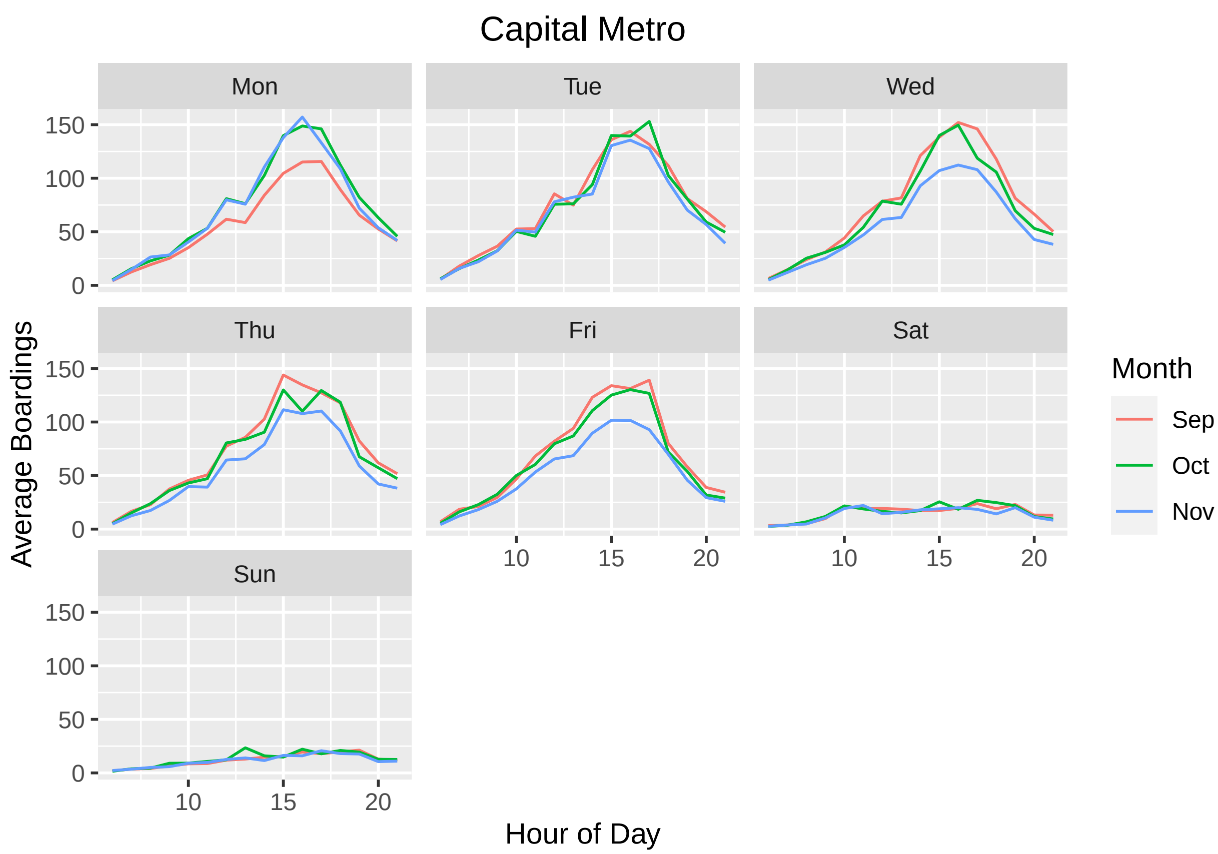
Exercise 2

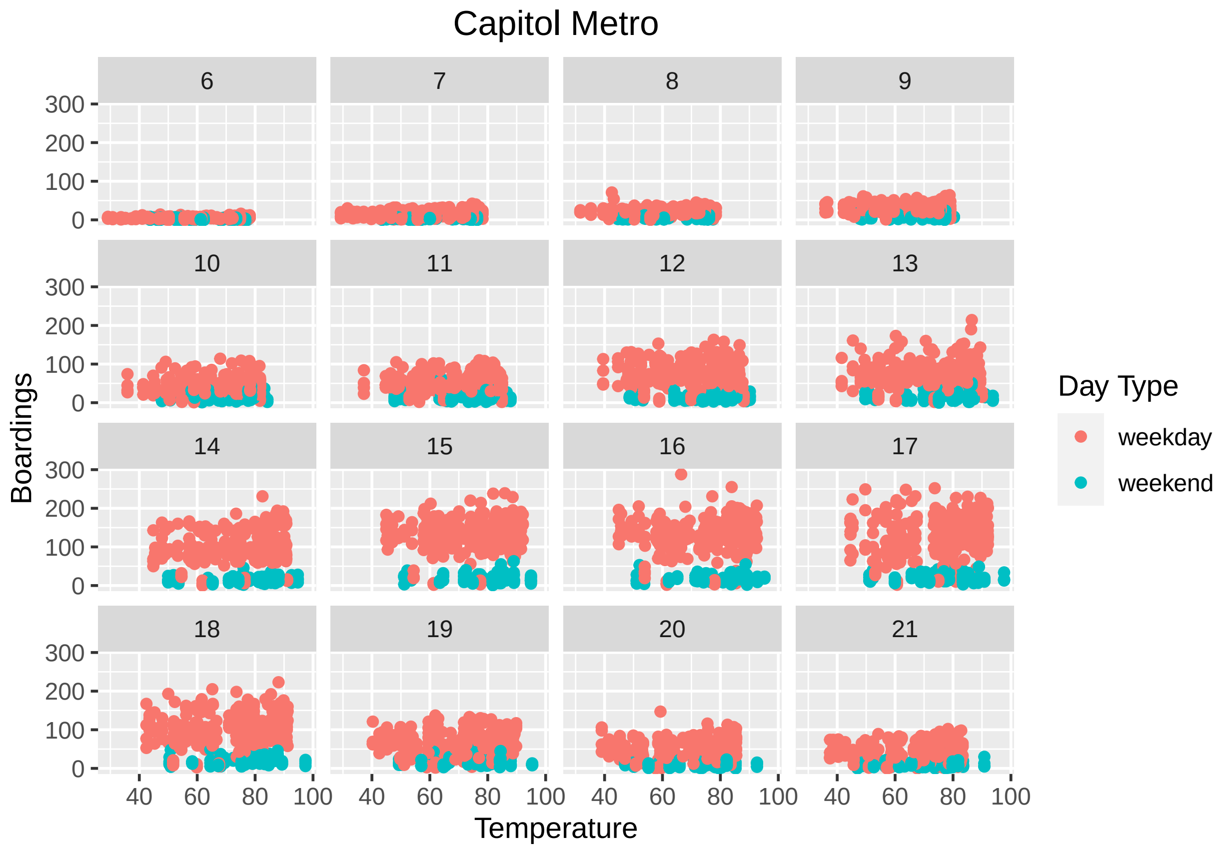
Question 1



Excluding the weekends, peak boardings are broadly similar, which is during evening rush hour.

The average boardings on Monday are lower in September because of Labor Day. Since its a national holiday, the university is closed and UT students would stay home ana would not need to use the bus.

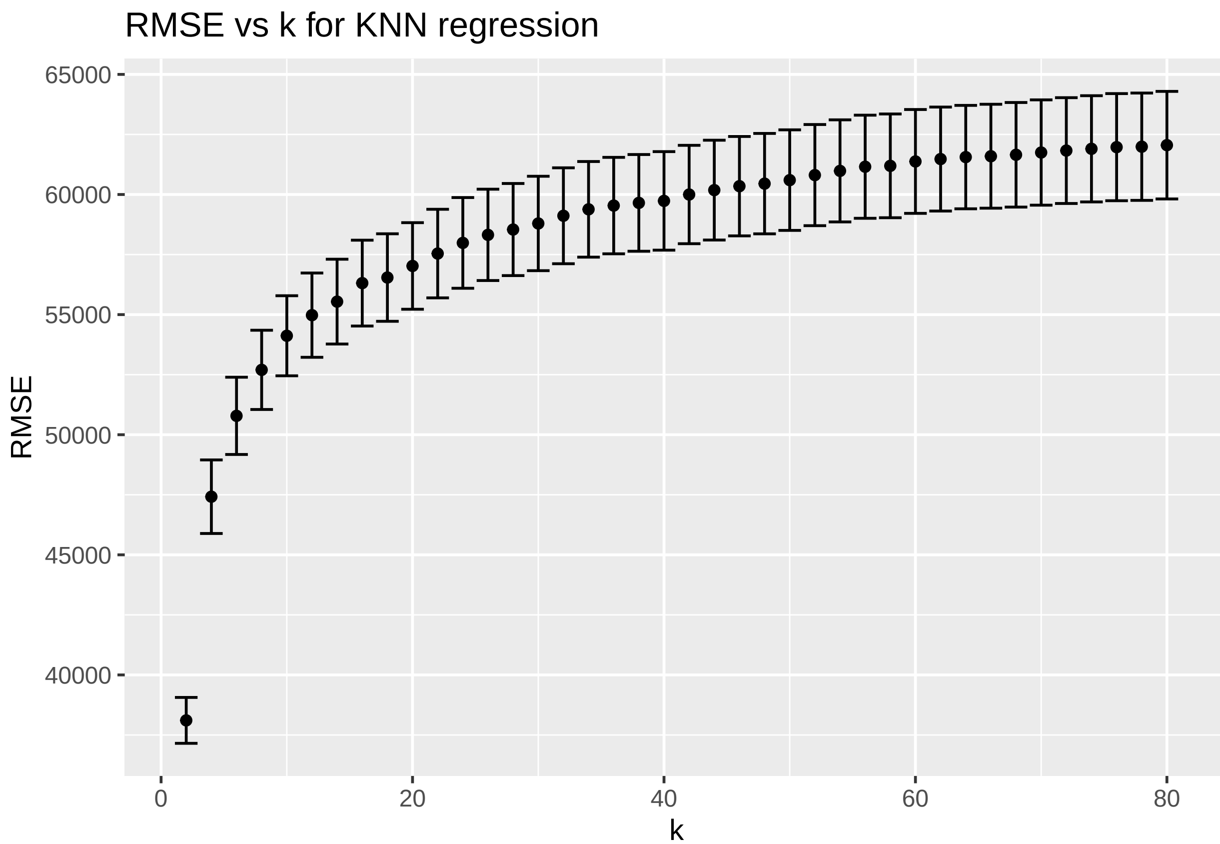
Additionally, average boarding is lower on Wed/Thurs/Fri for November because Thanksgiving lands on a Thursday in November, and UT students would most likely not travel during days of that particular week, but instead spending time with family/friends.



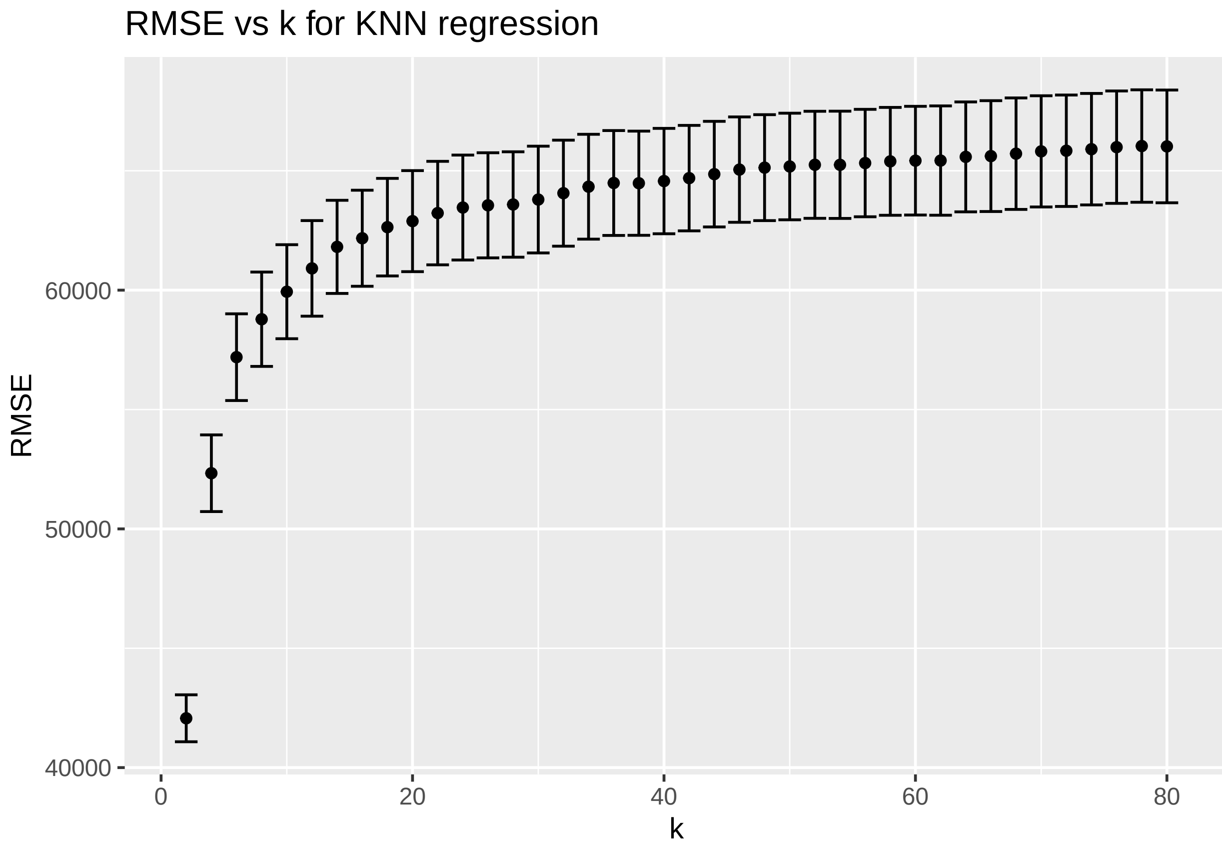
When holding both hour of day and weekend status constant, it does not appear that temperature has a noticeable effect on the number of UT students riding the bus. This is determined by the observable changes in ridership throughout the day. Within each hour of the day, the quantity of boardings remain similar as the temperature changes.

Question 2

Best Linear Model

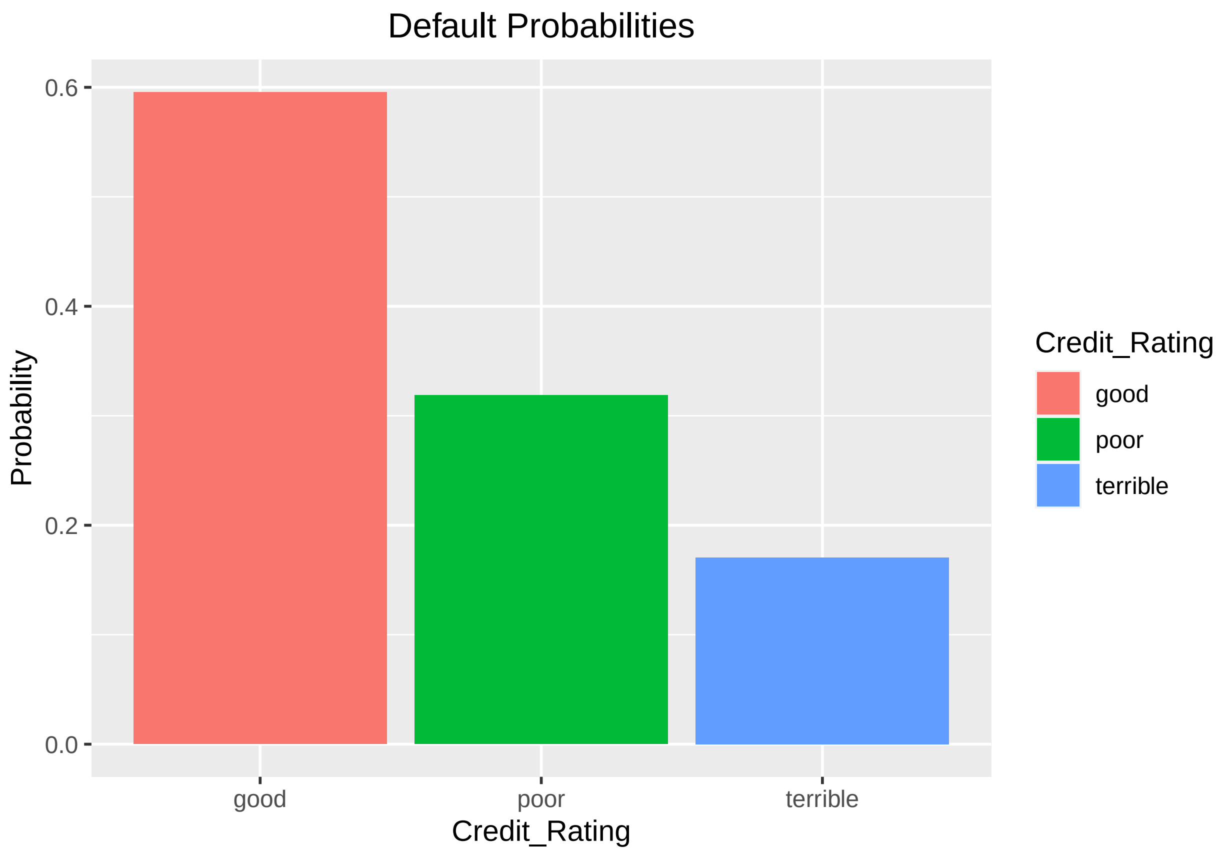


Medium Linear Model



Relative to the Medium Linear Model, the Best Linear Model has both a lower AIC and RMSE, while also a higher Adjusted R-squared. For these reasons, the Best Linear Model is the optimal choice since it seems to do better at achieving lower out-of-sample mean-squared error.

Question 3




```
##         (Intercept)          duration            amount        installment
##       -7.075258e-01      2.525834e-02      9.596288e-05       2.216019e-01
##                 age       historypoor   historyterrible        purposeedu
##       -2.018401e-02     -1.107586e+00     -1.884675e+00       7.247898e-01
## purposegoods/repair     purposenewcar    purposeusedcar      foreigngerman
##        1.049037e-01      8.544560e-01     -7.959260e-01      -1.264676e+00
```

It should be noted that the default probabilities are much lower for individuals with either a poor or terrible credit history relative to individuals with a good history. Although, when observing the glm, individuals with either a poor or terrible credit history have a negative effect in regards to defaulting on a loan. I believe that these conflicting observations are due to the bank having a more strict loan process for individuals with either a poor or terrible credit history. Individuals that are in either category are more likely to repay their loans back on time. In contrast, indivuals with a good credit history are given a more lenient loan process and as a result, have a higher probability in defaulting on their loans.

If the purpose of the model is to screen prospective borrowers to classify them into “high” versus “low” probability of default, then no, I do not believe this data set accomplished its goal. The conclusion from this data set should not be to loan less to individuals to a good credit history. Instead, the end result should be for the bank to apply the same strict loan standards that are applied to others that may have either a poor or terrible credit history. This would give a better source of data and better inform the bank on the probability of default.

Question 4

```
dev = read_csv('/Users/franklinstudent/Desktop/GitHub/Exercise-2/hotels_dev.csv')
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   hotel = col_character(),
##   meal = col_character(),
##   market_segment = col_character(),
##   distribution_channel = col_character(),
##   reserved_room_type = col_character(),
##   assigned_room_type = col_character(),
##   deposit_type = col_character(),
##   customer_type = col_character(),
##   required_car_parking_spaces = col_character(),
##   arrival_date = col_date(format = "")
## )

## See spec(...) for full column specifications.
```

```
dev_split = initial_split(dev, prop = 0.8)
dev_train = training(dev_split)
dev_test = testing(dev_split)


lm_dev1 = lm(children ~ market_segment + adults + customer_type + is_repeated_guest, data = dev_train)
lm_dev2 = lm(children ~ . - arrival_date, data = dev_train)
lm_devbest = lm(children ~ hotel + lead_time+ reserved_room_type + assigned_room_type + booking_changes
```

```
#Baseline 1: Small Model
phat_train_dev1 = predict(lm_dev1, dev_train)
yhat_train_dev1 = ifelse(phat_train_dev1 > 0.5, 1, 0)
confusion_in1 = table(y = dev_train$children, yhat = yhat_train_dev1)
confusion_in1
```

```
##    yhat
## y       0
##   0 33086
##   1  2915
```

```
sum(diag(confusion_in1))/sum(confusion_in1)
```

```
## [1] 0.91903
```

```
phat_test_dev1 = predict(lm_dev1, dev_test)
yhat_test_dev1 = ifelse(phat_test_dev1 > 0.5, 1, 0)
confusion_out1 = table(y = dev_test$children, yhat = yhat_test_dev1)
confusion_out1
```

```
##    yhat
## y      0
##   0 8279
##   1  720
```

```r
sum(diag(confusion_out1))/sum(confusion_out1)
```

```
## [1] 0.9199911
```

```r
#Baseline 2: Big Model
phat_train_dev2 = predict(lm_dev2, dev_train)
yhat_train_dev2 = ifelse(phat_train_dev2 > 0.5, 1, 0)
confusion_in2 = table(y = dev_train$children, yhat = yhat_train_dev2)
confusion_in2
```

```
##    yhat
## y       0     1
##   0 32632   454
##   1  1887  1028
```

```r
sum(diag(confusion_in2))/sum(confusion_in2)
```

```
## [1] 0.934974
```

```r
phat_test_dev2 = predict(lm_dev2, dev_test)
yhat_test_dev2 = ifelse(phat_test_dev2 > 0.5, 1, 0)
confusion_out2 = table(y = dev_test$children, yhat = yhat_test_dev2)
confusion_out2
```

```
##    yhat
## y      0    1
##   0 8158  121
##   1  464  256
```

```r
sum(diag(confusion_out2))/sum(confusion_out2)
```

```
## [1] 0.9349928
```

```r
#Best Linear Model
phat_train_devbest = predict(lm_devbest, dev_train)
yhat_train_devbest = ifelse(phat_train_devbest > 0.5, 1, 0)
confusion_in_devbest = table(y = dev_train$children, yhat = yhat_train_devbest)
confusion_in_devbest
```

```
##    yhat
## y       0     1
##   0 32632   454
##   1  1885  1030
```

```r
sum(diag(confusion_in_devbest))/sum(confusion_in_devbest)
```

```
## [1] 0.9350296
```

```r
phat_test_devbest = predict(lm_devbest, dev_test)
yhat_test_devbest = ifelse(phat_test_devbest > 0.5, 1, 0)
confusion_out_devbest = table(y = dev_test$children, yhat = yhat_test_devbest)
confusion_out_devbest
```

```
##    yhat
## y      0    1
##   0 8168  111
##   1  473  247
```

```r
sum(diag(confusion_out_devbest))/sum(confusion_out_devbest)
```

```
## [1] 0.9351039
```

```r
table(dev_train$children)
```

```
##
##     0     1
## 33086  2915
```

```r
33096/sum(table(dev_train$children))
```

```
## [1] 0.9193078
```

```r
table(dev_test$children)
```

```
##
##    0    1
## 8279  720
```

```r
8269/sum(table(dev_test$children))
```

```
## [1] 0.9188799
```

```r
#abolute improvement
0.9193078 - 0.9188799
```

```
## [1] 0.0004279
```

```r
#The relative improvement
0.9193078/0.9188799
```

```
## [1] 1.000466
```

```r
val = read_csv('/Users/franklinstudent/Desktop/GitHub/Exercise-2/hotels_val.csv')
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   hotel = col_character(),
##   meal = col_character(),
##   market_segment = col_character(),
##   distribution_channel = col_character(),
##   reserved_room_type = col_character(),
##   assigned_room_type = col_character(),
##   deposit_type = col_character(),
##   customer_type = col_character(),
##   required_car_parking_spaces = col_character(),
##   arrival_date = col_date(format = "")
## )

## See spec(...) for full column specifications.
```

```r
logit_val = glm(children ~ hotel + lead_time+ reserved_room_type + assigned_room_type +
                  booking_changes + adults + required_car_parking_spaces + booking_changes +
                  average_daily_rate + is_repeated_guest + arrival_date, data = val, family = 'binomial

coef(logit_val)
```

```
##                   (Intercept)              hotelResort_Hotel
##                  3.903448e+00                  -5.744624e-01
##                     lead_time            reserved_room_typeB
##                  1.611385e-03                   2.296209e+00
##           reserved_room_typeC            reserved_room_typeD
##                  1.605202e+00                  -4.926080e-01
##           reserved_room_typeE            reserved_room_typeF
##                  2.193115e-01                   2.131198e+00
##           reserved_room_typeG            reserved_room_typeH
##                  2.120902e+00                   1.766645e+01
##           assigned_room_typeB            assigned_room_typeC
##                 -2.305046e-01                   2.336804e+00
##           assigned_room_typeD            assigned_room_typeE
##                  9.003248e-01                   7.284416e-01
##           assigned_room_typeF            assigned_room_typeG
##                  8.684185e-01                   1.354173e+00
##           assigned_room_typeH            assigned_room_typeI
##                 -1.377491e+01                   1.887476e+00
##           assigned_room_typeK                booking_changes
##                 -1.196778e+01                   1.776438e-01
##                        adults required_car_parking_spacesparking
##                 -3.561486e-01                   4.323819e-01
##            average_daily_rate              is_repeated_guest
##                  1.247003e-02                  -1.011930e-01
##                  arrival_date
##                 -4.743331e-04
```

```
phat_test_logit_val = predict(logit_val, dev_test, type = 'response')
yhat_test_logit_val = ifelse(phat_test_logit_val > 0.5, 1, 0)
confusion_out_logit= table(y = dev_test$children, yhat = yhat_test_logit_val)
confusion_out_logit
```

```
##    yhat
## y      0    1
##   0 8185   94
##   1  484  236
```