Data Visualization (HW)

Kane.P

2022-11-02

I.CO2 emission data set from kaggle

Install packages and Import library

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(lubridate)
```

Data preparation

1.Import Data

```
co2e <- read.csv("CO2_emission.csv")
co2e <- tibble(co2e)
wpop <- read.csv("world_population.csv")
wpop <- tibble(wpop)
```

2.Overview Data

```
head(co2e,5)
```

```
## # A tibble: 5 x 35
##   Country~1 count~2 Region Indic~3  X1990  X1991   X1992   X1993   X1994   X1995
##   <chr>     <chr>   <chr>  <chr>    <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Aruba     ABW     Latin~ CO2 em~ NA     NA      NA      NA      NA      NA
## 2 Afghanis~ AFG     South~ CO2 em~  0.192  0.168  0.0960  0.0847  0.0755  0.0685
## 3 Angola    AGO     Sub-S~ CO2 em~  0.554  0.545  0.544   0.709   0.837   0.912
## 4 Albania   ALB     Europ~ CO2 em~  1.82   1.24   0.684   0.638   0.645   0.605
## 5 Andorra   AND     Europ~ CO2 em~  7.52   7.24   6.96    6.72    6.54    6.73
## # ... with 25 more variables: X1996 <dbl>, X1997 <dbl>, X1998 <dbl>,
## #   X1999 <dbl>, X2000 <dbl>, X2001 <dbl>, X2002 <dbl>, X2003 <dbl>,
## #   X2004 <dbl>, X2005 <dbl>, X2006 <dbl>, X2007 <dbl>, X2008 <dbl>,
## #   X2009 <dbl>, X2010 <dbl>, X2011 <dbl>, X2012 <dbl>, X2013 <dbl>,
## #   X2014 <dbl>, X2015 <dbl>, X2016 <dbl>, X2017 <dbl>, X2018 <dbl>,
## #   X2019 <dbl>, X2019.1 <dbl>, and abbreviated variable names 1: Country.Name,
## #   2: country_code, 3: Indicator.Name
```

3. transform from wide to long format

```
co2_l <- co2e %>%
  select(1:3,5:34) %>%
  gather(X1990:X2019,
         key = "XYear",
         value = "MT_per_cap") %>%
  mutate(Year = as.character(str_extract_all(XYear,"\\d+")))
```

4. change data type

```
co2_l <- co2_l %>%
  mutate(Year = year(as.Date(co2_l$Year,format = "%Y")))%>%
  select(-XYear)
head(co2_l,5)
```

```
## # A tibble: 5 x 5
##   Country.Name country_code Region                    MT_per_cap  Year
##   <chr>        <chr>        <chr>                          <dbl> <dbl>
## 1 Aruba        ABW          Latin America & Caribbean     NA      1990
## 2 Afghanistan  AFG          South Asia                     0.192  1990
## 3 Angola       AGO          Sub-Saharan Africa             0.554  1990
## 4 Albania      ALB          Europe & Central Asia          1.82   1990
## 5 Andorra      AND          Europe & Central Asia          7.52   1990
```

5. Edit country name

```
co2_l[co2_l == "Russian Federation"] <- "Russia"
co2_l[co2_l == "Iran, Islamic Rep."] <- "Iran"
co2_l[co2_l == "Venezuela, RB"] <- "Venezuela"
co2_l[co2_l == "Egypt, Arab Rep."] <- "Egypt"
co2_l[co2_l == "Yemen, Rep."] <- "Yemen"
co2_l[co2_l == "Syrian Arab Republic"] <- "Syria"
co2_l[co2_l == "Slovak Republic"] <- "Slovakia"
co2_l[co2_l == "Lao PDR"] <- "Laos"
co2_l[co2_l == "Korea, Rep."] <- "South Korea"
co2_l[co2_l == "Korea, Dem. People's Rep."] <- "North Korea"
```

Data Visualization

Chart1: CO2 emission metric tons per capita by country in 2019

```
co2e_2019 <- co2_l %>%
  filter(Year == 2019)
```

```
world_map <- map_data("world")
world_map[world_map == "USA"] <- "United States"
world_map[world_map == "Republic of Congo"] <- "Congo, Rep."
world_map[world_map == "Democratic Republic of the Congo"] <- "Congo, Dem. Rep."
world_map[world_map == "Turkey"] <- "Turkiye"
## Left join
co2e_map <- left_join(world_map, co2e_2019, by = c("region"="Country.Name"))
ggplot(co2e_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = MT_per_cap), color = "dark grey")+
```

```
  scale_fill_gradient(low = "light yellow", high = "red", na.value = NA)+
  theme_minimal()+
  theme(axis.text = element_text(size = 30),axis.title = element_text(size = 40),
        plot.title= element_text(size = 45),legend.text = element_text(size = 20),
        legend.title = element_text(size = 25),plot.caption = element_text(size = 25))+
  labs(title = "CO2 emission metric tons per capita by country in 2019",
       x = NULL, y = NULL,
       caption = "Source: CO2 Emissions Around the World from kaggle")
```

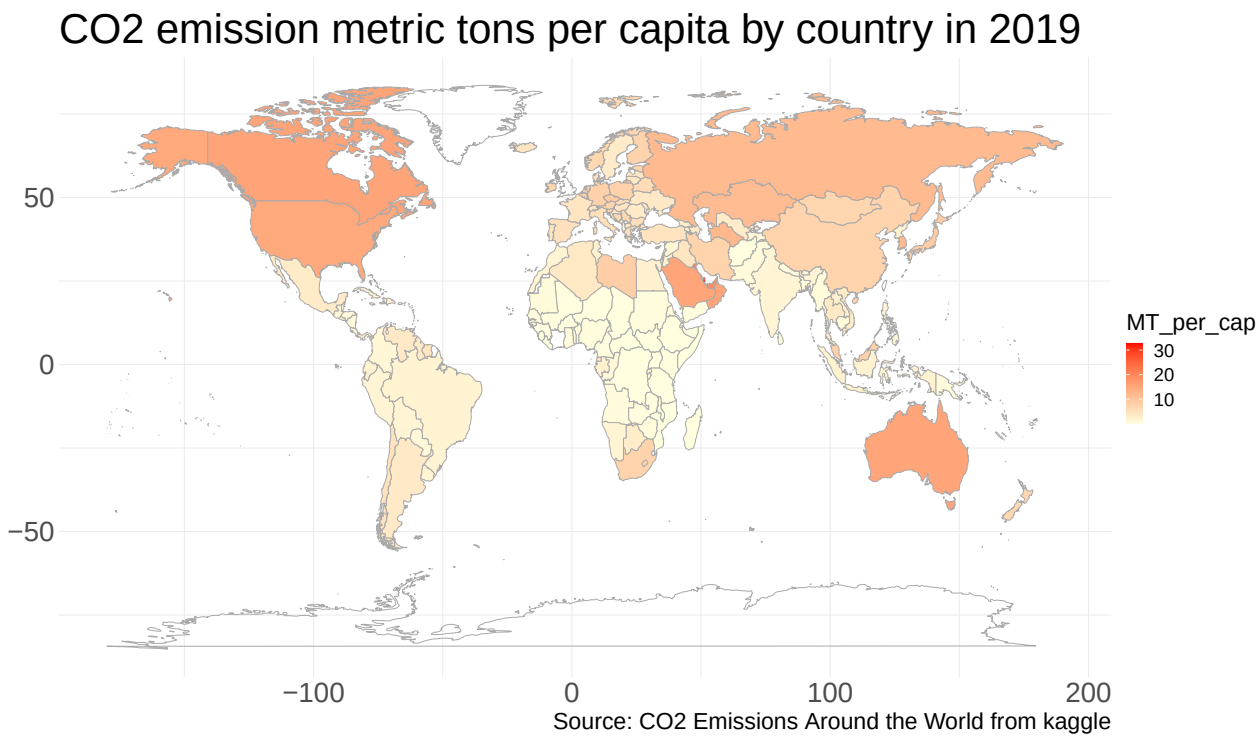


Chart2: CO2 emission metric tons by country in 2019

```
wpop2020 <- wpop %>%
  select(2,3,5,7)
pop_map <- left_join(co2e_2019,wpop2020, by = c("country_code"="CCA3"))
pop_map <- pop_map %>%
  mutate(total_co2 = (MT_per_cap*X2020.Population*1000)/(10**6))
#leftjoin
pop_map <- left_join(pop_map, world_map, by = c("Country.Name"="region"))
#Visualization
ggplot(pop_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = total_co2), color = "dark grey")+
  scale_fill_viridis_c(option = "C", na.value = NA)+
  theme_minimal()+
  theme(axis.text = element_text(size = 30),axis.title = element_text(size = 40),
        plot.title= element_text(size = 45),legend.text = element_text(size = 20),
        legend.title = element_text(size = 25),plot.caption = element_text(size = 25))+
  labs(title = "CO2 emission (metric tons) by country in 2019",
       x = NULL, y = NULL,
        caption = "Source: CO2 Emissions Around the World from kaggle
       (https://www.kaggle.com/datasets/koustavghosh149/co2-emission-around-the-world)")
```

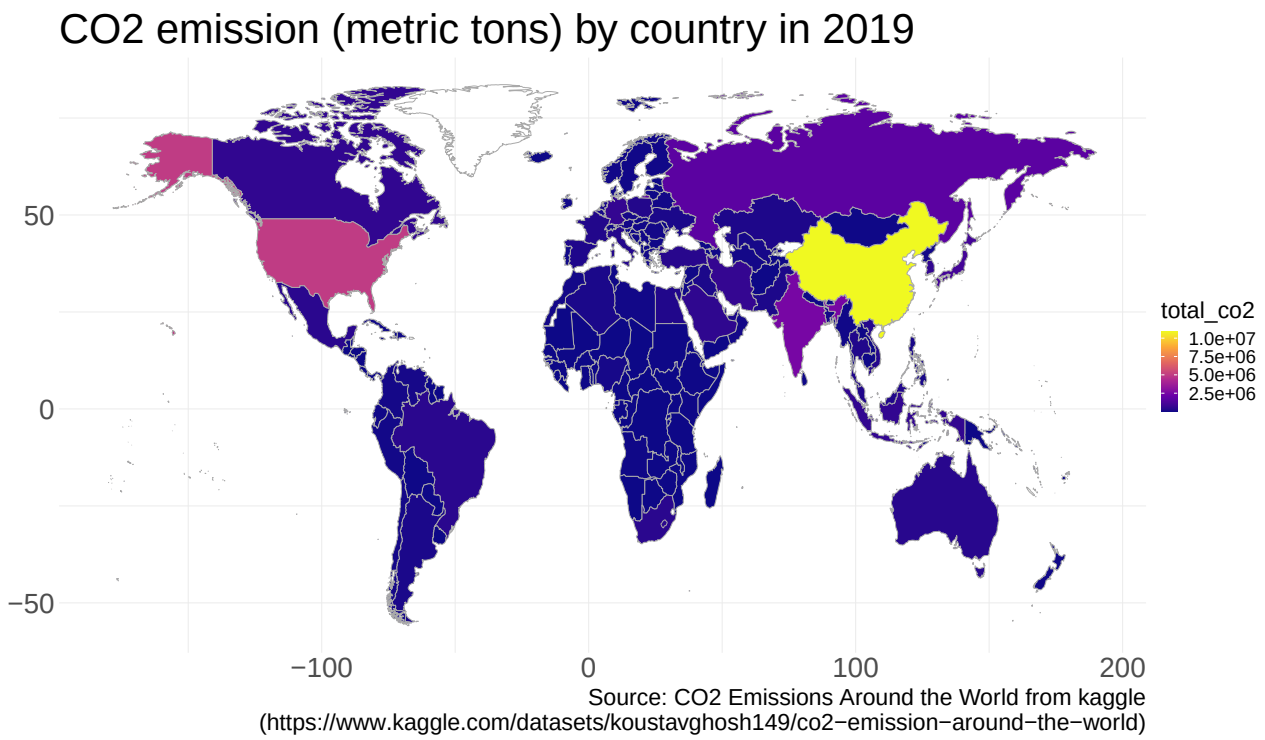


Chart3: Top 10 Most Polluting Countries Per Capita in 2019

```
co2_l %>%
  filter(Year == 2019) %>%
  arrange(desc(MT_per_cap)) %>%
  head(10) %>%
  ggplot(aes(reorder(Country.Name , -MT_per_cap),MT_per_cap , fill = MT_per_cap))+
  geom_col()+
  geom_text(aes(label=round(MT_per_cap,digits = 2),
                vjust = 2))+
  scale_fill_gradient(low = "light yellow", high = "red", na.value = NA)+
  theme_minimal()+
  theme(axis.text = element_text(size = 20),axis.title = element_text(size = 40),
        plot.title= element_text(size = 45),legend.text = element_text(size = 15),
        legend.title = element_text(size = 25),plot.caption = element_text(size = 25))+
  labs(title = "Top 10 countries with the most polluters in 2019",
       x = "Country", y = "Co2 emissions (metric tons per capita)",
       caption = "Source: CO2 Emissions Around the World from kaggle")
```

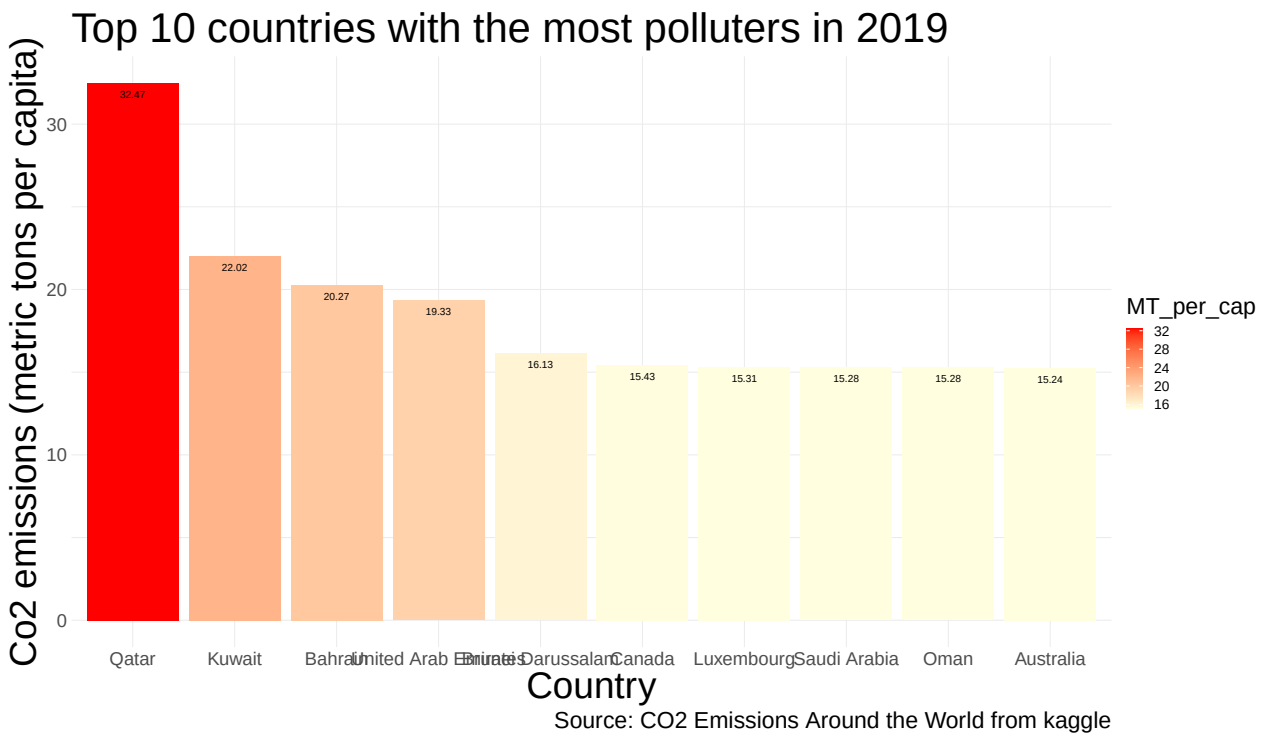
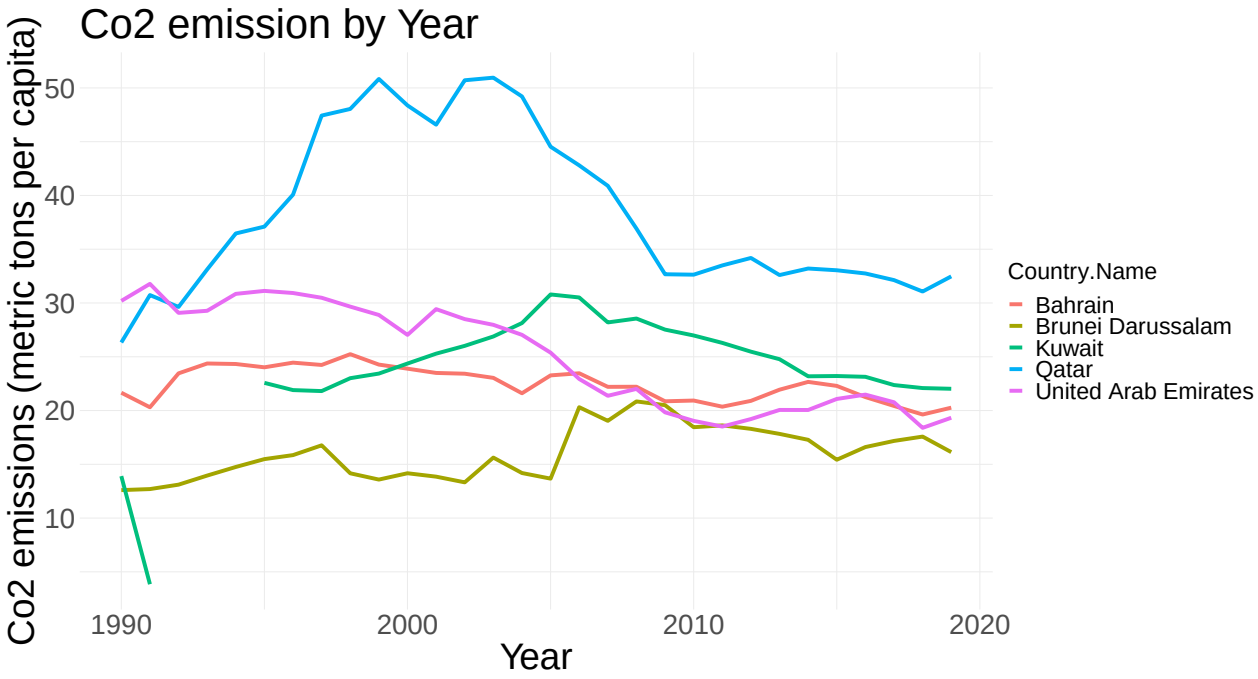


Chart4: Top 5 Most Polluting Countries by year

```
TOP5 <- c("QAT","KWT","BHR","ARE","BRN")
co2_l %>%
  filter(country_code %in% TOP5) %>%
  ggplot(aes(Year,MT_per_cap, col = Country.Name))+
           geom_line(size = 2)+
  theme_minimal()+
  theme(axis.text = element_text(size = 30),axis.title = element_text(size = 40),
        plot.title= element_text(size = 45),legend.text = element_text(size = 25),
        legend.title = element_text(size = 25),plot.caption = element_text(size = 25))+
  labs(title = "Co2 emission by Year",
       x = "Year", y = "Co2 emissions (metric tons per capita)",
       caption = "Source: CO2 Emissions Around the World from kaggle
       (https://www.kaggle.com/datasets/koustavghosh149/co2-emission-around-the-world)")
```



Source: CO2 Emissions Around the World from kaggle
(https://www.kaggle.com/datasets/koustavghosh149/co2−emission−around−the−world)

II.Diamonds dataset from R

Data preparation

1.Overview Data

```
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.~
```

Data Description

- carat : weight of the diamond
- cut : quality of the cut
- color : diamond color
- clarity : measurement of how clear the diamond is
- depth : total depth percentage
- table : width of top of diamond relative to widest point
- price : price in US dollars

- x : length in mm
- y : width in mm
- z : depth in mm

2.Check Missing Values

```
diamonds %>%
  is.na() %>%
  sum()
```

```
## [1] 0
```

There is not missing values in diamonds data set.

3.data sampling

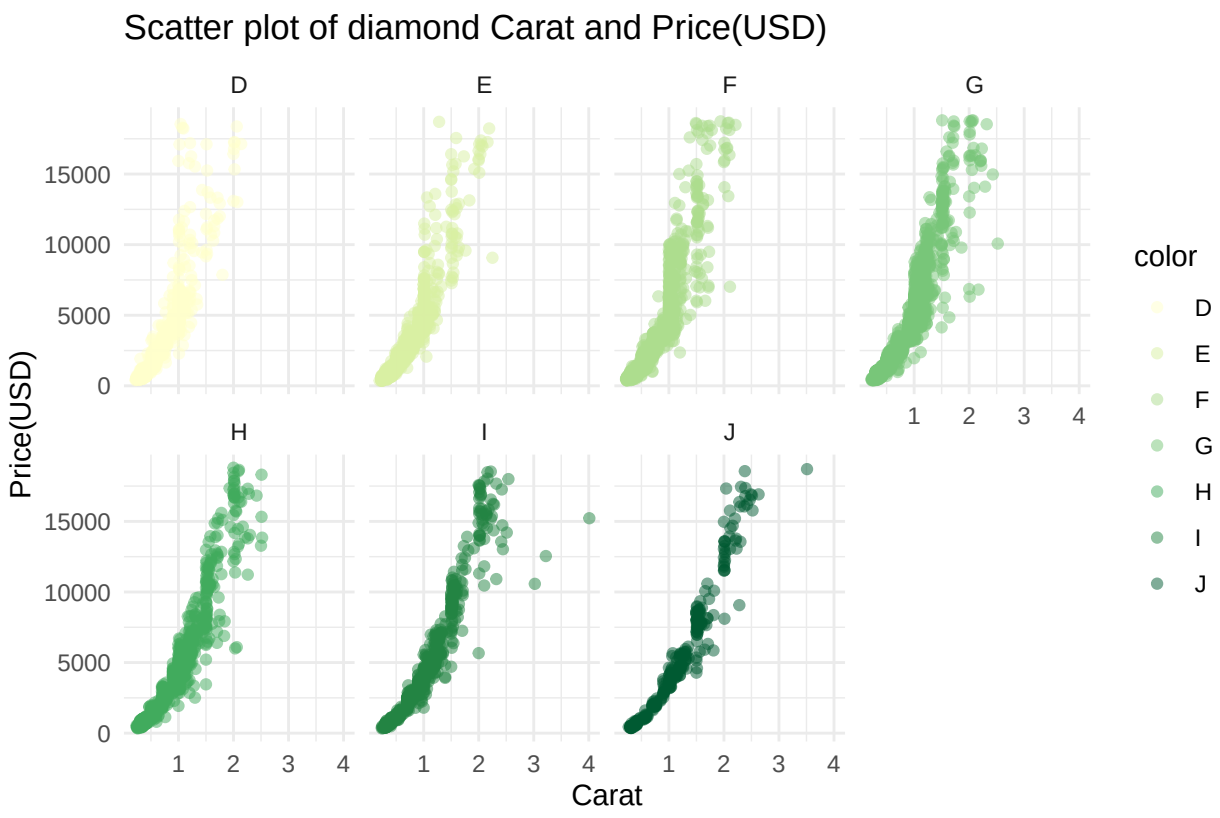
We sampled 10% of diamonds data set

```
set.seed(11)
diamonds_sampling <- diamonds %>%
  sample_n(5394)
```

Data Visualization

Chart01: The relationship between carat and price

```
diamonds_sampling %>%
  ggplot(aes(carat,price,color= color))+
  geom_point(alpha = 0.5)+
  facet_wrap(~color, ncol = 4)+
  theme_minimal()+
  scale_color_brewer(type = "qual", palette = "YlGn")+
  labs(title = "Scatter plot of diamond Carat and Price(USD)",
       x = "Carat", y = "Price(USD)",
       caption = "Source: Diamond dataset in r")
```

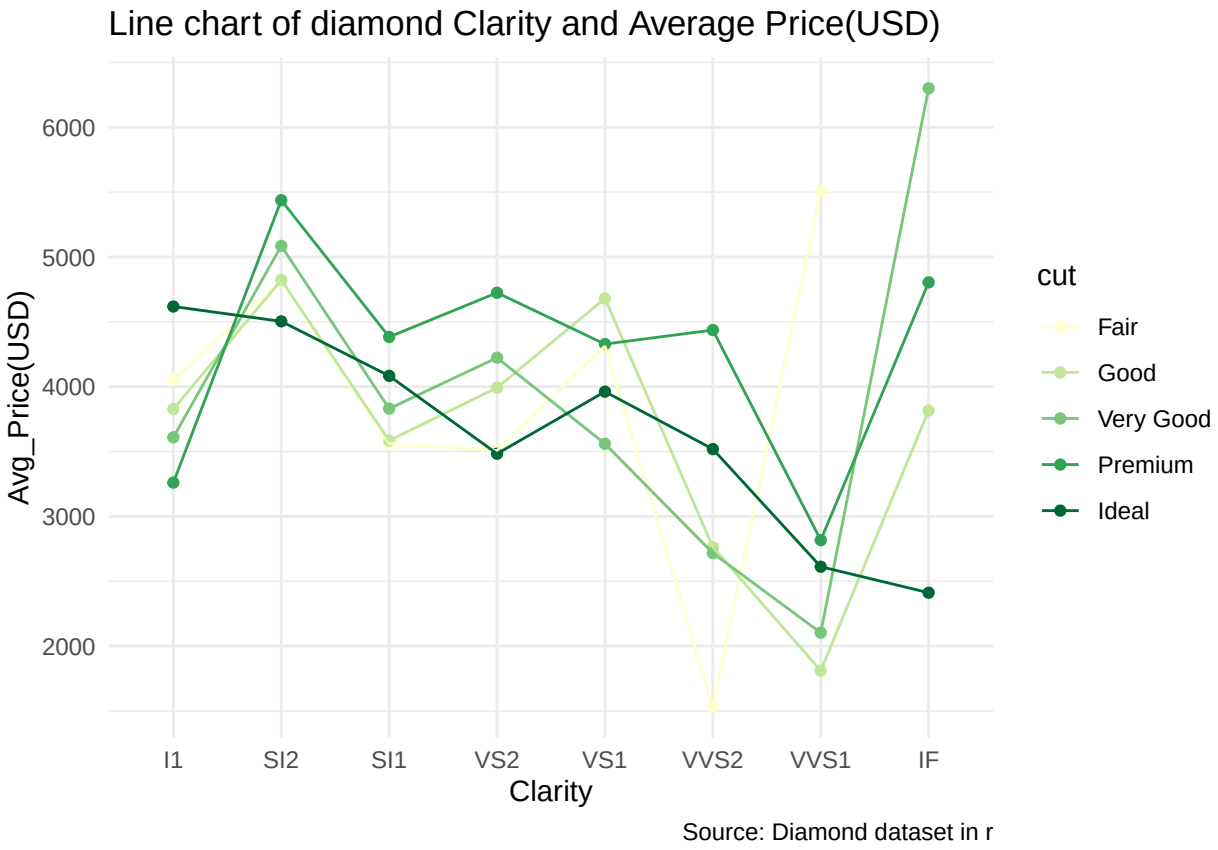


This scatter plot shows a positive correlation between carat and price.The higher the carat, the higher the price.

Chart02: Line chart between Clarity and Average Price

```
diamonds_sampling %>%
  group_by(clarity,cut) %>%
  summarise(avg_price = mean(price)) %>%
  ggplot(aes(clarity, avg_price,group = cut, col = cut))+
  geom_point()+
  geom_line()+
  theme_minimal()+
  scale_color_brewer(type = "qual", palette = "YlGn")+
  labs(title = "Line chart of diamond Clarity and Average Price(USD)",
       x = "Clarity", y = "Avg_Price(USD)",
       caption = "Source: Diamond dataset in r")
```

```
## `summarise()` has grouped output by 'clarity'. You can override using the
## `.groups` argument.
```


Line chart of diamond Clarity and Average Price(USD)
Avg_Price(USD)
6000
5000
4000
3000
2000
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
Clarity
cut
Fair
Good
Very Good
Premium
Ideal
Source: Diamond dataset in r