Pymaceuticals Inc - Option # 2

In [1]:
```
#Created on Sat Aug 25 20:33:57 2018
#@author: anthonyalvarez
#Test Files: Pharm_00.ipynb - Pharm_05.ipynb
```

Referenced Material

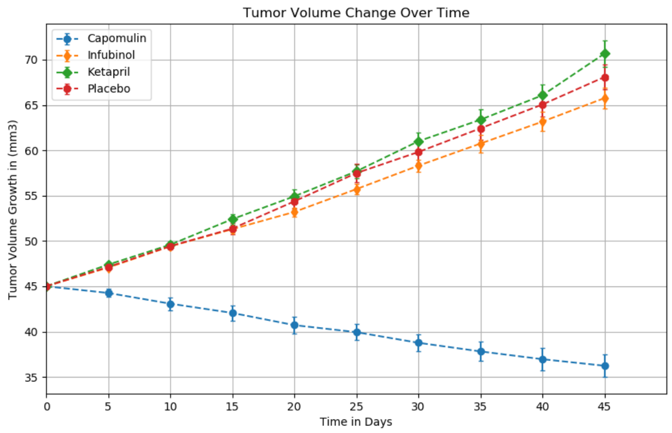
- https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_csv.html (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_csv.html)
- https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.reset_index.html (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.reset_index.html)
- https://stackoverflow.com/questions/31235745/python-querying-tables-from-database-and-having-conditions-in-data-frame (https://stackoverflow.com/questions/31235745/python-querying-tables-from-database-and-having-conditions-in-data-frame)
- https://pythonforbiologists.com/when-to-use-aggregatefiltertransform-in-pandas/ (https://pythonforbiologists.com/when-to-use-aggregatefiltertransform-in-pandas/)
- https://stackoverflow.com/questions/14657241/how-do-i-get-a-list-of-all-the-duplicate-items-using-pandas-in-python (https://stackoverflow.com/questions/14657241/how-do-i-get-a-list-of-all-the-duplicate-items-using-pandas-in-python)
- https://stackoverflow.com/questions/17141558/how-to-sort-a-dataframe-in-python-pandas-by-two-or-more-columns (https://stackoverflow.com/questions/17141558/how-to-sort-a-dataframe-in-python-pandas-by-two-or-more-columns)
- https://stackoverflow.com/questions/11869910/pandas-filter-rows-of-dataframe-with-operator-chaining (https://stackoverflow.com/questions/11869910/pandas-filter-rows-of-dataframe-with-operator-chaining)
- https://stackoverflow.com/questions/22086116/how-do-you-filter-pandas-dataframes-by-multiple-columns (https://stackoverflow.com/questions/22086116/how-do-you-filter-pandas-dataframes-by-multiple-columns)
- https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.unstack.html (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.unstack.html)
- https://stackoverflow.com/questions/44890713/selection-with-loc-in-python (https://stackoverflow.com/questions/44890713/selection-with-loc-in-python)
- https://stackoverflow.com/questions/17241004/pandas-how-to-get-the-data-frame-index-as-an-array (https://stackoverflow.com/questions/17241004/pandas-how-to-get-the-data-frame-index-as-an-array)
- https://stackoverflow.com/questions/39038358/function-chaining-in-python (https://stackoverflow.com/questions/39038358/function-chaining-in-python)
- https://stackoverflow.com/questions/4406389/if-else-in-a-list-comprehension (https://stackoverflow.com/questions/4406389/if-else-in-a-list-comprehension)
- https://en.wikipedia.org/wiki/Mortality_rate (https://en.wikipedia.org/wiki/Mortality_rate)
- https://en.wikipedia.org/wiki/Metastasis (https://en.wikipedia.org/wiki/Metastasis)
- https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet#links (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet#links)

Data Sources

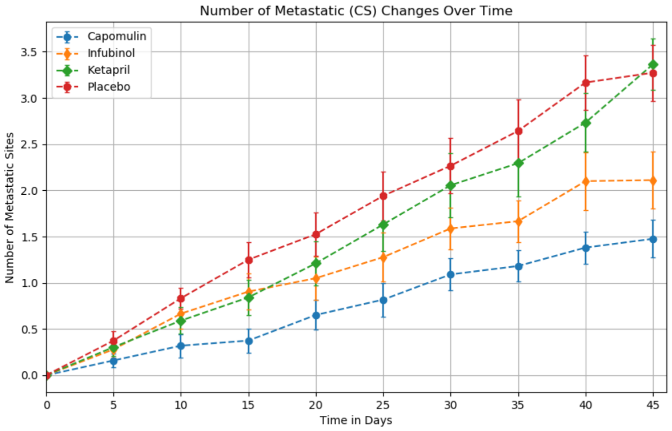
- Mouse Data (data/clinicaltrial_data.csv) Clinical Data including Time, Size and Metastatic Sites
- Clinical Data (data/mouse_drug_data.csv) Mouse ID and Type of Drug Given
- Merged Data (data/merged_clinical_mouse.csv) Combined Data from the 2 initial data sets
- Clean Clinical Data (data/clean_clinical_mouse) Data Source used for this project

Data Visualizations

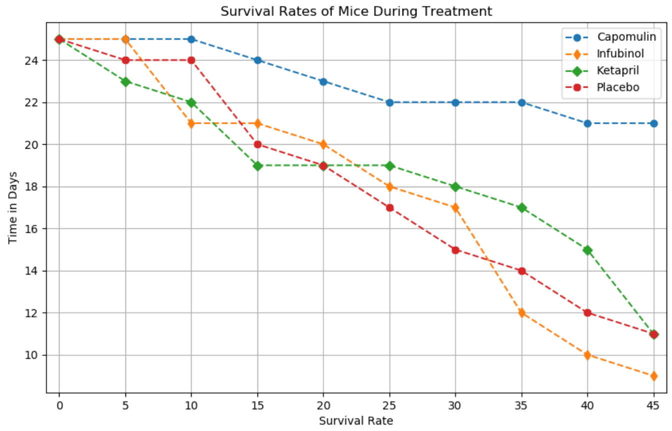
Tumor Volume Changes Over Time (images/01_tumorvolumechangesovertime.png)



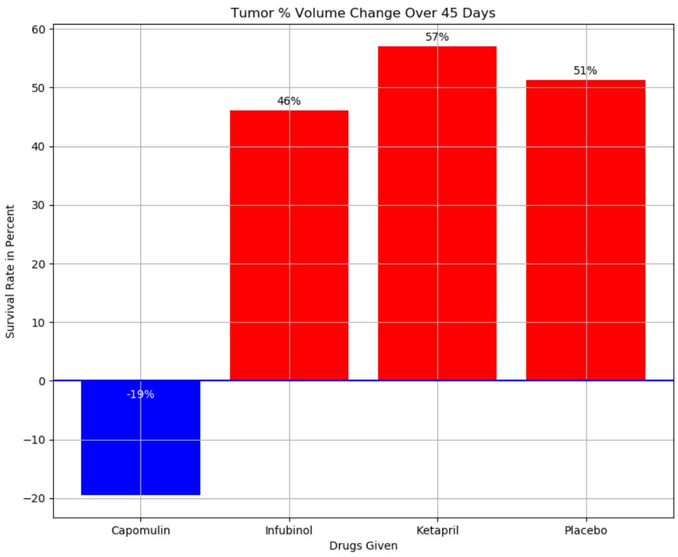
Metastatic Sites Change Over Time (images/01_tumorvolumechangesovertime.png)



Survival Rates Of Mice (images/01_tumorvolumechangesovertime.png)



Tumor Volume Change For Each Drug Over Time (images/01_tumorvolumechangesovertime.png)



Observable Trends

- From the Tumor volume change over time chart, a noticeable trend is displayed by the drug "Capomulin" in helping to reduce tumor size over. Immediate reductions can be observed within the first 5 days. Also quite evident, is the fact that the other 2 drugs (minus the placebo) did not seem to reduce nor accelerate tumor growth.
- For the number of Metastatic changes over time chart, Placebo is just slightly above or very near the effects for Ketapril. Ketapril however, seems to have the least amount of dramatic change where the data more linear than the other 3 types of drugs. This shows consistency for Ketapril use over time. This also exhibits a much slower decrease in Metastatic sites for Capomulin use.
- In viewing the survival rates of mice during treatment, Infubinol, Ketapril and Placebo all exhibit a much higher rate of mortality vs Capomulin. Its an average of 51% for the 3 drugs compared to the 32% difference in using Capomulin which is at -19% survival rate.

Dependencies

In [2]:

```
#dependencies
%matplotlib notebook

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Warnings

In [3]:

```
# Hide warning messages in notebook
import warnings
warnings.filterwarnings('ignore')
```

File Operations

In [4]:

```
#data sources
datafolder = 'data/'

file1 = 'clinicaltrial_data.csv'
file2 = 'mouse_drug_data.csv'

#read both files into dataframes
clinical_df = pd.read_csv(datafolder + file1)
mouse_df = pd.read_csv(datafolder + file2)
```

Data Verification

In [5]:

```
#physical data

#view clinical data
clinical_df.head(2)
```

Out[5]:

	Mouse ID	Timepoint	Tumor Volume (mm3)	Metastatic Sites
0	b128	0	45.0	0
1	f932	0	45.0	0

In [6]:

```
#view mouse data
mouse_df.head(2)
```

Out[6]:

	Mouse ID	Drug
0	f234	Stelasyn
1	x402	Stelasyn

In [7]:

```
#merge the data on clinical trials
combined_df = pd.merge(clinical_df, mouse_df, on="Mouse ID")

#sort the columns
combined_df.sort_values(["Mouse ID","Timepoint"], ascending=True, inplace=True
)

combined_df.head(2)
#combined_df.info()

#get the row count
initial_rowcount = len(combined_df)
#print(initial_rowcount)
```

In [8]:

```
#clear and reset the index
#https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.reset
_index.html
combined_df = combined_df.reset_index()
del combined_df['index']
combined_df.head()
```

Out[8]:

	Mouse ID	Timepoint	Tumor Volume (mm3)	Metastatic Sites	Drug
0	a203	0	45.000000	0	Infubinol
1	a203	5	48.508468	0	Infubinol
2	a203	10	51.852437	1	Infubinol
3	a203	15	52.777870	1	Infubinol
4	a203	20	55.173336	1	Infubinol

In [9]:

```
#output the merged data for viewing
combined_df.to_csv('data\merged_clinical_mouse.csv', header=True)
```

Data Cleansing

- verify the data makes sense
    - due to duplicate records and totals/numbers not making sense

In [10]:

```
#Clinical DF rows: 1893
#Mouse DF rows: 250
#Merged DF rows: 1906
print(f'Clinical DF rows: {len(clinical_df["Mouse ID"])}')
print(f'Mouse DF rows: {len(mouse_df["Mouse ID"])}')
print(f'Merged DF rows: {len(combined_df["Mouse ID"])}')

#something is wrong. we have more rows than we started with.
#there must be duplicate data someplace. MORE RESEARCH.
#output for visual
#combined_df.to_csv('combined_df.csv', header=True)

#imported to sql for due diligence and verification.
#issue verified

#FOUND AN ISSUE WITH MOUSE ID = g989
#MOUSE HAS 2 Different drugs associated with it causing duplicate rows???
```

```
Clinical DF rows: 1893
Mouse DF rows: 250
Merged DF rows: 1906
```

In [11]:

```
#Distinct Count of Drugs by MOUSE ID - aggregate summary
#https://stackoverflow.com/questions/18554920/pandas-aggregate-count-distinct
#df.groupby("date").agg({"duration": np.sum, "user_id": pd.Series.nunique})
#using response 97
distinct_count_df = combined_df.copy()
distinct_count_df = pd.DataFrame(distinct_count_df.groupby('Mouse ID').agg({'D
rug': pd.Series.nunique}))
print(len(distinct_count_df))
#distinct_count_df.head()
```

```
249
```

In [12]:

```
#search for ALL MICE causing disturbance in the data having a DISTINCT Drug co
unt > 1
distinct_count_df.loc[distinct_count_df['Drug']>1]
#mouseid countofdistinctdrugs
#g989   2

#what to do with the anomaly?
#SOLUTION is in the Project Description:
#Your objective is to analyze the data to show how four treatments
#(Capomulin, Infubinol, Ketapril, and Placebo)
```

Out[12]:

	Drug
Mouse ID	
g989	2

In [13]:

```
#get a new dataframe using only the data we need. in this case instructions sa
y to use
#the specific following drugs: (Capomulin, Infubinol, Ketapril, and Placebo)
#Create a dataframe for the filtered data
#research:
#https://stackoverflow.com/questions/11869910/pandas-filter-rows-of-dataframe-
with-operator-chaining
#https://stackoverflow.com/questions/22086116/how-do-you-filter-pandas-datafra
mes-by-multiple-columns
filtered_df = combined_df.copy()
filtered_pharma = filtered_df[
                                (filtered_df['Drug']=='Capomulin')
                                |
                                (filtered_df['Drug']=='Infubinol')
                                |
                                (filtered_df['Drug']=='Ketapril')
                                |
                                (filtered_df['Drug']=='Placebo')
                            ]

filtered_pharma.head(2)
```

Out[13]:

	Mouse ID	Timepoint	Tumor Volume (mm3)	Metastatic Sites	Drug
0	a203	0	45.000000	0	Infubinol
1	a203	5	48.508468	0	Infubinol

In [14]:

```
#double check our data anomalies. it should not show since it is not using one
 of the four
#required drug values - Mouse ID: g989
len(filtered_pharma[(filtered_pharma['Mouse ID']=='g989')])
```

Out[14]:

```
0
```

In [15]:

```
#rename the columns to a query friendly format because i hate spaces and paren
thesis.
#todo: comeback later and rename the columns to original names
print(filtered_pharma.columns)
filtered_pharma = filtered_pharma.rename(columns={
                                                    'Mouse ID':'MouseId'
                                                    ,'Tumor Volume (mm3)':'Tum
orVolume'
                                                    ,'Metastatic Sites':'MetaS
ites'
                                                })
```

```
Index(['Mouse ID', 'Timepoint', 'Tumor Volume (mm3)', 'Metastatic Sites',
       'Drug'],
      dtype='object')
```

In [16]:
```
#resort by timpoint for the first plot requirements
filtered_pharma.sort_values(["Timepoint","MouseId"], ascending=True, inplace=True)
#view renamed columns
filtered_pharma.head(2)
```

Out[16]:

	MouseId	Timepoint	TumorVolume	MetaSites	Drug
0	a203	0	45.0	0	Infubinol
10	a251	0	45.0	0	Infubinol

In [17]:
```
#reset index, get rid of INDEX or level_0 indices
filtered_pharma = filtered_pharma.reset_index()
del filtered_pharma['index']
filtered_pharma.head()
```

Out[17]:

	MouseId	Timepoint	TumorVolume	MetaSites	Drug
0	a203	0	45.0	0	Infubinol
1	a251	0	45.0	0	Infubinol
2	a262	0	45.0	0	Placebo
3	a457	0	45.0	0	Ketapril
4	a577	0	45.0	0	Infubinol

In [18]:
```
#create a working dataframe from clean data - this will be the basis for our graphs
main_df = filtered_pharma.copy()
print(len(main_df))
```

```
777
```

In [19]:
```
#output the clean data for viewing
main_df.to_csv('data\clean_clinical_mouse.csv', header=True)
```

Tumor Volume Changes Over Time

* Creating a scatter plot that shows how the tumor volume changes over time for each treatment.

In [20]:

```
#tumor response to treatment
#get the average tumor change over time
tumor_average_df = pd.DataFrame(main_df.groupby(['Drug','Timepoint']).mean())
#tumor_average_df.head(2)

#drop the MetaSites column
tumor_average_df = tumor_average_df[['TumorVolume']]
tumor_average_df.head(2)
```

Out[20]:

		TumorVolume
Drug	Timepoint	
Capomulin	0	45.000000
	5	44.266086

In [21]:

```
#dont forget the SEM is needed here as well
#tumor response to treatment
#find the sem of tumor change over time
tumor_sem_df = pd.DataFrame(main_df.groupby(['Drug','Timepoint']).sem())
#tumor_sem_df.head(2)

#drop the MetaSites column
tumor_sem_df = tumor_sem_df[['TumorVolume']]
tumor_sem_df.head(2)
```

Out[21]:

		TumorVolume
Drug	Timepoint	
Capomulin	0	0.000000
	5	0.448593

In [22]:

```
#pivot the table avg tumor table
tumor_volume = tumor_average_df.unstack(level = 0)
tumor_volume.head()
```

Out[22]:

	TumorVolume			
Drug	Capomulin	Infubinol	Ketapril	Placebo
Timepoint				
0	45.000000	45.000000	45.000000	45.000000
5	44.266086	47.062001	47.389175	47.125589
10	43.084291	49.403909	49.582269	49.423329
15	42.064317	51.296397	52.399974	51.359742
20	40.716325	53.197691	54.920935	54.364417

In [23]:

```
#pivot the SEM table
tumor_error = tumor_sem_df.unstack(level = 0)
tumor_error.head()
```

Out[23]:

	TumorVolume			
Drug	**Capomulin**	**Infubinol**	**Ketapril**	**Placebo**
Timepoint				
0	0.000000	0.000000	0.000000	0.000000
5	0.448593	0.235102	0.264819	0.218091
10	0.702684	0.282346	0.357421	0.402064
15	0.838617	0.357705	0.580268	0.614461
20	0.909731	0.476210	0.726484	0.839609

In [24]:

```
#get the values for the days on the XAXIS
xrange_df = pd.DataFrame(main_df.copy().drop_duplicates(['Timepoint'], keep="l
ast"))

begin_range = xrange_df['Timepoint'].min()
#print(f'Begin Range: {begin_range}')

end_range = xrange_df['Timepoint'].max()+5
#print(f'End Range: {end_range}')

step_range = 5
#print(f'Step Range: {step_range}')

x_axis = np.arange(begin_range, end_range, step_range)
#print(x_axis)

x_limit = end_range
#print(x_limit)
```

In [25]:

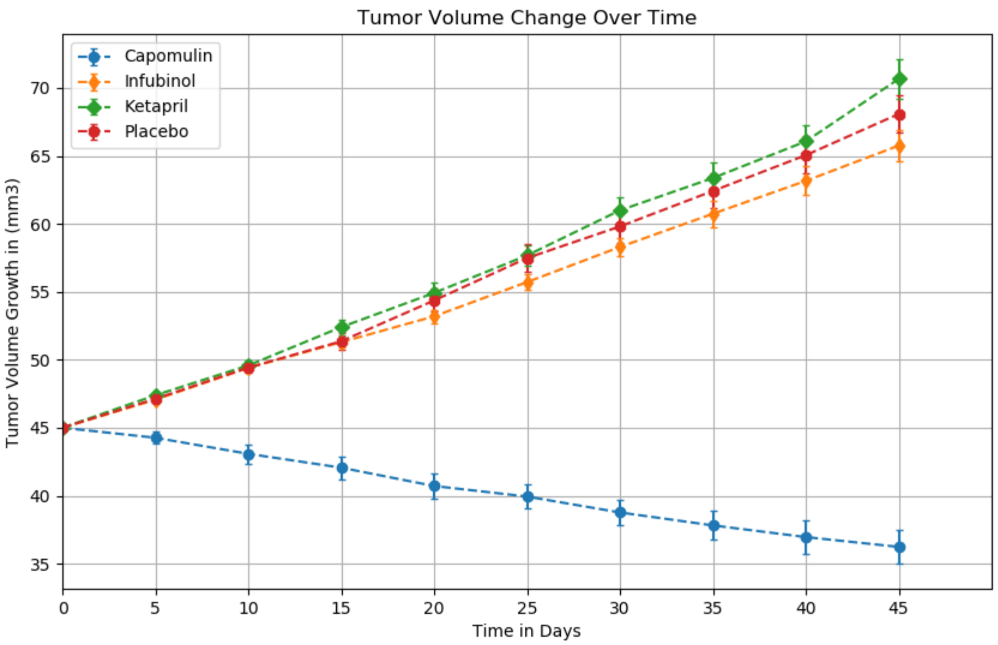
```
#drug values
drug_capomulin = tumor_volume[('TumorVolume', 'Capomulin')]
drug_infubinol = tumor_volume[('TumorVolume', 'Infubinol')]
drug_ketapril = tumor_volume[('TumorVolume', 'Ketapril')]
drug_placebo = tumor_volume[('TumorVolume', 'Placebo')]
```

In [26]:

```
#sem values
sem_capomulin = tumor_error[('TumorVolume', 'Capomulin')]
sem_infubinol = tumor_error[('TumorVolume', 'Infubinol')]
sem_ketapril = tumor_error[('TumorVolume', 'Ketapril')]
sem_placebo = tumor_error[('TumorVolume', 'Placebo')]
```

In [27]:

```
plt.figure(figsize=(10,6))

#make handles instead
scat_capomulin = plt.errorbar(x_axis, drug_capomulin, sem_capomulin, linestyle
='--', marker='o', capthick=1, capsize=2, label='Capomulin')
scat_infubinol = plt.errorbar(x_axis, drug_infubinol, sem_infubinol, linestyle
='--', marker='d', capthick=1, capsize=2, label='Infubinol')
scat_ketapril = plt.errorbar(x_axis, drug_ketapril, sem_ketapril, linestyle='-
-', marker='D', capthick=1, capsize=2, label='Ketapril')
scat_placebo = plt.errorbar(x_axis, drug_placebo, sem_placebo, linestyle='--',
 marker='8', capthick=1, capsize=2, label='Placebo')

#scatter plot 1
charttitle = "Tumor Volume Change Over Time"
xTitle = "Time in Days"
yTitle = "Tumor Volume Growth in (mm3)"

plt.title(charttitle)
plt.xlabel(xTitle)
plt.ylabel(yTitle)
plt.xlim(0, x_limit)

plt.xticks(x_axis)
plt.legend(handles=[scat_capomulin, scat_infubinol, scat_ketapril, scat_placeb
o], loc="best")
plt.grid()

#save the plot
plt.savefig("images/01_tumorvolumechangesovertime.png")

plt.show()
```



Metastatic Sites Change Over Time

* Creating a scatter plot that shows how the number of metastatic (cancer spreading) sites changes over time for each treatment.

In [28]:

```
#get the cancer spread average over time
cs_avg_df = pd.DataFrame(main_df.groupby(['Drug','Timepoint']).mean())
#cancer_average_df.head(2)

#drop the MetaSites column
cs_avg_df = cs_avg_df[['MetaSites']]
cs_avg_df.head(2)
```

Out[28]:

		MetaSites
Drug	**Timepoint**	
Capomulin	**0**	0.00
	5	0.16

In [29]:

```
#get the cancer spread sem over time
cs_sem_df = pd.DataFrame(main_df.groupby(['Drug','Timepoint']).sem())
#tumor_sem_df.head(2)

#drop the MetaSites column
cs_sem_df = cs_sem_df[['MetaSites']]
cs_sem_df.head(2)
```

Out[29]:

		MetaSites
Drug	**Timepoint**	
Capomulin	**0**	0.000000
	5	0.074833

In [30]:

```
#pivot the table avg tumor table
cs_spread_avg = cs_avg_df.unstack(level = 0)
cs_spread_avg.head()
```

Out[30]:

	MetaSites			
Drug	**Capomulin**	**Infubinol**	**Ketapril**	**Placebo**
Timepoint				
0	0.000000	0.000000	0.000000	0.000000
5	0.160000	0.280000	0.304348	0.375000
10	0.320000	0.666667	0.590909	0.833333
15	0.375000	0.904762	0.842105	1.250000
20	0.652174	1.050000	1.210526	1.526316

In [31]:

```
#pivot the SEM table
cs_spread_sem = cs_sem_df.unstack(level = 0)
cs_spread_sem.head()
```

Out[31]:

	MetaSites			
Drug	**Capomulin**	**Infubinol**	**Ketapril**	**Placebo**
Timepoint				
0	0.000000	0.000000	0.000000	0.000000
5	0.074833	0.091652	0.098100	0.100947
10	0.125433	0.159364	0.142018	0.115261
15	0.132048	0.194015	0.191381	0.190221
20	0.161621	0.234801	0.236680	0.234064

In [32]:

```
#get the values for the days on the XAXIS
xrange_cs_df = pd.DataFrame(main_df.copy().drop_duplicates(['Timepoint'], keep
="last"))

begin_cs_range = xrange_cs_df['Timepoint'].min()
#print(f'Begin Range: {begin_cs_range}')

end_cs_range = xrange_cs_df['Timepoint'].max()+1
#print(f'End Range: {end_cs_range}')

step_cs_range = 5
#print(f'Step Range: {step_cs_range}')
```

In [33]:

```
#cs_spread_avg.info()
```

In [34]:

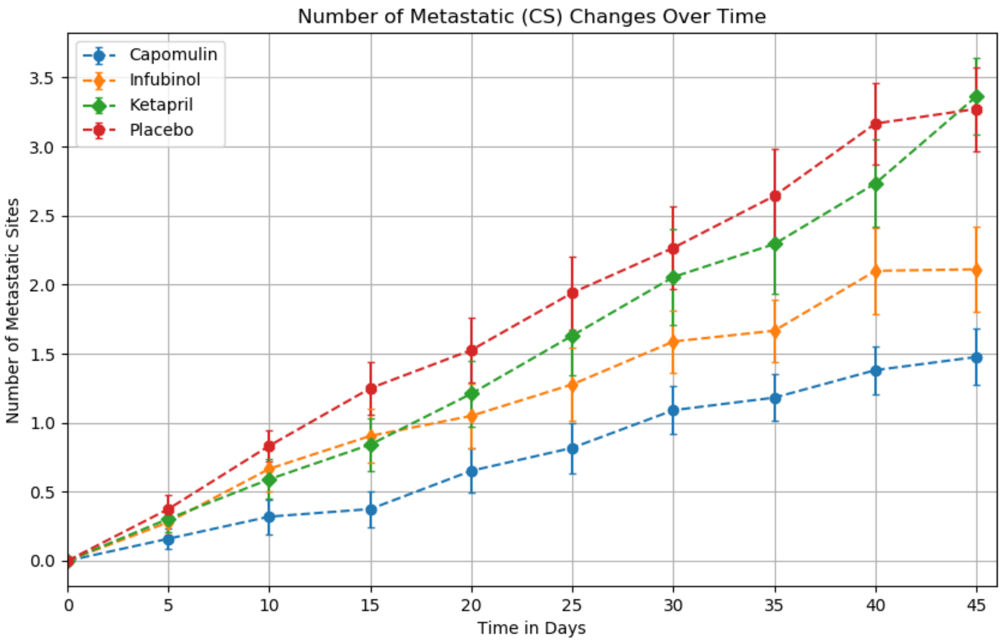
```
drug_cs_capomulin = cs_spread_avg[('MetaSites', 'Capomulin')]
drug_cs_infubinol = cs_spread_avg[('MetaSites', 'Infubinol')]
drug_cs_ketapril = cs_spread_avg[('MetaSites', 'Ketapril')]
drug_cs_placebo = cs_spread_avg[('MetaSites', 'Placebo')]
```

In [35]:

```
sem_cs_capomulin = cs_spread_sem[('MetaSites', 'Capomulin')]
sem_cs_infubinol = cs_spread_sem[('MetaSites', 'Infubinol')]
sem_cs_ketapril = cs_spread_sem[('MetaSites', 'Ketapril')]
sem_cs_placebo = cs_spread_sem[('MetaSites', 'Placebo')]
```

In [36]:

```
#chart info
plt.figure(figsize=(10,6))
charttitle_cs = "Number of Metastatic (CS) Changes Over Time"
xTitle_cs = "Time in Days"
yTitle_cs = "Number of Metastatic Sites"
x_limit_cs = end_cs_range
#x_axis = np.arange(xcancerrange_df['Timepoint'].min(), xcancerrange_df['Timepoint'].max(), 5)
#maybe add 5 to the end range to give us a nice number 0-50???
x_cs_axis = np.arange(begin_cs_range, end_cs_range, step_cs_range)
#print(x_cs_axis)

scat_cs_capomulin = plt.errorbar(x_cs_axis, drug_cs_capomulin, sem_cs_capomulin, linestyle='--', marker='o', capthick=1, capsize=2, label='Capomulin')
scat_cs_infubinol = plt.errorbar(x_cs_axis, drug_cs_infubinol, sem_cs_infubinol, linestyle='--', marker='d', capthick=1, capsize=2, label='Infubinol')
scat_cs_ketapril = plt.errorbar(x_cs_axis, drug_cs_ketapril, sem_cs_ketapril, linestyle='--', marker='D', capthick=1, capsize=2, label='Ketapril')
scat_cs_placebo = plt.errorbar(x_cs_axis, drug_cs_placebo, sem_cs_placebo, linestyle='--', marker='8', capthick=1, capsize=2, label='Placebo')


plt.title(charttitle_cs)
plt.xlabel(xTitle_cs)
plt.ylabel(yTitle_cs)
plt.xlim(0, x_limit_cs)
plt.xticks(x_cs_axis)

plt.legend(handles=[scat_cs_capomulin, scat_cs_infubinol, scat_cs_ketapril, scat_cs_placebo], loc="best")
plt.grid()

#save the plot
plt.savefig("images/02_metastaticchangesovertime.png")

plt.show()
```



Survival Rates Of Mice

* Creating a scatter plot that shows the number of mice still alive through the course of treatment (Survival Rate).

In [37]:
```
#get the count of mice within the timepoints, i believe that if they are on th
e timepoint and being given treatment then
#they are still alive
sr_mice_df = main_df.groupby(['Drug','Timepoint']).count()
#['MouseId']
#get only the mouseid data
sr_mice_df = pd.DataFrame(sr_mice_df['MouseId'])
sr_mice_df.head(2)
```

Out[37]:

		MouseId
Drug	**Timepoint**	
Capomulin	**0**	25
	5	25

In [38]:

```
#pivot the survival rates of mice
sr_mice_results = sr_mice_df.unstack(level = 0)
sr_mice_results.head()

#how would you add the standard margin of error or does it even come into play
 for this?
```

Out[38]:

	MouseId			
Drug	**Capomulin**	**Infubinol**	**Ketapril**	**Placebo**
Timepoint				
0	25	25	25	25
5	25	25	23	24
10	25	21	22	24
15	24	21	19	20
20	23	20	19	19

In [39]:

```
#https://stackoverflow.com/questions/44890713/selection-with-loc-in-python
#get the index in as a list or array
#https://stackoverflow.com/questions/17241004/pandas-how-to-get-the-data-frame
-index-as-an-array
sr_mice_results.index
```

Out[39]:

```
Int64Index([0, 5, 10, 15, 20, 25, 30, 35, 40, 45], dtype='int64', name='Timep
oint')
```

In [40]:

```
#get the values for the timepoints on the XAXIS
xrange_sr_df = pd.DataFrame(sr_mice_results.index)

begin_sr_range = xrange_sr_df['Timepoint'].min()
print(f'Begin Range: {begin_sr_range}')

end_sr_range = xrange_sr_df['Timepoint'].max()+1
print(f'End Range: {end_sr_range}')

step_sr_range = 5
print(f'Step Range: {step_sr_range}')

#print(xcancerrange_df['Timepoint'].min())
#print(xcancerrange_df['Timepoint'].max())
#print((end_range+step_range))
#x_axis = np.arange(begin_range, end_range, step_range)
#x_axis
```

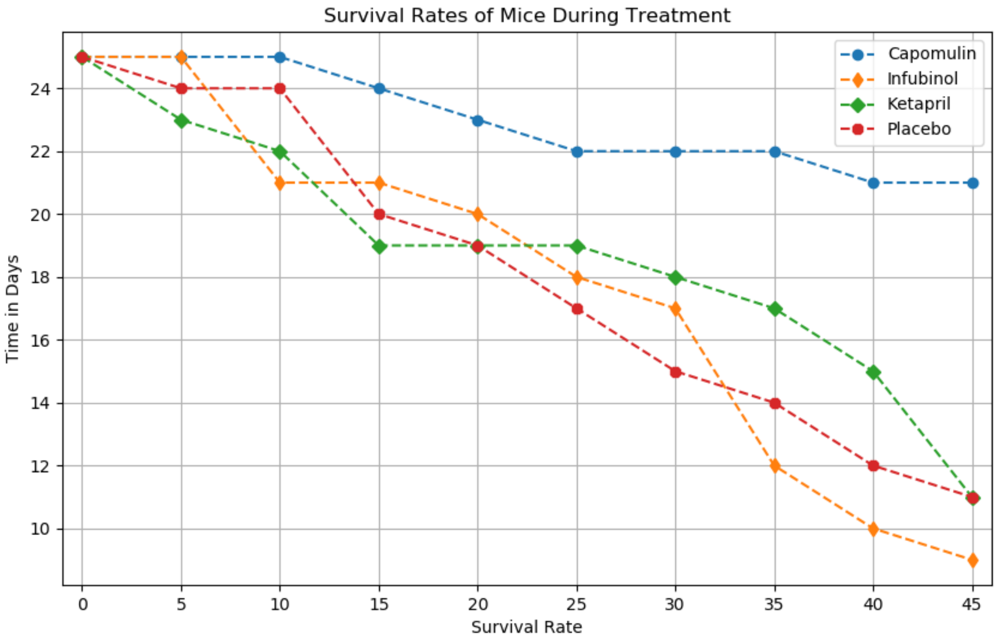
```
Begin Range: 0
End Range: 46
Step Range: 5
```

In [41]:
```
#chart info
charttitle_sr = "Survival Rates of Mice During Treatment"
xTitle_sr = "Survival Rate"
yTitle_sr = "Time in Days"

x_limit_sr = end_sr_range
print(x_limit_sr)

#x_axis = np.arange(xcancerrange_df['Timepoint'].min(), xcancerrange_df['Timep
oint'].max(), 5)
#maybe add 5 to the end range to give us a nice number 0-50???
x_sr_axis = np.arange(begin_sr_range, end_sr_range, step_sr_range)
print(x_sr_axis)
```

```
46
[ 0  5 10 15 20 25 30 35 40 45]
```

In [42]:
```
drug_sr_capomulin = sr_mice_results[('MouseId', 'Capomulin')]
drug_sr_infubinol = sr_mice_results[('MouseId', 'Infubinol')]
drug_sr_ketapril = sr_mice_results[('MouseId', 'Ketapril')]
drug_sr_placebo = sr_mice_results[('MouseId', 'Placebo')]
```

In [43]:
```
plt.figure(figsize=(10,6))

scat_sr_capomulin = plt.errorbar(x_sr_axis, drug_sr_capomulin, linestyle='--', marker='o', capthick=1, capsize=2, label='Capomulin')
scat_sr_infubinol = plt.errorbar(x_sr_axis, drug_sr_infubinol, linestyle='--', marker='d', capthick=1, capsize=2, label='Infubinol')
scat_sr_ketapril = plt.errorbar(x_sr_axis, drug_sr_ketapril, linestyle='--', marker='D', capthick=1, capsize=2, label='Ketapril')
scat_sr_placebo = plt.errorbar(x_sr_axis, drug_sr_placebo, linestyle='--', marker='8', capthick=1, capsize=2, label='Placebo')


#scatter plot 3
plt.title(charttitle_sr)
plt.xlabel(xTitle_sr)
plt.ylabel(yTitle_sr)
plt.xlim(-1, x_limit_sr)

plt.xticks(x_sr_axis)

plt.legend(handles=[scat_sr_capomulin, scat_sr_infubinol, scat_sr_ketapril, scat_sr_placebo], loc="best")

plt.grid()

#save the plot
plt.savefig("images/03_survivalratesofmice.png")

plt.show()
```



Tumor Volume Change For Each Drug Over Time

* Creating a bar graph that compares the total % tumor volume change for each drug across the full 45 days.

In [44]:

```
tumorvolume_avg_df = main_df.groupby(['Drug', 'Timepoint']).mean()
tumorvolume_avg_df = pd.DataFrame(tumorvolume_avg_df['TumorVolume'])
tumorvolume_avg_df.head(2)
```

Out[44]:

		TumorVolume
Drug	**Timepoint**	
Capomulin	**0**	45.000000
	5	44.266086

In [45]:

```
#pivot tumor values
tumorvalues = pd.DataFrame(tumorvolume_avg_df.unstack(level = 0))
tumorvalues
```

Out[45]:

	TumorVolume			
Drug	**Capomulin**	**Infubinol**	**Ketapril**	**Placebo**
Timepoint				
0	45.000000	45.000000	45.000000	45.000000
5	44.266086	47.062001	47.389175	47.125589
10	43.084291	49.403909	49.582269	49.423329
15	42.064317	51.296397	52.399974	51.359742
20	40.716325	53.197691	54.920935	54.364417
25	39.939528	55.715252	57.678982	57.482574
30	38.769339	58.299397	60.994507	59.809063
35	37.816839	60.742461	63.371686	62.420615
40	36.958001	63.162824	66.068580	65.052675
45	36.236114	65.755562	70.662958	68.084082

In [46]:

```
#get the ranges in the timepoints
tumorranges = tumorvalues.index
tumorranges
```

Out[46]:

```
Int64Index([0, 5, 10, 15, 20, 25, 30, 35, 40, 45], dtype='int64', name='Timep
oint')
```

In [47]:

```
#get the starting position of the array
tum_range_start = 0
#get the number of values in the array and subtract 1
#print(len(tumorranges))
tum_range_end = len(tumorranges)-1
#print(f'True Array length : {tum_range_end}')
```

In [48]:

```
#verify values are being calculated properly
percenttumor = (tumorvalues.iloc[tum_range_end,:] - tumorvalues.iloc[tum_range
_start,:] ) / tumorvalues.iloc[tum_range_start,:] * 100
#formula is end change - start change / start change * 100(for percentage)
percenttumor = pd.DataFrame(percenttumor)
percenttumor
```

Out[48]:

		0
	Drug	
TumorVolume	**Capomulin**	-19.475303
	Infubinol	46.123472
	Ketapril	57.028795
	Placebo	51.297960

In [49]:

```
#reset index, get rid of INDEX or level_0 indices
percenttumor = percenttumor.reset_index()
del percenttumor['level_0']
percenttumor.head()
```

Out[49]:

	Drug	0
0	Capomulin	-19.475303
1	Infubinol	46.123472
2	Ketapril	57.028795
3	Placebo	51.297960

In [50]:

```
#percenttumor = percenttumor.reset_index()
#del percenttumor['level_0']
#percenttumor.head()
percenttumor = percenttumor.rename(columns={0:'PercentChange'})
#percenttumor.info()
#percenttumor.head()
```

In [51]:

```
#get a list of plot values
col_percent_tumor = percenttumor['PercentChange']
col_percent_tumor
```

Out[51]:

```
0   -19.475303
1    46.123472
2    57.028795
3    51.297960
Name: PercentChange, dtype: float64
```

In [52]:

```
#find the proper color based on percent value
#percentagecolors = []
#for value in percenttumor['PercentChange']:
#    #print(value)
#    if value < 0:
#        percentagecolors.append('blue')
#    else:
#        percentagecolors.append('red')

#now change into list comprehension
#https://stackoverflow.com/questions/4406389/if-else-in-a-list-comprehension
percentagecolors = [ 'blue' if perc <0 else 'red' for perc in percenttumor['PercentChange']]
```

In [53]:

```
#get the names from dframe
tt_ticknames=percenttumor['Drug']
```

In [54]:

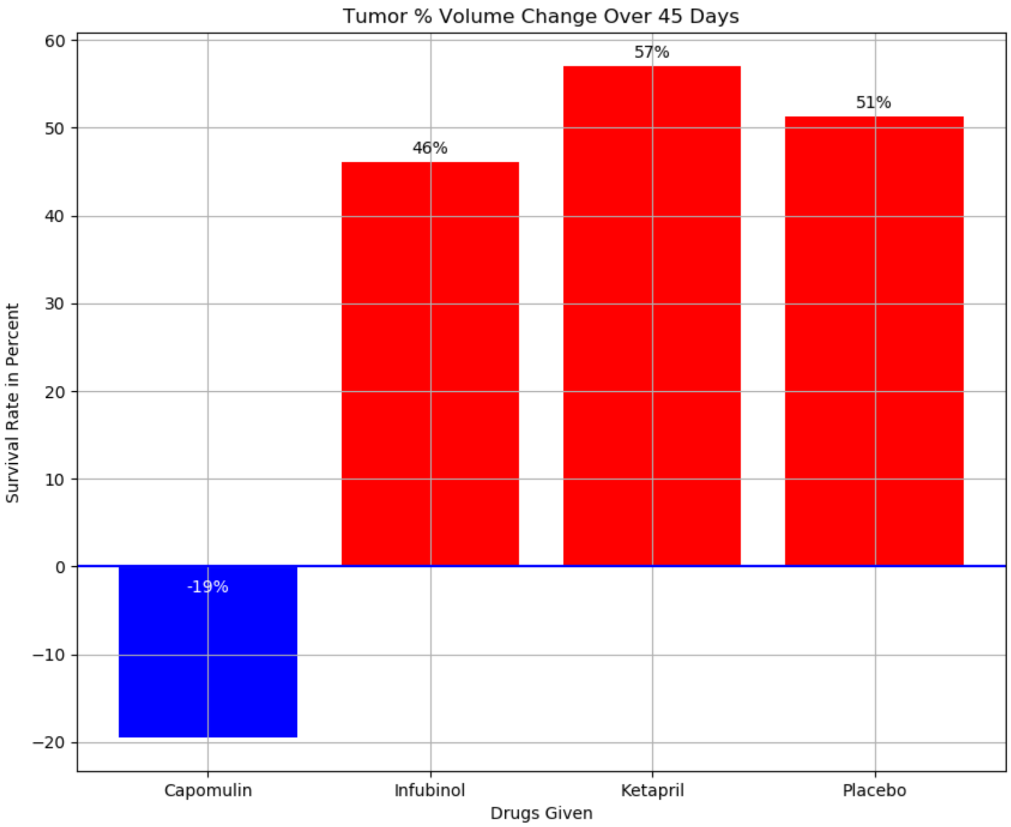
```
tt_charttitle ='Tumor % Volume Change Over 45 Days'
tt_ylabel = 'Survival Rate in Percent'
tt_xlabel = 'Drugs Given'

tt_xticks = np.arange(0, len(percenttumor.index), 1)
tt_xticks
```

Out[54]:

```
array([0, 1, 2, 3])
```

In [55]:

```
plt.figure(figsize=(10,8))

#plain bar, todo: check values
#plt.bar(tt_xticks, col_percent_tumor)

#add the colors
plt.bar(tt_xticks, col_percent_tumor, color=percentagecolors)

#add tick values
plt.xticks(tt_xticks, tt_ticknames)

#add a line on zero
plt.axhline(y=0, color = 'blue')


plt.title(tt_charttitle)
plt.xlabel(tt_xlabel)
plt.ylabel(tt_ylabel)


#format x, y, text, h align, color
#plt.text(1, 5, 'PERCENTAGE%', ha = 'center', color = 'white')

#add a counter to track the row
i = 0

for row in percenttumor['PercentChange']:
    numperc=round(row,2)
    tumorperc = "{0:.0f}%".format(numperc)

    #need to change the y value else all labels are on horizontal zero line
    if numperc < 0:
        yloc = -3
        ycolor = 'white'
    else:
        yloc = numperc + 1
        ycolor = 'black'

    plt.text(i, yloc, tumorperc,ha='center', color=ycolor)

    #increase counts
    i += 1


plt.grid()

#save the plot
plt.savefig("Images/04_tumorvolumechangesinpercent.png")

plt.show()
```


Tumor % Volume Change Over 45 Days
Survival Rate in Percent
60
50
40
30
20
10
0
−10
−20
46%
57%
51%
-19%
Capomulin
Infubinol
Ketapril
Placebo
Drugs Given

Unit 5 | Pymaceuticals Inc

While your data companions rushed off to jobs in finance and government, you remained adamant that science was the way for you. Staying true to your mission, you've since joined Pymaceuticals Inc., a burgeoning pharmaceutical company based out of San Diego, CA. Pymaceuticals specializes in drug-based, anti-cancer pharmaceuticals. In their most recent efforts, they've since begun screening for potential treatments to squamous cell carcinoma (SCC), a commonly occurring form of skin cancer.

As their Chief Data Analyst, you've been given access to the complete data from their most recent animal study. In this study, 250 mice were treated through a variety of drug regimes over the course of 45 days. Their physiological responses were then monitored over the course of that time. Your objective is to analyze the data to show how four treatments (Capomulin, Infubinol, Ketapril, and Placebo) compare.

To do this you are tasked with:

- Creating a scatter plot that shows how the tumor volume changes over time for each treatment.
- Creating a scatter plot that shows how the number of metastatic (https://en.wikipedia.org/wiki/Metastasis) (cancer spreading) sites changes over time for each treatment.
- Creating a scatter plot that shows the number of mice still alive through the course of treatment (Survival Rate)
- Creating a bar graph that compares the total % tumor volume change for each drug across the full 45 days.