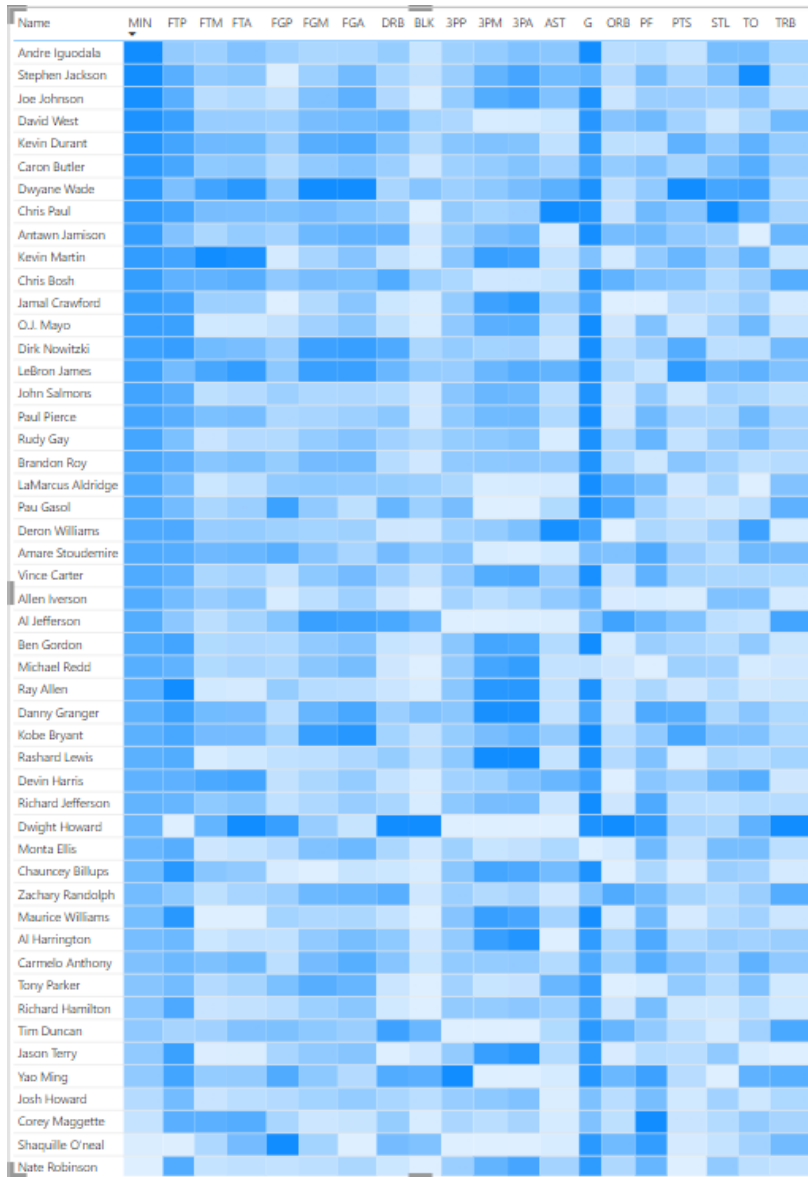


# DSC 640 – Week 9 & 10

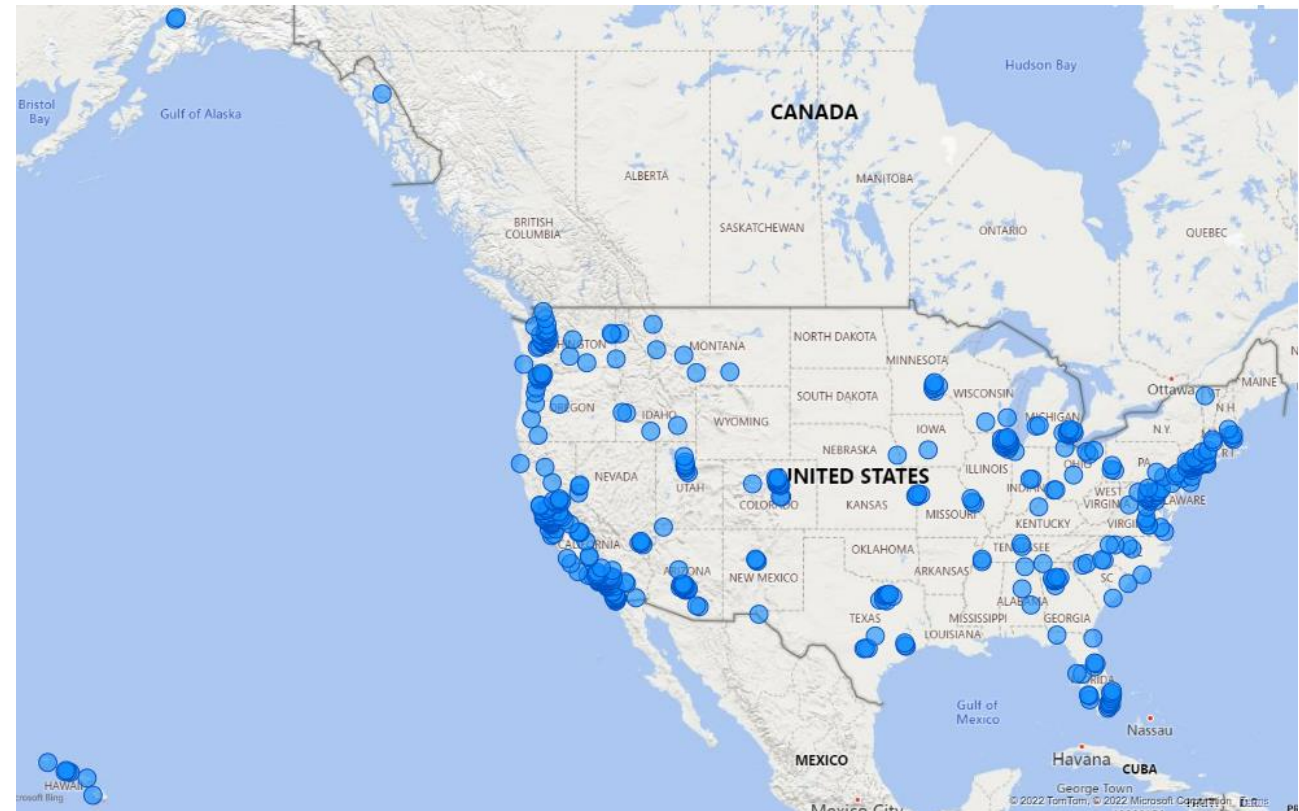
## Michael Ersevim

# Power BI: Heat Map and Spatial Map



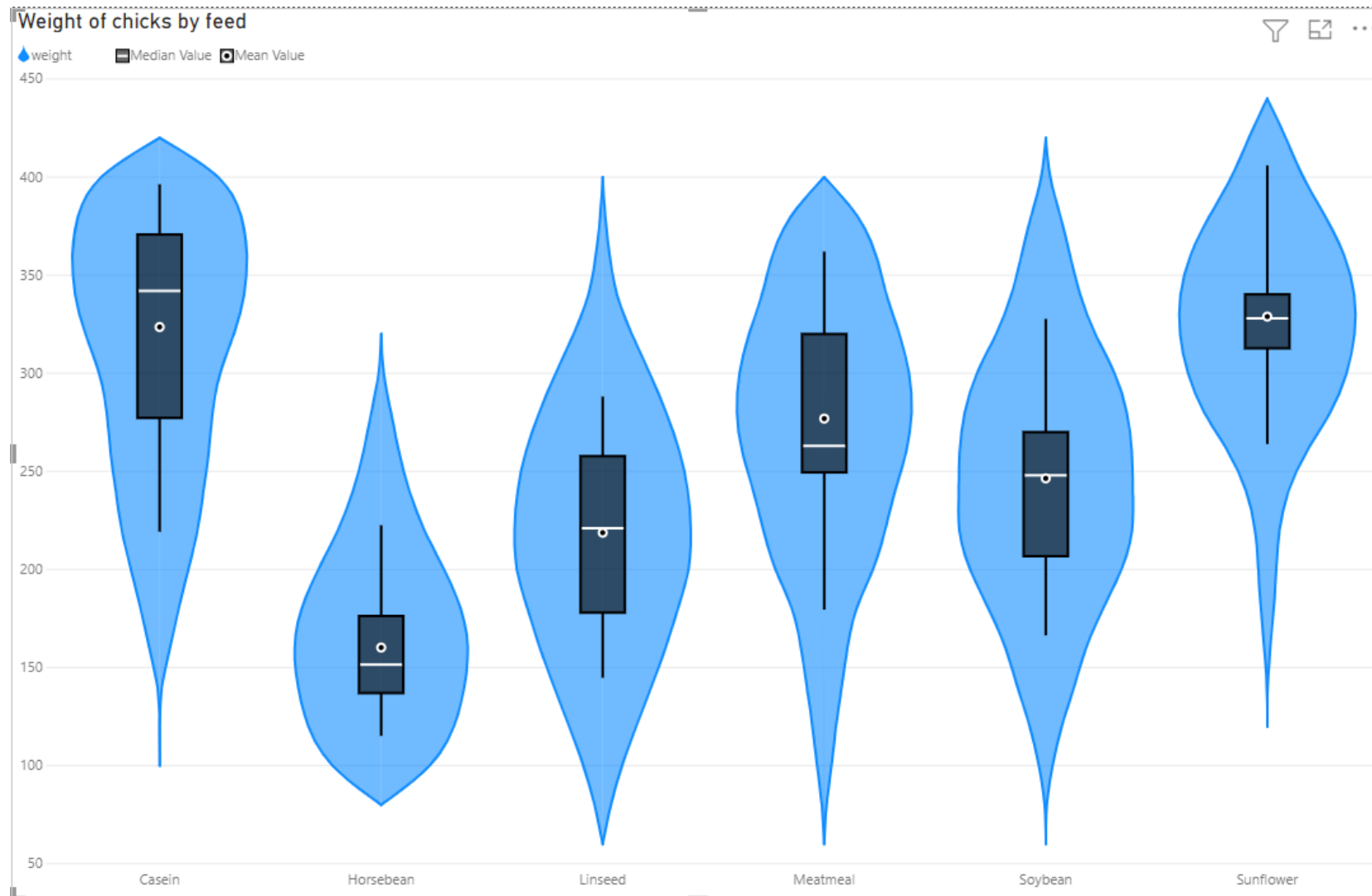
Left:  
A Heat  
map of  
top NBA  
players  
and their  
common  
game  
statistics  
relative to  
each  
column's  
high/low  
values  
regardless  
of value

Below: A spatial map of Costco  
locations across the United States

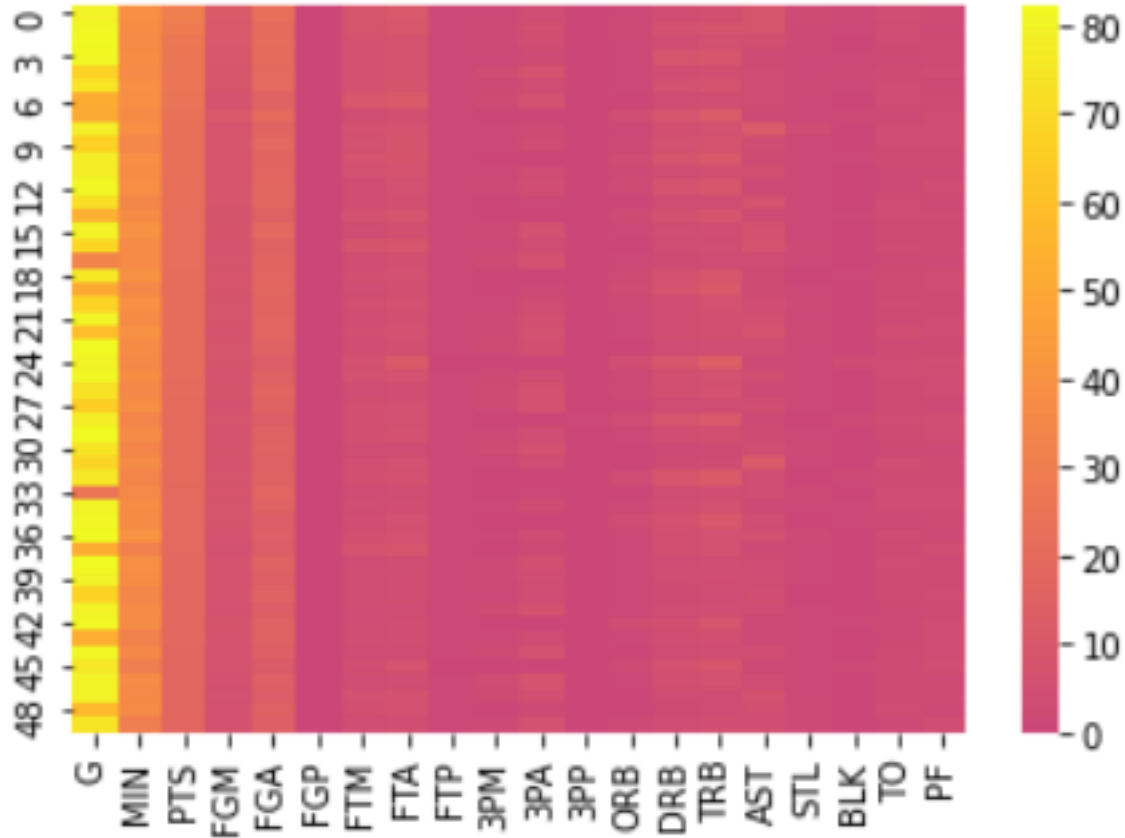


# Power BI: Violin Plot

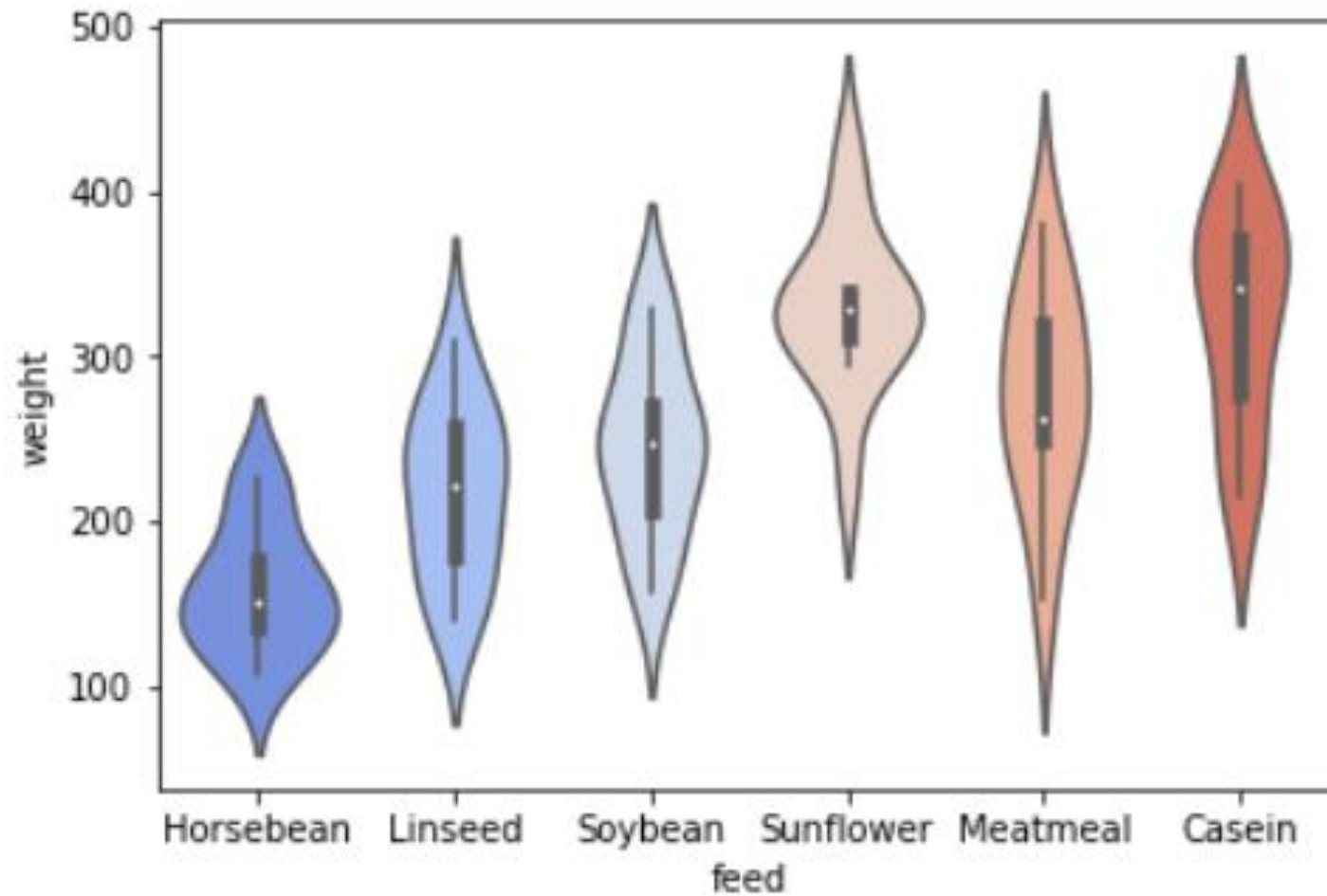
Violin plot of weight (in grams) of different groups of chicks being fed a certain feed type



# Python: Heat Map and Spatial Map



# Python: Violin Plot



# Python: CODE for generating graphs

## # DSC 640 - Michael Ersevrim - Weeks 9&10

```
In [1]: # Call in libraries
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

```
In [2]: #Read in data
df = pd.read_csv(r'C:\\Users\\Kate\\Documents\\Bellevue DS classes\\DSC640\\chick_weights.csv')
df.head()
```

Out[2]:

	id	weight	sex	feed
0	1.0	179.0	male	Horsebean
1	2.0	160.0	male	Horsebean
2	3.0	136.0	female	Horsebean
3	4.0	227.0	male	Horsebean
4	5.0	217.0	female	Horsebean

```
In [3]: #Making violin plot of weights of chick based of feedstock
Ax = sns.violinplot(x="feed", y="weight", data=df, palette="coolwarm")
```

# Python: CODE for generating graphs

```
In [30]: #Read in data
df2 = pd.read_excel('C:\\Users\\Kate\\Documents\\Bellevue DS classes\\DSC640\\ppg2008.xlsx')
df3=df2.iloc[:,1:]
df3.head()
```

Out[30]:

	G	MIN	PTS	FGM	FGA	FGP	FTM	FTA	FTP	3PM	3PA	3PP	ORB	DRB	TRB	AST	STL	BLK	TO	PF
0	79	38.6	30.2	10.8	22.0	0.491	7.5	9.8	0.765	1.1	3.5	0.317	1.1	3.9	5.0	7.5	2.2	1.3	3.4	2.3
1	81	37.7	28.4	9.7	19.9	0.489	7.3	9.4	0.780	1.6	4.7	0.344	1.3	6.3	7.6	7.2	1.7	1.1	3.0	1.7
2	82	36.2	26.8	9.8	20.9	0.467	5.9	6.9	0.856	1.4	4.1	0.351	1.1	4.1	5.2	4.9	1.5	0.5	2.6	2.3
3	81	37.7	25.9	9.6	20.0	0.479	6.0	6.7	0.890	0.8	2.1	0.359	1.1	7.3	8.4	2.4	0.8	0.8	1.9	2.2
4	67	36.2	25.8	8.5	19.1	0.447	6.0	6.9	0.878	2.7	6.7	0.404	0.7	4.4	5.1	2.7	1.0	1.4	2.5	3.1

```
In [6]: heatmap = sns.heatmap(data=df3, cmap="plasma", center = 0 , annot = False)
plt.show()
```



# Python: CODE for generating graphs

```
In [7]: dfgpd = pd.read_csv(r'C:\\Users\\Kate\\Documents\\Bellevue DS classes\\DSC640\\costcos-geocoded.csv')
dfgpd.head()
```

Out[7]:

	Address	City	State	Zip Code	Latitude	Longitude
0	1205 N. Memorial Parkway	Huntsville	Alabama	35801-5930	34.743095	-86.600955
1	3650 Galleria Circle	Hoover	Alabama	35244-2346	33.377649	-86.812420
2	8251 Eastchase Parkway	Montgomery	Alabama	36117	32.363889	-86.150884
3	5225 Commercial Boulevard	Juneau	Alaska	99801-7210	58.359200	-134.483000
4	330 West Dimond Blvd	Anchorage	Alaska	99515-1950	61.143266	-149.884217

```
In [29]: import plotly.express as px

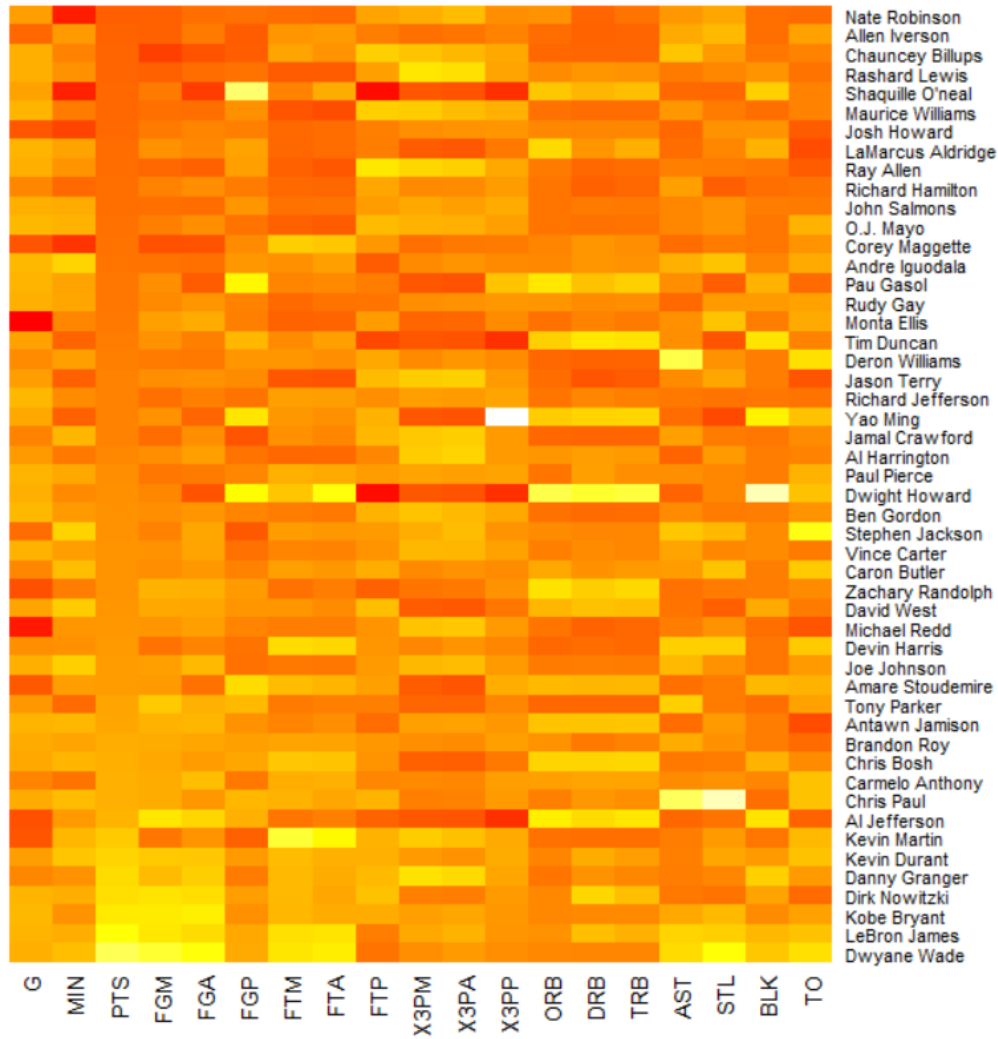
fig = px.density_mapbox(dfgpd, lat='Latitude', lon='Longitude', radius=1,
                        center=dict(lat=38, lon=-96), zoom=2.4,
                        mapbox_style="stamen-terrain")

fig.show()
```

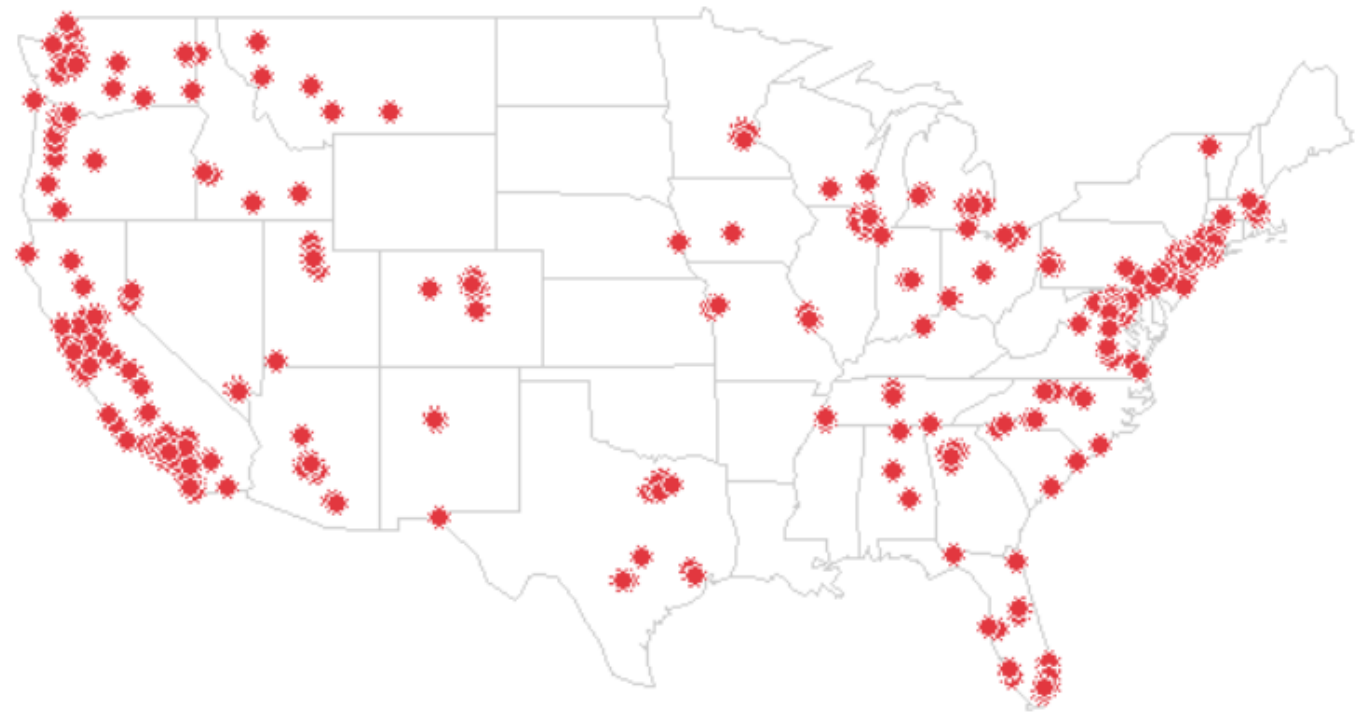


# R: Heat Map and Spatial Map

Heat map of NBA player's stats

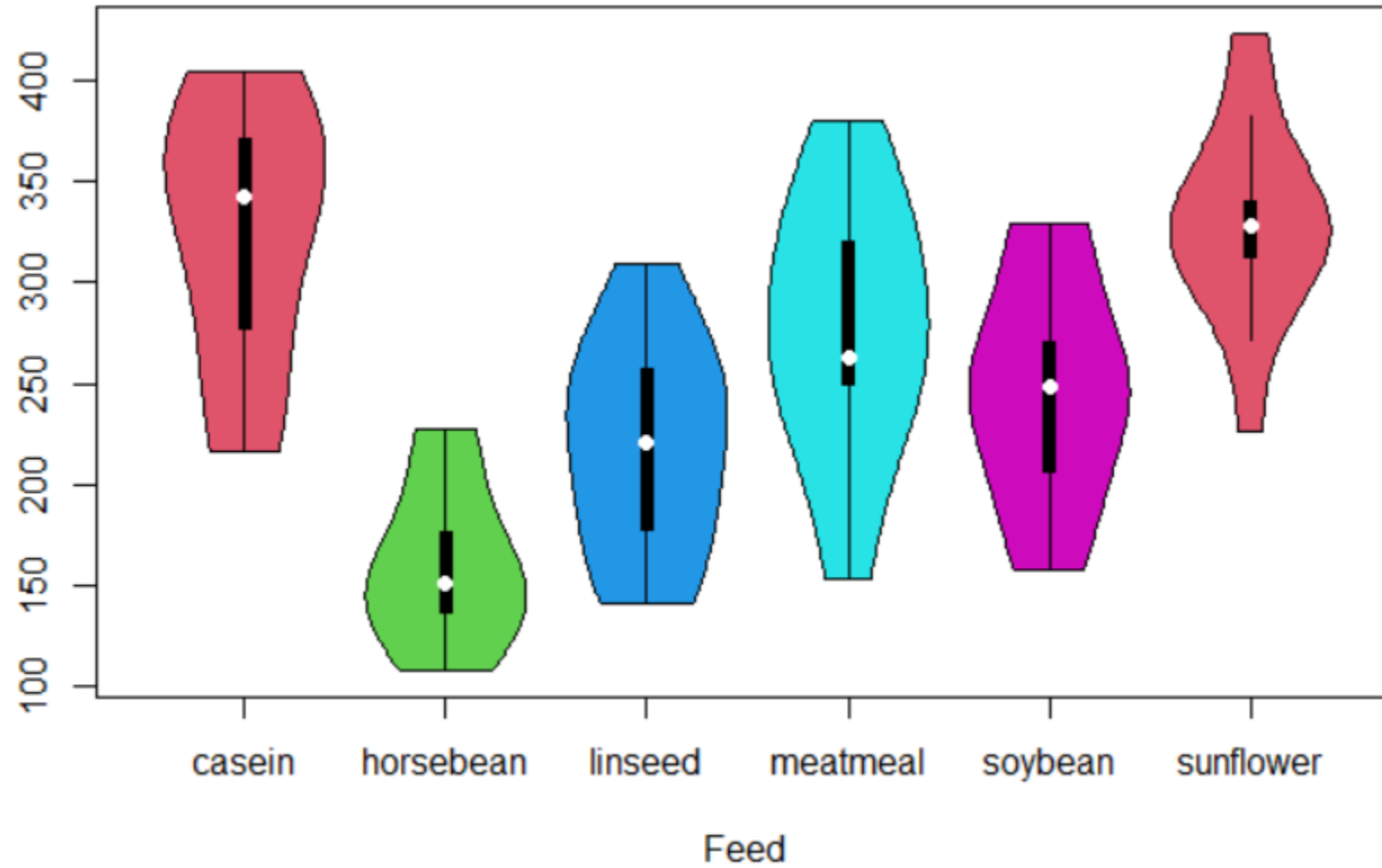


Spatial map of CostCo locations across the United States



# R: Violin Plot

Violin Plot of Chick weight in grams by feed type



# R: CODE for generating graphs

```
1 setwd("C:/Users/Kate/Documents/Bellevue DS classes/DSC640")
2
3 # Load the data for making a heat map
4 bball <- read.csv('ppg2008.csv', sep=";", header=TRUE)
5 row.names(bball) <- bball$Name #Defines the players' names as a row
6 bball <- bball[,2:20] #Now just the data
7 bball_matrix <- data.matrix(bball) #and is put in a matrix
8 bball_heatmap <- heatmap(bball_matrix, Rowv=NA, Colv=NA, col=heat.colors(256),
9                           scale='column', margins=c(5,10))
10
11 #Now for a spatial map
12 library(maps)
13
14 costco <- read.csv('costcos-geocoded.csv', sep=";", header=TRUE)
15 map(database="state", col="#cccccc")
16 symbols(costco$Longitude, costco$Latitude, bg="#e2373f", fg="#ffffff", lwd=0.5,
17          circles=rep(1, length(costco$Longitude)), inches=0.05, add=TRUE)
18
19 #And lastly, a violin plot
20 library(vioplot)
21 data <- chickwts #this data is apparently already loaded in the library
22
23 vioplot(data$weight ~ data$feed, col = 2:length(levels(data$feed)),
24          xlab = "Feed", ylab = "weight") #As directed by the Yau textbook
25
```