Final Report

Group 1: Jiawen Chen, Brooke Felsheim, Elena Kharitonova, Xinjie Qian, and Jairui Tang

4/29/2022

Introduction

As human populations are rising across the world, so is the proportion of people that live in urban areas. Estimates from the *UN World Urbanization Prospects* indicate that over 4.2 billion people (55% of the global population) currently live in urban areas, and by 2050, an additional 2.5 billion people (68% of the global population) could be living in urban areas[1]. More people living in urban areas calls for more space-, cost-, and energy-efficient systems of transportation as an alternative to cars. One such promising transportation alternative is the implementation of bicycle sharing programs.

Bicycle sharing programs are transportation schemes that allow individuals to rent bicycles on a short-term basis for either a set rate or for free. Most bicycle sharing programs have many computer-controlled bicycle rack "hubs" dispersed across a city that keep bikes locked and release them for use when a user enters the appropriate information/payment from a station or an app (Figure 1). A user can then ride the bike and return it to any other bicycle hub that is part of the same program. Many cities across the world have begun implementing bicycle sharing programs, including Chapel Hill, which has a Tar Heel Bikes sharing system[2]. Systems like these provide convenient, inexpensive, and eco-friendly transportation options for individuals residing in a city.

Successful implementations of bike sharing programs depend on proper management of these systems. It is important for a bike sharing program to provide a stable supply of rental bikes to its population so its users feel that they can rely on the system for their transportation needs. The analysis of bike sharing data allows for a better understanding of the demand of rental bikes in a city, which, in turn, can help inform a city about how to provide appropriate supplies of rental bikes for its population.

```r
library(devtools)
load_all("package/bikeSharing")
set.seed(1)
```

```r
str(london)
```

```
## 'data.frame':    2185 obs. of  14 variables:
##  $ Date        : chr  "01-01" "01-01" "01-01" "01-01" ...
##  $ Hour_chunks : Factor w/ 3 levels "[0,8)","[8,16)",..: 1 1 2 2 3 3 1 1 2 2 ...
##  $ Day         : num  1 1 1 1 1 1 2 2 2 2 ...
##  $ Is_weekend  : Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 2 1 2 ...
##  $ Is_holiday  : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 2 1 2 1 ...
##  $ Season      : Factor w/ 4 levels "Spring","Summer",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Min_temp    : num  3 5 3 5 3 5 1 9 1 9 ...
##  $ Max_temp    : num  9 10 9 10 9 10 6 11.5 6 11.5 ...
##  $ Min_humidity: num  76 81 76 81 76 81 71 82 71 82 ...
##  $ Max_humidity: num  87 93 87 93 87 93 93 94 93 94 ...
```

[1] United Nations, Department of Economic and Social Affairs, Population Division (2018). World Urbanization Prospects: The 2018 Revision, Online Edition.

[2] https://move.unc.edu/bike/bikeshare/

```
##  $ Year        : chr  "Year 1" "Year 2" "Year 1" "Year 2" ...
##  $ Wind_speed  : num  2.48 3.65 4.83 4.08 6.63 ...
##  $ Rain_or_snow: Factor w/ 2 levels "0","1": 1 2 2 2 2 2 1 2 1 2 ...
##  $ Bike_count  : int  2715 2962 4460 2450 2622 1009 438 475 7756 4263 ...
```

```r
london_train <- london[london$Year == "Year 1",]
london_test <- london[london$Year == "Year 2",]
```

Methods

Negative Binomial Generalized Linear Mixed Model

Random Forest

Results

Negative Binomial Generalized Linear Mixed Model

```r
glmm_fit <- MCEM_algorithm( beta_initial = c(8.3, 1.5, 1.5, -0.25, -0.50, 0,
                                             0, -0.25, 0, 0, 0, 0, 0, -0.25),
                        theta_initial = 10,
                        s2gamma_initial = 0.2,
                        M = 1000,
                        burn.in = 200,
                        tol = 10^-4,
                        maxit = 100,
                        data = london_train
                        )
```

```r
str(glmm_fit)
```

```
## List of 7
##  $ beta     : num [1:14] 8.353 1.534 1.415 -0.337 -0.393 ...
##  $ s2gamma  : num 0.0296
##  $ theta    : num 18.4
##  $ eps      : num 5.15e-05
##  $ qfunction: num -9520
##  $ day_ranef: num [1:365] 0.0653 -0.398 -0.5165 -0.2612 -0.0374 ...
##  $ iter     : num 23
```

```r
glmm_model_fit(glmm_fit, london_train, scale_to_reference_mean = "no",
               reference = london)
```

```
##       RMSE      MAE        R2
## 1 1886.267 1291.106 0.8831618
```

```r
glmm_model_fit(glmm_fit, london_test, scale_to_reference_mean = "no",
               reference = london)
```

```
##       RMSE      MAE        R2
## 1 2491.293 1647.064 0.8142036
```

```r
glmm_model_fit(glmm_fit, dc, scale_to_reference_mean = "yes",
               reference = london)
```

```
##      RMSE     MAE       R2
## 1 845.741 605.217 0.521788
```

```r
glmm_model_fit(glmm_fit, seoul, scale_to_reference_mean = "yes",
               reference = london)
```

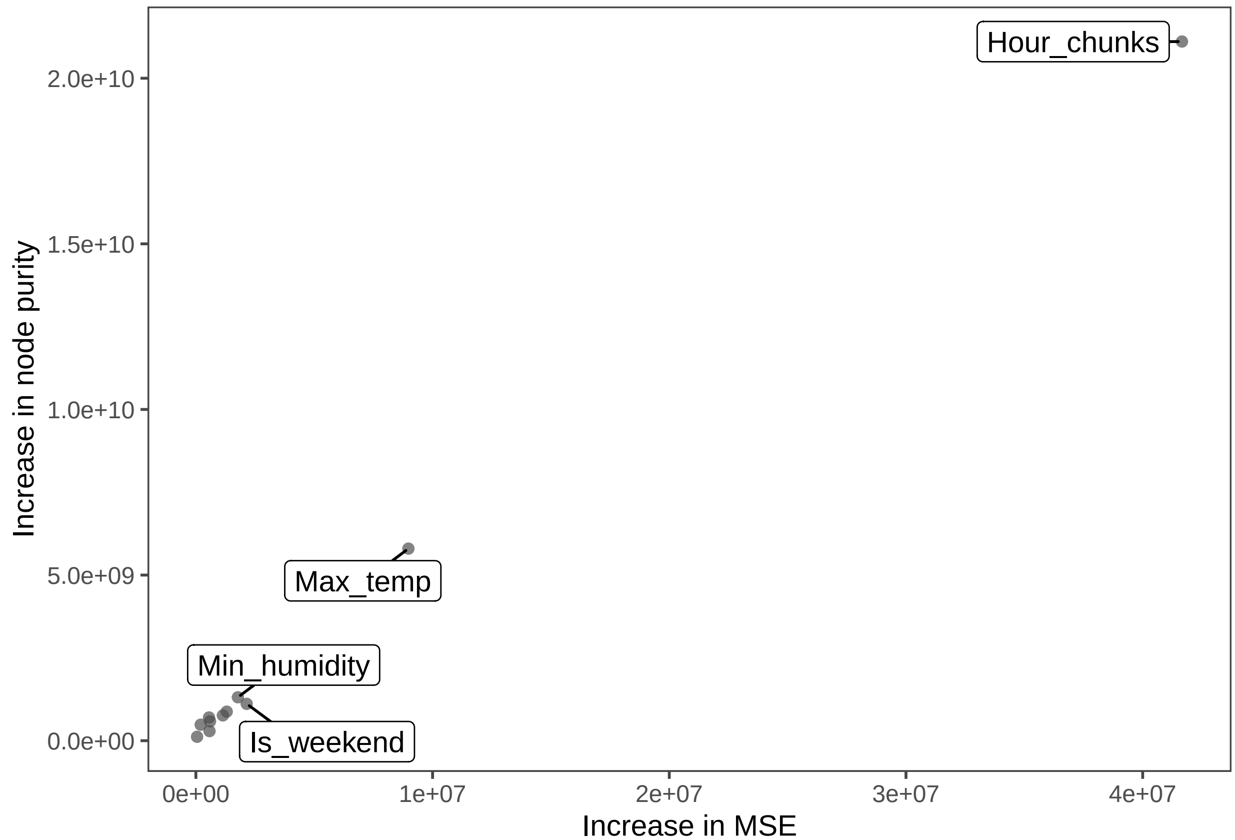
```
##       RMSE      MAE        R2
## 1 3519.413 2719.021 0.4999935
```

Random Forest

```r
rf_fit <- train_random_forest(data = london_train)
rf_fit
```

```
## Random Forest
##
## 1095 samples
##   11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 876, 878, 875, 875, 876
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##    2    2262.068  0.8841247  1721.956
##    6    1797.963  0.8980543  1174.833
##   11    1789.815  0.8964189  1167.026
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 11.
```

```r
plot_rf_importance(london_train)
```




```
rf_model_fit(rf_fit, london_train, scale_to_reference_mean = "no",
             reference = london)
```

```
##       RMSE      MAE        R2
## 1 714.5723 445.2132 0.9838003
```

```
rf_model_fit(rf_fit, london_test, scale_to_reference_mean = "no",
             reference = london)
```

```
##      RMSE      MAE        R2
## 1 1780.33 1172.378 0.9063308
```

```
rf_model_fit(rf_fit, dc, scale_to_reference_mean = "yes",
             reference = london)
```

```
##       RMSE      MAE       R2
## 1 737.5336 524.7186 0.641438
```

```
rf_model_fit(rf_fit, seoul, scale_to_reference_mean = "yes",
             reference = london)
```

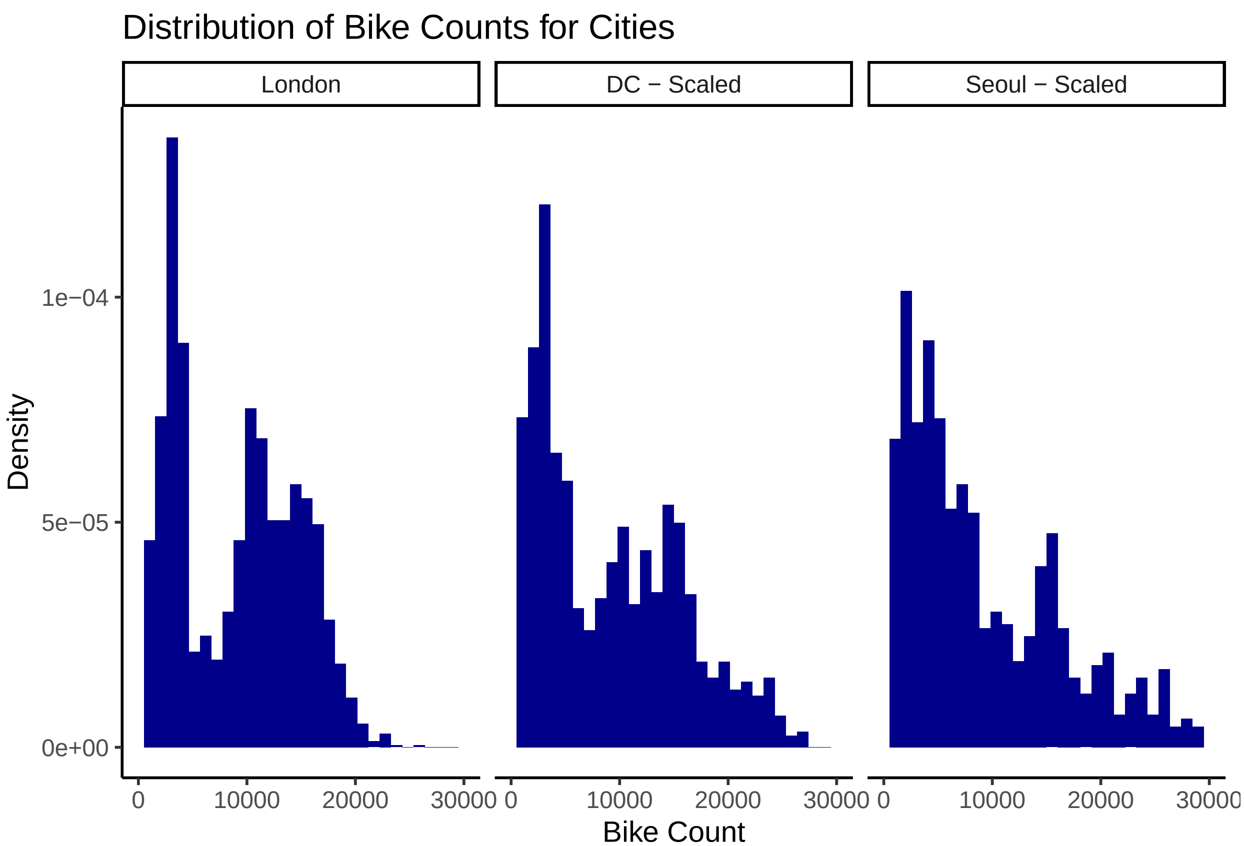
```
##       RMSE      MAE       R2
## 1 3465.028 2630.914 0.544965
```

```
seoul$city = "Seoul - Scaled"
london_train$city = "London"
london_test$city = "London"
dc$city = "DC - Scaled"
seoul$city2 = "Seoul - Scaled"
london_train$city2 = "London - Train"
```

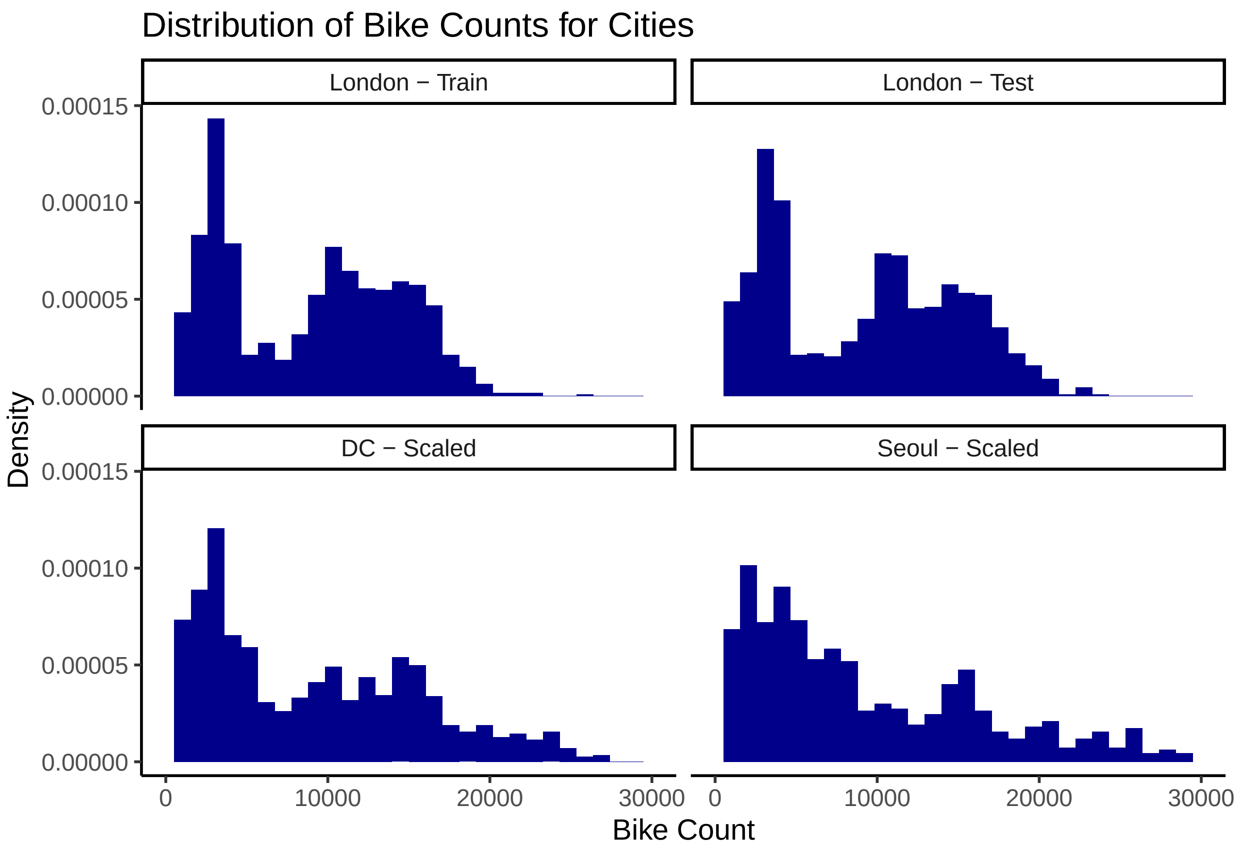
```
london_test$city2 = "London - Test"
dc$city2 = "DC - Scaled"
scale_seoul = mean(seoul$Bike_count)/
  mean(london$Bike_count)
scale_dc = mean(dc$Bike_count)/
  mean(london$Bike_count)
seoul$Bike_count2 = seoul$Bike_count / scale_seoul
london_test$Bike_count2 = london_test$Bike_count
london_train$Bike_count2 = london_train$Bike_count
dc$Bike_count2 = dc$Bike_count / scale_dc
seoul$Year = 1


all_data = rbind(seoul, london_train, london_test, dc)
all_data$city = factor(all_data$city,
                       levels = c("London",  "DC - Scaled", "Seoul - Scaled"))
all_data$city2 = factor(all_data$city2,
                        levels = c("London - Train", "London - Test",
                                   "DC - Scaled", "Seoul - Scaled"))

ggplot(all_data, aes(x = Bike_count2)) +
  geom_histogram(fill = "dark blue", aes(y = stat(density))) +
  facet_wrap(~city) + theme_classic() + xlim(0,30000) +
  labs(title = "Distribution of Bike Counts for Cities",
       y = "Density", x = "Bike Count")
```



```r
ggplot(all_data, aes(x = Bike_count2)) +
  geom_histogram(fill = "dark blue", aes(y = stat(density))) +
  facet_wrap(~city2) + theme_classic() + xlim(0,30000) +
  labs(title = "Distribution of Bike Counts for Cities",
       y = "Density", x = "Bike Count")
```




```r
mean(london_test$Bike_count)
```

```
## [1] 9286.037
```

```r
mean(london_train$Bike_count)
```

```
## [1] 8913.796
```

Discussion