

# Submission of exercise 2

Module Computational Movement Analysis: Patterns and Trends in Environmental Data

Yves Gubelmann

30.4.2021

## Task 0: Datenimport

In diesem Task werden die csv-Daten eingelesen und mit den Koordinaten E und N versehen:

```
## Import csv
wildschwein_BE <- read_delim("00_Rawdata/wildschwein_BE_2056.csv", ",", "# adjust path")
wildschwein_BE <- st_as_sf(wildschwein_BE, coords = c("E", "N"), crs = 2056, remove = FALSE)
```

## Task 1: Eine Übersicht verschaffen

In Task 1 werden die Daten nach Wildschwein gruppiert und die Zeitdifferenz zwischen den Samples berechnet. Weiter werden folgende Fragen beantwortet:

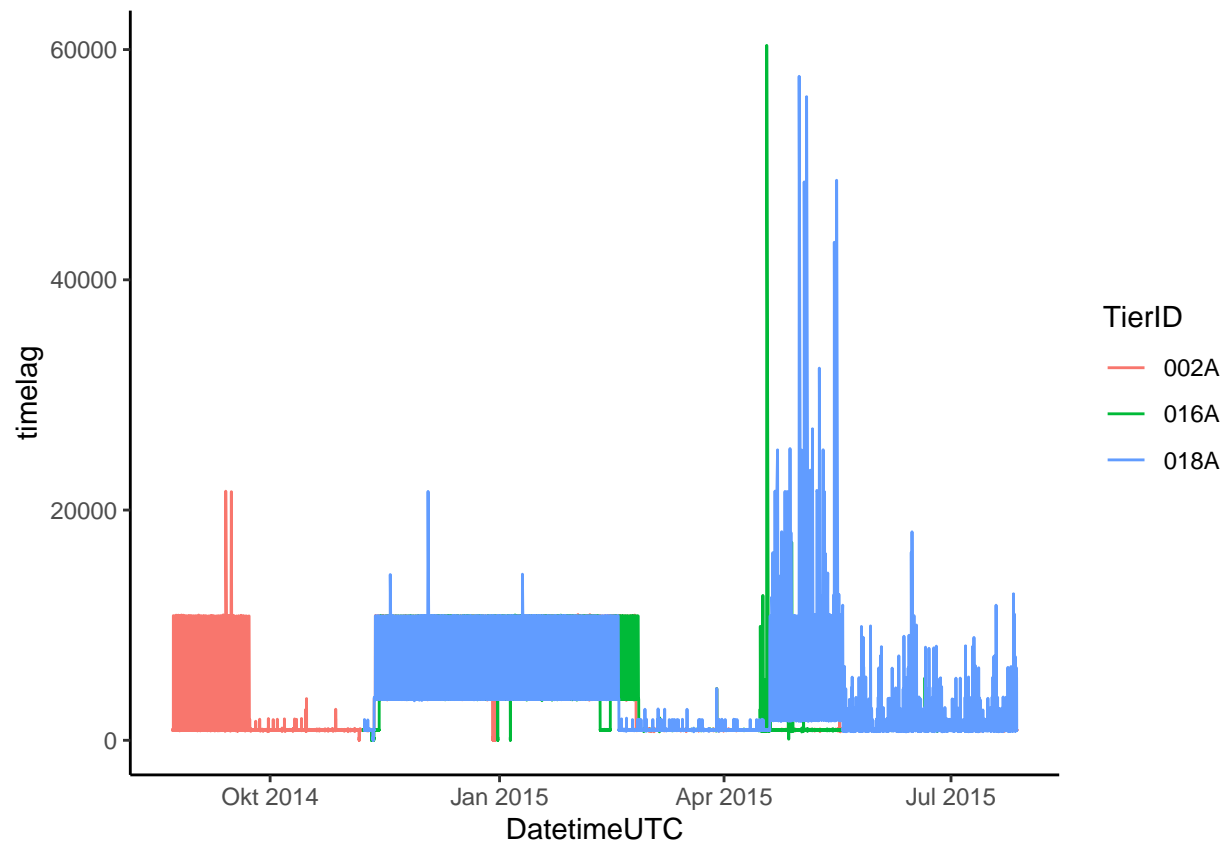
- Wieviele Individuen wurden beprobt? *drei, nämlich:*

```
n_boars <- as.factor(wildschwein_BE$TierName)
summary(n_boars)
```

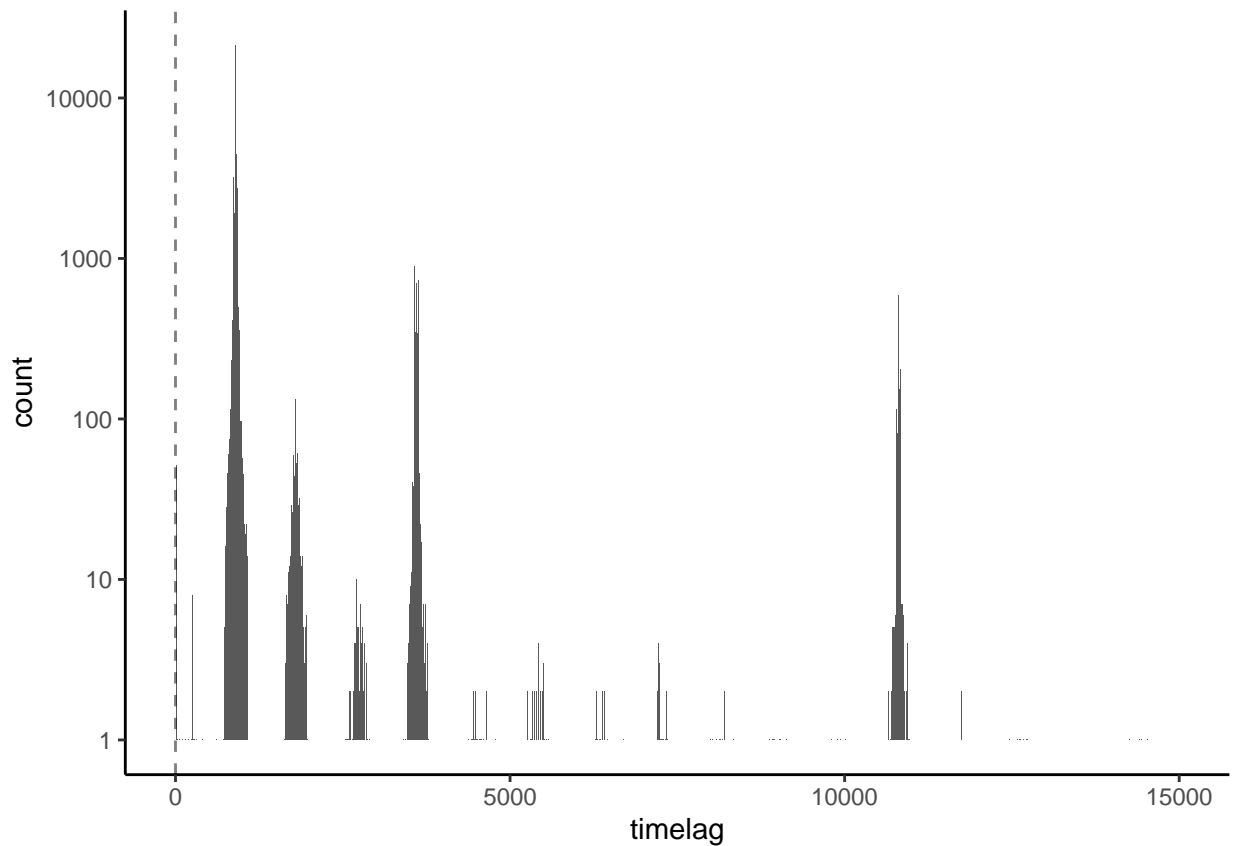
```
## Rosa Ruth Sabi
## 14364 14136 22746
```

- Für wie lange wurden die Wildschweine beprobt? *Die Wildschweine wurden vom 2014-08-22 21:00:12 bis 2015-07-27 11:00:14 beprobt. Die Daten wurden mehrheitlich gleichzeitig erhoben.*

```
ggplot(wildschwein_BE) +
  geom_line(mapping=aes(x=DateTimeUTC, y=timelag, colour=TierID)) +
  theme_classic()
```



- Was ist der zeitliche Intervall zwischen zwei Lokationen? Der durchschnittliche Intervall beträgt 1408 Sekunden. In einem Histogramm dargestellt:



## Task 2: Herleiten von Bewegungsparameter (Speed)

In Task 2 werden mit Piping grundlegende Bewegungsparameter berechnet. Mit der euklidischen Distanz werden aus zwei Lokationen die Schrittlänge *steplength* berechnet. Anschliessend wird diese durch den Zeitschritt *timelag* geteilt um die Geschwindigkeit zu berechnen.

```
# Task 2
wildschwein_BE <- wildschwein_BE %>%                                # Take a boar..
  group_by(TierID) %>%                                              # group it
  mutate(steplength = sqrt(((E-lead(E,1))^2+(N-lead(N,1))^2))) %>%
  mutate(speed = steplength/timelag)                                #calculate speed
```

## Task 3: Cross-scale Bewegungsanalyse

Mit einem neuen Dataset, welches nur ein Wildschwein enthält, werden Cross-scale Bewegungsanalysen durchgeführt. Mit den Funktionen *slice* und *seq* werden fix definierte Abschnitte herausextrahiert. Mit den extrahierten Daten werden die identischen Parameter aus Task 2 berechnet und mit den Originaldaten verglichen.

```

# Task 3
caro <- read_delim("00_Rawdata/caro60.csv",",") # adjust path
caro <- st_as_sf(caro, coords = c("E", "N"), crs = 2056, remove = FALSE)

caro_0 <- caro %>%                               # Take caro (grouping not needed)
  mutate(timelag = as.integer(difftime(lead(DatetimeUTC),DatetimeUTC, units = "secs")) %>%
  mutate(steplength = sqrt(((E-lead(E,1))^2+(N-lead(N,1))^2)) %>%
  mutate(speed = steplength/timelag)

caro_3 <- caro %>%
  slice(seq(1, nrow(caro), by = 3)) %>%
  mutate(timelag = as.integer(difftime(lead(DatetimeUTC),DatetimeUTC, units = "secs")) %>%
  mutate(steplength = sqrt(((E-lead(E,1))^2+(N-lead(N,1))^2)) %>%
  mutate(speed = steplength/timelag)

caro_6 <- caro %>%
  slice(seq(1, nrow(caro), by = 6)) %>%
  mutate(timelag = as.integer(difftime(lead(DatetimeUTC),DatetimeUTC, units = "secs")) %>%
  mutate(steplength = sqrt(((E-lead(E,1))^2+(N-lead(N,1))^2)) %>%
  mutate(speed = steplength/timelag)

caro_9 <- caro %>%
  slice(seq(1, nrow(caro), by = 9)) %>%
  mutate(timelag = as.integer(difftime(lead(DatetimeUTC),DatetimeUTC, units = "secs")) %>%
  mutate(steplength = sqrt(((E-lead(E,1))^2+(N-lead(N,1))^2)) %>%
  mutate(speed = steplength/timelag)

```

Die Datasets werden dadurch erheblich kürzer. Gut erkennbar hier:

```

# compare the sliced datasets
nrow(caro_0)
nrow(caro_3)
nrow(caro_6)
nrow(caro_9)

```

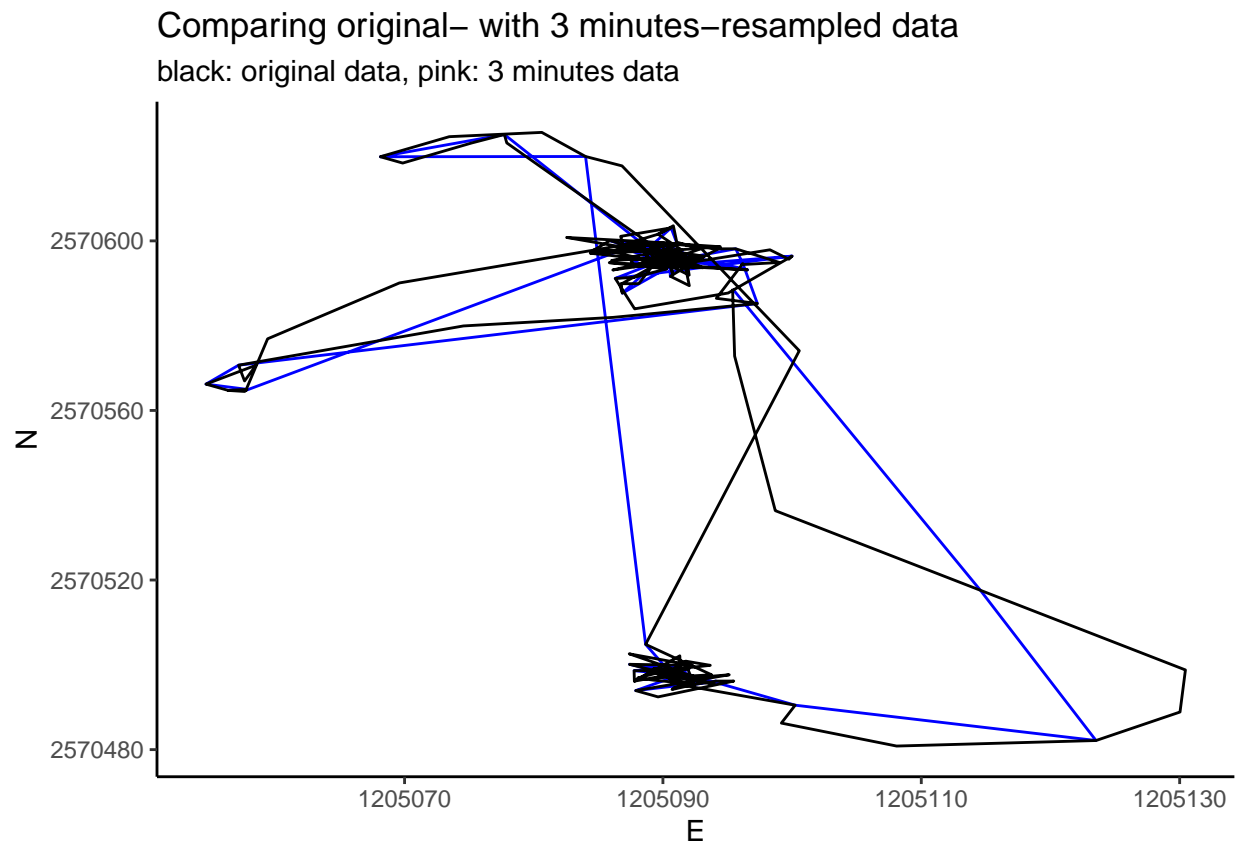
```

## [1] 200
## [1] 67
## [1] 34
## [1] 23

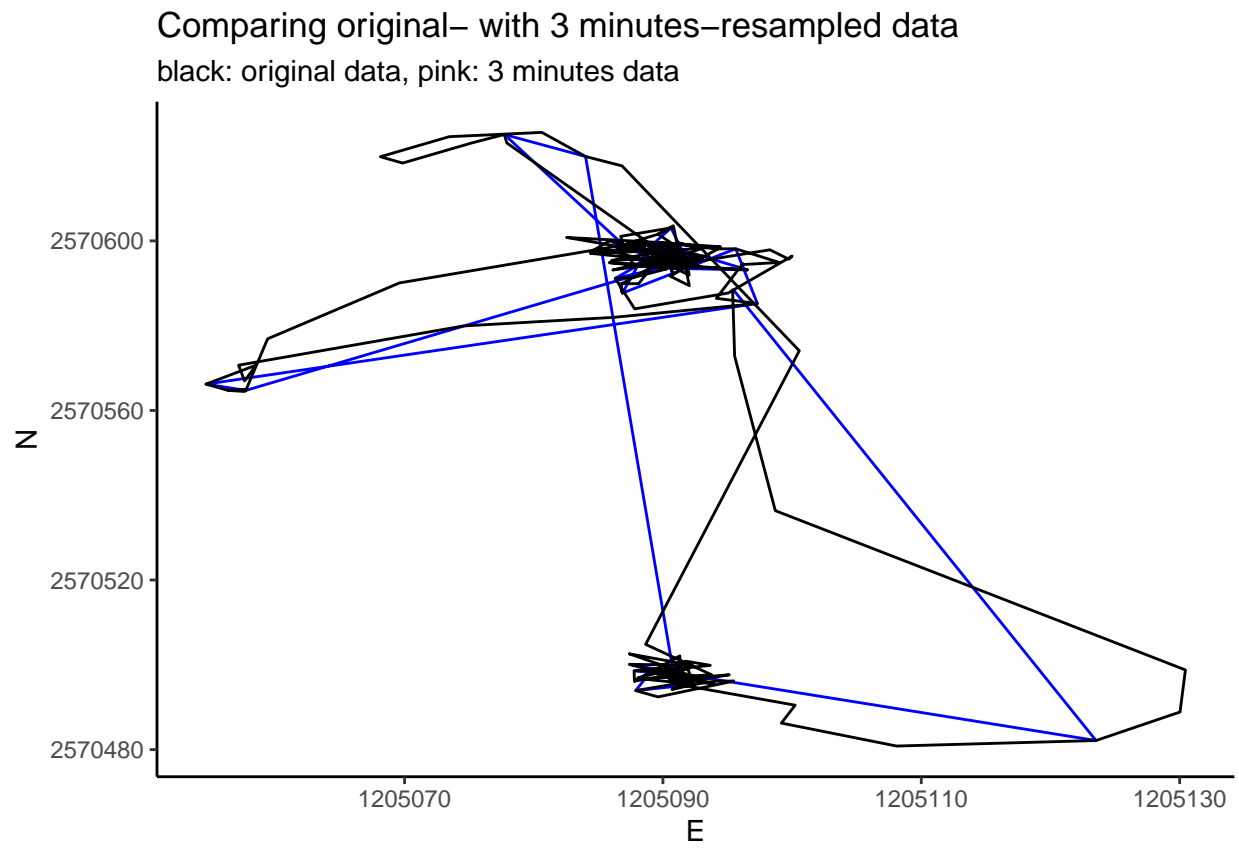
```

In einem Plot sind die Unterschiede zwischen den originalen und den extrahierten Daten dargestellt. Es wird gut ersichtlich, dass unterschiedliche slicing Intervalle zu verschiedenen Resultaten führen:

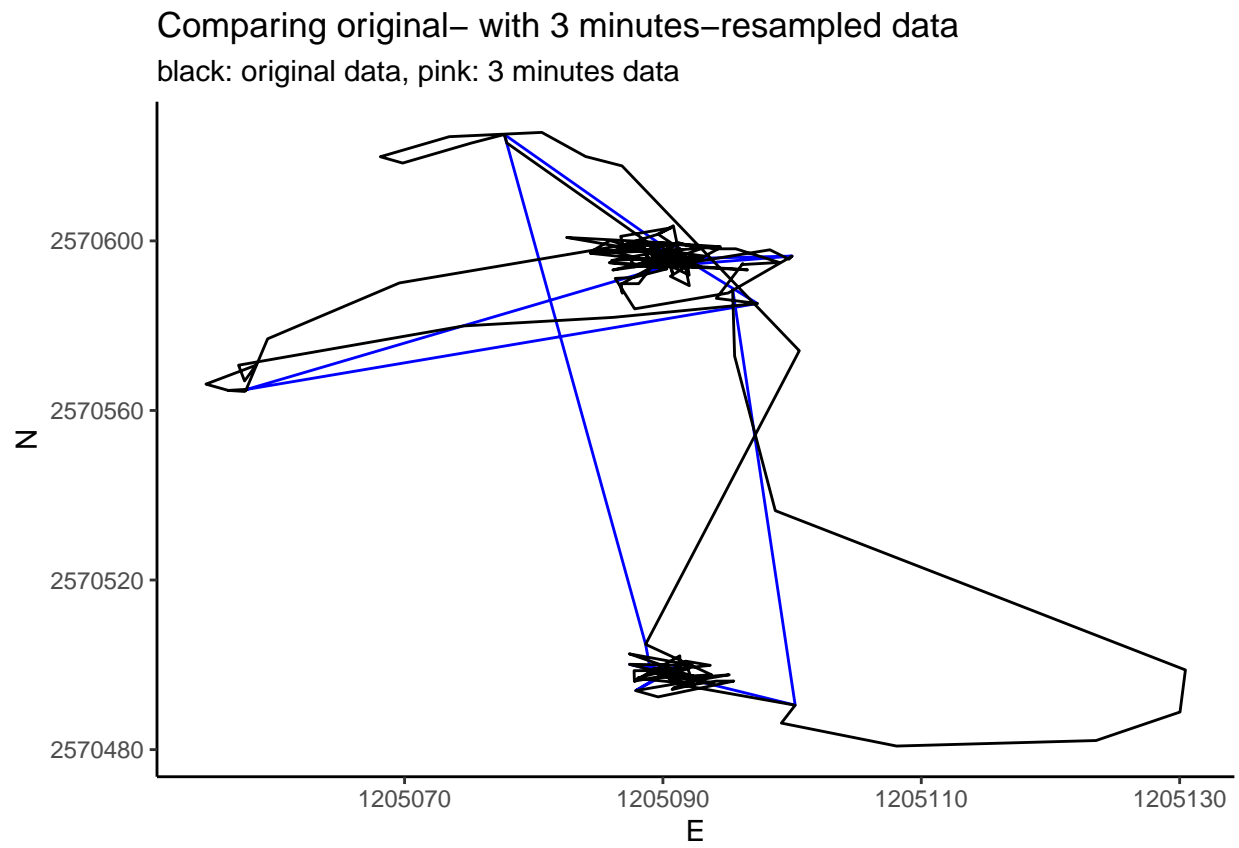
```
# compare 3 minutes data to original
ggplot() +
  geom_path(data = caro_3, aes(x=E,y=N), colour = "blue") +
  geom_path(data = caro, aes(x=E,y=N), colour = "black") +
  labs(title = "Comparing original- with 3 minutes-resampled data",
        subtitle = "black: original data, pink: 3 minutes data") +
  theme_classic()
```



```
# compare 6 minutes data to original
ggplot() +
  geom_path(data = caro_6, aes(x=E,y=N), colour = "blue") +
  geom_path(data = caro_0, aes(x=E,y=N), colour = "black") +
  labs(title = "Comparing original- with 3 minutes-resampled data",
        subtitle = "black: original data, pink: 3 minutes data") +
  theme_classic()
```



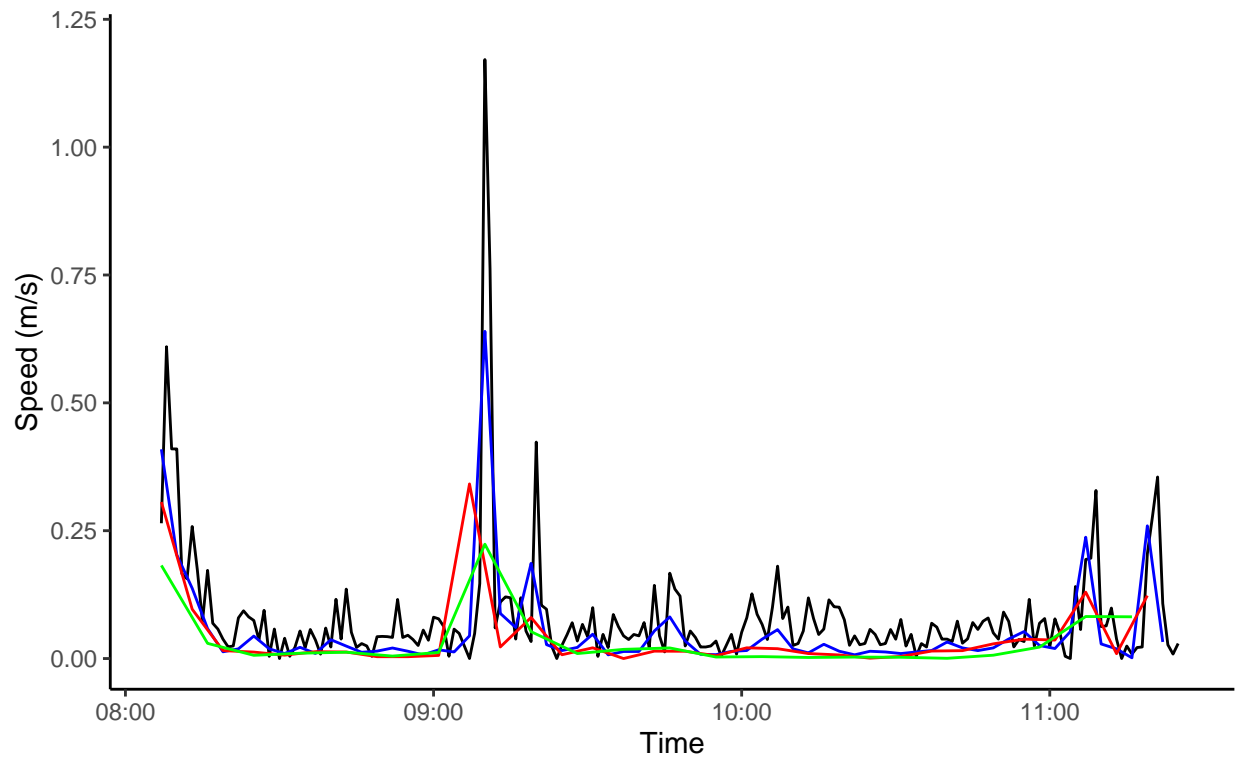
```
# compare 9 minutes data to original
ggplot() +
  geom_path(data = caro_9, aes(x=E,y=N), colour = "blue") +
  geom_path(data = caro_0, aes(x=E,y=N), colour = "black") +
  labs(title = "Comparing original- with 3 minutes-resampled data",
        subtitle = "black: original data, pink: 3 minutes data") +
  theme_classic()
```



Weiter können Unterschiede in den berechneten Geschwindigkeiten festgestellt werden:

### Comparing derived speed at different sampling intervals

black: original, blue: 3 minutes, red: 6 minutes, green: 9 minutes





#### Task 4: Herleiten von Bewegungsparameter (Rolling window functions)

##### Comparing derived speed at different sampling intervals

black: original, blue:  $k = 5$ , red:  $k = 15$ , green:  $k = 30$

