Dimensionality Reduction on Anthropometric Measurements

Student Number: 10707266

Dimensionality reduction is the process of summarising many variables of data as a reduced set of principal variables. When faced with a large set of correlated variables, principal components allows us to summarise this set with a smaller number of orthogonal representative variables that collectively explain most of the variability in the original dataset. Principal component analysis (PCA) is a process which computes principal components using the eigensystem of the data's correlation matrix.

In this work, 9 variables from the female dataset from the 2012 U.S Army Anthropometric Survey were used to carry out dimensionality reduction using PCA and factor analysis (FA):

- Ankle circumference
- Chest circumference
- Ear length
- Functional leg length
- Head length
- Sitting height
- Span
- Thigh circumference
- Weight

PCA RESULTS

As a preliminary task, the data were normalised to obtain a zero-mean dataset before applying PCA.

	PC1	PC2	PC3	PC4
anklecircumference	0.328203	0.143855	0.092553	0.191681
chestcircumference	0.353400	0.444276	-0.019952	-0.047740
earlength	0.171594	0.064647	0.832081	-0.408142
functionalleglength	0.372372	-0.387566	-0.121546	0.125015
headlength	0.236536	-0.288914	-0.245115	-0.774438
sittingheight	0.271329	-0.315615	0.388082	0.414004
span	0.331391	-0.506430	-0.103703	0.076135
thighcircumference	0.390311	0.359257	-0.230752	0.010974
weightkg	0.455581	0.242729	-0.095993	0.041224

Figure 1*: First 4 principal components of PCA performed on the female dataset with 9 variables*.

The first principal component has large positive association with weight, leg length, chest circumference and thigh circumference, which can be interpreted as body size. The second principal component has large positive association with chest circumference and large negative association with span, so this component measures the contrast between span and chest size. Similarly, the third principal component represents ear size and fourth principal component represents head size - 2 measurements that are uncorrelated with the other variables.

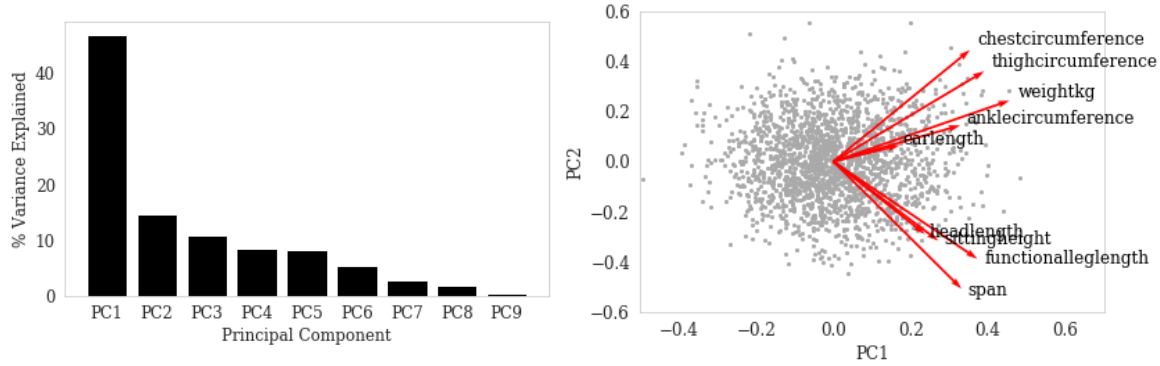


***Figure 2**: (Left) Scree plot outlining the proportion of variance that each principal component explains. (Right) Biplot showing the loadings and original data along the first two principal components, capturing 61.6% of the variability in the data.*

To obtain a good understanding of the data, one requires the smallest number of principal components required, but there is no simple method to derive the number of components required [1]. Therefore, PCA is successful when few components explain much of the variance, resulting in an L-shaped scree plot. Figure 2 shows the first 4 components is a desirable choice as they explain 81% of the variation of the data.

FA RESULTS

Factor Analysis (FA) is another dimensionality reduction technique which creates factors from the observed variables which explains the variance due to correlation among the observed variables. If a factor solution has an eigenvalue of 1 or above, it explains more variance than a single observed variable – which means it can be useful to reducing number of variables.

To uniquely determine the factor, multivariate normalise is assumed and the factors rotated using the varimax.

	0	1	2	3	communalities
anklecircumference	0.422650	0.124936	0.485048	0.153374	0.569031
chestcircumference	0.823313	0.113807	-0.002039	0.372996	0.931921
earlength	0.119788	0.074247	0.137518	0.464214	0.255533
functionalleglength	0.271338	0.831493	0.205582	0.099107	0.842686
headlength	0.164110	0.328984	0.121133	0.057247	0.263760
sittingheight	0.083365	0.323924	0.574413	0.282055	0.527365
span	0.124569	0.897464	0.147687	0.113773	0.885353
thighcircumference	0.899112	0.196397	0.140763	0.010338	0.944181
weightkg	0.856145	0.342369	0.280031	0.200353	0.995005

***Figure 3**: First 4 factors of FA performed on the female dataset with 9 variables.*

Figure 3 shows the loadings which indicate how much a factor explains a variable. Factor 0 has high factor loadings for chest circumference, thigh circumference and weight, so can deduce this factor explains the common variance in people who are large. Factor 1 has high factor loadings for functional leg length and span, explains the common variance in people with long limbs.

The communalities are high for all variables besides ear length and head length, indicating the other factors are well represented by the four factors.

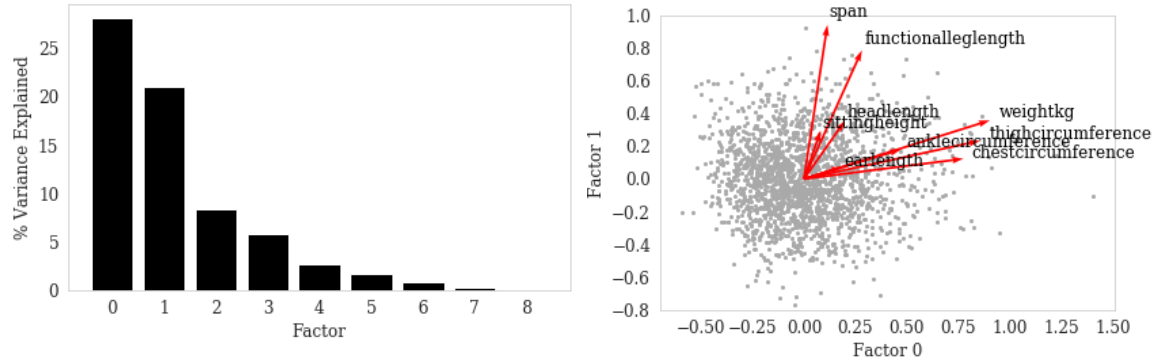


Figure 4*: (Left) Scree plot outlining the proportion of variance that each factor explains. (Right) Biplot showing the loadings and original data along the first two factors, capturing 49.2% of the variability in the data.*

In practice, for a constructing to be valid, the variance explain should be at least 60% [2]. Figure 4 shows the first 4 factors explain a 63.6% of the variability in the data and the shape is a less defined L-shape, the corresponding eigenvalue scree plot in the Appendix shows the first 3 factors have an eigenvalue less than 1, indicating that these factors are useful for dimensionality reduction. This evidence suggests that PCA produces better representative data after dimensionality reduction.

REFERENCES

[1] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer.

[2] Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2014). Pearson new international edition. In Multivariate data analysis, Seventh Edition. Pearson Education Limited Harlow, Essex.

CODE

```
%matplotlib inline
import numpy as np
import scipy.stats as st
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set_style("whitegrid", {'axes.grid' : False})
import sklearn as sk
from sklearn import preprocessing
from sklearn.decomposition import PCA

```

```
# Create a directory for figures
import os
if not os.path.exists('Figures'):
    os.makedirs('Figures')
```

```
# Set plot formatting
plt.rcParams['font.family'] = 'serif'
#plt.rcParams['font.size'] = 12
plt.rcParams['text.color'] = 'black'
```

```
# Read in the data as a Pandas frame
Femaledf = pd.read_csv('./ANSUR_II_FEMALE_Public.csv', encoding='latin-1')

#print(Femaledf.columns.values)
```

```
# Pull out 9 anthropometric meansurements to do PCA
df = Femaledf[["anklecircumference", "chestcircumference", "earlength",
               "functionalleglength", "headlength", "sittingheight",
               "span", "thighcircumference", "weightkg"]]

```

```
# Normalise the data
df = (df-np.mean(df, axis=0))/np.std(df, axis=0)
df.to_csv('myfemalemeasurements.csv')
df.head()
```

```
# Fit PCA with scikit learn package
pca = PCA(n_components=9)
reduced_df = pca.fit_transform(df)
print(reduced_df[:,0])

# Set columns and index for plotting an interpreting PCs
columns = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9']
index = ["anklecircumference", "chestcircumference", "earlength",
               "functionalleglength", "headlength", "sittingheight",
               "span", "thighcircumference", "weightkg"]

# Lets determine the signifant variables in each PC
component_df = pd.DataFrame(np.transpose(pca.components_), columns=columns, index=index)
print(component_df)

print("\n\nVariance Proportion:\n", pca.explained_variance_ratio_)
#print("numpy var: \n", La/sum(La)) # manual calc
print("\ne'values: \n", pca.explained_variance_)
```

```
percent_variance = np.round(pca.explained_variance_ratio_* 100, decimals =2)
columns = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9']
plt.bar(x= range(1,10), height=percent_variance, tick_label=columns, color='black')
plt.ylabel('% Variance Explained')
plt.xlabel('Principal Component')
plt.title('PCA Scree Plot')
plt.tight_layout()
plt.savefig('Figures/PCAScreePlot.png',format='png')
plt.show()
```

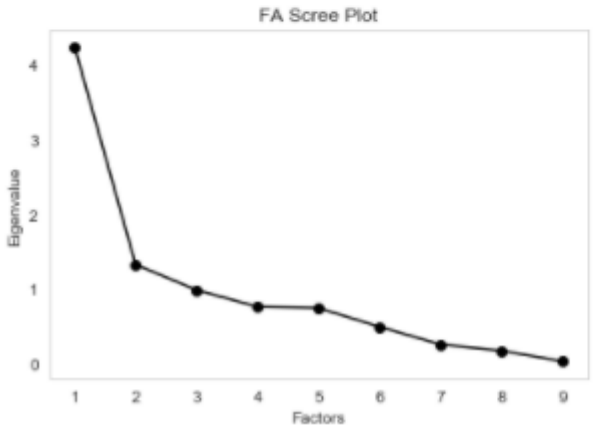
```
xs = reduced_df[:,0]
ys = reduced_df[:,1]

scalex = 1.0/(xs.max() - xs.min())
scaley = 1.0/(ys.max() - ys.min())

# Plot PCA
plt.scatter(xs * scalex, ys * scaley ,s=4, color='darkgray')
tail = [0,0]
plt.quiver(*tail, component_df["PC1"], component_df["PC2"],
           scale=1, scale_units='xy', angles='xy',
           width=0.005, color='red')
for i in range(0, len(index)):
    plt.annotate(str(index[i]), xy=(1.05*component_df["PC1"][i], 1.05*component_df["PC2"][i]))
plt.xlim(-0.5, 0.7)
plt.ylim(-0.6, 0.6)
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.savefig('Figures/loadingplot.png',format='png')
plt.show

```

```
# Perform Factor Analysis with scikit learn package
from factor_analyzer import FactorAnalyzer
```

```
# Create factor analysis object and perform FA
fa = FactorAnalyzer(n_factors=9, rotation='varimax')
fa.fit(df)
ev, v = fa.get_eigenvalues()
print(fa)
```

```
FactorAnalyzer(bounds=(0.005, 1), impute='median', is_corr_matrix=False,
               method='minres', n_factors=9, rotation='varimax',
               rotation_kwargs={}, use_smc=True)
```

```
# Create another scree plot
x = range(1,df.shape[1]+1)
y = ev
plt.scatter(x, y, color='black')
plt.ylabel('Eigenvalue')
plt.xlabel('Factors')
plt.title('FA Scree Plot')
plt.plot(x, y, color='black')
plt.savefig('Figures/fascree.png',format='png')
```




```
factor_df = pd.DataFrame(fa.loadings_,  index=index)
print(factor_df)
```

```
FA_vardf = pd.DataFrame(fa.get_factor_variance(),index=['Variance','Proportional Var','Cumulative Var'])
print(FA_vardf)

percent_variance = np.round(FA_vardf.loc['Proportional Var']* 100, decimals =2)
columns = ['0', '1', '2', '3', '4', '5', '6', '7', '8']
plt.bar(x= range(1,10), height=percent_variance, tick_label=columns, color='black')
plt.ylabel('% Variance Explained')
plt.xlabel('Factor')
plt.title('PCA Scree Plot')
plt.tight_layout()
plt.savefig('Figures/FAScreePlot.png',format='png')
plt.show()
```

```
communalities = fa.get_communalities()
factor_df["communalities"] = communalities
#print(factor_df)
```

```
fa_reduced = fa.transform(df)
#print(fa_reduced)
```

```
xs = fa_reduced[:,0]
ys = fa_reduced[:,1]

scalex = 2.0/(xs.max() - xs.min())
scaley = 2.0/(ys.max() - ys.min())

# Plot FA
plt.scatter(xs * scalex, ys * scaley ,s=4, color='darkgray')
tail = [0,0]
plt.quiver(*tail, factor_df[0], factor_df[1],
           scale=1, scale_units='xy', angles='xy',
           width=0.005, color='red')
for i in range(0, len(index)):
    plt.annotate(str(index[i]), xy=(1.05*factor_df[0][i], 1.05*factor_df[1][i]))
#plt.xlim(-0.5, 0.7)
plt.ylim(-0.8, 1)
plt.xlabel("Factor 0")
plt.ylabel("Factor 1")

plt.savefig('Figures/FAloadingplot.png',format='png')
plt.show
```