Tracking data from GPS-enabled devices in R with package 'trackeR'

Hannah Frick, Ioannis Kosmidis

http://www.ucl.ac.uk/~ucakhfr/

Outline

- Introduction
- Package structure and capabilities
- Training distribution profiles
- Real data example
- Summary & Outlook

Introduction

Aim: Access and analyse tracking data from GPS-enabled devices.

Options include:

- Plenty of commercial software from manufacturers of devices (Catapult, Garmin, TomTom, Polar, ...) or apps (endomondo, Runtastic, Nike+, ...) for different audiences.
- Open-source software such as Golden Cheetah which provides a lot of options for analysis.

R system for statistical computing: open-source with a vast collection of add-on packages for a large variety of tasks.

- However, relatively few packages with a focus on sports and/or GPS tracking.
- Package **cycleRtools** provides specialised cycling analysis in R.
- We aim to provide the infrastructure to handle training data from running and cycling to leverage the capabilities of R.

Package trackeR

Computational infrastructure: R with add-on packages, mainly **zoo** for ordered observations as well as **ggplot2** and **ggmap** for visualisations.

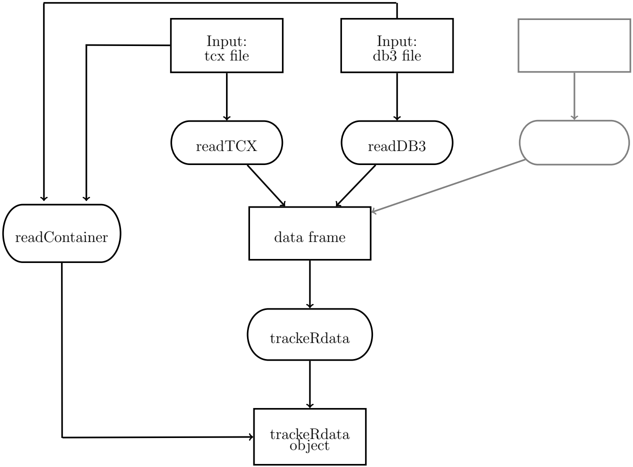
Available from Github via

```
R> # install.packages("devtools")
R> devtools::install_github("hfrick/trackeR")
```

Provides functionality for 4 main areas:

- read data
- process data
- summarise & analyse training sessions
- visualise training sessions

Read data


Input:
tcx file
Input:
db3 file
readTCX
readDB3
readContainer
data frame
trackeRdata
trackeRdata
object

Process data

All observations:

- Basic cleaning, e.g., no negative speeds or distances.
- Split observations into training sessions: Any observations further apart than a specified threshold are considered to belong to different training sessions. Default is 2 hours.

Per session:

- Distances as recorded by the GPS device can be corrected for elevation if necessary.
- Imputation of speeds: Speed 0 is imputed for time periods when the device is paused.

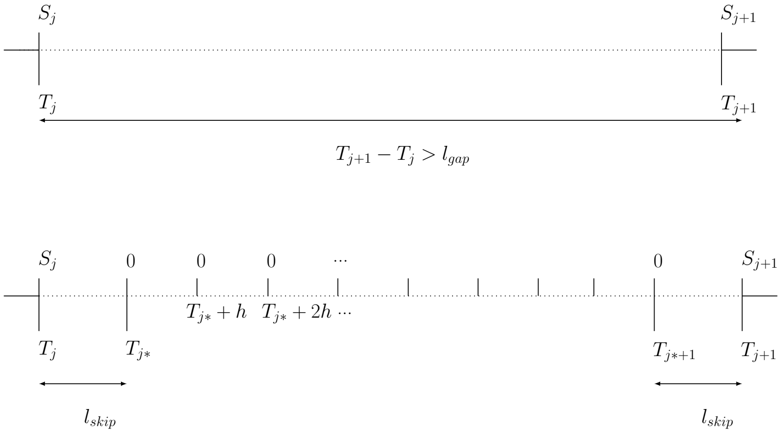
Imputation of speeds

- Either take speed measurements as provided or calculate from cumulative distances:

$$Speed_j = \frac{Distance_{j+1} - Distance_j}{Time_{j+1} - Time_j}. \quad (1)$$

- Remove observations with negative or missing speeds.
- Check time stamps of observations for gaps of more than l_{gap} seconds.
- In these gaps, impute m observations with 0 speed, latest position for latitude, longitude and altitude, and missing values for anything else.
- Add m similar observations at the start and end of the session, assuming the athlete does not immediately start or stop moving.
- Update distances based on imputed speeds via (1).

Imputation of speeds



Example for reading and processing data

Load package and read tcx file:

```
R> library("trackeR")
R> filepath <- system.file("extdata", "2013-06-04-174137.TCX",
+                            package = "trackeR")
R> df <- readTCX(file = filepath, timezone = "GMT")
```

Process data into *trackeRdata* object:

```
R> run <- trackeRdata(df,
+       ## basic information
+       units = NULL, cycling = FALSE,
+       ## separate sessions
+       sessionThreshold = 2,
+       ## elevation correction
+       correctDistances = FALSE, country = NULL, mask = TRUE,
+       ## impute speeds
+       fromDistances = TRUE, lgap = 30, lskip = 5, m = 11)
```

Example for reading and processing data

Read and process data from a single file in one step:

```
R> run <- readContainer(file = filepath, type = "tcx",
+       units = NULL, cycling = FALSE, sessionThreshold = 2)
```

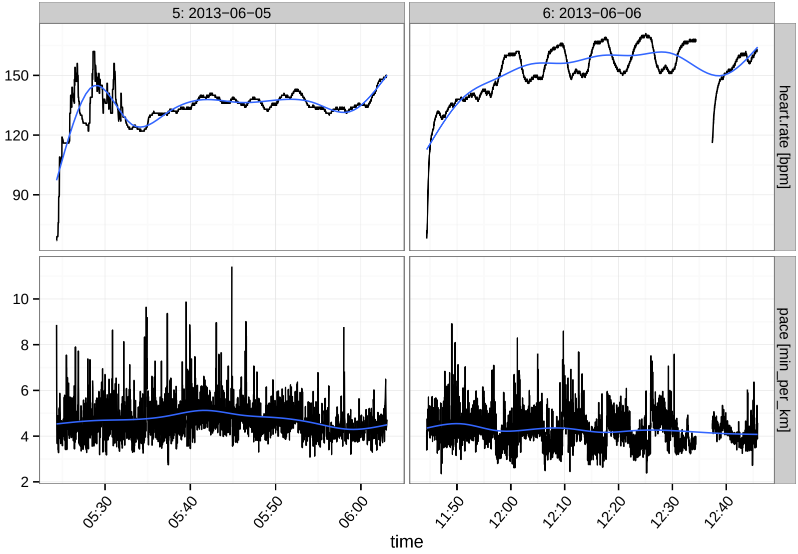
Read and process data from all files in a directory in one step:

```
R> runs <- readDirectory(directory = "~/path/to/directory/",
+       aggregate = TRUE,
+       speedunit = list(tcx = "m_per_s", db3 = "km_per_h"),
+       distanceunit = list(tcx = "m", db3 = "km"),
+       verbose = TRUE)
```

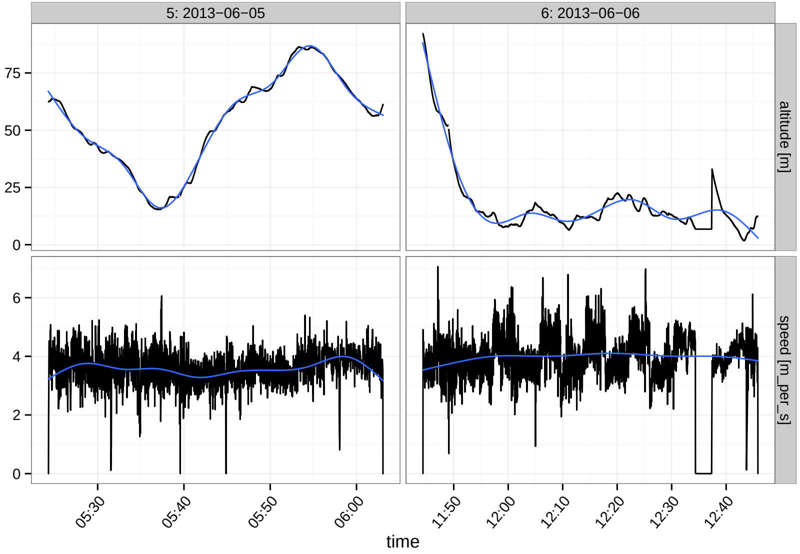
Analysis & Visualisation

- Visualise sessions:
  - Plot profiles for speed, pace, elevation level, heart rate, etc.
  - Plot route taken during a session on various maps.
- Summarise sessions.
  - Common summary statistics such as total distance, duration, time moving, averages of speed, pace, power, cadence and heart rate, etc.
  - Time spent training in specific heart rate or speed zones.
- Visualise summaries of sessions.

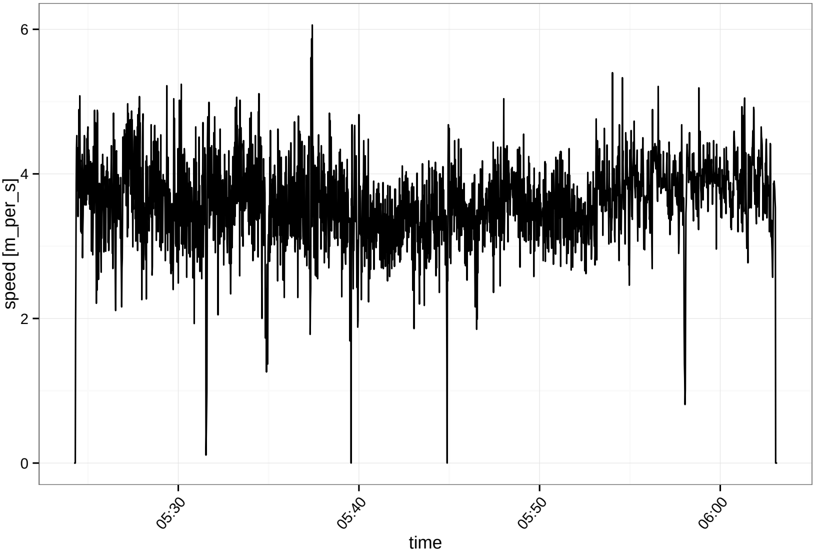
Visualise sessions



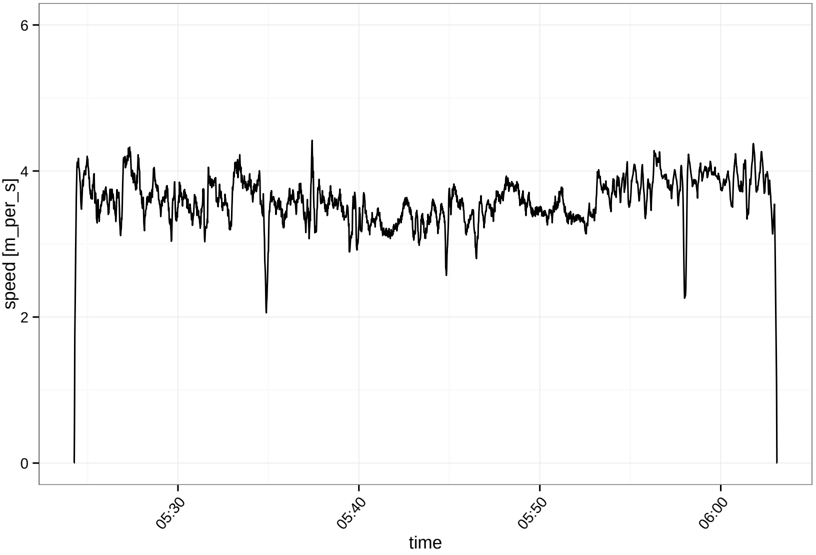
Visualise sessions



Visualise sessions



Visualise sessions



Visualise sessions

Summary of session data

```
R> summary(runs, session = 1)

*** Session 1 ***

 Session times: 2013-06-01 17:32:15 - 2013-06-01 18:37:56
 Distance: 14130.7 m
 Duration: 1.09 hours
 Moving time: 1.07 hours
 Average speed: 3.59 m_per_s
 Average speed moving: 3.66 m_per_s
 Average pace (per 1 km): 4:38 min:sec
 Average pace moving (per 1 km): 4:33 min:sec
 Average cadence: 88.66 steps_per_min
 Average cadence moving: 88.81 steps_per_min
 Average power: NA W
 Average power moving: NA W
 Average heart rate: 141.11 bpm
 Average heart rate moving: 141.12 bpm
 Average heart rate resting: 135.3 bpm
 Work to rest ratio: 48.26
```

Aggregation of high-frequency data

- Records made with high frequency, e.g., 1 or 5 Hz, generating a fairly big amount of data.
- Some summary needed to describe training sessions in a comparable way.
- Scalar summaries: time spent above x% of maximum aerobic speed, set of quantiles, etc.
- However, data are potentially noisy and appropriate degree of smoothing often is not straightforward.
- Training distribution profiles (Kosmidis & Passfield, 2015) extend the idea of “time spent above”.

Training distribution profile

The distribution profile is defined as the curve $\{v, \Pi_u(v) | v \geq 0\}$ for each session u lasting t_u seconds, with

$$\Pi_u(v) = \int_0^{t_u} I(v_u(t) > v) dt$$

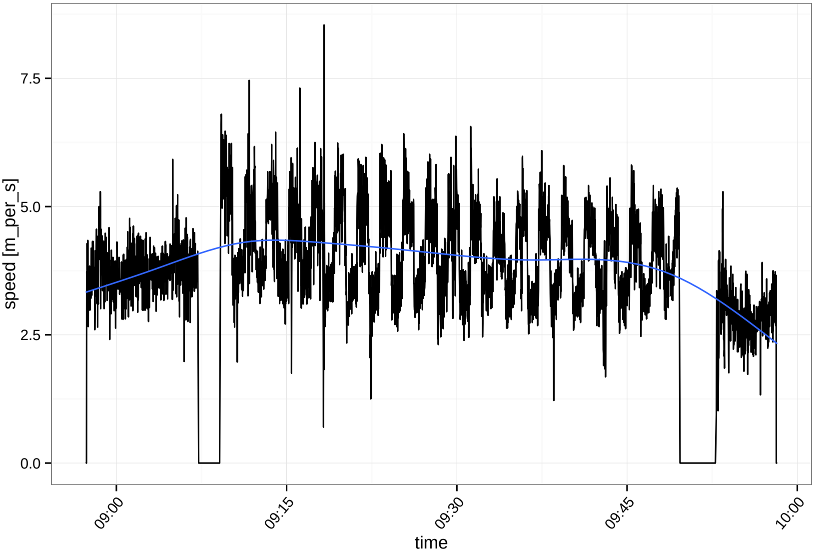
being a function of the variable v under consideration (e.g., speed or heart rate) and $I(\cdot)$ denoting the indicator function. This describes the time spent training above a certain threshold.

An observed version of $\Pi_u(v)$ can be calculated as

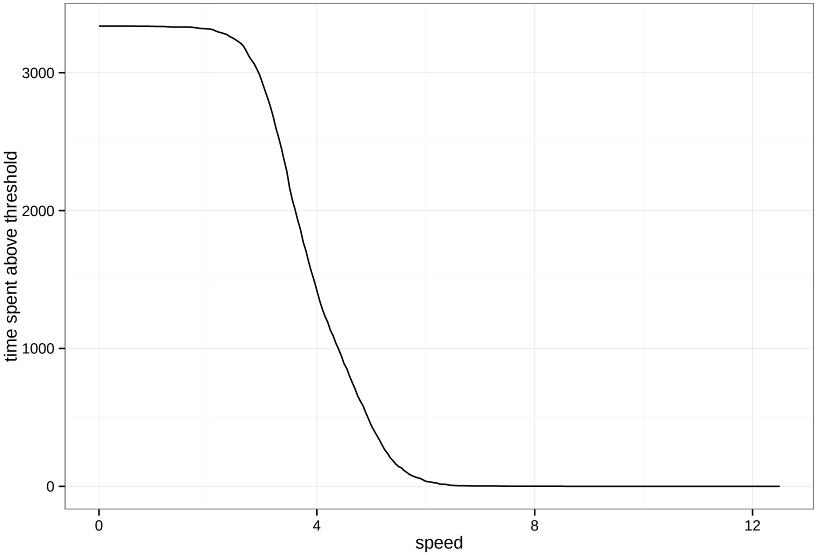
$$P_u(v) = \sum_{j=2}^{n_u} (T_{u,j} - T_{u,j-1}) I(V_{u,j} > v)$$

which can be conveniently smoothed respecting the positivity and monotonicity of $\Pi_u(v)$.

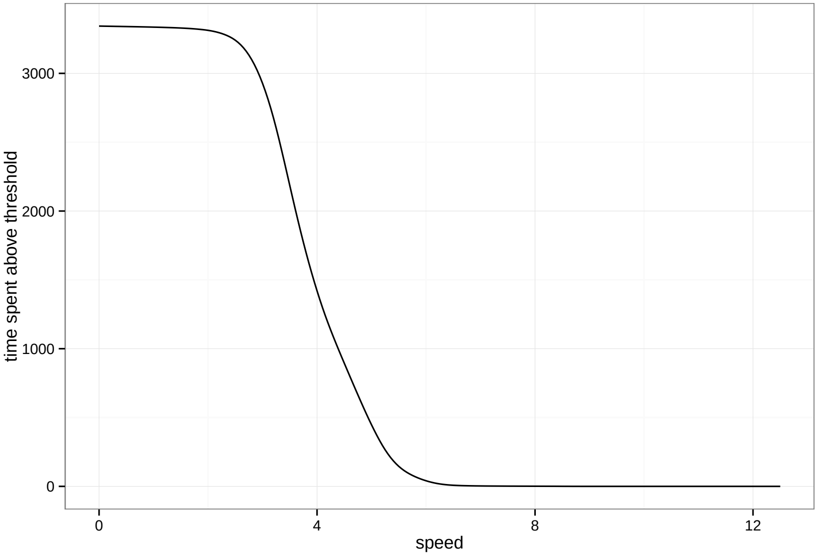
Speed profile



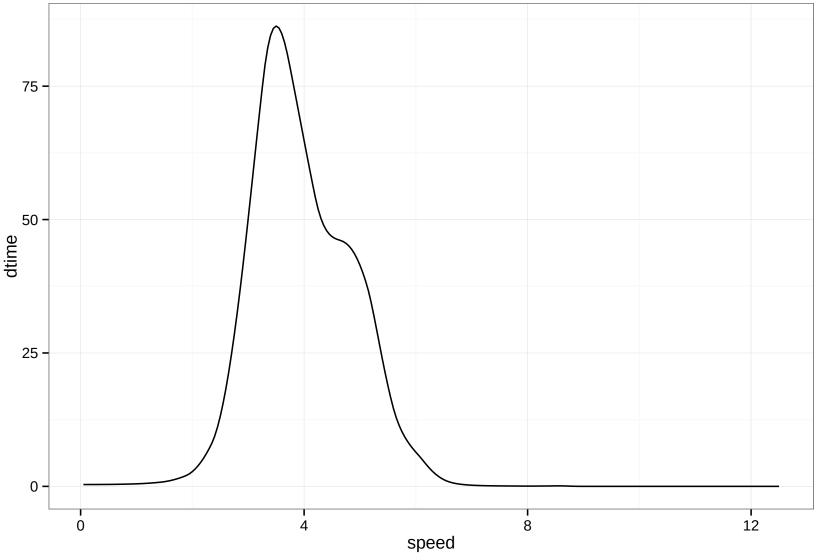
Distribution profile (unsmoothed)



Distribution profile (smoothed)



Concentration profile



Real data example

- 27 training session by one runner in June 2013.
- Data available from http://www.ucl.ac.uk/~ucakhfr/data/runs_ATI.rda.
- Brief summary of the data.
- Calculation of distribution and concentration profiles.
- Exploratory analysis of speed concentration profiles via functional principal component analysis.

Real data example

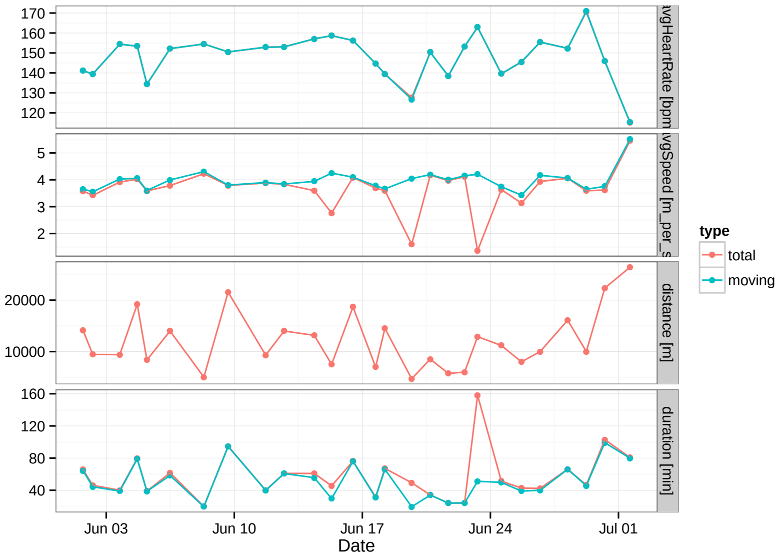
Load and summarise data:

```
R> library("trackeR")
R> load("runs.rda")
R> ## summarise sessions
R> runsSummary <- summary(runs)
R> plot(runsSummary, group = c("total", "moving"),
+       what = c("avgSpeed", "distance", "duration", "avgHeartRate"))
```

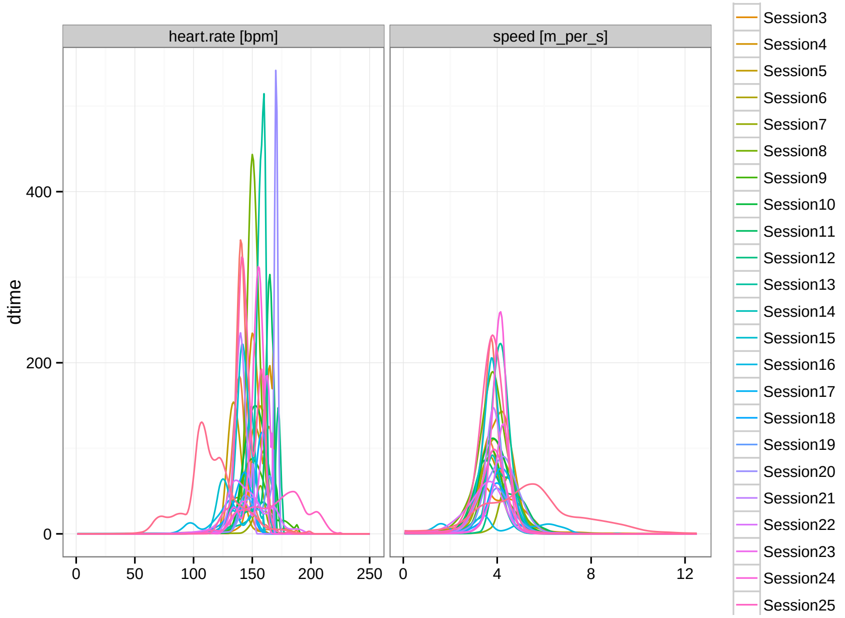
Calculate distribution and concentration profiles:

```
R> dpRuns <- distributionProfile(runs)
R> dpRunsS <- smoother(dpRuns)
R> cpRuns <- concentrationProfile(dpRunsS)
R> plot(cpRuns, multiple = TRUE, smooth = FALSE)
```

Real data example



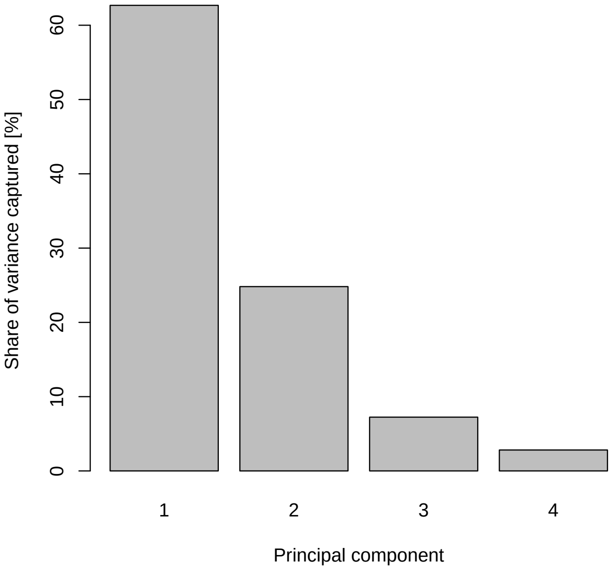
Real data example



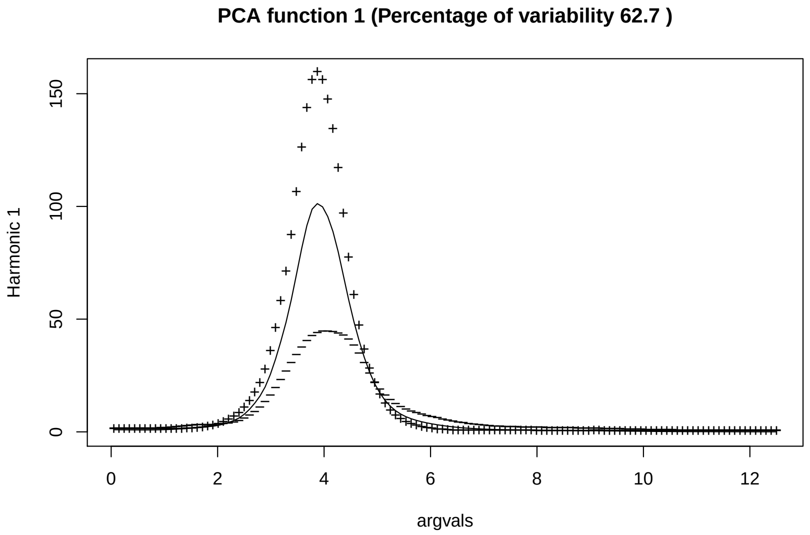
Functional principal components analysis

- Goal: explore the structure of variability in the data and describe characteristic features of the profiles.
- Tool: functional principal components analysis (PCA).
- Idea: Find a weight function such that it captures most of the variability. This is called the first principal component (PC) or first harmonic.
- The second PC is chosen to capture most of the remaining variability, and so on.
- Do the components capture interpretable concepts?

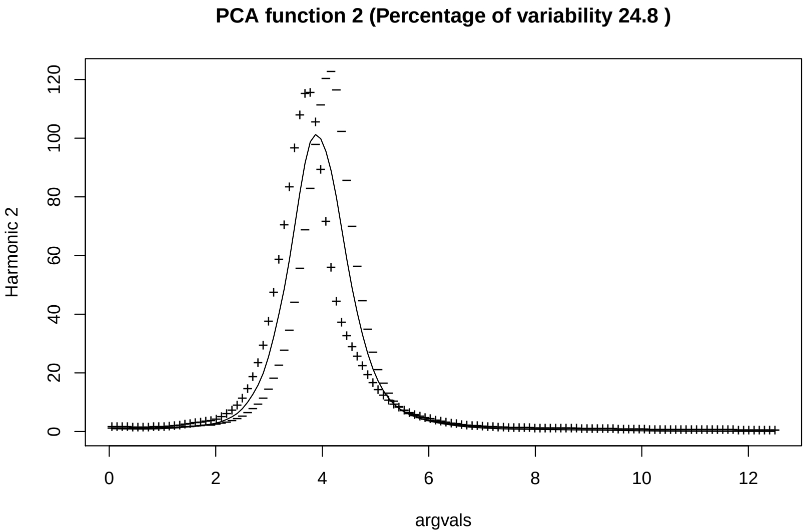
Real data example



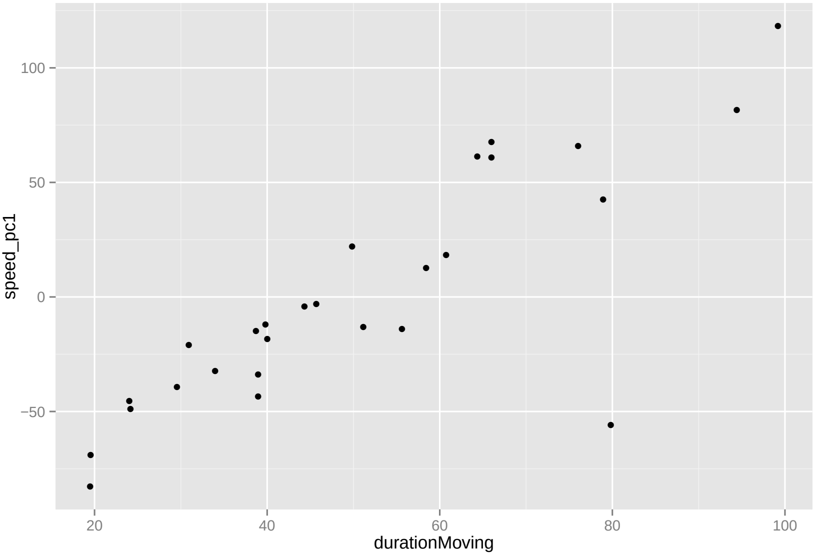
Real data example



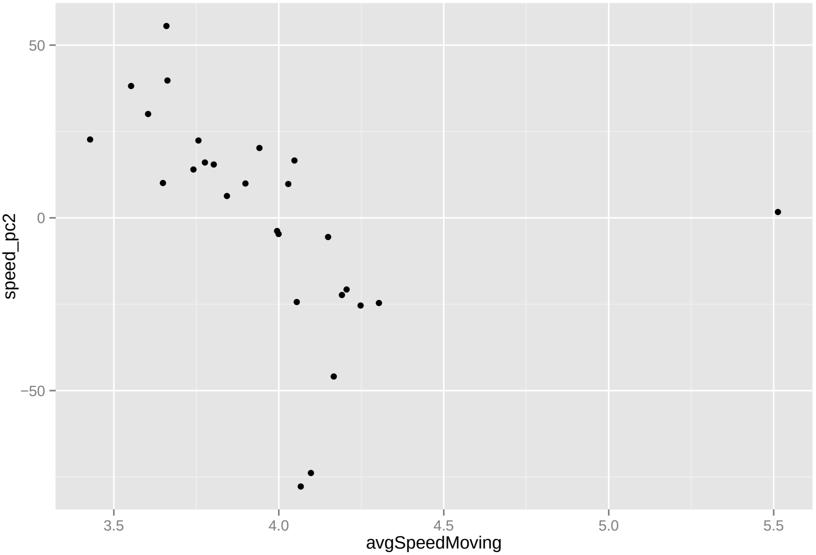
Real data example



Real data example



Real data example



Summary & Outlook

- Package **trackeR** provides basic infrastructure in R to read, process, summarise and analyse running and cycling data from GPS-enabled devices.
- Provides an implementation of training distribution profiles (Kosmidis & Passfield, 2015) as an aggregation of session data to functional objects for further analysis.
- Further work: Extend infrastructure
  - Reading capabilities for more formats, e.g., gpx and fit.
  - More analytic tools for cycling data, e.g., W', power distribution profile.
  - Suggestions welcome!
- Further work: Functional data analysis for training distribution profiles.

References

Golden Cheetah
`http://www.goldencheetah.org/`

Mackie J (2015). *cycleRtools: Tools for Cycling Data Analysis.*
R package version 1.0.4.
`https://github.com/jmackie4/cycleRtools`

Kosmidis I, Passfield L (2015). "Linking the Performance of Endurance Runners to Training and Physiological Effects via Multi-Resolution Elastic Net." ArXiv e-print arXiv:1506.01388.