# Plot positions and speeds

Version 3: calculates speed at the sudden changes in location. This is just an EXAMPLE code acting only on Latitude. Later versions will set up Longitud ealso as well as the actual ground speed.

Reads .xlsx files Outputs .rds files for columns not all NA

```
rm(list=ls())
setwd("~/WORKSHOP/GPS/")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
Rearth <- 6371*1e3 # meters
```

## Utility GC formula

```
# Calculates the geodesic distance between two points specified by radian latitude/longitude using the
# Haversine formula (hf)
gcd.hf <- function(long1, lat1, long2, lat2) {
  R <- 6371*1000 # Earth mean radius [m]
  delta.long <- (long2 - long1)
  delta.lat <- (lat2 - lat1)
  a <- sin(delta.lat/2)^2 + cos(lat1) * cos(lat2) * sin(delta.long/2)^2
  c <- 2 * asin(min(1,sqrt(a)))
  d = R * c
  return(d) # Distance in m
}
```

## Define function to calculate speed

```
getSpeed <- function(time,lon,lat)
{
  rtod <- pi/180
  speed <- NULL
  speed_smoo <- NULL
  for (it in 1:(length(time)-1))
  {
    # calc great-circle distance between pairs of points
    distance <- gcd.hf(rtod*lon[it+1],rtod*lat[it+1],rtod*lon[it],rtod*lat[it])
    delta_time <- as.numeric(time[it+1]-time[it])/60 # dt in hours now
    #browser()
    # calc speed
    speed <- c(speed,abs(distance/delta_time))
  }
  speed_smoo <- NULL
  for (j in 1:(length(speed)-1))
  {
    speed_smoo <- rbind(speed_smoo,c(j,median(c(speed[j-2],speed[j-1],speed[j],speed[j+1],speed[j+1]),na
  }
  speed_smoo <- rbind(speed_smoo[1,],speed_smoo,speed_smoo[nrow(speed_smoo),])
  #browser()
  return(list("speed"=speed,"speed_smoo"=speed_smoo))
}
```

## Plot coloured points

```
plotcolouredpoints <- function(x,y,limitdates,pair,ivar)
{
  idx <- which(df$POSIX >= limitdates[(pair-1)*2+1] & df$POSIX < limitdates[(pair-1)*2+2])
  points(x[idx],y[idx],type="p",cex=0.3,col=1+pair)
  if (ivar == 'speed') {print(paste(pair,' from ',limitdates[(pair-1)*2+1],' to ', limitdates[(pair-1)*

}
```

## plot positions and speeds etc

```
plot_stuff <- function(df,name,limitdates)
{
  par(mfrow=c(4,1))
  nlimits <- length(limitdates)
  statname <- strsplit(strsplit(name, "/")[[1]][2],".rds")[[1]][1]
  # First plot positions
  plot(df$Longitude,df$Latitude,type="p",cex=0.3,xlab="lon",ylab="lat",main=statname)
  plotcolouredpoints(df$Longitude,df$Latitude,limitdates,1,'')
  plotcolouredpoints(df$Longitude,df$Latitude,limitdates,2,'')
  plotcolouredpoints(df$Longitude,df$Latitude,limitdates,3,'')
  plotcolouredpoints(df$Longitude,df$Latitude,limitdates,4,'')
  # Plot lon vs time
  plot(df$POSIX,df$Longitude,type="p",cex=0.3,xlab="Date/Time",ylab="lon",main=statname)
  plotcolouredpoints(df$POSIX,df$Longitude,limitdates,1,'')
  plotcolouredpoints(df$POSIX,df$Longitude,limitdates,2,'')
  plotcolouredpoints(df$POSIX,df$Longitude,limitdates,3,'')
  plotcolouredpoints(df$POSIX,df$Longitude,limitdates,4,'')
  # Plot lat vs time
  plot(df$POSIX,df$Latitude,type="p",cex=0.3,xlab="Date/Time",ylab="lat",main=statname)
  plotcolouredpoints(df$POSIX,df$Latitude,limitdates,1,'')
  plotcolouredpoints(df$POSIX,df$Latitude,limitdates,2,'')
  plotcolouredpoints(df$POSIX,df$Latitude,limitdates,3,'')
  plotcolouredpoints(df$POSIX,df$Latitude,limitdates,4,'')
  # get speed
  out <- getSpeed(df$POSIX,df$Longitude,df$Latitude)
  speed <- out$speed
  speed_smoo <- out$speed_smoo[,2]
  df$speed <- c(speed[1],speed)
  df$speed_smoo <- speed_smoo
  #browser()
  # plot speed against time
  plot(df$POSIX,df$speed,type="p",cex=0.3,xlab="Date/Time",ylab="speed [m/hr]",main=statname)
  plotcolouredpoints(df$POSIX,df$speed,limitdates,1,'speed')
  plotcolouredpoints(df$POSIX,df$speed,limitdates,2,'speed')
  plotcolouredpoints(df$POSIX,df$speed,limitdates,3,'speed')
  plotcolouredpoints(df$POSIX,df$speed,limitdates,4,'speed')
  #lines(df$POSIX,df$speed_smoo)
}
```

## model positions and calculate speeds at jumps

```
model_motion <- function(df,name,limitdates)
{

  par(mfrow=c(3,1))
  nlimits <- length(limitdates)
  statname <- strsplit(strsplit(name, "/")[[1]][2],".rds")[[1]][1]
  # Latitude
```

```
latitude_pred_at_interval_left_right <- NULL
lat_speed <- NULL
# loop over limidates and model positions before and after each limitdate
for (ilimit in seq(from=1,to=(nlimits-1),by=2))
{
  idx <- which(df$POSIX >= limitdates[ilimit] & df$POSIX < limitdates[ilimit+1] & !is.na(df$Latitude)
  rlmfit <- rlm(df$Latitude[idx] ~ df$POSIX[idx])
  lat_speed <- rbind.data.frame(lat_speed,c(ilimit,summary(rlmfit)$coefficients[2]*3600/180*pi)) # ra
  #print(c(rlmfit$fitted.values[1],last(rlmfit$fitted.values)))
  if (ilimit == 1) {
    #browser()
    plot(df$POSIX[idx],df$Latitude[idx],type="p",xlim=range(df$POSIX),ylim=range(df$Latitude,na.rm=T)
    lines(df$POSIX[idx],rlmfit$fitted.values,col=2,lwd=3)
    # evaluate diff at jump
    latitude_pred_at_interval_left_right  <- c(first(rlmfit$fitted.values), last(rlmfit$fitted.values
  }
  if (ilimit  > 1) {
    points(df$POSIX[idx],df$Latitude[idx])
    lines(df$POSIX[idx],rlmfit$fitted.values,col=2,lwd=3)
    #
    latitude_pred_at_interval_left_right  <- rbind.data.frame(latitude_pred_at_interval_left_right, c

  }

}
colnames(lat_speed) <- c("segment_number","lat_speed_radperhr")
# Longitude
longitude_pred_at_interval_left_right <- NULL
lon_speed <- NULL
# loop over limidates and model positions before and after each limitdate
for (ilimit in seq(from=1,to=(nlimits-1),by=2))
{
  idx <- which(df$POSIX >= limitdates[ilimit] & df$POSIX < limitdates[ilimit+1] & !is.na(df$Longitude
  rlmfit <- rlm(df$Longitude[idx] ~ df$POSIX[idx])
  lon_speed <- rbind.data.frame(lon_speed,c(ilimit,summary(rlmfit)$coefficients[2]*3600/180*pi)) # ra
  #print(c(rlmfit$fitted.values[1],last(rlmfit$fitted.values)))
  if (ilimit == 1) {
    #browser()
    plot(df$POSIX[idx],df$Longitude[idx],type="p",xlim=range(df$POSIX),ylim=range(df$Longitude,na.rm=
    lines(df$POSIX[idx],rlmfit$fitted.values,col=2,lwd=3)
    # evaluate diff at jump
    longitude_pred_at_interval_left_right  <- c(first(rlmfit$fitted.values), last(rlmfit$fitted.value
  }
  if (ilimit  > 1) {
    points(df$POSIX[idx],df$Longitude[idx])
    lines(df$POSIX[idx],rlmfit$fitted.values,col=2,lwd=3)
    #
    longitude_pred_at_interval_left_right  <- rbind.data.frame(longitude_pred_at_interval_left_right,

  }

}
colnames(lon_speed) <- c("segment_number","lon_speed_radperhr")
```

```
  segment_speed <- Rearth*sqrt((lon_speed[,2]*cos(median(df$Latitude,na.rm=T)/180*pi))^2+(lat_speed[,2]
  #browser()
  return(list("lats"=latitude_pred_at_interval_left_right,"longs"=longitude_pred_at_interval_left_right
}
```

## fucntion to get interval and jump speeds

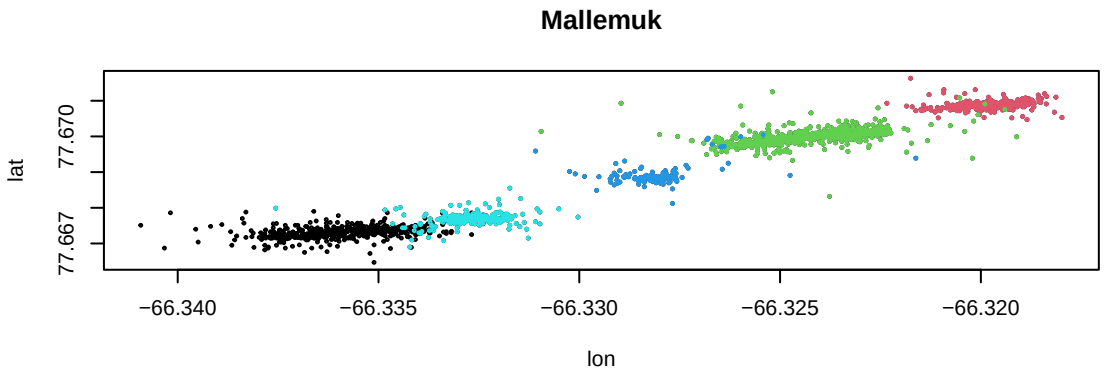
```
get_jump_speeds <- function(listerne,lon_in,lat_in)
{
  lon <- lon_in/180*pi # in radians
  lat <- lat_in/180*pi
  delta_t <- 3 # hours

  # calculate jump speeds
  n_segments <- nrow(listerne$lats)
  speed <- NULL
  for (iseg in 1:(n_segments-1))
  {
    delta_longitude <- listerne$longs[iseg+1,1]-listerne$longs[iseg,2] # in degrees
    delta_longitude <- delta_longitude/180*pi # in radians
    delta_latitude  <- listerne$lats[iseg+1,1]-listerne$lats[iseg,2] # in degrees
    delta_latitude <- delta_latitude/180*pi # in radians
    speed <- rbind.data.frame(speed,c(iseg,Rearth/delta_t*sqrt(delta_longitude^2*cos(lat)^2+delta_latitu
  }
  colnames(speed) <- c("jump_number","speed_metersph")
  return(list("speed_jump"=speed))
}
```

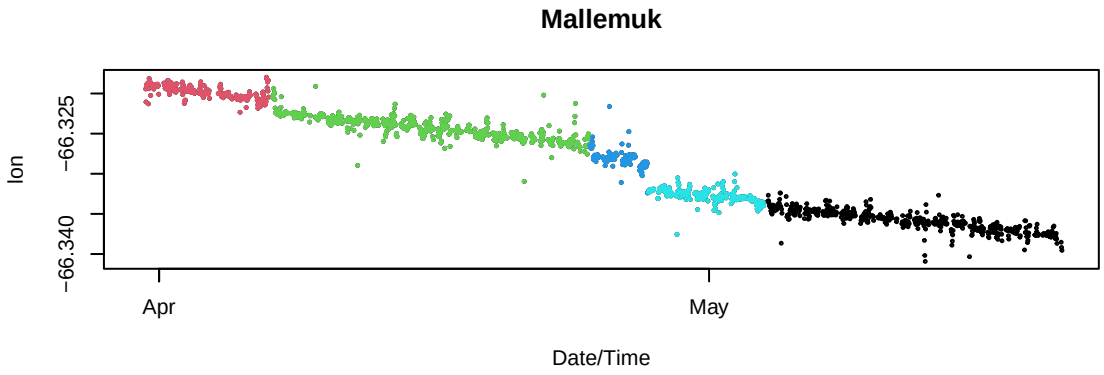
## read and plot each file

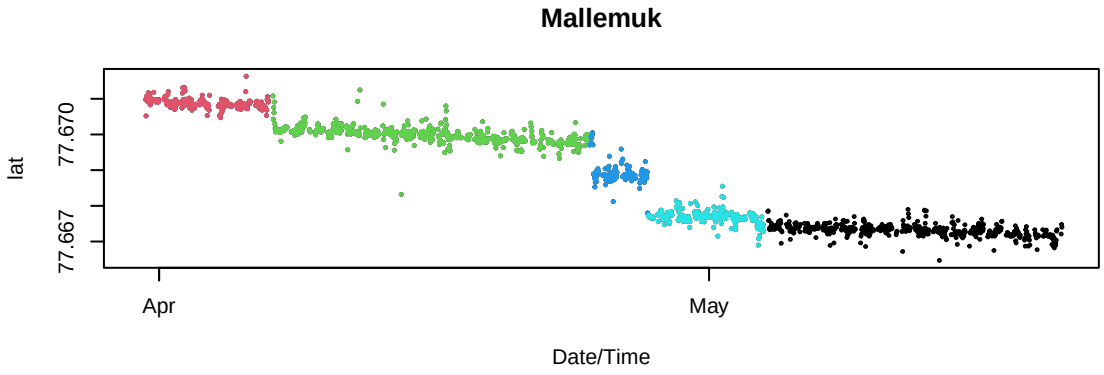
```
files <- c('OUTPUT/Mallemuk.rds','OUTPUT/Soekonge.rds','OUTPUT/Ismaage.rds','OUTPUT/Havterne.rds','OUTPU
#files <- files[1]
# list the important times - start, jumps, ending:
important_times <- c(as.POSIXct("2022-03-31 00:00:00",tz="UTC"),as.POSIXct("2022-04-7 00:00:00",tz="UTC
                     as.POSIXct("2022-04-24 12:00:00",tz="UTC"),as.POSIXct("2022-04-27 17:00:00",tz="UTC
                     as.POSIXct("2022-05-04 00:00:00",tz="UTC"),as.POSIXct("2022-05-31 00:00:00",tz="UTC
limitdates <- NULL
for (it in 1:(length(important_times)-1))
{
  limitdates <- c(limitdates,c(important_times[it],important_times[it+1]))
}

#limitdates2 <- c(as.POSIXct("2022-03-31 00:00:00",tz="UTC"),as.POSIXct("2022-04-7 00:00:00",tz="UTC"),
#                 as.POSIXct("2022-04-7 00:00:00",tz="UTC"),as.POSIXct("2022-04-24 12:00:00",tz="UTC"),
#                 as.POSIXct("2022-04-24 12:00:00",tz="UTC"),as.POSIXct("2022-04-27 17:00:00",tz="UTC"),
#                 as.POSIXct("2022-04-27 17:00:00",tz="UTC"),as.POSIXct("2022-05-04 00:00:00",tz="UTC"),
#                 as.POSIXct("2022-05-04 17:00:00",tz="UTC"),as.POSIXct("2022-05-31 00:00:00",tz="UTC"))
#browser()
```

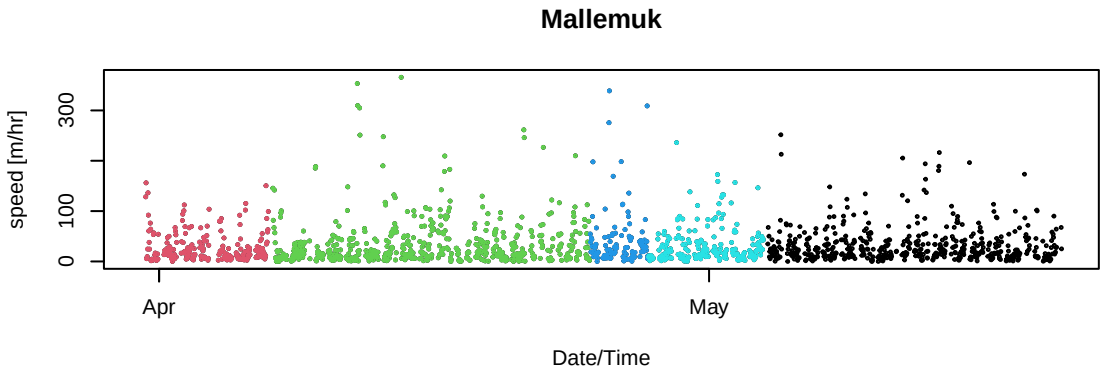
```
for (ifil in 1:length(files[1]))
{
  print("-----------------------------------------")
  print(paste(" Processing file ",files[ifil]))
  df <- readRDS(files[ifil])
  plot_stuff(df,files[ifil],limitdates)
  # more better way
  listerne <- model_motion(df,files[ifil],limitdates)
  segspeeds <- round(listerne$segspeed,2)
  paste(" Segment speeds: ",segspeeds," meters/hour")
  # get speeds of various kinds from the list of jumps
  speeds <- get_jump_speeds(listerne,lon=median(df$Longitude,na.rm=T),lat=median(df$Latitude,na.rm=T))
  print(speeds)

}
```

```
## [1] "-----------------------------------------"
## [1] " Processing file  OUTPUT/Mallemuk.rds"
```


Mallemuk
lat
77.670
77.667
−66.340
−66.335
−66.330
−66.325
−66.320
lon


Mallemuk
lon
−66.325
−66.340
Apr
May
Date/Time


Mallemuk
lat
77.670
77.667
Apr
May
Date/Time

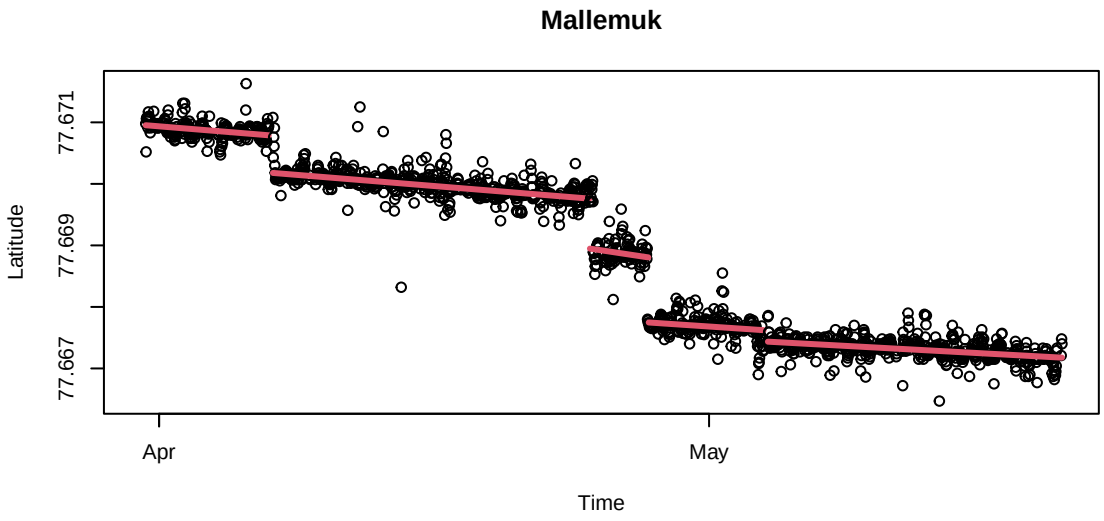

Mallemuk
speed [m/hr]
300
100
0
Apr
May
Date/Time

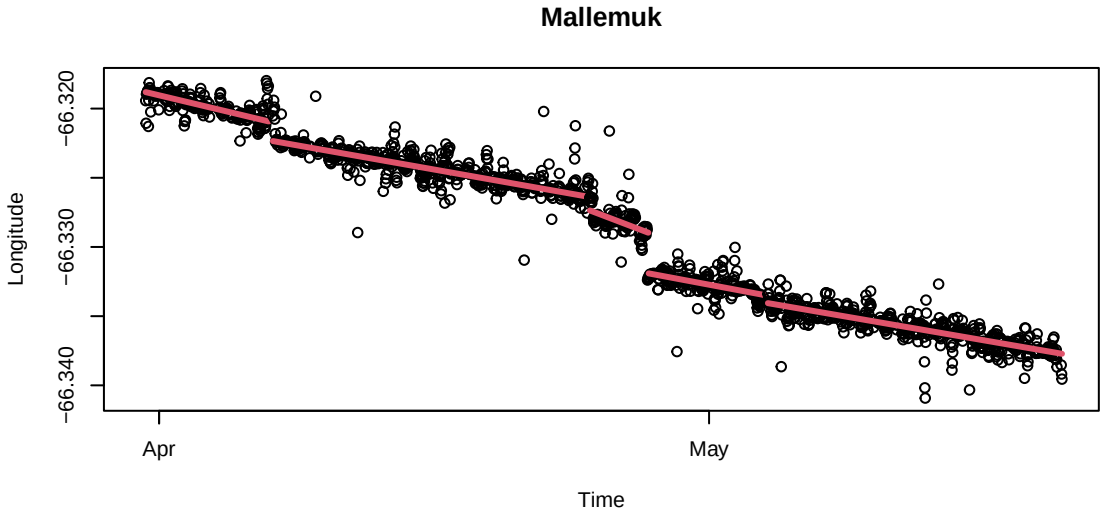
```
## [1] "1  from  1648684800  to  1649289600  speed:  16  +/-  1.8 m/hr"
## [1] "2  from  1649289600  to  1650801600  speed:  19.2  +/-  1.8 m/hr"
## [1] "3  from  1650801600  to  1651078800  speed:  27  +/-  4.9 m/hr"
## [1] "4  from  1651078800  to  1651622400  speed:  17.8  +/-  2.2 m/hr"
```

```
## $speed_jump
##   jump_number speed_metersph
## 1           1       74.51467
## 2           2       73.04962
## 3           3       68.91788
## 4           4       50.62154
```

```
print("-----------------------------------------")
```

```
## [1] "-----------------------------------------"
```


Mallemuk
Latitude
77.671
77.669
77.667
Apr
May
Time


Mallemuk
Longitude
−66.320
−66.330
−66.340
Apr
May
Time