大數據統計與預測 期中筆記 (Big Data Statistics Midterm Code Cheat sheet)

Created by Weber, YC Huang 2019-11-09

This is a cheat sheet for using R

- How to use this sheet ?

  The **Basics** part at top allows you to create the **input number list** and **frequency table**, use these to perform the further calculation below ...

  Some self-definition functions below required inputting the specific parameter, make sure you type in the exact value.

  $Good\ luck\ for\ the\ midterm\ exam\ ><$!!!

Basics

1. **Create number list by yourself in console line**

   This is a function for you to type in the data one by one, as `float(input('Enter number :'))` in python

```r
# input values as a numeric list
input_list <- function(num) {
  l = c()
  for (i in c(1:num)) {
    question <- as.numeric(readline(prompt="Enter number as numerical list:
"))
    l <- append(l,question)
  }
  return(l)
}
number_list <- input_list()

# Enter number as numerical list: 1
# Enter number as numerical list: 1
# Enter number as numerical list: 1
# Enter number as numerical list: 2
# Enter number as numerical list: 2

# ======================================================

# extra : frequency table
freq_table <- function(list){
  t <- as.data.frame(table(x))
  t$Percentage <- 100*(t$Freq/sum(t$Freq))
  return(t)
}
t <- freq_table()

#  x Freq Percentage
#1 1    3         30
#2 2    2         20
```

```
#3 3    4          40
#4 4    1          10

library(ggplot2)
ggplot(t, aes(x=x, y=Percentage)) + geom_bar(stat="identity") +
  labs(x="data_point", y="Percentage")
```

Ch_2 : Data Presentation

1. **Relative frequencies table and Cumulative percentage**

```
# numerical list to percent
list2percent <- function(list){
  output = 100*list/sum(list)
  cum = 100*cumsum(list)/sum(list)
  cat('Percentage :',output, '\n')
  cat('Cumulative percentage :',cum)
}
list2percent()
```

2. **Class interval**

```
# class interval
class_interval <- function(list_1, list_2){
  list_3 <- round((list_1 + list_2)/2)
  return(list_3)
}
class_interval()
# freq ploygon
plot(
  x = ans_interval, y = 'values',
  xlab = "Class interval", ylab = "frequency",
  main = "Frequency ploygon"
)
```

3. **Box-plot**

```
boxplot('list',
        main = "Number of 'list' distribution",
        xlab = "Number",
        ylab = "list",
        col = "powderblue",
        border = "blue",
        horizontal = TRUE,
        notch = TRUE
)
```

4. **Histogram**

```
hist(x,y)
```

Ch_3 : Numerical Summary Measure

1. **Summary of the data**

   Output :

   - Min
   - 1st Qu
   - Median
   - Mean
   - 3rd Qu
   - Max
   - Standard deviation
   - Range
   - IQR
   - Mode
   - Coefficient of variation

```
summary_data <- function(data){
  # Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  summ <- summary(data)
  # sd
  s <- sd(data)
  # range
  ran <- max(data)-min(data)
  # IQR
  iqr <- IQR(data)
  # Mode
  Mod <- as.numeric(names(table(data)))[which.max(table(data))]
  # Cv
  CV <- 100*sd(data)/mean(data)

  print(summ)
  cat('\n', 'sd :',s , ';range :', ran, ';iqr :', iqr, ';mode', Mod, ';CV',
CV)
}
summary_data()
```

2. **Chebyshev's Inequality & Empirical Value**

   - Empirical Value : The data should be symmetric and unimodal. (資料是對稱且單峰狀態下才能使用)

     Approximately 67% of the observations lie in the interval $\overline{x} \pm 1s$, About 95% in the interval

     $\overline{x} \pm 2s$

   - It allows us to say that for any number k that is greater than or equal to 1, at least of $[1-(\frac{1}{k})^2]$measurements in the set of data lie within k standard deviations of their mean. (至少有$[1-(\frac{1}{k})^2]$會落在 $\pm k$ 倍標準差內)

```
# Empirical Value
# num = {67% : 1, 95% : 2, 99.7% : 3}
empirical_value <- function(data, num){
  upper <- mean(data)+num*sd(data)
  lower <- mean(data)-num*sd(data)
  cat(upper, ',', lower)
```

```
}
empirical_value()

# Chebyshev's Inequality
chebyshev_inequality <- function(k){
  interval <- 100*(1-(1/k)^2)
  cat('At least ', interval, '% ', 'of data will lie within + -', k, '
standard deviation of their mean')
}
chebyshev_inequality()
# At least  75 %  of data will lie within + - 2  standard deviation of their
mean
```

Ch_6 : Probability

1. **Probability I**

   Calculate **both** and **intersection** (算聯集與交集) , and **conditional probability I (not independent)** :

   Input event a and event b probabilities and both or intersection

```
# If not independent, we shall calculate the both and intersection manually
probability_1<- function(a, b, x){
  # x can be either both or intersection
  inter_both <- a+b-x
  cat('The answer (both/inter) is : ',inter_both, '\n')

  cat('if input is intersection ... if not, ignore the outcome below
...','\n')
  # if probability under b, probability a?
  conditional_prob_b <- x/b
  # if probability under b, probability a?
  conditional_prob_a <- x/a
  cat('event A occurs given that event B occurs :',conditional_prob_b, '
;event B occurs given that event A occurs :',conditional_prob_a)
}
probability_1()
```

2. **Conditional Probability II (independent)**

   Applying from probability **table** :

```
# applying from probability table
conditional_probability <- function(list_a, list_b){
  cat('event A occurs given that event B occurs :',sum(list_a), ' ;event B
occurs given that event A occurs :',sum(list_b))
}
```

3. **Diagnostic Test**

	Disease	Disease	

	Disease	Disease	
Test	Present	Absent	Total
+	a	c	a+c
-	b	d	b+d
Total	a+b	c+d	a+b+c+d

$$Sensitive = P(T^+|D^+) = \frac{P(T^+ \bigcap D^+)}{P(D^+)} = \frac{a}{a+b}$$

$$Specificity = P(T^-|D^-) = \frac{P(T^- \bigcap D^-)}{P(D^-)} = \frac{d}{c+d}$$

$$FN = P(T^-|D^+) = 1 - Sensitive = 1 - P(T^+|D^+) = 1 - \frac{a}{a+b}$$

$$FP = P(T^+|D^-) = 1 - Specificity = 1 - P(T^-|D^-) = 1 - \frac{d}{c+d}$$

$$Predictive\ value\ positive = P(D^+|T^+) = \frac{P(D^+ \bigcap T^+)}{P(T^+)} = \frac{Sensitive \times P(D^+)}{Sensitive \times P(D^+) + (1-Specificity) \times (1-P(D^+))}$$

$$Predictive\ value\ negative = P(D^-|T^-) = \frac{P(D^- \bigcap T^-)}{P(T^-)} = \frac{Specificity \times (1-P(D^+))}{Specificity \times (1-P(D^+)) + (1-Sensitive) \times P(D^+)}$$

```
# givcen the table:
# a = D+ T+
# b = D+ T-
# c = D- T+
# d = D- T-
# e = prevalence (P(D+))
diagnostic_test_table <- function(a,b,c,d,e){
  sen <- a/(a+b)
  sp <- d/(c+d)
  fn <- 1 - sen
  fp <- 1 - sp
  pp <- sen*e/((sen*e)+(1-sp)*(1-e))
  pn <- sp*(1-e)/(sp*(1-e)+(1-sen)*e)

  cat(' Sensitive :', sen, '\n', 'Specificity :', sp, '\n', 'FN :', fn,
'\n', 'FP :', fp, '\n', 'P+ :', pp, '\n', 'P- :', pn)
}
diagnostic_test_table()

# givcen Sensitive, Specificity,  e = prevalence (P(D+)):
# fn, fp, pp, pn ?
diagnostic_test_sen_sp <- function(sen,sp,e){
  fn <- 1 - sen
  fp <- 1 - sp
  pp <- sen*e/((sen*e)+(1-sp)*(1-e))
  pn <- sp*(1-e)/(sp*(1-e)+(1-sen)*e)

  cat(' FN :', fn, '\n', 'FP :', fp, '\n', 'P+ :', pp, '\n', 'P- :', pn)
}
diagnostic_test_sen_sp()

# given P+, P- , e = prevalence (P(D+)):
# sen, sp, fn, fp ?
diagnostic_pp <- function(pp, pn, e){
  k1 <- (pp*(1-e))/(e*(1-pp))
```

```
  k2 <- (pn*e)/((1-e)*(1-pn))

  sen <- (k1-k1*k2)/(1-k1*k2)
  sp <- (k2-k1*k2)/(1-k1*k2)

  fn <- 1 - sen
  fp <- 1 - sp

  cat(' Sensitive :', sen, '\n', 'Specificity :', sp, '\n', 'FN :', fn,
'\n', 'FP :', fp)
}

diagnostic_pp()
```

4. **Relative Risk**

	Exposed	Unexposed
Disease	a	c
No Disease	b	d
Total	a+b	c+d

$$RR = \frac{P(disease|exposed)}{P(disease|unexposed)} = \frac{a/(a+b)}{c/(c+d)}$$

- RR=1, the probabilities of disease in the exposed and unexposed groups are identical; an association between the exposed and the disease does not exits.
- RR>1, there is an increased risk of disease among those with the exposure. (暴露者得病風險愈高)
- RR<1, there is a decreased risk of developing disease among the exposed individual. (非暴露者得病風險愈高)

```
  rr <- function(a,b,c,d){
    molecule <- a/(a+b)
    denominator <- c/(c+d)
    output <- molecule/denominator

    cat(' Relative Risk :', output)
  }
  rr()
```

5. **Odd Ratio**

	Exposed	Unexposed	total	P	P'
Disease	a	c	a+c	a/(a+c)	1-(a/(a+c))
No Disease	b	d	b+d	b/(b+d)	1-(b/(b+d))
Total	a+b	c+d	a+b+c+d		

$$Odd\ Ratio = \frac{P(Exposed|Disease)/(1-P(Exposed|Disease))}{P(Exposed|No\ Disease)/(1-P(Exposed|No\ Disease))} = \frac{(a/(a+c))/(1-(a/(a+c)))}{(b/(b+d))/(1-(b/(b+d)))}$$

- Odds ratio=1: the exposure does not have an effect on the probability of disease

- The relative risk and the odds ratio are two different measures that attempt to explain the same phenomenon. In any event, for rare disease, the odds ratio is a close approximation of the relative risk.
- E.g. OR =1.05, woman who have used oral contraceptives have an odds of developing breast cancer that is only 1.05 times the odds of nonusers.

```
# e.g. Disease relative to No Disease
odd_ratio <- function(a,b,c,d){
  molecule <- (a/(a+c))/(1-(a/(a+c)))
  denominator <- (b/(b+d))/(1-(b/(b+d)))
  output <- molecule/denominator

  cat(' Odd Ratio :', output)
}
odd_ratio()
```

Ch_7 : Theoretical Probability Distribution

1. **Binomial Experiment**

$\mu = np$

$\sigma = \sqrt{npq}$

```
# ===== Common to use
# q is a vector of numbers. Output probability
pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)

# p is a vector of probabilities. Output cumulative value
qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)

# ===== Not quite
# x is a vector of numbers. Output distribution, usually if you wanna
plot...
dbinom(x, size, prob, log = FALSE)
# e.g. plot(x,dbinom(x, size, prob, log = FALSE))

# n is number of observations. Output required number of random values of
given probability
rbinom(n, size, prob)
```

2. **Poisson Distribution**

$\frac{e^{-}\lambda \times \lambda^{x}}{x!}$

- When n is very large and p is very small, the binomial distribution is well approximated by Poisson distribution. It is used to model discrete events that occur infrequently in time or space. (the distribution of rare events)

```
# x, vector of (non-negative integer) quantiles.
dpois(x, lambda, log = FALSE)

# q, vector of quantiles.
ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)

# p, vector of probabilities.
```

```
qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)

# n, number of random values to return.
rpois(n, lambda)

# ===
# At most n __ will be reported P(x <= n):
ppois(n, lambda, lower.tail = TRUE, log.p = FALSE)

# n or more P(x >= 6):
ppois((n-1), lambda, lower.tail = FALSE, log.p = FALSE)
```

3. **Normal Distribution**

```
dnorm(x, mean, sd, log = FALSE)
pnorm(q, mean, sd, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean, sd, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean, sd)

# P(x < n):
pnorm(n, mean, sd, lower.tail = TRUE, log.p = FALSE)
# P(x > n):
pnorm(n, mean, sd, lower.tail = FALSE, log.p = FALSE)
# range i~j P(i<x<j) or ~i-j~ (x<=i & x>=j), sample size (s):
range_normal <- function(i,j,s,mean,sd){
  ans = (1-pnorm(i, mean, sd, lower.tail = TRUE, log.p = FALSE)-pnorm(j,
mean, sd, lower.tail = FALSE, log.p = FALSE))^s
  cat(' inside range ',i,' to ',j,' is :',ans,'\n',' outside range ',' is
:',(1-ans))
}
range_normal()
```

4. **Student T distribution**

```
# ncp : non-centrality parameter delta; currently except for rt(), only for
abs(ncp) <= 37.62. If omitted, use the central t distribution.

dt(x, df, ncp, log = FALSE)
pt(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
rt(n, df, ncp)
```

- Density of t-distribution. (t 分布的特質):
  - 以 μ 為中心左右對稱。
  - 形狀像鐘形。
  - 兩尾端向左右兩端無限延伸。
  - 自由度 df 越大，曲線分散程度越小，即越高窄。
  - t 分布的圖形較 N(0,1)來得矮寬。

ch_8 : Sampling Distribution of the Mean

$$Standard\ Error = \frac{\sigma}{\sqrt{(n)}}$$

```r
# What is the proportion of the means of samples of size n are larger/smaller
than k?
pnorm(k, mean, standard_error, lower.tail = FALSE, log.p = FALSE)
pnorm(k, mean, standard_error, lower.tail = TRUE, log.p = FALSE)

# range i~j P(i<x<j) or ~i-j~ (x<=i & x>=j) , sample size (s):
range_normal_sampling <- function(i,j,s,mean,standard_error){
  ans = (1-pnorm(i, mean, standard_error, lower.tail = TRUE, log.p = FALSE)-
pnorm(j, mean, standard_error, lower.tail = FALSE, log.p = FALSE))^s
  cat(' inside range ',i,' to ',j,' is :',ans,'\n',' outside range ',' is :',(1-
ans))
}
range_normal_sampling()

# Way to calculate 99% 95% 90% sample size
sample_determine <- function(x, sigma, ci_rate, ci){
  S <- 0
  if (ci_rate == 0.99){
    S <- (2.58*sigma/(ci/2))^2
  }
  else if (ci_rate == 0.95) {
    S <- (1.96*sigma/(ci/2))^2
  }
  else {
    S <- (1.645*sigma/(ci/2))^2
  }
  cat(ci_rate*100,'% n_size :',S)
}
sample_determine(13.3, 1.12, 0.95, 0.4) # Within ci = 0.2+0.2
# 95 % n_size : 120.4726
```

ch_9 : Confidence intervals

```r
# ci_normal
ci_normal <- function(x, sigma, ci_rate, size){
  up <- 0
  low <- 0
  if (ci_rate == 0.99){
    up <- x + 2.58*(sigma/sqrt(size))
    low <- x - 2.58*(sigma/sqrt(size))
  }
  else if (ci_rate == 0.95) {
    up <- x + 1.96*(sigma/sqrt(size))
    low <- x - 1.96*(sigma/sqrt(size))
  }
  else {
    up <- x + 1.645*(sigma/sqrt(size))
    low <- x - 1.645*(sigma/sqrt(size))
  }
  cat('(',low,',',up,')')
}
ci_normal()
# e.g. ( 122.6863 , 137.3137 )

# ======================
```

```r
# Student T
# ncp : non-centrality parameter delta; currently except for rt(), only for
abs(ncp) <= 37.62. If omitted, use the central t distribution.
dt(x, df, ncp, log = FALSE)
pt(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
rt(n, df, ncp)

# T distribution calculate confidence interval
t_dis_ci <- function(x, sd, n, ci_rate){
  ci <- 1 - ((1 - ci_rate)/2)
  error <- qt(ci,df = n-1)*s/sqrt(n)
  left <- x - error
  right <- x + error

  cat('(',left,',',right,')')
}
t_dis_ci()

# ======================

# one sided ci
# obs = number seq
one_side_95 <- function(obs){
  x <- mean(obs)
  s <- sd(obs)
  n <- length(obs)
  cat('sample mean :',x,';sample std :',s,';n size :',n,'\n')

  upper <- x+1.645*(s/sqrt(n))
  lower <- x-1.645*(s/sqrt(n))
  cat('upper ci :',upper,';lower ci :',lower)
}
one_side_95()
```