

Tentamen i Programmering i R, 7.5 hp

Skrivtid: 8.00-12.00
Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2" av Matt Baggot, referenskort för ggplot2 och RStudio 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 sparas i en körbar R-fil med namnet **tenta1.R** om du har tenta-ID 1.
Spara din lösning som R-fil i mapppen **Z:\Solutions**
Se filen **DocStudent.pdf** för övrig information om tentamen.
Kommentera direkt i 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 och beräkningar (4p)

- (a) Beräkna $y = \exp(x - 3) \cdot x^{(3 - \log_e(x))}$ där $x = 10$ och \log_e är den naturliga logaritmen. **1p**
- (b) Skapa en lista med 5 tomma element (som alltså innehåller NULL). **0.5p**
- (c) Skapa en data.frame enligt nedan: **1p**

	num	logi	char
1	10	TRUE	a
2	9	FALSE	b
3	8	TRUE	c
4	7	FALSE	d
5	6	TRUE	e
6	5	FALSE	f

7	4	TRUE	g
8	3	FALSE	h
9	2	TRUE	i
10	1	FALSE	j

- (d) Skapa en array enligt nedan. Glöm inte namnen på dimensionerna.
1.5p

```
, , z1

      y1 y2 y3 y4
x1 20 17 14 11
x2 19 16 13 10
x3 18 15 12  9

, , z2

      y1 y2 y3 y4
x1 20 17 14 11
x2 19 16 13 10
x3 18 15 12  9
```

2. Kontrollstrukturer (4p)

- (a) Nu ska en nästlad for-loop skapas, som skriver ut texten enligt nedan. Den yttre loopen ska loopa över vektorn 1,2 och den inre över vektorn 1,2,3,4. **2p**

```
[1] "yttre index 1"
[1] "inre index 1"
[1] "inre index 2"
[1] "inre index 3"
[1] "inre index 4"
[1] "yttre index 2"
[1] "inre index 1"
[1] "inre index 2"
[1] "inre index 3"
[1] "inre index 4"
```

- (b) Skapa en **while**-loop som skriver ut de 5 första heltalen som är jämt delbara med 17, med start på talet 17. **print()** ska användas. **2p**

3. Strängar och datum (4p)

- (a) Läs in paketen `lubridate` och `stringr` i R. Läs in textfilen `R_IDE.txt` i R. Filen innehåller information om olika integrerade utvecklingsmiljöer (IDE) för R. **0.5p**
- (b) Använd funktioner ur `stringr` för att: Skapa en textvektor där varje element innehåller namnet från en IDE, om du har gjort rätt ska den se ut som vektorn nedan. **1p**

```
[1] "Architect"
[2] "DataJoy"
[3] "Deducer"
[4] "Java GUI for R"
[5] "Number Analytics"
[6] "Rattle GUI"
[7] "R Commander"
[8] "Revolution R Productivity Environment"
[9] "RGUI"
[10] "RKWard"
[11] "RStudio"
```

- (c) Använd funktioner ur `lubridate` för att: Räkna ut hur många tisdagar det finns mellan 1992-12-02 och 2016-06-15 (datumen inkluderat). **1p**
- (d) Nu ska funktionen `my_formula(x)` skapas. Funktionen ska ta ett objekt av klassen `lm` (som erhålls vid linjär regression med funktionen `lm()`) och returnera regressionsformlen i text med de skattade regressionskoefficienterna. Det ska kunna vara godtyckligt antal förklarande variabler i modellen: $y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \dots + \beta_p \cdot X_p$. Koefficienterna ska avrundas till två decimaler. Se testfallen nedan för hur funktionen ska fungera. Notera att det ska vara ett "+" mellan alla termer, även om någon koefficient är negativ, se sista testfallet för exempel. **1.5p**

```
data("trees")
a1<-lm(Volume~Girth,data=trees)
my_formula(a1)

[1] "y = -36.94 + 5.07*Girth"

a2<-lm(Volume~Girth+Height,data=trees)
my_formula(a2)

[1] "y = -57.99 + 4.71*Girth + 0.34*Height"
```

```
data("iris")
a3<-lm(Sepal.Length~Sepal.Width+Petal.Length+Petal.Width,data=iris)
b1<-my_formula(a3)
print(b1)

[1] "y = 1.86 + 0.65*Sepal.Width + 0.71*Petal.Length + -0.56*Petal.Width"
```

4. Funktioner:

- (a) Nu ska en funktionen implementeras som beräknar tätheten för en poissonfördelad variabel. Funktionen ska heta `poisson_pdf(n,lambda)`. `n` är en vektor med heltal (som kan vara 0), `lambda` (λ) är parameteren i poissonfördelningen. Det är inte tillåtet att använda funktionerna `dpois()` eller `ppois()` i denna uppgift. Funktionen ska returnera en vektor med sannolikheter. Poissonfördelningen ges av uttrycket nedan. Se testfallen för exempel. Kom ihåg: $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ och $0! = 1$ **1p**

$$P(X = n) = \frac{\exp(-\lambda) \cdot \lambda^n}{n!}$$

```
poisson_pdf(n = 0:3,lambda = 1)

[1] 0.3678794 0.3678794 0.1839397 0.0613132

poisson_pdf(n = 3:5,lambda = 3)

[1] 0.224042 0.168031 0.100819
```

- (b) Implementera funktionen `my_math(x,a=1,b=1)`. Funktionen ska ta en numerisk vektor `x` som argument och beräkna den matematiska funktionen nedan och returnera värdet för $f(x)$. Se testfallen nedan för hur funktionen ska fungera. **1.5p**

$$f(x) = \begin{cases} a \cdot \exp(-x) & x > 0 \\ 1 & x = 0 \\ \sin(b \cdot x)/x & x < 0 \end{cases}$$

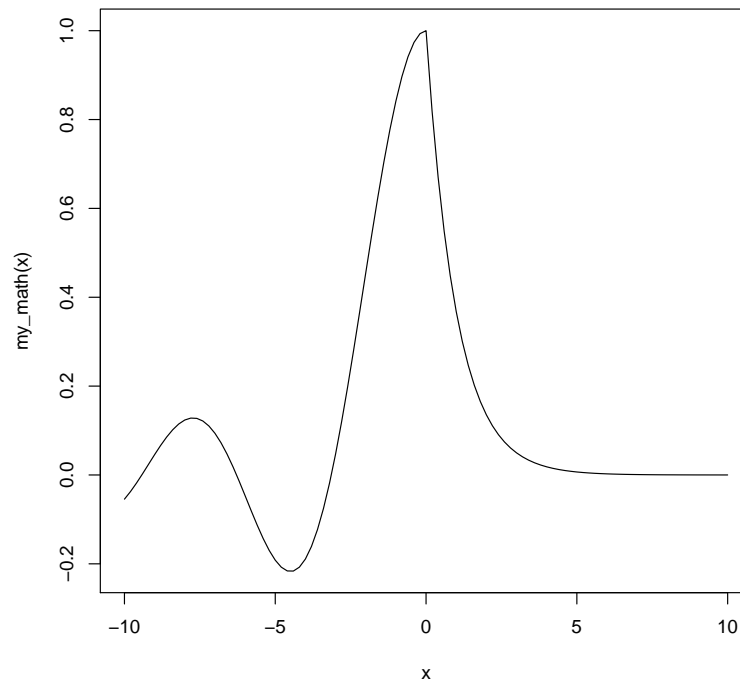
```
x<-seq(-2,2,1)
y<-my_math(x,a = 0.1,b = 2)
y

[1] -0.3784012 0.9092974 1.0000000 0.0367879 0.0135335
```

```
my_math(x = c(5,-4,0),a = 10,b = 12)

[1] 0.0673795 -0.1920637 1.0000000

curve(my_math,from=-10,to = 10)
```



- (c) Skapa funktionen `my_data_analysis(path,func,col)`. Funktionen tar och läser in en komma-separerad csv-fil, argumentet `path` ger sökvägen till filen. Sedan ska funktionen `func` användas på kolumnerna `col` i datamaterialet, och sedan ska resultatet returneras. Se testfallen nedan för hur funktionen ska fungera. Filen `AppleTest.csv` finns tillgänglig för att testa funktionen. **1.5p**

```
my_data_analysis(path = "AppleTest.csv",func=mean,col = 3:6)

  Open    High    Low   Close
352.365 355.432 348.971 352.372

my_data_analysis(path = "AppleTest.csv",col = 5:4,func = max)

  Low    High
426.51 431.37
```

```
temp<-my_data_analysis(path = "AppleTest.csv",col = c(6,3),func = sum)
temp

      Close    Open
122978 122975
```

5. Statistik och grafik:

- (a) Generera 250 slumpstal från en t-fördelning med 7 frihetsgrader och medelvärde 12. **1p**
- (b) Utgå från det inbyggda datamaterialet CO2 som finns i R. Använd ggplot2 för att: Skapa en scatter plot med `conc` på x-axeln och `uptake` på y-axeln. Punkterna i grafen ska ha olika färg beroende på värdet på variabeln `Treatment`. **1p**
- (c) Läs in filen `HUS_eng.txt` i R och spara som en `data.frame`. **2p**
 - i. Skapa en frekvenstabell mellan variablerna `no.bedroom` och `air.conditioning`.
 - ii. Räkna ut standardavvikelsen för variabeln `price`, men bara för de hus som som uppfyller: $3 \leq \text{sovrums} \leq 5$.
 - iii. Skapa ett histogram över variabeln `living.area` med relativ skala på y-axlen.

Lycka till!