Data Visualization Assignment

Prices of over 50,000 round cut diamonds

Load Library

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(patchwork)
```

Load dataset

```
data(diamonds)
```

View dataset

```
head(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  0.29 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
```

Glimpse Dataset

```
glimpse(diamonds)
```

```
## 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.~
```

Summarize dataset

```
summary(diamonds)
```

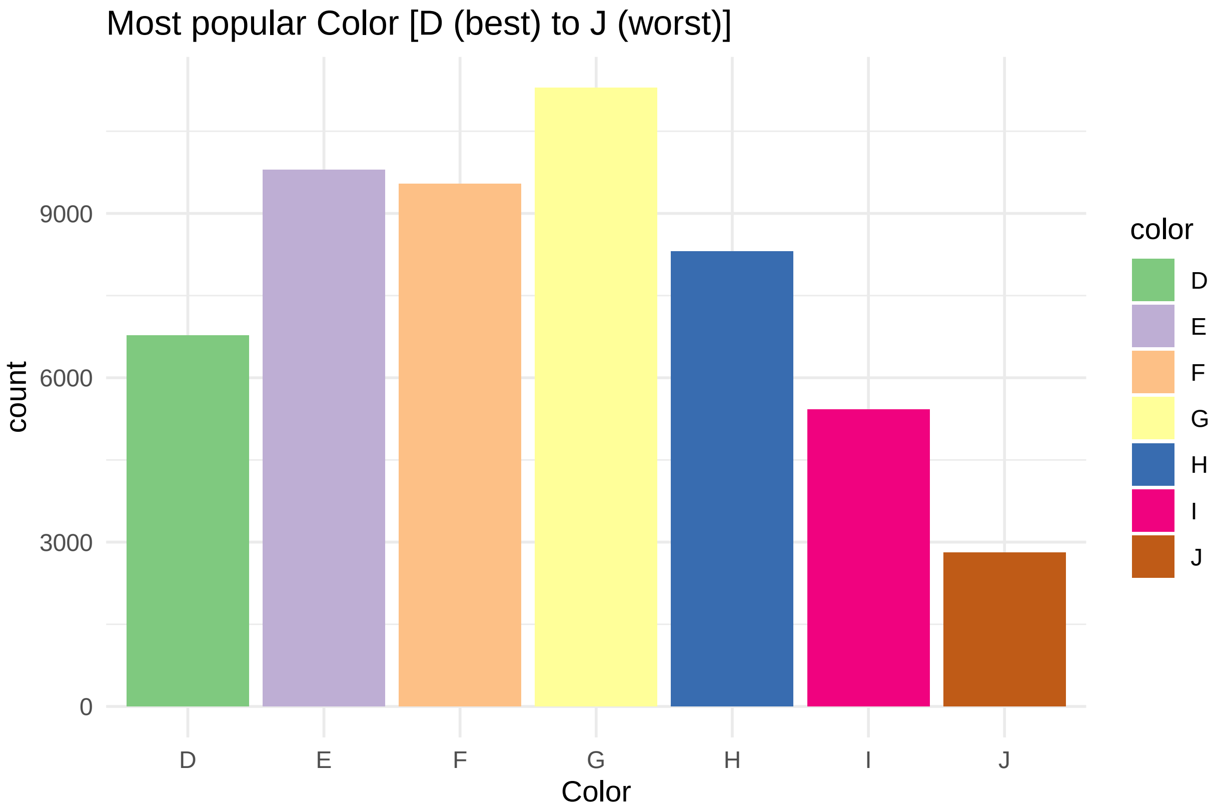
```
##      carat               cut        color        clarity          depth
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065   Min.   :43.00
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258   1st Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :61.80
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171   Mean   :61.75
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.   :79.00
##                                     J: 2808   (Other): 2531
##      table           price             x                y
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900
##
##        z
##  Min.   : 0.000
##  1st Qu.: 2.910
##  Median : 3.530
##  Mean   : 3.539
##  3rd Qu.: 4.040
##  Max.   :31.800
##
```

Sample data

```
set.seed(33)
sample_diamonds <- sample_n(diamonds, 1000)
```

Chart 1: Most popular Color [D (best) to J (worst)]

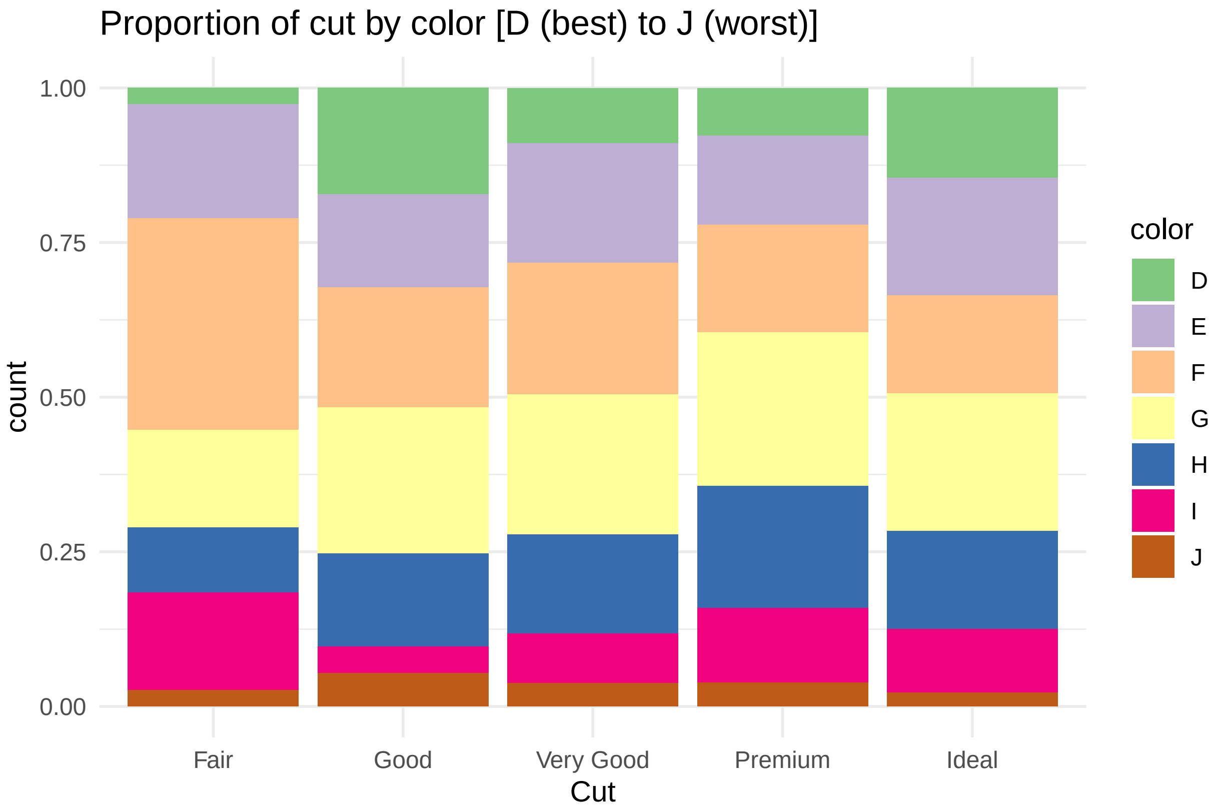
```
diamonds %>%
  ggplot(mapping = aes(x = color, fill = color)) +
  geom_bar() +
  labs(
    title = "Most popular Color [D (best) to J (worst)]",
    x = "Color",
    caption = " Source: ggplot package"
  ) +
  theme_minimal() +
  scale_fill_brewer(type = "qual", palette = "Accent")
```



Source: ggplot package

Chart 2: Proportion of cut by color [D (best) to J (worst)]

```
sample_diamonds %>%
  ggplot(mapping = aes(cut, fill = color)) +
  geom_bar(position = "fill") +
  labs(
    title = "Proportion of cut by color [D (best) to J (worst)]",
    x = "Cut",
    caption = " Source: ggplot package"
  ) +
  theme_minimal() +
  scale_fill_brewer(type = "qual", palette = "Accent")
```



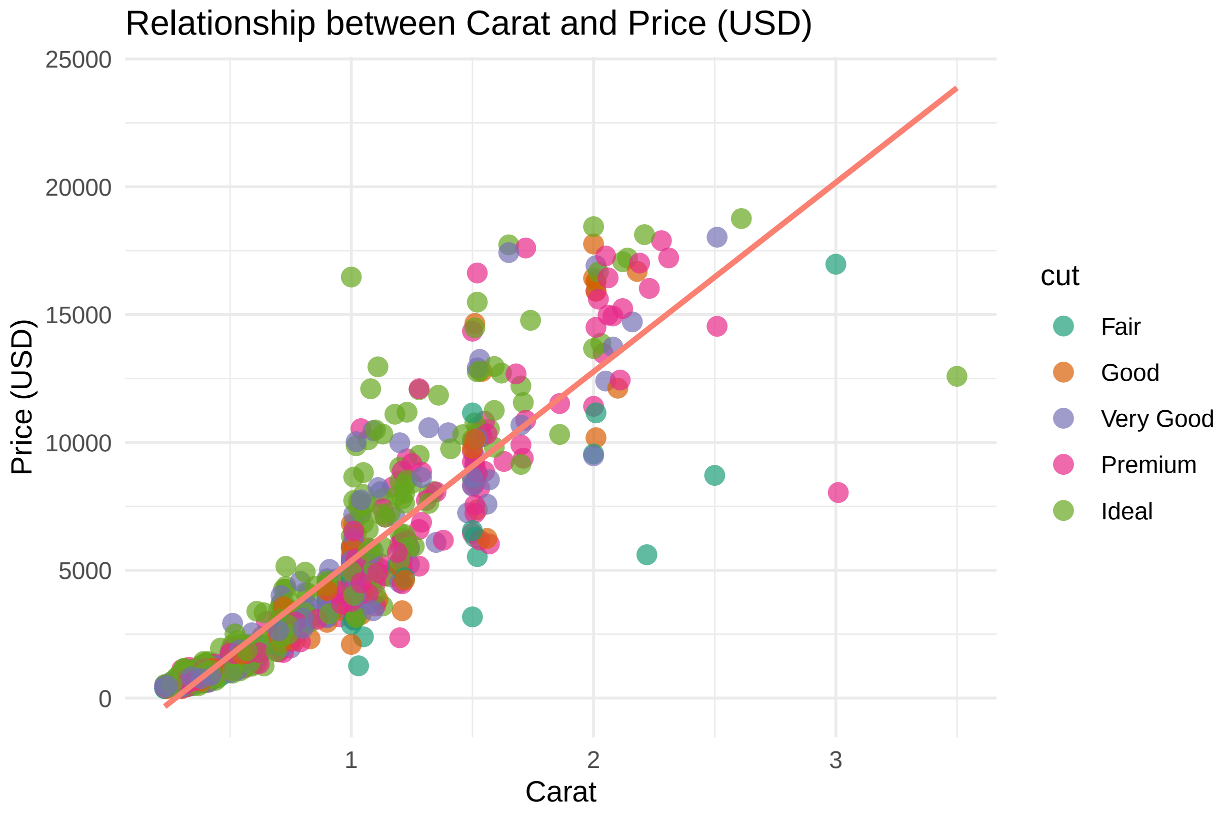
Proportion of cut by color [D (best) to J (worst)]

Source: ggplot package

Chart 3: Relationship between Carat and Price (USD)

```
sample_diamonds %>%
  ggplot(mapping = aes(carat, price, color = cut)) +
  geom_point(size = 3, alpha = .7) +
  geom_smooth(method = "lm", color = "salmon", se = F) +
  labs(
    title = "Relationship between Carat and Price (USD)",
    x = "Carat",
    y = "Price (USD)",
    caption = " Source: ggplot package"
  ) +
  theme_minimal() +
  scale_color_brewer(type = "qual", palette = "Dark2")
```

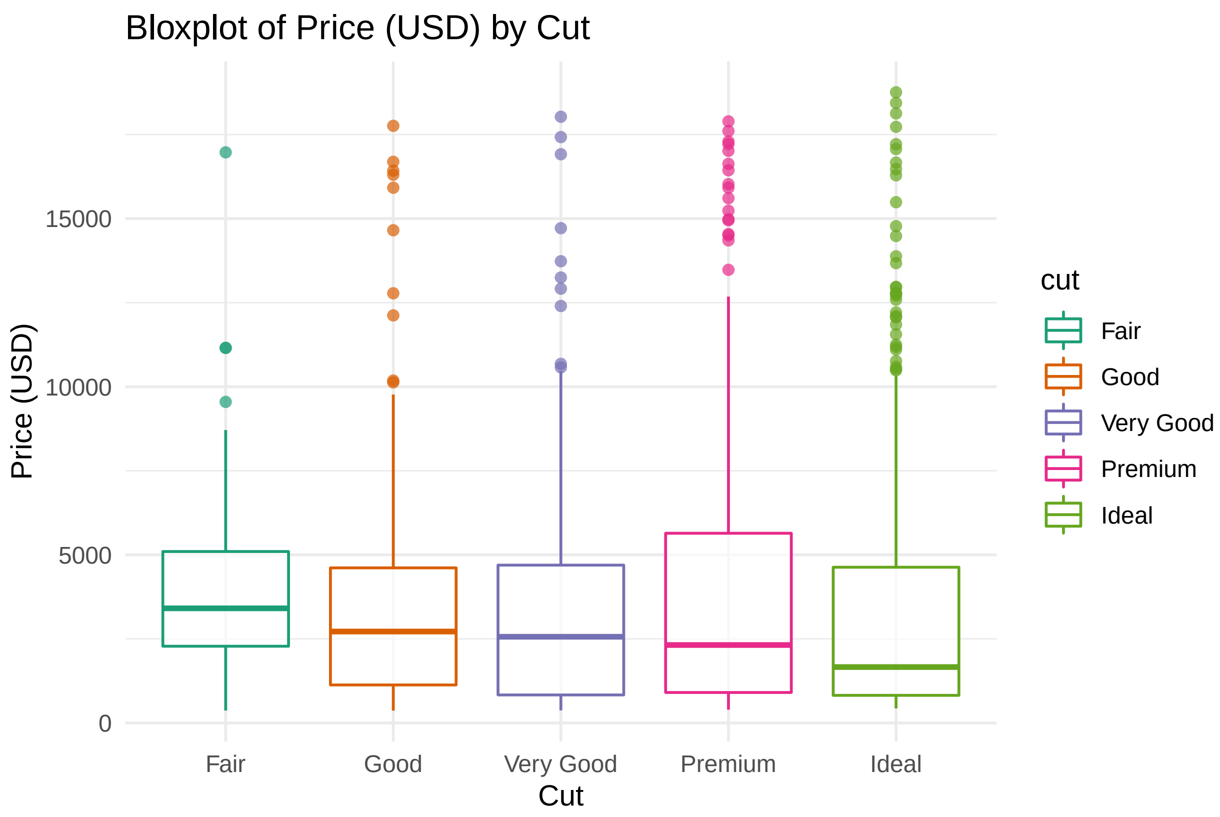
```
## `geom_smooth()` using formula 'y ~ x'
```



Source: ggplot package

Chart 4: Bloxplot of Price (USD) by Cut

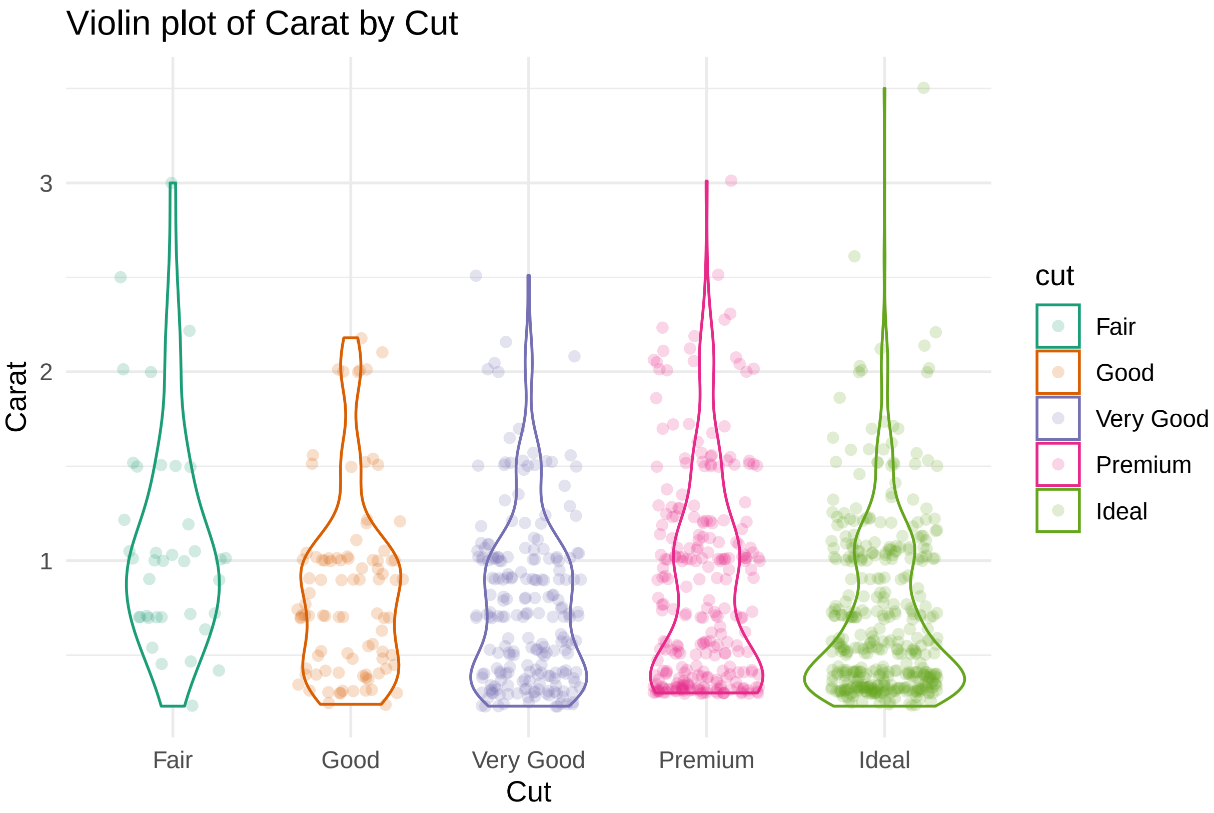
```
sample_diamonds %>%
  ggplot(mapping = aes(cut, price, color = cut)) +
  geom_boxplot(alpha = .7) +
  labs(
    title = "Bloxplot of Price (USD) by Cut",
    x = "Cut",
    y = "Price (USD)",
    caption = " Source: ggplot package"
  ) +
  theme_minimal() +
  scale_color_brewer(type = "qual", palette = "Dark2")
```



Source: ggplot package

Chart 5: Violin plot of Carat by Cut

```
sample_diamonds %>%
  ggplot(mapping = aes(cut, carat, color = cut)) +
  geom_violin() +
  geom_jitter(width = .3, alpha = .2) +
  labs(
    title = "Violin plot of Carat by Cut",
    x = "Cut",
    y = "Carat",
    caption = " Source: ggplot package"
  ) +
  theme_minimal() +
  scale_color_brewer(type = "qual", palette = "Dark2")
```



Source: ggplot package