Introduction to R

Jing Bu

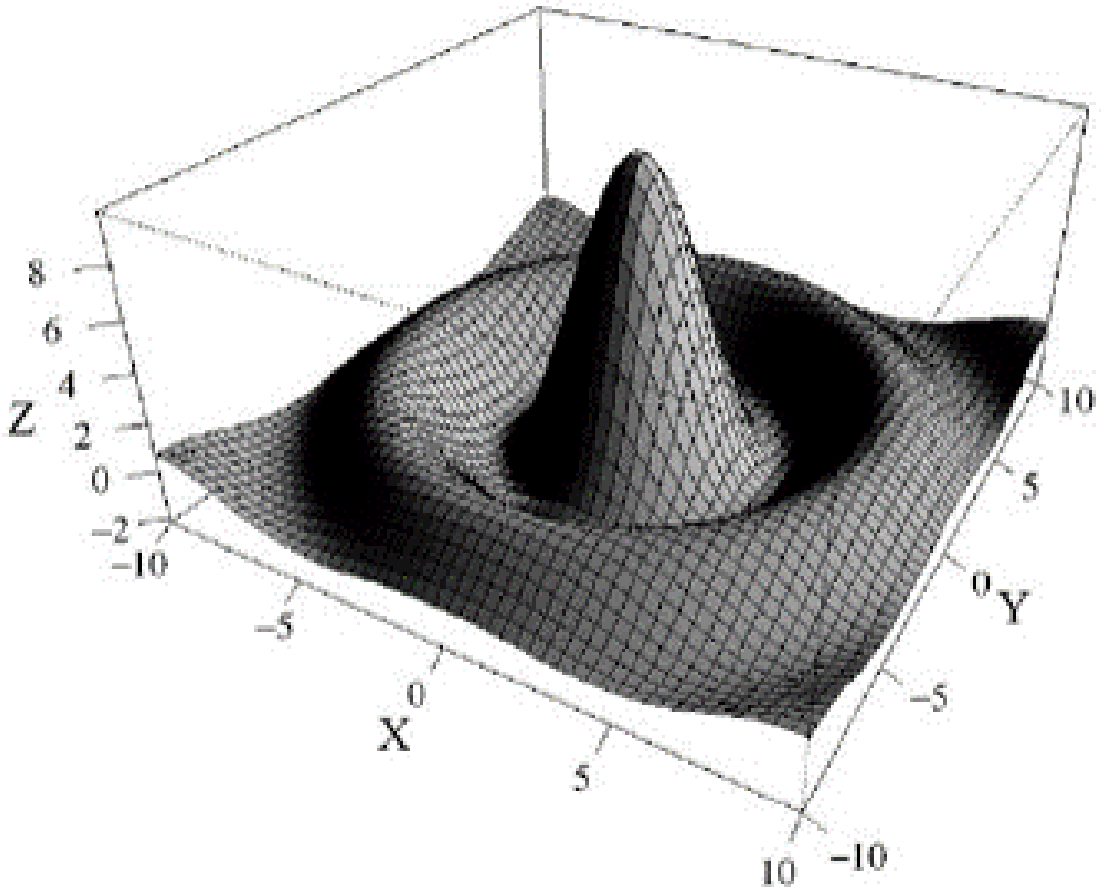
Business school,Nanjing University

10/9/2018

Getting Stared 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

$z=\mathrm{Sinc}\left(\sqrt{x^2+y^2}\right)$



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/

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

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.xmu.edu.cn
```

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 = "", ...)
```

Basic data Management in R

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
```

Changing the working directory

```
setwd("/Users/admin/Desktop/teaching assiatant/Econometrics/te
getwd()
```

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

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("caschool.dta")
cars_data <- read.dta("/Users/admin/Desktop/teaching assiatant
```

Importing Data: From CSV

```
caschool_csv <- read.csv("caschool.csv")
caschool_csv
head(caschool_csv)
```

Summary the Data

```
summary(cars_data)
```

```
##     observat        dist_cod      county             distr
##  Min.   :  1.0   Min.   :61382   Length:420         Length:
##  1st Qu.:105.8   1st Qu.:64308   Class :character   Class :
##  Median :210.5   Median :67761   Mode  :character   Mode  :
##  Mean   :210.5   Mean   :67473
##  3rd Qu.:315.2   3rd Qu.:70419
##  Max.   :420.0   Max.   :75440
##    gr_span            enrl_tot         teachers
##  Length:420         Min.   :   81.0   Min.   :   4.85   Min
##  Class :character   1st Qu.:  379.0   1st Qu.:  19.66   1st
##  Mode  :character   Median :  950.5   Median :  48.56   Med
##                     Mean   : 2628.8   Mean   : 129.07   Mea
##                     3rd Qu.: 3008.0   3rd Qu.: 146.35   3rd
##                     Max.   :27176.0   Max.   :1429.00   Max
##     meal_pct         computer         testscr          comp_
##  Min.   :  0.00   Min.   :   0.0   Min.   :605.5   Min.
```

Variables

```
#install.packages("dplyr")
names(cars_data)
```

```
##  [1] "observat" "dist_cod" "county"   "district" "gr_span"
##  [7] "teachers" "calw_pct" "meal_pct" "computer" "testscr"
## [13] "expn_stu" "str"      "avginc"   "el_pct"   "read_scr"
```

Variables

```
cars_data_small <- select(cars_data,observat,testscr,str,expn_
cars_data_small
```

Data Manipulation

- generate new variable

```
cars_data_small$logexp <- log(cars_data$expn_stu)
cars_data_small$el_high <- cars_data$el_pct>=50
head(cars_data_small)
```

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

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)
```

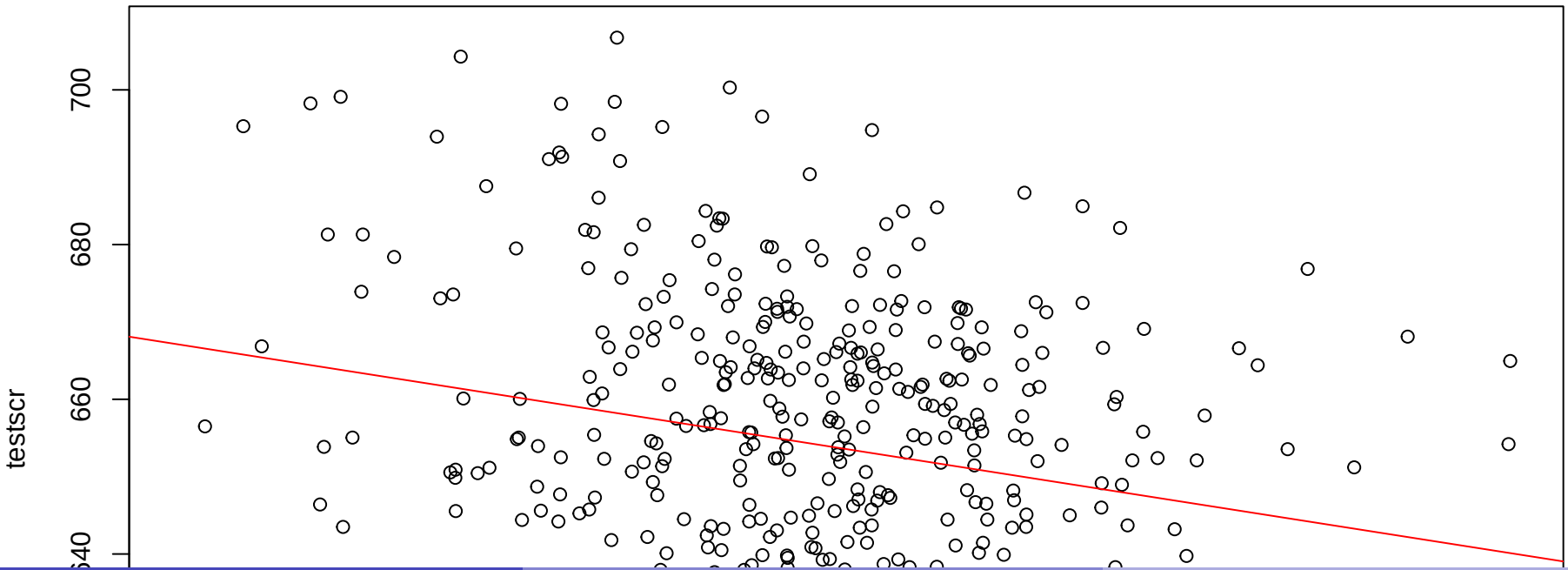
```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    605.5   640.0   654.5   654.2   666.7   706.8
```

Plot

Scatter Plot

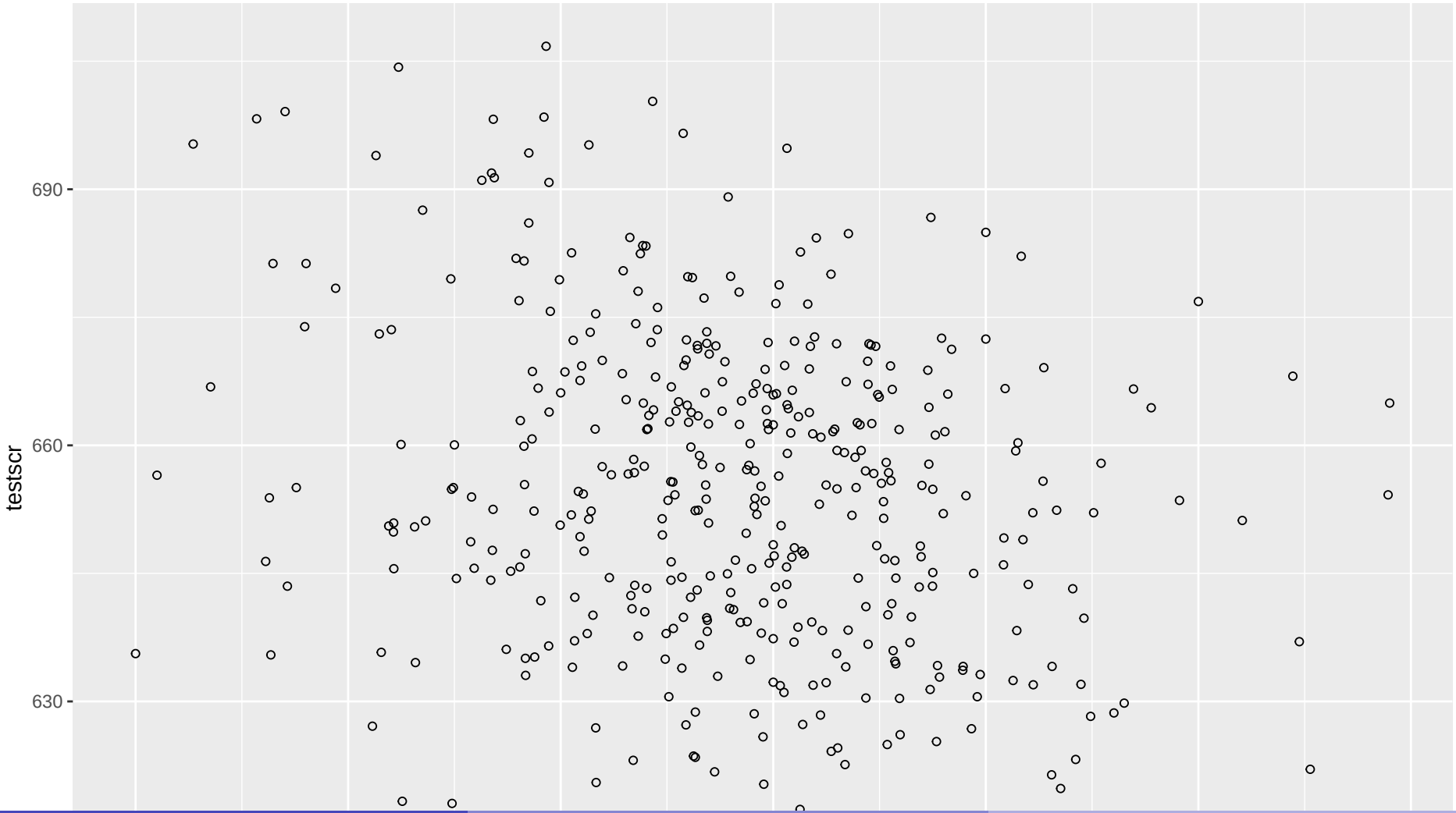
- Draw a scatter plot of the variable testsc against str:

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



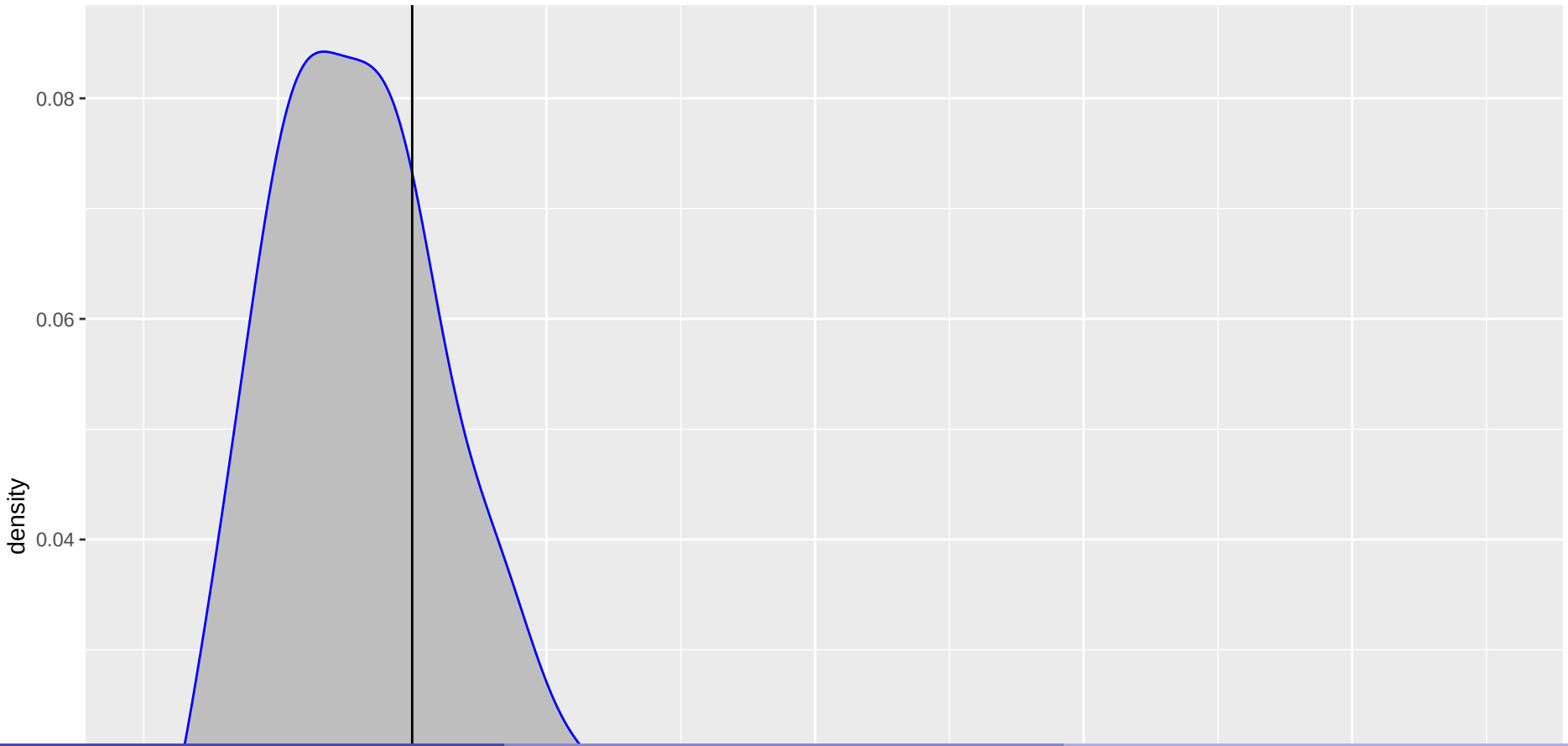
ggplot2

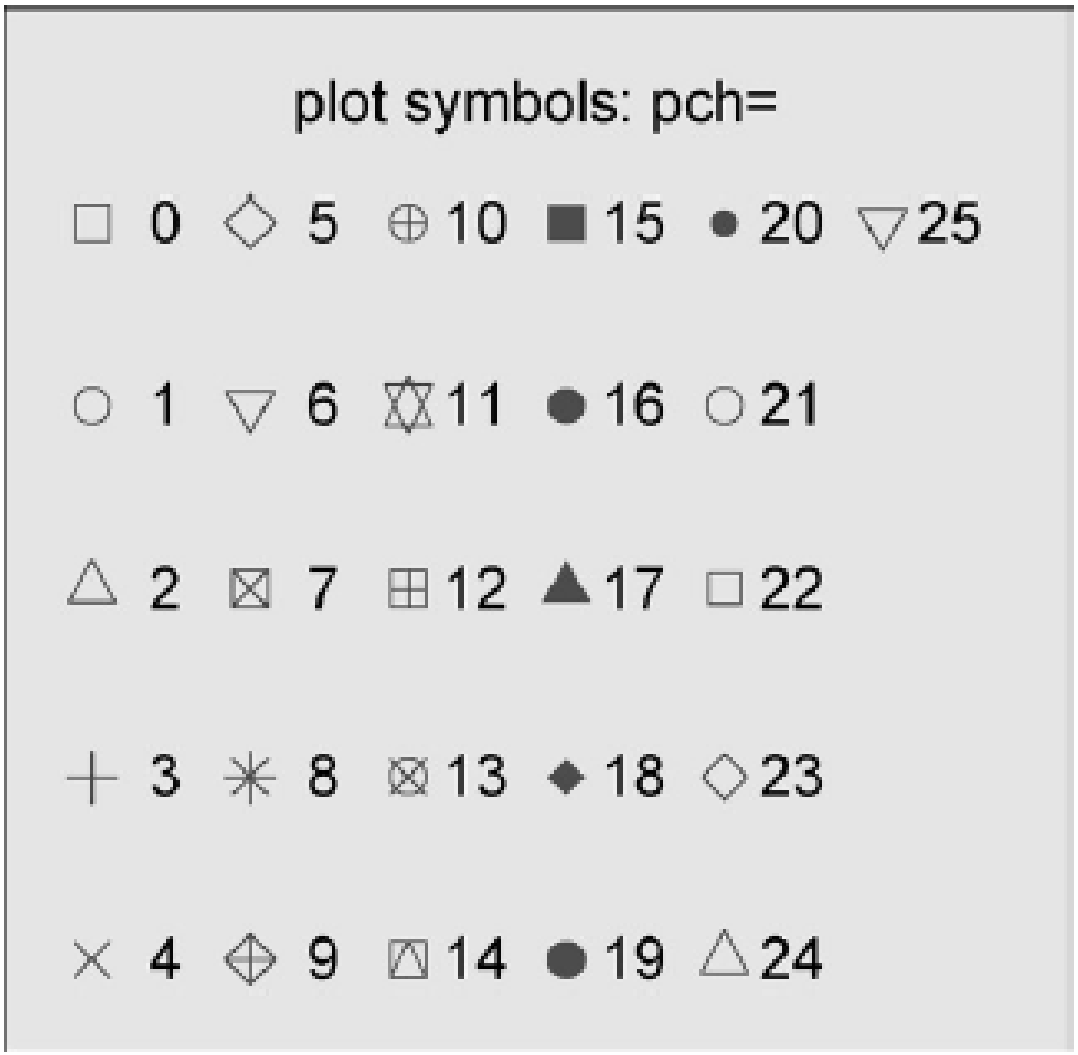
```
library("ggplot2")
ggplot(data =cars_data_small,aes(x=str, y=testscr)) +
geom_point(shape=1) # Use hollow circles
```



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)
```




plot symbols: pch=
0
5
10
15
20
25
1
6
11
16
21
2
7
12
17
22
3
8
13
18
23
4
9
14
19
24

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 ***
## ---
```

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 '
```

T-test in R

single sample

- t-test for scores

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

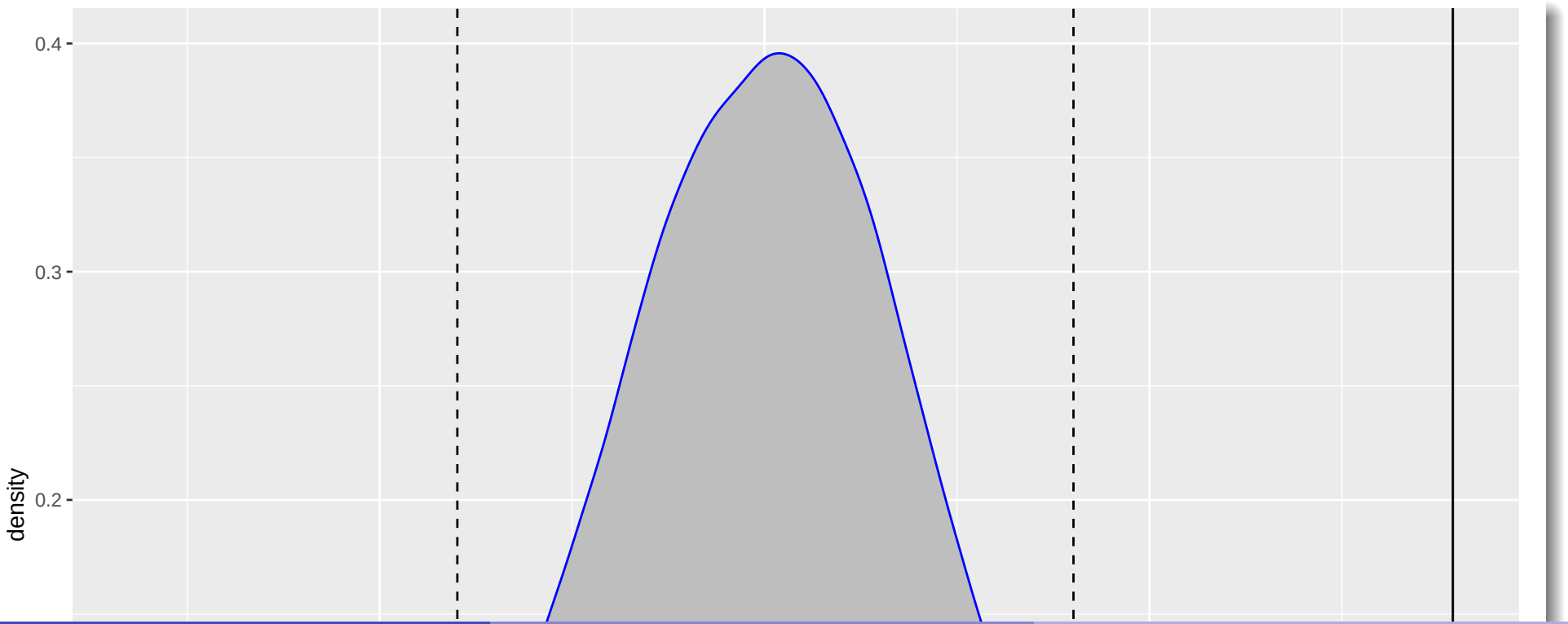
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   605.5   640.0   654.5   654.2   666.7   706.8
```

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

```
##
##  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$testscr,alternative = "tw
ggplot(data.frame(x=randT)) +
geom_density(aes(x=x),fill = "grey",color ="blue") +
geom_vline(xintercept = scoreTtest$statistic) +
geom_vline(xintercept = mean(randT) + c(-2,2)*sd(randT),linety
```



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.