Final_Project_STA135

Mark Faynboym, Vincent Barletta, Christian(Ian) Dimapasok, Jared Woolsey

2023-06-04

```
#Cities with the top SO2 levels
head(USairpollution[order(USairpollution$SO2, decreasing=TRUE),])
```

▸
Chicago
Providence
Philadelphia
Cleveland
Pittsburgh
Hartford

6 rows | 1-1 of 8 columns

```
dim(USairpollution)
```

```
## [1] 41  7
```

1 Introduction

Air pollution is a concerning global environmental issue that poses a significant risk to human health and ecosystems as a whole. It is caused by the release of harmful substances into the atmosphere, resulting from various human activities, industrial processes and natural sources. As the detrimental effects of air pollution continue to emerge, researchers and policymakers are increasingly relying on statistical analysis to gain a deeper insight into the patterns, sources and impacts involved.

What this project aims to delve into is the exploration and analysis of air pollution with data found in the HSAUR2 package. Mainly we will be looking into several main characteristics of large cities and with the use of Sulfur Dioxide as the measure of air pollution we will analyze the influence of each of the features involved.

Question of Interest: In this report we analyze the characteristics of 41 major cities to determine which of the provided six climate and ecological variables are the most significant in determining air pollution. Here we use the annual mean concentration of sulphur dioxide (SO2) as a measure of the air pollution of a city.

Impact of result: Identifying key causes of air pollution can influence future policies implemented in larger cities. - Insights gained can inform policymakers and government agencies in designing and implementing certain policies to combat air pollution. - By understanding factors influencing pollution levels, healthcare professionals can better educate the public about associated health risks. - By identifying the sources and factors contributing to pollution, environmental scientists can devise strategies to minimize the impact on wildlife of affected ecosystems which can in turn contribute to the conservation and restoration of natural habitats and promote sustainable development practices. - This can also influence urban planning decisions and infrastructure development, particularly in areas affected by high pollution levels. By understanding these factors, things like public transportation systems and shifts to cleaner energy sources could be a focus for change.

Overall, the impact of project such as this lies in its potential to inform decision-making processes, drive policy change and bring overall awareness to health and environmental changes brought about by the shifting climate.

2 Background: Data Overview

Introduce variables

Data frame with 41 observations on the following 7 variables

SO2 - Sulfur dioxide concentration in air (micrograms per cubic meter)

temp - average annual temperature in Farenheit

manu - number of manufacturing enterprises employing 20+ workers

popul - population size(1970 census); (in thousands)

wind - average annual wind speed (miles per hour)

precip - average annual precipitation in inches.

predays - average number of days with precipitation per year.

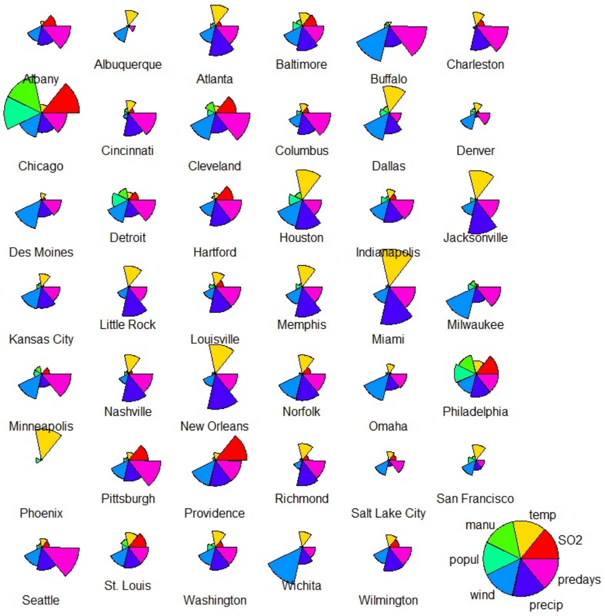
Source: https://search.r-project.org/CRAN/refmans/HSAUR3/html/USairpollution.html (https://search.r-project.org/CRAN/refmans/HSAUR3/html/USairpollution.html)

3 Exploratory Descriptive analysis (EDA)

Before we delve into supervised and unsupervised learning methods, we want to get a surface level look at our data in our EDA. First, we will begin by looking at icon-based analysis for each city. In our first introductory lesson on multivariate data analysis, we were introduced to this method to get basic observations of the overall data distribution, without necessarily trying to abstract correlational tendencies.

3.1 Icon-based analysis

```
df = USairpollution

palette(rainbow(7)) # set colors
stars(df, len=1, cex=0.5, key.loc=c(15, 2),
       labels=row.names(USairpollution), draw.segments=TRUE)
```



Let's pull some basic takeaways for cities of interest that we will want to observe later during higher level analysis.

At first site, in terms of SO2 levels, our highest offenders are: Chicago, Providence, Cleveland, Philadelphia, and Pittsburgh. Given that Chicago has both the largest manufacturing and population of all cities in this dataset, it may be indicative of the level of significance of those two variables as it also has the highest SO2 levels. Cleveland, Philadelphia, and Pittsburgh are also much larger cities and more densely populated manufacturing hubs in the Rust Belt which further confirms our inference.

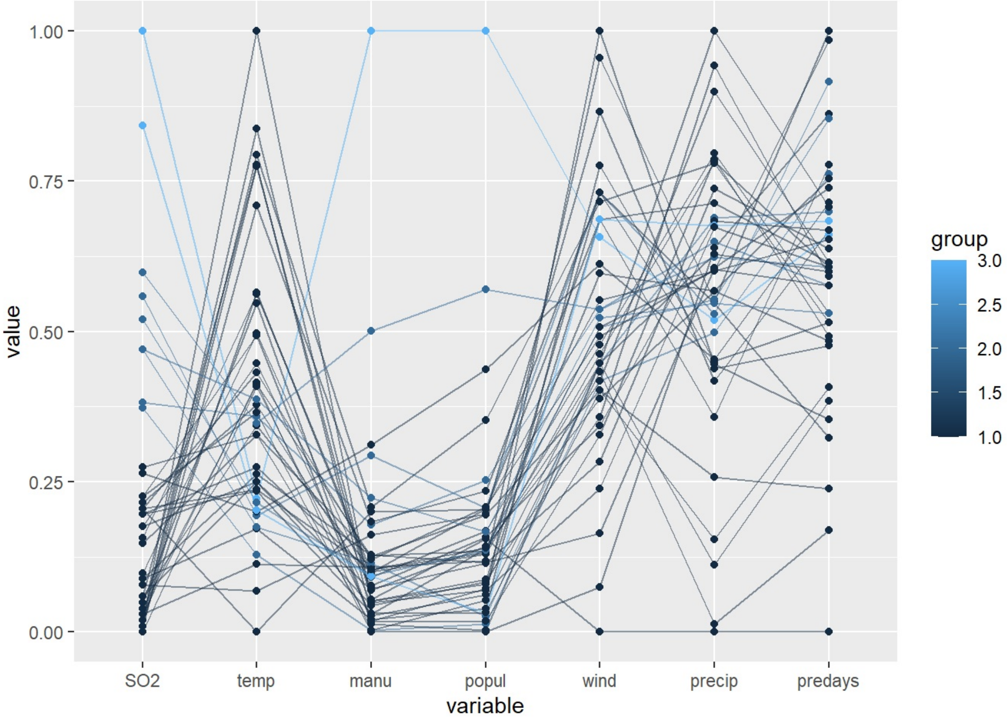
Providence, however, sticks out as the city with the second highest SO2 concentration rate. With a very small population and manufacturing arm at the very edge of New England, there may be other variables that are influencing such a high SO2 level. We will explore further.

3.2 Parallel coordinates graph

Next, we will look at a parallel coordinates graph.

```
#First, we scale all of our columns so that they are on similar grounds for graphing. GGally does this for us autom
atically, but it is good for factoring regardless.

cols <- c("SO2", "temp", "manu", "popul", "wind", "precip", "predays")
df2 <- as.data.frame(scale(USairpollution[, cols]))
```

```
num_groups <- 3
normalized_data = df2
normalized_data$group <- cut(normalized_data$SO2, breaks = num_groups, labels = FALSE)

ggparcoord(normalized_data, columns = 1:7,
           groupColumn = "group",
           scale = "uniminmax",
           alphaLines = 0.5,
           showPoints = TRUE)
```

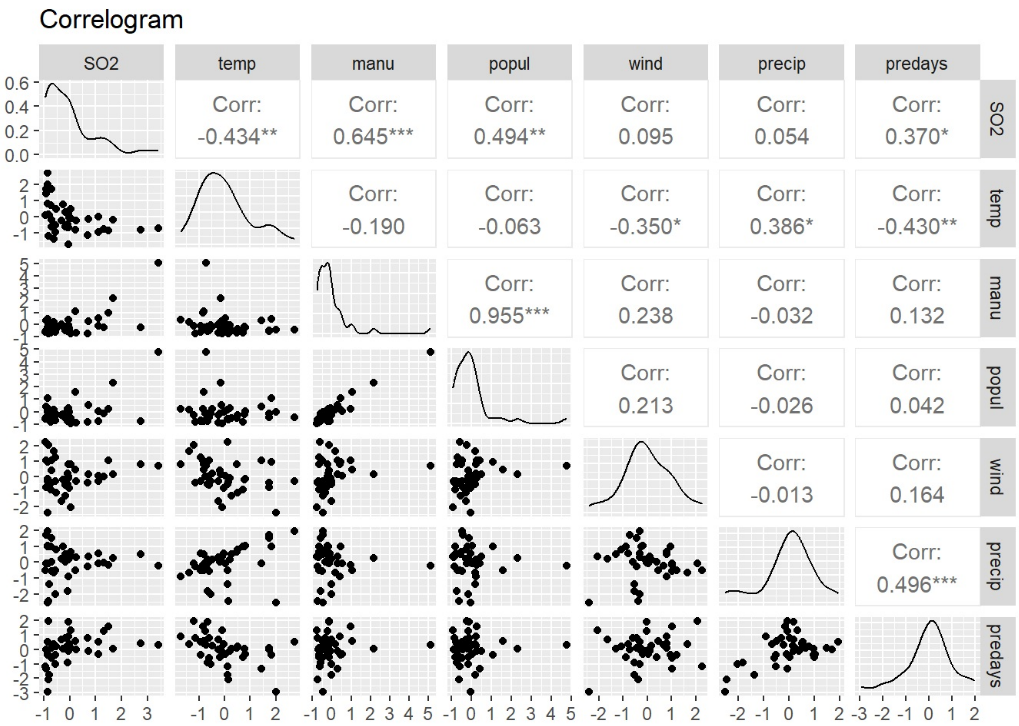


While it is not as intuitive as the iconography, we can clearly see a relationship between SO2, manufacturing and population size. We delineate the data into groups based on their SO2 values; we picked to do three groups as the data distribution looked most fair using cut().

#correlogram

Next, we will look a correlogram and histograms of all features to display the correlation between the different variables.

```
ggpairs(df2, title = "Correlogram", progress = FALSE)
```



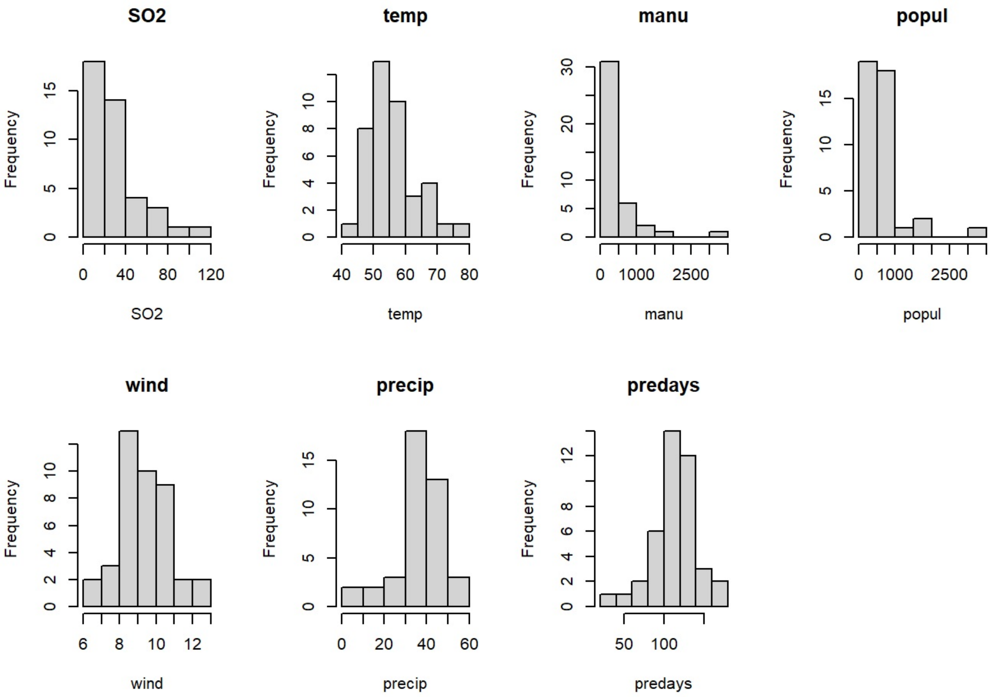
When comparing SO2 levels to each of our features, we notice that there is a strong positive correlation with manufacturing and population and a strong negative correlation with temperature. This further confirms our earlier analyses and is of no surprise. Some other interesting insights is that there is an extremely high collinearity between manufacturing and population. In reality, it makes sense to have such an observation as manufacturing within a given city will tend to increase with population size, however population size can have effects on SO2 levels outside of manufacturing and the same can be said of manufacturing. Without further industry knowledge we cannot give further insight.

What we should also keep in mind is that though precipitation in inches(*precip*) looks to be less correlated with SO2 levels than number of days of precipitation (*predays*), there is also a strong correlation between the two predictor variables as both are measures of precipitation. What may be happening here is that the effects of one variable is masking the other on our response variable as there is collinearity between the two predictors.

3.3 Histogram

Next we briefly take a look at the distributions of each of our variables.

```
#create histograms of scaled data
par(mfrow = c(2, 4))
for (i in 1:7) {
  hist(df[[i]], main = cols[i], xlab = cols[i])
}
```



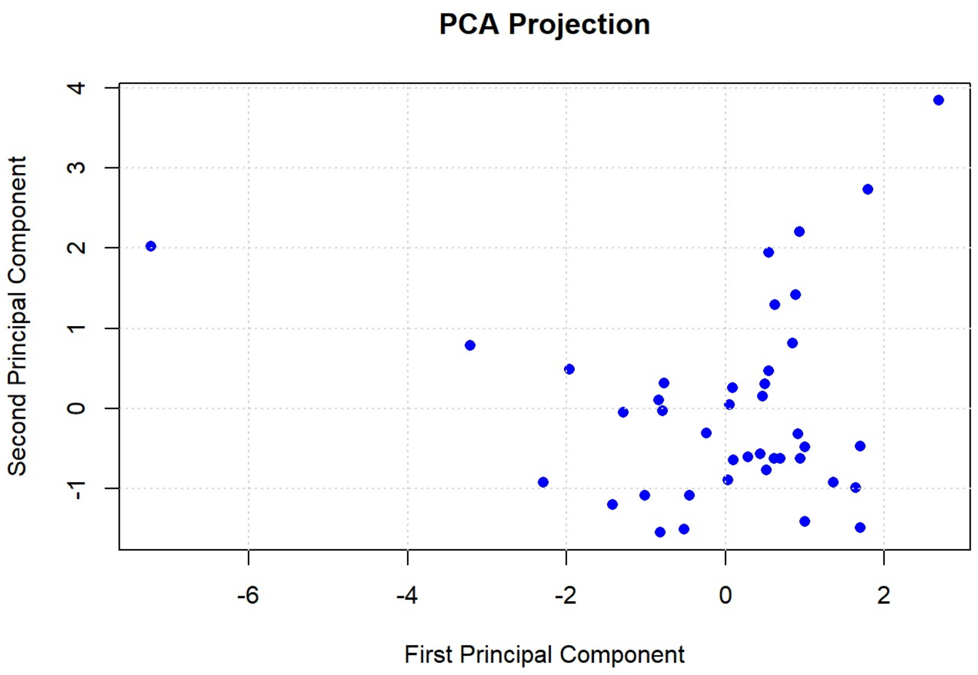
Based on our histograms, we see a similar distribution between SO2, manufacturing and population size and an opposite skew with precipitation and predays which can indicate positive correlation of the variables with a similar histogram skew and a negative correlation with variables of the opposing skew.

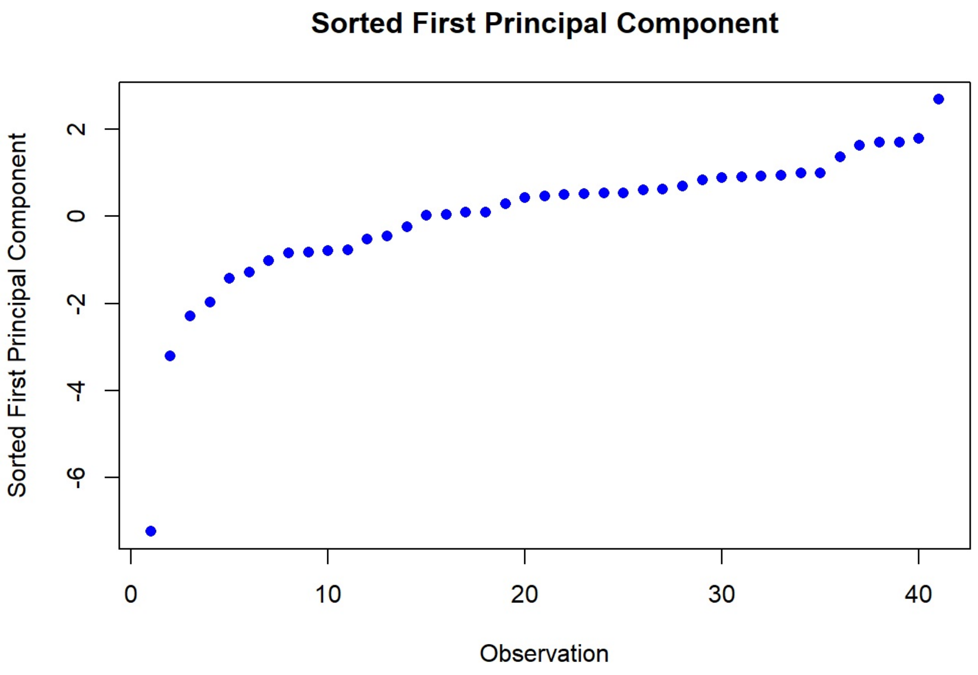
4 Method Selection: PCA

When performing PCA, it is important to consider several assumptions that are required for this method.

1. PCA relies on the assumption of correlation among variables within the data set.
2. The scale of variables in the data set can impact the results of PCA, making it sensitive to scale variations.
3. PCA assumes a linear relationship between features.
4. Outliers in the data can affect the performance of PCA.

```
suppressPackageStartupMessages(library(factoextra))


# Eigenvalues and eigenvectors
eigen_val = eigen((cov(df2)))$values
eigen_vec = eigen(cov(df2))$vectors

# First two eigenvectors multiplied by transpose of data set
l_1 = eigen_vec[1:7,1] %*% t(df2)
l_2 = eigen_vec[1:7,2] %*% t(df2)

# Plots
# Plot 1: PCA Projection
plot(-l_1[1, 1:41], l_2[1, 1:41], xlab = "First Principal Component", ylab = "Second Principal Component", main = "
PCA Projection", pch = 16, col = "blue", grid(lty = "dotted"))

# Plot 2: Sorted Values
plot(sort(-l_1[1, 1:41]), xlab = "Observation", ylab = "Sorted First Principal Component",
     main = "Sorted First Principal Component", pch = 16, col = "blue",grid(lty = "dotted"))
```





We first performed matrix multiplication between our first and second eigenvectors and the transpose of our data set in order to project our data into the eigenvectors. The eigenvectors are sorted in descending order based on their eigenvalues, where the maximum eigenvalue shows the most variance within the data. By doing this, we are trying to capture the primary and secondary sources of our data.

In order to correctly judge the performance of multiple predictor variables on SO2 levels, we have to remove SO2 from our dataset. We can use the resulting performance model in comparison to the overall SO2 distribution to see its accuracy.

We will now go through the multiple methods provided in lecture to determine the number of principal components we want in our model.

```
df3 = df2
df3 = df3[-1]

corr_matrix = cor(df3)
val = eigen(corr_matrix)$values
mean_eigenvalues = mean(val)
vec = eigen(corr_matrix)$vectors
```

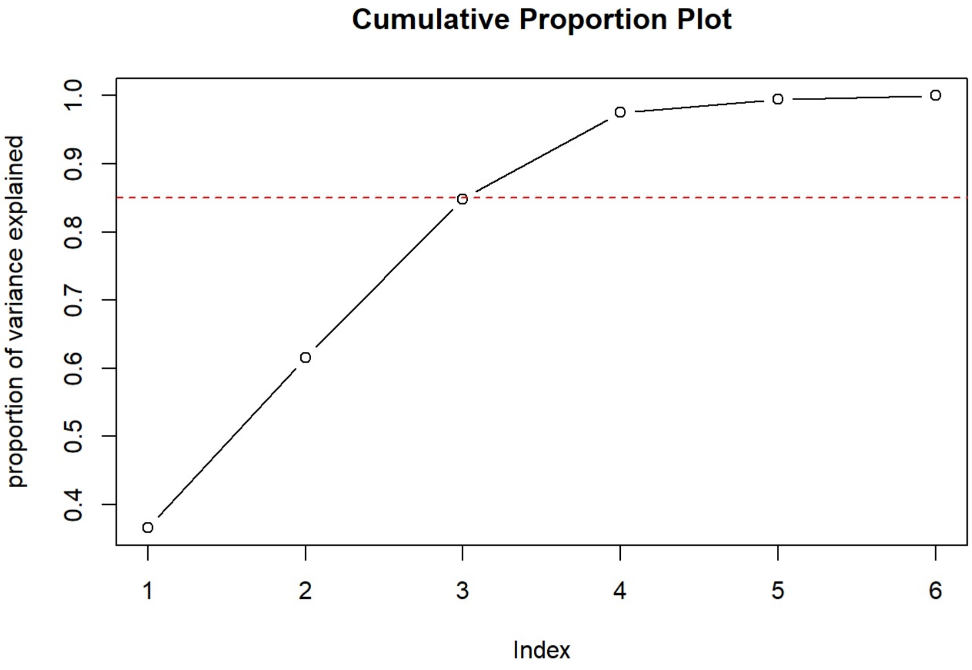
Method one: Contribution ratio approach

We will retain sufficient components to account for a specified percentage of the total variance, e.x. 85%.

```
# Cumulative Proportion for each Component
cumsum(val)/6
```

```
## [1] 0.3660271 0.6160177 0.8484592 0.9751637 0.9942588 1.0000000
```

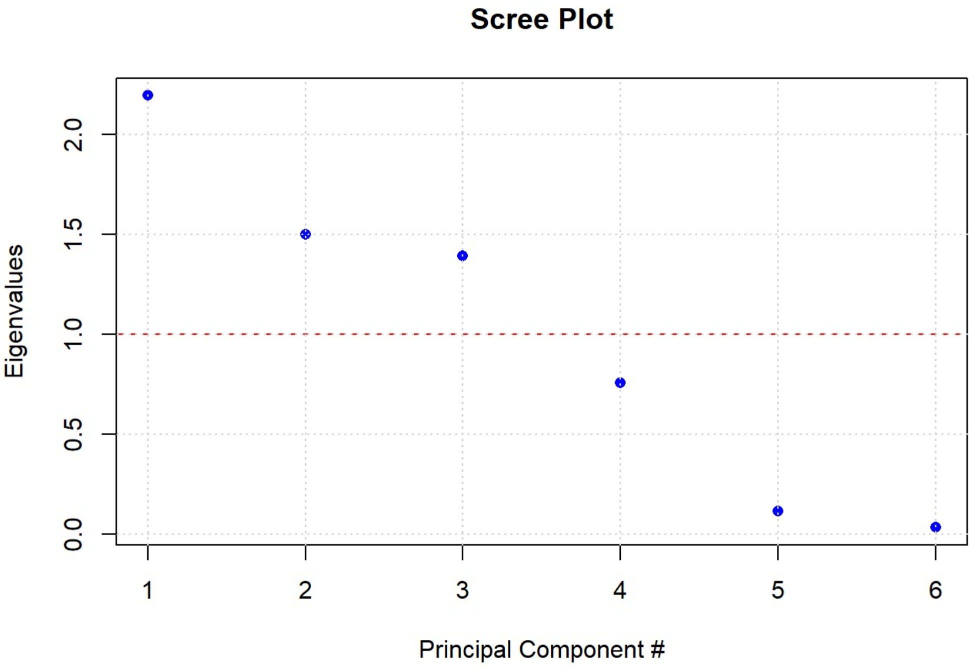
```
# Cumulative Proportion Plot
plot(cumsum(val)/6, type="b", main="Cumulative Proportion Plot",ylab="proportion of variance explained")
abline(h = 0.85, col = "red", lty = "dashed") # 85% of total variance
```



Based on the scree plot above, we see that the first three PCs account for just over 85% of the variance.

Method two: Overall threshold We want to retain the components whose eigenvalues are greater than the average of the eigenvalues.

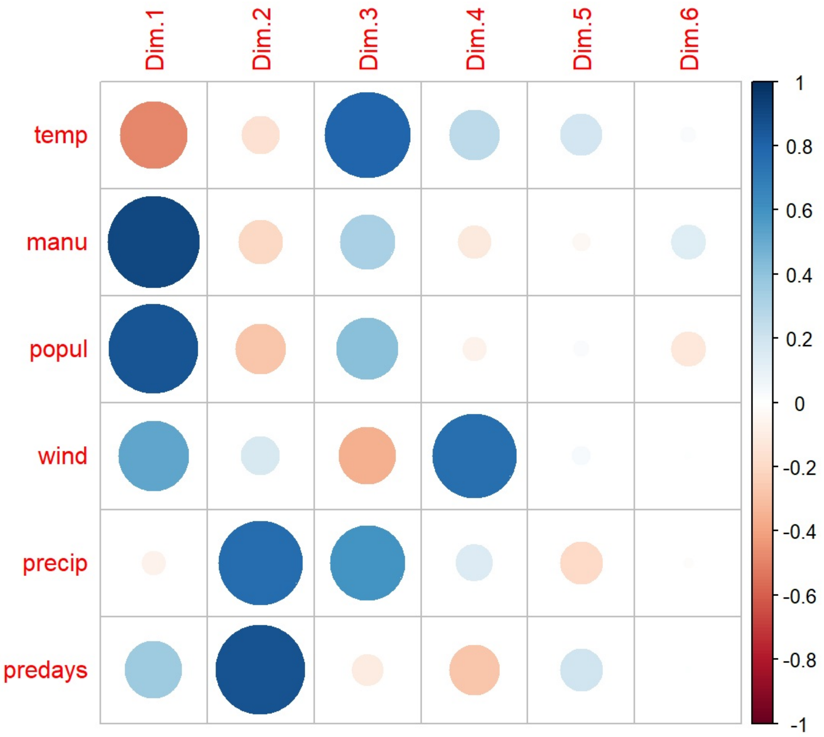
```
# Scree Plot with Eigenvalues
plot(val, pch = 16, col = "blue", xlab = "Principal Component #",
     ylab = "Eigenvalues", main = "Scree Plot")
abline(h = mean(val), col = "red", lty = "dashed") # Add a horizontal line at the maximum eigenvalue
grid(col = "lightgray", lty = "dotted") # Add gridlines
```



In the above graph, we see that the mean of eigenvalues fits squarely in between Lambda_3 and Lambda_4. With the two tests in agreement, we therefore conclude that we only need to use the first three PCs.

Our last method discussed in class for choosing PCs is to use a heatmap correlation plot.

```
df3.pca <- PCA(df3)
corrplot(df3.pca$var$cor,is.corr=TRUE)
```



Here, we can see that the first three PCs contain almost all (85%) of the necessary weights needed for our model. Interestingly, the highly correlated variables are almost all positive, except for temperature.

```
df3.pca$var$coord
```

```
##                Dim.1      Dim.2      Dim.3       Dim.4       Dim.5        Dim.6
## temp    -0.48851762 -0.1562713  0.7932295  0.26720314  0.18889252  0.025276422
## manu     0.90627258 -0.2058239  0.3222658 -0.11931281 -0.03453951  0.130471152
## popul    0.85630067 -0.2724433  0.4137753 -0.06319713  0.02642384 -0.128925256
## wind     0.52436980  0.1601832 -0.3510421  0.75806100  0.03833890  0.004551836
## precip  -0.06047377  0.7628275  0.5958649  0.14922586 -0.19232035 -0.011251469
## predays  0.35257846  0.8668156 -0.1099331 -0.27143159  0.19632137  0.004075886
```

Here, we can see the weights for our graph. Manufacturing and population, as expected, have very high weights on the first dimension. The first dimension is the most important for our performance model. As shown before, it has the largest eigenvalue and contributes the largest amount to the cumulative variance. In the calculation below for our performance model, we multiply the first three dimensions by their respective eigenvalues, and then sum them together. As $_1 > _2 > _3 $, Dimension 1 has the largest weight on the overall model indicating that manufacturing and population carry the most influence in determinging SO2.

```
PC_obs = df3.pca$ind$coord
PC_obs_df = as.data.frame(PC_obs)

val = df3.pca$eig[, 1]

# Ranking each city
rank_1 = sqrt(val[1]) * PC_obs_df[1]
rank_2 = sqrt(val[2]) * PC_obs_df[2]
rank_3 = sqrt(val[3]) * PC_obs_df[3]

total_rank = rank_1 + rank_2 + rank_3
colnames(total_rank) = "Performance"
sorted = total_rank[order(-total_rank$Performance), , drop = FALSE]
sorted
```

▸
Chicago
Philadelphia
Cleveland
Houston
Detroit

Miami
Buffalo
Seattle
Jacksonville
New Orleans

1-10 of 41 rows | 1-1 of 2 columns Previous **1** 2 Next

These performance rankings align greatly with the analysis made during EDA and the overall individual iconography graphics. Based on our heatmap, our PCA model strongly favors high population and manufacturing metrics; with a small negative weighting for high temperatures.

As a result, we find Phoenix in dead last as it is a city with very low metrics across the board except for a high temperature. Chicago, which has very high SO2, manufacturing, and population metrics, tops the board by a wide margin.

Something particularly interesting is the placement of Providence. As highlighted during the EDA section, Providence does not have high population or manufacturing metrics, but very high SO2. It is a possibility that there are other factors not included in our original data which may be contributing to this phenomenon. As a result, its performance ranks much lower than its SO2 value as it does not fit the mold of the model.

Overall in terms of predicting the lowest and highest rankings in air pollution, the predictions used by our dimension-reduction method seems to hold accurate in comparison to the true observations.

5 Model Fitting for comparison

We then test our findings by introducing a supervised method.

5.1 Mention of stepAIC

Originally we had run a stepAIC method which gave us the optimal set of features to be manufacturing and population. However this was based on a linear model.

```
# Model selection (stepAIC)
#fits a full 3rd order model with interaction terms
full_model = lm(SO2~.^3 , data = df2)
null_model = lm(SO2~1, data=df2)

# Forward-stepwise (start with null model, at each step consider adding or dropping variable)
step.aic = stepAIC(null_model, scope=list(upper=full_model, lower=null_model), direction='both', k=2, trace=FALSE)
step.aic$anova
```

Step <chr>
+ manu
+ popul
+ predays

4 rows | 1-1 of 6 columns

```
summary(step.aic)
```

```
##
## Call:
## lm(formula = SO2 ~ manu + popul + predays, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25444 -0.36009  0.01099  0.35369  2.12311
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.253e-16  1.004e-01   0.000  1.00000
## manu         1.784e+00  3.617e-01   4.934 1.73e-05 ***
## popul       -1.219e+00  3.588e-01  -3.396  0.00165 **
## predays      1.856e-01  1.071e-01   1.734  0.09129 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6431 on 37 degrees of freedom
## Multiple R-squared:  0.6174, Adjusted R-squared:  0.5864
## F-statistic:  19.9 on 3 and 37 DF,  p-value: 7.542e-08
```

Our stepAIC method gives the greatest significant to manufacturing and population as we expected given our EDA.

5.2 Feature interaction

We further explore a more complex model. We begin with the use of an interactive model which differs from our PCA in the sense that we can integrate interaction terms wheras PCA assumes a linear relationship between features.

```
 #final model, we test variables on original data. As we see, the best model
 #in terms of Adj R-Squared has very many terms and is not very interpretable
model1 = lm(SO2 ~ temp + manu + popul + wind + precip + predays + temp:manu +
    temp:popul + temp:wind + temp:predays + manu:wind + manu:precip +
    manu:predays + popul:wind + popul:precip + popul:predays +
    wind:precip + wind:predays, data = df)
summary(model1)
```

```
##
## Call:
## lm(formula = SO2 ~ temp + manu + popul + wind + precip + predays +
##     temp:manu + temp:popul + temp:wind + temp:predays + manu:wind +
##     manu:precip + manu:predays + popul:wind + popul:precip +
##     popul:predays + wind:precip + wind:predays, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6060  -5.0022   0.5262   4.1875  21.9881
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.166e+02  2.430e+02   2.537  0.01878 *
## temp          -9.408e+00  3.908e+00  -2.407  0.02491 *
## manu          -1.915e-01  2.633e-01  -0.727  0.47484
## popul          1.820e-01  2.073e-01   0.878  0.38946
## wind          -3.797e+01  2.486e+01  -1.527  0.14098
## precip        -6.357e+00  2.665e+00  -2.385  0.02610 *
## predays       -3.118e-01  1.281e+00  -0.243  0.81004
## temp:manu     -5.036e-03  3.734e-03  -1.349  0.19116
## temp:popul     3.847e-03  3.139e-03   1.225  0.23338
## temp:wind      4.974e-01  3.925e-01   1.267  0.21839
## temp:predays   2.580e-02  1.786e-02   1.445  0.16264
## manu:wind      5.545e-02  1.676e-02   3.309  0.00319 **
## manu:precip    5.083e-03  2.499e-03   2.034  0.05418 .
## manu:predays  -1.676e-03  9.376e-04  -1.787  0.08765 .
## popul:wind    -4.892e-02  1.430e-02  -3.420  0.00245 **
## popul:precip  -3.632e-03  2.278e-03  -1.594  0.12511
## popul:predays  1.483e-03  9.815e-04   1.511  0.14499
## wind:precip    7.764e-01  2.940e-01   2.641  0.01493 *
## wind:predays  -1.213e-01  8.237e-02  -1.473  0.15497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.59 on 22 degrees of freedom
## Multiple R-squared:  0.8881, Adjusted R-squared:  0.7966
## F-statistic: 9.705 on 18 and 22 DF,  p-value: 1.094e-06
```

```
 #This reduced model is reduced so that all included variables are individually
 #significant, but has lower adj R-Squared overall
model2 = lm(SO2~ popul+temp+manu+wind*precip, data = df)
summary(model2)
```

```
##
## Call:
## lm(formula = SO2 ~ popul + temp + manu + wind * precip, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.597  -6.451  -0.272   6.821  34.944
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 292.88863   62.55858   4.682 4.42e-05 ***
## popul        -0.04231    0.01312  -3.225 0.002783 **
## temp         -1.76199    0.41023  -4.295 0.000138 ***
## manu          0.06614    0.01363   4.853 2.66e-05 ***
## wind        -22.07602    5.79352  -3.810 0.000555 ***
## precip       -4.22018    1.37690  -3.065 0.004244 **
## wind:precip   0.55958    0.16449   3.402 0.001727 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.66 on 34 degrees of freedom
## Multiple R-squared:  0.7527, Adjusted R-squared:  0.709
## F-statistic: 17.25 on 6 and 34 DF,  p-value: 4.865e-09
```

As we can see from the summary table above, we have population, temperature manufacturing wind and precipitation to be significant as well as two interaction terms that help us explain 73.1% of our variance shown in the adjusted R^2. This confirms our original inferences and analyses done in both the EDA and PCA.

6 Discussion

We looked into the USairpollution dataset to analyze which of our 6 features were the most significant on levels of SO2 which we deemed to be our measure of air pollution. What we notice from PCA was that population, manufacturing, temperature and wind had the greatest contribution which we confirm through our F tests in the summary of our final linear regression model as well as our EDA. We noted that there do exist certain outliers like Providence which we observed to have high SO2 while having lower values of the aforementioned significant variables do to possible incomplete information. Also, we drew attention to the likelihood that days of precipitation and precipitation in inches had high collinearity which can render one variable insignificant with the presence of the other.

Based on our findings and the ranking of each city based on our principal components, here are some suggestions for future research and policy making.

First, since we find that Chicago, Philadelphia, and Cleveland were ranked the highest based on our components, we believe that it would beneficial to analyze and identify the possible key contributors to air pollution in these cities.

One suggestion would be to monitor the emissions from manufacturing and power-generating industries as well as vehicle emissions in these 3 cities. Doing this can help policymakers identify the main causes of air pollution and develop new policies in order to reduce pollution level.

Furthermore, we can study the lower ranked cities such as Salt Lake City, Albuquerque, and Phoenix to identify successful and policies that have led to cleaner air and reduced air pollution. We can share these findings with the cities with high pollution levels to guide decision-making and implement similar measures.

Lastly, we can explore more factors that impact the levels of air pollution in the city such as geographical climate, industrial composition, climate, and transportation infrastructure. By understanding these factors, policymakers can then find solutions that are tailored towards their specific city.

7 Contributions

Mark Faynboym: Introduction, Data explanation, EDA, ModelFitting, Conclusion Vincent Barletta: Introduction, EDA, PCA, Conclusion Christian(Ian) Dimapasok: PCA Jared Woolsey: Model Fitting, Discussion, Debugging

Session info

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.utf8 
## [2] LC_CTYPE=English_United States.utf8   
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.utf8    
## 
## attached base packages:
## [1] tools     stats     graphics  grDevices utils     datasets  methods  
## [8] base     
## 
## other attached packages:
##  [1] factoextra_1.0.7        fdm2id_0.9.8            FactoMineR_2.7         
##  [4] arulesViz_1.5-2         arules_1.7-6            Matrix_1.5-3           
##  [7] gplots_3.1.3            corrplot_0.92           viridis_0.6.2          
## [10] viridisLite_0.4.1       GGally_2.1.2            lubridate_1.9.2        
## [13] forcats_1.0.0           stringr_1.5.0           dplyr_1.1.0            
## [16] purrr_1.0.1             readr_2.1.4             tidyr_1.3.0            
## [19] tibble_3.1.8            ggplot2_3.4.1           tidyverse_2.0.0        
## [22] HSAUR2_1.1-19           ResourceSelection_0.3-5 MASS_7.3-58.2          
## 
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7           RColorBrewer_1.1-3     bslib_0.4.2           
##  [4] utf8_1.2.3             R6_2.5.1               DT_0.27               
##  [7] KernSmooth_2.23-20     colorspace_2.1-0       nnet_7.3-18           
## [10] withr_2.5.0            tidyselect_1.2.0       gridExtra_2.3         
## [13] emmeans_1.8.4-1        compiler_4.2.1         cli_3.4.1             
## [16] flashClust_1.01-2      labeling_0.4.2         sass_0.4.5            
## [19] caTools_1.18.2         scales_1.2.1           mvtnorm_1.1-3         
## [22] multcompView_0.1-8     digest_0.6.29          rmarkdown_2.20        
## [25] pkgconfig_2.0.3        htmltools_0.5.4        highr_0.10            
## [28] fastmap_1.1.0          htmlwidgets_1.6.1      rlang_1.0.6           
## [31] rstudioapi_0.14.0-9000 jquerylib_0.1.4        farver_2.1.1          
## [34] generics_0.1.3         jsonlite_1.8.4         mclust_6.0.0          
## [37] gtools_3.9.4           magrittr_2.0.3         leaps_3.1             
## [40] Rcpp_1.0.10            munsell_0.5.0          fansi_1.0.4           
## [43] lifecycle_1.0.3        scatterplot3d_0.3-42   stringi_1.7.12        
## [46] yaml_2.3.7             ggraph_2.1.0           plyr_1.8.8            
## [49] grid_4.2.1             pls_2.8-2              ggrepel_0.9.3         
## [52] lattice_0.20-45        graphlayouts_1.0.0     hms_1.1.2             
## [55] knitr_1.42             pillar_1.8.1           igraph_1.4.3          
## [58] estimability_1.4.1     glue_1.6.2             evaluate_0.20         
## [61] vctrs_0.5.2            tzdb_0.3.0             tweenr_2.0.2          
## [64] gtable_0.3.1           polyclip_1.10-4        reshape_0.8.9         
## [67] cachem_1.0.6           xfun_0.37              ggforce_0.4.1         
## [70] xtable_1.8-4           tidygraph_1.2.3        cluster_2.1.4         
## [73] timechange_0.2.0       ellipsis_0.3.2
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js