Lesson4:Problem Set

Quiz1

```
library(ggplot2)
data(diamonds)
names(diamonds)
```

```
##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"
##  [8] "x"       "y"       "z"
```

```
summary(diamonds)
```

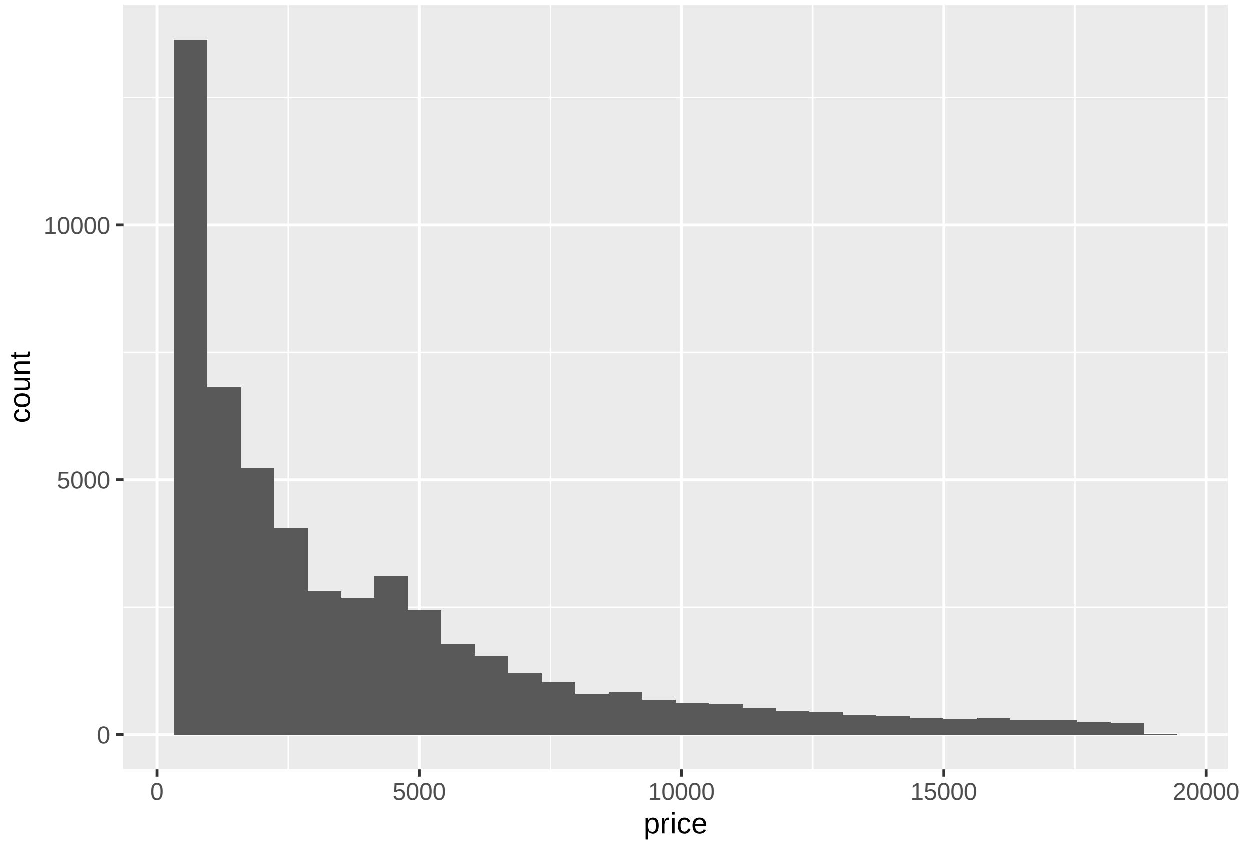
```
##      carat               cut        color       clarity         depth
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065   Min.   :43.00
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258   1st Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :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
##
```

```
?diamonds
```

Quiz2 Price Histogram

```
# Create a histogram of the price of
# all the diamonds in the diamond data set.

# TYPE YOUR CODE BELOW THE LINE
# =======================================
ggplot(aes(x=price),data=diamonds)+geom_histogram()
```

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



#Quiz3 Price Histogram Summary

```
# The distribution is tailed.
summary(diamonds$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     950    2401    3933    5324   18823
```

#Quiz4 Diamond Counts

```
price=subset(diamonds,diamonds$price< 500)
price2=subset(diamonds,diamonds$price< 250)
price3=subset(diamonds,diamonds$price>= 15000)
summary(price)
```

```
##      carat               cut      color      clarity        depth
##  Min.   :0.2000   Fair     :  7   D:156   SI1    :475   Min.   :55.20
##  1st Qu.:0.2600   Good     :226   E:266   VS2    :377   1st Qu.:61.20
##  Median :0.3000   Very Good:653   F:207   VS1    :326   Median :61.90
##  Mean   :0.2903   Premium  :215   G:272   SI2    :302   Mean   :61.81
##  3rd Qu.:0.3100   Ideal    :628   H:360   VVS2   :133   3rd Qu.:62.50
##  Max.   :0.4300                   I:310   VVS1   : 95   Max.   :66.40
##                                   J:158   (Other): 21
##      table           price             x               y               z
##  Min.   :44.00   Min.   :326.0   Min.   :3.730   Min.   :3.680   Min.   :2.24
##  1st Qu.:55.00   1st Qu.:421.0   1st Qu.:4.100   1st Qu.:4.130   1st Qu.:2.54
##  Median :57.00   Median :450.0   Median :4.280   Median :4.310   Median :2.66
##  Mean   :57.15   Mean   :444.8   Mean   :4.239   Mean   :4.269   Mean   :2.63
##  3rd Qu.:59.00   3rd Qu.:477.0   3rd Qu.:4.360   3rd Qu.:4.390   3rd Qu.:2.71
##  Max.   :66.00   Max.   :499.0   Max.   :4.780   Max.   :6.020   Max.   :4.44
##
```

```
summary(price2)
```

```
##      carat            cut    color    clarity      depth         table
##  Min.   : NA   Fair     :0   D:0   I1     :0   Min.   : NA   Min.   : NA
##  1st Qu.: NA   Good     :0   E:0   SI2    :0   1st Qu.: NA   1st Qu.: NA
##  Median : NA   Very Good:0   F:0   SI1    :0   Median : NA   Median : NA
##  Mean   :NaN   Premium  :0   G:0   VS2    :0   Mean   :NaN   Mean   :NaN
##  3rd Qu.: NA   Ideal    :0   H:0   VS1    :0   3rd Qu.: NA   3rd Qu.: NA
##  Max.   : NA                 I:0   VVS2   :0   Max.   : NA   Max.   : NA
##                              J:0   (Other):0
##      price           x             y             z
##  Min.   : NA   Min.   : NA   Min.   : NA   Min.   : NA
##  1st Qu.: NA   1st Qu.: NA   1st Qu.: NA   1st Qu.: NA
##  Median : NA   Median : NA   Median : NA   Median : NA
##  Mean   :NaN   Mean   :NaN   Mean   :NaN   Mean   :NaN
##  3rd Qu.: NA   3rd Qu.: NA   3rd Qu.: NA   3rd Qu.: NA
##  Max.   : NA   Max.   : NA   Max.   : NA   Max.   : NA
##
```

```
summary(price3)
```

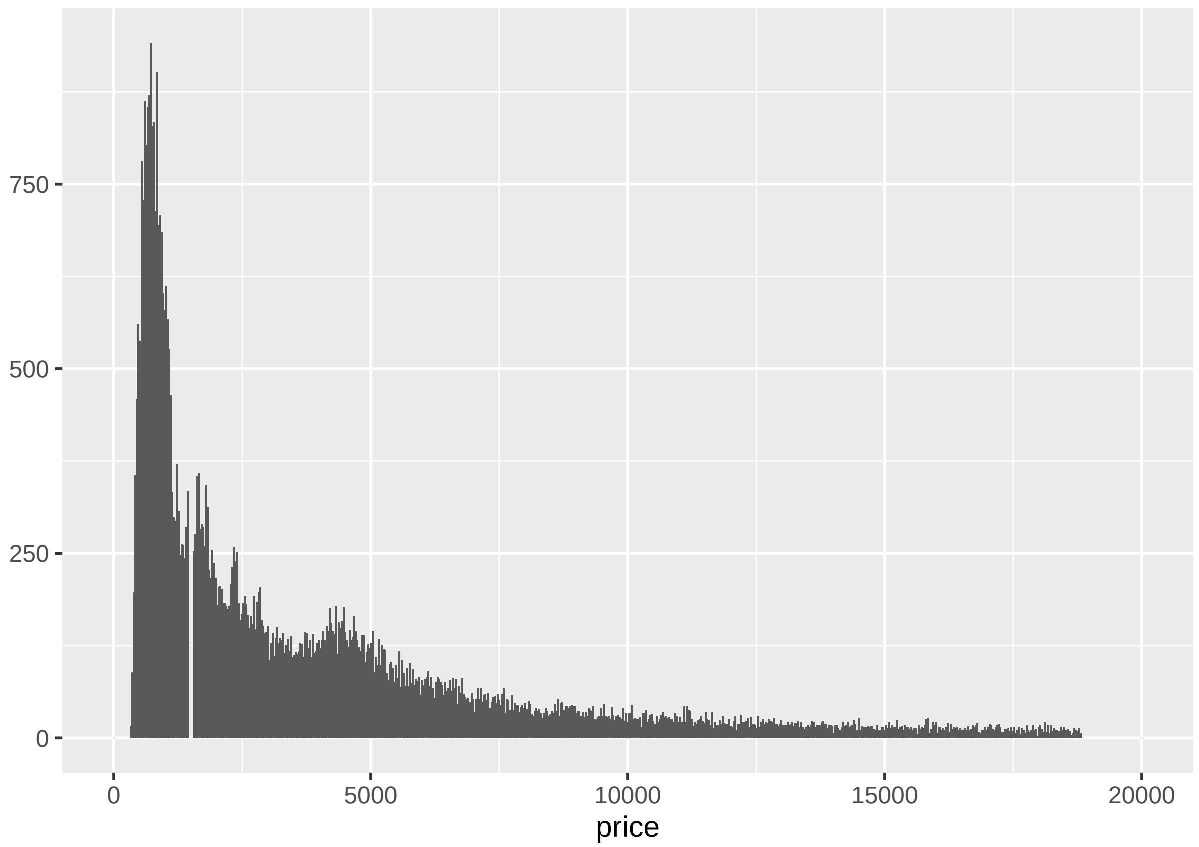
```
##      carat              cut      color      clarity        depth
##  Min.   :1.000   Fair     : 41   D:120   SI2    :518   Min.   :56.20
##  1st Qu.:1.720   Good     :129   E:161   SI1    :364   1st Qu.:60.70
##  Median :2.010   Very Good:367   F:232   VS2    :359   Median :61.80
##  Mean   :1.978   Premium  :587   G:334   VS1    :227   Mean   :61.61
##  3rd Qu.:2.120   Ideal    :532   H:319   VVS2   : 78   3rd Qu.:62.50
##  Max.   :5.010                   I:369   IF     : 51   Max.   :70.60
##                                  J:121   (Other): 59
##      table           price             x                y
##  Min.   :51.00   Min.   :15000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.85   1st Qu.:15835   1st Qu.: 7.720   1st Qu.: 7.690
##  Median :58.00   Median :16733   Median : 8.100   Median : 8.100
##  Mean   :58.02   Mean   :16783   Mean   : 8.013   Mean   : 8.005
##  3rd Qu.:59.00   3rd Qu.:17725   3rd Qu.: 8.290   3rd Qu.: 8.290
##  Max.   :69.00   Max.   :18823   Max.   :10.740   Max.   :10.540
```

```
##
##        z
##  Min.   :0.000
##  1st Qu.:4.750
##  Median :4.990
##  Mean   :4.922
##  3rd Qu.:5.090
##  Max.   :6.980
##
```

#Quiz5 Cheaper Diamonds

```r
# Explore the largest peak in the
# price histogram you created earlier.

# Try limiting the x-axis, altering the bin width,
# and setting different breaks on the x-axis.

# There won't be a solution video for this
# question so go to the discussions to
# share your thoughts and discover
# what other people find.

# You can save images by using the ggsave() command.
# ggsave() will save the last plot created.
# For example...
#                  qplot(x = price, data = diamonds)
#                  ggsave('priceHistogram.png')

# ggsave currently recognises the extensions eps/ps, tex (pictex),
# pdf, jpeg, tiff, png, bmp, svg and wmf (windows only).

# Submit your final code when you are ready.

# TYPE YOUR CODE BELOW THE LINE
# ======================================================================
qplot(x=price,data=diamonds,binwidth=30) +
         scale_x_continuous(limits=c(0,20000))
```

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



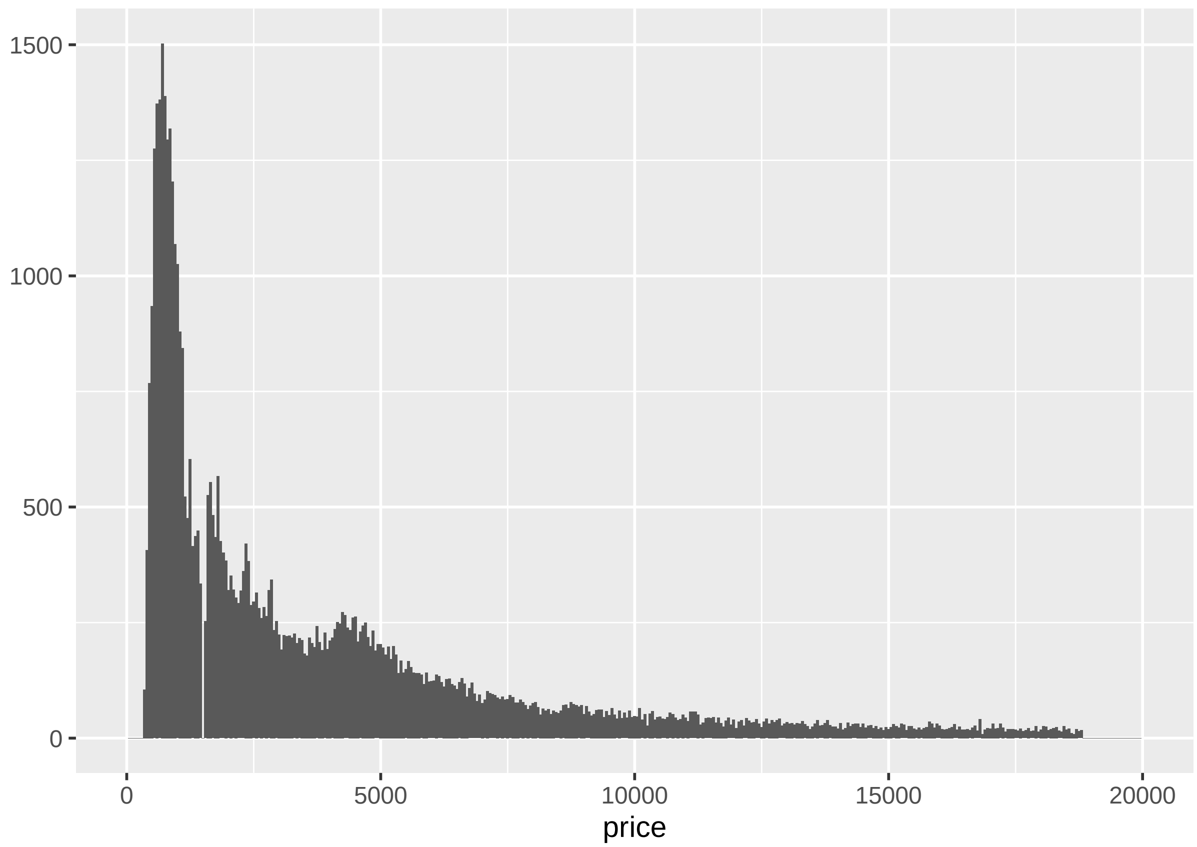
```
ggsave('priceHistogram.png')
```

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

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

```
qplot(x=price,data=diamonds,binwidth=50) +
          scale_x_continuous(limits=c(0,20000))
```

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



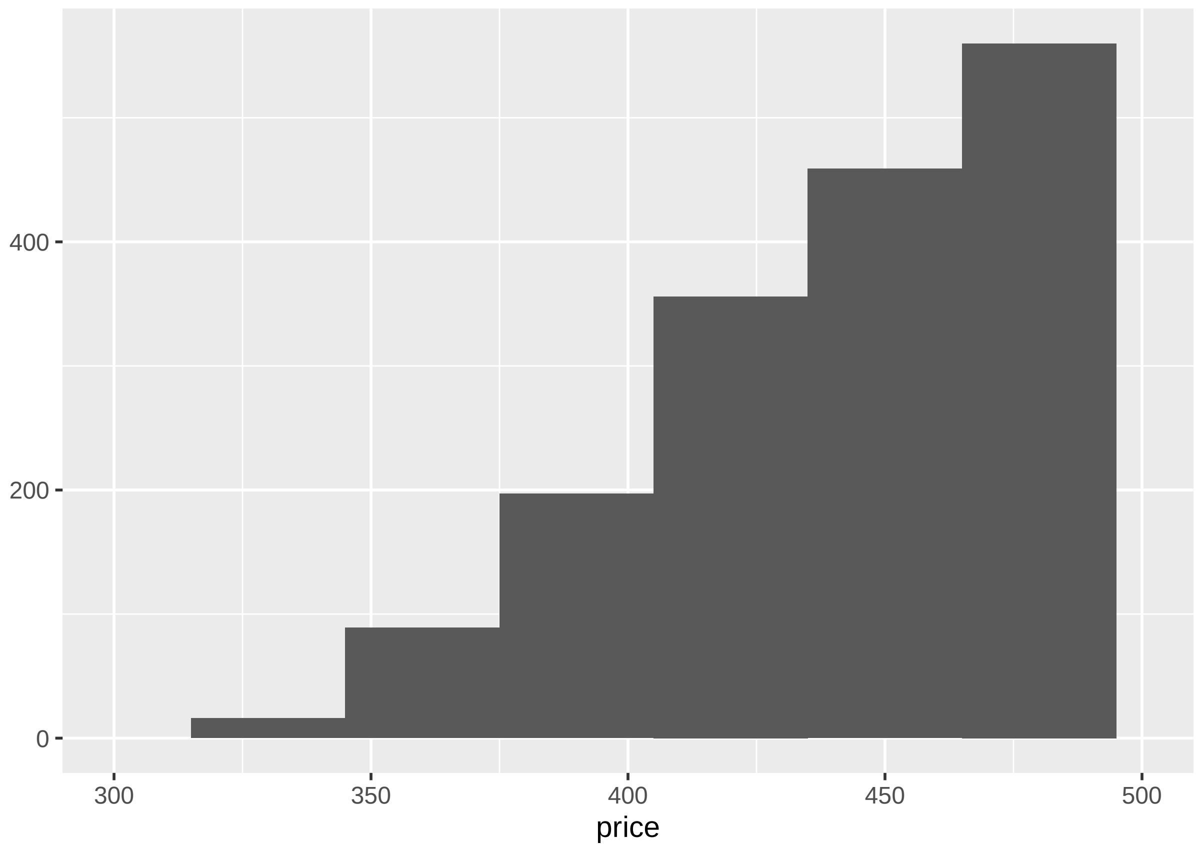
```
ggsave('priceHistogram2.png')
```

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

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

```
qplot(x=price,data=price,binwidth=30) +
        scale_x_continuous(limits=c(300,500))
```

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



```
ggsave('priceHistogram.png')
```

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

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

#Quiz6 Price by Cut Histograms

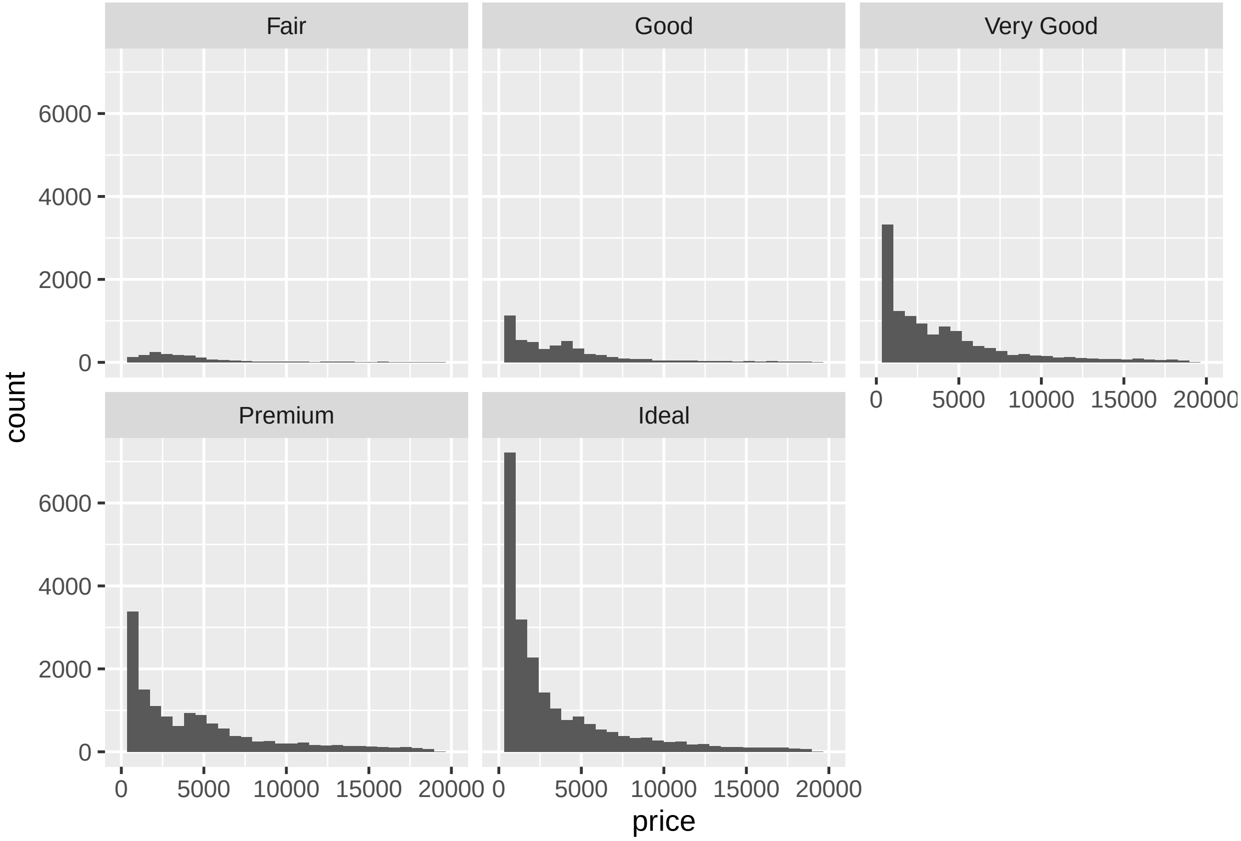
```
# Break out the histogram of diamond prices by cut.

# You should have five histograms in separate
# panels on your resulting plot.

# TYPE YOUR CODE BELOW THE LINE
# ======================================================
names(diamonds)
```

```
##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"
##  [8] "x"       "y"       "z"
```

```
ggplot(aes(x=price),data=diamonds)+geom_histogram()+
                  scale_x_continuous(limits = c(10,20000))+
                  facet_wrap(~cut)
```

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

```
## Warning: Removed 10 rows containing missing values (geom_bar).
```



Do you think the distributions look the same or different? response: No

Quiz7: Price by Cut

Which cut has the highest price diamond?

Which cut has the lowest priced diamond?

Which cut has the lowest median price?

```
by(diamonds$price,diamonds$cut,summary)
```

```
## diamonds$cut: Fair
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     337    2050    3282    4359    5206   18574 
## -------------------------------------------------------------- 
## diamonds$cut: Good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     327    1145    3050    3929    5028   18788 
## -------------------------------------------------------------- 
## diamonds$cut: Very Good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     336     912    2648    3982    5373   18818
## --------------------------------------------------------------
## diamonds$cut: Premium
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326    1046    3185    4584    6296   18823
## --------------------------------------------------------------
## diamonds$cut: Ideal
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     878    1810    3458    4678   18806
```

Which cut has the highest price diamond? response:Premium,Max is 18823

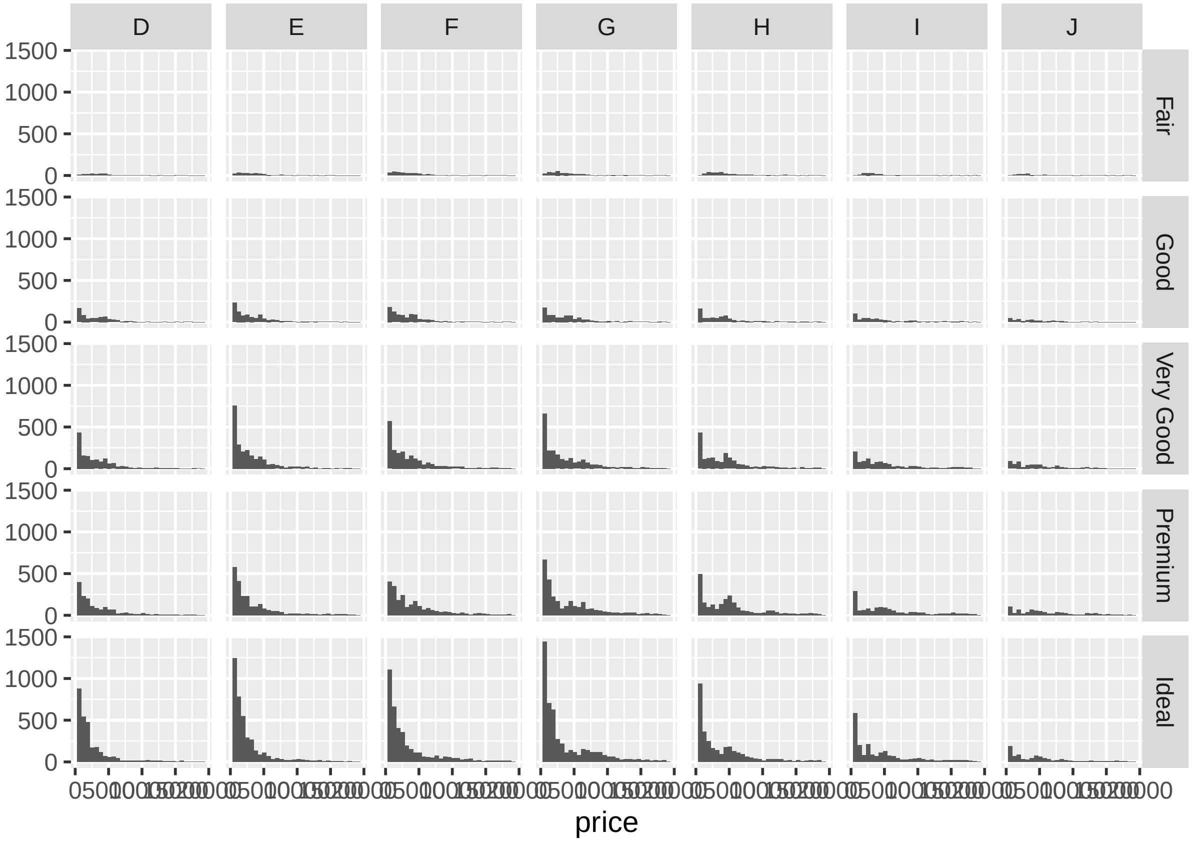
Which cut has the lowest priced diamond? response:Premium and Ideal,Min is 326

Which cut has the lowest median price? response:Ideal, Median is 1810

Quiz8:Scales and Multiple Histograms

```
# Look up the documentation for facet_wrap in R Studio.
# Then, scroll back up and add a parameter to facet_wrap so that
# the y-axis in the histograms is not fixed. You want the y-axis to
# be different for each histogram.

qplot(x = price, data = diamonds) + facet_grid(cut ~ color)
```

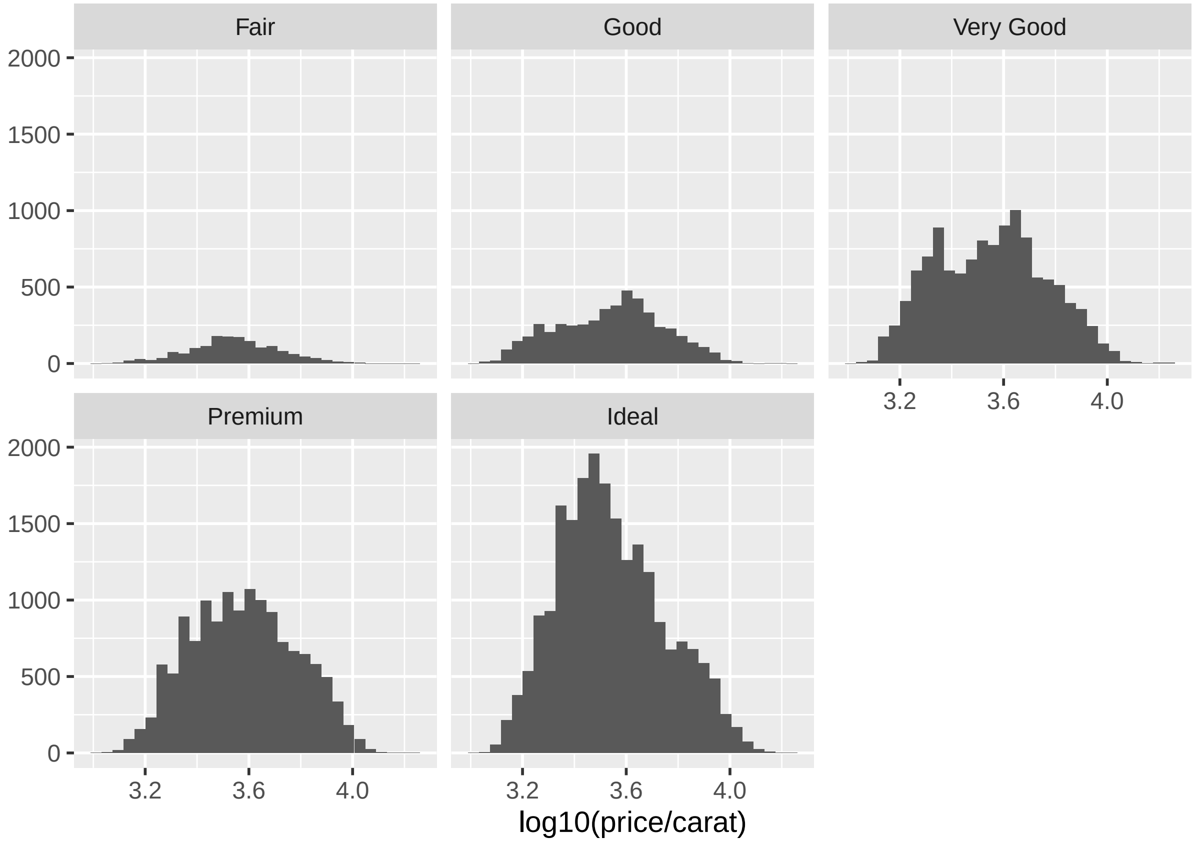
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



#Quiz9:Price per Carat by Cut

```
# Create a histogram of price per carat
# and facet it by cut. You can make adjustments
# to the code from the previous exercise to get
# started.

# Adjust the bin width and transform the scale
# of the x-axis using log10.

#Hint 1: You use the price and carat variables in the parameter for x. # What expression gives you pric

#Hint 2: For long tailed distributions, you can add a ggplot layer such #as scale_x_log10() to transfor

qplot(x=log10(price/carat),data=diamonds) +
  facet_wrap(~cut)
```

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

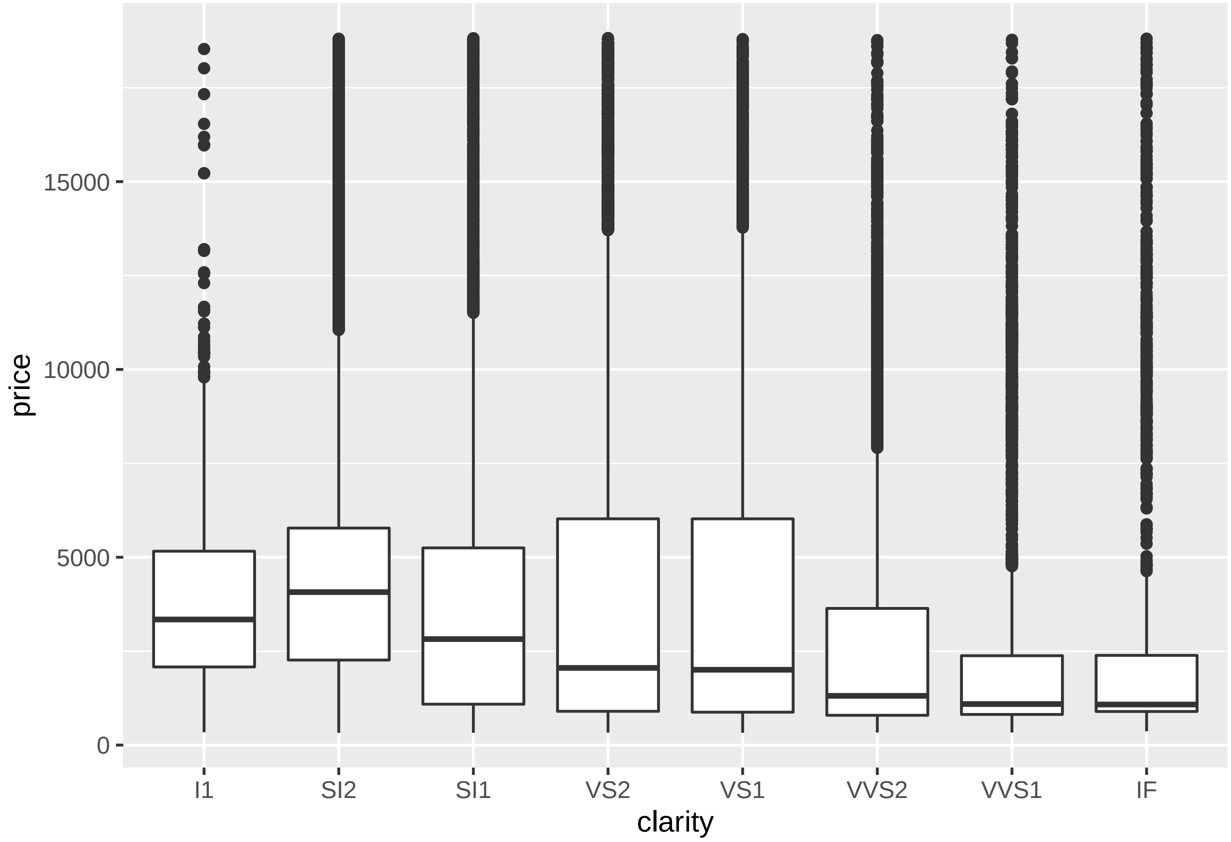


Quiz10:Price Box Plots}

```
# Investigate the price of diamonds using box plots,
# numerical summaries, and one of the following categorical
# variables: cut, clarity, or color.

# There won't be a solution video for this
# exercise so go to the discussion thread for either
# BOXPLOTS BY CLARITY, BOXPLOT BY COLOR, or BOXPLOTS BY CUT
# to share you thoughts and to
# see what other people found.

# You can save images by using the ggsave() command.
# ggsave() will save the last plot created.
# For example...
#                   qplot(x = price, data = diamonds)
#                   ggsave('priceHistogram.png')

# ggsave currently recognises the extensions eps/ps, tex (pictex),
# pdf, jpeg, tiff, png, bmp, svg and wmf (windows only).

# Copy and paste all of the code that you used for
# your investigation, and submit it when you are ready.
# ============================================================
summary(diamonds$clarity)
```

```
##    I1   SI2   SI1   VS2   VS1  VVS2  VVS1    IF 
##   741  9194 13065 12258  8171  5066  3655  1790
```

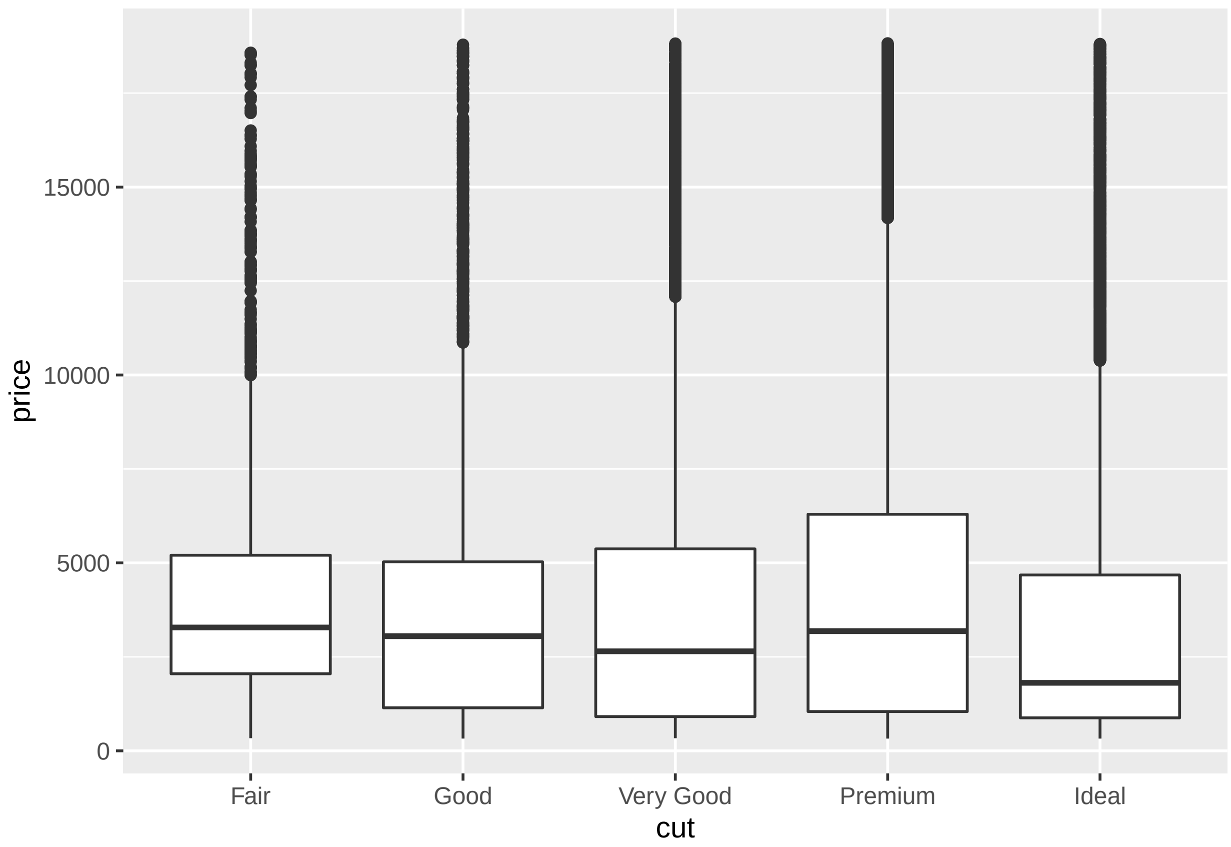
```
qplot(x=clarity,y=price, data=diamonds,geom = 'boxplot')
```



```
ggsave('clarityBox.png')
```

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

```
qplot(x=cut,y=price,data=diamonds,geom='boxplot')
```

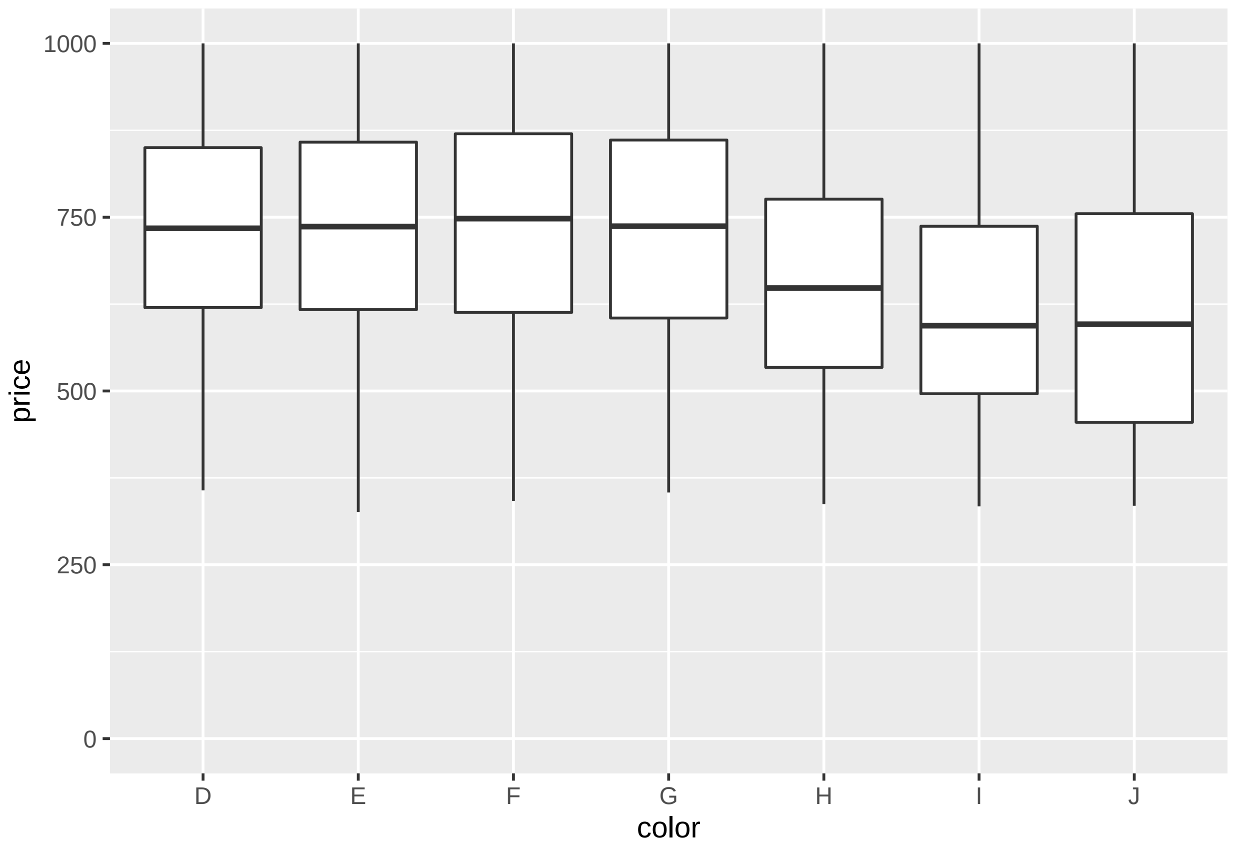


```
ggsave('cutBox.png')
```

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

```
qplot(x=color,y=price,data=diamonds,geom='boxplot')+
scale_y_continuous(limits = c(0,1000))
```

```
## Warning: Removed 39416 rows containing non-finite values (stat_boxplot).
```



```r
ggsave('colorBox.png')
```

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

```
## Warning: Removed 39416 rows containing non-finite values (stat_boxplot).
```

Quiz11:Interquartile Range - IQR}

```r
summary(diamonds$color)
```

```
##     D     E     F     G     H     I     J 
##  6775  9797  9542 11292  8304  5422  2808
```

```r
by(diamonds$price,diamonds$color,summary)
```

```
## diamonds$color: D
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     357     911    1838    3170    4214   18693 
## -------------------------------------------------------------- 
## diamonds$color: E
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     882    1739    3077    4003   18731
## -------------------------------------------------------------
## diamonds$color: F
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     342     982    2344    3725    4868   18791
## -------------------------------------------------------------
## diamonds$color: G
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     354     931    2242    3999    6048   18818
## -------------------------------------------------------------
## diamonds$color: H
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     337     984    3460    4487    5980   18803
## -------------------------------------------------------------
## diamonds$color: I
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1120    3730    5092    7202   18823
## -------------------------------------------------------------
## diamonds$color: J
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     335    1860    4234    5324    7695   18710
```

```
IQR(subset(diamonds, price <1000)$price)
```

```
## [1] 261
```

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

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

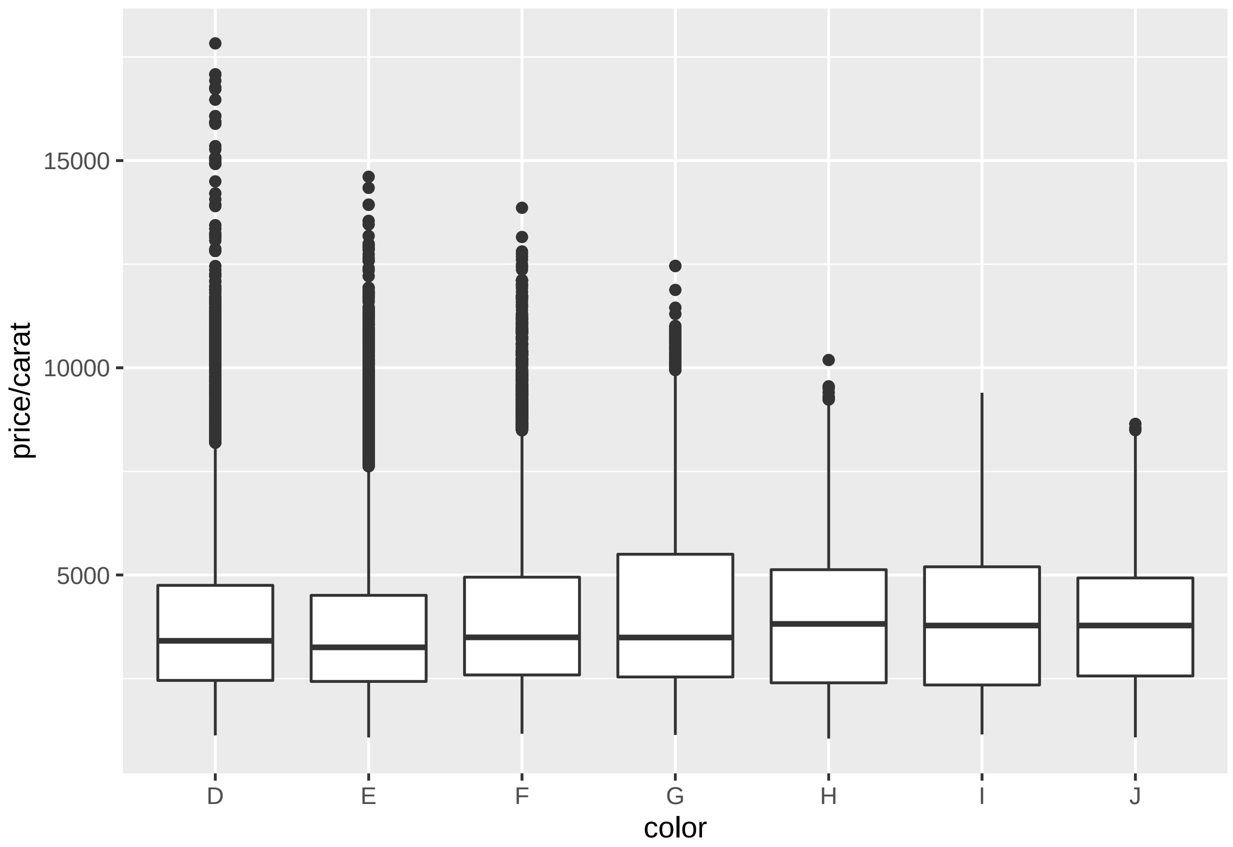
```
IQR(subset(diamonds, color=='J')$price)
```

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

a.What is the price range for the middle 50% of diamonds with color D? response:1st Qu is 911;3rd Qu is 4214 b.What is the price range for the middle 50% of diamonds with color J? response:1st Qu is 1860;3rd Qu is 7695 c.What is the IQR for diamonds with the best color? response:IQR-D is 3302.5 d.What is the IQR for diamonds with the worstco response:IQR-J is 5834.5.c

Quiz12:Price per Carat Box Plots by Color

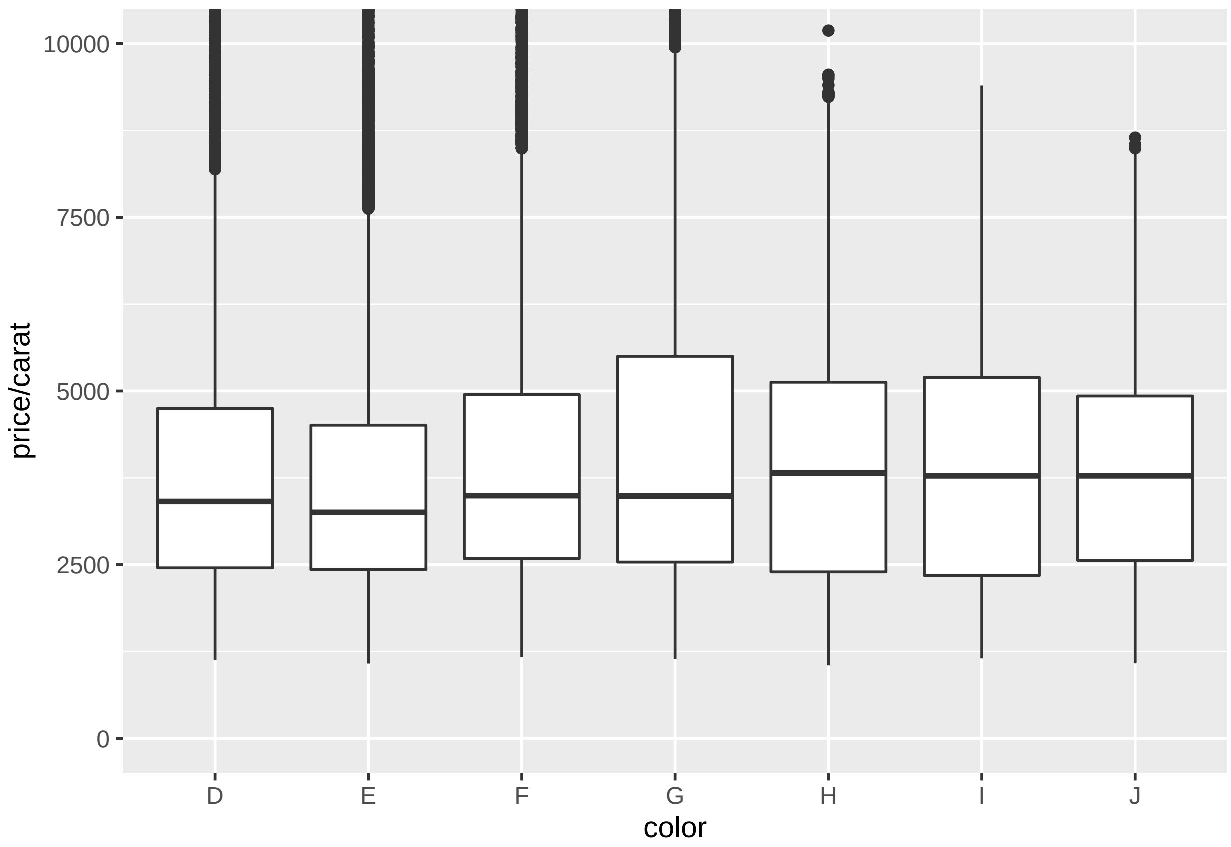
```
# Investigate the price per carat of diamonds across
# the different colors of diamonds using boxplots.

# SUBMIT YOUR CODE BELOW THIS LINE
# ===================================================================
qplot(x=color, y=price/carat, data=diamonds, geom='boxplot')
```



```
ggsave('boxplot_t.png')
```

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

```
qplot(x=color, y=price/carat, data=diamonds, geom='boxplot')+
  coord_cartesian(ylim = c(0,10000))
```



```
ggsave('boxplot_2.pdf')
```

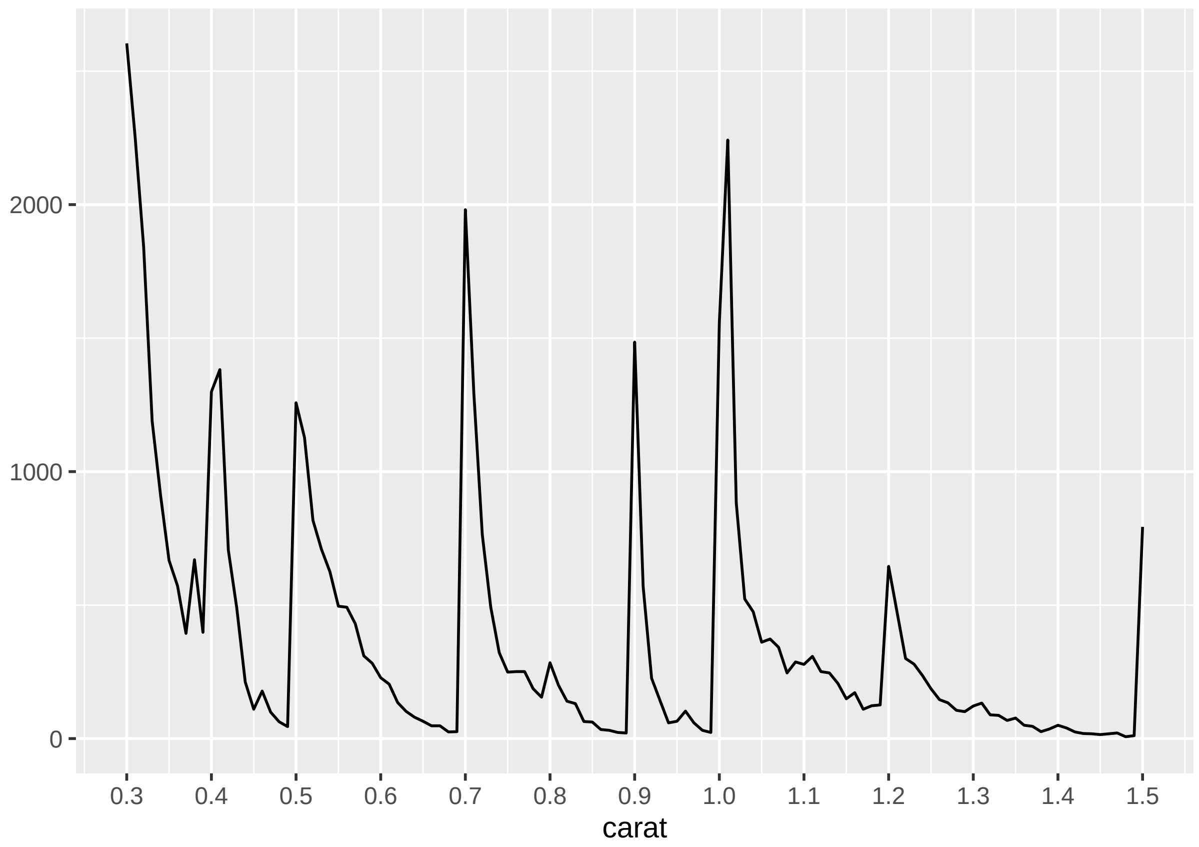
```
## Saving 6.5 x 4.5 in image
```

#Quiz13: Carat Frequenct Polygon Investigate the weight of the diamonds(carat)using a frequency polygon.Use different bin difths to see how the frequency polygon changes. what carat size has a count greater than 2000? – 0.3 and 1.01.

```
#ggplot(diamonds,aes(carat))+geom_freqpoly(binwidth=0.1)
qplot(x=carat,data=diamonds,
          binwidth=0.01,
          geom='freqpoly')+
          scale_x_continuous(lim=c(0.3,1.5),breaks=seq(0.3,1.5,0.1))
```

```
## Warning: Removed 7041 rows containing non-finite values (stat_bin).
```

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



```
ggsave('caratFreq.png')
```

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

```
## Warning: Removed 7041 rows containing non-finite values (stat_bin).
```

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

Data Wrangling with R

```
#install.packages("tidyr")
#library(tidyr)

#install.packages('dplyr')
#library(dplyr)
```

Quiz15:Gapminder Data

```r
# The Gapminder website contains over 500 data sets with information about
# the world's population. Your task is to download a data set of your choice
# and create 2-5 plots that make use of the techniques from Lesson 3.

# You might use a simple histogram, a boxplot split over a categorical variable,
# or a frequency polygon. The choice is yours!

# You can find a link to the Gapminder website in the Instructor Notes.

# Once you've completed your investigation, create a post in the discussions that includes:
#       1. any questions you answered, your observations, and summary statistics
#       2. snippets of code that created the plots
#       3. links to the images of your plots

# You can save images by using the ggsave() command.
# ggsave() will save the last plot created.
# For example...
#                  qplot(x = price, data = diamonds)
#                  ggsave('priceHistogram.png')

# ggsave currently recognises the extensions eps/ps, tex (pictex),
# pdf, jpeg, tiff, png, bmp, svg and wmf (windows only).

# Copy and paste all of the code that you used for
# your investigation, and submit it when you are ready.
# ====================================================================================
education= read.csv('expenditure_per_student_primary_percent_of_gdp_per_person.csv')
names(education)
```

```
##  [1] "country" "X1995"   "X1996"   "X1997"   "X1998"   "X1999"   "X2000"  
##  [8] "X2001"   "X2002"   "X2003"   "X2004"   "X2005"   "X2006"   "X2007"  
## [15] "X2008"   "X2009"   "X2010"   "X2011"   "X2012"   "X2013"   "X2014"  
## [22] "X2015"   "X2016"   "X2017"
```

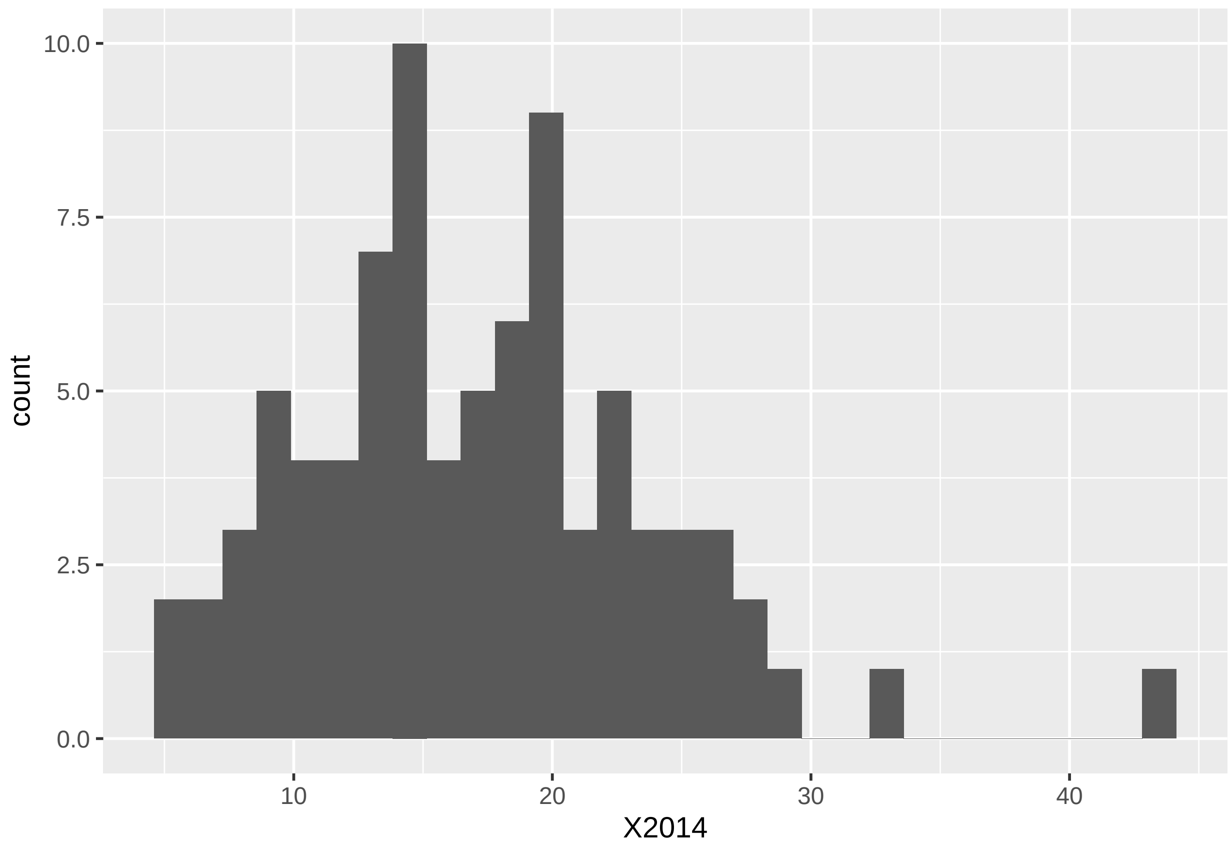
```r
education2=subset(education, !is.na(X2014))
summary(education)
```

```
##                  country        X1995       X1996             X1997      
##  Afghanistan        :  1   Min.   :15.6   Mode:logical   Min.   : 3.02  
##  Albania            :  1   1st Qu.:15.6   NA's:159       1st Qu.:10.13  
##  Algeria            :  1   Median :15.6                  Median :15.30  
##  Andorra            :  1   Mean   :15.6                  Mean   :18.78  
##  Antigua and Barbuda:  1   3rd Qu.:15.6                  3rd Qu.:19.20  
##  Argentina          :  1   Max.   :15.6                  Max.   :65.10  
##  (Other)            :153   NA's   :158                   NA's   :139    
##      X1998           X1999           X2000           X2001      
##  Min.   : 3.28   Min.   : 3.24   Min.   : 2.85   Min.   : 4.68  
##  1st Qu.: 9.37   1st Qu.:10.70   1st Qu.:10.60   1st Qu.:11.00  
##  Median :12.70   Median :14.40   Median :13.30   Median :14.60  
##  Mean   :14.05   Mean   :15.04   Mean   :14.37   Mean   :15.16  
##  3rd Qu.:17.20   3rd Qu.:18.85   3rd Qu.:18.90   3rd Qu.:19.30  
##  Max.   :41.80   Max.   :30.70   Max.   :28.30   Max.   :28.90  
##  NA's   :110     NA's   :101     NA's   :94      NA's   :94  
##      X2002           X2003           X2004            X2005      
##  Min.   : 5.12   Min.   : 5.12   Min.   : 3.480   Min.   : 3.11  
##  1st Qu.:10.40   1st Qu.:11.47   1st Qu.: 9.925   1st Qu.:10.20  
##  Median :14.50   Median :16.10   Median :13.500   Median :14.65  
##  Mean   :14.95   Mean   :15.65   Mean   :14.507   Mean   :15.16  
##  3rd Qu.:19.20   3rd Qu.:19.30   3rd Qu.:19.225   3rd Qu.:19.15  
##  Max.   :37.30   Max.   :25.70   Max.   :40.500   Max.   :40.30  
##  NA's   :86      NA's   :95      NA's   :83       NA's   :83     
##      X2006           X2007           X2008           X2009      
##  Min.   : 5.51   Min.   : 5.44   Min.   : 4.25   Min.   : 3.98  
##  1st Qu.:10.10   1st Qu.:10.80   1st Qu.:11.35   1st Qu.:10.68  
##  Median :14.90   Median :15.25   Median :15.70   Median :16.15  
##  Mean   :15.68   Mean   :16.21   Mean   :16.69   Mean   :16.92  
##  3rd Qu.:20.70   3rd Qu.:19.75   3rd Qu.:19.85   3rd Qu.:21.15  
##  Max.   :33.70   Max.   :53.80   Max.   :56.40   Max.   :58.10  
##  NA's   :88      NA's   :81      NA's   :69      NA's   :67     
##      X2010           X2011           X2012           X2013      
##  Min.   : 2.79   Min.   : 3.63   Min.   : 4.03   Min.   : 4.34  
##  1st Qu.:10.15   1st Qu.: 9.79   1st Qu.:10.20   1st Qu.:11.35  
##  Median :16.50   Median :16.00   Median :15.20   Median :16.20  
##  Mean   :16.62   Mean   :16.20   Mean   :15.86   Mean   :16.52  
##  3rd Qu.:21.75   3rd Qu.:20.80   3rd Qu.:20.60   3rd Qu.:20.90  
##  Max.   :54.20   Max.   :51.00   Max.   :38.90   Max.   :36.00  
##  NA's   :59      NA's   :54      NA's   :68      NA's   :65     
##      X2014           X2015              X2016            X2017    
##  Min.   : 5.30   Min.   : 0.0186   Min.   : 0.295   Min.   :7.7  
##  1st Qu.:12.60   1st Qu.:10.9500   1st Qu.: 9.402   1st Qu.:7.7  
##  Median :16.70   Median :13.6000   Median :14.200   Median :7.7  
##  Mean   :17.07   Mean   :15.7781   Mean   :15.223   Mean   :7.7  
##  3rd Qu.:21.00   3rd Qu.:17.7250   3rd Qu.:17.300   3rd Qu.:7.7  
##  Max.   :43.50   Max.   :46.6000   Max.   :47.500   Max.   :7.7  
##  NA's   :76      NA's   :111       NA's   :125      NA's   :158
```

```r
ggplot(aes(x=X2014),data=education,binwidth=0.1)+geom_histogram()
```

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

```
## Warning: Removed 76 rows containing non-finite values (stat_bin).
```



```r
ggsave('X2014.png')
```

```
## Saving 6.5 x 4.5 in image
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 76 rows containing non-finite values (stat_bin).
```

Quiz16.Exploring Your Friends’ Birthdays

```r
# Your task is to investigate the distribution of your friends'
# birth months and days.

# Here some questions you could answer, and we hope you think of others.

# **********************************************************************

#Q: How many people have the same birthday as you?
# A:0
# Which month contains the most number of birthdays?
# A:March
# How many birthdays are in each month?
# A:1  2  3  4  5  6  7  8  9 10 11 12
#   89 79 98 81 72 93 86 91 96 89 87 72
# Which day of the year has the most number of birthdays?
# A:14
# Do you have at least 365 friends that have birthdays on everyday
# of the year?
#A:yes.
# **********************************************************************

# You will need to do some data munging and additional research to
# complete this task. This task won't be easy, and you may encounter some
# unexpected challenges along the way. We hope you learn a lot from it though.

# You can expect to spend 30 min or more on this task depending if you
# use the provided data or obtain your personal data. We also encourage you
# to use the lubridate package for working with dates. Read over the documentation
# in RStudio and search for examples online if you need help.

# You'll need to export your Facebooks friends' birthdays to a csv file.
# You may need to create a calendar of your Facebook friends’ birthdays
# in a program like Outlook or Gmail and then export the calendar as a
# csv file.

# Once you load the data into R Studio, you can use the strptime() function
# to extract the birth months and birth days. We recommend looking up the
# documentation for the function and finding examples online.

# We've included some links in the Instructor Notes to help get you started.

# Once you've completed your investigation, create a post in the discussions
# that includes:
#   1. any questions you answered, your observations, and summary statistics
#   2. snippets of code that created the plots
#   3. links to the images of your plots

# You can save images by using the ggsave() command.
# ggsave() will save the last plot created.
# For example...
#                  qplot(x = price, data = diamonds)
#                  ggsave('priceHistogram.png')

# ggsave currently recognises the extensions eps/ps, tex (pictex),
# pdf, jpeg, tiff, png, bmp, svg and wmf (windows only).

# Copy and paste all of the code that you used for
# your investigation below the line. Submit it when you are ready.
# ===============================================================================
birthday = read.csv('birthdaysExample.csv')
names(birthday)
```

```
## [1] "dates"
```

```
summary(birthday)
```

```
##      dates
##  2/6/14 :  8
##  5/22/14:  8
##  7/16/14:  8
##  1/14/14:  7
##  2/2/14 :  7
##  2/23/14:  7
##  (Other):988
```

```
subset(birthday,dates == '6/1/84')
```

```
## [1] dates
## <0 rows> (or 0-length row.names)
```

```
library(lubridate)
```

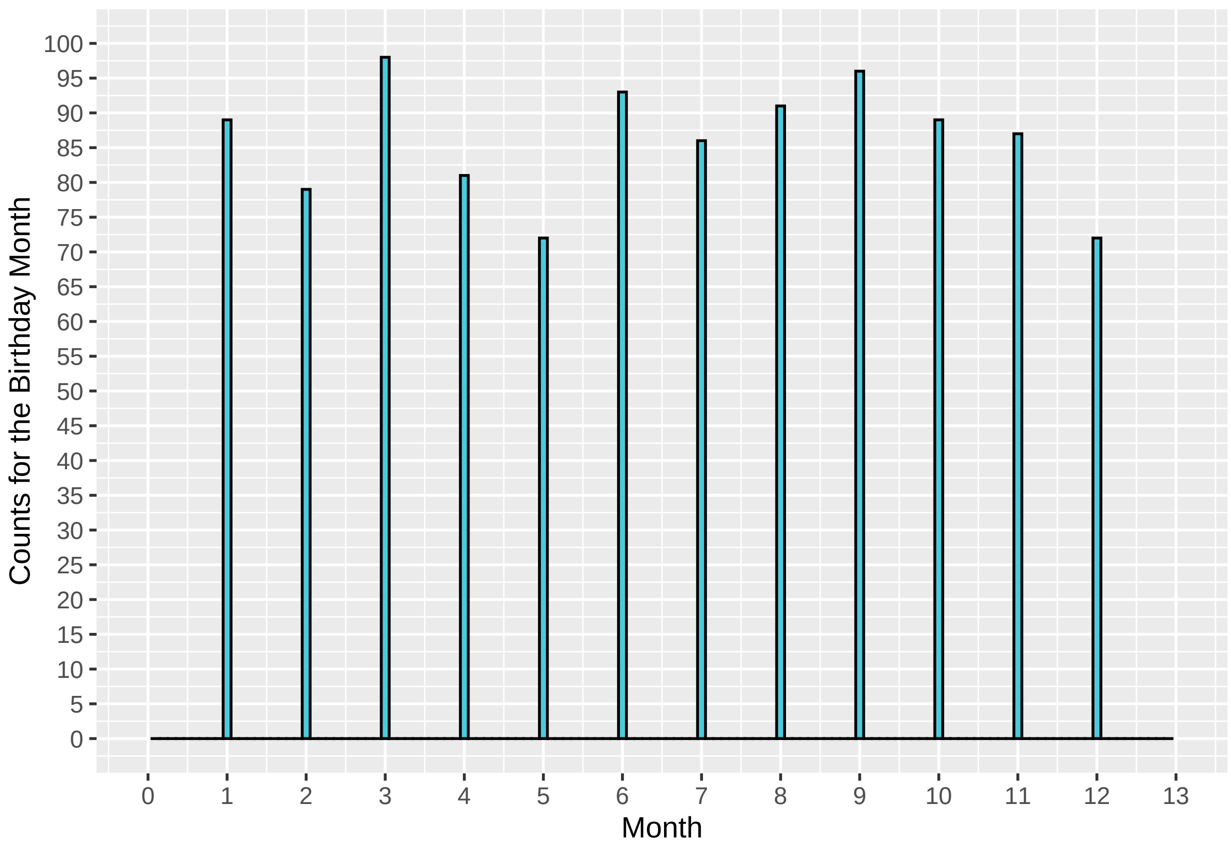
```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
ddf=data.frame(birthday)
ddf$date=as.Date(ddf$dates,format="%m/%d/%y")
ddf$year=year(ymd(ddf$date))
ddf$month=month(ymd(ddf$date))
ddf$day=day(ymd(ddf$date))

library(ggplot2)
ggplot(aes(x=ddf$month),data=ddf)+geom_histogram(color='black',fill='#48CCDD',binwidth =0.1) +scale_x_c
```

```
## Warning: Use of `ddf$month` is discouraged. Use `month` instead.
```

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



```
ggsave('monthofBirthday.png')
```

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

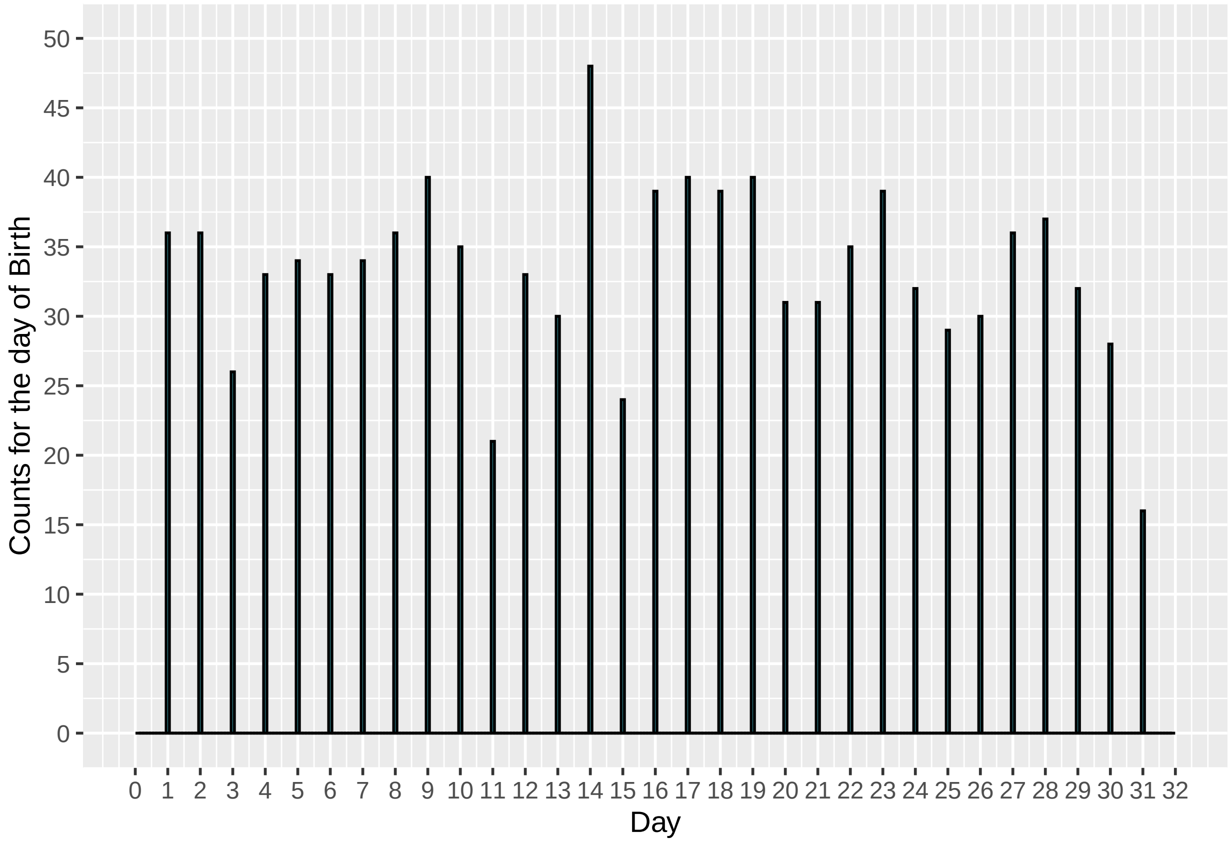
```
## Warning: Use of `ddf$month` is discouraged. Use `month` instead.
```

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

```
ggplot(aes(x=ddf$day),data=ddf)+geom_histogram(color='black',fill='#48CCDD',binwidth =0.1) +scale_x_con
```

```
## Warning: Use of `ddf$day` is discouraged. Use `day` instead.
```

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



```
ggsave('DayofBirthday.png')
```

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

```
## Warning: Use of `ddf$day` is discouraged. Use `day` instead.
```

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

```
#install.packages('tidyverse')
#devtools::install_github("tidyverse/lubridate")



birthMonthTable=table(ddf$month)
birthMonthTable
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12
## 89 79 98 81 72 93 86 91 96 89 87 72
```

```
mostCommonMonth=which(birthMonthTable==max(birthMonthTable))
birthMonthTable=factor(birthMonthTable,levels = c("Jan","Feb", "Mar", "Apr",
                                                    "May", "Jun", "Jul", "Aug",
                                                    "Sep", "Oct", "Nov", "Dec"))
month.abb[mostCommonMonth]
```

```
## [1] "Mar"
```

```
birthDayTable=table(ddf$day)
birthDayTable
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 36 36 26 33 34 33 34 36 40 35 21 33 30 48 24 39 40 39 40 31 31 35 39 32 29 30
## 27 28 29 30 31
## 36 37 32 28 16
```

```
mostCommonDay=which(birthDayTable==max(birthDayTable))

birthYearTable=table(ddf$year)
birthYearTable
```

```
##
## 2014
## 1033
```

```
mostCommonYear=which(birthYearTable==max(birthYearTable))
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

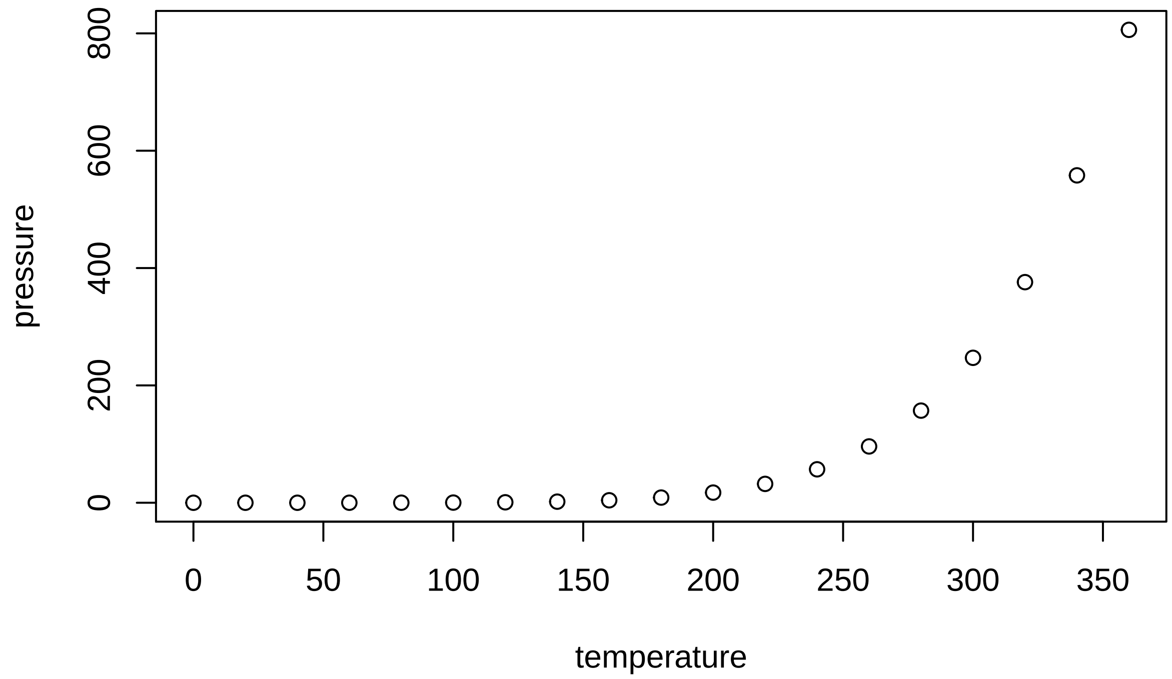
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.