Datorlaboration 8

Josef Wilzén och Måns Magnusson

22 april 2024

```
## Error in library(pxweb): there is no package called 'pxweb'
```

Instruktioner

- Denna laboration ska göras i grupper om **två och två**. Det är viktigt för gruppindelningen att inte ändra grupper.
- En av ska vara **navigatör** och den andra **programmerar**. Navigatörens ansvar är att ha ett helhetsperspektiv över koden. Byt position var 30:e minut. **Båda** ska vara engagerade i koden.
- Det är tillåtet att diskutera med andra grupper, men att plagiera eller skriva kod åt varandra är **inte tillåtet**. Det är alltså **inte** tillåtet att titta på andra gruppers lösningar på inlämningsuppgifterna.
- Använd gärna Teams för att ställa frågor. Det finns olika kanaler:
  - **Questions**: Skriv era frågor här. Svar kommer att ges öppet direkt i kanalen. Publicera inte kod till inlämningsuppgifter här (andra kan då se det). Det går bra att skriva frågor om inlämningsuppgifter här så länge ni inte inkluderar kod med lösningar till dessa uppgifter. Det går bra att publicera kod till övningsuppgifter här.
  - **Raise__your__hand**: Skriv här om ni vill ha hjälp men inte ställa er fråga öppet. Skriv något i stil med "Jag vill ha hjälp". Då kommer en lärare att kontaka er när de har tid (i chatten på Teams). Vill flera ha hjälp så bildar de olika kommentarerna en kö, och hjälp kommer att ges i ordning efter kön. En "tumme upp" på kommentaren innebär att läraren har börjat hjälpa den aktuella studenten. Ett "hjärta" på kommentaren innebär att läraren har hjälpt klart studenten.
- Använd inte `å`, `ä` eller `ö` i variabel- eller funktionsnamn.
- Utgå från laborationsmallen, som går att ladda ned **här** (obs ny mall jämfört med tidigare veckor), när du gör inlämningsuppgifterna. Spara denna som `labb[no]_grupp[no].R` , t.ex. labb5_grupp01.R om det är laboration 5 och ni är grupp 01. Ta inte med hakparenteser i filnamnet. Denna fil ska **inte** innehålla något annat än de aktuella funktionerna, namn- och ID-variabler och ev. kommentarer. Alltså **inga** andra variabler, funktionsanrop för att testa inlämningsuppgifterna eller anrop till markmyassignment-funktioner.
- Precis innan inlämning på Lisam, döp om er R-fil till en **`.txt`** fil, detta görs för att kunna skicka in filen till Ouriginal för plagieringskontroll. Exempel: `labb5_grupp01.R` blir då `labb5_grupp01.txt` Ladda upp den filen (som slutar på .txt) på Lisam under rätt inlämning innan deadline.
- Laborationen består av två delar:
  - Datorlaborationen (= övningsuppgifter)
  - Inlämningsuppgifter
- I laborationen finns det extrauppgifter markerade med *. Dessa kan hoppas över.
- Deadline för laboration framgår på LISAM
- **Tips!** Använd "fusklapparna" som finns här. Dessa kommer ni också få ha med på tentan.

Innehåll

Del I

Datorlaboration

Kapitel 1

Texthantering och regular expression i R med `stringr`

Gå igenom följande delar i *Handling and Processing Strings in R* (av Gaston Sanchez) och testa koden i exemplen.

Kap 2 Hela

Kap 3 3.1, 3.3

Kap 4 4.2.1 - 4.2.3

Kap 5 5 - 5.2.2, 5.2.6, 5.3.1-5.3.2

Kap 6 6-6.1.3, 6.2.2, 6.4-6.4.10

Kap 7 7.1, 7.2

Boken finns fritt tillgänglig här.

Kapitel 2

Modern datahantering

I följande kapitel gås de verktyg som idag är state-of-the-art för att snabbt och effektivt bearbeta stora datamängder i R.

2.1 Piping med %>%

När vi arbetar med databearbetning av stora datamaterial kan innebär det ofta ett stort antal funktionsanrop. För att göra en databearbetningsprocess överskådlig och snabb finns så kallade pipes, eller ”rör” i R för att skicka datamaterial i ett flöde av olika modifikationer. Pipingoperatorn innbär att

```
z <- a %>% fun1(b) %>% fun2(c) %>% fun3()
```

är exakt samma sak som

```
x <- fun(a, b)
y <- fun(x, c)
z <- fun3(y)
```

Detta flöde kan ofta göra det tydligare hur data bearbetas i olika databearbetningssteg.

2.2 `tidyr`

Datamaterial kan många gånger komma i ett otal olika tabellstrukturer. Alla statistiska metoder, databearbetningsverktyg som `dplyr` och visualiseringspaket som `ggplot2` kräver att datamaterialet är i ett så kallat `tidy` format. För att konvertera olika tabeller och datamaterial till ett `tidy` format används paketet `tidyr` och funktionen `gather()`.

Gå igenom och reproducera koden i följande introduktionstext.

2.3 `dplyr`

R-paketet dplyr har under kort tid att bli det huvudsakliga verktyget för att arbeta med större datamängder i R. Det finns framförallt tre anledningar till dess popularitet.

1. Paketet har bara ett fåtal funktioner för att arbeta med data vilket gör det snabbt att lära sig.
2. `dplyr` är skrivet i kraftigt optimerad C++ kod vilket gör hanteringen av stora datamängder snabbare än något annat statistik- eller analysverktyg.
3. `dplyr` kan kopplas mot databaser för att direkt bearbeta större datamängder. dplyrs verb används också i `sparlyr`, vilket är ett paket för att hantera data som inte får plats på enskilda datorer. Att lära sig `dplyr` är således en approach som möjliggör att hantera i princip hur stora datamaterial som helst.

Gå igenom och reproducera koden i följande introduktionstext.