UD651ProblemSet4

Load the diamonds dataset.

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

```
require(dplyr)
```

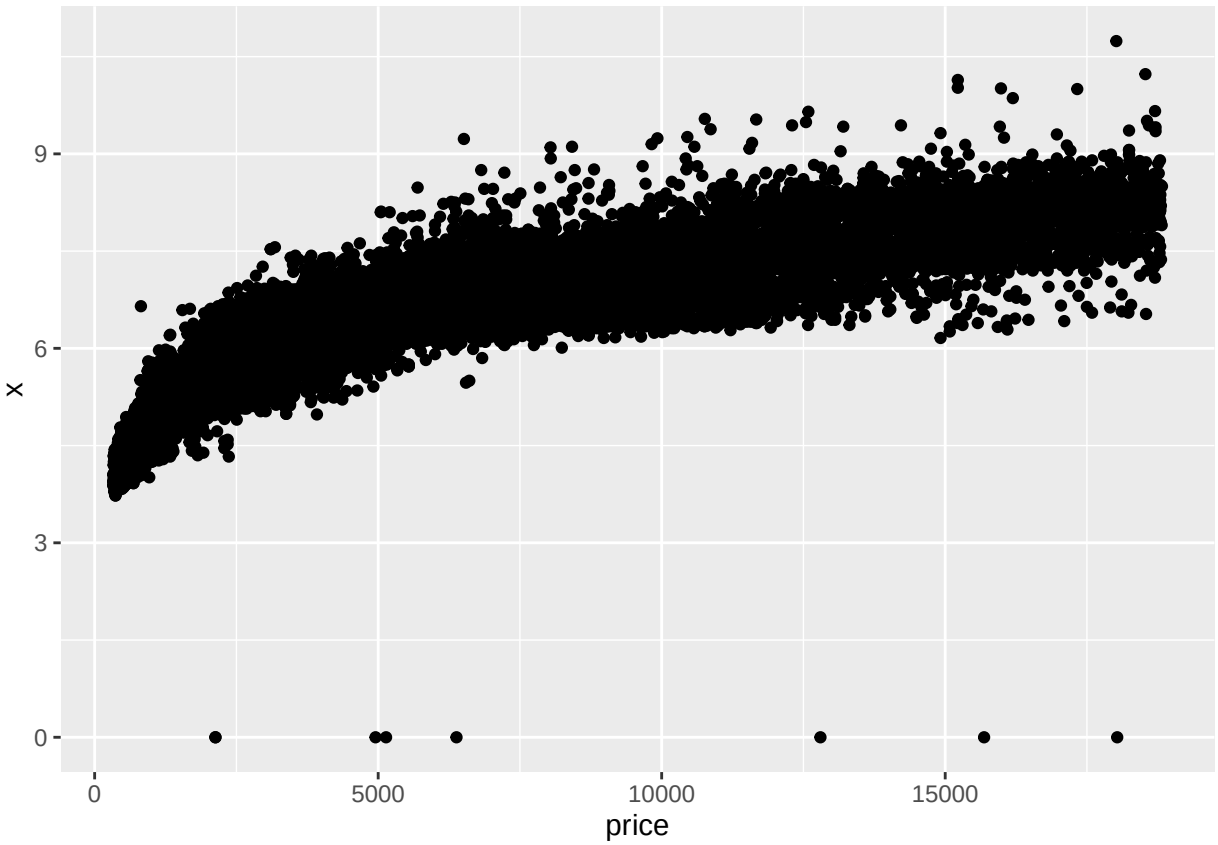
```
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(diamonds)

dm <- diamonds
```

scatter plot between Price and x

```
ggplot(aes(x=price, y=x), data=dm) + geom_point()
```



The scatter plot indicates that there is an exponential relation betweeb Price and x. There are few outliers which we can probably eliminate by limiting the X values.

Calculate the correlation between Price and (x,y,z) dimensions.

```
cor.test(dm$price, dm$x)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dm$price and dm$x
## t = 440.16, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8825835 0.8862594
## sample estimates:
##       cor
## 0.8844352
```

```
cor.test(dm$price, dm$y)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dm$price and dm$y
## t = 401.14, df = 53938, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8632867 0.8675241
## sample estimates:
##       cor
## 0.8654209
```

```
cor.test(dm$price, dm$z)
```

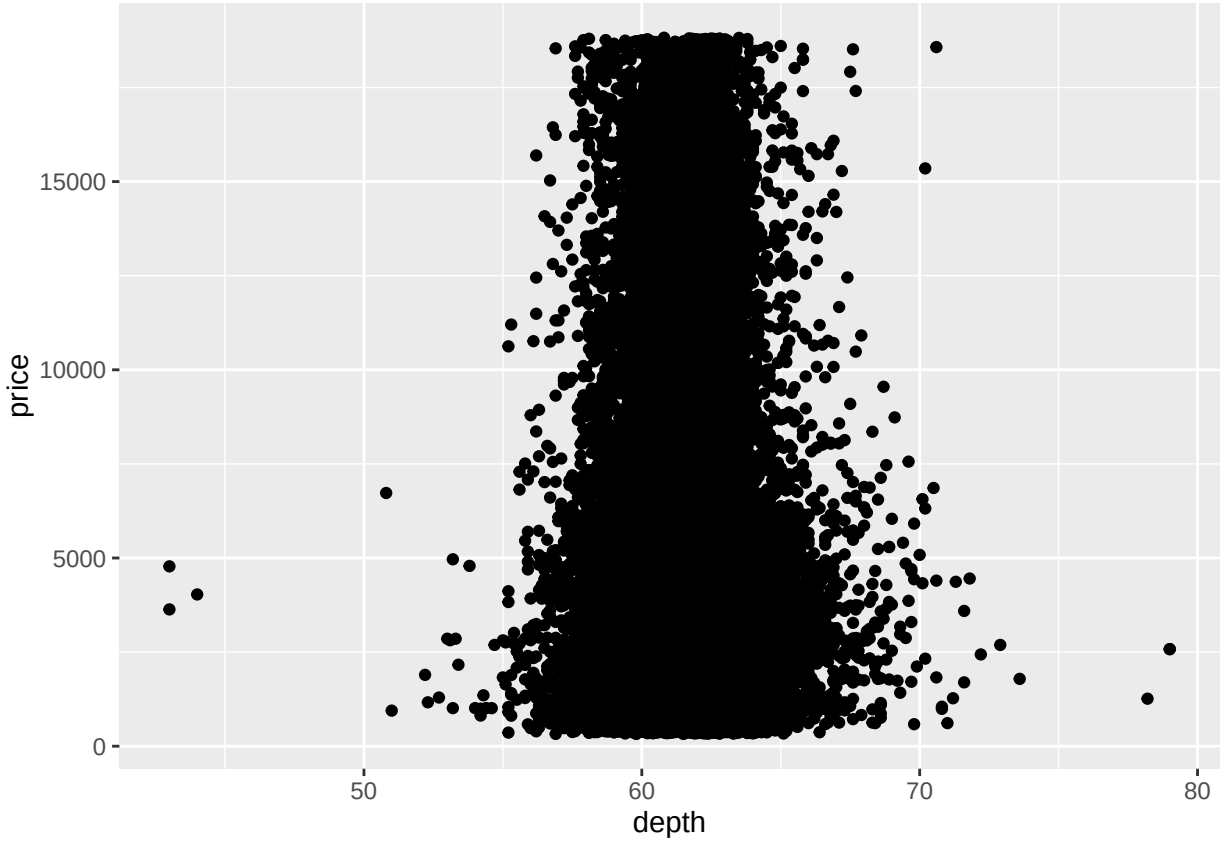
```
##
##  Pearson's product-moment correlation
##
## data:  dm$price and dm$z
## t = 393.6, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8590541 0.8634131
## sample estimates:
##       cor
## 0.8612494
```

From the above results we can conclude that the correlations are as follows:

Price and X: 0.88 Price and Y: 0.86 Price and Z: 0.86

Scatter Plot between Price and Depth:

```
ggplot(aes(x=depth, y=price), data=dm) + geom_point()
```



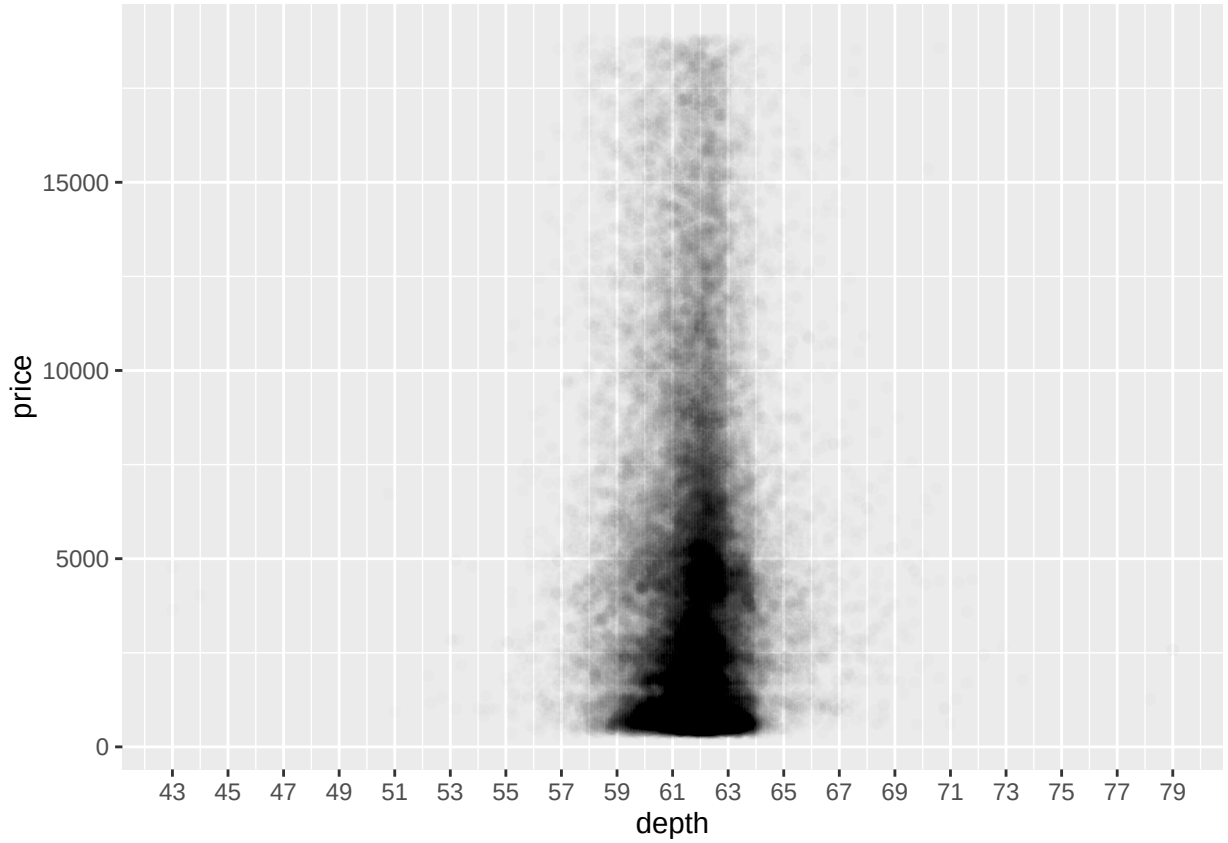
change the transparency of the points to 1/100 and plot x-axis for every two points. To plot X-axis, we need to find the range of the depth.

```
range(dm$depth)
```

```
## [1] 43 79
```

From the above output, we know that the range is 43 and 79. Setting the breaks now:

```
ggplot(data = diamonds, aes(x = depth, y = price)) +
    geom_point(alpha=1/100) +
    scale_x_continuous(breaks=seq(43, 79, 2))
```



Looking at the plot, we can see that most of the diamonds lie between the depths 59 and 64.

Calculating the correlation between Depth and Price:

```
cor.test(dm$depth, dm$price)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dm$depth and dm$price
## t = -2.473, df = 53938, p-value = 0.0134
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

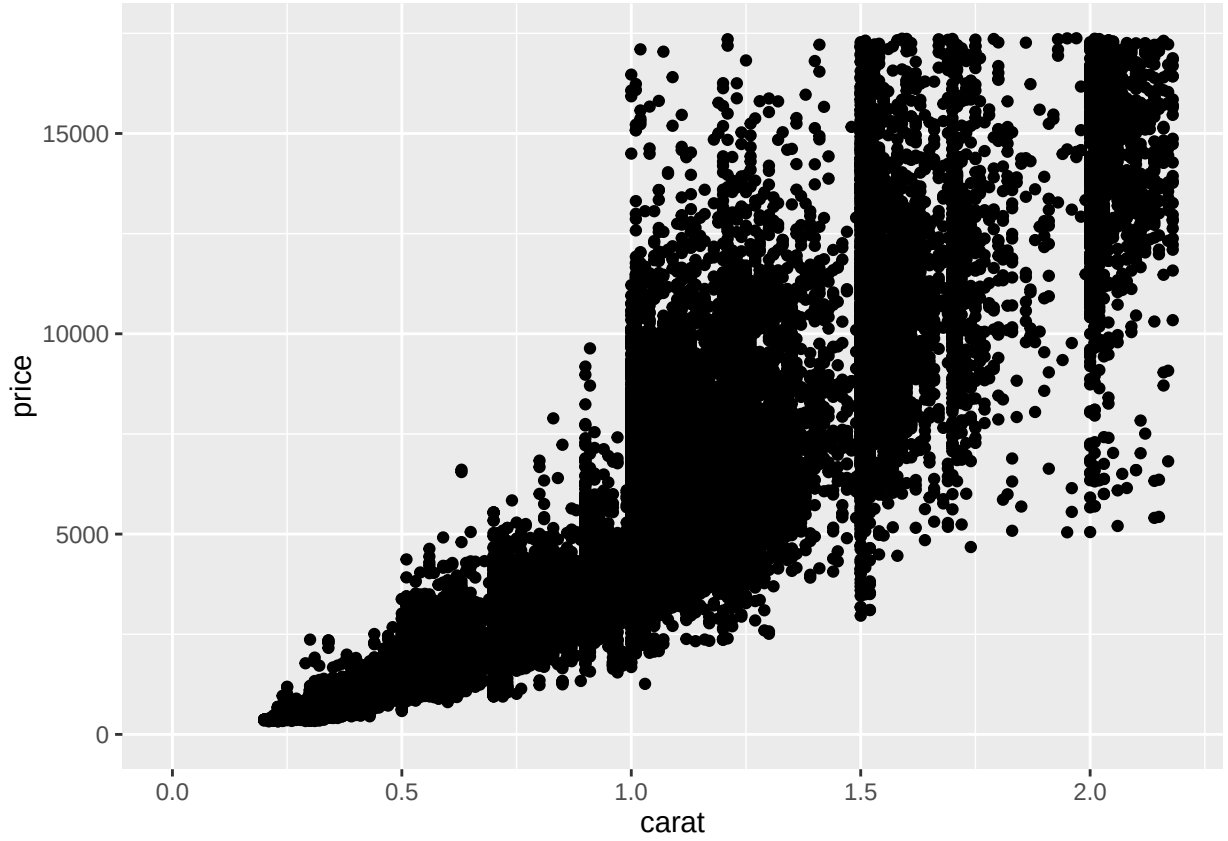
```
##  -0.019084756 -0.002208537
## sample estimates:
##        cor
## -0.0106474
```

The correlation of -0.10 indicates that the depth and price doesn't have a strong relation.

Lets draw a scatter plot between price and carat.

```
ggplot(aes(x=carat, y=price), data=dm) + geom_point() +
  xlim(0, quantile(dm$carat, 0.99)) +
  ylim(0, quantile(dm$price, 0.99))
```

```
## Warning: Removed 926 rows containing missing values (geom_point).
```

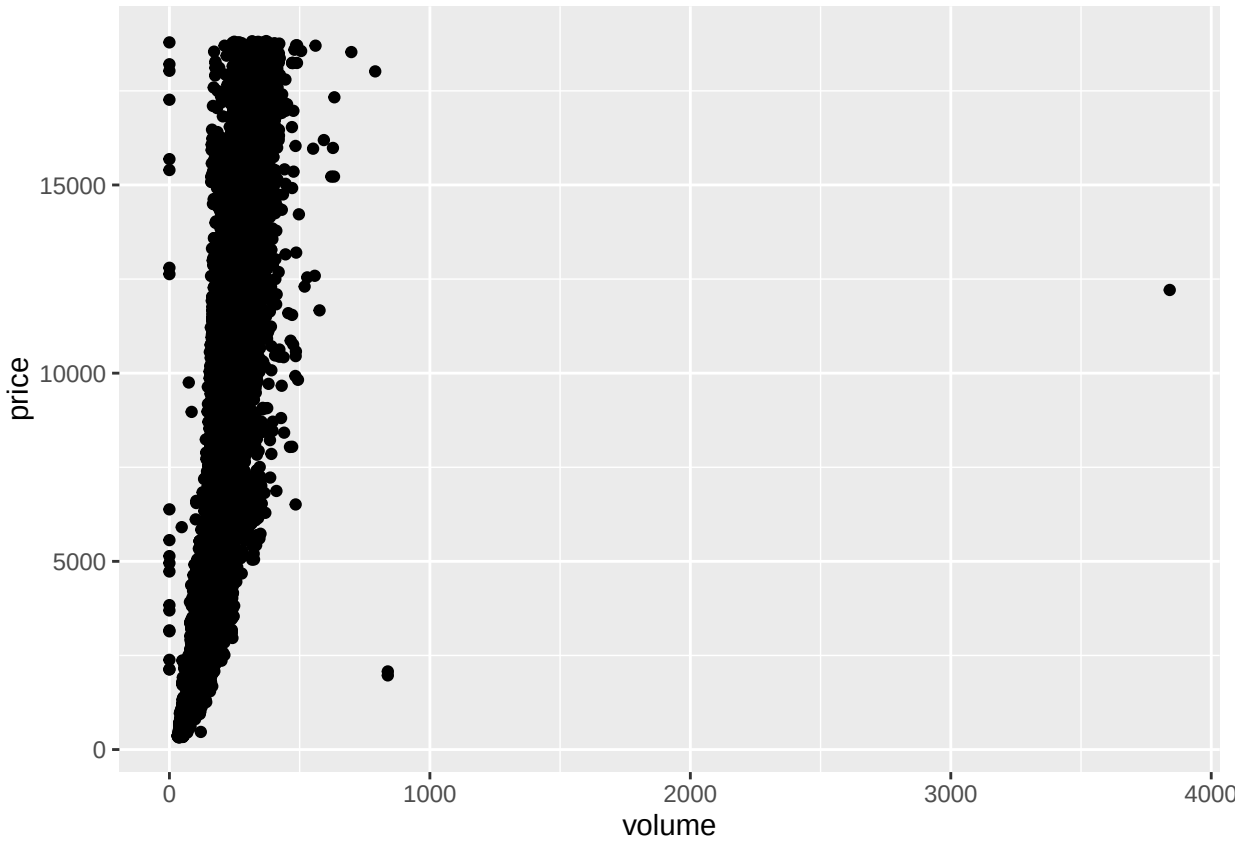


Add volume to the diamonds dataset, calculated as volume = x * y * z

```
dm <- dm %>%
  mutate(volume = x * y * z)
```

Now creating the scatter plot between price and volume:

```
ggplot(aes(x=volume, y=price), data=dm) + geom_point()
```



Calculating the correlation between price and volume by eliminating any volumes that are either equal to 0 or greater than 800.

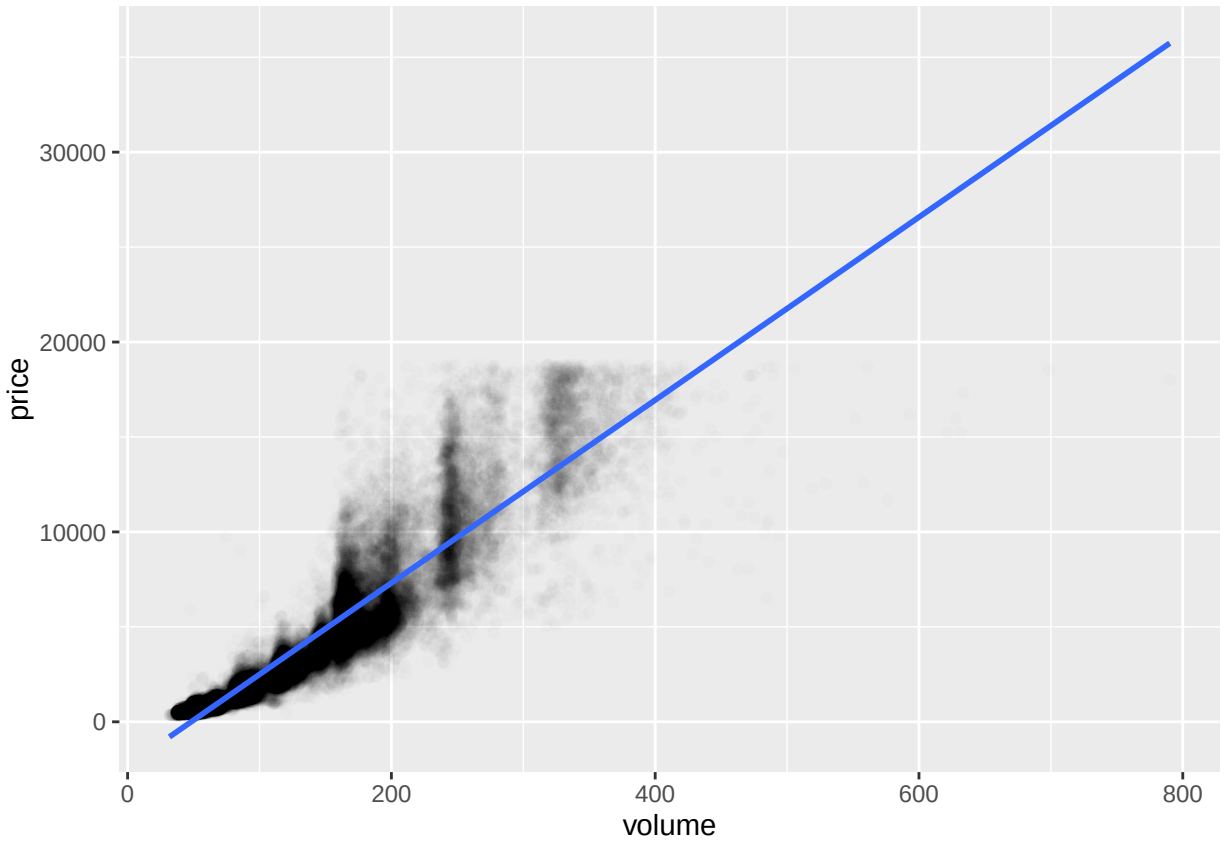
```
cor.test(dm[dm$volume > 0 & dm$volume<800,]$volume, dm[dm$volume > 0 & dm$volume<800,]$price)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dm[dm$volume > 0 & dm$volume < 800, ]$volume and dm[dm$volume > 0 & dm$volume < 800, ]$price
## t = 559.19, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9222944 0.9247772
## sample estimates:
##       cor
## 0.9235455
```

The correlation is very high at 0.92.

now lets draw a scater plot between volume and Price, but eliminating the outliers:

```
ggplot(aes(x=volume, y=price), data=dm[dm$volume > 0 & dm$volume < 800,]) +
  geom_point(alpha=1/100) +
  geom_smooth(method = "lm")
```



Creating a new dataframe called diamondsByClarity with the median, mean and other price calculations by Clarity.

```
diamondsByClarity <- dm %>%
  group_by(clarity) %>%
  summarize(mean_price = mean(price), median_price=median(price),
            min_price = min(price), max_price = max(price), n = n()) %>%
  arrange(clarity)
```

Create two more dataframes with different groupings, clarity and Color:

```
diamonds_by_clarity <- group_by(diamonds, clarity)
diamonds_mp_by_clarity <- summarise(diamonds_by_clarity, mean_price = mean(price))

diamonds_by_color <- group_by(diamonds, color)
diamonds_mp_by_color <- summarise(diamonds_by_color, mean_price = mean(price))
```

Plot bar graphs from the above two datasets and arrange them in same grid:

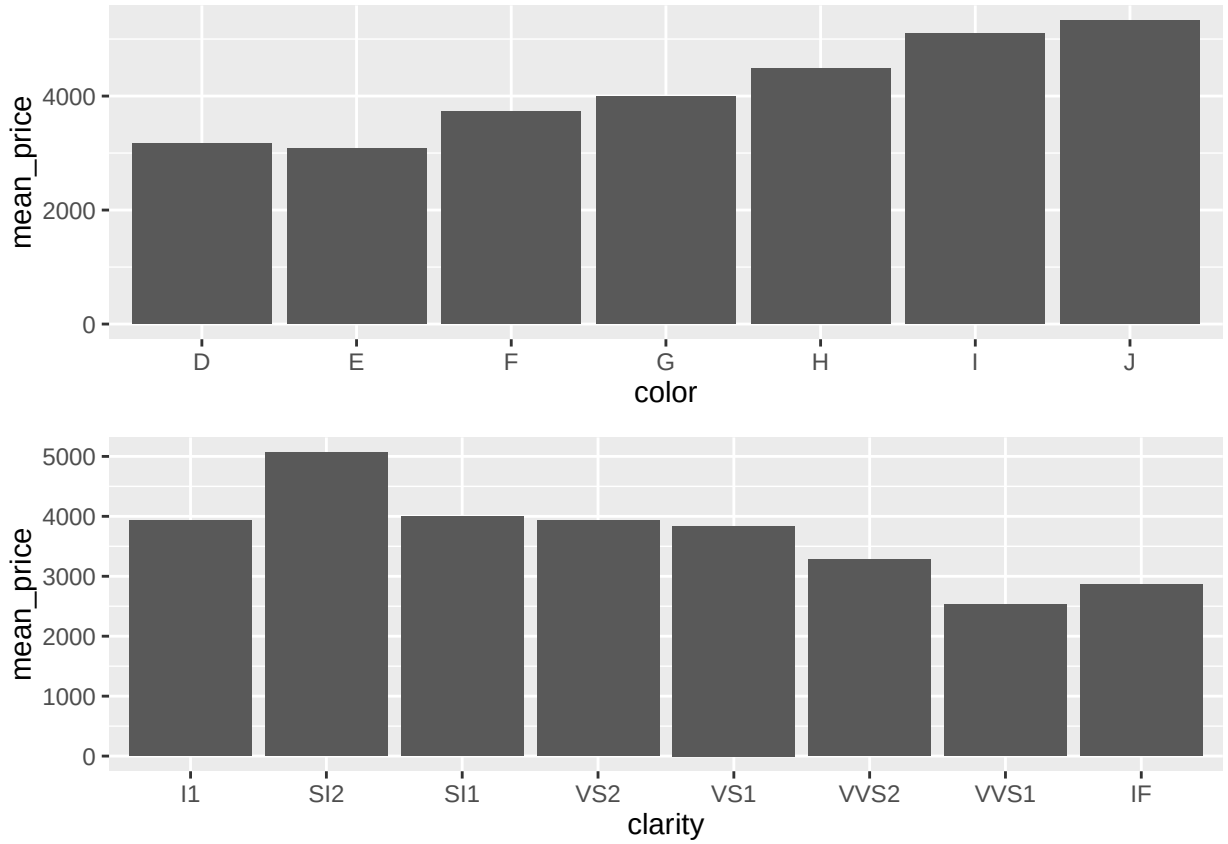
```
p1 <- ggplot(data=diamonds_mp_by_color, aes(x=color, y=mean_price)) + geom_bar(stat="identity")

p2 <- ggplot(data=diamonds_mp_by_clarity, aes(x=clarity, y=mean_price)) + geom_bar(stat="identity")

require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
grid.arrange(p1, p2, ncol=1)
```



From the graph it appears that the Mean price is directly proportional to Color, where as Clarity is inversely proportional with and exception.

Below is the final exercise where a dataset from Gapminder is downloaded.

Save the Unemployment data set for 15+ year olds for several countries for last 25+ years. The data is avaiable in Excel which was converted into CSV for the ease of reading into R.

```
require(tidyr)
```

```
## Loading required package: tidyr
```

```
unm <- read.csv("unemployment_15.csv")

head(unm)
```

```
##   Total.15..unemployment.... X1981 X1982 X1983 X1984 X1985 X1986 X1987
## 1                  Australia    NA    NA    NA    NA    NA   8.1   8.0
## 2                     Canada   7.6  11.0  11.9  11.3  10.6   9.6   8.8
## 3                  Czech Rep.    NA    NA    NA    NA    NA    NA    NA
## 4                    Estonia    NA    NA    NA    NA    NA    NA    NA
```

```
## 5                  Finland   4.8   5.3   5.4   5.0   4.9   5.2   5.0
## 6                   France   7.4   8.1   8.4   9.8  10.2  10.4  10.5
##   X1988 X1989 X1990 X1991 X1992 X1993 X1994 X1995 X1996 X1997 X1998 X1999
## 1   7.2   6.1   6.9   9.5  10.4  10.5   9.4   8.2   8.2   8.2   7.7   6.9
## 2   7.7   7.5   8.1  10.3  11.1  11.3  10.3   9.4   9.6   9.1   8.3   7.6
## 3    NA    NA    NA    NA    NA   4.3   4.3   4.0   3.9   4.8   6.4   8.7
## 4    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 5   4.4   3.1   3.1   6.5  11.6  16.1  16.4  15.2  14.4  12.6  11.3  10.2
## 6  10.0   9.4   8.9   9.4  10.2  11.5  12.1  11.4  12.0  12.1  11.5  10.8
##   X2000 X2001 X2002 X2003 X2004 X2005  X
## 1   6.2   6.7   6.4   6.0   5.5   5.1 NA
## 2   6.8   7.2   7.6   7.6   7.2   6.7 NA
## 3   8.8   8.2   7.3   7.8   8.3   8.3 NA
## 4  13.7  12.5  10.2  10.0   9.5   7.9 NA
## 5   9.7   9.1   9.0   9.0   8.8   8.3 NA
## 6   9.5   8.7   9.0   9.8  10.0   9.9 NA
```

```
names(unm) <- c("country",1981:2005,"dummy")
```

reshape the data, convert the data from columns to rows. Filter the NAs after the conversion.

```
unmg <- gather(unm, 'year',"unm",2:27)

unmg <- filter(unmg, !is.na(unm))
```

Let us create two dataframes, one at country level and another at the year level, so as to understand the relation between unemployment rate and these variables.

```
unmg_country <- unmg %>%
  group_by(country) %>%
  summarize(mean_unm = mean(unm), median_unm= median(unm), n = n())

head(unmg_country)
```

```
## Source: local data frame [6 x 4]
##
##      country  mean_unm median_unm     n
##       (fctr)     (dbl)      (dbl) (int)
## 1  Australia  7.560000       7.45    20
## 2     Canada  8.968000       8.80    25
## 3 Czech Rep.  6.546154       7.30    13
## 4    Estonia 10.633333      10.10     6
## 5    Finland  8.576000       8.80    25
## 6     France 10.040000      10.00    25
```

```
unmg_year <- unmg %>%
  group_by(year) %>%
  summarize(mean_unm = mean(unm), median_unm= median(unm), n = n())

head(unmg_year)
```

```
## Source: local data frame [6 x 4]
```

```
##
##     year mean_unm median_unm     n
##   (fctr)    (dbl)      (dbl) (int)
## 1   1981 5.611111        4.8     9
## 2   1982 6.666667        5.3     9
## 3   1983 7.810000        6.9    10
## 4   1984 8.010000        6.2    10
## 5   1985 7.781818        4.9    11
## 6   1986 7.757143        6.2    14
```

add a new column to unmg_year to convert the year to a numeric:

```
unmg_year$year_num <- as.numeric(unmg_year$year)
```

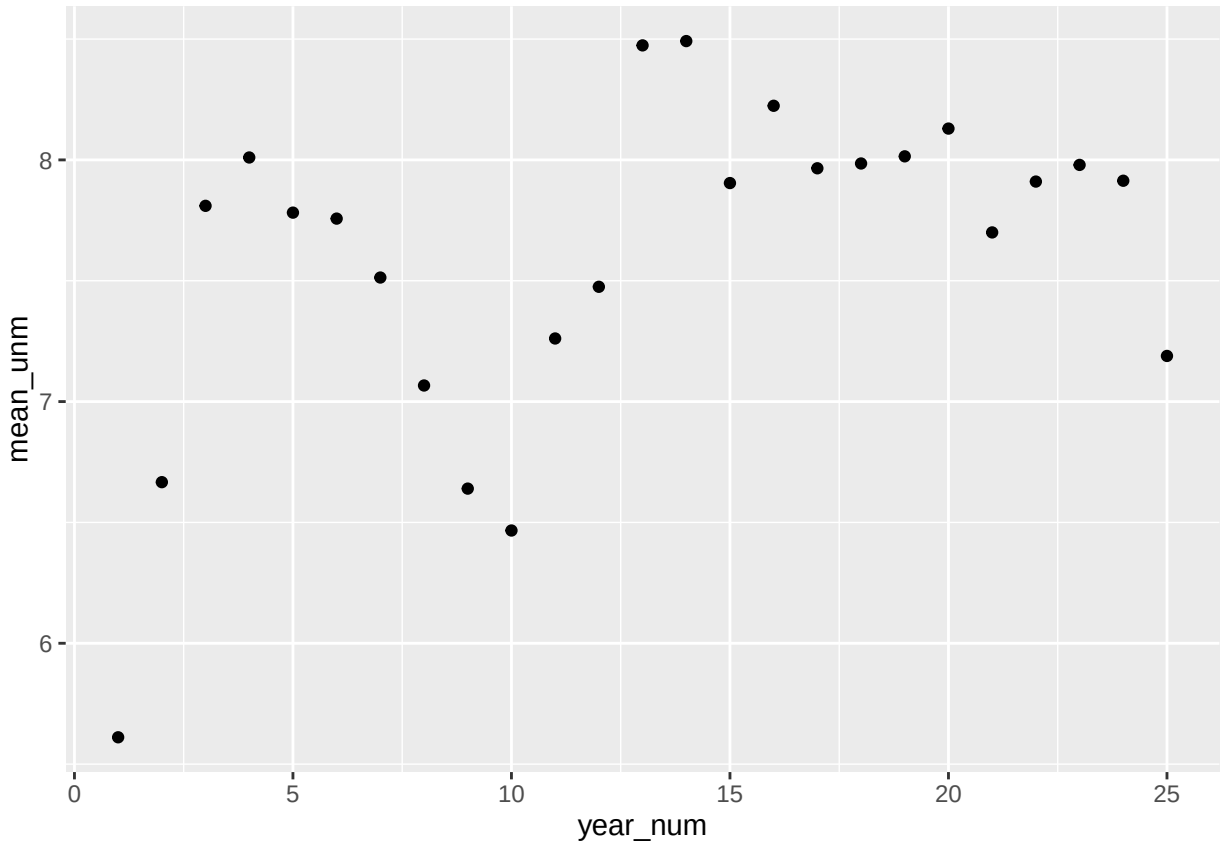
check to see if there is a correlation between Year and Mean unemployment rate:

```
cor.test(unmg_year$year_num, unmg_year$mean_unm)
```

```
##
##  Pearson's product-moment correlation
##
## data:  unmg_year$year_num and unmg_year$mean_unm
## t = 2.5321, df = 23, p-value = 0.01862
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08800603 0.72776811
## sample estimates:
##      cor
## 0.466901
```

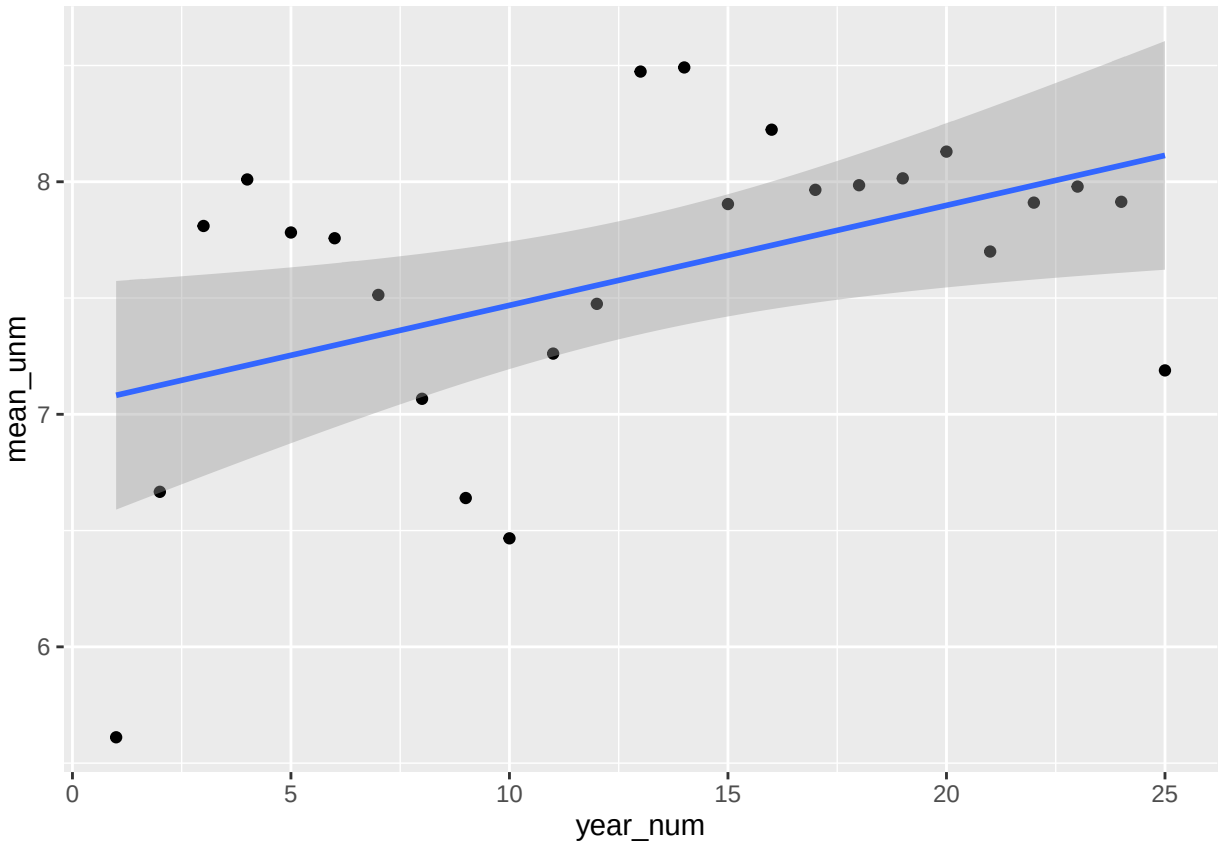
A correlation of 0.46 indicates that there is a considerable correlation among the two variables. Lets draw a scatter plot to visualize the relation:

```
ggplot(data = unmg_year, aes(x=year_num, y = mean_unm)) + geom_point()
```



Lets draw a Liner model to the graph:

```
ggplot(data = unmg_year, aes(x=year_num, y = mean_unm)) + geom_point() +
  geom_smooth(method = 'lm')
```

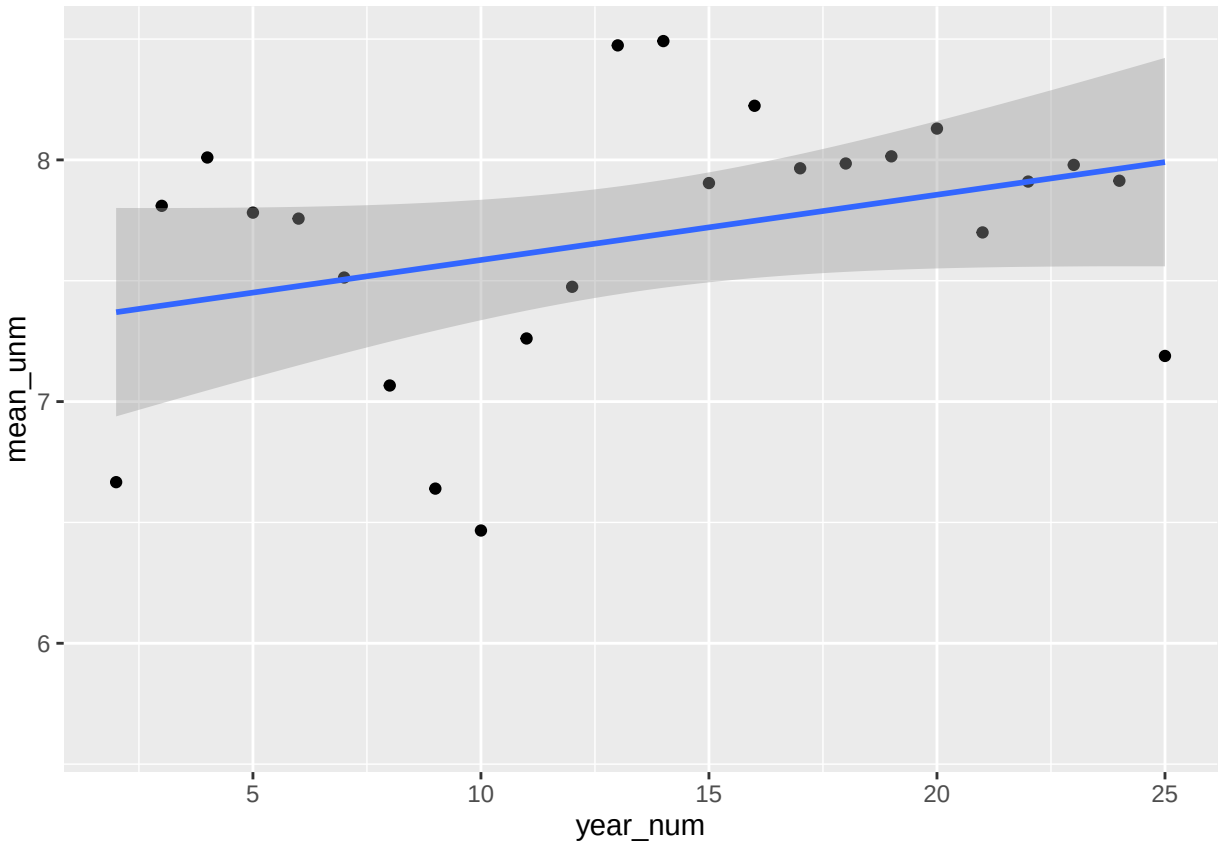


The linear model indicates a growing trend in unemployment rate as the years progress. from the scatter plot, one of the points looks like an outlier, lets try to remove this and plot the lm again:

```
ggplot(data = unmg_year, aes(x=year_num, y = mean_unm)) + geom_point() +
  xlim(c(2,25)) +
  geom_smooth(method = 'lm')
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



It would be a good idea to overlay this information with growth in population and see if there is a correlation between Population growth and unemployment rate. Analysis for some other day.