PH 245 Homework 4: Principal Components Analysis and Factor Analysis

Kunal Mishra

Problem 1

In [210]:
```
# Loading Data

women = read.table(file="Data-HW4-track-women.dat",
                   header=FALSE,
                   quote="",
                   sep="\t"
                  )
men   = read.table(file="Data-HW4-track-men.dat",
                   header=FALSE,
                   quote="",
                   sep=""
                  )

colnames(women) = c("Country", "100m", "200m", "400m", "800m", "1500m", "3000m", "Marathon")
colnames(men)   = c("Country", "100m", "200m", "400m", "800m", "1500m", "5000m", "10000m", "M

# Displaying Data
head(women)
head(men)
```

Country	100m	200m	400m	800m	1500m	3000m	Marathon
ARG	11.57	22.94	52.50	2.05	4.25	9.19	150.32
AUS	11.12	22.23	48.63	1.98	4.02	8.63	143.51
AUT	11.15	22.70	50.62	1.94	4.05	8.78	154.35
BEL	11.14	22.48	51.45	1.97	4.08	8.82	143.05
BER	11.46	23.05	53.30	2.07	4.29	9.81	174.18
BRA	11.17	22.60	50.62	1.97	4.17	9.04	147.41

Country	100m	200m	400m	800m	1500m	5000m	10000m	Marathon
Argentina	10.23	20.37	46.18	1.77	3.68	13.33	27.65	129.57
Australia	9.93	20.06	44.38	1.74	3.53	12.93	27.53	127.51
Austria	10.15	20.45	45.80	1.77	3.58	13.26	27.72	132.22
Belgium	10.14	20.19	45.02	1.73	3.57	12.83	26.87	127.20
Bermuda	10.27	20.30	45.26	1.79	3.70	14.64	30.49	146.37
Brazil	10.00	19.89	44.29	1.70	3.57	13.48	28.13	126.05

Problem 1A

In [234]:
```
# Standardizing and Centering data
center      = function(lst) {lst - mean(lst)}
standardize = function(lst) {center(lst) / sd(lst)}

standardizedWomen = apply(women[,-1], 2, center)
standardizedMen   = apply(men[,-1], 2, center)

# Finding the correlations among all variables
sampleCorrelationMatrix = cor(standardizedWomen)
sampleCorrelationMatrix
```

	100m	200m	400m	800m	1500m	3000m	Marathon
100m	1.0000000	0.9410886	0.8707802	0.8091758	0.7815510	0.7278784	0.6689597
200m	0.9410886	1.0000000	0.9088096	0.8198258	0.8013282	0.7318546	0.6799537
400m	0.8707802	0.9088096	1.0000000	0.8057904	0.7197996	0.6737991	0.6769384
800m	0.8091758	0.8198258	0.8057904	1.0000000	0.9050509	0.8665732	0.8539900
1500m	0.7815510	0.8013282	0.7197996	0.9050509	1.0000000	0.9733801	0.7905565
3000m	0.7278784	0.7318546	0.6737991	0.8665732	0.9733801	1.0000000	0.7987302
Marathon	0.6689597	0.6799537	0.6769384	0.8539900	0.7905565	0.7987302	1.0000000

In [235]:
```
# Finding the eigenvalues and vectors of the correlation matrix

sampleEig = eigen(sampleCorrelationMatrix)
sampleEig
```

$values

5.80762446399961 0.628693422921518 0.279334571750058 0.124554715461547 0.0909717393558767 0.0545188221667183 0.014302264344661

$vectors

-0.3777657	-0.4071756	-0.1405803	0.58706293	-0.16706891	-0.53969730	0.08893934
-0.3832103	-0.4136291	-0.1007833	0.19407501	0.09350016	0.74493139	-0.26565662
-0.3680361	-0.4593531	0.2370255	-0.64543118	0.32727328	-0.24009405	0.12660435
-0.3947810	0.1612459	0.1475424	-0.29520804	-0.81905467	0.01650651	-0.19521315
-0.3892610	0.3090877	-0.4219855	-0.06669044	0.02613100	0.18898771	0.73076817
-0.3760945	0.4231899	-0.4060627	-0.08015699	0.35169796	-0.24049968	-0.57150644
-0.3552031	0.3892153	0.7410610	0.32107640	0.24700821	0.04826992	0.08208401

Problem 1B

In [236]:
```
# The first two eigenvalues are the largest and thus represent the
# greatest proportion of the total variance of any two eigenvalues

firstTwoPrincipalComponents = sampleEig$vectors[,1:2]
rownames(firstTwoPrincipalComponents) = colnames(standardizedWomen)
firstTwoPrincipalComponents
```

100m	-0.3777657	-0.4071756
200m	-0.3832103	-0.4136291
400m	-0.3680361	-0.4593531
800m	-0.3947810	0.1612459
1500m	-0.3892610	0.3090877
3000m	-0.3760945	0.4231899
Marathon	-0.3552031	0.3892153

In [237]:
```
proportionOfTotalVariance = {
    sum(sampleEig$values[1:2]) / sum(sampleEig$values)
}
proportionOfTotalVariance
```

0.919473983845877

Problem 1C

In [238]:
```
# Interpreting the two principal components
pcaFit = princomp(standardizedWomen)

# Examining the correlation between the original variables and PCs
cor(x=standardizedWomen, y=pcaFit$scores)[,1:2]
```

	Comp.1	Comp.2
100m	-0.6776554	-0.58409087
200m	-0.6892444	-0.62645840
400m	-0.6874604	-0.72416308
800m	-0.8587726	-0.30930106
1500m	-0.7950136	-0.26725024
3000m	-0.8021609	-0.19347819
Marathon	-0.9998947	0.01448035

In examining the results of the correlations between each Principal Component and the original variables, it's clear that PC1 correlates strongly with and thus likely *relies* on the Marathon variable. If Marathon time increases, it is likely that the times for the other race distances also increases.

In PC2, as expected, Marathon lacks almost any correlation at all. This makes sense because our principal components are orthagonal, so things that are highly correlated with one should (in theory) be similarly uncorrelated with the other principal components. With PC2, the strongest correlation is from the 400m race and this fulfills much of the same role as Marathon in PC1 -- as the 400m time increases, other variables correlated with PC2 are also likely to varying degrees to increase, based on the strength of that correlation.

When we look at pcaFit's loadings many of these suspicions are confirmed. In R, 'loadings' are different from 'correlational loadings' (check out this link if you're interested: https://stats.stackexchange.com/questions/104306/what-is-the-difference-between-loadings-and-correlation-loadings-in-pca-and (https://stats.stackexchange.com/questions/104306/what-is-the-difference-between-loadings-and-correlation-loadings-in-pca-and)), which is why pcaFit's loadings don't necessarily represent the correlations between principal components and original variables

In [216]:
```
pcaFit$loadings
```

```
Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
100m            -0.115 -0.173  0.292  0.933              
200m            -0.290 -0.387  0.795 -0.354              
400m     -0.108 -0.938  0.226 -0.238                     
800m                                         0.377 -0.925
1500m                  -0.268                0.883  0.370
3000m                  -0.834 -0.471        -0.265       
Marathon -0.992  0.119                                   

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

Problem 1D

In [217]:
```
# Adding country names back to scores

PCWomen = cbind(women[,1], as.data.frame(pcaFit$scores))
colnames(PCWomen)[1] = "Country"
head(PCWomen)
```

Country	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7
ARG	3.2173904	-0.8550659	-0.06070576	-0.28983107	0.25592070	0.059231461	0.0005619794
AUS	10.4529924	2.2790525	-0.25585771	0.14908698	0.11270809	0.009042302	-0.0240115435
AUT	-0.5440192	1.5468370	0.20234078	0.09185523	-0.03281219	-0.052436540	0.0385870352
BEL	10.5872958	-0.5138938	0.09198800	-0.41590033	0.03304306	0.003534003	0.0150943480
BER	-20.5753000	1.1584461	0.28974010	-0.48505882	0.10460656	-0.082132802	-0.0062352914
BRA	6.3348740	0.7240189	-0.22250549	-0.18099578	0.01147622	0.008477327	0.0294356049

In [218]:
```
# Sorting countries based only on PC1
dimReducedWomen = PCWomen[,1:2]
head(dimReducedWomen)
dimReducedWomenOrdered = dimReducedWomen[order(-dimReducedWomen[,2]),]
head(dimReducedWomenOrdered)
```

Country	Comp.1
ARG	3.2173904
AUS	10.4529924
AUT	-0.5440192
BEL	10.5872958
BER	-20.5753000
BRA	6.3348740

	Country	Comp.1
19	GBR	18.58051
29	KEN	15.09708
9	CHN	14.45185
28	JPN	14.11345
54	USA	12.81715
18	GER	12.63928

In examining the results based on ordering countries by first principal component scores, we get countries that would intuitively be the best in the world at track.

Problem 1E

In [228]:
```
# Converting to time to m/s

womenSpeeds = cbind(
    100/women[,2],
    200/women[,3],
    400/women[,4],
    800/(women[,5]*60),
    1500/(women[,6]*60),
    3000/(women[,7]*60),
    42195/(women[,8]*60)
)
colnames(womenSpeeds) = c("100m", "200m", "400m", "800m", "1500m", "3000m", "Marathon")
head(womenSpeeds)

standardizedWomenSpeeds = apply(womenSpeeds, 2, center)
head(standardizedWomenSpeeds)
```

100m	200m	400m	800m	1500m	3000m	Marathon
8.643042	8.718396	7.619048	6.504065	5.882353	5.440696	4.678353
8.992806	8.996851	8.225375	6.734007	6.218905	5.793743	4.900355
8.968610	8.810573	7.902015	6.872852	6.172840	5.694761	4.556203
8.976661	8.896797	7.774538	6.768190	6.127451	5.668934	4.916113
8.726003	8.676790	7.504690	6.441224	5.827506	5.096840	4.037490
8.952551	8.849558	7.902015	6.768190	5.995204	5.530973	4.770708

100m	200m	400m	800m	1500m	3000m	Marathon
-0.1717296	0.05398771	-0.09301975	-0.1001494	-0.107334149	-0.10200509	0.05808862
0.1780338	0.33244299	0.51330791	0.1297923	0.229218382	0.25104126	0.28009115
0.1538379	0.14616458	0.18994764	0.2686378	0.183152416	0.15205932	-0.06406079
0.1618887	0.23238905	0.06247102	0.1639751	0.137763890	0.12623274	0.29584902
-0.0887685	0.01238148	-0.20737694	-0.1629906	-0.162181263	-0.44586154	-0.58277427
0.1377795	0.18514941	0.18994764	0.1639751	0.005516747	-0.01172805	0.15044333

In [246]:
```
# Running PCA on the new dataset and comparing to the previous set of loadings
pcaFitWomenSpeeds = princomp(standardizedWomenSpeeds)

# Examining the correlation between the original variables and PCs
cor(x=standardizedWomen, y=pcaFitWomenSpeeds$scores)[,1:2]

pcaFitWomenSpeeds$loadings
summary(pcaFitWomenSpeeds)
```

	Comp.1	Comp.2
100m	0.8919935	0.34956718
200m	0.9081678	0.36064894
400m	0.8779449	0.39229996
800m	0.9491733	-0.07404633
1500m	0.9410317	-0.19258749
3000m	0.9107122	-0.28356140
Marathon	0.8653738	-0.30107320

```
Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
100m     -0.310 -0.376         0.585         0.624  0.138
200m     -0.357 -0.434         0.323        -0.689 -0.311
400m     -0.379 -0.519  0.274 -0.667  0.187  0.124  0.132
800m     -0.299               -0.128 -0.894  0.136 -0.265
1500m    -0.391  0.211 -0.435        -0.127 -0.236  0.734
3000m    -0.460  0.396 -0.427 -0.184  0.357  0.199 -0.499
Marathon -0.423  0.445  0.730  0.237  0.136

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000

Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
Standard deviation     0.8476961 0.29065087 0.18100245 0.12124349 0.09320466
Proportion of Variance 0.8285389 0.09740377 0.03777473 0.01694921 0.01001631
Cumulative Proportion  0.8285389 0.92594269 0.96371742 0.98066663 0.99068294
                            Comp.6      Comp.7
Standard deviation     0.077803348 0.045025448
Proportion of Variance 0.006979577 0.002337484
Cumulative Proportion  0.997662516 1.000000000
```

In [243]:
```
# Adding country name back to scores
PCWomenSpeeds = cbind(women[,1], as.data.frame(pcaFitWomenSpeeds$scores))
colnames(PCWomenSpeeds)[1] = "Country"
head(PCWomenSpeeds)

# Sorting countries based only on PC1
dimReducedWomenSpeeds = PCWomenSpeeds[,1:2]
head(dimReducedWomenSpeeds)
dimReducedWomenSpeedsOrdered = {
    dimReducedWomenSpeeds[order(dimReducedWomenSpeeds[,2]),]
}
head(dimReducedWomenSpeedsOrdered)
```

Country	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7
ARG	0.1635073	0.04692099	0.11381196	0.03027142	0.05102581	-0.16919941	-0.04848137
AUS	-0.7307601	-0.19835239	0.09838941	-0.13986402	0.09675452	-0.06346773	0.02394173
AUT	-0.3667764	-0.13521031	-0.15313758	-0.07710711	-0.17103151	0.04762582	-0.01799413
BEL	-0.4429985	0.02515002	0.09155928	0.14629341	-0.05270801	-0.06033366	-0.01897176
BER	0.6651627	-0.34274202	-0.22271265	0.06416636	-0.11470105	-0.11505330	0.04795862
BRA	-0.2903061	-0.15852299	0.14326748	0.03018820	-0.08357439	-0.01153138	-0.03272404

Country	Comp.1
ARG	0.1635073
AUS	-0.7307601
AUT	-0.3667764
BEL	-0.4429985
BER	0.6651627
BRA	-0.2903061

	Country	Comp.1
54	USA	-1.201996
9	CHN	-1.176150
45	RUS	-1.123772
18	GER	-1.122766
19	GBR	-0.985712
17	FRA	-0.857734

So... our results are a bit different, but not drastically so. One thing I'm somewhat uncertain about is whether I needed to standardize, rather than just center my data before running princomp on it -- in both cases, I just centered so at least I kept it consistent, but I think there's something to be said about the need for standardization here with different units (seconds vs minutes).

In terms of interpretation, the components are actually quite different, possibly due to our shift in units (essentially de facto standardization by switching everything to m/s). That said, we still acheived roughly the same results (still matching intuition) because the first two PCs account for roughly the same variation in the dataset, albeit in subtly different ways.

Problem 1F

In [248]:
```
# Running PCA on the new dataset
pcaFitMen = princomp(standardizedMen)

# Examining the correlation between the original variables and PCs
cor(x=standardizedMen, y=pcaFitMen$scores)[,1:2]
```

	Comp.1	Comp.2
100m	-0.6863014	-0.48250693
200m	-0.7307341	-0.49239083
400m	-0.7257308	-0.68042774
800m	-0.8138640	-0.28621813
1500m	-0.8833311	-0.21656608
5000m	-0.9495998	-0.16363965
10000m	-0.9590991	-0.13293332
Marathon	-0.9997660	0.01998497

In [249]:
```
# Examining loadings and proportions of variance
pcaFitMen$loadings
summary(pcaFitMen)
```

```
Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
100m                          -0.324 -0.312  0.883              
200m            -0.253        -0.897  0.172 -0.292              
400m     -0.114 -0.916  0.253  0.288                            
800m                                        -0.127  0.194 -0.971
1500m                                -0.206 -0.110  0.945  0.215
5000m           -0.117 -0.377        -0.826 -0.305 -0.246       
10000m   -0.175 -0.209 -0.873         0.382  0.120              
Marathon -0.974  0.167  0.155                                   

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000

Importance of components:
                          Comp.1     Comp.2      Comp.3       Comp.4
Standard deviation     9.1072660 1.05839941 0.473844266 0.2812010715
Proportion of Variance 0.9828776 0.01327463 0.002660692 0.0009370383
Cumulative Proportion  0.9828776 0.99615224 0.998812929 0.9997499674
                             Comp.5       Comp.6       Comp.7       Comp.8
Standard deviation     0.1075227532 7.836237e-02 5.484458e-02 1.974378e-02
Proportion of Variance 0.0001370011 7.276768e-05 3.564436e-05 4.619384e-06
Cumulative Proportion  0.9998869686 9.999597e-01 9.999954e-01 1.000000e+00
```

In [251]:
```
# Adding country name back to scores
PCMen = cbind(men[,1], as.data.frame(pcaFitMen$scores))
colnames(PCMen)[1] = "Country"
head(PCMen)

# Sorting countries based only on PC1
dimReducedMen = PCMen[,1:2]
head(dimReducedMen)
dimReducedMenOrdered = {
    dimReducedMen[order(-dimReducedMen[,2]),]
}
head(dimReducedMenOrdered)
```

Country	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
Argentina	3.949866	-0.71642187	0.3502378	0.19323708	-0.125843666	0.02828136	0.06972650	-0.004182508
Australia	6.233111	0.77309952	-0.1908892	0.04604324	0.215548144	-0.02797992	0.03255537	-0.010344625
Austria	1.405618	0.05992883	0.6388328	0.08182631	0.006978615	-0.01742626	-0.01887815	-0.021609773
Belgium	6.576032	0.22934014	0.5478972	0.03103714	0.003313427	0.06765641	0.05335714	-0.001526373
Bermuda	-12.899964	2.20666160	-0.2616143	0.19567169	-0.205917545	0.07198977	-0.10741879	-0.016413862
Brazil	7.522252	0.45693596	-1.1846968	0.12143605	-0.064657949	-0.01605633	-0.02046180	0.040151598

Country	Comp.1
Argentina	3.949866
Australia	6.233111
Austria	1.405618
Belgium	6.576032
Bermuda	-12.899964
Brazil	7.522252

	Country	Comp.1
29	Kenya	9.325825
54	U.S.A.	8.528414
6	Brazil	7.522252
28	Japan	7.469135
17	France	7.340499
43	Portugal	7.201771

Yep, it looks like our results once again agree pretty closely (though not exactly) with our women's analysis. In this case, it is definitely very intuitive -- Kenya is known for its Olympic gold medal exploits and the fastest person in the world, so it's unsurprising when a component that accounts for so much of the variance in the data (98%+) can rank the countries with high accuracy.

The PC's relations to each of the original variables is actually fairly similar across genders so that is also something interesting of note.

Problem 2

In [254]:
```
# Loading data
airPollution = read.table(file="Data-HW4-pollution.dat",
                          header=FALSE,
                          quote="",
                          sep=""
                         )
colnames(airPollution) = c("Wind", "SolarRadiation", "CO",
                           "NO", "NO2", "O3", "HC")
head(airPollution)
```

Wind	SolarRadiation	CO	NO	NO2	O3	HC
8	98	7	2	12	8	2
7	107	4	3	9	5	3
7	103	4	3	5	6	3
10	88	5	2	8	15	4
6	91	4	2	8	10	3
8	90	5	2	12	12	4

Problem 2A

In [284]:
```
# Generating the covariance matrix

airPollutionCovariance = cor(airPollution)
```

Problem 2B

In [291]:
```
# Obtaining principal component solution

# 1. Performing spectral decomposition
decomposition = eigen(airPollutionCovariance)
decomposition
```

$values

2.33678264275777 1.38600066762446 1.20406592509298 0.727086483934652 0.653476542788251 0.536688791847867 0.155898945954013

$vectors

0.2368211	0.278445138	0.6434744	0.172719491	0.56053441	-0.223579220	-0.24146701
-0.2055665	-0.526613869	0.2244690	0.778136601	-0.15613432	-0.005700851	-0.01126548
-0.5510839	-0.006819502	-0.1136089	0.005301798	0.57342221	-0.109538907	0.58524622
-0.3776151	0.434674253	-0.4070978	0.290503052	-0.05669070	-0.450234781	-0.46088973
-0.4980161	0.199767367	0.1965567	-0.042428178	0.05021430	0.744968707	-0.33784371
-0.3245506	-0.566973655	0.1598465	-0.507915905	0.08024349	-0.330583071	-0.41707805
-0.3194032	0.307882771	0.5410484	-0.143082348	-0.56607057	-0.266469812	0.31391372

In [304]:
```
# 2. Estimating Communality
rootOfEigenvals = decomposition$values ** .5

L1 = as.data.frame( decomposition$vectors[,1] * rootOfEigenvals[1] )
L2 = as.data.frame( decomposition$vectors[,2] * rootOfEigenvals[2] )

colnames(L1) = ''
colnames(L2) = ''

rownames(L1) = colnames(airPollution)
rownames(L2) = colnames(airPollution)

print("L1:")
round(L1, 3)
print("L2:")
round(L2, 3)

# For m=1
communalityM1 = round(L1^2, 3)
print("Communality - M=1:")
communalityM1

# For m=2
communalityM2 = round(L1^2 + L2^2, 3)
print("Communality - M=2:")
communalityM2
```

```
[1] "L1:"
```

Wind	0.362
SolarRadiation	-0.314
CO	-0.842
NO	-0.577
NO2	-0.761
O3	-0.496
HC	-0.488

```
[1] "L2:"
```

Wind	0.328
SolarRadiation	-0.620
CO	-0.008
NO	0.512
NO2	0.235
O3	-0.667
HC	0.362

```
[1] "Communality - M=1:"
```

Wind	0.131
SolarRadiation	0.099
CO	0.710
NO	0.333
NO2	0.580
O3	0.246

HC	0.238

```
[1] "Communality - M=2:"
```

Wind	0.239
SolarRadiation	0.483
CO	0.710
NO	0.595
NO2	0.635
O3	0.692
HC	0.370

In [305]:
```
# 3. Estimating Specific Variation (psi)

# For m=1
specificVarianceM1 = round(1 - L1^2, 3)
print("Specific Variance - M=1:")
specificVarianceM1

# For m=2
specificVarianceM2 = round(1 - L1^2 - L2^2, 3)
print("Specific Variance - M=2:")
specificVarianceM2
```

```
[1] "Specific Variance - M=1:"
```

Wind	0.869
SolarRadiation	0.901
CO	0.290
NO	0.667
NO2	0.420
O3	0.754
HC	0.762

```
[1] "Specific Variance - M=2:"
```

Wind	0.761
SolarRadiation	0.517
CO	0.290
NO	0.405
NO2	0.365
O3	0.308
HC	0.630

As expected, our Specific Variance drops in almost all of the common variables with the addition of a second common factor. This is because the second common factor is accounting for more of the total variance and sicne it is zero-sum, the additional variance is being "taken" from specific variance and "given" to the second common factor.

Problem 2C

In [312]:
```
# Finding proportion of variation for one-factor model - m=1
proportionalVarianceM1 = sum(L1^2) / length(L1[,1])
proportionalVarianceM1
```

0.333826091822538

In [313]:
```
# Finding proportion of variation for two-factor model - m=2
proportionalVarianceM2 = {
    proportionalVarianceM1 + (sum(L2^2) / length(L2[,1]))
}

proportionalVarianceM2
```

0.531826187197461

Once again, as expected, our two-factor model accounts for more variation. This relates back to the end of 2B because as specific variation goes down, the total amount of variation being accounted for by our factors is going up.

Problem 2D

In [320]:
```
# Performing varimax rotation

rotation = varimax(x=as.matrix(cbind(L1, L2)), normalize=FALSE)
rotation
```

```
$loadings

Loadings:
               Var.1  Var.2
Wind            0.160  0.461
SolarRadiation        -0.695
CO             -0.735 -0.412
NO             -0.752  0.171
NO2            -0.781 -0.160
O3             -0.114 -0.824
HC             -0.602

               Var.1 Var.2
SS loadings    2.117 1.606
Proportion Var 0.302 0.229
Cumulative Var 0.302 0.532

$rotmat
           [,1]      [,2]
[1,]  0.8768458 0.4807718
[2,] -0.4807718 0.8768458
```

In computing the varimax rotation, we've just scaled the loadings by dividing them by their corresponding communality and maximizing this quantity. We've done this in order to interpret our results more easily and as such, the proportion of our variance has remained constant, despite the rotation and changing factor values.

In Factor 1's loadings, HC, NO2, NO, and CO have fairly significant (>.5) values. This means Factor 1 is primarily a measure of these variables and as each of these variables increase, so do the other 3. Thus, we may use this information to understand what underlying common factor Factor 1 is picking up on and the real-life mechanisms that may cause those variables to be associated with each other (i.e. diesel exhaust or coal power plant burns). In Factor 2, the most important significant values (>.5) come from O3 and Solar Radiation which means Factor 2 is primarily a measure of these variables. These variables also thus are associated with each other and a second underlying common factor could be investigated regarding the relationship between Ozone and Solar Radiation. From domain knowledge, I know increased sunlight and UV radiation is responsible for the *creation of* ozone throughout the atmosphere, so them being associated makes a lot of sense.