

## Tentamen i Programmering i R, 7.5 hp

---

Skrivtid: 14-18  
Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2" av Matt Baggot tillgängligt elektroniskt.  
Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt, 16 poäng ger Väl godkänt.  
Tänk på följande:  
Skriv dina lösningar i **fullständig och läsbar kod**.  
Lösningen skrivs i en körbar textfil med namnet **Main.R**.  
Se filen **DocStudent.pdf** för hur tentan ska lämnas in.  
**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 (4p)

- (a) Beräkna  $\sqrt{\pi} + |\sin(e)|$  **1p**

---

**Lösningsförslag:**

```
sqrt(pi) + abs(sin(exp(1)))  
[1] 2.18324
```

---

- (b) Skapa följande objekt: **1.5p**

- i. Skapa en textvektor, `textVec`, som innehåller elementen "1", "2", "3" o.s.v. upp till "25".

---

**Lösningsförslag:**

```
textVec <- as.character(1:25)
```

---

- ii.  $X$ : en numerisk vektor innehållande talen  $1^2, 2^2, 3^2, \dots, 14^2, 15^2$

---

**Lösningsförslag:**

```
X <- (1:15)^2
```

---

- iii. `myBoolean`: Skapa en logisk vektor som är lika lång som  $X$  och är `TRUE` när  $X$  är större än summan av siffrorna i `textVec` och `FALSE` annars.

---

**Lösningsförslag:**

```
myBoolean <- sum(as.numeric(textVec)) > X
```

---

- (c) Läs in csv-filen `Work_force_data.csv` som till en `data.frame` du kallar `workForce` och gör följande beräkningar: **2.5p**

---

**Lösningsförslag:**

```
workForce <- read.csv2("Work_force_data.csv")
```

---

- i. Vilka år har kvinnor haft en högre arbetslöshetsnivå än männen?

---

**Lösningsförslag:**

```
indicator1 <- workForce[, "Women.Unemployed"] > workForce[, "Men.Unemployed"]
workForce[indicator1, "Year"]
[1] 1975 1976 1977 1980 1987 1988 1989
```

---

- ii. Vad är den genomsnittliga andelen som arbetat heltid bland männen under hela perioden?

---

**Lösningsförslag:**

```
mean(workForce[, "Men.Full.time.35h"])
[1] 78.8465
```

---

- iii. Vad är den genomsnittliga andelen heltidsarbetande kvinnor under 1980-talet?

---

**Lösningsförslag:**

```
numYear <- as.numeric(workForce[, "Year"])
indicator2 <- numYear > 1979 & numYear < 1990
mean(workForce[indicator2, "Women.Full.time.35h"])
[1] 44.1
```

---

## 2. Kontrollstrukturer (4p)

- (a) Skapa en for - loop som itererar över talen 57 till 210 och returnerar X can be divided by 37 för de tal som är delbara med 37 där X är det aktuella talet som är delbart med 7. **2p**

---

**Lösningsförslag:**

```
for (i in 57:210) {
  if (i%%37 == 0)
    cat(i, "can be divided by 37.\n")
}
```

74 can be divided by 37.  
111 can be divided by 37.  
148 can be divided by 37.  
185 can be divided by 37.

---

- (b) Skapa en while - loop som genomför följande beräkning till dess att  $z > 100$ . Vad är  $z$  när loopen avbryts? Starta med  $z = 1$ . **2p**

$$z_i \leftarrow 2 - z_{i-1} + z_{i-1}^2$$

---

**Lösningsförslag:**

```
z <- 1
while (z <= 100) {
  z <- 2 - z + z^2
}
z

[1] 184
```

---

### 3. Strängar och datum (4p)

- (a) Läs in paketen `lubridate` och `stringr` i R. **0.5p**

---

**Lösningsförslag:**

```
library(lubridate)
library(stringr)
```

---

- (b) Ta reda på vilken veckodag det var vid följande datum: **1.5p**

- i. 2:a september 1945 (Andra världskriget tog slut)

---

**Lösningsförslag:**

```
weekdays(ymd("1945-09-02"))
[1] "Sunday"
```

---

- ii. 28 februari 1986 (Då Olof Palme sköts till döds)

---

**Lösningsförslag:**

```
weekdays(ymd("1986-02-28"))
[1] "Friday"
```

---

- iii. 17 december 1919 (Då beslut om allmän och lika rösträtt fattades i Sverige)

---

**Lösningförslag:**

```
weekdays(ymd("1919-12-17"))  
[1] "Wednesday"
```

---

- (c) Läs in textfilen `robot_writer.txt` i R så att varje rad blir ett element i en vektor, döp vektorn till `robotText`.

---

**Lösningförslag:**

```
robotText <- readLines("robot_writer.txt")
```

---

- (d) Räkna ut hur många gånger ord som inleds med `robot` förekommer i texten, oavsett stora eller små bokstäver. Ex. `Robots` ska också räknas. **1.5p**

---

**Lösningförslag:**

```
length(unlist(str_extract_all(robotText, pattern = "[Rr]obot")))  
[1] 2
```

---

#### 4. Funktioner: Statistik och grafik (4p)

- (a) Simulera 30 element från en exponentialfördela variabel med  $E(X) = 1$ , d.v.s. **0.5p**

$$X \sim \text{Exp}(1)$$

---

**Lösningförslag:**

```
X <- rexp(30, 1)
```

---

- (b) Vi ska nu undersöka centrala gränsvärdessatsen. Skapa en funktion du kallar `myCLT(n, noSim, theta)`. Funktionen ska ha argumentet `n`, `noSim` och `theta`. Skapa nu en funktion som drar `n` stycken utfall från en exponentialfördelad variabel med

$$X \sim \text{Exp}(\theta) \text{ där } E(X) = \frac{1}{\theta}$$

Beräkna sedan medelvärdet för dessa `n` värden och spara detta värde. Funktionen ska upprepa denna beräkning `noSim` gånger (d.v.s. simulera `noSim` dragningar av storlek `n`) och returnera en vektor med medelvärden av längd `noSim`. **2p**

*Exempel:*

```
set.seed(20140101)
myCLT(n = 100, noSim = 5, theta = 2)

[1] 0.418907 0.481597 0.437958 0.423957 0.531382
```

---

### Lösningsförslag:

```
myCLT

function(n, noSim, theta){
  resultat <- numeric(noSim)
  for(i in 1:noSim){
    resultat[i] <- mean(rexp(n, theta))
  }
  return(resultat)
}

# Example
set.seed(20140101)
myCLT(n = 100, noSim = 5, theta = 10)

[1] 0.0837815 0.0963194 0.0875917 0.0847913 0.1062764
```

---

- (c) Lägg till ett argument i funktionen som du kallar `statistic` och spara den nya funktionen som `myCLTstatistic(n, noSim, theta, statistic)`. Argumentet ska kunna ta en godtycklig funktion (ex. `mean` eller `median`) och använda denna funktion (statistika) på den vektor av värden som slumpas i varje simulering. På detta sätt kan vi studera samplingfördelningen för en godtycklig statistika. **1p**

*Exempel:*

```
set.seed(20140101)
myCLTstatistic(n = 100, noSim = 5, theta = 2, statistic = median)

[1] 0.323637 0.305810 0.343876 0.270617 0.401163
```

---

### Lösningsförslag:

```
myCLTstatistic

function(n, noSim, theta, statistic){
  resultat <- numeric(noSim)
  for(i in 1:noSim){
    resultat[i] <- statistic(rexp(n, theta))
  }
  return(resultat)
}

# Example
set.seed(20140101)
myCLTstatistic(n = 100, noSim = 5, theta = 10, statistic = median)

[1] 0.0647273 0.0611620 0.0687752 0.0541233 0.0802326
```

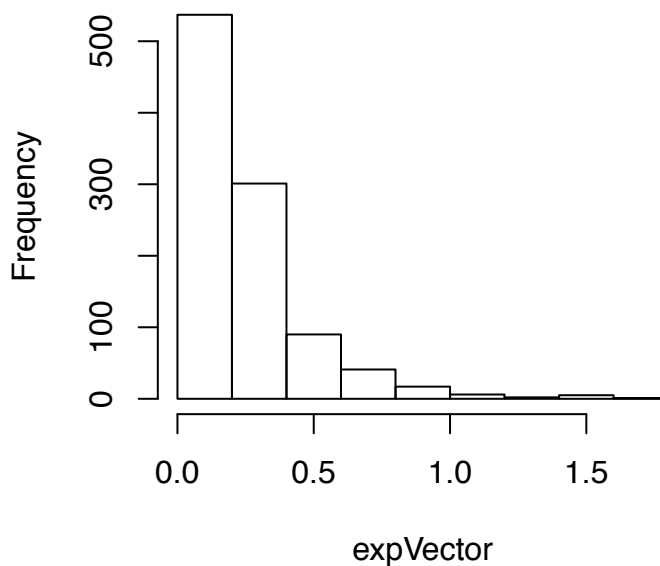
- 
- (d) Skapa ett histogram över de värden som returneras av `myCLTstatistic(n=10, noSim=1000, theta=2, statistic=var)`. Lyckades du inte skapa en vektor i c), simulera en godtycklig vektor av längd 1000 och skapa ett histogram över dessa värden. **0.5p**

---

### Lösningsförslag:

```
expVector <- myCLTstatistic(n = 10, noSim = 1000, theta = 2, statistic = var)
hist(expVector)
```

## Histogram of expVector



### 5. Funktioner: Skevhet och kurtosis (4p)

- (a) Skapa en funktion som baserat på en vektor beräknar urvalets skevhet på följande sätt.

**1.5p**

$$\text{skewness}(\mathbf{x}) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^{3/2}}$$

*Exempel:*

```
skewness(1:100)
```

```
[1] 0
```

### Lösningsförslag:

```
skewness

function(x){
  x_bar <- mean(x)
  n <- length(x)
  taljare <- sum(((x - x_bar)^3))/n
  namnare <- (sum(((x - x_bar)^2))/n)^(3/2)
  return(taljare/namnare)
}
```



```
# Example
set.seed(20140101)
skewness(rexp(100))

[1] 2.1615

skewness(rnorm(100))

[1] 0.086703
```

---

- (b) Skapa en funktion som beräknar urvalets kurtosis (tjocklek på svansarna) **1.5p**

$$\text{kurtosis}(\mathbf{x}) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^2} - 3$$

*Exempel:*

```
kurtosis(1:100)

[1] -1.20024
```

---

### Lösningsförslag:

```
kurtosis

function(x){
  x_bar <- mean(x)
  n <- length(x)
  taljare <- sum((x - x_bar)^4)/n
  namnare <- (sum((x - x_bar)^2)/n)^2
  return(taljare/namnare - 3)
}

# Example
set.seed(20140101)
kurtosis(rexp(100))

[1] 5.8386

kurtosis(rnorm(100))

[1] -0.136457
```

---

- (c) Generera tre vektorer  $\mathbf{x}$  (likformigt fördelad),  $\mathbf{y}$  (normalfördelad) och  $\mathbf{z}$  (exponentialfördelad). Vektorerna skavara 1000 element långa och ha följande parametrar.

$$x \sim \mathcal{U}(0, 1)$$

$$y \sim \mathcal{N}(0, 1)$$

$$z \sim \text{Exp}(1)$$

Beräkna dessa variablers skevhet och kurtosis med funktionerna du skapat ovan.

**1p**

---

**Lösningsförslag:**

```
x <- runif(1000)
y <- rnorm(1000)
z <- rexp(1000)
skewness(x)

[1] 0.0270336

kurtosis(x)

[1] -1.21138

skewness(y)

[1] -0.00722263

kurtosis(y)

[1] 0.100685

skewness(z)

[1] 2.01809

kurtosis(z)

[1] 6.20628
```

---

*Lycka till!*