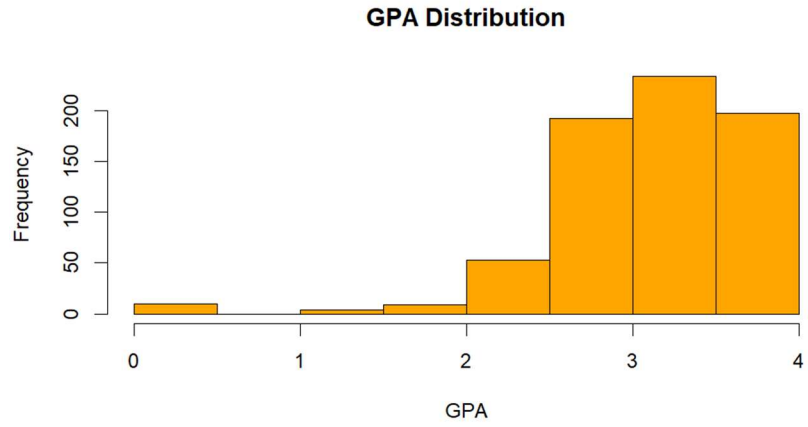
- Installing Packages: Use install.packages() to install packages only once for your system (or when updates are available). *install.packages("regclass")*
- Loading Libraries: Use library() every time you start a new session to load the required package(s). *library(regclass)*
    - **dplyr** - Data manipulation and transformation.
    - **ggplot2** - Data visualization using the grammar of graphics.
    - **tidyr** - Data tidying and reshaping.
    - **plotly** - Interactive and dynamic data visualization.
    - **shiny** - Building interactive web applications and dashboards.
    - **readr** - Efficient importing and exporting of data.
    - **flextable** - Creating professional and customizable tables.
- Understanding Data Management in R Studio
    - Read Data: Use functions like read.csv() or read_excel() to import data.
        - *data <- read.csv("file.csv")*
    - Save Data: Use write.csv() or similar functions to export data.
        - *write.csv(data, "output.csv")*
    - Explore Data: Use these commands to understand your dataset:
        - *head(data) # First 6 rows*
        - *str(data) # Structure of the data*
        - *summary(data) # Summary statistics*
- It is important to create a working directory because this is the location where your files are saved and read from.
    - Copy and paste the link below to learn more about working directories
    - *https://www.youtube.com/watch?v=dc8GMV3BPM0&list=PL-blpDu7mdw0dyQJgWw23xnqfm7FOUqNB&index=4*

Please enjoy this R Studio help sheet and please let me know if you have any questions!

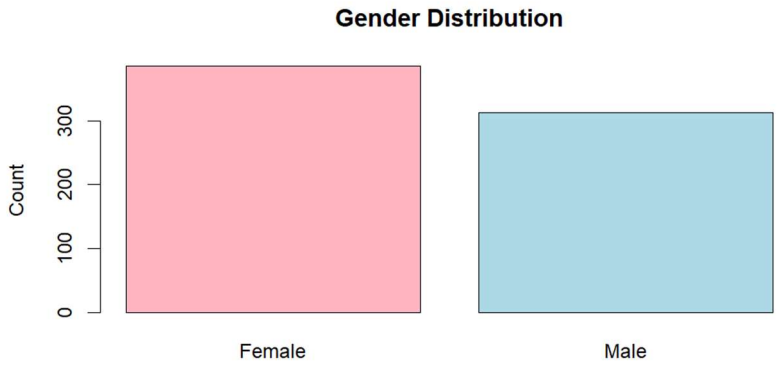
Histogram

```
hist(SURVEY10$GPA,
    main = "GPA Distribution",
    xlab = "GPA",
    ylab = "Frequency",
    col = "orange",
    border = "black")
```



Bar Chart

```
barplot(table(SURVEY10$Gender),
        main = "Gender Distribution",
        xlab = "Gender",
        ylab = "Count",
        col = c("lightpink", "lightblue"))
```

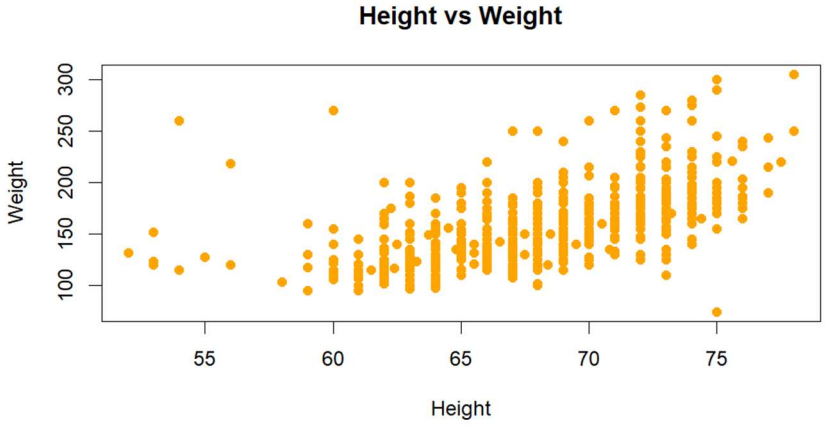


Box Plot

```
boxplot(SURVEY10$GPA,
        main = "Boxplot of GPA",
        ylab = "GPA",
        col = "orange")
```

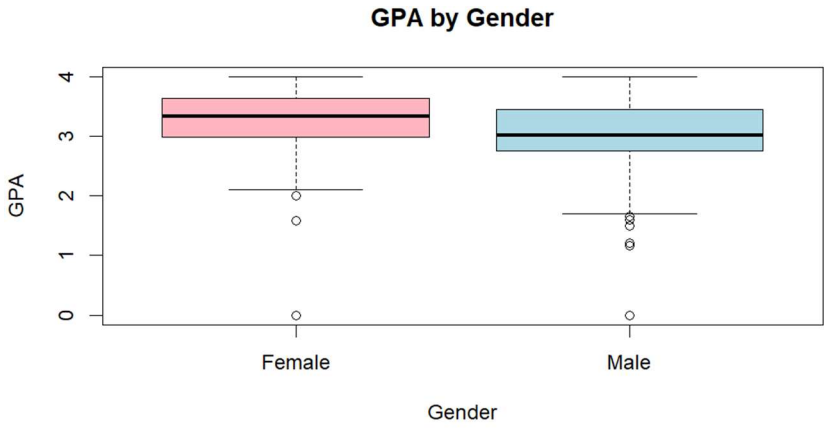
Scatter Plot

```
plot(SURVEY10$Height,
SURVEY10$Weight,
    main = "Height vs Weight",
    xlab = "Height",
    ylab = "Weight",
    col = "orange",
    pch = 19)
```



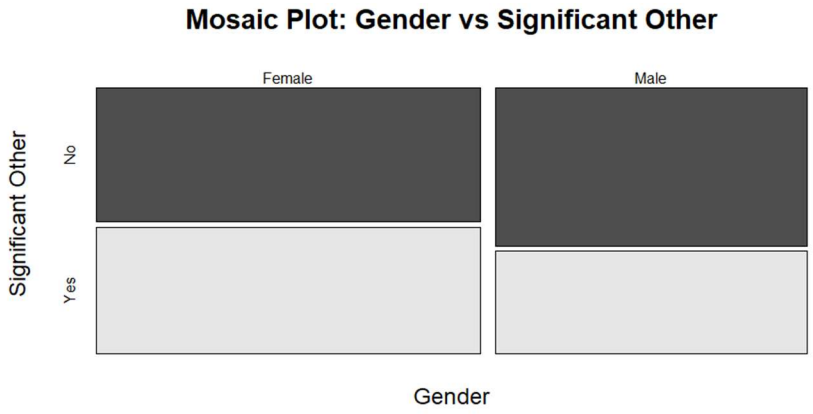
Side by Side Box Plots

```
boxplot(SURVEY10$GPA ~ SURVEY10$Gender,
        main = "GPA by Gender",
        xlab = "Gender",
        ylab = "GPA",
        col = c("lightpink", "lightblue"))
```



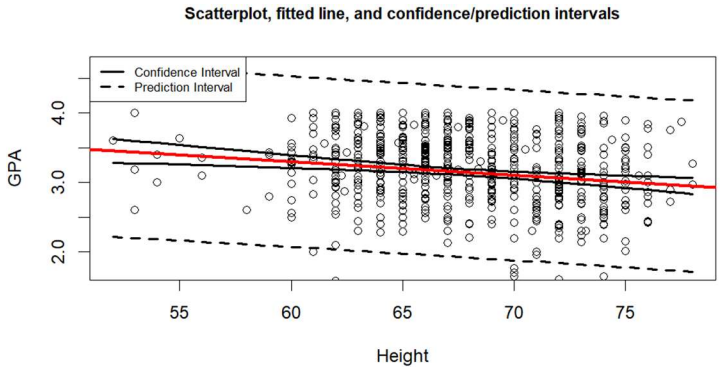
Mosaic Plot

```
mosaicplot(table(SURVEY10$Gender, SURVEY10$SigificantOther),
        main = "Mosaic Plot: Gender vs Significant Other",
        xlab = "Gender",
        ylab = "Significant Other",
        color = TRUE)
```

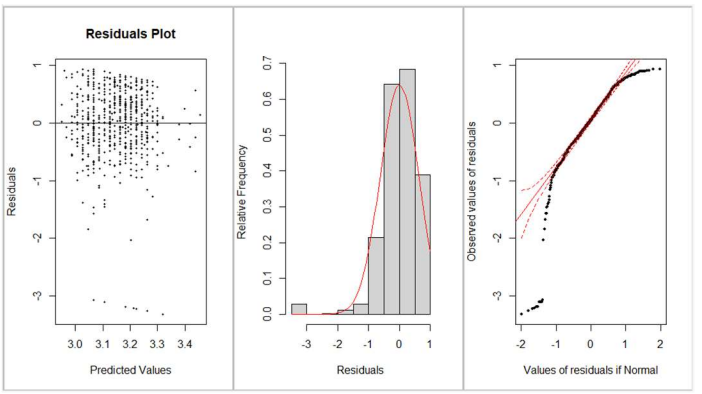


Simple Linear Regression

```
model <- lm(GPA ~ Height, data = SURVEY10)

visualize_model(model)

summary (model)

check_regression(model)
```

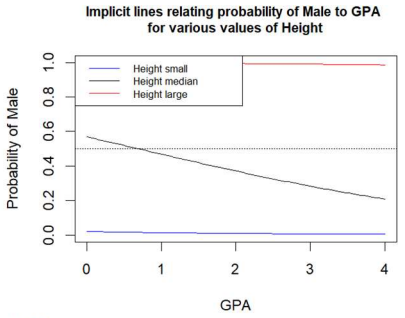


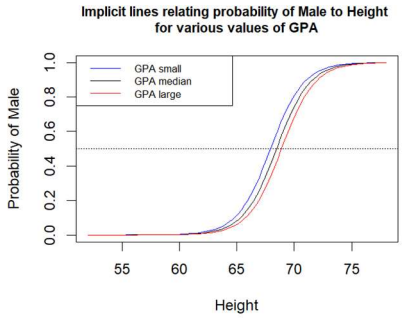

```
Call:
lm(formula = GPA ~ Height, data = SURVEY10)

Residuals:
     Min       1Q   Median       3Q      Max
-3.3196 -0.2856  0.0441  0.4087  0.9341

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.47113    0.36115  12.380  < 2e-16 ***
Height      -0.01952    0.00532  -3.669 0.000262 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6245 on 697 degrees of freedom
Multiple R-squared:  0.01895,    Adjusted R-squared:  0.01754
F-statistic: 13.46 on 1 and 697 DF,  p-value: 0.0002622
```



Logistic Regression

```
# Fit the logistic regression model
logit_model <- glm( Gender ~ GPA + Height,
                    data = SURVEY10,
                    family = binomial)
summary(logit_model)
visualize_model(logit_model)
confusion_matrix(logit_model)
```



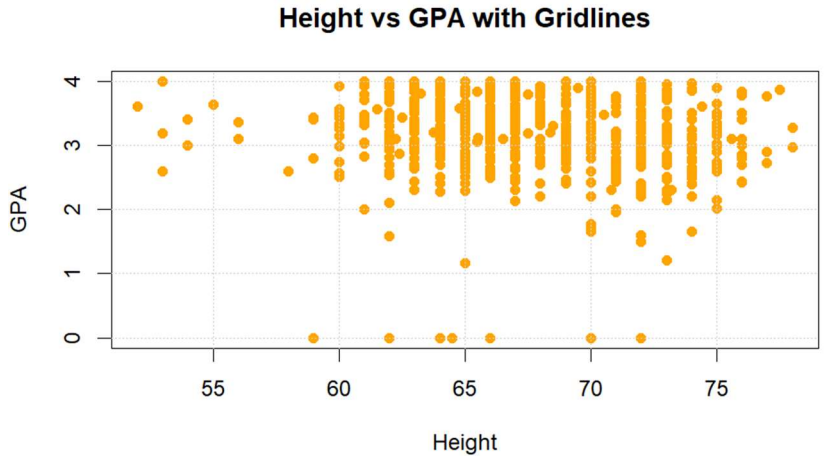



```
Call:
glm(formula = Gender ~ GPA + Height, family = binomial, data = SURVEY10)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -46.41885    3.57286 -12.992   <2e-16 ***
GPA          -0.40239    0.19278  -2.087   0.0369 *
Height        0.69694    0.05154  13.522   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

	Predicted Female	Predicted Male	Total
Actual Female	346	40	386
Actual Male	49	264	313
Total	395	304	699

Adding Gridlines

```
plot(SURVEY10$Height, SURVEY10$GPA,
    main = "Height vs GPA with Gridlines",
    xlab = "Height",
    ylab = "GPA",
    col = "blue",
    pch = 19)
grid()
```



Color Points by Group

```
plot(SURVEY10$Height, SURVEY10$GPA,
    col = ifelse(SURVEY10$Gender == "Male", "blue", "pink"),
    main = "Height vs GPA by Gender",
    xlab = "Height",
    ylab = "GPA",
    pch = 19)
legend("topright", legend = c("Male", "Female"), col = c("blue ", "pink"), pch = 19)
```