Task 4

Task 2: With the estimated model parameters and covariate values, you can calculate the predicted wind speed for each time point using the model equation. This way, you can track the hurricane and compare the predicted wind speeds with the actual wind speeds recorded during the hurricane. Please evaluate how well the estimated Bayesian model can track individual hurricanes.

Solution

Import parameters from task 1 & 2.

```
# load parameters
beta_list = read.csv("./data/B_list_partial.csv")
mu_list = read.csv("./data/mu_list.csv")
Sigma_list = read.csv("./data/Sigma_matrix_list.csv")
gamma_list = read.csv("./data/gamma_list.csv")
sigma_list = read.csv("./data/sigma_list.csv")$x

# burn in the MC chains. change this based on the resulting plots
burn = 500
# index of useful samples (used for estimates & CIs)
index = (burn + 1):10000
beta_sample = beta_list[index,]
mu_sample = mu_list[index,]
gamma_sample = gamma_list[index,]
```

Implement the prediction process of wind speed for each hurricane in R.

```
# final parameters to be used
para_beta = rbind(colMeans(beta_sample))
para_mu = rbind(colMeans(mu_sample))
para_gamma = rbind(colMeans(gamma_sample))

# prediction function
Windspeed_Predict = function(index_hurricane, index_time){
  predict_speed =
    Z[[index_hurricane]][index_time,] %*% para_beta[((index_hurricane - 1) * 5 + 1):((index_hurricane -
    (X %*% t(para_gamma))[index_hurricane, ]
  return(predict_speed)
}

# functional test
Windspeed_Predict(1, 26)
```

```
##          [,1]
## [1,] 26.54175
```

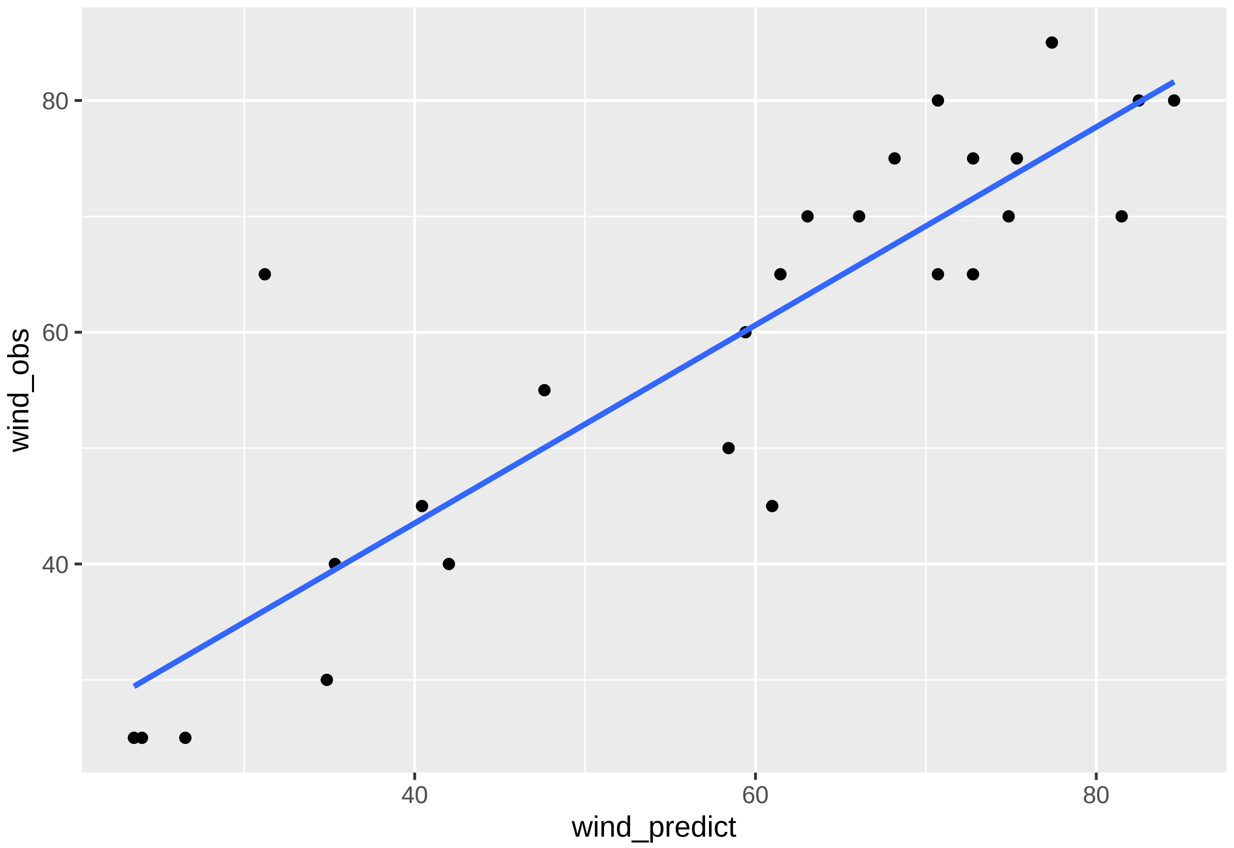
```
# initialize prediction table
Y_table = split(Training$Wind.kt, Training$ID) %>%
  lapply(function(x) x[-c(1:2)]) %>%
  lapply(as.data.frame) %>%
```

```r
  lapply(function(df) {
    df$wind_obs = df$`X[[i]]`
    df$wind_predict = df$wind_obs
    df = as.matrix(df)
    subset(df, select = c("wind_obs", "wind_predict"))
  })

# updating prediction table
for (i in 1:length(names(Z))) {
 for (j in 1:nrow(Z[[i]])) {
 Y_table[[i]][, 2][j] = Windspeed_Predict(i, j)
  }
}
```

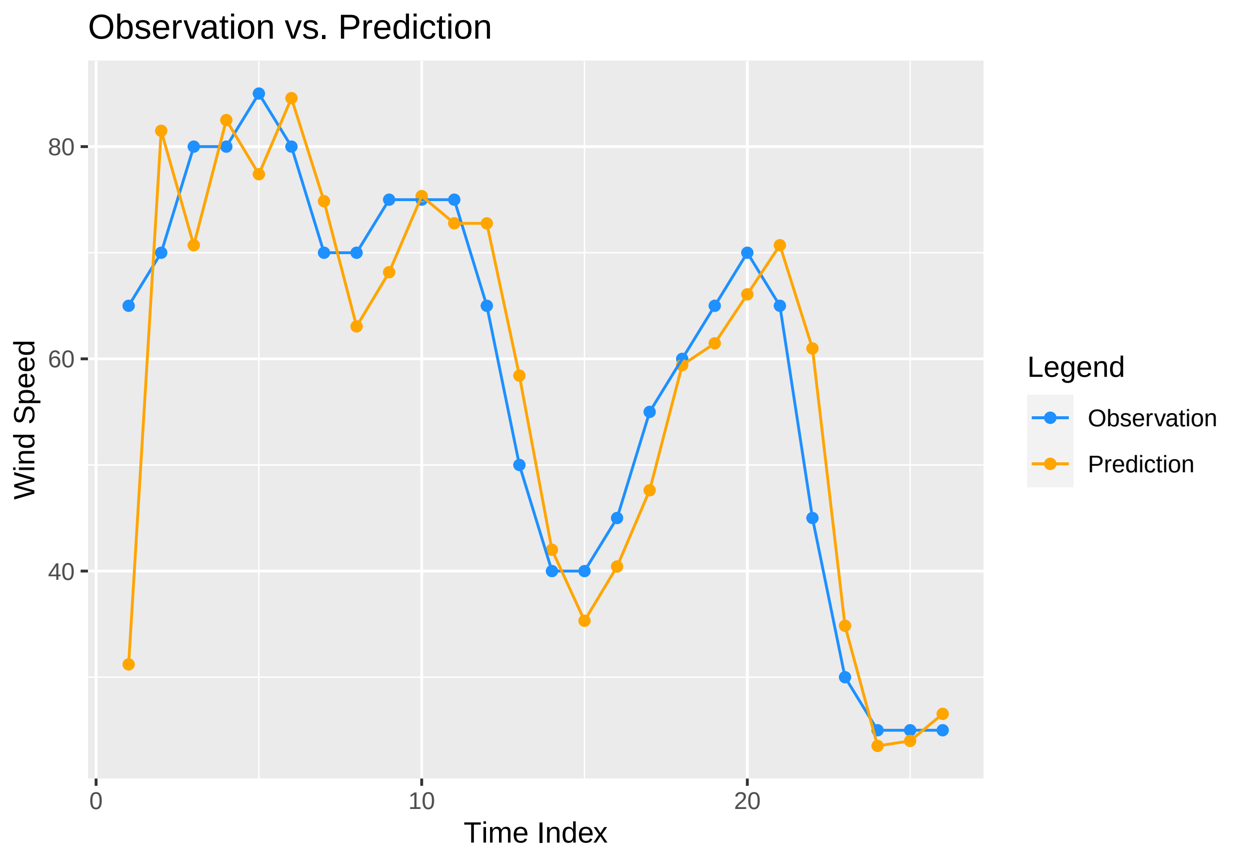
Data visualization for prediction results.

```r
hurri_res = data.frame(ID = "example",
                       RMSE = 0)
for (i in 1:length(names(Z))) {
  RMSE = sqrt(mean((Y_table[[i]][,1] - Y_table[[i]])^2))
  new_row = c(names(Z)[i], RMSE)
  hurri_res = rbind(hurri_res, new_row)
}

hurri_res = hurri_res[-1, ]
hurri_res$RMSE = as.numeric(hurri_res$RMSE)
hurricane_info = Training %>%
  group_by(ID) %>%
  slice(1) %>%
  dplyr::select(ID, Season, Month, Nature) %>%
  ungroup(ID)
hurri_res = left_join(hurri_res, hurricane_info, by = "ID")

# subset of non-NAs
hurri_res = hurri_res[1:90, ]

example_hurri = as.data.frame(Y_table[[1]])
example_hurri$index = 1:nrow(example_hurri)

# example visualization of index 1 hurricane
ggplot(example_hurri, aes(x = wind_predict, y = wind_obs)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

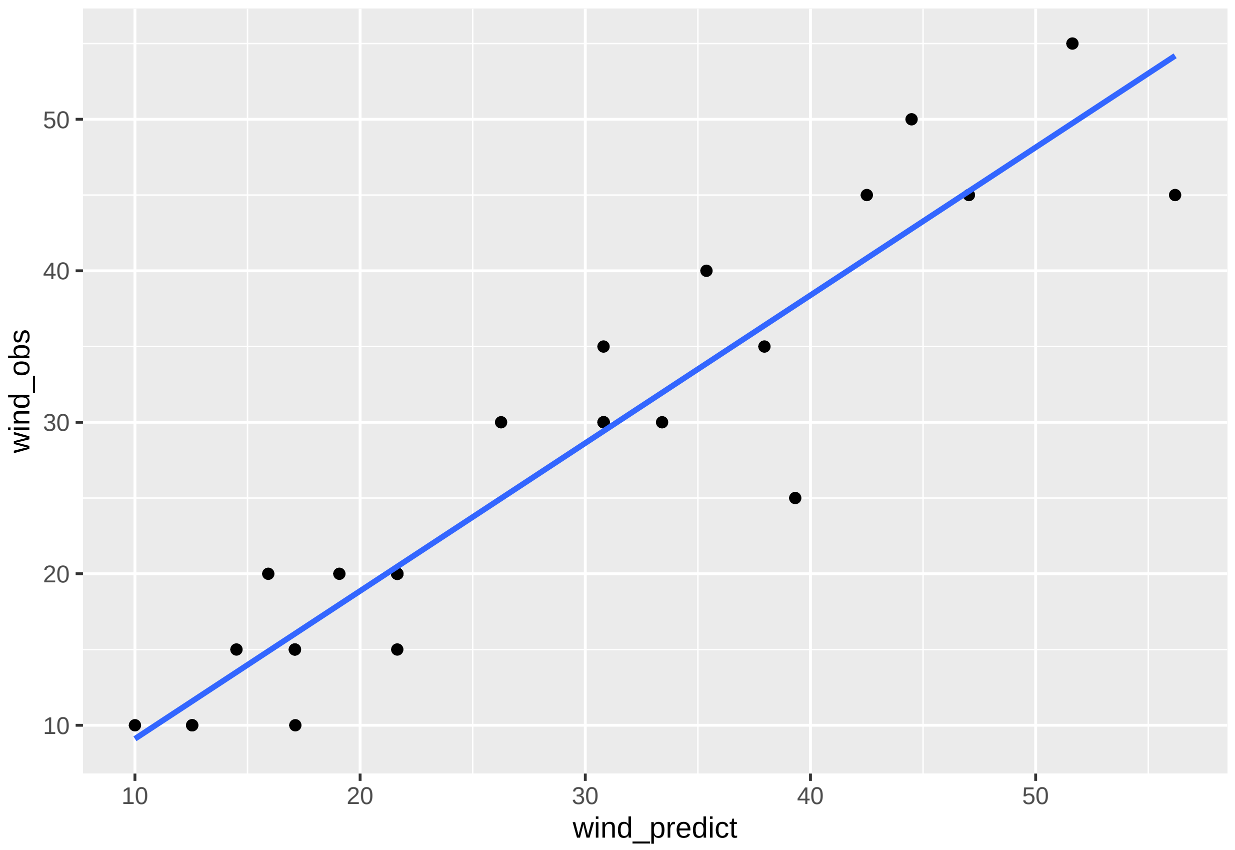


```
ggplot(example_hurri, aes(x = index)) +
  geom_point(aes(y = wind_obs, color = "Observed")) +
  geom_point(aes(y = wind_predict, color = "Predicted")) +
  geom_line(aes(y = wind_obs, color = "Observed")) +
  geom_line(aes(y = wind_predict, color = "Predicted")) +
  labs(title = "Observation vs. Prediction", x = "Time Index", y = "Wind Speed") +
  scale_color_manual(name = "Legend",
                     values = c("Observed" = "#1E90FF", "Predicted" = "orange"),
                     labels = c("Observation", "Prediction"))
```

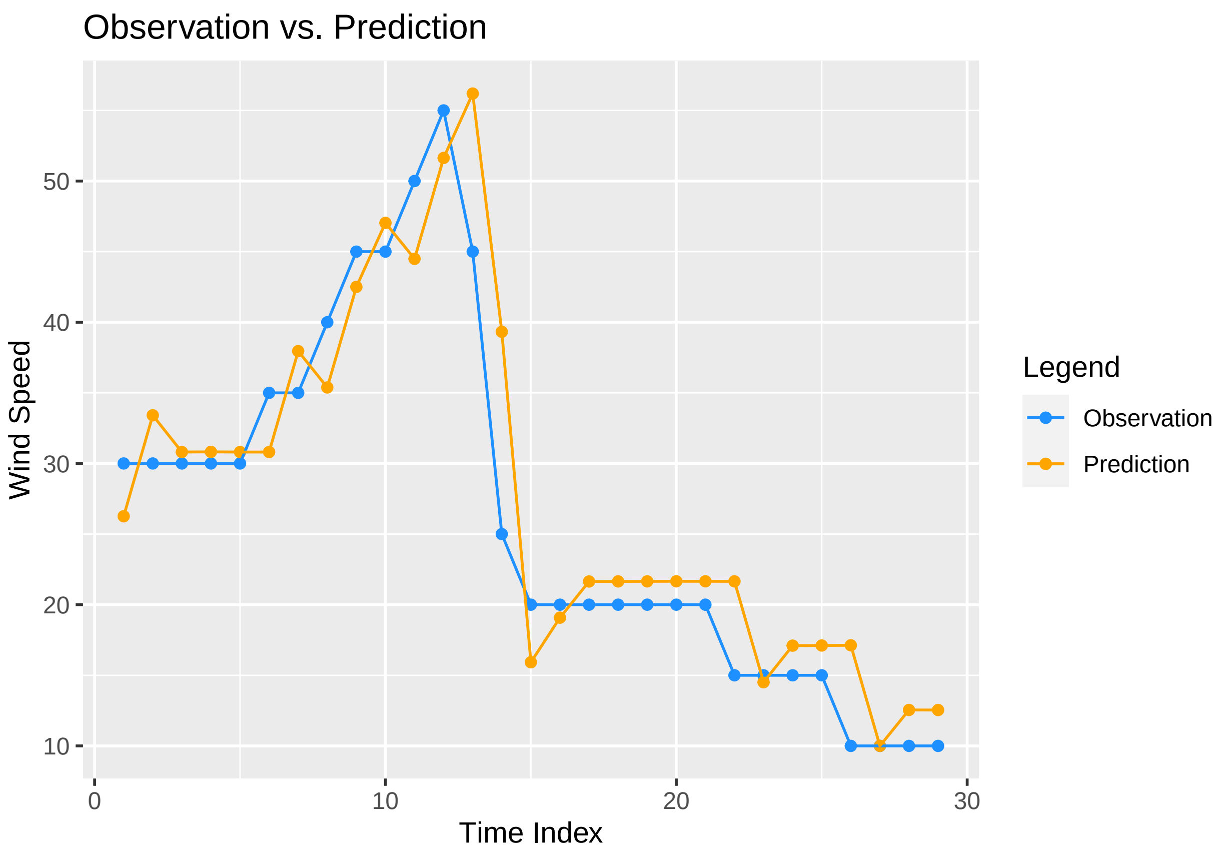


```r
example_hurri = as.data.frame(Y_table[[10]])
example_hurri$index = 1:nrow(example_hurri)
# example visualization of index 2 hurricane
ggplot(example_hurri, aes(x = wind_predict, y = wind_obs)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(example_hurri, aes(x = index)) +
  geom_point(aes(y = wind_obs, color = "Observed")) +
  geom_point(aes(y = wind_predict, color = "Predicted")) +
  geom_line(aes(y = wind_obs, color = "Observed")) +
  geom_line(aes(y = wind_predict, color = "Predicted")) +
  labs(title = "Observation vs. Prediction", x = "Time Index", y = "Wind Speed") +
  scale_color_manual(name = "Legend",
                     values = c("Observed" = "#1E90FF", "Predicted" = "orange"),
                     labels = c("Observation", "Prediction"))
```


Observation vs. Prediction
Wind Speed
Time Index
Legend
Observation
Prediction
10
20
30
40
50
0
10
20
30