PP_Final_Holman

Chris Holman

2023-01-22

Importing libraries

```
library(nycflights13)
library(dplyr)
library(ggplot2)
library(car)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Getting data

```
data(flights)
data(weather)
```

EDA

Checking for na in weather

```
# number of na per column
cbind(lapply(lapply(weather, is.na), sum))
```

```
##            [,1]
## origin     0
## year       0
## month      0
## day        0
## hour       0
## temp       1
## dewp       1
## humid      1
## wind_dir   460
## wind_speed 4
## wind_gust  20778
## precip     0
## pressure   2729
```

```
## visib      0
## time_hour  0
```

```r
# removing wind gust (too many NA and just a confusing var)
if ("wind_gust" %in% colnames(weather)) {
    # cleaned weather
    w = subset(weather, select = -c(wind_gust))
}
```

We decided to remove the wind gust variable because it had a large amount of NA values. About three quarters of the data do not have values for this predictor. Also, from reading the data dictionary it was hard to understand what the wind gust variable meant.

Cleaning flights data

```r
cbind(lapply(lapply(flights, is.na), sum))
```

```
##                [,1]
## year           0
## month          0
## day            0
## dep_time       8255
## sched_dep_time 0
## dep_delay      8255
## arr_time       8713
## sched_arr_time 0
## arr_delay      9430
## carrier        0
## flight         0
## tailnum        2512
## origin         0
## dest           0
## air_time       9430
## distance       0
## hour           0
## minute         0
## time_hour      0
```

```r
# cleaned flights
f = flights[!is.na(flights$air_time), ]

f$sigDelay = f$dep_delay > 15
```

Within the flights data set, there is another issue of NA values with the air time variable having the most. We interpreted NA for air time as meaning the flights was cancelled. Since cancellation is different from departure delays, we removed all observations with out an air time. This gets rid of the rest of NA values for the whole flights data set.

We also created a variable named "sigDelay" which is a binary variable that encodes whether of not a flight was delayed according to the FAA's definition of a 15 minute threshold for if a flight is delayed.

Checking for multicollinearity

Weather Correlation

```
# getting only numeric columns and removing na's
w_numeric <- w[, sapply(w, is.numeric)]
w_numeric <- na.omit(w_numeric[, c(5:12)])

cor(w_numeric)
```

```
##                   temp        dewp      humid   wind_dir  wind_speed
## temp        1.00000000  0.90162405  0.1071538 -0.1290057 -0.10894880
## dewp        0.90162405  1.00000000  0.5193125 -0.2460956 -0.18136793
## humid       0.10715384  0.51931252  1.0000000 -0.3249786 -0.20582870
## wind_dir   -0.12900567 -0.24609556 -0.3249786  1.0000000  0.25445501
## wind_speed -0.10894880 -0.18136793 -0.2058287  0.2544550  1.00000000
## precip     -0.02950825  0.04208537  0.1865167 -0.0780323  0.03005549
## pressure   -0.25366597 -0.28858075 -0.1803573 -0.1988064 -0.13249274
## visib       0.04323954 -0.12773124 -0.4523975  0.1873828  0.04883138
##                 precip   pressure       visib
## temp       -0.02950825 -0.2536660  0.04323954
## dewp        0.04208537 -0.2885808 -0.12773124
## humid       0.18651675 -0.1803573 -0.45239754
## wind_dir   -0.07803230 -0.1988064  0.18738277
## wind_speed  0.03005549 -0.1324927  0.04883138
## precip      1.00000000 -0.1079932 -0.34709198
## pressure   -0.10799318  1.0000000  0.12277393
## visib      -0.34709198  0.1227739  1.00000000
```

Before we look for trends with delays, its important to get a feel for the data and understand how variables related to each other. There seems to be some high correlations within the weather data set which could lead to come multicollinearity issues. This means that multiple variables encode similar information. When modeling, this can reduce the model's effectiveness. The variables with the highest correlations are temperature, dew point, and humidity.

Flights Correlation

```
# getting only numeric columns and removing na's
f_numeric <- f[, sapply(f, is.numeric)]
f_numeric <- na.omit(f_numeric[, c(4:12)])

cor(f_numeric)
```

```
##                  dep_time sched_dep_time  dep_delay   arr_time sched_arr_time
## dep_time       1.00000000     0.95482687 0.25961272 0.66250900     0.78444199
## sched_dep_time 0.95482687     1.00000000 0.19892350 0.64438677     0.78058744
## dep_delay      0.25961272     0.19892350 1.00000000 0.02942101     0.16049724
## arr_time       0.66250900     0.64438677 0.02942101 1.00000000     0.79078877
## sched_arr_time 0.78444199     0.78058744 0.16049724 0.79078877     1.00000000
## arr_delay      0.23230573     0.17389620 0.91480276 0.02448214     0.13326129
```

```
## flight          0.04153017     0.02840127  0.05396975 0.02500740     0.01394723
## air_time       -0.01461948    -0.01553213 -0.02240508 0.05429603     0.07891830
## distance       -0.01413373    -0.01293250 -0.02168090 0.04718917     0.07361354
##                  arr_delay      flight    air_time    distance
## dep_time        0.23230573  0.04153017 -0.01461948 -0.01413373
## sched_dep_time  0.17389620  0.02840127 -0.01553213 -0.01293250
## dep_delay       0.91480276  0.05396975 -0.02240508 -0.02168090
## arr_time        0.02448214  0.02500740  0.05429603  0.04718917
## sched_arr_time  0.13326129  0.01394723  0.07891830  0.07361354
## arr_delay       1.00000000  0.07286208 -0.03529709 -0.06186776
## flight          0.07286208  1.00000000 -0.47283836 -0.48146018
## air_time       -0.03529709 -0.47283836  1.00000000  0.99064965
## distance       -0.06186776 -0.48146018  0.99064965  1.00000000
```

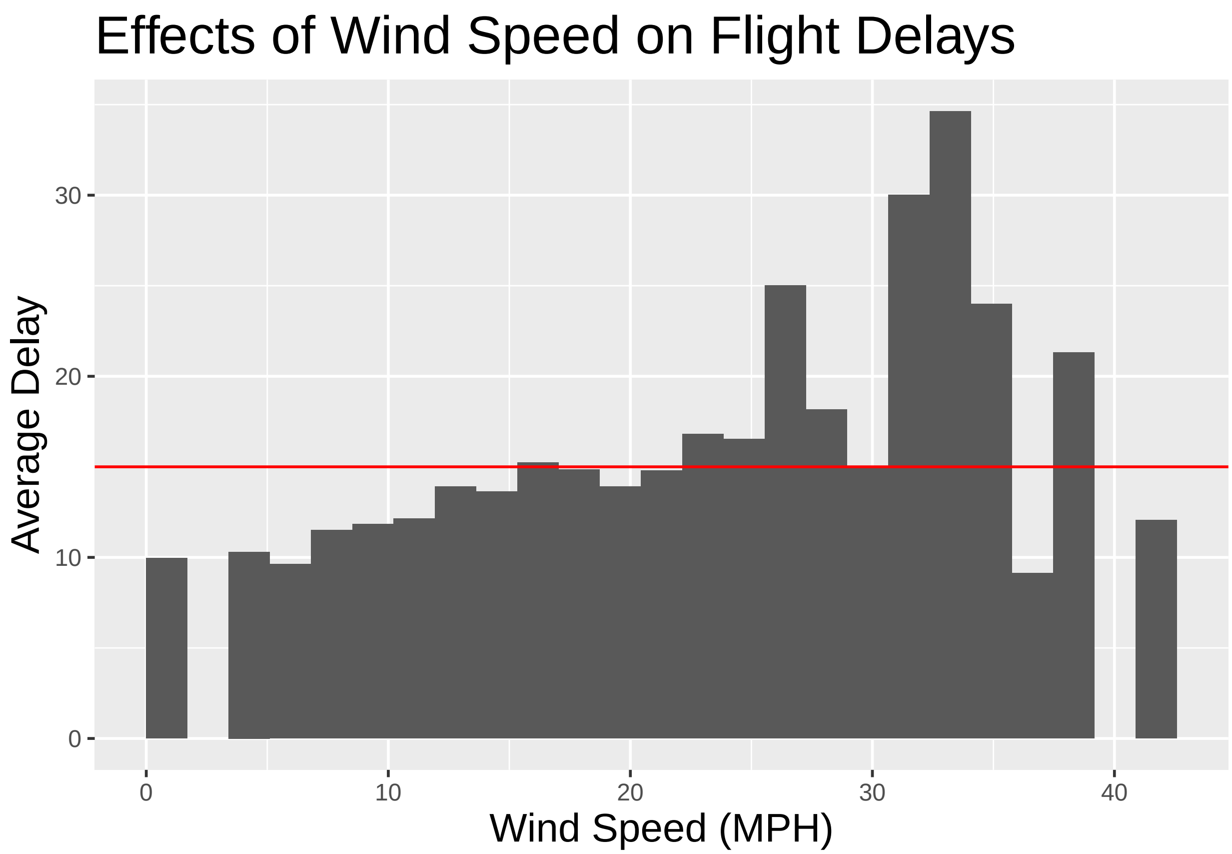
The flights data set looks to have more cases of multicollinearity. This makes sense when looking at the variables and what the represent. Since most of the variables are related to departure and arrival times, the scheduled and actual times for each particular flight are similar. Also, variables like departure delay and arrival delay are derived from the difference between the actual and scheduled times. This means that only a few of these variables will prove to be useful when we begin modeling. Finally, variables like arrival delay or arrival time should not be used because in practice those will not be known before a flight takes off from New York.

Looking for trends with Departure Delay

```r
# merging together cleaned weather and flights data
wf <- merge(f, w, by = c("time_hour", "origin"))
```

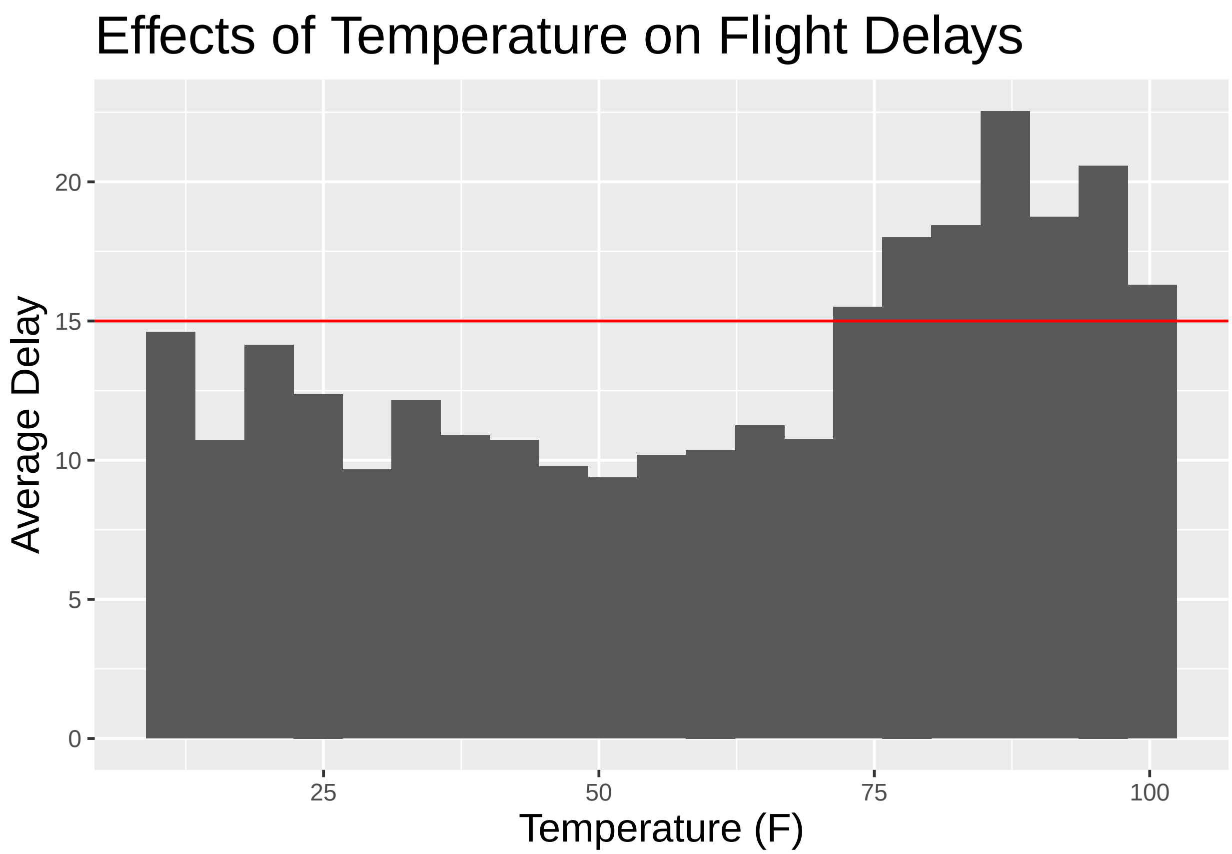
```r
ggplot(wf, aes(x = wind_speed, y = dep_delay)) + stat_summary_bin(fun = "mean",
    geom = "bar", bins = 25) + ylab("Average Delay") + xlab("Wind Speed (MPH)") +
    ggtitle("Effects of Wind Speed on Flight Delays ") + geom_hline(yintercept = 15,
    color = "red") + theme(axis.title = element_text(size = 15),
    plot.title = element_text(size = 20))
```

```
## Warning: Removed 78 rows containing non-finite values (‘stat_summary_bin()‘).
```



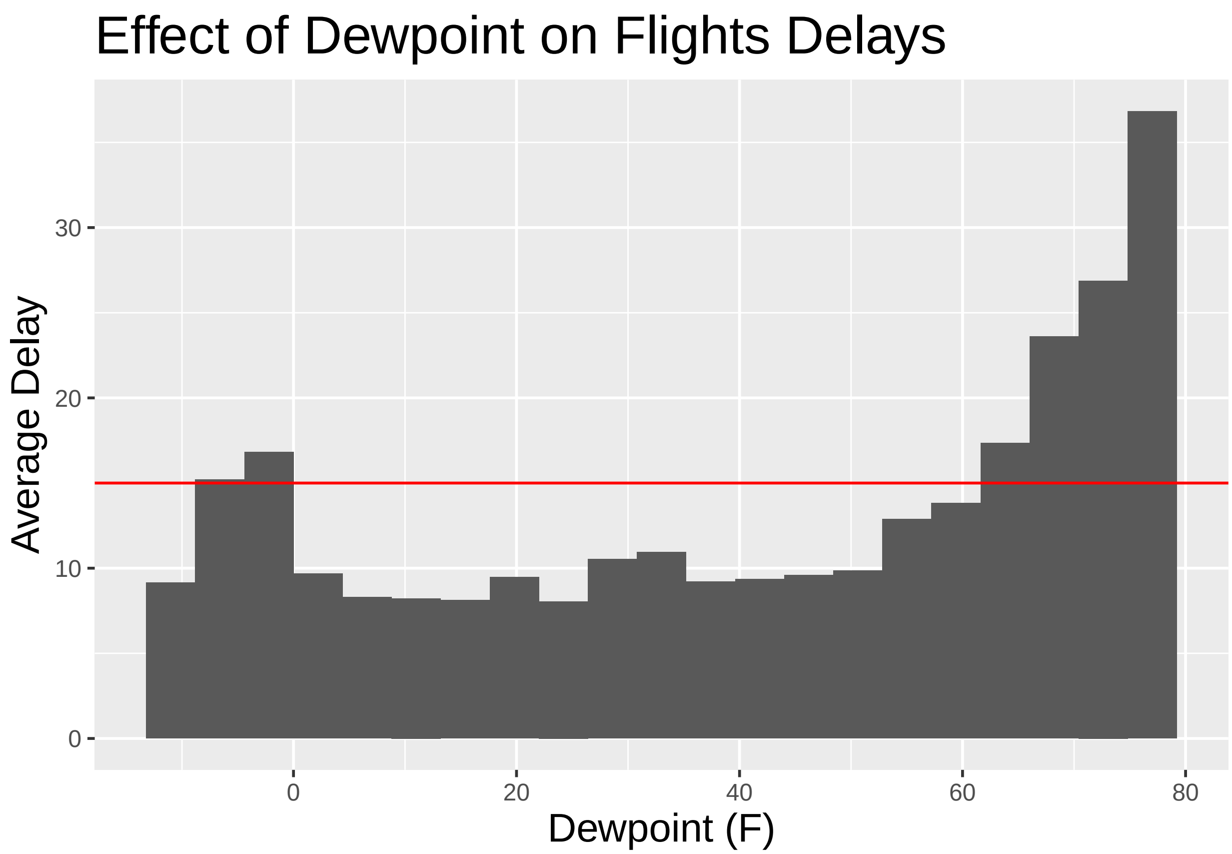

```
ggplot(wf, aes(x = temp, y = dep_delay)) + stat_summary_bin(fun = "mean",
    geom = "bar", bins = 20) + ylab("Average Delay") + xlab("Temperature (F)") +
    ggtitle("Effects of Temperature on Flight Delays ") + geom_hline(yintercept = 15,
    color = "red") + theme(axis.title = element_text(size = 15),
    plot.title = element_text(size = 20))
```

```
## Warning: Removed 17 rows containing non-finite values (`stat_summary_bin()`).
```



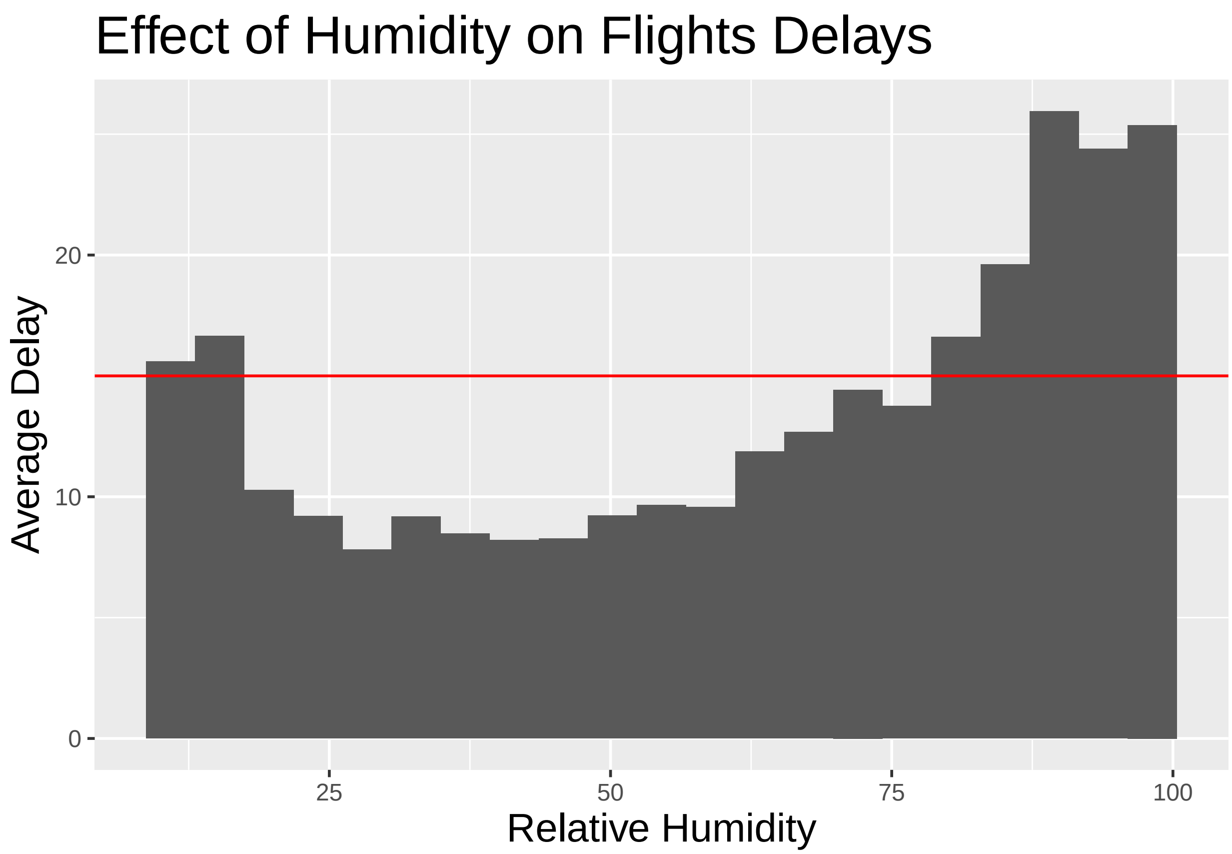

```r
ggplot(wf, aes(x = dewp, y = dep_delay)) + stat_summary_bin(fun = "mean",
    geom = "bar", bins = 20) + ylab("Average Delay") + xlab("Dewpoint (F)") +
    ggtitle("Effect of Dewpoint on Flights Delays") + geom_hline(yintercept = 15,
    color = "red") + theme(axis.title = element_text(size = 15),
    plot.title = element_text(size = 20))
```

```
## Warning: Removed 17 rows containing non-finite values (`stat_summary_bin()`).
```



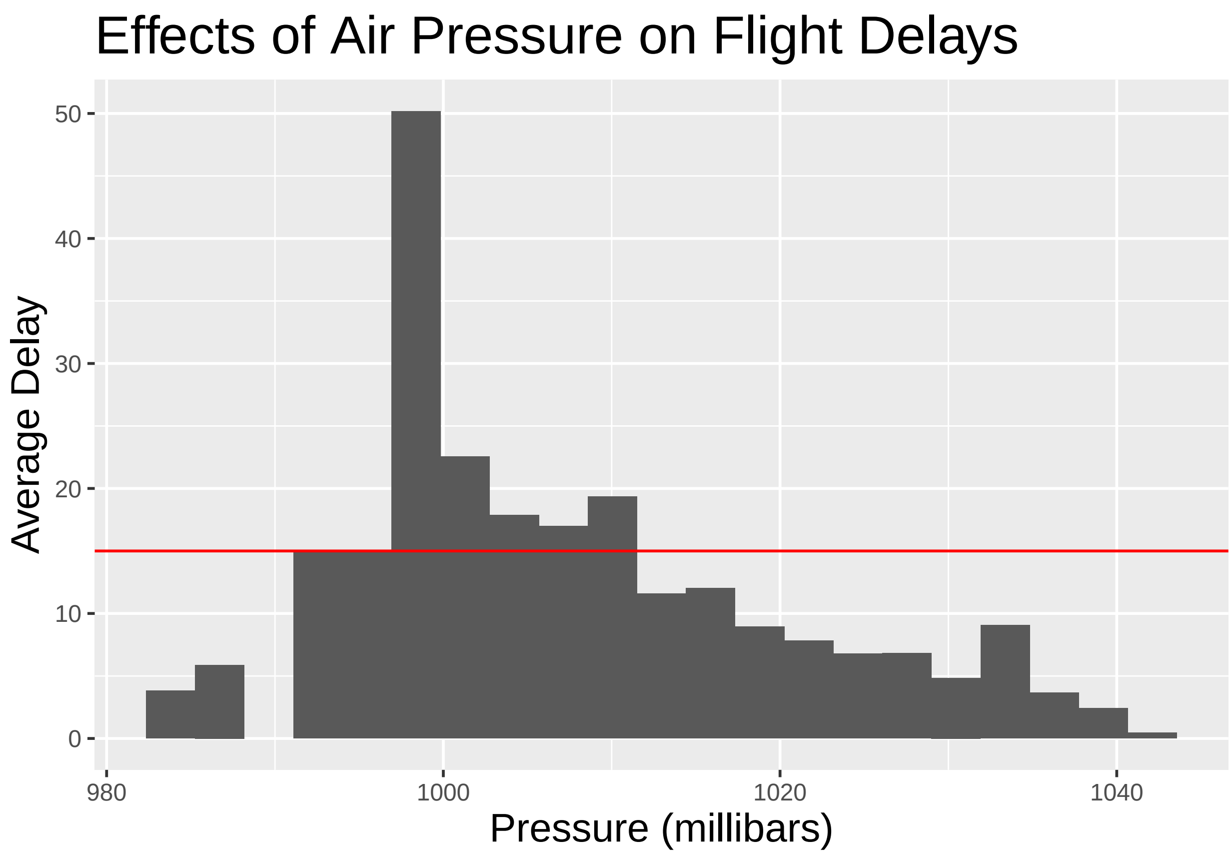

```
ggplot(wf, aes(x = humid, y = dep_delay)) + stat_summary_bin(fun = "mean",
    geom = "bar", bins = 20) + ylab("Average Delay") + xlab("Relative Humidity") +
    ggtitle("Effect of Humidity on Flights Delays") + geom_hline(yintercept = 15,
    color = "red") + theme(axis.title = element_text(size = 15),
    plot.title = element_text(size = 20))
```

```
## Warning: Removed 17 rows containing non-finite values (‘stat_summary_bin()‘).
```




```
ggplot(wf, aes(x = pressure, y = dep_delay)) + stat_summary_bin(fun = "mean",
    geom = "bar", bins = 20) + ylab("Average Delay") + xlab("Pressure (millibars)") +
    ggtitle("Effects of Air Pressure on Flight Delays") + geom_hline(yintercept = 15,
    color = "red") + theme(axis.title = element_text(size = 15),
    plot.title = element_text(size = 20))
```

```
## Warning: Removed 34615 rows containing non-finite values (‘stat_summary_bin()‘).
```



Takeways

It seems like wind speed, temperature, dew point, pressure, and humidity are the variables with strongest relationship with departure delay.

Modeling

- Goal: Create a model to help decide which variables are most useful and maybe get a hierarchy within them.

Checking Assumptions

Logistic regression has less assumptions than Linear regression. It requires a binary response, which we have with our 'sigDelay' variable. Also, there cannot be multicollinearity. Earlier from the correlation matrix, we saw that dew point and temperature had high correlation which could be an indication of multicollinearity. So dew point was chosen as the variable to add to the model because it had visually had the strongest relationship with the response. Finally, our sample size is still very large even after eliminating rows for missing data and removing variables.

```
wd = wf[, c(20, 26, 27, 29:32)]
sample <- sample(c(TRUE, FALSE), nrow(wd), replace = TRUE, prob = c(0.7,
    0.3))
```

```r
test = wd[sample, ]
train = wd[!sample, ]
test_x = test[, -1]
test_y = test$sigDelay

weather.model = glm(sigDelay ~ ., data = train, family = "binomial")

summary(weather.model)
```

```
##
## Call:
## glm(formula = sigDelay ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6090  -0.7003  -0.6259  -0.5178   2.2610
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) 25.6881357  1.3097771  19.613  < 2e-16 ***
## dewp         0.0062986  0.0005382  11.704  < 2e-16 ***
## humid        0.0029766  0.0006259   4.756 1.97e-06 ***
## wind_speed   0.0220762  0.0017015  12.974  < 2e-16 ***
## precip       3.8730951  0.6190816   6.256 3.94e-10 ***
## pressure    -0.0269180  0.0012756 -21.102  < 2e-16 ***
## visib       -0.0386733  0.0065024  -5.948 2.72e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 87660  on 87458  degrees of freedom
## Residual deviance: 86117  on 87452  degrees of freedom
##   (10326 observations deleted due to missingness)
## AIC: 86131
##
## Number of Fisher Scoring iterations: 4
```

```r
full_pred = predict(weather.model, newdata = test_x, type = "response")
delay_preds = rep(FALSE, length(full_pred))
delay_preds[full_pred > 0.4] = TRUE
print("TEST RESULTS")
```

```
## [1] "TEST RESULTS"
```

```r
table(delay_preds, test_y)
```

```
##            test_y
## delay_preds  FALSE   TRUE
##       FALSE 178507  48650
##       TRUE     529    348
```

```
mean(delay_preds == test_y)
```

```
## [1] 0.7843348
```

Model Results

Our model shows that dew point, wind speed, and pressure seem to be the most power predictors of a delay occurring. Variables such as precipitation and visibility should be taken with a grain of salt because they have very skewed distributions of data.

Using the model to predict on training data resulted in a 78% accuracy rate. At first glance this seems like a positive, however it overwhelmingly predicts flights to not be delayed. In fact, ~80% of flights were not delayed in this data set so always predicting no delay results in ~ 80% accuracy anyway. Therefore, this model is not ideal for prediction and should just be used to determine useful variables.