

Chapter 1 (R programs)

Example-1-1-1.r

```
1  ## =====
2  ## Example 1.1-1 on Page 13
3  ## (same as Example 1.4-6 on Page 40)
4  ## -----
5
6  # E = a set of outcomes
7  # M = 1:6
8  M = 1:6
9
10 # At least !
11 E = sample(1:6, replace=TRUE)
12 any(E == M)
13
14 # Setup for computer simulation.
15 N = 500
16 f = numeric(N)
17 # =====
18 # Start: Simulation
19 for ( n in 1:N ) {
20   tmp = numeric(n)
21   for ( i in 1:n ) {
22     E = sample(1:6, replace=TRUE)
23     tmp[i] = any(E == M)
24   }
25   f[n] = sum(tmp)
26 }
27 # End: Simulation
28 # =====
29
30 # Print the results,
31 cbind(1:N, f, f/(1:N))
32
33 # Figure 1.1-3 on Page 13
34 plot(1:N, f/(1:N))
35
36 # More cosmetic
37 plot(f/(1:N), type="l", col="blue")
38 plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
39 abline(h=1-(5/6)^6)
```

Example-1-1-2.r

```
1  ## =====
2  ## Example 1.1-2 on Page 14
3  ## -----
4  N = 50000 # sample size
5
6  x = runif(N, min=-2, max=2)
7  y = runif(N, min=-2, max=2)
8
9  Inside = ( abs(x) < 1 ) & ( abs(y) < 1 )
10 sum(Inside) / N
11
12 # Cosmetics
13 # plot(x, y, pch=".")
14 plot( x[Inside], y[Inside], pch=".", col="blue", xlim=c(-2,2), ylim=c(-2,2) )
15 points( x[!Inside], y[!Inside], pch=".", col="red")
```

```

16 rect( -1,-1, 1, 1, border ="green4")
17
18 ## =====
19 ## Example 1: Extra (Obtaining pi=3.14 using a circle)
20 ## -----
21 N = 50000 # sample size
22
23 x = runif(N, min=-1, max=1)
24 y = runif(N, min=-1, max=1)
25
26 Inside = ( x^2 + y^2 < 1 )
27 sum(Inside) / N * 4
28
29 # Cosmetics
30 # plot(x, y, pch=".")
31 plot( x[Inside], y[Inside], pch=".", col="blue", xlim=c(-1,1), ylim=c(-1,1))
32 points( x[!Inside],y[!Inside], pch=".", col="red")
33
34 angle = 2*pi* seq(0,1,length=101)
35 r1 = cos(angle)
36 r2 = sin(angle)
37 lines(r1, r2, col="green4")
38
39 #-----
40 # Convergence
41 #-----
42 ##NN = c(5, 10, 20, 30, 100, 500, 1000, seq(2000, 50000, by=1000) )
43 NN = seq(5, 5000, by=5)
44 n = length(NN)
45 PI = numeric(n)
46
47 for ( i in 1:length(NN) ) {
48     x = runif(NN[i], min=-1, max=1)
49     y = runif(NN[i], min=-1, max=1)
50     Inside = ( x^2 + y^2 < 1 )
51     PI[i] = sum(Inside) / NN[i] * 4
52 }
53
54 plot(NN, PI, type="l", col="blue" )
55 abline(h=3.14, col="red4")
56
57
58 ## =====
59 ## Example 2: Extra (Obtaining pi=3.14 using a ball)
60 ## -----
61 N = 50000 # sample size
62
63 x = runif(N, min=-1, max=1)
64 y = runif(N, min=-1, max=1)
65 z = runif(N, min=-1, max=1)
66
67 Inside = ( x^2 + y^2 + z^2 < 1 )
68 sum(Inside) / N * 6
69
70 #-----
71 # Convergence
72 #-----
73 ##NN = c(5, 10, 20, 30, 100, 500, 1000, seq(2000, 50000, by=1000) )
74 NN = seq(5, 5000, by=5)
75 n = length(NN)
76 PI3 = numeric(n)
77

```

```

78 for ( i in 1:length(NN) ) {
79   x = runif(NN[i], min=-1, max=1)
80   y = runif(NN[i], min=-1, max=1)
81   z = runif(NN[i], min=-1, max=1)
82   Inside = ( x^2 + y^2 + z^2 < 1 )
83   PI3[i] = sum(Inside) / NN[i] * 6
84 }
85
86 plot(NN, PI3, type="l", col="blue" )
87 abline(h=3.14, col="red4")
88
89 #-----
90 plot(NN, PI, type="l", col="blue" )
91 lines(NN, PI3, type="l", col="red" )
92 abline(h=3.14, col="black")

```

Example-1-2-4.r

```

1  ## =====
2  ## Example 1.2-4 on page 21
3  ## -----
4  factorial(26) / factorial(22)
5
6  ## =====
7  ## Example 1.2-5 on page 21
8  ## -----
9  factorial(10) / factorial(6)
10
11 ## =====
12 ## Example Extra
13 ## -----
14
15 factorial(100) / factorial(99)
16
17 factorial(200) / factorial(199)   # Something wrong
18
19 exp( lfactorial(200) - lfactorial(199) ) # Success

```

Example-1-2-9.r

```

1  ## =====
2  ## Example 1.2-9 on page 23
3  ## -----
4  factorial(52) / ( factorial(5) * factorial(47) )
5
6  ## more simple
7  choose(52,5)

```

Example-1-4-6-advanced.r

```

1  ## =====
2  ## Example 1.4-6 on Page 40
3  ## (Recall Example 1.1-1 on Page 13)
4  ## -----
5  date(); now <- proc.time() #####
6
7  # E = a set of outcomes
8  M = 1:6
9
10 # At least!
11 E = sample(1:6, replace=TRUE)

```

```

12 any(E == M)
13
14 # Setup for computer simulation.
15 N = 5000
16 f = numeric(N)
17
18 # =====
19 # Start: Simulation
20 for ( n in 1:N ) {
21     tmp = numeric(n)
22     for ( i in 1:n ) {
23         E = sample(1:6, replace=TRUE)
24         tmp[i] = any(E == M)
25     }
26     f[n] = sum(tmp)
27 }
28 # End: Simulation
29 # =====
30 (proc.time()-now)/3600; #####
31
32 # Print the results,
33 cbind(1:N, f, f/(1:N))
34
35 # Figure 1.1-3 on Page 13
36 plot(1:N, f/(1:N))
37
38 # More cosmetic
39 plot(f/(1:N), type="l", col="blue")
40 plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
41 abline(h=1-(5/6)^6)
42
43 #=====
44 # Very Advanced
45 #-----
46 date(); now <- proc.time() #####
47 N = 5000 # iteration
48 m = 6    # 6 faced die
49
50 f = numeric(N)
51 M = matrix( rep(1:m, N), ncol=m, byrow=TRUE )
52 E = matrix( sample(1:6, size=m*N, replace=TRUE), ncol=m, byrow=TRUE )
53
54 TF1 = (E==M)
55 TF2 = apply( TF1, 1, any )
56 f = cumsum(TF2)
57 (proc.time()-now)/3600; #####
58
59 plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
60 abline(h=1-(5/6)^6)

```

Example-1-4-6.r

```

1
2 ## =====
3 ## Example 1.4-6 on Page 40
4 ## (Recall Example 1.1-1 on Page 13)
5 ## -----
6
7 ## =====
8 ## (i) Using the inclusion-exclusion formula
9 ## -----
10

```

```

11 p = 1/6
12 choose(6,1)*p -
13 choose(6,2)*p^2 +
14 choose(6,3)*p^3 -
15 choose(6,4)*p^4 +
16 choose(6,5)*p^5 -
17 choose(6,6)*p^6
18
19 ## =====
20 ## (ii) Using P(B) = 1 - P(B')
21 ## -----
22 p = 1/6
23
24 1 - (1-p)^6
25
26
27 ## =====
28 ## (iii) Using Simulation
29 ## -----
30
31 # E = a set of outcomes
32 M = 1:6
33
34 # Toss a die.
35 E = sample(1:6, replace=TRUE)
36
37 # Check if there is at least one match.
38 any(E == M)
39
40 # Setup for simulation.
41 N = 1000
42 f = numeric(N)
43 # =====
44 # Start: Simulation
45 for ( n in 1:N ) {
46   tmp = numeric(n)
47   for ( i in 1:n ) {
48     E = sample(1:6, replace=TRUE)
49     tmp[i] = any(E == M)
50   }
51   f[n] = sum(tmp)
52 }
53 # End: Simulation
54 # =====
55
56 # Print the results,
57 cbind(1:N, f, f/(1:N))
58
59 # Figure 1.1-3 on Page 13
60 plot(1:N, f/(1:N))
61
62 # More cosmetic
63 plot(f/(1:N), type="l", col="blue")
64 plot(1:N, f/(1:N), type="l", col="blue", ylim=0:1)
65 abline(h=1-(5/6)^6, col="red")

```

Chapter 2 (R programs)

Example-2-1-7.r

```
1  ## =====
2  ## Example 2.1-7 on Page 54
3  ## -----
4  out = rep( 2:8, c(71,124,194,258,177,122,54) )
5
6  table(out)
7
8  hist(out) # Different from textbook
9
10 # Histogram
11 hist(out, breaks=seq(0.5,8.5,by=1) ) # slightly different
12
13 # Relative frequency histogram
14 hist(out, breaks=seq(0.5,8.5,by=1), prob=T ) #
15
16 #-----
17 # Theoretical pmf
18 x = 2:8
19 pmf = (4-abs(x-5))/16
20
21 # Let's check pmf
22 pmf > 0
23 sum(pmf)
24
25 #-----
26 # Relative frequency histogram with pmf
27 hist(out, breaks=seq(0.5,8.5,by=1), prob=T )
28 lines(x, pmf, type="h", col="red", lwd=10)
29
30 #-----
31 # Table 2.1-1
32 obs = table(out)
33 cbind(obs, obs/length(out), pmf)
34
35 #=====
36 # How is actually the textbook example made
37 #-----
38 N = 1000 # We can also try N=100000
39
40 x1 = sample(1:4, size=N, replace=TRUE)
41 x2 = sample(1:4, size=N, replace=TRUE)
42
43 out = x1+x2
44 hist(out, breaks=seq(0.5,8.5,by=1) ) # N.B: triangle shape.
45
46 # Continuous analogy
47 x1 = runif(N)
48 x2 = runif(N)
49
50 out = x1+x2
51 hist(out)
```

Example-2-4-7.r

```
1  ## =====
2  ## Example 2.4-7 on page 75
3  ## -----
```

```

4
5 # P( X <= 8 )
6 pbinom (8, size=10, prob=0.8)
7
8 # P( X <= 8 ) = 1 - P(X=9) - P(X=10)
9 1-dbinom(9,size=10,prob=0.8)-dbinom(10,size=10,prob=0.8)
10
11 # P( X <= 6 )
12 pbinom (6.0, size=10, prob=0.8)
13 pbinom (6.1, size=10, prob=0.8)
14 pbinom (6.4, size=10, prob=0.8)
15 pbinom (6.7, size=10, prob=0.8)
16 pbinom (6.9, size=10, prob=0.8)
17
18 pbinom (7.0, size=10, prob=0.8)
19
20 # -----
21 # Figure 2.4-2 on Page 77
22 x = 0:10
23 cdf = c(0, pbinom(x, size=10, prob=0.8) )
24 Fx = stepfun(x, cdf)
25 plot(Fx, vertical=FALSE, pch=20, col="blue", ylab="F(x)" )
26
27 # The above is better. But the below is easier.
28 xx = seq(0, 10, by = 0.1)
29 Fxx = pbinom(xx, size=10, prob=0.8)
30 plot(xx, Fxx, vertical=FALSE, type="l")

```

Figure-2-4-1.r

```

1 ## =====
2 ## Figure 2.4-1 on Page 76
3 ## -----
4
5 # Histogram in R is only for a sample.
6
7 xx = 0:16
8 pmf = dbinom(xx, size=16, prob=0.75)
9 plot(xx, pmf, type="h", lwd=10, xlab="Bin(16,0.75)", ylab="f(x)")
10
11 # The above is not bad, but different from the textbook.
12
13 #-----
14 names(pmf) = xx
15 barplot(pmf, col="white", xlab="Bin(16,0.75)", ylab="f(x)")
16
17 #=====
18 # Let's make four plots into one.
19 #-----
20 par ( mfrow=c(2,2) )
21
22 xx = 0:16
23 pmf = dbinom(xx, size=16, prob=0.75)
24 names(pmf) = xx
25 barplot(pmf, col="white", xlab="Bin(16,0.75)", ylab="f(x)")
26
27 xx = 0:16
28 pmf = dbinom(xx, size=16, prob=0.50)
29 names(pmf) = xx
30 barplot(pmf, col="white", xlab="Bin(16,0.50)", ylab="f(x)")
31
32 xx = 0:25

```

```

33 pmf = dbinom(xx, size=16, prob=0.35)
34 names(pmf) = xx
35 barplot(pmf, col="white", xlab="Bin(25,0.35)", ylab="f(x)")
36
37 xx = 0:25
38 pmf = dbinom(xx, size=16, prob=0.20)
39 names(pmf) = xx
40 barplot(pmf, col="white", xlab="Bin(25,0.20)", ylab="f(x)")

```

Example-2-5-4.r

```

1  ## =====
2  ## Example 2.5-4 on page 85
3  ## -----
4
5  #
6  r=1; p=0.25; x = 0:25
7  f = dnbinom(x, size=r, prob=p)
8  plot(x+r, f, type="h", xlim=c(0,25) )
9
10 r=4; p=0.6; x = 0:25
11 f = dnbinom(x, size=r, prob=p)
12 plot(x+r, f, type="h", xlim=c(0,25) )
13
14 r=7; p=0.7; x = 0:25
15 f = dnbinom(x, size=r, prob=p)
16 plot(x+r, f, type="h", xlim=c(0,25) )
17
18 r=15; p=0.7; x = 0:25
19 f = dnbinom(x, size=r, prob=p)
20 plot(x+r, f, type="h", xlim=c(15,35) )
21
22 #-----
23 # Plot the above four into one sheet.
24 #-----
25 par( mfrow=c(2,2))
26
27 # And then repeat the above four plots.

```

Example-2-6-1.r

```

1  ## =====
2  ## Example 2.6-1 on Page 90
3  ## Compare with Table III in Appendix B
4  ## -----
5
6  # -----
7  # P[ X <= 6 ] with lambda = 5
8  xx = 0:6
9  dpois ( xx, lambda=5 )
10
11 sum( dpois ( xx, lambda=5 ) )
12
13 ppois(6, lambda=5)
14
15 # -----
16 # P[ X > 5 ] = 1 - P[ X <= 5 ]
17 1 - ppois(5, lambda=5)
18
19 # Use the upper tail.
20 ppois(5, lambda=5, lower.tail=FALSE)
21

```



```

22 # -----
23 # P[ X = 6 ]
24 dpois(6, lambda=5)
25
26 ppois(6, lambda=5) - ppois(5, lambda=5)

```

Example-2-6-2.r

```

1 ## =====
2 ## Example 2.6-2 on Page 90
3 ## See Figure 2.6-1
4 ## -----
5
6 #=====
7 # Let's make four plots into one.
8 #-----
9 par ( mfrow=c(2,2) )
10
11 xx = 0:6
12 pmf = dpois(xx, lambda=0.7)
13 names(pmf) = xx
14 barplot(pmf, col="white", ylab="f(x)")
15
16 xx = 0:6
17 pmf = dpois(xx, lambda=1.3)
18 names(pmf) = xx
19 barplot(pmf, col="white", ylab="f(x)")
20
21 xx = 0:20
22 pmf = dpois(xx, lambda=6.5)
23 names(pmf) = xx
24 barplot(pmf, col="white", ylab="f(x)")
25
26 xx = 0:20
27 pmf = dpois(xx, lambda=10.5)
28 names(pmf) = xx
29 barplot(pmf, col="white", ylab="f(x)")

```

Example-2-6-5.r

```

1 ## =====
2 ## Example 2.6-5 on Page 92
3 ## Poisson approximation to Binomial
4 ## -----
5
6 xx = 0:3
7 dbinom(xx, size=100, prob=0.02)
8 dpois(xx, lambda=2)
9
10 #-----
11 fbin = dbinom(xx, size=100, prob=0.02)
12 fpoi = dpois(xx, lambda=2)
13
14 rbind(xx, fbin, fpoi)
15
16 #-----
17 sum(fbin)
18 sum(fpoi)

```

Chapter 3 (R programs)

Example-3-2-1.r

```
1 ## =====
2 ## Example 3.2-1 on Page 104
3 ## -----
4
5 # Note: R uses rate paramter instead of theta.
6
7 pexp(18, rate=1/20)
```

Example-3-2-2.r

```
1 ## =====
2 ## Example 3.2-2 on Page 105
3 ## -----
4
5 # Question: Find the median.
6
7 # Note: R uses rate paramter instead of theta.
8 qexp(1/2, rate=1/3)
```

Example-3-2-4.r

```
1 ## =====
2 ## Example 3.2-4 on Page 108
3 ## -----
4
5 # Note: R uses lambda (rate) and theta (scale) as well
6 #       alpha = shape
7 alpha=2
8 theta=2
9 x = 5
10
11 1 - pgamma(x, shape=alpha, scale=theta)
12 pgamma(x, shape=alpha, scale=theta, lower.tail=FALSE)
13
14 # Using ppois
15 ppois(alpha-1, lambda=x/theta)
16
17 # Using chisquare
18 1 - pchisq(x, df=4)
```

Example-3-2-6.r

```
1 ## =====
2 ## Example 3.2-6 on Page 109
3 ## -----
4
5 # Using chi-square distribution
6 pchisq(12.83, df=5) - pchisq(1.145, df=5)
7
8 # Note: we can also calculate it using xgamma.
9 pgamma(12.83, shape=5/2, scale=2) - pgamma(1.145, shape=5/2, scale=2)
```

Example-3-2-9.r

```
1 ## =====
2 ## Example 3.2-9 on Page 110
```

```

3  ## -----
4
5  # Using chi-square distribution
6  # Unit time: minute
7  1 - pchisq(9.390, df=18)
8
9  # Note: we can also calculate it using xgamma.
10 # Unit time: minute
11 1 - pgamma(9.390, shape=18/2, scale=2)
12
13 #-----
14 # Using xgamma.
15 # Unit time: hour
16 1 - pgamma(9.390/60, shape=18/2, scale=2/60)

```

Example-3-3-3.r

```

1  ## =====
2  ## Example 3.3-3 on Page 116
3  ## -----
4
5
6
7  #-----
8  pnorm(1.24, mean=0, sd=1)
9  pnorm(1.24) # The same as the above
10
11
12
13 #-----
14 pnorm(2.37) - pnorm(1.24)
15 pnorm(-1.24) - pnorm(-2.37) # The same as the above
16
17
18
19 #-----
20 1-pnorm(1.24)
21 pnorm(1.24, lower.tail=FALSE) # The same as the above
22
23
24
25 #-----
26 pnorm(-2.14)
27 pnorm(2.14, lower.tail=FALSE)
28
29
30
31 #-----
32 pnorm(0.77) - pnorm(-2.14)

```

Example-3-3-4.r

```

1  ## =====
2  ## Example 3.3-4 on Page 116
3  ## -----
4
5
6  a = qnorm(0.9147)
7  a
8
9
10

```

```
11 b = qnorm(1-0.0526)
12 b
```

Example-3-3-5.r

```
1 ## =====
2 ## Example 3.3-5 on Page 117
3 ## -----
4
5
6 qnorm(1 - 0.0125)
7
8
9
10 qnorm(1 - 0.05)
11
12
13
14 qnorm(1 - 0.025)
```

Example-3-3-6.r

```
1 ## =====
2 ## Example 3.3-6 on Page 118
3 ## -----
4
5
6 # (a) P( 4 < X < 8 )
7 # After standardization
8 pnorm(1.25) - pnorm(0.25)
9
10 # Withoug standardization
11 pnorm(8, mean=3, sd=4) - pnorm(4, mean=3, sd=4)
12
13
14 # (b) P( 0 < X < 5 )
15 # After standardization
16 pnorm(0.5) - pnorm(-0.75)
17
18 # Withoug standardization
19 pnorm(5, mean=3, sd=4) - pnorm(0, mean=3, sd=4)
```

Example-3-3-7.r

```
1 ## =====
2 ## Example 3.3-7 on Page 119
3 ## -----
4
5 tmp = qnorm(0.9772)
6
7 tmp * 6
```

Example-6-1-1.r

```

1  ## =====
2  ## Example 6.1.1 on Page 234
3  ## -----
4  data = c( 20.5, 20.7, 20.8, 21.0, 21.0, 21.4, 21.5, 22.0, 22.1, 22.5,
5           22.6, 22.6, 22.7, 22.7, 22.9, 22.9, 23.1, 23.3, 23.4, 23.5,
6           23.6, 23.6, 23.6, 23.9, 24.1, 24.3, 24.5, 24.5, 24.8, 24.8,
7           24.9, 24.9, 25.1, 25.1, 25.2, 25.6, 25.8, 25.9, 26.1, 26.7 )
8
9  # Make tally table
10 # Breaks = c(20.45, 23.35, ...
11 Breaks = seq(20.45, 26.75, by=0.9)
12
13 table( cut(data, breaks=Breaks ) )

```

Example-6-1-3.r

```

1  ## =====
2  ## Example 6.1-3 on Page 238
3  ## -----
4  data = c(0.98, 0.92, 0.89, 0.90, 0.94, 0.99,
5           0.86, 0.85, 1.06, 1.01, 1.03, 0.85, 0.95, 0.90, 1.03,
6           0.87, 1.02, 0.88, 0.92, 0.88, 0.88, 0.90, 0.98, 0.96,
7           0.98, 0.93, 0.98, 0.92, 1.00, 0.95, 0.88, 0.90, 1.01,
8           0.98, 0.85, 0.91, 0.95, 1.01, 0.88, 0.89, 0.99, 0.95,
9           0.90, 0.88, 0.92, 0.89, 0.90, 0.95, 0.93, 0.96, 0.93,
10          0.91, 0.92, 0.86, 0.87, 0.91, 0.89, 0.93, 0.93, 0.95,
11          0.92, 0.88, 0.87, 0.98, 0.98, 0.91, 0.93, 1.00, 0.90,
12          0.93, 0.89, 0.97, 0.98, 0.91, 0.88, 0.89, 1.00, 0.93,
13          0.92, 0.97, 0.97, 0.91, 0.85, 0.92, 0.87, 0.86, 0.91,
14          0.92, 0.95, 0.97, 0.88, 1.05, 0.91, 0.89, 0.92, 0.94,
15          0.90, 1.00, 0.90, 0.93)
16
17 ## R determines class intervals
18 hist(data)    ## frequency
19 hist(data, prob=TRUE) ## density
20
21 ## You can decide the class intervals
22 ## The following will give a similar picture as in the textbook.
23 Breaks = c(0.835, 0.865, 0.895, 0.925, 0.955, 0.985, 1.015, 1.045, 1.075)
24
25 hist(data, breaks=Breaks) ## similar to the textbook (Example 6.1.3).
26
27 hist(data, breaks=Breaks, prob=TRUE) ## the same as the textbook.
28 # -----
29 # Table 6.1-4
30 table( cut(data, breaks=Breaks ) )

```

Example-6-1-4.r

```

1  ## =====
2  ## Example 6.1-4 on Page 240
3  ## -----
4  data = c( rep(2,12), rep(3,4), rep(4,3), rep(5,4), rep(6,4),
5           8, 8, 9, 15, 17, 22, 23, 24, 24, 25, 27, 32, 43 )
6
7  Breaks = c(1.5, 2.5, 6.5, 29.5, 49.5 )
8
9  # Tabulation
10 freq = table( cut(data, breaks=Breaks ) )
11 cbind(freq)

```

```

12
13 # histogram
14 hist(data, breaks=Breaks, prob=TRUE)

```

Example-6-1-5.r

```

1 ## =====
2 ## Example 6.1-5 on Page 241
3 ## -----
4 data = c(
5 30, 17, 65, 8, 38, 35, 4, 19, 7, 14, 12, 4, 5, 4, 2,
6 7, 5, 12, 50, 33, 10, 15, 2, 10, 1, 5, 30, 41, 21, 31,
7 1, 18, 12, 5, 24, 7, 6, 31, 1, 3, 2, 2, 1, 30, 2,
8 1, 3, 12, 12, 9, 28, 6, 50, 63, 5, 17, 11, 23, 2, 46,
9 90, 13, 21, 55, 43, 5, 19, 47, 24, 4, 6, 27, 4, 6, 37,
10 16, 41, 68, 9, 5, 28, 42, 3, 42, 8, 52, 2, 11, 41, 4,
11 35, 21, 3, 17, 10, 16, 1, 68, 105, 45, 23, 5, 10, 12, 17
12 )
13
14 # The above needs comma (,) but the below does not.
15
16 x <- scan()
17 30 17 65 8 38 35 4 19 7 14 12 4 5 4 2
18 7 5 12 50 33 10 15 2 10 1 5 30 41 21 31
19 1 18 12 5 24 7 6 31 1 3 2 2 1 30 2
20 1 3 12 12 9 28 6 50 63 5 17 11 23 2 46
21 90 13 21 55 43 5 19 47 24 4 6 27 4 6 37
22 16 41 68 9 5 28 42 3 42 8 52 2 11 41 4
23 35 21 3 17 10 16 1 68 105 45 23 5 10 12 17
24
25 # Figure 6.1-4 (a): PDF
26 hist(x) # frequency
27
28 hist(x, prob=TRUE) # relative frequency
29 # The above is slightly different from the textbook (Figure 6.1-4 (a)).
30
31 # Let's change intervals
32 intervals = seq(0,108, by=9)
33 hist(x, breaks=intervals, prob=TRUE) # relative frequency
34 curve( (1/20)*exp(-x/20), 0, 108, add=TRUE, col="blue")
35
36 # Figure 6.1-4 (b): CDF
37 Fn = ecdf(x)
38 plot(Fn)
39 curve( 1- exp(-x/20), 0, 108, add=TRUE, col="red")

```

Example-6-2-2.r

```

1 ## =====
2 ## Example 6.2-2 on Page 250
3 ## -----
4 stem(data)
5
6 stem(data, scale=0.75)

```

Example-6-2-3.r

```

1 ## =====
2 ## Example 6.2-3 on Page 251
3 ## -----
4

```

```

5  ## Data Set from Table 6.1-3 on Page 238
6  data = c(0.98, 0.92, 0.89, 0.90, 0.94, 0.99,
7  0.86, 0.85, 1.06, 1.01, 1.03, 0.85, 0.95, 0.90, 1.03,
8  0.87, 1.02, 0.88, 0.92, 0.88, 0.88, 0.90, 0.98, 0.96,
9  0.98, 0.93, 0.98, 0.92, 1.00, 0.95, 0.88, 0.90, 1.01,
10 0.98, 0.85, 0.91, 0.95, 1.01, 0.88, 0.89, 0.99, 0.95,
11 0.90, 0.88, 0.92, 0.89, 0.90, 0.95, 0.93, 0.96, 0.93,
12 0.91, 0.92, 0.86, 0.87, 0.91, 0.89, 0.93, 0.93, 0.95,
13 0.92, 0.88, 0.87, 0.98, 0.98, 0.91, 0.93, 1.00, 0.90,
14 0.93, 0.89, 0.97, 0.98, 0.91, 0.88, 0.89, 1.00, 0.93,
15 0.92, 0.97, 0.97, 0.91, 0.85, 0.92, 0.87, 0.86, 0.91,
16 0.92, 0.95, 0.97, 0.88, 1.05, 0.91, 0.89, 0.92, 0.94,
17 0.90, 1.00, 0.90, 0.93)
18
19 summary(data)
20
21 boxplot(data)
22
23 boxplot(data, horizontal=TRUE)
24
25 boxplot(data, horizontal=TRUE, notch=TRUE)
26
27 median(data)
28
29 mean(data)
30
31 max(data)
32
33 min(data)
34
35 range(data)
36
37 IQR(data)

```

Example-6-2-4.r

```

1  ## =====
2  ## Example 6.2-4 on Page 251
3  ## -----
4
5  data = c(
6  4.90, 5.06, 5.07, 5.08, 5.15, 5.17, 5.18, 5.19, 5.24, 5.25,
7  5.25, 5.25, 5.25, 5.27, 5.27, 5.27, 5.27, 5.28, 5.28, 5.28,
8  5.29, 5.30, 5.30, 5.30, 5.30, 5.31, 5.31, 5.31, 5.31, 5.31,
9  5.32, 5.32, 5.33, 5.34, 5.35, 5.35, 5.35, 5.36, 5.37 )
10
11
12 summary(data)
13 fivenum(data)
14
15 boxplot(data)
16
17 boxplot(data, horizontal=TRUE)  # A little bit different from the textbook
18
19 boxplot(data, horizontal=TRUE, range=10) # set range=big value

```

Example-6-2-5.r

```

1  ## =====
2  ## Example 6.2-4 on Page 251
3  ## -----
4

```

```

5 data = c(
6 4.90, 5.06, 5.07, 5.08, 5.15, 5.17, 5.18, 5.19, 5.24, 5.25,
7 5.25, 5.25, 5.25, 5.27, 5.27, 5.27, 5.27, 5.28, 5.28, 5.28,
8 5.29, 5.30, 5.30, 5.30, 5.30, 5.31, 5.31, 5.31, 5.31, 5.31,
9 5.32, 5.32, 5.33, 5.34, 5.35, 5.35, 5.35, 5.36, 5.37 )
10
11
12 summary(data)
13 fivenum(data)
14
15 diff( range(data) )
16
17 IQR(data)
18
19
20 boxplot(data)
21
22 boxplot(data, horizontal=TRUE) # See Figure 6.2-3 on Page 253
23
24 boxplot(data, horizontal=TRUE, notch=TRUE)

```

Example-6-3-3.r

```

1 ## =====
2 ## Example 6.3-3 on Page 259
3 ## -----
4
5 ## NOTE: http://integrals.wolfram.com/index.jsp
6
7 g1 = function(y) { 10 * y * (1-y^2)^4 }
8 g2 = function(y) { 40 * y^3 * (1-y^2)^3 }
9 g3 = function(y) { 60 * y^5 * (1-y^2)^2 }
10 g4 = function(y) { 40 * y^7 * (1-y^2) }
11 g5 = function(y) { 10 * y^9 }
12
13
14 curve(g1, 0,1)
15 curve(g2, 0,1, add=TRUE)
16 curve(g3, 0,1, add=TRUE)
17 curve(g4, 0,1, add=TRUE)
18 curve(g5, 0,1, add=TRUE)
19
20 #
21 curve(g1, 0,1, ylim=c(0,10) )
22 curve(g2, 0,1, add=TRUE)
23 curve(g3, 0,1, add=TRUE)
24 curve(g4, 0,1, add=TRUE)
25 curve(g5, 0,1, add=TRUE)
26
27 #
28 curve(g1, 0,1, ylim=c(0,10) )
29 curve(g2, 0,1, add=TRUE, lty=2)
30 curve(g3, 0,1, add=TRUE, lty=3)
31 curve(g4, 0,1, add=TRUE, lty=4)
32 curve(g5, 0,1, add=TRUE, lty=5)
33
34
35 ##-----
36
37 G1 = function(y) { 1 - (1-y^2)^5 }
38 G2 = function(y) { y^4 * ( -4*y^6 + 15*y^4 -20*y^2 + 10) }
39 G3 = function(y) { y^6 * (6*y^4 -15*y^2 +10) }

```



```

40 G4 = function(y) { y^8 * (5 - 4*y^2) }
41 G5 = function(y) y^10
42
43 curve(G1, 0,1)
44 curve(G2, 0,1, add=TRUE, col="red")
45 curve(G3, 0,1, add=TRUE, col="green")
46 curve(G4, 0,1, add=TRUE, col="blue")
47 curve(G5, 0,1, add=TRUE, col="grey")

```

Example-6-3-4.r

```

1  ## =====
2  ## Example 6.3-4 on Page 261
3  ## -----
4
5  data = c(1013, 1019, 1021, 1024, 1026, 1028,
6           1033, 1035, 1039, 1040, 1043, 1047)
7
8  median(data)
9
10 quantile(data, probs=0.5)
11
12 quantile(data, probs=0.25)
13 quantile(data, probs=0.25, type=6) # type=6 is the textbook method
14
15 quantile(data, probs=0.75)
16 quantile(data, probs=0.75, type=6)
17
18 quantile(data, probs=0.60)
19 quantile(data, probs=0.60, type=6)

```

Example-6-3-5.r

```

1  ## =====
2  ## Example 6.3-5 on Page 262
3  ## -----
4
5  data = c(
6    1.24, 1.36, 1.28, 1.31, 1.35, 1.20, 1.39, 1.35, 1.41, 1.31,
7    1.28, 1.26, 1.37, 1.49, 1.32, 1.40, 1.33, 1.28, 1.25, 1.39,
8    1.38, 1.34, 1.40, 1.27, 1.33, 1.36, 1.43, 1.33, 1.29, 1.34 )
9
10 n = length(data)
11
12 kk = 1:30
13
14 yy = sort(data)
15
16 pp = kk/ (n+1)
17
18 qq = qnorm(pp)
19
20
21 cbind(kk, yy, pp, qq)
22
23 plot(yy,qq)
24
25 qqnorm(data)
26 qqline(data)

```

Example-6-4-4.r

```

1  ## =====
2  ## Example 6.4-4 on Page 269
3  ## -----
4
5  # Sample size = 4
6  L = function(theta, x) {
7      dunif(x[1],0,theta)*dunif(x[2],0,theta)*dunif(x[3],0,theta)*dunif(x[4],0,theta)
8  }
9
10
11 # For example, we have
12
13 x = c(1.9, 1.8, 1.7, 2.5)
14
15 TH = seq(0.1, 5, by=0.1)
16 plot(TH, L(TH,x), type="l")
17
18 # Lexical Scoping
19 L1 = function(theta) {
20     dunif(x[1],0,theta)*dunif(x[2],0,theta)*dunif(x[3],0,theta)*dunif(x[4],0,theta)
21 }
22
23 x = c(1.9, 1.8, 1.7, 2.5)
24 TH = seq(0.1, 5, by=0.1)
25 plot(TH, L1(TH), type="l")
26
27
28 #-----
29 # Sample size = n
30
31 L2 = function(theta, x) {
32     n = length(x)
33     tmp = rep(1, length(theta))
34     for ( i in 1:n ) {
35         tmp = tmp * dunif(x[i], 0, theta)
36     }
37     return(tmp)
38 }
39
40
41 # For example, we have
42
43 x = c(1.9, 1.8, 1.7, 2.5, 3.2, 1.1, 1.2, 0.1, 0.9)
44
45 TH = seq(0.1, 5, by=0.1)
46
47 plot(TH, L2(TH,x), type="l")

```

Example-6-5-1.r

```

1  ## =====
2  ## Example 6.5-1 on Page 278
3  ## -----
4  x = c( 70, 74, 72, 68, 58, 54, 82, 64, 80, 61 )
5  y = c( 77, 94, 88, 80, 71, 76, 88, 80, 90, 69 )
6
7  n = length(x)
8
9  xbar = mean(x)
10
11 ybar = mean(y)
12

```

```
13 alpha.hat = ybar
14
15 beta.hat = ( sum(x*y) - n*xbar*ybar) / ( sum(x*x) - n* xbar^2)
16
17 ### Using lm() function
18 ### Note y = alpha + beta x unlike the textbook setting: y = alpha + beta(x-xbar).
19
20 LM = lm(y~x)
21
22 summary(LM)
23
24 plot(x,y )
25 abline(LM)
```

Chapter 7 (R programs)

Example-7-1-4.r

```
1  ## =====
2  ## Example 7.1-4
3  ## -----
4  x = c(13.0, 18.5, 16.4, 14.8, 19.4, 17.3, 23.2, 24.9,
5        20.8, 19.3, 18.8, 23.1, 15.2, 19.9, 19.1, 18.1,
6        25.1, 16.8, 20.4, 17.4, 25.2, 23.1, 15.3, 19.4,
7        16.0, 21.7, 15.2, 21.3, 21.5, 16.8, 15.6, 17.6 )
8
9  xbar = mean(x)
10
11 s2 = var(x)
12
13 s = sqrt(var(x))
14
15 sd(x)
16
17 n = length(x)
18
19 alpha = 1-0.95    # 95% CI.
20
21 z = qnorm (1-alpha/2)
22
23 L = xbar - z * s/sqrt(n)
24 U = xbar + z * s/sqrt(n)
25
26 c(L,U)
```

Example-7-1-5.r

```
1  ## =====
2  ## Example 7.1.5 on Page 313
3  ## -----
4  x = c( 481, 537, 513, 583, 453, 510, 570, 500, 457, 555,
5        618, 327, 350, 643, 499, 421, 505, 637, 599, 392 )
6
7  xbar = mean(x)
8
9  s2 = var(x)
10
11 s = sqrt(var(x))
12
13 sd(x)
14
15 n = length(x)
16
17 alpha = 1-0.90    # 90% CI.
18
19 t = qt (1-alpha/2, df=n-1)
20
21 L = xbar - t * s/sqrt(n)
22 U = xbar + t * s/sqrt(n)
23 c(L,U)
24
25 #=====
26 # NOTE:
27 # The following methods can not be used for Examples 7.1.3 and 7.1.4
28 # because they are based on N(0,1) while Example 7.1.5 is based on t-dist.
```

```

29 #-----
30
31
32 #=====
33 # Usint t.test() function
34 t.test(x, conf.level=0.90)
35
36
37 #-----
38 # Using lm() function
39 mylm = lm(x~1)
40 confint(mylm, level=0.90)

```

Example-7-2-3.r

```

1 ## =====
2 ## Example 7.2-3 on Page 320
3 ## -----
4 set.seed(1)
5
6 n=6; m=18; sigma2x=1; sigma2y=36
7
8 # Calculate the d.f. using Eq. (7.2-1)
9 r = (sigma2x/n + sigma2y/m)^2 / ( 1/(n-1)*(sigma2x/n)^2+1/(m-1)*(sigma2y/m)^2 )
10
11 r
12
13 N = 500
14 T = numeric(N)
15 W = numeric(N)
16
17 #-----
18 for ( i in 1:N ) {
19   x = rnorm(n, 0, sqrt(sigma2x))
20   y = rnorm(m, 0, sqrt(sigma2y))
21   xbar = mean(x); ybar = mean(y)
22   s2x = var(x); s2y = var(y)
23   s2p = ( (n-1)*s2x + (m-1)*s2y ) / (n+m-2)
24   T[i] = (xbar-ybar) / sqrt( s2p * (1/n + 1/m) )
25   W[i] = (xbar-ybar) / sqrt( s2x/n + s2y/m )
26 }
27 #-----
28
29
30 #
31 # Figure 7.2-1 (a): T(22) quantiles versus T order statistics
32 #
33 qt22 = qt( ppoints(N), df=22)
34 qqplot(T,qt22, xlim=c(-3,3), ylim=c(-3,3) )
35 abline(h=0, v=0, lty=3)
36 abline(a=0, b=1, lty=1, col="blue")
37
38 hist(T, probability=TRUE, nclass=20)
39 pdf = dt( seq(-3,3, by=0.1), df=22 )
40 lines( seq(-3,3, by=0.1), pdf, type="l", col="red", add=TRUE )
41
42 #
43 # Figure 7.2-1 (b): T(19) quantiles versus T order statistics
44 #
45 qt19 = qt( ppoints(N), df=19)
46 qqplot(W,qt19, xlim=c(-3,3), ylim=c(-3,3) )
47 abline(h=0, v=0, lty=3)

```

```

48     abline(a=0, b=1, lty=1, col="blue")
49
50 hist(W, probability=TRUE, nclass=20)
51 pdf = dt( seq(-3,3, by=0.1), df=19 )
52 lines( seq(-3,3, by=0.1), pdf, type="l", col="red", add=TRUE )

```

Example-7-2-4.r

```

1  ## =====
2  ## Example 7.2-4 on Page 322
3  ## -----
4  x = c(0.30, 0.23, 0.41, 0.53, 0.24, 0.36, 0.38, 0.51)
5  y = c(0.43, 0.32, 0.58, 0.46, 0.27, 0.41, 0.38, 0.61)
6
7  t.test(x,y, paired=TRUE)
8
9  #-----
10 # same as the above
11 d = x - y
12 t.test(d)

```

Example-7-3-1.r

```

1  ## =====
2  ## Example 7.3.1 on Page 328
3  ## -----
4  a = 0.1; z0 = qnorm(1-a/2)
5
6  y=8; n=40
7  phat = y/n
8
9  L = phat - z0 * sqrt( phat*(1-phat)/n )
10 U = phat + z0 * sqrt( phat*(1-phat)/n )
11
12 c(L, U)
13
14 #
15 # Wilson CI (See EQ. 7.3.4)
16 #
17
18 prop.test(y, n=n, conf.level=0.90, correct=FALSE )
19
20
21 #-----
22 y=80; n=400    ### This is different.
23 phat = y/n
24
25 L = phat - z0 * sqrt( phat*(1-phat)/n )
26 U = phat + z0 * sqrt( phat*(1-phat)/n )
27
28 c(L, U)
29
30
31 #
32 # Wilson CI (See EQ. 7.3.4)
33 #
34
35 prop.test(y, n=n, conf.level=0.90, correct=FALSE )

```

Chapter 8 (R programs)

Example-8-2-1.r

```
1 #=====
2 # 8.2-1 (very similar to Exercise 7.2-12)
3 #-----
4 x = c( 0.8, 1.8, 1.0, 0.1, 0.9, 1.7, 1.0, 1.4, 0.9, 1.2, 0.5 )
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)
6
7 # It will give Welch's two sample t-test
8 t.test(x,y, alternative="less")
9
10 # It will give traditional two sample t-test
11 t.test(x,y, alternative="less", var.equal=TRUE)
12
13 # Five number summary (* can be different from the textbook results *)
14 summary(x)
15 summary(y)
16
17 # Box-whisker plots (side by side)
18 id = rep( c("X","Y"), c(length(x), length(y)) )
19 boxplot( c(x,y) ~ id ) # vertical mode
20 boxplot( c(x,y) ~ id, horizontal=TRUE ) # horizontal mode
21
22 id2 = rep( c("Y","X"), c(length(y), length(x)) )
23 boxplot( c(y,x) ~ id2, horizontal=TRUE )
24
25 id3 = factor( id , levels=c("Y", "X") )
26 boxplot( c(x,y) ~ id3, horizontal=TRUE )
```

Example-8-2-2.r

```
1 #=====
2 # 8.2.2 (very similar to 8.2.1 and Exercise 7.2-12)
3 #-----
4 x = c(1071, 1076, 1070, 1083, 1082, 1067, 1078, 1080, 1075, 1084, 1075, 1080)
5 y = c(1074, 1069, 1075, 1067, 1068, 1079, 1082, 1064, 1070, 1073, 1072, 1075)
6
7 # It will give traditional two sample t-test
8 t.test(x,y, alternative="two.sided", var.equal=TRUE)
9
10 # Five number summary (* can be different from the textbook results *)
11 summary(x)
12 summary(y)
13
14 id = rep( c("X","Y"), c(length(x), length(y)) )
15
16 id3 = factor( id , levels=c("Y", "X") )
17 boxplot( c(x,y) ~ id3, horizontal=TRUE )
```

Example-8-2-3.r

```
1 #=====
2 # 8.2.3 on
3 #-----
4
5 X = c(6.85, 6.6, 6.7, 6.75, 6.75, 6.9, 6.85, 6.9, 6.7, 6.85,
6       6.6, 6.7, 6.75, 6.7, 6.7, 6.7, 6.55, 6.6, 6.95, 6.95,
7       6.8, 6.8, 6.7, 6.75, 6.6, 6.7, 6.65, 6.55, 6.55, 6.6,
8       6.6, 6.7, 6.8, 6.75, 6.6, 6.75, 6.5, 6.75, 6.7, 6.65,
```

```

9         6.7, 6.7, 6.55, 6.65, 6.6, 6.65, 6.6, 6.65, 6.8, 6.6 )
10
11 Y = c(7.1, 7.05, 6.7, 6.75, 6.9, 6.9, 6.65, 6.6, 6.55, 6.55,
12       6.85, 6.9, 6.6, 6.85, 6.95, 7.1, 6.95, 6.9, 7.15, 7.05,
13       6.7, 6.9, 6.85, 6.95, 7.05, 6.75, 6.9, 6.8, 6.7, 6.75,
14       6.9, 6.9, 6.7, 6.7, 6.9, 6.9, 6.7, 6.7, 6.9, 6.95 )
15
16
17 Z = ( mean(X)-mean(Y) ) / sqrt( var(X)/length(X) + var(Y)/length(Y) )
18 p.value = pnorm(Z) # p-value
19 c(Z, p.value)
20
21
22 # Welch Two Sample t-test
23 t.test(X,Y, alternative="less", var.equal=FALSE)

```

Example-8-5-1.r

```

1 #=====
2 # Example 8.5.1 on Page 400
3 #-----
4
5 K = function(p, x, n) {
6   pbinom(x, size=n, prob=p)
7 }
8
9 pp = seq(0, 0.5, length=51)
10
11 K(pp, n=20)
12
13
14 # Figure 8.5-1
15 plot(pp, K(pp,x=6,n=20), type="l" )

```

Example-8-5-2.r

```

1 #=====
2 # Example 8.5.2 on Page 401
3 #-----
4
5 # Xi (i=1,2,...25) are from N(60, 100)
6 # Critical region:  $\bar{X} \geq 62$ .
7
8 # Figure 8.5-2 (theoretical power)
9 K = function(mu) {
10   1 - pnorm( (62-mu)/2 )
11 }
12 MU = seq(60, 68, length=81)
13
14 plot(MU, K(MU), xlim=c(58,68), ylim=c(0,1), type="l" )
15
16
17 # Empirical power (through simulation)
18 ITER = 500
19 n=25; sigma=10; MU = seq(60, 68, length=81)
20 power = numeric(length(MU))
21 for ( j in 1:length(MU) ) {
22   mu = MU[j]
23   for ( i in 1:ITER ) {
24     xx = rnorm(n=n, mean=mu, sd=sigma)
25     if ( mean(xx) >= 62 ) power[j] = power[j] + 1/ITER
26   }
27 }

```



```

27 }
28
29 # Compare the empirical power with the theoretical power
30 plot(MU, K(MU), xlim=c(58,68), ylim=c(0,1), type="l" )
31 lines(MU, power, col="red")

```

Figure-8-5-2.r

```

1  ## =====
2  ## Figure 8.5-2 on Page 402
3  ## -----
4  # n=25
5
6  mu = seq(60, 68, by=0.1)
7
8  K1 = 1-pnorm( (62-mu)/2 )
9  K2 = 1-pnorm( (63.29-mu)/2 )
10
11
12 #-----
13 plot (mu, K1)
14 lines(mu, K2)
15
16 #-----
17 plot (mu, K1, type="l", xlim=c(58,68), ylim=c(0,1), col="blue" )
18 lines(mu, K2, col="red")
19
20 #=====
21 # n=100
22
23 K3 = 1-pnorm( 61.645-mu )
24
25 plot (mu, K1, type="l", xlim=c(58,68), ylim=c(0,1), col="blue" )
26 lines(mu, K2, col="red")
27 lines(mu, K3, col="black", lty=2)
28
29
30 #=====
31 # Page 404 of Textbook
32
33 q1 = qnorm(0.05)
34 q2 = qnorm(0.975)
35
36 n = 4*(q2-q1)^2
37 n
38
39 c = ( 65*q2-60*q1) / (q2-q1)
40 c

```

Example-8-5-3.r

```

1  #=====
2  # Example 8.5.3 on Page 404
3  #-----
4
5  n = ( sqrt(3)*qnorm(0.90) - 2*qnorm(0.05) )^2
6
7  n = ceiling(n)
8
9  #-----
10 c=10.5; n=31; p=1/2
11 pbinom(c, size=n, prob=p)          # Exact

```

```
12 pnorm( (c-n*p)/sqrt(n*p*(1-p)) ) # Approximate
13
14 c=10.5; n=31; p=1/4
15 pbinom(c, size=n, prob=p) # Exact
16 pnorm( (c-n*p)/sqrt(n*p*(1-p)) ) # Approximate
17
18 c=11.5; n=32; p=1/2
19 pbinom(c, size=n, prob=p) # Exact
20 pnorm( (c-n*p)/sqrt(n*p*(1-p)) ) # Approximate
21
22 c=11.5; n=32; p=1/4
23 pbinom(c, size=n, prob=p) # Exact
24 pnorm( (c-n*p)/sqrt(n*p*(1-p)) ) # Approximate
```

Chapter 9 (R programs)

Example-9-1-1.r

```
1 #=====
2 # Example 9.1.1 on Page 425
3 # Test H0: random versus H1: not random
4 #-----
5 data = c(5,8,3,1,9,4,6,7,9,2,6,3,0,
6          8,7,5,1,3,6,2,1,9,5,4,8,0,
7          3,7,1,4,6,0,4,3,8,2,7,3,9,
8          8,5,6,1,8,7,0,3,5,2,5,2)
9 dist = diff(data)
10
11 # Check "SAME"
12 sum( dist==0 ) ## dangerous
13 sum( ( dist^2 < 0.0001) ) # better
14
15 # Check One away
16 sum( abs(dist) == 1 ) ## dangerous
17 sum( (abs(dist)-1) < 0.00001 ) ## better
18
19 # Check Other
20 sum( (abs(dist)-1) >= 0.00001 ) ## better
21
22 #-----
23 y1=0; y2=8; y3=42
24 p10=1/10; p20=2/10; p30=7/10
25 n = y1+y2+y3
26
27 Q2 = (y1-n*p10)^2 / (n*p10) + (y2-n*p20)^2 / (n*p20) + (y3-n*p30)^2 / (n*p30)
28
29 # chi-square critical value
30 qchisq(1-0.05, df=2)
31
32 # Compare Q2 with the above critical value
33 # Reject H0
34
35 #-----
36 # Using R function
37 chisq.test( x=c(0,8,42), p=c(1/10, 2/10, 7/10) )
38
39
40 #-----
41 # Note
42 O = c(y1,y2,y3)
43 E = n*c(p10, p20, p30)
44
45 sum( (O-E)^2 / E )
```

Example-9-1-2.r

```
1 #=====
2 # Example 9.1.2 on Page 426
3 # Test H0: Binomial(n=4, p=1/2) versus H1: not Binomial
4 #-----
5
6 #-----
7 O = c( 7, 18, 40, 31, 4)
8 n = sum(O)
9 E = n * c(1/16, 4/16, 6/16, 4/16, 1/16)
```

```

10
11 sum( (O-E)^2 / E )
12
13 alpha=0.05
14 qchisq(1-alpha, df=4)
15
16 #-----
17 # Using R function
18 chisq.test( x=c(7,18,40,31,4), p=c(1/16, 4/16, 6/16, 4/16, 1/16) )
19 chisq.test( x=c(7,18,40,31,4), p=dbinom(0:4, size=4,p=1/2) )

```

Example-9-1-3.r

```

1 #=====
2 # Example 9.1.3 on Page 428
3 # Test H0: Poisson versus H1: Multinomial
4 #-----
5 data = c(7, 4, 3, 6, 4, 4, 5, 3, 5, 3,
6          5, 5, 3, 2, 5, 4, 3, 3, 7, 6,
7          6, 4, 3,11, 9, 6, 7, 4, 5, 4,
8          7, 3, 2, 8, 6, 7, 4, 1, 9, 8,
9          4, 8, 9, 3, 9, 7, 7, 9, 3,10 )
10 xbar = mean(data)
11 n = length(data)
12
13 Obs = c(13, 9, 6, 5, 7, 10)
14 prob = c(sum(dpois(0:3,lambda=xbar)),dpois(4:7,lambda=xbar),1-ppois(7,lambda=xbar))
15 Exp = n*prob
16
17 Q = sum( (Obs-Exp)^2 / Exp )
18 Q
19
20 qchisq(1-0.05, df=4)
21 1-pchisq(Q, df=4)
22
23 #-----
24 # The below can NOT be used for this test b/c df is wrong.
25 # But, q (test statistics) can be used.
26
27 chisq.test( Obs, p=prob)

```

Example-9-1-4.r

```

1 ## =====
2 ## Example 9.1.4 on Page 430
3 ## -----
4 # H0: Exponential(theta=20) versus H1: not exponential
5 # Note: theta=20 is given.
6 # Data from Page 241
7 ##-----
8 data = c( 30,17,65, 8,38,35, 4,19, 7,14,12, 4, 5, 4, 2,
9          7, 5,12,50,33,10,15, 2, 10, 1, 5,30,41,21,31,
10          1,18,12, 5,24, 7, 6,31, 1, 3, 2,22, 1,30, 2,
11          1, 3,12,12, 9,28, 6,50, 63, 5,17,11,23, 2,46,
12          90,13,21,55,43, 5,19,47, 24, 4, 6,27, 4, 6,37,
13          16,41,68, 9, 5,28,42, 3, 42, 8,52, 2,11,41, 4,
14          35,21, 3,17,10,16, 1,68,105,45,23, 5,10,12,17 )
15
16 # Make tally table
17 Breaks = c(0, 9, 18, 27, 36, 45, 54, 63, 72, Inf)
18 table( cut(data, breaks=Breaks) )
19

```

```

20 CDFs = pexp( Breaks, rate=1/20)
21 Prob.in.class = diff(CDFs)
22
23 n = length(data)
24
25 O = as.numeric ( table(cut(data, breaks=Breaks ) ) )
26 E = n*Prob.in.class
27 cbind( Breaks[-length(Breaks)], Breaks[-1], O, E, Prob.in.class )
28
29 tmp = cbind( O, E, Prob.in.class )
30 rownames(tmp) = names( table(cut(data,breaks=Breaks)) ) # Facelift.
31 tmp
32
33 Q = sum ( (O-E)^2 / E )
34 Q
35
36 df = length(E) - 1
37 df
38
39 qchisq(1-0.05, df=8)
40
41 p.value = 1-pchisq(Q, df=8)
42 p.value
43
44 #-----
45 chisq.test(O, p=Prob.in.class) # Warning message due to small values in E.
46
47 #=====
48 # Same problem but theta is NOT given.
49 # H0: Exponential(theta) versus H1: not exponential
50 # Note: theta is NOT given.
51 #-----
52 xbar = mean(data)
53 xbar
54
55 CDFs = pexp( Breaks, rate=1/xbar) # Different from the above.
56 Prob.in.class = diff(CDFs)
57
58 E = n*Prob.in.class # Note: O is the same becuae these are observations.
59
60 tmp2 = cbind( O, E, Prob.in.class )
61 rownames(tmp2) = names( table(cut(data,breaks=Breaks)) ) # Facelift.
62 tmp2 # Slightly different from the above.
63
64 Q2 = sum ( (O-E)^2 / E )
65 Q2
66
67 df2 = length(E) - 1 - 1 # Due to the parameter estimation under H0
68 df2
69
70 qchisq(1-0.05, df=7) # Be careful. df=7
71
72 p.value2 = 1-pchisq(Q, df=7)
73 p.value2
74
75 #-----
76 # The following can be used only for Q.
77 # Not for df or p-value.
78 chisq.test(O, p=Prob.in.class)

```

```

1 #=====
2 # Example 9.2.1 on Page 434
3 # Test for Homogeneity
4 #-----
5 Group1 = c(8, 13, 16, 10, 3)
6 Group2 = c(4, 9, 14, 16, 7)
7
8 Data = rbind(Group1, Group2)
9 Data
10
11 rownames(Data) = c("Group I", "Group II")
12 colnames(Data) = c("A", "B", "C", "D", "F")
13 Data
14
15 n1 = sum(Group1); n2 = sum(Group2)
16 p = (Group1+Group2)/(n1+n2)
17 E = rbind( n1*p, n2*p )
18
19 cbind(Data, E)
20
21 colnames(E) = c("A", "B", "C", "D", "F") # Not needed. Only facelift.
22 cbind(Data, E)
23
24 O = Data # Not need. Only for notational convenience.
25 X2 = sum( (O-E)^2 / E )
26 X2
27
28 critical.value = qchisq(1-0.05, df=4)
29 p.value = 1-pchisq(X2, df=4)
30 p.value
31
32 #=====
33 # Using R function: chisq.test()
34 #-----
35
36 # Estimate pi
37 pi = (Group1+Group2) / (n1+n2)
38 chisq.test(Data, p=pi, correct=FALSE)
39
40 # Even more simple.
41 chisq.test(Data, correct=FALSE)

```

Example-9-2-2.r

```

1 #=====
2 # Example 9.2.2 on Page 436
3 # Test for Homogeneity
4 #-----
5 U = c(25, 31, 20, 42, 39, 19, 35, 36, 44, 26,
6       38, 31, 29, 41, 43, 36, 28, 31, 25, 38 )
7 V = c(28, 17, 33, 25, 31, 21, 16, 19, 31, 27,
8       23, 19, 25, 22, 29, 32, 24, 20, 34, 26 )
9
10 # Make tally table
11 BrandU = table( cut(U, breaks=c(-Inf, 23.5, 28.5, 34.5, Inf) ) )
12 BrandV = table( cut(V, breaks=c(-Inf, 23.5, 28.5, 34.5, Inf) ) )
13 Data = rbind(BrandU, BrandV)
14 Data
15
16 rownames(Data) = c("Braud U", "Bruan V")
17 colnames(Data) = c("A1", "A2", "A3", "A4")
18 # Let's follow the textbook Data (not needed tough).

```

```

19
20 # Turn off Yates's continuity correction for 2x2 table.
21 chisq.test(Data, correct=FALSE)

```

Example-9-2-3.r

```

1 #=====
2 # Example 9.2.3 on Page 437
3 # Test for Indenpendence
4 #-----
5 Male = c(21, 16, 145, 2, 6)
6 Female = c(14, 4, 175, 13, 4)
7
8 Data = rbind(Male, Female)
9 Data
10
11 rownames(Data) = c("Male", "Female")
12 Data
13
14 # Turn off Yates's continuity correction for 2x2 table.
15 chisq.test(Data, correct=FALSE)

```

Example-9-3-2.r

```

1 #=====
2 # Example 9.3.2 on Page 449
3 #-----
4
5 # Read the data from the URL
6 url =
7   "https://raw.githubusercontent.com/appliedstat/course/master/Statistics/R/Table-9-3-5.txt"
8 mydata = read.table(url, header=TRUE)
9
10 # Wrong version (without as.factor)
11 par(mfrow=c(2,1))
12 boxplot(force ~ position, horizontal=TRUE, data=mydata, xlab="force", ylab="position")
13
14 boxplot(force ~ position, at=rev(1:5), horizontal=TRUE, xlab="force",
15         ylab="position", data=mydata)
16
17 OUT = aov(force ~ position, data=mydata)
18 summary(OUT)
19
20 #
21 # Correct version (with as.factor)
22 #
23 url =
24   "https://raw.githubusercontent.com/appliedstat/course/master/Statistics/R/Table-9-3-5.txt"
25 mydata = read.table(url, header=TRUE)
26 mydata$position = as.factor(mydata$position)
27
28 par(mfrow=c(2,1))
29 boxplot(force ~ position, horizontal=TRUE, data=mydata, xlab="force", ylab="position")
30
31 boxplot(force ~ position, at=rev(1:5), horizontal=T, xlab="force", ylab="position",
32         data=mydata)
33
34 OUT = aov(force ~ position, data=mydata)
35 summary(OUT)

```

```
35 par(mfrow=c(2,2))
36 plot(OUT)
37
38 #
39 # Correct version (with as.factor)
40 #
41 url =
42     "https://raw.githubusercontent.com/appliedstat/course/master/Statistics/R/Table-9-3-5.txt"
43 mydata = read.table(url, header=TRUE)
44 attach(mydata) # For more convenience
45 position = as.factor(position)
46
47 par(mfrow=c(2,1))
48 boxplot(force ~ position, horizontal=TRUE, xlab="force", ylab="position")
49
50 boxplot(force ~ position, at=rev(1:5), horizontal=TRUE, xlab="force", ylab="position")
51
52 OUT = aov(force ~ position)
53 summary(OUT)
54
55 par(mfrow=c(2,2))
56 plot(OUT)
```


Tables in Appendix (R programs)

Appendix-B-Table-IV.r

```
1 ## =====
2 ## Table IV on Appendix B on Page 501
3 ## -----
4
5
6 ## p=fixed
7
8 qchisq( 0.010, df=1)
9 qchisq( 0.010, df=2)
10 qchisq( 0.010, df=30)
11
12 #
13
14 tmp = qchisq( 0.010, df=1:30)
15 round(tmp,3)
16
17 ## df=fixed
18
19 qchisq( 0.010, df=1)
20 qchisq( 0.025, df=1)
21 qchisq( 0.050, df=1)
22
23 qchisq( c(0.01,0.025,0.05,0.1,0.9,0.95,0.975,0.99) , df=1)
24
25 tmp = qchisq( c(0.01,0.025,0.05,0.1,0.9,0.95,0.975,0.99) , df=2)
26 round(tmp,3)
27
28 # =====
29 # None of p and df are fixed
30 # -----
31 p = c(0.01, 0.025, 0.050, 0.10, 0.90, 0.95, 0.975, 0.99)
32 df = c(1:30, 40, 50, 60, 70, 80)
33 f <- function(x,y) qchisq(p=y, df=x)
34 values = outer(df, p, f)
35 round(values, 3)
36 # Cosmetic
37 colnames(values) = p
38 rownames(values) = df
39 round(values, 3)
```

Appendix-B-Table-V.r

```
1 ## =====
2 ## Table V on Appendix B on Page 502
3 ## -----
4
5 # =====
6 # Table Va
7 # -----
8 z = seq(0, 3.09, by=0.01)
9 P = pnorm(z)
10
11 P2 = matrix(P, ncol=10, byrow=TRUE)
12 P3 = round(P2, 5)
13
14 # -----
15 # Cosmetic
```

```

16 colnames(P3) = seq(0, 0.09, by=0.01)
17 rownames(P3) = seq(0, 3.0, by=0.1)
18 # -----
19
20 z = seq(3, 5.09, by=0.01)
21 P = pnorm(z)
22
23 P2 = matrix(P, ncol=10, byrow=TRUE)
24 P3 = round(P2, 5)
25
26 # -----
27 # Cosmetic
28 colnames(P3) = seq(0, 0.09, by=0.01)
29 rownames(P3) = seq(3, 5.0, by=0.1)
30
31
32
33
34
35
36 #=====
37 alpha = c(0.4,0.3,0.2,0.1,0.05,0.025,0.02,0.01,0.005,0.001)
38 Z1 = qnorm(1-alpha)
39 Z2 = qnorm(1-alpha/2)
40
41 Z = rbind(Z1, Z2)
42
43 # Cosmetic
44 colnames(Z) = alpha
45 round(Z,3)
46
47
48
49 # =====
50 # Table Vb
51 # -----
52
53 z = seq(0, 3.49, by=0.01)
54 P = pnorm(z, lower.tail=FALSE)
55
56 P2 = matrix(P, ncol=10, byrow=TRUE)
57 P3 = round(P2, 4)
58
59 # -----
60 # Cosmetic
61 colnames(P3) = seq(0, 0.09, by=0.01)
62 rownames(P3) = seq(0, 3.4, by=0.1)

```