1 Chapter 1 (R programs)

Example–1–1–1.r

```
## ==========================
## Example 1.1-1 on Page 13
##  (same as Example 1.4-6 on Page 40)
## --------------------------

# E = a set of outcomes
# M = 1:6
M = 1:6

# At least !
E = sample(1:6, replace=TRUE)
any(E == M)


# Setup for computer simulation.
N = 500
f = numeric(N)
# =================
# Start: Simulation
for ( n in 1:N ) {
    tmp = numeric(n)
    for ( i in 1:n ) {
        E = sample(1:6, replace=TRUE)
        tmp[i] =  any(E == M)
    }
    f[n] = sum(tmp)
}
# End:  Simulation
# =================

# Print the results,
cbind(1:N, f, f/(1:N))

# Figure 1.1-3 on Page 13
plot(1:N, f/(1:N))

# More cosmetic
plot(f/(1:N), type="l", col="blue")
plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
abline(h=1-(5/6)^6)
```

Example–1–1–2.r

```
## ==========================
## Example 1.1-2 on Page 14
## --------------------------

N = 50000 # sample size

x = runif(N, min=-2, max=2)
y = runif(N, min=-2, max=2)

Inside = ( abs(x) < 1) & ( abs(y) < 1)
sum(Inside) / N

# Cosmetics
# plot(x, y, pch=".")
```

```r
  plot( x[Inside], y[Inside],  pch=".", col="blue", xlim=c(-2,2), ylim=c(-2,2) )
points( x[!Inside],y[!Inside], pch=".", col="red")
rect( -1,-1, 1, 1, border ="green4")


## =========================
## Example 1: Extra (Obtaining pi=3.14 using a circle)
## -------------------------
N = 50000 # sample size

x = runif(N, min=-1, max=1)
y = runif(N, min=-1, max=1)

Inside = ( x^2 + y^2 < 1 )
sum(Inside) / N * 4


# Cosmetics
# plot(x, y, pch=".")
  plot( x[Inside], y[Inside],  pch=".", col="blue", xlim=c(-1,1), ylim=c(-1,1))
points( x[!Inside],y[!Inside], pch=".", col="red")

angle = 2*pi* seq(0,1,length=101)
r1 = cos(angle)
r2 = sin(angle)
lines(r1, r2, col="green4")

#----------------------------
# Convergence
#----------------------------
##NN = c(5, 10, 20, 30, 100, 500, 1000, seq(2000, 50000, by=1000) )
NN = seq(5, 5000, by=5)
n  = length(NN)
PI = numeric(n)

for ( i in 1:length(NN) ) {
   x = runif(NN[i], min=-1, max=1)
   y = runif(NN[i], min=-1, max=1)
   Inside = ( x^2 + y^2 < 1 )
   PI[i] = sum(Inside) / NN[i] * 4
}

plot(NN, PI, type="l", col="blue" )
abline(h=3.14, col="red4")


## =========================
## Example 2: Extra (Obtaining pi=3.14 using a ball)
## -------------------------
N = 50000 # sample size

x = runif(N, min=-1, max=1)
y = runif(N, min=-1, max=1)
z = runif(N, min=-1, max=1)

Inside = ( x^2 + y^2 + z^2 < 1 )
sum(Inside) / N * 6

#----------------------------
# Convergence
#----------------------------
##NN = c(5, 10, 20, 30, 100, 500, 1000, seq(2000, 50000, by=1000) )
```

```r
NN = seq(5, 5000, by=5)
n  = length(NN)
PI3 = numeric(n)

for ( i in 1:length(NN) ) {
   x = runif(NN[i], min=-1, max=1)
   y = runif(NN[i], min=-1, max=1)
   z = runif(NN[i], min=-1, max=1)
   Inside = ( x^2 + y^2 + z^2 < 1 )
   PI3[i] = sum(Inside) / NN[i] * 6
}

plot(NN, PI3, type="l", col="blue" )
abline(h=3.14, col="red4")


#----------
 plot(NN, PI,  type="l", col="blue" )
lines(NN, PI3, type="l", col="red" )
abline(h=3.14, col="black")
```

Example–1–2–4.r

```r
## ==========================
## Example 1.2-4 on page 21
## --------------------------

factorial(26) / factorial(22)


## ==========================
## Example 1.2-5 on page 21
## --------------------------

factorial(10) / factorial(6)


## ==========================
## Example Extra
## --------------------------

factorial(100) / factorial(99)

factorial(200) / factorial(199)   # Something wrong

exp( lfactorial(200) - lfactorial(199) )  # Success
```

Example–1–2–9.r

```r
## ==========================
## Example 1.2-9 on page 23
## --------------------------

factorial(52) / ( factorial(5) * factorial(47) )



## more simple

choose(52,5)
```

Example–1–4–6–advanced.r

```r
## ==========================
## Example 1.4-6 on Page 40
##  (Recall Example 1.1-1 on Page 13)
## --------------------------

date(); now <- proc.time()  ##############

# E = a set of outcomes
M = 1:6

# At least!
E = sample(1:6, replace=TRUE)
any(E == M)

# Setup for computer simulation.
N = 5000
f = numeric(N)

# =================
# Start: Simulation
for ( n in 1:N ) {
    tmp = numeric(n)
    for ( i in 1:n ) {
        E = sample(1:6, replace=TRUE)
        tmp[i] =  any(E == M)
    }
    f[n] = sum(tmp)
}
# End:  Simulation
# =================
(proc.time()-now)/3600;  #################


# Print the results,
cbind(1:N, f, f/(1:N))

# Figure 1.1-3 on Page 13
plot(1:N, f/(1:N))

# More cosmetic
plot(f/(1:N), type="l", col="blue")
plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
abline(h=1-(5/6)^6)

#=========================
# Very Advanced

#-------------------------
date(); now <- proc.time()  ##############
N = 5000  # iteration
m = 6    # 6 faced die

f = numeric(N)
M = matrix( rep(1:m, N), ncol=m, byrow=TRUE )
E = matrix( sample(1:6, size=m*N, replace=TRUE), ncol=m, byrow=TRUE )

TF1 = (E==M)
TF2 = apply( TF1, 1, any )
f   = cumsum(TF2)
(proc.time()-now)/3600;  #################

```

```r
plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
abline(h=1-(5/6)^6)
```

Example–1–4–6.r

```r

## ======================================
## Example 1.4-6 on Page 40
##  (Recall Example 1.1-1 on Page 13)
## --------------------------------------

## ======================================
## (i) Using the inclusion-exclusion formula
## --------------------------------------

p = 1/6
choose(6,1)*p -
choose(6,2)*p^2 +
choose(6,3)*p^3 -
choose(6,4)*p^4 +
choose(6,5)*p^5 -
choose(6,6)*p^6

## ======================================
## (ii) Using P(B) = 1 - P(B')
## --------------------------------------
p = 1/6

1 - (1-p)^6


## ======================================
## (iii) Using Simulation
## --------------------------------------

# E = a set of outcomes
M = 1:6

# Toss a die.
E = sample(1:6, replace=TRUE)

# Check if there is at least one match.
any(E == M)

# Setup for simulation.
N = 1000
f = numeric(N)
# =================
# Start: Simulation
for ( n in 1:N ) {
    tmp = numeric(n)
    for ( i in 1:n ) {
        E = sample(1:6, replace=TRUE)
        tmp[i] =  any(E == M)
    }
    f[n] = sum(tmp)
}
# End:  Simulation
# =================

# Print the results,
cbind(1:N, f, f/(1:N))
```

```r

# Figure 1.1-3 on Page 13
plot(1:N, f/(1:N))

# More cosmetic
plot(f/(1:N), type="l", col="blue")
plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
abline(h=1-(5/6)^6, col="red")
```

2 Chapter 2 (R programs)

Example–2–1–7.r

```r
## ==========================
## Example 2.1-7 on Page 54
## --------------------------

out =  rep( 2:8, c(71,124,194,258,177,122,54) )

table(out)

hist(out)  # Different from textbook

# Histogram
hist(out, breaks=seq(0.5,8.5,by=1) ) # slightly different

# Relative frequency histogram
hist(out, breaks=seq(0.5,8.5,by=1), prob=T ) #

#-----------------------------
# Theoretical pmf
x = 2:8
pmf = (4-abs(x-5))/16

# Let's check pmf
pmf > 0
sum(pmf)

#-----------------------------
# Relative frequency histogram with pmf
hist(out, breaks=seq(0.5,8.5,by=1), prob=T )
lines(x, pmf, type="h", col="red", lwd=10)

#-----------------------------
# Table 2.1-1
obs = table(out)
cbind(obs, obs/length(out), pmf)


#============================================
# How is actually the textbook example made
#-------------------------------------------
N = 1000   # We can also try N=100000


x1 = sample(1:4, size=N, replace=TRUE)
x2 = sample(1:4, size=N, replace=TRUE)

out = x1+x2
hist(out, breaks=seq(0.5,8.5,by=1) )  # N.B: triangle shape.

# Continuous analogy
x1 = runif(N)
x2 = runif(N)

out = x1+x2
hist(out)
```

Example–2–4–7.r

```r
## ==========================
## Example 2.4-7 on page 75
## --------------------------

# P( X <= 8 )
 pbinom (8, size=10, prob=0.8)

# P( X <= 8 ) = 1 - P(X=9) - P(X=10)
 1-dbinom(9,size=10,prob=0.8)-dbinom(10,size=10,prob=0.8)

# P( X <= 6 )
 pbinom (6.0, size=10, prob=0.8)
 pbinom (6.1, size=10, prob=0.8)
 pbinom (6.4, size=10, prob=0.8)
 pbinom (6.7, size=10, prob=0.8)
 pbinom (6.9, size=10, prob=0.8)

 pbinom (7.0, size=10, prob=0.8)

# ------------------------------------------------
# Figure 2.4-2 on Page 77
x = 0:10
cdf = c(0, pbinom(x, size=10, prob=0.8) )
Fx = stepfun(x, cdf)
plot(Fx, vertical=FALSE, pch=20, col="blue", ylab="F(x)" )

# The above is better. But the below is easier.
xx = seq(0, 10, by = 0.1)
Fxx = pbinom(xx, size=10, prob=0.8)
plot(xx, Fxx,  vertical=FALSE, type="l")
```

Figure–2–4–1.r

```r
## ==========================
## Figure 2.4-1 on Page 76
## --------------------------

# Histogram in R is only for a sample.

xx = 0:16
pmf = dbinom(xx, size=16, prob=0.75)
plot(xx, pmf, type="h", lwd=10, xlab="Bin(16,0.75)", ylab="f(x)")

# The above is not bad, but different from the textbook.

#----------------------------
names(pmf) = xx
barplot(pmf, col="white", xlab="Bin(16,0.75)", ylab="f(x)")

#===============================
# Let's make four plots into one.
#-------------------------------
par ( mfrow=c(2,2) )

xx = 0:16
pmf = dbinom(xx, size=16, prob=0.75)
names(pmf) = xx
barplot(pmf, col="white", xlab="Bin(16,0.75)", ylab="f(x)")

xx = 0:16
pmf = dbinom(xx, size=16, prob=0.50)
names(pmf) = xx
```

```r
barplot(pmf, col="white", xlab="Bin(16,0.50)", ylab="f(x)")

xx = 0:25
pmf = dbinom(xx, size=16, prob=0.35)
names(pmf) = xx
barplot(pmf, col="white", xlab="Bin(25,0.35)", ylab="f(x)")

xx = 0:25
pmf = dbinom(xx, size=16, prob=0.20)
names(pmf) = xx
barplot(pmf, col="white", xlab="Bin(25,0.20)", ylab="f(x)")
```

Example–2–5–4.r

```r
## ==========================
## Example 2.5-4 on page 85
## --------------------------

#
r=1; p=0.25; x = 0:25
f = dnbinom(x, size=r, prob=p)
plot(x+r, f, type="h", xlim=c(0,25) )

r=4; p=0.6;  x = 0:25
f = dnbinom(x, size=r, prob=p)
plot(x+r, f, type="h", xlim=c(0,25) )

r=7; p=0.7;  x = 0:25
f = dnbinom(x, size=r, prob=p)
plot(x+r, f, type="h", xlim=c(0,25) )

r=15; p=0.7;  x = 0:25
f = dnbinom(x, size=r, prob=p)
plot(x+r, f, type="h", xlim=c(15,35) )

#---------------------------------------
# Plot the above four into one sheet.
#---------------------------------------
par( mfrow=c(2,2))

# And then repeat the above four plots.
```

Example–2–6–1.r

```r
## ==========================
## Example 2.6-1 on Page 90
## Compare with Table III in Appendix B
## --------------------------

# ---------------------------
# P[ X <= 6 ]  with lambda = 5
xx = 0:6
dpois ( xx, lambda=5 )

sum( dpois ( xx, lambda=5 ) )

ppois(6, lambda=5)

# ---------------------------
# P[ X > 5 ] = 1 - P[ X <= 5 ]
1 - ppois(5, lambda=5)

```

```r
# Use the upper tail.
ppois(5, lambda=5, lower.tail=FALSE)


# ----------------------------
# P[ X = 6 ]
dpois(6, lambda=5)

ppois(6, lambda=5) - ppois(5, lambda=5)
```

Example–2–6–2.r

```r
## ==========================
## Example 2.6-2 on Page 90
## See Figure 2.6-1
## --------------------------


#================================
# Let's make four plots into one.
#--------------------------------
par ( mfrow=c(2,2) )

xx = 0:6
pmf = dpois(xx, lambda=0.7)
names(pmf) = xx
barplot(pmf, col="white", ylab="f(x)")

xx = 0:6
pmf = dpois(xx, lambda=1.3)
names(pmf) = xx
barplot(pmf, col="white", ylab="f(x)")

xx = 0:20
pmf = dpois(xx, lambda=6.5)
names(pmf) = xx
barplot(pmf, col="white", ylab="f(x)")

xx = 0:20
pmf = dpois(xx, lambda=10.5)
names(pmf) = xx
barplot(pmf, col="white", ylab="f(x)")
```

Example–2–6–5.r

```r
## ==========================
## Example 2.6-5 on Page 92
## Poisson approximation to Binomial
## --------------------------


xx = 0:3
dbinom(xx, size=100, prob=0.02)
dpois(xx, lambda=2)

#---------------------------
fbin = dbinom(xx, size=100, prob=0.02)
fpoi = dpois(xx, lambda=2)

rbind(xx, fbin, fpoi)

#---------------------------
```

```
sum(fbin)
sum(fpoi)
```

3 Chapter 3 (R programs)

Example–3–2–1.r

```
## ==========================
## Example 3.2-1 on Page 104
## --------------------------


# Note: R uses rate paramter instead of theta.

pexp(18, rate=1/20)
```

Example–3–2–2.r

```
## ==========================
## Example 3.2-2 on Page 105
## --------------------------

# Question: Find the median.

# Note: R uses rate paramter instead of theta.
qexp(1/2, rate=1/3)
```

Example–3–2–4.r

```
## ==========================
## Example 3.2-4 on Page 108
## --------------------------

# Note: R uses lambda (rate) and theta (scale) as well
#       alpha = shape

alpha=2
theta=2
x = 5

1 - pgamma(x, shape=alpha, scale=theta)
pgamma(x, shape=alpha, scale=theta, lower.tail=FALSE)

# Using ppois
ppois(alpha-1, lambda=x/theta)

# Using chisquare
1 - pchisq(x, df=4)
```

Example–3–2–6.r

```
## ==========================
## Example 3.2-6 on Page 109
## --------------------------

# Using chi-square distribution
pchisq(12.83, df=5) - pchisq(1.145, df=5)

# Note: we can also calcuate it using xgamma.
pgamma(12.83, shape=5/2, scale=2) - pgamma(1.145, shape=5/2, scale=2)
```

Example–3–2–9.r

```r
## =========================
## Example 3.2-9 on Page 110
## -------------------------

# Using chi-square distribution
# Unit time: minute
1 - pchisq(9.390, df=18)

# Note: we can also calcuate it using xgamma.
# Unit time: minute
1 - pgamma(9.390, shape=18/2, scale=2)

#-----------------------------------------
# Using xgamma.
# Unit time: hour
1 - pgamma(9.390/60, shape=18/2, scale=2/60)
```

Example–3–3–3.r

```r
## =========================
## Example 3.3-3 on Page 116
## -------------------------



#--------------------------
pnorm(1.24, mean=0, sd=1)
pnorm(1.24) # The same as the above



#--------------------------
pnorm(2.37) - pnorm(1.24)
pnorm(-1.24) - pnorm(-2.37)  # The same as the above



#--------------------------
1-pnorm(1.24)
pnorm(1.24, lower.tail=FALSE) # The same as the above



#--------------------------
pnorm(-2.14)
pnorm(2.14, lower.tail=FALSE)



#--------------------------
pnorm(0.77) - pnorm(-2.14)
```

Example–3–3–4.r

```r
## =========================
## Example 3.3-4 on Page 116
## -------------------------


a = qnorm(0.9147)
a

```

```r
b = qnorm(1-0.0526)
b
```

Example–3–3–5.r

```r
## ==========================
## Example 3.3-5 on Page 117
## --------------------------


 qnorm(1 - 0.0125)



 qnorm(1 - 0.05)



 qnorm(1 - 0.025)
```

Example–3–3–6.r

```r
## ==========================
## Example 3.3-6 on Page 118
## --------------------------


# (a) P( 4 < X < 8 )
# After standardization
 pnorm(1.25) - pnorm(0.25)

# Withoug standardization
 pnorm(8, mean=3, sd=4) - pnorm(4, mean=3, sd=4)


# (b) P( 0 < X < 5 )
# After standardization
 pnorm(0.5) - pnorm(-0.75)

# Withoug standardization
 pnorm(5, mean=3, sd=4) - pnorm(0, mean=3, sd=4)
```

Example–3–3–7.r

```r
## ==========================
## Example 3.3-7 on Page 119
## --------------------------

 tmp = qnorm(0.9772)

 tmp * 6
```

Example–6–1–1.r

```r
## ==========================
## Example 6.1.1 on Page 234
## --------------------------
data = c( 20.5, 20.7, 20.8, 21.0, 21.0, 21.4, 21.5, 22.0, 22.1, 22.5,
          22.6, 22.6, 22.7, 22.7, 22.9, 22.9, 23.1, 23.3, 23.4, 23.5,
          23.6, 23.6, 23.6, 23.9, 24.1, 24.3, 24.5, 24.5, 24.8, 24.8,
          24.9, 24.9, 25.1, 25.1, 25.2, 25.6, 25.8, 25.9, 26.1, 26.7 )

# Make tally table
#  Breaks = c(20.45, 23.35, ...
 Breaks = seq(20.45, 26.75, by=0.9)

 table( cut(data, breaks=Breaks ) )
```

Example–6–1–3.r

```r
## ==========================
## Example 6.1-3 on Page 238
## --------------------------

data = c(0.98, 0.92, 0.89, 0.90, 0.94, 0.99,
0.86, 0.85, 1.06, 1.01, 1.03, 0.85, 0.95, 0.90, 1.03,
0.87, 1.02, 0.88, 0.92, 0.88, 0.88, 0.90, 0.98, 0.96,
0.98, 0.93, 0.98, 0.92, 1.00, 0.95, 0.88, 0.90, 1.01,
0.98, 0.85, 0.91, 0.95, 1.01, 0.88, 0.89, 0.99, 0.95,
0.90, 0.88, 0.92, 0.89, 0.90, 0.95, 0.93, 0.96, 0.93,
0.91, 0.92, 0.86, 0.87, 0.91, 0.89, 0.93, 0.93, 0.95,
0.92, 0.88, 0.87, 0.98, 0.98, 0.91, 0.93, 1.00, 0.90,
0.93, 0.89, 0.97, 0.98, 0.91, 0.88, 0.89, 1.00, 0.93,
0.92, 0.97, 0.97, 0.91, 0.85, 0.92, 0.87, 0.86, 0.91,
0.92, 0.95, 0.97, 0.88, 1.05, 0.91, 0.89, 0.92, 0.94,
0.90, 1.00, 0.90, 0.93)

## R determines class intervals
hist(data)   ## frequency
hist(data, prob=TRUE) ## density

## You can decide the class intervals
## The following will give a similar picture as in the textbook.
Breaks = c(0.835, 0.865, 0.895, 0.925, 0.955, 0.985, 1.015, 1.045, 1.075)

hist(data, breaks=Breaks)  ## similar to the textbook (Example 6.1.3).

hist(data, breaks=Breaks, prob=TRUE) ## the same as the textbook.

# ---------------------------------
# Table 6.1-4
table( cut(data, breaks=Breaks ) )
```

Example–6–1–5.r

```r
## ==========================
## Example 6.1-5 on Page 241
## --------------------------

data = c(
30, 17, 65,  8, 38, 35,  4, 19,   7, 14, 12,  4,  5,  4,  2,
 7,  5, 12, 50, 33, 10, 15,  2,  10,  1,  5, 30, 41, 21, 31,
 1, 18, 12,  5, 24,  7,  6, 31,   1,  3,  2,  2,  1, 30,  2,
 1,  3, 12, 12,  9, 28,  6, 50,  63,  5, 17, 11, 23,  2, 46,
```

```
90, 13, 21, 55, 43,  5, 19, 47,  24,  4,  6, 27,  4,  6, 37,
16, 41, 68,  9,  5, 28, 42,  3,  42,  8, 52,  2, 11, 41,  4,
35, 21,  3, 17, 10, 16,  1, 68, 105, 45, 23,  5, 10, 12, 17
)

# The above needs comma (,) but the below does not.

x <- scan()
30 17 65  8 38 35  4 19   7 14 12  4  5  4  2
 7  5 12 50 33 10 15  2  10  1  5 30 41 21 31
 1 18 12  5 24  7  6 31   1  3  2  2  1 30  2
 1  3 12 12  9 28  6 50  63  5 17 11 23  2 46
90 13 21 55 43  5 19 47  24  4  6 27  4  6 37
16 41 68  9  5 28 42  3  42  8 52  2 11 41  4
35 21  3 17 10 16  1 68 105 45 23  5 10 12 17

# Figure 6.1-4 (a): PDF
hist(x) # frequencey

hist(x, prob=TRUE)  # relative frequency
# The above is slightly different from the textbook (Figure 6.1-4 (a)).

# Let's change intervals
intervals = seq(0,108, by=9)
hist(x, breaks=intervals, prob=TRUE)  # relative frequency
curve( (1/20)*exp(-x/20), 0, 108, add=TRUE, col="blue")


# Figure 6.1-4 (b): CDF
Fn = ecdf(x)
plot(Fn)
curve( 1- exp(-x/20), 0, 108, add=TRUE, col="red")
```

Example–6–2–2.r

```
## ==========================
## Example 6.2-2 on Page 250
## --------------------------
stem(data)

stem(data, scale=0.75)
```

Example–6–2–3.r

```
## ==========================
## Example 6.2-3 on Page 251
## --------------------------

## Data Set from Table 6.1-3 on Page 238
data = c(0.98, 0.92, 0.89, 0.90, 0.94, 0.99,
0.86, 0.85, 1.06, 1.01, 1.03, 0.85, 0.95, 0.90, 1.03,
0.87, 1.02, 0.88, 0.92, 0.88, 0.88, 0.90, 0.98, 0.96,
0.98, 0.93, 0.98, 0.92, 1.00, 0.95, 0.88, 0.90, 1.01,
0.98, 0.85, 0.91, 0.95, 1.01, 0.88, 0.89, 0.99, 0.95,
0.90, 0.88, 0.92, 0.89, 0.90, 0.95, 0.93, 0.96, 0.93,
0.91, 0.92, 0.86, 0.87, 0.91, 0.89, 0.93, 0.93, 0.95,
0.92, 0.88, 0.87, 0.98, 0.98, 0.91, 0.93, 1.00, 0.90,
0.93, 0.89, 0.97, 0.98, 0.91, 0.88, 0.89, 1.00, 0.93,
0.92, 0.97, 0.97, 0.91, 0.85, 0.92, 0.87, 0.86, 0.91,
0.92, 0.95, 0.97, 0.88, 1.05, 0.91, 0.89, 0.92, 0.94,
0.90, 1.00, 0.90, 0.93)

```

```
summary(data)

boxplot(data)

boxplot(data, horizontal=TRUE)

boxplot(data, horizontal=TRUE, notch=TRUE)

median(data)

mean(data)

max(data)

min(data)

range(data)

IQR(data)
```

Example–6–3–3.r

```
## =========================
## Example 6.3-3 on Page 259
## -------------------------

## NOTE: http://integrals.wolfram.com/index.jsp

g1 = function(y) { 10 * y   * (1-y^2)^4 }
g2 = function(y) { 40 * y^3 * (1-y^2)^3 }
g3 = function(y) { 60 * y^5 * (1-y^2)^2 }
g4 = function(y) { 40 * y^7 * (1-y^2) }
g5 = function(y) { 10 * y^9 }


 curve(g1, 0,1)
 curve(g2, 0,1, add=TRUE)
 curve(g3, 0,1, add=TRUE)
 curve(g4, 0,1, add=TRUE)
 curve(g5, 0,1, add=TRUE)

#
 curve(g1, 0,1, ylim=c(0,10) )
 curve(g2, 0,1, add=TRUE)
 curve(g3, 0,1, add=TRUE)
 curve(g4, 0,1, add=TRUE)
 curve(g5, 0,1, add=TRUE)

#
 curve(g1, 0,1, ylim=c(0,10) )
 curve(g2, 0,1, add=TRUE, lty=2)
 curve(g3, 0,1, add=TRUE, lty=3)
 curve(g4, 0,1, add=TRUE, lty=4)
 curve(g5, 0,1, add=TRUE, lty=5)


##------------------------------------------------------------

G1 = function(y) { 1 - (1-y^2)^5 }
G2 = function(y) { y^4 * ( -4*y^6 + 15*y^4 -20*y^2 + 10) }
G3 = function(y) { y^6 * (6*y^4 -15*y^2 +10) }
G4 = function(y) { y^8 * (5 - 4*y^2) }
```

```r
G5 = function(y) y^10

 curve(G1, 0,1)
 curve(G2, 0,1, add=TRUE, col="red")
 curve(G3, 0,1, add=TRUE, col="green")
 curve(G4, 0,1, add=TRUE, col="blue")
 curve(G5, 0,1, add=TRUE, col="grey")
```

Example–6–3–4.r

```r
## ==========================
## Example 6.3-4 on Page 261
## --------------------------

data = c(1013, 1019, 1021, 1024, 1026, 1028,
         1033, 1035, 1039, 1040, 1043, 1047)

median(data)

quantile(data, probs=0.5)

quantile(data, probs=0.25)
quantile(data, probs=0.25, type=6) # type=6 is the textbook method

quantile(data, probs=0.75)
quantile(data, probs=0.75, type=6)

quantile(data, probs=0.60)
quantile(data, probs=0.60, type=6)
```

Example–6–3–5.r

```r
## ==========================
## Example 6.3-5 on Page 262
## --------------------------

data = c(
1.24, 1.36, 1.28, 1.31, 1.35, 1.20, 1.39, 1.35, 1.41, 1.31,
1.28, 1.26, 1.37, 1.49, 1.32, 1.40, 1.33, 1.28, 1.25, 1.39,
1.38, 1.34, 1.40, 1.27, 1.33, 1.36, 1.43, 1.33, 1.29, 1.34 )

n = length(data)

kk = 1:30

yy = sort(data)

pp = kk/ (n+1)

qq = qnorm(pp)


cbind(kk, yy, pp, qq)

plot(yy,qq)

qqnorm(data)
qqline(data)
```

Example–6–4–4.r

```r
## ==========================
## Example 6.4-4 on Page 269
## --------------------------

# Sample size = 4
L = function(theta, x) {
    dunif(x[1],0,theta)*dunif(x[2],0,theta)*dunif(x[3],0,theta)*dunif(x[4],0,theta)
}


# For example, we have

x = c(1.9, 1.8, 1.7, 2.5)

TH = seq(0.1, 5, by=0.1)
plot(TH, L(TH,x), type="l")

# Lexical Scoping
L1 = function(theta) {
    dunif(x[1],0,theta)*dunif(x[2],0,theta)*dunif(x[3],0,theta)*dunif(x[4],0,theta)
}

x = c(1.9, 1.8, 1.7, 2.5)
TH = seq(0.1, 5, by=0.1)
plot(TH, L1(TH), type="l")


#------------------------------
# Sample size = n

L2 = function(theta, x) {
   n = length(x)
   tmp = rep(1, length(theta))
   for ( i in 1:n ) {
       tmp = tmp * dunif(x[i], 0, theta)
   }
   return(tmp)
}


# For example, we have

x = c(1.9, 1.8, 1.7, 2.5, 3.2, 1.1, 1.2, 0.1, 0.9)

TH = seq(0.1, 5, by=0.1)

plot(TH, L2(TH,x), type="l")
```

Example–6–5–1.r

```r
## ==========================
## Example 6.5-1 on Page 278
## --------------------------

x = c( 70, 74, 72, 68, 58, 54, 82, 64, 80, 61 )
y = c( 77, 94, 88, 80, 71, 76, 88, 80, 90, 69 )

n = length(x)

xbar = mean(x)

ybar = mean(y)
```

```

alpha.hat = ybar

beta.hat = ( sum(x*y) - n*xbar*ybar) / ( sum(x*x) - n* xbar^2)

### Using lm() function
### Note y = alpha + beta x unlike the textbook setting: y = alpha + beta(x-xbar).

LM = lm(y~x)

summary(LM)

plot(x,y )
abline(LM)
```

4 Chapter 7 (R programs)

Example–7–1–4.r

```
## ==========================
## Example 7.1-4
## --------------------------
x = c(13.0, 18.5, 16.4, 14.8, 19.4, 17.3, 23.2, 24.9,
      20.8, 19.3, 18.8, 23.1, 15.2, 19.9, 19.1, 18.1,
      25.1, 16.8, 20.4, 17.4, 25.2, 23.1, 15.3, 19.4,
      16.0, 21.7, 15.2, 21.3, 21.5, 16.8, 15.6, 17.6 )

xbar = mean(x)

s2 = var(x)

s = sqrt(var(x))

sd(x)

n = length(x)

alpha = 1-0.95   # 95% CI.

z = qnorm (1-alpha/2)

L = xbar - z * s/sqrt(n)
U = xbar + z * s/sqrt(n)

c(L,U)
```

Example–7–1–5.r

```
## ==========================
## Example 7.1.5 on Page 313
## --------------------------
x = c( 481, 537, 513, 583, 453, 510, 570, 500, 457, 555,
       618, 327, 350, 643, 499, 421, 505, 637, 599, 392 )

xbar = mean(x)

s2 = var(x)

s = sqrt(var(x))

sd(x)

n = length(x)

alpha = 1-0.90   # 90% CI.

t = qt (1-alpha/2, df=n-1)

L = xbar - t * s/sqrt(n)
U = xbar + t * s/sqrt(n)
c(L,U)

#==============================================================
# NOTE:
#  The following methods can not be used for Examples 7.1.3 and 7.1.4
#  because they are based on N(0,1) while Example 7.1.5 is based on t-dist.
```

```
#------------------------------------------------------------


#============================================================
# Usint t.test() function
t.test(x, conf.level=0.90)


#----------------------------------------------------------
# Using lm() function
mylm = lm(x~1)
confint(mylm, level=0.90)
```

Example–7–2–3.r

```
## =========================
## Example 7.2-3 on Page 320
## -------------------------
set.seed(1)

n=6; m=18; sigma2x=1; sigma2y=36

# Calculate the d.f. using Eq. (7.2-1)
r = (sigma2x/n + sigma2y/m)^2 /( 1/(n-1)*(sigma2x/n)^2+1/(m-1)*(sigma2y/m)^2 )

r

N = 500
T = numeric(N)
W = numeric(N)

#----------------------------------------------------
for ( i in 1:N ) {
    x = rnorm(n, 0, sqrt(sigma2x))
    y = rnorm(m, 0, sqrt(sigma2y))
    xbar = mean(x); ybar = mean(y)
    s2x = var(x); s2y = var(y)
    s2p = ( (n-1)*s2x + (m-1)*s2y ) / (n+m-2)
    T[i] = (xbar-ybar) / sqrt( s2p * (1/n + 1/m) )
    W[i] = (xbar-ybar) / sqrt( s2x/n + s2y/m )
}
#----------------------------------------------------



#
# Figure 7.2-1 (a): T(22) quantiles versus T order statistics
#
qt22 = qt( ppoints(N), df=22)
  qqplot(T,qt22, xlim=c(-3,3), ylim=c(-3,3) )
  abline(h=0, v=0, lty=3)
  abline(a=0, b=1, lty=1, col="blue")

hist(T, probability=TRUE, nclass=20)
  pdf = dt( seq(-3,3, by=0.1), df=22 )
  lines(  seq(-3,3, by=0.1), pdf, type="l", col="red", add=TRUE )

#
# Figure 7.2-1 (b): T(19) quantiles versus T order statistics
#
qt19 = qt( ppoints(N), df=19)
  qqplot(W,qt19, xlim=c(-3,3), ylim=c(-3,3) )
  abline(h=0, v=0, lty=3)
```

```
  abline(a=0, b=1, lty=1, col="blue")

hist(W, probability=TRUE, nclass=20)
  pdf = dt( seq(-3,3, by=0.1), df=19 )
  lines(  seq(-3,3, by=0.1), pdf, type="l", col="red", add=TRUE )
```

Example–7–2–4.r

```
## ==========================
## Example 7.2-4 on Page 322
## --------------------------
x = c(0.30, 0.23, 0.41, 0.53, 0.24, 0.36, 0.38, 0.51)
y = c(0.43, 0.32, 0.58, 0.46, 0.27, 0.41, 0.38, 0.61)

t.test(x,y, paired=TRUE)

#------------------------------------
# same as the above
d = x - y
t.test(d)
```

Example–7–3–1.r

```
## ==========================
## Example 7.3.1 on Page 328
## --------------------------
a = 0.1; z0 = qnorm(1-a/2)

y=8; n=40
phat = y/n

L = phat - z0 * sqrt( phat*(1-phat)/n )
U = phat + z0 * sqrt( phat*(1-phat)/n )

c(L, U)

#
# Wilson CI (See EQ. 7.3.4)
#

prop.test(y, n=n, conf.level=0.90, correct=FALSE )


#--------------------------------
y=80; n=400   ###  This is different.
phat = y/n

L = phat - z0 * sqrt( phat*(1-phat)/n )
U = phat + z0 * sqrt( phat*(1-phat)/n )

c(L, U)


#
# Wilson CI (See EQ. 7.3.4)
#

prop.test(y, n=n, conf.level=0.90, correct=FALSE )
```

5 Chapter 8 (R programs)

Example–8–2–1.r

```
#======================================================
# 8.2-1 (very similar to Exercise 7.2-12)
#------------------------------------------------------
x = c( 0.8, 1.8, 1.0, 0.1, 0.9, 1.7, 1.0, 1.4, 0.9, 1.2, 0.5 )
y = c( 1.0, 0.8, 1.6, 2.6, 1.3, 1.1, 2.4, 1.8, 2.5, 1.4, 1.9, 2.0, 1.2)

# It will give Welch's two sample t-test
t.test(x,y, alternative="less")

# It will give traditional  two sample t-test
t.test(x,y, alternative="less", var.equal=TRUE)

# Five number summary  (* can be different from the textbook results *)
summary(x)
summary(y)

# Box-whisker plots (side by side)
id = rep( c("X","Y"), c(length(x), length(y)) )
boxplot( c(x,y) ~ id )   # vertical mode
boxplot( c(x,y) ~ id, horizontal=TRUE )   # horizontal mode

id2 = rep( c("Y","X"), c(length(y), length(x)) )
boxplot( c(y,x) ~ id2, horizontal=TRUE )

id3 = factor( id , levels=c("Y", "X") )
boxplot( c(x,y) ~ id3, horizontal=TRUE )
```

Example–8–2–2.r

```
#======================================================
# 8.2.2 (very similar to 8.2.1 and Exercise 7.2-12)
#------------------------------------------------------
x = c(1071, 1076, 1070, 1083, 1082, 1067, 1078, 1080, 1075, 1084, 1075, 1080)
y = c(1074, 1069, 1075, 1067, 1068, 1079, 1082, 1064, 1070, 1073, 1072, 1075)

# It will give traditional  two sample t-test
t.test(x,y, alternative="two.sided", var.equal=TRUE)

# Five number summary  (* can be different from the textbook results *)
summary(x)
summary(y)

id = rep( c("X","Y"), c(length(x), length(y)) )

id3 = factor( id , levels=c("Y", "X") )
boxplot( c(x,y) ~ id3, horizontal=TRUE )
```

Figure–8–5–2.r

```
## ==========================
## Figure 8.5-2 on Page 402
## --------------------------
# n=25

mu = seq(60, 68, by=0.1)

K1 = 1-pnorm( (62-mu)/2 )
```

```r
K2 = 1-pnorm( (63.29-mu)/2 )


#--------------------------------
plot (mu, K1)
lines(mu, K2)

#--------------------------------
plot (mu, K1, type="l", xlim=c(58,68), ylim=c(0,1), col="blue" )
lines(mu, K2, col="red")

#=====================================================================
# n=100

K3 = 1-pnorm( 61.645-mu )

plot (mu, K1, type="l", xlim=c(58,68), ylim=c(0,1), col="blue" )
lines(mu, K2, col="red")
lines(mu, K3, col="black", lty=2)


#=====================================================================
# Page 404 of Textbook

q1 = qnorm(0.05)
q2 = qnorm(0.975)

n = 4*(q2-q1)^2
n

c = ( 65*q2-60*q1) / (q2-q1)
c
```

Example–8–5–3.r

```r
#====================================================
# Example 8.5.3 on Page 404
#----------------------------------------------------

n = ( sqrt(3)*qnorm(0.90) - 2*qnorm(0.05) )^2

n = ceiling(n)

#--------------------------
c=10.5; n=31; p=1/2
pbinom(c, size=n, prob=p)          # Exact
pnorm( (c-n*p)/sqrt(n*p*(1-p)) )   # Approximate

c=10.5; n=31; p=1/4
pbinom(c, size=n, prob=p)          # Exact
pnorm( (c-n*p)/sqrt(n*p*(1-p)) )   # Approximate

c=11.5; n=32; p=1/2
pbinom(c, size=n, prob=p)          # Exact
pnorm( (c-n*p)/sqrt(n*p*(1-p)) )   # Approximate

c=11.5; n=32; p=1/4
pbinom(c, size=n, prob=p)          # Exact
pnorm( (c-n*p)/sqrt(n*p*(1-p)) )   # Approximate
```

6 Chapter 9 (R programs)

Example–9–1–1.r

```
#======================================================
# Example 9.1.1 on Page 425
#   Test H0: random versus H1: not random
#------------------------------------------------------
data = c(5,8,3,1,9,4,6,7,9,2,6,3,0,
         8,7,5,1,3,6,2,1,9,5,4,8,0,
         3,7,1,4,6,0,4,3,8,2,7,3,9,
         8,5,6,1,8,7,0,3,5,2,5,2)
dist = diff(data)

# Check "SAME"
sum( dist==0 )  ## dangerous
sum( ( dist^2 < 0.0001) )   # better

# Check One away
sum( abs(dist) == 1 )   ## dangerous
sum(  (abs(dist)-1) < 0.00001 )   ## better

# Check Other
sum(  (abs(dist)-1) >= 0.00001   )   ## better

#------------------------------------------------------
y1=0; y2=8; y3=42
p10=1/10; p20=2/10; p30=7/10
n = y1+y2+y3

Q2 = (y1-n*p10)^2 / (n*p10) + (y2-n*p20)^2 / (n*p20) + (y3-n*p30)^2 / (n*p30)

# chi-square critical value
qchisq(1-0.05, df=2)

# Compare Q2 with the above critical value
#  Reject H0

#------------------------------------------------------
# Using R function
chisq.test( x=c(0,8,42),  p=c(1/10, 2/10, 7/10) )


#------------------------------------------------------
# Note
O = c(y1,y2,y3)
E = n*c(p10, p20, p30)

sum( (O-E)^2 / E )
```

Example–9–1–2.r

```
#======================================================
# Example 9.1.2 on Page 426
#   Test H0: Binomial(n=4, p=1/2) versus H1: not Binomial
#------------------------------------------------------

#------------------------------------------------------
O = c( 7, 18, 40, 31, 4)
n = sum(O)
E = n * c(1/16, 4/16, 6/16, 4/16, 1/16)
```

```r

sum( (O-E)^2 / E )

alpha=0.05
qchisq(1-alpha, df=4)

#-------------------------------------------------------
# Using R function
chisq.test( x=c(7,18,40,31,4),  p=c(1/16, 4/16, 6/16, 4/16, 1/16) )
chisq.test( x=c(7,18,40,31,4),  p=dbinom(0:4, size=4,p=1/2) )
```

Example–9–1–3.r

```r
#=======================================================
# Example 9.1.3 on Page 428
#   Test H0: Poisson versus H1: Multinomial
#-------------------------------------------------------
data = c(7, 4, 3, 6, 4, 4, 5, 3, 5, 3,
         5, 5, 3, 2, 5, 4, 3, 3, 7, 6,
         6, 4, 3,11, 9, 6, 7, 4, 5, 4,
         7, 3, 2, 8, 6, 7, 4, 1, 9, 8,
         4, 8, 9, 3, 9, 7, 7, 9, 3,10 )
xbar = mean(data)
n = length(data)

Obs = c(13, 9, 6, 5, 7, 10)
prob =c(sum(dpois(0:3,lambda=xbar)),dpois(4:7,lambda=xbar),1-ppois(7,lambda=xbar))
Exp = n*prob

Q = sum(  (Obs-Exp)^2 / Exp  )
Q

qchisq(1-0.05, df=4)
1-pchisq(Q, df=4)

#------------------------------------------------------
# The below can NOT be used for this test b/c df is wrong.
# But, q (test statistics) can be used.

chisq.test( Obs, p=prob)
```

Example–9–1–4.r

```r
## ==========================
## Example 9.1.4 on Page 430
## --------------------------
# H0: Exponential(theta=20) versus H1: not exponential
#     Note: theta=20 is given.
# Data from Page 241
##-----------------------------------------------------
data = c( 30,17,65, 8,38,35, 4,19,  7,14,12, 4, 5, 4, 2,
           7, 5,12,50,33,10,15, 2, 10, 1, 5,30,41,21,31,
           1,18,12, 5,24, 7, 6,31,  1, 3, 2,22, 1,30, 2,
           1, 3,12,12, 9,28, 6,50, 63, 5,17,11,23, 2,46,
          90,13,21,55,43, 5,19,47, 24, 4, 6,27, 4, 6,37,
          16,41,68, 9, 5,28,42, 3, 42, 8,52, 2,11,41, 4,
          35,21, 3,17,10,16, 1,68,105,45,23, 5,10,12,17 )

# Make tally table
 Breaks = c(0, 9, 18, 27, 36, 45, 54, 63, 72, Inf)
 table( cut(data, breaks=Breaks ) )

```

```r
CDFs = pexp( Breaks, rate=1/20)
Prob.in.class = diff(CDFs)

n = length(data)

O = as.numeric ( table(cut(data, breaks=Breaks ) ) )
E = n*Prob.in.class
cbind( Breaks[-length(Breaks)], Breaks[-1], O, E, Prob.in.class )

tmp = cbind( O, E, Prob.in.class )
rownames(tmp) = names( table(cut(data,breaks=Breaks)) )  # Facelift.
tmp

Q = sum ( (O-E)^2 / E )
Q

df = length(E) - 1
df

qchisq(1-0.05, df=8)

p.value = 1-pchisq(Q, df=8)
p.value

#-----------------------------
chisq.test(O, p=Prob.in.class)   # Warning message due to small values in E.

#=====================================================
# Same problem but theta is NOT given.
# H0: Exponential(theta) versus H1: not exponential
#     Note: theta is NOT given.
#-----------------------------------------------------
xbar = mean(data)
xbar

CDFs = pexp( Breaks, rate=1/xbar)  # Different from the above.
Prob.in.class = diff(CDFs)

E = n*Prob.in.class   # Note: O is the same becuase these are observations.

tmp2 = cbind( O, E, Prob.in.class )
rownames(tmp2) = names( table(cut(data,breaks=Breaks)) )  # Facelift.
tmp2   # Slightly different from the above.

Q2 = sum ( (O-E)^2 / E )
Q2

df2 = length(E) - 1 - 1 # Due to the parameter estimation under H0
df2

qchisq(1-0.05, df=7)   # Be careful. df=7

p.value2 = 1-pchisq(Q, df=7)
p.value2

#-----------------------------
# The following can be used only for Q.
#  Not for df or p-value.
chisq.test(O, p=Prob.in.class)
```

Example–9–2–1.r

```r
#======================================================
# Example 9.2.1 on Page 434
#   Test for Homogeneity
#------------------------------------------------------
Group1 = c(8, 13, 16, 10, 3)
Group2 = c(4,  9, 14, 16, 7)

Data = rbind(Group1, Group2)
Data

rownames(Data) = c("Group I", "Group II")
colnames(Data) = c("A", "B", "C", "D", "F")
Data

n1 = sum(Group1); n2 = sum(Group2)
p = (Group1+Group2)/(n1+n2)
E = rbind( n1*p, n2*p )

cbind(Data, E)

colnames(E) = c("A", "B", "C", "D", "F")  # Not needed. Only facelift.
cbind(Data, E)

O = Data # Not need. Only for notational convenience.
X2 = sum( (O-E)^2 / E )
X2

critical.value = qchisq(1-0.05, df=4)
p.value = 1-pchisq(X2, df=4)
p.value

#===============================
# Using R function: chisq.test()
#-------------------------------

# Estimate pi
pi = (Group1+Group2) / (n1+n2)
chisq.test(Data, p=pi, correct=FALSE)

# Even more simple.
chisq.test(Data, correct=FALSE)
```

Example–9–2–2.r

```r
#======================================================
# Example 9.2.2 on Page 436
#   Test for Homogeneity
#------------------------------------------------------
U = c(25, 31, 20, 42, 39, 19, 35, 36, 44, 26,
      38, 31, 29, 41, 43, 36, 28, 31, 25, 38 )
V = c(28, 17, 33, 25, 31, 21, 16, 19, 31, 27,
      23, 19, 25, 22, 29, 32, 24, 20, 34, 26 )

# Make tally table
BrandU = table( cut(U, breaks=c(-Inf, 23.5, 28.5, 34.5, Inf) ) )
BrandV = table( cut(V, breaks=c(-Inf, 23.5, 28.5, 34.5, Inf) ) )
Data = rbind(BrandU, BrandV)
Data

rownames(Data) = c("Braud U", "Bruan V")
colnames(Data) = c("A1", "A2", "A3", "A4")
# Let's follow the textbook Data (not needed tough).
```

```

# Turn off Yates's continuity correction for 2x2 table.
chisq.test(Data, correct=FALSE)
```

Example–9–2–3.r

```
#=======================================================
# Example 9.2.3 on Page 437
#   Test for Indenpendence
#-------------------------------------------------------
Male   = c(21, 16, 145,  2, 6)
Female = c(14,  4, 175, 13, 4)

Data = rbind(Male, Female)
Data

rownames(Data) = c("Male", "Female")
Data

# Turn off Yates's continuity correction for 2x2 table.
chisq.test(Data, correct=FALSE)
```

Tables in Appendix (R programs)

Appendix–B–Table–IV.r

```r
## ====================================
## Table IV on Appendix B on Page 501
## ------------------------------------


## p=fixed

qchisq( 0.010, df=1)
qchisq( 0.010, df=2)
qchisq( 0.010, df=30)

#

tmp = qchisq( 0.010, df=1:30)
round(tmp,3)

## df=fixed

qchisq( 0.010, df=1)
qchisq( 0.025, df=1)
qchisq( 0.050, df=1)

qchisq( c(0.01,0.025,0.05,0.1,0.9,0.95,0.975,0.99) , df=1)

tmp = qchisq( c(0.01,0.025,0.05,0.1,0.9,0.95,0.975,0.99) , df=2)
round(tmp,3)

# ============================================================
# None of p and df are fixed
# ------------------------------------------------------------
p = c(0.01, 0.025, 0.050, 0.10, 0.90, 0.95, 0.975, 0.99)
df = c(1:30,  40, 50, 60, 70, 80)
f <- function(x,y) qchisq(p=y, df=x)
values = outer(df, p, f)
round(values, 3)
# Cosmetic
colnames(values) = p
rownames(values) = df
round(values, 3)
```

Appendix–B–Table–V.r

```r
## ====================================
## Table V on Appendix B on Page 502
## ------------------------------------

# ====================================
# Table Va
# ------------------------------------
z = seq(0, 3.09, by=0.01)
P = pnorm(z)

P2 = matrix(P, ncol=10, byrow=TRUE)
P3 = round(P2, 5)

# ------------------------------------
# Cosmetic
```

```r
colnames(P3) = seq(0, 0.09, by=0.01)
rownames(P3) = seq(0, 3.0,  by=0.1)
# -----------------------------------

z = seq(3, 5.09, by=0.01)
P = pnorm(z)

P2 = matrix(P, ncol=10, byrow=TRUE)
P3 = round(P2, 5)

# -----------------------------------
# Cosmetic
colnames(P3) = seq(0, 0.09, by=0.01)
rownames(P3) = seq(3, 5.0,  by=0.1)






#================================================================
alpha = c(0.4,0.3,0.2,0.1,0.05,0.025,0.02,0.01,0.005,0.001)
Z1 =  qnorm(1-alpha)
Z2 = qnorm(1-alpha/2)

Z = rbind(Z1, Z2)

# Cosmetic
colnames(Z) = alpha
round(Z,3)



# ====================================
# Table Vb
# -----------------------------------

z = seq(0, 3.49, by=0.01)
P = pnorm(z, lower.tail=FALSE)

P2 = matrix(P, ncol=10, byrow=TRUE)
P3 = round(P2, 4)

# -----------------------------------
# Cosmetic
colnames(P3) = seq(0, 0.09, by=0.01)
rownames(P3) = seq(0, 3.4,  by=0.1)
```