

Tentamen i Programmering i R, 7.5 hp

Skrivtid: 8-12
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, logik och beräkningar (4p)

(a) Beräkna $|\sin(\pi/3) + \sqrt{3}|$ **1p**

Lösningsförslag:

```
abs(sin(pi/3)+sqrt(3))  
[1] 2.59808
```

(b) Skapa en följande två vektorer **1p**

- i. `txt_vec` med elementen 'polis', 'polis', 'potatismos' upprepade 30 ggr.
- ii. `my_num` som består av talserien 2, 4, 6, 8,...,178,180

Lösningsförslag:

```
txt_vec <- rep(c('polis', 'polis', 'potatismos'), 30)  
my_num <- 2*(1:90)
```

- (c) Skapa en `data.frame` av variablerna ovan med variabelnamnen `txt` och `num`. **1p**

Lösningsförslag:

```
my_data <- data.frame(txt=txt_vec, num=my_num)
```

- (d) Beräkna summan av `num` för de rader som har värdet `'potatismos'`. **1p**

Lösningsförslag:

```
sum(my_data$num[my_data$txt=='potatismos'])  
  
[1] 2790
```

2. Kontrollstrukturer (4p)

- (a) Skapa en `for`-loop som itererar över talen 1 till 100 och beräkna den kumulativa summan av talen. Funktionen ska avbryta funktionen när summan blir större än 200.

Obs! Funktionen måste använda en `for`-loop. **2p**

Lösningsförslag:

```
cum_sum <- 0  
for(i in 1:100){  
  cum_sum <- cum_sum + i  
  if(cum_sum > 200) break  
}  
cum_sum  
  
[1] 210
```

- (b) Upprepa nu a) men med en `while`-loop som ska iterera fram till dess att den kumulativa summan uppgår till mer än 200 **2p**

Obs! Funktionen måste använda en `while`-loop.

Lösningsförslag:

```

cum_sum <- 0
i <- 1
while(cum_sum <= 200){
  cum_sum <- cum_sum + i
  i <- i + 1
}
cum_sum

[1] 210

```

3. Strängar och datum (4p)

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

Lösningsförslag:

```

library(stringr)
library(lubridate)

```

- (b) Weimarrepubliken, den demokratiska republik som störtades av Adolf Hitler bildades den 11 augusti 1919, då en demokratisk och parlamentarisk författning antogs i Weimar och bestod till den 30 januari 1933. Använd `lubridate` för att beräkna följande: **1.5p**
- Vilken veckodag avskaffade Adolf Hitler demokratin i Weimarrepubliken?
 - Hur många veckor bestod Weimarrepubliken?
 - Hur många månader bestod Weimarrepubliken?

Lösningsförslag:

```

begun <- ymd("1919-08-11")
ended <- ymd("1933-01-30")
# i.
wday(ended, label=TRUE)

[1] Mon
Levels: Sun < Mon < Tues < Wed < Thurs < Fri < Sat

# ii.
interval(begun, ended) / weeks(1)

[1] 703

```

```
# iii.
interval(begun, ended) / months(1)

[1] 161
```

- (c) Läs in dikten `hemmingway.txt` i R som poem. **0.5p**

Lösningsförslag:

```
poem <- readLines("hemmingway.txt")
```

- (d) Räkna ut med `stringr` vilka rader som innehåller ordet “old” (med eller utan versaler) och returnera dessa rader ur dikten. **1.5p**

Lösningsförslag:

```
poem[str_detect(poem, "[Oo]ld")]

[1] "Artillery makes the same old noise"
[2] "Old soldiers all have tired eyes"
[3] "All soldiers hear the same old lies"
```

4. Funktioner (4p)

- (a) Skapa en funktion för täthetsfunktionen för en binomialfördelad variabel med argumenten `x`, `n` och `p` kallad `binom_pdf()`. Om $x > n$ eller $x < 0$ ska funktionen avbrytas med `stop()`. **2p**

$$p(x, n, p) = \binom{n}{x} \cdot p^x (1 - p)^{n-x}$$

där

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

```
binom_pdf(3,5,0.5)

[1] 0.3125

binom_pdf(1,6,0.2)

[1] 0.393216
```

Lösningsförslag:

```
function(x,n,p){  
  if(x < 0 | x > n) stop()  
  comb <- (factorial(n) / (factorial(x)*factorial(n-x)))  
  res <- comb * p^x * (1-p)^(n-x)  
  return(res)  
}
```

- (b) Skapa en kumulativ fördelningfunktion, med samma argument som i (a) ovan kan beräknas på följande sätt: **1p**

$$P(X \leq x|n, p) = \sum_{i=0}^x p(x = i, n, p)$$

```
binom_cdf(3,5,0.5)  
[1] 0.8125  
  
binom_cdf(5,5,0.1)  
[1] 1
```

Lösningsförslag:

```
function(x,n,p){  
  res <- 0  
  for(k in 0:x){  
    res <- res + binom_pdf(k,n,p)  
  }  
  return(res)  
}
```

- (c) Beräkna med din funktion $P(X > 5)$ för variabeln $X \sim \mathcal{B}(n = 10, p = 0.5)$. **1p**
-

Lösningsförslag:

```
1 - binom_cdf(5,10,0.5)  
[1] 0.376953
```

5. Statistik och grafik (4p)

- (a) Skapa två vektorer x_1 och x_2 (längd 150) på följande sätt: **1p**

$$\begin{aligned}x_1 &\sim \mathcal{N}(\mu = 3, \sigma = 3) \\x_2 &\sim \mathcal{N}(\mu = 2, \sigma = 1)\end{aligned}$$

Lösningsförslag:

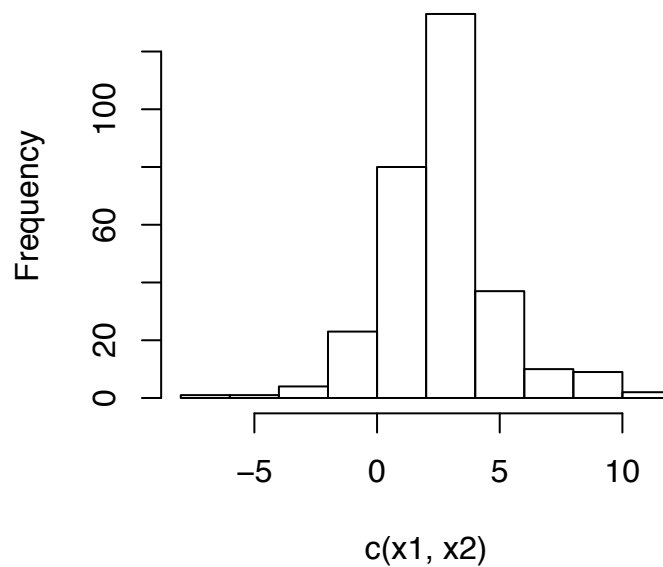
```
x1 <- rnorm(150, 3, 3)
x2 <- rnorm(150, 2, 1)
```

- (b) Visualisera de sammanlagda värdena för x_1 och x_2 i ett histogram. **1p**

Lösningsförslag:

```
hist(c(x1, x2))
```

Histogram of c(x1, x2)



-
- (c) Använd ett t-test för att jämföra om det är en skillnad på de två grupperna x_1 och x_2 . **1p**

Lösningsförslag:

```
t.test(x1, x2)
```

Welch Two Sample t-test

data: x1 and x2

t = 3.1664, df = 185.389, p-value = 0.001805

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

0.315262 1.357484

sample estimates:

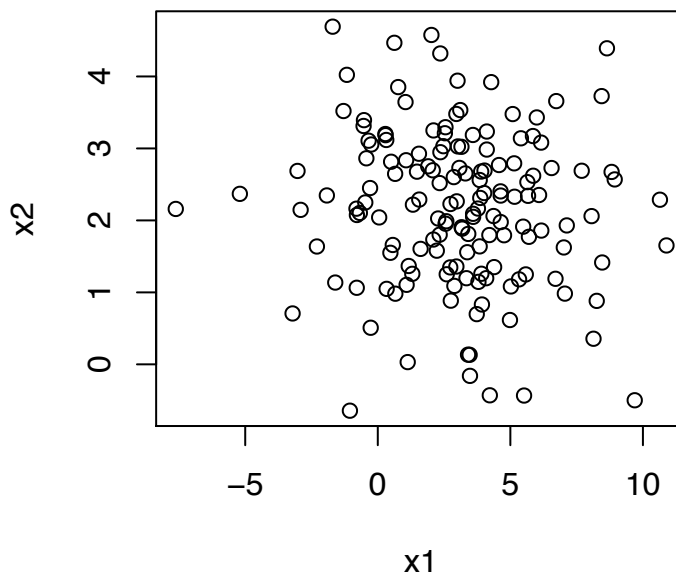
mean of x mean of y

3.01746 2.18109

-
- (d) Visualisera x_1 mot x_2 i en scatterplot. **1p**

Lösningsförslag:

```
plot(x1, x2)
```



Lycka till!