

# R-programmering VT2022

## Föreläsning 2

---

Josef Wilzén

2023-01-30

Linköpings Universitet

**Seminarie** på torsdag är 10:15-12:00 → se Teams!

- Sammanfattning: Föreläsning 1
- Datastrukturer:
  - Matriser
  - Data.frame
  - Listor
- Databearbetning
- Input och output (I/O)

# Sammanfattning Föreläsning 1

---

# Variabler, vektorer och typer

- Variabler använder vi för att spara värden
  - Sätts med `<-`
- Vektorer är en samling av likadana element
  - Skapas med `c()`
  - Välj element med `[ ]`
- Beräkningar med vektorer sker elementvis och cykliskt
- Värden kan vara av olika typer
  - Kollar typ med `typeof( )`
  - Byter typ med `as.`
  - Testa typ med `is.`

- En funktion utför något
- En funktion i R är uppbyggd av
  - ett funktionsnamn, t.ex. `area`
  - en funktionsdefinition: `function( )`
  - 0 eller flera argument, t.ex. `hojd` och `bredd`
  - “måsvingar” `{ }`
  - kod, t.ex. `area <- hojd * bredd`
  - returnera värde, t.ex. `return(area)`

- Logik är vanligt i programmering
  - Används i `if`-satser och indexering
- I R finns de logiska värdena `TRUE`, `FALSE`, och `NA`
- Skapas på två olika sätt
  - Som vanliga vektorer
  - Genom relationsoperatorer
- Kan användas för att välja element i vektorer

# Datastrukturer

---

- Lagring och hantering av data
- Vi kommer att diskutera:
  - Vektorer (Föreläsning 1)
  - Matriser
  - `data.frame`
  - Listor



# Matriser

---

- En tvådimensionell vektor
- Alla element har **samma** typ
- Skapas med `matrix( )`
- `+`, `-`, `*`, `/` etc. sker elementvis
- Matrisoperationer finns, kommer prata mer om det senare
- Hitta index med `[ "rad" , "kolumn" ]`
  - Om index för rad eller kolumn saknas väljs hela raden/kolumnen.

## Matriser, exempel

```
en_matris <- matrix(data = 5:8, ncol = 2)
en_matris
```

```
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
```

## Matriser, exempel

```
en_matris <- matrix(data = 5:8, ncol = 2)
en_matris
```

```
##      [,1] [,2]
## [1,]    5    7
## [2,]    6    8
```

```
en_matris[1, ] <- en_matris[2, ]
en_matris
```

```
##      [,1] [,2]
## [1,]    6    8
## [2,]    6    8
```

**data.frame**

---

- Dataset i R
- Olika kolumner kan ha olika datatyper
  - Varje kolumn är en vektor
- Indexering av variabler kan göras med variabelnamn  
["mittNamn"]
- Kan också indexera med [ "rad" , "kolumn" ]
- Finns många inbyggda datasets i paketet datasets
  - ladda in med funktionen data( )

# Exempel på inbyggt dataset



**Figure 1:** New York

Ladda in och undersök data

```
data("airquality")  
head(airquality)  
tail(airquality)  
summary(airquality)  
dim(airquality)
```

## Skapa en data.frame

```
minData <- data.frame(  
  namn = c('Johan', 'Therese', 'Hugo'),  
  vuxen = c(TRUE, TRUE, FALSE),  
  langd = c(180, 172, 110))  
minData
```

```
##      namn vuxen langd  
## 1   Johan  TRUE   180  
## 2 Therese  TRUE   172  
## 3    Hugo FALSE   110
```



- Varje kolumn är en vektor
- Kan välja en kolumn på olika sätt, följande tar fram samma kolumn.

```
minData$langd  
minData[, "langd"]  
minData[["langd"]]  
minData[, 3]  
minData[, colnames(minData) == "langd"]
```

## Nya variabler

- Lägga till en ny vektor
- Fungerar som vektorer

```
minData$langdMeter <- c(1.8, 1.7, 1.1)
minData$rolig <- "Ja"
minData
```

```
##      namn vuxen langd langdMeter rolig
## 1  Johan  TRUE  180      1.8      Ja
## 2 Therese  TRUE  172      1.7      Ja
## 3   Hugo FALSE  110      1.1      Ja
```

## Ta bort variabler

- Byt ut variabeln till NULL
- Kan också plocka bort med negativ indexering

```
minData <- minData[, -4]  
minData$rolig <- NULL  
minData
```

```
##      namn vuxen langd  
## 1   Johan  TRUE   180  
## 2 Therese  TRUE   172  
## 3   Hugo  FALSE   110
```

# Variabelnamn

- Variabelnamn är text som sparas i en vektor

```
colnames(minData)
```

```
## [1] "namn" "vuxen" "langd"
```

# Variabelnamn

- Variabelnamn är text som sparas i en vektor

```
colnames(minData)
```

```
## [1] "namn" "vuxen" "langd"
```

- Kan byta genom att skriva över värdet

```
colnames(minData)[2] <- "Inte Barn"  
minData
```

```
##      namn Inte Barn langd  
## 1   Johan      TRUE   180  
## 2 Therese      TRUE   172  
## 3   Hugo     FALSE   110
```

- Varje rad har sitt egna ID
- Alla rad IDn är en textvektor

```
rownames(minData)
```

```
## [1] "1" "2" "3"
```

- Varje rad har sitt egna ID
- Alla rad IDn är en textvektor

```
rownames(minData)
```

```
## [1] "1" "2" "3"
```

- Kan byta precis som med variabler

```
rownames(minData)[1] <- "Person 1"  
minData
```

```
##           namn Inte Barn langd  
## Person 1   Johan    TRUE   180  
## 2         Therese    TRUE   172  
## 3          Hugo    FALSE   110
```

## Listor

---



- En lista är en samling objekt
- Tänk en vektor där varje element är en låda
  - Lådan kan innehålla "vad som helst"

```
minLista <- list(namn = "Ash Ketchum",  
                c("Pikachu", "Caterpie", "Charmander"))  
minLista
```

```
## $namn  
## [1] "Ash Ketchum"  
##  
## [[2]]  
## [1] "Pikachu"      "Caterpie"     "Charmander"
```

# Indexering i listor

- Indexering görs med hakparanteser
  - För att komma åt ett eller flera objekt: [ ]
  - För att komma åt innehållet i ett objekt: [[ ]]
- Om namngivna objekt:
  - \texttt{minLista\$namn}
  - minList[["namn"]]

```
minLista[1]
```

```
## $namn
```

```
## [1] "Ash Ketchum"
```

```
minLista[[1]]
```

```
## [1] "Ash Ketchum"
```

# Databearbetning

---

- Man vill ofta kombinera olika dataset
- Vanliga sammanslagningar
  - Kombinera rader `rbind( )`
  - Kombinera kolumner `cbind( )`
  - Kombinera datasets `merge( )`
- Om man vill aggregera data används `aggregate( )`

# Input och Output

---

- Att läsa in data
  - Från filer på datorn/nätverket (.csv .xlsx .txt .Rdata .RDS)
  - Filer från webben (httr)
  - Från databaser (SQL)
  - Via något API (rOpenGov)

- Att läsa in data
  - Från filer på datorn/nätverket (.csv .xlsx .txt .Rdata .RDS)
  - Filer från webben (httr)
  - Från databaser (SQL)
  - Via något API (rOpenGov)
- För att läsa in filer i R använder vi
  - .csv och .txt
    - `read.table( )`, `read.csv( )` och `read.csv2( )`
  - .Rdata
    - `load( )`
  - .RDS
    - `readRDS( )`

- Att leverera data
  - Filer
  - Databaser/API
  - Interaktiva webbdatabaser (Shiny)
  - Rapporter/analyser/texter (knitr)
    - Detta kommer i miniprojekten



- Att leverera data
  - Filer
  - Databaser/API
  - Interaktiva webbdatabaser (Shiny)
  - Rapporter/analyser/texter (knitr)
    - Detta kommer i miniprojekten
- För att spara filer i R använder vi
  - .csv
    - `write.table( )`, `write.csv( )` och `write.csv2( )`
  - .Rdata
    - `save( )`
  - .RDS
    - `saveRDS( )`