Real Estate Data

Ibeh Amaka

2023-09-24

Introduction Real Estate Sales 2001-2020 The Office of Policy and Management maintains a listing of all real estate sales with a sales price of $2,000 or greater between October 1 and September 30 each year. For each sale record, the file includes town, property address, date of sale, property type (residential, apartment, commercial, industrial, or vacant land), sales price, and property assessment.

The dataset contains 997213 observations and 14 variables. I noticed the date.recorded columns were all recorded in the year 2020.

Analysis and Findings The library needed for this analysis is installed. This contained the harsh tage because it is installed on my system. You can take the harsh tag and run it in order to install it for your own use.

##load all the library needed  
#######First install the packages needed  
  
# install.packages("readr")  
# install.packages("tidyverse")  
# install.packages("tidyr")  
# install.packages("ggplot2")  
# install.packages("dylyer")  
# install.packages("data.table")  
#install.packages("knitr")  
#install.packages('forecast', dependencies = TRUE)

Then, the library is called

library(readr)  
library(tidyverse)  
library(tidyr)  
library(ggplot2)  
library(dplyr)  
library(data.table)  
library(knitr)  
library(forecast)  
library(tseries)

Next step is to load the data

sales <- read.csv("C:/Users/ibeha/OneDrive/Desktop/MY R DATASET/Business analystics/New folder/Real\_Estate\_Sales\_2001-2020\_GL.csv")

cleaning of the dataset. The last four columns are not useful for this analysis so I will drop it

sales = sales[-c(11:14)]

Next, I checked the structure of the dataset, this will enable me to see what data type it contained to enable me to run my analysis. This is important because if numbers (numeric datatype) are in character data type, then the analysis will not work.

str = sales

The Date variables were read in as characters, so i will convert them to date class

sales$Date.Recorded = as.Date(sales$Date.Recorded, format="%d/%m/%y")

I noticed I encountered NAs after converting to a date format because some dates were in mm-dd-yyy format. This returned na. I will check the percentage of missing values in the dataset.

missing\_values <- colMeans(is.na(sales)) \* 100  
missing\_values

The result shows that the data.recorded columns has a 61.2% missing values

Using the dataset's mean, I will replace it with 2020-05-29. I discovered this replacement is too much as almost half of the data (611570) will be replaced with the same means. I decided to leave them that way since I will mostly use the year column.

Proper Analysis: **I want to know the town with the highest sales, arranged in descending order.**

town\_sales <- sales %>%  
 group\_by(Town) %>%  
 summarise(sales = sum(Sales.Ratio)) %>%  
 arrange(desc(sales))  
sales\_10 = head (town\_sales,10)  
sales\_10

The results show that Salisbury, Newtown, New Fairfield, Westport and East Hartford has the highest sales.

Town | sales|

|:-------------|--------:|

|Newtown | 476270.4|

|Stamford | 256920.8|

|Guilford | 195562.3|

|Bethany | 195089.5|

|Hartford | 170403.1|

|Manchester | 169875.9|

|Mansfield | 163339.6|

|Torrington | 161971.1|

|East Hartford | 142804.5|

|East Lyme | 112560.3

ggplot(sales\_10, aes(x = reorder(Town, -sales), y = sales, fill = Town)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Sales by Town (Descending Order)", x = "Town", y = "Total Sales (in thousands)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_brewer(palette = "Set3") + # Color the bars  
 scale\_y\_continuous(labels = scales::comma\_format(scale = 1e-1))

A graph with colorful bars and text

Description automatically generated

**Property types that have more sales**

prop\_sales <- sales %>%  
 group\_by(Property.Type) %>%  
 summarise(sales = sum(Sales.Ratio)) %>%  
 arrange(desc(sales))  
prop\_sales

Property.Type | sales|

|:--------------|------------:|

|Single Family | 890281.85538|

|Condo | 36966.08138|

|Residential | 17740.11371|

|Two Family | 12486.91949|

|Three Family | 7087.72348|

|Commercial | 3701.32919|

|Vacant Land | 1259.08462|

|Four Family | 1048.90832|

|Industrial | 199.20715|

|Apartments | 118.48934|

|Public Utility | 0.02465|

The results shows that single family, condo and resisdentail has the highest sales I want to plot it however, there are empty rows so i will remove them

prop\_sales <- prop\_sales %>%  
 filter(!is.na(Property.Type) & Property.Type != "")  
   
ggplot(prop\_sales, aes(x = reorder(Property.Type, -sales), y = sales, fill = Property.Type)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Sales by property type (Descending Order)", x = "property type", y = "Total Sales (in thousands)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_brewer(palette = "Set3") +  
 scale\_y\_continuous(labels = scales::comma\_format(scale = 1))

A graph with text and numbers

Description automatically generated

From the data above, we can see that the difference between the sales of single family housing is large compared to the others

**Let's see the year with the highest sales**

year\_sales <- sales %>%  
 group\_by(List.Year) %>%  
 summarise(sales = sum(Sales.Ratio)) %>%  
 arrange(desc(sales))  
year\_sales  
  
year\_sales2 <- sales %>%  
 group\_by(List.Year) %>%  
 summarise(sales = sum(Sales.Ratio))   
   
year\_sales2  
year\_10 = head(year\_sales2, 10)

| List.Year| sales|

|---------:|----------:|

| 2005| 1197983.72|

| 2006| 754366.30|

| 2002| 194251.34|

| 2001| 134960.15|

| 2017| 53589.49|

| 2004| 49341.10|

| 2013| 39164.57|

| 2015| 39005.71|

| 2009| 38623.68|

| 2003| 36681.93|

| 2018| 35660.99|

| 2011| 31998.26|

| 2016| 31181.25|

| 2012| 30511.21|

| 2014| 30256.88|

| 2010| 30015.09|

| 2007| 26579.56|

| 2019| 24466.98|

| 2020| 23018.25|

| 2008| 18310.35|

From the result we see that year 2005, 2006, 2004, 2018 and 2003 has the highest sales.

ggplot(year\_10, aes(x = List.Year, y = sales)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 labs(title = "Total Sales by Year", x = "Year", y = "Total Sales") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) + # Rotate x-axis labels if needed  
 scale\_y\_continuous(labels = scales::comma\_format(scale = 1))

**Let's run some time series analysis, i want to predict the future years with good sales**

Using the Augmented Dickey-Fuller Test. Stationarity is an important concept in time series analysis because many time series models assume that the data is stationary, meaning that its statistical properties do not change over time. Here i want to visualize the properties of the time series

Now this will not work because my data.recorded as NAs, therefore i will omit the nas

clean\_data = na.omit(sales)  
  
time\_series <- ts(clean\_data$Date.Recorded, frequency = 12)  
  
autoplot(time\_series) + labs(title = "Your Time Series Data")  
  
decomposition <- decompose(time\_series)  
  
plot(decomposition)

lets run the Augmented Dickey-Fuller Test

adf.test(time\_series)

A graph with black lines

Description automatically generated

The Dickey-Fuller test statistic is strongly negative, indicating evidence against non-stationarity. The p-value (0.01) is less than the common significance level of 0.05, so I can reject the null hypothesis. Therefore, I have evidence to suggest that my time series data is stationary, which is a favorable condition for many time series models and analyses.

Augmented Dickey-Fuller Test

data: time\_series

Dickey-Fuller = -68.365, Lag order = 72, p-value = 0.01

alternative hypothesis: stationary

Let’s run the predictions, I will be using the year for this prediction because of the date. Recorded has only 2020, which will be biased.

# Create a time series using 'Year' as the time indicator  
ts\_data <- ts(sales$Sale.Amount, start = min(sales$List.Year), end = max(sales$List.Year), frequency = 1)  
# Fit a time series forecasting model (e.g., ARIMA)  
arima\_model <- auto.arima(ts\_data)  
  
# Number of future years to forecast  
n\_future\_years <- 5  
  
# Generate future years  
future\_years <- seq(max(sales$List.Year) + 1, length.out = n\_future\_years)  
  
# Generate forecasts for future years  
future\_forecasts <- forecast(arima\_model, h = n\_future\_years)  
  
# Extract the forecasts for future years  
future\_sales\_predictions <- future\_forecasts$mean  
future\_sales\_predictions

The results show that 367721.3 465727.8 465727.8 465727.8 465727.8, the year 2022- 2025, has the mean with the highest sales.

**Conclusion and recommendation**

The company should engage in Estate sales in the state with the highest sales. The company should focus more on marketing single-family, condo and residential housing because they have the highest sales. For the means of future sales, I noticed that the mean was the same for four years; this can be due to several reasons, including seasonality. However, we already confirmed that with our Dickey-Fuller test statistic. We can go ahead to perform more advanced tests using “prophet”; however, our data do not contain enough information, such as holidays or special events that may affect sales, marketing, inflation, etc.

I will recommend solid advertising and strong marketing teams in these areas to maximize the predicted future sales.