

Ruth's GPS

Version 4: used the longer data-set starting March 28

```
rm(list=ls())
library(MASS)
library(ie2misc)
library(dplyr)
library(lubridate)
setwd("~/WORKSHOP/GPS/")

df <- readRDS("OUTPUT/EVERYTHING3.rds")
# clean out columns with only NA
listen <- NULL
for (icol in 1:ncol(df))
{
  listen <- rbind(listen,c(icol,colSums(is.na(df[,icol]))))
}
idx <- which(listen[,2] == nrow(df))
df <- df[,-idx]
df$POSIX <- as.POSIXct(df$"Timestamp UTC",tz="UTC")
df_org <- df
#
unique_names <- unique(df$UnitName)
unique_names

## [1] "Fjeldrype 860640050232018"      "Havterne 300434066435700"
## [3] "Soekonge 300434066433690"        "Ismaage 300434066437720"
## [5] "Mallemuk 300434066431710"        "Edder 300434066433700"
## [7] "Havoern 300434066437680"         "Landsvale 860640050251737"
## [9] "Ravn 860640050244401"             "Strandskade 860640050251356"
## [11] "Stenpikkere 860640050244062"
```

Define that function for later use

```
kleaner <- function(df_in)
{
  # clean out columns with only NA
  listen <- NULL
  for (icol in 1:ncol(df_in)) {
    listen <- rbind(listen,c(icol,colSums(is.na(df_in[,icol]))))

    idx <- which(listen[,2] == nrow(df_in))
    if (length(idx) != 0) {df_in <- df_in[,-idx]}
    return(df_in)
}
```

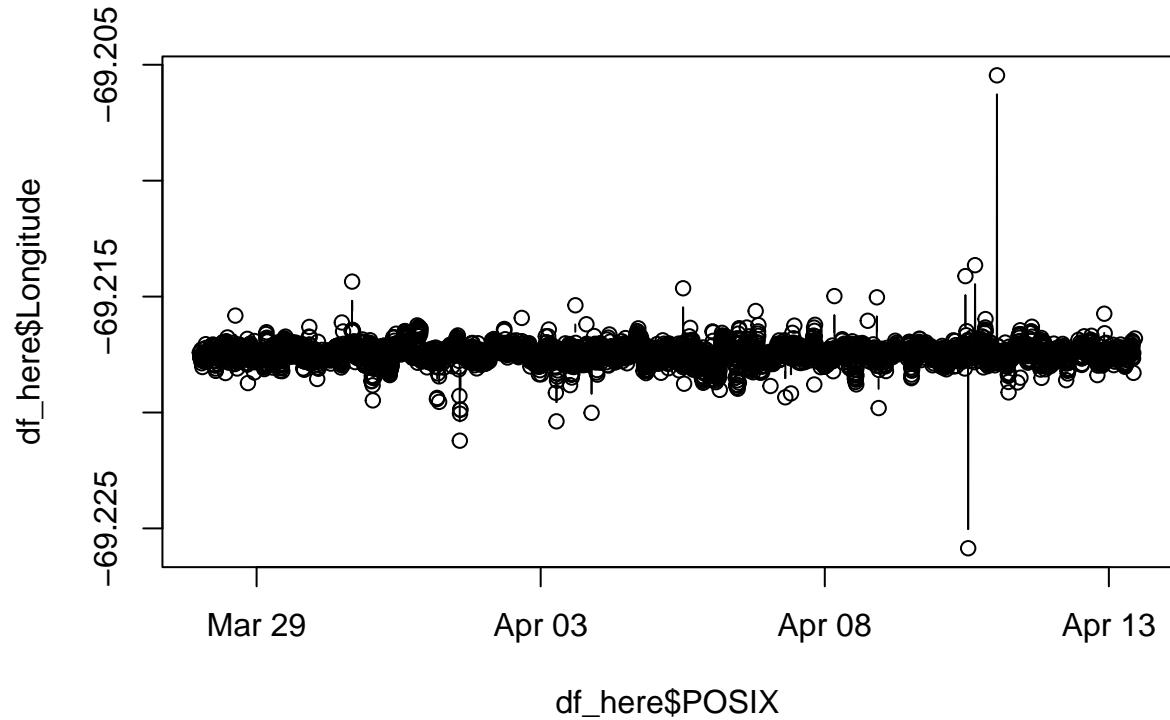
need to clean up Fjeldrype as it is the reference

```
par(mfrow=c(2,1))
df <- df_org
idx <- which(df$UnitName == "Fjeldrype 860640050232018")
# remove the old
df_org <- df_org[-idx,]
#
df_here <- df[idx,]
df_here <- na.omit(df_here)
plot(df_here$POSIX,df_here$Longitude,type="b",main="Fjeldrype before outlier removed")
rlmfit <- rlm(df_here$Longitude ~ df_here$POSIX)
z <- unname(abs(rlmfit$residuals/madstat(rlmfit$residuals))) # counting goes off due to omitted values
jdx <- which(z > 7)
df_here$Longitude[jdx] <- NA
print(paste("I removed in Fjeldrype ",length(jdx)," outliers in longitude"))

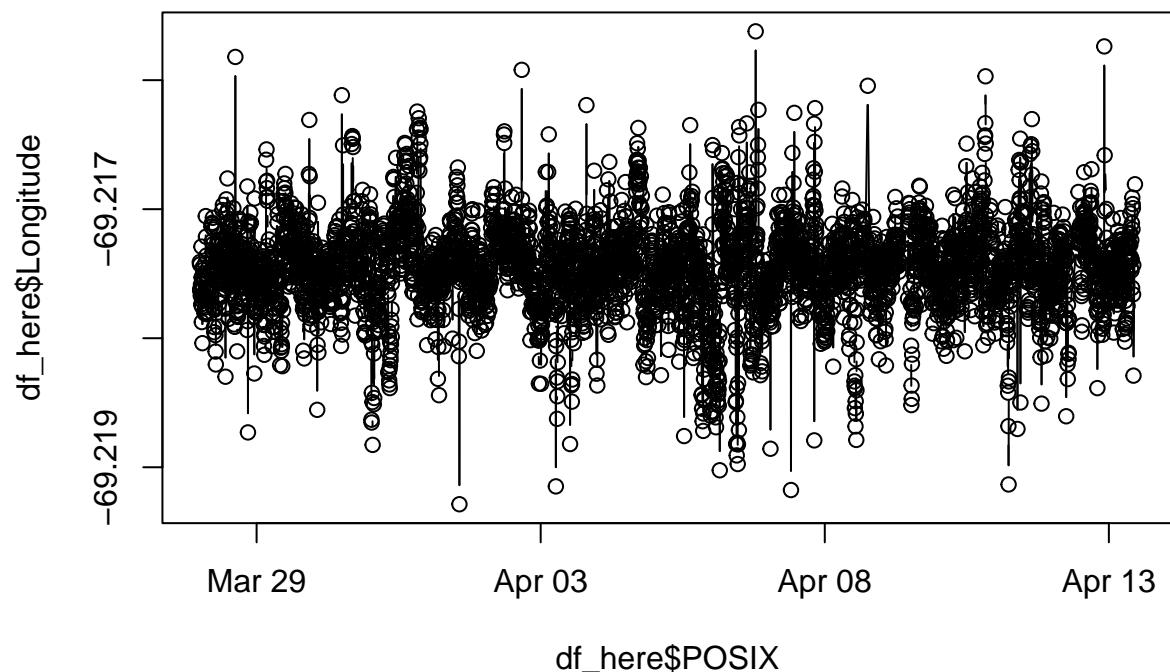
## [1] "I removed in Fjeldrype 21 outliers in longitude"

plot(df_here$POSIX,df_here$Longitude,type="b",main="Fjeldrype after outlier removed")
```

Fjeldrype before outlier removed



Fjeldrype after outlier removed



```
# rbind the new
df_org <- rbind.data.frame(df_org, df_here)
df <- df_org
```

Also clean up Ismaage

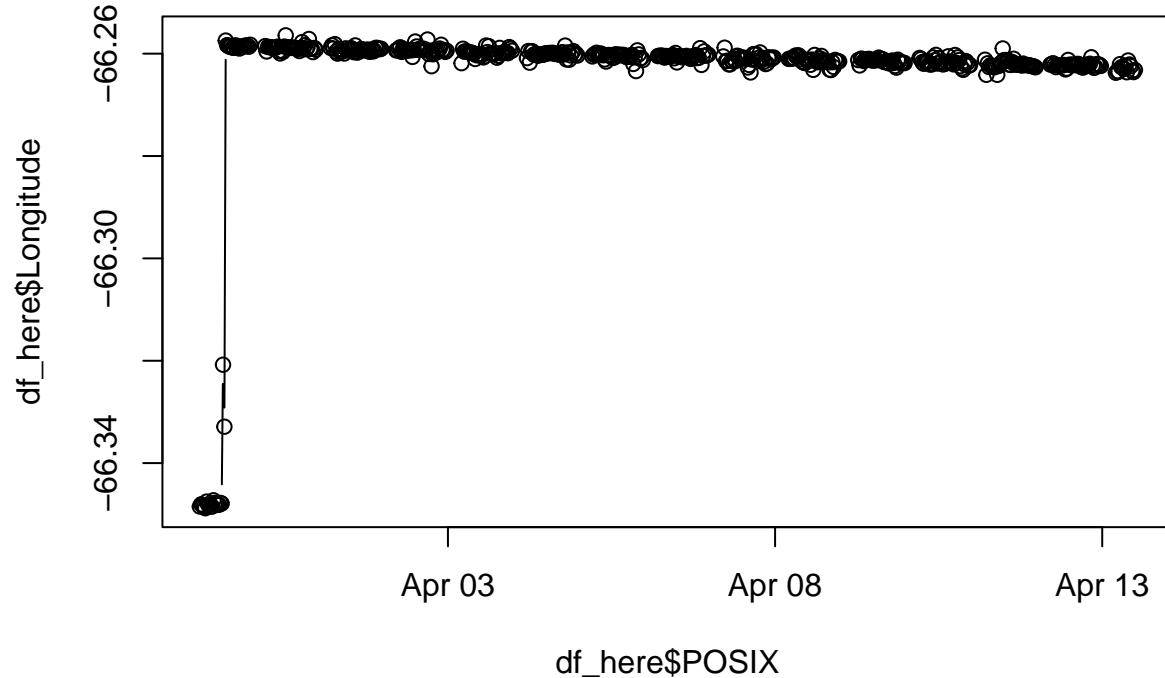
```
par(mfrow=c(2,1))
df <- df_org
idx <- which(df$UnitName == "Ismaage 300434066437720")

#
df_here <- df[idx,]
#df_here <- na.omit(df_here) # too general
lidx <- which(is.na(df_here$Longitude))
if (length(lidx) != 0) {df_here <- df_here[-lidx,]}
plot(df_here$POSIX,df_here$Longitude,type="b",main="Ismaage before outlier removed")
rlmfit <- rlm(df_here$Longitude ~ df_here$POSIX)
z <- unname(abs(rlmfit$residuals/madstat(rlmfit$residuals))) # counting goes off due to omitted values
jdx <- which(z > 7)
df_here$Longitude[jdx] <- NA
print(paste("I removed in Ismaage ",length(jdx)," outliers in longitude"))

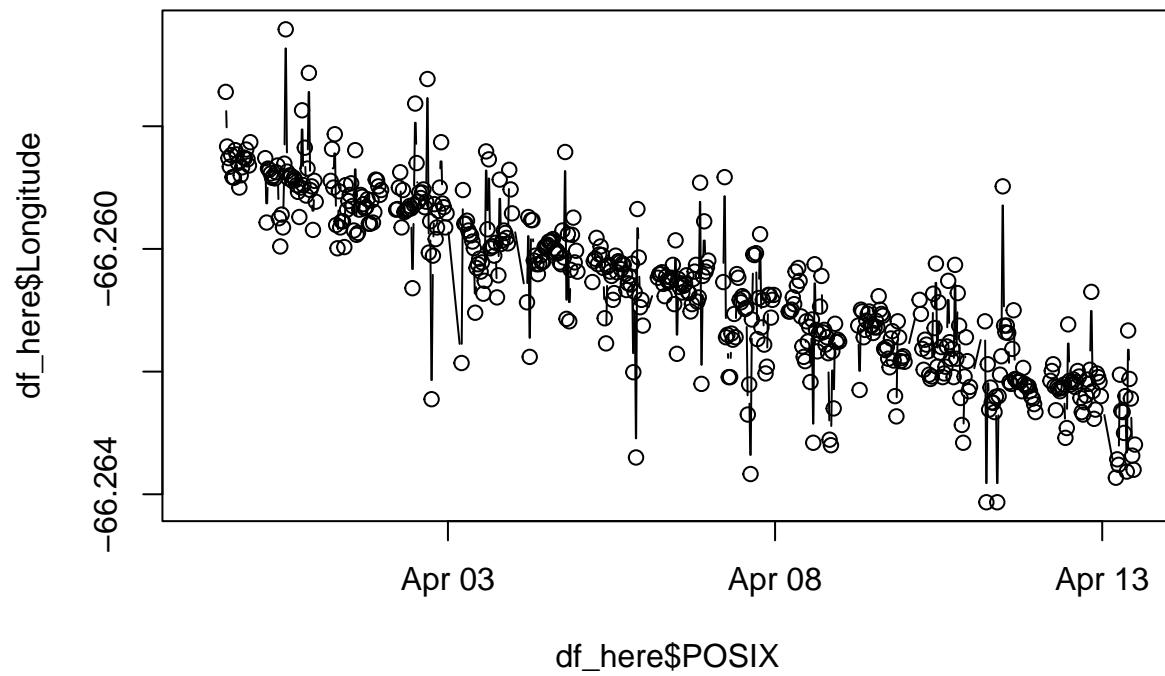
## [1] "I removed in Ismaage 19 outliers in longitude"

plot(df_here$POSIX,df_here$Longitude,type="b",main="Ismaage after outlier removed")
```

Ismaage before outlier removed



Ismaage after outlier removed



```

# remove the old
df_org <- df_org[-idx,]
# rbind the new
df_org <- rbind.data.frame(df_org,df_here)
df <- df_org

```

Also clean up Mallemuk lon

```

par(mfrow=c(2,1))
df <- df_org
idx <- which(df$UnitName == "Mallemuk 300434066431710")

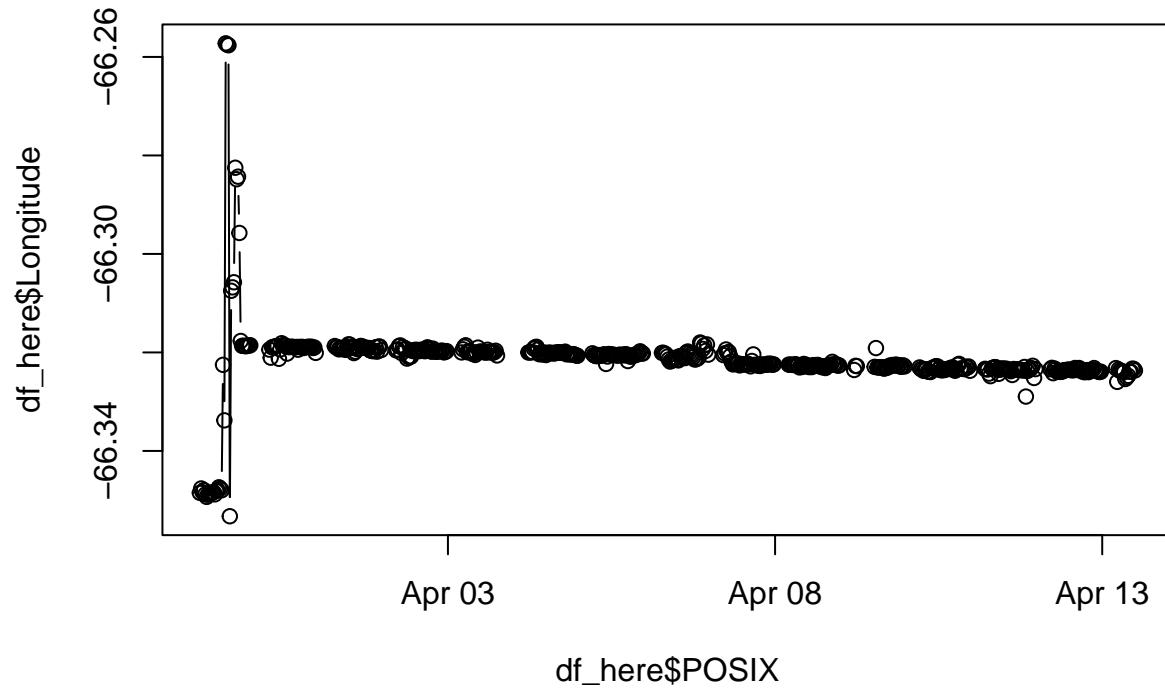
#
df_here <- df[idx,]
#df_here <- na.omit(df_here) # too general
lidx <- which(is.na(df_here$Longitude))
if (length(lidx) != 0) {df_here <- df_here[-lidx,]}
plot(df_here$POSIX,df_here$Longitude,type="b",main="Mallemuk before outlier removed")
rlmfit <- rlm(df_here$Longitude ~ df_here$POSIX)
z <- unname(abs(rlmfit$residuals/madstat(rlmfit$residuals))) # counting goes off due to omitted values
jdx <- which(z > 4)
df_here$Longitude[jdx] <- NA
print(paste("I removed in Mallemuk ",length(jdx)," outliers in longitude"))

## [1] "I removed in Mallemuk 29 outliers in longitude"

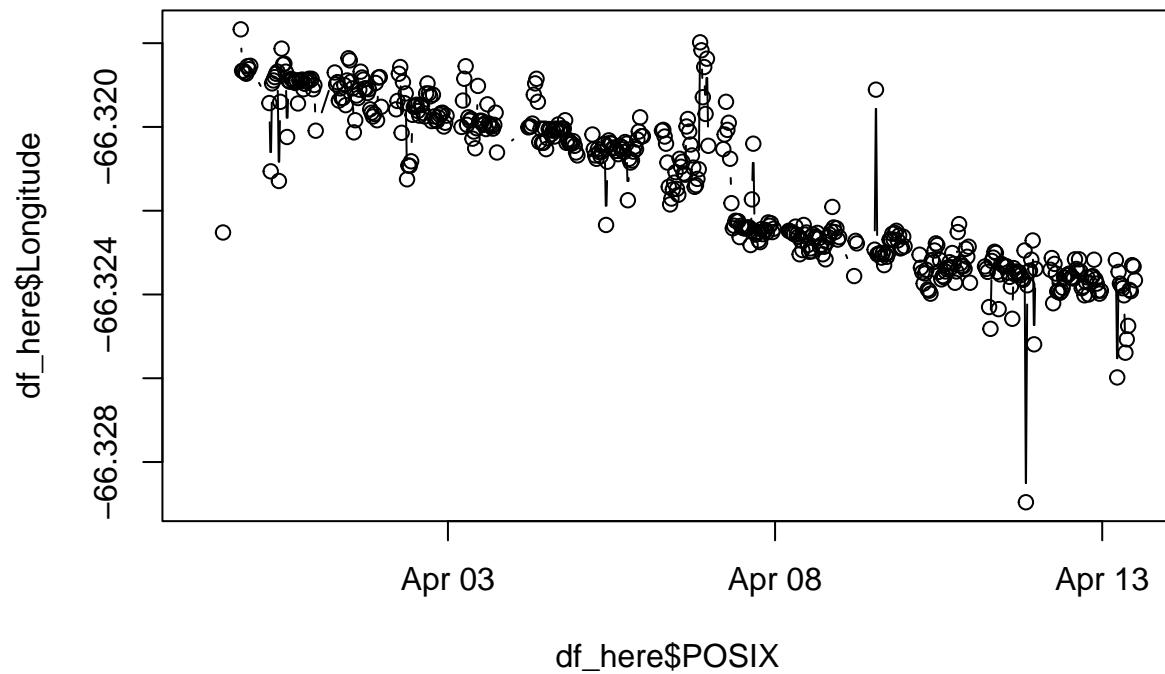
plot(df_here$POSIX,df_here$Longitude,type="b",main="Mallemuk after outlier removed")

```

Mallemuk before outlier removed



Mallemuk after outlier removed



```

# remove the old
df_org <- df_org[-idx,]
# rbind the new
df_org <- rbind.data.frame(df_org,df_here)
df <- df_org

```

Also clean up Mallemuk lat

```

par(mfrow=c(2,1))
df <- df_org
idx <- which(df$UnitName == "Mallemuk 300434066431710")

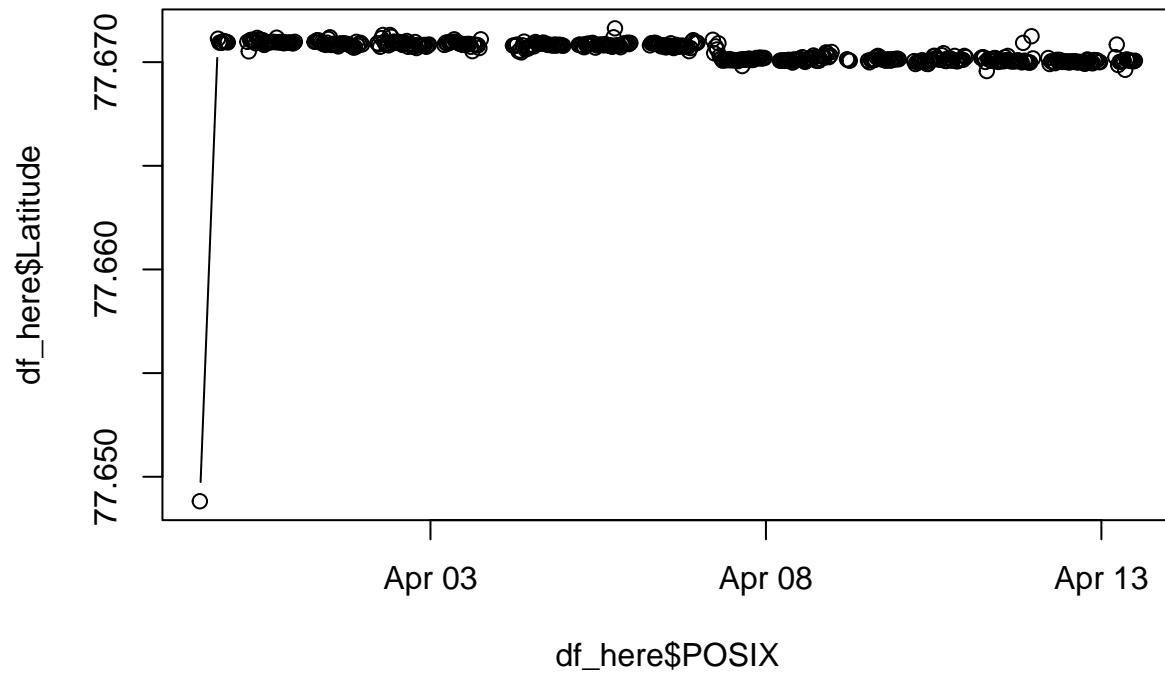
#
df_here <- df[idx,]
#df_here <- na.omit(df_here) # too general
lidx <- which(is.na(df_here$Longitude))
if (length(lidx) != 0) {df_here <- df_here[-lidx,]}
plot(df_here$POSIX,df_here$Latitude,type="b",main="Mallemuk before outlier removed")
rlmfit <- rlm(df_here$Latitude ~ df_here$POSIX)
z <- unname(abs(rlmfit$residuals/madstat(rlmfit$residuals))) # counting goes off due to omitted values
jdx <- which(z > 4)
df_here$Latitude[jdx] <- NA
print(paste("I removed in Mallemuk ",length(jdx)," outliers in Latitude"))

## [1] "I removed in Mallemuk 5 outliers in Latitude"

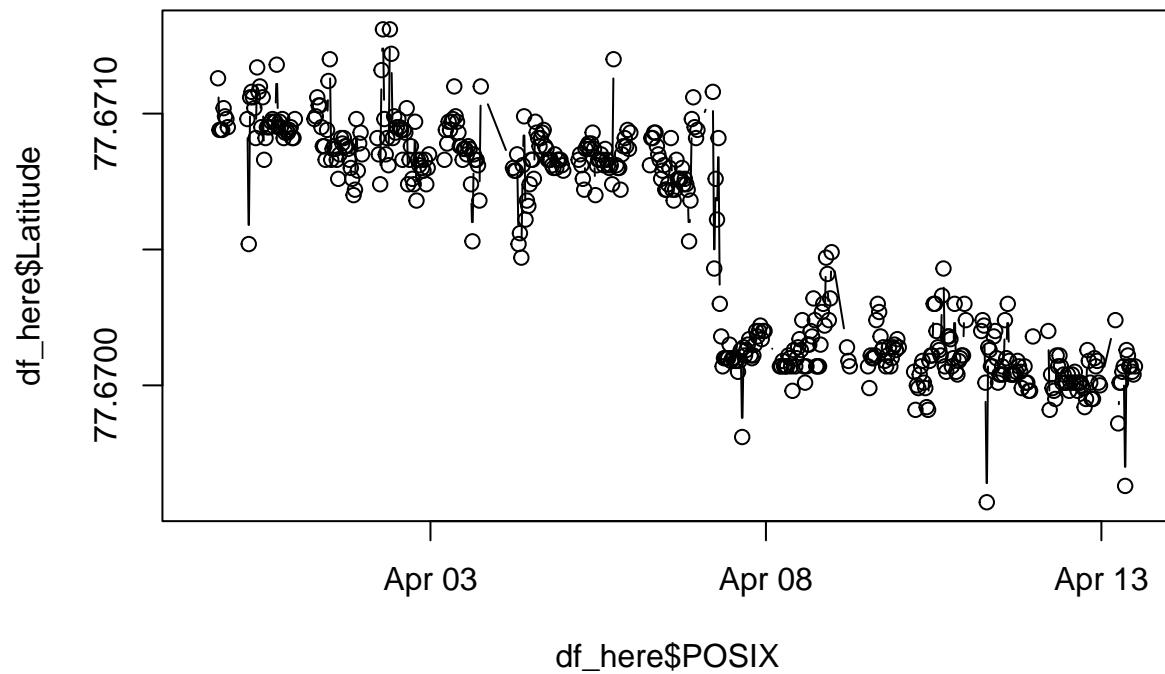
plot(df_here$POSIX,df_here$Latitude,type="b",main="Mallemuk after outlier removed")

```

Mallemuk before outlier removed



Mallemuk after outlier removed



```

# remove the old
df_org <- df_org[-idx,]
# rbind the new
df_org <- rbind.data.frame(df_org,df_here)
df <- df_org

```

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 # Earth mean radius [km]
  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 km
}

```

Define function to calculate speed

```

getSpeed <- function(time,lon,lat)
{
  rtod <- pi/180
  speed <- 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))
  }
  return(list("speed"=speed))
}

```

read

```

statdat <- NULL
alldf <- NULL
ic <- 1
for (istat in unique_names)
{
  par(mfrow=c(3,3))

```

```

idx <- which(df_org$UnitName == istat & df_org$Longitude < -65)
df2 <- df_org[idx,]
df2 <- kleaner(df2)
cnams <- colnames(df2)
time <- as.POSIXct(df2$"Timestamp UTC",tz="UTC")
#
lon <- df2$Longitude
lat <- df2$Latitude
temperature <- df2$"Temperature(°C)"
acceleration <- sqrt(df2$"AccelerationX(g)" ^2+df2$"AccelerationY(g)" ^2+df2$"AccelerationZ(g)" ^2)
lightlevel <- df2$"LightLevel"
speed0 <- df2$"GPS Speed(Km/h)"
speed <- getSpeed(time,lon,lat)$speed
speed <- c(speed,NA) # need one more value at the end

plot(lon,lat,main=istat,pch=19,cex=0.2,type="p")
plot(time,lon,main=istat,pch=19,cex=0.2,type="b")
plot(time,lat,main=istat,pch=19,cex=0.2,type="b")
if (length(temperature) > 3) {plot(time,temperature,main=istat,pch=19,cex=0.2,type="b")}
if (length(speed) > 3) {plot(time,speed,main=istat,pch=19,cex=0.2,type="b",ylim=c(0.01,5),log="y")}
if (length(acceleration) > 3) {
  plot(time,acceleration,main=istat,pch=19,cex=0.2,type="b")
  abline(h=1,col=2,lwd=3)
}
if (length(lightlevel) > 3) {plot(time,lightlevel,main=istat,pch=19,cex=0.2,type="b")}
# the set_of_variables
set <- c("time","lon","lat","temperature","acceleration","leightlevel","speed")
df3 <- cbind.data.frame(time,lon,lat)
colnames(df3)[1] <- "POSIX"
if (length(temperature) == nrow(df3)){ df3 <- cbind.data.frame(df3,temperature) }
if (length(acceleration) == nrow(df3)){ df3 <- cbind.data.frame(df3,acceleration) }
if (length(lightlevel) == nrow(df3)){ df3 <- cbind.data.frame(df3,lightlevel) }
if (length(speed) == nrow(df3)){ df3 <- cbind.data.frame(df3,speed) }
saveRDS(df3,paste0('OUTPUT/',istat,'.rds'))
#
}

## Warning: Unknown or uninitialized column: 'Temperature(°C)'.

## Warning: Unknown or uninitialized column: 'AccelerationX(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationY(g)'.

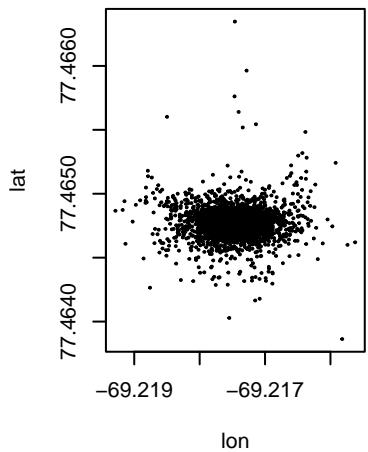
## Warning: Unknown or uninitialized column: 'AccelerationZ(g)'.

## Warning: Unknown or uninitialized column: 'LightLevel'.

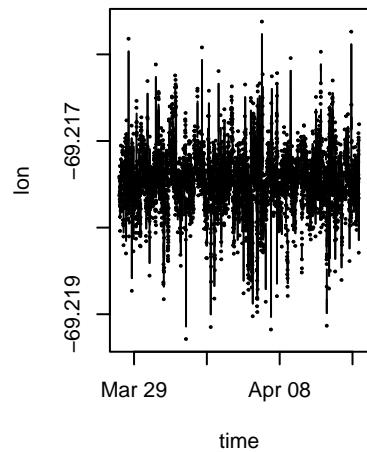
## Warning: Unknown or uninitialized column: 'GPS Speed(Km/h)'.

```

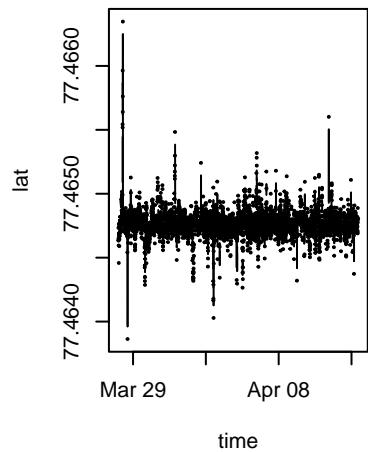
Fjeldrype 860640050232018



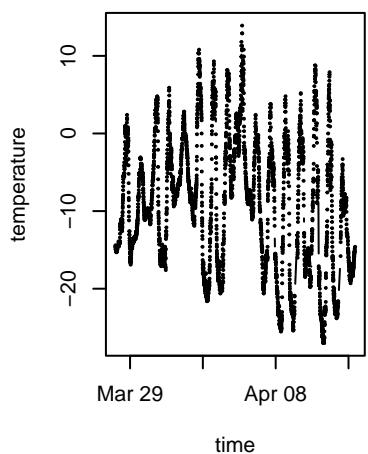
Fjeldrype 860640050232018



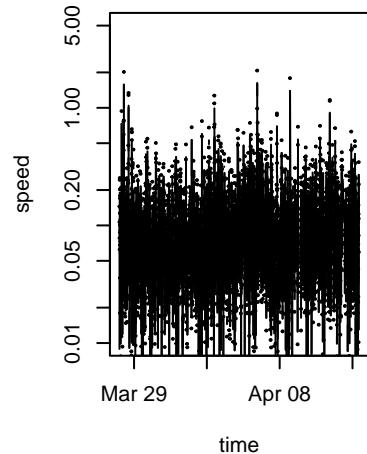
Fjeldrype 860640050232018



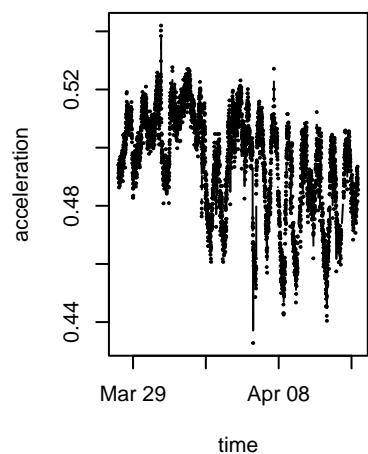
Fjeldrype 860640050232018



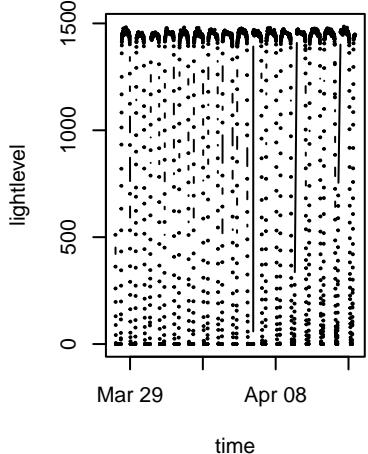
Fjeldrype 860640050232018



Fjeldrype 860640050232018



Fjeldrype 860640050232018



```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 6 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialized column: 'Temperature(°C)'.

## Warning: Unknown or uninitialized column: 'AccelerationX(g)'.

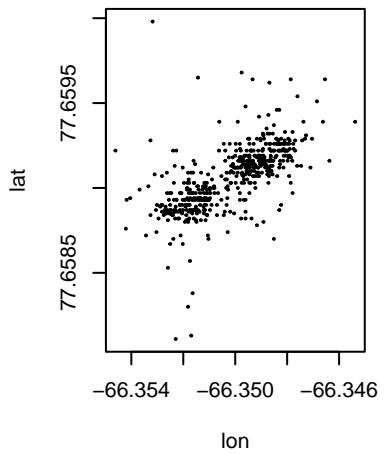
## Warning: Unknown or uninitialized column: 'AccelerationY(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationZ(g)'.

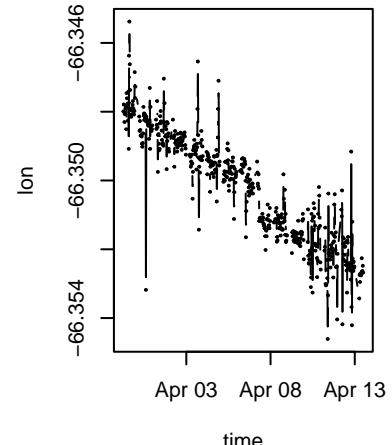
## Warning: Unknown or uninitialized column: 'LightLevel'.

## Warning: Unknown or uninitialized column: 'GPS Speed(Km/h)'.
```

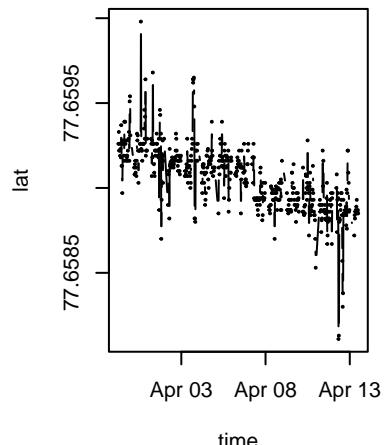
Havterne 300434066435700



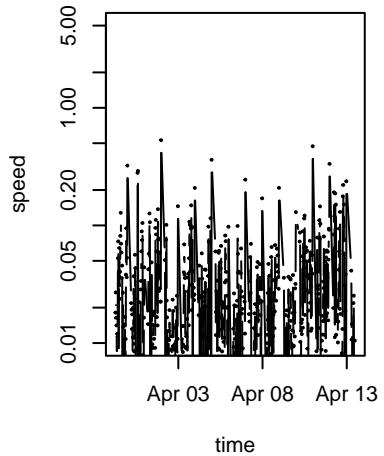
Havterne 300434066435700



Havterne 300434066435700



Havterne 300434066435700



```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 2 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialized column: 'Temperature(°C)'.

## Warning: Unknown or uninitialized column: 'AccelerationX(g)'.

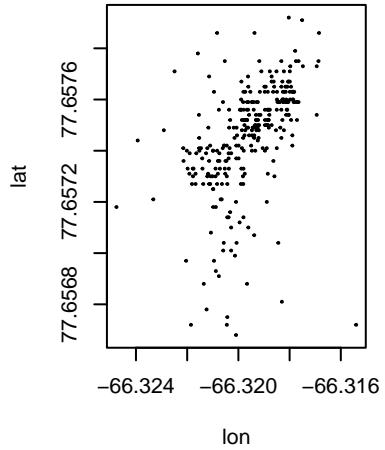
## Warning: Unknown or uninitialized column: 'AccelerationY(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationZ(g)'.

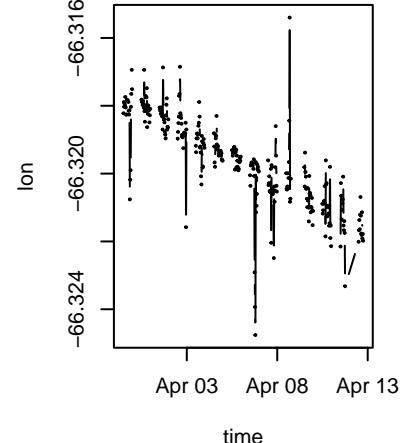
## Warning: Unknown or uninitialized column: 'LightLevel'.

## Warning: Unknown or uninitialized column: 'GPS Speed(Km/h)'.
```

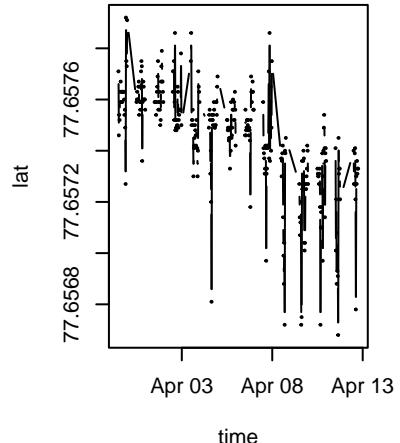
Soekonge 300434066433690



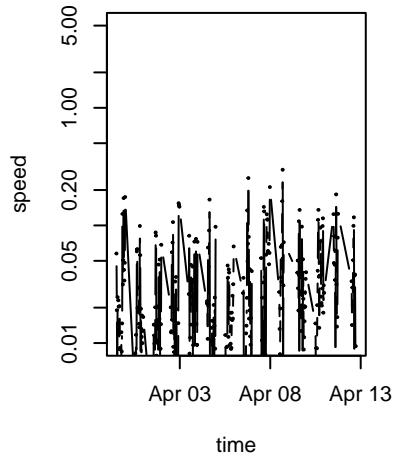
Soekonge 300434066433690



Soekonge 300434066433690



Soekonge 300434066433690



```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 6 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialized column: 'Temperature(°C)'.

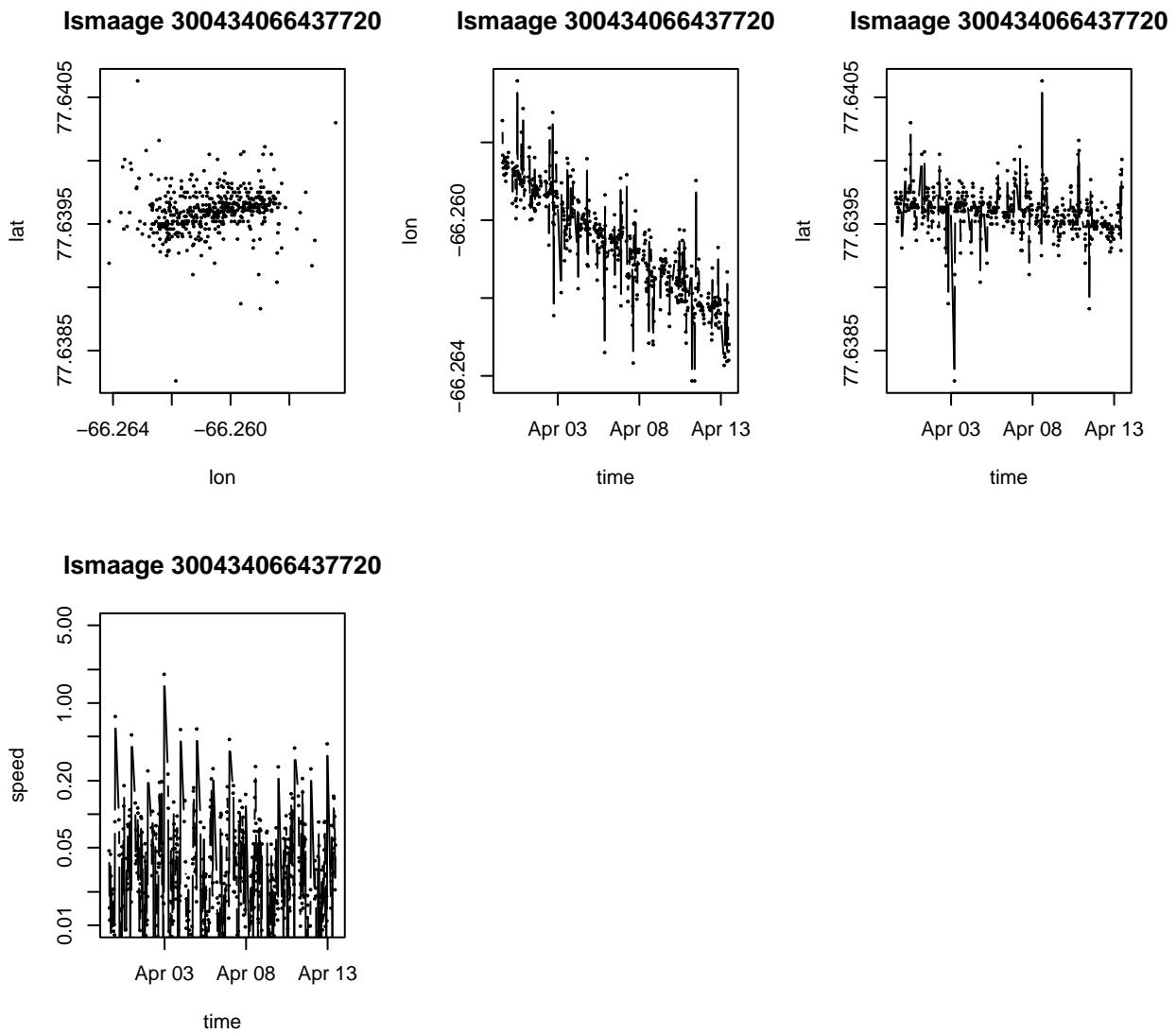
## Warning: Unknown or uninitialized column: 'AccelerationX(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationY(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationZ(g)'.

## Warning: Unknown or uninitialized column: 'LightLevel'.

## Warning: Unknown or uninitialized column: 'GPS Speed(Km/h)'.
```



```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 6 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialized column: 'Temperature(°C)'.

## Warning: Unknown or uninitialized column: 'AccelerationX(g)'.

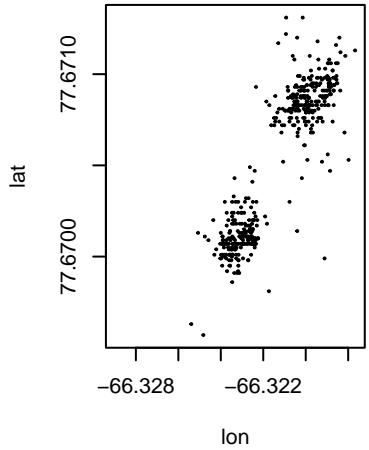
## Warning: Unknown or uninitialized column: 'AccelerationY(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationZ(g)'.

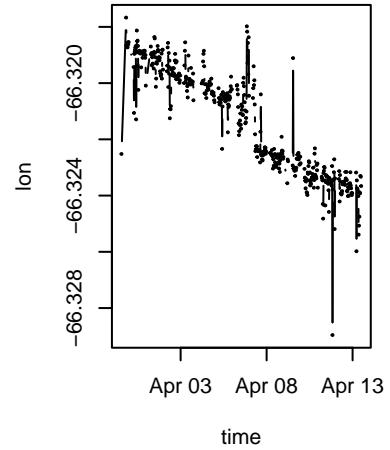
## Warning: Unknown or uninitialized column: 'LightLevel'.

## Warning: Unknown or uninitialized column: 'GPS Speed(Km/h)'.
```

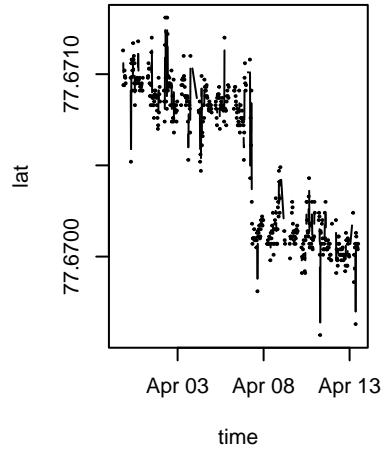
Mallemuk 300434066431710



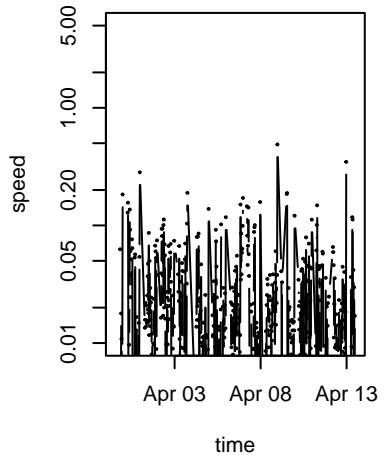
Mallemuk 300434066431710



Mallemuk 300434066431710



Mallemuk 300434066431710



```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 3 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialized column: 'Temperature(°C)'.

## Warning: Unknown or uninitialized column: 'AccelerationX(g)'.

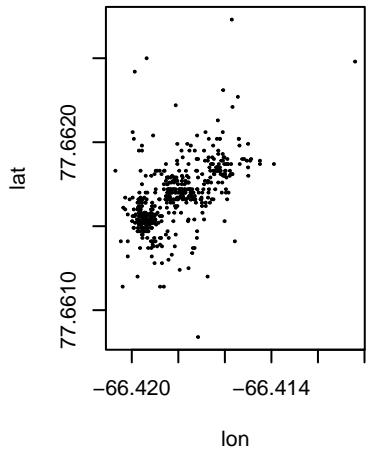
## Warning: Unknown or uninitialized column: 'AccelerationY(g)'.

## Warning: Unknown or uninitialized column: 'AccelerationZ(g)'.

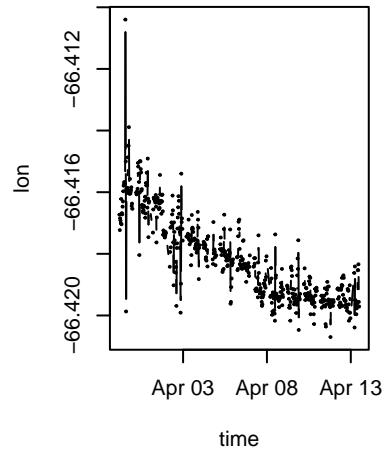
## Warning: Unknown or uninitialized column: 'LightLevel'.

## Warning: Unknown or uninitialized column: 'GPS Speed(Km/h)'.
```

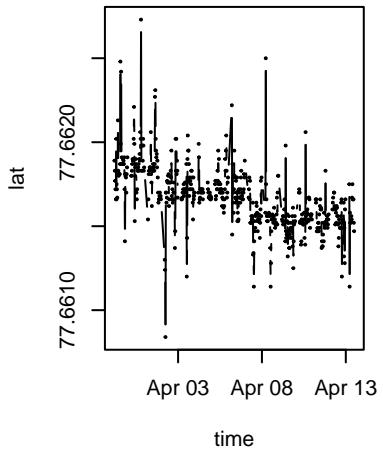
Edder 300434066433700



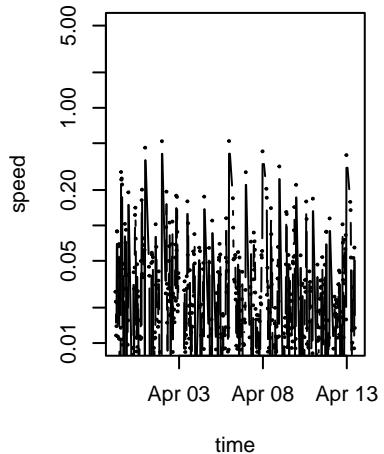
Edder 300434066433700



Edder 300434066433700

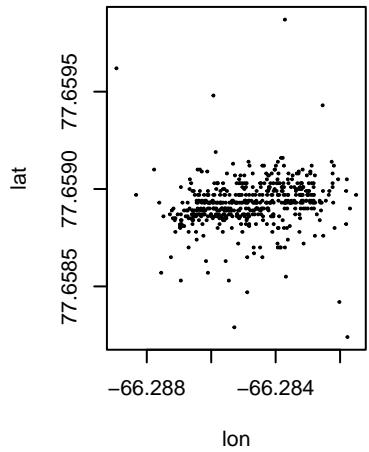


Edder 300434066433700

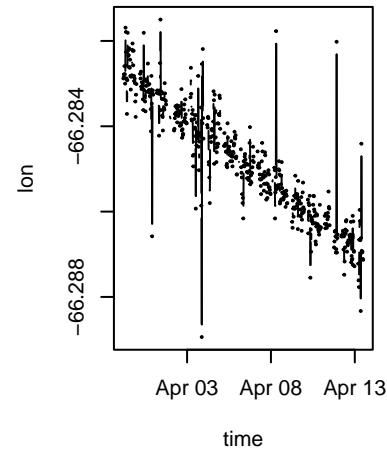


```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 4 y values <= 0 omitted from
## logarithmic plot
```

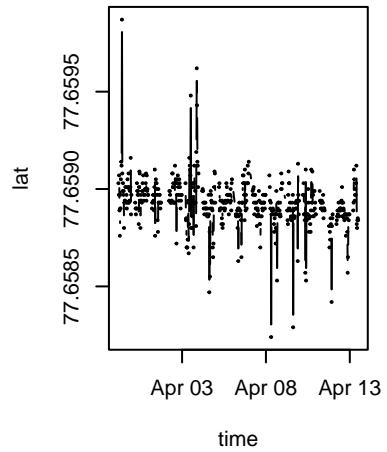
Havoern 300434066437680



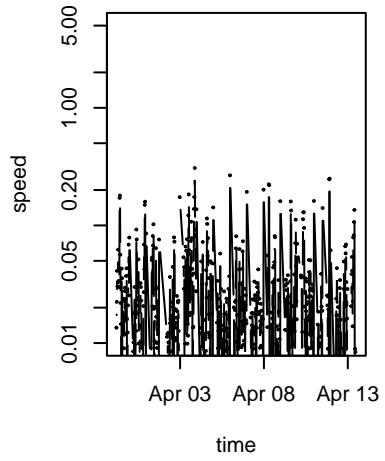
Havoern 300434066437680

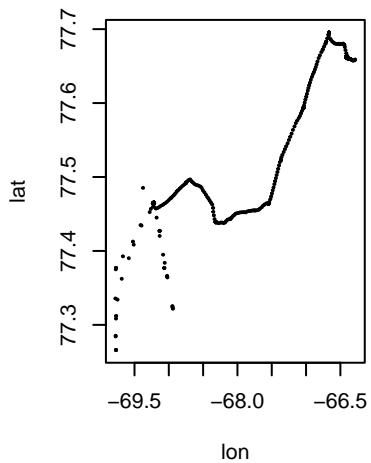
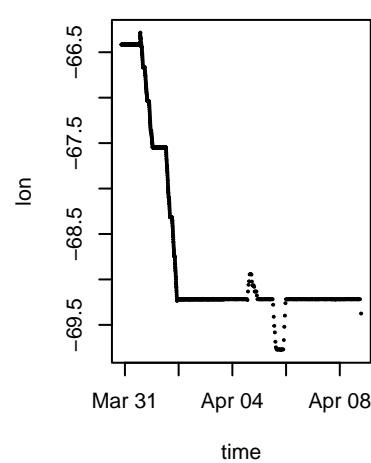
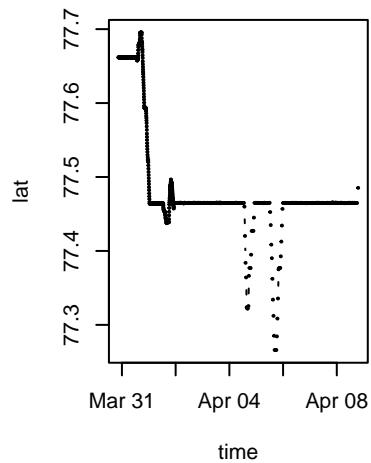
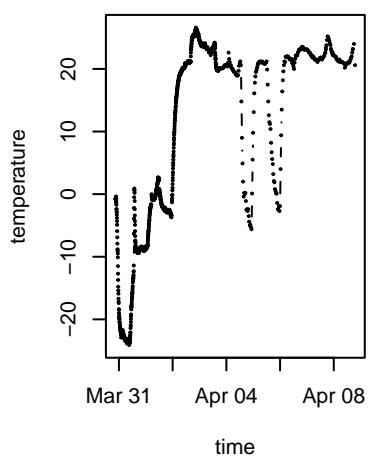
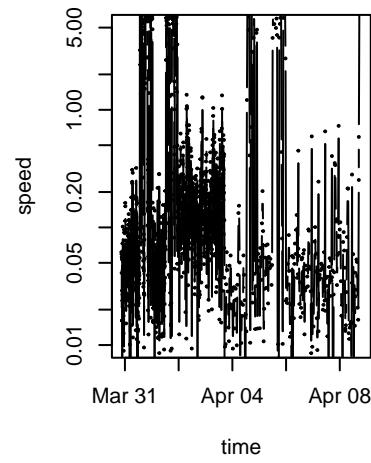
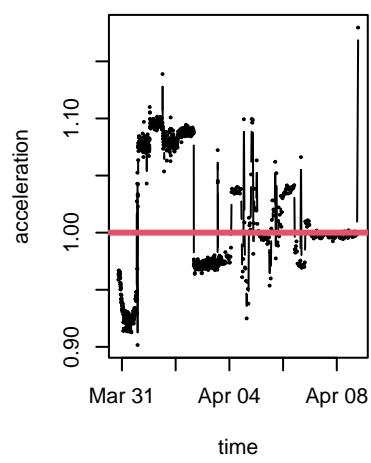
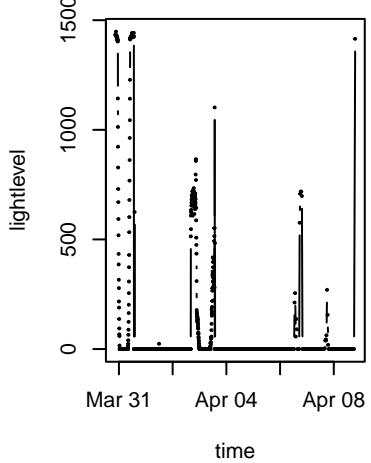


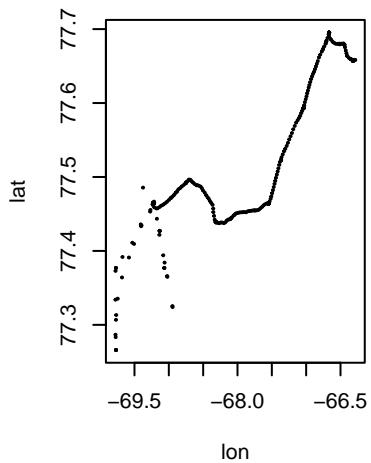
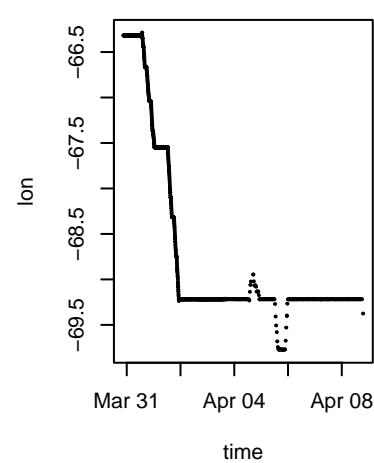
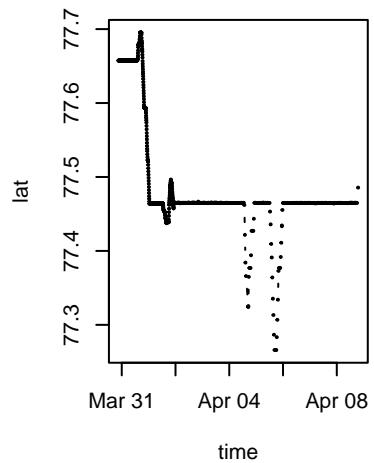
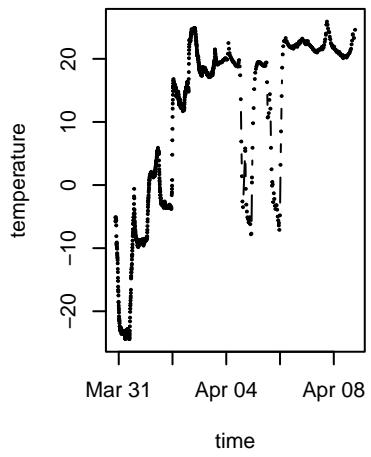
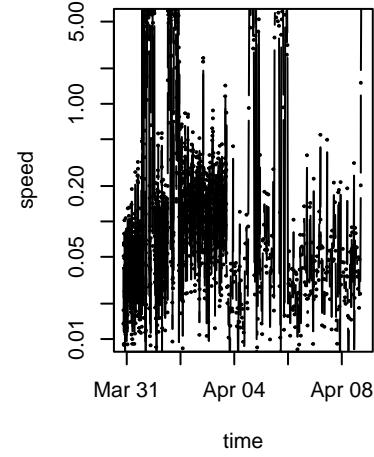
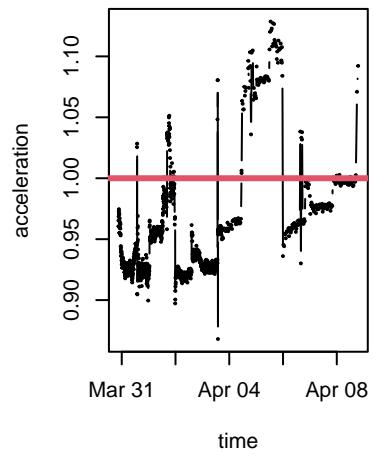
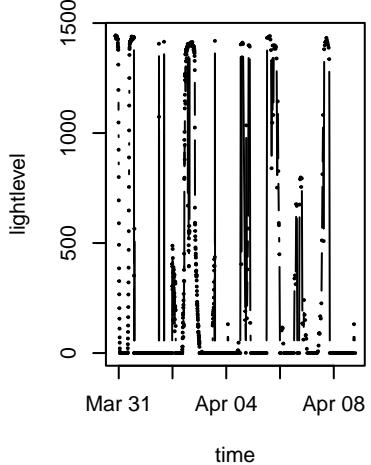
Havoern 300434066437680



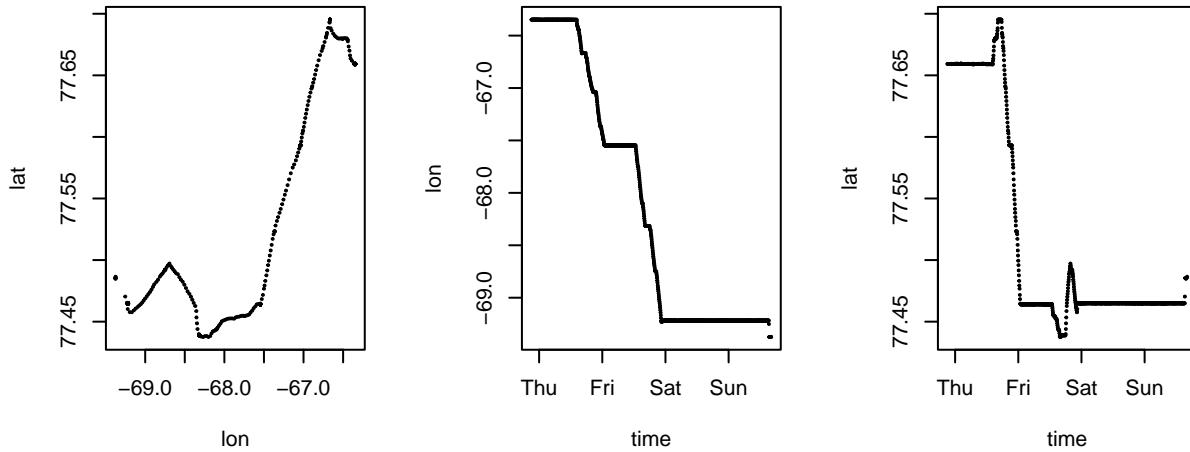
Havoern 300434066437680



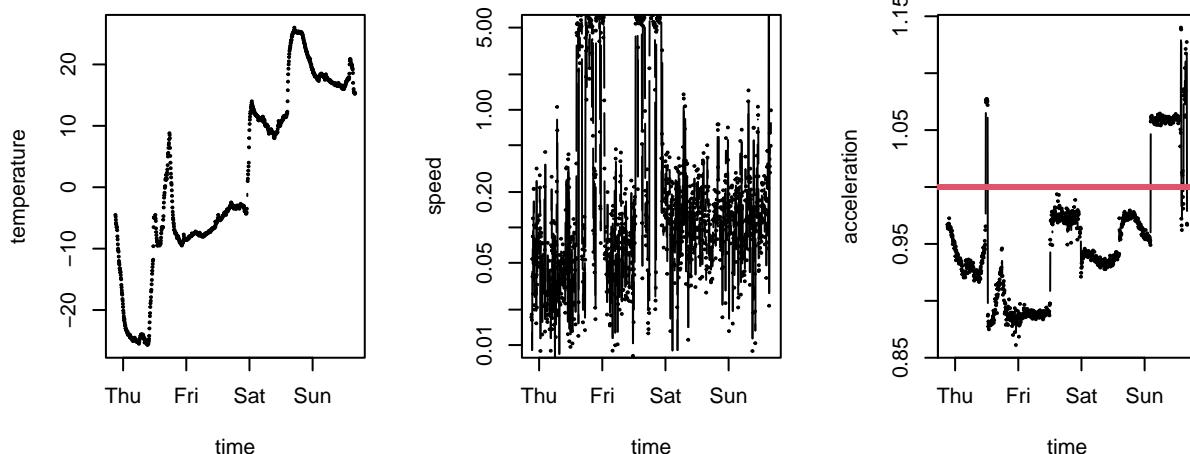
Landsvale 860640050251737**Landsvale 860640050251737****Landsvale 860640050251737****Landsvale 860640050251737****Landsvale 860640050251737****Landsvale 860640050251737****Landsvale 860640050251737**

Ravn 860640050244401**Ravn 860640050244401****Ravn 860640050244401****Ravn 860640050244401****Ravn 860640050244401****Ravn 860640050244401****Ravn 860640050244401**

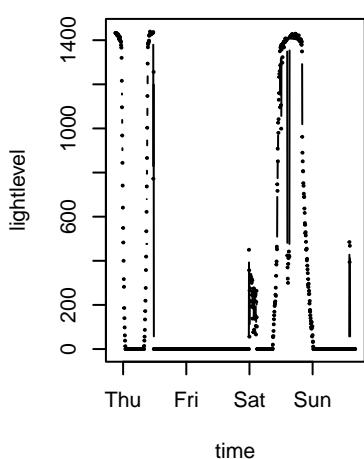
Strandskade 860640050251350 Strandskade 860640050251350 Strandskade 860640050251350



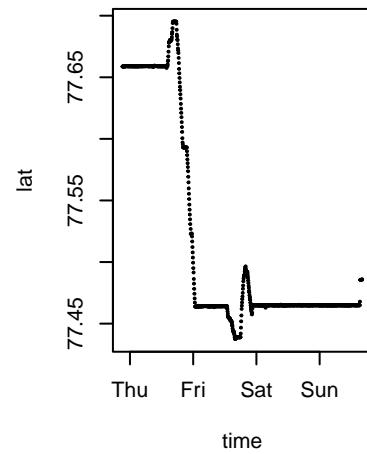
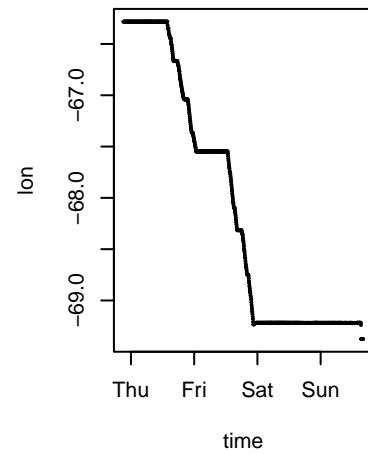
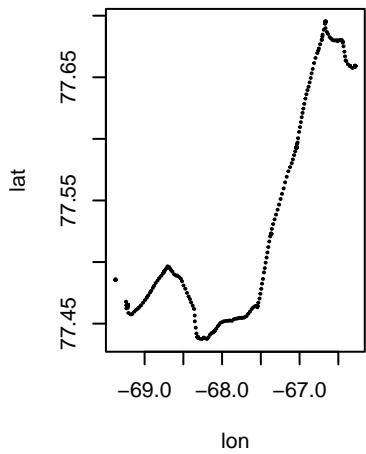
Strandskade 860640050251350 Strandskade 860640050251350 Strandskade 860640050251350



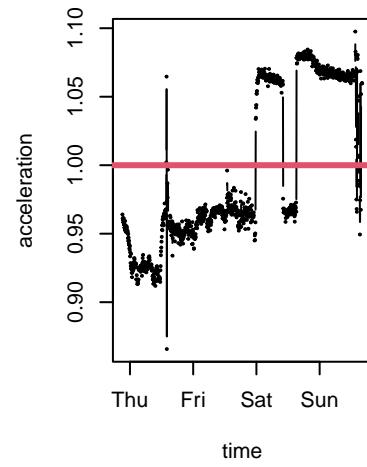
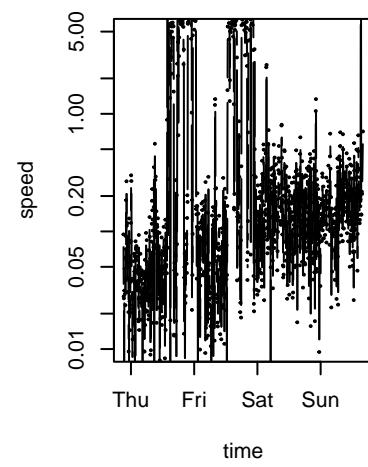
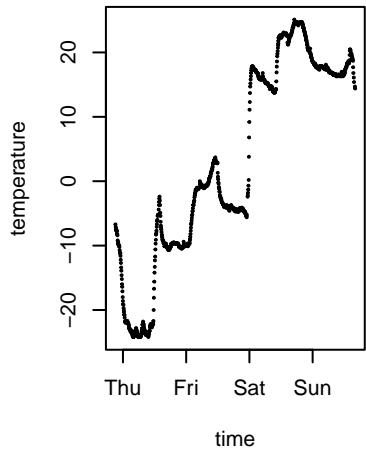
Strandskade 860640050251350



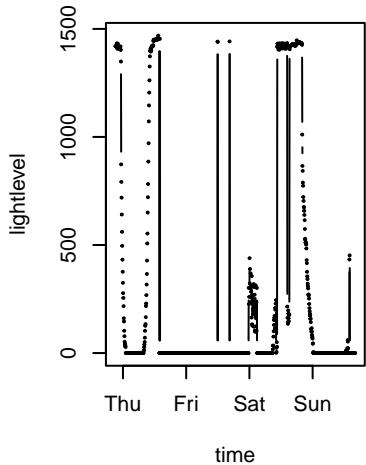
Stenpikkere 86064005024406: Stenpikkere 86064005024406: Stenpikkere 86064005024406:



Stenpikkere 86064005024406: Stenpikkere 86064005024406: Stenpikkere 86064005024406:



Stenpikkere 86064005024406:



Remove dogsled series

```
if_remove_dogsled <- TRUE
if (if_remove_dogsled){
  unique_names
  idx <- which(unique_names == "Landsvale 860640050251737")
  unique_names <- unique_names[-idx]
  idx <- which(unique_names == "Strandskade 860640050251356")
  unique_names <- unique_names[-idx]
  idx <- which(unique_names == "Ravn 860640050244401")
  unique_names <- unique_names[-idx]
  idx <- which(unique_names == "Stenpikkere 860640050244062")
  unique_names <- unique_names[-idx]
  unique_names
}

## [1] "Fjeldrype 860640050232018" "Havterne 300434066435700"
## [3] "Soekonge 300434066433690"   "Ismaage 300434066437720"
## [5] "Mallermuk 300434066431710"  "Edder 300434066433700"
## [7] "Havoern 300434066437680"

pdf("FIGURES/speed_vs_time.pdf")
# same t-axis
ic <- 1
legtext <- NULL
colnames <- c("green", "red", "blue", "azure", "orange", "black", "purple", "salmon", "grey", "hotpink", "yellow")
for (istat in unique_names)
{
  df <- readRDS(paste0('OUTPUT/' , istat , '.rds'))
  idx <- order(df$POSIX)
  df <- df[idx,]
  if (ic == 1){ plot(df$POSIX, df$speed, type="b", pch=ic+14, cex=0.8, xlab="Date/Time", ylab="speed [km/h]", xlim=c(as.POSIXct("2022-03-28 00:00:00", tz="UTC"), as.POSIXct("2022-03-31 23:59:59", tz="UTC")))
  else{ lines(df$POSIX, df$speed, col=colnames[ic], type="b", cex=0.8, pch=ic+14) }
  legtext <- c(legtext, paste(strsplit(istat, ' ')[[1]][1], colnames[ic]))
  ic <- ic+1
}
legend("topright", legend=legtext, cex=0.6)
#-----#
dev.off()

## pdf
## 2
```

relative to Fjeldrype

```
par(mfrow=c(1,1))
```

```

base_station <- readRDS("OUTPUT/Fjeldrype 860640050232018.rds")

alldf <- NULL
#
for (jstat in 1:length(unique_names))
{
  print(jstat)
  statname <- unique_names[jstat]
  print(statname)

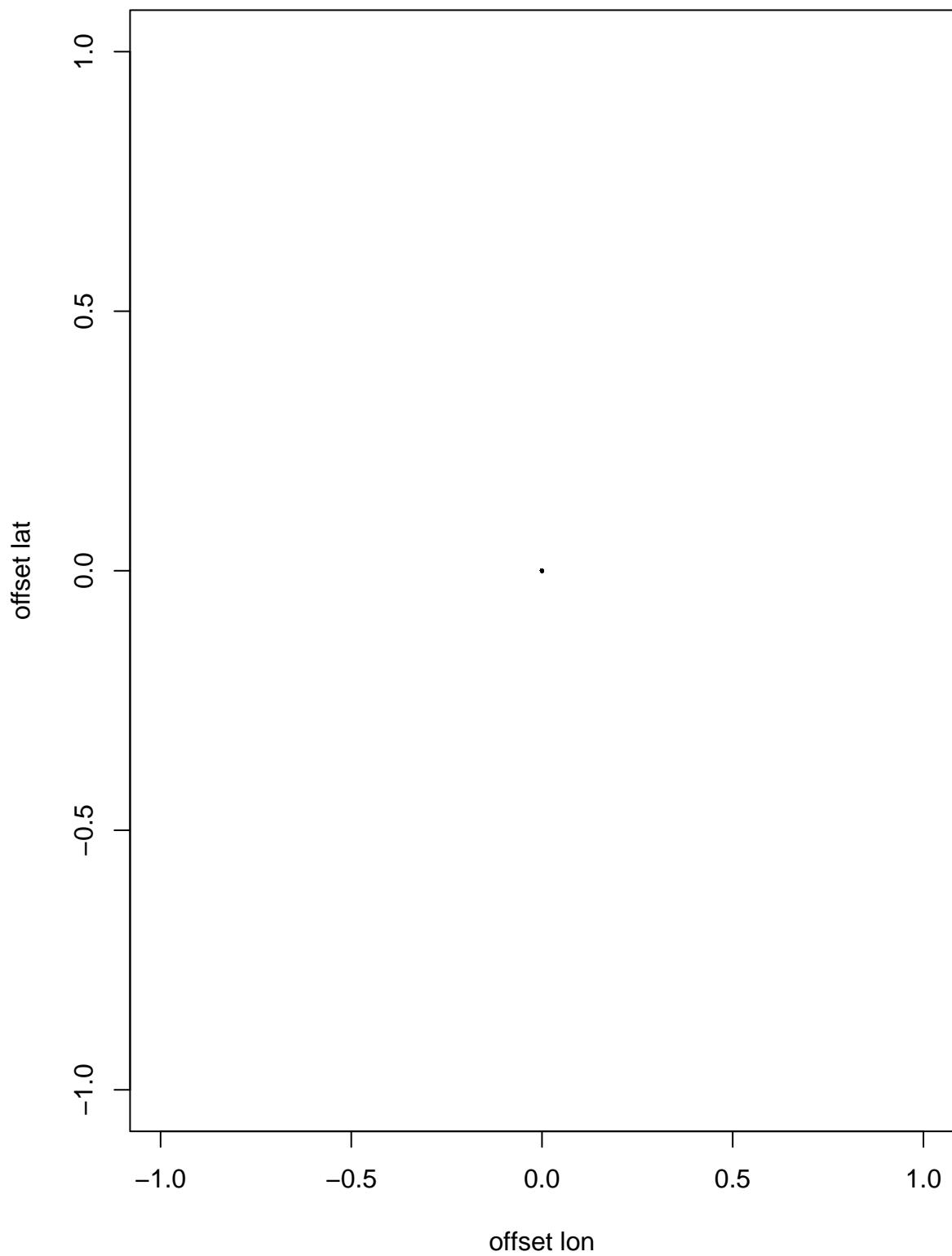
  other <- readRDS(paste0("OUTPUT/",statname,".rds"))

  tmin <- max(c(min(base_station$POSIX),min(other$POSIX)))
  tmax <- min(max(base_station$POSIX),max(other$POSIX))
  idx <- which(base_station$POSIX >= tmin & base_station$POSIX <= tmax)
  base_station <- base_station[idx,]
  idx <- which(other$POSIX >= tmin & other$POSIX <= tmax)
  other <- other[idx,]
  #Interpolate to same times as in 'base_station'
  common_t <- base_station$POSIX
  lon_other_interp <- approx(other$POSIX,other$lon,base_station$POSIX,na.rm=TRUE)$y
  lat_other_interp <- approx(other$POSIX,other$lat,base_station$POSIX,na.rm=TRUE)$y
  #
  interp_lon <- na.omit(cbind.data.frame(common_t,lon_other_interp))
  colnames(interp_lon) <- c("POSIX","lon_i")
  interp_lat <- na.omit(cbind.data.frame(common_t,lat_other_interp))
  colnames(interp_lat) <- c("POSIX","lat_i")
  together <- merge(base_station,interp_lon,by="POSIX")
  together <- merge(together,interp_lat,by="POSIX")
  delta_lon <- together$lon_i-together$lon
  delta_lat <- together$lat_i-together$lat
  together <- cbind(together,delta_lon,delta_lat)
  saveRDS(together,paste0("OUTPUT/processed_",statname,".rds"))
  print(paste(statname,round(sd(together$delta_lon),4),round(sd(together$delta_lat),4)))
  plot(together$delta_lon,together$delta_lat,main=statname,xlab="offset lon",ylab="offset lat",pch=19,col="red")
  #
}

## [1] 1
## [1] "Fjeldrype 860640050232018"
## [1] "Fjeldrype 860640050232018 0 0"

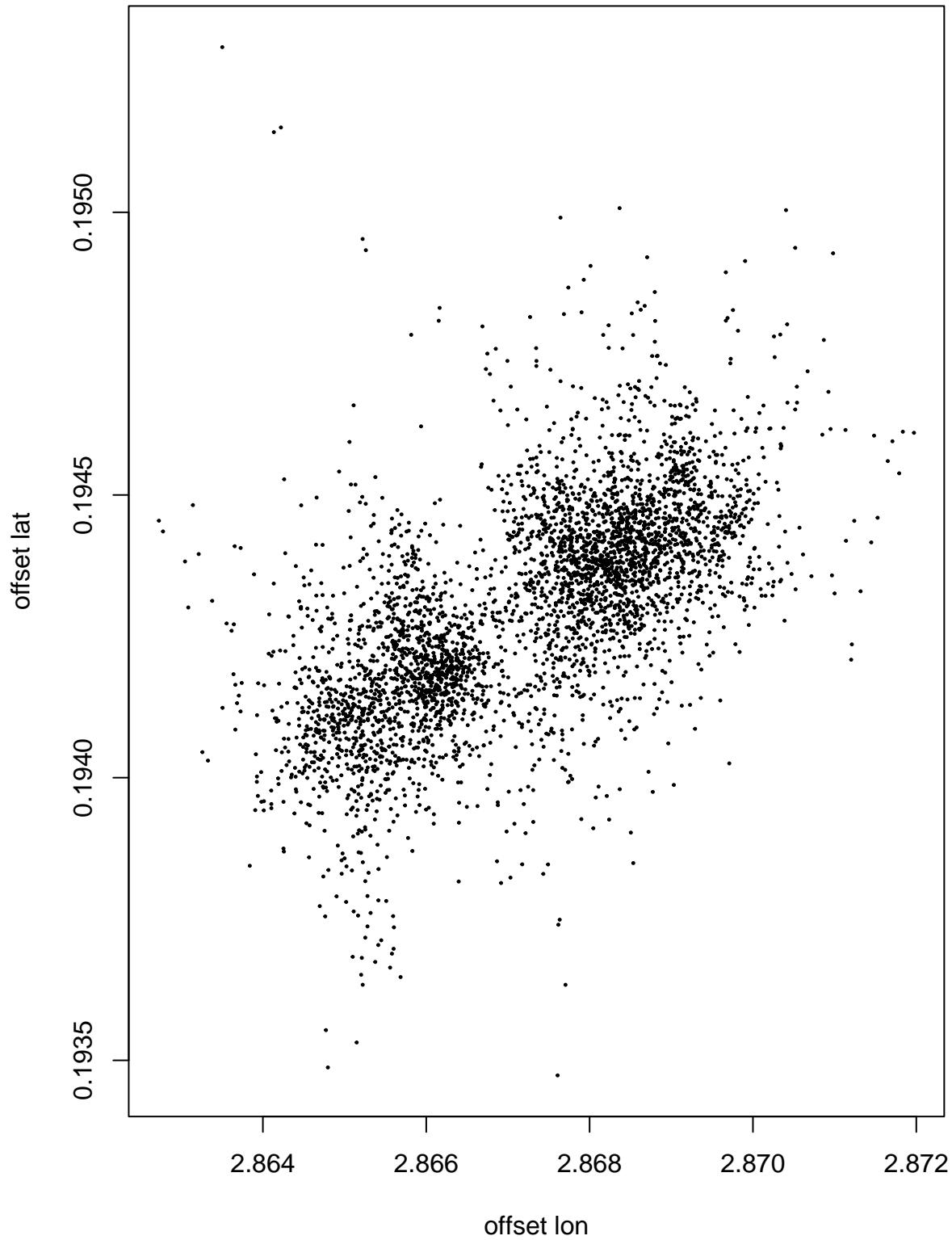
```

Fjeldrype 860640050232018



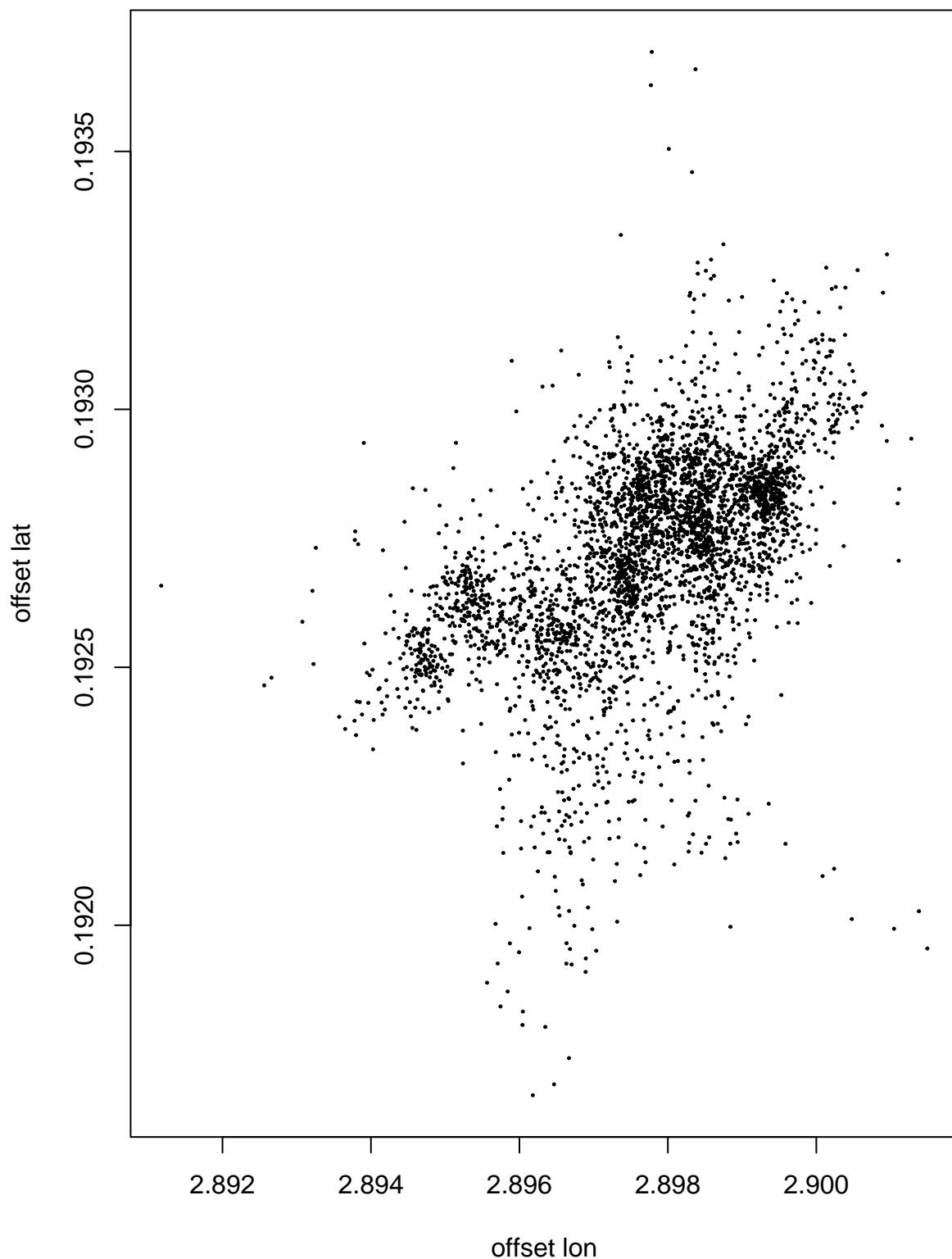
```
## [1] 2
## [1] "Havterne 300434066435700"
## [1] "Havterne 300434066435700 0.0016 2e-04"
```

Havterne 300434066435700



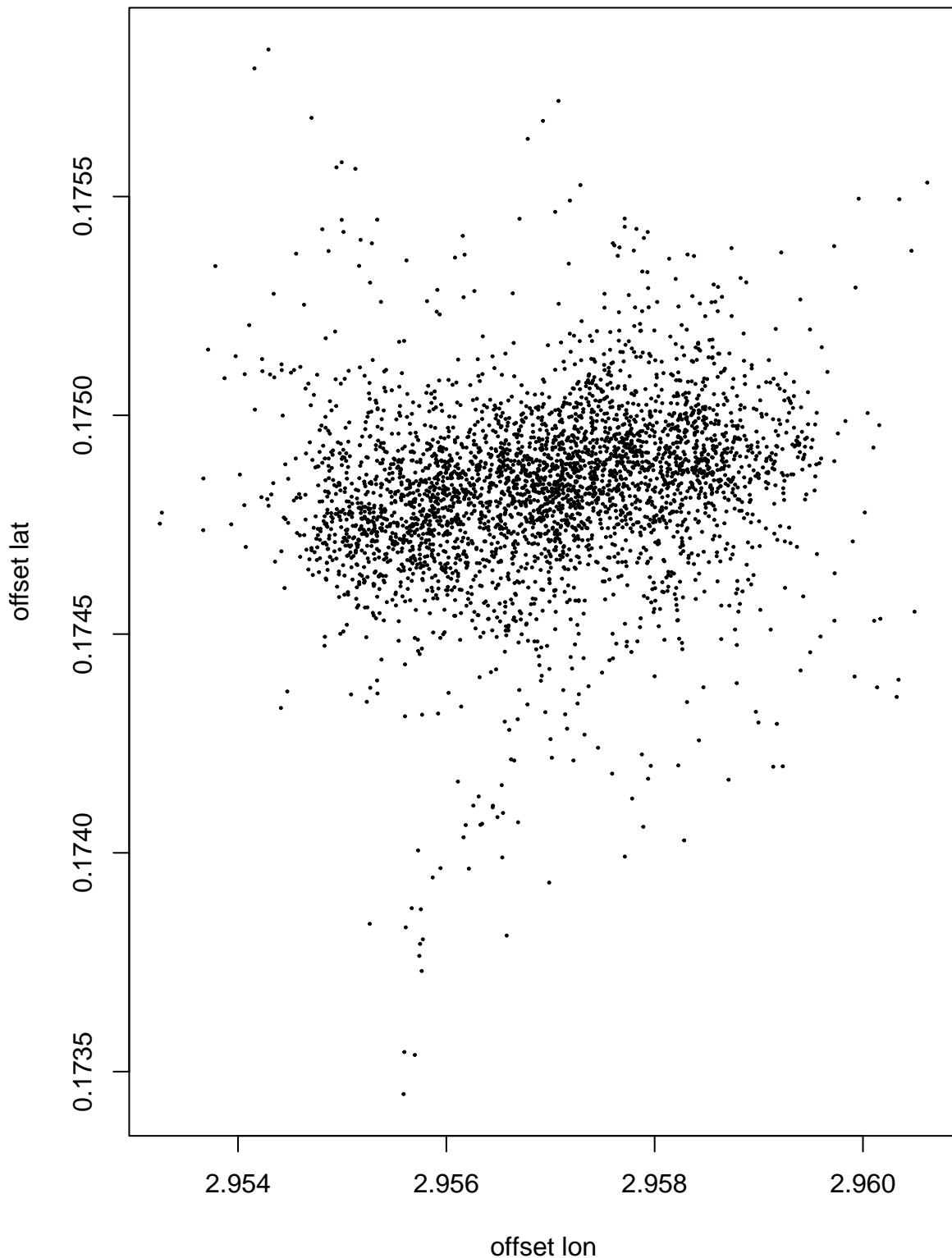
```
## [1] 3
## [1] "Soekonge 300434066433690"
## [1] "Soekonge 300434066433690 0.0014 2e-04"
```

Soekonge 300434066433690



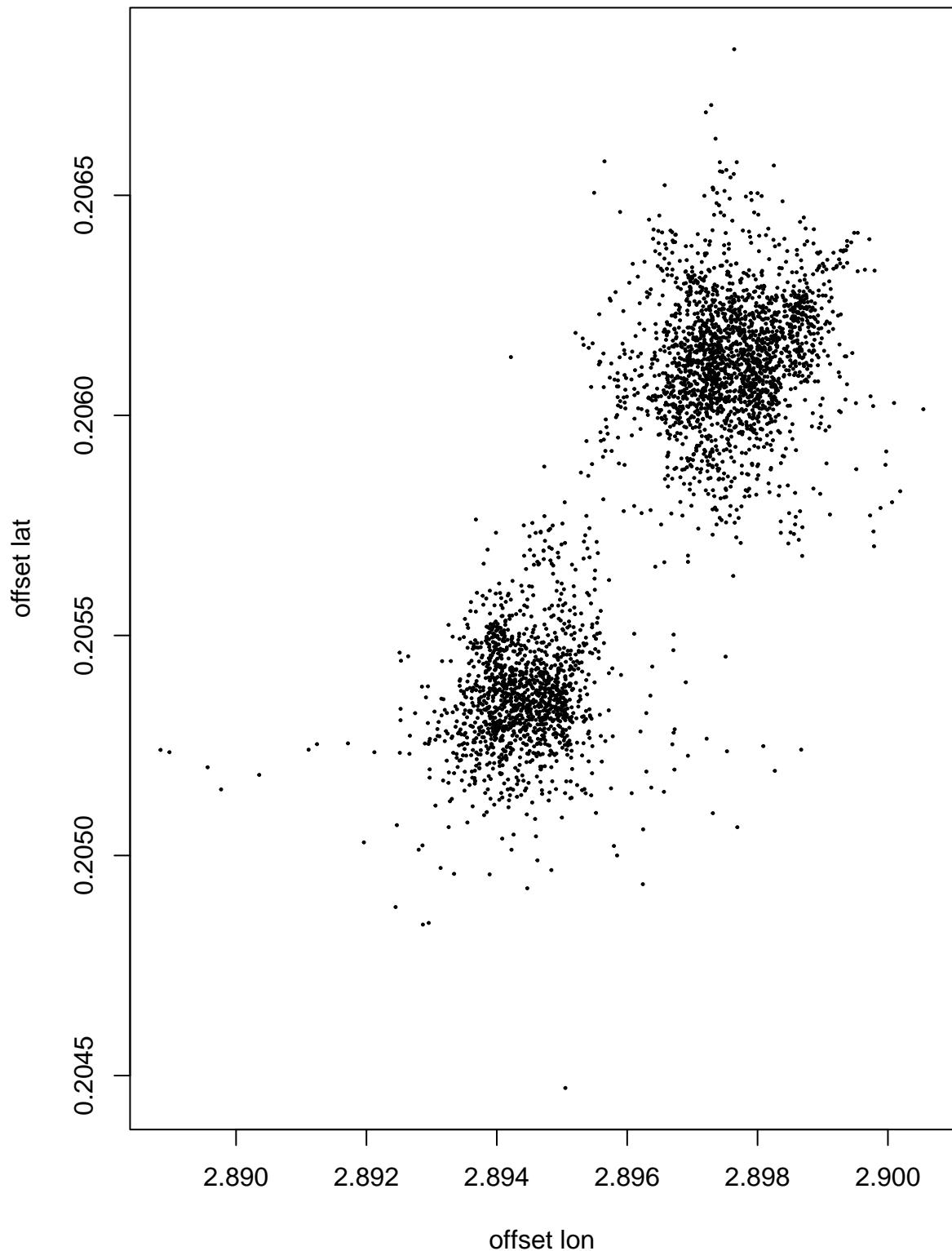
```
## [1] 4
## [1] "Ismaage 300434066437720"
## [1] "Ismaage 300434066437720 0.0012 2e-04"
```

Ismaage 300434066437720



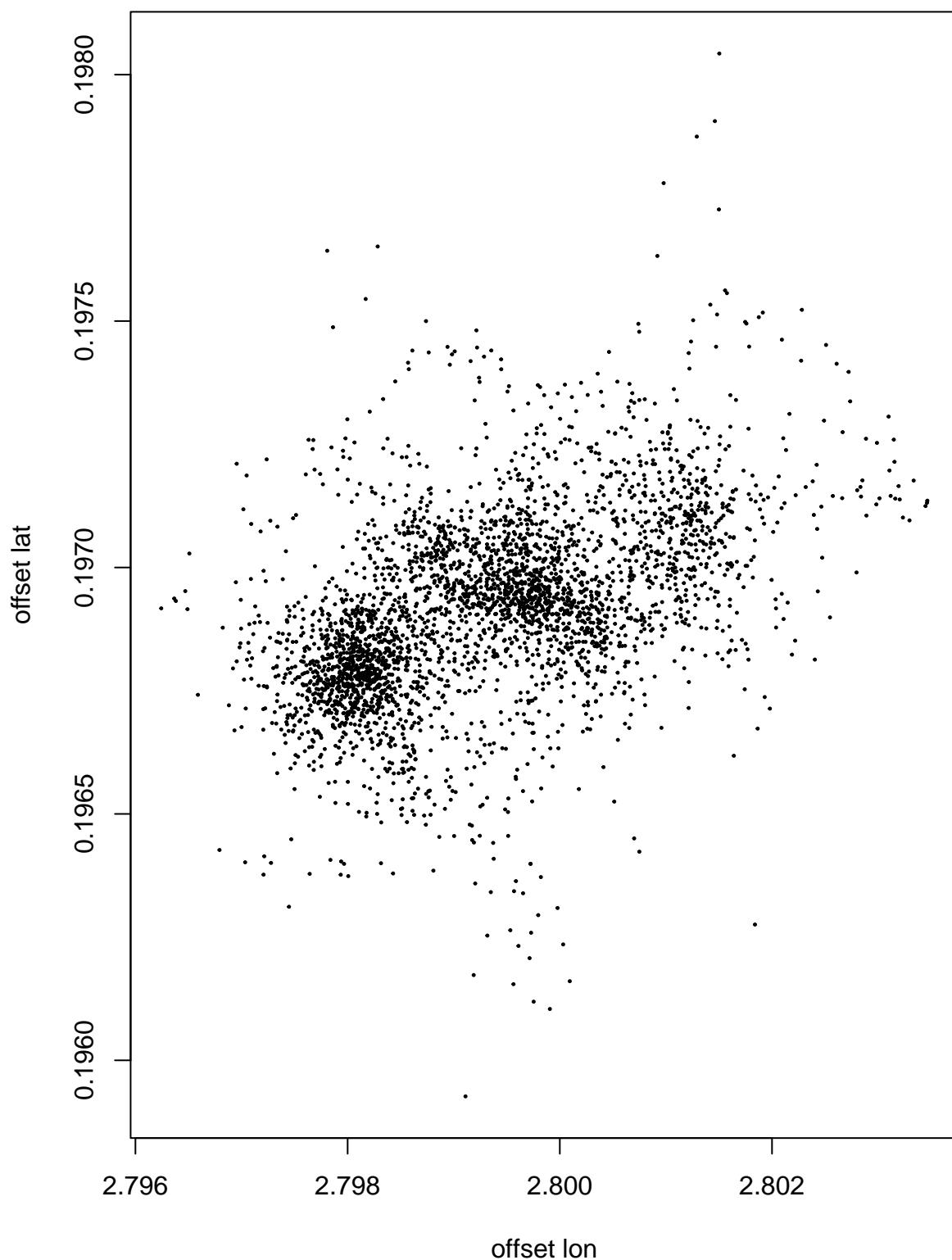
```
## [1] 5
## [1] "Mallemuk 300434066431710"
## [1] "Mallemuk 300434066431710 0.0018 4e-04"
```

Mallemuk 300434066431710



```
## [1] 6
## [1] "Edder 300434066433700"
## [1] "Edder 300434066433700 0.0012 2e-04"
```

Edder 300434066433700



```
## [1] 7
## [1] "Havoern 300434066437680"
## [1] "Havoern 300434066437680 0.0012 1e-04"
```

Havoern 300434066437680

