SuperMAG

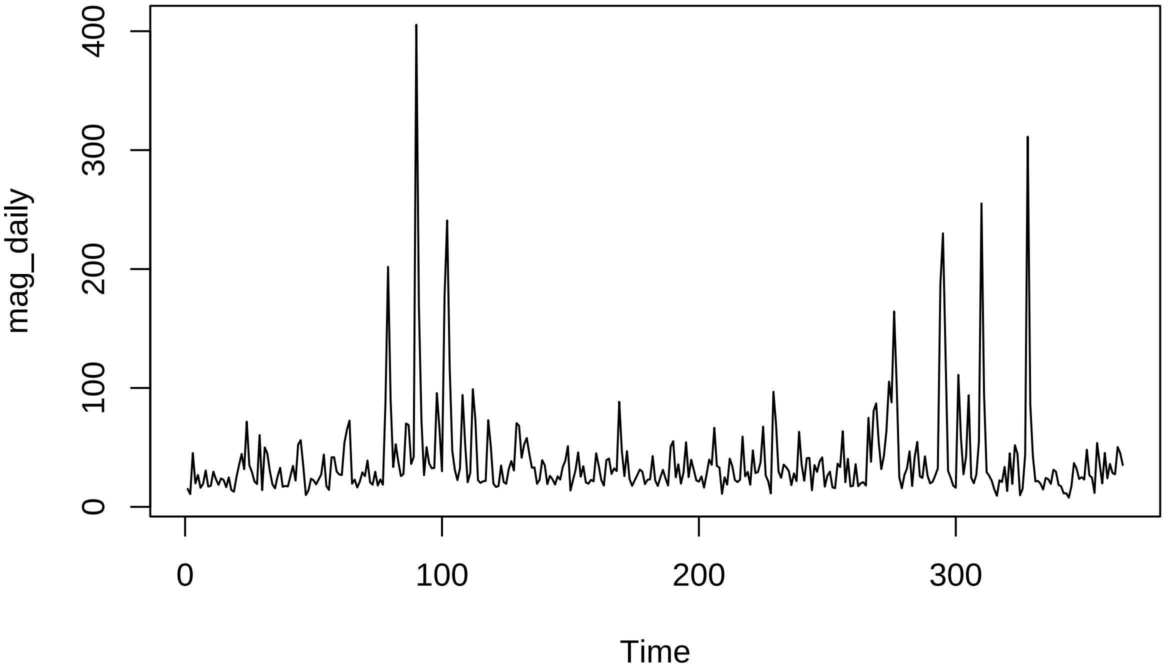
```r
library(ggplot2)
library(HDInterval)
library(dspCP)
library(tidyr)

max_abs = function(x){
  x = na.omit(x)
  if(length(x) == 0) return (0)
  else{
    return(x[which.max(abs(x))])
  }
}

rsq <- function (x, y) cor(x, y) ^ 2

daily_pooling = function(df, varname, func)
{
  y = as.vector(df[, varname])
  y = matrix(y, ncol = 24, byrow = T)
  y = apply(y, 1, func) # max pooling
  return(y)
}
```

Data Preprocessing

```r
path = "/Users/hejimi/MPS-Project/SuperMAG-data/frd2001_samples.csv"
tks = strsplit(path, "/", fixed=T)[[1]]
prefix = toupper(substr(tks[length(tks)],1,7))
df = read.table(path, sep = ',', header = T)
mat = as.matrix(df[, c("dbn_nez","dbn_geo","dbe_nez","dbe_geo","dbz_nez","dbz_geo" )])
mag = apply(mat[, c(1,3,5)], function(x)sqrt(sum(x^2)), MARGIN = 1) # Calculate the magnitude
mat_mag = matrix(mag, ncol = 24, byrow = T)
mag_daily = apply(mat_mag, 1, max_abs) # max pooling
log_mag_daily = log(mag_daily)


# ## Aggregated Data
# path = "/Users/hejimi/MPS-Project/SuperMAG-data/us_agg_2001.csv"
# tks = strsplit(path, "/", fixed=T)[[1]]
# prefix = toupper(substr(tks[length(tks)],1,7))
# df = read.table(path, sep = ',', header = T)
# mat = as.matrix(df[, c("dbn_nez","dbe_nez","dbz_nez" )])
# mag = apply(mat, function(x)sqrt(sum(x^2)), MARGIN = 1) # Calculate the magnitude
# mat_mag = matrix(mag, ncol = 24, byrow = T)
# mag_daily = apply(mat_mag, 1, max_abs) # max pooling
# log_mag_daily = log(mag_daily)
```

Visualize daily max magnitude

```
ts.plot(mag_daily)
```



Call ABCO on single direction

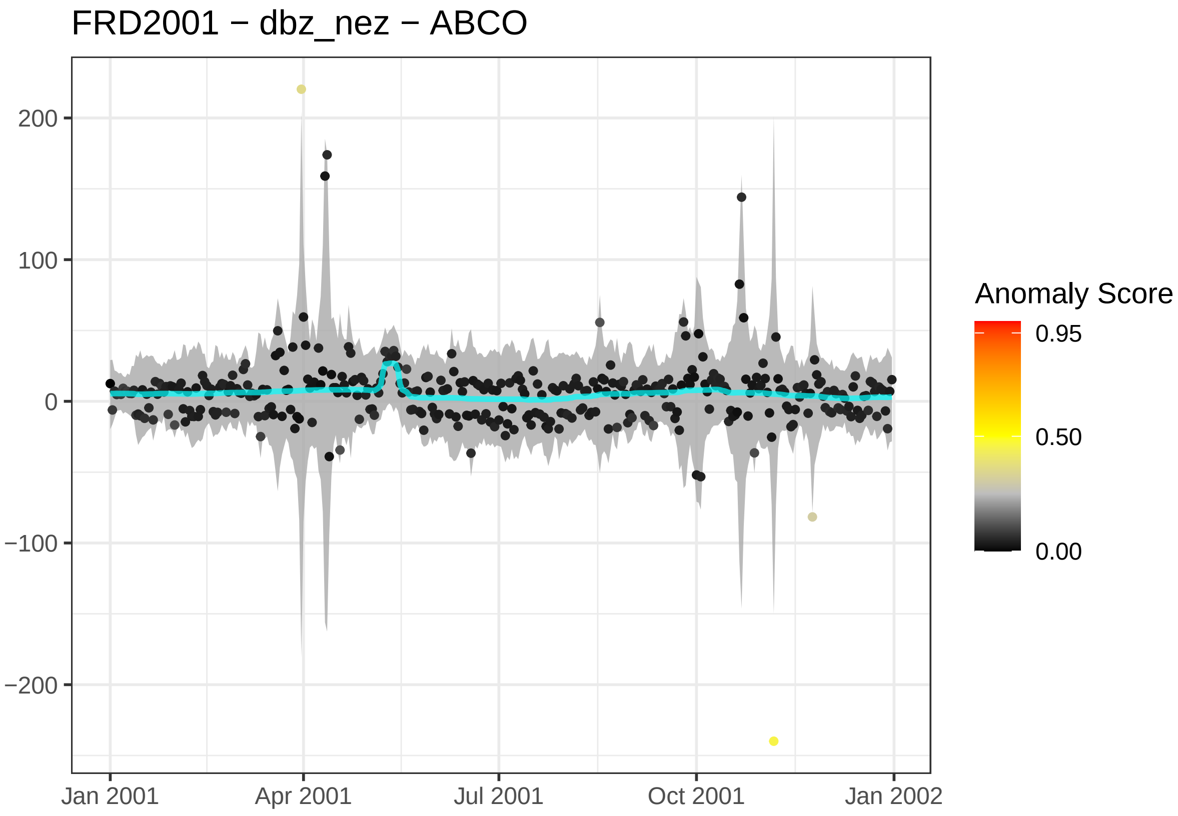
```
var_name = "dbz_nez" # select direction
y = daily_pooling(df, var_name, max_abs)


# Call ABCO
mcmc_output = dsp_cp(y, cp=TRUE, useAnom = TRUE, mcmc_params = list('yhat', 'mu', "omega", "r", 'obs_sig
```

```
## [1] "7.77 seconds remaining"
## [1] "4.29 seconds remaining"
## [1] "3.54 seconds remaining"
## [1] "2.57 seconds remaining"
## [1] "1.69 seconds remaining"
## [1] "0.85 seconds remaining"
## [1] "0 seconds remaining"
## [1] "Total time:  5 seconds"
```

```
cp = mcmc_output$cp

cred_int_1 = apply(mcmc_output$yhat, 2, hdi, credMass=0.95)
itr = dim(mcmc_output$zeta_sigma_t2)[1]
score = colMeans(
  cbind(
    rep(0, itr),
    mcmc_output$zeta_sigma_t2
  ) /
  (
    cbind(
      rep(0, itr),
      mcmc_output$zeta_sigma_t2
    )+mcmc_output$obs_sigma_t2
  )
)
mp1 = data.frame(y = as.matrix(y), x = seq(1, length(y)), mn = colMeans(mcmc_output$mu),
                 upper=cred_int_1[2,], lower = cred_int_1[1,], score=score)
```

Visualization of ABCO

```
  pl <- ggplot(data = mp1, aes(as.Date("2001-01-01")+x-1, y))+
    geom_ribbon(data=mp1,aes(ymin=lower,ymax=upper),alpha = 0.8, fill="darkgray") +
    geom_point(size = 1.0, aes(color = score))+
    scale_colour_gradientn(limits=c(0,1),colours=c("black", "gray", "yellow", "orange", "red"), breaks=
    labs(color = "Anomaly Score") +
    geom_line(data=mp1, aes(as.Date("2001-01-01")+x-1, mn), col="cyan", size=1.0, alpha = 0.7)+
    xlab("")+ylab("")+
    theme_bw()+
    # theme(legend.position="top")+
    ggtitle(paste(prefix,"-", var_name,"-","ABCO"))
  pl
```



Call ABCO on three directions

```
mcmc_output_n = dsp_cp(daily_pooling(df, "dbn_nez", max_abs), cp=TRUE, useAnom = TRUE, mcmc_params = li
```

```
## [1] "6.88 seconds remaining"
## [1] "4.79 seconds remaining"
## [1] "3.56 seconds remaining"
## [1] "2.68 seconds remaining"
## [1] "1.74 seconds remaining"
## [1] "0.85 seconds remaining"
## [1] "0 seconds remaining"
## [1] "Total time:  5 seconds"
```

```
mcmc_output_e = dsp_cp(daily_pooling(df, "dbe_nez", max_abs), cp=TRUE, useAnom = TRUE, mcmc_params = li
```

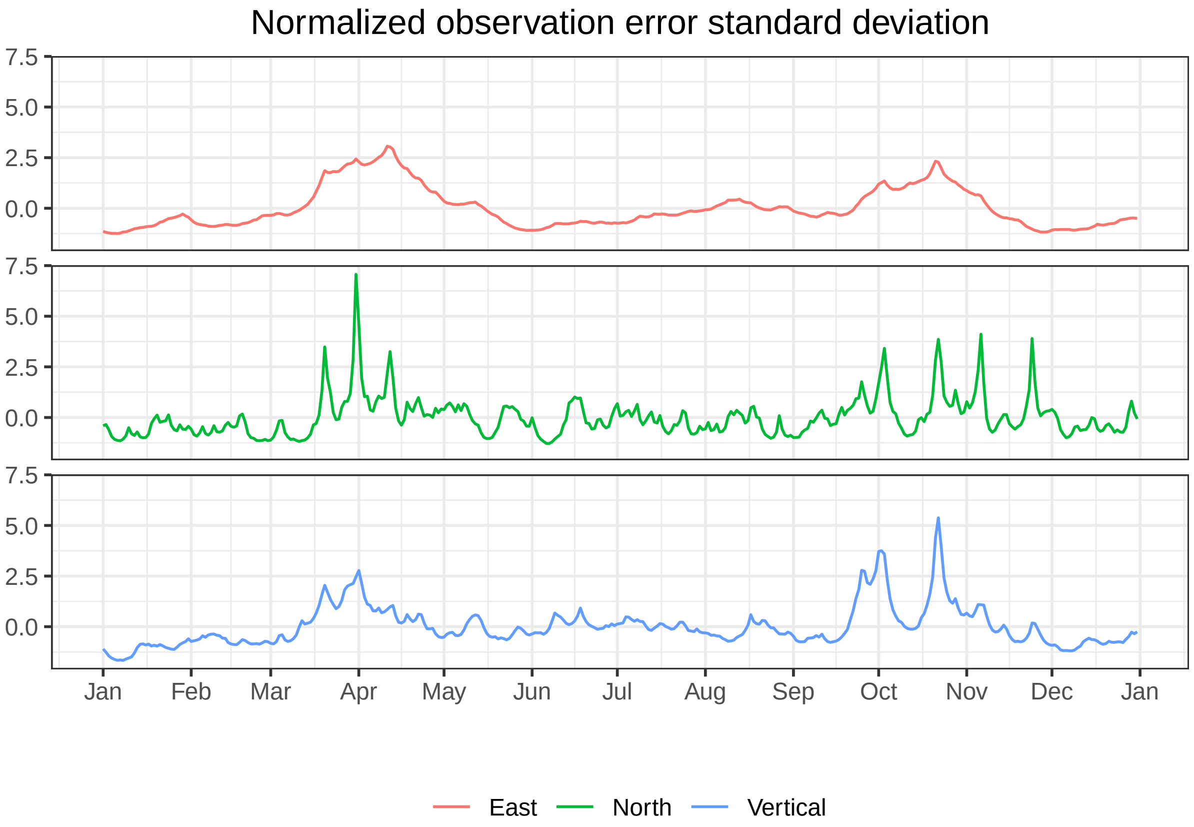
```
## [1] "5.68 seconds remaining"
## [1] "4.39 seconds remaining"
## [1] "3.51 seconds remaining"
## [1] "2.54 seconds remaining"
## [1] "1.66 seconds remaining"
## [1] "0.84 seconds remaining"
## [1] "0 seconds remaining"
## [1] "Total time:  5 seconds"
```

```
mcmc_output_z = dsp_cp(daily_pooling(df, "dbz_nez", max_abs), cp=TRUE, useAnom = TRUE, mcmc_params = li
```

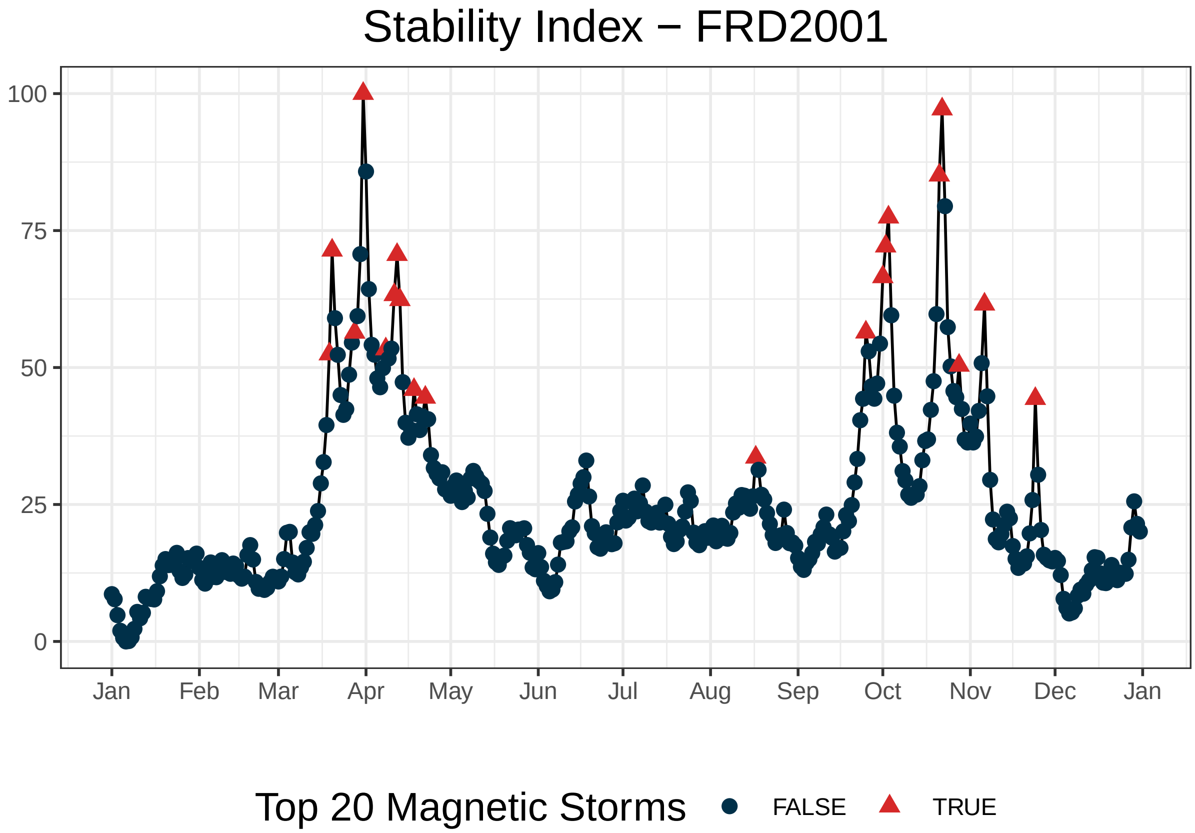
```
## [1] "4.78 seconds remaining"
## [1] "3.96 seconds remaining"
## [1] "3.18 seconds remaining"
## [1] "2.36 seconds remaining"
## [1] "1.57 seconds remaining"
## [1] "0.78 seconds remaining"
## [1] "0 seconds remaining"
## [1] "Total time:  5 seconds"
```

Calculate and visualize the observation error standard deviation $\sigma_{\epsilon,t}$

```
# observation error standard deviation (sigma_epsilon_t)
sd_n = apply(mcmc_output_n$obs_sigma_t2, 2, mean)
sd_e = apply(mcmc_output_e$obs_sigma_t2, 2, mean)
sd_z = apply(mcmc_output_z$obs_sigma_t2, 2, mean)

## evaluation sigma (sigma_beta_t)
# ev_n = c(0, apply(mcmc_output_n$zeta_sigma_t2, 2, mean))
# ev_e = c(0, apply(mcmc_output_e$zeta_sigma_t2, 2, mean))
# ev_z = c(0, apply(mcmc_output_z$zeta_sigma_t2, 2, mean))

df_sds = data.frame(
  sd_n = sqrt(sd_n), sd_e = sqrt(sd_e), sd_z = sqrt(sd_z)
  # ev_n = sqrt(ev_n), ev_e = sqrt(ev_e), ev_z = sqrt(ev_z)
)
df_sds <- as.data.frame(scale(df_sds))
df_sds$idx = 1:nrow(df_sds)
df_sds_plot = gather(df_sds, "direction", "value",  1:3)

p = ggplot(aes(x = as.Date("2001-01-01")+idx-1, y = value, colour = direction), data = df_sds_plot) +
  geom_line()  + facet_grid(rows = vars(direction))+
  scale_color_discrete(labels=c("East", "North", "Vertical"))+
  ggtitle("Normalized observation error standard deviation") +
  xlab("")+ylab("")+
  scale_x_date(date_labels="%b",date_breaks  ="1 month")+
  theme_bw()+
  theme(strip.text.y = element_blank(),
        plot.title = element_text(hjust = 0.5),
        legend.position="bottom",
        legend.title=element_blank())
p
```



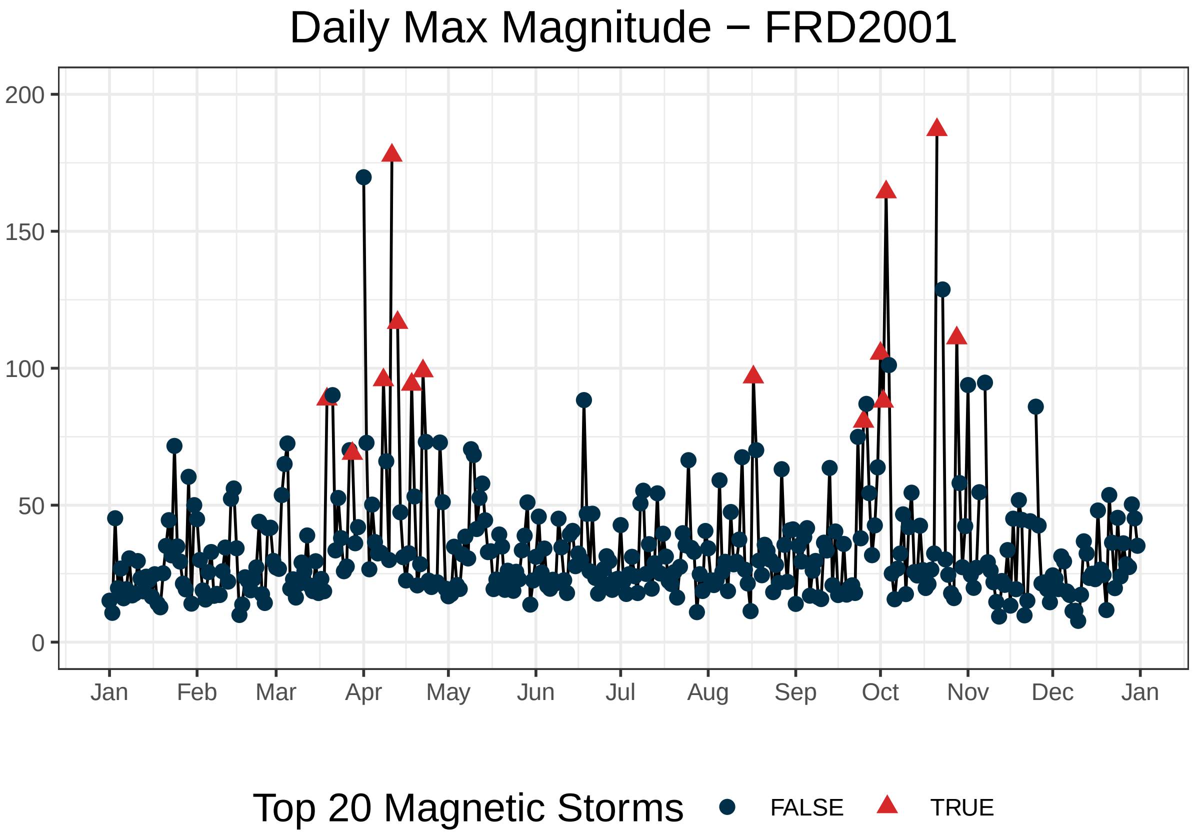
```
top20 = read.table("/Users/hejimi/MPS-Project/top20.csv", sep = ",", header = T)
normalize <- function(x, na.rm = TRUE) {
    return((x- min(x)) /(max(x)-min(x)))
}
df_sds$sd_nez = apply(df_sds[1:3], 1, mean)
df_sds$sd_nez = normalize(df_sds$sd_nez)*100
df_sds$mag_daily = mag_daily
df_sds$Date = as.Date("2001-01-01")+df_sds$idx-1
top20 = top20$yr2001


df_sds$top20 = df_sds$Date %in% as.Date(top20)

p = ggplot(aes(x = Date, y=sd_nez), data = df_sds)+
  geom_line()+
  geom_point(aes(color = top20, shape = top20), size = 2.5)+
  xlab("")+ylab("")+
  scale_x_date(date_labels="%b",date_breaks  ="1 month")+
  scale_color_manual(values=c("#003049", "#d62828"))+
  ggtitle(paste("Stability Index", "-",prefix))+
  theme_bw()+
  labs(shape = "Top 20 Magnetic Storms", color = "Top 20 Magnetic Storms")+
  theme(
    legend.position="bottom",
    plot.title = element_text(hjust = 0.5,size = 17),
    legend.title = element_text(size = 15))
p
```



Visualizing Daily Max Magnitude

```
p = ggplot(aes(x = Date, y=mag_daily), data = df_sds)+
  geom_line()+
  geom_point(aes(color = top20, shape = top20), size = 2.5)+
  xlab("")+ylab("")+ylim(0,200)+
  scale_x_date(date_labels="%b",date_breaks="1 month")+
  scale_color_manual(values=c("#003049", "#d62828"))+
  ggtitle(paste("Daily Max Magnitude", "-",prefix))+
  theme_bw()+
  labs(shape = "Top 20 Magnetic Storms", color = "Top 20 Magnetic Storms")+
  theme(
    legend.position="bottom",
    plot.title = element_text(hjust = 0.5,size = 17),
    legend.title = element_text(size = 15))
p
```

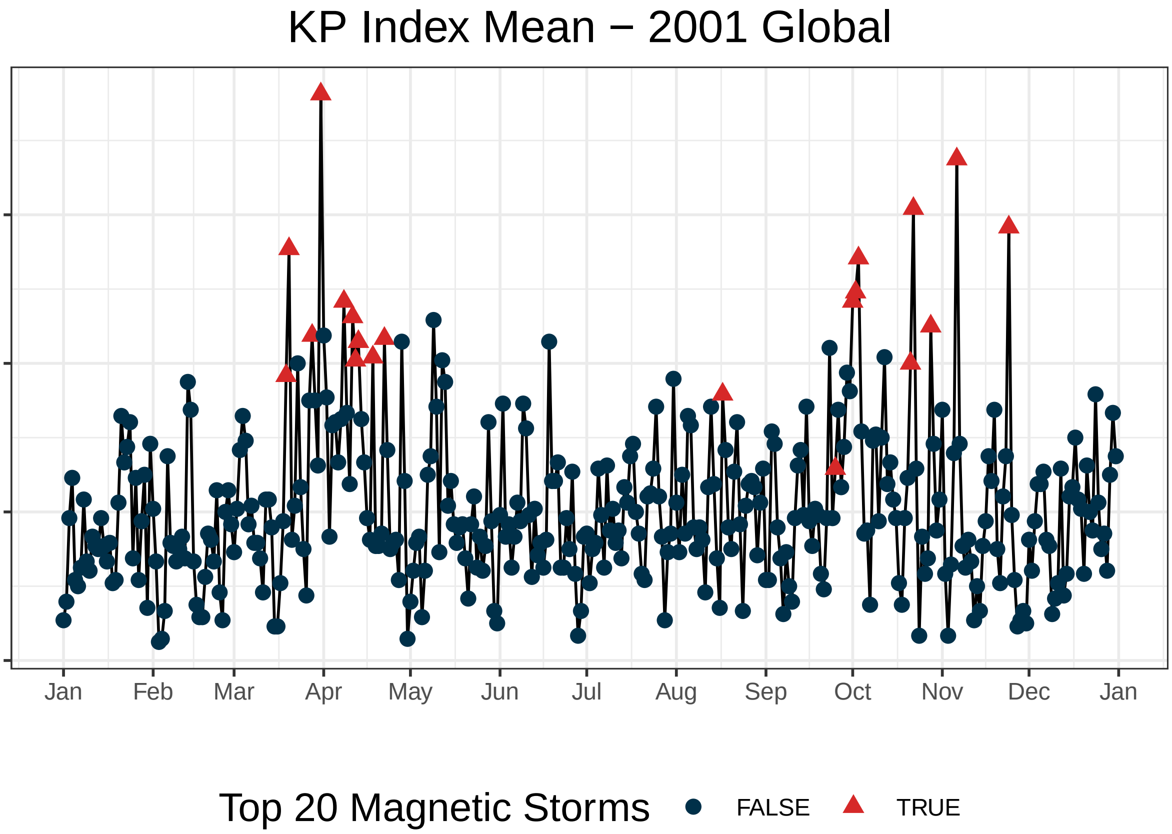
```
## Warning: Removed 6 rows containing missing values (geom_point).
```



Visualizing KP index

```
get_date = function(vec){
  s = paste(as.numeric(vec["YYYY"]), as.numeric(vec["MM"]), as.numeric(vec["DD"]),sep = "/")
  return (s)
}
df_kp = read.table("/Users/hejimi/MPS-Project/SuperMAG-data/kp.csv", sep = ",", header = T)
kp_mat = df_kp[,8:15]
kp_m = rowMeans(kp_mat)
year = 2001
dt = apply(df_kp, 1, get_date)
df_kpidx = data.frame(Date = dt, kp_m = kp_m)
slice_vec = as.Date(df_kpidx$Date) >= paste(year, "-01-01", sep = '') & as.Date(df_kpidx$Date) < paste(
df_kpidx = df_kpidx[slice_vec,]
```

```
df_sds$kp_m = df_kpidx$kp_m
p = ggplot(aes(x = Date, y=kp_m), data = df_sds)+
  geom_line()+
  geom_point(aes(color = top20, shape = top20), size = 2.5)+
  xlab("")+ylab("")+
  scale_color_manual(values=c("#003049", "#d62828"))+
  scale_x_date(date_labels="%b",date_breaks  ="1 month")+
  ggtitle(paste("KP Index Mean - 2001 Global"))+
  theme_bw()+
  labs(shape = "Top 20 Magnetic Storms", color = "Top 20 Magnetic Storms")+
  theme(
    legend.position="bottom",
    plot.title = element_text(hjust = 0.5,size = 17),
    legend.title = element_text(size = 15))
p
```


KP Index Mean − 2001 Global
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Top 20 Magnetic Storms
● FALSE
▲ TRUE