DEVUL Assignment 2

Introduction

Climate change represents one of the biggest challenges facing humankind. Therefore, monitoring atmospheric composition is crucial in understanding both the short- and long-term changes in the Earths climate resulting from human activities. This dataset presents monthly average measurements of greenhouse and trace gases in the environment taken from the Mauna Loa Observatory, from 2000 until 2019. It contains readings for five key gases: carbon dioxide (CO2), carbon monoxide (CO), methane, nitrous oxide (NO2), and chlorofluorocarbons (CFC11). In this study I apply data exploration and unsupervised learning techniques on the dataset to understand trends in environmental composition, with the aim of identifying how this corresponds to global events and shifts in policy.

Exploratory Data Analysis

As previously mentioned, the data contains monthly averages for key gases in the environment (Table S1). All the unprocessed data appears to have a linear relationship with time, except for CO which shows higher stochasticity (Figure 1). Outlying values can be seen in 2013 where, for all five variables, there appears to be a drastic spike, which appears to impact the data. This is corroborated with complex time series analysis, which found that the strong trend and seasonal features of this data, cannot account for a large residual value at 2013 (Figure 2). The presence of these extreme values may be due to improper data collection, for example equipment or recording error. Although due to the presence of missing data this will only be handled after missing data analysis as it may be important in understanding the mechanism of missingness at play. The relationships between the continuous variables can be observed in Figure 3. This shows us that the only strong correlation between variables is between NO2 and CO2 (rho = 0.939), although observation of the scatterplots show that these values may be heavily influenced by these extreme values.

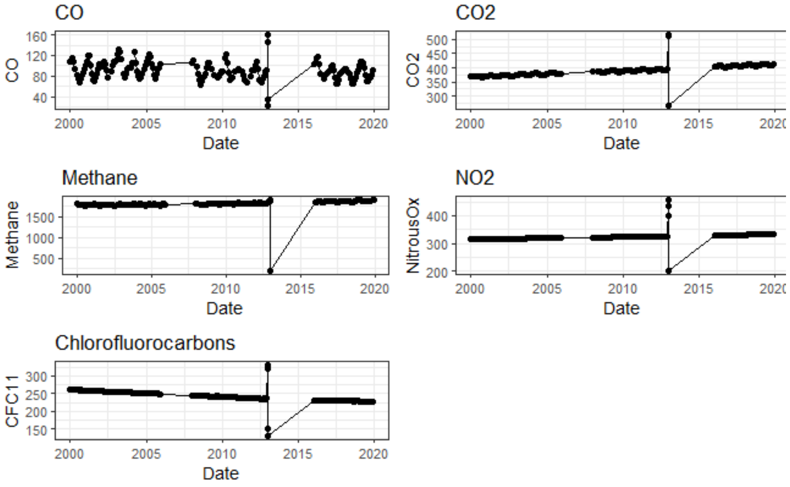


Figure 1. Change in all unprocessed environmental gas data with time. CO2 = parts per million, CO/Methane/NO2 = parts per billion, CFC = parts per trillion.

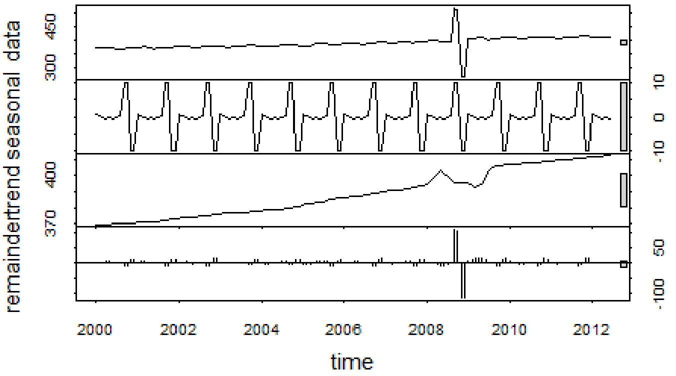


Figure 2. Seasonal decomposition results of the CO2 variable.

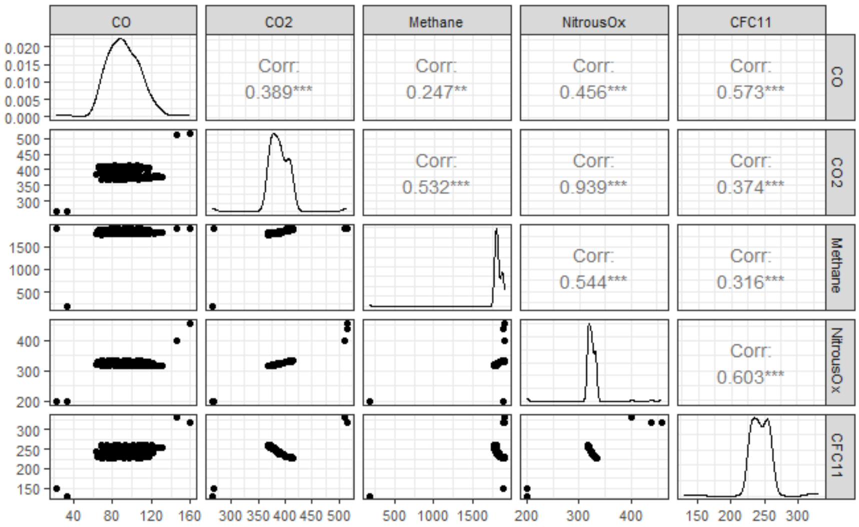


Figure 3. Scatterplot matrix of unprocessed continuous variables. Correlation value calculated is Spearman's Correlation Coefficient.

This data shows large amounts of missingness in the recorded values. At the dates recorded only the variables of CO and CO2 contain missing values, with the former showing a larger proportion of missingness (10.4%) then the latter (6.6%), with 1.1% of the data contains missing values for both CO and CO2 (Figure 4). Further, there exist gaps in the data regarding the dates of collection (Figure 5), with the time periods of 2006 to 2007, as well as 2013 to 2015. These gaps represent major limitations in the provided data, as there is no discernible way to understand how the trends continue during these periods of time, requiring assumptions about it keeping to those observed previously. The relationship between missing CO and CO2 values with time, suggests a moderate trend, with the rate remaining roughly constant from 2000-2007. Missing data for CO increased between 2008-13 whereas CO2 remained constant, followed by a near complete loss of missingness for both variables. This trend implies that the relationship is not Missing Completely at Random (MCAR) for CO. This is when the missingness is independent of all data and therefore ignoring it will not introduce bias of any sort. Despite this, when employing a Little's MCAR test, it strongly suggests insufficient evidence against the null hypothesis of a present MCAR missingness pattern (p-value = 0.98…). However, Little's test has low statistical power and can miss patterns of missingness, therefore may not identify temporal patterns (Kaur, 2025), and therefore an alternate relationship may be at play, Missing Not at Random (MNAR) or Missing at Random (MAR).

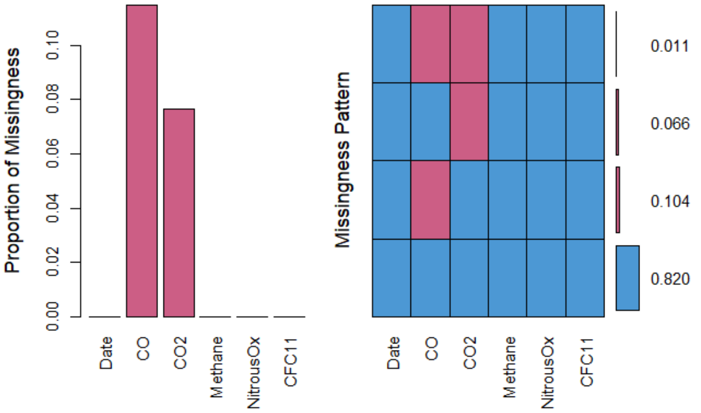


Figure 4. (a) Proportion of each variable that is missing data, (b) The missingness pattern observed in the data, pink represents missing data, numbers represent the proportion of the data.

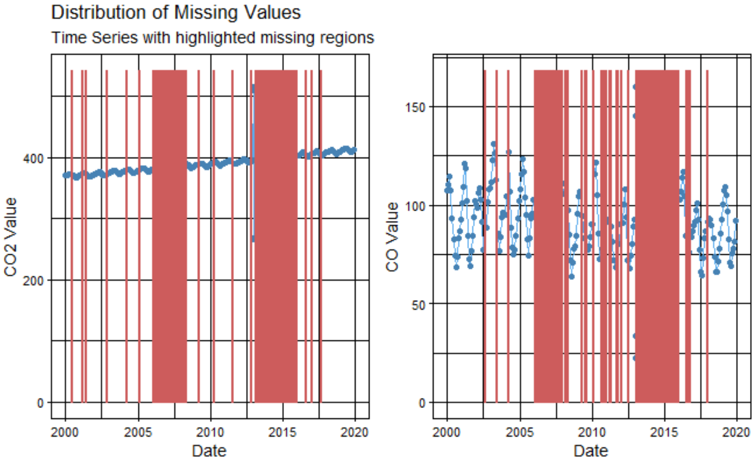


Figure 5. Distribution of missing values for CO2 (left) and CO (right) over time.

MNAR represents data that is missing dependent on the true value of the observed data, whereas MAR's missingness is dependent on the other observed variables. To discern between the two, you can visualise the spread of the data that contains a missing value, compared to the non-missing data. This can be seen in Figure 6/7. Figure 6 shows that the distributions for methane, CFC11, and NO2, all of which contain no missing values, which do not differ when missing values of CO or CO2 present. This is further supported by Figure 7, showing that extreme values for these variables are equally likely regardless of the presence of missing CO/CO2 data. Therefore, the presence of missing data in CO or CO2 has no relationship with the presence of extreme values for other continuous variables. The relationship between missingness of CO and CO2 can be observed in Figure 8. This shows that missing values of CO2 occur only at normal values of CO, with the opposite showing similar results, with the exception of a single point taken during 2013. Taken together, the lack of a relationship between missingness, extreme values, and distribution of data suggests that the data does not follow MAR relating to the other continuous variables, in line with the results from Little's test.

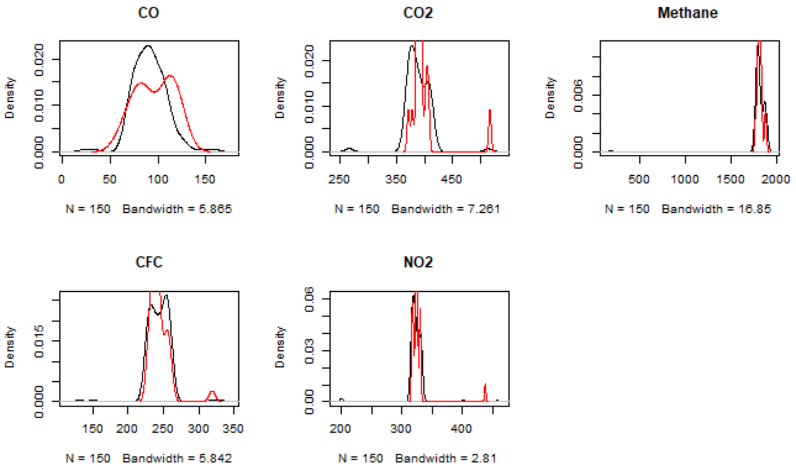


Figure 6. Distribution of variables that are present in partially filled rows, with NA's present (red), and filled rows (black).

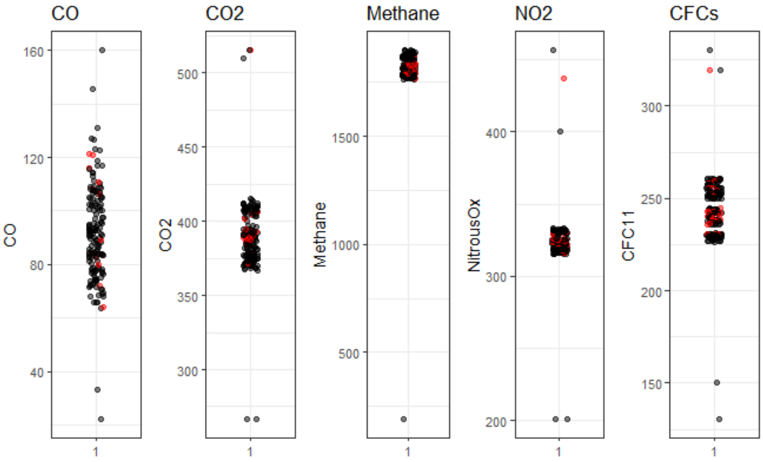


Figure 7. Jitter plot of the values of variables present in partially filled rows (red) and filled rows (back).

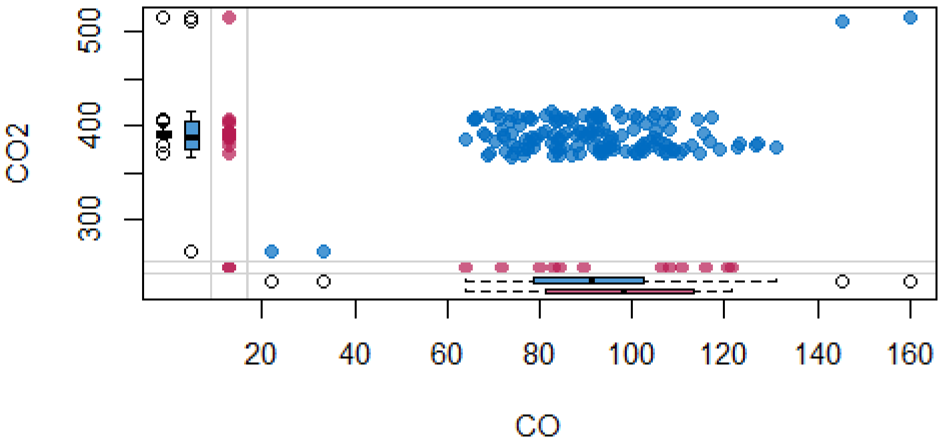


Figure 8. Scatterplot of the present (blue) and missing (pink) values for CO and CO2. Boxplots represent the distributions of these variables.

The previously mentioned outlying data points present in 2013 are visualised in Figure 9. Every row in the 2013 collected data had at least one variable with an outlying value, determined by the IQR rule for outliers (Table S2). Similarly, the data collection trends of 2013 do not resemble the pattern seen in the rest of the data, with readings being taken from January 1st till the 5th, with no other data being recorded following. This implies that repeat readings may have been taken due to complications with equipment, resulting in incorrect data. If real, the sudden environmental shifts would cause major life-threatening instances. For example, the lowest CO2 values recorded were 266 and 267, which would cause such significant global cooling, that similar levels were observed during the ice age (Lindsey, 2024). Therefore, the results recorded in 2013 are biologically implausible, and so due to errors in data recording. As a result, I will remove the data from 2013 from the data, and subsequent analysis. The relationships between the outlier removed variables can be seen in Figure 10.

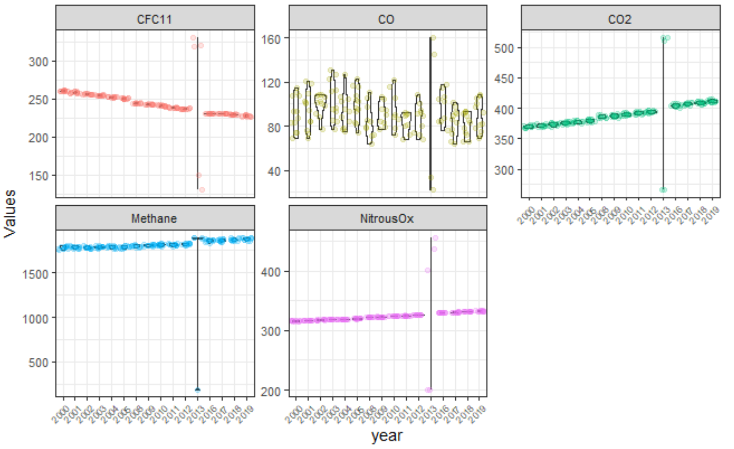


Figure 9. Violin plot of the distributions of all variables split by year.

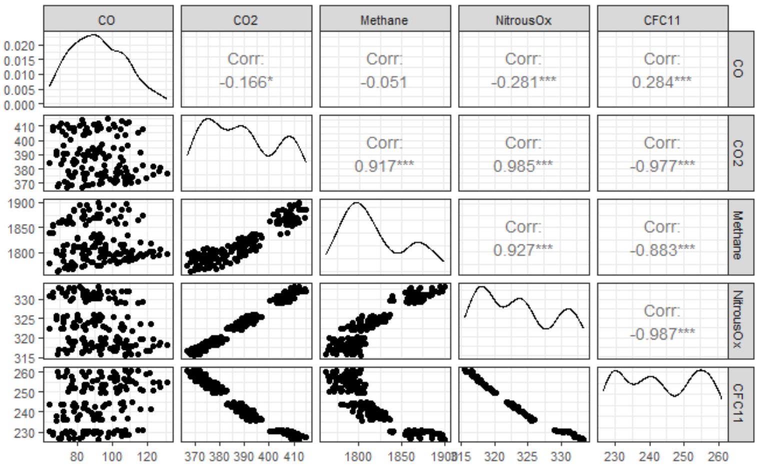


Figure 10. Scatterplot matrix of the outlier removed data

The strong correlations seen in Figure 10, imply that values of methane, NO2, and CFC11 are highly predictive of the value of CO2. As a result, the lack of relationship between missing values and extreme values for these variables (Figure 7) suggest that the missingness of CO2 is not dependent on whether its own value was extreme. These patterns of missingness suggest that CO2 shows a missingness mechanism of MCAR, whereas CO may show MCAR, MNAR, or MAR related to date. Due to this lack of concrete understanding, I will not employ complete or available case analysis as this may introduce bias if the data is not MCAR. Rather, therefore I will employ imputation techniques to mitigate this bias, whilst increasing the statistical power of my data. The techniques I will explore both multiple imputation techniques, upon the partially missing rows of data, as well as time series imputation, upon the entire range of dates between 2000 and 2019. To compare these, I will observe the differences in distributions of the imputed and observed data, and their interactions with other variables.

Dimensionality Reduction

To impute related to time series data I employed a Kalman smoothing method from the ImputeTS package. This imputed data showed major limitations when observing the imputed values interaction with time (Figure 10). The complex seasonal relationship was simplified, and the imputed data rather followed a simple polynomial of the trend. This major loss of information would significantly impact my analysis and therefore I elected to use a multiple imputation approach. I and going to use multiple imputation over single imputation, as it allows for the imputed data to account for the variability that would have been seen in the observed data. Comparing regression, random forest, and superMICE imputation techniques (Figure S1) the superMICE imputed data distributions were most similar to the observed (Figure 11). This technique is an ensemble approach utilising multiple approaches within a weighted average equation to calculate the imputed value. The methods I selected were random forest, regression, and Lasso regression. As these methods required complete data for imputation of a specific column, initial imputations were generated using linear regression, then replaced with the superMICE estimations (see Supplementary Code). The distributions and relationships between the imputed values align with the observed, suggesting that that these values were similar to what the actual values were.

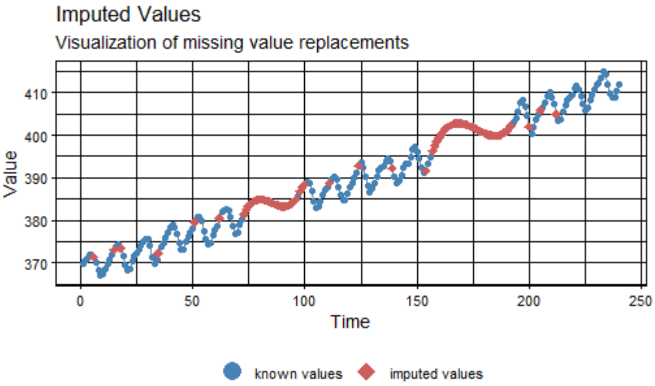


Figure 10. Time series of CO2 values of the present (blue) and imputed (pink) values.

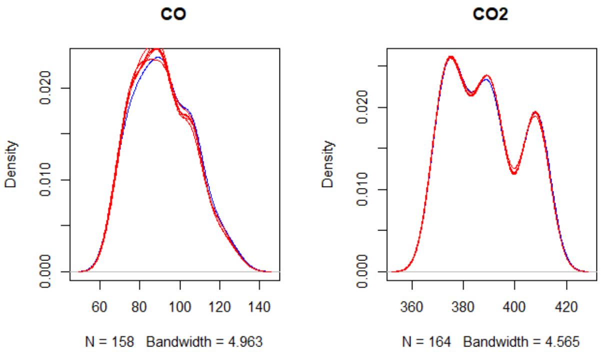


Figure 11. Kernel density estimations of my observed data (blue) and imputed data (red) for superMICE imputation, with m = 5.

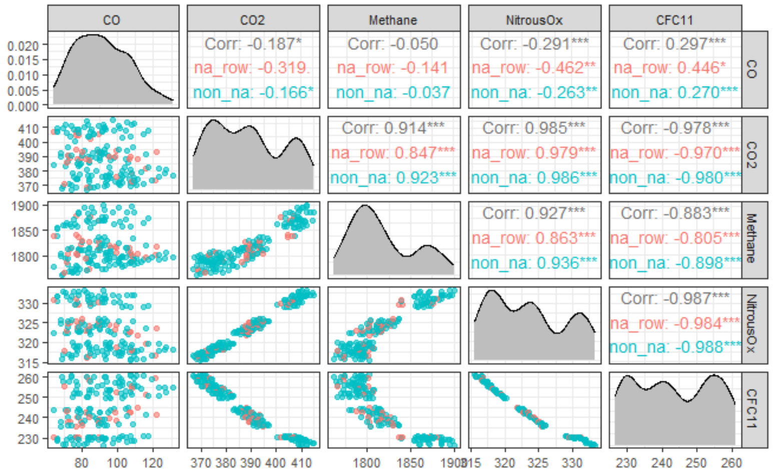


Figure 12. Scatterplot matrix of fully processed data. Blue = observed data, pink = row with imputed data.

Principal Component Analysis (PCA) was chosen as the primary dimensionality reduction method instead of alternative non-linear approaches like t-SNE and Kernel PCA, primarily because the data showed strong linear relationships among the variables. PCA is well-suited to linear data by identifying new, Principal Components (PCs) that maximise variance along each successive component. Then by selecting the PCs that explain much of the total variance, PCA reduces the datasets dimensionality whilst still retaining the majority of the information, by removing redundancy. Further, PCA is preferable due to the interpretable nature of the loadings, which allow us to understand how the original variables contribute to each PC (Awan, 2023).

To perform PCA I first scaled my imputed data, followed by the use of the prcomp() function in R. The resulting PCs are linear combinations of the original variables that maximise the information retained by removing all redundant information, causing the new variables to be uncorrelated with one another. The first PC moves in the direction of where the most variation lies in the data; this results in this PC containing the greatest proportion of the data (77.96%). The second PC is in the direction orthogonal to the first, thereby preventing redundancy, containing 19.66% of the proportion of variance in the original data (Figure 13, Table S3). I chose to utilise the first two PCs in my analysis based on the scree plot visualisation, as after the second PC the proportion of variance explained drops off, and therefore this area is the "elbow." Further, these two exclusively accounted for over 95% of the cumulative proportion of variance, I chose to include these.

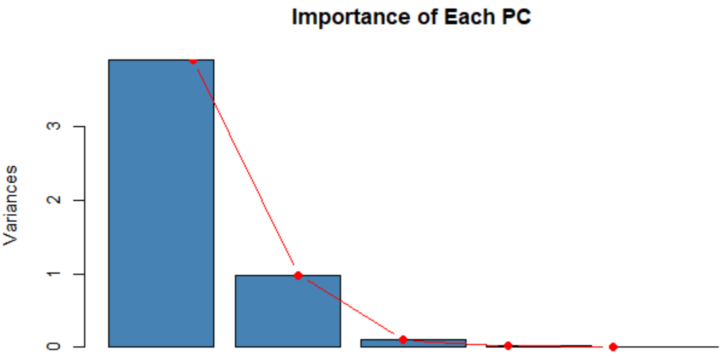


Figure 13. Scree plot showing the proportion of variance explained in each principal component.

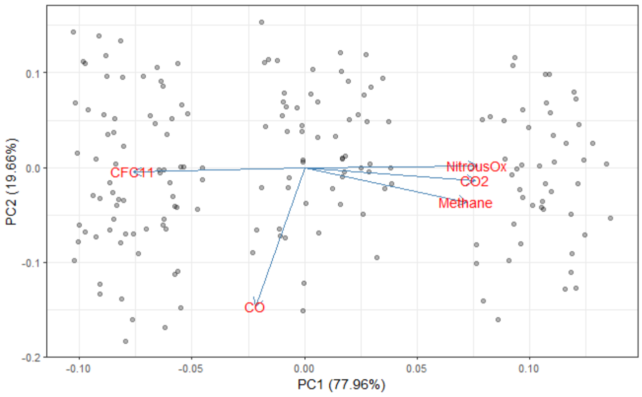


Figure 14. Distributions of data on PC 2-dimensional space. The arrows represent how the variables were represented in each PC.

Each point of data has been plotted in the new 2-dimensional space in Figure 14. The information in PC1 is primarily obtained from the variables that show a linear relationship with time: CO2, NO2, methane, and CFC11, with CFC11 being the only variable that decreases. This relationship means that all of these variables contain redundant information that is contained in PC1. Resulting from this, there is a clear positive trend in the values of PC1 over time. Whereas PC2 contains information almost exclusively from CO, and its stochasticity results in PC2 having no clear trend with time, with both PCs still containing strong seasonal components (Figure 15). There are apparent gaps in this new vector space between PC1 values of -0.05 and -0.025, as well as 0.025 and 0.075. These stem from the gaps in the data collection over time, which caused the values of each of the contributing variables to have jumps rather than continuous change (Figure 1).

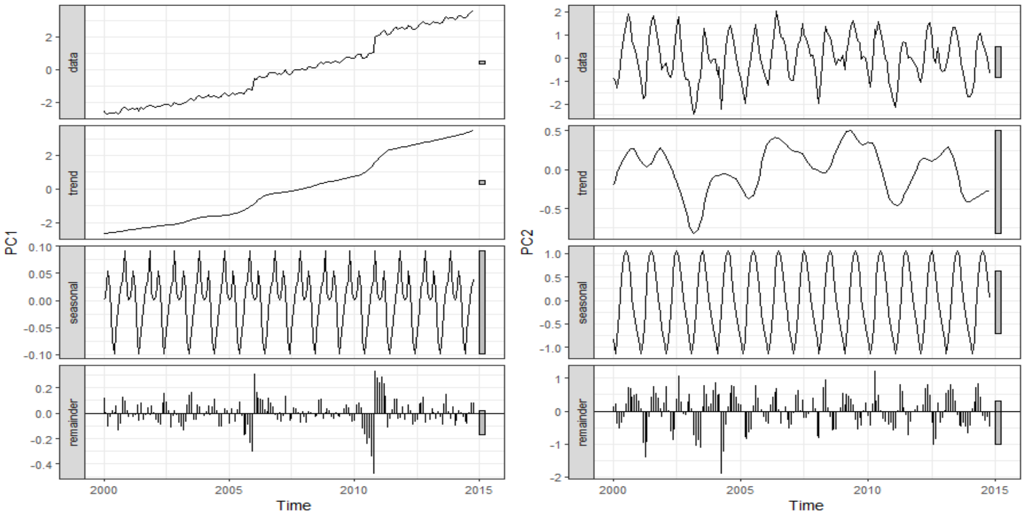


Figure 15. Seasonal decomposition analysis for PC1 (left) and PC2 (right)

Cluster Analysis

The application of clustering techniques means you can make groupings based on similarity between in the data, allowing for analysis of the similarities and differences between different points. It is also beneficial in identifying groupings, structures, and patterns not readily apparent in the data. The use of transformed two-dimensional data is preferable as it allows for simpler visualisation and interpretation of the clusters, allowing for easier identification of movement within and between clusters between data points. This is especially relevant for time series data. The combination of these elements allows us to identify the specific time periods associated with environmental compositions. Identification of movement in this two-dimensional clustered space can inform us about the trends in gas composition and we can align these with policy changes and global incidents to try and understand why it occurred.

To accomplish this, I will use two clustering methods, one hard and one soft, and utilise both results in tandem. Hard clustering techniques, including K-means, K-medoids, and hierarchal, assign a data point to a single cluster, based on a similarity measure. These “hard” boundaries are beneficial when ambiguity around clusters is undesirable. Soft techniques, however, involves the assignment of a probability or degree of membership to a cluster for each point. This allows for a single point to belong to multiple clusters, with varying degrees of membership (Mahmood, 2024), allowing for the capture of more uncertainty within the data. As we are unaware about the actual structure of the clusters it is beneficial to use models with differing methods and assumptions.

For my hard clustering method, I opted to use K-means clustering over hierarchal or K-medoids. The primary advantage of hierarchal clustering is its visualisations using dendrograms, however, as my data is already 2-dimensional, visualising differences between points is not a challenge. K-medoids is a method similar to k-means, though its centroid is plotted to an actual observed point. This makes it less susceptible to outliers, but these have already been removed from our data, therefore, K-means is optimal. It is an approach that minimises the sum of squared distances between data and their cluster centres. To find the optimal value of K, I elected to use internal validation measures rather than external to not introduce any bias. I used silhouette measure as my validation metric and this accounts for both intra- and inter- cluster variation, with values close to +1 suggesting a high affinity to its own cluster, and low to others. It was selected over within sum of squares as this metric only measures the variation within clusters, not between.

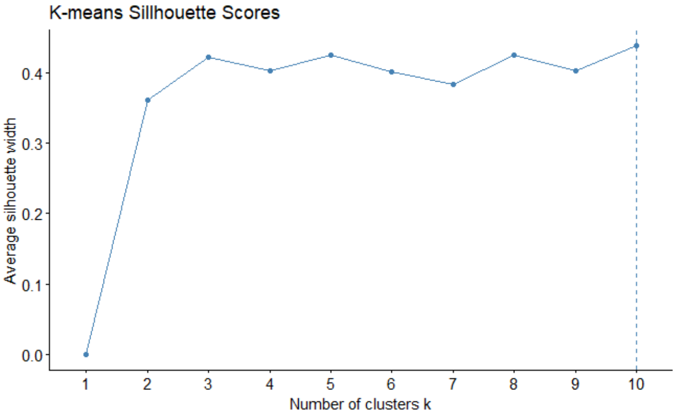


Figure 16. Average silhouette score for values of K from 1 to 10. The dashed blue line represents the greatest score.

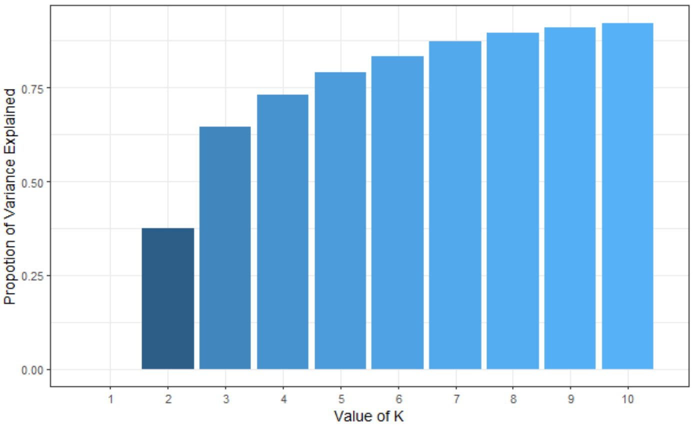


Figure 17. Proportion of variance explained by all clusters for values of K from 1 to 10.

Silhouette scores for all values of K can be seen in Figure 16. After K = 3, the average silhouette scores remained consistent but reached a peak at K = 10. But due to the similarities in values between 3 and 10, I opted to use K = 3, as the less clusters you use allows for better interpretability and understanding of the underlying trends in the data. This choice was corroborated when observing the variation explained by the clusters for each value of K (Figure 17). This variation is calculated by dividing the within sum of squares by the total sum of squares, this will naturally increase with K, but to prevent overfitting, the value of K = 3 is where it appears to level off. The resultant clusters can be seen in Figure 18. Clusters 1 and 2 are defined by lower values of PC1, but split by values of PC2, whereas cluster 3 accounts for points associated with higher values of PC1. All of the data appears to have been properly classified, except for point 122, which lies within the range of cluster 3 but belongs to 2.

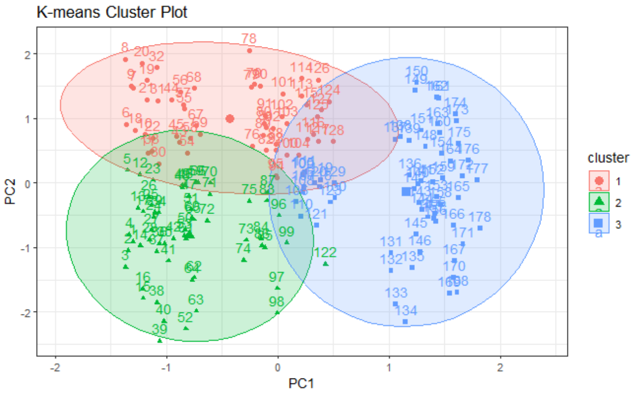


Figure 18. Clusters generated by K-means clustering.

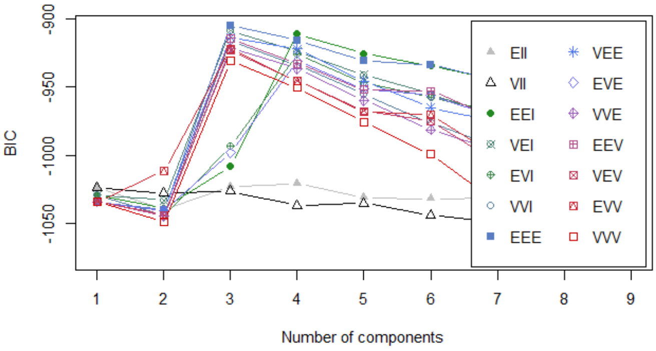


Figure 19. BIC values for different number of clusters, and parameters for Gaussian Mixture Modelling.

For my soft, clustering method I elected to use Gaussian Mixture Modelling. This method determines clustering using probability measures, with each cluster being modelled by a multivariate normal distribution, containing both a mean, covariance and mixing probability. Similarly to K-means, it requires you to explicitly state a value of K, and to validate this I used the Bayesian information criterion (BIC). This was selected over the Akaike information criterion, due to differences in parameter penalisation. BIC punishes model complexity more strongly, preventing overfitting, allowing us to find underlying structures better. In terms of BIC, the best performing model occurred at K = 3 (BIC = -904.99) (Figure 19), with model options stating that the clusters are of equal variance, shape and orientation within the PC dimensional space. This model resulted in the clusters seen in Figure 20/21. The joint probability distribution of each cluster shows circular and symmetric clusters, with distinct separation between groups, implying the presence of meaningful clusters. It classified the points in three distinct groups with separation purely in the PC1 axis, resulting in a linear relationship with time that is a lot less erratic than K-means (Figure 22). Importantly, however, boundaries between clusters occur at exclusively at the previously mentioned gaps in the PC dimensional space, which may have had a negative impact on the results.

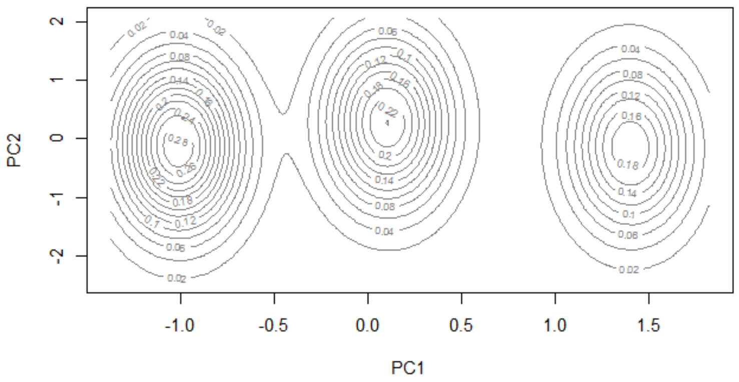


Figure 20. Gaussian distribution plot of each cluster. Contours represent the probabilities.

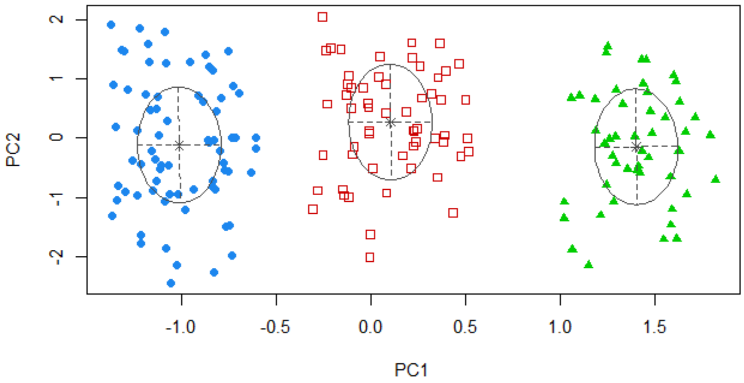


Figure 21. Classification of points using Gaussian model.

The gaussian assumptions that the clusters have equal variance, shape and orientation, are equivalent to the assumptions seen in K-means, implying that K-means was an appropriate method to employ in this investigation. Though, despite this, clusters between models took distinctly different shapes, due to the differences in optimisation procedure causing different fitting of clusters.

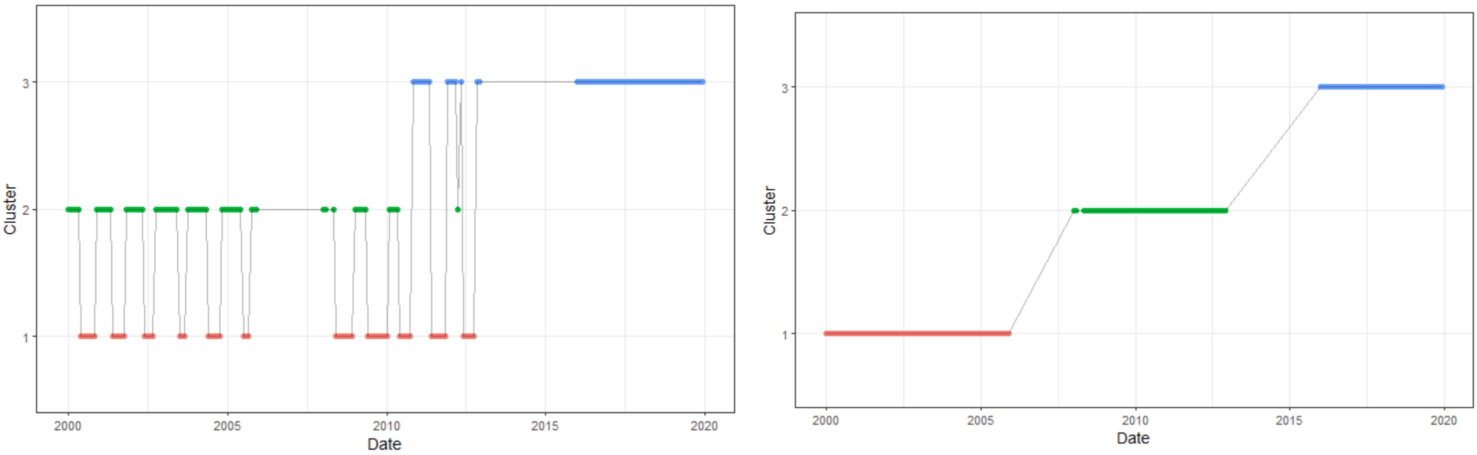


Figure 22. Change in cluster over time for K-means clustering model (left) and Gaussian Mixture Model (right).

Discussion and Conclusion

Both sets of clusters tell a similar story, in that the environment has significantly transitioned from compositions seen in previous years, though the specifics of which differ. My Gaussian clusters are purely separated by values of PC1, associated with methane, CO2, NO2, and CFC11. The time related transition from cluster 1 to 3 between 2000 and 2019 is suggestive of worsening environmental composition, though increased CO2, NO2, and methane levels. This trend can also be seen in the K-means clusters, though this is more complex. There is no association between cluster 1 and 2 with time, caused by the volatile nature of CO represented in PC1. However, as time passed the influence of CO on cluster boundaries weaned, resulting in cluster 3, similarly to Gaussian, only being influenced by PC1 value. Both models suggest slight differences in the transition into this most modern cluster (Figure 22). The Gaussian model shows a distinct transition between the years of 2013 and 16, when there is no data available, however K-means suggests a more gradual change from 2011. This likely results from the size of the clusters created, as the formers are spread over the entire range of PC1, and the latter's cluster 3 runs over all positive values of PC1. Despite the differences, both models suggest the same message however, that between 2000 and 2019 the environmental composition has been evermore associated with higher values of PC1, and subsequently higher methane, CO2, and NO2. Despite dangerous CFC11 levels decreasing, these results are still indicative of a worsening environment causing climate change.

The presence of missing data represents a unique challenge in this dataset. Missing data has a distribution which, as previously mentioned, is imputed into the data via multiple imputation techniques, allowing for its uncertainty to be accounted for. However, during my analysis I only used a single imputed dataset, therefore this robustness is not represented in my analysis. Comparison of the imputed points across all (m = 5) imputed datasets in my PC 2-dimensional space suggests that, for most points, there is not too great variation (Figure 23). Despite this, few points show greater variation than others, which may influence my cluster analysis. The use of different imputed datasets may influence the clusters formed and the conclusions gained from them. Further research should focus on the relationship between missing value distribution and cluster analysis, highlighting if there are any changes in the cluster composition, or Gaussian assumptions.

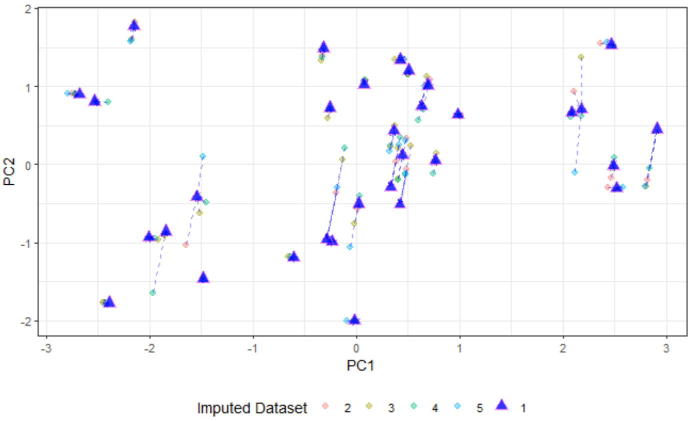


Figure 23. Differences between imputed values used in my cluster analysis (blue) against all other imputed values, in the PC 2-dimensional space. Lines represent the size of the difference of equivalent imputed values.

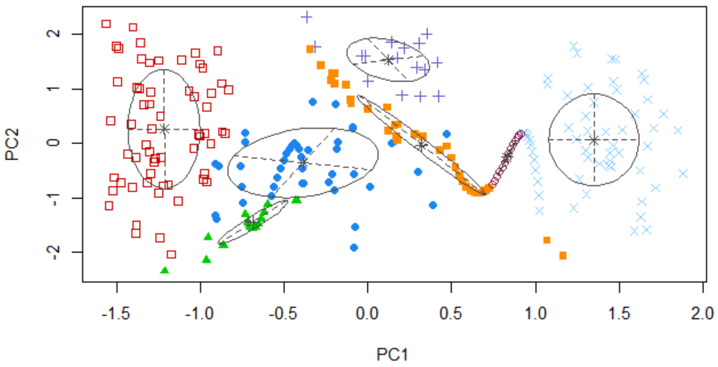
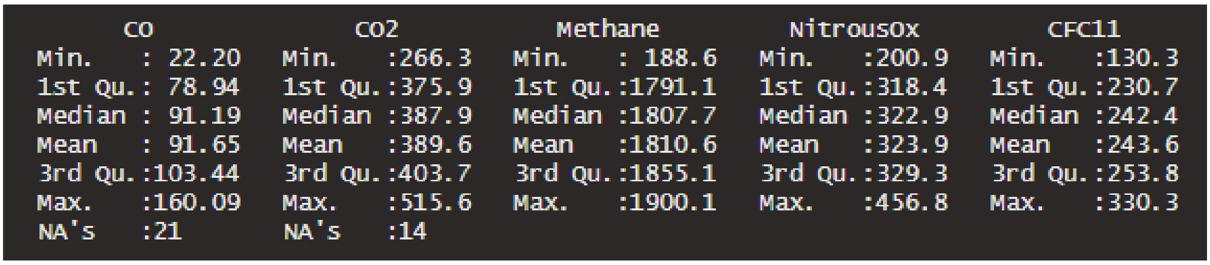


Figure 24. Cluster analysis of the time series imputed data using Kalman smoothing. The optimal parameters were K = 7, with parameters: variable shape, orientation, and volume.

Further, the choice of imputation method also has a large impact on the result of my clustering analysis. The method I selected (superMICE) did not address the periods of missing data, as it is reliant on partially filled data for imputation. This resulted in large gaps in my data, resulting in clusters that may not be representative of the true relationship. For example, in my Gaussian model, the boundaries were distinct and were associated with the gaps in the time series data, therefore we cannot make inferences about how the change between clusters occurred and the underlying structure of the data. When using time series missing value imputed data, with a Kalman smoothing approach, the optimal Gaussian mixture model shows suboptimal results (Figure 24). The lack of noise, alongside generalised seasonal trends, cause the points, when transformed into the PC dimensional space, to follow these extract trends, therefore making us unable to make any implications from this. Further research should focus on understanding what occurred during these periods of missing data, through more advanced data imputation techniques, such as supervised machine learning methods, like LSTM or GRU. These would allow for the capture of complex temporal relationships through their neural network architecture and therefore would make more accurate predictions than the Kalman smoothing approach I used. Good quality imputation will allow us to discover the trends and patterns that occurred during these transition periods. This will allow us to better understand what caused the transition from clusters associated with positive environmental composition to negative, which can inform future policy choices.

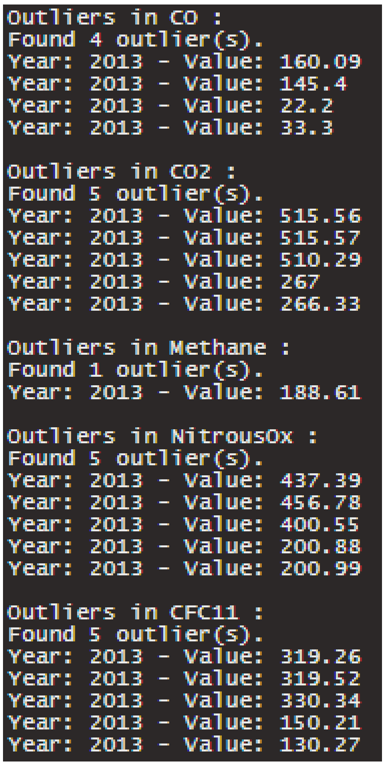
In conclusion the application of unsupervised learning techniques has allowed us to elucidate the structural changes in environmental composition over the years. Although the precise mechanisms of how and when the atmospheric composition shifted from clusters 1 and 2, to 3 is unknown, it is suggestive that the environment is worsening. Based on the findings from this paper, policy makers should focus on reducing methane and CO2 levels as they are responsible for driving the largest changes in PC1. Despite this the association between increased PC1 and lessened CFC11 is hopeful, as it suggests that meaningful change in policy can and will decrease the levels of gases in our environment harmful to the climate.

Supplementary Materials:



```
       CO                CO2             Methane           NitrousOx           CFC11
 Min.    : 22.20   Min.    :266.3   Min.    : 188.6   Min.    :200.9   Min.    :130.3
 1st Qu.: 78.94   1st Qu.:375.9   1st Qu.:1791.1   1st Qu.:318.4   1st Qu.:230.7
 Median : 91.19   Median :387.9   Median :1807.7   Median :322.9   Median :242.4
 Mean   : 91.65   Mean   :389.6   Mean   :1810.6   Mean   :323.9   Mean   :243.6
 3rd Qu.:103.44   3rd Qu.:403.7   3rd Qu.:1855.1   3rd Qu.:329.3   3rd Qu.:253.8
 Max.   :160.09   Max.   :515.6   Max.   :1900.1   Max.   :456.8   Max.   :330.3
 NA's   :21       NA's   :14
```

Table S1. Summary statistics for all continuous variables present in the data.



```
Outliers in CO :
Found 4 outlier(s).
Year: 2013 - Value: 160.09
Year: 2013 - Value: 145.4
Year: 2013 - Value: 22.2
Year: 2013 - Value: 33.3

Outliers in CO2 :
Found 5 outlier(s).
Year: 2013 - Value: 515.56
Year: 2013 - Value: 515.57
Year: 2013 - Value: 510.29
Year: 2013 - Value: 267
Year: 2013 - Value: 266.33

Outliers in Methane :
Found 1 outlier(s).
Year: 2013 - Value: 188.61

Outliers in NitrousOx :
Found 5 outlier(s).
Year: 2013 - Value: 437.39
Year: 2013 - Value: 456.78
Year: 2013 - Value: 400.55
Year: 2013 - Value: 200.88
Year: 2013 - Value: 200.99

Outliers in CFC11 :
Found 5 outlier(s).
Year: 2013 - Value: 319.26
Year: 2013 - Value: 319.52
Year: 2013 - Value: 330.34
Year: 2013 - Value: 150.21
Year: 2013 - Value: 130.27
```

Table S2. Outliers detected in the data using the IQR rule method. Value represents the value of the outlier, and year represents the year that the outlier was found.

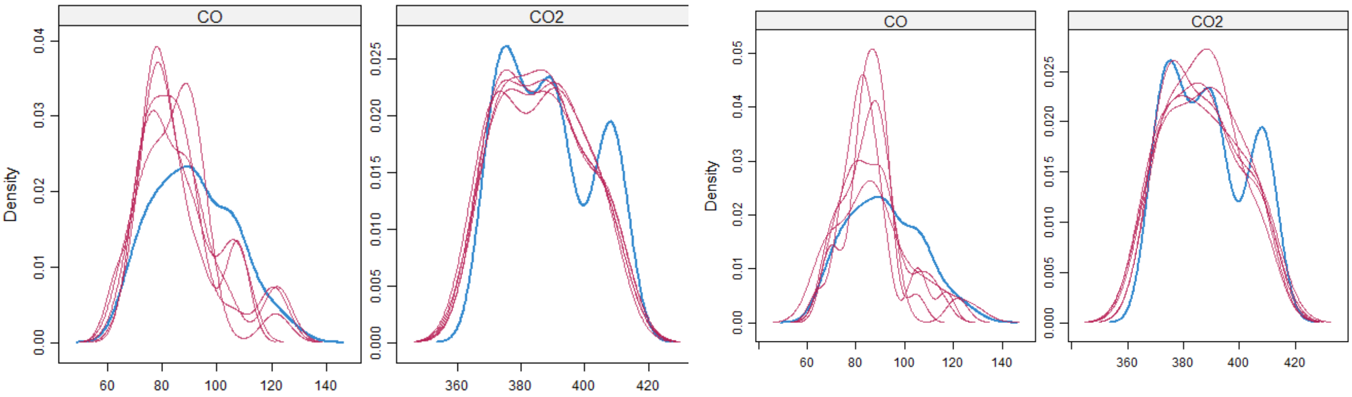
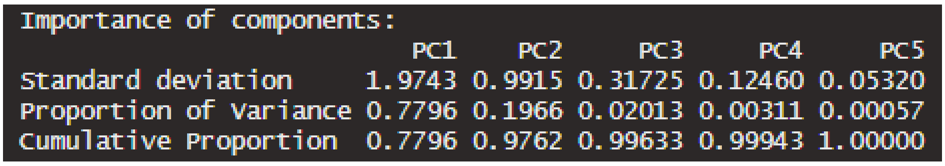


Figure S1. Kernel density estimates of the distributions of imputed data (red) and the observed data (blue), when m = 5. Left = regression imputation, right = random forest imputation. Performed using the mice package in R.



```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5
Standard deviation     1.9743 0.9915 0.31725 0.12460 0.05320
Proportion of Variance 0.7796 0.1966 0.02013 0.00311 0.00057
Cumulative Proportion  0.7796 0.9762 0.99633 0.99943 1.00000
```

Table S3. Proportion of variance, and cumulative proportion of each Principal Component generated during dimensionality reduction.

Bibliography:

Awan, A. A. (2023, March). *Python t-SNE with Matplotlib*. Www.datacamp.com. https://www.datacamp.com/tutorial/introduction-t-sne

GeeksForGeeks. (2020, May 1). *Difference between PCA VS t-SNE*. GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne/

GeeksForGeeks. (2023, December 9). *Demonstration of K-Means Assumptions*. GeeksforGeeks. https://www.geeksforgeeks.org/demonstration-of-k-means-assumptions/

Kaur, T. (2025, February 10). *Understanding Little's MCAR Test: A Key Tool in Missing Data Analysis*. Medium. https://medium.com/@tarangds/understanding-littles-mcar-test-a-key-tool-in-missing-data-analysis-47fd70698149

Lindsey, R. (2024, April 9). Climate Change: Atmospheric Carbon Dioxide. Climate.gov; National Oceanic and Atmospheric Administration. https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide

Mahmood, M. (2024, March 24). *Hard Clustering Vs Soft Clustering in NLP*. Medium. https://medium.com/@mohamad.razzi.my/hard-clustering-vs-soft-clustering-in-nlp-da8f1f2ba2ec

Appendix:

Code is in R notebook format

~~~
---
title: "Supplementary Code"
output: html_notebook
---


```{r}
library(ggplot2)
library(dplyr)
library(naniar)
library(VIM)
library(mice)
library(gridExtra)
library(ks)
library(scales)
library(tidyr)
library(lattice)
library(GGally)
~~~

~~~
library(tidyr)
library(missForest)
library(superMICE)
library(imputeTS)
library(factoextra)
library(ggfortify)
library(cluster)
library(mclust)
```

```{r}
data <- read_csv("C:/Users/benke/OneDrive - Durham University (1)/Uni Work/Master of Data Science/DEVUL/Assessments/Assignment 2/MaunaLoa_miss.csv", col_types = cols(Date = col_date(format = "%Y-%m-%d")))
```

Figure 1:
```{r}
co = ggplot(data = data, aes(x = Date, y = CO)) +
 geom_line() +
 geom_point() +
 theme_bw() +
 labs(title = "CO")

co2 = ggplot(data = data, aes(x = Date, y = CO2)) +
 geom_line() +
 geom_point() +
 theme_bw() +
 labs(title = "CO2")

meth = ggplot(data = data, aes(x = Date, y = Methane)) +
 geom_line() +
 geom_point() +
 theme_bw() +
 labs(title = "Methane")

no2 = ggplot(data = data, aes(x = Date, y = NitrousOx)) +
 geom_line() +
 geom_point() +
 theme_bw() +
 labs(title = "NO2")

cfc = ggplot(data = data, aes(x = Date, y = CFC11)) +
 geom_line() +
 geom_point() +
 theme_bw() +
 labs(title = "Chlorofluorocarbons")

grid.arrange(ncol = 2, co, co2, meth, no2, cfc)
```

Figure 2:
```{r}
temp_clean_data = na.omit(data)

start_year = as.numeric(format(temp_clean_data$Date[1], "%Y"))
start_month = as.numeric(format(temp_clean_data$Date[1], "%m"))
 #I need to make sure I specify the start year and month from within my data

CO2ts = ts(temp_clean_data$CO2,
        start = c(start_year, start_month),
        frequency = 12)

decomp_co2 = stl(CO2ts, s.window = "periodic")

plot(decomp_co2)
```
~~~

Figure 3:

~~~
```{r}
ggpairs(data[, -c(1)]) +theme_bw()
```
~~~

Figure 4:

~~~
```{r}
aggr(data, col = mdc(1:2), numbers = TRUE, softVars = TRUE, labels = names(data), cex.axis = 1, gap = 3, ylab = c("Proportion of Missingness", "Missingness Pattern"))
```
~~~

Figure 5/10:

~~~
```{r}
data$year = format(as.Date(data$Date, format="%Y-%m-%d"),"%Y")
  #Making a column for year

clean_data = data %>%
  filter(year != 2013)

full_data  = clean_data %>%
  complete(Date = seq.Date(from = min(Date),
                           to   = max(Date),
                           by   = "month"))
#The complete() function fills in missing data
#Fillls in rows of data that is missing from the min and max of my data by month
#Maintains ordering

full_data = full_data %>% select(-year)
  #Removing the unneeded year column

ggplot_na_distribution(full_data$CO2)
ggplot_na_distribution(full_data$CO)
  #Distribute the values of missing data well
  #Can see the veryt clearn gaps in the data


statsNA(data$CO)

statsNA(data$CO2)

data_imp = na_kalman(full_data)
  #IMputing the missing values

ggplot_na_imputations(full_data$CO, data_imp$CO)
ggplot_na_imputations(full_data$CO2, data_imp$CO2)
  #PLots hwo the values were imputed
```
~~~

Figure 6:

~~~
```{r}
data_na_colour = data %>%
  mutate(colour_na = ifelse(rowSums(is.na(across(c(1:6)))) == 0, "black", "red"))

par(mfrow = c(2,3))

plot(density(na.omit(filter(data_na_colour, colour_na =="black"))$CO), col = "black", main = "CO")
lines(density(na.omit(filter(data_na_colour, colour_na == "red")$CO)), col = "red")

plot(density(na.omit(filter(data_na_colour, colour_na =="black"))$CO2), col = "black", main = "CO2")
lines(density(na.omit(filter(data_na_colour, colour_na == "red")$CO2)), col = "red")

plot(density(na.omit(filter(data_na_colour, colour_na =="black"))$Methane), col = "black", main = "Methane")
lines(density(na.omit(filter(data_na_colour, colour_na == "red")$Methane)), col = "red")
~~~

~~~
plot(density(na.omit(filter(data_na_colour, colour_na =="black"))$CFC11), col = "black", main = "CFC")
lines(density(na.omit(filter(data_na_colour, colour_na == "red")$CFC11)), col = "red")

plot(density(na.omit(filter(data_na_colour, colour_na =="black"))$NitrousOx), col = "black", main = "NO2")
lines(density(na.omit(filter(data_na_colour, colour_na == "red")$NitrousOx)), col = "red")
```

Figure 7:
```{r}
data_na_colour = data %>%
 mutate(colour_na = ifelse(rowSums(is.na(across(c(1:6)))) == 0, "black", "red"))
 #This creates a neew df with a new column that stores a colour value based on whether NA values are present

co = ggplot(data = data_na_colour, aes(y = CO, x = factor(1))) +
 geom_jitter(color = data_na_colour$colour_na, alpha = 0.5, width = 0.1) +
 theme_bw() +
 labs(title = "CO",
    x = element_blank())

co2 = ggplot(data = data_na_colour, aes(y = CO2, x = factor(1))) +
 geom_jitter(color = data_na_colour$colour_na, alpha = 0.5, width = 0.1) +
 theme_bw() +
 labs(title = "CO2",
    x = element_blank())

meth = ggplot(data = data_na_colour, aes(y = Methane, x = factor(1))) +
 geom_jitter(color = data_na_colour$colour_na, alpha = 0.5, width = 0.1) +
 theme_bw() +
 labs(title = "Methane",
    x = element_blank())

no2 = ggplot(data = data_na_colour, aes(y = NitrousOx, x = factor(1))) +
 geom_jitter(color = data_na_colour$colour_na, alpha = 0.5, width = 0.1) +
 theme_bw() +
 labs(title = "NO2",
    x = element_blank())

cfc = ggplot(data = data_na_colour, aes(y = CFC11, x = factor(1))) +
 geom_jitter(color = data_na_colour$colour_na, alpha = 0.5, width = 0.1) +
 theme_bw() +
 labs(title = "CFCs",
    x = element_blank())

grid.arrange(ncol = 5, co, co2, meth, no2, cfc)

```

Figure 8:
```{r}
marginplot(data[,c("CO", "CO2")], col = mdc(1:2), cex.numbers = 1.2, pch = 19)
```

Figure 9:
```{r}

ggplot(data = data, aes(x = year)) +
 geom_bar(stat = "count") +
 theme_bw() +
 labs(x = "Year",
    title = "Number of Readings Taken per Year") +
 scale_y_discrete(name = "Number of Readings",
          limits = seq(0,12, 2))


data_long = pivot_longer(data, cols = c(2:6), names_to = "Name", values_to = "Values")
~~~

~~~
ggplot(data = data_long, aes(x = year, y = Values)) +
  geom_violin() +
  geom_jitter(alpha = 0.2, aes(color = Name)) +
  facet_wrap(.~Name, scales = "free_y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 6)) +
  theme(legend.position = "none")
```

Figure 10:
```{r}
ggpairs(clean_data[, -c(1,7)]) + theme_bw()
  #Not including the time and motnh in the scatterplot

```

Dimensionality Reduction:

```{r Regression}
na_rows = which(is.na(clean_data[,-c(1,7)]), arr.ind = TRUE)[1:34]

imp_reg = mice(clean_data[,-c(1,7)], m = 10, seed = 42, print = FALSE)
  #This creates 10 imputed datasets

xyplot(imp_reg, CO ~ CO2 | .imp, pch = 20, cex = 1.4)
  #PLotting the imputed values
  # | .imp factes the plot by imputatiib number

densityplot(imp_reg)

imp_reg = complete(imp_reg)
```



```{r Random Forest}

imp_rf = mice(clean_data[,-c(1,7)], method = "rf", m = 5, seed = 42, print = FALSE)
  #This creates 10 imputed datasets

xyplot(imp_rf, CO ~ CO2 | .imp, pch = 20, cex = 1.4)
  #PLotting the imputed values
  # | .imp factes the plot by imputatiib number

densityplot(imp_rf)

imp_rf = complete(imp_rf)
```

```{r PMM}
imp_pmm = mice(clean_data[,-c(1,7)], method = "pmm", m = 5, seed = 42, print = FALSE)
  #This creates 10 imputed datasets

xyplot(imp_pmm, CO ~ CO2 | .imp, pch = 20, cex = 1.4)
  #PLotting the imputed values
  # | .imp factes the plot by imputatiib number

densityplot(imp_pmm)

imp_pmm = complete(imp_pmm)
```

```{r superMice}
columns = c("CO", "CO2")
  #CO has more missing values -> this is being treated first
  #As less regression imputed (the worse imputation) is influencing this
~~~

```
imputed_list = vector("list", 5)
 #To store the dataframes

for (i in 1:5) {
  #Repeating 5 times, equivalent to m = 5

 imp_data = mice(clean_data[,-c(1,7)], m = 1, seed = 42 + i, print = FALSE)
 imp_data = complete(imp_data)
  #Filling missing values using single regression imputation
  #This is required as many SL.library methods require full data (except for the imputtaion column)

  for (col in columns) {
    #Iterating over each of the columns that require imputation (CO and CO2)


   na_index = which(is.na(clean_data[,-c(1,7)]), arr.ind = TRUE)

  #setting NA for target col
   for (j in seq_len(nrow(na_index))) {
    row_num = na_index[j, "row"]
    col_index = na_index[j, "col"]
      #Extracting the rowand column indexes for the na values

    if (colnames(imp_data)[col_index] != col) {
     next
    }

    imp_data[row_num, col_index] = NA
     #Changing back to NA (all fo the values that were NA)
   }

  #imputing the current column using supermice
   imp_data = mice(imp_data,
                   method = "SuperLearner",
                   m = 1,
                   seed = 42 + i,
                    #Varying the seed for variability among iterations
                   print = FALSE,
                   SL.library = c("SL.randomForest", "SL.glm", "SL.glmnet"))
   imp_data = complete(imp_data)
  }

 imputed_list[[i]] = imp_data
  #Save the final imputed data frame for this iteration
}

#Assigning each to a variable for analysis
imp_data_1 = imputed_list[[1]]
imp_data_2 = imputed_list[[2]]
imp_data_3 = imputed_list[[3]]
imp_data_4 = imputed_list[[4]]
imp_data_5 = imputed_list[[5]]

imp_data = imp_data_1 %>%
 mutate(na_row = ifelse(row_number() %in% na_rows, "na_row", "non_na"))
#Assigning values to rows that contain NA values (for analysis of the imputed values)

```

Figure 11/12:

```{r}
par(mfrow = c(1,2))
plot(density(na.omit(clean_data$CO)), col = "blue", main = "CO")
lines(density(imp_data_1$CO), col = "red")
lines(density(imp_data_2$CO), col = "red")
```

~~~
lines(density(imp_data_3$CO), col = "red")
lines(density(imp_data_4$CO), col = "red")
lines(density(imp_data_5$CO), col = "red")

plot(density(na.omit(clean_data$CO2)), col = "blue", main = "CO2")
lines(density(imp_data_1$CO2), col = "red")
lines(density(imp_data_2$CO2), col = "red")
lines(density(imp_data_3$CO2), col = "red")
lines(density(imp_data_4$CO2), col = "red")
lines(density(imp_data_5$CO2), col = "red")


ggpairs(
 imp_data_1 %>% select(-na_row),
  #exclude na_row from the data used in the plot
 mapping = aes(color = imp_data_1$na_row),
  #use it just for coloring
 upper = list(continuous = wrap("cor",
              method = "pearson",
              use = "pairwise.complete.obs")),
 diag = list(continuous = wrap("densityDiag", fill = "grey")),
 lower = list(continuous = wrap("points", alpha = 0.6))) +
 theme_bw()
```
~~~

Dimensionality reduction on the imputed data:

Figure 13

~~~
```{r}
pc_data = imp_data %>%
 select( -c( "na_row"))
 #Selecting just the numeric data for dimensionaltiy reduction


pc_out = prcomp(pc_data, scale = TRUE)

summary(pc_out)

pc_var = pc_out$sdev^2

plot(pc_out, col = "steelblue", main = "Importance of Each PC")
lines(x = 1:5, pc_var, type = "b", pch = 19, col = "red")

ggpairs(pc_out$x) +theme_bw()
```
~~~

Figure 14

~~~
```{r}

autoplot(pc_out, loadings = TRUE, loadings.colour = "steelblue",
     loadings.label = TRUE, col = rgb(0,0,0,0.3)) + theme_bw()

a = fviz_contrib(pc_out, choice = "var", axes = 1, top = 5)
b = fviz_contrib(pc_out, choice = "var", axes = 2, top = 5)

grid.arrange(a,b, ncol = 2)
```
~~~

Figure 15

~~~
```{r fig.width=10, fig.height=6}

sel_pc = cbind(pc_out$x[,1], pc_out$x[,2], clean_data["Date"])
 #Extracting the values of the first PC and the date
colnames(sel_pc) = c("PC1", "PC2", "Date")
~~~

~~~
start_year = as.numeric(format(sel_pc$Date[1], "%Y"))
start_month = as.numeric(format(sel_pc$Date[1], "%m"))
 #I need to make sure I specify the start year and month from within my data

PC1ts = ts(sel_pc$PC1,
        start = c(start_year, start_month),
        frequency = 12)

decomp_pc1 = stl(PC1ts, s.window = "periodic")

PC2ts = ts(sel_pc$PC2,
        start = c(start_year, start_month),
        frequency = 12)

decomp_pc2 = stl(PC2ts, s.window = "periodic")


a = autoplot(decomp_pc1) + theme_bw() +labs(y = "PC1")
b = autoplot(decomp_pc2) + theme_bw() + labs(y = "PC2")
fig.dim = c(8, 20)
grid.arrange(a, b, ncol = 2)
```
~~~

Cluster Analysis

Figure 16

~~~
```{r}
pc_scaled = scale(sel_pc[,-3])
 #Scaling the data
 #To account for potentially high variance

fviz_nbclust(pc_scaled, kmeans, method = "silhouette") +
 labs(title = "K-means Sillhouette Scores")
fviz_nbclust(pc_scaled, clara, method = "silhouette") +
 labs(title = "PAM")
fviz_nbclust(pc_scaled, hcut, method = "silhouette") +
 labs(title = "Hierarchal")

fviz_nbclust(pc_scaled, kmeans, method = "wss") +
 labs(title = "wss")
 #This does not have a major impact on the results (I would still have chosen 3)
```
~~~

Figure 17

~~~
```{r}
results = list()
 #To store the results
 #Use reults[[]] = , to store


print(results)

for (i in  1:10){
 #Iterating over every number of cluster

 temp_clus = kmeans(pc_scaled, i, nstart = 25)

 var_exp = temp_clus$betweenss / temp_clus$totss

 results[[i]] = var_exp
}


results = data.frame(results)
colnames(results) = c(1:10)
~~~

~~~
results = pivot_longer(results, cols = everything())
 #Converting to long formatting for plotting using ggplot

ggplot(data = results, aes(x = as.numeric(name), y = value, fill = value)) +
 geom_bar(stat = "identity") +
 theme_bw() +
 scale_x_discrete(limits = c(1:10)) +
 labs(x = "Value of K",
      y = "Propotion of Variance Explained") +
 theme(legend.position = "none")
```
~~~

Figure 18:

~~~
```{r}
km_clus = kmeans(pc_scaled, 3, nstart = 25)

var_exp = km_clus$betweenss / km_clus$totss
print(paste0("The variance explained by these clusters are: ", var_exp))

fviz_cluster(km_clus, pc_scaled, ellipse.type = "norm") + theme_bw() + labs(title = "K-means Cluster Plot")
```
~~~

Figure 19:

~~~
```{r}
gc_clus = Mclust(pc_scaled)

gc_clus$BIC

plot(gc_clus, what = "BIC")
```
~~~

Figure 20/21

~~~
```{r}
eee_clus = Mclust(pc_scaled, G = 3, modelNames = "EEE")

plot(eee_clus, what = "classification")
plot(eee_clus, what = "density")
```
~~~

Figure 22:

~~~
```{r}
time_km = data.frame(km_clus$cluster)
time_km = cbind(time_km, clean_data$Date)
colnames(time_km) = c("Cluster", "Date")

ggplot(data = time_km, aes(x = Date, y = Cluster)) +
 geom_point(aes(col = as.factor(Cluster))) +
 geom_line(alpha = 0.3) + theme_bw() +
 theme(legend.position = "none") +
 scale_y_discrete(limits = c(1:3))

class(time_km$Date)

time_gs = data.frame(eee_clus$classification)
time_gs = cbind(time_gs, clean_data$Date)
colnames(time_gs) = c("Cluster", "Date")

ggplot(data = time_gs, aes(x = Date, y = Cluster)) +
 geom_point(aes(col = as.factor(Cluster))) +
 geom_line(alpha = 0.3) + theme_bw() +
 theme(legend.position = "none") +
 scale_y_discrete(limits = c(1:3))

```
~~~

Figure 23

~~~~
```{r}
na_rows = na_index[, "row"]
 #This is all of the rows that contained NA values

to_pc = function(input_data){
 pc = prcomp(input_data, scale = TRUE)
  #Converts to PC

 pc = cbind(pc$x[,1], pc$x[,2])
 colnames(pc) = c("PC1", "PC2")
  #Extracts the PCs

 pc = data.frame(pc)

 pc = pc %>%
    mutate(was_na = ifelse(row_number() %in% na_rows, "na_row", "non_na"))
 #Variable assigned to rows that had values imputed

 pc = pc %>%
  filter(was_na == "na_row")
  #Only selecting the coords that had values imputed

 pc$number = 1:nrow(pc)

 return(pc)
}

temp = sel_pc %>%
 mutate(was_na = ifelse(row_number() %in% na_rows, "na_row", "non_na"))

temp$M = 1
 #Saving the number of the imputed dataset (1-5)

all_data = temp %>% select(- Date) %>%
 filter(was_na == "na_row")
 #This will store all of the coordinates for the NA values
 #For every imputed dataset

all_data$number = 1:nrow(all_data)

temp = to_pc(imp_data_2)
 #This will extract the PC's
temp$M = 2
all_data = rbind(all_data, filter(temp, was_na == "na_row"))

temp = to_pc(imp_data_3)
temp$M = 3
all_data = rbind(all_data, filter(temp, was_na == "na_row"))

temp = to_pc(imp_data_4)
temp$M = 4
all_data = rbind(all_data, filter(temp, was_na == "na_row"))

temp = to_pc(imp_data_5)
temp$M = 5
all_data = rbind(all_data, filter(temp, was_na == "na_row"))

imputed = all_data %>%
 filter(M == 1) %>%
 select(number, imputed_x = PC1, imputed_y = PC2)
 #This selects the points I used in my analysis
 #Where M = 1

others = all_data %>%
~~~~

~~~
 filter(M %in% 2:5)
 #Selecting all the other points generated during my multiple imputation

segments_data = others %>%
 left_join(imputed, by = "number")
 #Joining the two together


ggplot(data = filter(all_data, M != 1), aes(x = PC1, y = PC2, color = factor(M))) +
 geom_point(alpha = 0.4, size = 2) +
 geom_point(data = filter(all_data, M == 1), aes(x = PC1, y = PC2), fill = "blue" , size = 3, alpha = 0.8, shape = 24) +
 geom_segment(data = segments_data,
        aes(x = PC1, y = PC2, xend = imputed_x, yend = imputed_y),
        arrow = arrow(length = unit(0.15, "cm")),
        color = "blue", linewidth = 0.5, linetype = "dashed",
        alpha = 0.4) +
  #Adding a line connecting the point I used and the other points from M
 theme_bw() +
 labs(color = "Imputed Dataset") +
 theme(legend.position = "bottom")


```
~~~

Figure 24:

~~~
```{r}
pc = prcomp(select(data_imp, -c(1)), scale = TRUE)
pc = cbind(pc$x[,1], pc$x[,2])
colnames(pc) = c("PC1", "PC2")
pc = data.frame(pc)
 #Dataframe of the PC's

pc_scaled = scale(pc)

gc_time = Mclust(pc_scaled)

gc_time$BIC

plot(gc_time, what = "BIC")

best_time_clus = Mclust(pc_scaled, G = 7, modelNames = "VVV")

plot(best_time_clus, what = "classification")
plot(best_time_clus, what = "density")

```
~~~