Plotting Tutorial

Koko Friansa

8 November-2024

(Sekolah Bisnis Manajemen ITB)

Visualization in science

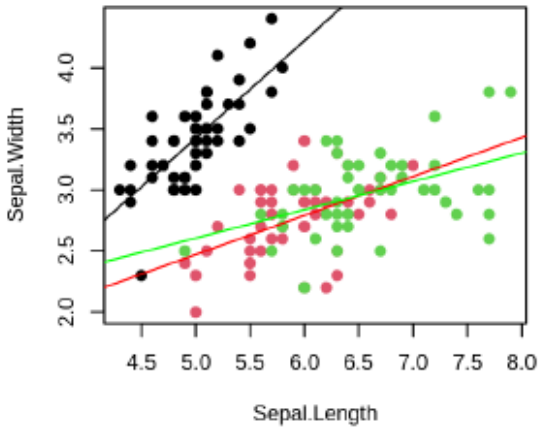
- **Important questions:**
  - What do you want to communicate?
  - Who is your audience?
  - What is the best way to do it?

Visualization in Science

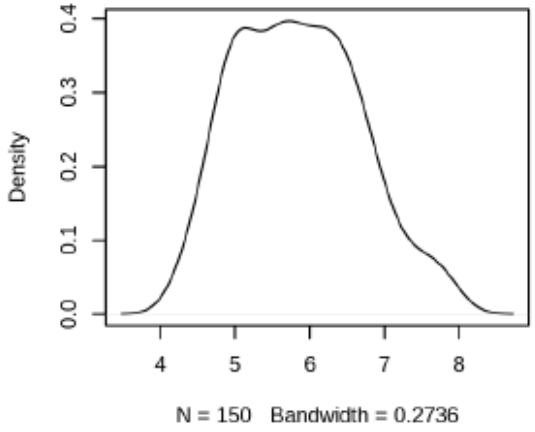
- 1. To represent results of statistical analyses
- 2. To formulate hypotheses and understand summarize theory
- 3. To explore your own data (exploratory analysis, outlier detection)
- 4. To communicate and report
  - Clearly (using good design principles)
  - Precisely and accurately (a plot is worth 1000 words)
  - Effectively and efficiently

Type of Visualization

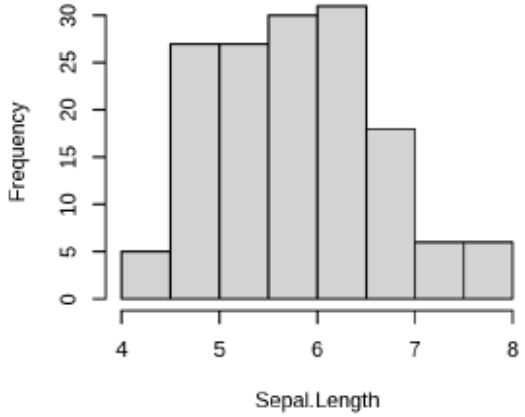
Scatter plot



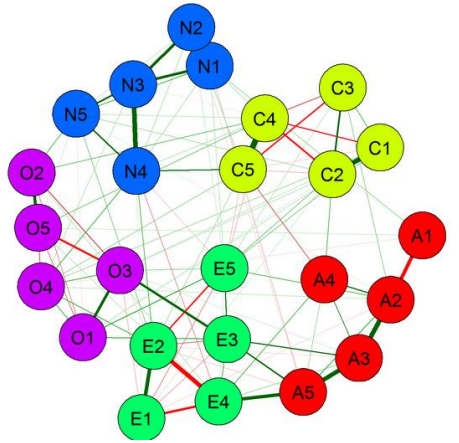
Density plot



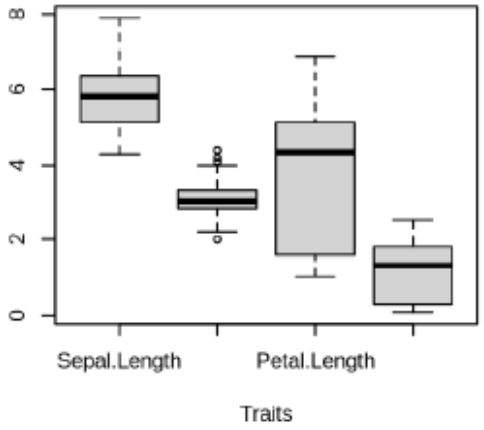
Histogram



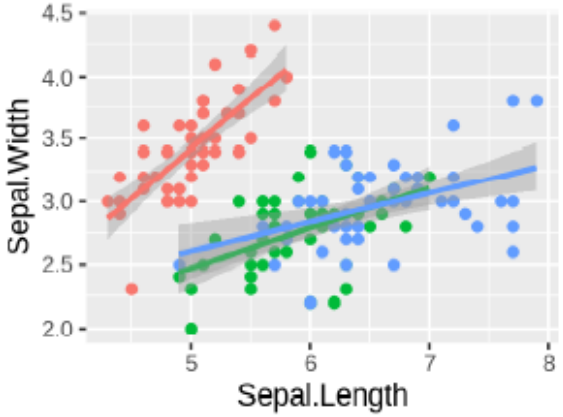
Network



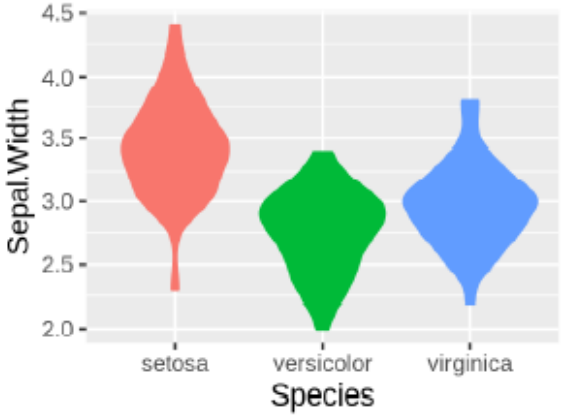
Boxplot



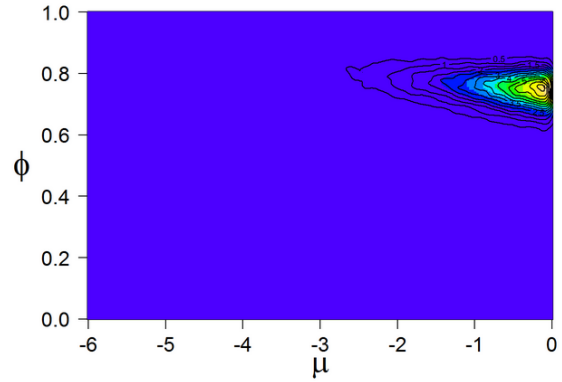
Scatter plot



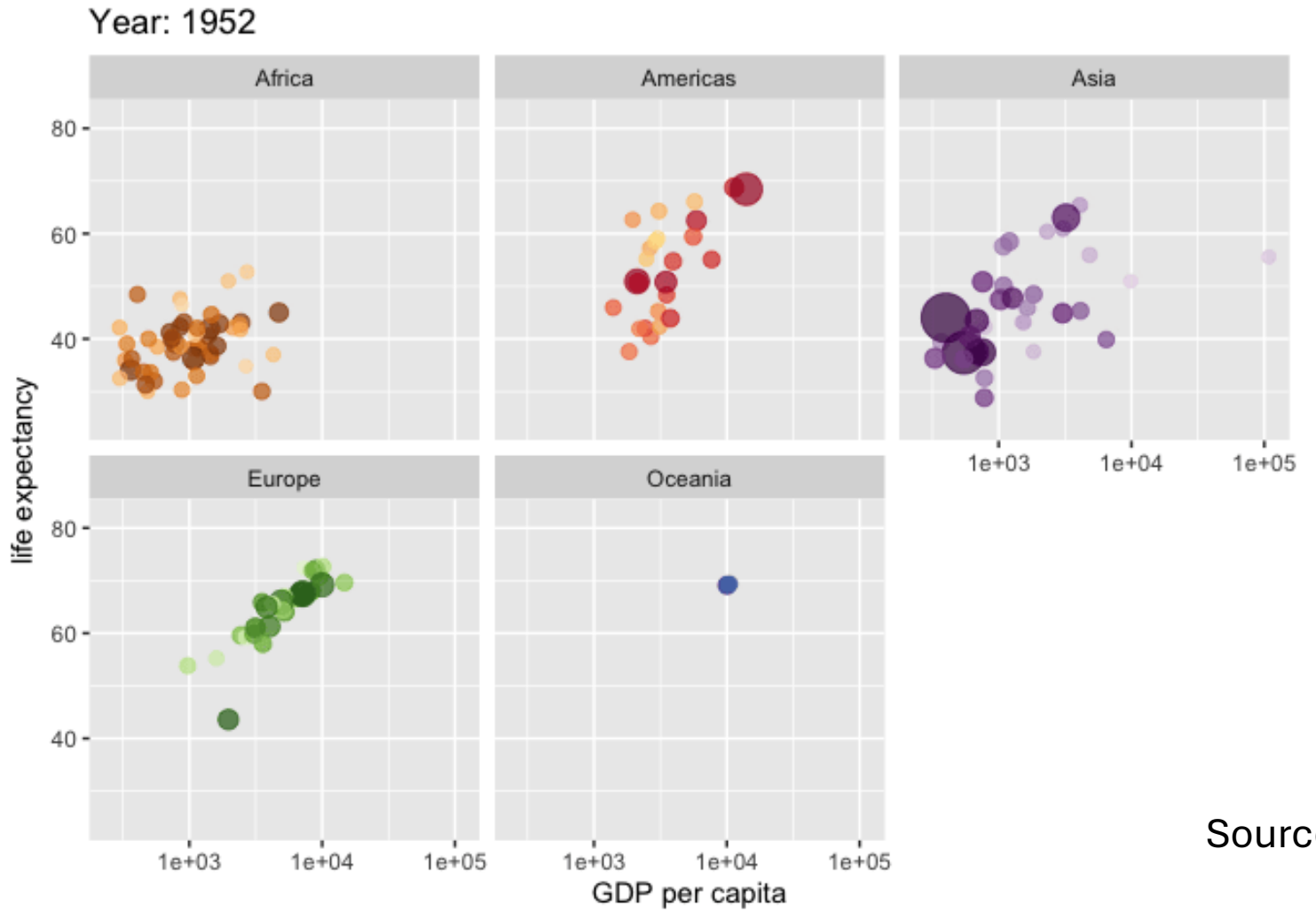
Violin plot



Heatmap and contour



Another Example: Animation

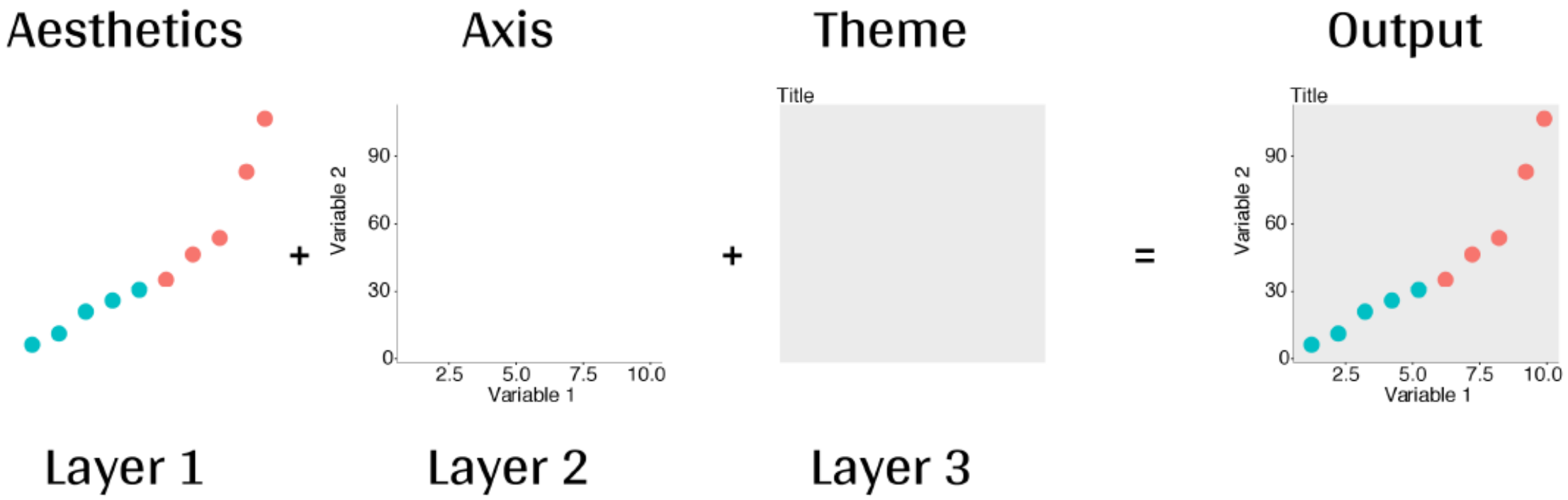


Source: gganimate.com

Resources

- https://www.shinyapps.org/apps/RGraphCompendium/index.php#line-plots
- https://wiki.qcbs.ca/r/workshop3
- https://bookdown.org/rdpeng/exdata/the-base-plotting-system-1.html
- https://plot.ly/r/
- https://r-spatial.github.io/mapview/
- https://exts.ggplot2.tidyverse.org/gallery/
- https://rstudio-pubs-static.s3.amazonaws.com/118501_94f1b69a4c7e4c5a842e1245228e4cc6.html
- https://mode.com/blog/r-ggplot-extension-packages/

ggplot2 Mechanics

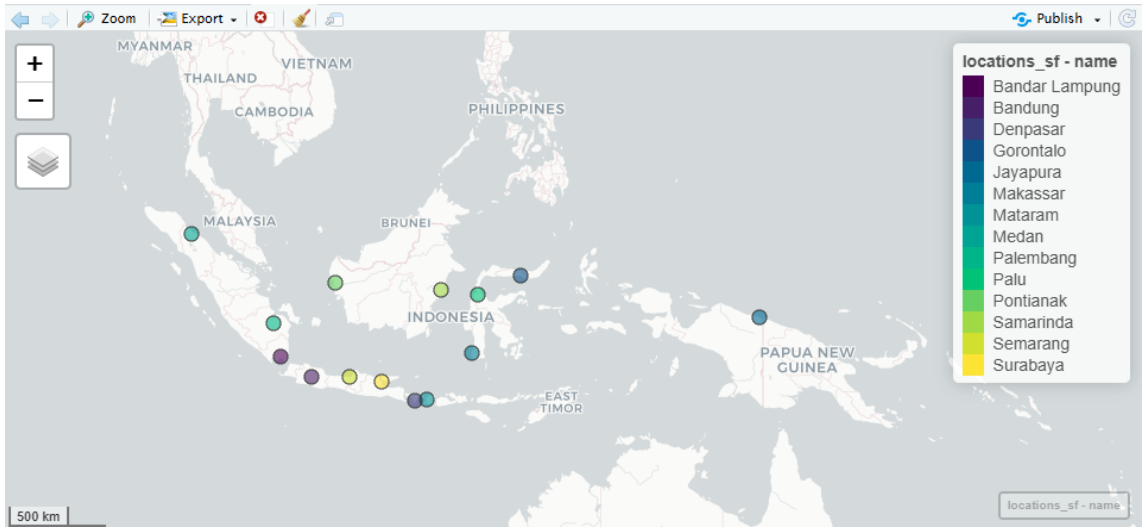


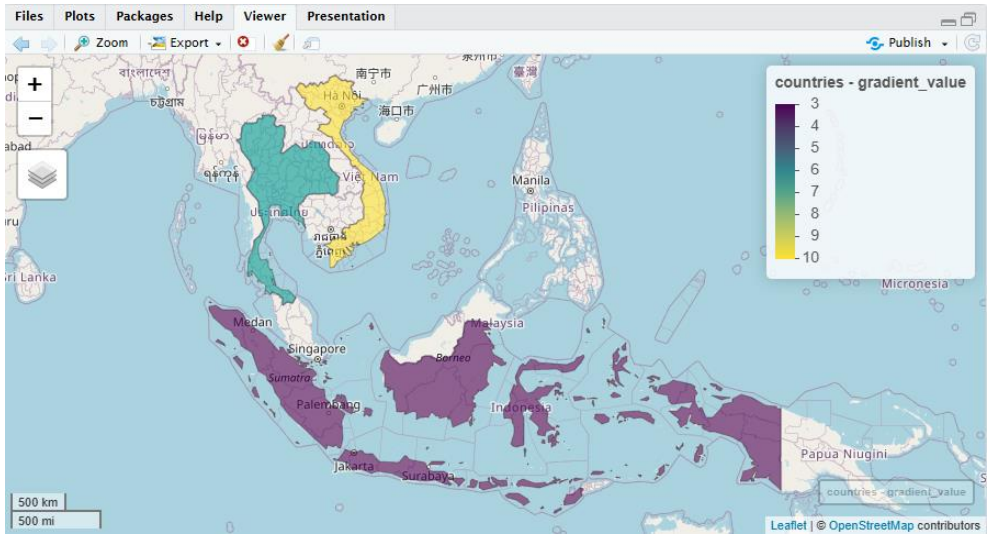
Great Multiple Package

https://exts.ggplot2.tidyverse.org/gallery/

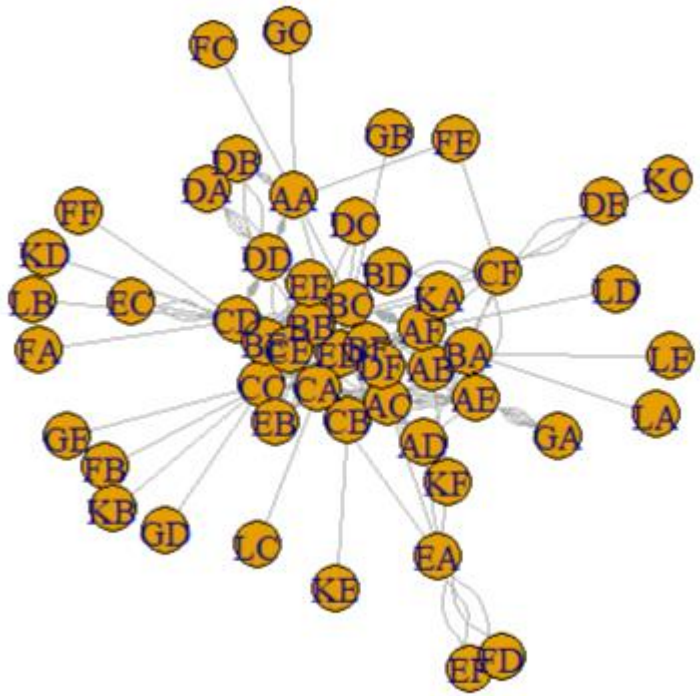
- ggstatsplot
- gganimate
- esquisse
- hrbrthemes
- ggrepel
- ggpubr
- ggraph
- ggforce
- ggdist
- cowplot
- igraph
- sjPlot
- mapview
- And more....

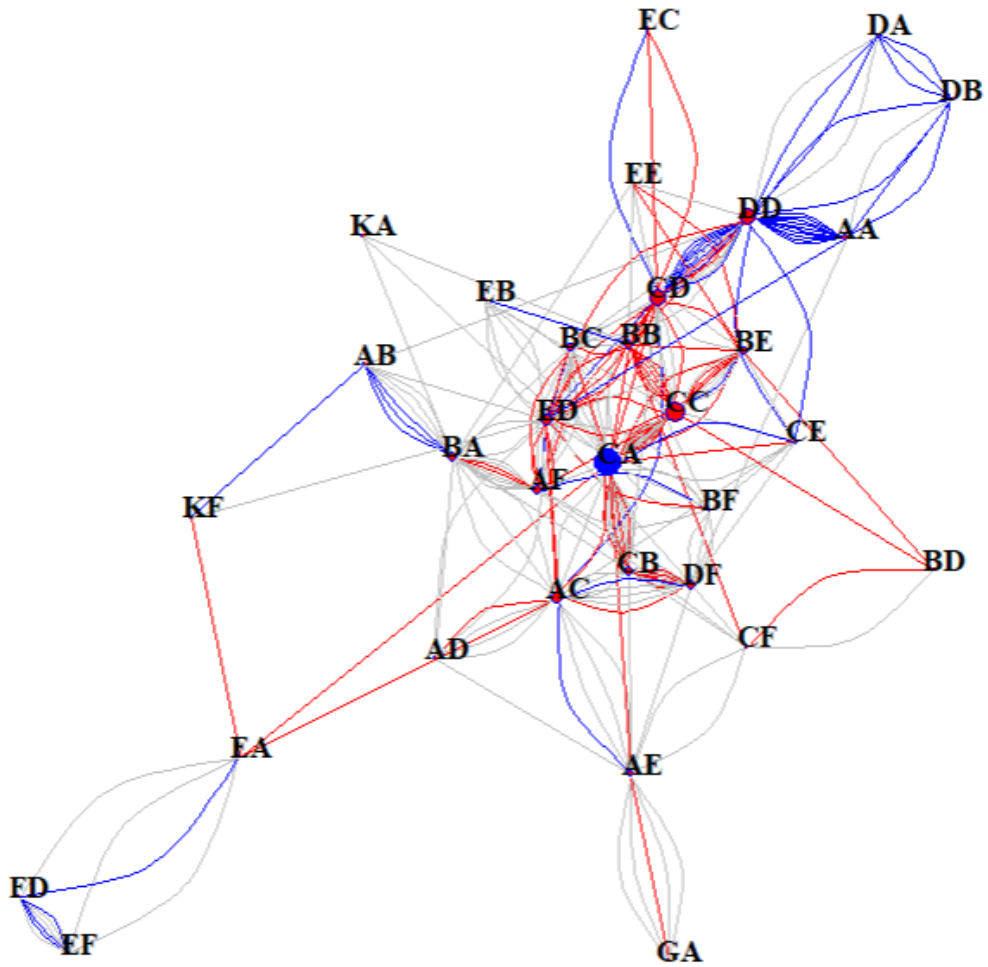
mapview Package



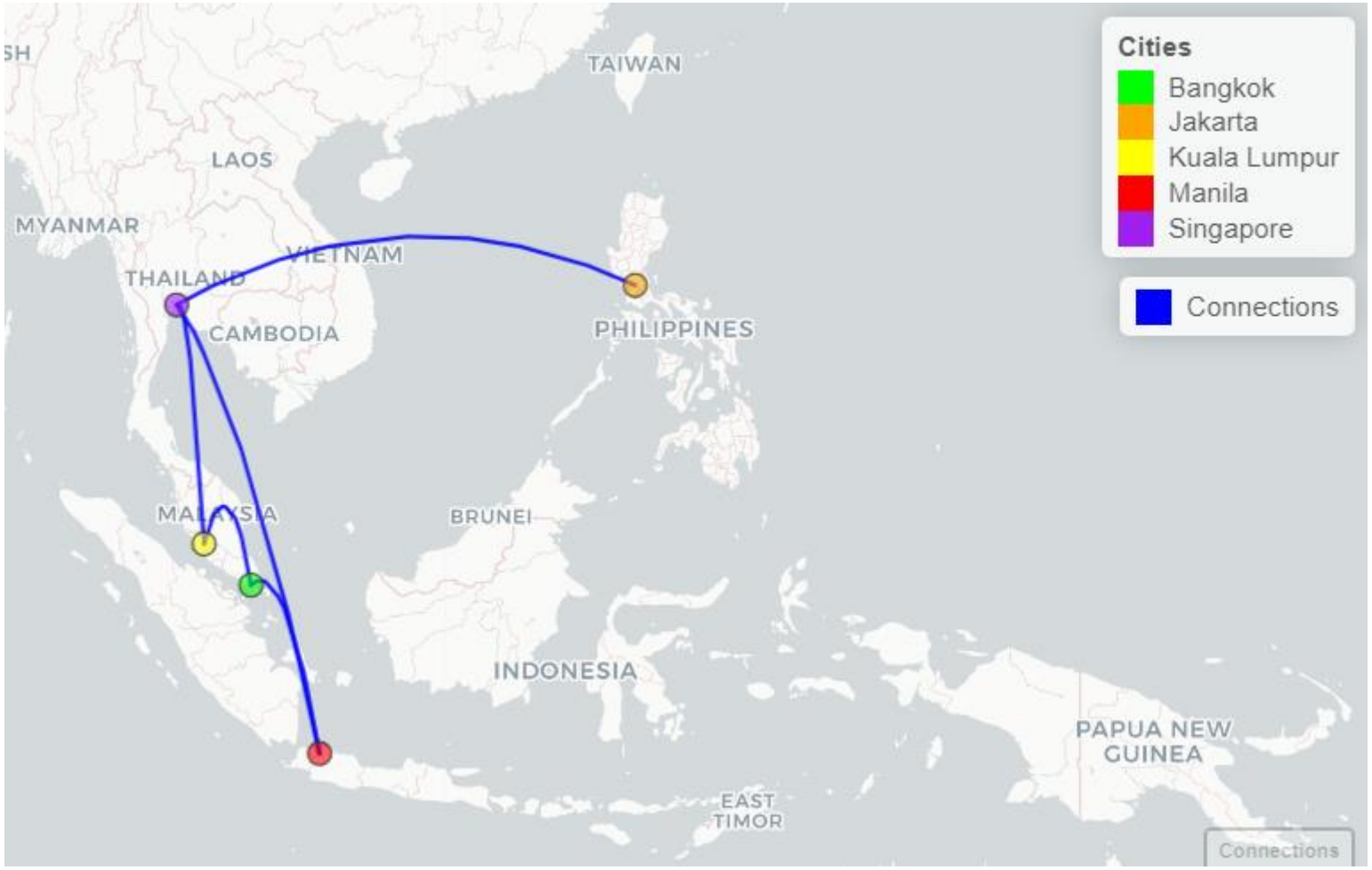


igraph Package





Combine Package: mapview and igraph?



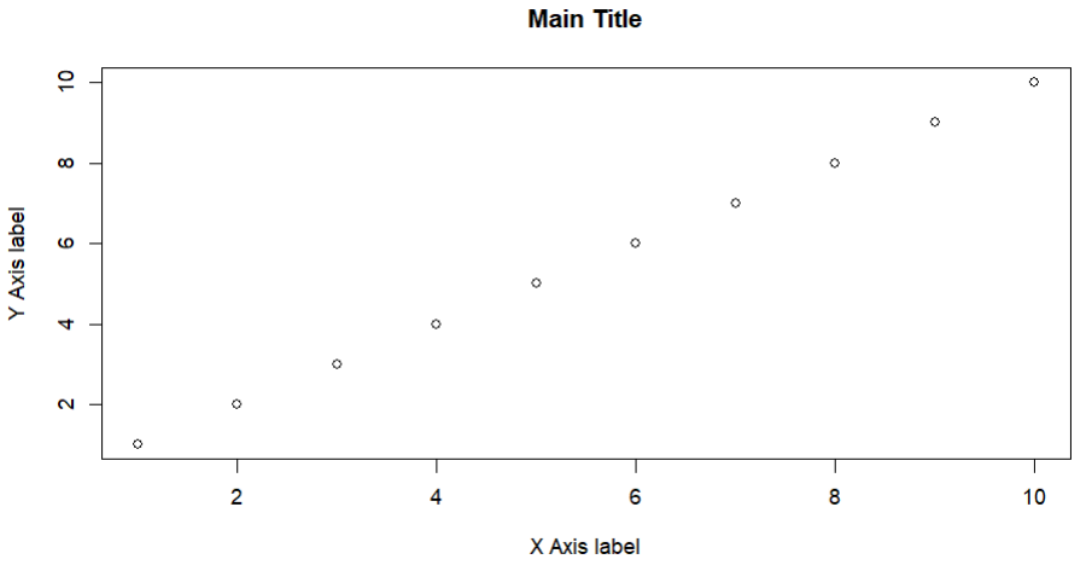
Plotting

1. Plot Basic

Plot.R

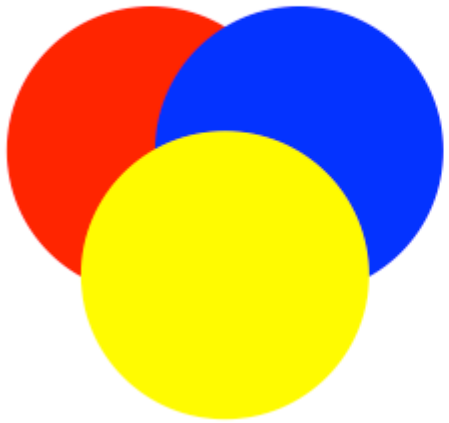
Source

```
1  # A basic scatterplot
2  plot(x = 1:10,
3       y = 1:10,
4       xlab = "X Axis label",
5       ylab = "Y Axis label",
6       main = "Main Title")
```



Color

Standard



Transparent

palegreen1	deeppink3	yellowgreen	gray100	orchid3	gray66	grey30	cyan3	azure4	lightskyblue1
tomato3	thistle4	whitesmoke	sienna	bisque3	grey70	lightpink	gold	gray19	lightgreen
gray89	gray40	grey74	royalblue3	tan4	honeydew2	orange	magenta	mistyrose4	chocolate1
grey16	khaki4	salmon4	lightblue3		gray59			gold3	lightcyan4
gray48	deepskyblue	gold1	gray14	grey96		darkgoldenrod	floralwhite	grey97	snow4
gray52	peachpuff3	mistyrose	orchid	hotpink3	grey40	midnightblue	pink4	dimgrey	gray34
grey46	seashell3	gray65	slateblue2	lightskyblue4	red2	darkslategrey	lavenderblush3	springgreen3	darkgreen
grey81	magenta3	turquoise2	mediumturquois		darkslategray1	navajowhite2	red4	grey85	gray22
lightcyan	salmon2	gray28	green3	navyblue	lightskyblue	dodgerblue4	gray76	gray77	lightsteelblue3
gray50	gray17	honeydew	burlywood	grey45	grey55	papayawhip	gray88	grey94	darkslategray3

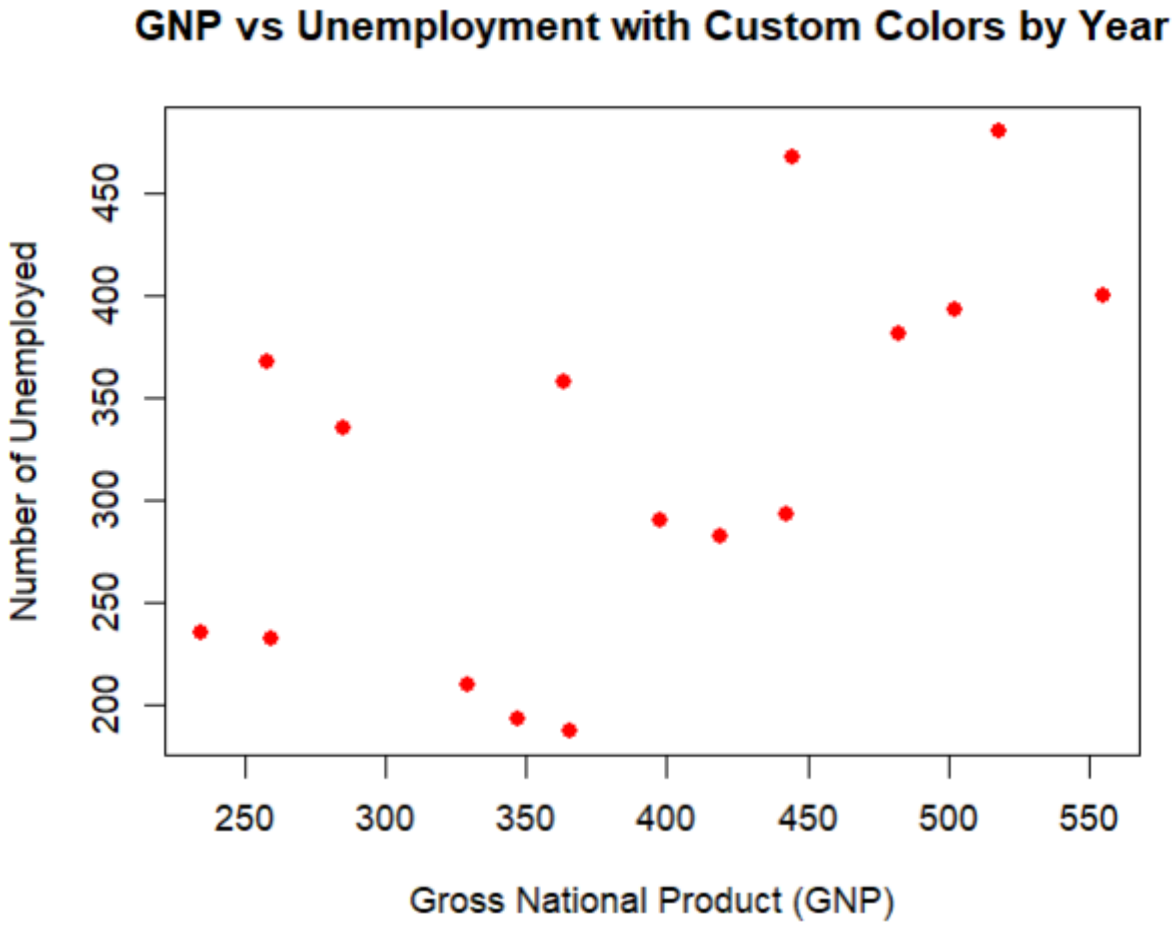
2. Color

Color.R

Source on Save | Run

```
1  data("longley")
2
3  plot(x = longley$GNP,
4       y = longley$Unemployed,
5       col = "red",
6       pch = 16,
7       main = "GNP vs Unemployment with Custom Colors by Year",
8       xlab = "Gross National Product (GNP)",
9       ylab = "Number of Unemployed")
10
```

pch = _

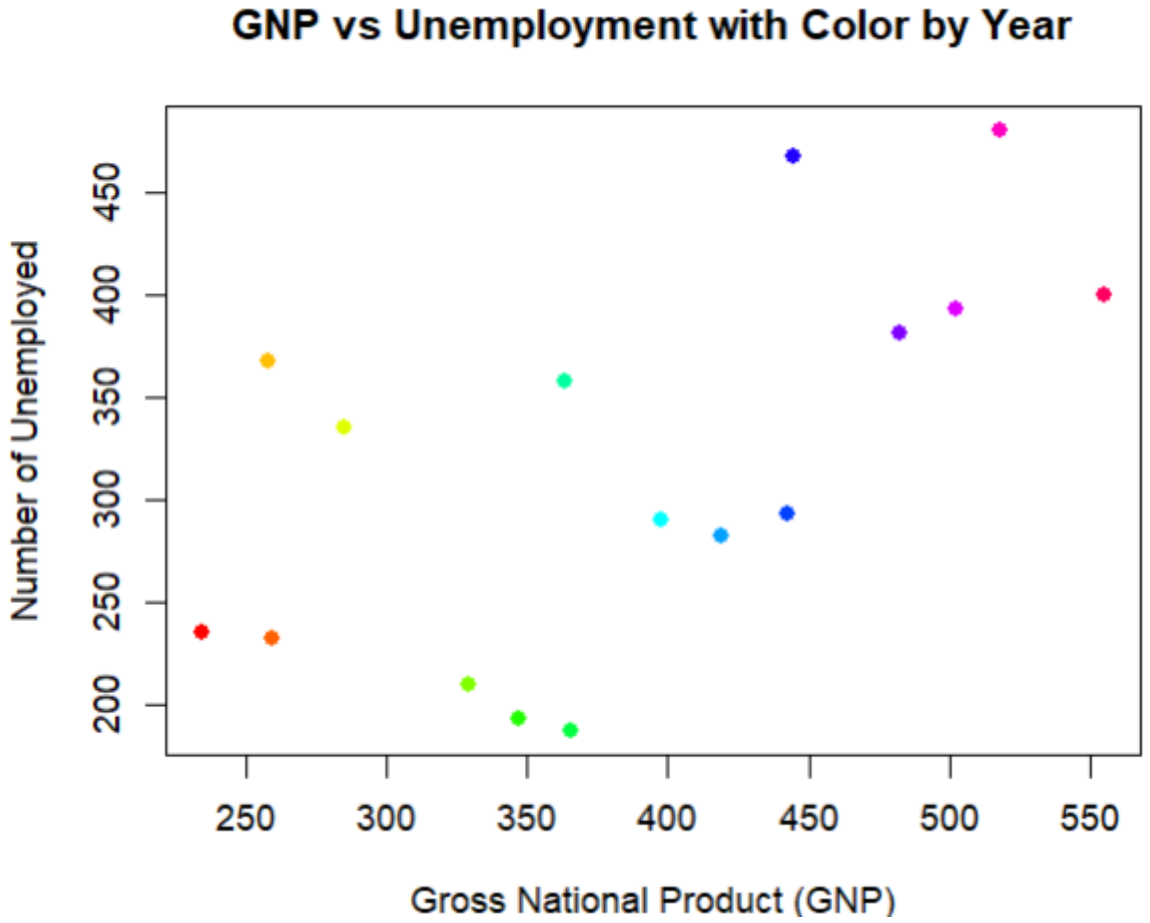


3. Rainbow

Color.R

Source on Save | Run | Source

```r
# Load the longley dataset (it's a built-in dataset, so no need to import)
data("longley")

# Create a color palette with unique colors for each year
colors <- rainbow(length(unique(longley$Year)))[as.factor(longley$Year)]

# Plot GNP vs Unemployed with colors representing Year
plot(x = longley$GNP,
     y = longley$Unemployed,
     col = colors,      # Apply the color palette to represent each year
     pch = 16,          # Solid circles
     main = "GNP vs Unemployment with Color by Year",
     xlab = "Gross National Product (GNP)",
     ylab = "Number of Unemployed")

```

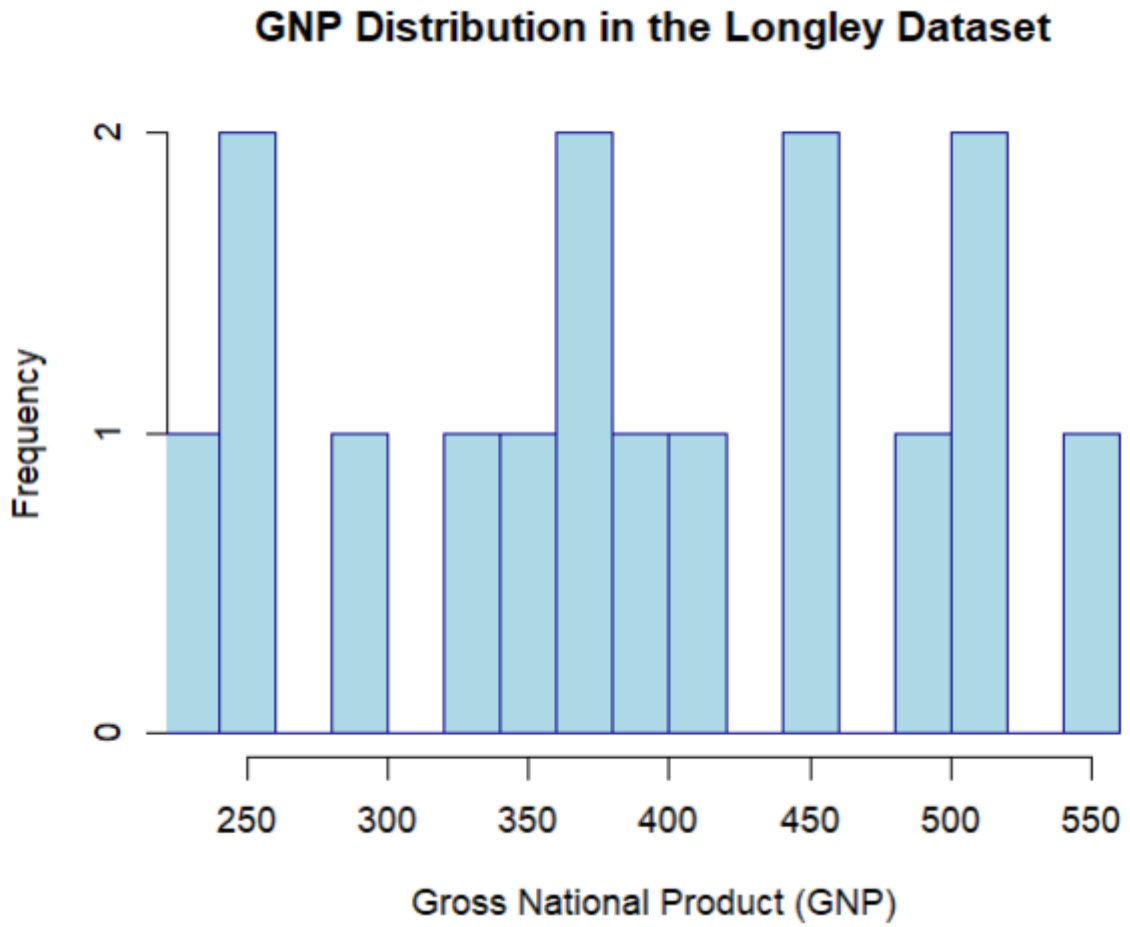


4. Histogram

histogram.R*

Source on Save | Run

```
1  data("longley")
2
3  hist(x = longley$GNP,
4       main = "GNP Distribution in the Longley Dataset",
5       xlab = "Gross National Product (GNP)",
6       ylab = "Frequency",
7       breaks = 20,          # 20 bins
8       xlim = c(min(longley$GNP), max(longley$GNP)),
9       col = "lightblue",
10      border = "darkblue")
11
12
```

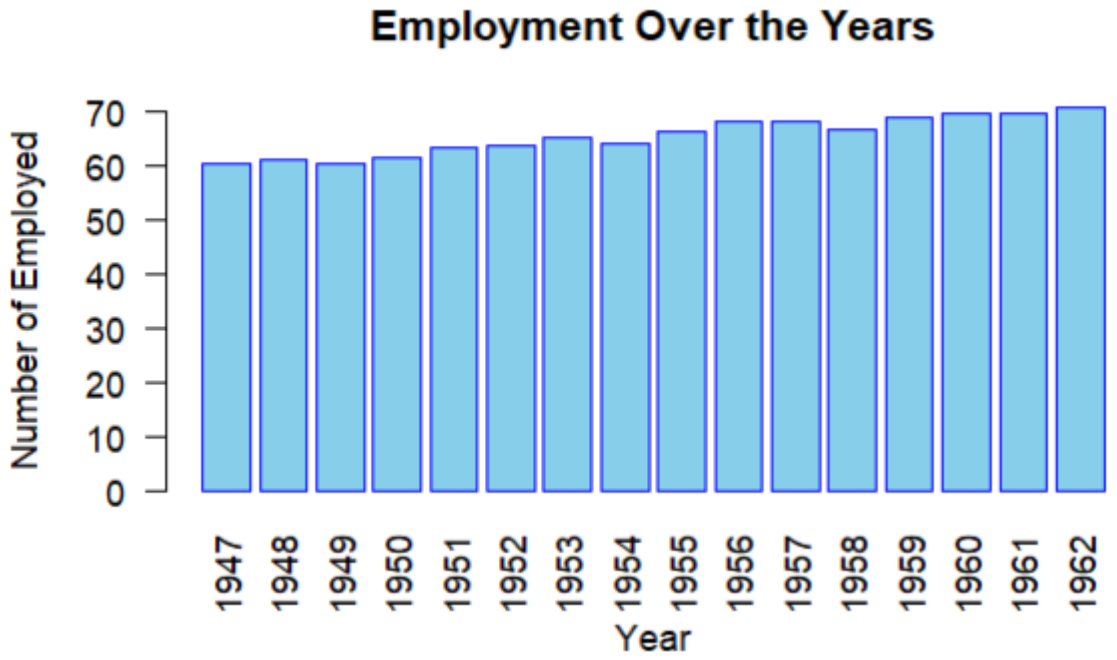


5. Barplot

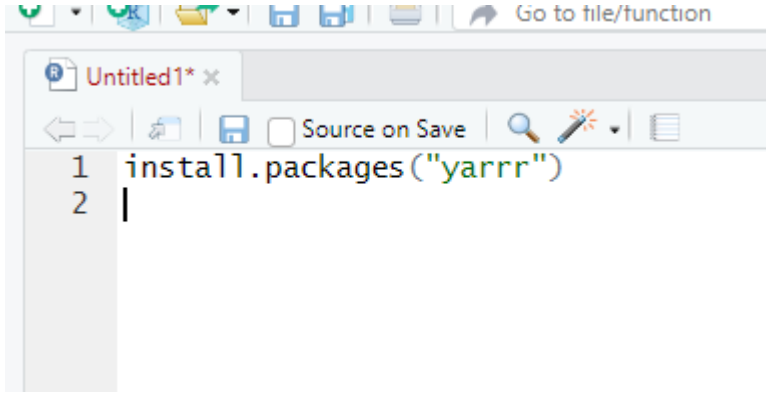
Barplot.R*

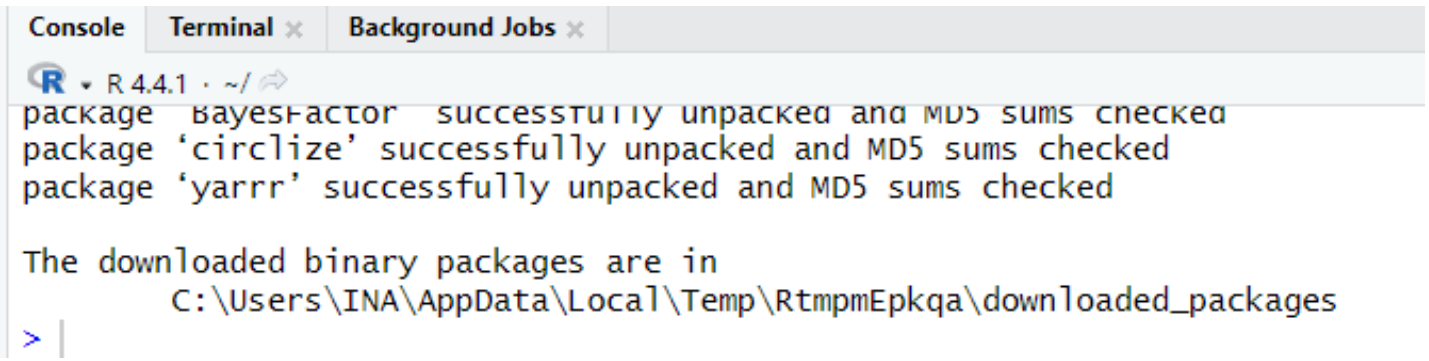
Source on Save | Run

```
1  data("longley")
2
3  barplot(height = longley$Employed,
4          names.arg = longley$Year,
5          main = "Employment Over the Years",
6          xlab = "Year",
7          ylab = "Number of Employed",
8          col = "skyblue",
9          border = "blue",
10         las = 2)
11
```



6. Yarr Package



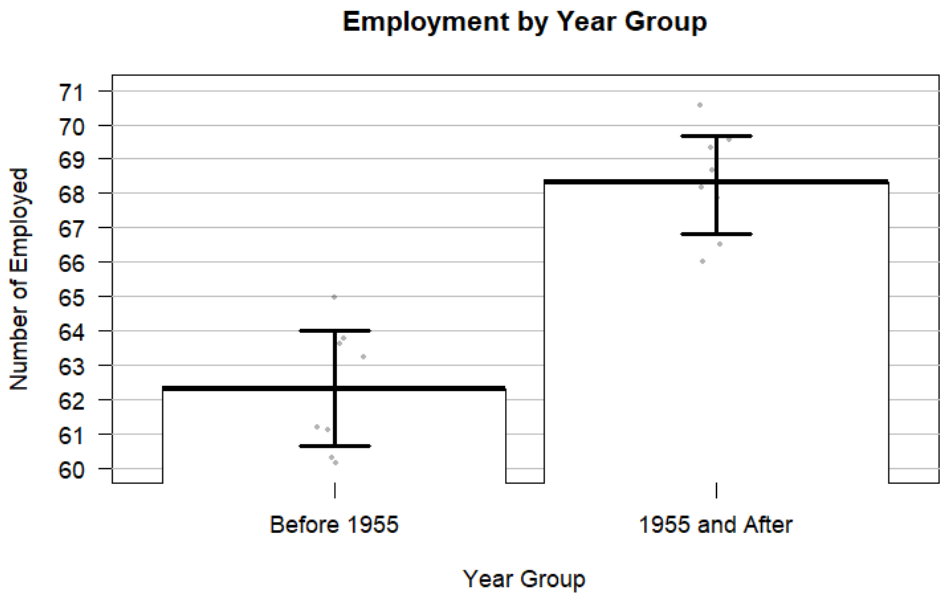


7. Pirate Plot

Pirate plot.R*

Source on Save | Run | Source

```r
library(yarrr)

data("longley")

longley$YearGroup <- ifelse(longley$Year < 1955, "Before 1955", "1955 and After")
longley$YearGroup <- factor(longley$YearGroup, levels = c("Before 1955", "1955 and After"))

# Create a pirate plot with a theme
pirateplot(formula = Employed ~ YearGroup,
           data = longley,
           theme = 4,
           main = "Employment by Year Group",
           xlab = "Year Group",
           ylab = "Number of Employed",
           pal = "info")
```

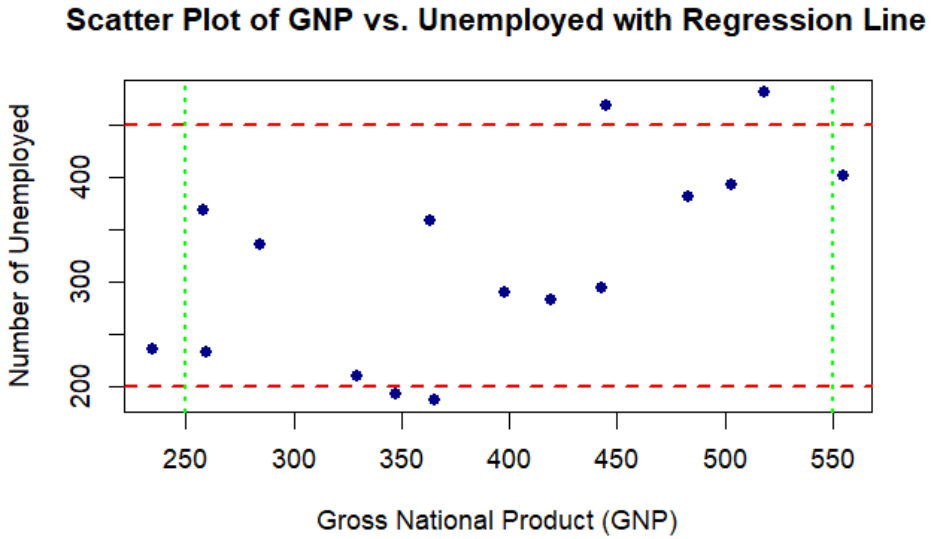


8. abline vertical and horizontal

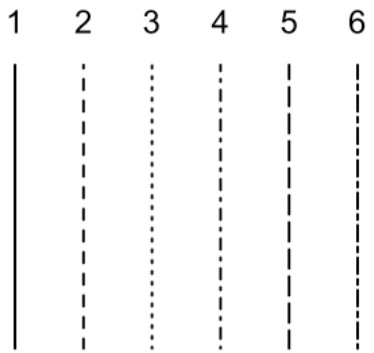
abline vertical.R*

Source on Save | Run

```r
# Load the longley dataset
data("longley")

# Create a scatter plot of GNP vs. Unemployed
plot(x = longley$GNP,
     y = longley$Unemployed,
     main = "Scatter Plot of GNP vs. Unemployed with Regression Line",
     xlab = "Gross National Product (GNP)",
     ylab = "Number of Unemployed",
     pch = 16,                  # Shape of the points
     col = "darkblue")          # Color of the points

abline(h = 450, col = "red", lwd = 2, lty = 2)  # Red dashed line
abline(h = 200, col = "red", lwd = 2, lty = 2)  # Red dashed line

abline(v = 250, col = "green", lwd = 2, lty = 3)  # Green dotted line
abline(v = 550, col = "green", lwd = 2, lty = 3)  # Green dotted line
```



lty = ...

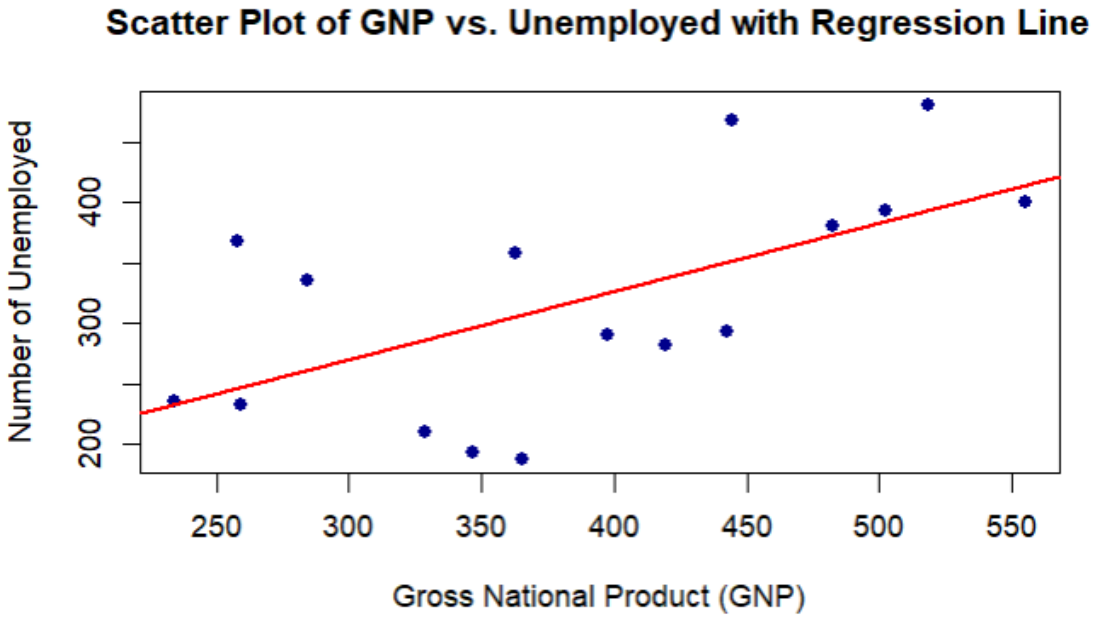


9. Abline regresi

abline.R*

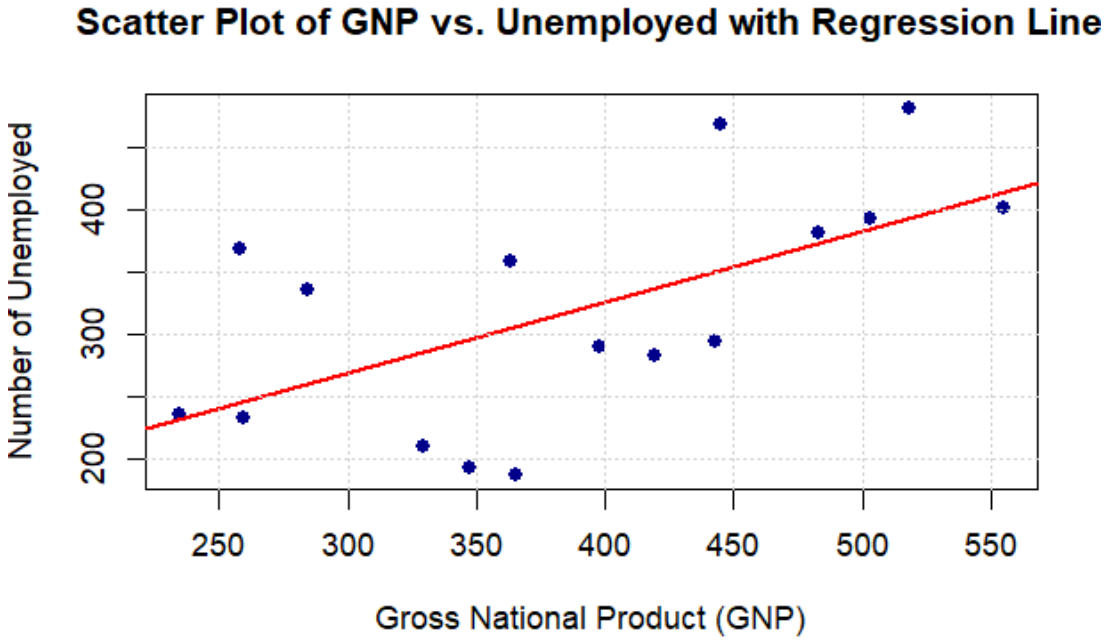
Source on Save | Run

```r
# Load the longley dataset
data("longley")

# Create a scatter plot of GNP vs. Unemployed
plot(x = longley$GNP,
     y = longley$Unemployed,
     main = "Scatter Plot of GNP vs. Unemployed with Regression Line",
     xlab = "Gross National Product (GNP)",
     ylab = "Number of Unemployed",
     pch = 16,                    # Shape of the points
     col = "darkblue")            # Color of the points

# Fit a linear model for Unemployed as a function of GNP
model <- lm(Unemployed ~ GNP, data = longley)

# Add a regression line based on the linear model
abline(model, col = "red", lwd = 2)  # lwd sets line thickness

```

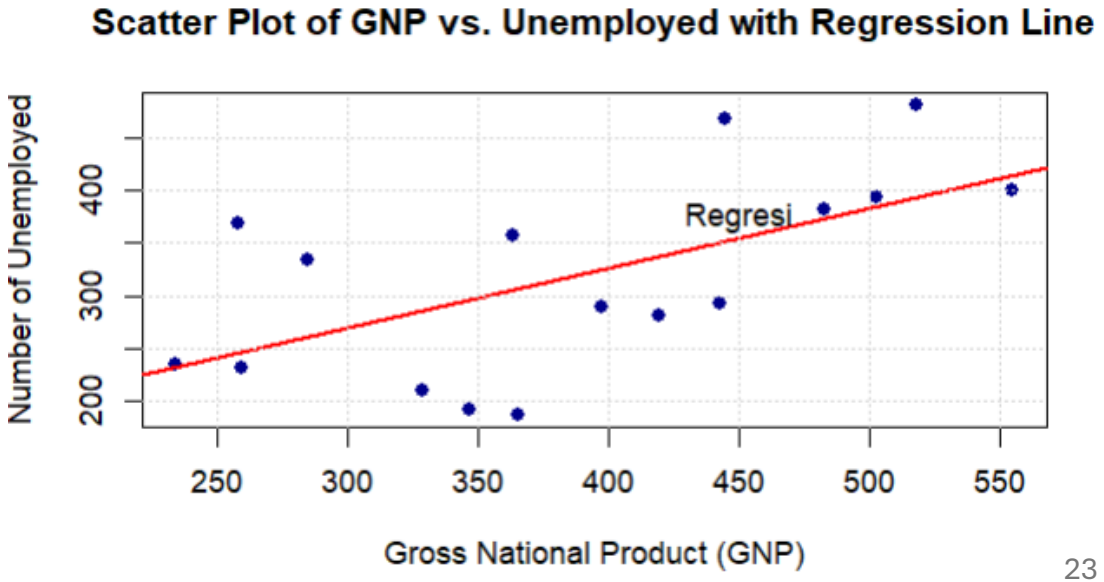


10. grid() and text()

```
17   abline(lmout
18
19   grid()
20
```



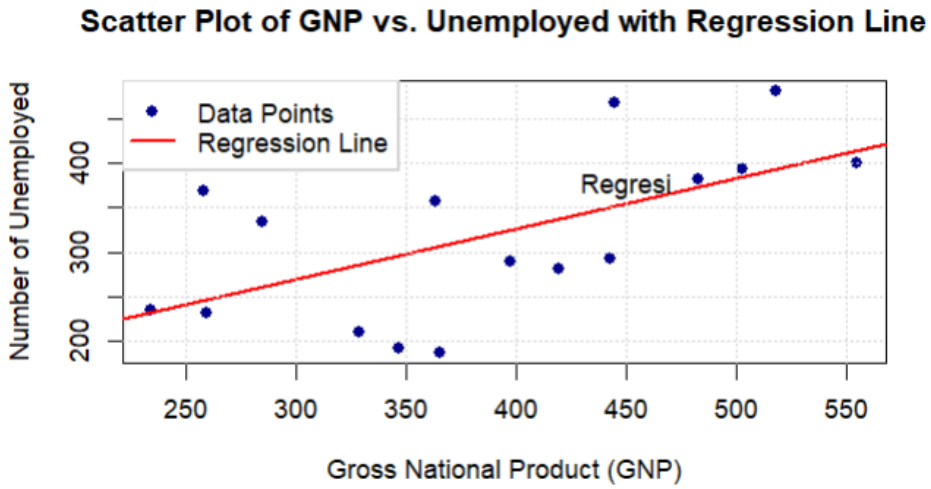

```
19  grid()
20
21  # Add id labels
22  text(x = 450,
23       y = 350,
24       labels = "Regresi",
25       pos = 3)
```



11. Legend

- Add this code, from previous slide

```
26
27  legend("topleft",
28         legend = c("Data Points", "Regression Line"),
29         col = c("darkblue", "red"),
30         pch = c(16, NA),
31         lty = c(NA, 1),
32         lwd = c(NA, 2),
33         bty = "1",
34         box.col = "lightgray",
35         box.lwd = 2)
```

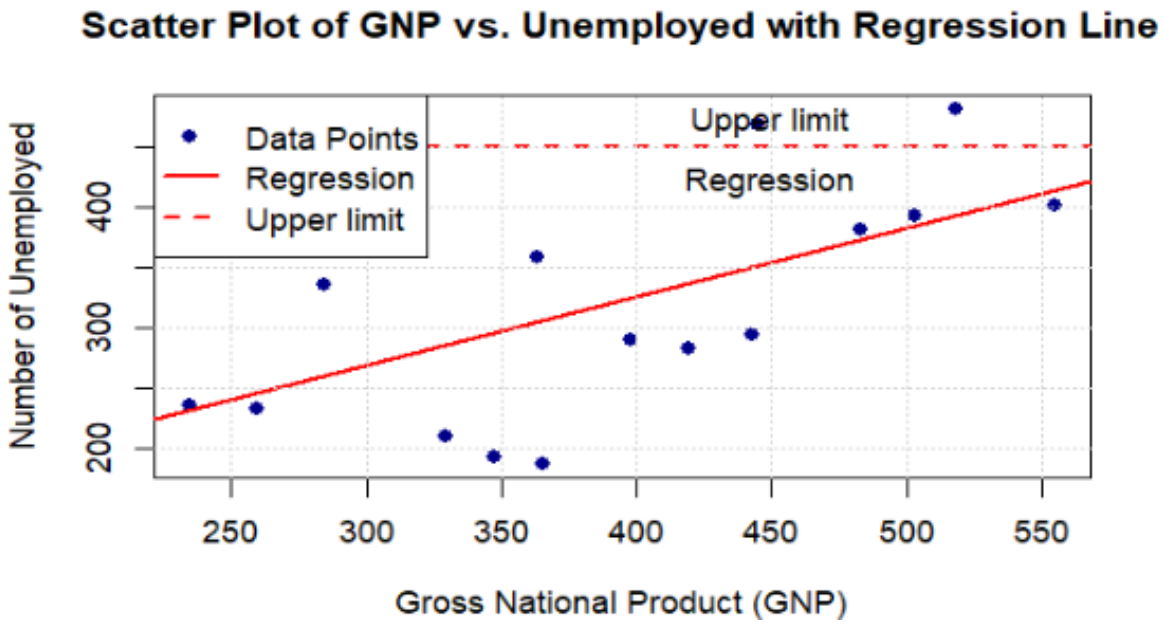


12. More legend and text

Segment.R | More legend.R* | abline vertical.R

Source on Save | Run

```
      xlab = "Gross National Product (GNP)",
      ylab = "Number of Unemployed",
      pch = 16,                  # Shape of the points
      col = "darkblue")          # Color of the points

# Fit a linear model for Unemployed as a function of GNP
model <- lm(Unemployed ~ GNP, data = longley)

# Add a regression line based on the linear model
abline(model, col = "red", lwd = 2)  # lwd sets line thickness

# Add a horizontal dashed line at y = 450
abline(h = 450, col = "red", lwd = 2, lty = 2)  # Red dashed line

# Add a grid for better readability
grid()

# Add ID labels for the regression line and the upper limit
text(x = 450, y = 400, labels = "Regression", pos = 3)
text(x = 450, y = 450, labels = "Upper limit", pos = 3)

# Add a legend
legend("topleft",
       legend = c("Data Points", "Regression", "Upper limit"),
       col = c("darkblue", "red", "red"),
       pch = c(16, NA, NA),
       lty = c(NA, 1, 2),
       lwd = c(NA, 2, 2),
       bty = "o")
```



ggplot2

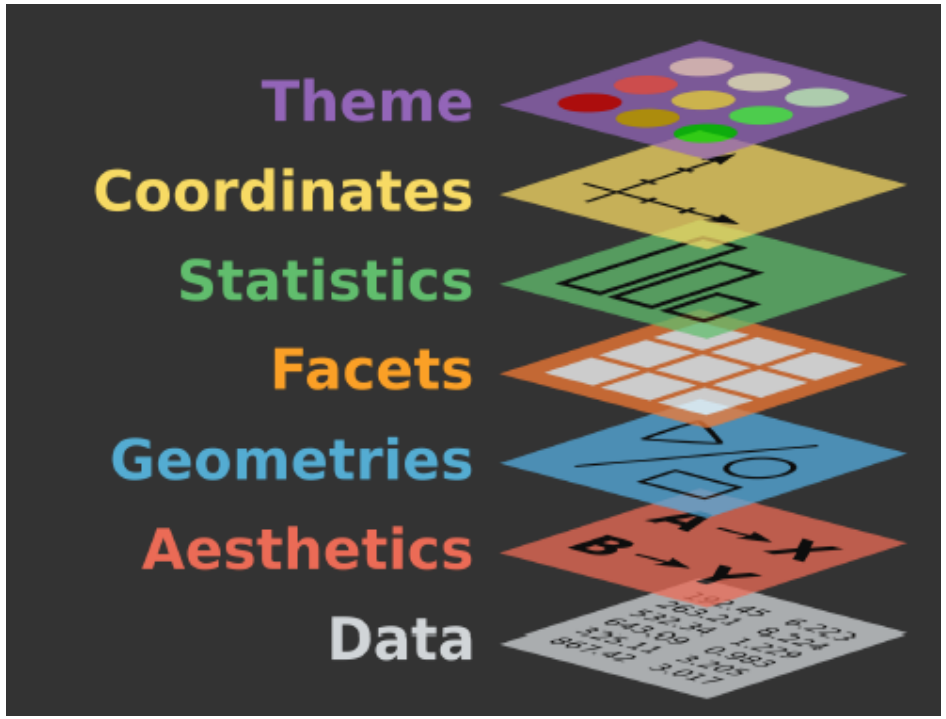
Grammar of Graphics

ggplot formula

```
ggplot(data, aesthetics ()) +
geometrics() +
facets() +
statistics() +
coordinates() +
theme() +
additional_theme()
```

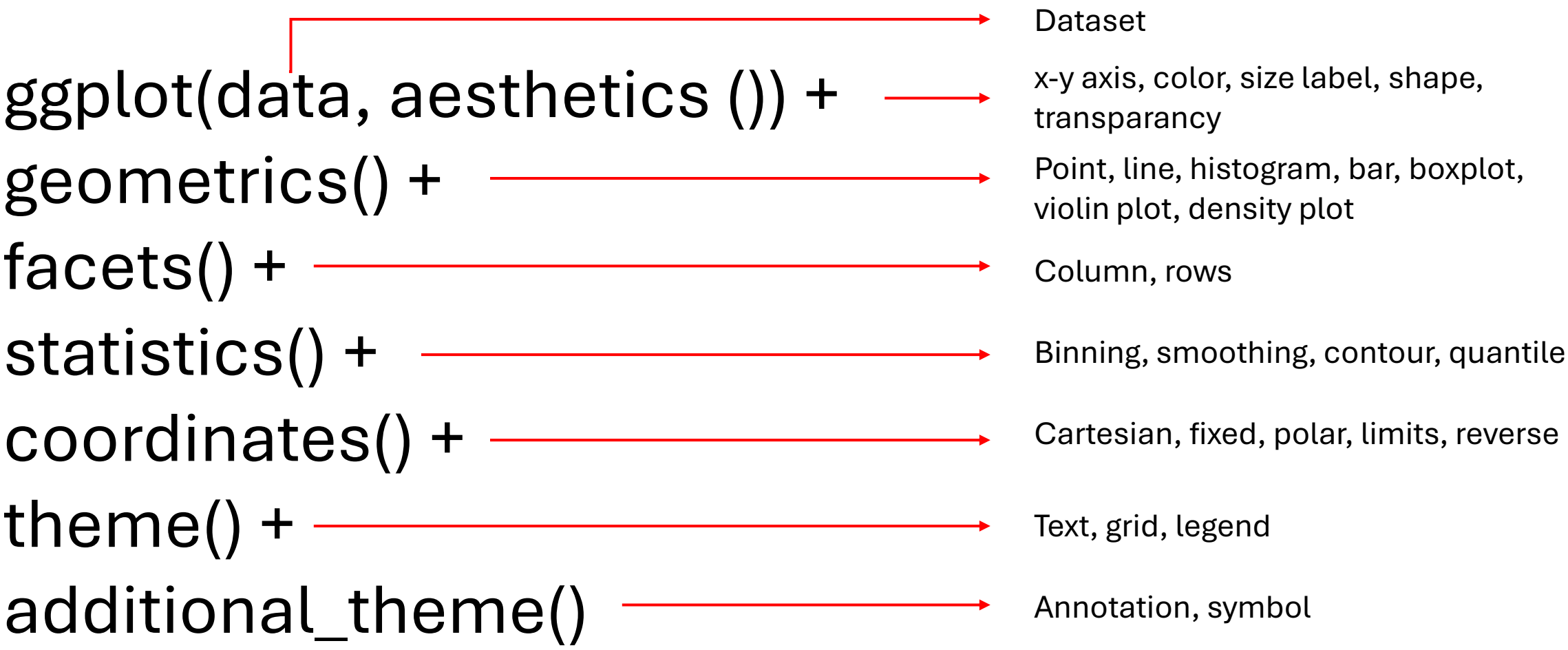
Basics

Advanced



https://link.springer.com/book/10.1007/0-387-28695-0

Grammar of Graphics Formula



Data Peparation

```
install.packages("ggplot2")


install.packages("ISLR")

library(ISLR)
data(Credit)
View(Credit)
```

Run this code

The preview of Credit dataset

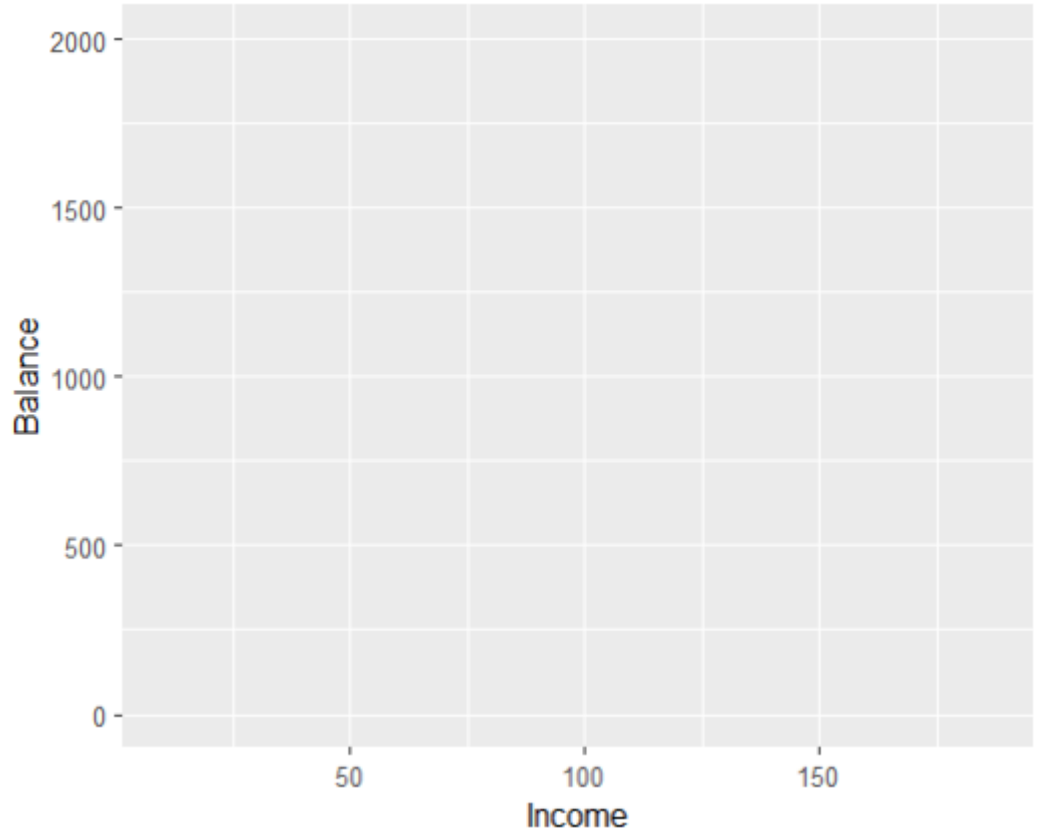
	ID	Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
1	1	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333
2	2	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903
3	3	104.593	7075	514	4	71	11	Male	No	No	Asian	580
4	4	148.924	9504	681	3	36	11	Female	No	No	Asian	964
5	5	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331
6	6	80.180	8047	569	4	77	10	Male	No	No	Caucasian	1151
7	7	20.996	3388	259	2	37	12	Female	No	No	African American	203
8	8	71.408	7114	512	2	87	9	Male	No	No	Asian	872
9	9	15.125	3300	266	5	66	13	Female	No	No	Caucasian	279

Layer 1 : Data

```
library(ggplot2)
library(ISLR)
data(Credit)

ggplot(data = Credit)
```

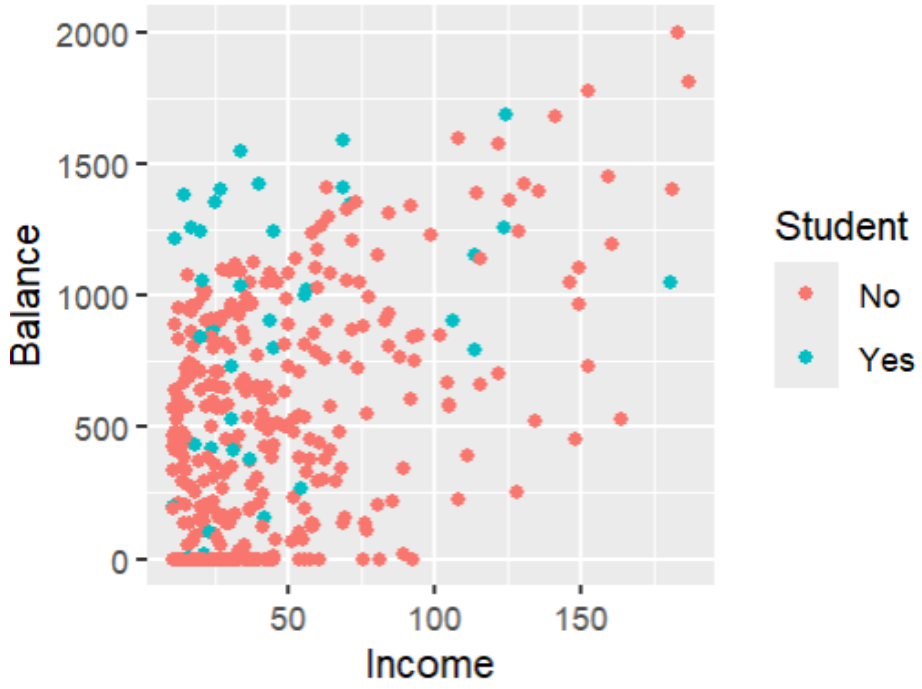
Layer 2 : Aesthetic

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student))
```



Layer 3 : Geometric

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point()      #Geometric layer
```

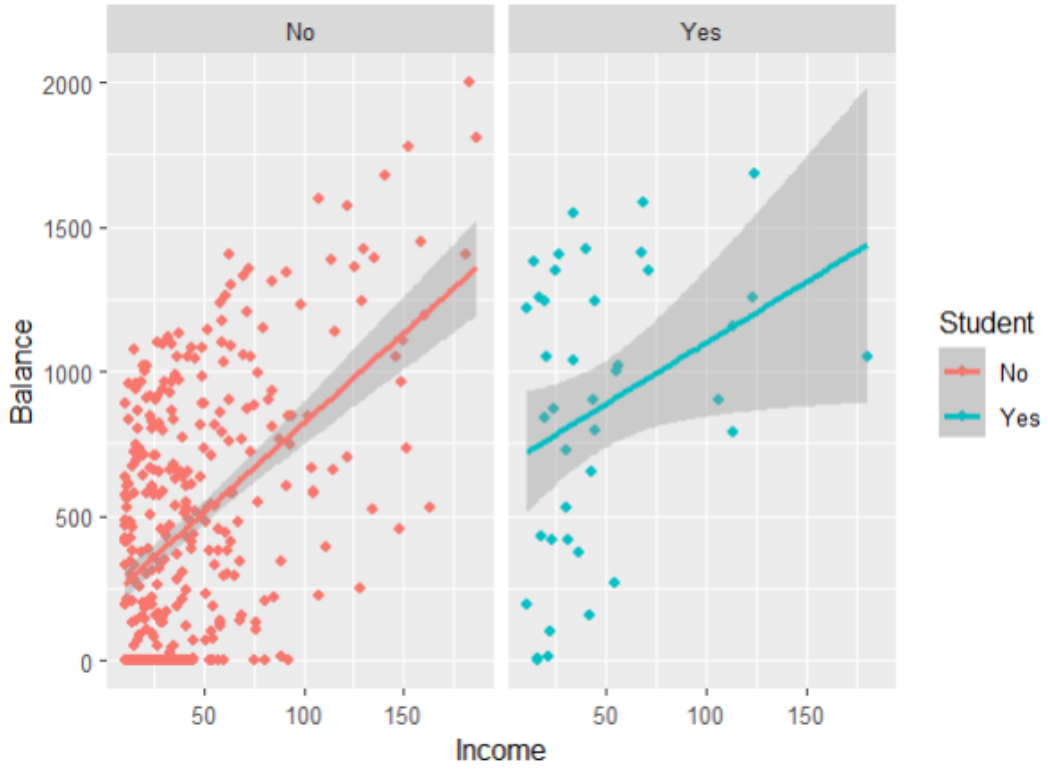


Layer 4 : Facet

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)     #facet layer
```

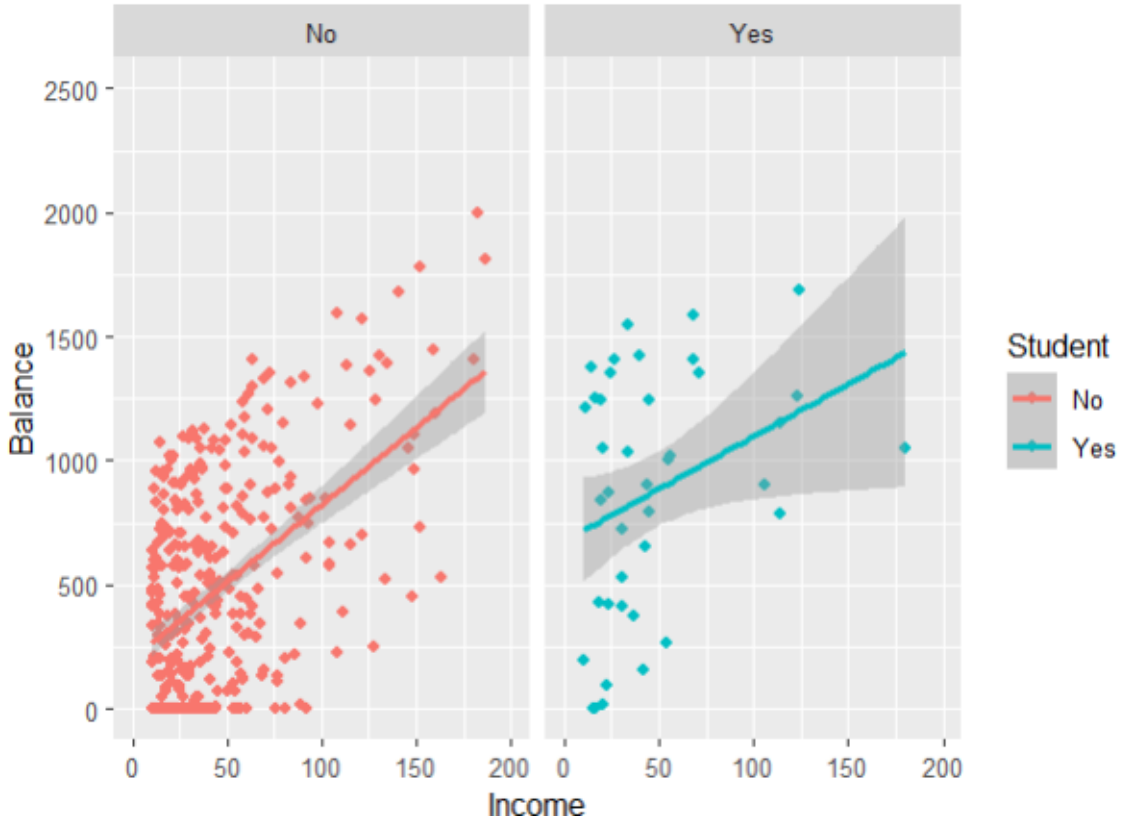
Layer 5 : Statistic

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_smooth(method = "lm", se = TRUE)       #Statistic layer
```



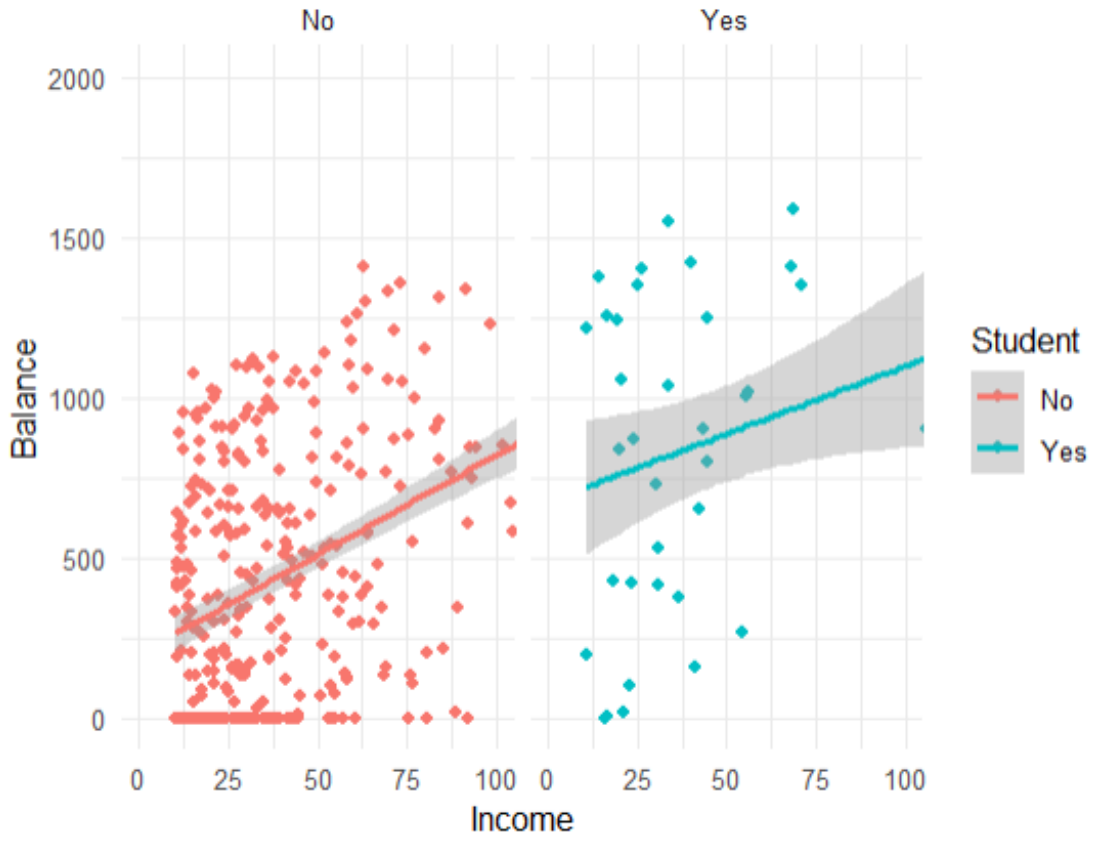
Layer 6 : Coordinate

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_smooth(method = "lm", se = TRUE) +
  coord_cartesian(xlim = c(0, 200), ylim = c(0, 2500))     #Coordinates layer
```



Layer 7 : Theme

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  facet_wrap(~ Student) +
  coord_cartesian(xlim = c(0, 100), ylim = c(0, 2000)) +
  theme_minimal()  #Theme layer
```



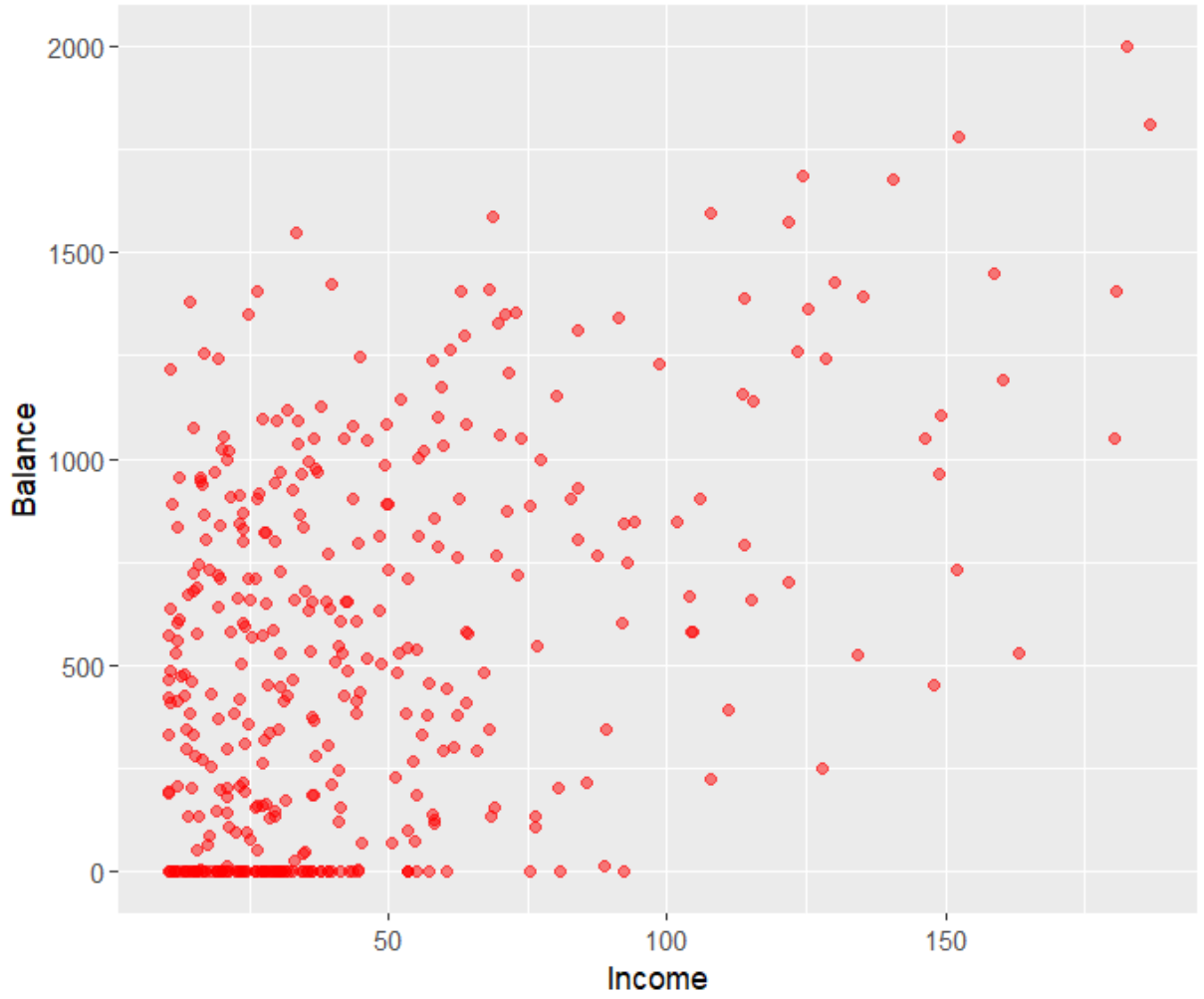
Basic tuning

ggplot Basic: Aesthetic

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income,
                   y = Balance,
                   color = Married,
                   size = Education,
                   shape = Gender,
                   alpha = Rating
                   )) +
  geom_point()
```

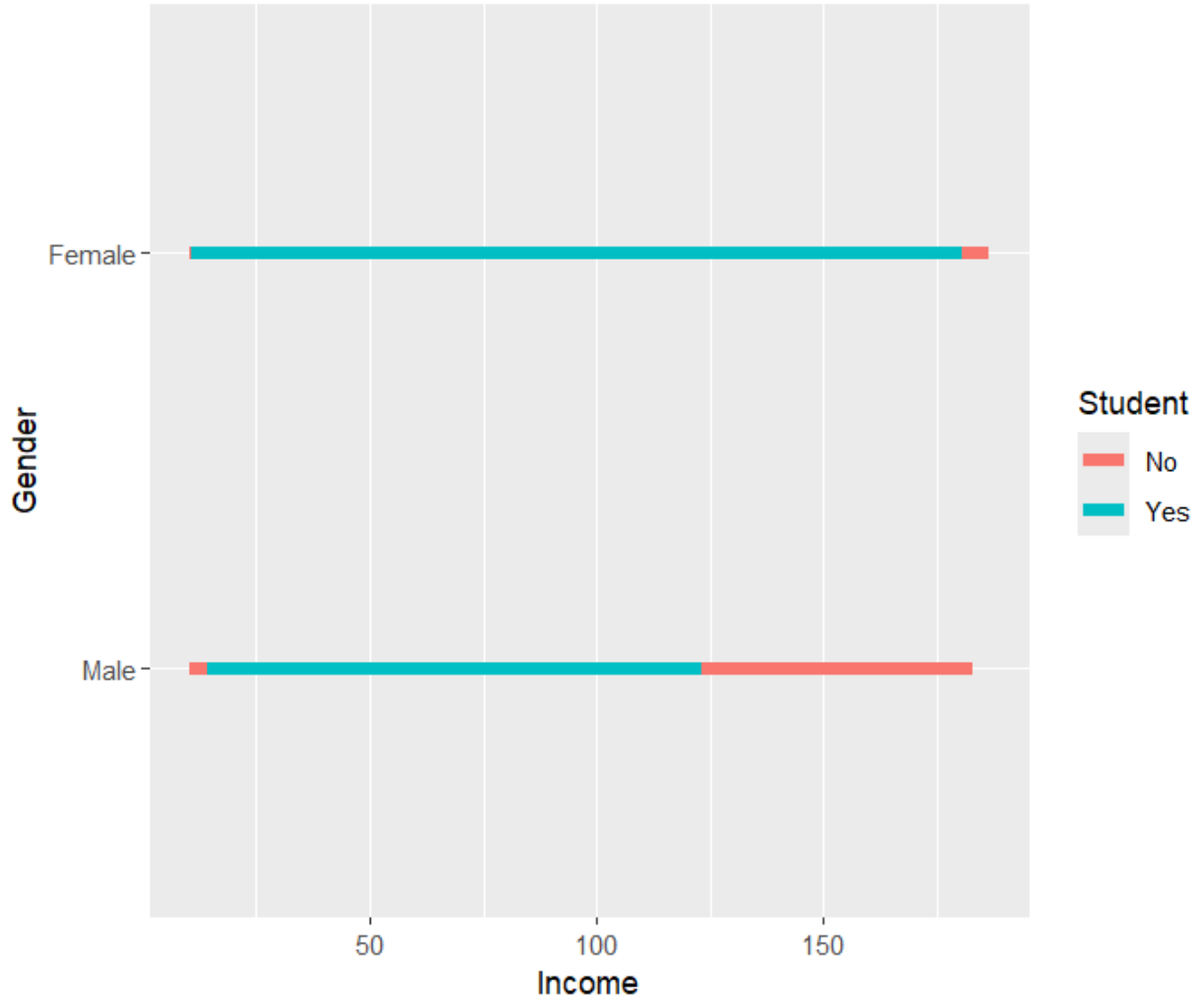
ggplot Basic: Geometric geom_point

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income,
                   y = Balance,
                   #color = Married,
                   #size = Education,
                   #shape = Gender,
                   #alpha = Rating
                  )) +
  geom_point(color = "red",
             #size = 10,
             #shape = 17,
             alpha = 0.5,
             )
```



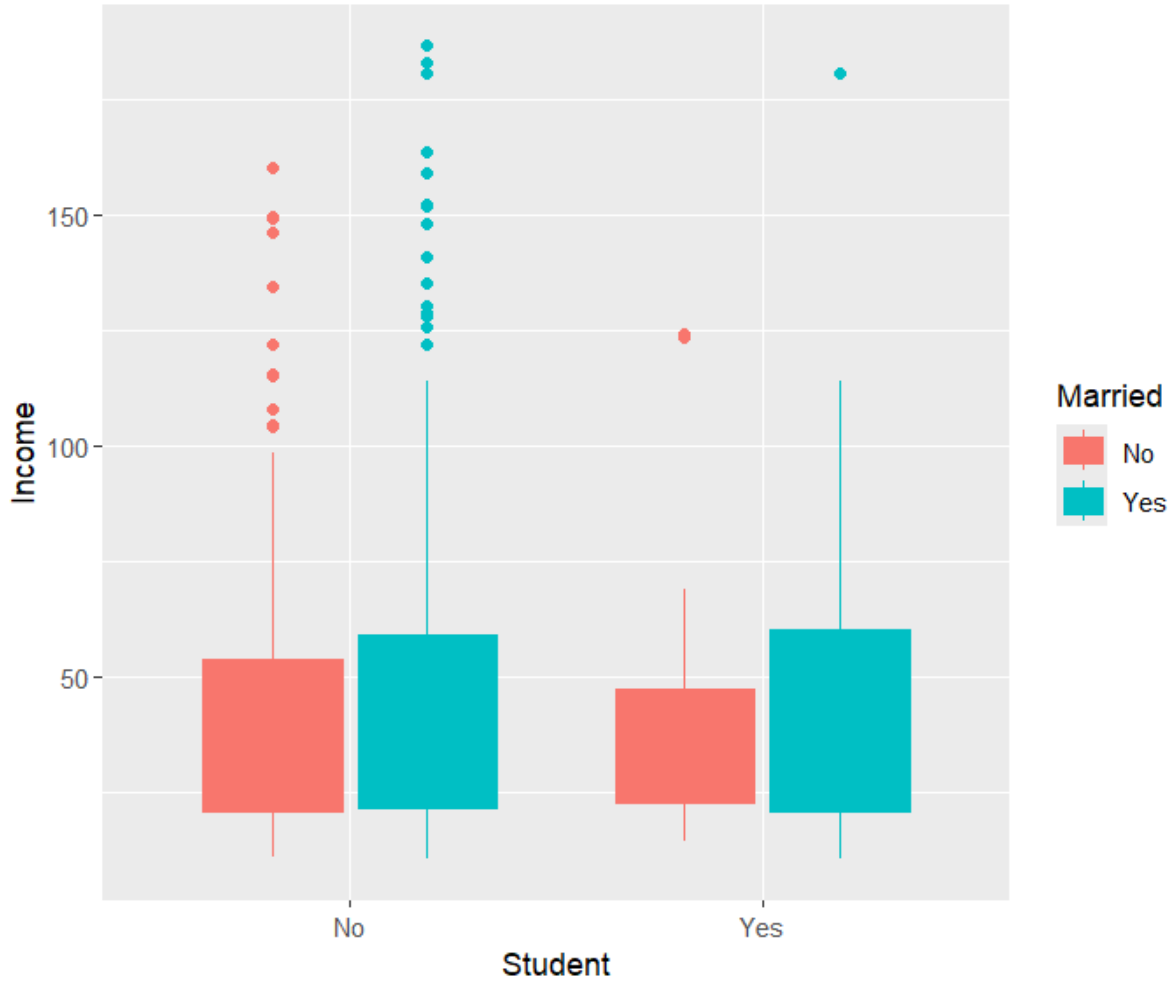
ggplot Basic: Geometric geom_line

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income,
                   y = Gender,
                   color = Student,
                   #size = Education,
                   #shape = Gender,
                   #alpha = Rating
                   )) +
    geom_line(
              #color = "red",
              size = 2,
              #shape = 5,
              #alpha = 0.5,
              )
```



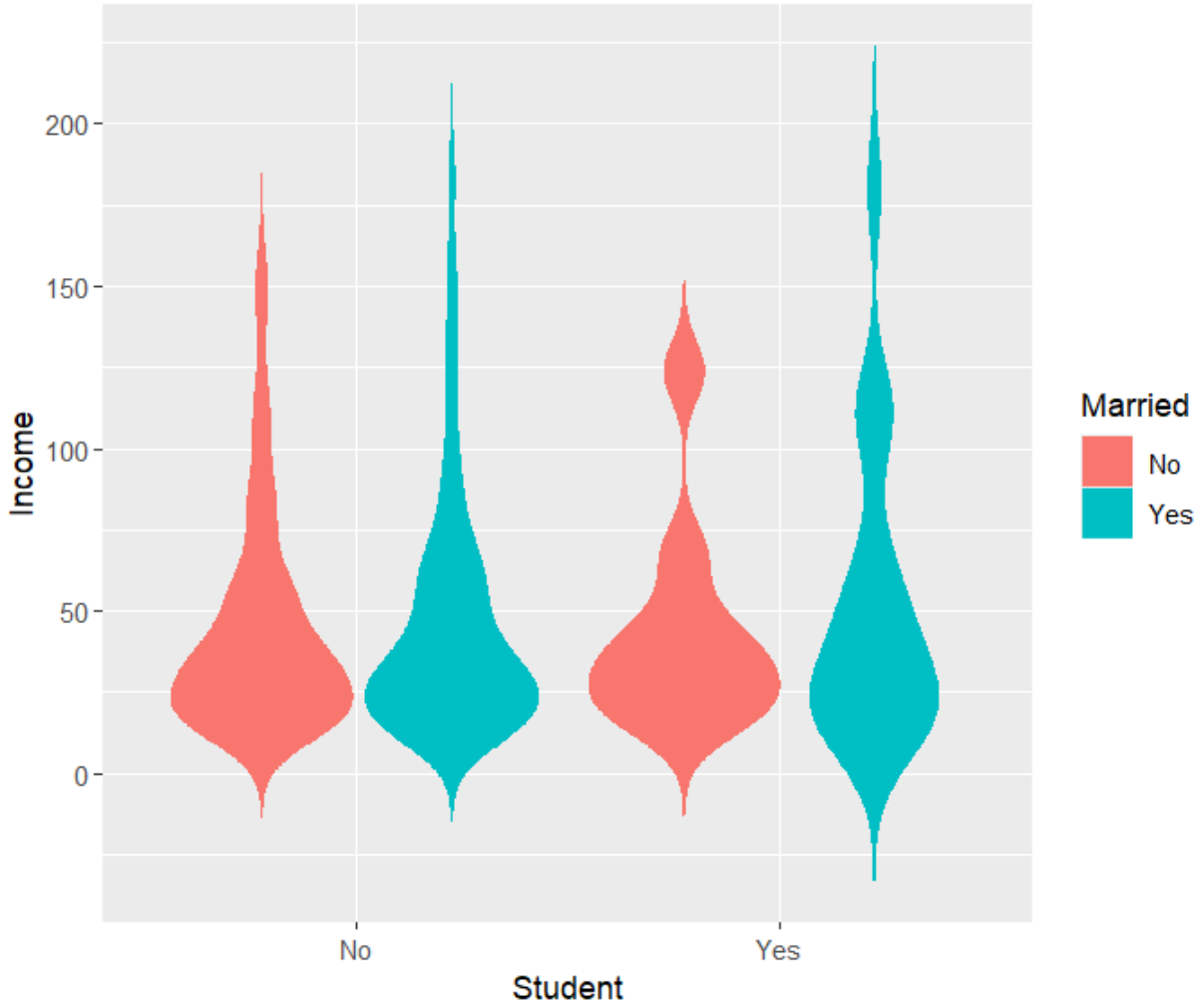
ggplot Basic: Geometric geom_boxplot

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Student,
                   y = Income,
                   color = Married,
                   fill = Married
)) +
  geom_boxplot()
```



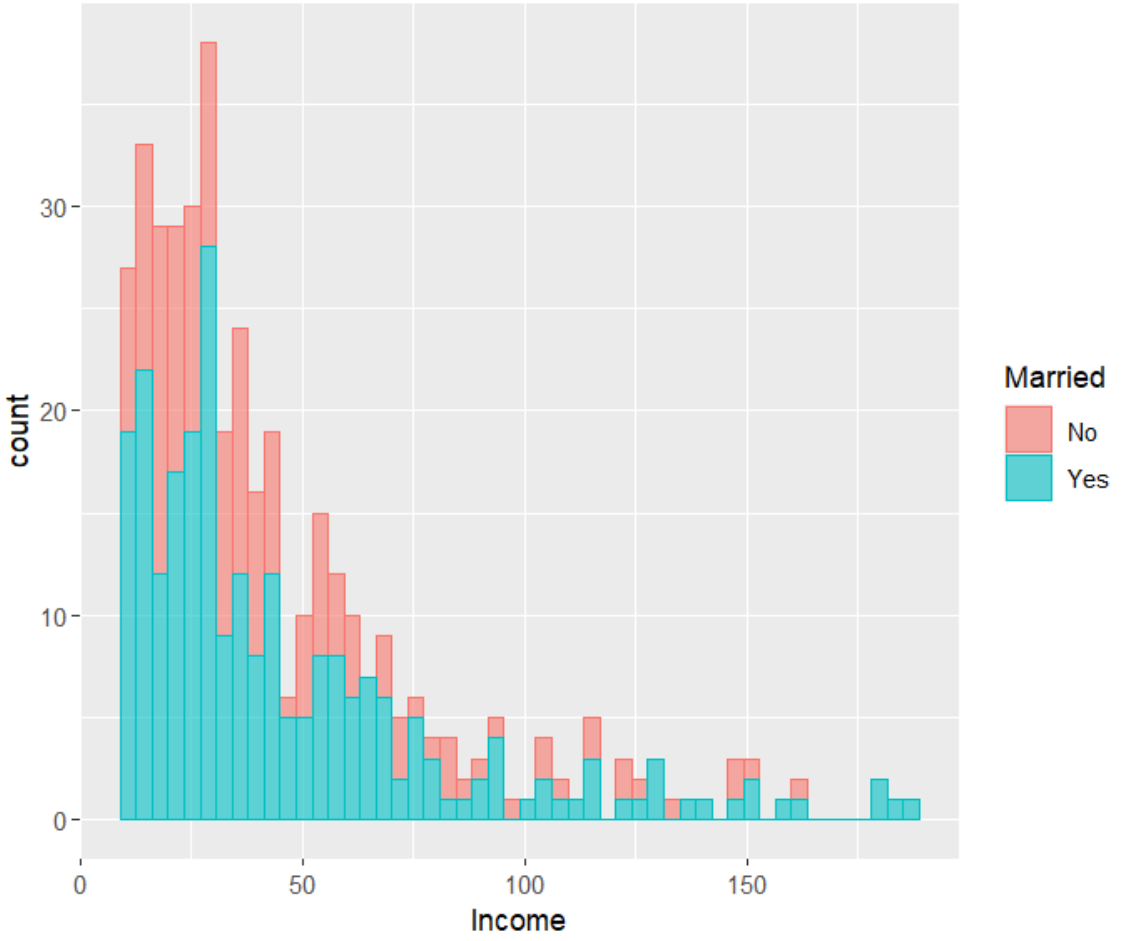
ggplot Basic: Geometric geom_violin

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Student,
                   y = Income,
                   color = Married,
                   fill = Married
)) +
  geom_violin(trim = FALSE)
```



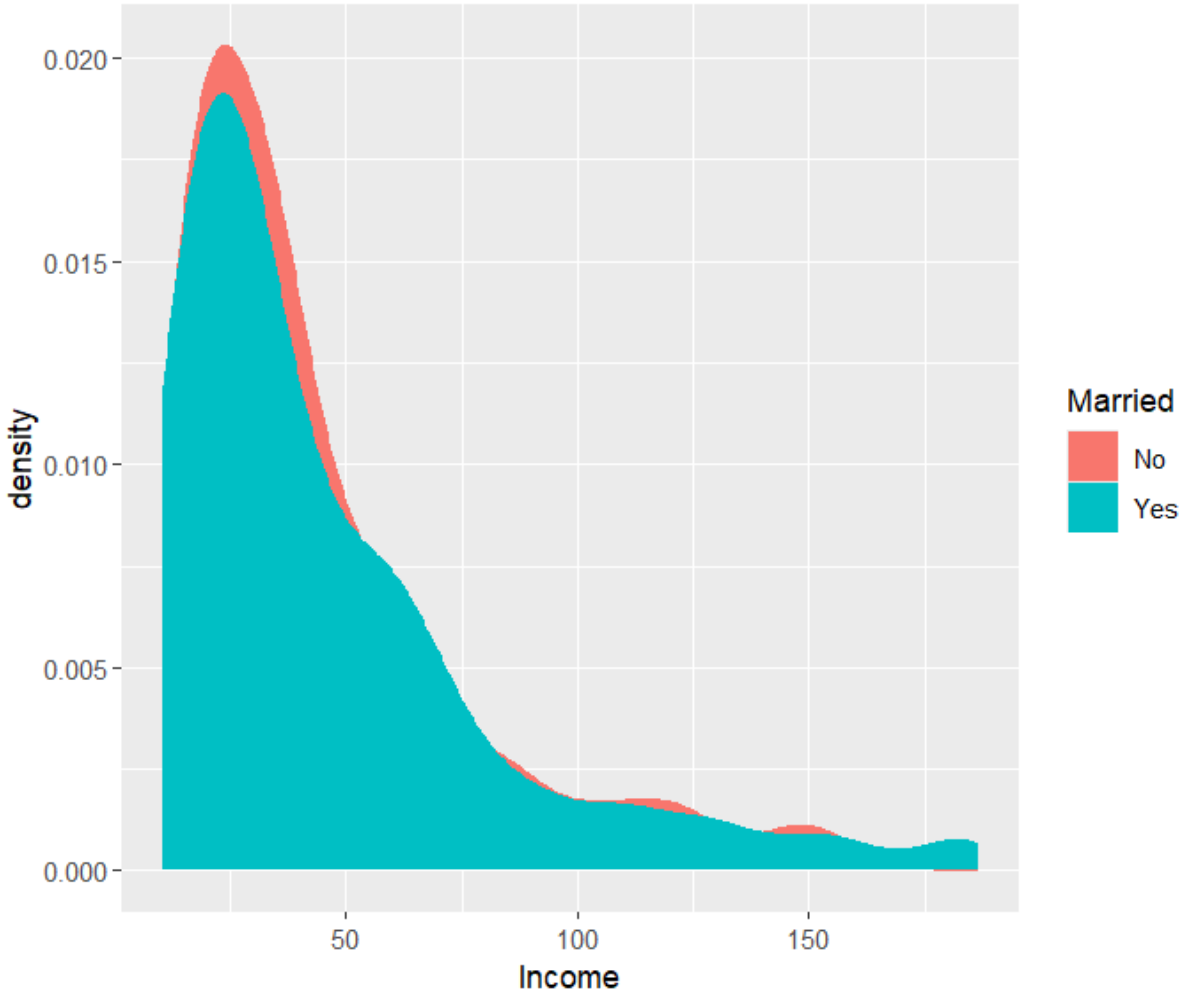
ggplot Basic: Geometric geom_histogram

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income,
                   color = Married,
                   fill = Married
                   )) +
  geom_histogram(bins = 50, alpha = 0.6)
```



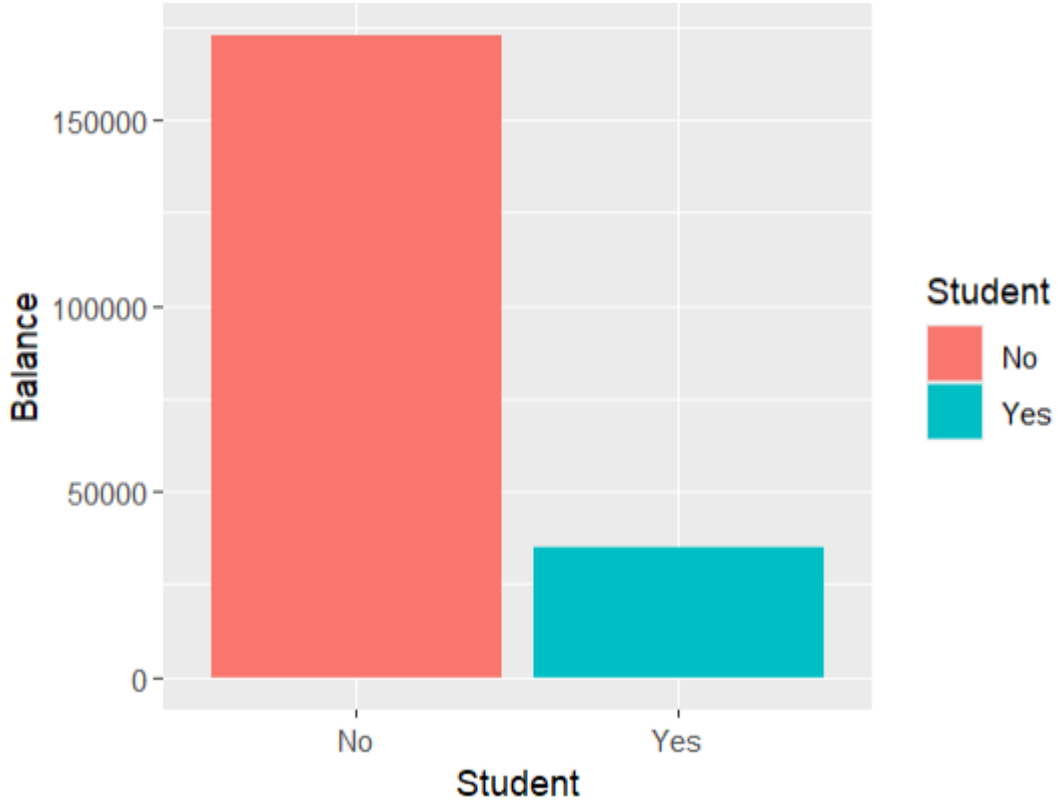
ggplot Basic: Geometric geom_density

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income,
                   color = Married,
                   fill = Married
)) +
  geom_density()
```



ggplot Basic: Geometric geom_bar

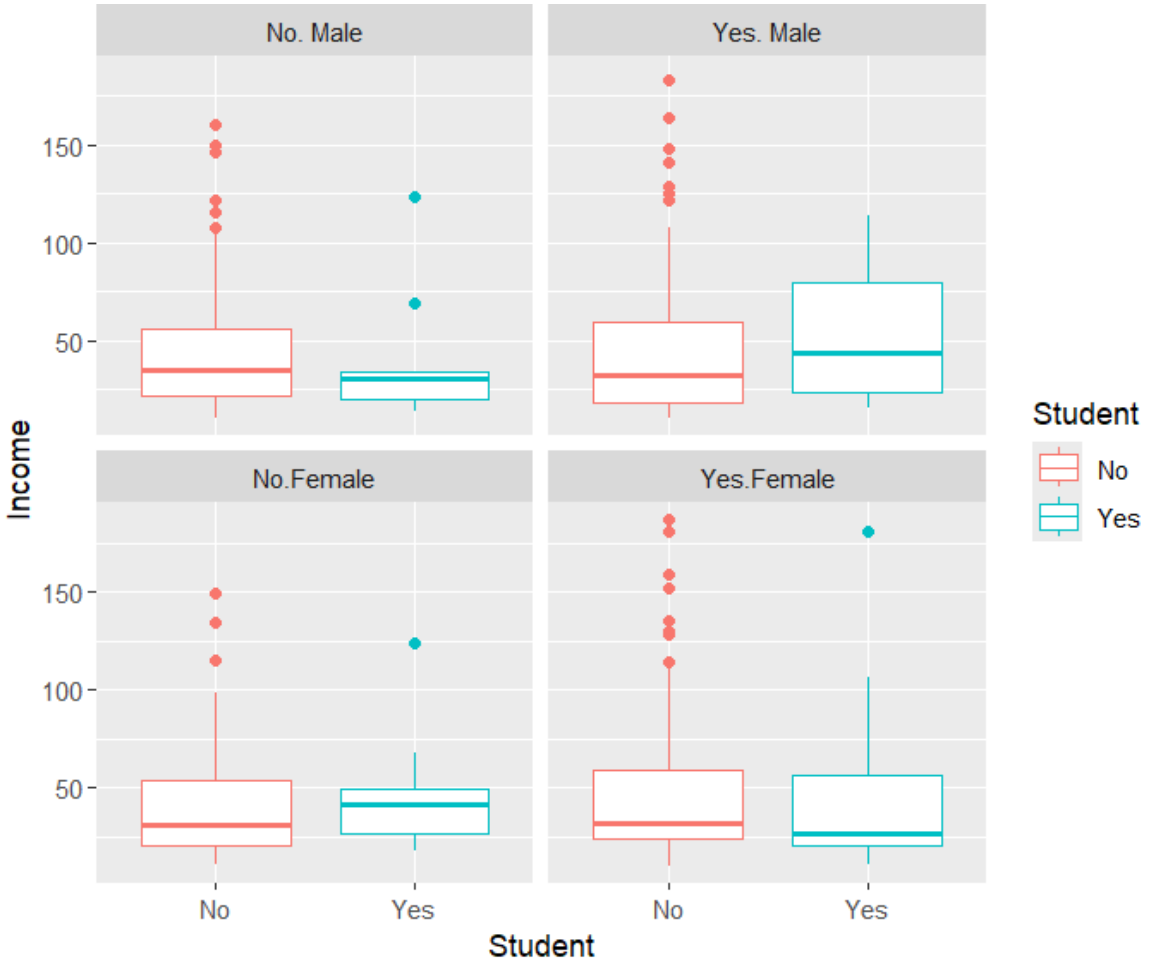
```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Student, y = Balance, fill = Student)) +
  geom_bar(stat = "identity")
```



Advance Tuning

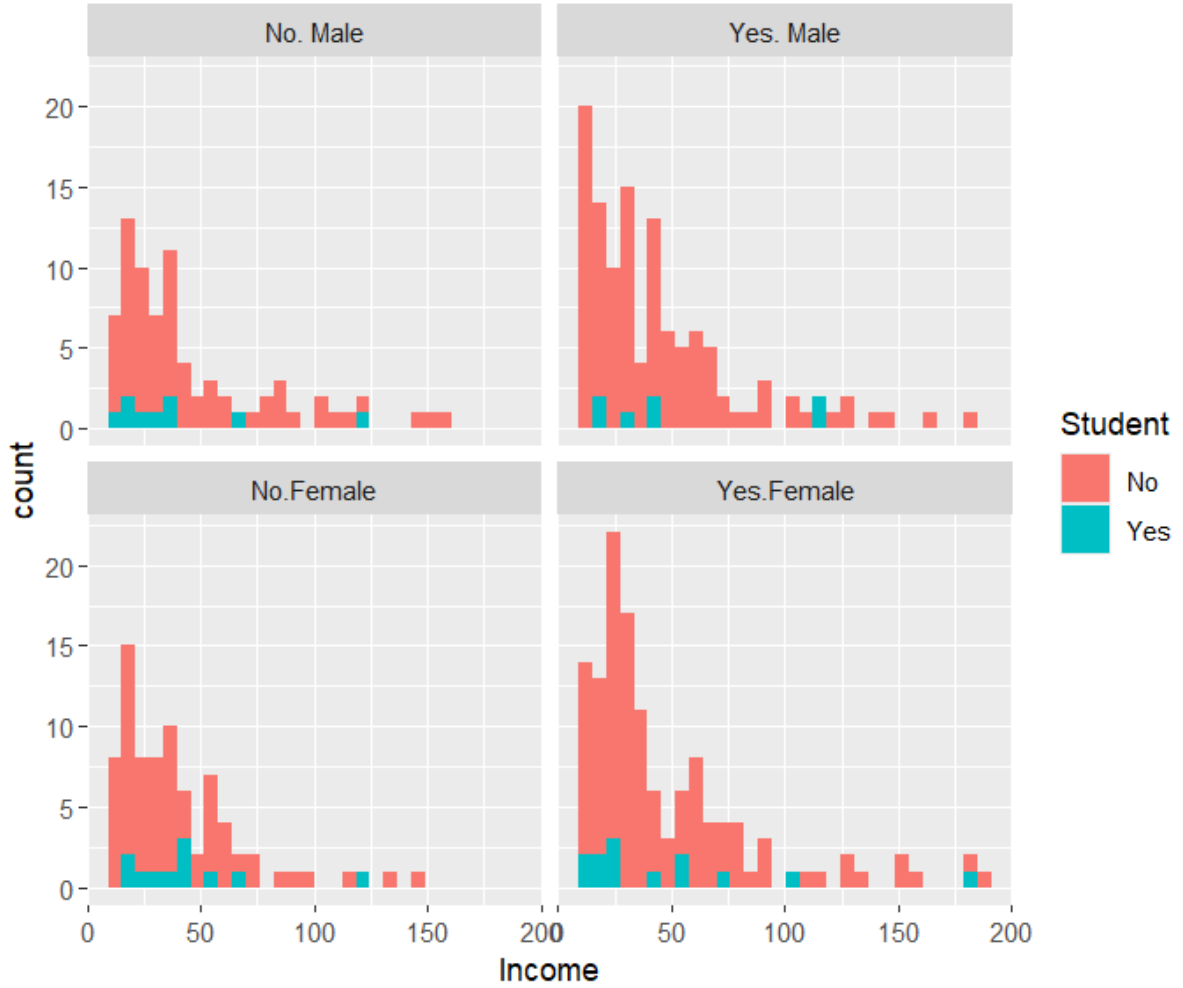
ggplot Advance: Facet facet_wrap

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Student,
                   y = Income,
                   color = Student
                   )) +
  geom_boxplot() +
  facet_wrap(~interaction(Married,Gender)
             )
```



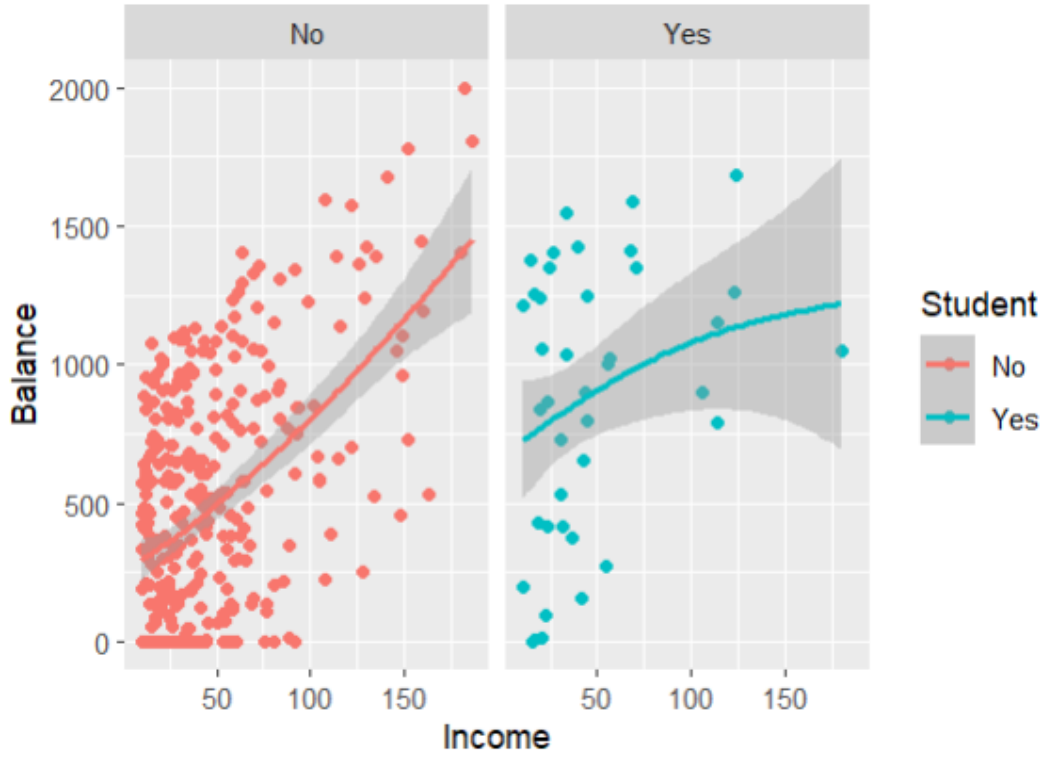
ggplot Advance: Facet facet_histogram

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income,
                   fill = Student,
                   )) +
  geom_histogram(bins = 30, Alpha = 0.7) +
  facet_wrap(~interaction(Married,Gender)
             )
```



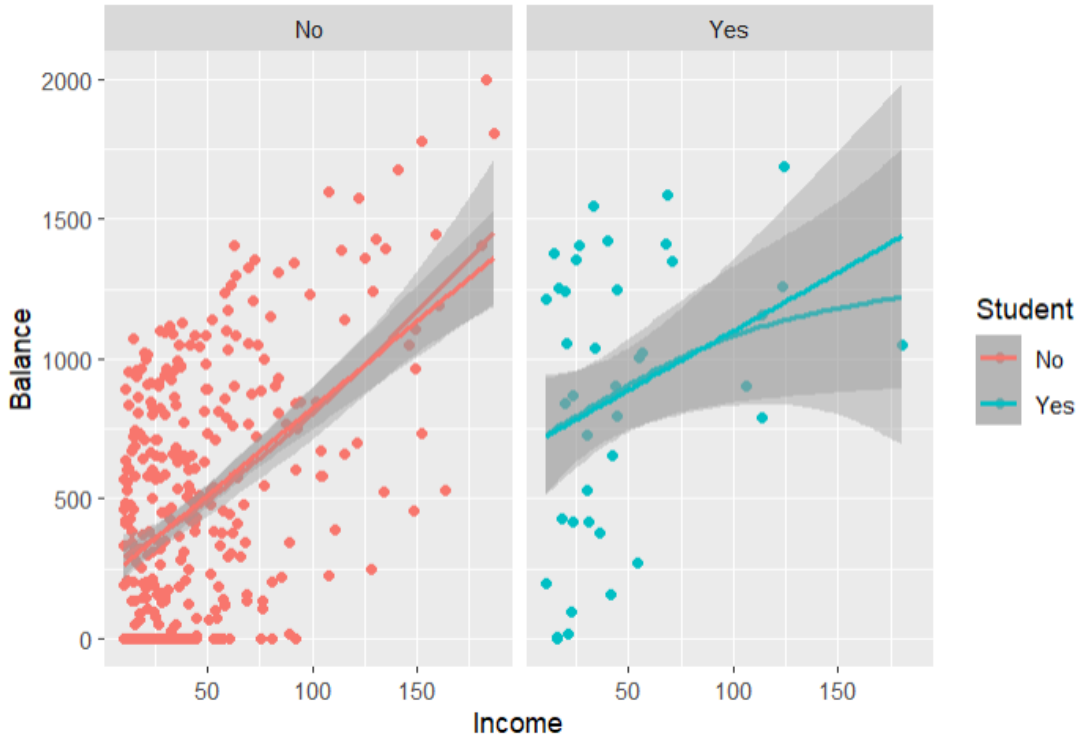
ggplot Advance: Statistic geom_smooth

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_smooth(method = "gam",     #lm, loess, gam
              se = TRUE)
```



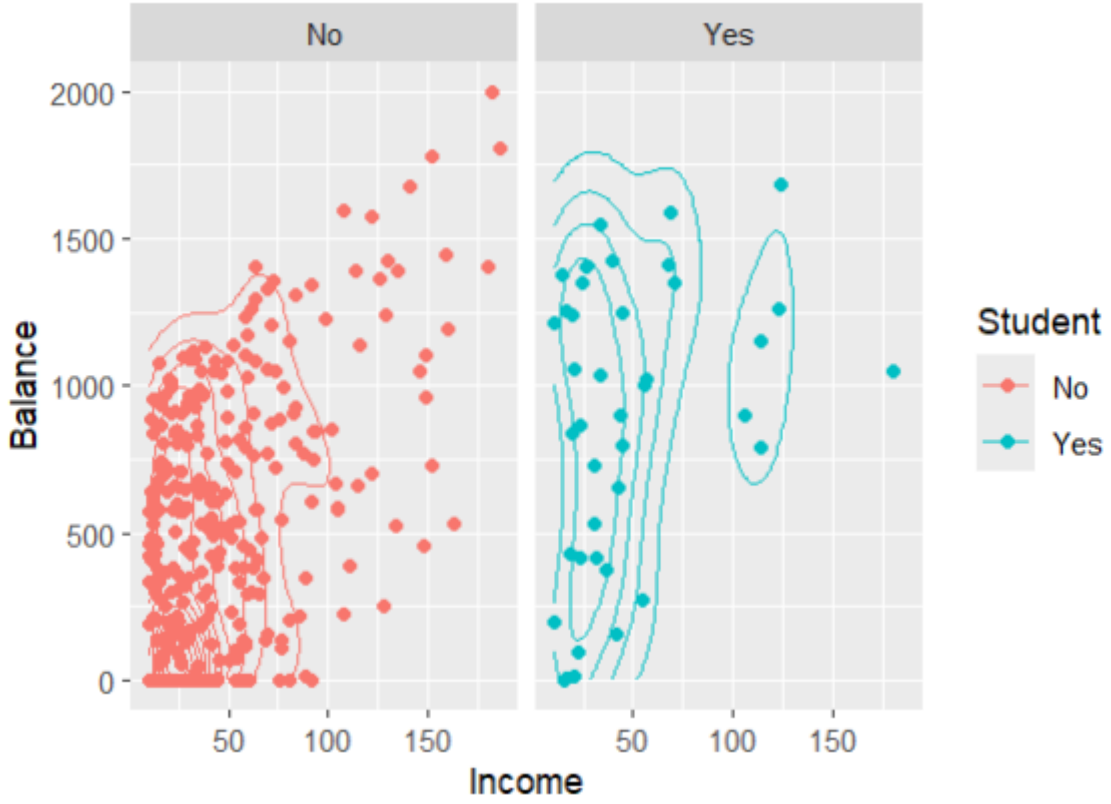
ggplot Advance: Statistic geom_smooth 2

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_smooth(method = "gam",     #lm, loess, gam
              se = TRUE)+
  geom_smooth(method = "lm")
```



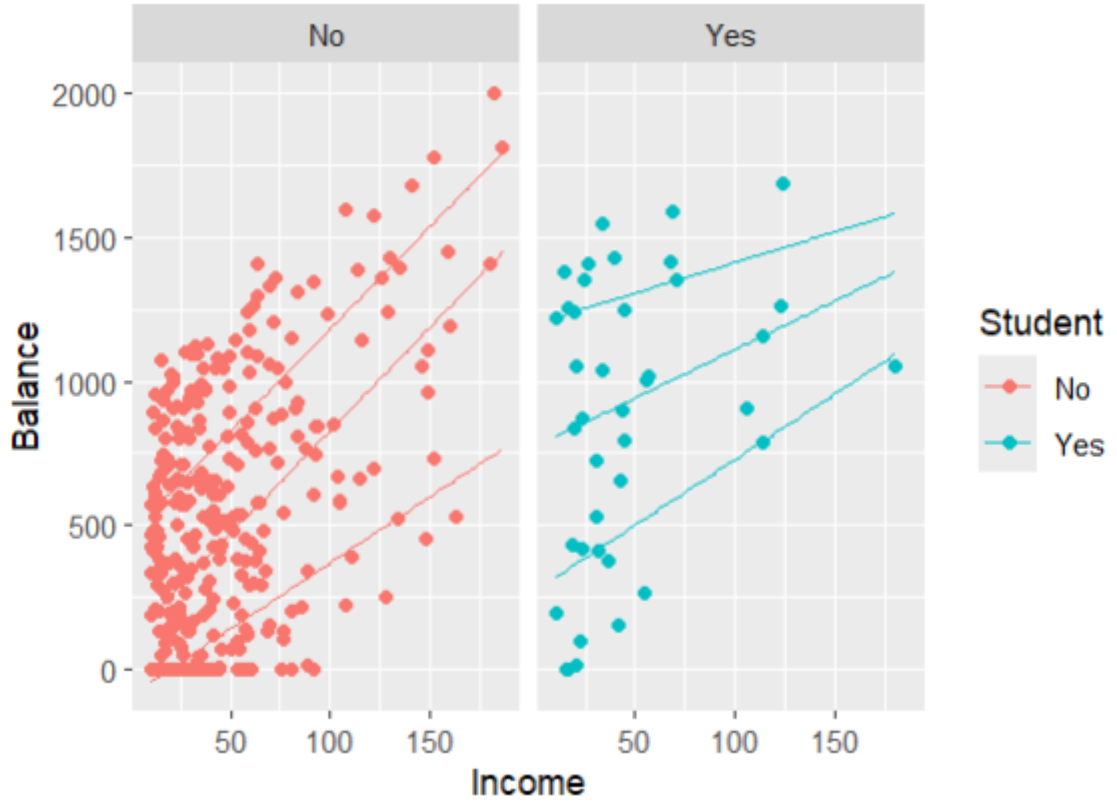
ggplot Advance: Statistic geom_smooth 2

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_density_2d()
```



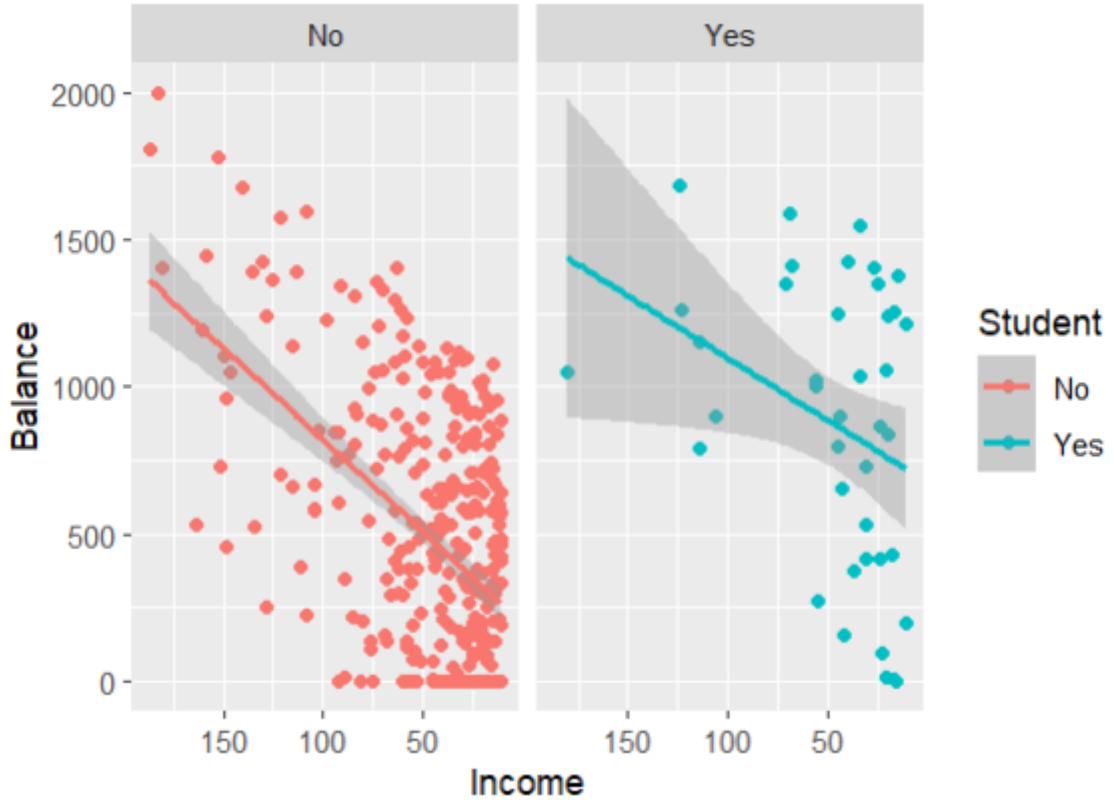
ggplot Advance: Statistic geom_quantile

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_quantile(quantiles = c(0.25, 0.5, 0.75))
```



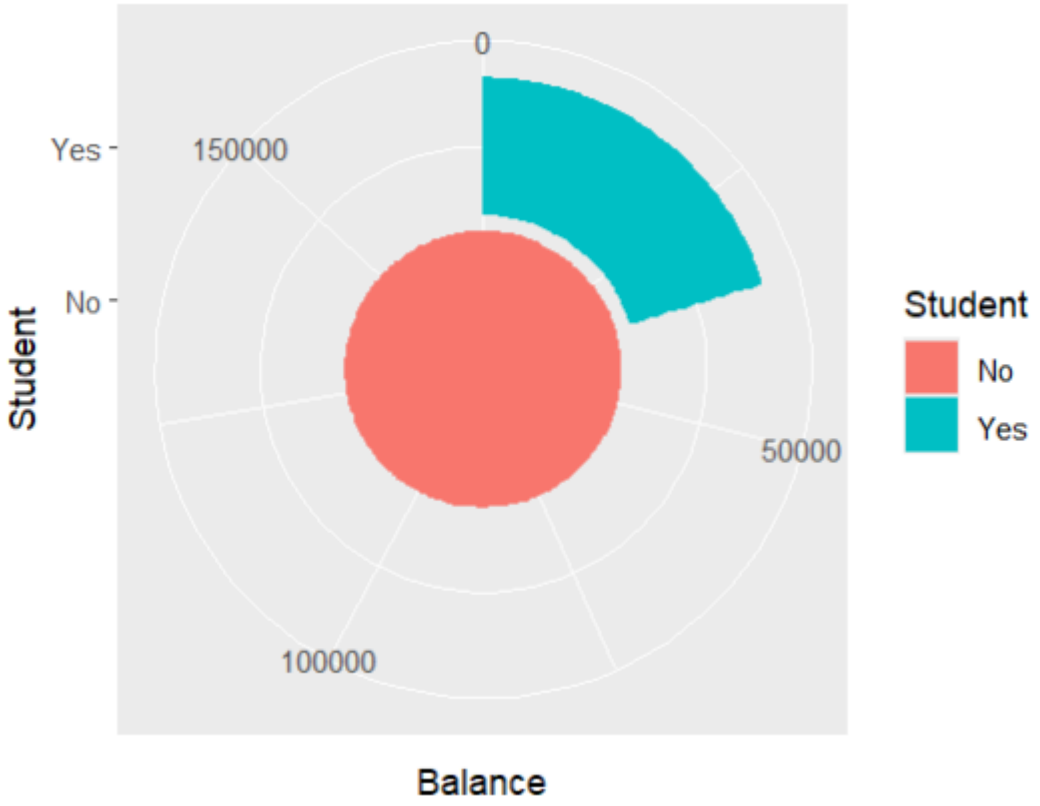
ggplot Advance: Coordinate reverse

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  facet_wrap(~ Student)+
  geom_smooth(method = "lm", se = TRUE) +
  scale_x_reverse()
```



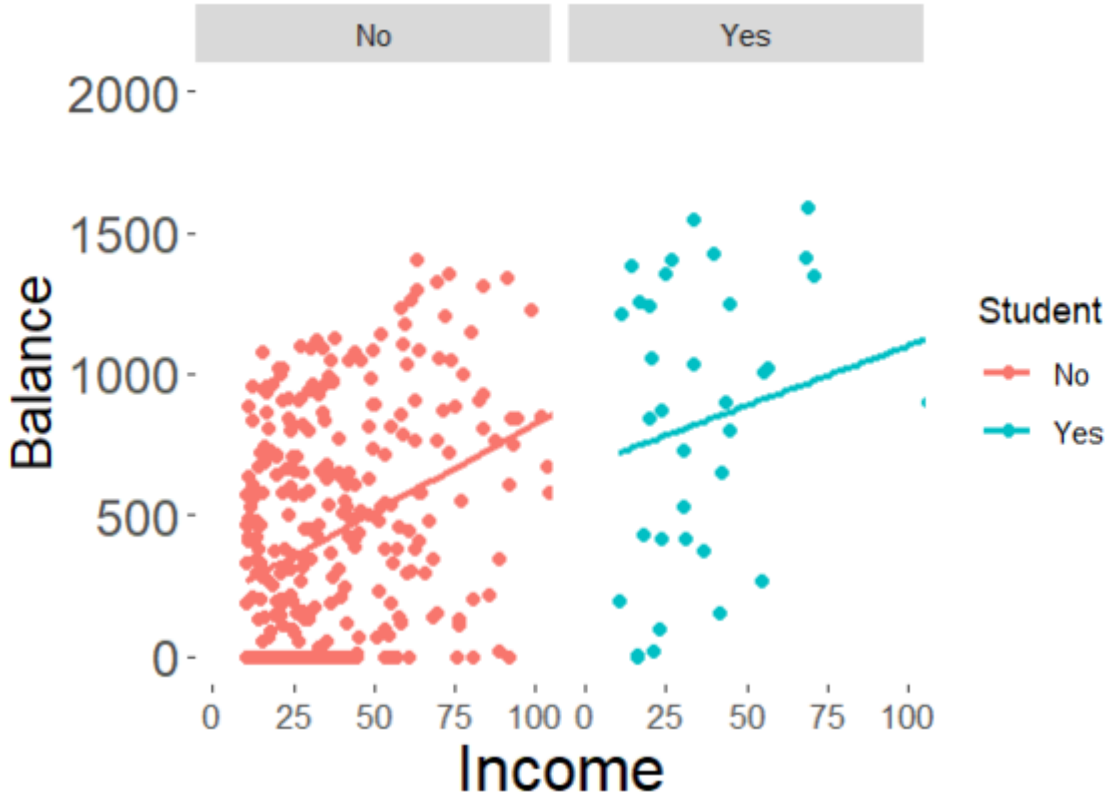
ggplot Advance: Coordinate coord_polar

```
library(ggplot2)
library(ISLR)
data(Credit)
names(Credit)

ggplot(Credit, aes(x = Student, y = Balance, fill = Student)) +
  geom_bar(stat = "identity")+
  coord_polar(theta = "y")
```



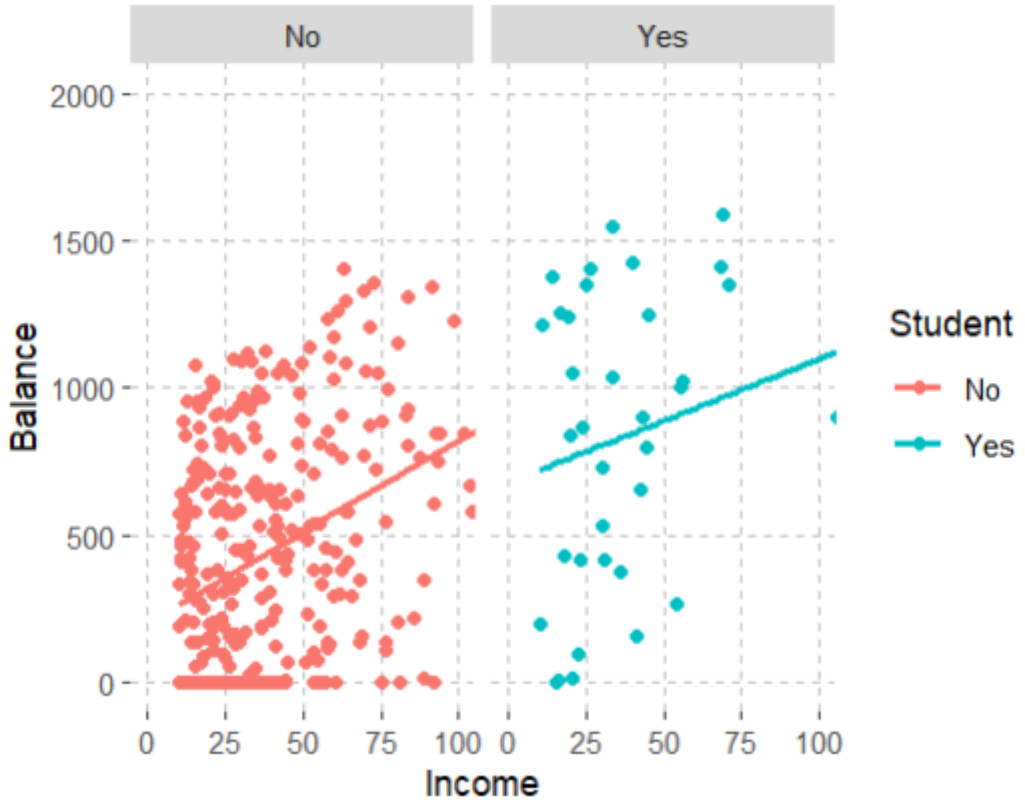
ggplot Advance: Theme Axis

```r
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, linetype = "solid") +
  facet_wrap(~ Student) +
  coord_cartesian(xlim = c(0, 100), ylim = c(0, 2000)) +
  theme(
    panel.background = element_rect(fill = "white"),  # Background color
    # Axis titles
    axis.title.x = element_text(size = 19),  # x-axis title font size
    axis.title.y = element_text(size = 16),  # y-axis title font size
    # Axis text
    axis.text.x = element_text(size = 10),  # x-axis text font size
    axis.text.y = element_text(size = 15)  # y-axis text font size
  )
```



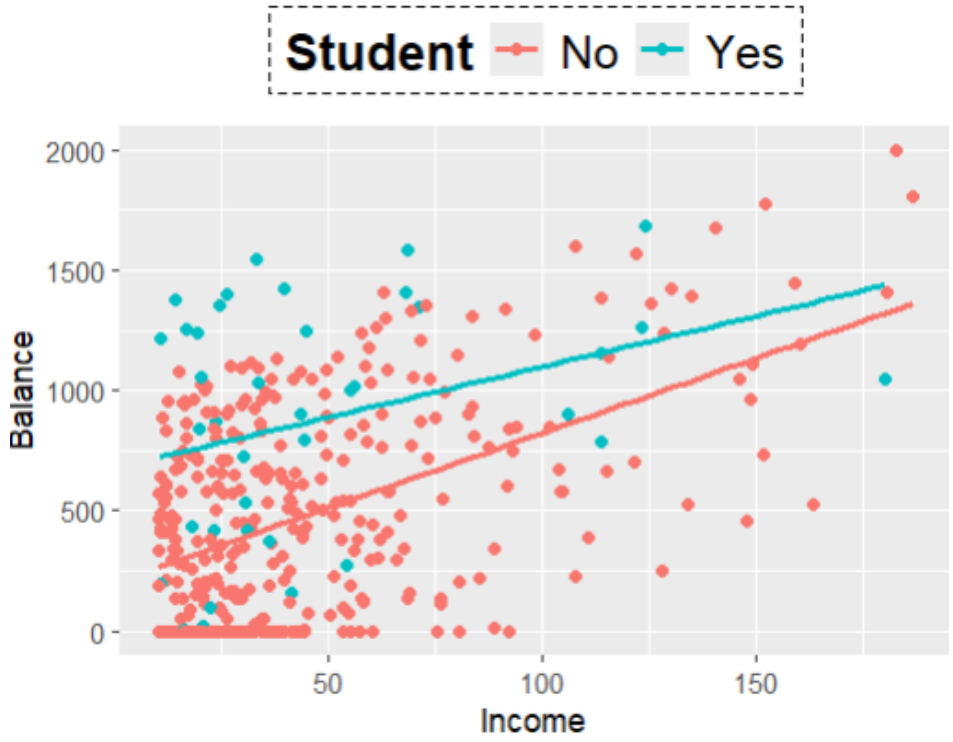
ggplot Advance: Theme Grid

```r
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, linetype = "solid") +
  facet_wrap(~ Student) +
  coord_cartesian(xlim = c(0, 100), ylim = c(0, 2000)) +
  theme(
    panel.background = element_rect(fill = "white"),  # Background color
    # Grid lines
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    panel.grid.minor = element_line(),  # Minor gridlines

  )
```



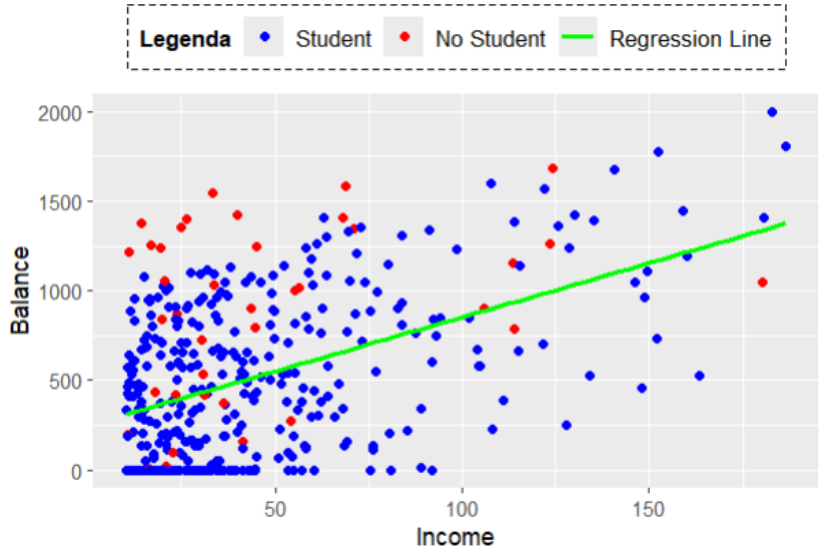
ggplot Advance: Theme Legend 1

```r
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, linetype = "solid") +
  theme(
    legend.position = ("top"),  # inside: c(0.2, 0.8), outside: ("top")(bottom")
    legend.text = element_text(size = 15),
    legend.title = element_text(size = 16, face = "bold"),  #"bold","italic",
    legend.background = element_rect(fill = "white", color = "black",linetype = "dashed"),
    #legend.background = element_blank() #removes the background (box)
  )
```



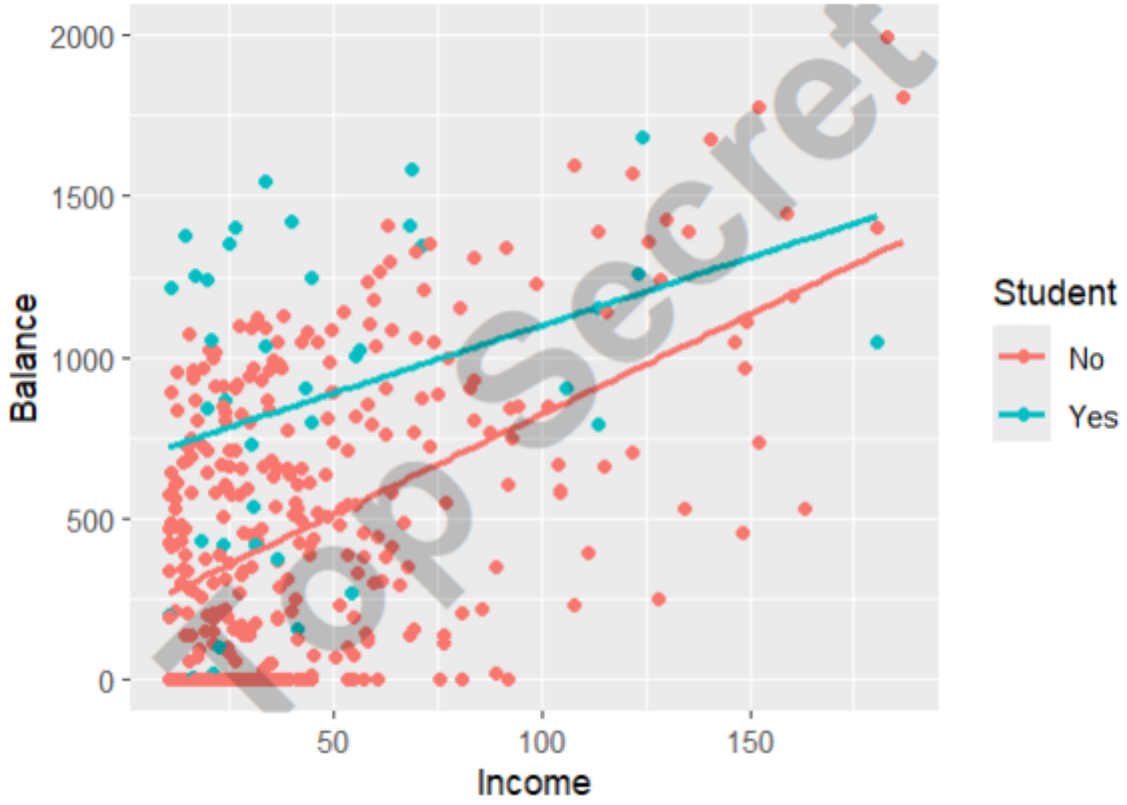
ggplot Advance: Theme Legend 2

```r
ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, linetype = "solid", aes(color = "Regression Line")) +
  theme(
    legend.position = "top",  # Position of the legend
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 10, face = "bold"),
    legend.background = element_rect(fill = "white", color = "black", linetype = "dashed")
  ) +
  labs(
    color = "Legenda"  # Set the title of the legend
  ) +
  scale_color_manual(values = c("blue", "red", "green"),  # Add one more color for regression line
                     labels = c("Student", "No Student", "Regression Line")
  )
```



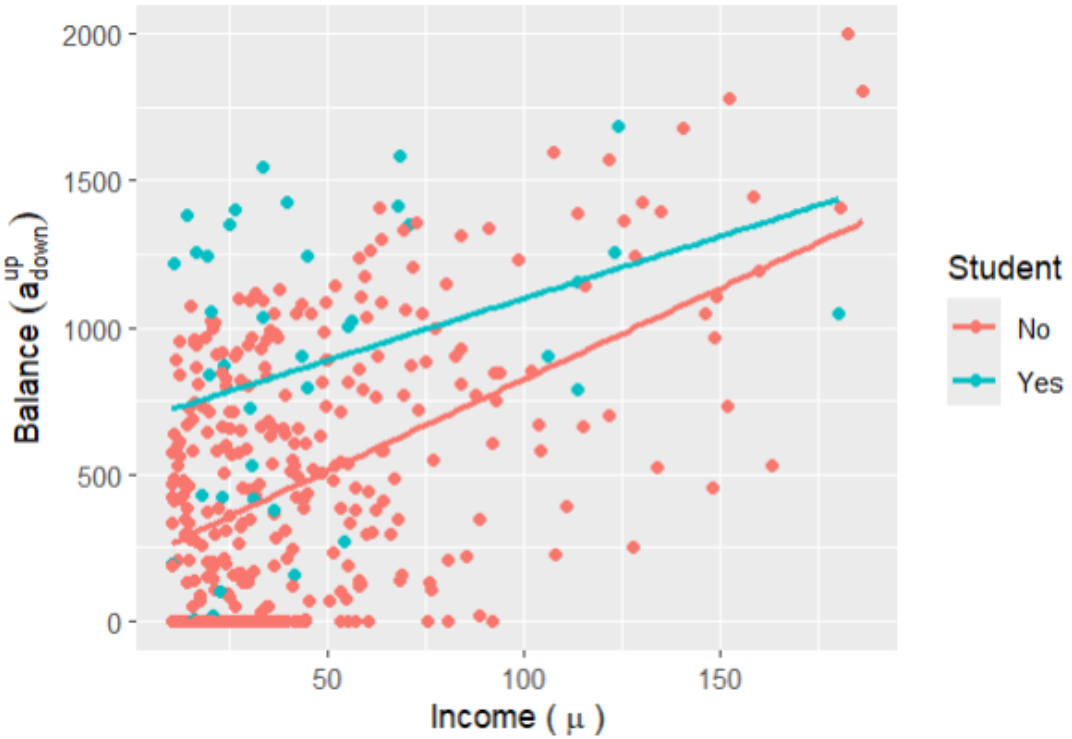
ggplot Advance: Theme Annotate

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, linetype = "solid") +
  theme() +
  annotate("text",
           x = 100, y = 1000,  # Position of the text
           label = "Top Secret",  # The text you want to add
           size = 20, color = "black",  # Text size and color
           fontface = "bold", # Text style
           angle = 45,
           alpha = 0.2
           )
```



ggplot Advance: Theme Symbol

```
library(ggplot2)
library(ISLR)
data(Credit)
head(Credit)

ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, linetype = "solid") +
  theme() +
  labs(
    x = expression("Income ("~mu~")"),  # ("~sigma^2~"),("~mu~")
    y = expression("Balance "(a[down]^up))  # subscript
  )
```



maps

```
#install.packages("maps")
library(maps)

crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
crimesm <- reshape2::melt(crimes, id = 1)
library(maps)
states_map <- map_data("state")
ggplot(crimes, aes(map_id = state)) +
  geom_map(aes(fill = Murder), map = states_map) +
  expand_limits(x = states_map$long, y = states_map$lat) +
  coord_map()
```