Housing Price Prediction with Principal Component Analysis

Authors: Preston O'Connor, Anthony Yasan, Matthew Jacob, Khoa Dao, Nick Wierzbowski

Date: 4/22/2025

Introduction

Our model uses principal component analysis to group houses on a variety of features, including geo-spatial variables and indicators of socioeconomic status of neighborhood. The data set is a subset of an original dataset obtained from the 1990 California census. Grouping housing in this manner may allow us to pinpoint different solutions most suitable for resolving the affordable housing crisis, to which Lento et al. speaks. The packages we used include cluster which is useful for cluster analysis functions, tidyverse for general visualization and data manipulation, factoextra for clustering and PCA visualization, and FactoMineR for PCA implementation. To summarize our findings, the PCA seemed to work fairly well in explaining our data and reducing it's dimensions. Significant differences in variance and other factors between clusters suggest that for different categories or groupings of houses, different strategies are needed.

Data Description

The dataset used for this analysis is titled **California Housing Prices**, originally sourced from the 1990 California census and made publicly available on Kaggle.

Data Structure and Size

The dataset comprises 20,640 observations (rows) and 10 variables (columns), all in numeric form except for the `ocean_proximity` variable which is categorical. Each row represents a block group, which is the smallest geographical unit for which the U.S. Census Bureau publishes sample data.

Variables

Below is a summary of each variable:

- **longitude**: Geographic coordinate, measured in degrees (negative for Western Hemisphere).
- **latitude**: Geographic coordinate, measured in degrees (positive for Northern Hemisphere).
- **housing_median_age**: Median age of houses in the block.
- **total_rooms**: Total number of rooms in all houses within the block.
- **total_bedrooms**: Total number of bedrooms in all houses within the block.
- **population**: Total population of the block.
- **households**: Total number of households in the block.
- **median_income**: Median income of households within the block (scaled in tens of thousands).
- **median_house_value**: Median house value for households within the block (target variable, in USD).
- **ocean_proximity**: Categorical variable indicating the block's proximity to the ocean (e.g., "INLAND", "<1H OCEAN", "NEAR OCEAN").

Data Cleaning

```
# get rid of any non numerical features
data_clean <- data %>%
  select(where(is.numeric))
str(data_clean)
```

```
'data.frame':   20640 obs. of  9 variables:
 $ longitude         : num  -122 -122 -122 -122 -122 ...
 $ latitude          : num  37.9 37.9 37.9 37.9 37.9 ...
 $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 ...
 $ total_rooms       : num  880 7099 1467 1274 1627 ...
 $ total_bedrooms    : num  129 1106 190 235 280 ...
 $ population        : num  322 2401 496 558 565 ...
 $ households        : num  126 1138 177 219 259 ...
 $ median_income     : num  8.33 8.3 7.26 5.64 3.85 ...
 $ median_house_value: num  452600 358500 352100 341300 342200 ...
```

Removing Unknown rows and Outliers

```
total <- sum(is.na(data_clean))
total
```

```
[1] 207
```

```
#cleaning the data points
data_clean <- na.omit(data_clean)

total <- sum(is.na(data_clean))
total
```

```
[1] 0
```

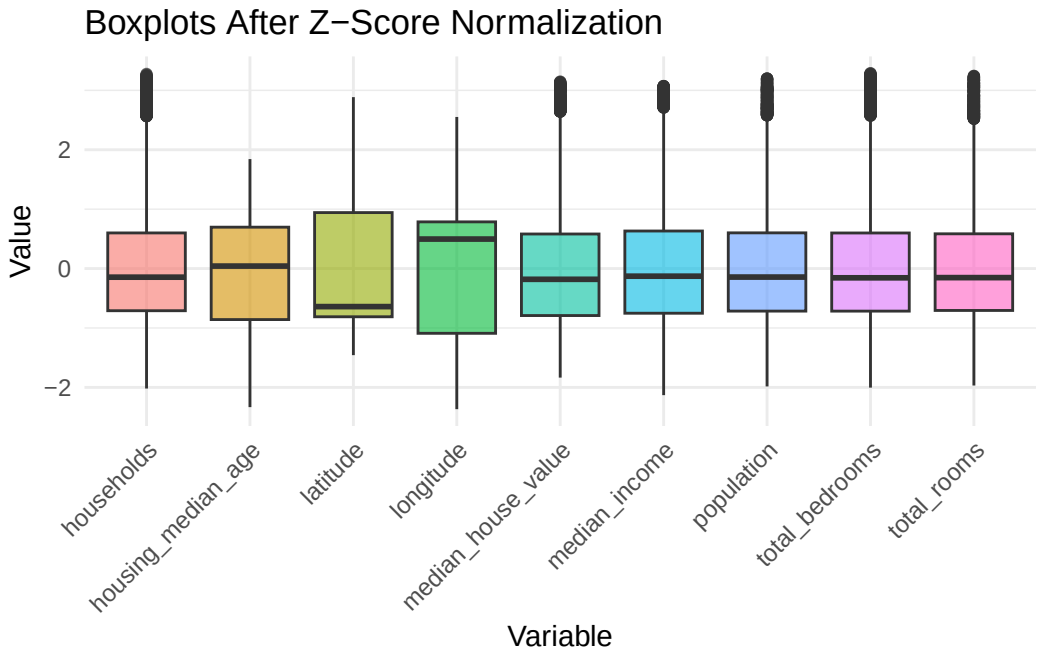
- removed 207 rows of data from the data set

IQR Outlier Removal

```
# Note it is fine to normalize latitude and longitude for out set up
Q1 <- apply(data_clean, 2, quantile, 0.25)
Q3 <- apply(data_clean, 2, quantile, 0.75)
IQR_vals <- Q3 - Q1

in_bounds <- apply(data_clean, 1, function(row) {
  all(row >= (Q1 - 1.5 * IQR_vals) & row <= (Q3 + 1.5 * IQR_vals))
})

data_filtered <- data_clean[in_bounds, ]



# Convert matrix result of scale() to data frame
data_scaled <- as.data.frame(scale(data_filtered))
```

Boxplot

```r
data_scaled %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to =
   ↪  "Value") %>%
  ggplot(aes(x = Variable, y = Value, fill = Variable)) +
  geom_boxplot(alpha = 0.6) +
  theme_minimal() +
  ggtitle("Boxplots After Z-Score Normalization") +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 45, hjust = 1))
```



- Here, We can see there are still some minor outliers outside the normalized data set we are implementing

Analysis

Applying PCA

```r
pca_res<- prcomp(data_scaled, center = TRUE, scale. = TRUE)
pca_res
```

```
Standard deviations (1, .., p=9):
[1] 1.9269684 1.3911402 1.2841223 1.0033831 0.5952655 0.4549082 0.2645666
[8] 0.2059364 0.1486145

Rotation (n x k) = (9 x 9):
                            PC1          PC2          PC3         PC4
longitude            0.05636352 -0.676252111 -0.203330273 -0.06635232
latitude            -0.07052784  0.696667094  0.083677673 -0.07688202
housing_median_age -0.16519696  0.004106531 -0.003681354  0.92399261
total_rooms          0.48747200  0.077725743  0.105785186 -0.03095765
total_bedrooms       0.49663834  0.078406672 -0.090840601  0.09893751
population           0.46384348  0.003525504 -0.146397679  0.09576755
households           0.50134068  0.066146312 -0.066517289  0.13219709
median_income        0.07883035 -0.138241291  0.683055417 -0.19376143
median_house_value   0.08702224 -0.147084547  0.663157027  0.24695382
                            PC5         PC6          PC7         PC8          PC9
longitude           -0.06550947 -0.20073655  0.254600454 -0.61471846 -0.08045854
latitude            -0.06024480 -0.07818647  0.244621407 -0.64990381 -0.06936019
housing_median_age -0.31686222 -0.12822310  0.008031856 -0.04145178  0.01707681
total_rooms         -0.20375929 -0.44394655  0.599786866  0.35950075 -0.12879990
total_bedrooms       0.19892202 -0.28515495 -0.287808125 -0.16113719  0.70673905
population          -0.25903379  0.78927267  0.217874329 -0.07677126  0.10402015
households           0.12229837 -0.07272538 -0.482657427 -0.11998922 -0.67437385
median_income       -0.59426359 -0.02568959 -0.312496789 -0.11370650  0.07857701
median_house_value   0.61268669  0.17276683  0.233698537 -0.09798786 -0.02408003
```

- From the principle components we have 9 standardized features that we extracted from the data
- PC1 is heavily influenced by the total_rooms, total_bedrooms, population, households
- PC2 is mostly driven by spatial features like long and lat
- PC3 relate strongly to median_income ad median_house_value
- PC4 is dominated by the housing_median_age

```r
pca_result <- prcomp(data_scaled, center = TRUE, scale. = TRUE)
pca_df <- as.data.frame(pca_result$x[, 1:4])  # Use first four principal
 ↪  components
summary(pca_result)
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
```

```
Standard deviation     1.9270 1.3911 1.2841 1.0034 0.59527 0.45491 0.26457
Proportion of Variance 0.4126 0.2150 0.1832 0.1119 0.03937 0.02299 0.00778
Cumulative Proportion  0.4126 0.6276 0.8108 0.9227 0.96206 0.98506 0.99283
                           PC8     PC9
Standard deviation     0.20594 0.14861
Proportion of Variance 0.00471 0.00245
Cumulative Proportion  0.99755 1.00000
```

- We chose PC1-PC4 because these capture over 92% of the total variance in the data
- PC1 explains 41% and PC2 adds 21.5% giving a strong reduction in dimensional with minimal information loss
  - by implementing the top 4 PCs ensures most of our data structure is preserved while reducing noise and complexity

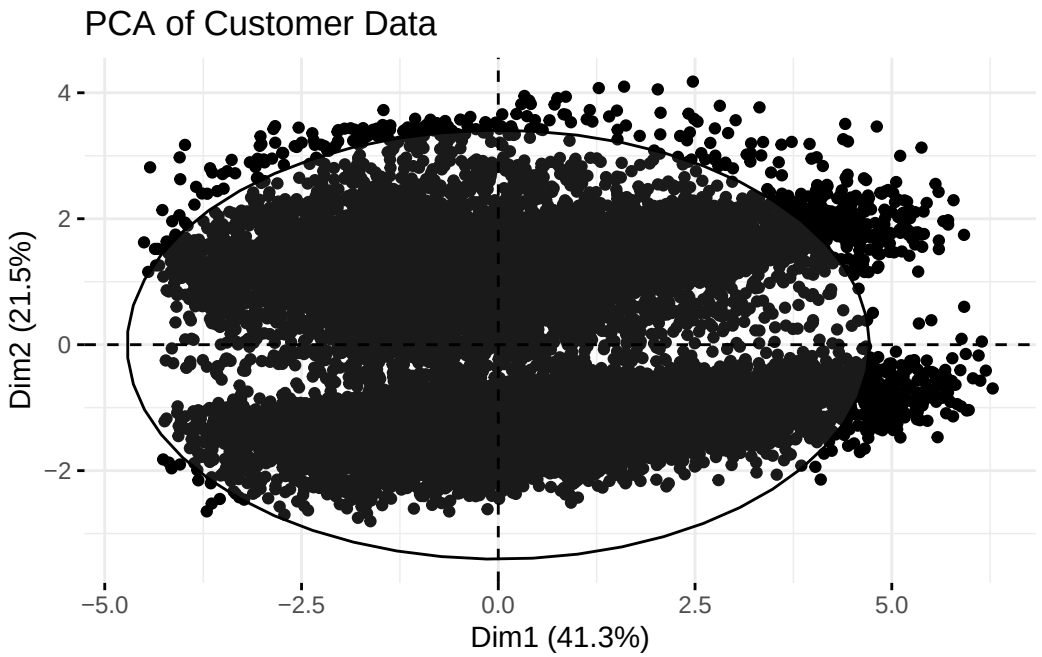
Cumulative Variance

```r
explained_var <- pca_res$sdev^2
prop_var <- explained_var / sum(explained_var)
cum_var <- cumsum(prop_var)
cum_var[1:3]
```

```
[1] 0.4125786 0.6276087 0.8108276
```

- From this, we can see we should keep the first 3 components since they explain a high level of variance (81%). this will allow us to retain the most signal and reduce the noise and dimension

```r
# select PC1 and PC2 for the data
# data_pca <- as.data.frame(pca_result$x[, 1:2])
# data_pca

data_scaled <- scale(data_filtered %>% select(where(is.numeric)))

# PCA
pca_res <- prcomp(data_scaled)
pca_df <- as.data.frame(pca_res$x[, 1:3])

# Plot first 2 principal components
fviz_pca_ind(pca_res, label = "none", addEllipses = TRUE, title = "PCA of
 ↪  Customer Data")
```

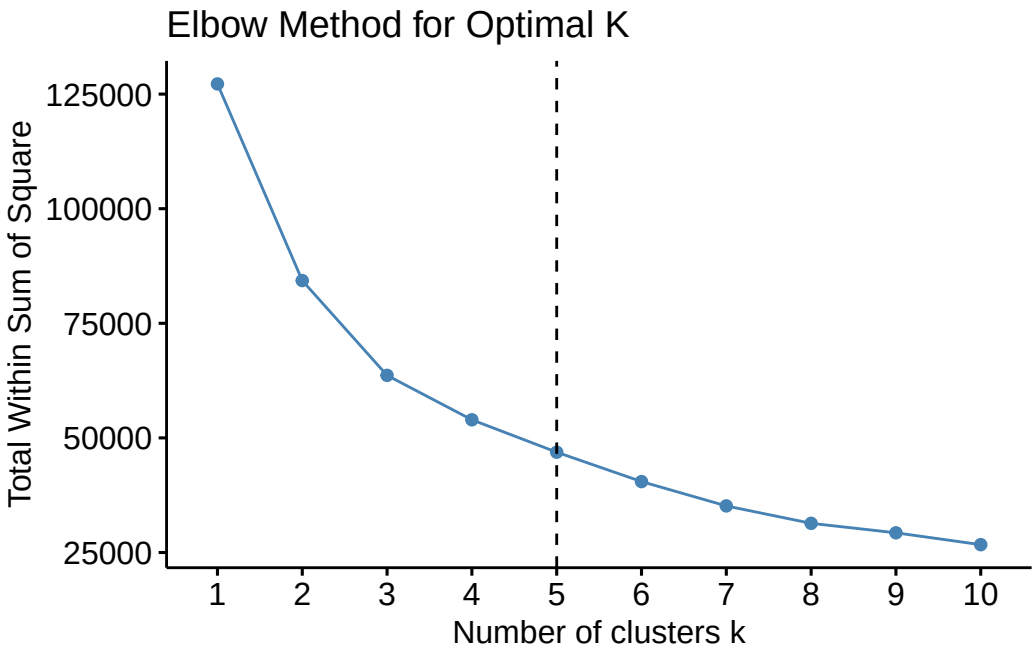


- Here we can see that Dim1 and Dim2 do a good job and together explain about 62.8% of the total variance, which is strong. The data is fairly spread out and has two visible groupings from what we can tell in the graph.This suggest that our structure is suitable for a clustering implementation. This justifies our reasoning to reduce the data to the first 4 components.

Determine Optimal K

```
fviz_nbclust(pca_df, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2) +
  labs(title = "Elbow Method for Optimal K")
```

```
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
```



- Here we see that the k=5 is the reasonable value as this is where the curve is starting to flatten out

Run K-means Clustering (Wondering how the test works here)

```
set.seed(123)

km <- kmeans(pca_df, centers = 5, nstart = 50)
```

```
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
```

```
sil <- silhouette(km$cluster, dist(pca_df))
avg_sil <- mean(sil[, 3])


variance_explained <- 1 - (km$tot.withinss / km$totss)


cat("Silhouette score (k = 5):", round(avg_sil, 3), "\n")
```

```
Silhouette score (k = 5): 0.302
```

```
cat("Variance explained (k = 5):", round(variance_explained, 3), "\n")
```

```
Variance explained (k = 5): 0.639
```

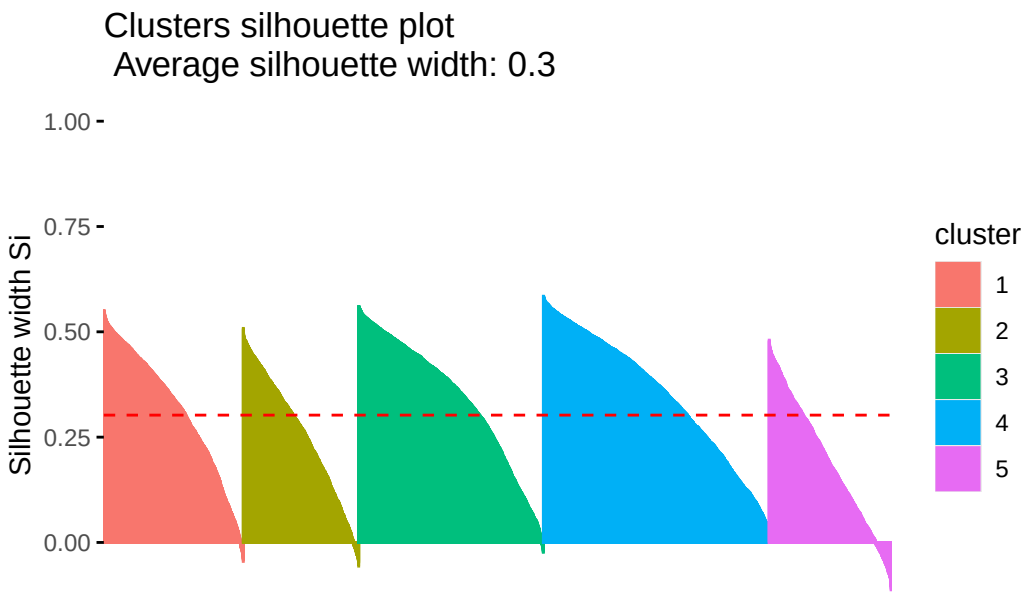
- For the Silhouette score of 0.302 we are in the somewhat moderate range. Many points may be reasonable close to their cluster centroids; however, there is bound to be overlap between the clusters.
- We captured a reasonably strong clustering that pertains to a meaningful structure

Model Evaluation and Prediction

Evaluation of silhouette

```
fviz_silhouette(sil)
```

```
  cluster size ave.sil.width
1       1 3088          0.31
2       2 2556          0.25
3       3 4089          0.34
4       4 5002          0.35
5       5 2699          0.19
```



- The average silhouette width is 0.3 showcasing moderate cluster separation
- Cluster 1 and 4 show somewhat high silhouette widths, suggesting better internal cohesion.
- Clusters 2 and 5 have lower and more variable silhouettes scores, implying overlap and poor separations
- Overall, while some of the clusters are indeed well-formed, others may benefit from re-evaluation or a different choice of k

Variance and Silhouette

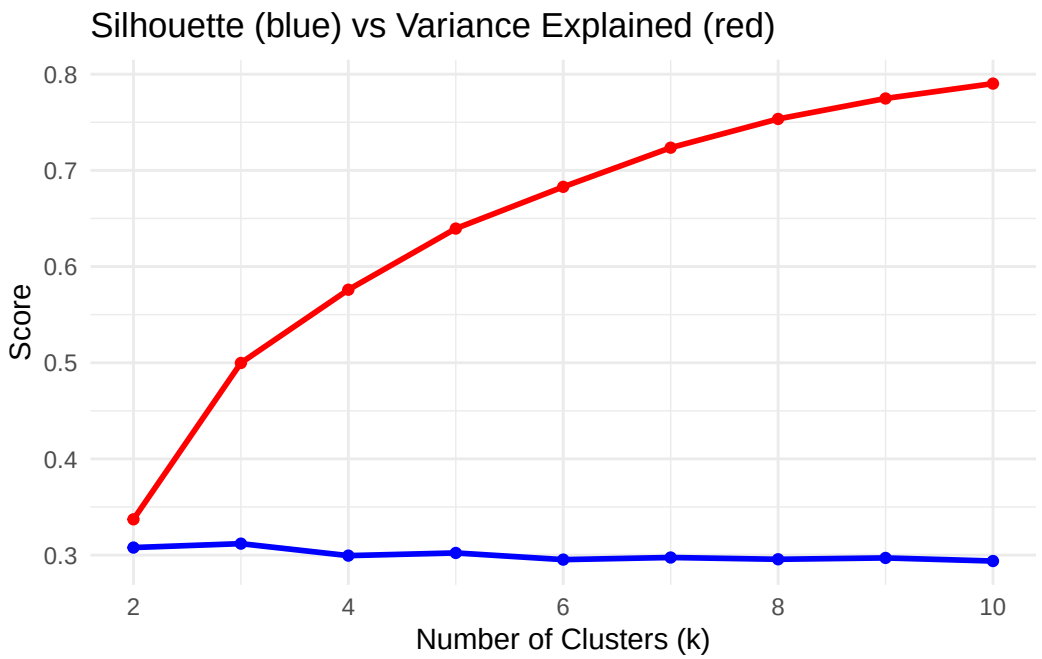
```
set.seed(123)

k_values <- 2:10
silhouette_scores <- numeric(length(k_values))
variance_scores <- numeric(length(k_values))

for (i in seq_along(k_values)) {
  km <- kmeans(pca_df, centers = k_values[i], nstart = 50)
  sil <- silhouette(km$cluster, dist(pca_df))

  silhouette_scores[i] <- mean(sil[, 3])
```

```
  variance_scores[i] <- 1 - (km$tot.withinss / km$totss)
}
```

```
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
```

```
results <- data.frame(
  k = k_values,
  silhouette = silhouette_scores,
  variance = variance_scores
)


ggplot(results, aes(x = k)) +
  geom_line(aes(y = silhouette), color = "blue", size = 1) +
  geom_point(aes(y = silhouette), color = "blue") +
  geom_line(aes(y = variance), color = "red", size = 1) +
  geom_point(aes(y = variance), color = "red") +
  labs(title = "Silhouette (blue) vs Variance Explained (red)",
       x = "Number of Clusters (k)",
       y = "Score") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



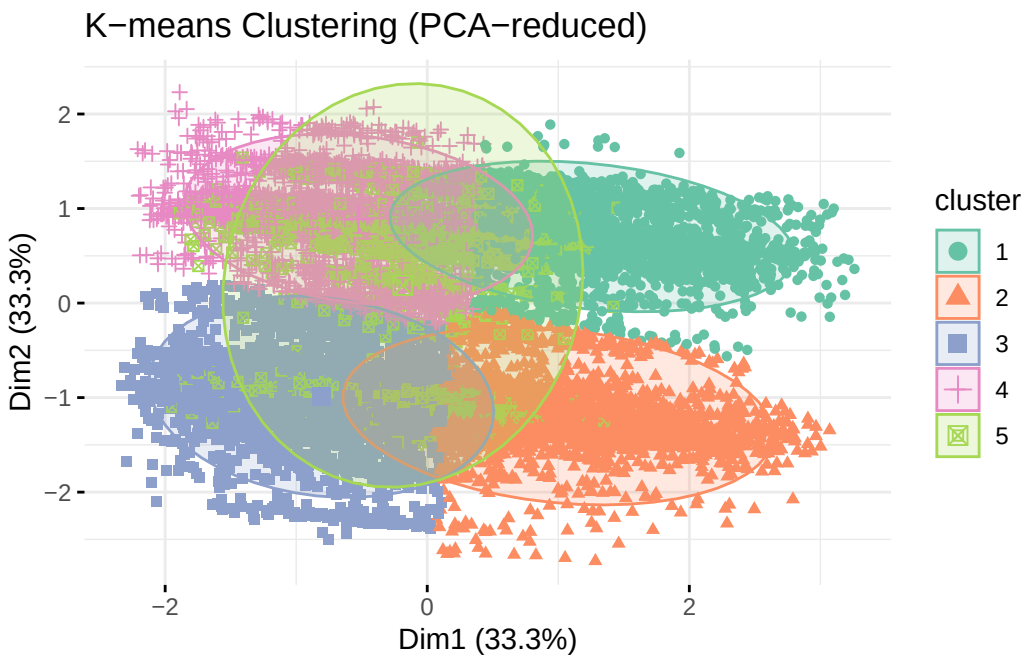
- This shows a trade off between silhouette and variance as the number of our clusters increase. While variance explained continues to rise with more clusters, the silhouette score peaks around k=2, 3, then declines indicating that the cluster quality drops beyond that point. This suggest an optimal balance

PCA Scatter Plot

```
set.seed(123)
km_result <- kmeans(pca_df, centers = 5, nstart = 25)
```

```
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
Warning: Quick-TRANSfer stage steps exceeded maximum (= 871700)
```

```
# Then plot
fviz_cluster(km_result, data = pca_df,
             geom = "point",           # points only, no text
             ellipse.type = "norm",    # shaded cluster region
             palette = "Set2",
             show.clust.cent = TRUE,   # show centroids
             ggtheme = theme_minimal(),
             main = "K-means Clustering (PCA-reduced)")
```



- This k-means cluster plot reduced by PCA with k = 5 indicates distinct but overlapping clusters, representing moderate suggesting moderate separation in the data. The two principal components (Dim1 and Dim2) explain around 66.6% of the variance, allowing one to meaningfully 2d-visualize the structure. While clusters 1 and 2 appear quite compact, other such cluster 5 overlap considerably more, indicating potential ambiguity in those group boundaries

Conclusion and Summary

In this project, we applied Principal Component Analysis (PCA) and K-means clustering to look at housing data from the 1990 California census into groups based upon various features including geographical location, socioeconomic indicators, and housing characteristics. After preprocessing the data by removing outliers and handling NA values, we successfully improved the quality of the dataset. By then extracting the first four principal components, we were able to capture over 92% of the total variance in the data, simplifying the complex dataset. Using our processed data, we implemented K-means clustering, determining that five clusters best fit the data via the elbow method. The clustering results showed moderate silhouette scores, indicating some grouping of the data but also areas of overlap between clusters. The clusters revealed areas for potential improvement in cluster separation, especially in clusters 2 and 5 where the silhouette scores were lower. The findings suggest that while PCA and K-means are somewhat useful tools for grouping houses based on key factors, the clustering model would benefit from further refinement. Additional consideration may be needed in choosing the

optimal number of clusters. Further developments could involve other clustering techniques or could incorporate more relevant variables in order to improve model capability. Nonetheless, this analysis offers some insight into the segmentation of California housing, insight that could lead to more targeted strategies for addressing the California housing crisis.

References

- Nugent, C. (2017, November 24). California housing prices. Kaggle. https://www.kaggle.com/datasets/ca housing-prices?select=housing.csv
- Kassambara. (2018, October 21). K-means clustering in R: Algorithm and practical examples. Datanovia. https://www.datanovia.com/en/lessons/k-means-clustering-in-r-algorith-and-practical-examples/
- Kryńska, K. (n.d.). Using K-means and PAM clustering for Customer Segmentation. RPubs. https://rpubs.com/kkrynska/USL_k-means »»»> 02ae82986c97859750f8e7d4ad40bae2c4c0faf6
- Lento, Rochelle E., Shaun Donovan, Sheila Crowley, Rebecca L. Peace, Mark H. Shelburne, Jeanne Peterson, Janet Kennedy, et al. "The Future of Affordable Housing." Journal of Affordable Housing & Community Development Law 20, no. 2 (2011): 215–50. http://www.jstor.org/stable/41429170.