Exploratory Data Analysis

Nik Bear Brown

In the this lesson, we look graphical techniques summarize their main characteristics of one, two variable, and multi-variable data sets. To find relationships amongst variables and to find the variables which are most interesting for a particular analysis task.

Rationale: Exploratory data analysis helps one understand the data, to form and change new theories, and decide which techniques are appropriate for analysis. After a model is finished, exploratory data analysis can look for patterns in these data that may have been missed by the original hypothesis tests. Successful exploratory analyses help the researcher modify theories and refine the analysis.

Additional packages needed

To run the code you may need additional packages.

- If necessary install `ggplot2` package.

`install.packages("ggplot2");

```
require(ggplot2)

## Loading required package: ggplot2
```

Topics

- Exploratory Data Analysis
- Types of Variables
- Descriptive Statistics
- Measures of Central Tendency
- Measures of spread
- Summary Statistics
- Univariate Data Analysis
- Box-Plots
- Bar charts
- Histograms
- Line plots
- Multivariate Data Analysis
- Aesthetic mappings
- Faceting
- Position Adjustments
- Scatter plots

    - Scatter Plot with No apparent relationship
    - Scatter Plot with Linear relationship
    - Scatter Plot with Regression Lines
    - Scatter Plot with Quadratic relationship
    - Scatter plot with Homoscedastic relationship
    - Scatter plot with Jittering
- QQplot
- Assingment

Exploratory Data Analysis

In statistics,exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods concepts apply to statistics and to graphical methods. EDA is for seeing what the data can tell us before the formal modeling or hypothesis testing task.

- from Exploratory Data Analysis - Wikipedia)

Early forms of Exploratory Data Analysis such as the box plot are often attributed to John Tukey (1970s)

John Tukey

John Wilder Tukey (June 16, 1915 - July 26, 2000) was an American mathematician best known for development of the FFT algorithm and box plot.The Tukey range test, the Tukey lambda distribution, the Tukey test all bear his name.

- from John Tukey - Wikipedia)

Goals of Exploratory Data Analysis

- get a general sense of the data

- data-driven (model-free)
- visual (Humans are great pattern recognizers)
- test assumptions (e.g. normal distributions or skewed?)
- identify useful raw data & transforms (e.g. log(x))
- distributions (symmetric, normal, skewed)
- data quality problems
- outliers
- correlations and inter-relationships
- subsets of interest
- suggest functional relationships

Data

We will be using the diamonds,economics, and mpg datasets (from ggplot2) as well as some simulated Twitter. Feel free to tweet questions to [@NikBearBrown](https://twitter.com/NikBearBrown)

```
# Load our data
data(mpg)
data(economics)
data(diamonds)
data_url <-
'http://54.198.163.24/YouTube/MachineLearning/M01/M01_quasi_twitter.csv
'
twitter <- read.csv(url(data_url))
```

R Functions for Understanding Data

- We can use `class()`, `dim()`, `nrow()`, `ncol()`, `names()` to understand datasets
  - `object.size(data.frame)` = returns how much space the dataset is using in memory
- `head(data.frame, n)`, `tail(data.frame, n)` = returns first/last n rows of data; default = 6
- `summary()` = provides summary statistics for each variable, depending on type of variable (numeric,factor).
  - for numerical variables, displays min max, mean median, etx.
  - for categorical (factor) variables, displays number of times each value occurs
- `table(data.frame$variable)` = table of all values of the variable, and how many observations there are for each
- `str(data.frame)` = structure of data, provides data class, num of observations vs variables, and name of class of each variable and preview of its contents
- `view(data.frame)` = opens and view the content of the data frame

ggplot2 diamonds data?

What is in the diamonds data? ?diamonds typed in the R prompt opens up a full description.

```
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
...
```

```
names(diamonds)
```

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

Quasi-Twitter data?

What is in the Quasi-Twitter data? This is a simulated Twitter data set based on real Twitter data.

```
str(twitter)
```

```
## 'data.frame':    21916 obs. of  25 variables:
##  $ screen_name            : Factor w/ 21916 levels
"+5400E1.","000D0se7",..: 4341 15303 21127 13570 14085 3607 14942 8653
15547 19146 ...
##  $ created_at_month       : int  2 11 4 3 4 2 7 5 1 1 ...
##  $ created_at_day         : int  9 21 1 24 23 9 15 23 23 13 ...
##  $ created_at_year        : int  2007 2009 2007 2007 2009 2009 2006
2008 2009 2009 ...
##  $ country                : Factor w/ 44 levels "
Germany","Argentina",..: 44 19 19 44 44 12 44 5 44 44 ...
##  $ location               : Factor w/ 378 levels "Akron
Ohio","Alabama",..: 188 202 25 233 211 79 365 41 242 83 ...
##  $ friends_count          : int  1087 5210 1015 338 641 917 1574
16300 8316 640 ...
##  $ followers_count        : int  22187643 6692814 6257020 3433218
2929559 2540842 1960373 1934803 1855827 1697620 ...
##  $ statuses_count         : int  60246 93910 118465 78082 93892
59397 41023 62178 56057 82912 ...
##  $ favourites_count       : int  1122 3825 1143 0 226 2122 20160 15
540 3 ...
##  $ favourited_count       : int  105005 40487 87968 25943 32589
19760 13558 25084 8732 24515 ...
##  $ dob_day                : int  29 24 4 22 9 1 2 6 15 26 ...
##  $ dob_year               : int  1999 1991 1997 1998 1963 1995 1999
1986 1991 1986 ...
##  $ dob_month              : int  4 10 3 8 11 1 11 10 2 9 ...
```

```
##  $ gender                 : Factor w/ 2 levels "female","male": 1 1
2 2 1 1 1 2 1 2 ...
##  $ mobile_favourites_count: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mobile_favourited_count: int  0 5032191 0 0 0 0 0 1934803 0 0 ...
##  $ education              : int  8 15 9 9 13 15 14 10 11 12 ...
##  $ experience             : int  0 0 0 44 24 21 31 0 27 20 ...
##  $ age                    : int  29 0 32 40 45 14 27 31 34 40 ...
##  $ race                   : Factor w/ 10 levels "arab","asian",..:
10 10 10 10 10 10 10 10 2 1 ...
##  $ wage                   : num  16.3 17.9 15.7 7 17.9 ...
##  $ retweeted_count        : int  1 1 2 0 1 2 1 2 0 0 ...
##  $ retweet_count          : int  30 6 65 8 7 64 13 14 15 10 ...
##  $ height                 : int  156 162 168 180 162 158 160 178 156
173 ...
```

```
names(twitter)
```

```
##  [1] "screen_name"             "created_at_month"
##  [3] "created_at_day"          "created_at_year"
##  [5] "country"                 "location"
##  [7] "friends_count"           "followers_count"
##  [9] "statuses_count"          "favourites_count"
## [11] "favourited_count"        "dob_day"
## [13] "dob_year"                "dob_month"
## [15] "gender"                  "mobile_favourites_count"
## [17] "mobile_favourited_count" "education"
## [19] "experience"              "age"
## [21] "race"                    "wage"
## [23] "retweeted_count"         "retweet_count"
## [25] "height"
```

Types of Variables

Continuous variables

Always numeric Integers - a whole number; a number that is not a fraction. e.g. -1,0,1,2,3,4, ...

Floating point numbers (a rational number or a 'float') - a rational number is any number that can be expressed as the quotient or fraction p/q of two integers, p and q, with the denominator q not equal to zero. Since q may be equal to 1, every integer is a rational number. e.g. -1.3, 0.0, 3.14159265359, 2.71828, ...

Continuous variables can be any number, positive or negative

Examples: age in years, weight, website 'hits' and other measurements

Categorical variables A categorical variable is a variable that can take on one of a limited, and usually fixed, number of possible values.

Types of categorical variables are ordinal, nominal and dichotomous (binary)

Nominal Variables

Nominal variable is a categorical variable without an intrinsic order

Examples of nominal variables: Where a person lives in the U.S. (Northeast, South, Midwest, etc.) Sex (male, female) Nationality (American, Mexican, French) Race/ethnicity (African American, Hispanic, White, Asian American)

Enum	*Nationality* ← nominal
1	American
2	Mexican
3	French
4	Brasilian

Ordinal Variables

Ordinal variables are categorical variable with some intrinsic order or numeric value

Examples of ordinal variables: Education (no high school degree, HS degree, some college, college degree) Agreement (strongly disagree, disagree, neutral, agree, strongly agree) Rating (excellent, good, fair, poor) Any other scale (e.g. On a scale of 1 to 5)

Rank	*Degree* ← ordinal
1	PhD
2	Master's
3	Bachelors
4	Associate's
5	High School

Dichotomous Variables

Dichotomous (or binary) (or boolean) variables – a categorical variable with only 2 levels of categories * yes or no * true or false * accept or reject * pass or fail

Factors

Factors represent *categorical data* in R

There are two types: **unordered** vs **ordered** just like **ordinal** vs **nominal** categorical variables.

```
* `table(factor_Variable)` = how many of each are in the factor
```

Factor is a data type that is unique to R (although similar to an Enumerated type in other languages.)

- The *levels* are the external presentation of the factors
- Internally it's stored as integers (what is known as an "enum" in many other languages)

Descriptive Statistics

Descriptive statistics is the discipline of quantitatively describing the main features of a collection of data. Common descriptive measures used are measures of central tendency and measures of variability or dispersion or spread.

Measures of central tendency

Measures of central tendency are used to describe the most typical measure.

Mode: the value in a string of numbers that occurs most often Median: the value whose occurrence lies in the middle of a set of ordered values

Mean: sometimes referred to as the arithmetic mean. It is the average value characterizing a set of numbers

Mode

- the value that occurs most frequently
- used w/ nominal data
- there can be "ties"

Median

- score at the center of the distribution
- sort and take the middle value (or average or two middle values)
- determine w/ : $\frac{N+1}{2}$
- equal to 50th percentile

Mean (or arithmetic mean)

- $\overline{X} = \frac{\sum X}{N}$
- can be misleading when there are large outliers

Properties of the Measures of Center

1. adding or subtracting a constant does the same to the measure of center
2. multiplying or dividing by a constant does the same to the measure of center

Measures of Spread

Measures of variability aare used to reveal the typical difference between the values in a set of values

Measures of Spread
* Range, Quartile 2nd Quartile is the median * Quartiles: sort and divide in 4 parts. *

Frequency distribution reveals the number (percent) of occurrences of each number or set of numbers * Range identifies the maximum and minimum values in a set of numbers * Standard deviation (or variance) indicates the degree of variation

Summary statistics

In descriptive statistics, summary statistics are used to summarize a set of observations, in order to communicate the largest amount of information as simply as possible. Statisticians commonly try to describe the observations in:

- a measure of location, or central tendency, such as the arithmetic mean
- a measure of statistical dispersion like the standard deviation
- a measure of the shape of the distribution like skewness or kurtosis
- if more than one variable is measured, a measure of statistical dependence such as a correlation coefficient

from Summary statistics - Wikipedia

```
mean(twitter$followers_count)

## [1] 5859.258

range(twitter$followers_count)

## [1]        0 22187643

diff(range(twitter$followers_count))

## [1] 22187643

head(twitter[,"followers_count"])

## [1] 22187643  6692814  6257020  3433218  2929559  2540842

median(twitter$followers_count)

## [1] 336

mean(twitter$followers_count)-median(twitter$followers_count)

## [1] 5523.258

var(twitter$followers_count)

## [1] 29142362169

sd(twitter$followers_count)

## [1] 170711.3

sqrt(var(twitter$followers_count))

## [1] 170711.3
```

```
summary(twitter$followers_count)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        0      105      336     5859     1075 22190000

summary(twitter)

##     screen_name    created_at_month created_at_day  created_at_year
##  +5400E1. :    1   Min.   : 1.000   Min.   : 1.00   Min.   :2006
##  000D0se7 :    1   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.:2009
##  001apdov :    1   Median : 6.000   Median :16.00   Median :2011
##  001RBTePh:    1   Mean   : 6.069   Mean   :15.78   Mean   :2011
##  003B0K2  :    1   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:2013
##  007unfasa:    1   Max.   :12.000   Max.   :31.00   Max.   :2015
##  (Other)  :21910
##       country                       location     friends_count
##  USA      :14905   Mexico              :  122   Min.   :   -84
##  Canada   :  943   Boston              :  108   1st Qu.:   123
##  India    :  890   Montreal            :  107   Median :   324
##  Earth    :  516   Nevada              :   80   Mean   :  1058
##  England  :  467   Bangalore           :   79   3rd Qu.:   849
##  Australia:  291   Indianapolis Indiana:   76   Max.   :660549
##  (Other)  : 3904   (Other)             :21344
##  followers_count    statuses_count    favourites_count
##  Min.   :       0   Min.   :      1   Min.   :      0
##  1st Qu.:     105   1st Qu.:    558   1st Qu.:     16
##  Median :     336   Median :   2341   Median :    164
##  Mean   :    5859   Mean   :  12486   Mean   :   2217
##  3rd Qu.:    1075   3rd Qu.:   9348   3rd Qu.:    950
##  Max.   :22187643   Max.   :1136198   Max.   :1140139
##
##  favourited_count       dob_day         dob_year      dob_month
##  Min.   :     0.00   Min.   : 1.00   Min.   :1900   Min.   :   1.000
##  1st Qu.:     2.00   1st Qu.: 5.00   1st Qu.:1965   1st Qu.:   3.000
##  Median :     9.00   Median :13.00   Median :1982   Median :   6.000
##  Mean   :    92.24   Mean   :13.49   Mean   :1976   Mean   :   6.398
##  3rd Qu.:    36.00   3rd Qu.:21.00   3rd Qu.:1990   3rd Qu.:   9.000
##  Max.   :105005.00   Max.   :35.00   Max.   :2000   Max.   :1992.000
##
##     gender      mobile_favourites_count mobile_favourited_count
##  female: 7319   Min.   :     0.0        Min.   :      0
##  male  :14569   1st Qu.:     0.0        1st Qu.:      0
##  NA's  :   28   Median :     0.0        Median :      0
##                 Mean   :   152.9        Mean   :    649
##                 3rd Qu.:     0.0        3rd Qu.:      0
##                 Max.   :377123.0        Max.   :5032191
##
##    education      experience          age                        race
##  Min.   : 3.0   Min.   :-32.00   Min.   :-6.00   white
:18032
```

```
##  1st Qu.:11.0   1st Qu.:  0.00   1st Qu.:28.00   latino          :
1115
##  Median :13.0   Median :  7.00   Median :36.00   asian           :
960
##  Mean   :12.5   Mean   : 10.88   Mean   :35.54   persian         :
376
##  3rd Qu.:14.0   3rd Qu.: 20.00   3rd Qu.:44.00   hispanic        :
353
##  Max.   :24.0   Max.   : 74.00   Max.   :91.00   pacific islander:
276
##                                                  (Other)         :
804
##       wage         retweeted_count    retweet_count         height
##  Min.   :  5.00   Min.   :  0.0000   Min.   :   0.00   Min.   :  1.0
##  1st Qu.: 13.52   1st Qu.:  0.0000   1st Qu.:   0.00   1st Qu.:165.0
##  Median : 20.36   Median :  1.0000   Median :   3.00   Median :172.0
##  Mean   : 22.97   Mean   :  0.9715   Mean   :  52.73   Mean   :171.5
##  3rd Qu.: 28.40   3rd Qu.:  1.0000   3rd Qu.:  19.00   3rd Qu.:178.0
##  Max.   :104.97   Max.   :705.0000   Max.   :5506.00   Max.   :203.0
##
```

Univariate Data Analysis

Univariate data analysis-explores each variable in a data set separately. This serves as a good method to check the quality of the data on a variable by variable basis. See Wikipedia Univariate analysis

Five point summary

Five point summary (min, Q1,Q2,Q3, max)

- the sample minimum (smallest observation)
- the lower quartile or first quartile
- the median (middle value)
- the upper quartile or third quartile
- the sample maximum (largest observation)

In R the function `summary()` = provides different output for each variable, depending on type of variable (numeric,factor)

```
r  summary(diamonds$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   ##     326     950
2401    3933    5324   18820
```

Note that summary can be applied to the whole data set.

```
r  summary(diamonds)
```

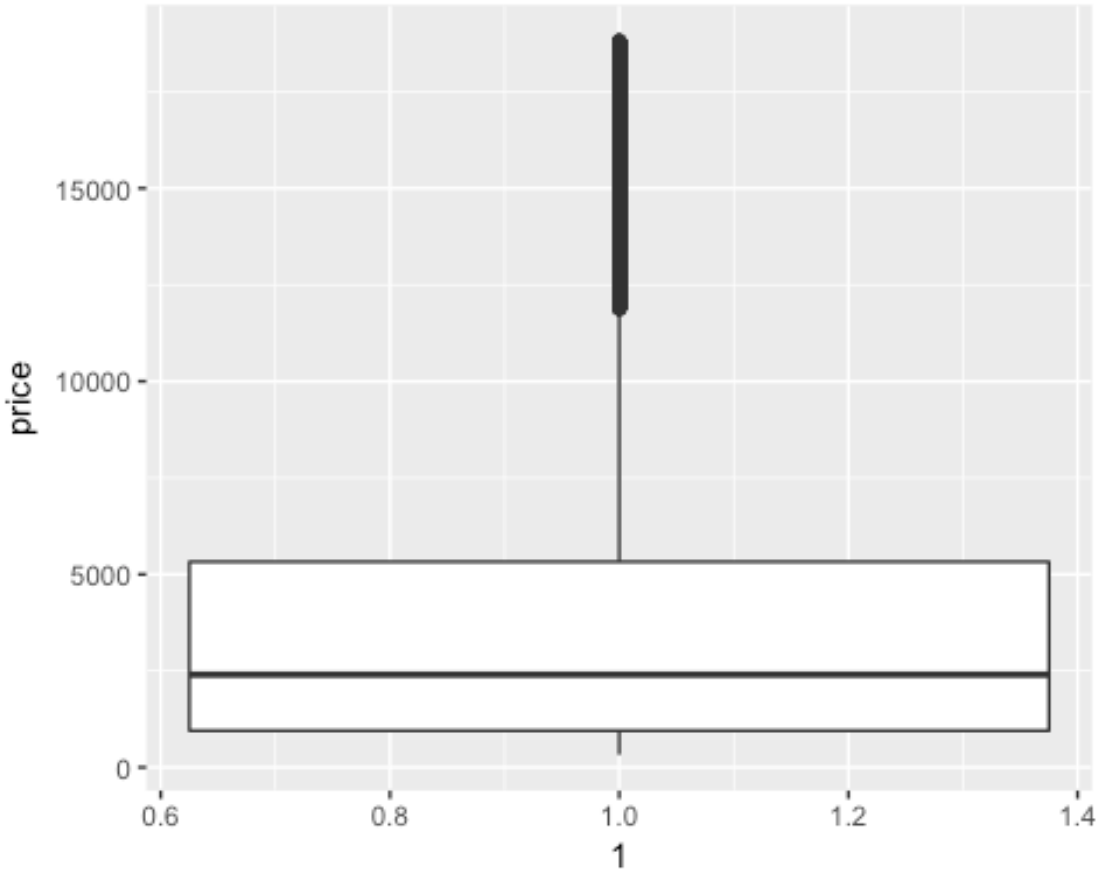
```
##      carat               cut        color        clarity       ##
Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065    ##  1st
Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258    ##  Median
:0.7000   Very Good:12082   F: 9542   SI2     : 9194    ##  Mean
:0.7979   Premium  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

Box-Plot

A Box-Plot is a visual representation of a five point summary, with some additional information about outliers (1.5 times the lower and upper quartiles)

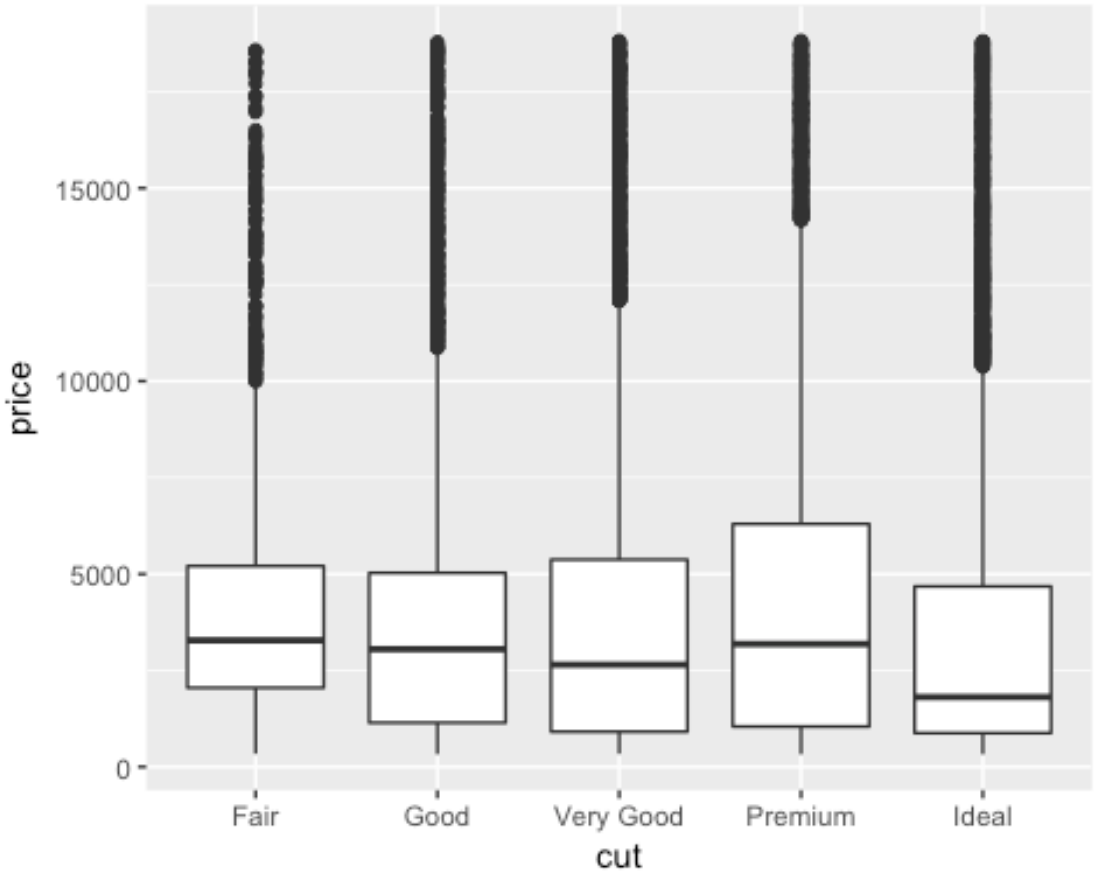
Box-Plot (< 1.5 times Q1 outliers, expected min, Q1,Q2,Q3, expected max, outliers > 1.5 times Q3)

```
qplot(1,price, data=diamonds, geom="boxplot")
```



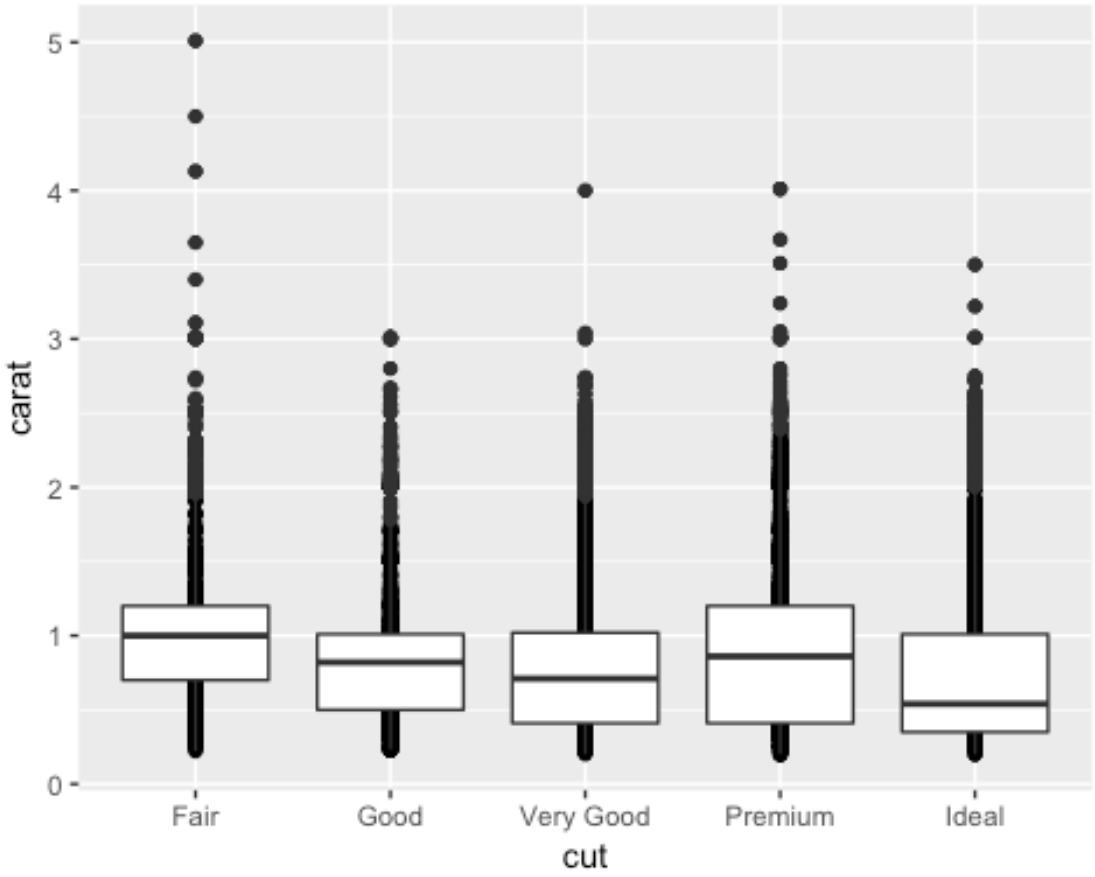
Box-Plots are useful for comparing five point summary of subsets of a data set confitioned on a factor.

```
qplot(cut,price, data=diamonds, geom="boxplot")
```



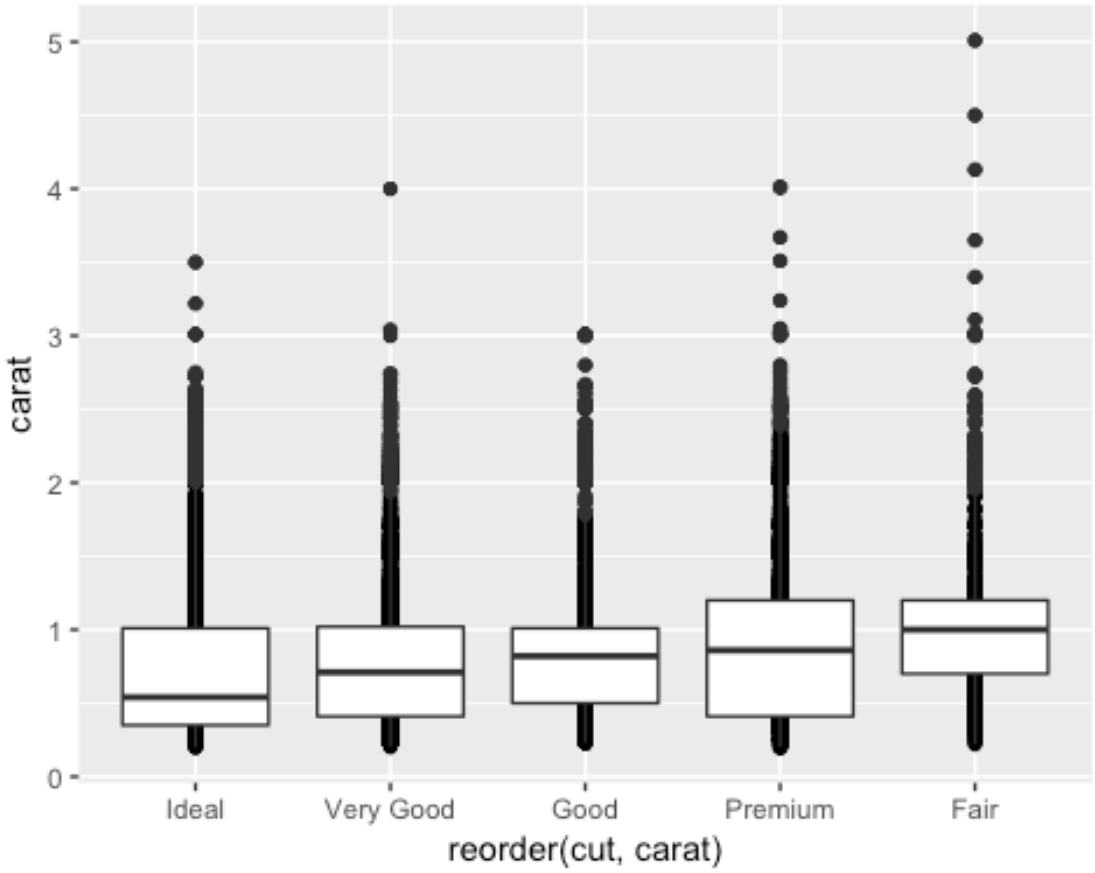
Alternatively it can be written using "layers."

```
qplot(cut,carat, data=diamonds)+geom_boxplot()
```



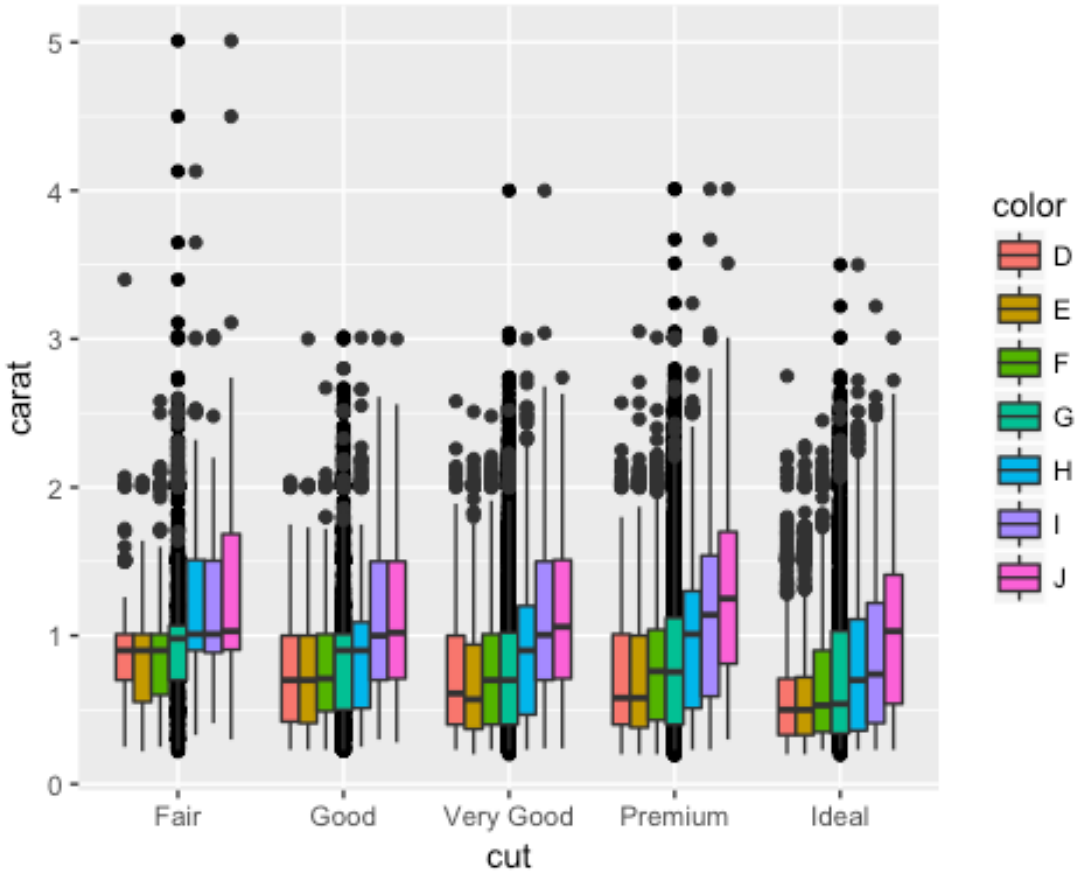
Reordering a boc-plot.

```
qplot(reorder(cut,carat),carat, data=diamonds)+geom_boxplot()
```

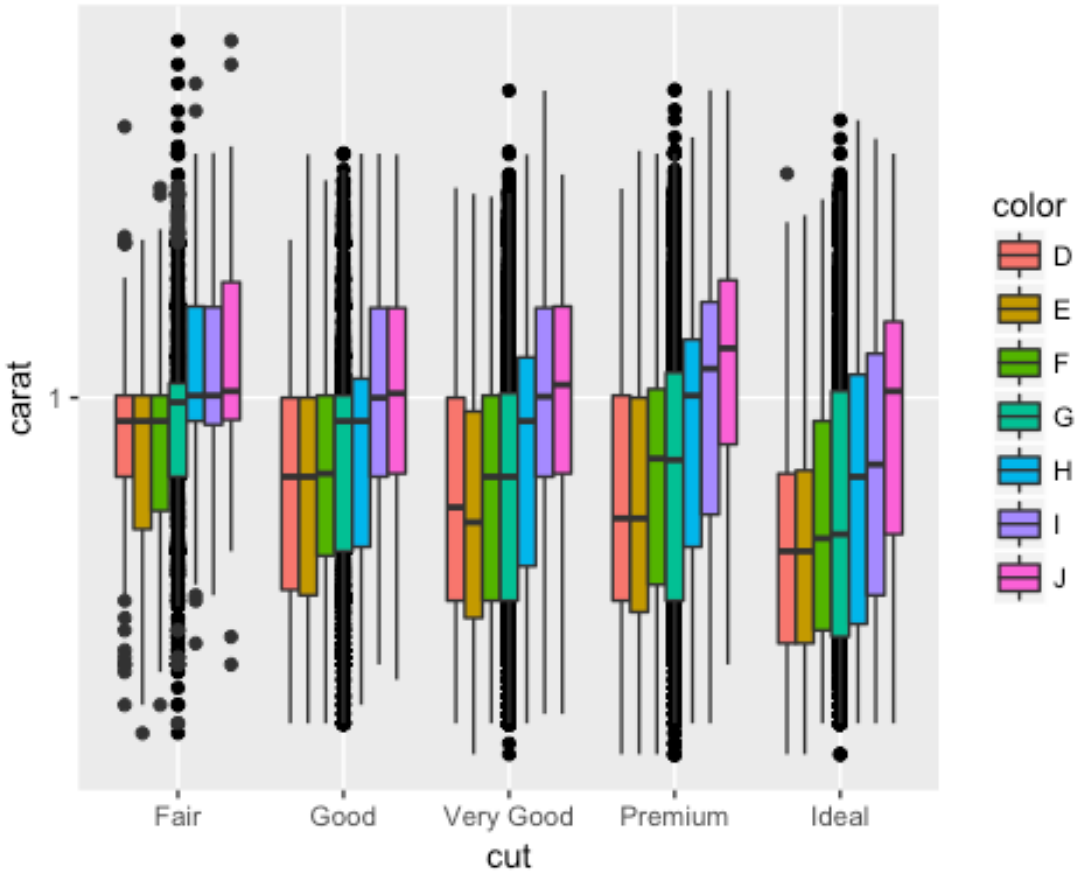


Filling by the color of the diamond.

```
qplot(cut,carat, data=diamonds,fill=color)+geom_boxplot()
```



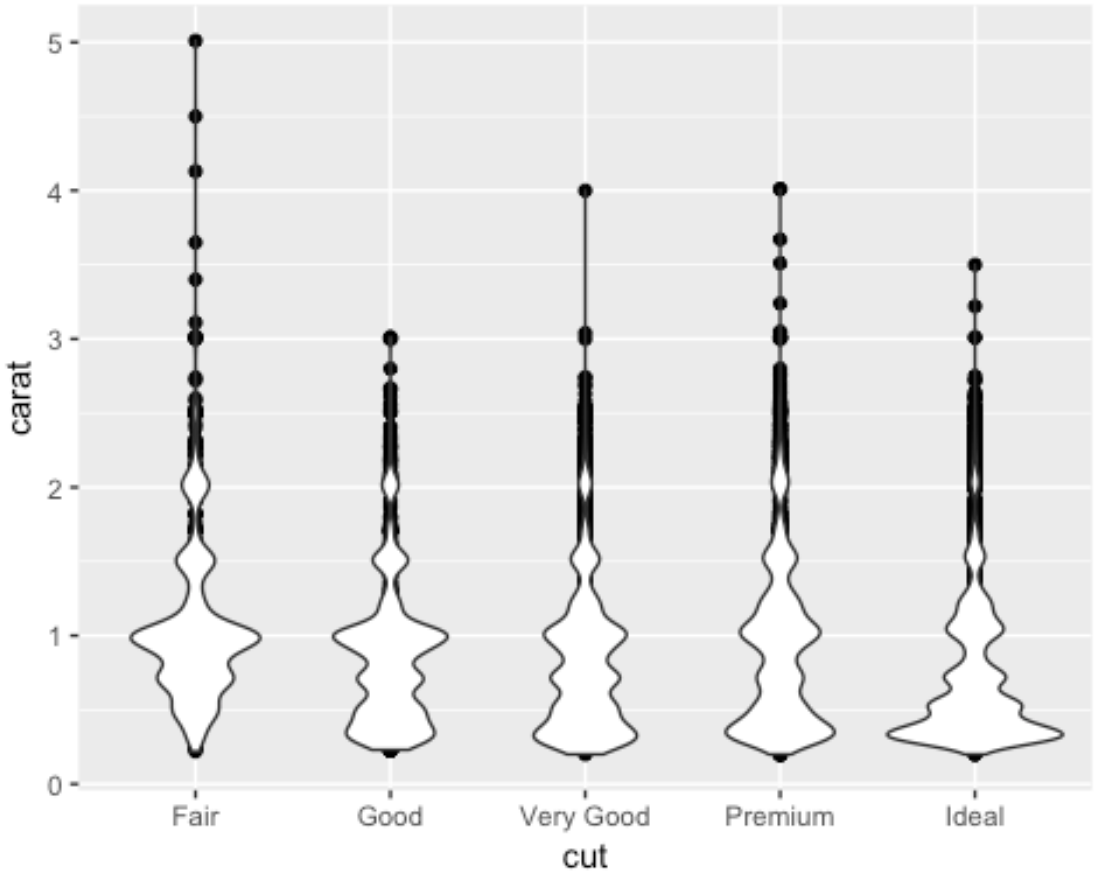

```
qplot(cut,carat, data=diamonds,fill=color)+geom_boxplot() +
scale_y_log10()
```



Violin plots

A violin plot is a method of plotting numeric data. It is a box plot with a rotated kernel density plot on each side. The violin plot is similar to box plots, except that they also show the probability density of the data at different values.

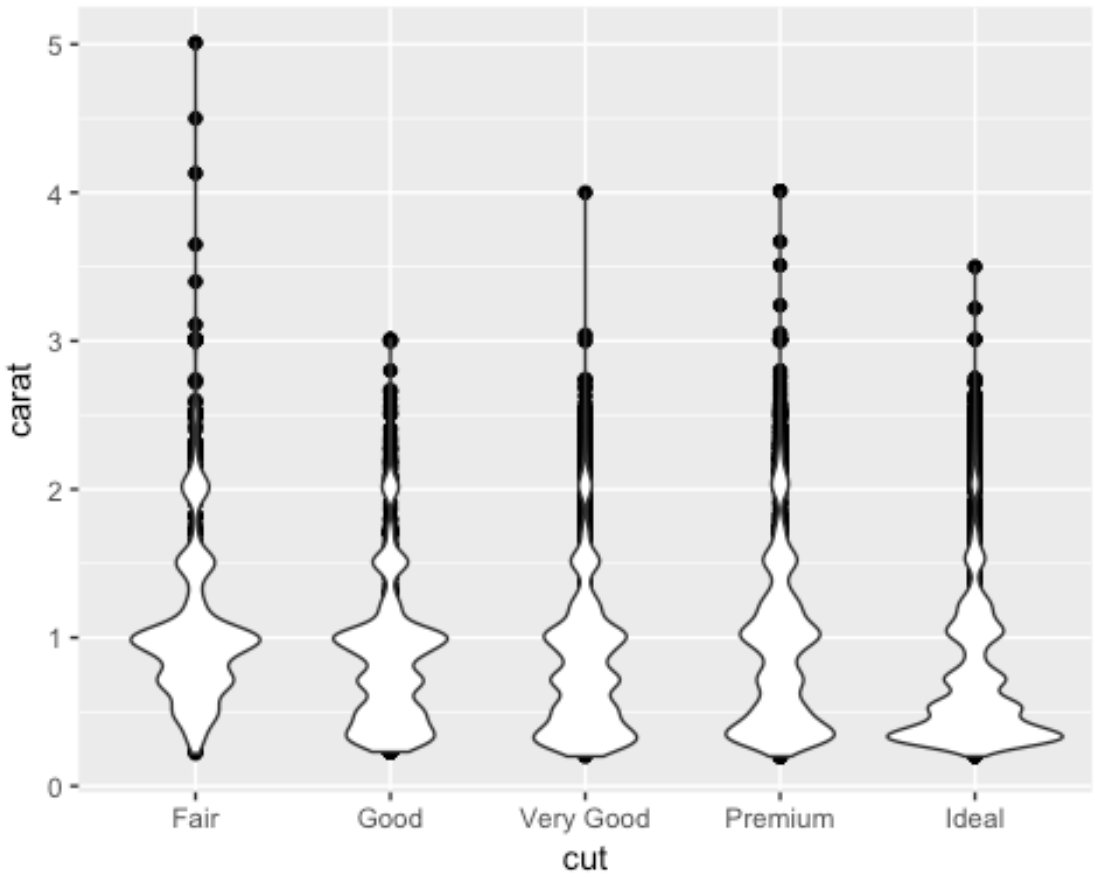
```
qplot(cut,carat, data=diamonds)+geom_violin()
```



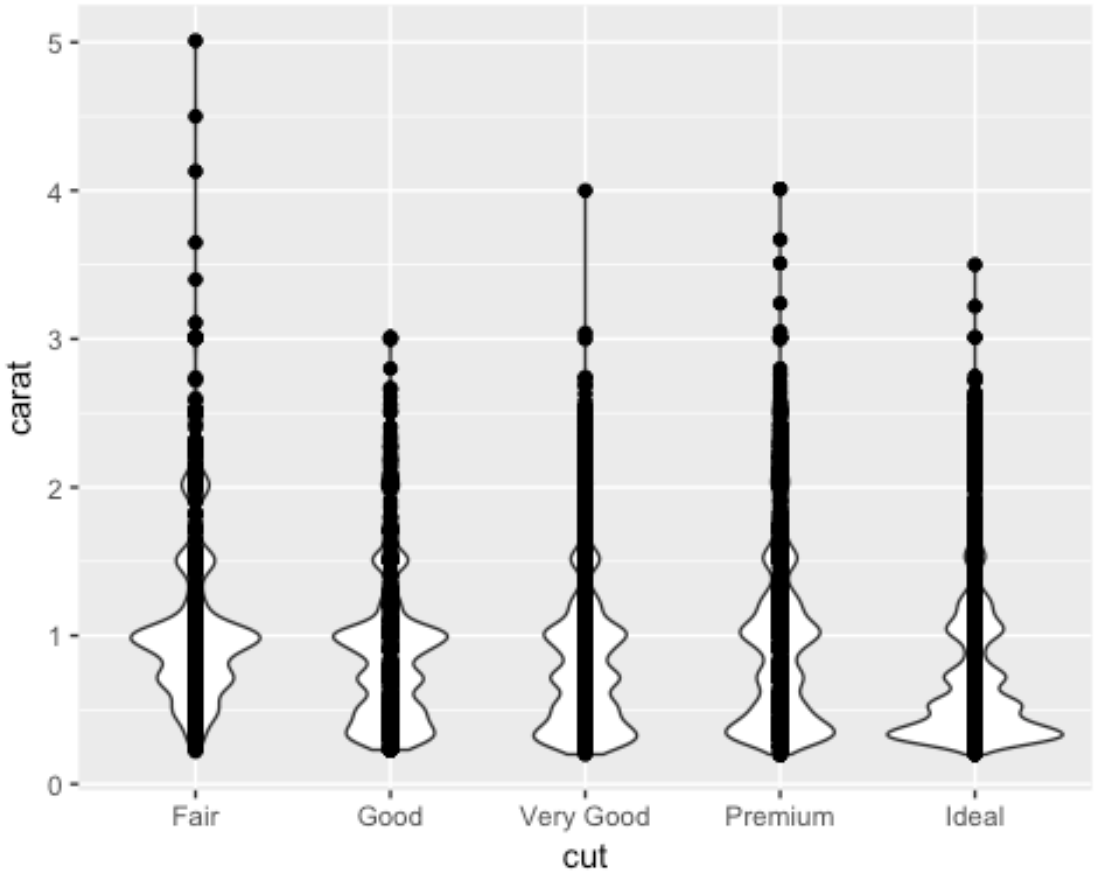
Saving a graph to a variable

Graph can be saved to a variable to easily play with geom's.

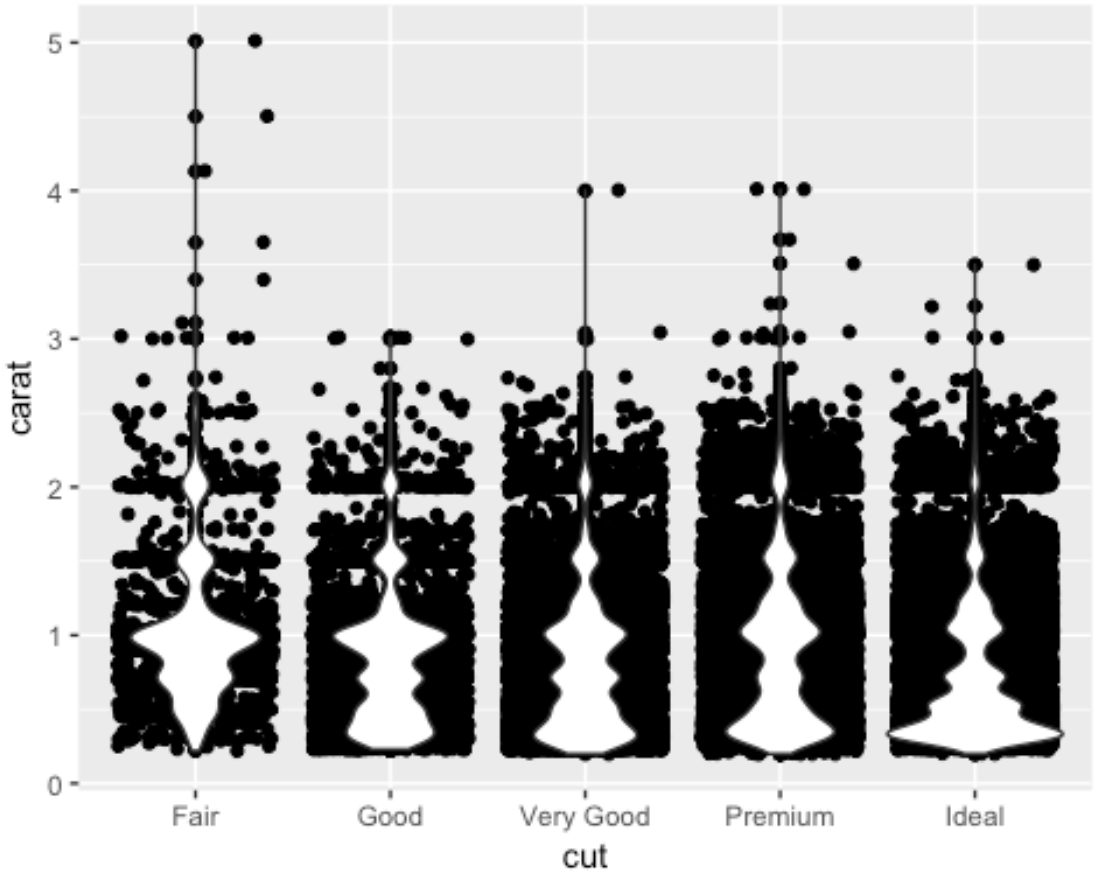
```
g<-qplot(cut,carat, data=diamonds)
g+geom_point()+geom_violin()
```




```
g+geom_violin()+geom_point()
```



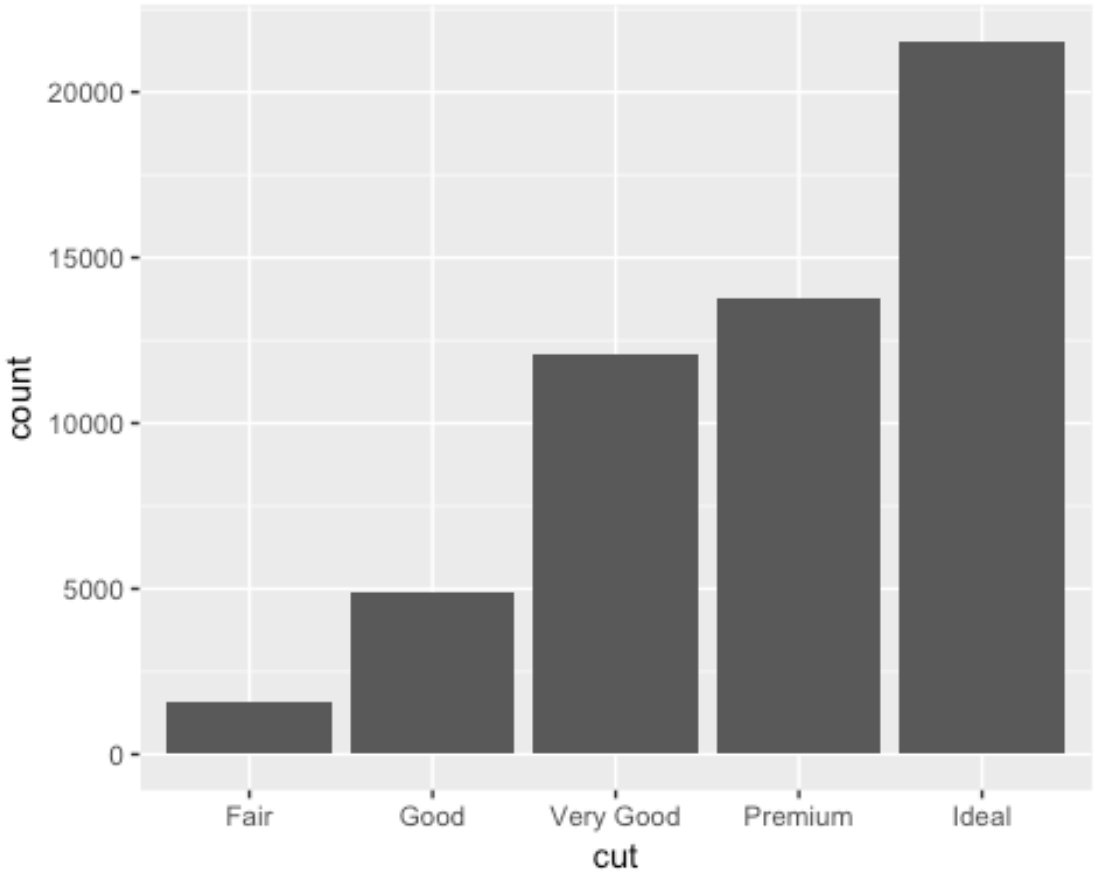

```
g+geom_jitter()+geom_point()+geom_violin()
```



Bar charts

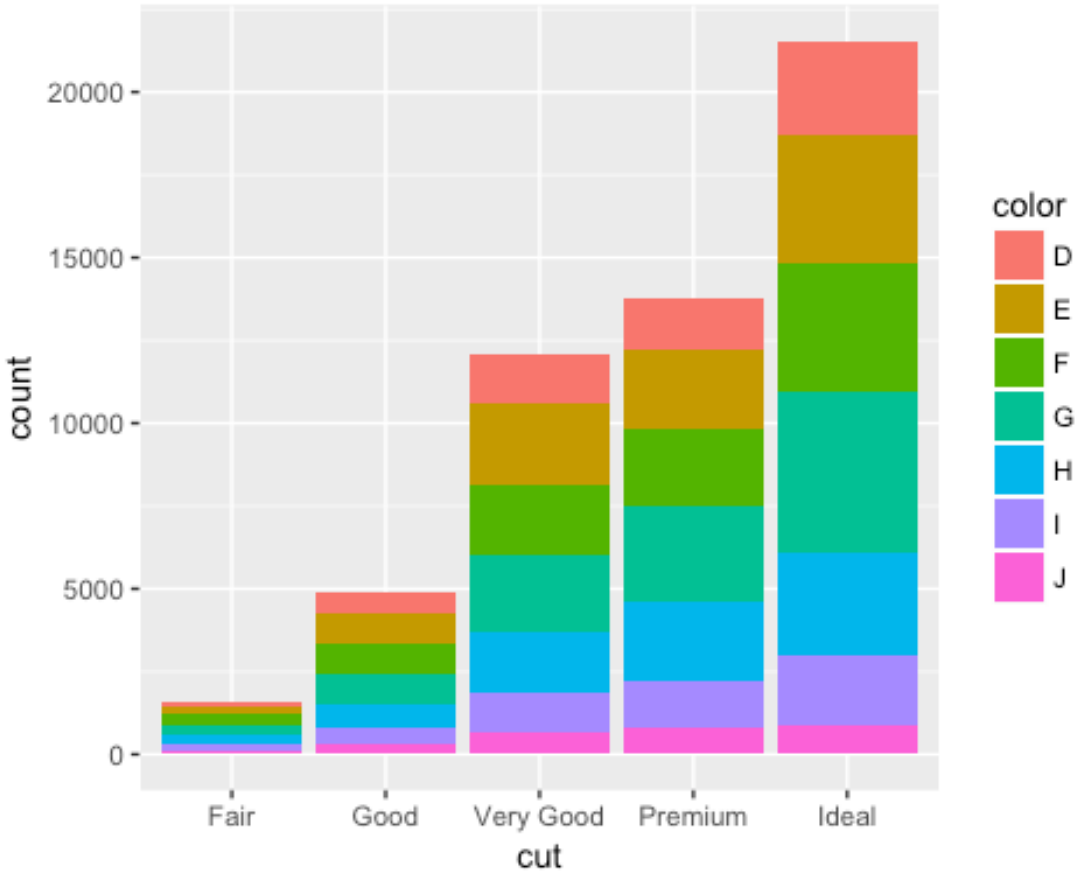
A bar chart or bar graph is a chart that presents Grouped data with rectangular bars with lengths proportional to the values that they represent. Bar chart - Wikipedia

```
qplot(cut, data=diamonds)
```



We can use an aesthetic mappings (e.g. color, size, shape) to dig deeper in the data by mapping variables to visual properties (aesthetics).

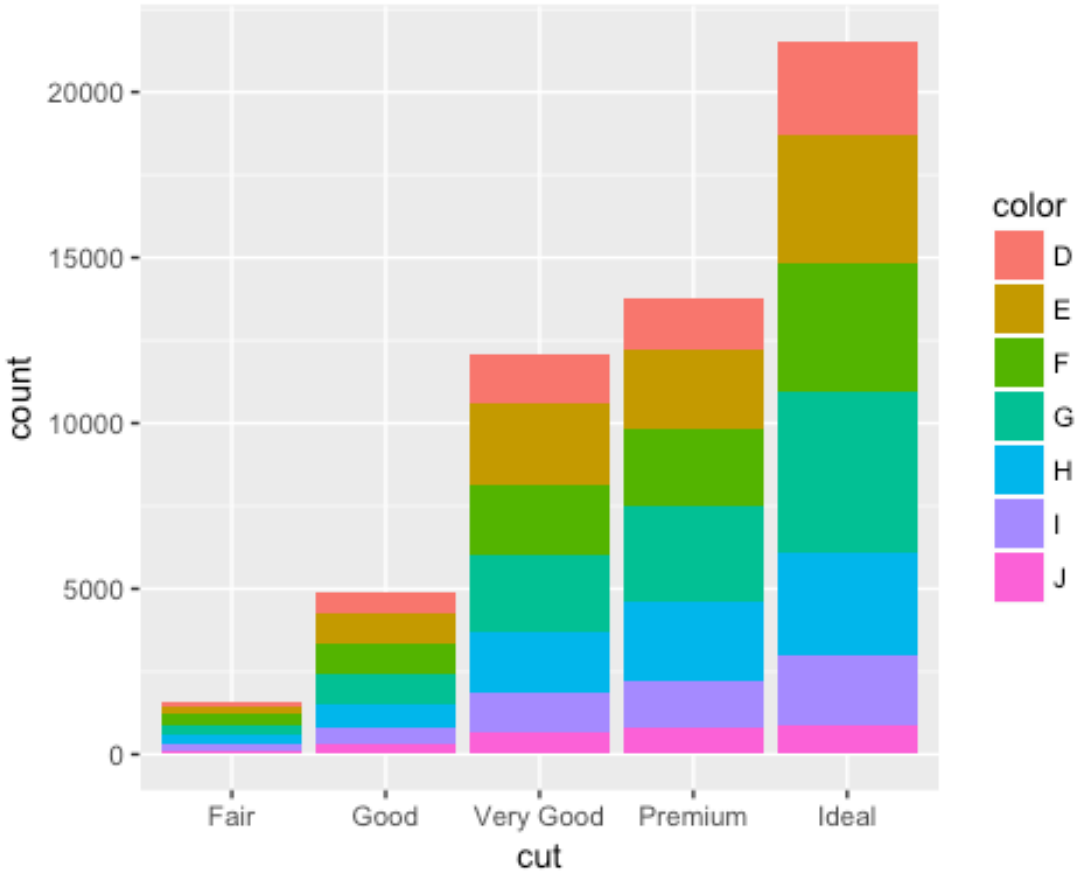
```
qplot(cut, data=diamonds, fill=color)
```



Position Adjustments

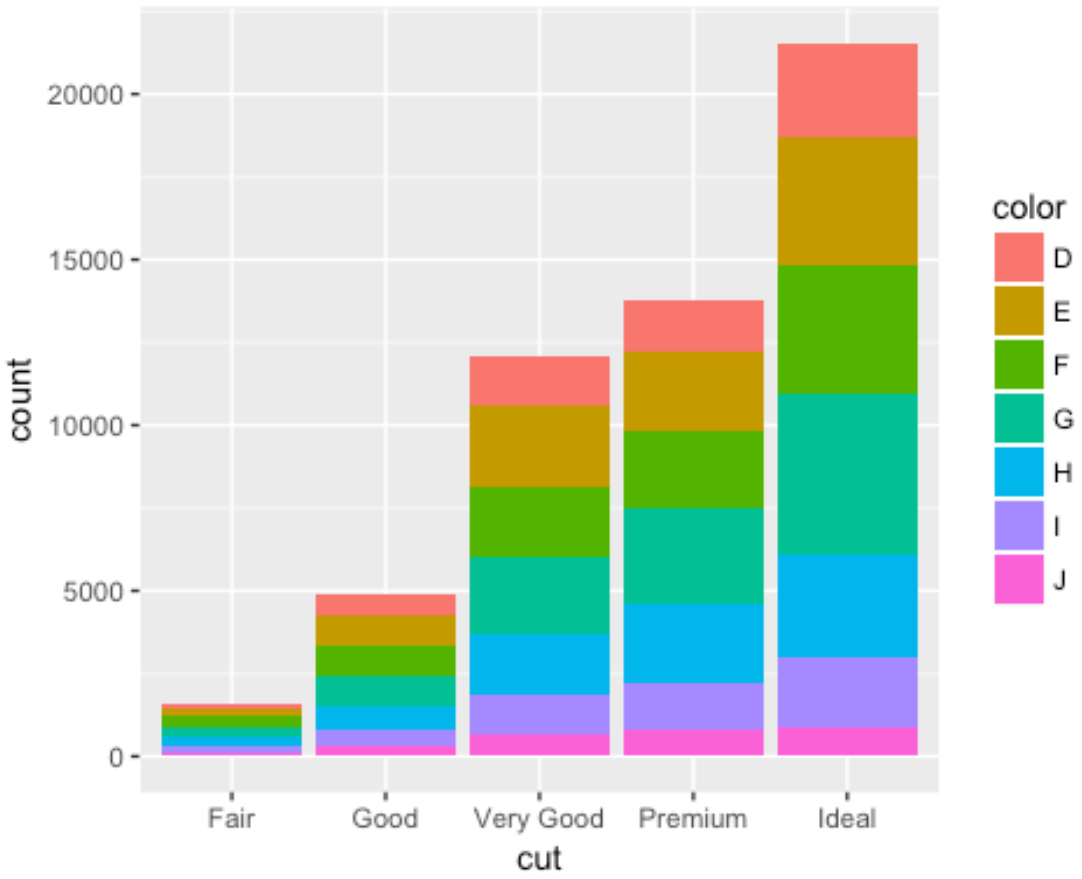
Position adjustments can be used to fine tune positioning of objects to achieve effects like dodging, jittering and stacking.
* position_dodge - Adjust position by dodging overlaps to the side.
* position_fill - Stack overlapping objects on top of one another, and standardise to have
* position_identity - Don't adjust position
* position_stack - Stack overlapping objects on top of one another.
* position_jitter - Jitter points to avoid overplotting (we will discuss jitter in the scatter plots section)

```
qplot(cut, data=diamonds, fill=color, position="stack")

## Warning: `position` is deprecated
```

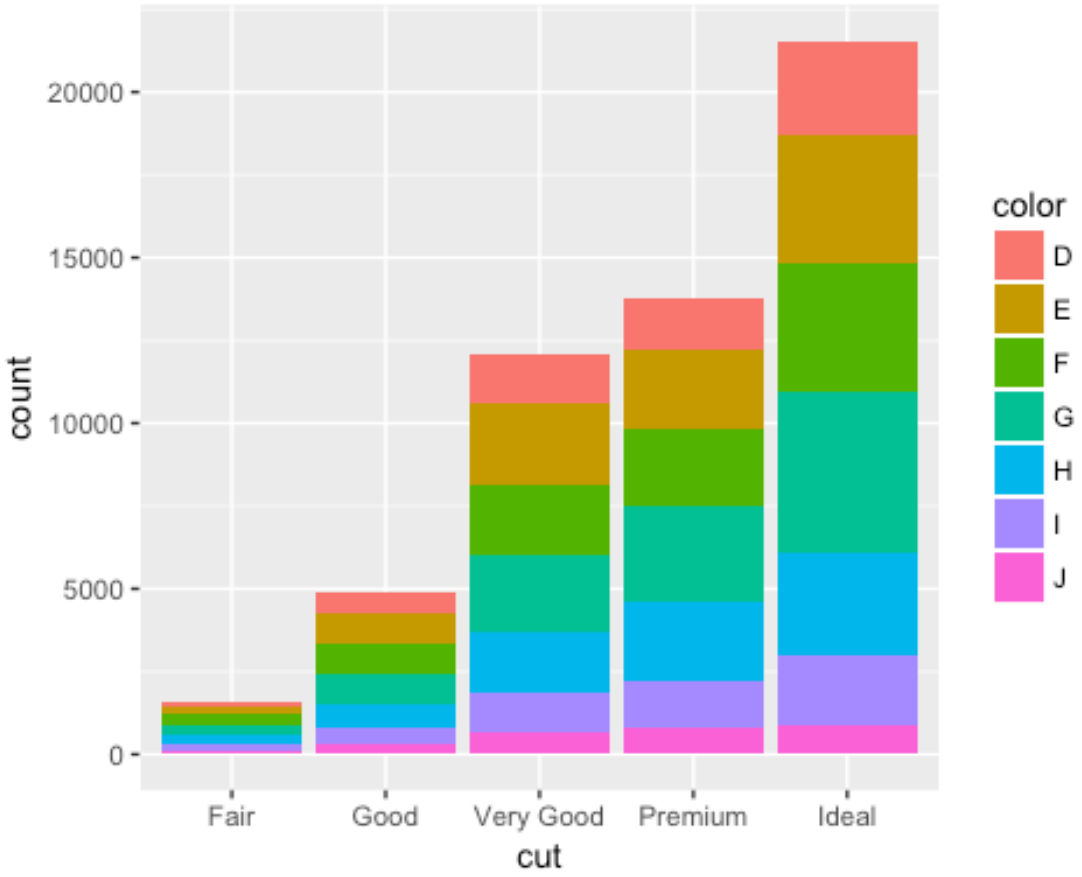


position stack - Stack overlapping objects on top of one another stack is the default.

```
qplot(cut, data=diamonds, fill=color)
```



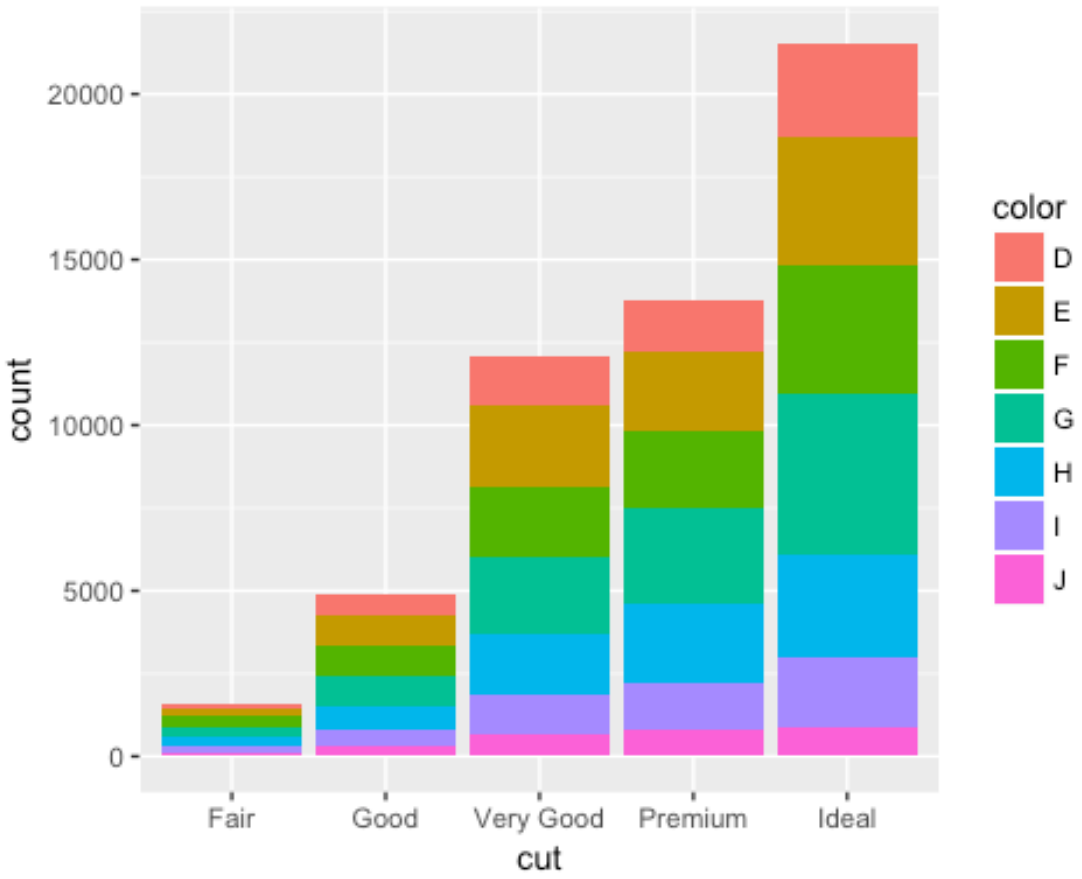

```
qplot(cut, data=diamonds, fill=color, position="dodge")

## Warning: `position` is deprecated
```



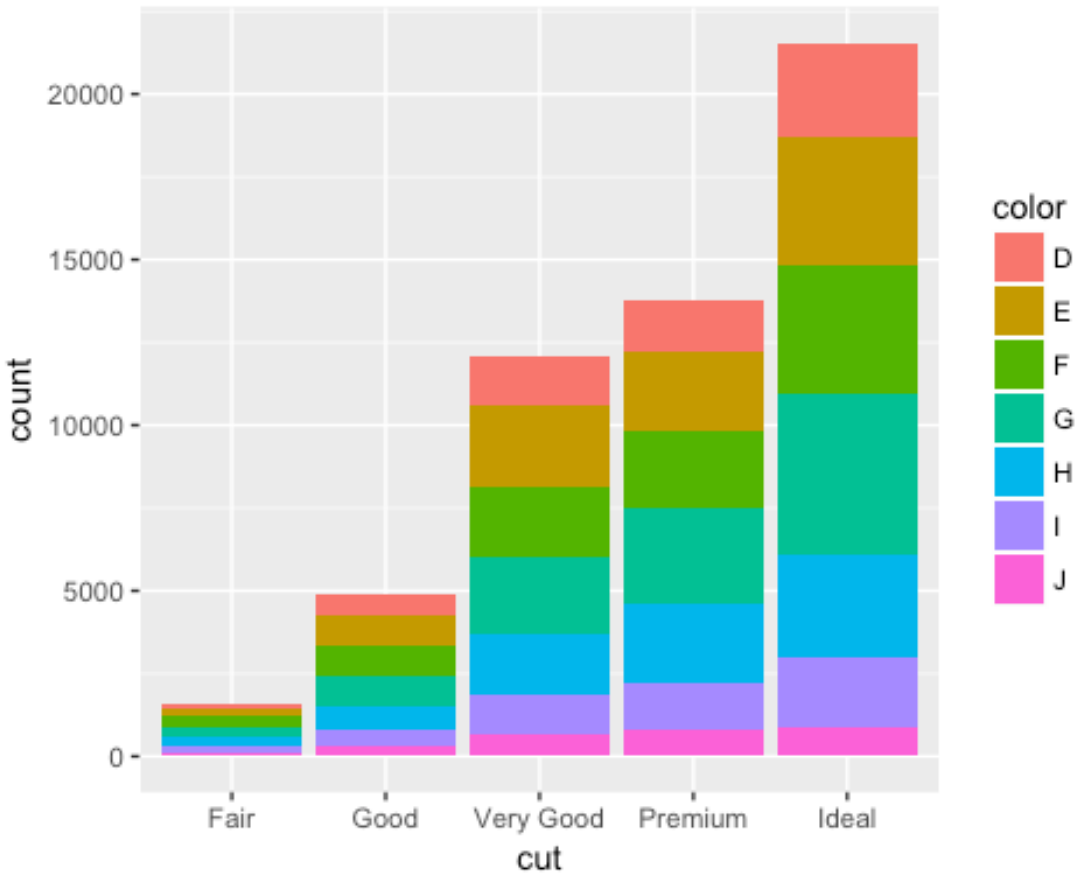
position dodge - Adjust position by dodging overlaps to the side

```
qplot(cut, data=diamonds, fill=color, position="identity")
```

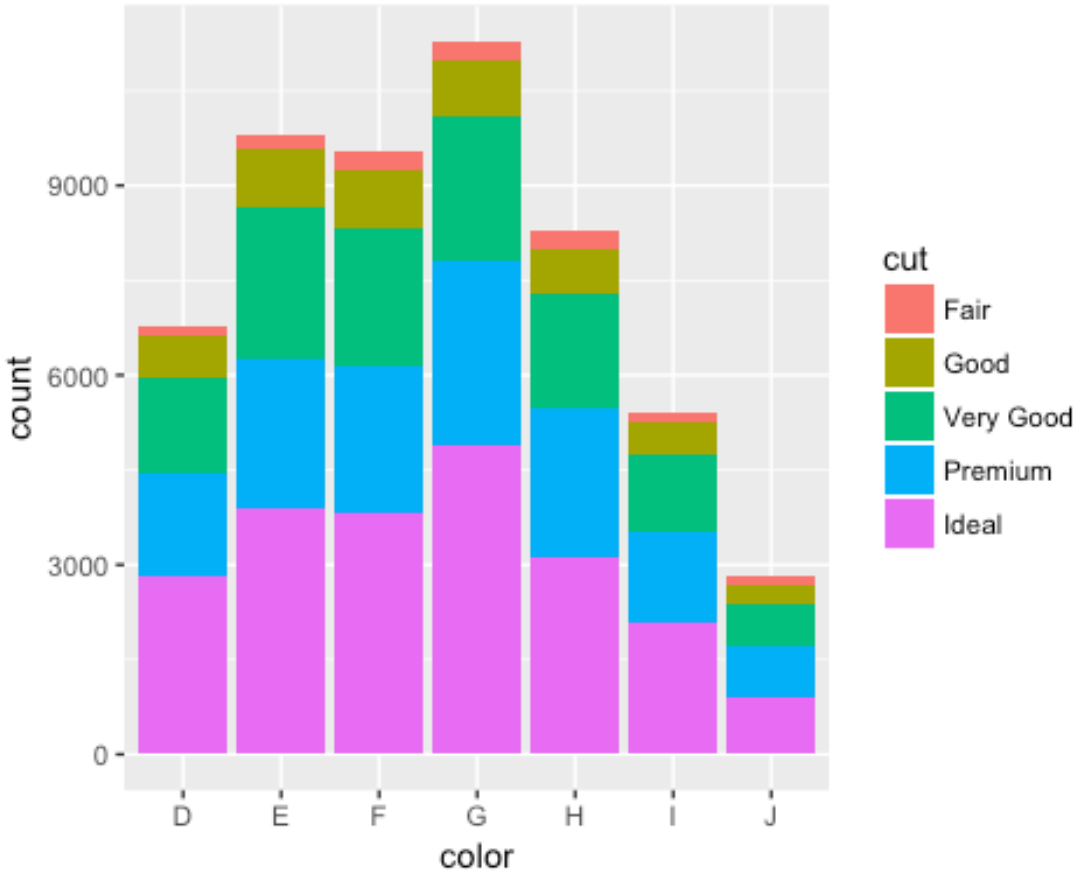
```
## Warning: `position` is deprecated
```




```
qplot(cut,data=diamonds,fill=color,position = "identity")

## Warning: `position` is deprecated
```

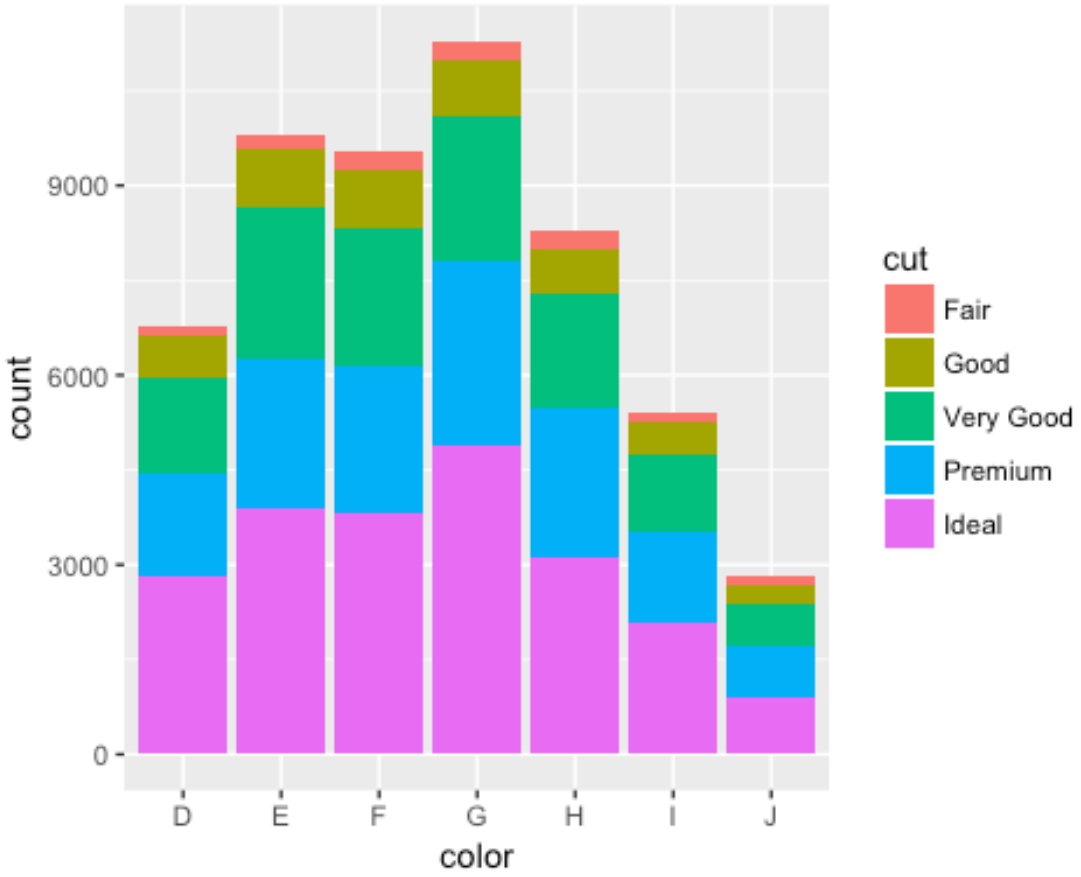



```
qplot(color,data=diamonds,fill=cut,position = "identity")

## Warning: `position` is deprecated
```



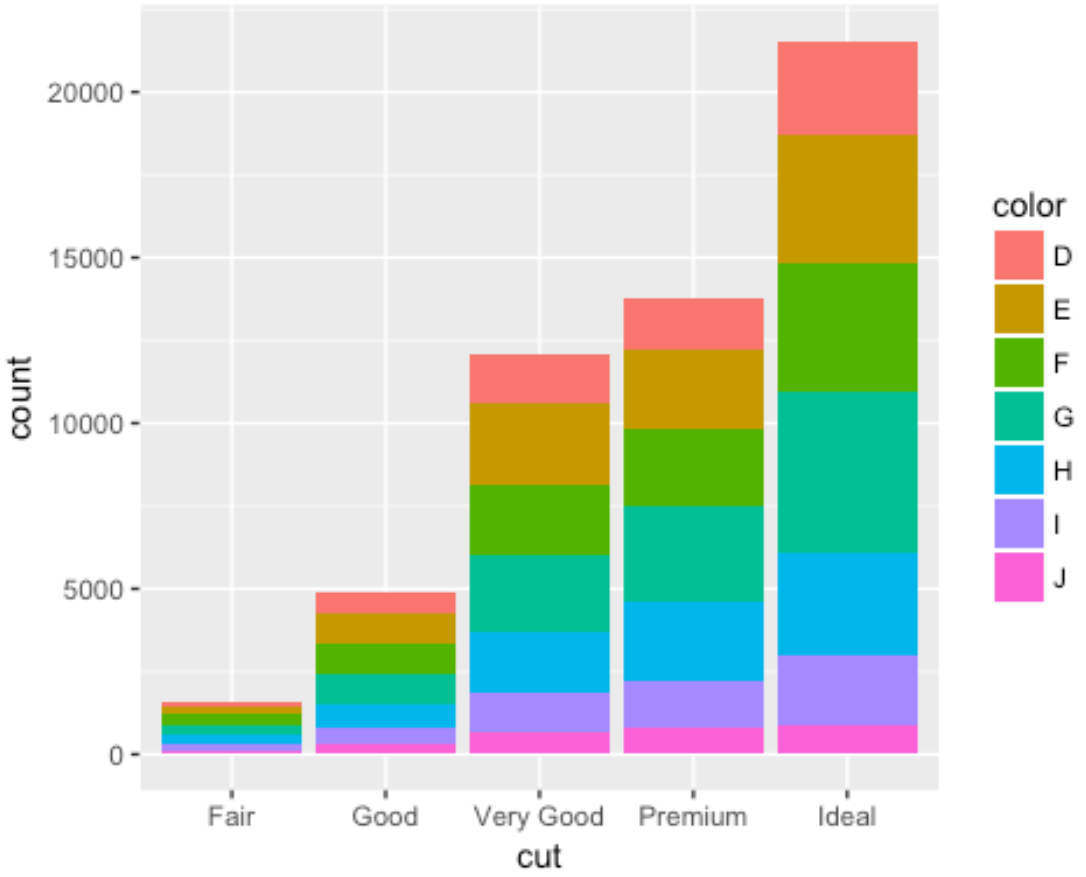

```
qplot(color,data=diamonds,fill=cut,position = "stack")
```

```
## Warning: `position` is deprecated
```



position identity - Don't adjust position

```
qplot(cut, data=diamonds, fill=color, position="fill")

## Warning: `position` is deprecated
```



position fill - Stack overlapping objects on top of one another, and standardise to have

Histograms

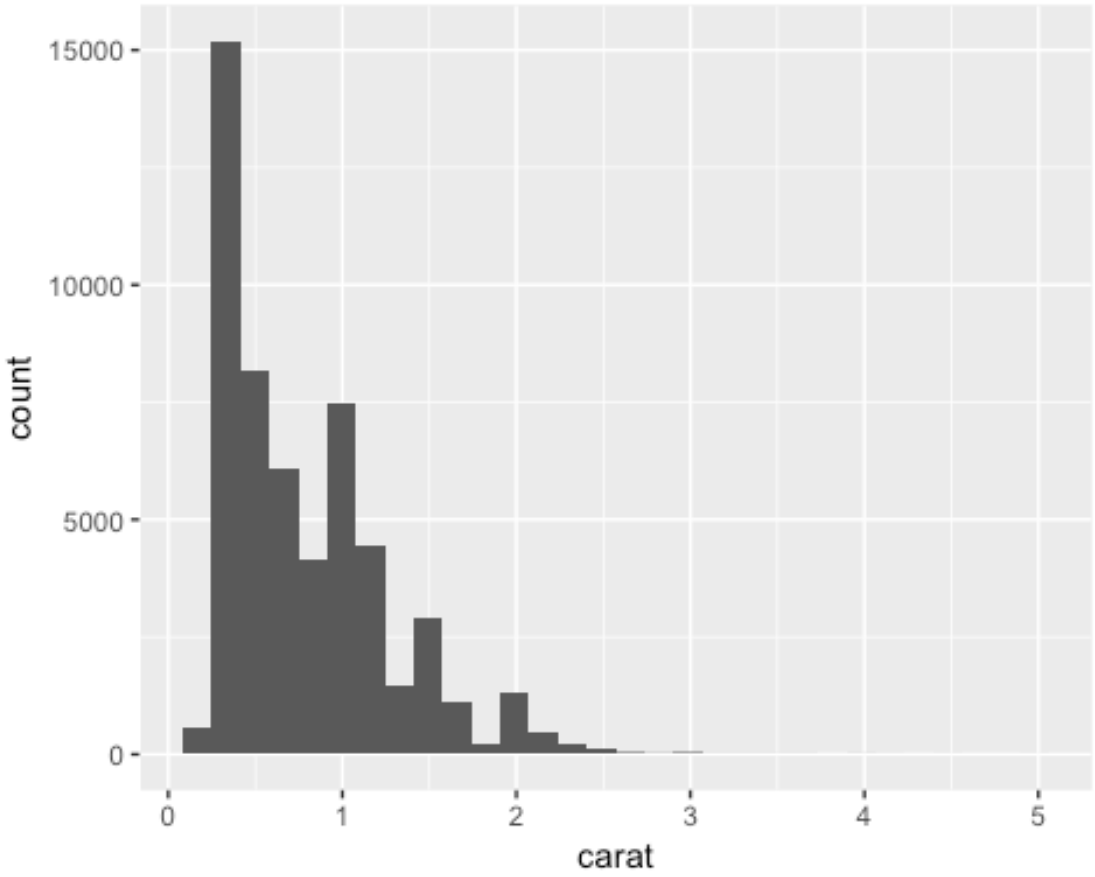
A histogram is like a bar chart but with continuous variables. To group with divide a continuous variable into intervals called *bins* then count the number of cases within each bin.

- use bars to reflect counts
- intervals on the horizontal axis
- counts on the vertical axis

A histogram is a form of density estimation. That is, the construction of an estimate, based on observed data, of an unobservable underlying probability density function.

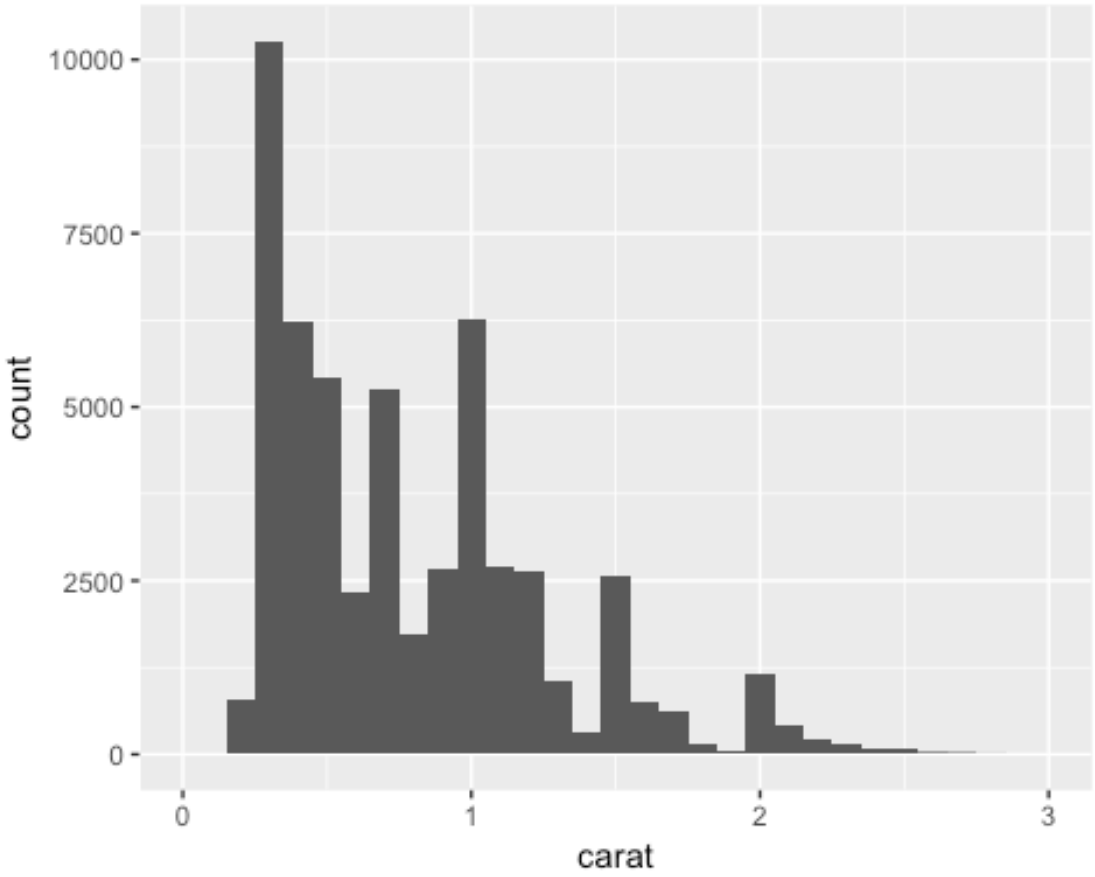
Histogram and density plots show the distribution of a single variable.

```
qplot(carat, data = diamonds, geom = "histogram")

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

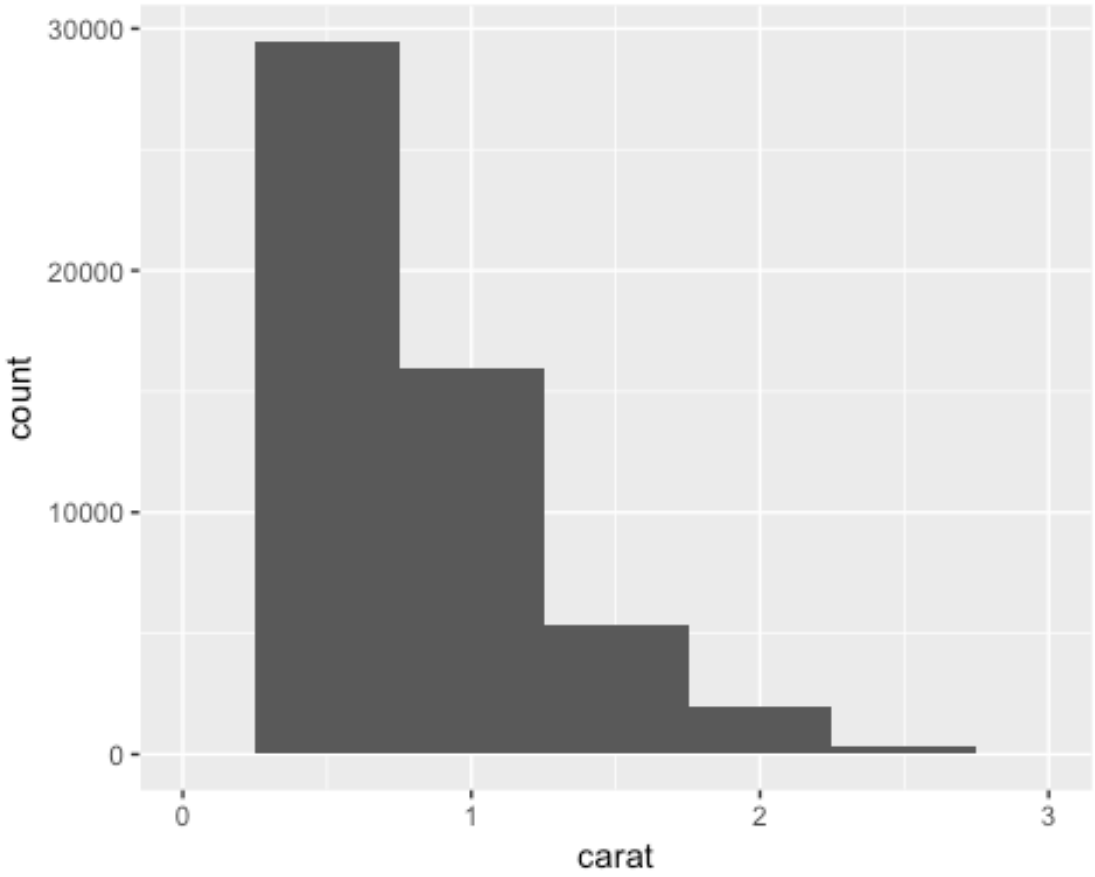


Playing with the histogram binwidth can reveal interesting aspects of the data.

```
qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.1,
  xlim = c(0,3))

## Warning: Removed 32 rows containing non-finite values (stat_bin).
```



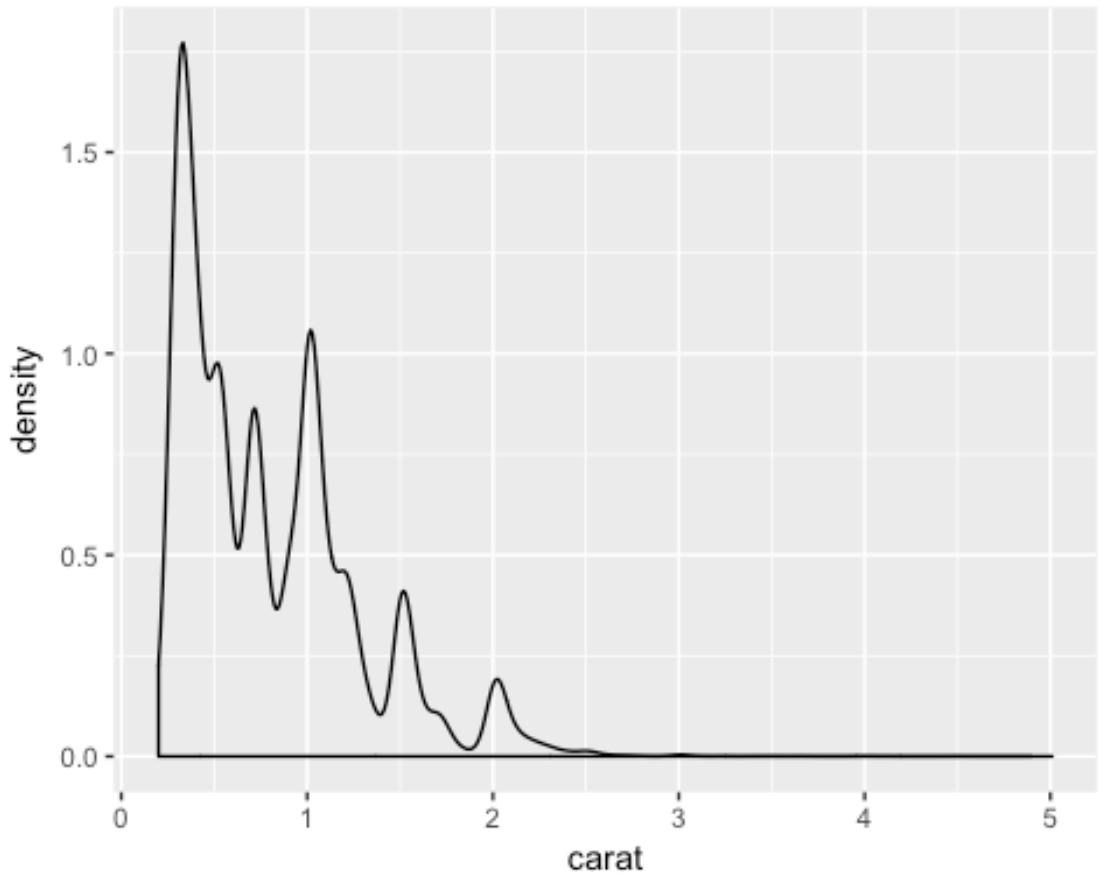

```
qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.5,
  xlim = c(0,3))

## Warning: Removed 32 rows containing non-finite values (stat_bin).
```



Density plots

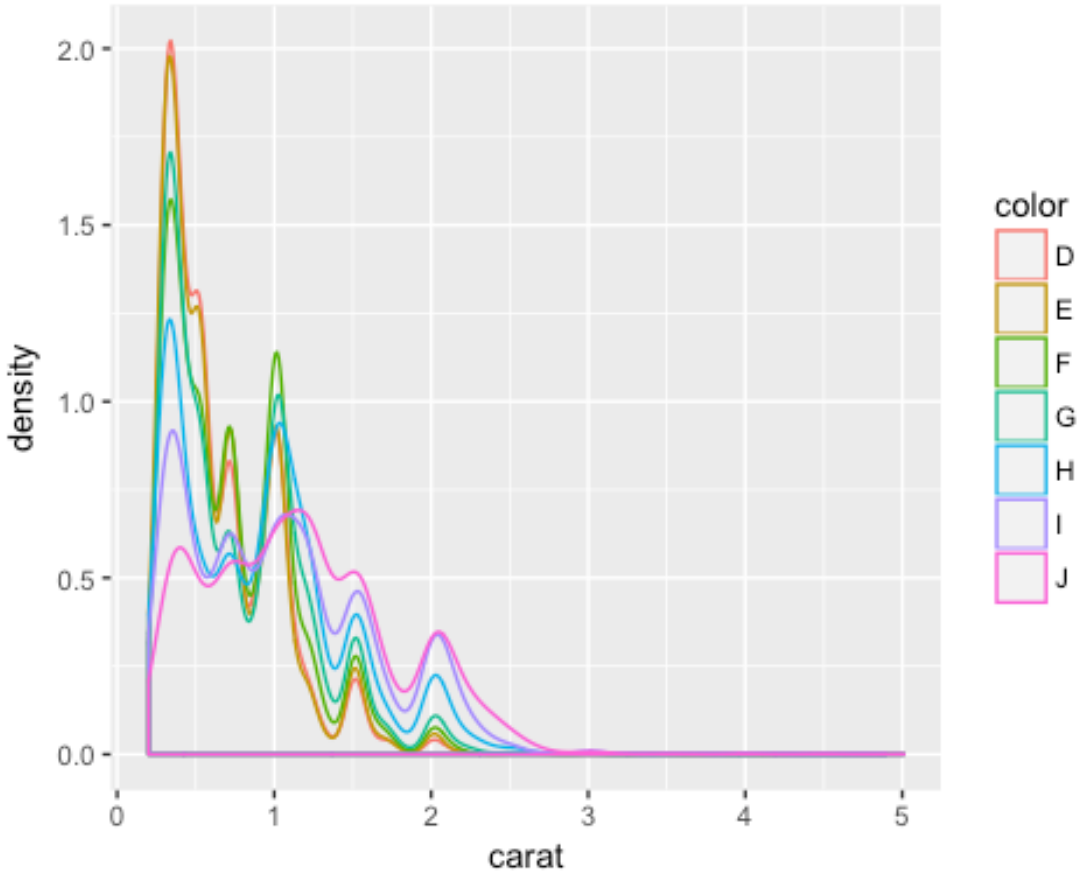
You can use probability densities instead of (or with) frequencies for density estimation

```
qplot(carat, data = diamonds, geom = "density")
```



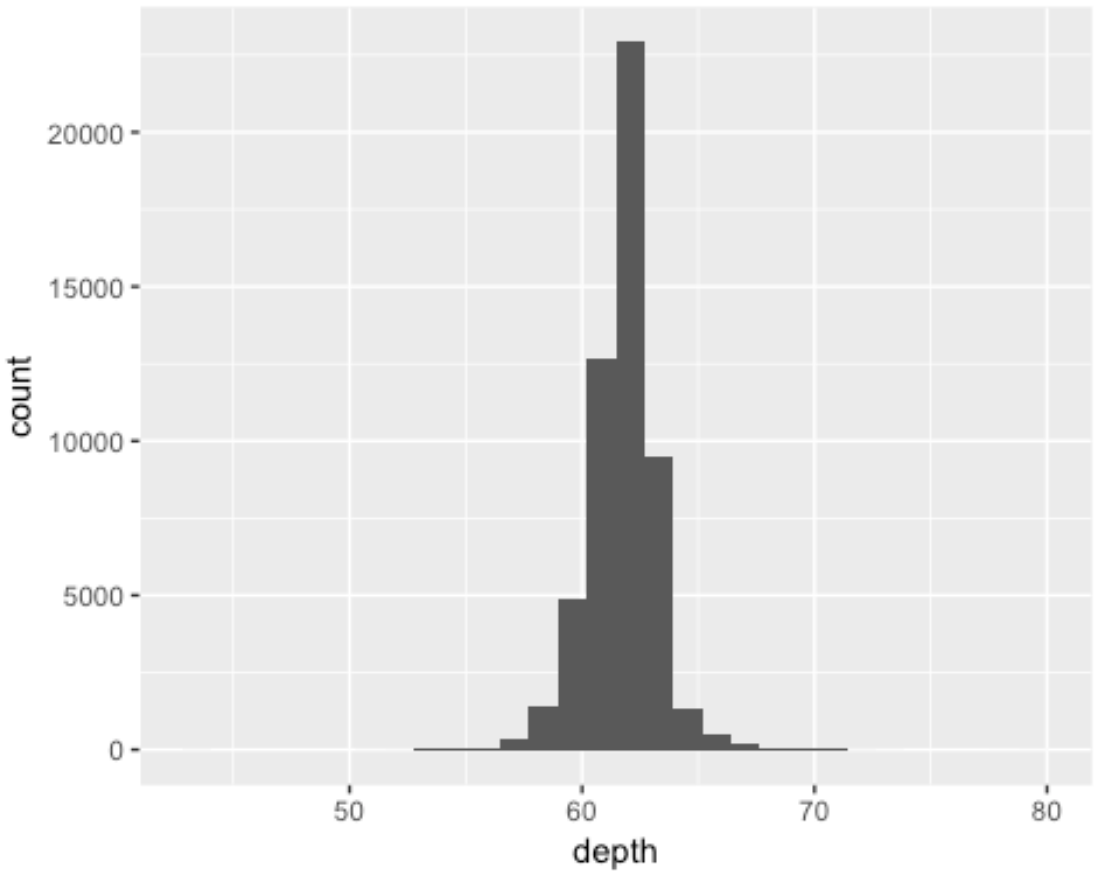
Mapping a categorical variable to an aesthetic

```
qplot(carat, data = diamonds, geom = "density", colour = color)
```




```
qplot(depth, data=diamonds)
```

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

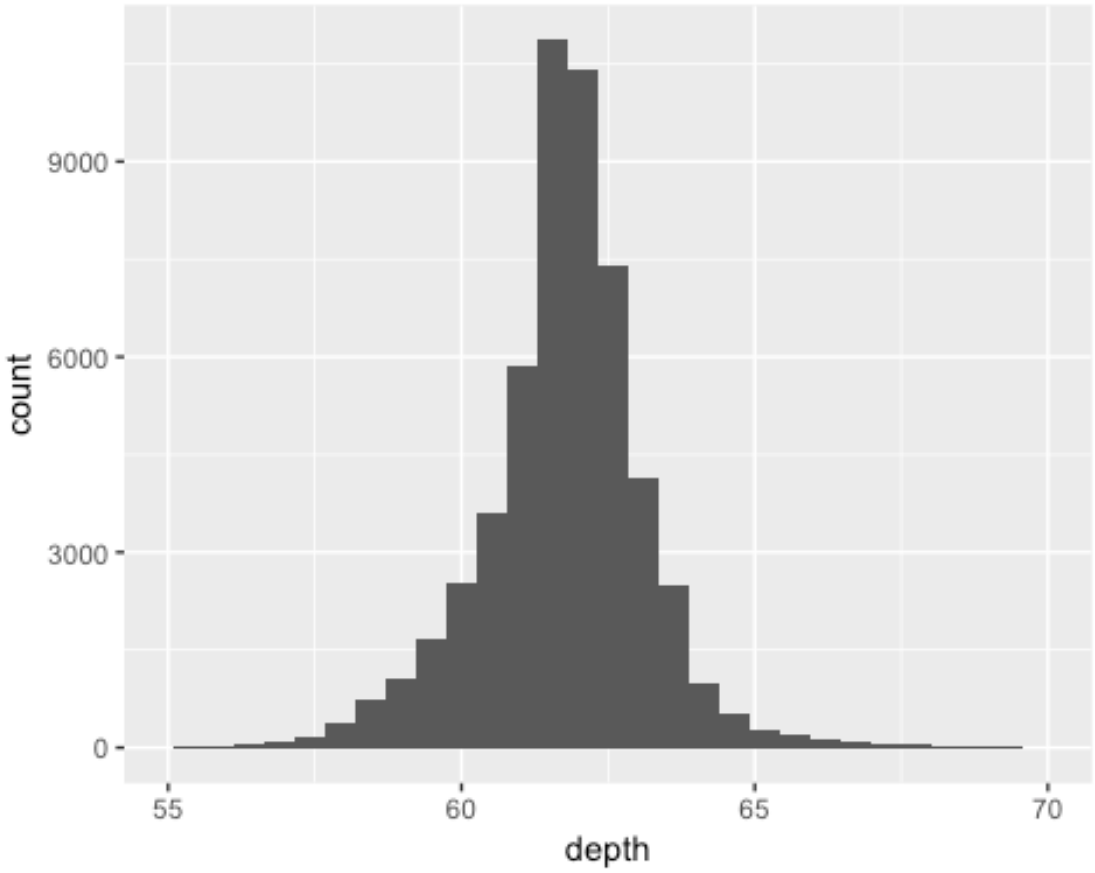



```
qplot(depth, data=diamonds) + xlim(55, 70)
```

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

```
## Warning: Removed 45 rows containing non-finite values (stat_bin).
```

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

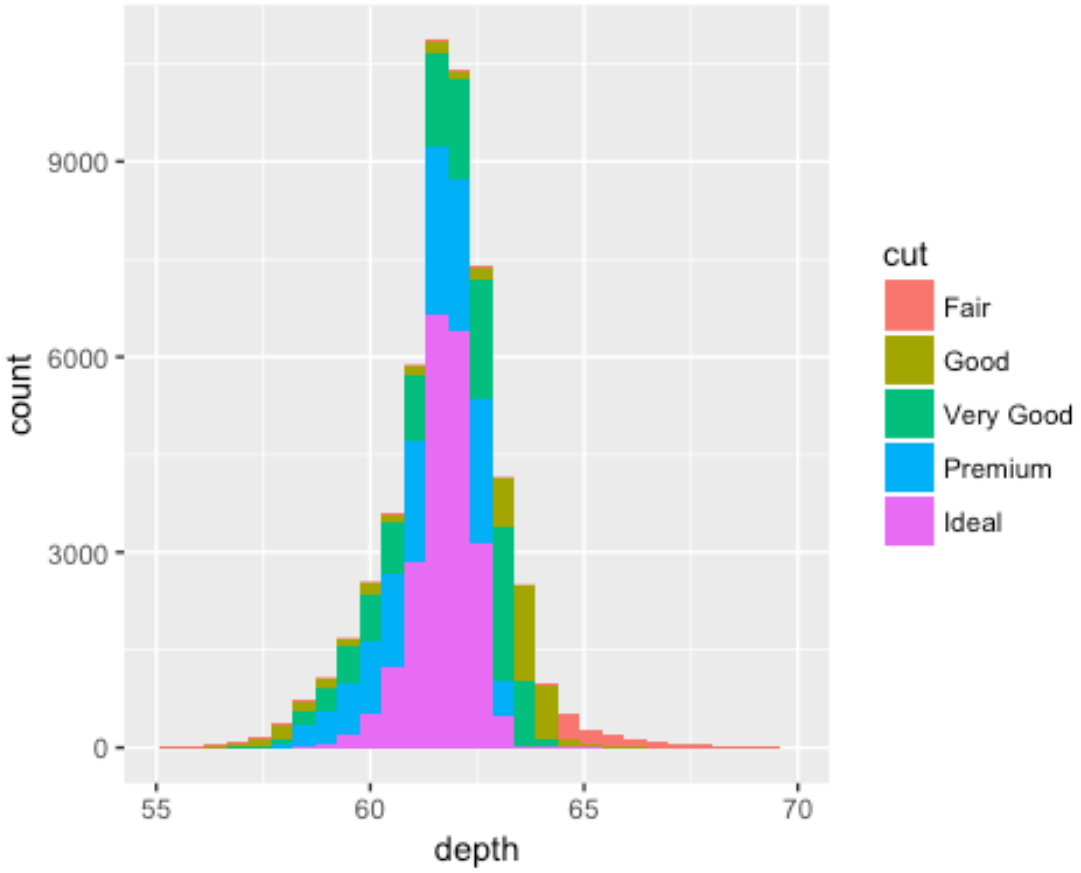



```
qplot(depth, data=diamonds, fill=cut) + xlim(55, 70)
```

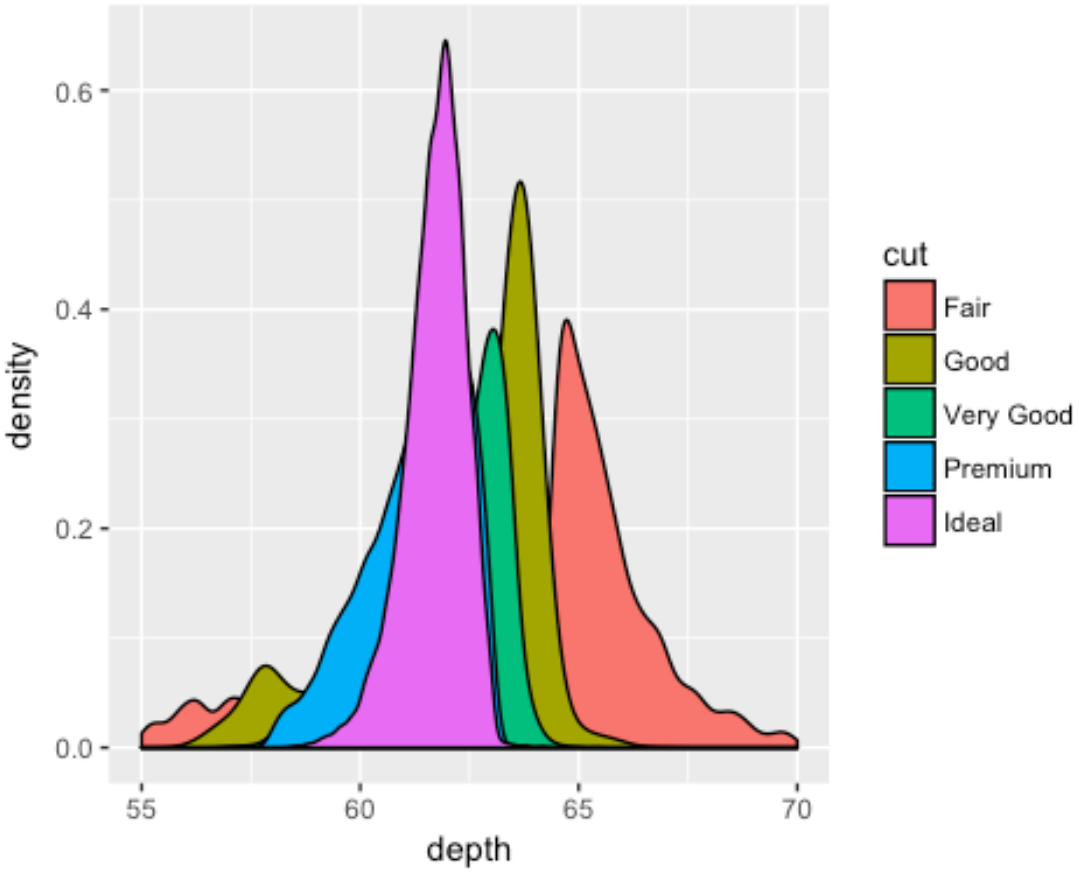
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 45 rows containing non-finite values (stat_bin).
```

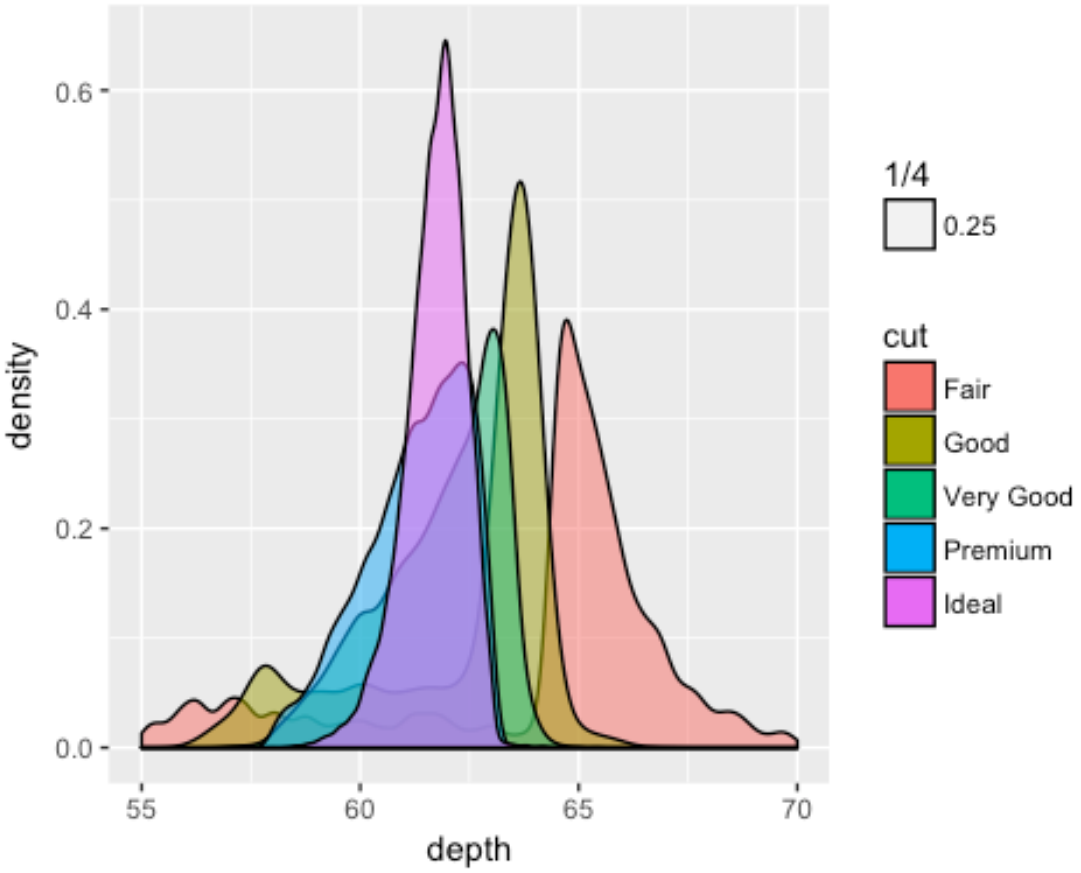
```
## Warning: Removed 5 rows containing missing values (geom_bar).
```



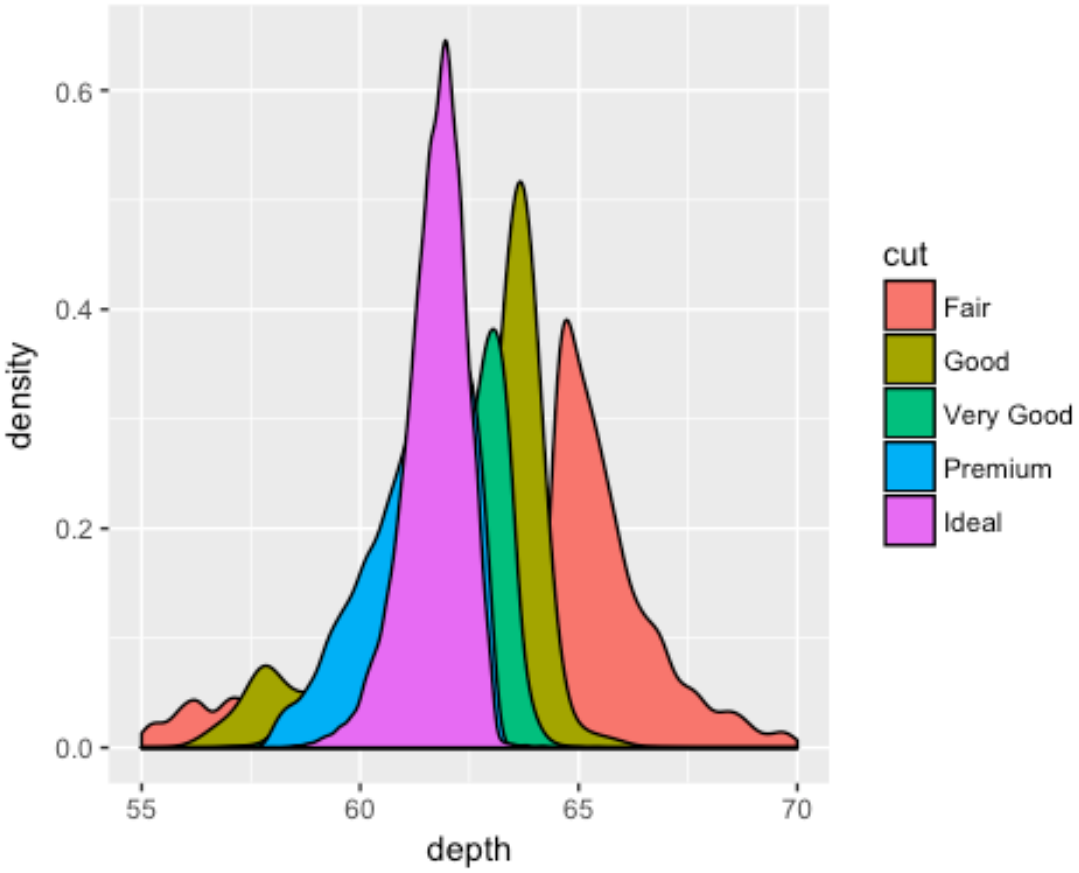

```
qplot(depth, data=diamonds, fill=cut, geom="density") + xlim(55, 70)

## Warning: Removed 45 rows containing non-finite values
(stat_density).
```




```
qplot(depth, data=diamonds, fill=cut, geom="density", alpha=1/4) +
xlim(55, 70)

## Warning: Removed 45 rows containing non-finite values
(stat_density).
```




```
qplot(depth, data=diamonds, fill=cut, geom="density", alpha=carat) +
xlim(55, 70)

## Warning: Removed 45 rows containing non-finite values
(stat_density).
```



ggplot2 help page

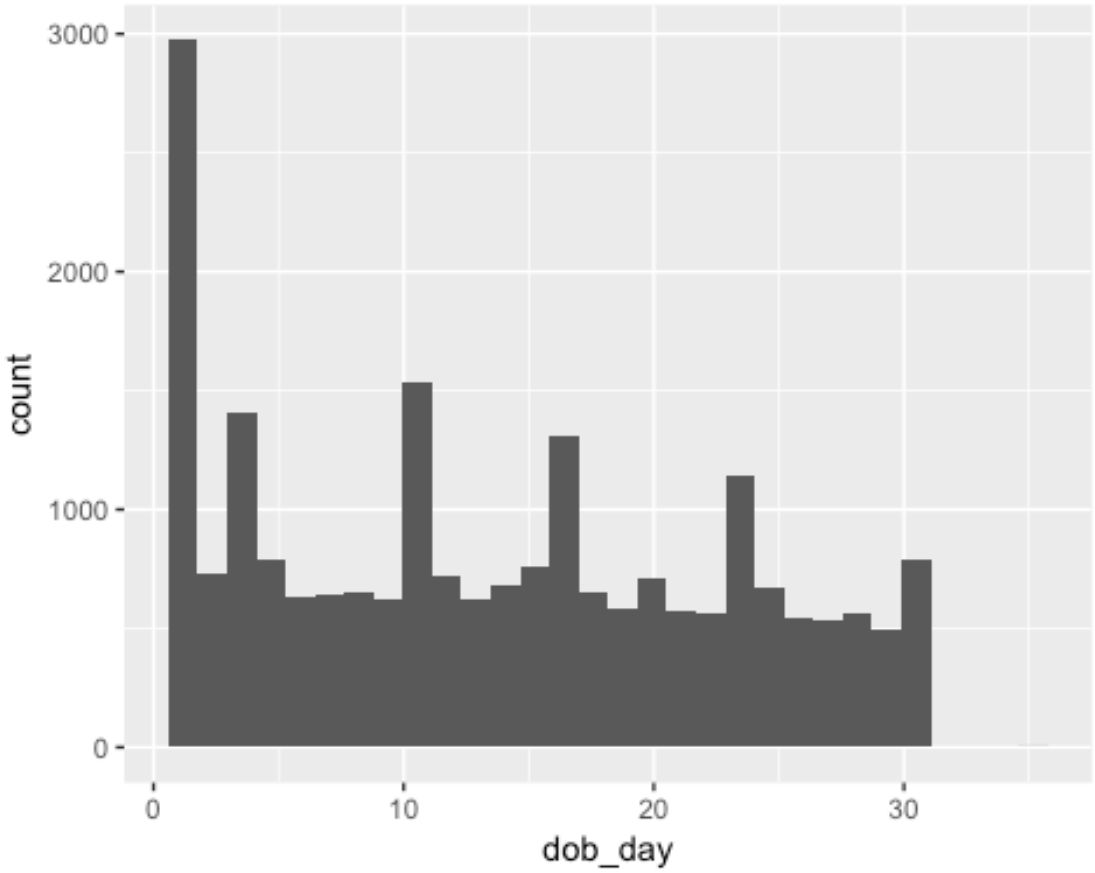
The ggplot2 help page is excellent. For more information on Geoms, Facets, Position adjustments, Themes, Aesthetics with many examples and much more go to the ggplot2 help page

Using Histograms & Density plots on the Quasi-Twitter data

Load and look at the Quasi-Twitter data.

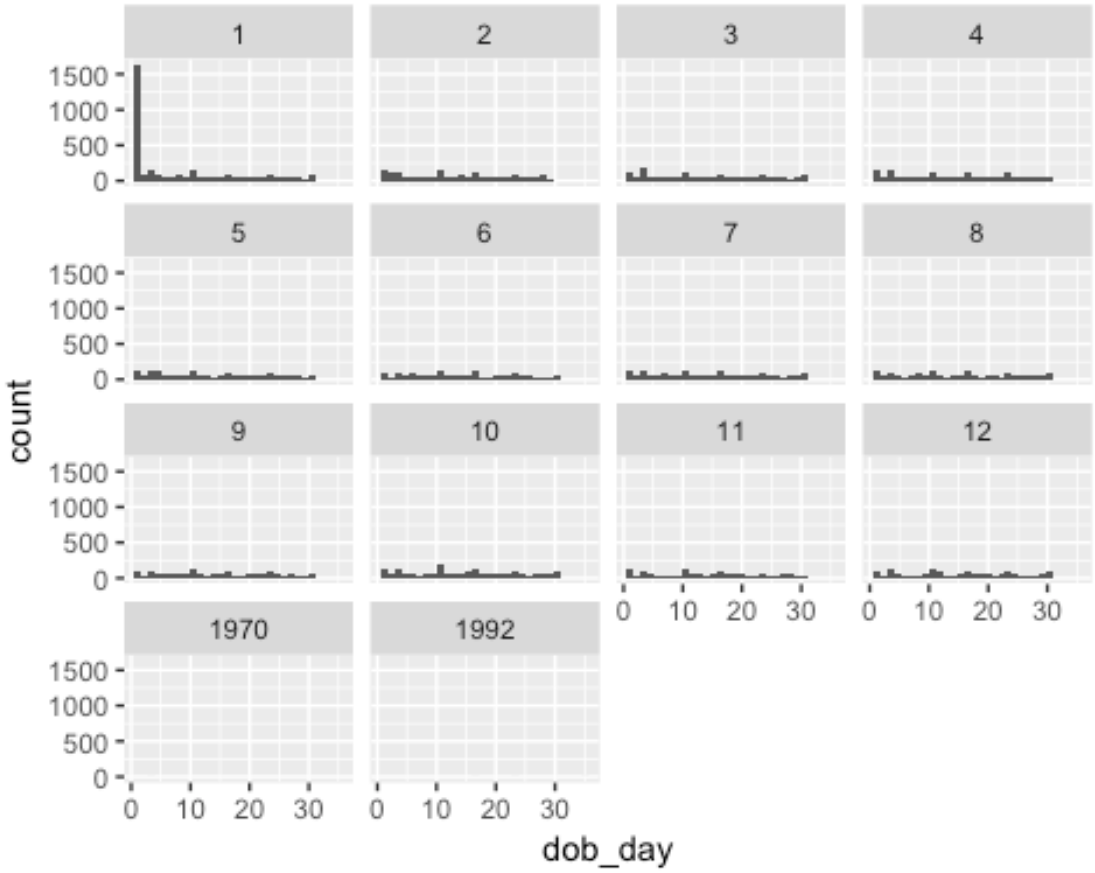
```
qplot(x=dob_day, data=twitter)
```

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



Notice January 1!

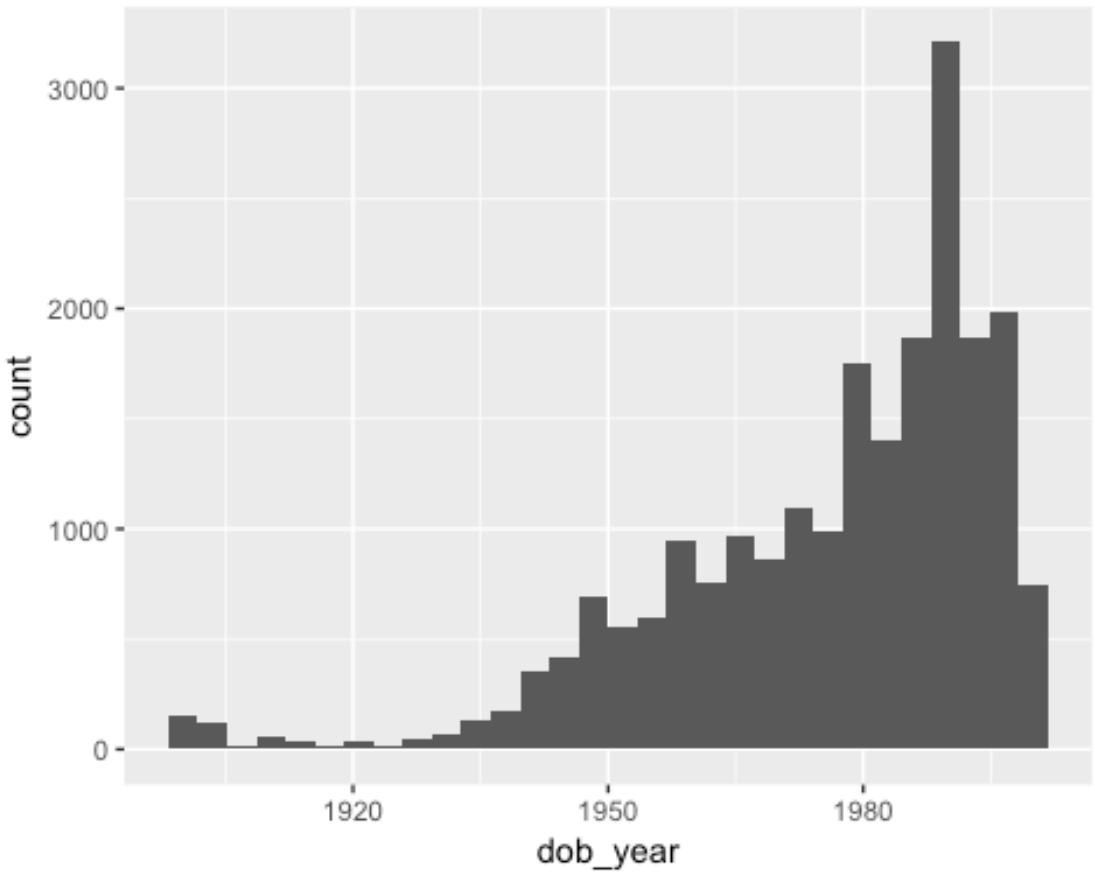
```
qplot(x=dob_day, data=twitter)+ facet_wrap(~dob_month) # a grid on one
extra variable

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



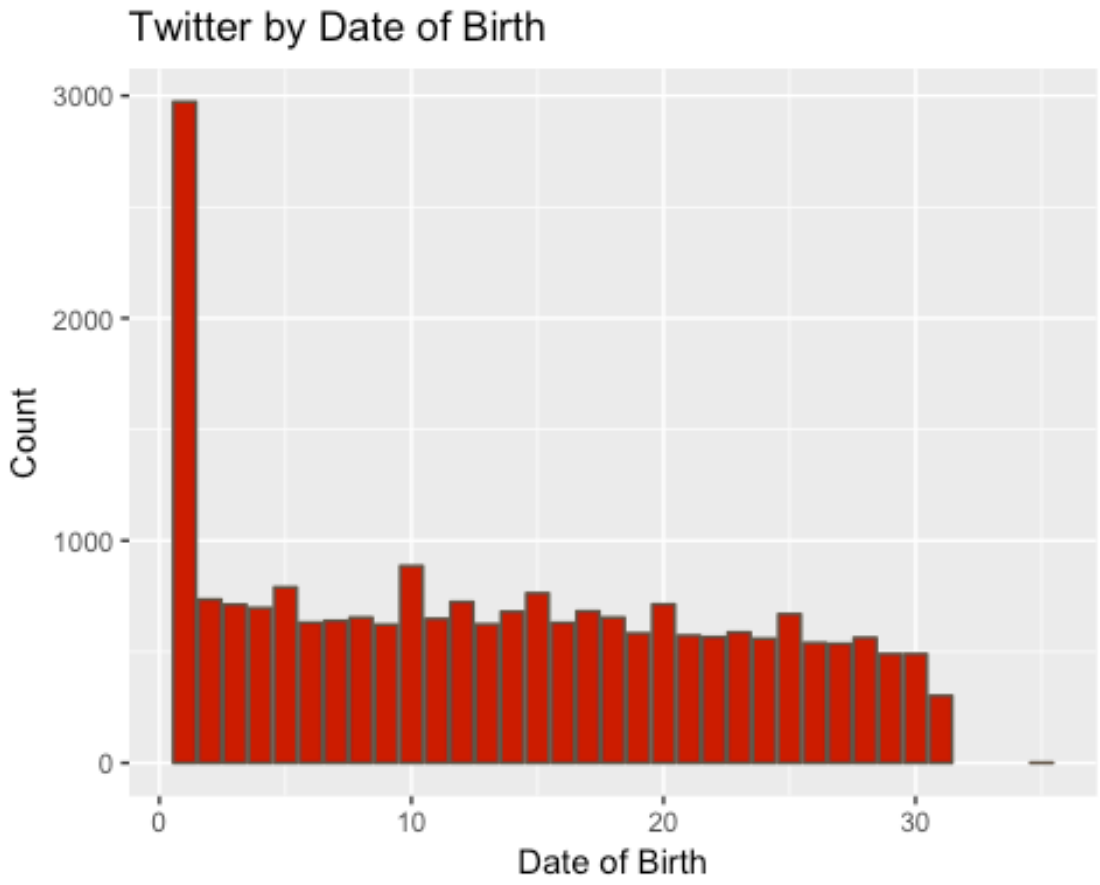
Bad data!!!

```
qplot(x=dob_year, data=twitter)
```

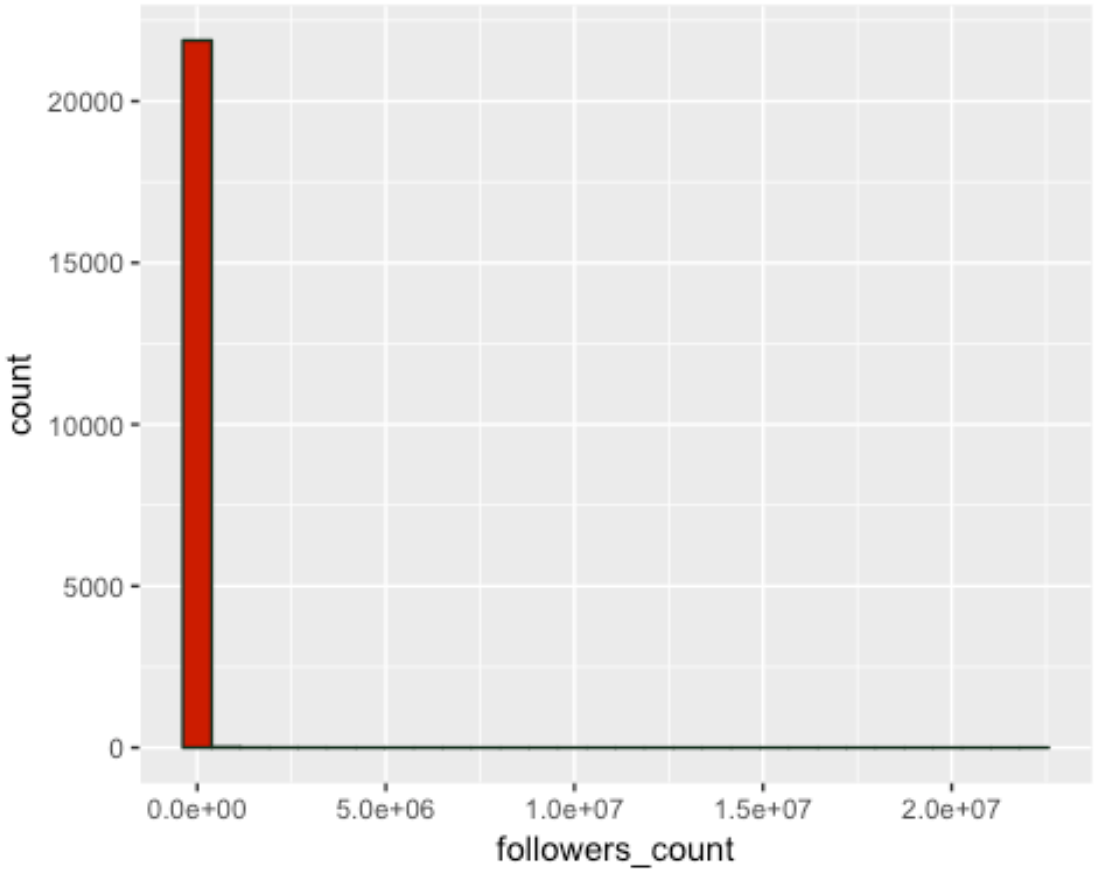
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(twitter) + aes(x=dob_day) + geom_bar(color=I('#615445'),
fill=I('#CC0000')) + labs(title="Twitter by Date of Birth", y="Count",
x="Date of Birth")
```

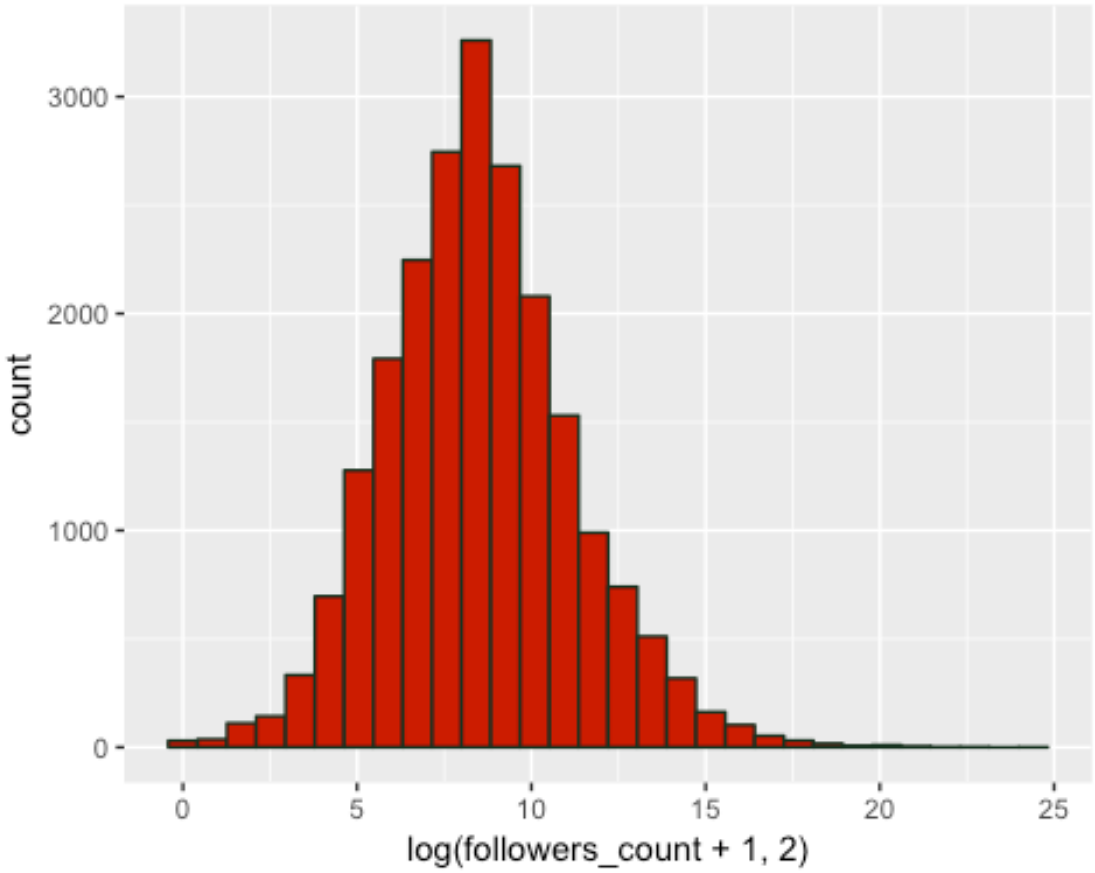



```
qplot(x=followers_count, data=twitter,color=I('#17331F'),
fill=I('#CC0000'))

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



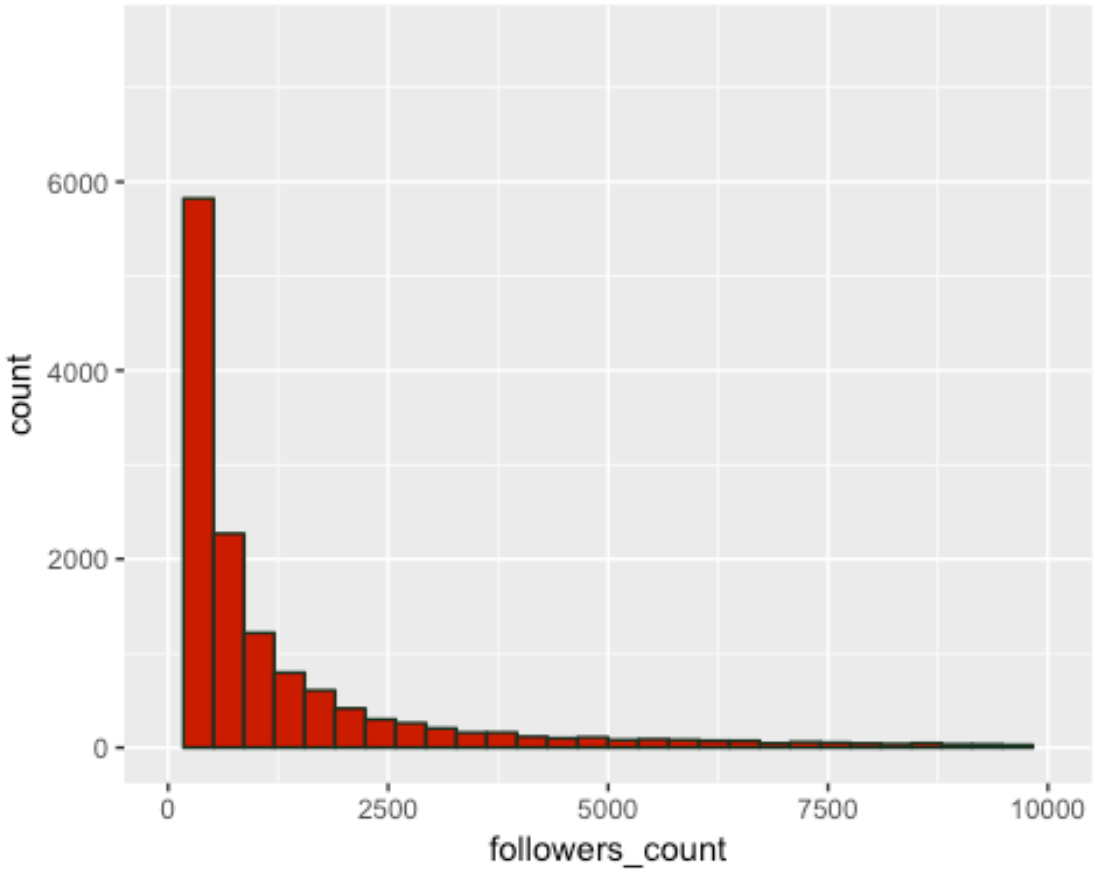
Fat-tailed! Log-transform.

```
qplot(x=log(followers_count+1,2), data=twitter,color=I('#17331F'),
fill=I('#CC0000'))

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

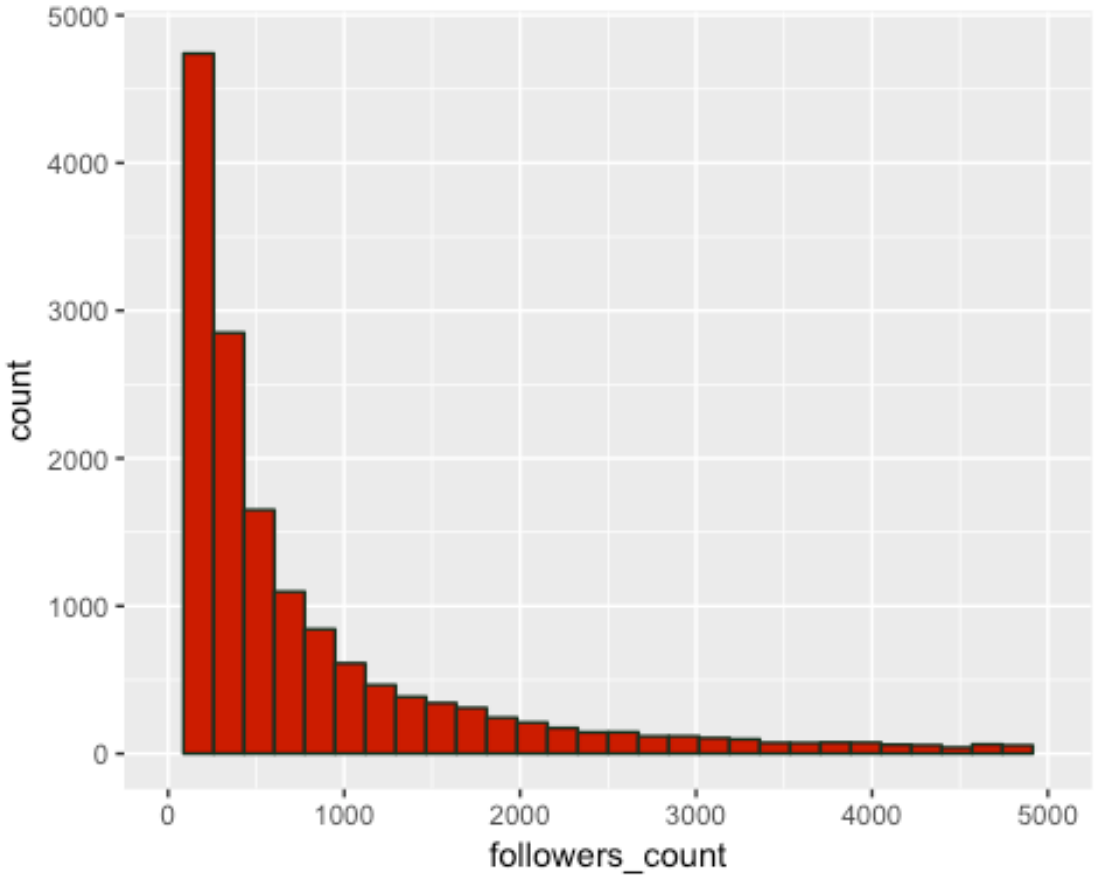


Zoom in?

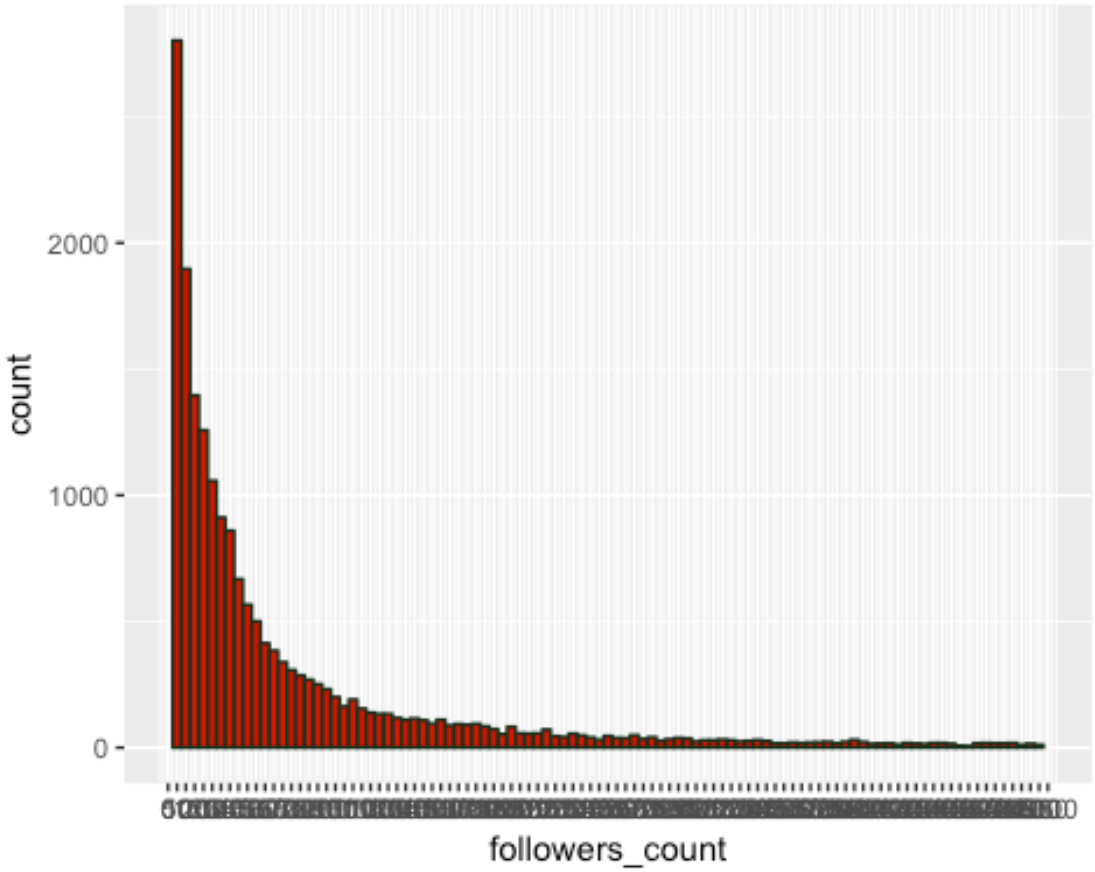
```
qplot(x=followers_count,
data=twitter,xlim=c(0,10000),color=I('#17331F'), fill=I('#CC0000'))

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

## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```



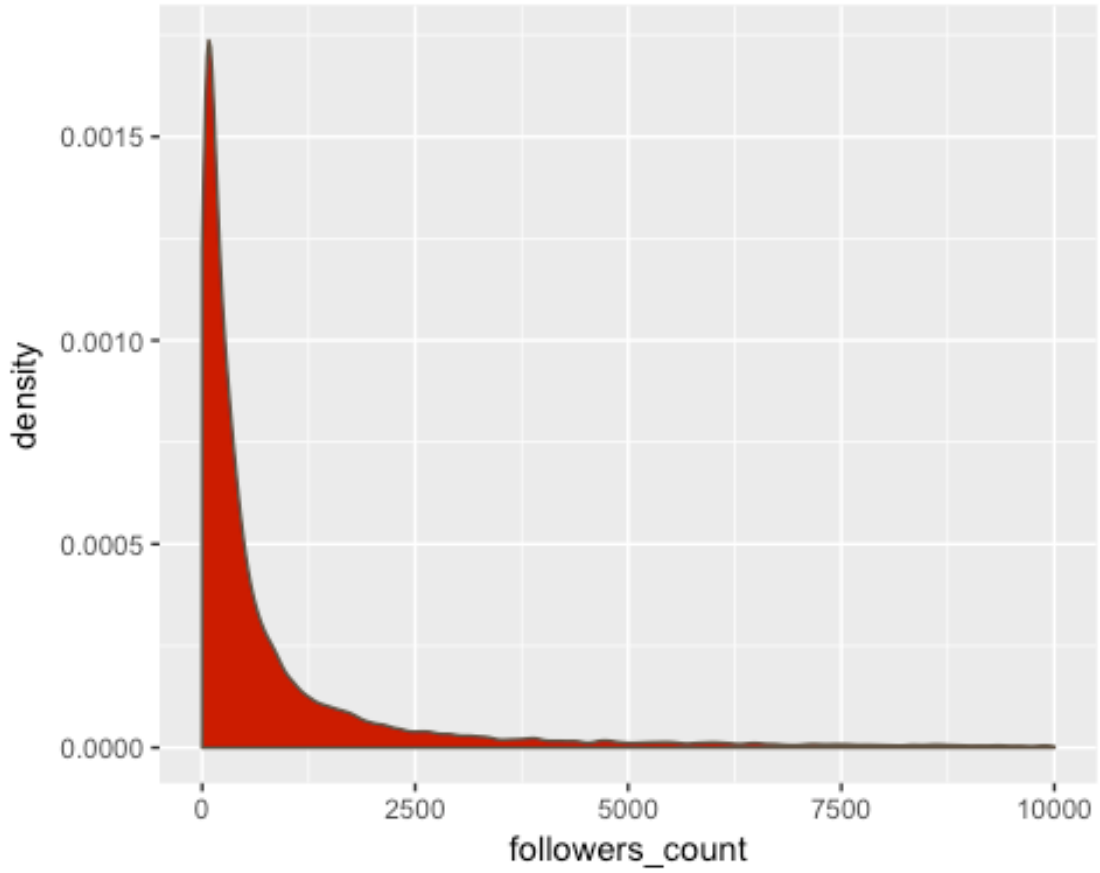

```
qplot(x=followers_count, data=twitter,color=I('#17331F'),
fill=I('#CC0000'))+scale_x_continuous(limits=c(0,5000))

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

## Warning: Removed 1870 rows containing non-finite values (stat_bin).
```




```
qplot(x=followers_count, data=twitter, binwidth=50,color=I('#17331F'),
fill=I('#CC0000'))+scale_x_continuous(limits=c(0,5000),
breaks=seq(0,5000,50))

## Warning: Removed 1870 rows containing non-finite values (stat_bin).
```




```
qplot(x=followers_count, data=twitter,xlim=c(0,10000),geom="density",
color=I('#615445'), fill=I('#CC0000'))

## Warning: Removed 1055 rows containing non-finite values
(stat_density).
```

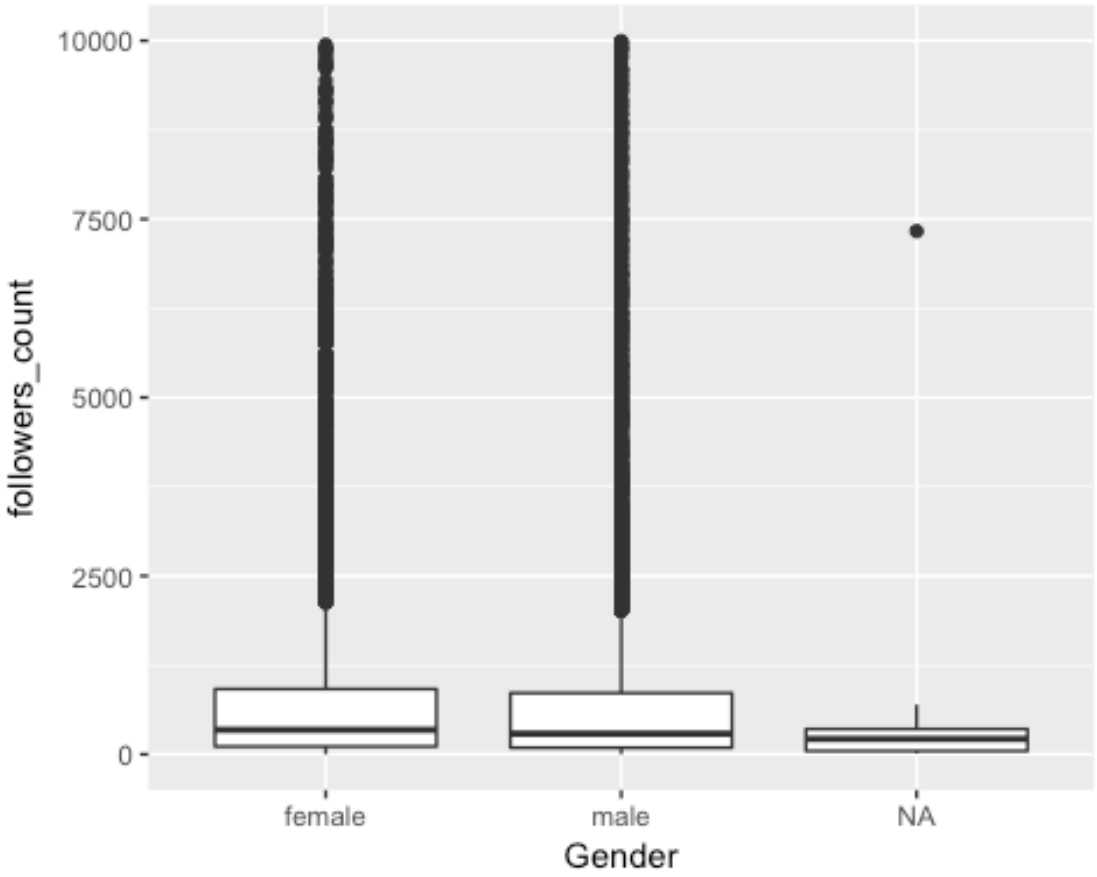



```
by(twitter$followers_count,twitter$gender,summary)
```

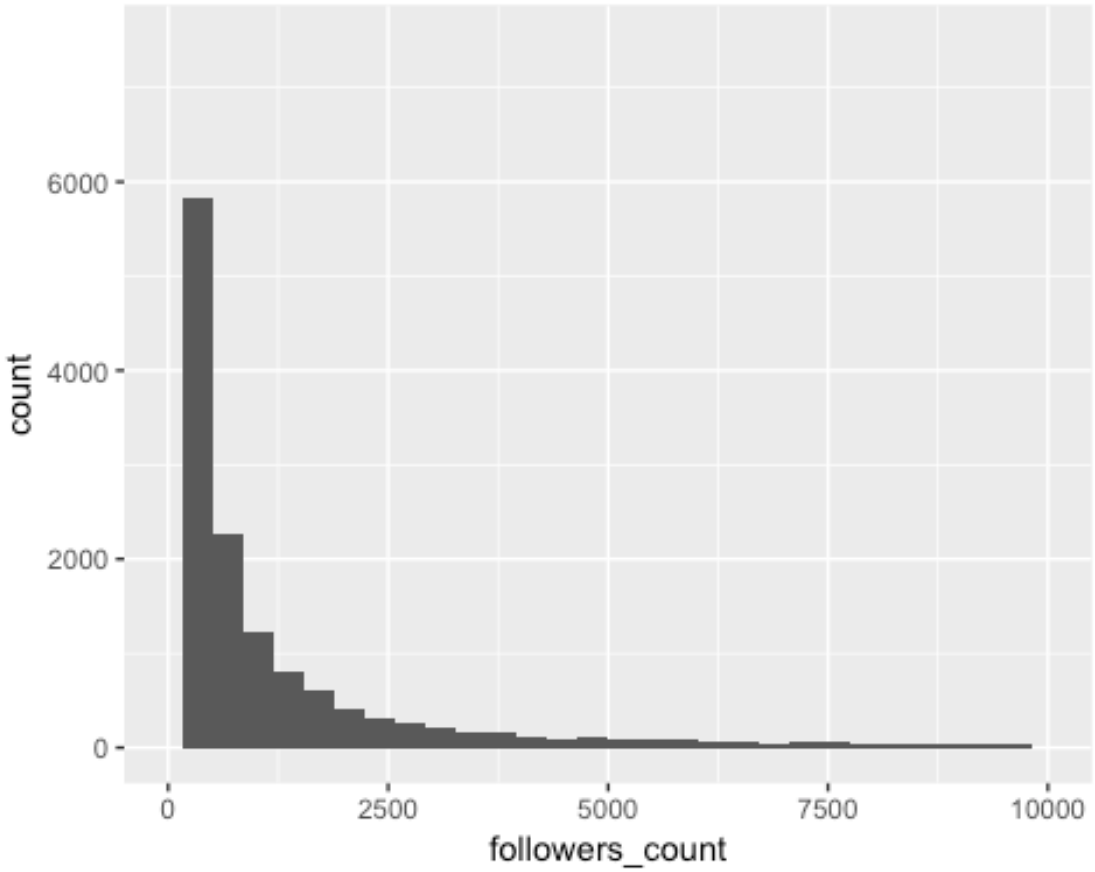
```
## twitter$gender: female
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        0      118      372     8580     1156 22190000
## ------------------------------------------------------------
## twitter$gender: male
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      98     316    4503    1050 6257000
```

```
qplot(gender,followers_count, data =twitter,
geom="boxplot",ylim=c(0,10000))+xlab("Gender")
```

```
## Warning: Removed 1055 rows containing non-finite values
(stat_boxplot).
```



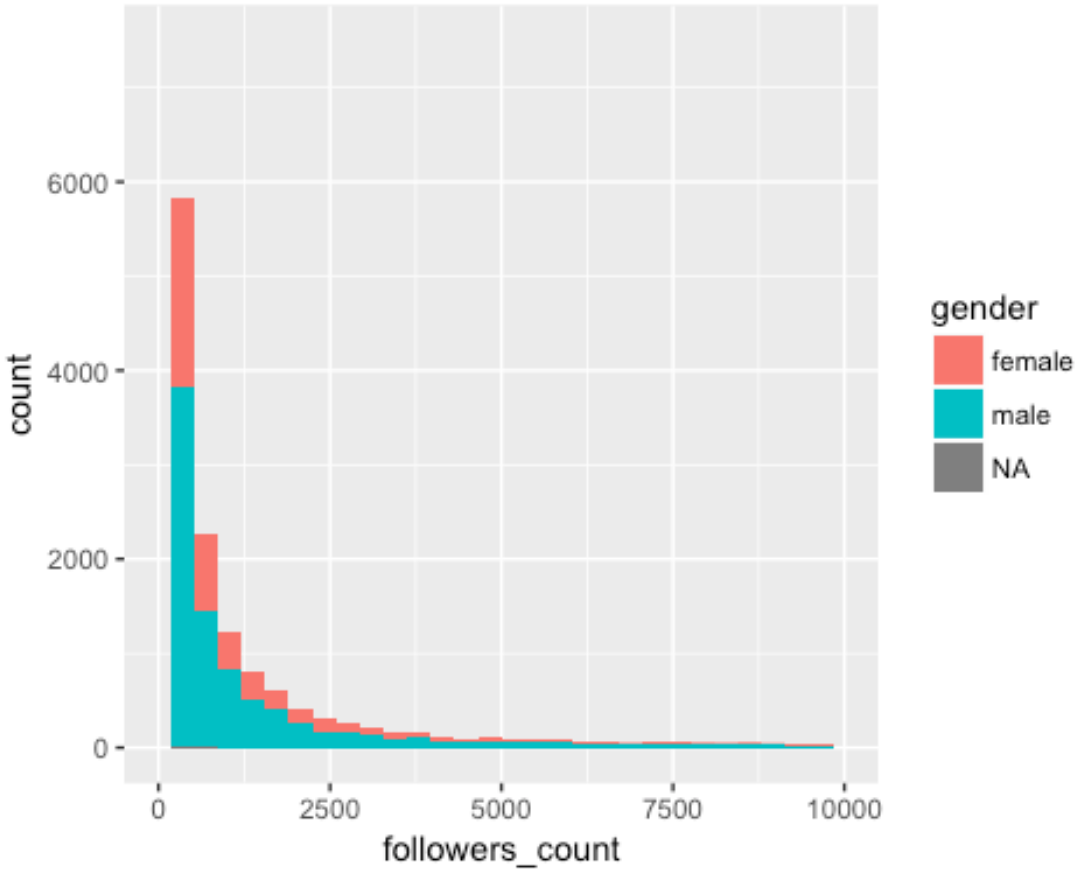

```
qplot(followers_count, data=twitter,xlim=c(0,10000))

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

## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```



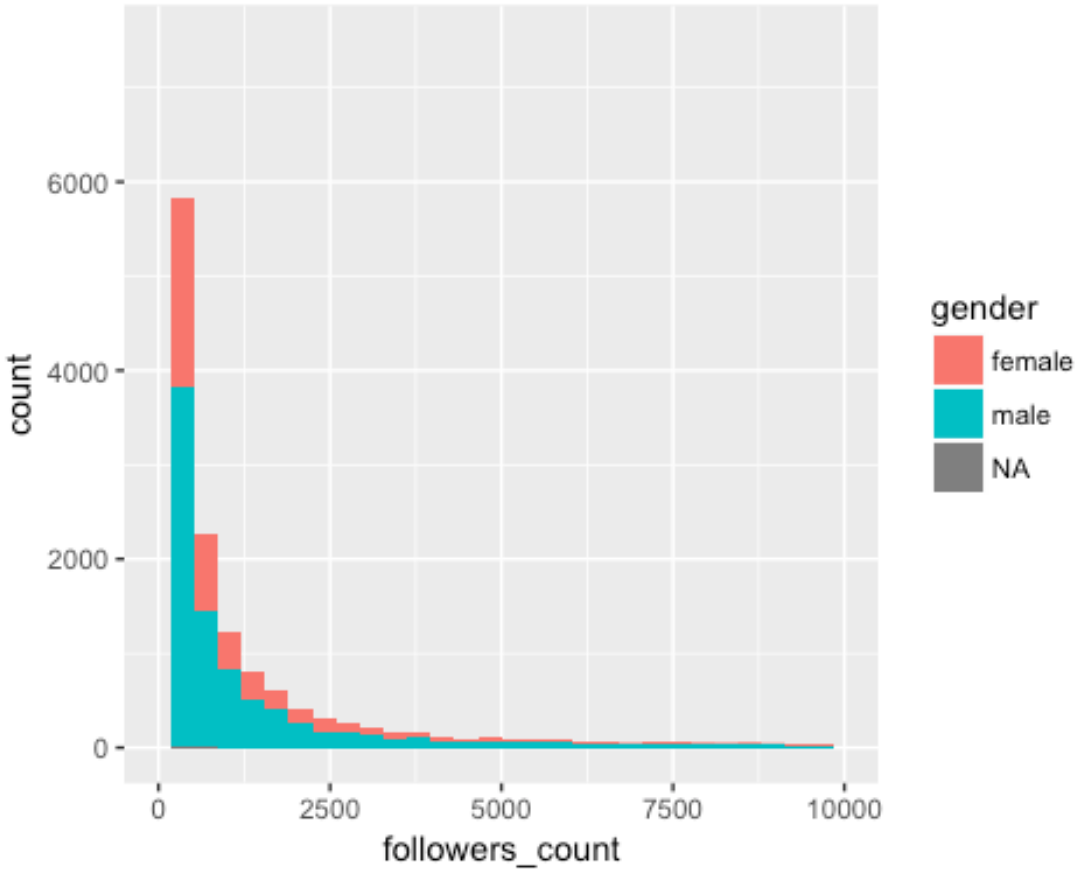

```
qplot(followers_count, data=twitter, fill=gender,xlim=c(0,10000))
```

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

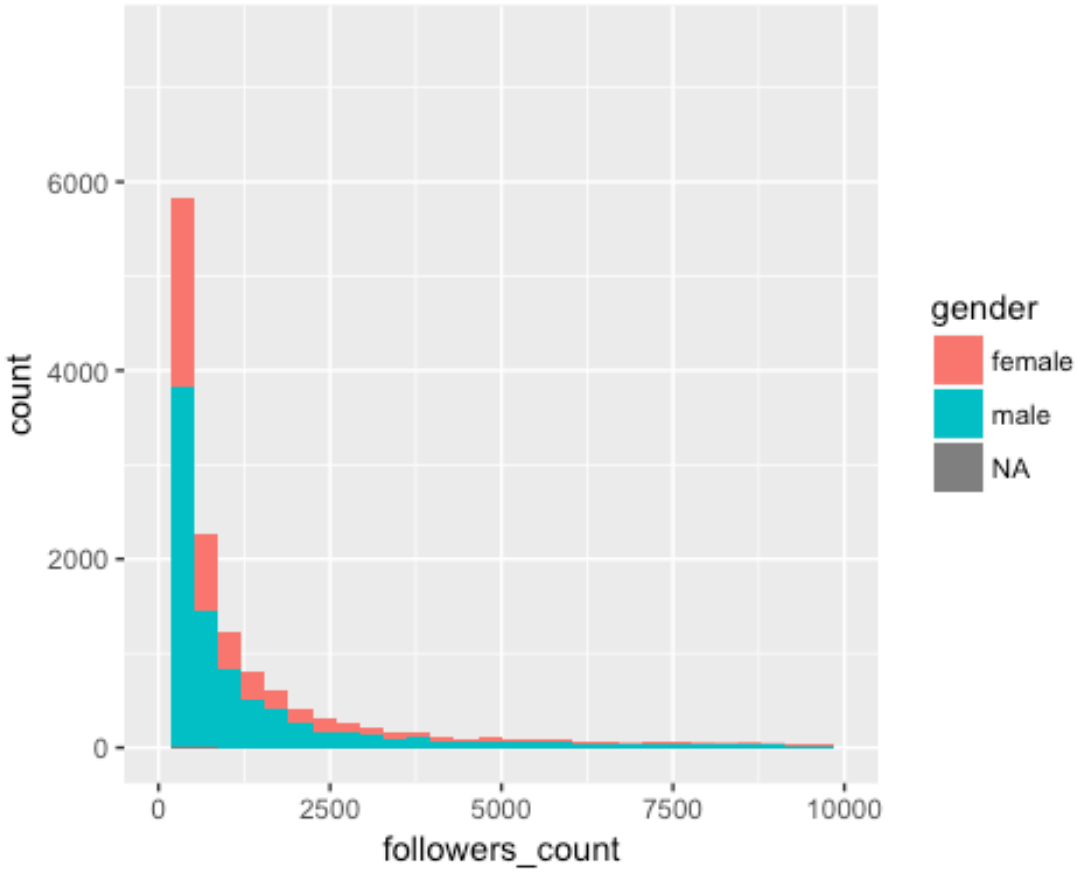
```
## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```



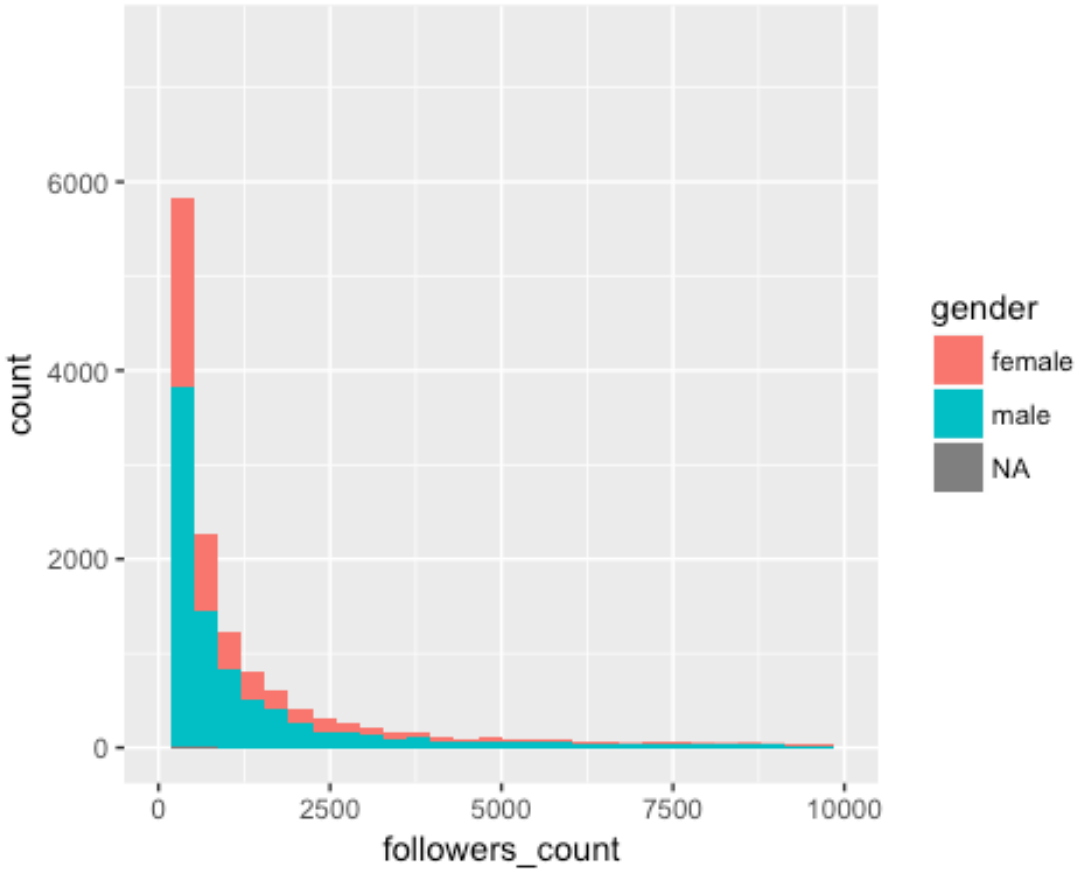

```
qplot(followers_count, data=twitter, fill=gender,
position="stack",xlim=c(0,10000))

## Warning: `position` is deprecated

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

## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```



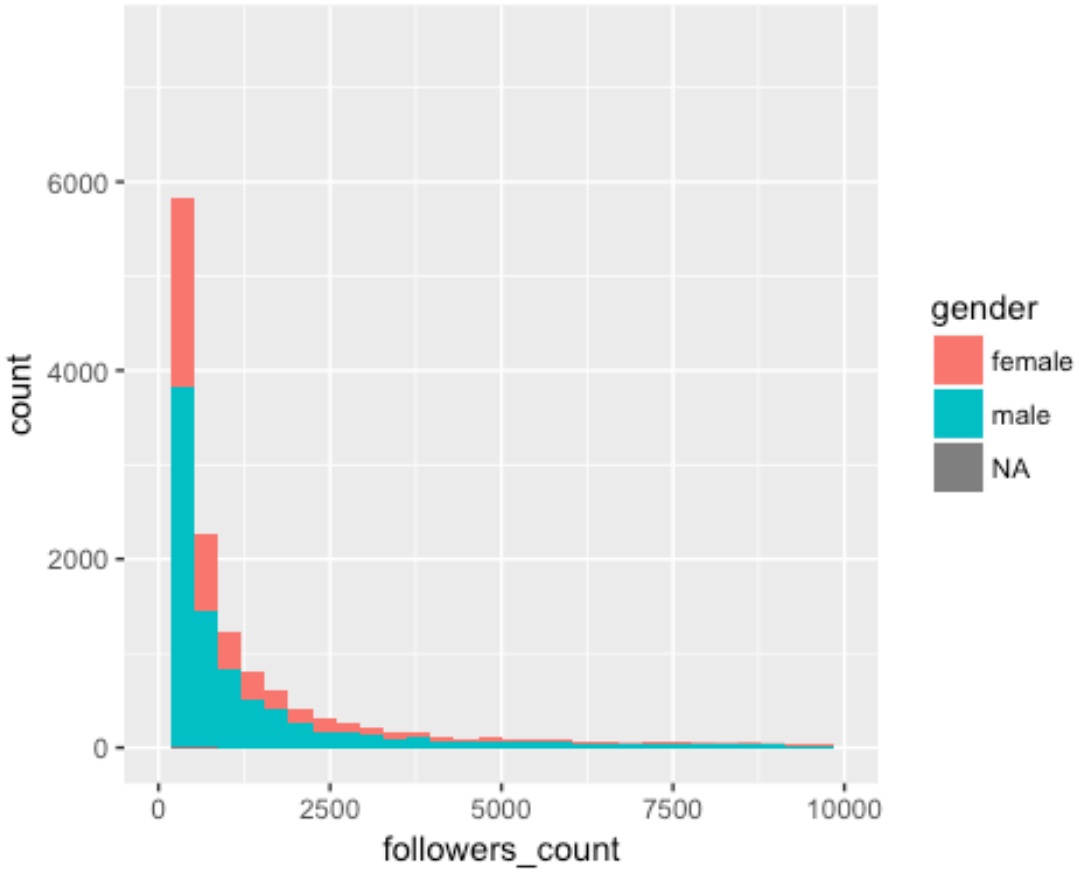

```
qplot(followers_count, data=twitter, fill=gender,
position="dodge",xlim=c(0,10000))

## Warning: `position` is deprecated

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

## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```



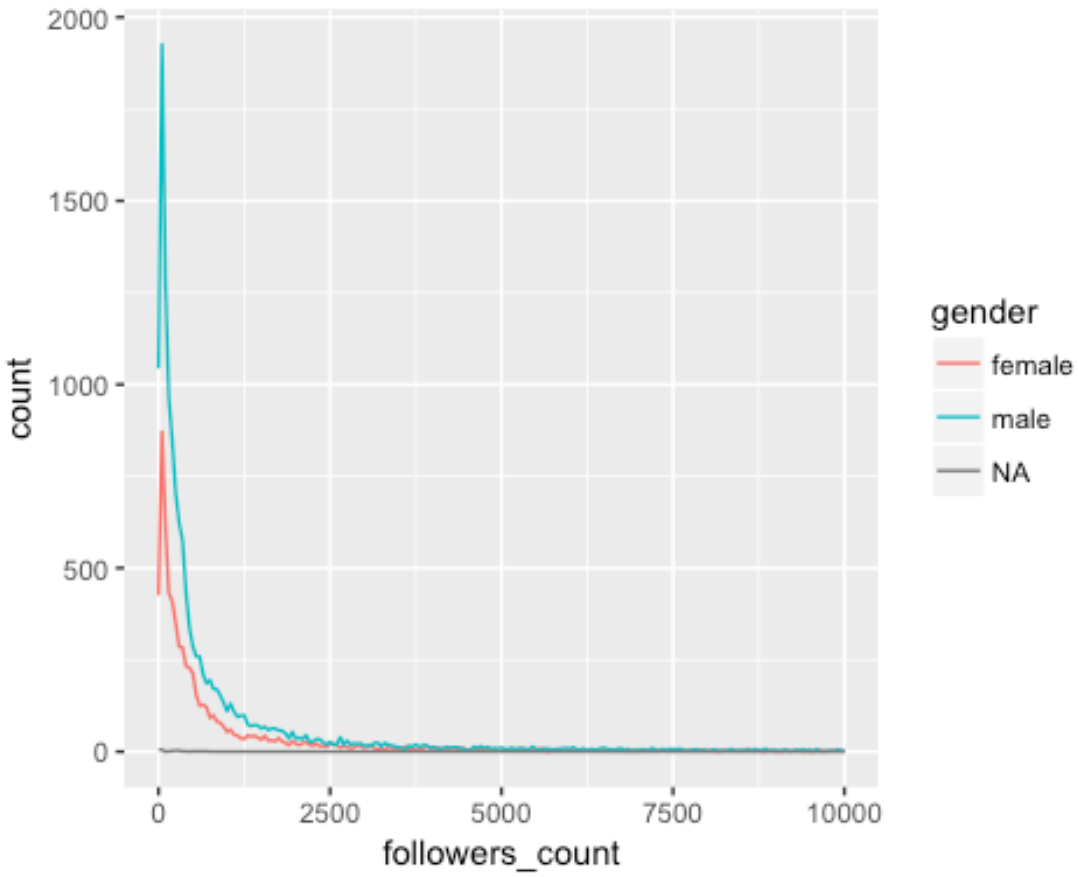

```
qplot(followers_count, data=twitter, fill=gender,
position="identity",xlim=c(0,10000))

## Warning: `position` is deprecated

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

## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```



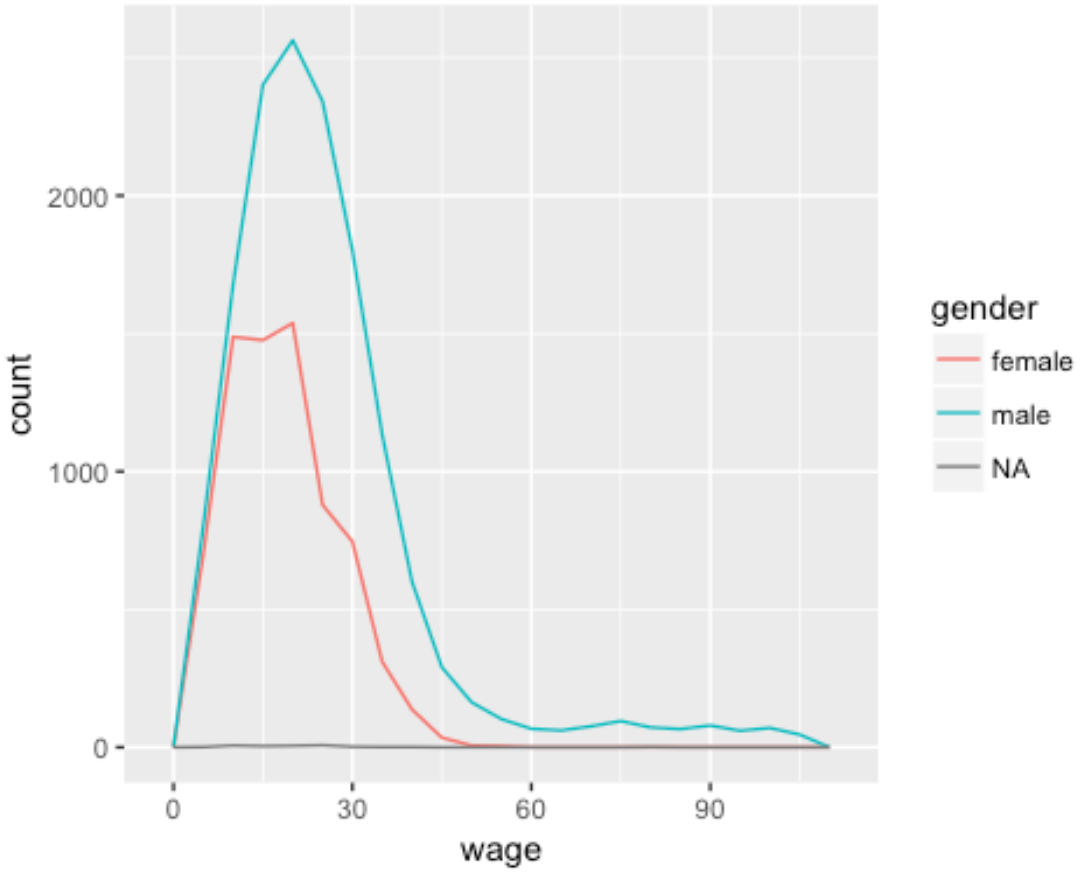

```
qplot(followers_count, data=twitter, fill=gender,
position="fill",xlim=c(0,10000))

## Warning: `position` is deprecated

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

## Warning: Removed 1055 rows containing non-finite values (stat_bin).
```




```
qplot(followers_count, data=twitter, binwidth=50, xlim=c(0,10000),
geom="freqpoly", color=gender)

## Warning: Removed 1055 rows containing non-finite values (stat_bin).

## Warning: Removed 6 rows containing missing values (geom_path).
```




```
qplot(wage, data=twitter, binwidth=5, geom="freqpoly", color=gender)
```



Line plots

Line and path plots are typically used for time series data. Line plots join the points from left to right, while path plots join them in the order that they appear in the dataset (a line plot is just a path plot of the data sorted by x value).

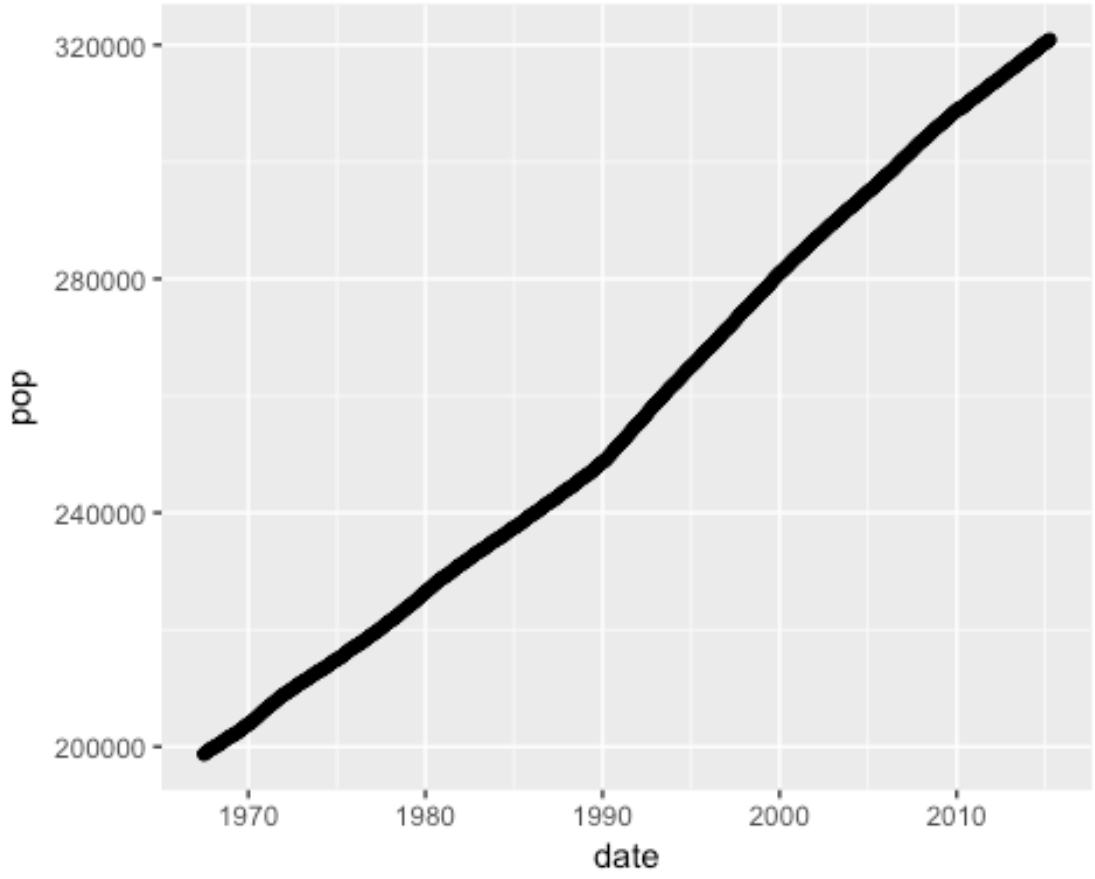
To show this we'll use the ggplot2 economics dataset, which contains economic time-series data on the US measured over the last 40 years.

```
data(economics)
str(economics)

## Classes 'tbl_df', 'tbl' and 'data.frame':    574 obs. of  6
variables:
##  $ date    : Date, format: "1967-07-01" "1967-08-01" ...
##  $ pce     : num  507 510 516 513 518 ...
##  $ pop     : int  198712 198911 199113 199311 199498 199657 199808
199920 200056 200208 ...
##  $ psavert : num  12.5 12.5 11.7 12.5 12.5 12.1 11.7 12.2 11.6 12.2
...
##  $ uempmed : num  4.5 4.7 4.6 4.9 4.7 4.8 5.1 4.5 4.1 4.6 ...
##  $ unemploy: int  2944 2945 2958 3143 3066 3018 2878 3001 2877 2709
...
```

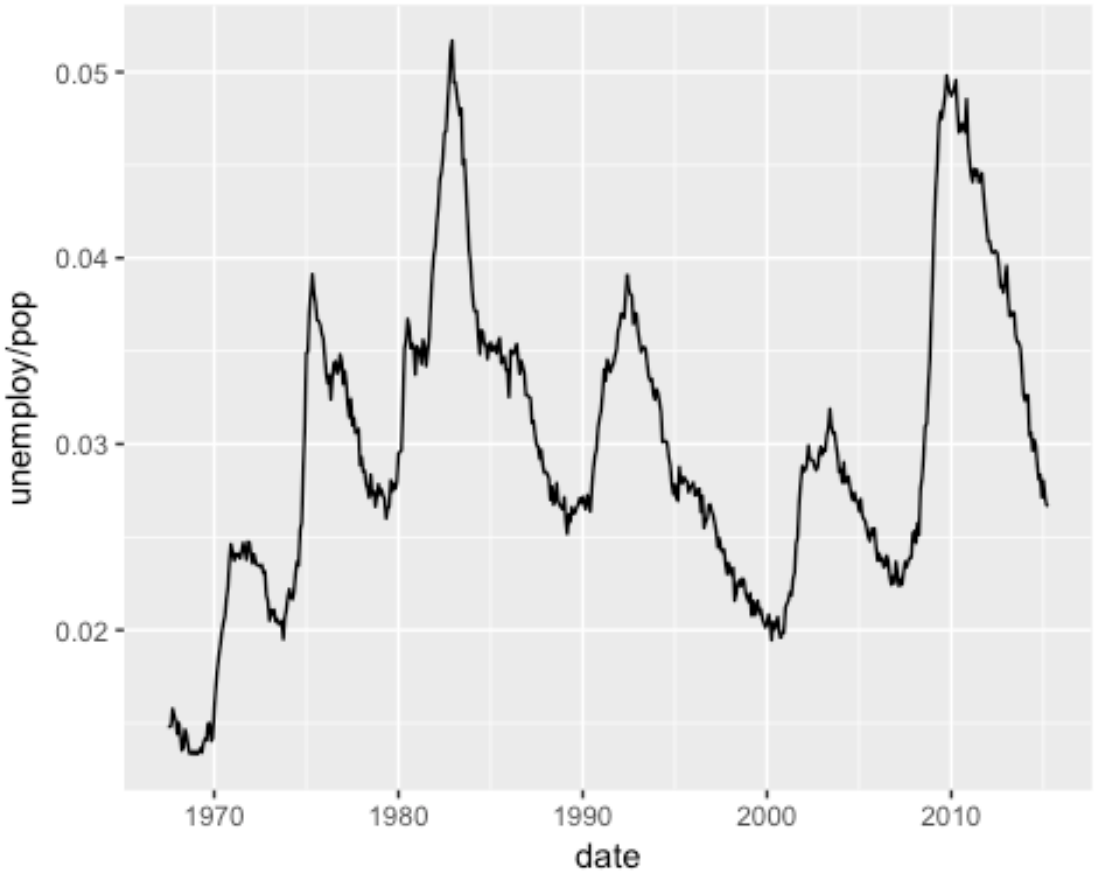
```r
names(economics)
```

```
## [1] "date"     "pce"      "pop"      "psavert"  "uempmed"
"unemploy"
```

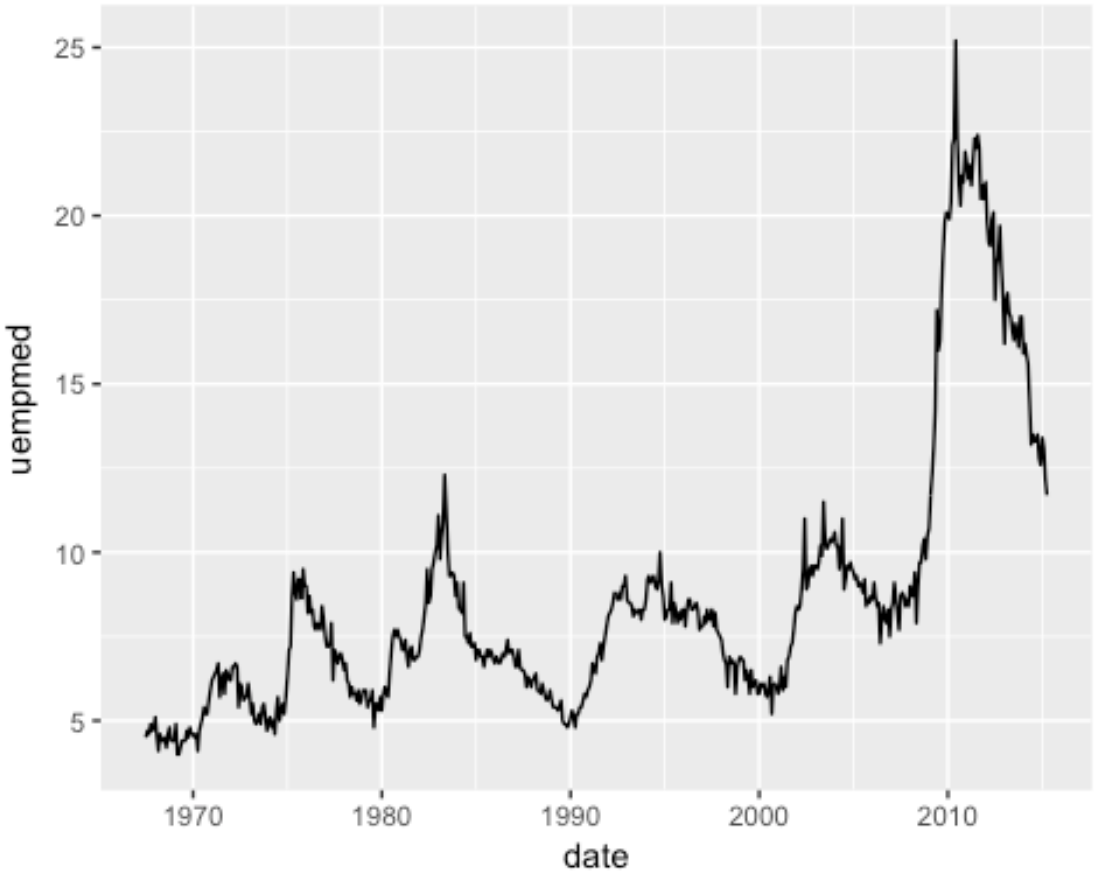
```r
qplot(date,pop, data=economics)+geom_line()
```




```r
qplot(date, unemploy / pop, data = economics, geom = "line")
```




```
qplot(date, uempmed, data = economics, geom = "line")
```



Multivariate Exploritory Data Analysis

Multivariate analysis (MVA) is based on the statistical principle of multivariate statistics, which involves observation and analysis of more than one statistical outcome variable at a time.

- from Multivariate analysis Wikipedia

Bivariate analysis is one of the simplest forms of quantitative (statistical) analysis. It involves the analysis of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them.

- from Bivariate analysis Wikipedia

Scatter plot (Bivariate Visualization)

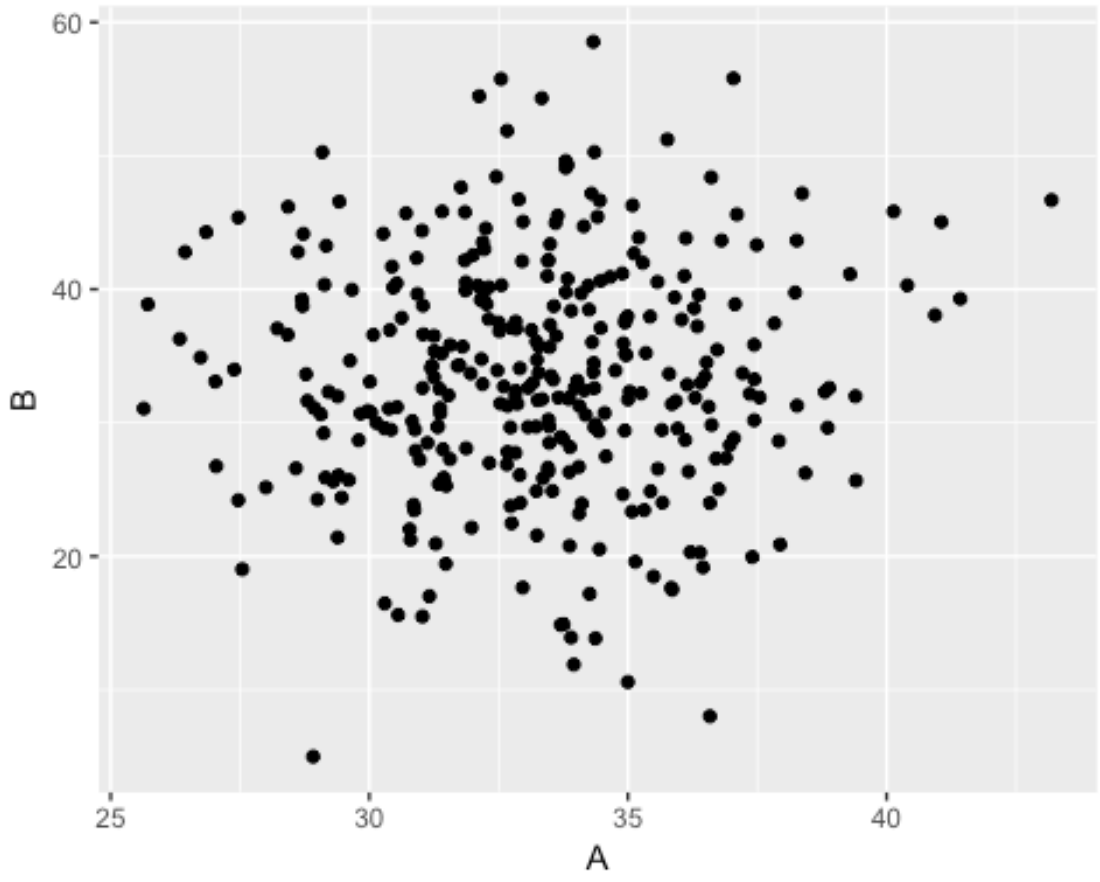
A scatter plot, scatterplot, or scattergraph is a type of mathematical diagram using Cartesian coordinates to display values for two variables for a set of data.

```
# Set a seed value for randomization
set.seed(333)
# create some random data
trails<-333
```

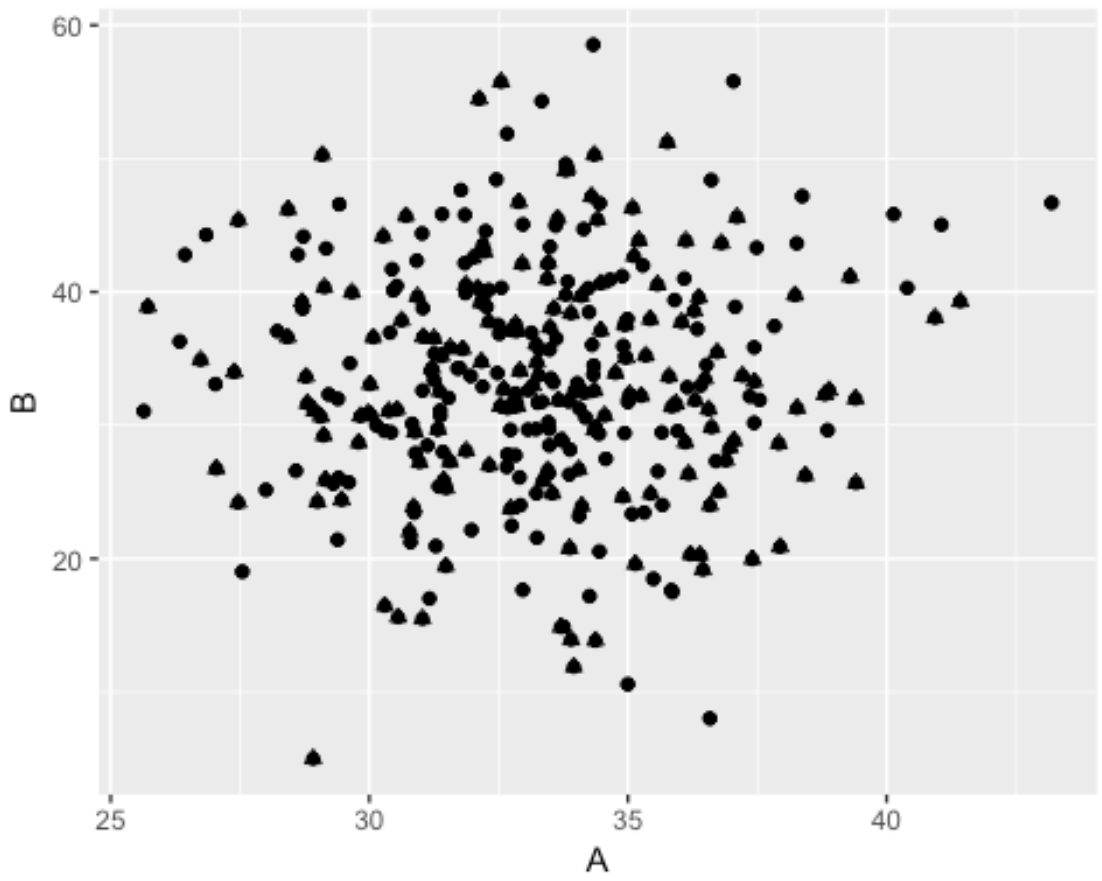
```
r_data <-data.frame(A=rnorm(n=trails,mean=33,sd=3),
                    B=rnorm(n=trails,mean=33,sd=9),
                    C=1:trails+rnorm(n=trails,sd=3),
                    D=1:trails+rnorm(n=trails,sd=3),
                    E=1:trails+rnorm(n=trails,sd=33),
                    F=1:trails+rnorm(n=trails,sd=33),

age=factor(sample(c(1,2,3,4),size=trails,replace=T),

levels=c(1,2,3,4),labels=c("Toddler","Child","Teen","Adult")),
                    gender=factor(sample(c(1,2),size=trails,replace=T),

levels=c(1,2),labels=c("Male","Female")),

wealth=factor(sample(c(1,2,3),size=trails,replace=T),
levels=c(1,2,3),labels=c("Poor","Middle","Rich")))
```

Scatter Plot with No apparent relationship

```
qplot(A,B, data=r_data) # random
```

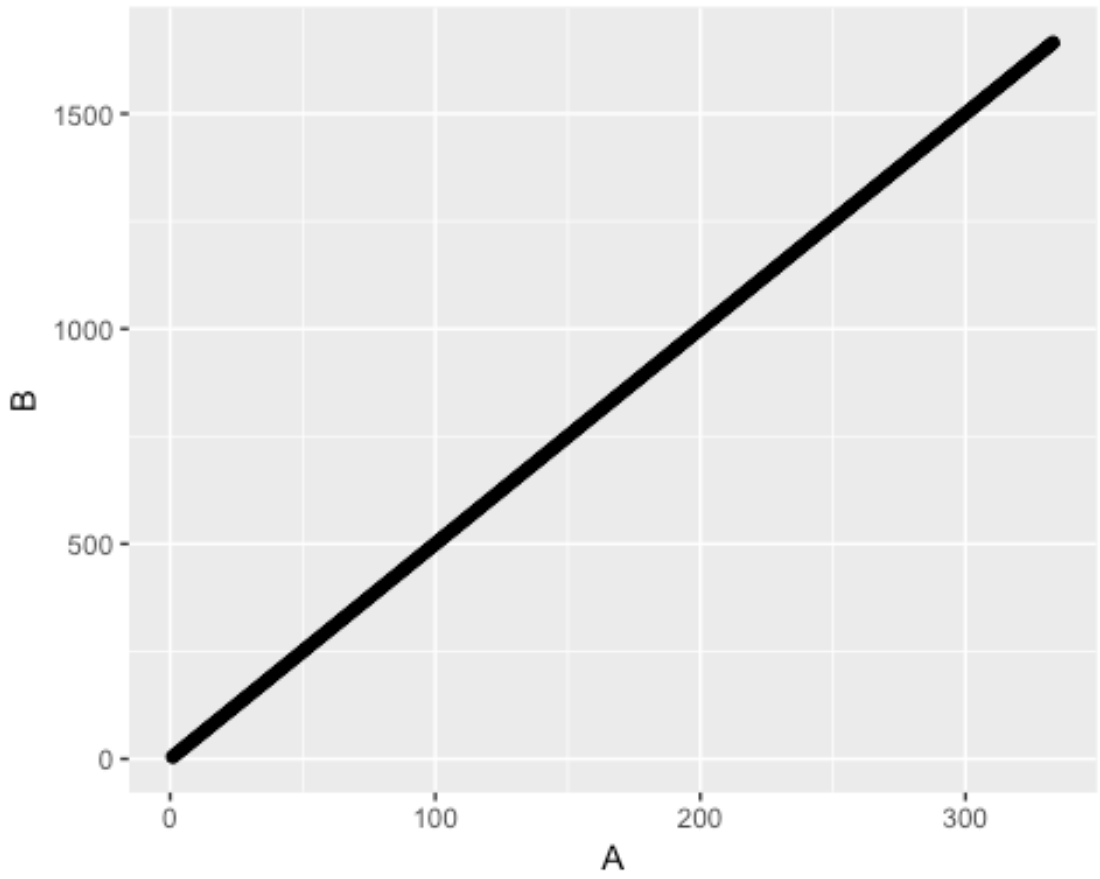



```
qplot(A,B, data=r_data) + geom_point(shape=r_data$gender)
```



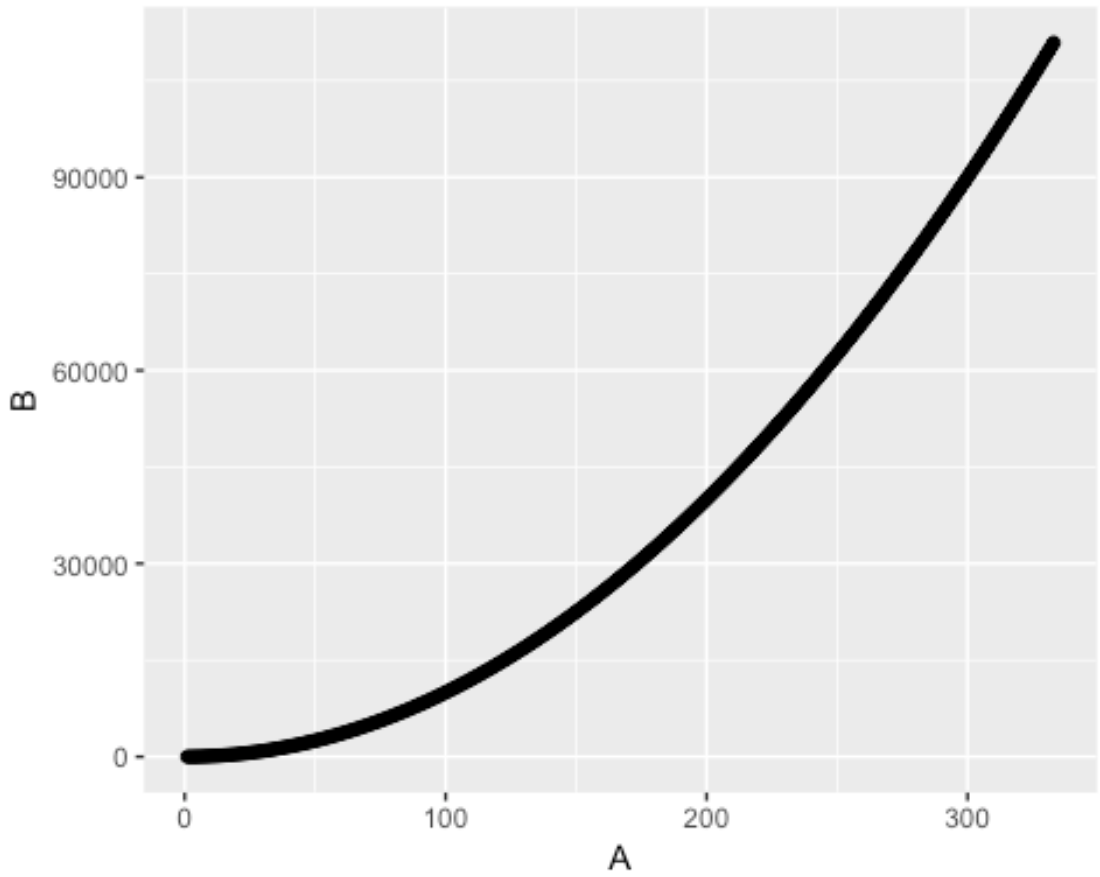
Scatter Plot with Linear relationship

```
A<-1:trails
B<-A*5
r_linear<-data.frame(A,B)
qplot(A,B, data=r_linear) # linear
```

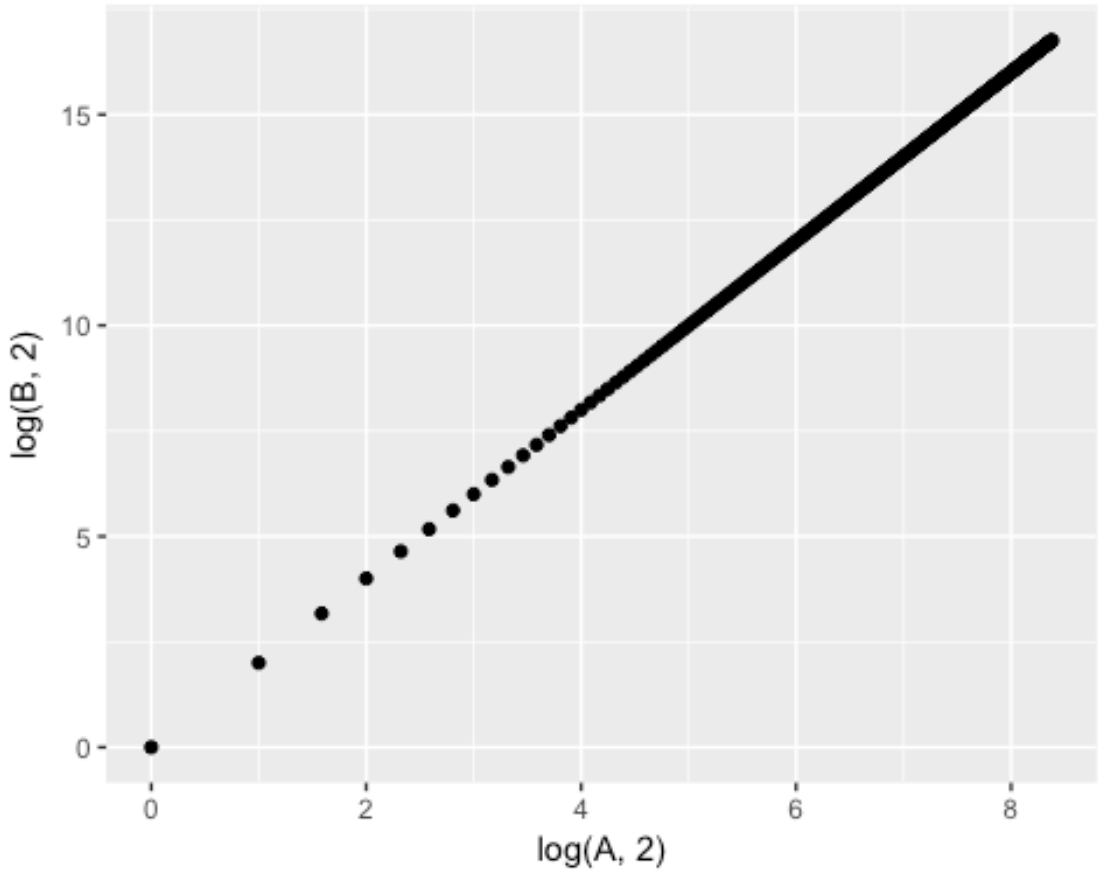


Scatter Plot with Quadratic relationship

```
B<-A^2
r_llog<-data.frame(A,B)
qplot(A,B, data=r_llog) # quadratic
```




```
qplot(log(A,2),log(B,2), data=r_llog) # log-log
```

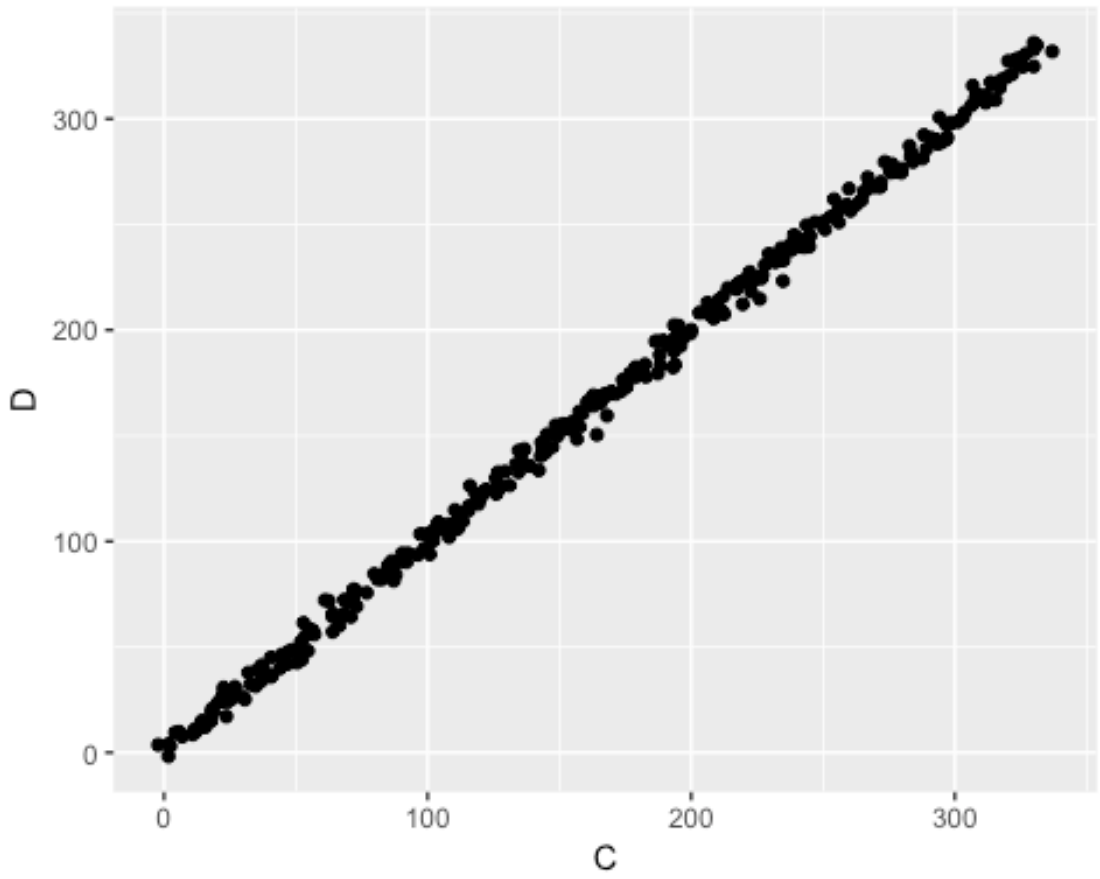


Scatter plot with Homoscedastic relationship

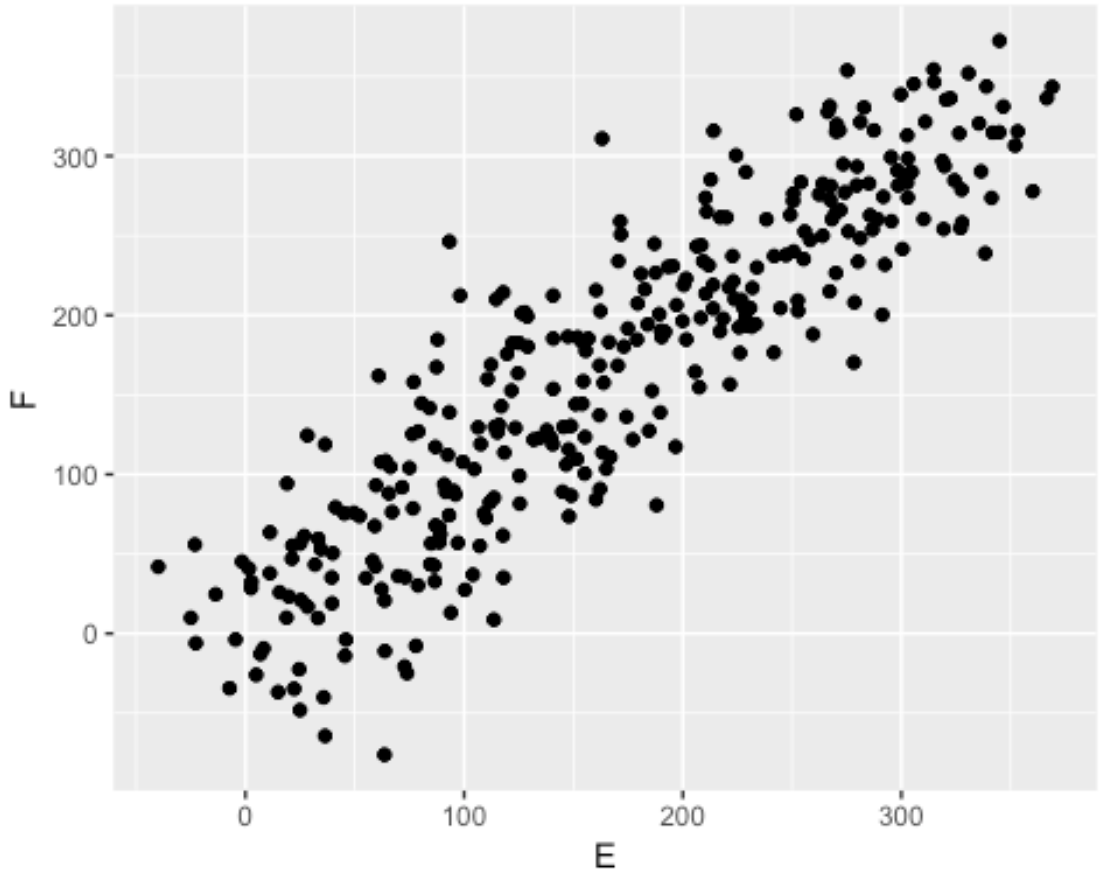
In statistics, a sequence or a vector of random variables is homoscedastic /??ho??mo??sk??'d?st??k/ if all random variables in the sequence or vector have the same finite variance. This is also known as homogeneity of variance. The complementary notion is called heteroscedasticity.

- from Homoscedasticity

```
qplot(C,D, data=r_data) #  homoscedastic
```



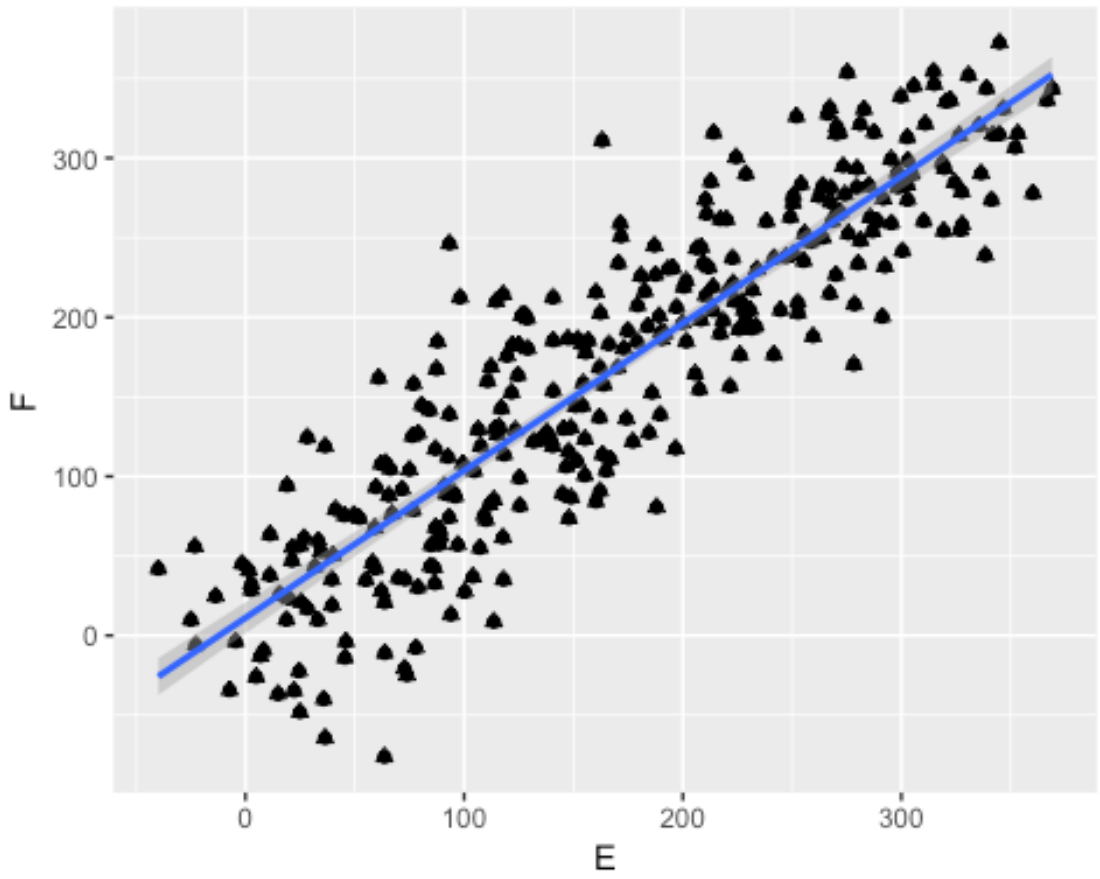

```
qplot(E,F, data=r_data) + geom_point(shape=1)  #  homoscedastic
```



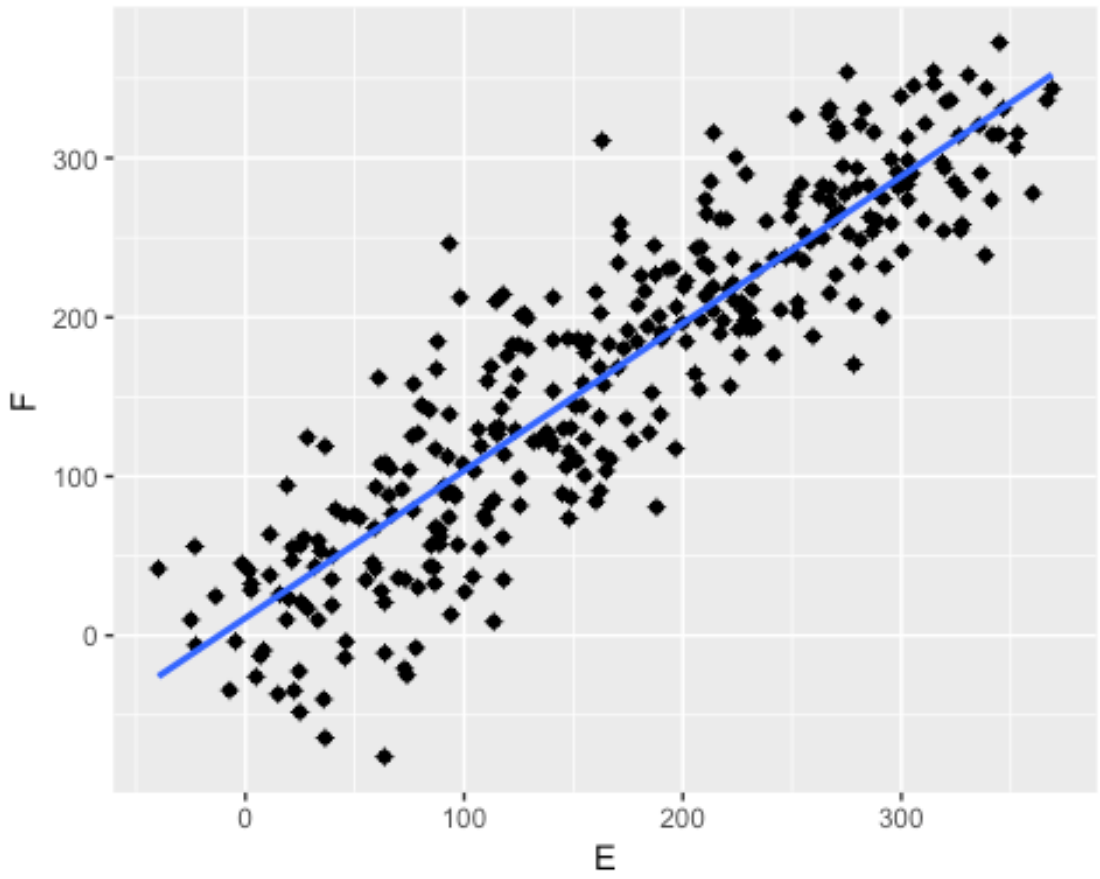
Scatter Plot with Regression Lines

Homoscedasticity means that the variance around the regression line is the same for all values of the predictor variable as in the graph below.

```
qplot(E,F, data=r_data) + geom_point(shape=2) + geom_smooth(method=lm)
```

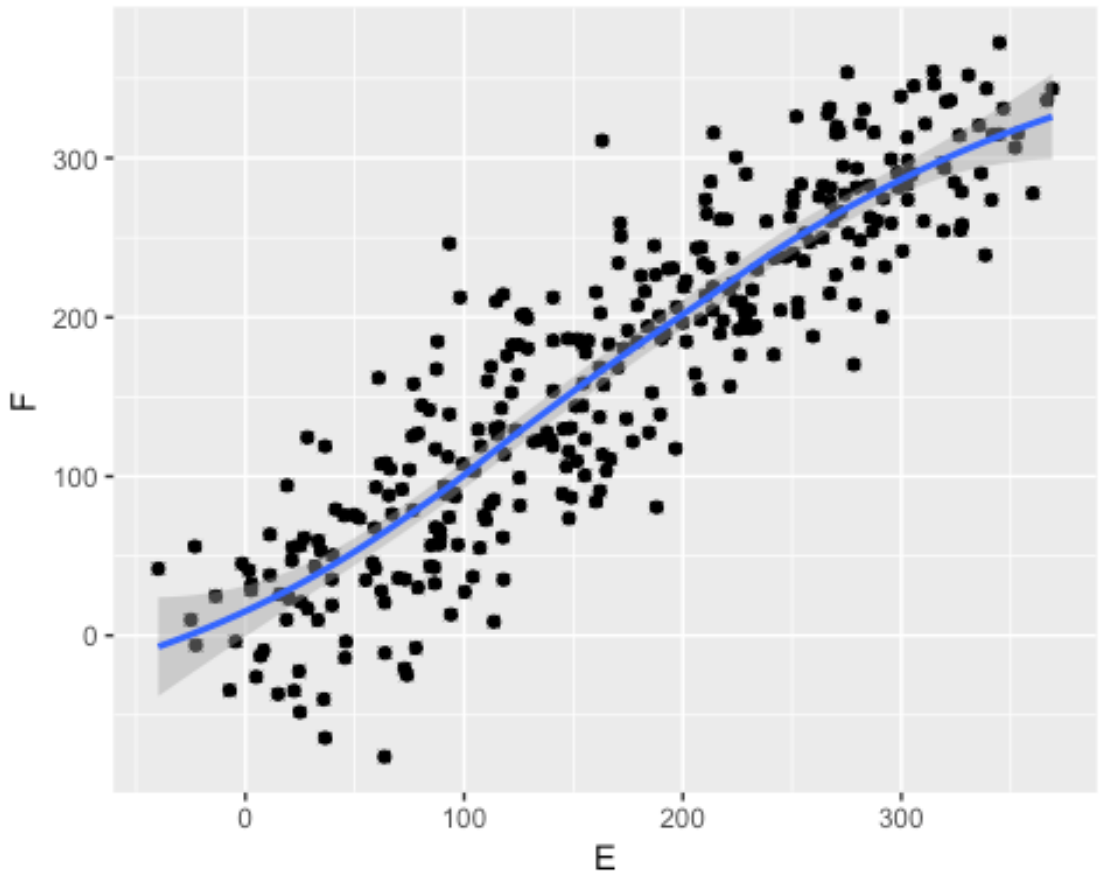



```
qplot(E,F, data=r_data) + geom_point(shape=3) + geom_smooth(method=lm,
se=FALSE)
```

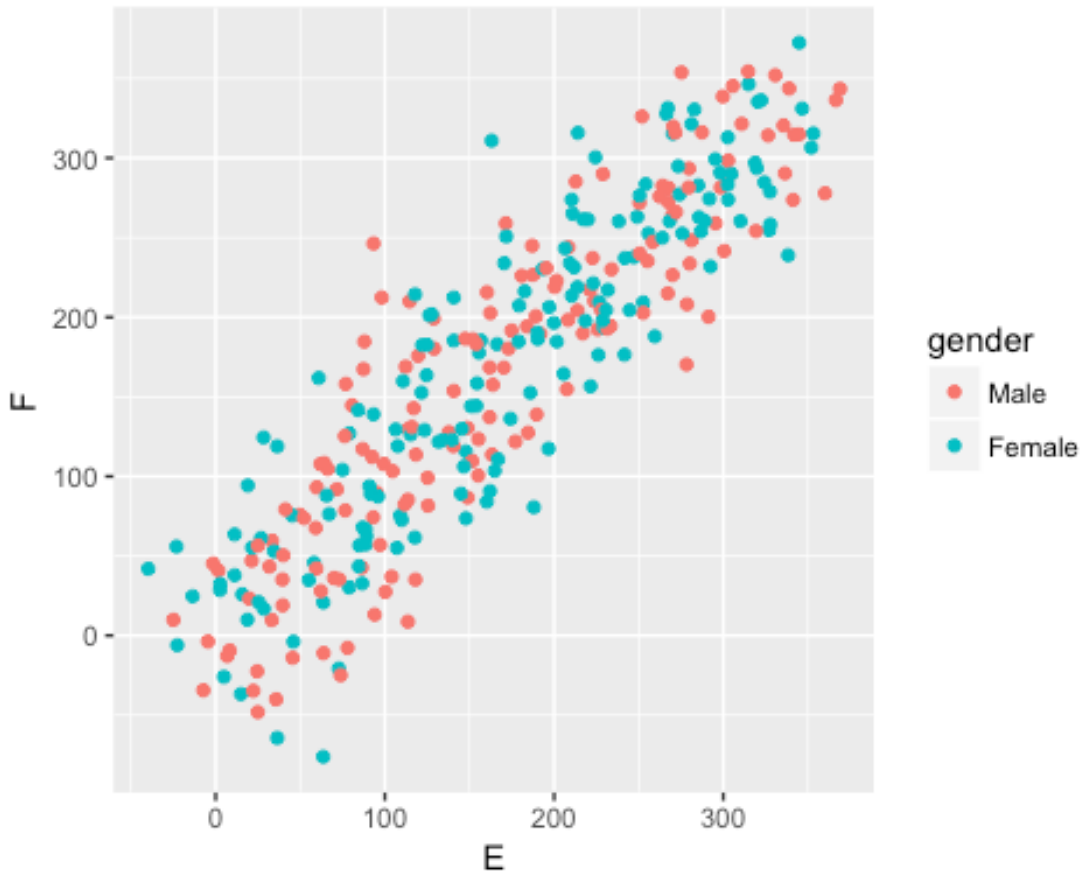



```
qplot(E,F, data=r_data) + geom_point(shape=4) + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```




```
qplot(E,F, data=r_data,colour=gender)
```



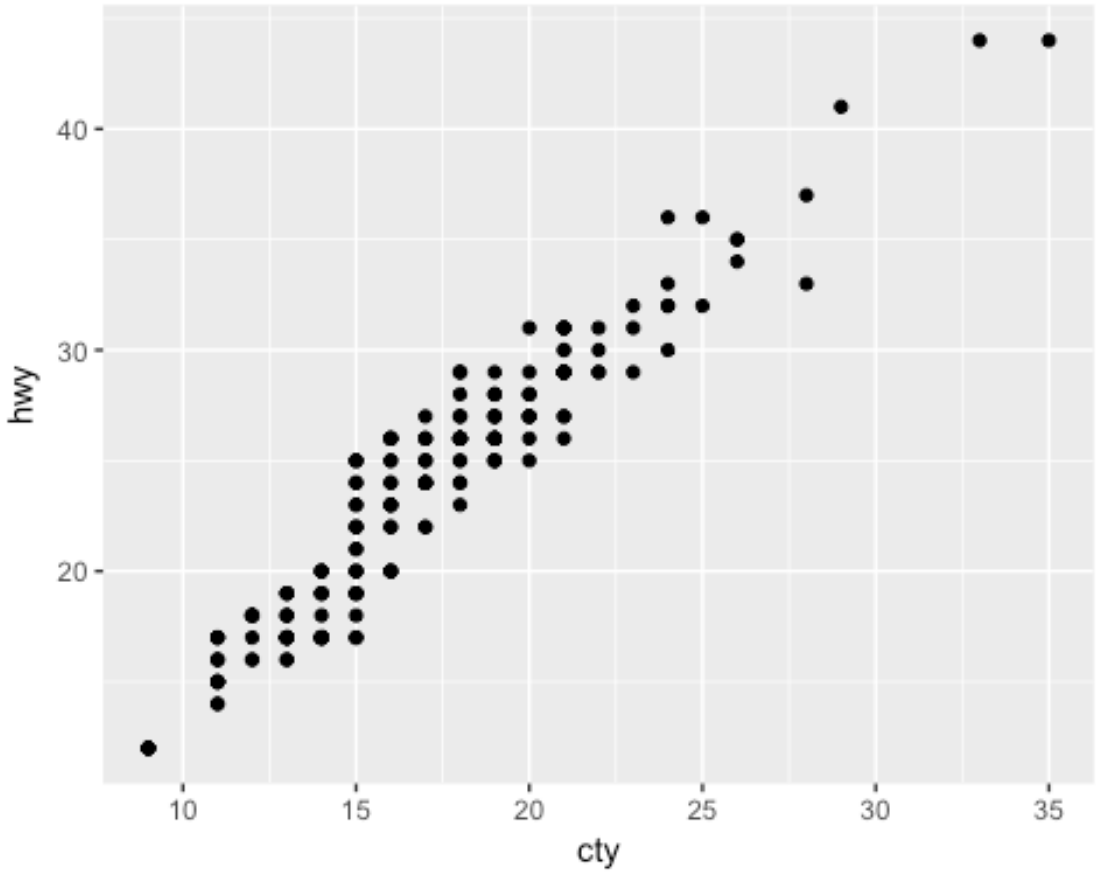
Scatter plot with Jittering

Integer data points can stack on top of one another hiding the distribution.

```
str(mpg)

## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11
variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999
2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)"
"auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...

names(mpg)
```

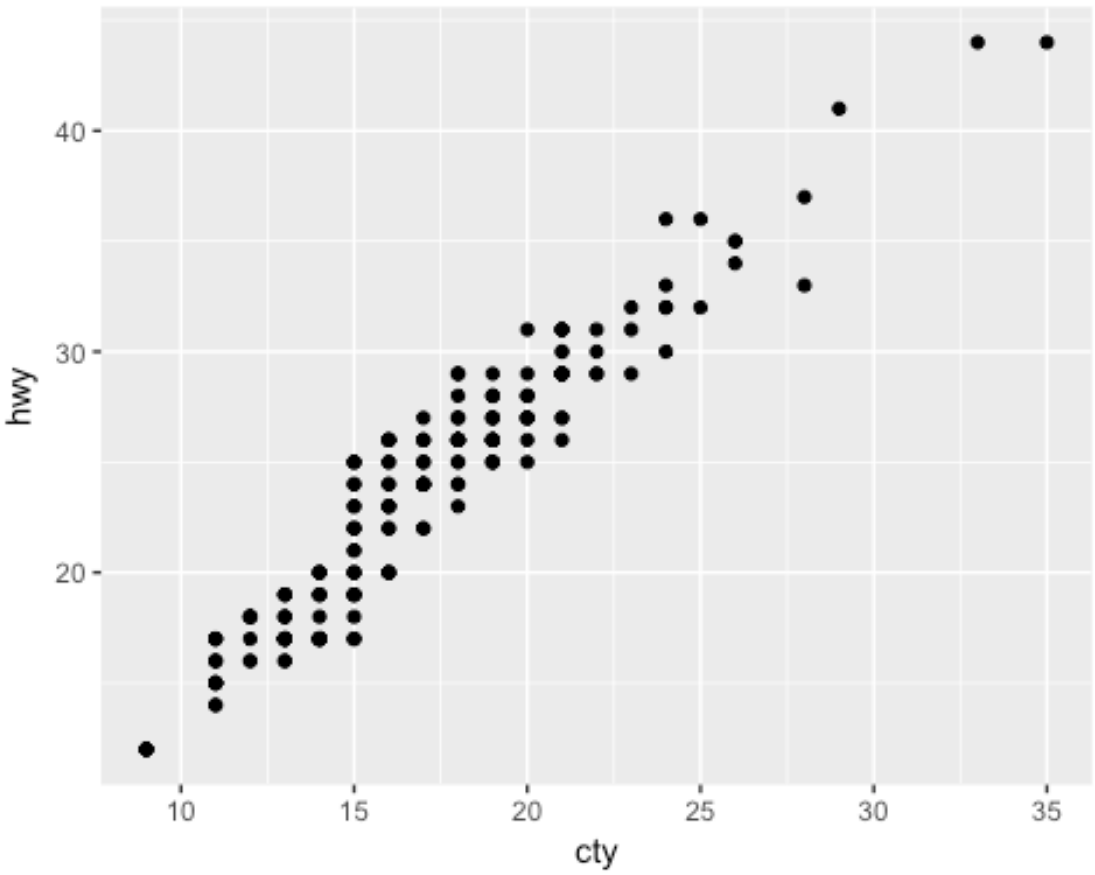
```
##  [1] "manufacturer" "model"        "displ"        "year"
##  [5] "cyl"          "trans"        "drv"          "cty"
##  [9] "hwy"          "fl"           "class"
```

```
qplot(cty,hwy,data=mpg)
```



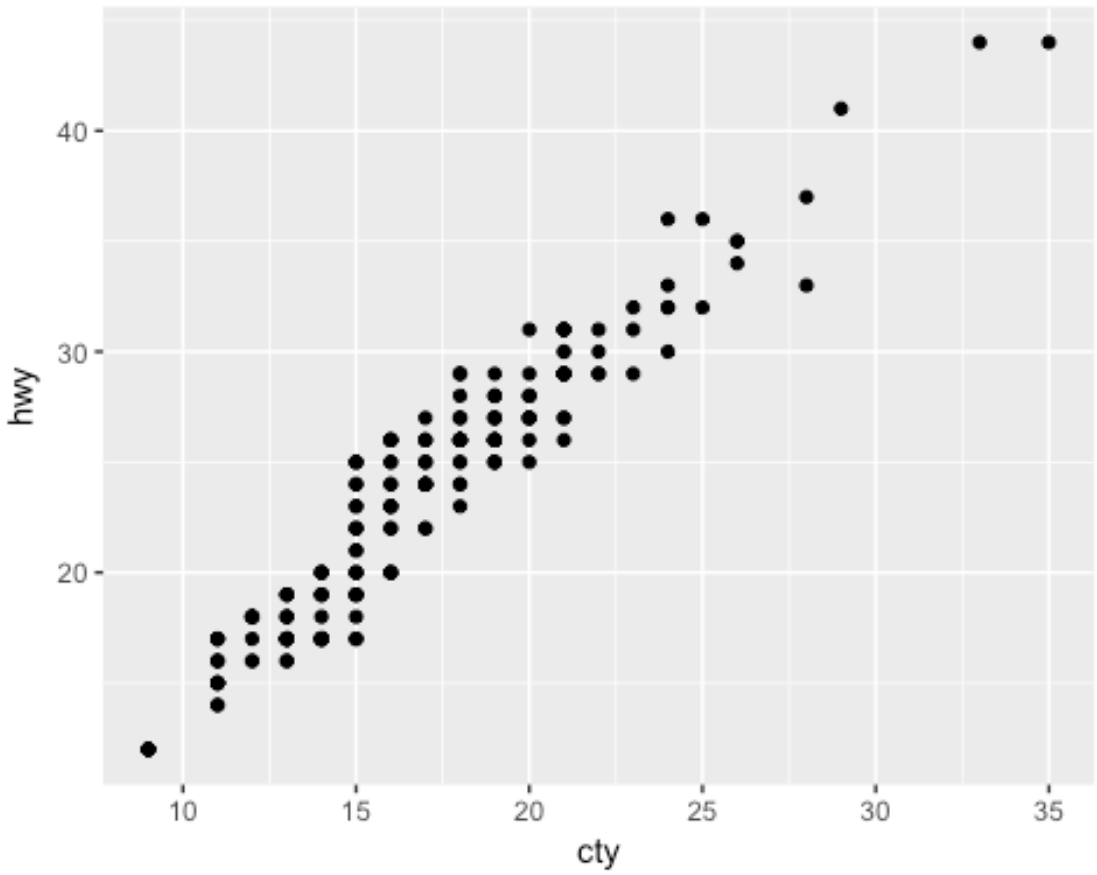

```
qplot(cty,hwy,data=mpg,position = "jitter")
```

```
## Warning: `position` is deprecated
```




```
qplot(cty,hwy,data=mpg,position = "jitter") # jitter changes
```

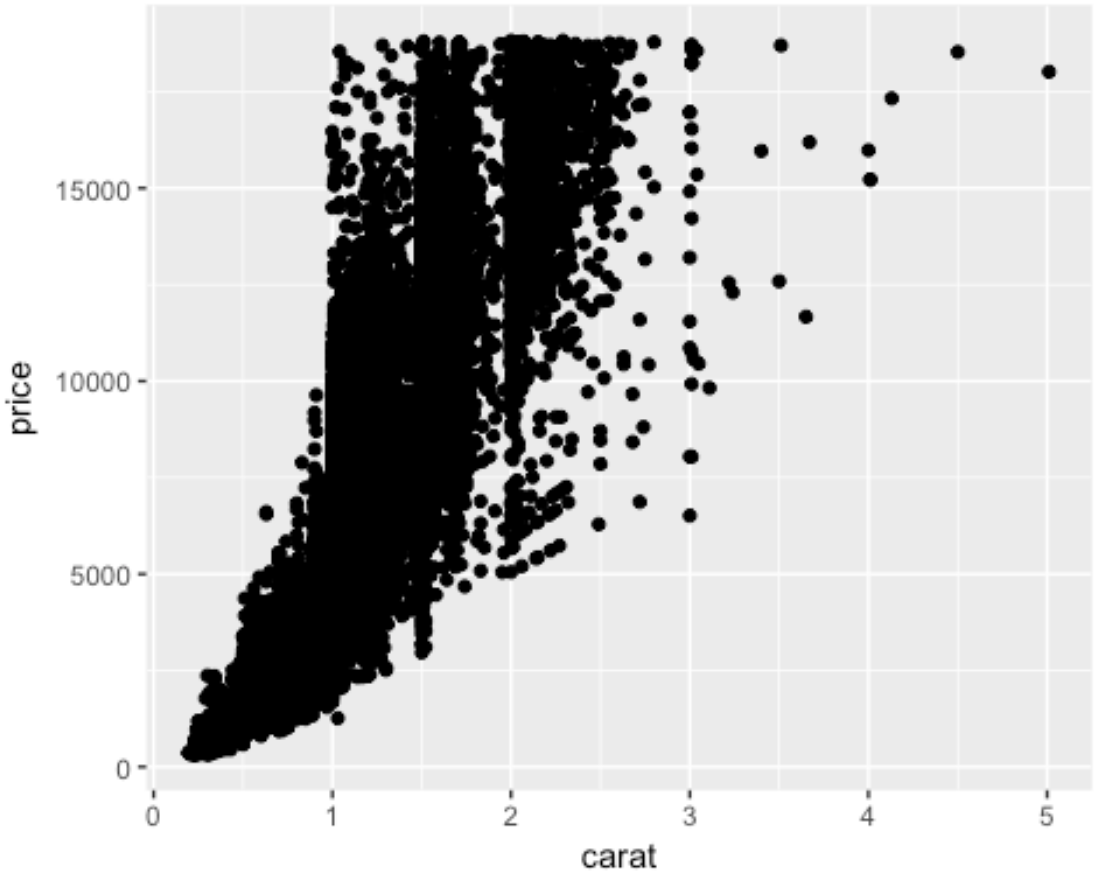
```
## Warning: `position` is deprecated
```



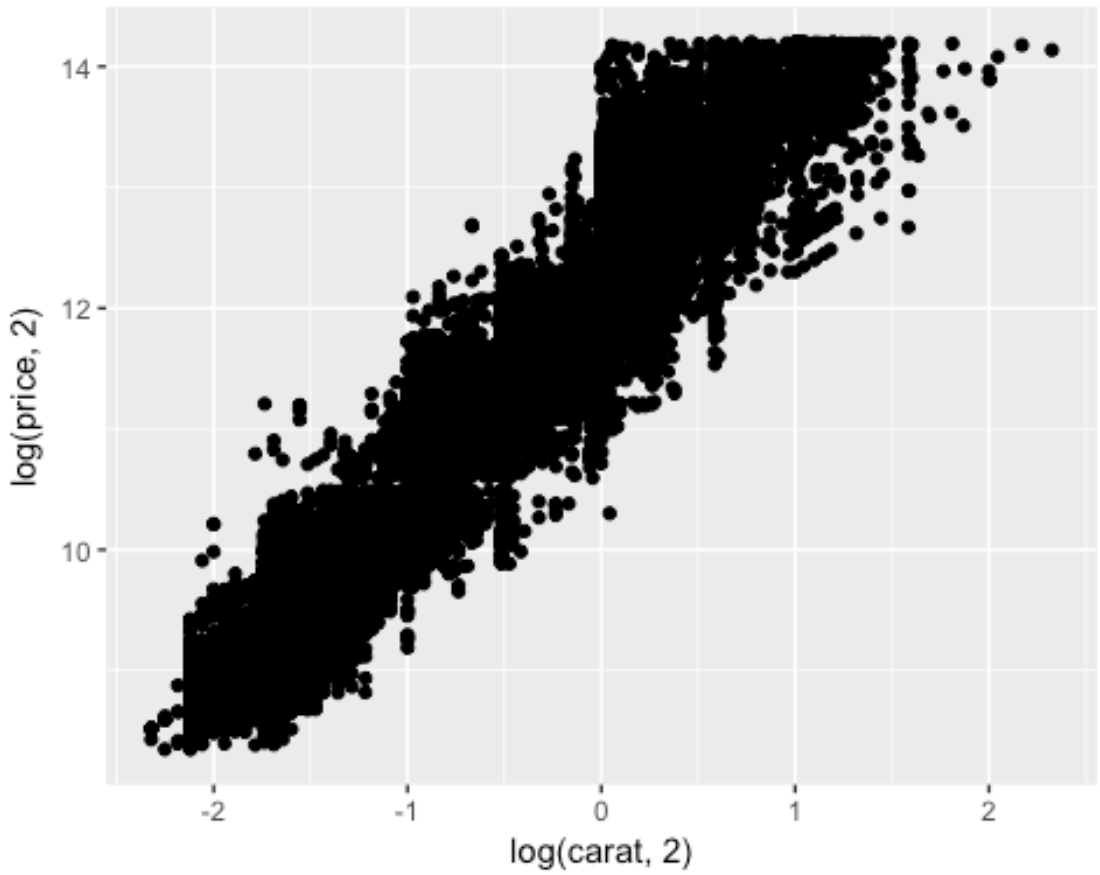
Playing with scatter plots

The scatter diagram or scatter plot is the workhorse bivariate and multivariate plots. Let's play with scatter plots a bit.

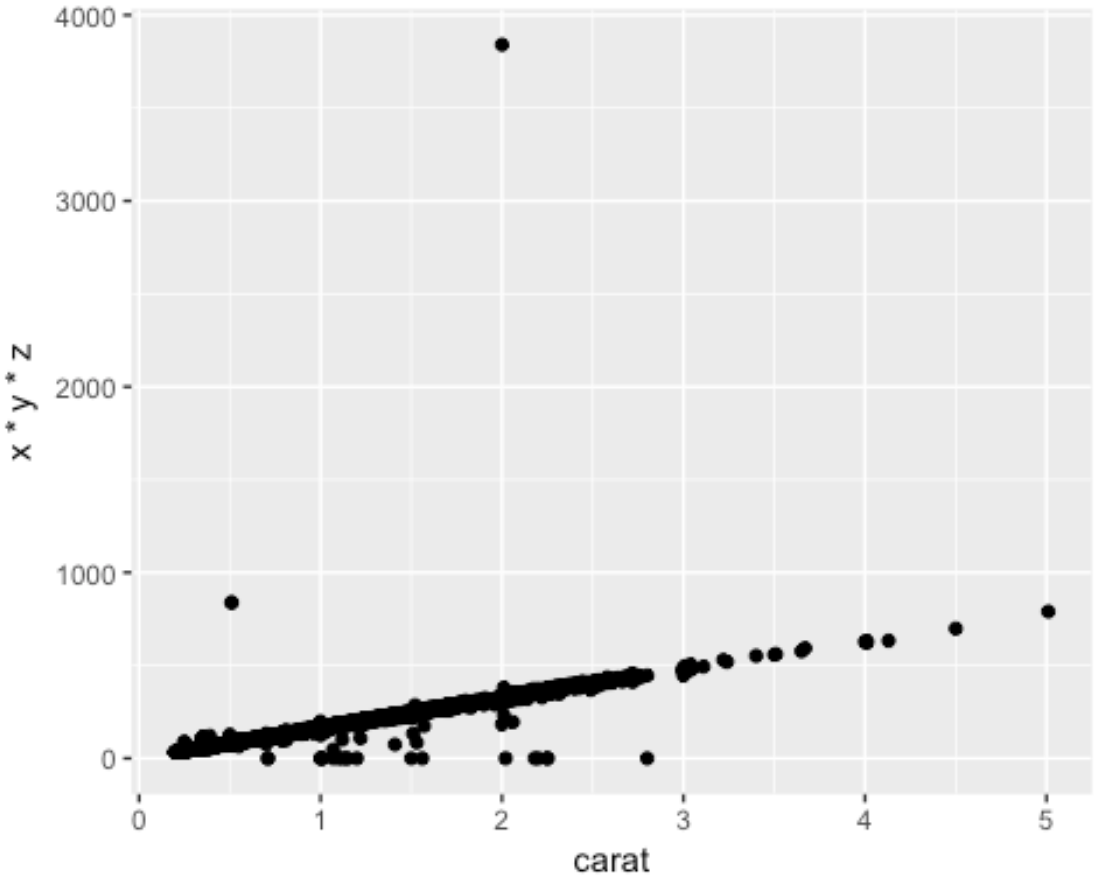
```
qplot(carat, price, data = diamonds)
```



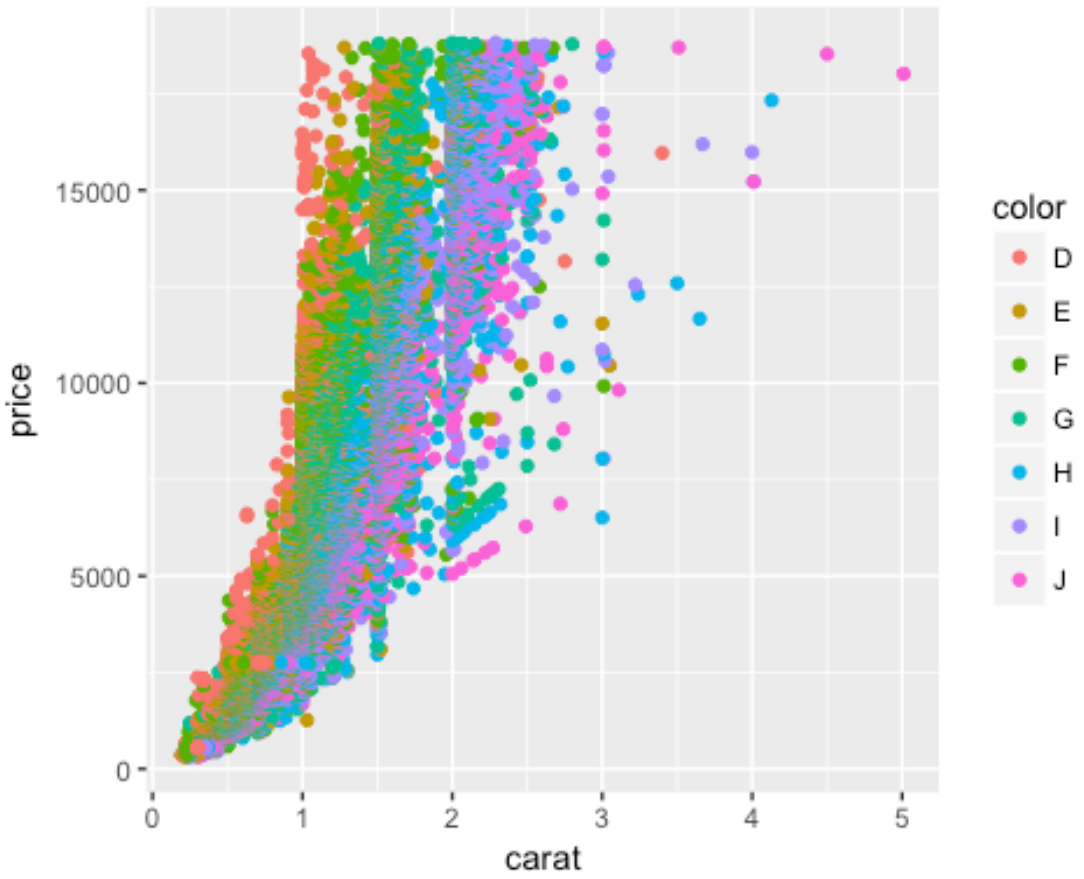

```
qplot(log(carat,2), log(price,2), data = diamonds)
```



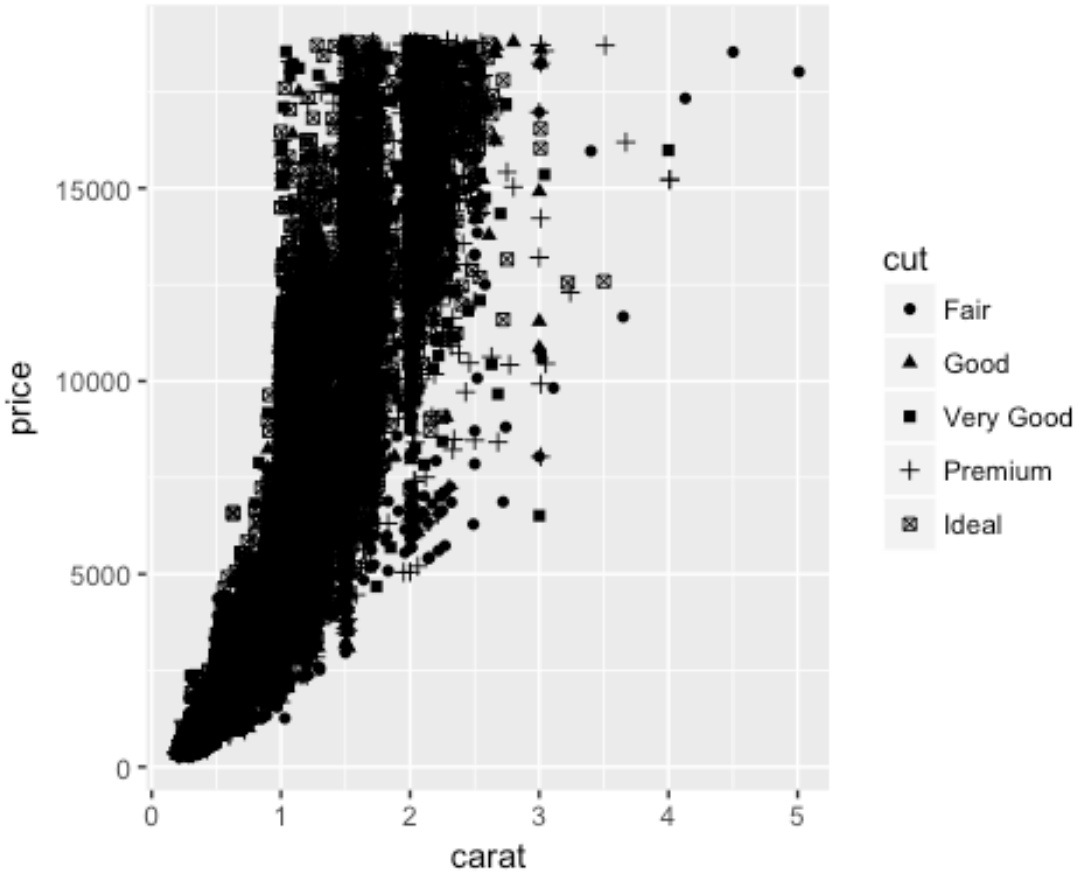

```
qplot(carat, x * y * z, data = diamonds)
```




```
qplot(carat, price, data = diamonds, colour = color)
```

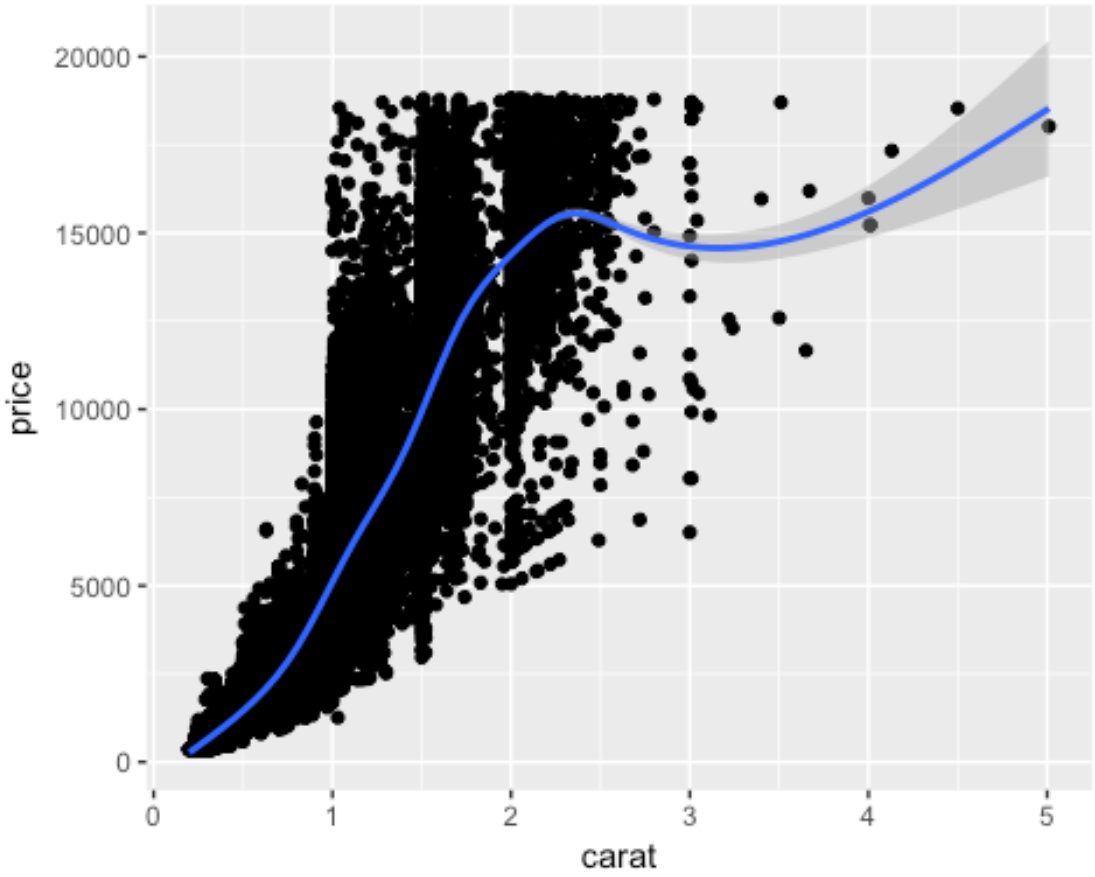



```
qplot(carat, price, data = diamonds, shape = cut)
```

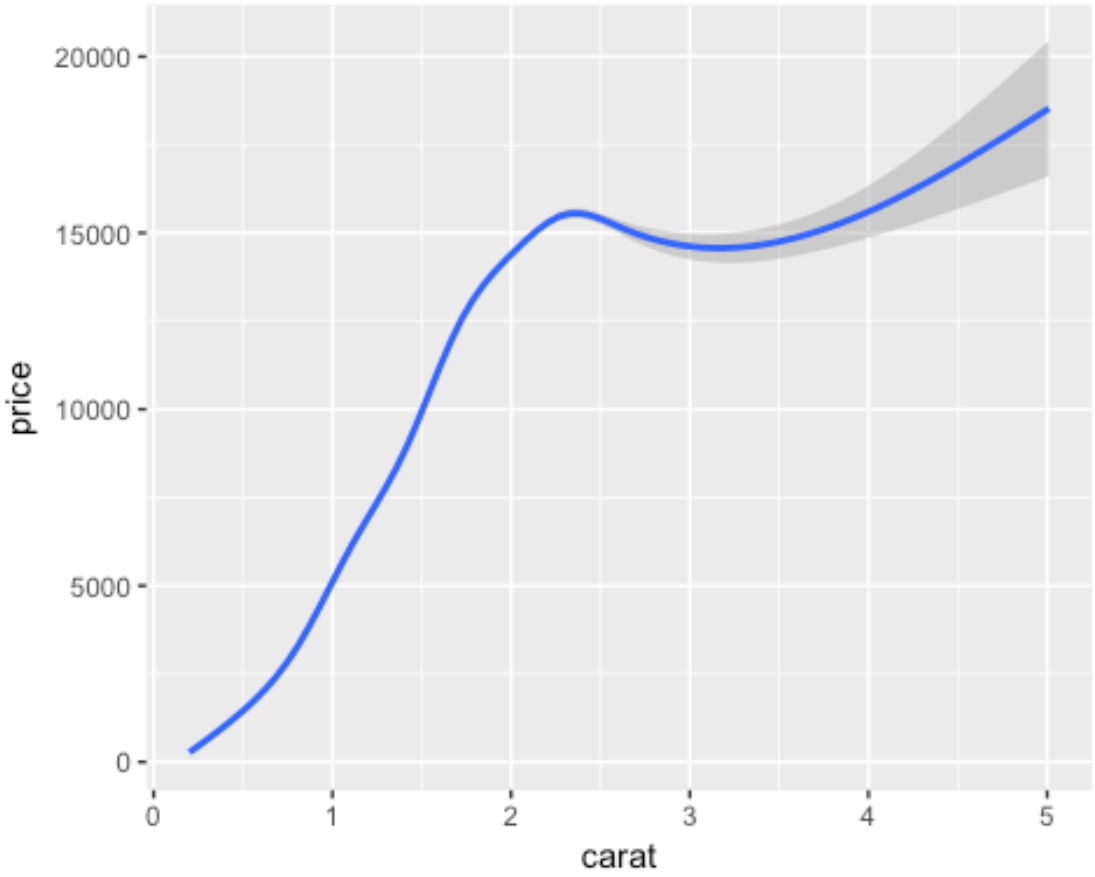



```
qplot(carat, price, data = diamonds, geom = c("point", "smooth"))
```

```
## `geom_smooth()` using method = 'gam'
```

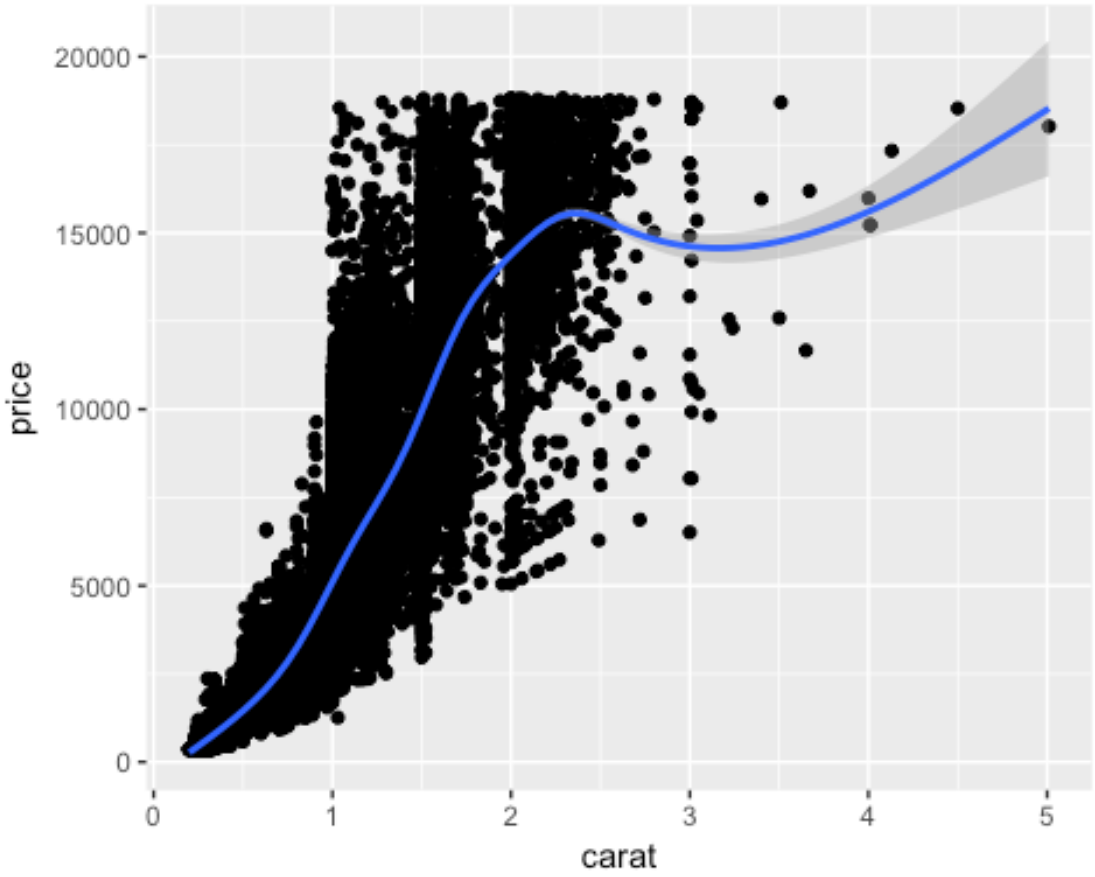



```
qplot(carat, price, data = diamonds, geom = "smooth")

## `geom_smooth()` using method = 'gam'
```

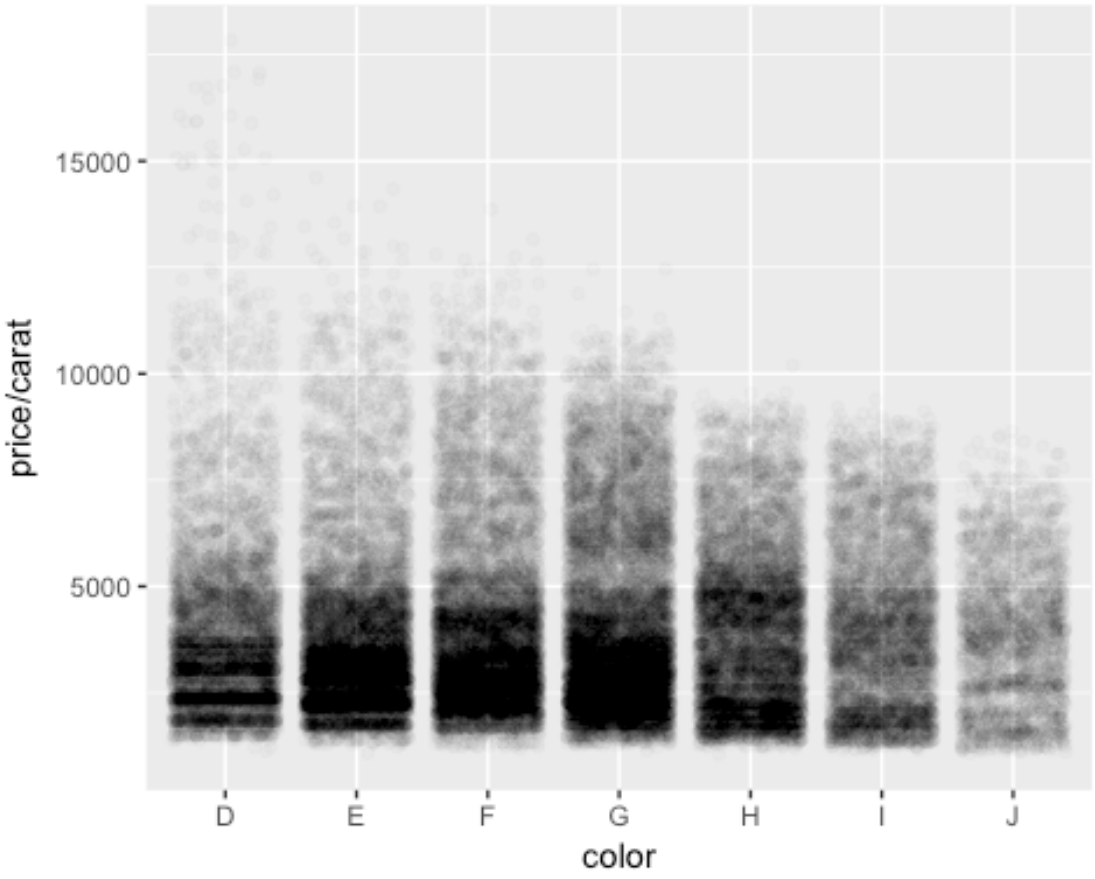



```
qplot(carat, price, data = diamonds)+geom_smooth()
```

```
## `geom_smooth()` using method = 'gam'
```



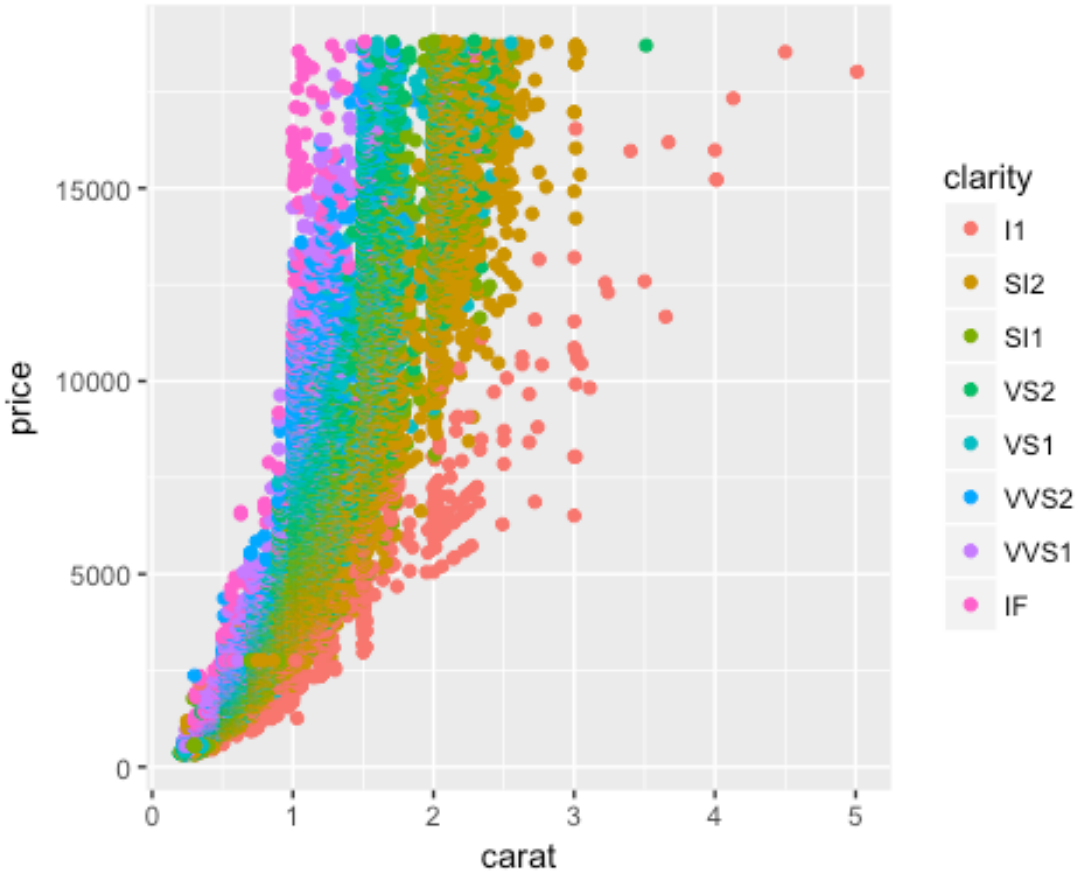

```
qplot(color, price / carat, data = diamonds, geom = "jitter",
  alpha = I(1 / 50))
```



Faceting and scaling

One option we could use is to color-code the points by their clarity. Here, we pass another `col=` argument with the variable we'd like to use for color-coding:

```
qplot(carat, price, data = diamonds, col = clarity)
```

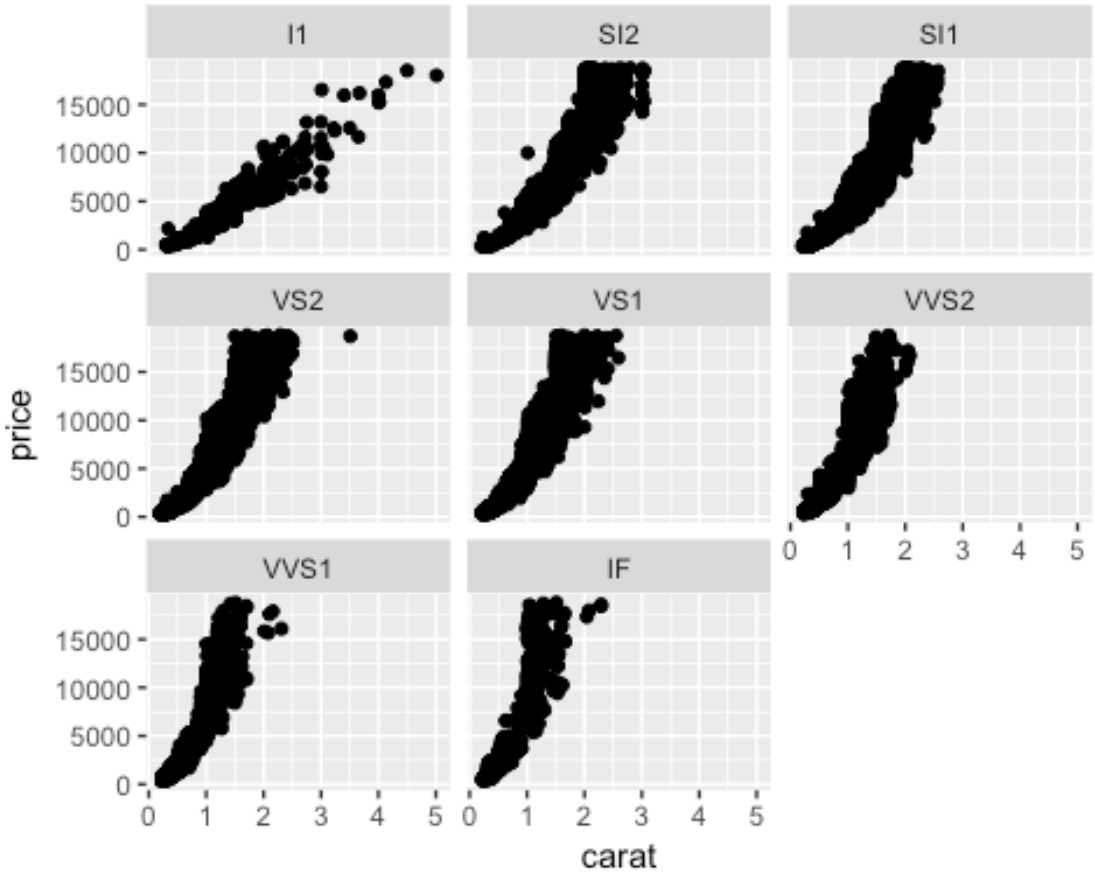


Faceting

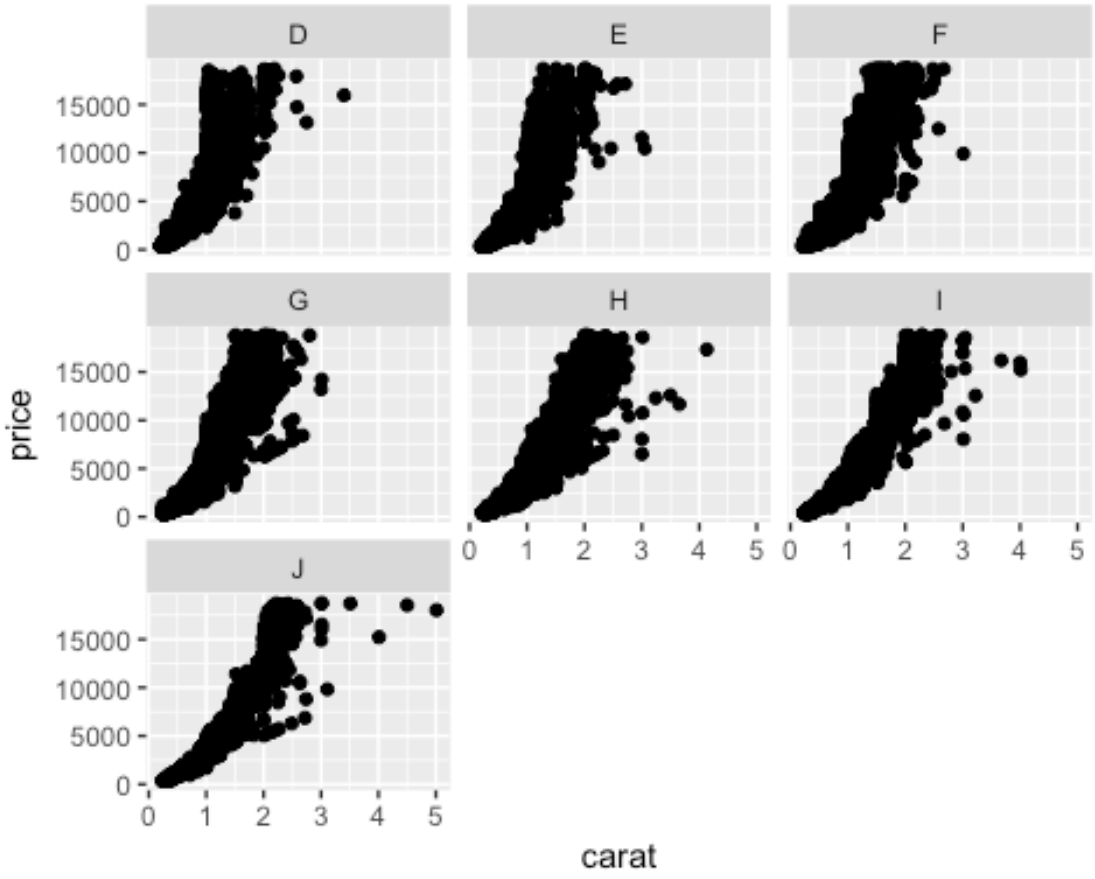
Facets display subsets of the dataset in different panels.

- facet_grid - Lay out panels in a grid.
- facet_null - Facet specification: a single panel.
- facet_wrap - Wrap a 1d ribbon of panels into 2d.
- label_both -Label facets with value and variable.
- label_bquote - Label facet with 'bquoted' expressions
- label_parsed - Label facets with parsed label.
- label_value - Label facets with their value

```
qplot(carat, price, data = diamonds, facets = ~ clarity)
```




```
qplot(carat, price, data = diamonds, facets = ~ color)
```



QQplot

In statistics, a Q-Q plot ("Q" stands for quantile) is a probability plot, which is a graphical method for comparing two probability distributions by plotting their quantiles against each other.

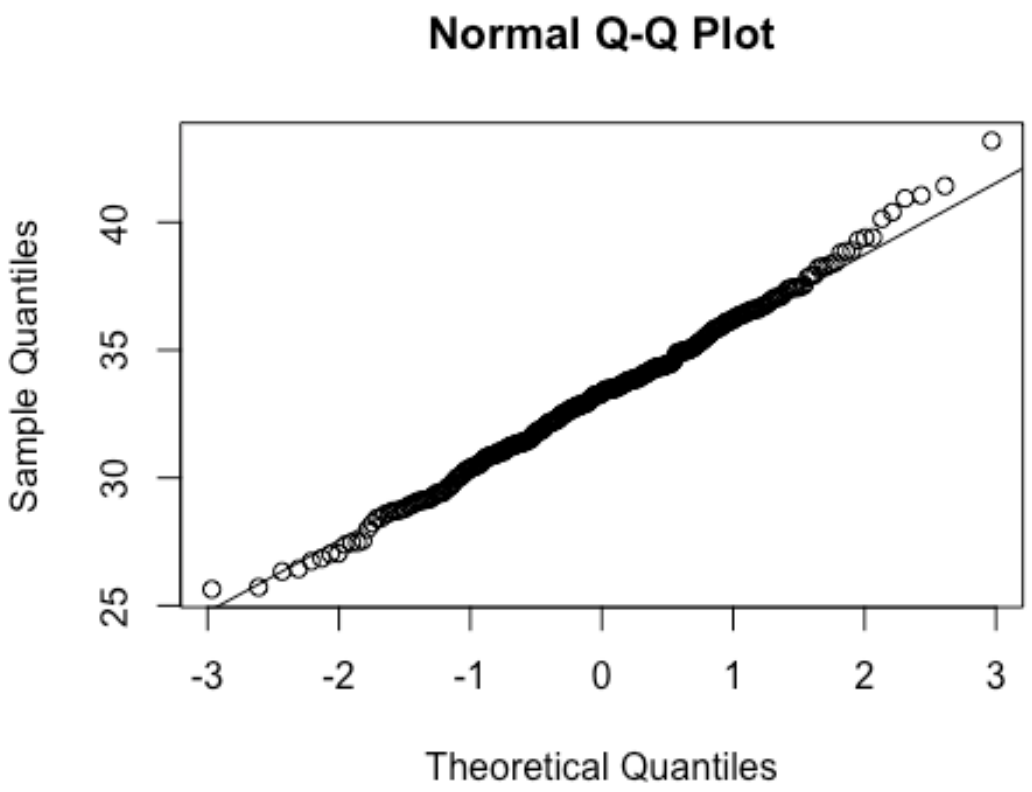
from Q-Q plot - Wikipedia

One of the first things we may ask about data is whether it deviates from an expectation to be normally distributed. The quantile-quantile (QQ) plot is a way to see whether the data deviate from normal (the plot can be set up to see if the data deviate from other distributions as well).

The process R goes through for creating a QQ plot involves determining what proportion of the 'observed' scores fall below any one score, then the Z-score that would fit that proportion if the data were normally distributed is calculated, and finally that Z-score that would cut off that proportion (the 'expected normal value') is translated back into the original metric to see what raw score that would be.

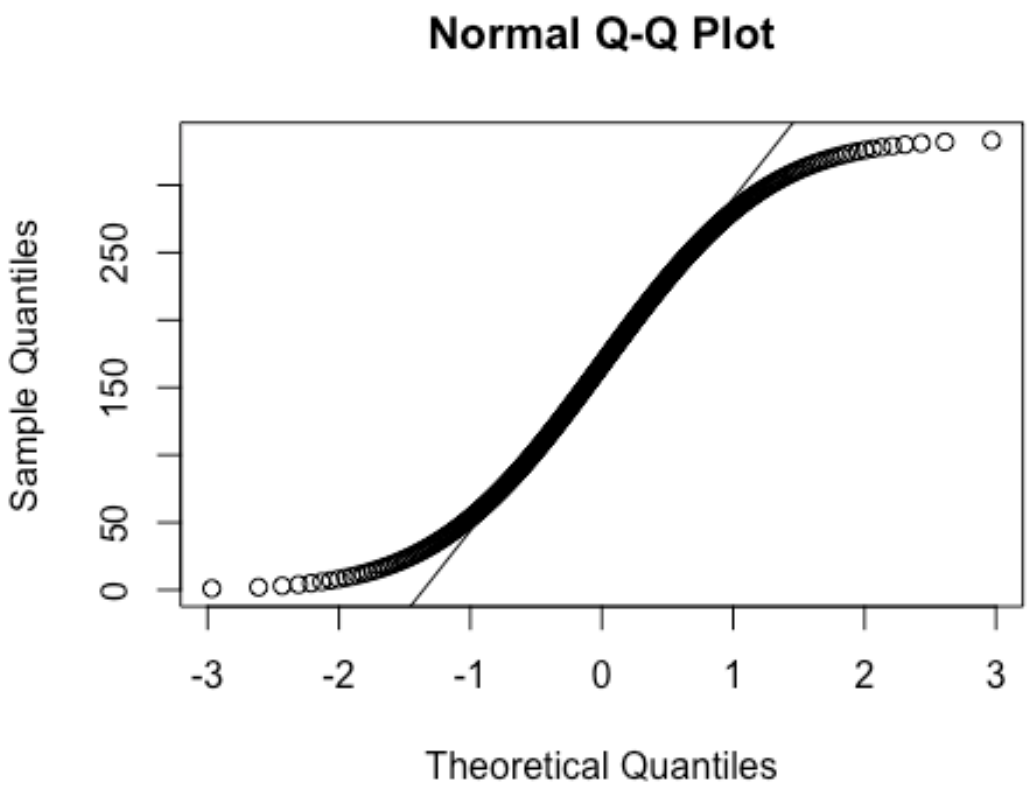
A scatter plot is then created that shows the relationship between the actual 'observed' values and what those values would be 'expected' to be if the data were normally distributed.

```
qqnorm(r_data$A)
qqline(r_data$A)
```



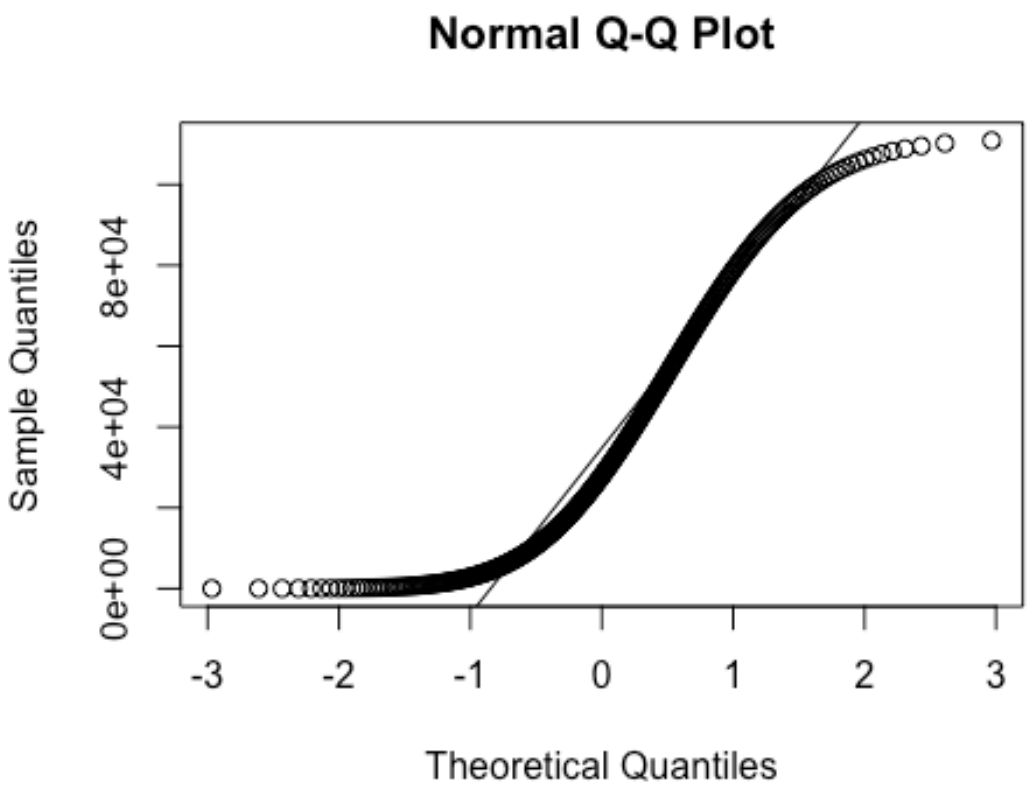
help(qqnorm)

```
qqnorm(r_linear$A)
qqline(r_linear$A)
```



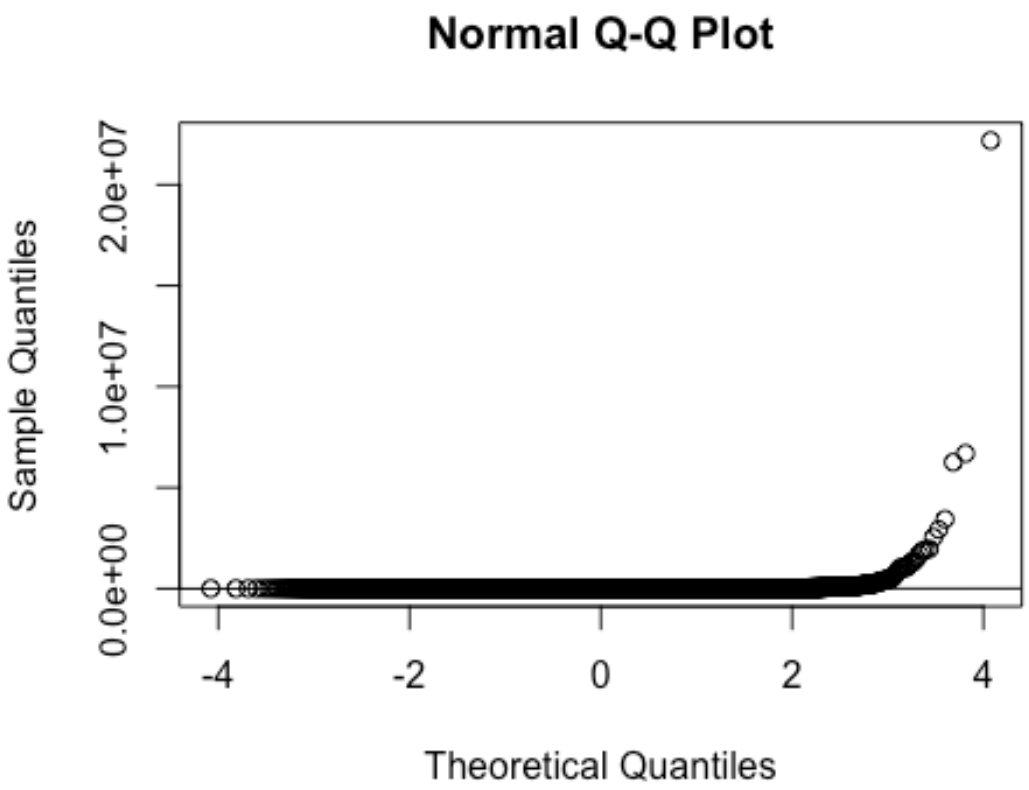
help(qqnorm)

```
qqnorm(r_llog$B)
qqline(r_llog$B)
```



help(qqnorm)

```
qqnorm(twitter$followers_count)
qqline(twitter$followers_count)
```



In statistics, a sequence or a vector of random variables is homoscedastic if all random variables in the sequence or vector

have the same finite variance.