敘述統計相關指令

- Textbook reading: Chapter 13.
- Summary statistics. Suppose that `x` is a numeric vector in R. Then the sample median, sample mean, sample standard deviation and sample variance of `x` can be computed using R commands

```
median(x)  #sample median
mean(x)    #sample mean
sd(x)      #sample standard deviation
var(x)     #sample variance
```

  respectively.
- The sample quantile of `x` can be calculated using `quantile`.
- The R command for drawing the histogram (直方圖) of `x` with break point vector `y` is `hist(x, breaks =y)`.
  - When no break points are given, it is recommanded to determine the break points based on Scott's rule to draw the histogram. The R command is `hist(x, nclass="scott")`.
  - To draw a normalized histogram (for density estimation), set `freq=FALSE` when using `hist`

Example 1. Generate a sample of size 1000 from $N(0,1)$ and save it to a vector `x` in R by running

```
set.seed(1)
x <- rnorm(100000)
```

(a) Compute the 25% and the 50% sample quantiles for the sample, and compare the sample quantiles with the quantiles of $N(0,1)$.

(b) Compute the sample mean, sample median, and sample variance for the sample.

(c) Draw a histogram of the sample using break points -5, -1, 0, 1, 5 and compare the histogram of the sample drawn using `hist(x, nclass="scott")`.

(d) Draw a normalized histogram of the sample and add the plot of the $N(0,1)$ density on $[-5,5]$ to it.

```
#a
quantile(x, c(0.25, 0.5)) ## the 25% and 50% sample quantiles of x
qnorm(c(0.25, 0.5))       ## the 0.25 and 0.5 quantiles for N(0,1)
#The sample quantiles and the normal quantiles are close.
```

```
#b
mean(x)
median(x)
var(x)

#c
par(mfrow=c(1,2))
hist(x, breaks=c(-5,-1,0,1,5))
hist(x, nclass="scott")

#d
par(mfrow=c(1,1))
hist(x, nclass="scott", freq=FALSE)
curve(dnorm, -5, 5, add=TRUE)
```

- Sample covariance and sample correlation.
  - The sample covariance between `x` and `y` can also be computed using the R command `cov(x,y)` or `var(x,y)`.
  - The sample correlation between `x` and `y` can also be computed using the R command `cor(x,y)`.
  - When `X` is a $n \times d$ data matrix of observations d variables, `cor(X)` and `var(X)` give the sample correlation matrix and the sample covariance matrix respectively.

Example 2. Suppose that X_1, X_2, X_3, X_4 are IID (independently and identically distributed) $N(0,1)$ random variables and

$$Y = X_1 + X_2 + X_4.$$

Generate 1000 IID observations for (Y, X_1, X_2, X_3, X_4) by running

```
set.seed(1)
X <- matrix(rnorm(4000, mean=0, sd=1), 1000,4)
y <- X[,1]+X[,2] + X[,4]
```

Write down R commands to compute the following results.

(a) The sample correlation between `y` and `X[,1]`.

(b) The sample correlation matrix for `y`, `X[,1]` and `X[,2]`.

(c) The sample covariance between `y` and `X[,1]`.

(d) The sample covariance matrix for `y`, `X[,1]` and `X[,2]`.

Sol.

```
#a
cor(y,X[,1])
#b
cor(cbind(y,X[,1:2]))
#c
cov(y,X[,1])
#or
f <- function(x,y){
 n <- length(x)
 x1 <- x-mean(x)
 y1 <- y-mean(y)
 return(sum(x1*y1)/(n-1))
}
f(y, X[,1])

#d
cov(cbind(y,X[,1:2]))
```

- Scatter plots (散佈圖).

  Example 3. Generate `x` and `y` such that `x` is the vector $(0, 1/20, 2/20, \ldots, 20/20)$ and `y` is `2*x+3` plus random errors from $N(0, 1/100)$, and then plot `y` against `x`. A fitted line `y=a+b*x` and the true line `y=2*x+3` are also added. For the fitted line, the slope `b=cor(x,y)*sd(y)/sd(x)` and the intercept `a=mean(y)-b*mean(x)`.

```
set.seed(1)
x <- seq(0, 1, by=0.05)
y <- 2*x+3 + rnorm(21, sd=1/10)
plot(x,y)
b <- cor(x,y)*sd(y)/sd(x)
a <- mean(y)-b*mean(x)
lines(x, a+b*x, col=4, lty=3)  #add the fitted line to the plot
lines(x, 2*x+3, col=2, lty=1)  #add the true line
legend(0,4.9, c("fitted line","true line"), col=c(4,2), lty=c(3,1))  #add legend
```

- To obtain a matrix of scatter plots for columns in a data matrix `X`, one can use `pairs(X)`.

  Example 4. Let `X` be the matrix in Example 2. Draw a matrix of pairwise scatter plots for the first three columns of `X`.

  Sol.

```
pairs(X[,1:3])
```

- When `x` is categorical, one can obtain the category frequencies using the command `"table(x)"`.

  Example 5. Define `x` as a vector of `"A"`'s, `"B"`'s and `"C"`'s and produce a summary table of frequencies.

```
x <- c(rep("A",3), rep("B",2), rep("C", 10))
table(x)
```

- In Example 5, one can find the mode of `x` from `table(x)`.

```
x <- c(rep("A",3), rep("B",2), rep("C", 10))
label <- names(table(x))
count <- as.numeric(table(x))
label[count==max(count)]  #mode of x
```