Introduction to R

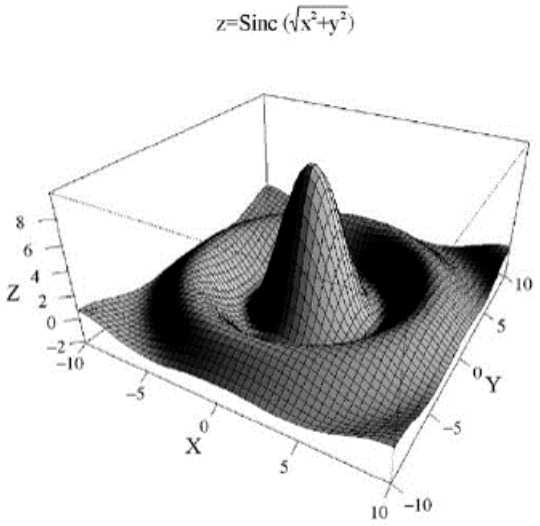
Jing Bu

10/9/2018

目录

1 Getting Started With R

- Not only a statistical programming language, but a computing environment for statistical computing and graphics.
- Powerful Programming and Extending Capability
- Multiple Platforms
- Very excellent graphics
- A big but not a determinate advantage: FREE Open Source



1.1 Installing R

- The first thing you have to do to use R is to download it from here:R
- Choose the nearest mirror in China

1. Tsinghua https://mirrors.tuna.tsinghua.edu.cn/CRAN/
2. USTC https://mirrors.ustc.edu.cn/CRAN/
3. LanZhou https://mirror.lzu.edu.cn/CRAN/
4. Xiamen http://mirrors.xmu.edu.cn/CRAN/

1.2 Using IDE: RStudio

- The most popular IDE for R
- Also Free(for basic version)
- Combine with `Markdown` and `Latex` to make scientific writings or presentation easier
- Download it from here: RStudio

1.3 Using R as Stata: Packages

- Many researchers provide their own R programs through the R project webpage.
- Many packages are already preinstalled in the basic R installation.
- They can be directly activated from RStudio.
- Or they are activated by issuing a command in the Console.

```
#install.packages("foreign",repos = "http://mirrors.ustc.edu.cn/CRAN/")
```

1.4 Where to get help

- The online help in R describes all basic R commands as well as commands in active packages.
- search the online help from the Help pane in RStudio.
- Alternatively, using the command

```
?load
```

```
## starting httpd help server ... done
```

```
# or
help("load")
# or
??load
# or
help.search("read")
```

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

2 Basic data Management in R

2.1 Opening and Saving Data: Working directory

- R will look for data or save data in the drive and working directory.
- The working directory is specified depending on the operation system

```
getwd()
```

```
## [1] "C:/Users/admin/Desktop/teaching assiatant/Econometrics/teaching assistant slide
```

2.2 Changing the working directory

```
setwd("/Users/admin/Desktop/teaching assiatant/Econometrics/teaching assistant slides/R
getwd()
```

```
## [1] "C:/Users/admin/Desktop/teaching assiatant/Econometrics/teaching assistant slide
```

2.3 Importing Data: From STATA

- R will look for data or save data in the drive and working directory.
- The working directory is specified depending on the operation system
- imports data from STATA

  (version<=12):

```
library("foreign")
caschool <- read.dta("/Users/admin/Desktop/teaching assiatant/Econometrics/teaching ass
cars_data <- read.dta("/Users/admin/Desktop/teaching assiatant/Econometrics/teaching as
```

2.4 Importing Data: From CSV

```
caschool_csv <- read.csv("/Users/admin/Desktop/teaching assiatant/Econometrics/teaching
```

2.5 Summary the Data

```
summary(cars_data)
```

```
##     observat        dist_cod        county            district
##  Min.   :  1.0   Min.   :61382   Length:420         Length:420
##  1st Qu.:105.8   1st Qu.:64308   Class :character   Class :character
##  Median :210.5   Median :67761   Mode  :character   Mode  :character
##  Mean   :210.5   Mean   :67473
##  3rd Qu.:315.2   3rd Qu.:70419
##  Max.   :420.0   Max.   :75440
##    gr_span             enrl_tot          teachers          calw_pct
##  Length:420         Min.   :   81.0   Min.   :   4.85   Min.   : 0.000
```

```
##  Class :character   1st Qu.:  379.0   1st Qu.:  19.66   1st Qu.: 4.395
##  Mode  :character   Median :  950.5   Median :  48.56   Median :10.520
##                     Mean   : 2628.8   Mean   : 129.07   Mean   :13.246
##                     3rd Qu.: 3008.0   3rd Qu.: 146.35   3rd Qu.:18.981
##                     Max.   :27176.0   Max.   :1429.00   Max.   :78.994
##     meal_pct         computer         testscr         comp_stu
##  Min.   :  0.00   Min.   :   0.0   Min.   :605.5   Min.   :0.00000
##  1st Qu.: 23.28   1st Qu.:  46.0   1st Qu.:640.0   1st Qu.:0.09377
##  Median : 41.75   Median : 117.5   Median :654.5   Median :0.12546
##  Mean   : 44.71   Mean   : 303.4   Mean   :654.2   Mean   :0.13593
##  3rd Qu.: 66.86   3rd Qu.: 375.2   3rd Qu.:666.7   3rd Qu.:0.16447
##  Max.   :100.00   Max.   :3324.0   Max.   :706.8   Max.   :0.42083
##     expn_stu         str            avginc           el_pct
##  Min.   :3926   Min.   :14.00   Min.   : 5.335   Min.   : 0.000
##  1st Qu.:4906   1st Qu.:18.58   1st Qu.:10.639   1st Qu.: 1.941
##  Median :5215   Median :19.72   Median :13.728   Median : 8.778
##  Mean   :5312   Mean   :19.64   Mean   :15.317   Mean   :15.768
##  3rd Qu.:5601   3rd Qu.:20.87   3rd Qu.:17.629   3rd Qu.:22.970
##  Max.   :7712   Max.   :25.80   Max.   :55.328   Max.   :85.540
##     read_scr        math_scr
##  Min.   :604.5   Min.   :605.4
##  1st Qu.:640.4   1st Qu.:639.4
##  Median :655.8   Median :652.5
##  Mean   :655.0   Mean   :653.3
##  3rd Qu.:668.7   3rd Qu.:665.9
##  Max.   :704.0   Max.   :709.5
```

2.6 Variables

```
#install.packages("dplyr", repos = "http://mirrors.ustc.edu.cn/CRAN/")
names(cars_data)
```

```
##  [1] "observat" "dist_cod" "county"   "district" "gr_span"  "enrl_tot"
##  [7] "teachers" "calw_pct" "meal_pct" "computer" "testscr"  "comp_stu"
```

```
## [13] "expn_stu" "str"      "avginc"   "el_pct"   "read_scr" "math_scr"
```

- https://www.rdocumentation.org/

2.7 Variables

```
cars_data_small <- select(cars_data,observat,testscr,str,expn_stu,el_pct)
```

2.8 Data Manipulation

- generate new variable

```
cars_data_small$logexp <- log(cars_data$expn_stu)
cars_data_small$el_high <- cars_data$el_pct
head(cars_data_small)
```

```
##   observat testscr      str expn_stu    el_pct   logexp   el_high
## 1        1  690.80 17.88991 6384.911  0.000000 8.761693  0.000000
## 2        2  661.20 21.52466 5099.381  4.583333 8.536874  4.583333
## 3        3  643.60 18.69723 5501.955 30.000002 8.612859 30.000002
## 4        4  647.70 17.35714 7101.831  0.000000 8.868108  0.000000
## 5        5  640.85 18.67133 5235.988 13.857677 8.563311 13.857677
## 6        6  605.55 21.40625 5580.147 12.408759 8.626970 12.408759
```

2.9 Descriptive Statistics

- summary a variable

```
summary(cars_data_small$testscr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   605.5   640.0   654.5   654.2   666.7   706.8
```

- if the dataframe is attached, simply

```
attach(cars_data_small)
summary(testscr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   605.5   640.0   654.5   654.2   666.7   706.8
```

```
detach(cars_data_small)
```

3 R Markdown

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

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

3.1 What is markdown?

Markdown is a markup language that can be written using a plain text editor. It allows plain text content to be formatted with simple markup syntax.

3.2 Why should we study markdown?

- Markdown's syntax is easy to learn, and its feature is more powerful than plain text.
- It is easy to turn markdown into ppt,pdf and a word document.

3.3 how to use markdown

3.3.1 title

H1 :# Header 1

H2 :## Header 2
H3 :### Header 3
H4 :#### Header 4
H5 :##### Header 5
H6 :###### Header 6

3.3.2 list

- document1
- document2
- document3

3.3.3 ordered list

1. documnet1
2. document2
3. document3

3.3.4 insert link

```
[nba](http://www.nba.com)
```

3.3.5 insert picture

```
![picture](C:\Users\admin\Desktop\teaching assiatant\Econometrics\teaching assistant slides\R\picture3)
```

3.3.6 write formula

$$F=ma$$

3.3.7 cite

> how to use markdown

3.3.8 Italic and bold

```
*Italic*
**bold**
```

Italic
bold

3.3.9 code

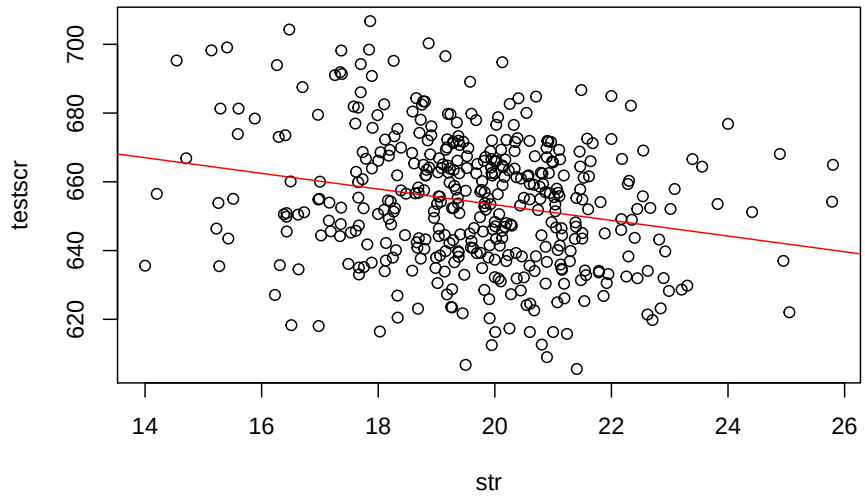
````
```{r reg, echo=TRUE}
fm1 <- lm(testscr ~ str,data = cars_data_small)
summary(fm1)
```
````

4 Plot

4.1 Scatter Plot

- Draw a scatter plot of the variable testscr against str:

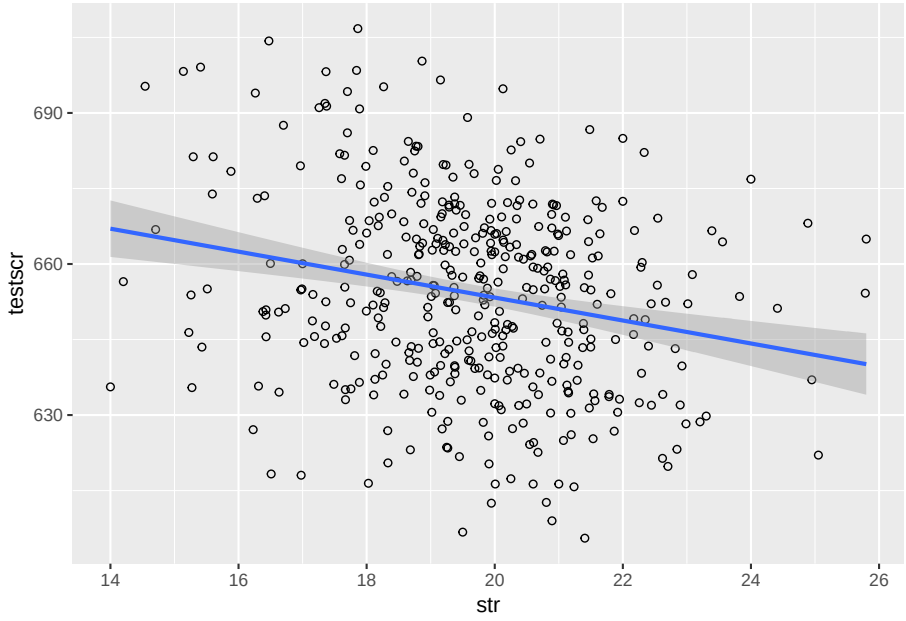
```
attach(cars_data_small)
plot(str, testscr)
abline(lm(testscr ~ str , data = cars_data_small),col = "red")
```




```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
   singular.ok = TRUE, contrasts = NULL, offset, ···)
```

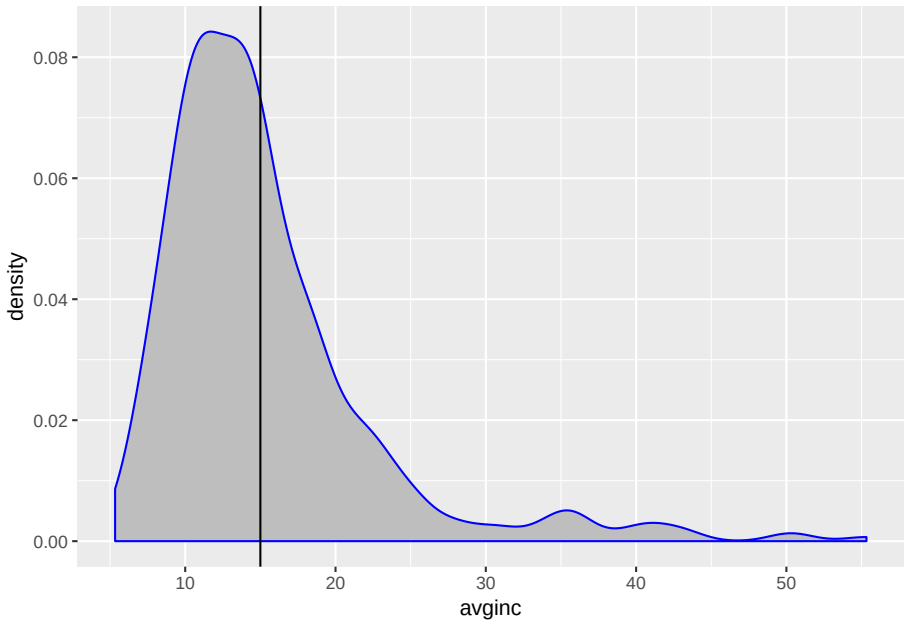
4.2 ggplot2

```
library("ggplot2")
ggplot(data =cars_data_small,aes(x=str, y=testscr)) +
geom_point(shape=1) + # Use hollow circles
geom_smooth(method=lm) # Add linear regression line
```

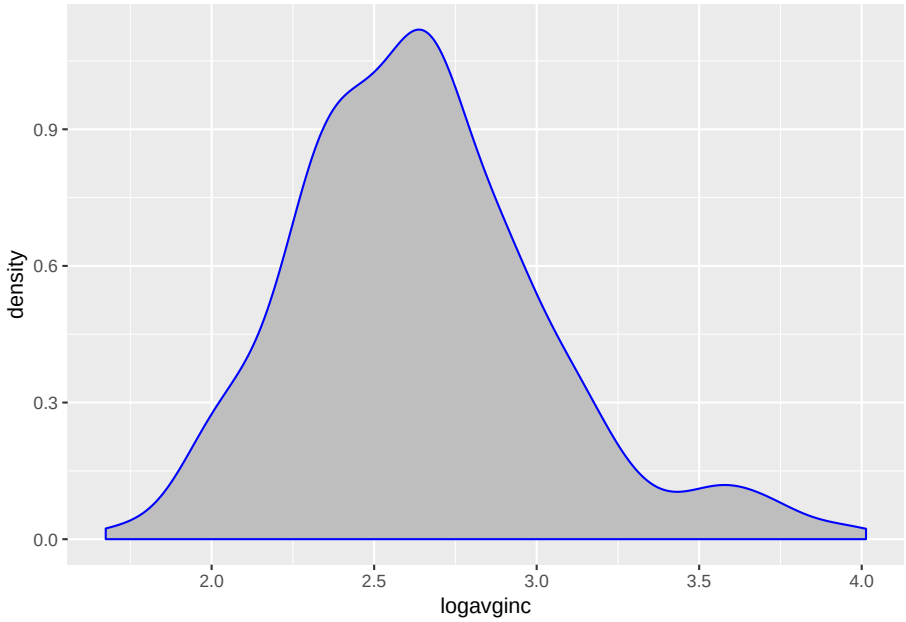


4.3 A kdensity distribution of income

```
cars_data$inc <- with(cars_data,avginc >=15)
ggplot(cars_data,aes(x=avginc))+
geom_density(fill="grey",color ="blue")+
geom_vline(xintercept = 15)
```

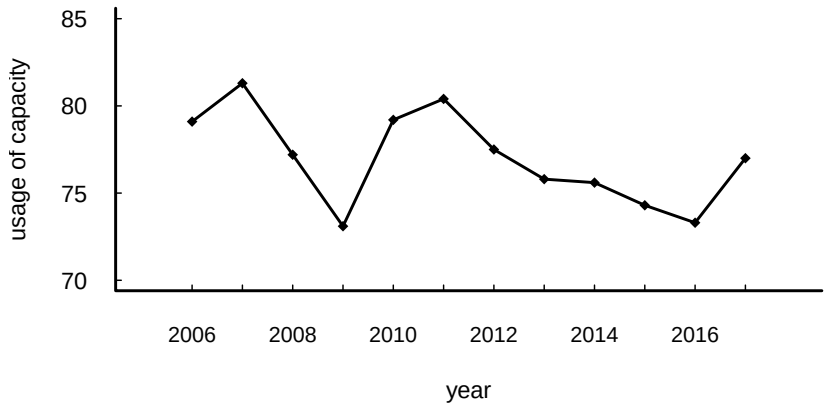



```
cars_data$logavginc <- log(cars_data$avginc)
ggplot(cars_data,aes(x=logavginc))+
geom_density(fill="grey",color ="blue")
```

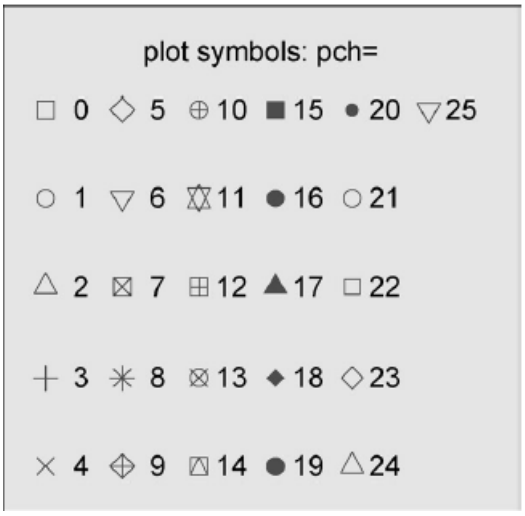


4.4 extra image

```
library(readxl)
image <- read_excel("/Users/admin/Desktop/teaching assiatant/Econometrics/teaching assi
attach(image)
opar<-par(no.readonly = TRUE)
par(pch=18,lwd=2)
par(cex=1,cex.axis=1,cex.lab=1)
par(font=1,font.axis=1,font.lab=1)
par(pin=c(5,2))
plot(year,rate,type="o",bty="l",ann=FALSE,xaxt="n",yaxt="n",xlim=c(2005,2018),ylim=c(70
title(xlab="year",ylab="usage of capacity")
axis(1,at=year,tck=0.02,cex.axis=0.95,las=0)
axis(2,tck=0.02,las=2,cex.axis=1)
```



```
par(opar)
```



5 OLS Regression

```
fm1 <- lm(testscr ~ str,data = cars_data_small)
summary(fm1)
```

```
##
## Call:
## lm(formula = testscr ~ str, data = cars_data_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.727 -14.251   0.483  12.822  48.540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 698.9330     9.4675  73.825  < 2e-16 ***
## str          -2.2798     0.4798  -4.751 2.78e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.58 on 418 degrees of freedom
## Multiple R-squared:  0.05124,    Adjusted R-squared:  0.04897
## F-statistic: 22.58 on 1 and 418 DF,  p-value: 2.783e-06
```

5.1 OLS Regression 2

```
fm2 <- lm(testscr ~ str,data = cars_data)

summary(fm2)
```

```
##
## Call:
## lm(formula = testscr ~ str, data = cars_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.727 -14.251   0.483  12.822  48.540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 698.9330     9.4675  73.825  < 2e-16 ***
## str          -2.2798     0.4798  -4.751 2.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.58 on 418 degrees of freedom
## Multiple R-squared:  0.05124,    Adjusted R-squared:  0.04897
## F-statistic: 22.58 on 1 and 418 DF,  p-value: 2.783e-06
```

6 T-test in R

6.1 single sample

- t-test for scores

```
summary(cars_data_small$testscr)
```

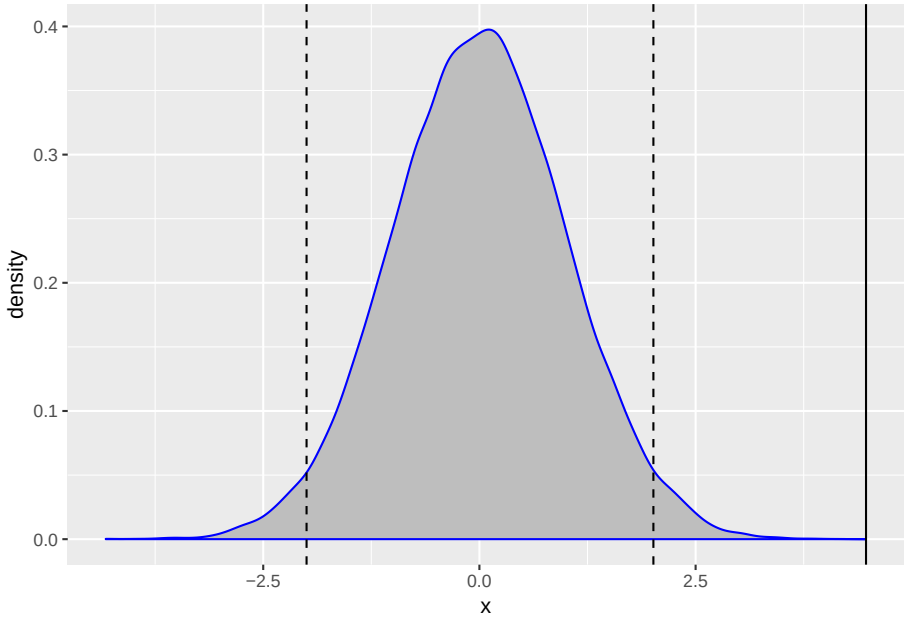
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   605.5   640.0   654.5   654.2   666.7   706.8
```

```
t.test(cars_data_small$testscr,alternative = "two.sided",mu = 650)
```

```
##
##  One Sample t-test
##
## data:  cars_data_small$testscr
## t = 4.4708, df = 419, p-value = 1.005e-05
## alternative hypothesis: true mean is not equal to 650
## 95 percent confidence interval:
##  652.3291 655.9840
## sample estimates:
## mean of x
##  654.1565
```

- Construct t-Statistics

```
randT <- rt(30000,df=NROW(testscr)-1) # build a distribution
scoreTtest <- t.test(cars_data_small$testscr,alternative = "two.sided",mu = 650)
ggplot(data.frame(x=randT)) +
geom_density(aes(x=x),fill = "grey",color ="blue") +
geom_vline(xintercept = scoreTtest$statistic) +
geom_vline(xintercept = mean(randT) + c(-2,2)*sd(randT),linetype = 2)
```




```
#attach(cars_data_small)
#t.test(testscr~el_high,data = cars_data_small)
```