Final Version of the Second Assignment

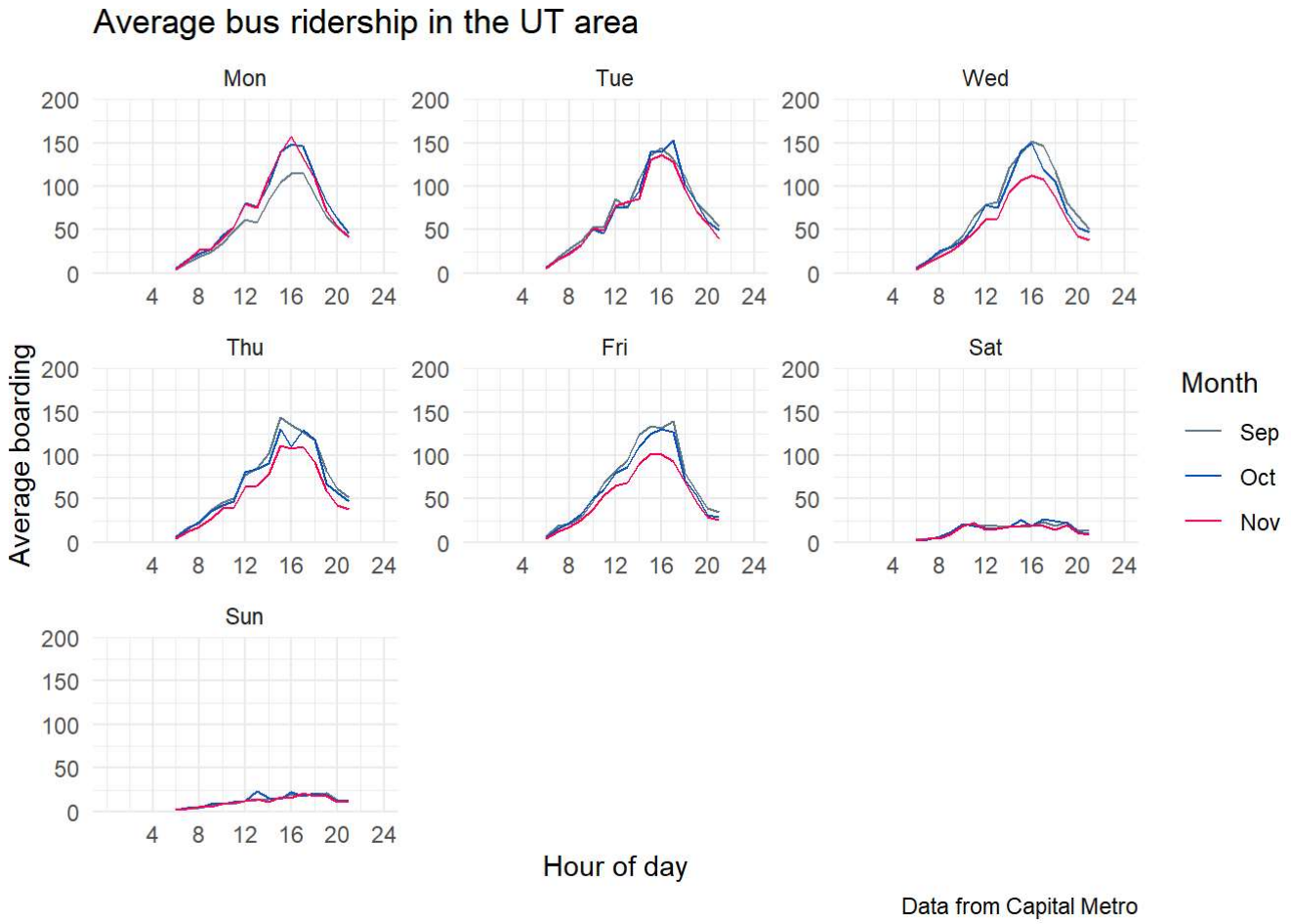
Fjolle Gjonbalaj

12/03/2021

PROBLEM 1

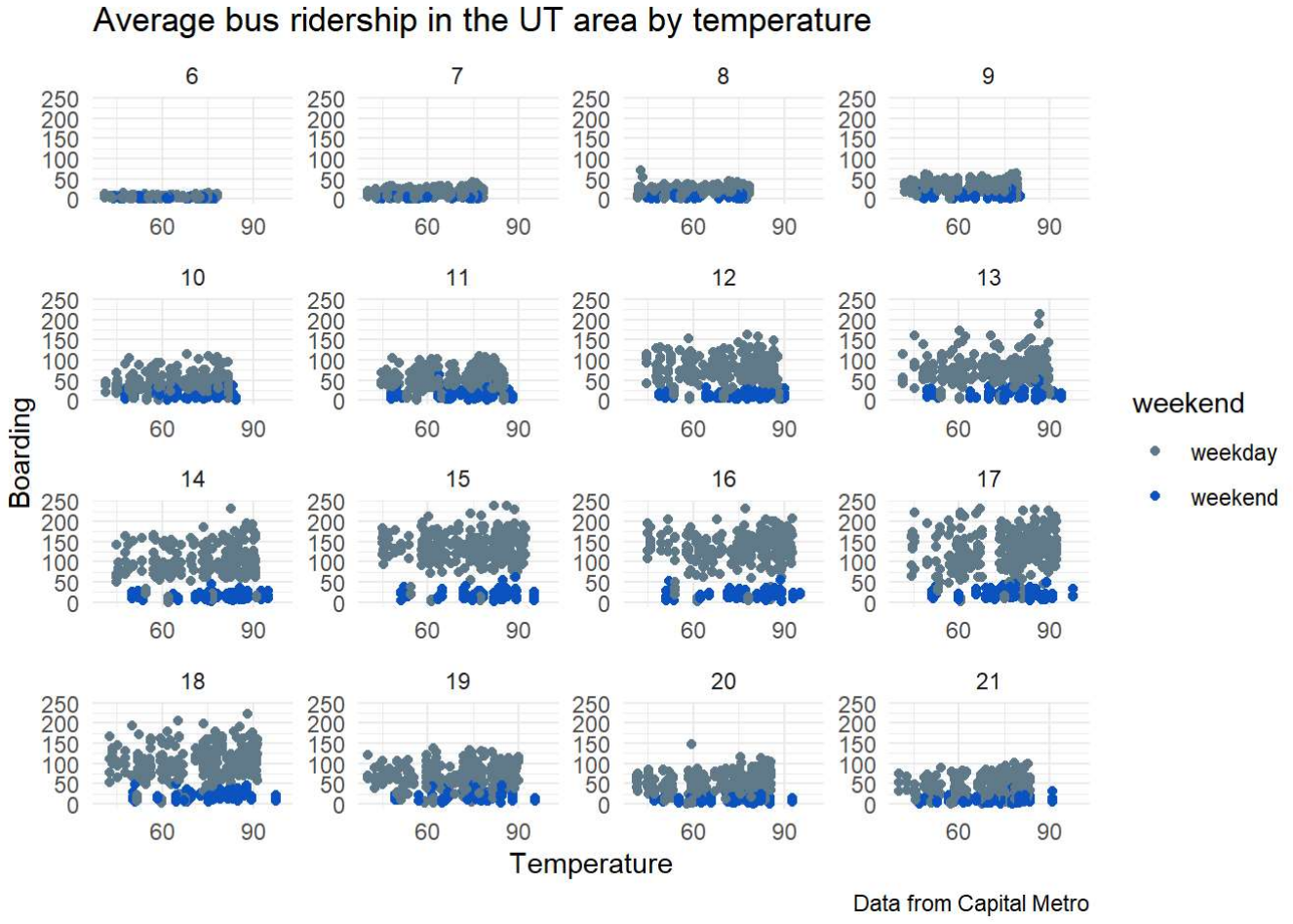
```
CapMetro1 = mutate(CapMetro,
                   day_of_week = factor(day_of_week,
                                        levels=c("Mon", "Tue", "Wed","Thu", "Fri", "Sat",
"Sun")),
                   month = factor(month,
                                  levels=c("Sep", "Oct","Nov")))
```

```
  CapMetro1 %>%
  group_by(hour_of_day, day_of_week, month) %>%
  mutate(avgboard = mean(boarding)) %>%
  ungroup() %>%  ggplot() +
  geom_line(aes(x = hour_of_day, y = avgboard, color = month)) +
  scale_x_continuous(limits = c(0, 24),breaks = seq(4, 24, 4)) +
  scale_y_continuous(expand = c(0,0), limits = c(0, 200)) +
  scale_color_ft("Month") +
  facet_wrap(~ day_of_week, scales = "free") +
  labs(x = "Hour of day", y = "Average boarding",
       title = "Average bus ridership in the UT area",
       caption = "Data from Capital Metro")+
  theme_minimal()
```



The panel of line graphs shows the average bus ridership in the UT area grouped by hour of day, day of week and month I facet by day of week to check whether or not the hour of peak boardings changes from day to day. I observe that the hour of peak boardings is fairly similar across week days, but on Saturdays and Sundays this peak boarding hour is quite rather constant and similar throughout the day.From the line graphs it can be observed that average boardings on Mondays in September are lower, compared to other days and months. This is likely due to the Labor day being a day off on September 7th. On the other hand, Austin does not have any Federal Holidays on Mondays during the October and November months. Similarly, I observe that average boardings on Wednesday, Thursday and Friday on November is lower than for other months. This is likely due to Veteran's day, Thanksgiving day and Day after Thanksgiving being days off.

```
CapMetro1 %>%
   group_by(timestamp, hour_of_day) %>%
   mutate(avg_boarding = mean(boarding)) %>%
   ggplot() +
   geom_point(aes(x = temperature, y = avg_boarding, color = weekend)) +
   scale_x_continuous(limits = c(40, 100),
                      breaks = seq(30, 90, 30)) +
   scale_y_continuous(limits = c(0, 240)) +
   scale_color_ft() +
   facet_wrap(. ~ hour_of_day, scales = "free") +
   labs(x = "Temperature", y = "Boarding",
        title = "Average bus ridership in the UT area by temperature",
        caption = "Data from Capital Metro") +
  theme_minimal()
```



I use a facet of scatter plots to show the effect of temperature (x) on average bus ridership in the UT area. I facet by hour of the day, with points colored according to whether it is a weekday or weekend. I aim to show whether or not temperature has a noticeable effect on the number of UT students riding the bus. I do not

observe any noticeable effects of temperature on ridership, regardless of whether it is a weekday or a weekend.

PROBLEM 2

```
data(SaratogaHouses)
x=matrix(c("Benchmark Model  (Medium Model)",
           "price = lotSize + age + livingArea + pctCollege + bedrooms + fireplaces+bathroom
s + rooms + heating + fuel + centralAir",
           "Price Model  (Manually Built Model)",
           "price = rooms + bathrooms + bathrooms*rooms + lotSize + newConstruction+livingAr
ea + livingArea*rooms + lotSize * livingArea + pctCollege + heating + fuel + livingArea * (he
ating + fuel) + centralAir + waterfront",
           "Price Model 2 (Forward Selected Model)",
           "price = livingArea + landValue + bathrooms + waterfront + newConstruction + heat
ing + lotSize + age + centralAir + rooms + bedrooms + landValue * newConstruction + bathrooms
* heating + livingArea * bathrooms + lotSize * age + livingArea * waterfront + landValue * lo
tSize + livingArea * centralAir + age * centralAir + livingArea * landValue + bathrooms * bed
rooms + bathrooms * waterfront + heating * bedrooms + heating * rooms + waterfront * centralA
ir + waterfront * lotSize + landValue * age + age * rooms + livingArea * lotSize + lotSize *
 rooms + lotSize * centralAir",
           "Price Model 3 (Forward Selected Model)",
           "price = livingArea + bathrooms + waterfront + newConstruction + heating+lotSize
 + age + centralAir + rooms + bedrooms + bathrooms * heating + livingArea * bathrooms + lotSi
ze * age + livingArea * waterfront + livingArea * centralAir + age * centralAir + bathrooms *
bedrooms + bathrooms * waterfront + heating * bedrooms + heating * rooms + waterfront * centr
alAir + waterfront * lotSize + age * rooms+livingArea * lotSize + lotSize * rooms + lotSize *
centralAir")
          , nrow=8, ncol=1)
 kable(x, caption="**Table 1.1 : Models With Price As the Target Variable**")%>%
   kable_styling(position="center", full_width = NULL)
```

Table 1.1 : Models With Price As the Target Variable

Benchmark Model (Medium Model)
price = lotSize + age + livingArea + pctCollege + bedrooms + fireplaces+bathrooms + rooms + heating + fuel + centralAir
Price Model (Manually Built Model)
price = rooms + bathrooms + bathrooms*rooms* + *lotSize* + *newConstruction+livingArea* + *livingArea*rooms + lotSize * livingArea + pctCollege + heating + fuel + livingArea * (heating + fuel) + centralAir + waterfront
Price Model 2 (Forward Selected Model)
price = livingArea + landValue + bathrooms + waterfront + newConstruction + heating + lotSize + age + centralAir + rooms + bedrooms + landValue * newConstruction + bathrooms * heating + livingArea * bathrooms + lotSize * age + livingArea * waterfront + landValue * lotSize + livingArea * centralAir + age * centralAir + livingArea * landValue + bathrooms * bedrooms + bathrooms * waterfront + heating * bedrooms + heating * rooms + waterfront * centralAir + waterfront * lotSize + landValue * age + age * rooms + livingArea * lotSize + lotSize * rooms + lotSize * centralAir
Price Model 3 (Forward Selected Model)

price = livingArea + bathrooms + waterfront + newConstruction + heating+lotSize + age + centralAir + rooms + bedrooms + bathrooms * heating + livingArea * bathrooms + lotSize * age + livingArea * waterfront + livingArea * centralAir + age * centralAir + bathrooms * bedrooms + bathrooms * waterfront + heating * bedrooms + heating * rooms + waterfront * centralAir + waterfront * lotSize + age * rooms+livingArea * lotSize + lotSize * rooms + lotSize * centralAir

```
rmse = function(y, yhat) {
  sqrt( mean( (y - yhat)^2 ) )}
```

```
 LoopRMSE = do(100)*{
   n = nrow(SaratogaHouses)
   n_train = round(0.8*n)
   n_test = n - n_train
   train_cases = sample.int(n, n_train, replace=FALSE)
   test_cases = setdiff(1:n, train_cases)
   saratoga_train = SaratogaHouses[train_cases,]
   saratoga_test = SaratogaHouses[test_cases,]

   lm_medium = lm(price ~ lotSize + age + livingArea + pctCollege + bedrooms +
                    fireplaces + bathrooms + rooms + heating + fuel + centralAir, data=sarato
ga_train)
   #improved model for price that includes land value
   lm_price = lm(price ~ rooms + bathrooms + bathrooms*rooms  + lotSize + newConstruction
                 + livingArea + livingArea*rooms + lotSize*livingArea + pctCollege + heating
 + fuel
                 +livingArea*(heating + fuel) + centralAir + waterfront
                 ,data=saratoga_train)

   lm_price1 = lm(price ~ livingArea + landValue + bathrooms + waterfront + newConstruction +
                    heating + lotSize + age + centralAir + rooms + bedrooms +
                    landValue:newConstruction + bathrooms:heating + livingArea:bathrooms +
                    lotSize:age + livingArea:waterfront + landValue:lotSize +
                    livingArea:centralAir + age:centralAir + livingArea:landValue +
                    bathrooms:bedrooms + bathrooms:waterfront + heating:bedrooms +
                    heating:rooms + waterfront:centralAir + waterfront:lotSize +
                    landValue:age + age:rooms + livingArea:lotSize + lotSize:rooms +
                    lotSize:centralAir, data=saratoga_train)


   lm_price2 = lm(price ~ livingArea +  bathrooms + waterfront + newConstruction + heating +
 lotSize + age +
                    centralAir + rooms + bedrooms + bathrooms:heating + livingArea:bathrooms
 +
                    lotSize:age + livingArea:waterfront +  livingArea:centralAir + age:centra
lAir  +
                    bathrooms:bedrooms + bathrooms:waterfront + heating:bedrooms + heating:ro
oms +
                    waterfront:centralAir + waterfront:lotSize + age:rooms + livingArea:lotSi
ze + lotSize:rooms +
                    lotSize:centralAir
                  ,data=saratoga_train)



   yhat_test_medium = predict(lm_medium, saratoga_test)
   yhat_test_price = predict(lm_price, saratoga_test)
   yhat_test_price1 = predict(lm_price1, saratoga_test)
   yhat_test_price2 = predict(lm_price2, saratoga_test)


   c(RmseMedium=rmse(saratoga_test$price, yhat_test_medium),
     rmsePrice =rmse(saratoga_test$price, yhat_test_price),
     RmsePrice1 = rmse(saratoga_test$price, yhat_test_price1),
     RmsePrice2 = rmse(saratoga_test$price, yhat_test_price2))

 }
```

```
RMSEMean = rbind("Baseline Model 1 " = mean(LoopRMSE$RmseMedium),
                 "Price Model 1 " = mean(LoopRMSE$rmsePrice),
                 "Price Model 2 " = mean(LoopRMSE$RmsePrice1),
                 "Price Model 2.1 " = mean(LoopRMSE$RmsePrice2))
kable(RMSEMean, caption="**Table 1.2 : RMSE for Price Models of Step 1**")%>%
  kable_styling(full_width = FALSE)%>%
  column_spec(1, width = "10em")
```

Table 1.2 : RMSE for Price Models of Step 1

Baseline Model 1	66752.14
Price Model 1	63648.43
Price Model 2	58552.86
Price Model 2.1	64718.47

Here I attempt at building the best linear model for price from the benchmark model specified in the very first lines of the code by including several interaction variables and features that have an effect on price. I obtain repeated results that the Price Model 2 is the best linear model and it beats the benchmark model by a significant amount by having a significantly lower rmse.

KNN regression

Mean rmse for the medium model :

```
## [1] 65688.29
```

Mean rmse for the best linear model:

```
## [1] 57047.62
```

Mean rmse with k nearest neighbors :

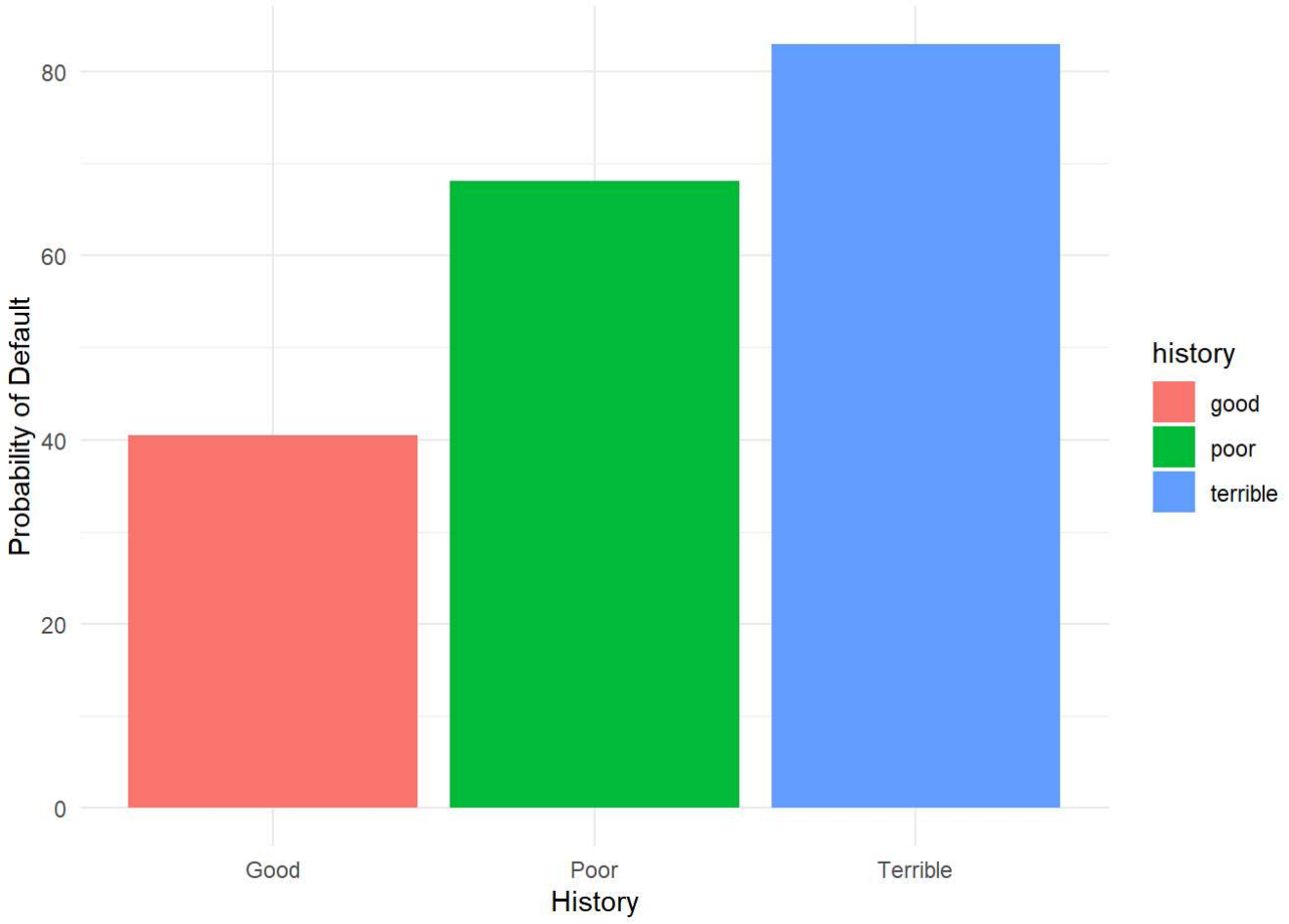
```
##            k     RMSE
## result.1   2 67051.99
## result.2   5 62231.51
## result.3  10 61555.09
## result.4  20 61496.46
## result.5  50 62766.15
## result.6  75 63731.47
## result.7 100 64445.73
## result.8 150 65841.50
## result.9 200 67008.07
```

In comparison to the linear models with price as a target variable, the price KNN regression has a higher minimum average out-of-sample RMSE for some K values. The linear regression model is better than the KNN model at predicting market values.

PROBLEM 3

```
GermanCredit<-read.csv("https://raw.githubusercontent.com/jgscott/ECO395M/master/data/german_credit.csv")
```

```
GermanCredit %>%
  group_by(Default, history) %>%
  add_tally() %>%
  rename(num_default = n) %>%
  distinct(history, num_default) %>%
  ungroup() %>%
  group_by(history) %>%
  mutate(tot_default = sum(num_default),
         prob_default = (num_default / tot_default) * 100) %>%
  filter(Default == 0) %>%
  ggplot() + geom_col(aes(x = history, y = prob_default,
              fill = history))  +scale_x_discrete(labels = c("Good", "Poor", "Terrible")) +
  labs(x = "History", y = "Probability of Default") +  theme_minimal()
```



The outcome variable of interest in this data set is default: a 0/1 indicator variable for whether or not a loan fell into default at some point before it was paid back to the bank. Of particular interest here is the “credit history” variable (history), in which a borrower’s credit rating is classified as “Good”, “Poor,” or “Terrible.”

I make a bar plot to show the probability of default by credit history. The bar plot shows what we would expect to see. That is, the probability of of default is lower for individuals with a credit history classified as “Good”, higher for individuals with a “Poor” credit history, and highest for those with a “Terrible” credit history.

```
credit_split <- initial_split(GermanCredit, strata = "Default", prop = 0.8)
credit_train <- training(credit_split)
credit_test  <- testing(credit_split)
LogisticModel <- glm(Default ~ duration + amount + installment+history + age + purpose + fore
ign , family=binomial, data = credit_train)
summary(LogisticModel)
```

```
##
## Call:
## glm(formula = Default ~ duration + amount + installment + history +
##     age + purpose + foreign, family = binomial, data = credit_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3963  -0.7956  -0.5599   0.9908   2.4481
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.058e+00  5.358e-01  -1.975 0.048245 *
## duration             3.362e-02  9.355e-03   3.594 0.000326 ***
## amount               9.504e-05  4.261e-05   2.231 0.025709 *
## installment          2.182e-01  8.730e-02   2.500 0.012416 *
## historypoor         -1.055e+00  2.720e-01  -3.878 0.000105 ***
## historyterrible     -1.813e+00  3.126e-01  -5.800 6.63e-09 ***
## age                 -2.094e-02  8.191e-03  -2.556 0.010592 *
## purposeedu           9.203e-01  4.140e-01   2.223 0.026233 *
## purposegoods/repair  2.929e-01  3.037e-01   0.964 0.334802
## purposenewcar        9.816e-01  3.247e-01   3.023 0.002499 **
## purposeusedcar      -6.238e-01  4.058e-01  -1.537 0.124227
## foreigngerman       -1.130e+00  6.316e-01  -1.789 0.073623 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 980.50  on 801  degrees of freedom
## Residual deviance: 848.79  on 790  degrees of freedom
## AIC: 872.79
##
## Number of Fisher Scoring iterations: 5
```

Further, I build a logistic regression model for predicting default probability, using the variables duration + amount + installment + age + history + purpose + foreign. We see that the history variable predicting results is highly significant. On the other hand, the purpose of the borrowing as well as whether the individual is foreign or German are not significant and could potentially be dropped from the model.

The data does not seem to be entirely appropriate for predicting default loans. This is mainly because the data set that is being used is not representative of all loans but rather a number of defaults. The risk of using such data is that the bank runs the risk of misclassifying individuals with a very bad credit history on being good candidates for future loans.

PROBLEM 4

```
hotels<-read.csv("https://raw.githubusercontent.com/jgscott/ECO395M/master/data/hotels_dev.csv")
head(hotels)
```

```
##          hotel lead_time stays_in_weekend_nights stays_in_week_nights adults
## 1   City_Hotel       217                       1                     3      2
## 2   City_Hotel         2                       0                     1      2
## 3 Resort_Hotel        95                       2                     5      2
## 4 Resort_Hotel       143                       2                     6      2
## 5 Resort_Hotel       136                       1                     4      2
## 6   City_Hotel        67                       2                     2      2
##   children meal market_segment distribution_channel is_repeated_guest
## 1        0   BB  Offline_TA/TO                TA/TO                 0
## 2        0   BB         Direct               Direct                 0
## 3        0   BB      Online_TA                TA/TO                 0
## 4        0   HB      Online_TA                TA/TO                 0
## 5        0   HB         Direct               Direct                 0
## 6        0   SC      Online_TA                TA/TO                 0
##   previous_cancellations previous_bookings_not_canceled reserved_room_type
## 1                      0                              0                 A
## 2                      0                              0                 D
## 3                      0                              0                 A
## 4                      0                              0                 A
## 5                      0                              0                 F
## 6                      0                              0                 A
##   assigned_room_type booking_changes deposit_type days_in_waiting_list
## 1                  A               0   No_Deposit                    0
## 2                  K               0   No_Deposit                    0
## 3                  A               2   No_Deposit                    0
## 4                  A               0   No_Deposit                    0
## 5                  F               0   No_Deposit                    0
## 6                  A               0   No_Deposit                    0
##     customer_type average_daily_rate required_car_parking_spaces
## 1 Transient-Party              80.75                         none
## 2       Transient             170.00                         none
## 3       Transient               8.00                         none
## 4       Transient              81.00                         none
## 5       Transient             157.60                         none
## 6       Transient              49.09                         none
##   total_of_special_requests arrival_date
## 1                         1   2016-09-01
## 2                         3   2017-08-25
## 3                         2   2016-11-19
## 4                         1   2016-04-26
## 5                         4   2016-12-28
## 6                         1   2016-03-13
```

```
hotels_val<-read.csv("https://raw.githubusercontent.com/jgscott/ECO395M/master/data/hotels_val.csv")
head(hotels_val)
```

```
##          hotel lead_time stays_in_weekend_nights stays_in_week_nights adults
## 1 Resort_Hotel        47                       0                     2      2
## 2 Resort_Hotel        46                       0                     2      2
## 3   City_Hotel        22                       1                     2      2
## 4 Resort_Hotel       209                       2                     5      2
## 5   City_Hotel         1                       1                     0      2
## 6 Resort_Hotel       171                       2                     5      2
##   children meal market_segment distribution_channel is_repeated_guest
## 1        1   BB         Direct               Direct                 0
## 2        0   BB  Offline_TA/TO                TA/TO                 0
## 3        0   BB      Online_TA                TA/TO                 0
## 4        0   BB  Offline_TA/TO                TA/TO                 0
## 5        0   SC      Online_TA                TA/TO                 0
## 6        0   BB      Online_TA                TA/TO                 0
##   previous_cancellations previous_bookings_not_canceled reserved_room_type
## 1                      0                              0                 C
## 2                      0                              0                 D
## 3                      0                              0                 D
## 4                      0                              0                 A
## 5                      0                              0                 A
## 6                      0                              0                 D
##   assigned_room_type booking_changes deposit_type days_in_waiting_list
## 1                  C               0   No_Deposit                    0
## 2                  D               0   No_Deposit                    0
## 3                  D               0   No_Deposit                    0
## 4                  A               0   No_Deposit                    0
## 5                  B               0   No_Deposit                    0
## 6                  D               2   No_Deposit                    0
##     customer_type average_daily_rate required_car_parking_spaces
## 1       Transient             289.00                        none
## 2       Transient             162.00                        none
## 3       Transient             121.33                        none
## 4       Transient              76.22                        none
## 5       Transient              98.00                        none
## 6 Transient-Party             182.86                        none
##   total_of_special_requests arrival_date
## 1                         1   2017-08-23
## 2                         0   2016-12-30
## 3                         0   2017-03-13
## 4                         0   2016-07-12
## 5                         0   2016-08-07
## 6                         1   2017-07-25
```

The files hotels_dev.csv and hotels_val.csv contains data on tens of thousands of hotel stays from a major U.S.-based hotel chain. The goal of this problem is simple: to build a predictive model for whether a hotel booking will have children on it.

The target variable of interest is children: a dummy variable for whether the booking has children on it.

```
#MODEL BUIDING
hotels_split = initial_split(hotels, prop = 0.8)
hotels_train = training(hotels_split)
hotels_test = testing(hotels_split)
```

Below you can see a logit model, regressed on market segment, adults, customer type, and repeated guest status.

```
base1 = glm(children~market_segment+adults+customer_type+is_repeated_guest, data = hotels_tra
in)
logit_hotels = predict(base1, hotels_test, type='response')
test_logit_hotels = ifelse(logit_hotels > 0.5, 1, 0)
confusion_out_base1 = table(y = hotels_test$children,
                            yhat = test_logit_hotels)
confusion_out_base1
```

```
##    yhat
## y      0
##   0 8274
##   1  725
```

```
sum(diag(confusion_out_base1))/sum(confusion_out_base1)
```

```
## [1] 0.9194355
```

This model has an accuracy rate of ~93% according to a probability threshold of 0.5.

The second base model uses a logistic regression to predicts children based on all other variables except arrival date.

```
base2 = glm(children ~ . - arrival_date, data = hotels_train)
logit_hotels = predict(base2, hotels_test, type='response')
test_logit_hotels = ifelse(logit_hotels > 0.5, 1, 0)
confusion_out_base2= table(y = hotels_test$children,
                            yhat = test_logit_hotels)
confusion_out_base2
```

```
##    yhat
## y      0    1
##   0 8171  103
##   1  472  253
```

```
sum(diag(confusion_out_base2))/sum(confusion_out_base2)
```

```
## [1] 0.936104
```

Using only the data in hotels.dev.csv, I compare the out-of-sample performance of the following models: baseline 1: a small model that uses only the market_segment, adults, customer_type, and is_repeated_guest variables as features. baseline 2: a big model that uses all the possible predictors except the arrival_date variable (main effects only). Moreover, I build two additional models with different interactions and engineering features to improve the performance.

The latter model shows a slightly better accuracy rate than the former model. Hence, adding more variables proved to be beneficial.

In the following model I use the lasso method to decide what variables to use in a linear model.

```
modmat = model.matrix(children ~ .-1, data=hotels) # do -1 to drop intercept!
child = hotels$children
cv = cv.gamlr(modmat, child, nfold = 20, family="binomial")
scbeta = coef(cv)
base3 = lm(children~ hotel+lead_time+adults+meal+market_segment+distribution_channel+
                 is_repeated_guest+previous_bookings_not_canceled+reserved_room_type+
                 booking_changes+customer_type+average_daily_rate+total_of_special_requests+
                 arrival_date, data = hotels)
logit_hotels = predict(base3, hotels_test, type='response')
test_logit_hotels = ifelse(logit_hotels > 0.5, 1, 0)
confusion_out_base3 = table(y = hotels_test$children,
                            yhat = test_logit_hotels)
confusion_out_base3
```

```
##    yhat
## y      0    1
##   0 8172  102
##   1  464  261
```

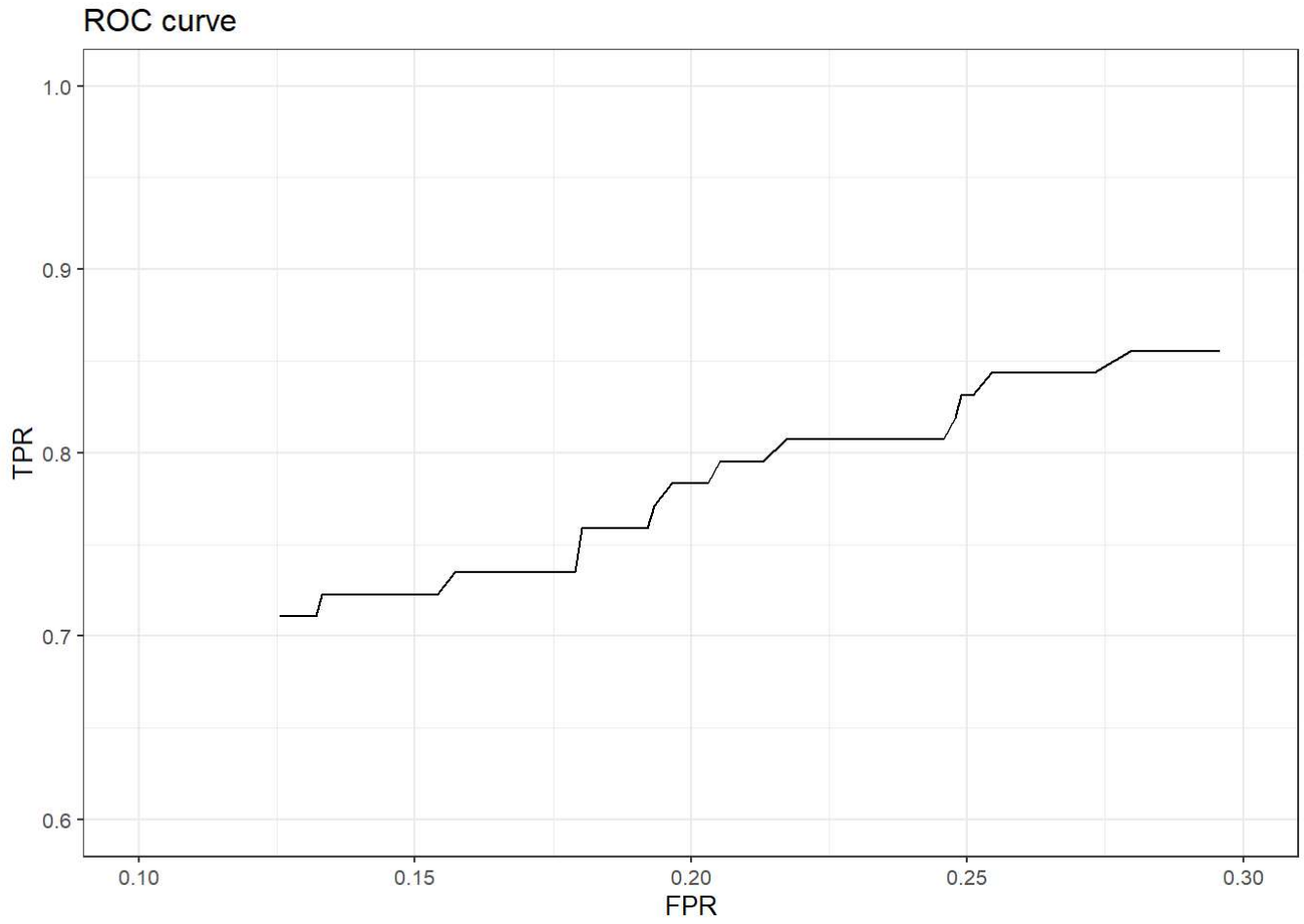
```
sum(diag(confusion_out_base3))/sum(confusion_out_base3)#out-of-sample accuracy
```

```
## [1] 0.9371041
```

Among the three models, the best performing is the linear model. It shows an accuracy rate of approximately 93.1%. Accordingly, I will use the following variables: hotel, lead_time, adults, meal, market_segment, distribution_channel, is_repeated_guest, previous_bookings_not_canceled, reserved_room_type, booking_changes, customer_type, average_daily_rate, total_of_special_requests, and arrival_date.

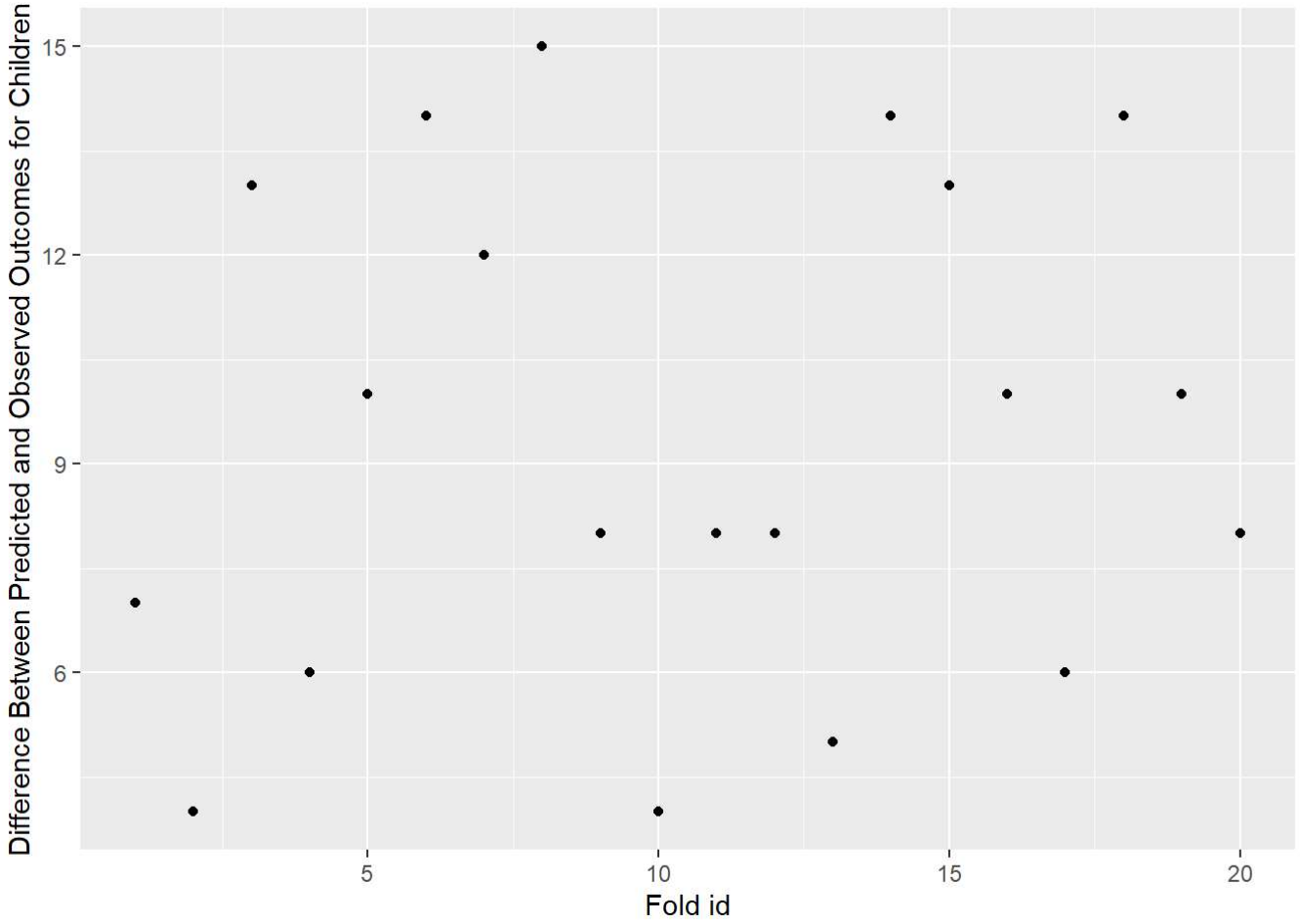
Validation Step 1

Predicting outcomes of the validation data set with my best model:



After I have built the best model and assessed its out-of-sample performance using hotels_dev, now I turn to the data in hotels_val. I validate the model using this entirely fresh subset of the data, i.e. one that wasn't used to fit OR test as part of the model-building stage.

Model Validation: Step 2



The boxplot shows the Difference between predicted and observed outcomes for children for each fold. There is quite a big difference among folds. This suggests that even our best model cannot consistently predict whether or not a child shows up unexpectedly across randomized observations.