

## 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  $\pi^e + \log_2 3$ . **1p**

**Lösningsförslag:**

```
pi^exp(1) + log(x = 3, base = 2)
[1] 24.0441
```

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

- x**: innehåller talen (2, 4, 6, 8, ..., 50)
- myText**: som innehåller följande `c("Mon", "Tue", "Wed", "Thu", "Fri")` upprepad 5 gånger.
- y**: skapas genom formeln  $y = 10 + \log_e x$  där  $x$  är vektorn **x** ovan.

**Lösningsförslag:**

```
x <- 2 * 1:25
myText <- rep(c("Mon", "Tue", "Wed", "Thu", "Fri"), 5)
y <- 10 + log(x)
```

- (c) Skapa en dataframe som du kallar `myDF` och som består av variablerna `x`, `myText` och `y`. Lägg till en (logisk) variabel i din dataframe som du kallar `check` och som anger `TRUE` om  $x > y$ , annars `FALSE`. Räkna ut i R hur många rader som är `FALSE`. **1.5p**

**Lösningsförslag:**

```
myDF <- data.frame(x = x, myText = myText, y = y)
myDF$check <- x > y
sum(!myDF$check)

[1] 6
```

## 2. Kontrollstrukturer

- (a) Skriv en loop som skriver ut alla heltal mellan 45 och 61 till skärmen. **1p**

**Lösningsförslag:**

```
for (i in 45:61) {
  print(i)
}

[1] 45
[1] 46
[1] 47
[1] 48
[1] 49
[1] 50
[1] 51
[1] 52
[1] 53
[1] 54
[1] 55
[1] 56
[1] 57
[1] 58
[1] 59
[1] 60
[1] 61
```

- (b) Skriv en ny loop som går igenom talen 139 till 177 och i varje iteration undersöker (1) om talet är delbart med 7 - skriv då ut "X can be divided with 7." till skärmen, (2) om talet är delbart med 11 - skriv då ut "X can be divided with 11." till skärmen. Om talet både är delbart med både 7 och 11 ska den skriva ut "X is my favorite number!". Om varken (1) eller (2) är uppfyllt ska den inte skriva ut något till skärmen. **2p**

**Lösningsförslag:**

```
for (i in 139:177) {
  if (i%%7 == 0 & i%%11 == 0) {
    print(paste(i, "is my favorite number!"))
  } else if (i%%7 == 0) {
```

```

    print(paste(i, "can be divided with 7. "))
  } else if (i%%11 == 0) {
    print(paste(i, "can be divided with 11. "))
  }
}

[1] "140 can be divided with 7. "
[1] "143 can be divided with 11. "
[1] "147 can be divided with 7. "
[1] "154 is my favorite number!"
[1] "161 can be divided with 7. "
[1] "165 can be divided with 11. "
[1] "168 can be divided with 7. "
[1] "175 can be divided with 7. "
[1] "176 can be divided with 11. "

```

- (c) Skriv en loop som hittar det **största** heltal  $Z$  som uppfyller olikheten  $\log_e Z \leq 6$  och skriver ut detta till skärmen. Börja med att  $Z = 1000$ . **1p**

**Lösningsförslag:**

```

Z <- 1000
while (log(Z) > 6) {
  Z <- Z - 1
}
Z

[1] 403

```

### 3. Strängar och datum

- (a) Läs in paketen `lubridate` och `stringr` i R. Din kompis är född den 3 maj 1989, vilket datum kommer hon fylla 10000 dagar? **1p**

**Lösningsförslag:**

```

library(stringr)
library(lubridate)
birth <- ymd("1989-05-03")
birth + days(10000)

[1] "2016-09-18 UTC"

```

- (b) Generera en vektor med de datum hon fyller 2000,4000 o.s.v. t.o.m. 10000 dagar. Skriv ut vektorn på skärmen. **1p**

**Lösningsförslag:**

```

birth + days(2000 * (1:5))

[1] "1994-10-24 UTC" "2000-04-15 UTC" "2005-10-06 UTC" "2011-03-29 UTC"
[5] "2016-09-18 UTC"

```

- (c) Läs in textfilen `BBCarticle.txt` i R. Räkna det totala antalet tecken i artikeln. **1p**  
**Lösningssförslag:**

```
BBC <- readLines(con = "BBCarticle.txt")
sum(str_length(BBC))

[1] 4270
```

- (d) Räkna antalet gånger orden "the" förekommer i texten med funktionerna. Alla ord ska räknas, oavsett versaler/gemenor eller om ordet är i början av en mening eller i början av ett citat. Tänk dock på att inte få med ord som "other". **1p**  
**Lösningssförslag:**

```
sum(str_count(BBC, "(^|\\\"| ) [Tt]he($| )"))

[1] 44
```

#### 4. Funktioner

- (a) Talet  $e$  är ett gränsvärde som kan beräknas på följande sätt:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad \text{för alla } x$$

Skapa en egen funktion `myExp()` som beräknar  $e$  upphöjt till valfritt värde  $x$ . I formeln ovan är serien oändlig, men ju även vid en begränsat antal summationer blir approximationen bra. Lägg därför till ett argument  $N$  som anger  $n = 0, 1, 2, \dots, N$ . Jämför med `exp(1)`, hur stort behöver  $N$  vara för att approximationen av  $e$  ska vara samma tal upp till den 5:e decimalen? **2p**.

**Lösningssförslag:**

```
myExp <- function(x, N) {
  xvec <- rep(x, N + 1)
  n <- 0:N
  return(sum(xvec^n/factorial(n)))
}

for (i in 1:10) {
  print(myExp(1, i))
}

[1] 2
[1] 2.5
[1] 2.66667
[1] 2.70833
[1] 2.71667
[1] 2.71806
[1] 2.71825
[1] 2.71828
[1] 2.71828
[1] 2.71828
```

```
exp(1)
[1] 2.71828
```

- (b) Skapa en funktion som kan ta ett personnummer som en textsträng på formatet YYMMDD-NNNN. Om den tredje kontrollsiffran är jämn ska funktionen returnera **Female**, annars returneras **Male**. Pröva din funktion på 811218-9876. **2p**

**Lösningsförslag:**

```
pnr <- function(pnrText) {
  thirdNo <- as.numeric(str_sub(pnrText, start = 10, end = 10))
  if (thirdNo%%2 == 0) {
    return("Female")
  } else {
    return("Male")
  }
}
pnr("811218-9876")
[1] "Male"
```

## 5. 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 en vektor av 1000 värden från weibullfördelningen med parametrarna shape=10 och scale=10. Skapa ett histogram över fördelningen. **1p**

**Lösningsförslag:**

```
hist(rweibull(n = 1000, shape = 10, scale = 10))
```

- (b) Simulera fram en vektor av storlek 400 som du kallar  $x$  från den uniforma fördelning  $[0, 1]$ . Simulera sedan fram en vektor  $y \sim \mathcal{N}(\mu_i, \sigma)$ , normalfördelad med  $\mu_i = 10 \cdot x_i$  och  $\sigma = 1$ , där  $x_i$  är elementen i vektorn  $x$  och  $\mu_i$  är medelvärdet i normalfördelningen. Skapa ett histogram över  $y$ . **1.5p**

**Lösningsförslag:**

```
x <- runif(400)
y <- rnorm(400, x * 10)
hist(y)
```

- (c) Gör en linjär regression med  $y$  som beroende variabel och  $x$  som hjälpvariabel (regressor), d.v.s. modellen  $y = \alpha + \beta x + \epsilon$ . Är  $\alpha$  och  $\beta$  signifikanta? Varför, varför inte? **1.5p**

**Lösningsförslag:**

```
model <- lm(y ~ x)
summary(model)
```

$\alpha$  är inte signifikant, vilket är förväntat då det inte ingår en intercept i modellen.  $\beta$  är signifikant skild från noll, vilket stämmer då det “sanna” värdet på  $\beta$  är 10.

*Lycka till!*