# Dependencies and Setup

%matplotlib inline

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

# Incorporate Seaborn if preferred

# import seaborn as sns

mouse\_drug\_data\_to\_load = "raw\_data/mouse\_drug\_data.csv"

clinical\_trial\_data\_to\_load = "raw\_data/clinicaltrial\_data.csv"

# Read the Mouse and Drug Data and the Clinical Trial Data

mouse\_drug\_data = pd.read\_csv(mouse\_drug\_data\_to\_load)

clinical\_data = pd.read\_csv(clinical\_trial\_data\_to\_load)

# Combine the data into a single dataset

clinical\_data\_complete = pd.merge(clinical\_data, mouse\_drug\_data, how="left", on=["Mouse ID", "Mouse ID"])

# Display the data table for preview

clinical\_data\_complete.head()

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

Tumor Response to Treatment

# Store the Mean Tumor Volume Data Grouped by Drug and Timepoint

tumor\_vols\_mean = clinical\_data\_complete.groupby(["Drug", "Timepoint"]).mean()["Tumor Volume (mm3)"]

# Convert to DataFrame

tumor\_vols\_mean = pd.DataFrame(tumor\_vols\_mean)

# Preview DataFrame

tumor\_vols\_mean

|  |  | Tumor Volume (mm3) |
| --- | --- | --- |
| Drug | Timepoint |  |
| Capomulin | 0 | 45.000000 |
| 5 | 44.266086 |
| 10 | 43.084291 |
| 15 | 42.064317 |
| 20 | 40.716325 |
| 25 | 39.939528 |
| 30 | 38.769339 |
| 35 | 37.816839 |
| 40 | 36.958001 |
| 45 | 36.236114 |
| Ceftamin | 0 | 45.000000 |
| 5 | 46.503051 |
| 10 | 48.285125 |
| 15 | 50.094055 |
| 20 | 52.157049 |
| 25 | 54.287674 |
| 30 | 56.769517 |
| 35 | 58.827548 |
| 40 | 61.467895 |
| 45 | 64.132421 |
| Infubinol | 0 | 45.000000 |
| 5 | 47.062001 |
| 10 | 49.403909 |
| 15 | 51.296397 |
| 20 | 53.197691 |
| 25 | 55.715252 |
| 30 | 58.299397 |
| 35 | 60.742461 |
| 40 | 63.162824 |
| 45 | 65.755562 |
| ... | ... | ... |
| Ramicane | 0 | 45.000000 |
| 5 | 43.944859 |
| 10 | 42.531957 |
| 15 | 41.495061 |
| 20 | 40.238325 |
| 25 | 38.974300 |
| 30 | 38.703137 |
| 35 | 37.451996 |
| 40 | 36.574081 |
| 45 | 34.955595 |
| Stelasyn | 0 | 45.000000 |
| 5 | 47.527452 |
| 10 | 49.463844 |
| 15 | 51.529409 |
| 20 | 54.067395 |
| 25 | 56.166123 |
| 30 | 59.826738 |
| 35 | 62.440699 |
| 40 | 65.356386 |
| 45 | 68.438310 |
| Zoniferol | 0 | 45.000000 |
| 5 | 46.851818 |
| 10 | 48.689881 |
| 15 | 50.779059 |
| 20 | 53.170334 |
| 25 | 55.432935 |
| 30 | 57.713531 |
| 35 | 60.089372 |
| 40 | 62.916692 |
| 45 | 65.960888 |

100 rows × 1 columns

# Store the Standard Error of Tumor Volumes Grouped by Drug and Timepoint

tumor\_vols\_sem = clinical\_data\_complete.groupby(["Drug", "Timepoint"]).sem()["Tumor Volume (mm3)"]

# Convert to DataFrame

tumor\_vols\_sem = pd.DataFrame(tumor\_vols\_sem)

# Preview DataFrame

tumor\_vols\_sem.head()

|  |  | Tumor Volume (mm3) |
| --- | --- | --- |
| Drug | Timepoint |  |
| Capomulin | 0 | 0.000000 |
| 5 | 0.448593 |
| 10 | 0.702684 |
| 15 | 0.838617 |
| 20 | 0.909731 |

# Minor Data Munging to Re-Format the Data Frames

tumor\_vols\_mean = tumor\_vols\_mean.reset\_index()

tumor\_vols\_pivot\_mean = tumor\_vols\_mean.pivot(index="Timepoint", columns="Drug")["Tumor Volume (mm3)"]

tumor\_vols\_sem = tumor\_vols\_sem.reset\_index()

tumor\_vols\_pivot\_sem = tumor\_vols\_sem.pivot(index="Timepoint", columns="Drug")["Tumor Volume (mm3)"]

# Preview that Reformatting worked

tumor\_vols\_pivot\_mean.head()

| Drug | Capomulin | Ceftamin | Infubinol | Ketapril | Naftisol | Placebo | Propriva | Ramicane | Stelasyn | Zoniferol |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Timepoint |  |  |  |  |  |  |  |  |  |  |
| 0 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 |
| 5 | 44.266086 | 46.503051 | 47.062001 | 47.389175 | 46.796098 | 47.125589 | 47.248967 | 43.944859 | 47.527452 | 46.851818 |
| 10 | 43.084291 | 48.285125 | 49.403909 | 49.582269 | 48.694210 | 49.423329 | 49.101541 | 42.531957 | 49.463844 | 48.689881 |
| 15 | 42.064317 | 50.094055 | 51.296397 | 52.399974 | 50.933018 | 51.359742 | 51.067318 | 41.495061 | 51.529409 | 50.779059 |
| 20 | 40.716325 | 52.157049 | 53.197691 | 54.920935 | 53.644087 | 54.364417 | 53.346737 | 40.238325 | 54.067395 | 53.170334 |

# Generate the Plot (with Error Bars)

plt.errorbar(tumor\_vols\_pivot\_mean.index, tumor\_vols\_pivot\_mean["Capomulin"], yerr=tumor\_vols\_pivot\_sem["Capomulin"], color="r", marker="o", markersize=5, linestyle="dashed", linewidth=0.50)

plt.errorbar(tumor\_vols\_pivot\_mean.index, tumor\_vols\_pivot\_mean["Infubinol"], yerr=tumor\_vols\_pivot\_sem["Infubinol"], color="b", marker="^", markersize=5, linestyle="dashed", linewidth=0.50)

plt.errorbar(tumor\_vols\_pivot\_mean.index, tumor\_vols\_pivot\_mean["Ketapril"], yerr=tumor\_vols\_pivot\_sem["Ketapril"], color="g", marker="s", markersize=5, linestyle="dashed", linewidth=0.50)

plt.errorbar(tumor\_vols\_pivot\_mean.index, tumor\_vols\_pivot\_mean["Placebo"], yerr=tumor\_vols\_pivot\_sem["Placebo"], color="k", marker="d", markersize=5, linestyle="dashed", linewidth=0.50)

plt.title("Tumor Response to Treatment")

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

plt.xlabel("Time (Days)")

plt.grid(True)

plt.legend(loc="best", fontsize="small", fancybox=True)

# Save the Figure

plt.savefig("analysis/Fig1.png")

# Show the Figure

plt.show()

Metastatic Response to Treatment

# Store the Mean Met. Site Data Grouped by Drug and Timepoint

met\_sites\_mean = clinical\_data\_complete.groupby(["Drug", "Timepoint"]).mean()["Metastatic Sites"]

# Convert to DataFrame

met\_sites\_mean = pd.DataFrame(met\_sites\_mean)

# Preview DataFrame

met\_sites\_mean.head()

|  |  | Metastatic Sites |
| --- | --- | --- |
| Drug | Timepoint |  |
| Capomulin | 0 | 0.000000 |
| 5 | 0.160000 |
| 10 | 0.320000 |
| 15 | 0.375000 |
| 20 | 0.652174 |

# Store the Standard Error associated with Met. Sites Grouped by Drug and Timepoint

met\_sites\_sem = clinical\_data\_complete.groupby(["Drug", "Timepoint"]).sem()["Metastatic Sites"]

# Convert to DataFrame

met\_sites\_sem = pd.DataFrame(met\_sites\_sem)

# Preview DataFrame

met\_sites\_sem.head()

|  |  | Metastatic Sites |
| --- | --- | --- |
| Drug | Timepoint |  |
| Capomulin | 0 | 0.000000 |
| 5 | 0.074833 |
| 10 | 0.125433 |
| 15 | 0.132048 |
| 20 | 0.161621 |

# Minor Data Munging to Re-Format the Data Frames

met\_sites\_mean = met\_sites\_mean.reset\_index()

met\_sites\_pivot\_mean = met\_sites\_mean.pivot(index="Timepoint", columns="Drug")["Metastatic Sites"]

met\_sites\_sem = met\_sites\_sem.reset\_index()

met\_sites\_pivot\_sem = met\_sites\_sem.pivot(index="Timepoint", columns="Drug")["Metastatic Sites"]

# Preview that Reformatting worked

tumor\_vols\_pivot\_mean.head()

| Drug | Capomulin | Ceftamin | Infubinol | Ketapril | Naftisol | Placebo | Propriva | Ramicane | Stelasyn | Zoniferol |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Timepoint |  |  |  |  |  |  |  |  |  |  |
| 0 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 | 45.000000 |
| 5 | 44.266086 | 46.503051 | 47.062001 | 47.389175 | 46.796098 | 47.125589 | 47.248967 | 43.944859 | 47.527452 | 46.851818 |
| 10 | 43.084291 | 48.285125 | 49.403909 | 49.582269 | 48.694210 | 49.423329 | 49.101541 | 42.531957 | 49.463844 | 48.689881 |
| 15 | 42.064317 | 50.094055 | 51.296397 | 52.399974 | 50.933018 | 51.359742 | 51.067318 | 41.495061 | 51.529409 | 50.779059 |
| 20 | 40.716325 | 52.157049 | 53.197691 | 54.920935 | 53.644087 | 54.364417 | 53.346737 | 40.238325 | 54.067395 | 53.170334 |

# Generate the Plot (with Error Bars)

plt.errorbar(met\_sites\_pivot\_mean.index, met\_sites\_pivot\_mean["Capomulin"], yerr=met\_sites\_pivot\_sem["Capomulin"], color="r", marker="o", markersize=5, linestyle="dashed", linewidth=0.50)

plt.errorbar(met\_sites\_pivot\_mean.index, met\_sites\_pivot\_mean["Infubinol"], yerr=met\_sites\_pivot\_sem["Infubinol"], color="b", marker="^", markersize=5, linestyle="dashed", linewidth=0.50)

plt.errorbar(met\_sites\_pivot\_mean.index, met\_sites\_pivot\_mean["Ketapril"], yerr=met\_sites\_pivot\_sem["Ketapril"], color="g", marker="s", markersize=5, linestyle="dashed", linewidth=0.50)

plt.errorbar(met\_sites\_pivot\_mean.index, met\_sites\_pivot\_mean["Placebo"], yerr=met\_sites\_pivot\_sem["Placebo"], color="k", marker="d", markersize=5, linestyle="dashed", linewidth=0.50)

plt.title("Metastatic Spread During Treatment")

plt.ylabel("Met. Sites")

plt.xlabel("Treatment Duration (Days)")

plt.grid(True)

plt.legend(loc="best", fontsize="small", fancybox=True)

# Save the Figure

plt.savefig("analysis/Fig2.png")

# Show the Figure

plt.show()

Survival Rates

# Store the Count of Mice Grouped by Drug and Timepoint (W can pass any metric)

survival\_count = clinical\_data\_complete.groupby(["Drug", "Timepoint"]).count()["Tumor Volume (mm3)"]

# Convert to DataFrame

survival\_count = pd.DataFrame({"Mouse Count": survival\_count})

# Preview DataFrame

survival\_count.head()

|  |  | Mouse Count |
| --- | --- | --- |
| Drug | Timepoint |  |
| Capomulin | 0 | 25 |
| 5 | 25 |
| 10 | 25 |
| 15 | 24 |
| 20 | 23 |

# Minor Data Munging to Re-Format the Data Frames

survival\_count = survival\_count.reset\_index()

survival\_count\_pivot = survival\_count.pivot(index="Timepoint", columns="Drug")["Mouse Count"]

# Preview the Data Frame

survival\_count\_pivot.head()

| Drug | Capomulin | Ceftamin | Infubinol | Ketapril | Naftisol | Placebo | Propriva | Ramicane | Stelasyn | Zoniferol |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Timepoint |  |  |  |  |  |  |  |  |  |  |
| 0 | 25 | 25 | 25 | 25 | 25 | 25 | 26 | 25 | 26 | 25 |
| 5 | 25 | 21 | 25 | 23 | 23 | 24 | 25 | 25 | 25 | 24 |
| 10 | 25 | 20 | 21 | 22 | 21 | 24 | 23 | 24 | 23 | 22 |
| 15 | 24 | 19 | 21 | 19 | 21 | 20 | 17 | 24 | 23 | 21 |
| 20 | 23 | 18 | 20 | 19 | 20 | 19 | 17 | 23 | 21 | 17 |

# Generate the Plot (Accounting for percentages)

plt.plot(100 \* survival\_count\_pivot["Capomulin"] / 25, "ro", linestyle="dashed", markersize=5, linewidth=0.50)

plt.plot(100 \* survival\_count\_pivot["Infubinol"] / 25, "b^", linestyle="dashed", markersize=5, linewidth=0.50)

plt.plot(100 \* survival\_count\_pivot["Ketapril"] / 25, "gs", linestyle="dashed", markersize=5, linewidth=0.50)

plt.plot(100 \* survival\_count\_pivot["Placebo"] / 25 , "kd", linestyle="dashed", markersize=6, linewidth=0.50)

plt.title("Survival During Treatment")

plt.ylabel("Survival Rate (%)")

plt.xlabel("Time (Days)")

plt.grid(True)

plt.legend(loc="best", fontsize="small", fancybox=True)

# Save the Figure

plt.savefig("analysis/Fig3.png")

# Show the Figure

plt.show()

Summary Bar Graph

# Calculate the percent changes for each drug

tumor\_pct\_change = 100 \* (tumor\_vols\_pivot\_mean.iloc[-1] - tumor\_vols\_pivot\_mean.iloc[0]) / tumor\_vols\_pivot\_mean.iloc[0]

tumor\_pct\_change\_sem = 100 \* (tumor\_vols\_pivot\_sem.iloc[-1] - tumor\_vols\_pivot\_sem.iloc[0]) / tumor\_vols\_pivot\_sem.iloc[0]

# Display the data to confirm

tumor\_pct\_change

Drug

Capomulin -19.475303

Ceftamin 42.516492

Infubinol 46.123472

Ketapril 57.028795

Naftisol 53.923347

Placebo 51.297960

Propriva 47.241175

Ramicane -22.320900

Stelasyn 52.085134

Zoniferol 46.579751

dtype: float64

# Store all Relevant Percent Changes into a Tuple

pct\_changes = (tumor\_pct\_change["Capomulin"],

tumor\_pct\_change["Infubinol"],

tumor\_pct\_change["Ketapril"],

tumor\_pct\_change["Placebo"])

# Splice the data between passing and failing drugs

fig, ax = plt.subplots()

ind = np.arange(len(pct\_changes))

width = 1

rectsPass = ax.bar(ind[0], pct\_changes[0], width, color='green')

rectsFail = ax.bar(ind[1:], pct\_changes[1:], width, color='red')

# Orient widths. Add labels, tick marks, etc.

ax.set\_ylabel('% Tumor Volume Change')

ax.set\_title('Tumor Change Over 45 Day Treatment')

ax.set\_xticks(ind + 0.5)

ax.set\_xticklabels(('Capomulin', 'Infubinol', 'Ketapril', 'Placebo'))

ax.set\_autoscaley\_on(False)

ax.set\_ylim([-30,70])

ax.grid(True)

# Use functions to label the percentages of changes

def autolabelFail(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x() + rect.get\_width()/2., 3,

'%d%%' % int(height),

ha='center', va='bottom', color="white")

def autolabelPass(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x() + rect.get\_width()/2., -8,

'-%d%% ' % int(height),

ha='center', va='bottom', color="white")

# Call functions to implement the function calls

autolabelPass(rectsPass)

autolabelFail(rectsFail)

# Save the Figure

fig.savefig("analysis/Fig4.png")

# Show the Figure

fig.show()

C:\Users\Ahmed\Anaconda3\envs\PythonData\lib\site-packages\matplotlib\figure.py:397: UserWarning: matplotlib is currently using a non-GUI backend, so cannot show the figure

"matplotlib is currently using a non-GUI backend, "

© 2019 GitHub, Inc.

[Terms](https://github.com/site/terms)

[Privacy](https://github.com/site/privacy)

[Security](https://github.com/security)

[Status](https://githubstatus.com/)

[Help](https://help.github.com)