LINKÖPINGS UNIVERSITET
Institutionen för datavetenskap
Avdelningen för statistik
Josef Wilzén & Måns Magnusson

2013-03-22
Programmering i R, 7.5 hp
732G33

Tentamen i Programmering i R, 7.5 hp

Skrivtid: 14-18
Hjälpmedel: Valfritt tryckt material (böcker, anteckningar, utskrifter mm)

Jourhavande lärare: Josef Wilzén & Måns Magnusson
Besöker tentan kl 15.15 och kl 17.00

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.
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 (4p)

   (a) Skapa följande vektorer:

      i. `myBoolean`: innehåller vektorn `c(TRUE,NA,FALSE,TRUE)` upprepad 10 gånger. Spara i `answer.1a1`

      ii. `X`: innehåller sekvensen som börjar på 1 och slutar på 40, och har längd 40. Spara i `answer.1a2`

      iii. `Y`: skapas genom formeln $Y = X * exp(X/10) * sin(X^2)$ där X är vektorn `X` ovan. Spara i `answer.1a3` 1.5p.

   (b) Skapa en data.frame där vektorerna i (a) är kolumner och har ordningen som ges ovan. Döp den till `answer.`1b. Se till att kolumnerna i `answer.`1b har namnen "`myBoolean`","`X`" och "`Y`". 1p.

   (c) Kopiera `answer.1b` till `answer.1c`, gör sedan följande med `answer.1c`: Skriv en kod som som tar bort alla rader där `myBoolean` är `NA`. Öka sedan `X` med 10 % på de rader där `myBoolean` är `TRUE`. Uppdatera sedan `Y` med de nya X-värdena enligt formeln i (a). 1.5p.

2. Kontrollstrukturer (4p)

   (a) Skriv en for-loop som loppar över talen $1, 2, ..., 9, 10$ och i varje iteration beräknar i^3 där `i` är loop-index, och skriver ut texten nedan: 1p.

```
## [1] "Current value:  1"
## [1] "Current value:  8"
## [1] "Current value:  27"
## [1] "Current value:  64"
## [1] "Current value:  125"
## [1] "Current value:  216"
## [1] "Current value:  343"
## [1] "Current value:  512"
## [1] "Current value:  729"
## [1] "Current value:  1000"
```

   (b) Skriv en if-else-sats som gör följande: Givet ett tal `myValue`, testa om `myValue` är jämt delbart med 4, om så är fallet skriv ut **`‘‘Jämt delbart med 4’’`** till skärmen, annars testa om `myValue` är jämt delbart med 3, om så är fallet skriv ut **`‘‘Jämt delbart med 3’’`** till skärmen, annars skriv ut **`‘‘Kan ej delas med 3 eller 4’’`**. Spara din kod i en funktionskropp där `myValue` är parameter enligt nedan. 2p.

```
answer.2b <- function(myValue) {
    # kod f<U+00F6>r if-else h<U+00E4>r...
}
```

   (c) Skriv en loop som hittar det **minsta** heltal Z som uppfyller olikheten $5639/Z \leq 100$. Spara Z i `answer.2c`. 1p

3. Strängar och datum (3.5p)

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

   (b) Skapa en vektor av datum för hela 2013, d.v.s. en vektor med 365 element av datum från “2013-01-01” till “2013-12-31”. Spara som `answer.3b` 1p

   (c) Räkna ut hur många veckor det är mellan “2013-03-22” och “2013-06-01”. Spara som `answer.3c` 1p.

   (d) Läs in textfilen `transtrom.txt` i R och räkna antalet tecken i varje rad genom att använda funktioner i R. Ge ditt svar som en vektor och spara i `answer.3.d` 1p

4. Funktioner: Matematik (4p)

(a) Rötterna till andragradsekvationen $ax^2 + bx + c = 0$ ges av formlerna:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ och } x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Skriv en funktion som beräknar rötterna till en andragradsekvation enligt formeln ovan, med parameterar `a`, `b` och `c`. Funktionen ska returnera en vektor med lösningarna enligt: `c(x1,x2)`. Spara funktionen som `answer.4aFunc`. Testa din funktion med parametrarna nedan och spara i `answer.4aTest`. 2p.

```
answer.4aTest <- answer.4aFunc(a = 4, b = -9, c = -5)
```

(b) Nu ska funktionen i (a) göras mer generell. Nu ska `a`,`b` och `c` kunna vara vektorer av samma längd. Detta innebär att funktionen ska lösa en eller flera andragradsekvationer i samma körning. Funktionen ska nu returnera en data.frame med två kolumner som ska heta "`X1`" och "`X2`". Spara som `answer.4dFunc`. Testkör funktionen med parametrarna enligt nedan, och spara i `answer.4dTest`. 2p.

```
answer.4dTest <- answer.4dFunc(a = c(4, 1, -5), b = c(-9, -4, -10), c = c(-5,
    4, 8))
```

5. Funktioner: Statistik och grafik (4.5p)

(a) Skriv en funktion som gör följande:

- Parameterar: `mySample` - numerisk vektor, `conf` - kondidensnivån, tal mellan 0 och 1.
- Funktionen ska beräkna ett konfidensinterval för medelvärdet för vektorn `mySample` med konfidensnivån `conf`, (om `conf=0.95` så ska ett 95 % konfidensinterval beräknas)
- Funktionen ska returnera en vektor med elementen: Undre konfidensgräns, medelvärde, övre konfidensgräns. Spara funktionen som `answer.5a`. 1p

(b) Skriv en funktion som gör följande:

- Parametrar: `myData` - en numerisk vektor, `borders` - en numerisk vektor med två element.
- Funktionen ska göra ett histogram över `myData`. Två lodräta linjer ska läggas in i histogrammet på de positioner som vektorn `borders` anger. Bredden på de lodräta linjerna ska vara 5. Tips: `?abline`
- Spara funktionen som `answer.5b`. 1p

(c) Använd funktionerna från (a) och (b) för att skapa följande funktion:

- Parameterar: `pop` - en numerisk vektor, `sampleSize` - en heltalsvektor , `conf` - signifikansnivån

- Funktionen ska först dra ett stickprov (utan återläggning) av storlek `sampleSize` från `pop`.
- Därefter ska ett konfidensintervall för stickprovet beräknas med konfidensgrad `conf`.
- Vektorn `pop` ska plottas i ett histogram. Där ska konfidensintervallets gränser läggas in som lodräta linjer (som har bredd 5) i plotten.
- Funktionen ska returnera en vektor med: Undre konfidensgräns, medelvärde, övre konfidensgräns. Spara funktionen som `answer.`5c.
- Testa funktionen med parameterarna nedan och spara i `answer.5Test` 2.5p

```
myPopulation <- round(rchisq(n = 5000, df = 6), 1)
answer.5Test <- answer.5c(pop = myPopulation, sampleSize = 20, conf = 0.01)
```