MTH 301 Final Project: Analysis

Cole Rutledge

December 11th 2024

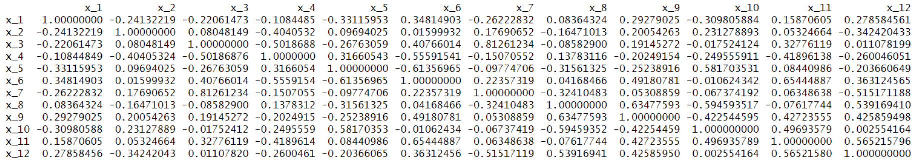
1 Introduction

2 PCA

3 Analysis of results

3.1 Method 1

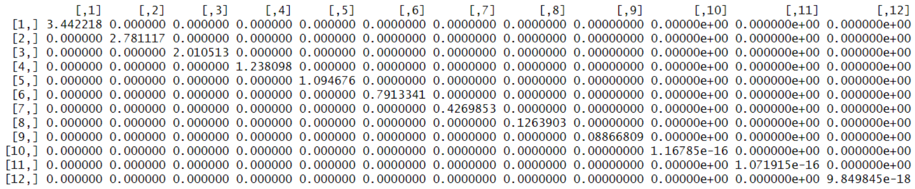
For Method 1 of our analysis, we are going to focus on those values that we determine to be the least important to the overall picture provided by the weather station data. In practice, this begins by creating the correlation matrix for our data, as shown here.



```
            x_1         x_2         x_3        x_4         x_5         x_6         x_7         x_8         x_9         x_10        x_11         x_12
x_1   1.00000000 -0.24132219 -0.22061473 -0.1084485 -0.33115953  0.34814903 -0.26222832  0.08364324  0.29279025 -0.309805884  0.15870605  0.278584561
x_2  -0.24132219  1.00000000  0.08048149 -0.4040532  0.09694025  0.01599932  0.17690652 -0.16471013  0.20054263  0.231278893  0.05324664 -0.342420433
x_3  -0.22061473  0.08048149  1.00000000 -0.5018688 -0.26763059  0.40766014  0.81261234 -0.08582900  0.19145272 -0.017524124  0.32776119  0.011078199
x_4  -0.10844849 -0.40405324 -0.50186876  1.0000000  0.31660543 -0.55591541 -0.15070552  0.13783116 -0.20249154 -0.249555911 -0.41896138 -0.260046051
x_5  -0.33115953  0.09694025 -0.26763059  0.3166054  1.00000000 -0.61356965 -0.09774706 -0.31561325 -0.25238916  0.581703531  0.08440986 -0.203660649
x_6   0.34814903  0.01599932  0.40766014 -0.5559154 -0.61356965  1.00000000  0.22357319  0.04168466  0.49180781 -0.010624342  0.65444887  0.363124565
x_7  -0.26222832  0.17690652  0.81261234 -0.1507055 -0.09774706  0.22357319  1.00000000 -0.32410483  0.05308859 -0.067374192  0.06348638 -0.515171188
x_8   0.08364324 -0.16471013 -0.08582900  0.1378312 -0.31561325  0.04168466 -0.32410483  1.00000000  0.63477593 -0.594593517 -0.07617744  0.539169410
x_9   0.29279025  0.20054263  0.19145272 -0.2024915 -0.25238916  0.49180781  0.05308859  0.63477593  1.00000000 -0.422544595  0.42723555  0.425859498
x_10 -0.30980588  0.23127889 -0.01752412 -0.2495559  0.58170353 -0.01062434 -0.06737419 -0.59459352 -0.42254459  1.000000000  0.49693579  0.002554164
x_11  0.15870605  0.05324664  0.32776119 -0.4189614  0.08440986  0.65444887  0.06348638 -0.07617744  0.42723555  0.496935789  1.00000000  0.565215796
x_12  0.27858456 -0.34242043  0.01107820 -0.2600461 -0.20366065  0.36312456 -0.51517119  0.53916941  0.42585950  0.002554164  0.56521580  1.000000000
```

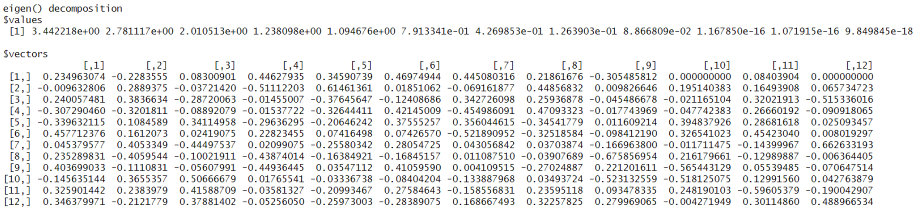
Figure 1: Initial 12x12 correlation matrix

The correlation matrix allows us to see how closely correlated any station is with any other station. For example, if we look at row 7 of our correlation matrix (this is the row for station 7) and travel along it to the third column (station 3) we can see a significantly higher value here than anywhere else in the matrix. In this case, we can assume that the data between stations 7 and 3 are highly correlated. Once we have a correlation matrix, the next step is to compute our eigenvalues and eigenvectors, to perform this task we will use R.



```
          [,1]     [,2]     [,3]     [,4]     [,5]      [,6]      [,7]      [,8]       [,9]        [,10]        [,11]        [,12]
 [1,] 3.442218 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [2,] 0.000000 2.781117 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [3,] 0.000000 0.000000 2.010513 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [4,] 0.000000 0.000000 0.000000 1.238098 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [5,] 0.000000 0.000000 0.000000 0.000000 1.094676 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [6,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.7913341 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [7,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.4269853 0.0000000 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [8,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.1263903 0.00000000  0.00000e+00 0.000000e+00 0.000000e+00
 [9,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.08866809  0.00000e+00 0.000000e+00 0.000000e+00
[10,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  1.16785e-16 0.000000e+00 0.000000e+00
[11,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 1.071915e-16 0.000000e+00
[12,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  0.00000e+00 0.000000e+00 9.849845e-18
```

Figure 2: 12x12 eigenvalue matrix

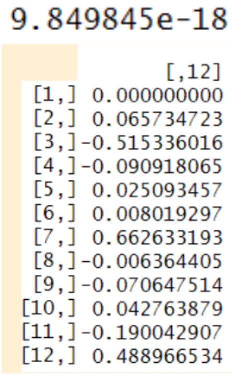


```
eigen() decomposition
$values
 [1] 3.442218e+00 2.781117e+00 2.010513e+00 1.238098e+00 1.094676e+00 7.913341e-01 4.269853e-01 1.263903e-01 8.866809e-02 1.167850e-16 1.071915e-16 9.849845e-18

$vectors
             [,1]       [,2]        [,3]        [,4]        [,5]        [,6]         [,7]        [,8]         [,9]        [,10]       [,11]        [,12]
 [1,]  0.234963074 -0.2283555  0.08300901  0.44627935  0.34590739  0.46974944  0.445080316  0.21861676 -0.305485812  0.000000000  0.08403904  0.000000000
 [2,] -0.009632806  0.2889375 -0.03721420 -0.51112203  0.61461361  0.01851062 -0.069161877  0.44856832  0.009826646  0.195140383  0.16493908  0.065734723
 [3,]  0.240057481  0.3836634 -0.28720063 -0.01455007 -0.37645647 -0.12408686  0.342726098  0.25936878 -0.045486678 -0.021165104  0.32021913 -0.515336016
 [4,] -0.307290460 -0.3201811 -0.08892079 -0.01537722 -0.32644411  0.42145009 -0.454986091  0.47093323 -0.017743969 -0.047742383  0.26660192 -0.090918065
 [5,] -0.339632115  0.1084589  0.34114958 -0.29636295 -0.20646242  0.37555257  0.356044615 -0.34541779  0.011609214  0.394837926  0.28681618  0.025093457
 [6,]  0.457712376  0.1612073  0.02419075  0.22823455  0.07416498  0.07426570 -0.521890952 -0.32518584 -0.098412190  0.326541023  0.45423040  0.008019297
 [7,]  0.045379577  0.4053349 -0.44497537  0.02099075 -0.25580342  0.28054725  0.043056842  0.03703874 -0.166963800 -0.011711475 -0.14399967  0.662633193
 [8,]  0.235289831 -0.4059544 -0.10021911 -0.43874014 -0.16384921 -0.16845157  0.011087510 -0.03907689 -0.675856954  0.216179661 -0.12989887 -0.006364405
 [9,]  0.403699033 -0.1110831 -0.05607991 -0.44936445  0.03547112  0.41059590  0.004109515 -0.27024887  0.221201611 -0.565443129  0.05539485 -0.070647514
[10,] -0.145635144  0.3655357  0.50666679  0.01765541 -0.03336738 -0.08404204 -0.133887968  0.03493724 -0.523132559 -0.518125075  0.12991560  0.042763879
[11,]  0.325901442  0.2383979  0.41588709 -0.03581327 -0.20993467  0.27584643 -0.158556831  0.23595118  0.093478335  0.248190103 -0.59605379 -0.190042907
[12,]  0.346379971 -0.2121779  0.37881402 -0.05256050 -0.25973003 -0.28389075  0.168667493  0.32257825  0.279969065 -0.004271949  0.30114860  0.488966534
```

Figure 3: The full eigen decomposition of our data, eigenvalues across the top and the eigenvector matrix underneath.

Now that we have our eigenvalues/vectors, we can examine them to find the least important station. We know that the smallest eigenvalue corresponds to the least important eigenvector, so let us take a look at that column a bit closer.

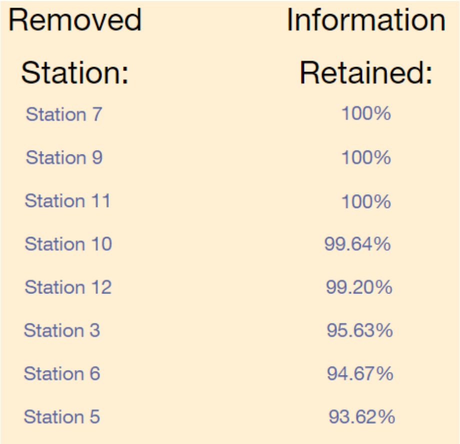


```
9.849845e-18

        [,12]
 [1,] 0.000000000
 [2,] 0.065734723
 [3,]-0.515336016
 [4,]-0.090918065
 [5,] 0.025093457
 [6,] 0.008019297
 [7,] 0.662633193
 [8,]-0.006364405
 [9,]-0.070647514
[10,] 0.042763879
[11,]-0.190042907
[12,] 0.488966534
```

Figure 4: Enter Caption

What we want to do is find the value that has the highest magnitude in the eigenvector corresponding to the smallest eigenvalue, in layman terms, we are finding the most important station in the least important column. This is the station that we will remove. So we'll begin by removing station 7 and then repeat the process. We must recompute the correlation matrix and continue to remove stations from the least important eigenvector until we have removed a preset number of stations. In our case we have the station removals organized by how much information is retained after a given station has been removed, if the client wants to retain 95% of the total information we remove stations up to that information retention value.



Removed Station:	Information Retained:
Station 7	100%
Station 9	100%
Station 11	100%
Station 10	99.64%
Station 12	99.20%
Station 3	95.63%
Station 6	94.67%
Station 5	93.62%

Figure 5: Removed stations on the left and the percentage of information retained after their removal

Given the 95% information retention goal, it would be in the state's best interest to remove stations 7, 9, 11, 10, 12, and 3. Removing anymore than these would push the data retention threshold under 95%.

3.2 Method 2

Method 2 is going to focus on the largest eigenvalues rather than the smallest. We start the same as for method one, computing the correlation matrix and then computing the eigenvectors and eigenvalues. PCA tells us that the eigenvectors associated with the largest eigenvalues have the most bearing on the overall picture the data provides. So what if we try to find the most important principal components and then remove stations based on their contribution to those. We made a scree plot to determine where our elbow point is,

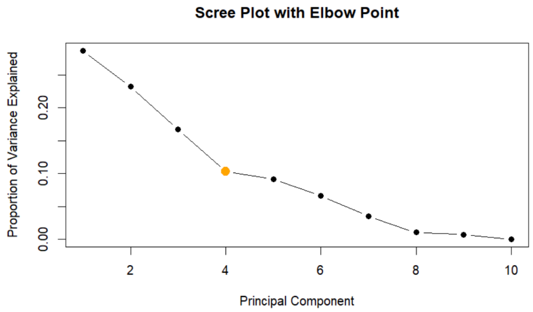
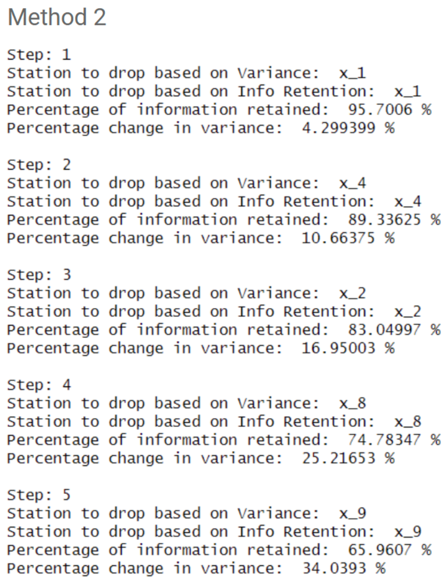


Figure 6: Scree plot showing the elbow point in orange.

R tells us that the elbow point is at component four but we chose to extend it to principal component 5 that way we would be accounting for more than 80% of the cumulative variance (approximately 88%). Now that we have determined the most important principal components, we are going to set up a loop in R that removes a single station and checks the percentage change in variance among the first 5 principal components. Whichever station provides the lowest percentage change is the one that gets removed. After a station is removed the correlation matrix and eigen analysis is re-computed and another station is removed; we can repeat this process up to any arbitrary stopping point. This method didn't really perform any better than the original method but it did result in a smaller loss in overall variance among the data which could be noteworthy.



Method 2

```
Step: 1
Station to drop based on Variance:  x_1
Station to drop based on Info Retention:  x_1
Percentage of information retained:  95.7006 %
Percentage change in variance:  4.299399 %

Step: 2
Station to drop based on Variance:  x_4
Station to drop based on Info Retention:  x_4
Percentage of information retained:  89.33625 %
Percentage change in variance:  10.66375 %

Step: 3
Station to drop based on Variance:  x_2
Station to drop based on Info Retention:  x_2
Percentage of information retained:  83.04997 %
Percentage change in variance:  16.95003 %

Step: 4
Station to drop based on Variance:  x_8
Station to drop based on Info Retention:  x_8
Percentage of information retained:  74.78347 %
Percentage change in variance:  25.21653 %

Step: 5
Station to drop based on Variance:  x_9
Station to drop based on Info Retention:  x_9
Percentage of information retained:  65.9607 %
Percentage change in variance:  34.0393 %
```

Figure 7: Results for method 2

The previous loop removed stations one at a time, in addition to iteratively removing a single station we tried removing combinations of three stations to see if the results would change at all and they did not. We also attempted to do Kmeans shown here:

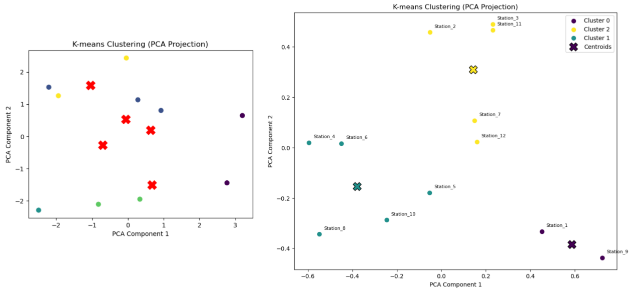


Figure 8: Two different kmeans plots, neither of which we ended up using.

But our results were particularly inconclusive so we had to scrap that plan.

4 Conclusion