

1. Introduksjon til R og RStudio

September 10, 2024

0.1 Introduksjon til R og RStudio

0.1.1 Hva er R?

R er et statistisk programmeringsspråk som kan brukes til dataanalyse, statistisk modellering og grafikk. Det er et kraftig verktøy for datavitenskap og maskinlæring, med en stor mengde tilgjengelige pakker og et aktivt fellesskap.

0.1.2 Hva er RStudio?

RStudio er et integrert utviklingsmiljø (IDE) for R. Det gir et brukervennlig grensesnitt for å skrive, kjøre og feilsøke R-kode. RStudio består av flere komponenter:

- **Konsoll:** Der koden kjøres
- **Editor:** Der man kan skrive og lagre R-skript
- **Miljøvindu:** Viser alle objektene som er opprettet i R-sesjonen, dvs. hva som er i dataminnnet
- **Plot-vindu:** Viser grafer og visualiseringer

0.2 R-biblioteker

Her gis en kort forklaring på hva funksjoner og pakker er i R. Det vil ikke gjennomgås hvordan man lager egne funksjoner eller pakker i dette kurset, kun hvordan man tar i bruk allerede eksisterende og installerte pakker og funksjoner.

0.2.1 Funksjoner

En funksjon i R er en blokk med kode som utfører en spesifikk oppgave. Funksjoner tar vanligvis inn én eller flere argumenter (input), utfører beregninger eller operasjoner basert på disse argumentene, og returnerer et resultat (output). Funksjoner hjelper med å strukturere koden på en modulær og gjenbrukbar måte.

For å få mer informasjon om en R-pakke kan man skrive `? <navn>` etterfulgt av navnet på pakken (f.eks. `?sum`). Da får man opp en hjelpefil med alle gyldige argumenter og eksempler på hvordan man tar i bruk funksjonen.

“Base” R-funksjoner “Base” R-funksjoner er funksjoner som er tilgjengelige direkte når du starter R, uten at du trenger å laste inn noen ekstra pakker. Disse funksjonene er en del av selve R-språket og dekker en rekke grunnleggende operasjoner, som matematiske beregninger, statistiske analyser, manipulering av datastrukturer, og mer.

Eksempler på “base” R-funksjoner:

- `sum()`: Summerer elementene i et vektor.
- `mean()`: Beregner gjennomsnittet av et sett med tall.
- `print()`: Skriver ut verdier til konsollen.

Funksjoner som krever pakker Det finnes også mange funksjoner som ikke er en del av base R, men som kommer fra eksterne pakker. For å bruke disse funksjonene må du først installere den relevante pakken, og deretter laste den inn i arbeidsområdet ditt ved å bruke `library()`.

Eksempel:

- `ggplot()` fra `ggplot2`-pakken: Brukes til avansert plotting.
- `filter()` fra `dplyr`-pakken: Brukes til å filtrere rader i et datasett.

Det er også mulig å kalle på en spesifikk funksjon fra en pakke uten å laste inn hele pakken med `library()`. Da skriver man `pakkenavn::funksjon()` og man trenger ikke å ha lastet inn pakken først. Dette kan være nyttig for å dokumentere hvilken pakke du har hentet funksjonen fra og for å unngå navnekollisjoner mellom funksjoner med samme navn fra forskjellige pakker.

Tidyverse Tidyverse er en samling av R-pakker som er designet for å jobbe sammen og gjøre dataanalyse enklere, mer konsistent og mer lesbar. Disse pakkene følger en felles designfilosofi og deler en lignende syntaks, noe som gjør det lettere å lære og bruke dem sammen.

Når du installerer og laster inn Tidyverse-pakken, får du tilgang til en rekke andre kraftige pakker som er svært nyttige for datamanipulering, visualisering, importering, og mer. Vi vil se nærmere på funksjoner fra Tidyverse senere i kurset (under “Databehandling”).

```
[ ]: # Gjør dette bare én gang dersom pakken ikke er installert
      # install.packages("tidyverse")

      # Dette må du gjøre hver gang du starter en ny R-sesjon
      library(tidyverse)
```

For å unngå å få opp oppstartsmeldinger når du laster inn pakker kan man bruke funksjonen `suppressPackageStartupMessages()`:

```
[ ]: suppressPackageStartupMessages({
      library(tidyverse)
    })
```