Marketing Data Analytics on Olist data

"Vikrant Nakod"

9/12/2020

BUAN 6357.003 PROJECT

Executive Summary

The online retail market is growing at a rapid pace where customers and vendors are actively looking for more engaging and highly personalized retail experiences. To achieve success and stay afloat in a highly competitive and volatile market, e-commerce businesses must be able to stay one step ahead of their customers. I have tried my hands at analyzing the data provided by the Olist. It is probably not relevant to Olist as it is not ecommerce company itself, but due to it's multidimesnional data which covers various important aspects of an customer order, I just want to provide with important metrics and predictive analytics that will help ecommerce company to find various new ways to survive. Based on the transactional data from 2016 to 2018, I performed extensive Exploratory Data Analysis and used ARIMA and ETS techniques to forecast the monthly revenue. I have utilized RFM analysis and the k-means algorithm for customer segmentation.

Introduction:

Olist is a Brazilian departmental store (marketplace) that operates in e-commerce segment but is not an e-commerce itself (as she says). It operates as a SaaS (Software as a Service) technology company since 2015. It offers a marketplace solution (of e-commerce segment) to shopkeepers of all sizes (and for most segments) to increase their sales whether they have online presence or not.

Data Description:

The complete dataset contains 10 csv files with almost 100k records from 2016 to 2018. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes.

What are the dimensions of the final dataset?

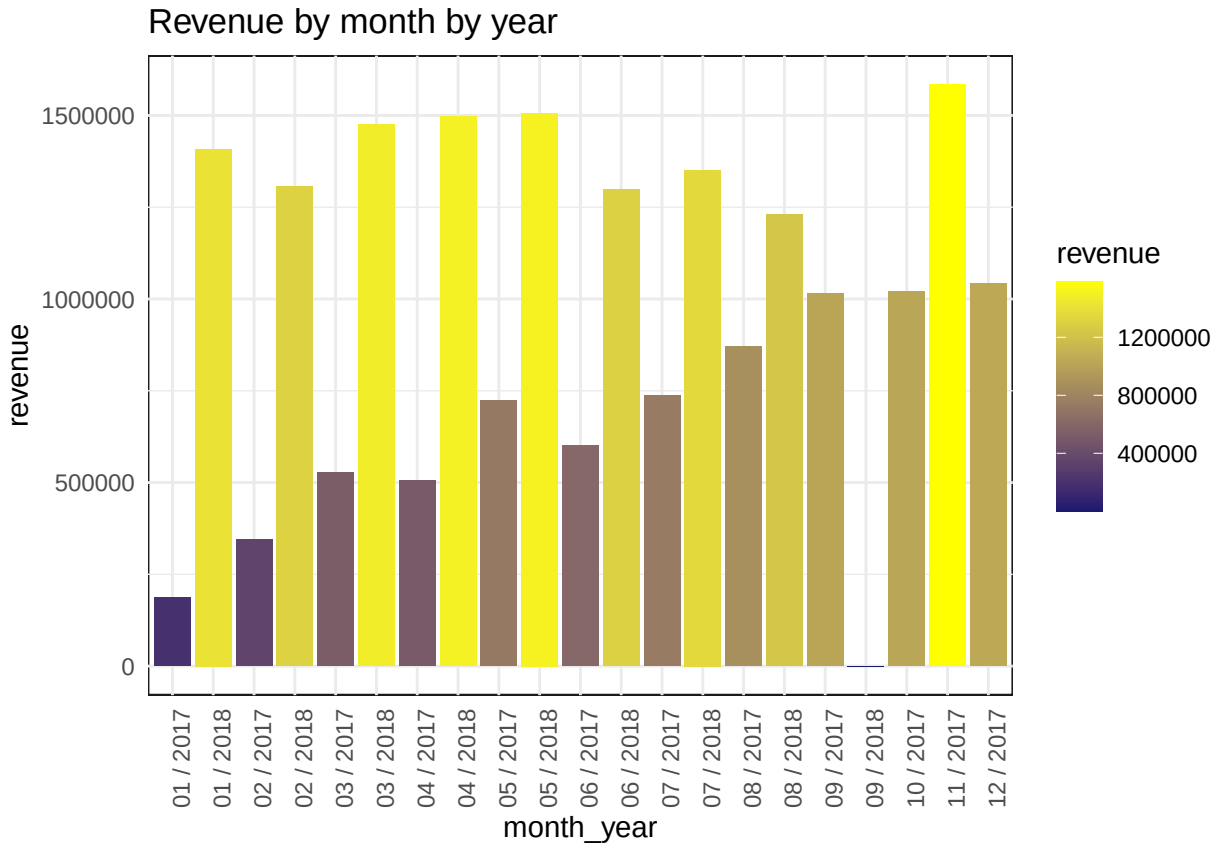
```
dim(ordered_df)
```

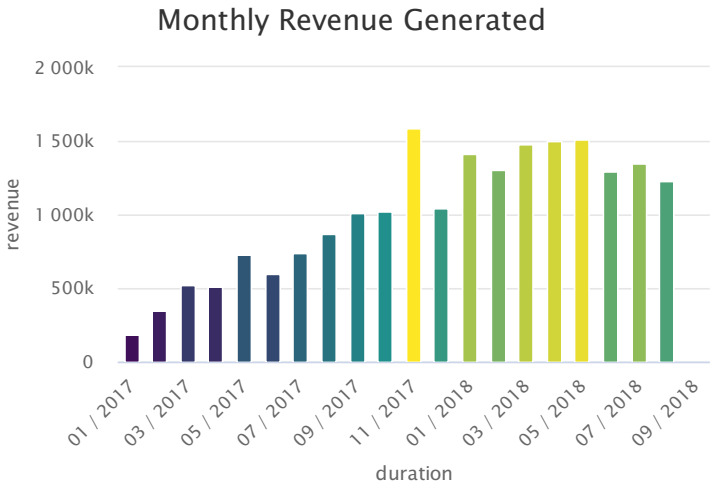
```
## [1] 117601     23
```

D. Exploratory Data Analysis

Starting with the monthly revenue generated(in braziilian real)

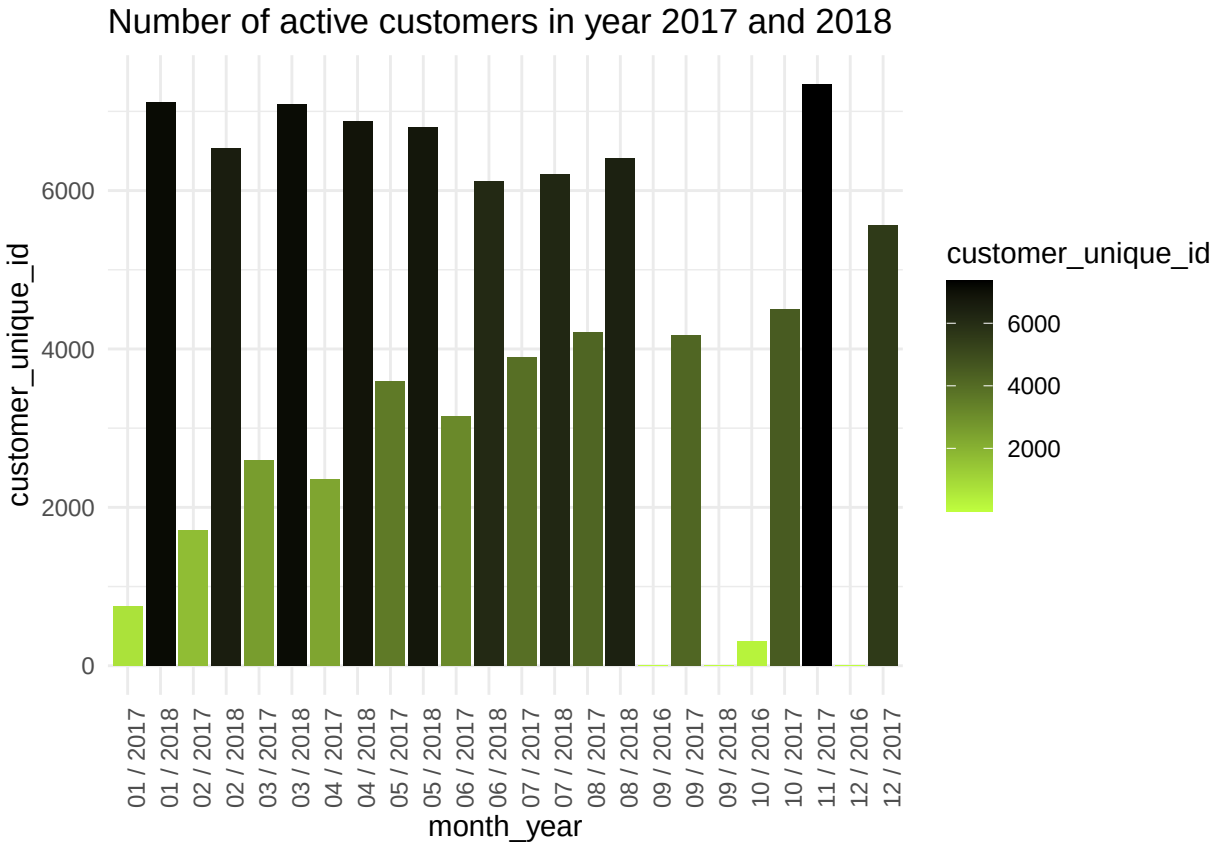
Here, we can see the revenue increases with the time





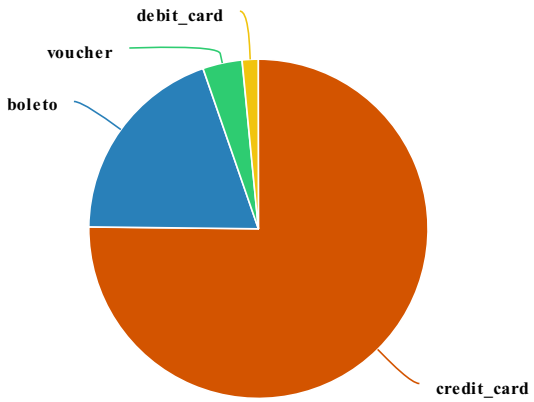
In above bar chart, we can see the the monthly comparisons of 2017 and 2018 in terms of the revenue. The data is only of few transactions for September 2018 and hence no bar for that duration

Next, we will look at the active monthly customers Olist had in year 2017 and 2018.



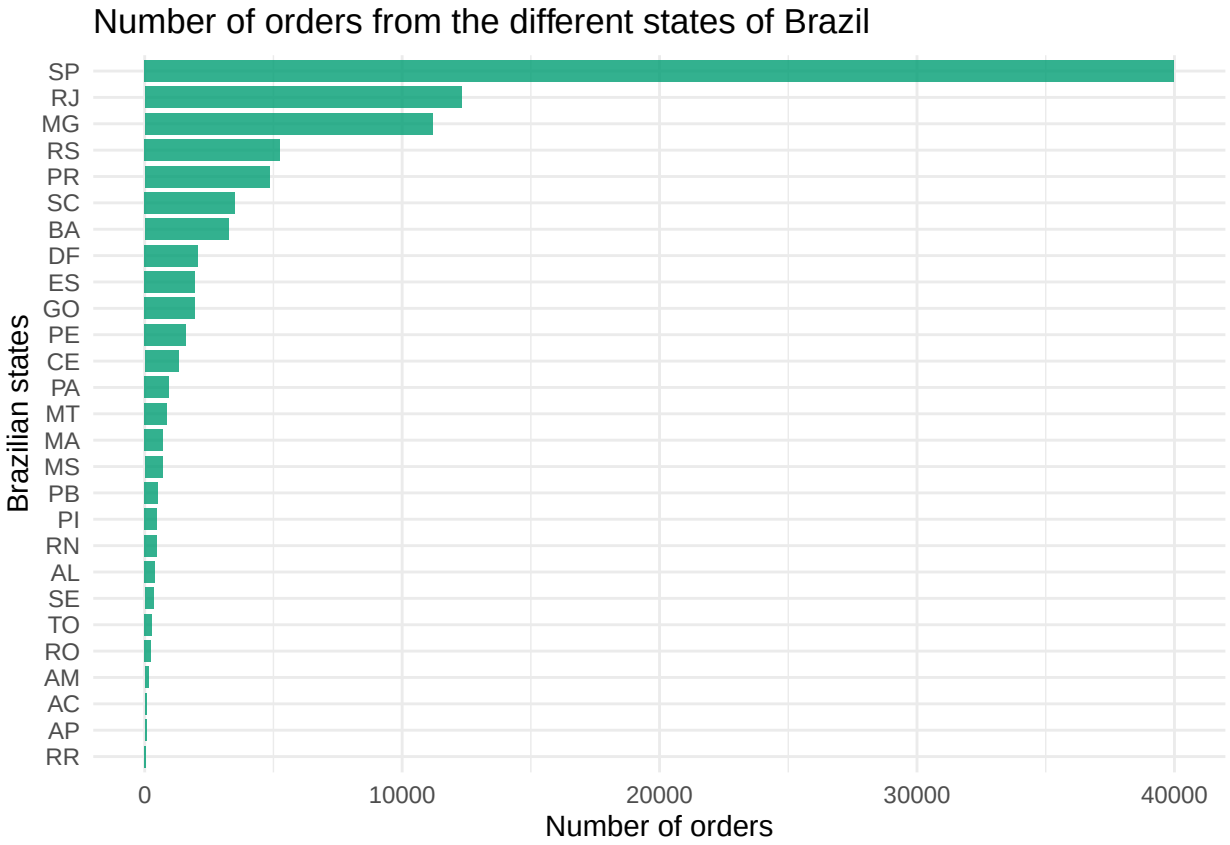
###Moving ahead with the type of payments chosen by the customers

Distribution of Payments



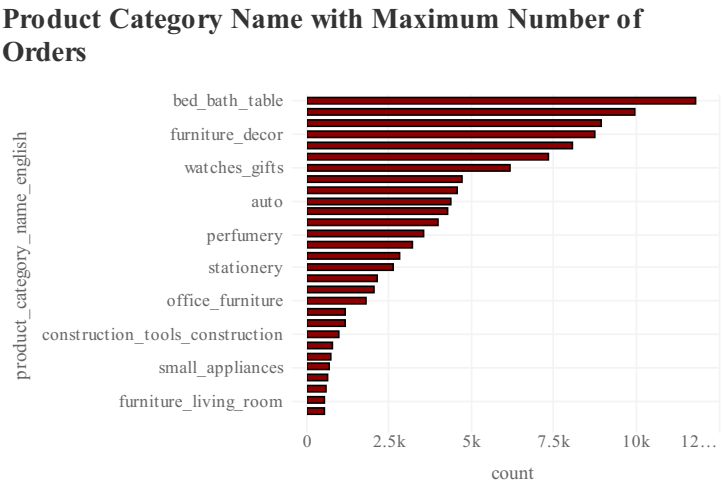
From above interactive pie chart we can see that credit card was most preferred payment type followed by the boleto which is brazilian payment type regulated by brazilian federation of Banks and is used for ecommerce and utility payments.

Now, customers from which state does the more shopping?



Sao Paulo has the most number of shoppers for Olist with almost 40K customers. It is followed by the Rio De Janeiro and Minas Gerais.

Which products are most popular among the customers?

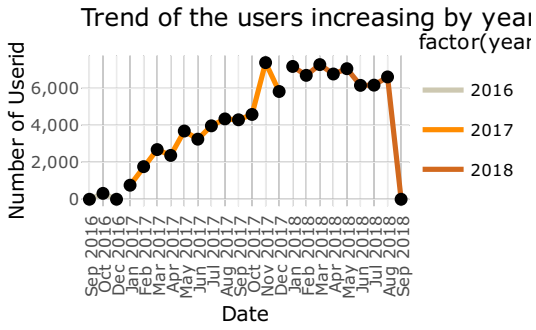


Bed_Bath_Table is most sought after product category with almost 12k orders. Health beauty products are sold is most numbers of the bed_bath_tabel category.

How the customers are increasing each month?

```
## Warning: package 'plotly' was built under R version 4.0.3
```

```
## Warning: `group_by_()` is deprecated as of dplyr 0.7.0.
## Please use `group_by()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

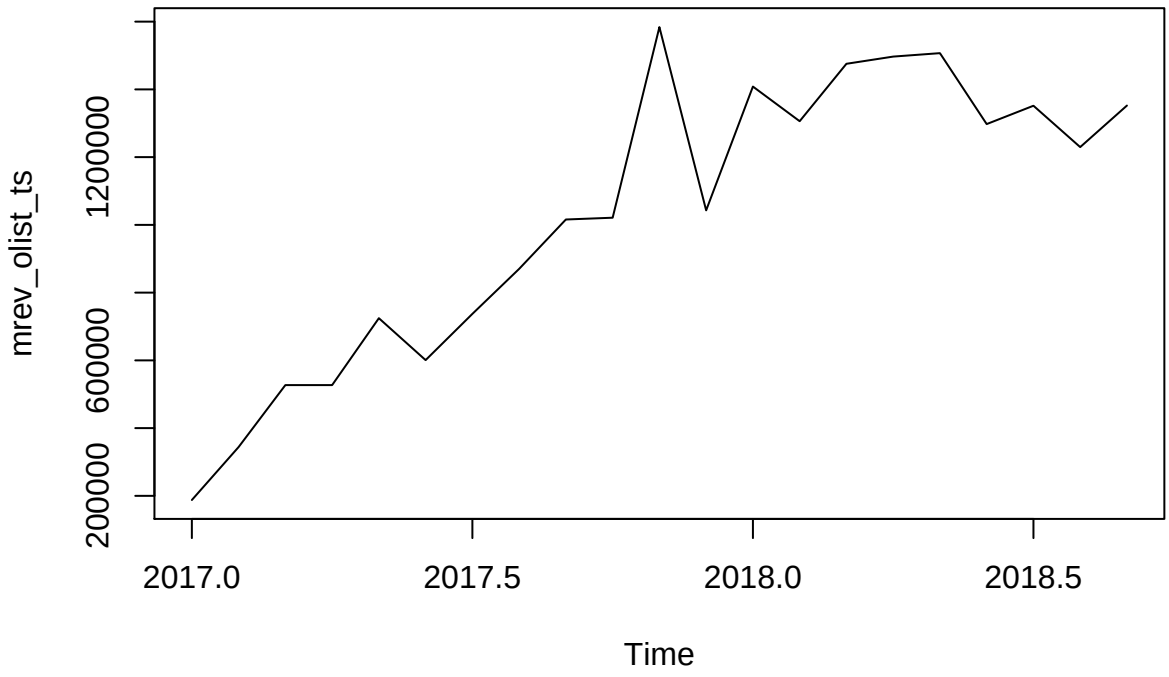


The overall trend of number of increasing customers is positive with month of November in 2017 saw the maximum number of customers.

STARTING WITH FORECASTING

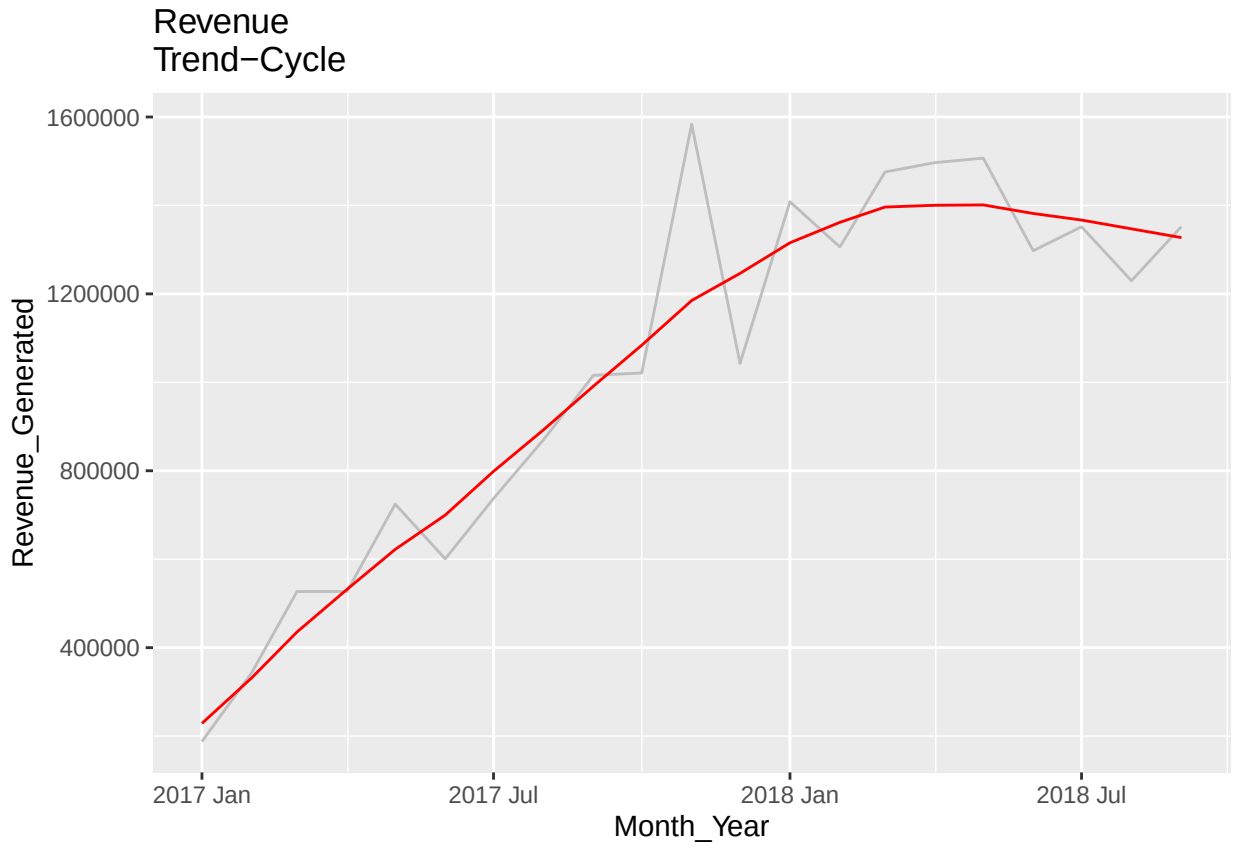
This is the monthly time series of revenue

Plotting the timeseries



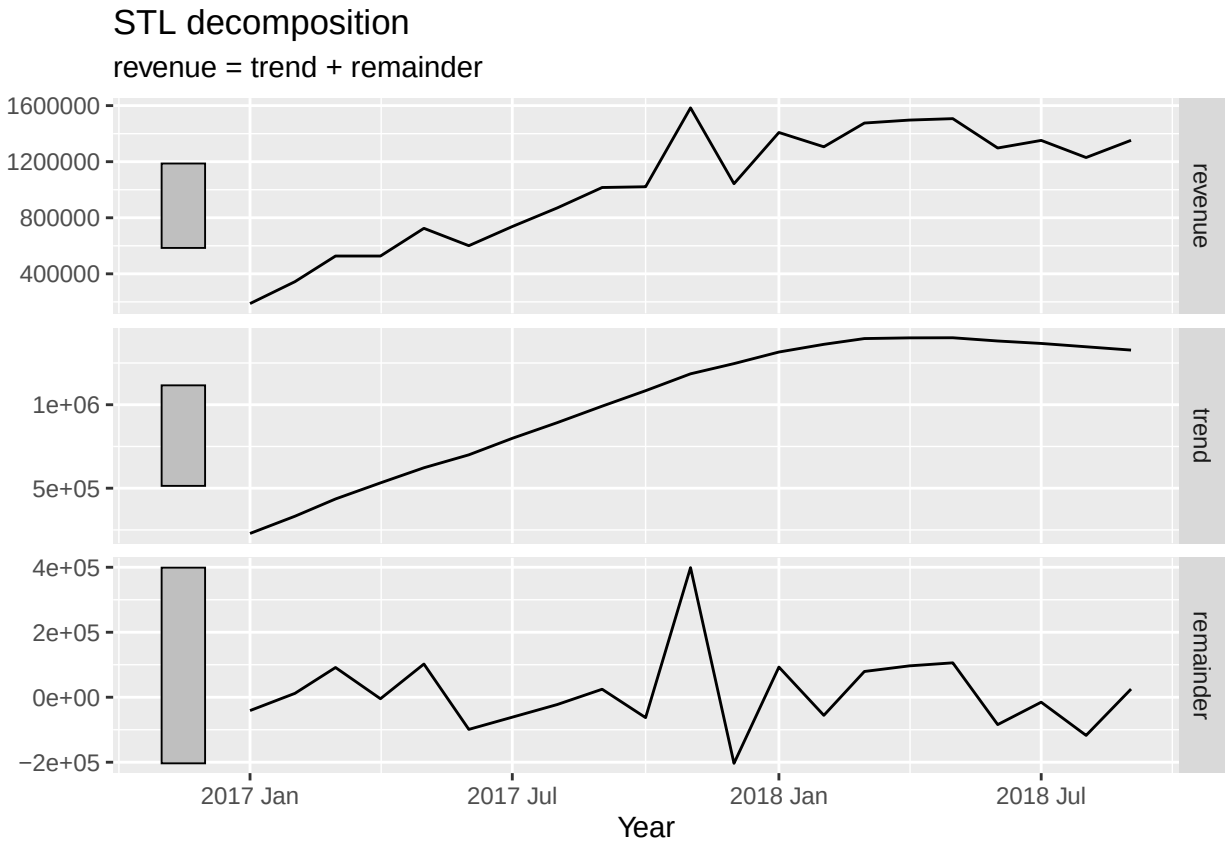
The above plot has a positive trend with some sort of seasonality

Following plot Shows the Trend Cycle

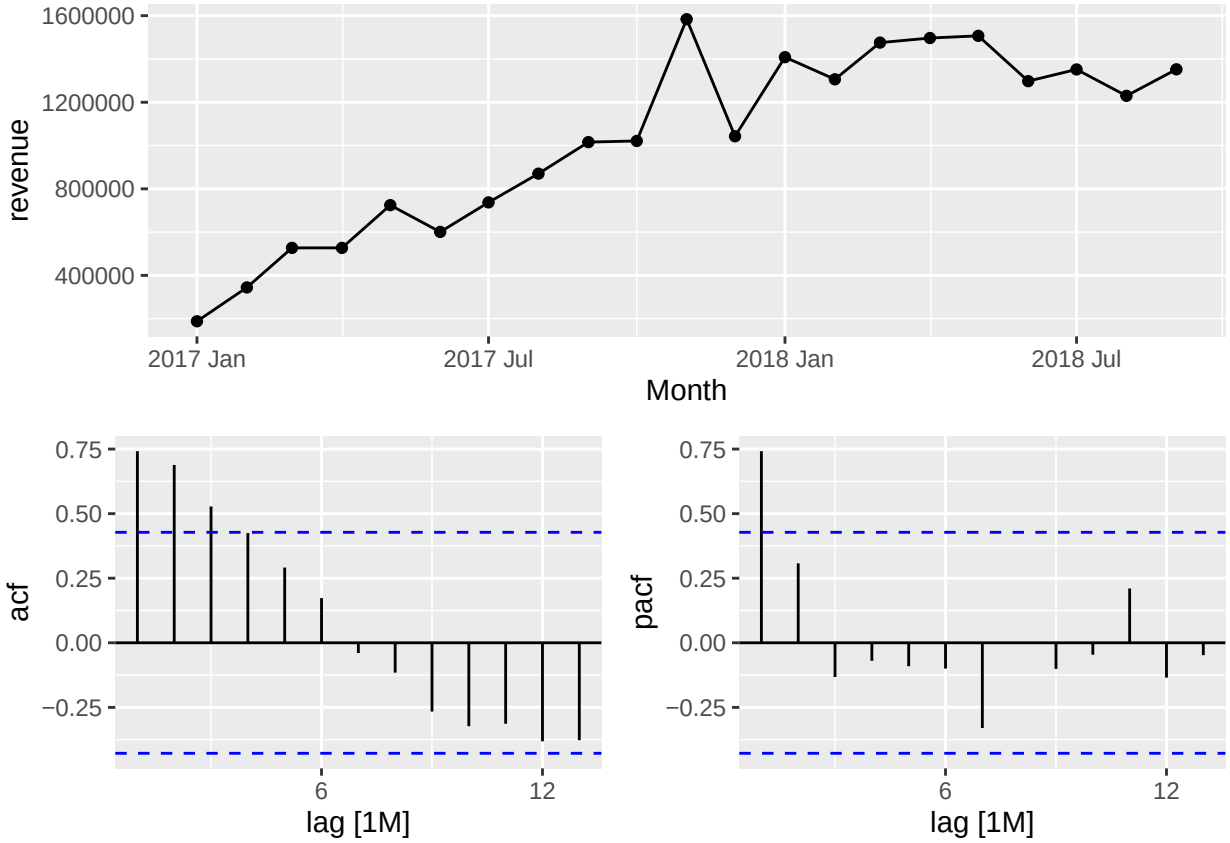


STL decomposition

STL is a versatile and robust method for decomposing time series. STL is an acronym for "Seasonal and Trend decomposition using Loess", while Loess is a method for estimating nonlinear relationships.



Plotting of ACF and PACF



With ACF we can see that there is an exponential decay. There is also correlation in the residuals. This combination of ACF and PACF suggests that the underlying time series follows an autoregressive model.

KPSS shows a differencing is required.

```
rev_olist %>%
  features(revenue, unitroot_kpss)
```

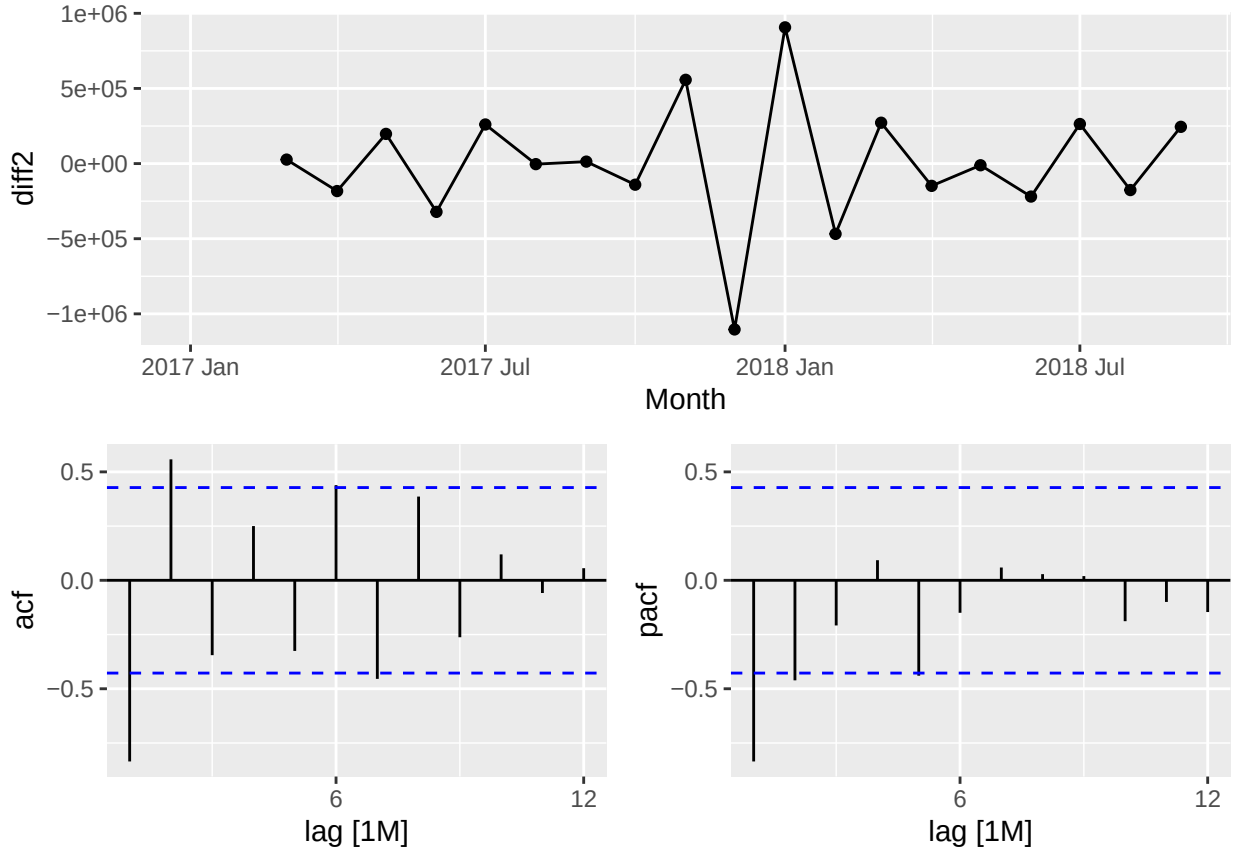
```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1     0.708      0.0128
```

Taking a seasonal difference first and then followed by a normal difference to make the data stationary and stabilize the mean.

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



After differencing twice, we can see that our data is now almost a white noise

Above ACF suggests it is an AR(1) model. So fitting the values of p,d and q manually.

```
fit_manual <- rev_olist %>% model(arima = ARIMA(revenue ~ pdq(1, 2, 0)))
```

```
## Warning: Having 3 or more differencing operations is not recommended. Please
## consider reducing the total number of differences.
```

```
report(fit_manual)
```

```
## Series: revenue
## Model: ARIMA(1,2,0)(0,1,0)[12]
##
## Coefficients:
```

```
##           ar1
##       -0.5345
## s.e.   0.4145
##
## sigma^2 estimated as 2.645e+10:  log likelihood=-96.05
## AIC=196.1   AICc=199.1   BIC=195.99
```

AICc for the manual p,d,and q is 199.1

Force run all combinations

```
auto_cv <- rev_olist %>%
  model(ARIMA(revenue ~ pdq(d=2), stepwise = FALSE, approximation = FALSE)) %>%
  report()
```

```
## Warning: Having 3 or more differencing operations is not recommended. Please
## consider reducing the total number of differences.
```

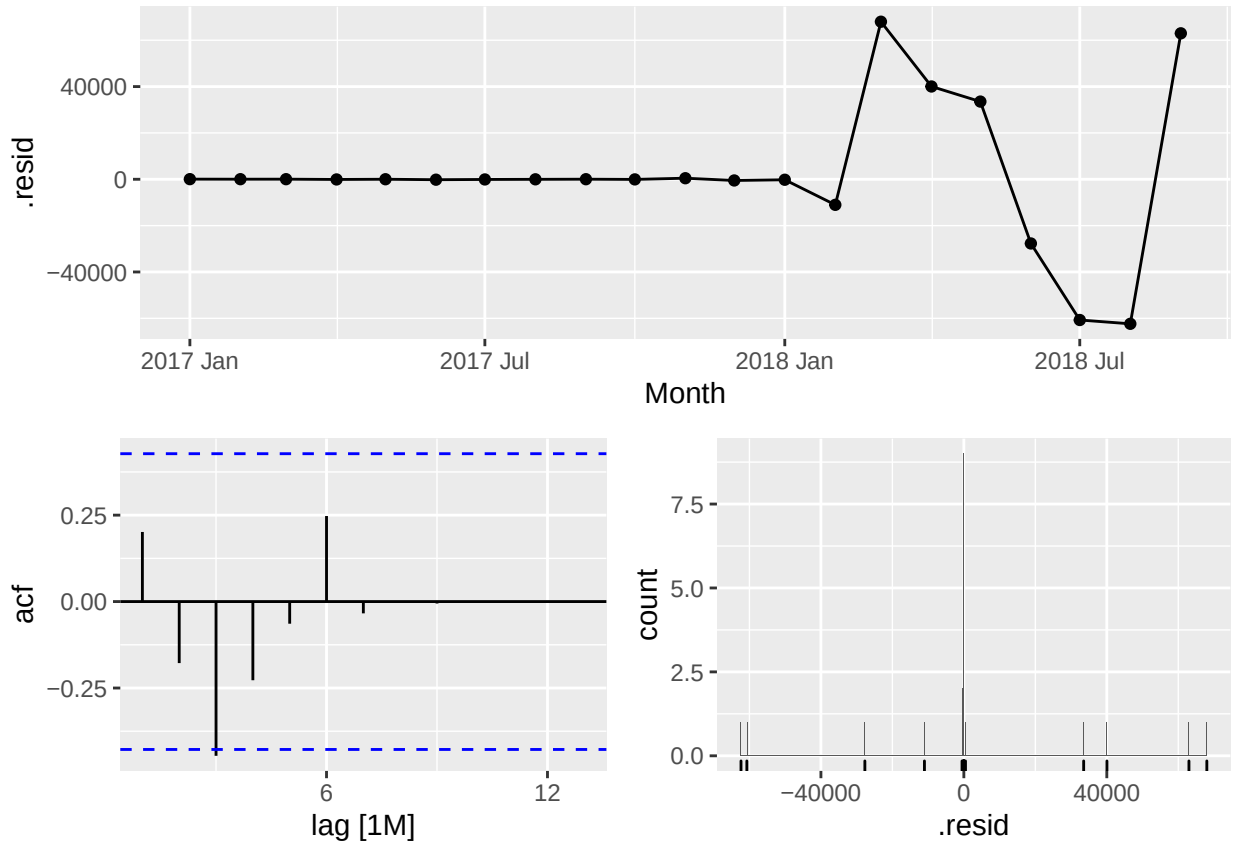
```
## Warning in sqrt(diag(best$var.coef)): NaNs produced
```

```
## Series: revenue
## Model: ARIMA(3,2,3)(0,1,0)[12]
##
## Coefficients:
##           ar1      ar2     ar3      ma1      ma2     ma3
##       -0.4193  -0.5638  0.3617  -0.1872  -0.8803  0.0981
## s.e.   0.2519      NaN  0.2263      NaN      NaN     NaN
##
## sigma^2 estimated as 1.978e+10:  log likelihood=-91.53
## AIC=197.05   AICc=85.05   BIC=196.68
```

```
rev_olist %>%
  model(ARIMA(revenue ~ pdq(d=2))) %>%
  report()
```

```
## Warning: Having 3 or more differencing operations is not recommended. Please
## consider reducing the total number of differences.
```

```
## Series: revenue
## Model: ARIMA(0,2,0)(0,1,0)[12]
##
## sigma^2 estimated as 2.841e+10:  log likelihood=-96.69
## AIC=195.38   AICc=196.18   BIC=195.33
```



There are no lags outside the confidence interval and the residuals are normally distributed. This mean the data values have no correlation left.

ljung box test

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 1 x 3
##   .model                                                    lb_stat lb_pvalue
##   <chr>                                                       <dbl>     <dbl>
## 1 ARIMA(revenue ~ pdq(d = 2), stepwise = FALSE, approximation~    10.7     0.151
```

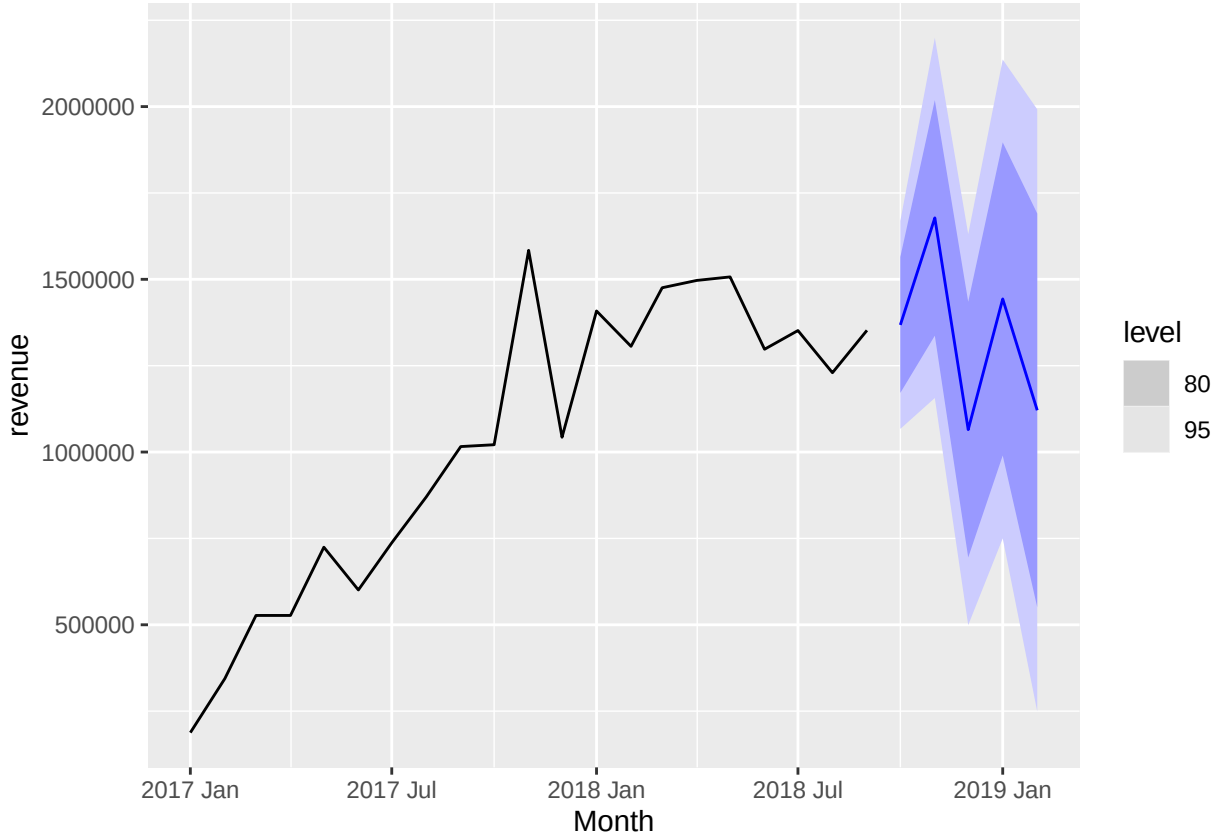
P-value is much greater than 0.05 hence, we can conclude that data values are independent.

```
auto_cv %>%
  forecast(h = 5)
```

```
## Warning: `...` is not empty.
```

```
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A fable: 5 x 4 [1M]
## # Key:     .model [1]
##   .model                                          Month             revenue   .mean
##   <chr>                                           <mth>              <dist>   <dbl>
## 1 ARIMA(revenue ~ pdq(d = 2), stepwise = F~ 2018 Oct N(1368056, 2.4e+10)  1.37e6
## 2 ARIMA(revenue ~ pdq(d = 2), stepwise = F~ 2018 Nov N(1677809, 7.1e+10)  1.68e6
## 3 ARIMA(revenue ~ pdq(d = 2), stepwise = F~ 2018 Dec N(1064850, 8.3e+10)  1.06e6
## 4 ARIMA(revenue ~ pdq(d = 2), stepwise = F~ 2019 Jan N(1443186, 1.3e+11)  1.44e6
## 5 ARIMA(revenue ~ pdq(d = 2), stepwise = F~ 2019 Feb   N(1120816, 2e+11)  1.12e6
```




```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 1 x 9
##   .model                      .type    ME   RMSE    MAE    MPE  MAPE   MASE  ACF1
##   <chr>                       <chr> <dbl>  <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl>
## 1 ARIMA(revenue ~ pdq(d = 2~ Trai~ 2011. 30692. 17529. 0.0751  1.28 0.0229 0.201
```

ETS method

Holt's Method

ETS(A,A,N)

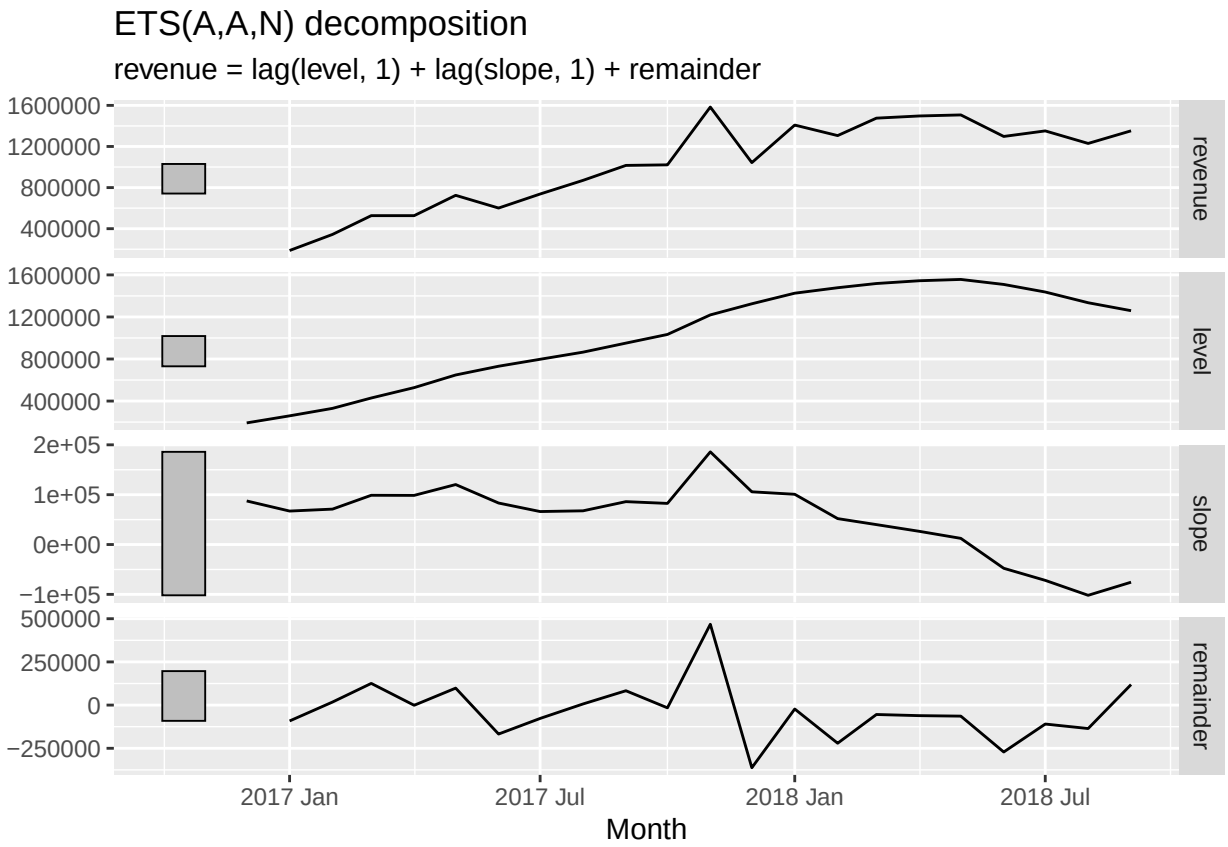
```
## Series: revenue
## Model: ETS(A,A,N)
##   Smoothing parameters:
##     alpha = 0.2210484
##     beta  = 0.221048
##
##   Initial states:
##         l        b
##  192170.7 87427.88
##
##   sigma^2:  35525306131
##
##      AIC     AICc      BIC
## 579.6612 583.6612 584.8838
```

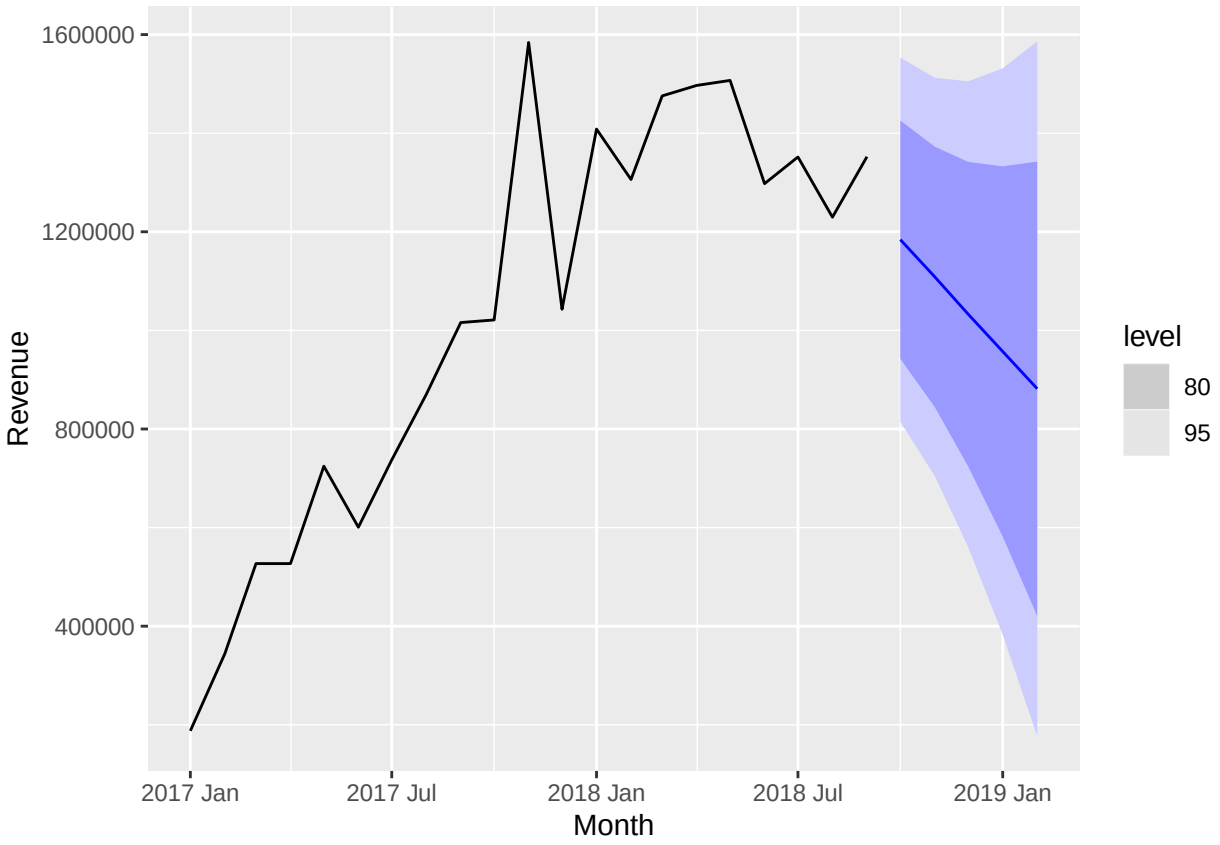
ETS(A,N,A)

```
## Series: revenue
## Model: ETS(A,N,A)
##   Smoothing parameters:
##     alpha = 0.844217
##     gamma = 0.0001000081
##
##   Initial states:
##        l        s1       s2       s3       s4       s5        s6        s7
##  1044208 -56493.94 547972.6 48726.19 106859.3 24568.31 -44790.34 -117876.2
##        s8        s9       s10     s11       s12
##  69326.98 -64761.56 -1310.199 -120684 -391537.2
##
##   sigma^2:  134041930898
##
##      AIC     AICc      BIC
## 608.9139 704.9139 624.5817
```

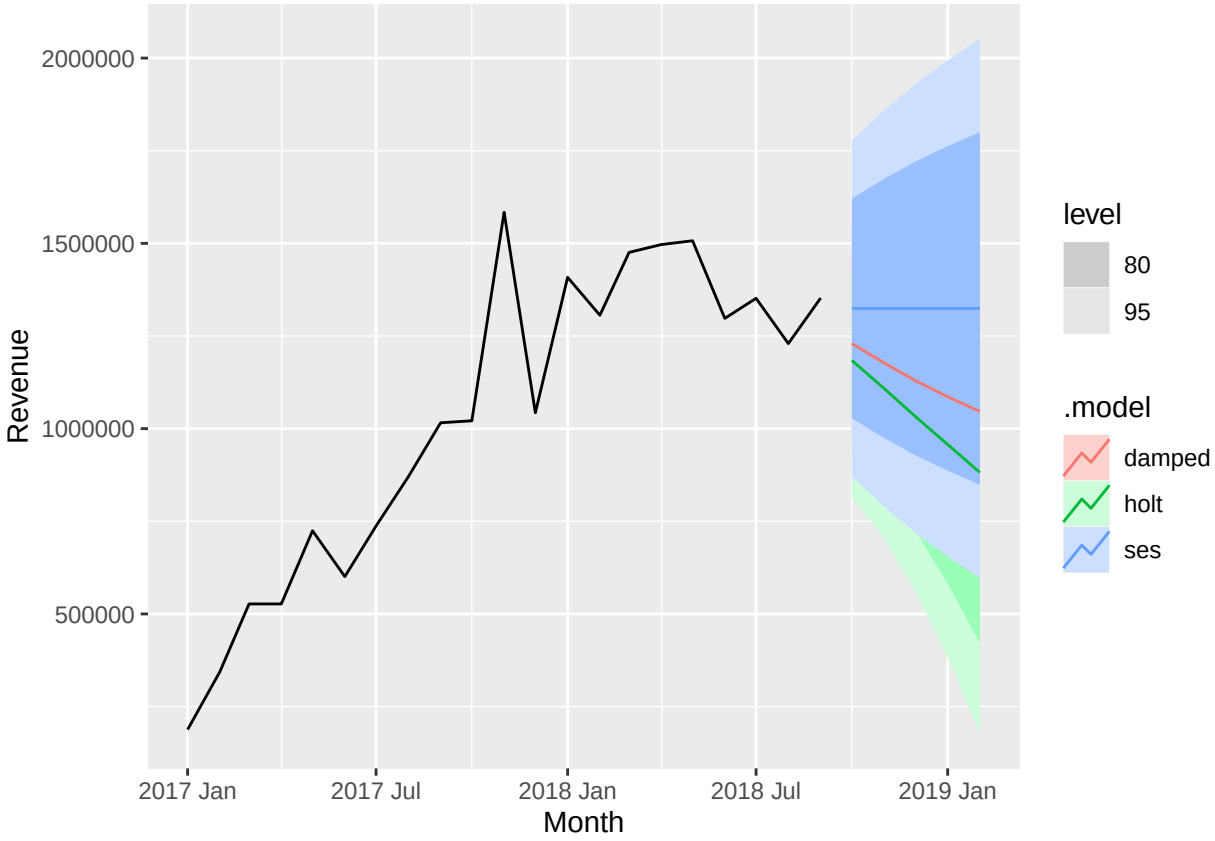
```
components(ets_fit) %>% autoplot()
```

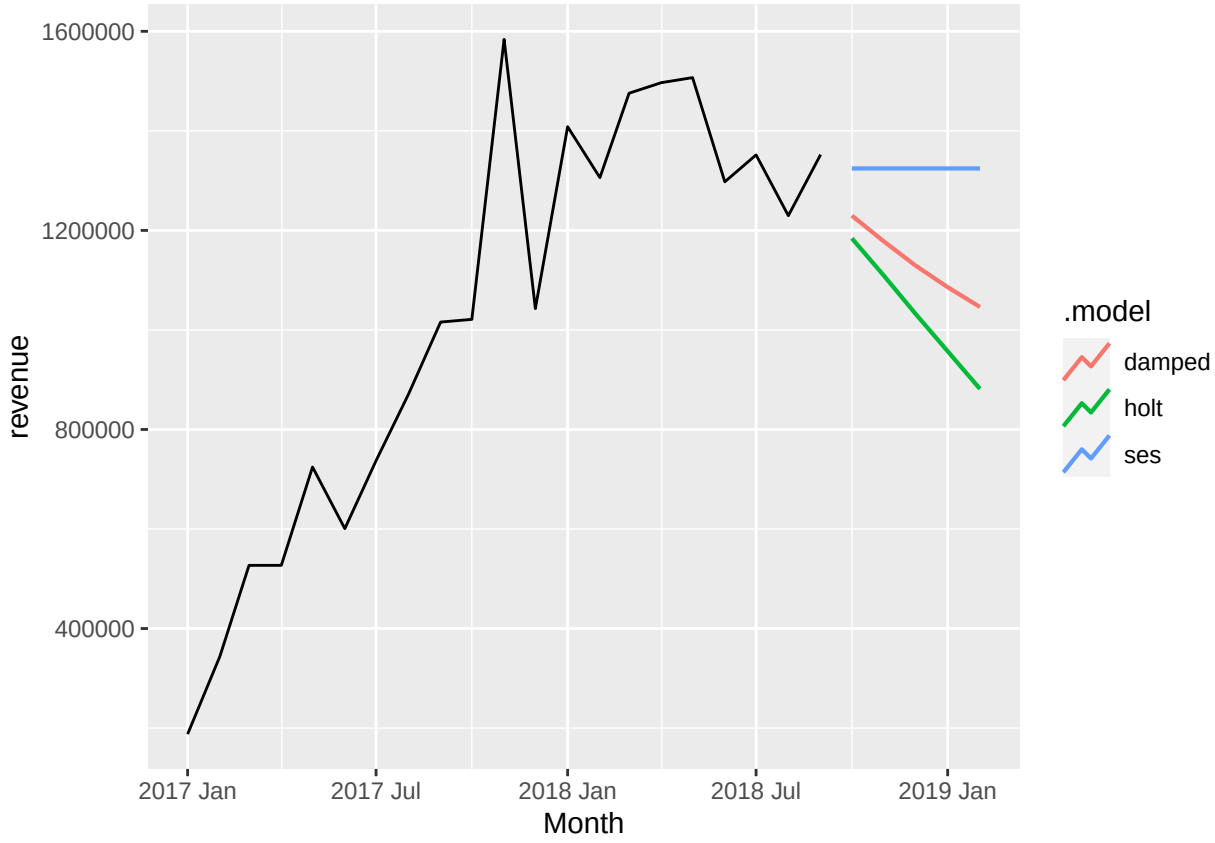
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```


ETS(A,A,N) decomposition
revenue = lag(level, 1) + lag(slope, 1) + remainder
1600000
1200000
800000
400000
revenue
level
2e+05
1e+05
0e+00
−1e+05
slope
500000
250000
0
−250000
remainder
2017 Jan
2017 Jul
2018 Jan
2018 Jul
Month


1600000
1200000
800000
400000
Revenue
2017 Jan
2017 Jul
2018 Jan
2018 Jul
2019 Jan
Month
level
80
95

Comparing SES, Holt's Method, and Damped Holt's Method





Comparing the errors for all the 3 methods:

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 3 x 9
##   .model .type         ME    RMSE     MAE   MPE  MAPE  MASE   ACF1
##   <chr>  <chr>      <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ses    Training  50764. 219916. 164825. -4.60  25.8 0.215 -0.208
## 2 holt   Training -35123. 169584. 122536. -4.98  13.5 0.160 -0.227
## 3 damped Training  -9431. 166402. 118018. -1.81  13.4 0.154 -0.197
```

Damped method gives the least error.

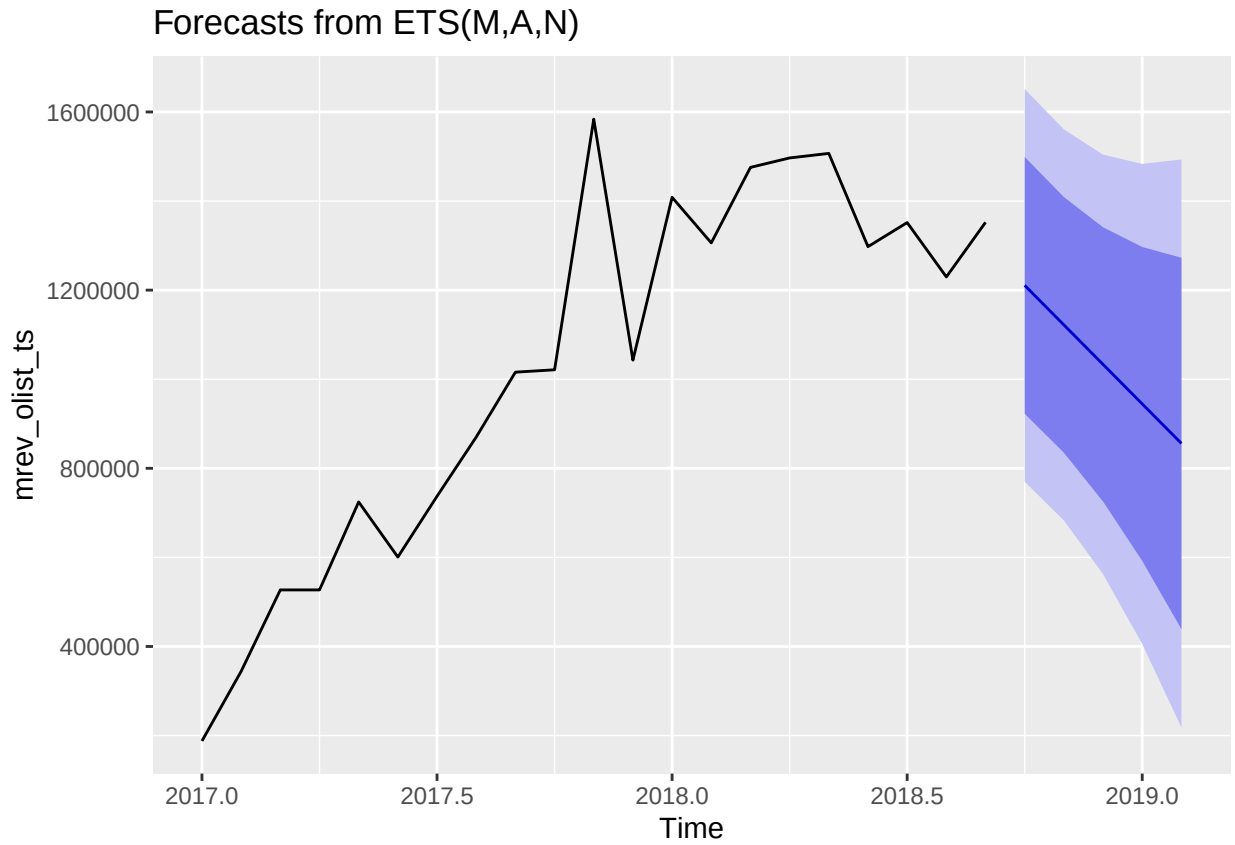
Letting automatic ets() function to choose the best model

```
## ETS(M,A,N)
##
## Call:
```

```
##  ets(y = mrev_olist_ts)
##
##   Smoothing parameters:
##     alpha = 0.179
##     beta  = 0.1789
##
##   Initial states:
##     l = 174700.9185
##     b = 87428.8417
##
##   sigma:  0.1858
##
##      AIC     AICc      BIC
## 576.9920 580.9920 582.2146
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set -46873.1 171922.8 126463.9 -5.447873 13.40152 0.1651765 -0.1330138
```

ets() gives ETS(M,A,N) as the model with better AICc. M is multiplicative error, A is additive trend and N is with none seasonality. It is a Multiplicative Holt-Winter's method with additive errors

Forecasting with ETS(M,A,N) model

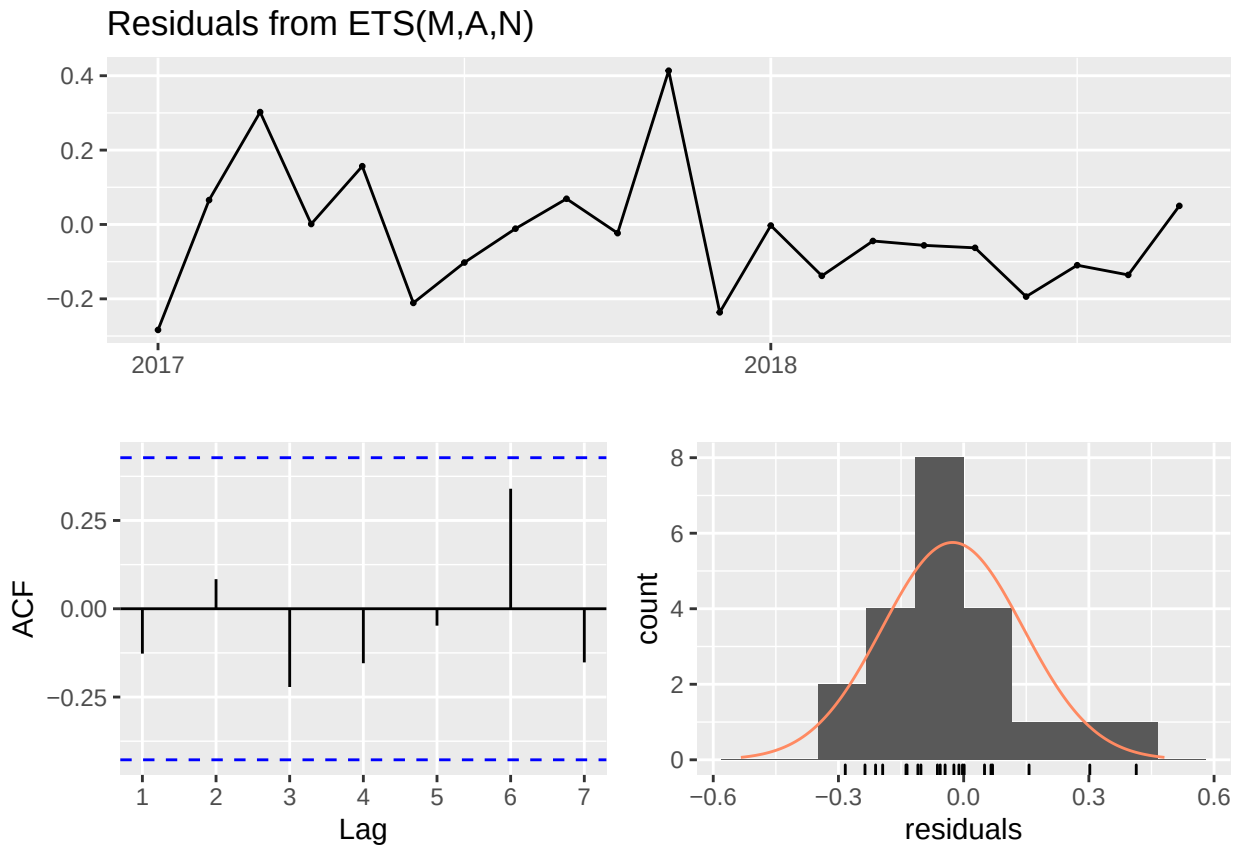



```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
```

```
## Oct 2018      1210681.9 922439.3 1498924 769852.9 1651511
## Nov 2018      1121966.6 834976.7 1408957 683053.3 1560880
## Dec 2018      1033251.4 725369.8 1341133 562387.1 1504116
## Jan 2019       944536.1 592169.5 1296903 405637.9 1483434
## Feb 2019       855820.9 438986.1 1272656 218327.2 1493315
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = mrev_olist_ts)
##
##   Smoothing parameters:
##     alpha = 0.179
##     beta  = 0.1789
##
##   Initial states:
##     l = 174700.9185
##     b = 87428.8417
##
##   sigma:  0.1858
##
##      AIC     AICc      BIC
## 576.9920 580.9920 582.2146
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set -46873.1 171922.8 126463.9 -5.447873 13.40152 0.1651765 -0.1330138
```

We can see that damped model and ETS(M,A,N) are almost similar with damped method has very little error difference from the latter.




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,N)
## Q* = 7.1444, df = 3, p-value = 0.06743
##
## Model df: 4.   Total lags used: 7
```

Performing cross validation on the time series for ETS and Arima.

```
## [1] 393166.4
```

```
## [1] 421376.7
```

Henc, we can infer that ETS model is better in terms of error than ARIMA.

CONCLUSION ON FORECASTING:

For ARIMA, the data was made stationary with seasonal differencing at the lag 12 as the timeseries has monthly data values followed by the normal differencing. Automatic ARIMA model with stepwise and approximation equal to false returns the lowest AICc values.

Futher for ETS:

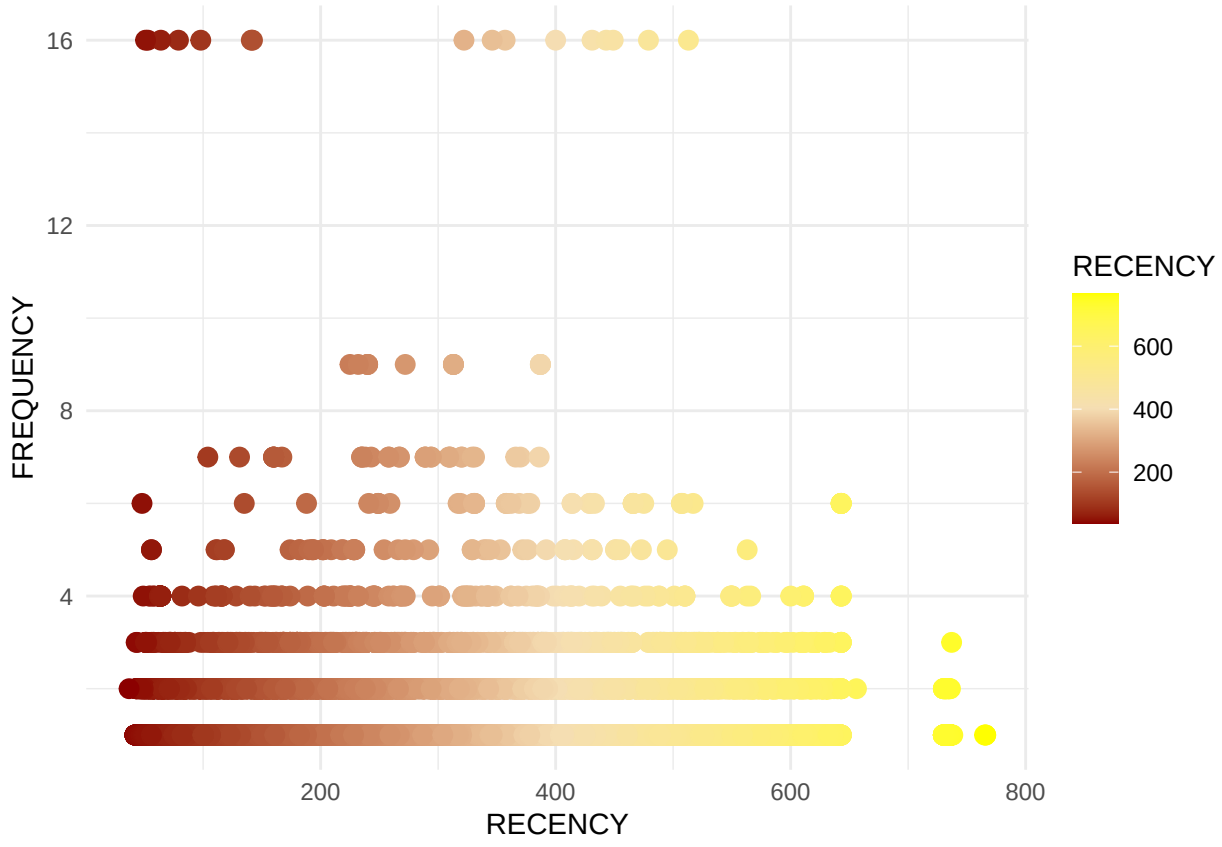
I went ahead and tried ses, holt and damped method. Among all the three models Damped one had the least errors and was better of them. Next, I opted for ets() function to choose an optimal model based on the data values and it returned ETS(M,A,N) model with errors values almost similar to the damped method. The tsCV() for the ETS gives lesser error than ARIMA. Therefore, conncluding that ETS is performing better on the given time series data values.

Recenecy, Frequency and Monetary Value

RFM (recency, frequency, monetary) analysis is a behavior based technique used to segment customers by examining their transaction history such as

How recently a customer has purchased (RECENCY)? How often they purchase (FREQUENCY)? How much the customer spends (MONETARY_VALUE)?

Why is Customer segmentation Important? Ans: General mass marketing is often expensive, time consuming and sometimes not that responsive. Focussing on particulare segment of customers helps to prevent customer churn, build customer loyalty and also increasing the brand equity.



rankMonetary columns

recency rank colums

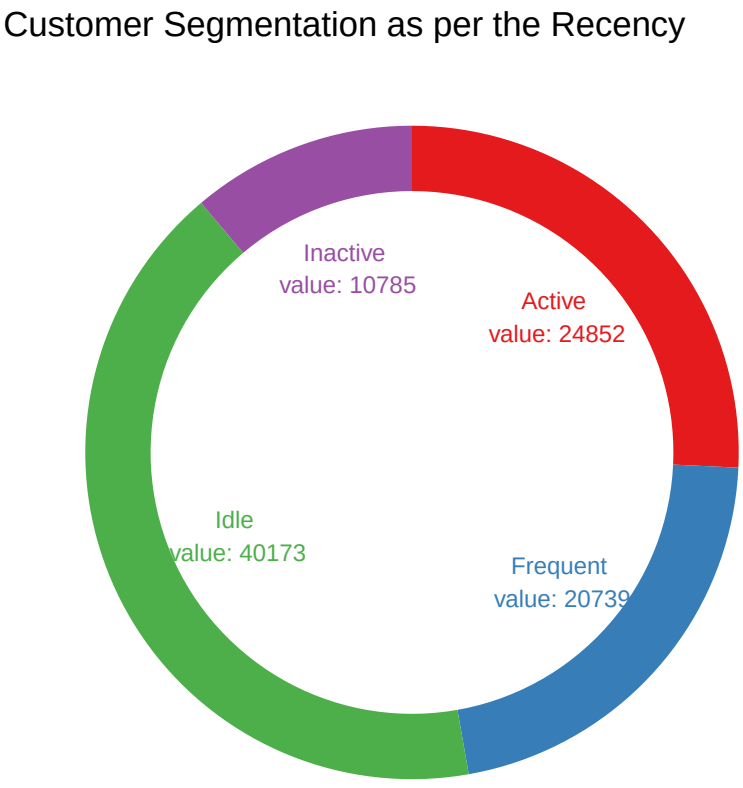
```
RFM_Scores[RFM_Scores$CustomerID == 2785,]
```

```
##      CustomerID Recency_rank Frequency_rank Monetary_rank
## 3407       2785            3              1             3
```

Segmenting the customers based on their Rececny Value

Following is the Segmentation of customers based on their Recency.

```
##   r_segment CustomerID
## 1    Active      24852
## 2  Frequent      20739
## 3      Idle      40173
## 4  Inactive      10785
```

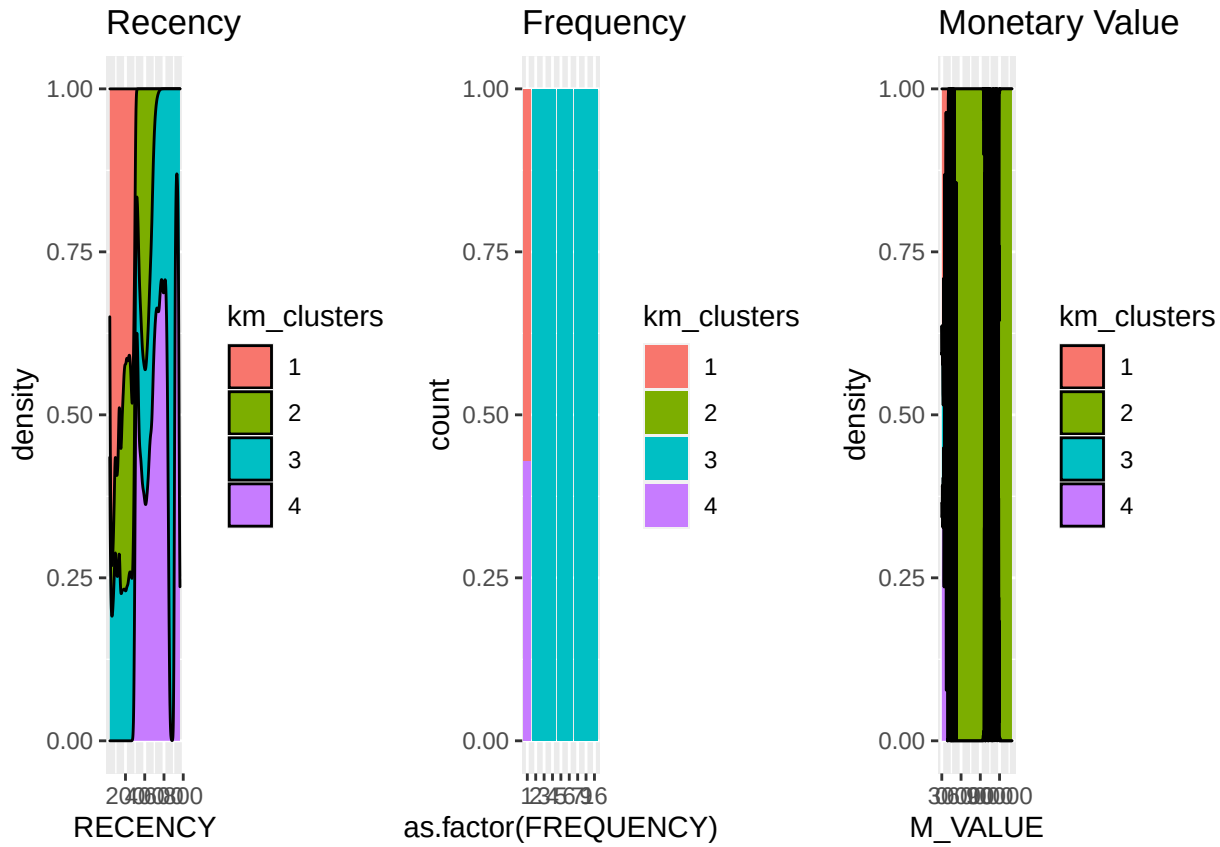


We can see that there are maximum number of Idle customers.

Finding Optimal number of K for K means

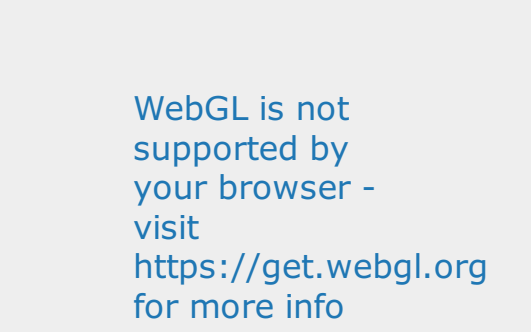
```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5880050)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5880050)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5880050)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5880050)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5880050)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 5880050)
```

conditional density and frequency plots:




```
final_seg <- plotly::plot_ly(RFM_df, x = ~RECENCY, y = ~M_VALUE, z = ~FREQUENCY, color = ~km_clusters)
final_seg
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



Above plot shows the clusters of the segmented customers based on their RFM values. (This may not get displayed in the knitted PDF. However, I have created a flexdashboard which contains all the outputs)

Following are the segments/clusters of customer based on RFM values:

Concluding with the above segmented clusters:

Cluster 1: Here the customers have moderate Recency, high Frequency and moderate Monetary value. This segment of the customers are one of the best and loyal customers. Rewarding them with a ways of exciting offers can result in improving the Recency and Monetary value further more.

Cluster 2: This segment has the customers with the lowest Recency, Frequency and Monetary Value. These are the customers are on the verge of churning. The churn can be avoided by offering one time free offers and preimum services trial periods.

Cluster 3: This cluster is similar to the cluster 1 but has a higher Recency value.

Cluster 4: Customers of this segments are one time buyers, which represents the largest segment Olist has. It has preety high value of monetary value. Tailored promotions like giving cashback and limited premium services to increase their frequency.

APPENDIX

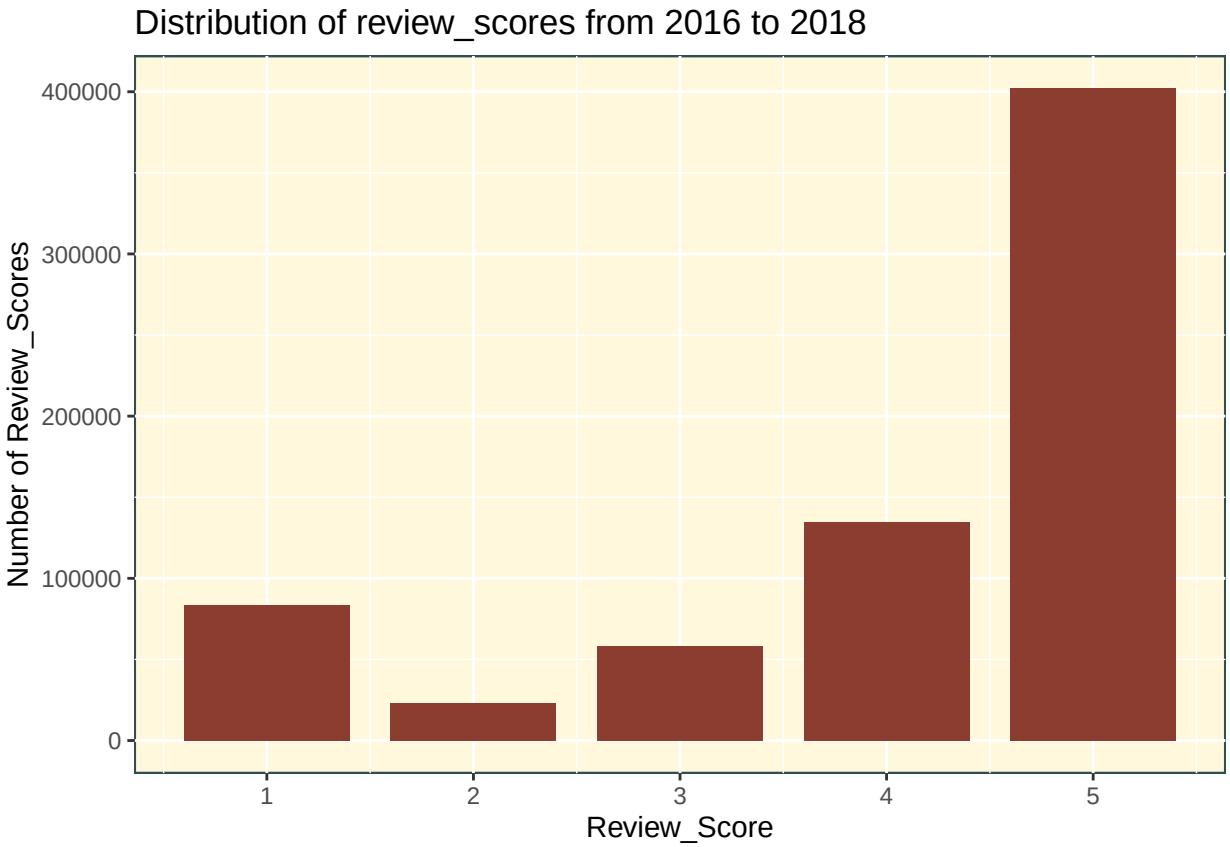
Although the following analyses is not relevant to the algorithms applied, it still has many important insights with respect to customer reviews and information on sellers.

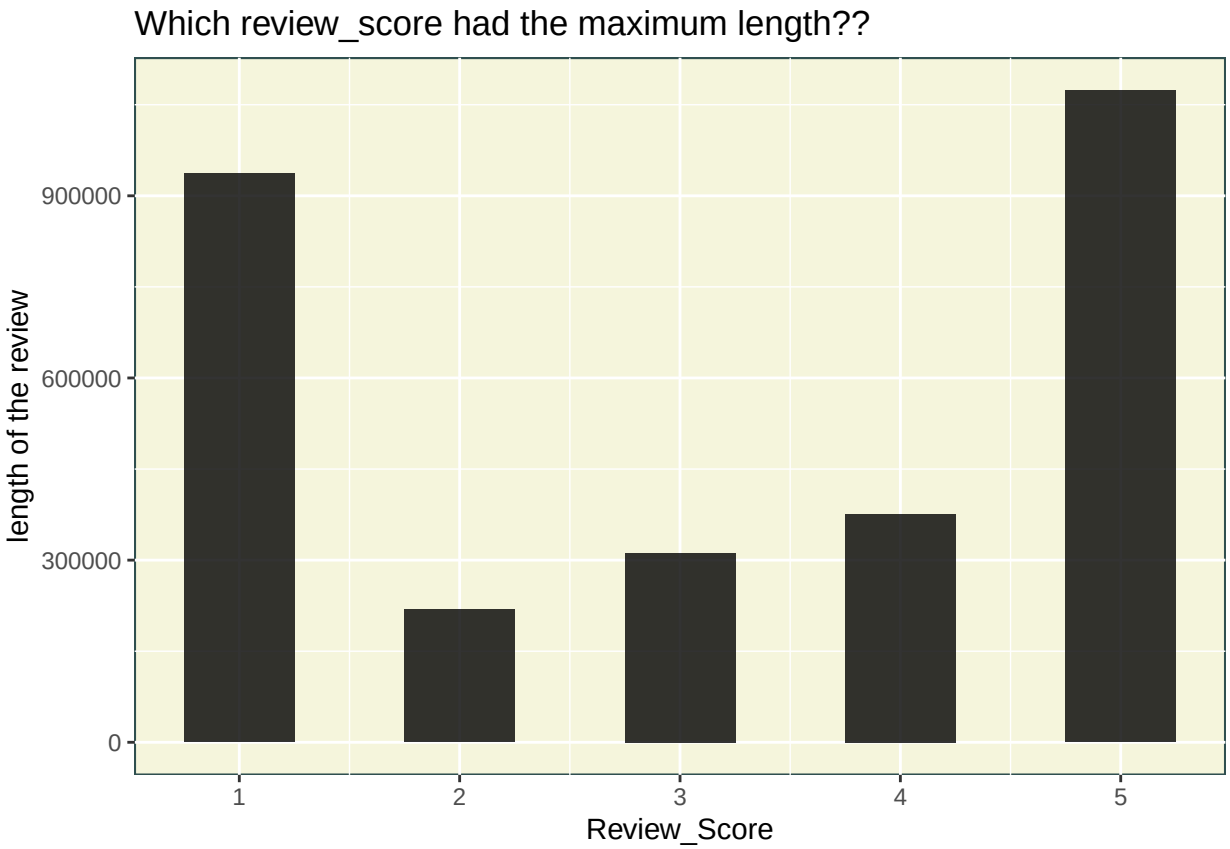
Let's explore the reviews and reviews_score given by the cutomers.

Review score of 5 was the highest which was given by 57000 customers.

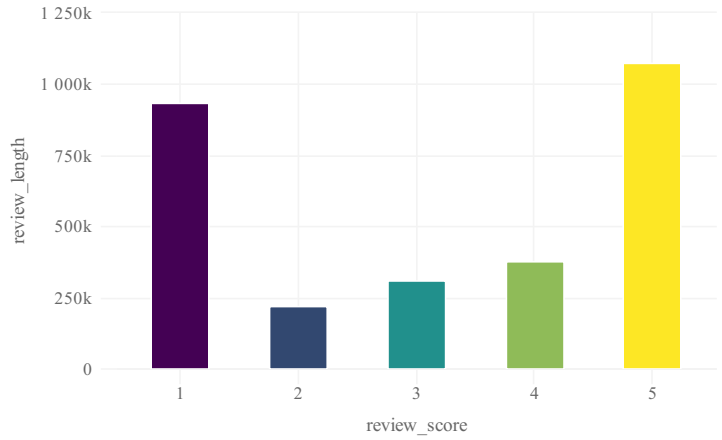
Whar percent score is 5?

What is the average review_score?




Which review_score had the maximum length??
length of the review
900000
600000
300000
0
1
2
3
4
5
Review_Score

Which review_score had the maximum length??

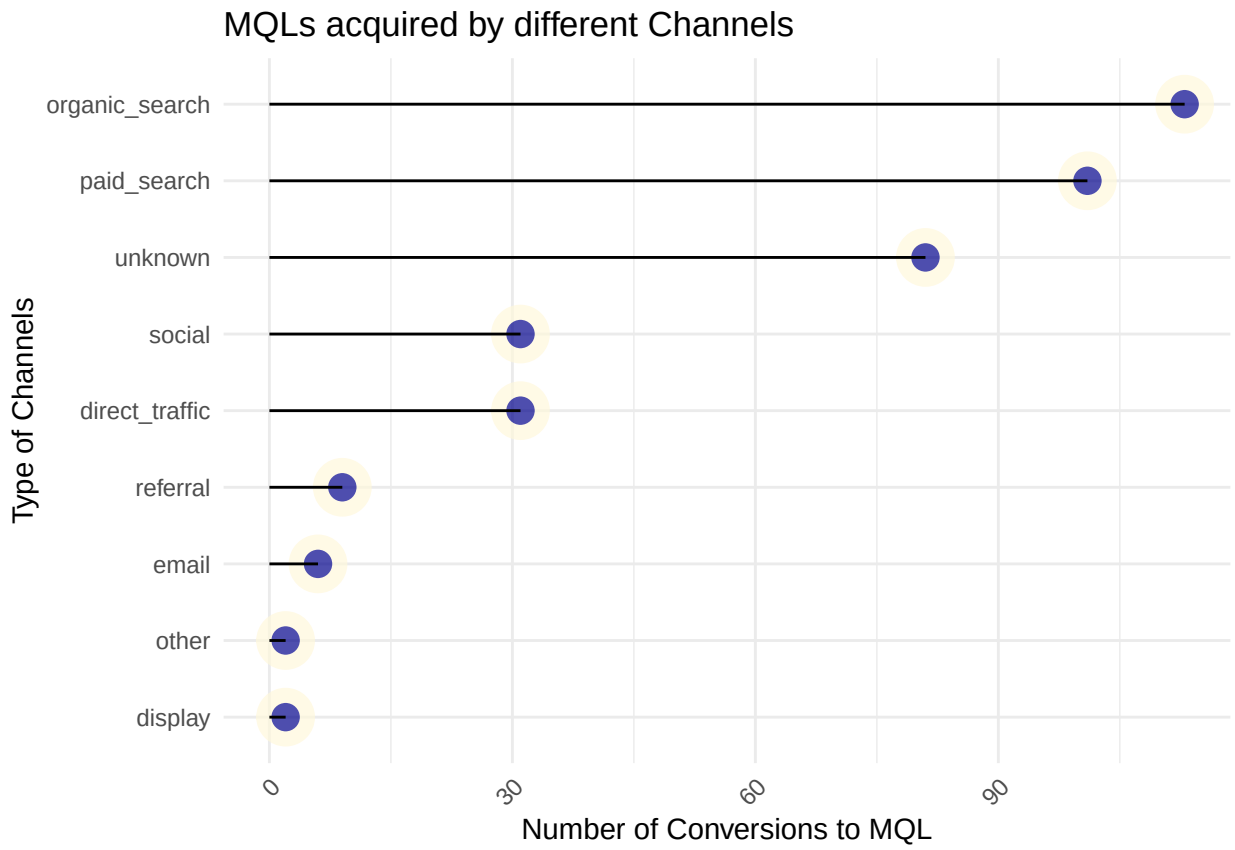


Review_score of 5 had the highest review length

MQL(Marketing Qualified Lead) INSIGHTS

The term 'Marketing Qualified Lead(MQL)' means a potential reseller/manufacturer who has an interest in selling their products on Olist. Olist acquired sellers through various different marketing channels. Let's find out which channel was the most effective in the lead generation.

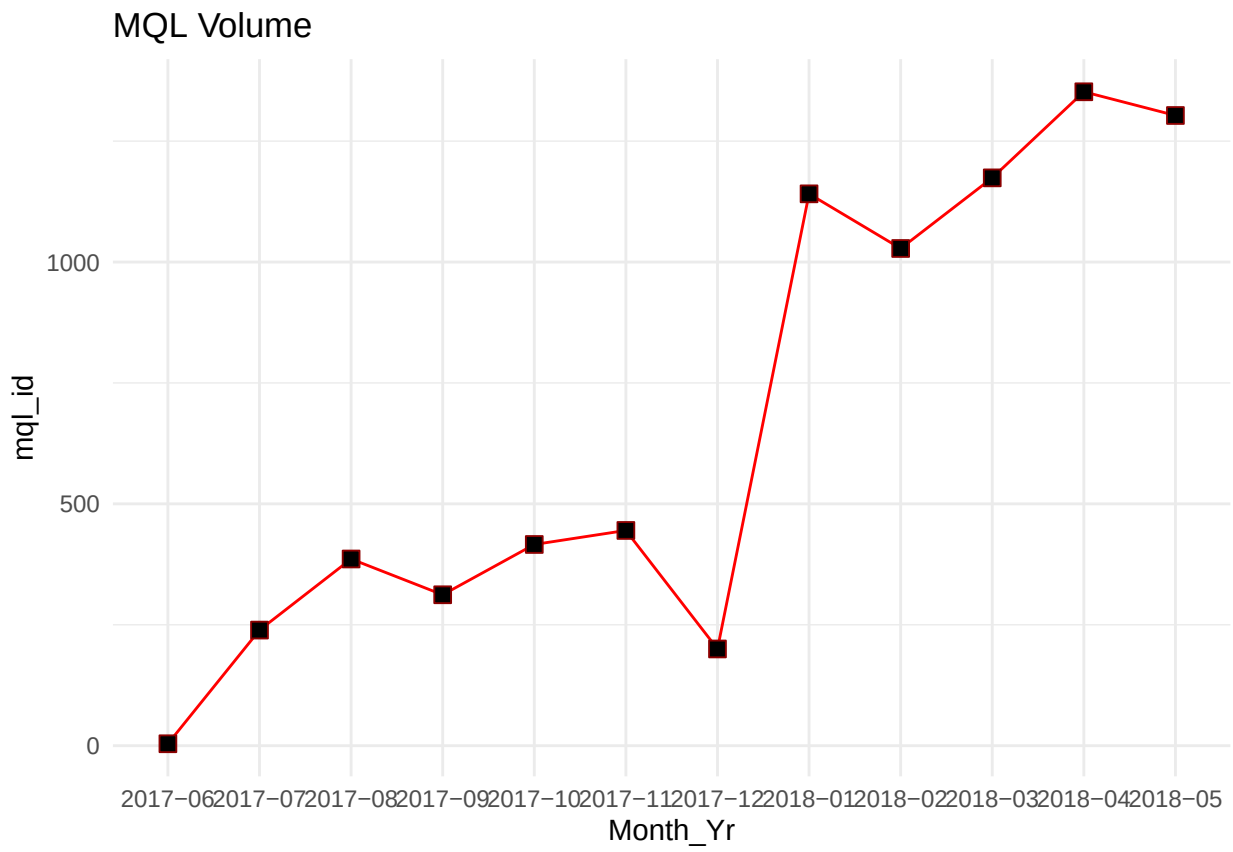
We can see that Organic Search followed by the paid search generated most number of MQLs.



MARKETING CHANNEL EFFECTIVENESS

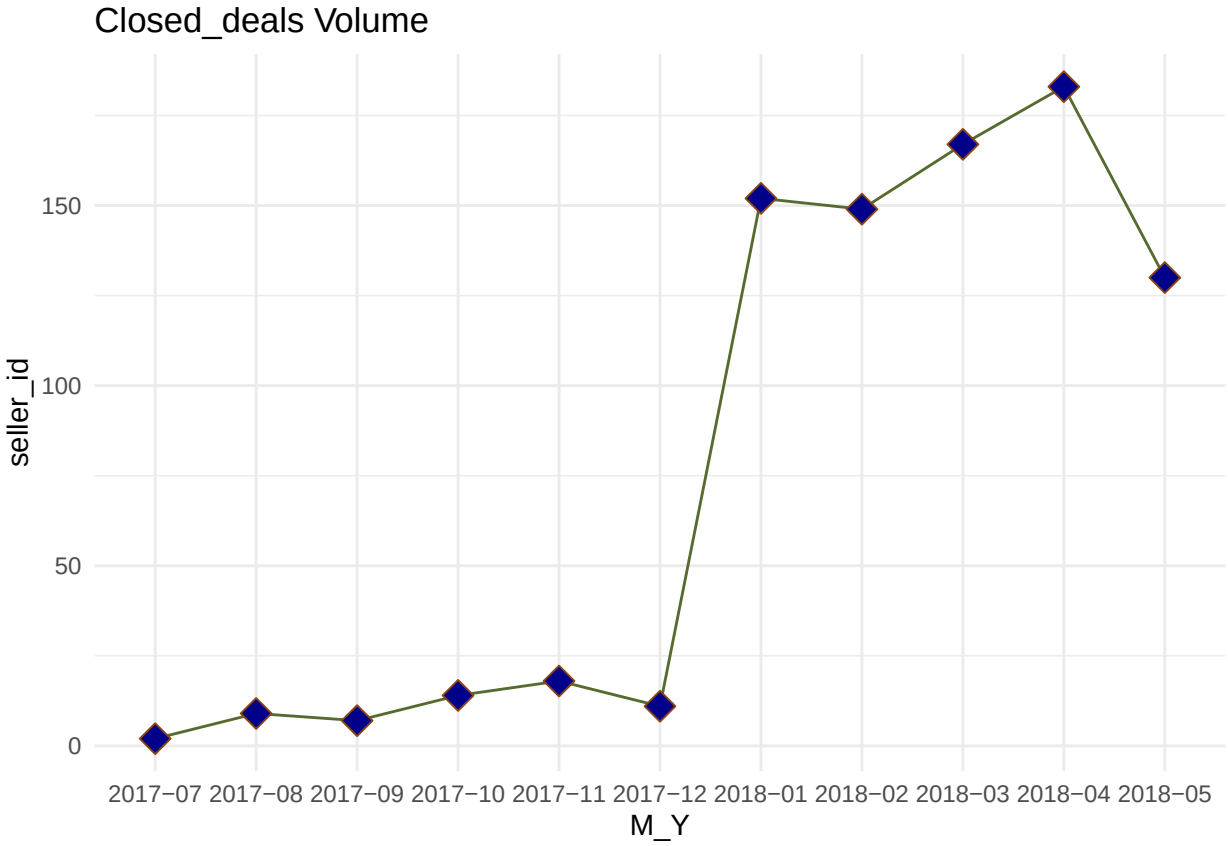
Let's see the number of MQLs generated over the time

Following plot shows the timeseries of MQLs acquired by the Olist



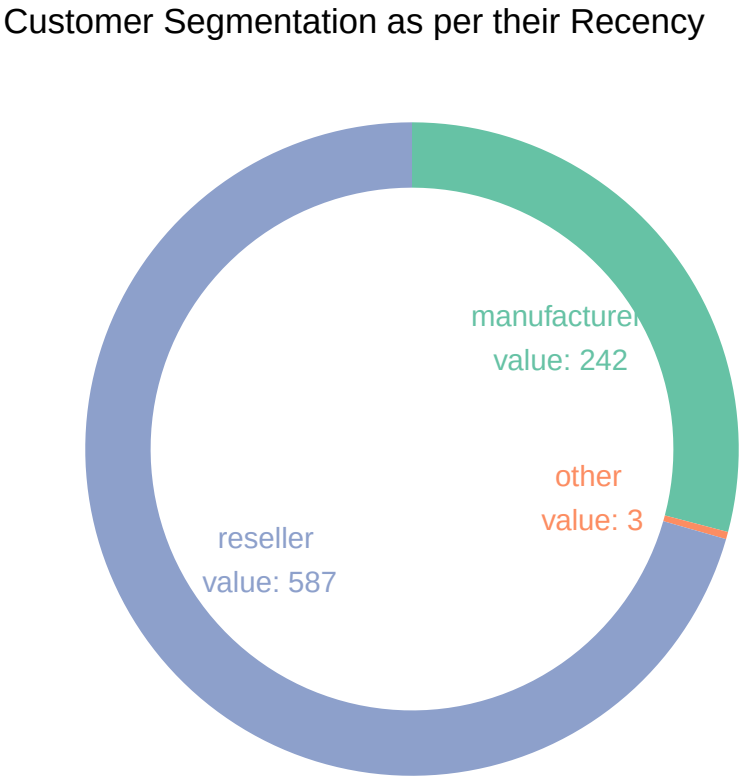
We can infer that the MQL volume grew maximum in January of 2018

Now let's take a look at the Closed deals. A MQL who eventually signs up for the Seller portfolio is called as a closed deal.



Conversion rate also increased with the MQL volume.

Following Doghnut chart describes the business types of sellers



References:

1. https://www.kaggle.com/olistbr/brazilian-ecommerce
2. Book Rob J Hyndman and George Athanasopoulos. Forecasting: Principles and Practice. 3rd edition, 2020. OTexts. https://otexts.com/fpp3/.
3. https://towardsdatascience.com/exploring-highcharts-in-r-f754143efda7