248 Final Project: Principal Component Analysis

Monty Gash

1/3/2022

Introduction

Data set: 'Multilevel Monitoring of Activity and Sleep in Healthy People'. From "psysionet.org".

This research was conducted by health scientists, psychologists, and chemists from the University of Pisa in Italy.

These data were collected on 22 "healthy" adult males pertaining to their sleep and physiological characteristics.

The variables of interest for this research consisted of a long list of scores to self-reported questionnaires and daily movement recorded by wearable trackers.

Data collection methods

Data was collected constantly over a 24-hour period as participants underwent their daily routines. There were two major ways that the researchers collected the data; via self-reporting and via bio-trackers.

Self-reporting: A large part of the data relied on user's self-recorded answers to standardized questionnaires. A problem arises with the fact that self-reporting relies on subjective experience. We will have to assume that since the questionnaires that were used are all standardized, the creators of the questionnaires likely accounted for the self-reporting factor.

Bio tracker data: The other half of the data was recorded using bio trackers that constantly recorded movements over the 24-hour period. The heart rate was used to track sleep stages, and would indicate the physical rigor associated with each of the participant's daily activities.

The study does not mention if the sample of participants is random. Participants of the study were volunteers. We can not be sure that the research practiced random selection, and therefore we should refrain from generalizing the conclusions gained from the data set to a population of individuals that is too different from the set of participants.

There are also no details about the way these researchers quantified being 'healthy'.

Guiding questions: Which variables in the data set contain the most variance between the data points? What variables do we need in order to effectively explain the variance of the data points found in the data set?

Methodology: *Principal Component Analysis (PCA)*

PCA is a form of dimensionality reduction that was first invented in 1901 by Karl Pearson.

The background mathematics for PCA are based in linear algebra; eigenvectors/values, vector projections, derivatives of matrix operations, etc.

This project omits the deeper mathematical explanation of PCA, and rather discusses what Principal Component Analysis is, why we use it, how to implement it in R, and how to analyze the results.

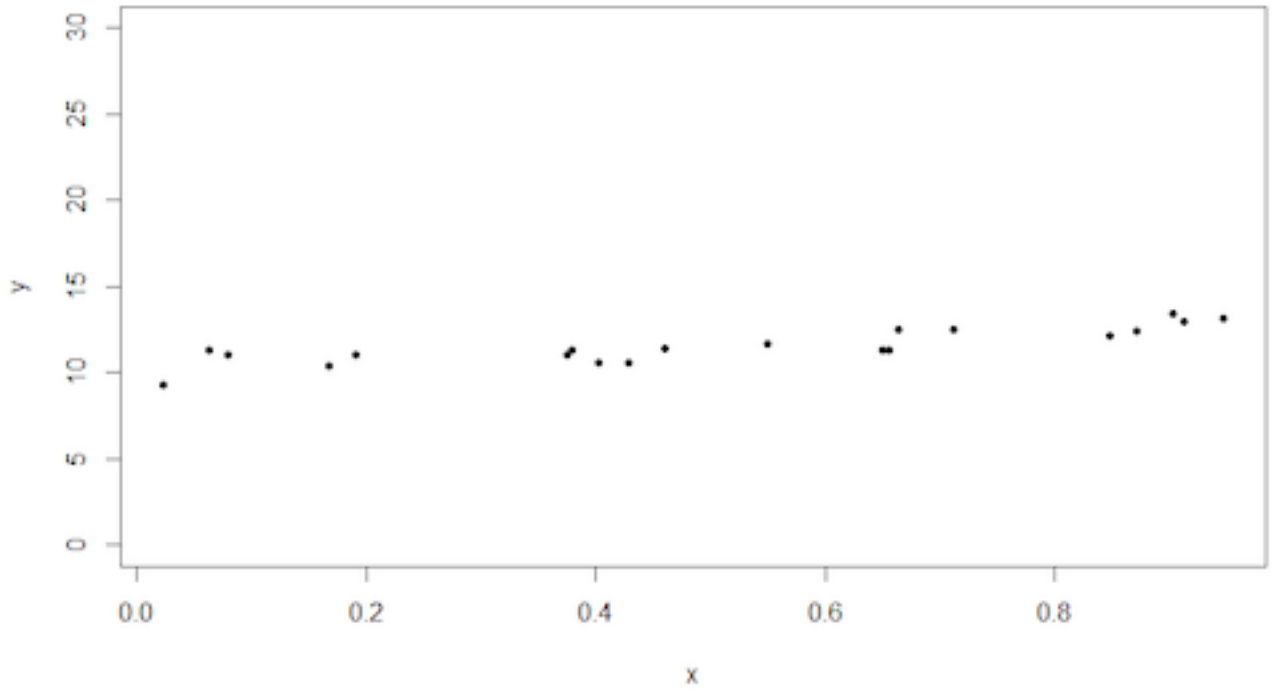
Principal Component Analysis

We can begin by thinking of each variable in a data set as one dimension.

Consider two variables, x and y, each representing a dimension.

A scatter plot of the data points given the two variables can be seen below.

```
knitr::include_graphics("ScatterPlot.png")
```



We can see that most of the variation between each data point has to do with the values for the x-axis, so we can remove the y-axis and express the variation using a number line, without getting rid of too much of the original variation described by the two variables.

In this case, we are transitioning from two dimensions to one dimension, while maintaining a similar level of variability among the data points.

This is what principal component analysis seeks to do with multiple dimensions.

PCA is trying to find the line that captures the most variability from end to end, and minimizes the error; the distance from the line to each data point.

Why PCA?

Principal component analysis helps us find variables that explain the most variability.

Allows for a more clear view of the data set, where we can more easily identify trends, clusters, or jumps in the data.

Helps remove multicollinearity. PCA creates new variables, called principal components, which are linear combinations of the original variables. PCA groups variables with high correlation into the same principal components, which ultimately creates a new set of variables are are less correlated with each other than the original set.

Our data set.

Lets begin by taking a look at the data

```
library(readr)
finalData248<-read_csv("finalDataSet248.csv")
```

```
## Rows: 22 Columns: 20
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (20): weightKg, heightCm, age, MEQ, STAI1, STAI2, pittsburgh, dailyStres...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(finalData248)
```

```
## # A tibble: 6 x 20
##   weightKg heightCm   age   MEQ STAI1 STAI2 pittsburgh dailyStress panasPos22
##      <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl>       <dbl>      <dbl>
## 1       65      169    29    47    41    43          5          23         18
## 2       95      183    27    52    24    39          7          26         27
## 3       70      174    34    59    27    27          8          11         28
## 4       76      180    27    60    28    40          4          10         19
## 5       80      196    25    52    54    47          8          41         27
## 6       62      178    27    48    32    47          9          41         25
## # ... with 11 more variables: panasNeg22 <dbl>, Latency <dbl>,
## #   Efficiency <dbl>, totalMinutesInBed <dbl>, totalSleepTime <dbl>,
## #   wakeAfterSleepOnset <dbl>, numberOfAwakenings <dbl>,
## #   averageAwakeningLength <dbl>, movementIndex <dbl>, fragentationIndex <dbl>,
## #   sleepFragmentationIndex <dbl>
```

We see physical characteristics on the far left, scores for various questionnaires, and scores of their sleep times recorded by their trackers. Let's find out which of these variables account for the most variance between individuals in the data set by using PCA.

Performing PCA.

Scale Data.

We must first scale the variables so that each of them have a mean of 1 and a standard deviation of zero. We have to do this because the variables in question measure different things. PCA seeks to find the variables that explain the most variability, so the most variability would immediately be the variable that has the largest values for its measurements if we did not first scale our variables.

Using the scale function in R.

```
scaledData<-scale(finalData248)
summary(scaledData)[1:6,1:5]
```

```
##     weightKg          heightCm            age                MEQ
##  Min.   :-1.1764   Min.   :-1.32766   Min.   :-1.58371   Min.   :-1.7531
##  1st Qu.:-0.6291   1st Qu.:-0.59745   1st Qu.:-0.44509   1st Qu.:-0.7213
##  Median :-0.3945   Median : 0.01106   Median : 0.01035   Median :-0.1164
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.3874   3rd Qu.: 0.37617   3rd Qu.: 0.18114   3rd Qu.: 0.6663
##  Max.   : 3.1240   Max.   : 3.05363   Max.   : 2.97075   Max.   : 1.9472
##      STAI1
##  Min.   :-1.3970
##  1st Qu.:-0.7667
##  Median :-0.2565
```

```
##  Mean   : 0.0000
##  3rd Qu.: 0.5839
##  Max.   : 2.2046
```

The above scaled variables show a mean of 0 and a standard deviation of about 1.

Now we can use R to perform PCA on the scaled data with the 'princomp' function.

```
data.pca<-princomp(scaledData)
summary(data.pca)
```

```
## Importance of components:
##                            Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     2.2306357 1.7472302 1.5702883 1.31988388 1.20276937
## Proportion of Variance 0.2606338 0.1599093 0.1291612 0.09125251 0.07577712
## Cumulative Proportion  0.2606338 0.4205430 0.5497043 0.64095679 0.71673391
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     1.12367529 0.99227399 0.91354159 0.88787060 0.72127900
## Proportion of Variance 0.06613861 0.05157469 0.04371495 0.04129265 0.02725084
## Cumulative Proportion  0.78287252 0.83444721 0.87816216 0.91945481 0.94670566
##                           Comp.11    Comp.12     Comp.13     Comp.14     Comp.15
## Standard deviation     0.5763273 0.47930711 0.425214424 0.391733529 0.279564379
## Proportion of Variance 0.0173985 0.01203375 0.009470859 0.008038127 0.004093898
## Cumulative Proportion  0.9641042 0.97613791 0.985608767 0.993646894 0.997740792
##                            Comp.16      Comp.17      Comp.18 Comp.19 Comp.20
## Standard deviation     0.172064970 0.1067051849 0.0462382691       0       0
## Proportion of Variance 0.001550809 0.0005964093 0.0001119893       0       0
## Cumulative Proportion  0.999291601 0.9998880107 1.0000000000       1       1
```

Here we have the principle components. We should focus on the Proportion of variance and the cumulative proportion values.

Proportion of Variance: How much variance each principal component accounts for after the ones that come before it.

Cumulative Proportion: How much variance in total is described by the principal component and the ones that come before it.

We can choose a cut off point for how many principal components we want to keep based on how much total variance we want to describe with our new dimensions.

Based on the amount of variability we desire to explain, we can choose a point to stop including principal components by examining the cumulative proportion. In this case, it seems that we would need 14 variables to reach a point where 99% of the variability is explained; 13 if we were to round up from 0.9856.

Constructing a Scree plot.

A Scree Plot is a graph showing the amount of the variability described by each principal component.

To create one, we first must calculate the total percentage of variance explained by each of the principal components. We can do so by dividing the individual variations explained by each component by the sum of the variation explained by each component.

Using R to calculate the variance explained by each principal component.

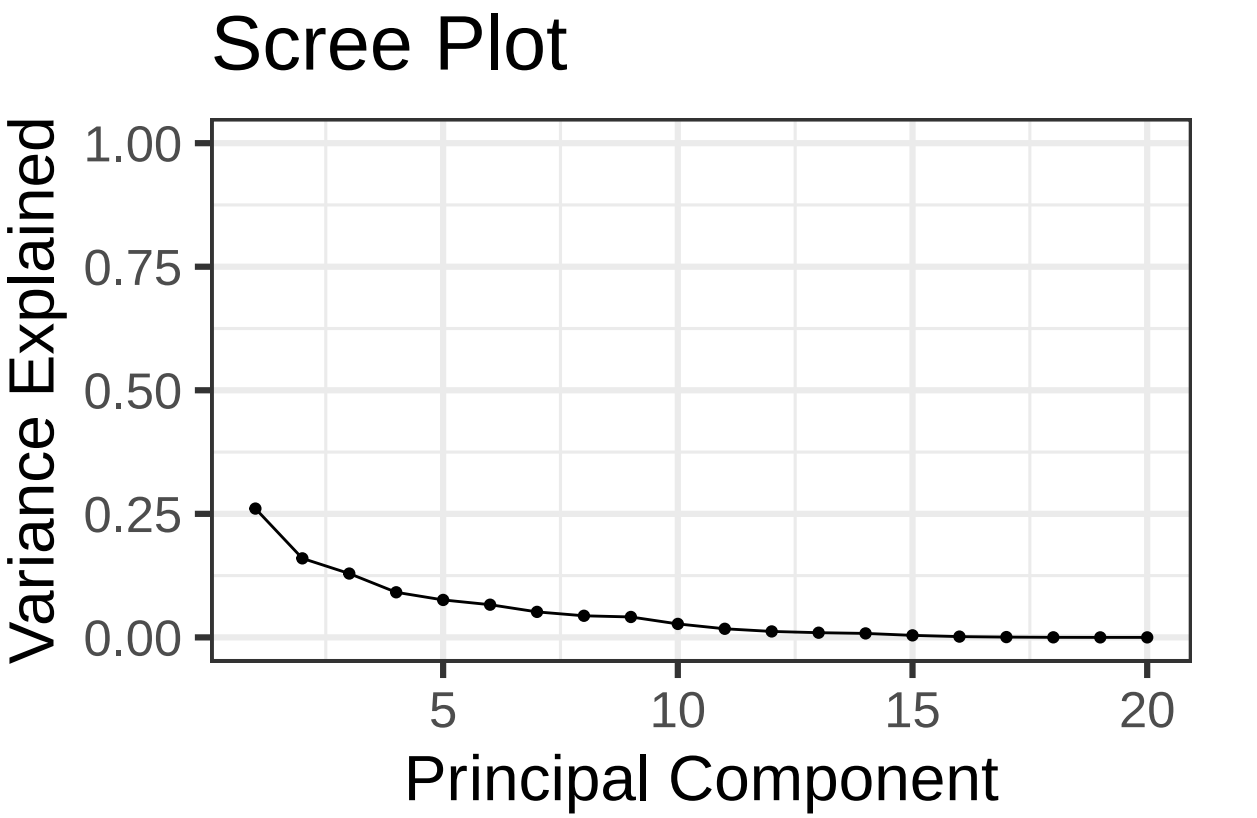
```
varExplained<-data.pca$sdev^2/sum(data.pca$sdev^2)
head(varExplained)
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
## 0.26063377 0.15990928 0.12916123 0.09125251 0.07577712 0.06613861
```

These values correspond to the "Proportion of Variance" for each principal component.

Creating the Scree Plot with R.

```
library(ggplot2)
qplot(, varExplained) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)
```



Here is a visualization of the variance described by each of the principal components. All graphs of principal components look similar to this one, since the first PC will always account for the most variance, followed by PC 2, and so on an so on. All Scree Plots show variation decreasing as the principal component number increases.

Loadings.

The loadings of a principal component are the weights on each of the variables within that principal component. Each loading shows how much each variable contributes to a given principal component.

From the loadings, we can determine what each principle component is actually taking into consideration. Since each principal component is a linear combination of multiple variables in the data set, the loadings can be thought of as the scalars attached to each of the vectors(variables) in the linear combination. So for example, if we were to have a principal component that has high loadings for the width, length, height of a dog, we can say that the first PC captures the overall size of the dog.

We can use the loadings function in R and view the loadings for the first 5 PC's.

```
data.pca$loadings[,1:5]
```

```
##                          Comp.1       Comp.2       Comp.3       Comp.4
```

```
## weightKg                 0.09818569  0.02926067  0.508715436  0.059650766
## heightCm                -0.09271686  0.05997502  0.331054952  0.330825748
## age                     -0.04033502  0.18774853  0.109126898  0.435224816
## MEQ                     -0.02588934  0.18488153 -0.162707352  0.415777333
## STAI1                    0.22280032  0.07529449 -0.055100961  0.094671037
## STAI2                    0.22204039  0.03283118  0.192185100 -0.001744506
## pittsburgh               0.04433960 -0.35589333  0.145257781  0.200812649
## dailyStress             -0.19729016  0.04147207 -0.228899217  0.302605200
## panasPos22               0.14012300 -0.16377508  0.322855815  0.345850171
## panasNeg22              -0.01662811 -0.21083825  0.130512104 -0.349847391
## Latency                 -0.16400170 -0.18255898  0.301445540  0.080916816
## Efficiency               0.36149685  0.02036750 -0.247223768  0.124686752
## totalMinutesInBed       -0.18487397 -0.41606076 -0.217849267  0.188732206
## totalSleepTime          -0.06535763 -0.42206627 -0.289604689  0.219040209
## wakeAfterSleepOnset     -0.39308689 -0.13511971  0.096606202 -0.014837775
## numberOfAwakenings      -0.32536426  0.13819641 -0.033970580  0.104210995
## averageAwakeningLength  -0.03226042 -0.48840456  0.016654353 -0.083425326
## movementIndex           -0.30667064  0.02514761  0.231283777 -0.111435518
## fragentationIndex       -0.34058004  0.18338096 -0.104404391 -0.032114438
## sleepFragmentationIndex -0.39017379  0.15674578  0.004347392 -0.068234905
##                              Comp.5
## weightKg                 0.211377865
## heightCm                 0.432456511
## age                     -0.285966732
## MEQ                     -0.334236110
## STAI1                    0.338612968
## STAI2                   -0.161463539
## pittsburgh               0.268975406
## dailyStress              0.396386332
## panasPos22              -0.103931866
## panasNeg22               0.062332877
## Latency                 -0.375208996
## Efficiency               0.089977395
## totalMinutesInBed       -0.028994008
## totalSleepTime          -0.007730211
## wakeAfterSleepOnset     -0.049006637
## numberOfAwakenings      -0.045667650
## averageAwakeningLength  -0.015355017
## movementIndex           -0.008234445
## fragentationIndex        0.148165998
## sleepFragmentationIndex  0.115765382
```

Above are the loadings for the first 5 principal components given each original variable.

A positive loading represents a positive correlation between the variable and PC.

A negative loading represents a negative correlation between the variable and PC.

Principal component 1.

Largest weights for principal component 1:

'Efficiency', 0.3614, percentage of sleep time on total sleep in bed.

'wakeAfterSleepOnset', -0.3931, time spent awake after falling asleep for the first time.

'numberOfAwakenings', -0.3254, number of awakenings after falling asleep for the first time.

'movementIndex', -0.3407, number of minutes without movement expressed as a percentage of the movement phase.

'fragmentationIndex', -.3406, number of minutes with movement expressed as a percentage of the immobile phase.

'sleepFragmentationIndex', -0.3902, ratio between movement and fragmentation indices.

So we can see that the first principal component considers the above variables, and these variables explain the most variability between individuals in the data set. These variables can be summarized by thinking as the first PC as considering the time asleep, awakenings during sleep, and movement during sleep.

Principal component 2.

The next principal component, 2, has different variables with the highest weights:

'pittsburgh', -0.3559, self-reported questionnaire about sleep quality over the past month.

'totalMinutesInBed', -0.4161, time in bed.

'totalTimeAsleep', -0.4221, total time asleep.

'averageAwakeningLength', -0.4884, average time of each awakening after first falling asleep.

So the second principal component accounts for the above variables that have to do with sleep quality. These variables within the second principal component account for the second most variation among individuals within the data set.

We can note that the variables that have the highest weights for PC 2 seem to be very similar to the ones in PC 1; each set of variables has to do with sleep in general. So when labeling our principal components we have to make sure we explain what exactly each one is measuring in terms of the original variables in the data set.

Principal component 4.

'heightCm', 0.330825748, height in centimeters.

'age', 0.435224816, age in years.

'MEQ', 0.415777333, value for the response to Morningness–eveningness questionnaire(morning person or evening person?)

'dailyStress', 0.302605200, reported stress levels from questionnaire.

'panasPos22', 0.345850171, reported positive emotions from questionnaire.

'panasNeg22', -0.349847391, reported negative emotions from questionnaire.

So PC 4 has the highest weights coming from height and age, two physical characteristics, and responses to questionnaires accounting for mood.

Key Idea: Each principal component acts as a new variable that has weights corresponding to each original variable. We can classify each principal component as accounting for something that characterizes the group of original variables that have the highest weights for that principal component.

Scores. The scores tell us where each of our participants will lie on the new dimension we created. The 0 mark is an intercept, negative numbers indicate small values for the PC, positive numbers indicate large values for the PC.

Scores for PC 1.

```
data.pca$scores[,1]
```

```
##  [1]  2.5136192 -0.6991074 -0.7210470 -0.1118773  1.0389694  0.8494147
##  [7] -4.8852226 -3.2433385  1.5990006 -0.7673439  3.9616666  0.9752558
## [13] -1.3532686  1.9457510  1.8271066 -2.9560856  0.6548140 -0.7792053
```

```
## [19] -4.1027221  1.9797776  2.5656547 -0.2918119
```

Score for participant 1: 2.514

Score for participant 19: -4.103

Given the above scores, we should expect to see the first participant have larger values for the variables with the highest weights for principal component 1, compared to the 19th participant.

For PC 1, we can note that the largest weights are 'Efficiency', 0.3614, 'wakeAfterSleepOnset', -0.3931, 'numberOfAwakenings', -0.3254, 'movementIndex', -0.3407, 'fragmentationIndex', -.3406, 'sleepFragmentationIndex', -0.3902.

Lets examine rows 1 and 19 of the data set to see if our conjecture is true.

```
finalData248[c(1,19),1:9]
```

```
## # A tibble: 2 x 9
##   weightKg heightCm   age   MEQ STAI1 STAI2 pittsburgh dailyStress panasPos22
##      <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl>       <dbl>      <dbl>
## 1       65      169    29    47    41    43          5          23         18
## 2       70      183    22    44    26    35          4          31         17
```

```
finalData248[c(1,19),10:15]
```

```
## # A tibble: 2 x 6
##   panasNeg22 Latency Efficiency totalMinutesInB~ totalSleepTime wakeAfterSleepO~
##        <dbl>   <dbl>      <dbl>            <dbl>          <dbl>            <dbl>
## 1         10       0       87.3              165            144               21
## 2         15       1       74.1              459            340              118
```

```
finalData248[c(1,19),16:19]
```

```
## # A tibble: 2 x 4
##   numberOfAwakenings averageAwakeningLength movementIndex fragentationIndex
##                <dbl>                  <dbl>         <dbl>             <dbl>
## 1                  9                   2.33          9.09              10
## 2                 44                   2.68         19.0               22.2
```

```
finalData248[c(1,19),19:20]
```

```
## # A tibble: 2 x 2
##   fragentationIndex sleepFragmentationIndex
##               <dbl>                   <dbl>
## 1              10                      19.1
## 2              22.2                    41.2
```

Efficiency: Participant 1 has a higher value for efficiency than participant 19, which concurs with our hypothesis.

wakeAfterSleepOnset: Participant 1 has a lower value for this variable than participant 19, but this makes sense, since we see that the weight associated with 'wakeAfterSLeepOnset' is negative. If the weight is negative, we should expect the first participant to have a lower value than the 19th for the given variable, since the scores for participant 1 and 19 are 2.514 and -4.103, respectively.

numberOfAwakenings: Participant 1 has a lower value for this variable than participant 19, which concurs with our hypothesis.

movementIndex: Participant 1 has a lower value for this variable than participant 19, which concurs with our hypothesis.

fragmentationIndex: Participant 1 has a lower value for this variable than participant 19, which concurs with our hypothesis.

sleepFragmentationIndex: Participant 1 has a lower value for this variable than participant 19, which concurs with our hypothesis.

The above analysis shows how the scores of each user for each PC relates to the values they have for each variable in the data set. The score of each data point explains the relationship between that data point and the principal component.

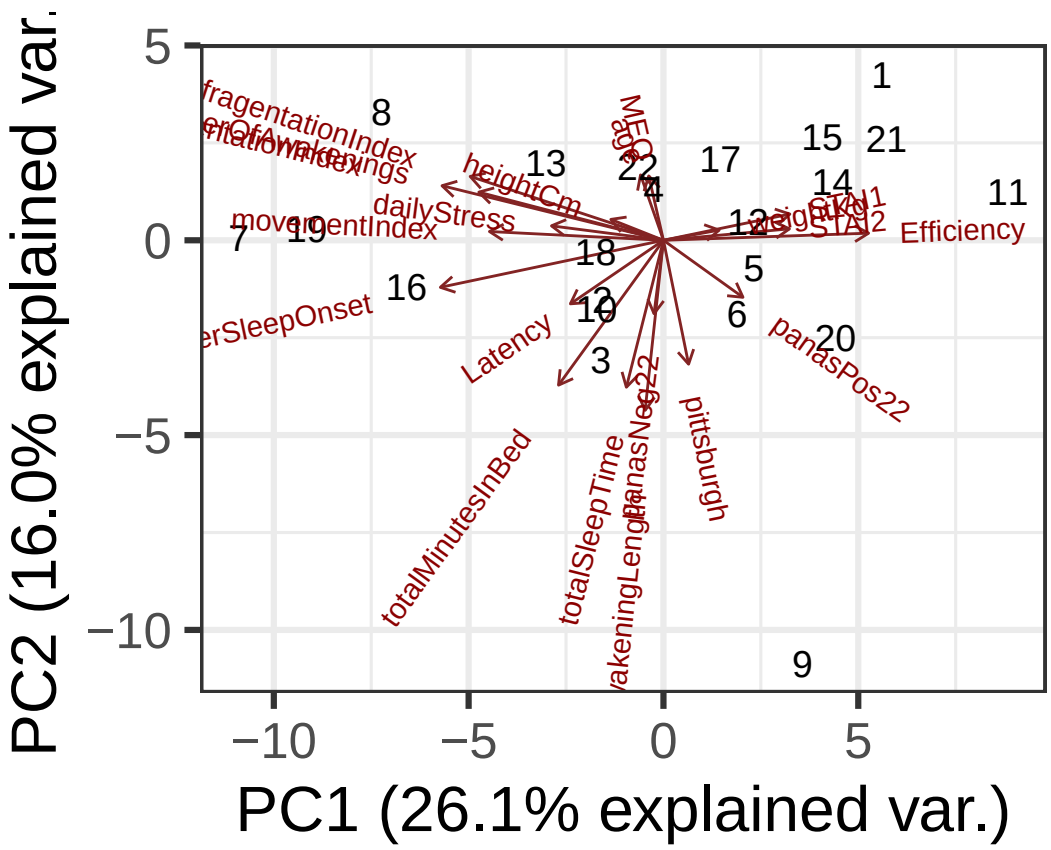
A positive score demonstrates a positive relationship with the given principal component, and a negative score demonstrates a negative relationship with each principal component.

To determine what the score means in the context of the original variables, we refer to the loadings(weights) of each variable given the principal component.

Biplot. A biplot is a graph that has one principal component on each axis. Using it, we can interpret the values of the data points in 2 dimensions, given the new variables(principal components) we created.

We can use R to create a biplot using the first two principal components.

```
biplot = ggbiplot(pcobj = data.pca,
                  choices = c(1,2),
                  obs.scale = 2, var.scale = 2,
                  labels = row.names(finalData248),
                  labels.size = 5,
                  varname.size = 4,
                  varname.abbrev = FALSE,
                  var.axes = TRUE,
                  circle = FALSE,
                  ellipse = FALSE)
print(biplot)
```



The lines seen in the biplot correspond to the loadings for the Principal components. The loadings are the weights associated with each variable within each principal component.

Variables whose lines are close to each other have high correlation.

Variables with lines that are far from each other have weak correlation.

We can note that the lines are centered at (0,0).

Another important aspect of the biplot to know is that differences along the first PC axis are more substantial than those coming after. So the differences between the clusters given PC 1 are more substantial than the differences given PC 2.

Variables with lines that fall to the right of vertical-axis the PC1 axis have a positive correlation with PC 1. So pittsburgh and Efficiency have positive correlations with PC1. Lines on the left of that axis have a negative correlation with PC 1.

Variables with lines that fall above the horizontal-axis have a positive correlation with PC 2, while those variables whose lines fall below have a negative correlation with PC 2.

We can test this idea by examining the loadings of the first two principal components.

For 'pittsburgh', we should expect to see a positive loading for PC1, and a negative loading for PC2.

For 'Efficiency', we should expect to see a positive loading for PC1, and a, just barely, positive loading for PC2.

Loadings of the first two principal components.

```
data.pca$loadings[,1:2]
```

```
##                              Comp.1      Comp.2
## weightKg                 0.09818569  0.02926067
```

```
## heightCm                -0.09271686  0.05997502
## age                     -0.04033502  0.18774853
## MEQ                     -0.02588934  0.18488153
## STAI1                    0.22280032  0.07529449
## STAI2                    0.22204039  0.03283118
## pittsburgh               0.04433960 -0.35589333
## dailyStress             -0.19729016  0.04147207
## panasPos22               0.14012300 -0.16377508
## panasNeg22              -0.01662811 -0.21083825
## Latency                 -0.16400170 -0.18255898
## Efficiency               0.36149685  0.02036750
## totalMinutesInBed       -0.18487397 -0.41606076
## totalSleepTime          -0.06535763 -0.42206627
## wakeAfterSleepOnset     -0.39308689 -0.13511971
## numberOfAwakenings      -0.32536426  0.13819641
## averageAwakeningLength  -0.03226042 -0.48840456
## movementIndex           -0.30667064  0.02514761
## fragentationIndex       -0.34058004  0.18338096
## sleepFragmentationIndex -0.39017379  0.15674578
```

The loading for 'pittsburgh' is positive given PC1 and negative given PC2, which concurs with the inferences we made given the biplot.

The loading for 'Efficiency' is positive given PC1, and positive given PC2, which concurs with the inferences we made given the biplot. We can also note that the loading for 'Efficiency' given PC2 is 0.02037, which is a small positive number. The fact that it is a small positive number relates to the fact that the line for 'Efficiency' seen in the biplot is just barely above the horizontal-zero-axis.

Where the data points lie tell us their value for the principal component, and ultimately which type of values they will have for each of the variables that make up each principal component. So for example, participant 11 should have a much higher value for 'efficiency' than participant 7.

Let's see if this is the case by showing the values of 'Efficiency' for participants 7 and 11.

```
finalData248[c(7,11),12]
```

```
## # A tibble: 2 x 1
##   Efficiency
##        <dbl>
## 1       75.3
## 2       92.0
```

This is the case.

Using a biplot is key in order to get a visualization of the distribution of the data points given two distinct principal components.

Conclusion: What can we do once we create principal components?

Reduce multicollinearity. The main goal of principal component analysis is to gather variables that are highly correlated into specific principal components. By doing so, we are reducing the dimensionality of our data set, which makes it easier to do the following:

Identify the "important" variables. The initial question that we sought to answer was which variables explain the most variability, and principal components lays that out for us. More specifically, we should look at the weights associated with the most prominent principal components to decide which variables explain the most variability. "Most prominent" meaning up until we are satisfied with the amount of variability that each principal component explains. Remember that we choose the stopping point of how much variability we want

our principal components to explain. Which ever variables have the highest weights in the most prominent principal components should be the ones that we keep when analyzing our data.

Identify clustering. We can look for clusters in the data. These clusters are easier to see when we have less dimensions. There are some clusters that we can see in the biplot, but since our data set only contains 22 samples, the clustering is not that prominent. Data sets with more samples will show more clear evidence of clustering when using PCA. The data points that are clustered together are similar based on the two particular principal components in the biplot.

Identify outliers. From the biplot, we can see that participant 9 seems to be an outlier based on the first two principal components. Removing that outlier may improve the statistical significance of our conclusions.

Model conditions and drawbacks to PCA

Model conditions.

There should be examples of variables with high correlation among the data set.

The original variables must be continuous.

Drawbacks.

It is hard to interpret the new variables. As we saw above, we really need to do some digging and really understand what is going on in order to accurately describe what each principal component represents.

There are case where the principal components don't reduce that many variables. We saw that we would need about 14 out of the original 20 variables in order to describe 99% of the variability with the newly created principal components.

We will be losing some information depending on the end amount of variability we want to maintain. If we are at 90% with 3 PC's but 99% with 8, is is reasonable to remove 5 variables while losing 9% of the explained variability? The answer to this question depends on the goals of the researcher.

References

https://www.youtube.com/watch?v=TJdH6rPA-TI

https://www.youtube.com/watch?v=kw9R0nD69OU

https://www.youtube.com/watch?v=g-Hb26agBFg

https://www.datacamp.com/community/tutorials/pca-analysis-r

https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component- analysis-pca-and-how-it-is-used-507186#: ~:text=Principal%20component%20analysis% 2C%20or%20PCA,more%20easily%20visualized%20and

https://github.com/vqv/ggbiplot/issues/53

https://en.wikipedia.org/wiki/Principal_component_analysis)

https://www.youtube.com/watch?v=83c2Y5gErjg

https://www.statology.org/scree-plot-r/

https://www.youtube.com/watch?v=HMOI_lkzW08