Using Turnstile Data To Forecast Student Worker Staffing

Chad Madding, Shane Weinstock

April 11, 2020

### Clients

This report is being prepared for the management of the Dedman Center for Lifetime Sports at Southern Methodist University.

### Directive

We have been tasked with providing general usage of the facilities in the form of entry data. The main goal will be to use ID swipes as a response variable to assist in hiring patters for the facilities’ student staff. It is the goal of the report to assist in the spotting of trends and thus correctly identify staffing needs. Forecasting and general knowledge about building usage will also assist with budgeting needs.

### Data

The data provided to us has been collected from three turnstiles at the Dedman Center for Lifetime Sports on the campus of Southern Methodist University. The data set is a record of the time of swipe, the turnstile used, and an anonymized student ID number. The data was collected from January 2nd, 2019 through March 11th, 2020, and consists of 414,156 entries. Entry and error swipes are included in the data. It should be noted that the turnstiles are only used to enter the facility, and no swipe is required to exit the building. We have also collected hourly weather data from NOAA for the same period to assist in forecasting.

To get ready for merging the temperature data we needed to clean up the hours in the SMU data.

#Group hours  
SMUSwipe$TempTime <- format(SMUSwipe$LDT,"%Y-%m-%d")  
SMUSwipe$Hour <-ifelse(SMUSwipe$Hour <= '04','04',  
 ifelse(SMUSwipe$Hour <= '08','08',  
 ifelse(SMUSwipe$Hour <= 12, 12,  
 ifelse(SMUSwipe$Hour <= 16, 16,  
 ifelse(SMUSwipe$Hour <= 20, 20,  
 ifelse(SMUSwipe$Hour <= 24, 24))))))  
SMUSwipe$TempTime<-paste(SMUSwipe$TempTime, SMUSwipe$Hour, sep=" ")

We made the data the same format for the weather data as well as in the SMU information.

#Create a TempTime in HourlyWeather to merge Temperature data  
HourlyWeather$TempTime <- format(HourlyWeather$DATE,"%Y-%m-%d")  
HourlyWeather$Hour <- format(HourlyWeather$DATE,"%H")  
  
HourlyWeather$Hour <-ifelse(HourlyWeather$Hour <= '04','04',  
 ifelse(HourlyWeather$Hour <= '08','08',  
 ifelse(HourlyWeather$Hour <= 12, 12,  
 ifelse(HourlyWeather$Hour <= 16, 16,  
 ifelse(HourlyWeather$Hour <= 20, 20,  
 ifelse(HourlyWeather$Hour <= 24, 24))))))  
HourlyWeather$TempTime<-paste(HourlyWeather$TempTime, HourlyWeather$Hour, sep=" ")  
#Recreate the Hour data  
SMUSwipe$Hour <- format(SMUSwipe$LDT,"%H")

Here we are merging on TempTime to pull in hourly weather data

#Merge on TempTime to pull in hourly weather data  
SMUSwipe = merge(SMUSwipe, HourlyWeather, by.x='TempTime', by.y='TempTime',all.x = TRUE, all.y = TRUE)  
#Remove dup after the left join  
SMUSwipe = distinct(SMUSwipe, LDT, .keep\_all = TRUE)  
SMUSwipe = SMUSwipe[which(!is.na(SMUSwipe$`LDT`)),]  
  
#sum(is.na(SMUSwipe$HourlyDryBulbTemperature))  
  
#Fill in NA's from Temperature data with number close to the mean  
SMUSwipe$HourlyDryBulbTemperature[is.na(SMUSwipe$HourlyDryBulbTemperature)] <- 65  
SMUSwipe$HourlyAltimeterSetting[is.na(SMUSwipe$HourlyAltimeterSetting)] <- 30  
SMUSwipe$HourlyDewPointTemperature[is.na(SMUSwipe$HourlyDewPointTemperature)] <- 49  
SMUSwipe$HourlyRelativeHumidity[is.na(SMUSwipe$HourlyRelativeHumidity)] <- 58  
SMUSwipe$HourlyWindSpeed[is.na(SMUSwipe$HourlyWindSpeed)] <- 0  
  
#Renaming some information for ease of use  
SMUSwipe$Temperature <- SMUSwipe$HourlyDryBulbTemperature  
SMUSwipe$Hour <- SMUSwipe$Hour.x  
  
#Dropping some data we no longer need  
drop <- c("Message Type","Hour.x","Hour.y", "TempTime", "DATE", "HourlyPressureChange", "HourlyPressureTendency", "HourlySeaLevelPressure", "HourlyWetBulbTemperature", "HourlyDryBulbTemperature")  
SMUSwipe = SMUSwipe[,!(names(SMUSwipe) %in% drop)]

Create and export datasets for time series studies

#Read in the data  
SMU = read.csv('SMUSwipe.csv',header = TRUE)  
#Look at the top of the data  
head(SMU)

## LDT ID. Secondary.Object.Name Time Day Date  
## 1 2019-01-02 07:14:08 1751 DEDM (101.2LB) 07:14:08 Wednesday 2019-01-02  
## 2 2019-01-02 07:20:53 6472 DEDM (101.2LB) 07:20:53 Wednesday 2019-01-02  
## 3 2019-01-02 07:11:36 4341 DEDM (101.3LB) 07:11:36 Wednesday 2019-01-02  
## 4 2019-01-02 06:16:37 247 DEDM (101.3LB) 06:16:37 Wednesday 2019-01-02  
## 5 2019-01-02 07:08:26 1752 DEDM (101.2LB) 07:08:26 Wednesday 2019-01-02  
## 6 2019-01-02 07:08:32 2515 DEDM (101.3LB) 07:08:32 Wednesday 2019-01-02  
## Hours Month HourlyAltimeterSetting HourlyDewPointTemperature  
## 1 2019-01-02 07 2019-01 30.23 30  
## 2 2019-01-02 07 2019-01 30.23 30  
## 3 2019-01-02 07 2019-01 30.23 30  
## 4 2019-01-02 06 2019-01 30.23 30  
## 5 2019-01-02 07 2019-01 30.23 30  
## 6 2019-01-02 07 2019-01 30.23 30  
## HourlyPrecipitation HourlyPresentWeatherType HourlyRelativeHumidity  
## 1 NA -RA:02 |RA |RA 93  
## 2 NA -RA:02 |RA |RA 93  
## 3 NA -RA:02 |RA |RA 93  
## 4 NA -RA:02 |RA |RA 93  
## 5 NA -RA:02 |RA |RA 93  
## 6 NA -RA:02 |RA |RA 93  
## HourlySkyConditions HourlyStationPressure HourlyVisibility  
## 1 OVC:08 7 29.54 4  
## 2 OVC:08 7 29.54 4  
## 3 OVC:08 7 29.54 4  
## 4 OVC:08 7 29.54 4  
## 5 OVC:08 7 29.54 4  
## 6 OVC:08 7 29.54 4  
## HourlyWindDirection HourlyWindGustSpeed HourlyWindSpeed Temperature Hour  
## 1 020 18 13 32 7  
## 2 020 18 13 32 7  
## 3 020 18 13 32 7  
## 4 020 18 13 32 6  
## 5 020 18 13 32 7  
## 6 020 18 13 32 7  
## Minutes interval\_15 Minutes\_15 WeekdayCount  
## 1 14 1 2019-01-02 07 1 2019-01-02 Wednesday  
## 2 20 2 2019-01-02 07 2 2019-01-02 Wednesday  
## 3 11 1 2019-01-02 07 1 2019-01-02 Wednesday  
## 4 16 2 2019-01-02 06 2 2019-01-02 Wednesday  
## 5 8 1 2019-01-02 07 1 2019-01-02 Wednesday  
## 6 8 1 2019-01-02 07 1 2019-01-02 Wednesday

#Dropping some data we no longer need  
drop <- c("ID.", "interval\_15", "Minutes\_15","Time")  
SMU = SMU[,!(names(SMU) %in% drop)]  
  
SMU$Week<-paste(SMU$Month, SMU$Day, sep=" ")  
  
#Renaming some information for ease of use  
#SMUSwipe$Temperature <- SMUSwipe$HourlyDryBulbTemperature  
#SMUSwipe$Hour <- SMUSwipe$Hour.x  
  
colnames(SMU)[2] = "Turnstile"  
  
SMU$Date <- as.Date(SMU$LDT)  
  
# Extract day of the week (Saturday = 6)  
SMU$Week\_Day <- as.numeric(format(format.Date(SMU$LDT,"%w")))  
  
# Adjust end-of-week date (first saturday from the original Date)  
SMU$End\_of\_Week <- SMU$Date + (6 - SMU$Week\_Day)  
  
HourSwipes = dplyr::count(SMU,Hours)  
DaySwipes = dplyr::count(SMU,Date)  
WeekSwipes = dplyr::count(SMU,End\_of\_Week)  
MonthSwipes = dplyr::count(SMU,Month)  
  
HourSwipesTemp = merge(HourSwipes,SMU, by='Hours')  
HourSwipesTemp = distinct(HourSwipesTemp, Hours, .keep\_all = TRUE)  
  
DaySwipesTemp = merge(DaySwipes,SMU, by='Date')  
DaySwipesTemp = distinct(DaySwipesTemp, Date, .keep\_all = TRUE)  
# Aggregate over Date and Temperature  
DaySwipesTemp$DayTemperature = aggregate(Temperature~Date, FUN=mean, data=SMU, na.rm=TRUE)  
DaySwipesTemp = DaySwipesTemp[,!(names(DaySwipesTemp) %in% 'Temperature')]  
  
WeekSwipesTemp = merge(WeekSwipes,SMU, by='End\_of\_Week')  
WeekSwipesTemp = distinct(DaySwipesTemp, End\_of\_Week, .keep\_all = TRUE)  
# Aggregate over week and Temperature  
WeekSwipesTemp$WeekTemperature = aggregate(Temperature~End\_of\_Week, FUN=mean, data=SMU, na.rm=TRUE)  
WeekSwipesTemp = WeekSwipesTemp[,!(names(WeekSwipesTemp) %in% 'Temperature')]  
  
MonthSwipesTemp = merge(MonthSwipes,SMU, by='Month')  
MonthSwipesTemp = distinct(MonthSwipesTemp, Month, .keep\_all = TRUE)  
# Aggregate over week and Temperature  
MonthSwipesTemp$MonthTemperature = aggregate(Temperature~Month, FUN=mean, data=SMU, na.rm=TRUE)  
MonthSwipesTemp = MonthSwipesTemp[,!(names(MonthSwipesTemp) %in% 'Temperature')]  
  
#Change n to IDSwipes  
colnames(HourSwipesTemp)[2] = "IDSwipes"  
colnames(DaySwipesTemp)[2] = "IDSwipes"  
colnames(WeekSwipesTemp)[2] = "IDSwipes"  
colnames(MonthSwipesTemp)[2] = "IDSwipes"  
  
#Dropping some data we no longer need  
drop <- c("LDT", "Minutes", "Week", "Turnstile")  
HourSwipesTemp = HourSwipesTemp[,!(names(HourSwipesTemp) %in% drop)]  
DaySwipesTemp = DaySwipesTemp[,!(names(DaySwipesTemp) %in% drop)]  
WeekSwipesTemp = WeekSwipesTemp[,!(names(WeekSwipesTemp) %in% drop)]  
MonthSwipesTemp = MonthSwipesTemp[,!(names(MonthSwipesTemp) %in% drop)]  
  
#Export the dataset  
write.csv(HourSwipesTemp,"DedmanHourlySwipe.csv", row.names = FALSE)  
write.csv(DaySwipesTemp,"DedmanDailySwipe.csv", row.names = FALSE)  
write.csv(WeekSwipesTemp,"DedmanWeeklySwipe.csv", row.names = FALSE)  
write.csv(MonthSwipesTemp,"DedmanMonthlySwipe.csv", row.names = FALSE)

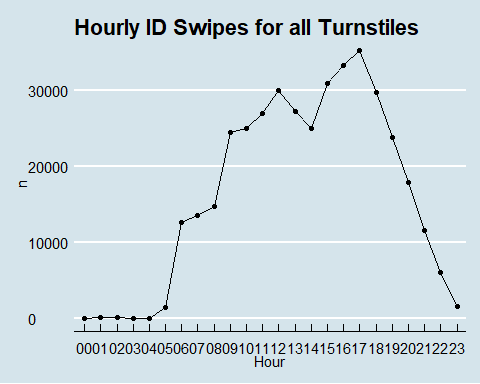
We now have a data set with weather data along with card swipes.

head(SMUSwipe)

## LDT ID# Secondary Object Name Time Day Date  
## 2 2019-01-02 07:14:08 1751 DEDM (101.2LB) 07:14:08 Wednesday 2019-01-02  
## 3 2019-01-02 07:20:53 6472 DEDM (101.2LB) 07:20:53 Wednesday 2019-01-02  
## 4 2019-01-02 07:11:36 4341 DEDM (101.3LB) 07:11:36 Wednesday 2019-01-02  
## 5 2019-01-02 06:16:37 247 DEDM (101.3LB) 06:16:37 Wednesday 2019-01-02  
## 6 2019-01-02 07:08:26 1752 DEDM (101.2LB) 07:08:26 Wednesday 2019-01-02  
## 7 2019-01-02 07:08:32 2515 DEDM (101.3LB) 07:08:32 Wednesday 2019-01-02  
## Hours Month HourlyAltimeterSetting HourlyDewPointTemperature  
## 2 2019-01-02 07 2019-01 30.23 30  
## 3 2019-01-02 07 2019-01 30.23 30  
## 4 2019-01-02 07 2019-01 30.23 30  
## 5 2019-01-02 06 2019-01 30.23 30  
## 6 2019-01-02 07 2019-01 30.23 30  
## 7 2019-01-02 07 2019-01 30.23 30  
## HourlyPrecipitation HourlyPresentWeatherType HourlyRelativeHumidity  
## 2 NA -RA:02 |RA |RA 93  
## 3 NA -RA:02 |RA |RA 93  
## 4 NA -RA:02 |RA |RA 93  
## 5 NA -RA:02 |RA |RA 93  
## 6 NA -RA:02 |RA |RA 93  
## 7 NA -RA:02 |RA |RA 93  
## HourlySkyConditions HourlyStationPressure HourlyVisibility  
## 2 OVC:08 7 29.54 4  
## 3 OVC:08 7 29.54 4  
## 4 OVC:08 7 29.54 4  
## 5 OVC:08 7 29.54 4  
## 6 OVC:08 7 29.54 4  
## 7 OVC:08 7 29.54 4  
## HourlyWindDirection HourlyWindGustSpeed HourlyWindSpeed Temperature Hour  
## 2 020 18 13 32 07  
## 3 020 18 13 32 07  
## 4 020 18 13 32 07  
## 5 020 18 13 32 06  
## 6 020 18 13 32 07  
## 7 020 18 13 32 07

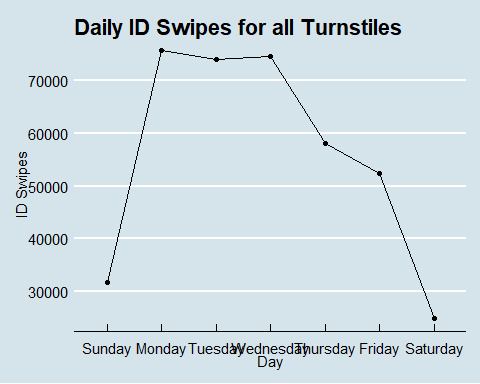
Hourly ID Swipes for all Turnstiles?

#Group all badge swipes by the hour  
  
hourplot = dplyr::count(SMUSwipe,Hour)  
  
hourplot %>%  
 ggplot(aes(x=Hour,y=n, group=1))+  
 geom\_line()+  
 geom\_point()+  
 theme\_economist()+  
 scale\_colour\_economist()+  
 ggtitle('Hourly ID Swipes for all Turnstiles')



What are the Daily ID Swipes for all Turnstiles?

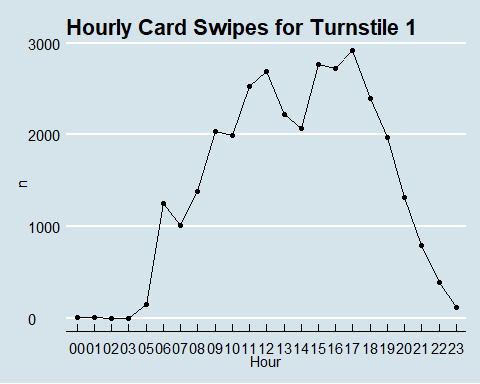
dayCount=dplyr::count(SMUSwipe, Day)  
dayCount$Day <- factor(dayCount$Day, levels= c("Sunday", "Monday",   
 "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))  
  
dayCount=dayCount[order(dayCount$Day), ]  
   
dayCount %>%  
 ggplot(aes(x=Day,y=n, group=1))+  
 geom\_line()+  
 geom\_point()+  
 theme\_economist()+  
 scale\_colour\_economist()+  
 theme\_economist()+  
 ggtitle('Daily ID Swipes for all Turnstiles')+  
 ylab('ID Swipes')



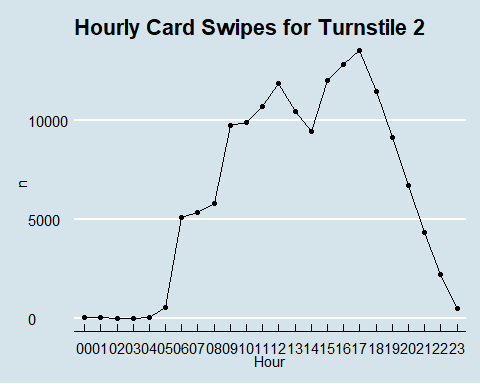
Look at turnstile information.  
Create individual turnstile data

#Create a dataset for each turnstyle  
SMUSwipeTurn1 = SMUSwipe[SMUSwipe[,3]=="DEDM (101.1LB)",]  
SMUSwipeTurn2 = SMUSwipe[SMUSwipe[,3]=="DEDM (101.2LB)",]  
SMUSwipeTurn3 = SMUSwipe[SMUSwipe[,3]=="DEDM (101.3LB)",]

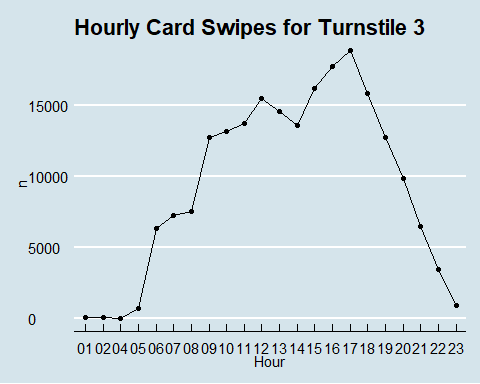
df = dplyr::count(SMUSwipeTurn3, Hours)  
  
HTurn1 = dplyr::count(SMUSwipeTurn1, Hour)  
HTurn1 %>%  
 ggplot(aes(x=Hour,y=n, group=1))+  
 geom\_line()+  
 geom\_point()+  
 theme\_economist()+  
 scale\_colour\_economist()+  
 ggtitle('Hourly Card Swipes for Turnstile 1')



HTurn2 = dplyr::count(SMUSwipeTurn2, Hour)  
HTurn2 %>%  
 ggplot(aes(x=Hour,y=n, group=1))+  
 geom\_line()+  
 geom\_point()+  
 theme\_economist()+  
 scale\_colour\_economist()+  
 ggtitle('Hourly Card Swipes for Turnstile 2')



HTurn3 = dplyr::count(SMUSwipeTurn3, Hour)  
HTurn3 %>%  
 ggplot(aes(x=Hour,y=n, group=1))+  
 geom\_line()+  
 geom\_point()+  
 theme\_economist()+  
 scale\_colour\_economist()+  
 ggtitle('Hourly Card Swipes for Turnstile 3')



Looks like turnstile 3 gets used more than the others but the pattern of usage looks the same across each.

This data is anamonized but we can still see who are some of the top users.

#This will count up the times users swiped in.   
IDCount = dplyr::count(SMUSwipe, `ID#`)  
#Print off the top 10 users ordering nuency in decending order  
head(IDCount[order(IDCount$n, decreasing = TRUE),],10)

## # A tibble: 10 x 2  
## `ID#` n  
## <dbl> <int>  
## 1 1445 429  
## 2 1168 413  
## 3 1429 390  
## 4 1469 369  
## 5 1140 361  
## 6 912 354  
## 7 1778 349  
## 8 104 339  
## 9 152 338  
## 10 214 335

Look at some user data.

#Breakdown user numbers  
summary(IDCount$n)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 7.00 21.00 35.99 48.00 429.00

#how many users  
dplyr::count(IDCount)

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 10873