

R Notebook

TODO:

- a) split series into shorter bits and see if the spectrum changes

```
rm(list=ls())
setwd("~/WORKSHOP/QAANAAQ/")
library(sp)
library(nleqslv)
library(lubridate)

##
## Attaching package: 'lubridate'

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

library(oce)

## Loading required package: gsw

library(nloptr)
library(maps)
library(ggplot2)
library(ggmap)

## i Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
## i Please cite ggmap if you use it! Use 'citation("ggmap")' for details.

library(parallel)
library(sf)

## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1; sf_use_s2() is TRUE

library(e1071)
if (!requireNamespace("astrochron", quietly = TRUE)) {
  install.packages("astrochron")
}
library(astrochron)
```

```

## Welcome to astrochron v1.2 (2023-08-25)

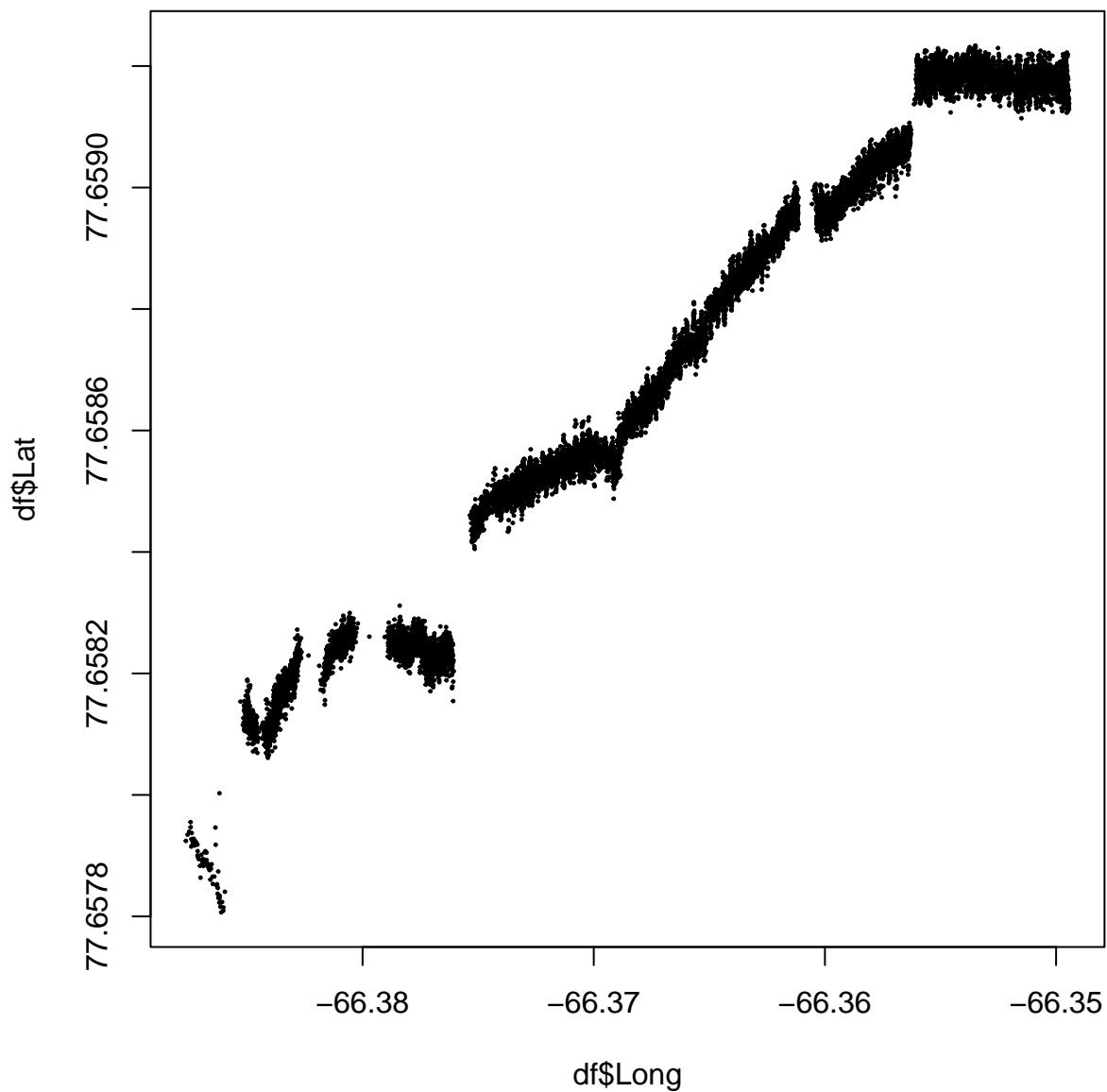
##
## Attaching package: 'astrochron'

## The following object is masked from 'package:e1071':
##
##      tune

## The following objects are masked from 'package:oce':
##
##      detrend, lowpass

#
df <- read.csv("DATA/EXB_csv.csv",sep=",",header=T)
parsed_date <- strptime(df$DateTime.UTC..YYMMDDHHmmss., format="%Y-%m-%dT%H:%M:%S")
# Convert to POSIXct
df$POSIX <- as.POSIXct(parsed_date) #, tz="UTC")
df <- df[,-3]
#
plot(df$Long,df$Lat,pch=19,cex=0.2)

```



```

# length of series in hours
len_h <- difftime(max(df$POSIX), min(df$POSIX), units = "hour")
# expected resolution
HWFM <- (1/as.numeric(len_h)) # in freq units
# width
half_width_24 <- (1/(1/24+HWFM) - 24)/2 # in period units (hours, here)
half_width_5p6 <- (1/(1/5.63+HWFM) - 5.63)/2 # in period units (hours, here)

```

lat/lon to northings/eastings

```

# Create a SpatialPoints object
points <- SpatialPoints(cbind(df$Long, df$Lat), proj4string = CRS("+proj=longlat +ellps=WGS84"))

# Define the target projection

```

```

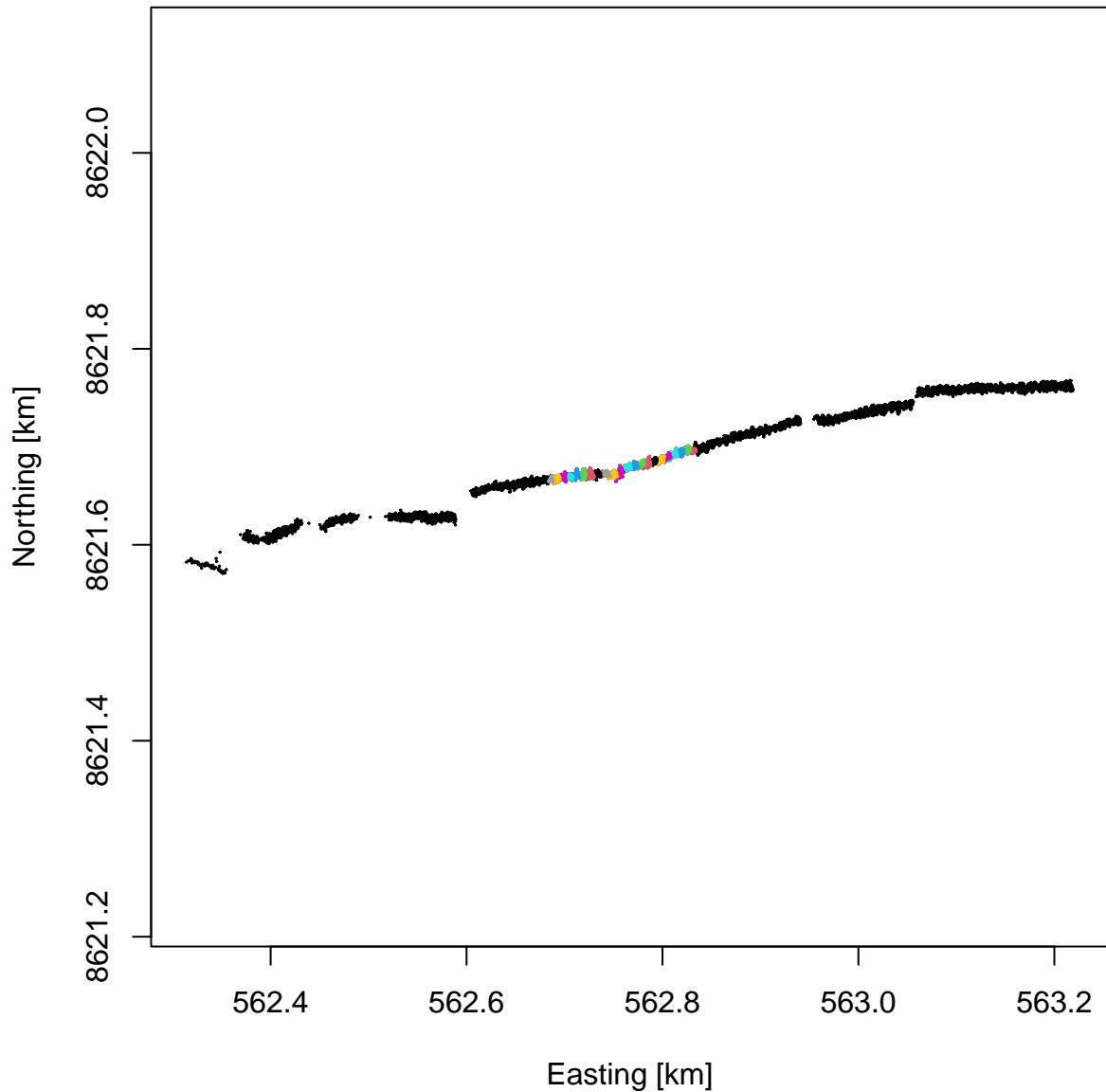
#target_projection <- "+proj=tmerc +lat_0=49 +lon_0=-2 +k=0.9996012717 +x_0=400000 +y_0=-100000 +ellps="
#target_projection <- "+proj=utm +zone=19X +ellps=WGS84 +datum=WGS84 +units=m +no_defs"
target_projection <- "+proj=utm +zone=19 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

# Transform the coordinates to the target projection
transformed_points <- spTransform(points, CRS(target_projection))

# Extract the eastings and northings
eastings <- transformed_points@coords[,1]
northings <- transformed_points@coords[,2]

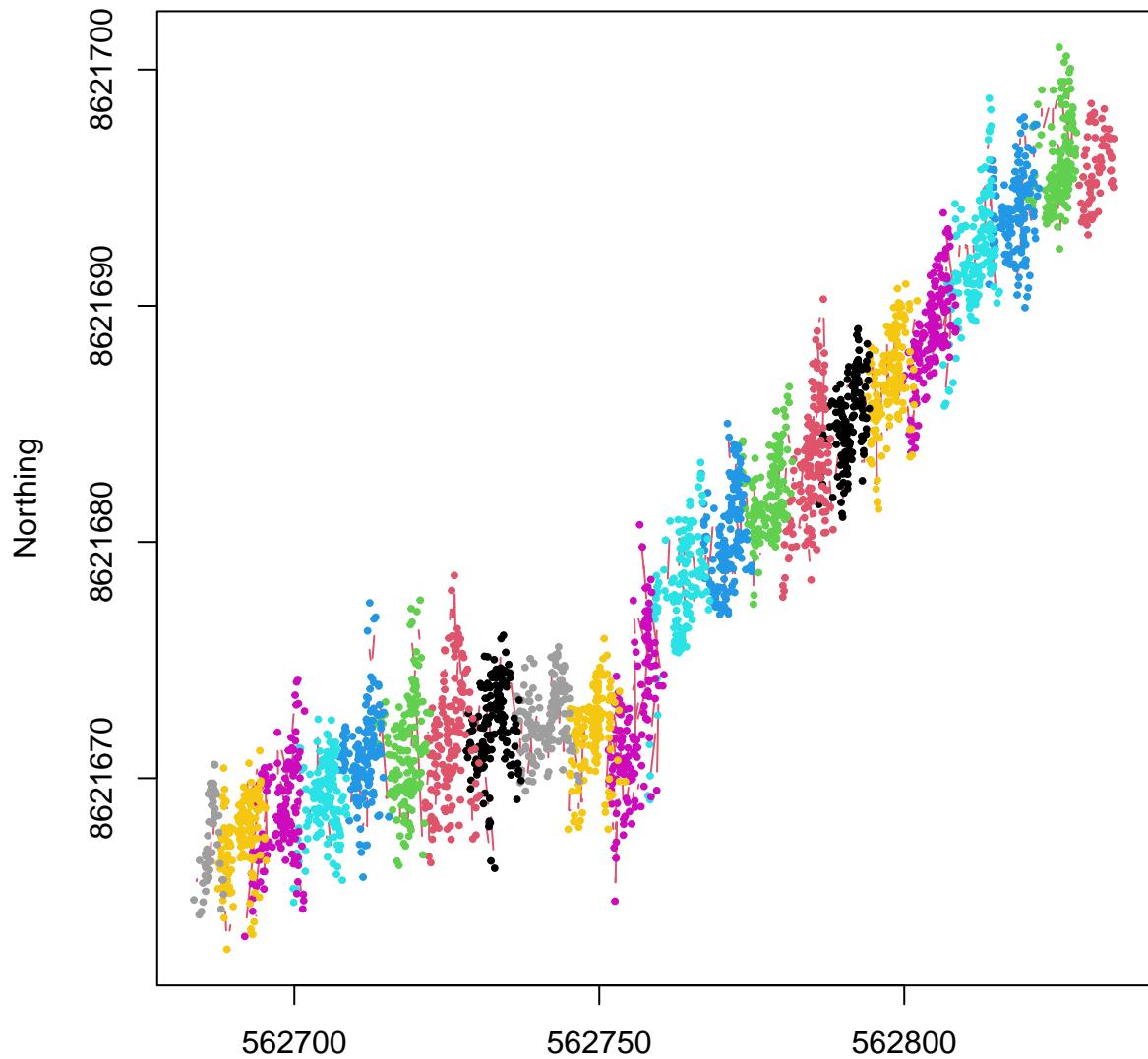
# Create a data frame with the eastings and northings
df$easting <- eastings
df$northings <- northings
plot(df$easting/1000,df$northings/1000,asp=1,xlab="Easting [km]",ylab="Northing [km]",pch=19,cex=0.1)
idx <- 8000:11000
points(df$easting[idx]/1000,df$northings[idx]/1000,pch=19,cex=0.1,col=day(df$POSIX))

```



```
plot(df$easting[idx], df$northings[idx], type="b", col=day(df$POSIX), pch=19, main="14 days - colour = day of
```

14 days – colour = day of month



Why the positive Easting?? Because UTM zone 19X has ‘its meridian’ at 108 degrees west
Note: There is a tendency for the points each day to start in the North and then as time passes (right to left) the points drift South.

calculate speeds between points observed

```
speed <- NULL
u <- NULL
v <- NULL
for (i in 1:(nrow(df)-1))
{
  d_easting <- df$easting[i+1]-df$easting[i]
```

```

d_northing <- df$northings[i+1] - df$northings[i]
d_easting <- (d_easting)
d_northing <- (d_northing)
dr <- sqrt((d_easting)^2 + (d_northing)^2)
dt <- as.numeric(df$POSIX[i+1] - df$POSIX[i])
speed <- c(speed, dr/dt)
u <- c(u, d_easting/dt)
v <- c(v, d_northing/dt)
}
speed <- c(speed, NA)
u <- c(u, NA)
v <- c(v, NA)
df$speed <- speed
df$u <- u
df$v <- v

```

plots

```

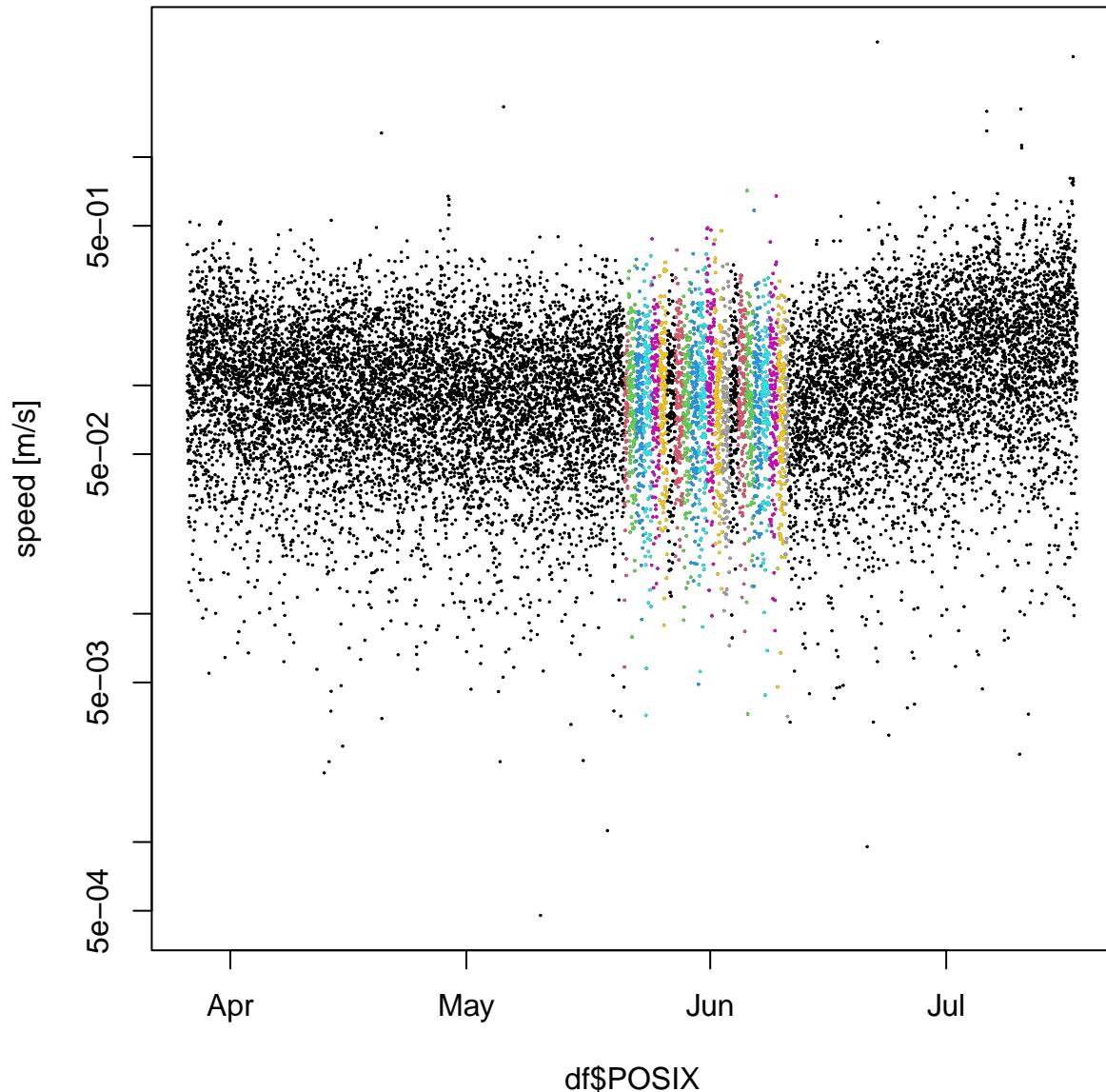
plot(df$POSIX, df$speed, pch=19, cex=0.1, main="EXB buoy at Qaanaaq 2024", ylab="speed [m/s]", log="y")

## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 y value <= 0 omitted from
## logarithmic plot

points(df$POSIX[idx], df$speed[idx], pch=19, cex=0.1, col=day(df$POSIX))

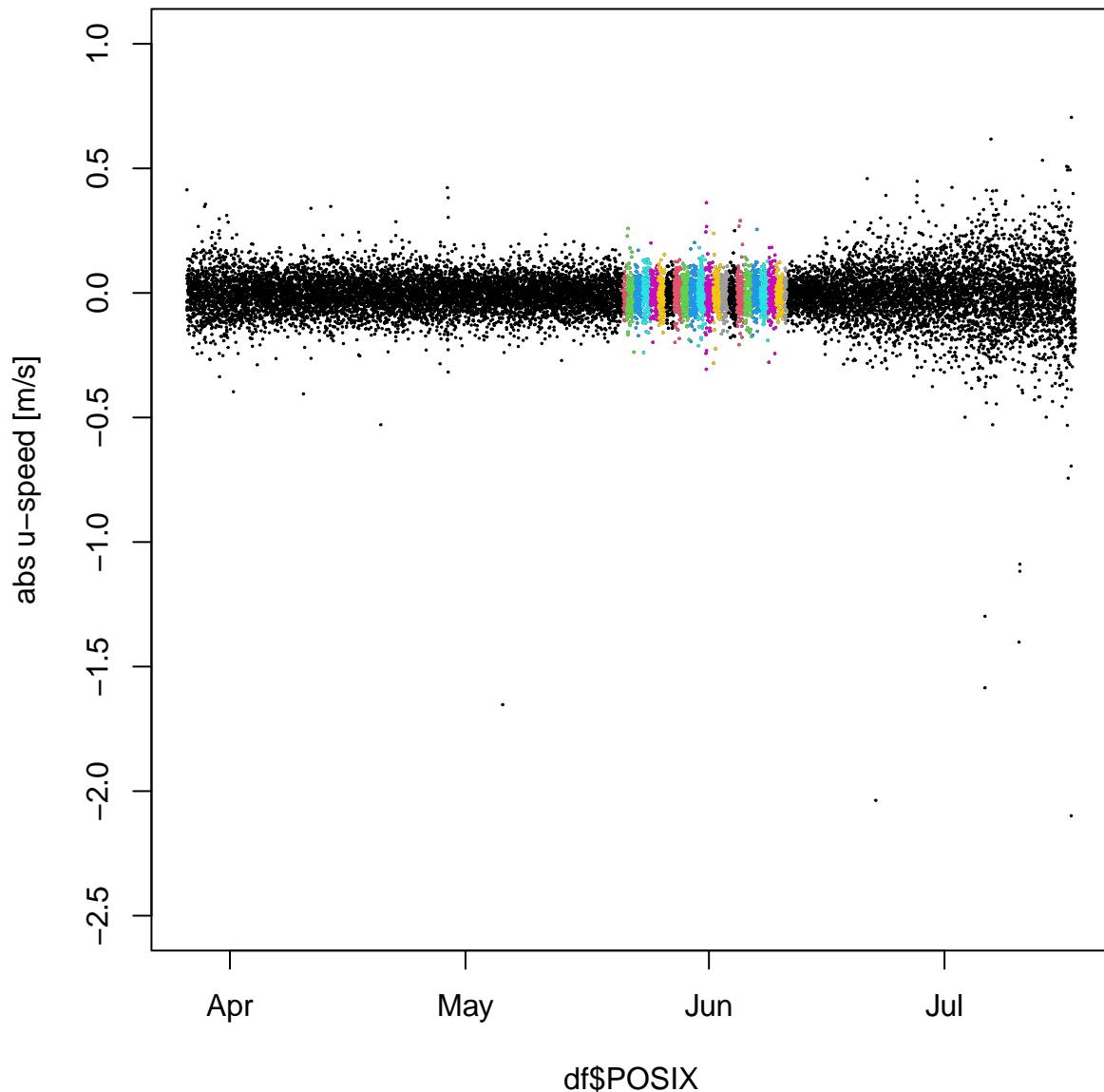
```

EXB bouy at Qaanaaq 2024

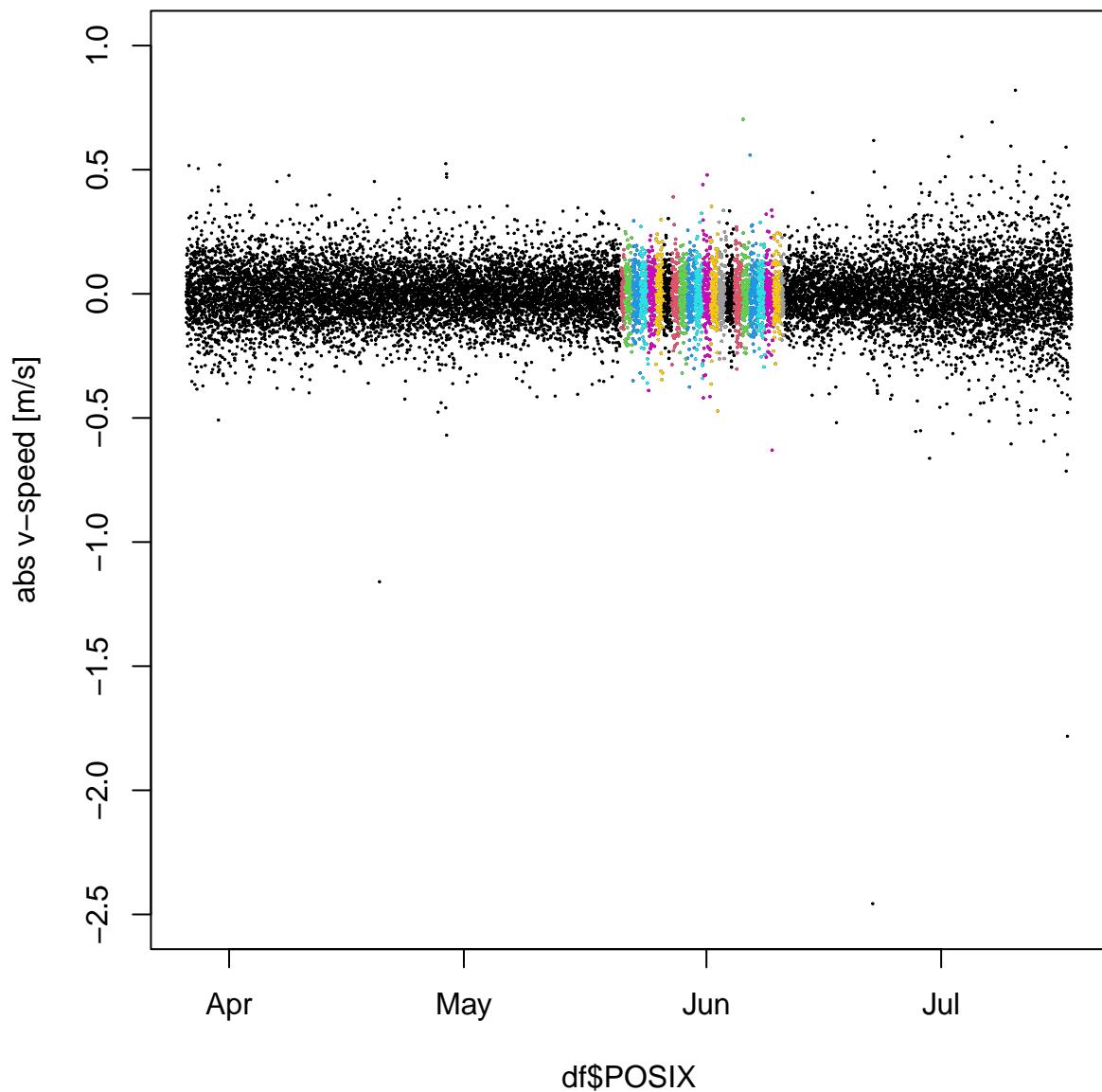


```
uv_ran <- c(-2.5,1)
plot(df$POSIX,df$u,pch=19,cex=0.1,main="EXB bouy at Qaanaaq 2024",ylab="abs u-speed [m/s]",ylim=uv_ran)
points(df$POSIX[idx],df$u[idx],pch=19,cex=0.1,col=day(df$POSIX))
```

EXB bouy at Qaanaaq 2024

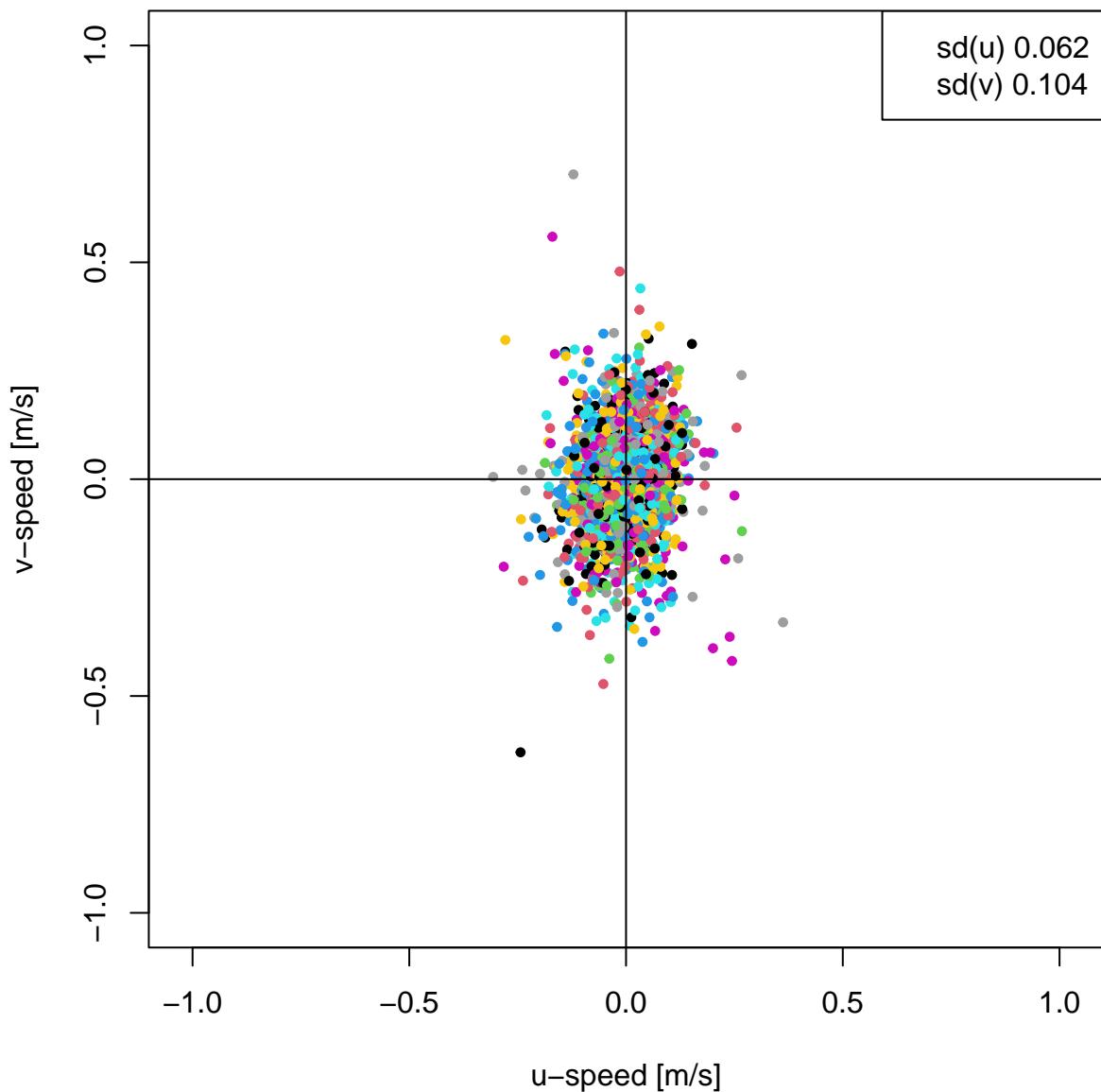


EXB bouy at Qaanaaq 2024



```
#  
plot(df$u[idx],df$v[idx],pch=19,cex=0.6,main="EXB bouy at Qaanaaq 2024",ylab="v-speed [m/s]",col=hour(d  
abline(h=0)  
abline(v=0)  
disp_u <- round(sd(df$u[idx]),3)  
disp_v <- round(sd(df$v[idx]),3)  
legend("topright",legend=c(paste("sd(u)",disp_u),paste("sd(v)",disp_v)))
```

EXB bouy at Qaanaaq 2024



Comment

The northing speed dispersion is greater than the easting dispersion - more

Scargle periodogram

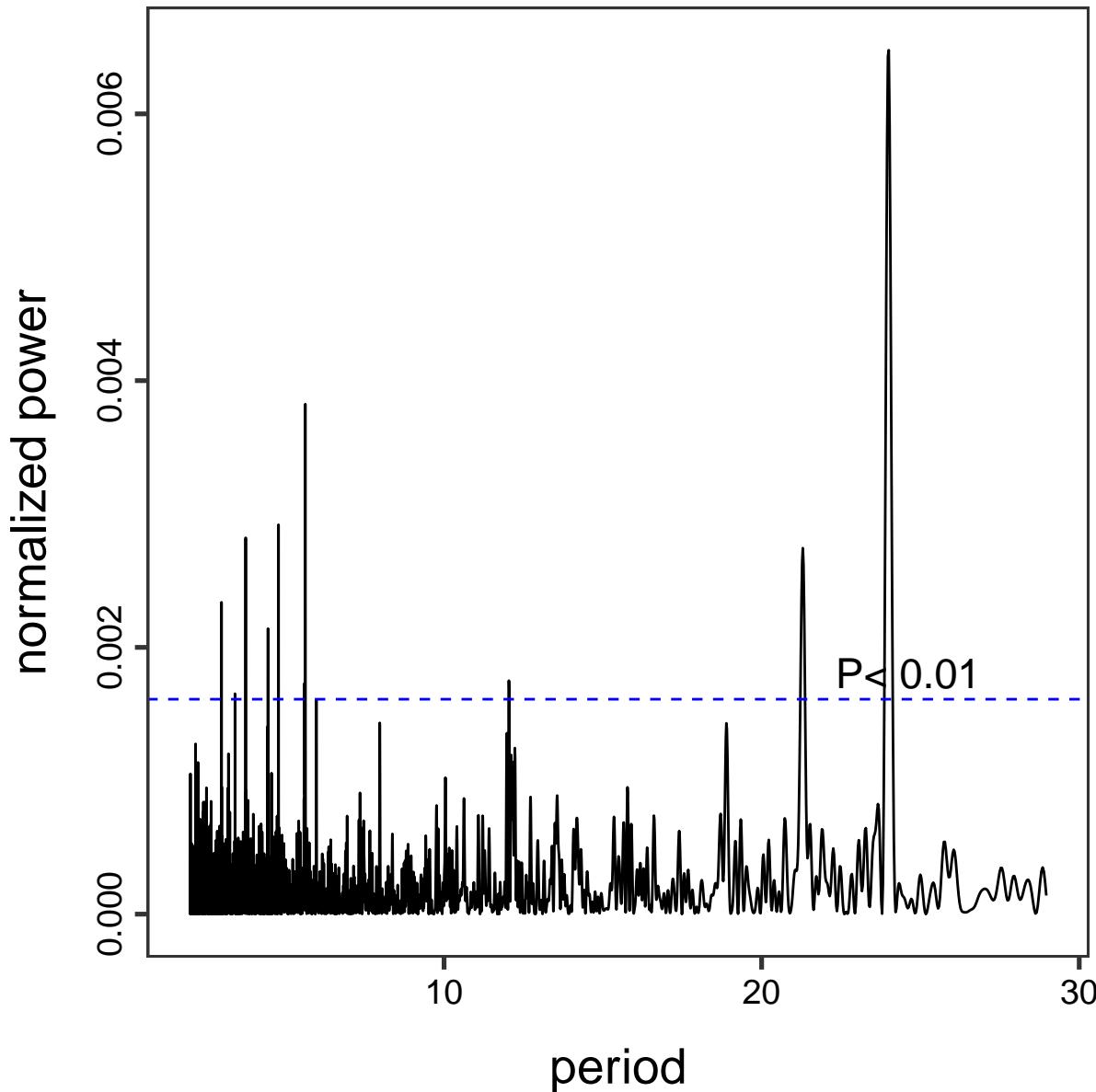
```
library(lomb)
df <- na.omit(df)
df$hours <- as.numeric(difftime(df$POSIX, min(df$POSIX), units = "hours"))
# speed
```

```

test_period_in_hours <- 15
test_signal <- sin(2*pi/test_period_in_hours*df$hours)*0.0 #0.005
# Compute Lomb-Scargle periodogram
ls_periodogram <- lsp(cbind(df$hours, df$speed-mean(df$speed)+test_signal), fit.sin = TRUE, type="period")

```

Lomb–Scargle Periodogram

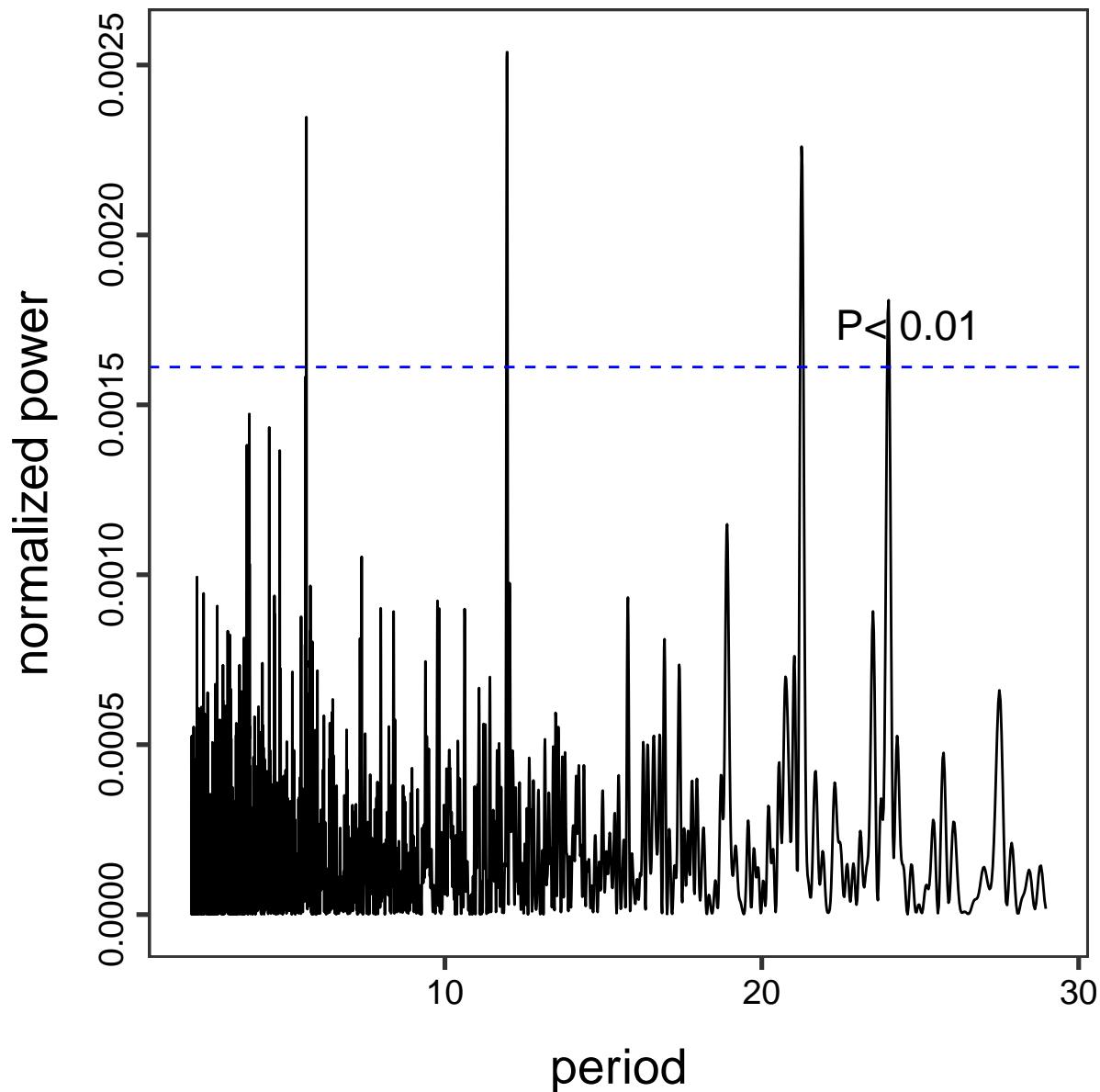


```

# u
# Compute Lomb-Scargle periodogram
ls_periodogram_u <- lsp(cbind(df$hours, abs(df$u-mean(df$u))), fit.sin = TRUE, type="period", from=2,

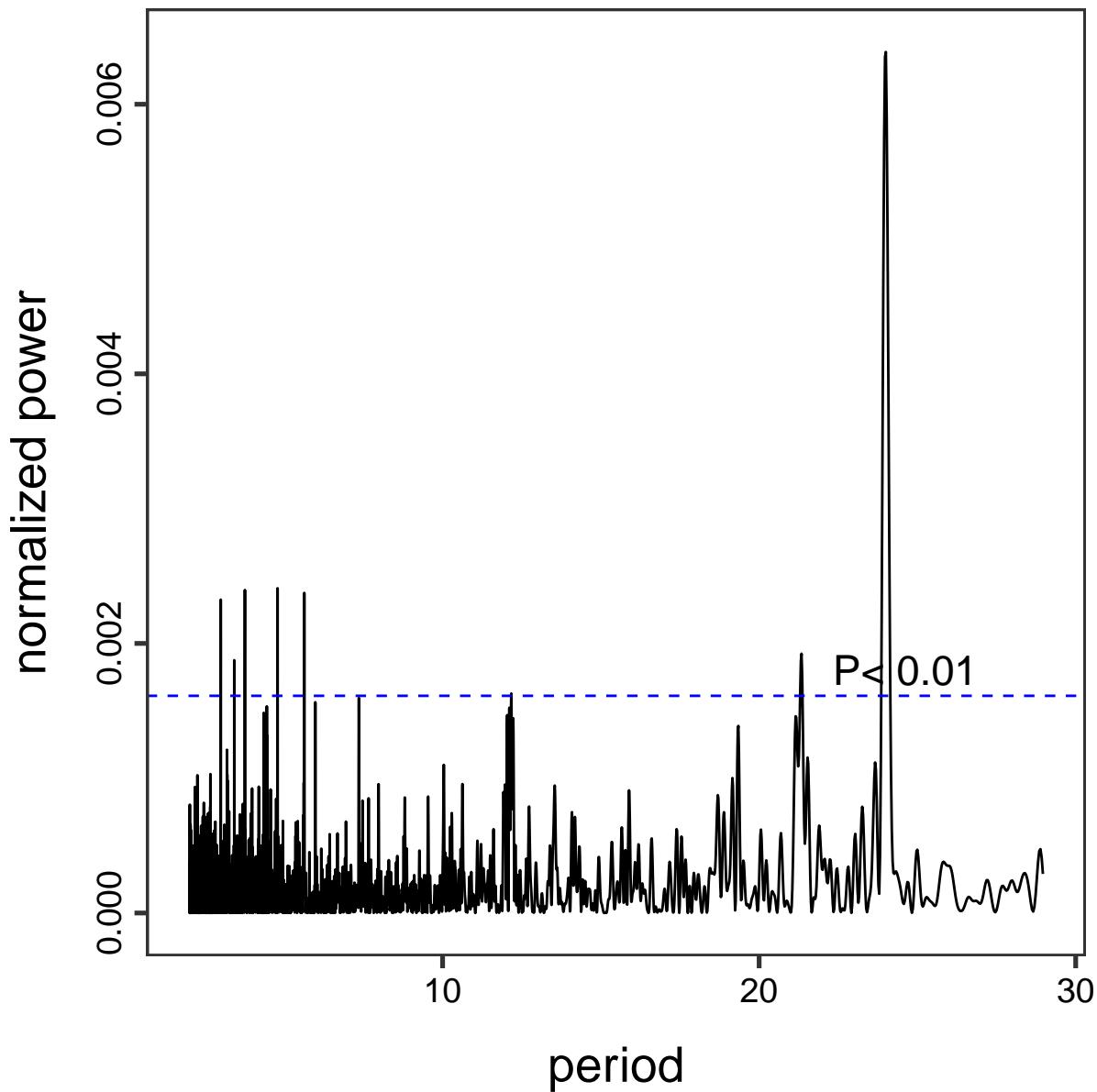
```

Lomb–Scargle Periodogram



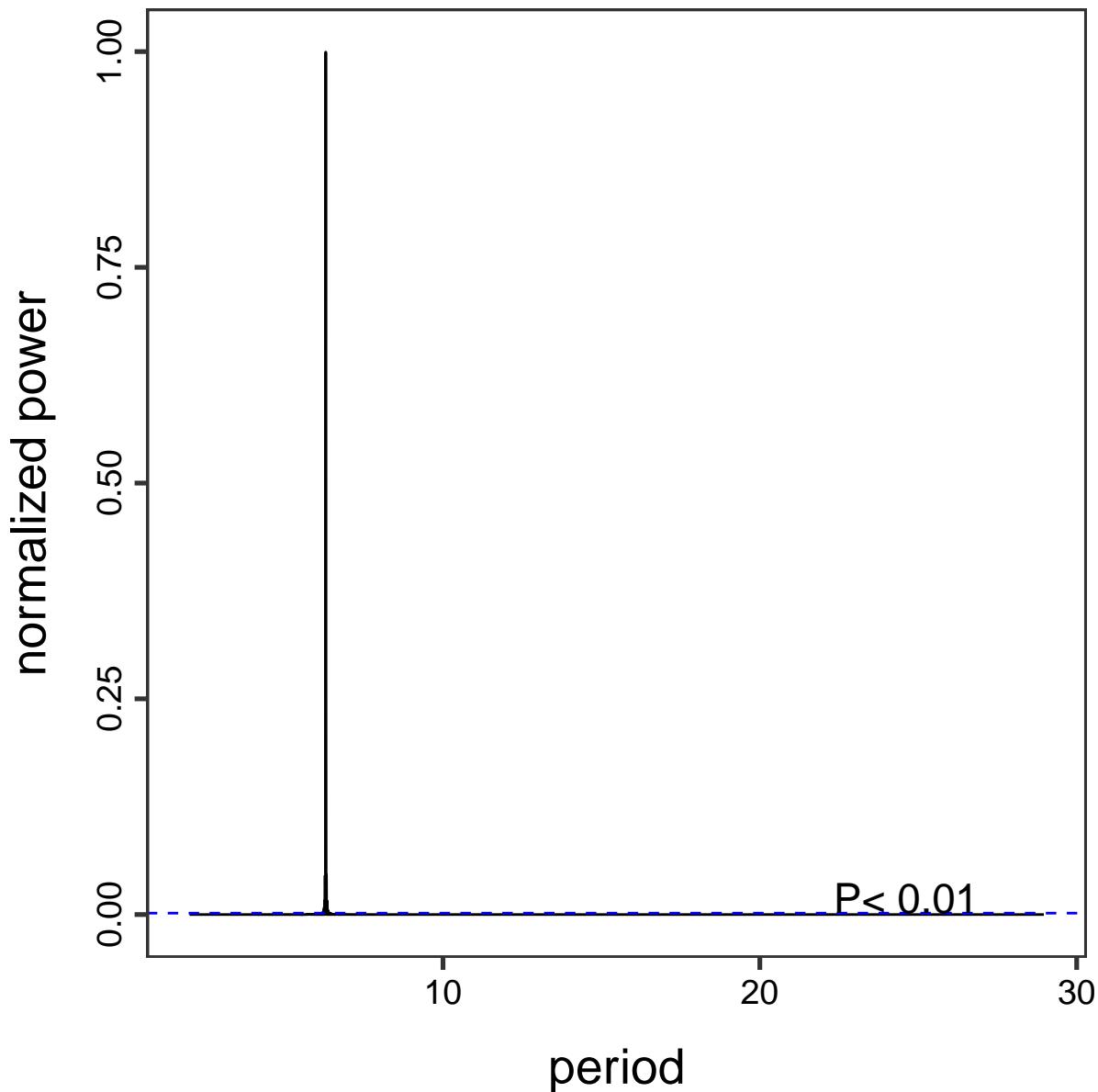
```
# v
# Compute Lomb-Scargle periodogram
ls_periodogram_v <- lsp(cbind(df$hours, abs(df$v - mean(df$v))), fit.sin = TRUE, type="period", from=2,
```

Lomb–Scargle Periodogram



```
# test signal
test_period_in_hours <- 6.3
test_signal <- sin(2*pi/test_period_in_hours*df$hours)*0.0001
ls_periodogram_test <- lsp(cbind(df$hours, test_signal), fit.sin = TRUE, type="period", from=2, to = 30)
```

Lomb–Scargle Periodogram



plots

```
pdf("FIGURES/spectra.pdf")
par(mfrow=c(3,1))
plot(ls_periodogram$scanned, ls_periodogram$power, type = 'l', xlab = 'Period [hours]', ylab = 'Power in')
#abline(v=test_period_in_hours,col=2,lwd=1)
#abline(v=24,col=2,lwd=1)
#abline(v=12,col=2,lwd=1)
#abline(v=12.41666,col=4,lty=3,lwd=3)
#abline(v=24/3,col=2,lwd=1)
```

```

#abline(v=24/4,col=2,lwd=1)
#abline(v=24/5,col=2,lwd=1)
lines(ls_periodogram_u$scanned, ls_periodogram_u$power,col=2,lwd=2)
lines(ls_periodogram_v$scanned, ls_periodogram_v$power,col=4,lwd=2)
abline(h=ls_periodogram$sig.level,col="green",lty=3,lwd=2)
legend("top",legend="black : speed, red : u, blue : v")

plot(ls_periodogram$scanned, ls_periodogram$power, type = 'l', xlab = 'Period [hours]', ylab = 'Power in')
abline(v=test_period_in_hours,col=2,lwd=1)
abline(v=24,col=2,lwd=1)
abline(v=12,col=2,lwd=1)
abline(v=12.41666,col=4,lty=3,lwd=3)
abline(h=ls_periodogram$sig.level,col="green",lty=3,lwd=2)
legend("topleft",legend="M2 marked with blue stippled")
lines(ls_periodogram_u$scanned, ls_periodogram_u$power,col=2,lwd=2)
lines(ls_periodogram_v$scanned, ls_periodogram_v$power,col=4,lwd=2)

plot(ls_periodogram$scanned, ls_periodogram$power, type = 'l', xlab = 'Period [hours]', ylab = 'Power in')
abline(v=24+half_width_24,lty=3)
abline(v=24-half_width_24,lty=3)
abline(v=test_period_in_hours,col=2,lwd=1)
abline(v=24,col=2,lwd=1)
abline(v=12,col=2,lwd=1)
abline(v=12.41666,col=4,lty=3,lwd=3)
abline(v=21.2895,col=4,lty=3,lwd=3)
abline(h=ls_periodogram$sig.level,col="green",lty=3,lwd=2)
legend("top",legend="24 and 21.2895 hours")
lines(ls_periodogram_u$scanned, ls_periodogram_u$power,col=2,lwd=2)
lines(ls_periodogram_v$scanned, ls_periodogram_v$power,col=4,lwd=2)

plot(ls_periodogram$scanned, ls_periodogram$power, type = 'l', xlab = 'Period [hours]', ylab = 'Power in')
abline(v=5.6299+half_width_5p6,lty=3)
abline(v=5.6299-half_width_5p6,lty=3)

abline(v=test_period_in_hours,col=2,lwd=1)
abline(v=5.6299,col=2,lwd=1)
abline(h=ls_periodogram$sig.level,col="green",lty=3,lwd=2)
lines(ls_periodogram_u$scanned, ls_periodogram_u$power,col=2,lwd=2)
lines(ls_periodogram_v$scanned, ls_periodogram_v$power,col=4,lwd=2)
legend("topleft",legend=c("vertical : 5.6299 hours","u red, v blue, speed black"))

# spectrum of test signal
plot(ls_periodogram_test$scanned, ls_periodogram_test$power, type = 'l', xlab = 'Period [hours]', ylab =
dev.off()

## pdf
## 2

```

Comments

There is a strong speed signal at 24 hours - mainly due to a strong northing speed. This implies a north-south oscillating motion.

consider the phase

```
# Example data (replace this with your actual time series)
#set.seed(123)

time_points <- difftime(df$POSIX, min(df$POSIX), units = "hour")
regular_time_grid <- seq(min(time_points), max(time_points), length.out = (max(time_points)-min(time_points)))

interpolated_values <- approx(x = time_points, y = df$speed, xout = regular_time_grid)$y

# Compute Lomb-Scargle periodogram with phase using astrochron
lsp_result <- periodogram(
  cbind(regular_time_grid, interpolated_values),
  demean=T,
  detrend=T,
  background=1,
  bc=1,
  pl=2,
  verbose=T,
  genplot=T,
  output=1
)

## -----
## ----- CALCULATING PERIODOGRAM FOR STRATIGRAPHIC SERIES -----
## * Number of data points in stratigraphic series: 16292
## * Stratigraphic series length (space or time): 2715.333
## * Sampling interval (space or time): 0.1666769
## * Mean value removed= 0.1184243
## * Linear trend removed. m= 1.880493e-05 b= -0.02553083
## * Nyquist frequency: 2.999816
## * Rayleigh frequency: 0.0003682563
## * Padded to 32584 points
## * Estimated AR1 coefficient = 0.4829095

# Extract frequency, amplitude, and phase
frequency <- lsp_result$Frequency
amplitude <- lsp_result$Amplitude
power <- lsp_result$Power
phase <- lsp_result$Phase

# Plot the Lomb-Scargle periodogram
plot(1/frequency, power, type = "b", xlab = "1/Frequency", ylab = "Power", col = "blue", lty = 1, xlim=
```

```
# Optionally, plot the phase information
plot(1/frequency, phase, type = "b", xlab = "1/Frequency", ylab = "Phase", col = "red", lty = 1, xlim=c
```

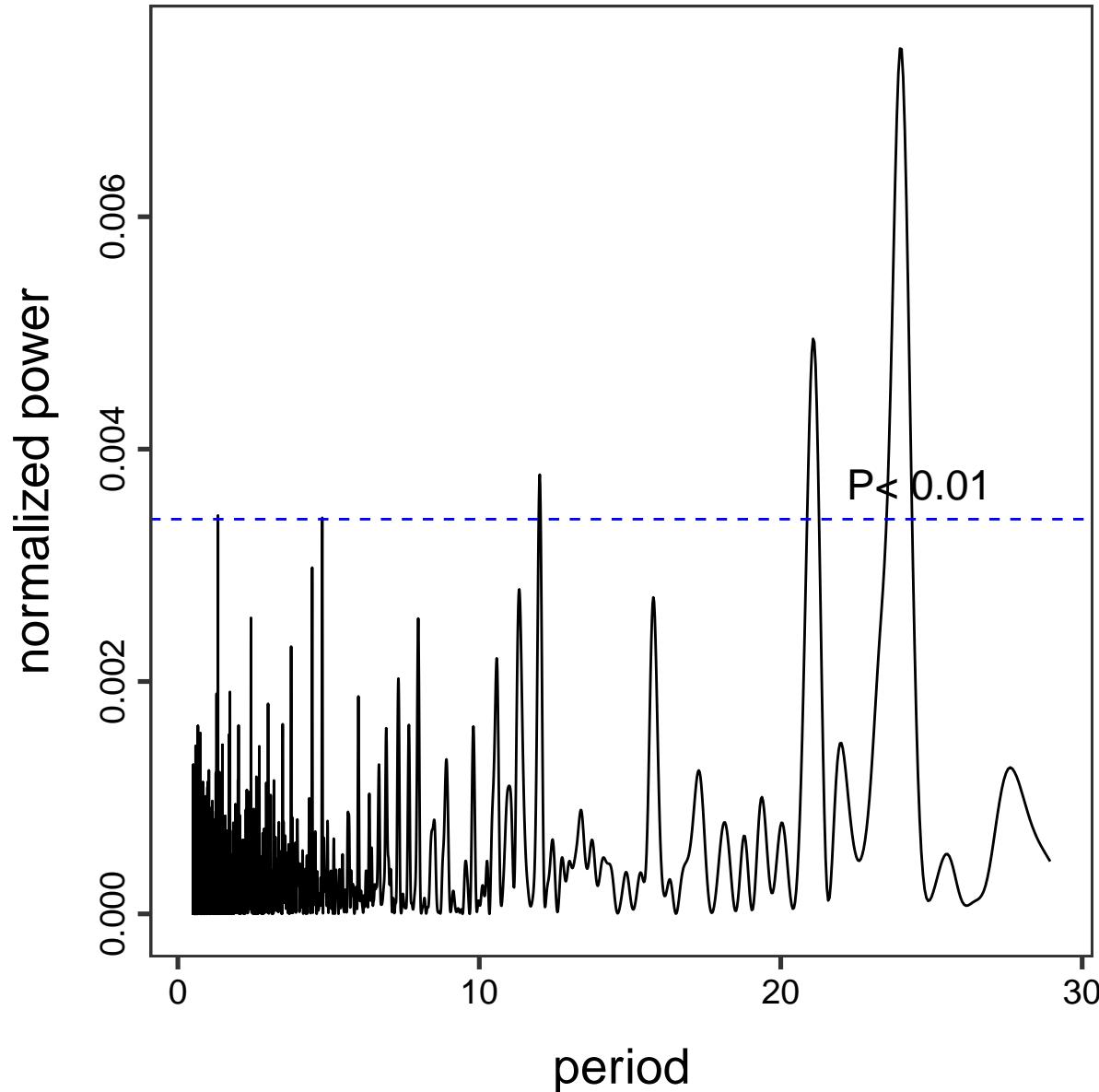
Define function to perform Lomb-Scargle with user options

```
lsp_pth <-
  function(x_orig,
          y_orig,
          type_str = 'period',
          ofac = 1,
          ifwindow = 'none',
          from = .5,
          to = 29,
          if_test = FALSE,
          test_signal_period=5.5)
{
  # echo inputs
  print(paste("length of x and y :", length(x_orig), length(y_orig)))
  # type_str : 'period' or 'frequency'
  x <- x_orig
  y <- y_orig - mean(y_orig, na.omit = T)
  # apply window
  if (ifwindow == 'Hanning') {
    win <- hanning.window(length(y_orig))
    y <- y * win
  }
  # add test_signal if required
  if (if_test){
    testsignal <- 0.01*sin(2*pi/test_signal_period*x)
    y <- y + testsignal
    print(paste("Added a test signal at period ",test_signal_period))
  }
  lsp_out <-
    lsp(
      cbind(x, y),
      fit.sin = TRUE,
      type = type_str,
      from = from,
      to = to,
      ofac = ofac
    )
  return(list(
    'freq_or_period' = lsp_out$scanned,
    'power' = lsp_out$power
  ))
}
```

calculation

```
idx <- 1:(nrow(df)/2)
type_str <- 'period'
out1 <- lsp_pth(df$hours[idx], df$speed[idx], type_str, ofac=8, if_test=F, test_signal_period=20, ifwindow="T")
## [1] "length of x and y : 8140 8140"
```

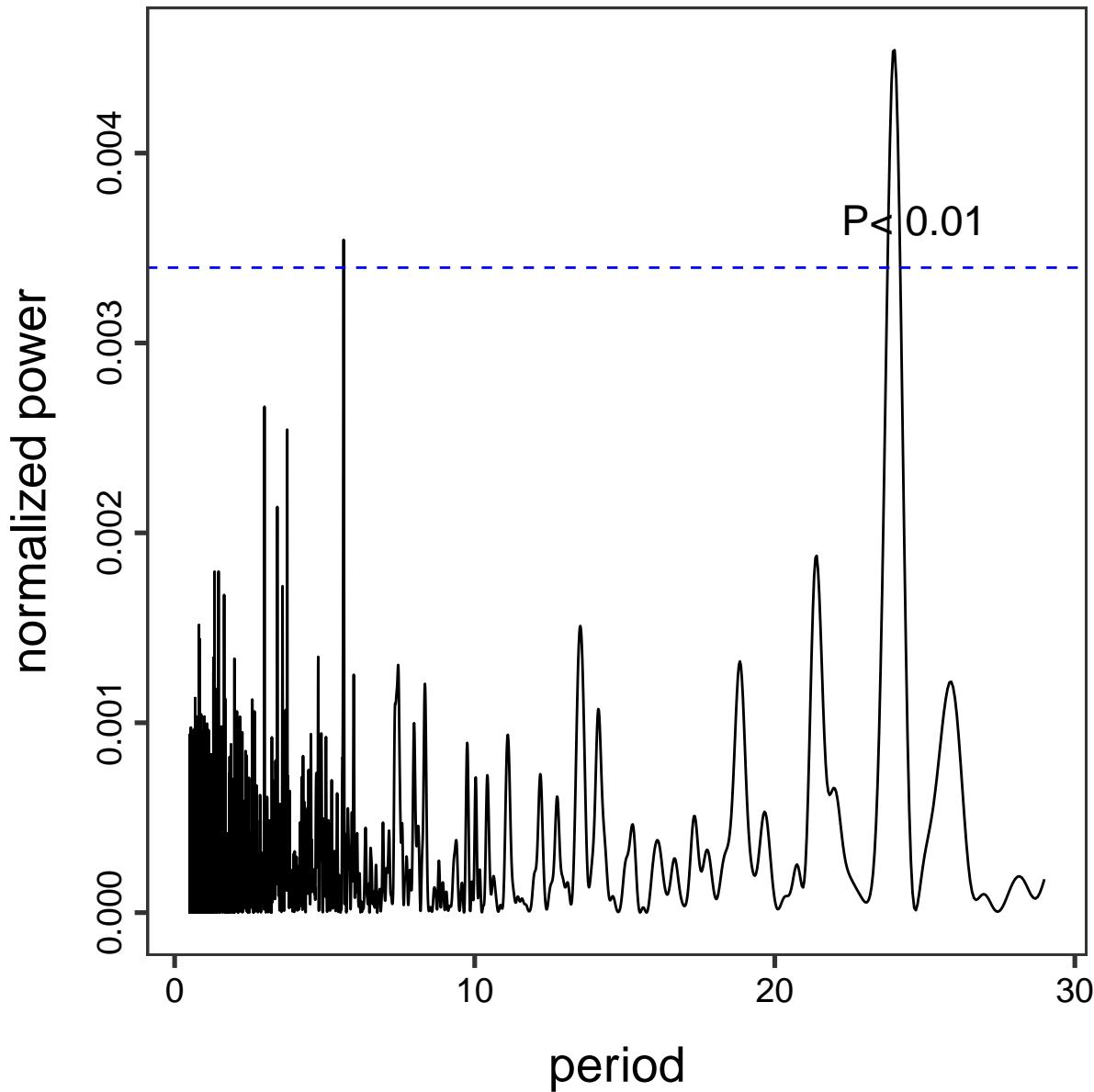
Lomb–Scargle Periodogram



```
#  
idx <- (nrow(df)/2):nrow(df)  
out2 <- lsp_pth(df$hours[idx], df$speed[idx], type_str, ofac=8, if_test=F, test_signal_period=20, ifwindow="T")
```

```
## [1] "length of x and y : 8141 8141"
```

Lomb–Scargle Periodogram



```
#
```

Plots

```
xlab_str <- "f [1/hours]"
if (type_str == 'period'){xlab_str='Period [hours]'}
#
pdf("FIGURES/spectra_2.pdf")
```

```

par(mfrow=c(2,1))
plot(out1$freq_or_period,out1$power,col='red',type="l",xlab=xlab_str,ylab="Power",main="First and second")
lines(out2$freq_or_period,out2$power,col='blue',type="l",lwd=3,xlim=c(10,25))
abline(v=12.41666,col=4,lty=3,lwd=3)
abline(v=24,col=4,lty=3,lwd=3)
#
plot(out1$freq_or_period,out1$power,col='red',type="l",xlab=xlab_str,ylab="Power",main="First and second")
lines(out2$freq_or_period,out2$power,col='blue',type="l",lwd=3,xlim=c(5,11))
#
plot(out1$freq_or_period,out1$power,col='red',type="l",xlab=xlab_str,ylab="Power",main="First and second")
lines(out2$freq_or_period,out2$power,col='blue',type="l",lwd=1.7,xlim=c(2,6))
#
plot(out1$freq_or_period,out1$power,col='red',type="l",xlab=xlab_str,ylab="Power",main="First and second")
lines(out2$freq_or_period,out2$power,col='blue',type="l",lwd=1.7,xlim=c(.5,2.1))
dev.off()

## pdf
## 2

```