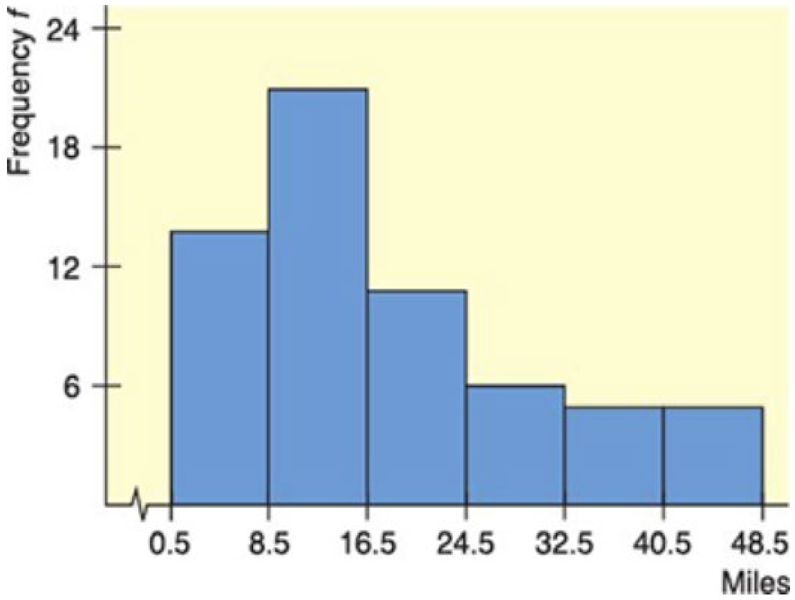
Data Visualization – Part 1

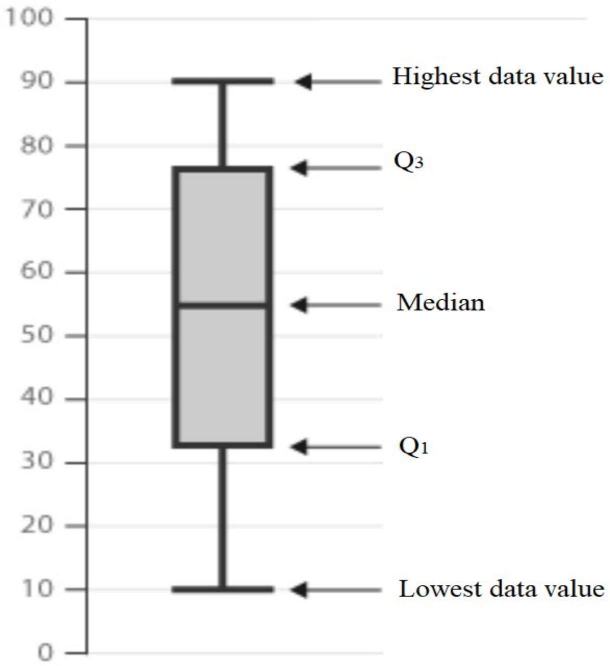
Histograms & Density

Commuting_Dist
1
5
5
6
7
4
8
7
6
5
5
6
7
…

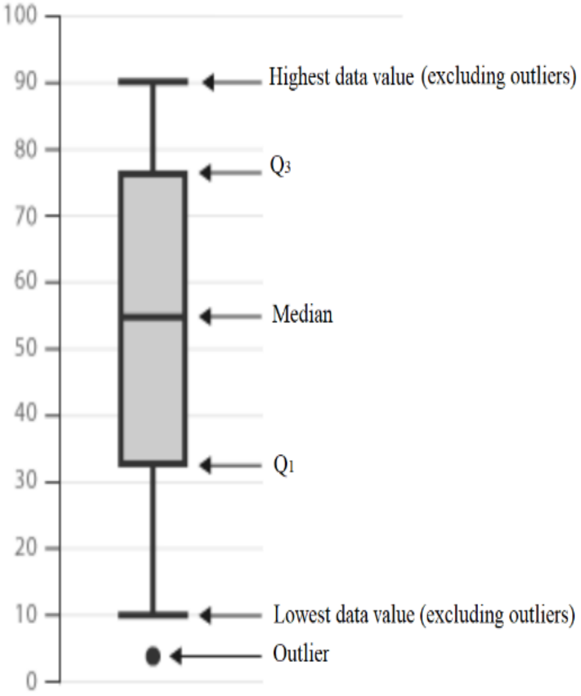


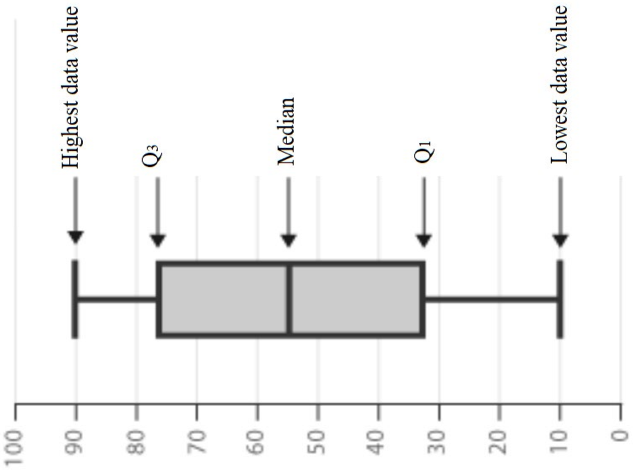
Boxplots

Time_Exam
90
80
75
80
50
55
45
40
10
20
25
30
35
40
…



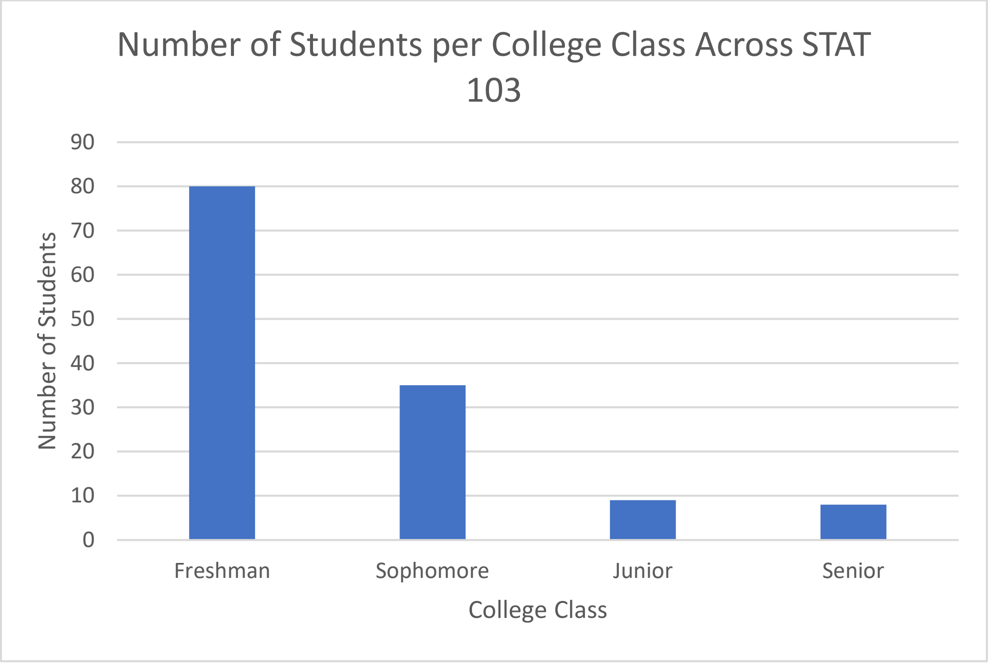
Other ways of representing boxplots:





Bar Graphs

Grad_Class
Freshman
Freshman
Freshman
Junior
Senior
Sophomore
Freshman
Sophomore
Freshman
Freshman
Junior
Sophomore
Freshman
Freshman
Sophomore
…



`ggplot()` – Function

- **Data:** The dataset you're working with.
- **Aesthetics Mapping (aes):** How data variables map to plot aesthetics like position, color, shape, etc.

```
x = variable1
y = variable2
color = variable3
fill = variable4
shape = variable5
```

- **Geometric Objects (geom):** The visual elements to represent the data (points, lines, bars, etc.).

```
geom_histogram()
geom_density()
geom_boxplot()
geom_line()
geom_bar()
geom_point()
```

- **Facets (facet_wrap or facet_grid):** Splitting data into subplots based on a variable.

```
facet_wrap(~ Variable1)
facet_grid(Variable1 ~ Variable2)
```

- **Theme:** Controlling the overall appearance of the plot.

```
theme_gray()
theme_bw()
theme_minimal()
theme_void()
```

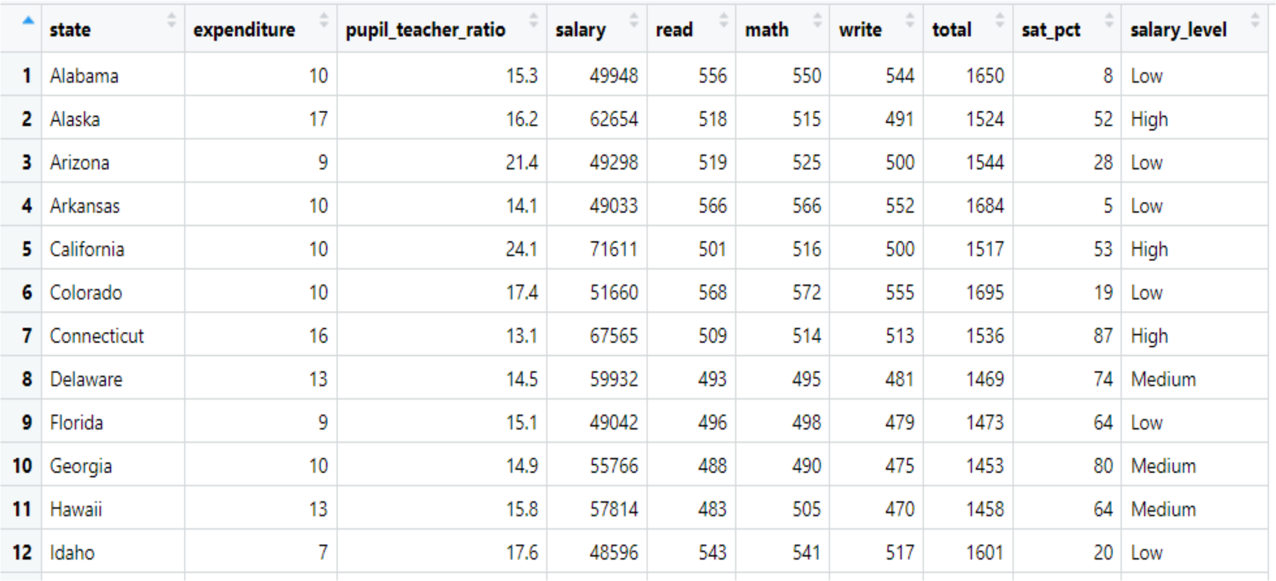
- **Add ons**

```
coord_flip()
geom_smooth(method = "lm", se = FALSE)
geom_smooth(method = "loess", se = TRUE)
scale_fill_manual(values = c("red", "green", "blue"))
scale_color_manual(values = c("red", "green", "blue"))
labs()
```

Command Illustration

```
Plot_Name <- name_daframe %>%
ggplot(aes(x = column_name)) +
    geom_OBJECT()+
    facet()+
    theme()
```

Data



	state	expenditure	pupil_teacher_ratio	salary	read	math	write	total	sat_pct	salary_level
1	Alabama	10	15.3	49948	556	550	544	1650	8	Low
2	Alaska	17	16.2	62654	518	515	491	1524	52	High
3	Arizona	9	21.4	49298	519	525	500	1544	28	Low
4	Arkansas	10	14.1	49033	566	566	552	1684	5	Low
5	California	10	24.1	71611	501	516	500	1517	53	High
6	Colorado	10	17.4	51660	568	572	555	1695	19	Low
7	Connecticut	16	13.1	67565	509	514	513	1536	87	High
8	Delaware	13	14.5	59932	493	495	481	1469	74	Medium
9	Florida	9	15.1	49042	496	498	479	1473	64	Low
10	Georgia	10	14.9	55766	488	490	475	1453	80	Medium
11	Hawaii	13	15.8	57814	483	505	470	1458	64	Medium
12	Idaho	7	17.6	48596	543	541	517	1601	20	Low

Histograms

Figure 1: Create a Histogram on the Average Math SAT Score.

```
fig_1 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_histogram()
fig_1
```

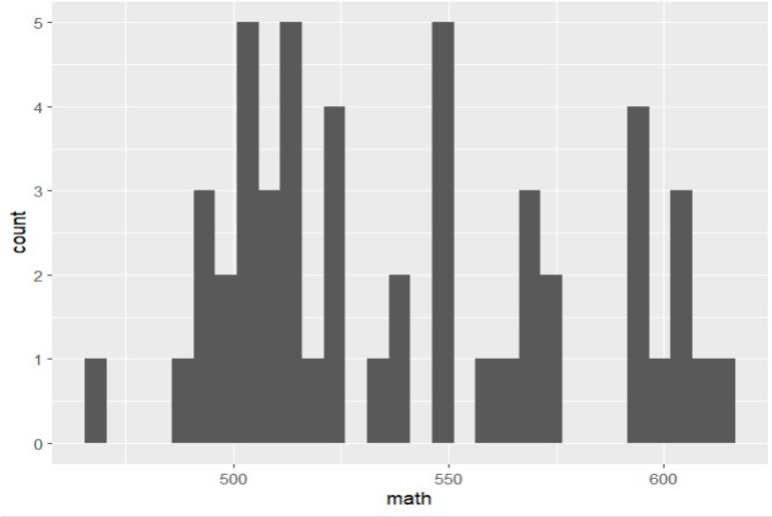


Figure 2: Create a Histogram on the Average Math SAT Score, but change the break points to start from 450, end in 625, and increase by 25

```
fig_2 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_histogram(breaks = c(450,475,500,525,550,575,600,625))
fig_2
```

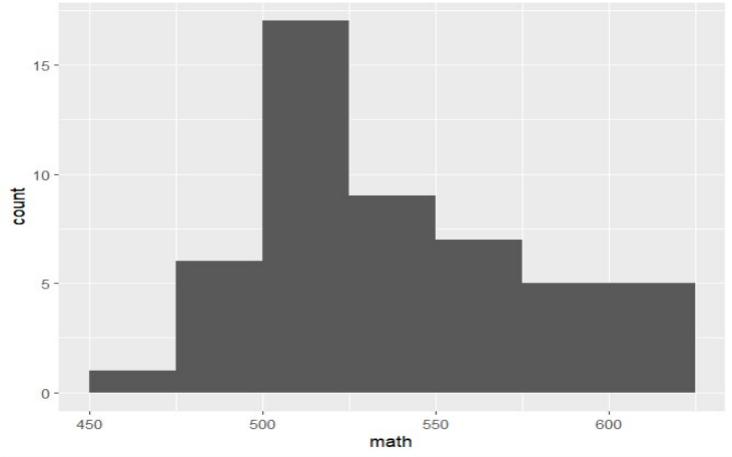
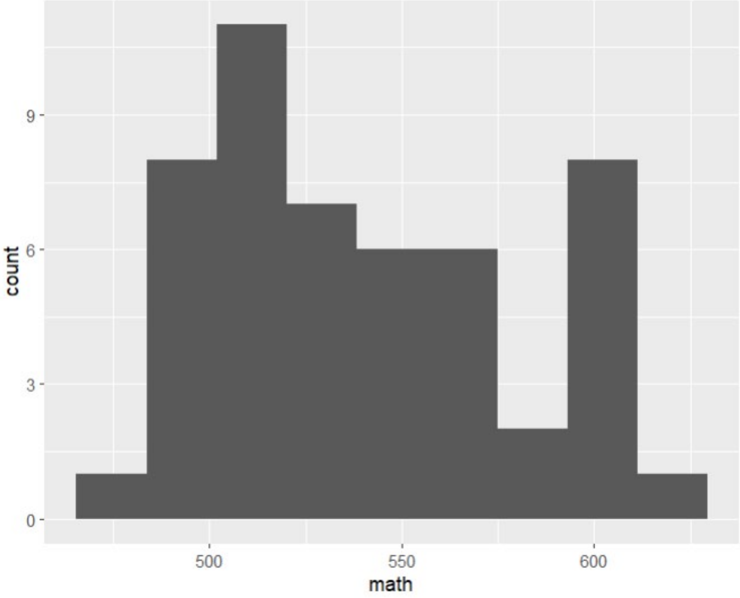


Figure 3: Create a Histogram on the Average Math SAT Score, but choose the bins to be 9.

```
fig_3 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_histogram(bins = 9)
fig_3
```



Density Plots

Figure 4: Create a density plot on the Average Math SAT Score

```
fig_4 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_density()
fig_4
```

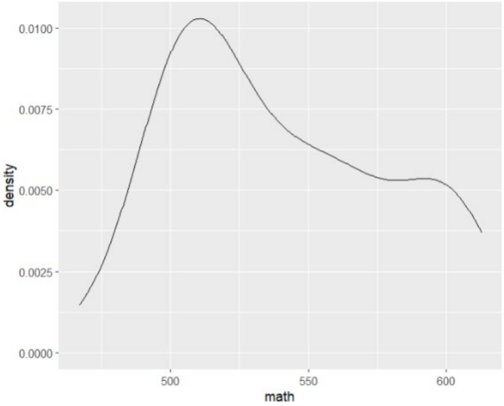


Figure 5: Create a density plot on the Average Math SAT Score. Change the bandwith to 10.

Note: Changing the bandwidth to a smaller number makes it more jagged.

```
fig_5 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_density(bw = 10)
fig_5
```

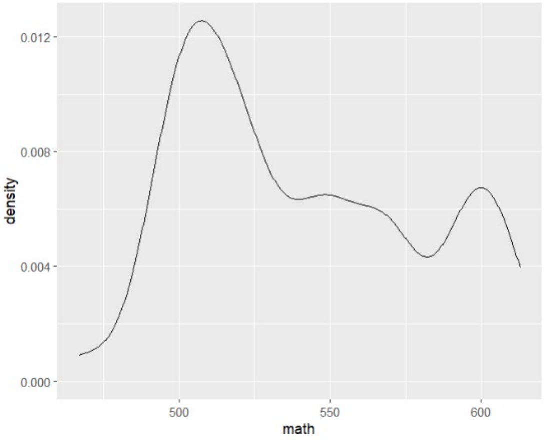
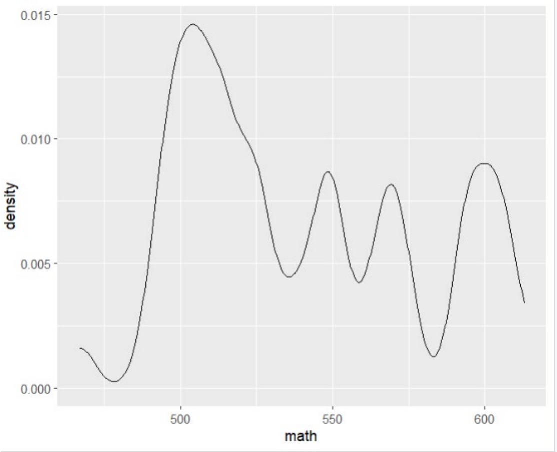


Figure 6: Create a density plot on the Average Math SAT Score. Change the bandwith to 10.

```
fig_6 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_density(bw = 5)
fig_6
```



Boxplots

Figure 7: Create a box plot on the Average Math SAT Score.

```
fig_7 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_boxplot()
fig_7
```

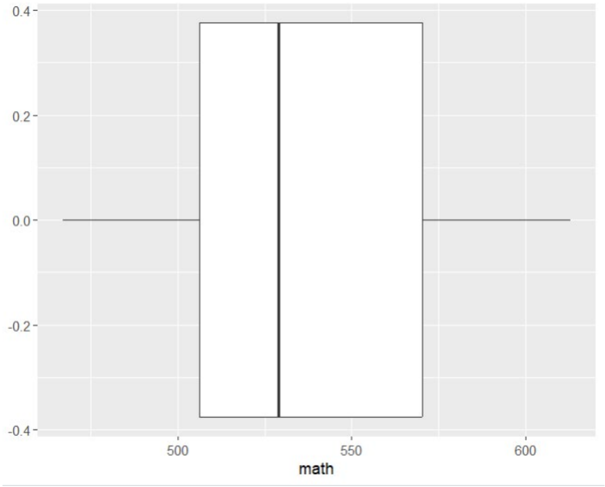
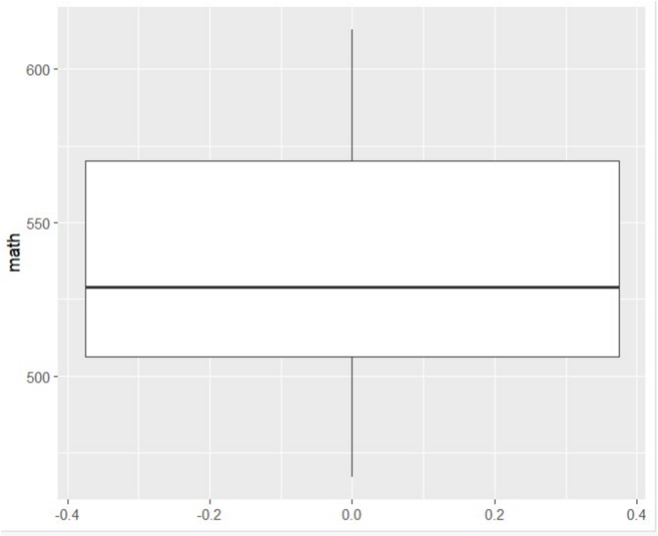


Figure 8: Create a box plot on the Average Math SAT Score. Make the boxplot vertical.

```
fig_8 <- SAT_2010 %>%
  ggplot(aes(x = math)) +
  geom_boxplot() +
  coord_flip()
fig_8
```



Bar Plots

Figure 9: Create a bar graph on the Salary Level (the new variable you created).

```
fig_9 <- SAT_2010 %>%
  ggplot(aes(x = salary_level)) +
  geom_bar()
fig_9
```

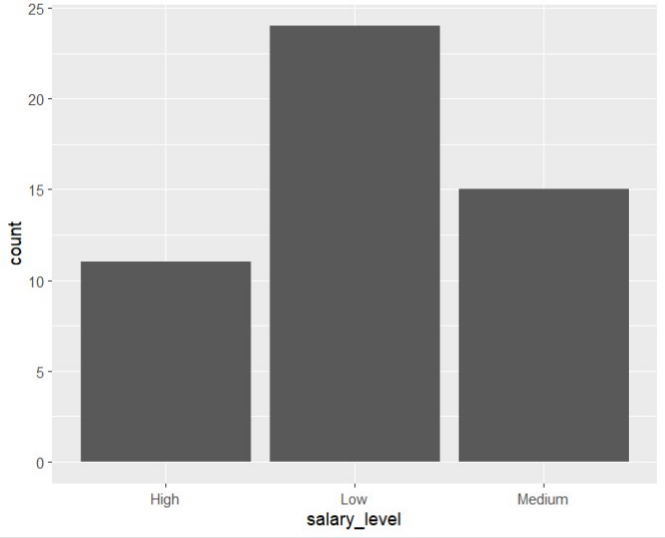
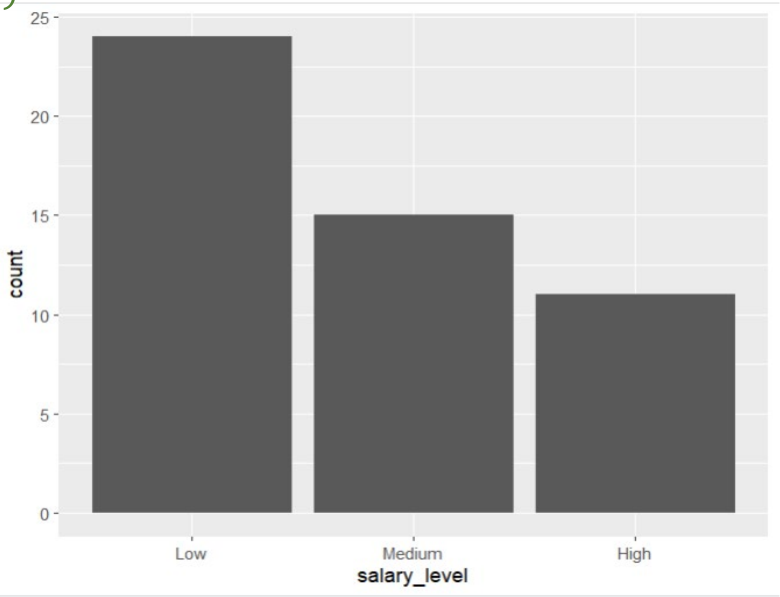


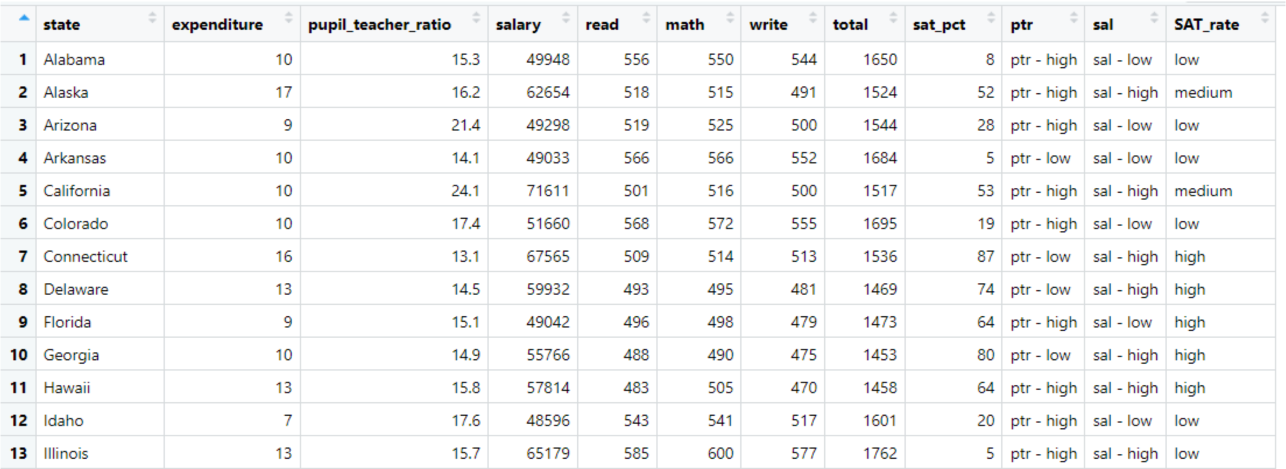
Figure 10: Create a bar graph on the Salary Level (the new variable you created). Have the category "Low" appear first, then "Medium", then "High".

```
fig_10 <- SAT_2010 %>%
     mutate(salary_level = factor(salary_level, levels =
     c("Low",  "Medium", "High"))) %>%
     ggplot(aes(x = salary_level)) +
     geom_bar()
fig_10
```



Data Visualization – Part 2

Data



	state	expenditure	pupil_teacher_ratio	salary	read	math	write	total	sat_pct	ptr	sal	SAT_rate
1	Alabama	10	15.3	49948	556	550	544	1650	8	ptr - high	sal - low	low
2	Alaska	17	16.2	62654	518	515	491	1524	52	ptr - high	sal - high	medium
3	Arizona	9	21.4	49298	519	525	500	1544	28	ptr - high	sal - low	low
4	Arkansas	10	14.1	49033	566	566	552	1684	5	ptr - low	sal - low	low
5	California	10	24.1	71611	501	516	500	1517	53	ptr - high	sal - high	medium
6	Colorado	10	17.4	51660	568	572	555	1695	19	ptr - high	sal - low	low
7	Connecticut	16	13.1	67565	509	514	513	1536	87	ptr - low	sal - high	high
8	Delaware	13	14.5	59932	493	495	481	1469	74	ptr - low	sal - high	high
9	Florida	9	15.1	49042	496	498	479	1473	64	ptr - high	sal - low	high
10	Georgia	10	14.9	55766	488	490	475	1453	80	ptr - low	sal - high	high
11	Hawaii	13	15.8	57814	483	505	470	1458	64	ptr - high	sal - high	high
12	Idaho	7	17.6	48596	543	541	517	1601	20	ptr - high	sal - low	low
13	Illinois	13	15.7	65179	585	600	577	1762	5	ptr - high	sal - high	low

We added three categorical variables to the dataset (SAT_2010):

```
SAT_2010 <- SAT_2010 %>%
  mutate(ptr = ifelse(pupil_teacher_ratio >= 15, "ptr - high","ptr -
  low"),
         sal = ifelse(salary >= 52000, "sal - high","sal - low"),
         SAT_rate = cut(
           sat_pct,
           breaks = c(0, 30, 60, 100),
           labels = c("low", "medium", "high")
   ))
```

Multivariate Displays

Bar Graphs

Figure 1: Make a bar graph with SAT_rate. Notice that this is a One Variable Bar Graph (it counts by default)

```
fig_1 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate)) +
  geom_bar()
fig_1
```

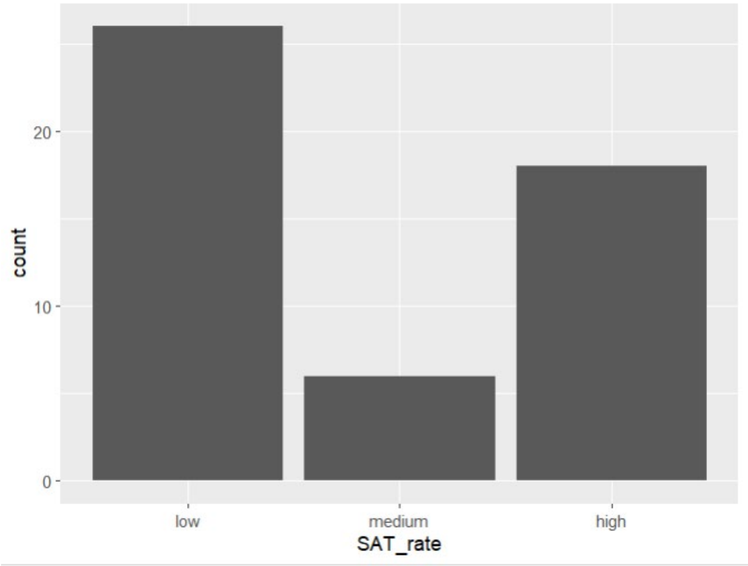


Figure 2: Make a bar graph with SAT_rate on the x axis and The Average Math Score on the y axis. Notice that this is a two Variable Bar Graph (you need stat = "identity")

```
fig_2 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate, y = math)) +
  geom_bar(stat = "identity")
fig_2
```

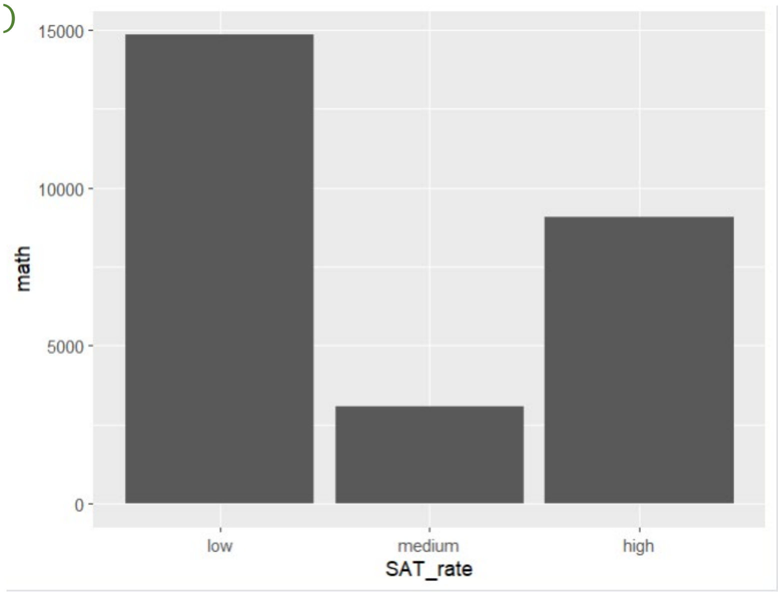


Figure 3: Make a bar graph with SAT_rate on the x axis and The Average Math Score on the y axis. Make every column a different color.

```
fig_3 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate, y = math, fill = SAT_rate)) +
  geom_bar(stat = "identity")
fig_3
```

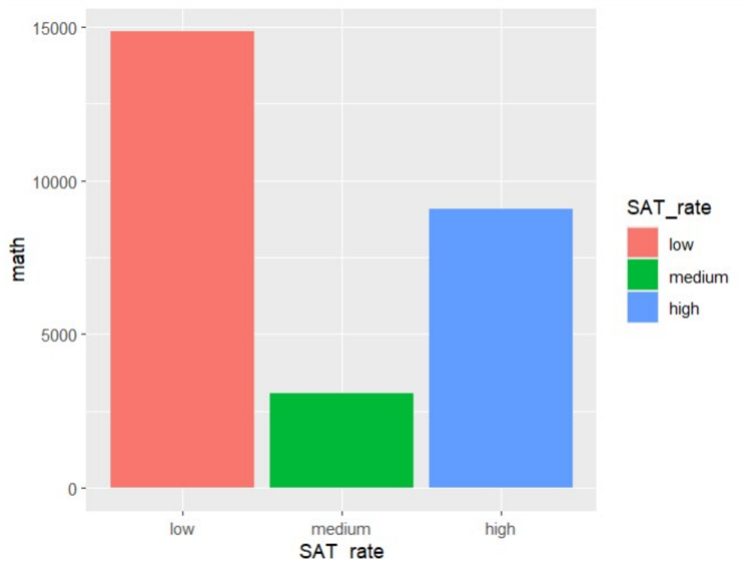


Figure 4: (Stacking by a third variable). Make a bar graph with SAT_rate on the x axis and The Average Math Score on the y axis. Stack the bars by the variable ptr.

```
fig_4 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate, y = math, fill = ptr)) +
  geom_bar(stat = "identity")
fig_4
```

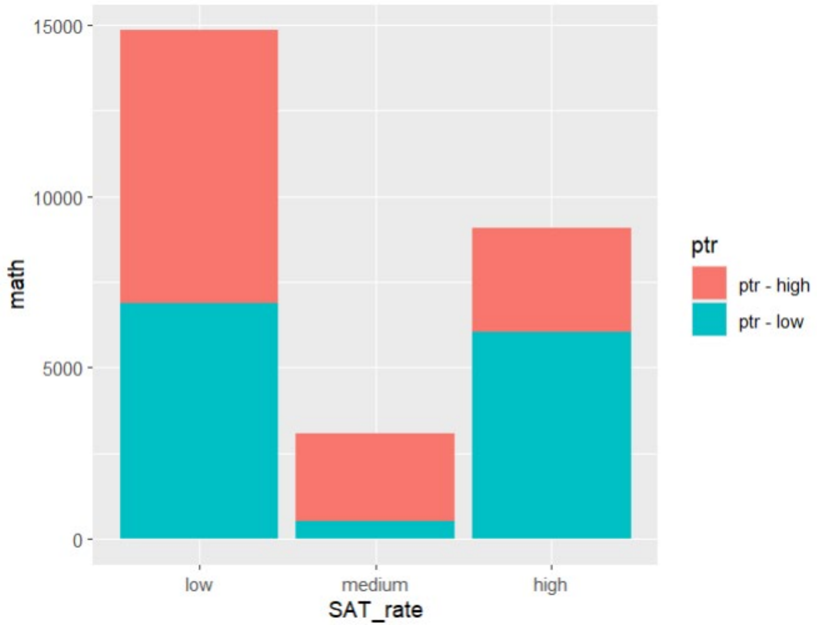
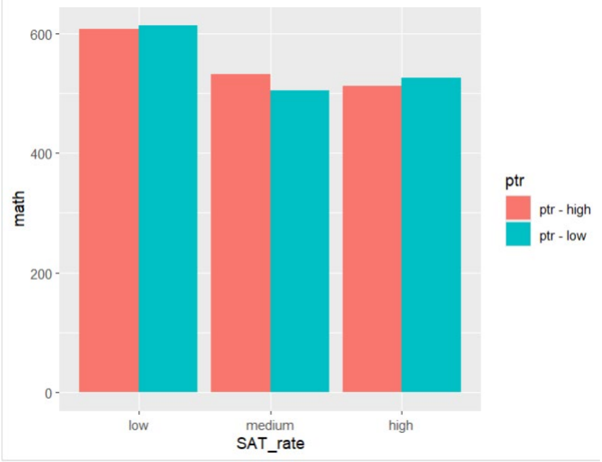


Figure 5: (Grouped Bar Graph). Make a bar graph with SAT_rate on the x axis and The Average Math Score on the y axis. Group the bars by the variable ptr.

```
fig_5 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate, y = math, fill = ptr)) +
  geom_bar(stat = "identity", position = "dodge")
fig_5
```



Box Plots

Figure 6: Make side by side box plots with SAT_rate on the x axis and The Average Math Score on the y axis.

```
fig_6 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate, y = math)) +
  geom_boxplot()
fig_6
```

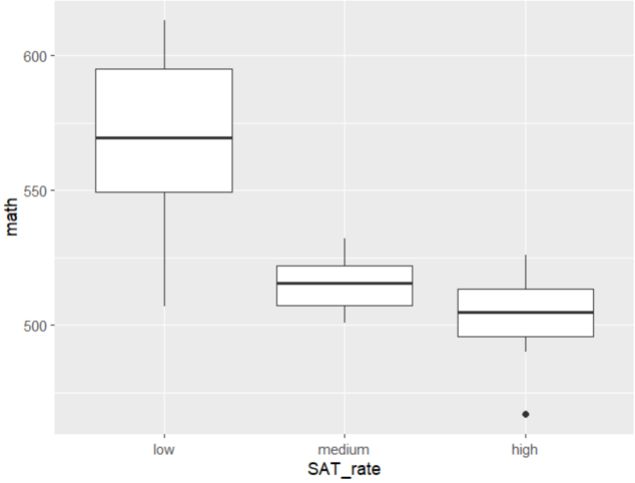
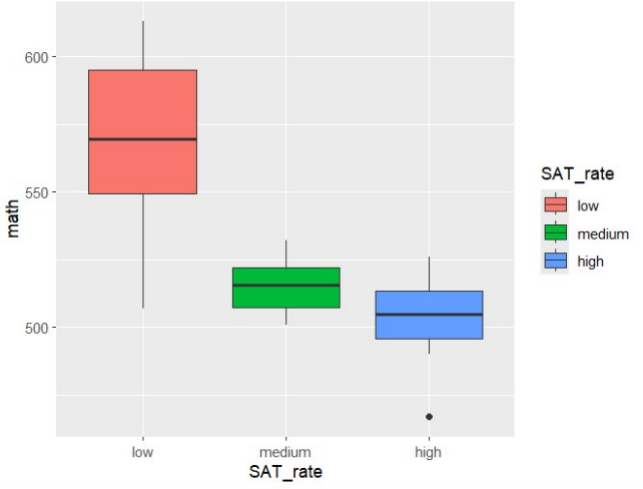


Figure 7: Make side by side box plots with SAT_rate on the x axis and The Average Math Score on the y axis. Change the filling color of each SAT_rate.

```
fig_7 <- SAT_2010 %>%
  ggplot(aes(x = SAT_rate, y = math, fill = SAT_rate)) +
  geom_boxplot()
fig_7
```



Scatter Plots

Figure 8: Create a Scatter Plot on the Expenditure and The Average Math SAT Score.

```
fig_8 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math)) +
  geom_point()
fig_8
```

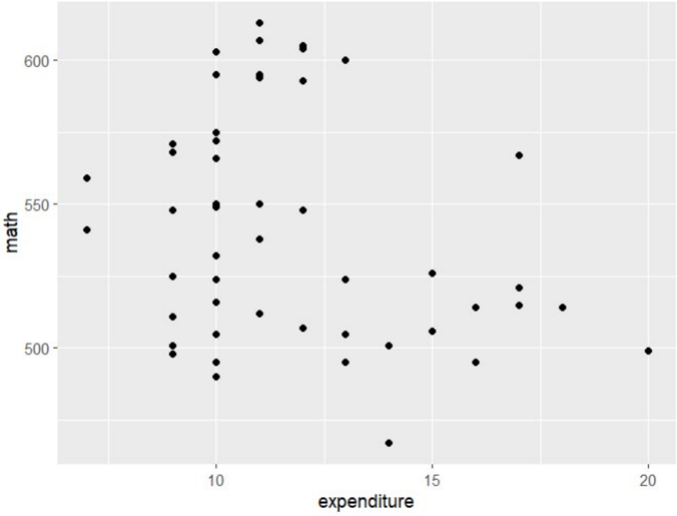


Figure 9: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and add a trend line with ggplot.

```
fig_9 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
fig_9
```

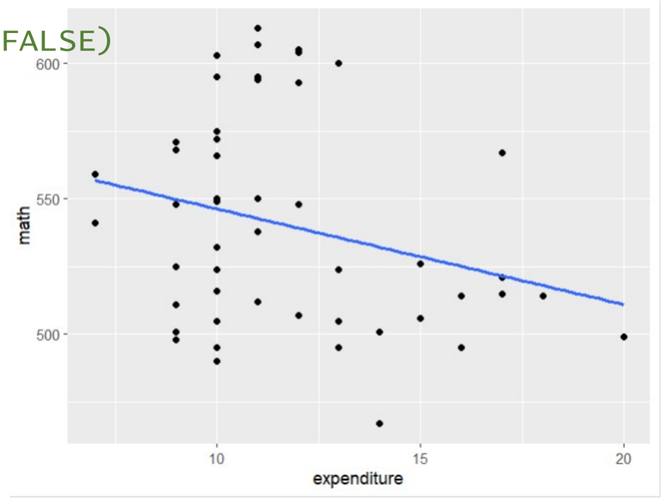


Figure 10: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and add a polynomial fitting with the standard error band.

```
fig_10 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math)) +
  geom_point() +
  geom_smooth(method = "loess", se = TRUE)
fig_10
```

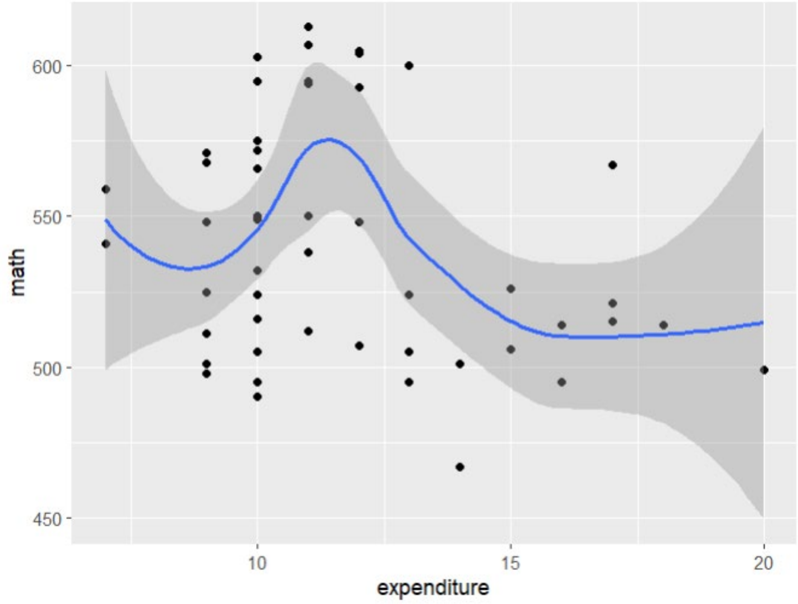


Figure 11: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and split the data with different colors by SAT_rate.

```
fig_11 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math, color = SAT_rate)) +
  geom_point()
fig_11
```

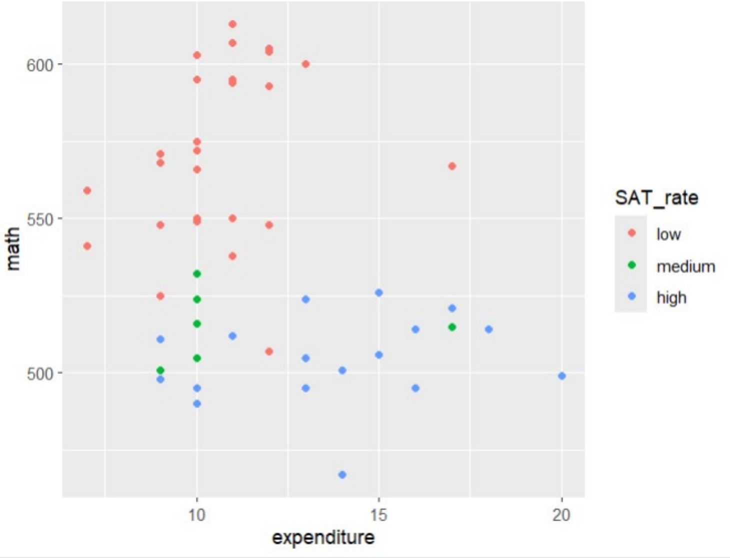


Figure 12: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and split the data with different colors by SAT_rate. Add a trend line.

```
fig_12 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math, color = SAT_rate)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
fig_12
```

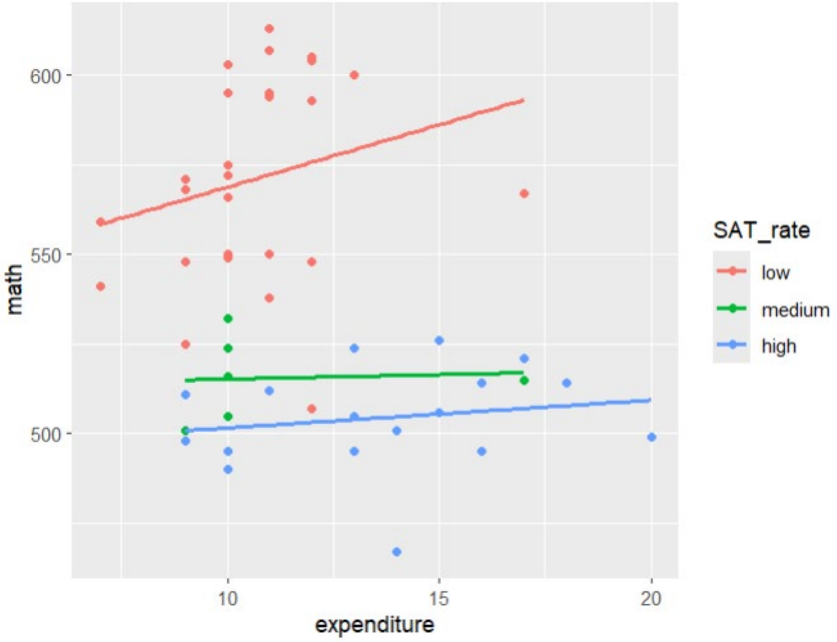


Figure 13: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and split the data with different shapes by SAT_rate.

```
fig_13 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math, shape = SAT_rate)) +
  geom_point()
fig_13
```

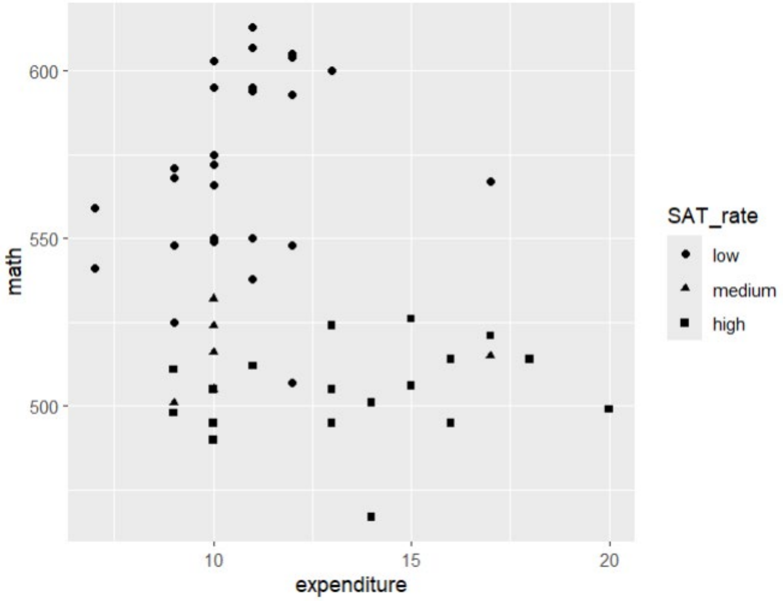
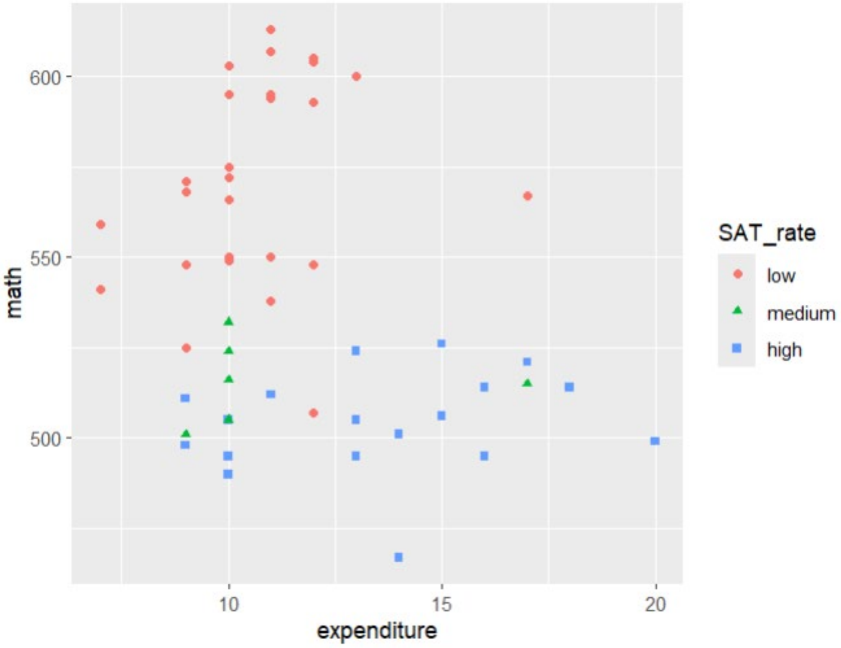


Figure 14: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and split the data with different shapes and colors by SAT_rate.

```
fig_14 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math, color = SAT_rate, shape =
SAT_rate)) +
  geom_point()
fig_14
```



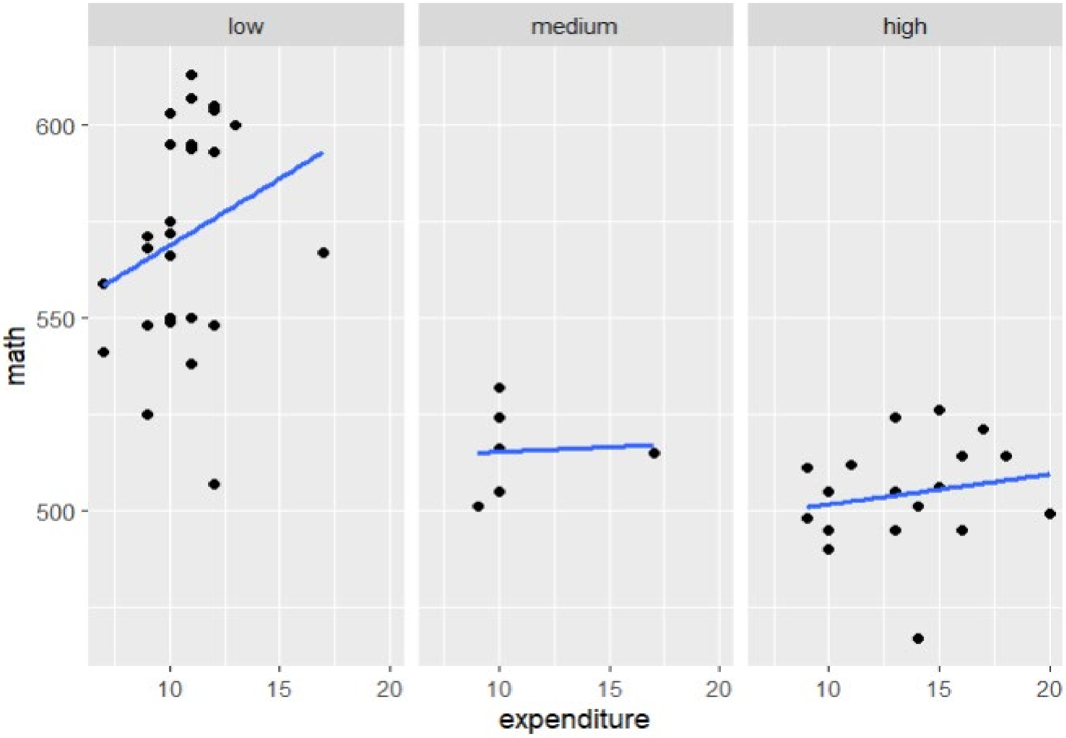
Faceting

This is similar to using shape or color for a categorical variables but puts them on separate plots.

Facet with 1 variable

Figure 15: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and split the data by SAT_rate where each category is a separate plot. Include a trend line.

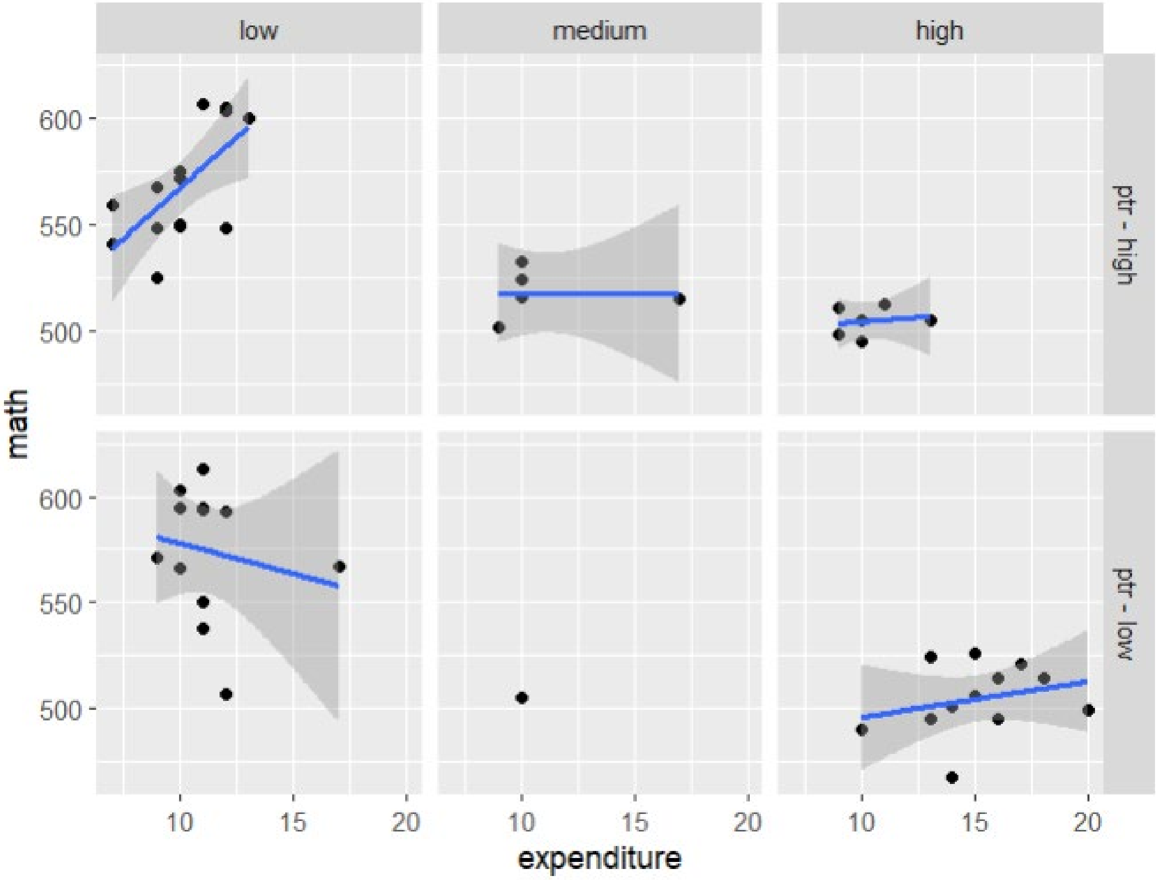
```
fig_15 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~SAT_rate)
fig_15
```



Facet with 2 variables

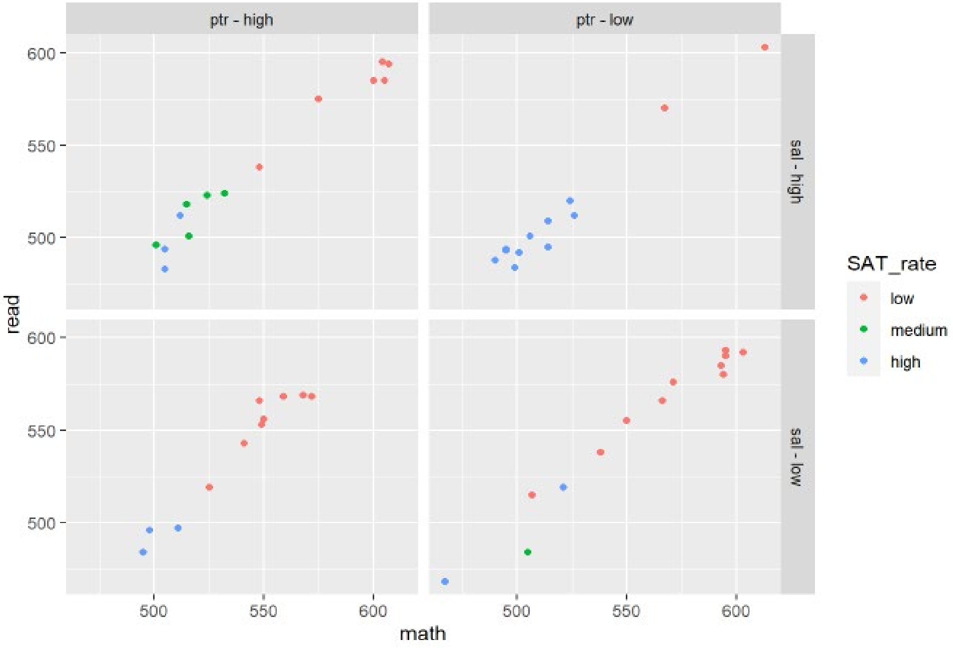
Figure 16: Create a Scatter Plot on the Expenditure and The Average Math SAT Score and split the data by SAT_rate and ptr where each pair of categories is a separate plot. Include a trend line with a standard error band.

```
fig_16 <- SAT_2010 %>%
  ggplot(aes(x = expenditure, y = math)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  facet_grid(ptr ~ SAT_rate)
fig_16
```

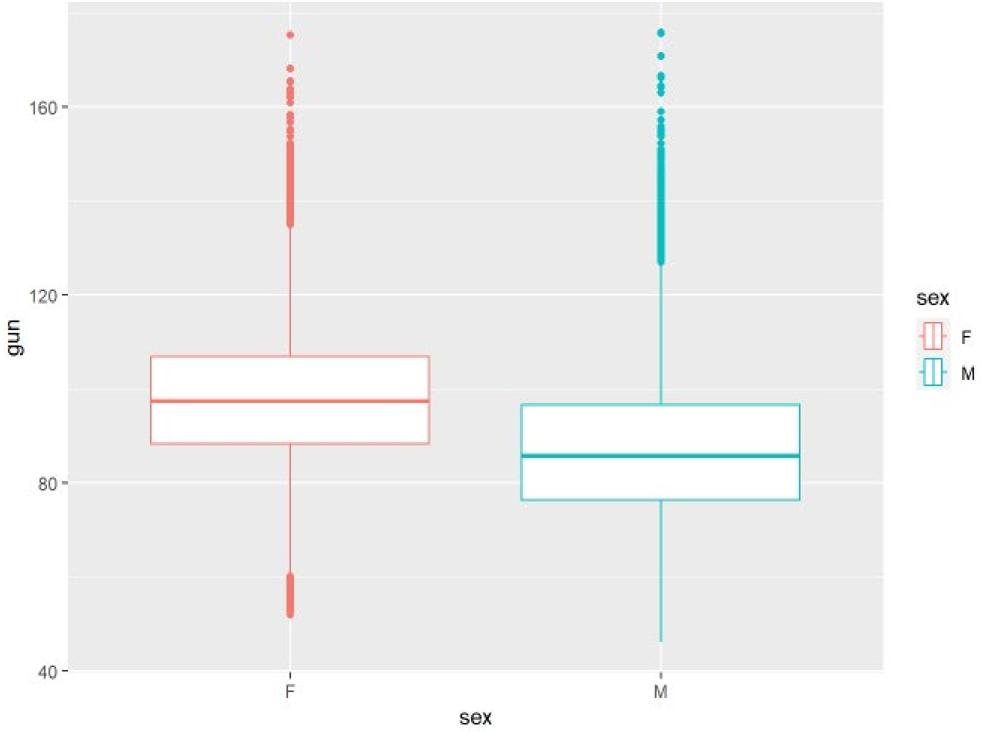


Do it Yourself – Recreate These Plots

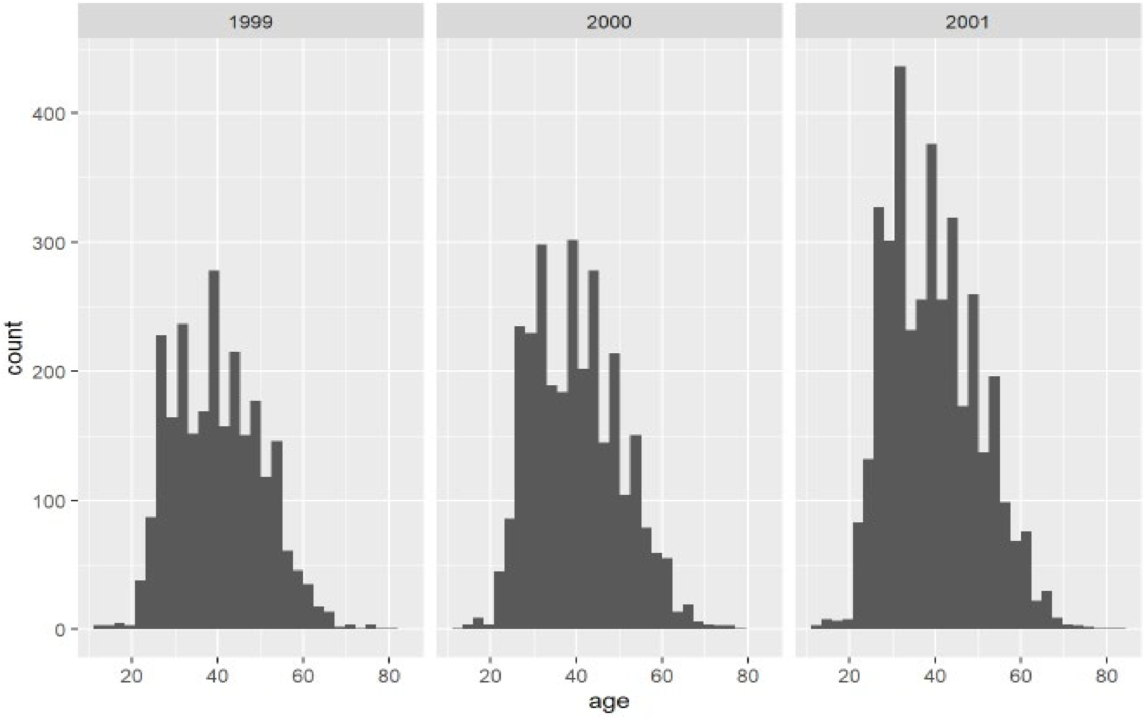
Plot # 1



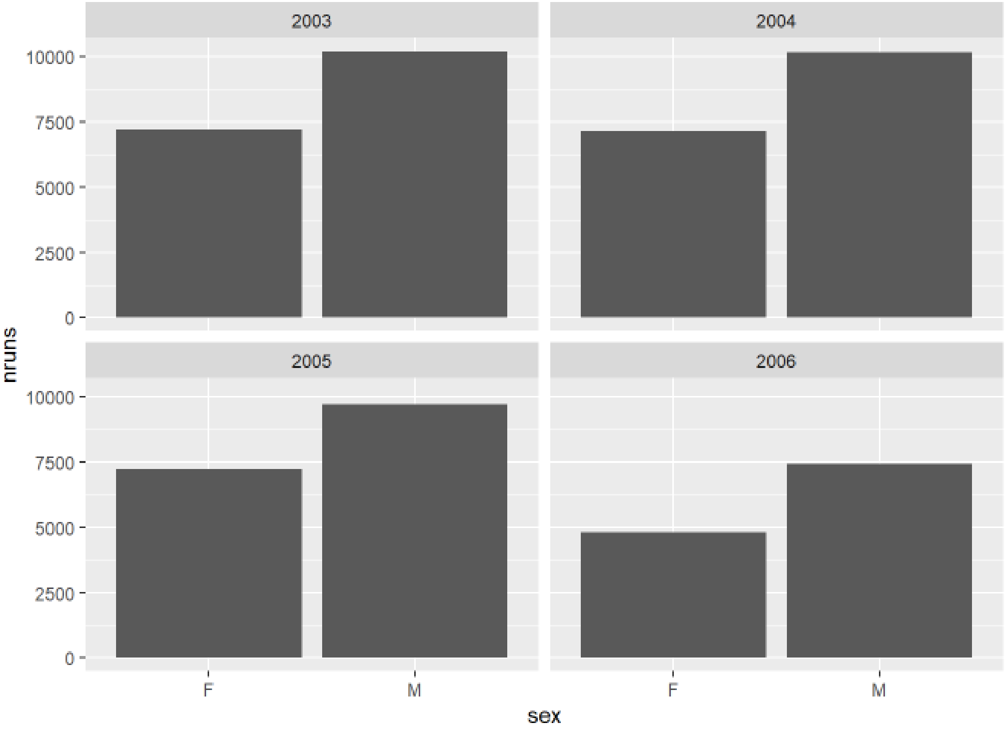
Plot # 2



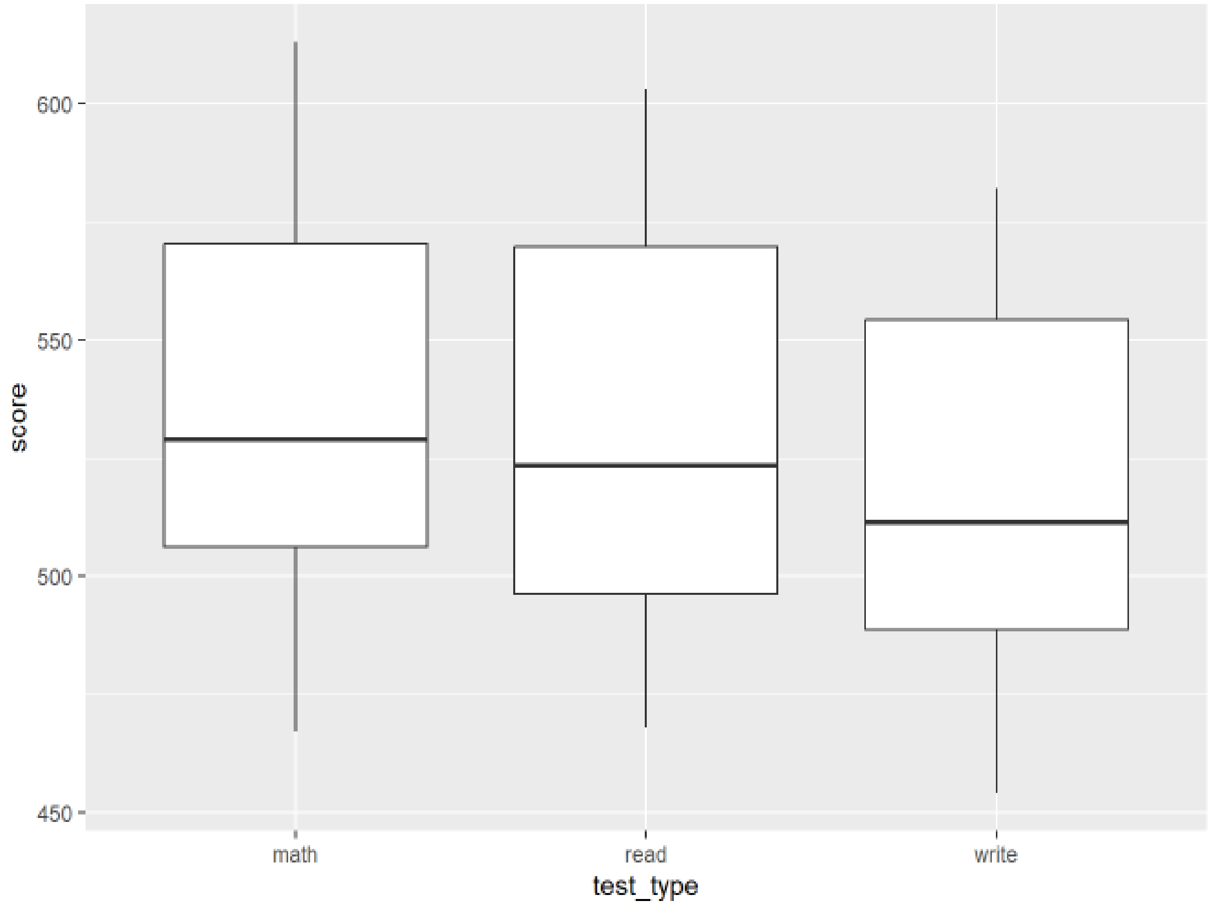
Plot # 3


1999
2000
2001
count
400
300
200
100
0
20
40
60
80
age

Plot # 4

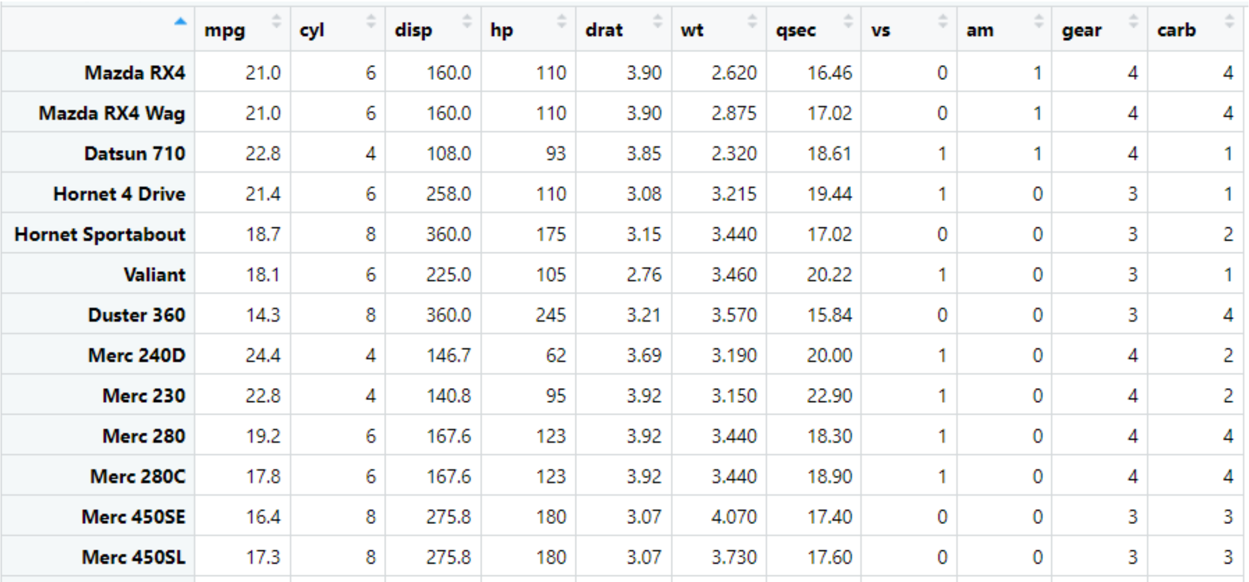

2003
2004
2005
2006
nruns
10000
7500
5000
2500
0
F
M
sex

Plot # 5



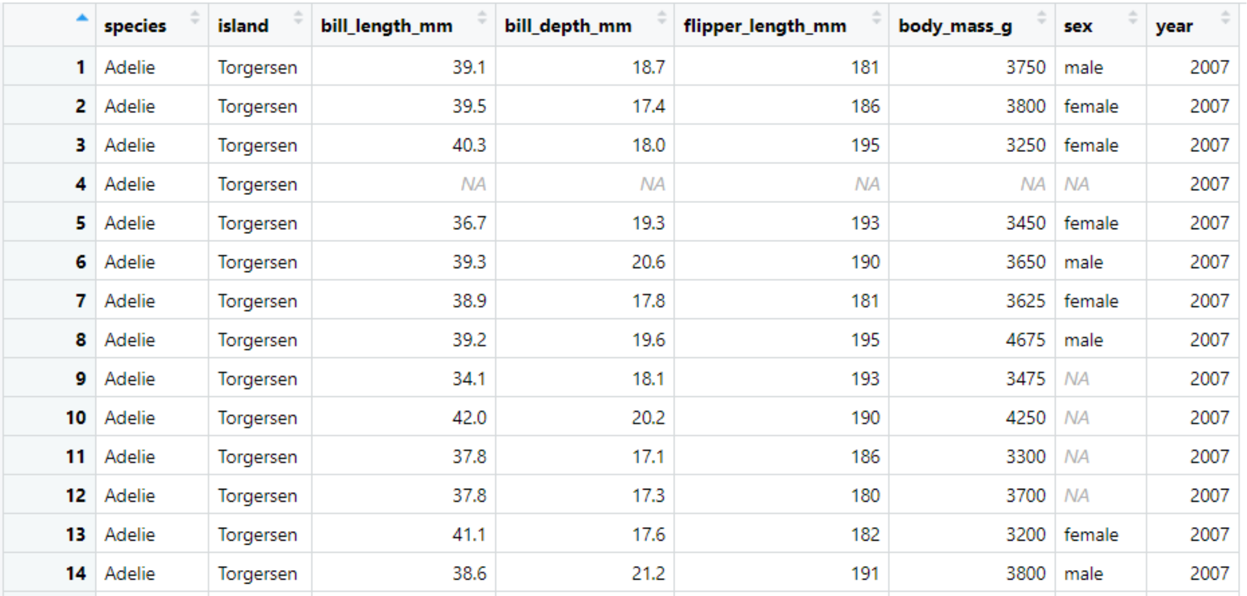
Data Visualization – Part 3

Data 1 - mtcars



	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3

Data 2 – penguins



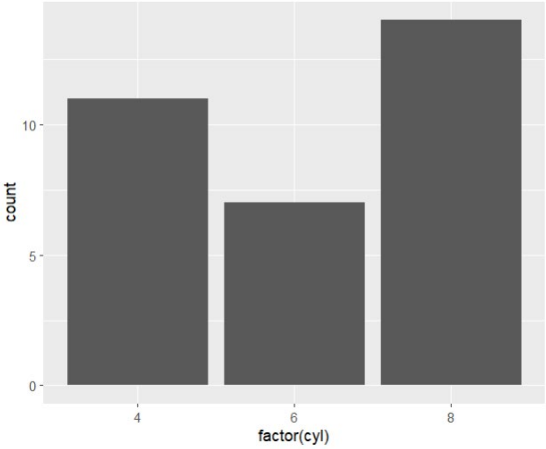
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
1	Adelie	Torgersen	39.1	18.7	181	3750	male	2007
2	Adelie	Torgersen	39.5	17.4	186	3800	female	2007
3	Adelie	Torgersen	40.3	18.0	195	3250	female	2007
4	Adelie	Torgersen	NA	NA	NA	NA	NA	2007
5	Adelie	Torgersen	36.7	19.3	193	3450	female	2007
6	Adelie	Torgersen	39.3	20.6	190	3650	male	2007
7	Adelie	Torgersen	38.9	17.8	181	3625	female	2007
8	Adelie	Torgersen	39.2	19.6	195	4675	male	2007
9	Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
10	Adelie	Torgersen	42.0	20.2	190	4250	NA	2007
11	Adelie	Torgersen	37.8	17.1	186	3300	NA	2007
12	Adelie	Torgersen	37.8	17.3	180	3700	NA	2007
13	Adelie	Torgersen	41.1	17.6	182	3200	female	2007
14	Adelie	Torgersen	38.6	21.2	191	3800	male	2007

Changing Colors in Plots

Univariate Displays

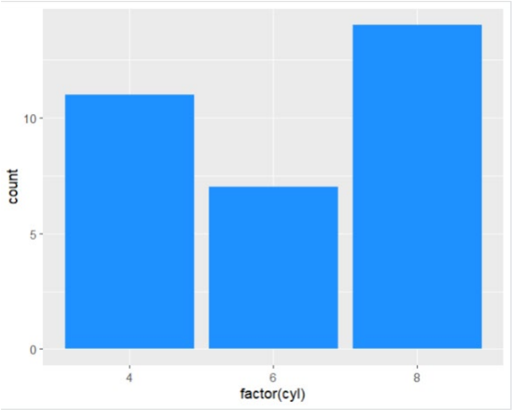
Example 1: Univariate simple bar graph (no color)

```
fig_1 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar()
fig_1
```



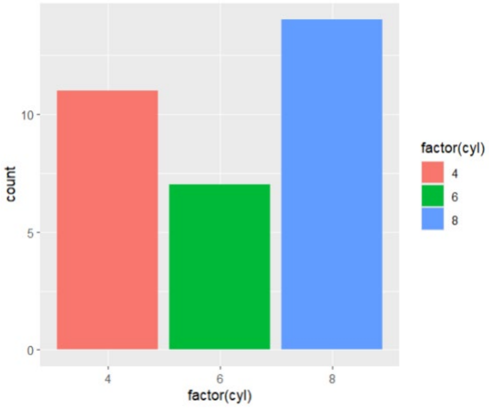
Example 2: Fills all the bars with the same color.

```
fig_2 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(fill = "dodgerblue")
fig_2
```



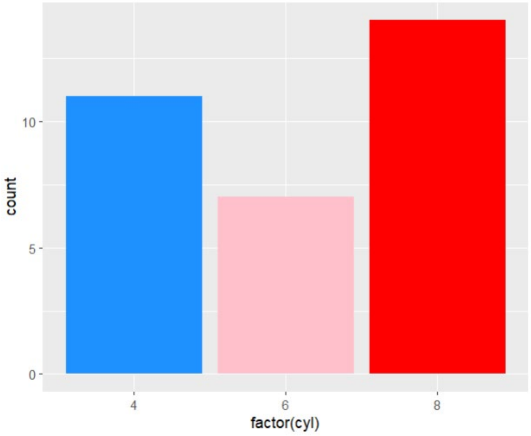
Example 3: Change each bar to a different color using fill inside "aes"

```
fig_3 <- mtcars %>%
  ggplot(aes(x = factor(cyl),
             fill = factor(cyl))) +
  geom_bar()
fig_3
```



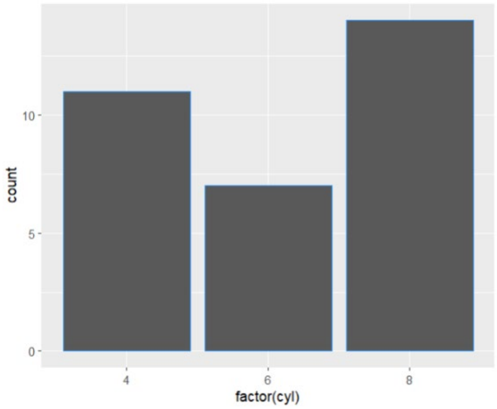
Example 4: Change each bar to a different color without the legend.

```
fig_4 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(fill = c("dodgerblue", "pink", "red"))
fig_4
```



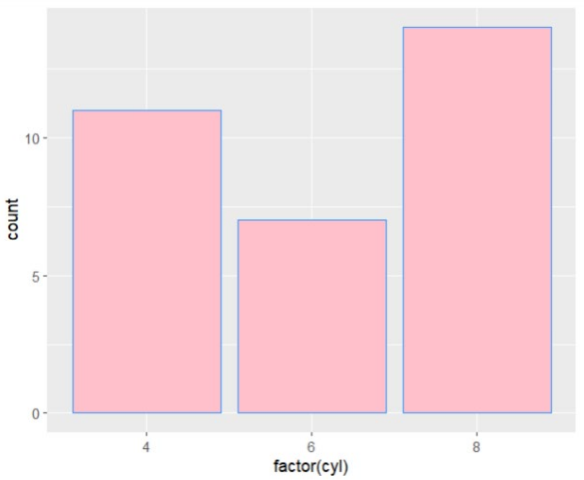
Example 5: Attempt to change bar color using color instead of fill. What happens?

```
fig_5 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(color = "dodgerblue")
fig_5
```



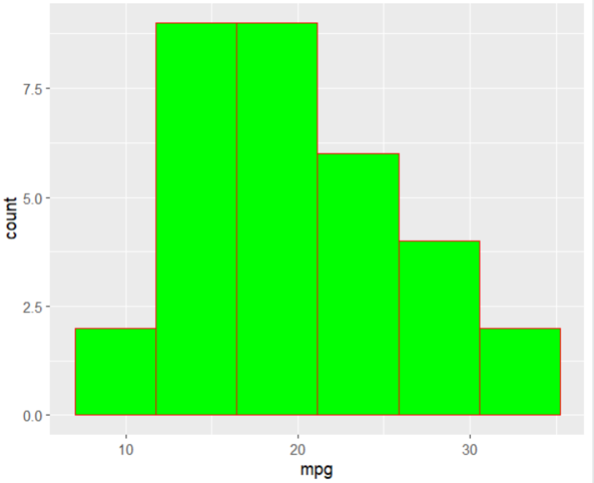
Example 6: Change the lines of the bars to dodgerblue and the bar colors to pink.

```
fig_6 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(fill = "pink", color = "dodgerblue")
fig_6
```



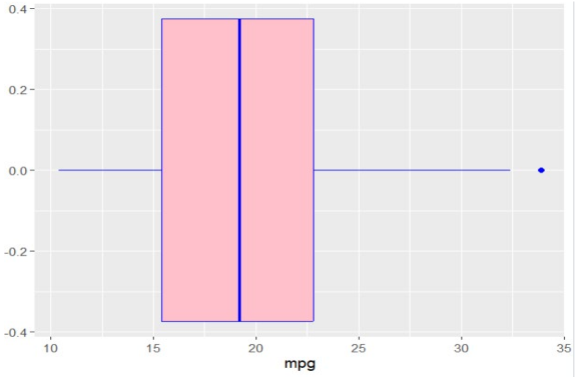
Example 7: Create a histogram for mpg variable with 6 bins. Then, change the lines of the histogram to red and the bar colors to green.

```
fig_7 <- mtcars %>%
  ggplot(aes(x = mpg)) +
  geom_histogram(bins = 6, fill = "green", color = "red")
fig_7
```



Example 8: Create a boxplot for mpg variable. Then, change the lines of the plot to blue and the inside colors to pink.

```
fig_8 <- mtcars %>%
  ggplot(aes(x = mpg)) +
  geom_boxplot(color = "blue", fill = "pink")
fig_8
```



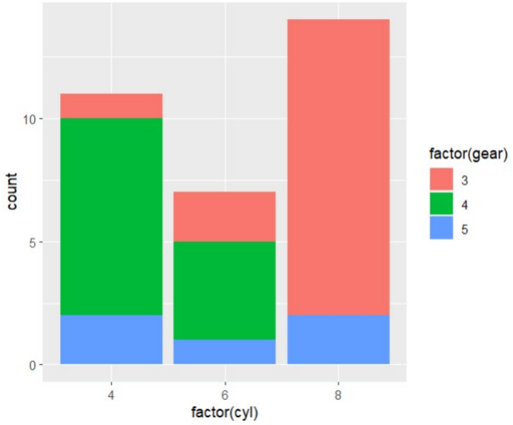
Multivariate Displays

For multivariate displays, you typically need to use color or fill inside the aes. To change the colors for a multivariable plot, you will need the functions:

- `scale_fill_manual()`: manually changes fill in the aes.
- `scale_color_manual()`: manually changes color in the aes.

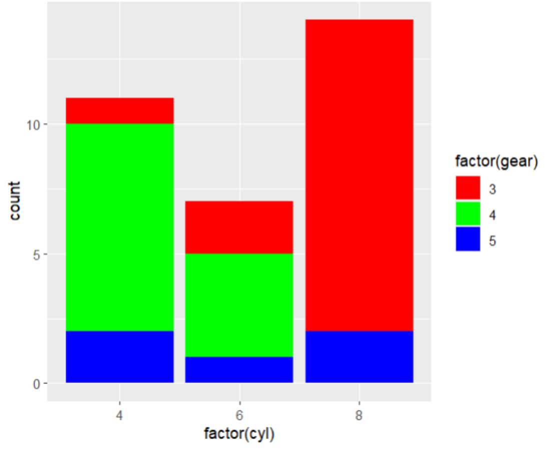
Example 9: Create a stacked bar graph with cyl in the x axis and gear as the categorical variable to split each column.

```
fig_9 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar()
fig_9
```



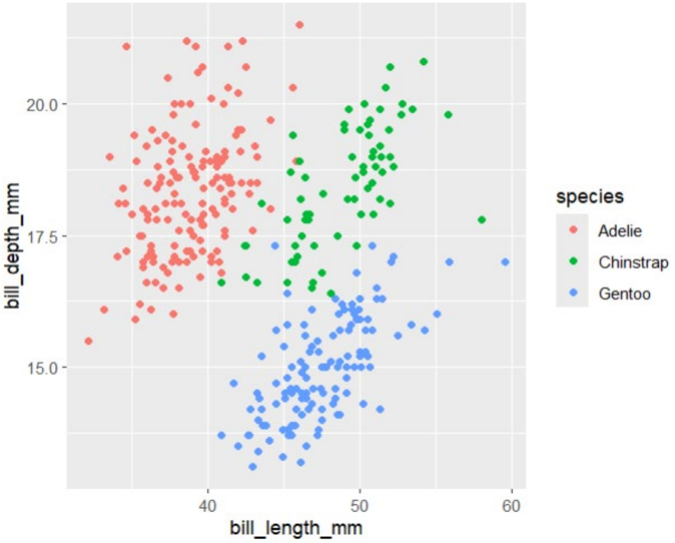
Example 10: Change the colors from Example 9 to red, green, and blue (in that order).

```
fig_10 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar()+
  scale_fill_manual(values = c("red", "green", "blue"))
fig_10
```



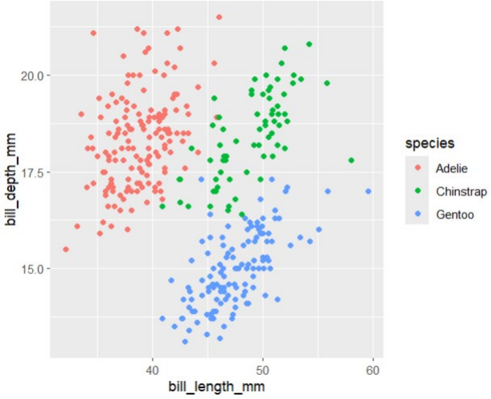
Example 11: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable.

```
fig_11 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point()
fig_11
```



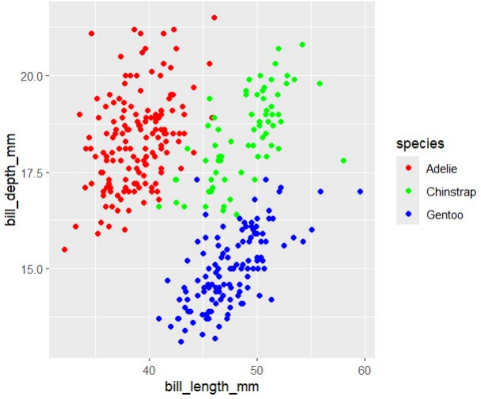
Example 12: Change the colors from Example 11 to red, green, and blue, in that order. What is wrong with these codes? Why isn't this one working?

```
fig_12 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point() +
  scale_fill_manual(values = c("red", "green", "blue"))
fig_12
```



Example 13: Change the colors from Example 11 to red, green, and blue, in that order.

```
fig_13 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue"))
fig_13
```

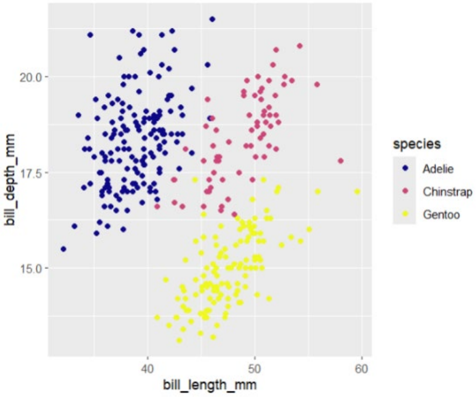


Using External Color Packages

Sometimes, you may want to use specialized color palettes from external packages to enhance your data visualizations. Packages like `wesanderson` and `viridis` provide aesthetically pleasing color palettes that can make your plots more visually appealing.

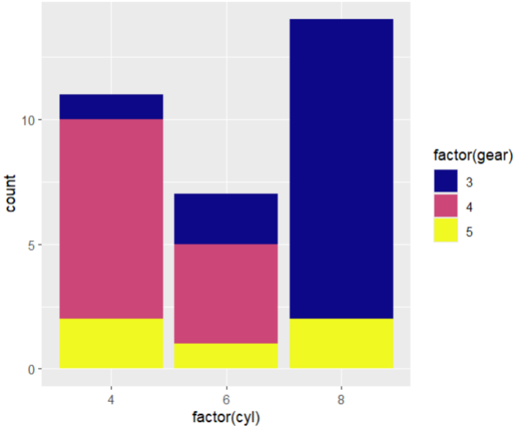
Example 14: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use the viridis package to change the colors.

```
fig_14 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_viridis(discrete = TRUE, option = "C", alpha = 1)
fig_14
```



Example 15: Create a stacked bar graph with cyl in the x axis and gear as the categorical variable to split each column. Use the viridis package to change the colors.

```
fig_15 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear))+
  geom_bar() +
  scale_fill_viridis(discrete = TRUE, option = "C", alpha = 1)
fig_15
```



Themes

1. `theme_gray()`:

   - This is the default theme in ggplot2.
   - It uses a simple gray background with white gridlines.
   - A good choice when you want a clean, minimalist look for your plot.

2. `theme_bw()`:

   - This theme provides a white background with black gridlines.
   - It offers a high-contrast, black-and-white appearance.
   - Useful for creating plots that need to be easily readable in black and white.

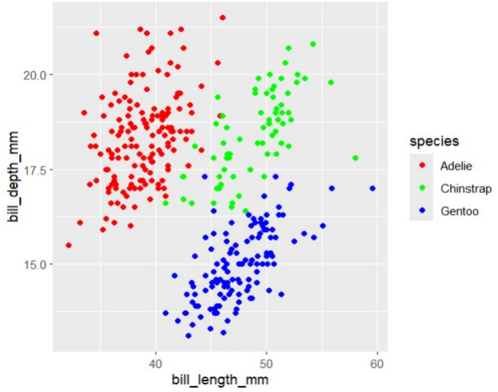
3. `theme_minimal()`:

   - As the name suggests, this theme is minimalistic.
   - It removes gridlines and most background elements, leaving a clean, white canvas.
   - Suitable for plots where you want to focus on the data without distractions.

4. `theme_void()`:

   - This theme removes nearly all elements, providing a blank canvas.
   - It's useful when you want to start with a clean slate and add custom elements.

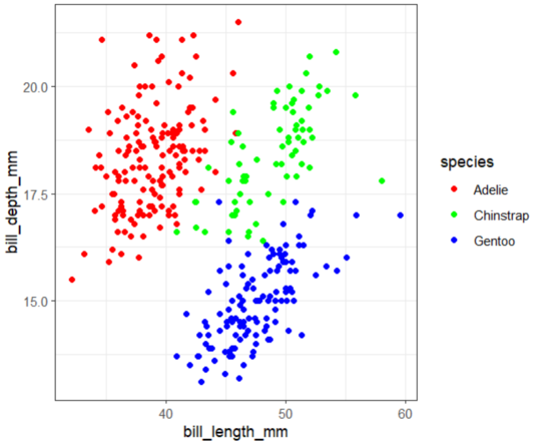
Example 16: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a gray theme



```r
fig_16 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_gray()
fig_16
```

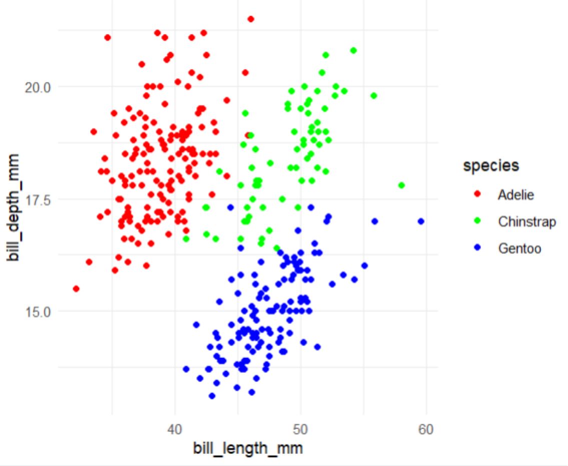
Example 17: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a black and white theme

```
fig_17 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_bw()
fig_17
```



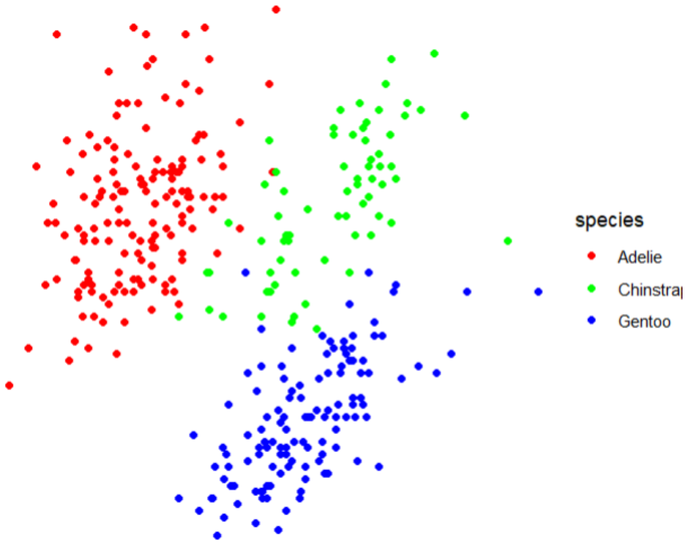
Example 18: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a minimal theme.

```
fig_18 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_minimal()
fig_18
```



Example 19: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a void theme.

```
fig_19 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_void()
fig_19
```

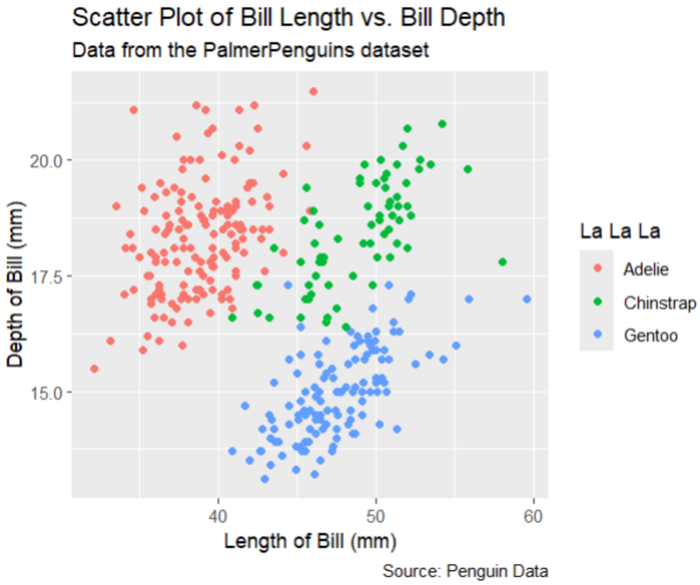


Customizing Labels with labs()

The `labs()` function in ggplot2 allows you to customize plot labels, including titles, subtitles, captions, and axis labels. You can use `labs()` to change the text displayed in various parts of your plot.

Example 20: Customizing Plot Labels

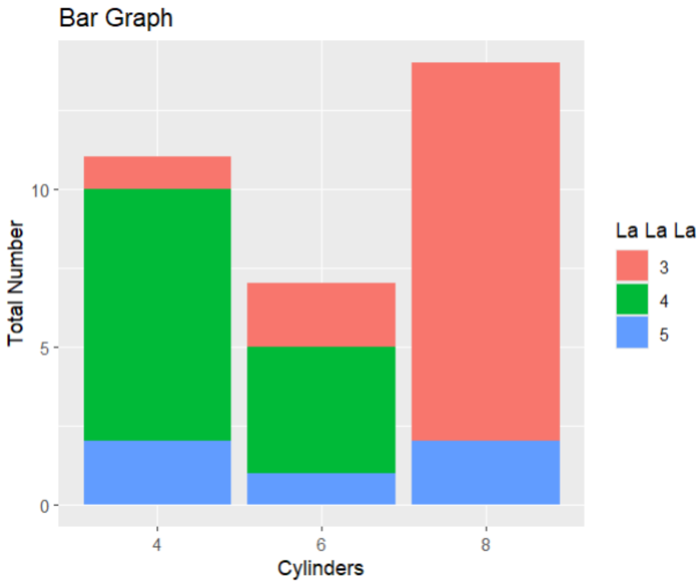
```
fig_20 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  labs(
    title = "Scatter Plot of Bill Length vs. Bill Depth",
    x = "Length of Bill (mm)",
    y = "Depth of Bill (mm)",
    subtitle = "Data from the PalmerPenguins dataset",
    caption = "Source: Penguin Data",
    color = "La La La"
  )
fig_20
```



Example 21: Customizing Plot Labels

```
fig_21 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar() +
  labs(
    title = "Bar Graph",
    x = "Cylinders",
    y = "Total Number",
    fill = "La La La"
  )

fig_21
```


Bar Graph
Total Number
Cylinders
La La La