Project Multivariate data Analysis

Sara Vasques Rita Juan Pablo Martinez Aldana

2025-11-21

Contents

1 Abstract

Air pollution in urban environments is a significant public health concern. Some evidence indicates that air pollution exposure has way more consecuences that the onces that we thought previously. In terms of adverse health impacts like the reduction in life expectancy, and hospital admissions, birth outcomes, and asthma. Nevertheless, these effects exist in both economically developing and developed countries [1].

Classical PCA can be expressed as a projection-based approach, finding the low-dimensional space that best represents a cloud of high-dimensional points. With this we can do a dimensional reduction using our principal components (PC). [2]

Due to the reasons mention before, This project aims to analyze a dataset of 41 US cities to identify underlying patterns and relationships between pollution levels, climatic conditions, and demographic or industrial factors.

The study utilizes multivariate techniques, primarily **Principal Component Analysis (PCA)** and **Cluster Analysis**, to reduce the complexity of the data and group cities with similar profiles. The objective is to translate complex statistical relationships into clear, interpretable insights.

The dataset contains the following variables: **so2**: Sulfur dioxide content of air in micrograms per cubic meter **temp**: Average annual temperature in Fahrenheit **manuf**: Number of manufacturing enterprises employing 20 or more workers **pop**: Population size (1970 census) in thousands **wind**: Average wind speed in miles per hour **precip**: Average annual precipitation in inches **days**: Average number of days with precipitation per year

2 Preliminary analysis of the data

Before applying multivariate methods, a preliminary analysis is essential to understand the individual variables and their pairwise relationships.

```
##       city so2 temp manuf pop wind precip days
## 1  Phoenix  10 70.3   213 582  6.0   7.05   36
## 2 Little R  13 61.0    91 132  8.2  48.52  100
## 3 San Fran  12 56.7   453 716  8.7  20.66   67
## 4   Denver  17 51.9   454 515  9.0  12.95   86
## 5 Hartford  56 49.1   412 158  9.0  43.37  127
## 6 Wilmingt  36 54.0    80  80  9.0  40.25  114
```

```
## [1] 41  8
```

```
## 'data.frame':    41 obs. of  8 variables:
##  $ city  : chr  "Phoenix" "Little R" "San Fran" "Denver" ...
##  $ so2   : int  10 13 12 17 56 36 29 14 10 24 ...
##  $ temp  : num  70.3 61 56.7 51.9 49.1 54 57.3 68.4 75.5 61.5 ...
##  $ manuf : int  213 91 453 454 412 80 434 136 207 368 ...
##  $ pop   : int  582 132 716 515 158 80 757 529 335 497 ...
##  $ wind  : num  6 8.2 8.7 9 9 9 9.3 8.8 9 9.1 ...
##  $ precip: num  7.05 48.52 20.66 12.95 43.37 ...
##  $ days  : int  36 100 67 86 127 114 111 116 128 115 ...
```

2.1 Numerical Data

```
## 
## 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
```

```
##   so2 temp manuf pop wind precip days
## 1  10 70.3   213 582  6.0   7.05   36
## 2  13 61.0    91 132  8.2  48.52  100
## 3  12 56.7   453 716  8.7  20.66   67
## 4  17 51.9   454 515  9.0  12.95   86
## 5  56 49.1   412 158  9.0  43.37  127
## 6  36 54.0    80  80  9.0  40.25  114
```

2.2 Descriptive Statistics

```
##       so2              temp           manuf             pop        
##  Min.   :  8.00   Min.   :43.50   Min.   :  35.0   Min.   :  71.0  
##  1st Qu.: 13.00   1st Qu.:50.60   1st Qu.: 181.0   1st Qu.: 299.0  
##  Median : 26.00   Median :54.60   Median : 347.0   Median : 515.0  
##  Mean   : 30.05   Mean   :55.76   Mean   : 463.1   Mean   : 608.6  
##  3rd Qu.: 35.00   3rd Qu.:59.30   3rd Qu.: 462.0   3rd Qu.: 717.0  
##  Max.   :110.00   Max.   :75.50   Max.   :3344.0   Max.   :3369.0  
##       wind           precip           days      
##  Min.   : 6.000   Min.   : 7.05   Min.   : 36.0  
##  1st Qu.: 8.700   1st Qu.:30.96   1st Qu.:103.0  
##  Median : 9.300   Median :38.74   Median :115.0  
##  Mean   : 9.444   Mean   :36.77   Mean   :113.9  
##  3rd Qu.:10.600   3rd Qu.:43.11   3rd Qu.:128.0  
##  Max.   :12.700   Max.   :59.80   Max.   :166.0
```

We identify that `manuf` and `pop` have a mean is significantly larger than the median, for `manuf` was found 463 vs 347 respectivetyly and for `pop` was 608 vs 515 respectivetily. Furthermore, the maximum values (3344, 3369) are in order of magnitude greater than the 75th percentile (462, 717) for both variables.

With this we could infer that the dataset does not contain 41 similar cities. It contains a majority of "typical" cities and a few massive outliers, metropolitan big cities. In other to avoid that our analysis was dominated by these few outliers ew decide to use a corretation analysis to normalize our data.

Also we notice that the data, is in different units, and every unit meassure different things which make very important the use of the correlational matriz to perform the analysis.

```r
# Calculate standard deviation for each variable.
library(dplyr)
data5_variables %>% summarise_if(is.numeric, sd)
```

```
##        so2     temp    manuf     pop     wind   precip     days
## 1 23.47227 7.227716 563.4739 579.113 1.428644 11.77155 26.50642
```

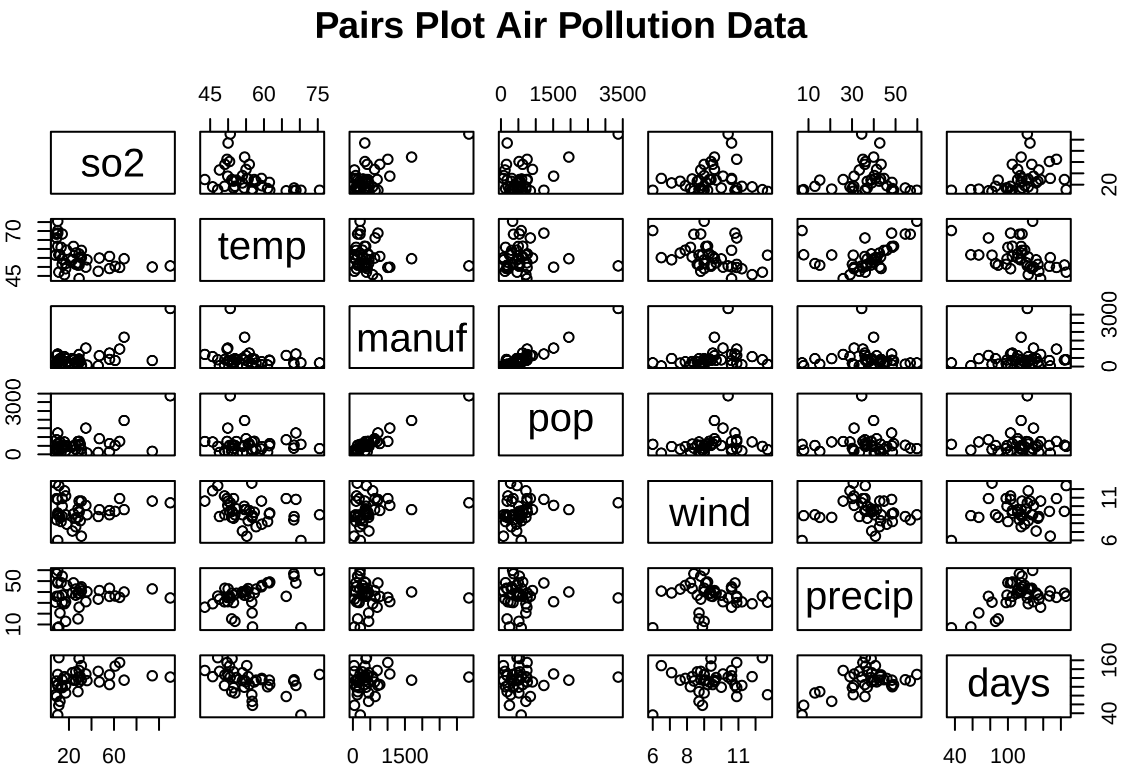
The standard deviations for `pop` and `manuf` are orders of magnitude larger than for other variables, confirming their high variance.

For example, the sd of `pop` is 579.733 and the sd of `wind` is 1.4 in this case if we run a PCA with this data, it will determine that all the variation is happening around `pop` and not `wind`. This is why we need to perform a correlation analysis so all our data, can have a standard deviation of 1 and a mean of 0. So when we run the PCA we can find the direction that will maximize the variance, based on the correlation between the variables.

For the reasons mention above we will ues the correlation matrix in order to perform our PCA.

2.3 Correlation Analysis

```r
# Visualize pairwise relationships between all variables.
pairs(data5_variables, main="Pairs Plot Air Pollution Data")
```



```r
# Calculate and display the correlation matrix.
cor(data5_variables)
```

```
##                so2        temp       manuf         pop        wind      precip
## so2     1.00000000 -0.43360020  0.64476873  0.49377958  0.09469045  0.05429434
## temp   -0.43360020  1.00000000 -0.19004216 -0.06267813 -0.34973963  0.38625342
## manuf   0.64476873 -0.19004216  1.00000000  0.95526935  0.23794683 -0.03241688
## pop     0.49377958 -0.06267813  0.95526935  1.00000000  0.21264375 -0.02611873
## wind    0.09469045 -0.34973963  0.23794683  0.21264375  1.00000000 -0.01299438
## precip  0.05429434  0.38625342 -0.03241688 -0.02611873 -0.01299438  1.00000000
## days    0.36956363 -0.43024212  0.13182930  0.04208319  0.16410559  0.49609671
##               days
## so2     0.36956363
## temp   -0.43024212
## manuf   0.13182930
## pop     0.04208319
## wind    0.16410559
## precip  0.49609671
## days    1.00000000
```

By looking at the plot and the correlation values between the variables we can observe that the population size and the number of manufacturing enterprises employing 20 or more workers is strongly correlated (0.955). We can see that the most populated cities in this study (looking at the initial data, data5), employ more people. Then, we can observe that the number of manufacturing enterprises employing workers and the sulfur dioxide content of air (mg/m3) are correlated as well (0.645), which means that the manufacturing enterprises that employ 20 or more workers end up influencing so2 levels, which makes sense. Population and so2 levels are correlated as well (0.494), and precipitation and number of days too (0.496). Even though with a lower value, precipitation and temperature variables are correlated (0.386).

The correlation matrix reveals several key relationships: - A very strong positive correlation (0.96) between `manuf` and `pop`. - A moderate positive correlation (0.65) between `manuf` and `so2`. This suggests that industrial and population factors are closely linked and are associated with higher SO2 pollution.

3 Principal Component Analysis (PCA)

PCA will be used to reduce the dimensionality of the dataset. By creating a smaller set of uncorrelated components, we can simplify the data while retaining most of the original variance. We will use the correlation matrix (`scale = TRUE`) to account for the different units and scales of the original variables.

```r
# Perform PCA.
pca_results <- prcomp(data5_variables, scale = TRUE)
summary(pca_results)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.6517 1.2298 1.1811 0.9445 0.58888 0.31668 0.15973
## Proportion of Variance 0.3897 0.2160 0.1993 0.1274 0.04954 0.01433 0.00364
## Cumulative Proportion  0.3897 0.6058 0.8051 0.9325 0.98203 0.99636 1.00000
```

3.1 Justification for Number of Components

We use two criteria to select the number of components to retain.

3.1.1 Kaiser's Criterion

```r
eigenvalues <- pca_results$sdev^2
print(eigenvalues)
```

```
## [1] 2.72811968 1.51233485 1.39497299 0.89199129 0.34677866 0.10028759 0.02551493
```

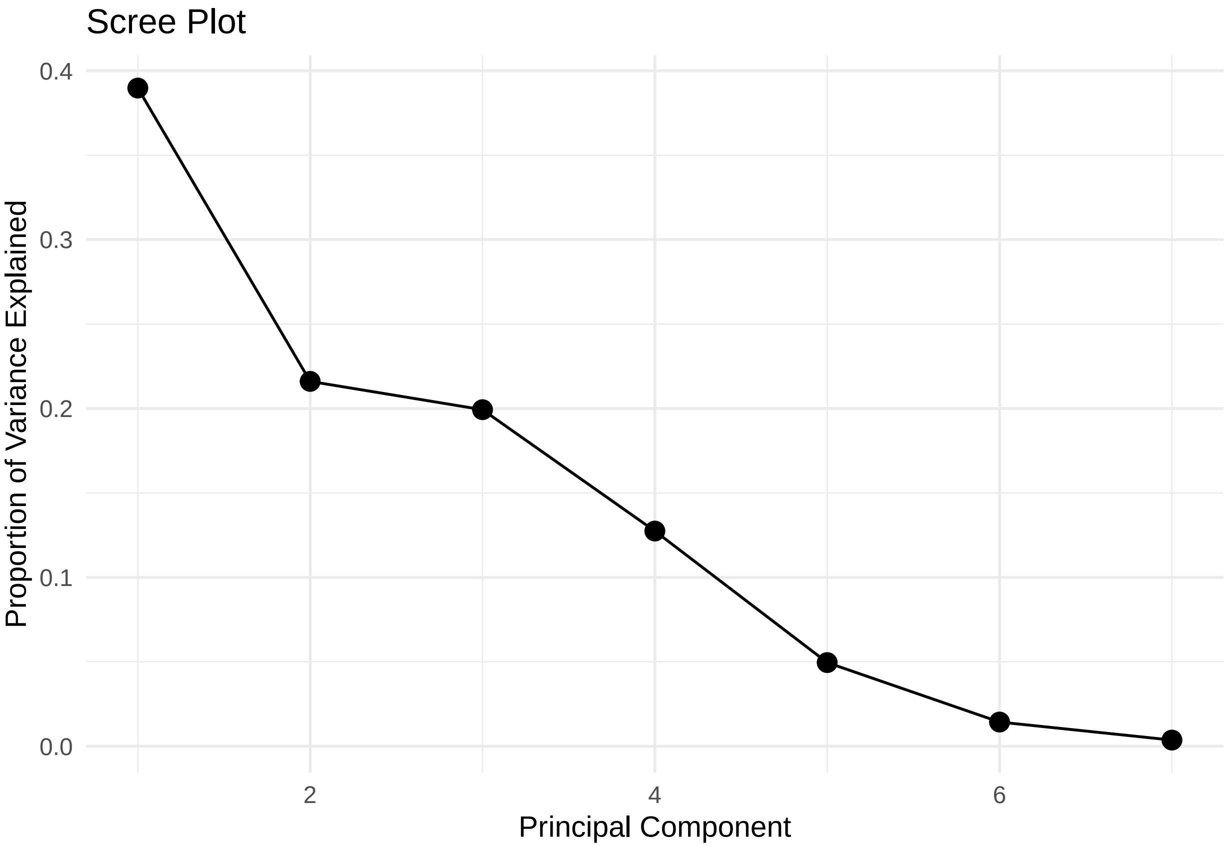
```r
cat("\nNumber of eigenvalues > 1:", sum(eigenvalues > 1), "\n")
```

```
##
## Number of eigenvalues > 1: 3
```

The first two components have eigenvalues greater than 1, suggesting they are significant.

3.1.2 Scree Plot

```r
library(ggplot2)
scree_data <- data.frame(
  component = 1:length(eigenvalues),
  variance_explained = eigenvalues / sum(eigenvalues)
)
ggplot(scree_data, aes(x = component, y = variance_explained)) +
  geom_line() + geom_point(size=3) +
  labs(title = "Scree Plot", x = "Principal Component", y = "Proportion of Variance Explained"
  theme_minimal()
```



The scree plot shows a distinct "elbow" after the second component. Both criteria indicate that **2 principal components** are sufficient for our analysis, capturing ~65% of the total variance.

3.2 Component Interpretation (Loadings)

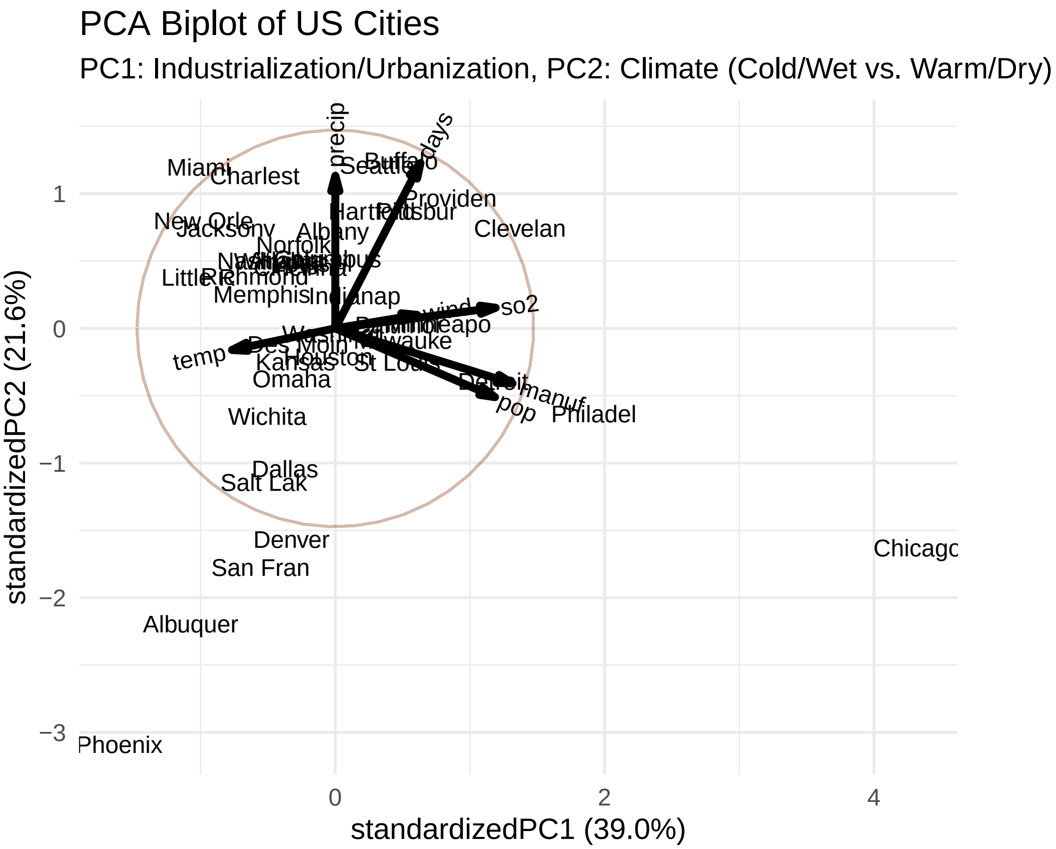
```r
# The loadings show how original variables contribute to each PC.
print(pca_results$rotation)
```

```
##                PC1         PC2       PC3         PC4        PC5         PC6
## so2      0.4896988171  0.08457563  0.0143502 -0.40421007  0.7303942 -0.18334573
## temp    -0.3153706901 -0.08863789  0.6771362  0.18522794  0.1624652 -0.61066107
## manuf    0.5411687028 -0.22588109  0.2671591  0.02627237 -0.1641011  0.04273352
## pop      0.4875881115 -0.28200380  0.3448380  0.11340377 -0.3491048  0.08786327
## wind    0.2498749284  0.05547149 -0.3112655  0.86190131  0.2682549 -0.15005378
## precip  0.0001873122  0.62587937  0.4920363  0.18393719  0.1605988  0.55357384
## days    0.2601790729  0.67796741 -0.1095789 -0.10976070 -0.4399698 -0.50494668
##                  PC7
## so2    -0.149529278
## temp    0.023664113
## manuf   0.745180920
## pop    -0.649125507
## wind   -0.015765377
## precip  0.010315309
## days   -0.008217393
```

- **Principal Component 1 (PC1):** High positive loadings for `manuf`, `pop`, and `so2`. This is an **"Industrialization and Urbanization"** axis.
- **Principal Component 2 (PC2):** High positive loadings for `precip` and `days` and a high negative loading for `temp`. This is a **"Climate"** axis, separating cold, wet cities from warm, dry ones.

3.3 Visualization

```r
library(ggbiplot)
ggbiplot(pca_results, labels = data5$city, ellipse = TRUE, circle = TRUE) +
  theme_minimal() +
  labs(title = "PCA Biplot of US Cities",
       subtitle = "PC1: Industrialization/Urbanization, PC2: Climate (Cold/Wet vs. Warm/Dry)")
```



The biplot provides a visual confirmation of our interpretations and shows how cities relate to these new axes.

So, if we did a graphic for the first 3 PCs:

#############________________________ review the 3d in the pfd format.

```
library(plotly)

# Supondo que você tenha feito o PCA:
data5_2 <- prcomp(data5_variables, scale. = TRUE)

pca_scores <- as.data.frame(data5_2$x)

var_explained <- (data5$sdev)^2 / sum((data5$sdev)^2)

pc1_label <- paste0("PC1 (", round(var_explained[1] * 100, 1), "%)")
pc2_label <- paste0("PC2 (", round(var_explained[2] * 100, 1), "%)")
pc3_label <- paste0("PC3 (", round(var_explained[3] * 100, 1), "%)")

plot_ly(pca_scores,
        x = ~PC1,
        y = ~PC2,
        z = ~PC3,
        type = 'scatter3d',
        mode = 'markers',
        text = rownames(pca_scores),
        marker = list(size = 4)
)%>%
  layout(
    title = "3D Graphic with the first 3 PCs",
    scene = list(
      xaxis = list(title = pc1_label),
      yaxis = list(title = pc2_label),
      zaxis = list(title = pc3_label)
    )
  )
```

4 Cluster Analysis

Now we will group similar cities using hierarchical clustering. We will perform this on the PCA scores, not the original data, to cluster on the most important, de-noised patterns.

```
# GOAL: Perform hierarchical clustering on the first two PCA scores.
# 1. Extract the first two columns from pca_results$x.
# 2. Calculate the distance matrix using dist().
# 3. Perform clustering using hclust().
# 4. Plot the resulting dendrogram.
#
# Placeholder:
# pca_scores <- as.data.frame(pca_results$x[, 1:2])
# distance_matrix <- dist(pca_scores)
# hclust_results <- hclust(distance_matrix, method = "complete")
# plot(hclust_results, labels = data5$city, main = "Hierarchical Clustering Dendrogram")
```

4.1 Determining the Number of Clusters & Characterization

Based on the dendrogram, you must decide how many clusters to form and justify it. Then, you will characterize them.

```
# GOAL: Cut the tree and analyze the resulting clusters.
# 1. Choose a number of clusters (e.g., 3, 4) based on the dendrogram.
# 2. Use cutree() to get the cluster assignments for each city.
# 3. Add the cluster assignments as a new column to the `data5` dataframe.
# 4. Use group_by() and summarise() from dplyr to calculate the mean of the original variables
# 5. Interpret the `cluster_means` table to describe the profile of each cluster.
#
# Placeholder:
# num_clusters <- 4
# city_clusters <- cutree(hclust_results, k = num_clusters)
# data5_clustered <- cbind(data5, cluster = city_clusters)
# cluster_means <- data5_clustered %>%
#   group_by(cluster) %>%
#   summarise_at(vars(so2:days), mean)
# print(cluster_means)
```

5 Spatial Analysis & Comparison

This section elevates the project by mapping the results, as inspired by the provided literature. We will investigate if the clusters we identified have a geographic pattern.

```
# GOAL: Prepare for mapping.
# You will need a way to get latitude and longitude for the US cities in your dataset.
# The `ggmap` or `tidygeocoder` packages in R are excellent for this.
# You will need to geocode the city names to get their coordinates.
#
# Placeholder:
# library(tidygeocoder)
# lat_longs <- geocode(data5_clustered, city = city)
```

```
# GOAL: Create a map of the US with cities colored by cluster.
# Use a mapping library like `leaflet` or `ggplot2` with `maps`.
#
# Placeholder using ggplot2:
# library(ggplot2)
# library(maps)
# us_map <- map_data("state")
# ggplot() +
#   geom_polygon(data = us_map, aes(x = long, y = lat, group = group), fill = "grey80", color
#   geom_point(data = lat_longs, aes(x = longitude, y = latitude, color = as.factor(cluster)),
#   coord_map("albers", lat0 = 39, lat1 = 45) +
#   theme_minimal() +
#   labs(title = "Geographic Distribution of City Clusters", color = "Cluster")
```

5.1 Discussion of Spatial Patterns

6 Conclusion

- What were the main patterns you found in the data? (e.g., “Our analysis identified two primary axes of variation among US cities: one related to industrialization and a second related to climate.”)
- What types of city clusters did you identify? (e.g., “We found four distinct city profiles: …”)
- Did these clusters show a geographic pattern?
- What are the practical implications of these findings for a city planner or environmental agency?