Reshuffling Strategy

Lee, Woo Chan

11/23/2021

Defining top 20 stations (highest variance of availability)

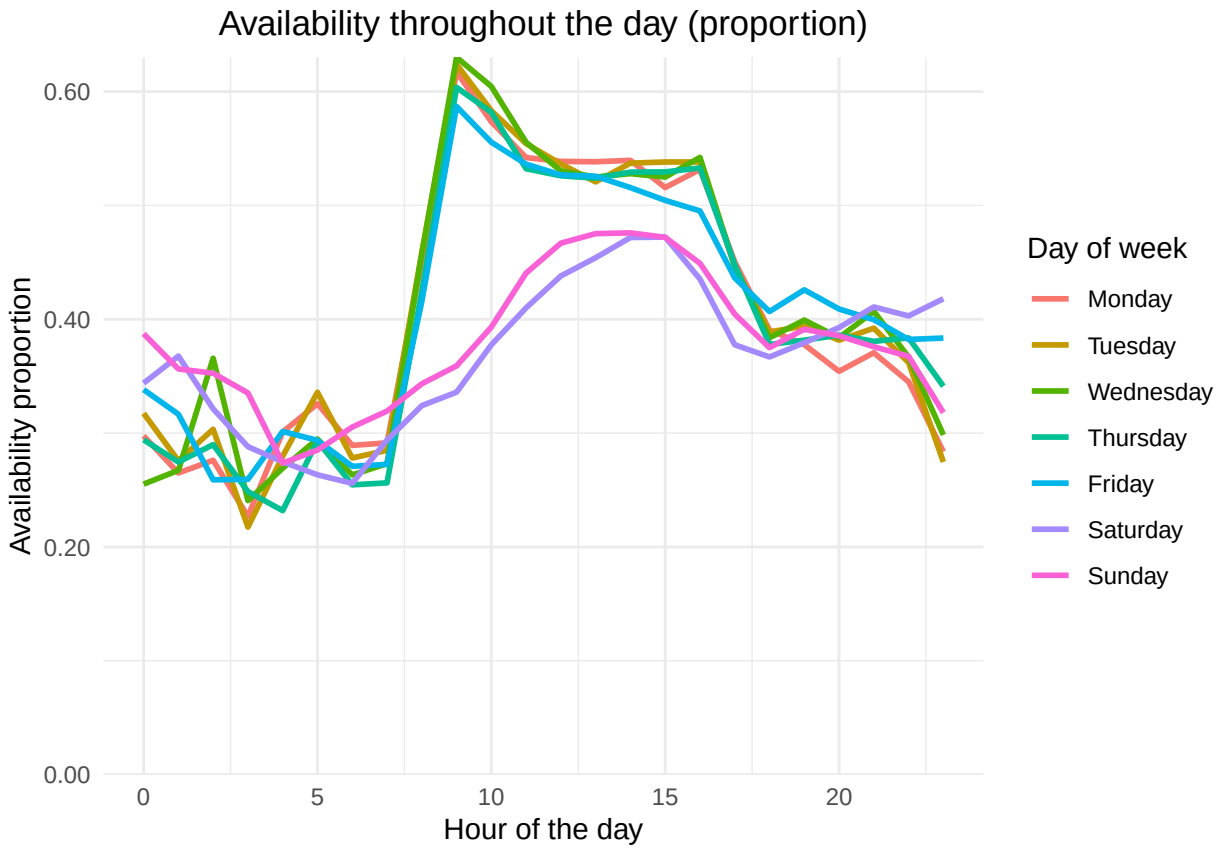
Generating the availability plot throughout the day

```
bike1 <- bike %>%
  arrange(date) %>%
  dplyr::select(date, is_weekend, reshuffle, capacity, availability_p) %>%
  mutate(
    month = month(date),
    hour_of_day = hour(date),
    day_of_week = weekdays(date)
    ) %>% filter(reshuffle == 0) %>%
  group_by(day_of_week, hour_of_day) %>%
  summarise(avail_p = mean(availability_p))
```

```
## `summarise()` has grouped output by 'day_of_week'. You can override using the `.groups` argument.
```

```
# Reordering day of week factor levels correctly
bike1$day_of_week <- factor(bike1$day_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "
```

```
# ggplot for availability curve throughout the day, each line representing the day of week
# We can see that weekdays and weekends show distinctive patterns close enough so that they could be gr
ggplot(bike1, aes(x = hour_of_day, y = avail_p, color=day_of_week)) +
  geom_line(size=1) +
  ggtitle("Availability throughout the day (proportion)") +
  ylab("Availability proportion") +
  xlab("Hour of the day") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(color='Day of week') +
  scale_y_continuous(label = comma, expand = c(0, 0), limits = c(0, NA))
```



Availability plot, but with weekdays and weekends grouped

```
bike2 <- bike %>%
  arrange(date) %>%
  dplyr::select(date, is_weekend, reshuffle, capacity, availability_p) %>%
  mutate(
    month = month(date),
    hour_of_day = hour(date),
    day_of_week = weekdays(date),
    week = ifelse(day_of_week == "Saturday" | day_of_week == "Sunday", "weekend", "weekday")
    ) %>%
  group_by(week, hour_of_day) %>%
  summarise(avail_p = mean(availability_p))
```

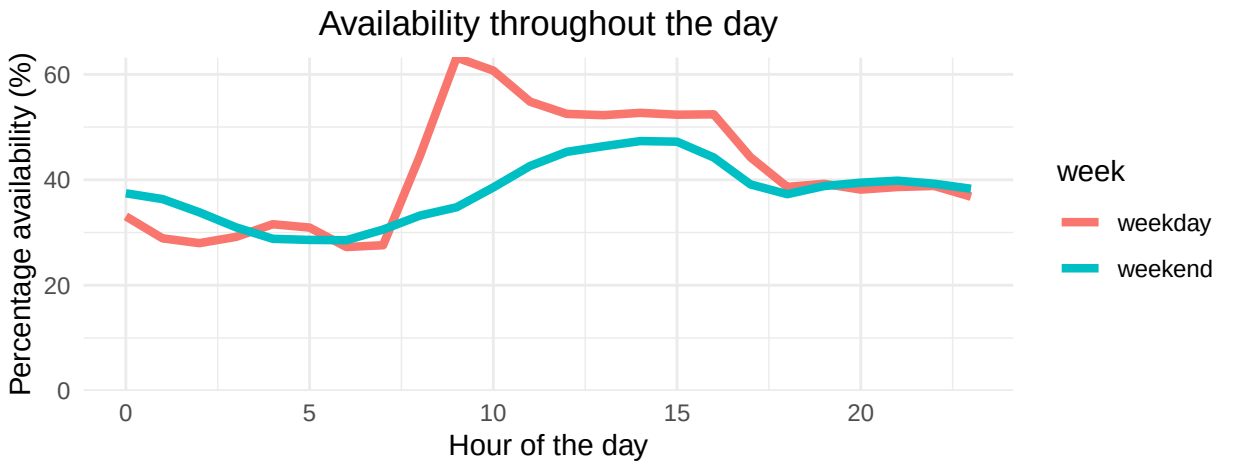
```
## `summarise()` has grouped output by 'week'. You can override using the `.groups` argument.
```

Weekday: The lowest availability is at 0.2 around 6am. This is probably due to the fact that capital share wants the stations to have enough space for the huge influx of riders to come in during rush hours.

Weekend: slight rise during afternoon to evening time, but not caused by reshuffle. Mostly caused by riders. Availability drops close to midnight, considering a lot of people take bikes to go back home.

```
# ggplot for availability plot weekend vs weekday
ggplot(bike2, aes(x = hour_of_day, y = avail_p * 100, color=week)) +
  geom_line(size=1.5) +
  labs(title = "Availability throughout the day") +
  ylab("Percentage availability (%)") +
```

```
  xlab("Hour of the day") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(label = comma, expand = c(0, 0), limits = c(0, NA))
```

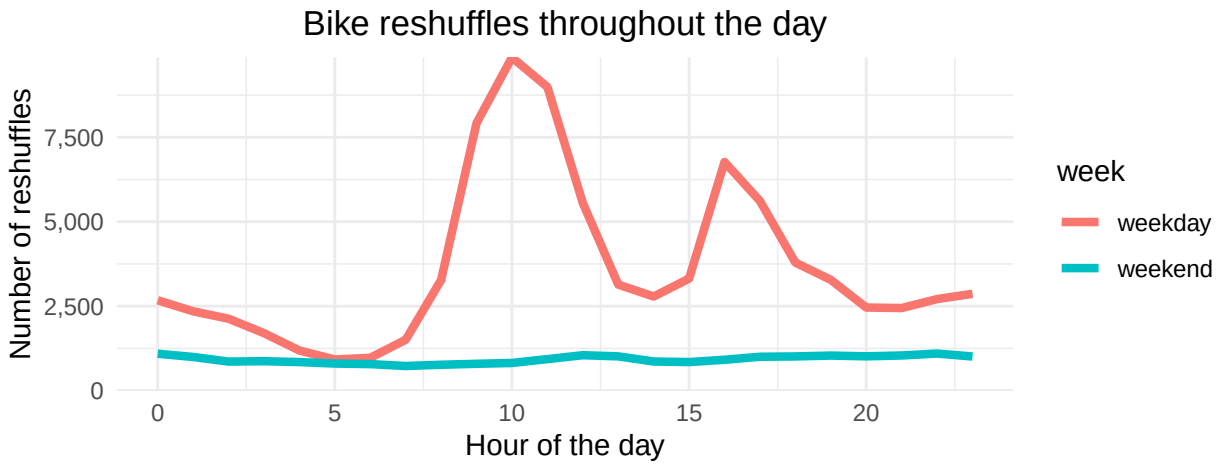


Reshuffling activities throughout the day

```
# Create apprORpriate dataframe to plot reshuffling activities
bike3 <- bike %>%
  arrange(date) %>%
  dplyr::select(date, is_weekend, reshuffle, capacity, availability_p) %>%
  mutate(
    month = month(date),
    hour_of_day = hour(date),
    day_of_week = weekdays(date),
    week = ifelse(day_of_week == "Saturday" | day_of_week == "Sunday", "weekend", "weekday")
    ) %>%
  filter(reshuffle == 1) %>%
  group_by(week, hour_of_day) %>%
  summarise(n = n())
```

```
## `summarise()` has grouped output by 'week'. You can override using the `.groups` argument.
```

```
# ggplot showing reshuffling activities throughout the day (Weekends and weekdays grouped)
ggplot(bike3, aes(x = hour_of_day, y = n, color=week)) +
  geom_line(size=1.5) +
  labs(title = "Bike reshuffles throughout the day") +
  ylab("Number of reshuffles") +
  xlab("Hour of the day") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(label = comma, expand = c(0, 0), limits = c(0, NA))
```

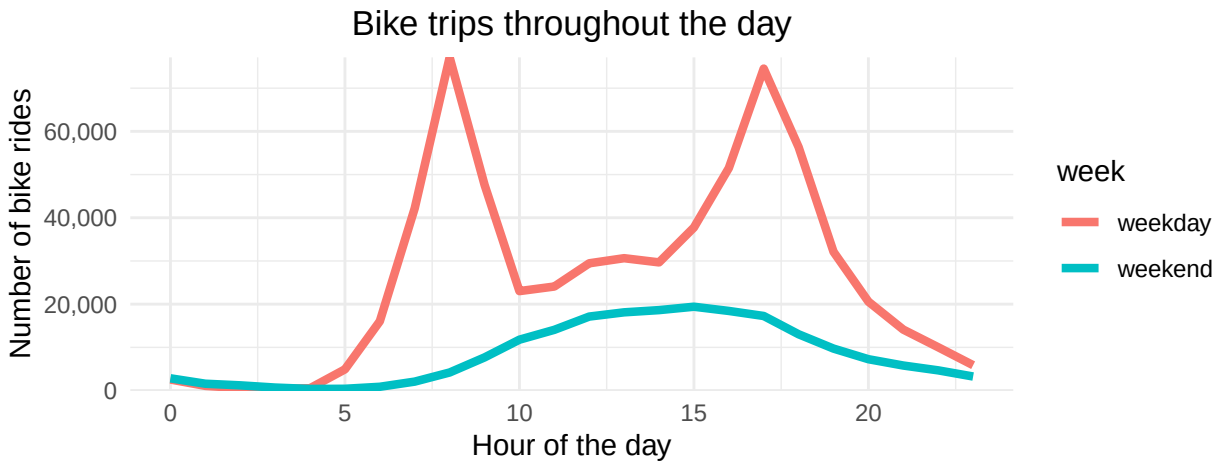


Number of bike activities

```
# Create data frame that will be used to plot total bike activities throughout the day
bike4 <- bike %>%
  arrange(date) %>%
  dplyr::select(date, is_weekend, reshuffle, capacity, availability_p) %>%
  mutate(
    month = month(date),
    hour_of_day = hour(date),
    day_of_week = weekdays(date),
    week = ifelse(day_of_week == "Saturday" | day_of_week == "Sunday", "weekend", "weekday")
    ) %>%
  filter(reshuffle == 0) %>%
  group_by(week, hour_of_day) %>%
  summarise(n = n())
```

```
## `summarise()` has grouped output by 'week'. You can override using the `.groups` argument.
```

```
# ggplot to show total bike activities throughout the day
ggplot(bike4, aes(x = hour_of_day, y = n, color=week)) +
  geom_line(size=1.5) +
  labs(title = "Bike trips throughout the day") +
  ylab("Number of bike rides") +
  xlab("Hour of the day") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(label = comma, expand = c(0, 0), limits = c(0, NA))
```



Logistic Regression on reshuffle

```
reshuffle_mod <- glm(reshuffle ~ availability_p, family=binomial(link='logit'),data=bike)
```

We can see from the coefficients below - unit increase in availability results in an increase in log odds of reshuffle. More specifically, a 0.1 increase in availability proportion in a station, increases the odds of reshuffling by approximately 3.7%

exp(0.32) = 1.37 -> unit increase leads to 37%, but 0.1 increase leads to 3.7%.

Makes sense because the stations we are looking at are areas with concentrated office buildings and busy urban areas, where reshuffling of bikes is more likely to occur when the availability at a station is high, during rush hours when there is a huge influx of riders that come in.

```
summary(reshuffle_mod)
```

```
##
## Call:
## glm(formula = reshuffle ~ availability_p, family = binomial(link = "logit"),
##     data = bike)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5401  -0.5102  -0.4923  -0.4766   2.1330
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.166439   0.006457 -335.49   <2e-16 ***
## availability_p  0.315274   0.011976   26.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 680934  on 943588  degrees of freedom
## Residual deviance: 680243  on 943587  degrees of freedom
```

```
## AIC: 680247
##
## Number of Fisher Scoring iterations: 4
```

Most likely the reshufflings are done to re-distribute and scatter the bikes away from these stations to neighboring ones. In order to find out, we produced a table showing the average availability proportions for reshuffles that took away bikes (-1) vs brought in bikes (+1).

```
bike_reshuffle <- bike %>% filter(reshuffle == 1)
bike_reshuffle$act <- factor(bike_reshuffle$act, levels = c("1", "-1"))
```

We can see that reshufflings that take out bikes (-1) are more likley to happen when the availability percentage at a station is high. Reshufflings that bring in bikes (+1) are more likely to happen when availability is lower (On average).

```
# Producing tables to compare two types of reshuffles (bringing in bikes vs taking out bikes)
bike_reshuffle2 <- bike_reshuffle %>%
  group_by(act) %>%
  summarise(avg_availability_p = mean(availability_p))

bike_reshuffle2
```

```
## # A tibble: 2 x 2
##   act   avg_availability_p
##   <fct>              <dbl>
## 1 1                  0.396
## 2 -1                 0.488
```

Our hypothesis was correct. As availability in a station increases, the odds of a reshuffling happening increases as well. And the majority of those reshuffles that happen at the peak availability, is when bikes are taken out from those stations.

This means that capital bikeshare does try to take bikes away after the morning rush hours when there is a peak of bike availability in the stations. And they also try to bring back bikes when availabilities start to go low. So the reshuffling strategy that capital bikeshare is implementing right now is in fact in the right direction.

```
test <- read_csv("/Users/lee14257/Development/CMU/Perspectives_in Data_Science/Project/2019/test.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 10000 Columns: 11
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): month, logi, pred
## dbl  (7): ...1, is_holiday, capacity, is_weekday, PRCP, TAVG, corres
## time (1): h_m
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
test
```

```
## # A tibble: 10,000 x 11
##     ...1 is_holiday capacity month   is_weekday  PRCP  TAVG logi  h_m    corres
##    <dbl>      <dbl>    <dbl> <chr>        <dbl> <dbl> <dbl> <chr> <time>  <dbl>
##  1    10          0       31 January          1  0       46 Yes   19:30       1
##  2    20          0       31 January          1  0       46 Yes   15:30       1
##  3    27          0       31 January          1  0       47 Yes   19:30       1
##  4    29          0       31 January          1  0       47 Yes   20:30       1
##  5    41          0       31 January          1  0       47 Yes   13:00       1
##  6    60          0       31 January          1  0.04    43 Yes   06:30       1
##  7    82          0       31 January          0  0.04    48 Yes   13:00       1
##  8   100          0       31 January          0  0       50 Yes   13:00       1
##  9   124          0       31 January          1  0       39 Yes   18:00       1
## 10   188          0       31 January          1  0       33 Yes   20:30       1
## # ... with 9,990 more rows, and 1 more variable: pred <chr>
```