Chapter 20

Computer Montcarlo simulations

Supplementary materials

Urbano Lorenzo-Seva, 0000-0001-5369-3099

September 2026

Box 20.1

Generating a Population of Two Independent Normally Distributed Variables

```
Z <- matrix(rnorm(1000000 * 2), ncol = 2)
```

Box 20.2

Visualizing the Distribution of a Variable

```
hist(Z[,1], main = "Histogram of the First Variable", xlab =
"Value", col = "lightblue", border = "black")
```

Box 20.3

Checking Independence Through Pearson Correlation

```
cor(Z)
```

Box 20.4

Defining the Population Correlation Matrix

```
R<-rbind(c(1,0.60),c(0.60,1))
R
```

Box 20.5

Generating Correlated Variables Using Cholesky Decomposition

```
X<-Z %*% chol(R)
```

Box 20.6

Verifying the Population Correlation of the Simulated Variables

```
cor(X)
```

Box 20.7

Defining Thresholds for the Ordinal Transformation

```
TR <- as.matrix(rbind(c(-1,-1),c(1,1)))
TR
```

Box 20.8

R Function for Transforming Continuous Scores into Ordinal Categories

```
discretize<-function(X, THRES){
  siz <- size(X)
  n <- siz[1]
  siz <- size(THRES)
  k <- siz[1]
  m <- siz[2]
  Y <- matrix(0,n,m)
  for (i in 1:n){
    for (j in 1:m){
      Y[i,j] <- k+1
      for (h in 1:k){
        if (X[i,j] <= THRES[h,j]){
          Y[i,j]=h
          break
        }
      }
    }
  }
  return(Y)
}
```

Box 20.9

Applying the Ordinal Transformation

```
Y <- discretize(X,TR)
```

Box 20.10

Drawing a Frequency Distribution of an Ordinal Variable

```
barplot(table(Y[,1]))
```

Box 20.11

Randomly Sampling Participants from the Simulated Population

```
tmp <- sort(runif(1000000))
J <- tmp[,2]
Yi<-Y[J[1:30],]
```

Box 20.12

Computing Pearson Correlation in the Simulated Sample

```
cor(Yi)
```

Box 20.13

Computing Pearson Correlation in the Population

```
cor(Y)
```

Box 20.14

Estimating the Population Association with Polychoric Correlation

```
polychoric(Y)$R
```

The example.r function runs all the code

```
example<-function(){
Z <- matrix(rnorm(1000000 * 2), ncol = 2)
hist(Z[,1], main = "Histogram of the First Variable", xlab =
"Value", col = "lightblue", border = "black")
cor(Z)
R<-rbind(c(1,0.60),c(0.60,1))
R
X<-Z %*% chol(R)
cor(X)
TR <- as.matrix(rbind(c(-1,-1),c(1,1)))
TR
Y <- discretize(X,TR)
barplot(table(Y[,1]))
tmp <- sort(runif(1000000))
J <- tmp[,2]
Yi<-Y[J[1:30],]
cor(Yi)
cor(Y)
polychoric(Y)$R
}

discretize<-function(X, THRES){
  siz <- size(X)
  n <- siz[1]
  siz <- size(THRES)
  k <- siz[1]
  m <- siz[2]
  Y <- matrix(0,n,m)
  for (i in 1:n){
    for (j in 1:m){
      Y[i,j] <- k+1
      for (h in 1:k){
        if (X[i,j] <= THRES[h,j]){
          Y[i,j]=h
          break
        }
      }
    }
  }
  return(Y)
}
```