Analysis Report

This report is structured as follows.

Contents

Methodology

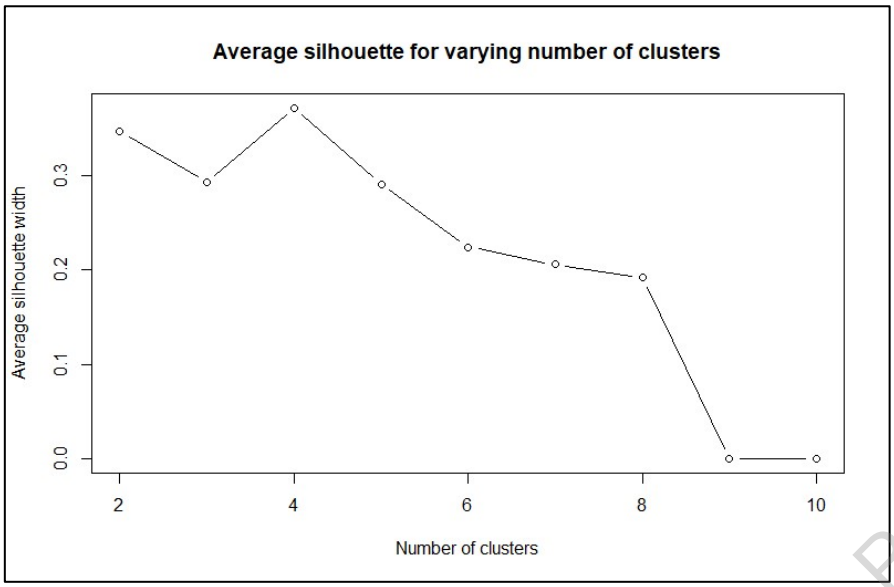
The goal of this analysis was to generate market segments from a dataset of households. Cluster analysis in marketing serves as a technique to segment a heterogeneous customer base into distinct categories, or clusters, based on similar behaviors, preferences, or characteristics. This segmentation allows marketing teams to tailor their strategies to the specific needs and behaviors of each cluster, thereby improving targeting, personalization, and overall campaign effectiveness. The primary aim is to maximize intra-cluster similarity while minimizing inter-cluster similarity, ensuring that members within each cluster are as alike as possible but distinct from members in other clusters. By doing so, companies can identify niche markets, optimize product offerings, and develop targeted marketing campaigns, enhancing customer engagement and ultimately increasing profitability.

In this data analysis, the following variables were selected to generate the clusters: 'Bedrooms', 'Bathrooms', 'Price.Per.sq.ft', 'Has.HOA', 'Property.Age', 'Total.Structure.Area.sqft', 'Has.Garage', 'Months.Since.Purchase', 'Returning.Customer', and 'State'. 'Returning.Customer' was a calculated binary variable (Yes/No) to identify customers with more than one order in the dateset. ‘Months.Since.Purchase’ was calculated subtracting the house selling date with the job date. Other variables were not included due to their correlational nature with variables that are already in the procedure. ‘Aggregate.Sub.Total’ and ‘Job.Dates’ were used to help profiling the clusters after the algorithm was executed.

The data was first cleaned to deal with issues like date conversion, duplication, and missing values. The data contains a substantial amount of missing data, which is harmful for cluster analysis. For instance, Total Structure Area had 4,007 missing cases. Other variables had from 2,441 cases to 8,519 cases (Months.Since.Purchase). The original dataframe (29,514) was reduced to 23,741 after duplicate customers were removed. It was further reduced to 21,254 considering only high-confidence data points and then to 12,417 following a filter for complete cases (no missing data) to maintain data integrity.

For the clustering algorithm, Partitioning Around Medoids (PAM) with Gower distance was chosen. This approach effectively handles a mix of continuous and categorical variables, as seen in the dataset.

To identify the optimal number of clusters, silhouette width was employed. The silhouette method gives insight into the distance between the resulting clusters. More distant clusters lead to better clusterings. After evaluating silhouette widths for various cluster numbers (k), k=4 was selected as the optimal number. The graph of silhouette widths is shown below. The highest width was observed when k = 4, which means there are 4 particular groups of customers in the dataset.



The procedure was executed for 4 cluster and the resulting groups were then profiled to interpret their characteristics, which was followed by the generation of visualizations such as bar charts, box plots, and pie charts. These graphical outputs provide a clear and precise interpretation of the clustering results, enabling actionable insights. Finally, the cluster assignment was integrated back into the original data frame, thus completing the clustering process with the comparison of sales statistics and job dates.

R Script Summary

This section provides a summary of the R script, which is present in the appendix.

Data Cleaning

Date Conversion: Excel-based dates in "Job.Date" and "Sold.Date" were converted into R-friendly date formats.

Months.Since.Purchase: A new variable was calculated to represent the time interval between the sold date and job date in months.

Duplicate Removal: Duplicate entries, based on addresses, were eliminated while retaining the entry with the earliest 'Months.Since.Purchase'.

Returning.Customer: A binary indicator for returning customers was generated.

State Extraction: The state information was extracted from the full address and added as a separate column.

Data Preprocessing

Missing Values: A summary of missing values across important variables was generated.

Subsetting: The dataset was filtered based on discovery confidence level and variables relevant to clustering were selected.

Complete Cases: Only observations without any missing values were retained.

Clustering Algorithm

Data Transformation: Variables were transformed into appropriate data types (numeric for numerical variables and factor for categorical variables).

Gower Distance Calculation: A Gower distance matrix was computed to handle mixed data types.

Optimal Cluster Number: The optimal number of clusters was determined by maximizing the average silhouette width.

PAM Clustering: Partitioning Around Medoids (PAM) was applied with the optimal number of clusters and the resultant cluster assignments were added to the original data.

Cluster Profiling

Summary Statistics: For each cluster, key summary statistics like mean and proportion for each variable were calculated.

Visualization: Bar charts, boxplots, and pie charts were generated to visualize the cluster profiles.

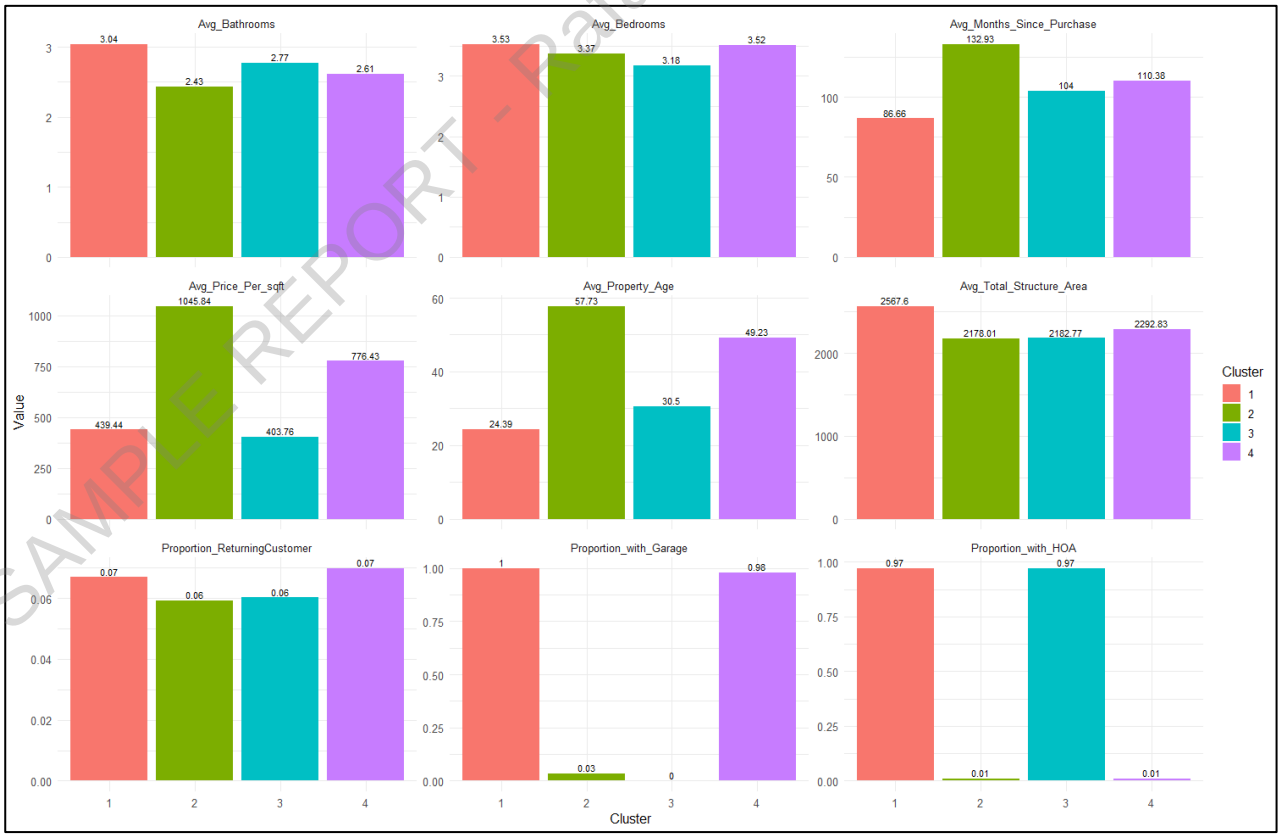
Cluster Profiling

After clusters were assigned to each member of the dataset, the cluster sizes were examined (table below). The cluster analysis reveals four distinct customer segments with varying sizes and proportions. Cluster 4 is the largest, comprising 31.35% of the total customer base, followed by Cluster 2 at 28.96%. Clusters 1 and 3 are relatively smaller, accounting for 23.23% and 16.45% respectively. This distribution suggests that marketing resources could be most effectively allocated by focusing on Clusters 4 and 2, which collectively make up over 60% of the customer base. However, Clusters 1 and 3 should not be overlooked, as they represent significant, albeit smaller, market segments.

Cluster	Size	Relative_Percentage
1	2885	23.23
2	3596	28.96
3	2043	16.45
4	3893	31.35

The following table provides statistics of each found cluster, followed by a figure with bar charts that illustrate the differences across clusters.

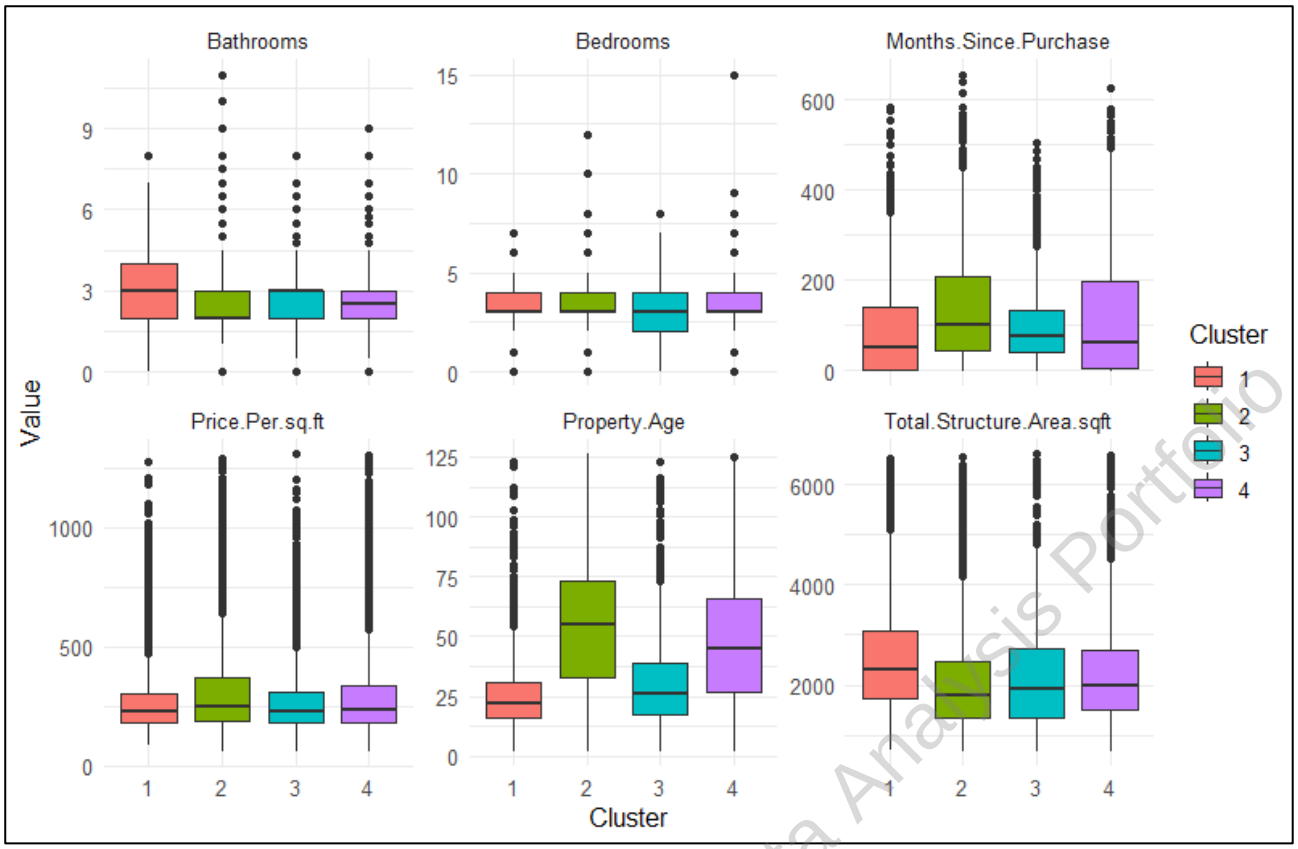
Cluster	1	2	3	4
Avg_Price_Per_sqft	$439.44	$1045.84	$403.76	$776.43
Avg_Property_Age	24.39	57.73	30.50	49.23
Proportion_with_HOA	97%	1%	97%	1%
Proportion_with_Garage	100%	3%	0%	98%
Avg_Bedrooms	3.53	3.37	3.18	3.52
Avg_Bathrooms	3.04	2.43	2.77	2.61
Avg_Total_Structure_Area	2567.60	2178.01	2182.77	2292.83
Avg_Months_Since_Purchase	86.66	132.93	104.00	110.38
Proportion_ReturningCustomer	7%	6%	6%	7%
Most_Common_State	FL	PA	FL	MI



A brief interpretation of the cluster profiles is given below:

- Cluster 1: This group generally prefers moderately priced properties with an average price of $439.44 per sqft. The properties are relatively new with an average age of 24.39 years. Almost all properties have Homeowners Associations (HOA) and garages. The properties tend to be spacious with an average of 3.53 bedrooms and 3.04 bathrooms.
- Cluster 2: This is a high-cost segment with an average price of $1045.84 per sqft. The properties are older with an average age of 57.73 years. HOA and garages are uncommon in this group. Despite the high cost, the properties are smaller in size, with an average of 3.37 bedrooms and 2.43 bathrooms.
- Cluster 3: A more economical choice, properties in this cluster have an average price of $403.76 per sqft. These are middle-aged properties with an average age of 30.5 years. Almost all have HOA but lack garages. The properties have an average of 3.18 bedrooms and 2.77 bathrooms.
- Cluster 4: This segment is marked by a balance between price and quality, with an average price of $776.43 per sqft. The properties are older with an average age of 49.23 years. Garages are common, but HOA is rare. The properties offer an average of 3.52 bedrooms and 2.61 bathrooms.

The figure below shows boxplots of the numerical variables that were used in the clustering process. A boxplot provides a graphical representation of a dataset's five-number summary: the minimum, first quartile (Q1), median, third quartile (Q3), and maximum. The "box" in the plot encapsulates the interquartile range (IQR), stretching from Q1 to Q3, with a line marking the median. Whiskers extend from the box to the minimum and maximum values that are not outliers.



The table below provides a breakdown of clusters by the state where respondents are located, along with the number (n) and the proportion (%) within each cluster.

Cluster 1, primarily based in Florida (FL), comprises 24.06% of the cluster's total. Other notable states include Colorado (CO), Georgia (GA), California (CA), and Virginia (VA).

In Cluster 2, Pennsylvania (PA) is the most represented state with 12.71%, followed by states like Nebraska (NE), New Jersey (NJ), Georgia (GA), and Virginia (VA).

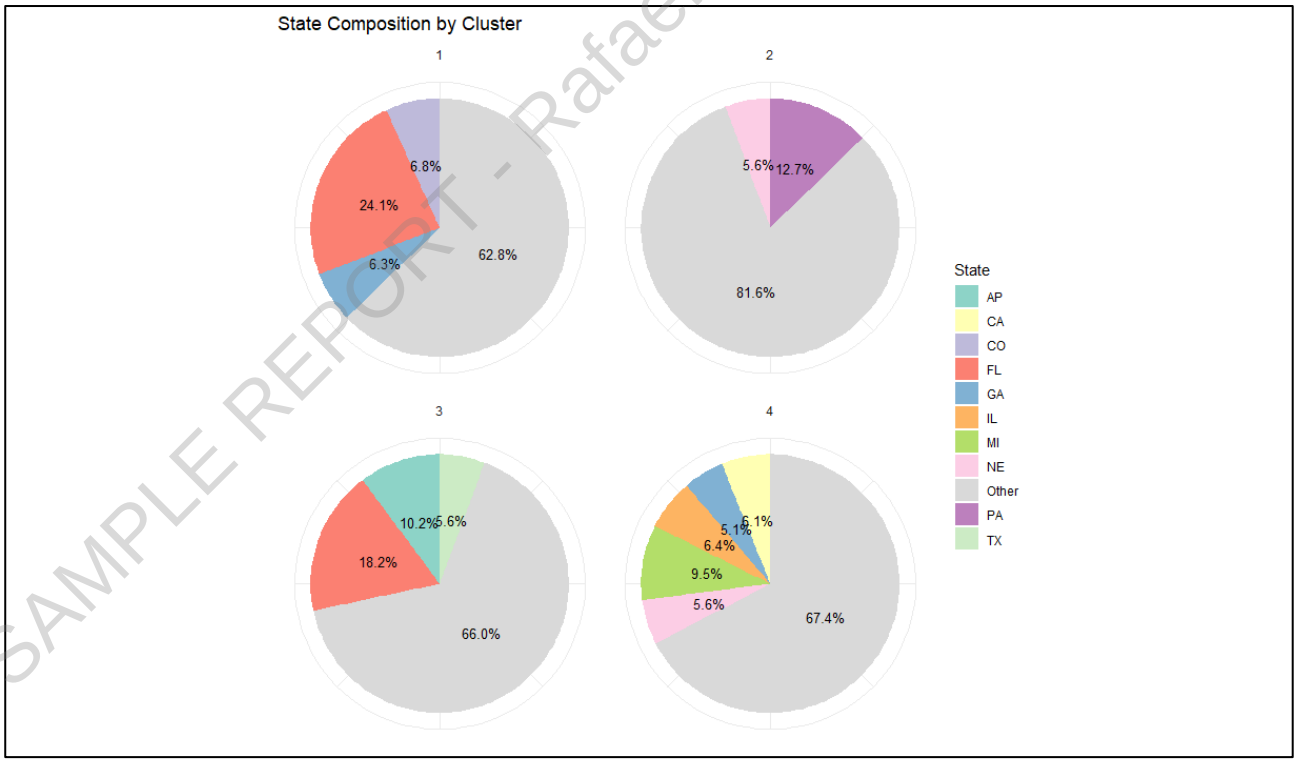
Cluster 3 is also notably influenced by respondents from Florida (FL), representing 18.16%. Other significant states include Armed Forces Pacific (AP), Texas (TX), Virginia (VA), and Maryland (MD).

Cluster 4 is most prevalent in Michigan (MI), accounting for 9.45% of the cluster, followed by Illinois (IL), California (CA), Nebraska (NE), and Georgia (GA).

This geographic information can be useful in tailoring marketing strategies specific to each cluster's dominant locations.

Cluster	State	n	% within Cluster
1	FL	694	24.06
1	CO	196	6.79
1	GA	183	6.34
1	CA	135	4.68
1	VA	114	3.95
2	PA	457	12.71
2	NE	203	5.65
2	NJ	165	4.59
2	GA	164	4.56
2	VA	162	4.51
3	FL	371	18.16
3	AP	208	10.18
3	TX	115	5.63
3	VA	102	4.99
3	MD	96	4.70
4	MI	368	9.45
4	IL	248	6.37
4	CA	239	6.14
4	NE	218	5.60
4	GA	198	5.09

Figure below shows the same information in Pie Charts.



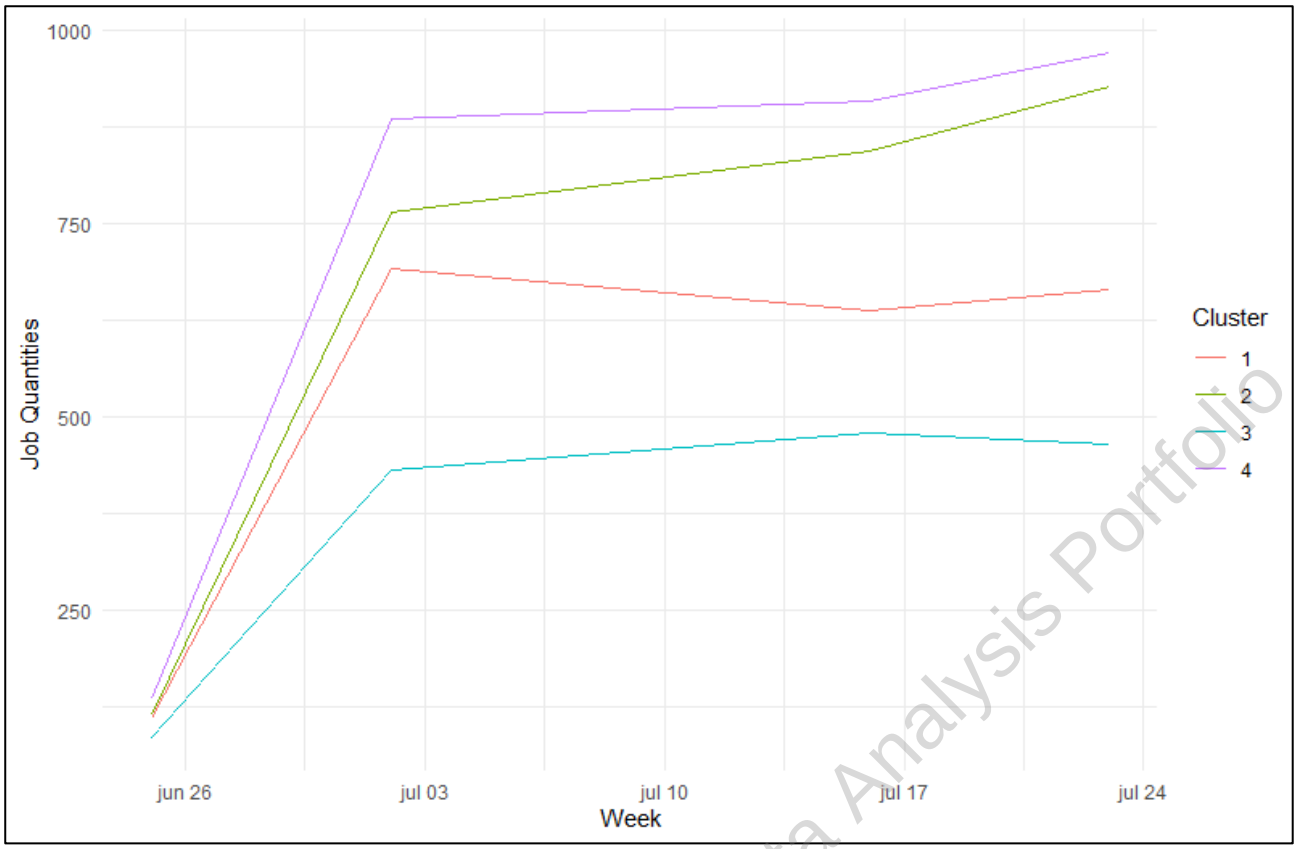
Finally, the report now proceeds to evaluating sales statistics. The table below presents key statistics on the average order value for each of the four clusters: mean, median, minimum, and maximum values.

Cluster 4 exhibits the highest mean order value at $343.59, as well as the highest median value at $295.00. Conversely, Cluster 3 shows the lowest mean order value at $310.69 and a median value of $269.00, slightly below the others. Clusters 1 and 2 are relatively similar in their average order values, with means of $329.51 and $325.84, respectively.

These statistics provide insights into the purchasing power or spending habits of each cluster, which can be instrumental for targeted marketing efforts.

Cluster	mean	median	min	max
1	$ 329.51	$ 289.00	$ -	$ 989.00
2	$ 325.84	$ 274.00	$ -	$ 999.97
3	$ 310.69	$ 269.00	$ -	$ 974.00
4	$ 343.59	$ 295.00	$ -	$ 998.00

Finally, the number of orders per cluster was evaluated weekly and is plotted in the graph shown below. For the week ending June 25, 2023, Cluster 4 had the highest job count (136), while Cluster 3 had the lowest (86). The subsequent week, ending July 2, saw a significant increase in job numbers across all clusters, with Cluster 4 again taking the lead at 884 jobs and Cluster 3 lagging at 431. This trend of Cluster 4 having the highest and Cluster 3 having the lowest count persists in later weeks as well. The job numbers for the week ending July 16 decreased compared to July 2 but still maintained a level considerably higher than the initial week. The week ending July 23 shows a minor increase in job counts compared to July 16. In summary, Cluster 4 consistently has the highest job engagement, and the job counts are notably volatile from week to week.



Regional Segmentation

This section presents the results of a regional segmentation in three levels: by state; by region and by subregions.

States

California stands out with an exceptionally high average price per sqft of 773.709, significantly above the national average. Conversely, states like Oklahoma register low prices per sqft, at just 136.792. Property age varies considerably, with Washington, D.C., having an average property age of 79.297 years, in contrast to states like Arizona where the average property is much newer, at 31.846 years.

The presence of HOAs is more prevalent in states like Alabama (77.1%) compared to Connecticut (4.9%). Interestingly, the proportion of properties with garages is higher in Arkansas (77.6%) than in states like California (58.1%). Finally, Washington, D.C. has the highest percentage of returning customers at 10.9%.

Regions

Mid-Atlantic region stands out for its high average price per square foot at $2651.739, significantly above other regions. It also leads in terms of average total structure area and property age, indicating larger and older properties. Notably, New England follows with an average property age of 63.8 years.

The South Atlantic region leads in the number of orders, yet the properties are generally younger with an average age of 34.7 years. Only 53.1% of properties in this region have a HOA.

In terms of customer retention, no region stands out significantly, with the numbers fluctuating modestly around 6%.

Megaregions

The Northeast region has an average price per square foot of $1953.432 and lower percentages of properties with HOAs and garages at 20.5% and 44.8%, respectively. The South region has the highest number of orders at 4999 and an average property age of 34.482 years. It also shows that 52.0% of properties have HOAs. The West region has an average price per square foot of $601.422 and shows that 37.5% of properties have HOAs and 62.3% have garages. Across all macro regions, customer retention rates are relatively similar, ranging from 6.3% to 6.6%.

State	N. of Orders	Avg Bedrooms	Avg Bathrooms	Avg Price Per sqft	Avg Property Age	Avg Total Structure Area	Avg Months Since Purchase	Avg Aggregate Sub Total	Prop Has HOA	Prop Has Garage	Prop Returning Customer
AL	35	3.571	2.829	225.857	18.657	2288.000	50.857	260.434	77.1%	40.0%	2.9%
AR	85	3.447	2.688	180.306	26.353	2248.118	57.059	308.123	49.4%	77.6%	2.4%
AZ	65	3.415	2.450	274.415	31.846	2130.523	67.969	282.571	70.8%	49.2%	6.2%
CA	677	3.386	2.651	773.709	45.638	2053.579	126.793	294.609	41.1%	58.1%	7.1%
CO	505	3.539	2.818	307.598	33.709	2345.552	110.752	311.021	55.6%	69.9%	6.1%
CT	243	3.560	2.858	303.646	57.329	2513.856	146.202	332.834	4.9%	60.1%	5.3%
DC	175	3.017	2.631	591.166	79.297	1881.377	110.629	373.699	32.6%	16.6%	10.9%
DE	53	3.453	2.462	187.849	43.566	2210.321	124.434	233.563	60.4%	54.7%	11.3%
FL	1871	3.196	2.519	295.148	31.862	2131.760	103.004	331.638	65.4%	56.9%	6.7%
GA	775	3.825	3.118	825.539	29.853	2690.143	108.883	326.046	44.4%	59.0%	5.5%
IA	57	3.509	2.816	210.789	36.316	1742.175	94.035	245.932	31.6%	66.7%	8.8%
ID	26	3.654	2.500	286.269	33.500	2419.385	138.923	213.859	57.7%	38.5%	7.7%
IL	572	3.374	2.657	660.406	47.921	2293.757	98.267	328.197	37.1%	65.4%	8.4%
IN	115	3.704	3.027	151.826	36.096	3302.417	71.043	248.639	59.1%	73.0%	10.4%
KS	81	3.444	2.735	173.407	46.272	2093.049	123.728	288.074	40.7%	66.7%	6.2%
KY	33	3.455	2.955	174.000	32.091	2285.636	106.212	287.684	39.4%	42.4%	0.0%
LA	24	3.417	2.333	152.708	27.958	2024.000	93.042	307.700	33.3%	25.0%	12.5%
MA	343	3.493	2.472	444.994	70.551	2293.087	139.548	383.544	7.6%	43.7%	5.5%
MD	408	3.716	3.344	279.304	42.441	2587.895	87.775	423.192	47.5%	41.2%	6.4%
MI	664	3.316	2.471	216.586	53.767	2113.217	91.753	326.796	27.6%	58.0%	6.3%
MN	388	3.405	2.458	220.809	48.887	2058.152	121.312	247.790	26.3%	60.8%	5.4%
MO	151	3.272	2.667	194.411	44.172	2206.828	138.834	350.345	51.0%	60.3%	8.6%
NC	417	3.302	2.697	235.717	32.153	2620.412	106.573	297.585	43.9%	48.2%	6.5%
NE	112	3.277	2.712	182.295	41.438	2159.402	102.964	326.551	43.8%	62.5%	4.5%
NH	34	3.235	2.456	346.794	48.029	2246.912	91.971	431.560	5.9%	55.9%	5.9%
NJ	274	3.485	2.776	9044.387	51.763	2814.631	122.566	365.134	19.0%	26.6%	6.2%
NV	71	3.070	2.444	305.324	26.817	1987.986	93.887	283.357	62.0%	67.6%	5.6%
NY	219	3.680	2.893	437.078	61.849	2608.174	91.187	393.490	8.2%	59.4%	5.9%
OH	411	3.421	2.660	219.839	44.234	2241.983	116.061	289.334	34.5%	63.0%	4.1%
OK	48	3.271	2.302	136.792	34.438	2105.396	77.938	267.697	16.7%	75.0%	6.3%
OR	223	3.399	2.575	333.121	41.170	2176.359	119.713	320.013	33.6%	67.7%	4.5%
PA	660	3.432	2.575	2397.139	53.724	2321.779	143.002	364.533	19.7%	41.8%	7.9%
RI	94	3.309	2.537	313.543	61.723	2124.000	111.617	368.700	11.7%	56.4%	6.4%

State	N. of Orders	Avg Bedrooms	Avg Bathrooms	Avg Price Per sqft	Avg Property Age	Avg Total Structure Area	Avg Months Since Purchase	Avg Aggregate Sub Total	Prop Has HOA	Prop Has Garage	Prop Returning Customer
SC	280	3.554	2.850	1062.071	32.139	2384.914	93.443	306.685	38.9%	44.3%	5.7%
TN	318	3.412	2.744	259.579	39.447	2453.858	101.689	308.957	34.0%	53.5%	7.2%
TX	395	3.473	2.748	236.220	31.727	2470.420	101.906	296.988	59.2%	44.1%	7.3%
UT	12	4.083	2.521	278.833	51.667	2241.500	113.750	267.714	16.7%	66.7%	0.0%
VA	531	3.605	2.997	336.936	40.712	2421.072	126.373	323.153	44.8%	46.7%	5.3%
WA	738	3.329	2.413	799.141	47.927	2131.167	120.718	401.705	17.2%	60.8%	6.5%
WI	222	3.365	2.525	257.450	52.716	2339.261	75.450	325.680	14.9%	63.1%	5.4%
WV	12	3.167	2.542	159.833	30.917	1846.500	139.583	408.556	25.0%	16.7%	0.0%

Region	N. of Orders	Avg Bedrooms	Avg Bathrooms	Avg Price Per sqft	Avg Property Age	Avg Total Structure Area	Avg Months Since Purchase	Avg Aggregate Sub Total	Prop Has HOA	Prop Has Garage	Prop Returning Customer
East North Central	1984	3.383	2.602	346.035	48.965	2286.166	95.642	314.797	32.2%	62.6%	6.6%
East South Central	386	3.430	2.769	249.205	36.933	2424.438	97.466	303.076	38.3%	51.3%	6.2%
Mid-Atlantic	1614	3.547	2.843	2651.739	51.308	2507.919	117.931	378.524	26.4%	41.9%	7.1%
Mountain	679	3.492	2.726	302.859	33.119	2288.567	106.025	302.025	57.1%	66.4%	6.0%
New England	714	3.479	2.611	374.906	63.817	2343.763	135.870	366.108	7.1%	51.5%	5.6%
Pacific	1638	3.362	2.533	725.185	46.061	2105.252	123.092	345.362	29.3%	60.6%	6.5%
South Atlantic	4061	3.397	2.742	460.964	34.726	2332.150	107.325	326.277	53.1%	52.4%	6.4%
West North Central	789	3.373	2.588	204.700	45.750	2081.734	120.338	281.325	35.4%	62.0%	6.2%
West South Central	552	3.449	2.682	215.333	30.971	2385.038	92.531	296.789	52.9%	51.1%	6.7%

Macro Region	N. of Orders	Avg Bedrooms	Avg Bathrooms	Avg Price Per sqft	Avg Property Age	Avg Total Structure Area	Avg Months Since Purchase	Avg Aggregate Sub Total	Prop Has HOA	Prop Has Garage	Prop Returning Customer
Midwest	2773	3.380	2.598	305.821	48.050	2227.999	102.669	305.068	33.1%	62.4%	6.5%
Northeast	2328	3.526	2.772	1953.432	55.144	2457.572	123.433	374.777	20.5%	44.8%	6.6%
South	4999	3.405	2.737	417.489	34.482	2345.116	104.930	321.125	52.0%	52.1%	6.4%
West	2317	3.400	2.590	601.422	42.268	2158.972	118.091	332.437	37.5%	62.3%	6.3%

Appendix – R Script

```
library(dplyr)
library(openxlsx)
library(lubridate)
library(stringr)
library(ggplot2)

setwd("C:/Users/rafre/Dropbox/Fiverr/Trabalhos/2023/kevinh75")

df <- read.xlsx("ClusterData_9_6_23.xlsx")

## DATA CLEANING ##

# Define the function to convert Excel date to R date
convert_excel_date <- function(excel_date) {
  as.Date(excel_date, origin="1899-12-30")
}

# Apply the function to the Job.Date and Sold.Date columns
df <- df %>%
  mutate(Job.Date = convert_excel_date(Job.Date),
         Sold.Date = convert_excel_date(Sold.Date))

# Calculate the difference in months and add as a new column
df <- df %>%
  mutate(Months.Since.Purchase = as.numeric(round(interval(Sold.Date, Job.Date) / months(1), 0)))

# Removing duplicates by address

# Group by address and sort by Months.Since.Purchase
df <- df %>%
  arrange(Zillow.Full.Address, Months.Since.Purchase)

# Remove duplicates by keeping the first entry for each address group
df_unique <- df %>%
  group_by(Zillow.Full.Address) %>%
  slice_head(n = 1) %>%
  ungroup()

# Add a column for the number of occurrences for each address
count_occurrences <- df %>%
  group_by(Zillow.Full.Address) %>%
  summarise(Num_Occurrences = n(), .groups = 'drop')

# Merge the occurrence counts into df_unique
df_unique <- left_join(df_unique, count_occurrences, by = "Zillow.Full.Address")

# Print the number of cases with 1, 2, 3, and more occurrences
occurrence_count <- table(count_occurrences$Num_Occurrences)
print(occurrence_count)

#Recode Num_Occurrences
```

```
df_unique$Num_Occurrences <- ifelse(df_unique$Num_Occurrences == 1, "No", "Yes")

# Rename the column
colnames(df_unique)[colnames(df_unique) == "Num_Occurrences"] <- "Returning.Customer"

## Extracting the State

# Extract the state as a new column
df_unique <- df_unique %>%
  mutate(State = str_extract(Zillow.Full.Address, "(?<=, )([A-Z]{2})(?=,)"))

# View the modified dataframe to confirm
head(df_unique)

## DIAGNOSE MISSING VALUES

# Count missing values in specific columns
missing_values_summary <- df_unique %>%
  summarise(
    missing_bedrooms = sum(is.na(Bedrooms)),
    missing_bathrooms = sum(is.na(Bathrooms)),
    missing_price_per_sqft = sum(is.na(Price.Per.sq.ft)),
    missing_has_hoa = sum(is.na(Has.HOA)),
    missing_property_age = sum(is.na(Property.Age)),
    missing_total_structure_area_sqft = sum(is.na(`Total.Structure.Area.(sqft)`)),
    missing_has_garage = sum(is.na(Has.Garage)),
    missing_months_since_purchase = sum(is.na(Months.Since.Purchase)),
    missing_num_occurrences = sum(is.na(Returning.Customer)),
    missing_state = sum(is.na(State))
  )

# Print the summary to the console
print(missing_values_summary)

# Rename columns
df_unique <- df_unique %>% rename(Total.Structure.Area.sqft = `Total.Structure.Area.(sqft)`)


# Calculate the frequency table for the Discovery_Confidence_Level column
freq_table <- table(df_unique$Discovery.Confidence.Level)


# Print the frequency table
print(freq_table) ## LOW NUMBER OF LOW AND MEDIUM CONFIDENCE

# Subset the dataframe to only include rows where Discovery_Confidence_Level is 'High'
df_unique_high <- subset(df_unique, Discovery.Confidence.Level == "High")

# Subset data with relevant variables
df_unique_selected <- df_unique_high[, c('Bedrooms', 'Bathrooms', 'Price.Per.sq.ft', 'Has.HOA',
'Property.Age',
                                         'Total.Structure.Area.sqft', 'Has.Garage', 'Months.Since.Purchase',
```

```
                                'Returning.Customer', 'State')]
# Get Complete Cases Only

# Find the complete cases
complete_cases_index <- complete.cases(df_unique_selected)

# Count the number of complete cases
num_complete_cases <- sum(complete_cases_index)

# Create a new data frame with only complete cases
df_complete_cases <- df_unique_selected[complete_cases_index, ]

# Create a new whole data frame with only complete cases
df_unique_high <- df_unique_high[complete_cases_index, ]

## CLUSTERING ALGORITHM

library(cluster)
library(fpc)

# Convert to factor for categorical variables
df_complete_cases$Has.HOA <- as.factor(df_complete_cases$Has.HOA)
df_complete_cases$Has.Garage <- as.factor(df_complete_cases$Has.Garage)
df_complete_cases$State <- as.factor(df_complete_cases$State)
df_complete_cases$Returning.Customer <- as.factor(df_complete_cases$Returning.Customer)

# Convert to numeric for numerical variables
df_complete_cases$Bedrooms <- as.numeric(df_complete_cases$Bedrooms)
df_complete_cases$Bathrooms <- as.numeric(df_complete_cases$Bathrooms)
df_complete_cases$Price.Per.sq.ft <- as.numeric(df_complete_cases$Price.Per.sq.ft)
df_complete_cases$Property.Age <- as.numeric(df_complete_cases$Property.Age)
df_complete_cases$Total.Structure.Area.sqft                                    <-
as.numeric(df_complete_cases$Total.Structure.Area.sqft)
df_complete_cases$Months.Since.Purchase                                        <-
as.numeric(df_complete_cases$Months.Since.Purchase)

## DEFINING OPTIMAL NUMBER OF CLUSTERS

# Calculate Gower distance matrix for df_complete_cases
gower_dist <- daisy(df_complete_cases, metric = "gower")

# Initialize the vector to store silhouette widths
silhouette_width <- numeric(7)

# Loop to calculate silhouette widths for k = 2 to k = 7
for (k in 2:7) {
  pam_fit <- pam(gower_dist, diss = TRUE, k = k)
  s_width <- silhouette(pam_fit)
  silhouette_width[k] <- mean(s_width[, 3])
}

# Plotting the silhouette widths
```

```
plot(2:7, silhouette_width[2:7], type = 'b',
     xlab = "Number of clusters", ylab = "Average silhouette width",
     main = "Average silhouette for varying number of clusters")

#Run Solution with 4 clusters
set.seed(123)
pam_fit <- pam(gower_dist, diss = TRUE, k = 4)


# Adding the cluster assignments to your original data frame
df_complete_cases$Cluster <- pam_fit$clustering

## CLUSTER PROFILING

library(dplyr)

# CLUSTER PROFILES

# Calculate cluster sizes and relative percentages
cluster_sizes <- df_complete_cases %>%
  group_by(Cluster) %>%
  summarize(
    Size = n(),
    Relative_Percentage = (n() / nrow(.)) * 100
  )

# Display the dataframe
print(cluster_sizes)


cluster_profile <- df_complete_cases %>%
  group_by(Cluster) %>%
  summarize(
    Avg_Price_Per_sqft = mean(Price.Per.sq.ft, na.rm = TRUE),
    Avg_Property_Age = mean(Property.Age, na.rm = TRUE),
    Proportion_with_HOA = mean(Has.HOA == "Yes", na.rm = TRUE),
    Proportion_with_Garage = mean(Has.Garage == "Yes", na.rm = TRUE),
    Avg_Bedrooms = mean(Bedrooms, na.rm = TRUE),
    Avg_Bathrooms = mean(Bathrooms, na.rm = TRUE),
    Avg_Total_Structure_Area = mean(Total.Structure.Area.sqft, na.rm = TRUE),
    Avg_Months_Since_Purchase = mean(Months.Since.Purchase, na.rm = TRUE),
    Proportion_ReturningCustomer = mean(Returning.Customer == "Yes", na.rm = TRUE),
    Most_Common_State = names(sort(table(State), decreasing = TRUE)[1])
  )

# Count the occurrences of each state in each cluster and calculate total per cluster
common_states <- df_complete_cases %>%
  group_by(Cluster, State) %>%
  summarise(n = n()) %>%
  mutate(total_per_cluster = sum(n, na.rm = TRUE))

# Calculate the percentage for each state within its cluster
```

```
common_states <- common_states %>%
  mutate(percentage = (n / total_per_cluster) * 100) %>%
  arrange(Cluster, desc(percentage))

# Get the top 5 states for each cluster, with percentages
top_states_per_cluster <- common_states %>%
  group_by(Cluster) %>%
  slice_head(n = 5) %>%
  ungroup()

library(ggplot2)
library(tidyverse)

# Create a long-form version of the cluster_profile dataframe for easier plotting
cluster_profile_long <- cluster_profile %>%
  select(-Most_Common_State) %>%
  pivot_longer(cols = -Cluster, names_to = "Variable", values_to = "Value")


# Generate the charts
ggplot(cluster_profile_long, aes(x = factor(Cluster), y = Value, fill = factor(Cluster))) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label=round(Value, 2)), vjust=-0.3, size = 3) +
  facet_wrap(~ Variable, scales = "free_y") + # Notice the "free_y" to allow for a dynamic y-axis
  theme_minimal() +
  theme(text=element_text(size=12)) +
  xlab("Cluster") +
  ylab("Value") +
  labs(fill = "Cluster")

#BoxPlots

numerical_vars <- c("Price.Per.sq.ft", "Property.Age", "Bedrooms", "Bathrooms",
                    "Total.Structure.Area.sqft", "Months.Since.Purchase")

## Remove Extreme outliers
df_filtered_boxplots <- df_complete_cases %>%
  filter(
    between(Price.Per.sq.ft, quantile(Price.Per.sq.ft, 0.01), quantile(Price.Per.sq.ft, 0.99)),
    between(Property.Age, quantile(Property.Age, 0.01), quantile(Property.Age, 0.99)),
    between(Total.Structure.Area.sqft,                quantile(Total.Structure.Area.sqft,                0.01),
quantile(Total.Structure.Area.sqft, 0.99))
  )

df_complete_cases_long <- df_filtered_boxplots %>%
  select(Cluster, all_of(numerical_vars)) %>%
  pivot_longer(cols = -Cluster, names_to = "Variable", values_to = "Value")

# Create the boxplots
ggplot(df_complete_cases_long, aes(x = factor(Cluster), y = Value, fill = factor(Cluster))) +
  geom_boxplot() +
  facet_wrap(~ Variable, scales = "free_y") +
```

```
  theme_minimal() +
  theme(text=element_text(size=12)) +
  xlab("Cluster") +
  ylab("Value") +
  labs(fill = "Cluster")

# Pie Charts
# Calculate state frequency within each cluster
state_frequency <- df_complete_cases %>%
  group_by(Cluster, State) %>%
  summarise(n = n()) %>%
  mutate(percentage = n / sum(n) * 100)

# Group states with less than 5% representation within each cluster
state_frequency_grouped <- state_frequency %>%
  mutate(State = if_else(percentage < 5, "Other", as.character(State))) %>%
  group_by(Cluster, State) %>%
  summarise(percentage = sum(percentage), .groups = 'drop')

# Generate pie charts
ggplot(state_frequency_grouped, aes(x = "", y = percentage, fill = State)) +
  geom_bar(stat = "identity", width = 1) +
  geom_text(aes(label = sprintf("%.1f%%", percentage)), position = position_stack(vjust = 0.5), size
= 3) +
  coord_polar("y") +
  facet_wrap(~ Cluster) +
  scale_fill_brewer(palette = "Set3") +
  labs(title = "State Composition by Cluster", x = NULL, y = NULL, fill = "State") +
  theme_minimal() +
  theme(axis.text.x = element_blank(), legend.position = "right")


# COMPARE NON CLUSTERING VARIBLES

# Initialize a new column in df for cluster labels, setting it to NA initially
df_unique_high$Cluster <- NA

# Assign cluster labels to the original df using the same indices as in df_complete_cases
df_unique_high$Cluster[complete_cases_index] <- df_complete_cases$Cluster

# Remove rows where Cluster is NA
df_unique_high_filtered <- df_unique_high %>% filter(!is.na(Cluster))

# Filter out incomplete weeks
complete_weeks <- df_unique_high_filtered %>%
  group_by(Week) %>%
  summarise(Num_Jobs = n(), .groups = 'drop') %>%
  filter(Num_Jobs != min(Num_Jobs) & Num_Jobs != max(Num_Jobs)) %>%
  select(Week)

df_unique_high_filtered <- df_unique_high_filtered %>%
  filter(Week %in% complete_weeks$Week)
```

```
# Re-generate df_weekly with the complete weeks
df_weekly <- df_unique_high_filtered %>%
  group_by(Week, Cluster) %>%
  summarise(Num_Jobs = n(), .groups = 'drop')


# Create the graph
ggplot(df_weekly, aes(x = Week, y = Num_Jobs, color = as.factor(Cluster))) +
  geom_line() +
  theme_minimal() +
  xlab("Week") +
  ylab("Job Quantities") +
  labs(color = "Cluster")

df_unique_high_filtered$Aggregate.Sub.Total <-
as.numeric(df_unique_high_filtered$Aggregate.Sub.Total)

# Order Service Statistic

df_sales_stats_perCluster <- df_unique_high_filtered %>%
  group_by(Cluster) %>%
  summarise(mean = mean(Aggregate.Sub.Total, na.rm = TRUE),
            median = median(Aggregate.Sub.Total, na.rm = TRUE),
            min = min(Aggregate.Sub.Total, na.rm = TRUE),
            max = max(Aggregate.Sub.Total, na.rm = TRUE),
            .groups = 'drop')

# Create the boxplot
ggplot(df_unique_high_filtered, aes(x = as.factor(Cluster), y = Aggregate.Sub.Total, color =
as.factor(Cluster))) +
  geom_boxplot(outlier.shape = NA) +  # exclude outliers for better visualization
  theme_minimal() +
  xlab("Cluster") +
  ylab("Aggregate Sub Total") +
  labs(color = "Cluster")


### A PRIOR SEGMENTATION

# Create columns for regions
df_unique_high <- df_unique_high %>%
  mutate(Region = case_when(
    State %in% c("CT", "ME", "MA", "NH", "RI", "VT") ~ "New England",
    State %in% c("DE", "MD", "NJ", "NY", "PA") ~ "Mid-Atlantic",
    State %in% c("IL", "IN", "MI", "OH", "WI") ~ "East North Central",
    State %in% c("IA", "KS", "MN", "MO", "NE", "ND", "SD") ~ "West North Central",
    State %in% c("DC", "FL", "GA", "MD", "NC", "SC", "VA", "WV") ~ "South Atlantic",
    State %in% c("AL", "KY", "MS", "TN") ~ "East South Central",
    State %in% c("AR", "LA", "OK", "TX") ~ "West South Central",
    State %in% c("AZ", "CO", "ID", "MT", "NV", "NM", "UT", "WY") ~ "Mountain",
    State %in% c("AK", "CA", "HI", "OR", "WA") ~ "Pacific",
```

```
    TRUE ~ "Other"
  ))

df_unique_high <- df_unique_high %>%
  mutate(Macro_Region = case_when(
    State %in% c("CT", "ME", "MA", "NH", "RI", "VT", "DE", "MD", "NJ", "NY", "PA") ~
"Northeast",
    State %in% c("IL", "IN", "MI", "OH", "WI", "IA", "KS", "MN", "MO", "NE", "ND", "SD") ~
"Midwest",
    State %in% c("DC", "FL", "GA", "MD", "NC", "SC", "VA", "WV", "AL", "KY", "MS", "TN",
"AR", "LA", "OK", "TX") ~ "South",
    State %in% c("AZ", "CO", "ID", "MT", "NV", "NM", "UT", "WY", "AK", "CA", "HI", "OR",
"WA") ~ "West",
    TRUE ~ "Other"
  ))

# Create a dataframe

# Convert to factor for categorical variables
df_unique_high$Has.HOA <- as.factor(df_unique_high$Has.HOA)
df_unique_high$Has.Garage <- as.factor(df_unique_high$Has.Garage)
df_unique_high$State <- as.factor(df_unique_high$State)
df_unique_high$Returning.Customer <- as.factor(df_unique_high$Returning.Customer)

# Convert to numeric for numerical variables
df_unique_high$Bedrooms <- as.numeric(df_unique_high$Bedrooms)
df_unique_high$Bathrooms <- as.numeric(df_unique_high$Bathrooms)
df_unique_high$Price.Per.sqft <- as.numeric(df_unique_high$Price.Per.sq.ft)
df_unique_high$Property.Age <- as.numeric(df_unique_high$Property.Age)
df_unique_high$Total.Structure.Area.sqft <- as.numeric(df_unique_high$Total.Structure.Area.sqft)
df_unique_high$Months.Since.Purchase <- as.numeric(df_unique_high$Months.Since.Purchase)
df_unique_high$Aggregate.Sub.Total <- as.numeric(df_unique_high$Aggregate.Sub.Total)

# Grouping by Region, Macro Region and State
df_by_region_all <- df_unique_high %>%
  group_by(Macro_Region, Region, State) %>%
  summarise(
    Avg_Bedrooms = mean(Bedrooms, na.rm = TRUE),
    Avg_Bathrooms = mean(Bathrooms, na.rm = TRUE),
    Avg_Price_Per_sqft = mean(Price.Per.sq.ft, na.rm = TRUE),
    Avg_Property_Age = mean(Property.Age, na.rm = TRUE),
    Avg_Total_Structure_Area = mean(Total.Structure.Area.sqft, na.rm = TRUE),
    Avg_Months_Since_Purchase = mean(Months.Since.Purchase, na.rm = TRUE),
    Avg_Aggregate_Sub_Total = mean(Aggregate.Sub.Total, na.rm = TRUE),
  .groups = 'drop'
  )

# Summarize by State
df_by_state <- df_unique_high %>%
  group_by(State) %>%
  summarise(
    Sample_Size = n(),
```

```
    Avg_Bedrooms = mean(Bedrooms, na.rm = TRUE),
    Avg_Bathrooms = mean(Bathrooms, na.rm = TRUE),
    Avg_Price_Per_sqft = mean(Price.Per.sqft, na.rm = TRUE),
    Avg_Property_Age = mean(Property.Age, na.rm = TRUE),
    Avg_Total_Structure_Area = mean(Total.Structure.Area.sqft, na.rm = TRUE),
    Avg_Months_Since_Purchase = mean(Months.Since.Purchase, na.rm = TRUE),
    Avg_Aggregate_Sub_Total = mean(Aggregate.Sub.Total, na.rm = TRUE),
    Prop_Has_HOA = sum(Has.HOA == "Yes") / n(),
    Prop_Has_Garage = sum(Has.Garage == "Yes") / n(),
    Prop_Returning_Customer = sum(Returning.Customer == "Yes") / n(),
    .groups = 'drop'
  )

# Filter cases with sample_size >= 5
df_by_state_filtered <- df_by_state %>% filter(Sample_Size >= 5)

# Summarize by Region
df_by_region <- df_unique_high %>%
  group_by(Region) %>%
  summarise(
    Sample_Size = n(),
    Avg_Bedrooms = mean(Bedrooms, na.rm = TRUE),
    Avg_Bathrooms = mean(Bathrooms, na.rm = TRUE),
    Avg_Price_Per_sqft = mean(Price.Per.sqft, na.rm = TRUE),
    Avg_Property_Age = mean(Property.Age, na.rm = TRUE),
    Avg_Total_Structure_Area = mean(Total.Structure.Area.sqft, na.rm = TRUE),
    Avg_Months_Since_Purchase = mean(Months.Since.Purchase, na.rm = TRUE),
    Avg_Aggregate_Sub_Total = mean(Aggregate.Sub.Total, na.rm = TRUE),
    Prop_Has_HOA = sum(Has.HOA == "Yes", na.rm = TRUE) / n(),
    Prop_Has_Garage = sum(Has.Garage == "Yes",na.rm = TRUE) / n(),
    Prop_Returning_Customer = sum(Returning.Customer == "Yes", na.rm = TRUE) / n(),
    .groups = 'drop'
  )

# Summarize by Macro Region
df_by_macro_region <- df_unique_high %>%
  group_by(Macro_Region) %>%
  summarise(
    Sample_Size = n(),
    Avg_Bedrooms = mean(Bedrooms, na.rm = TRUE),
    Avg_Bathrooms = mean(Bathrooms, na.rm = TRUE),
    Avg_Price_Per_sqft = mean(Price.Per.sqft, na.rm = TRUE),
    Avg_Property_Age = mean(Property.Age, na.rm = TRUE),
    Avg_Total_Structure_Area = mean(Total.Structure.Area.sqft, na.rm = TRUE),
    Avg_Months_Since_Purchase = mean(Months.Since.Purchase, na.rm = TRUE),
    Avg_Aggregate_Sub_Total = mean(Aggregate.Sub.Total, na.rm = TRUE),
    Prop_Has_HOA = sum(Has.HOA == "Yes", na.rm = TRUE) / n(),
    Prop_Has_Garage = sum(Has.Garage == "Yes",na.rm = TRUE) / n(),
    Prop_Returning_Customer = sum(Returning.Customer == "Yes", na.rm = TRUE) / n(),
    .groups = 'drop'
  )
```

```
## Visualizations by Region

df_by_macro_region <- df_by_macro_region %>% rename("N.of.Customers" = Sample_Size)
df_by_region <- df_by_region %>% rename("N.of.Customers" = Sample_Size)
df_by_state_filtered <- df_by_state_filtered %>% rename("N.of.Customers" = Sample_Size)

# List of variables to plot
vars_to_plot <- c("N.of.Customers", "Avg_Bedrooms", "Avg_Bathrooms", "Avg_Price_Per_sqft",
                  "Avg_Property_Age", "Avg_Total_Structure_Area", "Avg_Months_Since_Purchase",
                  "Avg_Aggregate_Sub_Total", "Prop_Has_HOA", "Prop_Has_Garage",
"Prop_Returning_Customer")

# Function to create bar plots
create_sorted_bar_plot <- function(df, var, group_var) {
  df <- df %>%
    arrange(!!sym(var)) %>%
    mutate(!!sym(group_var) := factor(!!sym(group_var), levels = unique(!!sym(group_var))))

  p <- ggplot(df, aes_string(x = var, y = group_var, fill = var)) +
    geom_bar(stat = "identity") +
    scale_fill_gradient(low = "lightblue", high = "darkblue") +
    theme_minimal() +
    ylab(group_var) +
    xlab(var) +
    ggtitle(paste("Sorted Bar Plot of", var, "by", group_var))

  return(p)
}

plot_list_by_state <- list()
plot_list_by_region <- list()
plot_list_by_macro_region <- list()

for (var in vars_to_plot) {
  plot_list_by_state[[var]] <- create_sorted_bar_plot(df_by_state_filtered, var, "State")
  plot_list_by_region[[var]] <- create_sorted_bar_plot(df_by_region, var, "Region")
  plot_list_by_macro_region[[var]] <- create_sorted_bar_plot(df_by_macro_region, var,
"Macro_Region")
}

save_plots <- function(plot_list, directory) {
  if (!dir.exists(directory)) {
    dir.create(directory)
  }

  for (var in names(plot_list)) {
    ggsave(paste0(directory, "/", var, ".png"), plot = plot_list[[var]], width = 10, height = 8)
  }
}

save_plots(plot_list_by_state, "State_Plots")
save_plots(plot_list_by_region, "Region_Plots")
```

```
save_plots(plot_list_by_macro_region, "Macro_Region_Plots")


# Export Tables
# Create a new workbook
wb <- createWorkbook()

# Add data frames as sheets
addWorksheet(wb, "Cluster Profile")
writeData(wb, "Cluster Profile", cluster_profile)

addWorksheet(wb, "Cluster Sizes")
writeData(wb, "Cluster Sizes", cluster_sizes)

addWorksheet(wb, "Common States")
writeData(wb, "Common States", common_states)

addWorksheet(wb, "Sales Stats per Cluster")
writeData(wb, "Sales Stats per Cluster", df_sales_stats_perCluster)

addWorksheet(wb, "State Frequency")
writeData(wb, "State Frequency", state_frequency)

addWorksheet(wb, "State Frequency Grouped")
writeData(wb, "State Frequency Grouped", state_frequency_grouped)

addWorksheet(wb, "Top States per Cluster")
writeData(wb, "Top States per Cluster", top_states_per_cluster)

# Save workbook to a file
saveWorkbook(wb, "Cluster_Analysis_Output.xlsx", overwrite = TRUE)

write.csv(df_unique_high_filtered, "Data_wClusters.csv")


# Export Data with Regions

# Create a new workbook
wb <- createWorkbook()

addWorksheet(wb, "By State")
writeData(wb, "By State", df_by_state_filtered)

addWorksheet(wb, "By Macro Region")
writeData(wb, "By Macro Region", df_by_macro_region)

addWorksheet(wb, "By Region")
writeData(wb, "By Region", df_by_region)

# Save workbook to a file
saveWorkbook(wb, "Region_Analysis_Output.xlsx", overwrite = TRUE)
```