Lab 1

Mu Lu

2015-09-21

The Data

Load `.Rdata` file, a dataframe `cdc` will be loaded:

```
load(url("http://assets.datacamp.com/course/dasi/cdc.Rdata"))
```

Take a look at the names of variables:

```
names(cdc)
```

```
## [1] "genhlth"  "exerany"  "hlthplan" "smoke100" "height"   "weight"  
## [7] "wtdesire" "age"      "gender"
```

Have a look at the first or last few entries(rows):

```
head(cdc)
```

```
##     genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1      good       0        1        0     70    175      175  77      m
## 2      good       0        1        1     64    125      115  33      f
## 3      good       1        1        1     60    105      105  49      f
## 4      good       1        1        0     66    132      124  42      f
## 5 very good       0        1        0     61    150      130  55      f
## 6 very good       1        1        0     64    114      114  55      f
```

```
tail(cdc)
```

```
##         genhlth exerany hlthplan smoke100 height weight wtdesire age
## 19995      good       0        1        1     69    224      224  73
## 19996      good       1        1        0     66    215      140  23
## 19997 excellent       0        1        0     73    200      185  35
## 19998      poor       0        1        0     65    216      150  57
## 19999      good       1        1        0     67    165      165  81
## 20000      good       1        1        1     69    170      165  83
##       gender
## 19995      m
## 19996      f
## 19997      m
## 19998      f
## 19999      f
## 20000      m
```

Use `dim` we can tell there are 20,000 cases and 9 variables in the data.

```r
dim(cdc)
```

```
## [1] 20000     9
```

genhlth is ordinal categorical variable; weight is continuous numberical variale; smoke100 is not ordinal categorical variable.

Numerical data

Use functions mean, var and median to calculate the mean, variance and median of certain variables of your data frame.

```r
mean(cdc$weight)
```

```
## [1] 169.683
```

```r
var(cdc$weight)
```

```
## [1] 1606.484
```

```r
median(cdc$weight)
```

```
## [1] 165
```

The function summary() returns a numerical summary: minimum, first quartile, median, mean, third quartile, and maximum.

```r
summary(cdc$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    68.0   140.0   165.0   169.7   190.0   500.0
```

Categorical data

The function table() counts the number of times each kind of category occurs in a variable. Create the frequency table for genhlth:

```r
tab<-table(cdc$genhlth)
tab
```

```
##
## excellent very good      good      fair      poor
##      4657      6972      5675      2019       677
```

Create the relative frequency table:

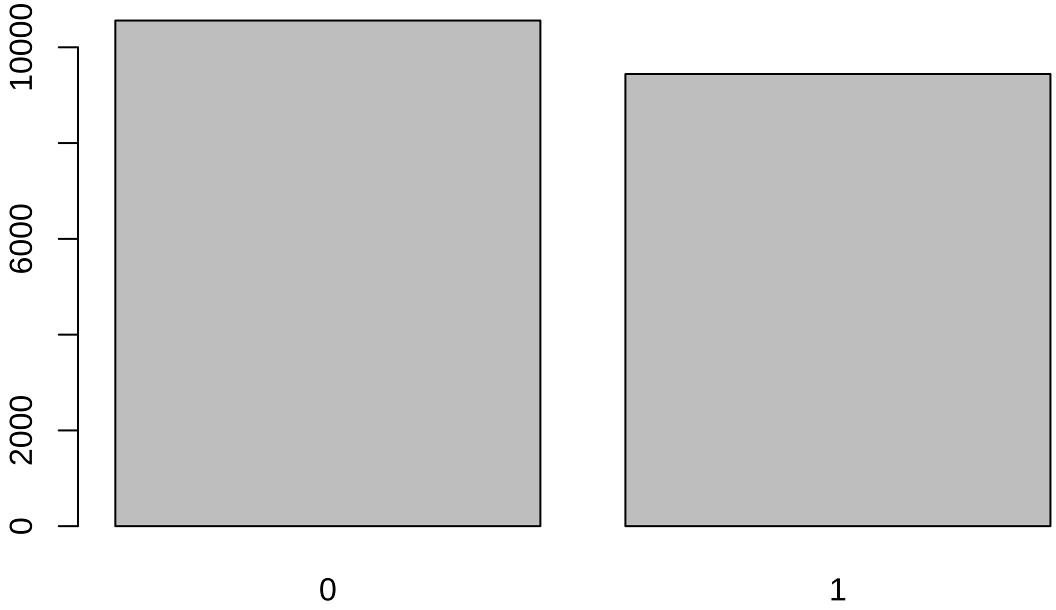
```r
tab/sum(tab)
```

```
##
## excellent very good      good      fair      poor
##   0.23285   0.34860   0.28375   0.10095   0.03385
```

Bar plot

Plotting categorical data of `smoke100`:

```r
barplot(table(cdc$smoke100))
```



Mosaic plot

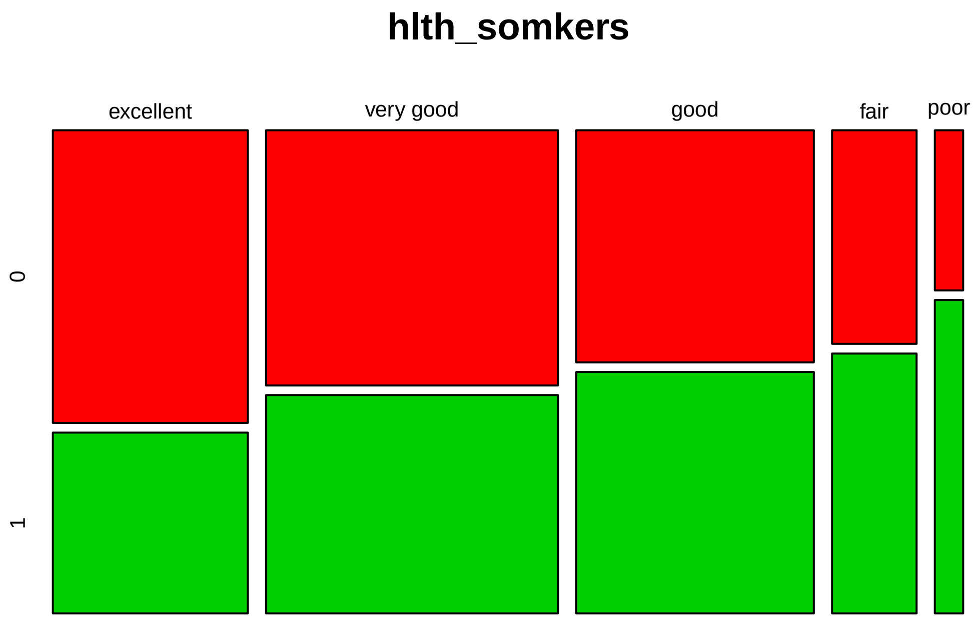
The `table` command can be used to tabulate any number of variables that you provide. This means you can investigate how different categories relate to each other. For example, you can see how many people have smoked at least 100 cigarettes in the different general health groups by executing :

```r
hlth_somkers <- table(cdc$genhlth, cdc$smoke100)
hlth_somkers
```

```
##
##                0    1
##   excellent 2879 1778
##   very good 3758 3214
##   good      2782 2893
##   fair       911 1108
##   poor       229  448
```

Plotting it:

```
mosaicplot(hlth_somkers,color=2:3,shade = FALSE)
```



Interlude: How R thinks about data

Data in a dataframe can be indexed by row(each row is a different observation) and column (each column is a different variable), index starts from 1. e.g. the height of the 1337th respondent is (use `names` we can see `height` is the 5th variable):

```
cdc[1337,5]
```

```
## [1] 70
```

You can also subset using an index range, e.g., to see the weights for the first 10 respondents you can type

```
cdc[1:10, 6]
```

```
##  [1] 175 125 105 132 150 114 194 170 150 180
```

To see all variables for specified rows(observations). e.g. All variables for the first 10 respondents:

```
cdc[1:10,]
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 2       good       0        1        1     64    125      115  33      f
## 3       good       1        1        1     60    105      105  49      f
## 4       good       1        1        0     66    132      124  42      f
## 5  very good       0        1        0     61    150      130  55      f
## 6  very good       1        1        0     64    114      114  55      f
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 9       good       0        1        1     65    150      130  27      f
## 10      good       1        1        0     70    180      170  44      m
```

This annotation also works for row, e.g. the first variable of all respondents: `cdc[,1]`. You can also use variable name here, e.g. weight of the 567th respondent :

```
cdc$weight[567]
```

```
## [1] 160
```

It's often useful to extract all individuals (cases) in a data frame that have specific characteristics. You can accomplish this through conditioning commands.

First, consider expressions like `cdc$gender == "m"` or `cdc$age > 30`. These commands produce a series of `TRUE` and `FALSE` values. There is one value for each respondent, where `TRUE` indicates that the person was male or older than 30, respectively.

Suppose now you want to extract just the data for the men in the sample, or just for those over 30. You can simply use `subset` to do that. For example, the command

```
s <- subset(cdc, cdc$gender == "m")
dim(s)
```

```
## [1] 9569    9
```

```
head(s)
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 10      good       1        1        0     70    180      170  44      m
## 11 excellent       1        1        1     69    186      175  46      m
## 12      fair       1        1        1     69    168      148  62      m
```

```
tail(s)
```

```
##         genhlth exerany hlthplan smoke100 height weight wtdesire age
## 19991 excellent       1        1        0     71    195      190  43
## 19992 very good       1        1        1     72    210      175  52
## 19993 very good       1        1        0     71    180      180  36
## 19995      good       0        1        1     69    224      224  73
## 19997 excellent       0        1        0     73    200      185  35
## 20000      good       1        1        1     69    170      165  83
##       gender
## 19991      m
## 19992      m
## 19993      m
## 19995      m
## 19997      m
## 20000      m
```

will return a data frame that only contains the men from the cdc data frame. (Note the double equal sign!)

What makes conditioning commands really powerful is the fact that you can use several of these conditions together with the logical operators `&` and `|`.

The `&` is read "and" so that `subset(cdc, cdc$gender == "f" & cdc$age > 30)` will give you the data for women over the age of 30.

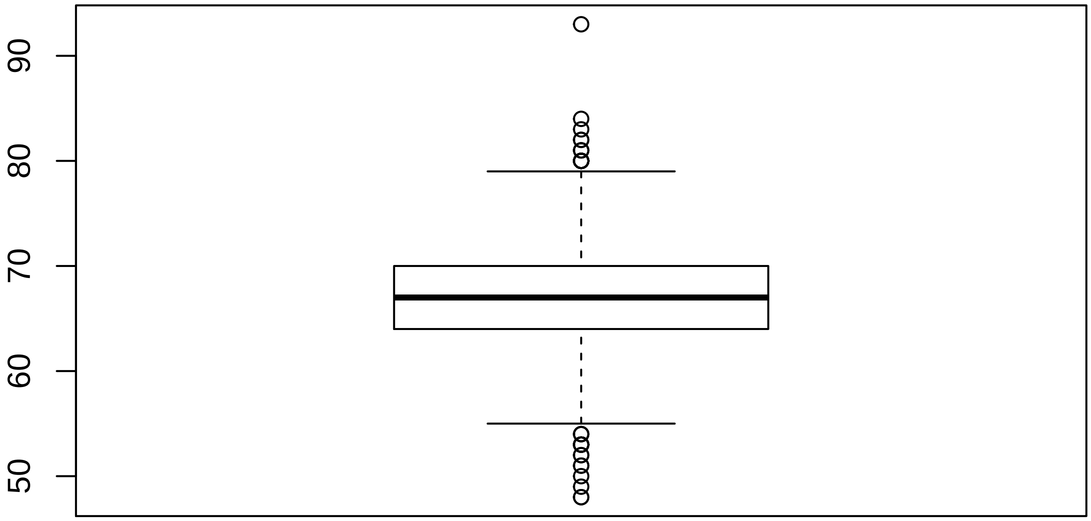
The `|` character is read "or" so that `subset(cdc, cdc$gender == "f" | cdc$age > 30)` will take people who are women or over the age of 30.

In principle, you may use as many "and" and "or" clauses as you like when forming a subset.

Plotting numerical data

Draw the box plot of the respondents heights:

```
boxplot(cdc$height)
```

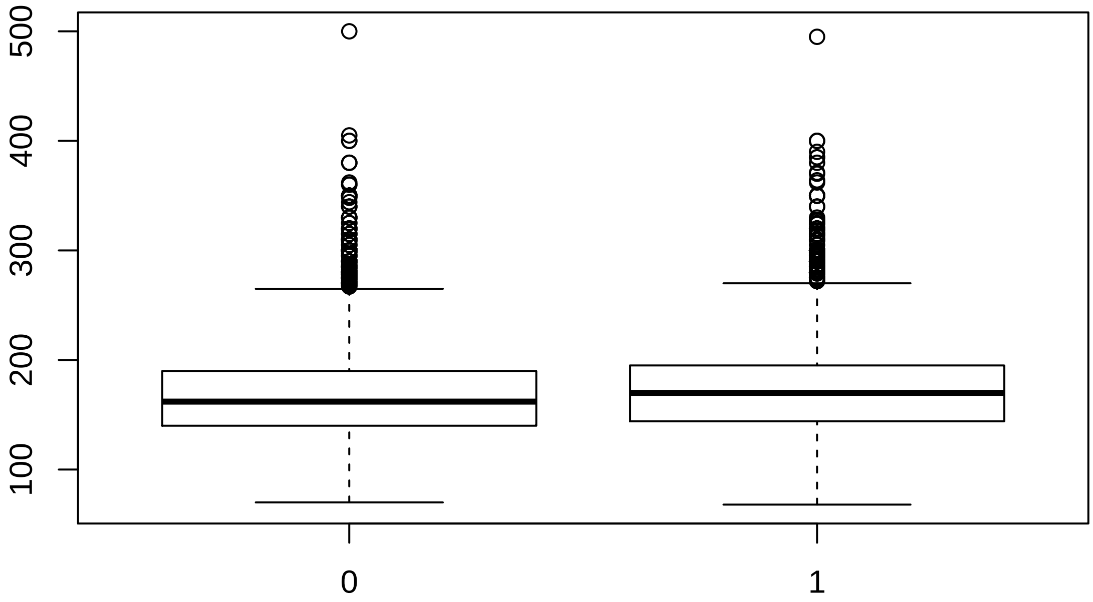


```
summary(cdc$height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   48.00   64.00   67.00   67.18   70.00   93.00
```

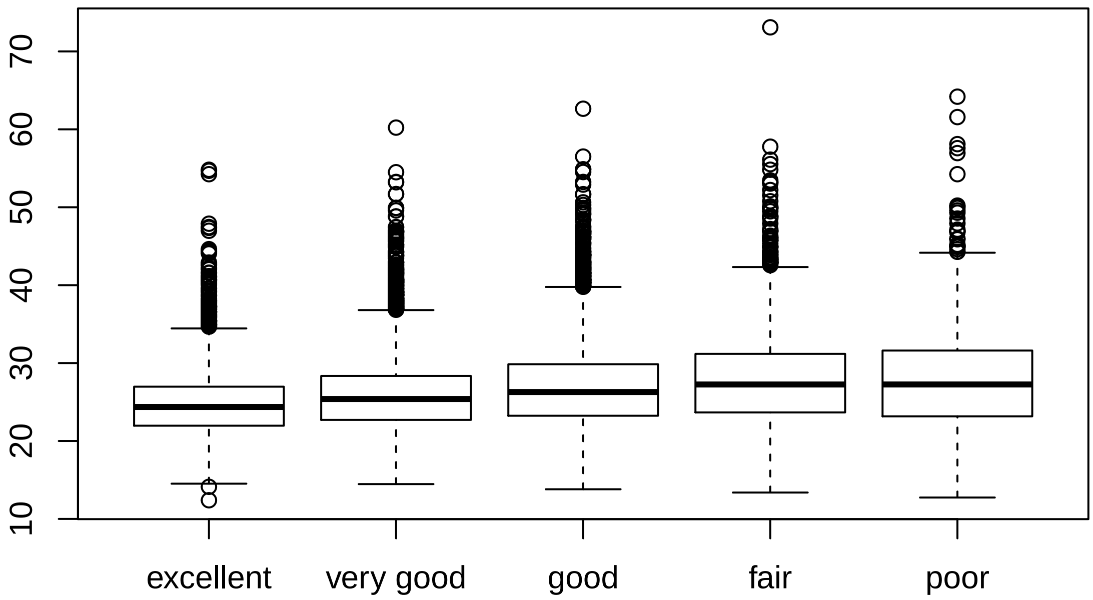
We use operator `~` (reads "versus" or "as a function of") to compare across several categories. e.g. box plot the weight of respondents as a function of whether or not they smoke.

```
boxplot(cdc$weight~cdc$smoke100)
```



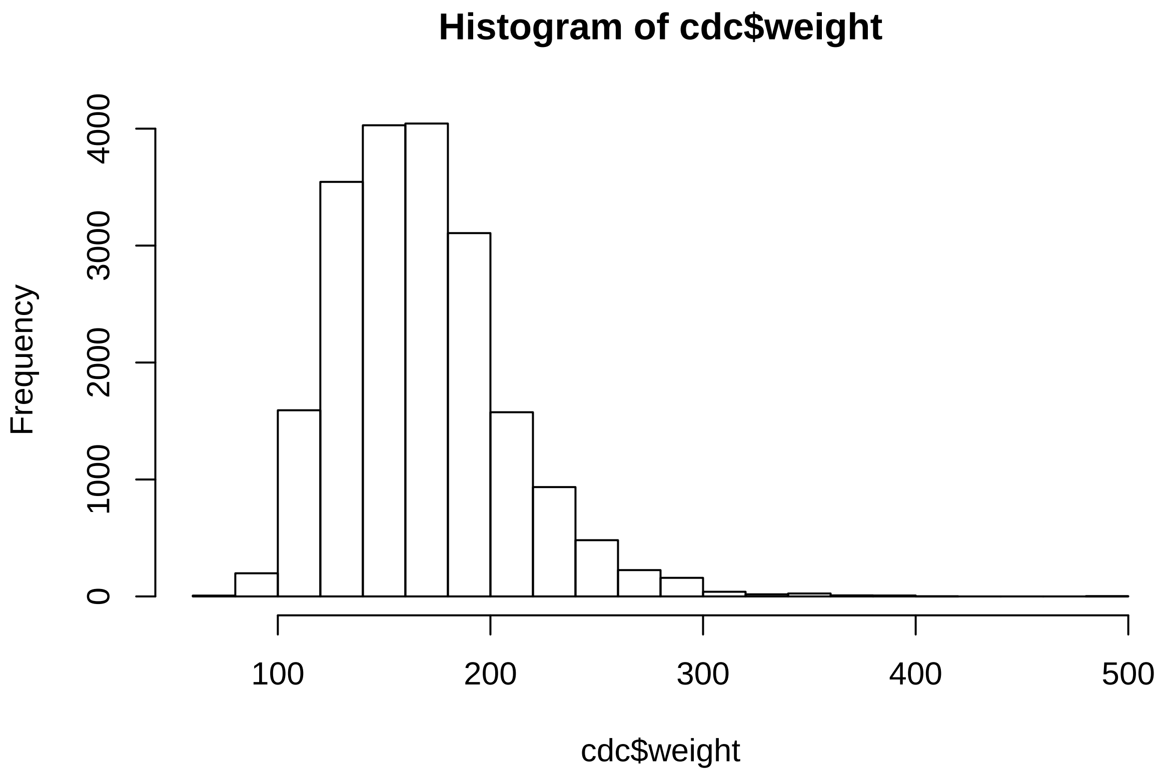
We can also do some calculations. e.g. box plot the BMI versus the general health:

```
bmi <- cdc$weight*703/cdc$height^2
boxplot(bmi~cdc$genhlth)
```

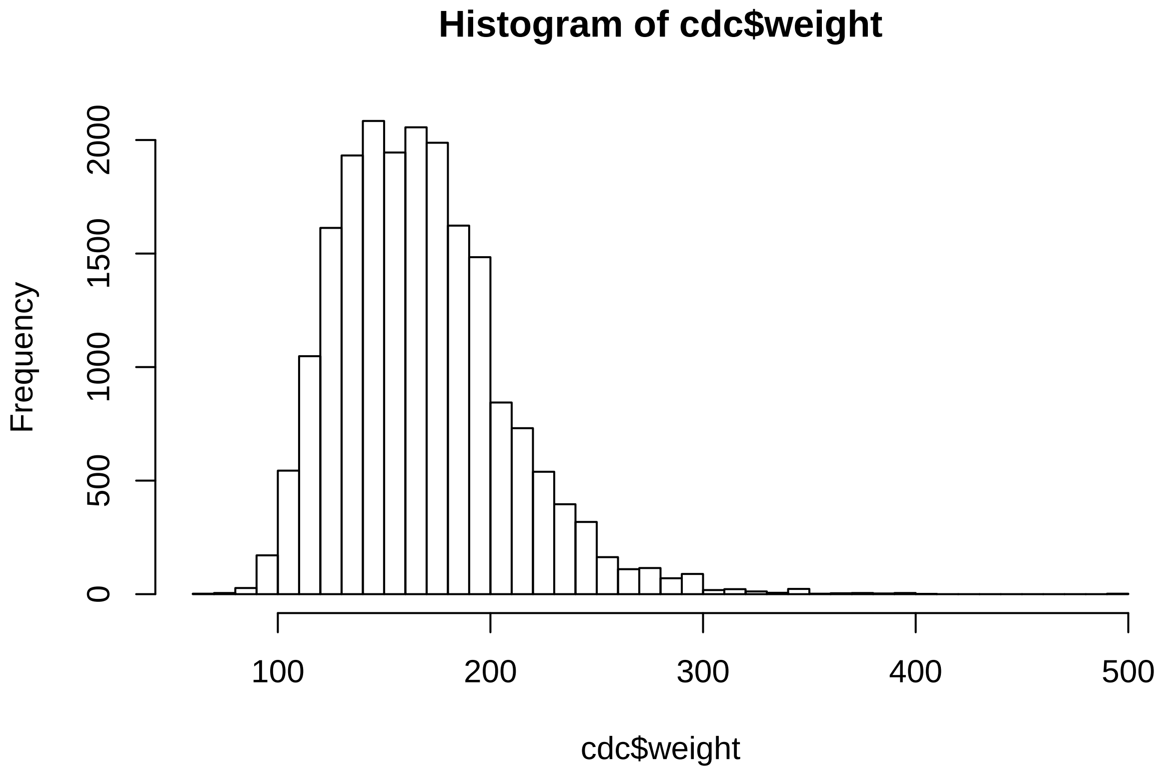


Use `breaks` parameter to set the number of bins.

```
hist(cdc$weight)
```

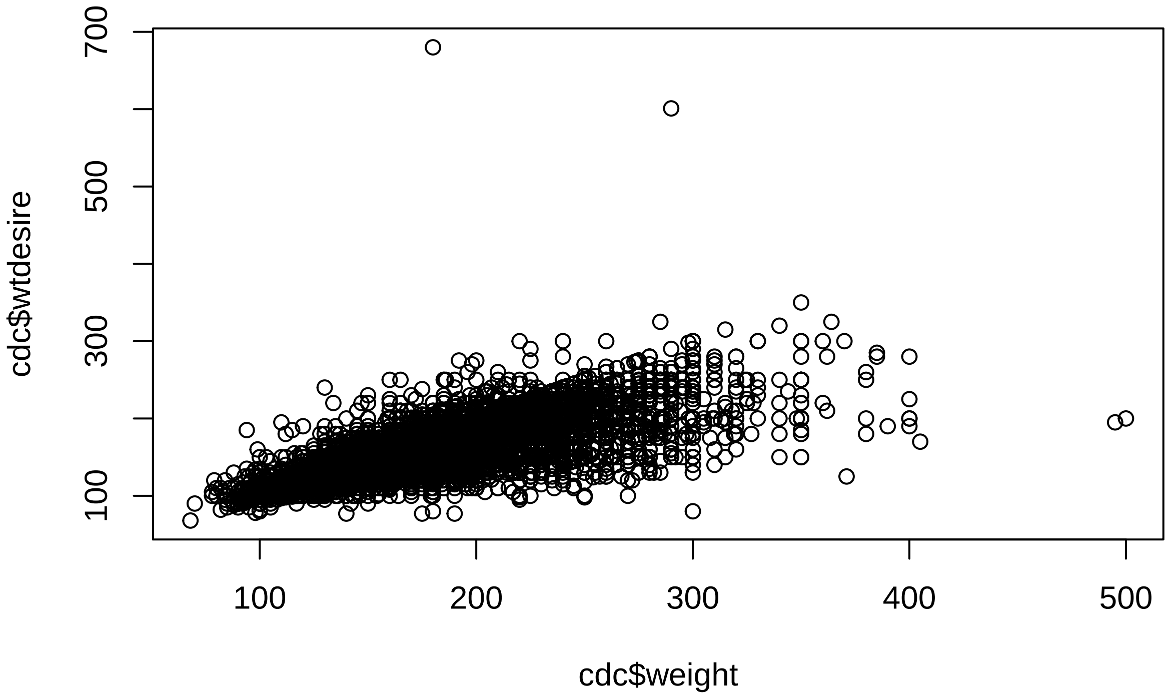


```
hist(cdc$weight,breaks=50)
```



If we plot the relationship between weight and desired weight:

```
plot(x=cdc$weight,y=cdc$wtdesire)
```



We can see moderately strong positive linear association.