Econ294A R programming final

Feng Hong

March 17, 2016

Extract Data from Database

```
flights_sqlite <- tbl(nycflights13_sqlite(), "flights")
airlines_sqlite <- tbl(nycflights13_sqlite(), "airlines")
airports_sqlite <- tbl(nycflights13_sqlite(), "airports")
planes_sqlite <- tbl(nycflights13_sqlite(), "planes")
weather_sqlite <- tbl(nycflights13_sqlite(), "weather")
```

Join the table *flights* and *planes*

```
inner_flights_planes <- inner_join(flights, planes, by = "tailnum") %>% tbl_df
names(inner_flights_planes)
```

```
##  [1] "year.x"       "month"        "day"          "dep_time"
##  [5] "dep_delay"    "arr_time"     "arr_delay"    "carrier"
##  [9] "tailnum"      "flight"       "origin"       "dest"
## [13] "air_time"     "distance"     "hour"         "minute"
## [17] "year.y"       "type"         "manufacturer" "model"
## [21] "engines"      "seats"        "speed"        "engine"
```

```
colnames(inner_flights_planes)[1] <- "flight_year"
colnames(inner_flights_planes)[15] <- "dep_hour"
colnames(inner_flights_planes)[17] <- "plane_year"
```

Create the date index

```
flights_planes <- inner_flights_planes %>%
  mutate(
    date = paste(flight_year, month, day, sep = "-"),
    date = as.Date(date, format = "%Y-%m-%d"), # create date to merge with weather
    cancelled = ifelse(is.na(arr_time), 1, 0)  # question requires this
  )
```

Change character variable to factor variable

```
flights_planes$carrier <- as.factor(flights_planes$carrier)
flights_planes$origin <- as.factor(flights_planes$origin)
flights_planes$dest <- as.factor(flights_planes$dest)
flights_planes$manufacturer <- as.factor(flights_planes$manufacturer)
```

Change integer variable to factor variable

```
flights_planes$month <- as.factor(flights_planes$month)
flights_planes$flight <- as.factor(flights_planes$flight)
```

Weather Data Refinement

```
weather <- weather_sqlite %>%
  collect() %>%
  mutate(
    date = paste(year, month, day, sep = "-"),
    date = as.Date(date, format = "%Y-%m-%d"),
    weekday = weekdays(date),
    weekday = as.factor(weekday) # add the weekday variable
  )
```

```
weather_mean <- weather %>% group_by(date) %>%
  summarise(
    weekday = first(weekday),
    mean_temp = mean(temp),
    mean_dewp = mean(dewp),
    mean_humid = mean(humid),
    # mean_wind_dir = mean(wind_dir), wind direction has too many NA's.
    mean_wind_speed = mean(wind_speed),
    mean_wind_gust = mean(wind_gust),
    mean_precip = mean(precip),
    # mean_pressure = mean(pressure), pressure has too many NA's.
    mean_visib = mean(visib)
  )
```

Identify the highly correlated data

```
corr_weather_mean <- cor(na.omit(weather_mean[,3:9]))
print(corr_weather_mean)
```

```
##                  mean_temp  mean_dewp mean_humid mean_wind_speed
## mean_temp        1.00000000  0.9472889  0.2438439    -0.229014227
## mean_dewp        0.94728890  1.0000000  0.5379544    -0.287368426
## mean_humid       0.24384393  0.5379544  1.0000000    -0.271040655
## mean_wind_speed -0.22901423 -0.2873684 -0.2710407     1.000000000
## mean_wind_gust  -0.22901423 -0.2873684 -0.2710407     1.000000000
## mean_precip     -0.01126335  0.1145454  0.4074626     0.005096705
## mean_visib       0.11205170 -0.1038539 -0.6729998     0.142012649
##                 mean_wind_gust  mean_precip mean_visib
## mean_temp         -0.229014227 -0.011263354  0.1120517
## mean_dewp         -0.287368426  0.114545363 -0.1038539
## mean_humid        -0.271040655  0.407462572 -0.6729998
## mean_wind_speed    1.000000000  0.005096705  0.1420126
```

```
## mean_wind_gust     1.000000000  0.005096705  0.1420126
## mean_precip        0.005096705  1.000000000 -0.4623199
## mean_visib         0.142012649 -0.462319935  1.0000000
```

Delete columns that are highly correlated from weather_mean

```
weather2 <- weather_mean %>%
  dplyr::select(date, weekday, mean_temp, mean_wind_speed, mean_precip, mean_visib)
corr_weather2<- cor(na.omit(weather2[,3:6]))
print(corr_weather2)
```

```
##                  mean_temp mean_wind_speed  mean_precip mean_visib
## mean_temp        1.00000000    -0.229014227 -0.011263354  0.1120517
## mean_wind_speed -0.22901423     1.000000000  0.005096705  0.1420126
## mean_precip     -0.01126335     0.005096705  1.000000000 -0.4623199
## mean_visib       0.11205170     0.142012649 -0.462319935  1.0000000
```

Join flights, planes and weather data and obtain the final data

```
final_data <- inner_join(flights_planes, weather2, by = "date")
length(final_data[final_data$cancelled == 1])
```

```
## [1] 4547
```

```
names(final_data)
```

```
##  [1] "flight_year"     "month"           "day"
##  [4] "dep_time"        "dep_delay"       "arr_time"
##  [7] "arr_delay"       "carrier"         "tailnum"
## [10] "flight"          "origin"          "dest"
## [13] "air_time"        "distance"        "dep_hour"
## [16] "minute"          "plane_year"      "type"
## [19] "manufacturer"    "model"           "engines"
## [22] "seats"           "speed"           "engine"
## [25] "date"            "cancelled"       "weekday"
## [28] "mean_temp"       "mean_wind_speed" "mean_precip"
## [31] "mean_visib"
```

OLS for *dep_delay*

I expect the following regressors to be significant. Some similar variables such as mean_temp, mean_wind_speed are not included as arguments mostly because their impact on departure delay can be ambiguous.

```
model.delay <- lm(dep_delay ~ month + as.factor(weekday) + carrier + origin +
                plane_year + seats + mean_precip + mean_visib,
              data = final_data)
summary(model.delay)
```

```
##
## Call:
## lm(formula = dep_delay ~ month + as.factor(weekday) + carrier +
##     origin + plane_year + seats + mean_precip + mean_visib, data = final_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -77.67  -18.12  -10.08    0.48 1302.36
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  205.755670  34.704574   5.929 3.06e-09 ***
## month2                         0.916951   0.389117   2.356 0.018449 *
## month3                         6.049689   0.374667  16.147  < 2e-16 ***
## month4                         6.838469   0.376252  18.175  < 2e-16 ***
## month5                         3.312550   0.371874   8.908  < 2e-16 ***
## month6                        12.417621   0.380368  32.646  < 2e-16 ***
## month7                        15.555095   0.374606  41.524  < 2e-16 ***
## month8                         5.528696   0.374189  14.775  < 2e-16 ***
## month9                         0.026098   0.379965   0.069 0.945241
## month10                       -0.694563   0.372429  -1.865 0.062189 .
## month11                       -1.791855   0.378258  -4.737 2.17e-06 ***
## month12                        6.462775   0.378077  17.094  < 2e-16 ***
## as.factor(weekday)Monday      -0.854723   0.277070  -3.085 0.002037 **
## as.factor(weekday)Saturday    -6.991724   0.296295 -23.597  < 2e-16 ***
## as.factor(weekday)Sunday      -3.292526   0.281827 -11.683  < 2e-16 ***
## as.factor(weekday)Thursday     2.329743   0.277379   8.399  < 2e-16 ***
## as.factor(weekday)Tuesday     -4.330445   0.277813 -15.588  < 2e-16 ***
## as.factor(weekday)Wednesday   -3.689374   0.277307 -13.304  < 2e-16 ***
## carrierAA                     -9.560631   0.619017 -15.445  < 2e-16 ***
## carrierAS                    -12.181077   1.542126  -7.899 2.82e-15 ***
## carrierB6                     -4.620915   0.355492 -12.999  < 2e-16 ***
## carrierDL                     -9.326508   0.413355 -22.563  < 2e-16 ***
## carrierEV                      2.189111   0.406559   5.384 7.27e-08 ***
## carrierF9                      2.457422   1.618348   1.518 0.128896
## carrierFL                      0.858047   0.818992   1.048 0.294784
## carrierHA                    -13.771479   2.308885  -5.965 2.46e-09 ***
## carrierMQ                    -11.363331   1.416456  -8.022 1.04e-15 ***
## carrierOO                     -0.555190   7.347354  -0.076 0.939767
## carrierUA                     -6.782975   0.430371 -15.761  < 2e-16 ***
## carrierUS                    -14.324738   0.463010 -30.938  < 2e-16 ***
## carrierVX                     -4.874944   0.652174  -7.475 7.75e-14 ***
## carrierWN                     -0.225239   0.512890  -0.439 0.660549
## carrierYV                      1.526848   1.741014   0.877 0.380494
## originJFK                     -0.682926   0.253037  -2.699 0.006957 **
## originLGA                     -0.782260   0.232691  -3.362 0.000774 ***
## plane_year                    -0.081739   0.017291  -4.727 2.28e-06 ***
## seats                          0.006086   0.001413   4.306 1.66e-05 ***
## mean_precip                  246.102409   9.817433  25.068  < 2e-16 ***
## mean_visib                    -2.973681   0.061364 -48.460  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.51 on 274122 degrees of freedom
```

```
##   (9358 observations deleted due to missingness)
## Multiple R-squared:  0.05255,    Adjusted R-squared:  0.05242
## F-statistic: 400.1 on 38 and 274122 DF,  p-value: < 2.2e-16
```

From the regression we can see that dummy variables "month" and "carrier" are generally significant regressor. "Weekday, "origin, "plane_year", "seats", "mean_precip", "mean_visib" are all significant. The coefficients stated in the regression results.
However, the model has an Adjusted R-squared as small as 0.05242. So based on this model, we might not be able to predict the departure delay precisely.

GLS for *cancel*

For regression of cancellation, use logit model because "cancelled" is a binary variable.

```
model.cancel <- glm(cancelled ~ month + as.factor(weekday) + carrier + origin +
                      seats +  mean_visib,
                    data = final_data, family=binomial(link="logit"))
summary(model.cancel)
```

```
##
## Call:
## glm(formula = cancelled ~ month + as.factor(weekday) + carrier +
##     origin + seats + mean_visib, family = binomial(link = "logit"),
##     data = final_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3304  -0.1698  -0.0952  -0.0522   4.2731
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -2.4393036  0.1562902 -15.608  < 2e-16 ***
## month2                        1.2048077  0.0746512  16.139  < 2e-16 ***
## month3                        0.8599651  0.0789870  10.887  < 2e-16 ***
## month4                        0.6527097  0.0856683   7.619 2.56e-14 ***
## month5                        0.2789271  0.0836728   3.334 0.000857 ***
## month6                        1.0834795  0.0784206  13.816  < 2e-16 ***
## month7                        1.2298009  0.0794558  15.478  < 2e-16 ***
## month8                        0.3539360  0.0915396   3.866 0.000110 ***
## month9                        0.3244573  0.0946035   3.430 0.000604 ***
## month10                      -0.5761280  0.1134244  -5.079 3.79e-07 ***
## month11                      -0.3623568  0.1110580  -3.263 0.001103 **
## month12                       0.6810983  0.0768904   8.858  < 2e-16 ***
## as.factor(weekday)Monday     -0.4322734  0.0538827  -8.022 1.04e-15 ***
## as.factor(weekday)Saturday   -0.3774389  0.0620898  -6.079 1.21e-09 ***
## as.factor(weekday)Sunday     -0.6874177  0.0631751 -10.881  < 2e-16 ***
## as.factor(weekday)Thursday    0.2302250  0.0504070   4.567 4.94e-06 ***
## as.factor(weekday)Tuesday    -0.3688324  0.0556339  -6.630 3.37e-11 ***
## as.factor(weekday)Wednesday -0.3249462  0.0543958  -5.974 2.32e-09 ***
## carrierAA                     1.6964487  0.1450800  11.693  < 2e-16 ***
```

```
## carrierAS                   -0.3015237  0.7226137  -0.417 0.676482
## carrierB6                    0.8911258  0.1290850   6.903 5.08e-12 ***
## carrierDL                    0.5303609  0.1377119   3.851 0.000118 ***
## carrierEV                    2.4170370  0.1290509  18.729  < 2e-16 ***
## carrierF9                   -1.2785345  1.0108560  -1.265 0.205942
## carrierFL                    1.2479201  0.1762006   7.082 1.42e-12 ***
## carrierHA                   -9.2171723 75.4569783  -0.122 0.902779
## carrierMQ                    1.9384151  0.1901058  10.197  < 2e-16 ***
## carrierOO                    3.1923504  0.6351356   5.026 5.00e-07 ***
## carrierUA                   -1.3228198  0.1851520  -7.145 9.03e-13 ***
## carrierUS                   -1.1994161  0.2207254  -5.434 5.51e-08 ***
## carrierVX                    0.7939935  0.2123517   3.739 0.000185 ***
## carrierWN                    1.0108355  0.1510457   6.692 2.20e-11 ***
## carrierYV                    2.7750120  0.1930360  14.376  < 2e-16 ***
## originJFK                   -0.2453609  0.0633199  -3.875 0.000107 ***
## originLGA                    0.3986579  0.0430716   9.256  < 2e-16 ***
## seats                       -0.0024326  0.0004009  -6.068 1.30e-09 ***
## mean_visib                  -0.3539327  0.0079566 -44.483  < 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: 46604  on 283518  degrees of freedom
## Residual deviance: 37218  on 283482  degrees of freedom
## AIC: 37292
##
## Number of Fisher Scoring iterations: 14
```

I've tried several other regressors, which all turned out to be insignificant.
In this particular model above, the factors chosen are almost all significant,
the coefficients stated in the regression results.
Using this model, we can predict the probability of cancellation based on the
independent variabls.

I'm also interested in how different weather conditions would affect the daily
mean cancellation. So I find daily total cancellation and keep weather variables
to run the regression.

```
total_cancel_by_date <- final_data %>% group_by(date) %>%
  summarise(cancel_total = sum(cancelled),
            mean_temp = mean(mean_temp, na.rm = T),
            mean_wind_speed = mean(mean_wind_speed, na.rm = T),
            mean_precip = mean(mean_precip, na.rm = T),
            mean_visib = mean(mean_visib, na.rm = T))

model.total_cancel <- lm(cancel_total ~ mean_temp + mean_wind_speed +
                           mean_precip + mean_visib,
                         data = total_cancel_by_date)
summary(model.total_cancel)
```

```
##
## Call:
```

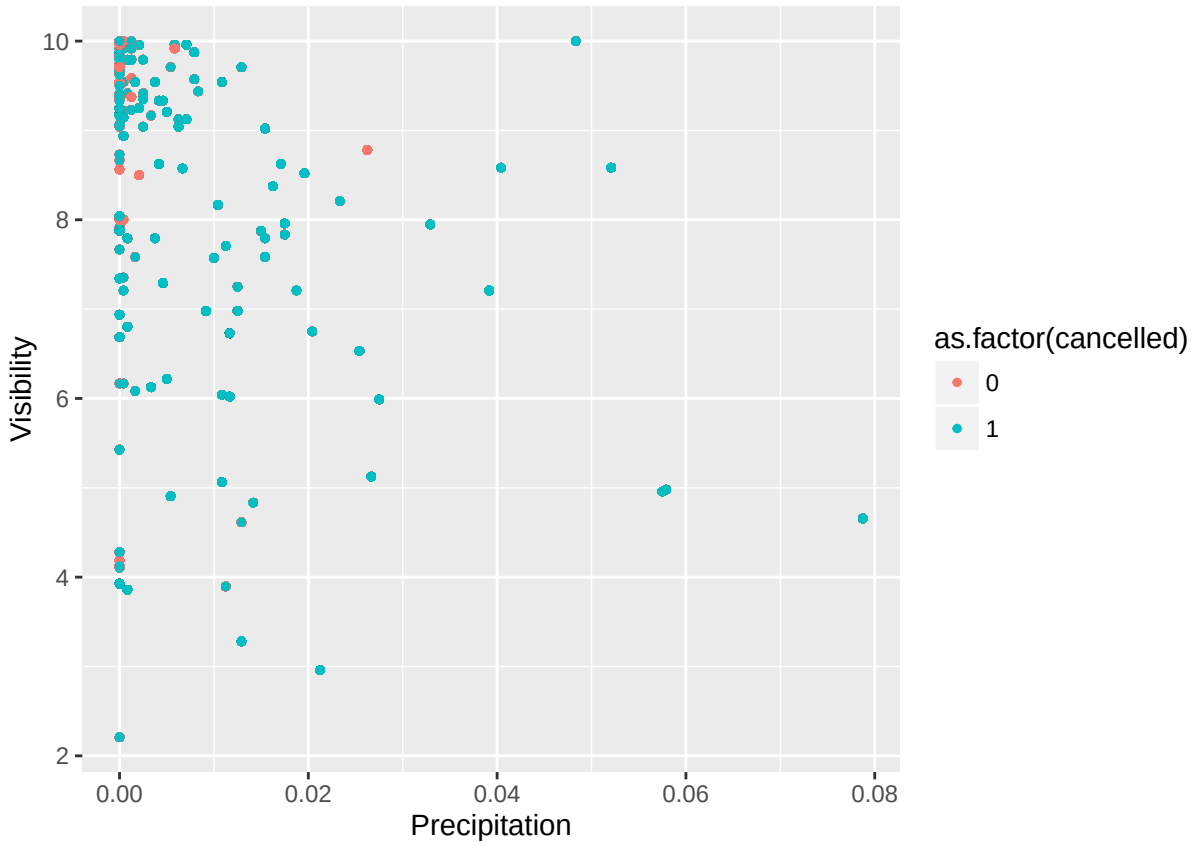
```
## lm(formula = cancel_total ~ mean_temp + mean_wind_speed + mean_precip +
##     mean_visib, data = total_cancel_by_date)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.385  -8.037  -3.902   1.052 225.323
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      57.93786    9.37411   6.181 1.74e-09 ***
## mean_temp         0.12050    0.07189   1.676  0.09458 .
## mean_wind_speed   0.87577    0.29079   3.012  0.00278 **
## mean_precip     120.97813  153.84485   0.786  0.43218
## mean_visib       -6.57796    0.93831  -7.010 1.19e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.04 on 357 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.1705, Adjusted R-squared:  0.1612
## F-statistic: 18.35 on 4 and 357 DF,  p-value: 1.011e-13
```

The independent variable mean_wind_speed and mean_visib are both significant. Using this model we can predict how many cancellations might occur based on weather conditions. But again, adjusted r-squared is not high enough to justify the model.

Plotting with ggplot2

(a) weather

```
plot_weather <- ggplot(data = final_data, aes(mean_precip, mean_visib))
plot_weather + geom_point( aes(color = as.factor(cancelled)),size = 1) +
  xlab("Precipitation") + ylab("Visibility")
```

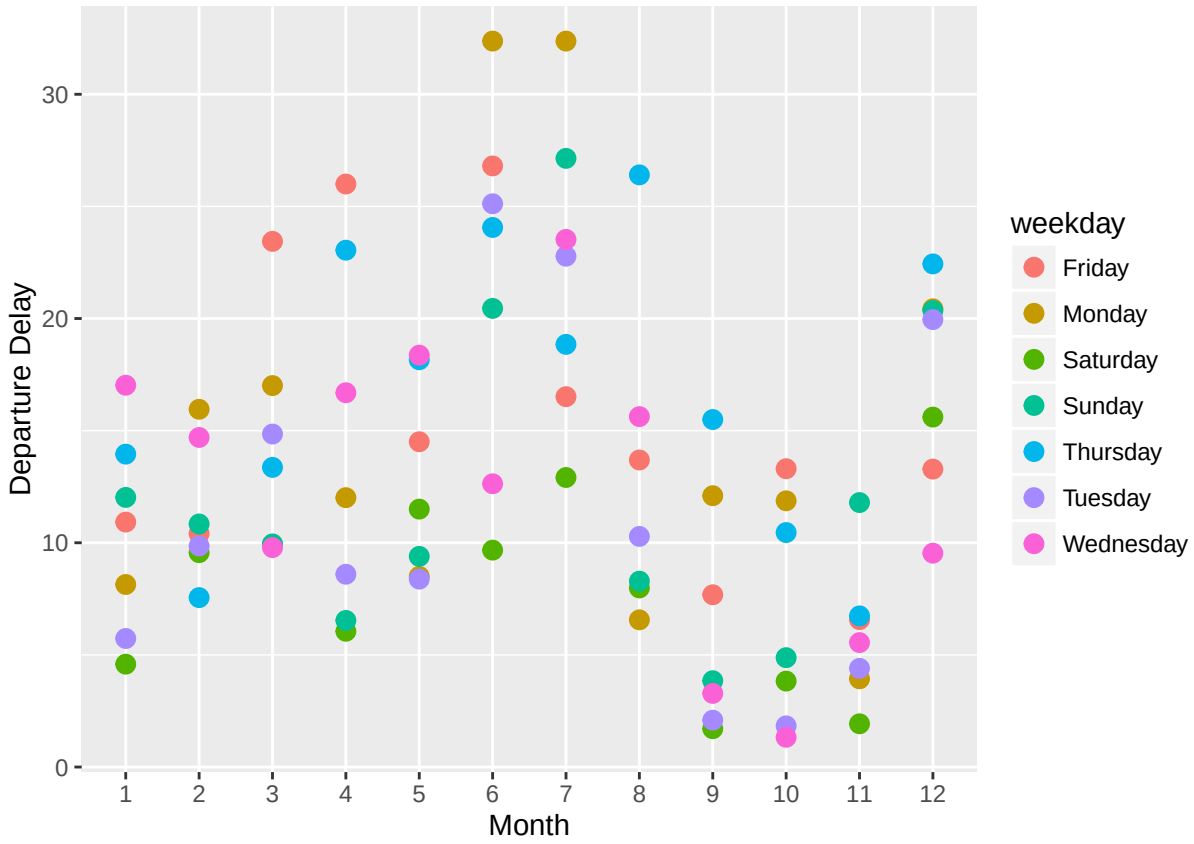


The plot shows a simple relationship between cancellation and 2 major weather conditions: visibility and precipitation. As we can see for most sunny clear days, cancellation relatively rarely happens.

(b) day of week and time of year

```
month_weekday <- final_data %>%
  group_by(month, weekday) %>%
  summarise(mean.dep_delay = mean(dep_delay, na.rm = T))

plot_time <- ggplot(month_weekday, aes(x = month, y= mean.dep_delay))
plot_time + geom_point(aes(color = weekday), size = 3) +
  xlab("Month") + ylab("Departure Delay")
```

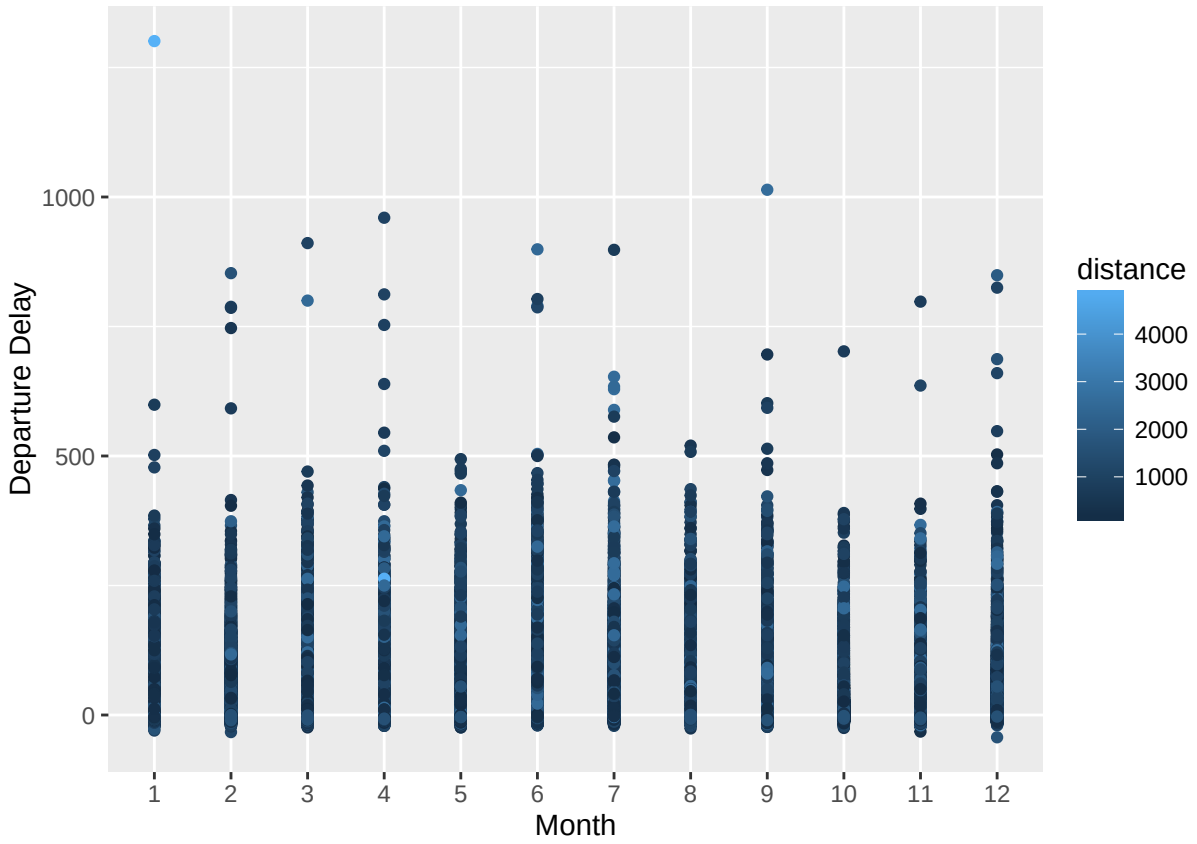


This picture gives us an idea of the pattern of departure delay with respect to weekday for each month. Weekend flights seem to have less departure delay.

(c) airport destination

```
plot_destination <- ggplot(data = final_data, aes(month, dep_delay, color = distance))
plot_destination + geom_point() +
  xlab("Month") + ylab("Departure Delay")
```

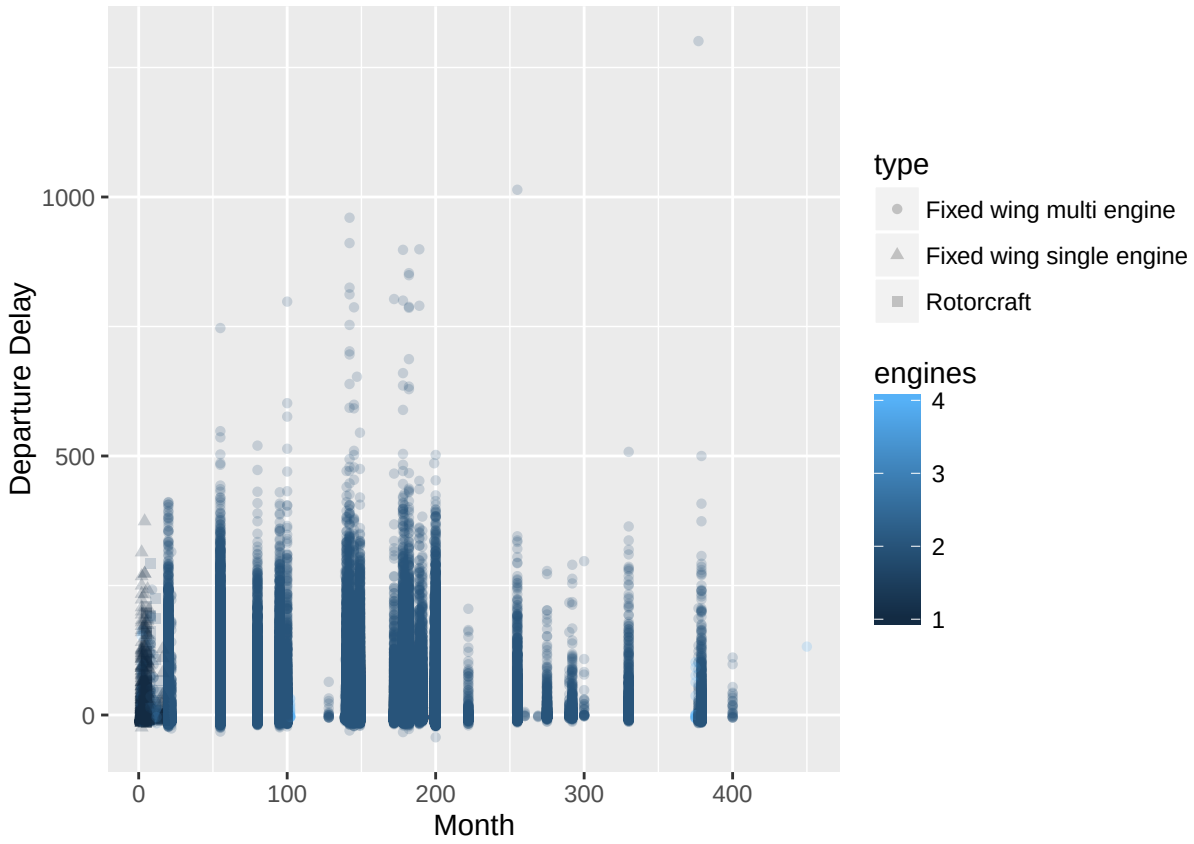
```
## Warning: Removed 4195 rows containing missing values (geom_point).
```



(d) characteristics of the plane

```
plot_plane <- ggplot(data = final_data, aes(x = seats, y = dep_delay,
                                            color = engines, shape = type))
plot_plane + geom_point(alpha=0.2) +
  xlab("Month") + ylab("Departure Delay")
```

```
## Warning: Removed 4195 rows containing missing values (geom_point).
```



The last picture here shows the relationship between engine number, plane type and departure delay for each month.

Try some data mining - Based on KNN

Got the error message:
Error in knn(train = knn_train, test = knn_test, cl = knn_train_target, :
'train' and 'class' have different lengths

Have to give up due to limit of time
KNN reference: https://www.youtube.com/watch?v=DkLNb0CXw84

```
table(final_data$cancelled)
set.seed(9850)
gp <- runif(nrow(final_data))

final_data_knn <- final_data[order(gp),]

# DEBUG:
# str(final_data_knn)
# summary(final_data_knn)

knn_no_na <- final_data_knn[(!is.na(final_data_knn$mean_temp)) &
                    (!is.na(final_data_knn$mean_wind_speed)) &
                    (!is.na(final_data_knn$mean_precip)) &
```

```
                    (!is.na(final_data_knn$mean_visib)),]

normalize <- function(x){ return( (x- min(x))/(max(x) - min(x)) ) }
knn_normal <- as.data.frame(lapply(knn_no_na[, c(28,29,30,31)], normalize))
summary(knn_normal)


knn_train <- knn_normal[1:250000,]
knn_test <- knn_normal[250001:281860,]

# DEBUG:
# names(knn_no_na)
knn_train_target <- knn_no_na[1:250000, 26]
knn_test_target <-knn_no_na[250001:281860, 26]

k = 500
m1 <- knn(train = knn_train, test = knn_test, cl=knn_train_target, k =500)
```