EDAProblemSet3

Load the required libraries

```
require(gridExtra)
```

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

```
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
```

```
require(tidyr)
```

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

```
require(ggplot2)
```

```
## Loading required package: ggplot2

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

Looking at the structure of the diamonds dataset

```
require(datasets)
str(diamonds)
```

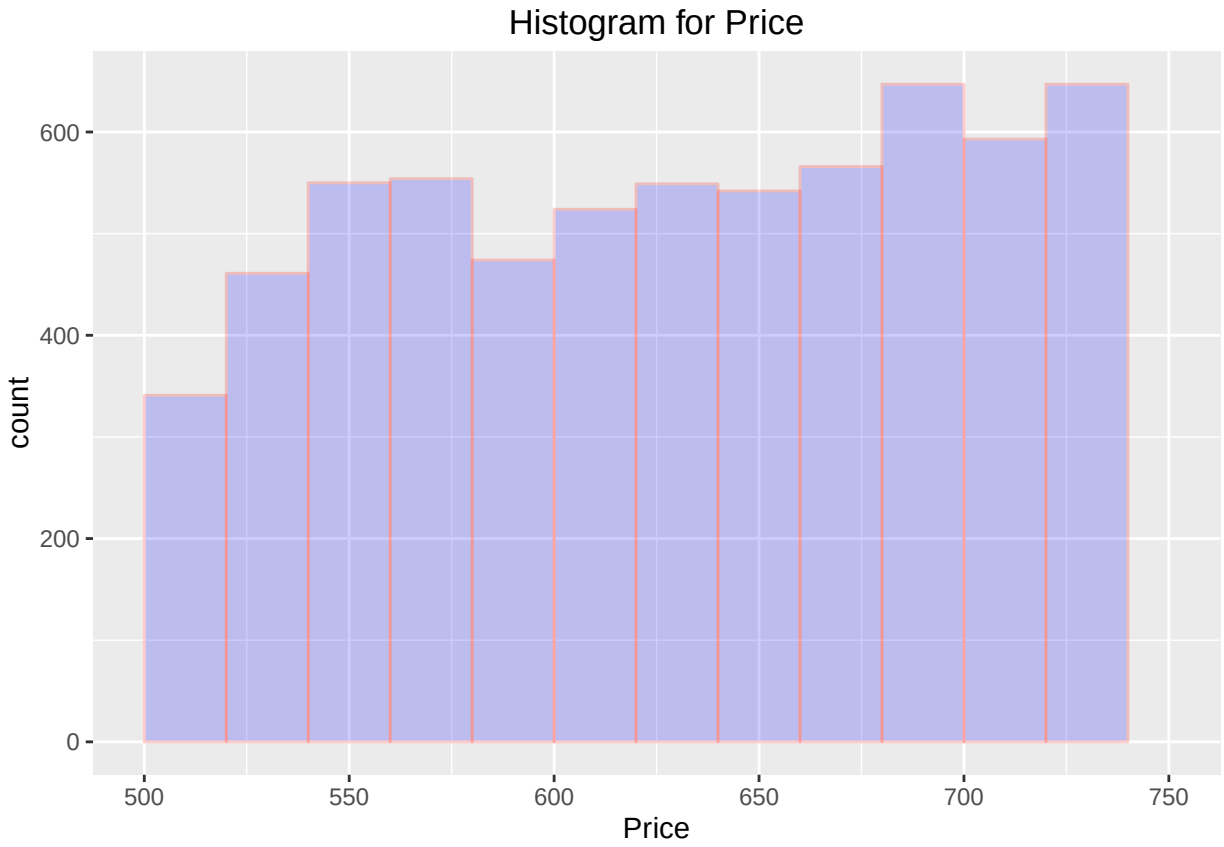
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
#unique(diamonds$color)
```

Plot the histogram for Price

```
qplot(diamonds$price,
      geom="histogram",
     # binwidth=0.1,
      main="Histogram for Price",
      xlab="Price",
      fill=I("blue"),
      col=I("red"),
      alpha=I(.2),
    breaks=seq(500, 750,by=20),
      xlim = c(500,750))
```

```
## Warning: Removed 47241 rows containing non-finite values (stat_bin).
```



Plot a graph for each type of cut and then arrange them in a grid.

```
p1 <- qplot(diamonds[diamonds$cut == 'Fair',]$price,
            geom="histogram",
            # binwidth=0.1,
            main="Histogram for Price",
            xlab="Price",
            fill=I("blue"),
```

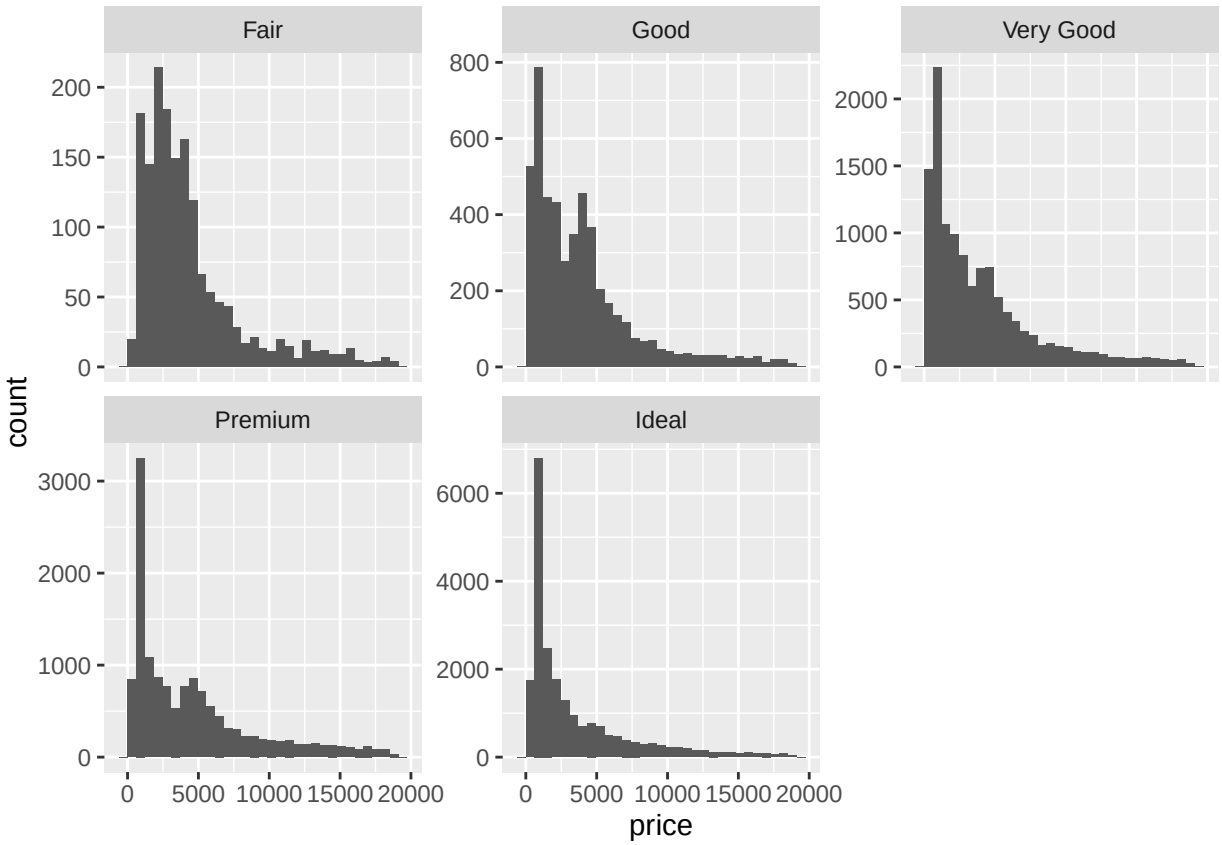
```
            col=I("black"),
            alpha=I(.2))

p2 <- qplot(diamonds[diamonds$cut == 'Good',]$price,
            geom="histogram",
            # binwidth=0.1,
            main="Histogram for Price",
            xlab="Price",
            fill=I("blue"),
            col=I("black"),
            alpha=I(.2))

p3 <- qplot(diamonds[diamonds$cut == 'Very Good',]$price,
            geom="histogram",
            # binwidth=0.1,
            main="Histogram for Price",
            xlab="Price",
            fill=I("blue"),
            col=I("black"),
            alpha=I(.2))

p4 <- qplot(diamonds[diamonds$cut == 'Premium',]$price,
            geom="histogram",
            # binwidth=0.1,
            main="Histogram for Price",
            xlab="Price",
            fill=I("blue"),
            col=I("black"),
            alpha=I(.2))

p5 <- qplot(diamonds[diamonds$cut == 'Ideal',]$price,
            geom="histogram",
            # binwidth=0.1,
            main="Histogram for Price",
            xlab="Price",
            fill=I("blue"),
            col=I("black"),
            alpha=I(.2))

grid.arrange(p1, p2, p3,p4,p5,ncol=2)
```

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

Implement the same graph, using facet_wrap functionality in ggplot and not have a fixed y-axis scale.

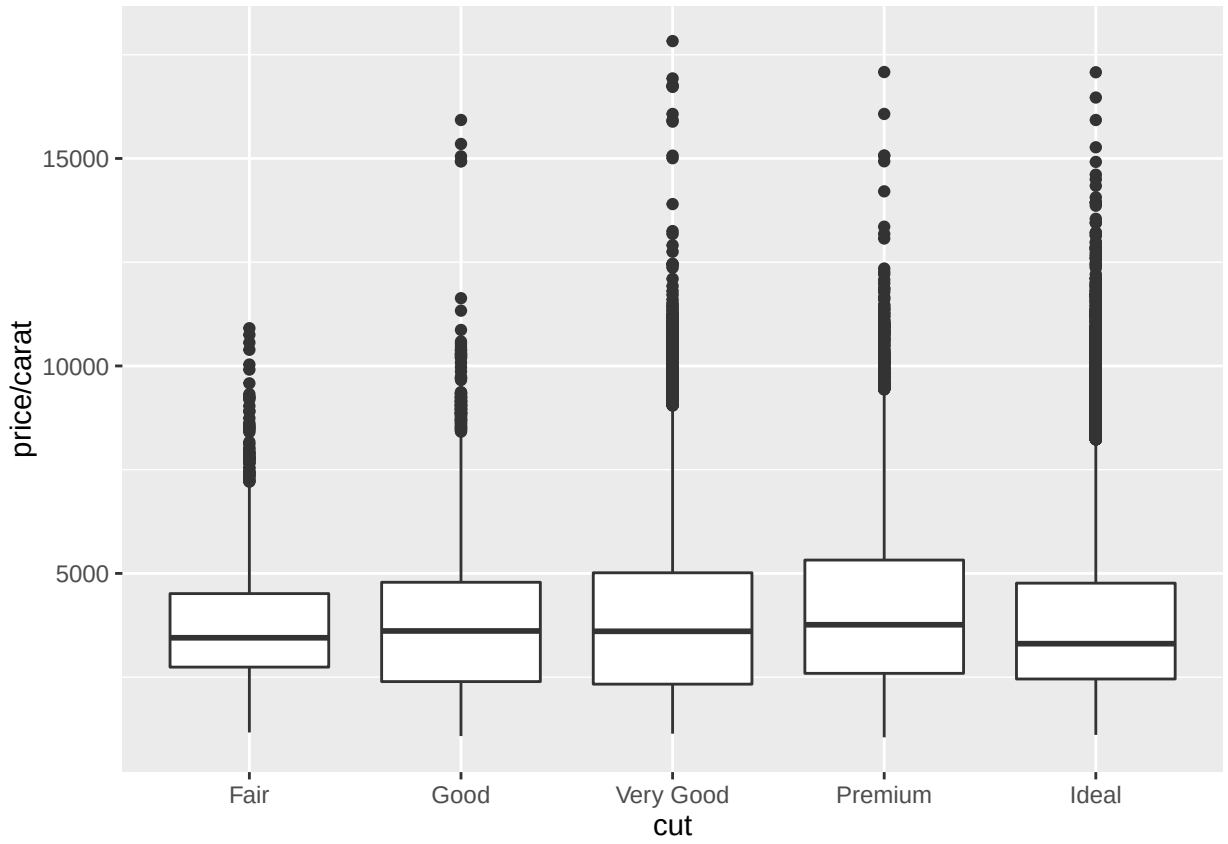
```
qplot(x = price, data = diamonds) + facet_wrap(~cut, scales="free_y")
```

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



Plot a graph (boxplot) for the price/carat for each type of cut and save the graphs to a local directory.

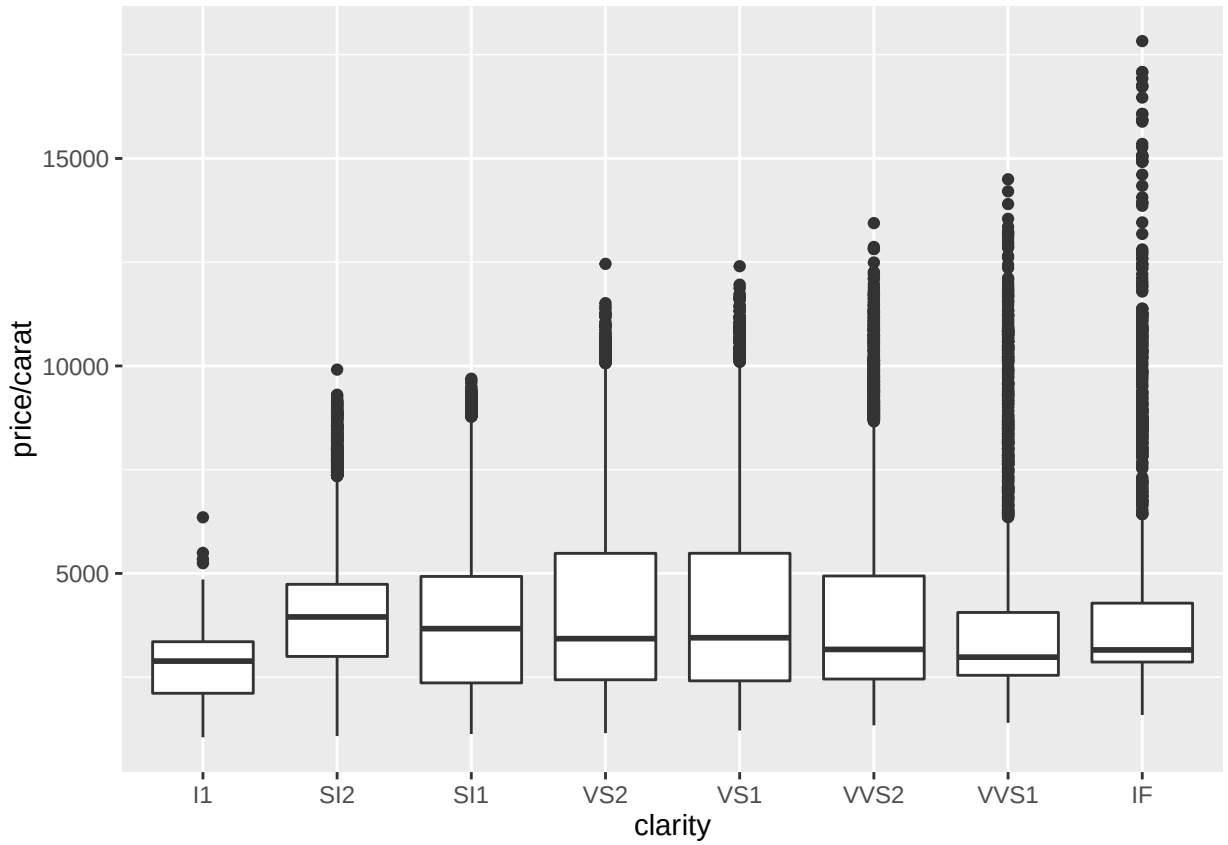
```
ggplot(diamonds, aes(cut,price/carat)) + geom_boxplot()
```




```
ggsave("box_by_cut.png")
```

```
## Saving 6.5 x 4.5 in image
```

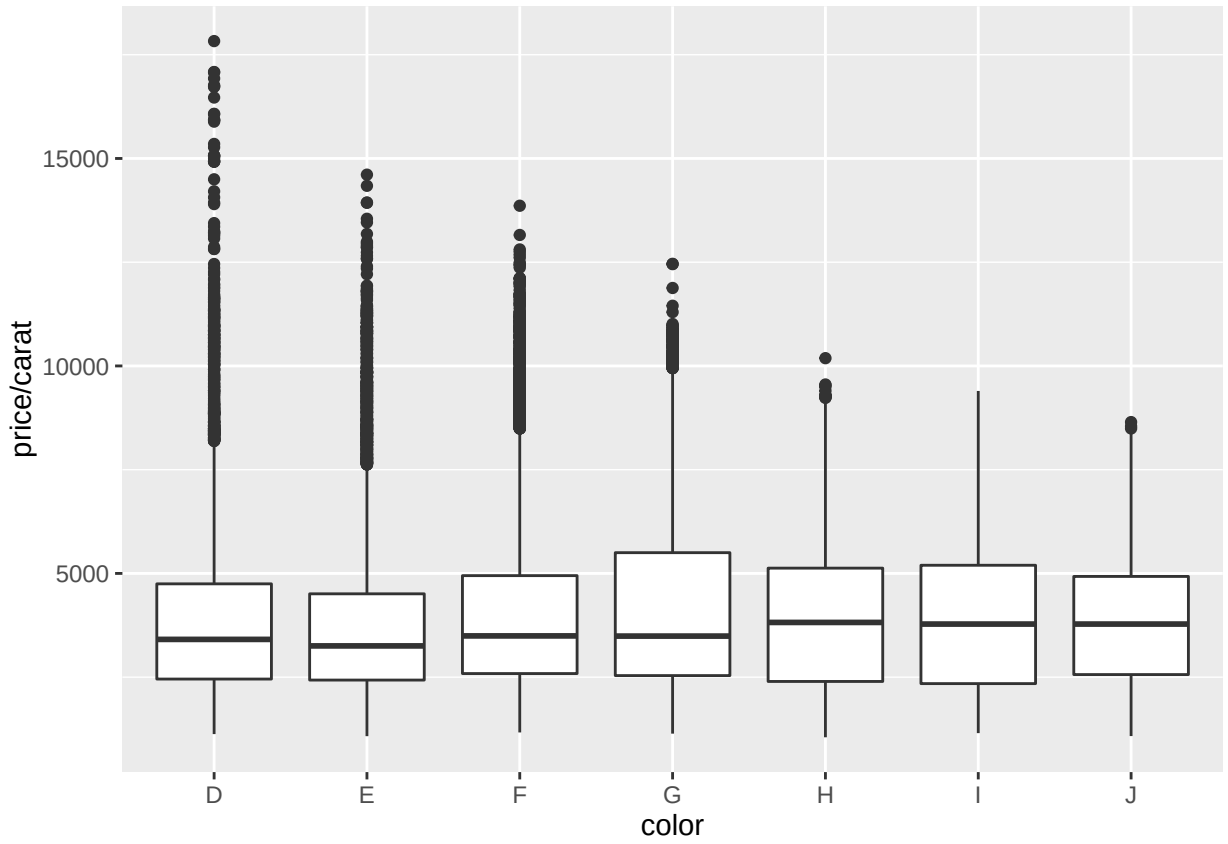
```
ggplot(diamonds, aes(clarity,price/carat)) + geom_boxplot()
```




```
ggsave("box_by_clarity.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggplot(diamonds, aes(color,price/carat)) + geom_boxplot()
```




```
ggsave("box_by_color.png")
```

```
## Saving 6.5 x 4.5 in image
```

Looking at the summary of Price for best and worst colors.

```
summary(subset(diamonds, color=="D")$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     357     911    1838    3170    4214   18690
```

```
summary(subset(diamonds, color=="J")$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     335    1860    4234    5324    7695   18710
```

```
IQR(subset(diamonds, color=="D")$price)
```

```
## [1] 3302.5
```

```
IQR(subset(diamonds, color=="J")$price)
```

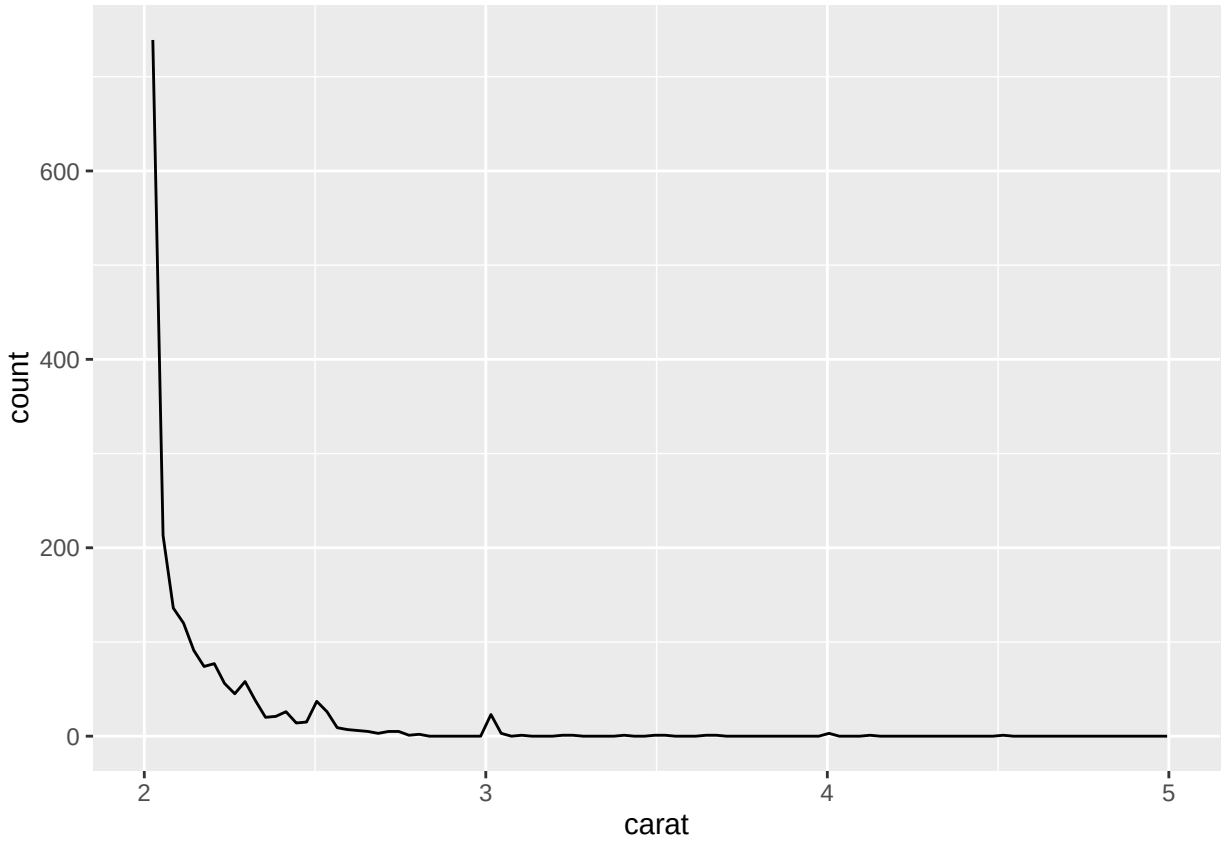
```
## [1] 5834.5
```

Draw a frequency polygib to check the frequency of different weights of the diamonds.

```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(2.0,5.0))
```

```
## Warning: Removed 51787 rows containing non-finite values (stat_bin).
```

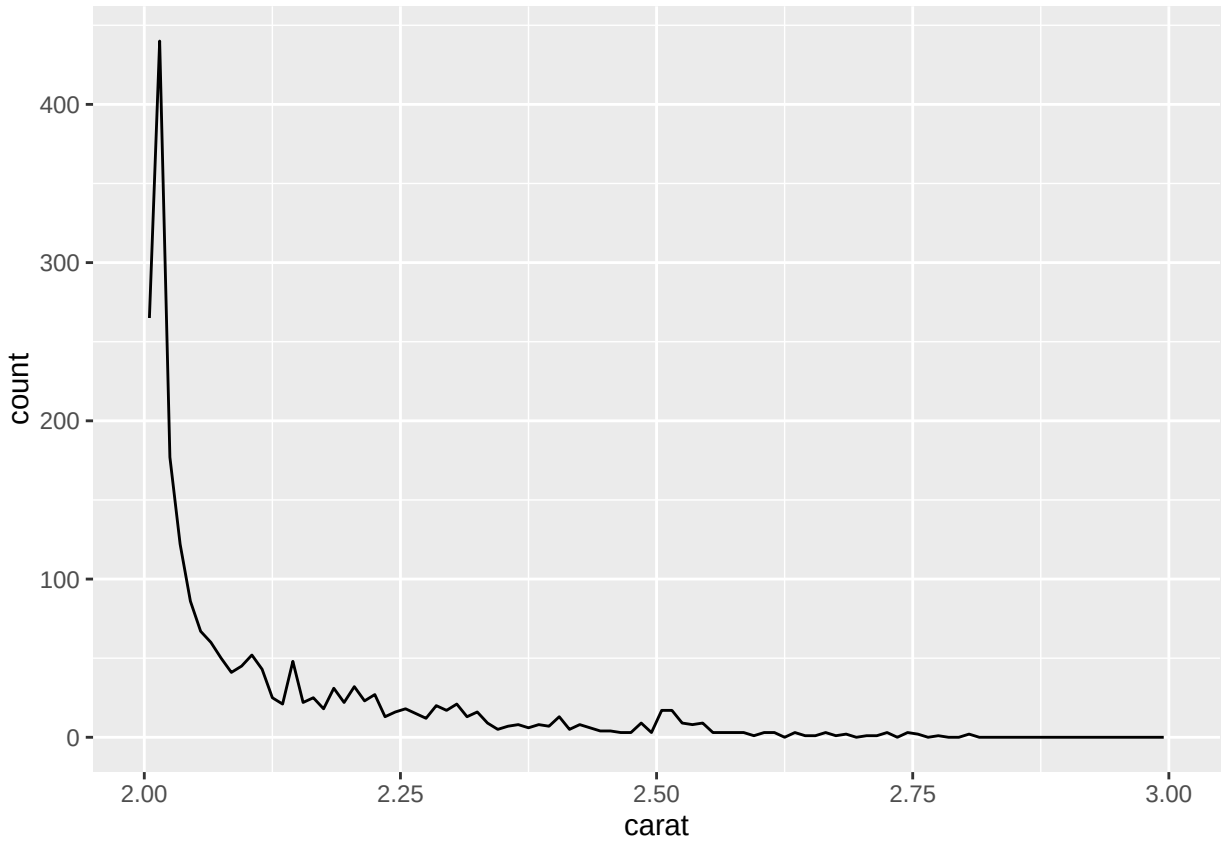
```
## Warning: Removed 3 rows containing missing values (geom_path).
```




```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(2.0,3.0))
```

```
## Warning: Removed 51818 rows containing non-finite values (stat_bin).
```

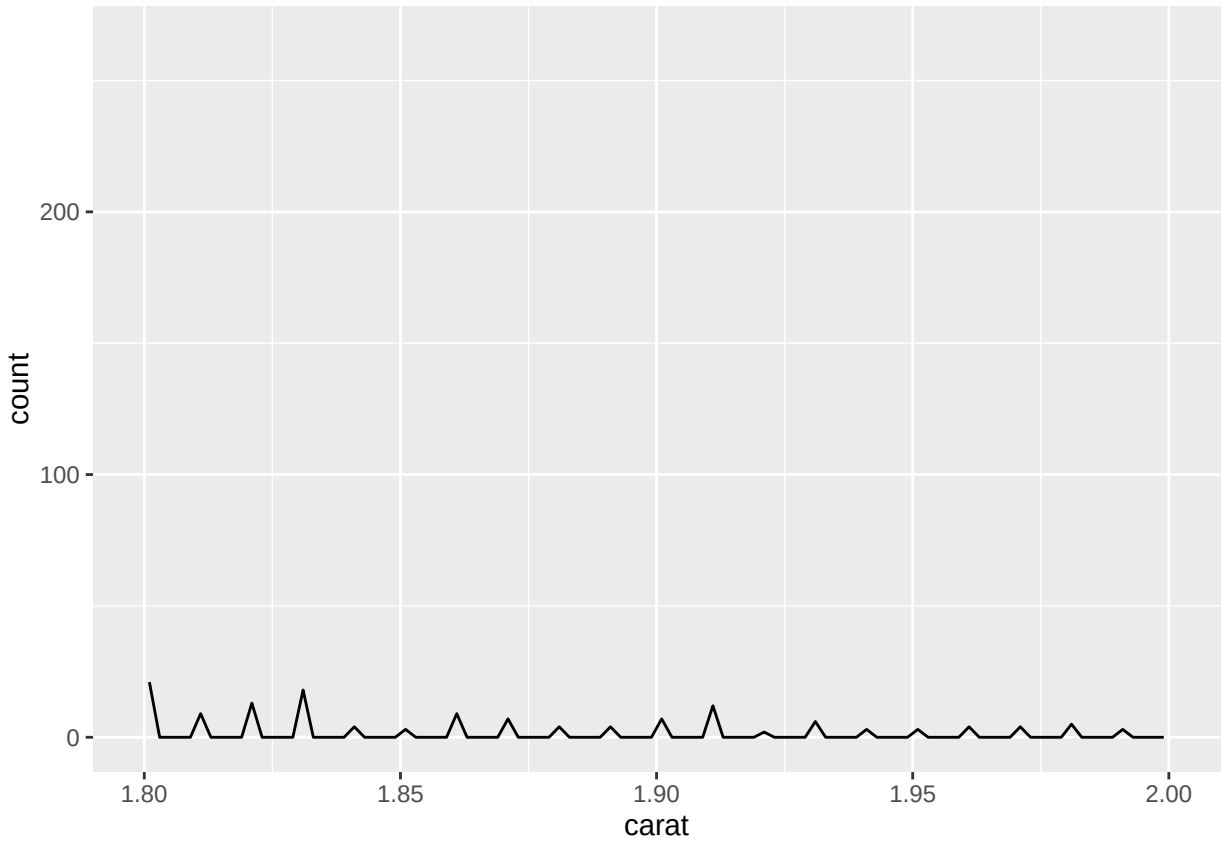
```
## Warning: Removed 2 rows containing missing values (geom_path).
```




```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(1.8,2.0))
```

```
## Warning: Removed 53534 rows containing non-finite values (stat_bin).
```

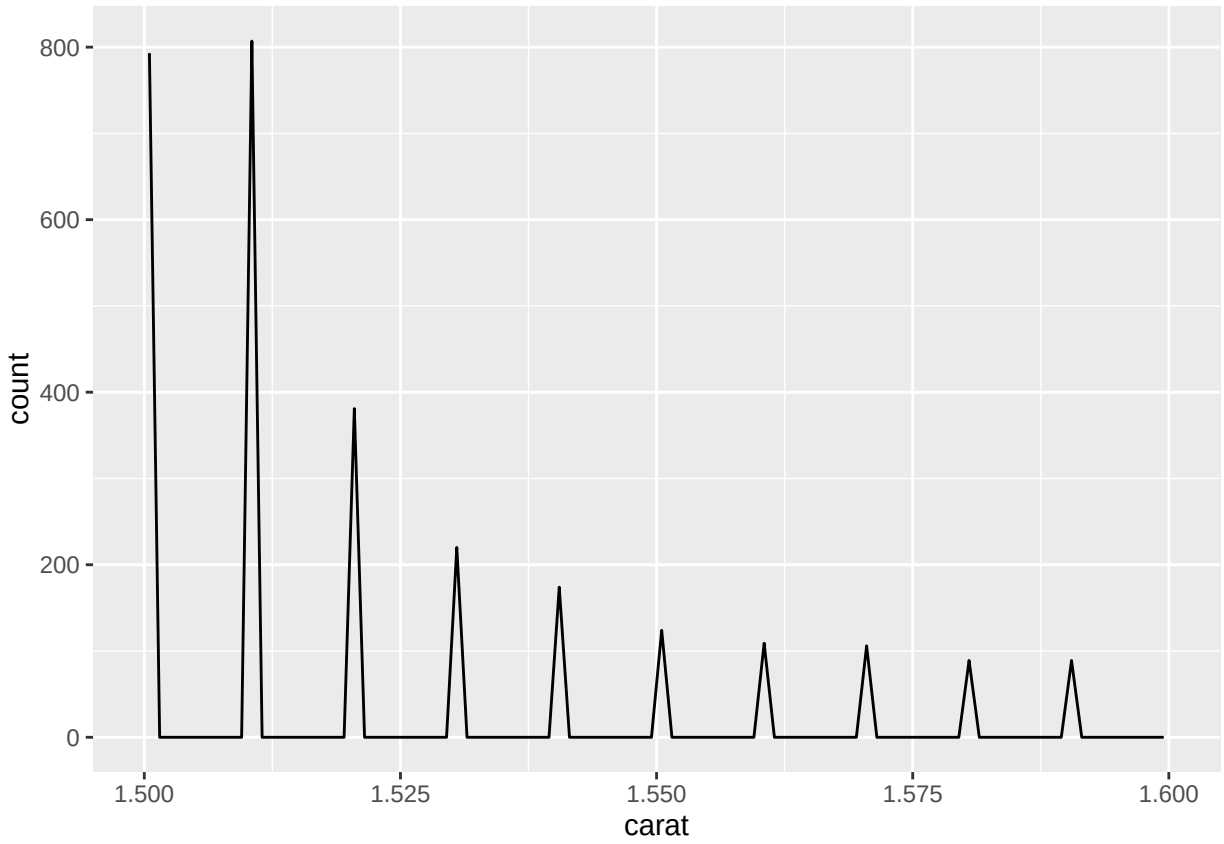
```
## Warning: Removed 3 rows containing missing values (geom_path).
```




```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(1.5,1.6))
```

```
## Warning: Removed 50953 rows containing non-finite values (stat_bin).
```

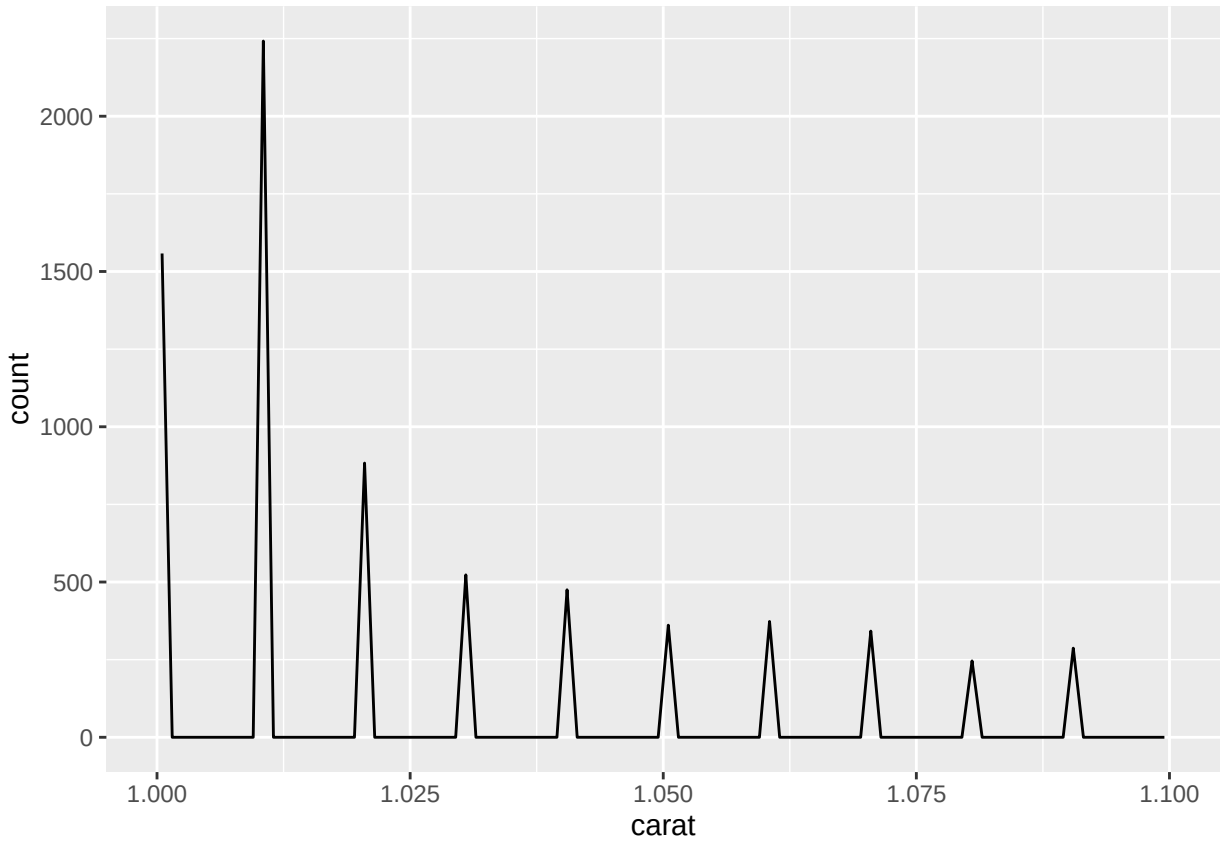
```
## Warning: Removed 3 rows containing missing values (geom_path).
```




```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(1.0,1.1))
```

```
## Warning: Removed 46372 rows containing non-finite values (stat_bin).
```

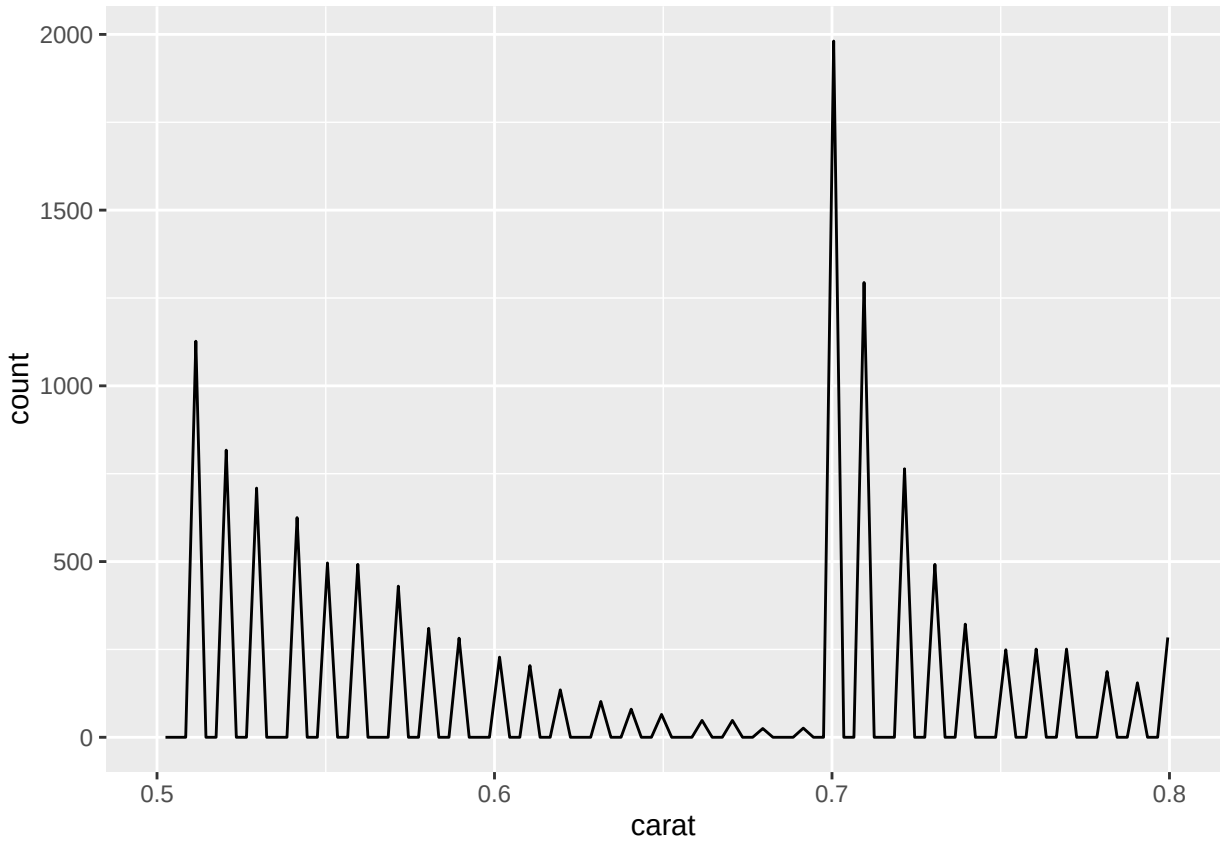
```
## Warning: Removed 3 rows containing missing values (geom_path).
```




```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(0.5,0.8))
```

```
## Warning: Removed 40203 rows containing non-finite values (stat_bin).
```

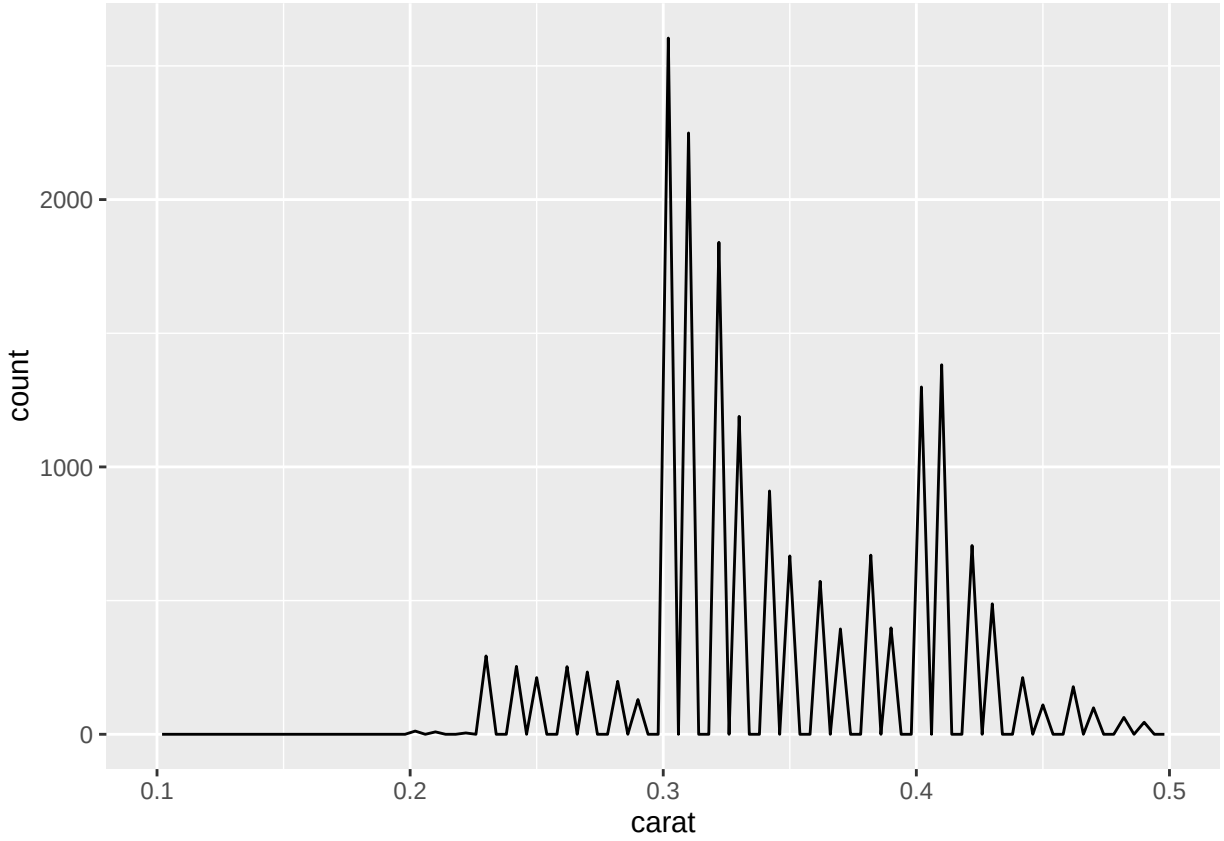
```
## Warning: Removed 3 rows containing missing values (geom_path).
```




```
qplot(carat, data = diamonds, geom = "freqpoly", bins=100, xlim=c(0.1,0.5))
```

```
## Warning: Removed 35008 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```



Below code is to answer the last set of questions in Problem Set 3.

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.

```
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
```

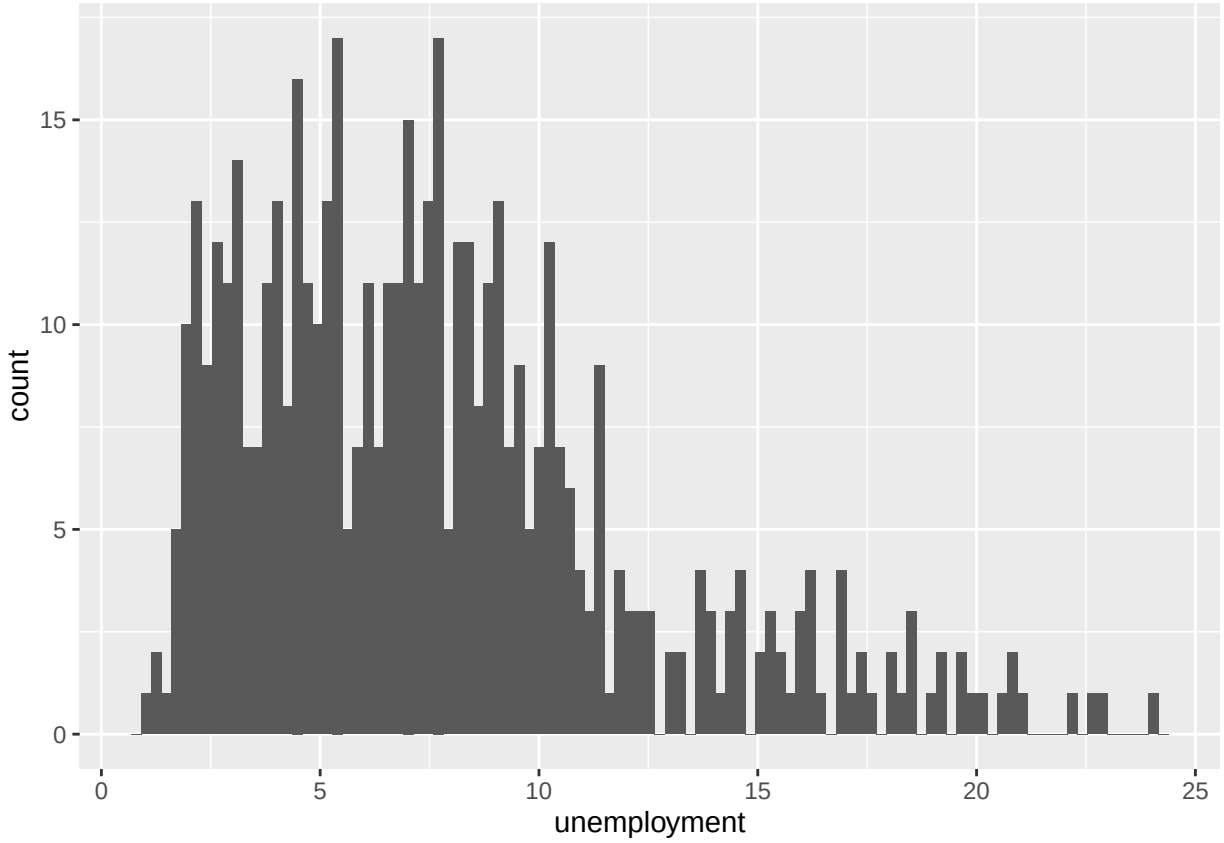
```
names(unm) <- c("country",1981:2005,"dummy")
```

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))
```

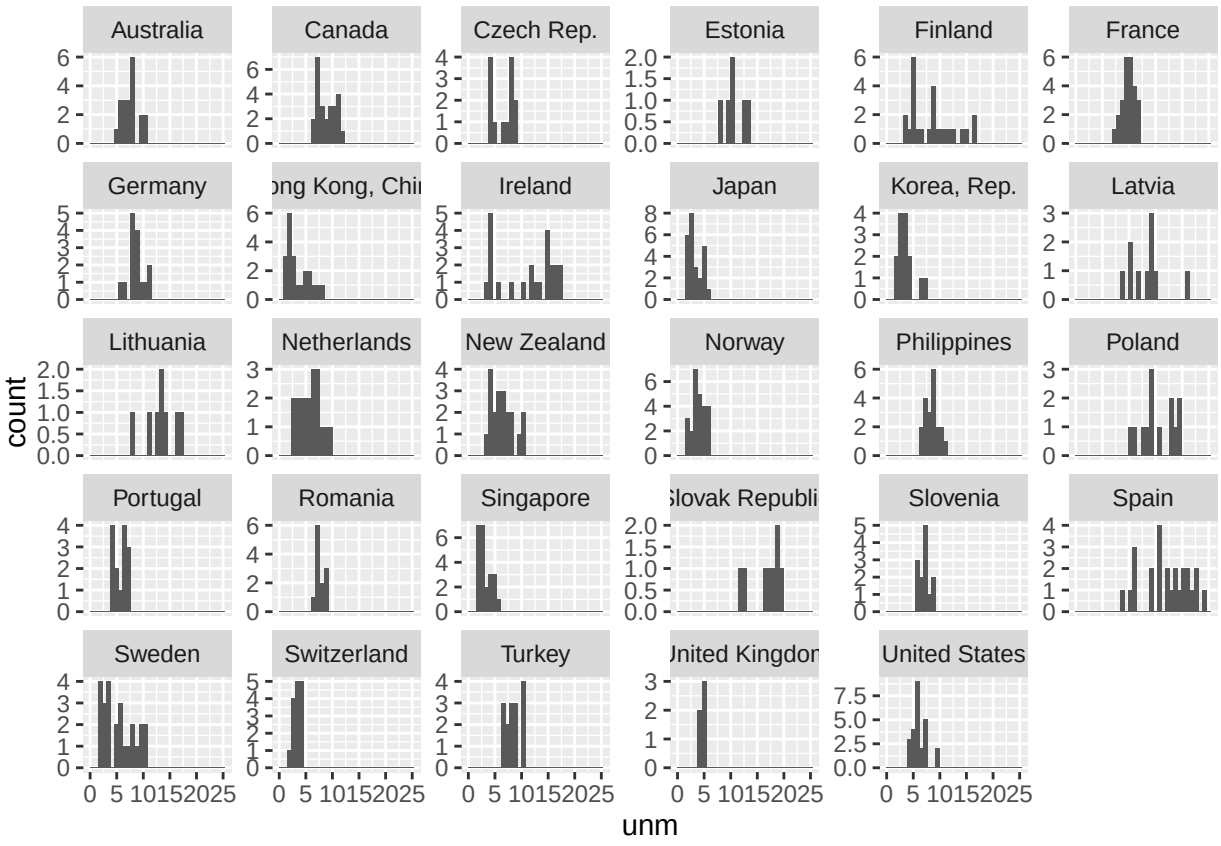
Plot the histrigrams for unemployment rates

```
qplot(unmg$unm, geom="histogram", xlab = "unemployment", bins=100)
```

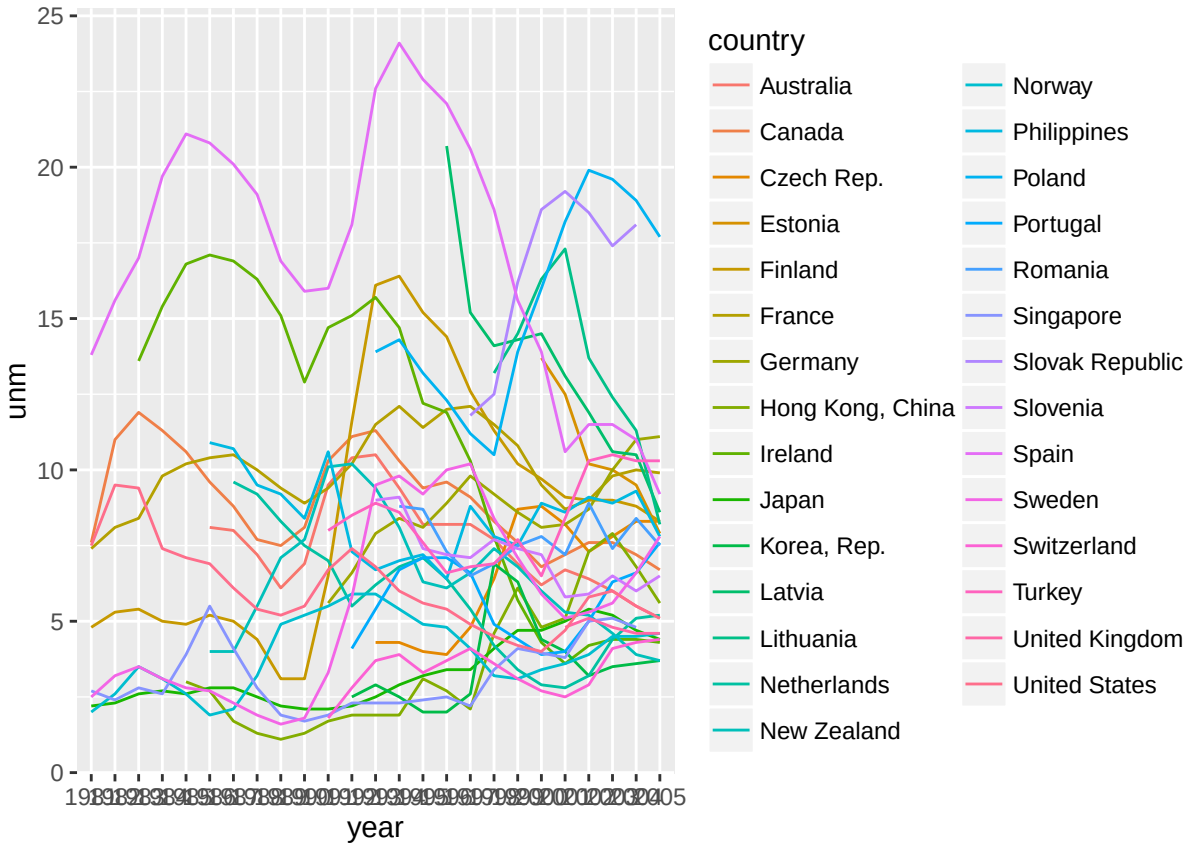



```
qplot(x=unm, data=unmg) + facet_wrap(~country, scales="free_y")
```

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

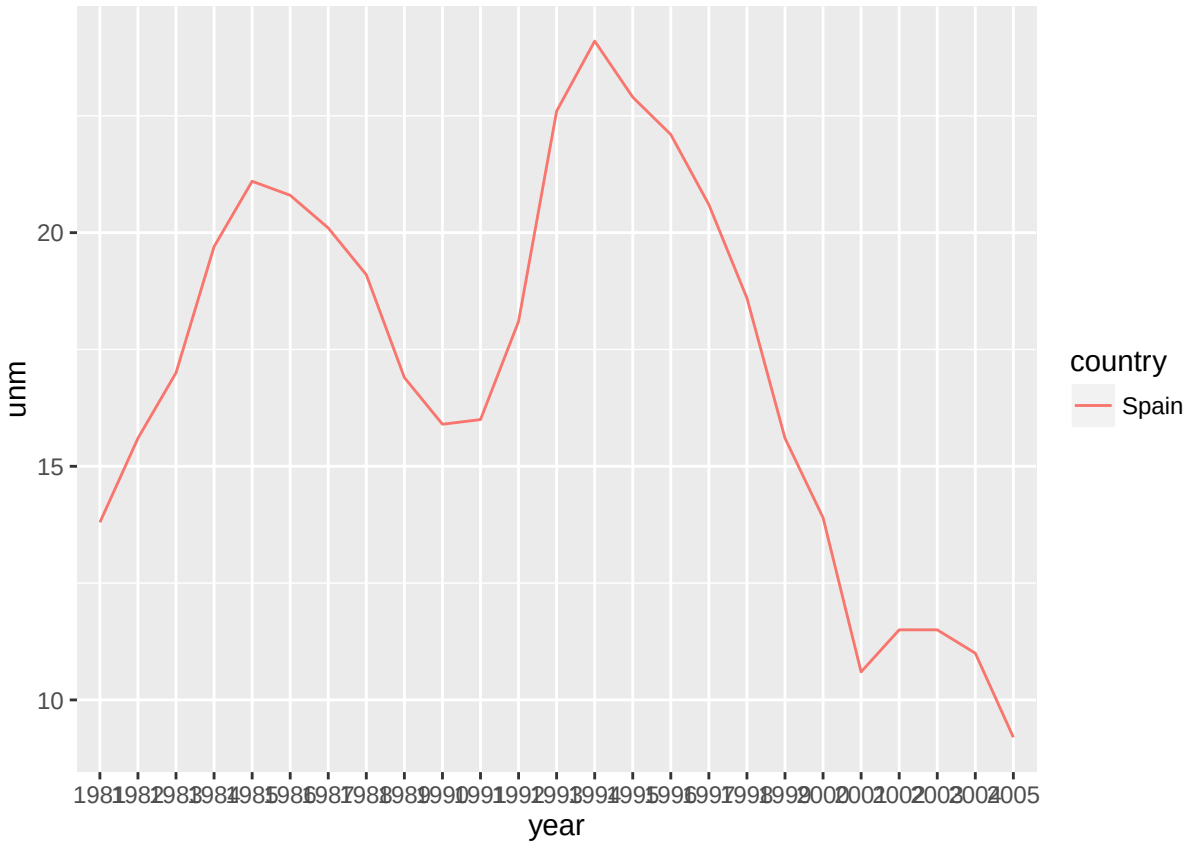



```
ggplot() + geom_line(data=unmg, aes(x=year,y=unm,group=country, color=country))
```

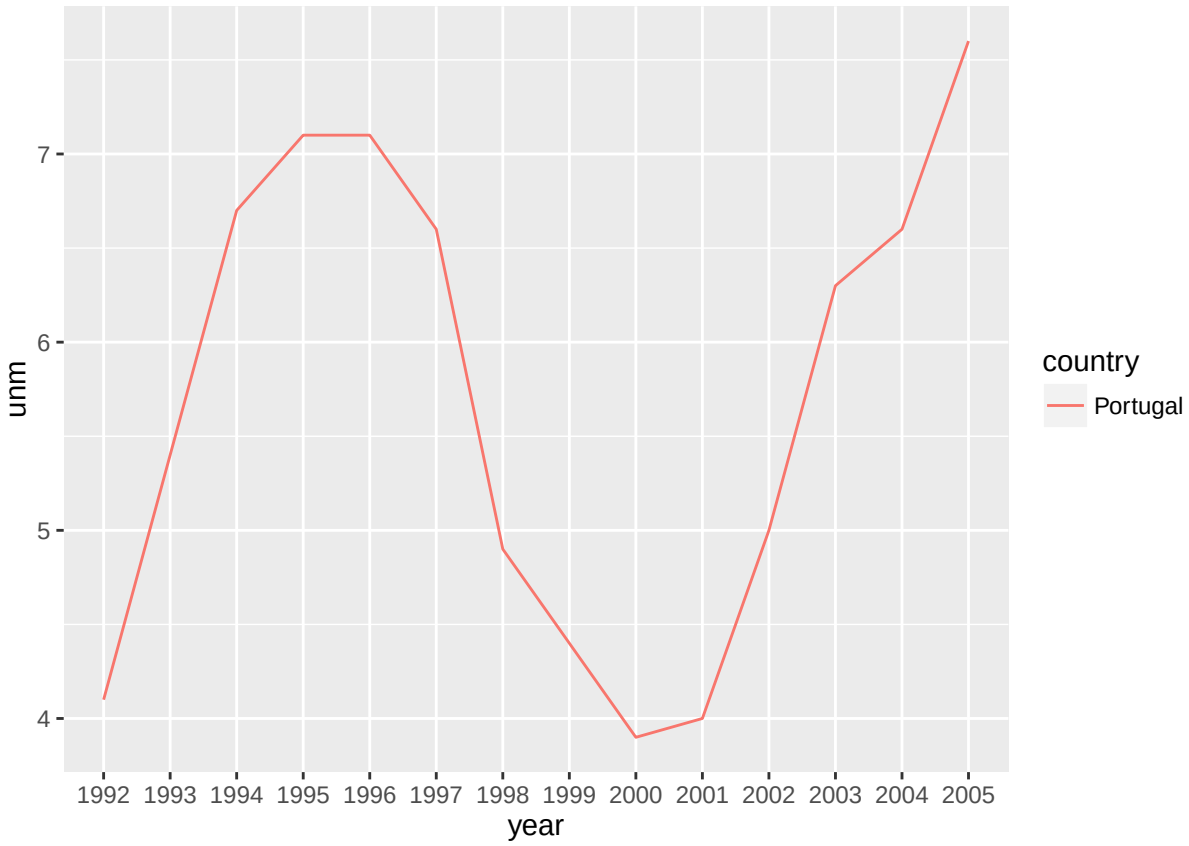


Check the plots for few countries

```
ggplot() + geom_line(data=filter(unmg, country=="Spain"), aes(x=year,y=unm,group=country, color=country)
```




```
ggplot() + geom_line(data=filter(unmg, country=="Portugal"), aes(x=year,y=unm,group=country, color=coun
```


unm
7
6
5
4
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
year
country
Portugal