

Xinru Wang

xw2676

2020/5/6

```
data=read.csv("hurrican356.csv") %>% janitor::clean_names()

## select 80% hurricanes as train data set
hurricanes = unique(as.character(data$id))
trRows <- createDataPartition(1:356,
                             p = 0.8,
                             list = FALSE)
train_hurricanes = hurricanes[trRows]
test_hurricanes = hurricanes[-trRows]

df=data %>% separate(time,into=c("sign","date","hour","sign2"),sep = c(1,9,18)) %>% dplyr::select(-sign
  filter(hour == " 00:00:00" | hour == " 06:00:00" | hour == " 12:00:00" | hour == " 18:00:00") %>%
  group_by(id) %>%
  mutate(
    date=as.Date(date, '%y-%m-%d'),
    begin=str_c(as.character(season), '0101'),
    initial=as.Date(begin, "%Y%m%d"),
    days=as.numeric(date-initial),
    final=str_c(date,hour),
    time=as.numeric(difftime(final,final[1],units = "hour")),
    nature = as.numeric(nature)-1
  ) %>% dplyr::select(-begin,-initial)

## DS ET NR SS TS

df_train = df %>%
  filter(id %in% train_hurricanes) %>%
  mutate(
    shift_lat = shift(latitude, fill=NA,type="lag"),
    diff_lat = latitude - shift_lat,
    shift_long = shift(longitude, fill=NA,type="lag"),
    diff_long = longitude - shift_long,
    shift_wind = shift(wind_kt, fill=NA,type="lag"),
    diff_wind = wind_kt - shift_wind,
  ) %>%
  drop_na()

df_test = df %>%
  filter(id %in% test_hurricanes) %>%
  mutate(
    shift_lat = shift(latitude, fill=NA,type="lag"),
```

```

diff_lat = latitude - shift_lat,
shift_long = shift(longitude, fill=NA,type="lag"),
diff_long = longitude - shift_long,
shift_wind = shift(wind_kt, fill=NA,type="lag"),
diff_wind = wind_kt - shift_wind,
) %>%
drop_na()

## time, days and id are created useful variables

loglikeli_func = function(dat,beta,sigma, rho){
  dat = dat %>%
    mutate(
      shift_wind_2 = shift(wind_kt),
      mu = beta[1] + days*beta[2] + season*beta[3] + nature*beta[4] + diff_lat*beta[5] + diff_long*beta[6]
      loglikeli = log(dnorm(wind_kt, mean = mu, sd = sigma))
    ) %>%
  drop_na()

  loglikelihood = sum(dat$loglikeli) + log(dmvnorm(beta, rep(0,7), diag(1,7))) + log(dtruncnorm(rho, a=0, b=1, mean=mu, sd=sigma))
  return(loglikelihood)
}

componentwiseMHstep <- function(x, a, logp, dat) {
  p <- length(x)
  res <- x
  for(i in 1:p) {
    prop <- res
    prop[i] <- x[i] + 2 * (runif(1) - 0.5) * a[i]
    if(log(runif(1)) < loglikeli_func(dat, prop[1:7],prop[8],prop[9]) - loglikeli_func(dat, res[1:7],res[8]))
      res[i] <- prop[i]
  }
  return(res)
}

## 1: -37 2:-0.5   3: a is 1  4: a is 5, -3   5: -1   6: a is 5, 0  7: a is 2, 1  8: a is 2, 30  9: a is 1, 0

nrep <- 20000
x = c(-30,-0.1,0.04,-1,-0.5,0,1,5,1)
x2 = c(-38.5,-0.1,0.03,-0.7,-0.5,0,1,4.2,1)
x3 = c(-38,-0.05,0.01,-0.35,-0.3,0,1,2,1)
x4 = c(-38,-0.005,0.02,0,0,1,0.25,1)
x5 = c(-28,0,0.015,0.009,0,0.01,1,0.1,1)
x = c(-26,0,0.012,0.009,0,0.01,1,0.1,1)
a1 =c(0.1,0.01,0.1,0.1,0.1,0.2,0.1,0.2,0.1)
a2=c(0.1,0.001,0.001,0.1,0.4,0.4,0.1,0.01,0.001)

chain <- matrix(NA, nrep, length(x))
for(i in 1:nrep) {
  newx <- componentwiseMHstep(x, a=c(0.1,0.0001,0.0001,0.01,0.001,0.01,0.001,0.02,0.0003),
    logp=loglikeli_func, dat=df_train)
  chain[i,] <- newx
  x <- newx
  print(i)
}

```

```
}
```

```
chain = as.data.frame(chain) %>%
  mutate(
    index = row_number()
  )
```

```
write.csv(chain, "chain")
chain = read_csv("./chain")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
```

```
## cols(
##   X1 = col_double(),
##   V1 = col_double(),
##   V2 = col_double(),
##   V3 = col_double(),
##   V4 = col_double(),
##   V5 = col_double(),
##   V6 = col_double(),
##   V7 = col_double(),
##   V8 = col_double(),
##   V9 = col_double(),
##   index = col_double()
## )
```

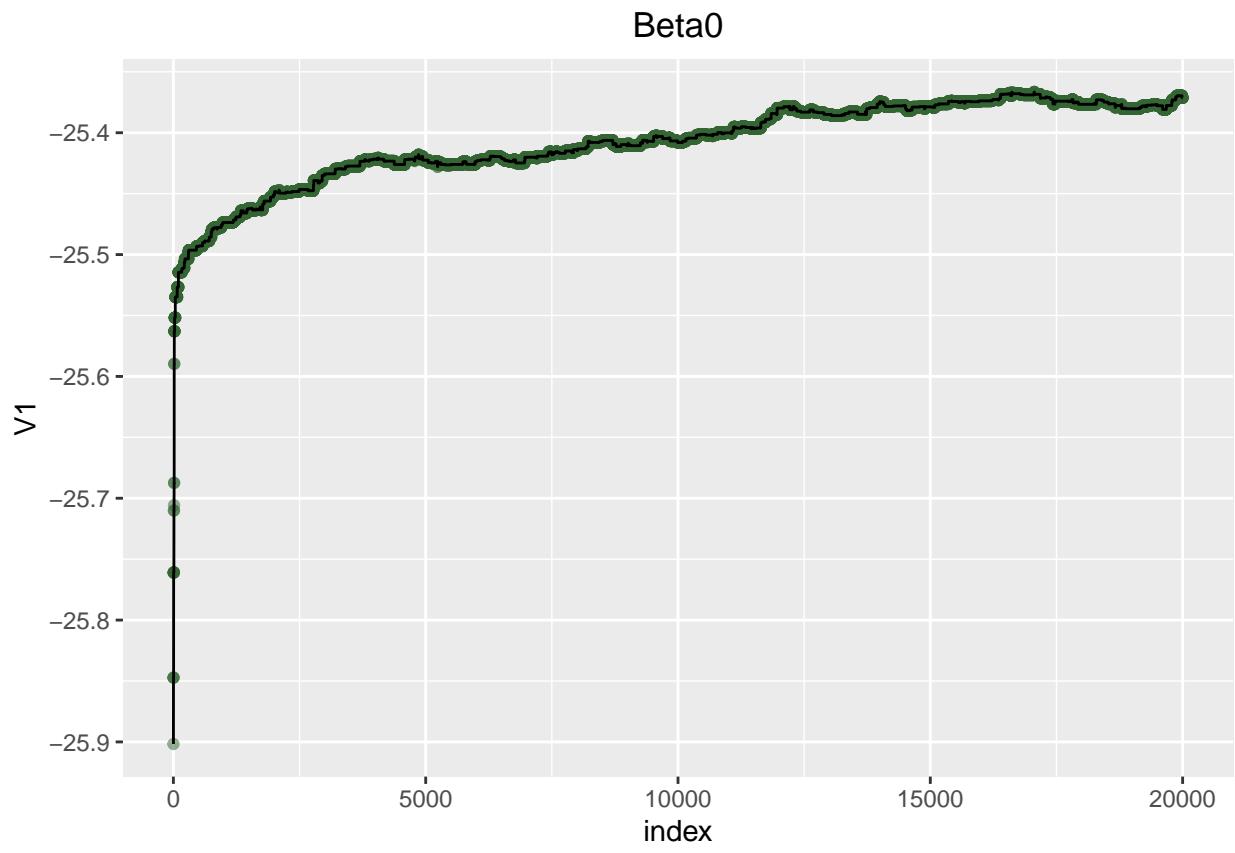
```
p <- c(paste0('plot_fluc', 1:9))
```

```
for(i in 1:7){
  assign(p[i], ggplot(data = chain, aes_string(x = "index", y = paste("V", i, sep = ""))) + geom_point(color =
  geom_line() +
  ggtitle(paste("Beta", i - 1, sep = "")) + theme(plot.title = element_text(hjust = 0.5)))
}
```

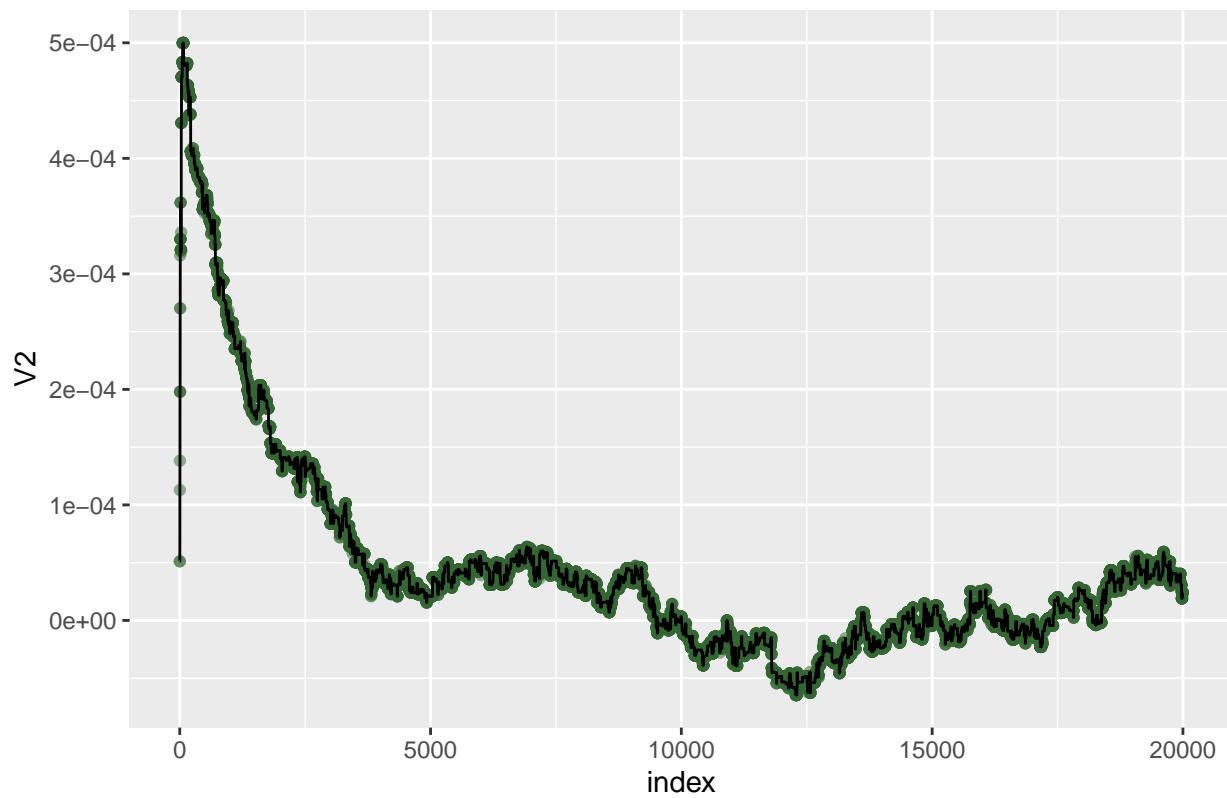
```
assign(p[8], ggplot(data = chain, aes_string(x = "index", y = paste("V", i, sep = ""))) + geom_point(color =
  geom_line() +
  ggtitle("Sigma") + theme(plot.title = element_text(hjust = 0.5)))
```

```
assign(p[9], ggplot(data = chain, aes_string(x = "index", y = paste("V", i, sep = ""))) + geom_point(color =
  geom_line() +
  ggtitle("Rho") + theme(plot.title = element_text(hjust = 0.5)))
```

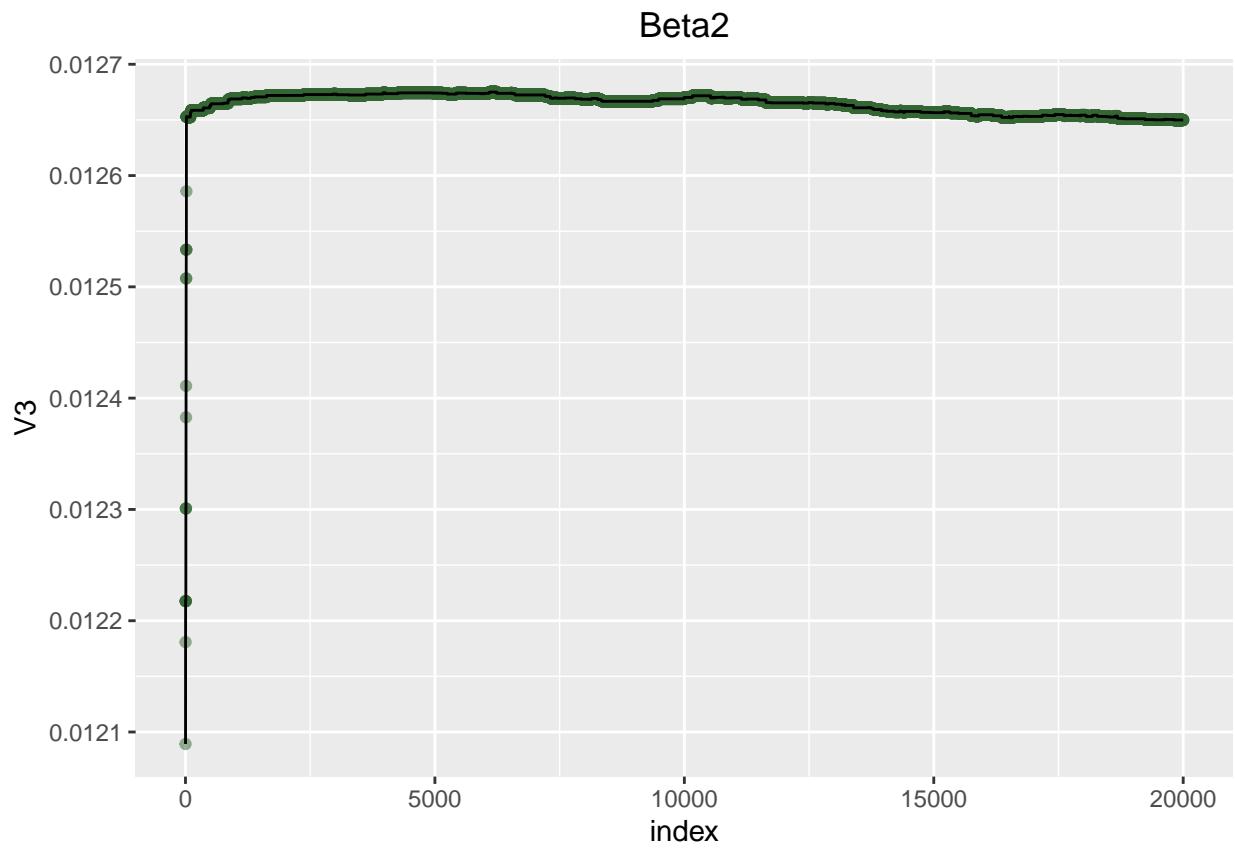
```
plot_fluc1
```



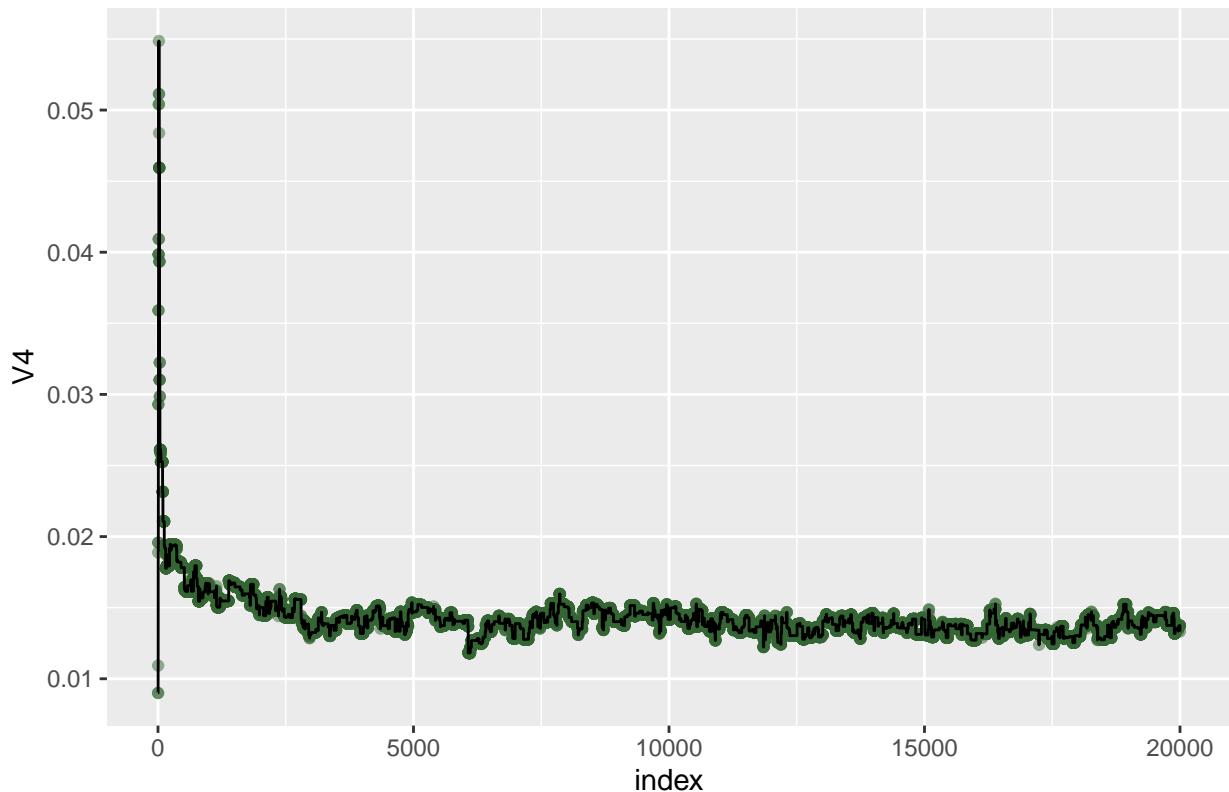
Beta1



plot_fluc3

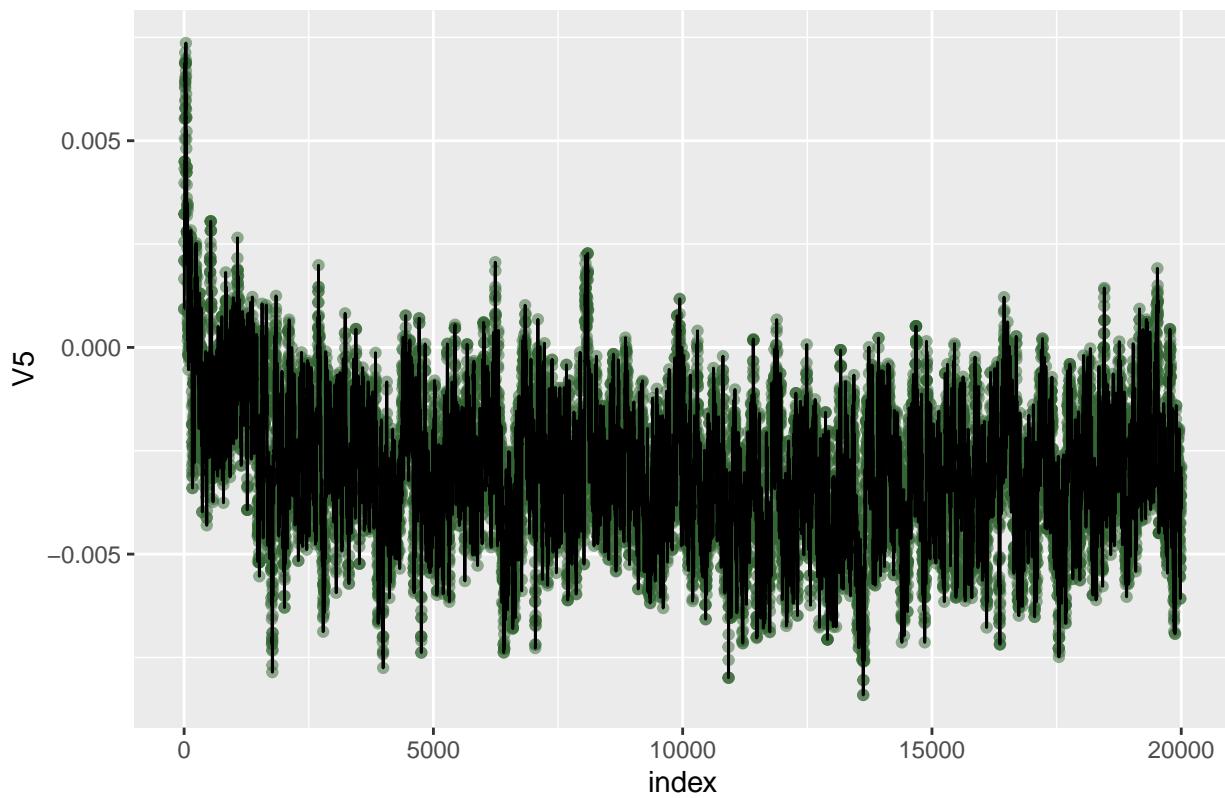


Beta3



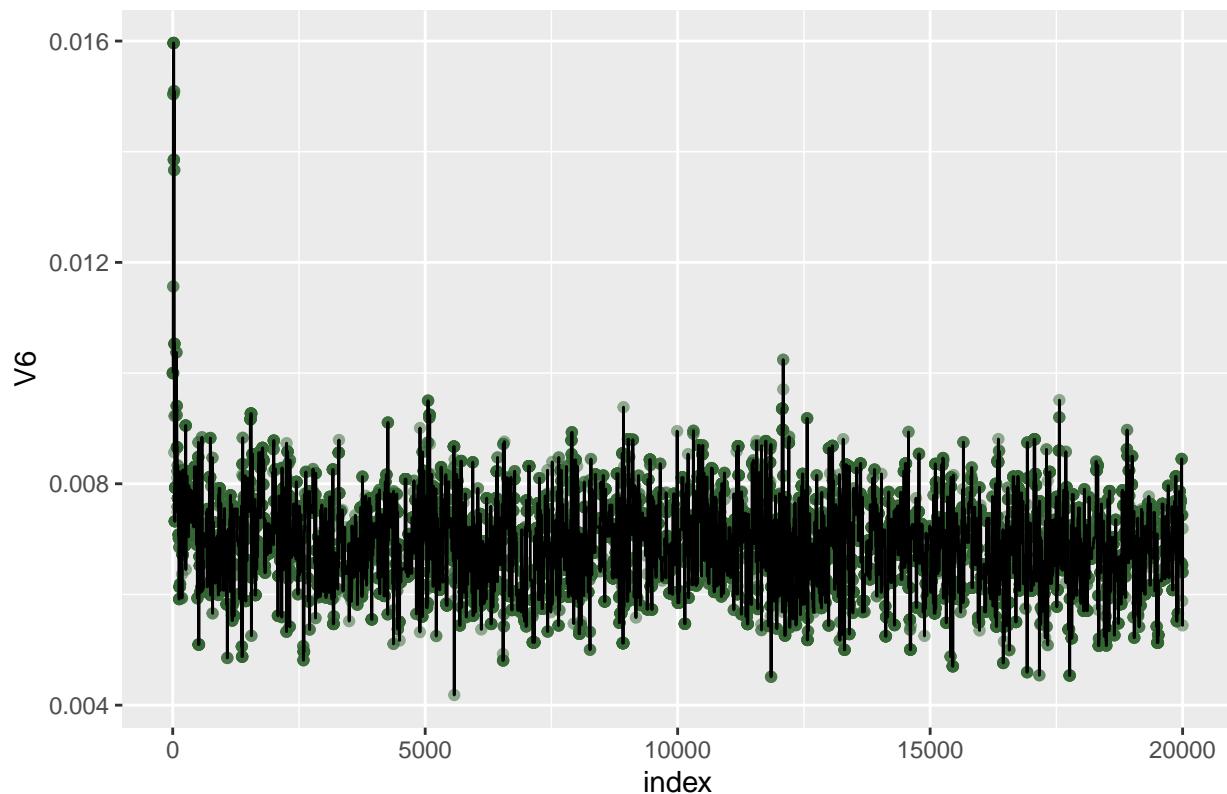
plot_fluc5

Beta4

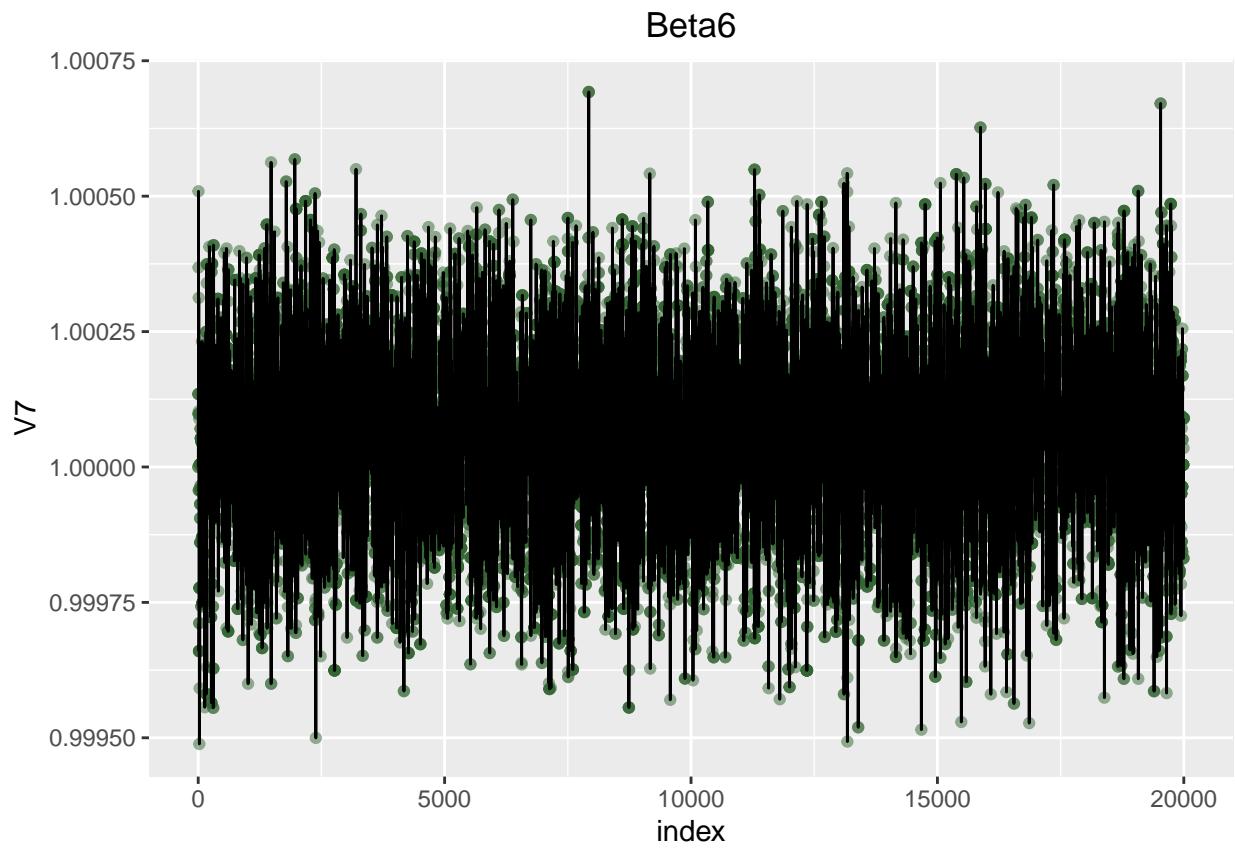


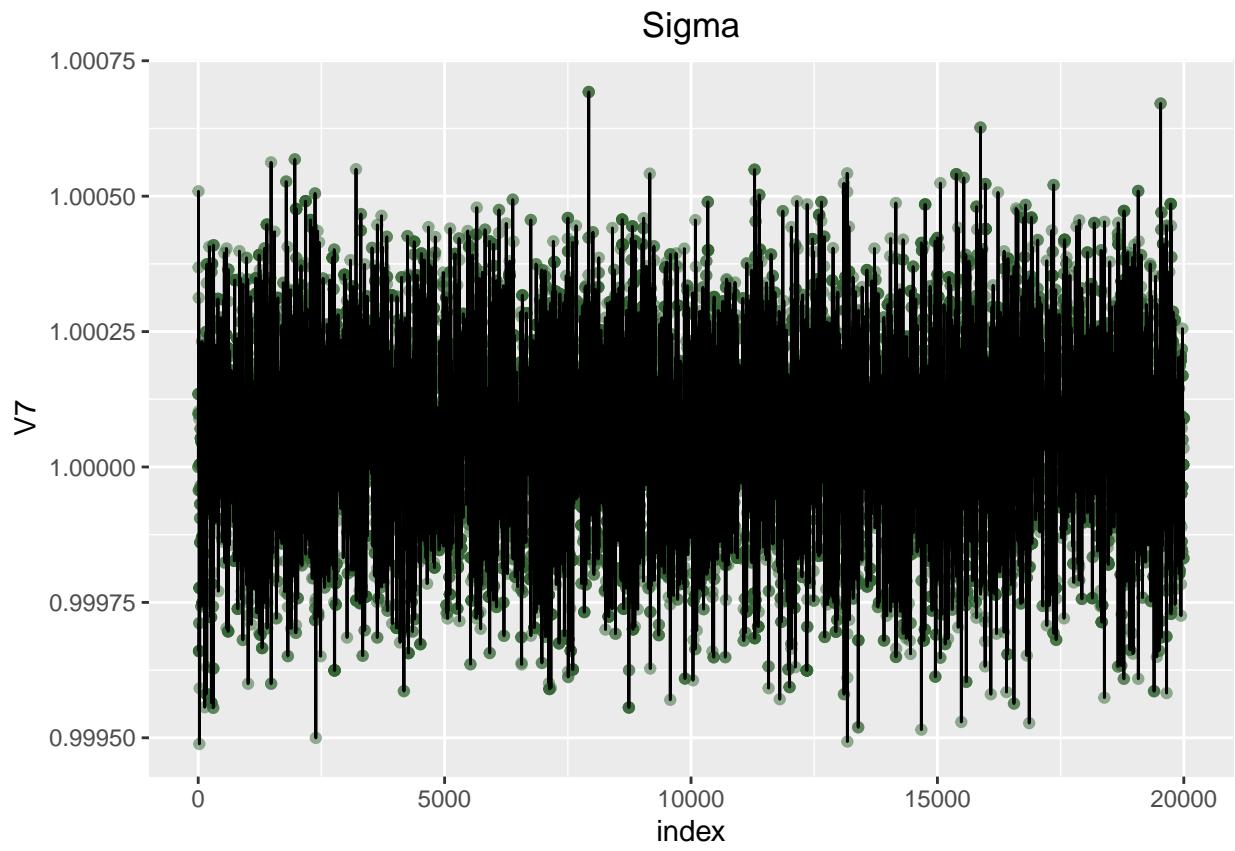
plot_fluc6

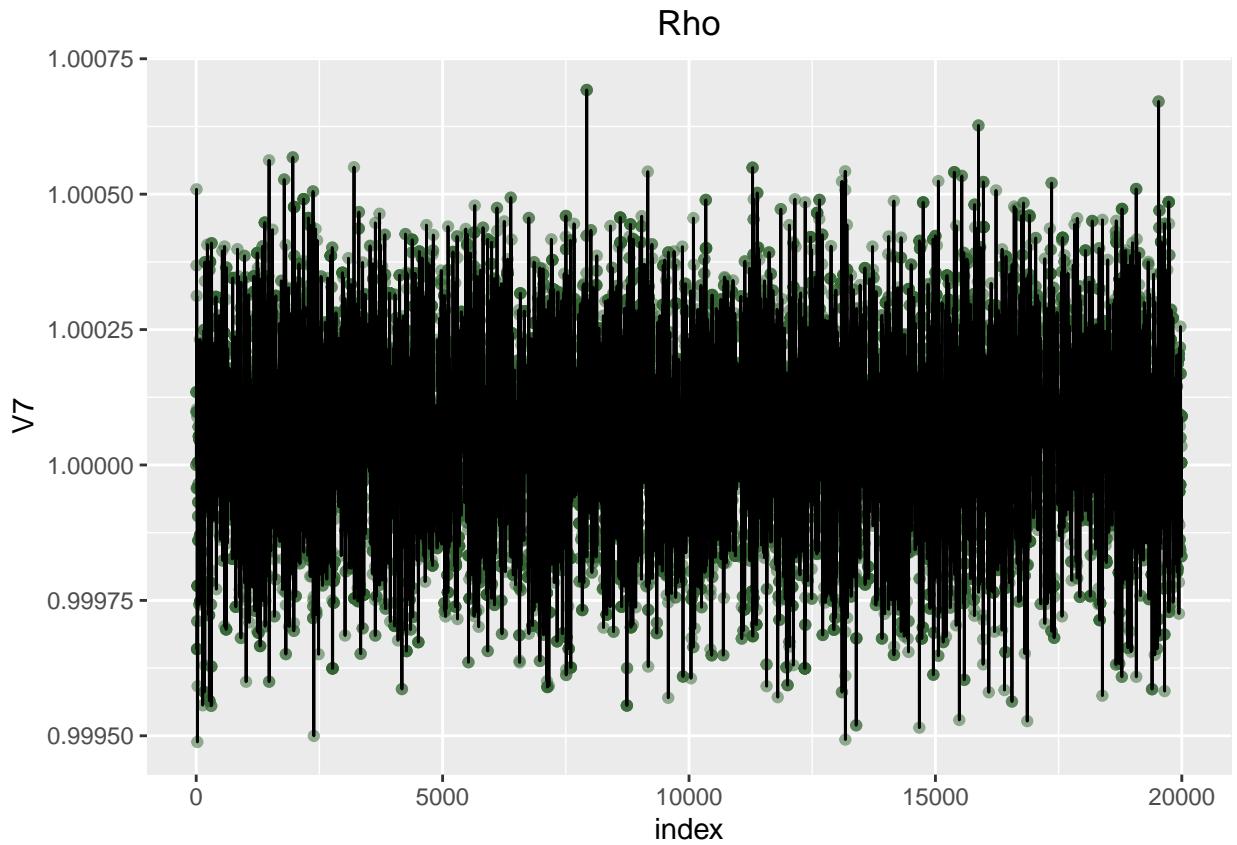
Beta5



plot_fluc7







```

## use the last 5000 numbers

chain_new = chain[15000:20000,]

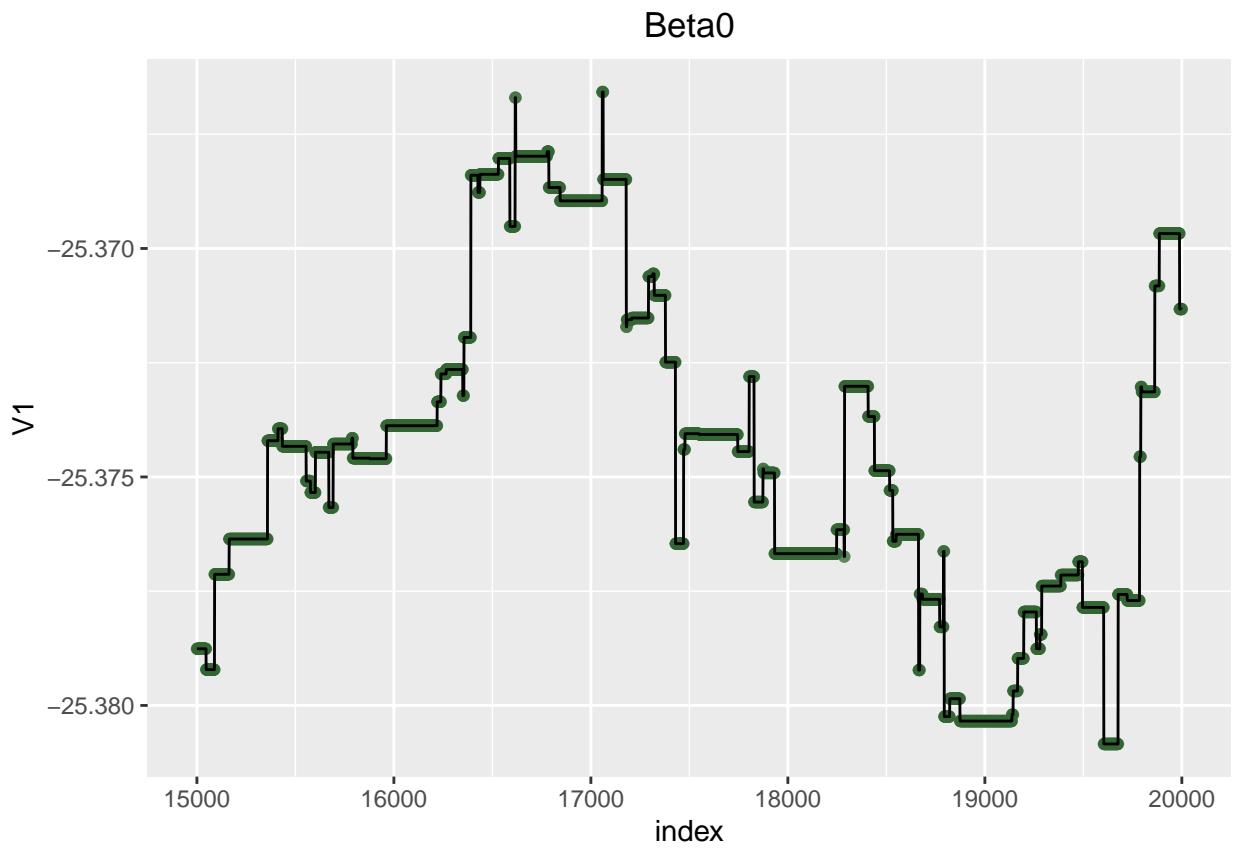
q <- c(paste0('plot_final',1:9))
for(i in 1:7){
  assign(q[i],ggplot(data = chain_new, aes_string(x = "index", y = paste("V",i,sep="")))+geom_point(col
  geom_line() +
  ggtitle(paste("Beta",i-1,sep=""))+theme(plot.title = element_text(hjust = 0.5)))
}

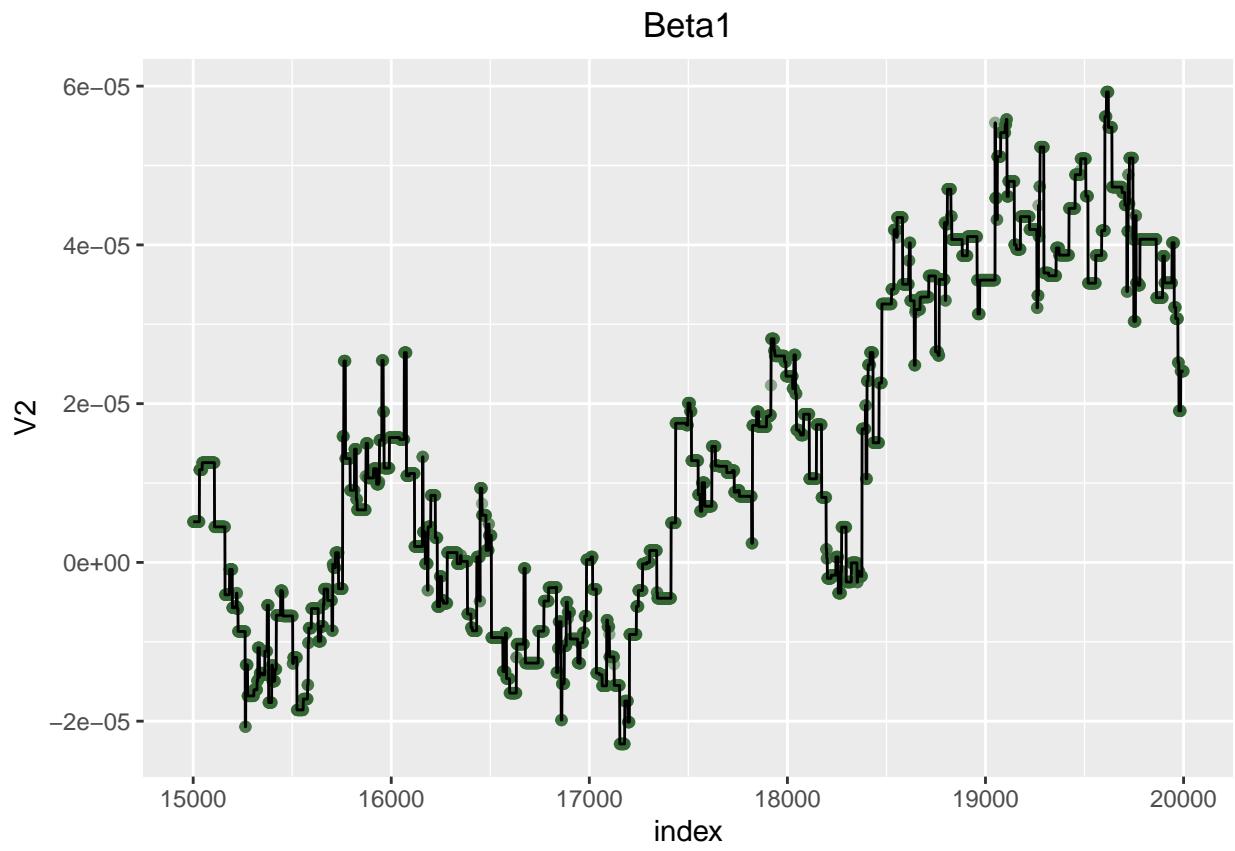
assign(q[8],ggplot(data = chain_new, aes_string(x = "index", y = paste("V",i,sep="")))+geom_point(col
  geom_line() +
  ggtitle("Sigma") + theme(plot.title = element_text(hjust = 0.5)))

assign(q[9],ggplot(data = chain_new, aes_string(x = "index", y = paste("V",i,sep="")))+geom_point(col
  geom_line() +
  ggtitle("Rho") + theme(plot.title = element_text(hjust = 0.5)))

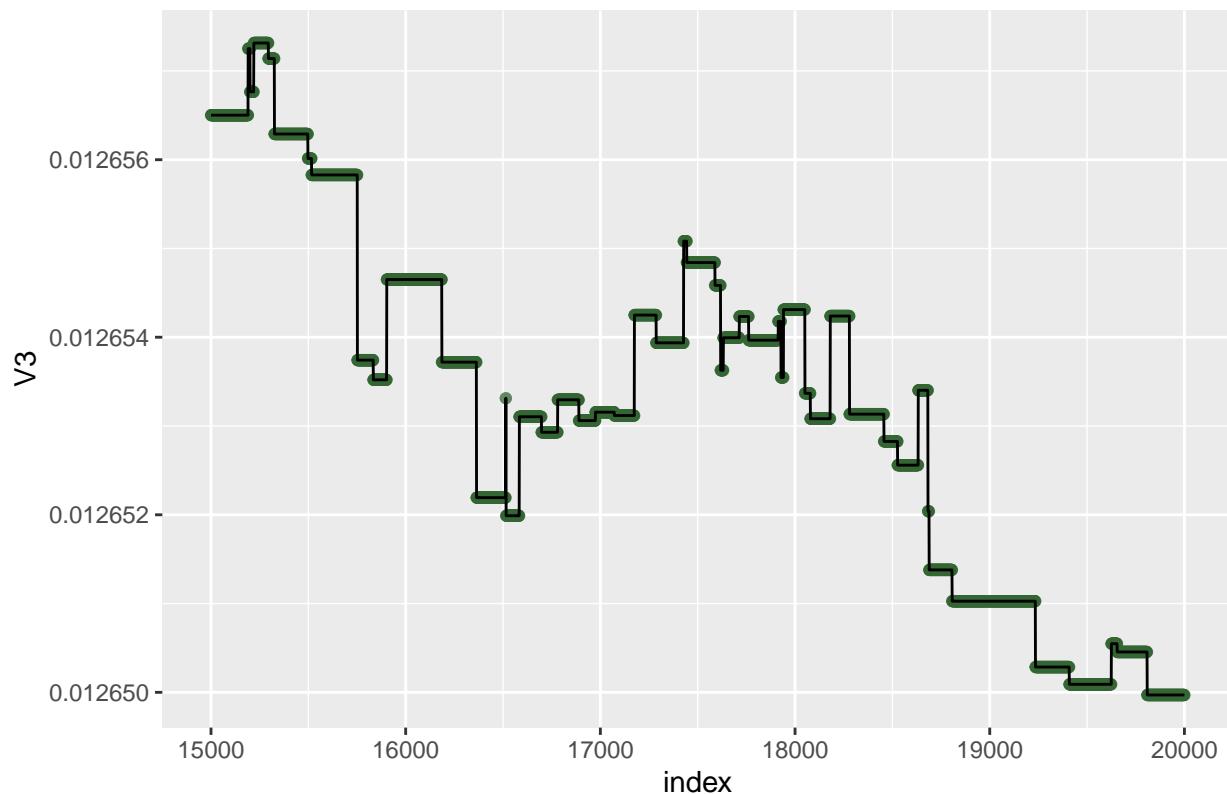
plot_final1

```



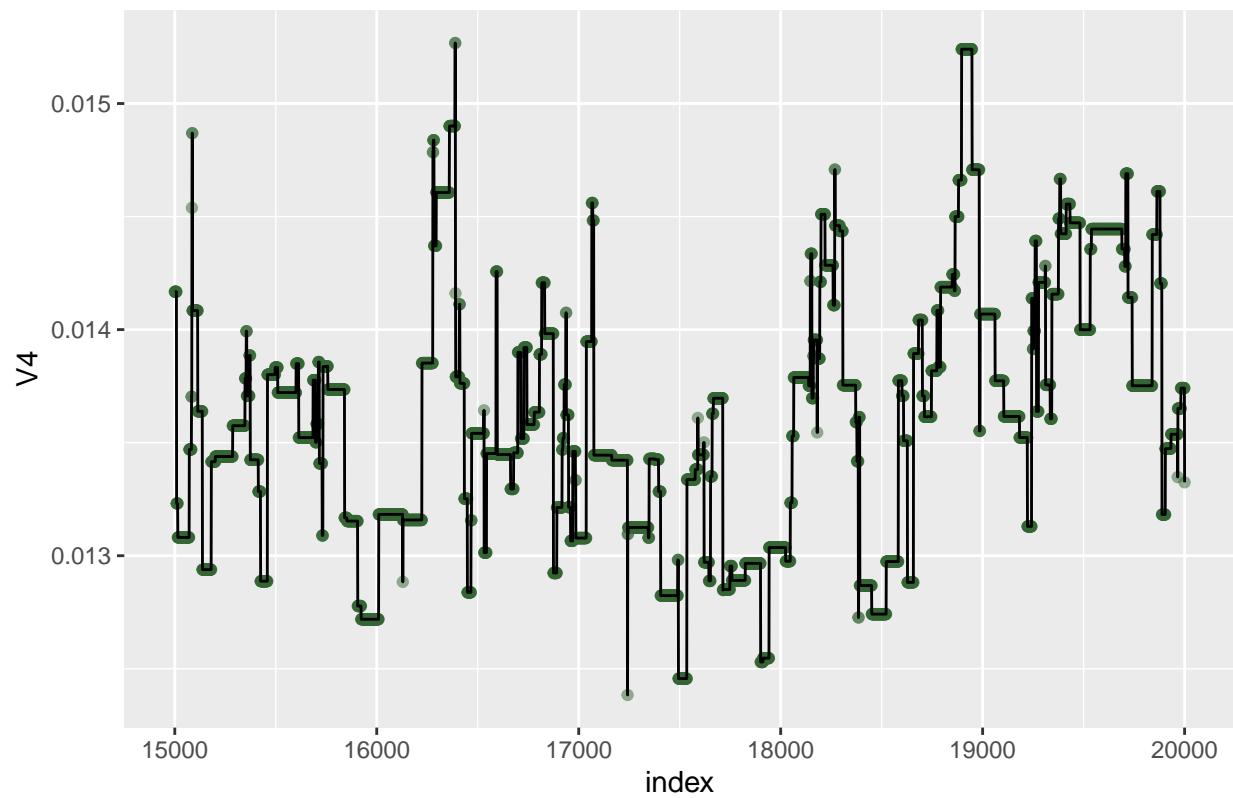


Beta2

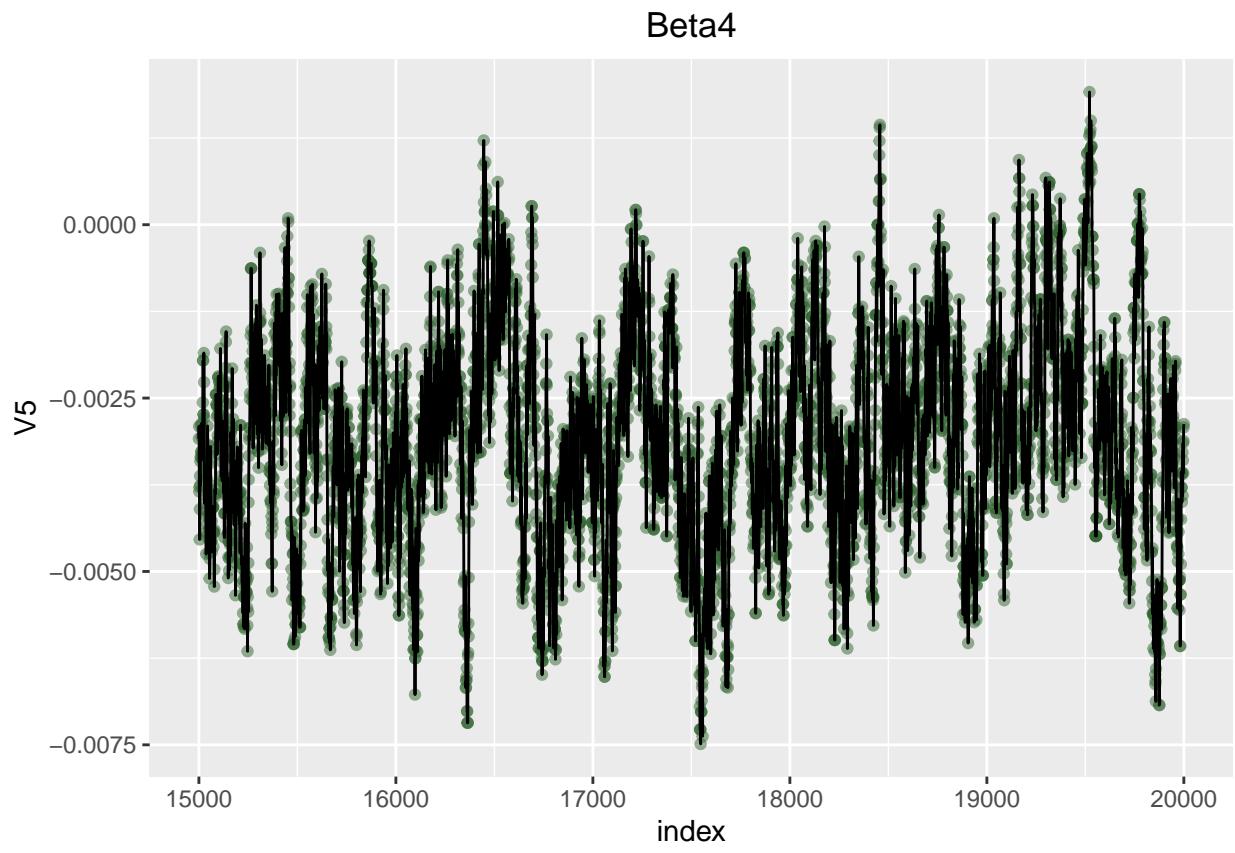


plot_final4

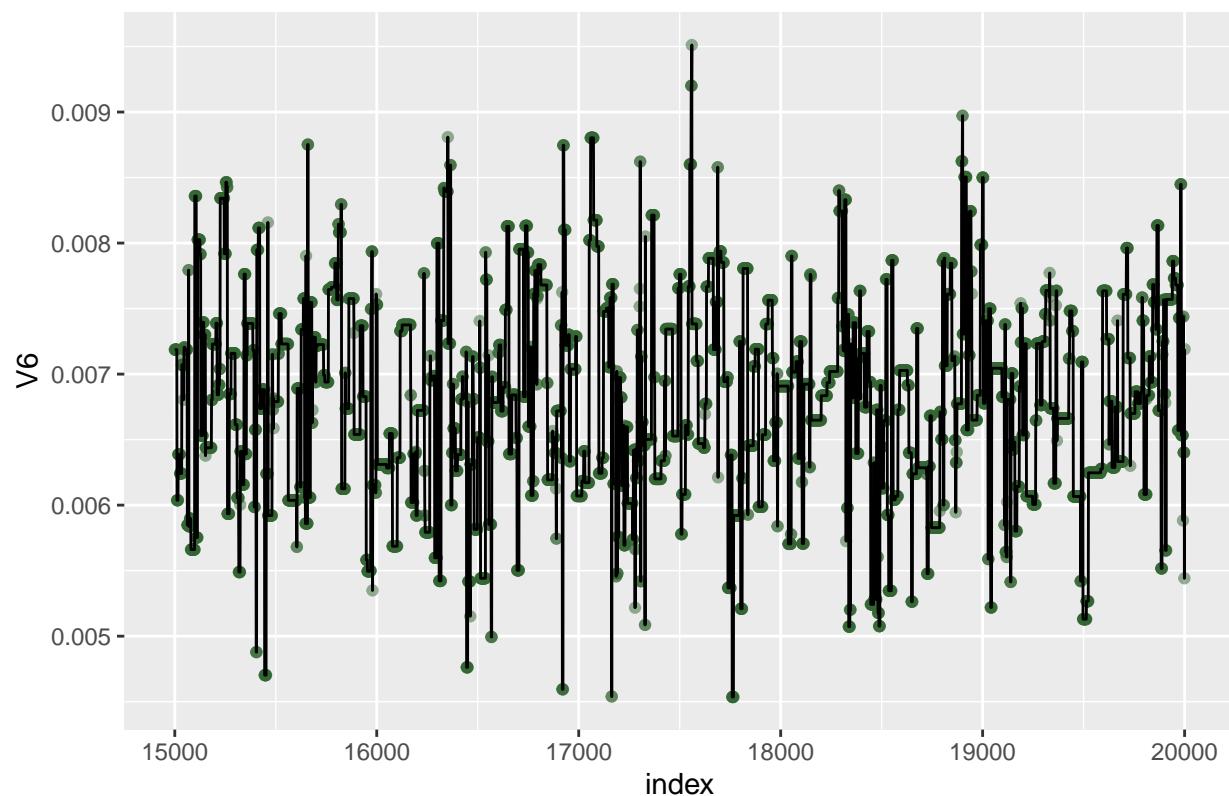
Beta3



plot_final5



Beta5



plot_final7

