Final Project

Roberto Cancel

11/30/2021

## Import Libraries

#Import Libraries  
library(readr) # for quick importing of files  
library(readxl) # for importing excel files  
library(dplyr) # for easy data manipulation  
library(ggplot2) # for generating plots  
library(forecast) # for time series analysis  
library(knitr) # for table generation and other functions

## Import the data

#read csv and excel files. Suppress warnings due to size of raw datasets  
properties\_2016 <- read\_csv("properties\_2016.csv", col\_types = cols())  
properties\_2017 <- read\_csv('properties\_2017.csv', col\_types = cols())  
data\_dict <- read\_excel("zillow\_data\_dictionary.xlsx")  
train\_2016 <- read\_csv("train\_2016\_v2.csv", col\_types = cols())  
test\_2017 <- read\_csv("train\_2017.csv", col\_types = cols())

# Exploratory Data Analysis

## View the data

head(properties\_2016) # 2016 parcel id and housing attributes (described in data\_dict) for ALL houses in this market

## # A tibble: 6 x 58  
## parcelid airconditioning~ architecturalst~ basementsqft bathroomcnt bedroomcnt  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 10754147 NA NA NA 0 0  
## 2 10759547 NA NA NA 0 0  
## 3 10843547 NA NA NA 0 0  
## 4 10859147 NA NA NA 0 0  
## 5 10879947 NA NA NA 0 0  
## 6 10898347 NA NA NA 0 0  
## # ... with 52 more variables: buildingclasstypeid <dbl>,  
## # buildingqualitytypeid <dbl>, calculatedbathnbr <dbl>, decktypeid <dbl>,  
## # finishedfloor1squarefeet <dbl>, calculatedfinishedsquarefeet <dbl>,  
## # finishedsquarefeet12 <dbl>, finishedsquarefeet13 <dbl>,  
## # finishedsquarefeet15 <dbl>, finishedsquarefeet50 <dbl>,  
## # finishedsquarefeet6 <dbl>, fips <chr>, fireplacecnt <dbl>,  
## # fullbathcnt <dbl>, garagecarcnt <dbl>, garagetotalsqft <dbl>, ...

### Comment: Contains parcel id and home attribues for ALL houses in this market. We note extensive missing values in properties\_2016.

head(properties\_2017) # 2017 parcel id and housing attributes (described in data\_dict)

## # A tibble: 6 x 58  
## parcelid airconditioning~ architecturalst~ basementsqft bathroomcnt bedroomcnt  
## <dbl> <dbl> <dbl> <lgl> <dbl> <dbl>  
## 1 10754147 NA NA NA 0 0  
## 2 10759547 NA NA NA 0 0  
## 3 10843547 NA NA NA 0 0  
## 4 10859147 NA NA NA 0 0  
## 5 10879947 NA NA NA 0 0  
## 6 10898347 NA NA NA 0 0  
## # ... with 52 more variables: buildingclasstypeid <dbl>,  
## # buildingqualitytypeid <dbl>, calculatedbathnbr <dbl>, decktypeid <dbl>,  
## # finishedfloor1squarefeet <dbl>, calculatedfinishedsquarefeet <dbl>,  
## # finishedsquarefeet12 <dbl>, finishedsquarefeet13 <dbl>,  
## # finishedsquarefeet15 <dbl>, finishedsquarefeet50 <dbl>,  
## # finishedsquarefeet6 <dbl>, fips <chr>, fireplacecnt <dbl>,  
## # fullbathcnt <dbl>, garagecarcnt <dbl>, garagetotalsqft <dbl>, ...

### Comment: Contains parcel id and home attributes for ALL houses in this market. We note extensive missing values in properties\_2017.

head(train\_2016)

## # A tibble: 6 x 3  
## parcelid logerror transactiondate  
## <dbl> <dbl> <date>   
## 1 11016594 0.0276 2016-01-01   
## 2 14366692 -0.168 2016-01-01   
## 3 12098116 -0.004 2016-01-01   
## 4 12643413 0.0218 2016-01-02   
## 5 14432541 -0.005 2016-01-02   
## 6 11509835 -0.270 2016-01-02

### Comment: Contains parcel id, log error, and transaction date - needs to be joined with properties\_2016. After joining only the properties that sold (have logerror values) will be retained.

head(test\_2017)

## # A tibble: 6 x 3  
## parcelid logerror transactiondate  
## <dbl> <dbl> <date>   
## 1 14297519 0.0256 2017-01-01   
## 2 17052889 0.0556 2017-01-01   
## 3 14186244 0.00538 2017-01-01   
## 4 12177905 -0.103 2017-01-01   
## 5 10887214 0.00694 2017-01-01   
## 6 17143294 -0.0205 2017-01-01

### Comment: Contains parcel id, log error, and transaction date - needs to be joined with properties\_2017. After joining only the properties that sold (have logerror values) will be retained.

#since this is a dictionary, print as a table for easy reference  
kable(data\_dict%>%select(`Housing Feature` = Feature, Description),  
 caption = 'Table 1: List of Attributes with description')

Table 1: List of Attributes with description

| Housing Feature | Description |
| --- | --- |
| ‘airconditioningtypeid’ | Type of cooling system present in the home (if any) |
| ‘architecturalstyletypeid’ | Architectural style of the home (i.e. ranch, colonial, split-level, etc…) |
| ‘basementsqft’ | Finished living area below or partially below ground level |
| ‘bathroomcnt’ | Number of bathrooms in home including fractional bathrooms |
| ‘bedroomcnt’ | Number of bedrooms in home |
| ‘buildingqualitytypeid’ | Overall assessment of condition of the building from best (lowest) to worst (highest) |
| ‘buildingclasstypeid’ | The building framing type (steel frame, wood frame, concrete/brick) |
| ‘calculatedbathnbr’ | Number of bathrooms in home including fractional bathroom |
| ‘decktypeid’ | Type of deck (if any) present on parcel |
| ‘threequarterbathnbr’ | Number of 3/4 bathrooms in house (shower + sink + toilet) |
| ‘finishedfloor1squarefeet’ | Size of the finished living area on the first (entry) floor of the home |
| ‘calculatedfinishedsquarefeet’ | Calculated total finished living area of the home |
| ‘finishedsquarefeet6’ | Base unfinished and finished area |
| ‘finishedsquarefeet12’ | Finished living area |
| ‘finishedsquarefeet13’ | Perimeter living area |
| ‘finishedsquarefeet15’ | Total area |
| ‘finishedsquarefeet50’ | Size of the finished living area on the first (entry) floor of the home |
| ‘fips’ | Federal Information Processing Standard code - see <https://en.wikipedia.org/wiki/FIPS_county_code> for more details |
| ‘fireplacecnt’ | Number of fireplaces in a home (if any) |
| ‘fireplaceflag’ | Is a fireplace present in this home |
| ‘fullbathcnt’ | Number of full bathrooms (sink, shower + bathtub, and toilet) present in home |
| ‘garagecarcnt’ | Total number of garages on the lot including an attached garage |
| ‘garagetotalsqft’ | Total number of square feet of all garages on lot including an attached garage |
| ‘hashottuborspa’ | Does the home have a hot tub or spa |
| ‘heatingorsystemtypeid’ | Type of home heating system |
| ‘latitude’ | Latitude of the middle of the parcel multiplied by 10e6 |
| ‘longitude’ | Longitude of the middle of the parcel multiplied by 10e6 |
| ‘lotsizesquarefeet’ | Area of the lot in square feet |
| ‘numberofstories’ | Number of stories or levels the home has |
| ‘parcelid’ | Unique identifier for parcels (lots) |
| ‘poolcnt’ | Number of pools on the lot (if any) |
| ‘poolsizesum’ | Total square footage of all pools on property |
| ‘pooltypeid10’ | Spa or Hot Tub |
| ‘pooltypeid2’ | Pool with Spa/Hot Tub |
| ‘pooltypeid7’ | Pool without hot tub |
| ‘propertycountylandusecode’ | County land use code i.e. it’s zoning at the county level |
| ‘propertylandusetypeid’ | Type of land use the property is zoned for |
| ‘propertyzoningdesc’ | Description of the allowed land uses (zoning) for that property |
| ‘rawcensustractandblock’ | Census tract and block ID combined - also contains blockgroup assignment by extension |
| ‘censustractandblock’ | Census tract and block ID combined - also contains blockgroup assignment by extension |
| ‘regionidcounty’ | County in which the property is located |
| ‘regionidcity’ | City in which the property is located (if any) |
| ‘regionidzip’ | Zip code in which the property is located |
| ‘regionidneighborhood’ | Neighborhood in which the property is located |
| ‘roomcnt’ | Total number of rooms in the principal residence |
| ‘storytypeid’ | Type of floors in a multi-story house (i.e. basement and main level, split-level, attic, etc.). See tab for details. |
| ‘typeconstructiontypeid’ | What type of construction material was used to construct the home |
| ‘unitcnt’ | Number of units the structure is built into (i.e. 2 = duplex, 3 = triplex, etc…) |
| ‘yardbuildingsqft17’ | Patio in yard |
| ‘yardbuildingsqft26’ | Storage shed/building in yard |
| ‘yearbuilt’ | The Year the principal residence was built |
| ‘taxvaluedollarcnt’ | The total tax assessed value of the parcel |
| ‘structuretaxvaluedollarcnt’ | The assessed value of the built structure on the parcel |
| ‘landtaxvaluedollarcnt’ | The assessed value of the land area of the parcel |
| ‘taxamount’ | The total property tax assessed for that assessment year |
| ‘assessmentyear’ | The year of the property tax assessment |
| ‘taxdelinquencyflag’ | Property taxes for this parcel are past due as of 2015 |
| ‘taxdelinquencyyear’ | Year for which the unpaid propert taxes were due |

## Describe the data

# Print the size of each data set (rows then columns)  
dim(properties\_2016) # Contains 2985217 rows and 58 columns

## [1] 2985217 58

dim(properties\_2017) # Contains 2985217 rows and 58 columns

## [1] 2985217 58

dim(data\_dict) # Contains 58 rows and 2 columns

## [1] 58 2

dim(train\_2016) # Contains 90275 rows and 3 columns

## [1] 90275 3

dim(test\_2017) # Contains 77613 and 3 columns

## [1] 77613 3

# Confirm no duplicates in properties data sets  
uniqueproperties2016 <- unique(properties\_2016$parcelid)  
length(uniqueproperties2016) # All unique properties

## [1] 2985217

uniqueproperties2017 <- unique(properties\_2017$parcelid)  
length(uniqueproperties2017) # All unique properties

## [1] 2985217

uniquesales2016 <- unique(train\_2016$parcelid)  
length(uniquesales2016) # 90275 - 90150 = 125 properties sold more than once in 2016

## [1] 90150

uniquesales2017 <- unique(test\_2017$parcelid)  
length(uniquesales2017)# 77613 - 77414 = 199 properties sold more than once in 2017

## [1] 77414

# Count missing values in each data set  
sum(is.na(properties\_2016))

## [1] 85129239

sum(is.na(properties\_2017))

## [1] 84835659

sum(is.na(train\_2016))

## [1] 0

sum(is.na(test\_2017))

## [1] 0

## Join Properties to their respective Train/Test Data Sets to retain only sold home data

sold\_train <- left\_join(train\_2016, properties\_2016, by = "parcelid") # to join by parcelid & only retain sold homes  
sold\_test <- left\_join(test\_2017, properties\_2017, by = "parcelid") # to join by parcelid & only retain sold homes

### Comment: Join the properties & train/test (2016/2017, respectively) by parcel id to retain only the houses sold (with logerror values) before handling missing data.

### Confirm joined data sets only contain sold homes

dim(train\_2016)

## [1] 90275 3

dim(sold\_train)

## [1] 90275 60

dim(test\_2017)

## [1] 77613 3

dim(sold\_test)

## [1] 77613 60

head(sold\_train) #visually inspect joined training data set

## # A tibble: 6 x 60  
## parcelid logerror transactiondate airconditioningtypeid architecturalstyletyp~  
## <dbl> <dbl> <date> <dbl> <dbl>  
## 1 11016594 0.0276 2016-01-01 1 NA  
## 2 14366692 -0.168 2016-01-01 NA NA  
## 3 12098116 -0.004 2016-01-01 1 NA  
## 4 12643413 0.0218 2016-01-02 1 NA  
## 5 14432541 -0.005 2016-01-02 NA NA  
## 6 11509835 -0.270 2016-01-02 1 NA  
## # ... with 55 more variables: basementsqft <dbl>, bathroomcnt <dbl>,  
## # bedroomcnt <dbl>, buildingclasstypeid <dbl>, buildingqualitytypeid <dbl>,  
## # calculatedbathnbr <dbl>, decktypeid <dbl>, finishedfloor1squarefeet <dbl>,  
## # calculatedfinishedsquarefeet <dbl>, finishedsquarefeet12 <dbl>,  
## # finishedsquarefeet13 <dbl>, finishedsquarefeet15 <dbl>,  
## # finishedsquarefeet50 <dbl>, finishedsquarefeet6 <dbl>, fips <chr>,  
## # fireplacecnt <dbl>, fullbathcnt <dbl>, garagecarcnt <dbl>, ...

head(sold\_test) # visually inspect joined test data set

## # A tibble: 6 x 60  
## parcelid logerror transactiondate airconditioningtypeid architecturalstyletyp~  
## <dbl> <dbl> <date> <dbl> <dbl>  
## 1 14297519 0.0256 2017-01-01 NA NA  
## 2 17052889 0.0556 2017-01-01 NA NA  
## 3 14186244 0.00538 2017-01-01 NA NA  
## 4 12177905 -0.103 2017-01-01 NA NA  
## 5 10887214 0.00694 2017-01-01 1 NA  
## 6 17143294 -0.0205 2017-01-01 NA NA  
## # ... with 55 more variables: basementsqft <lgl>, bathroomcnt <dbl>,  
## # bedroomcnt <dbl>, buildingclasstypeid <dbl>, buildingqualitytypeid <dbl>,  
## # calculatedbathnbr <dbl>, decktypeid <dbl>, finishedfloor1squarefeet <dbl>,  
## # calculatedfinishedsquarefeet <dbl>, finishedsquarefeet12 <dbl>,  
## # finishedsquarefeet13 <dbl>, finishedsquarefeet15 <dbl>,  
## # finishedsquarefeet50 <dbl>, finishedsquarefeet6 <dbl>, fips <chr>,  
## # fireplacecnt <dbl>, fullbathcnt <dbl>, garagecarcnt <dbl>, ...

## Investigate and Handle Missing Values / Feature Redundancy

#Determine the extent of missing data in each data set  
sum(is.na(sold\_train)) # 2,537,678 missing values

## [1] 2537678

sum(is.na(sold\_test)) # 2,173,827 missing values

## [1] 2173827

### Comment: Extensive missing data exists - we will explore the extent of missing data per attribute.

summary(sold\_train) # displays basic descriptive statistics and # of NA's per feature

## parcelid logerror transactiondate   
## Min. : 10711738 Min. :-4.60500 Min. :2016-01-01   
## 1st Qu.: 11559500 1st Qu.:-0.02530 1st Qu.:2016-04-05   
## Median : 12547337 Median : 0.00600 Median :2016-06-14   
## Mean : 12984656 Mean : 0.01146 Mean :2016-06-11   
## 3rd Qu.: 14227552 3rd Qu.: 0.03920 3rd Qu.:2016-08-19   
## Max. :162960842 Max. : 4.73700 Max. :2016-12-30   
##   
## airconditioningtypeid architecturalstyletypeid basementsqft   
## Min. : 1.00 Min. : 2.00 Min. : 100.0   
## 1st Qu.: 1.00 1st Qu.: 7.00 1st Qu.: 407.5   
## Median : 1.00 Median : 7.00 Median : 616.0   
## Mean : 1.82 Mean : 7.23 Mean : 713.6   
## 3rd Qu.: 1.00 3rd Qu.: 7.00 3rd Qu.: 872.0   
## Max. :13.00 Max. :21.00 Max. :1555.0   
## NA's :61494 NA's :90014 NA's :90232   
## bathroomcnt bedroomcnt buildingclasstypeid buildingqualitytypeid  
## Min. : 0.000 Min. : 0.000 Min. :4 Min. : 1.00   
## 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.:4 1st Qu.: 4.00   
## Median : 2.000 Median : 3.000 Median :4 Median : 7.00   
## Mean : 2.279 Mean : 3.032 Mean :4 Mean : 5.57   
## 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.:4 3rd Qu.: 7.00   
## Max. :20.000 Max. :16.000 Max. :4 Max. :12.00   
## NA's :90259 NA's :32911   
## calculatedbathnbr decktypeid finishedfloor1squarefeet  
## Min. : 1.000 Min. :66 Min. : 44   
## 1st Qu.: 2.000 1st Qu.:66 1st Qu.: 938   
## Median : 2.000 Median :66 Median :1244   
## Mean : 2.309 Mean :66 Mean :1348   
## 3rd Qu.: 3.000 3rd Qu.:66 3rd Qu.:1614   
## Max. :20.000 Max. :66 Max. :7625   
## NA's :1182 NA's :89617 NA's :83419   
## calculatedfinishedsquarefeet finishedsquarefeet12 finishedsquarefeet13  
## Min. : 2 Min. : 2 Min. :1056   
## 1st Qu.: 1184 1st Qu.: 1172 1st Qu.:1392   
## Median : 1540 Median : 1518 Median :1440   
## Mean : 1773 Mean : 1745 Mean :1405   
## 3rd Qu.: 2095 3rd Qu.: 2056 3rd Qu.:1440   
## Max. :22741 Max. :20013 Max. :1584   
## NA's :661 NA's :4679 NA's :90242   
## finishedsquarefeet15 finishedsquarefeet50 finishedsquarefeet6  
## Min. : 560 Min. : 44 Min. : 257   
## 1st Qu.: 1648 1st Qu.: 938 1st Qu.:1112   
## Median : 2104 Median :1248 Median :2028   
## Mean : 2380 Mean :1356 Mean :2303   
## 3rd Qu.: 2862 3rd Qu.:1619 3rd Qu.:3431   
## Max. :22741 Max. :8352 Max. :7224   
## NA's :86711 NA's :83419 NA's :89854   
## fips fireplacecnt fullbathcnt garagecarcnt   
## Length:90275 Min. :1.00 Min. : 1.000 Min. : 0.00   
## Class :character 1st Qu.:1.00 1st Qu.: 2.000 1st Qu.: 2.00   
## Mode :character Median :1.00 Median : 2.000 Median : 2.00   
## Mean :1.19 Mean : 2.241 Mean : 1.81   
## 3rd Qu.:1.00 3rd Qu.: 3.000 3rd Qu.: 2.00   
## Max. :5.00 Max. :20.000 Max. :24.00   
## NA's :80668 NA's :1182 NA's :60338   
## garagetotalsqft hashottuborspa heatingorsystemtypeid latitude   
## Min. : 0.0 Mode:logical Min. : 1.00 Min. :33339295   
## 1st Qu.: 0.0 TRUE:2365 1st Qu.: 2.00 1st Qu.:33811538   
## Median : 433.0 NA's:87910 Median : 2.00 Median :34021500   
## Mean : 345.5 Mean : 3.93 Mean :34005411   
## 3rd Qu.: 484.0 3rd Qu.: 7.00 3rd Qu.:34172742   
## Max. :7339.0 Max. :24.00 Max. :34816009   
## NA's :60338 NA's :34195   
## longitude lotsizesquarefeet poolcnt poolsizesum   
## Min. :-119447865 Min. : 167 Min. :1 Min. : 28.0   
## 1st Qu.:-118411692 1st Qu.: 5703 1st Qu.:1 1st Qu.: 420.0   
## Median :-118173431 Median : 7200 Median :1 Median : 500.0   
## Mean :-118198868 Mean : 29110 Mean :1 Mean : 519.8   
## 3rd Qu.:-117921588 3rd Qu.: 11686 3rd Qu.:1 3rd Qu.: 600.0   
## Max. :-117554924 Max. :6971010 Max. :1 Max. :1750.0   
## NA's :10150 NA's :72374 NA's :89306   
## pooltypeid10 pooltypeid2 pooltypeid7 propertycountylandusecode  
## Min. :1 Min. :1 Min. :1 Length:90275   
## 1st Qu.:1 1st Qu.:1 1st Qu.:1 Class :character   
## Median :1 Median :1 Median :1 Mode :character   
## Mean :1 Mean :1 Mean :1   
## 3rd Qu.:1 3rd Qu.:1 3rd Qu.:1   
## Max. :1 Max. :1 Max. :1   
## NA's :89114 NA's :89071 NA's :73578   
## propertylandusetypeid propertyzoningdesc rawcensustractandblock  
## Min. : 31.0 Length:90275 Length:90275   
## 1st Qu.:261.0 Class :character Class :character   
## Median :261.0 Mode :character Mode :character   
## Mean :261.8   
## 3rd Qu.:266.0   
## Max. :275.0   
##   
## regionidcity regionidcounty regionidneighborhood regionidzip   
## Min. : 3491 Min. :1286 Min. : 6952 Min. : 95982   
## 1st Qu.: 12447 1st Qu.:1286 1st Qu.: 46736 1st Qu.: 96193   
## Median : 25218 Median :3101 Median :118887 Median : 96393   
## Mean : 33761 Mean :2525 Mean :190647 Mean : 96586   
## 3rd Qu.: 45457 3rd Qu.:3101 3rd Qu.:274800 3rd Qu.: 96987   
## Max. :396556 Max. :3101 Max. :764167 Max. :399675   
## NA's :1803 NA's :54263 NA's :35   
## roomcnt storytypeid threequarterbathnbr typeconstructiontypeid  
## Min. : 0.000 Min. :7 Min. :1.00 Min. : 4.00   
## 1st Qu.: 0.000 1st Qu.:7 1st Qu.:1.00 1st Qu.: 6.00   
## Median : 0.000 Median :7 Median :1.00 Median : 6.00   
## Mean : 1.479 Mean :7 Mean :1.01 Mean : 6.01   
## 3rd Qu.: 0.000 3rd Qu.:7 3rd Qu.:1.00 3rd Qu.: 6.00   
## Max. :18.000 Max. :7 Max. :4.00 Max. :13.00   
## NA's :90232 NA's :78266 NA's :89976   
## unitcnt yardbuildingsqft17 yardbuildingsqft26 yearbuilt   
## Min. : 1.00 Min. : 25.0 Min. : 18.0 Min. :1885   
## 1st Qu.: 1.00 1st Qu.: 180.0 1st Qu.: 100.0 1st Qu.:1953   
## Median : 1.00 Median : 259.5 Median : 159.0 Median :1970   
## Mean : 1.11 Mean : 310.1 Mean : 311.7 Mean :1969   
## 3rd Qu.: 1.00 3rd Qu.: 384.0 3rd Qu.: 361.0 3rd Qu.:1987   
## Max. :143.00 Max. :2678.0 Max. :1366.0 Max. :2015   
## NA's :31922 NA's :87629 NA's :90180 NA's :756   
## numberofstories fireplaceflag structuretaxvaluedollarcnt taxvaluedollarcnt   
## Min. :1.00 Mode:logical Min. : 100 Min. : 22   
## 1st Qu.:1.00 TRUE:222 1st Qu.: 81245 1st Qu.: 199023   
## Median :1.00 NA's:90053 Median : 132000 Median : 342872   
## Mean :1.44 Mean : 180093 Mean : 457673   
## 3rd Qu.:2.00 3rd Qu.: 210534 3rd Qu.: 540589   
## Max. :4.00 Max. :9948100 Max. :27750000   
## NA's :69705 NA's :380 NA's :1   
## assessmentyear landtaxvaluedollarcnt taxamount taxdelinquencyflag  
## Min. :2015 Min. : 22 Min. : 49.1 Length:90275   
## 1st Qu.:2015 1st Qu.: 82228 1st Qu.: 2872.8 Class :character   
## Median :2015 Median : 192970 Median : 4542.8 Mode :character   
## Mean :2015 Mean : 278335 Mean : 5984.0   
## 3rd Qu.:2015 3rd Qu.: 345420 3rd Qu.: 6901.1   
## Max. :2015 Max. :24500000 Max. :321936.1   
## NA's :1 NA's :6   
## taxdelinquencyyear censustractandblock  
## Min. : 6.0 Min. :6.037e+13   
## 1st Qu.:13.0 1st Qu.:6.037e+13   
## Median :14.0 Median :6.038e+13   
## Mean :13.4 Mean :6.049e+13   
## 3rd Qu.:15.0 3rd Qu.:6.059e+13   
## Max. :99.0 Max. :6.111e+13   
## NA's :88492 NA's :605

### Comment: All missing values are property-specific attributes rather than transactional attributes. We will initially remove all property-specific attributes with 5% or more missing data before complete case analysis.

na\_train <- data.frame(col = as.character(colnames(sold\_train)),   
 pct\_null = colSums(is.na(sold\_train))\*100/(colSums(is.na(sold\_train))+colSums(!is.na(sold\_train))))%>%  
 filter(col != 'parcelid')  
  
train <- sold\_train[,colnames(sold\_train) %in% na\_train$col[na\_train$pct\_null < 5]] #retain attributes w/ <5% missing  
test <- sold\_test[,colnames(sold\_test) %in% na\_train$col[na\_train$pct\_null < 5]] #retain same attributes in test  
removed <- sold\_train[,colnames(sold\_train) %in% na\_train$col[na\_train$pct\_null > 5]]

### Print Rationalized Features

colnames(removed) # 35 features removed

## [1] "airconditioningtypeid" "architecturalstyletypeid"  
## [3] "basementsqft" "buildingclasstypeid"   
## [5] "buildingqualitytypeid" "decktypeid"   
## [7] "finishedfloor1squarefeet" "finishedsquarefeet12"   
## [9] "finishedsquarefeet13" "finishedsquarefeet15"   
## [11] "finishedsquarefeet50" "finishedsquarefeet6"   
## [13] "fireplacecnt" "garagecarcnt"   
## [15] "garagetotalsqft" "hashottuborspa"   
## [17] "heatingorsystemtypeid" "lotsizesquarefeet"   
## [19] "poolcnt" "poolsizesum"   
## [21] "pooltypeid10" "pooltypeid2"   
## [23] "pooltypeid7" "propertyzoningdesc"   
## [25] "regionidneighborhood" "storytypeid"   
## [27] "threequarterbathnbr" "typeconstructiontypeid"   
## [29] "unitcnt" "yardbuildingsqft17"   
## [31] "yardbuildingsqft26" "numberofstories"   
## [33] "fireplaceflag" "taxdelinquencyflag"   
## [35] "taxdelinquencyyear"

### Print Retained Features for further rationalization

colnames(train) # 24 retained features

## [1] "logerror" "transactiondate"   
## [3] "bathroomcnt" "bedroomcnt"   
## [5] "calculatedbathnbr" "calculatedfinishedsquarefeet"  
## [7] "fips" "fullbathcnt"   
## [9] "latitude" "longitude"   
## [11] "propertycountylandusecode" "propertylandusetypeid"   
## [13] "rawcensustractandblock" "regionidcity"   
## [15] "regionidcounty" "regionidzip"   
## [17] "roomcnt" "yearbuilt"   
## [19] "structuretaxvaluedollarcnt" "taxvaluedollarcnt"   
## [21] "assessmentyear" "landtaxvaluedollarcnt"   
## [23] "taxamount" "censustractandblock"

## Address Redundcancies in Features

We notice redundancies in location data (fips, latitude & longitude, regionidcity, regionidcounty, rawcensustractandblock, censustractblock, and regionidzip) we will retain regionzip since it provides city/county information and lat/long are too property-specific.

We also notice redundancies in propertycountylandusecode, propertylandusetypeid, and landusetypeid. We will retain property landusetypeid.

We also we notice redundancies in bathroomcnt, calculatedbathnbr, and fullbathcnt. We will retain bathroomcnt.

We also notice redundancies in structuretaxvaluedollarcnt, taxvaluedollarcnt, and landtaxvaluedollarcnt. We will retain the cumulative taxvaluedollarcnt.

loc\_red <- c("fips", "latitude", "longitude", "regionidcity", "regionidcounty", "rawcensustractandblock", "censustractandblock", "propertycountylandusecode", "calculatedbathnbr", "fullbathcnt", "structuretaxvaluedollarcnt", "landtaxvaluedollarcnt", "assessmentyear" )  
train.df <- train[ , !(names(train) %in% loc\_red)]  
test.df <- test[ , !(names(test) %in% loc\_red)]

### Comment:

# Calculate the percentage change in missing data from removing features >5% missing & feature reduction  
sum(is.na(train.df))/sum(is.na(sold\_train))-1 \*100 # 6,613 missing values

## [1] -99.99943

sum(is.na(test.df))/sum(is.na(sold\_test))-1 \*100 # 4,343 missing values

## [1] -99.99962

### Comment: Missing data is SIGNIFICANTLY reduced by removing attributes/features with >5% missing values

# Complete Cases  
train\_fin <- train.df[complete.cases(train), ]  
test\_fin <- test.df[complete.cases(test), ]  
  
# Calculate % of observations retained  
nrow(train\_fin)/nrow(train) \*100

## [1] 96.39989

nrow(test\_fin)/nrow(test) \*100

## [1] 96.86522

### Comment: Complete Cases repesent 96.4% of our final training set and 96.9% of our final testing set. Since our data loss is minimal (~3-4%) we will proceed with complete case review rather than imputatino.

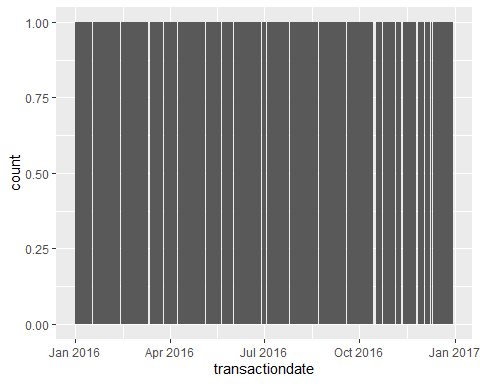
## Transform Time variable

train.clean <- train\_fin%>%  
 group\_by(transactiondate)%>%  
 summarise\_all(funs(mean)) # shows only the mean value per day rather than each value of the day  
  
test.clean <- test\_fin%>%  
 group\_by(transactiondate)%>%  
 summarise\_all(funs(mean)) # shows only the mean value per day rather than each value of the day

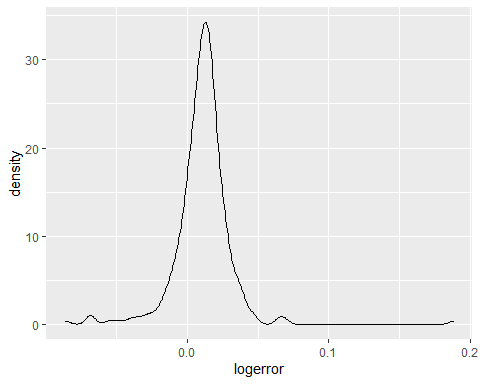
## Outlier Detection

my\_plots <- lapply(names(train.clean), function(var\_x){  
 p <-  
 ggplot(train.clean) +  
 aes\_string(var\_x)  
 if(is.numeric(train.clean[[var\_x]])) {  
 p <- p + geom\_density()  
 } else {  
 p <- p + geom\_bar()  
 }  
})  
my\_plots

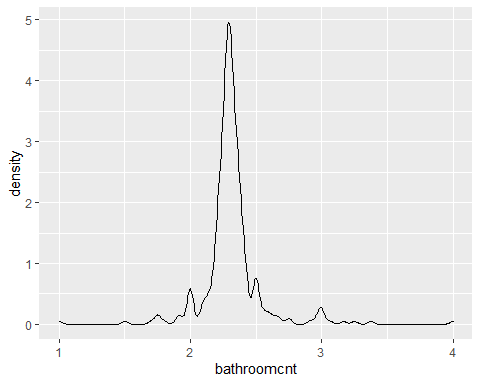
## [[1]]



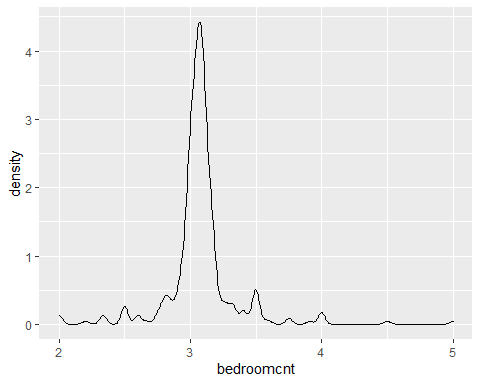
##   
## [[2]]



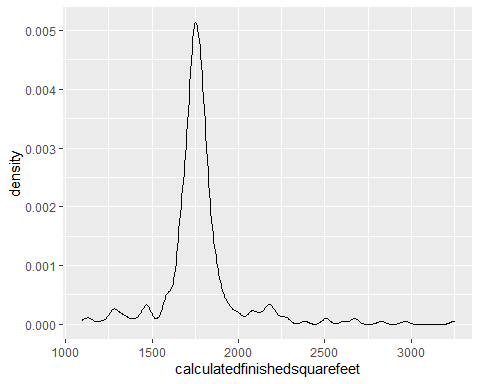
##   
## [[3]]



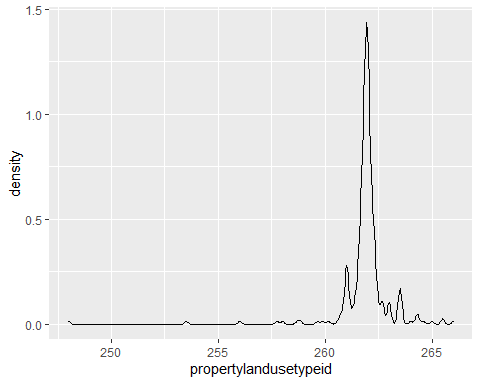
##   
## [[4]]



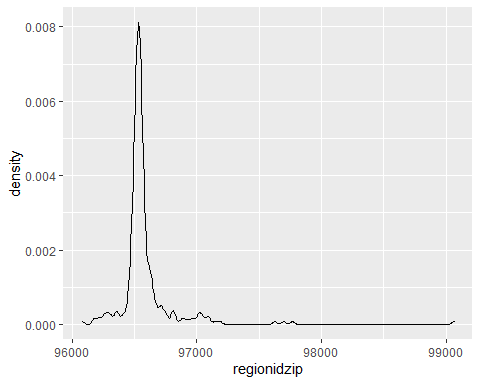
##   
## [[5]]



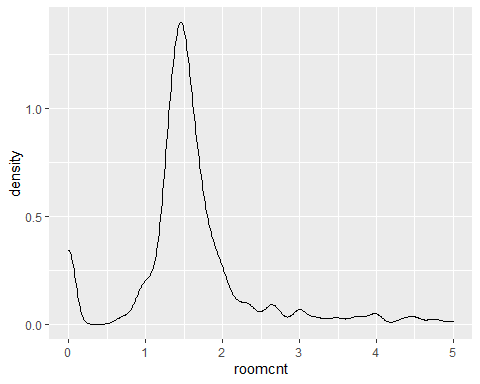
##   
## [[6]]



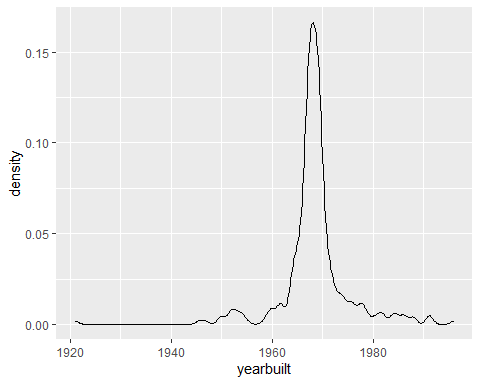
##   
## [[7]]



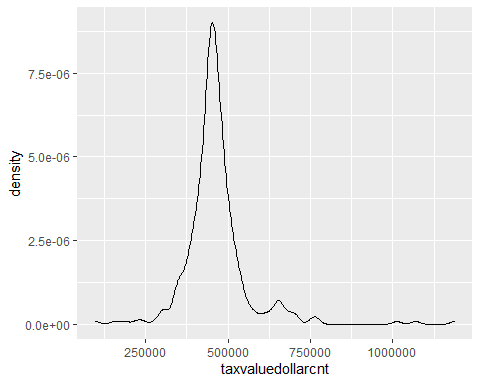
##   
## [[8]]



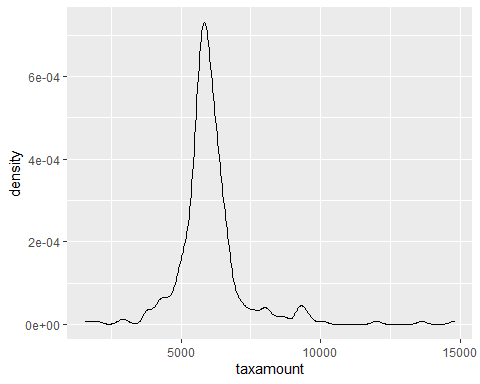
##   
## [[9]]



##   
## [[10]]

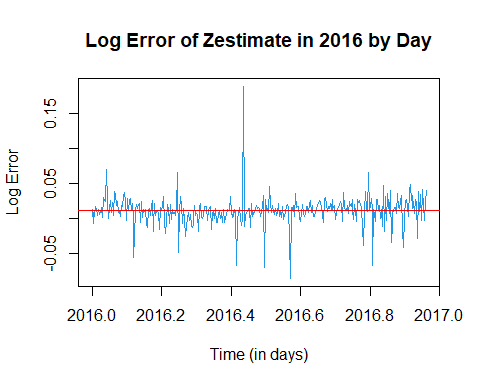


##   
## [[11]]



## Evaluating the Log Error over time

zillow.ts <- ts(train.clean, frequency = 365, start=c(2016,1,1))  
logerror <- zillow.ts[,'logerror']  
  
test.ts <- ts(test.clean, frequency = 365, start=c(2017,1,1))  
logerror\_test <- test.ts[,'logerror']  
  
ts.plot(logerror, ylab="Log Error",xlab = "Time (in days)", main="Log Error of Zestimate in 2016 by Day", col=4)  
logrerror\_mean <- mean(logerror)  
abline( h=logrerror\_mean, col="red")

 ### Comment: The Log Error resembles white noise with a mean near .01. Also the variance over time is not consistent (note postive and negative spikes centered around the mean) - indicating potential conditional heteroscedacity. We will confirm stationarity be evaluating its trend with linear regression.

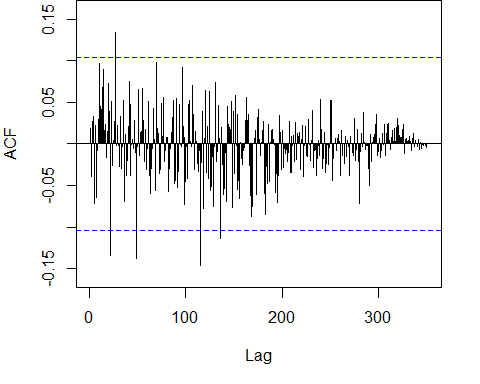
#Confirming no trend exists  
fit <- lm(logerror~time(zillow.ts), na.action=NULL)  
summary(fit)

##   
## Call:  
## lm(formula = logerror ~ time(zillow.ts), na.action = NULL)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.097156 -0.006970 0.001248 0.008506 0.177795   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -10.828483 7.767627 -1.394 0.164  
## time(zillow.ts) 0.005375 0.003852 1.395 0.164  
##   
## Residual standard error: 0.02012 on 350 degrees of freedom  
## Multiple R-squared: 0.005533, Adjusted R-squared: 0.002691   
## F-statistic: 1.947 on 1 and 350 DF, p-value: 0.1638

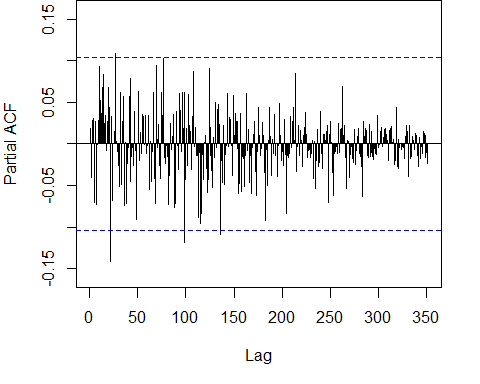
### Comment: with a statistically insignificant slope coefficient of 0.005 - log error has no trend. This combined with it’s consant mean indicates stationarity but with possible conditional heteroscedacity - which will be investigated addressed for the OLS and ARIMA models.

# Evaluating Log Error over time

par(mar=c(5,4,0,2)+.01)  
Acf(logerror, lag.max = 365)



par(mar=c(5,4,0,2)+.01)  
Pacf(logerror)

 ### Comment: The ACF and PACF tail off - suggesting an ARMA model - like AR(3).

# Model Development:

## Ordinary Least Squares with backward selection

library(olsrr)

##   
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':  
##   
## rivers

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:olsrr':  
##   
## cement

## The following object is masked from 'package:dplyr':  
##   
## select

# stepwise regression  
model <- lm(logerror ~ ., data = train.clean%>%dplyr::select(-transactiondate))  
ols\_step\_both\_p(model,pent=.03, progress= FALSE)

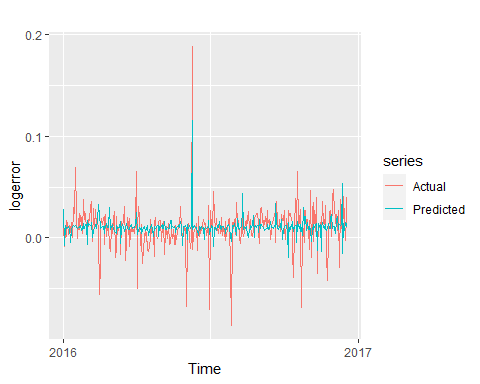
##   
## Stepwise Selection Summary   
## ----------------------------------------------------------------------------------------------------  
## Added/ Adj.   
## Step Variable Removed R-Square R-Square C(p) AIC RMSE   
## ----------------------------------------------------------------------------------------------------  
## 1 propertylandusetypeid addition 0.144 0.141 17.9570 -1799.6400 0.0187   
## 2 bathroomcnt addition 0.156 0.151 14.8300 -1802.6073 0.0186   
## 3 taxvaluedollarcnt addition 0.182 0.175 5.4390 -1811.8982 0.0183   
## ----------------------------------------------------------------------------------------------------

lm.fit <- tslm(logerror ~ propertylandusetypeid+bathroomcnt+taxvaluedollarcnt, data = zillow.ts)  
summary(lm.fit)

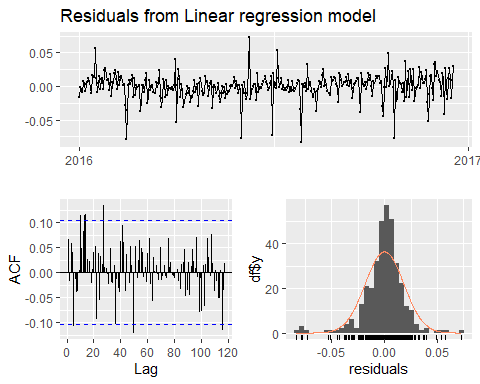
##   
## Call:  
## tslm(formula = logerror ~ propertylandusetypeid + bathroomcnt +   
## taxvaluedollarcnt, data = zillow.ts)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.083022 -0.007803 0.001703 0.009159 0.072464   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.415e+00 2.160e-01 6.551 2.06e-10 \*\*\*  
## propertylandusetypeid -5.459e-03 8.143e-04 -6.705 8.17e-11 \*\*\*  
## bathroomcnt 1.850e-02 5.053e-03 3.662 0.000289 \*\*\*  
## taxvaluedollarcnt -3.820e-08 1.134e-08 -3.368 0.000842 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01829 on 348 degrees of freedom  
## Multiple R-squared: 0.1825, Adjusted R-squared: 0.1754   
## F-statistic: 25.89 on 3 and 348 DF, p-value: 3.848e-15

method[1] = 'Ordinary Least Squares'  
accuracy[[1]] = data.frame(accuracy(lm.fit))

autoplot(logerror, series="Actual") +  
 forecast::autolayer(fitted(lm.fit), series="Predicted")



checkresiduals(lm.fit)



##   
## Breusch-Godfrey test for serial correlation of order up to 70  
##   
## data: Residuals from Linear regression model  
## LM test = 71.03, df = 70, p-value = 0.4432

### Comment: The OLS model explained 19.27% of the variance in the data. The residual plot does not indicate obvious heteroscedasticity since they are approximately normally distributed with a slightly longer left tail. Simple Exponential Smoothing or ARIMA will likely outperform the OLS model.

## Simple Exponential Smoothing

ses.fit <- ses(logerror)  
summary(ses.fit)

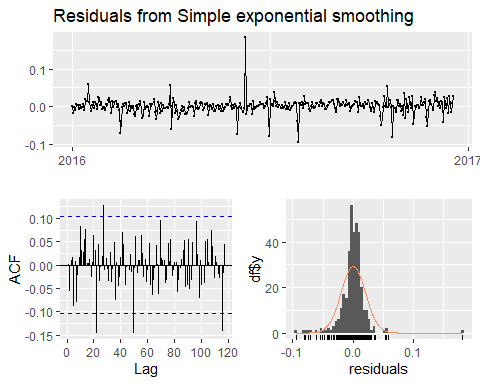
##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = logerror)   
##   
## Smoothing parameters:  
## alpha = 0.0171   
##   
## Initial states:  
## l = 0.0112   
##   
## sigma: 0.0202  
##   
## AIC AICc BIC   
## -679.7478 -679.6789 -668.1570   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.000435473 0.02012305 0.01211301 98.4266 216.6508 NaN  
## ACF1  
## Training set 0.002017516  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2016.9644 0.01384301 -0.01201930 0.03970531 -0.02570998 0.05339599  
## 2016.9671 0.01384301 -0.01202306 0.03970908 -0.02571573 0.05340175  
## 2016.9699 0.01384301 -0.01202683 0.03971284 -0.02572149 0.05340750  
## 2016.9726 0.01384301 -0.01203059 0.03971660 -0.02572724 0.05341325  
## 2016.9753 0.01384301 -0.01203435 0.03972036 -0.02573299 0.05341901  
## 2016.9781 0.01384301 -0.01203811 0.03972412 -0.02573875 0.05342476  
## 2016.9808 0.01384301 -0.01204187 0.03972788 -0.02574450 0.05343051  
## 2016.9836 0.01384301 -0.01204563 0.03973164 -0.02575025 0.05343626  
## 2016.9863 0.01384301 -0.01204939 0.03973540 -0.02575600 0.05344201  
## 2016.9890 0.01384301 -0.01205315 0.03973916 -0.02576175 0.05344776

method[2] = 'Simple Exponential Smoothing'  
accuracy[[2]] = data.frame(accuracy(ses.fit))

### Comment:

The optimal simple exponential smoothing function for the zillow.ts data uses and an initial value of .0112.

checkresiduals(ses.fit)



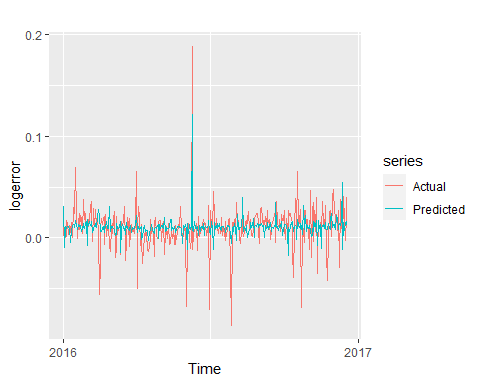
##   
## Ljung-Box test  
##   
## data: Residuals from Simple exponential smoothing  
## Q\* = 64.334, df = 68, p-value = 0.6036  
##   
## Model df: 2. Total lags used: 70

## ARIMA

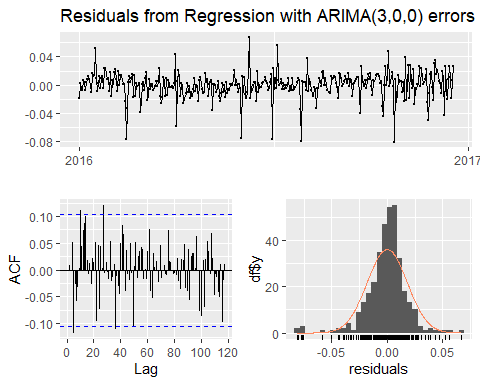
# Define xreg  
xreg <- as.matrix(zillow.ts[,c('propertylandusetypeid','bathroomcnt','taxvaluedollarcnt','roomcnt')])  
  
arima.fit <- auto.arima(logerror, xreg = xreg)  
  
arima.fit <- Arima(logerror, xreg=xreg, order=c(3,0,0))  
method[3] = 'ARIMA'  
accuracy[[3]] = data.frame(accuracy(arima.fit))

### Comment: The auto-arima indicated an ARIMA(3,0,0) model as the best fit.

autoplot(logerror, series="Actual") +  
 forecast::autolayer(fitted(arima.fit), series="Predicted")



checkresiduals(arima.fit)



##   
## Ljung-Box test  
##   
## data: Residuals from Regression with ARIMA(3,0,0) errors  
## Q\* = 66.356, df = 62, p-value = 0.3293  
##   
## Model df: 8. Total lags used: 70

## Summary Model Evaluation & Selection:

The fitted values of the ARIMA model follow a similar pattern to the linear model. Fitted values follow a similar, if more conservative, pattern as the observed values and, while there are some outlying residuals, the bulk of the values are clustered around zero.

mod.health <- bind\_rows(accuracy)%>%  
 bind\_cols(data.frame(method))%>%  
 dplyr::select(ME, RMSE, MAE, method)  
  
mod.health

## ME RMSE MAE  
## Training set...1 -2.605300e-19 0.01819038 0.01243723  
## Training set...2 4.354730e-04 0.02012305 0.01211301  
## Training set...3 5.066436e-06 0.01802417 0.01231528  
## method  
## Training set...1 Ordinary Least Squares  
## Training set...2 Simple Exponential Smoothing  
## Training set...3 ARIMA

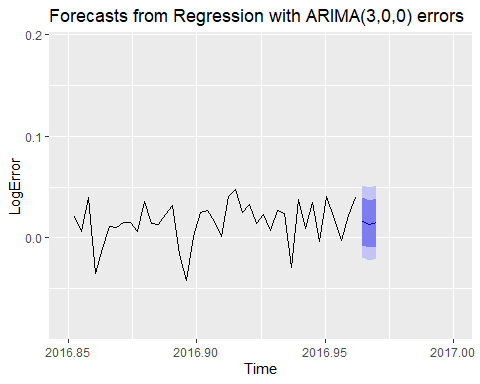
### Comment: ARIMA outperformed OLS with an MAE of 0.0123 vers 0.0124. It should be noted due to technical issues, SES could not be fit with our housing features.

# Forecasting October, November, December of 2017 using the ARIMA.Fit

test.zillow.ts <- ts(test.clean, frequency = 365, start=c(2017,1,1))  
test.full <- ts(bind\_rows(train.clean, test.clean)%>%  
 filter(transactiondate %in% c(as.Date('2016-10-01'),as.Date('2016-11-01'),as.Date('2016-12-01'),  
 as.Date('2017-10-01'),as.Date('2017-11-01'),as.Date('2017-12-01'))),   
 frequency = 365, start = c(2016,1,1))

forecast(arima.fit, xreg = test.full[,c('propertylandusetypeid','bathroomcnt','taxvaluedollarcnt', 'roomcnt')])%>%  
 autoplot()+  
 labs(y = 'LogError')+  
 xlim(2016.85,2017)

## Scale for 'x' is already present. Adding another scale for 'x', which will  
## replace the existing scale.

 # Conclusion The ARIMA and multiple regression models appeared to perform best and th ARIMA model was used to make predictions about out of sample data points.