Ryan Long
DSC640
6.2 Exercises: Charts

Dataset used:
education.csv
disney.xlsx

Summary
Histograms and Boxplots were a good refresher in Python and R, and new to me in Power BI. I much rather preferred them in the programming languages due to familiarity. I also think they would be used there more frequently during EDA. I think the inverse would be true for the bullet chart as I would set it more as a dashboard / KPI tracking which would get built out in Power BI. For the new charts, I did a Sankey in Power BI using a dataset of Disney Expenses. I used the same dataset in Python and R and created Violin charts. They were an option in a previous exercise, but I did not create them.

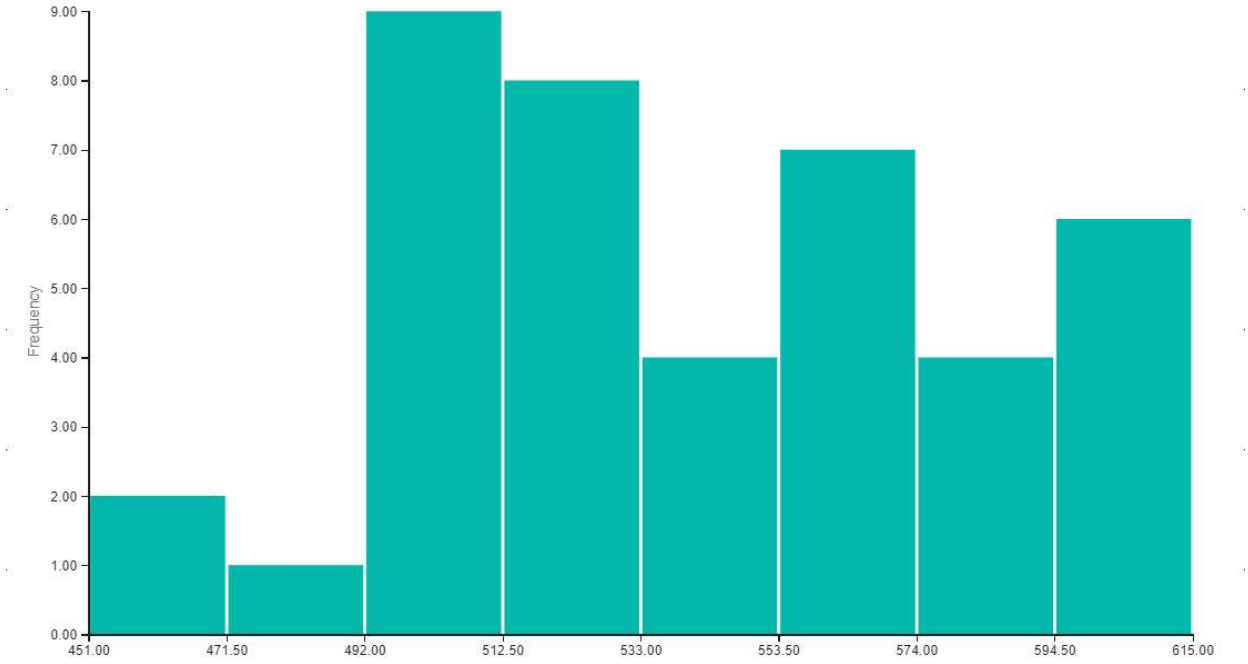
The following pages contain:
Power BI:
Histogram
Box Plot
Bullet Chart
Sankey

Python:
Histogram
Box Plot
Bullet Chart
Violin Chart

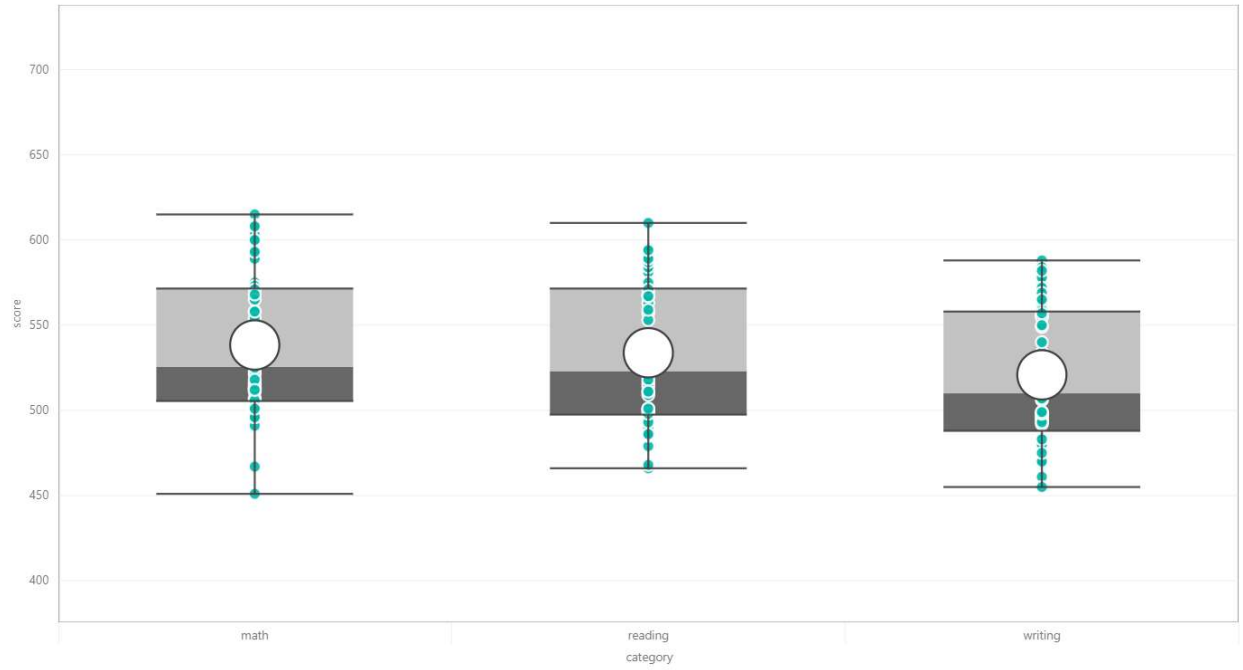
R:
Histogram
Box Plot
Bullet Chart
Violin Chart

Appendix
Code support for both Python and R notebooks

Power BI - Histogram: Distribution of SAT Math Scores

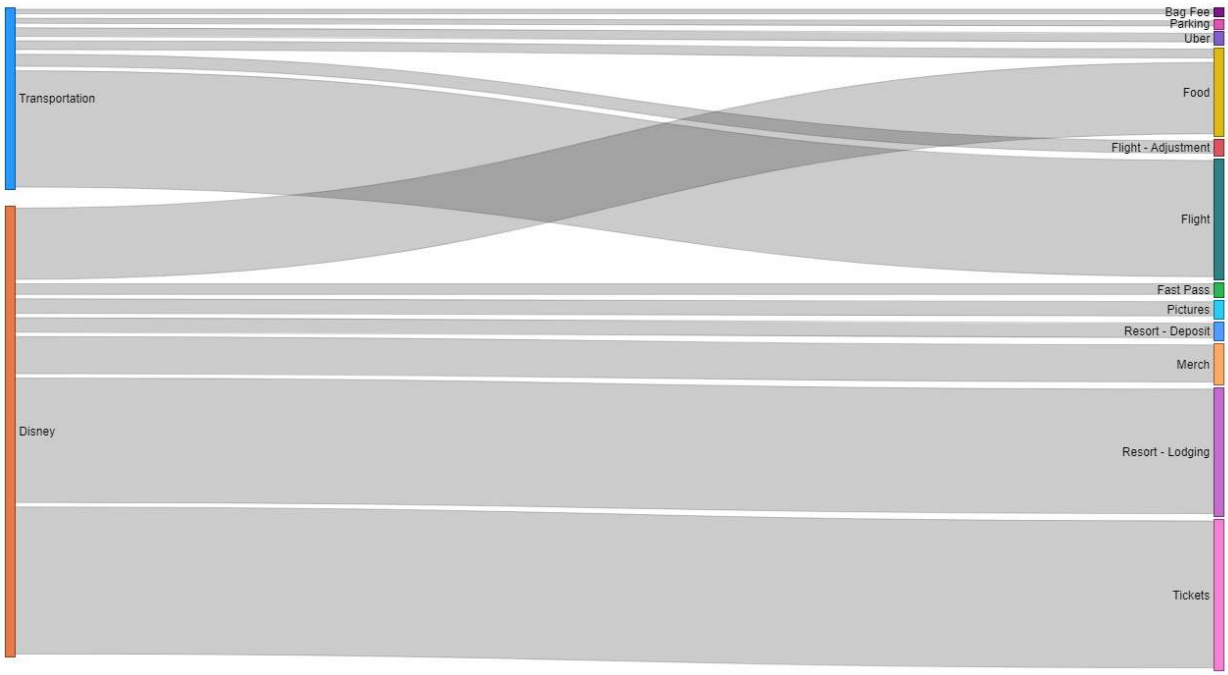


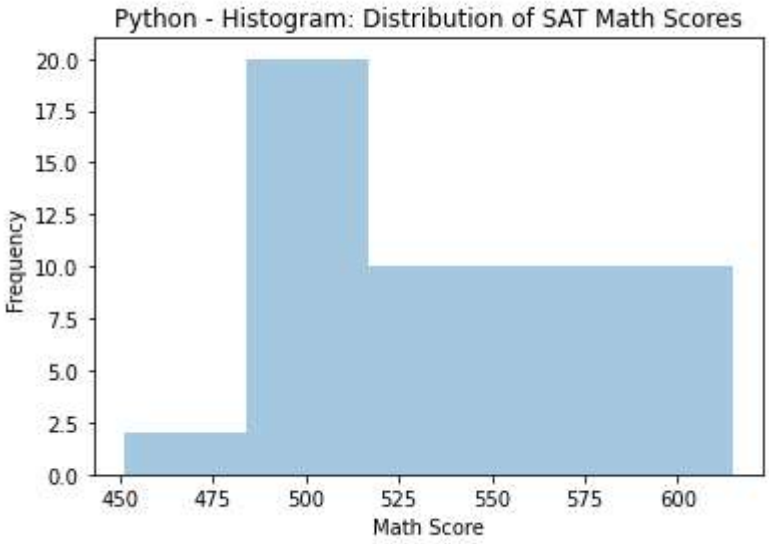
Power BI - Box Plot: SAT Scores

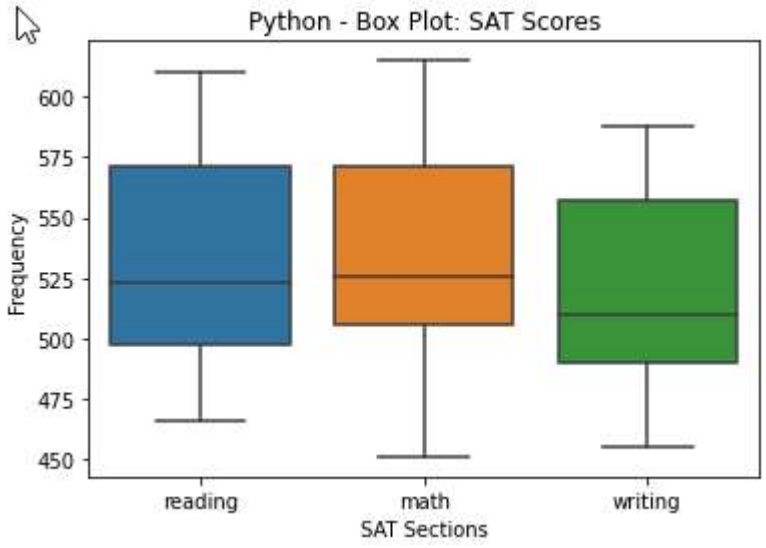


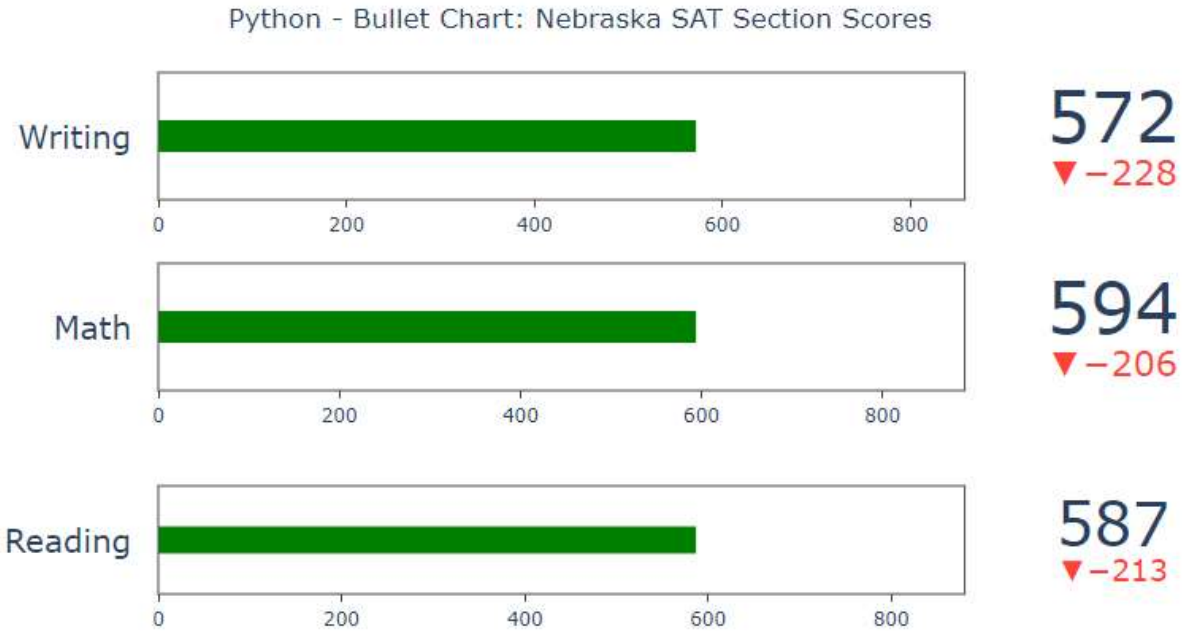
Power BI - Bullet Chart: Nebraska SAT Scores

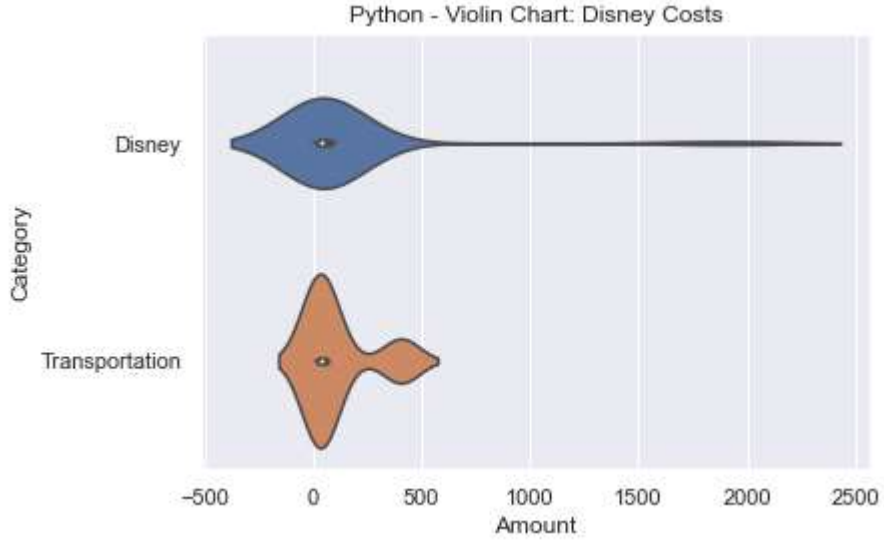
Power BI - Sankey: Disney Expenses

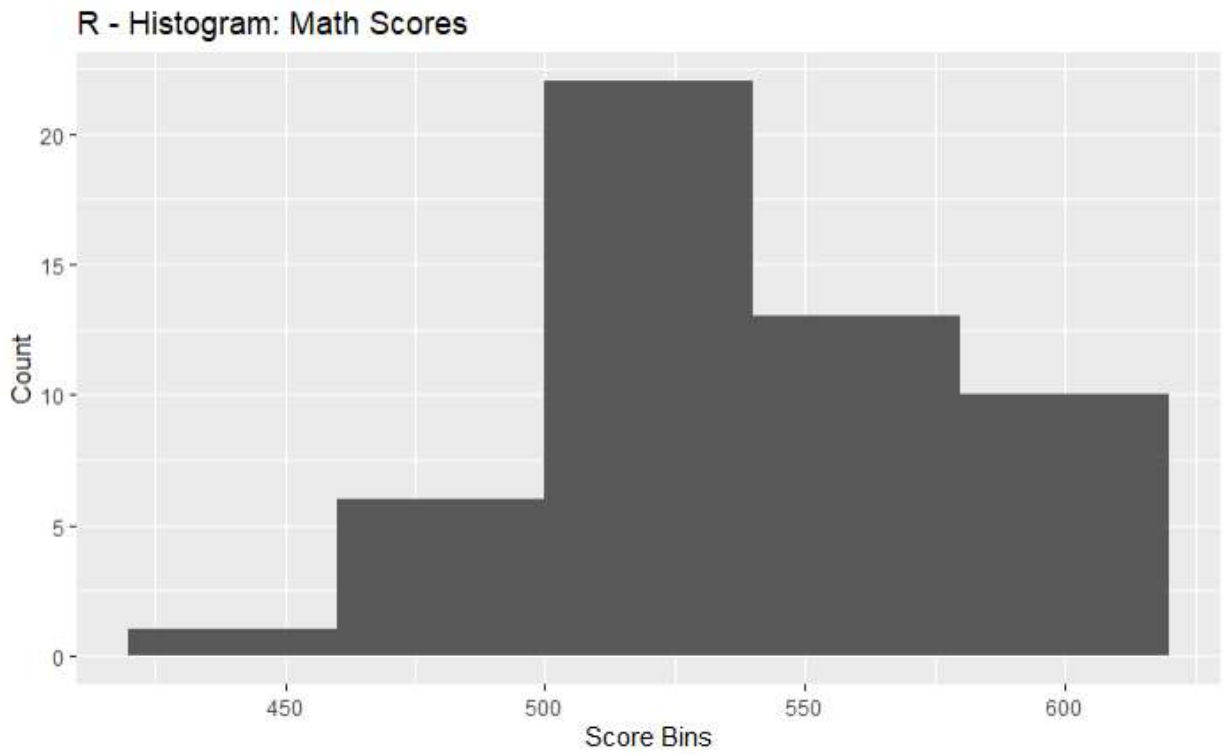


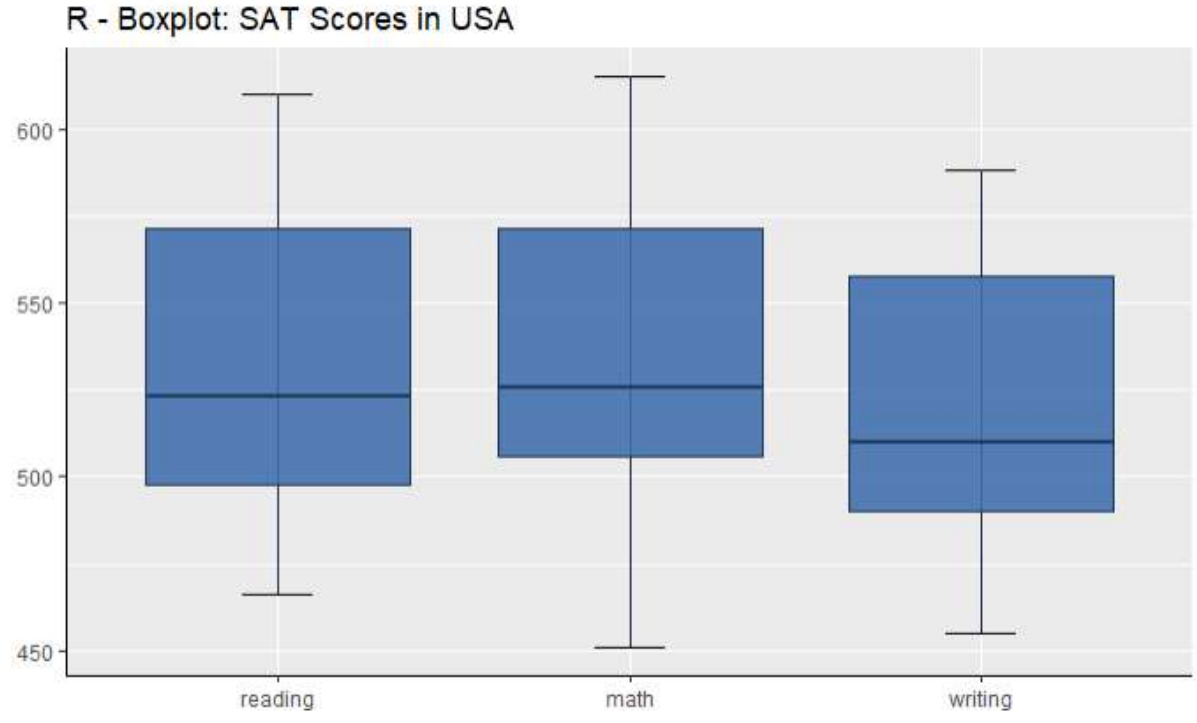

Python - Histogram: Distribution of SAT Math Scores
Frequency
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
450
475
500
525
550
575
600
Math Score


Python - Box Plot: SAT Scores
Frequency
600
575
550
525
500
475
450
reading
math
writing
SAT Sections

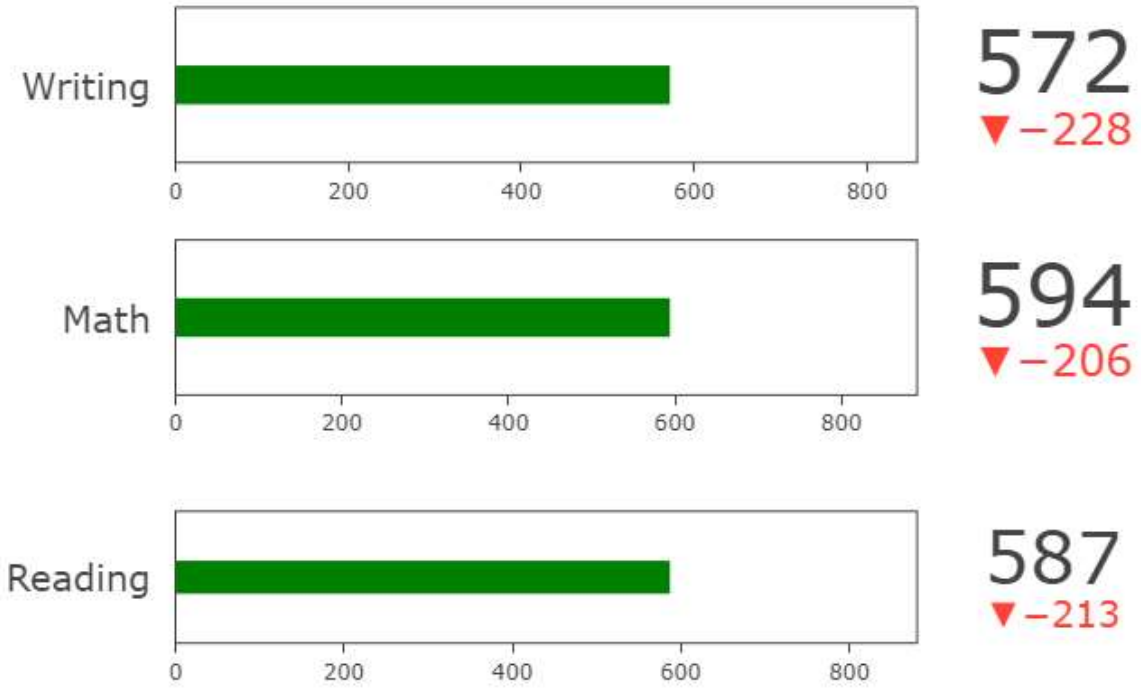

Python - Bullet Chart: Nebraska SAT Section Scores
Writing
Math
Reading
0
200
400
600
800
572
▼−228
594
▼−206
587
▼−213


Python - Violin Chart: Disney Costs
Disney
Transportation
Category
−500
0
500
1000
1500
2000
2500
Amount

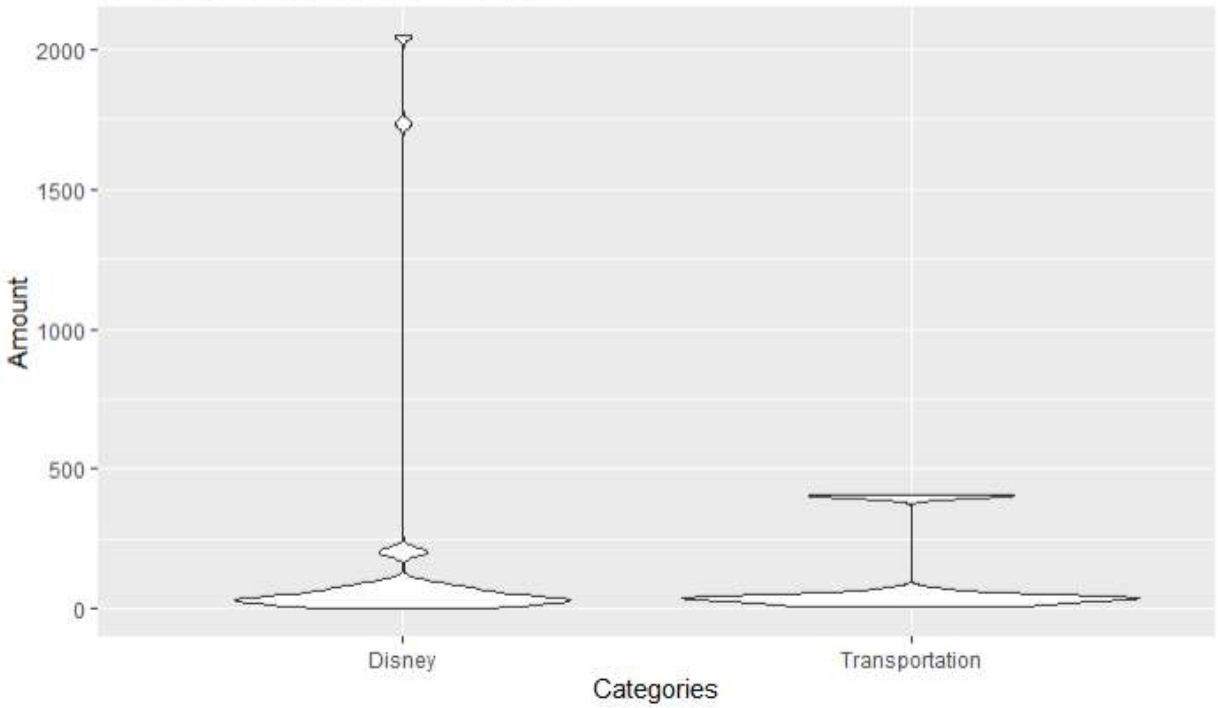

R - Histogram: Math Scores
Count
20
15
10
5
0
450
500
550
600
Score Bins


R - Boxplot: SAT Scores in USA
600
550
500
450
reading
math
writing

R - Bullet Chart: Nebraska SAT Section Scores



R - Violin Chart: Disney Expenses



APPENDIX

In [1]:
```
#load libraries
import pandas as pd
import seaborn as sns

import numpy as np
import matplotlib.pyplot as plt

import plotly.graph_objects as go

#import squarify
#import matplotlib.ticker as plticker # for plot ticks
```

In [ ]:
```
```

In [2]:
```
#import data as dataframe

bdf = pd.read_csv('education.csv')
ddf = pd.read_excel('disney.xlsx')
```

In [3]:
```
#bdf.head()
ddf.head()
```

Out[3]:

	Category	Sub Category	Amount
0	Disney	Fast Pass	38.36
1	Disney	Fast Pass	63.92
2	Disney	Fast Pass	42.60
3	Disney	Food	2.25
4	Disney	Food	2.25

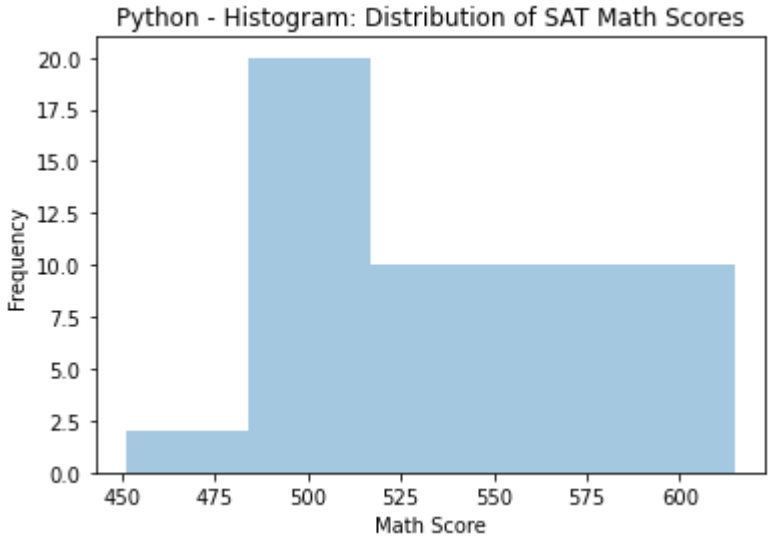
In [4]:
```
#bdf.info()
#ddf.info()
```

Histogram

In [5]:
```
# Plot the histogram thanks to the distplot function
sns.distplot( a=bdf["math"], hist=True, kde=False, rug=False )
plt.xlabel("Math Score")
plt.ylabel("Frequency")
plt.title("Python - Histogram: Distribution of SAT Math Scores")
```

```
C:\Users\longr\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future v
ersion. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```

Out[5]:
```
Text(0.5, 1.0, 'Python - Histogram: Distribution of SAT Math Scores')
```



Box Plot

In [6]:
```
bpdf = bdf.drop(['percent_graduates_sat', 'pupil_staff_ratio', 'dropout_rate'], a
```

In [7]:
```
bpdf
```

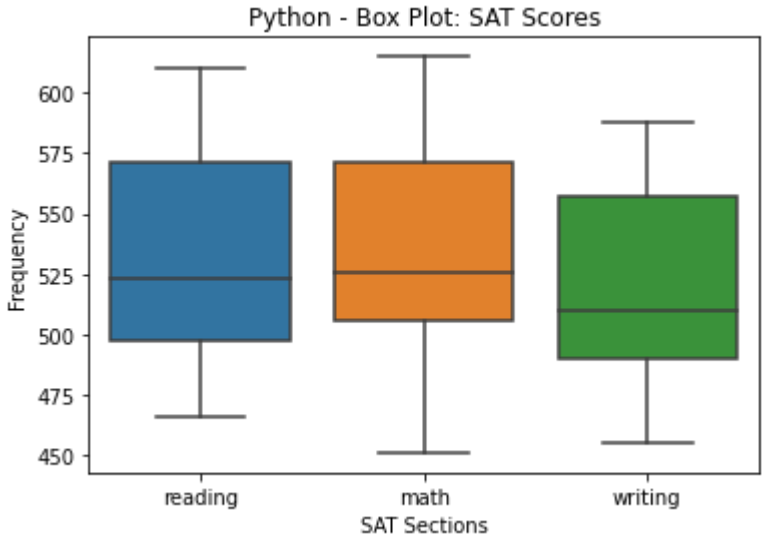
Out[7]:

	state	reading	math	writing
0	United States	501	515	493
1	Alabama	557	552	549
2	Alaska	520	516	492
3	Arizona	516	521	497
4	Arkansas	572	572	556
5	California	500	513	498
6	Colorado	568	575	555
7	Connecticut	509	513	512
8	Delaware	495	498	484
9	District of Columbia	466	451	461
10	Florida	497	498	480
11	Georgia	490	491	479
12	Hawaii	479	502	469
13	Idaho	541	540	520
14	Illinois	588	604	583
15	Indiana	496	507	480
16	Iowa	610	615	588
17	Kansas	581	589	564
18	Kentucky	573	573	561
19	Louisiana	563	558	555
20	Maine	468	467	455
21	Maryland	500	502	495
22	Massachusetts	514	526	510
23	Michigan	584	603	575
24	Minnesota	595	609	578
25	Mississippi	567	554	559
26	Missouri	595	600	584
27	Montana	541	542	519
28	Nebraska	587	594	572
29	Nevada	501	505	479
30	New Hampshire	523	523	510
31	New Jersey	496	513	496
32	New Mexico	553	546	534
33	New York	485	502	478

	state	reading	math	writing
34	North Carolina	495	511	480
35	North Dakota	590	593	566
36	Ohio	537	546	523
37	Oklahoma	575	571	557
38	Oregon	523	525	499
39	Pennsylvania	493	501	483
40	Rhode Island	498	496	494
41	South Carolina	486	496	470
42	South Dakota	589	600	569
43	Tennessee	571	565	565
44	Texas	486	506	475
45	Utah	559	558	540
46	Vermont	518	518	506
47	Virginia	511	512	498
48	Washington	524	531	507
49	West Virginia	511	501	499
50	Wisconsin	594	608	582
51	Wyoming	567	568	550

In [8]:
```
sns.boxplot(data=bpdf)
plt.xlabel("SAT Sections")
plt.ylabel("Frequency")
plt.title("Python - Box Plot: SAT Scores")
```

Out[8]:
```
Text(0.5, 1.0, 'Python - Box Plot: SAT Scores')
```



Bullet Chart

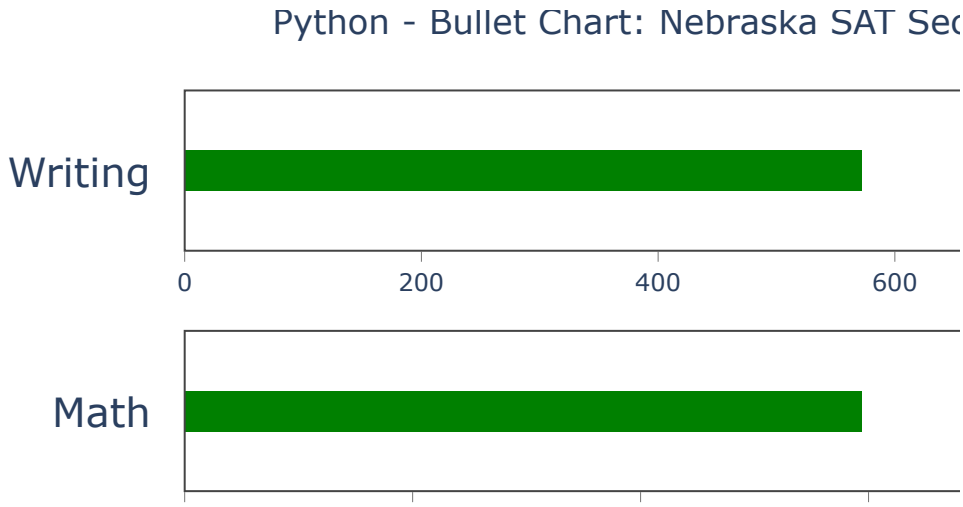
In [9]:
```
bcdf=bpdf.loc[(bpdf['state'] == 'Nebraska') |  (bpdf['state'] == 'United States')]
```

In [10]:
```
bcdf
```

Out[10]:

	state	reading	math	writing
0	United States	501	515	493
28	Nebraska	587	594	572

In [11]:
```
fig = go.Figure()

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 587,
    delta = {'reference': 800},
    domain = {'x': [0.25, 1], 'y': [0.08, 0.25]},
    title = {'text': "Reading"},
    gauge = {'shape': "bullet"}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 594,
    delta = {'reference': 800},
    domain = {'x': [0.25, 1], 'y': [0.4, 0.6]},
    title = {'text': "Math"},
    gauge = {'shape': "bullet"}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 572,
    delta = {'reference': 800},
    domain = {'x': [0.25, 1], 'y': [0.7, 0.9]},
    title = {'text' :"Writing"},
    gauge = {'shape': "bullet",}))

fig.update_layout(title={
        'text': "Python - Bullet Chart: Nebraska SAT Section Scores",
        'y':1,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},height = 400 , margin = {'t':0, 'b':0, 'l':0})

fig.show()
```



Violin

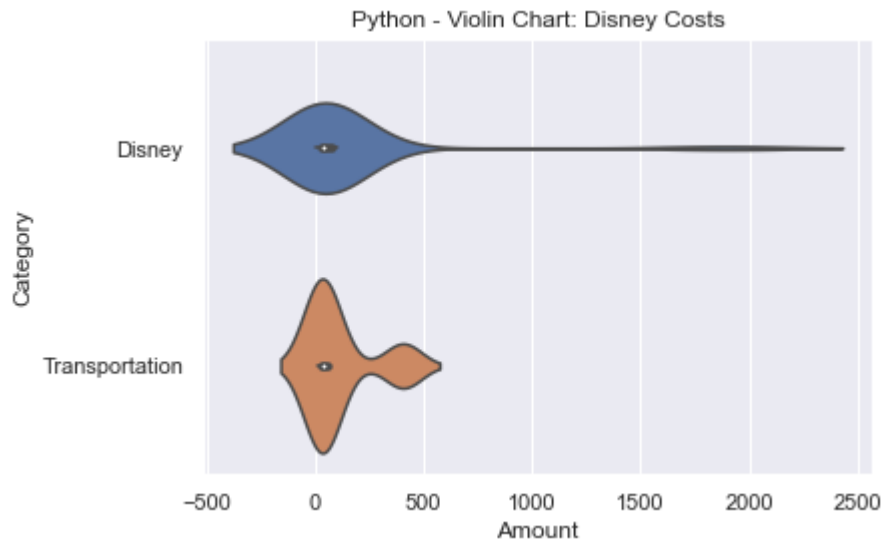
In [19]:
```
ddf
```

Out[19]:

	Category	Sub Category	Amount
0	Disney	Fast Pass	38.36
1	Disney	Fast Pass	63.92
2	Disney	Fast Pass	42.60
3	Disney	Food	2.25
4	Disney	Food	2.25
...	...	...	...
57	Transportation	Parking	70.00
58	Transportation	Uber	53.01
59	Transportation	Uber	10.40
60	Transportation	Uber	9.14
61	Transportation	Uber	45.74

62 rows × 3 columns

In [33]:
```
# library & dataset
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="darkgrid")
# plot
sns.violinplot(x=ddf["Amount"],y=ddf["Category"])

plt.title("Python - Violin Chart: Disney Costs")
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Week 11 & 12

Code

Hide

```
#load libraries
library(ggplot2)
```

```
Use suppressPackageStartupMessages() to eliminate package startup messages
```

Hide

```
library(dplyr)
```

```


Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

Hide

```
library(tidyr)
library(tidyverse)
```

```
Registered S3 methods overwritten by 'dbplyr':
  method         from
  print.tbl_lazy
  print.tbl_sql
-- Attaching packages ---------------------------------------------------------------------
--------------------------------------------------------------- tidyverse 1.3.1 --
v tibble  3.1.6     v stringr 1.4.0
v readr   2.1.0     v forcats 0.5.1
v purrr   0.3.4
-- Conflicts ------------------------------------------------------------------------------
-------------------------------------------------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

Hide

```
library(hrbrthemes)
```

```
Registering Windows fonts with R
NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
      Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
      if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

Hide

```
library(pivottabler)
library(areaplot)
library(plotly)
```

```


Attaching package: ‘plotly’

The following object is masked from ‘package:ggplot2’:

    last_plot

The following object is masked from ‘package:stats’:

    filter

The following object is masked from ‘package:graphics’:

    layout
```

Hide

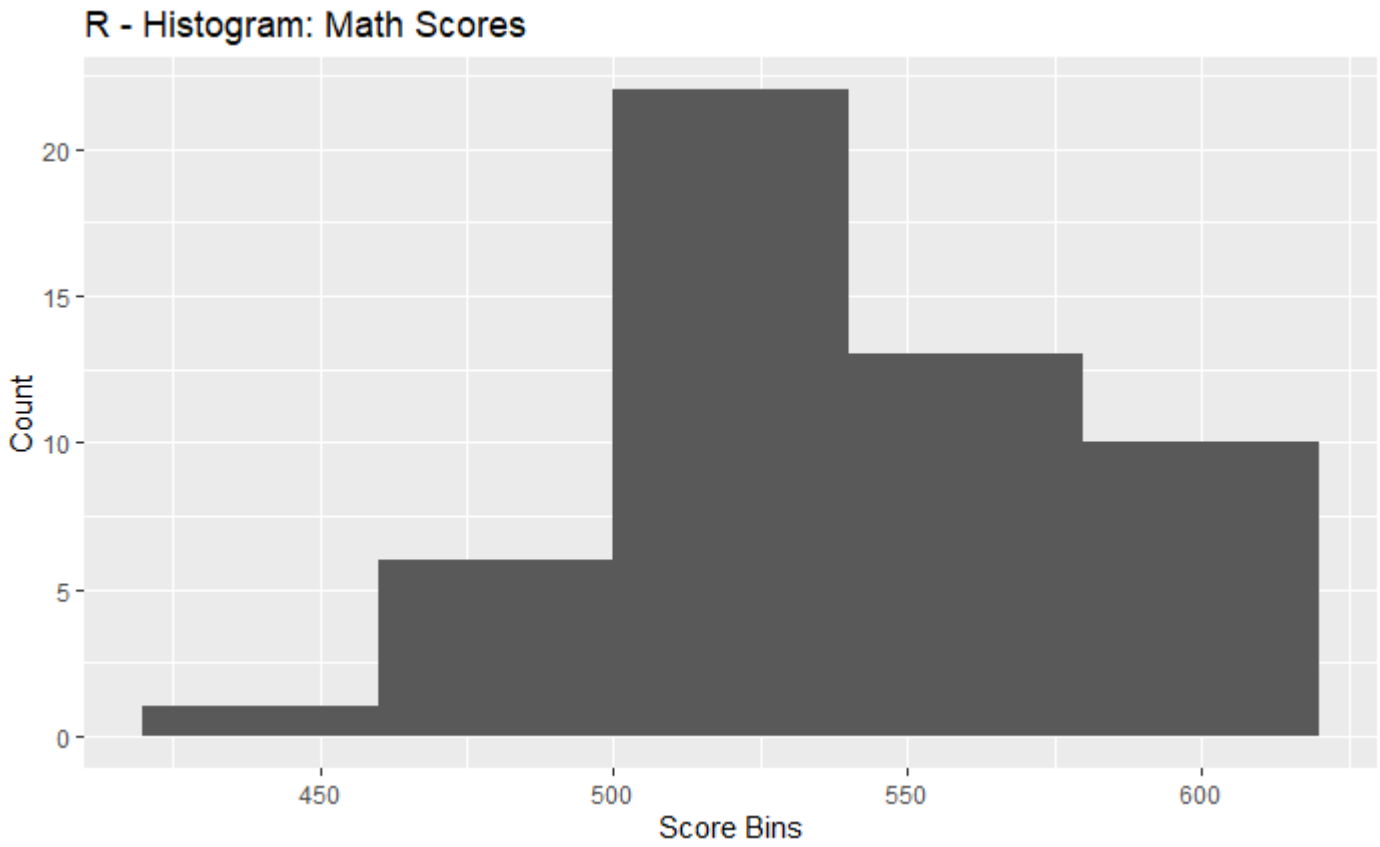
```
library(readxl)
```

Hide

```
#import data
satdf = read.csv("C://Users//longr//Documents//DSC 640//Weeks 11 & 12//Exercises 6.2//education.
csv")
```

Hide

```
#Histogram
ggplot(satdf, aes(x=math)) +
  geom_histogram(binwidth=40)+
  xlab("Score Bins")+ylab("Count")+
  ggtitle("R - Histogram: Math Scores")
```



Hide

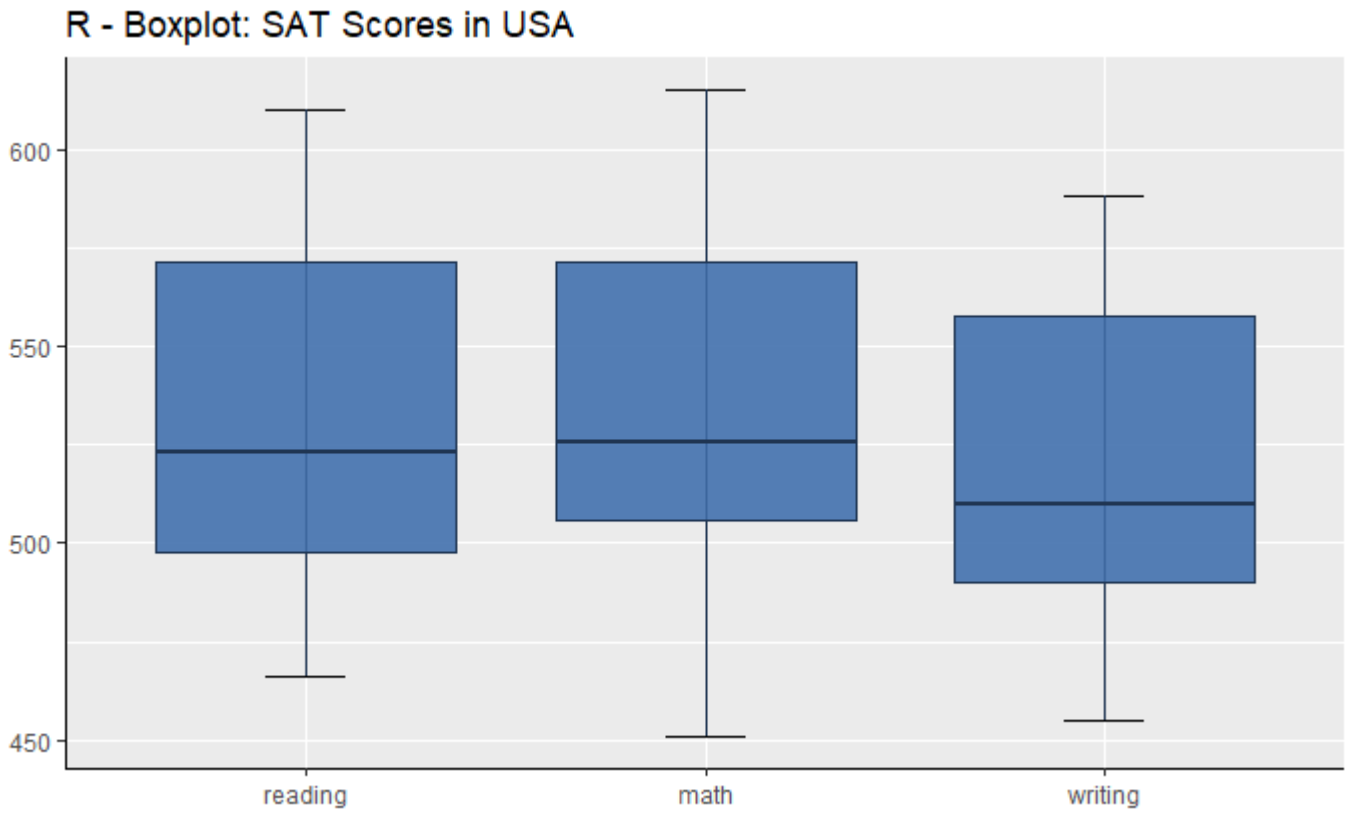
```
#Box Plot
bpdf <- satdf[ -c(1,5:7)]#new df for boxplot
stacked_bpdf <- stack(bpdf)
```

Hide

```
#boxplot(stacked_bpdf$values ~ stacked_bpdf$ind,
#        col = rainbow(ncol(bpdf)))
```

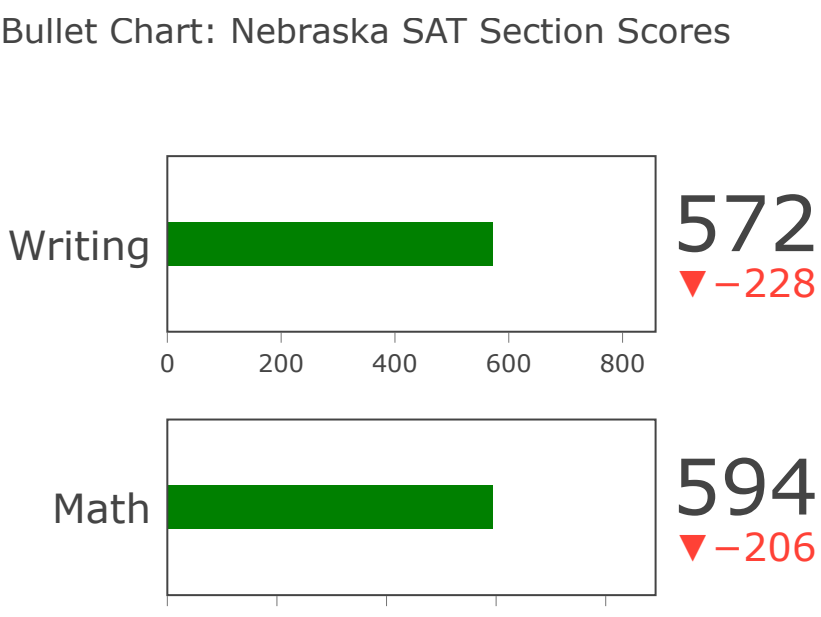
Hide

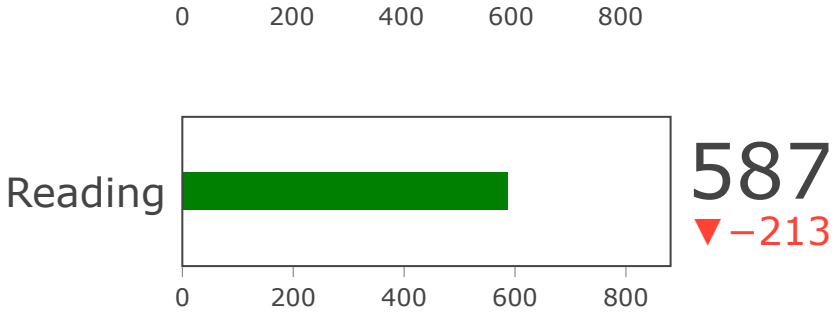
```
#https://r-coder.com/boxplot-r/

ggplot(stacked_bpdf, aes(x = ind, y = values)) +
       stat_boxplot(geom = "errorbar",width = 0.2) +
       geom_boxplot(fill = "#4271AE", colour = "#1F3552",alpha = 0.9, outlier.colour = "red") +
       scale_y_continuous(name = "") +
       scale_x_discrete(name = "") +
       ggtitle("R - Boxplot: SAT Scores in USA") +
       theme(axis.line = element_line(colour = "black", size = 0.25))
```


R - Boxplot: SAT Scores in USA
600
550
500
450
reading
math
writing

Hide

```
#Bullet Chart
fig <- plot_ly()

fig <- fig %>%
  add_trace(
    type = "indicator",
    mode = "number+gauge+delta",
    value = 587,
    delta = list(reference = 800),
    domain = list(x = c(0.25, 1), y = c(0.08, 0.25)),
    title =list(text = "Reading"),
    gauge = list(shape = "bullet"))
fig <- fig %>%
  add_trace(
    type = "indicator",
    mode = "number+gauge+delta",
    value = 594,
    delta = list(reference = 800),
    domain = list(x = c(0.25, 1), y = c(0.4, 0.6)),
    title = list(text = "Math"),
    gauge = list(shape = "bullet"))
fig <- fig %>%
  add_trace(
    type =  "indicator",
    mode = "number+gauge+delta",
    value = 572,
    delta = list(reference = 800 ),
    domain = list(x = c(0.25, 1), y = c(0.7, 0.9)),
    title = list(text = "Writing"),
    gauge = list(shape = "bullet"))

fig<- fig%>%
  layout(title = 'R - Bullet Chart: Nebraska SAT Section Scores', plot_bgcolor = "#e5ecf6")

fig
```





Hide

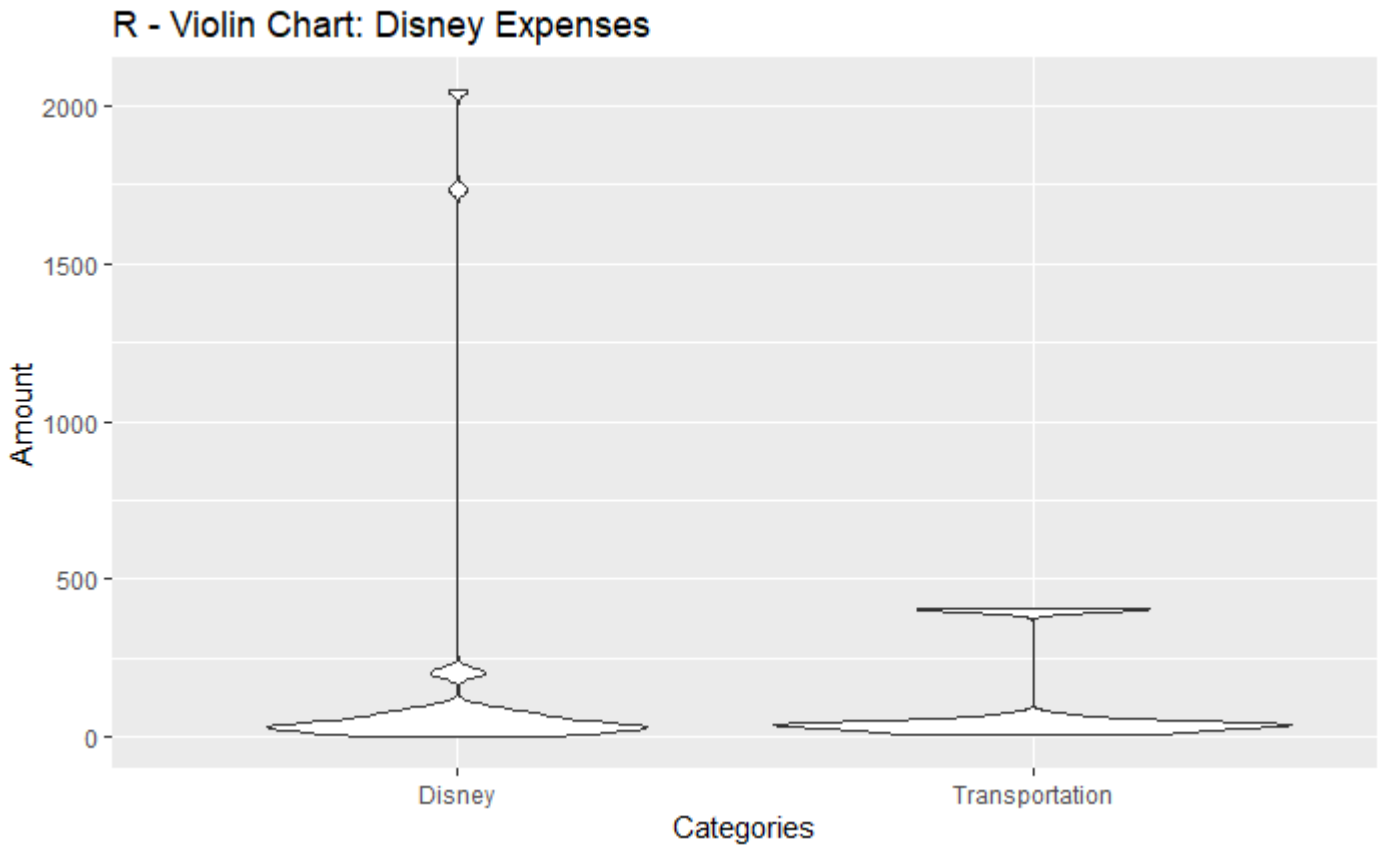
```
NA
```

Hide

```
#import data
ddf = read_excel("C://Users//longr//Documents//DSC 640//Weeks 11 & 12//Exercises 6.2//disney.xls
x")
```

Hide

```
# Most basic violin chart
ggplot(ddf, aes(x=Category, y=Amount)) + # fill=name allow to automatically dedicate a color for
each group
  geom_violin()+
    xlab("Categories")+ylab("Amount")+
    ggtitle("R - Violin Chart: Disney Expenses")
```


R - Violin Chart: Disney Expenses
2000
1500
1000
500
0
Amount
Disney
Transportation
Categories