Inlämningsuppgift block 3 – tabeller, grafer, fördelningar, centrala gränssnittssatsen

Statistiska Metoder med R

Försättsblad

Gruppnamn 10

Namn och personnumer på dem som deltagit aktivt:

Jacob Widaeus 950729

Hej

Denna gruppuppgift berör materialet i **ISwR kapitel 3 och 4**. Ni kommer att arbeta med funktioner som sammanfattar data i tabeller. En sport inom dataanalys är att försöka sammanfatta en stor datamängd med bara två siffror: ett centralmått och ett variationsmått. Ni får sammanfatta datamängder med olika typer av grafer. Ni kommer att undersöka fördelningar och funktioner för fördelningar i R. Varför är så många variabler normalfördelade? Här få ni testa ett argument som utgår ifrån approximation av binomialfördelningen. Det ingår även en simulering som illustrerar centrala gränssnittssatsen. **Om du inte läst så mycket statistik tidigare bör du ha Lind eller en annan bra statistikbok till hands för att få ut det mesta av gruppuppgift 3!**

Om du kör fast men någon uppgift ta hjälp av dina kurskamrater i första hand. Skulle det vara så att ni tillsammans ändå inte lyckas lösa uppgiften ta kontakt med lärare.

Skriva tabeller

1.1 Ladda ISwR paketet och printa ut de fem första raderna i data.frame stroke med

```
library(ISwR)
```

```
Warning: package 'ISwR' was built under R version 4.4.1
```

```
stroke[1:5,]
```

```
     sex       died       dstr age dgn coma diab minf han  dead   obsmonths
1   Male 1991-01-07 1991-01-02  76 INF   No   No  Yes  No  TRUE  0.16339869
2   Male       <NA> 1991-01-03  58 INF   No   No   No  No FALSE 59.60784314
3   Male 1991-06-02 1991-01-08  74 INF   No   No  Yes Yes  TRUE  4.73856209
4 Female 1991-01-13 1991-01-11  77 ICH   No  Yes   No Yes  TRUE  0.06535948
5 Female       <NA> 1991-01-13  76 INF   No  Yes   No Yes FALSE 59.28104575
```

Varför står det ett ensamt komma efter 1:5?

För att markera att det gäller samtliga kolumner.

1.2 Använd funktionen names() för att ta ut en lista på variabler i stroke

```
names(stroke)
```

```
 [1] "sex"       "died"      "dstr"      "age"       "dgn"       "coma"
 [7] "diab"      "minf"      "han"       "dead"      "obsmonths"
```

1.3 Är age tillgänglig? Testa:

```
age
```

```
Error in eval(expr, envir, enclos): object 'age' not found
```

1.4 Använd attach för att namnen i stroke ska blir tillgängliga för R

```
attach(stroke)
```

Testa igen om age är tillgänglig:

```
print(age[1:20])
```

```
 [1] 76 58 74 77 76 48 81 53 73 69 86 79 69 58 71 84 63 85 81 77
```

1.5 Vad är tanken med attach egentligen? Vad finns det för fördel med att R inte automatiskt ser namnen inne i ett objekt? Och om det nu är så bra, finns det något sätt att få R att glömma namnen i ett objekt? (ISwR sid.36)

Tanken med attach är att göra variabler "direkta" i en workspace utan att behöva referera till data frame. I större projekt där man använder många data set och frames och kanske

listor underlättar namngivning etc utan attach. detach() kan användas för att återställa variabeltillgängligheten.

1.6 Använd tapply() för att beräkna medelvärde för variabeln age, uppdelat på patienter som varit i koma eller inte efter sin stroke.

```
tapply(stroke$age, stroke$coma, mean, na.rm = TRUE)
```

```
      No      Yes
69.59463 72.18750
```

1.7 Använd tapply() för att beräkna medianålder för patienten uppdelat på olika diagnos, dgn.

```
tapply(stroke$age, stroke$dgn, median, na.rm = TRUE)
```

```
ICH  ID INF SAH
 68  81  70  60
```

1.8 Använd funktionen table() för att göra en korstabell över variablerna dgn och sex i stroke

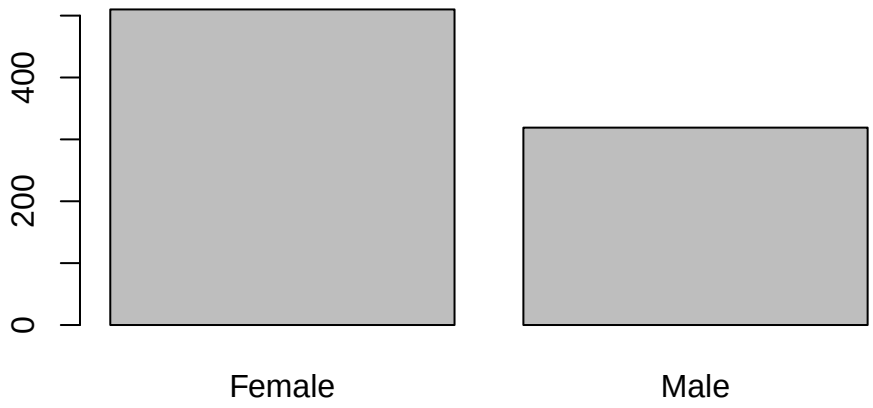
```
table(stroke$sex, stroke$dgn)
```

```
        ICH  ID INF SAH
  Female  48 140 295  27
  Male    31  62 206  20
```

Använd valfria data i stroke för att skapa grafer:

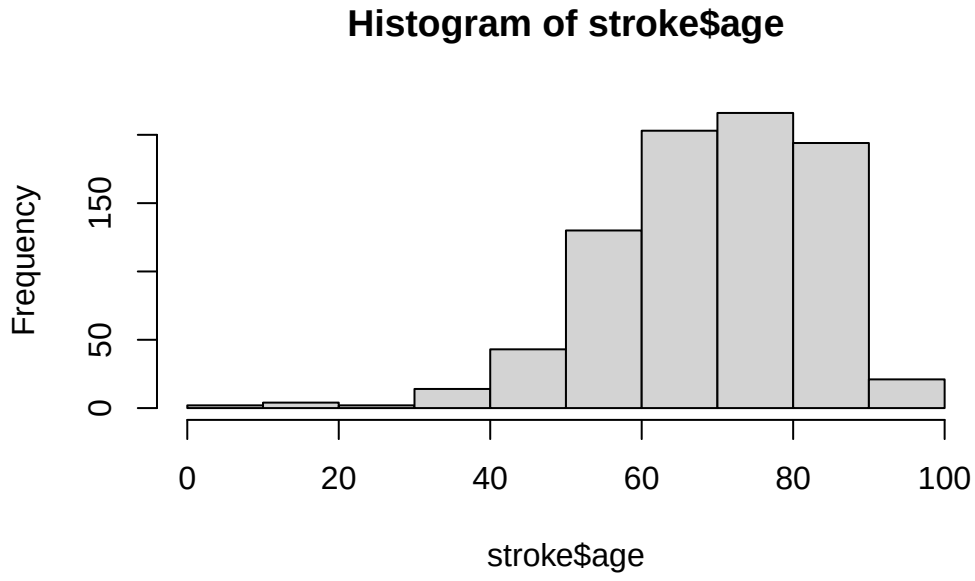
2.1 stapeldiagram

```
barplot(table(stroke$sex))
```



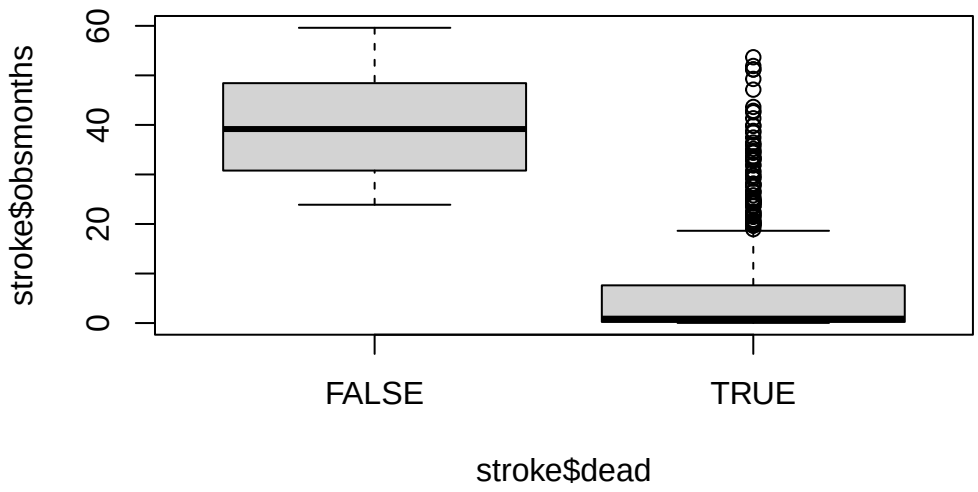
2.2 histogram

```
hist(stroke$age)
```



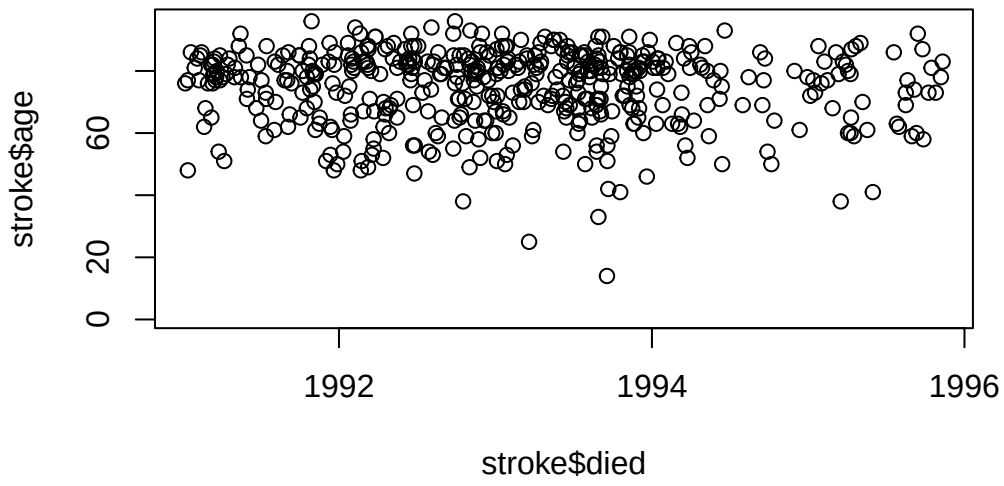
2.3 box plot

```
boxplot(stroke$obsmonths ~ stroke$dead)
```



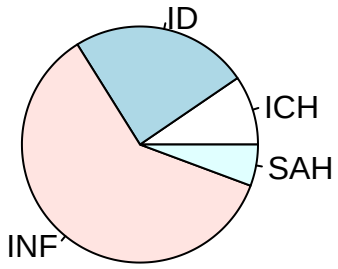
2.4 spridningsdiagram (scatter plot)

```
plot(stroke$died, stroke$age)
```



2.5 cirkeldiagram

```
pie(table(stroke$dgn))
```



Förutom att redovisa din kod och klippa in den resulterande grafen vill jag att du kommenterar kort vad varje typ av graf är lämplig för att illustrera.

Kanske valde du att besvara fråga 2.1 med den enkla koden plot(table(dgn))

I så fall får du godkänt. Men om du svarat lika enkelt på alla frågor 2.1-5 så vill jag att du utforskar några möjligheter att göra en mer avancerad graf.

3.1 Rita plot(table(dgn)) men lägg till något flärdfullt som bakgrundsfärg, titelrad, tjockare staplar.

Välj, beräkna och motivera val av centralmått och spridningsmått för dessa variabler:

4.1 age i stroke

4.2 dgn i stroke

4.3 rnorm(10000)

Om du vet vad rnorm betyder – testa annars ?rnorm() – så är det lätt att argumentera för det rätta svaret. Men jag vill påminna här att du även kan titta på data innan du bestämmer dig. Kanske dessa metoder kan vara till nytta:

hist(rnorm(10000))

boxplot(rnorm(10000))

Fördelningar

För att ha glädje av avsnittet fördelningar i inlämningsuppgiften måste du ha grunderna klara för dig. Läs i Lind eller någon anna text om diskreta och kontinuerliga fördelningar. Du kan fokusera på exemplen binomialfördelning (sid. 184-193) och normalfördelning (214-229).

Bionomialfördelningen

Passar för att beskriva fördelningar där man upprepar ett experimentet ett antal gånger (i R heter det size) Experimentet kan få två olika utfall, dessa brukar betecknas success och failure. Sannolikheten för success (i R heter det prob) förblir densamma vid varje nytt försök.

Exempel:

Två patienter får en behandling som har sannolikheten P(bota) = 0,7 att bota sjukdomen.

5.1 Använd min kod nedan för att med binomialfördelningen beräkna och plotta sannolikheterna att 0 eller 1 eller 2 patienter botas. Lägg till en lämplig titel till grafen.

x <- 0:2

plot(x, dbinom(x, 2, 0.7), type = “h”, col = “red”, lwd=10)

Det är bra att vid ett tillfälle beräkna sannolikheterna för varje utfall för hand. Formeln finner du på sid 57 i ISwR. Här följer formeln med värdena för utfallet 0 tillfrisknar ifyllda.

P(x=0) =( ) 0,70 * 0,32 =

5.2 Beräkna i R med formeln ovan sannolikheterna att 0 eller 1 eller 2 patienter botas, rita en graf. Använd funktionen choose(). Blev det samma resultat som i 5.1?

För att lösa uppgiften kan du behöva komplettera texten i ISwR med en annan text om binomialfördelningen. Om du har läst tidigare räcker det kanske att läsa igenom formler som jag klippt ur wikipedia binomial distribution:

Probability mass function

for $k = 0, 1, 2, \ldots, n$, where

5.3 Sex patienter får samma behandling. Sannolikheten att tillfriskna har visat sig vara 80%.

Rita ett stapeldiagram som visar sannolikheten för varje tänkbart utfall från noll patienter tillfrisknar till sex patienter tillfrisknar. Använd dbinom().

5.4 Beräkna sannolikheten att exakt två patienter tillfrisknar.

5.5 Beräkna sannolikheten att minst 5 patienter tillfrisknar.

Normalfördelnigen

Normalfördelnigen är ett mycket användbart verktyg inom statistiken.

6.1 Rita en standardiserad normalfördelning med medelvärdet noll och standardavvikelsen ett med mitt kodförslag nedan:

x <- seq(-4,4,0.05)

plot (dnorm(x))

eller

curve(dnorm(x), from= -4, to=4)

Funktionen dnorm(x), som i density function, returnerar sannolikheten att i ett experiment få ett ufallet x eller mycket nära x

6.2 Rita en graf som beskriver fördelningen av blodtryck i mmHg för människor. Variabeln ska vara normalfödelad med medelvärdet 90 och standardavvikelsen 10.

Funktionen pnorm(x), som i probability function, returnerar sannolikheten att i ett experiment få ett utfall x eller lägre än x.

Använd pnorm() för att beräkna andelen människor i världsbefolkningen som har ett blodtryck:

6.3 80 eller lägre

6.4 100 eller lägre

6.5 högre än 100

6.6 högre än 90 (stanna till här och tänk efter om resultatet för 6.6 verkar stämma, kommentera)

6.7 Av 50000 svenska män födda 1980, hur många är 190 cm eller längre? Utgå ifrån medellängd 180 cm och standardavvikelse 7 cm. (Värdena är påhittade.)

6.8 Hur många män födda 1980 är mellan 180 och 190 cm långa?

Funktionen qnorm(), som i quantile function, kan du använda om du söker ett värde för en normalfördelad variabel som är högre eller lika med en viss andel av alla observationer.

Nedan har jag använt qnorm() för att beräkna det intervall av blodtryck som innesluter 95% av befolkningen.

> qnorm(c(0.025, 0.975), 90, 10)

[1] 70.40036 109.59964

6.9 Justera min kod så att R returnerar blodtrycksintervallet angett i heltal.

Nedan har jag ritat en graf som visar fördelningen av blodtryck i befolkningen. Medel 90 mmHg, standardavvikelse 10. Dessutom har jag ritat in gränserna som innesluter 95% av befolkningen.

6.10 Återskapa min graf med R kod.

Den sortens beräkningar du har gjort med normalfördelningen är mycket vanliga. Därför finns så kallade z-tabeller publicerade. De anger ofta sannolikheten, p , att få ett utfall z eller lägre än z, där z är normalfördelad med medelvärdet 0 och standardavvikelsen 1.

Testa att googla z table och titta på bildresultaten, så hittar du z-tabeller i olika utformningar.

Du kan använda länken nedan om du vill, men lova då att klura på hur adressen är utformad, det kan du ha glädje av någon annan gång när du hämtar sidor automatiskt.

https://www.google.com/search?q=z+table

6.11 Skriv några rader i R som skapar en kolumn med p-värden, från z= -4 till z = 0

Välj ett antal värdesiffror som du tycker verkar vara vanligt i z-tabeller. Vi bryr oss inte om smärre avvikelser i avrundning.

Varför är normalfördelningen användbar?

Nu har du räknat ett antal exempel med normalfördelningen. Men varför är normalfördelningen så användbar egentligen? Jag vill att du gör två experiment som illustrerar varför vi har stor glädje av denna fördelning. Det första utgår ifrån ett resonemang om biologi och binomialfördelningen, det andra är mer matematiskt och handlar om centrala gränsvärdessatsen.

Många fenomen i naturen beror på ett stort antal underliggande faktorer som är oberoende av varandra. Ofta duger det bra att anta oberoende även i fall där det finns små kopplingar mellan

faktorer. Tänk till exempel att du ska skapa detaljerad modell av kroppslängd. Troligen kan du hitta påverkan från ett stort antal gener, vi kan gissa på 200. Dessutom kan du säkert hitta ett så stort antal yttre faktorer att det verkar rimligt att klassa dem som oberoende för vårt resonemang. Mammas nutritionstillstånd under graviditeten. Mamma rökte eller ej. Min poäng här är bara att listan kan göras lång.

Tänk dig en biologisk variabel x. Du kan simulera ett värde för x genom att singla slant ett antal gånger och lägga samman resultatet: varje krona ger 1 poäng, varje klave ger 0 poäng. Använd binomialfördelningen för att se hur x fördelas i popultaionen om variabeln styrs av två underliggande faktorer. (7.1)

Jämförelsen haltar något, eftersom vi inte har med koncepten recessiv och dominant i vår enkla modell, men för att konkretisera kan du tänka dig en enklare egenskap som ögonfärg. Mycket förenklat kan den anta några få olika lägen: blå, grön, brun. Och den styrs av få underliggande faktorer: ett par gener.

7.1 Rita upp binomialfördelningen för att singla slant två ggr med följande kod.

Rita in en normalfördelning i samma graf.

x <- 0:2

plot(x, dbinom(x, 2, 0.5), type = “h”, col = “blue”, lwd=4, ylim= c(0,0.6))

7.2 Rita nu upp en variabel som styrs av 8 underliggande oberoende faktorer

Rita in en normalfördelning i samma graf.

7.3 Rita nu upp en variabel som styrs av 30 underliggande oberoende faktorer

Rita in en normalfördelning i samma graf.

7.4 Kommentera resultatet av undersökningen 7.1 till 7.3

Frivillig uppgift: Det finns ett elegant sätt att välja standardavvikelse för normalfördelningen som du ritar in över binomialfördelningen i graferna. Hur kan man beräkna ett lämpligt värde? Tips: Läs Lind formel [6-5]

Centrala gränsvärdessatsen (Central Limit theorem)

Att annat skäl till att normalfördelningen är så användbar inom statistiken följer av centrala gränsvärdessatsen. Tänk att du drar många lika stora stickprov ur en population och varje gång beräknar stickprovets medelvärde. Vi ska nu undersöka hur stickprovets medelvärde fördelas.

8.1 Skapa ett antal fördelningar NORM, UNIF, SKEV med koden nedan

Använd gruppens nummer som seed, tex grupp A = seed(1), grpp B, seed(2)

set.seed(400)

NORM <- rnorm(10000)

UNIF <- runif(10000)

SKEV <- rep(1:100, 1:100)

8.2 Rita histogram över fördelningarna och beräkna medelvärde och standardavvikelse (även om man kan ifrågasätta iden att räkna medel och standardavvikelse för SKEV)

8.3 Börja arbeta med NORM. Tag ur NORM 1000 stickprov med återläggning av storleken n=3, tag sedan 1000 stickprov av storleken n=6, tag slutligen 1000 stickprov av storleken n=300.

Beräkna medelvärde för varje stickprov och rita histogram över medelvärdena (du har gjort en empirisk samplingsfördelning). Beräkna medelvärdet och standardavvikelsen för dina samplingsfördelningar n=3, n=6, n=300 .

8.4 Upprepa uppgift 8.3 med UNIF och till sist med SKEV

8.5 Formulera centrala gränsvärdessatsen (CGS), citera gärna ur en bok (tex Lind) eller från internet. Kommentera resultatet i 8.2 till 8.4 med hjälp av CGS.

frivillig uppgift 8.6 Om du har läst statistik tidigare eller vill fördjupa lite här förelår jag att du undersöker standardavvikelsen i populationen NORM <- rnorm(10000) och hur den förhåller sig till standardavvikelserna i de tre samplingsfördelningarna du skapat ur NORM. Kan du hitta en formel som visar förhållandet mellan populationens standardavvikelse och samplingsfördelningens standardavvikelse?

frivillig uppgift 8.7 Är din kod bra eller dålig? Kvalitet kan vara att koden är

-lätt att läsa och förstå, rikligt kommenterad

-kort

-snabb

Om snabbhet räknas kan du använda system.time() i ditt arbete att optimera kod. Använd koden nedan för att mäta tiden för att räkna x + y på två olika sätt. Pröva att mäta din egen kod med system.time()

fyll x och y med värden, skapa en tom vector z

x <- rnorm(100000)

y<- rnorm(100000)

z <- rep(NA, 100000)

mät hur lång tid det tar att fylla z med x + y på ett omständigt sätt

system.time({

for (i in 1:100000) {

z[i] <- x[i] + y[i]

}

})

mät hur lång tid det tar att räkna x + y på ett snabbt sätt

system.time(k <- x + y)