✪ Psychological Statistics ✪

Week 02: *Probability & Distribution*

- Edited by Prof. **Changwei Wu**
- Graduate Institute of Mind, Brain and Consciousness (***GIMBC***), Taipei Medical University

In [ ]:

```
### [ Setup the working directory ]

setwd("/Users/wesley/[Course]/Python/R_Script")
# → Please edit the directory name in your computer.
getwd()
```

In [7]:

```
### [ Loading the required libraries ]

#install.packages("pastecs")
#install.packages("readxl")
#install.packages("reshape2")
#install.packages("stringr")

library("pastecs")
library("readxl")
library("reshape2")
library("stringr")
```

(1) Check Input Data / Output

In [2]:

```
### 1-1.[ Import data from CSV data files ]
# → Function: read.csv {utils}

Exp<-read.csv("Ex1.csv",header=T)
class(Exp)

# → "data frame" is a 2-D datasheet, like the data storage in an Excel file

Exp[1:5,]
```

'data.frame'

A data.frame: 5 × 5

	time	latitude	longitude	depth	mag
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>
1	1900-07-29T06:59:00.000Z	-10	165	0	7.6
2	1900-10-09T12:28:00.000Z	60	-142	0	7.7
3	1900-10-29T09:11:00.000Z	11	-66	0	7.7
4	1901-08-09T13:01:00.000Z	-22	170	0	7.9
5	1901-08-09T18:33:00.000Z	40	144	0	7.5

In [3]:

```
### 1-2(1).[ Check input data ]
# → Function: dim (dimension) & colnames

dim(Exp)

colnames(Exp)

Exp[1:5,"mag"]

Exp$mag[1:5]

head(Exp$mag)
```

1292 · 5

'time' · 'latitude' · 'longitude' · 'depth' · 'mag'

7.6 · 7.7 · 7.7 · 7.9 · 7.5

7.6 · 7.7 · 7.7 · 7.9 · 7.5

7.6 · 7.7 · 7.7 · 7.9 · 7.5 · 7.5

In [4]:

```
### 1-2(2).[ Check input data ]
# → Select certain data out from the original datasheet

select <- Exp[Exp$depth>40,]
head(select)
```

A data.frame: 6 × 5

	time	latitude	longitude	depth	mag
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>
16	1906-01-21T13:49:33.000Z	34.167	138.101	300	7.4
18	1906-08-17T00:11:10.000Z	51.853	178.180	110	8.3
26	1909-07-07T21:37:47.000Z	35.387	70.251	200	7.7
29	1910-04-12T00:22:24.000Z	25.911	123.973	235	8.1
30	1910-06-16T06:30:43.000Z	-19.572	169.438	100	7.8
36	1911-06-15T14:25:53.000Z	29.049	128.778	100	7.9

In [5]:

```
### 1-2(3).[ Check input data ]
# → Select certain data out from the original datasheet

select <- Exp[Exp$depth>40 & Exp$latitude>0,]
dim(select)

dim(select)[1]
dim(Exp)[1]

round(dim(select)[1]/dim(Exp)[1],5)
```

130 · 5

130

1292

0.10062

In [8]:

```
### 1-3.[ Descriptive statistics ]
# → Function: mean & var {base} & stat.desc {pastecs}

mean(Exp$latitude)

var(Exp$latitude)

stat.desc(Exp$latitude, basic=F, desc=T, norm=T)
```

7.3894290247678

834.362678537461

median: 4.24 **mean:** 7.3894290247678 **SE.mean:** 0.803611564660203 **CI.mean.0.95:** 1.57652775896075 **var:** 834.362678537461 **std.dev:** 28.8853367392084 **coef.var:** 3.90900794126188 **skewness:** -0.106546308204452 **skew.2SE:** -0.782649674802826 **kurtosis:** -0.717243957444 **kurt.2SE:** -2.63633119521276 **normtest.W:** 0.976812881688002 **normtest.p:** 1.4819932028787e-13

In [9]:

```
### 1-4.[ Export data to files ]
# → Function: write.csv or write.table {utils}

write.table(select, "Table.txt", sep="\t", row.names = FALSE)
write.csv(select, "TEST.csv")
```

In [10]:

```
### 1-5.[ Import data from Excel files ]
# → Function: excel_sheets or read_excel {readxl}

excel_sheets('ExcelExample.xlsx')
tomatoXL <- read_excel('ExcelExample.xlsx')
wineXL1 <- read_excel('ExcelExample.xlsx', sheet=2)

head(wineXL1)
```

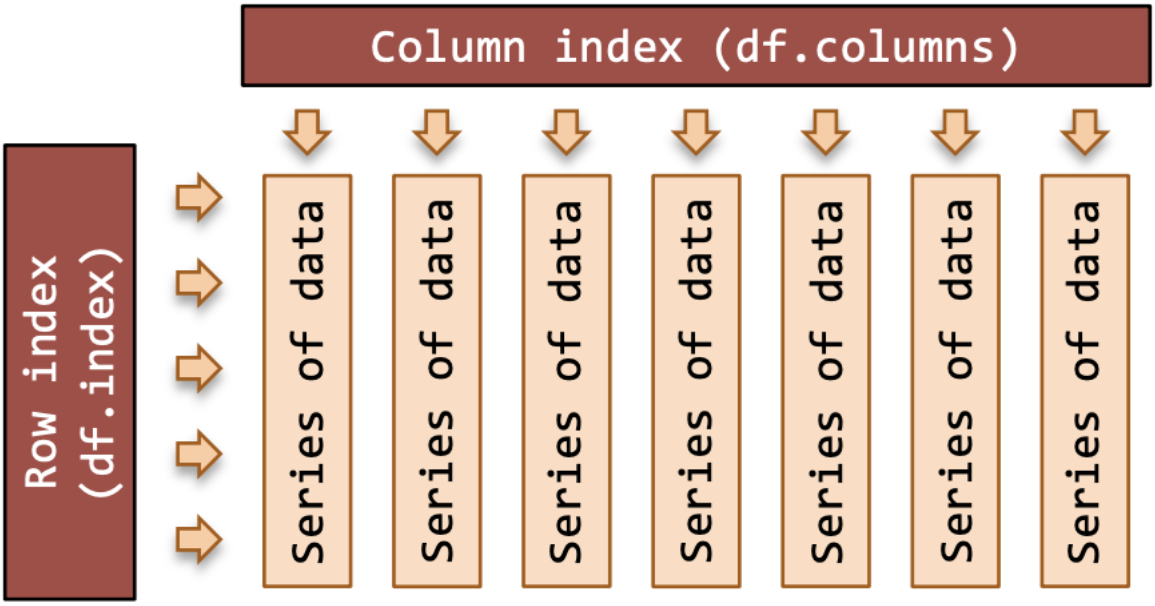
'Tomato' · 'Wine' · 'ACS'

A tibble: 6 × 14

Cultivar	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735
1	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450

(2) Restructure into Data Frame

What is Data Frame?



In [11]:

```
### 2-1.[ Combine string variables ]
# → Function: cbind {base}

sport <- c("Hockey", "Baseball", "Football")
league <- c("NHL", "MLB", "NFL")
trophy <- c("Stanley Cup", "Commissioner's Trophy", "Vince Lombardi Trophy")
(trophies1 <- cbind(sport, league, trophy))
```

A matrix: 3 × 3 of type chr

sport	league	trophy
Hockey	NHL	Stanley Cup
Baseball	MLB	Commissioner's Trophy
Football	NFL	Vince Lombardi Trophy

In [12]:

```
### 2-2.[ Combine the vectors ]
# → Function: data.frame {base}

trophies2 <- data.frame(sport=c("Basketball", "Golf"), league=c("NBA", "PGA"),
                        trophy=c("Larry O'Brien", "Championship"))
(trophies <- rbind(trophies1, trophies2))
```

A data.frame: 5 × 3

sport	league	trophy
<fct>	<fct>	<fct>
Hockey	NHL	Stanley Cup
Baseball	MLB	Commissioner's Trophy
Football	NFL	Vince Lombardi Trophy
Basketball	NBA	Larry O'Brien
Golf	PGA	Championship

In [13]:

```
### 2-3.[ Combine the vectors into a data frame ]
# → Function: data.frame
# Data Frame: collection of vectors, like data in Excel file

x <- 10:1
y <- -4:5
q <- c("Hockey", "Football", "Baseball", "Curling", "Rugby", "Lacrosse",
       "Basketball", "Tennis", "Cricket", "Soccer")
theDF <- data.frame(x, y, q)
theDF <- data.frame(First = x, Second = y, Sport = q)

nrow(theDF); ncol(theDF); dim(theDF) # check the dimension of the data frame

theDF
```

10

3

10 · 3

A data.frame: 10 × 3

First <int>	Second <int>	Sport <fct>
10	-4	Hockey
9	-3	Football
8	-2	Baseball
7	-1	Curling
6	0	Rugby
5	1	Lacrosse
4	2	Basketball
3	3	Tennis
2	4	Cricket
1	5	Soccer

In [14]:

```
### 2-4.[ To remove a column (vector) in a data frame ]
# → Using "NULL"

theDF[c(3,5), 2:3]
theDF$Sport <- NULL
theDF
```

A data.frame: 2 × 2

	Second <int>	Sport <fct>
3	-2	Baseball
5	0	Rugby

A data.frame: 10 × 2

First <int>	Second <int>
10	-4
9	-3
8	-2
7	-1
6	0
5	1
4	2
3	3
2	4
1	5

In [15]:

```
### 2-5.[ Changing between the long form and the wide form ] 【Practice off-class】
# → Function: melt/dcast {reshape2} & str_sub {stringr}

Aid_00s <- read.csv("US_Foreign_Aid_00s.csv")
melt00 <- melt(Aid_00s, id.vars=c("Country.Name", "Program.Name"),
               variable.name="Year", value.name="Dollars")

melt00$Year <- as.numeric(str_sub(melt00$Year, start=3, 6))

cast00 <- dcast(melt00, Country.Name + Program.Name ~ Year, value.var = "Dollars")

head(Aid_00s)
head(cast00)
```

A data.frame: 6 × 12

	Country.Name	Program.Name	FY2000	FY2001	FY2002	FY2003	FY2004	FY2005	FY2006	FY2007	FY2008	FY2009
	<fct>	<fct>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<int>	<int>	<int>
1	Afghanistan	Child Survival and Health	NA	NA	2586555	56501189	40215304	39817970	40856382	72527069	28397435	NA
2	Afghanistan	Department of Defense Security Assistance	NA	NA	2964313	NA	45635526	151334908	230501318	214505892	495539084	552524990
3	Afghanistan	Development Assistance	NA	4110478	8762080	54538965	180539337	193598227	212648440	173134034	150529862	3675202
4	Afghanistan	Economic Support Fund/Security Support Assistance	NA	61144	31827014	341306822	1025522037	1157530168	1357750249	1266653993	1400237791	1418688520
5	Afghanistan	Food For Education	NA	NA	NA	3957312	2610006	3254408	386891	NA	NA	NA
6	Afghanistan	Global Health and Child Survival	NA	NA	NA	NA	NA	NA	NA	NA	63064912	1764252

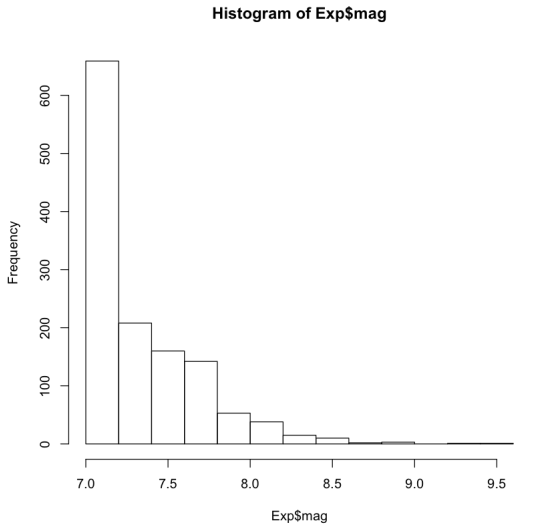
A data.frame: 6 × 12

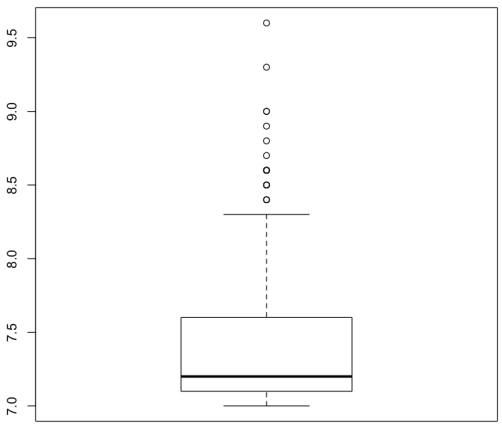
	Country.Name	Program.Name	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Afghanistan	Child Survival and Health	NA	NA	2586555	56501189	40215304	39817970	40856382	72527069	28397435	NA
2	Afghanistan	Department of Defense Security Assistance	NA	NA	2964313	NA	45635526	151334908	230501318	214505892	495539084	552524990
3	Afghanistan	Development Assistance	NA	4110478	8762080	54538965	180539337	193598227	212648440	173134034	150529862	3675202
4	Afghanistan	Economic Support Fund/Security Support Assistance	NA	61144	31827014	341306822	1025522037	1157530168	1357750249	1266653993	1400237791	1418688520
5	Afghanistan	Food For Education	NA	NA	NA	3957312	2610006	3254408	386891	NA	NA	NA
6	Afghanistan	Global Health and Child Survival	NA	NA	NA	NA	NA	NA	NA	NA	63064912	1764252

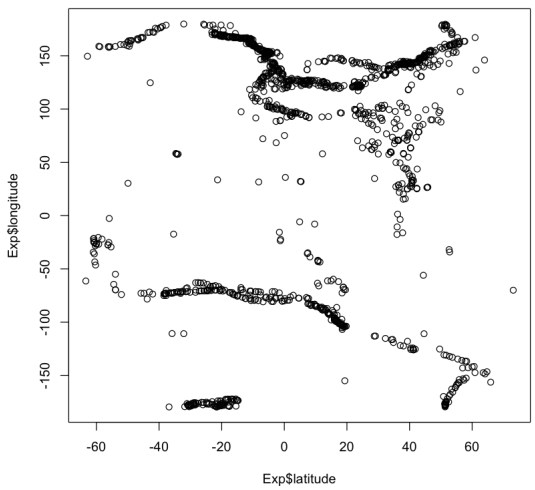
(3) Basic plotting: Distribution

In [16]:

```
### 3-1.[ Plot Histogram (frequency counts) ]
# → Function: hist, plot, boxplot {graphics}

hist(Exp$mag)
boxplot(Exp$mag)
plot(Exp$latitude, Exp$longitude)
```

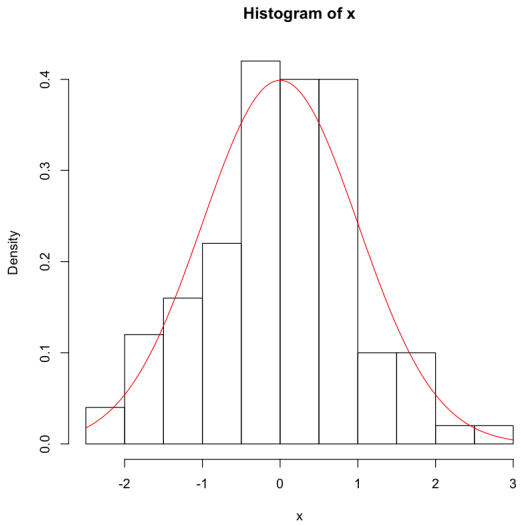






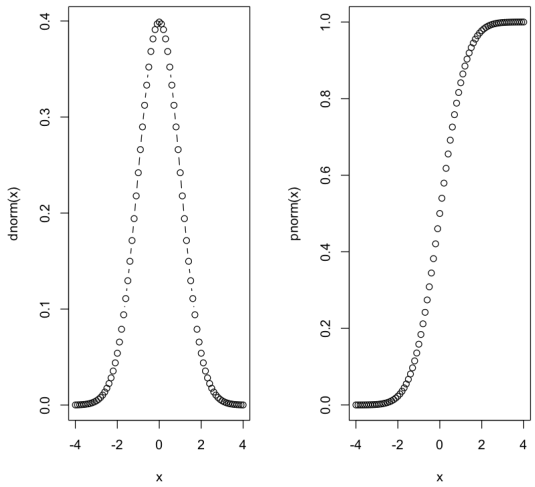
In [17]:

```
### 3-2.[ Plot the normal distribution ]
# → Function: rnorm {stats} & curve {graphics}

x<-rnorm(100) # → random variables in normal distribution
hist(x,freq=F)
curve(dnorm(x),add=T, col="red")
```



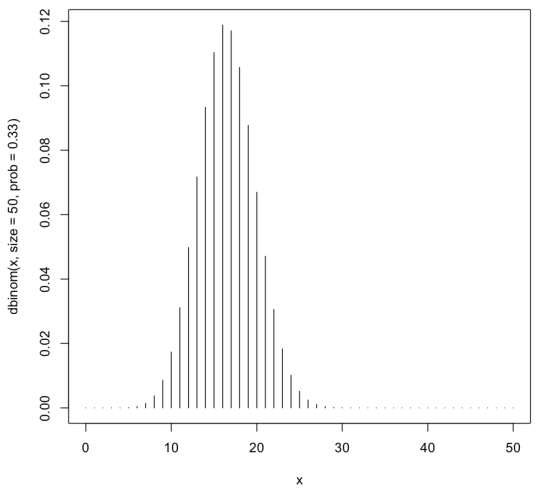
In [18]:

```
### 3-3.[ Plot both PDF & CDF of normal distribution]
# → Function: dnorm & pnorm {stats}

par(mfrow=c(1,2)) # (cut the column into 2) plot 2 fig in a row
x<-seq(-4,4,0.1)
plot(x,dnorm(x),type="b")
plot(x,pnorm(x),type="b")
```



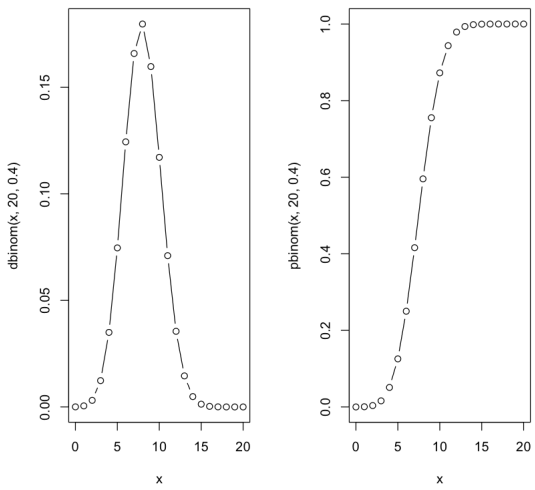
In [19]:

```
### 3-4.[ Function of "binom" (binomial distribution) ]
# → Function: dbinom {stats}
# {Note} the inputs of distribution function are different!

x<-0:50 # 50 trials, probability of success = 0.33
plot(x,dbinom(x,size=50, prob=.33),type="h")
```



In [29]:

```
# → Function: dbinom & pbinom {stats} for binomial distribution

par(mfrow=c(1,2))
x<-seq(0,20)
plot(x,dbinom(x,20,0.4),type="b")
plot(x,pbinom(x,20,0.4),type="b")
```



【Suggested practice】Please figure out the disparities between 'dnorm', 'pnorm', 'rnorm', 'qnorm'

【Suggested practice】Please read the Cheatsheet of "Dplyr" for data arrangement.