# Project 6 : EDA Visualization of the Diamond Dataset Using R and R Markdown

Phubordin Phanyosri

2025-07

## Contents

# 1 คำแนะนำ :

ถ้าคุณดูผ่าน Github แนะนำให้กด Ctrl+F เพื่อไปยังหัวข้อที่สนใจ (ที่บอกมาจาก My Portfolio Website)

# 2 Load Library

```
library(knitr) # ใช้สำหรับรันโค้ด R ที่ฝังในเอกสาร Markdown / LaTeX
library(tidyverse) # แพ็คเกจที่รวบรวมเครื่องใช้จัดการข้อมูล และนำเสนอข้อมูล
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

# 3 Explore Data

## 3.1 Data Glimpse

```
diamonds |> glimpse() # โครงสร้างตาราง diamonds กี่แถว กี่คอลัมน์ ประเภทคอลัมน์
```

```
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

## 3.2 Preview Data

ตาราง diamonds เป็นชุดข้อมูลเกี่ยวกับเพชร ที่เป็นตัวช่วยดูราคาเพชรโดยมีคอลัมน์ ต่อไปนี้

| Cols | Description | Type |
|---|---|---|
| carat | น้ำหนักของเพชร มีหน่วยเป็น “กะรัต” (carats) หน่วยวัดน้ำหนักเพชร | Numeric |
| cut | ระดับคุณภาพการเจียระไน (cut quality) เช่น Fair, Good, Very Good, Premium, Ideal (เรียงจากต่ำไปสูง) | Ordered Factor |
| color | สีของเพชร (D ถึง J โดยที่ D คือใสที่สุด) | Ordered Factor |
| clarity | ระดับความใสของเพชร โดยดูจากตำหนิหรือจุดบกพร่อง (I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF เรียงจากตำหนิชัดเจนถึงไม่มีเลย) | Ordered Factor |
| depth | อัตราส่วนระหว่างความลึกของเพชรกับเส้นผ่านศูนย์กลางเฉลี่ย มีหน่วยเป็นเปอร์เซ็นต์ (%): (z / mean(x, y)) * 100 | Numeric |
| table | ความกว้างของโต๊ะเพชร (ส่วนเรียบด้านบนของเพชร) เทียบกับเส้นผ่านศูนย์กลางเฉลี่ย มีหน่วยเป็นเปอร์เซ็นต์ (%) | Numeric |
| price | ราคาของเพชร มีหน่วยเป็นดอลลาร์สหรัฐ | Integer |
| x | ความยาว (length) ของเพชรในหน่วยมิลลิเมตร | Numeric |
| y | ความกว้าง (width) ของเพชรในหน่วยมิลลิเมตร | Numeric |
| z | ความลึก (depth) ของเพชรในหน่วยมิลลิเมตร | Numeric |

- ขยายความหมาย: depth : ( z / mean(x, y)) * 100
  - ค่านี้แสดงถึง ความสมดุลของรูปร่างเพชร:
    - ค่า depth ต่ำเกินไป: เพชรอาจแบนเกินไป
    - ค่า depth สูงเกินไป: เพชรอาจลึกหรือหนาเกินไป
  - ค่าที่เหมาะสมสำหรับ เพชรทรงกลม (round cut) มักอยู่ที่ประมาณ 59-62% เพื่อให้เพชรมีประกายดีที่สุด

```
diamonds |> head() |> kable() # ดู 6 แถวแรก(ไม่รวมหัวตาราง) kable ปรับตารางให้ดูเหมาะสม
```

| carat | cut | color | clarity | depth | table | price | x | y | z |
|---|---|---|---|---|---|---|---|---|---|
| 0.23 | Ideal | E | SI2 | 61.5 | 55 | 326 | 3.95 | 3.98 | 2.43 |
| 0.21 | Premium | E | SI1 | 59.8 | 61 | 326 | 3.89 | 3.84 | 2.31 |
| 0.23 | Good | E | VS1 | 56.9 | 65 | 327 | 4.05 | 4.07 | 2.31 |
| 0.29 | Premium | I | VS2 | 62.4 | 58 | 334 | 4.20 | 4.23 | 2.63 |
| 0.31 | Good | J | SI2 | 63.3 | 58 | 335 | 4.34 | 4.35 | 2.75 |
| 0.24 | Very Good | J | VVS2 | 62.8 | 57 | 336 | 3.94 | 3.96 | 2.48 |

```
diamonds |> tail() |> kable()# ดู 6 แถวท้าย(ไม่รวมหัวตาราง) kable ปรับตารางให้ดูเหมาะสม
```

| carat | cut | color | clarity | depth | table | price | x | y | z |
|---|---|---|---|---|---|---|---|---|---|
| 0.72 | Premium | D | SI1 | 62.7 | 59 | 2757 | 5.69 | 5.73 | 3.58 |
| 0.72 | Ideal | D | SI1 | 60.8 | 57 | 2757 | 5.75 | 5.76 | 3.50 |
| 0.72 | Good | D | SI1 | 63.1 | 55 | 2757 | 5.69 | 5.75 | 3.61 |
| 0.70 | Very Good | D | SI1 | 62.8 | 60 | 2757 | 5.66 | 5.68 | 3.56 |
| 0.86 | Premium | H | SI2 | 61.0 | 58 | 2757 | 6.15 | 6.12 | 3.74 |
| 0.75 | Ideal | D | SI2 | 62.2 | 55 | 2757 | 5.83 | 5.87 | 3.64 |

## 3.3 Duplicate Rows

```
diamonds |> distinct() |> count() !=  diamonds |> count() # ตรวจสอบว่ามีแถวซ้ำกันไหม TRUE มี, False ไม่มี
```

```
##       n
## [1,] TRUE
```

### 3.3.1 View Only Duplicate Rows

```
# ดูเฉพาะแถวที่ซ้ำ
diamonds |>
 group_by(across(everything())) |> # group โดยใช้ทุกคอลัมน์
 filter(n() > 1) # เลือกเฉพาะกลุ่มที่มีแถวมากกว่า 1  นั่นคือแถวที่ซ้ำกัน
```

```
## # A tibble: 289 x 10
## # Groups:   carat, cut, color, clarity, depth, table, price, x, y, z [143]
##   carat cut   color clarity depth table price     x     y     z
##   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.79 Ideal G    SI1     62.3    57  2898  5.9   5.85  3.66
## 2  0.79 Ideal G    SI1     62.3    57  2898  5.9   5.85  3.66
## 3  0.79 Ideal G    SI1     62.3    57  2898  5.9   5.85  3.66
## 4  0.79 Ideal G    SI1     62.3    57  2898  5.9   5.85  3.66
## 5  0.79 Ideal G    SI1     62.3    57  2898  5.9   5.85  3.66
## 6  1.52 Good  E    I1      57.3    58  3105  7.53  7.42  4.28
## 7  1.52 Good  E    I1      57.3    58  3105  7.53  7.42  4.28
## 8  1    Fair  E    SI2     67      53  3136  6.19  6.13  4.13
## 9  1    Fair  E    SI2     67      53  3136  6.19  6.13  4.13
## 10  1    Fair  F    SI2     65.1    55  3265  6.26  6.23  4.07
## # i 279 more rows
```

### 3.3.2 View Only Unique Rows

```
# ดูเฉพาะแถวที่ไม่ซ้ำ (ไม่รวมแถวซ้ำ)
diamonds |>
 group_by(across(everything())) |> # group โดยใช้ทุกคอลัมน์
 filter(n() == 1) # เลือกเฉพาะแถวที่ไม่ซ้ำกันเท่านั้น
```

```
## # A tibble: 53,651 x 10
## # Groups:   carat, cut, color, clarity, depth, table, price, x, y, z [53,651]
##   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  0.29 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
## 7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
## 8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
## 9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H    VS1      59.4    61   338  4     4.05  2.39
## # i 53,641 more rows
```

### 3.3.3 Identify Duplicates with Counts

```
# ดูเฉพาะแถวที่ซ้ำว่ามีกี่แถว แต่ละแถวซ้ำกันกี่ครั้ง
diamonds |>
 group_by(across(everything())) |>
 tally() |>
 filter(n > 1)  # แสดงเฉพาะแถวที่ซ้ำ
```

```
## # A tibble: 143 x 11
## # Groups:   carat, cut, color, clarity, depth, table, price, x, y [143]
##   carat cut       color clarity depth table price     x     y     z     n
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>
## 1  0.3  Good      J     VS1      63.4    57   394  4.23  4.26  2.69     2
## 2  0.3  Very Good G     VS2      63      55   526  4.29  4.31  2.71     2
## 3  0.3  Very Good J     VS1      63.4    57   506  4.26  4.23  2.69     2
## 4  0.3  Premium   D     SI1      62.2    58   709  4.31  4.28  2.67     2
## 5  0.3  Ideal     G     VS2      63      55   675  4.31  4.29  2.71     2
## 6  0.3  Ideal     G     IF       62.1    55   863  4.32  4.35  2.69     2
## 7  0.3  Ideal     H     SI1      62.2    57   450  4.26  4.29  2.66     2
## 8  0.3  Ideal     H     SI1      62.2    57   450  4.27  4.28  2.66     2
## 9  0.31 Good      D     SI1      63.5    56   571  4.29  4.31  2.73     2
## 10  0.31 Very Good D    SI1      63.5    56   732  4.31  4.29  2.73     2
## # i 133 more rows
```

### 3.3.4 Remove Duplicate Rows (Keeping First Instance)

```
diamonds |> distinct() # ลบเฉพาะแถวซ้ำ แต่ยังคงไว้แค่แถวแรกไว้
```

```
## # A tibble: 53,794 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  0.29 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
## 7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
## 8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
## 9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H    VS1      59.4    61   338  4     4.05  2.39
## # i 53,784 more rows
```

## 3.4 Explore NA

### 3.4.1 Rows with at least one NA in any column

```
diamonds |> filter(if_any(everything(), is.na)) # เลือกแถวที่มีค่า NA อย่างน้อย 1 คอลัมน์
```

```
## # A tibble: 0 x 10
## # i 10 variables: carat <dbl>, cut <ord>, color <ord>, clarity <ord>,
## #   depth <dbl>, table <dbl>, price <int>, x <dbl>, y <dbl>, z <dbl>
```

### 3.4.2 Columns with at least one NA in any row

```
diamonds |> select(where(~ any(is.na(.)))) # เลือกคอลัมน์ที่มี NA อย่างน้อย 1 แถว
```

```
## # A tibble: 53,940 x 0
```

# 4 Clensing Diamonds Data

```
# ลบแถวซ้ำ
prep_diamonds <- diamonds |> distinct()
prep_diamonds
```

```
## # A tibble: 53,794 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  0.29 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
## 7  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
## 8  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
## 9  0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10  0.23 Very Good H    VS1      59.4    61   338  4     4.05  2.39
## # i 53,784 more rows
```

## 4.1 Check Duplicate Rows

```
# ตรวจสอบว่ามีแถวซ้ำกันไหม TRUE มี, False ไม่มี
prep_diamonds |> distinct() |> count() != prep_diamonds |> count()
```

```
##         n
## [1,] FALSE
```

# 5 Visualize Diamonds Data

```
# สร้างธีมที่เราต้องการเป็นมาตรฐานในการแสดงผล
custom_theme <- theme_linedraw() +
 theme(
  # Darker plot title
  plot.title = element_text( # ชื่อหัวข้อของกราฟ
   face = "bold",        # เป็นตัวหนา
   size = rel(1.1),      # 110% ของขนาดฟอนต์ปกติ เทียบกับค่า default ของธีม
   color = "black",      # ฟอนต์สีดำ
   margin = margin(b = 10) # ระยะห่างจากด้านล่าง title กับ subtitle หรือกราฟ 10 pt
  ),
  # Darker subtitle
  plot.subtitle = element_text( # ชื่อหัวข้อย่อยลงมา
   size = rel(0.9),      # 90% ของขนาดฟอนต์ปกติ เทียบกับค่า default ของธีม
   color = "#2F4F4F",    # Dark slate gray
   margin = margin(b = 15) # ระยะห่างจากด้านล่าง title กับ subtitle หรือกราฟ 15 pt
  ),
  # Darker caption
  plot.caption = element_text( # ชื่อแหล่งที่มามุมล่างขวา
   size = 8,             # ขนาด 8 pt
   color = "#4A4A4A",    # Darker gray
   margin = margin(t = 10) # ระยะห่างจากด้านบน title กับ subtitle หรือกราฟ 15 pt
  ),
  # Darker axis titles
  axis.title = element_text( # ชื่อแกน
   size = 11,        # ขนาด 11 pt
   color = "#1C1C1C",  # Very dark gray
   face = "bold"      # ตัวหนา
  ),
  # Darker axis text
  axis.text = element_text( # เลขบนแกน
   size = 9,         # ขนาด 9 pt
   color = "#2F4F4F"  # Dark slate gray
  ),
  # Customize facet labels
  strip.text = element_text( # ชื่อแกน x, y บน facet
   size = 10,        # ขนาด 10 pt
   color = "white",   # สีขาว
   face = "bold"      # ตัวหนา
  ),
  strip.background = element_rect( # กำหนดพื้นหลังชื่อแกน facet
   fill = "#2F4F4F",  # กำหนดสีพื้นหลังของชื่อแกน facet
   color = "black"    # สีขอบพื้นหลังของชื่อแกน facet
  ),
  # Darker panel elements
  panel.grid.major = element_line( # เส้นกริดหลัก (เส้นใหญ่) บนพื้นหลังกราฟ (แนวนอน-แนวตั้ง)
   color = "#BEBEBE",  # Medium gray
   size = 0.3         # ความบาง-หนา ยิ่งเข้าใกล้ 1 ยิ่งหนามาก
  ),
  panel.grid.minor = element_line( # เส้นกริดรอง (เส้นเล็ก) บนพื้นหลังกราฟ (แนวนอน-แนวตั้ง)
   color = "#D3D3D3",  # Light gray
```

```
    size = 0.2        # ความบาง-หนา ยิ่งเข้าใกล้ 1 ยิ่งหนามาก
  ),
  panel.border = element_rect( # เส้นขอบภายในแกนวาดกราฟเท่านั้น
    color = "black",   # สีดำ
    size = 0.8        # ความบาง-หนา ยิ่งเข้าใกล้ 1 ยิ่งหนามาก
  ),
  # Add margin around the plot
  # กำหนดให้ element ทั้งหมดภายใน canvas top, right, bottom, left ห่าง 15 pt
  plot.margin = margin(15, 15, 15, 15),
  panel.background = element_rect( # พื้นหลังภายในแกนเท่านั้น
    fill = "white"  # เป็นสีขาว
  )
)
```

## 5.1 Distribution of price groups of diamonds by Density, Table.

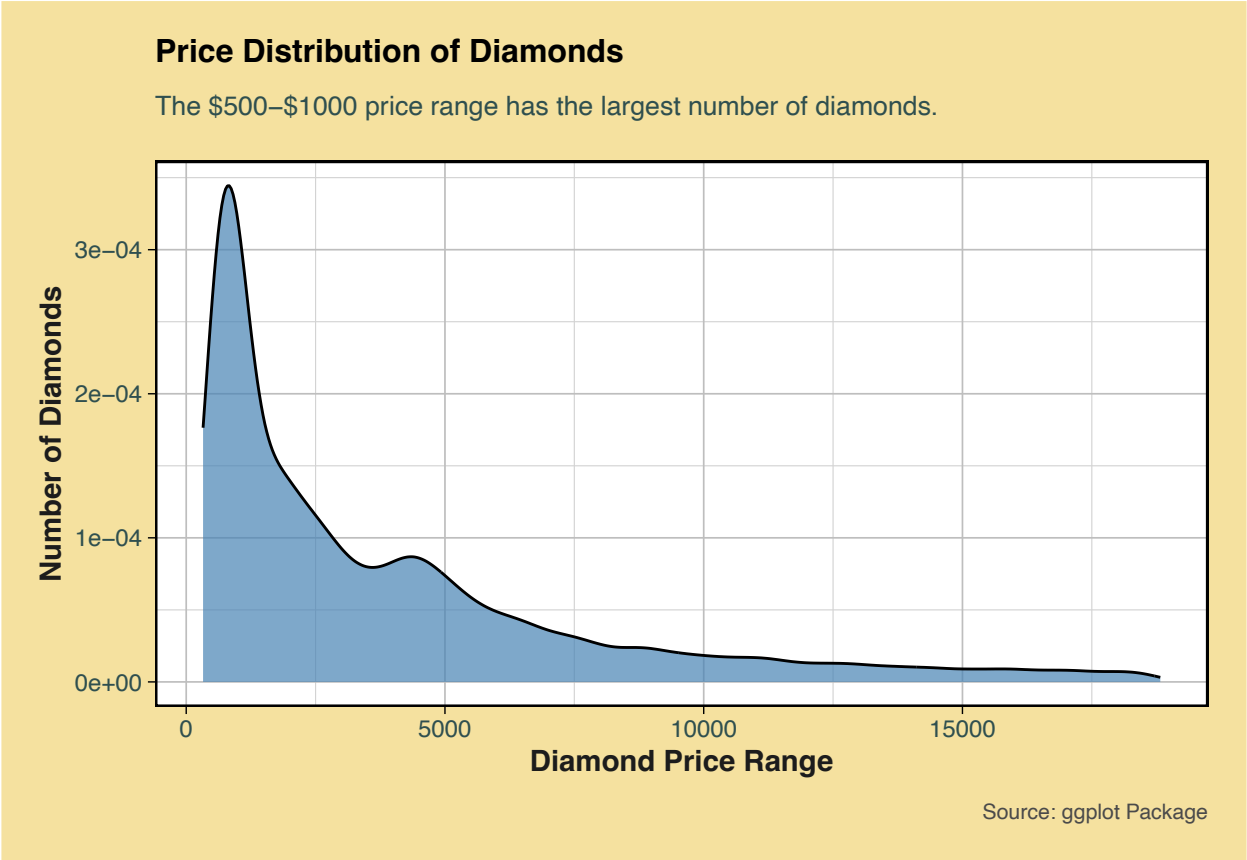
### 5.1.1 Table: Price Ranges of Diamonds

```
# สร้างตารางความถี่ของราคาเพชร
prep_diamonds %>%
 mutate( # สร้างคอลัมน์ใหม่
  price_range = cut( # ชื่อคอลัมน์ใหม่ price_range โดยการแบ่งช่วงราคา
   # นำ price มาสร้างจุดแบ่ง(break) ทีละ 500 เช่น 0, 500, 1000, ..,  18000 18500(max)
   price, breaks = seq(0, max(price), by = 500),
   labels = paste(
    seq(0, max(price) - 500, by = 500),
    seq(500, max(price), by = 500), sep = "-"
   ) # สร้าง labels เพื่อบอกว่าคอลัมน์ price อยู่ช่วงราคาเพชรไหนด้วย ขอบบน-ขอบล่าง
  )
 ) %>%
 count(price_range) %>% # นับเพชรในแต่ละช่วงราคา
 head(10) %>% # เลือก 10 แถวบนสุด
 # ทำเป็นตารางที่ดูเหมาะสมขึ้น พร้อมเปลี่ยนชื่อคอลัมน์ใหม่
 kable(col.names = c("Price Range ($)", "Number of Diamonds"))
```

| Price Range ($) | Number of Diamonds |
|---|---|
| 0-500 | 1745 |
| 500-1000 | 12725 |
| 1000-1500 | 5463 |
| 1500-2000 | 4194 |
| 2000-2500 | 3332 |
| 2500-3000 | 2789 |
| 3000-3500 | 2159 |
| 3500-4000 | 2056 |
| 4000-4500 | 2470 |
| 4500-5000 | 2183 |

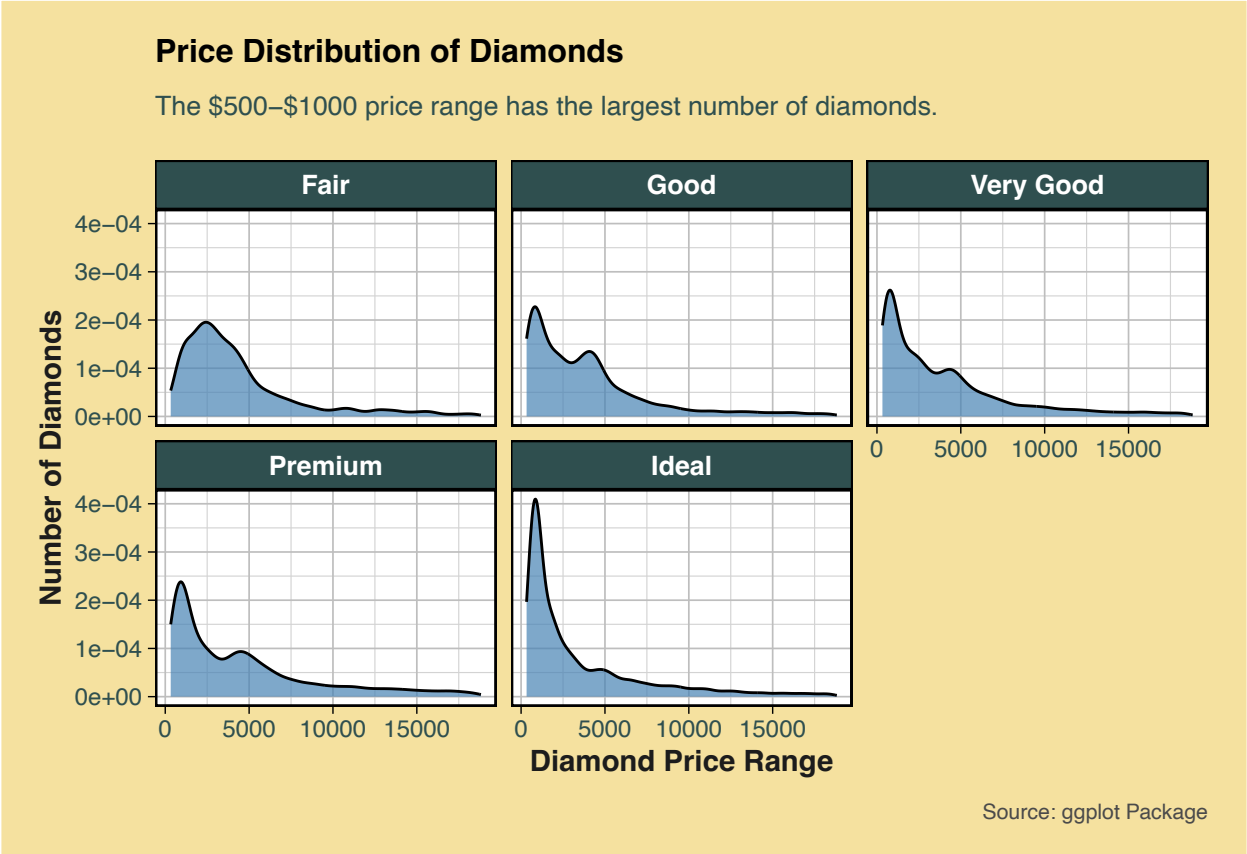
### 5.1.2 Density: Price Ranges of Diamonds

```
# Density Chart ดูการกระจายตัวของช่วงราคาเพชร
prep_diamonds %>%
 ggplot(aes(price)) + # กำหนดแกน x เป็น price
 geom_density(fill = "#4682B4", alpha = 0.7) + # ลงสีพื้นที่ใต้ดราฟเป็นสีออกน้ำเงิน ค่อนไปทางเข้มๆ
 custom_theme + # ปรับแต่งธีมของ Chart ตามตัวอย่างแรก
 theme(plot.background = element_rect(fill = "#f5e19f")) + # เพิ่มสีของ canvas เป็นสีครีม
 labs(title = "Price Distribution of Diamonds",# ชื่อของ Chart การกระจายตัวราคาเพชร
    # กำหนดหัวข้อรองลงมาว่าจำนวนเพชรมากสุดอยู่ที่ช่วงไหน
    subtitle = "The $500-$1000 price range has the largest number of diamonds.",
    caption = "Source: ggplot Package", # เพิ่ม caption มุมล่างขวาของ canvas
    x = "Diamond Price Range",  # ชื่อแกน x
    y = "Number of Diamonds")  # ชื่อแกน y
```



## 5.2 Sub-Diagram of Diamond Price Groups, Divided by cut.

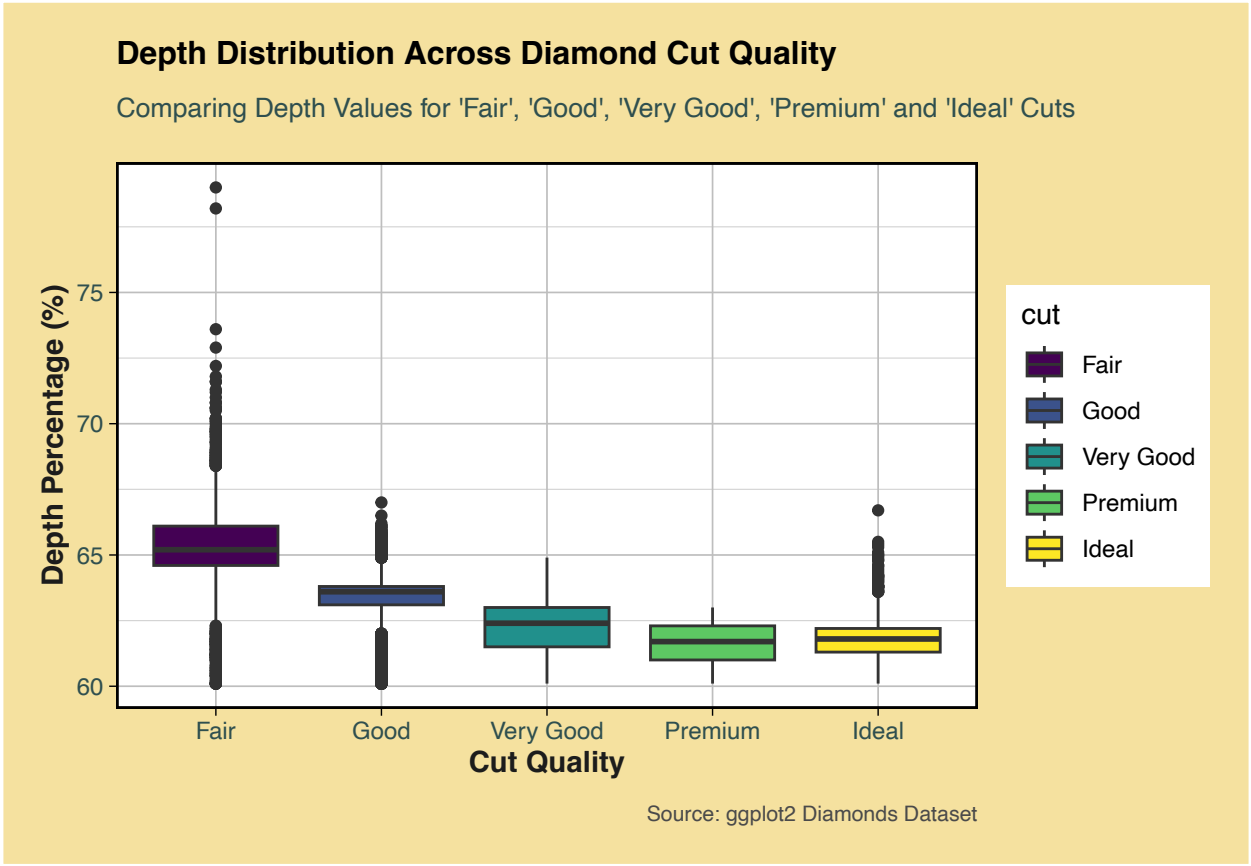
```
# Density Chart ดูการกระจายตัวของช่วงราคาเพชร แบ่งตามเกรด fair, good, very good, premium, Ideal
prep_diamonds %>%
 ggplot(aes(price)) + # กำหนดแกน x เป็น price
 geom_density(fill = "#4682B4", alpha = 0.7) + # ลงสีพื้นที่ใต้ดราฟเป็นสีออกน้ำเงิน ค่อนไปทางเข้มๆ
 custom_theme + # ปรับแต่งธีมของ Chart ตามตัวอย่างอรก
 theme(plot.background = element_rect(fill = "#f5e19f")) + # เพิ่มสีของ canvas เป็นสีครีม
 labs(title = "Price Distribution of Diamonds",# ชื่อของ Chart การกระจายตัวราคาเพชร
    # กำหนดหัวข้อรองลงมาว่าจำนวนเพชรมากสุดอยู่ที่ช่วงไหน
    subtitle = "The $500-$1000 price range has the largest number of diamonds.",
    caption = "Source: ggplot Package", # เพิ่ม caption มุมล่างขวาของ canvas
    x = "Diamond Price Range",  # ชื่อแกน x
    y = "Number of Diamonds") +  # ชื่อแกน y
 facet_wrap(~ cut)  # แบ่งตามเกรด
```



## 5.3 Depth Distribution Across Diamond Cut Quality.

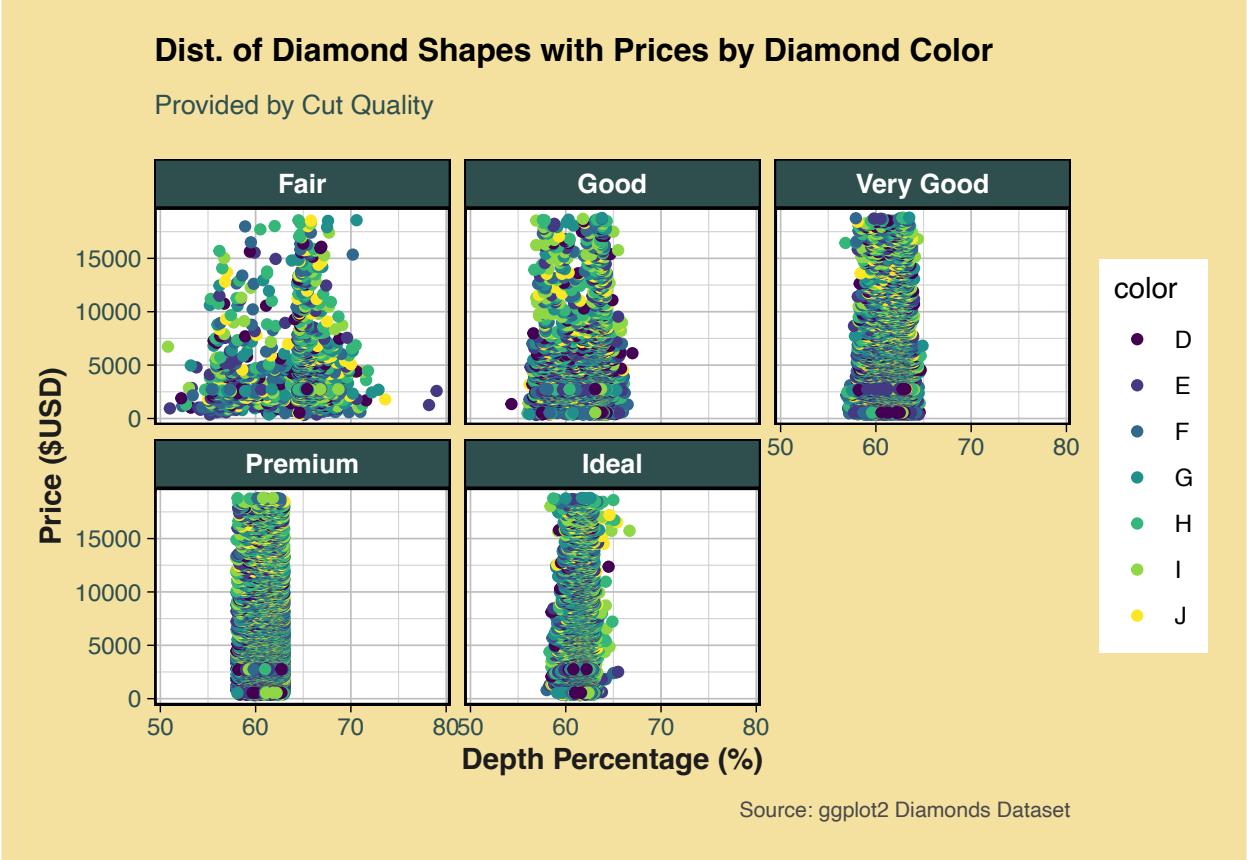
### 5.3.1 Boxplot: Comparing Depth Values for Cut Quality

```
# Box plot ดูการกระจายตัวของ รูปร่างของเพชรที่มีค่ามากกว่า 60 %
prep_diamonds %>%
 # กรอง สัดส่วนของความลึก เทียบกับเส้นผ่านศูนย์กลางของเพชรมากกว่า 60% ขึ้นไป
 filter(depth > 60) %>%
 # กำหนดแกน x, y เป็น cut, depth ตามลำดับ เติมสีแยกกันตาม cut
 ggplot(aes(cut, depth, fill = cut)) +
 geom_boxplot() + # กำหนด Box Plot
 custom_theme +  # ปรับแต่งธีมของ Chart ตามตัวอย่างแรก
 theme(plot.background = element_rect(fill = "#f5e19f")) + # เพิ่มสีของ canvas เป็นสีครีม
 labs(  # ชื่อของ Chart การกระจายตัวราคาเพชร
  # กำหนดชื่อหัวข้อใหญ่
  title = "Depth Distribution Across Diamond Cut Quality",
  # กำหนดหัวข้อรองลงมา
  subtitle = "Comparing Depth Values for 'Fair', 'Good', 'Very Good', 'Premium' and 'Ideal' Cuts",
  caption = "Source: ggplot2 Diamonds Dataset",  # เพิ่ม caption มุมล่างขวาของ canvas
  x = "Cut Quality",  # ชื่อแกน x
  y = "Depth Percentage (%)"  # ชื่อแกน y
 )
```



### 5.3.2 Scatter: Dist. of Diamond Shapes with Prices by Diamond Color

```
# Scatter plot ดูการกระจายตัวของ รูปร่างของเพชรที่มีค่ามากกว่า 50%
prep_diamonds %>%
 # กรอง สัดส่วนของความลึก เทียบกับเส้นผ่านศูนย์กลางของเพชรมากกว่า 50% ขึ้นไป
 filter(depth > 50) %>%
 # กำหนดแกน x, y เป็น depth, price ตามลำดับ เติมสีแยกกันตาม color
 ggplot(aes(depth, price, color = color)) +
 geom_point() +  # กำหนด Scatter Plot
 facet_wrap(~ cut) +  # แบ่งตามเกรด
 custom_theme +  # ปรับแต่งธีมของ Chart ตามตัวอย่างแรก
 theme(plot.background = element_rect(fill = "#f5e19f")) + # เพิ่มสีของ canvas เป็นสีครีม
 labs(
  # กำหนดชื่อหัวข้อใหญ่
  title = "Dist. of Diamond Shapes with Prices by Diamond Color",
  # กำหนดหัวข้อรองลงมา
  subtitle = "Provided by Cut Quality",
  caption = "Source: ggplot2 Diamonds Dataset",  # เพิ่ม caption มุมล่างขวาของ canvas
  x = "Depth Percentage (%)",  # ชื่อแกน x
  y = "Price ($USD)"  # ชื่อแกน y
 )
```



## 5.4 Correlation Between Carat and Diamond Price.

```
# Scatter Plot vs. Line Chart ดูความสัมพันธ์น้ำหนักเพชรกับราคาเพชร แบ่งตาม clarity, cut
prep_diamonds %>%
 filter(
  clarity %in% c("VS1", "VVS2", "VVS1", "IF"), # clarity ที่มี VS1, VVS2, VVS1, IF
  color %in% c("G", "E", "F"), # color มี G, E, F
  cut %in% c("Ideal", "Premium", "Very Good") #  cut Ideal, Premium, Very Good
 ) %>%
 # กำหนดแกน x, y เป็น carat, price ตามลำดับ เติมสีแยกกันตาม clarity
 ggplot(aes(carat, price, col = clarity)) +
 geom_point() +  # กำหนด Scatter Plot
 geom_smooth(method = "lm", col = "red") + # สร้าง LR สีแดง carat vs. price
 # แยก cut ตามแนวแกน x และแยก color ตามแนวแกน y
 facet_grid(color ~ cut) +
 custom_theme +  # ปรับแต่งธีมของ Chart ตามตัวอย่างแรก
 theme(plot.background = element_rect(fill = "#f5e19f")) + # เพิ่มสีของ canvas เป็นสีครีม
 labs(
  title = "Correlation Between Carat and Diamond Price", # กำหนดชื่อหัวข้อใหญ่
  subtitle = "The overall correlation coefficient is 0.9215913.",# กำหนดหัวข้อรองลงมา
  caption = "Source: ggplot2 Diamonds Dataset",  # เพิ่ม caption มุมล่างขวาของ canvas
  x = "Weigth (Carats)",  # ชื่อแกน x
  y = "Price ($USD)"  # ชื่อแกน y
 )
```