LINKÖPINGS UNIVERSITET
Institutionen för datavetenskap
STIMA
Josef Wilzén

2020-08-13
Programmering i R, 7.5 hp
732G33 och 732G83

Tentamen i Programmering i R, 7.5 hp

Skrivtid: 8.00-12.00

Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt,
16 poäng ger Väl godkänt.

Skriv dina lösningar i **fullständig och läsbar kod**. Kommentera din kod och använd en god kodstil.
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.

Instruktioner

- Se filen "**Information om hemtenta i 732G33 och 732G83.pdf**" för regler och detaljer kring tentan, där finns bland annat information om:
  - Inlämning
  - Kontakt med lärare
  - Hjälpmedel
- När ni är klara med tentan: På Lisam kan ni se ert Anonym-id, vilket är ert unika tenta-id. Detta ska ni använda när ni lämnar in er fil, ni namnger filen på formen [Anonym-id].txt Till exempel om ni har Anonym-id: 12345, så ska er inlämnade fil ha namnet 12345.txt Ni ska alltså lämna in en .txt-fil och inte en .R-fil. Eftersom det är en anonym tenta så ska denna fil ska **inte** innehålla ert namn eller Liu-ID.
- Era lösningar ska innehålla lämpliga kommenater, där ni förklarar de övergripande dragen och de viktiga stegen i er kod. Ni behöver inte föklara alla detaljer. Koden ska även ha en god kodstil. T.ex. så ska ni ha lämplig indentering och bra variabelnamn. Lösningar med fungerade kod men med brister i kommenater eller kodstil får poängavdrag.
- **Spara era lösningar ofta, om R kraschar kan kod förloras.**
- Tentan består av 5 uppgifter som ger 4 poäng vardera.

Uppgifter

Uppgift 1: Kontrollstrukturer och datastrukturer

Del A 2p

Nu ska funktionen `while_list(x)` skapas. Funktionen ska kunna utgår från den numeriska vektorn x, och ska med hjälp av en while loop hitta alla tal i `x` som är jämt delbara med 2, 3 och 5. Dessa tal ska sparas i en lista, där första elementet innehåller de tal som är jämt delbara med 2, andra elementet innehåller de tal som är jämt delbara med 3 och sista elementet innehåller de tal som är jämt delbara med 5. Om ett tal är jämt delbart med flera av talen 2, 3 och 5 så ska talet finnas med i alla motsvarande listelement i listan. Tex så ska talet 10 vara med i både första och andra elementet i listan. Om inget tal är jämt delbara med 2, 3 eller 5, så ska en lista med tre tomma element returneras. Se testfallen nedan.

```
while_list(x = 13)

[[1]]
NULL

[[2]]
NULL

[[3]]
NULL

a<-while_list(x = 3)
a

[[1]]
NULL

[[2]]
[1] 3

[[3]]
NULL

while_list(x = c(2,4))

[[1]]
[1] 2 4

[[2]]
NULL

[[3]]
NULL

while_list(x = c(5,25,35))
```

```
[[1]]
NULL

[[2]]
NULL

[[3]]
[1]  5 25 35
```

```
while_list(x = 1:12)
```

```
[[1]]
[1]  2  4  6  8 10 12

[[2]]
[1]  3  6  9 12

[[3]]
[1]  5 10
```

```
while_list(x = 25:35)
```

```
[[1]]
[1] 26 28 30 32 34

[[2]]
[1] 27 30 33

[[3]]
[1] 25 30 35
```

Del B 2p

Ni ska nu skapa funktionen `mat_func(n,m)`. Argmenten `n` och `m` är heltal större än 0. Funktionen ska skapa en matris med `n` rader och `m` kolumner, där varje element i matrisen ska ges av

$$y_{i,j} = \frac{(-1)^{i+1}}{j+i} \qquad i = 1, 2, \ldots, n \qquad j = 1, 2, \ldots, m$$

där i är radindex och j är kolumnindex. Varje element ska avrundas till 3 decimaler. Matrisen ska sedan returneras. Denna uppgift ska lösas med en nästlad loop.

Uppgift 2: Dubbelsidiga konfidensinterval 4p

Nu ska en funktion skapas som kan beräkna dubbelsidiga konfidensinterval (KI) för medelvärden och för andelar. Funktionen ska heta `my_conf(x,alpha)`. Argumentet `x` ska vara en numerisk vektor eller en vektor med bara ettor och nollor. `alpha` anger konfidensgrad, där `alpha=0.05` ger 95 % KI. Inga inbyggda funktioner för att beräkna KI i R får används utan de ska "räknas för hand", tex så är inte funktionerna `t.test()` eller `prop.test()` tillåtna. KI för medelvärden ska beräknas enligt:

$$\bar{x} \pm z_* \cdot \frac{s}{\sqrt{n}}$$

Där $\bar{x}$ är stickprovets medelvärdet, s är stickprovets standardavvikelse och n är antalet observationer. z_* är lämplig kvantil från standardiserade normalfördelningen ($N(x, \mu = 0, \sigma = 1)$). T.ex. så är $z_* = 1.959964$ korrekt för dubbelsidigt KI med `alpha=0.05`. KI för andelar ska beräknas enligt:

$$p \pm z_* \cdot \sqrt{\frac{p(1-p)}{n}}$$

Där p stickprovsandelen, z_* och n är samma som ovan. Om `x` bara består av ettor och nollor så ska KI för andelar beräknas, annars ska KI för medelvärde beräknas. Om `x` inte är numerisk eller heltal så ska funktionen avbrytas med `stop()` och skriva ut följande meddelande ``` ``x is not numeric'' ```. KI ska returneras som en numerisk vektor. Se testfallen för exempel på hur funktionen ska fungera.

```
set.seed(322)
a1<-rnorm(100,3,4)
b1<-rbinom(n = 75,size = 1,prob = 0.8)

a2<-c(19,12,32,42,53,22,43,11,12,45)
b2<-c(0,1,1,1,1,0,0,0,1,1,0,1,0,0,1,0,0,0,0)

ci1<-my_conf(x = a1,alpha = 0.05)
ci1

[1] 1.74726 3.37930

my_conf(x = a2,alpha = 0.01)

[1] 16.2011 41.9989

my_conf(x = a1,alpha = 0.001)

[1] 1.19328 3.93327

my_conf(x = b1,alpha = 0.1)

[1] 0.664062 0.829272

my_conf(x = b2,alpha = 0.05)

[1] 0.199050 0.643056
```

Uppgift 3: Datum och strängar 4p

Nu ska en funktion skapas som ska räkna ut antal hela veckor eller hela månader mellan två datum. Funktionen ska heta `date_count(date1,date2,unit)`, där argumenten `date1` och `date2` är två textsträngar med datum. Argumentet `unit` anger om det ska vara veckor ("week") eller månader ("month"). Vid fallet med veckor ska även antalet sekunder beräknas som motsvarar de hela veckorna (alltså sju dygn per vecka).

Funktionen ska testa följande saker, om inte ska fel genereras med meddelande som ges av exemplen:

- Testa att `date1` och `date2` är textsträngar
- Testa om `unit` har värdet "week" eller "month"

Funktion ska retunera en texsträng på formen som ges av exemplen nedan.

```
date_count(date1 = TRUE,date2 = "2020-08-31",unit = "week")

Error in date_count(date1 = TRUE, date2 = "2020-08-31", unit = "week"): date1 is not
character

date_count(date1 = "2020-08-3",date2 = 1:10,unit = "week")

Error in date_count(date1 = "2020-08-3", date2 = 1:10, unit = "week"): date2 is not
character

date_count(date1 = "2020-08-3",date2 = "2020-08-31",unit = "day")

Error in date_count(date1 = "2020-08-3", date2 = "2020-08-31", unit = "day"): wrong
format of unit

date_count(date1 = "2020-08-3",date2 = "2020-08-31",unit = 233.1)

Error in date_count(date1 = "2020-08-3", date2 = "2020-08-31", unit = 233.1): wrong
format of unit

D<-date_count(date1 = "2020-08-3",date2 = "2020-08-7",unit = "week")
class(D)

[1] "character"

D

[1] "date1: 2020-08-03 date2: 2020-08-07 weeks: 0 seconds: 0"

date_count(date1 = "2020-08-3",date2 = "2020-08-10",unit = "week")

[1] "date1: 2020-08-03 date2: 2020-08-10 weeks: 1 seconds: 604800"

date_count(date1 = "2020-08-3",date2 = "2020-08-17",unit = "week")

[1] "date1: 2020-08-03 date2: 2020-08-17 weeks: 2 seconds: 1209600"

date_count(date1 = "2020-04-14",date2 = "2020-09-29",unit = "week")
```

```
[1] "date1: 2020-04-14 date2: 2020-09-29 weeks: 24 seconds: 14515200"
```

```
date_count(date1 = "2020-08-01",date2 = "2020-08-31",unit = "month")
```

```
[1] "date1: 2020-08-01 date2: 2020-08-31 months: 0"
```

```
date_count(date1 = "2020-08-01",date2 = "2020-09-01",unit = "month")
```

```
[1] "date1: 2020-08-01 date2: 2020-09-01 months: 1"
```

```
date_count(date1 = "1930-02-18",date2 = "1999-09-10",unit = "month")
```

```
[1] "date1: 1930-02-18 date2: 1999-09-10 months: 834"
```

Uppgift 4: Binomialfördelningen 4p

En binomialfördelad slumpvariabel anger sannolikheten att erhålla ett antal positiva utfall givet ett antal försök, där varje försök har samma sannolikhet att vara positivt.

Ni ska nu skapa funktionen som ska kunna beräkna sannolikhetsfunktionen och den kumulativa sannolikhetsfunktionen för binomialfördelad slumpvariabel. Funktionen ska heta `my_binom(x,n,p,cdf)`. `x` och `n` är heltal som uppfyller

$$n \geq 0 \qquad x \geq 0 \tag{1}$$

och `p` anger sannolikheten för ett positivt utfall, och är ett reellt tal mellan 0 och 1. Funktionen ska testa att vllkoren i (1) är uppfyllda, om det inte är det ska valfria felmeddelanden genereras. Sannolikhetsfunktionen ges av

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

där

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

där ! betyder fakultet. Den kumulativa sannolikhetsfunktionen ges av

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

Exempel: $p(x=1,n=10,p=0.5)$ anger sannolikheten att få ett positivt utfall när 10 oberoende försök görs som alla har sannolikhet 0.5 att vara positivt. Om `cdf=TRUE` så ska $P(X \leq x|n,p)$ beräknas, annars om `cdf=FALSE` så ska $p(x,n,p)$ beräknas. Notera att det inte är tillåtet att använda några funktioner som direkt beräknar $p(x,n,p)$ och $P(X \leq x|n,p)$, tex `dbinom()` och `pbinom()` inte tillåtna, men ni kan använda dessa för att testa om ni räknat rätt. Se exemplen nedan på hur funktionen ska fungera.

```
my_binom(x = -3,n = 10,p = 0.6,cdf = TRUE)
```

```
Error in my_binom(x = -3, n = 10, p = 0.6, cdf = TRUE): x is negative
```

```
my_binom(x = 3,n = -2,p = 0.6,cdf = TRUE)
```

```
Error in my_binom(x = 3, n = -2, p = 0.6, cdf = TRUE): n is negative
```

```
B<-my_binom(x = 3,n = 10,p = 0.6,cdf = FALSE)
class(B)
```

```
[1] "numeric"
```

```
B
```

```
[1] 0.0424673
```

```
dbinom(x = 3,size = 10,prob = 0.6) # jämför med detta värde
```

```
[1] 0.0424673
```

```r
my_binom(x = 9,n = 29,p = 0.2,cdf = FALSE)
```

```
[1] 0.0591182
```

```r
my_binom(x = 0,n = 15,p = 0.1,cdf = FALSE)
```

```
[1] 0.205891
```

```r
my_binom(x = 3,n = 10,p = 0.6,cdf = TRUE)
```

```
[1] 0.0547619
```

```r
pbinom(q = 3,size = 10,prob = 0.6) # jämför med detta värde
```

```
[1] 0.0547619
```

```r
my_binom(x = 3,n = 12,p = 0.2,cdf = TRUE)
```

```
[1] 0.794569
```

```r
my_binom(x = 11,n = 17,p = 0.7,cdf = TRUE)
```

```
[1] 0.403181
```

```r
my_binom(x = 17,n = 17,p = 0.7,cdf = TRUE)
```

```
[1] 1
```

Uppgift 5: Grafik och datahantering 4p

Del A

Utgå från det inbyggda datamaterialet trees. Använd ggplot2 för att skapa dessa tre plottar och spara dem i variabler:

- ett histogram för Height
- en boxplot för Girth
- ett punktdiagram med Height på x-axeln och Volume på y-axeln.

Lägg sedan dessa tre plottar tillsammans i samma plot. Den plotten ska ha tre rader med en plot på varje rad. Det ska alltså bli en plot med tre subplottar som rader. Spara plotten i en variabel som ni kallar `my_plots`.

Del B

Läs in datamaterialet "HUS_eng.txt" i R. Varje rad representerar ett hus som har sålts, och kolumnerna visar olika egenskaper hos husen.

Välj ut de hus som uppfyller kriterierna: Det ska inte ha luftkonditionering (=0), ha exakt tre sovrum och vara byggt mellan 1920 till 1950 (inkludera gränserna). Beräkna sedan medianen för dessa hus för variablen "living.area". Spara medianen i variabeln `my_median_val`.

Kom ihåg: När ni är klara med tentan: På Lisam kan ni se ert Anonym-id, vilket är ert unika tenta-id. Detta ska ni använda när ni lämnar in er fil, ni namnger filen på formen [Anonym-id].txt Till exempel om ni har Anonym-id: 12345, så ska er inlämnade fil ha namnet 12345.txt Ni ska alltså lämna in en .txt-fil och inte en .R-fil. Eftersom det är en anonym tenta så ska denna fil ska **inte** innehålla ert namn eller Liu-ID.

Lycka till!