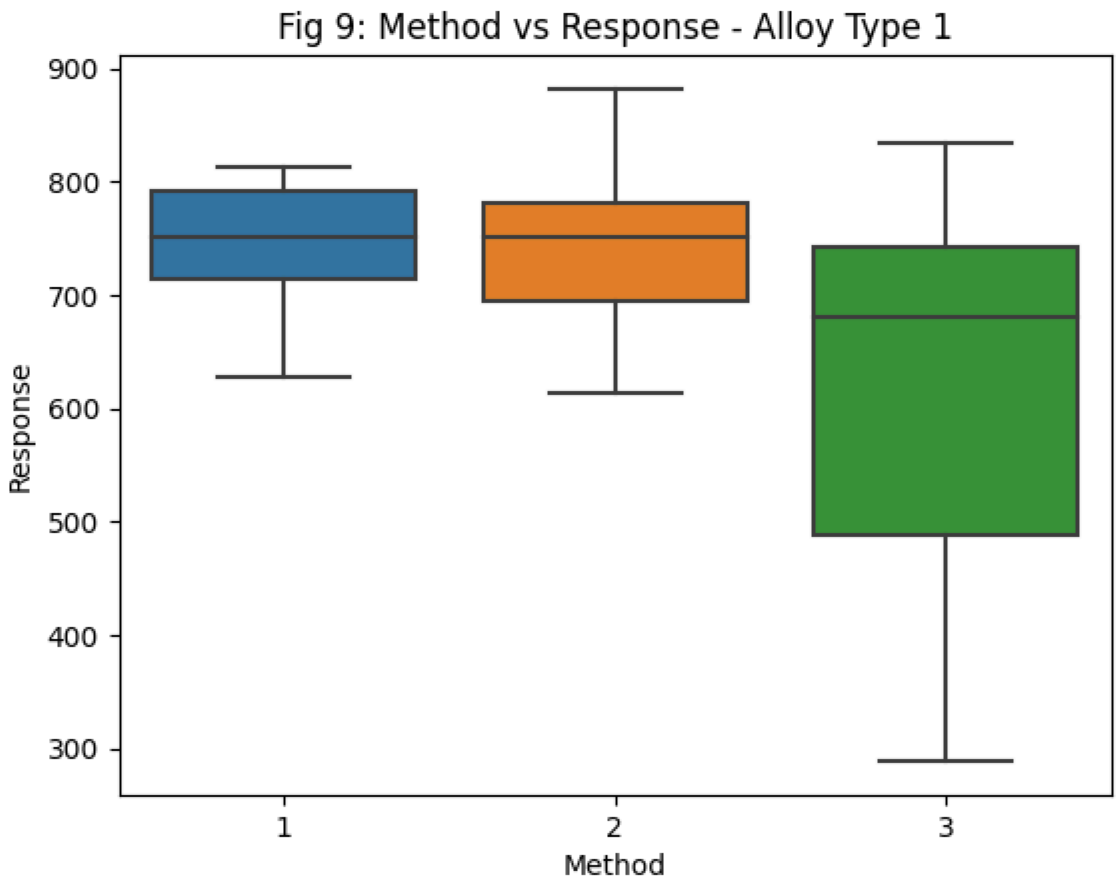
Thus, we can conclude that there is no statistically significant difference between the means of Dentists for both Alloy Types.

4.2 How does the hardness of implants vary depending on methods?

Visualize data

In [467…

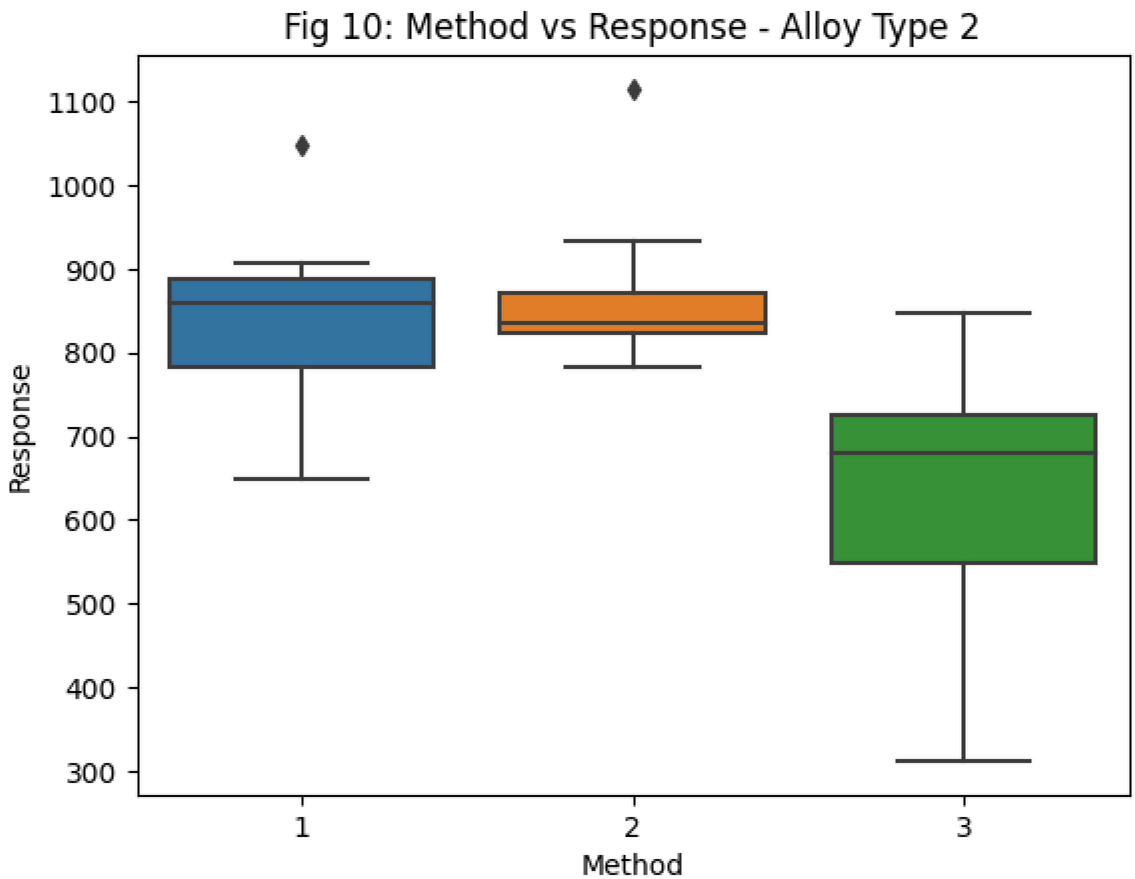
```
# visual analysis of the Response for the the Methods (Alloy Type 1)

sns.boxplot(x="Method", y="Response", data = df_d_a1)
plt.title("Fig 9: Method vs Response - Alloy Type 1")
plt.show()
```



- The distribution of Response seems to differ among the Methods.
- Method 2 seems to impact the highest Response.
- The median Response seems to be very close for the Method 1 and 2 but the variation is higher in the Response by Method 3 in comparison to other Methods.

In [468…

```
# visual analysis of the Response for the the Methods (Alloy Type 2)

sns.boxplot(x="Method", y="Response", data = df_d_a2)
plt.title("Fig 10: Method vs Response - Alloy Type 2")
plt.show()
```



- The distribution of Response seems to differ among the Methods.
- Method 2 seems to impact the highest Response.
- The median Response seems to be very close for the Method 1 and 2 but the variation is higher in the Response by Method 3 in comparison to other Methods.

null and alternative hypotheses for the two types of alloys:

For Alloy Type 1:

Null Hypothesis (H_0): There is no difference among the methods on the implant hardness.
Alternative Hypothesis (H_a): There is a difference among the methods on the implant hardness.

For Alloy Type 2:

Null Hypothesis (H_0): There is no difference among the methods on the implant hardness.
Alternative Hypothesis (H_a): There is a difference among the methods on the implant hardness.

This is a problem, concerning three population means. One-way ANOVA is an appropriate test here provided normality and equality of variance assumptions are verified.

The dependent variable (the variable of interest) is on a continuous scale.
For testing of normality, Shapiro-Wilk's test is applied to the response variable.
For equality of variance, Levene test is applied to the response variable.

Significance level:

α = 0.05

Shapiro-Wilk's test

We will test the null hypothesis

> H_0 : Response follow a normal distribution

against the alternative hypothesis

> H_a : Response do not not follow a normal distribution

In [469…

```
# Assumption 1: Normality (Alloy Type 1)

# Use the shapiro function for the scipy.stats library for this test

# find the p-value

w, p_value = stats.shapiro(df_d_a1['Response'])

print('The p-value is', p_value)
```

```
The p-value is 1.1945070582441986e-05
```

In [470…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 1.1945070582441986e-05 is less than the level of significance, we rej
ect the null hypothesis.
```

In [471…

```
# Assumption 1: Normality (Alloy Type 2)

# Use the shapiro function for the scipy.stats library for this test

# find the p-value

w, p_value = stats.shapiro(df_d_a2['Response'])

print('The p-value is', p_value)
```

```
The p-value is 0.00040293222991749644
```

In [472…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

As the p-value 0.00040293222991749644 is less than the level of significance, we reject the null hypothesis.

Levene's test

We will test the null hypothesis

> H_0: All the population variances are equal

against the alternative hypothesis

> H_a: At least one variance is different from the rest

In [473...

```
#Assumption 2: Homogeneity of Variance (Alloy Type 1)

# use levene function from scipy.stats library for this test

# find the p-value

statistic, p_value = stats.levene(df_d_a1[df_d_a1['Method']==1]['Response'],
                                  df_d_a1[df_d_a1['Method']==2]['Response'],
                                  df_d_a1[df_d_a1['Method']==3]['Response']
                                        )
print('The p-value is', p_value)
```

```
The p-value is 0.003416038146023399
```

In [474...

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 0.003416038146023399 is less than the level of significance, we reject the null hypothesis.
```

In [475...

```
#Assumption 2: Homogeneity of Variance (Alloy Type 2)

# use levene function from scipy.stats library for this test

# find the p-value

statistic, p_value = stats.levene(df_d_a2[df_d_a2['Method']==1]['Response'],
                                  df_d_a2[df_d_a2['Method']==2]['Response'],
                                  df_d_a2[df_d_a2['Method']==3]['Response']
                                        )
print('The p-value is', p_value)
```

```
The p-value is 0.04469269939158666
```

In [476...

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 0.04469269939158666 is less than the level of significance, we reject the null hypothesis.
```

In [477…

```
# For Alloy Type 1

# create separate variables to store the Response with respect to the Dentists

Method_1 = df_d_a1[df_d_a1['Method']==1]['Response']
Method_2 = df_d_a1[df_d_a1['Method']==2]['Response']
Method_3 = df_d_a1[df_d_a1['Method']==3]['Response']
```

In [478…

```
# find the p-value
test_stat, p_value = f_oneway(Method_1, Method_2, Method_3)
print('The p-value is', p_value)
```

```
The p-value is 0.004163412167505541
```

In [479…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 0.004163412167505541 is less than the level of significance, we rejec
t the null hypothesis.
```

In [480…

```
# For Alloy Type 2

# create separate variables to store the Response with respect to the Dentists

Method_1 = df_d_a2[df_d_a2['Method']==1]['Response']
Method_2 = df_d_a2[df_d_a2['Method']==2]['Response']
Method_3 = df_d_a2[df_d_a2['Method']==3]['Response']
```

In [481…

```
# find the p-value
test_stat, p_value = f_oneway(Method_1, Method_2, Method_3)
print('The p-value is', p_value)
```

```
The p-value is 5.415871051443185e-06
```

In [482…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 5.415871051443185e-06 is less than the level of significance, we reje
ct the null hypothesis.
```

Since the p-value is less than the level of significance (5%) for both Alloy Types, we reject the null hypothesis. Hence, we have enough statistical evidence to say that there is difference among the methods on the implant hardness for both Alloy Types.

In [483…

```
# pairwise tukey test between Response and Method (Alloy Type 1)

tukey = pairwise_tukeyhsd(endog=df_d_a1['Response'],
                          groups=df_d_a1['Method'],
                          alpha=0.05)
print(tukey)
```

```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
=========================================================
group1 group2  meandiff p-adj    lower     upper   reject
---------------------------------------------------------
     1      2   -6.1333  0.987  -102.714  90.4473  False
     1      3    -124.8 0.0085 -221.3807 -28.2193   True
     2      3 -118.6667 0.0128 -215.2473  -22.086   True
---------------------------------------------------------
```

In [484…

```python
# pairwise tukey test between Response and Method (Alloy Type 2)

tukey = pairwise_tukeyhsd(endog=df_d_a2['Response'],
                          groups=df_d_a2['Method'],
                          alpha=0.05)
print(tukey)
```

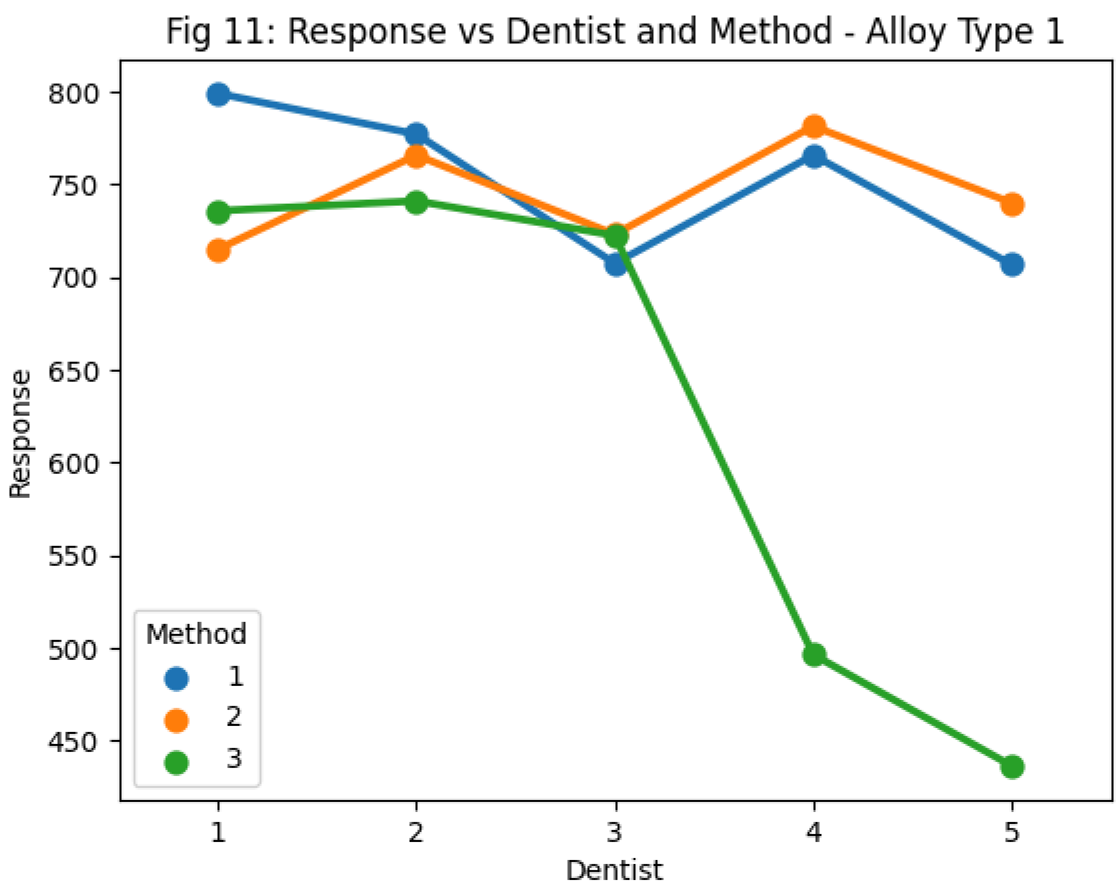
```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
=========================================================
group1 group2 meandiff p-adj    lower     upper   reject
---------------------------------------------------------
     1      2     27.0 0.8212  -82.4546 136.4546  False
     1      3   -208.8 0.0001 -318.2546  -99.3454   True
     2      3   -235.8    0.0 -345.2546 -126.3454   True
---------------------------------------------------------
```

Thus, we can conclude that there is a statistically significant difference between the means of Method 1 and 3 and Method 2 and 3 for both Alloy Types, but not a statistically significant difference between the means of Method 1 and 2 for both Alloy Types.

4.3 What is the interaction effect between the dentist and method on the hardness of dental implants for each type of alloy?

In [485…

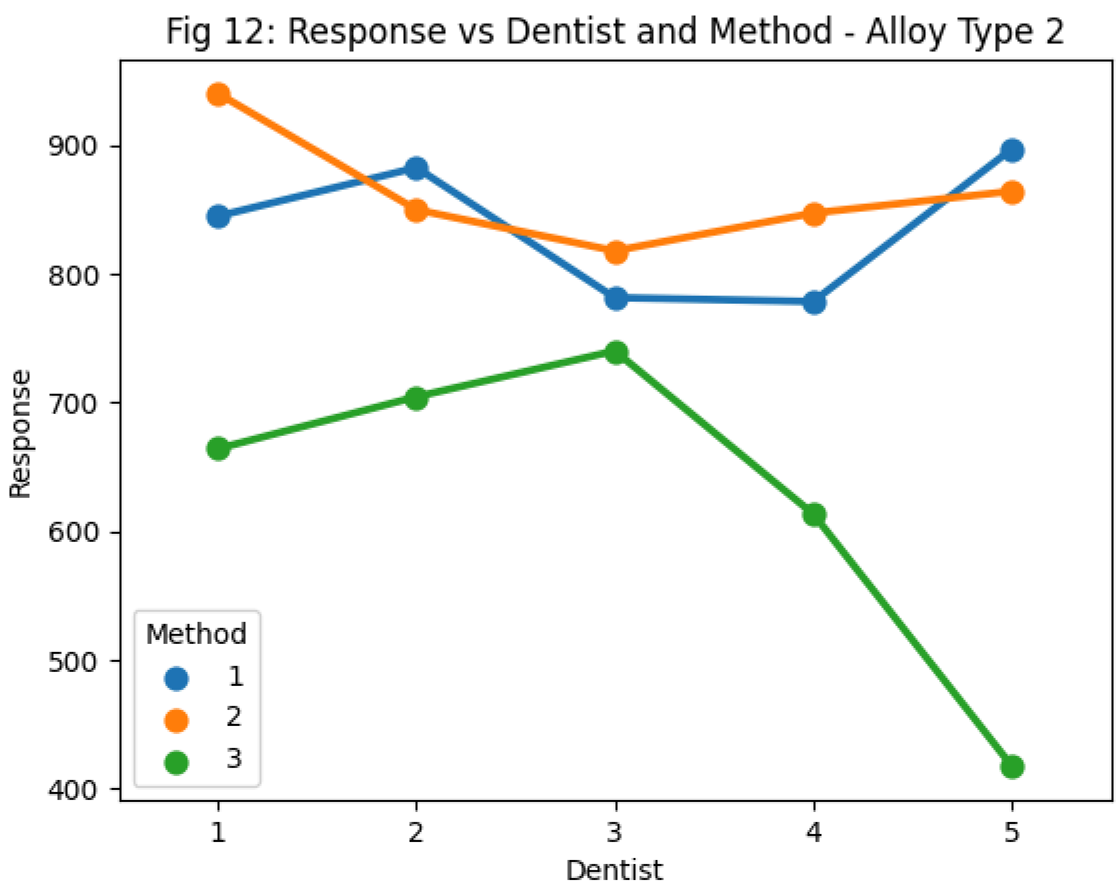
```python
# visual analysis of interaction between Response vs Dentists and Methods (Alloy Ty

sns.pointplot(x='Dentist', y='Response', data=df_d_a1, hue='Method', errorbar=None)
plt.title("Fig 11: Response vs Dentist and Method - Alloy Type 1")
plt.show()
```



There is significant interaction effect between the Dentists and Methods on the hardness of dental implants for Alloy Type 1.

In [486…

```
# visual analysis of interaction between Response vs Dentists and Methods (Alloy Ty

sns.pointplot(x='Dentist', y='Response', data=df_d_a2, hue='Method', errorbar=None)
plt.title("Fig 12: Response vs Dentist and Method - Alloy Type 2")
plt.show()
```



There is significant interaction effect between the Dentists and Methods 1, 2 on the hardness of dental implants for Alloy Type 2. However, there is no interaction effect between the Dentists and Method 3 on the hardness of dental implants for Alloy Type 2.

4.4 How does the hardness of implants vary depending on dentists and methods together?

In [487…

```
import statsmodels.api as sm
from statsmodels.formula.api import ols # For n-way ANOVA
from statsmodels.stats.anova import _get_covariance,anova_lm # For n-way ANOVA
```

null and alternative hypotheses for the two types of alloys:

For Alloy Type 1:

Null Hypothesis (H_0): Hardness of implants does not vary depending on Dentists and Methods together.
Alternative Hypothesis (H_a): Hardness of implants vary depending on Dentists and Methods together.

For Alloy Type 2:

Null Hypothesis (H_0): Hardness of implants does not vary depending on Dentists and Methods together.

Alternative Hypothesis (H_a): Hardness of implants vary depending on Dentists and Methods together.

Here response is affected by two factors. Hence Two-way ANOVA is an appropriate test here provided normality and equality of variance assumptions are verified.

The dependent variable (the variable of interest) is on a continuous scale.
For testing of normality, Shapiro-Wilk's test is applied to the response variable.
For equality of variance, Levene test is applied to the response variable.

Significance level:

α = 0.05

Shapiro-Wilk's test

We will test the null hypothesis

> H_0 : Response follow a normal distribution

against the alternative hypothesis

> H_a : Response do not not follow a normal distribution

In [488…
```
# Assumption 1: Normality (Alloy Type 1)

# Use the shapiro function for the scipy.stats library for this test

# find the p-value

w, p_value = stats.shapiro(df_d_a1['Response'])

print('The p-value is', p_value)
```

```
The p-value is 1.1945070582441986e-05
```

In [489…
```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we
```

```
As the p-value 1.1945070582441986e-05 is less than the level of significance, we rej
ect the null hypothesis.
```

In [490…
```
# Assumption 1: Normality (Alloy Type 2)

# Use the shapiro function for the scipy.stats library for this test

# find the p-value

w, p_value = stats.shapiro(df_d_a2['Response'])
```

```
print('The p-value is', p_value)
```

```
The p-value is 0.00040293222991749644
```

In [491…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we 
```

```
As the p-value 0.00040293222991749644 is less than the level of significance, we rej
ect the null hypothesis.
```

Levene's test

We will test the null hypothesis

> H_0: All the population variances are equal

against the alternative hypothesis

> H_a: At least one variance is different from the rest

In [492…

```
# For Alloy Type 1

# create separate variables to store the Response with respect to the Dentists and

Dentist_1_Method_1 = df_d_a1[(df_d_a1['Dentist']==1) & (df_d_a1['Method']==1)]['Res
Dentist_1_Method_2 = df_d_a1[(df_d_a1['Dentist']==1) & (df_d_a1['Method']==2)]['Res
Dentist_1_Method_3 = df_d_a1[(df_d_a1['Dentist']==1) & (df_d_a1['Method']==3)]['Res

Dentist_2_Method_1 = df_d_a1[(df_d_a1['Dentist']==2) & (df_d_a1['Method']==1)]['Res
Dentist_2_Method_2 = df_d_a1[(df_d_a1['Dentist']==2) & (df_d_a1['Method']==2)]['Res
Dentist_2_Method_3 = df_d_a1[(df_d_a1['Dentist']==2) & (df_d_a1['Method']==3)]['Res

Dentist_3_Method_1 = df_d_a1[(df_d_a1['Dentist']==3) & (df_d_a1['Method']==1)]['Res
Dentist_3_Method_2 = df_d_a1[(df_d_a1['Dentist']==3) & (df_d_a1['Method']==2)]['Res
Dentist_3_Method_3 = df_d_a1[(df_d_a1['Dentist']==3) & (df_d_a1['Method']==3)]['Res

Dentist_4_Method_1 = df_d_a1[(df_d_a1['Dentist']==4) & (df_d_a1['Method']==1)]['Res
Dentist_4_Method_2 = df_d_a1[(df_d_a1['Dentist']==4) & (df_d_a1['Method']==2)]['Res
Dentist_4_Method_3 = df_d_a1[(df_d_a1['Dentist']==4) & (df_d_a1['Method']==3)]['Res

Dentist_5_Method_1 = df_d_a1[(df_d_a1['Dentist']==5) & (df_d_a1['Method']==1)]['Res
Dentist_5_Method_2 = df_d_a1[(df_d_a1['Dentist']==5) & (df_d_a1['Method']==2)]['Res
Dentist_5_Method_3 = df_d_a1[(df_d_a1['Dentist']==5) & (df_d_a1['Method']==3)]['Res
```

In [493…

```
#Assumption 2: Homogeneity of Variance (Alloy Type 1)

# use levene function from scipy.stats library for this test

# find the p-value

statistic, p_value = stats.levene(Dentist_1_Method_1, Dentist_1_Method_2, Dentist_1
                              Dentist_2_Method_2, Dentist_2_Method_3, Dentist_3_Met
```

```
                              Dentist_3_Method_3, Dentist_4_Method_1, Dentist_4_Met
                              Dentist_5_Method_1, Dentist_5_Method_2, Dentist_5_Met
print('The p-value is', p_value)
```

```
The p-value is 0.3128166652989495
```

In [494…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we 
```

```
As the p-value 0.3128166652989495 is greater than the level of significance, we fail
to reject the null hypothesis.
```

In [495…

```
# For Alloy Type 1

# create separate variables to store the Response with respect to the Dentists and

Dentist_1_Method_1 = df_d_a2[(df_d_a2['Dentist']==1) & (df_d_a2['Method']==1)]['Res
Dentist_1_Method_2 = df_d_a2[(df_d_a2['Dentist']==1) & (df_d_a2['Method']==2)]['Res
Dentist_1_Method_3 = df_d_a2[(df_d_a2['Dentist']==1) & (df_d_a2['Method']==3)]['Res

Dentist_2_Method_1 = df_d_a2[(df_d_a2['Dentist']==2) & (df_d_a2['Method']==1)]['Res
Dentist_2_Method_2 = df_d_a2[(df_d_a2['Dentist']==2) & (df_d_a2['Method']==2)]['Res
Dentist_2_Method_3 = df_d_a2[(df_d_a2['Dentist']==2) & (df_d_a2['Method']==3)]['Res

Dentist_3_Method_1 = df_d_a2[(df_d_a2['Dentist']==3) & (df_d_a2['Method']==1)]['Res
Dentist_3_Method_2 = df_d_a2[(df_d_a2['Dentist']==3) & (df_d_a2['Method']==2)]['Res
Dentist_3_Method_3 = df_d_a2[(df_d_a2['Dentist']==3) & (df_d_a2['Method']==3)]['Res

Dentist_4_Method_1 = df_d_a2[(df_d_a2['Dentist']==4) & (df_d_a2['Method']==1)]['Res
Dentist_4_Method_2 = df_d_a2[(df_d_a2['Dentist']==4) & (df_d_a2['Method']==2)]['Res
Dentist_4_Method_3 = df_d_a2[(df_d_a2['Dentist']==4) & (df_d_a2['Method']==3)]['Res

Dentist_5_Method_1 = df_d_a2[(df_d_a2['Dentist']==5) & (df_d_a2['Method']==1)]['Res
Dentist_5_Method_2 = df_d_a2[(df_d_a2['Dentist']==5) & (df_d_a2['Method']==2)]['Res
Dentist_5_Method_3 = df_d_a2[(df_d_a2['Dentist']==5) & (df_d_a2['Method']==3)]['Res
```

In [496…

```
#Assumption 2: Homogeneity of Variance (Alloy Type 2)

# use levene function from scipy.stats library for this test

# find the p-value

statistic, p_value = stats.levene(Dentist_1_Method_1, Dentist_1_Method_2, Dentist_1
                              Dentist_2_Method_2, Dentist_2_Method_3, Dentist_3_Met
                              Dentist_3_Method_3, Dentist_4_Method_1, Dentist_4_Met
                              Dentist_5_Method_1, Dentist_5_Method_2, Dentist_5_Met
print('The p-value is', p_value)
```

```
The p-value is 0.7831735515657827
```

In [497…

```
if p_value < 0.05:
    print(f'As the p-value {p_value} is less than the level of significance, we rej
else:
    print(f'As the p-value {p_value} is greater than the level of significance, we 
```

```
As the p-value 0.7831735515657827 is greater than the level of significance, we fail
to reject the null hypothesis.
```

Two-way ANOVA

In [498…

```python
# For Alloy Type 1

mod = ols('Response ~ Dentist*Method', data = df_d_a1).fit()
aov_tbl = sm.stats.anova_lm(mod, type = 2)
aov_tbl
```

Out[498…

	df	sum_sq	mean_sq	F	PR(>F)
Dentist	4.0	106683.688889	26670.922222	3.899638	0.011484
Method	2.0	148472.177778	74236.088889	10.854287	0.000284
Dentist:Method	8.0	185941.377778	23242.672222	3.398383	0.006793
Residual	30.0	205180.000000	6839.333333	NaN	NaN

Since the p-values 0.011484, 0.000284 for Dentist and Method are both less than the level of significance (5%) for Alloy Type 1, this means that both factors have a statistically significant effect on Response.

Since the p-value 0.006793 is less than the level of significance (5%) for Alloy Type 1, we reject the null hypothesis. Hence, we have enough statistical evidence to say that hardness of implants vary depending on Dentists and Methods together.

In [499…

```python
# pairwise tukey test between Response and Dentist (Alloy Type 1)

tukey = pairwise_tukeyhsd(endog=df_d_a1['Response'],
                          groups=df_d_a1['Dentist'],
                          alpha=0.05)
print(tukey)
```

```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
========================================================
group1 group2  meandiff p-adj    lower    upper   reject
--------------------------------------------------------
     1      2   11.3333 0.9996 -145.0423  167.709  False
     1      3  -32.3333 0.9757  -188.709 124.0423  False
     1      4  -68.7778 0.7189 -225.1535  87.5979  False
     1      5 -122.2222 0.1889 -278.5979  34.1535  False
     2      3  -43.6667 0.9298 -200.0423  112.709  False
     2      4  -80.1111 0.5916 -236.4868  76.2646  False
     2      5 -133.5556 0.1258 -289.9312  22.8201  False
     3      4  -36.4444 0.9626 -192.8201 119.9312  False
     3      5  -89.8889 0.4805 -246.2646  66.4868  False
     4      5  -53.4444 0.8643 -209.8201 102.9312  False
--------------------------------------------------------
```

Thus, we can conclude that there is no statistically significant difference between the means of Dentists for Alloy Type 1.

In [500…

```
# pairwise tukey test between Response and Method (Alloy Type 1)

tukey = pairwise_tukeyhsd(endog=df_d_a1['Response'],
                          groups=df_d_a1['Method'],
                          alpha=0.05)
print(tukey)
```

```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
========================================================
group1 group2  meandiff p-adj    lower    upper   reject
--------------------------------------------------------
     1      2   -6.1333  0.987  -102.714  90.4473  False
     1      3    -124.8 0.0085 -221.3807 -28.2193   True
     2      3 -118.6667 0.0128 -215.2473  -22.086   True
--------------------------------------------------------
```

Thus, we can conclude that there is a statistically significant difference between the means of Method 1 and 3 and Method 2 and 3 for Alloy Type 1, but not a statistically significant difference between the means of Method 1 and 2 for Alloy Type 1.

In [501…

```
# For Alloy Type 2

mod = ols('Response ~ Dentist*Method', data = df_d_a2).fit()
aov_tbl = sm.stats.anova_lm(mod, type = 2)
aov_tbl
```

Out[501…

	df	sum_sq	mean_sq	F	PR(>F)
Dentist	4.0	56797.911111	14199.477778	1.106152	0.371833
Method	2.0	499640.400000	249820.200000	19.461218	0.000004
Dentist:Method	8.0	197459.822222	24682.477778	1.922787	0.093234
Residual	30.0	385104.666667	12836.822222	NaN	NaN

Since the p-value 0.371833 for Dentist is more than the level of significance (5%) for Alloy Type 2, this means it does not have statistically significant effect on Response.

Since the p-value 0.000004 for Method is less than the level of significance (5%) for Alloy Type 2, this means it has statistically significant effect on Response.

Since the p-value 0.093234 is more than the level of significance (5%) for Alloy Type 2, we fail to reject the null hypothesis. Hence, we have enough statistical evidence to say that hardness of implants does not vary depending on Dentists and Methods together.

In [502…

```
# pairwise tukey test between Response and Dentist (Alloy Type 2)

tukey = pairwise_tukeyhsd(endog=df_d_a2['Response'],
                          groups=df_d_a2['Dentist'],
                          alpha=0.05)
print(tukey)
```

```
  Multiple Comparison of Means - Tukey HSD, FWER=0.05
=========================================================
group1 group2 meandiff p-adj    lower    upper   reject
---------------------------------------------------------
     1      2  -4.1111    1.0 -225.5687 217.3465  False
     1      3 -36.5556 0.9895 -258.0131  184.902  False
     1      4    -70.0 0.8941 -291.4576 151.4576  False
     1      5 -90.1111 0.7724 -311.5687 131.3465  False
     2      3 -32.4444 0.9933  -253.902 189.0131  False
     2      4 -65.8889 0.9132 -287.3465 155.5687  False
     2      5    -86.0 0.8008 -307.4576 135.4576  False
     3      4 -33.4444 0.9925  -254.902 188.0131  False
     3      5 -53.5556 0.9574 -275.0131  167.902  False
     4      5 -20.1111  0.999 -241.5687 201.3465  False
---------------------------------------------------------
```

Thus, we can conclude that there is no statistically significant difference between the means of Dentists for Alloy Type 2.

In [503…

```
# pairwise tukey test between Response and Method (Alloy Type 2)

tukey = pairwise_tukeyhsd(endog=df_d_a2['Response'],
                          groups=df_d_a2['Method'],
                          alpha=0.05)
print(tukey)
```

```
  Multiple Comparison of Means - Tukey HSD, FWER=0.05
==========================================================
group1 group2 meandiff p-adj    lower     upper   reject
----------------------------------------------------------
     1      2     27.0 0.8212  -82.4546  136.4546  False
     1      3   -208.8 0.0001 -318.2546  -99.3454   True
     2      3   -235.8    0.0 -345.2546 -126.3454   True
----------------------------------------------------------
```

Thus, we can conclude that there is a statistically significant difference between the means of Method 1 and 3 and Method 2 and 3 for Alloy Type 2, but not a statistically significant difference between the means of Method 1 and 2 for Alloy Type 2.