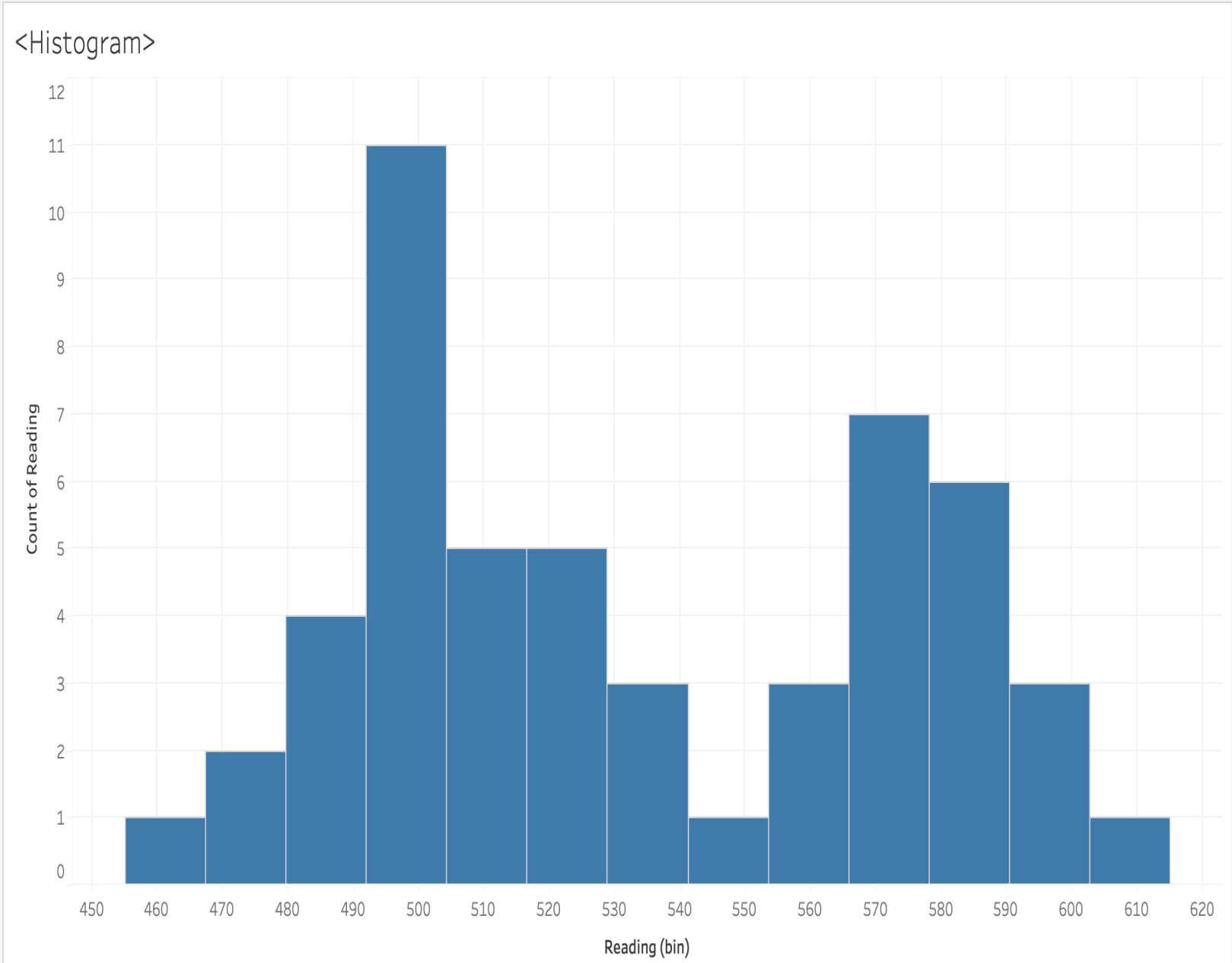
Charts in Tableau

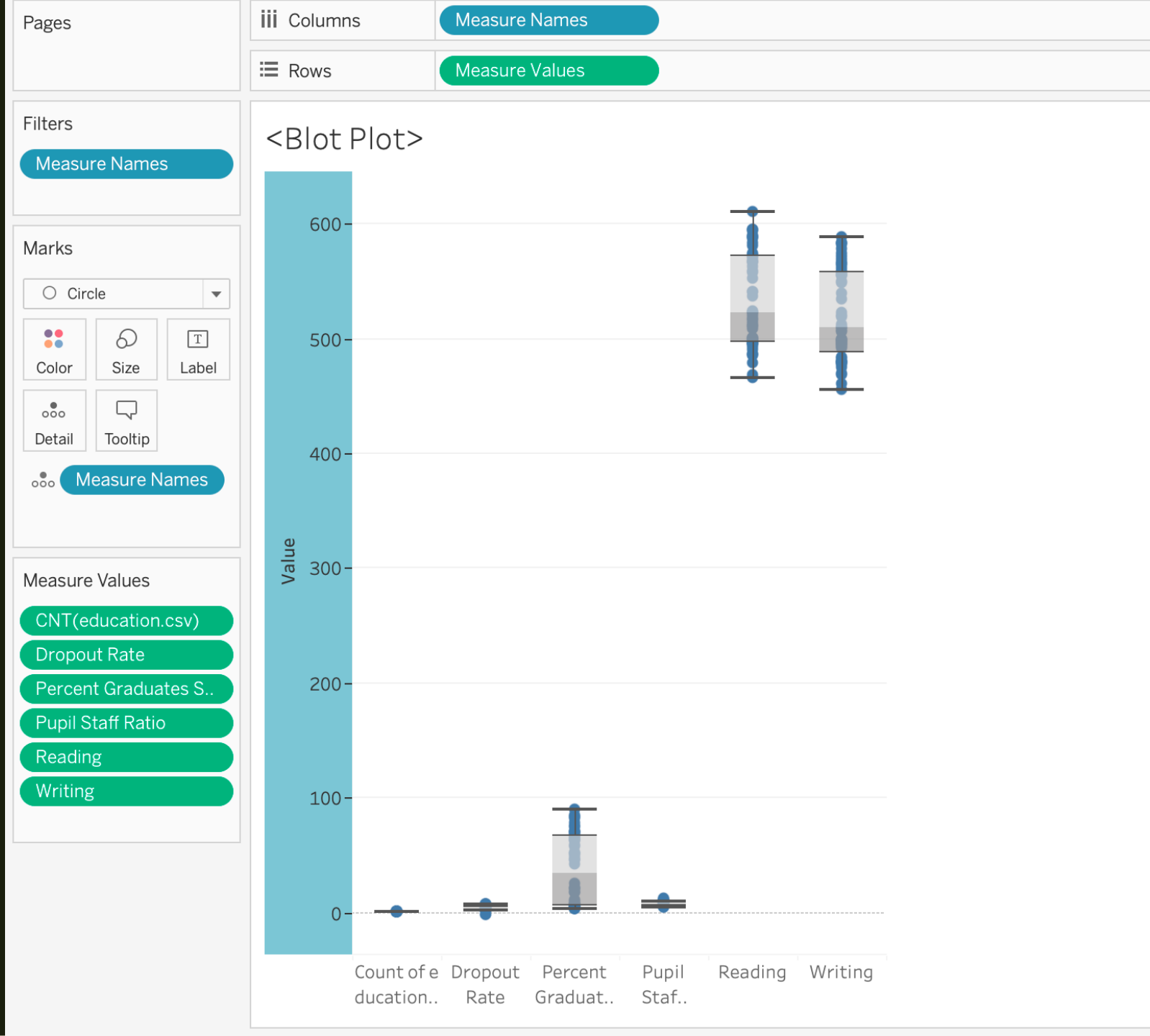
Histogram

Columns: Reading (bin)

Rows: CNT(Reading)

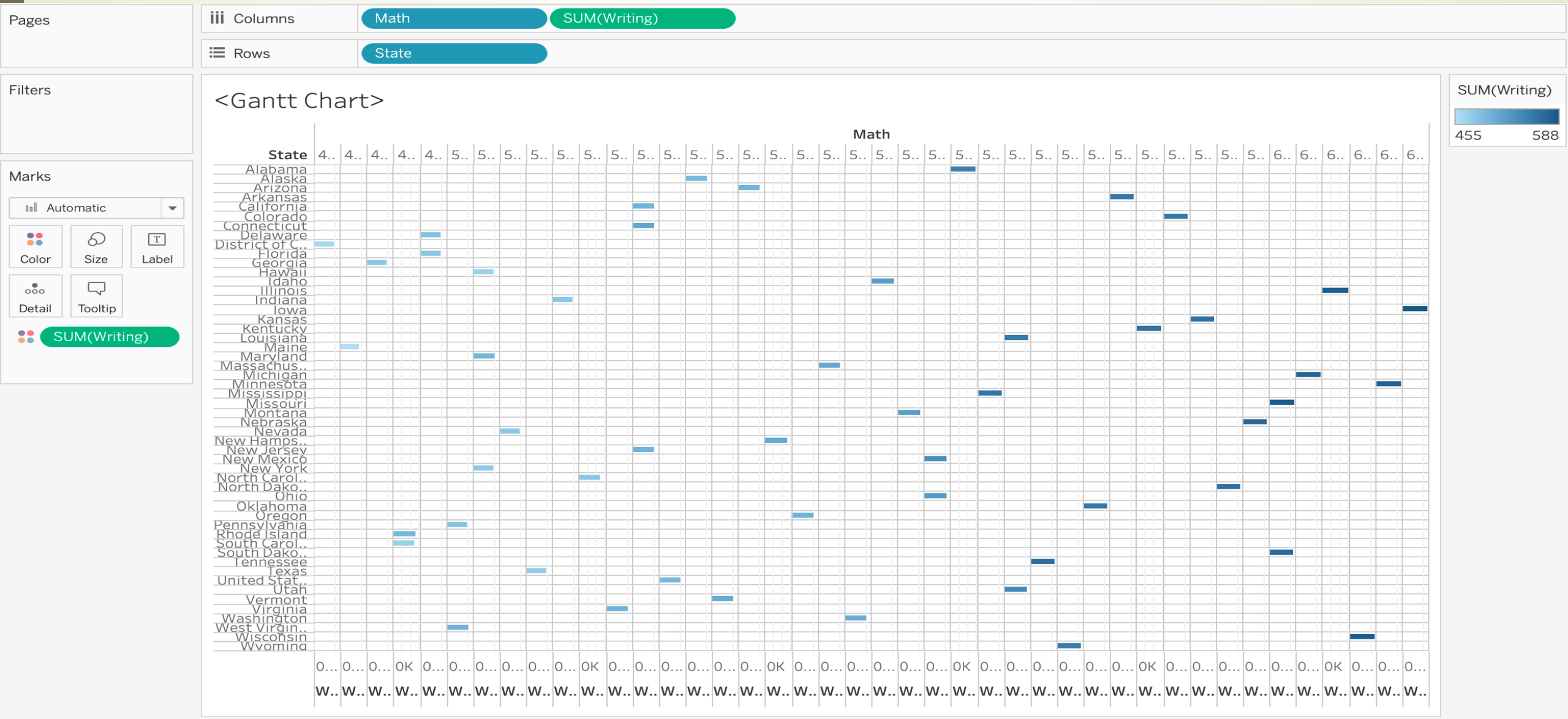


Box Plot


Pages
Columns
Measure Names
Rows
Measure Values
Filters
Measure Names
Marks
Circle
Color
Size
Label
Detail
Tooltip
Measure Names
Measure Values
CNT(education.csv)
Dropout Rate
Percent Graduates S..
Pupil Staff Ratio
Reading
Writing
<Blot Plot>
Value
600
500
400
300
200
100
0
Count of education..
Dropout Rate
Percent Graduat..
Pupil Staf..
Reading
Writing

Bullet Chart

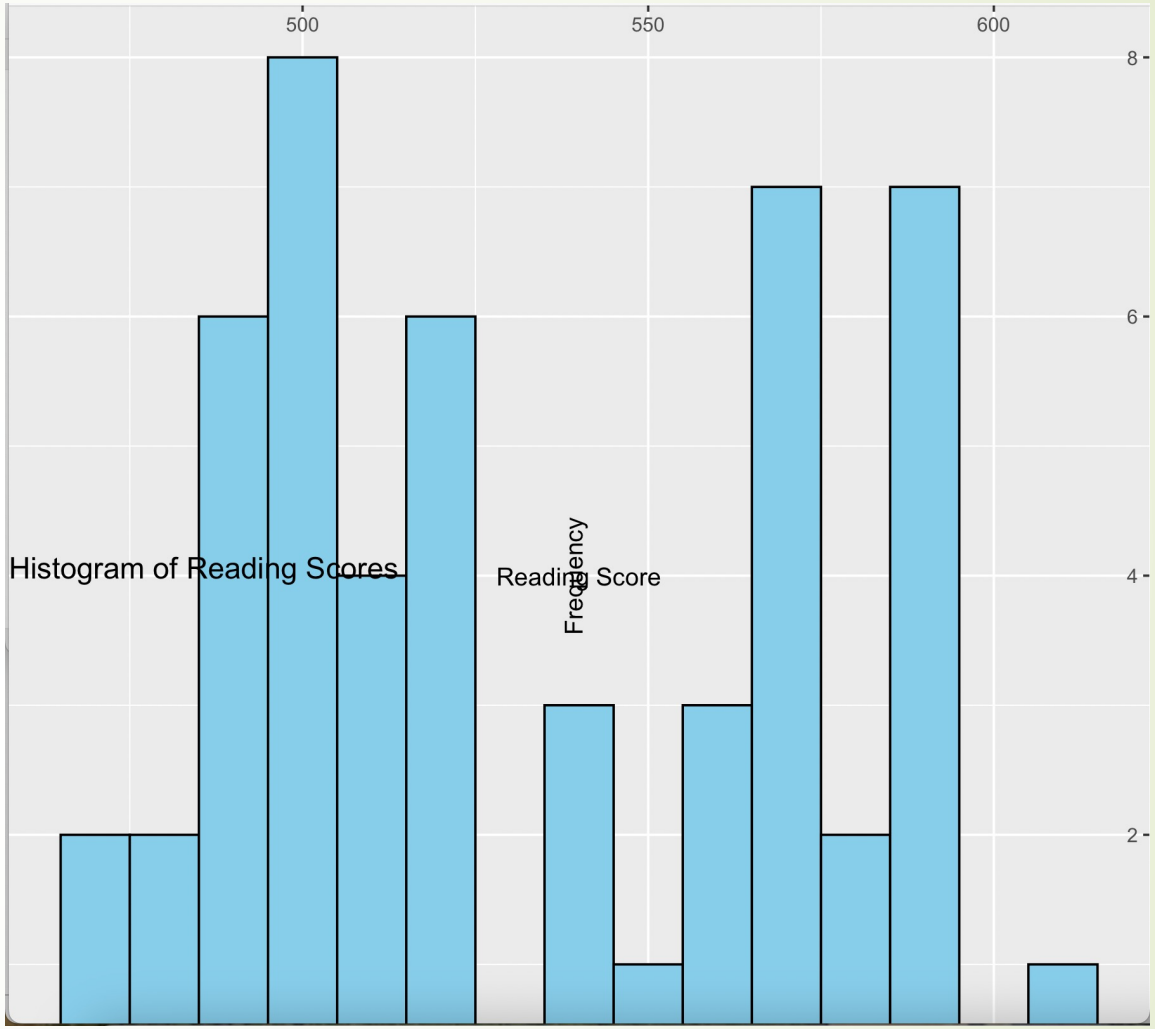
Gnatt Chart


Pages
Columns
Math
SUM(Writing)
Rows
State
Filters
<Gantt Chart>
Math
State
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Writing)
SUM(Writing)
455
588

Charts in R

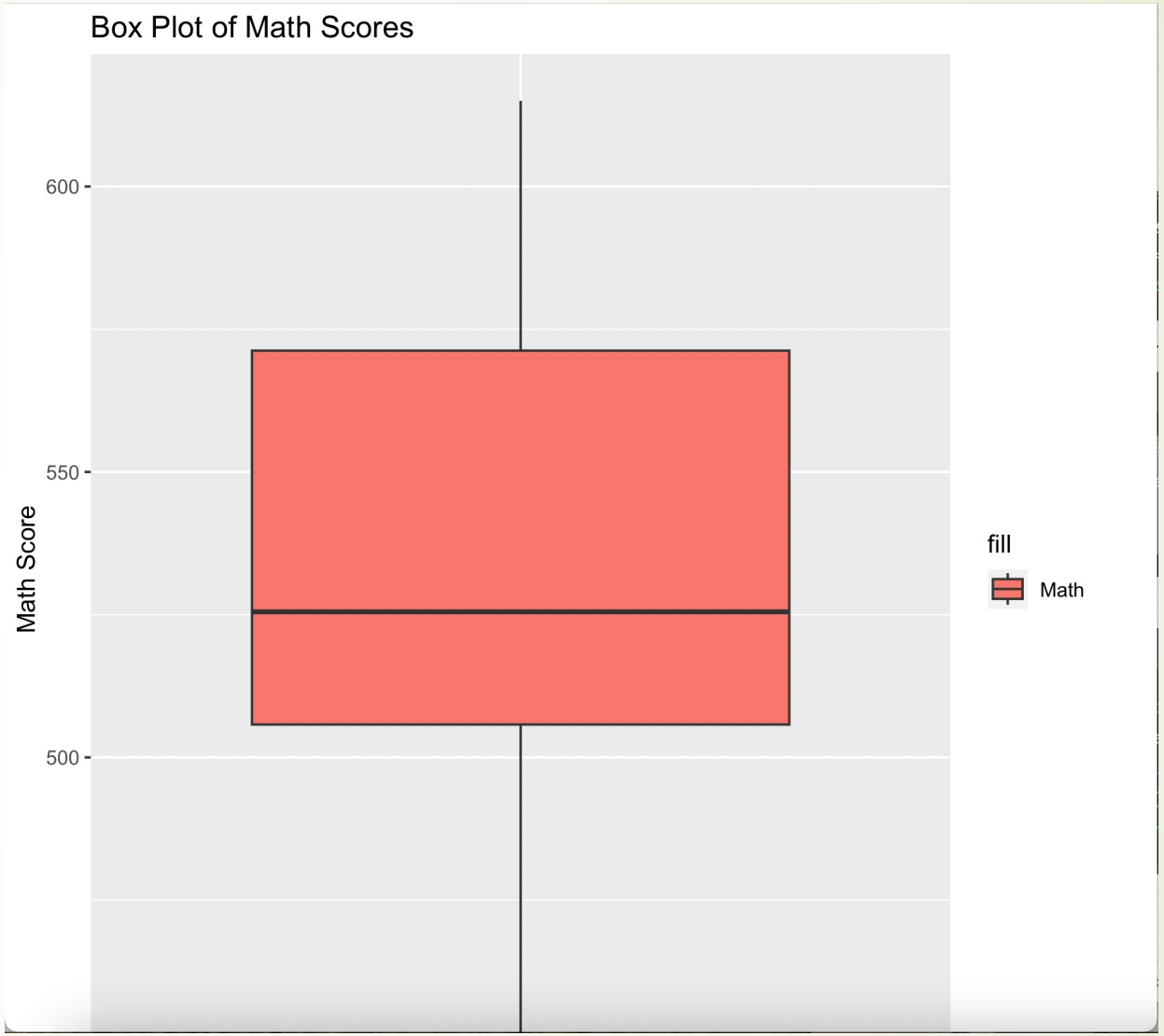
Histogram

```
> # Set the path to the CSV file
> file_path <- "/Users/josephmadden/Downloads/ex6-2/education.csv"
> # Read the CSV file into a data frame
> education.csv <- read.csv(/Users/josephmadden/Downloads/ex6-2)
Error: unexpected '/' in "education.csv <- read.csv(/"
> # Read the CSV file into a data frame
> education_data <- read.csv(file_path)
> library(ggplot2)
> # Create a histogram of reading scores
> ggplot(education_data, aes(x = reading)) +
+   geom_histogram(binwidth = 10, fill = "skyblue", color = "black") +
+   labs(title = "Histogram of Reading Scores", x = "Reading Score", y =
"Frequency")
```



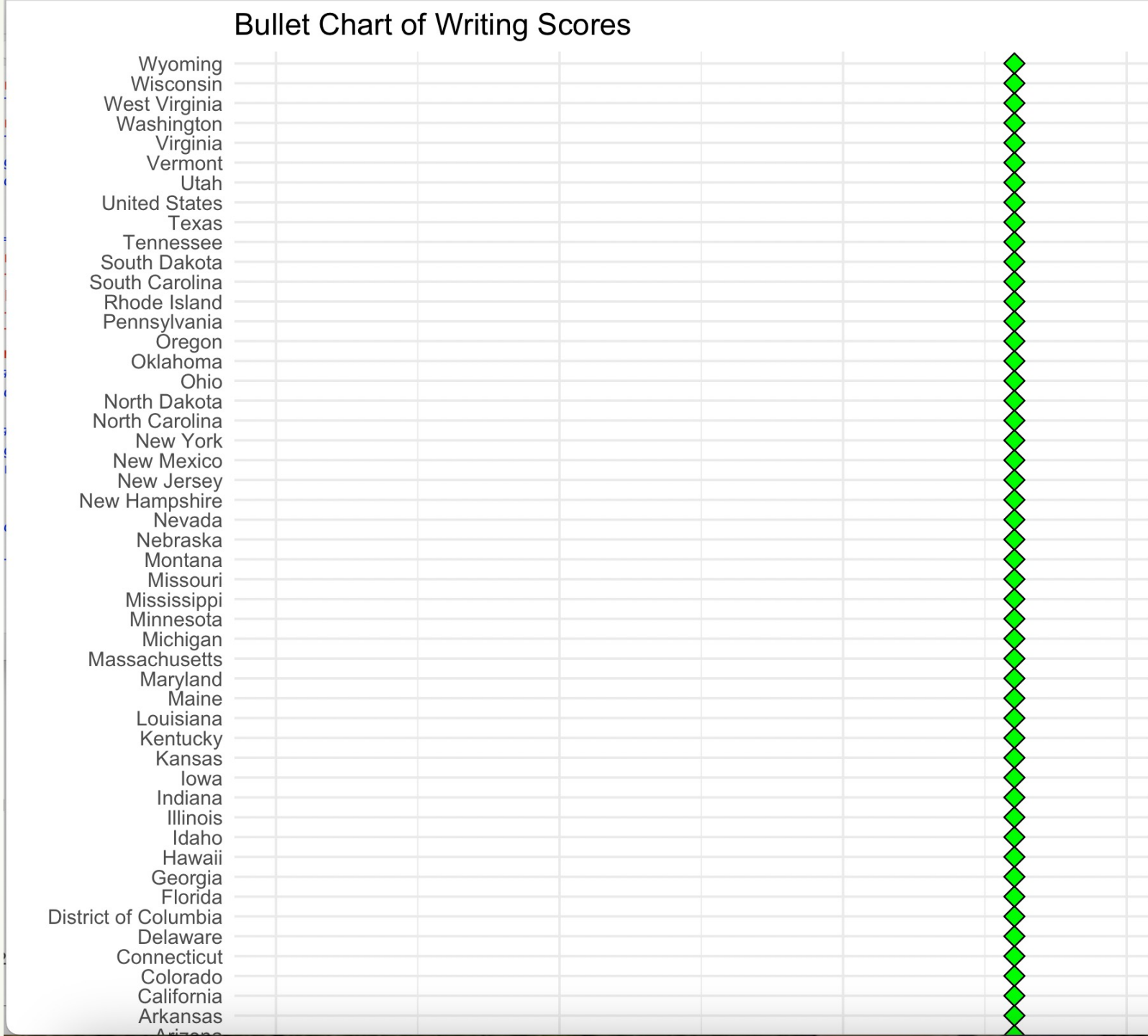
Blox Plot

```
> # Create a box plot of math scores
> ggplot(education_data, aes(x = "", y = math, fill = "Math")) +
+   geom_boxplot() +
+   labs(title = "Box Plot of Math Scores", x = "", y = "Math Score")
>
```



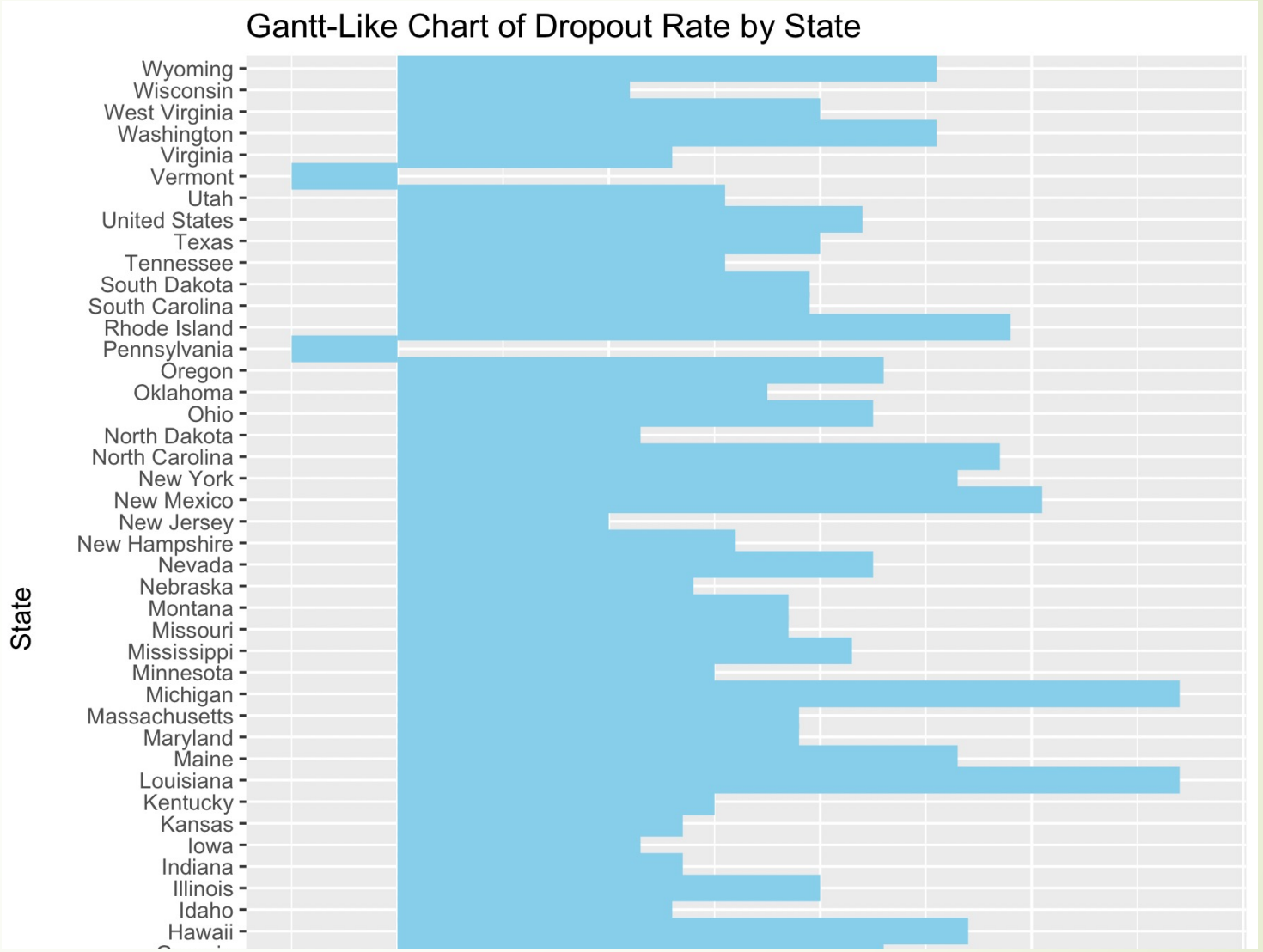
Bullet Chart

```
> # Calculate the average writing score
> avg_writing_score <- mean(education_data$writing)
>
> # Create a bullet chart of writing scores
> ggplot(education_data, aes(x = 0, xmin = 0, xmax = writing, y = state, fill =
"Writing")) +
+   geom_rect(aes(ymin = state, ymax = state, fill = "Writing"), alpha = 0.5) +
+   geom_point(aes(x = avg_writing_score, y = state), color = "black", size = 3,
shape = 23) +
+   labs(title = "Bullet Chart of Writing Scores", x = "Writing Score", y = "",
fill = "") +
+   scale_fill_manual(values = c("Writing" = "green"), guide = "none") +
+   theme_minimal()
```



Gnatt Chartt

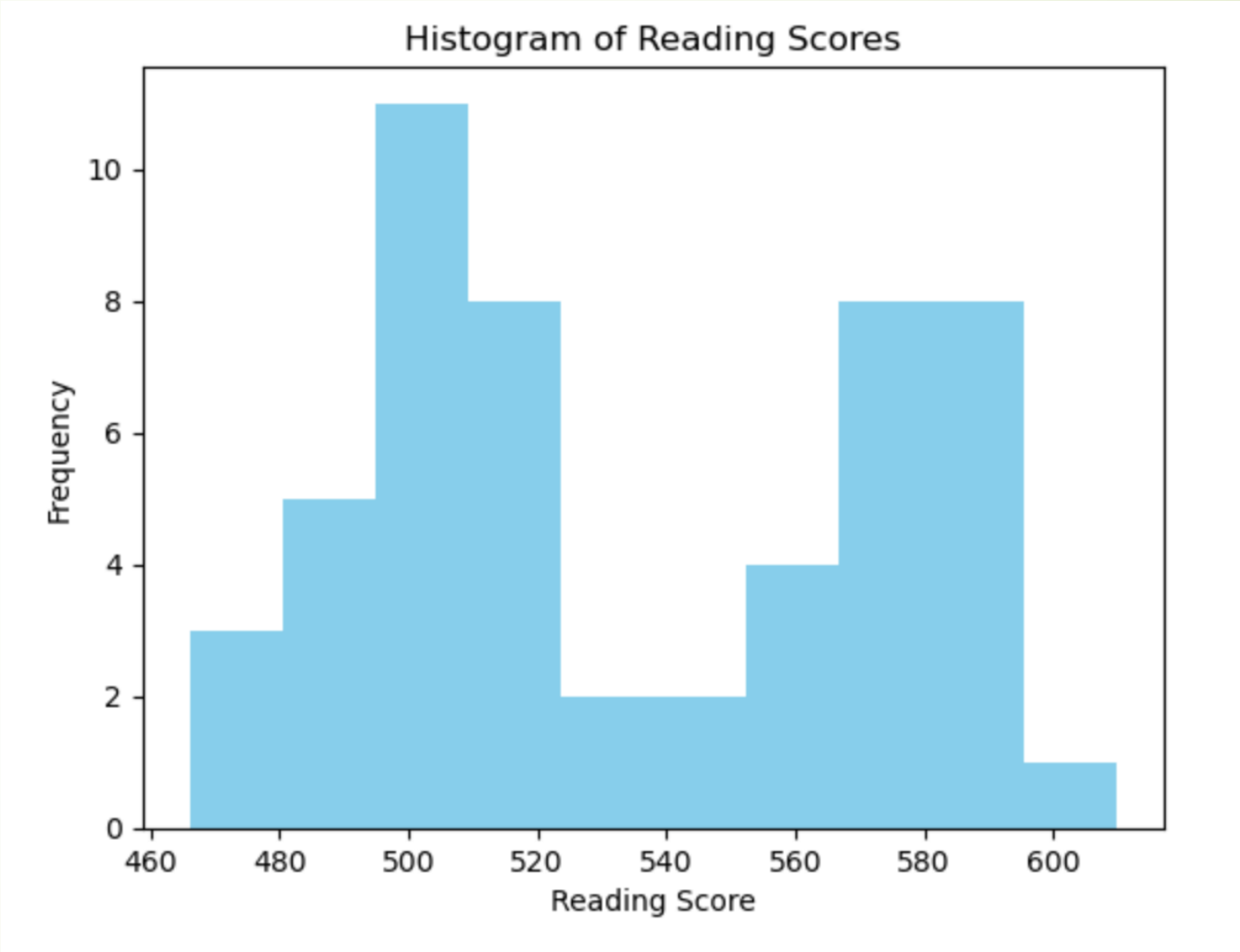
```
> library(ggplot2)
> ggplot(education_data, aes(x = 0, xend = dropout_rate, y = state, yend =
state)) +
+   geom_segment(size = 5, color = "skyblue") +
+   labs(title = "Gantt-Like Chart of Dropout Rate by State", x = "Dropout Rate",
y = "State")
```



Charts in Python

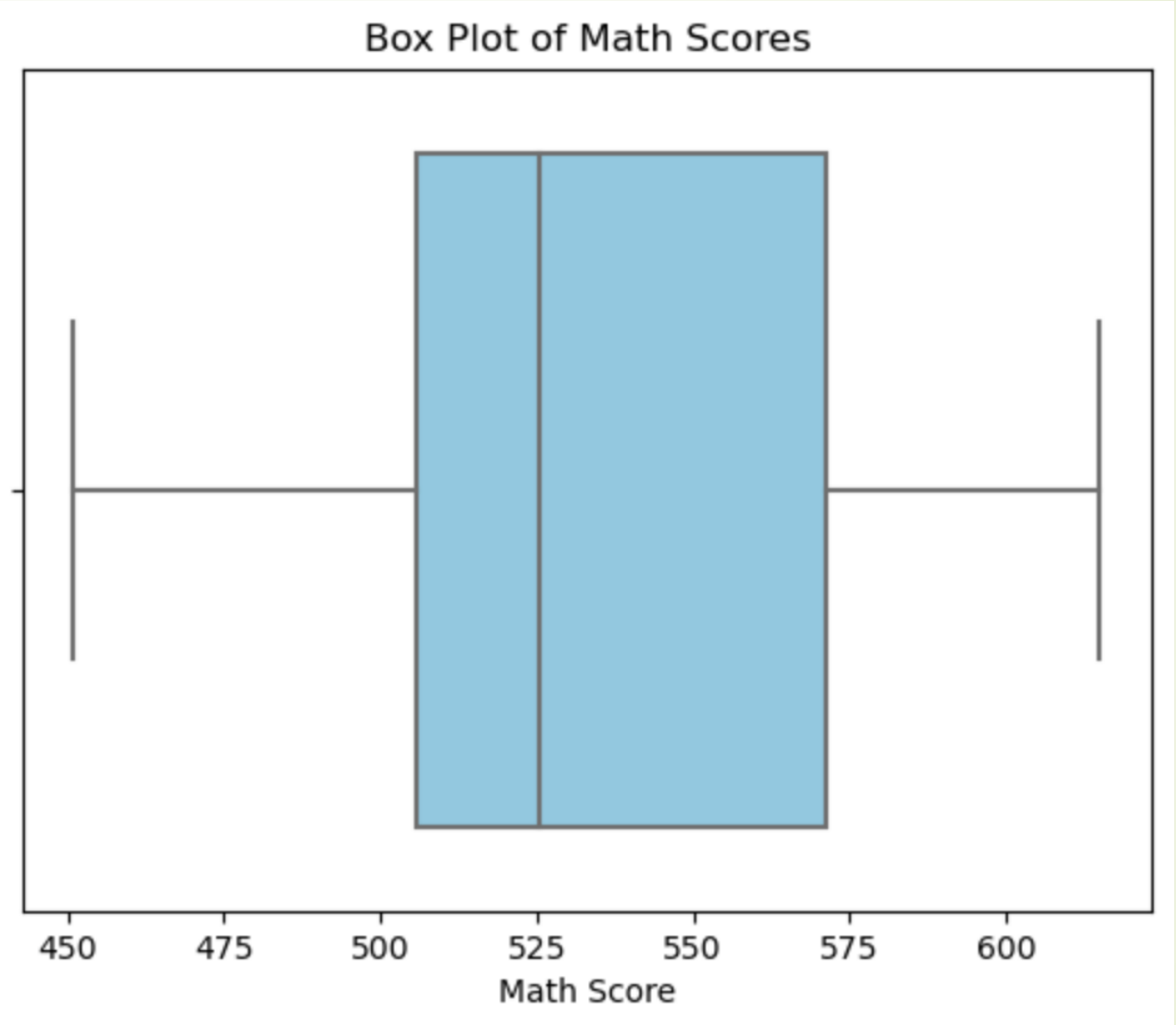
Histogram

```
[1]: import pandas as pd
     import matplotlib.pyplot as plt

[2]: # Read the dataset
     education_data = pd.read_csv("education.csv")

[3]: # Create a histogram of reading scores
     plt.hist(education_data["reading"], bins=10, color="skyblue")
     plt.title("Histogram of Reading Scores")
     plt.xlabel("Reading Score")
     plt.ylabel("Frequency")
     plt.show()
```



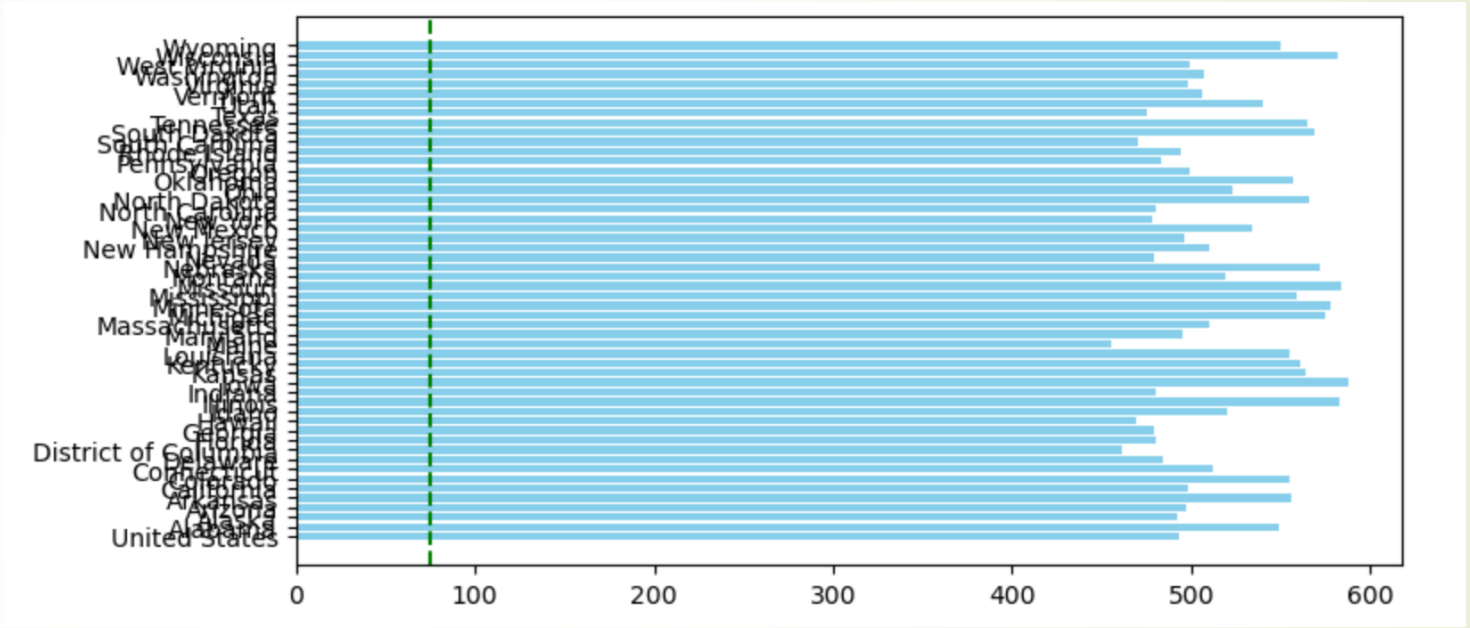
Box Plot

```
import seaborn as sns

# Create a box plot of math scores
sns.boxplot(x=education_data["math"], color="skyblue")
plt.title("Box Plot of Math Scores")
plt.xlabel("Math Score")
plt.show()
```



Bullet Chart

```
# Calculate the average writing score
avg_writing_scores = [avg_writing_score] * len(education_data["state"])

# Create a bullet chart of writing scores
plt.figure(figsize=(8, 4))
plt.barh(y=education_data["state"], width=education_data["writing"], color="skyblue")
plt.axvline(x=75, color="green", linestyle="--")
plt.scatter(x=avg_writing_score, y=education_data["state"], color="black", label="Average Score", zorder=5)
plt.xlabel("Writing Score")
plt.ylabel("State")
plt.title("Bullet Chart of Writing Scores")
plt.legend()
plt.show()
```



Gnatt Chart

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

```
# Sort the data by state
education_data.sort_values(by="state", inplace=True)
```

```
# Create a Gantt chart
plt.figure(figsize=(12, 6))
for i, (state, dropout_rate) in enumerate(education_data[["state", "dropout_rate"]].values):
    plt.barh(y=i, width=dropout_rate, color="skyblue")
plt.yticks(range(len(education_data)), education_data["state"])
plt.xlabel("Dropout Rate")
plt.ylabel("State")
plt.title("Gantt Chart of Dropout Rate by State")
plt.show()
```