LINKÖPINGS UNIVERSITET
Institutionen för datavetenskap
STIMA
Josef Wilzén

2019-08-14
Programmering i R, 7.5 hp
732G33

Tentamen i Programmering i R, 7.5 hp

Skrivtid: 8.00-12.00
Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2"
och några andra referenskort tillgängliga 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**. Kommentera din kod och använd en god kodstil.
Spara filen med namnet **tentaX.R** där **X** är ditt SC-nummer (klientnummer). Det numret kan du se i studentklienten. Exempel: om du har SC-nummer SC12345 så ska filen heta tentaSC12345.R
Filen lämnas sedan in via studentklienten. Notera att du ska lämna in **en** fil med alla dina lösningar.
När du har loggat ut från datorn är tentan avslutad. Frågor kan ställas till lärare via studentklienten.
Spara filer på skrivbordet: `~/Desktop/`
Kommentera direkt i din R-fil 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.

Programvaror

För att ladda in kursmodulen, kör följande kommando i en terminal

```
module load courses/732G33
```

så kommer du få tillgång till R, RStudio samt de R-paket som behövs för tentan. Öppna en terminal genom att trycka ctrl+alt+T

Kör sedan i terminalen

```
rstudio
```

för att öppna Rstudio.
Övriga program:

- `caja` används som filhanterare
- `pluma` används för att läsa textfiler, kan användas för att titta på datafiler innan inläsning.

Uppgifter

1. Datastrukturer och beräkningar(4p)

   (a) Skapa en data.frame med namnet `bikes`, med följande variabler. **1.5p**

      i. `type` (factor): racer, hybrid, MTB, hybrid, racer
      ii. `color` (factor): red, green, blue, red, blue
      iii. `year` (numeric): 2017, 1999, 2005, 2001, 2012
      iv. `ID` (character): A23, B11, C45, B23, A88

   (b) Beräkna

   $$y = \frac{sin(\pi \cdot x) \cdot exp\,(-4 \cdot x)}{x^2}$$

   där $x = 0.11$ och avrunda till fyra decimaler. Spara svaret i variabeln y. **1p**

   (c) Skapa en vektor av längd 10 som innehåller slumptal dragna från en t-fördelning[1], döp till `my_vect`. Skapa en matris med 3 rader och 5 kolumner, och fyll den radvis med heltalen 1 till 15, döp till `my_mat`. Lägg i en lista som du döper till `my_list`. Om du gjort rätt så ska listan se ut enligt nedan. Notera att slumptalen ändras beroende på seed. **1.5p**

```
print(my_list)

$my_vect
 [1]  0.153863  1.108810 -0.682758  0.702163  1.843797  2.341816  1.038251
 [8]  1.470971  0.239759  1.027838

$my_mat
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
```

[1] Du väljer själv antal frihetsgrader i fördelningen.

2. Kontrollstrukturer (4p)

(a) Skapa en for-loop som gör följande: loopar över heltalen mellan 1 till 30: Om talet är jämt delbart med 3 ska texten "dela med 3!" skrivas ut till konsolen. Om talet är jämt delbart 5 så ska texten "dela med 5!" skrivas ut. Om talet är jämt delbart med både 3 och 5 så ska texten "dela med 3 och 5!" skrivas ut. **2p**

```
[1] "dela med 3!"
[1] "dela med 5!"
[1] "dela med 3!"
[1] "dela med 3!"
[1] "dela med 5!"
[1] "dela med 3!"
[1] "dela med 3 och 5!"
[1] "dela med 3!"
[1] "dela med 5!"
[1] "dela med 3!"
[1] "dela med 3!"
[1] "dela med 5!"
[1] "dela med 3!"
[1] "dela med 3 och 5!"
```

(b) Använd en while-loop för att göra följande: utgår från talet $x_0 = 1$. I varje iteration i ska x_i beräknas enligt följande:

$$x_i = cos\left(x_{i-1}\right) \cdot 2 \cdot x_{i-1}^3$$

I varje iteration ska x_i skrivas ut till konsolen. Loppen ska avbryta om den absoluta skillnaden mellan x_i och x_{i-1} är mindre än 0.001, alltså om $|x_i - x_{i-1}| < 0.001$. Notera att denna beräkning måste finnas med i loopen. **2p**

```
[1] 1.0806
[1] 1.18813
[1] 1.25254
[1] 1.22978
[1] 1.24405
[1] 1.23594
[1] 1.24089
[1] 1.23798
[1] 1.23973
[1] 1.23869
[1] 1.23932
```

3. Strängar och datum (4p)

   (a) Läs in paketen `lubridate` och `stringr` i R. Läs sedan in datamaterialet "Ada_Lovelace.txt" och spara som vektorn `ada_text`. **0.5p**

   (b) Använd stringr för att: Ta reda på alla de årtal som finns i `ada_text` och spara dessa i en textvektor med namnet `ada_year`. Sortera vektorn i kronlogisk ordning. Ett årtal defineras som fyra siffor på rad. Skapa sen en ny textvektor, med namnet `year_text`, som innehåller alla de rader som innehåller något årtal i `ada_text`. **1.5p**

   (c) Använd funktioner i R för att göra följande: **2p**

      i. Räkna ut hur många hela veckor som som som det gått mellan denna tentamen (2019-08-14) och och den förra tentamen (2019-06-10). Spara som `exam_weeks.`

      ii. Ta reda på hur många hela månader är det mellan 1748-10-12 och 1890-03-05. Spara som `my_months`.

      iii. Ta reda på hur många gånger som julafton (24/12) inföll på en måndag i intervallet 1800-01-01 till 2018-12-31. Spara som `monday_count`.

4. Funktioner och grafik: **4p**

   (a) Du ska nu skapa en function som du kallar `my_mad(x)`. Funktionen ska beräkna "Median absolute deviation" (MAD) på vektorn `x`, som beräknas på följande sätt: Åtgå från ett dataset $X_1, X_2, \ldots X_N$. Låt $\tilde{X}$ vara medianen för alla datapunkter X_i. Beräkna den absoluta skillnaden mellan alla datapunkter och medianen $\bar{X}_i = \left|X_i - \tilde{X}\right|$. MAD ges sen som medianen för alla $\bar{X}_i$. Kort kan detta utryckas som:

$$y = median\left(\left|X_i - \tilde{X}\right|\right)$$

   Funktionen ska först testa att `x` är numerisk, om den inte är det ska funktionen avbrytas och valfritt felmeddelande genereras. Se exemplen nedan på hur funktionen ska fungera. Det är inte tillåtet att använda några inbygga funktioner för att beräkana MAD, tex funktionen `mad()` är inte tillåten. **2p**

```
test1<-my_mad(x = c(10,12,42,2,3,5000))
test1
```

```
[1] 8.5
```

```
my_mad(x = "abc")
```

```
Error in my_mad(x = "abc"): x is not numeric!
```

```
data("trees")
my_mad(x = trees[,1])
```

```
[1] 1.9
```

```
my_mad(x = trees[,2])
```

```
[1] 4
```

```
data("chickwts")
my_mad(x = chickwts$weight)
```

```
[1] 62
```

(b) Du ska nu skapa en function som du kallar `my_mean(x)`. Funktionen ska beräkna medelvärdet på vektorn `x` enligt:

$$y = \frac{1}{N}\sum_{i=1}^{N} X_i$$

där N är antalet element i `x`. Beräkningen ska ske med hjälp av en *while-loop*, och inbyggda funktioner som beräknar medelvärden eller summor (tex `mean()` och `sum()`) är **inte** tillåtna. Du ska göra beräkningen "manuellt". Funktionen ska returnera en lista med medelvärdet och antal observationer. Se exemplen nedan för hur funktionen ska fungera.

```
test1<-my_mean(x = 1:10)
str(test1)

List of 2
 $ mean: num 5.5
 $ N   : int 10

mean(x=1:10)

[1] 5.5

data("trees")
my_mean(x = trees[,1])

$mean
[1] 13.2484

$N
[1] 31

data("chickwts")
my_mean(x = chickwts$weight)

$mean
[1] 261.31

$N
[1] 71
```

5. Linjär algebra, statistik, grafik (4p)

(a) Läs in filen"tecator_cor.csv" i R. Filen innehåller en kvadratisk matris med 100 rader och 100 kolumner. Notera att filen inte innehåller några variabelnamn. Beräkna egenvärden för matrisen, spara dessa i variabeln `e_val` (tips: `eigen`). Gör sedan en barplot över de 6 största egenvärdena. Du ska alltså ha 6 staplar där höjden bestämms av värdet på egenvärdena. Andra detaljer på plotten får du bestämma själv. **1.5p**

(b) Läs in datamaterialet "lake2.csv" i R och spara som en data.frame med namnet `lake`. Datamaterialet innehåller mätningar på vattenkvaliten i ett antal sjöar, och värdet på

olika kemiska variabler har mäts. Notera att i vissa sjöar har har det gjorts flera mätningar. Varje rad i datamaterialet motsvarar en mätning. Pga svenska namn på sjöarna så måste du ange `encoding="latin1"` för att kunna läsa in data korrekt. **1.5p**

i. Gör ett valfri boxplot över variablen temperatur ("Temp..øC").

ii. Dela in data i två grupper beroende på uppfyller $pH \geq 7$ (grupp 1) eller $pH < 7$ (grupp 2). Beräkna korrelationen mellan variablerna "Alk..Acid.meq.l" och "pH" för hela datamaterialet. Beräkna sen korrelationen för grupp 1 och grupp 2 separat. Om du gjort rätt ska du erhålla tre olika korrelationsvärden. Spara dessa i en vektor med namnet `cor_vect`, med ordningen: alla, grupp 1, grupp 2.

(c) Återgå nu till din data.frame `lake`, och använd ggplot2 för att göra följande: Gör en scatterplot mellan variablerna "Alk..Acid.meq.l" (x-axeln) och "pH" (y-axeln). Alla punkter i grupp 1 ska en färg och alla punkter i grupp 2 ska ha en annan färg. **1p**

Kom ihåg: Skriv alla dina lösningar i en körbar **R-fil**. Spara filen med namnet **tentaX.R** där **X** är ditt SC-nummer (klientnummer) Det numret kan du se i studentklienten. Exempel: om du har SC-nummer SC12345 så ska filen heta tentaSC12345.R Lämna sedan in din fil via studentklienten. När du har loggat ut från datorn är tentan avslutad.

Lycka till!