510 Project: Predicting Flight Delays

Shune Kawaoto

2025-06-26

Motivation

As the holiday season approaches, it is common for many people to travel, whether it be for visiting friends and family or for vacation. One method of long distance travel is through flying, and it would be great if flights can accurately be predicted to have a delayed arrival.

We chose to predict arrival delay rather than departure delay because:

1. Arrival time is more important when transferring flights is required, especially with short layover times.
2. Accommodations like hotels, hostels, and Airbnbs may only allow guests to check in within a certain time window and a delayed arrival can determine whether the guests get there in time.
3. After flying, some locations may require a shuttle, bus, or even train in order to leave the airport, and it is possible that a plane can arrive after these services are no longer running.

It is also possible for flight situations to change: while the aircraft can leave the gate on time (and therefore classified as an on-time departure) it is possible that the plane can be delayed during the taxi and takeoff process. The opposite is also true where a flight can have a delayed departure but arrive on time. Because of this last point, we are taking the perspective of a passenger during a flight who wants to predict whether or not the plane we're on will have a delayed arrival.

Importing Libraries & Data

A quick note is that we obtained this data from Kaggle, and it records domestic flight data in the U.S. from 2019 - 2023.

```r
install.packages('ggcorrplot')
```

```
## package 'ggcorrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\kawao\AppData\Local\Temp\RtmpiojU3E\downloaded_packages
```

```r
library(ggcorrplot)
library(MASS)
library(car)
library(skimr)
library(tidyverse)
library(caret)
library(glmnet)

flights_full <- read.csv("flights_sample_3m.csv")
```

EDA & Data Cleaning

```
head(flights_full)
```

```
##      FL_DATE                AIRLINE                AIRLINE_DOT AIRLINE_CODE
## 1 2019-01-09  United Air Lines Inc.  United Air Lines Inc.: UA           UA
## 2 2022-11-19   Delta Air Lines Inc.   Delta Air Lines Inc.: DL           DL
## 3 2022-07-22  United Air Lines Inc.  United Air Lines Inc.: UA           UA
## 4 2023-03-06   Delta Air Lines Inc.   Delta Air Lines Inc.: DL           DL
## 5 2020-02-23       Spirit Air Lines       Spirit Air Lines: NK           NK
## 6 2019-07-31 Southwest Airlines Co. Southwest Airlines Co.: WN           WN
##   DOT_CODE FL_NUMBER ORIGIN         ORIGIN_CITY DEST            DEST_CITY
## 1    19977      1562    FLL Fort Lauderdale, FL  EWR           Newark, NJ
## 2    19790      1149    MSP     Minneapolis, MN  SEA          Seattle, WA
## 3    19977       459    DEN          Denver, CO  MSP      Minneapolis, MN
## 4    19790      2295    MSP     Minneapolis, MN  SFO     San Francisco, CA
## 5    20416       407    MCO         Orlando, FL  DFW Dallas/Fort Worth, TX
## 6    19393       665    DAL          Dallas, TX  OKC     Oklahoma City, OK
##   CRS_DEP_TIME DEP_TIME DEP_DELAY TAXI_OUT WHEELS_OFF WHEELS_ON TAXI_IN
## 1         1155     1151        -4       19       1210      1443       4
## 2         2120     2114        -6        9       2123      2232      38
## 3          954     1000         6       20       1020      1247       5
## 4         1609     1608        -1       27       1635      1844       9
## 5         1840     1838        -2       15       1853      2026      14
## 6         1010     1237       147       15       1252      1328       3
##   CRS_ARR_TIME ARR_TIME ARR_DELAY CANCELLED CANCELLATION_CODE DIVERTED
## 1         1501     1447       -14         0                          0
## 2         2315     2310        -5         0                          0
## 3         1252     1252         0         0                          0
## 4         1829     1853        24         0                          0
## 5         2041     2040        -1         0                          0
## 6         1110     1331       141         0                          0
##   CRS_ELAPSED_TIME ELAPSED_TIME AIR_TIME DISTANCE DELAY_DUE_CARRIER
## 1              186          176      153     1065                NA
## 2              235          236      189     1399                NA
## 3              118          112       87      680                NA
## 4              260          285      249     1589                 0
## 5              181          182      153      985                NA
## 6               60           54       36      181               141
##   DELAY_DUE_WEATHER DELAY_DUE_NAS DELAY_DUE_SECURITY DELAY_DUE_LATE_AIRCRAFT
## 1                NA            NA                 NA                      NA
## 2                NA            NA                 NA                      NA
## 3                NA            NA                 NA                      NA
## 4                 0            24                  0                       0
## 5                NA            NA                 NA                      NA
## 6                 0             0                  0                       0
```

```
skim(flights_full)
```

Table 1: Data summary

Name	flights_full
Number of rows	3000000
Number of columns	32

Table 1: Data summary

Column type frequency:	
character	9
numeric	23
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
FL_DATE	0	1	10	10	0	1704	0
AIRLINE	0	1	9	34	0	18	0
AIRLINE_DOT	0	1	13	38	0	18	0
AIRLINE_CODE	0	1	2	2	0	18	0
ORIGIN	0	1	3	3	0	380	0
ORIGIN_CITY	0	1	8	34	0	373	0
DEST	0	1	3	3	0	380	0
DEST_CITY	0	1	8	34	0	373	0
CANCELLATION_CODE	0	1	0	1	2920860	5	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
DOT_CODE	0	1.00	19976.29	377.28	19393	19790	19930	20368	20452	
FL_NUMBER	0	1.00	2511.54	1747.26	1	1051	2152	3797	9562	
CRS_DEP_TIME	0	1.00	1327.06	485.88	1	915	1320	1730	2359	
DEP_TIME	77615	0.97	1329.78	499.31	1	916	1323	1739	2400	
DEP_DELAY	77644	0.97	10.12	49.25	-90	-6	-2	6	2966	
TAXI_OUT	78806	0.97	16.64	9.19	1	11	14	19	184	
WHEELS_OFF	78806	0.97	1352.36	500.87	1	931	1336	1752	2400	
WHEELS_ON	79944	0.97	1462.50	527.24	1	1049	1501	1908	2400	
TAXI_IN	79944	0.97	7.68	6.27	1	4	6	9	249	
CRS_ARR_TIME	0	1.00	1490.56	511.55	1	1107	1516	1919	2400	
ARR_TIME	79942	0.97	1466.51	531.84	1	1053	1505	1913	2400	
ARR_DELAY	86198	0.97	4.26	51.17	-96	-16	-7	7	2934	
CANCELLED	0	1.00	0.03	0.16	0	0	0	0	1	
DIVERTED	0	1.00	0.00	0.05	0	0	0	0	1	
CRS_ELAPSED_TIME	14	1.00	142.28	71.56	1	90	125	172	705	
ELAPSED_TIME	86198	0.97	136.62	71.68	15	84	120	167	739	
AIR_TIME	86198	0.97	112.31	69.75	8	61	95	142	692	
DISTANCE	0	1.00	809.36	587.89	29	377	651	1046	5812	
DELAY_DUE_CARRIER	2466137	0.18	24.76	71.77	0	0	4	23	2934	
DELAY_DUE_WEATHER	2466137	0.18	3.99	32.41	0	0	0	0	1653	
DELAY_DUE_NAS	2466137	0.18	13.16	33.16	0	0	0	17	1741	
DELAY_DUE_SECURITY	2466137	0.18	0.15	3.58	0	0	0	0	1185	
DELAY_DUE_LATE_AIRCRAFT	2466137	0.18	25.47	55.77	0	0	0	30	2557	

One thing that we noticed is that some variables like FL_DATE, AIRLINE, ..., and DIVERTED (see code below) were incorrectly encoded as characters and numeric values. We converted these to factors as our first step. The second thing we want to point out is that our response variable ARR_DELAY is a numeric, continuous value. Because we want to predict whether the flight is delayed or not, we changed ARR_DELAY to be a binary factor variable, where all values greater than 0 are considered "delayed", denoted as 1, and less than equal to 0 are considered "not delayed", denoted as 0.

```
flights_full$FL_DATE <- as.factor(as.character(flights_full$FL_DATE))
flights_full$AIRLINE <- as.factor(flights_full$AIRLINE)
flights_full$AIRLINE_DOT <- as.factor(flights_full$AIRLINE_DOT)
flights_full$AIRLINE_CODE <- as.factor(flights_full$AIRLINE_CODE)
flights_full$DOT_CODE <- as.factor(flights_full$DOT_CODE)
flights_full$FL_NUMBER <- as.factor(flights_full$FL_NUMBER)
flights_full$ORIGIN <- as.factor(flights_full$ORIGIN)
flights_full$ORIGIN_CITY <- as.factor(flights_full$ORIGIN_CITY)
flights_full$DEST <- as.factor(flights_full$DEST)
flights_full$DEST_CITY <- as.factor(flights_full$DEST_CITY)
flights_full$CANCELLED <- as.factor(as.character(flights_full$CANCELLED))
flights_full$CANCELLATION_CODE <- as.factor(flights_full$CANCELLATION_CODE)
flights_full$DIVERTED <- as.factor(as.character(flights_full$DIVERTED))

flights_full$DELAYED <- as.factor(ifelse(flights_full$ARR_DELAY > 0, 1, 0))

flights <- flights_full %>% dplyr::select(-ARR_DELAY)
# delayed = 1, early or on-time = 0
```

The second thing we decided to do was to delete all variables that we didn't need:

- AIRLINE_DOT, AIRLINE_CODE, and DOT_CODE all were unique identifiers for every specific airline, so we decided to keep AIRLINE and delete these three variables instead.
- ORIGIN_CITY & DEST_CITY were locations that airports are in, so they overlap quite a lot with ORIGIN and DEST, which gives us the airport codes. Since we are more concerned with where the planes depart and arrive, which is the airport itself, we decided to delete ORIGIN_CITY & DEST_CITY.
- If flights are cancelled, then there is no possible way for flights to depart in the first place, so there is no arrival data. Therefore, because we are trying to predict arrival data, we deleted CANCELLED and CANCELLATION_CODE.
- We are taking the perspective of someone who is midflight and wants to predict if their plane will arrive on time. Therefore, we can only use data that we know prior to being airborne. Using this condtion, these following variables were also removed:
  - WHEELS_ON is the time when the plane lands (wheels touch the floor).
  - TAXI_IN is the time between landing and being taxied to the arrival gate.
  - ARR_TIME is the recorded arrival time, not the scheduled arrival time.
  - ELAPSED_TIME is the recorded time of how long the flight took.
  - AIR_TIME is the recorded time of how long the plane was airborne.
  - DELAY_DUE_CARRIER is how many minutes the departure and arrival delay was attributed to the plane.
  - DELAY_DUE_WEATHER is how many minutes the departure and arrival delay was attributed to the weather.
  - DELAY_DUE_NAS is how many minutes the departure and arrival delay was attributed to the NAS (National Airspace System).

  - DELAY_DUE_SECURITY is how many minutes the departure and arrival delay was attributed to security issues and protocols.
  - DELAY_DUE_LATE_AIRCRAFT is how many minutes the departure and arrival delay was attributed to the aircraft arriving late prior to departure.

```r
flights <- flights %>%
  dplyr::select(-c(AIRLINE_DOT, AIRLINE_CODE, DOT_CODE, ORIGIN_CITY, DEST_CITY, WHEELS_ON,
                   TAXI_IN, ARR_TIME, CANCELLED, CANCELLATION_CODE, ELAPSED_TIME, AIR_TIME,
                   DELAY_DUE_CARRIER, DELAY_DUE_WEATHER, DELAY_DUE_NAS, DELAY_DUE_SECURITY,
                   DELAY_DUE_LATE_AIRCRAFT))
```

The next step is to change FL_DATE. It was imported in the yyyy-mm-dd format, and we decided to change that to three separate variables YEAR, MONTH, and DAY instead. Since the flight data is from 2019 - 2023, we believe that the COVID-19 pandemic could've had a possible effect on the flights. It is also important to note that in the airline industry, days of the week are considered more important than days of the month. Therefore, we encoded YEAR and DAY as factors, with levels ("2019", "2020", "2021", "2022", "2023") and ("Sunday", "Monday", "Tuesday", "Wednesday". "Thursday", "Friday", "Saturday"), respectively.

```r
day_of_week <- weekdays(as.Date(as.character(flights$FL_DATE), format = "%Y-%m-%d"))

# Making FL_DATE into three separate columns: Year, Month, Day of the Week
flights <- flights %>%
  separate(col = FL_DATE, into = c("YEAR", "MONTH", "DAY"), sep = "-", convert = TRUE)

# COVID-19 happened in this time span so I will be making YEAR a categorical variable.
flights$YEAR <- as.factor(flights$YEAR)

flights$DAY <- as.factor(day_of_week)
```

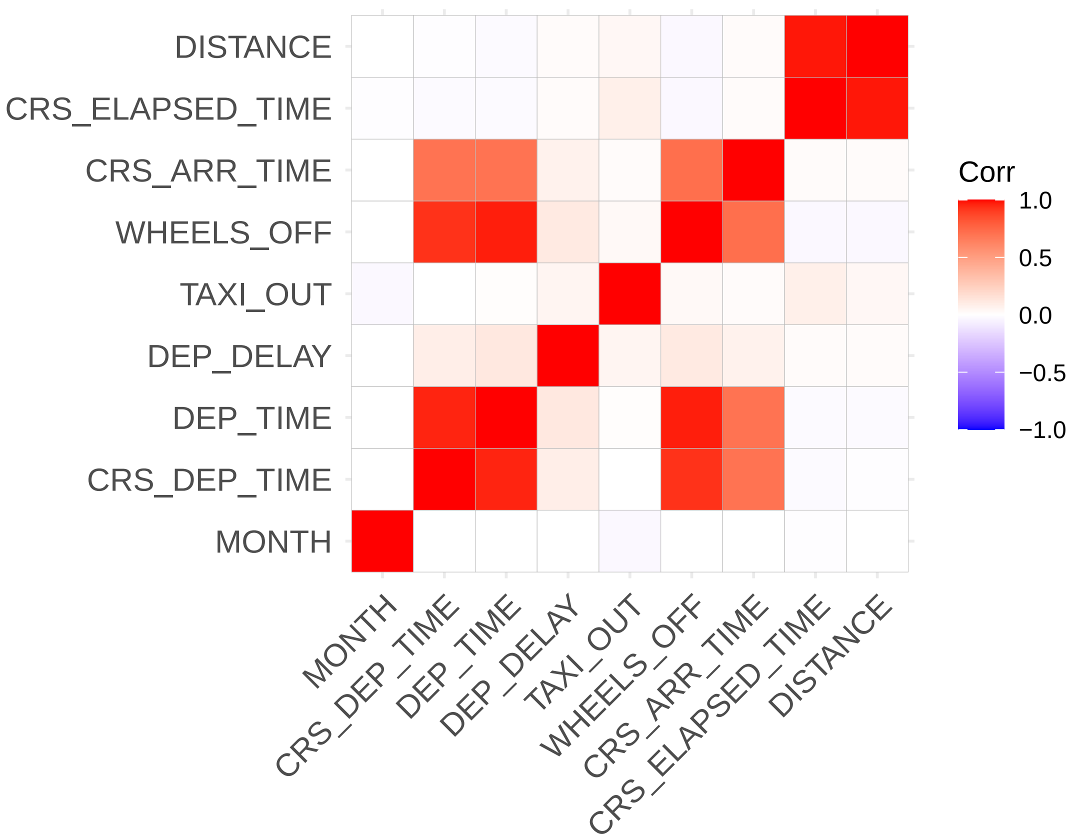
The following variables are numeric and have been recorded in the "hhmm" form, which doesn't really make sense for our analysis. Therefore we are changing them into "minutes after midnight".

- CRS_DEP_TIME is the scheduled departure time.
- CRS_ARR_TIME is the scheduled arrival time.
- DEP_TIME is the recorded departure time.
- WHEELS_OFF is the recorded time when the flight takes off (wheels leave the floor).

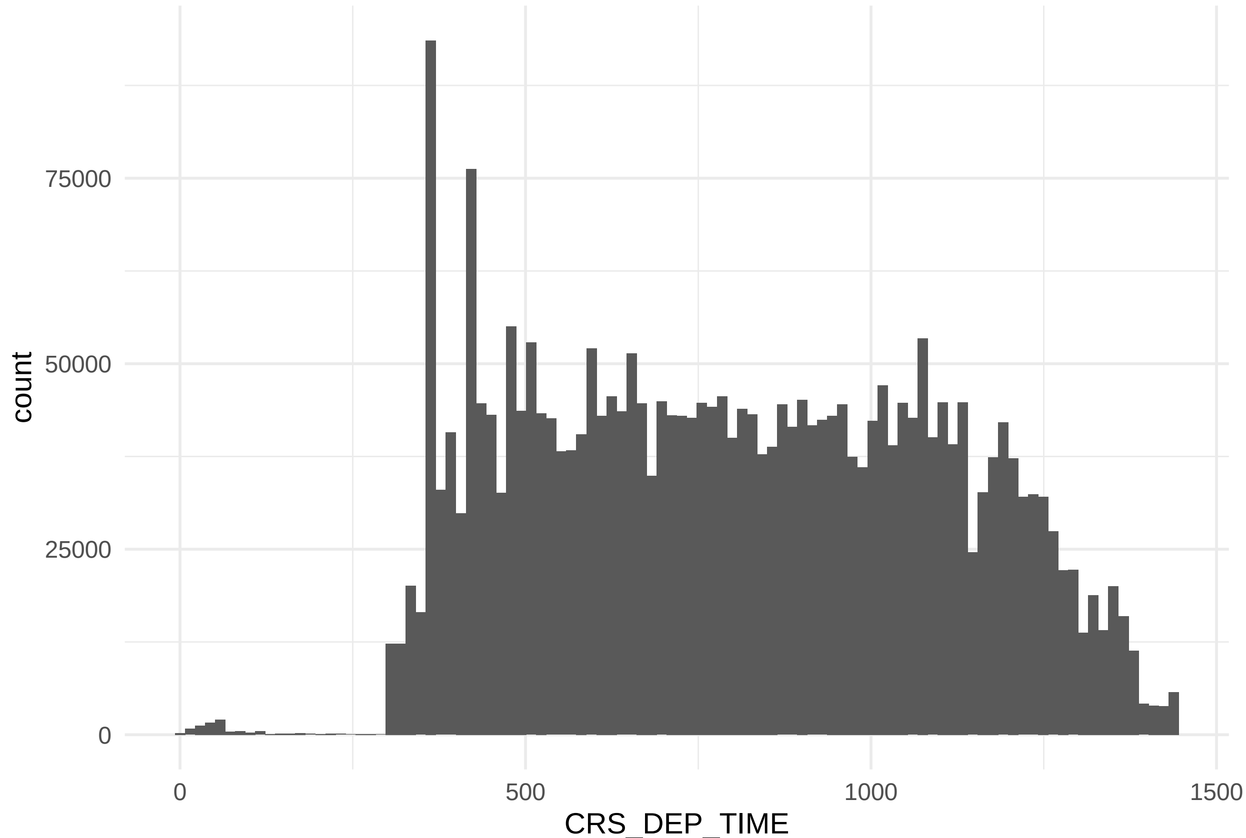
```r
# CRS_DEP_TIME, CRS_ARR_TIME, DEP_TIME, WHEELS_OFF have to be converted
# into numeric values that make sense (same format as CRS_ELAPSED_TIME).
#
# Solution: I will make them into minutes after midnight.
hours_crs_dep <- floor(flights$CRS_DEP_TIME / 100)
mins_crs_dep <- flights$CRS_DEP_TIME %% 100
flights$CRS_DEP_TIME <- hours_crs_dep * 60 + mins_crs_dep

hours_crs_arr <- floor(flights$CRS_ARR_TIME / 100)
mins_crs_arr <- flights$CRS_ARR_TIME %% 100
flights$CRS_ARR_TIME <- hours_crs_arr * 60 + mins_crs_arr

hours_dep <- floor(flights$DEP_TIME / 100)
mins_dep <- flights$DEP_TIME %% 100
flights$DEP_TIME <- hours_dep * 60 + mins_dep

hours_off <- floor(flights$WHEELS_OFF / 100)
mins_off <- flights$WHEELS_OFF %% 100
flights$WHEELS_OFF <- hours_off * 60 + mins_off
```

We believe the data is clean enough to starting creating visualizations at this point to better understand the data. We first remove all observations that contain NA and then create a correlation heat map and histograms for our numeric variables, and create bar graphs for our categorical variables. The heat map will tell us the relationships between the variables, and the histograms will give us a clear view about the distribution of the data. The bar graphs will give us a picture of the frequencies of each category in the variables.

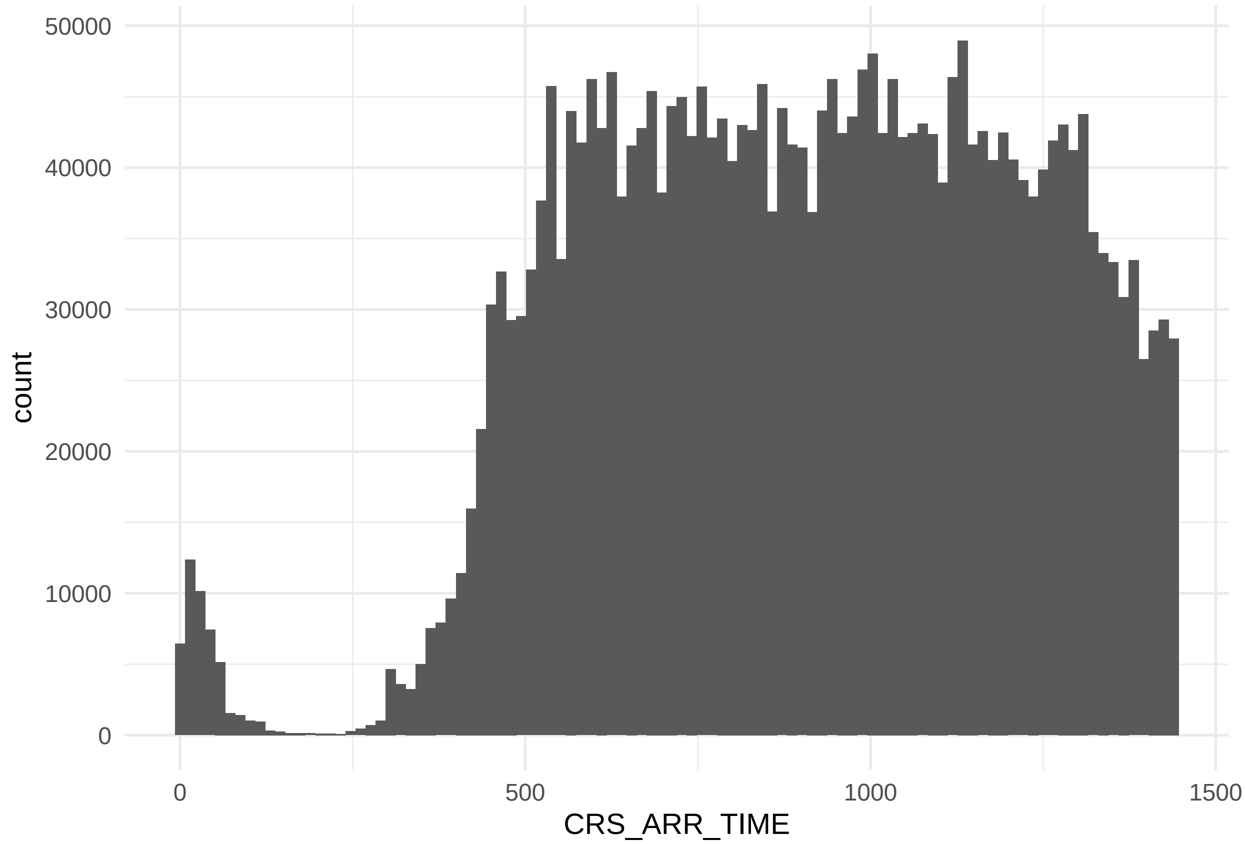
```
flights <- na.omit(flights)

# Correlation plot (numeric)
ggcorrplot(cor(flights[, sapply(flights, is.numeric)]), method = "square")
```



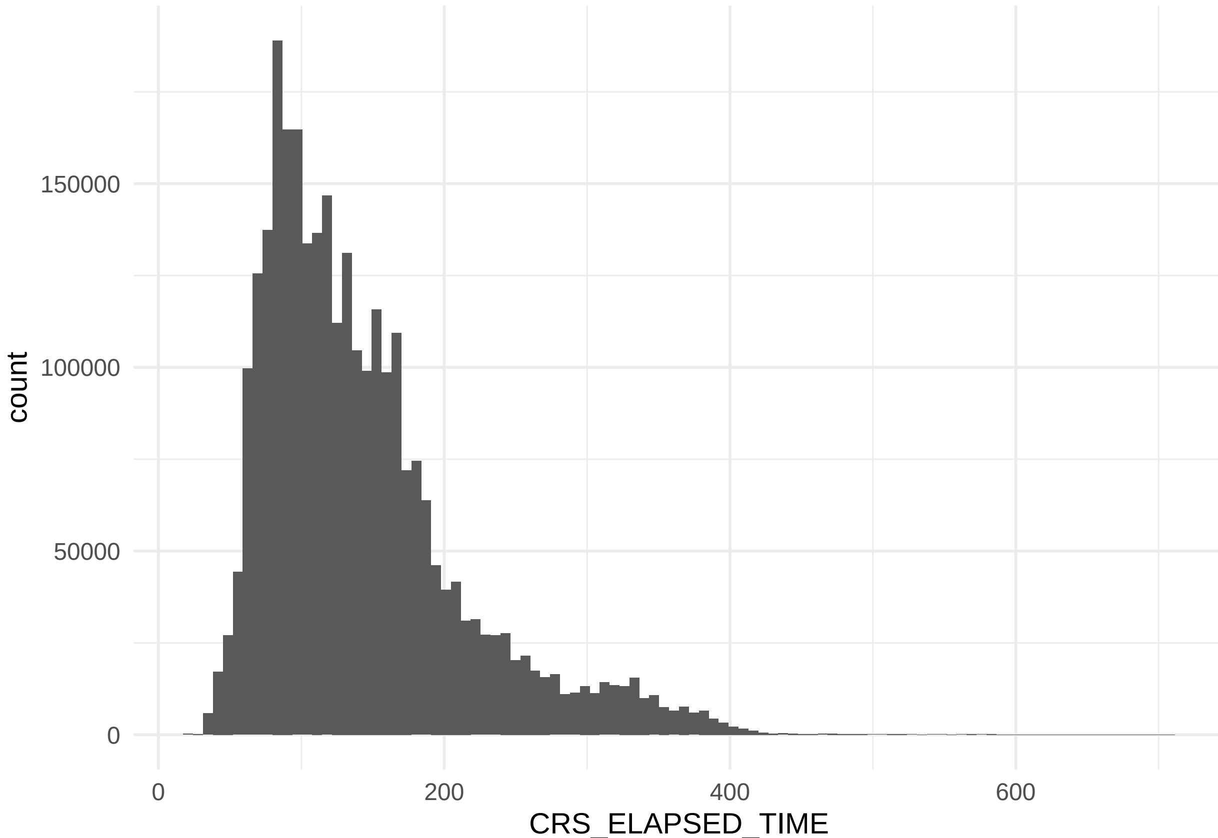

```
# Histograms (numeric)
ggplot(flights, aes(x = CRS_DEP_TIME)) + geom_histogram(bins = 100) + theme_minimal()
```



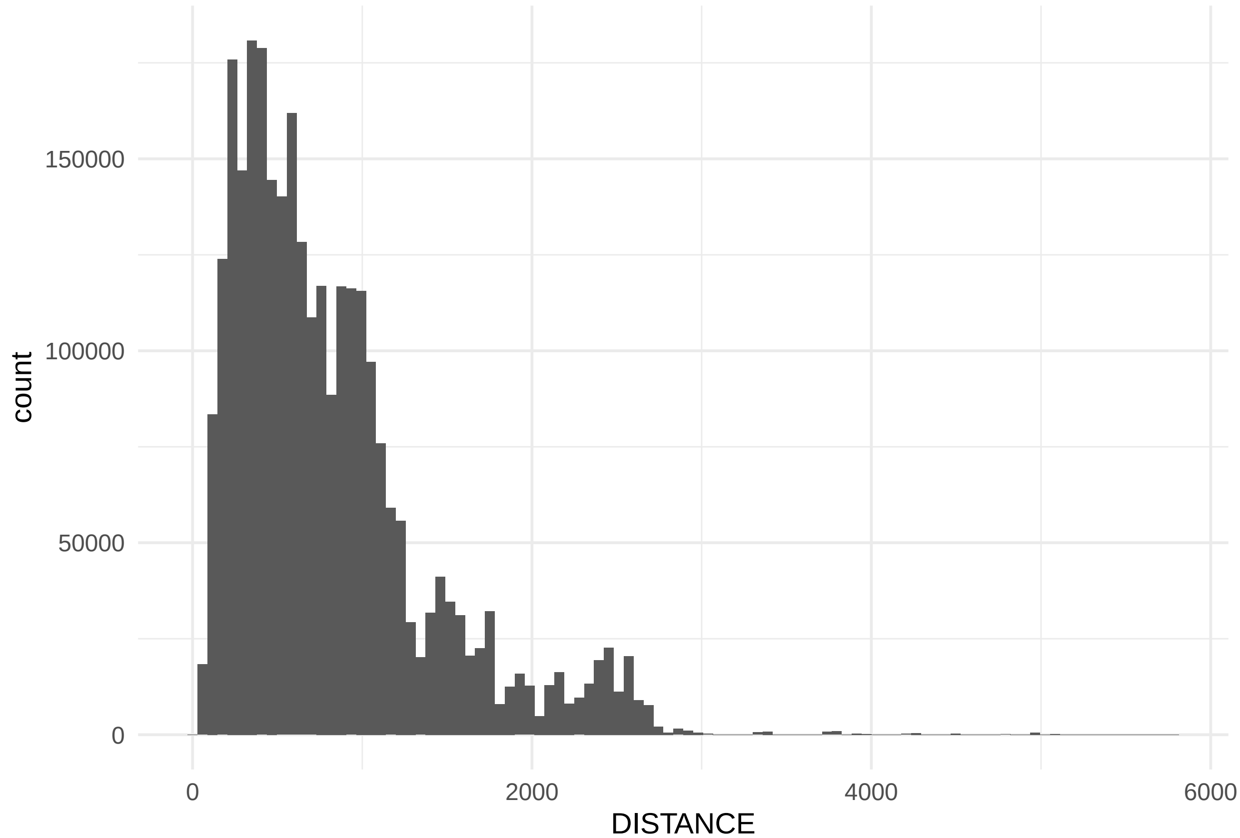

```
ggplot(flights, aes(x = CRS_ARR_TIME)) + geom_histogram(bins = 100) + theme_minimal()
```



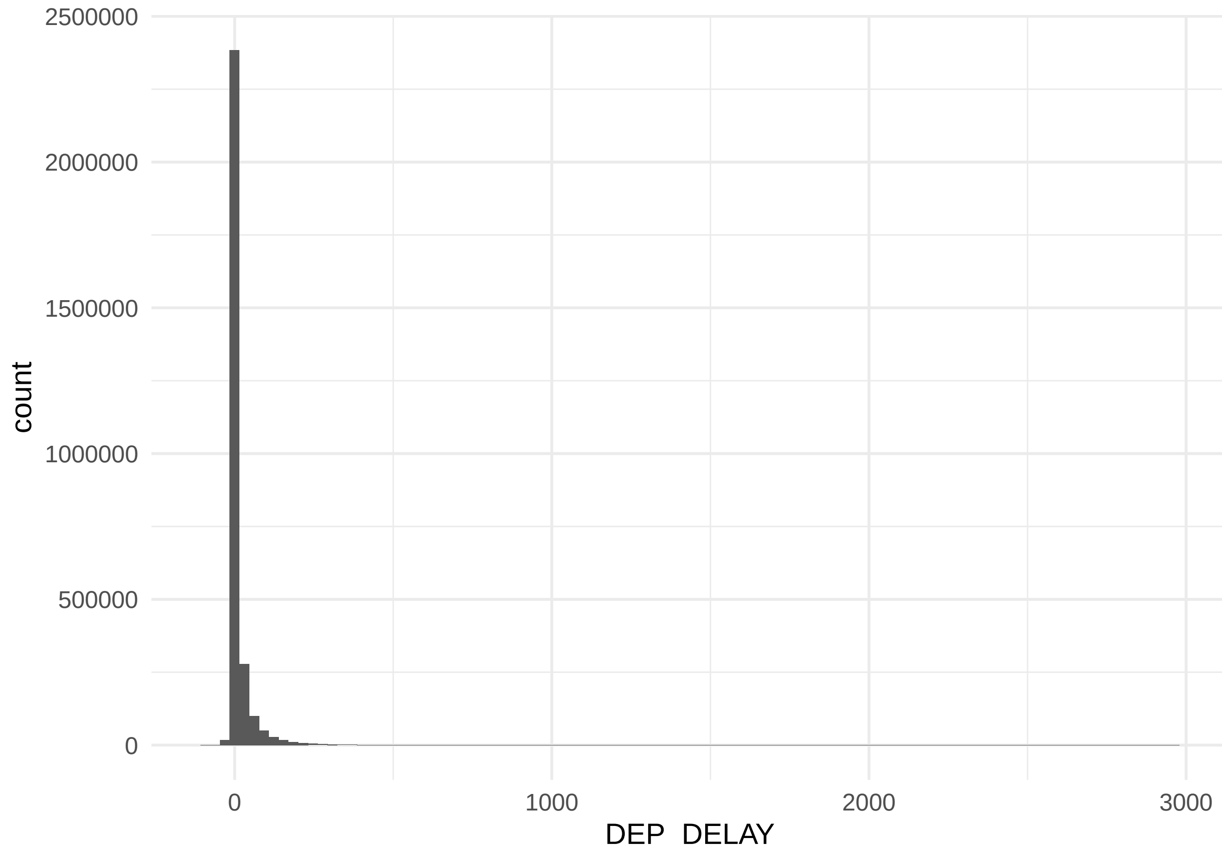

```
ggplot(flights, aes(x = CRS_ELAPSED_TIME)) + geom_histogram(bins = 100) + theme_minimal()
```



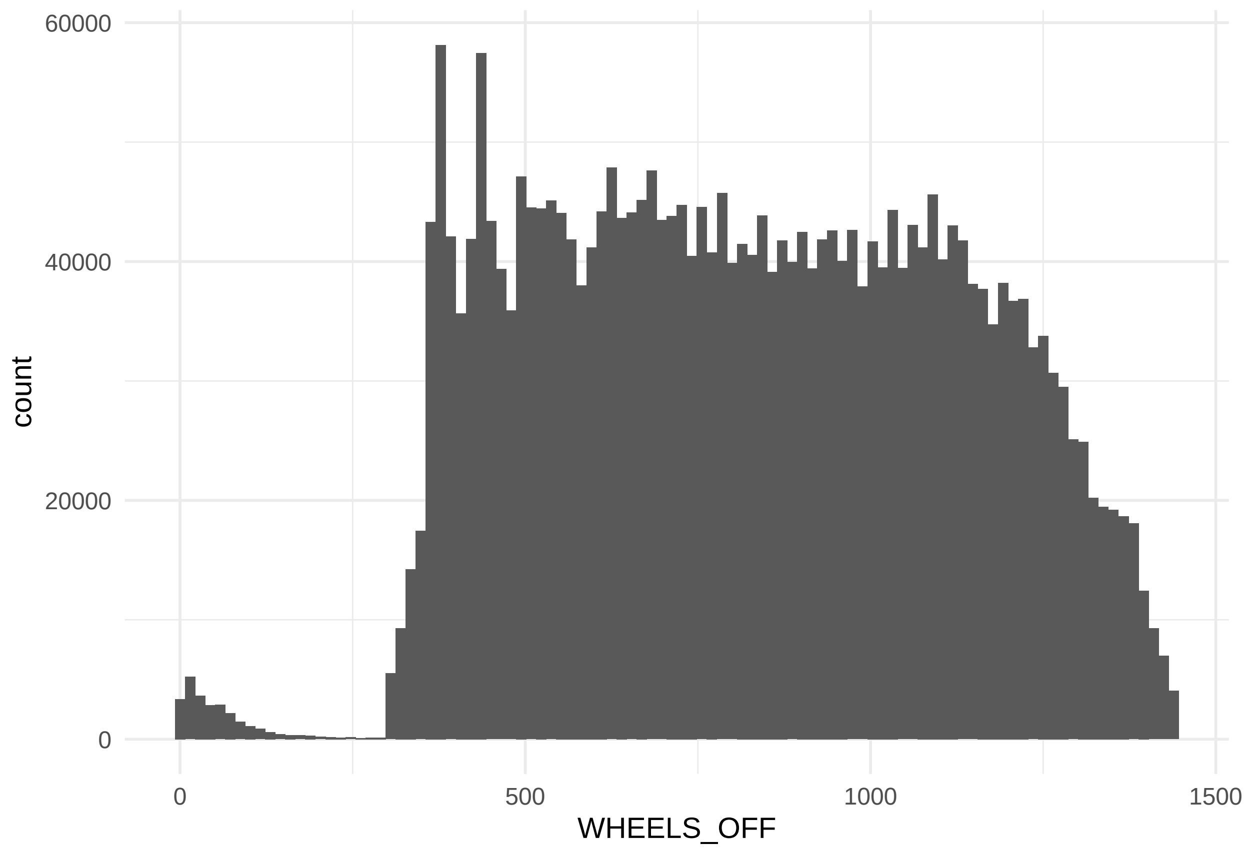

```
ggplot(flights, aes(x = DISTANCE)) + geom_histogram(bins = 100) + theme_minimal()
```



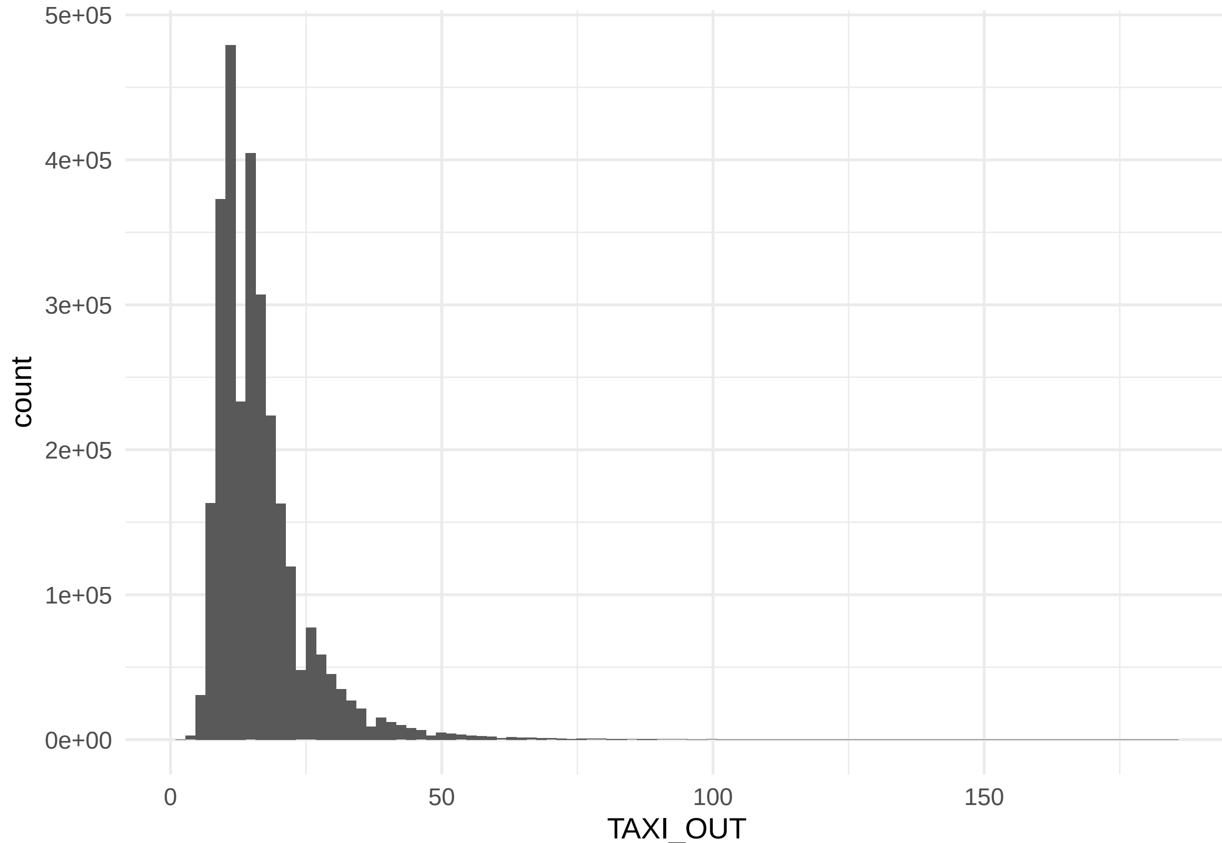

```
ggplot(flights, aes(x = DEP_DELAY)) + geom_histogram(bins = 100) + theme_minimal()
```



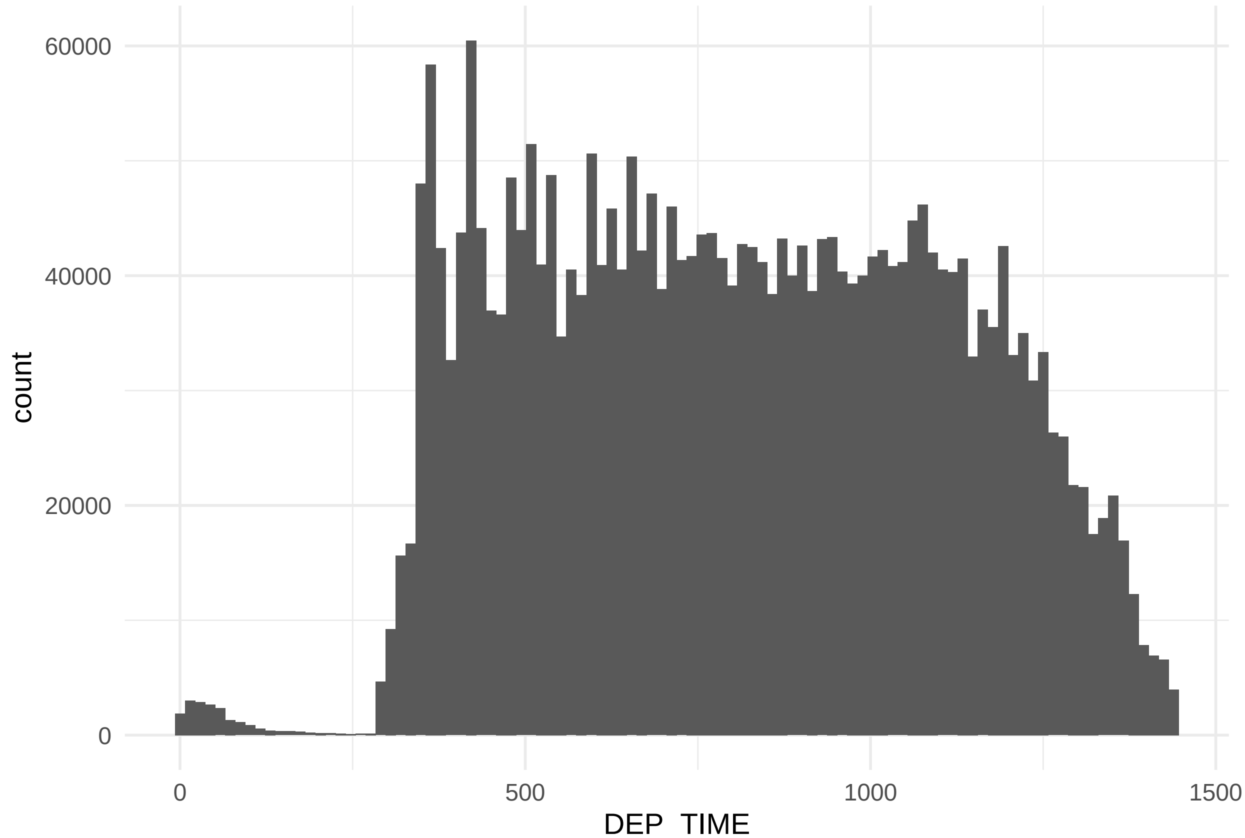

```
ggplot(flights, aes(x = WHEELS_OFF)) + geom_histogram(bins = 100) + theme_minimal()
```



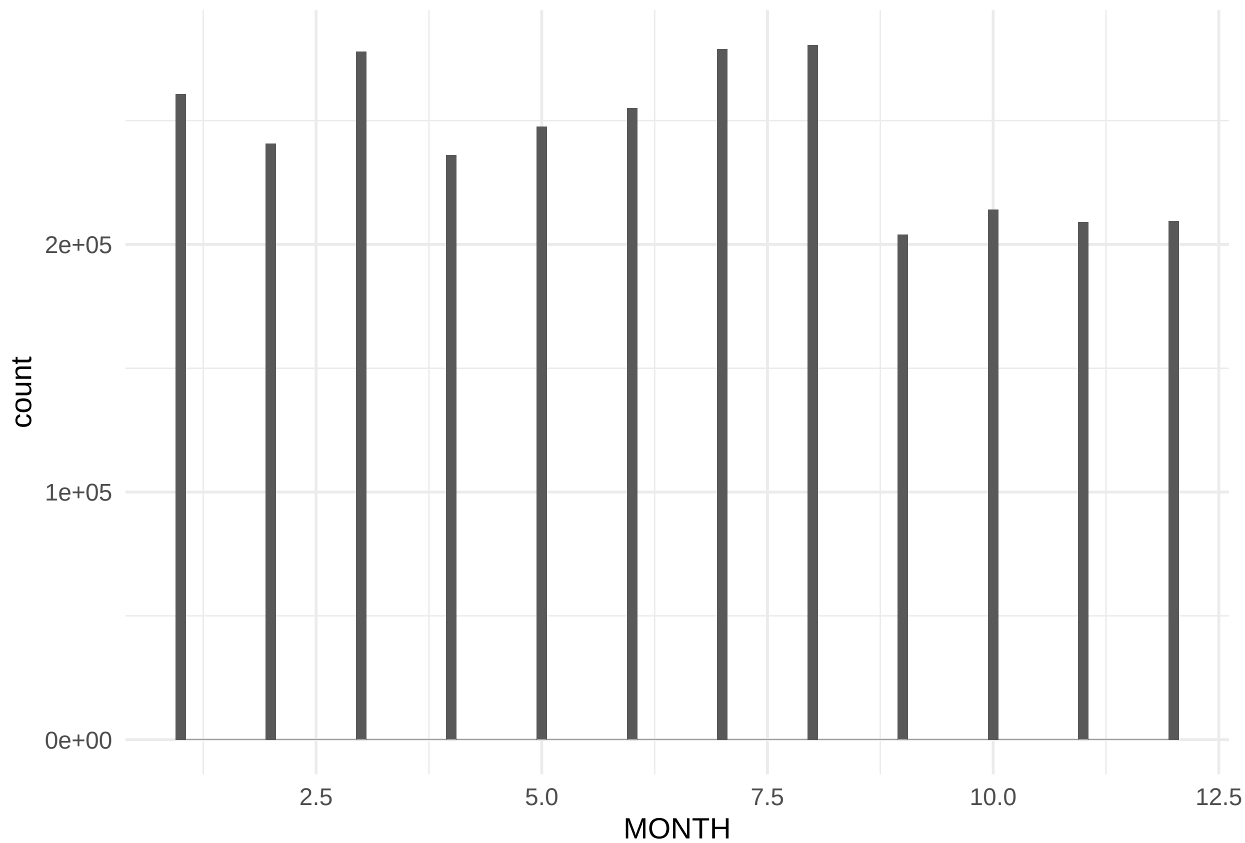

```
ggplot(flights, aes(x = TAXI_OUT)) + geom_histogram(bins = 100) + theme_minimal()
```



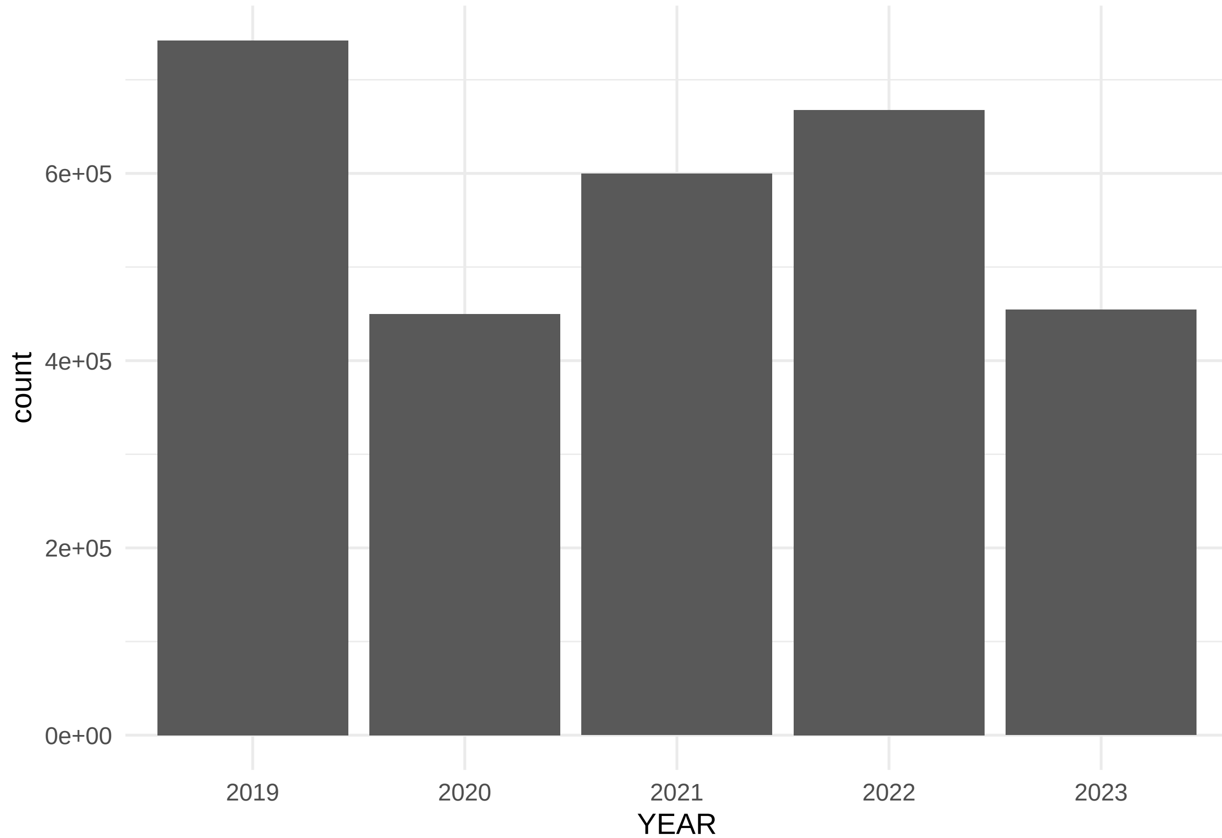

```
ggplot(flights, aes(x = DEP_TIME)) + geom_histogram(bins = 100) + theme_minimal()
```



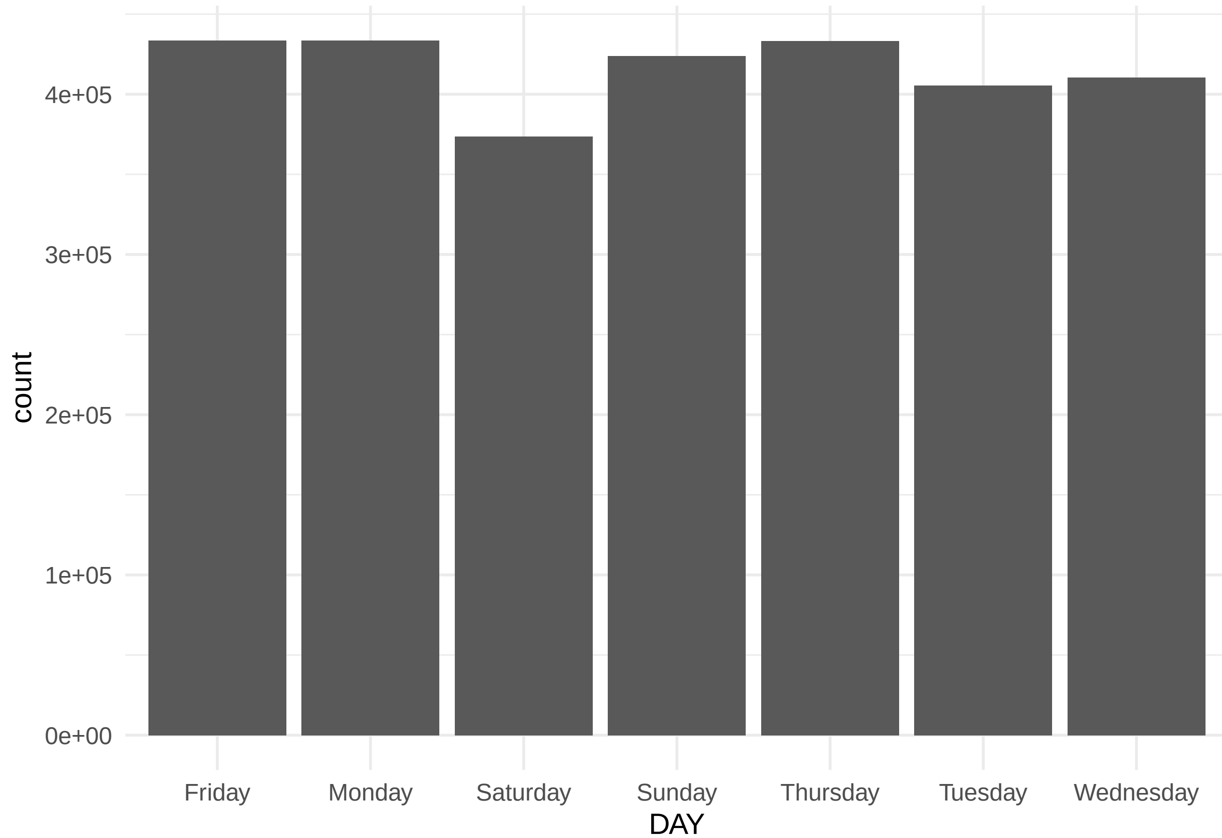

```
ggplot(flights, aes(x = MONTH)) + geom_histogram(bins = 100) + theme_minimal()
```



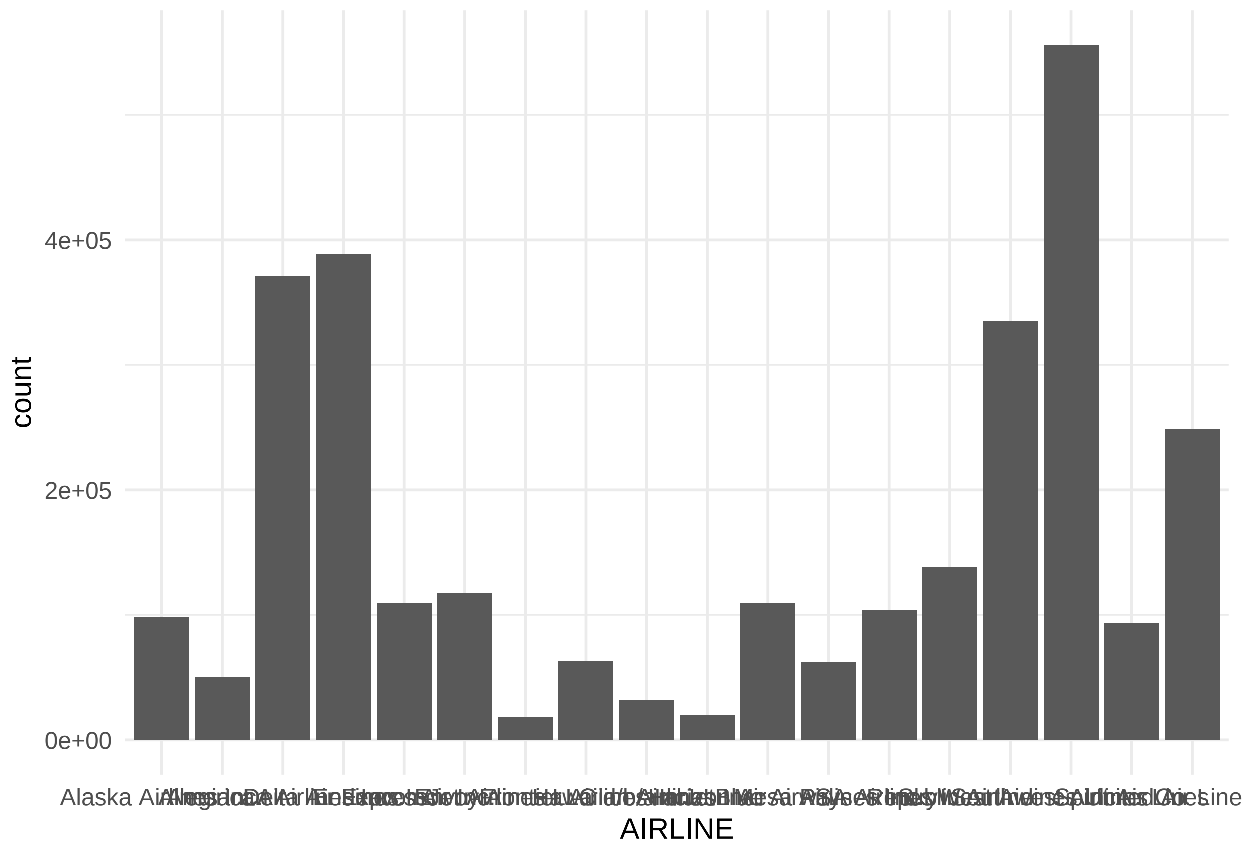

```r
# Bar graph (categorical)
ggplot(flights, aes(x = YEAR)) + geom_bar() + theme_minimal()
```



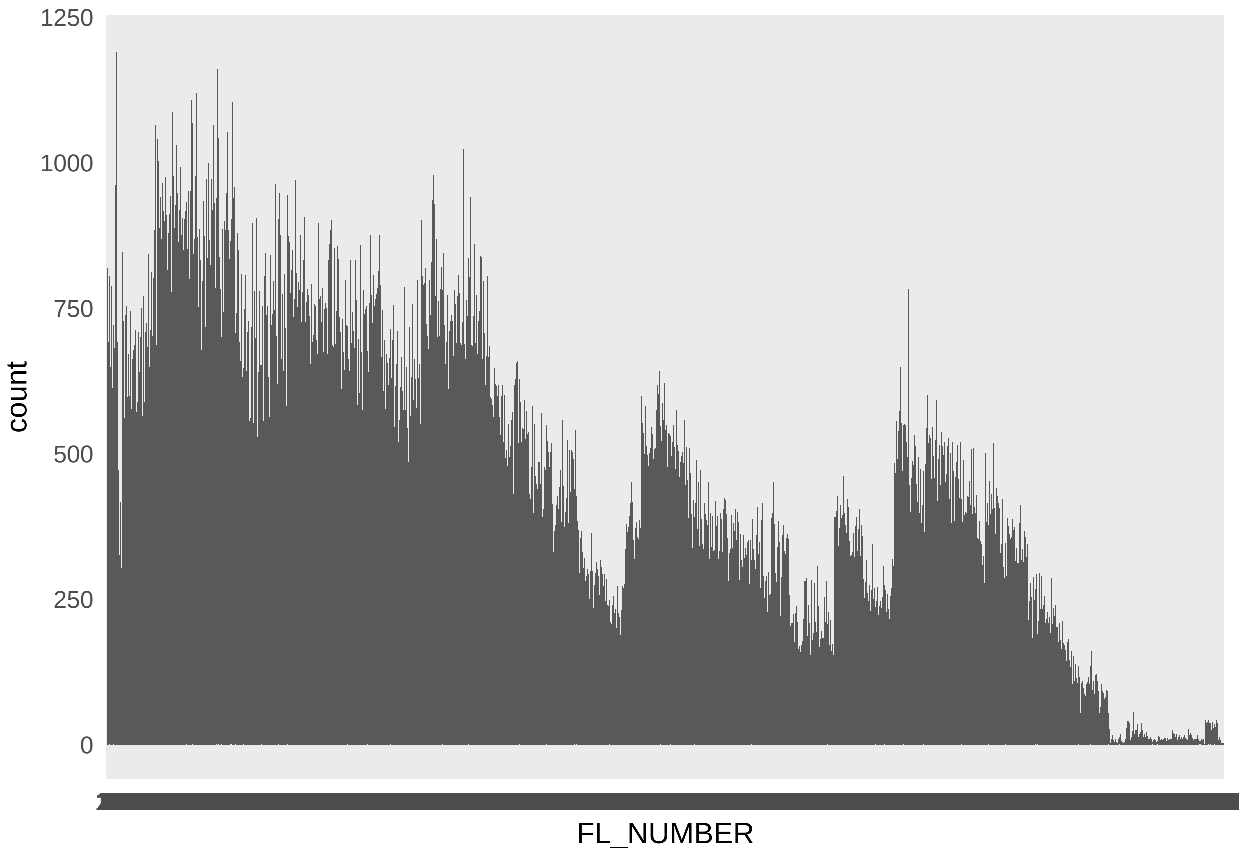

```
ggplot(flights, aes(x = DAY)) + geom_bar() + theme_minimal()
```



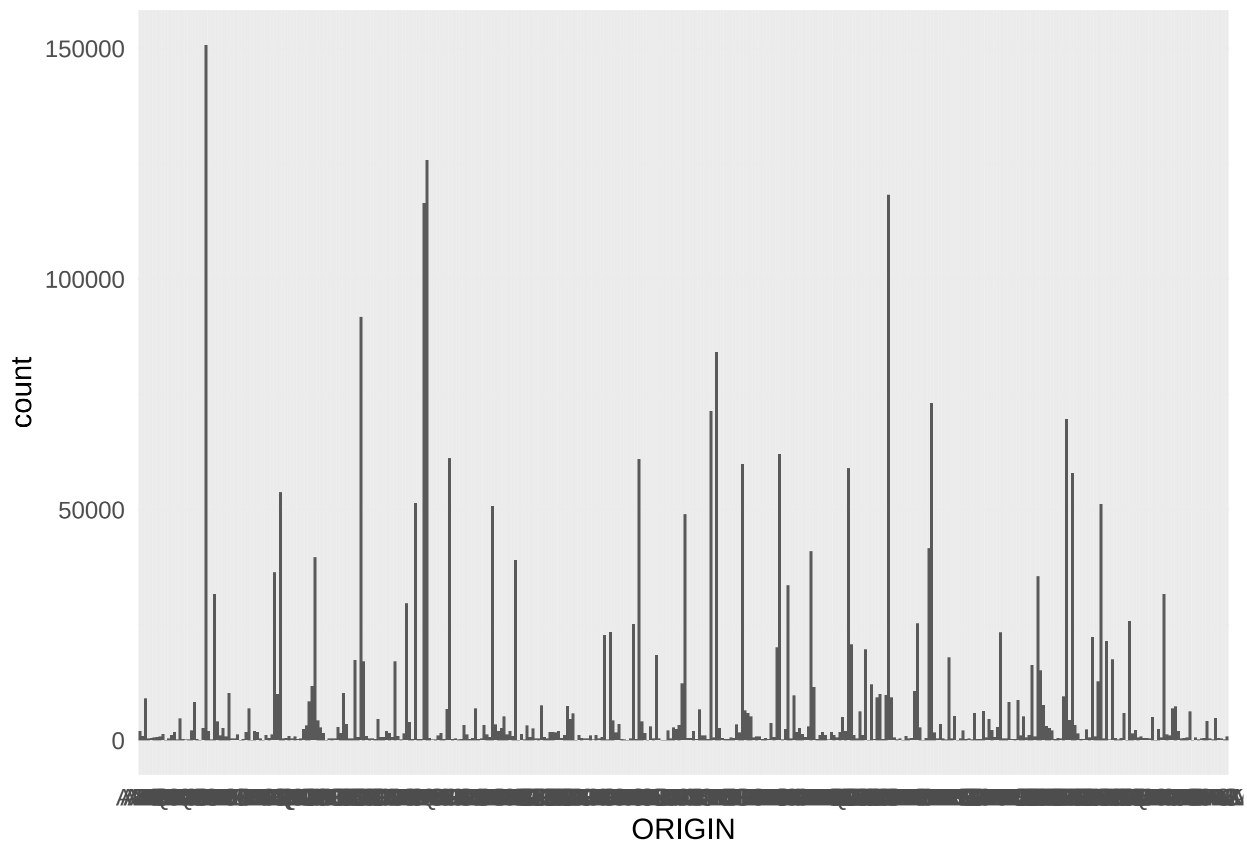

```
ggplot(flights, aes(x = AIRLINE)) + geom_bar() + theme_minimal()
```



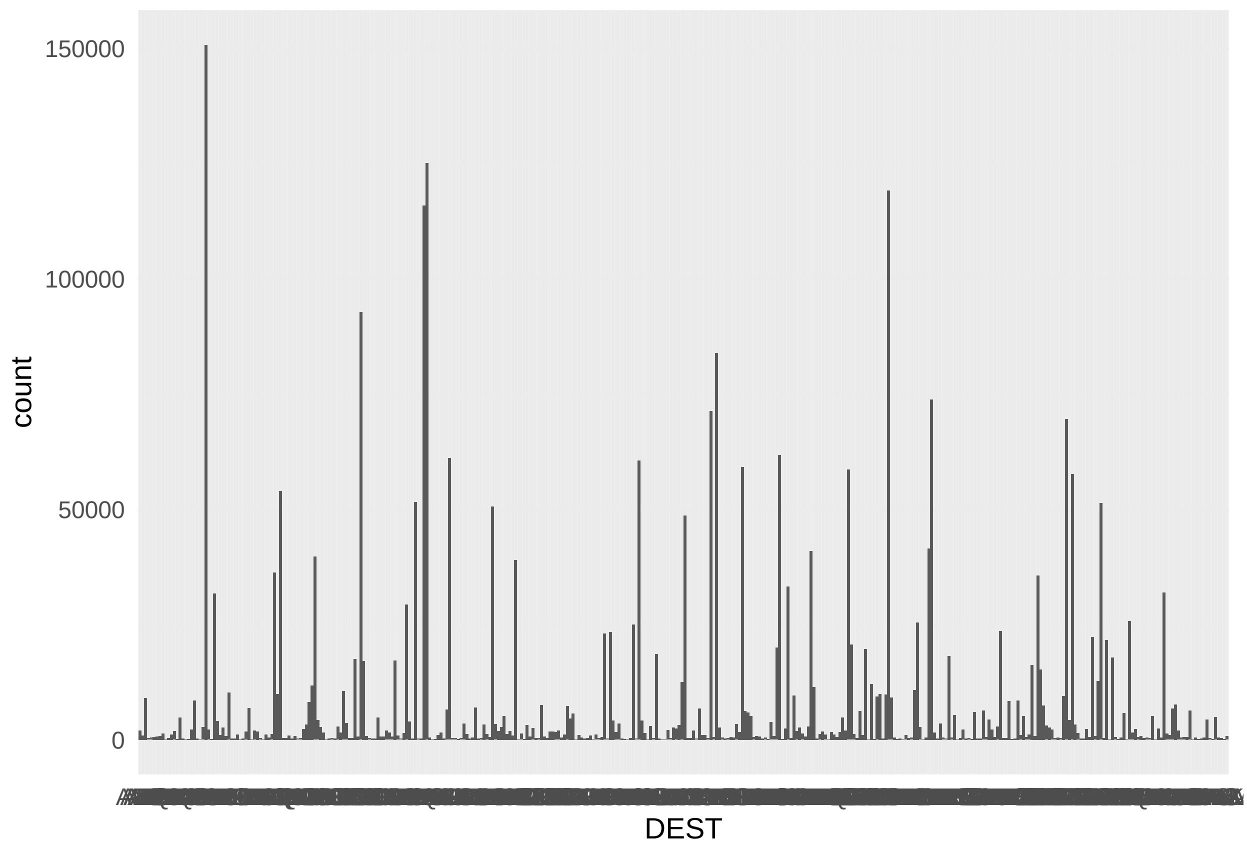

```r
ggplot(flights, aes(x = FL_NUMBER)) + geom_bar() + theme_minimal()
```




```r
ggplot(flights, aes(x = ORIGIN)) + geom_bar() + theme_minimal()
```

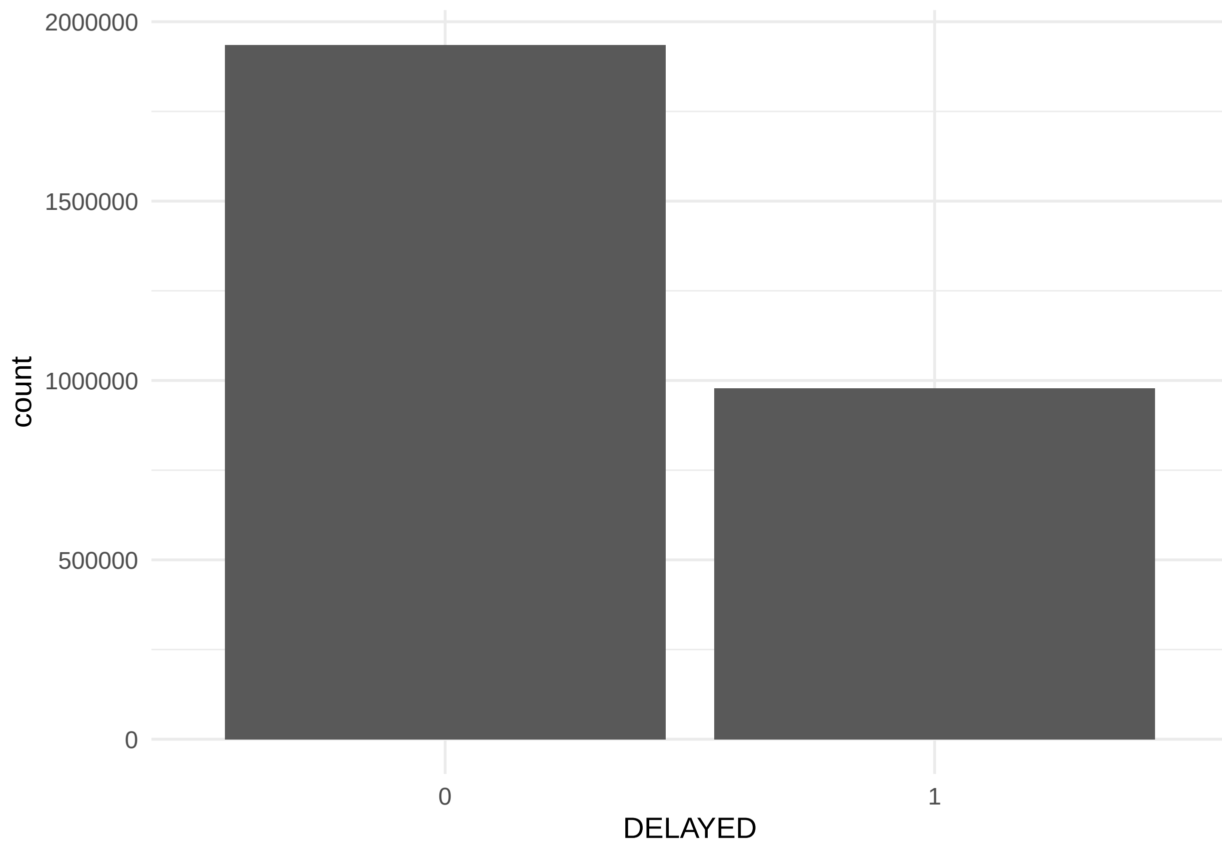



```
ggplot(flights, aes(x = DEST)) + geom_bar() + theme_minimal()
```




```r
ggplot(flights, aes(x = DIVERTED)) + geom_bar() + theme_minimal()
```


```
ggplot(flights, aes(x = DELAYED)) + geom_bar() + theme_minimal()
```




```r
# AIRLINE, ORIGIN, DEST, FL_NUMBER have too many levels
```

Some things we noticed:

1. Some of the numeric variables like CRS_ELAPSED_TIME, DEP_DELAY, and TAXI_OUT are heavily right-skewed. Logarithmic or root transformations may be required later on.
2. CRS_DEP_TIME, CRS_ARR_TIME, WHEELS_OFF, and DEP_TIME all look *close enough* to being normally distributed, so we probably won't use any transformations on them.
3. There are too many categories in FL_NUMBER, ORIGIN, DEST, and AIRLINE. We will consider only using the most popular levels in each.
4. DIVERTED essentially has no observations marked as "0". So, we will delete this variable.

Our first step was to delete DIVERTED.

```r
# Delete DIVERTED because it is heavily skewed towards "no"
flights <- flights %>% dplyr::select(-DIVERTED)
```

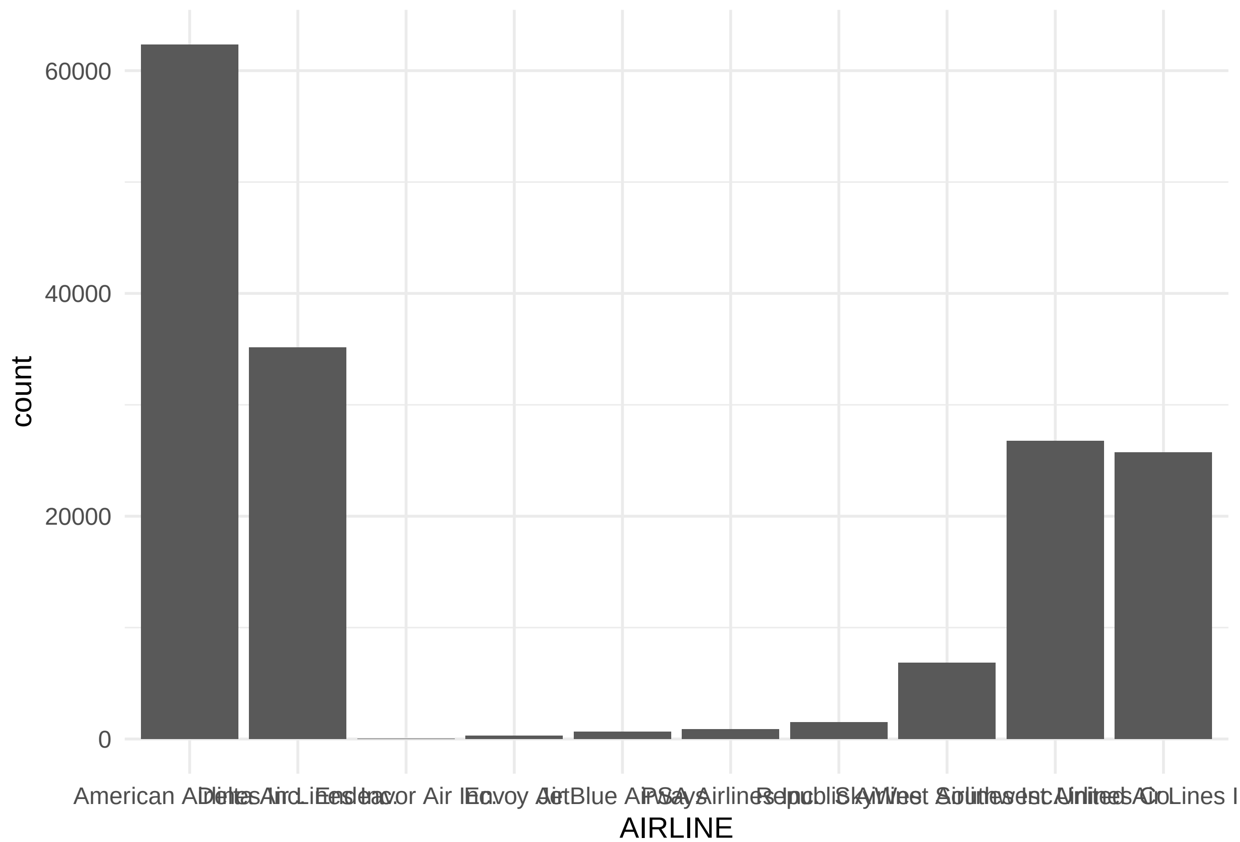
At this point, we realized that flight numbers can be thought of license plates on cars for planes, except each plane is given a unique flight number based on their route. Since we can't really group planes ID's together, we will scrap this variable.

```r
flights <- flights %>% select(-FL_NUMBER)
```

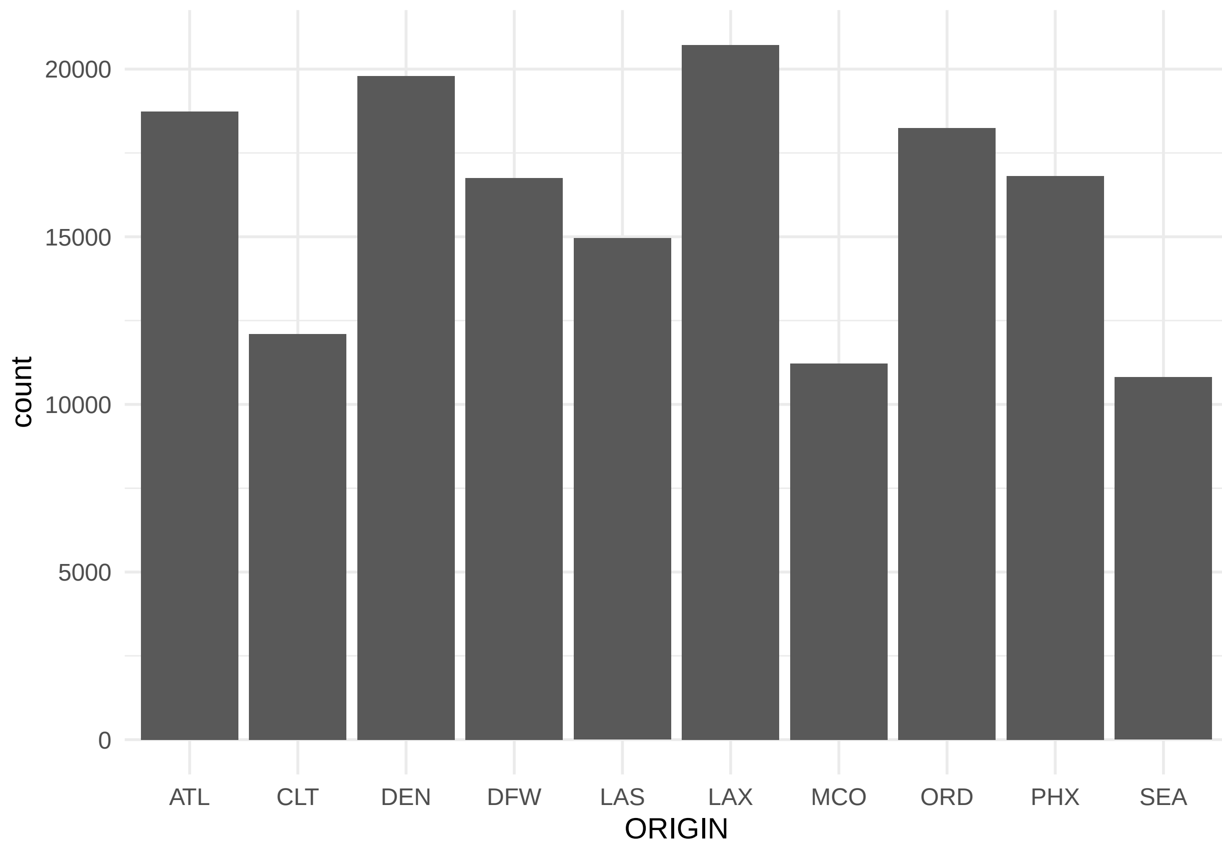
At first, we decided to only keep the top 10 popular airlines and airports. Unfortunately, it there are still a few airlines that fly a lot and a few airlines that don't fly often in comparison. To make it more even, we will take only the top 4 popular airlines, instead of 10.

```r
keep_top_10 <- function(var) {
  freq <- table(var)
```

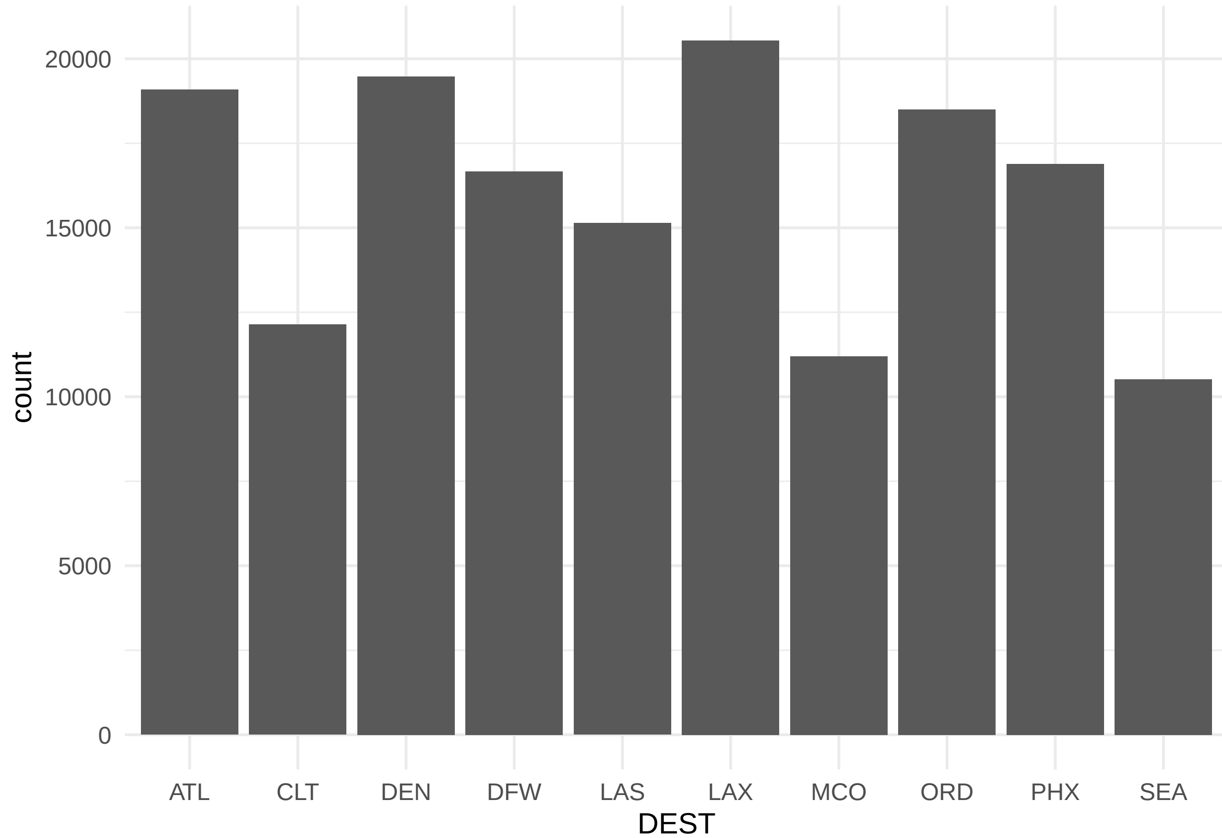
```r
  top_levels <- names(sort(freq, decreasing = TRUE)[1:10])
  as.factor(ifelse(var %in% top_levels, as.character(var), "other"))
}

flights <- flights %>%
  mutate(
    AIRLINE = keep_top_10(AIRLINE),
    ORIGIN = keep_top_10(ORIGIN),
    DEST = keep_top_10(DEST)
  ) %>%
  filter(
    AIRLINE != "other",
    DEST != "other",
    ORIGIN != "other"
  )

ggplot(flights, aes(x = AIRLINE)) + geom_bar() + theme_minimal()
```



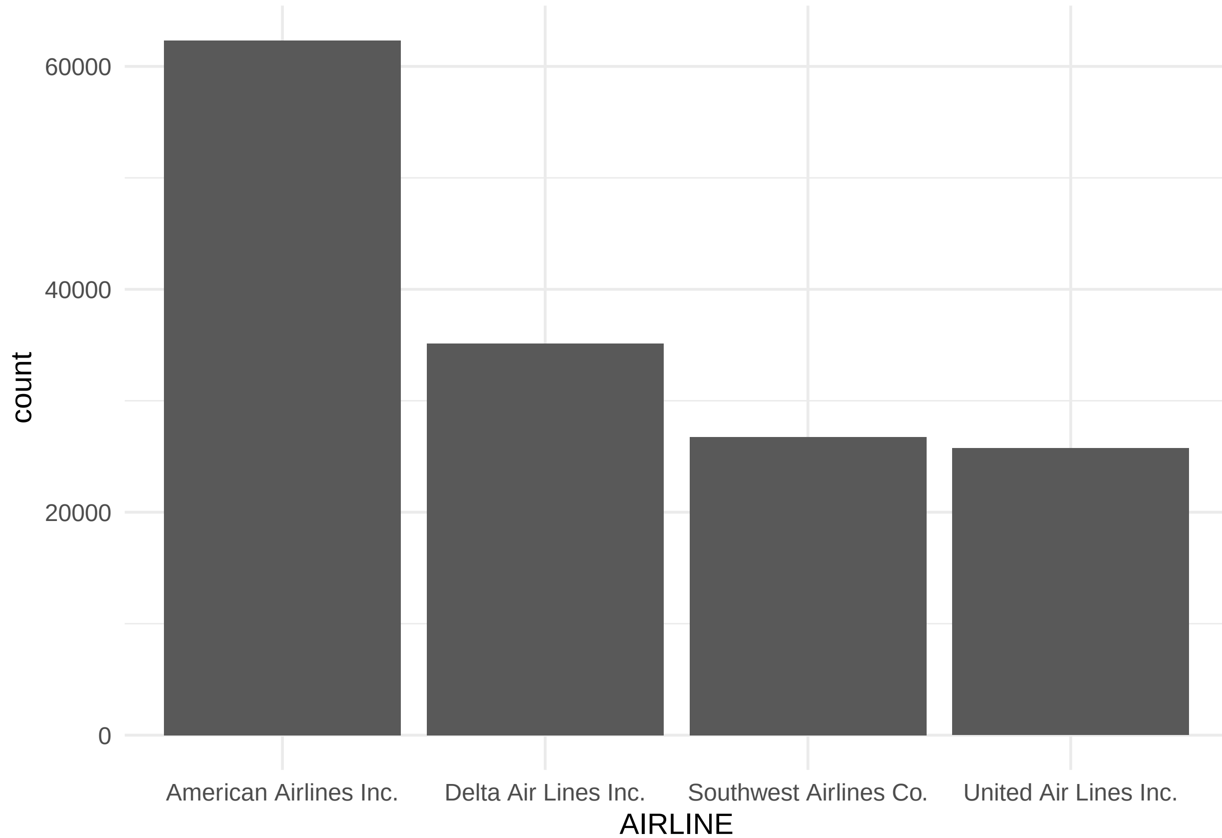

```r
ggplot(flights, aes(x = ORIGIN)) + geom_bar() + theme_minimal()
```




```
ggplot(flights, aes(x = DEST)) + geom_bar() + theme_minimal()
```




```
# AIRLINE still looks uneven --> just keep the top 4 instead


keep_top_4 <- function(var) {
  freq <- table(var)
  top_levels <- names(sort(freq, decreasing = TRUE)[1:4])
  as.factor(ifelse(var %in% top_levels, as.character(var), "other"))
}

flights <- flights %>%
  mutate(
    AIRLINE = keep_top_4(AIRLINE),
  ) %>%
  filter(
    AIRLINE != "other",
  )

ggplot(flights, aes(x = AIRLINE)) + geom_bar() + theme_minimal()
```




```
# looks much more even, includes all 3 major U.S. airlines as well
```

Now that our data is clean, we will take a random sample of 50,000 from the cleaned data. We then split this into a training/testing split, and trained the default model based on the training split.

```
dim(flights)
```

```
## [1] 149948     15
```

```
# use a small sample of dataset instead

set.seed(12345678)
index <- sample(nrow(flights), 50000)
flights_sample <- flights[index, ]


# 80/20 training/testing split

set.seed(12345678)
index2 <- createDataPartition(flights_sample$DELAYED, p = 0.8, list = FALSE)
train <- flights_sample[index2, ]
test <- flights_sample[-index2, ]

default_model <- glm(DELAYED ~ ., data = train, family = "binomial")
summary(default_model)
```

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

```
## glm(formula = DELAYED ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2618  -0.5238  -0.3024   0.0993   3.3055
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.513e+00  1.797e-01  -8.418  < 2e-16 ***
## YEAR2020                     -1.685e-01  5.253e-02  -3.207  0.00134 **
## YEAR2021                     -1.468e-01  4.656e-02  -3.152  0.00162 **
## YEAR2022                     -1.441e-01  4.562e-02  -3.158  0.00159 **
## YEAR2023                      2.864e-02  5.037e-02   0.569  0.56958
## MONTH                        -1.388e-02  4.827e-03  -2.876  0.00403 **
## DAYMonday                     5.416e-02  5.745e-02   0.943  0.34580
## DAYSaturday                  -8.484e-02  6.034e-02  -1.406  0.15974
## DAYSunday                    -2.359e-03  5.816e-02  -0.041  0.96764
## DAYThursday                   3.687e-02  5.727e-02   0.644  0.51974
## DAYTuesday                    4.514e-02  5.849e-02   0.772  0.44027
## DAYWednesday                  1.647e-01  5.820e-02   2.831  0.00465 **
## AIRLINEDelta Air Lines Inc.  -1.038e-01  6.097e-02  -1.703  0.08864 .
## AIRLINESouthwest Airlines Co. -2.764e-02  5.908e-02  -0.468  0.63992
## AIRLINEUnited Air Lines Inc.  -3.748e-01  5.645e-02  -6.638 3.17e-11 ***
## ORIGINCLT                    -1.766e-01  8.736e-02  -2.021  0.04328 *
## ORIGINDEN                    -8.370e-01  8.286e-02 -10.101  < 2e-16 ***
## ORIGINDFW                    -6.972e-01  8.409e-02  -8.292  < 2e-16 ***
## ORIGINLAS                    -1.298e+00  8.819e-02 -14.716  < 2e-16 ***
## ORIGINLAX                    -1.251e+00  8.127e-02 -15.393  < 2e-16 ***
## ORIGINMCO                     2.458e-01  8.125e-02   3.025  0.00249 **
## ORIGINORD                    -5.292e-01  7.823e-02  -6.765 1.33e-11 ***
## ORIGINPHX                    -9.485e-01  8.671e-02 -10.939  < 2e-16 ***
## ORIGINSEA                    -1.428e+00  8.837e-02 -16.156  < 2e-16 ***
## DESTCLT                       1.215e-01  8.931e-02   1.361  0.17361
## DESTDEN                       1.188e+00  7.973e-02  14.899  < 2e-16 ***
## DESTDFW                       9.850e-01  8.436e-02  11.676  < 2e-16 ***
## DESTLAS                       1.155e+00  8.309e-02  13.902  < 2e-16 ***
## DESTLAX                       1.039e+00  8.230e-02  12.625  < 2e-16 ***
## DESTMCO                       1.640e-01  8.197e-02   2.001  0.04536 *
## DESTORD                       8.573e-01  8.227e-02  10.420  < 2e-16 ***
## DESTPHX                       1.150e+00  8.337e-02  13.795  < 2e-16 ***
## DESTSEA                       1.824e+00  1.070e-01  17.039  < 2e-16 ***
## CRS_DEP_TIME                 -3.732e-04  2.128e-04  -1.754  0.07947 .
## DEP_TIME                      2.043e-04  2.544e-04   0.803  0.42181
## DEP_DELAY                     1.764e-01  2.237e-03  78.858  < 2e-16 ***
## TAXI_OUT                      1.734e-01  2.832e-03  61.218  < 2e-16 ***
## WHEELS_OFF                   -9.803e-05  1.616e-04  -0.607  0.54417
## CRS_ARR_TIME                  4.175e-04  7.279e-05   5.735 9.74e-09 ***
## CRS_ELAPSED_TIME             -6.060e-02  2.341e-03 -25.889  < 2e-16 ***
## DISTANCE                      7.139e-03  2.744e-04  26.013  < 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: 52029  on 40000  degrees of freedom
## Residual deviance: 26121  on 39960  degrees of freedom
## AIC: 26203
##
## Number of Fisher Scoring iterations: 8
```

Influential Points & Outliers

Now that the data is cleaned and split into training and testing portions, we used Cook's Distance to find our influential points in the training data. We analyzed the possible ranges of these points and determined that all of these are feasible, so we decided to keep all of these observations.

```r
# Influential points

cooks_dist <- cooks.distance(default_model)
influential_points <- which(cooks_dist > 4 / nrow(train))
influential_data <- train[influential_points, ]

summary(influential_data)
```

```
##    YEAR         MONTH               DAY                            AIRLINE
##  2019:882   Min.   : 1.000   Friday   :518   American Airlines Inc.:1564
##  2020:592   1st Qu.: 3.000   Monday   :475   Delta Air Lines Inc.  : 930
##  2021:715   Median : 6.000   Saturday :484   other                 :   0
##  2022:765   Mean   : 6.237   Sunday   :528   Southwest Airlines Co.: 477
##  2023:622   3rd Qu.: 9.000   Thursday :551   United Air Lines Inc. : 605
##             Max.   :12.000   Tuesday  :507
##                              Wednesday:513
##      ORIGIN          DEST       CRS_DEP_TIME       DEP_TIME
##  ATL    : 487   LAX    : 471   Min.   :   6.0   Min.   :   1.0
##  LAX    : 459   ORD    : 457   1st Qu.: 535.0   1st Qu.: 526.0
##  ORD    : 403   DEN    : 411   Median : 780.0   Median : 767.5
##  PHX    : 372   ATL    : 403   Mean   : 798.4   Mean   : 784.6
##  DFW    : 348   DFW    : 367   3rd Qu.:1036.2   3rd Qu.:1025.2
##  DEN    : 329   PHX    : 358   Max.   :1439.0   Max.   :1440.0
##  (Other):1178   (Other):1109
##    DEP_DELAY          TAXI_OUT       WHEELS_OFF      CRS_ARR_TIME
##  Min.   :-15.000   Min.   : 5.00   Min.   :   1.0   Min.   :   1.0
##  1st Qu.: -4.000   1st Qu.:13.00   1st Qu.: 532.0   1st Qu.: 653.8
##  Median : -1.000   Median :16.00   Median : 769.0   Median : 905.0
##  Mean   :  1.103   Mean   :17.27   Mean   : 782.5   Mean   : 886.3
##  3rd Qu.:  4.000   3rd Qu.:20.00   3rd Qu.:1028.0   3rd Qu.:1143.2
##  Max.   : 45.000   Max.   :52.00   Max.   :1440.0   Max.   :1439.0
##
##  CRS_ELAPSED_TIME    DISTANCE    DELAYED
##  Min.   : 60.0    Min.   : 226   0: 731
##  1st Qu.:145.0    1st Qu.: 802   1:2845
##  Median :185.0    Median :1199
##  Mean   :193.8    Mean   :1230
##  3rd Qu.:249.0    3rd Qu.:1744
##  Max.   :381.0    Max.   :2554
##
```

```r
influential_data %>%
  filter(DISTANCE == 2554 | DISTANCE == 226)
```

```
##    YEAR MONTH       DAY                AIRLINE ORIGIN DEST CRS_DEP_TIME
## 1  2022     7    Sunday   Delta Air Lines Inc.    MCO  SEA         1095
## 2  2021    10    Friday American Airlines Inc.    CLT  ATL         1244
## 3  2020    12 Wednesday   Delta Air Lines Inc.    CLT  ATL          756
## 4  2021     7    Friday   Delta Air Lines Inc.    ATL  CLT          652
## 5  2021     4  Saturday   Delta Air Lines Inc.    ATL  CLT          645
## 6  2022     2    Friday   Delta Air Lines Inc.    MCO  SEA         1080
## 7  2019     5 Wednesday   Delta Air Lines Inc.    SEA  MCO          530
## 8  2023     3  Thursday American Airlines Inc.    CLT  ATL         1225
## 9  2022     4    Friday American Airlines Inc.    ATL  CLT          978
## 10 2019     3    Sunday American Airlines Inc.    CLT  ATL          799
## 11 2023     5   Tuesday   Delta Air Lines Inc.    SEA  MCO         1425
## 12 2019     4  Thursday   Delta Air Lines Inc.    ATL  CLT          940
## 13 2021     3  Thursday   Delta Air Lines Inc.    CLT  ATL          775
## 14 2022    11  Thursday   Delta Air Lines Inc.    CLT  ATL         1018
## 15 2023     6 Wednesday   Delta Air Lines Inc.    SEA  MCO         1417
## 16 2019     9    Friday American Airlines Inc.    CLT  ATL         1102
## 17 2021     7 Wednesday   Delta Air Lines Inc.    ATL  CLT          750
## 18 2022     4  Thursday   Delta Air Lines Inc.    ATL  CLT         1048
## 19 2021     4  Thursday   Delta Air Lines Inc.    ATL  CLT          894
## 20 2023     8  Thursday   Delta Air Lines Inc.    SEA  MCO         1425
## 21 2020    12  Thursday   Delta Air Lines Inc.    ATL  CLT          625
## 22 2019     8   Tuesday   Delta Air Lines Inc.    ATL  CLT          843
## 23 2022    11 Wednesday American Airlines Inc.    CLT  ATL         1218
## 24 2022     2    Friday   Delta Air Lines Inc.    ATL  CLT         1203
## 25 2022    10    Sunday   Delta Air Lines Inc.    CLT  ATL          723
## 26 2019     4   Tuesday   Delta Air Lines Inc.    CLT  ATL         1145
## 27 2023     4 Wednesday American Airlines Inc.    ATL  CLT          520
## 28 2020     8 Wednesday American Airlines Inc.    CLT  ATL          904
## 29 2021     6  Thursday   Delta Air Lines Inc.    SEA  MCO          500
## 30 2023     2   Tuesday   Delta Air Lines Inc.    ATL  CLT          748
## 31 2022    10    Friday American Airlines Inc.    ATL  CLT         1222
## 32 2019    11    Sunday American Airlines Inc.    ATL  CLT          427
## 33 2019     3    Monday American Airlines Inc.    ATL  CLT          661
## 34 2021     8  Saturday   Delta Air Lines Inc.    CLT  ATL          360
## 35 2023     1 Wednesday   Delta Air Lines Inc.    CLT  ATL         1045
## 36 2023     3    Sunday   Delta Air Lines Inc.    SEA  MCO          492
## 37 2022     6  Thursday   Delta Air Lines Inc.    CLT  ATL         1130
## 38 2021     9 Wednesday American Airlines Inc.    CLT  ATL         1105
## 39 2021     5    Friday   Delta Air Lines Inc.    SEA  MCO         1390
## 40 2019    10  Saturday American Airlines Inc.    ATL  CLT          743
## 41 2019     2  Thursday American Airlines Inc.    CLT  ATL          910
## 42 2021     6   Tuesday   Delta Air Lines Inc.    SEA  MCO          515
## 43 2019    12    Sunday American Airlines Inc.    ATL  CLT          737
## 44 2023     6 Wednesday   Delta Air Lines Inc.    CLT  ATL          420
## 45 2023     1    Friday   Delta Air Lines Inc.    SEA  MCO          480
## 46 2022     6  Saturday American Airlines Inc.    CLT  ATL         1240
## 47 2021    12    Monday   Delta Air Lines Inc.    ATL  CLT          655
## 48 2021     1    Friday   Delta Air Lines Inc.    SEA  MCO          480
## 49 2020     3   Tuesday   Delta Air Lines Inc.    CLT  ATL          641
## 50 2022     9  Saturday   Delta Air Lines Inc.    SEA  MCO         1313
## 51 2019     1   Tuesday American Airlines Inc.    ATL  CLT          746
```

```
## 52 2019     8  Saturday   Delta Air Lines Inc.    CLT  ATL          1070
## 53 2023     7    Sunday   Delta Air Lines Inc.    CLT  ATL          1133
## 54 2022     4   Tuesday American Airlines Inc.    CLT  ATL          1241
## 55 2023     8    Sunday   Delta Air Lines Inc.    SEA  MCO           450
## 56 2022    11   Tuesday   Delta Air Lines Inc.    SEA  MCO          1420
## 57 2022    10 Wednesday American Airlines Inc.    ATL  CLT           532
## 58 2023     5    Friday   Delta Air Lines Inc.    SEA  MCO           515
## 59 2019     2    Monday American Airlines Inc.    CLT  ATL           565
## 60 2020    12 Wednesday   Delta Air Lines Inc.    ATL  CLT           735
## 61 2022    10    Monday   Delta Air Lines Inc.    ATL  CLT           597
## 62 2023     3    Friday   Delta Air Lines Inc.    CLT  ATL           453
## 63 2023     5  Saturday American Airlines Inc.    ATL  CLT          1210
## 64 2021    11   Tuesday American Airlines Inc.    ATL  CLT          1221
## 65 2022    12   Tuesday American Airlines Inc.    CLT  ATL          1215
## 66 2019     9    Sunday   Delta Air Lines Inc.    ATL  CLT           530
## 67 2019    12    Monday   Delta Air Lines Inc.    ATL  CLT          1027
## 68 2023     6    Friday American Airlines Inc.    ATL  CLT           395
## 69 2021     5  Saturday   Delta Air Lines Inc.    ATL  CLT           645
## 70 2023     5   Tuesday American Airlines Inc.    CLT  ATL          1102
## 71 2022    11    Monday American Airlines Inc.    CLT  ATL           460
## 72 2019    12   Tuesday American Airlines Inc.    ATL  CLT           738
## 73 2020     8    Monday   Delta Air Lines Inc.    CLT  ATL           460
##    DEP_TIME DEP_DELAY TAXI_OUT WHEELS_OFF CRS_ARR_TIME CRS_ELAPSED_TIME
## 1      1118        23       11       1129         1279              364
## 2      1242        -2       13       1255         1320               76
## 3       751        -5       24        775          829               73
## 4       646        -6       15        661          721               69
## 5       639        -6       15        654          715               70
## 6      1073        -7       18       1091         1281              381
## 7       524        -6       28        552         1046              336
## 8      1219        -6       26       1245         1298               73
## 9       972        -6       16        988         1046               68
## 10      795        -4       27        822          880               81
## 11     1421        -4       14       1435          496              331
## 12      937        -3       20        957         1012               72
## 13      784         9       13        797          851               76
## 14     1012        -6       25       1037         1097               79
## 15     1417         0       15       1432          486              329
## 16     1111         9       15       1126         1184               82
## 17      746        -4       23        769          822               72
## 18     1046        -2       16       1062         1118               70
## 19      894         0       11        905          967               73
## 20     1434         9       15          9          499              334
## 21      623        -2       19        642          691               66
## 22      843         0       16        859          921               78
## 23     1213        -5       16       1229         1290               72
## 24     1207         4       10       1217         1275               72
## 25      749        26       10        759          804               81
## 26     1143        -2       23       1166         1230               85
## 27      513        -7       22        535          596               76
## 28      911         7       10        921          978               74
## 29      500         0       35        535         1023              343
## 30      746        -2       17        763          816               68
## 31     1220        -2       25       1245         1298               76
```

```
## 32      421       -6        32       453        510              83
## 33      666        5        16       682        742              81
## 34      357       -3        22       379        427              67
## 35     1066       21        10      1076       1127              82
## 36      488       -4        30       518       1000             328
## 37     1127       -3        36      1163       1210              80
## 38     1095      -10        24      1119       1178              73
## 39     1384       -6        14      1398        453             323
## 40      742       -1        25       767        825              82
## 41      927       17        14       941        990              80
## 42      515        0        28       543       1034             339
## 43      734       -3        24       758        815              78
## 44      418       -2        20       438        489              69
## 45      482        2        32       514        988             328
## 46     1243        3        18      1261       1313              73
## 47      657        2        16       673        724              69
## 48      479       -1        28       507        983             323
## 49      636       -5        18       654        716              75
## 50     1338       25        10      1348        386             333
## 51      744       -2        21       765        823              77
## 52     1070        0        19      1089       1153              83
## 53     1138        5        32      1170       1219              86
## 54     1238       -3        17      1255       1317              76
## 55      454        4        31       485        976             346
## 56     1417       -3        22      1439        469             309
## 57      527       -5        19       546        609              77
## 58      510       -5        20       530       1036             341
## 59      564       -1        19       583        640              75
## 60      732       -3        14       746        796              61
## 61      593       -4        21       614        668              71
## 62      452       -1        24       476        536              83
## 63     1208       -2        20      1228       1288              78
## 64     1220       -1        13      1233       1310              89
## 65     1210       -5        24      1234       1286              71
## 66      527       -3        24       551        609              79
## 67     1028        1        19      1047       1100              73
## 68      390       -5        18       408        473              78
## 69      642       -3        24       666        715              70
## 70     1097       -5        17      1114       1170              68
## 71      453       -7        17       470        528              68
## 72      742        4        14       756        814              76
## 73      456       -4        17       473        520              60
##    DISTANCE DELAYED
## 1      2554       0
## 2       226       1
## 3       226       1
## 4       226       1
## 5       226       1
## 6      2554       1
## 7      2554       1
## 8       226       1
## 9       226       1
## 10      226       1
## 11     2554       1
```

```
## 12      226       1
## 13      226       1
## 14      226       1
## 15     2554       1
## 16      226       1
## 17      226       1
## 18      226       1
## 19      226       1
## 20     2554       0
## 21      226       1
## 22      226       1
## 23      226       1
## 24      226       1
## 25      226       0
## 26      226       1
## 27      226       1
## 28      226       1
## 29     2554       0
## 30      226       1
## 31      226       1
## 32      226       1
## 33      226       1
## 34      226       1
## 35      226       0
## 36     2554       0
## 37      226       0
## 38      226       1
## 39     2554       1
## 40      226       1
## 41      226       0
## 42     2554       1
## 43      226       1
## 44      226       1
## 45     2554       0
## 46      226       1
## 47      226       1
## 48     2554       0
## 49      226       1
## 50     2554       0
## 51      226       1
## 52      226       1
## 53      226       0
## 54      226       1
## 55     2554       0
## 56     2554       1
## 57      226       1
## 58     2554       1
## 59      226       1
## 60      226       1
## 61      226       1
## 62      226       1
## 63      226       1
## 64      226       1
## 65      226       1
```

```
## 66      226       1
## 67      226       1
## 68      226       1
## 69      226       1
## 70      226       1
## 71      226       1
## 72      226       1
## 73      226       1
```

```
# Long flights are between SEA and MCO  and short flights are between ATL and CLT

influential_data %>%
  filter(CRS_DEP_TIME > 1440 | DEP_TIME > 1440, WHEELS_OFF > 1440 |
           CRS_ARR_TIME > 1440 | CRS_ELAPSED_TIME > 1440)
```

```
##  [1] YEAR             MONTH            DAY              AIRLINE
##  [5] ORIGIN           DEST             CRS_DEP_TIME     DEP_TIME
##  [9] DEP_DELAY        TAXI_OUT         WHEELS_OFF       CRS_ARR_TIME
## [13] CRS_ELAPSED_TIME DISTANCE         DELAYED
## <0 rows> (or 0-length row.names)
```

```
# None of the times are past 1440 minutes past midnight

# Everything looks good!
```

Our next step was to check for outliers. We decided to use the IQR method of creating the outlier bounds using Q1 - 1.5(IQR) and Q3 + 1.5(IQR). Luckily for us, we did not have any outliers in the training data.

```
# Outliers

train_num <- train[, sapply(train, is.numeric)]
outliers <- train_num %>%
  mutate(row_id = row_number()) %>%
  rowwise() %>%
  mutate(outlier = any(across(everything(), ~ {
    Q1 <- quantile(., 0.25, na.rm = TRUE)
    Q3 <- quantile(., 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    . < (Q1 - 1.5 * IQR) | . > (Q3 + 1.5 * IQR)
  }))) %>%
  ungroup() %>%
  filter(outlier) %>%
  pull(row_id)

outliers
```

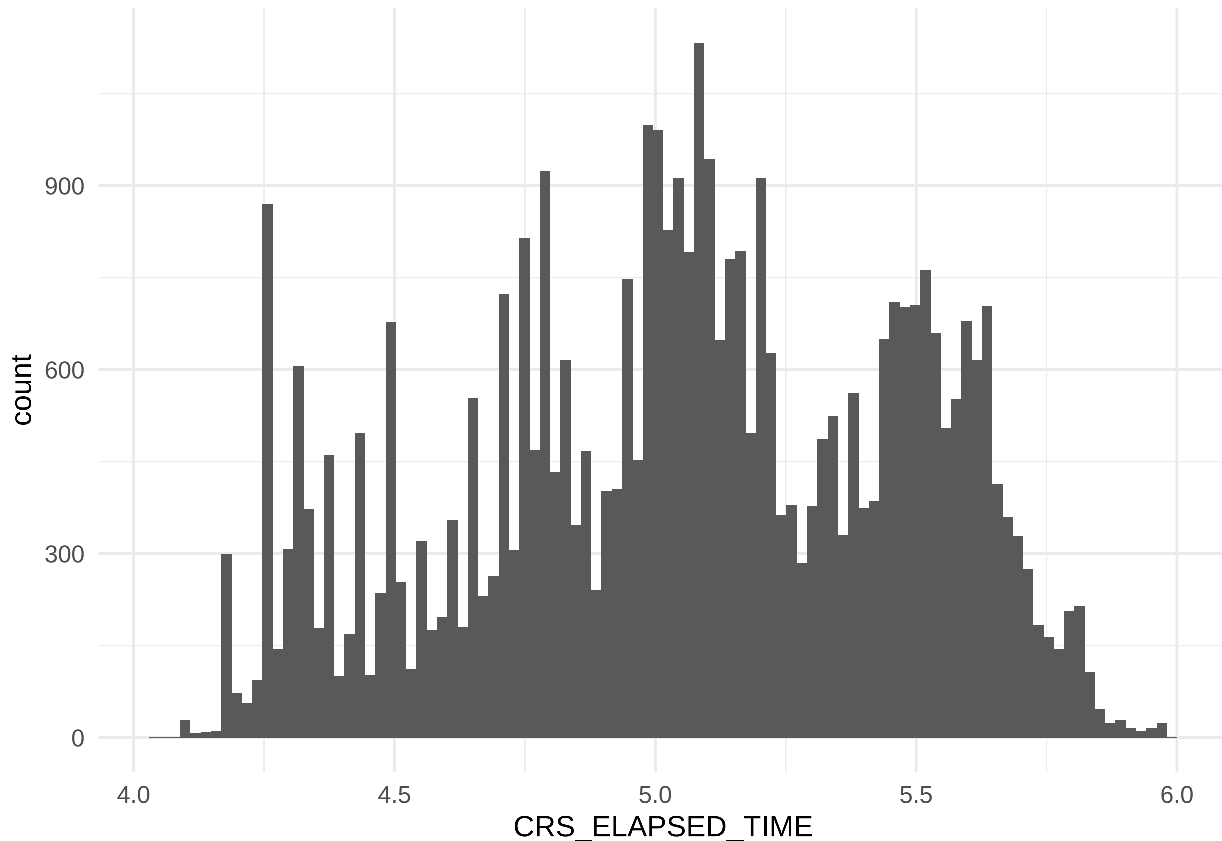
```
## integer(0)
```

```
# No outliers!
```

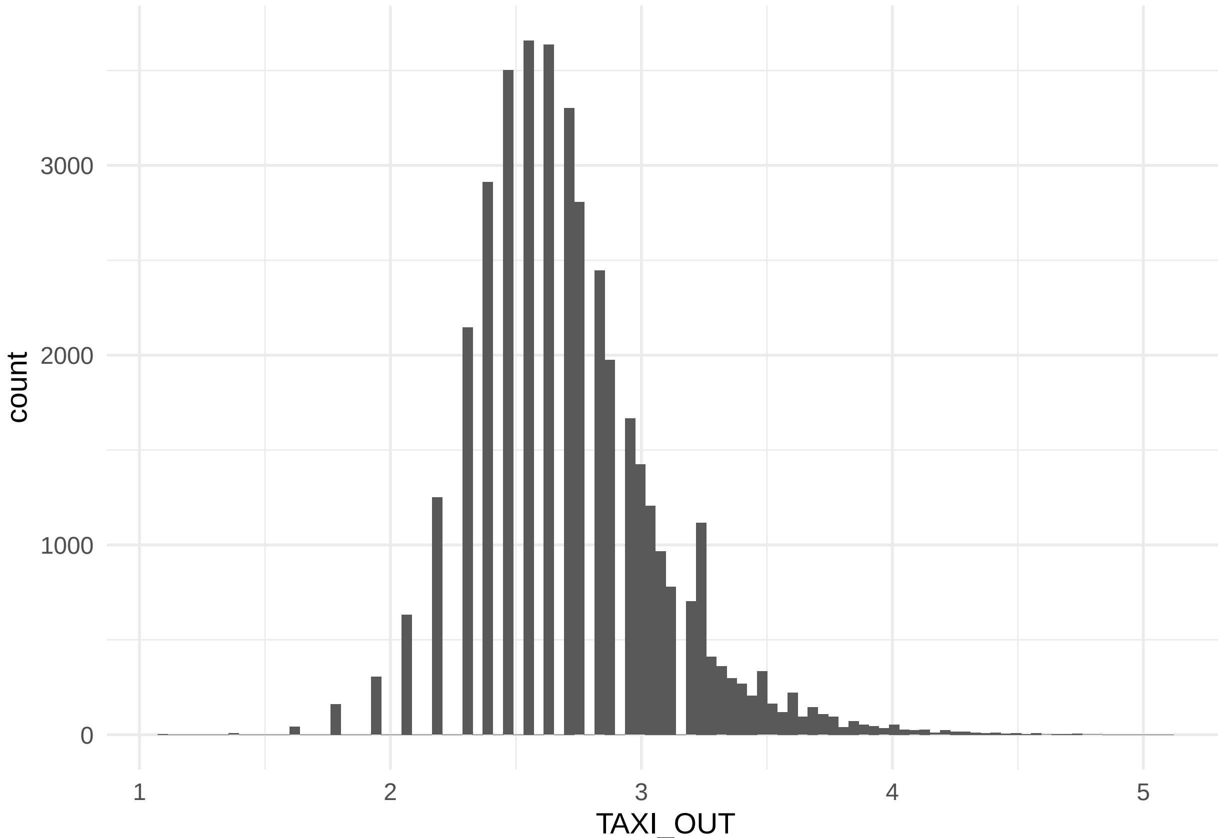
Variable Transformation

From the EDA, we saw that some of our numeric variables require transformations. CRS_ELAPSED_TIME and TAXI_OUT are skewed right and consist of positive values, so we applied a logarithmic transformation to them. DEP_DELAY contains negative numbers, so we applied a cube root transformation instead.

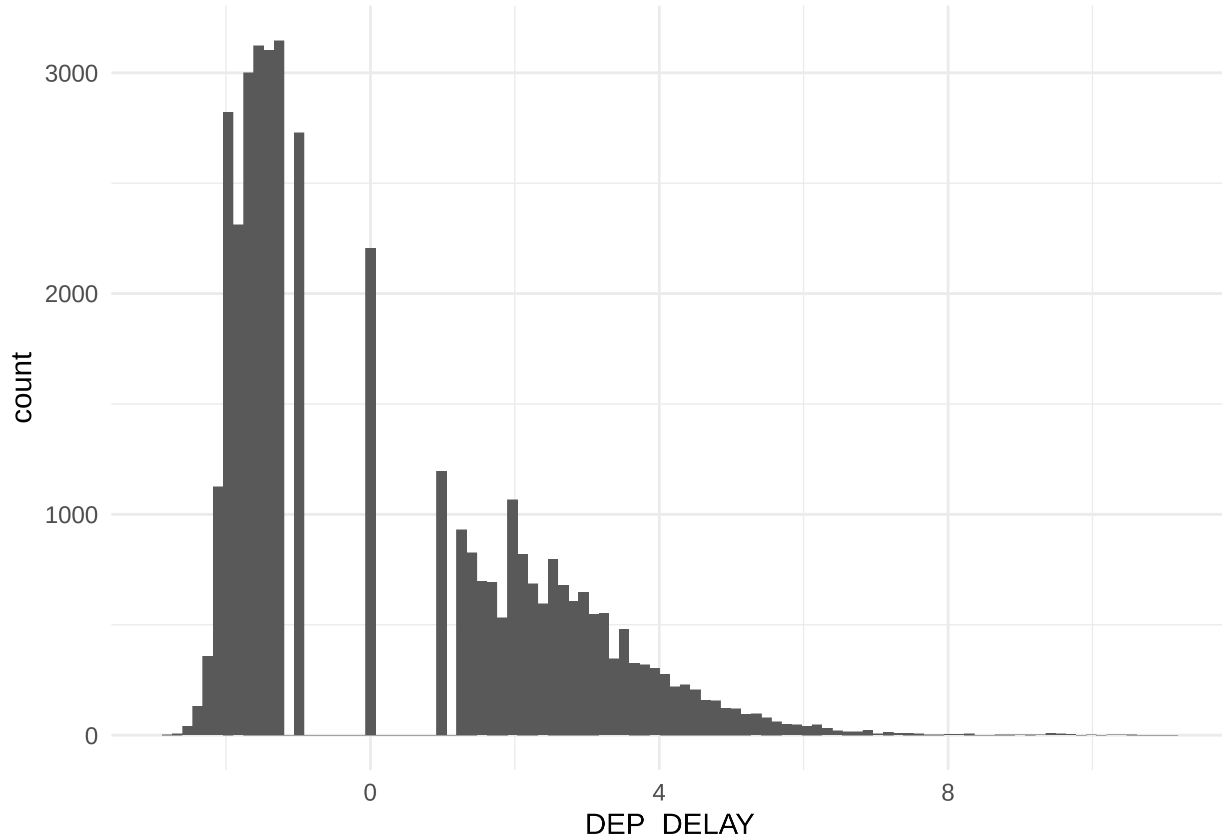
```r
train_new <- train %>%
  # Right skew > 0 -> log()
  mutate(
    CRS_ELAPSED_TIME = log(CRS_ELAPSED_TIME),
    TAXI_OUT = log(TAXI_OUT),
    DISTANCE = log(DISTANCE),
  ) %>%
  # Right-skew with negative #'s -> sign(x) * abs(x)^(1/3)
  mutate(
    DEP_DELAY = sign(DEP_DELAY) * abs(DEP_DELAY)^(1/3)
  )


ggplot(train_new, aes(x = CRS_ELAPSED_TIME)) + geom_histogram(bins = 100) + theme_minimal()
```



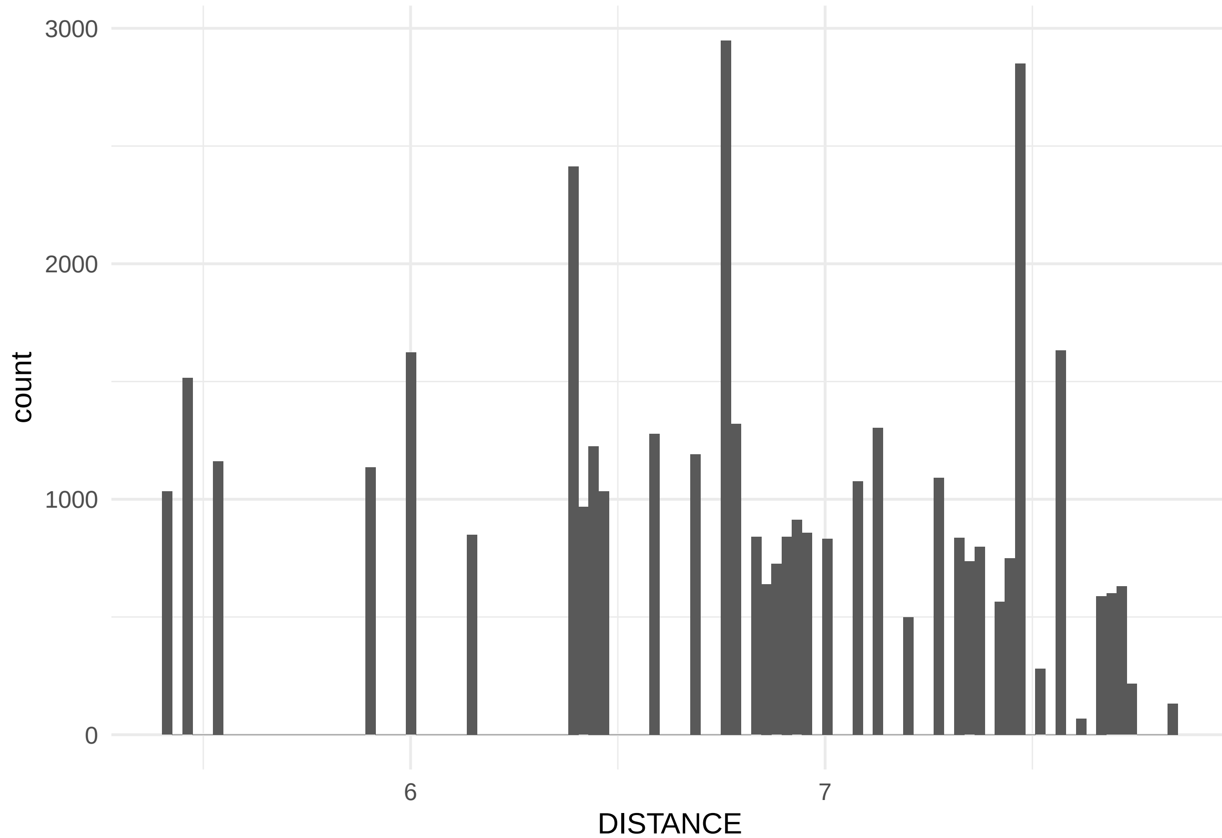

```r
ggplot(train_new, aes(x = TAXI_OUT)) + geom_histogram(bins = 100) + theme_minimal()
```




```
ggplot(train_new, aes(x = DEP_DELAY)) + geom_histogram(bins = 100) + theme_minimal()
```




```
ggplot(train_new, aes(x = DISTANCE)) + geom_histogram(bins = 100) + theme_minimal()
```




```
# these look closer to normal now
# DEP_DELAY is still a little right skewed, but better than before
```

Variable Selection

For our variable selection process, we decided to use stepwise selection going both directions, and BIC as our selection criterion. Since we are making an explanatory model, we decided that because BIC is stricter on the number of predictors, our model will have a simpler model that will be easier to explain. Our selected variables were DEP_DELAY, TAXI_OUT, ORIGIN, AIRLINE, WHEELS_OFF, and DEST.

```
model <- glm(DELAYED ~ ., data = train_new, family = "binomial")

bic_model <- step(glm(DELAYED ~ 1, family="binomial", data=train_new), scope = formula(model),
                  direction = "both", trace = 0, k = log(nrow(train_new)))

summary(bic_model)
```

```
##
## Call:
## glm(formula = DELAYED ~ DEP_DELAY + TAXI_OUT + ORIGIN + AIRLINE +
##     WHEELS_OFF + DEST, family = "binomial", data = train_new)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5134  -0.5362  -0.2958   0.4938   3.1219
##
```

```
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                  -9.230e+00  1.725e-01 -53.521  < 2e-16 ***
## DEP_DELAY                     8.477e-01  8.538e-03  99.289  < 2e-16 ***
## TAXI_OUT                      2.809e+00  4.696e-02  59.822  < 2e-16 ***
## ORIGINCLT                    -2.772e-01  8.011e-02  -3.461 0.000539 ***
## ORIGINDEN                     1.149e-01  6.881e-02   1.670 0.094947 .  
## ORIGINDFW                    -1.874e-01  7.479e-02  -2.506 0.012212 *  
## ORIGINLAS                    -1.132e-01  7.135e-02  -1.587 0.112492    
## ORIGINLAX                    -1.017e-01  6.494e-02  -1.566 0.117455    
## ORIGINMCO                     1.429e-01  7.589e-02   1.883 0.059662 .  
## ORIGINORD                    -3.628e-01  7.129e-02  -5.088 3.61e-07 ***
## ORIGINPHX                     3.108e-01  6.974e-02   4.457 8.32e-06 ***
## ORIGINSEA                    -3.371e-01  7.392e-02  -4.561 5.10e-06 ***
## AIRLINEDelta Air Lines Inc.   1.161e-01  5.553e-02   2.090 0.036596 *  
## AIRLINESouthwest Airlines Co.  1.945e-01  5.278e-02   3.684 0.000230 ***
## AIRLINEUnited Air Lines Inc. -2.595e-01  5.167e-02  -5.022 5.12e-07 ***
## WHEELS_OFF                    2.805e-04  4.971e-05   5.643 1.67e-08 ***
## DESTCLT                       1.204e-01  8.133e-02   1.480 0.138780    
## DESTDEN                       5.290e-01  6.958e-02   7.603 2.89e-14 ***
## DESTDFW                       3.884e-01  7.419e-02   5.236 1.65e-07 ***
## DESTLAS                       3.751e-01  7.026e-02   5.339 9.32e-08 ***
## DESTLAX                       1.399e-02  6.548e-02   0.214 0.830784    
## DESTMCO                       3.282e-01  7.484e-02   4.386 1.16e-05 ***
## DESTORD                       2.714e-01  7.216e-02   3.761 0.000170 ***
## DESTPHX                       3.340e-01  7.002e-02   4.770 1.84e-06 ***
## DESTSEA                       1.593e-01  7.381e-02   2.158 0.030892 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 52029  on 40000  degrees of freedom
## Residual deviance: 30377  on 39976  degrees of freedom
## AIC: 30427
##
## Number of Fisher Scoring iterations: 5
```

Regularization

Our BIC model doesn't appear to be heavily affected by multicollinearity. However, alleviate the effects of overfitting, we still used Ridge Regression. We used the default 10-fold cross validation to find the optimal lambda value for ridge regression, and then we applied ridge regression to the variables selected from the stepwise selection process. Our ridge regression model is our final model.

```
vif(bic_model)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## DEP_DELAY  1.276994  1        1.130042
## TAXI_OUT   1.387168  1        1.177781
## ORIGIN     3.429609  9        1.070868
## AIRLINE    4.702428  3        1.294356
## WHEELS_OFF 1.078845  1        1.038675
## DEST       2.984716  9        1.062633
```

```r
X <- model.matrix(~ DEP_DELAY + TAXI_OUT + ORIGIN + AIRLINE + DEST + WHEELS_OFF,
                  data = train_new)[, -1]

y <- as.numeric(as.character(train_new$DELAYED))

# Ridge Regression
cv_ridge <- cv.glmnet(X, y, alpha = 0, family = "binomial")
lambda_ridge <- cv_ridge$lambda.min
ridge_model <- glmnet(X, y, alpha = 0, family = "binomial", lambda = lambda_ridge)


ridge_coefficients <- coef(cv_ridge, s = "lambda.min")
coef_matrix <- as.matrix(ridge_coefficients)
ridge_variables <- rownames(coef_matrix)[coef_matrix != 0]
ridge_variables <- ridge_variables[ridge_variables != "(Intercept)"]
ridge_variables
```

```
##  [1] "DEP_DELAY"                       "TAXI_OUT"
##  [3] "ORIGINCLT"                       "ORIGINDEN"
##  [5] "ORIGINDFW"                       "ORIGINLAS"
##  [7] "ORIGINLAX"                       "ORIGINMCO"
##  [9] "ORIGINORD"                       "ORIGINPHX"
## [11] "ORIGINSEA"                       "AIRLINEDelta Air Lines Inc."
## [13] "AIRLINESouthwest Airlines Co." "AIRLINEUnited Air Lines Inc."
## [15] "DESTCLT"                         "DESTDEN"
## [17] "DESTDFW"                         "DESTLAS"
## [19] "DESTLAX"                         "DESTMCO"
## [21] "DESTORD"                         "DESTPHX"
## [23] "DESTSEA"                         "WHEELS_OFF"
```

```r
# made up of DEP_DELAY, TAXI_OUT, ORIGIN, DEST, WHEELS_OFF, and AIRLINE
```

Comparisons Between Default and Final Models

```r
# First need to transform testing data to match training data
test_new <- test %>%
  # Right skew > 0 -> log()
  mutate(
    TAXI_OUT = log(TAXI_OUT),
  ) %>%
  # Right-skew with negative #'s -> sign(x) * abs(x)^(1/3)
  mutate(
    DEP_DELAY = sign(DEP_DELAY) * abs(DEP_DELAY)^(1/3)
  ) %>%
  select(
    c(DELAYED, DEP_DELAY, TAXI_OUT, ORIGIN, DEST, WHEELS_OFF, AIRLINE)
  )

X_ridge_test <- model.matrix(~ DEP_DELAY + TAXI_OUT + ORIGIN + AIRLINE + DEST + WHEELS_OFF,
                             data = test_new)[, -1]

X_ridge_train <- model.matrix(~ DEP_DELAY + TAXI_OUT + ORIGIN + AIRLINE + DEST + WHEELS_OFF,
                              data = train_new)[, -1]
```

```r
# default model predicting the testing data
pred_default_test <- predict(default_model, test, type = "response")
confusionMatrix(as.factor(ifelse(pred_default_test > 0.5, 1, 0)),
                                       as.factor(test$DELAYED))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6098  994
##          1  354 2553
##
##                Accuracy : 0.8652
##                  95% CI : (0.8583, 0.8718)
##     No Information Rate : 0.6453
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.693
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9451
##             Specificity : 0.7198
##          Pos Pred Value : 0.8598
##          Neg Pred Value : 0.8782
##              Prevalence : 0.6453
##          Detection Rate : 0.6099
##    Detection Prevalence : 0.7093
##       Balanced Accuracy : 0.8324
##
##        'Positive' Class : 0
##
```

```r
# ridge model predicting the testing split
pred_ridge_test <- predict(ridge_model, X_ridge_test, type = "response")
confusionMatrix(as.factor(ifelse(pred_ridge_test > 0.5, 1, 0)),
                                     as.factor(test_new$DELAYED))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5882 1030
##          1  570 2517
##
##                Accuracy : 0.84
##                  95% CI : (0.8326, 0.8471)
##     No Information Rate : 0.6453
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.64
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
##             Sensitivity : 0.9117
##             Specificity : 0.7096
##          Pos Pred Value : 0.8510
##          Neg Pred Value : 0.8154
##              Prevalence : 0.6453
##          Detection Rate : 0.5883
##    Detection Prevalence : 0.6913
##       Balanced Accuracy : 0.8106
##
##        'Positive' Class : 0
##
```

```
# default model predicting the training split
pred_default_train <- predict(default_model, train, type = "response")
confusionMatrix(as.factor(ifelse(pred_default_train > 0.5, 1, 0)),
                                       as.factor(train$DELAYED))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 24384  3865
##          1  1426 10326
##
##                Accuracy : 0.8677
##                  95% CI : (0.8644, 0.871)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6995
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9448
##             Specificity : 0.7276
##          Pos Pred Value : 0.8632
##          Neg Pred Value : 0.8787
##              Prevalence : 0.6452
##          Detection Rate : 0.6096
##    Detection Prevalence : 0.7062
##       Balanced Accuracy : 0.8362
##
##        'Positive' Class : 0
##
```

```
# ridge model predicting the training split
pred_ridge_train <- predict(ridge_model, X_ridge_train, type = "response")
confusionMatrix(as.factor(ifelse(pred_ridge_train > 0.5, 1, 0)),
                                     as.factor(train_new$DELAYED))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 23439  4062
##          1  2371 10129
```

```
##
##                Accuracy : 0.8392
##                  95% CI : (0.8355, 0.8428)
##     No Information Rate : 0.6452
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.639
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9081
##             Specificity : 0.7138
##          Pos Pred Value : 0.8523
##          Neg Pred Value : 0.8103
##              Prevalence : 0.6452
##          Detection Rate : 0.5860
##    Detection Prevalence : 0.6875
##       Balanced Accuracy : 0.8109
##
##        'Positive' Class : 0
##
```

```
c(length(coef(default_model)), length(coef(ridge_model)))
```

```
## [1] 41 28
```

Ultimately, our final model's accuracy is lower than the default model on both the training and testing data. However, we believe our final model is easier to interpret as it only has 27 coefficients compared to the default model's 40, while only suffering a 0.0252 decrease in accuracy.

Testing data prediction comparisons (Default vs. Final):

- **Accuracy**: 0.8652 vs. 0.84
- **Sensitivity**: 0.9451 vs. 0.9117
- **Specificity**: 0.7198 vs. 0.7096
- **Prevalence**: 0.6453 vs. 0.6453

Training data prediction comparisons (Default vs. Final):

- **Accuracy**: 0.8677 vs. 0.8392
- **Sensitivity**: 0.9448 vs. 0.9081
- **Specificity**: 0.7276 vs. 0.7138
- **Prevalence**: 0.6452 vs. 0.6452

What can we take away from the final model?

```
ridge_coefficients <- coef(ridge_model)
ridge_coeff_matrix <- as.matrix(ridge_coefficients)
formatted_coeff <- format(ridge_coeff_matrix, digits = 10, scientific = FALSE)

print(formatted_coeff)
```

```
##                                  s0
## (Intercept)                      "-6.3430838886554"
## DEP_DELAY                        " 0.6143755862040"
## TAXI_OUT                         " 1.8639212933082"
```

```
## ORIGINCLT                         "-0.1589958274203"
## ORIGINDEN                         " 0.0896482564605"
## ORIGINDFW                         "-0.0902584878751"
## ORIGINLAS                         "-0.0785097521077"
## ORIGINLAX                         "-0.0752749348200"
## ORIGINMCO                         " 0.0505855128161"
## ORIGINORD                         "-0.1958561390079"
## ORIGINother                       " 0.0000000000000"
## ORIGINPHX                         " 0.1706939263593"
## ORIGINSEA                         "-0.1829401449899"
## AIRLINEDelta Air Lines Inc.       " 0.0006767465172"
## AIRLINEother                      " 0.0000000000000"
## AIRLINESouthwest Airlines Co.     " 0.1693151865436"
## AIRLINEUnited Air Lines Inc.      "-0.1757705867800"
## DESTCLT                           "-0.0518126690704"
## DESTDEN                           " 0.2056410683805"
## DESTDFW                           " 0.1328973941692"
## DESTLAS                           " 0.1357389440032"
## DESTLAX                           "-0.0891056993291"
## DESTMCO                           " 0.1189327317487"
## DESTORD                           " 0.0513680161926"
## DESTother                         " 0.0000000000000"
## DESTPHX                           " 0.0907569479593"
## DESTSEA                           " 0.0088708094930"
## WHEELS_OFF                        " 0.0003550489789"
```

```
# coefficients represent log odds
```

For the ORIGIN variable, the reference level is ATL.

- If you are departing from the following airports: {DEN, MCO, PHX} then the odds of having a delayed arrival are higher compared to departing from ATL. The origin airport that is attributed with the highest odds of a late arrival is PHX.
- If you are departing from the following airports: {CLT, DFW, LAS, LAX, ORD, SEA} then the odds of having a delayed arrival are lower compared to departing from ATL. The origin airport that is attributed with the lowest odds of a late arrival is ORD.

For the DEST variable, the reference level is ATL.

- If you are arriving at the following airports: {DEN, DFW, LAS, MCO, ORD, PHX, SEA} then the odds of having a delayed arrival are higher compared to arriving at ATL. The destination airport that is attributed with the highest odds of a late arrival is DEN.
- If you are arriving at the following airports: {CLT, LAX} then the odds of having a delayed arrival are lower compared to arriving at ATL. The destination airport that is attributed with the lowest odds of a late arrival is LAX.

Overall, it looks like the two best airports that contribute to an on-time arrival schedule are Los Angeles's LAX and Charlotte's CLT. On the other hand, it also seems the worst airports that contribute to a late arrival schedule are Denver's DEN, Orlando's MCO, and Phoenix's PHX.

For the AIRLINE variable, the reference level is American Airlines.

- If you are flying with Southwest Airlines, then the odds of having a late arrival are higher compared to flying with American Airlines.
- Delta Airlines also has higher odds of a late arrival compared to American Airlines, but because the coefficient of 0.00067 is effectively 0, the change in odds between the two are minimal.

- United Airlines has the lowest odds of a late arrival compared to the other three airlines.

Now on to the numeric predictors.

- Since TAXI_OUT has the logarithmic transformation applied to it, for a 1-unit increase in log(TAXI_OUT), the log odds of a delayed arrival increases by 1.8639. If we think of it in normal odds, a 1-unit increase in log(TAXI_OUT) multiplies the odds of a delayed arrival by $e^{1.8639} = 6.4488$.
- Since DEP_DELAY has the cube root transformation applied to it, for a 1-unit increase in cube root of DEP_DELAY, the log odds of delayed arrived increases 0.6144, multiplies the odds of a delayed arrival by $e^{0.6144} = 1.8485$.
- WHEELS_OFF has no transformation and the coefficient 0.000355 is very close to 0. It has a positive, but mostly negligible effect on the odds.

Domain Insight: The most common causes for flight disruptions are bad weather, air traffic control issues, mechanical problems, and crew availability. Does this match our findings? We would argue it does. If certain airports have lousy air traffic control operations, then it would make sense that it might take longer for planes to depart and arrive at these airports. The taxi process also falls under the air traffic control at each airport, and TAXI_OUT has the largest positive magnitude out of all of our coefficients. Our interpretations about ORIGIN, DEST, and TAXI_OUT are consistent with what experts in the commercial aviation field say.