Effective Visualizations in R

Ahmed Yahya Khaled

9/19/2020

These visualizations are based on ‘Grammar of Graphics’ using R Package - **‘ggplot2’**

Install Packages

```
# install.packages('ggplot2')
# install.packages('tidyverse')
# install.packages("ggalt")
# install.packages('GGally')
# install.packages('ggridges')
```

notes :

1. **‘tidyverse’** is for overall data handling. It contains **‘ggplot2’** with other required packages like **‘dplyr’**.
2. **‘ggalt’** is especially required for **Dumbbell plot**
3. **‘GGally’** is especially required for **Matrix Scatter plot**
4. **‘ggridges’** is especially required for **Ridge plot**
5. Remove the **‘#’** to uncomment and run the above codes to install the packages
6. You will need to install the packages only once
7. R is case-sensitive !

Load packages

```
library(tidyverse)
library(ggalt)
library(GGally)
library(ggridges)
```

Explore Datasets

Inbuild datasets

```
# check dataset information
#?iris
#?diamonds

# load datasets
data_iris <- iris
data_diamonds <- diamonds

# look-over datasets
head(data_iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
head(data_diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

External Datasets

```
# set working directory
setwd("E:/My_R/github/R_Viz")
# note : write your own directory where you saved the csv files. make sure to change '\' by '/'
 !

# import datasets
data_names <- read.csv("names.csv", header = TRUE)
data_life_exp <- read.csv("life_expectency.csv", header = TRUE)

# look-over datasets
head(data_names)
```

```
##   year     name   n
## 1 1880    Helen 636
## 2 1881    Helen 612
## 3 1882    Helen 838
## 4 1883    Helen 862
## 5 1884    Helen 986
## 6 1884 Patricia   6
```

```
head(data_life_exp)
```

```
##        country  Y1967  Y2007    gap
## 1        Egypt 49.293 71.338 22.045
## 2        Nepal 41.472 63.785 22.313
## 3 Saudi Arabia 49.901 72.777 22.876
## 4        Libya 50.227 73.952 23.725
## 5    Indonesia 45.964 70.650 24.686
## 6  Yemen, Rep. 36.984 62.698 25.714
```

View Datasets

```
View(data_iris)
View(data_diamonds)
View(data_names)
View(data_life_exp)
```

Scatter Plot

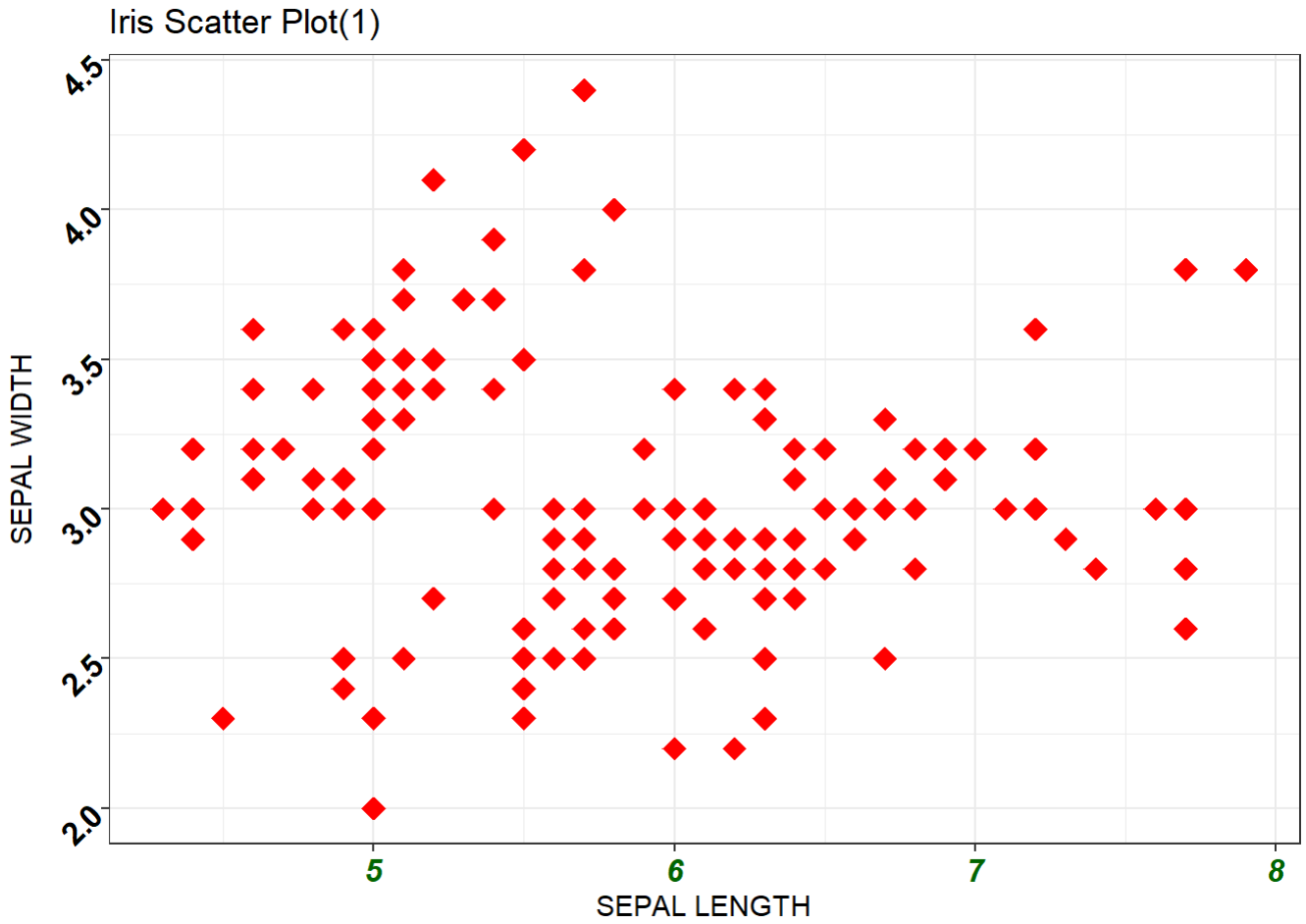
dataset : iris

Basic Scatter Plot

```
ggplot(data_iris) +
  geom_point(aes(x = Sepal.Length , y = Sepal.Width), color = 'red', size = 4, pch = 18) +

  ggtitle("Iris Scatter Plot(1)") +
  xlab("SEPAL LENGTH") +
  ylab("SEPAL WIDTH") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold.italic', color = 'darkgreen',
                                   size = 12, angle = 0),
        axis.text.y = element_text(face = 'bold', color = 'black',
                                   size = 12, angle = 45))
```

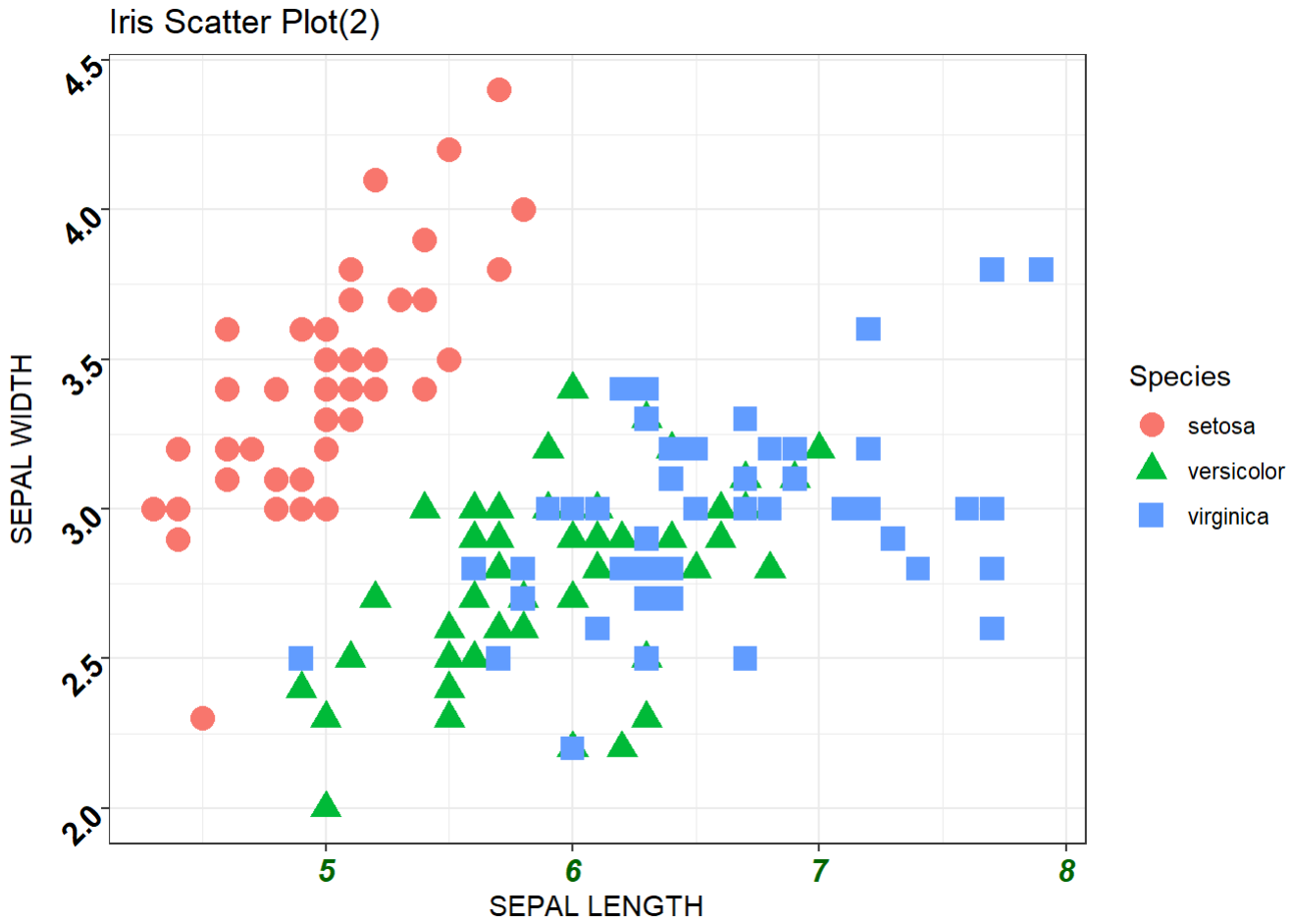


Scatter Plot with Feature differentiation by Color & Shape

```
ggplot(data_iris) +
  geom_point(aes(x = Sepal.Length , y = Sepal.Width,
                 color = Species, shape = Species), size = 4) +

  ggtitle("Iris Scatter Plot(2)") +
  xlab("SEPAL LENGTH") +
  ylab("SEPAL WIDTH") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold.italic', color = 'darkgreen',
                                   size = 12, angle = 0),
        axis.text.y = element_text(face = 'bold', color = 'black',
                                   size = 12, angle = 45))
```



Scatter Plot with Feature differentiation by Size and Transparency

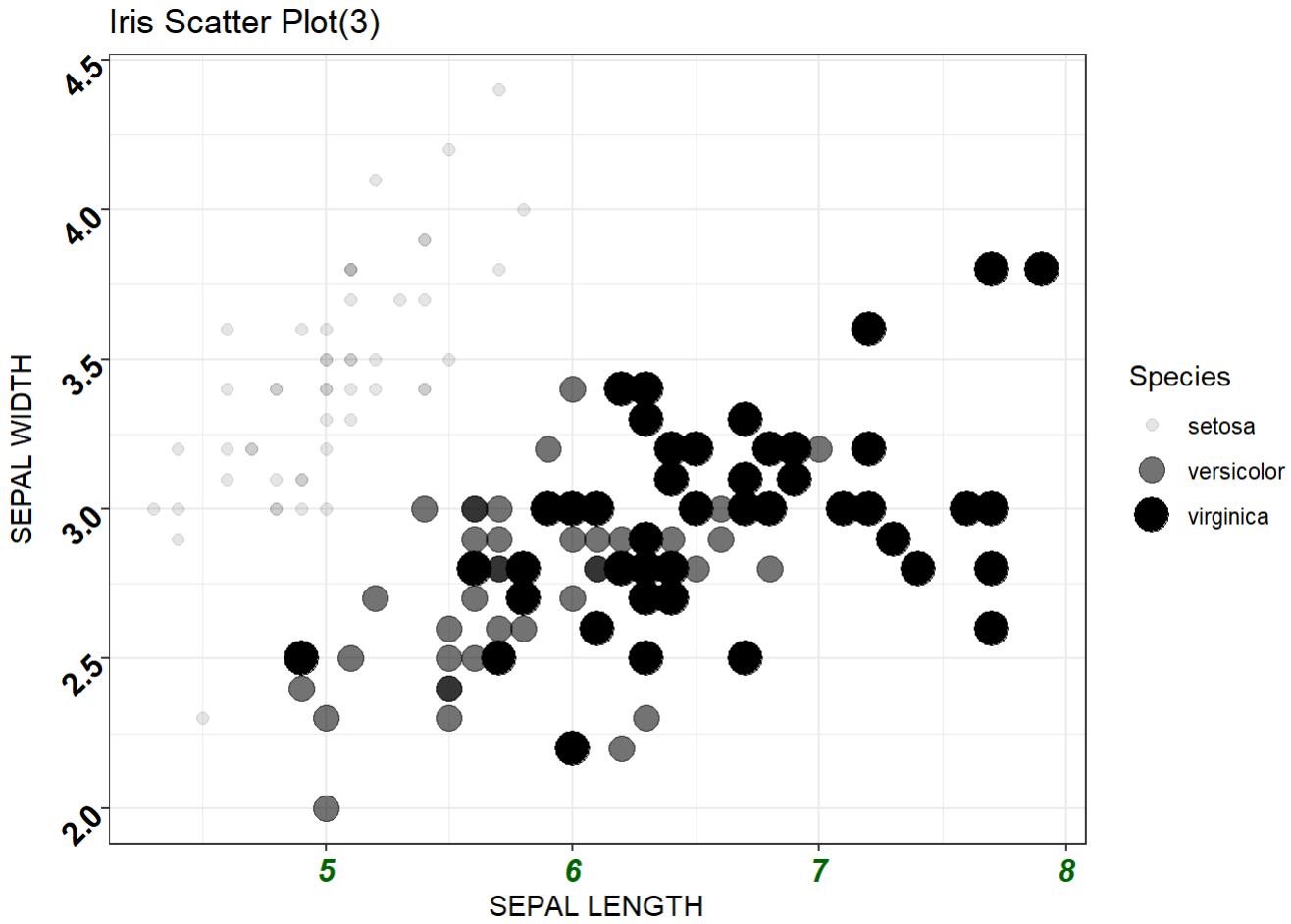
```
ggplot(data_iris) +
  geom_point(aes(x = Sepal.Length , y = Sepal.Width,
                 size = Species, alpha = Species)) +

  ggtitle("Iris Scatter Plot(3)") +
  xlab("SEPAL LENGTH") +
  ylab("SEPAL WIDTH") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold.italic', color = 'darkgreen',
                                   size = 12, angle = 0),
        axis.text.y = element_text(face = 'bold', color = 'black',
                                   size = 12, angle = 45))
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: Using alpha for a discrete variable is not advised.
```



Line Plot

dataset : names

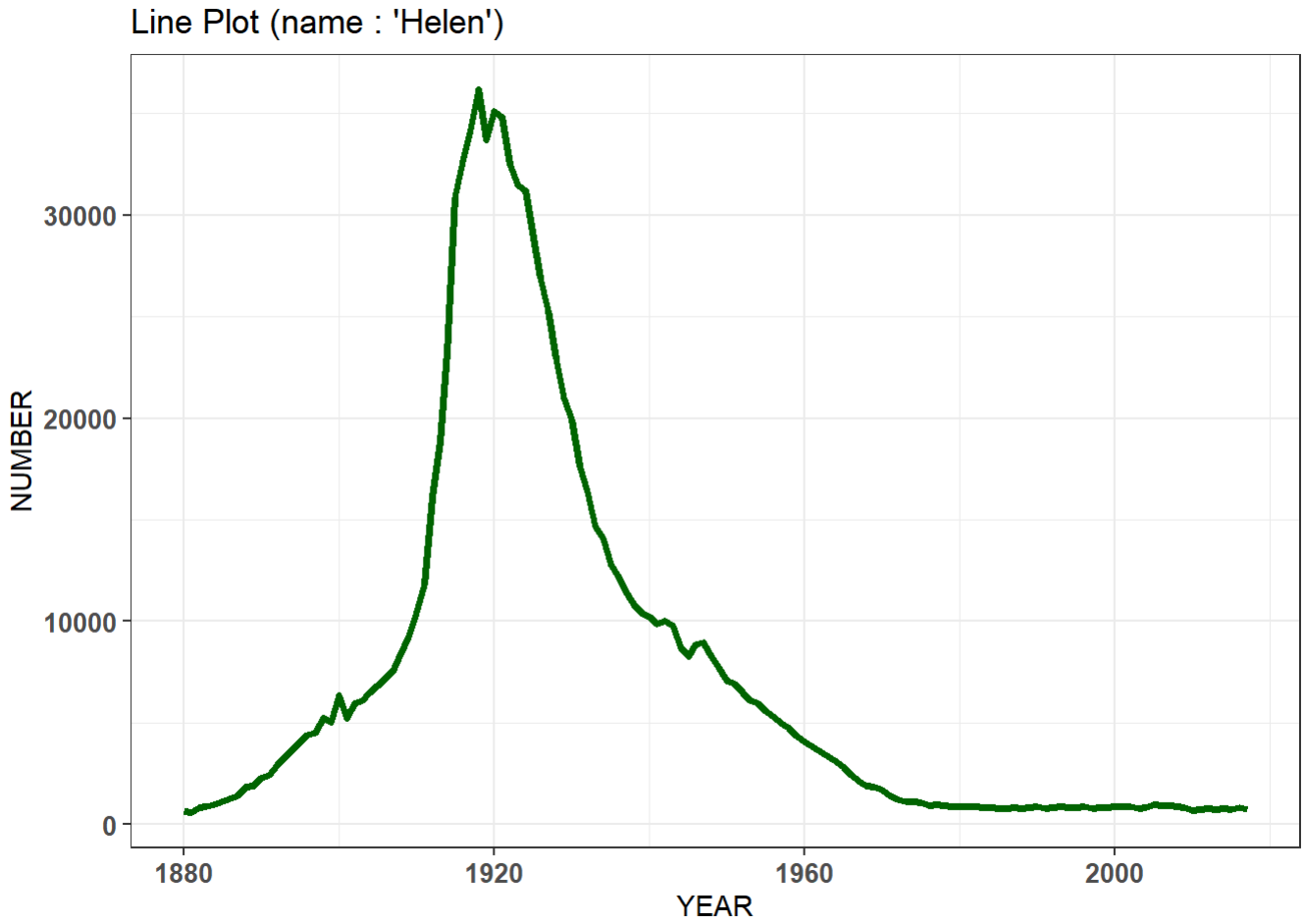
Basic Line Plot

```
d0 <- data_names %>% filter(name == 'Helen')

ggplot(d0) +
  geom_line(aes(x = year, y = n), lwd = 1.25, color = 'darkgreen') +

  ggtitle("Line Plot (name : 'Helen')") +
  xlab("YEAR") +
  ylab("NUMBER") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

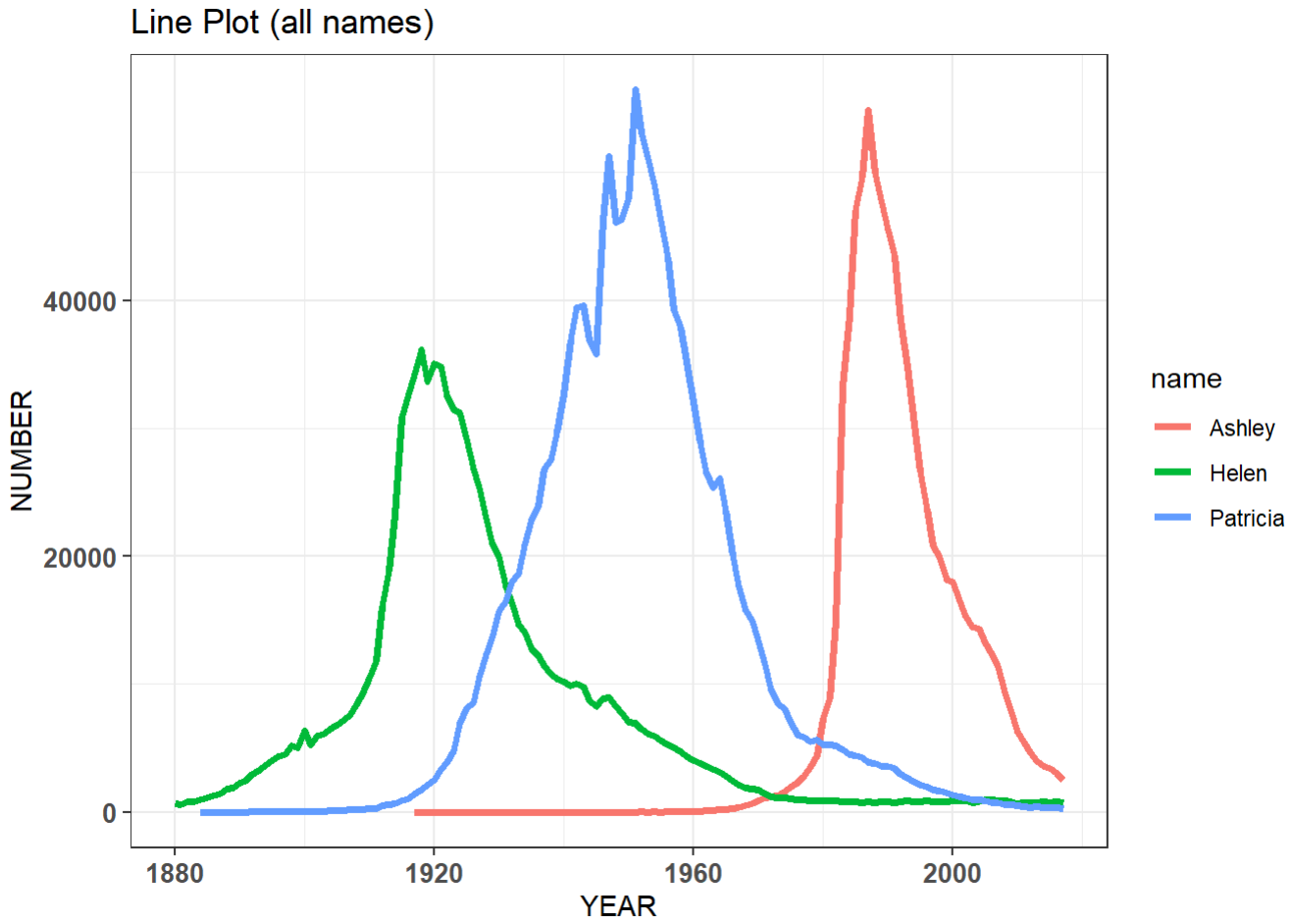


Line Plot with Feature differentiation by Color

```
ggplot(data_names) +
  geom_line(aes(x = year, y = n, color = name), lwd = 1.25) +

  ggtitle("Line Plot (all names)") +
  xlab("YEAR") +
  ylab("NUMBER") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Bar Plot

dataset : diamonds

Basic Bar Plot (1)

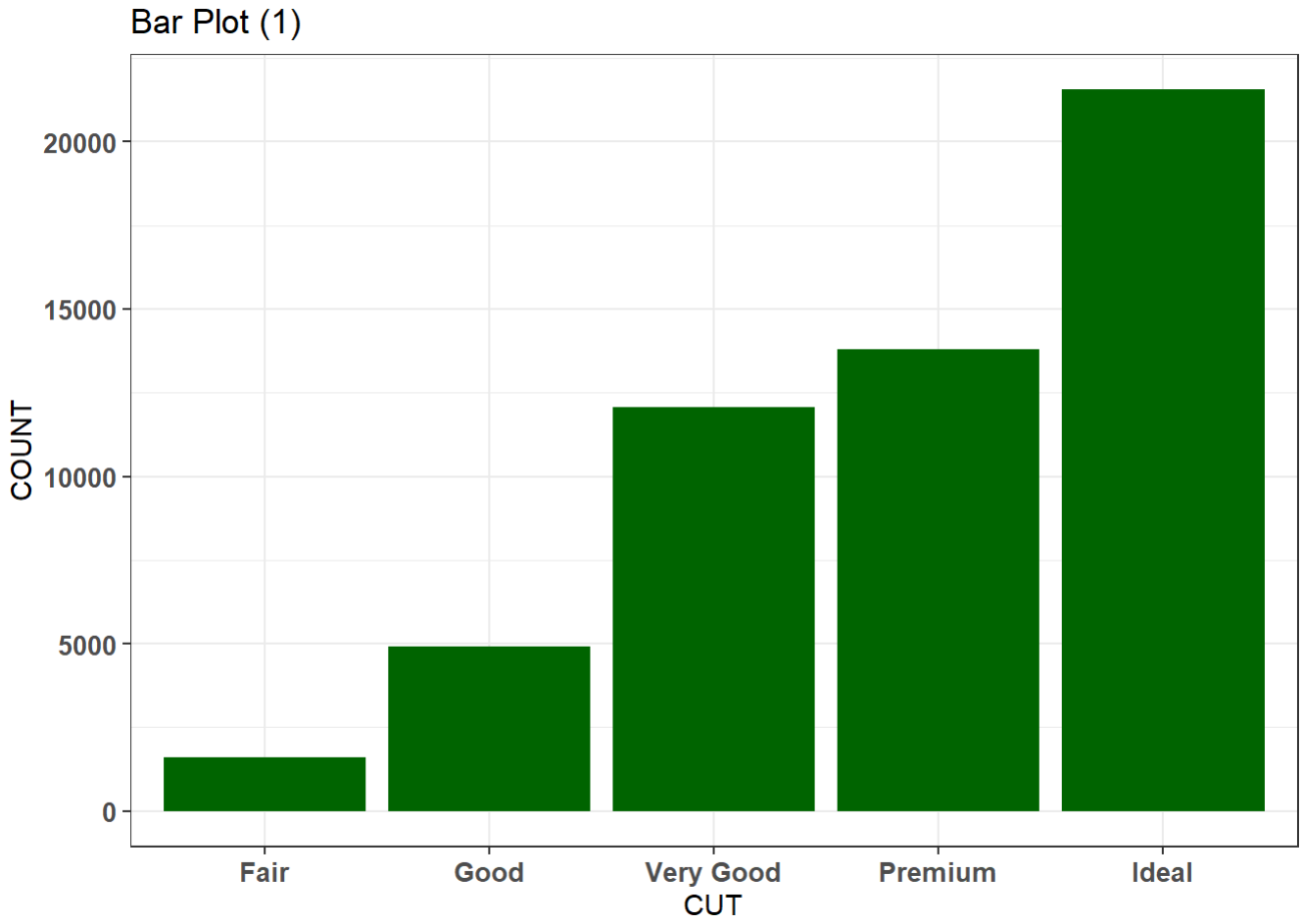
```
d1 <- as.data.frame(table(data_diamonds$cut))
head(d1)
```

```
##        Var1  Freq
## 1      Fair  1610
## 2      Good  4906
## 3 Very Good 12082
## 4   Premium 13791
## 5     Ideal 21551
```

```
ggplot(d1) +
  geom_bar(aes(x = Var1, y = Freq), stat = 'identity', fill = 'darkgreen') +

  ggtitle("Bar Plot (1)") +
  xlab("CUT") +
  ylab("COUNT") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Basic Bar Plot transformed to Polar

```
d1 <- as.data.frame(table(data_diamonds$cut))
head(d1)
```

```
##        Var1  Freq
## 1      Fair  1610
## 2      Good  4906
## 3 Very Good 12082
## 4   Premium 13791
## 5     Ideal 21551
```

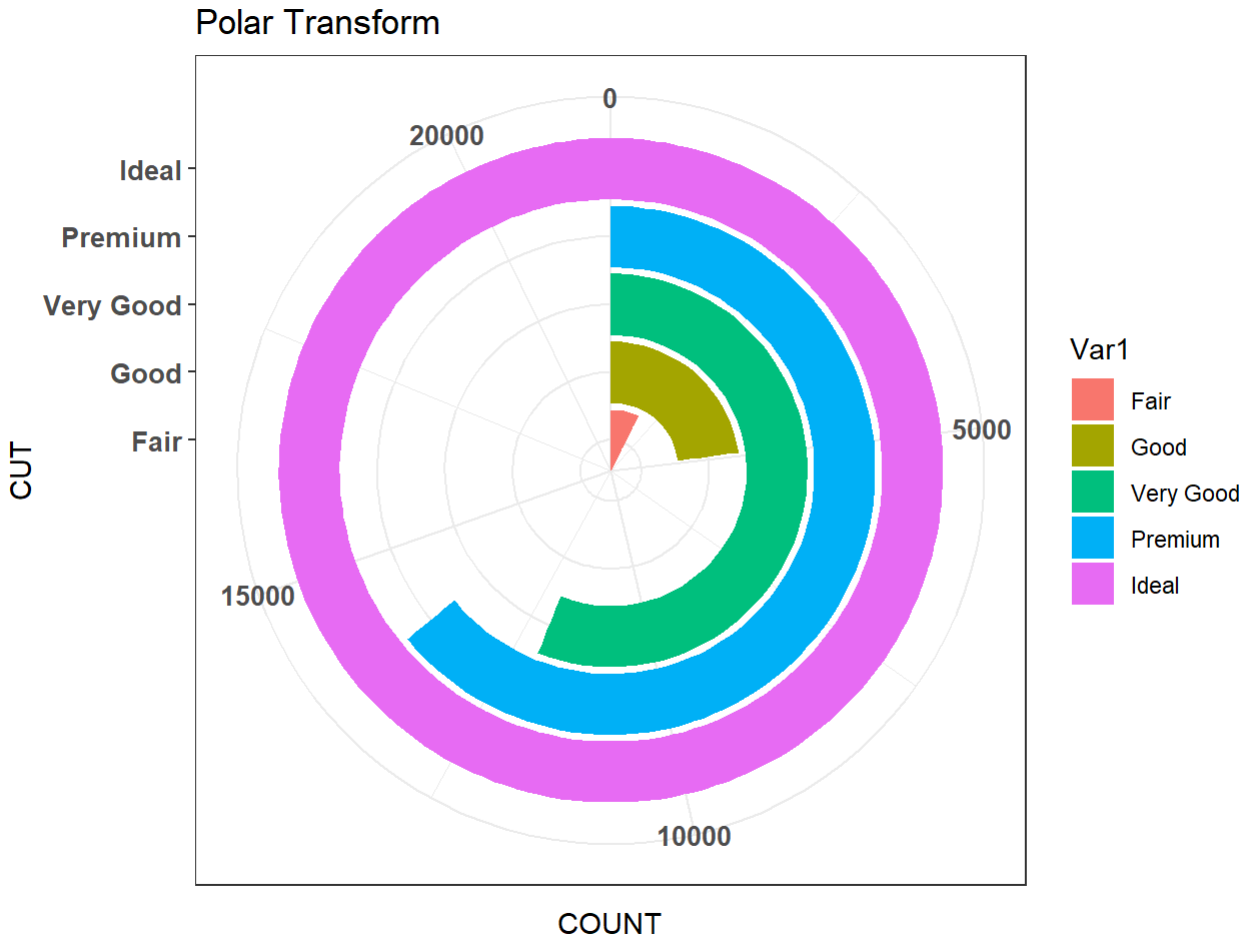
```
ggplot(d1) +
  geom_bar(aes(x = Var1, y = Freq, fill = Var1), stat = 'identity') +

  ggtitle("Polar Transform") +
  xlab("CUT") +
  ylab("COUNT") +

  coord_polar("y") +

  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

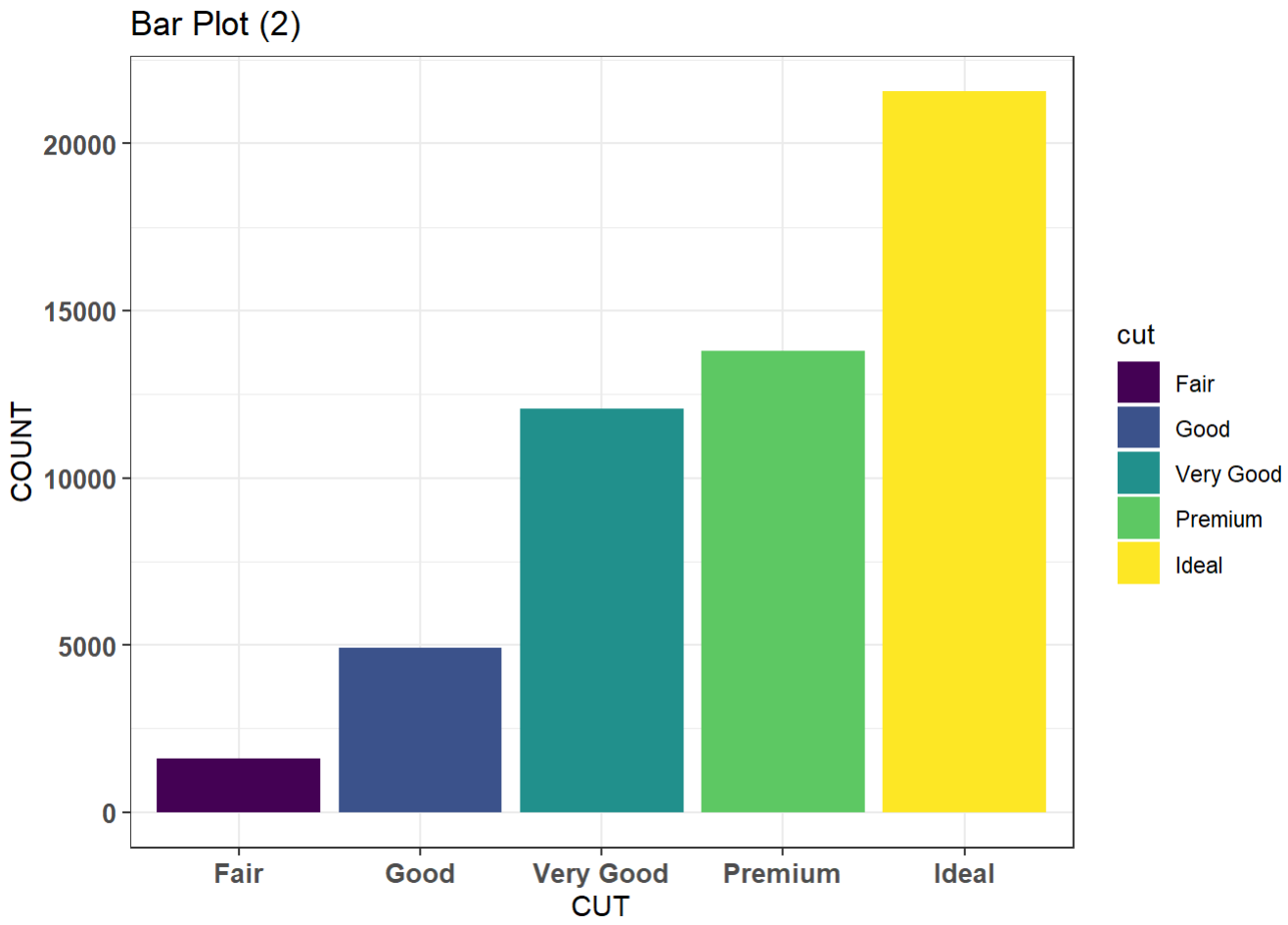


Basic Bar Plot (2)

```
ggplot(data_diamonds) +
  geom_bar(aes(x = cut, fill = cut)) +

  ggtitle("Bar Plot (2)") +
  xlab("CUT") +
  ylab("COUNT") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

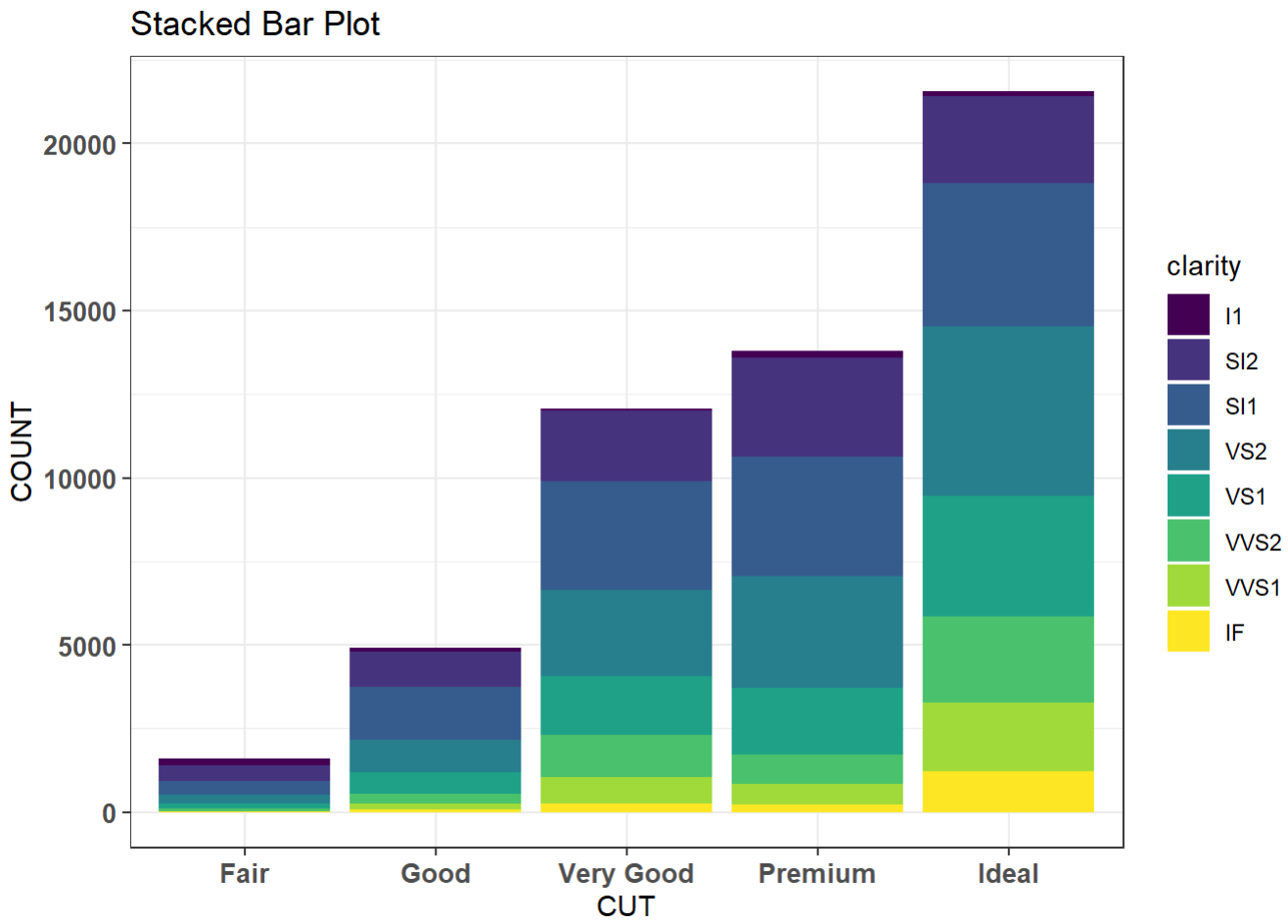


Stacked Bar Plot

```
ggplot(data_diamonds) +
  geom_bar(aes(x = cut, fill = clarity)) +

  ggtitle("Stacked Bar Plot") +
  xlab("CUT") +
  ylab("COUNT") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

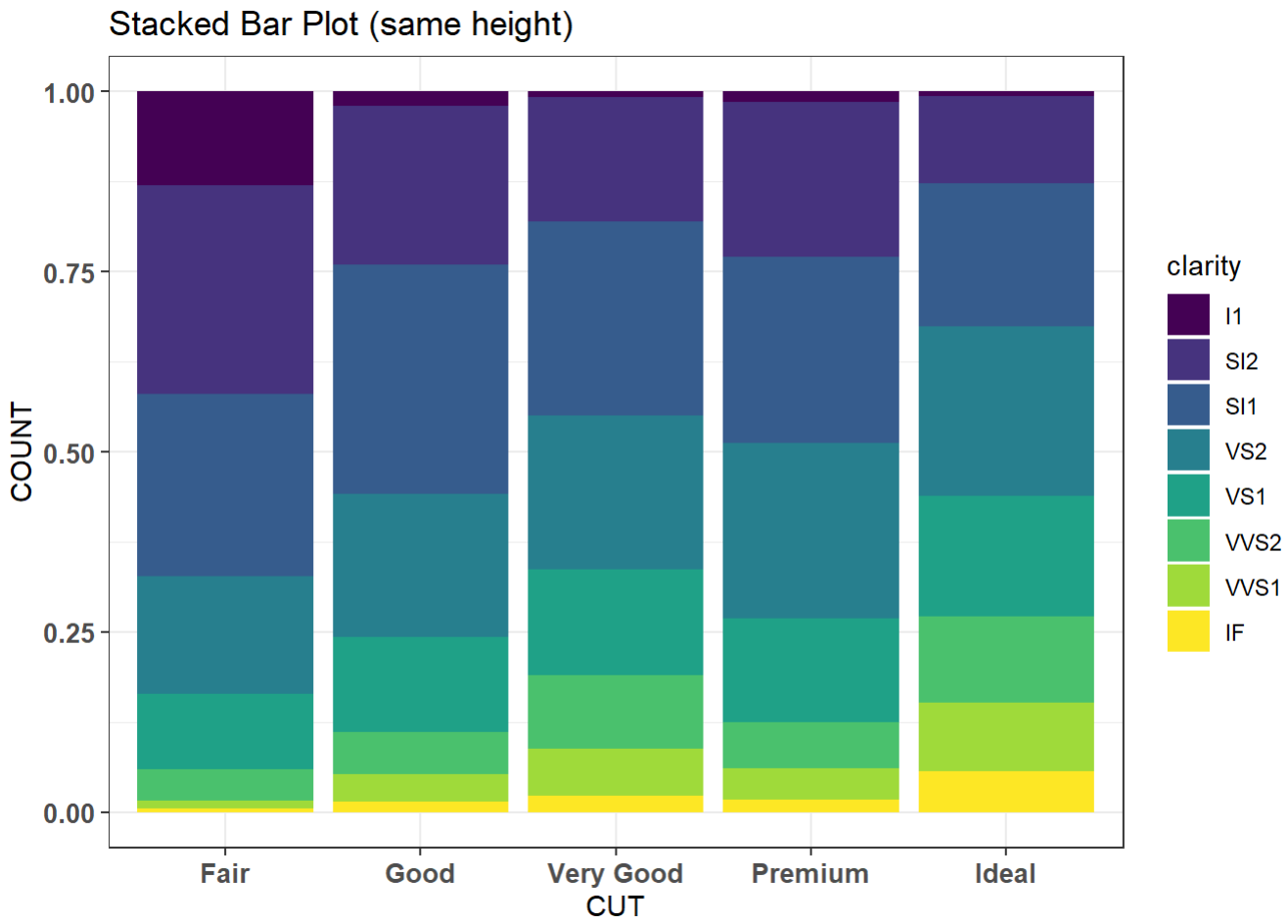


Stacked Bar Plot (Same Height)

```
ggplot(data_diamonds) +
  geom_bar(aes(x = cut, fill = clarity), position = 'fill') +

  ggtitle("Stacked Bar Plot (same height)") +
  xlab("CUT") +
  ylab("COUNT") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

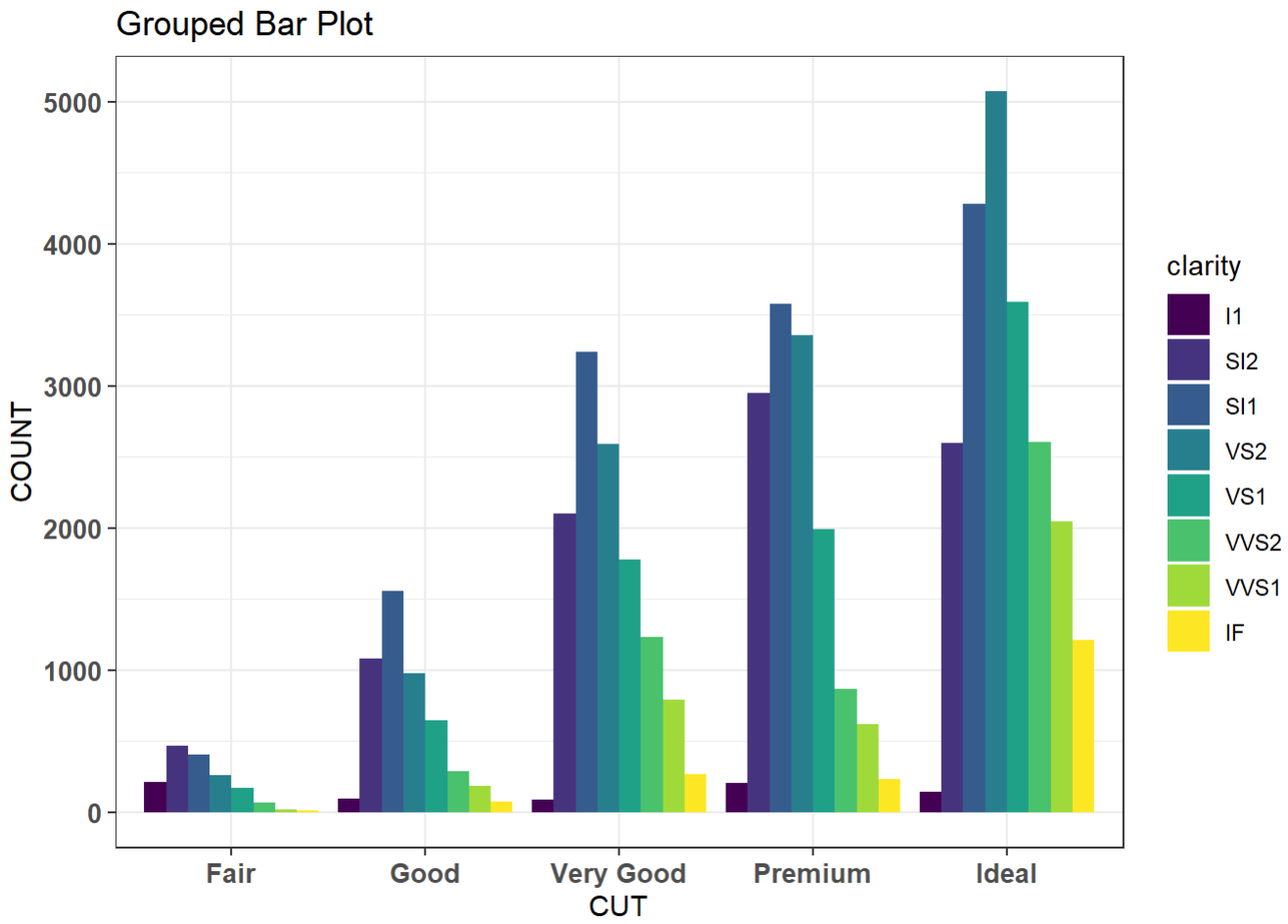


Grouped Bar Plot

```
ggplot(data_diamonds) +
  geom_bar(aes(x = cut, fill = clarity), position = 'dodge') +

  ggtitle("Grouped Bar Plot") +
  xlab("CUT") +
  ylab("COUNT") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Stacked Bar Plot transformed to Polar (1)

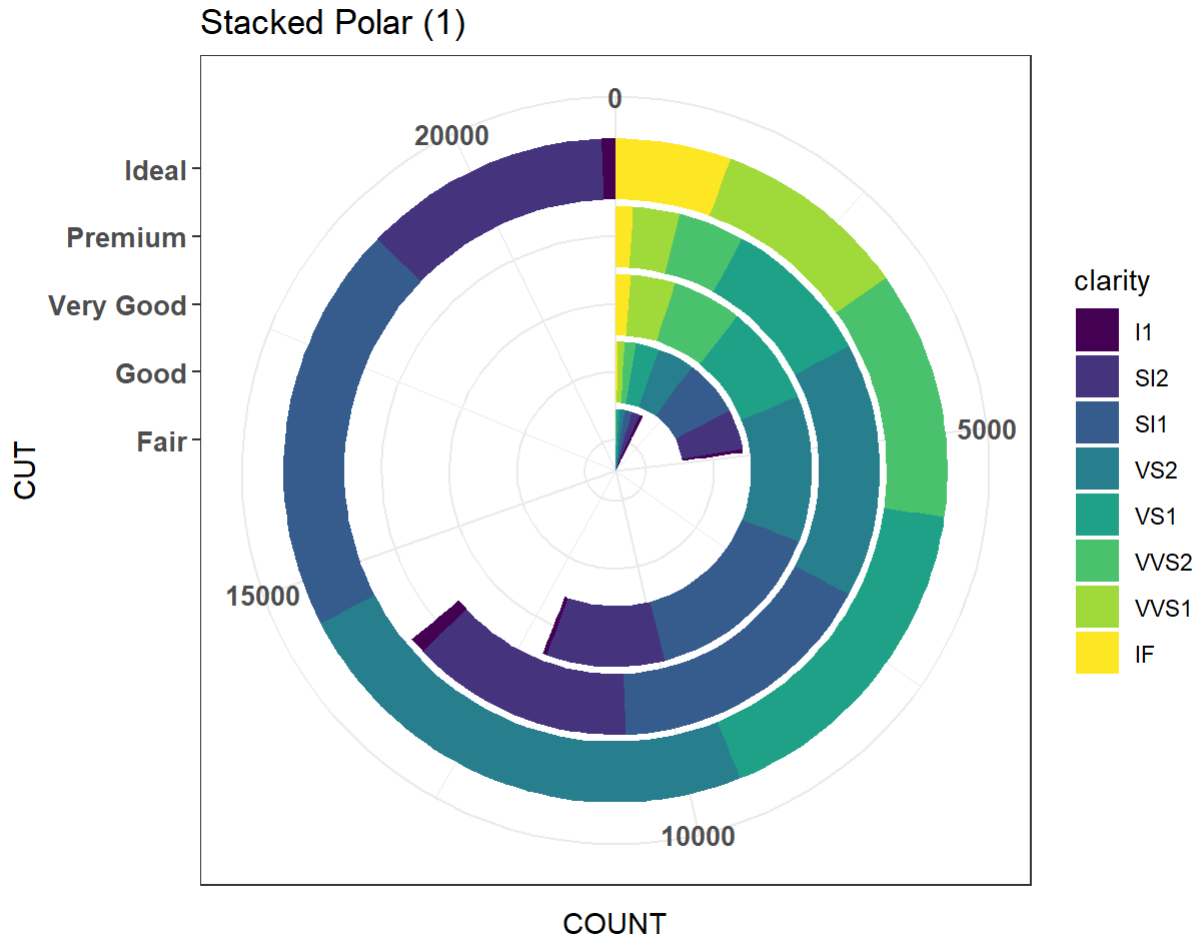
```
ggplot(data_diamonds) +
  geom_bar(aes(x = cut, fill = clarity)) +

  ggtitle("Stacked Polar (1)") +
  xlab("CUT") +
  ylab("COUNT") +

  coord_polar("y") +

  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Stacked Bar Plot transformed to Polar (2)

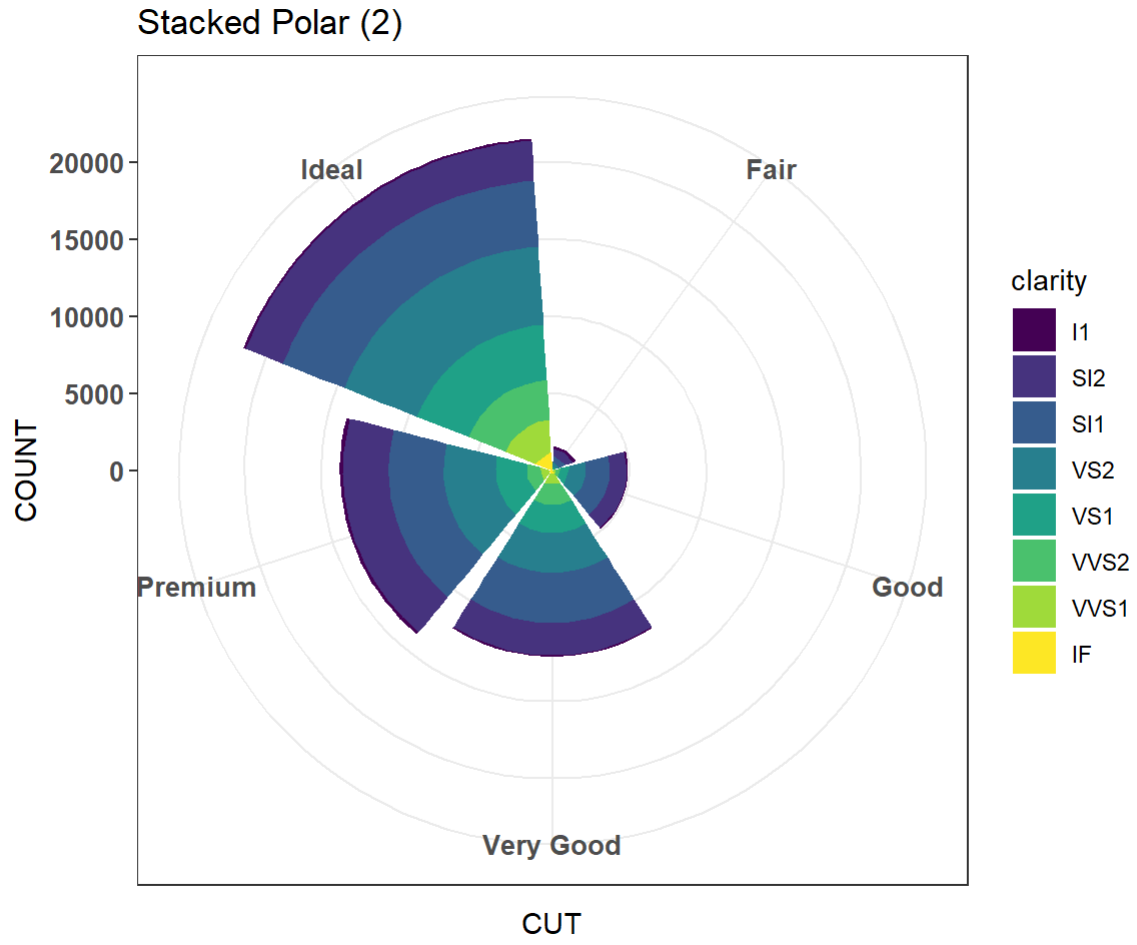
```
ggplot(data_diamonds) +
  geom_bar(aes(x = cut, fill = clarity)) +

  ggtitle("Stacked Polar (2)") +
  xlab("CUT") +
  ylab("COUNT") +

  coord_polar() +

  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Tiles Plot

dataset : diamonds

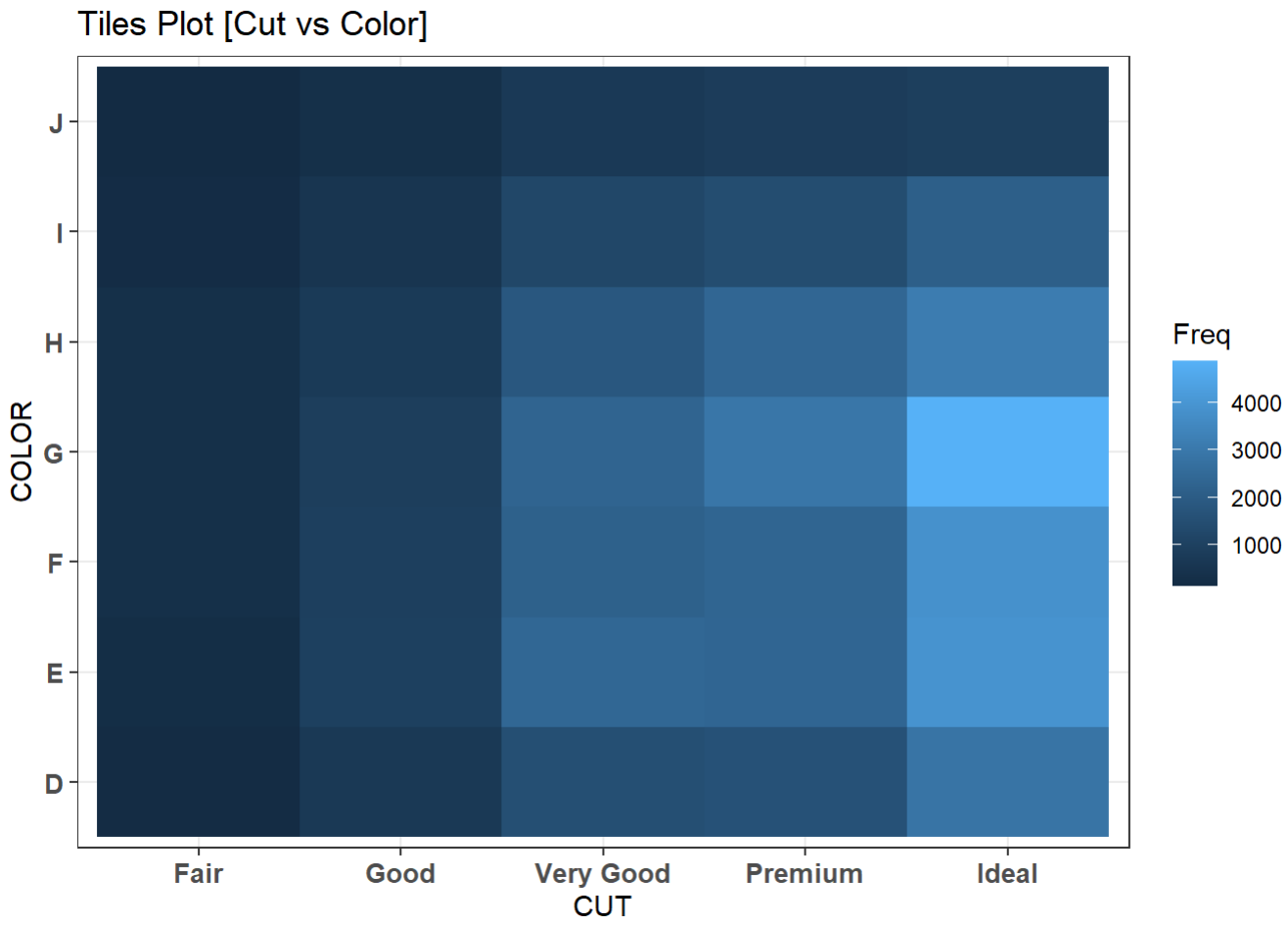
```
d2 <- as.data.frame(table(data_diamonds$cut, data_diamonds$color))
head(d2)
```

```
##        Var1 Var2 Freq
## 1      Fair    D  163
## 2      Good    D  662
## 3 Very Good    D 1513
## 4   Premium    D 1603
## 5     Ideal    D 2834
## 6      Fair    E  224
```

```
ggplot(d2) +
  geom_tile(aes(x = Var1, y = Var2, fill = Freq)) +

  ggtitle("Tiles Plot [Cut vs Color]") +
  xlab("CUT") +
  ylab("COLOR") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



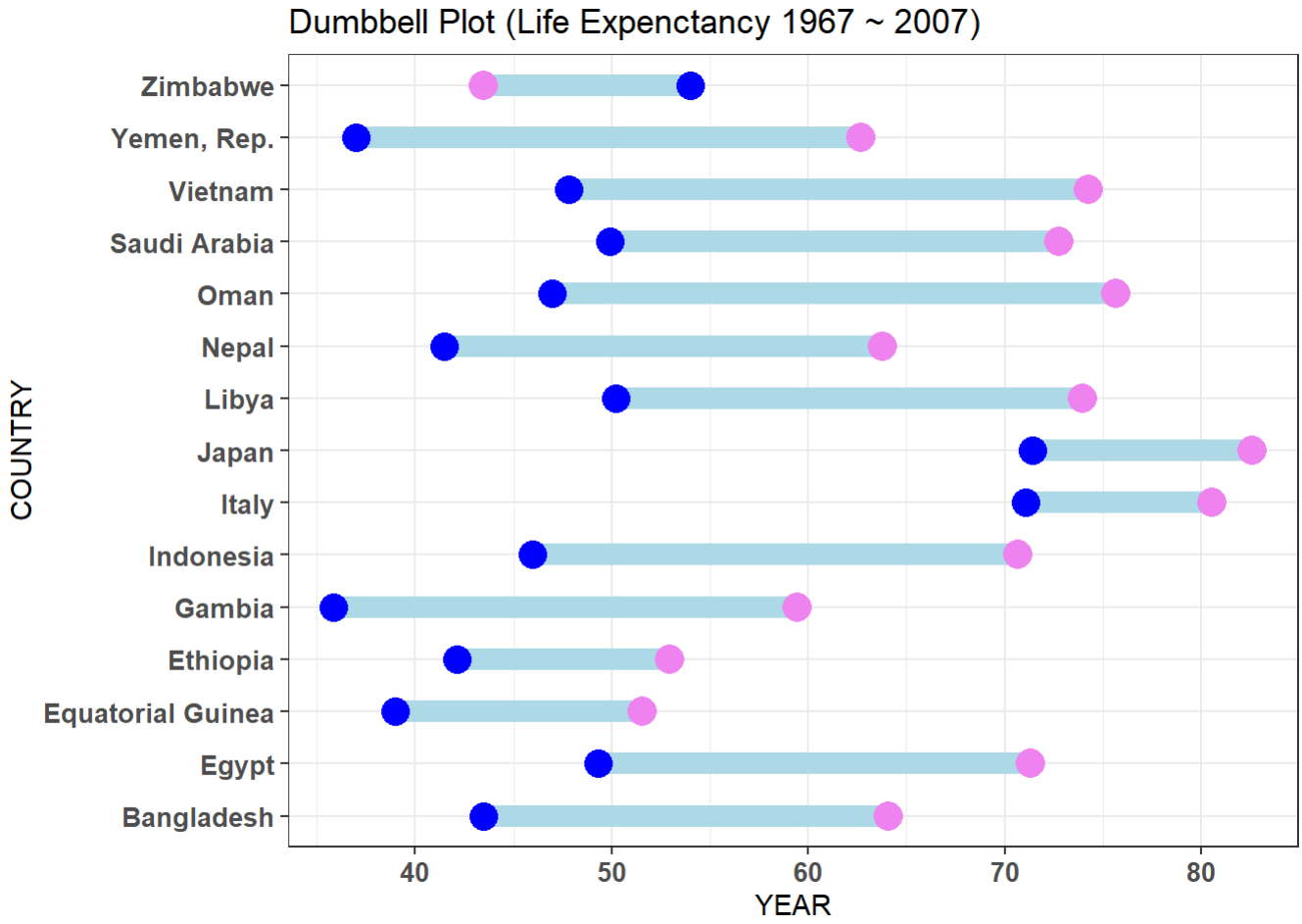
Dumbbell Plot

dataset : life_exp

```
ggplot(data_life_exp) +
  geom_dumbbell(aes(x = Y1967, xend = Y2007, y = country, group = country),
                colour = 'lightblue', size = 4,
                colour_x = 'blue', colour_xend = 'violet') +

  ggtitle('Dumbbell Plot (Life Expenctancy 1967 ~ 2007)') +
  xlab('YEAR') +
  ylab('COUNTRY') +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

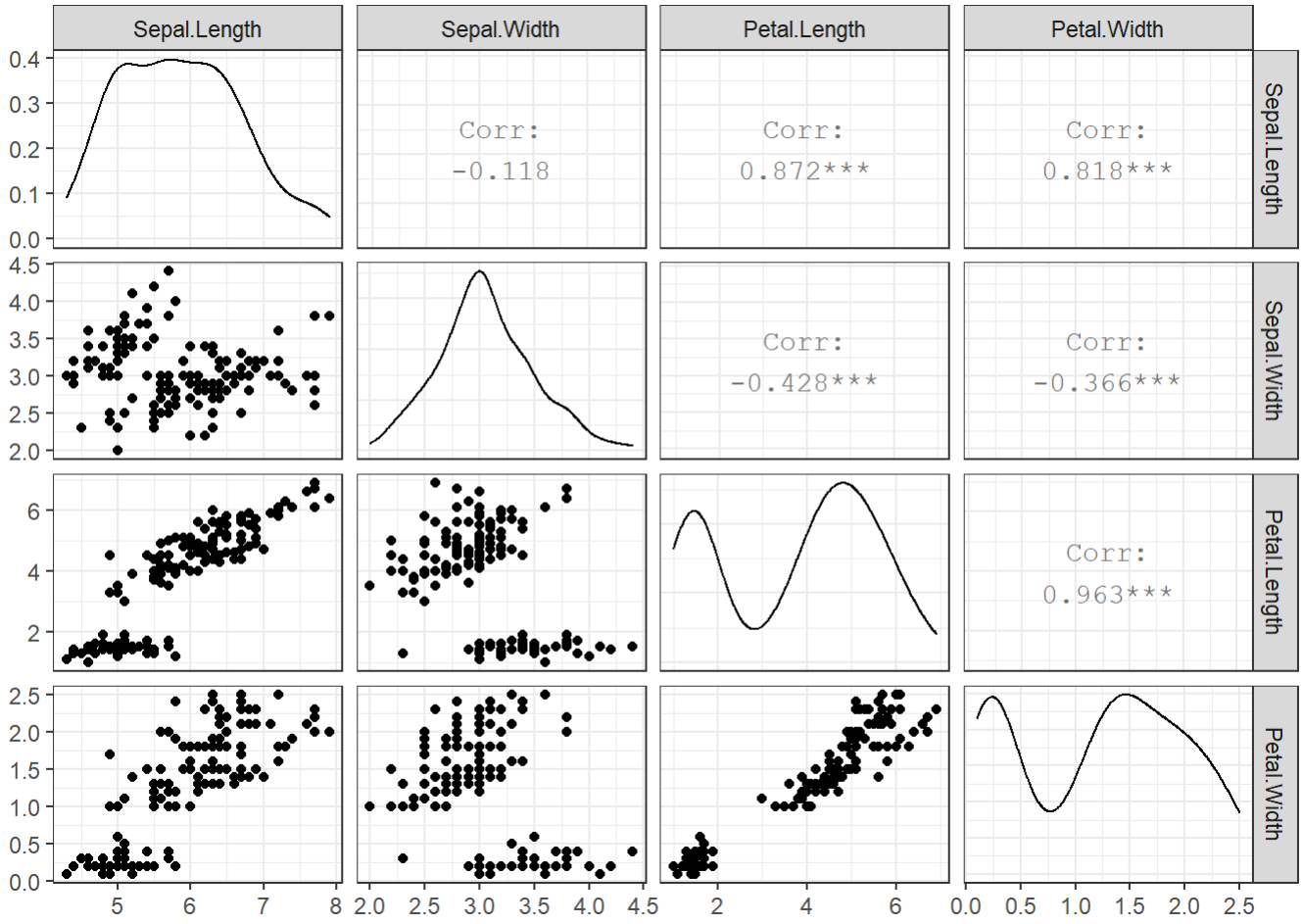


Matrix Scatter Plot

dataset : iris

Basic Matrix Scatter Plot

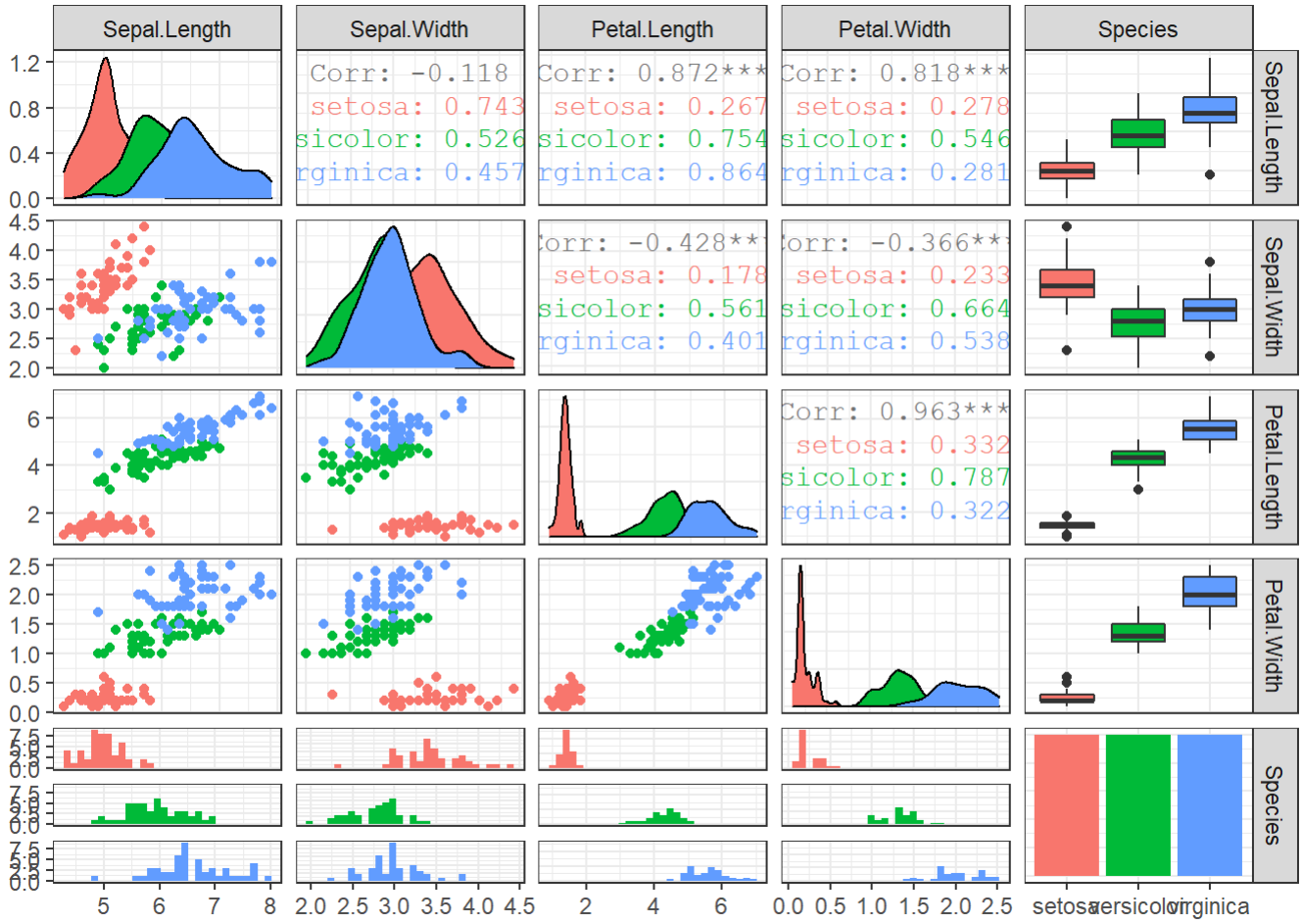
```
ggpairs(data_iris[ , 1:4]) +
  theme_bw()
```



Matrix Scatter Plot with Feature Differentation

```
ggpairs(data_iris, aes(color = Species)) +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram

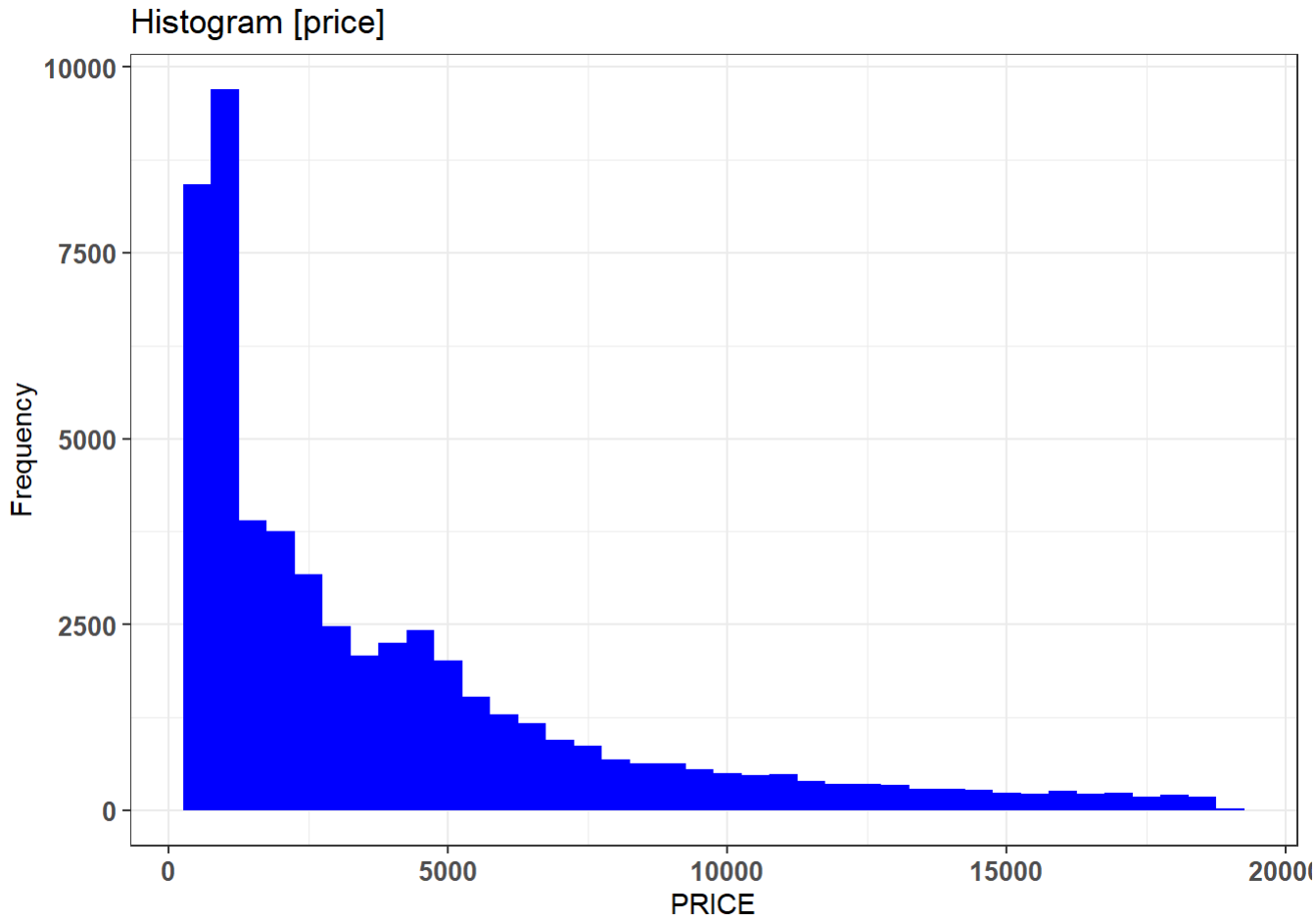
dataset : diamonds

Basic Histogram

```
ggplot(data_diamonds) +
  geom_histogram(aes(x = price), fill = 'blue', binwidth = 500) +

  ggtitle("Histogram [price]") +
  xlab("PRICE") +
  ylab("Frequency") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

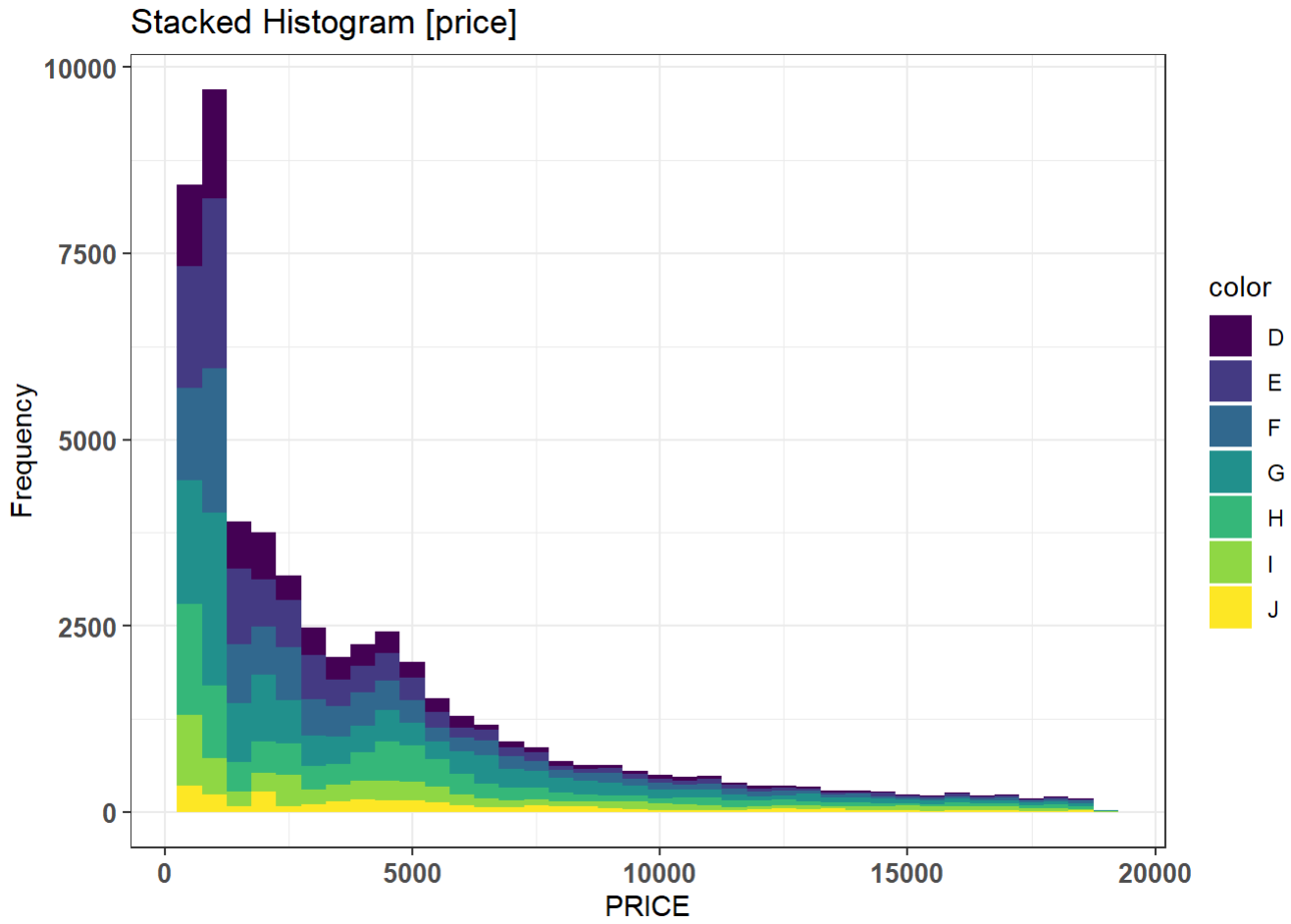


Stacked Histogram

```
ggplot(data_diamonds) +
  geom_histogram(aes(x = price, fill = color), binwidth = 500) +

  ggtitle("Stacked Histogram [price]") +
  xlab("PRICE") +
  ylab("Frequency") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Box Plot

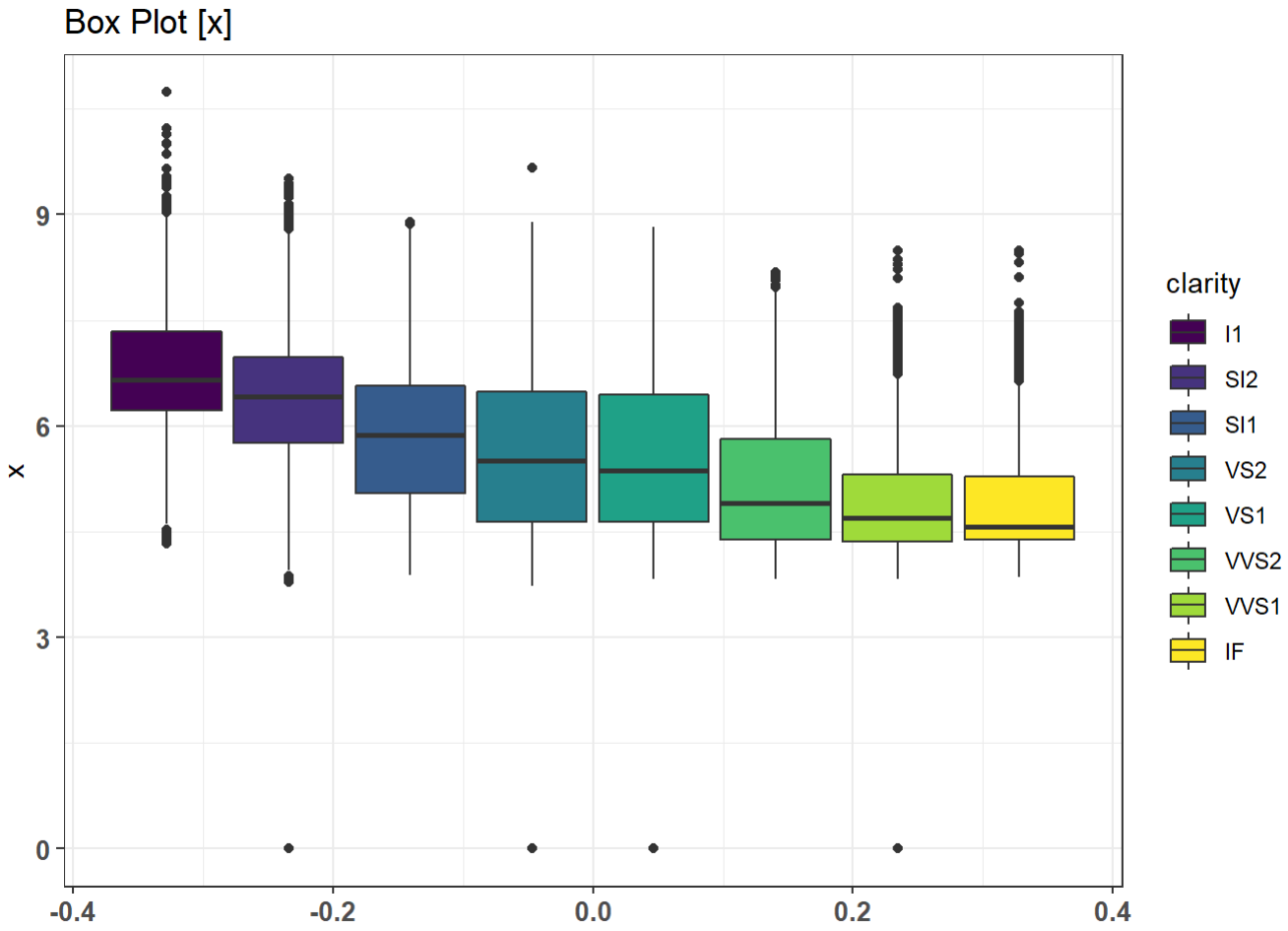
dataset : diamonds

Basic Box Plot

```
ggplot(data_diamonds) +
  geom_boxplot(aes(x = x, fill = clarity)) +
  coord_flip() +

  ggtitle("Box Plot [x]") +
  xlab("x") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



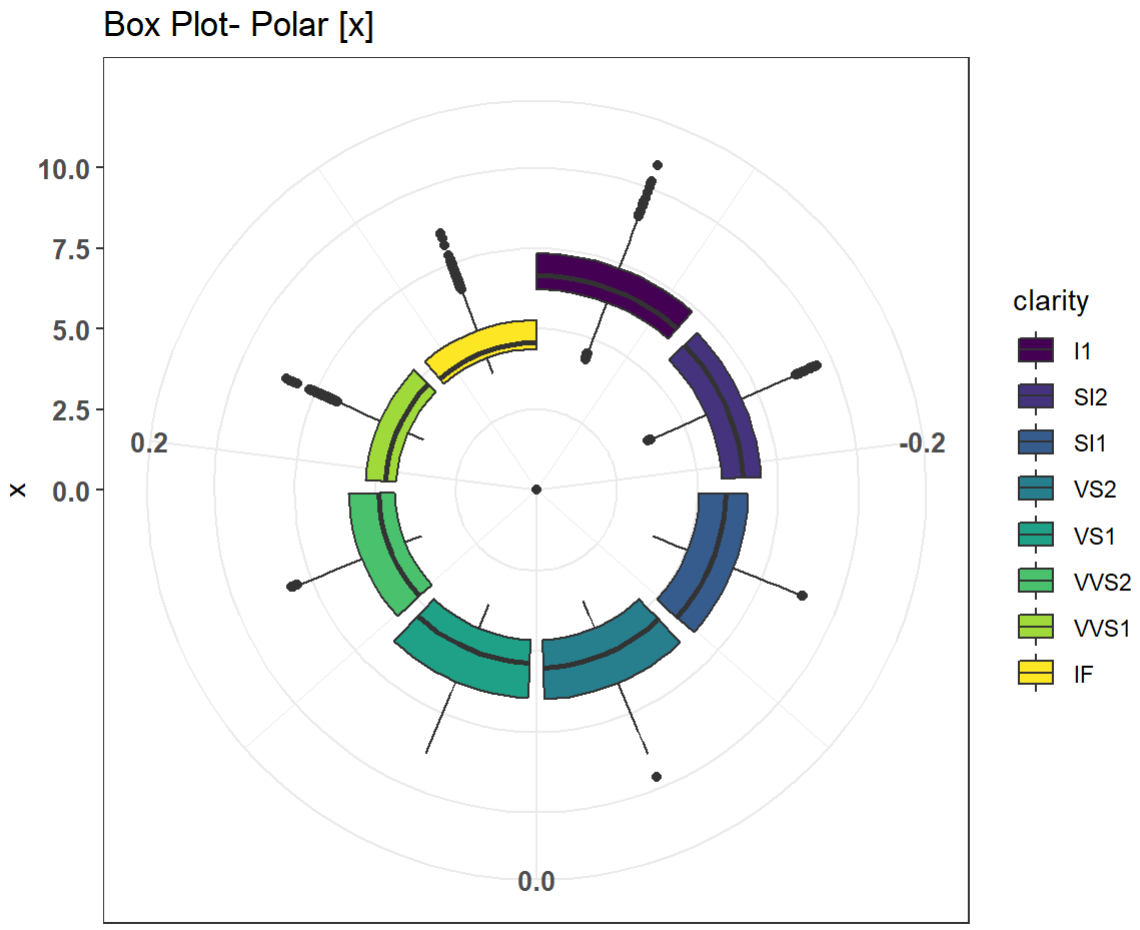
Box Plot Transferred to Polar

```
ggplot(data_diamonds) +
  geom_boxplot(aes(x = x, fill = clarity)) +
  coord_flip() +
  coord_polar("y") +

  ggtitle("Box Plot- Polar [x]") +
  xlab("x") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the exist
ing one.
```



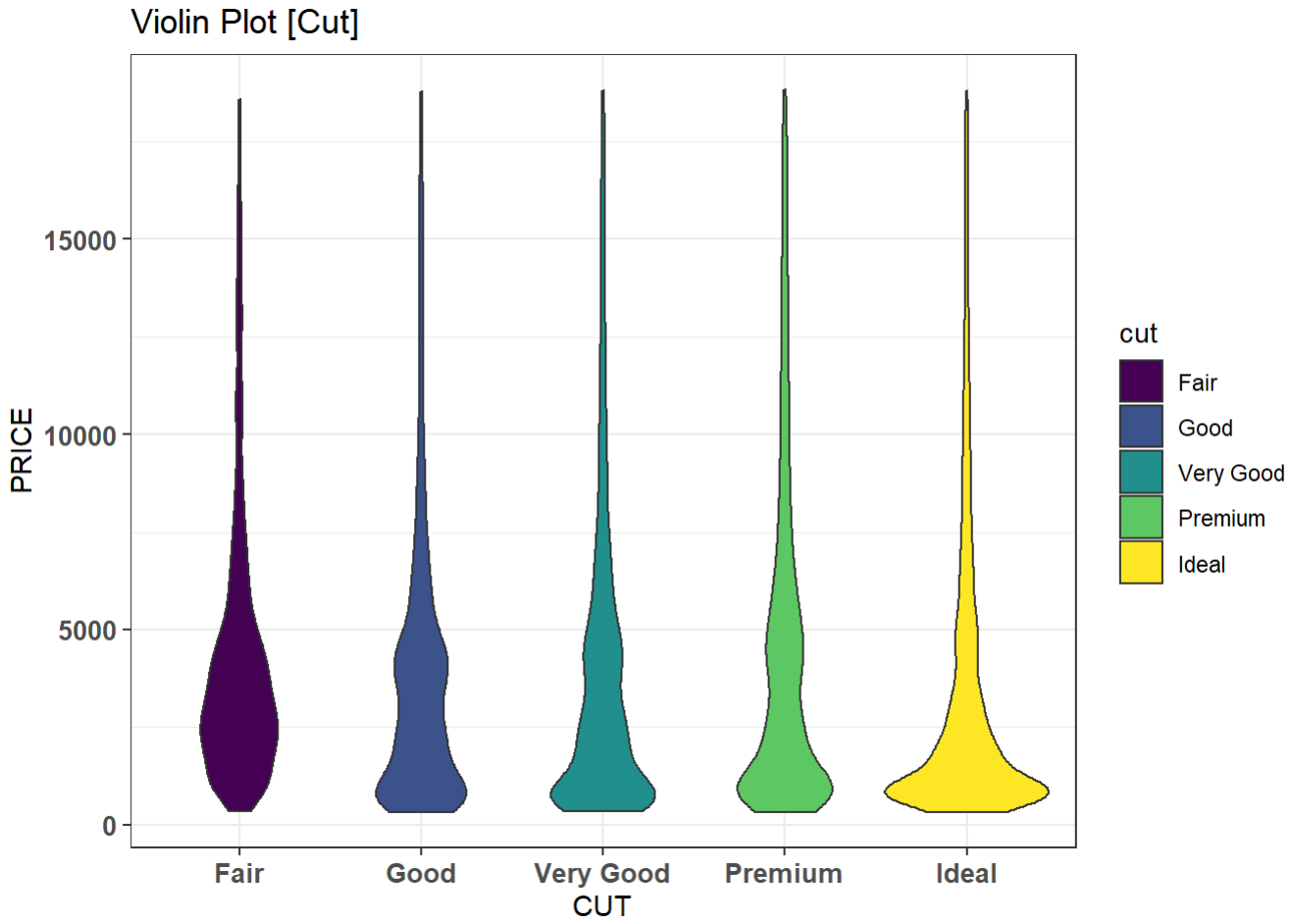
Violin Plot

dataset : diamonds

```
ggplot(data_diamonds) +
  geom_violin(aes(x = cut , y = price, fill = cut)) +

  ggtitle("Violin Plot [Cut]") +
  xlab("CUT") +
  ylab("PRICE") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```



Ridge Plot

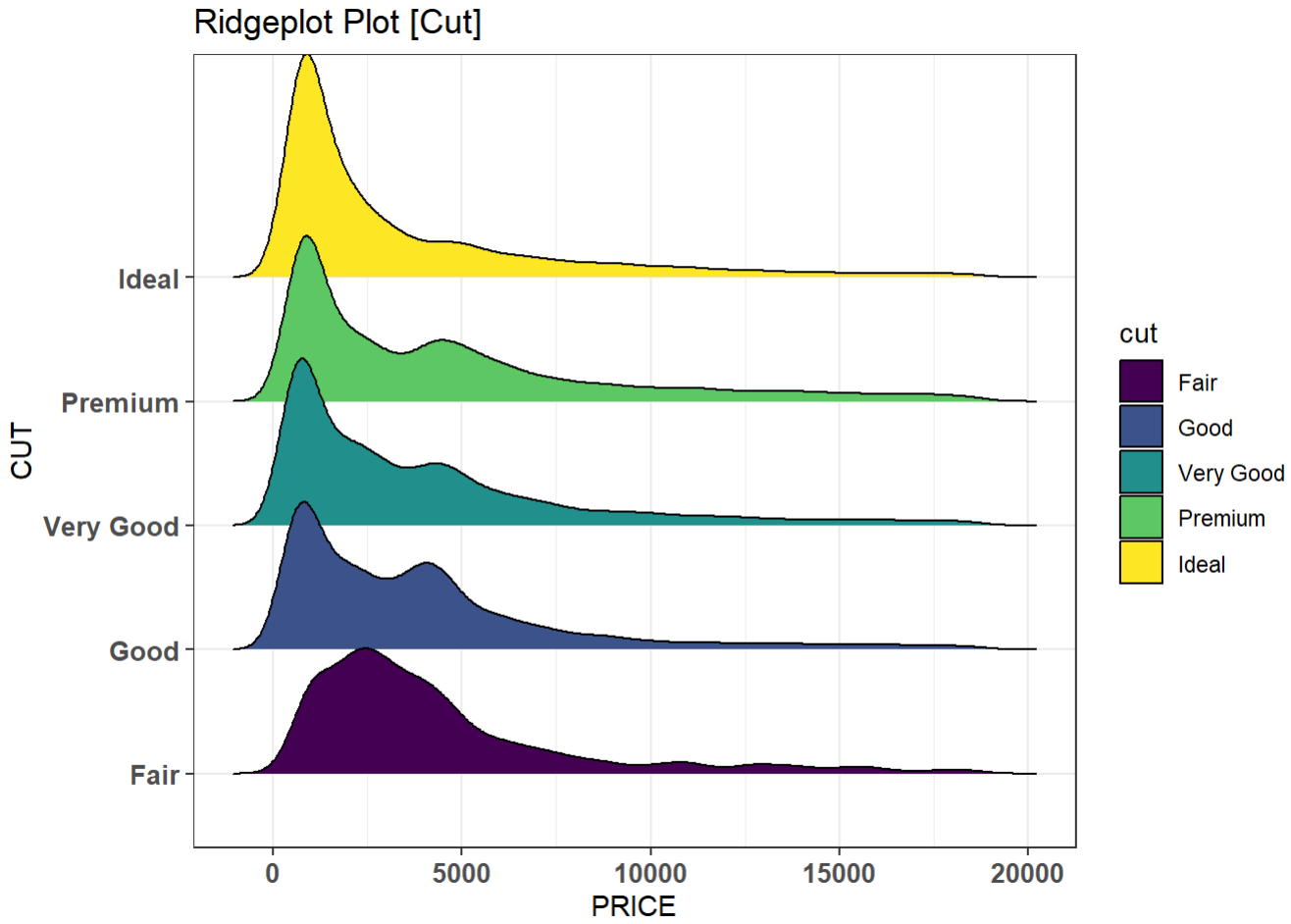
dataset : diamonds

```
ggplot(data_diamonds) +
  geom_density_ridges(aes(x = price , y = cut, fill = cut)) +

  ggtitle("Ridgeplot Plot [Cut]") +
  xlab("PRICE") +
  ylab("CUT") +
  theme_bw() +

  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10))
```

```
## Picking joint bandwidth of 458
```



If you like to contact me :

- https://www.linkedin.com/in/aykhaled/ (https://www.linkedin.com/in/aykhaled/)
- https://github.com/aykhaled/ (https://github.com/aykhaled/)
- https://medium.com/@aykhaled (https://medium.com/@aykhaled)
- myds2020@gmail.com (mailto:myds2020@gmail.com)