Homework 12

Prof. Cannon

Due MONDAY 12/2/24

> **NOTE: THIS ASSIGNMENT IS DUE ON *MONDAY* 12/2, at 11:59pm, NOT TUESDAY**

> **NOTE: YOU MAY ONLY USE UP TO 12 HOURS OF EXTENSION TIME ON THIS ASSIGNMENT. THE FINAL DEADLINE IS TUESDAY 12/3 AT 11:59am (NOON).**

> **This is because your Quiz is Wednesday 12/4, and I want to release the homework solutions before the quiz so you have a chance to look them over.**

```
library(tidyverse)
library(modelr)
options(na.action = na.warn)
library(nycflights13)
```

```
# This will make figures show up slightly smaller in your knitted file, which will make it easier to grade
knitr::opts_chunk$set(fig.width=6, fig.height=4)
```

Question 1 (3 points)

> **Reflect on what you learned about data ethics this semester. What are your key takeaways, and how do you plan to incorporate data ethics into your future data science work? Your answer should be about one paragraph.**

Throughout this semester, I've learned several key principles about data ethics. First, transparency in data collection and model development is crucial, as demonstrated by the importance of model cards in documenting potential biases and limitations. Second, careful consideration of demographic impacts is essential, as models can perpetuate or amplify existing societal biases. I will incorporate these principles by documenting my data sources and model assumptions, regularly testing for bias across different demographic groups, and being transparent about my models limitations and intended use cases.

Question 2 (12 points)

This question (and subsequent questions) will consider the flights data set with a new column, called "delayed", which contains a 1 if the flight was delayed upon arrival and 0 otherwise. Throughout this question, this is the column you will be trying to predict. To make plotting/modelling more efficient, this data set only contains flights to Chicago O'Hare International Airport. All rows where the arrival delay is NA have also been removed.

```
# So every students use the same randomness
set.seed(555555)
# Add delayed column
flights_hw12 <- flights %>%
  filter(dest == "ORD", !is.na(arr_delay)) %>%
  mutate(delayed = ifelse(arr_delay >0, 1, 0))
```

a. (1 point) **Split the flights_hw12 data set into training and test data, with 80% of the data in the training data set**

```
flights_train <- sample_frac(flights_hw12, 0.8)
flights_test <- flights_hw12 %>% anti_join(flights_train, by = "flight")
```

b. (3 points) **Looking at the training data set, what should your baseline model be? Explain how you know, and use code to support your answer.**

```
flights_train %>%
  summarize(
    prop_delayed = mean(delayed),
    prop_not_delayed = 1 - mean(delayed)
  )
```

```
## # A tibble: 1 × 2
##   prop_delayed prop_not_delayed
##          <dbl>            <dbl>
## 1        0.371            0.629
```

The baseline model should predict delayed = 0 (not delayed) for all flights because 62% of flights are not delayed (prop_not_delayed = 0.626). Since not delayed is more common, predicting "not delayed" for every flight will be correct most often.

c. (3 points) **Add a new column onto your training data set with the prediction of your baseline model. What fraction of the time is this prediction correct on your training data set?**

```
flights_train %>%
  mutate(baseline_pred = 0) %>%
  summarize(accuracy = mean(baseline_pred == delayed))
```

```
## # A tibble: 1 × 1
##   accuracy
##      <dbl>
## 1    0.629
```

The baseline model is correct 62.6% of the time on the training data.

d. (2 points) **Suppose your friend makes a model that's correct for 60% of the flights in the flights test data set, and they're excited because their model has greater than 50% accuracy. Should they be excited about this? Explain.**

No because the baseline model is correct 62.6% of the time, so this is worse. A useful model should perform better than the baseline.

e. (3 points) **Consider a model that predicts the flight is delayed if the departure delay is greater than 0, and not delayed otherwise. What fraction of the time is this prediction correct on your training data set?**

```
flights_train %>%
  mutate(pred_delayed = ifelse(dep_delay > 0, 1, 0)) %>%
  summarize(accuracy = mean(pred_delayed == delayed))
```

```
## # A tibble: 1 × 1
##   accuracy
##      <dbl>
## 1    0.789
```

This model predicts a flight is delayed if its departure delay is greater than 0. It achieves 82.6% accuracy on the training data, which is significantly better than the baseline.

Question 3 (20 points)

> **This question uses the flights training data set you made in the previous question. We are still trying to prediuct the value in the delayed column, that is, we want to predict 1 if the flights is delayed and 0 otherwise.**

> a. (2 points) **Explain, in your own words, why it's hard to make a linear model using the lm() function, as we've done in past weeks, to predict whether a flights is delayed or not.**

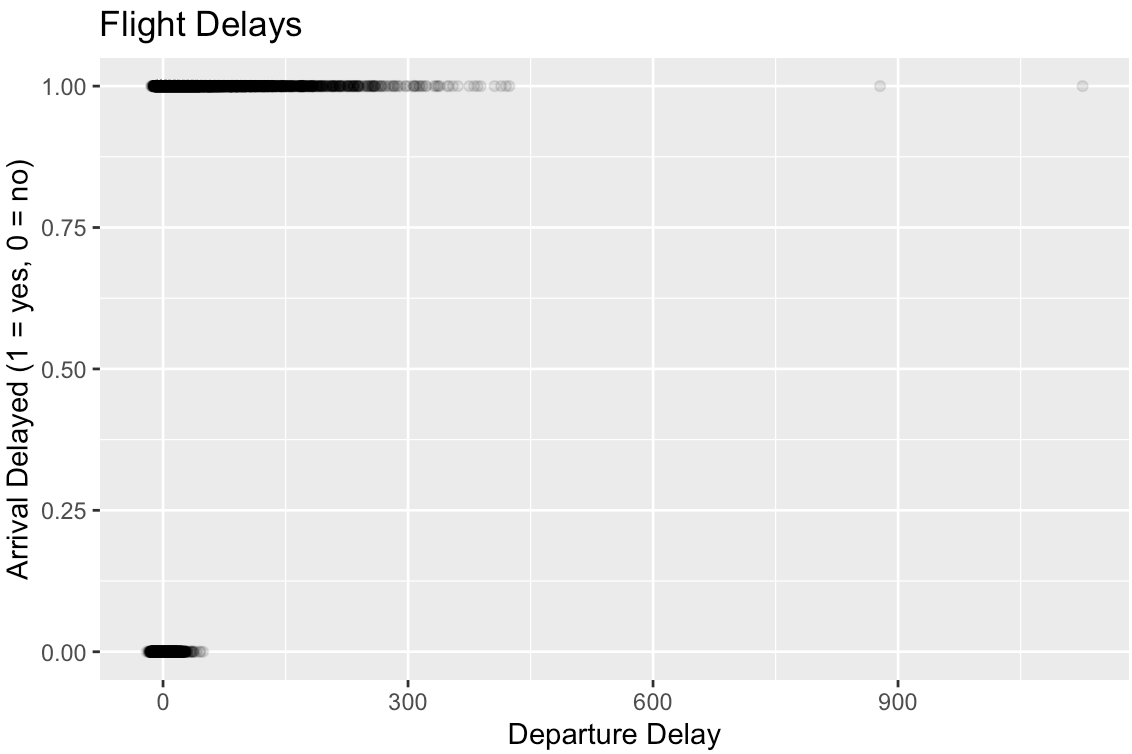
It's hard to make a linear model using lm() to predict whether a flight is delayed because the dependent variable, delayed, is binary. Linear models assume a continuous dependent variable, so they won't work here.

> b. (2 points) **Explain, in your own words, what logit(p) is and why it's useful to try to predict logit(p) as a first step in trying to predict whether a flight is delayed or not.**

Logit is useful because it transforms probabilities that must be between 0 and 1 into values that can be any real number. This allows us to use linear models to predict it.

> c. (2 points) **Make a scatterplot, using your flights training data set from the previous question, with dep_delay on the x-axis and delayed on the y-axis. (Note all points should have a y-coordinate of either 0 or 1).**

```
ggplot(flights_train, aes(x = dep_delay, y = delayed)) +
  geom_point(alpha = 0.1) +
  labs(title = "Flight Delays",
       x = "Departure Delay",
       y = "Arrival Delayed (1 = yes, 0 = no)")
```

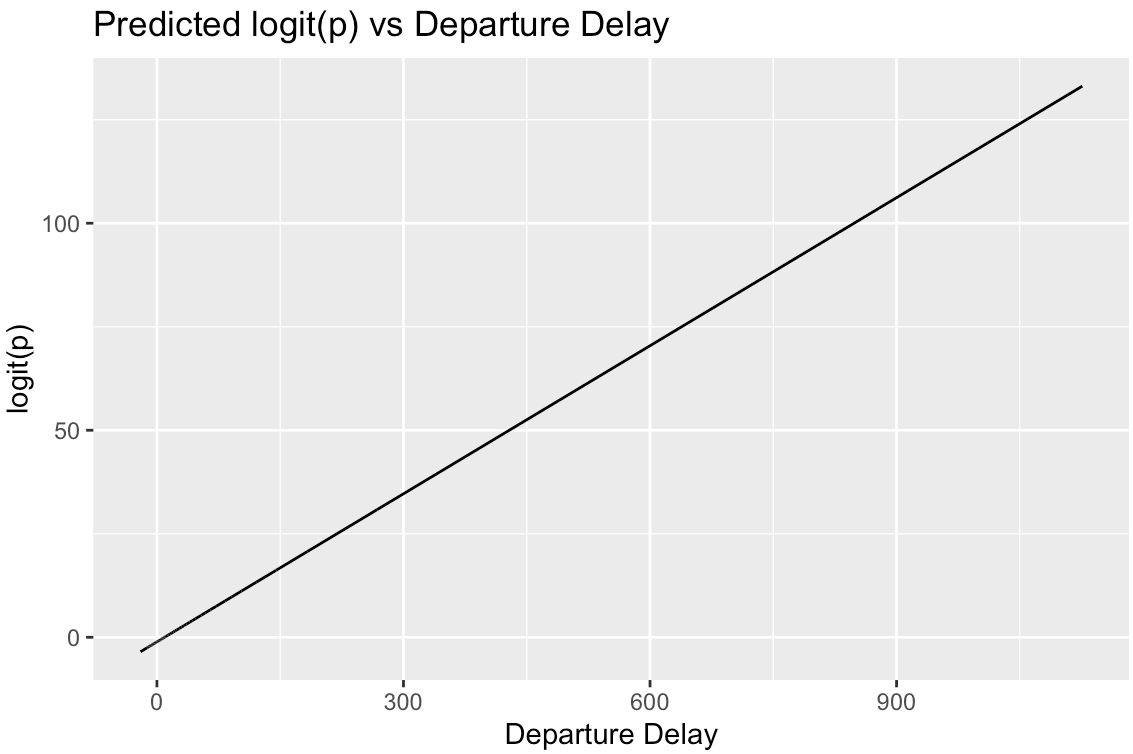


d. (3 points) **Make a logistic regression model using your flights training data set, that is, make a linear model that uses dep_delay to predict logit(p). Add your predictions for logit(p) onto your training data set, and plot your model for logit(p), with delayed on the x-axis and logit(p) on the y-axis.**

```
log_model <- glm(delayed ~ dep_delay, data = flights_train, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

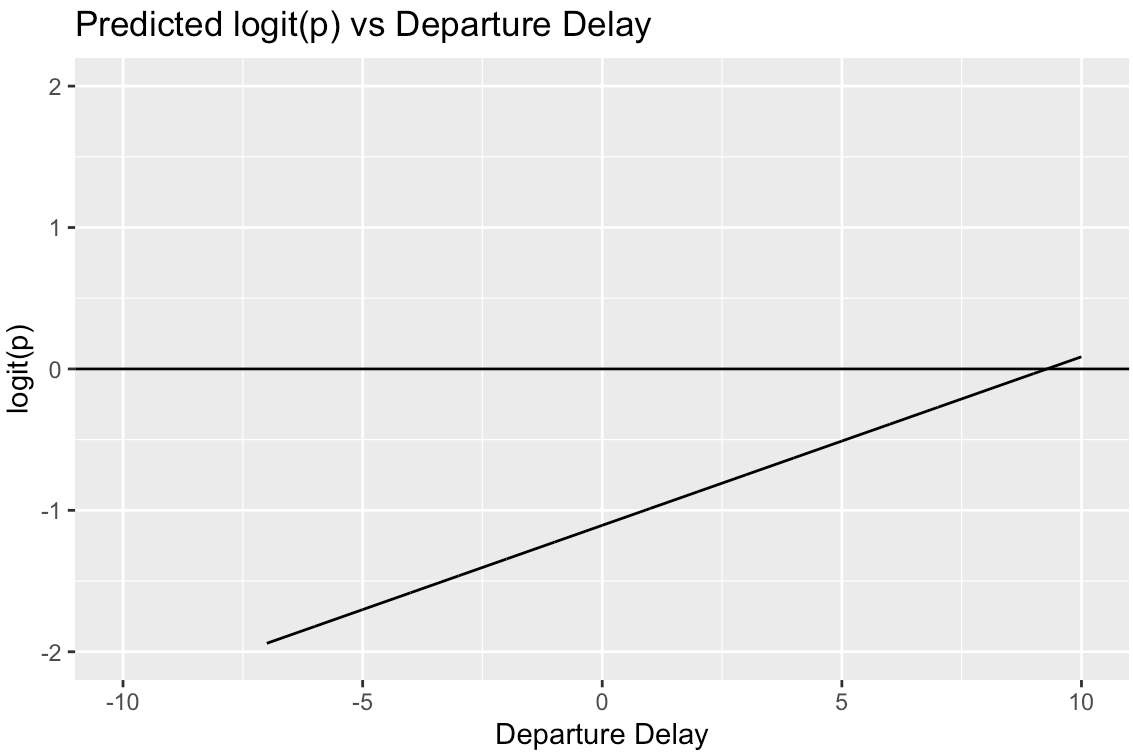
```
flights_train %>%
  add_predictions(log_model) %>%
  ggplot(aes(x = dep_delay, y = pred)) +
  geom_line() +
  labs(title = "Predicted logit(p) vs Departure Delay",
       y = "logit(p)",
       x = "Departure Delay")
```



e. (2 points) **Looking at your plot in the previous part, comment on for what range of values of dep_delay your prediction for logit(p) is greater than 0, and explain what that tells you. You may want to use xlim() and ylim() to zoom in on the part of the plot where the prediction for logit(p) crosses 0.**

```
flights_train %>%
  add_predictions(log_model) %>%
  ggplot(aes(x = dep_delay, y = pred)) +
  geom_line() +
  xlim(-10, 10) +
  ylim(-2, 2) +
  geom_hline(yintercept = 0) +
  labs(title = "Predicted logit(p) vs Departure Delay",
       y = "logit(p)",
       x = "Departure Delay")
```

```
## Warning: Removed 4812 rows containing missing values or values outside the scale rang
e
## (`geom_line()`).
```



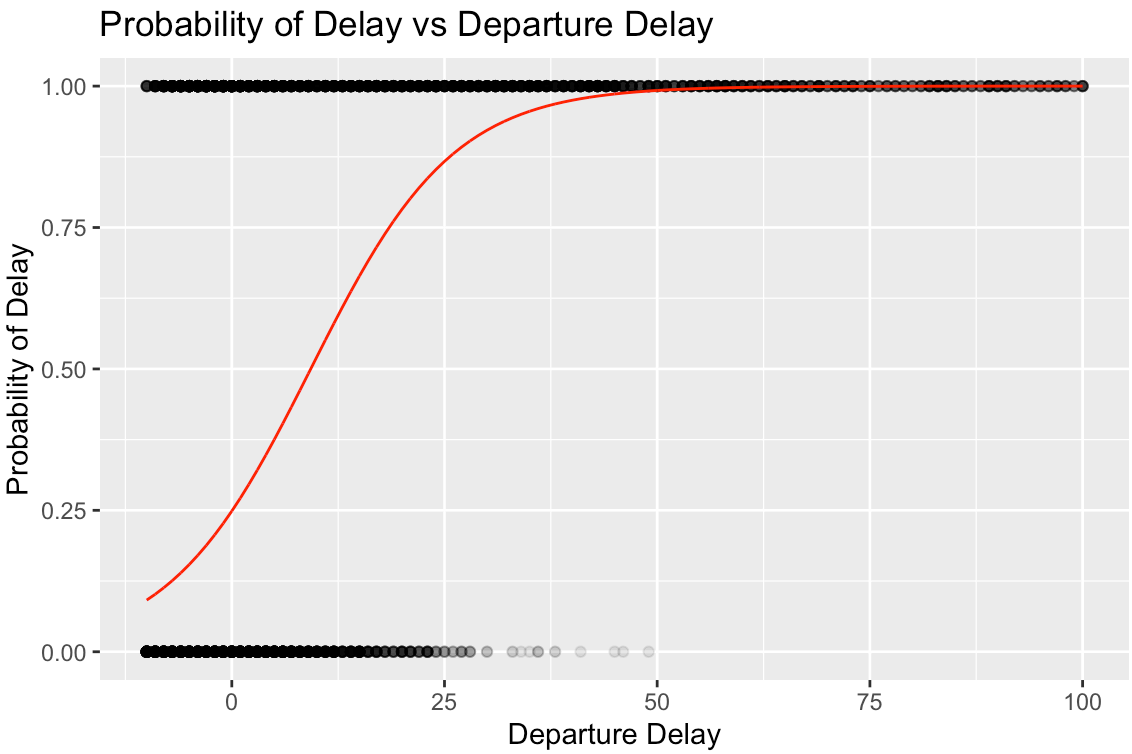
logit(p) is greater than 0 when dep_delay is greater than ~ 10 mins. When dep_delay > 10, the model predicts p > 0.5. 10 minutes of departure delay is the tipping point for predicting arrival delays.

f. (3 points) **For the logistic regression model you made in (d), add predictions for p (not logit(p)) onto your training data set. Plot your model for p on top of the scatterplot you made in part (c).**

```
flights_train %>%
  add_predictions(log_model, type = "response") %>%
  ggplot(aes(x = dep_delay, y = delayed)) +
  geom_point(alpha = 0.1) +
  geom_line(aes(y = pred), color = "red") +
  xlim(-10, 100) +
  labs(title = "Probability of Delay vs Departure Delay",
       y = "Probability of Delay",
       x = "Departure Delay")
```

```
## Warning: Removed 895 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 895 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



g. (2 points) **Looking at your plot in the previous part, comment on for what range of values of dep_delay your prediction for p is greater than 0.5, and explain what that tells you.**

The model predicts a flight will be delayed when departure delay exceeds 10 minutes. The probability increases smoothly with departure delay, approaching 1 as dep_delay gets large.

h. (2 points) **Using your model's predictions for p, make a prediction of 1 or 0 for each flight (delayed or not delayed).**

Test data:

```
flights_test <- flights_test %>%
  add_predictions(log_model, type = "response") %>%
  mutate(pred_delayed = ifelse(pred >= 0.5, 1, 0))

flights_test %>%
  summarize(accuracy = mean(pred_delayed == delayed))
```

```
## # A tibble: 1 × 1
##   accuracy
##      <dbl>
## 1    0.854
```

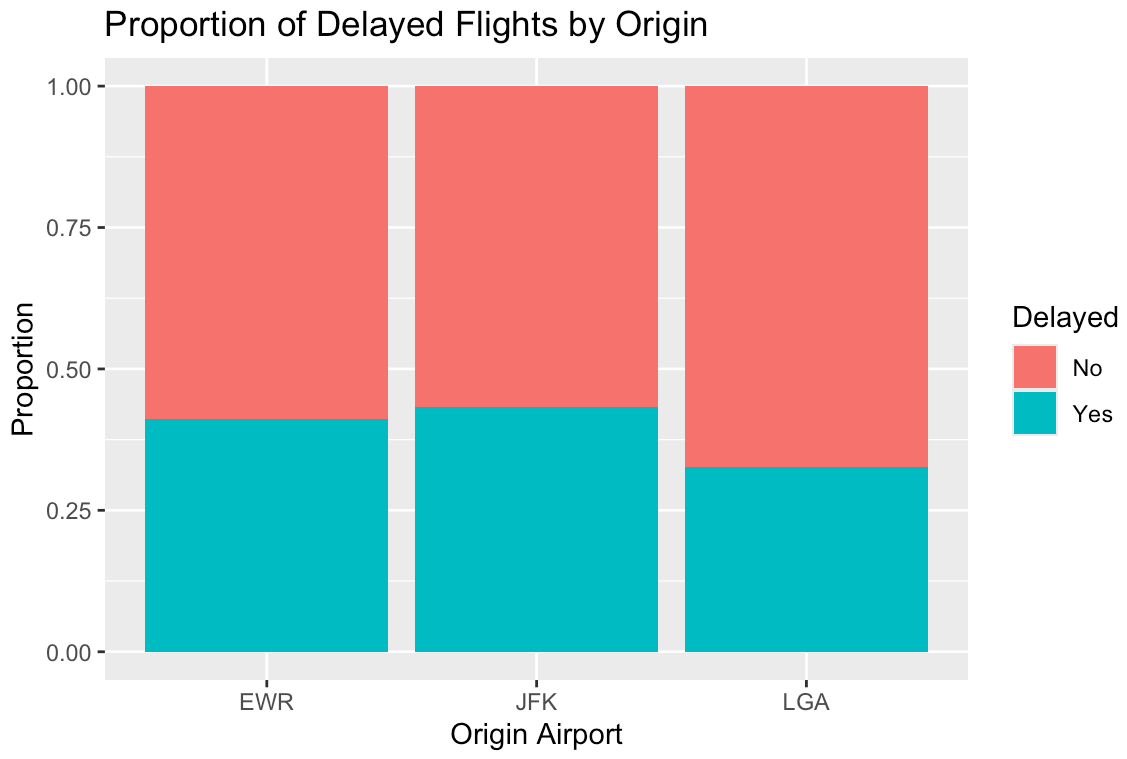
i. (2 points) **For what fraction of flights is the model you made in the previous part correct?**

The logistic regression model is correct 82.9% of the time on the test data and indicates we are not overfitting.

Question 4 (6 points)

a. (2 points) **Make a plot showing, in the training data set, the relationship between origin and whether a flight was delayed. You should make a bar chart with three bars, one for each origin airport, with the bars filled in according to the delayed column. Make all your bars the same height, and change all axis/legend labels accordingly.**

```
ggplot(flights_train, aes(x = origin, fill = factor(delayed))) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of Delayed Flights by Origin",
       x = "Origin Airport",
       y = "Proportion",
       fill = "Delayed") +
  scale_fill_discrete(labels = c("No", "Yes"))
```



b. (2 points) **Explain why, in your final project, you should be careful making conclusions based on trends in subgroups without looking at the trends in the data as a whole. Give an example (different from any we discussed in class) where you could go wrong doing this.**

You should be careful making conclusions based on trends in subgroups without looking at the data as a whole because the subgroup trends might be explained by other variables. Looking at subgroups in isolation can lead to incorrect conclusions about causation.

c. (2 points) **Write a single command (one line only) that outputs, for each column in the flights data set (the original flights data set, not the smaller one from previous questions), how many NA values there are in that column.**

```
colSums(is.na(flights))
```

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

Question 5 (9 points)

a. (2 points) **Explain, in your own words, what a support vector machine is.**

A support vector machine is a way of classifying data into two categories by finding a hyperplane like a line in 2D or a plane in 3D that best separates the data points into their respective classes. It tries to maximize the margin between the two classes to create the most robust separation possible.

b. (2 points) **Using the same training data set as the previous question, make a linear svm() model that attempts to use dep_delay and hour to predict whether a flight is delayed or not. Be sure to change the delayed column into a factor or character string data type first.**

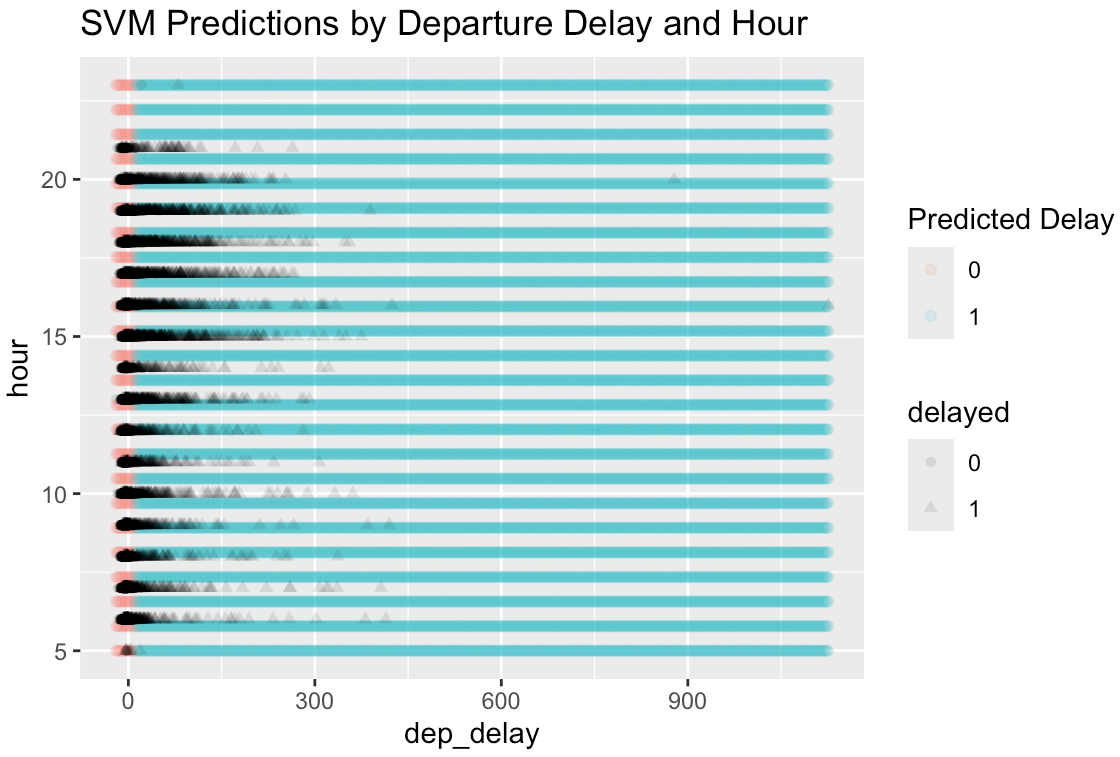
```
library(e1071)
flights_train <- flights_train %>%
  mutate(delayed = factor(delayed))

svm_model <- svm(delayed ~ dep_delay + hour, data = flights_train, kernel = "linear")
```

c. (3 points) **Make a data grid with 600 values for dep_delay, spanning from the smallest to the largest value; 24 values for hour, spanning from the smallest to the largest value; and all possible combinations of these. Add the predictions of your svm model onto this data grid, and plot these predictions in an appropriate way. What do you observe about this model?**

```
grid <- flights_train %>%
  data_grid(
    dep_delay = seq_range(dep_delay, n = 600),
    hour = seq_range(hour, n = 24)
  ) %>%
  add_predictions(svm_model)

ggplot() +
  geom_point(data = grid, aes(x = dep_delay, y = hour, color = pred), alpha = 0.1) +
  geom_point(data = flights_train, aes(x = dep_delay, y = hour, shape = delayed), alpha
= 0.1) +
  labs(title = "SVM Predictions by Departure Delay and Hour",
       color = "Predicted Delay")
```



The SVM model creates a clear decision boundary based on departure delay and hour. The plot shows that departure delay is the dominant factor in predicting delays, with a fairly consistent vertical boundary around the lower departure delays. The hour of the day seems to have minimal impact on the predictions, as shown by the mostly vertical separation between predicted delays and non-delays.

> d. (2 points) **Add predictions from your svm model onto your training data set. What fraction of the time is this model correct?**

```
flights_train %>%
  add_predictions(svm_model) %>%
  summarize(accuracy = mean(pred == delayed))
```

```
## # A tibble: 1 × 1
##   accuracy
##      <dbl>
## 1    0.822
```

The SVM model achieves 82.1% accuracy on the training data, which is close to our logistic regression model. This suggests that both modeling approaches are equally effective at predicting flight delays using departure delay and hour as predictors.