Atmospheric Gas Concentration Analysis – Mauna Loa Observatory

Introduction

This study examines a dataset of average monthly concentrations of CO2, CO, methane, nitrous oxide, and CFC-11 in the atmosphere, collected at the Mauna Loa Volcano Observatory from 2000 to 2019. The variables represent CO_2 measurements in parts per million (ppm), CO, Methane (CH_4), and Nitrous Oxide (N_2O) measurements in parts per billion (ppb), and CFC-11 measurements in parts per trillion (ppt). The data contains about 18% induced missing values. The goal of this analysis is to explore the dataset using unsupervised learning techniques, including dimensionality reduction and cluster analysis.

1. Exploratory Data Analysis

Based on an initial study of the dataset, it is observed that the dataset contains 186 observations of 6 variables. These variables and their types are *Date, CO, CO2, Methane, NitrousOx,* and *CFC11*. The "*Date"* variable is of the character (chr) type (it was later transformed into numeric Date type for time series observations), while all the other variables are of the numeric type, being continuous numeric measurements in ppm/ppb of atmospheric gas concentrations.

Boxplots were generated to observe outliers.

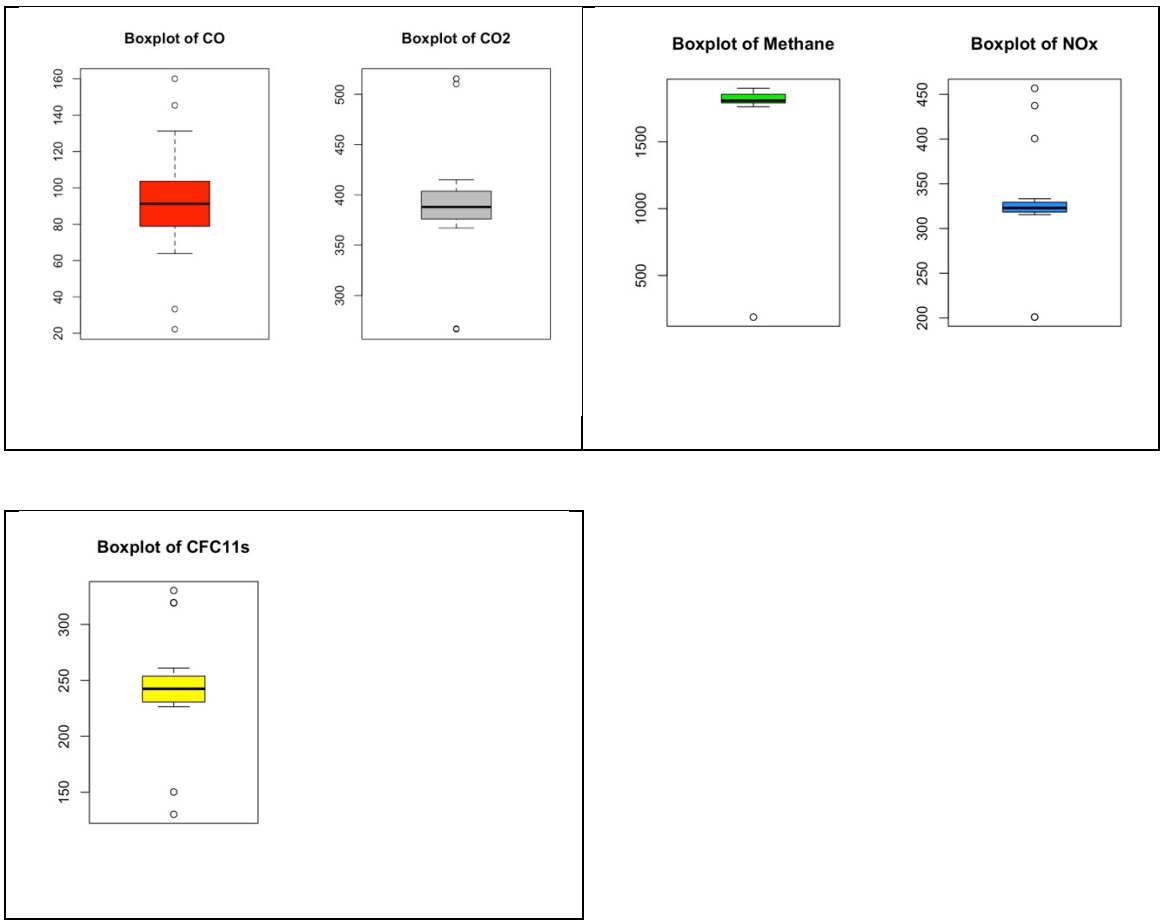


Fig 1.1 - Boxplots of variables

From these visualisations, we can see that every variable has outliers. Methane seems to have a lone low outlier – this could be a data collection error or an anomalous event. For the other gases, we see outliers in both extremes, indicating variable emission rates at the observatory site.

Next, a histogram matrix and a scatterplot matrix were generated to study variable distribution, and discern any notable patterns.

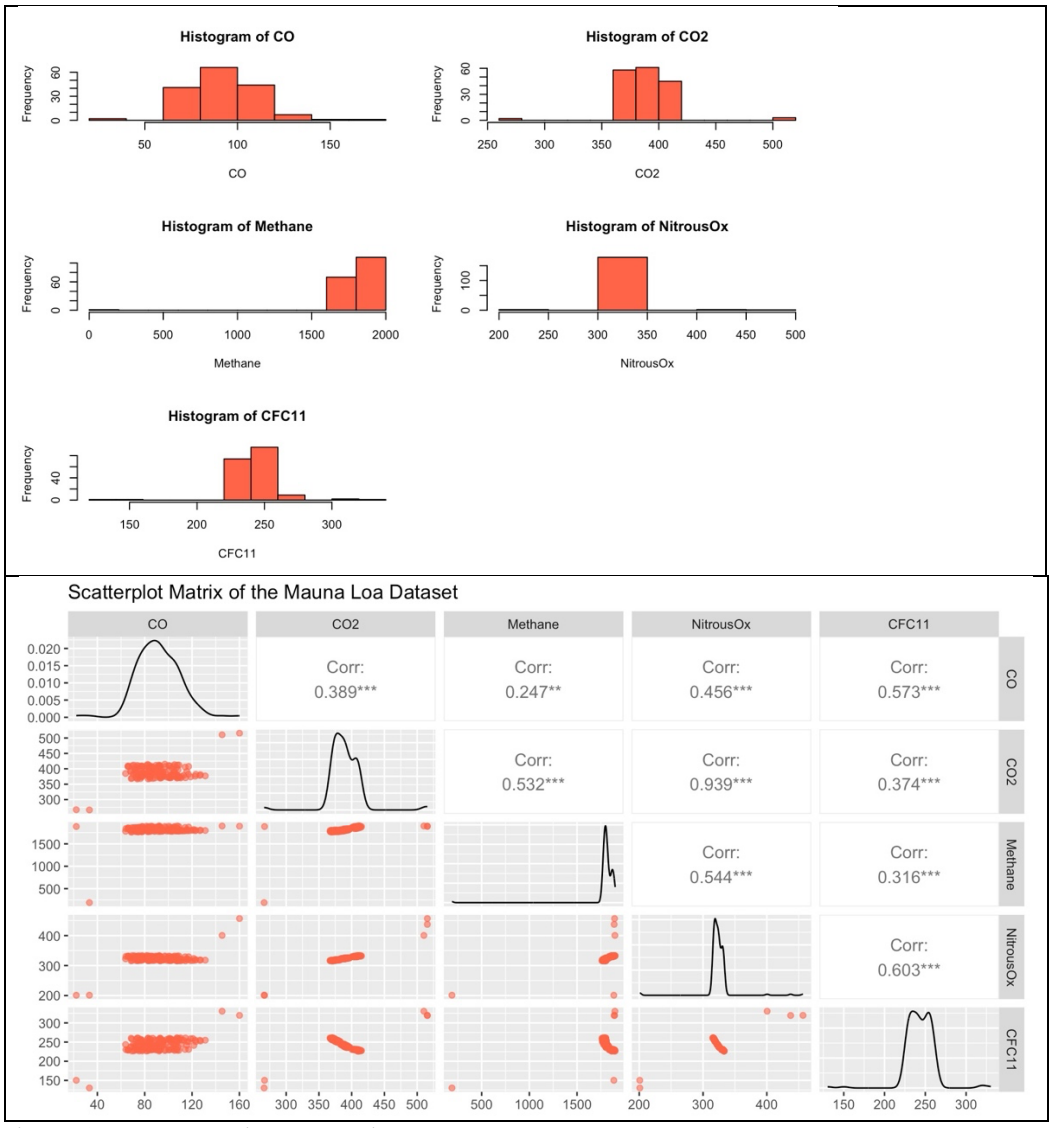


Fig 1.2 – Scatterplot and Histogram Matrices

From these graphs, Methane has a highly skewed distribution, while the other variables have a largely unimodal distribution. It is also observed that many variables have a moderate to high level of correlation – this indicates redundancy and tells us that we can apply dimension reduction techniques to simplify the analysis of this dataset. From the scatterplot matrix, we can also see that the data can roughly be divided into largely centrally located clusters. To round off the EDA, a correlation heatmap was generated to confirm our findings with regards to the interrelations between variables.

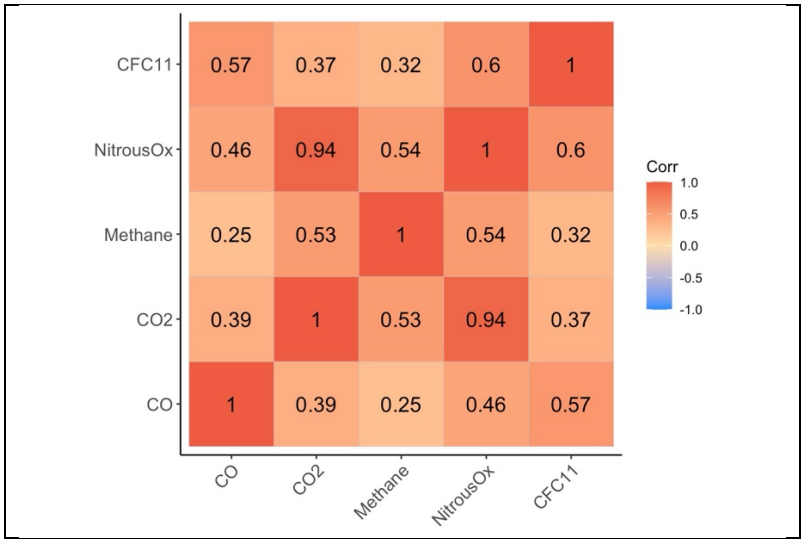
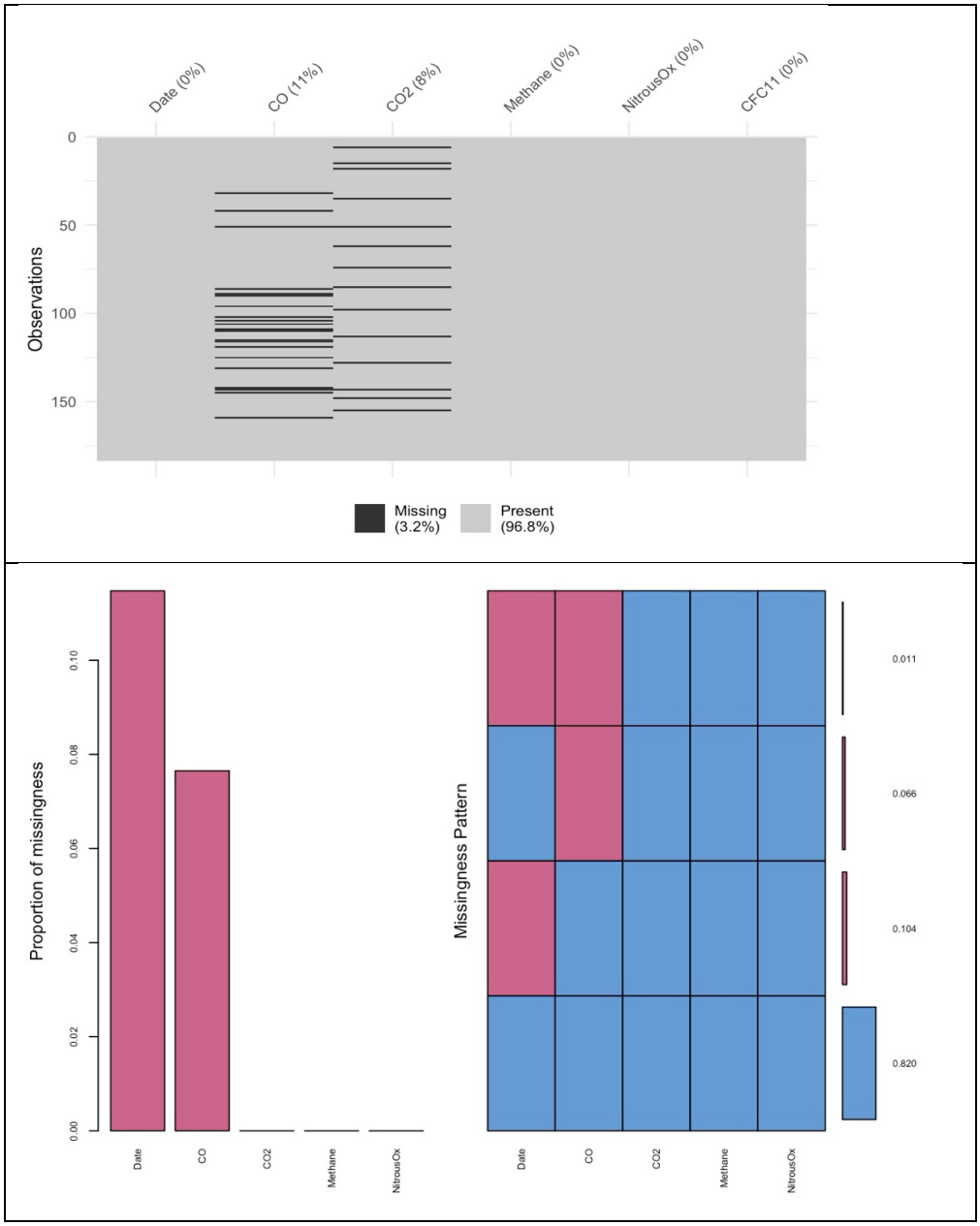


Fig 1.3 – Correlation Heatmap

Missing variables were studied next and dealt with based on the observations made by looking at missingness visualisations.



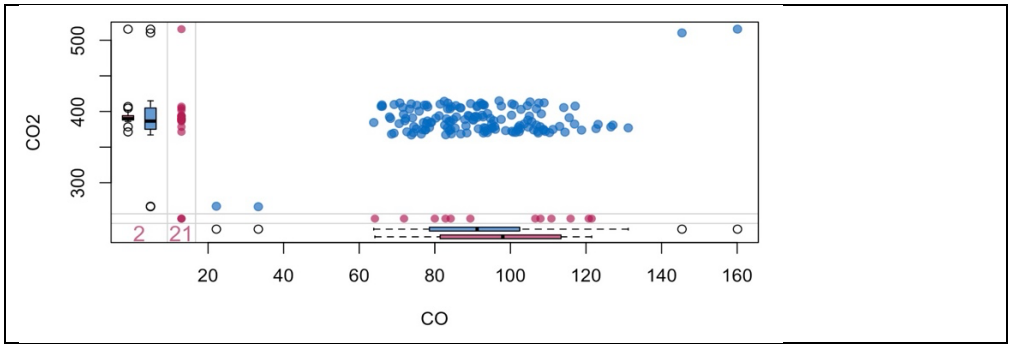


Fig 1.4 – Visualisations of missingness proportions and patterns

From the graphs above, we can discern a certain pattern to the missingness. Only certain variables exhibit missing data, and these variables have missing data only for specific ranges/dates. The missing variables are therefore seemingly linked to other observed variables, than to the missing variables themselves – thus, we can conclude that these variables are Missing At Random (MAR). To deal with them, the missing values were imputed using the MICE (Multiple Imputation by Chained Equations) algorithm. Using the very same plots as shown above, it was confirmed that the missing variables had been dealt with, and the original distribution of the dataset remained unchanged even after imputation.

2. Dimensionality Reduction

Often, when analysing multivariate datasets, we encounter situations where the number of variables is so many that the process of analysis becomes extremely tedious and confusing, requiring multiple scatterplots, histograms and other exploratory visualisations. Such a situation is not tenable, especially when dealing with variables that produce hundreds, possibly even thousands of visualisations and mathematical analyses. This problem is often referred to as the "curse of dimensionality". To address this, it is possible that by using the principles of statistics and linear algebra, we can reduce the number of dimensions of a given dataset to something that is more manageable, producing a smaller set of uncorrelated variables from a larger set of correlated variables. This allows us to produce meaningful and easy to interpret visualisations, apply various unsupervised learning techniques (such as clustering, factor analysis, etc.) and uncover hidden patterns in the data and discern new information that would be otherwise extremely difficult to do so. Of these dimensionality reduction techniques, the two most popular techniques are Principal Component Analysis (PCA) and Correspondence Analysis (CA).

For the given dataset, the dimensionality reduction technique applied was **PCA**. This is because of two main reasons. Firstly, the dataset (*MaunaLoa_miss.csv*) consists of numerical variables that are continuous in nature and are on the same scale (*i.e.* atmospheric gas measurements in ppm/ppb). PCA is designed specifically for these types of variables – CA would not be suitable here as it deals with categorical variables. Secondly, the mathematical techniques (zero mean and unit variance transformation) applied to perform PCA mean that it effectively captures the variance in the dataset, unlike CA which is based on chi-square distances between categorical variables.

To perform PCA, the issue of missing data was dealt with using MICE (as described in the previous section). Then, the data was scaled using the *scale()* function, to standardise the datapoints (*i.e.* transform each variable to zero mean and unit variance) and ensure fair comparisons, such that values of higher magnitude do not dominate the dataset and apply a bias to the output of the PCA. It is important to note that the variable "*Date*" was ignored during the scaling process, as it is irrelevant to the aim of the analysis at hand – as such, it is not present in the scaled dataset. Having already observed moderate to high correlations between multiple variables (refer to the correlation heatmap and scatterplot matrix in the previous section), dimensionality reduction can help remove redundant dimensions. To this end, the *prcomp()* function was applied to the scaled dataset. From the summary of the output, we observe the following:

```
Importance of components:
                          PC1    PC2    PC3     PC4    PC5
Standard deviation     1.7506 0.9781 0.7514 0.63101 0.1264
Proportion of Variance 0.6129 0.1913 0.1129 0.07964 0.0032
Cumulative Proportion  0.6129 0.8042 0.9172 0.99680 1.0000
```

We can see that the first two dimensions capture 80.42% of the variance in the dataset. Usually, the threshold for deciding the number of dimensions retained is 80%; this is greater than that. We confirm these results via a scree plot, where we observe the same:

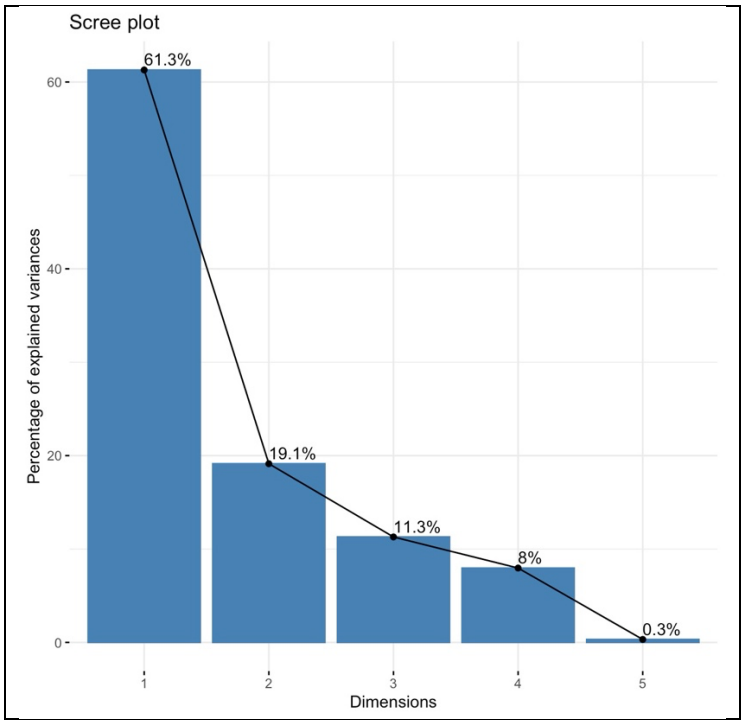


Fig 2.1 – Scree Plot of the given PCA output

Based on the scree plot and the summary statistics for the PCA output, the first two principal components are retained as they explain 80.42% of the variance, thus making it possible for us to drop the other PCs. To analyse how and what these two dimensions represent, A variable loadings plot, and two contribution histograms (for each PC) were constructed.

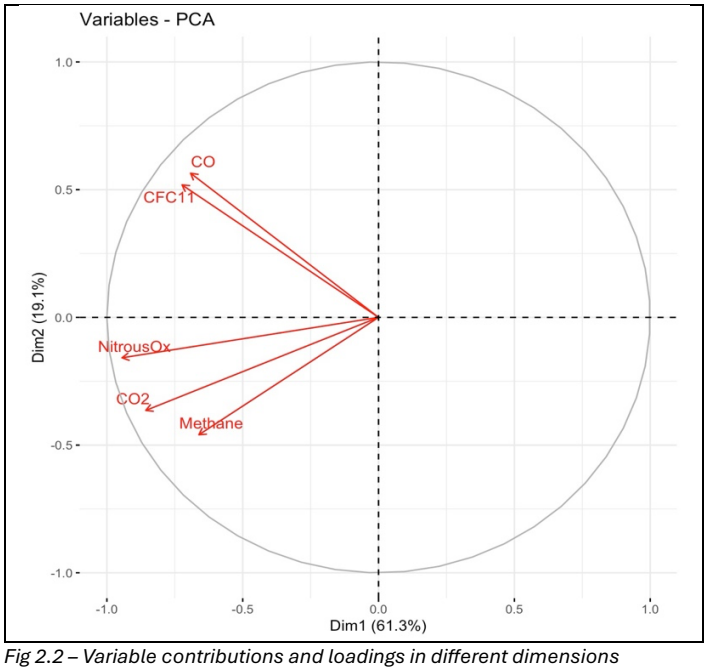


Fig 2.2 – Variable contributions and loadings in different dimensions

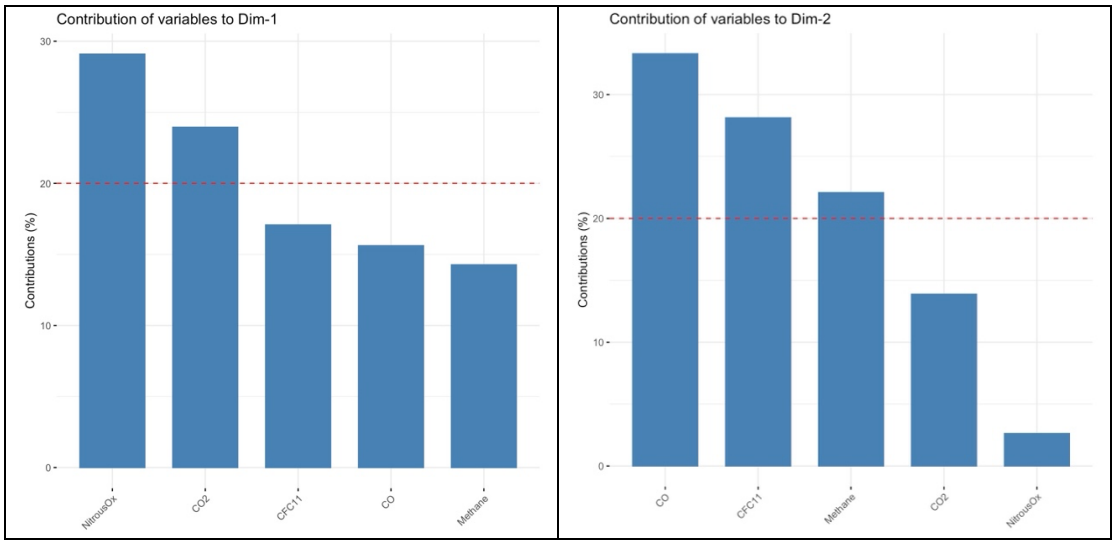


Fig 2.3 – Variable contributions to each PC

Based on our findings from visualizing PCs 1 and 2, we can posit that PC1 seems to capture N_2O, CO_2, and CFC-11 concentrations. These gases are volatile in nature and indicate violent eruption events. However, PC2 seems to capture only carbon-based greenhouse gases – CO, CFC-11 and CH_4 – indicating that it can filter between gas types and is not just capturing general gas concentrations influenced by violent eruptions.

3. Cluster Analysis

Cluster Analysis is an unsupervised learning technique that allows us to group or "bunch" datapoints in a dataset together based on similarities or distances. This allows us to uncover hidden relationships, unique trends and inherent structures within a dataset. In this dataset, clustering is particularly important as the datapoints seem very closely bunched together and distinguishing them from one another is difficult. Furthermore, the presence of outliers and missing values exacerbates the problem of separation by introducing bias. It is important to note, however, that the problem of missing data has been dealt with in the initial phases of the analysis by imputing values via MICE – the complete dataset with imputed values was used for the cluster analysis to minimise the risk of datapoints skewing the interrelations between variables due to missingness. Additionally, the dataset with reduced dimensions (with PC1 and PC2 retained), extracted earlier, was used.

Various clustering methods were explored, which included K-Means Clustering, K-Medoids Clustering and the Gaussian Mixed Model of clustering. Finally, after much deliberation, K-Means and K-Medoids were the chosen methods. What follows is an analysis and contrast of these methods.

For **K-Means Clustering**, the first step was to find the optimal value for K, the number of clusters. This was done via two plots – a WSS (Within-Cluster Sum of Squares) graph and a Silhouette Graph. Both gave wildly different recommendations for K values.

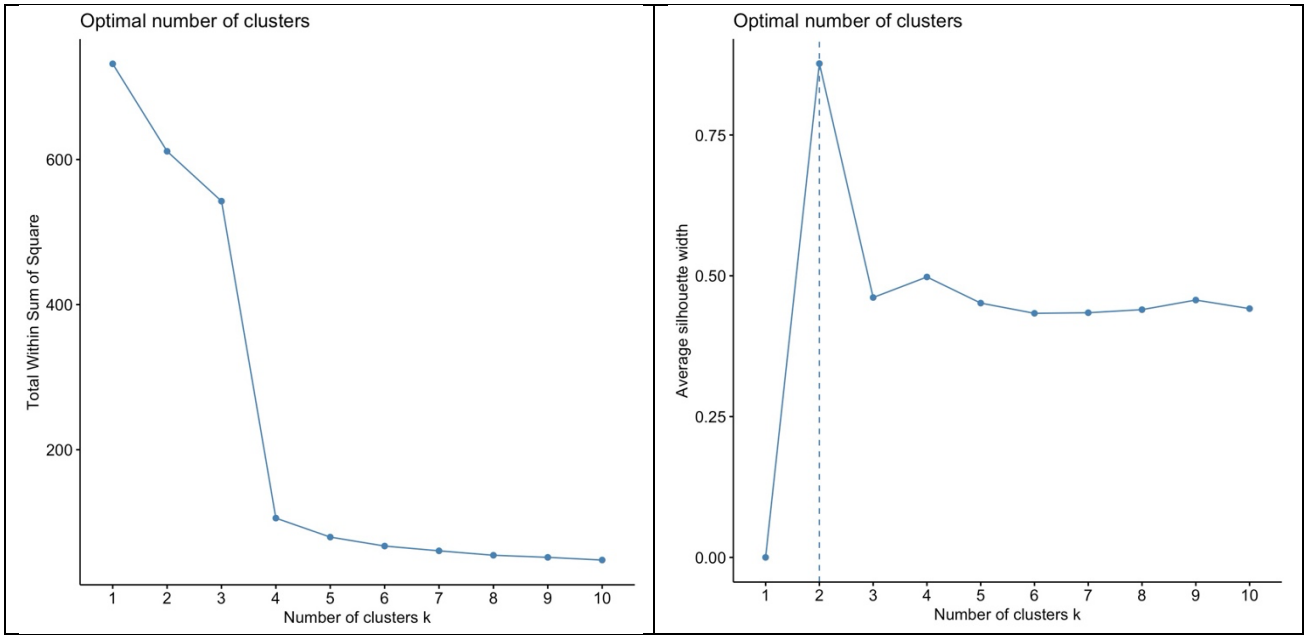


Fig 3.1 – Optimal number clusters as per minimizing WSS/maximizing Silhouette Scores

To decide between K=2 and K=4, the graphs for both were studied. Based on interpretability and increased granularity and based on what was observed with K-Medoids (discussed ahead), it was decided that K=4 is the best value of K. Thus, a visualization for K=4 clusters was produced.

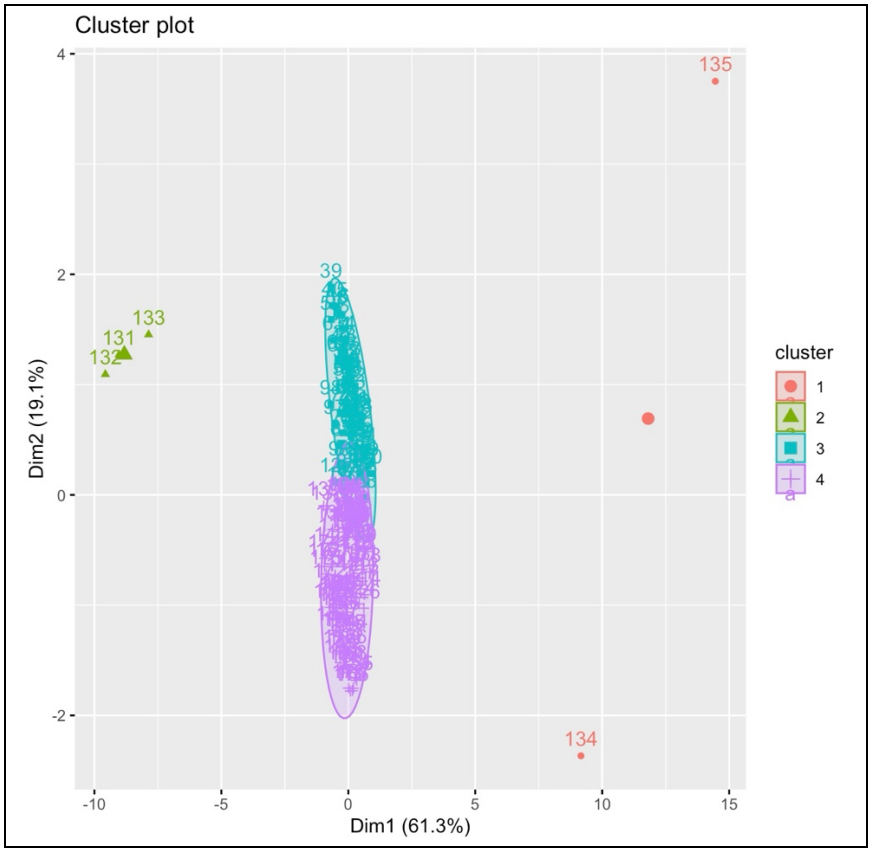


Fig 3.2 – Cluster Analysis for K=4 Clusters

From this plot, we can see that the points are clumped together very tightly, and making out the different clusters is very difficult. Furthermore, the graph is heavily influenced by outliers, with tight confidence intervals and clusters 1 and 2 spread far away from the main "body" of the plot. This main "body" is made up by clusters 3 and 4, which likely represent groupings of similar gases that are emitted in similar, larger, quantities, whereas the outlying clusters are likely less prominent gases present in lower concentrations.

A similar methodology was followed for **K-Medoids**. First, a silhouette graph was produced, which suggested K=9 as the optimal number of clusters for K-Medoids.

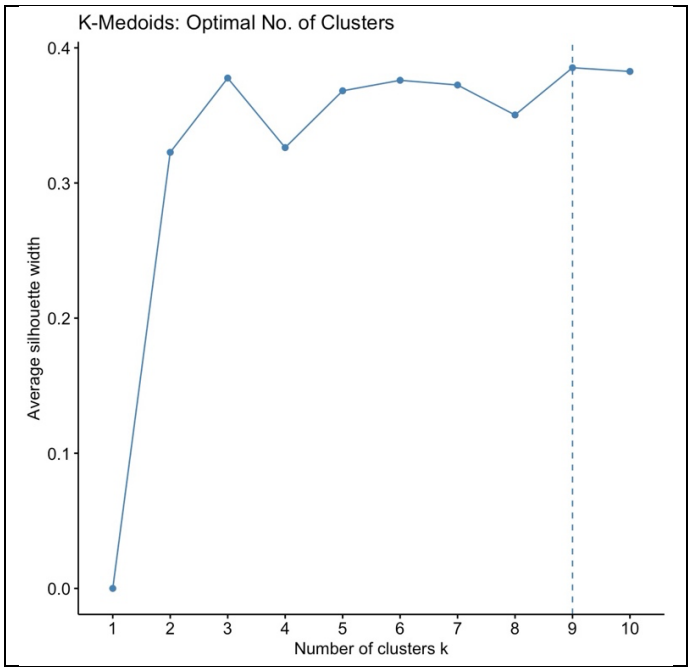


Fig 3.3 – Silhouette Plot for K-Medoids

Based on this, a graph with K=9 clusters using K-Medoids Clustering was produced.

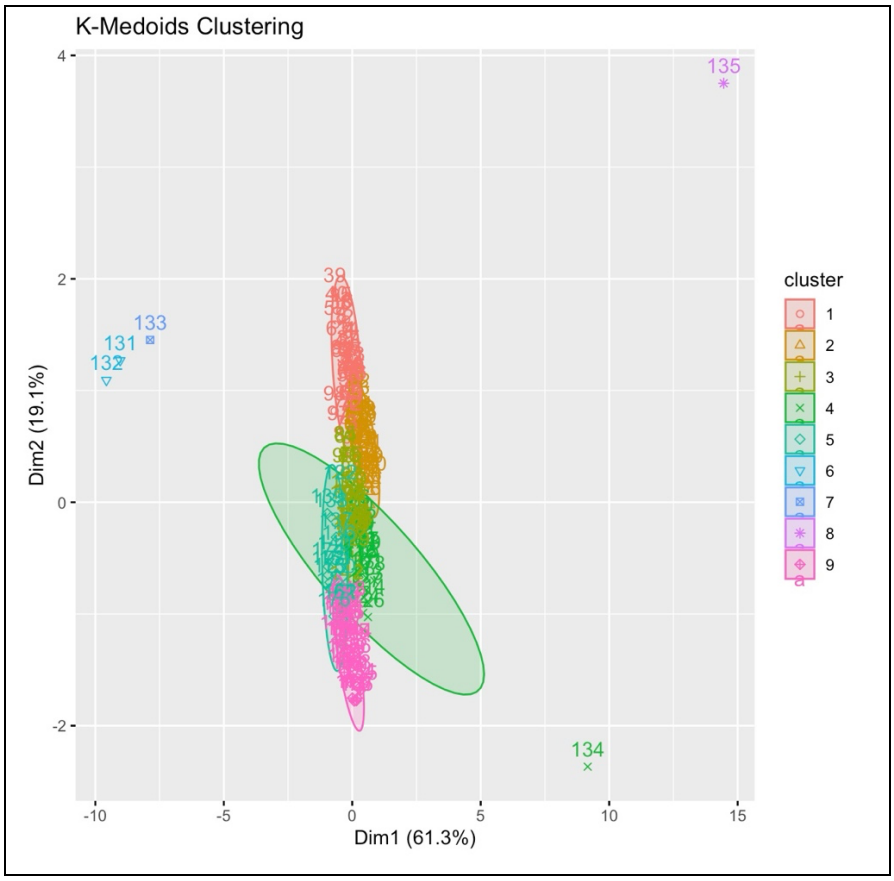


Fig 3.4 – K-Medoids Clustering with K=9

The K-Medoids plots shows a far more granular structure, with fragmented points organized into 9 clusters. 3 clusters capture outliers, while the remaining 6 clump together and show severe overlap, much like the K-Means clustering approach studied earlier. However, this graph is proof of K-Medoids being a more robust alternative to K-Means in the face of outliers. While both graphs are largely similar, the centers of the cluster ellipses are not warped by outliers. This is because K-Medoids uses Medoids (the real central datapoint of a cluster) as centers, while K-

Means uses the mean values of clusters as the cluster centers. Furthermore, due to the influence of sum of squared error values, the clusters in the K-Means plots are swayed by outliers; this is not the case for K-Medoids as it groups datapoints based on the PAM (Partitioning Around Medoids) algorithm. This can be seen visibly when comparing both graphs with one another – datapoint 134 severely distorts the confidence interval in the K-Means graph but fails to do so in the K-Medoids graph.

4. Discussion and Conclusion

The *MaunaLoa_miss.csv* dataset, containing readings for concentrations of five gases in the atmosphere near the Mauna Loa Observatory at the Mauna Loa volcano, was explored, cleaned, and analysed using several unsupervised learning techniques. Initial explorations revealed key insights about the nature of the variables contained within and the structure of the dataset, including variable distributions and correlations between variables and the missingness of the variables, determined to be Missing At Random (MAR). These missing values were imputed using the MICE algorithm and a complete dataset was produced. Then, after scaling the dataset to standardize the values, dimensionality reduction was performed via Principal Component Analysis, or PCA for short. PCA outcomes revealed that the first 2 Principal Components (PCs), PC1 and PC2, captured more than 80% of the variance in the dataset. Thus, a dataset with reduced dimensionality, containing only two dimensions, was extracted. This extracted dataset was then subjected to cluster analysis via K-Means and K-Medoids clustering, two clustering approaches. They both produced similar clusters, with K-Means producing a tightly clumped graph with 4 clusters, heavily influenced with outliers; K-Medoids also produced tightly-clumped graph with overlapping clusters, albeit with 9 clusters and more robust to outliers due to the use of Medoids as cluster centers instead of Means. A key insight here was that despite having extremely similar clusters, K-Medoids produced a far more granular and robust analysis than K-Means.

Two alternative approaches that could have been considered are methods for missing value imputation and clustering. While MICE was an effective tool for imputing missing values, a more accurate imputation could have been achieved using *missForest*. This approach uses a Random Forest machine learning model to understand the distribution and patterns of the dataset and uses that as a basis for imputing missing values. While it is a more computationally intensive method, *missForest* can produce more accurate imputations with a lower risk of inducing bias in a dataset by creating class imbalance. As far as clustering is concerned, it was seen that both K-Means and K-Medoids clustering produced very tight, overlapping cluster plots that were difficult to interpret. This problem could have been addressed using two alternative clustering approaches – the Gaussian Mixed Model (GMM) or Hierarchical Clustering. GMM would have addressed the issue of overlapping clusters by allowing us to mix clusters and contain any misclustering. GMM is also less sensitive to outliers, so it may have produced better groupings by ignoring their influence. Hierarchical Clustering does not require a predefined value of K – this would have been the perfect solution to the initial dilemma, where the WSS and Silhouette plots for the K-Means cluster were producing conflicting values of K. It would have done so by finding the optimal number of clusters via dendograms, producing natural splits in the data, much like a decision tree. Furthermore, despite being more sensitive to outliers and more computationally expensive, the use of linkages would have potentially produced completely different patterns, thus unveiling hitherto unseen features within the dataset.

One of the most important questions that remains unanswered is the question of temporal patterns in the dataset. The dataset used for the analysis contains data collected over approximately 19 years, from 2000 to 2001. The focus of the analysis so far was with regards to the variables and their interrelations divorced from the aspect of time, with the "*Date*" variable excluded from analysis; if clustering analysis had also taken time into account, perhaps we would have seen the evolution of different clusters and discerned changes in atmospheric gas concentration over time, allowing us to understand volcanic activity a lot better than by simply analyzing the variables representing the gases alone. Furthermore, the techniques applied here simply reveal patterns in the dataset and allow for a superficial analysis of absolute trends – they do not reveal any causal relationships between variables or reveal anything about the impact of the variables themselves. For example, we cannot tell if an increase in CO_2 concentrations can be linked to an increase in CO or Nitrous Oxide concentrations, or whether there are more

factors at play. Similarly, we cannot determine what impact these readings had at the time they were made, or how the overall trends in the dataset have affected the environment over the years.

Appendix

```
library(naniar)
library(MASS)
# install.packages("beeswarm")
library(beeswarm)
# install.packages("mice")
library(mice)
library(VIM)
library(plotly)
library(ggplot2)
library(corrplot)
library(GGally)
# install.packages("ggcorrplot")
library(ggcorrplot)
library(beeswarm)
library(tidyr)
library(factoextra)
# install.packages("NbClust")
library(NbClust)
library(cluster)
install.packages("mclust")
library(mclust)


MLoa <- read.csv("/Users/mukundranjan/Documents/Academics/Epiphany/DEVUL/Assignments/Assignment 2/MaunaLoa_miss.csv")

# TASK 1 - EDA:


head(MLoa)
str(MLoa)
summary(MLoa)

#Before proceeding, I wish to convert 'Date' to a proper numeric date value for time series plots.
str(MLoa$Date)
head(MLoa$Date)
MLoa$Date <- as.Date(MLoa$Date, format="%Y-%m-%d")
str(MLoa)    #Ensuring the switch has taken place
sapply(MLoa, class)

#Proceeding with EDA and relevant visualisations:
colSums(is.na(MLoa))
rowSums(is.na(MLoa))
par(mfrow=c(1,2))
boxplot(MLoa$CO, main="Boxplot of CO", col = "red")                #Visualising outliers
boxplot(MLoa$CO2, main="Boxplot of CO2", col = "gray")
boxplot(MLoa$Methane, main="Boxplot of Methane", col = "green2")
boxplot(MLoa$NitrousOx, main="Boxplot of NOx", col = "dodgerblue")
boxplot(MLoa$CFC11, main="Boxplot of CFC11s", col = "yellow")
par(mfrow=c(1,1))

ggpairs(MLoa[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")],
        title = "Scatterplot Matrix of the Mauna Loa Dataset",
        lower = list(continuous = wrap("points", alpha = 0.6, color = "tomato")),
        upper = list(continuous = wrap("cor", size = 4)))                        #Scatterplot matrix


MLCor <- cor(MLoa[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")],
             use = "pairwise.complete.obs")       #Correlation and Correlation Plot
ggcorrplot(MLCor, lab=TRUE,lab_size = 5, , colors = c("dodgerblue", "navajowhite", "tomato2" ),
           ggtheme = theme_classic())

beeswarm(MLoa[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")],
         horizontal = TRUE, col = "dodgerblue", pch = 16, alpha=0.6)

selected_vars <- c(selected_vars <- c("CO", "CO2", "Methane", "NitrousOx", "CFC11"))
par(mfrow = c(3, 2))

```

```r
for (var in selected_vars) {
  hist(MLoa[[var]], main = paste("Histogram of", var), xlab = var, col = "tomato", border = "black")
}


vis_miss(MLoa)        #Visualising missing values

md.pairs(MLoa[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")])

aggr(MLoa[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")], col=mdc(3:4), numbers=TRUE, sortVars=TRUE,
labels=names(MLoa),
     cex.axis=.7, gap=3, ylab=c("Proportion of missingness","Missingness Pattern"))

par(mfrow = c(1,1))
marginplot(MLoa[, c("CO", "CO2")], col = mdc(1:2),
           cex.numbers = 1.2, pch = 19)              #Visualising the missingness for CO and CO2


#Having observed missing values in the dataset, MICE is used to impute values:
imp_MLoa <- mice(MLoa, seed = 123, method = "norm.predict")
MLoa_nomiss <- complete(imp_MLoa)
head(MLoa_nomiss)
str(MLoa_nomiss)

md.pairs(MLoa_nomiss[c("CO","CO2", "Methane", "NitrousOx", "CFC11")])                #Verifying that
missing vals have been removed
aggr(MLoa_nomiss[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")], col=mdc(3:4), numbers=TRUE,
sortVars=TRUE, labels=names(MLoa),
     cex.axis=.7, gap=3, ylab=c("Proportion of Missingness","Missingness Pattern"))
vis_miss(MLoa_nomiss)

ggpairs(MLoa_nomiss[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")],
        title = "Scatterplot Matrix of the Mauna Loa Dataset",
        lower = list(continuous = wrap("points", alpha = 0.6, color = "tomato")),
        upper = list(continuous = wrap("cor", size = 4)))                        #Scatterplot matrix


```

```r
# TASK 2 - PCA:
```

```r

MLoa_scaled <- scale(MLoa_nomiss[c("CO", "CO2", "Methane", "NitrousOx", "CFC11")])    #Scaling to ensure
distribution remians unchanged for PCA

#Running PCA
MLoa_pca <- prcomp(MLoa_scaled, center = TRUE, scale. = TRUE)
summary(MLoa_pca)

#Scree plot and Variable Plot to visualize variance explained
fviz_screeplot(MLoa_pca, addlabels=TRUE)
fviz_pca_var(MLoa_pca, axes = c(1, 2), repel = TRUE,
             col.var = "red")


fviz_contrib(MLoa_pca, choice = "var", axes = 1, top = 10)     #Visualising contributions of varianbles to
PC1 and PC2
fviz_contrib(MLoa_pca, choice = "var", axes = 2, top = 10)     #...PC2

MLoa_pca_comp <- as.data.frame(MLoa_pca$x[, 1:2])        #Extracting components


```

```r
# TASK 3 - Clustering
```

```r

#Using the cluster scree plot to compute the required clusters for K-means clustering:
fviz_nbclust(MLoa_pca_comp, kmeans, method = "wss")
fviz_nbclust(MLoa_pca_comp, kmeans, method = "silhouette")

#As the WSS and Silhouette methods suggest different Ks for clustering, we will test both approaches:
set.seed(123)

```

```r
k2 <- kmeans(MLoa_pca_comp, centers = 2, nstart = 25)
k4 <- kmeans(MLoa_pca_comp, centers = 4, nstart = 25)

fviz_cluster(k2, data = MLoa_scaled, ellipse.type = "norm")       #Clustering for both values of k are compared
                                                                  #And k=4 is chosen as it has the highest granularity and
                                                                  #most easily interpreted graph.
fviz_cluster(k4, data = MLoa_scaled, ellipse.type = "norm")



#K-medoids:
fviz_nbclust(MLoa_scaled, clara, method = "silhouette") + labs(title = "K-Medoids: Optimal No. of Clusters")
MLoa_kmed <- clara(MLoa_scaled, k=9)
fviz_cluster(MLoa_kmed, data = MLoa_scaled, ellipse.type = "norm") + labs(title = "K-Medoids Clustering")


#Gaussian Mixture Model (GMM):
MLoa_mclust <- Mclust(MLoa_scaled, G = 1:10)
plot(MLoa_mclust, what = "BIC")

#Choosing the EVI model and making relevant plots:
MLoa_mc_EVI <- Mclust(MLoa_scaled, G = 6, modelNames = "EVI")
plot(MLoa_mc_EVI, what = "classification")
plot(MLoa_mc_EVI, what = "uncertainty")
plot(MLoa_mc_EVI, what = "density")
plot(MLoa_mc_EVI, what = "BIC")

```