Datorlaboration 8

Josef Wilzén

10 mars 2014

Instruktioner

- Denna laboration ska göras i **grupper om två personer**. Det är viktigt att att följa gruppindelningen och inte ändra grupper. Om det är problem med grupperna så ska ni prata med Josef eller Måns.
- Det är tillåtet att diskutera med andra grupper, men att plagiera eller skriva kod åt varandra är **inte tillåtet**.
- Utgå från laborationsfilen som går att ladda ned **här**
- Laborationen består av två delar:
  - Datorlaborationen
  - Inlämningsuppgifter
- Innan du lämnar in laborationen:
  1. Starta om R-Studio eller rensa den globala miljön (Global environment) med `rm(list = ls())`.
  2. Ladda in funktionerna i R med `source`.
  3. Kontrollera att inget annat än funktionerna laddas in.
  4. Testa att funktionerna fungerar en sista gång.
- Deadline för labben framgår på **kurshemsidan**

Innehåll

Parprogrammering

Tanken är att ni ska öva på parprogrammering under labb4 till labb8.

- Detta innebär att två personer samarbetar och tillsammans löser programmeringsproblem vid en dator.
- Personerna turars om att ha rollerna:
  - **Föraren:** har kontroll över tangentbordet och skriver koden
  - **Navigatören:** Är delaktig i problemt genom att kommentera, diskutera och analysera koden som skrivs. Den här personen ska **inte** sitta och titta passivit.
- Det är viktigt att byta roller **ofta** och att båda personerna är involverade i lösningen av problemet. Vi rekomenderar att ni byter roller minst varje 30 min.
- Det är viktigt att kommentera sin kod så att ni båda kan förstå koden i efterhand. Tänk på skriva ner syftet med koden och inte exakt vad koden gör.
- Syftet med parprogrammering är:
  - Lära av varandra
  - Lära sig olika sätt att skriva kod (stilar) och olika sätt att lösa problem
  - Skriva mer effektiv kod med mindre buggar
  - Lära sig skriva kod som är lätt att förstå för andra

Del I
Datorlaboration

1 Mer statistik

1. Skapa nu arrayen nedan:

```
x <- array(1:27, c(3, 3, 3))
```

   (a) Välj ut det element som ligger på indexplatsen x=2,y=1,z=2.

   (b) Välj ut det element som ligger på indexplatsen x=3,y=3,z=2.

   (c) Välj ut det element som ligger på indexplatsen alla x och y=1,z=2.

   (d) Välj ut det element som ligger på indexplatsen alla y och x=3,z=1.

   (e) Testa nu att välja ut den "skiva" med data som motsvarar z-dim=2

   (f) Testa nu att välja ut den "skiva" med data som motsvarar x-dim=1

   (g) Testa nu att välja ut den "skiva" med data som motsvarar y-dim=3

   (h) Använd `apply()` för beräkna medvärden över z-dim. Vi vill alltså få en matris som innehåller medelvärdet från alla matriser med data i z-dem. Tänk på att MARGIN= kan ta en vektor som argument (alltså flera dimensioner). Resultatet ska bli.

```
     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18
```

   (i) Använd `apply()` för beräkna medvärden över x-dim.

   (j) Att fundera över: När skulle det kunna vara bra att använda array i en riktig datanalys?

2. Läs in google-data från labb5.

   (a) Skatta en linjär regressionsmodel där `Close` är responsvariabel och `Open` är föklarande variabel. Gör en scatterplot mellan Open och Close och lägg till regressionslinjen till plotten. Använd `summary()` på ert `lm`-objekt. Kör ?lm.summary() och läs under "Value" hur ni kan välja ut p-värdet för β_1. Är det större än 0.05? Plotta residualerna i ett histogram. Tips: `lm(), abline(), residuals()`

(b) Skatta en linjär regressionsmodel där `Close` är responsvariabel och `Open` och `High` är föklarande variabler. Använd `summary()` på ert `lm`-objekt. Plotta residualerna i ett histogram.

(c) Skatta en linjär regressionsmodel där `Close` är responsvariabel och alla övriga variabler i google är föklarande variabler. Går det att ange "`fomula=`" för modellen **utan** att explicit skriva ut alla namnen på de förklarande varaiblerna? Använd `summary()` på ert `lm`-objekt. Plotta residualerna i ett histogram.

3. Kör koden nedan. Vad är egentligen skillnaden mellan de två plottarna? Skatta en linjär regressionsmodell, där `z` är den beroende variabel (y) och `x` är förklarande variabel (x).

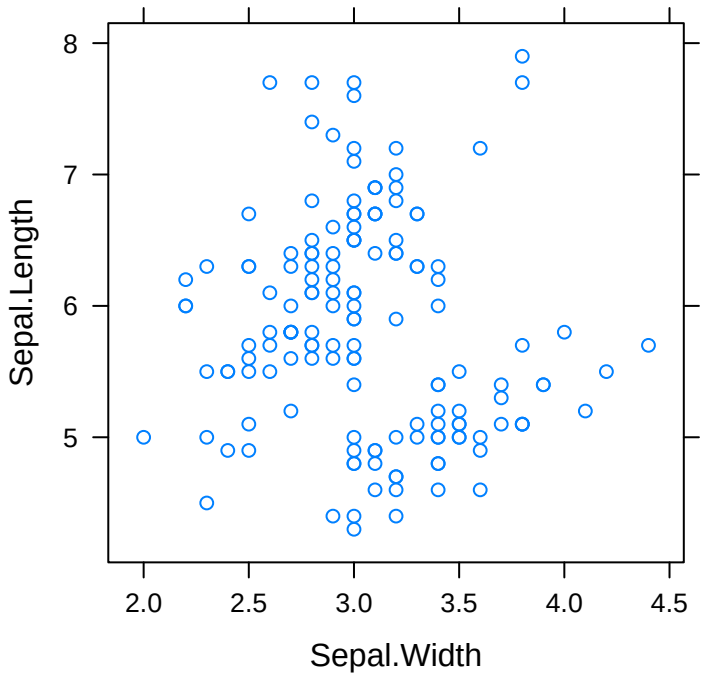
```
x <- 1:120
set.seed(12422)
monthDiff <- rnorm(12, sd = 10)
y <- 10 + 0.46 * x + monthDiff + rnorm(120, sd = 2)
# plot 1:
plot(x, y, type = "l")
z <- y - monthDiff
# plot 2:
plot(x, z, type = "l")
```

2 Mer grafik

2.1 lattice package

1. Ladda in lattice paktet i er R-session.

2. Ladda in iris data med `data(iris)`. För info kör `?iris`. Skapa plotten nedan med `xyplot()`.



3. Gör om koden i 2 så att gruppvisa erhålls beseras på värdet på variablen `Species`.

   (a) Färgkoda grupperna i en plot. Tips: "`groups=`"

   (b) Gör en subplot för varje grupp. Tips: "`|`"

4. Dela upp variabeln `Petal.Length` i två överlappande interval med funktionen `equal.count()`. Döp den till `Petal.Length.cut`.

5. Använd `Petal.Length.cut` för att gruppera plotten som skapas i 2 i två olika plottar.

6. Kör koden nedan, ändra den sedan så att du får ett histogram/tätheteskurva för varje grupp i `Species`.

```
histogram(~Sepal.Length, data = iris)
densityplot(~Sepal.Length, data = iris)
```

7. Kör koden nedan för att göra två olika 3D-plottar. Vad är skillnaden mellan dessa? Ändra så att plottarna innehåller en annan kombiantion av

variabler från iris-data.

```
cloud(x = Sepal.Length ~ Sepal.Width * Petal.Length, data = iris)
cloud(x = Sepal.Length ~ Sepal.Width * Petal.Length, data = iris, groups = Species)
```

8. Kör `example(cloud)` för att se några exempel på hur några 3D-plottar kan skapas.
9. Kör koden nedan. Ändra på konstanterna `a`, `b` och `c`, och notera hur plotten ändras.

```
x <- 1:10
y <- 1:10
a <- -2
b <- 3
c <- 1
mySurface <- expand.grid(x = x, y = y)
mySurface$z <- a * mySurface$x + b * mySurface$y + c
wireframe(x = z ~ x + y, data = mySurface)
```

2.2 ggplot2 package

1. Ladda in ggplot2 paktet i er R-session.
2. Läs in HUS data från labb 6 i din R-session och spara som `hus`. Resa bort de hus med extremt höga priser med koden nedan:

```
index <- hus[, 1] < quantile(hus[, 1])[4]
hus <- hus[index, ]
```

3. Kör koden nedan. Testa sedan att göra liknande plottar, fast med andra variabler från HUS data.

```
# scatter plot:
qplot(Bostadsyta, Försäljningspris, data = hus)
qplot(Bostadsyta, Försäljningspris, data = hus, color = as.factor(Antal.sovrum))
qplot(Bostadsyta, Försäljningspris, data = hus, geom = c("point", "smooth"))
# histogram:
qplot(Försäljningspris, data = hus)
qplot(Försäljningspris, data = hus, fill = as.factor(Antal.sovrum))
```

4. Skapa nu följande ggplot-objekt.

```
g <- ggplot(hus, aes(Bostadsyta, Försäljningspris))
```

   (a) Addera ett lager som ger er en scatter plot. Ändra färgen så att punkterna blir röda. Tips: `geom_point()`

   (b) Addera ytterligare ett lager som som anpassar en "smoother" (kurva) , testa både en linjär med method="lm" och icke-linjär ned method="loess". Tips: `geom_smooth()`

   (c) Addera ytterligare ett lager som ger grupperade subplots. Testa att gruppera både på `Pool` och `Antal.badrum`. Hur gör ni om subplots ska vara radvis? Eller kolumnvis? Tips: `facet_grid()`

   (d) Addera ytterligare ett lager som ändrar det övergripande temat. tips: `theme_bw(), theme_bw(base_family="Times")`

   (e) Använd funktionerna `xlab(), ylab(), labs()` och `ggtitle()` för att ändra på axeltexterna och titeln.

5. Välj två numeriska variabler från `iris`-data och gör om föregående uppgift, gruppering görs nu på variabeln `Species`.

6. Kör koden nedan. Vad skiljer fallen med `ylim(-3,3)` och `coord_cartesian(ylim=c(-3,3))`?

```
# gränser för axlarna:
testData <- data.frame(x = 1:50, y = rnorm(50))
testData[2, 2] <- 57
# outlier plot(testData[,1],testData[,2],type='l',ylim=c(-3,3))
b <- ggplot(testData, aes(x = x, y = y))
b + geom_line()
# tar bort data som ligger utanför gränsen
b + geom_line() + ylim(-3, 3)
# indkluderar data som ligger utanför gränsen
b + geom_line() + coord_cartesian(ylim = c(-3, 3))
```

3 Intro till spatiala data

Nu ska ni testa att göra en enkel spatial analys.

1. Läs in paktet `maptools` i er R-session.

2. Ladda ner filen "Lan_SCB.zip" från kurshemsidan. Packa sedan upp den i mappen som är ert workning directory. Det är viktigt att alla filerna finns tillgängliga där.

3. Läs in data med följande kod:

```
# läsa in data
sweMap <- readShapePoly("Lan_SCB_07")
```

4. `sweMap` innehåller nu ett objekt som representerar en karta på länsnivå. Titta på kartan med `plot()`.

5. Kör koden nedan:

```
names(sweMap@data)
sweMap@data
temp <- sweMap@data
```

6. Ladda ner datasetet "bokbussar.csv" från kurshemsidan. Datasetet innehåller information om hur många bokbussar det finns i olika län i Sverige. Spara datasetet som `bokbuss`.

7. Tanken är nu att ni ska slå samman `temp` och `bokbuss`, för att senare kunna plotta upp antalet bokbussar på er karta. Detta görs förslagsvis med `merge()`, kolla här eller i hjälpen hur funktionen fungerar. Nedan är ett exempel på hur data kan slås ihop:

```
library(stringr)
ID <- factor(str_sub(bokbuss$region, start = 1, end = 2))
bok <- data.frame(bokbussar = bokbuss$X2012, ID = ID)
temp2 <- base::merge(x = sweMap@data, y = bok, by.x = "LNKOD", by.y = "ID")
```

8. Spara sen ert sammanslagna data på följande sätt:

```
sweMap@data <- temp2
```

9. Plotta nu er karta:

```
spplot(sweMap, "bokbussar")
```

Del II
Inlämningsuppgifter

4 Miniprojekt

Inlämningsuppgifterna i labb8 är ett miniprojekt, det innebär ni som grupp kommer att ha en del frihet i hur ni utför datanalysen som beskrivs nedan. Generellt så ska ni de er data (se nedan) och utföra en enkel dataanalys, genom att ta fram beskrivande statistik (såsom medelvärde, median, standardavvikelse, frekvenstabeller mm), göra tester konfidensintervall, olika grafer och linjär regression. Ni får anväda vilket grafiksystem ni vill för att göra plottarna i projektet.

4.1 Sammanställning av resultatet

Lägg upp arbetet på följande sätt. Alla funktioner som ni skapar själva ni spara filen `LabD8_function.R`. Övrig kod ska ni lägga i `LabD8_analysis.R`. Ni ska sedan läsa in funktionerna i `LabD8_function.R` med source() i från filen `LabD8_analysis.R`.

Ni ska sätta samman era resultat till en kort rapport på följande sätt:

- Öppna ett Word-dokument (eller liknande)
- Skriv in era **namn, LiuID och gruppnummer** i sidhuvudet.
- Beskriv kort vad ni har för data, och vad de olika variablerna innehåller.
- Kopiera in era resutlat (plottar, tabeller mm) i dokumentet. Separera olika avsnitt med lämpliga rubriker.
- Skriv en kort kommentar till varje resultat, det räcker med 1-2 meningar.
- Spara dokumentet som en **pdf-fil**.

4.2 Data

Ni ska ha tre olika dataset. Vilka data som ska användas beskrivs i labb 7, i sektion 3 "Inför labb 8 ". Om ni inte följt dessa instruktioner så ska ni göra dem innan ni forstätter med dessa uppgifter. Tänk på att variabler som är frekvensdata (tex antal invånare) kan i praktiken ses som en numerisk (intervall eller kvotdata) om den kan anta många olika värden.

Nu ska ni ha tre dataset, två stycken med kommundata och ett med tidseriedata. Dataseten med kommundata ska ni ha slagit samman, vilket ger er två dataset:

- Kommundata: Innehåller minst 4 variabler med data för alla Sveriges kommuner (eller i alla fall en majoritet av kommunerna), plus i variabel för kommun. Spara som en kommaseparerad csv-fil med nament "kommun.csv"

- Tidsseriedata: Innehåller en variabel av intesse som observerats i minst 36 månder, med tillhörande tidsvarabel. Spara som en kommaseparerad csv-fil med nament "tidserie.csv"

4.3 Dataanalys av kommundata

Ta fram beskrivande statistk för **alla** variabler i data. Beroende på hur data ser ut så kan det vara medelvärden, frekvenstabeller mm. Ni kan göra relevanta transformationer av era variabler om ni vill, tex göra en numeriska variabel till en binär och räkna med andelar. Ett annat exemepel: om ni har med folkmängd, dela in kommunerna i stora, medelstora och små när det gäller befolkning.

Följande saker ska ni göra med data:

1. Ta fram minst en scatterplot mellan två variabler
2. Ta fram minst ett histogram/stapeldiagram
3. Ta fram minst en scatterplot/histogram/stapeldiagram som är grupperat på en annan variabel i minst två grupper.
4. Göra minst ett hypotestest, där ni ställer upp en nollhypotes och sen testar om ni kan förkasta den. Beroende på hur er data ser ut så kan det vara ett t-test, test av andelar eller ett chitvå-test/fishers test. I rapporten ska ni skirva upp både nollhypotesen och mothypotesen, ange även p-värdet.
5. Beräkna minst ett konfindensintervall, antingen för medelvärde eller för andel.
6. Om data är numerisk så ska ni beräkna korrelationer mellan alla sådana variabler.
7. Beräkna två olika linjära regressionsmodeller. Ni väljer en responsvariabel (y-variabel) ni tror kan bero/ha ett samband med de andra variablerna. Välj sedan ut två olika variabler som ni kan ha som förklarande variabler (x-variabler). Skatta sedan två regressionsmodeller med de två olika förklarande variablerna i var sin regressionsmodell, dvs om era föklarande variabler heter X_1och X_2, så blir modellerna: $y = \beta_0 + \beta_1 * X_1$ och $y = \beta_0 + \beta_1 * X_2$. Det är okej att använda frekvensdata i regressionmodellerna (se 4.2). Följande saker ska vara med från modellerna:
   (a) Scatterplot mellan föklarande variabel och responsvariabel, tillsammans med skattad regressionlinje.
   (b) Histogram över residualerna (felen)
   (c) Hurvida lutningen på regressionslinjen är signifikant skild från noll, dvs om p-värdet för β_1i de olika modellern är mindre är 0.05.

8. Spatial analys på kommunnivå: ni ska ha med minst en sverigekarta där ni plottar någon beskrivande statistk för en variabel över kommunerna. För tips kolla på Intro till spatiala data. Ni behöver ladda ner "Kommun_SCB.zip" och packa upp den för att få tillgång till en karta på kommunnivå.

4.4 Dataanalys av tidseriedata

Låt `X` vara er variabel i datasetet. Utför nu följande:

1. Gör en linjeplot mellan `X` och er tidsvariabel. Skalan på x-axeln ska vara en lämplig tidsskala.
2. Beräkna månadsmedelvärden, spara dessa i `monthMeans`.
3. Använd funktionen `summary()` för att fram beskrivande statistik för varje år (det ska vara minst tre år i data)
4. Subtrahera månadsmedelvärden från X, så ni tar bort säsongsvariationen i data. Månadsmedelvärdet för januari ska subtraheras från alla januarivärden i data, och likadant för de andra månaderna. Spara den nya tidserie som `newX`.
5. Gör en linjeplot mellan `newX` och tid.
6. Skatta en linjär regressionsmodell mellan `newX` och dess tidsindex. Rapportera följande:
   (a) Scatterplot (punkter) mellan föklarande variabel och responsvariabel, tillsammans med skattad regressionlinje.
   (b) Histogram över residualerna (felen)
   (c) Hurvida lutningen på regressionslinjen är signifikant skild från noll, dvs om p-värdet för β_1i de olika modellern är mindre är 0.05.
7. Verkar det finnas någon trend i data? Dvs ökar/minskar data med tiden, eller är data konstant över tid.

4.5 Inlämning

Lägg följande filer i en mapp:

- er rapport, som **pdf**
- LabD8_function.R (era funktioner)
- LabD8_analysis.R (er övirga kod)
- Er data, dvs filerna: "kommun.csv" och "tidserie.csv".

Zippa (packa ihop) er mapp till en fil och lämna in den på Lisam innan deadline. Tanken är att läraren i princip ska kunna gör om er dataanalys med dessa filer.

Nu är du klar!