Module 2 HW

Alanna Hazlett

2024-06-02

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) and use it as a basis for your solution.

1. Flexible vs Inflexible Methods

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.

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

We would expect that a more flexible statistical learning method would be better. With a large number of observations the flexible model is more likely to accurately capture the true f than a non-flexible model that imposes assumptions on the data. The change of a single observation would not likely change the predicted model.

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

We would expect that a more flexible statistical learning method would be worse. The model would be prone to overfitting, a change for a single observation would make a large difference in the predicted f.

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

We would expect that a more flexible statistical learning method would be better. Flexible models have a wider range of possible f that they can approximate than non-flexible models.

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

We would expect that a more flexible statistical model would be worse. A flexible model would follow the errors more than a non-flexible model and would lead to poor predictions for new data.

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

```
airfare_raw<-read.csv("Airfares.csv.gz",na = c("*"))
airfare_raw %>%
  mutate(S_CODE=as.factor(S_CODE),
         S_CITY=as.factor(S_CITY),
         E_CODE=as.factor(E_CODE),
         E_CITY=as.factor(E_CITY),
         NEW=as.factor(NEW),
         VACATION=as.factor(VACATION),
         SW=as.factor(SW),
         SLOT=as.factor(SLOT),
         GATE=as.factor(GATE))
airfare_raw %>% glimpse()
```

(b.) Explore the numerical 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?

```
ggpairs(airfare_nocat)
```

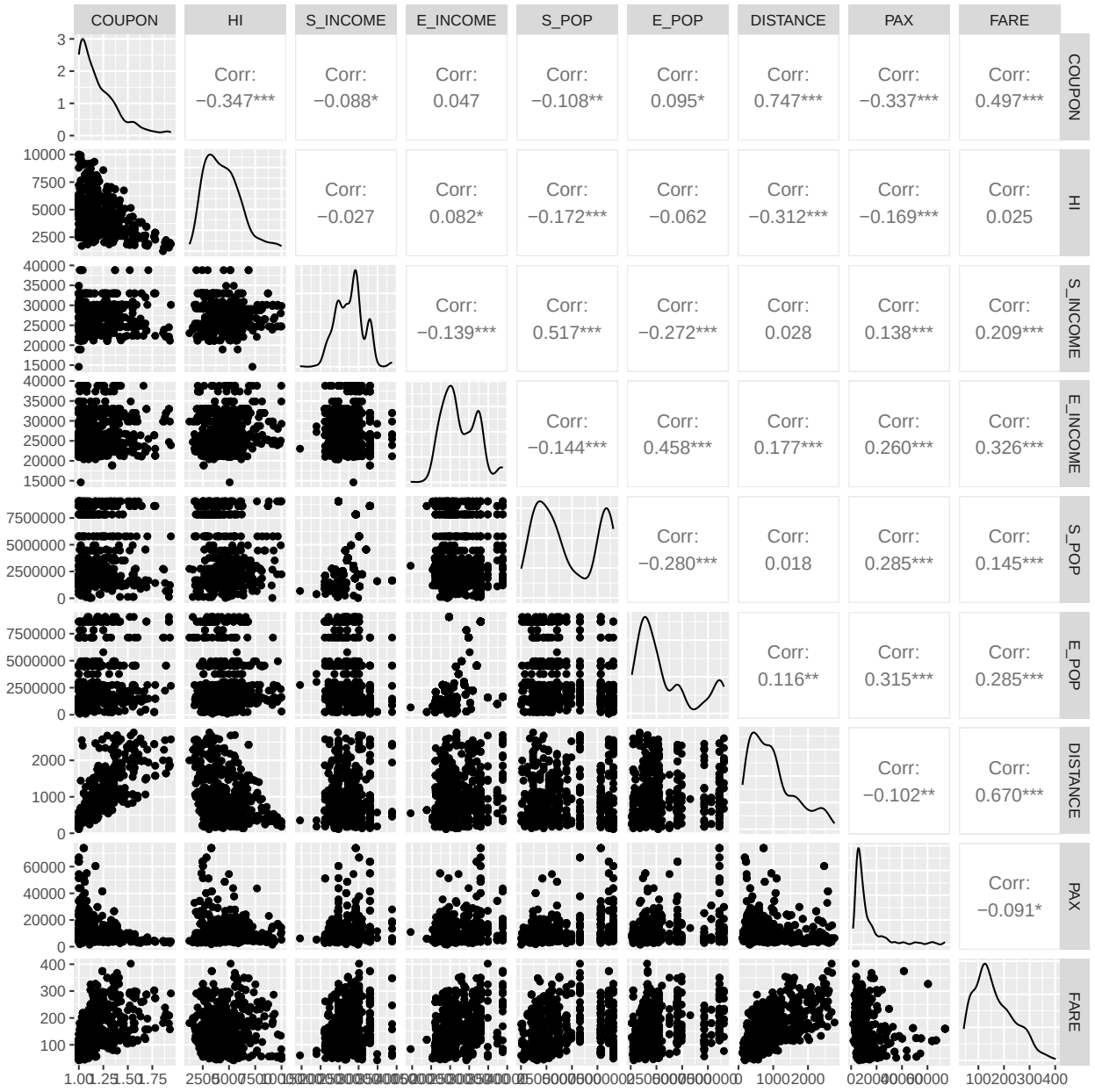


Figure 1: Pairs plot of the Airfare dataset

- Generally most of our numerical predictors have very weak correlations with FARE, our response variable. We have two variables with some significant correlation COUPON and DISTANCE, with DISTANCE having a stronger correlation. We see that COUPON and DISTANCE are highly correlated with eachother as well, so we will likely only need one for our model.

```
 airfare_raw %>%
    dplyr::select(COUPON,HI,S_INCOME,E_INCOME,S_POP,E_POP,DISTANCE,PAX,FARE) %>%
    pivot_longer(-c(FARE),names_to="predictor") %>%
ggplot(aes(x=value, y=FARE)) +
    geom_point(alpha=0.5) +
```

```
    geom_smooth() +
    facet_wrap(~ predictor, scales="free_x")
```

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

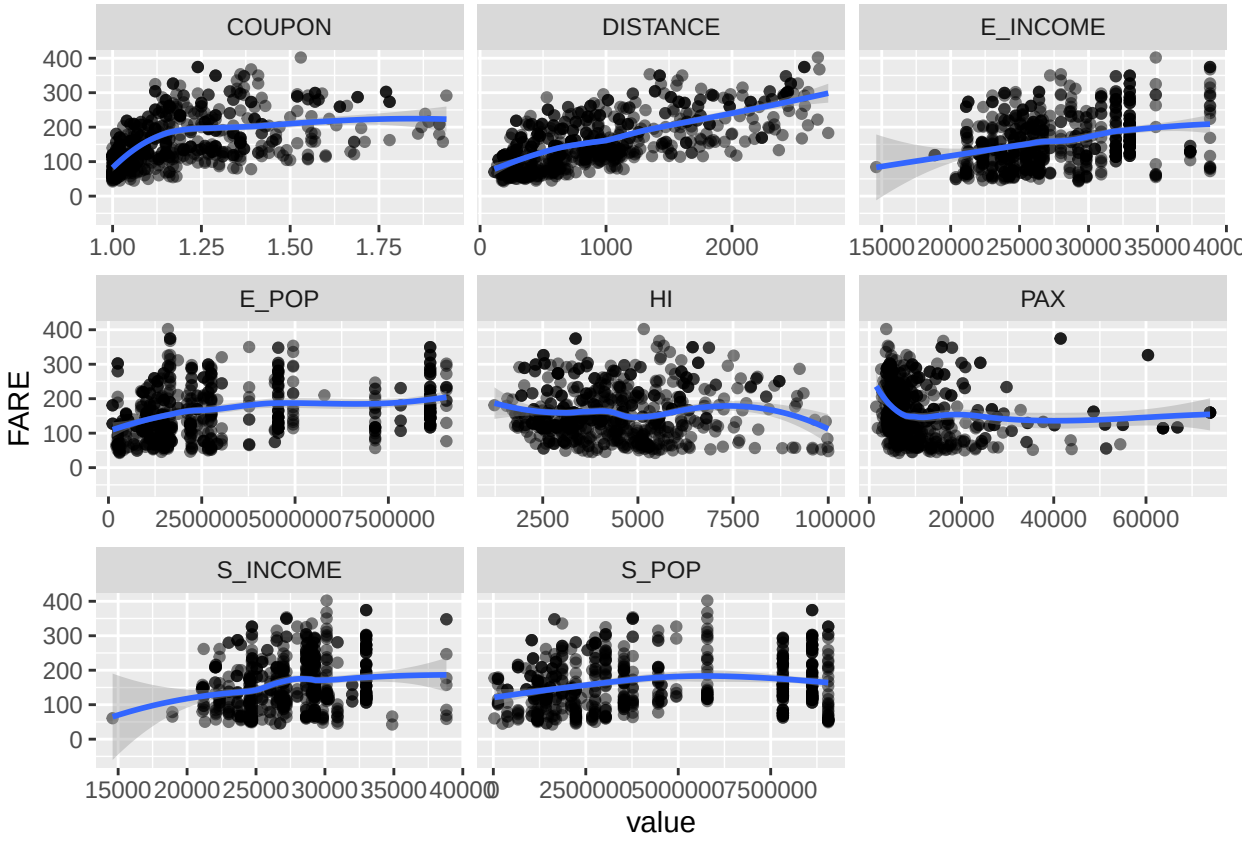


Figure 2: Numerical Scatterplots of the Airfare dataset

- Most of our scatterplots show relatively no pattern: E_POP, HI, PAX, S_INCOME, S_POP. We see a slight positive linear association with E_INCOME. There is a pattern displayed in COUPON, which may indicate that there is a relationship, but not a simple linear relationship. There is a positive linear association of DISTANCE and FARE. DISTANCE seems to be the best single predictor of FARE.

(c.) 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?

```
p1<-ggplot(airfare_raw, aes(x=VACATION, y=FARE))+
  geom_boxplot()+
  labs(title="Fare against Vacation Destination")
p2<-ggplot(airfare_raw, aes(x=SW, y=FARE))+
  geom_boxplot()+
  labs(title="Fare against Southwest Airline")
p3<-ggplot(airfare_raw, aes(x=SLOT, y=FARE))+
  geom_boxplot()+
  labs(title="Fare against Slot Controlled")
p4<-ggplot(airfare_raw, aes(x=GATE, y=FARE))+
  geom_boxplot()+
  labs(title="Fare against Gate Constraints")
```

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

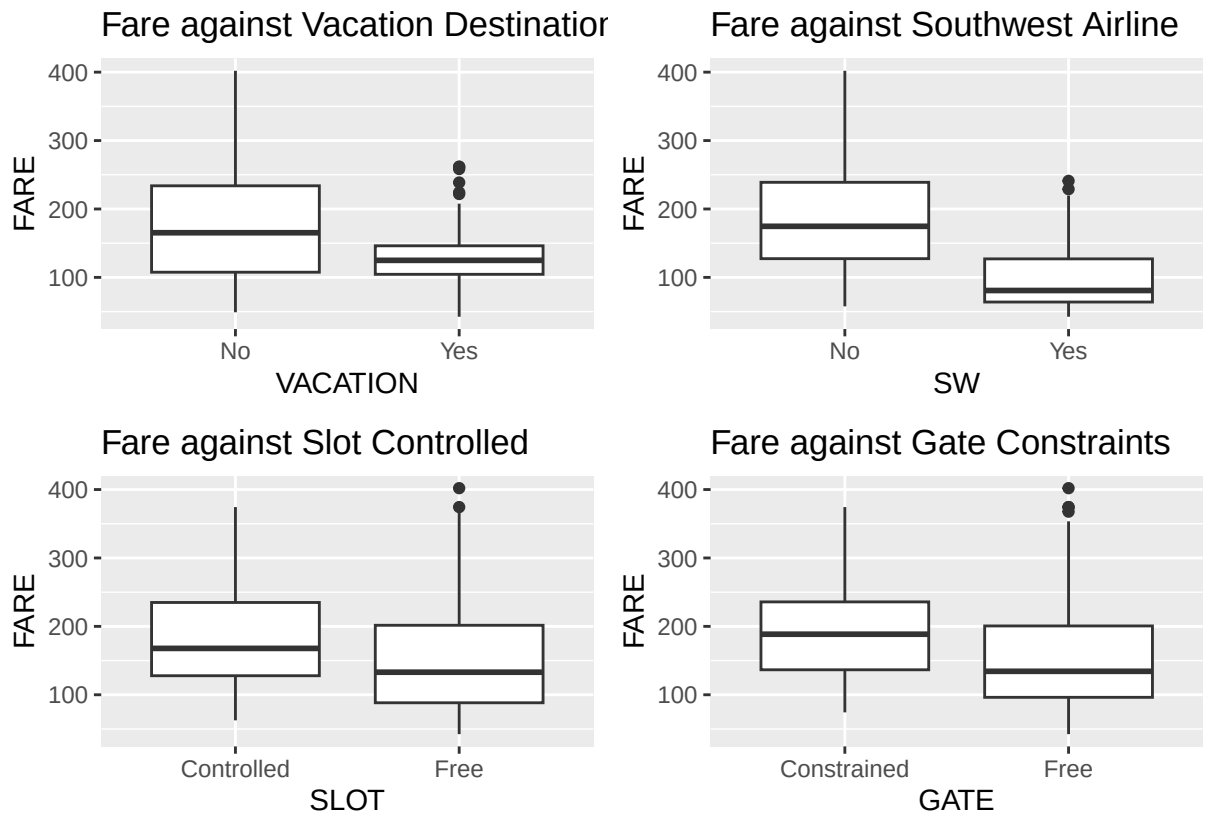


Figure 3: Categorical Boxplots of the Airfare dataset

* In fare against vacation destination we see that the median fare price is lower for a vacation destination. Vacation destinations also have a significantly smaller interquartile range, indicating that the price does not vary as much as destinations that are not vacation destinations.
* In fare against Southwest we see that fare prices are significantly lower for Southwest flights compared to other airlines. Southwest also has a smaller variance in their fare prices compared to other airlines.
* In fare against slot controlled the controlled airports have higher fare prices, but fare prices vary about the same among controlled versus free airports.
* In fare against gate constraints we see that constrained airports have higher fare prices, but fare prices vary about the same amount among contrained versus free airports.

(d.) Find a model for predicting the average fare on a new route:

(i.) 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)

```
set.seed(1)  # for reproducibility
airfare_split <- initial_split(airfare_raw, prop=0.75, strata=FARE)
train <- training(airfare_split)
test <- testing(airfare_split)
```

(ii.) 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)

```
formula <- FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME +E_INCOME + S_POP + E_POP + SLOT + GATE +

lm_spec <- linear_reg(engine="lm", mode="regression")
lm_model <- lm_spec %>% fit(formula, data=train)
```

```
lm_model %>%
    tidy() %>%
    knitr::kable(digits=3, caption="Linear regression model parameters extracted using the tidy functio
```

Table 2: Linear regression model parameters extracted using the tidy function

term	estimate	std.error	statistic	p.value
(Intercept)	17.362	31.641	0.549	0.583
COUPON	0.330	14.415	0.023	0.982
NEW	-1.559	2.167	-0.720	0.472
VACATIONYes	-34.918	4.088	-8.541	0.000
SWYes	-40.084	4.371	-9.171	0.000
HI	0.008	0.001	6.909	0.000
S_INCOME	0.001	0.001	2.021	0.044
E_INCOME	0.001	0.000	3.242	0.001
S_POP	0.000	0.000	4.514	0.000
E_POP	0.000	0.000	5.030	0.000
SLOTFree	-18.111	4.420	-4.097	0.000
GATEFree	-19.791	4.518	-4.380	0.000
DISTANCE	0.076	0.004	18.384	0.000
PAX	-0.001	0.000	-5.324	0.000

- COUPON: A one unit increase in COUPON will result in an increase in FARE on average by $0.33, while the other predictors are held constant.

- NEW: A one unit increase in NEW will result in a decrease in FARE on average by $-1.56, while the other predictors are held constant.

- VACATION: If the destination is a vacation destination the FARE will decrease on average compared to a non-vacation destination by $34.92, while the other predictors are held constant.

- SW: If the flight is chartered by Southwest airline the FARE will decrease on average compared to other airlines by $40.08, while the other predictors are held constant.

- HI: A one unit increase in the Herfindahl index, a measure of market concentration, will result in an increase in FARE on average by $0.008, while the other predictors are held constant.

- S_INCOME: A one unit increase in the starting city's average personal income will result in an increase in FARE on average of $0.001, while the other predictors are held constant.

- E_INCOME: A one unit increase in the ending city's average personal income will result in an increase in FARE on average of $0.001, while the other predictors are held constant.

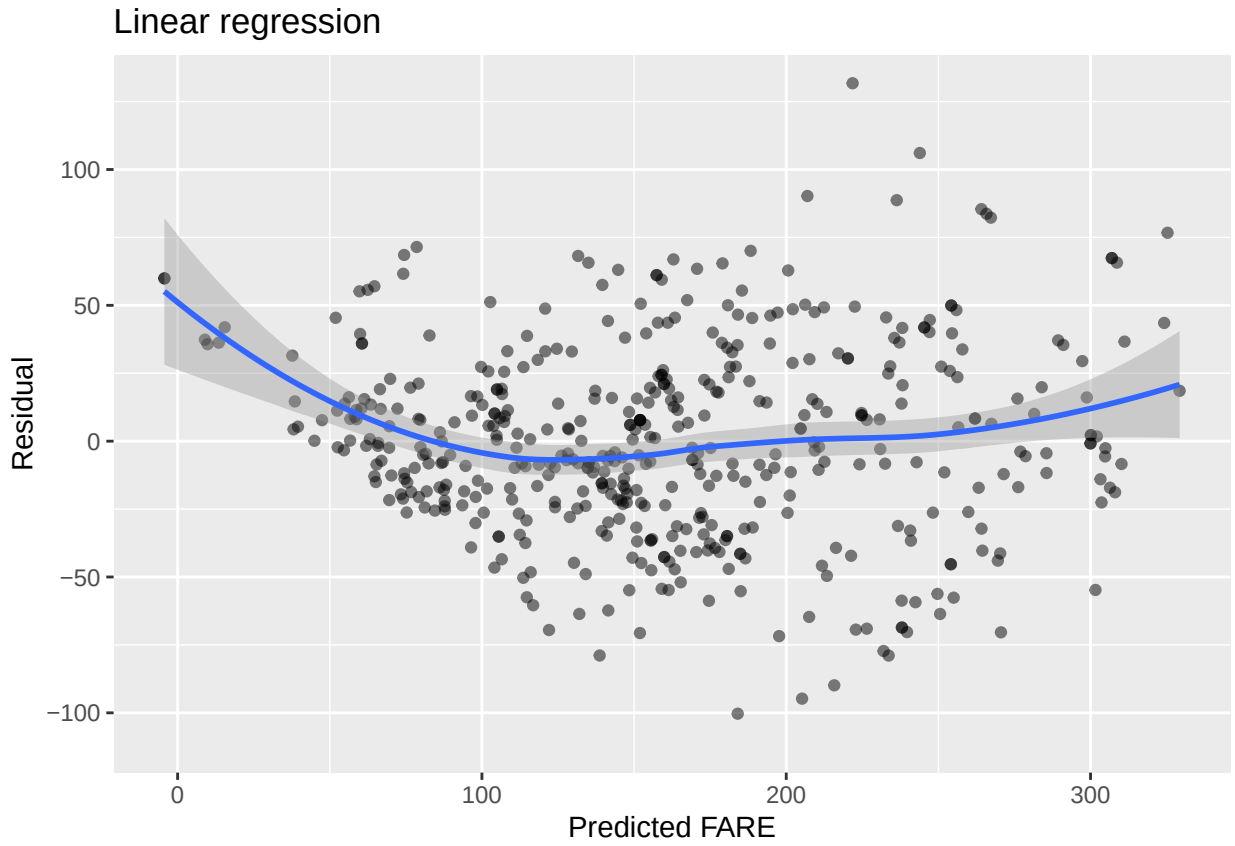
- S_POP: A one unit increase in the starting city's population results in no change to FARE on average, while the other predictors are held constant.
- E_POP: A one unit increase in the ending city's population results in no change to FARE on average, while the other predictors are held constant.
- SLOT: If the endpoint airport is not slot-controlled (free) the FARE will decrease on average compared to a controlled slot by $18.11, while the other predictors are held constant.
- GATE: If the endpoint airport does not have gate constraints (free) the FARE will decrease on average compared to a constrained gate by $19.79, while the other predictors are held constant.
- DISTANCE: A one unit increase (one mile) in DISTANCE will result in an increase in FARE on average of $0.08, while the other predictors are held constant.
- PAX: A one unit increase in PAX will result in a decrease in FARE on average of $0.001, while the other predictors are held constant.

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)

```
residual_plot <- function(model, data, xlabel, title) {
    data <- model %>%
        augment(data)
    g <- ggplot(data, aes(x=.pred, y=.resid)) +
        geom_point(alpha=0.5) +
        geom_smooth() +
        labs(x=xlabel, y="Residual", title=title)
    return (g)
}
```

```
g1 <- residual_plot(lm_model, train, "Predicted FARE", "Linear regression") +
    coord_cartesian(ylim=c(-110, 130))
g1
```

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



* We can see a pattern in our residual plot, which indicates that this model is not sufficient in it's current form.

```
calculate_metrics <- function(model, train, test, model_name) {
    bind_rows(
        bind_cols(
            model=model_name,
            dataset="train",
            metrics(model %>% augment(train), truth=FARE, estimate=.pred),
        ),
        bind_cols(
            model=model_name,
            dataset="test",
            metrics(model %>% augment(test), truth=FARE, estimate=.pred),
        ),
    )
}

metrics_table <- function(metrics, caption) {
  metrics %>%
      pivot_wider(names_from=.metric, values_from=.estimate) %>%
      select(-.estimator) %>%
      knitr::kable(caption=caption, digits=3) %>%
      kableExtra::kable_styling(full_width=FALSE)
}
```

```
full_metrics<-calculate_metrics(lm_model, train, test, "Linear regression")
full_metrics
```

```
## # A tibble: 6 x 5
##   model             dataset .metric .estimator .estimate
##   <chr>             <chr>   <chr>   <chr>          <dbl>
## 1 Linear regression train   rmse    standard      34.9
## 2 Linear regression train   rsq     standard       0.795
## 3 Linear regression train   mae     standard      27.4
## 4 Linear regression test    rmse    standard      35.8
## 5 Linear regression test    rsq     standard       0.760
## 6 Linear regression test    mae     standard      27.8
```

* This model is not overfitting the data. We would know that it was overfitted if the model has a small training RMSE and a large test RMSE. In this case the RMSE is about the same for both the training and the test data sets. Generally this model doesn't do a great job, but not a horrible job representing and predicting our data as the R squared values are 0.79 and 0.76 out of a range of 0 to 1. This indicates that about 76-79% of the variance in FARE can be explained by our model.

(iii.) Taking the results from the (b) and (c.ii) into account, build a model that includes only the most important predictors. Determine the model performance and compare with the full model from (c.ii).

- From part b I concluded that the variables that may play the largest role in FARE are DISTANCE, and E_INCOME. COUPON also played a large role in FARE, but was highly correlated with DISTANCE, so I have left it out.

- From c.ii we can see that there is a large difference between vacation destination or not, Southwest or not, if the gate is constricted or free, and if the slot is controlled or free.

```
reduced_formula <- FARE ~  VACATION + SW + E_INCOME + SLOT + GATE + DISTANCE

reduced_lm_spec <- linear_reg(engine="lm", mode="regression")
reduced_lm_model <- reduced_lm_spec %>% fit(reduced_formula, data=train)
```

```
reduced_lm_model %>%
    tidy() %>%
    knitr::kable(digits=3, caption="Linear regression model parameters extracted using the tidy functio
```

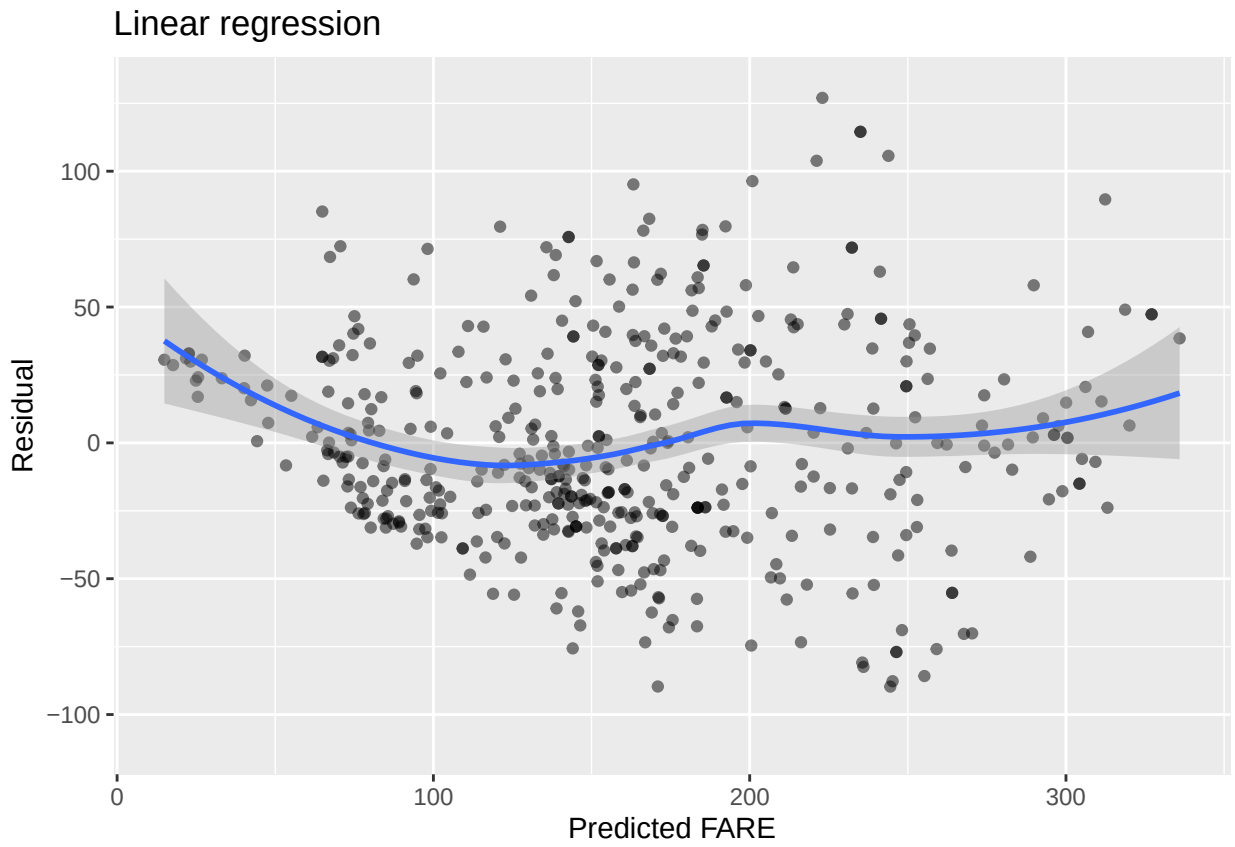
Table 3: Linear regression model parameters extracted using the tidy function

term	estimate	std.error	statistic	p.value
(Intercept)	111.190	13.172	8.441	0.000
VACATIONYes	-48.003	4.047	-11.862	0.000
SWYes	-48.587	4.541	-10.700	0.000
E_INCOME	0.001	0.000	3.347	0.001
SLOTFree	-17.176	4.269	-4.023	0.000
GATEFree	-26.027	4.787	-5.437	0.000
DISTANCE	0.073	0.003	25.553	0.000

```
g2<-residual_plot(reduced_lm_model, train, "Predicted FARE", "Linear regression") +
    coord_cartesian(ylim=c(-110, 130))
```

```
g2
```

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



- We can see a pattern in our residual plot, which indicates that this model is not sufficient in it's current form.

```
reduced_metrics<-calculate_metrics(reduced_lm_model, train, test, " Reduced linear regression")
reduced_metrics
```

```
## # A tibble: 6 x 5
##   model                          dataset .metric .estimator .estimate
##   <chr>                          <chr>   <chr>   <chr>          <dbl>
## 1 " Reduced linear regression" train   rmse    standard      38.7
## 2 " Reduced linear regression" train   rsq     standard       0.748
## 3 " Reduced linear regression" train   mae     standard      30.9
## 4 " Reduced linear regression" test    rmse    standard      39.8
## 5 " Reduced linear regression" test    rsq     standard       0.701
## 6 " Reduced linear regression" test    mae     standard      30.7
```

```
metrics<- bind_rows(
  calculate_metrics(lm_model,train,test,"Linear regression"),
  calculate_metrics(reduced_lm_model,train,test,"Reduced linear regression")
)
metrics_table(metrics, "Metrics for regression models")
```

Table 4: Metrics for regression models

model	dataset	rmse	rsq	mae
Linear regression	train	34.937	0.795	27.442
Linear regression	test	35.803	0.760	27.778
Reduced linear regression	train	38.729	0.748	30.869
Reduced linear regression	test	39.845	0.701	30.676

- With our predictors as VACATION, SW, E_INCOME, SLOT, GATE, DISTANCE we are seeing worse performance than from our original full model, as there are higher values for RMSE and MAE in the reduced model. R squared is lower for the reduced model than the full model, indicating that the amount of variance in our response variable accounted for by our model has decreased from the full model to reduced model.

(iv.) Using the models from (d.ii) and (d.iii), 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.

- Dealt with the categorical variables by including the stringAsFactors argument and set to TRUE.

Create new dataframe with variable values we want to base our prediction on:

```
prediction_values<-data.frame(COUPON <- c(1.202),
                              NEW <- c(3),
                              VACATION <- c('No'),
                              SW<-c('No'),
                              HI<-c(4442.141),
                              S_INCOME <- c(28760),
                              E_INCOME<-c(27664),
                              S_POP <- c(4557004),
                              E_POP <- c(3195503),
                              SLOT <- c('Free'),
                              GATE<-c('Free'),
                              PAX<-c(12782),
                              DISTANCE<-c(1976),
                              stringsAsFactors=TRUE)
```

```
full_preds<-predict(lm_model, prediction_values)
full_preds
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  249.
```

* The predicted FARE is $249.10 with the given predictor values for the full model.

```
reduced_preds<-predict(reduced_lm_model,prediction_values)
reduced_preds
```

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

- The predicted FARE is $250.77 with the given predictor values for the reduced model.

(v.) Using model (iii), predict the reduction in average fare on the route in (iv) if Southwest decides to cover this route.

```
sw_preds<-data.frame(COUPON <- c(1.202),
                             NEW <- c(3),
                             VACATION <- c('No'),
                             SW<-c('Yes'),
                             HI<-c(4442.141),
                             S_INCOME <- c(28760),
                             E_INCOME<-c(27664),
                             S_POP <- c(4557004),
                             E_POP <- c(3195503),
                             SLOT <- c('Free'),
                             GATE<-c('Free'),
                             PAX<-c(12782),
                             DISTANCE<-c(1976),
                             stringsAsFactors=TRUE)
```

```
sw_pred_fare<-predict(reduced_lm_model,sw_preds)
sw_pred_fare
```

```
## # A tibble: 1 x 1
##   .pred
##   <dbl>
## 1  202.
```

- Utilizing the reduced model and changing SW from 'No' to 'Yes' resulted in a change of 250.7668 - 202.1799 = 48.5869.

(e.) 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?

- I think that NEW, HI, and PAX would not be available to predict the average fare from a new airport until the routes start operating.

- NEW - you can't calculate the addition of flights for a specific time period that has already passed for the airport just opening. This would also be skewed as we would be starting with no carriers and adding a bunch all at once for the opening of an airport.

- HI - the measure of concentration, I don't think this is measurable until airplanes are taking off and coming in.

- PAX - passengers on that route during period of data collection; I think this could be estimated for number of passengers they suspect will take each route, but the during the period of data collection would indicate that it is for that specific time period which has past already.

(i.) Train a model that includes only factors that are available before flights begin to operate on the new route.

```
preflight_formula <- FARE ~ COUPON + VACATION + SW + S_INCOME + E_INCOME + S_POP + E_POP + SLOT + GATE +

preflight_lm_spec <- linear_reg(engine="lm", mode="regression")
preflight_lm_model <- lm_spec %>% fit(preflight_formula, data=train)
```

(ii.) Compare the predictive accuracy of this model with models from (d). Is this model good enough, or is it worthwhile reevaluating the model once flights begin on the new route?

```
metrics2<- bind_rows(
  calculate_metrics(lm_model,train,test,"Linear regression"),
  calculate_metrics(reduced_lm_model,train,test,"Reduced linear regression"),
  calculate_metrics(preflight_lm_model,train,test,"Preflight linear regression")
)
metrics_table(metrics2, "Metrics for the four regression models")
```

Table 5: Metrics for the four regression models

model	dataset	rmse	rsq	mae
Linear regression	train	34.937	0.795	27.442
Linear regression	test	35.803	0.760	27.778
Reduced linear regression	train	38.729	0.748	30.869
Reduced linear regression	test	39.845	0.701	30.676
Preflight linear regression	train	38.553	0.750	30.830
Preflight linear regression	test	39.834	0.701	30.628

- Currently we see that the exclusion of NEW, HI, and PAX for our preflight model does about the same as our reduced model, which are both not as good as the full model. We see this as the RMSE and MAE are lower in our full model and the R squared is higher for our full model. for This means that this model is not good enough and it would be better to collect data once the airport is running and we can run the full model to predict FARE.