Part_1-4

Ruiqi Yan (ry2417)

5/4/2022

Hurrican Data

hurricane703.csv collected the track data of 703 hurricanes in the North Atlantic area since 1950. For all the storms, their location (longitude & latitude) and maximum wind speed were recorded every 6 hours. The data includes the following variables

1. **ID**: ID of the hurricanes
2. **Season**: In which **year** the hurricane occurred
3. **Month**: In which **month** the hurricane occurred
4. **Nature**: Nature of the hurricane

- ET: Extra Tropical
- DS: Disturbance
- NR: Not Rated
- SS: Sub Tropical
- TS: Tropical Storm

5. **time**: dates and time of the record

6. **Latitude** and **Longitude**: The location of a hurricane check point
7. **Wind.kt** Maximum wind speed (in Knot) at each check point

load and clean data

```
load("df_rm.RData")
## create variables of interest
dt = df_rm %>%
  janitor::clean_names() %>%
  group_by(id) %>%
  mutate(wind_lag = lag(wind_kt,1), ##previous wind speed
         lat_shift = lag(latitude, 1) - lag(latitude, 2), ## change of lat
         long_shift = lag(longitude, 1) - lag(longitude, 2), ## change of long
         wind_shift = lag(wind_kt, 1) - lag(wind_kt, 2)) %>%  ##change of wind speed
  drop_na()

## group by hurricane and create design matrix for each hurricane
df_mcmc = dt %>%
  dplyr::select(id,wind_kt,wind_lag,lat_shift, long_shift, wind_shift) %>%
  nest(y = wind_kt, x_matrix=wind_lag:wind_shift) %>%
  mutate(x_matrix = map(.x = x_matrix, ~model.matrix(~., data = .x)),
         y = map(y, pull))

y_obs = df_mcmc$y ## list of wind speed for each hurricane
```

```
x_matrix = df_mcmc$x_matrix ## list of design matrix for each hurricane
n_obs = nrow(dt) ## number of

## list of observation data: wind speed and design matrix
obs_list = list(y_obs = y_obs,
                x_matrix = x_matrix)

n_freq = dt %>% group_by(id) %>% count() %>% pull(n) ## number of obs for each hurricane
d = 5  ## dimension of beta_i
m = 681  ## number of hurricanes
n_beta = m*d  ## total number of beta
n_theta = n_beta+d+1+15  ## total number of parameters
mu_ind = (n_beta+1):(n_beta+d) ## index of mu
sigma_ind = n_beta+d+1 ## index of sigma
cov_m_ind = (n_beta+d+2):n_theta ## index of elements of inverse covariance matrix
rep_time = rep(n_freq,d) ## repeat time for each beta
```

Part I: Likelihood, Prior, Posterior and Conditional Posterior

```
## log posterior for beta_j

log_posterior_beta = function(beta, sigma, mu, sigma_p,j){
  x = obs_list$x_matrix[[j]]
  y = obs_list$y_obs[[j]]
  return(-t(y - x %*% beta) %*% (y - x %*% beta)/(2*sigma)-(1/2)*t(beta-mu)%*%sigma_p%*%(beta-mu))
}

## log posterior for sigma

log_posterior_sigma = function(beta_frame, sigma){
  ## make sure sigma is positive
  if(sigma <= 0){
    return(-Inf)
  } else{
    log_lik = -(n_obs/2)*log(sigma)-sum((dt$wind_kt-beta_frame[,1]-dt$wind_lag*beta_frame[,2]-dt$lat_shi
    log_prior = -log(sigma)
    return(log_lik + log_prior)
  }
}

## calculate the number of unique value
numunique = function(x){length(unique(x))}

trace_inverse = function(x){
  m = matrix(0,5,5)
  m[lower.tri(m, diag = TRUE)] = x[cov_m_ind]
  m[upper.tri(m)] = t(m)[upper.tri(m)]
  return(sum(diag(m)))
}
```

Likelihood

$$L(\boldsymbol{Y} \mid \boldsymbol{\theta}) \propto \prod_{i=1}^{m} \left(\sigma^2\right)^{-\frac{n_i}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)^T \left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)\right\}$$

where m is the number of hurricane and n_i is the number of observations for i_{th} hurricane

Prior

$$\pi\left(\boldsymbol{\theta} = (\boldsymbol{B}, \boldsymbol{\mu}, \Sigma^{-1}, \sigma^2)\right) \propto \left(\sigma^2\right)^{-1} \left|\Sigma^{-1}\right|^{d+1} \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\right)\right\} \prod_{i=1}^{m} \left|\Sigma^{-1}\right|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1}(\boldsymbol{\beta}_i - \boldsymbol{\mu})\right\}$$

where d is the dimension of $\boldsymbol{\mu}$

Posterior

$$\begin{aligned}
\pi(\boldsymbol{\theta} \mid \boldsymbol{Y}) \propto{}& \left(\sigma^2\right)^{-1} \left|\Sigma^{-1}\right|^{d+1} \exp\left\{-\frac{1}{2} + \operatorname{tr}\left(\Sigma^{-1}\right)\right\} \\
&\times \prod_{i=1}^{m} \left(\sigma^2\right)^{-\frac{n_i}{2}} \left|\Sigma^{-1}\right|^{\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)^T \left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)\right\} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)\right\} \\
&= \left(\sigma^2\right)^{-\left(1+\frac{\sum_{i=1}^{m} n_i}{2}\right)} \left|\Sigma^{-1}\right|^{d+1+\frac{m}{2}} \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\right)\right\} \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)\right\} \\
&\times \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)^T \left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)\right\}
\end{aligned}$$

Conditional Posterior

For $\boldsymbol{\beta}_i$:

$$\pi(\boldsymbol{\beta}_i \mid \boldsymbol{\theta}_{(-\boldsymbol{\beta}_i)}\boldsymbol{Y}) \propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right) - \frac{1}{2\sigma^2}\left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)^T \left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)\right\}$$

For $\boldsymbol{\mu}$:

$$\begin{aligned}
\pi\left(\boldsymbol{\mu} \mid \boldsymbol{\theta}_{(-\boldsymbol{\mu})}, \boldsymbol{Y}\right) &\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)\right\} \\
(\boldsymbol{\beta}_i - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{\beta}_i - \boldsymbol{\mu}) &= \operatorname{tr}\left(\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)\right) \\
&= \operatorname{tr}\left(\Sigma^{-1}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T\right) \\
\pi\left(\boldsymbol{\mu} \mid \boldsymbol{\theta}_{(-\boldsymbol{\mu})}, Y\right) &\propto \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\sum_{i=1}^{m}\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\beta}_i - \boldsymbol{\mu}\right)^T\right)\right\} \\
&= \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}m\left(\boldsymbol{\mu} - \bar{\boldsymbol{\beta}}\right)\left(\boldsymbol{\mu} - \bar{\boldsymbol{\beta}}\right)^T\right)\right\} \\
&= \exp\left\{-\frac{1}{2}(\boldsymbol{\mu} - \bar{\boldsymbol{\beta}})^T\Sigma^{-1}m(\boldsymbol{\mu} - \bar{\boldsymbol{\beta}})\right\} \\
&\Rightarrow \boldsymbol{\mu} \mid \boldsymbol{\theta}_{(-\boldsymbol{\mu})}, \boldsymbol{Y} \sim N(\bar{\boldsymbol{\beta}}, \Sigma/m)
\end{aligned}$$

For σ^2:

$$\pi\left(\sigma^2 \mid \boldsymbol{\theta}_{(-\sigma^2)}, \boldsymbol{Y}\right) \propto \left(\sigma^2\right)^{-\left(1+\frac{\sum_{i=1}^{m} n_i}{2}\right)} \times \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)^T \left(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\beta}_i\right)\right\}$$

$$
\begin{aligned}
\pi\left(\Sigma^{-1} \mid \boldsymbol{\theta}_{(-\Sigma^{-1})}, \boldsymbol{Y}\right) &\propto \left|\Sigma^{-1}\right|^{d+1+\frac{m}{2}} \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\right)\right\} \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}\left(\boldsymbol{\beta}_i-\boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{\beta}_i-\boldsymbol{\mu}\right)\right\} \\
&= \left|\Sigma^{-1}\right|^{d+1+\frac{m}{2}} \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\right)-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\sum_{i=1}^{m}\left(\boldsymbol{\beta}_i-\boldsymbol{\mu}\right)\left(\boldsymbol{\beta}_i-\boldsymbol{\mu}\right)^T\right)\right\} \\
&= \left|\Sigma^{-1}\right|^{d+1+\frac{m}{2}} \exp\left\{-\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\left(I+\sum_{i=1}^{m}\left(\beta_i-\mu\right)\left(\beta_1-\mu\right)^T\right)\right)\right\} \\
&\Rightarrow \Sigma^{-1} \mid \boldsymbol{\theta}_{(-\Sigma^{-1})}, \boldsymbol{Y} \sim \text{Wishart}\left(3d+3+m, \left(\boldsymbol{I}+\sum_{i=1}^{m}\left(\boldsymbol{\beta}_i-\boldsymbol{\mu}\right)\left(\boldsymbol{\beta}_i-\boldsymbol{\mu}\right)^T\right)^{-1}\right)
\end{aligned}
$$

Part II: MCMC Algorithm

We could generate full conditional posterior distribution for partial parameters, so we apply hybrid algorithm consisting with Metropolis-Hastings steps and Gibbs steps.

MH steps for β_{ij}:

Sampling proposed β'_{ij} $j=0,1...4$ for i_{th} hurricane from proposal distribution $U\left(\beta_{ij}^{(t)}-a_{ij}, \beta_{ij}^{(t)}+a_{ij}\right)$, where a_{ij} is the search window for β_{ij}. Since the proposals are symmetry, the accepting or rejecting the proposed β'_{ij} depends on the ratio of posterior distribution. Some of the parameters in $\boldsymbol{\theta}$ could be cancelled out, so the ratio simplified to be:

$$
\frac{\pi\left(\boldsymbol{\beta}'_i, \boldsymbol{\theta}^{(t)}_{(-\boldsymbol{\beta}_i)} \mid \boldsymbol{Y}\right)}{\pi\left(\boldsymbol{\beta}^{(t)}_i, \boldsymbol{\theta}^{(t)}_{(-\boldsymbol{\beta}_i)} \mid \boldsymbol{Y}\right)} = \frac{\exp\left\{-\frac{1}{2\sigma^{2(t)}}\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}'_i\right)^T\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}'_i\right)-\frac{1}{2}\left(\boldsymbol{\beta}'_i-\boldsymbol{\mu}^{(t)}\right)^T\Sigma^{-1^{(t)}}\left(\boldsymbol{\beta}'_i-\boldsymbol{\mu}^{(t)}\right)\right\}}{\exp\left\{-\frac{1}{2\sigma^{2(t)}}\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t)}_i\right)^T\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t)}_i\right)-\frac{1}{2}\left(\boldsymbol{\beta}^{(t)}_i-\boldsymbol{\mu}^{(t)}\right)^T\Sigma^{-1(t)}\left(\boldsymbol{\beta}^{(t)}_i-\boldsymbol{\mu}^{(t)}\right)\right\}}
$$

where $\boldsymbol{\beta}'_i$ consisting with β'_{ij}, $\beta^{(t)}_{ik}$ for $k>j$ and $\beta^{(t+1)}_{ik}$ for $k<j$.

The log of the ratio:

$$
\begin{aligned}
\log\frac{\pi\left(\boldsymbol{\beta}'_i, \boldsymbol{\theta}^{(t)}_{(-\boldsymbol{\beta}_i)} \mid \boldsymbol{Y}\right)}{\pi\left(\boldsymbol{\beta}^{(t)}_i, \boldsymbol{\theta}^{(t)}_{(-\boldsymbol{\beta}_i)} \mid \boldsymbol{Y}\right)} &= -\frac{1}{2}\left(\frac{\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}'_i\right)^T\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}'_i\right)}{\sigma^{2(t)}}+\left(\boldsymbol{\beta}'_i-\boldsymbol{\mu}^{(t)}\right)^T\Sigma^{-1^{(t)}}\left(\boldsymbol{\beta}'_i-\boldsymbol{\mu}^{(t)}\right)\right) \\
&\quad+\frac{1}{2}\left(\frac{\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t)}_i\right)^T\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t)}_i\right)}{\sigma^{2(t)}}+\left(\boldsymbol{\beta}^{(t)}_i-\boldsymbol{\mu}^{(t)}\right)^T\Sigma^{-1^{(t)}}\left(\boldsymbol{\beta}^{(t)}_i-\boldsymbol{\mu}^{(t)}\right)\right)
\end{aligned}
$$

Then we randomly sample u from $U(0,1)$ and compare $\log(u)$ with the log ratio, if the $\log(u)$ is smaller, we accept $\beta'_{ij}=\beta^{(t+1)}_{ij}$, otherwise we reject β'_{ij} and $\beta^{(t)}_{ij}=\beta^{(t+1)}_{ij}$.

Then, Gibb step for $\boldsymbol{\mu}$: Sample $\boldsymbol{\mu}^{(t+1)}$ from $N\left(\bar{\boldsymbol{\beta}}^{(t+1)}, \Sigma^{(t)}/m\right)$.

Next, MH step to generate $\sigma^{2'}$. The log posterior ratio:

$$
\begin{aligned}
\log\frac{\pi\left(\sigma^{2'}, \boldsymbol{\theta}^{(t)}_{(-\sigma^2)} \mid \boldsymbol{Y}\right)}{\pi\left(\sigma^{2^{(t)}}, \boldsymbol{\theta}^{(t)}_{(-\sigma^2)} \mid \boldsymbol{Y}\right)} &= -\left(1+\frac{M}{2}\right)\log(\sigma^{2'})-\frac{1}{2\sigma^{2'}}\sum_{i=1}^{m}\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t+1)}_i\right)^T\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t+1)}_i\right) \\
&\quad+\left(1+\frac{M}{2}\right)\log(\sigma^{2^{(t)}})+\frac{1}{2\sigma^{2^{(t)}}}\sum_{i=1}^{m}\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t+1)}_i\right)^T\left(\boldsymbol{Y}_i-\boldsymbol{X}_i\boldsymbol{\beta}^{(t+1)}_i\right)
\end{aligned}
$$

where M is total number of observation for all hurricanes.

Check whether $\sigma^{2'}$ is positive, if not, we reject $\sigma^{2'}$. Then, we randomly sample u from $U(0,1)$ and compare $\log(u)$ with the log ratio. If the $\log(u)$ is smaller, we accept $\sigma^{2'} = \sigma^{2^{(t+1)}}$.

Finally, we sample $\Sigma^{-1^{(t+1)}}$ from Wishart $\left(3d+3+m, \left(\boldsymbol{I} + \sum_{i=1}^{m}\left(\boldsymbol{\beta}_i^{(t+1)} - \boldsymbol{\mu}^{(t+1)}\right)\left(\boldsymbol{\beta}_i^{(t+1)} - \boldsymbol{\mu}^{(t+1)}\right)^T\right)^{-1}\right)$

```
componentwisemixedMH=function(a, a_sigma, beta_0,sigma_0,cov_0, nrep=10000){
  sigma_p = cov_0
  beta = beta_0
  mu = colMeans(beta)
  sigma = sigma_0
  theta_chain_mix = matrix(0, nrow = nrep, ncol = n_theta)
  for(i in 1:nrep){
    sigma_m = solve(sigma_p)
    # update beta by block of each hurricane
    for(j in 1:m){
      for(k in 1:5){
        prop = beta[j,]
        prop[k] = prop[k]+2*(runif(1)-0.5)*a[(k-1)*681+j]
        if(log(runif(1)) < (log_posterior_beta(prop, sigma, mu, sigma_p,j) - log_posterior_beta(beta[j,]
          beta[j,k] = prop[k]
        }
      }
    }
    # update mu
    beta_mean = colMeans(beta)
    mu = mvrnorm(1,beta_mean,Sigma = sigma_m/m)

    # update sigma
    beta_frame = matrix(rep(c(beta),times = rep_time), ncol = d)
    prop = sigma
    prop = prop+2*(runif(1)-0.5)*a_sigma
    if(log(runif(1)) < (log_posterior_sigma(beta_frame, prop) - log_posterior_sigma(beta_frame, sigma))
      sigma = prop
    }
    # update inverse covariance matrix
    beta_mu = t(beta)-mu
    S = solve(diag(d)+beta_mu%*%t(beta_mu))
    sigma_p = rWishart(1,df = 3*d+3+m,Sigma = S)
    sigma_p = apply(sigma_p,2,c)
    theta_chain_mix[i,] = c(c(beta),mu,sigma,sigma_p[lower.tri(sigma_p, diag = T)])
  }
  return(theta_chain_mix)
}
```

Part III: Implementation and Estimations

Starting Value and Search Window

```
## beta_i_0: MLR on ith hurricane and get the coefficient as beta_i_0
## mu_0: average over the beta_i_0 as the mu_0
## sigma_0: get the MSE of each MLR and take average as sigma_0
## sigma_m_0^-1: take the inverse of covariance matrix of beta_i_0 as the initial sigma_m_0^-1
```

```
starting_beta = df_mcmc %>%
  mutate(lm_fit = map2(.x = x_matrix, .y = y, ~lm(.y~.x[,-1], )),
         lm_0 = map_dbl(.x = lm_fit, ~coef(.x)[1]),
         lm_1 = map_dbl(.x = lm_fit, ~coef(.x)[2]),
         lm_2 = map_dbl(.x = lm_fit, ~coef(.x)[3]),
         lm_3 = map_dbl(.x = lm_fit, ~coef(.x)[4]),
         lm_4 = map_dbl(.x = lm_fit, ~coef(.x)[5]),
         sigma = map_dbl(.x = lm_fit, ~mean(.x$residuals^2))) %>%
  ungroup() %>%
  dplyr::select(lm_0:sigma)

sigma_0 = mean(starting_beta$sigma)

starting_beta = starting_beta%>%dplyr::select(-sigma) %>% as.matrix()
beta_0 = lm(wind_kt~wind_lag+lat_shift+long_shift+wind_shift, data = dt)
starting_beta[which(is.na(starting_beta[,2])),2] = coef(beta_0)[2]
starting_beta[which(is.na(starting_beta[,5])),5] = coef(beta_0)[5]

cov_0 = solve(cov(starting_beta))
```

```
## search window

ind_06 =  c(1,3,4,6,16,19,24,28,31,41,43,45,46,47,50,51,52,54,62,68,69,73,82,83,89,93:97,99,108,112,113

ind_12 = c(8,13,38,57,63,64,66,72,80,84,114,115,137,145,149,159,164,165,186,196,204,207,218,223,234,236

a2 = rep(0.08,m)
a2[ind_06] = 0.04
a2[ind_12] = 0.1
a3 = rep(1, m)
a3[203] = 0.8
a4 = rep(0.6,m)
a4[c(4,18,20,32,40,50,62,68,94,106,118,124,134,140,166,216,220,250,272,274,282,306,328,364, 376, 400, 4

a5 = rep(0.3,m)
ind_14 = c(3,6,12,15,18,21,24,30,36,39,42 ,48, 57 ,63,66,72,75,78,81,84,87,93,102, 108,114,120,126,132,
a5[ind_14] = 0.4
a = c(rep(1.1,m),a2,a3,a4,a5)
```

Implement 10000 MCMC iterations

```
set.seed(1971)
theta_chain = componentwisemixedMH(a, a_sigma = 2, beta_0 = starting_beta, sigma_0, cov_0, nrep=10000)
```

Acceptance Rate

```
load("mcmc_chain_1.RData")
load("mcmc_chain_2.RData")
load("mcmc_chain_3.RData")
load("mcmc_chain_4.RData")
load("mcmc_chain_5.RData")
load("mcmc_chain_6.RData")
```

Table 1: Range of Search Window and Acceptance Rate for each paraemter in MH Algorithm

	search window	acceptance rate (%)
$\boldsymbol{\beta}_0$	1.1	45.87-51.36
$\boldsymbol{\beta}_1$	0.04-0.1	31.67-63.68
$\boldsymbol{\beta}_2$	0.8-1	38.6-45.6
$\boldsymbol{\beta}_3$	0.5-0.6	33.2-61.32
$\boldsymbol{\beta}_4$	0.4-0.5	34.95-60.45
$\boldsymbol{\sigma}^2$	2	44.93

```
theta_chain = rbind(theta_chain_1, theta_chain_2, theta_chain_3,
                    theta_chain_4, theta_chain_5, theta_chain_6)

a_rate = apply(theta_chain, MARGIN = 2, numunique)/10000

a_rate_matrix = matrix(a_rate[1:n_beta], ncol = 5)

a_0_range = range(a_rate_matrix[,1])*100 # search window 1.1
a_1_range = range(a_rate_matrix[,2])*100 # search window: 0.04-0.1
a_2_range = range(a_rate_matrix[,3])*100 # search window: 0.8-1
a_3_range = range(a_rate_matrix[,4])*100 # search window:0.5-0.6
a_4_range = range(a_rate_matrix[,5])*100 # search window: 0.4-0.5
a_sigma = a_rate[sigma_ind]*100

df_a = tibble(parameter = c("$\\boldsymbol{\\beta}_0$", "$\\boldsymbol{\\beta}_1$",
                            "$\\boldsymbol{\\beta}_2$", "$\\boldsymbol{\\beta}_3$",
                            "$\\boldsymbol{\\beta}_4$", "$\\boldsymbol{\\sigma}^2$"),
              searchwindow = c("1.1",
                               "0.04-0.1",
                               "0.8-1",
                               "0.5-0.6",
                               "0.4-0.5",
                               "2"),
              acceptancerate = c(paste0(a_0_range[1],"-",a_0_range[2]),
                                 paste0(a_1_range[1],"-",a_1_range[2]),
                                 paste0(a_2_range[1],"-",a_2_range[2]),
                                 paste0(a_3_range[1],"-",a_3_range[2]),
                                 paste0(a_4_range[1],"-",a_4_range[2]),
                                 paste0(a_sigma)))

label_2 = c("","search window", "acceptance rate ($\\%$)")

df_a %>% kable(format = "latex",escape=FALSE, booktabs = TRUE, col.names = label_2,
               centering = T, vline = "|", linesep = c("\\addlinespace", "\\addlinespace",
                                                       "\\addlinespace", "\\addlinespace",
                                                       "\\addlinespace","\\addlinespace"),
               caption = "Range of Search Window and Acceptance Rate for each paraemter in MH Algorithm
```

Convergence and Distribution

Estimates (burn-in first 5500 runs)

```
## take average over 5501-10000 as estimates
bs_estimates = unname(apply(theta_chain[5501:10000,], 2, mean))


inv_cov_m = matrix(0,5,5)
inv_cov_m[lower.tri(inv_cov_m, diag = TRUE)] = bs_estimates[cov_m_ind]
inv_cov_m[upper.tri(inv_cov_m)] = t(inv_cov_m)[upper.tri(inv_cov_m)]


cov_m = solve(inv_cov_m)

## estimates
sigma = bs_estimates[3411]
mu = bs_estimates[3406:3410]
beta = matrix(bs_estimates[1:n_beta], ncol = d)
```

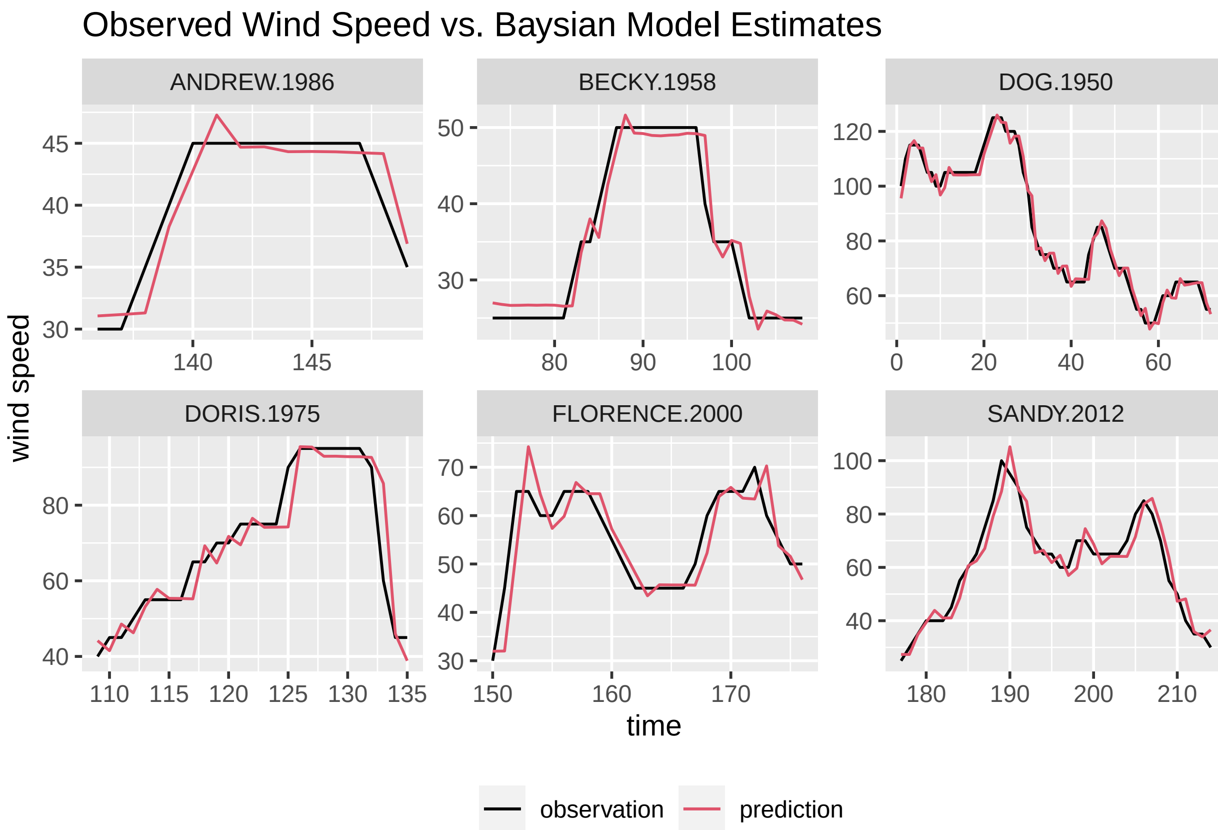
Part IV: Model Performance

```
dt_res = dt %>%
  dplyr::select(id, nature, season, month, wind_kt,wind_lag,lat_shift, long_shift, wind_shift) %>%
  nest(data = nature:wind_shift) %>%
  ungroup() %>%
  mutate(beta_0 = beta[,1],
         beta_1 = beta[,2],
         beta_2 = beta[,3],
         beta_3 = beta[,4],
         beta_4 = beta[,5]) %>%
  unnest(data) %>%
  mutate(wind_kt_pred = beta_0+beta_1*wind_lag+beta_2*lat_shift+beta_3*long_shift+beta_4*wind_shift)

## adj r-square and r-square are 0.96
r_square = (1-sum((dt_res$wind_kt_pred-dt_res$wind_kt)^2)/sum((dt_res$wind_kt-mean(dt_res$wind_kt))^2))
p_value = 1-pchisq(sum((dt_res$wind_kt_pred-dt_res$wind_kt)^2)/sigma, n_obs-n_beta)
adj_r_square = 1-(1-r_square)*(n_obs-1)/(n_obs-n_beta)

## visualize some hurricane prediction

dt_res %>% filter(id %in% c("DOG.1950","BECKY.1958", "DORIS.1975","ANDREW.1986","FLORENCE.2000", "SANDY
  mutate(time = 1:n()) %>%
  ggplot(aes(y = wind_kt, x = time))+
  geom_line(aes(color = "observation"))+
  geom_line(aes(y = wind_kt_pred, color = "prediction"))+
  scale_colour_manual(name = NULL,
values = c( "observation" = 1, "prediction" = 2))+
  facet_wrap(.~id, nrow = 2, scales = "free")+
  labs(x = "time",
       y = "wind speed",
       title = "Observed Wind Speed vs. Baysian Model Estimates")
```



Observed Wind Speed vs. Baysian Model Estimates
ANDREW.1986
BECKY.1958
DOG.1950
DORIS.1975
FLORENCE.2000
SANDY.2012
wind speed
time
observation
prediction