Inlämningsuppgift block 2 – funktioner, objekt, grafer, import av data

Statistiska Metoder med R

Inlämningsuppgift block 2 – funktioner, objekt, grafer, import av data

Statistiska Metoder med R

Försättsblad

Grupp 10:

Ange namn och personnummer(utan fyra sista siffrorna) på dem som deltagit aktivt:

Jacob Widaeus 950729

Namn

Namn

Namn

Namn

Namn

1. Session

1.1. Kolla vilka paket och objekt som är laddade just nu med funktionen search() och klipp in här:

```
search()
```

```
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
```

1.2. Slå upp hjälptexten om funktionen search() så här ?search()

```
?search()
```

```
starting httpd help server ... done
```

läs om paketet base så här ?base

```
?base
```

2. Matematiska operationer

Använd R som räknare. Klistra din kod och vad R svarar här

2.1. Arean på en kvadrat med sidan 5 cm.

```
area_kvadrat = 5^2
print(area_kvadrat)
```

```
[1] 25
```

2.2. Volymen på en kub med sidan 5 cm.

```
volym_kub = 5^3
print(volym_kub)
```

```
[1] 125
```

2.3. Arean på en cirkel med radien 5 cm. Använd en inbyggd variabel som heter pi.

```
area_cirkel = pi * 5^2
print(area_cirkel)
```

```
[1] 78.53982
```

(formeln framgår i uppgift 3.1)

3. Function

Om du utför samma beräkning ofta kanske du vill skapa en egen funktion i R

3.1. klistra in följande kod i R

CircleArea <- function(radius) {

radius * *radius* * pi

}

```
CircleArea <- function(radius) {
  radius*radius*pi
}
```

3.2. Vad gör koden ovan?

Den automatiserar beräkning av arean av en cirkel till funktionen CircleArea() som tar emot ett argument, radius som är radien i cm.

3.3. Använd din nya funktion CircleArea() för att beräkna arean på en cirkel med radien 15 cm

```
CircleArea(15)
```

```
[1] 706.8583
```

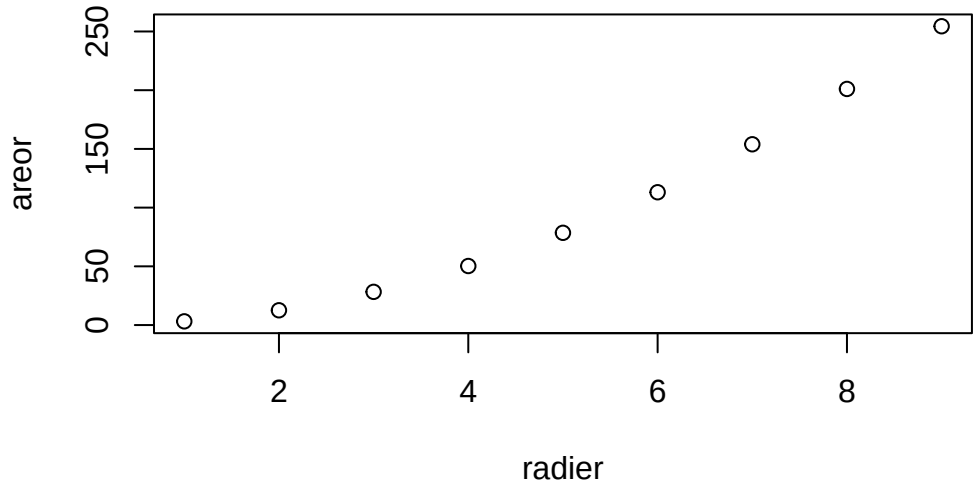
3.4. Skapa en vector med radierna 1,2,3,4,5,6,7,8,9 med koden radier <- c(1:9)

Skapa en vector med namn areor beräknade utifrån radierna i vectorn radier. Använd din funktion CircleArea()

```
radier <- c(1:9)
areor <- c(CircleArea(radier))
```

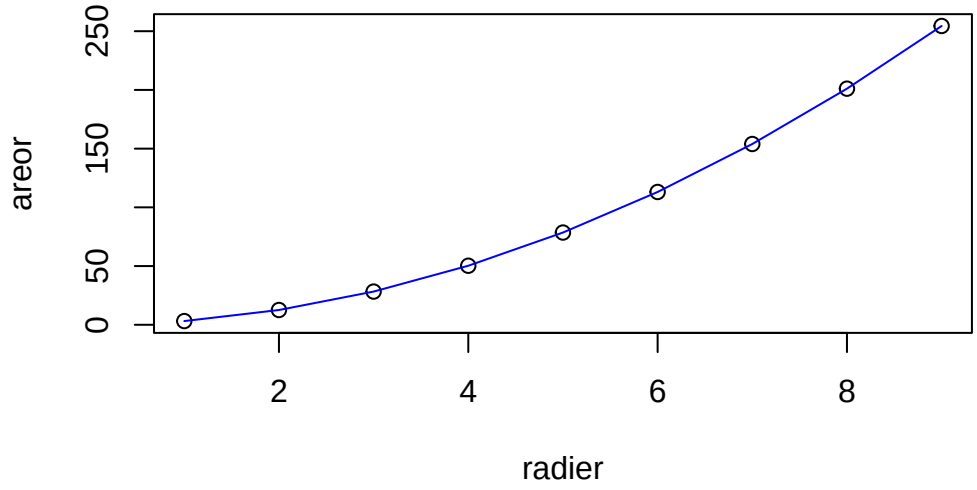
3.5. Klistra in resultatet av funktionen plot(radier, areor)

```
plot(radier, areor)
```



3.6. Bind ihop punkterna i grafen med en linje Tips: sid 8 ISwR

```
plot(radier, areor)
lines(radier, areor, col = "blue")
```



Tips: använd pil uppåt för att hämta tillbaka kommandon bakåt i historien, du kan redigera dessa och på så sätt slippa skriva om tex långa variabelnamn.

3.7. Klistra in den resulterande grafen här. Kopiera grafen med copy paste, eller i Windows RGui klicka på kamerasymbolen uppe till höger, eller välj File/copy to the clipboard.

Olika typer av objekt.

Jag använder i den här texten den engelska stavningen för objekt-typerna:

vector, matrix, factor, list, data.frame

4. vector

Skapa ett antal vector med några olika funktioner

4.1. fyll V1 med talen 1, 2, 3, 3, 6, 17, 23, 12, 32, använd funktionen c()

4.2. fyll V2 med alla jämna tal mellan 2 och 100. Använd funktionen seq()

4.3. fyll V3 med alla heltal mellan 1 och 250. Använd funktionen :

Tips: Vad händer som du skriver 1:250 i prompten?1:250

4.4. fyll kundlista_radnamn med orden namn, adress, email, telefonnummer. Varför är det viktigt att använda ” ” eller ’ ’ här? Använd funktionen c()

4.5. fyll kunder200 med orden i kundlista_radnamn 200 gånger efter varandra med funktionen rep()

Kopiera endast in några rader här som svar när du printar ut kunder200

4. 6. plocka ut andra ordet i kunder200 med hjälp av funktionen []

4.7. letters är en inbyggt vector. Printa ut innehållet i letters.

(Printa ut betyder i detta fall att R skriver i prompten)

4.8. kontrollera hur många bokstäver som ingår i letters med funktionen length()

4.9. printa bokstäverna c, d, e. använd letters och funktionerna [] och :

Tips: Vad händer om du skriver letters[20:23]

4.10. Fyll abc med bokstäverna a, b, c , a, b, c, a, b, c. använd letters och funktionerna rep(), []

4.11. Undersök funktionen sample(letters, 5, replace=TRUE) Skriv kort här vad sample() gör.

du kan även läsa ?sample()

4.12. testa att köra sample(letters, 5, replace=TRUE) igen. Fick du samma resultat?

Om man behöver kontrollera vilka slumptal R genererar, till exempel som när man samarbetar om en inlämningsuppgift och behöver kunna reproducera resultaten, så kan man ställa in ett så kallat seed med funktionen set.seed()

4.13. Använd ditt gruppnummer som seed och kör koden nedan. I exemplet använder jag gruppnumret 17, som du får byta ut mot ditt eget. Grupp A = 1, grupp B = 2, etc.

gruppnummer <- 17

set.seed(gruppnummer)

sample(letters, 5, replace=TRUE)

sample(letters, 5, replace=TRUE)

set.seed(gruppnummer)

sample(letters, 5, replace=TRUE)

sample(letters, 5, replace=TRUE)

Klipp in och kommentera resultatet här. Lägg märke till att set.seed() anger en startpunkt för en serie av slumpningar.

Jag kommer att påminna er att ställa in seed i vissa inlämningsuppgifterna eftersom den rättande läraren ska kunna reproducera era beräkningar snabbt och fullständigt.

5. Att söka igenom och justera data.

5.1. Simulera 400 tärningskast i en vector, kalla den dice

Använd funktionerna sample() och : (Tips: vad sägs om 1:6)

Skapa ett misstag genom att sätta det omöjliga utfallet 66 i position 12 i dice

Tips: sid 21 ISwR, Indexing

5.2 Så här kan du använda en graf för att söka igenom data efter avvikande observationer. Skapa en plot av dice med kommandot plot() Notera att en observation avviker från de övriga, och dessutom har ett värde utanför de möjliga utfallen. Klistra in grafen här.

5.3. Vad händer om du skriver

which(dice > 6)

5.4.Byt ut den avvikande observationen mot NA

Tips: du kan hantera tecknen NA precis som en siffra

6. matrix

6.1. använd funktionen is.vector() för att kontrollera om V1 som du skapat är en vector

Tips. Det borde se ut så här:

is.vector(V1)

[1] TRUE

6.2. Skapa en matrix, M1, genom att kopiera V1 och tilldela M1 dimensionerna 3x3 med funktionen dim() Tips: sid 17 ISwR

printa ut M1

6.3. testa med funktionen is.vector() om M1 är en vector och

testa med funktionen is.matrix() om M1 är en matrix

6.4.vad sker om du ger kommandot t(M1)

7. factor

På sid 18 (ISwR) står beskrivet hur man kan skapa en faktor, en typ av variabel som kan hantera kategoriska variabler. Här ska du få ett alternativt sätt att skapa faktorer.

börja med set.seed(gruppnummer)

7.1 fyll inkomsttagare med 50 tecken, en slumpmässig följd av bokstäverna k, l, m.

k, l och m är någon godtycklig kategorisk variabel.

använd funktionerna sample() och : och []

7.2 vad händer om du glömmer skriva replace=TRUE

printa inkomstagare

7.3 använd is.vector() för att testa om inkomsttagare är en vector

7.4 fyll inkomsttagare_faktor med hjälp av funktionen factor(inkomsttagare)

7.5 använd is.vector() för att testa om inkomsttagare_faktor är en vector och

använd is.factor()för att testa om inkomsttagare_faktor är en factor

7.6 printa inkomsttagare_faktor

lägg märke till ”levels”

8. list

Objekttypen list kan innehålla en blandning av olika objekttyper

8.1 skapa en lista av några vector och matrix du har skapat

enlista <- list(V1, M1, inkomsttagare)

printa enlista

9. data.frame

data.frame är ett lämpligt format att lagra data i R

du kan tänka dig en data.frame som en tabell med numrerade rader där varje rad motsvarar ett case (till exempel en försöksperson) och varje kolumn motsvarar en uppmätt variabel (blodtryck, ålder, vikt för att ge tre exempel). En data.frame är som en list av flera vector. En vector för varje variabel. Om du skapar en data.frame av en vector som består av ord så kommer den stöpas om till factor. R kommer tolka den som en kategorisk variabel.

9.1 skapa en data.frame, D1, av din matrix M1 (tips: sid 20 ISwR)D1

printa ut D1 och notera att kolumn och radnamn tillkommit och ersatt noteringen [,1] [,2] osv

9.3 printa ut första kolumnen av D1 med hjälp av $

9.4 printa ut första raden av D1 med hjälp av []

Editera en data.frame med fix()

9.5 Öppna din data.frame D1 med fix() och lägg till en kolumn med värdena: låg, låg, hög

klicka på kolumn-namnet för dina nya värden och notera vilken variabeltyp R har valt.

Pröva att göra samma sak med matrix M1, dva öppna med fix() och lägga till låg, låg, hög. Det blir lite annorlunda resultat. Kommentera.

9.6 Använd funktionen copy i word samt funktionen read.table("clipboard", header=T) för att skapa en data.frame D2 av tabellen nedan. Tips: sid 53 ISwR

Har du Mac? Då ser kommandot lite annorlunda ut. Pröva:

D2 <- read.table(pipe("pbpaste"),header=T)

D2 <- read.delim(pipe("pbpaste"), header=T)

Tänk på att jag inte har senaste allt i mac – du kan behöva leta lite själv här för att hitta rätt metod!

9.7 Använd funktionen plot() och $ för att rita ett spridningsdiagram med data från D2 i uppgift 9.6 Vikt på y-axeln och längd på x-axeln. Tips om $: sid 21 ISwR

Lägg märke till att tabellen du importerat till D2 har samma format som den färdiga data.frame

En rad per case

En kolumn per variabel

Det är inte ovanligt att man stöter på data i något annat format

I följande tabell har en en kategorisk variabel (treatment) delats upp i en kolumn per nivå

9.8 Formattera om tabellen på lämpligt sätt och importera den med read.table() till d3

Blodtryck för 30 försökspersoner har slumpmässigt delats in i tre grupper: kontroll, medicinering, träning

Ytterligare ett sätt att skapa en data.frame är att lägga ihop vektorer med funktionen data.frame()

9.9 Skapa slumpmässiga, avrundade blodtrycksvärden till 10 personer per grupp genom att köra koden nedan. Först ska du använda set.seed()

set.seed(gruppnummer)

bt_kontroll <- round(rnorm(10, 90, 10))

bt_medicin <- round(rnorm(10, 80, 10))

bt_springa <- round(rnorm(10, 80, 10))

printa ut och klipp in blodtrycksvärdena här

9.10 Kommentera: Vad gör funktionen round()?

Blodtryck rapporteras alltid som heltal. Undersök round() genom att skapa en en serie slumptal som i bt_kontroll, men med 2 decimaler.

9.11 lägg samman alla observationer i en vector, bt

bt <- c(bt_kontroll, bt_medicin, bt_springa)

printa och klipp in bt här

9.12 nu skapar vi en vector med treatment som passar med vector bt

treatment <- rep(c("kontroll", "medicin", "springa"), c(10,10,10))

Sätt ihop treatment och bt till en data.frame

btdataframe <- data.frame(treatment, bt)

printa btdataframe och klipp in här

9.14 Slutligen ska pröva strategin att importera data från en txt fil till en data.frame

Ladda ner filen learningstrategies.txt – ligger under "Filer->course_inlämningsuppgift" i Canvas

Varje rad representerar en försöksperson

Kolumnen strategies anger vilken metod personen instruerats att lära in ord

Kolumnen words visar hur många ord personen lyckades lära in

Ta reda på vilket ditt working directory är just nu ged getwd()

Lägg filen learningstrategies där. Skapa sedan en data.frame med namnet learningstrategies

Tips: sid 47 ISwR

Använd funktionen read.table()

Filen har en header

Om filen är sparad i ditt working directory behöver du inte ange filens hela path

Det ska räcka med ”learningstrategies.txt”

9.15 Printa en översikt av innehållet i learningstrategies och klistra in här

summary(learningstrategies)

9.16 Printa och klistra in ett histogram med hist()

Titta på uttycket i hist() nedan.

Förklara vad följande funktioner gör

$words

[]

strategy == “intentional”

hist(learningstrategies$words[learningstrategies$strategy == “intentional”])

9.17 Boxplot

Använd attach(learningstrategies) så att namnen i learningstrategies blir tillgängliga för R.

Rita en boxplot över variabeln words som beror på strategy.

boxplot(words ~strategy)

Klipp in blox-plotten här. Om du har tid och lust kvar, pröva om du kan ändra färg eller annat utseende på grafen. Inspiration härifrån kanske: