

## Tentamen i Programmering i R, 7.5 hp

---

Skrivtid: 14-18  
Hjälpmedel: Valfritt tryckt material (böcker, anteckningar mm)

Jourhavande lärare: Josef Wilzén & Måns Magnusson  
Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt. 16 poäng ger Väl godkänt.

**Skriv dina lösningar i fullständig och läsbar kod.**

**Lösningen skrivs i en körbar textfil med namnet Main.R.**

**Kommentera direkt i Main.R filen när något behöver förklaras eller diskuteras.**

**Eventuella grafer som skapas under tentans gång behöver INTE skickas in för rättning, det räcker med att skicka in den kod som producerar figurerna.**

---

**OBS: Glöm inte att spara din fil ofta! Om R krashar kan kod förloras.**

### 1. Datastrukturer

- (a) Beräkna  $\sin\left(\frac{\pi}{7}\right) + \left|\log_3 \frac{1}{e}\right| + 2^{\frac{1}{2}}$ . **1p**  
**Lösningsförslag:**

```
sin(pi/7) + abs(log(x = (1/exp(1)), base = 3)) + 2^(1/2)

[1] 2.75834
```

- (b) Skapa följande vektorer: **1.5p**

- num**: innehåller talen (0.02, 0.04, 0.06, ..., 1), d.v.s. totalt 50 element.
- myText**: som innehåller följande `c("Mon", "Tue", NA, "Thu", "Fri")` upprepad 10 gånger.
- cosine**: skapas genom formeln  $y = \cos(x)$  där  $x$  är vektorn **num** ovan och  $y$  är vektorn **cosine** som ska skapas.

**Lösningsförslag:**

```
num <- seq(from = 0.02, by = 0.02, length = 50)
myText <- rep(c("Mon", "Tue", NA, "Thu", "Fri"), 10)
cosine <- cos(num)
```

- (c) Skapa en dataframe som du kallar `myDF` och som består av variablerna `num`, `myText` och `cosine`. Lägg till en (logisk) variabel i din dataframe som du kallar `missing` och som anger TRUE om `myText` saknar värde (NA), annars FALSE. Räkna ut summan av variabeln `num` för de observationer som saknar värde i `myText`. **1.5p**

**Lösningsförslag:**

```
myDF <- data.frame(num = num, myText = myText, cosine = cosine)
myDF$check <- is.na(myDF$myText)
sum(myDF$num[myDF$check])

[1] 5.1
```

## 2. Kontrollstrukturer

- (a) Skriv en loop som beräknar och skriver ut resten (modulo) från divisionen mellan heltalen 19 till och med 31 (i täljaren) och talet 10 (i nämnaren). Ex.  $13 \bmod 10 = 3$ . **1p**

**Lösningsförslag:**

```
for (i in 19:31) {
  print(i%10)
}

[1] 9
[1] 0
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 0
[1] 1
```

- (b) Skapa en textvektor med elementen `c('Sat', 'Mon', 'Wed', 'Tue', 'Fri')`. Skriv en ny loop som går igenom denna vektor och skriver ut ett meddelande för varje element. Är det en veckodag ska "Weekday." skrivas ut till skärmen annars ska "Week-end!" skrivas ut. Observera att loopen ska fungera för en godtyckling vektor med veckodagsförkortningar.

**2p**

**Lösningsförslag:**

```
weekdays <- c("Sat", "Mon", "Wed", "Tue", "Fri")
for (day in weekdays) {
  if (day == "Sat" | day == "Sun") {
    print("Week-end!")
  } else {
    print("Weekday.")
  }
}
```

```

    }
  }

[1] "Week-end!"
[1] "Weekday."
[1] "Weekday."
[1] "Weekday."
[1] "Weekday."

```

- (c) Skriv en loop som hittar det **minsta** heltal  $Z > 1000$  som är jämt delbart med 23 och skriv ut detta tal till skärmen. **1p**

**Lösningsförslag:**

```

Z <- 1000
while (Z%%23 != 0) {
  Z <- Z + 1
}
Z

[1] 1012

```

### 3. Funktioner

- (a) En vanlig operation inom statistik är att multiplicera en matris med dess transponat och sedan ta matrisinversen av denna multiplikation, d.v.s.  $B = (X^T X)^{-1}$ . Skapa en funktion som kan ta en godtycklig numerisk matris och genomföra denna operation. Kalla funktionen `myFun(X)` och pröva funktionen på följande matris. **2p**.

$$X = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

**Lösningsförslag:**

```

myFun <- function(X) {
  res <- solve(t(X) %*% X)
  return(res)
}

myFun(2 * diag(5))

      [,1] [,2] [,3] [,4] [,5]
[1,] 0.25 0.00 0.00 0.00 0.00
[2,] 0.00 0.25 0.00 0.00 0.00
[3,] 0.00 0.00 0.25 0.00 0.00
[4,] 0.00 0.00 0.00 0.25 0.00
[5,] 0.00 0.00 0.00 0.00 0.25

```

- (b) Skapa en funktion som du kallar `change` som kan ta en textsträng som argument, argumentet ska heta `Text`. Funktionen ska söka igenom textsträngen efter - (bindestreck) och byta ut detta mot ett " till ". Pröva din funktion på "12.00-13.00". **2p**

**Lösningsförslag:**

```
change <- function(Text) {  
  textVec <- strsplit(x = Text, split = "-")[[1]]  
  res <- paste(textVec, collapse = " till ")  
  return(res)  
}  
Text <- "12.00-13.00"  
change(Text)  
[1] "12.00 till 13.00"
```

#### 4. Statistik och grafik

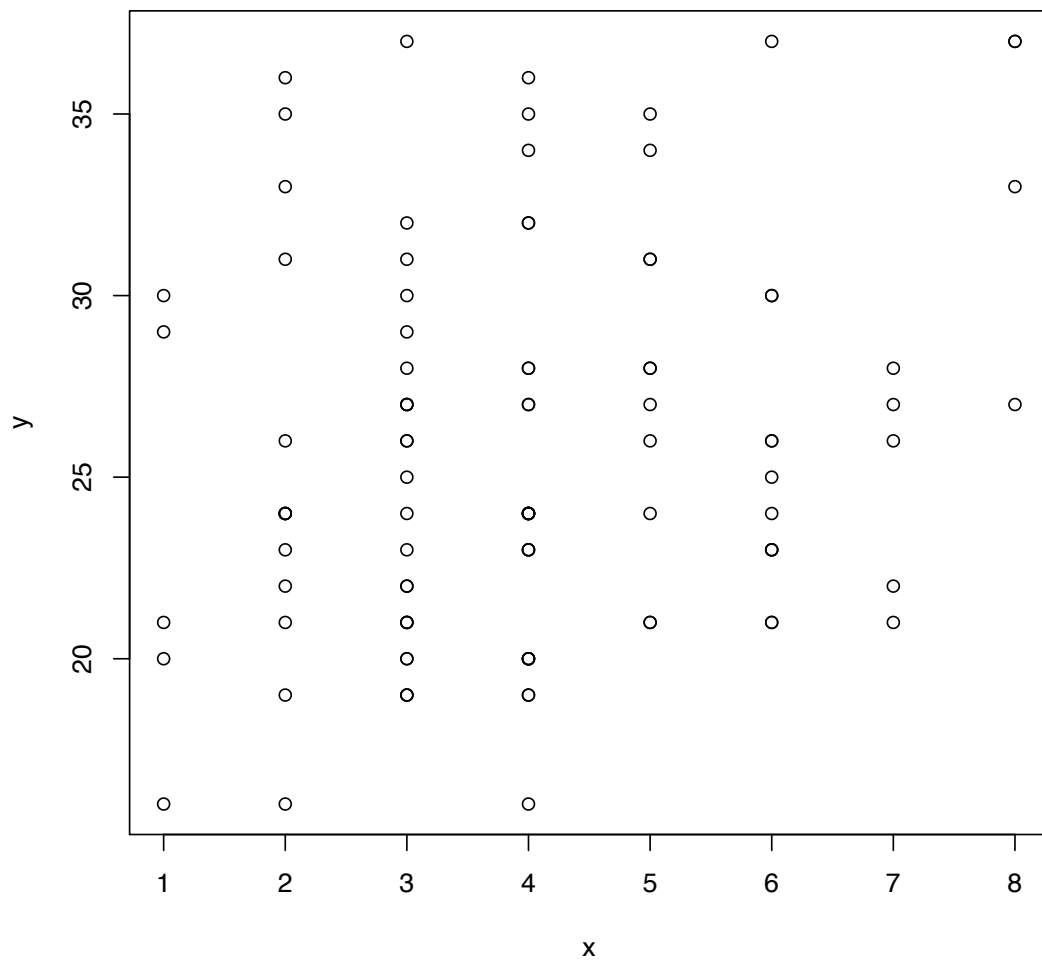
För att det ska vara möjligt att reproducera dina resultat vid rättning behöver du bestämma från vilken seed som slumpalen ska genereras. Kör följande kod i R:

```
set.seed(73233)
```

- (a) Generera/simulera två vektorer,  $x$  och  $y$ , som består av 100 värden från poissonfördelningen med parametern  $\lambda_x = 4$  och en vektor  $\lambda_y = 25$ . Skapa ett scatterplot mellan  $x$  och  $y$ . **1p**

**Lösningsförslag:**

```
x <- rpois(100, 4)  
y <- rpois(100, 25)  
plot(x, y)
```



- (b) Räkna ut medelvärdet och variansen för  $x$  och  $y$ . Beräkna även kovariansen och korrelationen mellan  $x$  och  $y$ . **1p**

**Lösningsförslag:**

```
mean(x)
```

```
[1] 3.99
```

```
mean(y)
```

```
[1] 25.69
```

```
var(x)
```

```
[1] 2.95949
```

```
var(y)
```

```
[1] 27.7918
```

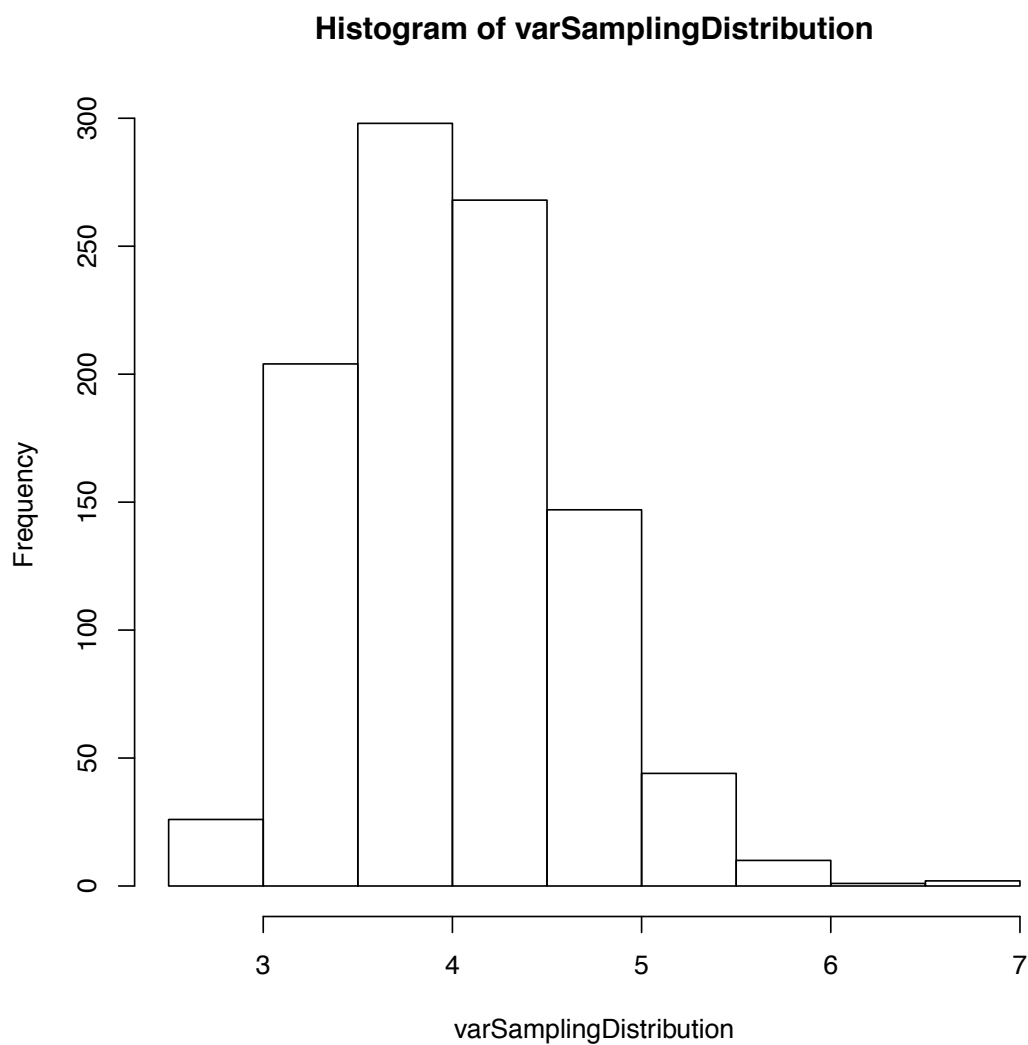
```
cov(x, y)
[1] 1.99687

cor(x, y)
[1] 0.220182
```

- (c) Skapa en tom numerisk vektor som du kallar `varSamplingDistribution` av längd 1000. Gör en loop som i varje steg  $i$  simulerar  $x$  enligt uppgift a) ovan. Därefter ska variansen beräknas för den aktuella simuleringen och sparas som element  $i$  i vektorn `varSamplingDistribution`. Skapa sedan ett histogram över `varSamplingDistribution`. **2p**

**Lösningsförslag:**

```
varSamplingDistribution <- numeric(1000)
for (i in 1:1000) {
  x <- rpois(100, 4)
  varSamplingDistribution[i] <- var(x)
}
hist(varSamplingDistribution)
```



## 5. Strängar och datum

- (a) Läs in paketet `lubridate` i R. Din kompis är född den 5 november 1987, hur många hela veckor hade hon levt den 1 januari 2013? **1p**

**Lösningsförslag:**

```
library(stringr)
library(lubridate)

Error: there is no package called 'lubridate'

new_interval("1987-11-05", "2013-01-01")/weeks(1)

Error: could not find function "new_interval"
```

- (b) Skapa en datumvektor med den 1:a i varje månad under 2013. Skriv ut vektorn på skärmen. **1p**

**Lösningsförslag:**

```
ymd("2013-01-01") + months(0:11)

Error: could not find function "ymd"
```

- (c) Läs in paketet `stringr` i R. Läs in textfilen `BBCarticle201302.txt` i R. Varje mening är ett eget textobjekt. Räkna ut hur många meningar som innehåller landsnamnet “Burma” samt den genomsnittliga teckenlängden i per mening. **2p**

**Lösningsförslag:**

```
BBC <- readLines("BBCarticle201302.txt")
mean(str_length(BBC))

[1] 125.531

sum(str_detect(BBC, "Burma"))

[1] 7
```

*Lycka till!*