Assignment #4

Peyman Kor

11/28/2019

#4.1

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts -------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
data <- read.csv('A4_Kulhuse.csv')
head(data)
```

```
##    Temp   Sal Depth   pH  Chl ODOsat  ODO Battery            DateTime
## 1 18.22 18.03 5.191 8.25 2.91  108.6 9.19    12.7 2017-08-24 10:00:00
## 2 18.30 17.96 5.213 8.26 4.11  107.7 9.10    12.7 2017-08-24 10:30:00
## 3 18.29 18.00 5.234 8.26 4.01  108.5 9.17    12.7 2017-08-24 11:00:00
## 4 18.30 18.00 5.258 8.26 4.33  108.8 9.20    12.7 2017-08-24 11:30:00
## 5 18.34 18.02 5.277 8.26 3.74  110.1 9.29    12.7 2017-08-24 12:00:00
## 6 18.35 18.04 5.282 8.26 4.45  110.9 9.36    12.7 2017-08-24 12:30:00
```

Now, let's have alook on the data for possible "NA" values,

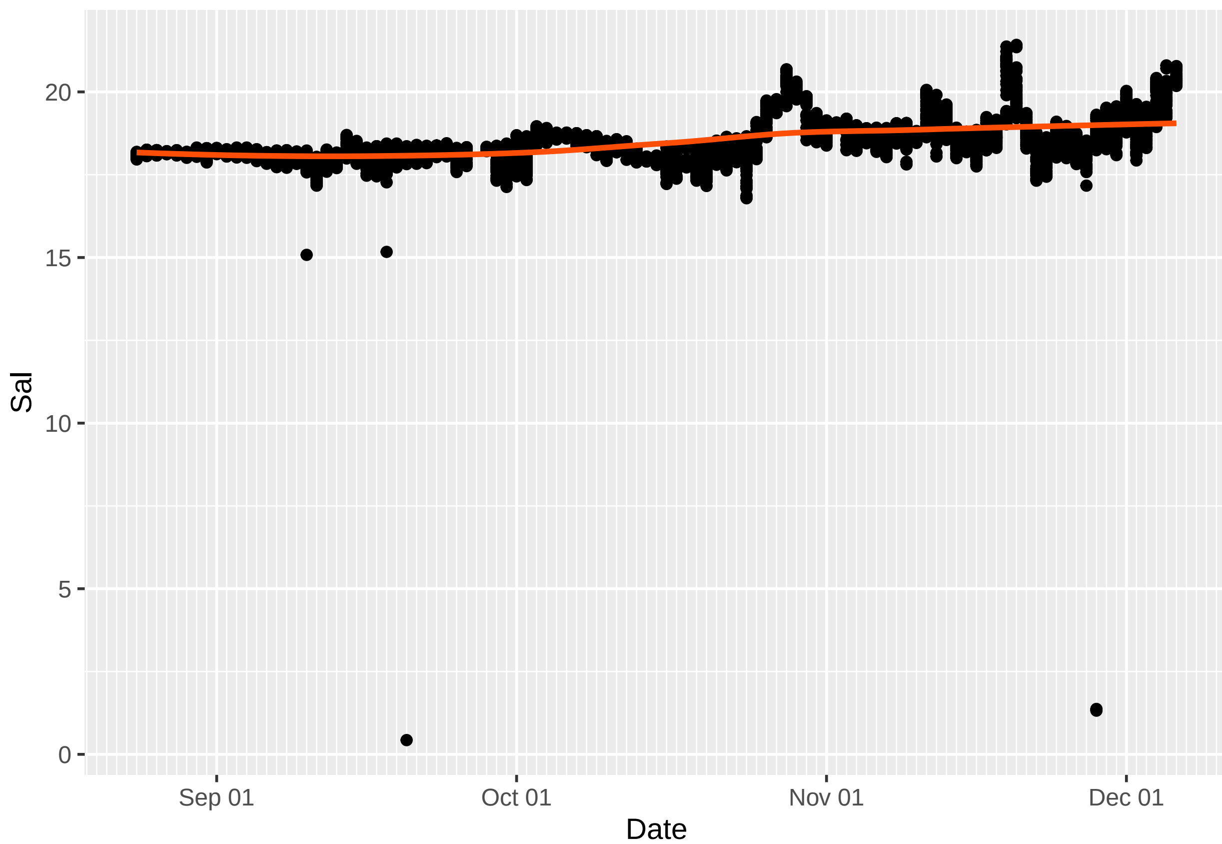
```r
data_fil <- data %>% filter(is.na(Sal))
```

We see that the column Salinity has 111 rows whcih are NA, also we could see other columns as well are NA, so could safely removes these 111 rows beacuse they contain no information.

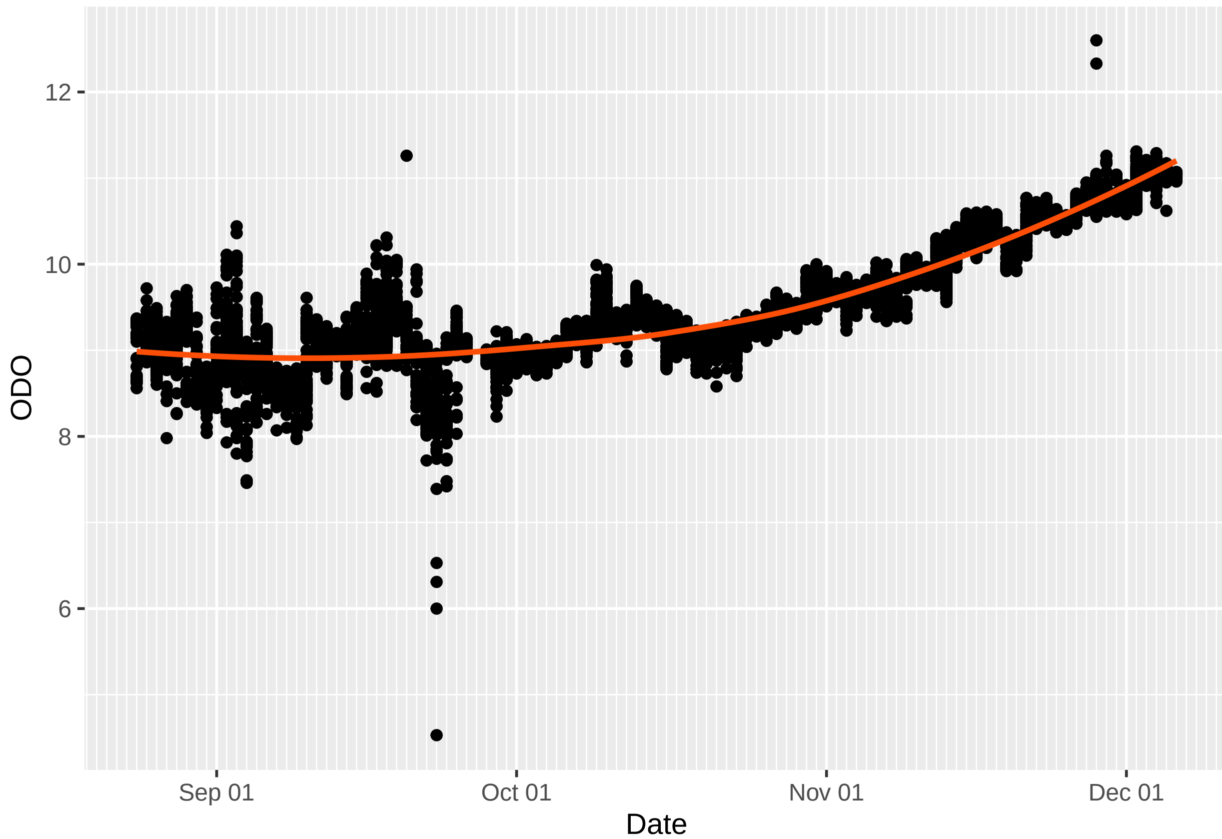
```r
data_nona <- data %>% filter(!is.na(Sal))
```

Having this now, we visulize the Salinity versus time:

```r
library(ggplot2)
data_nona %>% mutate(Date = as.Date(DateTime)) %>%
  ggplot(mapping = aes(Date,Sal)) +
  geom_point() +
  scale_x_date(date_minor_breaks = "1 day ",date_labels = "%b %d") +
  stat_smooth(
  color = "#FC4E07",
  method = "loess"
  )
```




```r
library(ggplot2)
data_nona %>% mutate(Date = as.Date(DateTime)) %>%
  ggplot(mapping = aes(Date,ODO)) +
  geom_point() +
  scale_x_date(date_minor_breaks = "1 day ",date_labels = "%b %d") +
  stat_smooth(
  color = "#FC4E07",
  method = "loess"
  )
```



4.2

The salinity assumed to be random walk process as the follow:

$$X_t = X_{t-1} + \eta_t$$
$$Y_t = X_t + \epsilon_t$$

So, the following is the state space model for salinity, where the η_t and ϵ_t are the white noise with the standard deviation of σ_η and σ_ϵ.

On the other hand, the salinity could be written as the ARIMA process (0,1,1) as the follow:

$$Y_t - Y_{t-1} = \eta_t + \epsilon_t - \epsilon_{t-1}$$

In the state space model as the matrix,

$$X_t = AX_{t-1} + G\epsilon_t$$
$$Y = CX_t$$

Where the $A = C = 1$

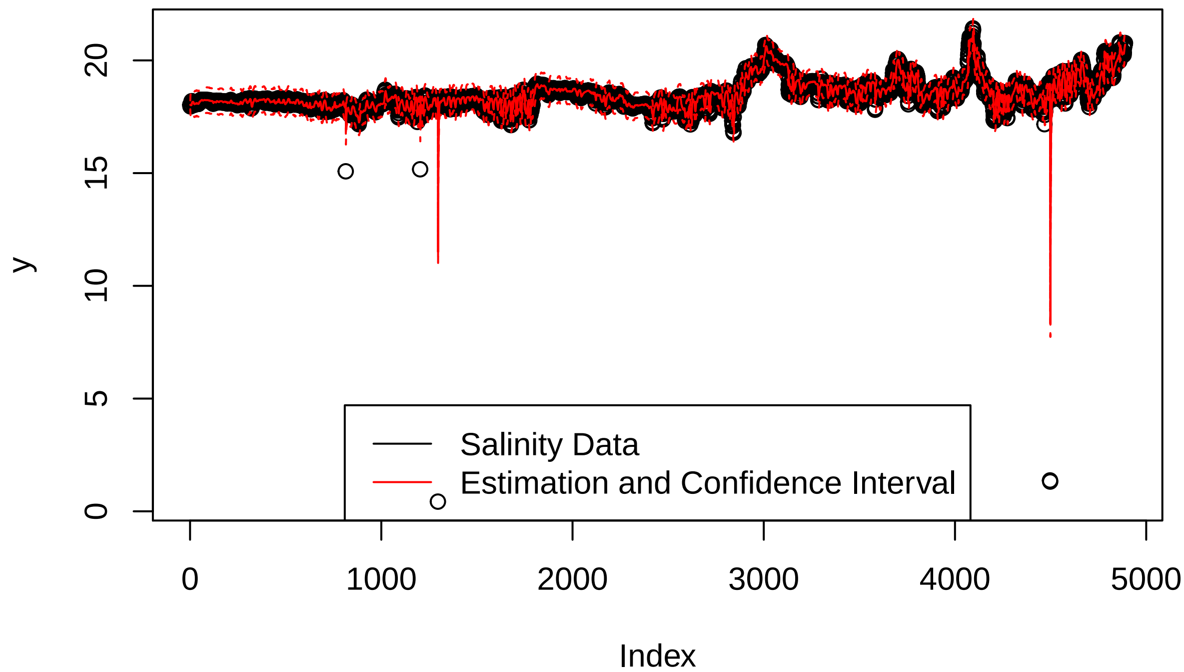
4.3

```r
library(FKF)
```

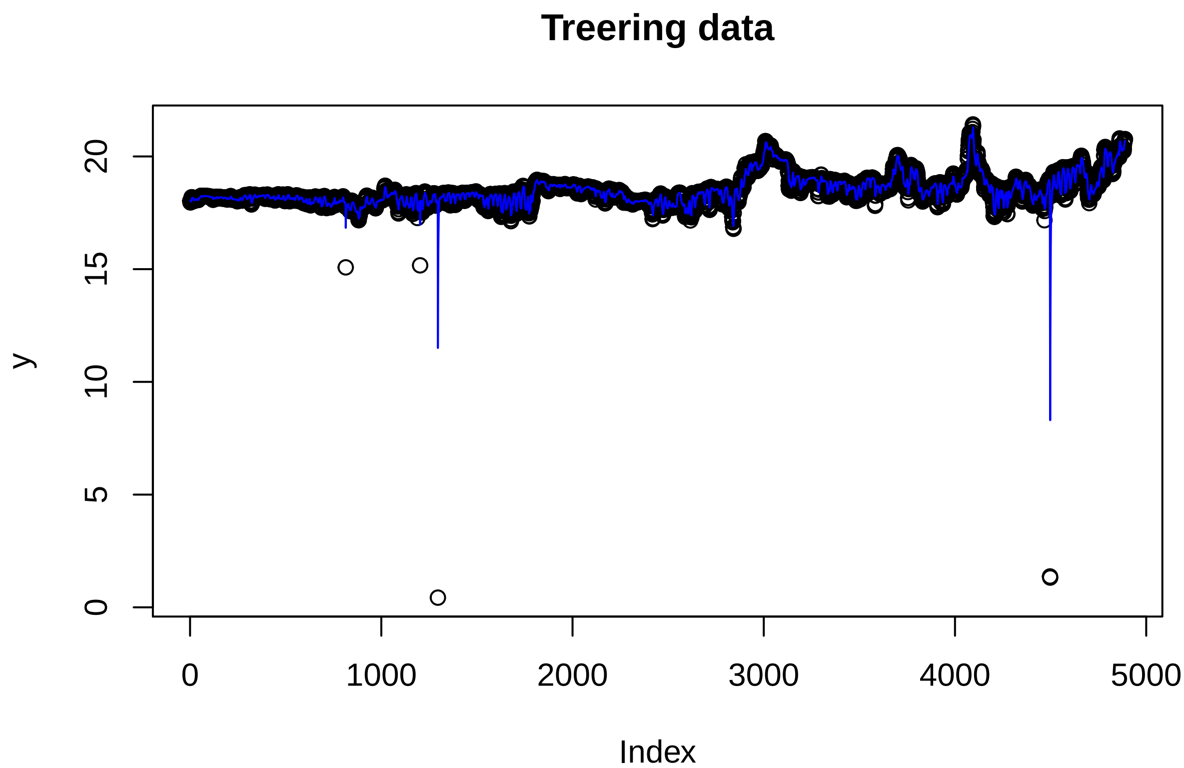
```
## Loading required package: RUnit
```

```r
y <- data_nona$Sal
dt <- ct <- matrix(0)
Zt <- Tt <- matrix(1)
a0 <- y[1] # Estimation of the first year flow
P0 <- matrix(0.01) # Variance of 'a0'
fit.fkf <- optim(c(HHt = 0.01 ,
                   GGt = 0.005 ),
                 fn = function(par, ...)
                   -fkf(HHt = matrix(par[1]), GGt = matrix(par[2]), ...)$logLik,
                 yt = rbind(y), a0 = a0, P0 = P0, dt = dt, ct = ct,
                 Zt = Zt, Tt = Tt)
## Filter Nile data with estimated parameters:plot.fkf 7
fkf.obj <- fkf(a0, P0, dt, ct, Tt, Zt, HHt = matrix(fit.fkf$par[1]),
               GGt = matrix(fit.fkf$par[2]), yt = rbind(y))
```

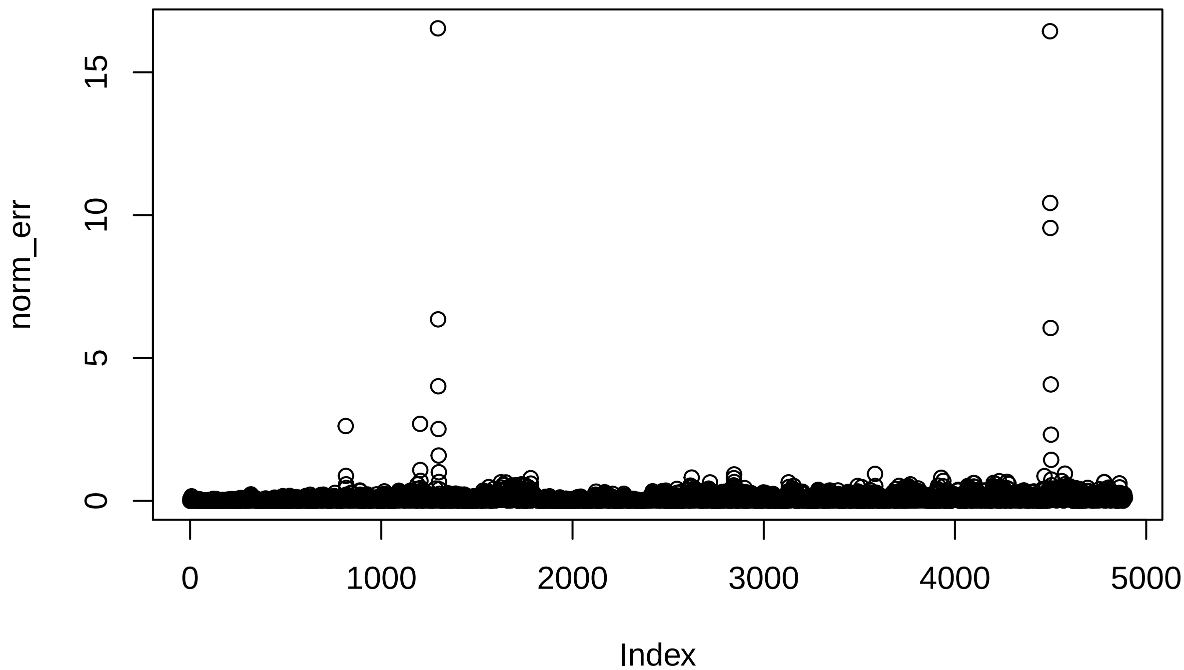
```r
plot(y)
with(fkf.obj, matlines((at[1,]) + cbind(0,-1.96*sqrt(Pt[1,1,]),1.96*sqrt(Pt[1,1,])),type="l", lty=c(1,2
legend("bottom", c("Salinity Data", "Estimation and Confidence Interval"),
       col = c("black", "red"), lty = 1)
```



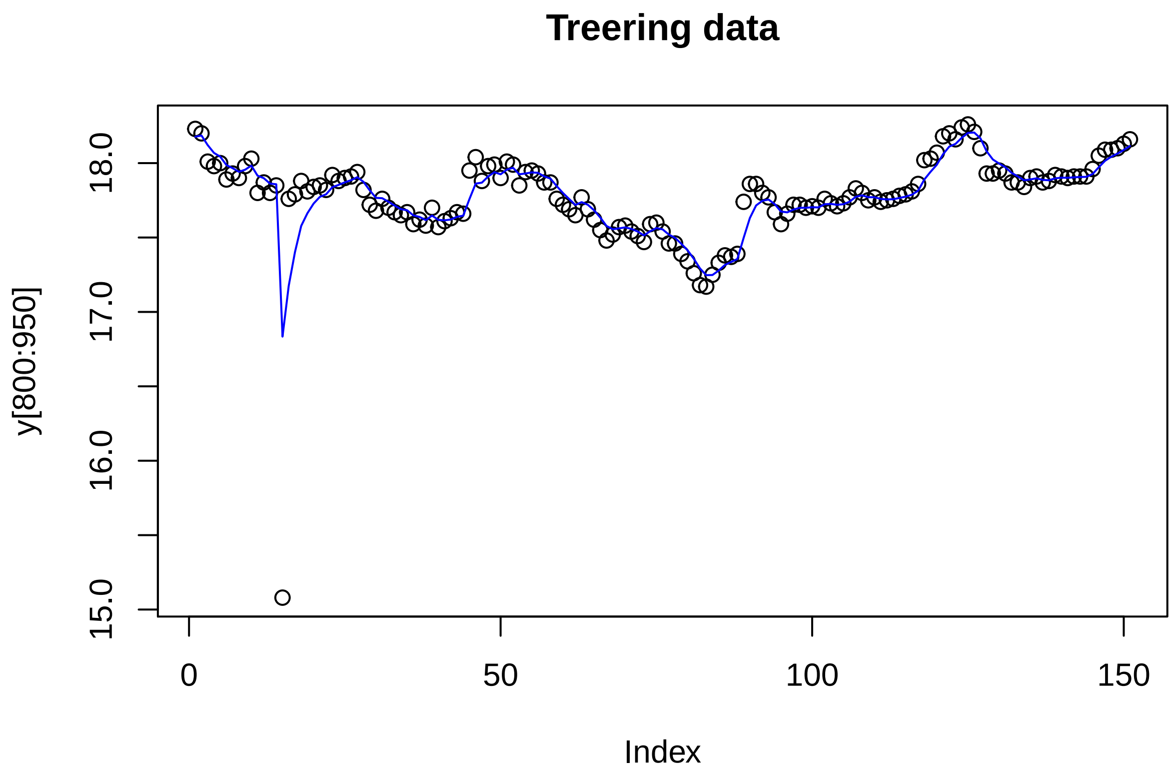

```
plot(y, main = "Treering data")
lines(ts(fkf.obj$att[1, ], start = start(y), frequency = frequency(y)), col = "blue")
```



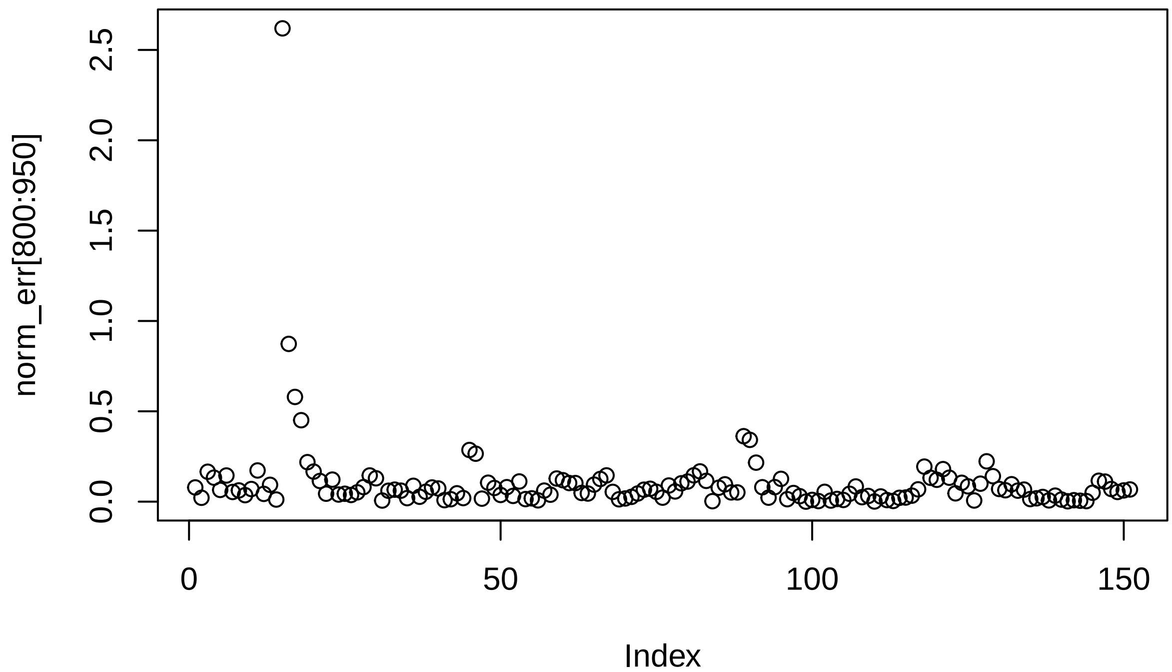

```
prediction <- fkf.obj$att[1, ]
error <- abs(y-prediction)
norm_err <- error/sd(prediction)
plot(norm_err)
```




```
plot(y[800:950], main = "Treering data")
lines(ts(fkf.obj$att[1, ][800:950], start = start(y), frequency = frequency(y)), col = "blue")
```




```
prediction <- fkf.obj$att[1, ]
error <- abs(y-prediction)
norm_err <- error/sd(prediction)
plot(norm_err[800:950])
```




```
prediction[4889]
```

```
## [1] 20.69859
```

4

```
sd_pre <- sd(prediction)
sum(abs(prediction-y) > 6*sd_pre)
```

```
## [1] 6
```

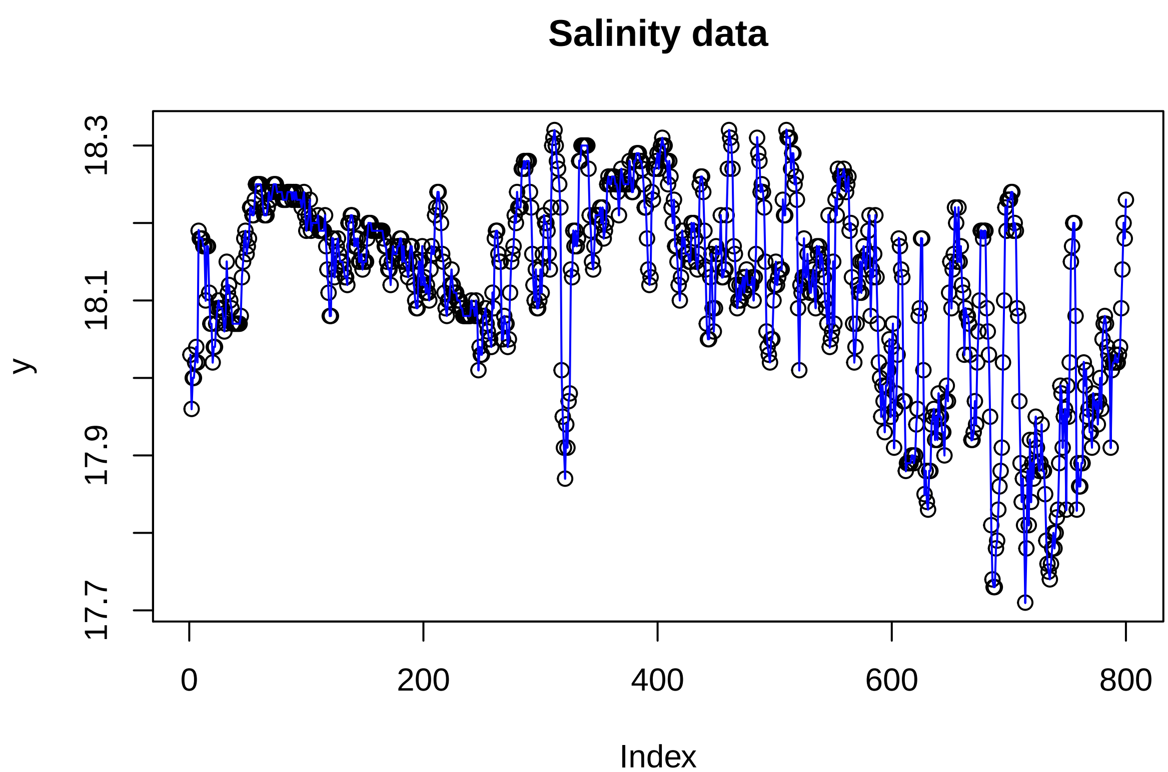
5

```
y <- data_nona$Sal[1:800]
dt <- ct <- matrix(0)
Zt <- Tt <- matrix(1)
a0 <- y[1] # Estimation of the first year flow
P0 <- matrix(0.01) # Variance of 'a0'
fit.fkf <- optim(c(HHt = 0.0 ,
```

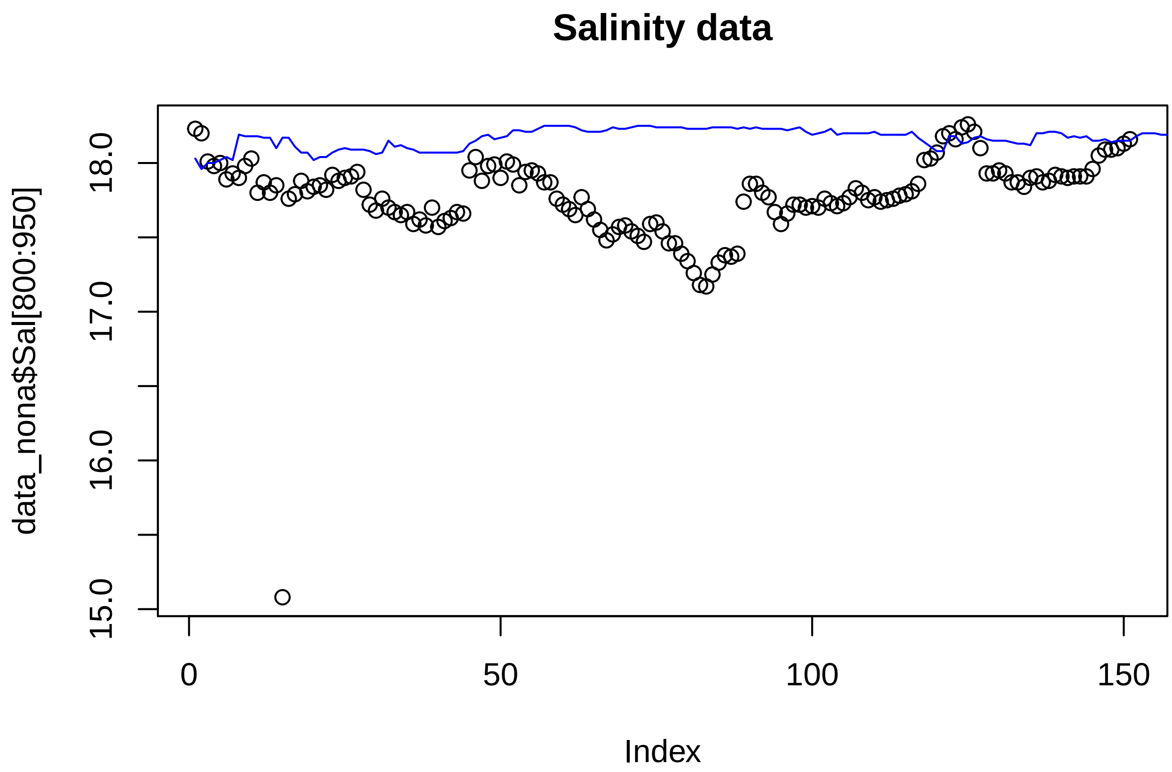
```
                      GGt = 0.005 ),
                    fn = function(par, ...)
                      -fkf(HHt = matrix(par[1]), GGt = matrix(par[2]), ...)$logLik,
                    yt = rbind(y), a0 = a0, P0 = P0, dt = dt, ct = ct,
                    Zt = Zt, Tt = Tt)

fit.fkf$par
```

```
##           HHt           GGt
##  1.910121e-03 -1.175588e-05
```

```
fkf.obj1 <- fkf(a0, P0, dt, ct, Tt, Zt, HHt = matrix(fit.fkf$par[1]),
                GGt = matrix(fit.fkf$par[2]), yt = rbind(y))
plot(y, main = "Salinity data")
lines(ts(fkf.obj1$att[1, ], start = start(y), frequency = frequency(y)), col = "blue")
```




```
plot(data_nona$Sal[800:950], main = "Salinity data")
lines(ts(fkf.obj1$att[1, ], start = start(y), frequency = frequency(y)), col = "blue")
```


Salinity data
data_nona$Sal[800:950]
18.0
17.0
16.0
15.0
0
50
100
150
Index