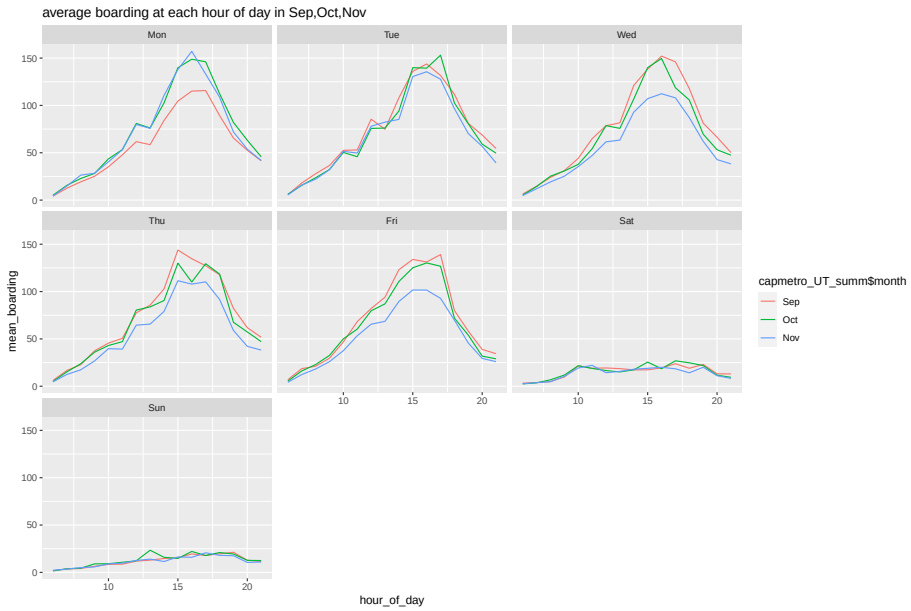
ECO 395 Homework 2:Chengkan Tao

chengkan_tao

```
devtools::install_github(c('rstudio/rmarkdown', 'yihui/tinytex'))
tinytex::install_tinytex()
```

URL:https://github.com/chengkan-Tao/ECO395M-HW2

Q1



The hour of peak boardings is actually fixed. In weekdays, we can find about 16:00 is the hour of peak boardings, and in weekends, it is hard to find hour of peak boarding because the slope of line is almost

flat. The reason why average boardings on Mondays in September look lower may be that some students doesn't have to go to school or the weather is bad at the end of summer. And the holiday or weather may leads to lower average boardings on Weds/Thurs/Fri in November.

If we hold hour of day and weekend status constant, temperature doesn't have obvious effect on boarding. In each figure, average boardings at the same temperature are nearly the same. We can find points with same color are evenly distributed.

##Q2

```
##        V1       V2
## 65718.99 59344.71
```

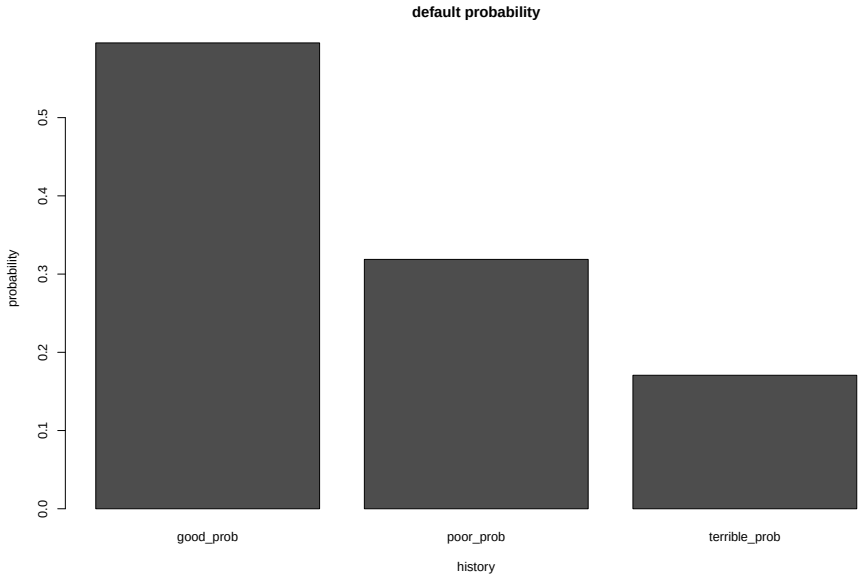
V1—BASELINE V2—STEP SELECTION MODEL

I use log transformation and interaction :logarithm of land value and three way interactions. These are root mean square errors of baseline medium model and step selection model. Evidently, the root mean square error of step selection model is lower than baseline medium model.

```
## [1] 62815.94
```

This is the lowest rmse of knn model. I use three methods to get a best regression model. I find the root mean square error of stepwise selection is the lowest and linear regression is the highest. With log transformation and three way interaction, the stepwise selection model can predict the price with all the variables in the table.Among all the variables, age has a great negative effect on the price, and the larger the livingarea, the higher the price.

##Q3



People with terrible history are more likely to pay up.

```
##          (Intercept)             duration               amount          installment
##              -0.70753              0.02526              0.00010              0.22160
##                  age          historypoor     historyterrible          purposeedu
##              -0.02018              -1.10759             -1.88467              0.72479
## purposegoods/repair       purposenewcar      purposeusedcar       foreigngerman
##              0.10490               0.85446              -0.79593             -1.26468
```

```
exp(-1.10759)
```

```
## [1] 0.3303542
```

```
exp(-1.88467)
```

```
## [1] 0.1518792
```

From coefficient matrix, the odd(p/(1-p)) is multiplied by 0.3303542(0.1518792) if his/her history changes from good to poor(terrible), holding all else fixed. So, this will lead to a drop in the default probability. It is also shown in the figure that good history responds to high default probability. But, it is not reasonable that poor people are more likely to pay up than rich people. The data set is strange and cannot be used to predict because of substantial oversampling of defaults. They find too much all reasonably close matches. People with good history also have some bad conditions. I think random sampling is better.

##Q4 baseline1

```
## lm(formula = children ~ market_segment + adults + customer_type +
##     is_repeated_guest, data = hotels_dev)
```

baseline2

```
## lm(formula = children ~ . - (arrival_date) - (children), data = hotels_dev)
```

my model

```
getCall(lm_forward)
```

```
## lm(formula = children ~ average_daily_rate + market_segment +
##     meal + poly(adults, 2) + customer_type + lead_time + is_repeated_guest +
##     average_daily_rate:market_segment + average_daily_rate:poly(adults,
```

```
##     2) + market_segment:poly(adults, 2) + average_daily_rate:meal +
##     poly(adults, 2):customer_type + average_daily_rate:customer_type +
##     market_segment:customer_type + meal:poly(adults, 2) + average_daily_rate:lead_ti
##     poly(adults, 2):lead_time + market_segment:lead_time + meal:lead_time +
##     customer_type:lead_time + meal:customer_type + average_daily_rate:is_repeated_gu
##     lead_time:is_repeated_guest + poly(adults, 2):is_repeated_guest,
##     data = hotels_dev)
```

```
c(baseline1_rmse,baseline2_rmse,mymodel_rmse)
```

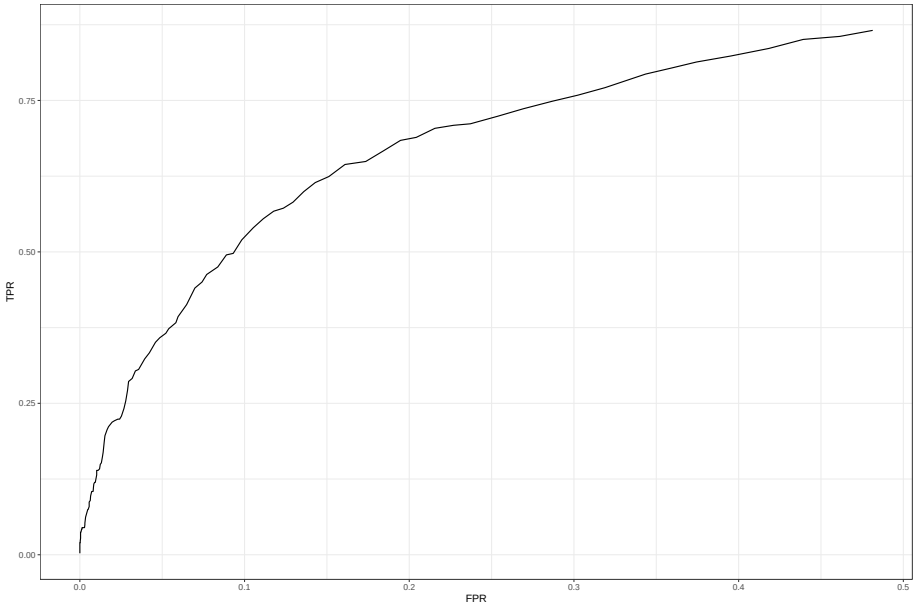
```
## [1] 0.2787395 0.2431506 0.2613746
```

My model doesn't perform well out of sample.

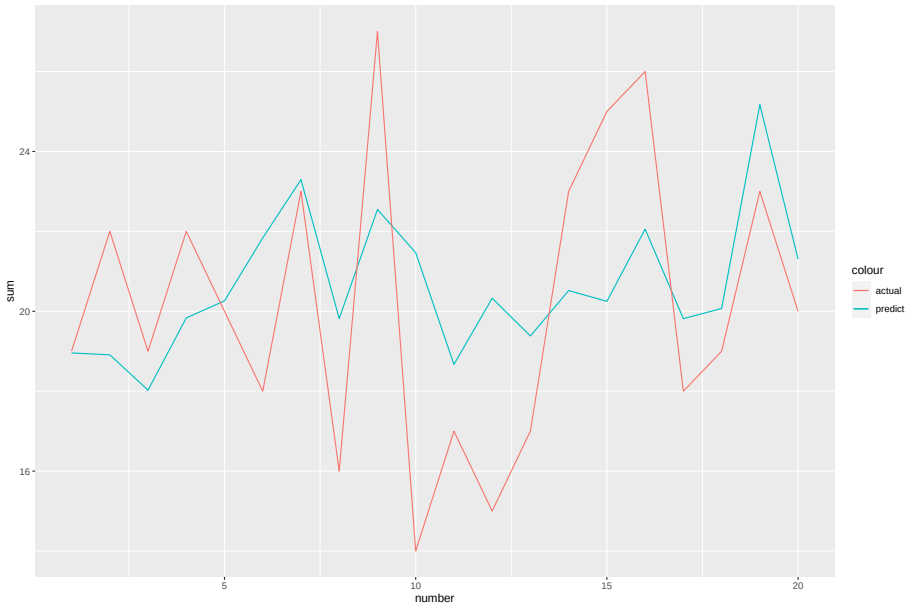
#Model validation: step 1

```
## [1] 0.2528123
```

With new dataset, rmse of new set is smaller than rmse of split set, but still close. The rmse of baseline2 is still the lowest.



#ROC



Not pretty good. The predict is terrible. Although both numbers always move in the same direction, the actual numbers fluctuate more violently than predict probability.