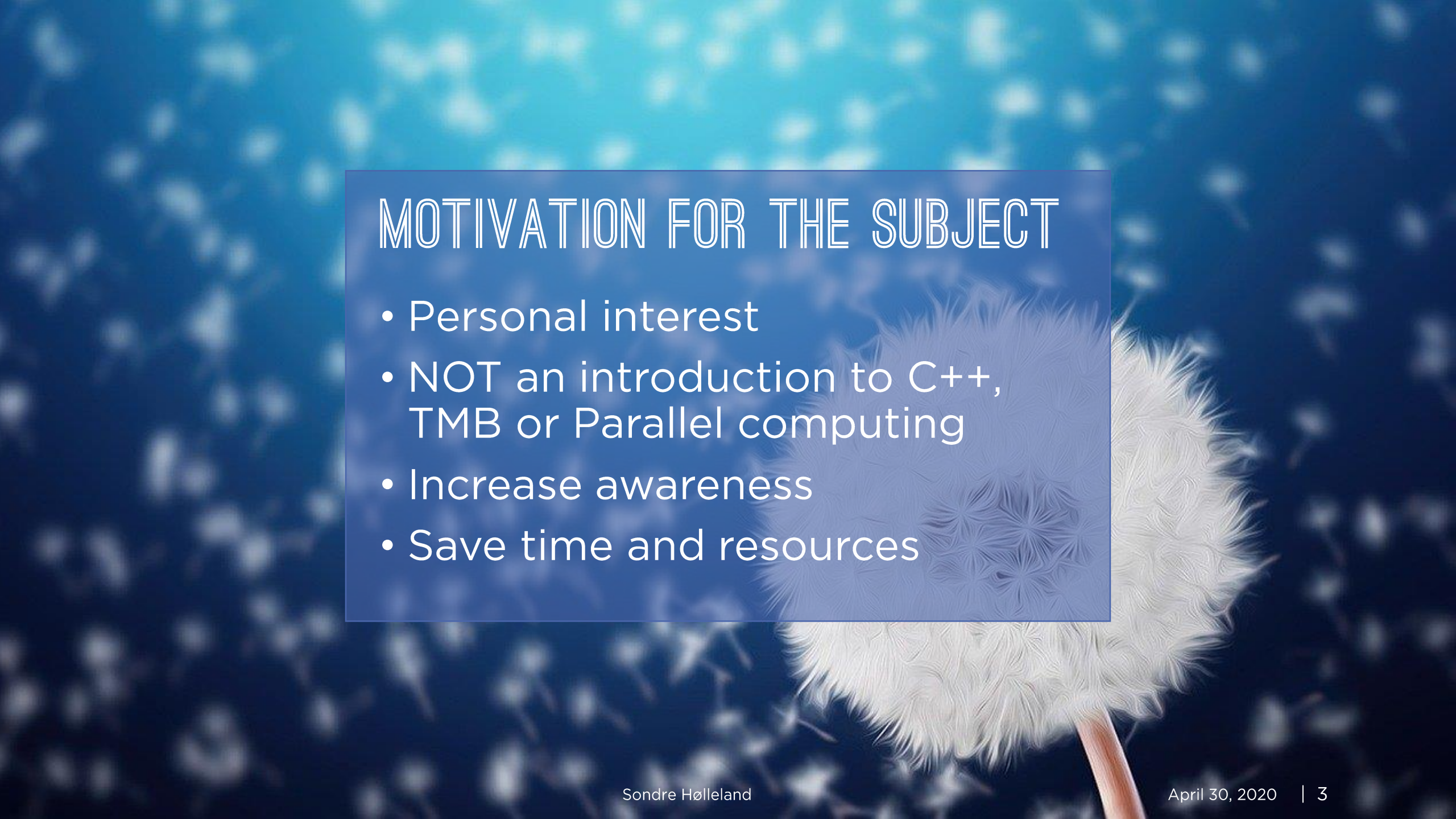

WAYS OF SPEEDING UP YOUR R
by Sondre Hølleland
Seminar on self-selected topic MAT930
April 30, 2020

TOPICS OF THE TALK

C++ WITH R

TEMPLATE MODEL BUILDER

PARALLEL COMPUTING WITH R



MOTIVATION FOR THE SUBJECT

- Personal interest
- NOT an introduction to C++, TMB or Parallel computing
- Increase awareness
- Save time and resources

Source: data-flair.training/blogs/pros-and-cons-of-r-programming-language/

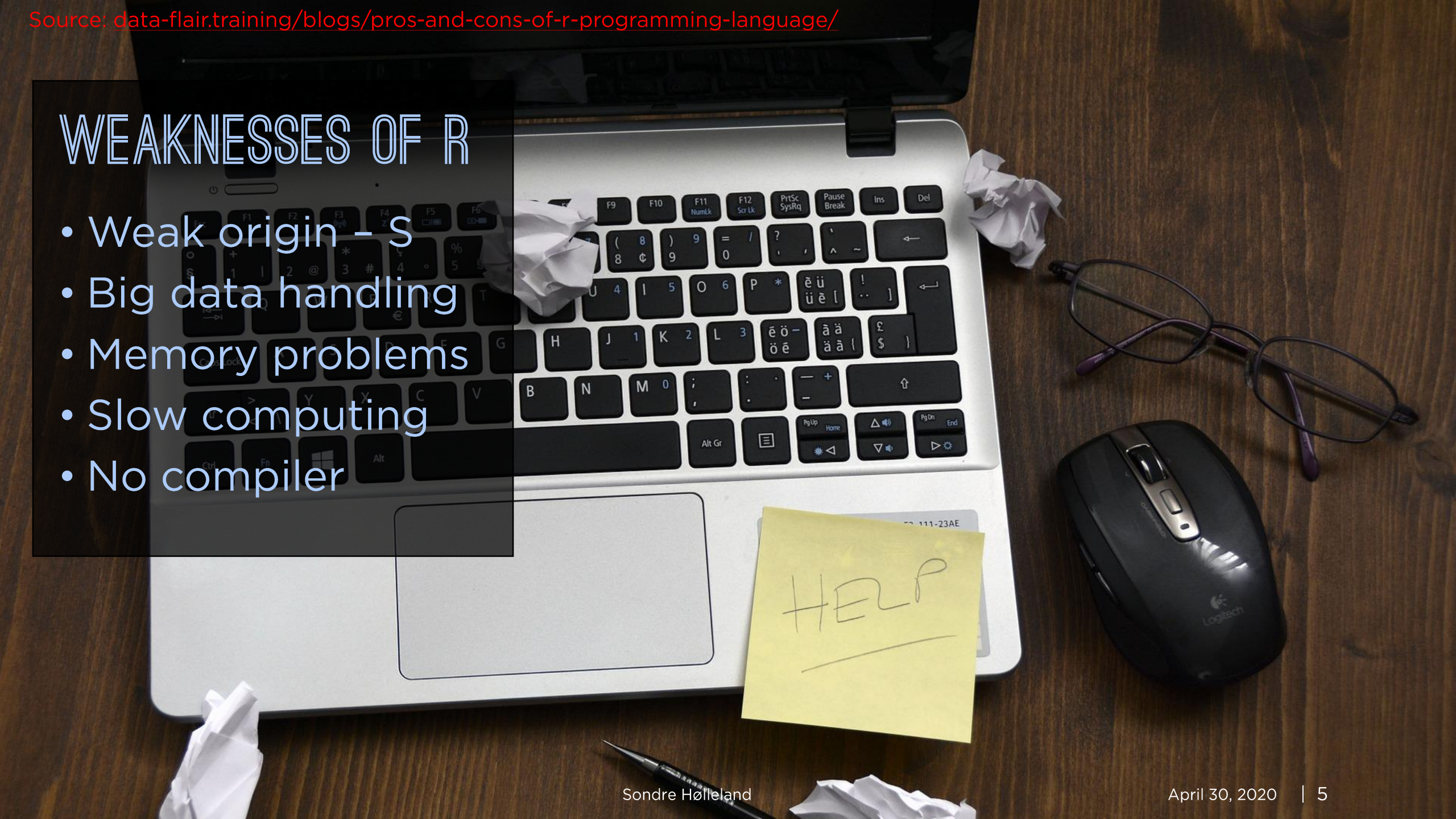
STRENGTHS OF R

- Easy to use
- Platform independent
- Line-by-line
- Nice graphics
- Compatible with other languages
- Open source with a large community
  - 15 540 packages on CRAN (April 20, 2020)
  - Continously growing
- Solving statistical problems (primarily)

Source: data-flair.training/blogs/pros-and-cons-of-r-programming-language/

WEAKNESSES OF R

- Weak origin – S
- Big data handling
- Memory problems
- Slow computing
- No compiler



COMPARING PERFORMANCE

```
t1 <- Sys.time()
.......
t2 <- Sys.time()
t2-t1
```

```
system.time({
.......
})
```

```
library(microbenchmark)
microbenchmark(......., times = 100)
```

C++ WITH R

The Rcpp package

Source: - D. Eddelbuettel and R. Francois (2011). Rcpp: Seamless R and C++ Integration. Journal of Statistical Software, 40(8), 1-18.
- H. Wickham. Advanced R.

RCPP PACKAGE

- Create a C++ function executable from R
- Requires a C++ compiler: Install Rtools/xcode/r-base-dev
- Two main syntaxes: cppFunction or sourceCpp

```
signR <- function(x) {
    if(x > 0) {
       1
    } else if(x==0) {
       0
    } else {
       -1
    }
}
```

```
cppFunction('int signC(int x) {
  if(x > 0) {
     return 1;
  } else if(x==0) {
     return 0;
  } else {
     return -1;
  }
}')
```

Source: H. Wickham. Advanced R.

PROBLEM 1: SUM A VECTOR

```
sumR <- function(x){
  n <- length(x)
  total <- 0
  for(i in 1:n)
    total <- total + x[i]
  total
}
```

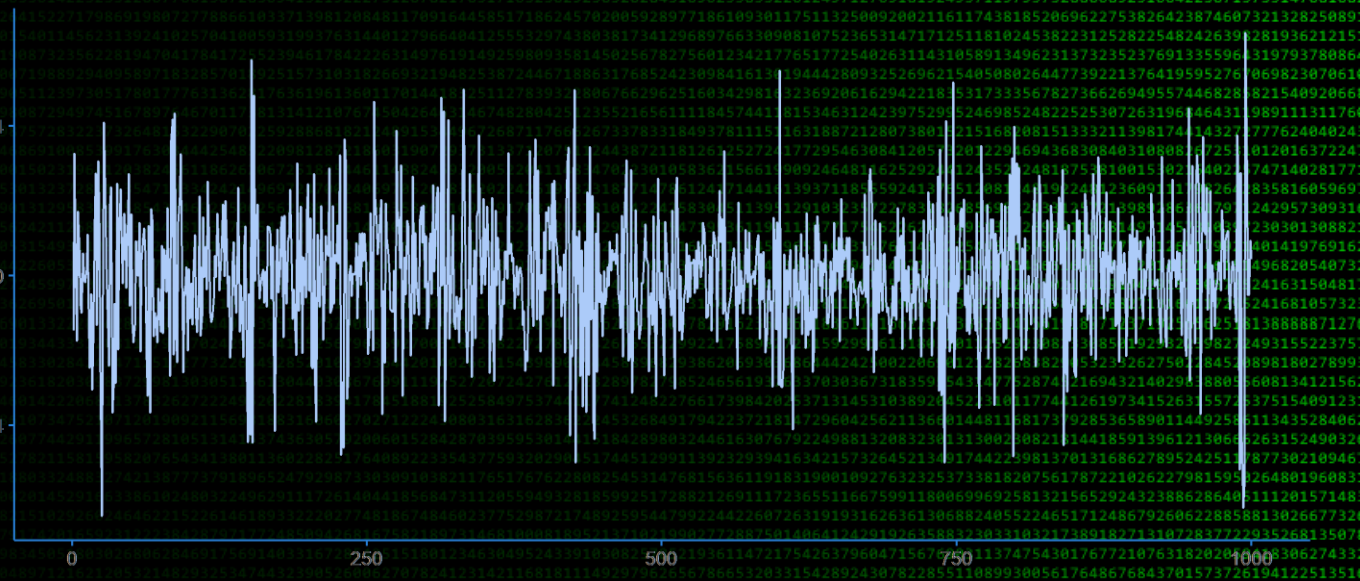
```
cppFunction('
double sumC(NumericVector x) {
  int n = x.size();
  double total = 0;
  for(int i = 0; i<n; i++){
    total = total + x[i];
  }
  return total;
}')
```

PROBLEM 2: TIME SERIES SIMULATION

$$X_t = \sigma_t Z_t,$$

$$\sigma_t^2 = \omega + \alpha X_{t-1}^2 + \beta \sigma_{t-1}^2,$$

where $Z_t \sim \mathrm{N}(0,1)$ i.i.d.



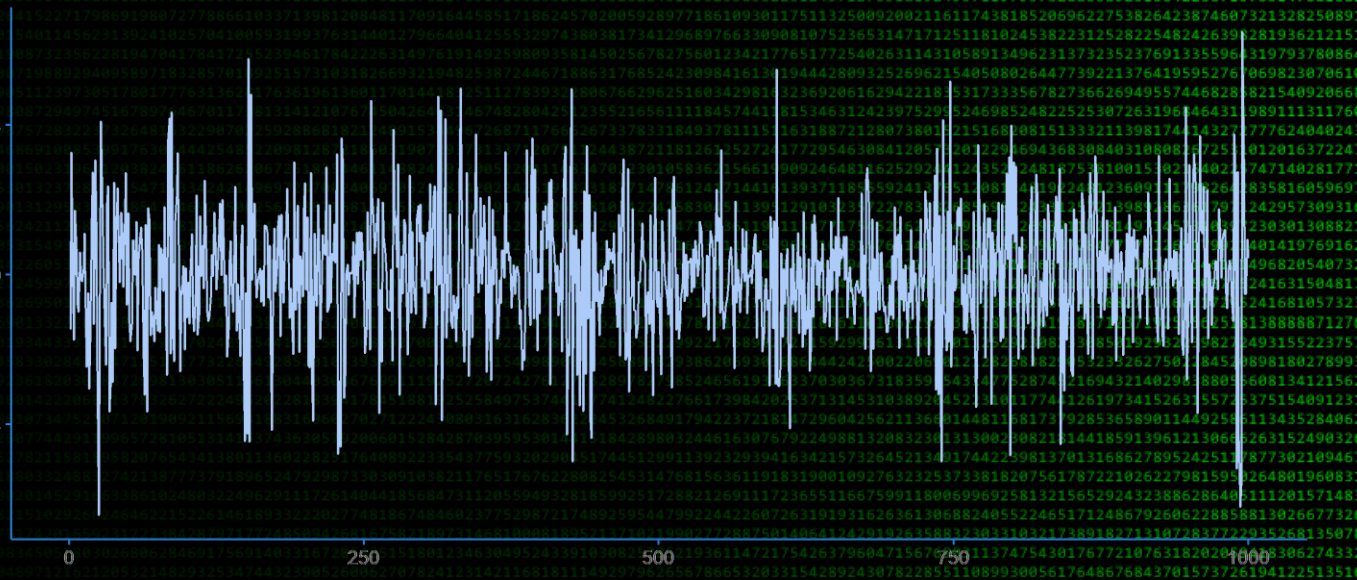
PROBLEM 2: TIME SERIES SIMULATION

$$X_t = \sigma_t Z_t,$$

$$\sigma_t^2 = \omega + \alpha X_{t-1}^2 + \beta \sigma_{t-1}^2,$$

where $Z_t \sim \mathrm{N}(0,1)$ i.i.d.

Repeat this 60 000 times
Rcpp ~2 minutes
R ~1 hour



TEMPLATE MODEL BUILDER

Efficient maximum likelihood estimation

Source: K. Kristensen, A. Nielsen, CW. Berg, H. Skaug, BM. Bell (2016). TMB: Automatic Differentiation and Laplace Approximation. Journal of Statistical Software, 70(5), 1-21.

TMB PACKAGE

- Developed by Kristensen et. al (2016), including Hans J. Skaug (UiB)
- Detailed description and installation info:
  - https://github.com/kaskr/adcomp/wiki
- Automatic Differentiation and Laplace approximation
- Workflow:
  1. Create a likelihood template (in C++).
  2. Make AD function (in R)
  3. Maximize it (in R) using e.g. nlminb

PROBLEM 3: GARCH LOG-LIKELIHOOD

$$\mathcal{L}(\theta) = -\frac{1}{2}\sum_{t=1}^{n}\left\{\log h_t(\theta) + \frac{X_t^2}{h_t(\theta)}\right\}$$

$$h_t(\theta) = \omega + \alpha X_{t-1}^2 + \beta h_{t-1}$$

$$\widehat{\theta} = \operatorname{argmax}\, \mathcal{L}(\theta)$$

For more examples: https://github.com/kaskr/adcomp

PARALLEL COMPUTING WITH R

The doParallel package and MI-LOKE

PARALLEL COMPUTING

- Multitasking
- Central Processing Unit (CPU)
  - Cores and threads
  - Hyperthreading
- Examples
  - Monte Carlo simulations
  - Bootstrap
  - Machine Learning / Deep learning
    - Graphics Processing Unit (GPU)

DOPARALLEL PACKAGE

- detectCores() – returns number of threads
- cl <- makeCluster(cores) – creates a cluster of threads
- parApply, parSapply, foreach, etc.
- stopCluster(cl) – close cluster

MI-LOKE

- Norse god
- 32 cores (64 threads)
- 376 GB RAM
- Relatively few active users

Loke (Unknown artist)
Source: norron-mytologi.info

CONNECTING TO LOKE

- Open terminal (cmd on Windows) and write:

  ```
  ssh -l [username] mi-loke.klientdrift.uib.no
  ```

- Type password: xxxxxxxx
- If not at UIB, use VPN.
  - https://it.uib.no/VPN,_Virtual_private_network

KEY COMMANDS

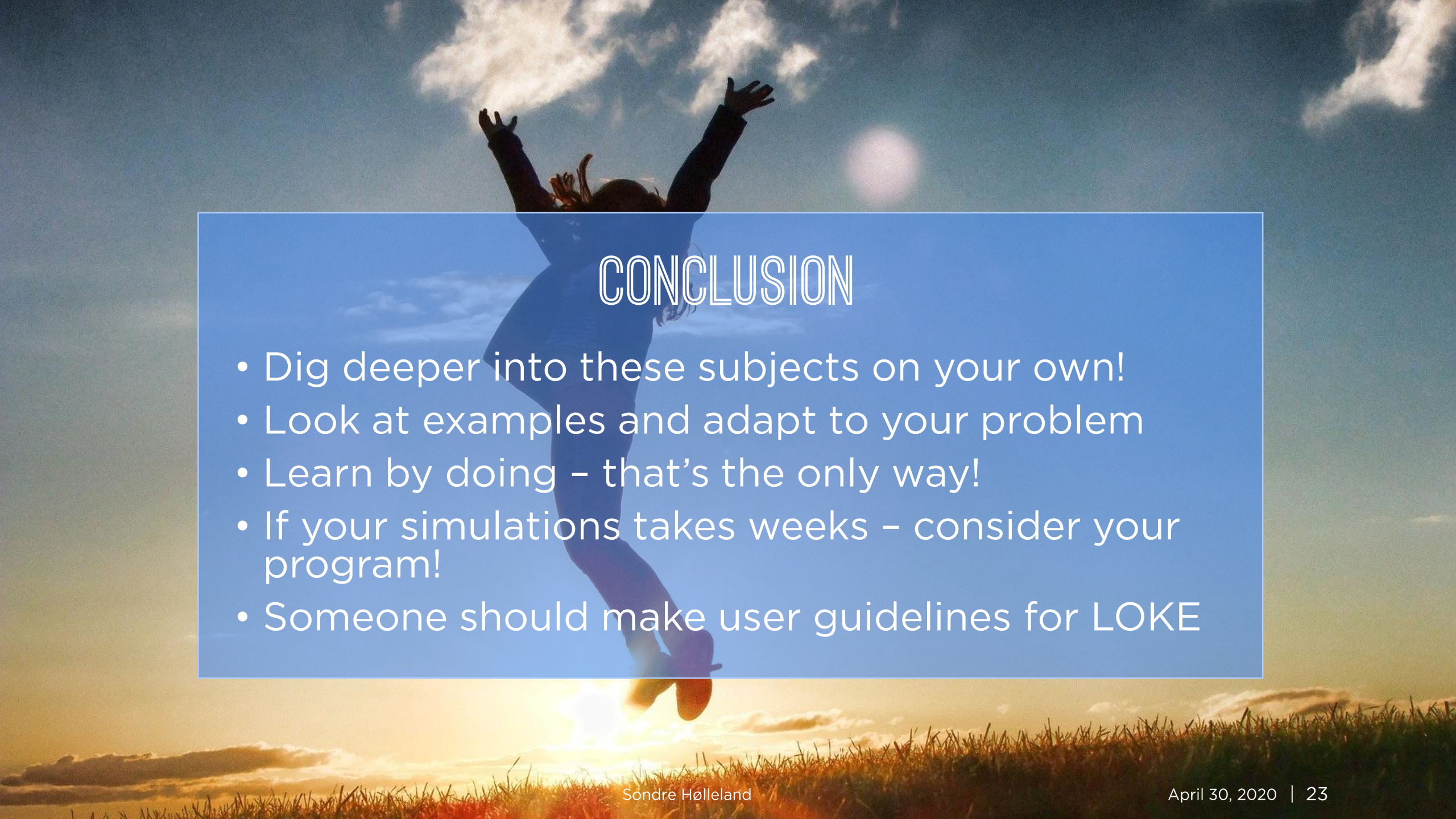
- htop or top – for viewing resources in real time
- screen – leaving code running
  - screen –ls (list screens)
  - screen –r (reattach screen)
  - CTRL+A+D – detach screen
  - CTRL+D – close screen

BE NICE!!!

- Set a priority when starting your session:
  - nice –n [nice_value] [program_name]
  - Example: nice –n 10 R
- nice_value ranges between -19 to 20, but only sudo users can use negative nice.
- Priority = 20 + nice_value
- Low nice = High priority

PROBLEM 4 MONTE CARLO SIMULATION OF GARCH MLE

- Simulate and estimate a GARCH model
- Compare parallell versus serial



CONCLUSION

- Dig deeper into these subjects on your own!
- Look at examples and adapt to your problem
- Learn by doing – that's the only way!
- If your simulations takes weeks – consider your program!
- Someone should make user guidelines for LOKE