FinalDSC520

Bernard Owusu Sefah

2024-05-26

```
# Load required libraries
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(cluster)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
# Read in the datasets
housing_data <- read_csv("HousingData2.csv")
```

```
## Rows: 1029 Columns: 20

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (14): Region, Date, Median Sale Price, Median Sale Price MoM, Median Sal...
## dbl  (3): Days on Market, Days on Market MoM, Days on Market YoY
## num  (3): Homes Sold, New Listings, Inventory
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
income_data <- read_csv("kaggle_income2.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 32526 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): id, State_Code, State_Name, State_ab, County, City, Place, Type, P...
## dbl  (9): Area_Code, ALand, AWater, Lat, Lon, Mean, Median, Stdev, sum_w
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
unrate_data <- read_csv("UNRATE.csv")
```

```
## Rows: 916 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): UNRATE
## date (1): DATE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
demand_data <- read_csv("demand.csv")
```

```
## Rows: 81 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): DATE
## dbl (6): CSUSHPISA, MORTGAGE30US, UMCSENT, INTDSRUSM193N, MSPUS, GDP
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
city_data <- read_csv("City_time_series1.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 1048575 Columns: 81
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Date, RegionName
## dbl  (1): ZHVI_TopTier
## lgl (78): InventorySeasonallyAdjusted_AllHomes, InventoryRaw_AllHomes, Media...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
aspus_data <- read_csv("ASPUS.csv")
```

```
## Rows: 245 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): ASPUS
## date (1): DATE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Prepare the housing data
housing_data <- housing_data %>%
  rename(
    Median_Sale_Price = `Median Sale Price`,
    Homes_Sold = `Homes Sold`,
    Inventory = Inventory
  ) %>%
  mutate(Median_Sale_Price = as.numeric(gsub("[^0-9]", "", Median_Sale_Price)))

# Prepare the income data
income_data <- income_data %>%
  mutate(
    City = tolower(City),
    Median_Income = as.numeric(gsub(",", "", Median)),
    Mean_Income = as.numeric(gsub(",", "", Mean))
  ) %>%
  select(Median_Income, Mean_Income)

# Prepare the city time series data
city_data <- city_data %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%Y")) %>%
  separate(RegionName, into = c("City", "State"), sep = ",\\s*") %>%
  mutate(City = tolower(City)) %>%
  select(ZHVI_TopTier)
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 804 rows [1, 2, 3, 4, 5,
## 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```r
# Select required columns from each dataset
aspus_data <- aspus_data %>%
  select(ASPUS)

demand_data <- demand_data %>%
  select(CSUSHPISA, MORTGAGE30US, UMCSENT, INTDSRUSM193N, MSPUS, GDP)

unrate_data <- unrate_data %>%
  select(UNRATE)

# Function to pad datasets to the required length
pad_dataset <- function(dataset, target_length) {
  pad_size <- target_length - nrow(dataset)
  if (pad_size > 0) {
    pad_df <- as.data.frame(matrix(NA, nrow = pad_size, ncol = ncol(dataset)))
    colnames(pad_df) <- colnames(dataset)
    padded_dataset <- bind_rows(dataset, pad_df)
  } else {
    padded_dataset <- dataset
  }
  return(padded_dataset)
}

# Determine the target length (the largest dataset length)
target_length <- max(nrow(housing_data), nrow(income_data), nrow(unrate_data), nrow(demand_data), nrow(

# Pad each dataset to the target length
housing_data_padded <- pad_dataset(housing_data, target_length)
income_data_padded <- pad_dataset(income_data, target_length)
unrate_data_padded <- pad_dataset(unrate_data, target_length)
demand_data_padded <- pad_dataset(demand_data, target_length)
city_data_padded <- pad_dataset(city_data, target_length)
aspus_data_padded <- pad_dataset(aspus_data, target_length)

# Combine datasets by adding columns together
combined_data <- bind_cols(
  housing_data_padded %>% select(Date, Region, Median_Sale_Price, Homes_Sold, Inventory),
  income_data_padded,
  unrate_data_padded,
  demand_data_padded,
  city_data_padded,
  aspus_data_padded
)

# Exploratory Data Analysis (EDA)
summary(combined_data)
```

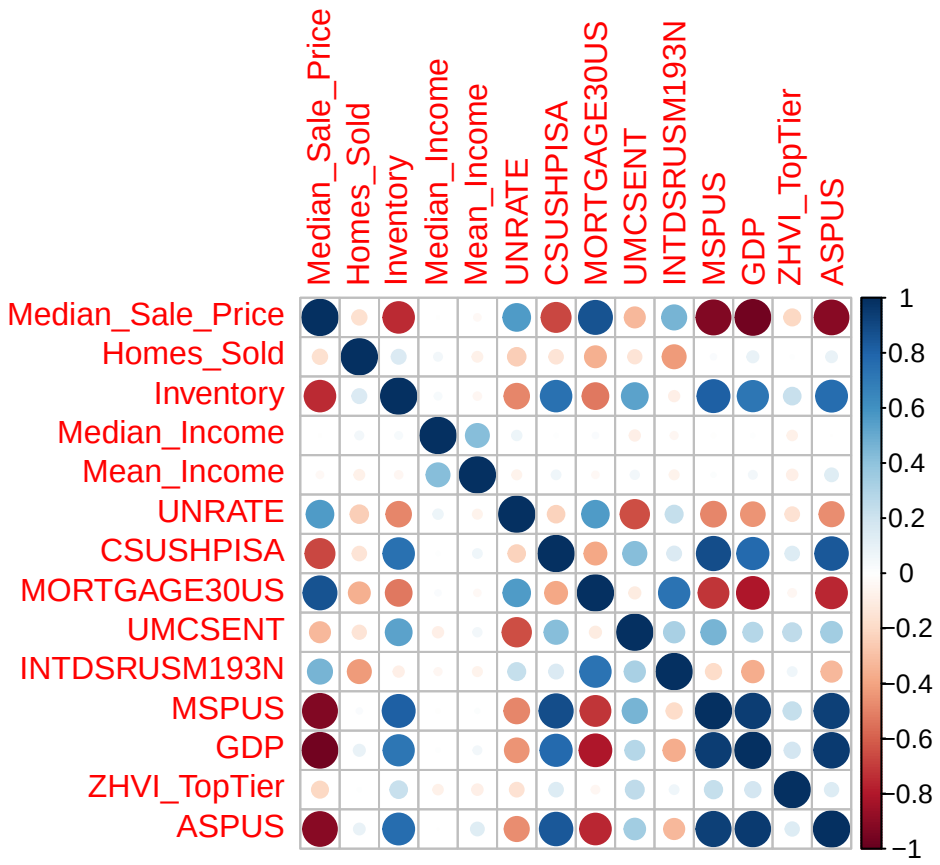
```
##      Date              Region          Median_Sale_Price   Homes_Sold
##  Length:1048575     Length:1048575     Min.   :134.0     Min.   :  972
##  Class :character   Class :character   1st Qu.:250.0     1st Qu.: 3322
##  Mode  :character   Mode  :character   Median :367.0     Median : 5186
```

```
##                                     Mean   :395.7     Mean   : 74198
##                                     3rd Qu.:510.0     3rd Qu.:  7389
##                                     Max.   :900.0     Max.   :726430
##                                     NA's   :1047546   NA's   :1047546
##    Inventory       Median_Income      Mean_Income          UNRATE
##  Min.   :   1052   Min.   :     0    Min.   :     0    Min.   : 2.5
##  1st Qu.:   7347   1st Qu.: 36506    1st Qu.: 46447    1st Qu.: 4.4
##  Median :  12592   Median : 52276    Median : 61265    Median : 5.5
##  Mean   : 224253   Mean   : 86464    Mean   : 67353    Mean   : 5.7
##  3rd Qu.:  22544   3rd Qu.: 81957    3rd Qu.: 82957    3rd Qu.: 6.7
##  Max.   :2160020   Max.   :300000    Max.   :242857    Max.   :14.8
##  NA's   :1047546   NA's   :1017426   NA's   :1017426   NA's   :1047659
##    CSUSHPISA        MORTGAGE30US        UMCSENT        INTDSRUSM193N
##  Min.   :129.3     Min.   :2.8       Min.   :56.1      Min.   :0.2
##  1st Qu.:148.2     1st Qu.:3.8       1st Qu.:73.9      1st Qu.:0.8
##  Median :172.3     Median :4.4       Median :83.0      Median :1.0
##  Mean   :180.7     Mean   :4.7       Mean   :82.1      Mean   :2.0
##  3rd Qu.:196.8     3rd Qu.:5.8       3rd Qu.:93.1      3rd Qu.:2.6
##  Max.   :303.4     Max.   :6.7       Max.   :98.9      Max.   :6.2
##  NA's   :1048495   NA's   :1048494   NA's   :1048494   NA's   :1048501
##      MSPUS              GDP           ZHVI_TopTier         ASPUS
##  Min.   :186000    Min.   :11174     Min.   :  31200   Min.   : 19200
##  1st Qu.:228100    1st Qu.:14449     1st Qu.: 106350   1st Qu.: 61600
##  Median :258400    Median :16629     Median : 142600   Median :151200
##  Mean   :281105    Mean   :17299     Mean   : 177386   Mean   :183869
##  3rd Qu.:318400    3rd Qu.:19895     3rd Qu.: 191850   3rd Qu.:286300
##  Max.   :479500    Max.   :26466     Max.   :4478100   Max.   :552600
##  NA's   :1048494   NA's   :1048494   NA's   :1047932   NA's   :1048330
```

```r
str(combined_data)
```

```
## tibble [1,048,575 x 16] (S3: tbl_df/tbl/data.frame)
##  $ Date             : chr [1:1048575] "29-May-22" "26-Jun-22" "24-Apr-22" "25-Jun-23" ...
##  $ Region           : chr [1:1048575] "National" "National" "National" "National" ...
##  $ Median_Sale_Price: num [1:1048575] 432 429 426 425 422 420 420 418 415 414 ...
##  $ Homes_Sold       : num [1:1048575] 593287 617893 554272 525266 450442 ...
##  $ Inventory        : num [1:1048575] 931679 1124101 809812 1003409 1022510 ...
##  $ Median_Income    : num [1:1048575] 30506 19528 31930 52814 67225 ...
##  $ Mean_Income      : num [1:1048575] 38773 37725 54606 63919 77948 ...
##  $ UNRATE           : num [1:1048575] 3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
##  $ CSUSHPISA        : num [1:1048575] 129 132 135 139 143 ...
##  $ MORTGAGE30US     : num [1:1048575] 5.84 5.51 6.03 5.92 5.6 ...
##  $ UMCSENT          : num [1:1048575] 80 89.3 89.3 92 98 ...
##  $ INTDSRUSM193N    : num [1:1048575] 2.25 2.17 2 2 2 ...
##  $ MSPUS            : num [1:1048575] 186000 191800 191900 198800 212700 ...
##  $ GDP              : num [1:1048575] 11174 11313 11567 11772 11923 ...
##  $ ZHVI_TopTier     : num [1:1048575] 108700 168400 147900 74500 131100 ...
##  $ ASPUS            : num [1:1048575] 19300 19400 19200 19600 19600 20200 20500 20900 21500 21000 ..
```

```r
# Plotting correlations
cor_matrix <- cor(combined_data %>% select(-Date, -Region), use = "complete.obs")
corrplot(cor_matrix, method = "circle")
```

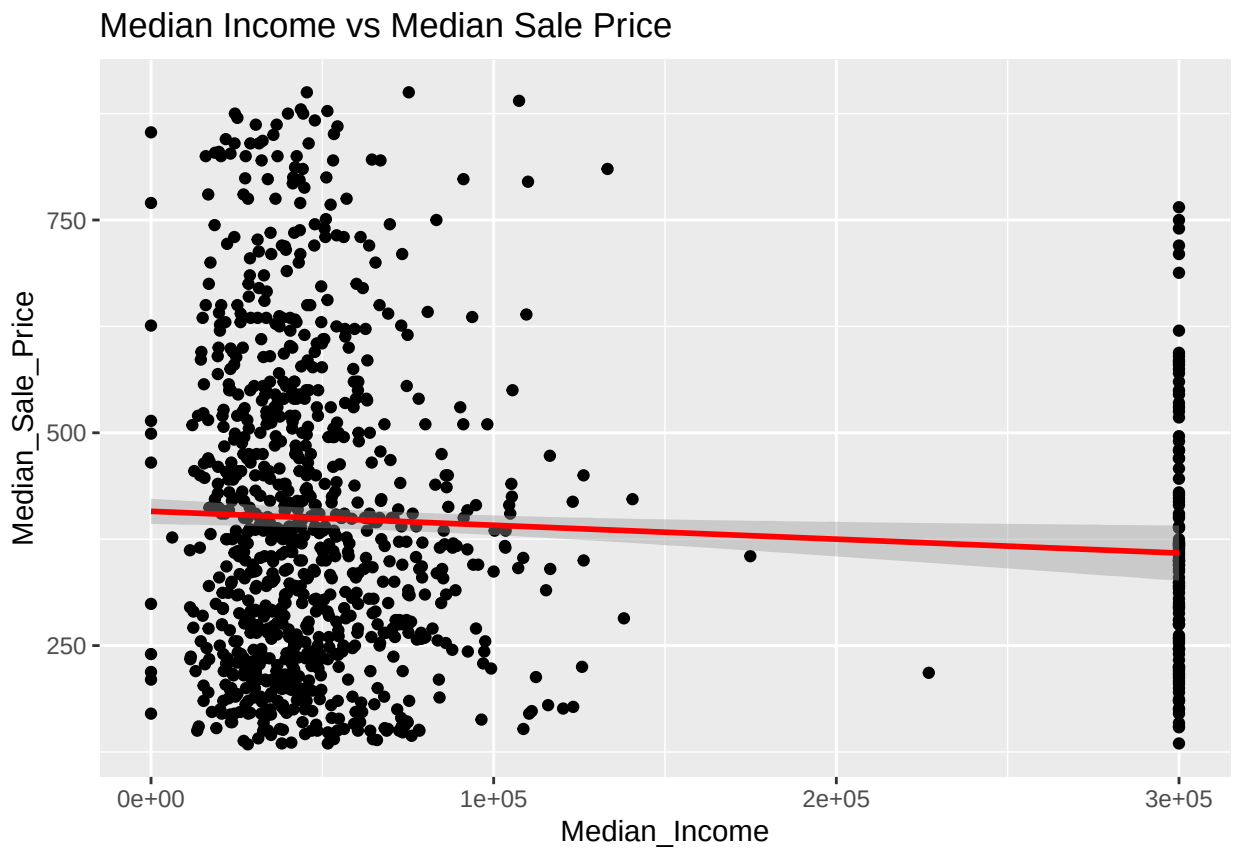



```
# Scatter plots to visualize relationships
ggplot(combined_data, aes(x = Median_Income, y = Median_Sale_Price)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  ggtitle("Median Income vs Median Sale Price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 1047546 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

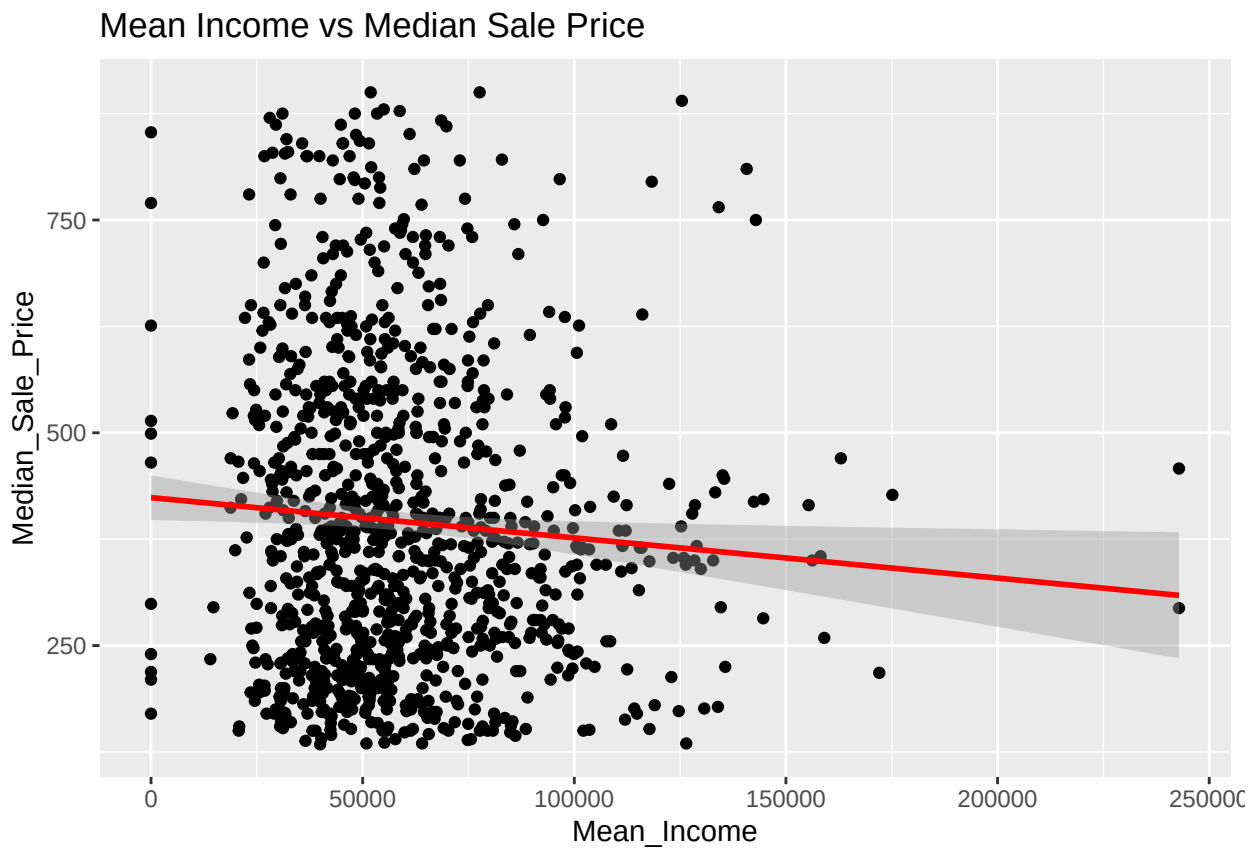
```
## Warning: Removed 1047546 rows containing missing values or values outside the scale
## range (`geom_point()`).
```




```
ggplot(combined_data, aes(x = Mean_Income, y = Median_Sale_Price)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  ggtitle("Mean Income vs Median Sale Price")
```

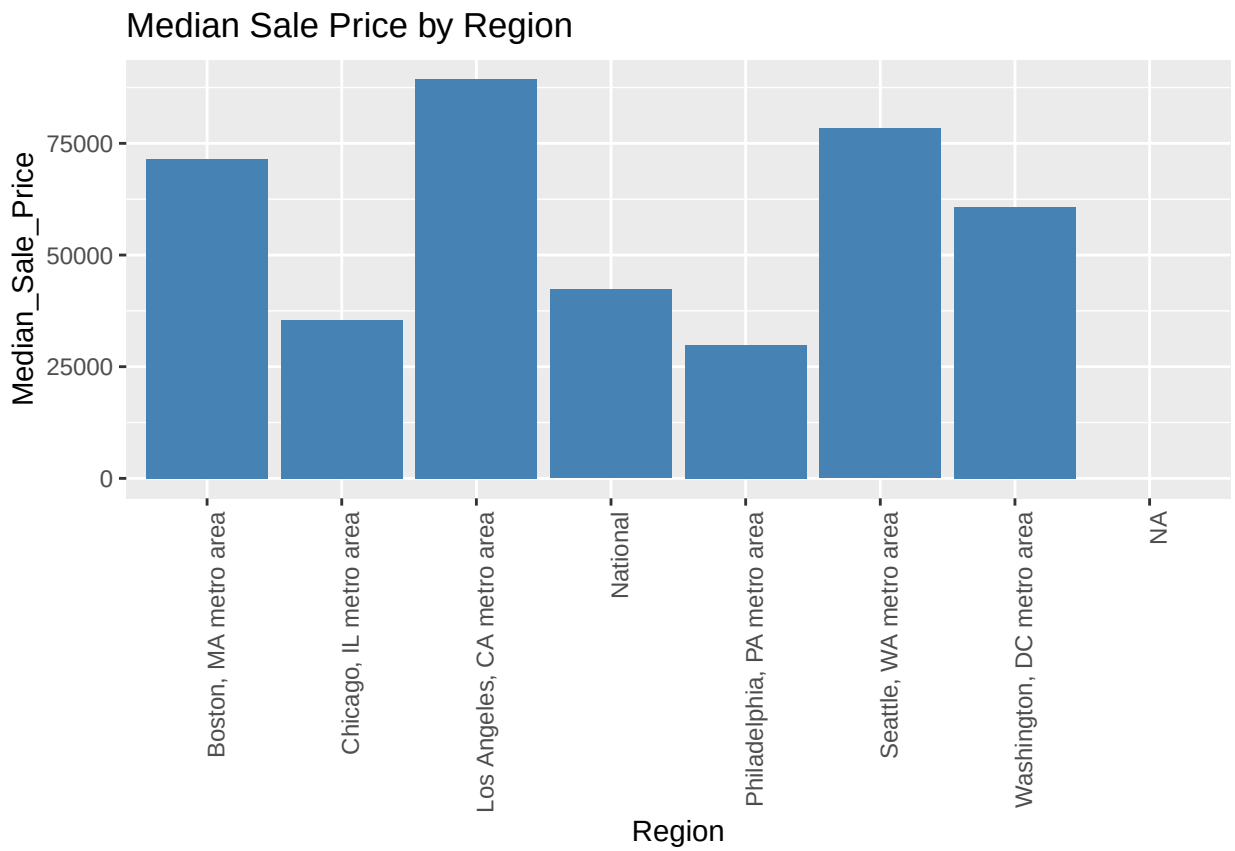
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 1047546 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 1047546 rows containing missing values or values outside the scale
## range (`geom_point()`).
```



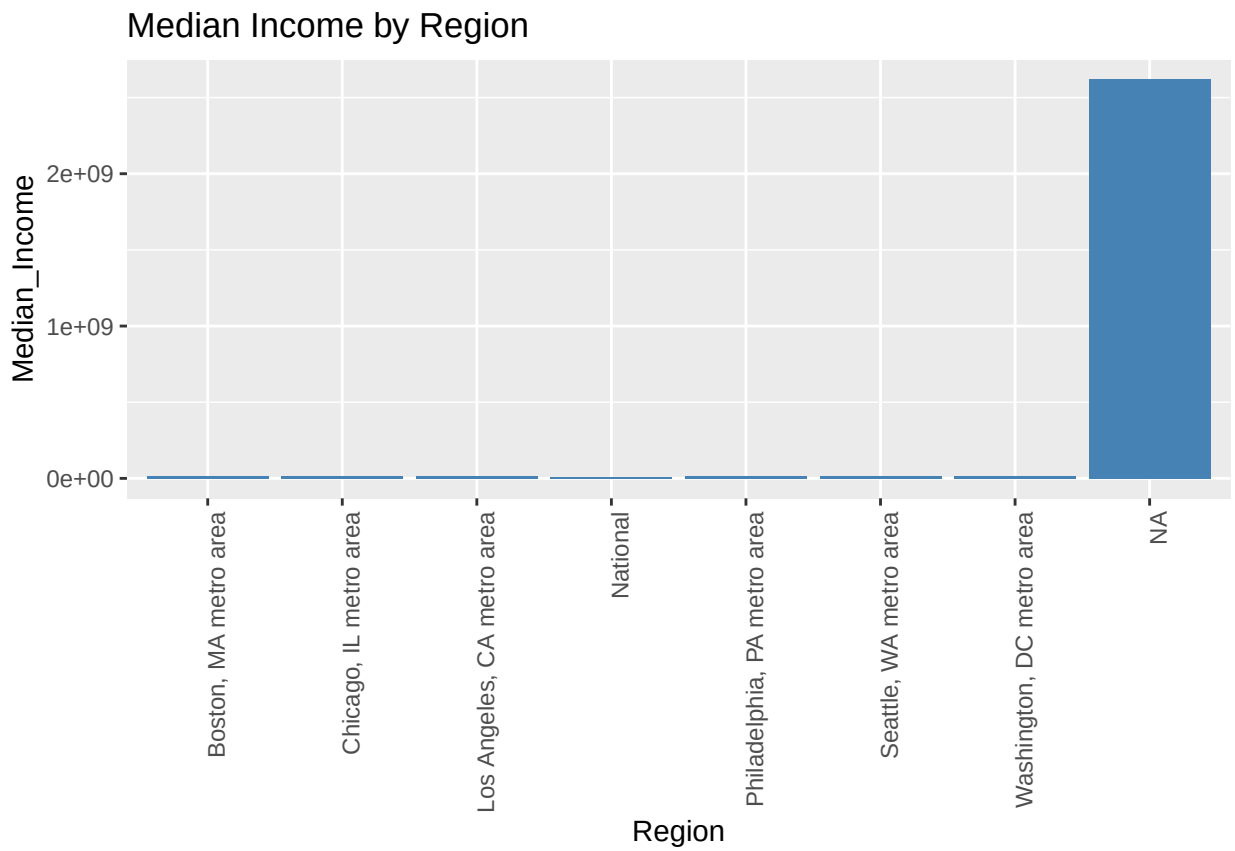

```
# Bar plots to visualize regional differences
ggplot(combined_data, aes(x = Region, y = Median_Sale_Price)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Median Sale Price by Region")
```

```
## Warning: Removed 1047546 rows containing missing values or values outside the scale
## range (`geom_bar()`).
```



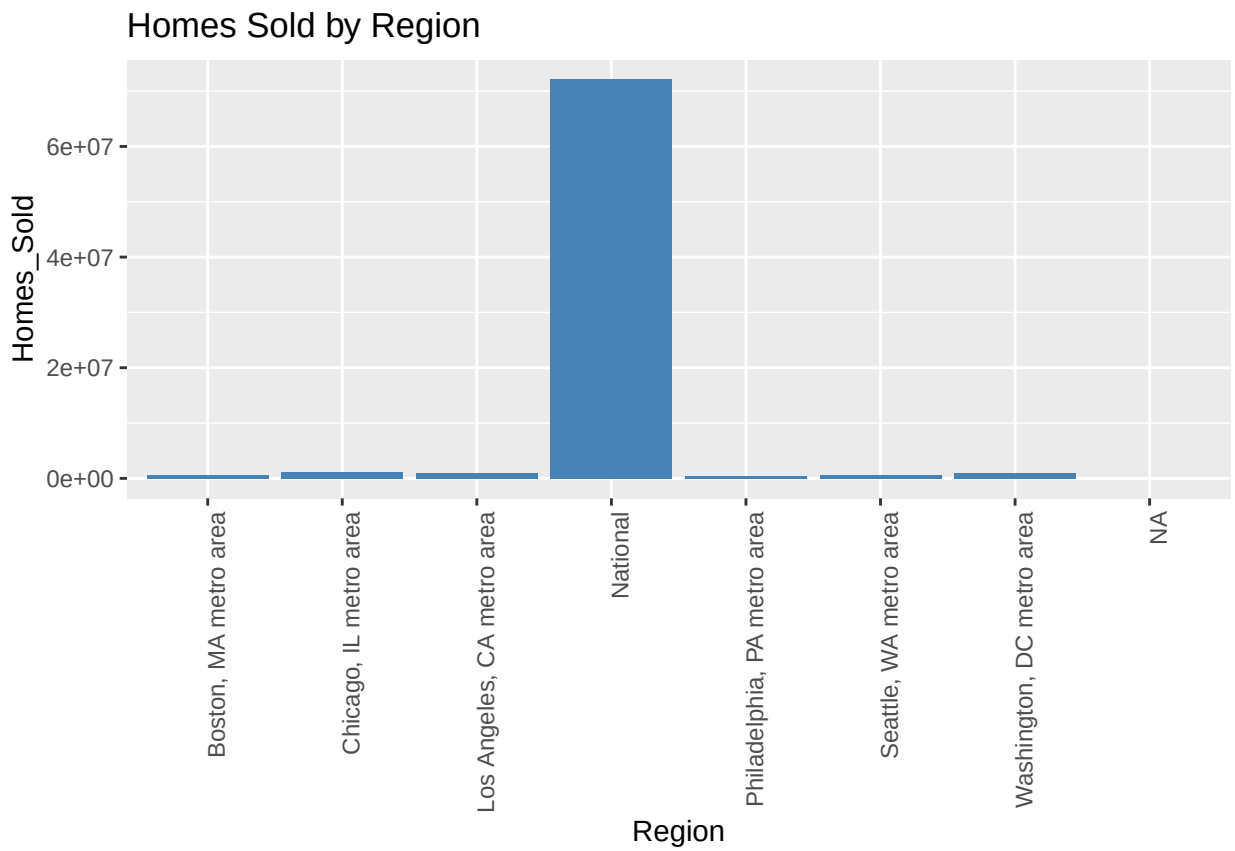

```
ggplot(combined_data, aes(x = Region, y = Median_Income)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Median Income by Region")
```

```
## Warning: Removed 1017426 rows containing missing values or values outside the scale
## range (`geom_bar()`).
```



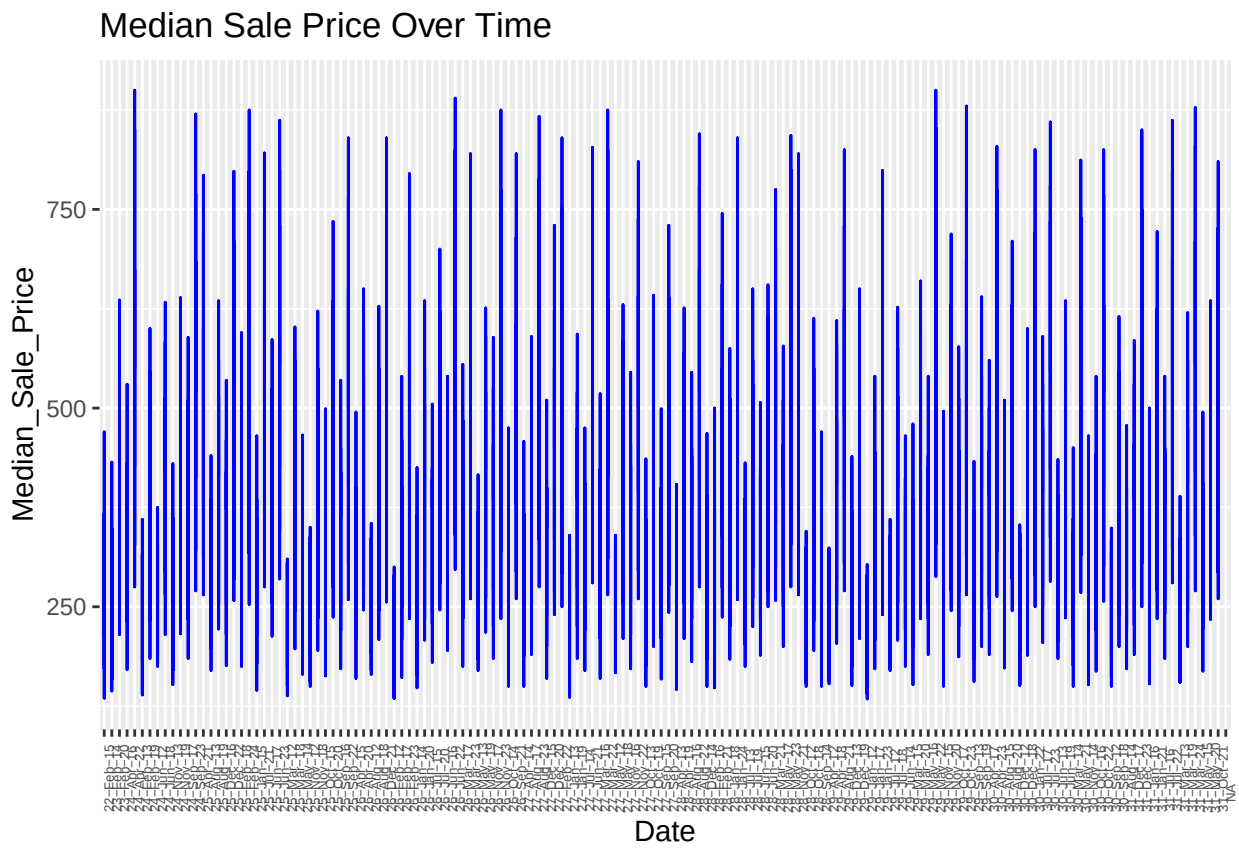

```
ggplot(combined_data, aes(x = Region, y = Homes_Sold)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Homes Sold by Region")
```

```
## Warning: Removed 1047546 rows containing missing values or values outside the scale
## range (`geom_bar()`).
```



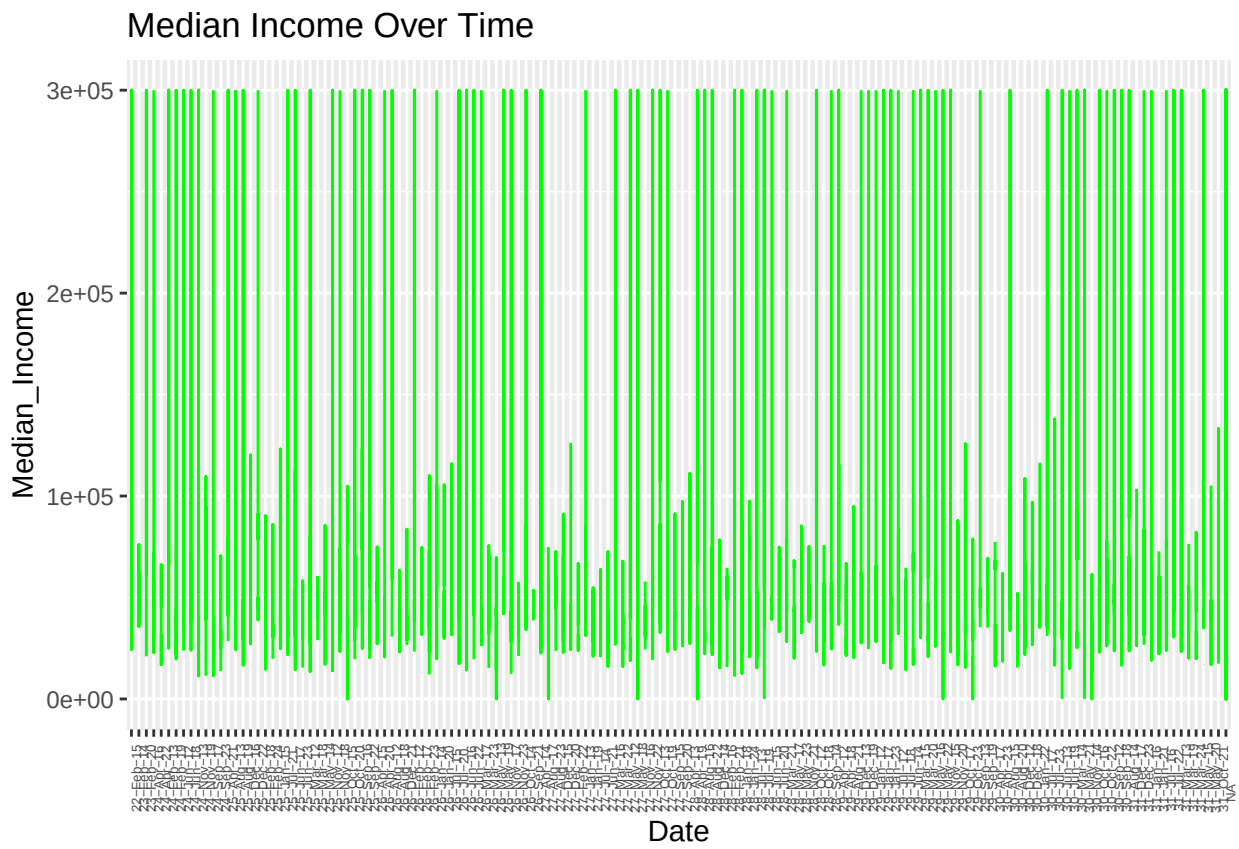

```
# Time series plots to visualize trends over time
ggplot(combined_data, aes(x = Date, y = Median_Sale_Price)) +
  geom_line(color = "blue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 0.2,size =5 )) +
  ggtitle("Median Sale Price Over Time")
```

```
## Warning: Removed 1047546 rows containing missing values or values outside the scale
## range (`geom_line()`).
```



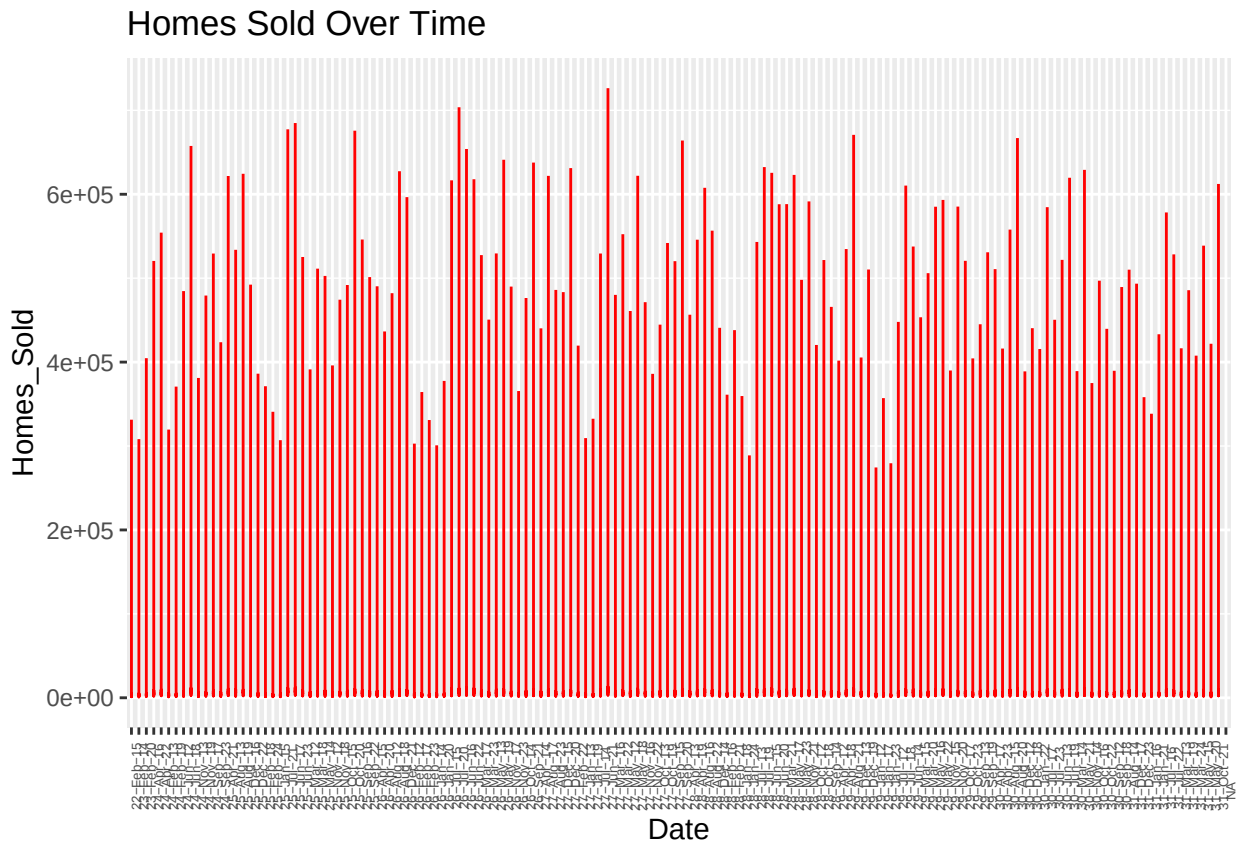

```
ggplot(combined_data, aes(x = Date, y = Median_Income)) +
  geom_line(color = "green") +
  theme(axis.text.x = element_text(angle = 90, hjust = 0.2,size =5 )) +
  ggtitle("Median Income Over Time")
```

```
## Warning: Removed 1017426 rows containing missing values or values outside the scale
## range (`geom_line()`).
```




```
ggplot(combined_data, aes(x = Date, y = Homes_Sold)) +
  geom_line(color = "red") +
  theme(axis.text.x = element_text(angle = 90, hjust = 0.2,size =5 )) +
  ggtitle("Homes Sold Over Time")
```

```
## Warning: Removed 1047546 rows containing missing values or values outside the scale
## range (`geom_line()`).
```




```
# Modeling
# Prepare the data for modeling
model_data <- combined_data %>%
  select(Median_Sale_Price, Median_Income, Mean_Income, UNRATE, CSUSHPISA, MORTGAGE30US, UMCSENT, INTDSI
  drop_na()

# Split data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(model_data$Median_Sale_Price, p = 0.8, list = FALSE)
train_data <- model_data[trainIndex, ]
test_data <- model_data[-trainIndex, ]

# Linear regression model
lm_model <- lm(Median_Sale_Price ~ ., data = train_data)
summary(lm_model)
```

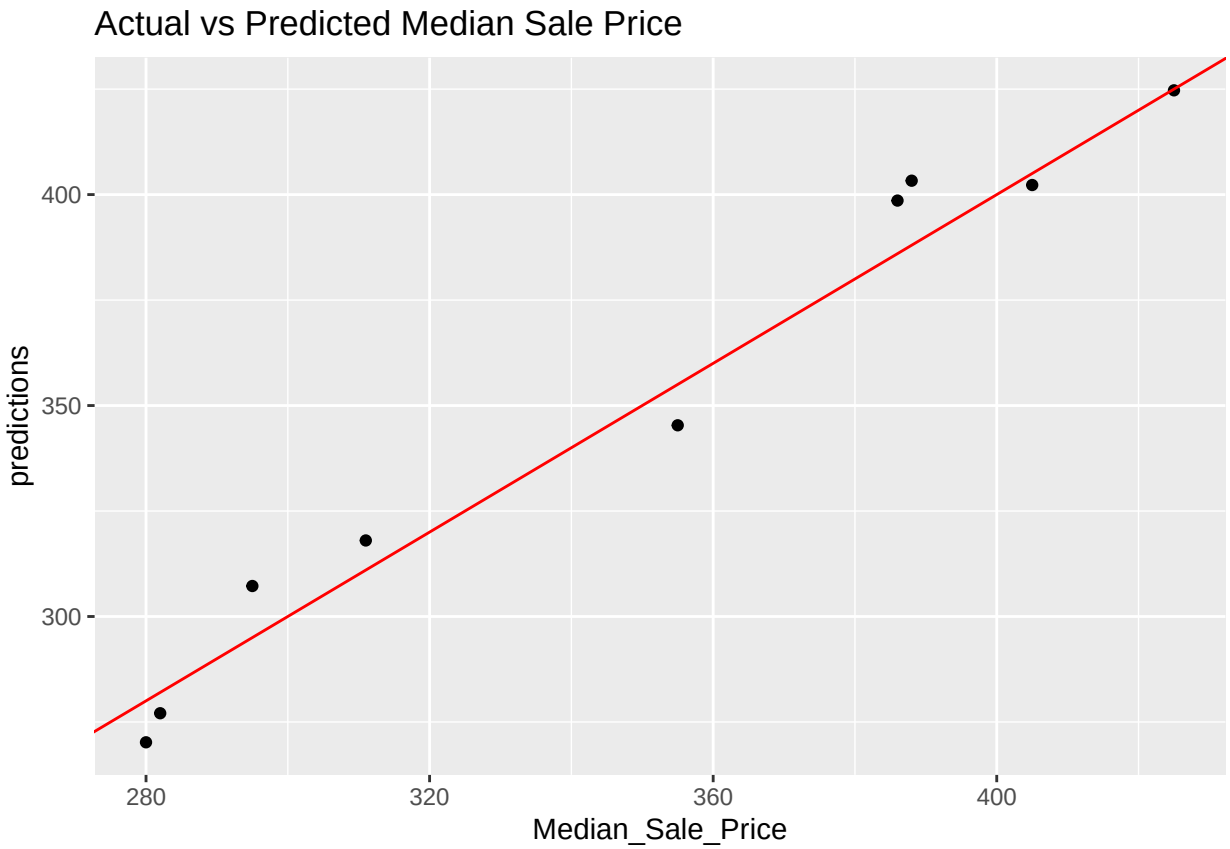
```
##
## Call:
## lm(formula = Median_Sale_Price ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4454  -4.2797   0.1592   4.0031  15.5483
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    5.686e+02  4.219e+01  13.478 2.03e-15 ***
## Median_Income -5.407e-06  2.558e-05  -0.211 0.833798
## Mean_Income   -9.606e-05  9.551e-05  -1.006 0.321396
## UNRATE         3.573e+00  2.208e+00   1.618 0.114570
## CSUSHPISA      7.546e-01  3.455e-01   2.184 0.035744 *
## MORTGAGE30US  -9.983e-02  4.178e+00  -0.024 0.981073
## UMCSENT        1.283e-01  2.373e-01   0.541 0.592107
## INTDSRUSM193N  3.322e-03  2.694e+00   0.001 0.999023
## MSPUS         -6.814e-04  2.292e-04  -2.973 0.005309 **
## GDP           -1.100e-02  2.749e-03  -4.001 0.000311 ***
## ZHVI_TopTier   4.754e-06  9.666e-06   0.492 0.625915
## ASPUS         -1.432e-04  4.941e-04  -0.290 0.773714
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.443 on 35 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9761
## F-statistic: 171.9 on 11 and 35 DF,  p-value: < 2.2e-16
```

```r
# Predict on the test set
predictions <- predict(lm_model, test_data)
# Calculate RMSE
rmse <- sqrt(mean((predictions - test_data$Median_Sale_Price)^2))
print(paste("RMSE:", rmse))
```

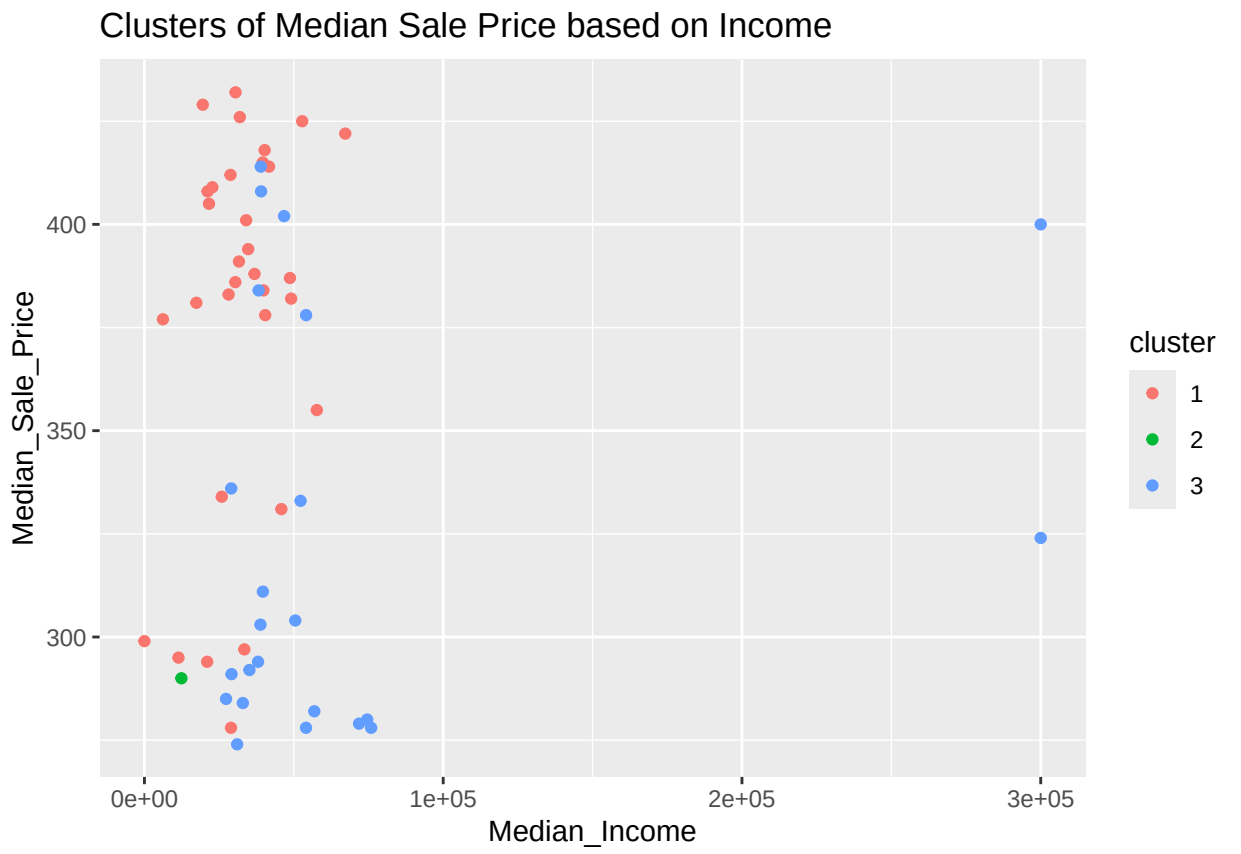
```
## [1] "RMSE: 9.50208637404978"
```

```r
# Plotting actual vs predicted values
ggplot(data = test_data, aes(x = Median_Sale_Price, y = predictions)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, col = "red") +
  ggtitle("Actual vs Predicted Median Sale Price")
```




```r
# Clustering analysis
set.seed(123)
kmeans_model <- kmeans(model_data %>% select(-Median_Sale_Price), centers = 3)
model_data$cluster <- as.factor(kmeans_model$cluster)

# Visualize clusters
ggplot(model_data, aes(x = Median_Income, y = Median_Sale_Price, color = cluster)) +
  geom_point() +
  ggtitle("Clusters of Median Sale Price based on Income")
```



##

Analysis on Linear Regression Model and RMSE

The linear regression model aims to predict the median sale price of homes using various predictor variables.

Residuals:

Min: -20.4454 1Q (First Quartile): -4.2797 Median: 0.1592 3Q (Third Quartile): 4.0031 Max: 15.5483 These values summarize the distribution of the residuals (differences between the observed and predicted values). Ideally, the residuals should be normally distributed around zero.

Coefficients: Each row corresponds to a predictor variable in the model, showing its estimated effect on the median sale price. Here are the key columns:

Estimate: The estimated effect of the predictor variable on the median sale price. Std. Error: The standard error of the estimate, measuring its precision. t value: The t-statistic for the hypothesis test that the coefficient is different from zero. Pr(>|t|): The p-value for the hypothesis test. A small p-value (typically $<$ 0.05) indicates that the predictor variable is statistically significant. Key findings:

Intercept: The baseline median sale price when all predictors are zero is approximately 568.6. Significant Predictors: CSUSHPISA (Consumer Price Index for Urban Consumers), MSPUS (Monthly Supply of Houses in the U.S.), and GDP (Gross Domestic Product) are statistically significant predictors of median sale price. Their p-values are less than 0.05, indicating a significant relationship with the median sale price.

Root Mean Squared Error (RMSE)

RMSE Calculation: The RMSE is calculated using the predictions from the linear regression model on the test dataset. It measures the average magnitude of the prediction errors, giving an idea of how well the model performs on new data.

Predictions: The predicted median sale prices using the test data. Observed Values: The actual median sale prices in the test data.

The RMSE value is 9.50208637404978. This means that, on average, the predicted median sale prices differ from the actual values by approximately 9.5 units. The lower the RMSE, the better the model's performance.

The linear regression model explains a significant portion of the variance in housing prices, with an R-squared value of 0.9818. Key predictors like CSUSHPISA, MSPUS, and GDP significantly influence housing prices. The RMSE of 9.5 indicates that the model's predictions are reasonably close to the actual values, demonstrating its effectiveness in predicting median sale prices. However, there is room for improvement, possibly by incorporating additional data sources or using more complex modeling techniques.