Matplotlib Project

November 19, 2021

0.1 # Pymaceuticals Inc.

0.1.1 Background

Pymaceuticals Inc. is a burgeoning feaux pharmaceutical company based out of San Diego. Pymaceuticals specializes in anti-cancer pharmaceuticals. In its most recent efforts, it began screening for potential treatments for squamous cell carcinoma (SCC), a commonly occurring form of skin cancer.

I have been given access to the complete data from their most recent animal study. In this study, 250 mice identified with SCC tumor growth were treated through a variety of drug regimens. Over the course of 45 days, tumor development was observed and measured. The purpose of this study was to compare the performance of Pymaceuticals' drug of interest, Capomulin, versus the other treatment regimens. I have been tasked by the executive team to generate all of the tables and figures needed for the technical report of the study. The executive team also has asked for a top-level summary of the study results.

[1]:

```
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as st

mouse_metadata_path = "Mouse_metadata.csv"
study_results_path = "Study_results.csv"

mouse_metadata = pd.read_csv(mouse_metadata_path)
study_results = pd.read_csv(study_results_path)

study_data_complete = pd.merge(study_results, mouse_metadata, how="left",␣
 ↪on="Mouse ID")

#pd.set_option("display.max_rows", None, "display.max_columns", None)
display(study_data_complete)
```

```
     Mouse ID  Timepoint  Tumor Volume (mm3)  Metastatic Sites Drug Regimen  \
0        b128          0           45.000000                 0    Capomulin
1        f932          0           45.000000                 0     Ketapril
2        g107          0           45.000000                 0     Ketapril
3        a457          0           45.000000                 0     Ketapril
4        c819          0           45.000000                 0     Ketapril
…         …         …                  …                …          …
```

```
1888     r944         45           41.581521                 2    Capomulin
1889     u364         45           31.023923                 3    Capomulin
1890     p438         45           61.433892                 1     Ceftamin
1891     x773         45           58.634971                 4      Placebo
1892     b879         45           72.555239                 2     Stelasyn

         Sex  Age_months  Weight (g)
0     Female           9          22
1       Male          15          29
2     Female           2          29
3     Female          11          30
4       Male          21          25
…      …         …         …
1888    Male          12          25
1889    Male          18          17
1890  Female          11          26
1891  Female          21          30
1892  Female           4          26

[1893 rows x 8 columns]
```

0.2 Summary Statistics

[2]:

```
means = study_data_complete.groupby('Drug Regimen').mean()['Tumor Volume (mm3)']
medians = study_data_complete.groupby('Drug Regimen').median()['Tumor Volume␣
 ↪(mm3)']
variances = study_data_complete.groupby('Drug Regimen').var()['Tumor Volume␣
 ↪(mm3)']
sds = study_data_complete.groupby('Drug Regimen').std()['Tumor Volume (mm3)']
sems = study_data_complete.groupby('Drug Regimen').sem()['Tumor Volume (mm3)']
summary_table = pd.DataFrame({"Mean Tumor Volume":means,
                              "Median Tumor Volume":medians,
                              "Tumor Volume Variance":variances,
                              "Tumor Volume Std. Dev.":sds,
                              "Tumor Volume Std. Err.":sems})
summary_table
```

[2]:

```
              Mean Tumor Volume  Median Tumor Volume  Tumor Volume Variance  \
Drug Regimen
Capomulin             40.675741            41.557809              24.947764
Ceftamin              52.591172            51.776157              39.290177
Infubinol             52.884795            51.820584              43.128684
Ketapril              55.235638            53.698743              68.553577
Naftisol              54.331565            52.509285              66.173479
Placebo               54.033581            52.288934              61.168083
Propriva              52.322552            50.854632              42.351070
Ramicane              40.216745            40.673236              23.486704
```

```
Stelasyn              54.233149           52.431737              59.450562
Zoniferol             53.236507           51.818479              48.533355

              Tumor Volume Std. Dev.  Tumor Volume Std. Err.
Drug Regimen
Capomulin                   4.994774                0.329346
Ceftamin                    6.268188                0.469821
Infubinol                   6.567243                0.492236
Ketapril                    8.279709                0.603860
Naftisol                    8.134708                0.596466
Placebo                     7.821003                0.581331
Propriva                    6.507770                0.512884
Ramicane                    4.846308                0.320955
Stelasyn                    7.710419                0.573111
Zoniferol                   6.966589                0.516398
```

[3]:

```python
# Alternate method
summary_table = study_data_complete.groupby("Drug Regimen").agg({"Tumor Volume␣
↪(mm3)":["mean","median","var","std","sem"]})
summary_table
```

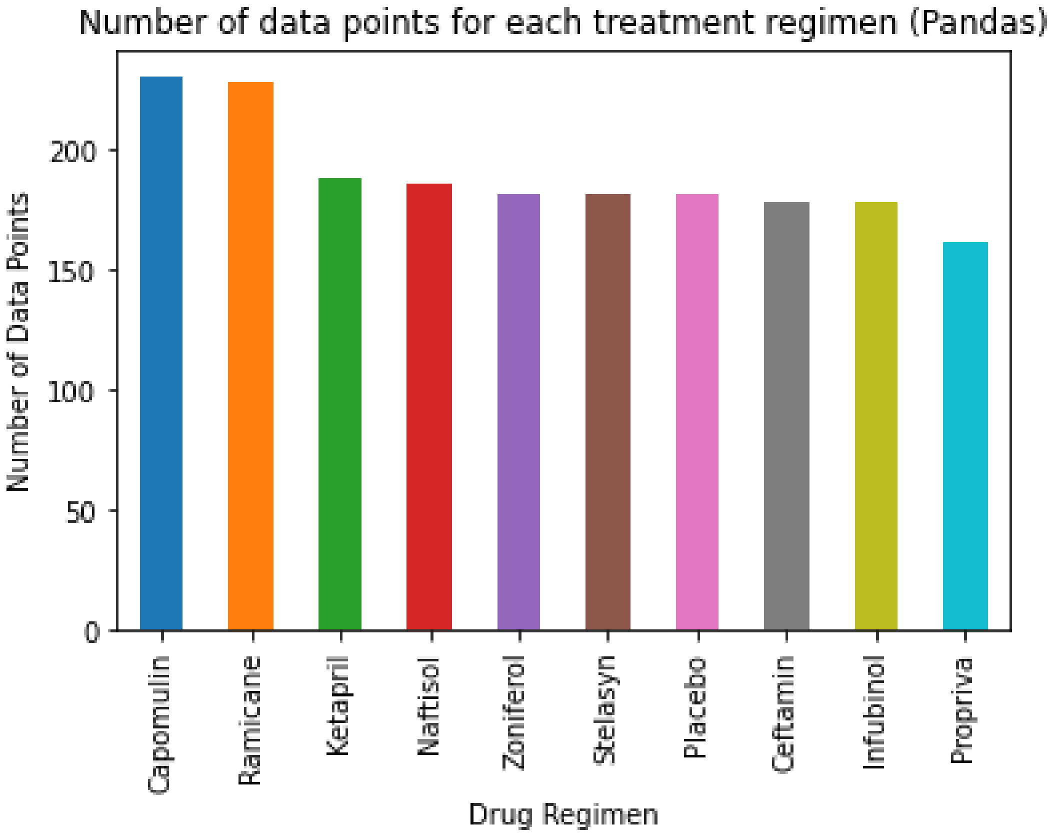
[3]:

```
             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              52.322552  50.854632  42.351070  6.507770  0.512884
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
```

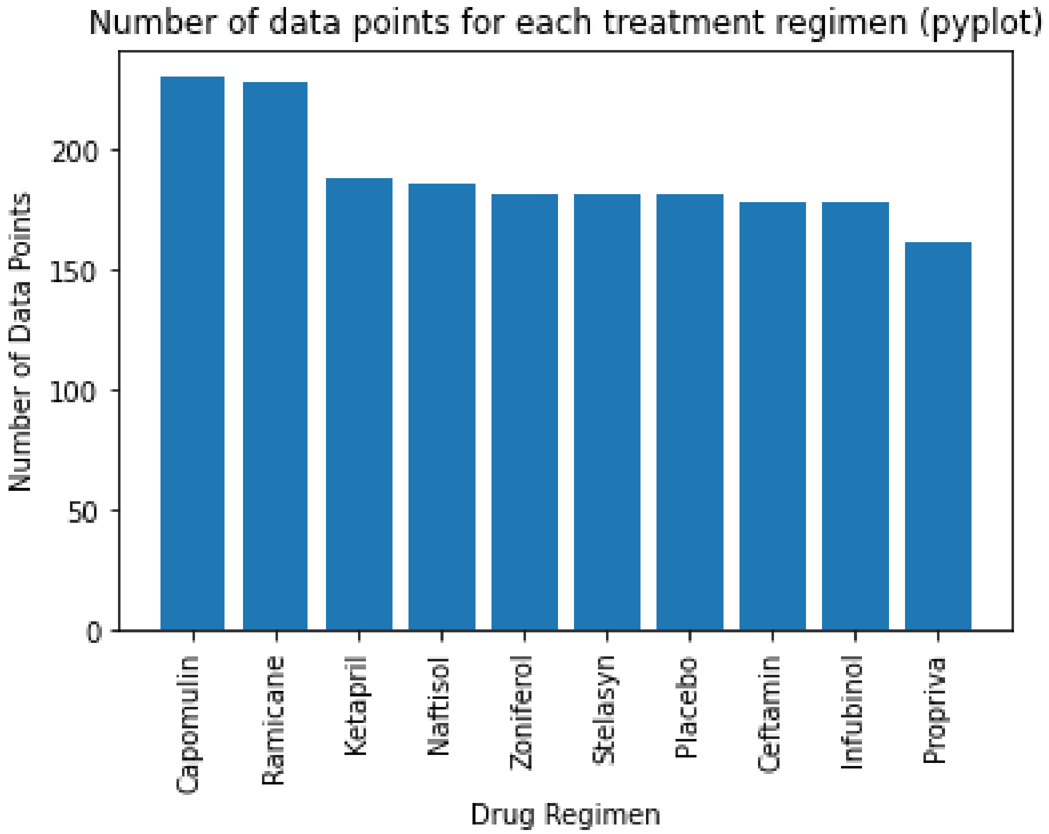
0.3 Bar and Pie Charts

[4]:

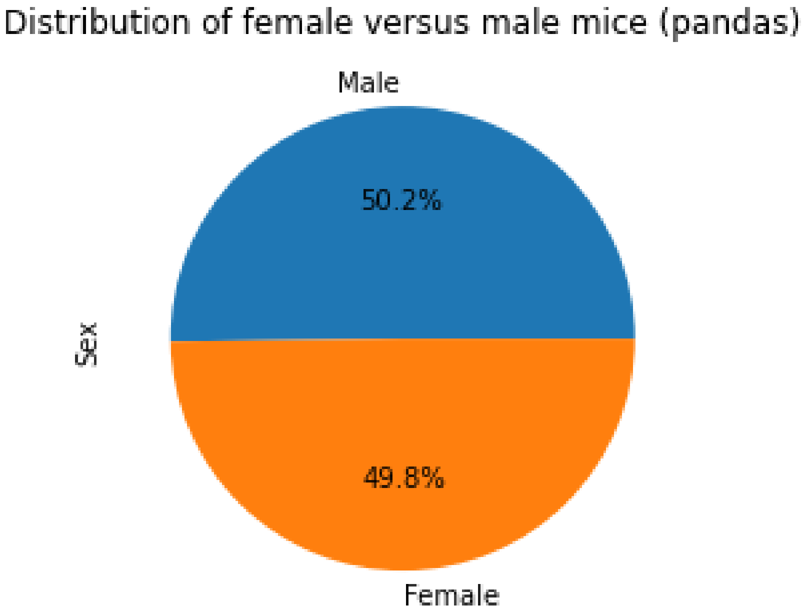
```python
counts = study_data_complete['Drug Regimen'].value_counts()
tableau_colors = ['tab:blue','tab:orange','tab:green','tab:red','tab:
↪purple','tab:brown','tab:pink','tab:gray','tab:olive','tab:cyan']
counts.plot(kind="bar", color=tableau_colors)
plt.title("Number of data points for each treatment regimen (Pandas)")
plt.xlabel("Drug Regimen")
plt.xticks(rotation=90)
plt.ylabel("Number of Data Points")
plt.show()
```



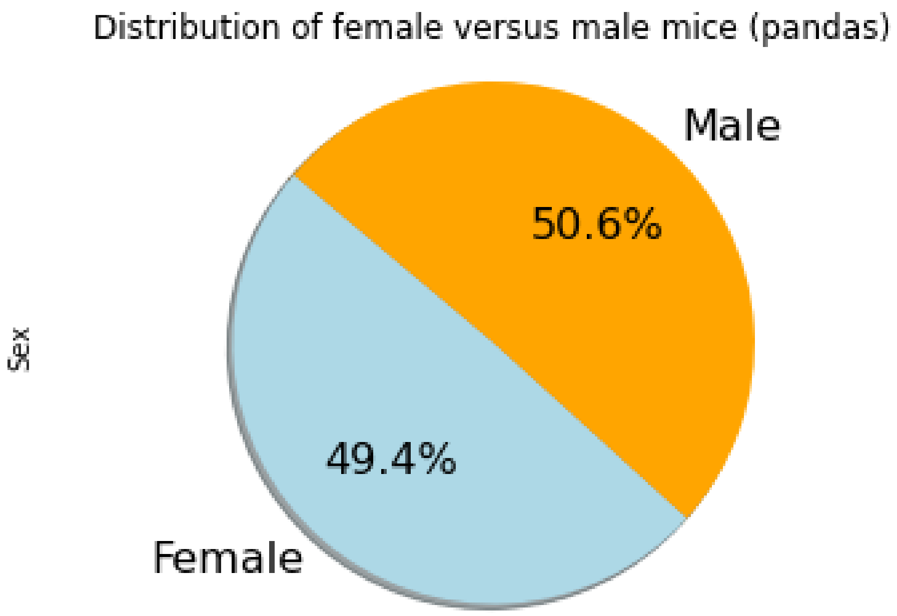
[5]:
```
counts = study_data_complete['Drug Regimen'].value_counts()
plt.bar(counts.index.values,counts.values)
plt.title("Number of data points for each treatment regimen (pyplot)")
plt.xlabel("Drug Regimen")
plt.xticks(rotation=90)
plt.ylabel("Number of Data Points")
plt.show()
```



[6]:
```
counts = mouse_metadata.Sex.value_counts()
counts.plot(kind="pie",autopct='%1.1f%%')
#plt.pie(counts.values,labels=counts.index.values,autopct='%1.1f%%')
plt.title("Distribution of female versus male mice (pandas)")
plt.show()
```

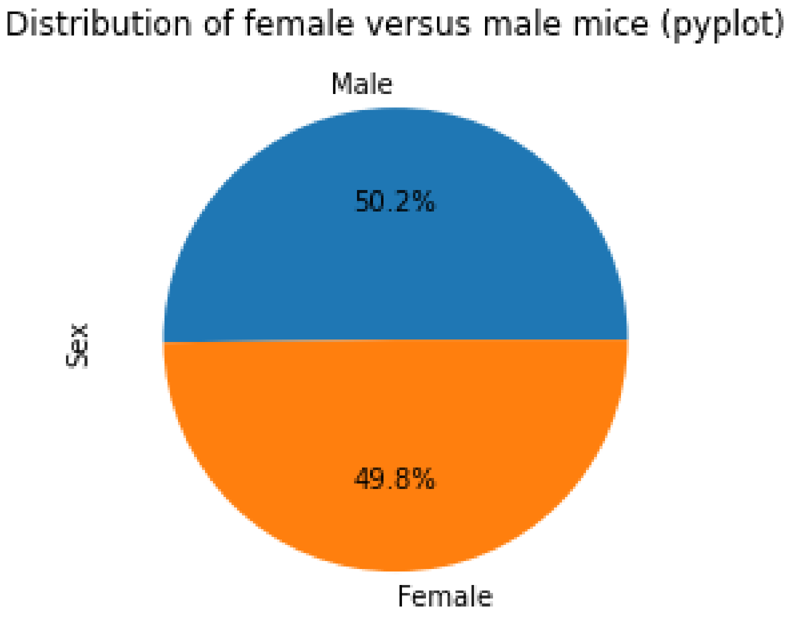


```
[7]: # Another way
gendergroup = study_data_complete.groupby('Sex')
gendercount = pd.DataFrame(gendergroup['Sex'].count())
gendercount.plot(kind='pie', y='Sex', title="Distribution of Female vs Male␣
 ↪Mice", startangle=140, autopct='%1.1f%%',shadow=True, fontsize=16, colors␣
 ↪=["lightblue","orange"],legend =False)
plt.axis("equal")
plt.title("Distribution of female versus male mice (pandas)")
plt.show()
```



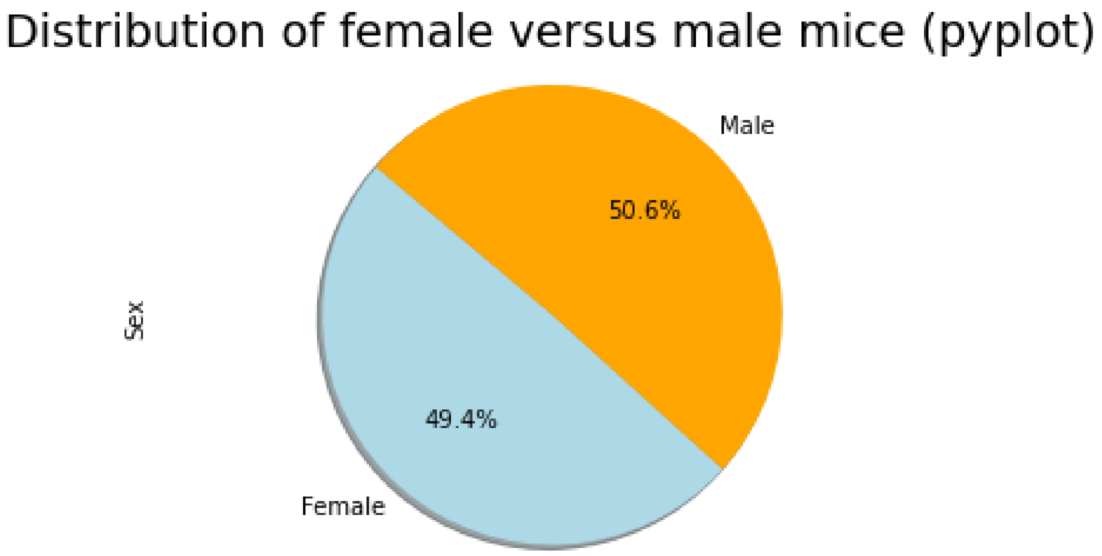
[8]:

```python
counts = mouse_metadata.Sex.value_counts()
plt.pie(counts.values,labels=counts.index.values,autopct='%1.1f%%')
plt.ylabel("Sex")
plt.title("Distribution of female versus male mice (pyplot)")
plt.show()
```



[9]:

```
# Another way
genders = list(gendercount.index.values)
gendercounts = gendercount['Sex']
colors = ["lightblue", "orange"]
plt.pie(gendercounts, labels=genders, colors=colors, autopct="%1.1f%%",␣
 ↪shadow=True, startangle=140)
plt.rcParams['font.size'] = 16
plt.title("Distribution of female versus male mice (pyplot)")
plt.ylabel("Sex")
plt.axis("equal")
plt.show()
```



0.4 Quartiles, Outliers and Boxplots

[10]:

```
# IQR and quantitative determination of any potential outliers

# Create a list of the four drugs to examine
druglist = ['Capomulin', 'Ramicane', 'Infubinol', 'Ceftamin']

# Slice the original combined_data dataframe using the list of four drugs
drugs = study_data_complete[study_data_complete['Drug Regimen'].isin(druglist)]

# Groupby 'Mouse ID'
finaltumor = drugs.groupby(['Drug Regimen','Mouse ID']).
 ↪agg(finaltumorsize=('Tumor Volume (mm3)',lambda x: x.iloc[-1])).round(3)
```

```python
# Reshape dataframe of the final tumor volume of each mouse across four of the most promising treatment regimens
finaltumornew = finaltumor.stack(level=0).unstack(level=0)

counter = 0
# Quartile calculations for each drug
for drug in druglist:
    quartiles = finaltumornew[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 counter == 0:
        print(f"--------------------------------------------------------")
    print(f"{drug} IQR data is:")
    print(f"The lower quartile of {drug} is: {lowerq}")
    print(f"The upper quartile of {drug} is: {upperq}")
    print(f"The interquartile range of {drug} is: {iqr}")
    print(f"The the median of {drug} is: {quartiles[0.5]} ")
    print(f"Values below {lower_bound} for {drug} could be outliers.")
    print(f"Values above {upper_bound} for {drug} could be outliers.")
    print(f"--------------------------------------------------------")
    counter += 1
```

```
--------------------------------------------------------
Capomulin IQR data is:
The lower quartile of Capomulin is: 32.38
The upper quartile of Capomulin is: 40.16
The interquartile range of Capomulin is: 7.78
The the median of Capomulin is: 38.12
Values below 20.71 for Capomulin could be outliers.
Values above 51.83 for Capomulin could be outliers.
--------------------------------------------------------
Ramicane IQR data is:
The lower quartile of Ramicane is: 31.56
The upper quartile of Ramicane is: 40.66
The interquartile range of Ramicane is: 9.1
The the median of Ramicane is: 36.56
Values below 17.91 for Ramicane could be outliers.
Values above 54.31 for Ramicane could be outliers.
--------------------------------------------------------
Infubinol IQR data is:
The lower quartile of Infubinol is: 54.05
The upper quartile of Infubinol is: 65.53
The interquartile range of Infubinol is: 11.48
The the median of Infubinol is: 60.16
Values below 36.83 for Infubinol could be outliers.
Values above 82.75 for Infubinol could be outliers.
--------------------------------------------------------
Ceftamin IQR data is:
The lower quartile of Ceftamin is: 48.72
The upper quartile of Ceftamin is: 64.3
The interquartile range of Ceftamin is: 15.58
The the median of Ceftamin is: 59.85
Values below 25.35 for Ceftamin could be outliers.
Values above 87.67 for Ceftamin could be outliers.
--------------------------------------------------------
```

[11]:

```
# Another way

# Start by getting the last (greatest) timepoint for each mouse
max_tumor = study_data_complete.groupby(["Mouse ID"]).max()
max_tumor = max_tumor.reset_index()

# Merge this group df with the original dataframe to get the tumor volume at␣
 ↪the last timepoint
merged_data = max_tumor[['Mouse ID','Timepoint']].
 ↪merge(study_data_complete,on=['Mouse ID','Timepoint'],how="left")

capomulin = merged_data.loc[merged_data["Drug Regimen"] == "Capomulin"]['Tumor␣
 ↪Volume (mm3)']
ramicane = merged_data.loc[merged_data["Drug Regimen"] == "Ramicane"]['Tumor␣
 ↪Volume (mm3)']
infubinol = merged_data.loc[merged_data["Drug Regimen"] == "Infubinol"]['Tumor␣
 ↪Volume (mm3)']
ceftamin = merged_data.loc[merged_data["Drug Regimen"] == "Ceftamin"]['Tumor␣
 ↪Volume (mm3)']
```

[12]:

```
# Quantitatively determine capomulin outliers
cap_quartiles = capomulin.quantile([.25,.5,.75])
cap_lowerq = cap_quartiles[0.25]
cap_upperq = cap_quartiles[0.75]
cap_iqr = cap_upperq-cap_lowerq
cap_lower_bound = cap_lowerq - (1.5*cap_iqr)
cap_upper_bound = cap_upperq + (1.5*cap_iqr)
print(f"Capomulin potential outliers: {capomulin.loc[(capomulin <␣
 ↪cap_lower_bound) | (capomulin > cap_upper_bound)]}")
```

```
Capomulin potential outliers: Series([], Name: Tumor Volume (mm3), dtype:
float64)
```

[13]:
```
# Quantitatively determine ramicane outliers
ram_quartiles = ramicane.quantile([.25,.5,.75])
ram_lowerq = ram_quartiles[0.25]
ram_upperq = ram_quartiles[0.75]
ram_iqr = ram_upperq-ram_lowerq
ram_lower_bound = ram_lowerq - (1.5*ram_iqr)
ram_upper_bound = ram_upperq + (1.5*ram_iqr)
print(f"Ramicane potential outliers: {ramicane.loc[(ramicane < ram_lower_bound)␣
 ↪| (ramicane > ram_upper_bound)]}")
```

```
Ramicane potential outliers: Series([], Name: Tumor Volume (mm3), dtype:
float64)
```

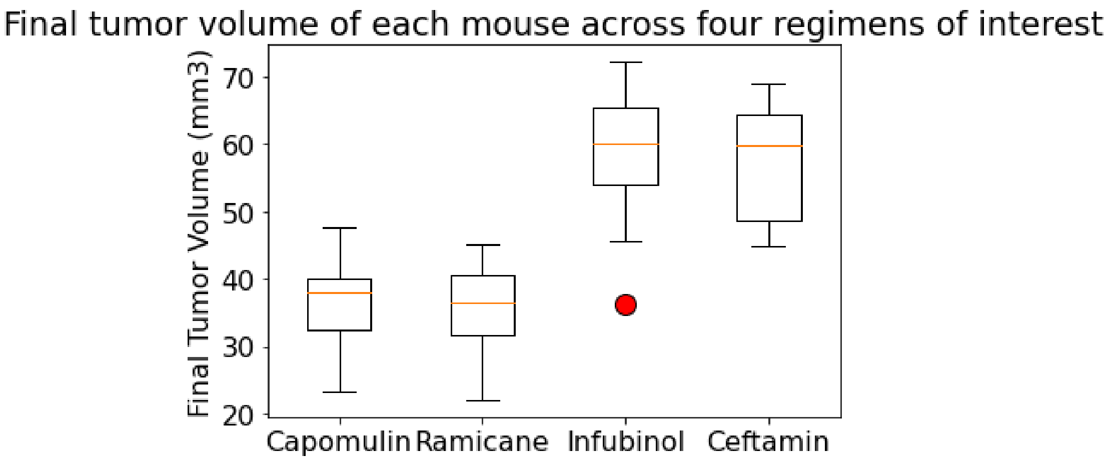
[14]:
```
# Quantitatively determine infubinol outliers
# The only one with outliers
inf_quartiles = infubinol.quantile([.25,.5,.75])
inf_lowerq = inf_quartiles[0.25]
inf_upperq = inf_quartiles[0.75]
inf_iqr = inf_upperq-inf_lowerq
inf_lower_bound = inf_lowerq - (1.5*inf_iqr)
inf_upper_bound = inf_upperq + (1.5*inf_iqr)
print(f"Infubinol potential outliers: {infubinol.loc[(infubinol <␣
 ↪inf_lower_bound) | (infubinol > inf_upper_bound)]}")
```

```
Infubinol potential outliers: 31    36.321346
Name: Tumor Volume (mm3), dtype: float64
```

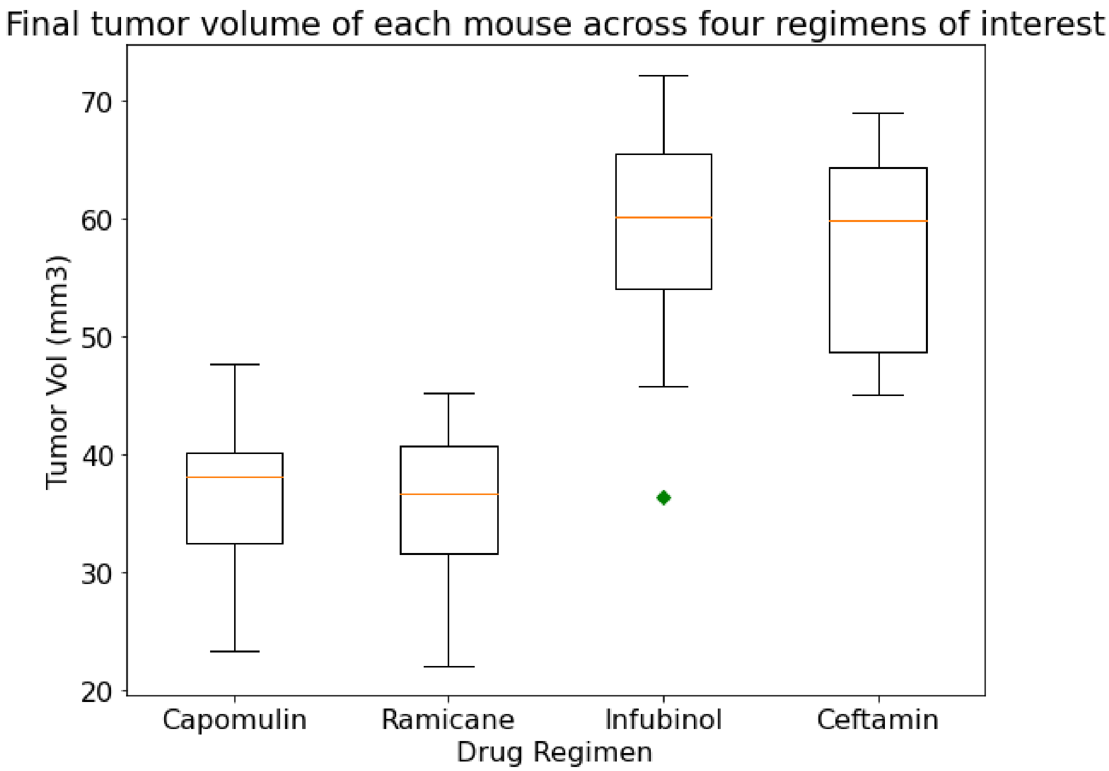
[15]:
```
# Quantitatively determine ceftamin outliers
cef_quartiles = ceftamin.quantile([.25,.5,.75])
cef_lowerq = cef_quartiles[0.25]
cef_upperq = cef_quartiles[0.75]
cef_iqr = cef_upperq-cef_lowerq
cef_lower_bound = cef_lowerq - (1.5*cef_iqr)
cef_upper_bound = cef_upperq + (1.5*cef_iqr)
print(f"Ceftamin potential outliers: {ceftamin.loc[(ceftamin < cef_lower_bound)␣
 ↪| (ceftamin > cef_upper_bound)]}")
```

```
Ceftamin potential outliers: Series([], Name: Tumor Volume (mm3), dtype:
float64)
```

[16]:
```
orange_out = dict(markerfacecolor='red',markersize=12)
plt.
 ↪boxplot([capomulin,ramicane,infubinol,ceftamin],labels=['Capomulin','Ramicane','Infubinol',
plt.ylabel('Final Tumor Volume (mm3)')
plt.title('Final tumor volume of each mouse across four regimens of interest')
plt.show()
```



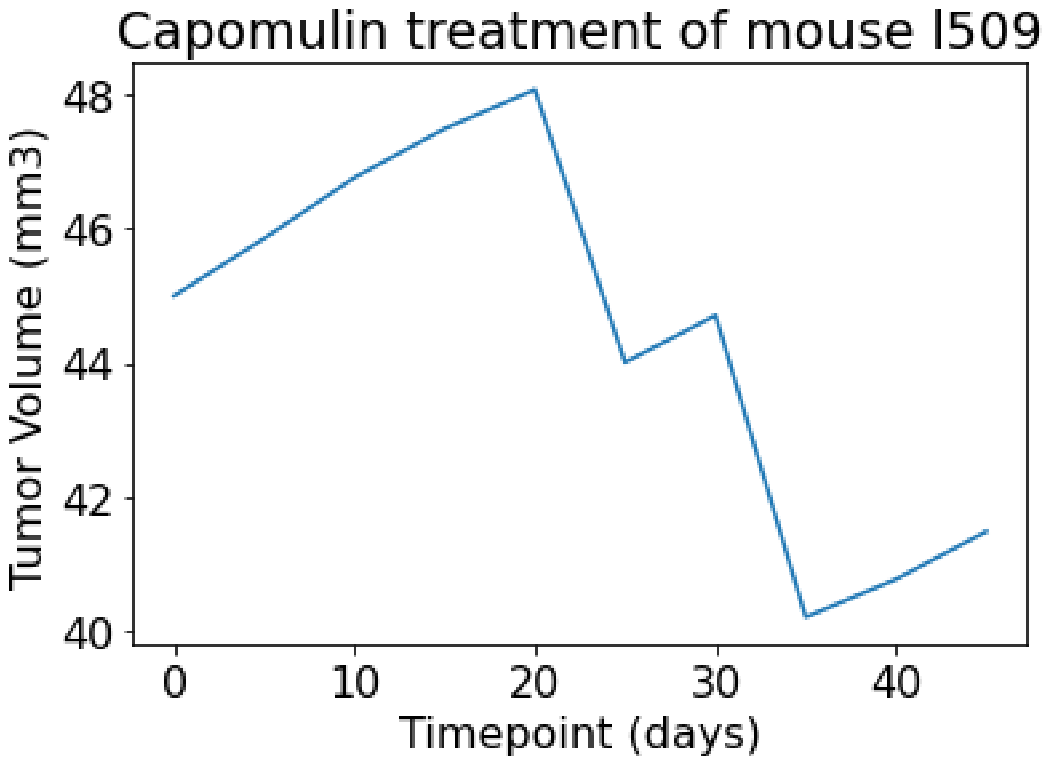
[17]:
```
# Another way
boxplotlist = []

for drug in druglist:
    boxplotlist.append(list(finaltumornew[drug].dropna()))

fig, ax = plt.subplots(figsize=(9,7))
ax.set_title('Final tumor volume of each mouse across four regimens of␣
 ↪interest')
ax.set_xlabel('Drug Regimen')
ax.set_ylabel('Tumor Vol (mm3)')
ax.boxplot(boxplotlist,notch=0,sym='gD')
plt.xticks([1,2,3,4],druglist)
plt.show()
```



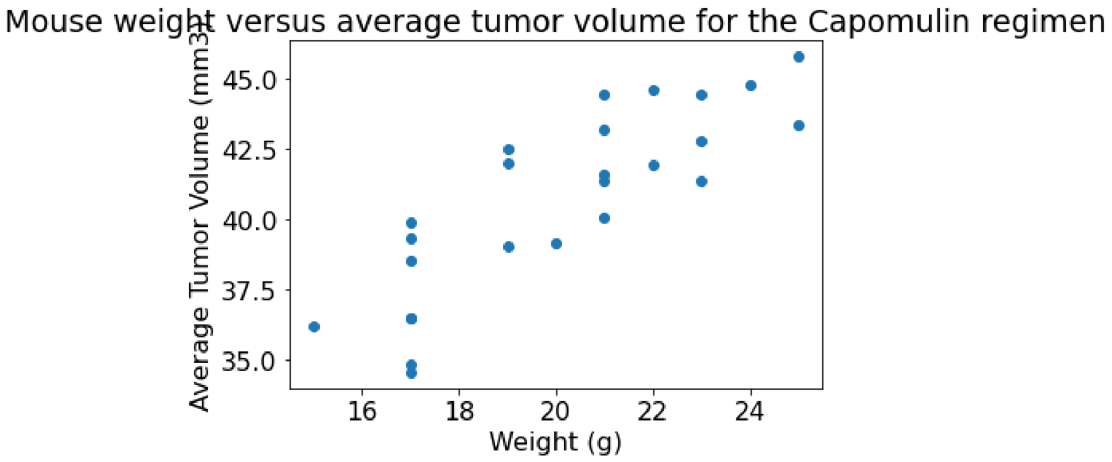
0.5 Line and Scatter Plots

[18]:

```
capomulin_table = study_data_complete.loc[study_data_complete['Drug Regimen'] == "Capomulin"]
mousedata = capomulin_table.loc[capomulin_table['Mouse ID']== 'l509']
plt.plot(mousedata['Timepoint'],mousedata['Tumor Volume (mm3)'])
plt.xlabel('Timepoint (days)')
plt.ylabel('Tumor Volume (mm3)')
plt.title('Capomulin treatment of mouse l509')
plt.show()
```



```
[19]: capomulin_average = capomulin_table.groupby(['Mouse ID']).mean()
      plt.title('Mouse weight versus average tumor volume for the Capomulin regimen')
      plt.scatter(capomulin_average['Weight (g)'],capomulin_average['Tumor Volume␣
       ↪(mm3)'])
      plt.xlabel('Weight (g)')
      plt.ylabel('Average Tumor Volume (mm3)')
      plt.show()
```

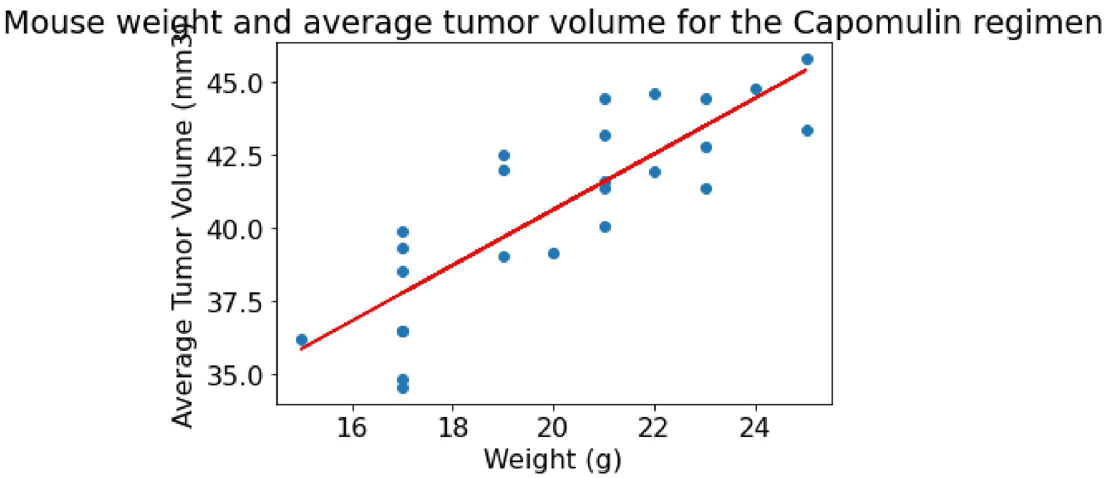


0.6 Correlation and Regression

[20]:

```
corr=round(st.pearsonr(capomulin_average['Weight (g)'],capomulin_average['Tumor␣
 ↪Volume (mm3)'])[0],2)
print(f"The correlation between mouse weight and the average tumor volume is␣
 ↪{corr}")
model = st.linregress(capomulin_average['Weight (g)'],capomulin_average['Tumor␣
 ↪Volume (mm3)'])
y_values = capomulin_average['Weight (g)']*model[0]+model[1]
plt.scatter(capomulin_average['Weight (g)'],capomulin_average['Tumor Volume␣
 ↪(mm3)'])
plt.plot(capomulin_average['Weight (g)'],y_values,color="red")
plt.title('Mouse weight and average tumor volume for the Capomulin regimen')
plt.xlabel('Weight (g)')
plt.ylabel('Average Tumor Volume (mm3)')
plt.show()
```

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

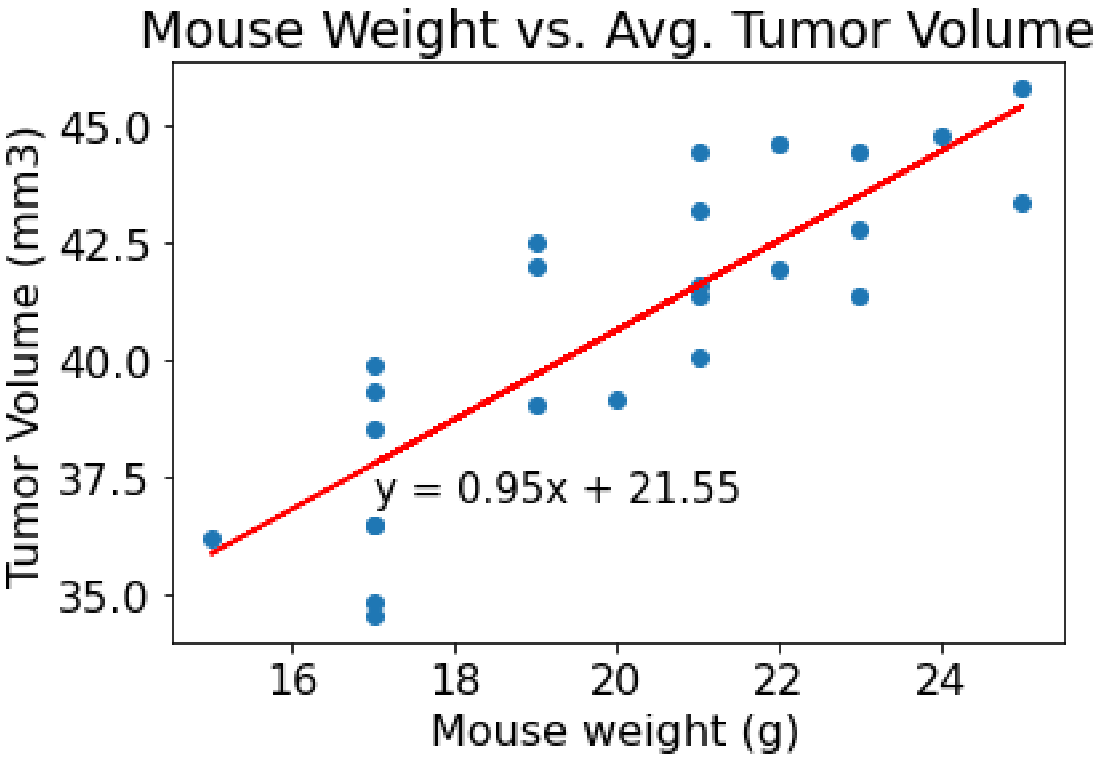


[21]:

```
# Another way
# Groupby Mouse ID and get weight and mean of tumor volume
capmouse = capomulin_table.groupby(['Mouse ID']).agg(MouseWeight=('Weight (g)',␣
 ↪np.mean), TumorMean=('Tumor Volume (mm3)', np.mean)).round(3)
correlation = st.pearsonr(capmouse['MouseWeight'],capmouse['TumorMean'])
print(f"The correlation between both factors is {round(correlation[0],2)}")

# Print out the r-squared value
from scipy.stats import linregress
x_values = capmouse['MouseWeight']
```

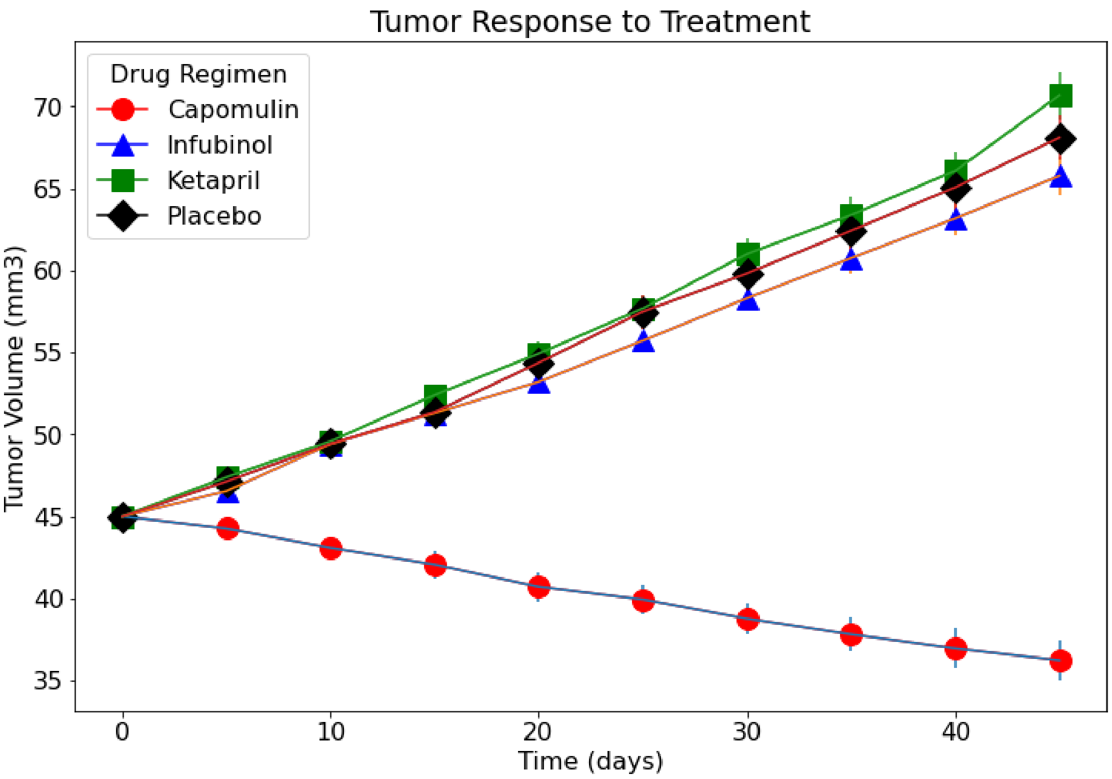
```
y_values = capmouse['TumorMean']
(slope, intercept, rvalue, pvalue, stderr) = linregress(x_values, y_values)
regress_values = x_values * slope + intercept
line_eq = f'y = {str(round(slope,2))}x + {str(round(intercept,2))}'

plt.scatter(x_values,y_values)
plt.plot(x_values,regress_values,"r-")
plt.annotate(line_eq,(17,37),fontsize=15,color="black")
plt.title("Mouse Weight vs. Avg. Tumor Volume")
plt.xlabel("Mouse weight (g)")
plt.ylabel("Tumor Volume (mm3)")

print(f"The r-squared is: {rvalue}")
print(f"The equation of the regression line is: {line_eq}")

plt.show()
```

```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

The correlation between both factors is 0.84
The r-squared is: 0.8419461020261079
The equation of the regression line is: y = 0.95x + 21.55
```



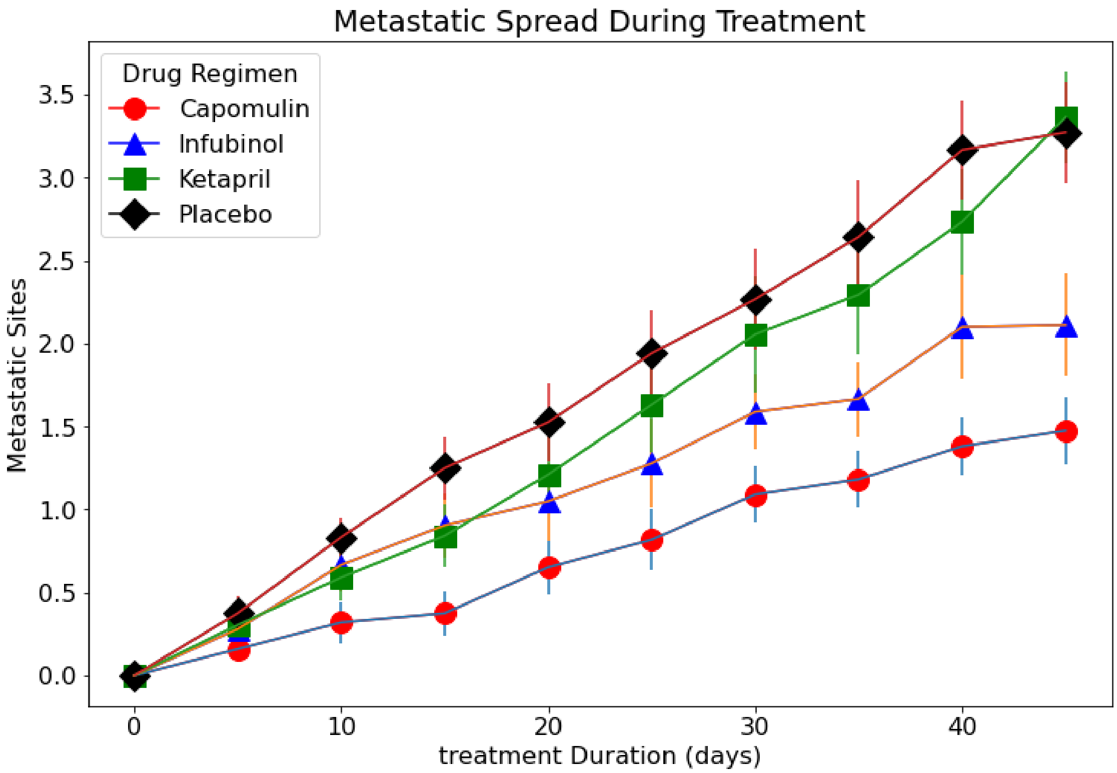
[22]:

```
druglist2 = ['Capomulin', 'Infubinol', 'Ketapril', 'Placebo']
drugs2 = study_data_complete[study_data_complete['Drug Regimen'].
 ↪isin(druglist2)]
tumor_means = drugs2.groupby(['Timepoint','Drug Regimen'],␣
 ↪as_index=False)['Tumor Volume (mm3)'].mean()
tumor_means = tumor_means.pivot(index='Timepoint', columns='Drug Regimen',␣
 ↪values='Tumor Volume (mm3)')

tumor_errs = drugs2.groupby(['Timepoint','Drug Regimen'],␣
 ↪as_index=False)['Tumor Volume (mm3)'].sem()
tumor_errs = tumor_errs.pivot(index='Timepoint', columns='Drug Regimen',␣
 ↪values='Tumor Volume (mm3)')
ax = tumor_means.plot(figsize=(12,8), yerr = tumor_errs, legend = False)
ax.set_prop_cycle(None)

tumor_means.plot(figsize=(12,8), style=['ro-', 'b^-', 'gs-', 'kD-'],␣
 ↪markersize=14, ax = ax)
plt.title('Tumor Response to Treatment')
plt.xlabel('Time (days)')
plt.ylabel('Tumor Volume (mm3)')
plt.show()
```



[23]:

```
met_means = drugs2.groupby(['Timepoint','Drug Regimen'],␣
 ↪as_index=False)['Metastatic Sites'].mean()
met_means = met_means.pivot(index='Timepoint', columns='Drug Regimen',␣
 ↪values='Metastatic Sites')

met_errs = drugs2.groupby(['Timepoint','Drug Regimen'],␣
 ↪as_index=False)['Metastatic Sites'].sem()
met_errs = met_errs.pivot(index='Timepoint', columns='Drug Regimen',␣
 ↪values='Metastatic Sites')
ax = met_means.plot(figsize=(12,8), yerr = met_errs, legend = False)
ax.set_prop_cycle(None)

met_means.plot(figsize=(12,8), style=['ro-', 'b^-', 'gs-', 'kD-'],␣
 ↪markersize=14, ax = ax)
plt.title('Metastatic Spread During Treatment')
plt.xlabel('treatment Duration (days)')
plt.ylabel('Metastatic Sites')
plt.show()
```



[24]:

```
# Dropping duplicate mice
drop_dup_mouse_id = study_results.loc[study_results.duplicated(subset=['Mouse␣
 ↪ID', 'Timepoint',]),'Mouse ID'].unique()
```

```
clean_clinical_trial_df = study_results[study_results['Mouse ID'].
 ↪isin(drop_dup_mouse_id)==False]
clean_mouse_df = mouse_metadata[mouse_metadata['Mouse ID'].
 ↪isin(drop_dup_mouse_id)==False]
combined_data = pd.merge(clean_clinical_trial_df, clean_mouse_df, on = "Mouse␣
 ↪ID")

# Sorting by Timepoint
sort_by_time = combined_data.sort_values("Timepoint", ascending= True)
all_sort_by_time = sort_by_time.reset_index(drop=True)
all_sort_by_time.head()
```

[24]:

```
  Mouse ID  Timepoint  Tumor Volume (mm3)  Metastatic Sites Drug Regimen  \
0     b128          0                45.0                 0    Capomulin
1     v409          0                45.0                 0      Placebo
2     u946          0                45.0                 0     Propriva
3     w140          0                45.0                 0    Zoniferol
4     a577          0                45.0                 0    Infubinol

      Sex  Age_months  Weight (g)
0  Female           9          22
1  Female          16          25
2    Male           5          30
3  Female          19          30
4  Female           6          25
```

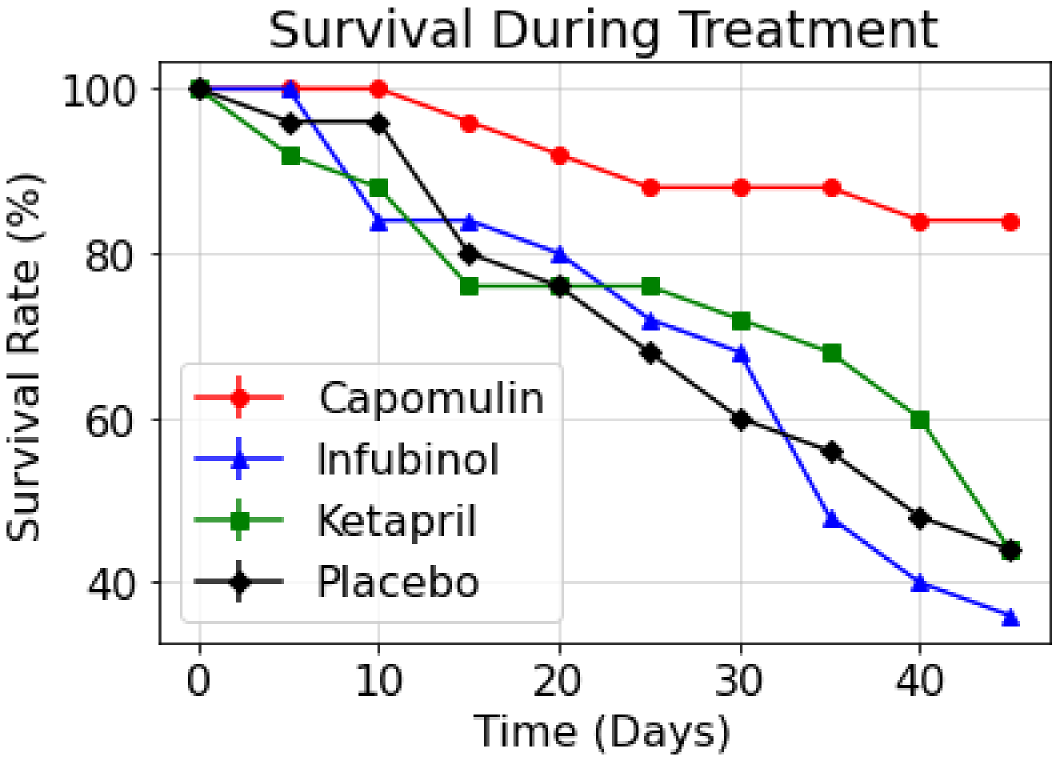
[25]:

```
metastatic_response = all_sort_by_time.drop('Tumor Volume (mm3)', axis = 1)
metastatic_sem = metastatic_response.pivot_table(metastatic_response, index =␣
 ↪['Drug Regimen','Timepoint',], aggfunc='sem')
metastatic_sem_table = metastatic_sem.pivot_table('Metastatic Sites',␣
 ↪['Timepoint'],'Drug Regimen')

mouse_survive = metastatic_response.drop('Metastatic Sites', axis = 1)

mouse_survival = mouse_survive.pivot_table(mouse_survive, index=['Drug␣
 ↪Regimen','Timepoint'], aggfunc='count')
mouse_survival_rename = mouse_survival.rename(columns={"Mouse ID":"Mouse␣
 ↪Count"})

mouse_survival_tbl = mouse_survival_rename.pivot_table('Mouse␣
 ↪Count',['Timepoint'],'Drug Regimen')

# Survival rate
percent_surviving = (1-(mouse_survival_tbl.iloc[0]- mouse_survival_tbl)/
 ↪mouse_survival_tbl.iloc[0])*100
```

```
x_axis = np.arange(0,50,5)
count = np.arange(0,len(druglist2))
colors = ['red','blue','green','black']
markers = ['o', '^', 's', 'D']

plt.title("Survival During Treatment")
plt.xlabel("Time (Days)")
plt.ylabel("Survival Rate (%)")
plt.grid(alpha = 0.5)

from scipy import stats
for i in count:
    graph_data = stats.sem(metastatic_sem_table[druglist2[i]])
    plt.errorbar(x_axis, percent_surviving[druglist2[i]], yerr = graph_data,␣
 ↪marker= markers[i], color= colors[i], label = druglist2[i])
plt.legend(loc='best')
plt.show()
```

```
<IPython.core.display.Javascript object>
```

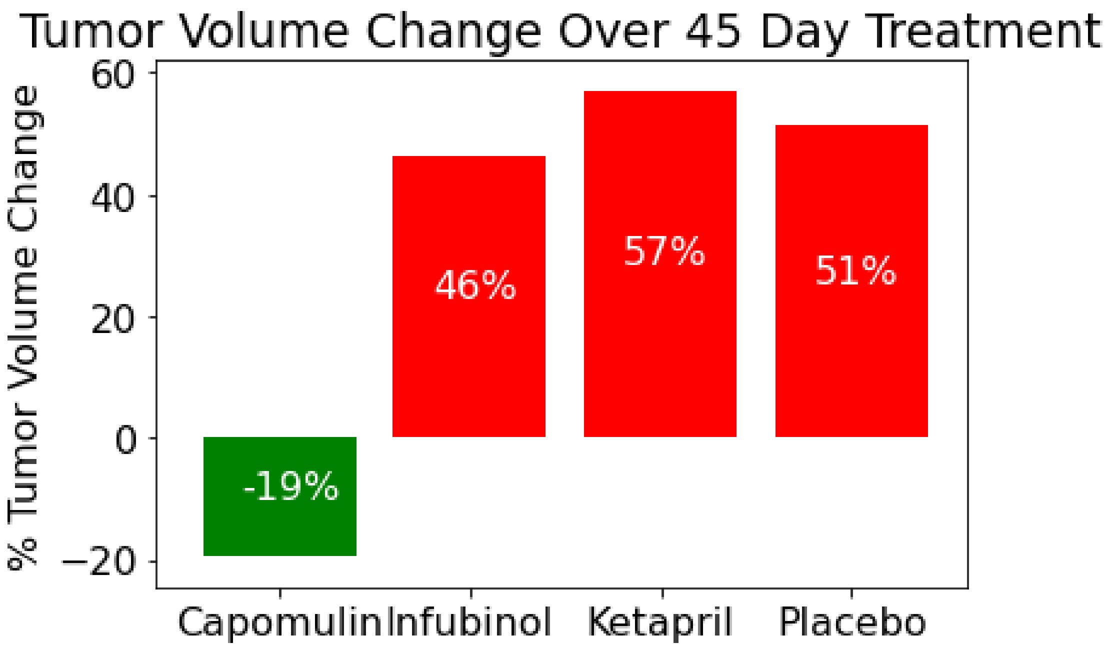
```
<IPython.core.display.Javascript object>
```



[26]:

```
tumor_volume = all_sort_by_time.drop('Metastatic Sites', axis=1)

drug_groups = tumor_volume.pivot_table(tumor_volume, index=['Drug␣
 ↪Regimen','Timepoint'], aggfunc='mean')

sem_tumor_volume = tumor_volume.pivot_table(tumor_volume, index=['Drug␣
 ↪Regimen','Timepoint'], aggfunc='sem')

sem_table = sem_tumor_volume.pivot_table('Tumor Volume (mm3)', ['Timepoint'],␣
 ↪'Drug Regimen')
tumor_vol_table = drug_groups.pivot_table('Tumor Volume (mm3)', ['Timepoint'],␣
 ↪'Drug Regimen')

# Tumor volume change
summ_tumor_vol = tumor_vol_table.iloc[[0,-1]]
percent_change_tumor_vol= (((summ_tumor_vol -tumor_vol_table.iloc[0])/
 ↪tumor_vol_table.iloc[0]))*100
percent_changes = percent_change_tumor_vol.iloc[1:]
percent_changes.sum()

# Bar graph indicating tumor growth as red and tumor reduction as green
performance = {}
for x in count:
    performance[druglist2[x]] = float(percent_changes[druglist2[x]])
x_axis = np.arange(0, len(druglist2))
tick_locations = []
for x in x_axis:
    tick_locations.append(x + 0.4)
plt.xlim(-0.25, len(druglist2))
plt.ylim(min(performance.values()) - 5, max(performance.values()) + 5)
plt.title("Tumor Volume Change Over 45 Day Treatment")
plt.ylabel("% Tumor Volume Change")
bar_colors = pd.Series(list(performance.values()))
bar_colors = bar_colors > 0
bar_colors = bar_colors.map({True: "Red", False: "Green"})
plt.xticks(tick_locations, performance)
plt.bar(x_axis, performance.values(), color=bar_colors, align="edge")

for index,data in enumerate(list(performance.values())):
    plt.text(x = index+0.2, y = data/2, s = str(int(data))+'%', color = 'white')

plt.show()
```

```
<IPython.core.display.Javascript object>
```



0.6.1 Analysis

- Overall, it is clear that Capomulin is a viable drug regimen to reduce tumor growth.
- Capomulin had the most number of mice complete the study, with the exception of Remicane, all other regimens observed a number of mice deaths across the duration of the study.
- There is a strong correlation between mouse weight and tumor volume, indicating that mouse weight may be contributing to the effectiveness of any drug regimen. I.e. Mouse weight correlated strongly (R-squared of 0.84) with average tumor volume so one would want to factor in mouse weight whenever considering how effective a drug was in reducing a tumor.
- There was one potential outlier within the Infubinol regimen. While most mice showed tumor volume increase, there was one mouse that had a reduction in tumor growth in the study.
- When sample sizes are 30 or larger then the data can be seen as statistically significant and as shown in the bar plot, for each Drug Regimen there were at least 100 data points. Therefore, the next two observations are said to be statistically significant.
- The ratio of male to female mice is almost identical. Further analysis into specific sexes would be interesting.