Enjoy R Statistics with BK

Byoungkwan Lee, Ph. D.
Department of Advertising and Public Relations
Hanyang Iniversity

March 5, 2017

Contents

Chapter 1

What is R?

Chapter 2

Descriptive Statistics: Tabular and Graphical Methods

When analysing data, such as the marks achieved by 100 students for a piece of coursework, it is possible to use both descriptive and inferential statistics in your analysis of their marks. Typically, in most research conducted on groups of people, you will use both descriptive and inferential statistics to analyse your results and draw conclusions. So what are descriptive and inferential statistics? And what are their differences?

Descriptive statistics is the term given to the analysis of data that helps describe, show or summarize data in a meaningful way such that, for example, patterns might emerge from the data. Descriptive statistics do not, however, allow us to make conclusions beyond the data we have analysed or reach conclusions regarding any hypotheses we might have made. They are simply a way to describe our data. Descriptive statistics are very important because if we simply presented our raw data it would be hard to visulize what the data was showing, especially if there was a lot of it. Descriptive statistics therefore enables us to present the data in a more meaningful way, which allows simpler interpretation of the data. For example, if we had the results of 100 pieces of students' coursework, we may be interested in the overall performance of those students. We would also be interested in the distribution or spread of the marks. Descriptive statistics allow us to do this.

Typically, there are two general types of statistic that are used to describe data:

Measures of central tendency: these are ways of describing the central position of a frequency distribution for a group of data. In this case, the frequency distribution is simply the distribution and pattern of marks scored by the 100 students from the lowest to the highest. We can describe this central position using a number of statistics, including the mode, median, and mean. You can

read about measures of central tendency here.

Measures of spread: these are ways of summarizing a group of data by describing how spread out the scores are. For example, the mean score of our 100 students may be 65 out of 100. However, not all students will have scored 65 marks. Rather, their scores will be spread out. Some will be lower and others higher. Measures of spread help us to summarize how spread out these scores are. To describe this spread, a number of statistics are available to us, including the range, quartiles, absolute deviation, variance and standard deviation.

When we use descriptive statistics it is useful to summarize our group of data using a combination of tabulated description (i.e., tables), graphical description (i.e., graphs and charts) and statistical commentary (i.e., a discussion of the results).

Descriptive Statistics are used to condense and summarize data that have been collected. Can be done in three ways:

1) Tables
2) Graphs or figures
3) Statistic: A rule that reduces data to a single number

2.1 Making tables

2.1.1 Frequency distribution

A tabular summary of data showing the frequency (or number) of items in each of several nonoverlapping classes. The objective is to provide insights about the data that cannot be quickly obtained by looking only at the original data.

Example: President Hotel

Guests staying at President Hotel were asked to rate the quality of their accommodations as being excellent, above average, average, below average, or poor. The ratings provided by a sample of 20 guests are shown below.

Evaluation on Customer Satisfaction of Guest House

Below Average	Average	Above Average
Above Average	Above Average	Above Average
Above Average	Below Average	Below Average
Average	Poor	Poor
Above Average	Excellent	Above Average
Average	Above Average	Average
Above Average	Average	

```
setwd("~/Google Drive/furious lion/king/Big Mac/R Book/Descriptive Stats")
bk <- read.csv("guest_house.csv", header=TRUE) # Reading data
attach(bk)
detach(bk)
```

```
class(bk$evlauation)

## [1] "integer"

guest.house <- ordered(bk$evlauation,
        levels = c(1, 2, 3, 4, 5),
        labels = c("poor", "below average", "average",
                   "above average", "excellent"))
table(guest.house)

## guest.house
##          poor below average       average above average     excellent
##             2             3             5             9             1

data.frame(table(guest.house))

##     guest.house Freq
## 1          poor    2
## 2 below average    3
## 3       average    5
## 4 above average    9
## 5     excellent    1
```

2.1.2 Relative frequency distribution table

```
prop.table(table(guest.house))

## guest.house
##          poor below average       average above average     excellent
##          0.10          0.15          0.25          0.45          0.05

data.frame(prop.table(table(guest.house)))
```

```
##     guest.house Freq
## 1          poor 0.10
## 2 below average 0.15
## 3       average 0.25
## 4 above average 0.45
## 5     excellent 0.05
```

```
round(prop.table(table(guest.house)), 1)
```

```
## guest.house
##          poor below average       average above average     excellent
##           0.1           0.2           0.2           0.4           0.0
```

```
round(100*(prop.table( table(guest.house))))
```

```
## guest.house
##          poor below average       average above average     excellent
##            10            15            25            45             5
```

2.1.3 Cumulative frequency distribution tables

```
cumsum(prop.table(table(guest.house)))
```

```
##          poor below average       average above average     excellent
##          0.10          0.25          0.50          0.95          1.00
```

How about percentages of these values?
Let's make tables by using a package, called "*COUNT*".

```
require(COUNT)
myTable(guest.house)
```

```
##               x Freq Prop CumProp
## 1          poor    2 0.10    0.10
## 2 below average    3 0.15    0.25
## 3       average    5 0.25    0.50
## 4 above average    9 0.45    0.95
## 5     excellent    1 0.05    1.00
```

Anothe Examples
Let's practice making tables by using a real data set.

```
bk2<- read.csv("rawdata_2012.csv", header=T) # Reading data
names(bk2)
```

```
##   [1] "ID"     "age"    "SQ1_2"  "SQ2"    "SQ3"    "PA1_1"  "PA1_2"
##   [8] "PA1_3"  "PA1_4"  "PA1_5"  "PA1_6"  "PA2_1"  "PA2_2"  "PA2_3"
##  [15] "PA2_4"  "PA2_5"  "PA2_6"  "PA2_7"  "PA2_8"  "PA2_9"  "PA2_10"
##  [22] "PA2_11" "PA3_1"  "PA3_2"  "PA3_3"  "PA3_4"  "PA3_5"  "PA3_6"
##  [29] "PA3_7"  "PA3_8"  "PA3_9"  "PA3_10" "PA3_11" "PA3_12" "PA3_13"
##  [36] "PA3_14" "PA3_15" "PA3_16" "PA3_17" "PA4_1"  "PA4_2"  "PA4_3"
##  [43] "PA4_4"  "PA4_5"  "PA4_6"  "PA4_7"  "PB1"    "PB2"    "PB3_1"
##  [50] "PB3_2"  "PB3_3"  "PB3_4"  "PB3_5"  "PB3_6"  "PB3_7"  "PB3_8"
##  [57] "PB3_9"  "PB3_10" "PB3_11" "PB4_1"  "PB4_2"  "PB4_3"  "PB4_4"
##  [64] "PB4_5"  "PB4_6"  "PB4_7"  "PB4_8"  "PB4_9"  "PB4_10" "PB4_11"
##  [71] "PB5"    "PC1"    "PC2"    "PC3"    "PC4"    "PC5"    "PD1"
##  [78] "PD2"    "PD3"    "PD4"    "PD5"    "PD6"    "A1"     "A2"
##  [85] "A3"     "B1"     "B2"     "B3"     "B4"     "B5"     "B6_1"
##  [92] "B6_2"   "B6_3"   "C1"     "C2"     "C3"     "C4"     "C5"
##  [99] "C6"     "C7_1"   "C7_2"   "D1"     "D2"     "D3"     "D4"
## [106] "D5"     "D6"     "D7_1"   "D7_2"   "D8_1"   "D8_2"   "D9"
## [113] "D10"    "x"
```

```
dim(bk2)
```

```
## [1] 1000  114
```

In this data set, *SQ3* is gender variable

```
bk2$gender <- ordered(bk2$SQ3,
                      levels = c(1,2),
                      labels = c("male", "female"))
class(bk2$gender)
```

```
## [1] "ordered" "factor"
```

```
myTable(bk2$gender)
```

```
##        x Freq  Prop CumProp
## 1   male  513 0.513   0.513
## 2 female  487 0.487   1.000
```

Guidelines for Selecting Number of Classes

Use between 5 and 20 classes. Data sets with a larger number of elements usually require a larger number of classes. Smaller data sets usually require fewer classes.

```
summary(bk2$age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   28.00   39.00   38.23   49.00   59.00

class(bk2$age)

## [1] "integer"

bat <- myTable(bk2$age)
range(bk2$age)

## [1] 15 59

span <- seq(10, 60, by = 10)
Aage <- cut(bk2$age, span, right=FALSE)
levels(Aage) <- c("10th", "20th", "30th", "40th", "50th")
myTable(Aage)

##      x Freq  Prop CumProp
## 1 10th   93 0.093   0.093
## 2 20th  193 0.193   0.286
## 3 30th  238 0.238   0.524
## 4 40th  260 0.260   0.784
## 5 50th  216 0.216   1.000
```

The option, *right=FALSE* makes that the lower value of the next interval will be excluded from the present interval. Now the interval goes from 3.0 to 3.99999....

Here is another way to make a table with a numeric variable.

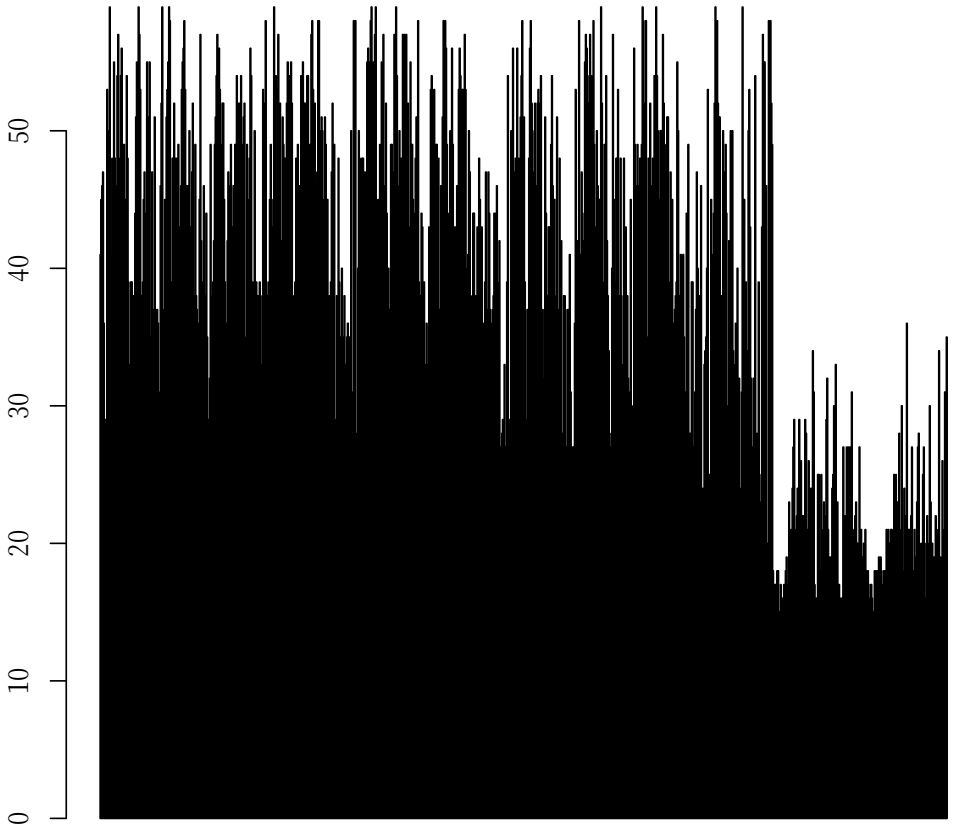
```
Age <- cut(bk2$age, br=c(0, 19, 29, 39, 49, 59),
           labels=c("10th", "20th", "30th", "40th", "50th"))
myTable(Age)

##      x Freq  Prop CumProp
## 1 10th   93 0.093   0.093
## 2 20th  193 0.193   0.286
## 3 30th  238 0.238   0.524
```

```
## 4 40th  260 0.260   0.784
## 5 50th  216 0.216   1.000
```

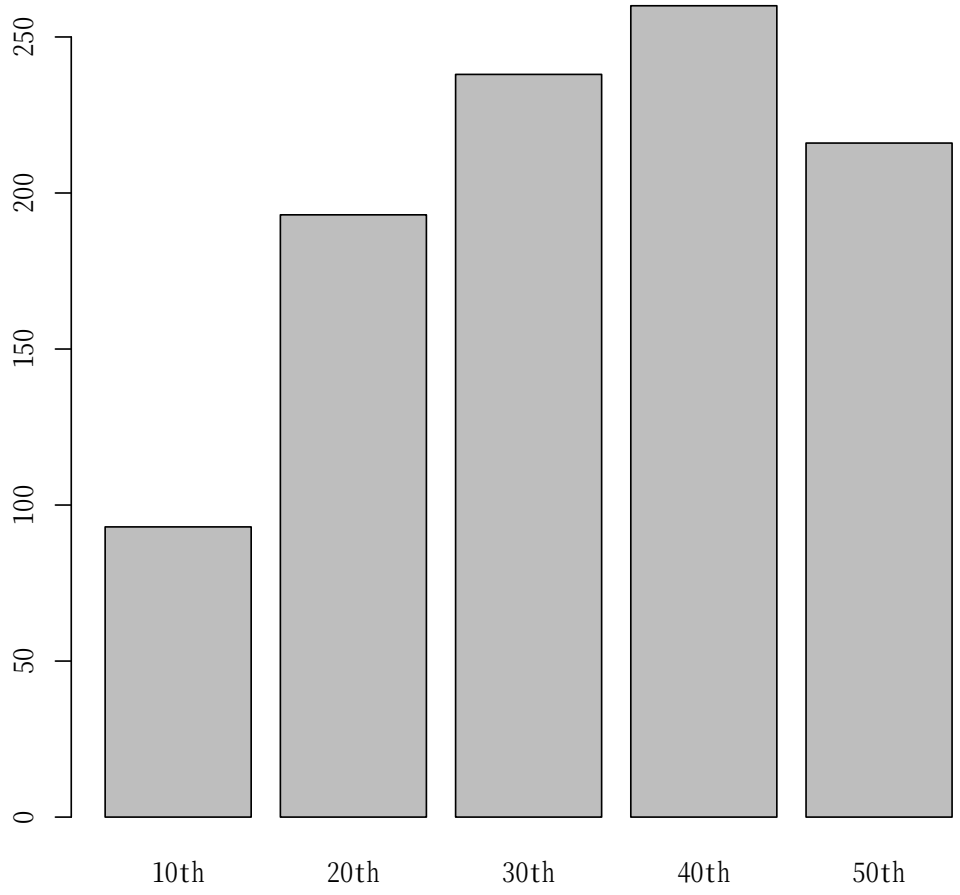
2.2 Making graphs

2.2.1 Bar Plots

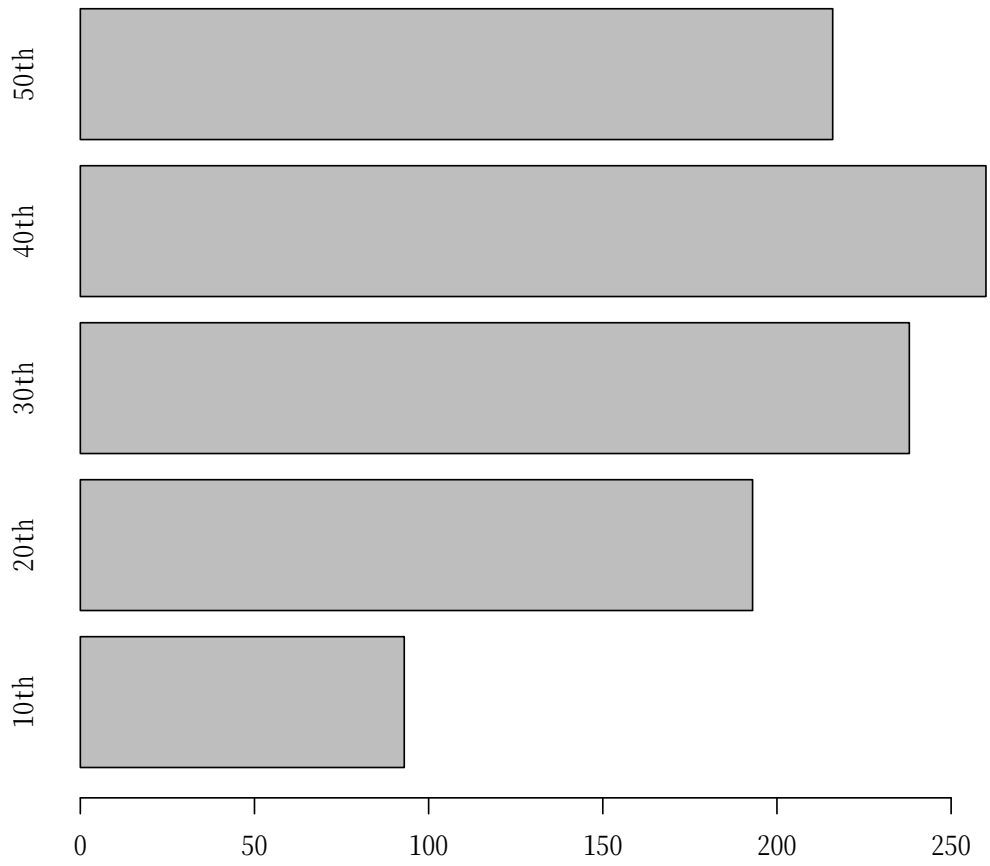
```
barplot(bk2$age)
```



```
barplot(table(Aage))
```

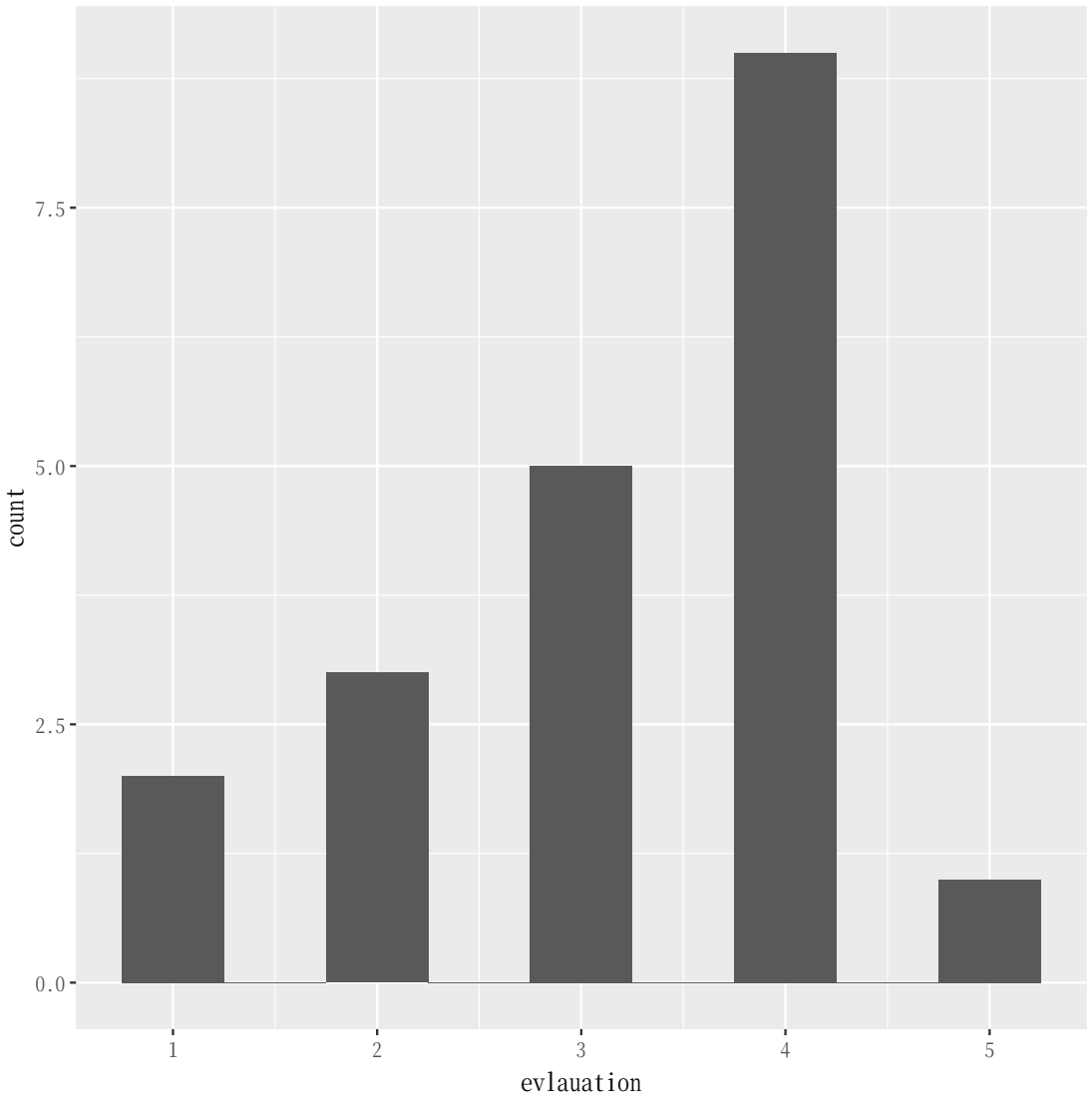



```
barplot(table(Aage), horiz = TRUE)
```

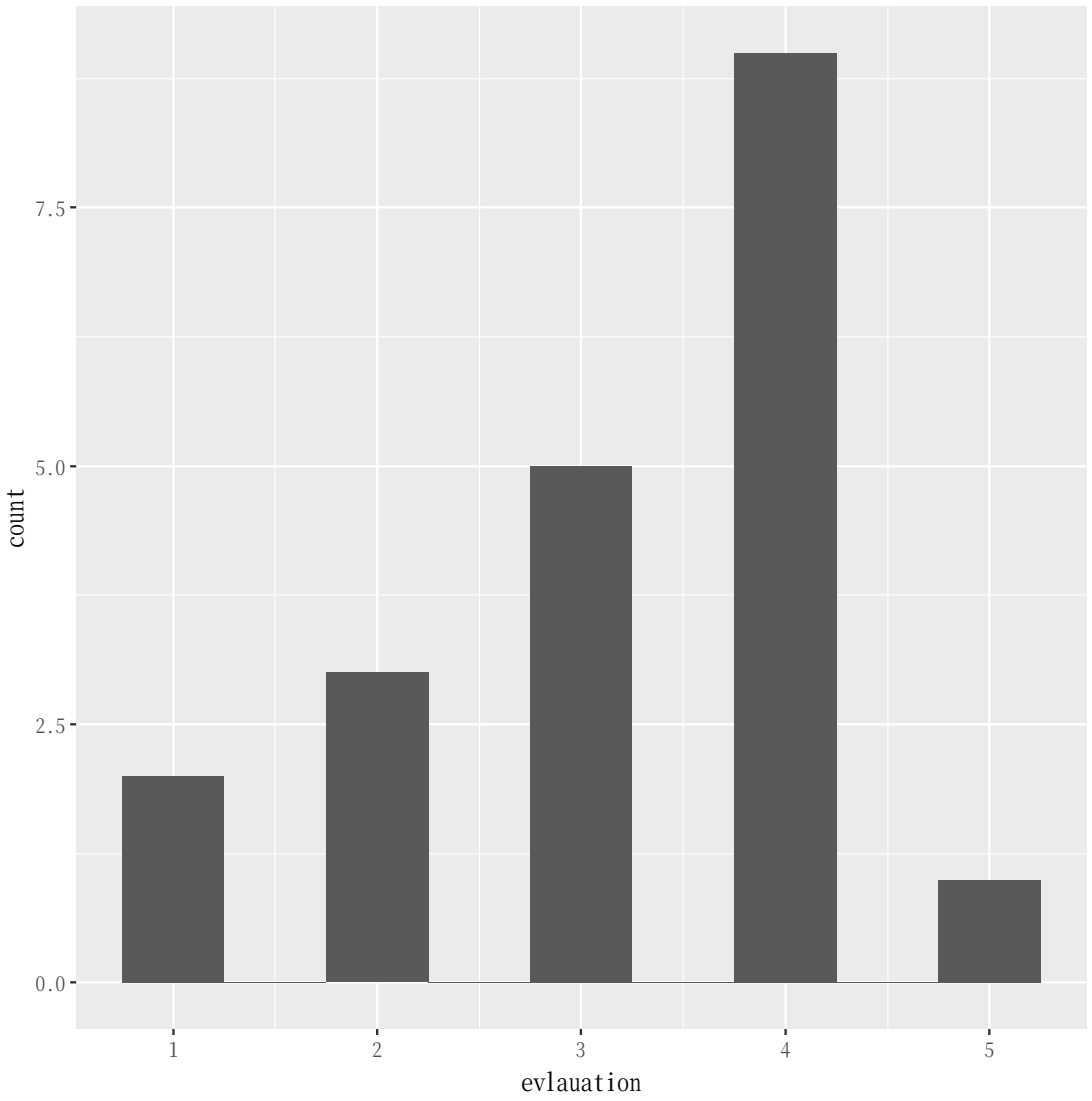


Let's make plots by using a package, called "*ggplot2*".

```
library(ggplot2)
ggplot(bk, aes(x=evlauation), binwidth = x) + geom_bar(stat="bin", binwidth = .5)

## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please
use `geom_histogram()` instead.
```

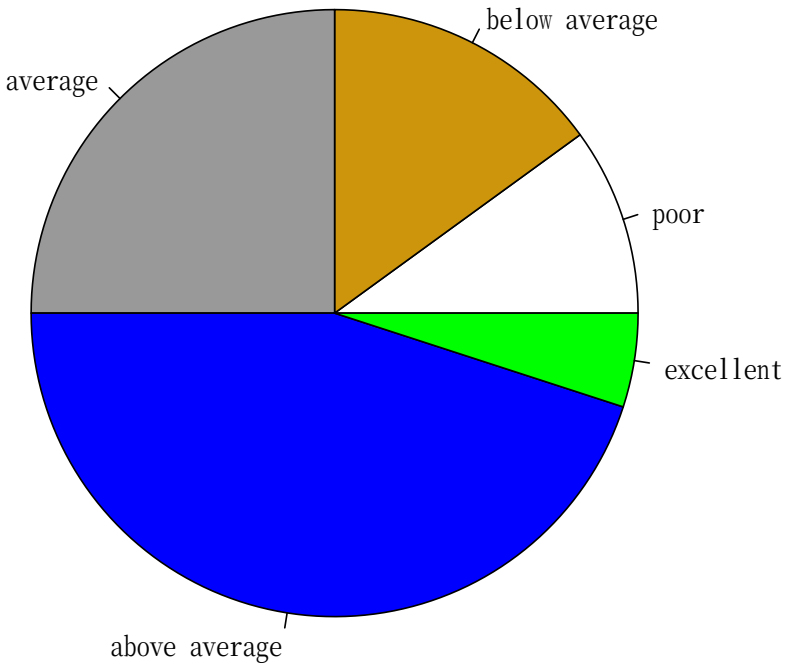



```
qplot(evlauation, data=bk, geom="bar", binwidth=.5)

## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please
use `geom_histogram()` instead.
```



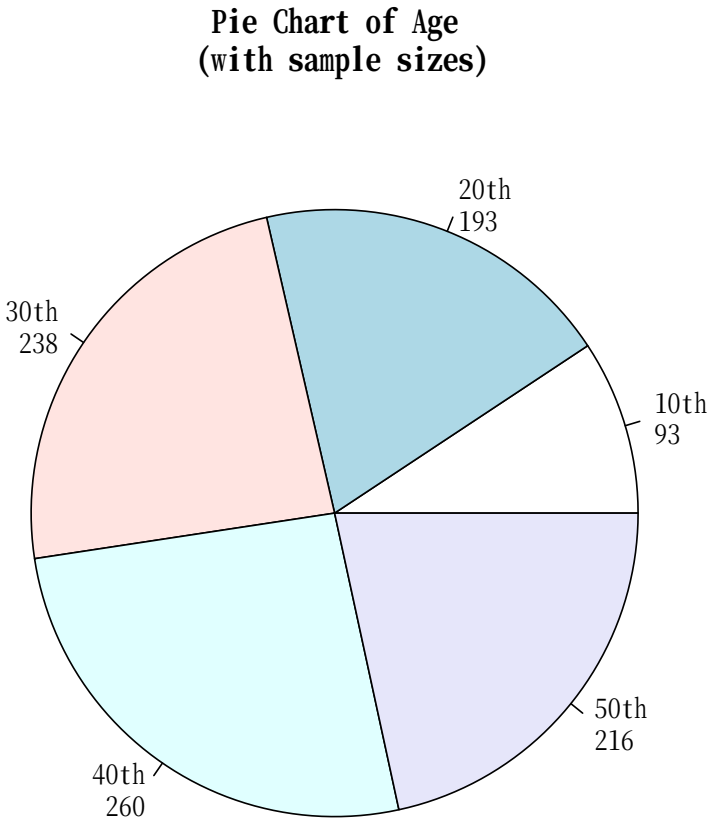
2.2.2 Pie chart

```
pie(table(guest.house), col= c("white","darkgoldenrod3","gray60","blue","green"))
```



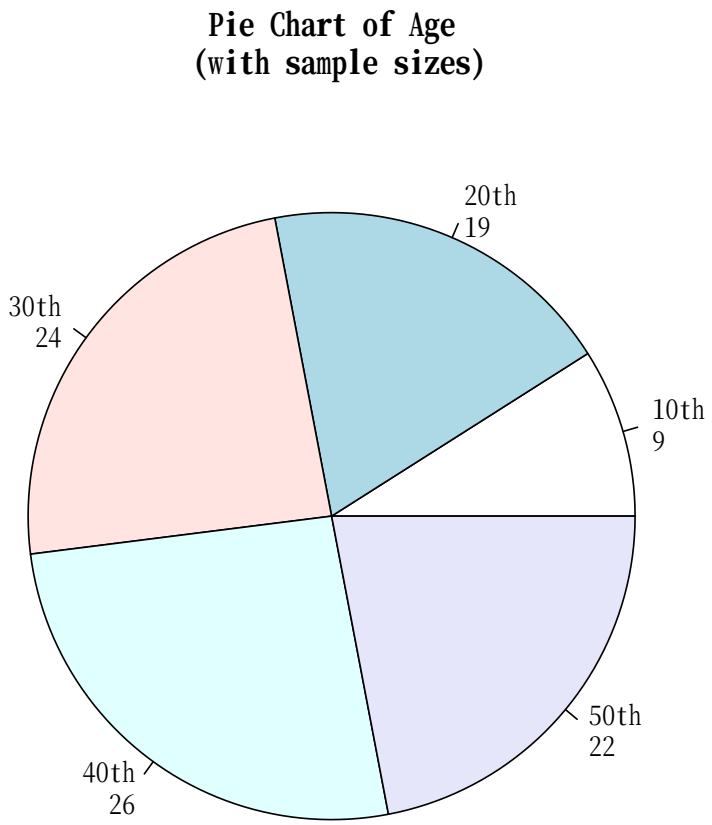
Pie Chart from data frame with Appended Sample Sizes

```
mytable <- table(Aage)
Age <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable, labels = Age, main="Pie Chart of Age\n (with sample sizes)")
```



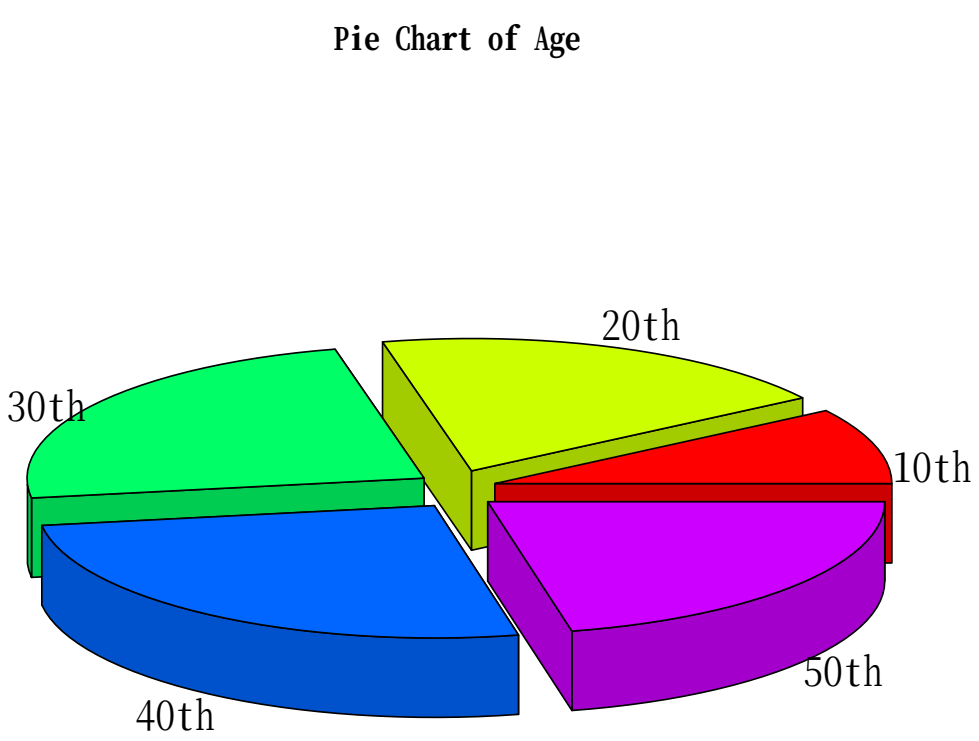
Pie Chart from data frame with Pecent

```
mytable <- round(100* (prop.table(table(Aage))))
Age <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable, labels = Age, main="Pie Chart of Age\n (with sample sizes)")
```

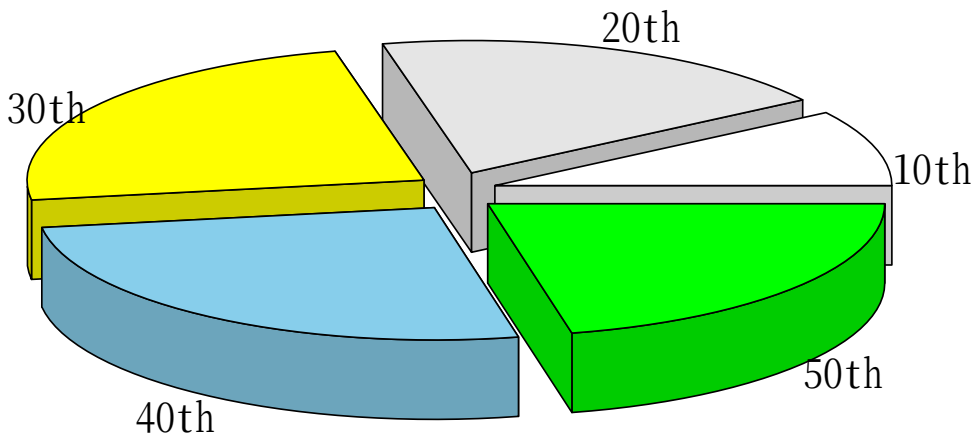


3D Exploded Pie Chart

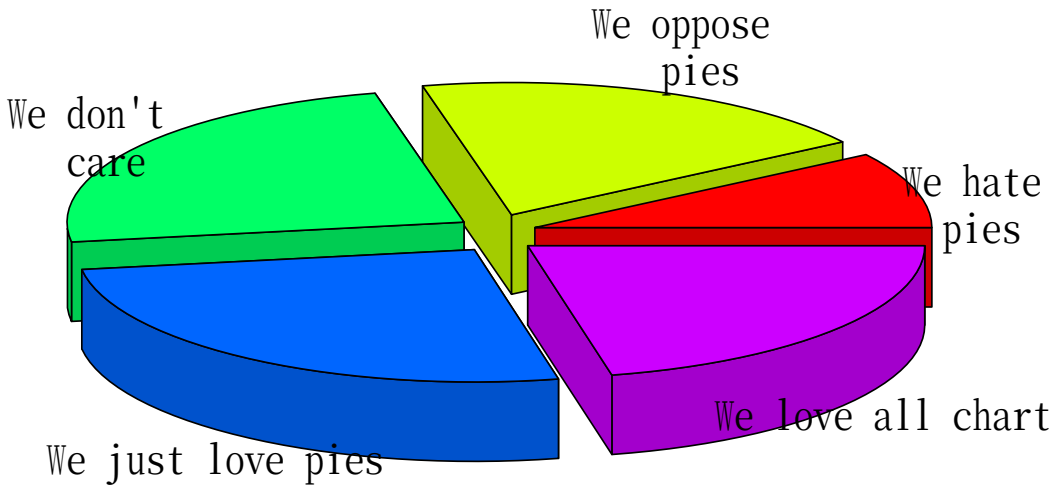
```
library(plotrix)
slices <- c(93, 193, 238, 260, 216)
lbls <- c("10th", "20th", "30th", "40th", "50th")
pie3D(slices,labels=lbls,explode=0.1, main="Pie Chart of Age")
```




```
pie3D(slices,labels=lbls,explode=0.1, col=c("white","gray90",
                                            "yellow","sky blue","green"),
                                            main="Pie Chart of Age")
```

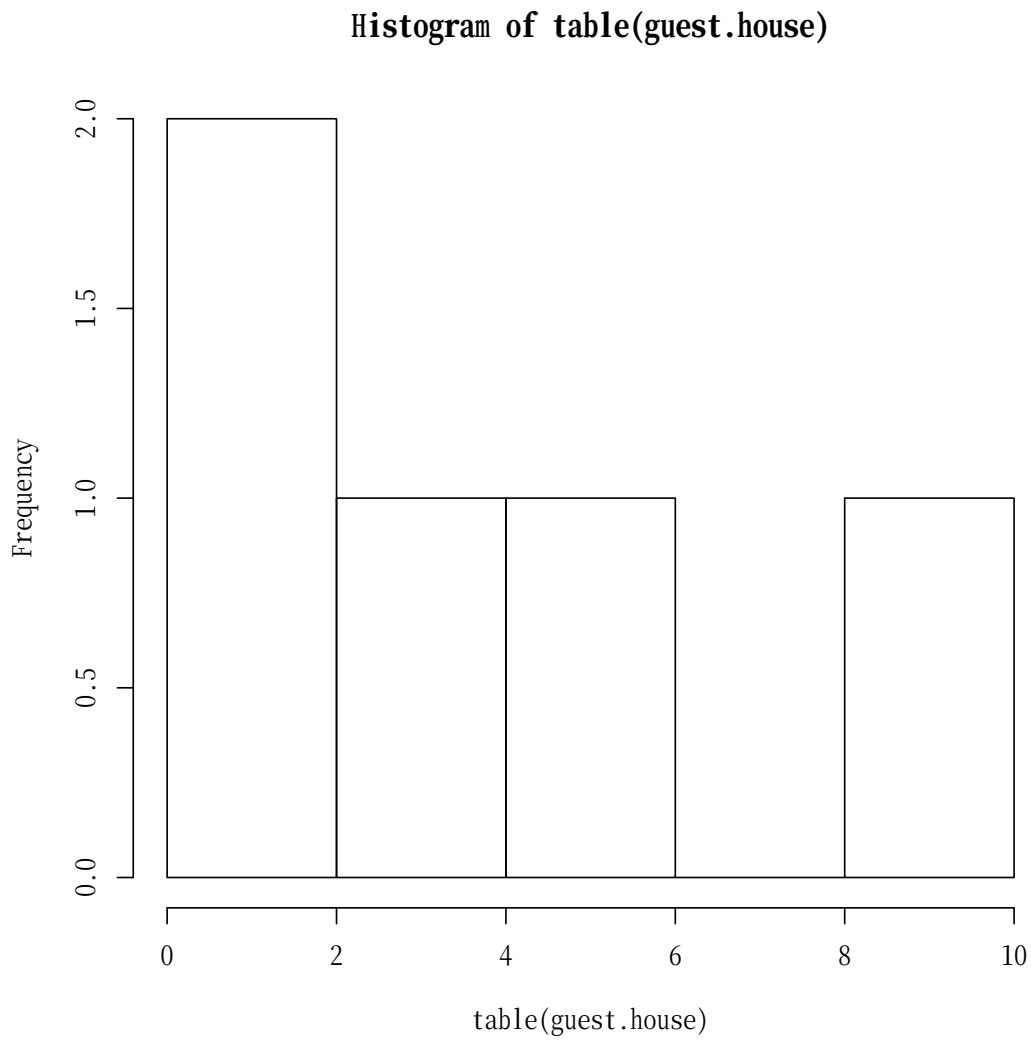



```
pieval <- c(93, 193, 238, 260, 216)
bisectors <- pie3D(pieval,explode=0.1,main="3D PIE OPINIONS")
pielabels <- c("We hate\n pies","We oppose\n  pies",
               "We don't\n  care","We just love pies", "We love all chart")
pie3D.labels(bisectors,labels=pielabels)
```

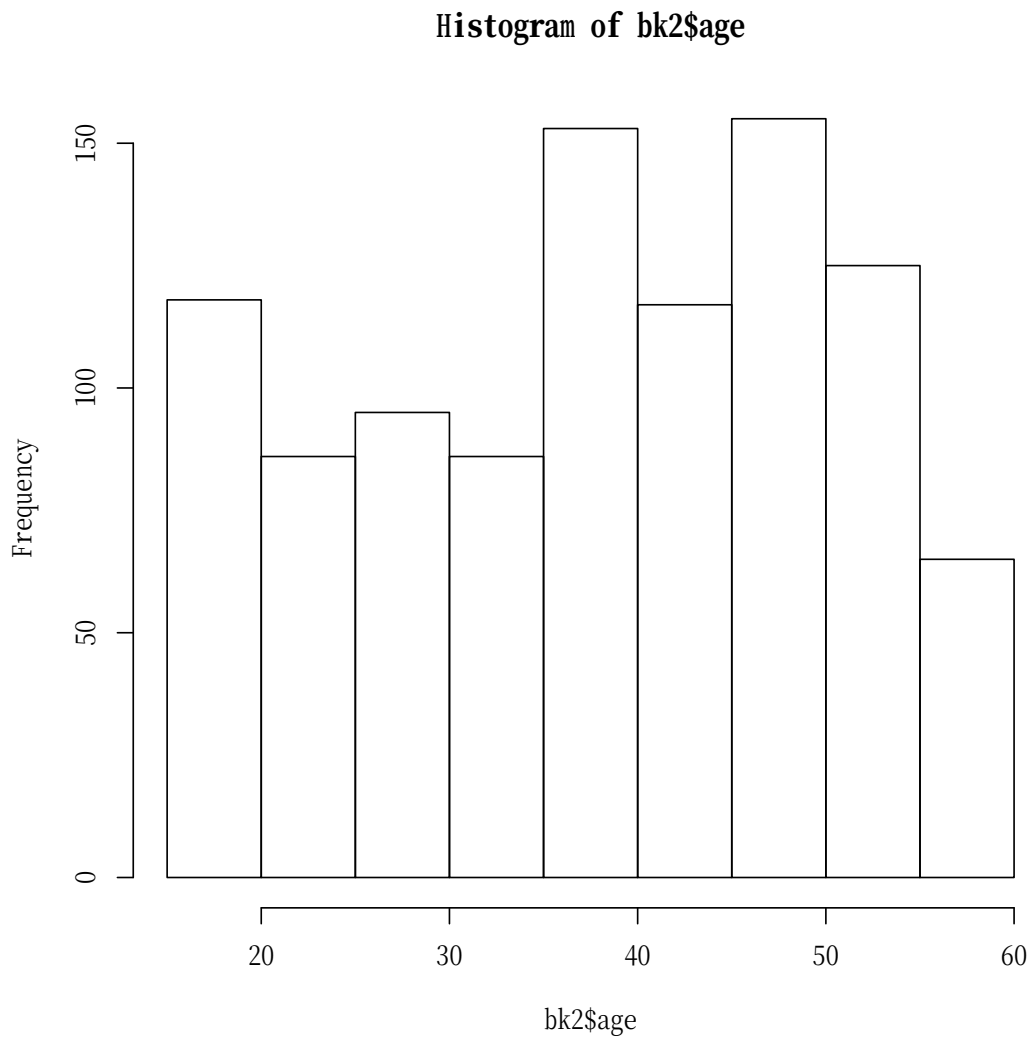


2.2.3 Histogram

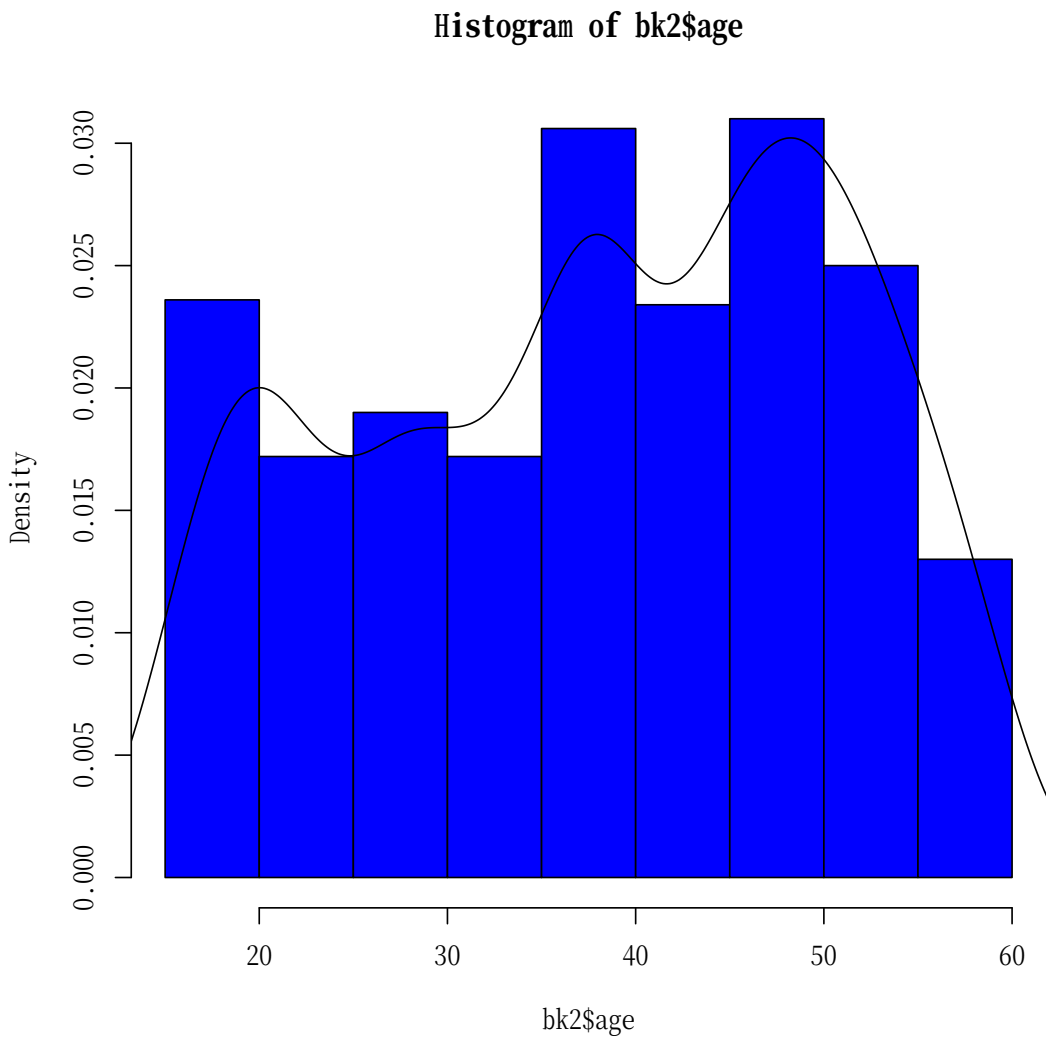
```
hist(table(guest.house))
```



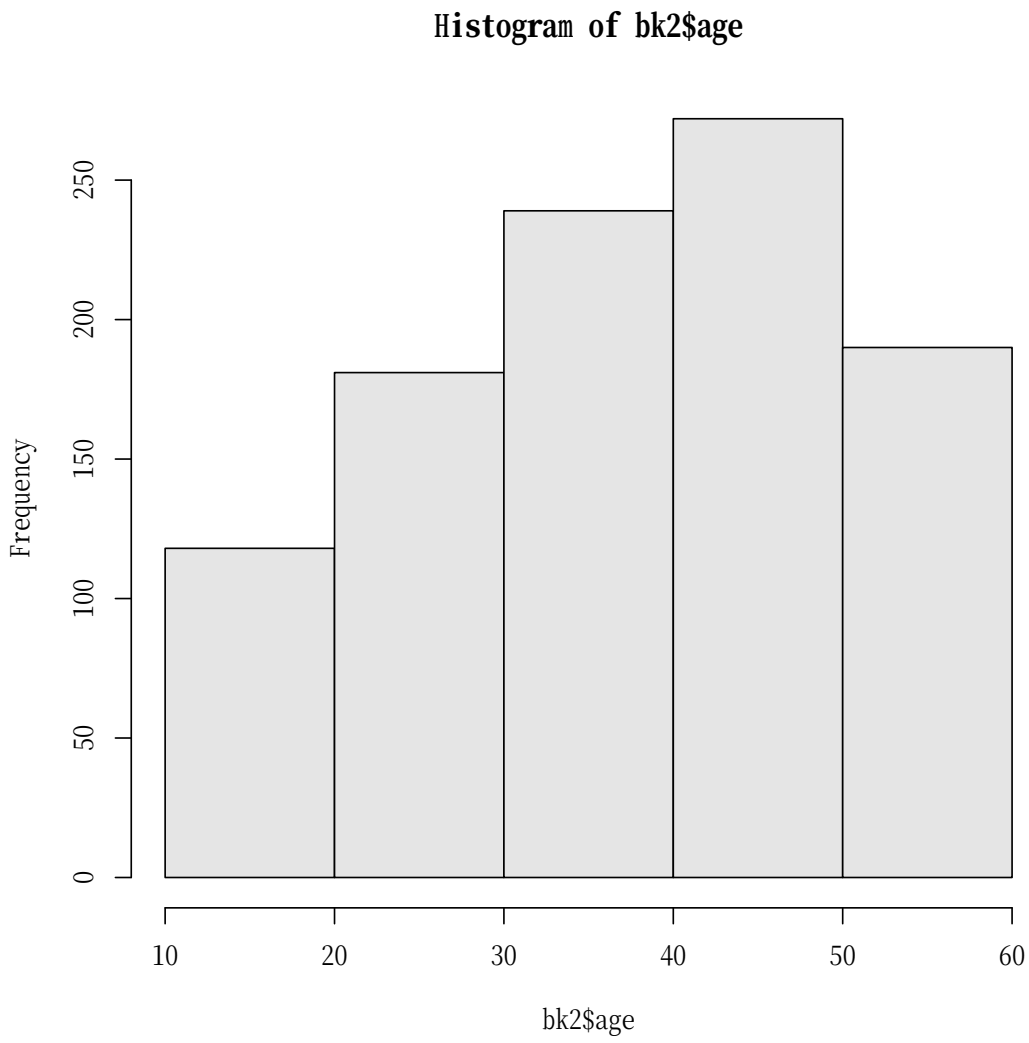

```
hist(bk2$age, breaks=10)
```



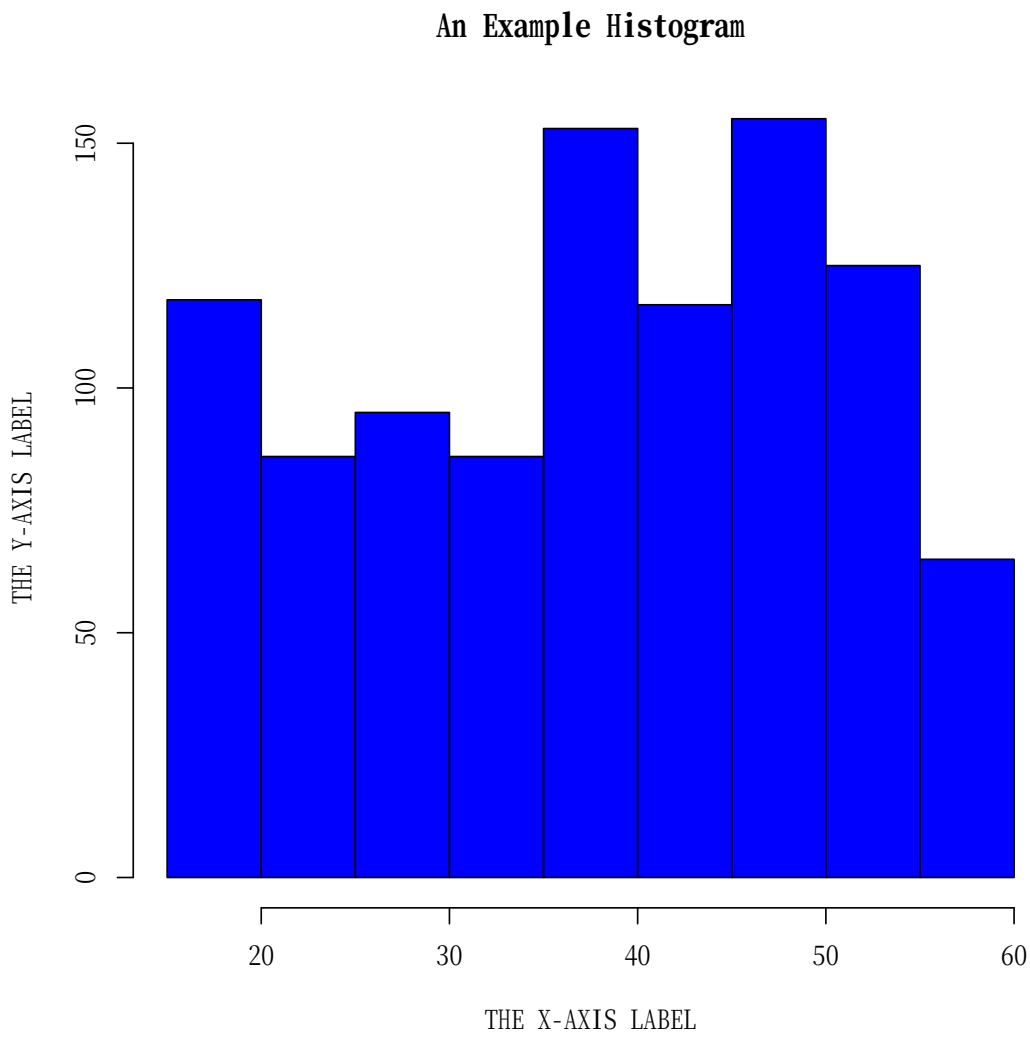

```
hist(bk2$age, breaks=10, probability = TRUE, col="blue")
lines( density(bk2$age) )
```



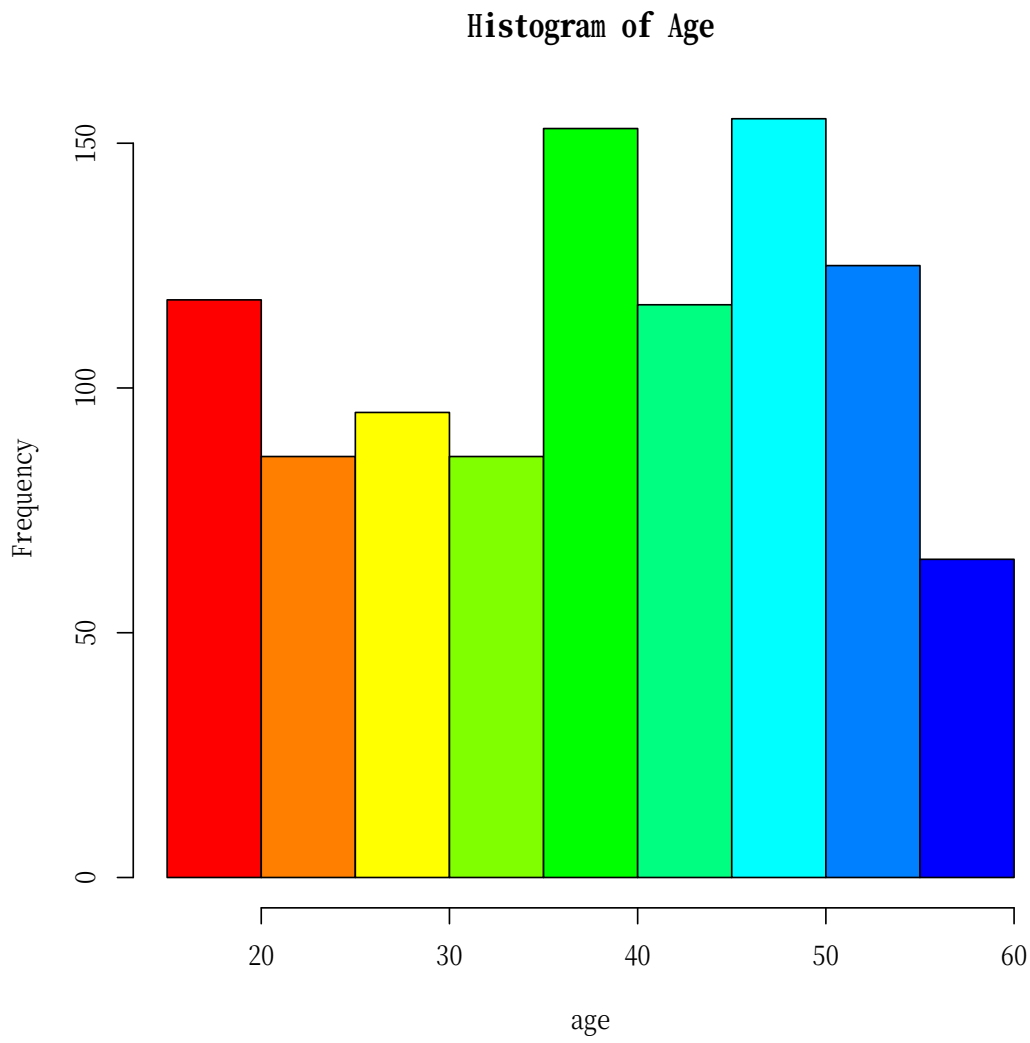

```
hist(bk2$age, col = "gray90", breaks= seq(10, 60, by=10) )
```




```
hist(bk2$age,
     col="blue",                              ### define your color
     main="An Example Histogram",  ### main label
     xlab="THE X-AXIS LABEL",          ### x-label
     ylab="THE Y-AXIS LABEL")          ### y-label
```

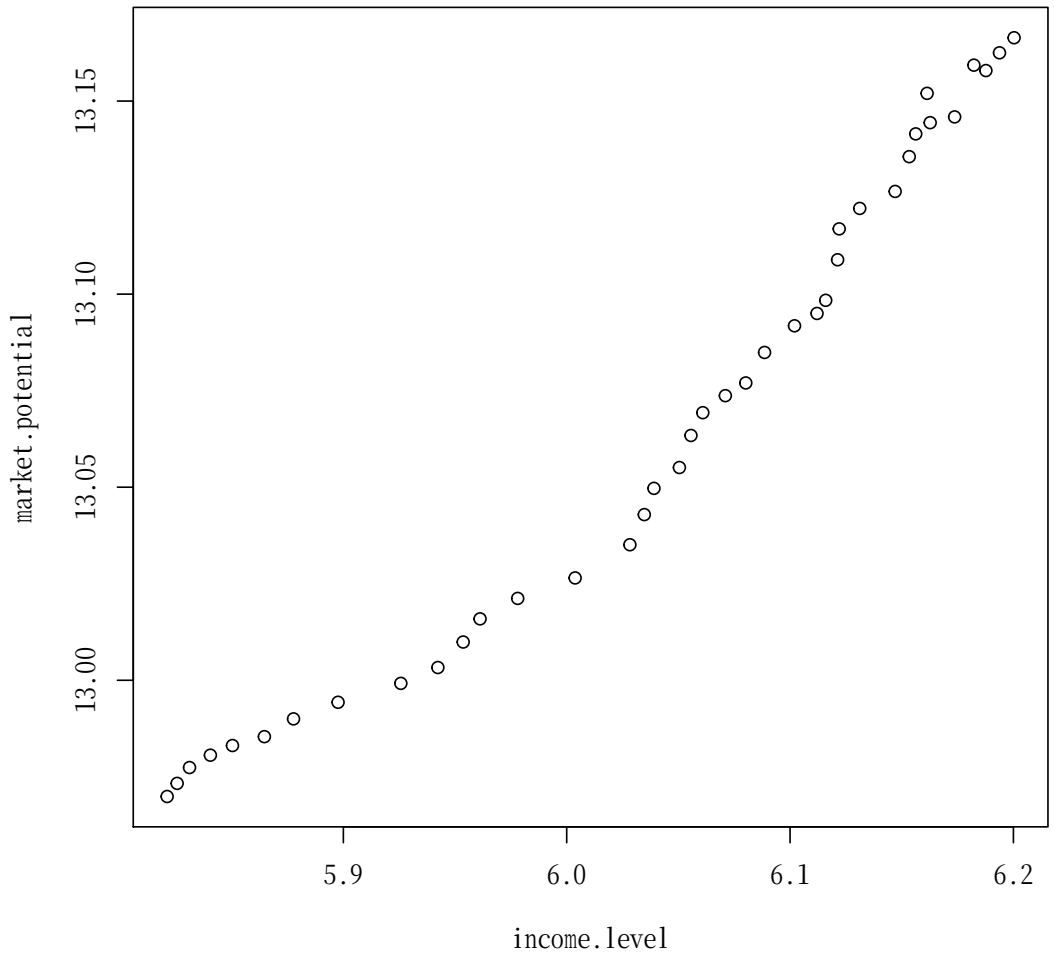



```
hist(bk2$age,     # apply the hist function
     col=rainbow(12),     # adds a rainbow pattern to graphics
     main="Histogram of Age", # the main title
     xlab="age")       # x-axis label
```

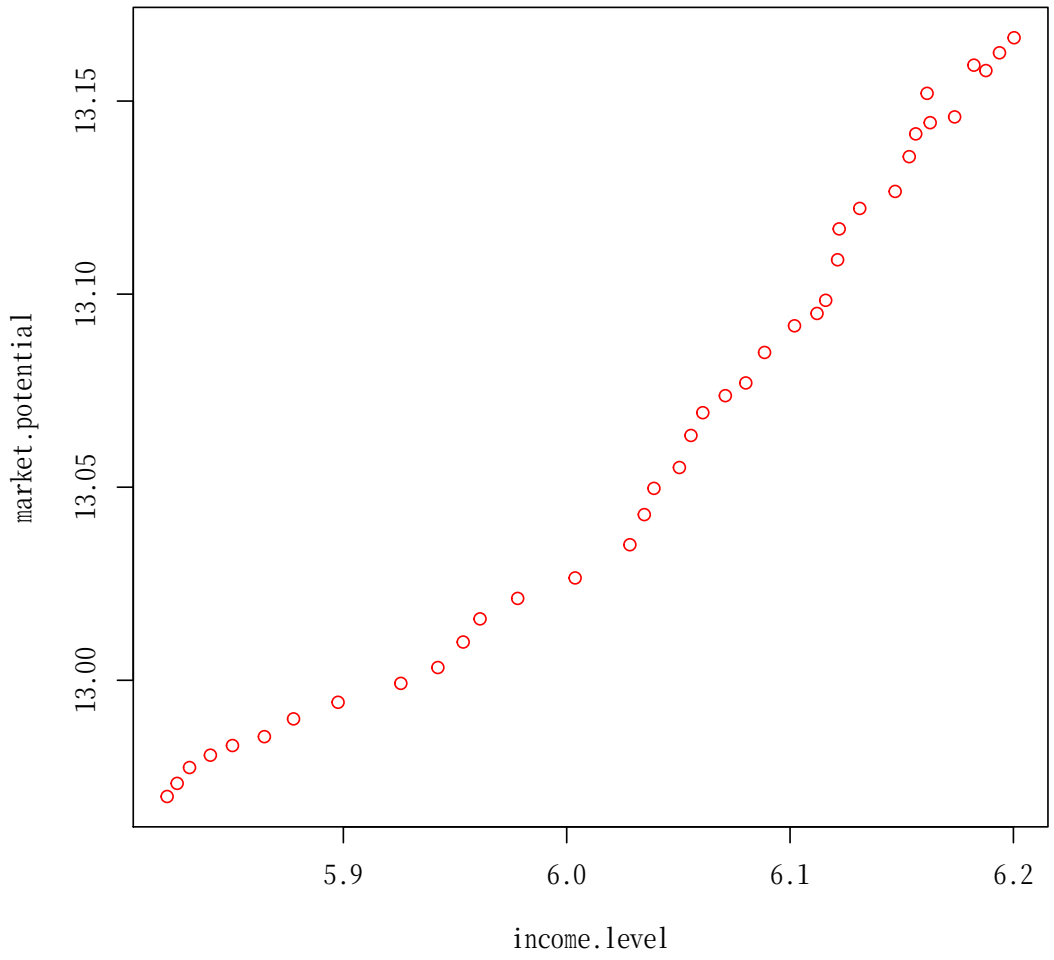


2.2.4 Scatter Diagram

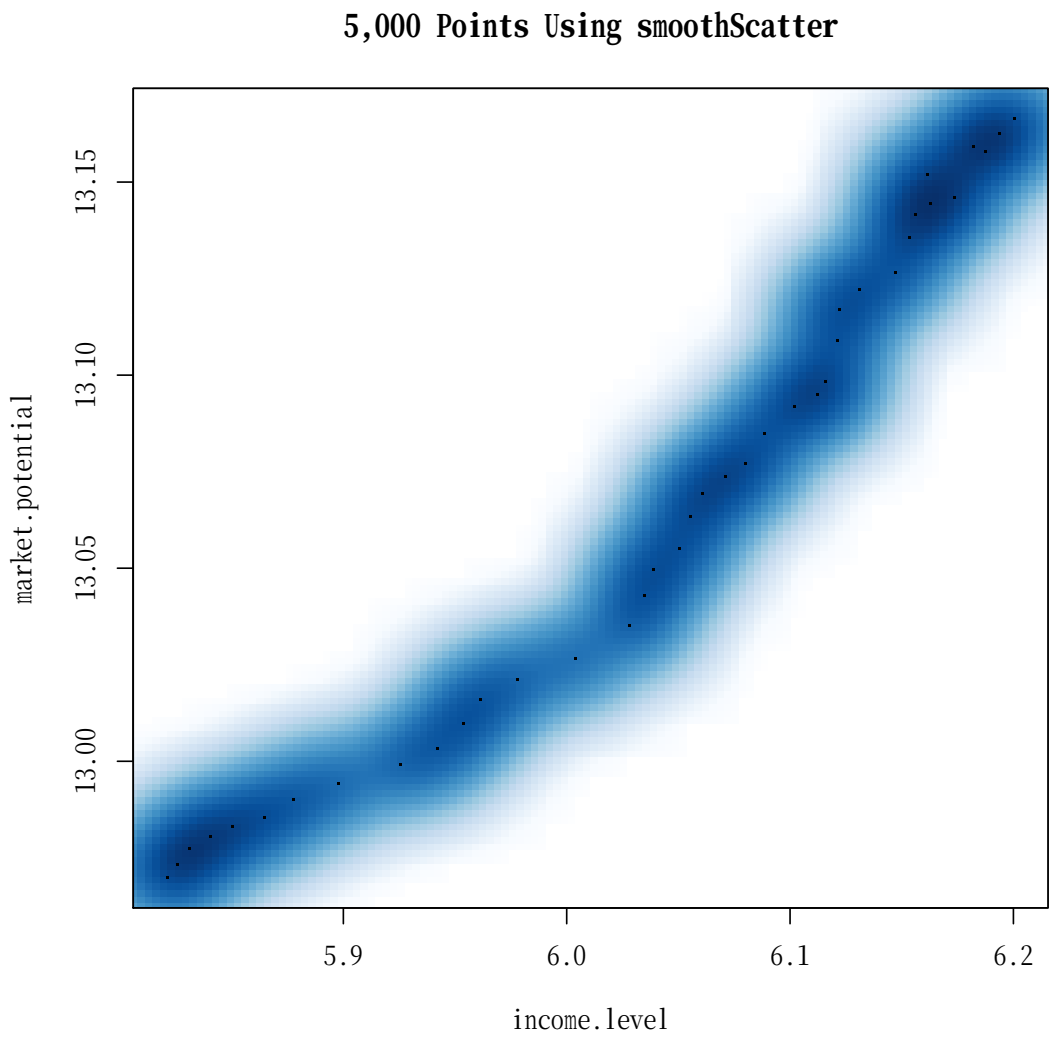
```
data(freeny)
names(freeny)

## [1] "y"                    "lag.quarterly.revenue" "price.index"
## [4] "income.level"         "market.potential"

attach(freeny)
plot(income.level, market.potential)
```



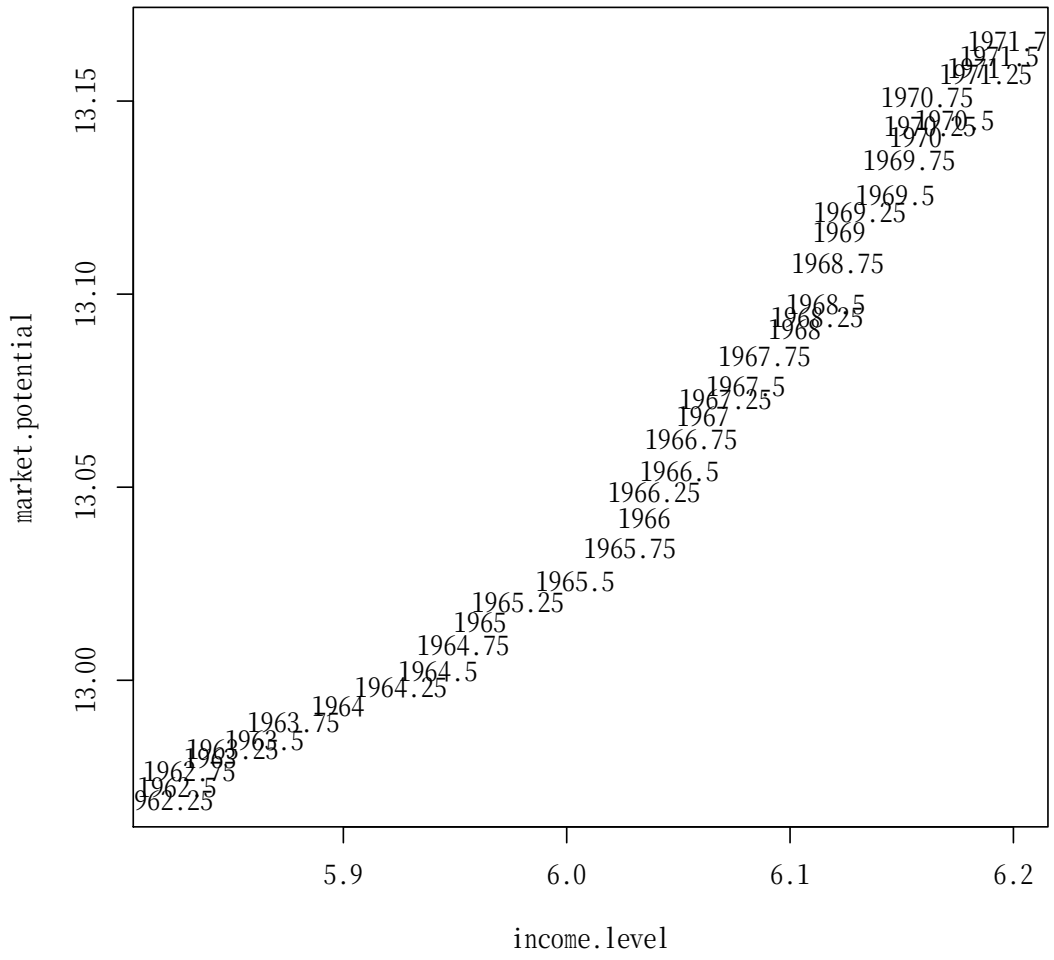

```
plot(income.level, market.potential, col="red")
```




```
smoothScatter(income.level, market.potential,
              main="5,000 Points Using smoothScatter")
```




```
plot(income.level, market.potential, type = "n" )
text(income.level, market.potential, label = row.names(freeny) )
```

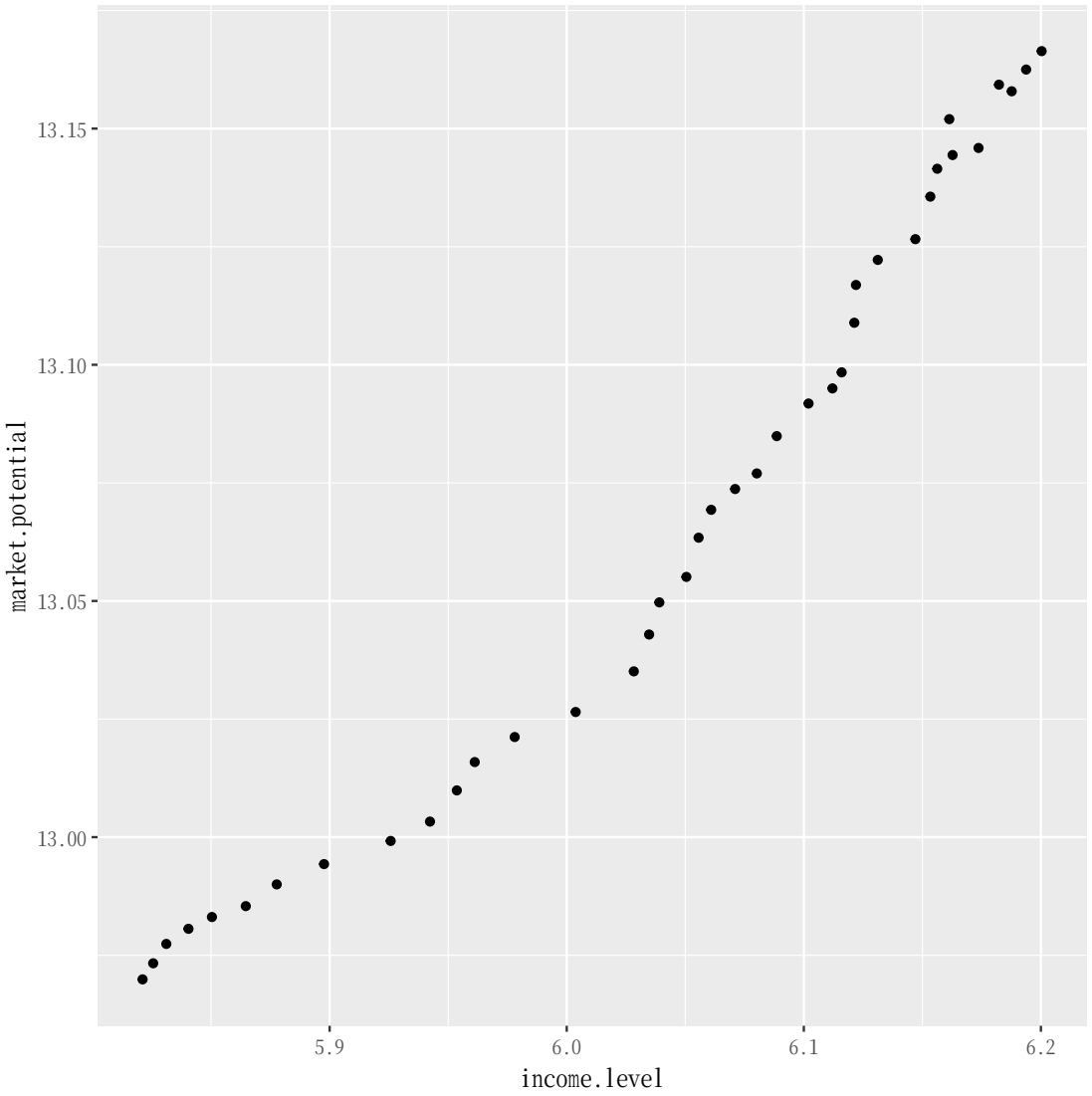



```
stem(income.level) # stem & leaf plots

##
##   The decimal point is 1 digit(s) to the left of the |
##
##   58 | 2334
##   58 | 568
##   59 | 034
##   59 | 568
##   60 | 0334
##   60 | 566789
```

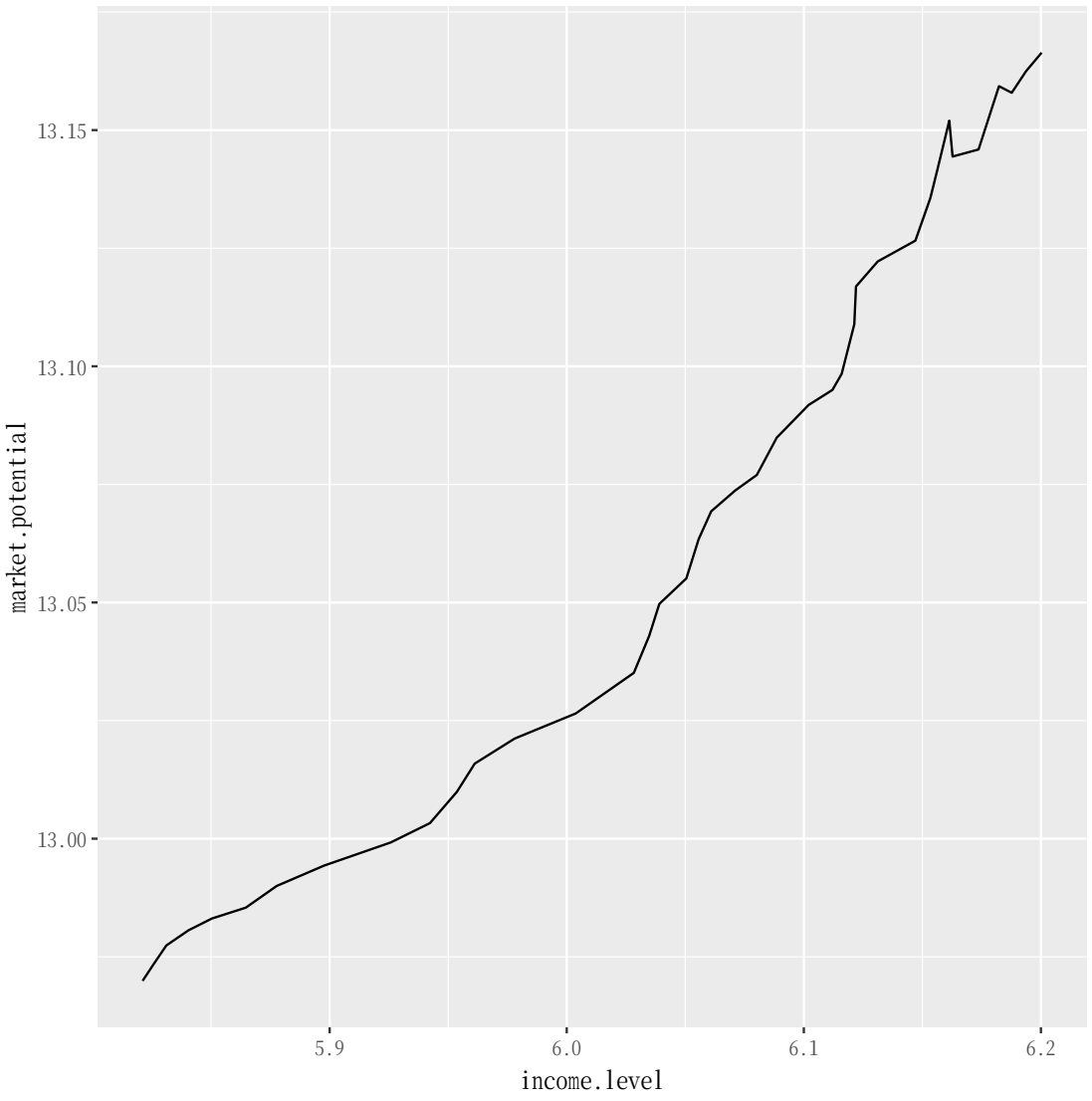
```
##   61 | 012223
##   61 | 556667899
##   62 | 0
```

Let's produce plots by using ggplot2

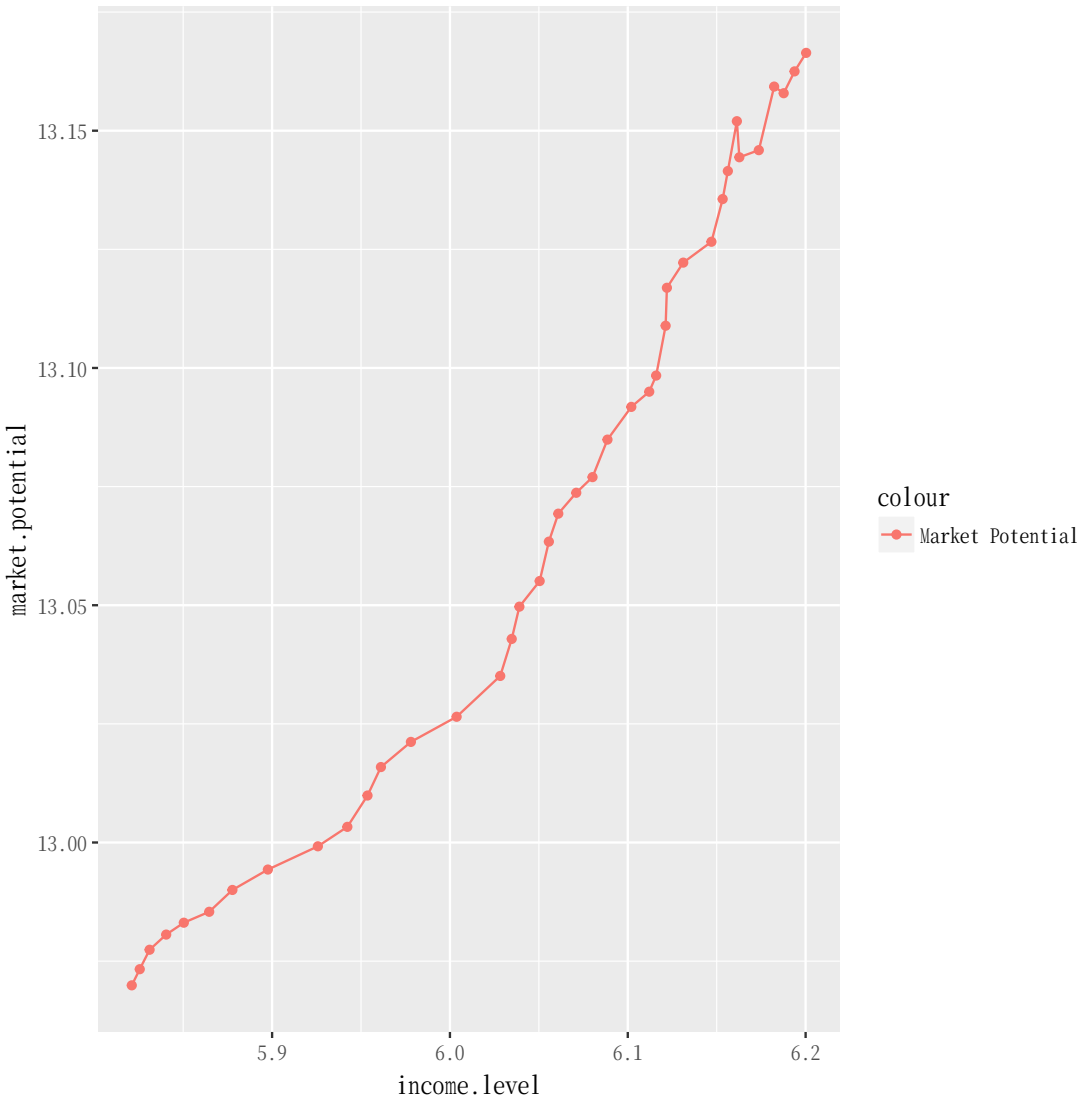
```
qplot(income.level, market.potential)
```



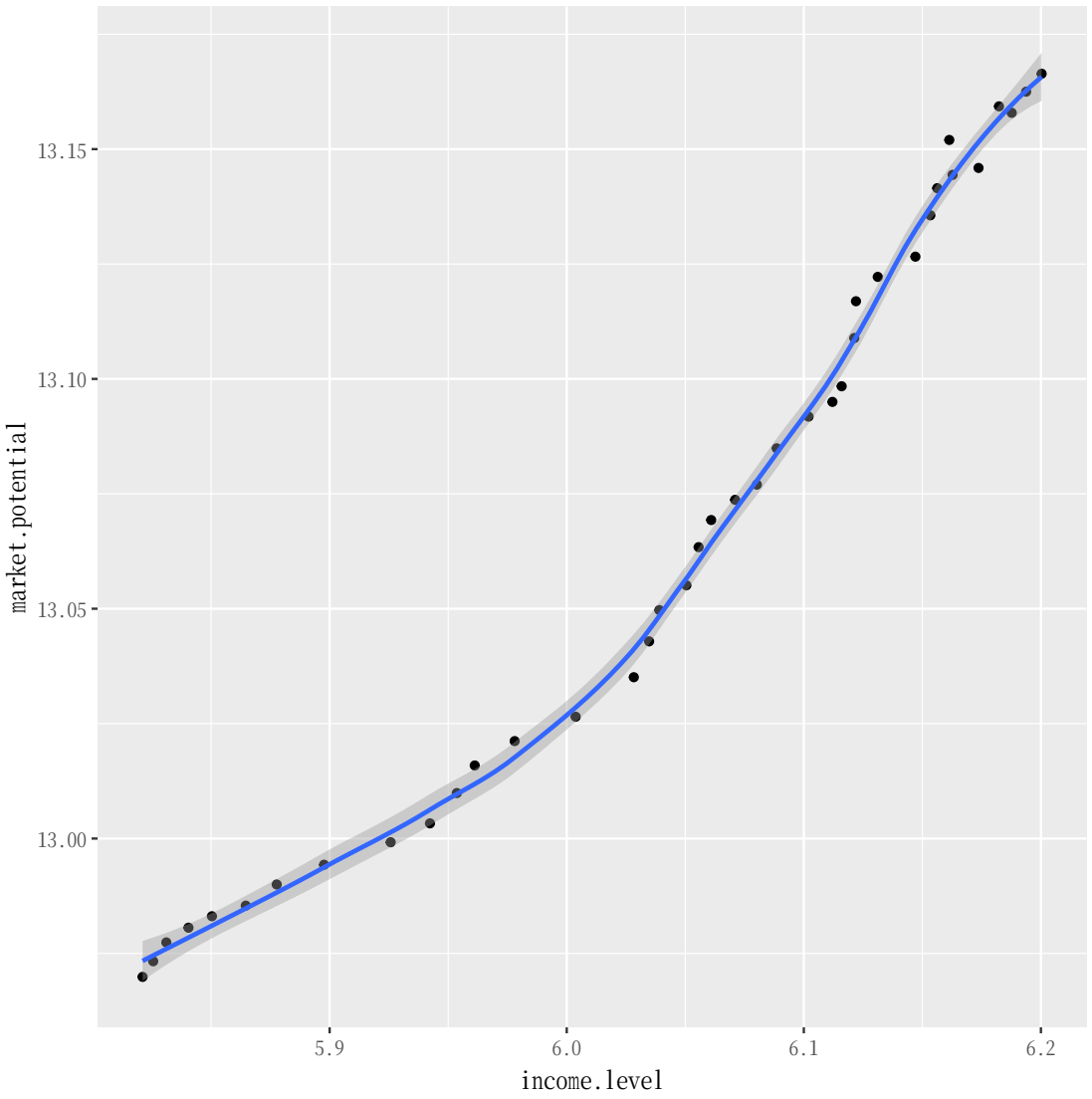

```
qplot(income.level, market.potential, geom="line")
```



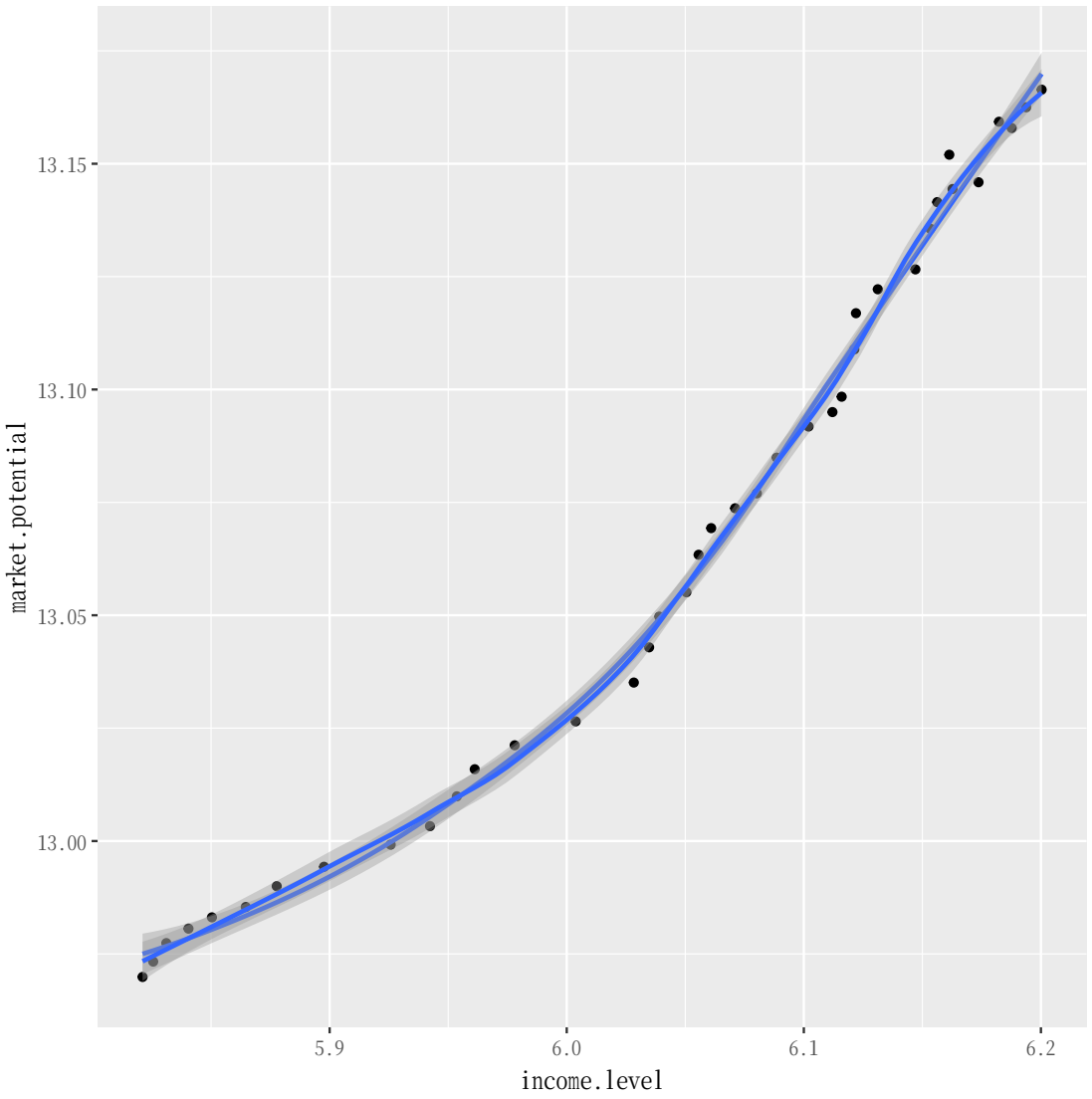

```
qplot(income.level, market.potential, geom=c("line", "point"),
      col="Market Potential")
```



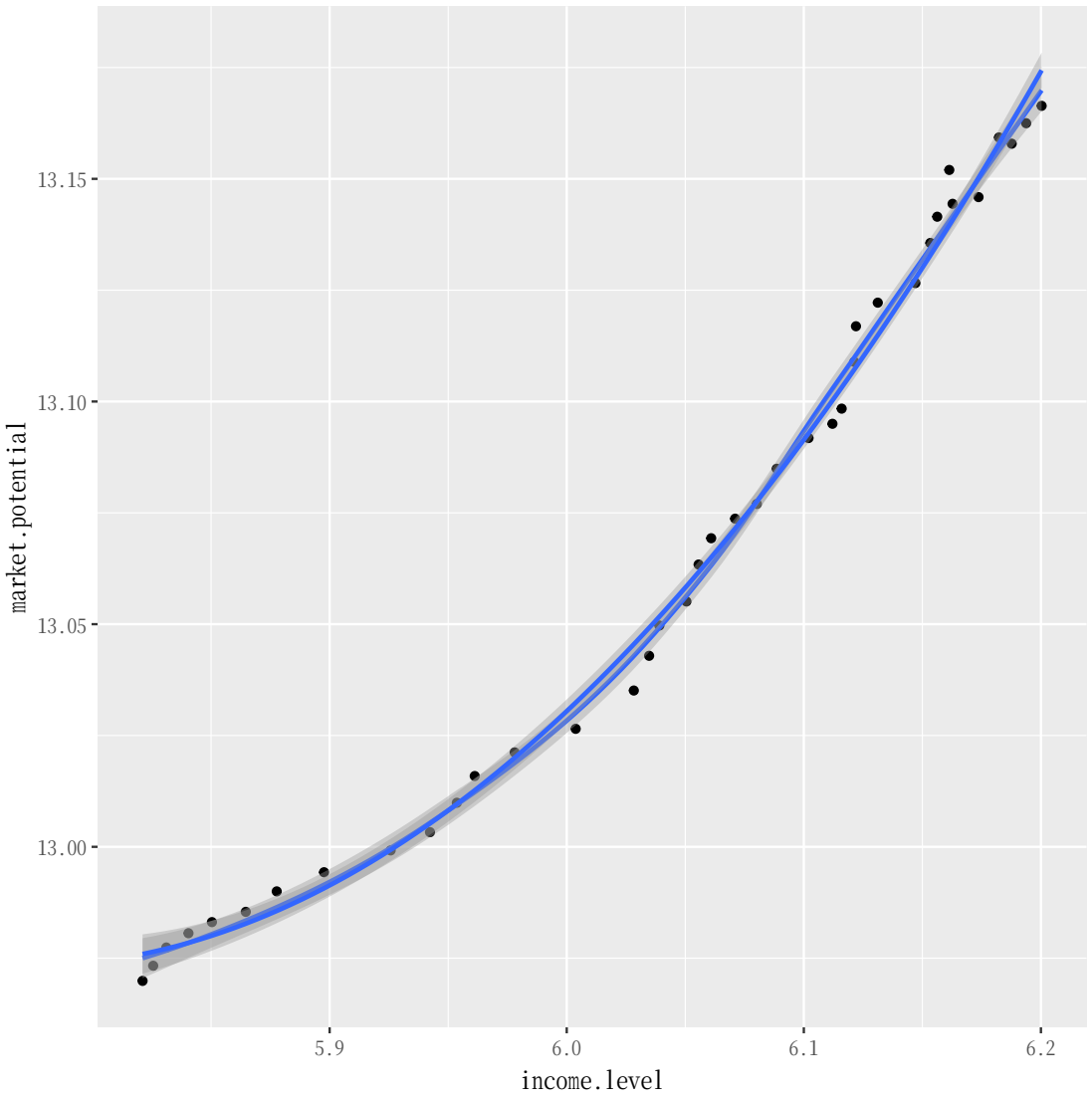

```
qplot(income.level, market.potential, geom=c("point", "smooth"), span=0.5)

## Warning:  Ignoring unknown parameters:  span
## `geom_smooth()` using method = 'loess'
```



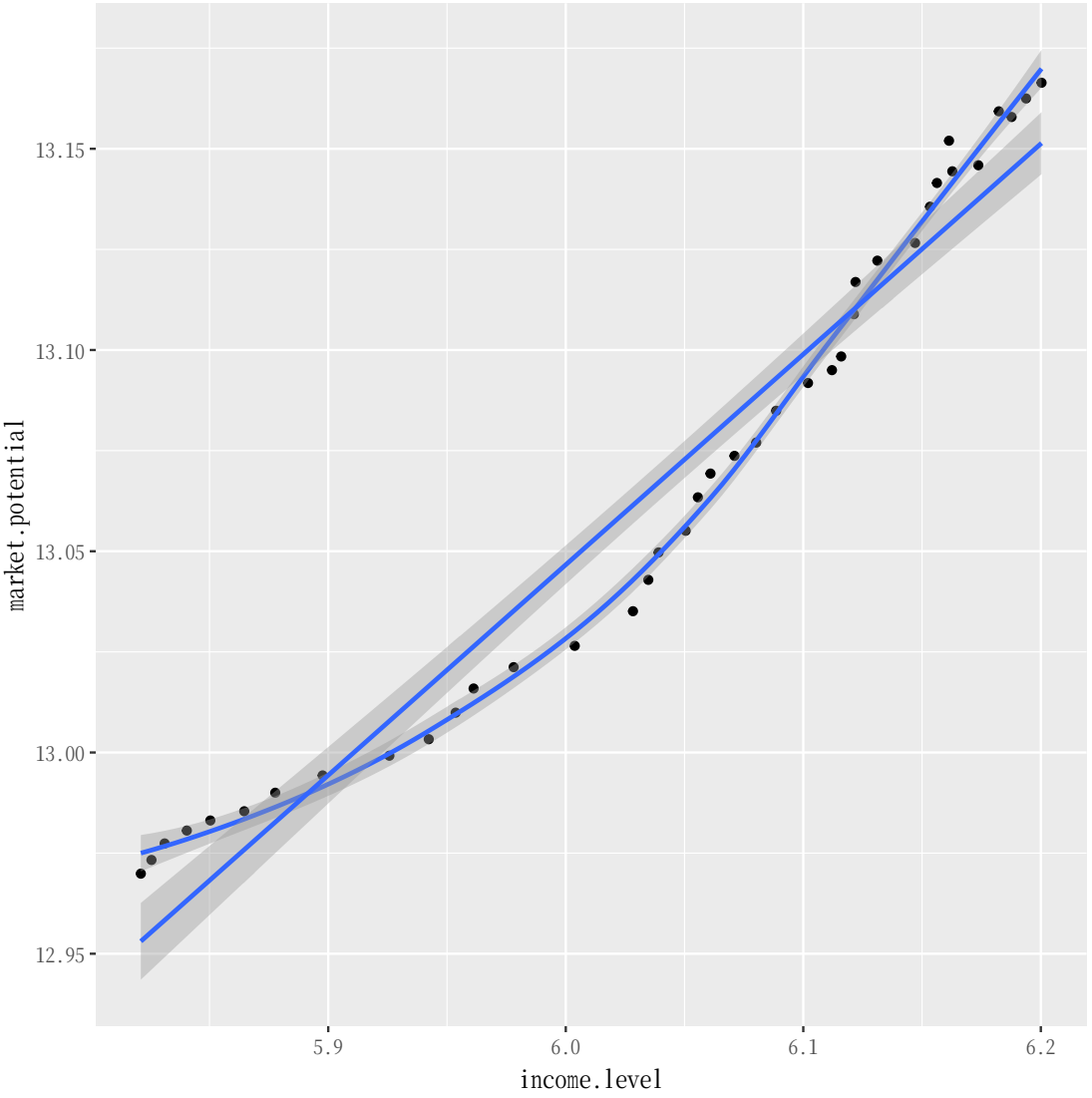

```
qplot(income.level, market.potential, geom=c("point", "smooth")) + stat_smooth(spar

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




```
qplot(income.level, market.potential, geom=c("point", "smooth")) + stat_smooth(spar

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

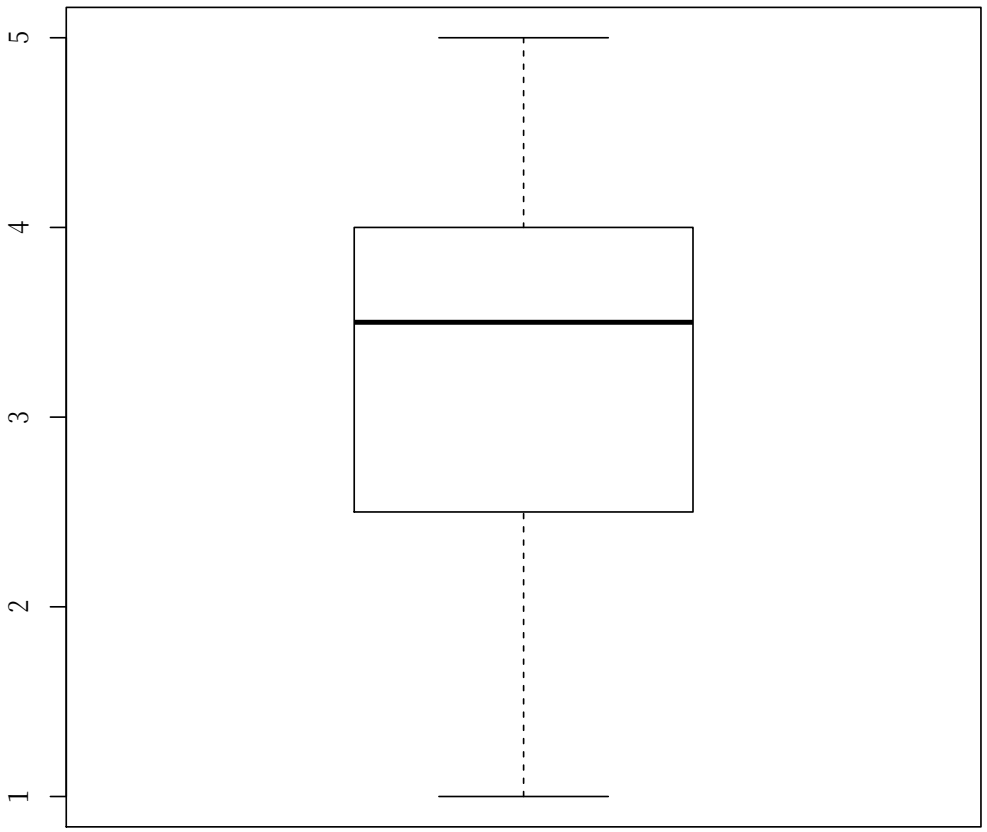



```
qplot(income.level, market.potential, geom=c("point", "smooth")) + stat_smooth(meth

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



2.2.5 box plot

```
boxplot(bk$evlauation)
```



2.3 Statistics

2.3.1 Measures of Central Tendency

```
mean(bk2$age)

## [1] 38.231

median(bk2$age)
```

```
## [1] 39

mode(bk2$age)

## [1] "numeric"
```

calculating Mode

```
bk.mode <- table(as.vector(bk2$age))
bk.mode

##
## 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
## 15 15 19 25 19 25 31 18 12 10 15  8 31 23 20 13 17 16 21 14 18 26 30 47 36
## 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
## 14 11 19 20 28 39 27 22 38 42 26 22 36 22 27 18 13 18 21 13

names(bk.mode)[bk.mode==max(bk.mode)]

## [1] "38"
```

Bk's function

```
bk.mode <- function(x){
  a=table(as.vector(x))
  names(a)[a==max(a)]
}

bk.mode(bk2$age)

## [1] "38"
```

Calculating Mode by using package, "*prettyR*"

```
require(prettyR)

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, :  there is no package called 'prettyR'

Mode(bk2$age, na.rm=T)

## Error in eval(expr, envir, enclos):  could not find function "Mode"
```

Calculating IQR

```
sort(bk2$age)
```

```
range(bk2$age)

## [1] 15 59

summary(bk2$age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   28.00   39.00   38.23   49.00   59.00

q <- quantile(bk2$age)
q

##   0%  25%  50%  75% 100%
##   15   28   39   49   59

q[4]-q[2]

## 75%
##  21

IQR(bk2$age)

## [1] 21
```

2.3.2 Measures of Variation

```
sd(bk2$age)

## [1] 12.41917

var(bk2$age)

## [1] 154.2359
```

Using *My Stat* function

```
mystats=function(x) {result=c(length(x),mean(x, na.rm=T),sd(x, na.rm=T),
                              var(x, na.rm=T), median(x, na.rm=T),
                              IQR(x, na.rm=T))
                     names(result)=c("n","Mean","SD","Var","Median","IQR")
                     result}

mystats(bk2$age)

##           n        Mean          SD         Var      Median         IQR
## 1000.00000    38.23100    12.41917  154.23587    39.00000    21.00000
```

Another funtion of mystats

```
mystats<-function(x)
{
  m = mean(x, na.rm=T)
  v = var(x, na.rm=T)
  s = sqrt(v)
  M = median(x, na.rm=T)
  return(list(mean=m, variance=v, sd=s, Median=m))
}
mystats(bk2$age)

## $mean
## [1] 38.231
##
## $variance
## [1] 154.2359
##
## $sd
## [1] 12.41917
##
## $Median
## [1] 38.231
```

Again, another function of mystats

```
mystats<-function(x) {print (mean(x, na.rm=T))
                      print (sd(x, na.rm=T))
                      print (median(x, na.rm=T))
                      print (length(x))
```

```
                    print (IQR(x, na.rm=T))
                    print (var(x, na.rm=T))
}
mystats(bk2$age)

## [1] 38.231
## [1] 12.41917
## [1] 39
## [1] 1000
## [1] 21
## [1] 154.2359
```

Detecting outliers by using "*outliers*" package

```
require(outliers)

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, :  there is no package called 'outliers'

outlier(bk2$age)

## Error in eval(expr, envir, enclos):  could not find function "outlier"

outlier.age <- subset(bk2, age <= 15, select=-age)
##Selecting the cases of outliers
dim(outlier.age)

## [1]  15 114
```

2.3.3 Basic descriptive statistics

```
require(pastecs)

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, :  there is no package called 'pastecs'

stat.desc(bk2$age)

## Error in eval(expr, envir, enclos):  could not find function "stat.desc"
```

```
options(scipen=100)
options(digits=2)
stat.desc(bk2$age)

## Error in eval(expr, envir, enclos):  could not find function "stat.desc"

stat.desc(bk2$age, basic=F)

## Error in eval(expr, envir, enclos):  could not find function "stat.desc"

data.frame(stat.desc(bk2$age, basic=F))

## Error in data.frame(stat.desc(bk2$age, basic = F)): could not find
function "stat.desc"
```

Calculating skewness and kurtosis by using "*e1071*" package

```
library(e1071)
skewness(bk2$age)

## [1] -0.23

kurtosis(bk2$age)

## [1] -1.1
```

Conducting descriptive statistics by using "*psych*" package

```
require(psych)
describe(bk)

##            vars  n mean  sd median trimmed  mad min max range skew
## evlauation    1 20  3.2 1.1    3.5     3.3 0.74   1   5     4 -0.6
##            kurtosis   se
## evlauation    -0.74 0.25

describe(bk2$age)

##    vars    n mean sd median trimmed mad min max range  skew kurtosis   se
## X1    1 1000   38 12     39      39  15  15  59    44 -0.23     -1.1 0.39

describeBy(bk2$age, bk2$gender) ##Summary Statistics by Group ##
```

```
## group: male
##    vars   n mean sd median trimmed mad min max range  skew kurtosis   se
## X1    1 513   38 13     39      38  15  15  59    44 -0.22     -1.1 0.55
## ------------------------------------------------------------
## group: female
##    vars   n mean sd median trimmed mad min max range  skew kurtosis   se
## X1    1 487   38 12     39      39  15  15  59    44 -0.24     -1.1 0.56
```

```
skew(bk2$age, na.rm = TRUE)
```

```
## [1] -0.23
```

```
kurtosi(bk2$age, na.rm = TRUE)
```

```
## [1] -1.1
```

GOOD NIGHT AND GOOD LUCK !!