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"
```

```
if_remove_dogsled <- FALSE
```

Define that function for later use

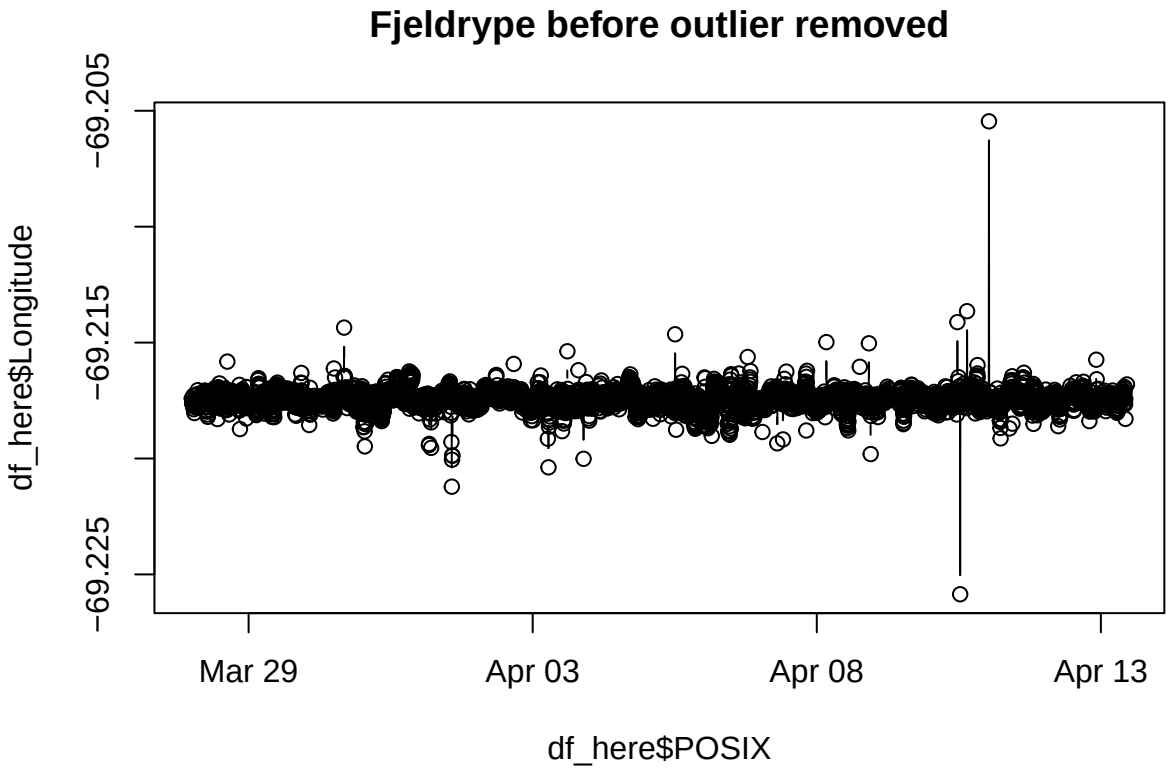
```
kleaner <- function(df_in)
{
  # clean out columns with only NA
  listen <- NULL
  for (icol in 1:ncol(df_in))  {    listen <- rbind(listen,c(icol,colSums(is.na(df_in[,icol]))))}

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

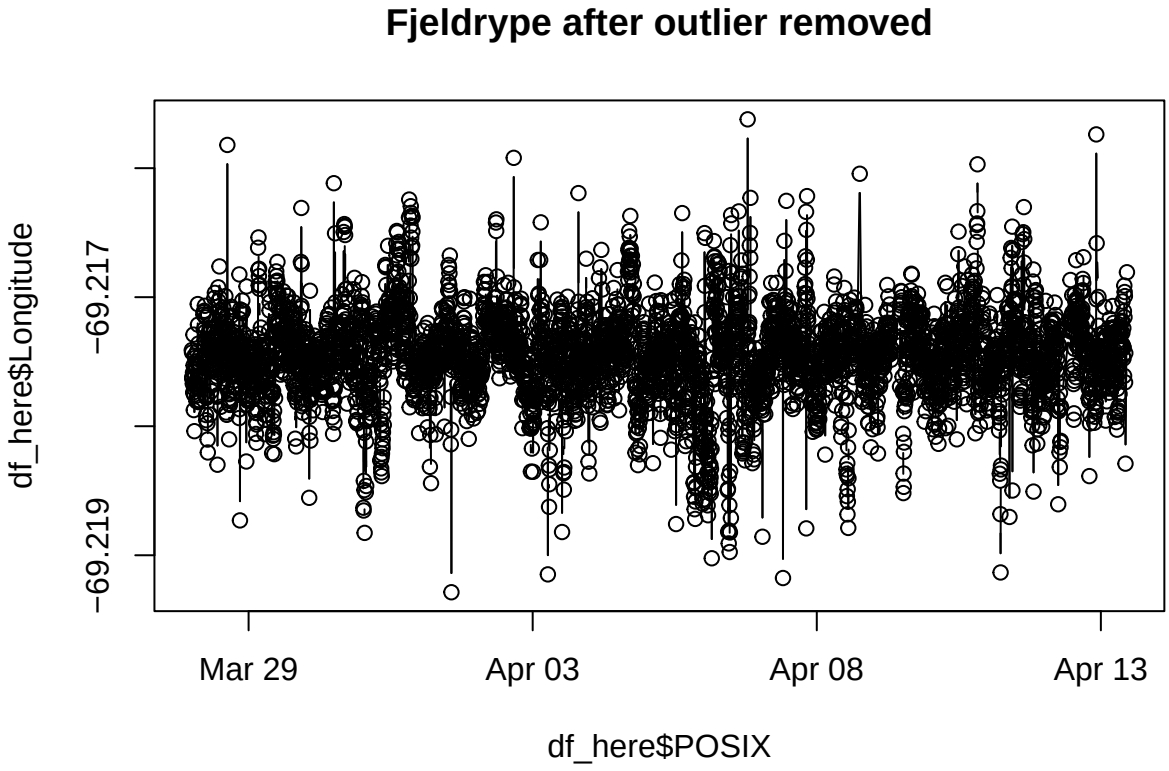
need to clean up Fjeldrype as it is the reference

```
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
df_here$Longitude
−69.205
−69.215
−69.225
Mar 29
Apr 03
Apr 08
Apr 13
df_here$POSIX


Fjeldrype after outlier removed
df_here$Longitude
−69.217
−69.219
Mar 29
Apr 03
Apr 08
Apr 13
df_here$POSIX

```
dlon <- sd(df_here$Longitude,na.rm=TRUE)
dtor <- pi/180
dx_lon <- dlon/360*2*pi*cos(mean(df_here$Latitude)*dtor)*6371*1000
print(paste0("SD of longitude for stationary Fjeldrype : ",round(dlon,6)," degrees lon, or ",round(dx_l
```

```
## [1] "SD of longitude for stationary Fjeldrype : 0.000348 degrees lon, or 8.4 m."
```

```
dlat <- sd(df_here$Latitude,na.rm=TRUE)
dx_lat <- dlat*2*pi*6371*1000/360
print(paste("SD of latitude for stationary Fjeldrype : ",round(dlat,6)," degrees lat, or ",round(dx_lat
```

```
## [1] "SD of latitude for stationary Fjeldrype :  0.00012  degrees lat, or  13.4  m."
```

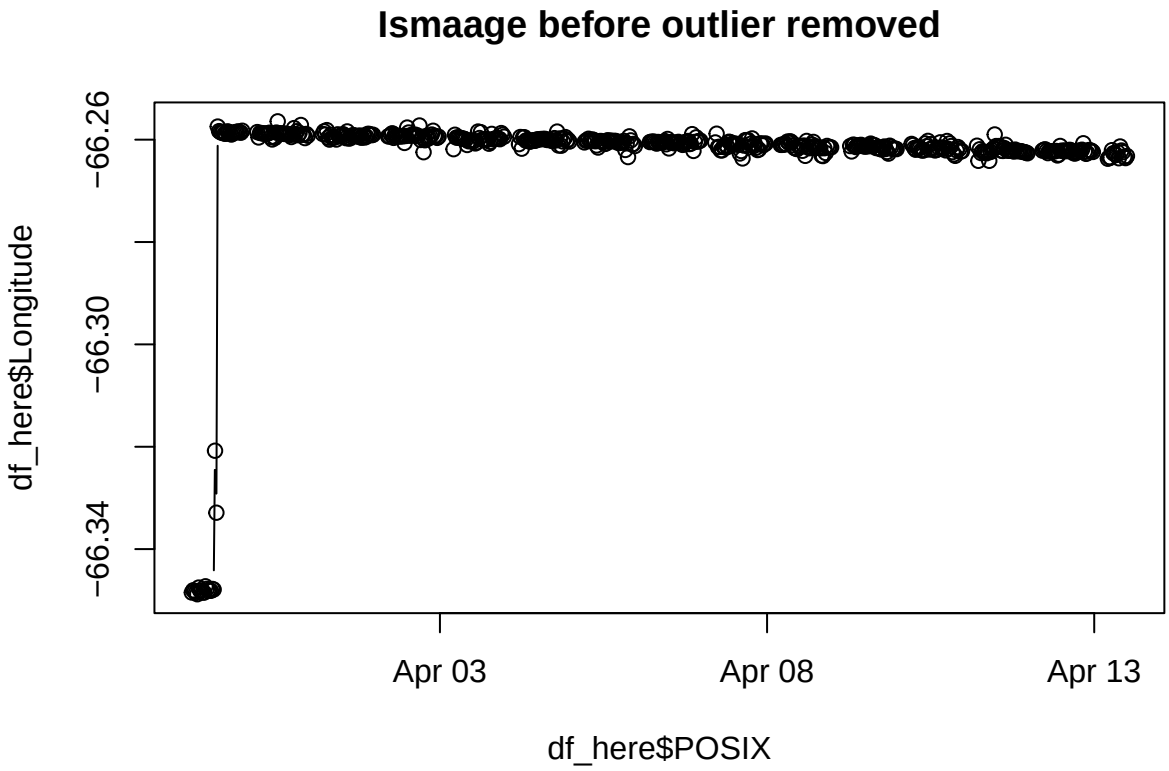
```
# rbind the new
df_org <- rbind.data.frame(df_org,df_here)
df <- df_org
```

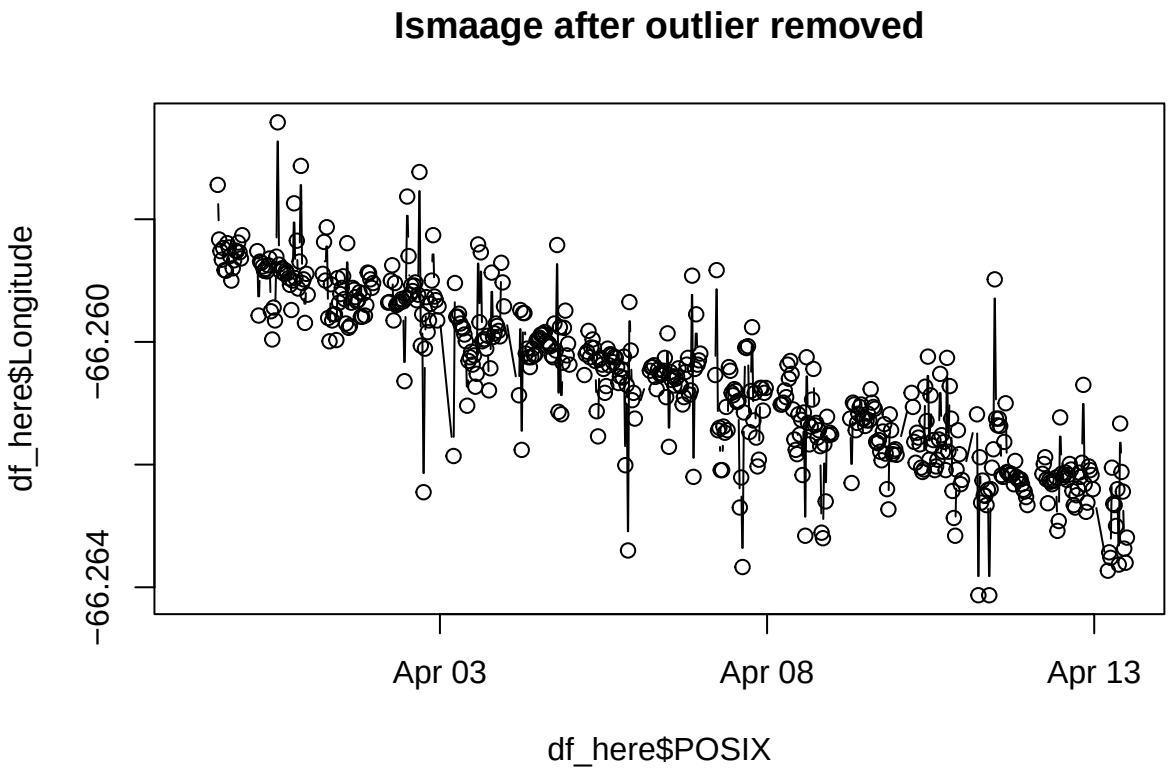
Also clean up Ismaage

```
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
ldx <- which(is.na(df_here$Longitude))
if (length(ldx) != 0) {df_here <- df_here[-ldx,]}
plot(df_here$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
df_here$Longitude
−66.26
−66.30
−66.34
Apr 03
Apr 08
Apr 13
df_here$POSIX


Ismaage after outlier removed
df_here$Longitude
−66.260
−66.264
Apr 03
Apr 08
Apr 13
df_here$POSIX

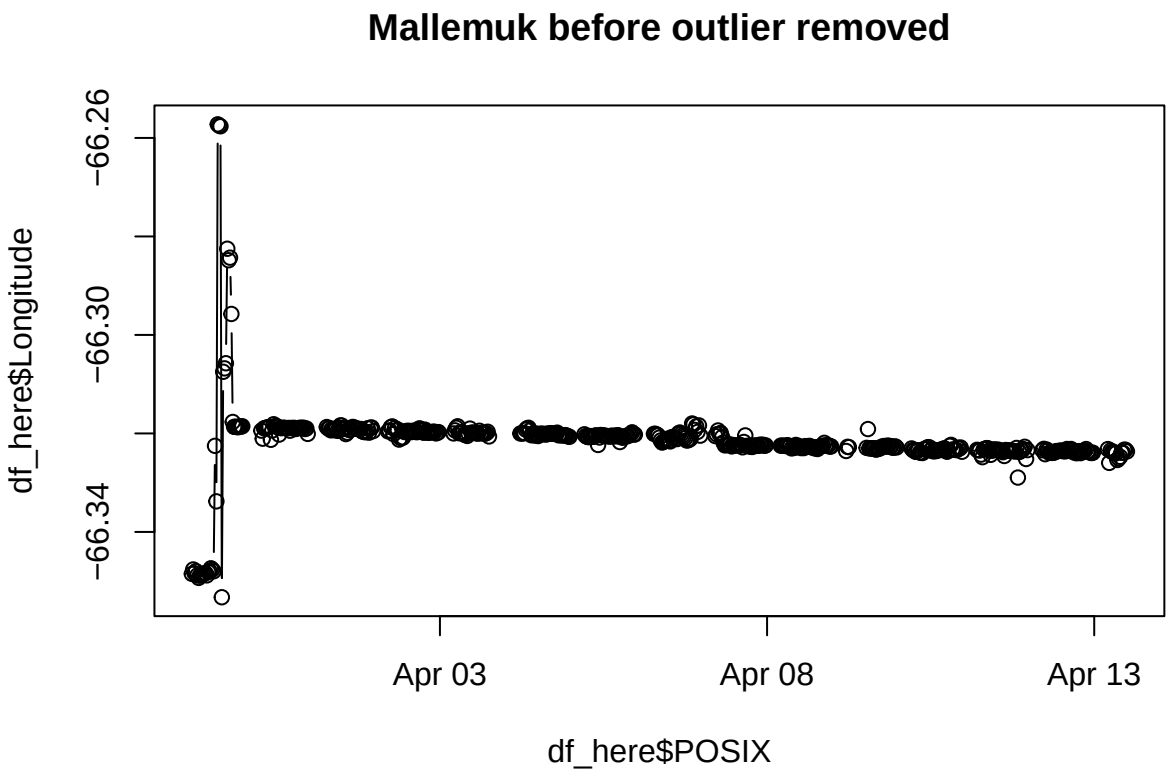
```
# remove the old
df_org <- df_org[-idx,]
# rbind the new
df_org <- rbind.data.frame(df_org,df_here)
df <- df_org
```

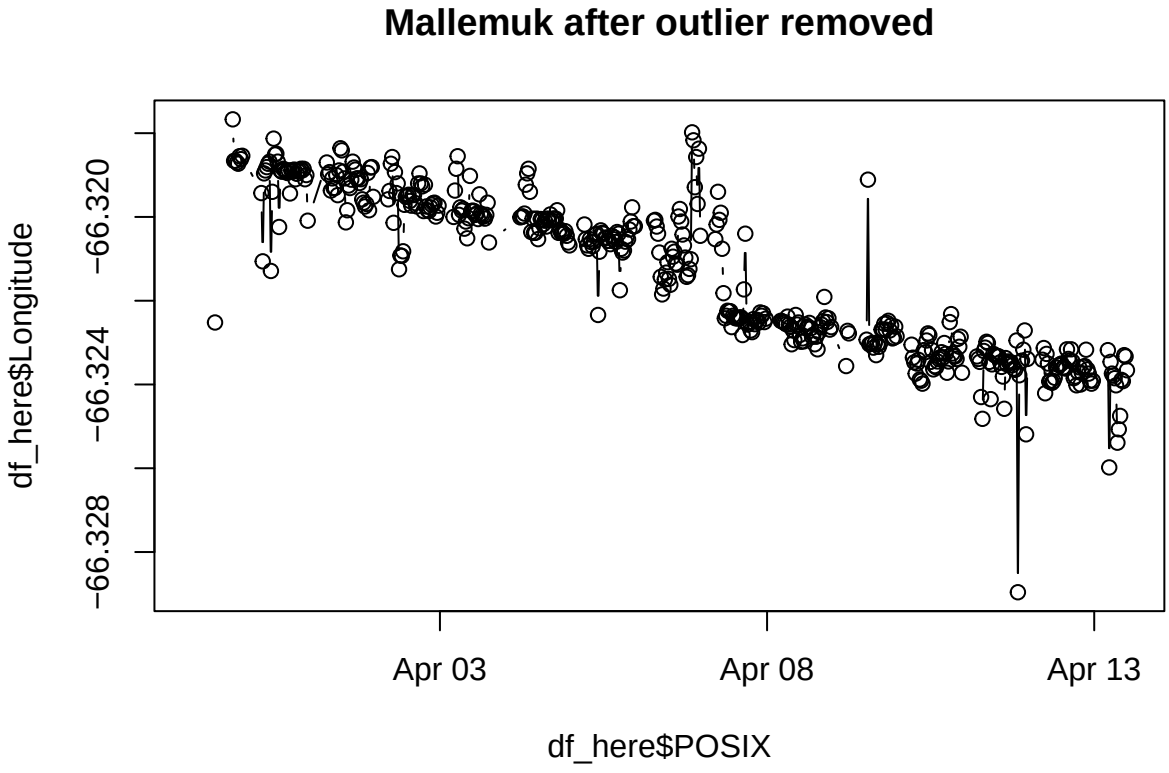
Also clean up Mallemuk lon

```
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
ldx <- which(is.na(df_here$Longitude))
if (length(ldx) != 0) {df_here <- df_here[-ldx,]}
plot(df_here$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
df_here$Longitude
−66.26
−66.30
−66.34
Apr 03
Apr 08
Apr 13
df_here$POSIX


Mallemuk after outlier removed
df_here$Longitude
−66.320
−66.324
−66.328
Apr 03
Apr 08
Apr 13
df_here$POSIX

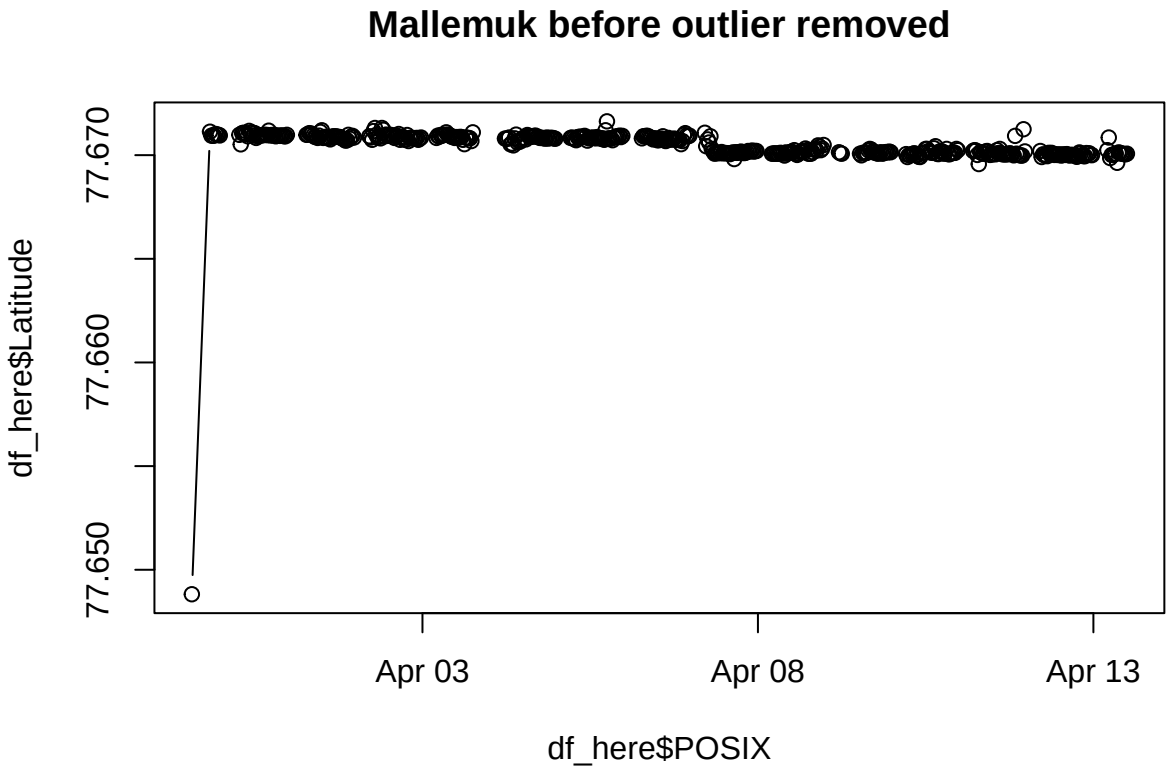
```
# remove the old
df_org <- df_org[-idx,]
# rbind the new
df_org <- rbind.data.frame(df_org,df_here)
df <- df_org
```

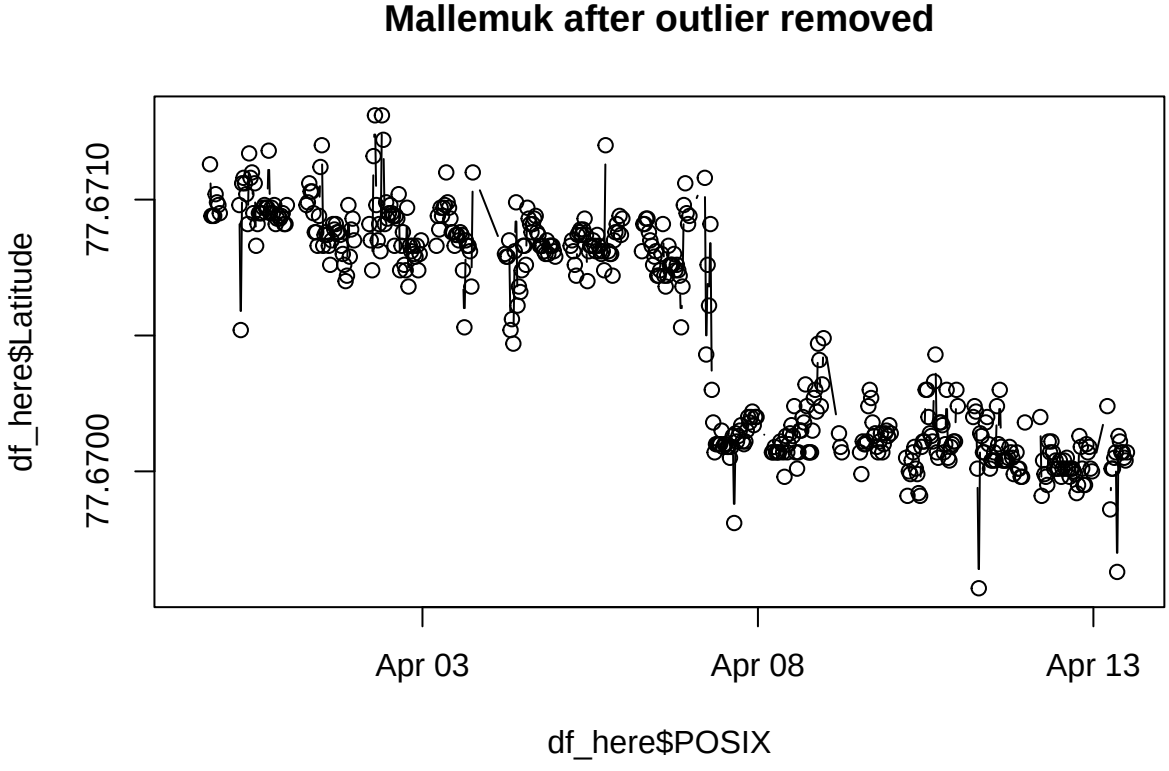
Also clean up Mallemuk lat

```
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
ldx <- which(is.na(df_here$Longitude))
if (length(ldx) != 0) {df_here <- df_here[-ldx,]}
plot(df_here$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
df_here$Latitude
77.650
77.660
77.670
Apr 03
Apr 08
Apr 13
df_here$POSIX


Mallemuk after outlier removed
df_here$Latitude
77.6700
77.6710
Apr 03
Apr 08
Apr 13
df_here$POSIX

```
# 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))
}
```

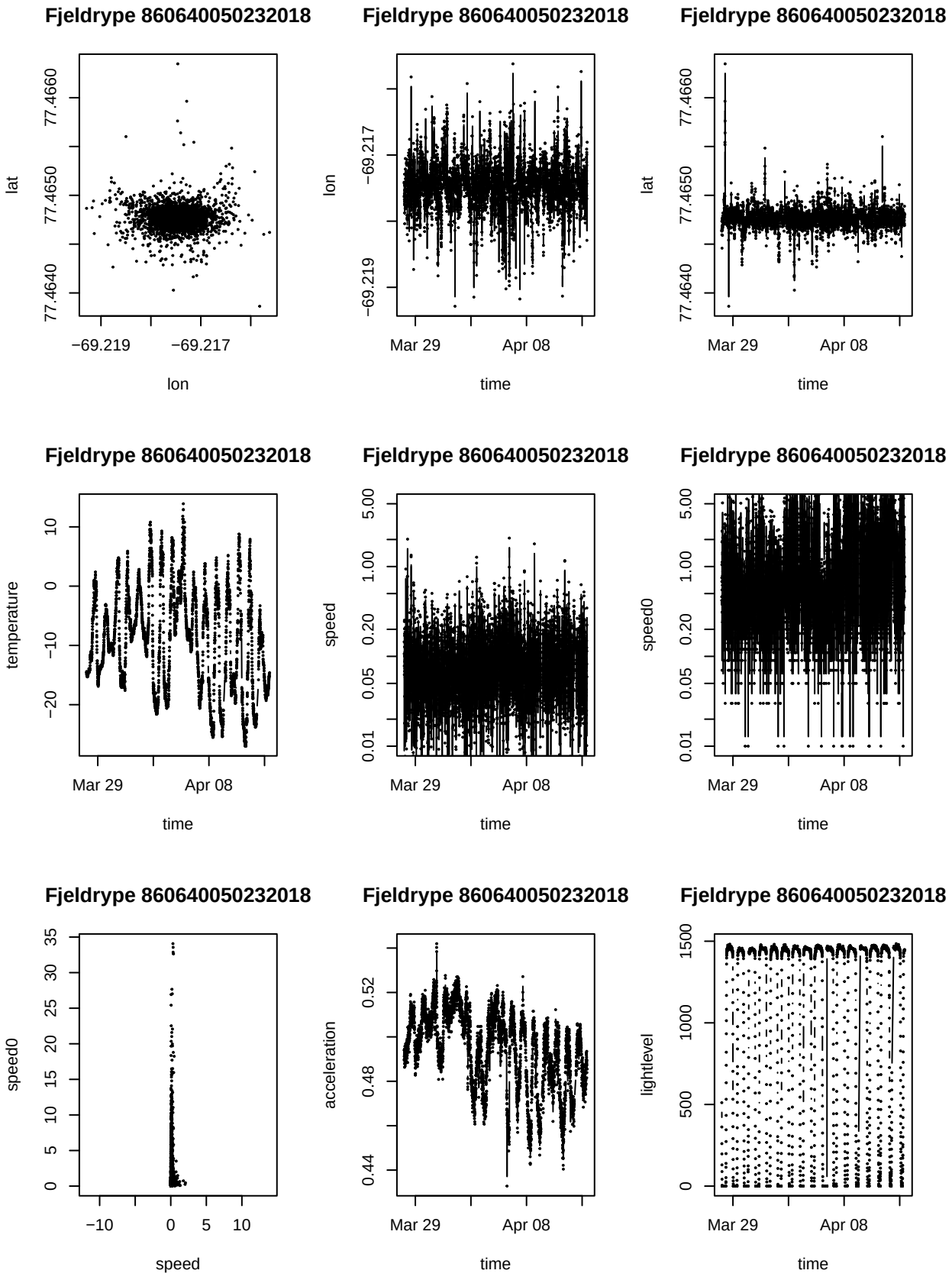
read

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

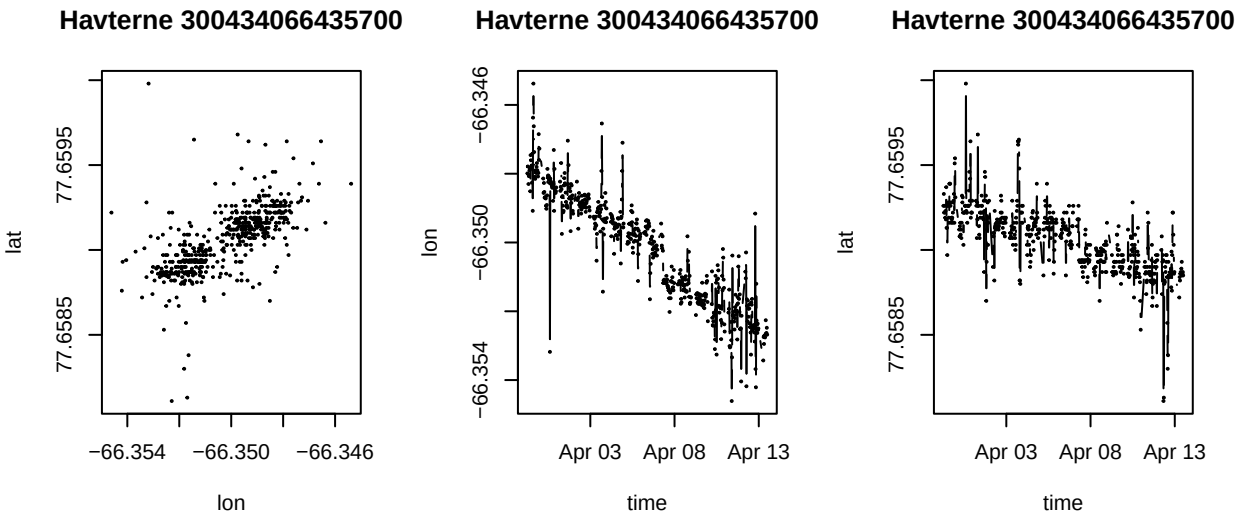
```
  idx <- which(df_org$UnitName == istat & df_org$Longitude < -65)
  df2 <- df_org[idx,]
  df2 <- kleaner(df2)
  cnams <- colnames(df2)
  time <- as.POSIXct(df2$"Timestamp UTC",tz="UTC")
  #
  lon <- df2$Longitude
  lat <- df2$Latitude
  temperature <- df2$"Temperature(°C)"
  acceleration <- sqrt(df2$"AccelerationX(g)" ^2+df2$"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(speed0 > 3)) {plot(time,speed0,main=istat,pch=19,cex=0.2,type="b",ylim=c(0.01,5),log="y")}
  if (length(speed0 > 3) & length(speed0) > 3) {plot(speed,speed0,main=istat,pch=19,cex=0.2,type="p",as
  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)  }
  if (length(speed0) == nrow(df3)){    df3 <- cbind.data.frame(df3,speed0)  }
  saveRDS(df3,paste0('OUTPUT/',istat,'.rds'))
  #

}
```

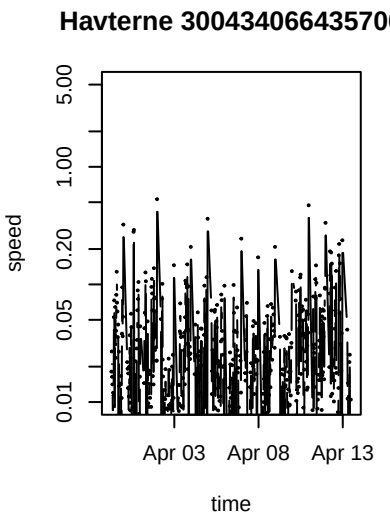
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 2 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialised column: ‘Temperature(°C)‘.

## Warning: Unknown or uninitialised column: ‘AccelerationX(g)‘.

## Warning: Unknown or uninitialised column: ‘AccelerationY(g)‘.

## Warning: Unknown or uninitialised column: ‘AccelerationZ(g)‘.

## Warning: Unknown or uninitialised column: ‘LightLevel‘.
```

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

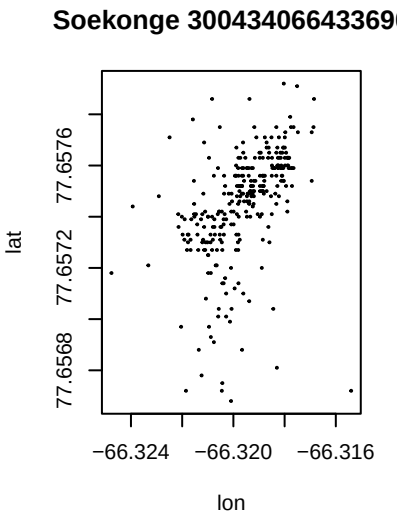

Fjeldrype 860640050232018
lat
77.4640
77.4650
77.4660
−69.219
−69.217
lon
Fjeldrype 860640050232018
lon
−69.219
−69.217
Mar 29
Apr 08
time
Fjeldrype 860640050232018
lat
77.4640
77.4650
77.4660
Mar 29
Apr 08
time
Fjeldrype 860640050232018
temperature
−20
−10
0
10
Mar 29
Apr 08
time
Fjeldrype 860640050232018
speed
0.01
0.05
0.20
1.00
5.00
Mar 29
Apr 08
time
Fjeldrype 860640050232018
speed0
0.01
0.05
0.20
1.00
5.00
Mar 29
Apr 08
time
Fjeldrype 860640050232018
speed0
0
5
10
15
20
25
30
35
−10
0
5
10
speed
Fjeldrype 860640050232018
acceleration
0.44
0.48
0.52
Mar 29
Apr 08
time
Fjeldrype 860640050232018
lightlevel
0
500
1000
1500
Mar 29
Apr 08
time

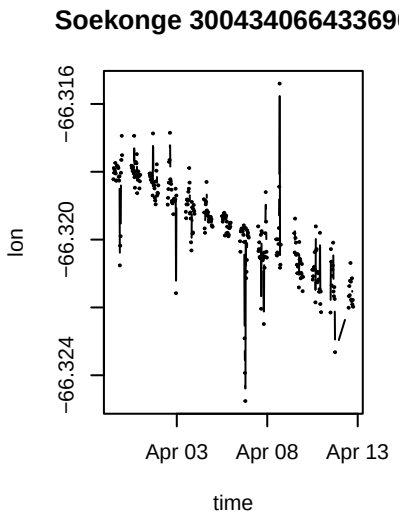
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 6 y values <= 0 omitted from
## logarithmic plot

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

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

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

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

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

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

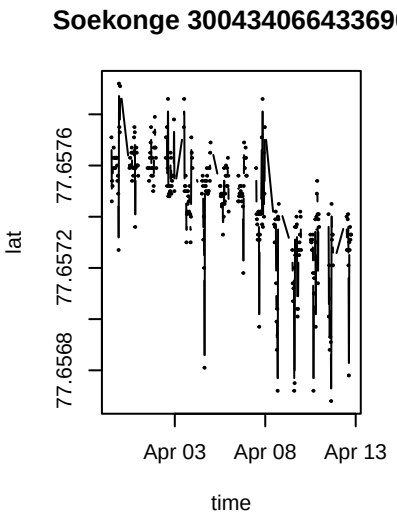

Havterne 300434066435700
lat
77.6595
77.6585
−66.354
−66.350
−66.346
lon
Havterne 300434066435700
lon
−66.346
−66.350
−66.354
Apr 03
Apr 08
Apr 13
time
Havterne 300434066435700
lat
77.6595
77.6585
Apr 03
Apr 08
Apr 13
time

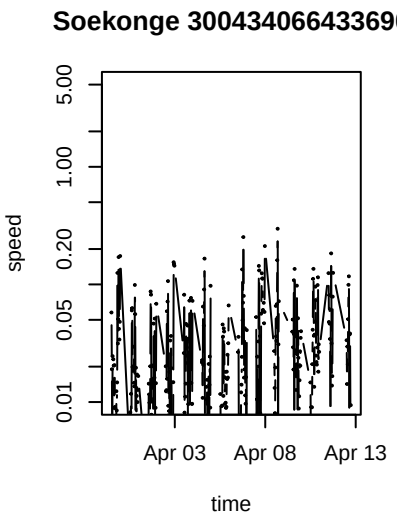

Havterne 300434066435700
speed
5.00
1.00
0.20
0.05
0.01
Apr 03
Apr 08
Apr 13
time

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

## Warning: Unknown or uninitialised column: `Temperature(°C)`.

## Warning: Unknown or uninitialised column: `AccelerationX(g)`.

## Warning: Unknown or uninitialised column: `AccelerationY(g)`.

## Warning: Unknown or uninitialised column: `AccelerationZ(g)`.

## Warning: Unknown or uninitialised column: `LightLevel`.

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

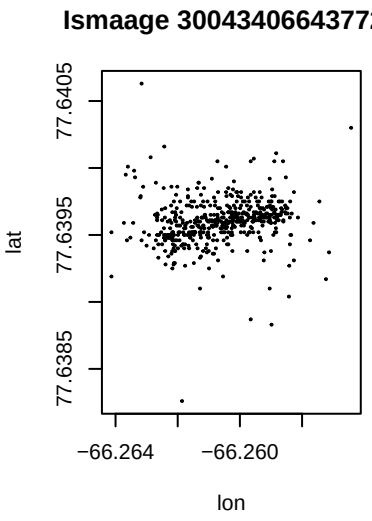

Soekonge 300434066433690
lat
77.6568
77.6572
77.6576
−66.324
−66.320
−66.316
lon

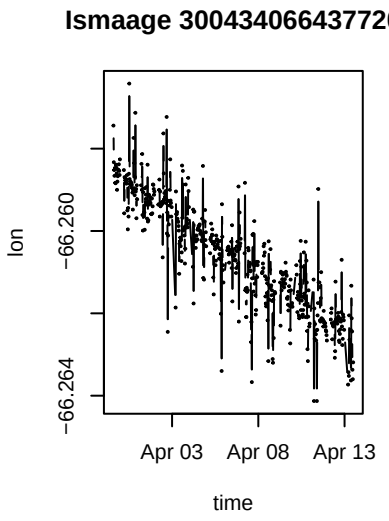

Soekonge 300434066433690
lon
−66.324
−66.320
−66.316
Apr 03
Apr 08
Apr 13
time

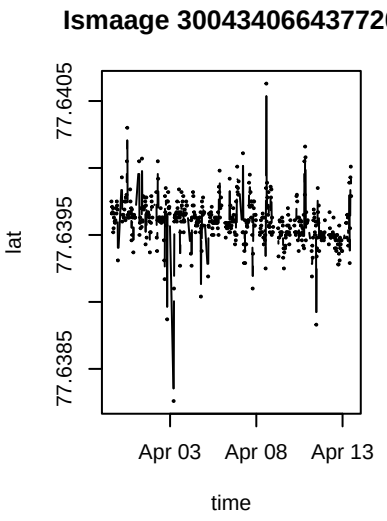

Soekonge 300434066433690
lat
77.6568
77.6572
77.6576
Apr 03
Apr 08
Apr 13
time

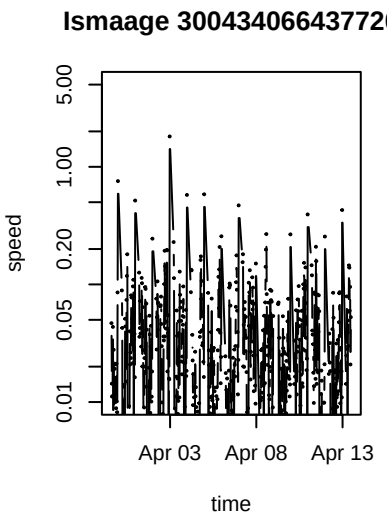

Soekonge 300434066433690
speed
0.01
0.05
0.20
1.00
5.00
Apr 03
Apr 08
Apr 13
time

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

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

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

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

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

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

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

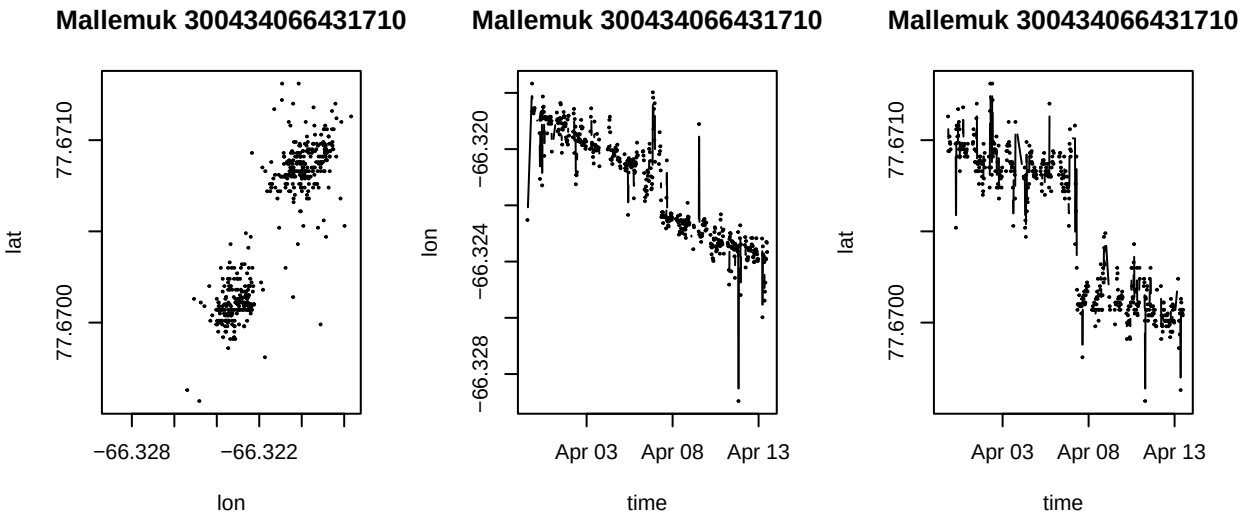

Ismaage 300434066437720
lat
77.6405
77.6395
77.6385
−66.264
−66.260
lon

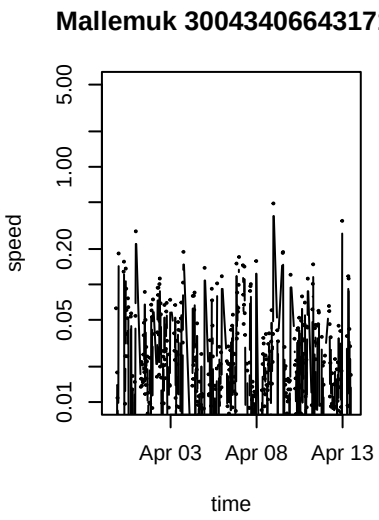

Ismaage 300434066437720
lon
−66.260
−66.264
Apr 03
Apr 08
Apr 13
time


Ismaage 300434066437720
lat
77.6405
77.6395
77.6385
Apr 03
Apr 08
Apr 13
time

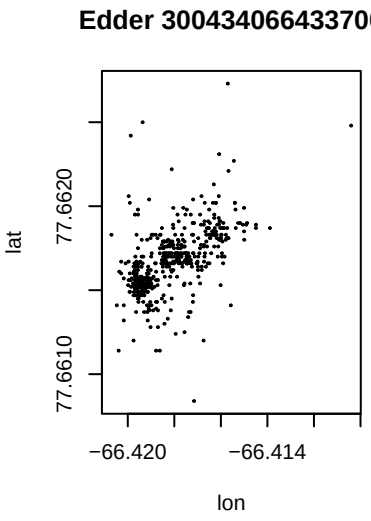

Ismaage 300434066437720
speed
5.00
1.00
0.20
0.05
0.01
Apr 03
Apr 08
Apr 13
time

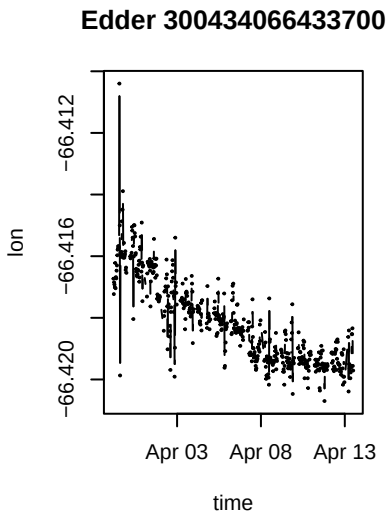
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 6 y values <= 0 omitted from
## logarithmic plot

## Warning: Unknown or uninitialised column: `Temperature(°C)`.

## Warning: Unknown or uninitialised column: `AccelerationX(g)`.

## Warning: Unknown or uninitialised column: `AccelerationY(g)`.

## Warning: Unknown or uninitialised column: `AccelerationZ(g)`.

## Warning: Unknown or uninitialised column: `LightLevel`.

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

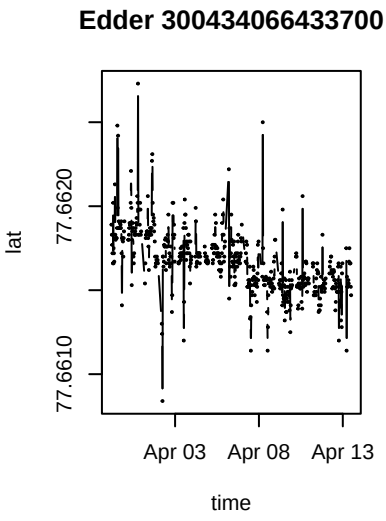

Mallemuk 300434066431710
lat
77.6710
77.6700
−66.328
−66.322
lon
Mallemuk 300434066431710
lon
−66.320
−66.324
−66.328
Apr 03
Apr 08
Apr 13
time
Mallemuk 300434066431710
lat
77.6710
77.6700
Apr 03
Apr 08
Apr 13
time

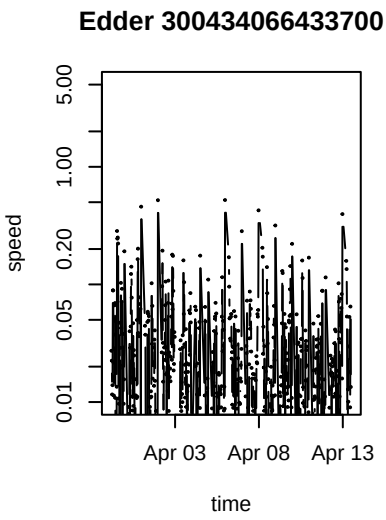

Mallemuk 300434066431710
speed
5.00
1.00
0.20
0.05
0.01
Apr 03
Apr 08
Apr 13
time

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

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

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

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

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

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

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

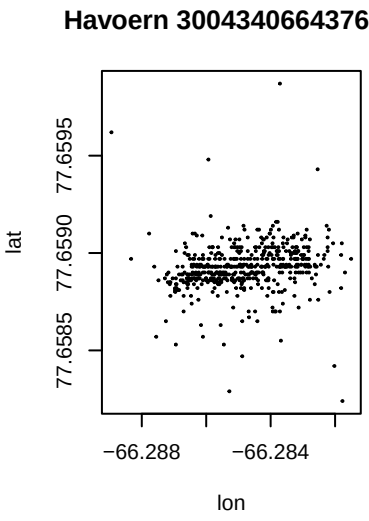

Edder 300434066433700
lat
77.6620
77.6610
−66.420
−66.414
lon

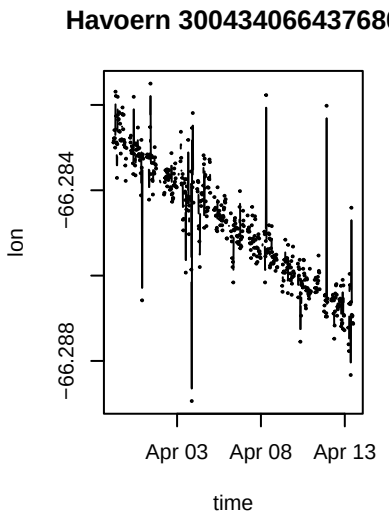

Edder 300434066433700
lon
−66.412
−66.416
−66.420
Apr 03
Apr 08
Apr 13
time

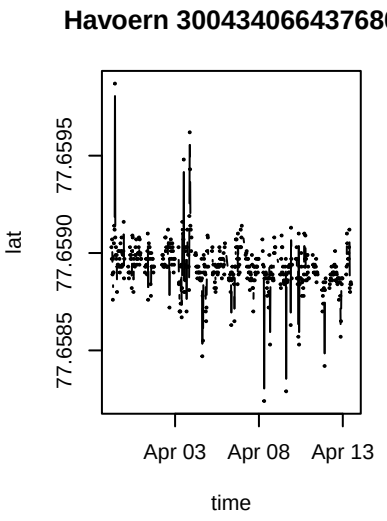

Edder 300434066433700
lat
77.6620
77.6610
Apr 03
Apr 08
Apr 13
time

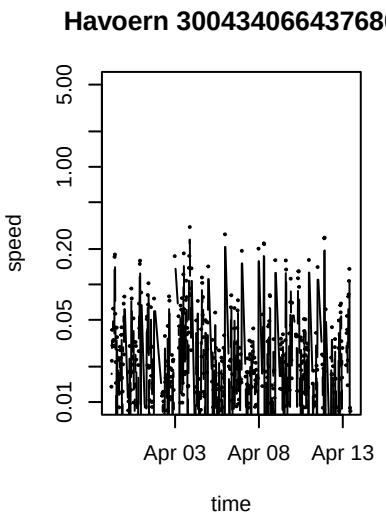

Edder 300434066433700
speed
5.00
1.00
0.20
0.05
0.01
Apr 03
Apr 08
Apr 13
time

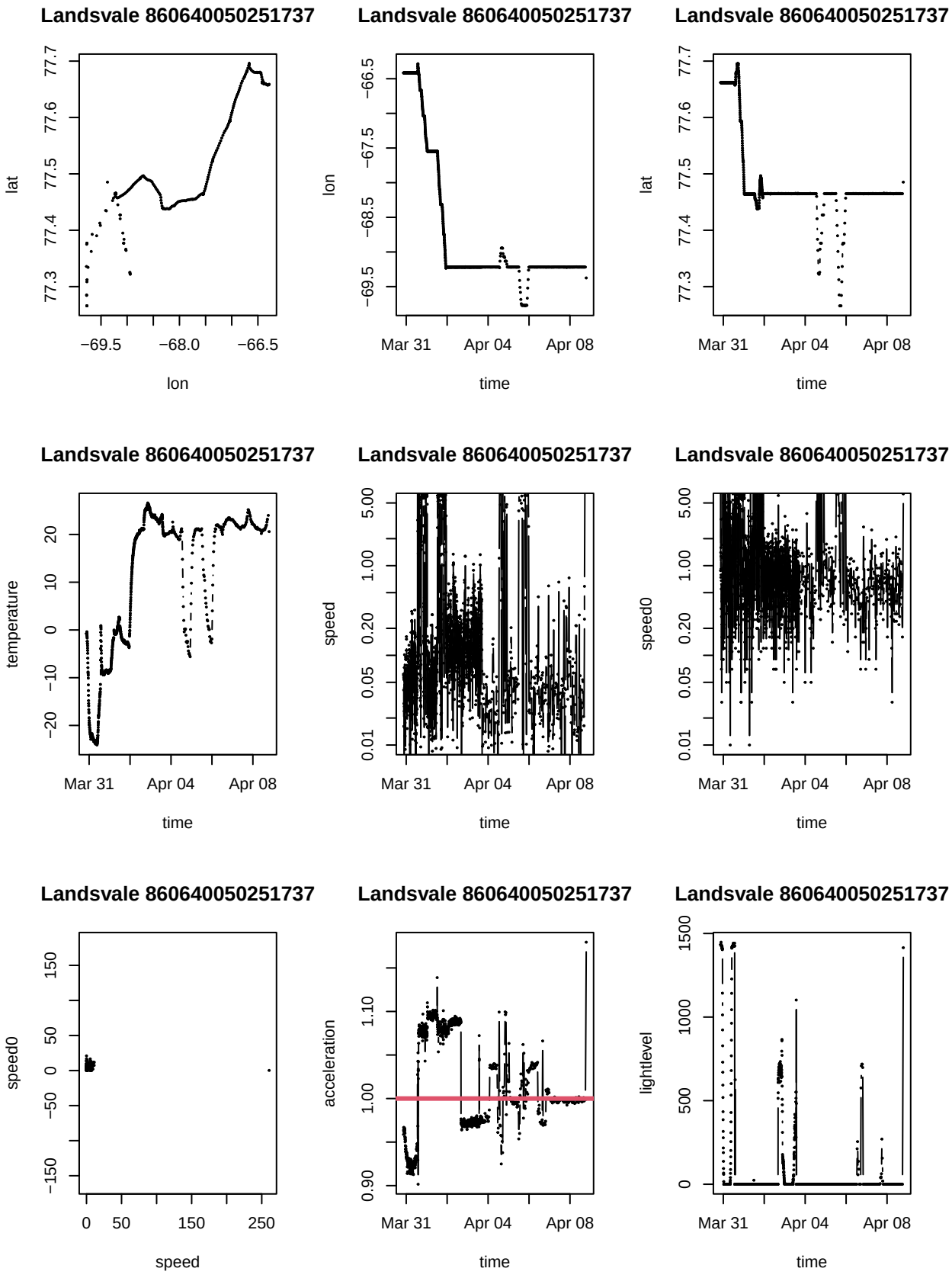
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 4 y values <= 0 omitted from
## logarithmic plot
```


Havoern 300434066437680
lat
77.6585
77.6590
77.6595
−66.288
−66.284
lon

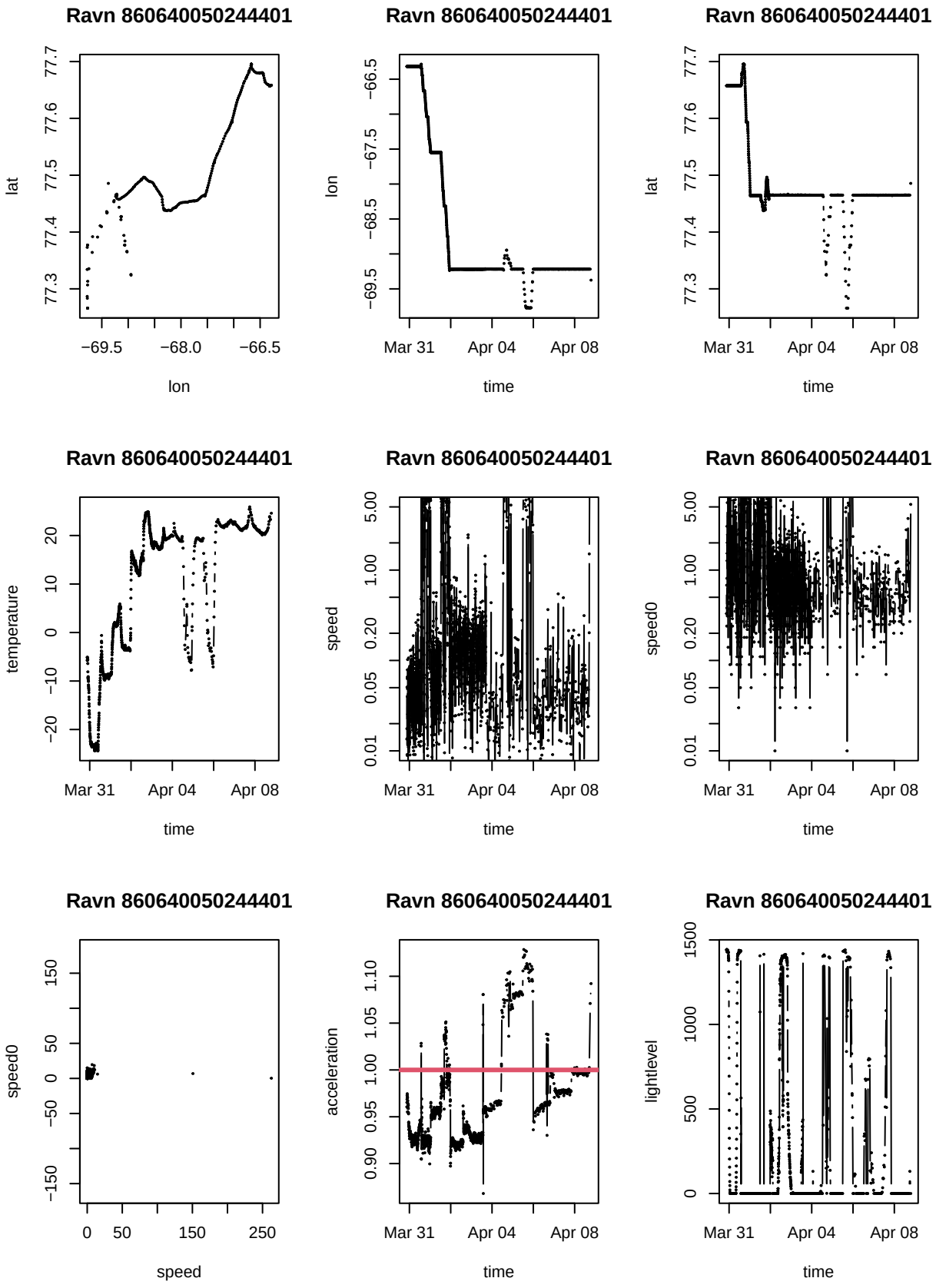

Havoern 300434066437680
lon
−66.284
−66.288
Apr 03
Apr 08
Apr 13
time

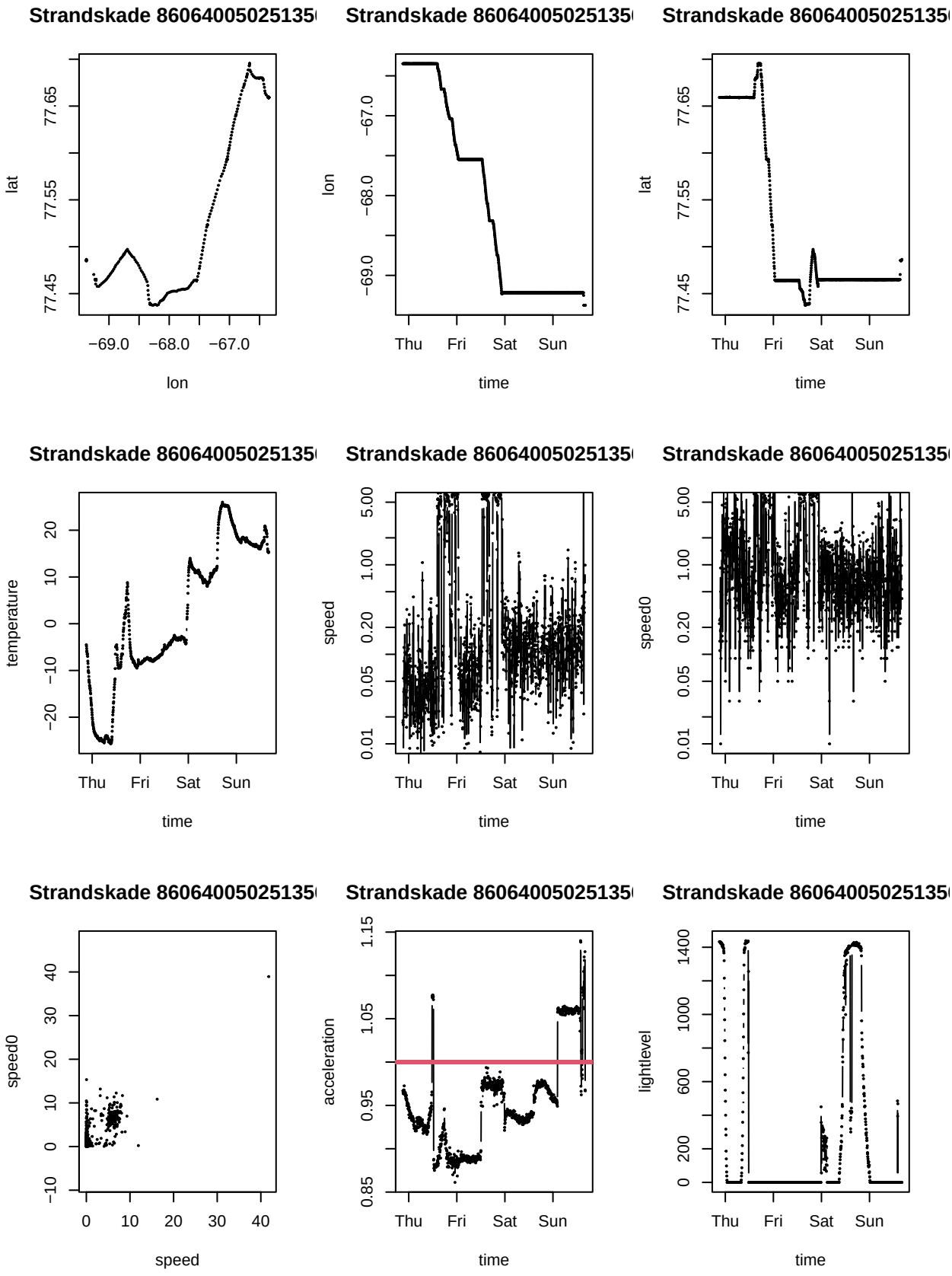

Havoern 300434066437680
lat
77.6585
77.6590
77.6595
Apr 03
Apr 08
Apr 13
time

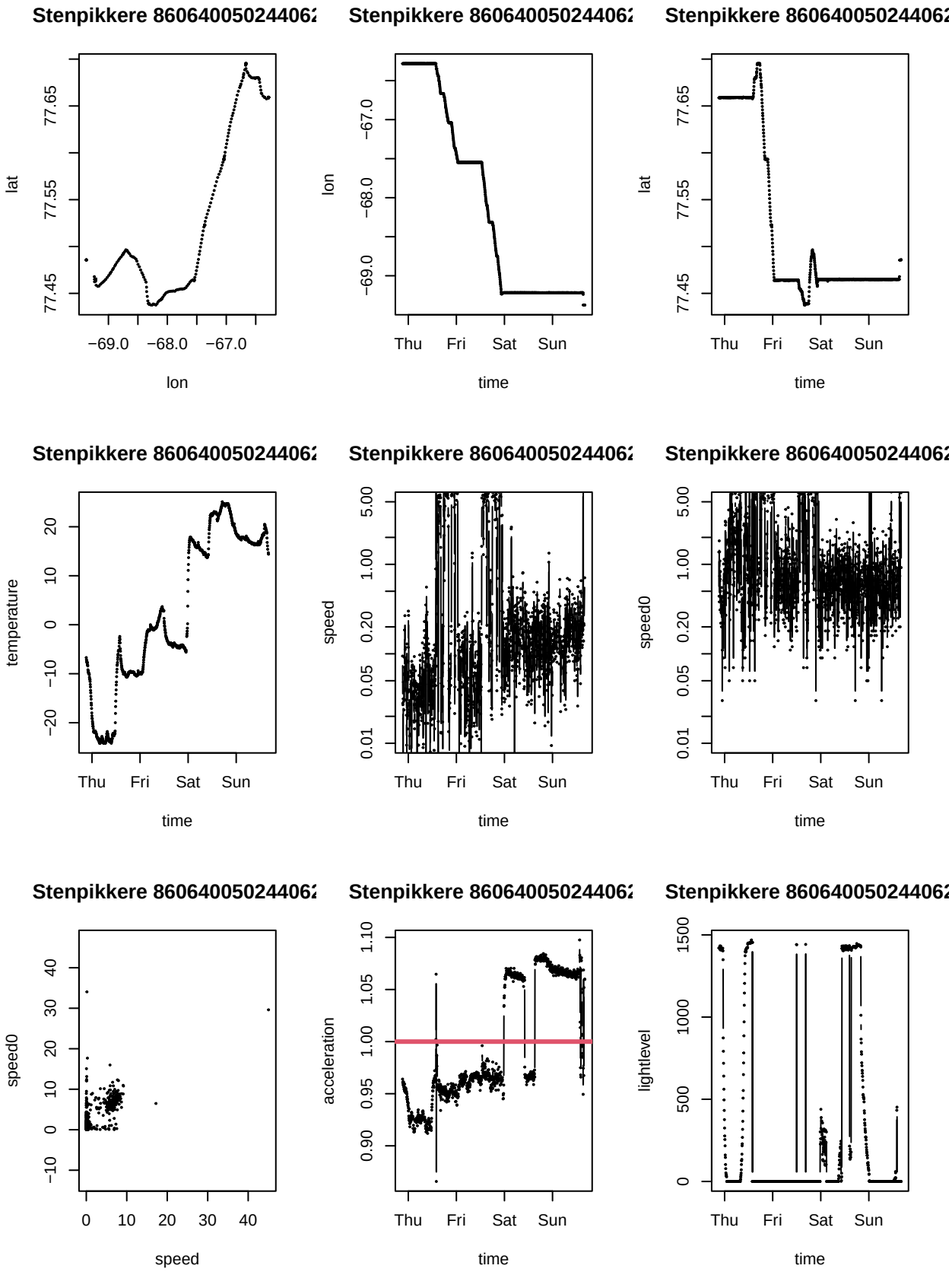

Havoern 300434066437680
speed
0.01
0.05
0.20
1.00
5.00
Apr 03
Apr 08
Apr 13
time


Landsvale 860640050251737
lat
77.3
77.4
77.5
77.6
77.7
−69.5
−68.0
−66.5
lon
Landsvale 860640050251737
lon
−66.5
−67.5
−68.5
−69.5
Mar 31
Apr 04
Apr 08
time
Landsvale 860640050251737
lat
77.3
77.4
77.5
77.6
77.7
Mar 31
Apr 04
Apr 08
time
Landsvale 860640050251737
temperature
−20
−10
0
10
20
Mar 31
Apr 04
Apr 08
time
Landsvale 860640050251737
speed
0.01
0.05
0.20
1.00
5.00
Mar 31
Apr 04
Apr 08
time
Landsvale 860640050251737
speed0
0.01
0.05
0.20
1.00
5.00
Mar 31
Apr 04
Apr 08
time
Landsvale 860640050251737
speed0
−150
−50
0
50
150
0
50
150
250
speed
Landsvale 860640050251737
acceleration
0.90
1.00
1.10
Mar 31
Apr 04
Apr 08
time
Landsvale 860640050251737
lightlevel
0
500
1000
1500
Mar 31
Apr 04
Apr 08
time

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


Ravn 860640050244401
lat
lon
77.3
77.4
77.5
77.6
77.7
−69.5
−68.0
−66.5
Ravn 860640050244401
lon
time
−66.5
−67.5
−68.5
−69.5
Mar 31
Apr 04
Apr 08
Ravn 860640050244401
lat
time
Ravn 860640050244401
temperature
time
−20
−10
0
10
20
Ravn 860640050244401
speed
time
0.01
0.05
0.20
1.00
5.00
Ravn 860640050244401
speed0
time
Ravn 860640050244401
speed0
speed
−150
−50
0
50
150
0
50
150
250
Ravn 860640050244401
acceleration
time
0.90
0.95
1.00
1.05
1.10
Ravn 860640050244401
lightlevel
time
0
500
1000
1500


Strandskade 86064005025135
lat
lon
77.45
77.55
77.65
−69.0
−68.0
−67.0
Strandskade 86064005025135
lon
time
Thu
Fri
Sat
Sun
Strandskade 86064005025135
lat
time
Strandskade 86064005025135
temperature
time
−20
−10
0
10
20
Strandskade 86064005025135
speed
time
0.01
0.05
0.20
1.00
5.00
Strandskade 86064005025135
speed0
time
Strandskade 86064005025135
speed0
speed
0
10
20
30
40
Strandskade 86064005025135
acceleration
time
0.85
0.95
1.05
1.15
Strandskade 86064005025135
lightlevel
time
0
200
600
1000
1400


Stenpikkere 860640050244062
lat
77.45
77.55
77.65
−69.0
−68.0
−67.0
lon
Stenpikkere 860640050244062
lon
−67.0
−68.0
−69.0
Thu
Fri
Sat
Sun
time
Stenpikkere 860640050244062
lat
77.45
77.55
77.65
Thu
Fri
Sat
Sun
time
Stenpikkere 860640050244062
temperature
−20
−10
0
10
20
Thu
Fri
Sat
Sun
time
Stenpikkere 860640050244062
speed
0.01
0.05
0.20
1.00
5.00
Thu
Fri
Sat
Sun
time
Stenpikkere 860640050244062
speed0
0.01
0.05
0.20
1.00
5.00
Thu
Fri
Sat
Sun
time
Stenpikkere 860640050244062
speed0
−10
0
10
20
30
40
0
10
20
30
40
speed
Stenpikkere 860640050244062
acceleration
0.90
0.95
1.00
1.05
1.10
Thu
Fri
Sat
Sun
time
Stenpikkere 860640050244062
lightlevel
0
500
1000
1500
Thu
Fri
Sat
Sun
time

Remove dogsled series

```
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
}
```

```
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$speed, type="b", pch=ic+14,cex=0.8, xlab="Date/Time", ylab="speed [km
                       xlim=c(as.POSIXct("2022-03-28 00:00:00",tz="UTC"), as.POSIXct("2022-03-31 23:59:5
  else{   lines(df$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
```

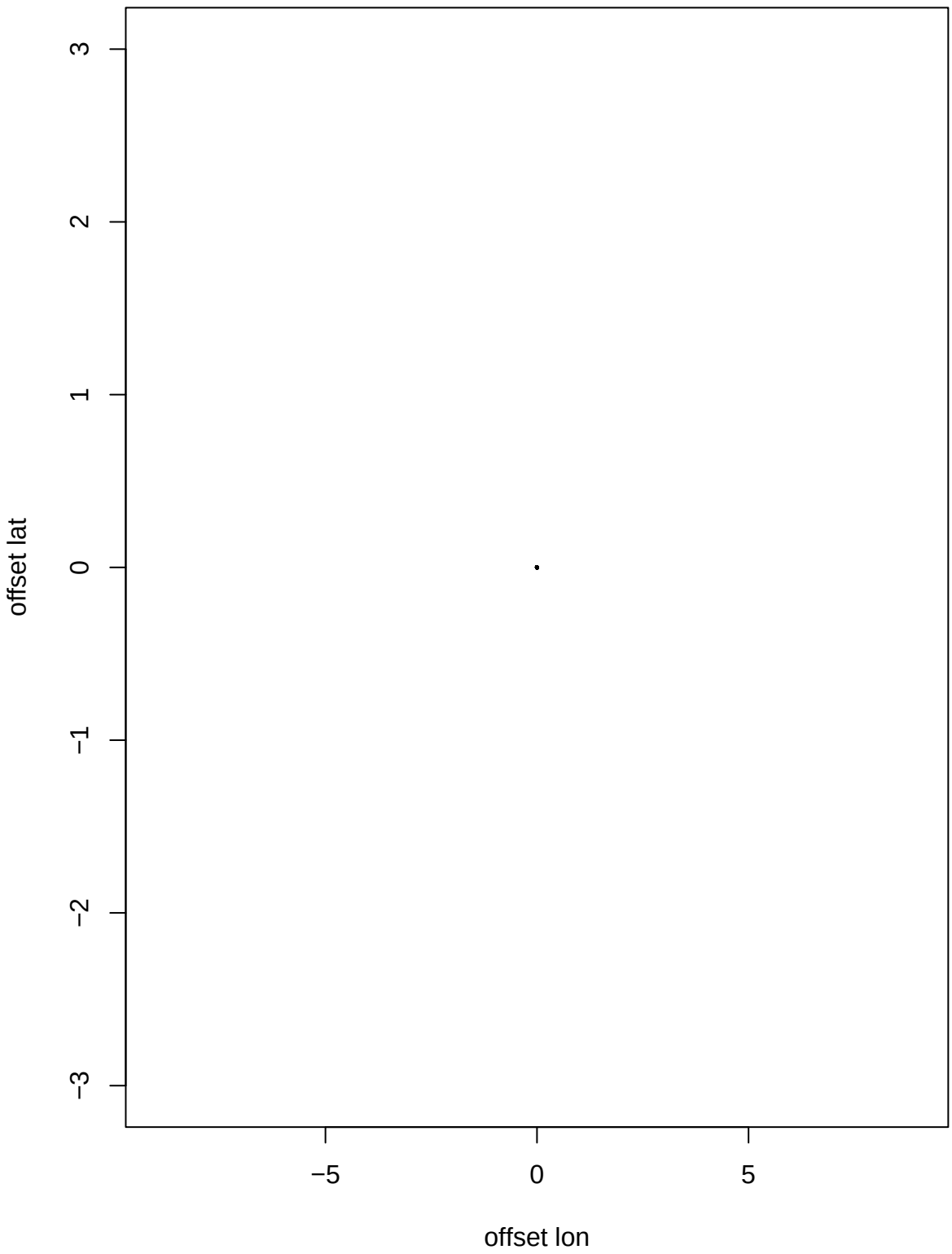
positions relative to Fjeldrype

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

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

alldf <- NULL
#
for (jstat in 1:length(unique_names))
{
```

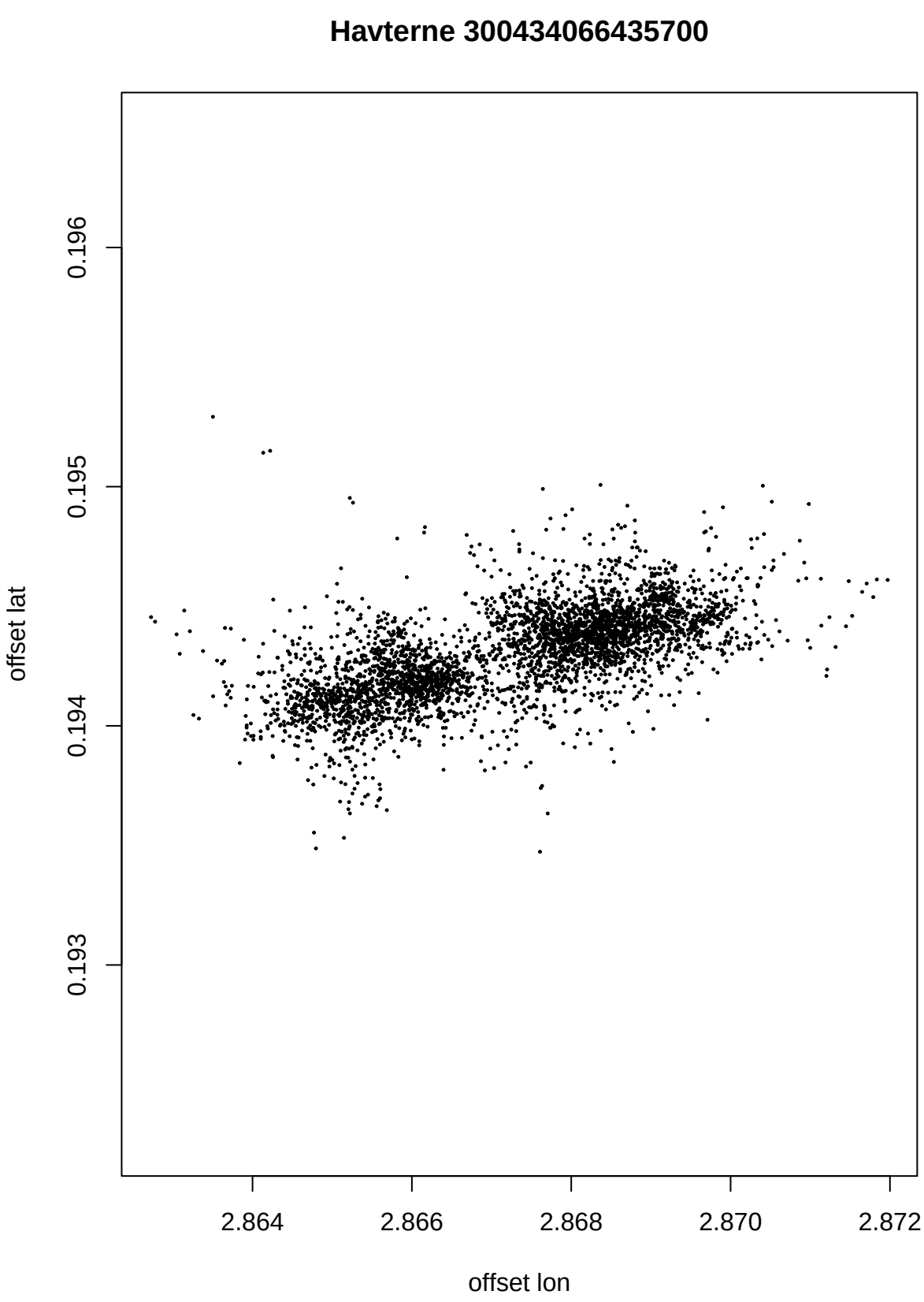
```
  #print(jstat)
  statname <- unique_names[jstat]
  #print(statname)
  png(paste0("FIGURES/offset_lon_lat_rel_Fjeldrype_",statname,".png"))

  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$delta_lat,main=statname,xlab="offset lon",ylab="offset lat",pch=19,c
  #
  dev.off()
  plot(together$delta_lon,together$delta_lat,main=statname,xlab="offset lon",ylab="offset lat",pch=19,c
}
```

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

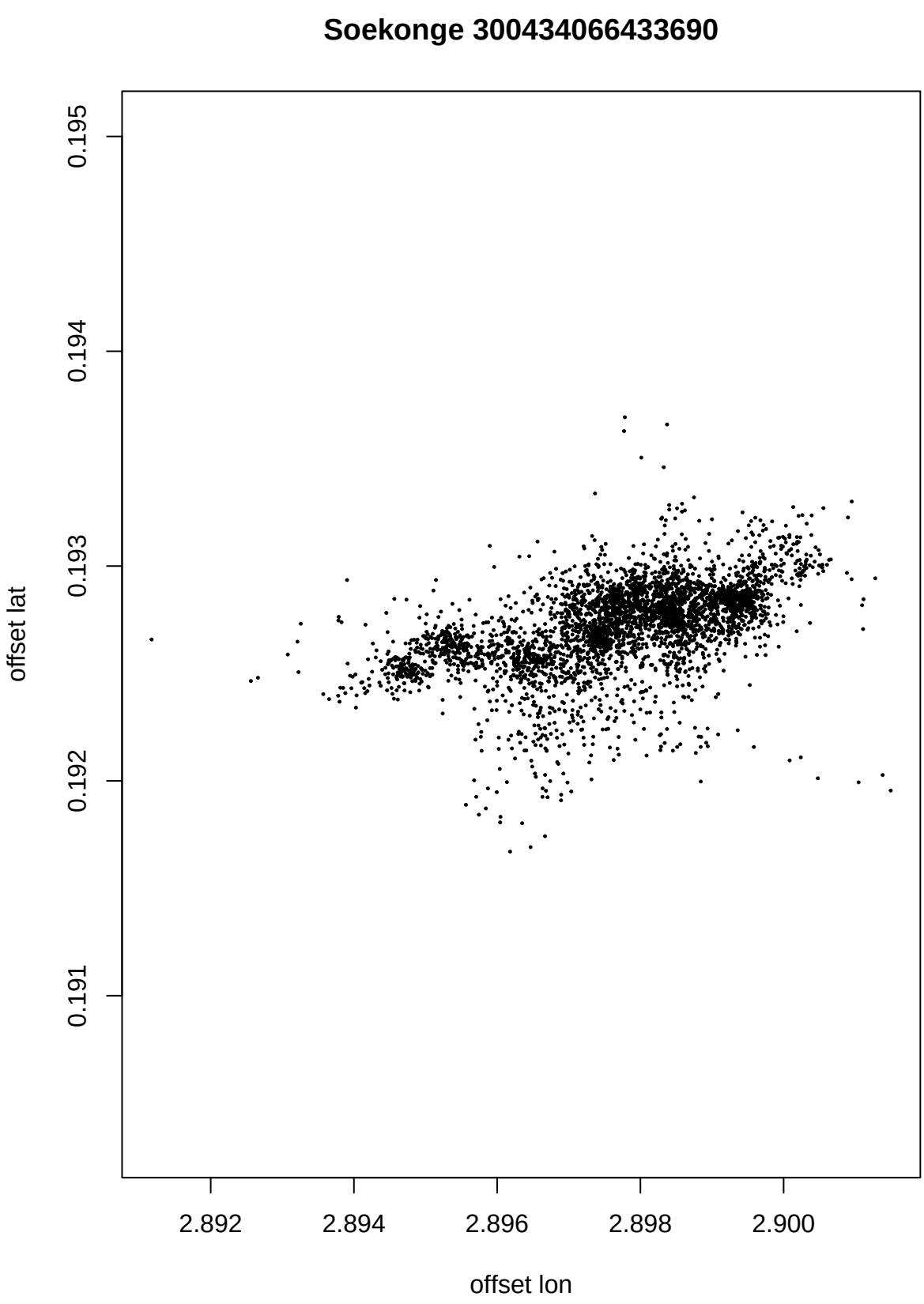
```
## [1] "Havterne 300434066435700 0.0016 2e-04"
```


Fjeldrype 860640050232018
offset lat
3
2
1
0
−1
−2
−3
−5
0
5
offset lon

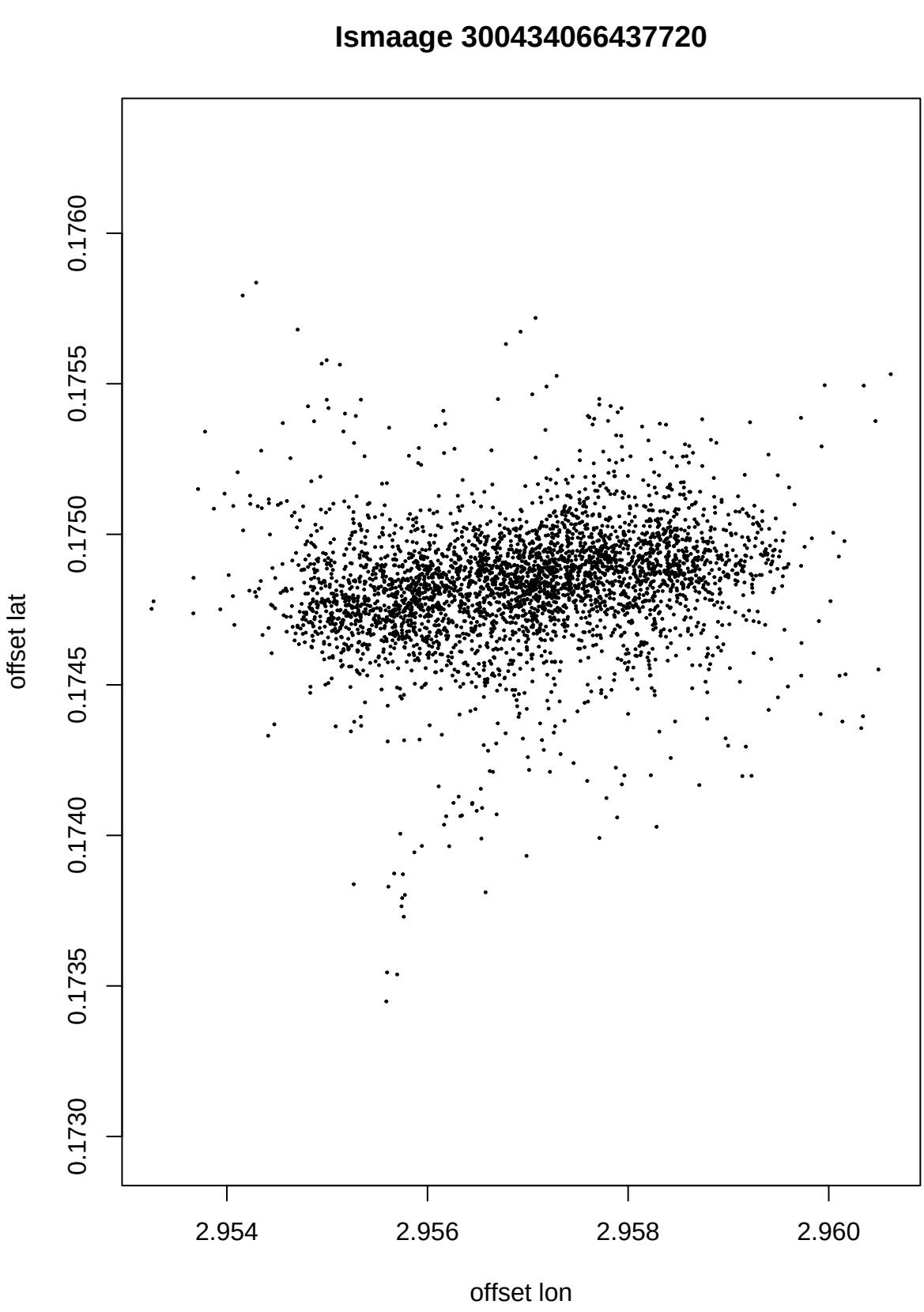
```
## [1] "Soekonge 300434066433690 0.0014 2e-04"
```


Havterne 300434066435700
offset lat
0.196
0.195
0.194
0.193
2.864
2.866
2.868
2.870
2.872
offset lon

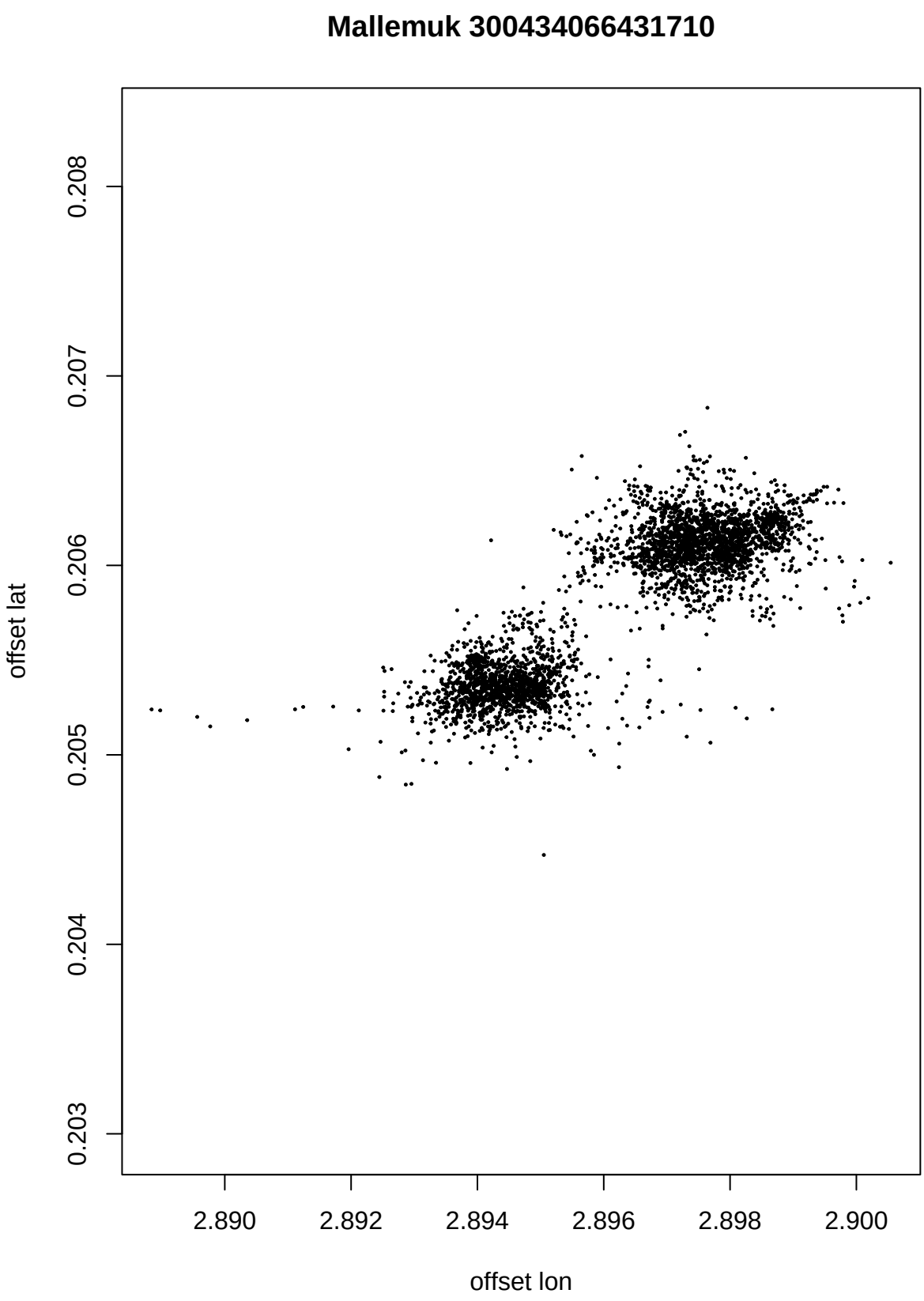
```
## [1] "Ismaage 300434066437720 0.0012 2e-04"
```


Soekonge 300434066433690
offset lat
0.195
0.194
0.193
0.192
0.191
2.892
2.894
2.896
2.898
2.900
offset lon

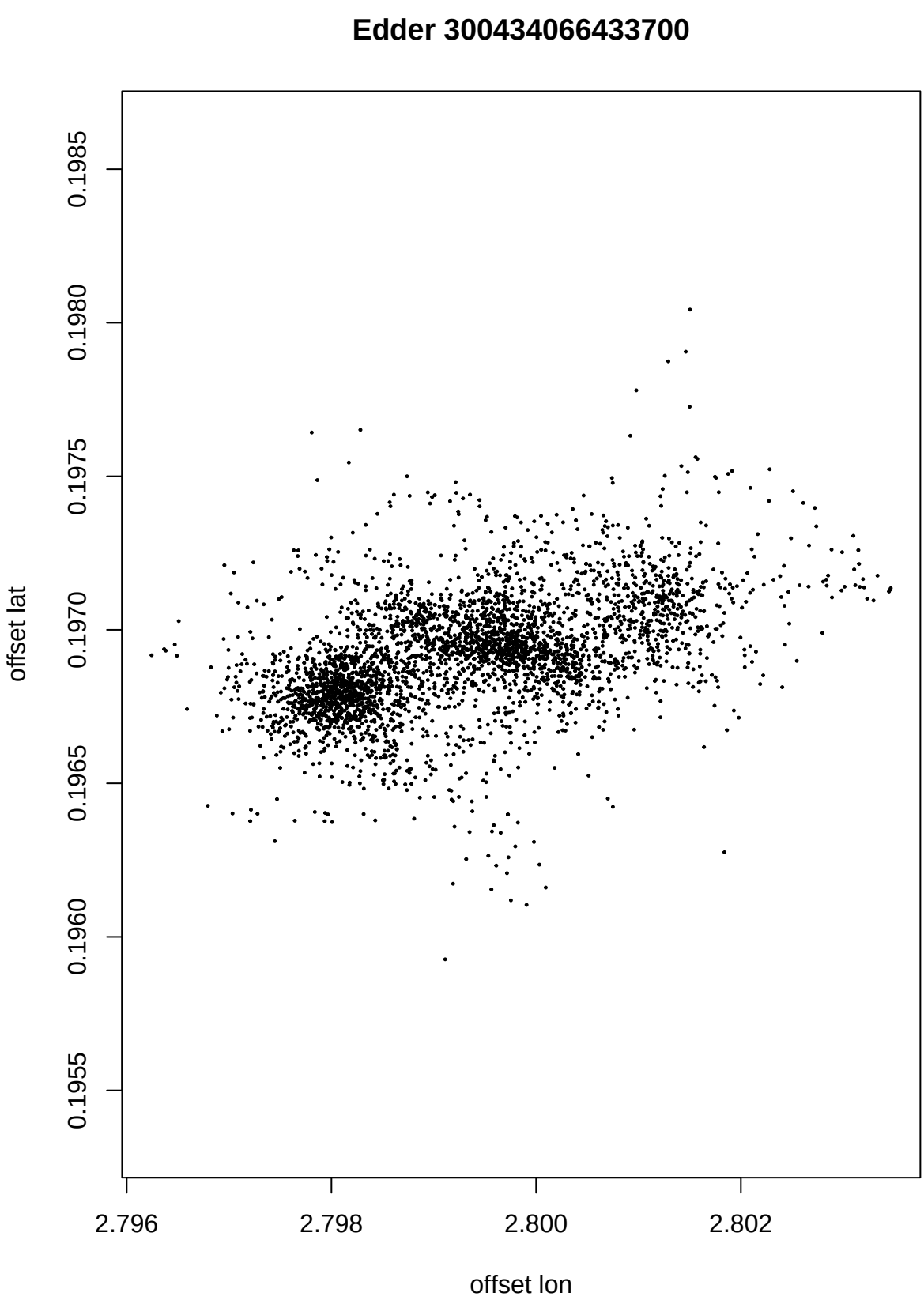
```
## [1] "Mallemuk 300434066431710 0.0018 4e-04"
```


Ismaage 300434066437720
offset lat
0.1730
0.1735
0.1740
0.1745
0.1750
0.1755
0.1760
2.954
2.956
2.958
2.960
offset lon

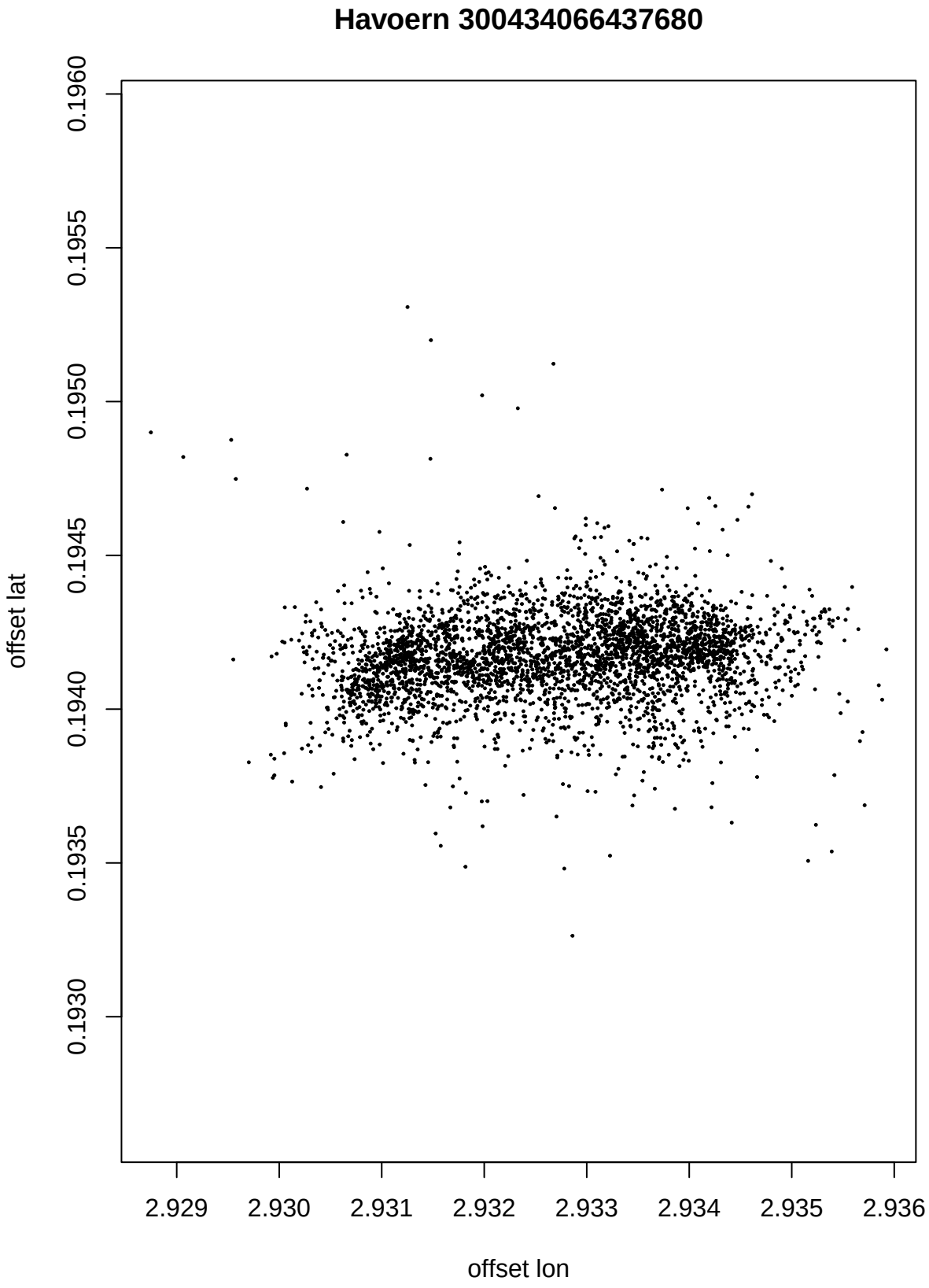
```
## [1] "Edder 300434066433700 0.0012 2e-04"
```


Mallemuk 300434066431710
offset lat
0.203
0.204
0.205
0.206
0.207
0.208
2.890
2.892
2.894
2.896
2.898
2.900
offset lon

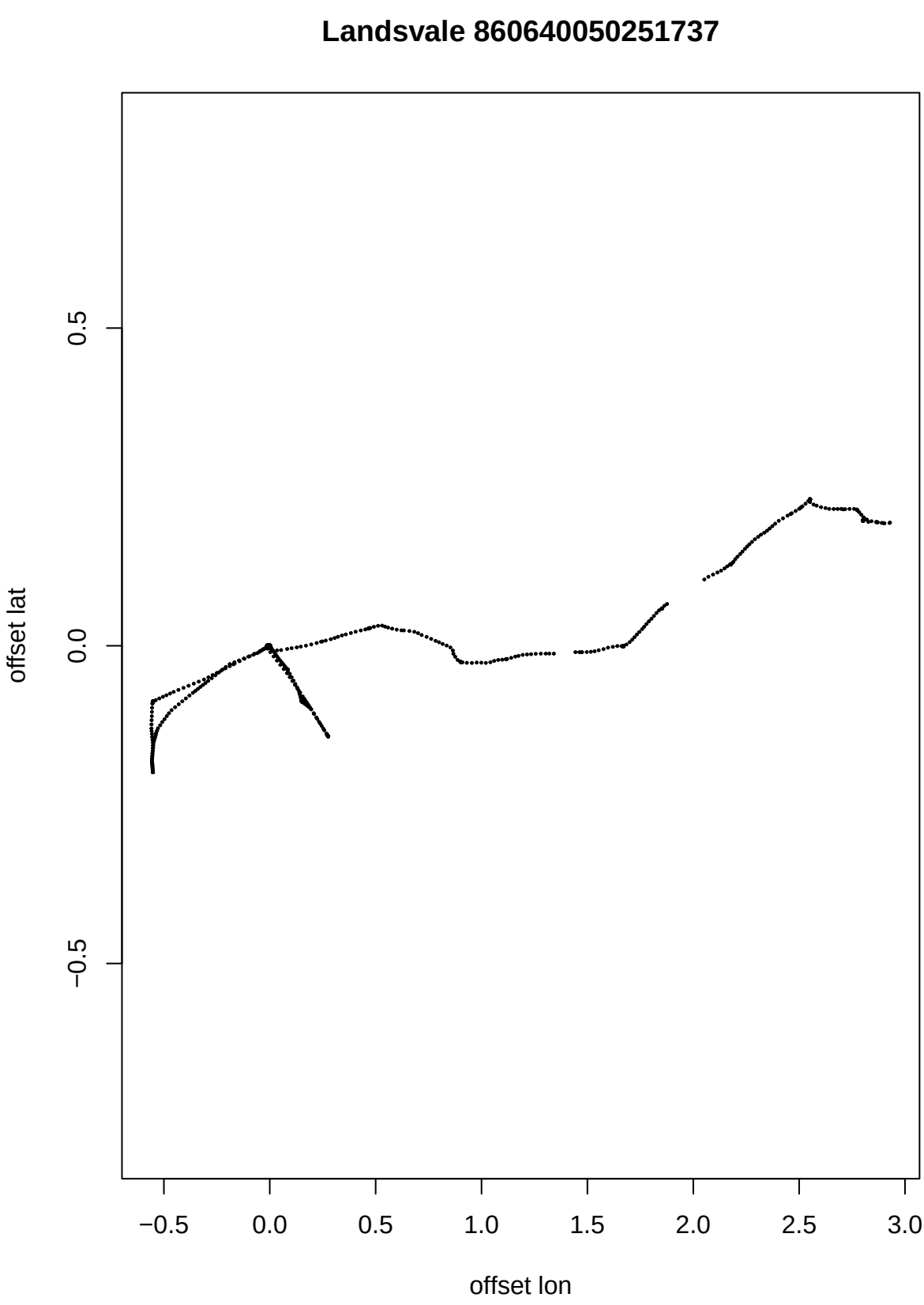
```
## [1] "Havoern 300434066437680 0.0012 1e-04"
```


Edder 300434066433700
offset lat
0.1955
0.1960
0.1965
0.1970
0.1975
0.1980
0.1985
2.796
2.798
2.800
2.802
offset lon

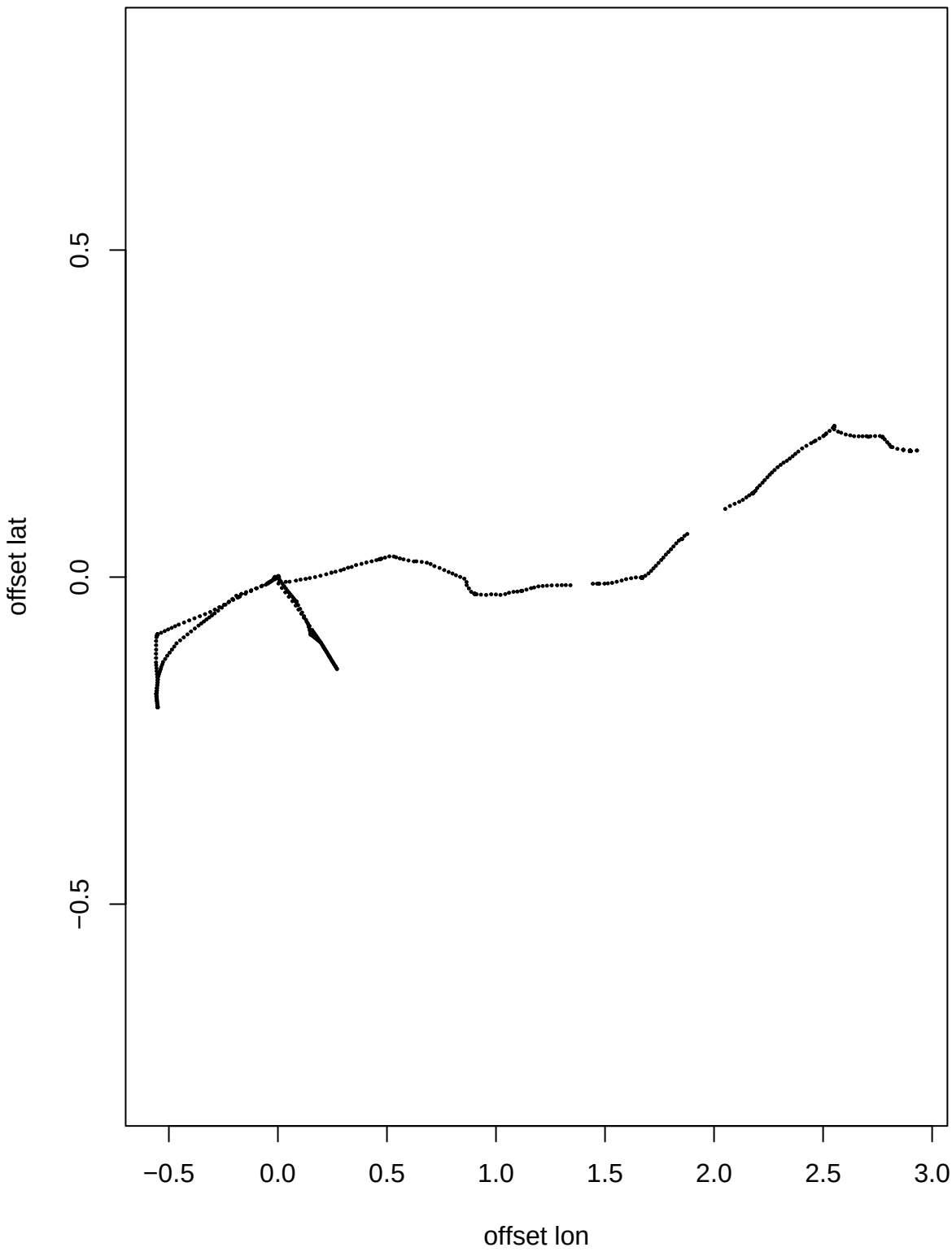
```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] "Landsvale 860640050251737 0.9391 0.0738"
```


Havoern 300434066437680
offset lat
0.1930
0.1935
0.1940
0.1945
0.1950
0.1955
0.1960
2.929
2.930
2.931
2.932
2.933
2.934
2.935
2.936
offset lon

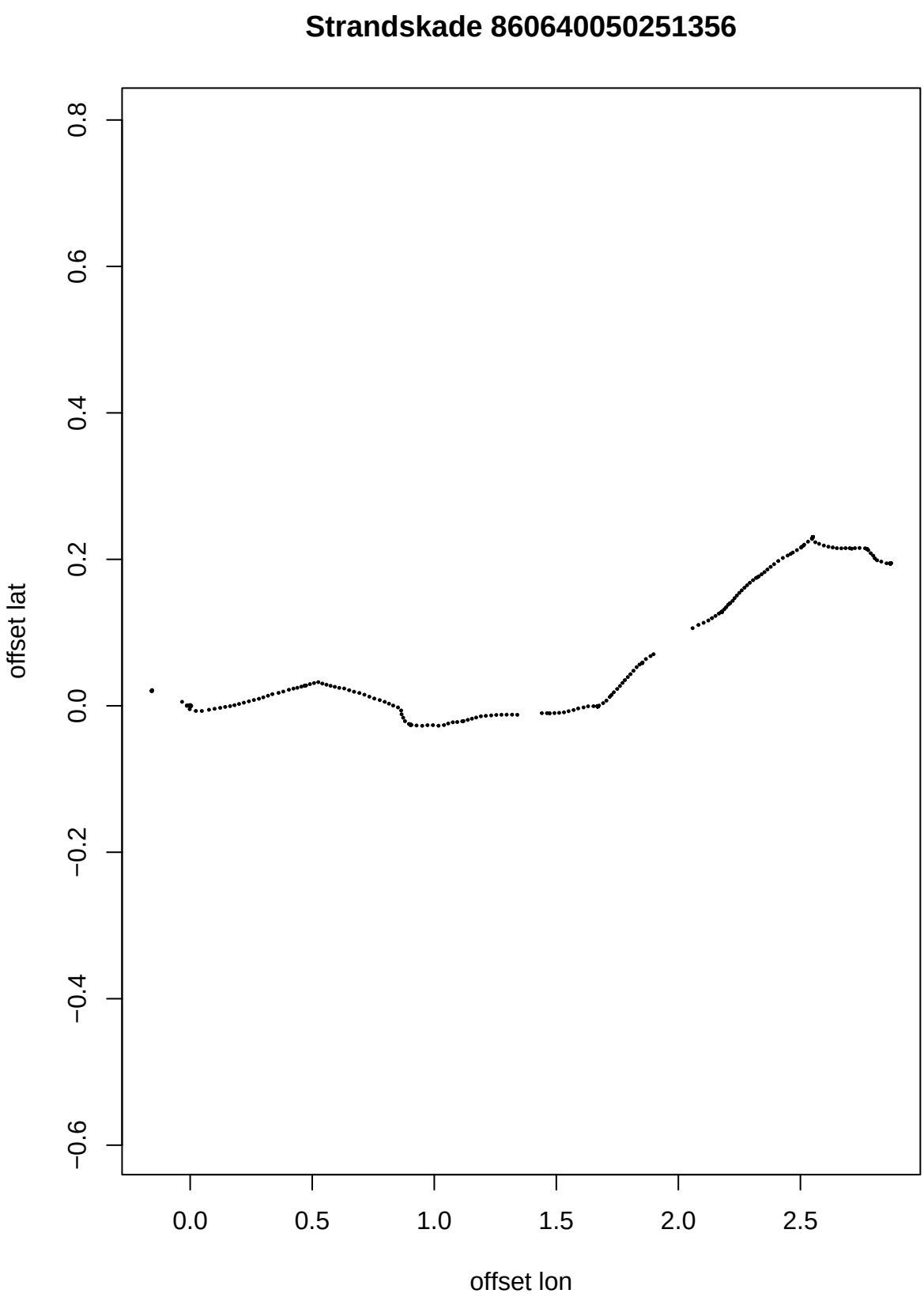
```
## [1] "Ravn 860640050244401 0.9556 0.0726"
```

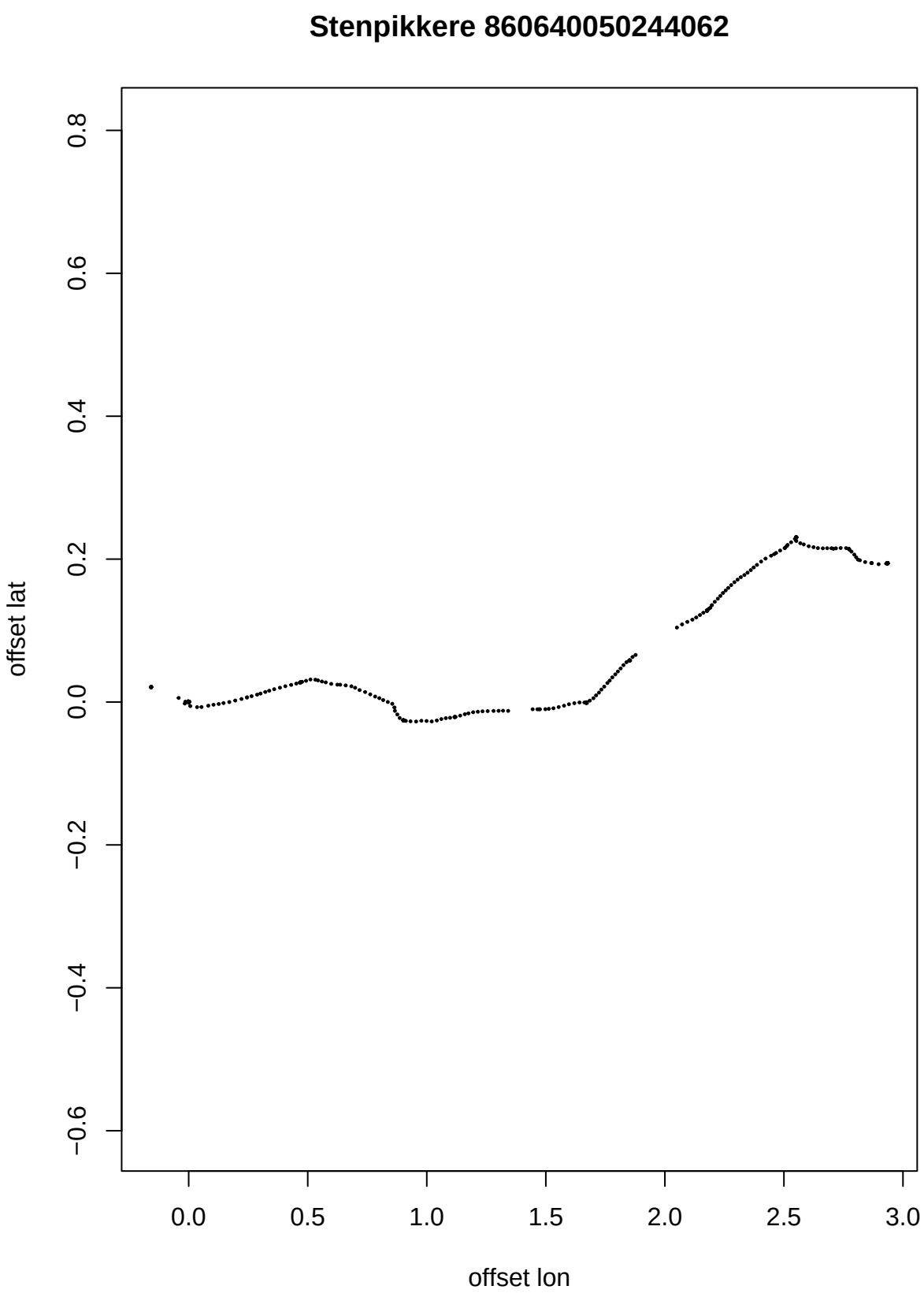

Landsvale 860640050251737
offset lat
0.5
0.0
−0.5
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
offset lon

```
## [1] "Strandskade 860640050251356 1.1759 0.0864"
```


Ravn 860640050244401
offset lat
0.5
0.0
−0.5
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
offset lon

```
## [1] "Stenpikkere 860640050244062 1.1927 0.0862"
```


Strandskade 860640050251356
offset lat
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.5
1.0
1.5
2.0
2.5
offset lon

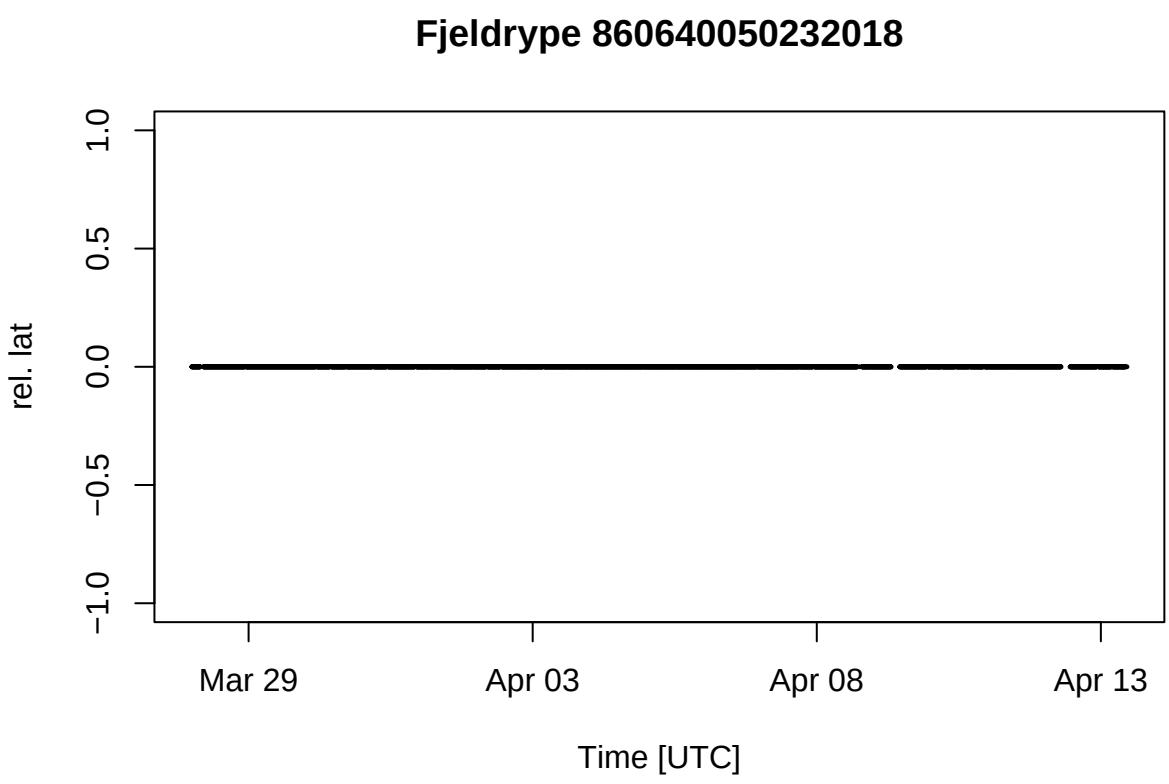

Stenpikkere 860640050244062
offset lat
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.5
1.0
1.5
2.0
2.5
3.0
offset lon

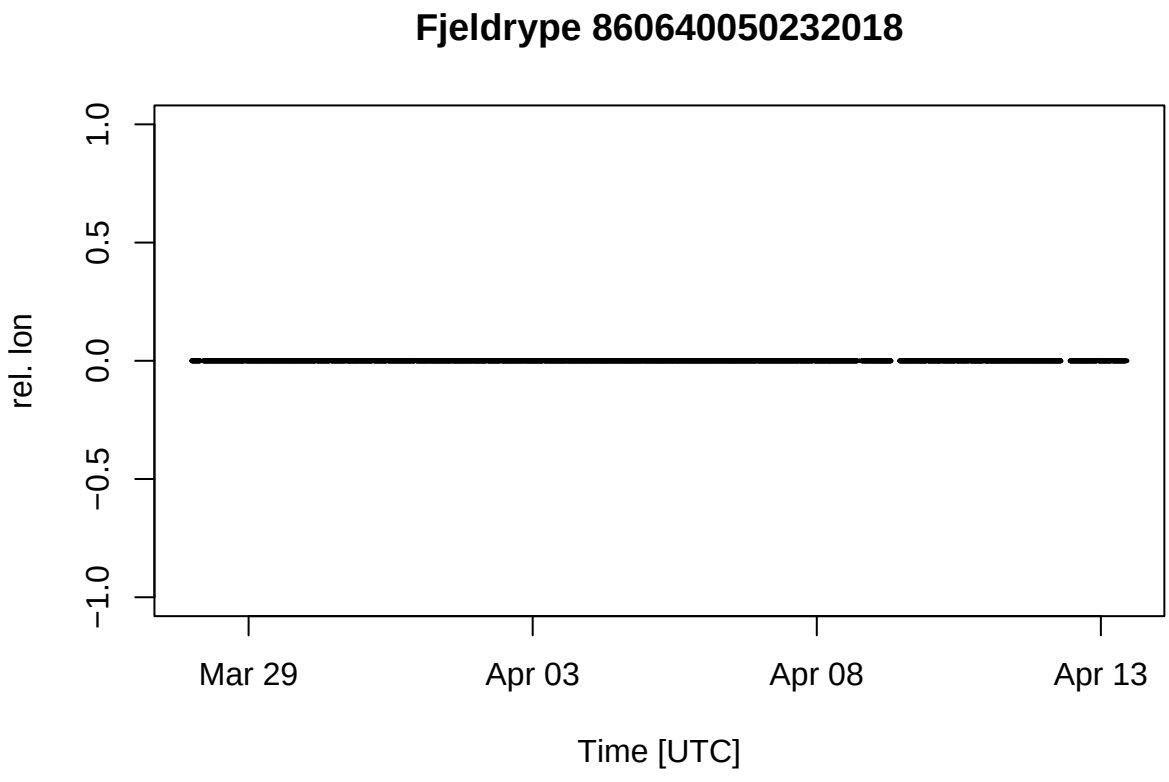
lon/lat relative to Fjeldrype vs time

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

alldf <- NULL
#
for (jstat in 1:length(unique_names))
{

  #print(jstat)
  statname <- unique_names[jstat]
  #print(statname)
  png(paste0("FIGURES/offset_lon_lat_rel_Fjeldrype_vs_time_",statname,".png"))
  par(mfrow=c(2,1))
  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$delta_lat),4)))
  plot(together$POSIX,together$delta_lat,main=statname,xlab="Time [UTC]",ylab="rel. lat",pch=19,cex=0.2
  plot(together$POSIX,together$delta_lon,main=statname,xlab="Time [UTC]",ylab="rel. lon",pch=19,cex=0.2
  #
  dev.off()
  par(mfrow=c(2,1))
  plot(together$POSIX,together$delta_lat,main=statname,xlab="Time [UTC]",ylab="rel. lat",pch=19,cex=0.2
  plot(together$POSIX,together$delta_lon,main=statname,xlab="Time [UTC]",ylab="rel. lon",pch=19,cex=0.2
}
```

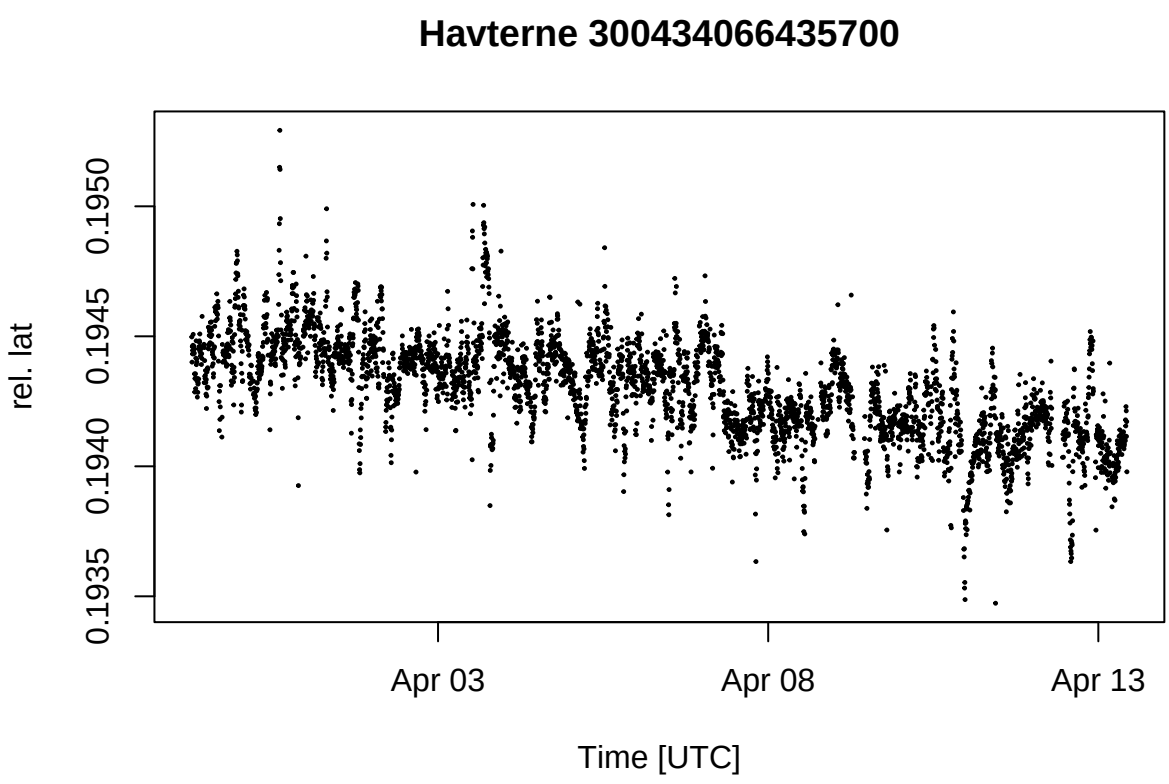
```
## [1] "Fjeldrype 860640050232018 0 0"
```

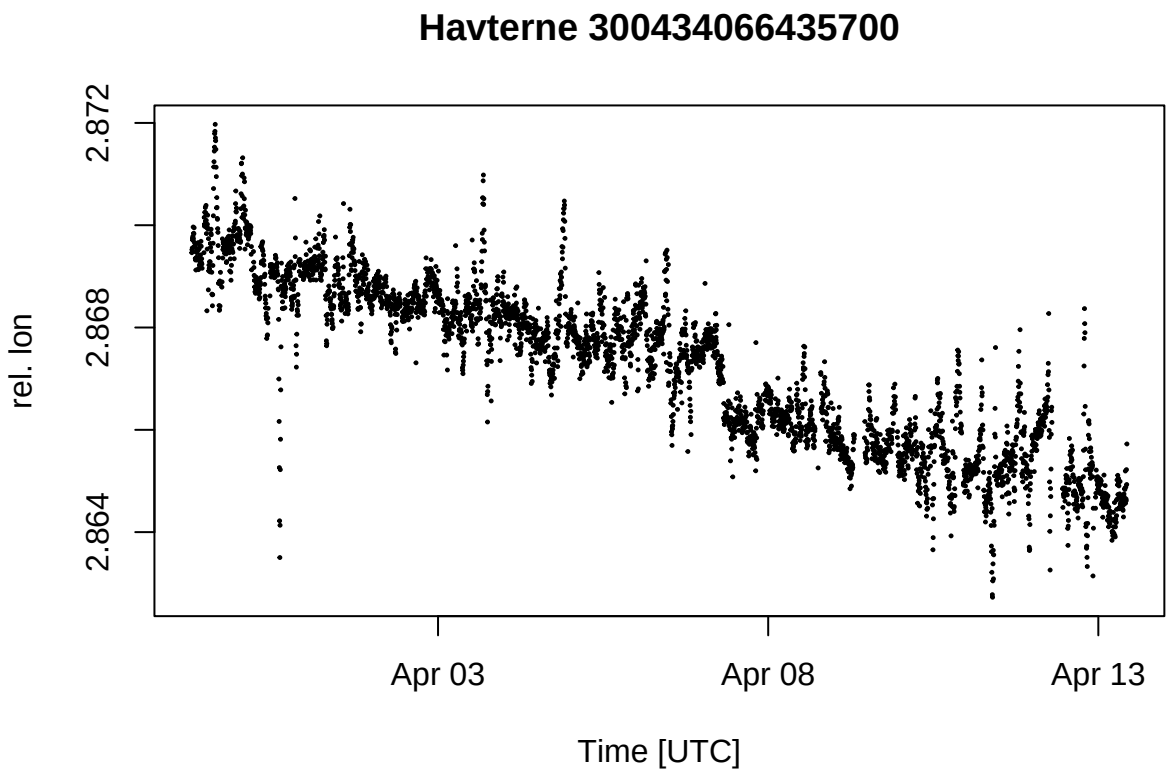
```
## [1] "Havterne 300434066435700 0.0016 2e-04"
```


Fjeldrype 860640050232018
rel. lat
1.0
0.5
0.0
−0.5
−1.0
Mar 29
Apr 03
Apr 08
Apr 13
Time [UTC]

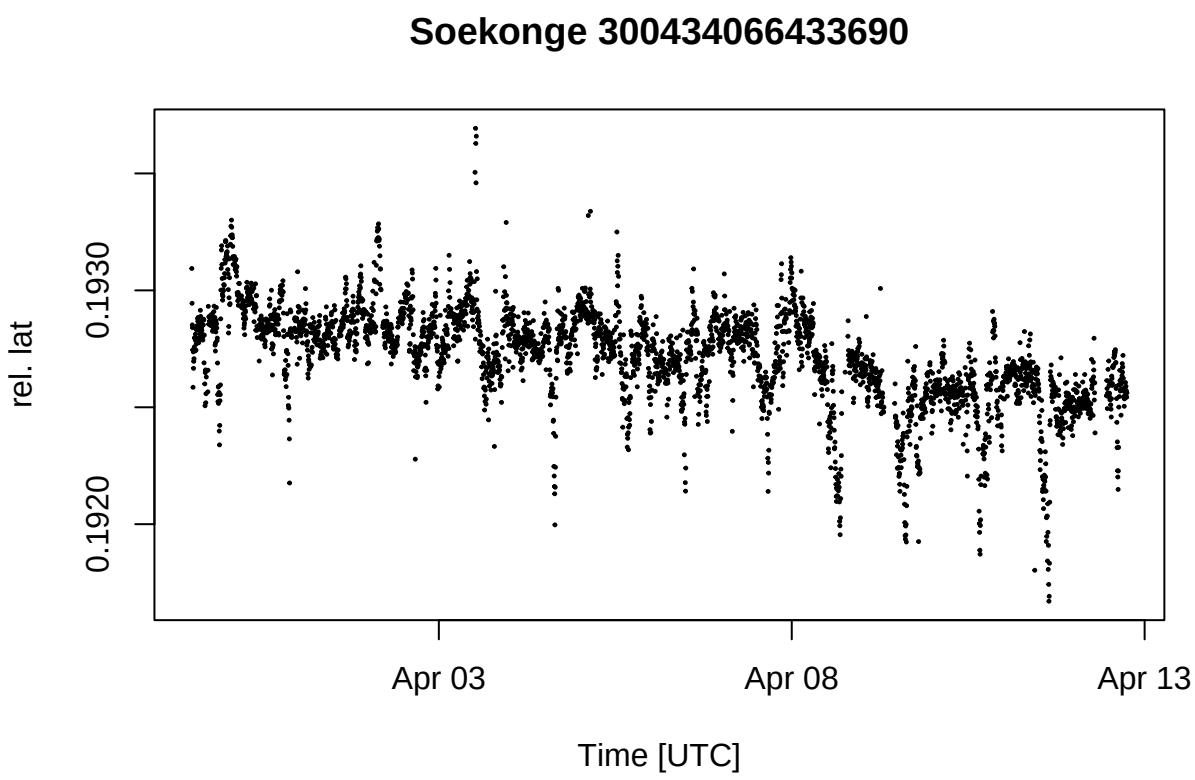

Fjeldrype 860640050232018
rel. lon
1.0
0.5
0.0
−0.5
−1.0
Mar 29
Apr 03
Apr 08
Apr 13
Time [UTC]

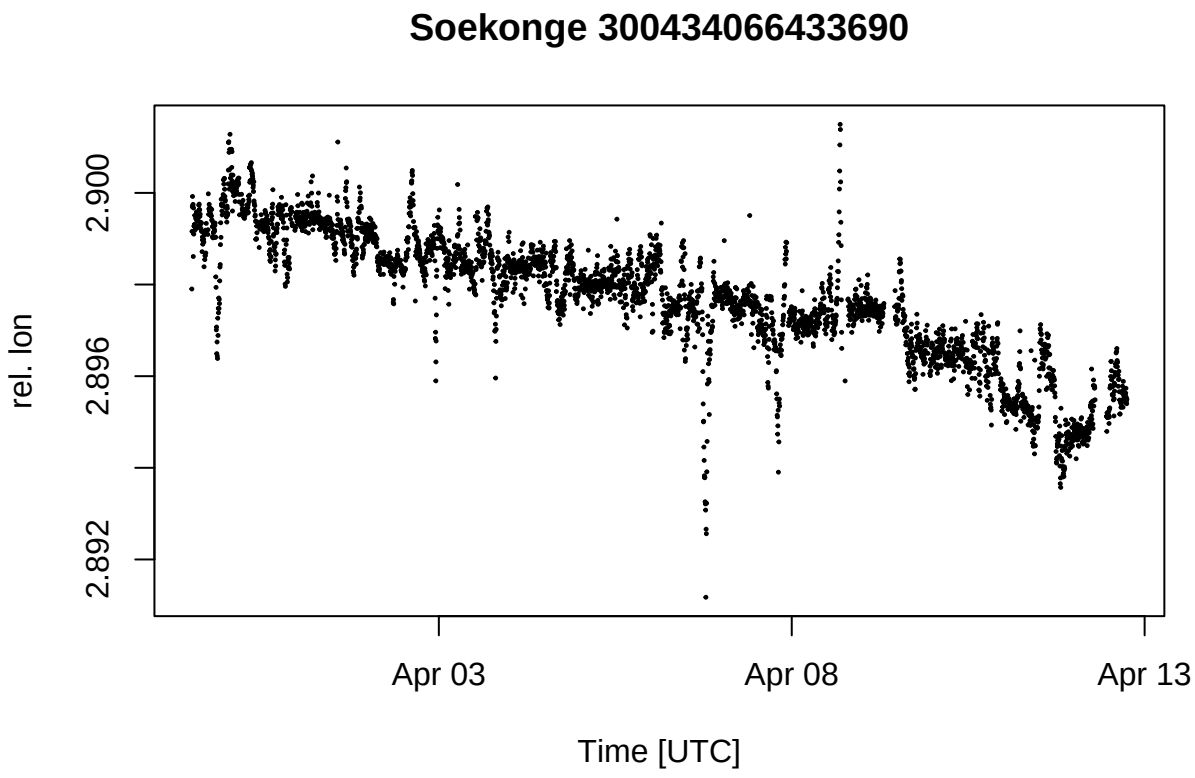
```
## [1] "Soekonge 300434066433690 0.0014 2e-04"
```


Havterne 300434066435700
rel. lat
0.1935
0.1940
0.1945
0.1950
Apr 03
Apr 08
Apr 13
Time [UTC]

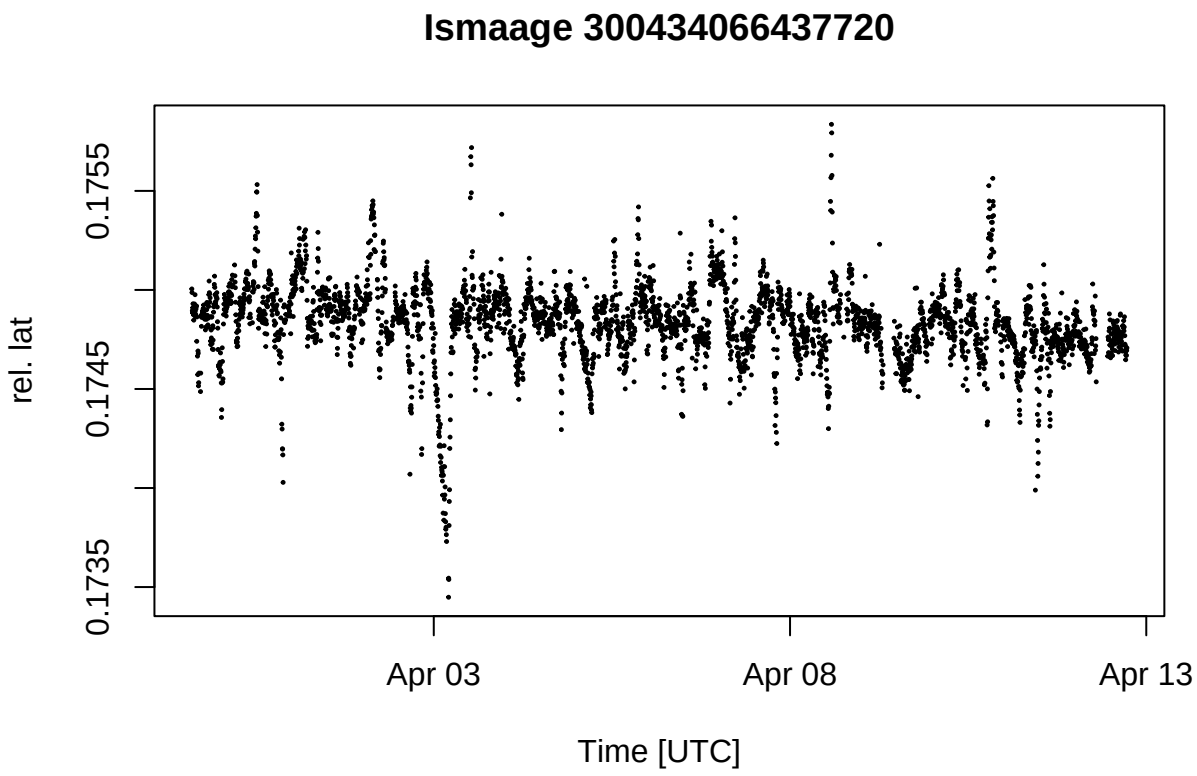

Havterne 300434066435700
rel. lon
2.864
2.868
2.872
Apr 03
Apr 08
Apr 13
Time [UTC]

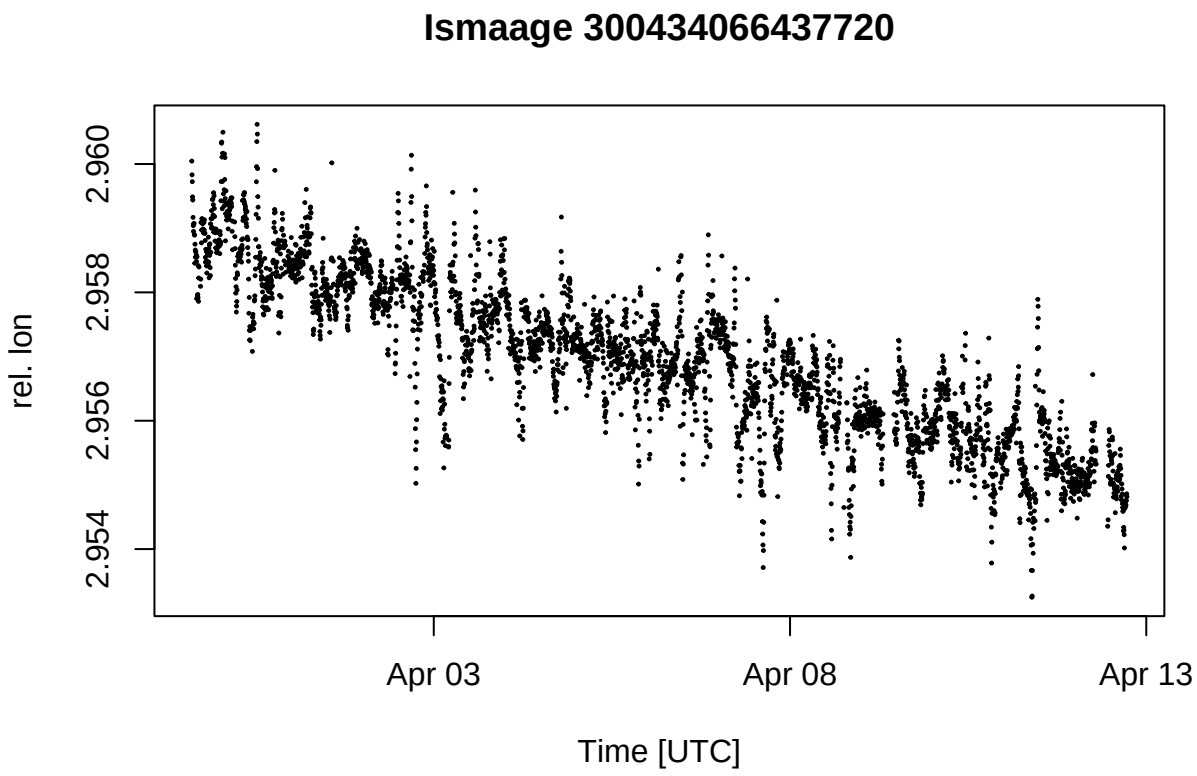
```
## [1] "Ismaage 300434066437720 0.0012 2e-04"
```


Soekonge 300434066433690
rel. lat
0.1930
0.1920
Apr 03
Apr 08
Apr 13
Time [UTC]

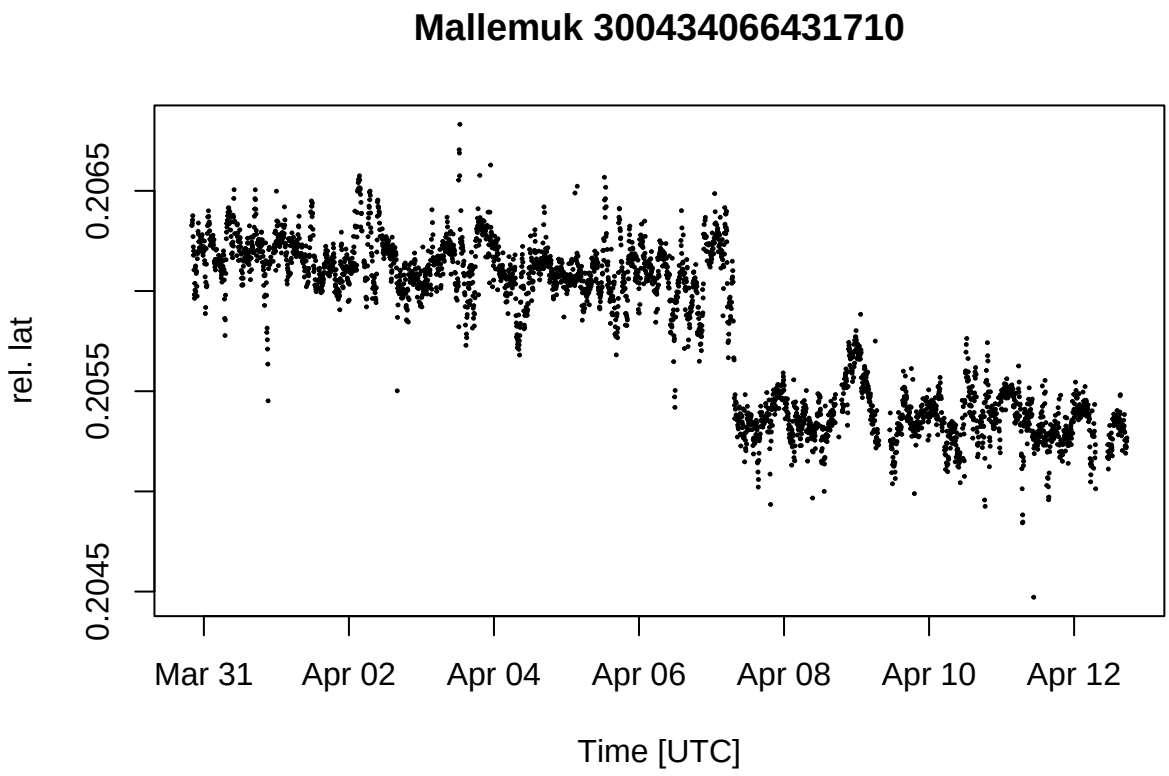

Soekonge 300434066433690
rel. lon
2.900
2.896
2.892
Apr 03
Apr 08
Apr 13
Time [UTC]

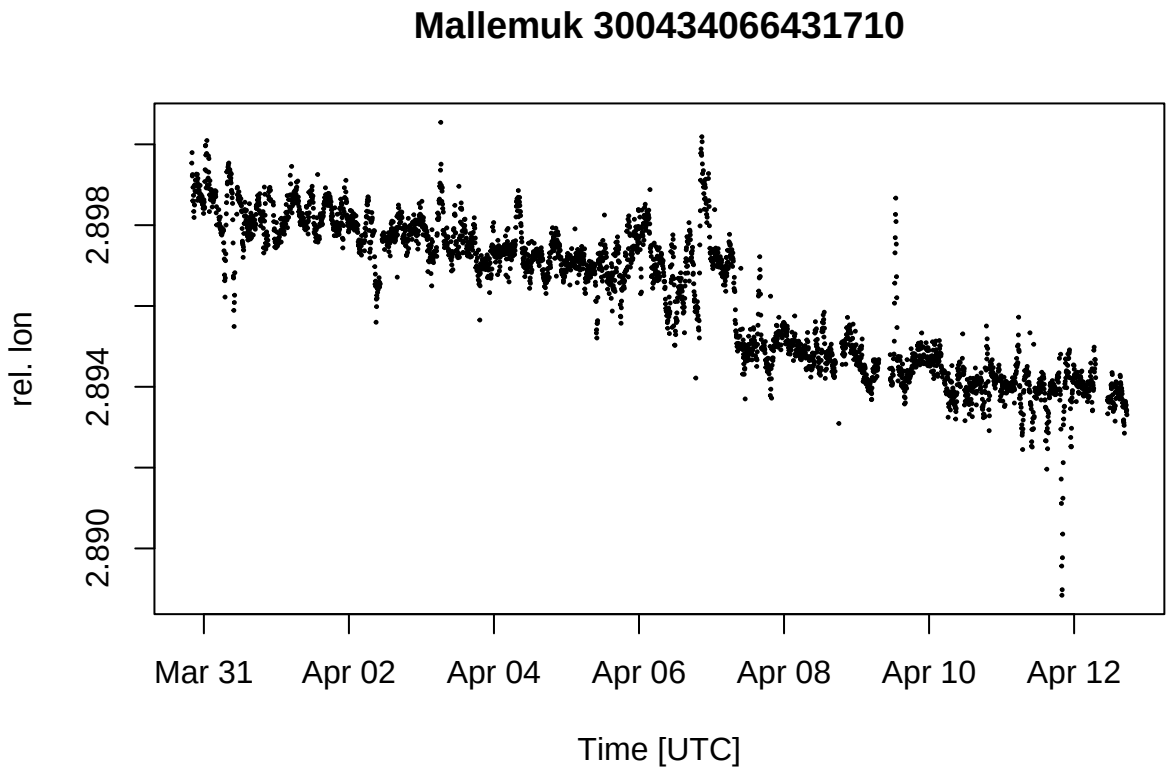
```
## [1] "Mallemuk 300434066431710 0.0018 4e-04"
```


Ismaage 300434066437720
rel. lat
0.1735
0.1745
0.1755
Apr 03
Apr 08
Apr 13
Time [UTC]

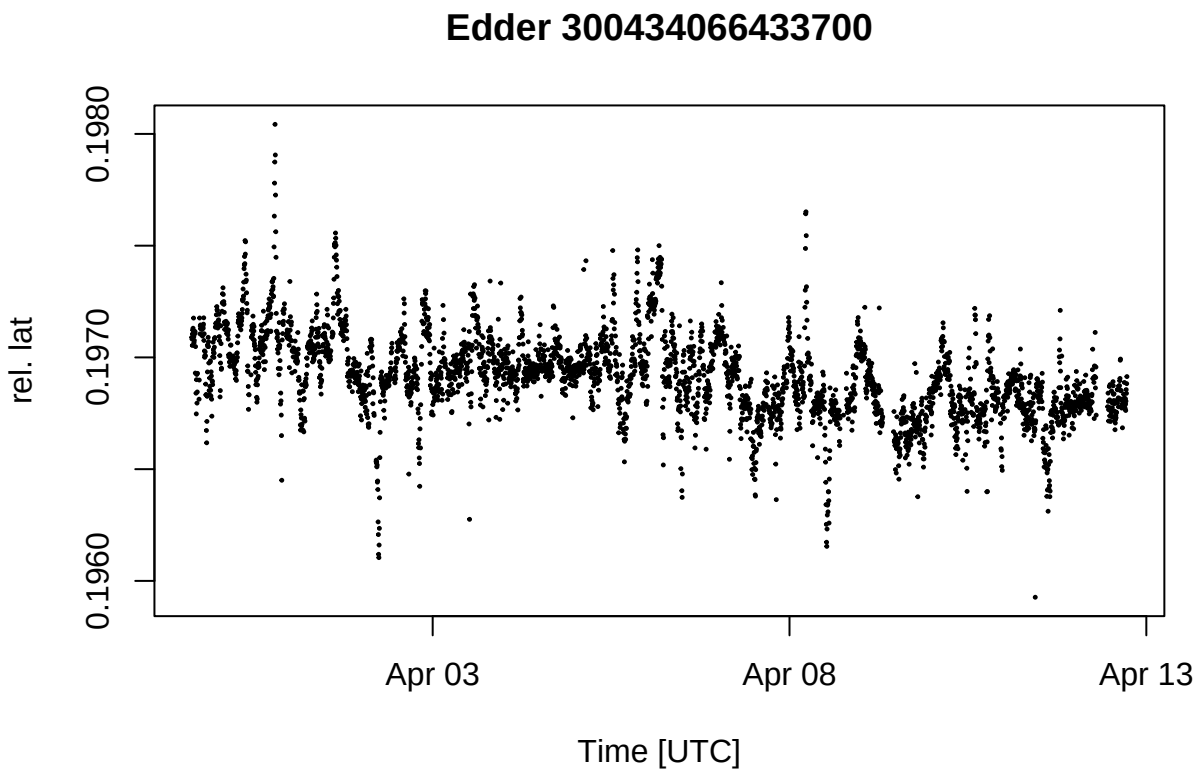

Ismaage 300434066437720
rel. lon
2.954
2.956
2.958
2.960
Apr 03
Apr 08
Apr 13
Time [UTC]

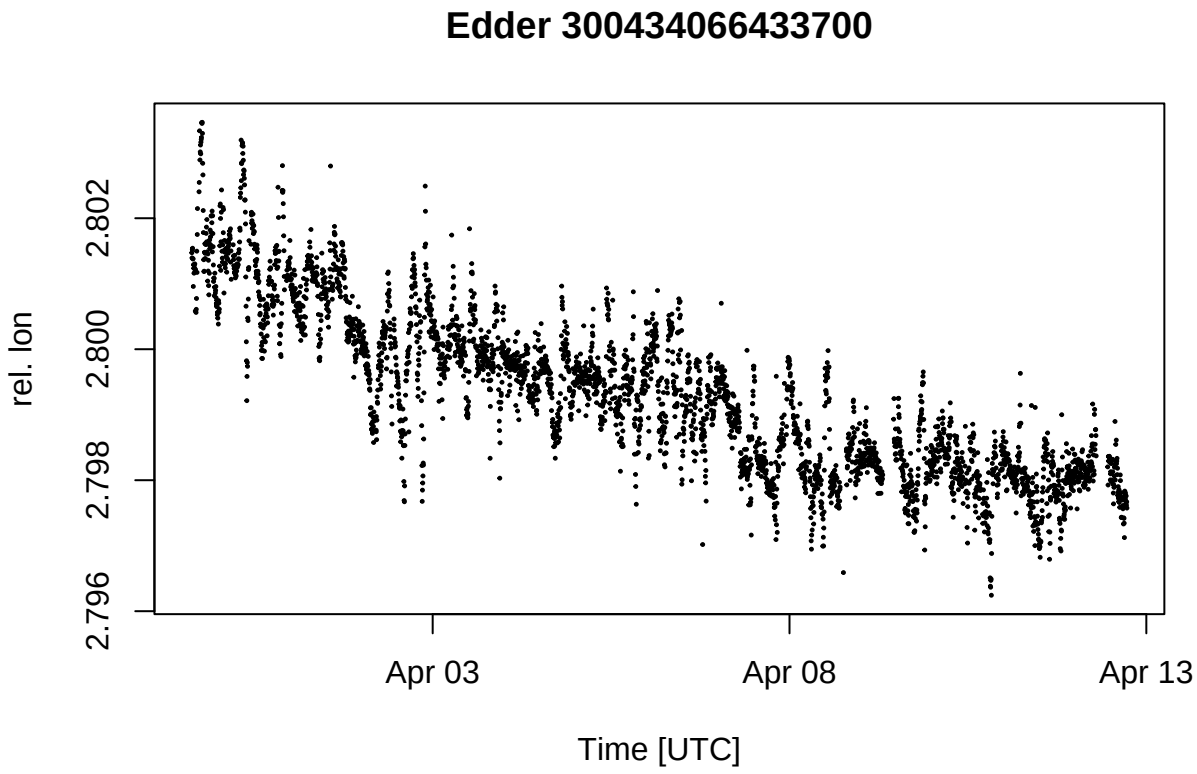
```
## [1] "Edder 300434066433700 0.0012 2e-04"
```


Mallemuk 300434066431710
rel. lat
0.2065
0.2055
0.2045
Mar 31
Apr 02
Apr 04
Apr 06
Apr 08
Apr 10
Apr 12
Time [UTC]

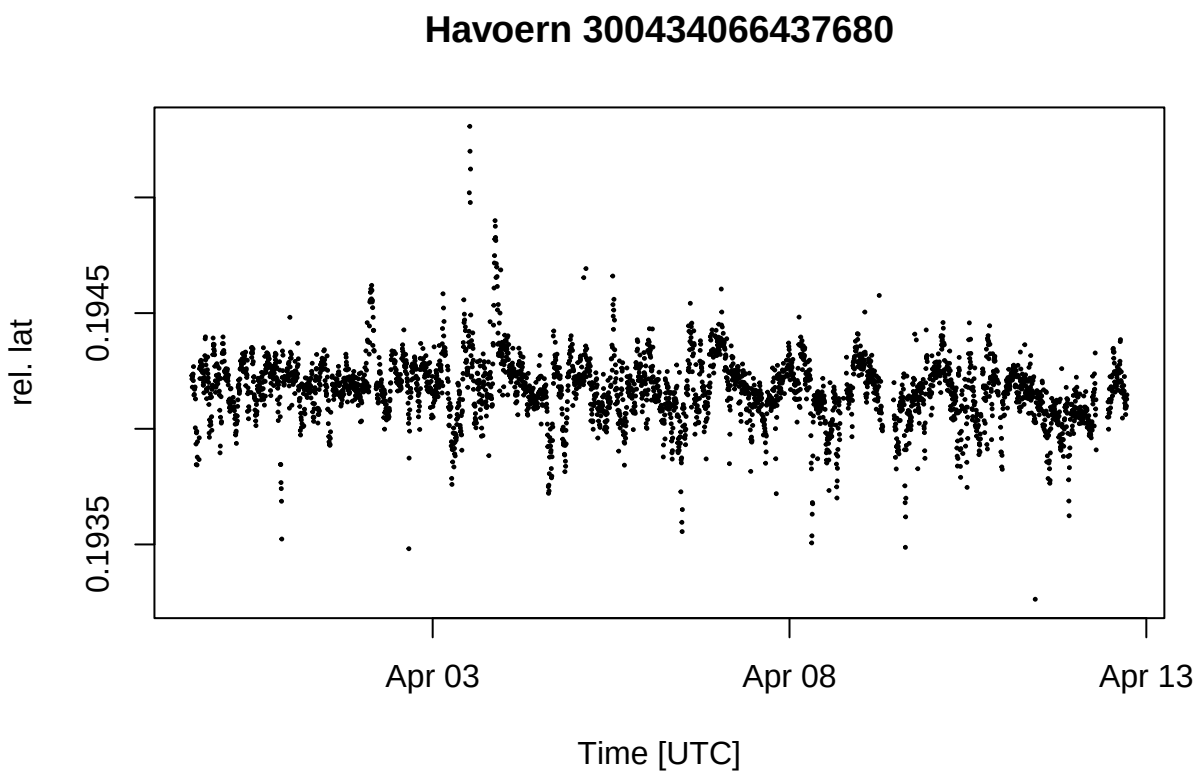

Mallemuk 300434066431710
rel. lon
2.898
2.894
2.890
Mar 31
Apr 02
Apr 04
Apr 06
Apr 08
Apr 10
Apr 12
Time [UTC]

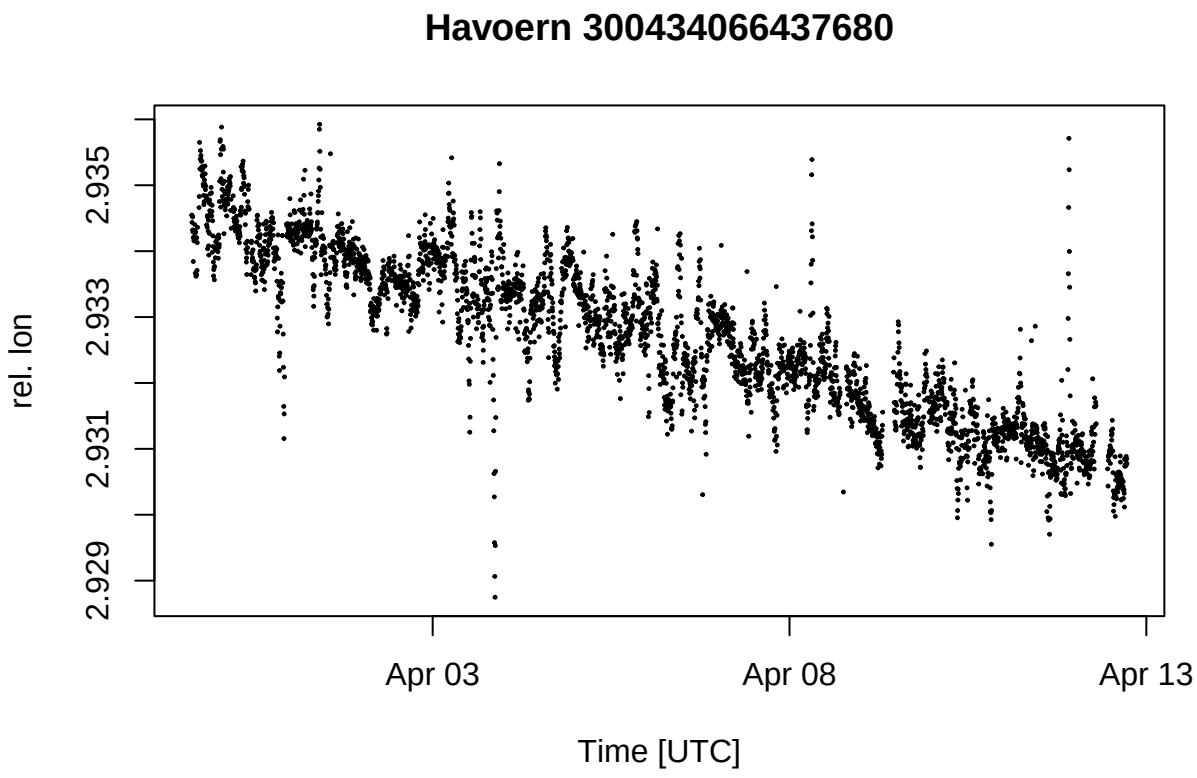
```
## [1] "Havoern 300434066437680 0.0012 1e-04"
```


Edder 300434066433700
rel. lat
0.1980
0.1970
0.1960
Apr 03
Apr 08
Apr 13
Time [UTC]

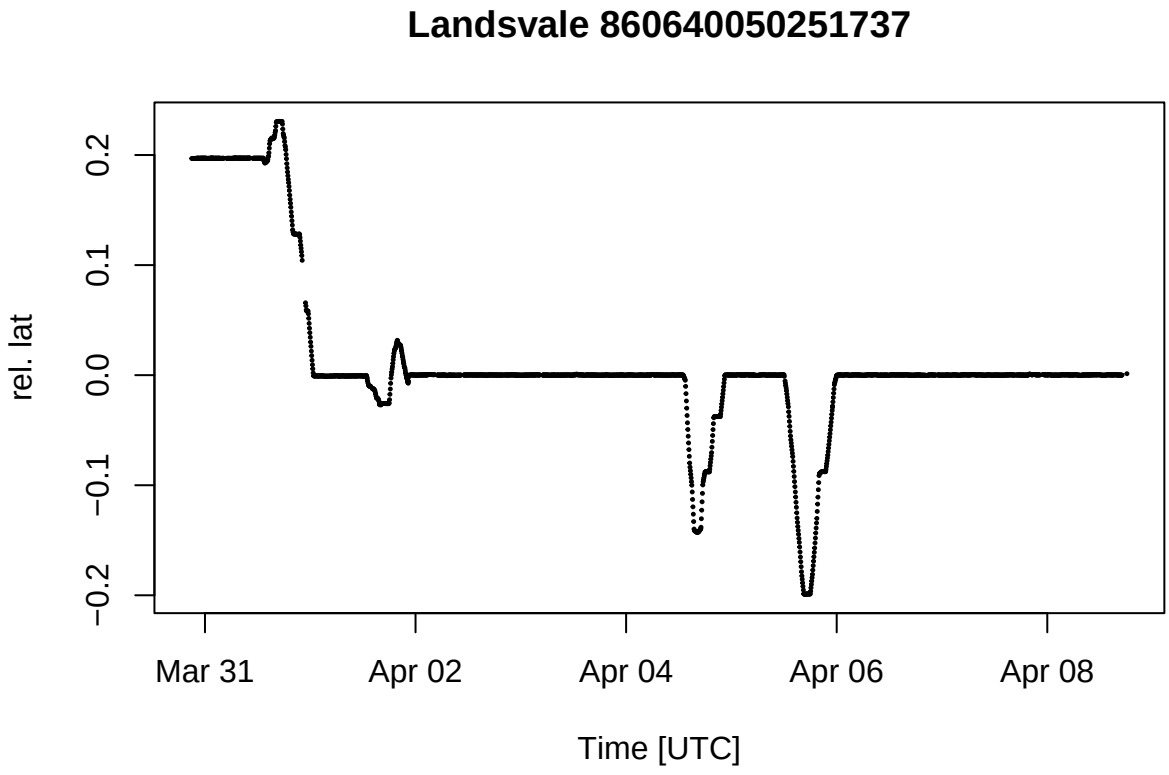

Edder 300434066433700
rel. lon
2.802
2.800
2.798
2.796
Apr 03
Apr 08
Apr 13
Time [UTC]

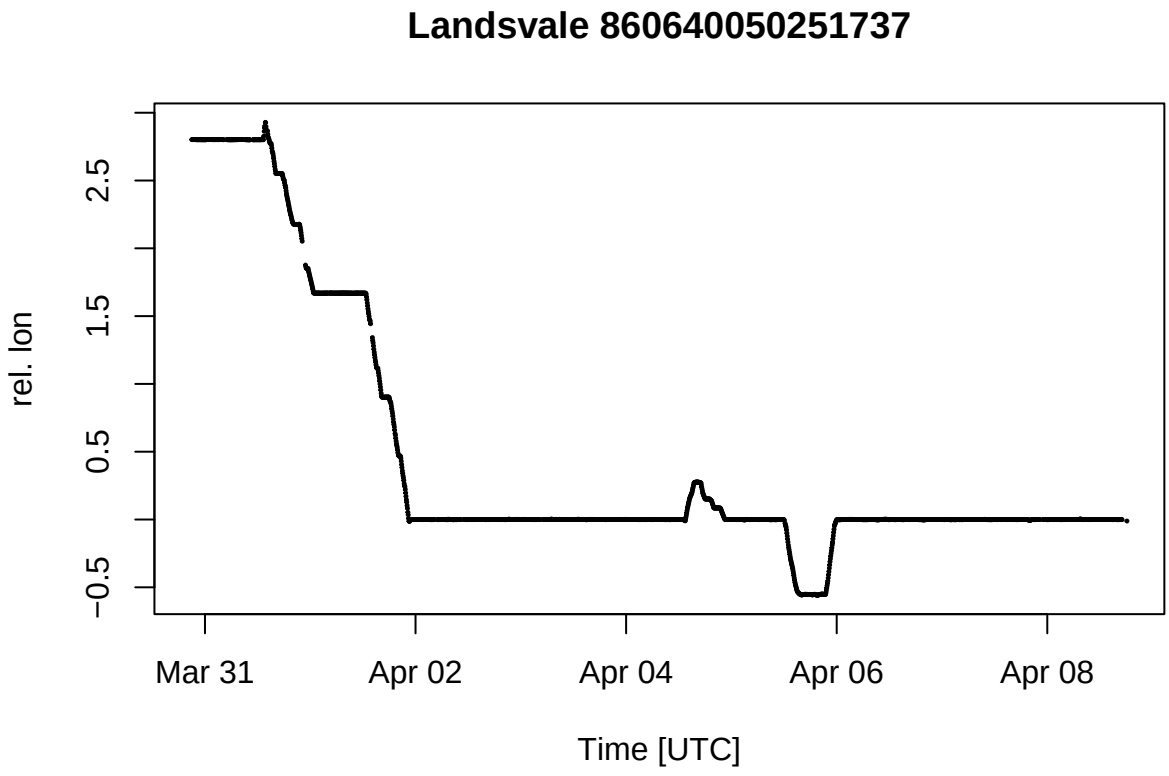
```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## [1] "Landsvale 860640050251737 0.9391 0.0738"
```


Havoern 300434066437680
rel. lat
0.1935
0.1945
Apr 03
Apr 08
Apr 13
Time [UTC]

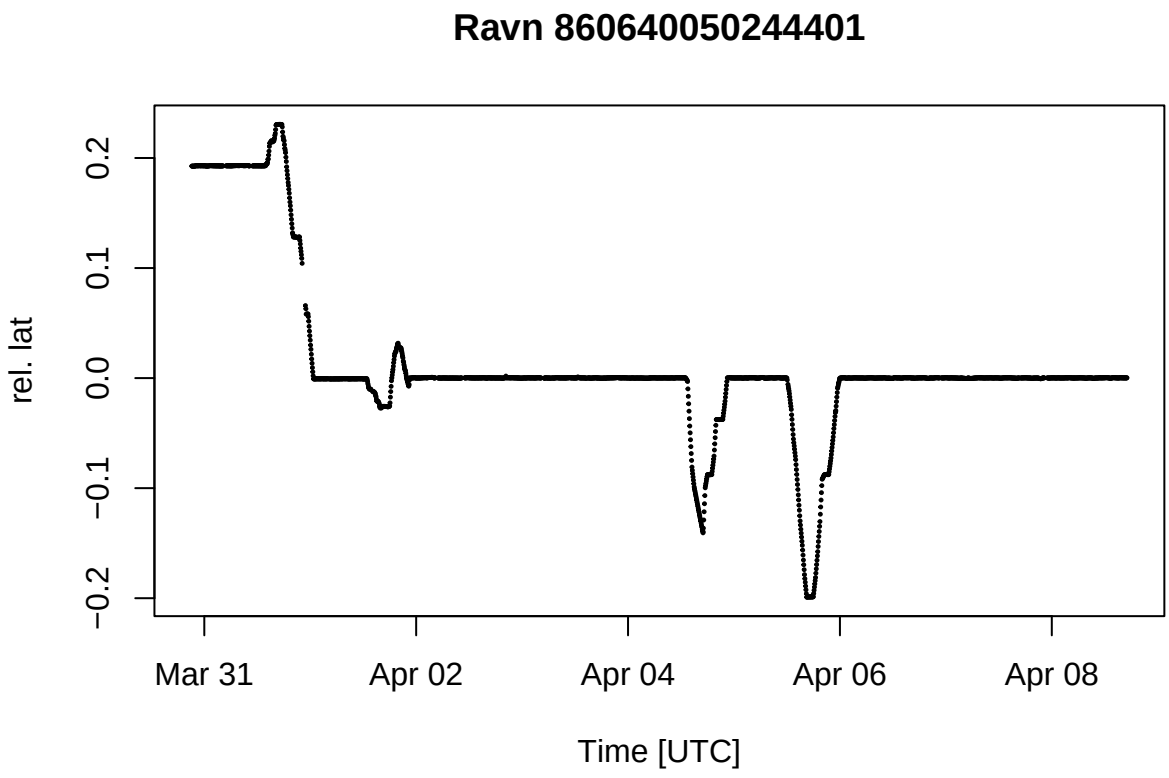

Havoern 300434066437680
rel. lon
2.929
2.931
2.933
2.935
Apr 03
Apr 08
Apr 13
Time [UTC]

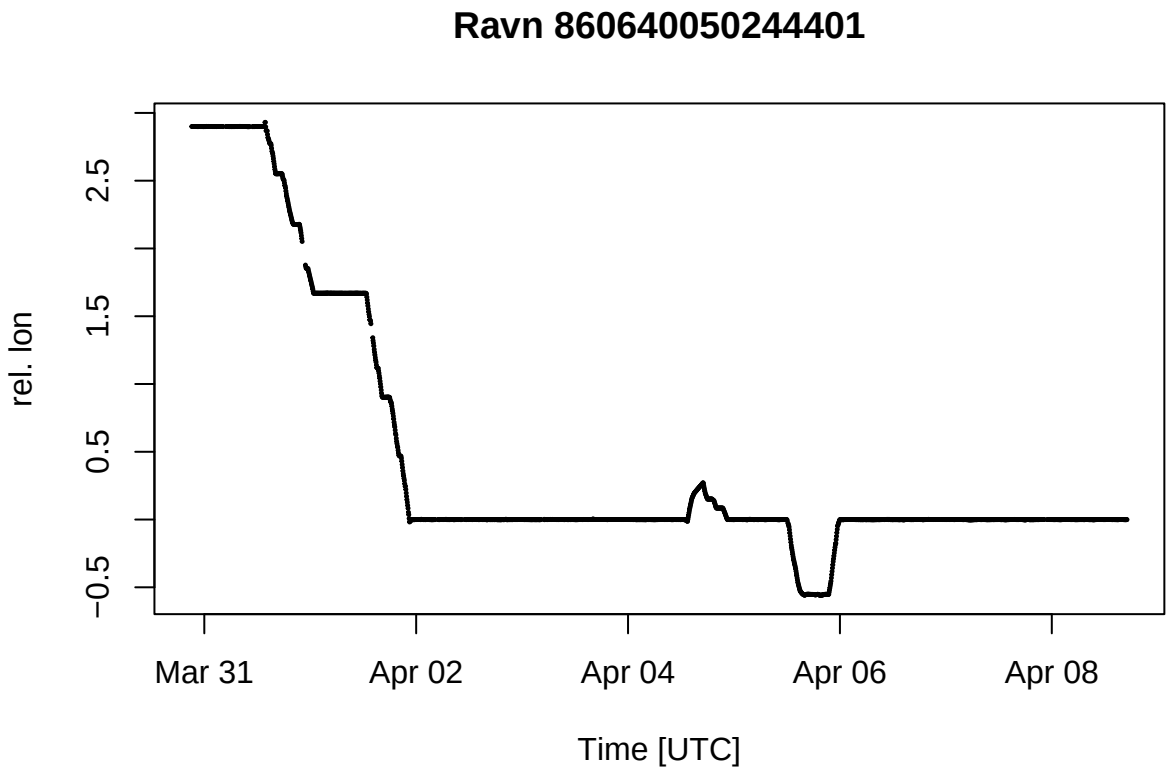
```
## [1] "Ravn 860640050244401 0.9556 0.0726"
```


Landsvale 860640050251737
rel. lat
0.2
0.1
0.0
−0.1
−0.2
Mar 31
Apr 02
Apr 04
Apr 06
Apr 08
Time [UTC]

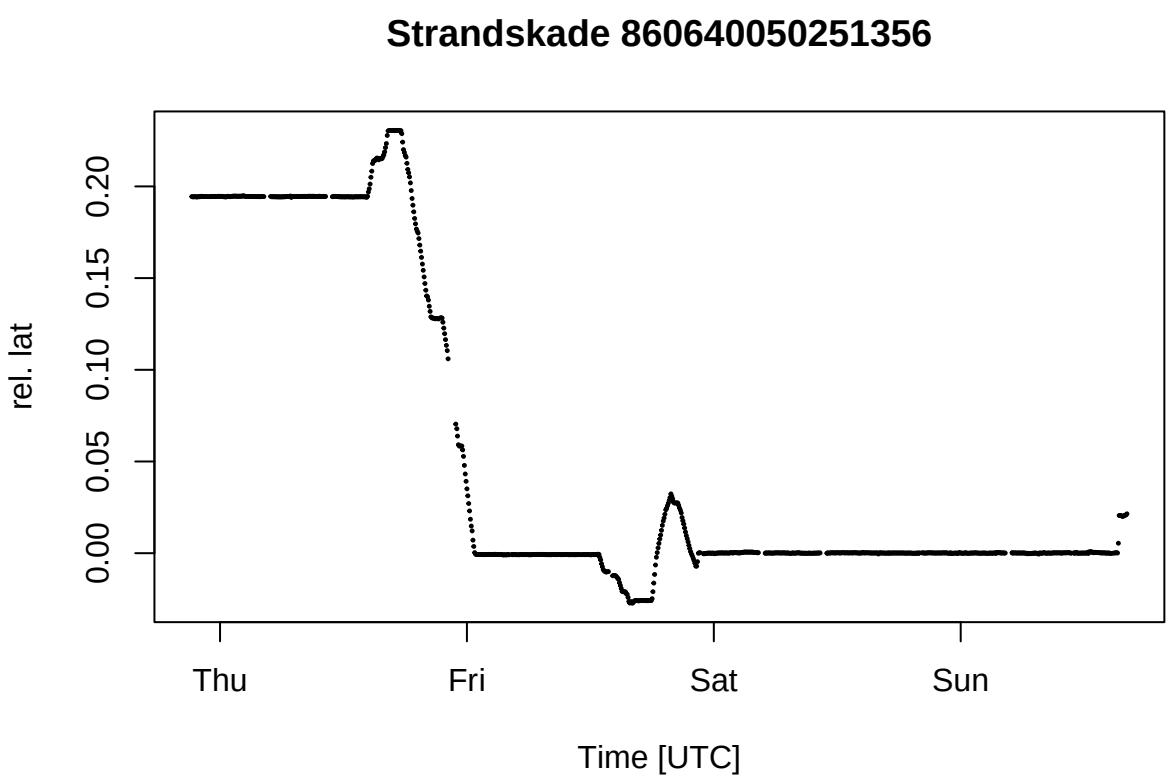

Landsvale 860640050251737
rel. lon
2.5
1.5
0.5
−0.5
Mar 31
Apr 02
Apr 04
Apr 06
Apr 08
Time [UTC]

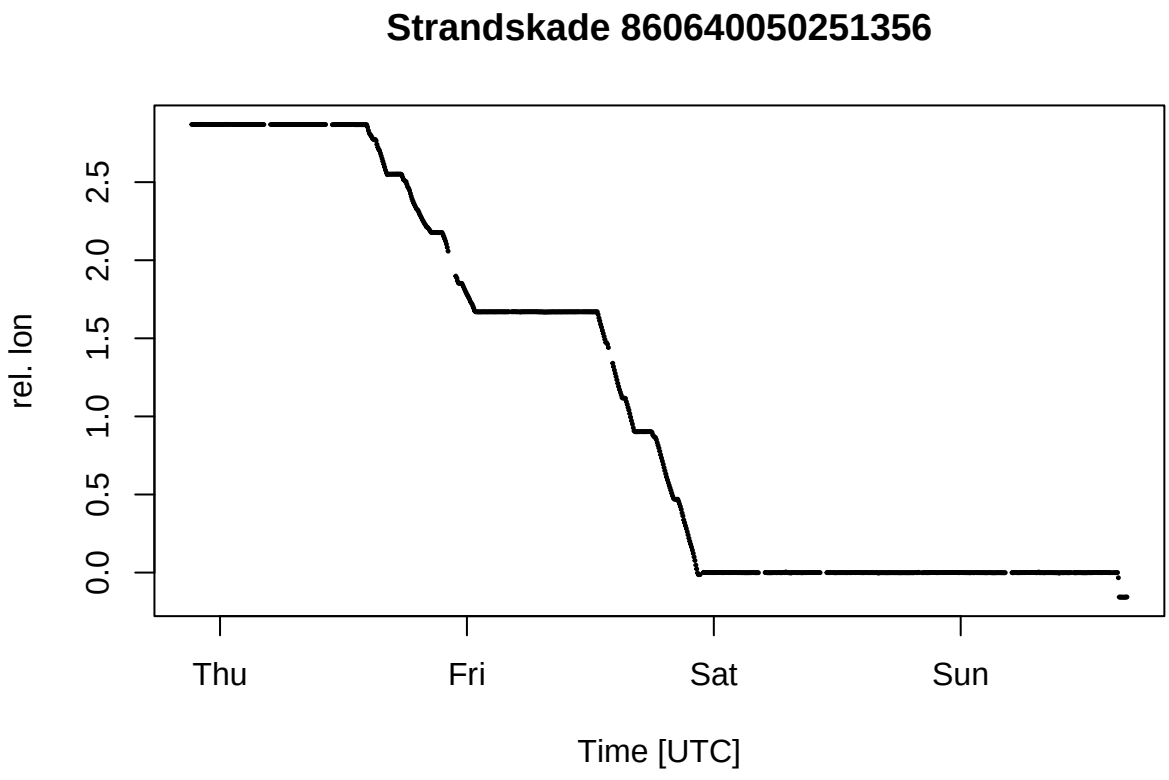
```
## [1] "Strandskade 860640050251356 1.1759 0.0864"
```

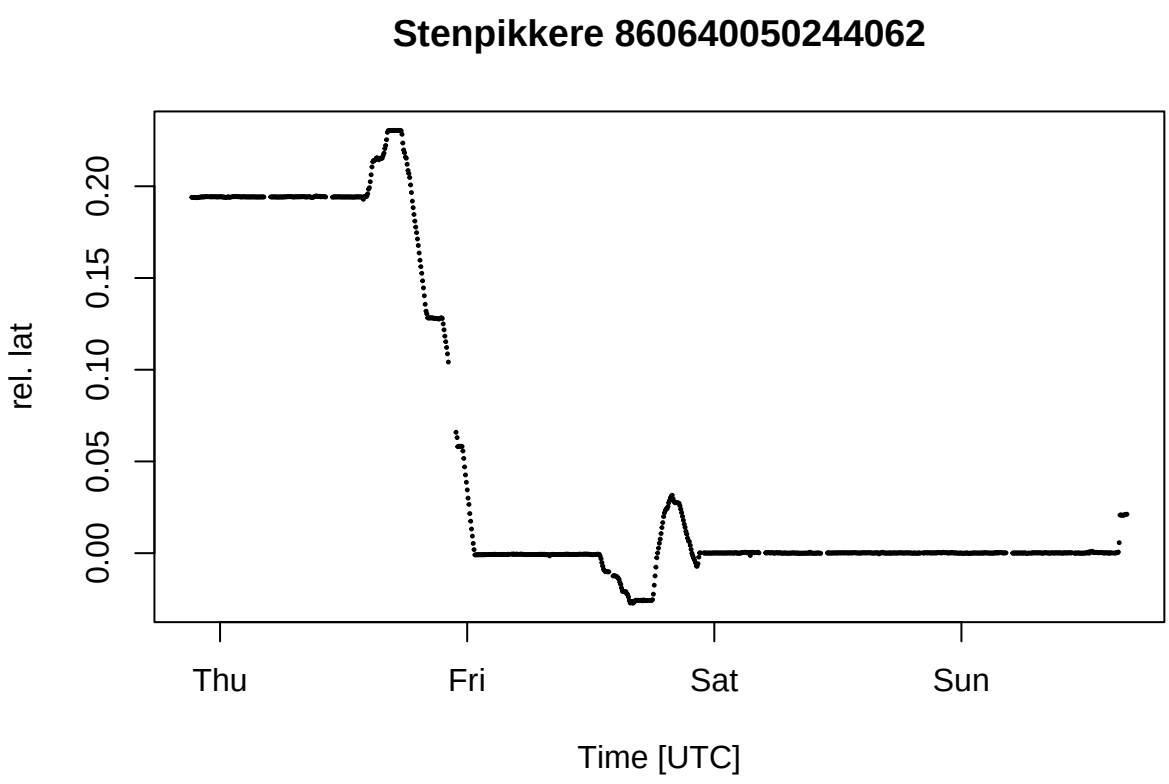

Ravn 860640050244401
rel. lat
0.2
0.1
0.0
−0.1
−0.2
Mar 31
Apr 02
Apr 04
Apr 06
Apr 08
Time [UTC]

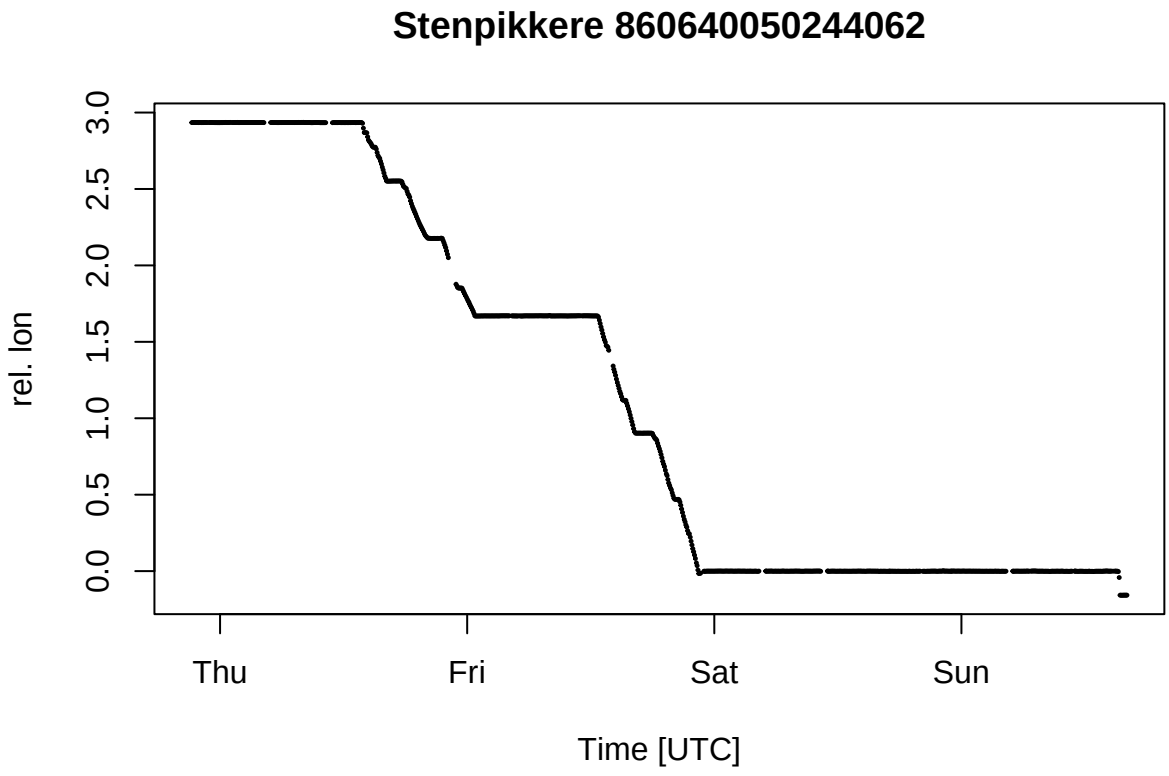

Ravn 860640050244401
rel. lon
2.5
1.5
0.5
−0.5
Mar 31
Apr 02
Apr 04
Apr 06
Apr 08
Time [UTC]

```
## [1] "Stenpikkere 860640050244062 1.1927 0.0862"
```


Strandskade 860640050251356
rel. lat
0.00
0.05
0.10
0.15
0.20
Thu
Fri
Sat
Sun
Time [UTC]


Strandskade 860640050251356
rel. lon
0.0
0.5
1.0
1.5
2.0
2.5
Thu
Fri
Sat
Sun
Time [UTC]


Stenpikkere 860640050244062
rel. lat
0.00
0.05
0.10
0.15
0.20
Thu
Fri
Sat
Sun
Time [UTC]


Stenpikkere 860640050244062
rel. lon
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Thu
Fri
Sat
Sun
Time [UTC]