**Observation Analysis:**

* The bar graph showed the Drug Regimen Capomulin has the maximum mice number (230), and Zoniferol has the smaller mice number (182).By removing duplicates the total number of mice is 248. The total count of mice by gender also showed that 124 female mice and 125 male mice.
* The correlation between mouse weight, and average tumor volume is 0.84. It is a strong positive correlation, when the mouse weight increases the average tumor volume also increases.
* The regression analysis helped to understand how much the average tumor volume (dependent variable) will change when weight of mice change(independent variables). The R-squared value is 0.70, which means 70% the model fit the data, wich is fairely good to predict the data from the model. Higher R-squared values represent smaller differences between the observed data, and the fitted value. 70% the model explains all of the variation in the response variable around its mean.
* From the selected treatments Capomulin and Ramicane reduces the size of tumors better.

# Dependencies and Setup

import matplotlib.pyplot as plt

import pandas as pd

import scipy.stats as st

import numpy as np

from scipy.stats import linregress

# Study data files

mouse\_metadata\_path = "data/Mouse\_metadata.csv"

study\_results\_path = "data/Study\_results.csv"

# Read the mouse data

mouse\_metadata = pd.read\_csv(mouse\_metadata\_path)

mouse\_metadata.head()

|  | **Mouse ID** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** |
| --- | --- | --- | --- | --- | --- |
| **0** | k403 | Ramicane | Male | 21 | 16 |
| **1** | s185 | Capomulin | Female | 3 | 17 |
| **2** | x401 | Capomulin | Female | 16 | 15 |
| **3** | m601 | Capomulin | Male | 22 | 17 |
| **4** | g791 | Ramicane | Male | 11 | 16 |

#Read the study results

study\_results = pd.read\_csv(study\_results\_path)

study\_results.head()

|  | **Mouse ID** | **Timepoint** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- |
| **0** | b128 | 0 | 45.0 | 0 |
| **1** | f932 | 0 | 45.0 | 0 |
| **2** | g107 | 0 | 45.0 | 0 |
| **3** | a457 | 0 | 45.0 | 0 |
| **4** | c819 | 0 | 45.0 | 0 |

**Combine the above data into a single dataset**

# Combine the above data into a single dataset

combined\_df = pd.merge(mouse\_metadata, study\_results, how='outer', on="Mouse ID")

# Display the data table for preview

combined\_df.head()

|  | **Mouse ID** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Timepoint** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | k403 | Ramicane | Male | 21 | 16 | 0 | 45.000000 | 0 |
| **1** | k403 | Ramicane | Male | 21 | 16 | 5 | 38.825898 | 0 |
| **2** | k403 | Ramicane | Male | 21 | 16 | 10 | 35.014271 | 1 |
| **3** | k403 | Ramicane | Male | 21 | 16 | 15 | 34.223992 | 1 |
| **4** | k403 | Ramicane | Male | 21 | 16 | 20 | 32.997729 | 1 |

# Checking the number of mice.

num\_mice = combined\_df["Mouse ID"].nunique()

num\_mice

249

**Remove Duplicate:**

\*Before beginning the analysis, check the data for any mouse ID with duplicate time points and remove any data associated with that mouse ID.

# Getting the duplicate mice by ID number that shows up for Mouse ID and Timepoint.

dup\_mice\_ID = combined\_df.loc[combined\_df.duplicated(subset=['Mouse ID', 'Timepoint']),'Mouse ID'].unique()

dup\_mice\_ID

array(['g989'], dtype=object)

# Optional: Get all the data for the duplicate mouse ID.

dup\_mice\_df = combined\_df.loc[combined\_df["Mouse ID"] == "g989", :]

dup\_mice\_df

|  | **Mouse ID** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Timepoint** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **908** | g989 | Propriva | Female | 21 | 26 | 0 | 45.000000 | 0 |
| **909** | g989 | Propriva | Female | 21 | 26 | 0 | 45.000000 | 0 |
| **910** | g989 | Propriva | Female | 21 | 26 | 5 | 48.786801 | 0 |
| **911** | g989 | Propriva | Female | 21 | 26 | 5 | 47.570392 | 0 |
| **912** | g989 | Propriva | Female | 21 | 26 | 10 | 51.745156 | 0 |
| **913** | g989 | Propriva | Female | 21 | 26 | 10 | 49.880528 | 0 |
| **914** | g989 | Propriva | Female | 21 | 26 | 15 | 51.325852 | 1 |
| **915** | g989 | Propriva | Female | 21 | 26 | 15 | 53.442020 | 0 |
| **916** | g989 | Propriva | Female | 21 | 26 | 20 | 55.326122 | 1 |
| **917** | g989 | Propriva | Female | 21 | 26 | 20 | 54.657650 | 1 |
| **918** | g989 | Propriva | Female | 21 | 26 | 25 | 56.045564 | 1 |
| **919** | g989 | Propriva | Female | 21 | 26 | 30 | 59.082294 | 1 |
| **920** | g989 | Propriva | Female | 21 | 26 | 35 | 62.570880 | 2 |

# Create a clean DataFrame by dropping the duplicate mouse by its ID.

clean\_df = combined\_df[combined\_df['Mouse ID'].isin(dup\_mice\_ID)==False]

clean\_df.head()

|  | **Mouse ID** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Timepoint** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | k403 | Ramicane | Male | 21 | 16 | 0 | 45.000000 | 0 |
| **1** | k403 | Ramicane | Male | 21 | 16 | 5 | 38.825898 | 0 |
| **2** | k403 | Ramicane | Male | 21 | 16 | 10 | 35.014271 | 1 |
| **3** | k403 | Ramicane | Male | 21 | 16 | 15 | 34.223992 | 1 |
| **4** | k403 | Ramicane | Male | 21 | 16 | 20 | 32.997729 | 1 |

# Checking the number of mice in the clean DataFrame.

clean\_mice = clean\_df["Mouse ID"].nunique()

clean\_mice

248

**Summary Statistics**

* Use the cleaned data for the remaining steps.
* Generate a summary statistics table consisting of the mean, median, variance, standard deviation, and SEM of the tumor volume for each drug regimen.

# Use groupby and summary statistical methods to calculate the following properties of each drug regimen:

# mean, median, variance, standard deviation, and SEM of the tumor volume.

mean = clean\_df['Tumor Volume (mm3)'].groupby(clean\_df['Drug Regimen']).mean()

median = clean\_df['Tumor Volume (mm3)'].groupby(clean\_df['Drug Regimen']).median()

var = clean\_df['Tumor Volume (mm3)'].groupby(clean\_df['Drug Regimen']).var()

std = clean\_df['Tumor Volume (mm3)'].groupby(clean\_df['Drug Regimen']).std()

sem = clean\_df['Tumor Volume (mm3)'].groupby(clean\_df['Drug Regimen']).sem()

summary\_stat = pd.DataFrame({"Mean Tumor Volume":mean,

"Median Tumor Volume":median,

"Tumor Volume Variance":var,

"Tumor Volume Std. Dev.":std,

"Tumor Volume Std. Err.":sem})

# Display the Summary statistics table grouped by 'Drug Regimen' column

summary\_stat

|  | **Mean Tumor Volume** | **Median Tumor Volume** | **Tumor Volume Variance** | **Tumor Volume Std. Dev.** | **Tumor Volume Std. Err.** |
| --- | --- | --- | --- | --- | --- |
| **Drug Regimen** |  |  |  |  |  |
| **Capomulin** | 40.675741 | 41.557809 | 24.947764 | 4.994774 | 0.329346 |
| **Ceftamin** | 52.591172 | 51.776157 | 39.290177 | 6.268188 | 0.469821 |
| **Infubinol** | 52.884795 | 51.820584 | 43.128684 | 6.567243 | 0.492236 |
| **Ketapril** | 55.235638 | 53.698743 | 68.553577 | 8.279709 | 0.603860 |
| **Naftisol** | 54.331565 | 52.509285 | 66.173479 | 8.134708 | 0.596466 |
| **Placebo** | 54.033581 | 52.288934 | 61.168083 | 7.821003 | 0.581331 |
| **Propriva** | 52.320930 | 50.446266 | 43.852013 | 6.622085 | 0.544332 |
| **Ramicane** | 40.216745 | 40.673236 | 23.486704 | 4.846308 | 0.320955 |
| **Stelasyn** | 54.233149 | 52.431737 | 59.450562 | 7.710419 | 0.573111 |
| **Zoniferol** | 53.236507 | 51.818479 | 48.533355 | 6.966589 | 0.516398 |

# Generate a summary statistics table of mean, median, variance, standard deviation, and SEM of the tumor volume for each regimen

#stat = ['mean', 'median', 'var', 'std', 'sem']

# Using the aggregation method, produce the same summary statistics in a single line

summary\_agg = clean\_df.groupby(['Drug Regimen'])[['Tumor Volume (mm3)']].agg(['mean', 'median', 'var', 'std', 'sem'])

summary\_agg

|  | **Tumor Volume (mm3)** | | | | |
| --- | --- | --- | --- | --- | --- |
|  | **mean** | **median** | **var** | **std** | **sem** |
| **Drug Regimen** |  |  |  |  |  |
| **Capomulin** | 40.675741 | 41.557809 | 24.947764 | 4.994774 | 0.329346 |
| **Ceftamin** | 52.591172 | 51.776157 | 39.290177 | 6.268188 | 0.469821 |
| **Infubinol** | 52.884795 | 51.820584 | 43.128684 | 6.567243 | 0.492236 |
| **Ketapril** | 55.235638 | 53.698743 | 68.553577 | 8.279709 | 0.603860 |
| **Naftisol** | 54.331565 | 52.509285 | 66.173479 | 8.134708 | 0.596466 |
| **Placebo** | 54.033581 | 52.288934 | 61.168083 | 7.821003 | 0.581331 |
| **Propriva** | 52.320930 | 50.446266 | 43.852013 | 6.622085 | 0.544332 |
| **Ramicane** | 40.216745 | 40.673236 | 23.486704 | 4.846308 | 0.320955 |
| **Stelasyn** | 54.233149 | 52.431737 | 59.450562 | 7.710419 | 0.573111 |
| **Zoniferol** | 53.236507 | 51.818479 | 48.533355 | 6.966589 | 0.516398 |

**Bar and Pie Charts**

\*Generate a bar plot using both Pandas's DataFrame.plot() and Matplotlib's pyplot that shows the number of total mice for each treatment regimen throughout the course of the study.

NOTE: These plots should look identical.

#first create a group series using the 'Drug Regimen' column and, counting the number of data points for each Drug Regimen

mice\_count = clean\_df["Drug Regimen"].value\_counts()

mice\_count

Capomulin 230

Ramicane 228

Ketapril 188

Naftisol 186

Zoniferol 182

Placebo 181

Stelasyn 181

Infubinol 178

Ceftamin 178

Propriva 148

Name: Drug Regimen, dtype: int64

# Create a Pandas bar plot based off of the group series from before and label the title

plot\_pandas = mice\_count.plot.bar(color='b')

# Set the xlabel, ylabel, and title using class methods

plt.xlabel("Drug Regimen")

plt.ylabel("Number of Mice")

plt.title("Number of Mice per Treatment")

Text(0.5, 1.0, 'Number of Mice per Treatment')

Icon

Description automatically generated

# Generate a bar plot showing the total number of unique mice tested on each drug regimen using pyplot.

x\_axis = mice\_count.index.values

y\_axis = mice\_count.values

# Create a Pyplot bar plot based off of the group series from before and label the title

plt.bar(x\_axis, y\_axis, color='b', alpha=0.8, align='center')

# Set the xlabel and ylabel, title using class methods

plt.title("Number of Mice Tested per Treatment")

plt.xlabel("Drug Regimen")

plt.ylabel("Number of Mice")

plt.xticks(rotation="vertical")

plt.show()

Icon

Description automatically generated with medium confidence

# Generate a pie plot showing the distribution of female versus male mice using pandas

gender\_data = clean\_df["Sex"].value\_counts()

plt.title("Female vs. Male Mice")

gender\_data.plot.pie(autopct= "%1.1f%%")

plt.show()

Chart, pie chart

Description automatically generated

# Generate a pie plot showing the distribution of female versus male mice using pyplot

labels = ['Female', 'Male']

sizes = [49.7999197, 50.200803]

plot = gender\_data.plot.pie(y='Total Count', autopct="%1.1f%%")

plt.title('Male vs Female Mouse Population')

plt.ylabel('Sex')

plt.show()

Chart, pie chart

Description automatically generated

**Quartiles, Outliers and Boxplots**

* Put treatments into a list (and later for plot labels)
* Create empty list to fill with tumor vol data (for plotting)
* Calculate the IQR and quantitatively determine if there are any potential outliers.
* Locate the rows which contain mice on each drug and get the tumor volumes add subset
* Determine outliers using upper and lower bounds

# Calculate the final tumor volume of each mouse across four of the treatment regimens:

# Capomulin, Ramicane, Infubinol, and Ceftamin

Capomulin\_df = clean\_df.loc[clean\_df["Drug Regimen"] == "Capomulin",:]

Ramicane\_df = clean\_df.loc[clean\_df["Drug Regimen"] == "Ramicane", :]

Infubinol\_df = clean\_df.loc[clean\_df["Drug Regimen"] == "Infubinol", :]

Ceftamin\_df = clean\_df.loc[clean\_df["Drug Regimen"] == "Ceftamin", :]

# Start by getting the last (greatest) timepoint for each mouse

# Capomulin

Capomulin\_last = Capomulin\_df.groupby('Mouse ID').max()['Timepoint']

Capomulin\_vol = pd.DataFrame(Capomulin\_last)

Capomulin\_merge = pd.merge(Capomulin\_vol, clean\_df, on=("Mouse ID","Timepoint"),how="left")

Capomulin\_merge.head()

|  | **Mouse ID** | **Timepoint** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | b128 | 45 | Capomulin | Female | 9 | 22 | 38.982878 | 2 |
| **1** | b742 | 45 | Capomulin | Male | 7 | 21 | 38.939633 | 0 |
| **2** | f966 | 20 | Capomulin | Male | 16 | 17 | 30.485985 | 0 |
| **3** | g288 | 45 | Capomulin | Male | 3 | 19 | 37.074024 | 1 |
| **4** | g316 | 45 | Capomulin | Female | 22 | 22 | 40.159220 | 2 |

Capomulin\_tumors = Capomulin\_merge["Tumor Volume (mm3)"]

quartiles =Capomulin\_tumors.quantile([.25,.5,.75])

lowerq = quartiles[0.25]

upperq = quartiles[0.75]

iqr = upperq-lowerq

print(f"The lower quartile of Capomulin tumors: {lowerq}")

print(f"The upper quartile of Capomulin tumors: {upperq}")

print(f"The interquartile range of Capomulin tumors: {iqr}")

print(f"The median of Capomulin tumors: {quartiles[0.5]} ")

lower\_bound = lowerq - (1.5\*iqr)

upper\_bound = upperq + (1.5\*iqr)

print(f"Values below {lower\_bound} could be outliers.")

print(f"Values above {upper\_bound} could be outliers.")

The lower quartile of Capomulin tumors: 32.37735684

The upper quartile of Capomulin tumors: 40.1592203

The interquartile range of Capomulin tumors: 7.781863460000004

The median of Capomulin tumors: 38.125164399999996

Values below 20.70456164999999 could be outliers.

Values above 51.83201549 could be outliers.

# Merge this group df with the original dataframe to get the tumor volume at the last timepoint

Ramicane\_last = Ramicane\_df.groupby('Mouse ID').max()['Timepoint']

Ramicane\_vol = pd.DataFrame(Ramicane\_last)

Ramicane\_merge = pd.merge(Ramicane\_vol, clean\_df, on=("Mouse ID","Timepoint"),how="left")

Ramicane\_merge.head()

Ramicane\_merge.to\_csv("output.csv")

Ramicane\_tumors = Ramicane\_merge["Tumor Volume (mm3)"]

quartiles =Ramicane\_tumors.quantile([.25,.5,.75])

lowerq = quartiles[0.25]

upperq = quartiles[0.75]

iqr = upperq-lowerq

print(f"The lower quartile of Ramicane tumors is: {lowerq}")

print(f"The upper quartile of Ramicane tumors is: {upperq}")

print(f"The interquartile range of Ramicane tumors is: {iqr}")

print(f"The median of Ramicane tumors is: {quartiles[0.5]} ")

lower\_bound = lowerq - (1.5\*iqr)

upper\_bound = upperq + (1.5\*iqr)

print(f"Values below {lower\_bound} could be outliers.")

print(f"Values above {upper\_bound} could be outliers.")

The lower quartile of Ramicane tumors is: 31.56046955

The upper quartile of Ramicane tumors is: 40.65900627

The interquartile range of Ramicane tumors is: 9.098536719999998

The median of Ramicane tumors is: 36.56165229

Values below 17.912664470000003 could be outliers.

Values above 54.30681135 could be outliers.

# Infubinol

# Determine outliers using upper and lower bounds

Infubinol\_last = Infubinol\_df.groupby('Mouse ID').max()['Timepoint']

Infubinol\_vol = pd.DataFrame(Infubinol\_last)

Infubinol\_merge = pd.merge(Infubinol\_vol, clean\_df, on=("Mouse ID","Timepoint"),how="left")

Infubinol\_merge.head()

|  | **Mouse ID** | **Timepoint** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | a203 | 45 | Infubinol | Female | 20 | 23 | 67.973419 | 2 |
| **1** | a251 | 45 | Infubinol | Female | 21 | 25 | 65.525743 | 1 |
| **2** | a577 | 30 | Infubinol | Female | 6 | 25 | 57.031862 | 2 |
| **3** | a685 | 45 | Infubinol | Male | 8 | 30 | 66.083066 | 3 |
| **4** | c139 | 45 | Infubinol | Male | 11 | 28 | 72.226731 | 2 |

Infubinol\_tumors = Infubinol\_merge["Tumor Volume (mm3)"]

quartiles =Infubinol\_tumors.quantile([.25,.5,.75])

lowerq = quartiles[0.25]

upperq = quartiles[0.75]

iqr = upperq-lowerq

print(f"The lower quartile of Infubinol tumors is: {lowerq}")

print(f"The upper quartile of Infubinol tumors is: {upperq}")

print(f"The interquartile range of Infubinol tumors is: {iqr}")

print(f"The median of Infubinol tumors is: {quartiles[0.5]} ")

lower\_bound = lowerq - (1.5\*iqr)

upper\_bound = upperq + (1.5\*iqr)

print(f"Values below {lower\_bound} could be outliers.")

print(f"Values above {upper\_bound} could be outliers.")

Infubinol\_merge.to\_csv("output.csv")

The lower quartile of Infubinol tumors is: 54.04860769

The upper quartile of Infubinol tumors is: 65.52574285

The interquartile range of Infubinol tumors is: 11.477135160000003

The median of Infubinol tumors is: 60.16518046

Values below 36.83290494999999 could be outliers.

Values above 82.74144559000001 could be outliers.

# Ceftamin

Ceftamin\_last = Ceftamin\_df.groupby('Mouse ID').max()['Timepoint']

Ceftamin\_vol = pd.DataFrame(Ceftamin\_last)

Ceftamin\_merge = pd.merge(Ceftamin\_vol, clean\_df, on=("Mouse ID","Timepoint"),how="left")

Ceftamin\_merge.head()

|  | **Mouse ID** | **Timepoint** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | a275 | 45 | Ceftamin | Female | 20 | 28 | 62.999356 | 3 |
| **1** | b447 | 0 | Ceftamin | Male | 2 | 30 | 45.000000 | 0 |
| **2** | b487 | 25 | Ceftamin | Female | 6 | 28 | 56.057749 | 1 |
| **3** | b759 | 30 | Ceftamin | Female | 12 | 25 | 55.742829 | 1 |
| **4** | f436 | 15 | Ceftamin | Female | 3 | 25 | 48.722078 | 2 |

# Ceftamin quartiles and IQR

# Determine outliers using upper and lower bounds

Ceftamin\_tumors = Ceftamin\_merge["Tumor Volume (mm3)"]

quartiles = Ceftamin\_tumors.quantile([.25,.5,.75])

lowerq = quartiles[0.25]

upperq = quartiles[0.75]

iqr = upperq-lowerq

print(f"The lower quartile of treatment is: {lowerq}")

print(f"The upper quartile of temperatures is: {upperq}")

print(f"The interquartile range of temperatures is: {iqr}")

print(f"The the median of temperatures is: {quartiles[0.5]} ")

# Determine outliers using upper and lower bounds

lower\_bound = lowerq - (1.5\*iqr)

upper\_bound = upperq + (1.5\*iqr)

print(f"Values below {lower\_bound} could be outliers.")

print(f"Values above {upper\_bound} could be outliers.")

The lower quartile of treatment is: 48.72207785

The upper quartile of temperatures is: 64.29983003

The interquartile range of temperatures is: 15.577752179999997

The the median of temperatures is: 59.85195552

Values below 25.355449580000002 could be outliers.

Values above 87.66645829999999 could be outliers.

# Generate a box plot of the final tumor volume of each mouse across four regimens of interest

data\_to\_plot = [Capomulin\_tumors, Ramicane\_tumors, Infubinol\_tumors, Ceftamin\_tumors]

Regimen = ['Capomulin', 'Ramicane', 'Infubinol','Ceftamin']

#data\_to\_plot=data\_to\_plot.values

fig1, ax1 = plt.subplots()

ax1.set\_title('Tumor Volume at Selected Mouse')

ax1.set\_ylabel('Final Tumor Volume (mm3)')

ax1.set\_xlabel('Drug Regimen')

ax1.boxplot(data\_to\_plot, labels=Regimen, widths = 0.4, vert=True)

plt.show()

Chart, box and whisker chart

Description automatically generated

**Recalculated: Quartiles, Outliers and Boxplots using for loop**

* Create empty list to fill with tumor vol data (for plotting)
* Calculate the IQR and quantitatively determine if there are any potential outliers.
* Locate the rows which contain mice on each drug and get the tumor volumes add subset
* Determine outliers using upper and lower bounds

# Calculate the final tumor volume of each mouse across four of the treatment regimens:

# Capomulin, Ramicane, Infubinol, and Ceftamin

# Merge this group df with the original dataframe to get the tumor volume at the last timepoint

drug\_list = ["Capomulin", "Ramicane", "Infubinol", "Ceftamin"]

drugs = combined\_df[combined\_df["Drug Regimen"].isin(drug\_list)]

drugs.head()

|  | **Mouse ID** | **Drug Regimen** | **Sex** | **Age\_months** | **Weight (g)** | **Timepoint** | **Tumor Volume (mm3)** | **Metastatic Sites** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | k403 | Ramicane | Male | 21 | 16 | 0 | 45.000000 | 0 |
| **1** | k403 | Ramicane | Male | 21 | 16 | 5 | 38.825898 | 0 |
| **2** | k403 | Ramicane | Male | 21 | 16 | 10 | 35.014271 | 1 |
| **3** | k403 | Ramicane | Male | 21 | 16 | 15 | 34.223992 | 1 |
| **4** | k403 | Ramicane | Male | 21 | 16 | 20 | 32.997729 | 1 |

# Start by getting the last (greatest) timepoint for each mouse

last\_timepoint = drugs.groupby(["Drug Regimen", "Mouse ID"]).agg(tumor\_size=("Tumor Volume (mm3)", lambda x: x.iloc[-1]))

#Reshape dataframe

last\_timepoint = last\_timepoint.stack(level=0).unstack(level=0)

for drug in drug\_list:

print(drug)

Capomulin

Ramicane

Infubinol

Ceftamin

# Put treatments into a list for for loop (and later for plot labels)

# Create empty list to fill with tumor vol data (for plotting)

# Calculate the IQR and quantitatively determine if there are any potential outliers.

# Locate the rows which contain mice on each drug and get the tumor volumes add subset

# Determine outliers using upper and lower bounds

treatment = 0

for drug in drug\_list:

quartiles = last\_timepoint[drug].quantile([.25,.5,.75]).round(2)

lowerq = quartiles[0.25].round(2)

upperq = quartiles[0.75].round(2)

iqr = round(upperq-lowerq,2)

lower\_bound = round(lowerq - (1.5\*iqr),2)

upper\_bound = round(upperq + (1.5\*iqr),2)

if treatment == 0:

print(f"------------------------------------------------------------")

print(f"The lower quartile of {drug} treatments is: {lowerq}")

print(f"The upper quartile of {drug} treatments is: {upperq}")

print(f"The interquartile range of {drug} treatments is: {iqr}")

print(f"Values below {lower\_bound} could be {drug} outliers.")

print(f"Values above {upper\_bound} could be {drug} outliers.")

print(f"------------------------------------------------------------")

treatment+=1

------------------------------------------------------------

The lower quartile of Capomulin treatments is: 32.38

The upper quartile of Capomulin treatments is: 40.16

The interquartile range of Capomulin treatments is: 7.78

Values below 20.71 could be Capomulin outliers.

Values above 51.83 could be Capomulin outliers.

------------------------------------------------------------

The lower quartile of Ramicane treatments is: 31.56

The upper quartile of Ramicane treatments is: 40.66

The interquartile range of Ramicane treatments is: 9.1

Values below 17.91 could be Ramicane outliers.

Values above 54.31 could be Ramicane outliers.

------------------------------------------------------------

The lower quartile of Infubinol treatments is: 54.05

The upper quartile of Infubinol treatments is: 65.53

The interquartile range of Infubinol treatments is: 11.48

Values below 36.83 could be Infubinol outliers.

Values above 82.75 could be Infubinol outliers.

------------------------------------------------------------

The lower quartile of Ceftamin treatments is: 48.72

The upper quartile of Ceftamin treatments is: 64.3

The interquartile range of Ceftamin treatments is: 15.58

Values below 25.35 could be Ceftamin outliers.

Values above 87.67 could be Ceftamin outliers.

------------------------------------------------------------

boxplot\_list = []

for drug in drug\_list:

boxplot\_list.append(list(last\_timepoint[drug].dropna()))

# Generate a box plot of the final tumor volume of each mouse across four regimens of interest

fig = plt.figure()

plt.xlabel("Regimen")

plt.xticks([1,2,3,4], drug\_list, rotation=45)

plt.ylabel("Tumor Volume")

plt.title("Tumor Volume by Drug Regimen")

plt.boxplot(boxplot\_list)

plt.show()

Chart, box and whisker chart

Description automatically generated

**Line and Scatter Plots**

* Calculate the final tumor volume of each mouse across four of the treatment regimens:
* Capomulin, Ramicane, Infubinol, and Ceftamin
* Start by getting the last (greatest) timepoint for each mouse
* Merge this group df with the original dataframe to get the tumor volume at the last timepoint
* Put treatments into a list for for loop (and later for plot labels)
* Create empty list to fill with tumor vol data (for plotting)
* Calculate the IQR and quantitatively determine if there are any potential outliers.

# Generate a line plot of tumor volume vs. time point for a mouse treated with Capomulin

forline\_df = Capomulin\_df.loc[Capomulin\_df["Mouse ID"] == "l509",:]

forline\_df.head()

x\_axis = forline\_df["Timepoint"]

tumsiz = forline\_df["Tumor Volume (mm3)"]

fig1, ax1 = plt.subplots()

plt.title('Capomulin treatmeant of mouse l509')

plt.plot(x\_axis, tumsiz,linewidth=2, markersize=15,marker="o",color="blue", label="Fahreneit")

plt.xlabel('Timepoint (Days)')

plt.ylabel('Tumor Volume (mm3)')

Text(0, 0.5, 'Tumor Volume (mm3)')

Chart, line chart

Description automatically generated

# Generate a scatter plot of average tumor volume vs. mouse weight for the Capomulin regimen

fig1, ax1 = plt.subplots()

avg\_capm\_vol =Capomulin\_df.groupby(['Mouse ID']).mean()

marker\_size=15

plt.scatter(avg\_capm\_vol['Weight (g)'],avg\_capm\_vol['Tumor Volume (mm3)'], color="blue")

plt.title('Mouse Weight Versus Average Tumor Volume')

plt.xlabel('Weight (g)',fontsize =14)

plt.ylabel('Averag Tumor Volume (mm3)')

Text(0, 0.5, 'Averag Tumor Volume (mm3)')

Chart, scatter chart

Description automatically generated

**Correlation and Regression**

# Calculate the correlation and regression model for mouse weight and average tumor volume for the Capomulin regimen

correlation = st.pearsonr(avg\_capm\_vol['Weight (g)'],avg\_capm\_vol['Tumor Volume (mm3)'])

print(f"The correlation between mouse weight and the average tumor volume is {round(correlation[0],2)}")

The correlation between mouse weight and the average tumor volume is 0.84

# Add the linear regression equation and line to plot

# Calculate the correlation coefficient and linear regression model for mouse weight and average tumor volume for the Capomulin regimen

(slope, intercept,rvalue, pvalue, stderr)= linregress(avg\_capm\_vol["Weight (g)"],avg\_capm\_vol["Tumor Volume (mm3)"])

regress\_values=avg\_capm\_vol["Weight (g)"]\* slope + intercept

line\_eq= f"y = {round(slope, 2)} x + {round(intercept, 2)}"

plt.scatter(avg\_capm\_vol["Weight (g)"],avg\_capm\_vol["Tumor Volume (mm3)"],color='b')

plt.plot(avg\_capm\_vol["Weight (g)"], regress\_values, color='red')

plt.annotate(line\_eq,(20,36), fontsize=14)

plt.xlabel("Weight (g)")

plt.ylabel("Tumor Volume (mm3)")

plt.title("Weight vs Tumor Volume for Capomulin")

print(f"The r-squared is: {round(rvalue\*\*2,3)}")

plt.show()

The r-squared is: 0.709

Chart, scatter chart

Description automatically generated