Module 2

> In this assignment you will build a ordinary linear regression models.
>
> Use *Tidyverse* and *Tidymodels* packages for the assignments.
>
> You can download the R Markdown file (https://gedeck.github.io/DS-6030/homework/Module-2.Rmd (https://gedeck.github.io/DS-6030/homework/Module-2.Rmd)) and use it as a basis for your solution.

1. Flexible vs Inflexible Methods (2 points)

For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.

(1.1) The sample size n is extremely large, and the number of predictors p is small.

A flexible method would be better. Because n is large and p is small, a flexible method would have enough data to learn complex patterns without overfitting.

(1.2) The number of predictors p is extremely large, and the number of observations n is small.

A flexible method would be worse. A flexible method has a high chance of overfitting when p is large, relative to a small n.

(1.3) The relationship between the predictors and response is highly non-linear.

A flexible method would be better. A flexible method are designed specifically to model non-linear relationships.

(1.4) The variance of the error terms, i.e. $\sigma^2 = Var(\epsilon)$, is extremely high.

A flexible method would be worse. With a high variance of error terms, a flexible method can lead to a high variance of predictions, which can cause overfitting.

2. Predicting Airfare on New Routes

The following problem takes place in the United States in the late 1990s, when many major US cities were facing issues with airport congestion, partly as a result of the 1978 deregulation of airlines. Both fares and routes were freed from regulation, and low-fare carriers such as Southwest (SW) began competing on existing routes and starting nonstop service on routes that previously lacked it. Building completely new airports is generally not feasible, but sometimes decommissioned military bases or smaller municipal airports can be reconfigured as regional or larger commercial airports. There are numerous players and interests involved in the issue (airlines, city, state and federal authorities, civic groups, the military, airport operators), and an aviation consulting firm is seeking advisory contracts with these players. The firm needs predictive models to support its consulting service. One thing the firm might want to be able to predict is fares, in the event a new airport is brought into service. The firm starts with the dataset *Airfares.csv.gz*, which contains real data that were collected between Q3-1996 and Q2-1997. The variables in these data are listed in the following Table, and are believed to be

important in predicting FARE. Some airport-to-airport data are available, but most data are at the city-to-city level. One question that will be of interest in the analysis is the effect that the presence or absence of Southwest has on FARE.

Variable	Description
S_CODE	Starting airport's code
S_CITY	Starting city
E_CODE	Ending airport's code
E_CITY	Ending city
COUPON	Average number of coupons (a one-coupon flight is a nonstop flight,a two-coupon flight is a one-stop flight, etc.) for that route
NEW	Number of new carriers entering that route between Q3-96 and Q2-97
VACATION	Whether (Yes) or not (No) a vacation route
SW	Whether (Yes) or not (No) Southwest Airlines serves that route
HI	Herfindahl index: measure of market concentration
S_INCOME	Starting city's average personal income
E_INCOME	Ending city's average personal income
S_POP	Starting city's population
E_POP	Ending city's population
SLOT	Whether or not either endpoint airport is slot-controlled (this is a measure of airport congestion)
GATE	Whether or not either endpoint airport has gate constraints (this is another measure of airport congestion)
DISTANCE	Distance between two endpoint airports in miles
PAX	Number of passengers on that route during period of data collection
FARE	Average fare on that route

A. Data Exploration:

(2.1) Load the data from `https://gedeck.github.io/DS-6030/datasets/homework/Airfares.csv.gz` and preprocess the data; convert categorical variables to factors. (1 point - coding)

```
library(tidyverse)
library(tidymodels)
library(GGally)  # scatterplot matrix
library(patchwork)  # for combining plots

# Load the data
airfares <- read_csv("https://gedeck.github.io/DS-6030/datasets/homework/Airfares.csv.
gz")

# Convert relevant columns to factors
airfares <- airfares %>%
  mutate(
    VACATION = as.factor(VACATION),
    SW = as.factor(SW),
    SLOT = as.factor(SLOT),
    GATE = as.factor(GATE)
  )

# View structure
glimpse(airfares)
```

```
## Rows: 638
## Columns: 18
## $ S_CODE   <chr> "*", "*", "*", "ORD", "MDW", "*", "*", "*", "*", "*", "*", "*…
## $ S_CITY   <chr> "Dallas/Fort Worth   TX", "Atlanta             GA", "Boston  …
## $ E_CODE   <chr> "*", "*", "*", "*", "*", "*", "*", "*", "*", "*", "*", "*", "…
## $ E_CITY   <chr> "Amarillo            TX", "Baltimore/Wash Intl MD", "Baltimor…
## $ COUPON   <dbl> 1.00, 1.06, 1.06, 1.06, 1.06, 1.01, 1.28, 1.15, 1.33, 1.60, 1…
## $ NEW      <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3…
## $ VACATION <fct> No, No, No, No, No, No, No, Yes, No, No, Yes, No, No, No, No,…
## $ SW       <fct> Yes, No, No, Yes, Yes, Yes, No, Yes, Yes, Yes, Yes, Yes, Yes,…
## $ HI       <dbl> 5291.99, 5419.16, 9185.28, 2657.35, 2657.35, 3408.11, 6754.48…
## $ S_INCOME <dbl> 28637, 26993, 30124, 29260, 29260, 26046, 28637, 26752, 27211…
## $ E_INCOME <dbl> 21112, 29838, 29838, 29838, 29838, 29838, 29838, 29838, 29838…
## $ S_POP    <dbl> 3036732, 3532657, 5787293, 7830332, 7830332, 2230955, 3036732…
## $ E_POP    <dbl> 205711, 7145897, 7145897, 7145897, 7145897, 7145897, 7145897,…
## $ SLOT     <fct> Free, Free, Free, Controlled, Free, Free, Free, Free, Free, F…
## $ GATE     <fct> Free, Free, Free, Free, Free, Free, Free, Free, Free, Free, F…
## $ DISTANCE <dbl> 312, 576, 364, 612, 612, 309, 1220, 921, 1249, 964, 2104, 232…
## $ PAX      <dbl> 7864, 8820, 6452, 25144, 25144, 13386, 4625, 5512, 7811, 4657…
## $ FARE     <dbl> 64.11, 174.47, 207.76, 85.47, 85.47, 56.76, 228.00, 116.54, 1…
```

(2.2) Explore the numerical (continuous) predictors and response (FARE) by creating a correlation table and examining some scatterplots between FARE and those predictors. What seems to be the best single predictor of FARE? (2 points - coding/discussion)

```
# Select numeric variables w/ FARE
numeric_vars <- airfares %>%
  select(where(is.numeric), FARE)

# Correlation matrix
cor_matrix <- cor(numeric_vars, use = "complete.obs")
cor_matrix
```

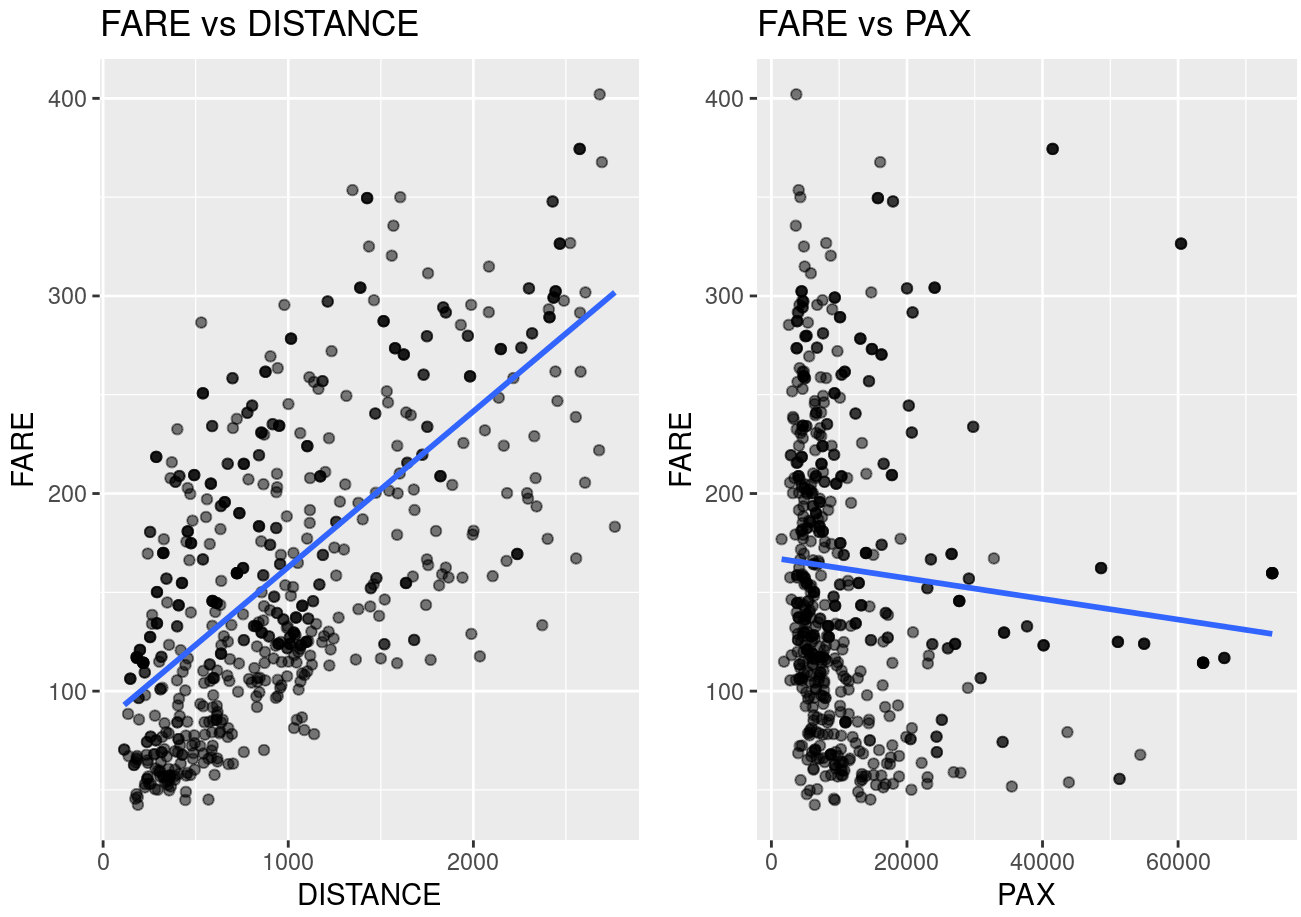
```
##                COUPON         NEW          HI    S_INCOME   E_INCOME       S_POP
## COUPON    1.00000000  0.02022307 -0.34725207 -0.08840265  0.0468892 -0.10776336
## NEW       0.02022307  1.00000000  0.05414685  0.02659673  0.1133766 -0.01667212
## HI       -0.34725207  0.05414685  1.00000000 -0.02738221  0.0823926 -0.17249541
## S_INCOME -0.08840265  0.02659673 -0.02738221  1.00000000 -0.1388642  0.51718718
## E_INCOME  0.04688920  0.11337664  0.08239260 -0.13886420  1.0000000 -0.14405857
## S_POP    -0.10776336 -0.01667212 -0.17249541  0.51718718 -0.1440586  1.00000000
## E_POP     0.09496994  0.05856818 -0.06245600 -0.27228027  0.4584181 -0.28014283
## DISTANCE  0.74680521  0.08096520 -0.31237457  0.02815334  0.1765307  0.01843667
## PAX      -0.33697358  0.01049527 -0.16896078  0.13819710  0.2599611  0.28461056
## FARE      0.49653696  0.09172969  0.02519492  0.20913485  0.3260923  0.14509708
##                E_POP    DISTANCE         PAX        FARE
## COUPON    0.09496994  0.74680521 -0.33697358  0.49653696
## NEW       0.05856818  0.08096520  0.01049527  0.09172969
## HI       -0.06245600 -0.31237457 -0.16896078  0.02519492
## S_INCOME -0.27228027  0.02815334  0.13819710  0.20913485
## E_INCOME  0.45841806  0.17653074  0.25996105  0.32609229
## S_POP    -0.28014283  0.01843667  0.28461056  0.14509708
## E_POP     1.00000000  0.11563970  0.31469750  0.28504299
## DISTANCE  0.11563970  1.00000000 -0.10248160  0.67001599
## PAX       0.31469750 -0.10248160  1.00000000 -0.09070541
## FARE      0.28504299  0.67001599 -0.09070541  1.00000000
```

```
# FARE vs DISTANCE and PAX
p1 <- ggplot(airfares, aes(x = DISTANCE, y = FARE)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "FARE vs DISTANCE")

p2 <- ggplot(airfares, aes(x = PAX, y = FARE)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "FARE vs PAX")

# Combine plots
p1 + p2
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



It seems that DISTANCE has the strongest correlation with FARE.

(2.3) Explore the categorical predictors (excluding the first four) by creating individual graphs comparing the distribution of average fare for each category (e.g. box plots). Which categorical predictor seems best for predicting FARE? (2 points - coding/discussion)

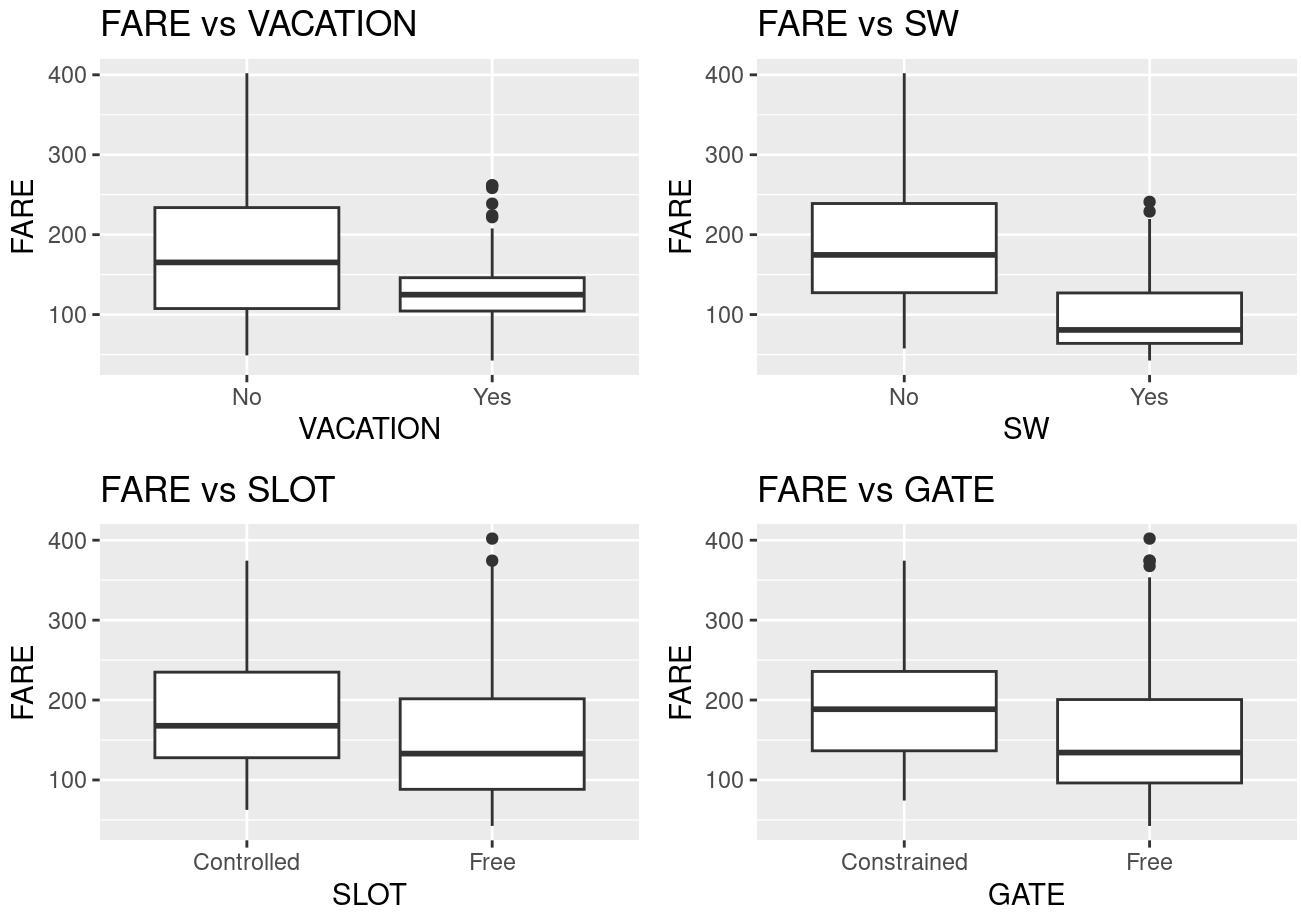
```
# Boxplots of FARE w/ categorical variables
p1 <- ggplot(airfares, aes(x = VACATION, y = FARE)) + geom_boxplot() + ggtitle("FARE v
s VACATION")

p2 <- ggplot(airfares, aes(x = SW, y = FARE)) + geom_boxplot() + ggtitle("FARE vs SW")

p3 <- ggplot(airfares, aes(x = SLOT, y = FARE)) + geom_boxplot() + ggtitle("FARE vs SL
OT")

p4 <- ggplot(airfares, aes(x = GATE, y = FARE)) + geom_boxplot() + ggtitle("FARE vs GA
TE")

(p1 | p2) / (p3 | p4)
```



The categorical variable that seems to be best for predicting FARE is SW. Routes that are with SW tend to have lower FARE.

B. Find a model for predicting the average fare on a new route:

(2.4) Partition the data into training and holdout sets. The model will be fit to the training data and evaluated on the holdout set. (see DS-6030: Creating an initial split of the data into training and holdout set (https://gedeck.github.io/DS-6030/book/sampling.html#creating-an-initial-split-of-the-data-into-training-and-holdout-set)) (1 point - coding)

```
set.seed(123)

# Split data into training and testing sets
airfare_split <- initial_split(airfares, prop = 0.8, strata = FARE)
airfare_train <- training(airfare_split)
airfare_test  <- testing(airfare_split)
```

(2.5) Train a linear regression model with *tidymodels* using all predictors. You can ignore the first four predictors (S_CODE, S_CITY, E_CODE, E_CITY). Examine the model coefficients and interpret them. Which predictors are significant? (see DS-6030: Linear regression models (https://gedeck.github.io/DS-6030/book/linear-regression.html))

```
# exclude S_CODE, S_CITY, E_CODE, E_CITY
fare_recipe_full <- recipe(FARE ~ ., data = airfare_train) %>%
  update_role(S_CODE, S_CITY, E_CODE, E_CITY, new_role = "ID") %>%
  step_dummy(all_nominal_predictors())

# Model specification
lm_spec <- linear_reg() %>% set_engine("lm")

# Workflow
fare_wf_full <- workflow() %>%
  add_recipe(fare_recipe_full) %>%
  add_model(lm_spec)

# Fit model
fare_fit_full <- fit(fare_wf_full, data = airfare_train)

# View model summary
model_summary <- fare_fit_full %>% extract_fit_parsnip() %>% tidy()
model_summary %>% filter(p.value < 0.05)
```

```
## # A tibble: 11 × 5
##    term             estimate   std.error statistic  p.value
##    <chr>               <dbl>       <dbl>     <dbl>    <dbl>
##  1 HI             0.00855    0.00114          7.54 2.34e-13
##  2 S_INCOME       0.00128    0.000579         2.21 2.78e- 2
##  3 E_INCOME       0.00132    0.000411         3.21 1.42e- 3
##  4 S_POP          0.00000343 0.000000722      4.75 2.71e- 6
##  5 E_POP          0.00000441 0.000000841      5.25 2.31e- 7
##  6 DISTANCE       0.0757     0.00399         19.0  8.81e-61
##  7 PAX           -0.000837   0.000168        -5.00 8.12e- 7
##  8 VACATION_Yes -35.9        4.03            -8.90 1.05e-17
##  9 SW_Yes       -40.0        4.26            -9.40 2.01e-19
## 10 SLOT_Free    -17.2        4.32            -3.98 7.91e- 5
## 11 GATE_Free    -19.6        4.54            -4.33 1.81e- 5
```

It seems that the most significant predictors are DISTANCE, SW, and COUPON.

Determine the model performance using `r^2`, `RMSE` and `MAE` on the training and test set. How does the model perform on the test set? Is the model overfitting? How can you tell? (see DS-6030: Measuring performance of regression models (https://gedeck.github.io/DS-6030/book/regression-metrics.html)) (2 points - coding/discussion)

```
# Performance metrics
fare_results_train <- fare_fit_full %>%
  predict(new_data = airfare_train) %>%
  bind_cols(airfare_train %>% select(FARE)) %>%
  metrics(truth = FARE, estimate = .pred)

fare_results_test <- fare_fit_full %>%
  predict(new_data = airfare_test) %>%
  bind_cols(airfare_test %>% select(FARE)) %>%
  metrics(truth = FARE, estimate = .pred)

fare_results_train
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      35.3
## 2 rsq     standard       0.788
## 3 mae     standard      27.6
```

```
fare_results_test
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      34.4
## 2 rsq     standard       0.779
## 3 mae     standard      27.3
```

Because the training has a higher RMSE and RSQ value, it looks like the model is not overfitting.

(2.6) Taking the results from **(2.2)**, **(2.3)**, and **(2.5)** into account, build a model that includes only the most important predictors. Determine the model performance and compare with the full model from **(2.5)**. (2 points - coding/discussion)

```
# reduced model
fare_recipe_reduced <- recipe(FARE ~ DISTANCE + SW + COUPON + PAX + HI, data = airfare
_train) %>%
  step_dummy(all_nominal_predictors())

# Workflow
fare_wf_reduced <- workflow() %>%
  add_recipe(fare_recipe_reduced) %>%
  add_model(lm_spec)

# Fit reduced model
fare_fit_reduced <- fit(fare_wf_reduced, data = airfare_train)

# Evaluate reduced model
reduced_train_metrics <- fare_fit_reduced %>%
  predict(new_data = airfare_train) %>%
  bind_cols(airfare_train %>% select(FARE)) %>%
  metrics(truth = FARE, estimate = .pred)

reduced_test_metrics <- fare_fit_reduced %>%
  predict(new_data = airfare_test) %>%
  bind_cols(airfare_test %>% select(FARE)) %>%
  metrics(truth = FARE, estimate = .pred)

reduced_train_metrics
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      45.7
## 2 rsq     standard       0.644
## 3 mae     standard      37.0
```

```
reduced_test_metrics
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      43.6
## 2 rsq     standard       0.648
## 3 mae     standard      35.4
```

The performance of the reduced model differs from the full model, as the values of the reduced model are higher than those of the full. Therefore, the full model is preferred.

(2.7) Using the models from **(2.5)** and **(2.6)**, predict the average fare on a route with the following characteristics: COUPON = 1.202, NEW = 3, VACATION = No, SW = No, HI = 4442.141, S_INCOME = $28,760, E_INCOME = $27,664, S_POP = 4,557,004, E_POP = 3,195,503, SLOT = Free, GATE = Free, PAX = 12,782, DISTANCE = 1976 miles. *Hint*: make sure that you treat the categorical variables in the same way as in the training data. (1 point - coding)

```
# Example data point
new_route <- tibble(
  S_CODE = "",
  S_CITY = "",
  E_CODE = "",
  E_CITY = "",
  COUPON = 1.202,
  NEW = 3,
  VACATION = factor("No", levels = levels(airfare_train$VACATION)),
  SW = factor("No", levels = levels(airfare_train$SW)),
  HI = 4442.141,
  S_INCOME = 28760,
  E_INCOME = 27664,
  S_POP = 4557004,
  E_POP = 3195503,
  SLOT = factor("Free", levels = levels(airfare_train$SLOT)),
  GATE = factor("Free", levels = levels(airfare_train$GATE)),
  DISTANCE = 1976,
  PAX = 12782,

)

# Full model prediction
predict(fare_fit_full, new_data = new_route)
```

```
## # A tibble: 1 × 1
##   .pred
##   <dbl>
## 1  250.
```

```
# Reduced model prediction
predict(fare_fit_reduced, new_data = new_route)
```

```
## # A tibble: 1 × 1
##   .pred
##   <dbl>
## 1  251.
```

The predicted value for the full model is 250.91 and the predicted value for the reduced model is 250.76.

(2.8) Using the smaller model from **(2.6)**, predict the reduction in average fare on the route in **(2.7)** if Southwest decides to cover this route. (1 point - coding/discussion)

```
# New route with SW
new_route_SW <- new_route %>%
  mutate(SW = factor("Yes", levels = levels(airfares$SW)))

# Predict for both scenarios using reduced model
fare_no_SW <- predict(fare_fit_reduced, new_data = new_route)$.pred
fare_with_SW <- predict(fare_fit_reduced, new_data = new_route_SW)$.pred

fare_no_SW - fare_with_SW
```

```
## [1] 60.03794
```

The reduction in average fare is 60.04.

C. Predictors

(2.9) In reality, which of the factors will not be available for predicting the average fare from a new airport (i.e., before flights start operating on those routes)? Which ones can be estimated? How? (1 point - discussion)

The factors that will not be available include COUPON, PAX, FARE, and NEW. VACATION and DISTANCE can be estimated because distance can measured from airport locations and vacations can be measured based on demand.

(2.10) Train a model that includes only factors that are available before flights begin to operate on the new route. (1 point - coding)

```
# only available-before-flight variables
preop_recipe <- recipe(FARE ~ DISTANCE + SW + VACATION + HI + S_INCOME + E_INCOME + S_
POP + E_POP + SLOT + GATE,
                       data = airfare_train) %>%
  step_dummy(all_nominal_predictors())

# Workflow
preop_wf <- workflow() %>%
  add_recipe(preop_recipe) %>%
  add_model(lm_spec)

# Fit model
preop_fit <- fit(preop_wf, data = airfare_train)

# Evaluate performance
preop_train_metrics <- preop_fit %>%
  predict(new_data = airfare_train) %>%
  bind_cols(airfare_train %>% select(FARE)) %>%
  metrics(truth = FARE, estimate = .pred)

preop_test_metrics <- preop_fit %>%
  predict(new_data = airfare_test) %>%
  bind_cols(airfare_test %>% select(FARE)) %>%
  metrics(truth = FARE, estimate = .pred)

preop_train_metrics
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      36.5
## 2 rsq     standard       0.773
## 3 mae     standard      28.8
```

```
preop_test_metrics
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      35.8
## 2 rsq     standard       0.761
## 3 mae     standard      28.5
```

(2.11) Compare the predictive accuracy of this model with models from **(2.5)** and **(2.6)**. Is this model good enough, or is it worthwhile reevaluating the model once flights begin on the new route? (1 point - discussion)

It seems the model is good enough, especially when its values are compared to those in 2.5 and 2.6.