Fair Air Fare

Diptanshu Singh

December 6, 2018

Access the codes at : [github: https://github.com/diptanshu-singh/air_fair]

Abstract

Optimal time to buy the tickets can be viewed as tussle between buyers and sellers. Sellers can analyze huge amount of their past ticket data to judge demand, and model the prices accordingly. Dynamic airline pricing keeps the buyers guessing whether it is right time to book their ticket. The major roadblock that buyers face is lack of past data regarding price trend which is readily available to sellers. This project aims to build a tool that would help buyers make data driven decision on whether it is the right time to buy ticket. In order to make the decision easier, we will try to quantify the effect of waiting by providing the estimated reward and risk involved in waiting for next few days. A multilevel mixed effects model is used to predict price change occurring in the coming days. Based on previous data, effect of factors such as days to departure, day of booking and airline carrier has been used to explain price change. Multiple simulations are run to estimate the price of ticket 'n' days into the future. The implications and limitations for choosing multilevel mixed effects model are also discussed later. The report offers directional inferences to explain fluctuation in the price variance.

Introduction

Background

Generic guidelines to book flights are given by research studies. These guidelines are not optimized according to route and hence will not be applicable in every situation. The project aims to build framework for analyzing each route based on its past data. A data warehouse can be created to store past daily ticket data for flights. While predicting the price trend in coming days, the model will draw inference from change in daily ticket prices over two weeks for a travel period of 3 months. Model validation can then be done to analyze how good the prediction has been. A more informed decision can be made by drawing inferences both from this model build only the particular route in conjecture to a generic model build utilizing data from multiple routes. It is expected the predictions from model build only on a particular route will have less bias and more variance. Whereas the model build on multiple route will have less variance. Weights can be decided based on result from model validation. Currently, the model validation has been done using additional 7 days' data for the same travel duration.

Previous Work

Majority of the work on air fares involve prediction of air fare which is outside the scope of problem this project hopes to solve. Work done on modelling air ticket price is limited to routes or particular flights. As a result, such studies cannot be applied directly to a plane route. Many generic guidelines are also available to users while booking but these don't provide any quantitative support to ease decision making. This project aims to come up with a framework to use past data in order to make predictions for price trend of next few days. The current analysis has been done on one particular route from Boston to Los Angeles.

Methods

Data source

A WebCrawler on a major airline ticketing site has been setup. The crawler can be assessed from the GitHub link. It was scheduled to automatically pull data at 11pm daily for 21 days. First 14 days have been for building model. Next 5 days have been used for model validation. Currently, the crawler pulls data for route Boston(BOS) to Los Angeles(LAX) for travel dates from today to 104 days (90 days + 14 days) in future. The crawler can be easily modified to pull data for any other route by changing origin and destination station. The first page of this travel site which has 35 itineraries that have been scrapped. The data consists of the airline information, type of carrier, route and price information. Sample row being scrapped is shown below.

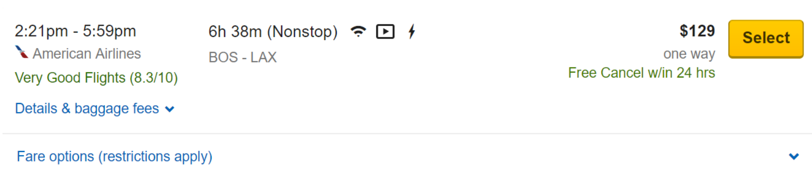


figure 2.1 : Data Elements scrapped

Exploratory Data analysis

After initial data cleaning, and null value treatment, we have plotted the average ticket prices for the travel period below.

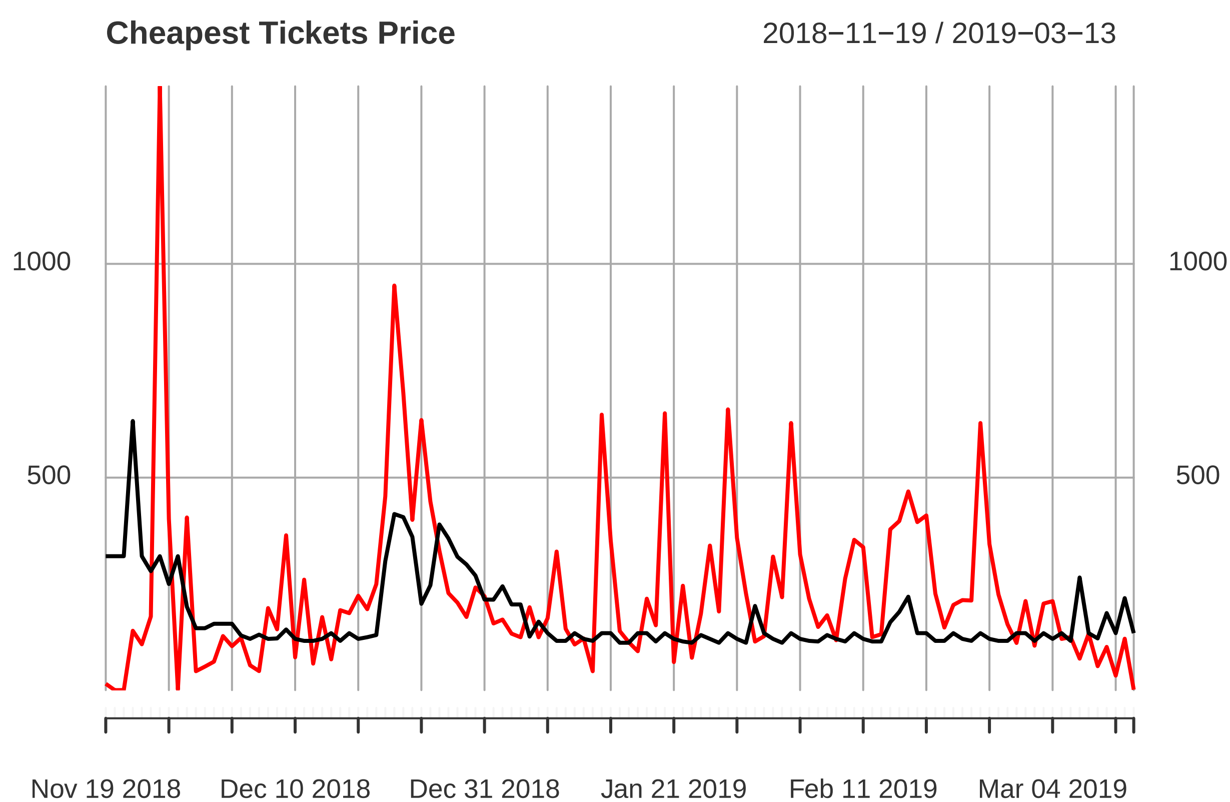


Figure 2.2 : Time Series of the cheapest price [Black] and standard deviation [Red] in price of ticket in 14 days.

Figure 2.2 shows there are few travel dates on which the fare is quite high compared to others due to high demand. The standard deviation in prices is also high for those days. The data consists of 14 observations of 35 cheapest itineraries for each travel date. Since we are not considering demand as our predictor variable, we need to normalize and aggregate this data in meaningful way in order to model it. We can do this by:
1. Dividing all the prices by representative (average/cheapest) price in 14 days
2. Taking percentage difference of price with respect to previous day and modelling for this percent change
The 2nd method is chosen as the 1st doesn't favor scalability. Also for the first method, the representative price for a travel date will depend on the days since departure.

The periodic spikes in standard deviation also suggests that the prices change a lot on particular days of a week. This indicates the day of booking ticket might be a good predictor estimating percent change.

Figure 2.3 : Box plot of cheapest ticket price in 14 days on different day of week

Figure 2.3 shows the distribution of cheapest ticket price as checked on various days. The travel period for all these days is same. The prices on Tuesday and Thursday are slightly less than the other days, indicating airlines might be giving better deals on these days. The 50 percentile price is low for Tuesday, Thursday and Friday. Price on Wednesday have high variance and tend to more on the higher side. This indicates that day of booking might have some effect on ticket prices

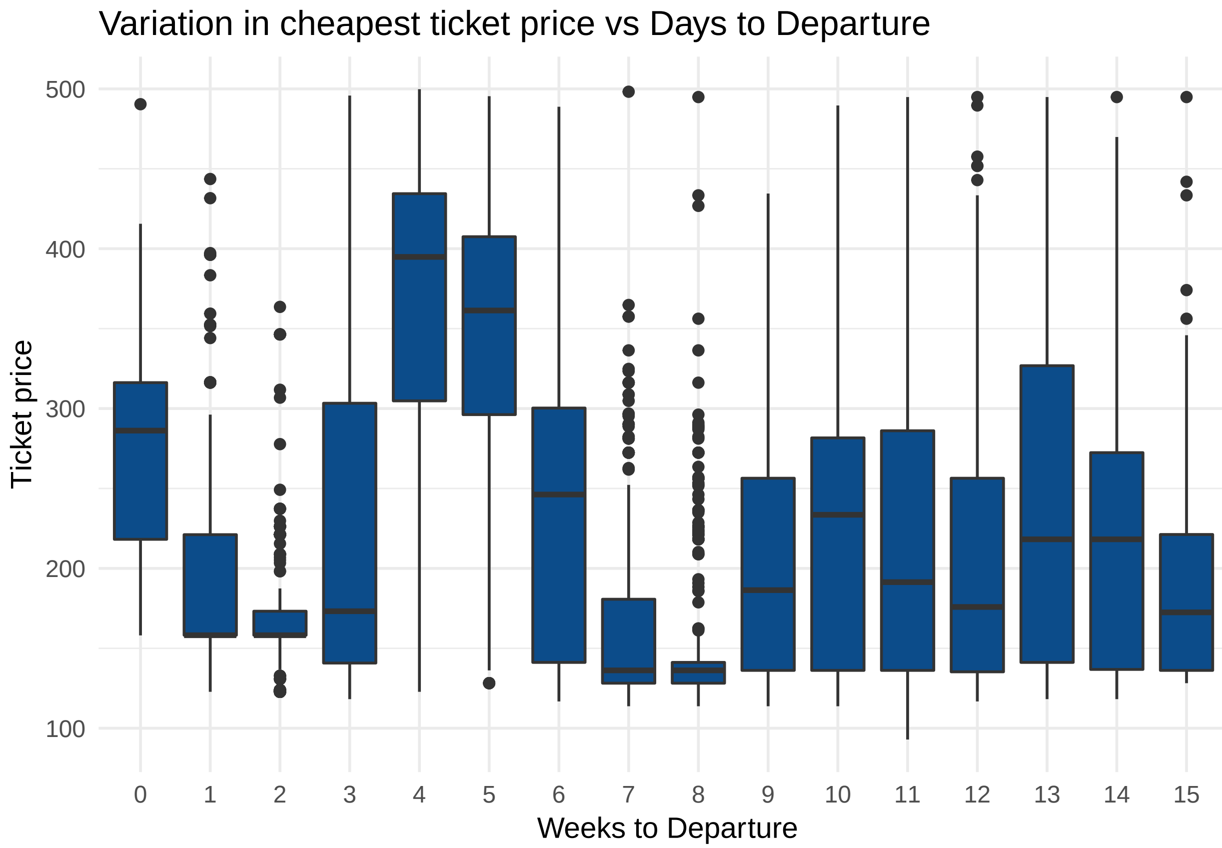


Figure 2.4 : Box plot of cheapest ticket price in 14 days vs weeks to departure

Figure 2.4 indicates that that the weeks from departure has a non-linear effect on the ticket prices. We also see that there is substantial difference in the average ticket prices, but the cheapest ticket prices are not varying a lot. Much of this is because of some very low demand flights at odd times. The buyers may not be interested in such flight. In order to account for this, we will model the change in average prices. We will be predicting the average ticket price and using the predicted average price, we will calculate the cheapest price of ticket. Hence a two-step prediction would be done.

There is significant difference in prices. From Week 8 onwards, there is very less change in price with no visible trend whatsoever. So the price doesn't vary much if the weeks to departure greater than 8. In between weeks 8 to 5, there is an increase in price. Sellers may be increasing the prices slowly in order to create perception of increasing prices for purchaser that is checking at regular intervals. The prices again decrease in weeks 3 to 5. This may be due to sellers reducing price in order to fill their seats. From week 3 to days of departure, there is consistent visible increase in price. The differences in ticket prices in the above graph can be due to two factors.
1. The demand of tickets price on certain days leading to high ticket
2. Price variation in tickets prices on different days of booking

In order to account for demand, we will be using random intercept of the date of travel.

Travel.Period	Day.Start	Day.End	Reason
Immediate	0	3	No prediction
Already too late	4	17	Prices are continuously increasing
Prime booking window	18	35	Prices are decreasing in this period
Fish for offer	36	56	Fake price increases to make customers book tickets
Constant pricing	56	90	Very less change in prices

In order to estimate price variations on different days, we will analyze the effect of days from departure.

Since this will not be a linear relationship in the entire travelling period, we are dividing the variable days to departure into 5 travel periods given above.

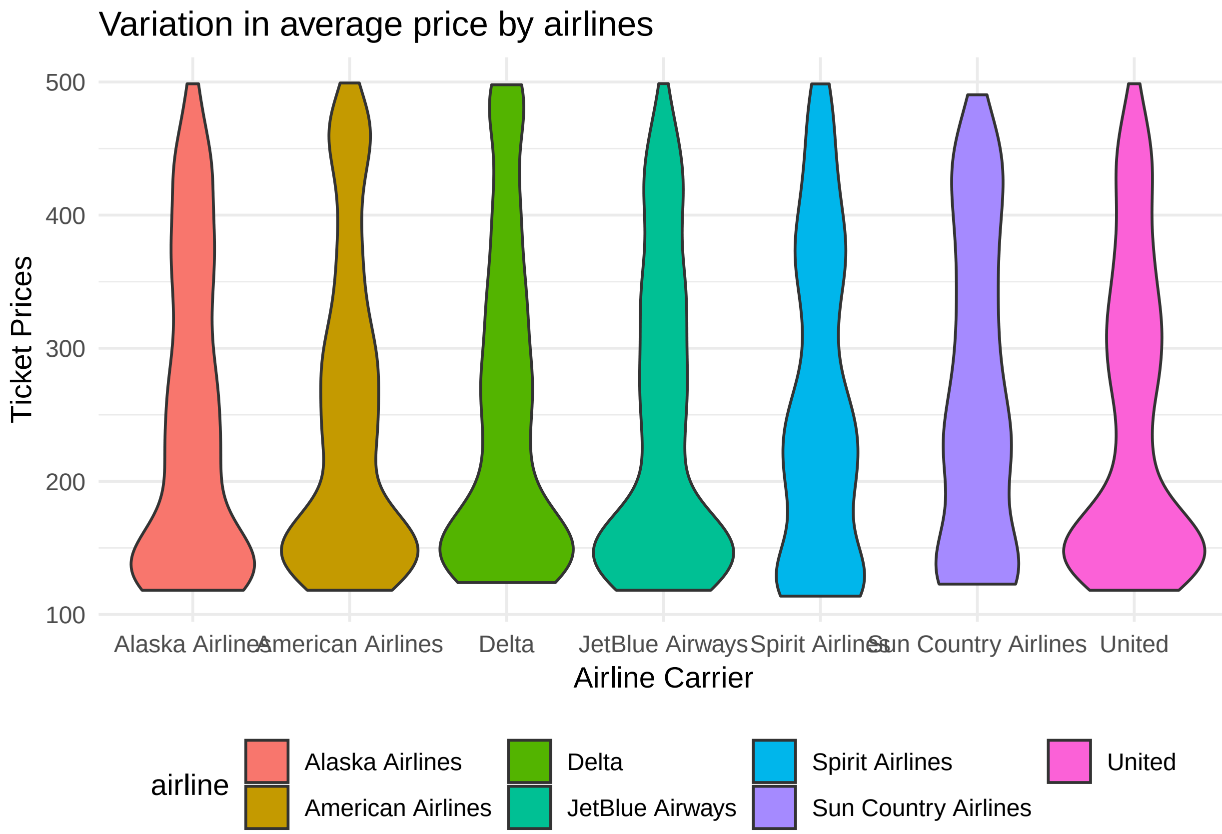


Figure 2.5 : Violin plot of average ticket price in 14 days for different airlines

The data consisted of many small airlines data as well. We have removed airlines that are present in less than 10% of the dates. From figure 2.5, we see that United, Jet Blue, Delta and American Airlines have similar distribution of prices. This indicates that they may be involved in dynamic pricing based on competitor pricing. Spirit Airlines has cheapest tickets but the number of options available to users seem very less. Sun Country Airlines also has a very uniform distribution in prices. This may also be due to the fact that Sun Country and Spirit Airlines may not be having as many flight as the other four carriers.

Model used

In order to model average prices, we are using a multilevel linear effects model. For easier interpretation of coefficients of model, we used rstan_lmer function from rstanarm to fit the model. Based on the 50% credible interval for the predicted average price, we can calculate the estimated lower average price and upper average price. Using a simple linear regression model, we can estimate cheapest price of ticket. The probability to buy the ticket can then be found by calculated based on whether the cheapest price of ticket is closer to upper or lower bound of cheapest ticket price. A simple linear regression model is run on top of this and predicts the cheapest price for every day using the average price for each airline.

$y = lmer((1|Date) + dd + week + airline + weekDay + (dd|travelPeriod))$
where dd = days to departure

$cheapestPrice = lm(AveragePrice, week)$

$Buy = I(chd < 0.9ch1)$

where chd = cheapest price on day 'd' , 'd+1' and 'd-1'
and d = day on which the cheapest price is predicted
and ch1 = cheapest price on day 1

Summary of the model can be found below:

```
## stan_lmer
##  family:       gaussian [identity]
##  formula:      y_trans ~ (1 | Date) + n_day + week + airline + wk_day + (scale(n_day) |
##     tp)
##  observations: 6100
## ------
##                            Median MAD_SD
## (Intercept)                 2.4    0.1
## n_day                       0.0    0.0
## week1                      -0.2    0.1
## week2                      -0.2    0.1
## week3                      -0.3    0.1
## week4                      -0.3    0.1
## week5                      -0.3    0.2
## week6                      -0.4    0.2
## week7                      -0.4    0.2
## week8                      -0.3    0.2
## week9                      -0.2    0.2
## week10                      0.0    0.2
## week11                     -0.1    0.2
## week12                     -0.1    0.2
## week13                      0.1    0.2
## week14                      0.1    0.2
## week15                      0.2    0.3
## airlineAmerican Airlines    0.1    0.0
## airlineDelta                0.1    0.0
## airlineJetBlue Airways      0.0    0.0
## airlineSpirit Airlines     -0.1    0.0
## airlineSun Country Airlines  0.0    0.0
## airlineUnited               0.1    0.0
## wk_dayMonday                0.1    0.0
## wk_daySaturday              0.1    0.0
## wk_daySunday                0.2    0.0
## wk_dayThursday              0.0    0.0
## wk_dayTuesday               0.1    0.0
## wk_dayWednesday             0.1    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.4    0.0
##
## Error terms:
##  Groups   Name         Std.Dev. Corr
##  Date     (Intercept)  0.473
##  tp       (Intercept)  0.089
##           scale(n_day) 0.134    0.19
##  Residual              0.365
## Num. levels: Date 115, tp 5
##
```

```
## Sample avg. posterior predictive distribution of y:
##          Median MAD_SD
## mean_PPD 2.3    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Result

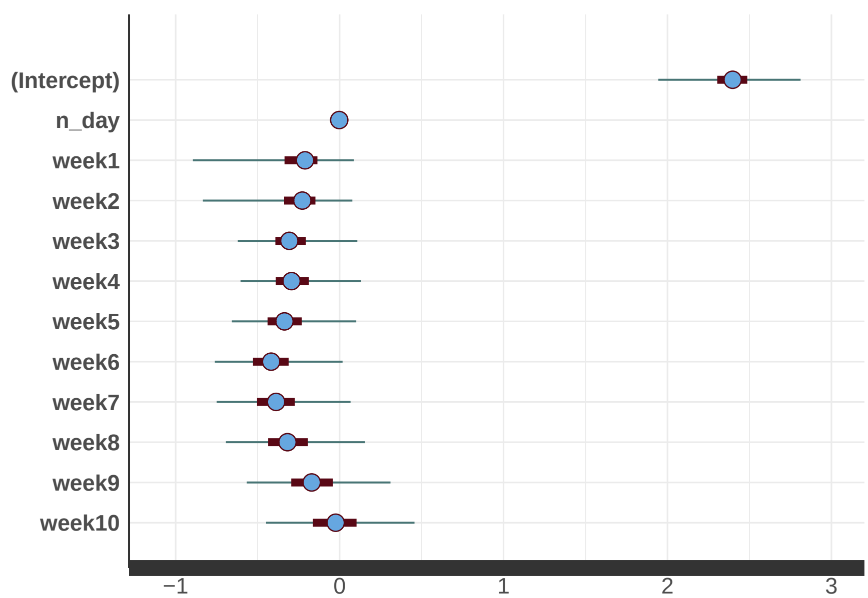
Model choice

Average prices are repeated measures of the ticket prices on various travel dates for 14 continuous days. The price will be driven by each day's demand in general. So we need to take into account the random effect of each travel date. After removing the random effect of each date due to demand, we are trying to understand the effect of days of departure, day of week and airlines on the average ticket prices. We see that the airlines follow each other closely, hence we have not included 'airline' as fixed effect. Summary of the model coefficients can be seen in the appendix. Using the average price predicted from the model, we will calculate the cheapest ticket price on the following days. Based on prediction of cheapest ticket price, we will find the day with the lowest ticket price and wait up until that day. The cheapest ticket price will be checked daily in this period daily. The strategy to buy ticket is that if the price falls below 10% of the cheapest ticket price on first day in this period. Since we are only performing the check in 7 days, there will be many cases in which the purchaser should not be buying ticket in the seven days. But since we don't know the future price, we will force the purchaser to buy the ticket within the seven days only.

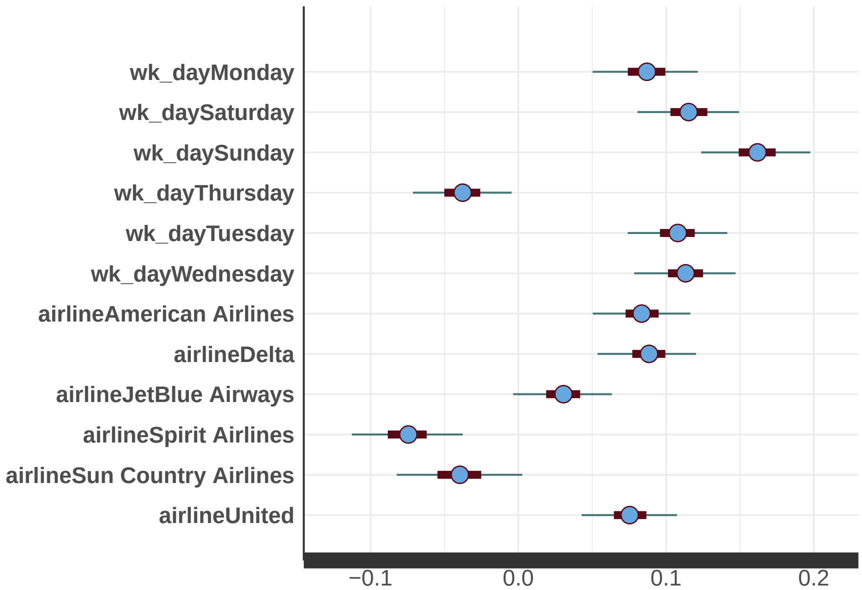
Interpretation

All the coefficients have been calculated according to the y_transformed. If the coefficient is > 0, it has positive effect on the price, hence it will increase the price. If the coefficient is <0, it has positive effect on the price.

Fixed Effects:

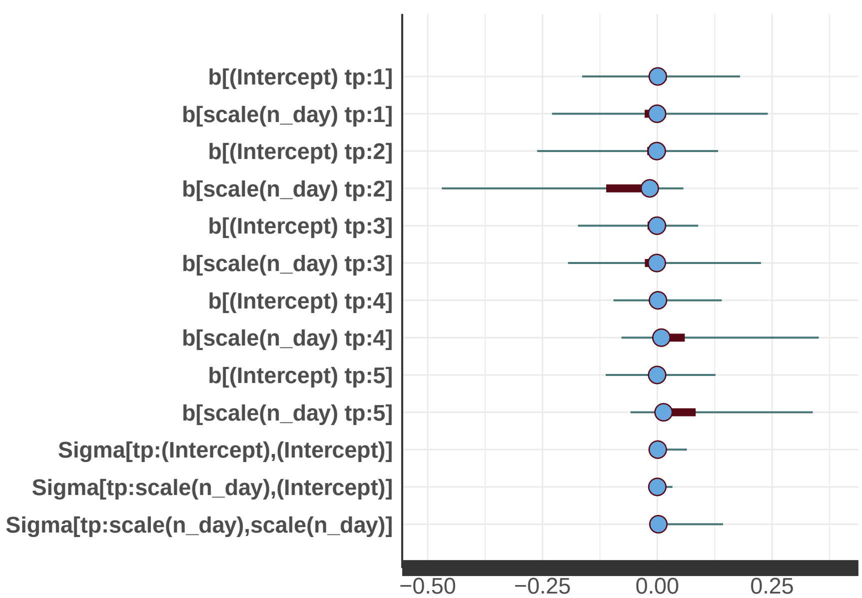


Variable ‘n_day’ represents the days until departure. Week is taken as a categorical variable. This allows for the slope to positive or negative in different weeks. The coefficients of weeks are all significant. The effect of days since depature is dependent of both n_day and week. n_day is negative (median value : 0.00165). As the days to departure increase, the negative coefficient of n_day suggests decrease in price. On the other hand, the week parameter is positive. This allows for both positive and negative slopes along the travel duration.



The coefficient for Thursday is negative indicating that there is decrease of price tickets on Thursdays. The coefficients on Sunday is highest indicating that the sellers are hiking up the price on Sundays when majority of people have free time and hence book their tickets.
Among airlines, Spirit airlines and Sun Country airlines have cheapest tickets. American Airlines, Delta and United airlines follow the same pattern as we saw in our data before. Among the most frequent flights, Jet Blue has tickets slightly on the cheaper side.

Random Effects:



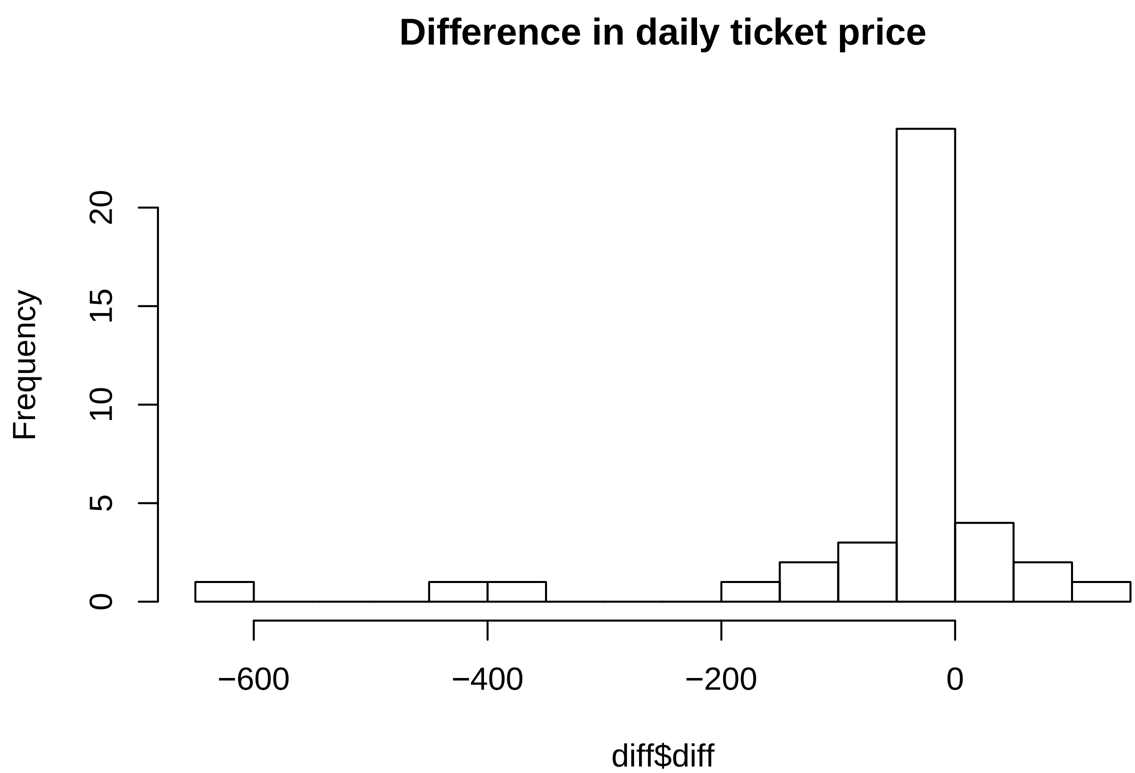
Random effect of travel period

is applied on top of n_day to allow the slope of n_day to be different in different travel periods.We shouldn't be looking at the significance level for the group level coefficients because they help in explaining some of the variance by pooling but they themselves may not neccessarily be significant. This model will be the least as predictive as a model without random effects.

As expected the within group variance in group: Date is very less compared to the overall standard deviation. Hence it is a good variable for grouping. Also, the within group variance of travel period is less than total standard deviation which tells the effect of days from departure is different from period to period.

Model Testing framework

In order to compare the model, we will use a naive strategy of buying the ticket as early as possible. Under this strategy, the purchaser buys the cheapest ticket at the time he checks. The purchaser checks the ticket prices on 2 December(Sunday) which is a neutral day with neither very high or low price. This is compared against the current strategy and the price difference is calculated. Since the posterior prediction are probabilistic, in order to get estimate of the money saved by using the strategy, we will run 1000 simulations for both the strategy. For the 1000 simulations, the difference in the ticket prices is calculated. The distribution of the total price difference can be seen in the histogram below. The user on an average saves US $29.44 with this strategy.



Model checking

We checked for the assumptions of linear mixed effects model. For checking the accuracy of 1st model, an out of time frame validation was done. The histogram of actual and predicted values can be seen in appendix. Since the average prices were right skewed, we transformed the average price under the following transformation:

$y < -log(\sqrt[2]{averagePrice - min(averagePrice)})$

This reduces the right skew present in the data, but doesn't remove the inherent bimodal distribution in the data. The errors are normally distributed and the distribution of errors with respect to various important variables can be seen in appendix. We performed posterior predictive checking. If the data is a good fit for our model, the data generated from the model should resemble our actual model. The results can be seen under Section: Limitations. We also compared the average price in the predicted period. Also the error in the average prices (plot in appendix) resembled a normal distribution.

Discussion

Limitation

The model applies the condition of linearity on the average price with respect to days of departure. In order to compensate for this effect, we divided the entire travel period into different travel period. But within the categories, we are imposing average price to follow a linear trend. A better approximation would be allowing quadratic or higher order function to be used for estimating the trend. Hence regression using non-linear function might give us better result.

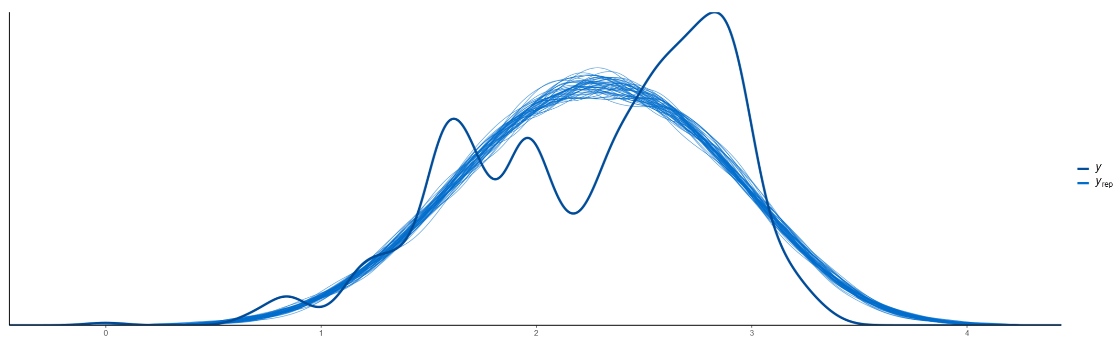


figure 2.1

The distribution of average price is multi-modal and multilevel linear regression model can typically handle unimodal distributions. In order to fit this model better, we might need to divide the data into two categories ie high ticket price days and low ticket price days. We can determine whether the day lies in high or low day Using Expectation Maximization algorithm. The two corresponding unimodal distributions can then be better modelled using linear mixed effects.

The strategy for buying tickets can be modelled as well. The current strategy involves taking the ticket in coming few days, if the cheapest ticket is less than 10% of first day value. We can model the optimum percentage change based on the difference between the current price and the predicted price.

Future direction

We will look into making the analysis more robust and analyze more flight paths using the methodology setup. The coefficients from different flight paths can then be compared. Factors having similar effects can then be grouped to set up a generic model for modelling price for any domestic US airport. This can be used in case when past data of the route of interest is not available for analysis. The factors with effects varying by route can be used for route specific analysis. Also in order to overcome the limitations mentioned above, we will use non-linear functions while evaluating the parameters. The final aim for this project is to come up with an app to provide users data to guide and judge the correct time to buy tickets. After completing the analysis and setting up generic as well as route based model, we will aim to create a webapp for the visualizing the results.

Reference

Gelman, A., and Hill, J. (2007). Data analysis using regression and multilevel/hierarchical models. New York: Cambridge University Press.
https://mjskay.github.io/tidybayes/articles/tidy-rstanarm.html

Appendix

```
## [1] "Summary of model"
```

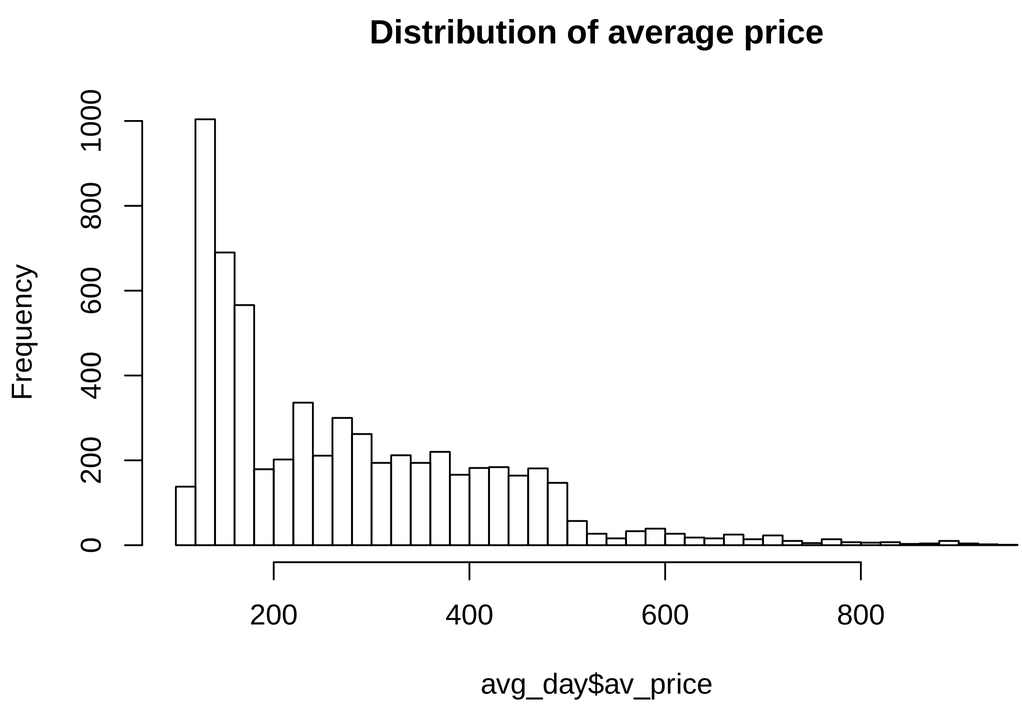
```
##                                 Rhat   mean  X2.5.  X97.5.
## (Intercept)                     1.01   2.39   1.94    2.81
## n_day                           1.01   0.00  -0.01    0.00
## week1                           1.02  -0.27  -0.89    0.09
## week2                           1.01  -0.27  -0.83    0.08
## week3                           1.01  -0.29  -0.62    0.11
## week4                           1.01  -0.28  -0.60    0.13
## week5                           1.01  -0.33  -0.66    0.10
## week6                           1.01  -0.41  -0.76    0.02
## week7                           1.01  -0.38  -0.75    0.07
## week8                           1.01  -0.31  -0.69    0.15
## week9                           1.01  -0.16  -0.57    0.31
## week10                          1.01  -0.03  -0.45    0.46
## week11                          1.01  -0.11  -0.55    0.39
## week12                          1.01  -0.06  -0.53    0.44
## week13                          1.01   0.07  -0.43    0.60
## week14                          1.01   0.15  -0.37    0.70
## week15                          1.01   0.18  -0.39    0.74
## airlineAmerican Airlines        1.00   0.08   0.05    0.12
## airlineDelta                    1.00   0.09   0.05    0.12
## airlineJetBlue Airways          1.00   0.03   0.00    0.06
## airlineSpirit Airlines          1.00  -0.07  -0.11   -0.04
## airlineSun Country Airlines     1.00  -0.04  -0.08    0.00
## airlineUnited                   1.00   0.08   0.04    0.11
## wk_dayMonday                    1.00   0.09   0.05    0.12
## wk_daySaturday                  1.00   0.12   0.08    0.15
## wk_daySunday                    1.00   0.16   0.12    0.20
## wk_dayThursday                  1.00  -0.04  -0.07    0.00
## wk_dayTuesday                   1.00   0.11   0.07    0.14
## wk_dayWednesday                 1.00   0.11   0.08    0.15
## b[(Intercept) Date:01/01/2019]  1.02   0.70   0.57    0.83
## b[(Intercept) Date:01/02/2019]  1.02   0.72   0.58    0.86
## b[(Intercept) Date:01/03/2019]  1.02   0.61   0.48    0.75
## b[(Intercept) Date:01/04/2019]  1.01   0.47   0.33    0.60
## b[(Intercept) Date:01/05/2019]  1.01   0.46   0.32    0.60
## b[(Intercept) Date:01/06/2019]  1.02   0.49   0.36    0.62
## b[(Intercept) Date:01/07/2019]  1.02   0.14   0.02    0.27
## b[(Intercept) Date:01/08/2019]  1.01  -0.59  -0.73   -0.45
## b[(Intercept) Date:01/09/2019]  1.01  -0.11  -0.25    0.03
## b[(Intercept) Date:01/10/2019]  1.01  -0.26  -0.41   -0.11
## b[(Intercept) Date:01/11/2019]  1.01  -0.32  -0.45   -0.18
## b[(Intercept) Date:01/12/2019]  1.01  -0.57  -0.70   -0.44
## b[(Intercept) Date:01/13/2019]  1.01   0.54   0.41    0.67
```

```
## b[(Intercept) Date:01/14/2019]      1.01    0.15    0.01    0.29
## b[(Intercept) Date:01/15/2019]      1.02   -0.58   -0.71   -0.44
## b[(Intercept) Date:01/16/2019]      1.01   -0.53   -0.66   -0.40
## b[(Intercept) Date:01/17/2019]      1.01   -0.32   -0.46   -0.18
## b[(Intercept) Date:01/18/2019]      1.01   -0.23   -0.37   -0.09
## b[(Intercept) Date:01/19/2019]      1.02   -0.75   -0.88   -0.62
## b[(Intercept) Date:01/20/2019]      1.01    0.49    0.37    0.62
## b[(Intercept) Date:01/21/2019]      1.01   -0.27   -0.42   -0.13
## b[(Intercept) Date:01/22/2019]      1.01   -0.03   -0.17    0.11
## b[(Intercept) Date:01/23/2019]      1.01   -0.93   -1.07   -0.78
## b[(Intercept) Date:01/24/2019]      1.02   -0.04   -0.17    0.10
## b[(Intercept) Date:01/25/2019]      1.01    0.09   -0.05    0.23
## b[(Intercept) Date:01/26/2019]      1.01   -0.17   -0.30   -0.04
## b[(Intercept) Date:01/27/2019]      1.01    0.09   -0.05    0.22
## b[(Intercept) Date:01/28/2019]      1.01    0.03   -0.11    0.16
## b[(Intercept) Date:01/29/2019]      1.01   -0.70   -0.84   -0.56
## b[(Intercept) Date:01/30/2019]      1.01    0.37    0.23    0.50
## b[(Intercept) Date:01/31/2019]      1.01   -0.26   -0.43   -0.10
## b[(Intercept) Date:02/01/2019]      1.01   -0.06   -0.20    0.08
## b[(Intercept) Date:02/02/2019]      1.01   -0.48   -0.62   -0.34
## b[(Intercept) Date:02/03/2019]      1.02    0.48    0.35    0.61
## b[(Intercept) Date:02/04/2019]      1.02    0.04   -0.10    0.17
## b[(Intercept) Date:02/05/2019]      1.01    0.16    0.02    0.29
## b[(Intercept) Date:02/06/2019]      1.01   -0.06   -0.20    0.08
## b[(Intercept) Date:02/07/2019]      1.02   -0.14   -0.27    0.00
## b[(Intercept) Date:02/08/2019]      1.02   -0.53   -0.67   -0.38
## b[(Intercept) Date:02/09/2019]      1.02   -0.40   -0.54   -0.26
## b[(Intercept) Date:02/10/2019]      1.01    0.04   -0.09    0.18
## b[(Intercept) Date:02/11/2019]      1.01    0.08   -0.05    0.23
## b[(Intercept) Date:02/12/2019]      1.02   -0.55   -0.70   -0.41
## b[(Intercept) Date:02/13/2019]      1.01   -0.42   -0.56   -0.28
## b[(Intercept) Date:02/14/2019]      1.01    0.34    0.21    0.49
## b[(Intercept) Date:02/15/2019]      1.01    0.45    0.31    0.60
## b[(Intercept) Date:02/16/2019]      1.01    0.63    0.49    0.79
## b[(Intercept) Date:02/17/2019]      1.01    0.55    0.41    0.70
## b[(Intercept) Date:02/18/2019]      1.01    0.00   -0.13    0.15
## b[(Intercept) Date:02/19/2019]      1.01   -0.19   -0.33   -0.05
## b[(Intercept) Date:02/20/2019]      1.01   -0.29   -0.43   -0.15
## b[(Intercept) Date:02/21/2019]      1.01   -0.51   -0.67   -0.34
## b[(Intercept) Date:02/22/2019]      1.01   -0.24   -0.39   -0.08
## b[(Intercept) Date:02/23/2019]      1.01   -0.16   -0.31   -0.02
## b[(Intercept) Date:02/24/2019]      1.02    0.35    0.21    0.49
## b[(Intercept) Date:02/25/2019]      1.01    0.04   -0.11    0.19
## b[(Intercept) Date:02/26/2019]      1.01    0.03   -0.13    0.18
## b[(Intercept) Date:02/27/2019]      1.01   -0.60   -0.76   -0.44
## b[(Intercept) Date:02/28/2019]      1.01   -0.45   -0.61   -0.30
## b[(Intercept) Date:03/01/2019]      1.01    0.27    0.12    0.42
## b[(Intercept) Date:03/02/2019]      1.01   -0.58   -0.76   -0.41
## b[(Intercept) Date:03/03/2019]      1.01    0.08   -0.08    0.24
## b[(Intercept) Date:03/04/2019]      1.01   -0.09   -0.26    0.07
## b[(Intercept) Date:03/05/2019]      1.01   -0.10   -0.27    0.06
## b[(Intercept) Date:03/06/2019]      1.00   -1.07   -1.26   -0.88
## b[(Intercept) Date:03/07/2019]      1.01    0.17   -0.02    0.37
## b[(Intercept) Date:03/08/2019]      1.00   -0.17   -0.36    0.02
```

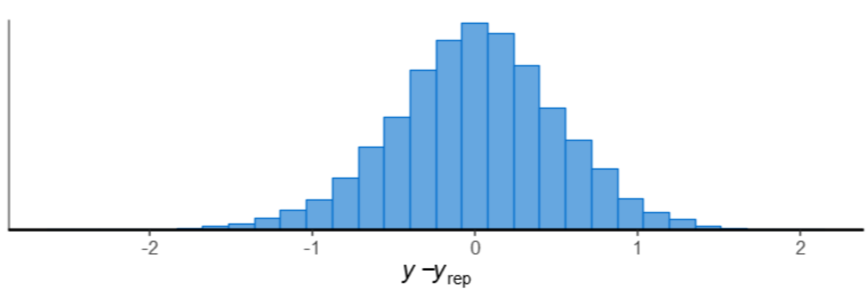
```
## b[(Intercept) Date:03/09/2019]       1.00    -0.42    -0.62    -0.22
## b[(Intercept) Date:03/10/2019]       1.00     0.06    -0.17     0.29
## b[(Intercept) Date:03/11/2019]       1.00    -0.68    -1.00    -0.37
## b[(Intercept) Date:03/12/2019]       1.00    -0.19    -0.52     0.13
## b[(Intercept) Date:03/13/2019]       1.00    -0.85    -1.20    -0.50
## b[(Intercept) Date:11/19/2018]       1.00     0.08    -0.26     0.41
## b[(Intercept) Date:11/20/2018]       1.00     0.09    -0.32     0.48
## b[(Intercept) Date:11/21/2018]       1.00     0.09    -0.29     0.49
## b[(Intercept) Date:11/22/2018]       1.00     0.52     0.16     0.88
## b[(Intercept) Date:11/23/2018]       1.00     0.22    -0.11     0.56
## b[(Intercept) Date:11/24/2018]       1.01     0.38     0.16     0.60
## b[(Intercept) Date:11/25/2018]       1.01     0.63     0.40     0.85
## b[(Intercept) Date:11/26/2018]       1.01     0.28     0.11     0.46
## b[(Intercept) Date:11/27/2018]       1.00     0.09    -0.25     0.42
## b[(Intercept) Date:11/28/2018]       1.01     0.00    -0.18     0.17
## b[(Intercept) Date:11/29/2018]       1.01    -0.12    -0.30     0.05
## b[(Intercept) Date:11/30/2018]       1.01    -0.10    -0.30     0.09
## b[(Intercept) Date:12/01/2018]       1.00    -0.11    -0.31     0.09
## b[(Intercept) Date:12/02/2018]       1.01    -0.28    -0.46    -0.10
## b[(Intercept) Date:12/03/2018]       1.01    -0.26    -0.42    -0.09
## b[(Intercept) Date:12/04/2018]       1.01    -0.30    -0.46    -0.14
## b[(Intercept) Date:12/05/2018]       1.01    -0.50    -0.65    -0.35
## b[(Intercept) Date:12/06/2018]       1.01    -0.37    -0.53    -0.22
## b[(Intercept) Date:12/07/2018]       1.01    -0.23    -0.38    -0.08
## b[(Intercept) Date:12/08/2018]       1.01    -0.41    -0.57    -0.25
## b[(Intercept) Date:12/09/2018]       1.01    -0.23    -0.38    -0.08
## b[(Intercept) Date:12/10/2018]       1.01    -0.46    -0.61    -0.32
## b[(Intercept) Date:12/11/2018]       1.01    -0.60    -0.75    -0.46
## b[(Intercept) Date:12/12/2018]       1.02    -0.73    -0.87    -0.59
## b[(Intercept) Date:12/13/2018]       1.02    -0.24    -0.38    -0.09
## b[(Intercept) Date:12/14/2018]       1.02    -0.32    -0.47    -0.18
## b[(Intercept) Date:12/15/2018]       1.02     0.18     0.03     0.32
## b[(Intercept) Date:12/16/2018]       1.02    -0.14    -0.28     0.00
## b[(Intercept) Date:12/17/2018]       1.02    -0.21    -0.35    -0.08
## b[(Intercept) Date:12/18/2018]       1.02     0.18     0.04     0.31
## b[(Intercept) Date:12/19/2018]       1.01     0.48     0.33     0.62
## b[(Intercept) Date:12/20/2018]       1.02     0.77     0.64     0.91
## b[(Intercept) Date:12/21/2018]       1.01     0.90     0.76     1.04
## b[(Intercept) Date:12/22/2018]       1.02     0.99     0.85     1.13
## b[(Intercept) Date:12/23/2018]       1.01     0.80     0.66     0.93
## b[(Intercept) Date:12/24/2018]       1.02     0.71     0.57     0.85
## b[(Intercept) Date:12/25/2018]       1.02     0.79     0.65     0.93
## b[(Intercept) Date:12/26/2018]       1.01     0.86     0.72     1.01
## b[(Intercept) Date:12/27/2018]       1.01     0.88     0.74     1.02
## b[(Intercept) Date:12/28/2018]       1.01     0.83     0.69     0.97
## b[(Intercept) Date:12/29/2018]       1.01     0.81     0.68     0.95
## b[(Intercept) Date:12/30/2018]       1.01     0.78     0.64     0.92
## b[(Intercept) Date:12/31/2018]       1.01     0.68     0.54     0.81
## b[(Intercept) tp:1]                  1.00     0.00    -0.16     0.18
## b[scale(n_day) tp:1]                 1.01    -0.01    -0.23     0.24
## b[(Intercept) tp:2]                  1.01    -0.01    -0.26     0.13
## b[scale(n_day) tp:2]                 1.01    -0.08    -0.47     0.06
## b[(Intercept) tp:3]                  1.00    -0.01    -0.17     0.09
## b[scale(n_day) tp:3]                 1.02     0.00    -0.19     0.23
```

```
## b[(Intercept) tp:4]                    1.01     0.01    -0.10     0.14
## b[scale(n_day) tp:4]                   1.02     0.05    -0.08     0.35
## b[(Intercept) tp:5]                    1.01     0.00    -0.11     0.13
## b[scale(n_day) tp:5]                   1.02     0.05    -0.06     0.34
## sigma                                  1.00     0.36     0.36     0.37
## Sigma[Date:(Intercept),(Intercept)] 1.02     0.22     0.17     0.29
## Sigma[tp:(Intercept),(Intercept)]   1.01     0.01     0.00     0.06
## Sigma[tp:scale(n_day),(Intercept)]  1.01     0.00    -0.01     0.03
## Sigma[tp:scale(n_day),scale(n_day)] 1.03     0.02     0.00     0.14
## mean_PPD                               1.00     2.27     2.26     2.28
## log-posterior                          1.00 -2724.94 -2750.26 -2701.35
```

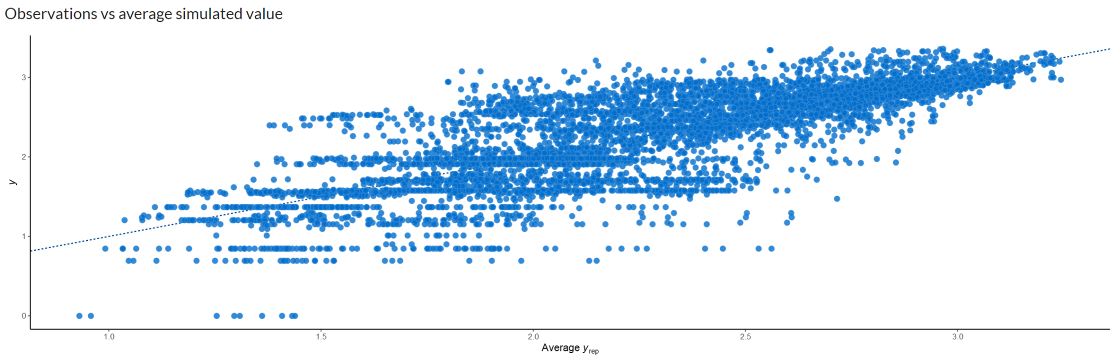
```
## [1] "Distribution of average price"
```




```
## [1] "Error Distribution"
```

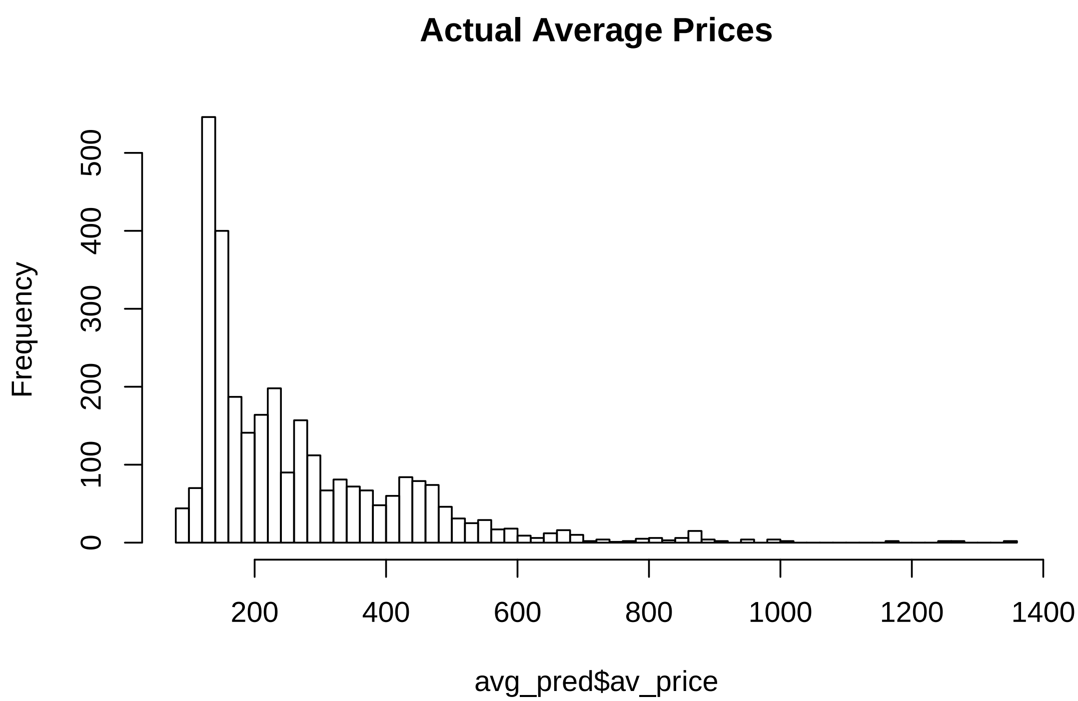



```
## [1] "Actual vs Predicted values"
```

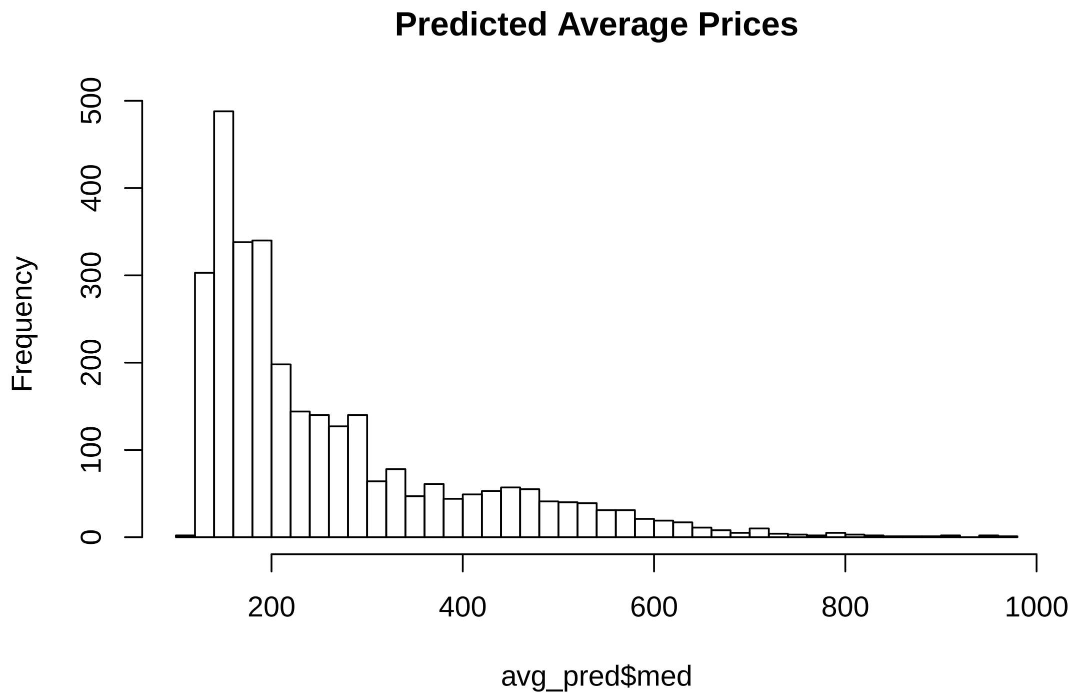



```
## [1] "Histograms of average ticket prices from out of frame validation"
```

```
## [1] "Histograms of actual average ticket prices"
```




```
## [1] "Histograms of predicted average ticket prices"
```


Predicted Average Prices
Frequency
0
100
200
300
400
500
200
400
600
800
1000
avg_pred$med