TALL SOM FORTELLER

Introduction course

DAY 4 – 25TH JANUARY 2021

SUSIE JENTOFT & ASLAUG HURLEN FOSS

Statistisk sentralbyrå
Statistics Norway

Goals

- Better understanding of vectors, lists and dataset structure.
- Understand the value of “for” and “while” loops
- Able to write own functions
- Able to create RMarkdown documents and visualize data in different ways
- Understand how to create your own packages
- Knowledge of other packages and functions useful in production processes

Statistisk sentralbyrå
Statistics Norway

Agenda

	Monday **25th January**	**Tuesday** **26th January**	**Wednesday** **27th January**
12:00	• General: indexing, vectors and lists	• Summary • Functions	• Building packages
12:45	Exercise 7	Exercise 9	Exercise 11
13:15	Break	Break	Break
13:30	• Sorting • Control with if and else • Loops	• RMarkdown • Dashboards	• Other useful packages and resources
14:15 – 15:00	Exercise 8	Exercise 10	Discusion and summary

Statistisk sentralbyrå
Statistics Norway

Summary

- Remember library( )
- Read in files: read_csv( ) read_dta( )
- New variable: mutate( )
- Select some rows: filter( )
- Summary: summarise( )
- Plot: ggplot( ), aes( ), geom_...( )
- Draw sample: sample_n(), sample_frac()
- Validate: validator(), confront(), summary()
- Rule based imputation: modifier(), modify()
- Model based imputation: impute_<model>()
- Logging changes with «lumberjack»

Homework - plots

GitHub

- Repository is updated on **GitHub**:

  https://github.com/statisticsnorway/R_introduction_Ukraine

- In **RStudio**:
  - **Save files** you change (for example exercises) with a **new name**
  - **Press** pull to get new changes 
- Example code for today is called **Rcode_day4.R**

Vectors

- Objects which hold several values
- Must be same object type

```
vector_name <- c(value1, value2)
```

- Calculate directly on a vector
- Test a vector

```
vector_name * 2
vector_name == 2
2 %in% vector_name
```

Access an element [ ]

- Use [ ] with an index number to fetch a value
- Indexing starts from 1 in R!
- Possible to index several values
  - 1:5
  - seq(1, 5)
- Exclude an value with “-”

```
vector_name[2]
vector_name[c(2,3)]
vector_name[2:4]
```

Change a value in a vector

- Use indexing together with <-
- Same length and same type

```
vector_name[2] <- new_value
```

match

- Find the first occurence of a value and return the **index**

```
match(x, y)
```

Value to find

Vector to search in

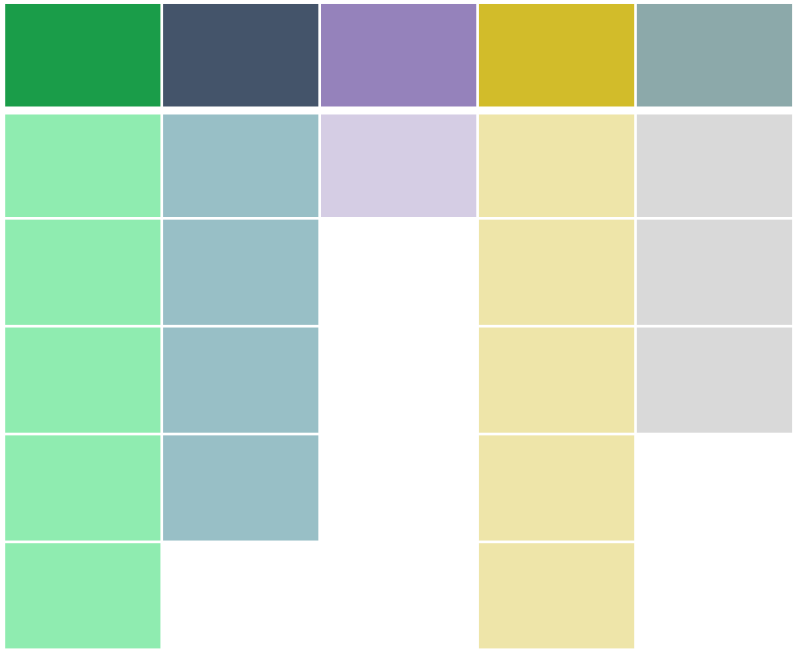
```
match(3, c(4,3,5))
```

```
2
```

- Can be used to join (quicker than join_() function)

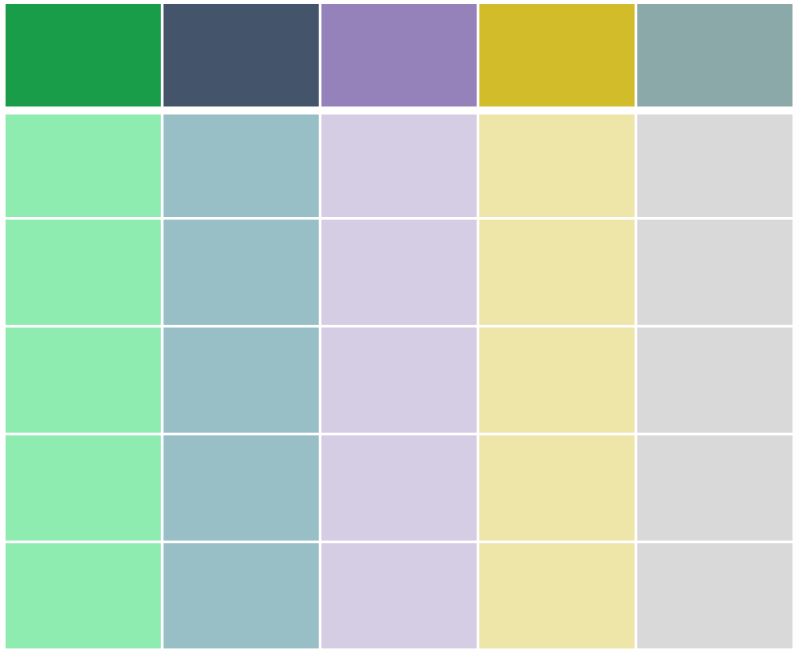
Statistisk sentralbyrå
Statistics Norway

Lists



- Created with `list()`
- Collection of vectors/numbers/datasets
- Use $ to access a vector
- Can be different lengths and types
- Use $ and [ ] together to access an element in a vector

Datasett



- Created with `data.frame()`
- Collection of vectors with the same length
- Access a variable with $

```
dataset$variable_name
```

- (Can also use [, index])
- Use $ and [ ] together to access and element

Statistisk sentralbyrå
Statistics Norway

Dataset functions

`nrow()`	Number of rows
`ncol()`	Number of columns
`names()`	Name of variables
`summary()`	Summary of dataset (min, max, average)
`head()`	First five rows
`tail()`	Last 5 rows

Dataset type

- data frame (data.frame in base) ← `as.data.frame()`
- tibble (in tidyverse package) ← `as_tibble()`
- data table (data.table) ← `as.data.table()`

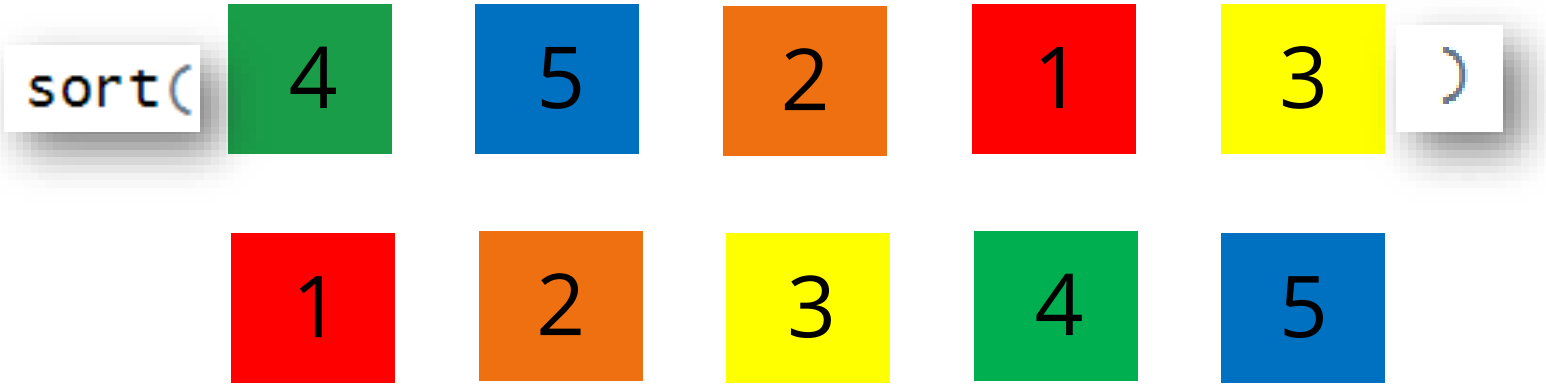
In **tidyverse** we use the **variable name** instead of **$**.
This has consequences for run time and some other limitations. However tidyverse is very intuative and good package for analysing data.

Exercise 7

- Complete exercise 7 in the file Exercises_day4.R

sort()

- Sort by number or alphabet (smallest to largest)

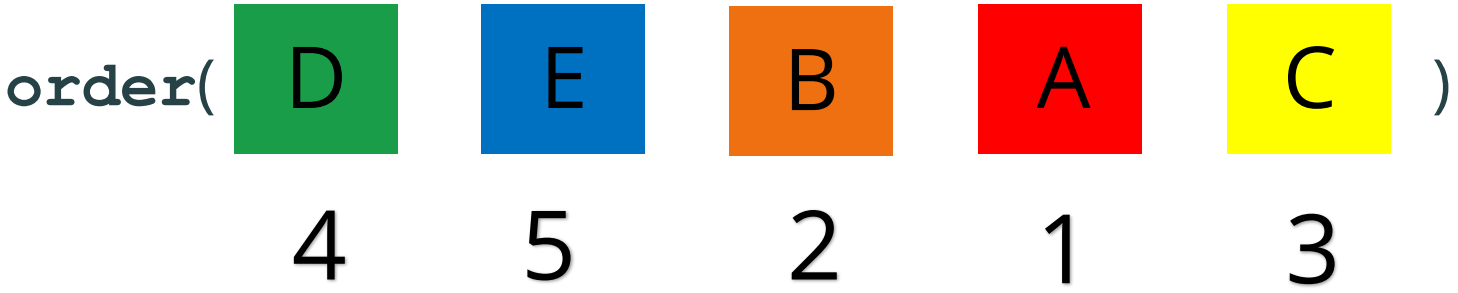


- To sort in reverse order use `decreasing = T`
- Combine with runif() for random sorting

Statistisk sentralbyrå
Statistics Norway

order()

- Returns the **index** of a sorted vector



ifelse

- Test a condition and return a value based on it

```
ifelse(sted == "Oslo", "0301", "3401")
```

If condition is false

condition

If condition is true

if...

- We use **if** to control running of processes under a specific condition

```
if (condition){
  do this ...
}
```

- Use { } for processes that are over several lines

if and else

- For controlling processes with multiple branches

```
if (condition){
  do this ...
} else {
  do this instead...
}
```

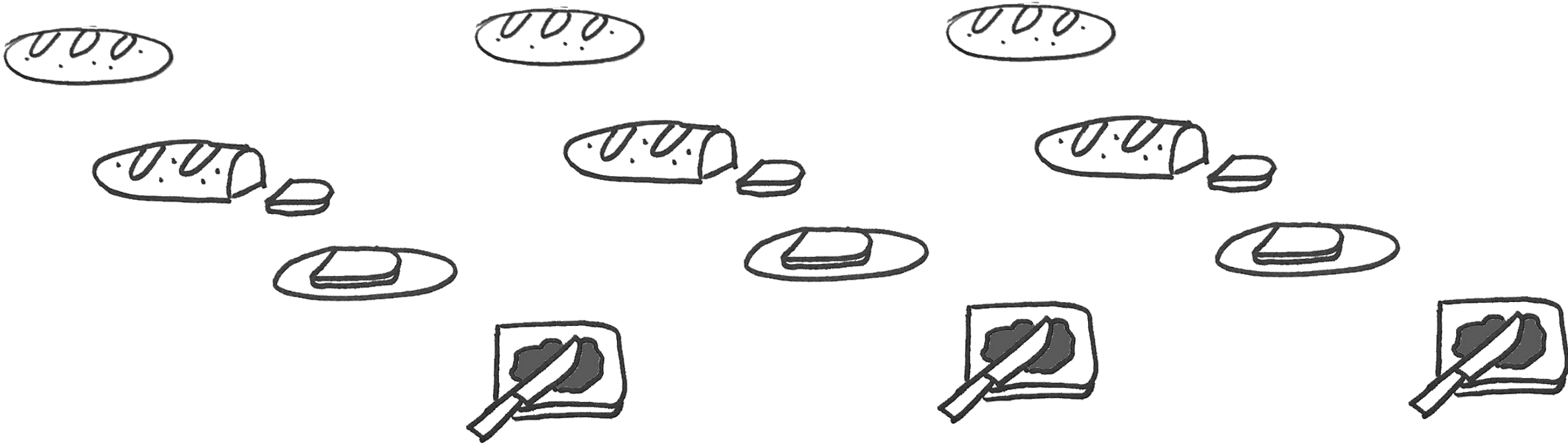
Multiple conditions

- Combine several conditions

```
if (condition1){
  do this ...
} else if (condition2){
  do this instead...
} else {
  do this with the rest...
}
```

Loops: why do we use loops?

- Make a sandwhich with jam



"for" loops

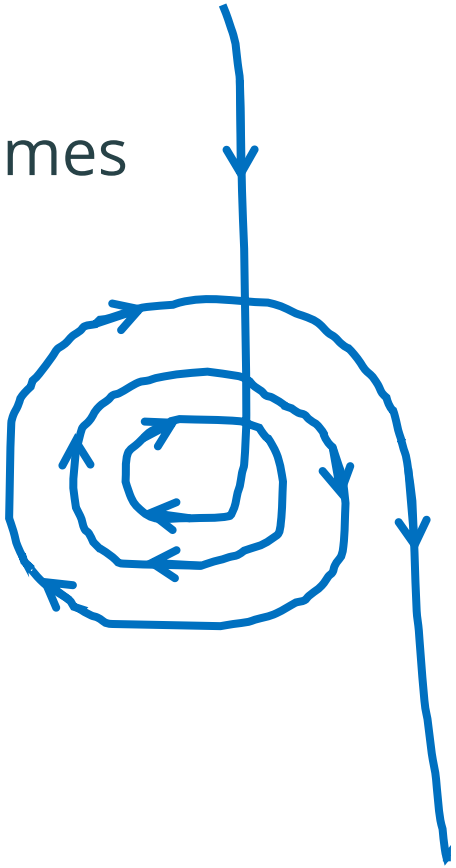
- For processes that repeat themselves a set number of times

Object name used in loop

Value the object will take

```
for (i in 1:10){
  do this with i...
}
```

What should happen



Statistisk sentralbyrå
Statistics Norway

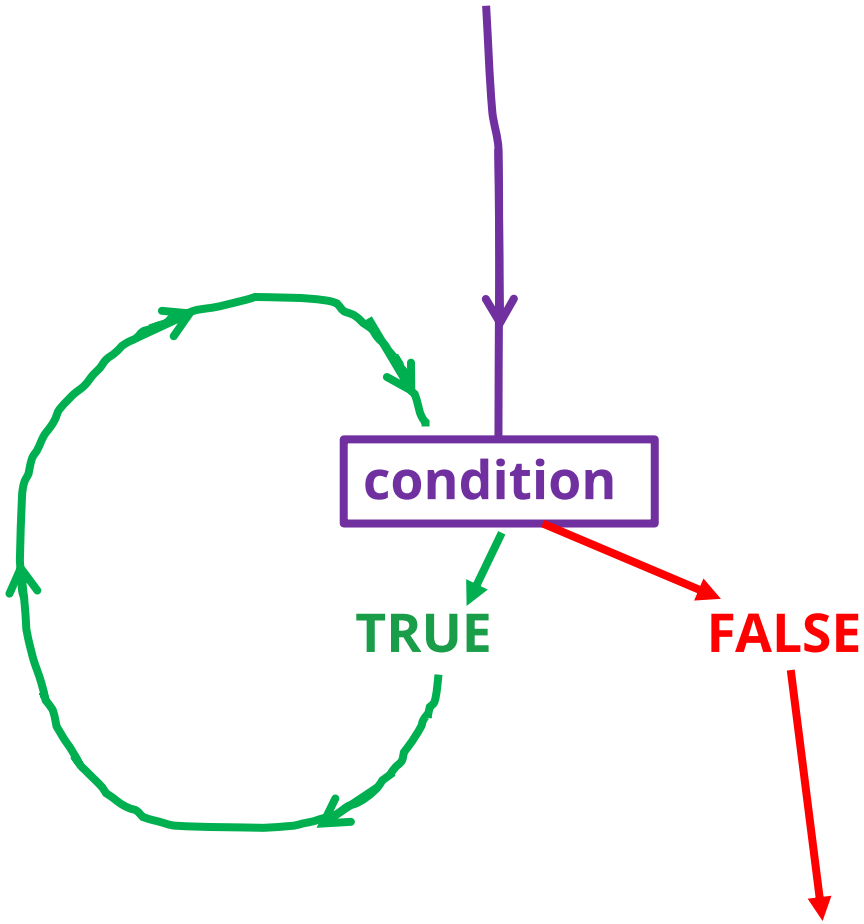
"for" loops

- Use a **sequence** (eg 1:10) to repeat x times
- Use a **vector** (eg. c("jam", "peanutbutter")) to loop with specific values

"while" loops

- Check a condition

```
while (condition){
  do this ...
}
```



- Not a specific number of repetitions

Exercise 8

- Complete exercise 8 in the file: Exercises_day4.R