Principal Component Analysis to Address Multicollinearity

Lexi V. Perez

May 13, 2017

Contents

Abstract

In multiple linear regression models, covariates are sometimes correlated with one another. Multicollinearity can cause parameter estimates to be inaccurate, among many other statistical analysis problems. When these problems arise, there are various remedial measures we can take. Principal component analysis is one of these measures, and uses the manipulation and analyzation of data matrices to reduce covariate dimensions, while maximizing the amount of variation.

1 Introduction

We will begin by reviewing simple linear regression, multiple linear regression and matrix representations of each model. An introduction to multicollinearity will follow, where it is important to notice the inaccuracy and variability of parameter estimations in each of the examples. Before exploring principal component analysis (PCA), we will look into related matrix algebra and concepts to help us understand the PCA process. Finally, as a solution to multicollinearity, we will walk through the steps of PCA and an example showing this as a remedial measure to the parameter estimation problem previously demonstrated.

2 Simple Linear Regression

2.1 Regression Model

In Chapter One of *Applied Linear Regression Models* [KNN04], a simple linear regression model is defined as:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

where:

Y_i is the value of the response variable in the i^{th} trial;
β_0 and β_1 are parameters;
X_i is the value of the predictor variable in the i^{th} trial;
ε_i is a random error term with mean $E\{\varepsilon_i\} = 0$ and variance $\sigma^2\{\varepsilon_i\} = \sigma^2$.

This model looks similar to the commonly used linear equation, $y = mx + b$. In simple linear regression, β_0 is the y-intercept value and β_1 is the slope of the model. We estimate these parameters based on data that we are working with and the line that best fits these data. After estimating β_0 and β_1, we are able to analyze many other aspects of the data. For example, we can create confidence intervals, analyze variance, test for randomness, normality, outliers and many other values. Essentially, these parameter values are important and interesting in analyzing the relationship between our X and Y variables.

2.2 Matrix Approach

When a data set is very large, we need an easier way to keep track of and manipulate the data. Chapter Five of *Applied Linear Regression Models* [KNN04], reviews matrix algebra and operations, which we can then apply to analyzing simple linear regression models. Converting the simple linear regression model, $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_1$, $i = 1, ..., n$, into matrix notation we get:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

We can now write these matrices in a similar formatted equation:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Notice the column of 1's in the $\mathbf{X}$ matrix. This is to account for the constant β_0 intercept that doesn't depend directly on any X values.

Using these matrices, we can easily find values that assist us in regression analysis such as fitted values, residuals and sums of squares. For example, finding residuals is calculated using the vector of residuals $e_i = Y_i - \hat{Y}_i$ which is denoted:

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}.$$

In a matrix equation we then have:

$$\mathbf{e} = \boldsymbol{Y} - \hat{\boldsymbol{Y}} = \boldsymbol{Y} - \mathbf{X}\mathbf{b},$$

where $\mathbf{b}$ is the estimated vector of $\boldsymbol{\beta}$. We will go into more detail about the $\mathbf{b}$ vector in 4, but for now it is important to notice that it is much easier to calculate these values using matrices. [Lay06]

3 Multiple Linear Regression

3.1 Regression Model

In Chapter Six of *Applied Linear Regression Models*, the general linear regression model is defined as:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{i,p} + \varepsilon_i$$

where:

$\beta_0, \beta_1, ..., \beta_p$ are parameters;
$X_{i1}, ..., X_{i,p}$ are known constants;
ε_i are independent $N(0, \sigma^2)$;
$i = 1, ..., n$.

Some special cases of multiple linear regression include polynomial regression, transformed variables, interaction effects, or any combination of these. Each case has a specific way of transforming the equation we have back into the familiar multiple regression model form. Even though there are many analysis techniques that are similar to simple linear regression, there are also some specialized topics unique to multiple linear regression. For example, calculating extra sums of squares, the standardized version of the multiple linear regression model, and multicollinearity. We will specifically be focusing on multicollinearity in this paper, but each of these topics have their own effect on data analyzation. [KNN04]

3.2 Matrix Approach

Similar to matrix notation for simple linear regression, we will use the multiple linear regression model

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + ... + \beta_p X_{i,p} + \varepsilon$$

and convert this into matrix notation:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \mathbf{X} = \begin{bmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,p} \\ 1 & X_{2,1} & X_{2,2} & \dots & X_{2,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{n,1} & X_{n,2} & \dots & X_{n,p} \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

So in matrix terms, the model is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where:

$\mathbf{Y}$ is a vector of responses;
$\boldsymbol{\beta}$ is a vector of parameters;
$\mathbf{X}$ is a matrix of constants;
$\boldsymbol{\varepsilon}$ is a vector of independent normal random variables.

We again see the column of 1's in the $\mathbf{X}$ matrix. Sometimes in a multiple regression model, especially in examples we'll be using, we will work with what is called the design matrix. The design matrix is the $\mathbf{X}$ matrix without the first column of 1's. We use this to focus specifically on the relationship between the covariates. [KNN04]

4 Multicollinearity

Chapter Seven of *Applied Linear Regression Models* [KNN04] gives the following definition of multicollinearity.

Definition 4.1. Multicollinearity exists among the predictor variables when these variables are correlated among themselves.

Notice that multicollinearity can only occur when when we have two or more covariates, or in multiple linear regression. This phenomenon can have effects on the extra sums of squares, fitted values and predictions, regression coefficients, and many other parts of multiple linear regression. We will be focusing specifically on how multicollinearity affects parameter estimates in Sections 4.1, 4.2 and 4.3. [KNN04]

4.1 Example: Simulation

In this example, we will use a simple two-variable model,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon,$$

to get us started with multicollinearity.

Let the sample size be $n = 100$, and the parameter values to be $\beta_0 = 4$, $\beta_1 = 2$, and $\beta_2 = 8$. To create a sample, we will generate 100 X_1 and X_2 values each, over the Uniform distribution. We set

the range for X_1 to be 4 to 25, and the range for X_2 to be 3 to 10. These minimum and maximum values are arbitrary; we are simply generating random values in these ranges for the covariates. Finally, we will simulate 100 different ε values, or error values over the Normal distribution. We will consider two cases.

Case One: First let's examine the case where X_1 and X_2 are uncorrelated so we can see how the parameters are supposed to behave under a simulation of data. X_1 and X_2 will be randomly generated completely independent from each other. We will look at the full model,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon,$$

in the R output below.

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.95216 -0.56536  0.01235  0.55097  2.77369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.89673    0.33462   11.64   <2e-16 ***
## X1           1.99619    0.01447  137.92   <2e-16 ***
## X2           8.00489    0.01543  518.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9126 on 97 degrees of freedom
## Multiple R-squared:  0.9997,Adjusted R-squared:  0.9997
## F-statistic: 1.418e+05 on 2 and 97 DF,  p-value: < 2.2e-16
```

Under "Coefficients" in the "Estimate" column, we will see the parameter estimates for β_0, β_1 and β_2 in that order. The simulated data did a great job estimating the parameter values of 4, 2, and 8 that we set. Next, let's look at the output for the following two models:

$$Y = \beta_0 + \beta_2 X_2 + \varepsilon, \qquad \text{and} \qquad Y = \beta_0 + \beta_1 X_1 + \varepsilon.$$

```
##
## Call:
## lm(formula = Y ~ X2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.6225 -11.4740   0.8744  10.5931  19.8163
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  34.9569     3.4567   10.11   <2e-16 ***
## X2            7.9319     0.2154   36.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.75 on 98 degrees of freedom
## Multiple R-squared:  0.9326,Adjusted R-squared:  0.9319
## F-statistic:  1356 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = Y ~ X1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.772 -43.930   8.703  40.648  74.529
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 127.1906    12.3488  10.300   <2e-16 ***
## X1            1.7385     0.7583   2.293    0.024 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.84 on 98 degrees of freedom
## Multiple R-squared:  0.05091,Adjusted R-squared:  0.04123
## F-statistic: 5.257 on 1 and 98 DF,  p-value: 0.024
```

Notice that both of these models also predict the β_1 and β_2 parameters very well. However, we see that the β_0 values are quite inaccurate. This just means that the regression line is shifting, so, in fact, we expect this to change as we remove or add predictor variables to the model. We are really concerned with the β_1 and β_2 values because these describe the relationship between the X and Y variables; they shouldn't be changing much at all. In the first model, with just X_2, β_2 is almost exactly 8, and in the second model, with just X_1 is also very close to it's initial set value of 2. This is because there is very little to no correlation between X_1 and X_2. In case two we will see the effects of multicollinearity on parameter estimates.

Case Two: We will now consider the case where X_1 and X_2 are correlated with one another. To do this, when we are simulating data we will add the random X_1 value onto the randomly generated X_2 value. This causes X_2 to depend significantly on what the random X_1 value is, and causes them to be very highly correlated with one another. First, let's consider the full model,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon.$$

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.95216 -0.56536  0.01235  0.55097  2.77369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.87227    0.39317   9.849 2.84e-16 ***
## X1           1.98151    0.04802  41.265  < 2e-16 ***
## X2           8.01468    0.04629 173.156  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9126 on 97 degrees of freedom
## Multiple R-squared:  0.9998,Adjusted R-squared:  0.9998
## F-statistic: 2.497e+05 on 2 and 97 DF,  p-value: < 2.2e-16
```

Similar to when the covariates are uncorrelated, we see that with both X_1 and X_2 in the model, our parameter values are very accurately estimated. However, we will begin to see problems when we remove either X_1 or X_2. Now we will consider the following two models:

$$Y = \beta_0 + \beta_2 X_2 + \varepsilon, \qquad \text{and} \qquad Y = \beta_0 + \beta_1 X_1 + \varepsilon.$$

```
##
## Call:
## lm(formula = Y ~ X2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0577 -3.2520 -0.4856  3.6934  7.8949
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.81401    1.35165  -4.301 4.02e-05 ***
## X2           9.83593    0.05975 164.615  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.911 on 98 degrees of freedom
## Multiple R-squared:  0.9964,Adjusted R-squared:  0.9964
## F-statistic: 2.71e+04 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = Y ~ X1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.867 -14.955   2.536  13.394  24.971
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.3782     4.1272   14.14   <2e-16 ***
## X1            9.9102     0.2534   39.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.99 on 98 degrees of freedom
## Multiple R-squared:  0.9398,Adjusted R-squared:  0.9392
## F-statistic:  1529 on 1 and 98 DF,  p-value: < 2.2e-16
```

In the first R output, the estimate for β_2 is 9.835 which is not very close to it's original set value of 8. This tells us that there could be multicollinearity affecting the estimates, as we would suspect.

Similarly, the β_1 estimate is 9.910, which is very far away from its set value of 2. This is suggesting a completely different relationship between X_1 and Y than what we initially set. By removing either the X_1 or the X_2 term from the model, we see the effects of multicollinearity on the model due to the correlation between these two predictor variables.

4.2 Example: Parameter Estimates

Consider a set of data where the variables are defined as X_1= number of pennies, nickles, and dimes, X_2= total number of coins, and Y= amount of money in pocket. Intuitively, as X_1 changes, this affects the X_2 and Y values as well. We will use the following sample data and put them in vectors we can easily work with.

$$\boldsymbol{X_1} = \begin{bmatrix} 6 \\ 15 \\ 2 \\ 2 \\ 10 \\ 5 \\ 4 \\ 12 \\ 8 \\ 5 \end{bmatrix} \boldsymbol{X_2} = \begin{bmatrix} 10 \\ 25 \\ 4 \\ 3 \\ 14 \\ 7 \\ 7 \\ 18 \\ 13 \\ 7 \end{bmatrix} \boldsymbol{Y} = \begin{bmatrix} \$1.50 \\ \$4.23 \\ \$0.45 \\ \$0.23 \\ \$2.10 \\ \$1.45 \\ \$2.75 \\ \$2.43 \\ \$1.01 \\ \$0.59 \end{bmatrix}$$

Notice when we look at the linear model with just one of the predictor variables, either X_1 or X_2, then the parameter values are both positive values.

```
##
## Call:
## lm(formula = Y ~ X1)
##
## Coefficients:
## (Intercept)           X1
##      0.1136       0.2261
```

```
##
## Call:
## lm(formula = Y ~ X2)
##
## Coefficients:
## (Intercept)           X2
##     0.07181      0.14835
```

However, when we add both X_1 and X_2 to our model, then one of the parameter values turns negative.

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Coefficients:
## (Intercept)           X1           X2
##      0.1215      -0.3077       0.3404
```

We know that the relationship between X_1 and X_2 and Y should all be positive, because if the number of coins increases in either category, then the total number should also increase. Because X_1 and X_2 are highly correlated, we have trouble estimating the parameter values accurately. This is another one of the effects multicollinearity can have on parameter estimates, and why we need methods to fix it.

4.3 Example: Sample From Population

As a final example of multicollinearity effects on parameter estimates, we will simulate a large population, take multiple samples and compare the parameter values over samples. We will first create a population of 10,000 over a uniform distribution with X_1 and X_2 as random values between 5 and 100. We will also set $\beta_0 = 2$, $\beta_1 = 8$ and $\beta_2 = 4$. Now we will take 100 samples of size 100

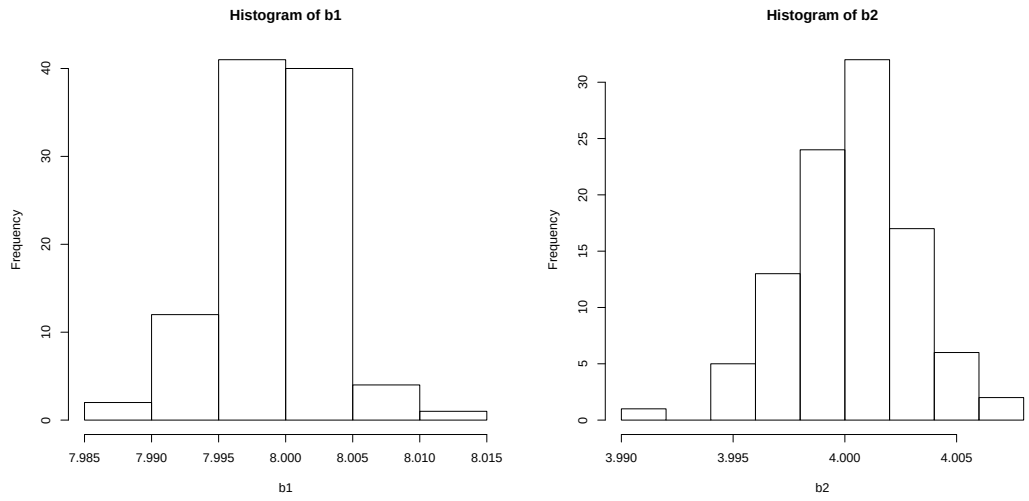


Figure 1: Histograms of b_1 and b_2 values for $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$ model.

from these X_1 and X_2 populations. Figure 1 shows histograms of the estimated β_1 and β_2 values from the 100 samples, when we are using the full model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$.

We should notice that all of the values estimated for β_1 over the 100 samples are between 7.985 and 8.015, which are very good estimates for our $\beta_1 = 8$. Similarly, we have very good estimates for $\beta_2 = 4$, as the 100 samples produced values between 3.990 and 4.005.
Now, let's remove X_2 and examine the estimates for β_1 from the model $Y = \beta_0 + \beta_1 X_1 + \varepsilon$.

We should notice the effects that multicollinearity has on our model in Figure 2. Our estimates for β_1 are now between 6 and 9.5, a much larger range around the actual value of 8. Finally, let's examine the model $Y = \beta_0 + \beta_2 X_2 + \varepsilon$ when we remove X_1.

Once again we see a much larger range of values in Figure 3, from 2 to 5.5, to estimate β_2 which was originally set to 4. The original model estimated our β values very well with both X_1 and X_2 included, but since these two variables were correlated with one another, we begin to see problems when one of them is removed.

5 Matrix Algebra

5.1 The b Vector

We briefly mentioned in Section 2.2, that **b** is the estimated vector of $\boldsymbol{\beta}$. This concept can also be used when looking at multiple linear regression. Taking a closer look at the **b** vector in particular, we know we use **b** to compute the vector of residuals, **e**, but we can also use it to determine the least squares estimator. Let **A'** denote the transpose of matrix **A** and let

$$S(b) = \sum \mathbf{e}_i^2 = \mathbf{e'e} = \mathbf{Y'Y\text{-}Y'Xb\text{-}b'X'Y{+}b'X'Xb}.$$

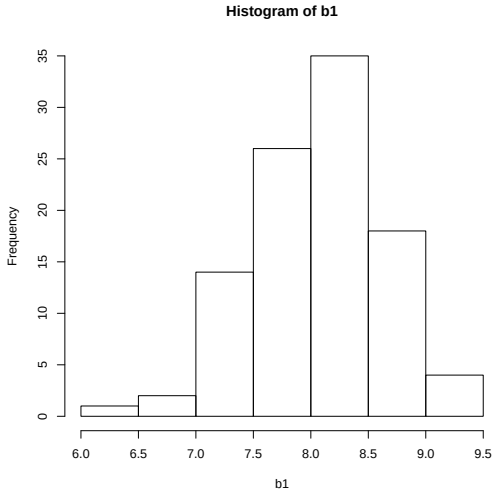


Figure 2: Histogram of b_1 values for $Y = \beta_0 + \beta_1 X_1 + \varepsilon$ model.

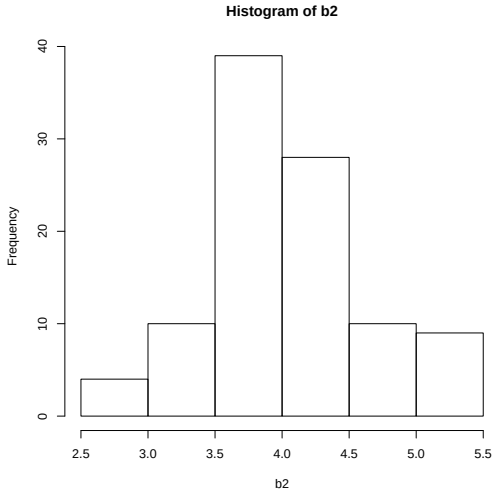


Figure 3: Histogram of b_2 values for $Y = \beta_0 + \beta_2 X_2 + \varepsilon$ model.

Then

$$\frac{d\mathbf{S}}{d\mathbf{b}} = -2\mathbf{X'Y} + 2\mathbf{X'Xb}.$$

Finally, since the least squares estimator is a minimum of $S(\mathbf{b})$, we set

$$\frac{d\mathbf{S}}{d\mathbf{b}} = 0,$$

and solving for $\mathbf{b}$ we get

$$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'Y}.$$

Moving into a more geometric understanding of the $\mathbf{b}$ vector, we can rewrite the residuals as

$$\mathbf{e} = \mathbf{MY}$$

where

$$\mathbf{M} = \mathbf{I\text{-}X(X'X)}^{-1}\mathbf{X'}.$$

We can then write

$$\mathbf{e} = \mathbf{Y} + \hat{\boldsymbol{Y}}$$

where the y-hat matrix, $\hat{\boldsymbol{Y}}$, and the hat matrix, $\mathbf{H}$, are defined as

$$\hat{\boldsymbol{Y}} = \mathbf{Xb} = \mathbf{HY}$$

and

$$\mathbf{H} = \mathbf{X(X'X)}^{-1}\mathbf{X'}$$

respectively.

In Figure 4 we see the geometric representation of the least-squares approximation using these vectors and matrices. We can see that the vectors $\mathbf{Y}$ and $\mathbf{e}$ are orthogonal to each other, and the orthogonal projection of $\mathbf{Y}$ onto the $\mathbf{X}$ plane is equal to the vector $\hat{\boldsymbol{Y}} = \mathbf{Xb}$. By creating this projection, we are minimizing the residual vector $\mathbf{e}$ so that we obtain the linear combination of $\mathbf{Xb}$ of the independent variables that are as close as possible to $\mathbf{Y}$. Relating this back to multiple linear regression directly, we are estimating the $\boldsymbol{\beta}$ vector of $\hat{\boldsymbol{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$, turning this equation into $\hat{\boldsymbol{Y}} = \mathbf{Xb}$ with the estimated $\mathbf{b}$ vector. [Hei+04]

5.2 Variance-Covariance Matrix of b

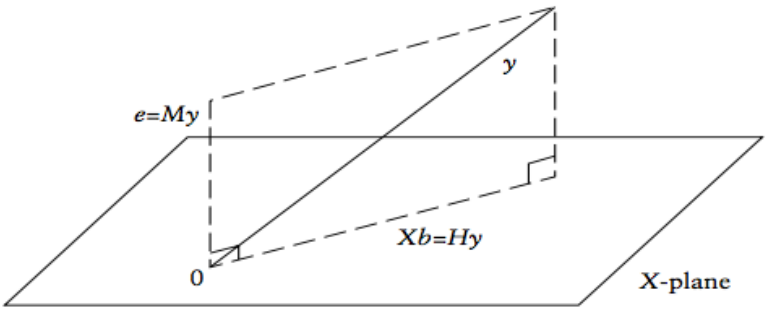
An interesting value to look at is the variance-covariance matrix of $\mathbf{b}$, or $\boldsymbol{\sigma^2}\{\mathbf{b}\}$. First, let us discuss and define variance as we would for simple linear regression. The variance of the sampling distribution of b_1 is

$$\sigma^2 b_1 = \sigma^2 \frac{1}{\sum(X_i - \bar{X})^2}$$

and the variance of the sampling distribution of b_0 is

$$\sigma^2 b_0 = \sigma^2 \left[\frac{1}{n} + \frac{1}{\sum(X_i - \bar{X})^2}\right].$$

Figure 4: Geometric representation of least-squares.



[Hei+04]

Each of these σ^2 values can be estimated using the s^2 or MSE value defined as

$$s^2 = MSE = \frac{\sum e_i^2}{n-2},$$

where the numerator is the sum of the squared residuals, and n is the number of trials in the population.

Next, we will look at the variance-covariance matrix of $\mathbf{b}$, $\boldsymbol{\sigma^2}\{\mathbf{b}\}$, for simple linear regression. The values inside the matrix are as follows,

$$\boldsymbol{\sigma^2}\{\mathbf{b}\} = \begin{bmatrix} \sigma^2\{b_0\} & \sigma\{b_0, b_1\} \\ \sigma\{b_1, b_0\} & \sigma^2\{b_1\} \end{bmatrix},$$

and can be calculated using

$$\boldsymbol{\sigma^2}\{\mathbf{b}\} = \sigma^2(\mathbf{X'X})^{-1}.$$

We can also write this as

$$\boldsymbol{\sigma^2}\{\mathbf{b}\} = \begin{bmatrix} \frac{\sigma^2}{n} + \frac{\sigma^2 \bar{X}^2}{\sum(X_i - \bar{X})^2} & \frac{-\bar{X}\sigma^2}{\sum(X_i - \bar{X})^2} \\ \frac{-\bar{X}\sigma^2}{\sum(X_i - \bar{X})^2} & \frac{\sigma^2}{\sum(X_i - \bar{X})^2} \end{bmatrix}.$$

To estimate this matrix, we can again replace each σ^2 value with the MSE value. The variance, σ^2, essentially tells us the spread of these estimates of regression coefficients and the covariance, σ, tells us the relationship or correlation between the coefficients between two or more random variables. Finally, let's look at how we would estimate the variance-covariance matrix of $\mathbf{b}$ for multiple linear regression in matrix form. This is very similar to what we just looked at in simple linear regression, but with a larger matrix telling us the relationships between the estimated coefficients. So the variance-covariance matrix

$$\boldsymbol{\sigma^2}\{\mathbf{b}\} = \begin{bmatrix} \sigma^2\{b_0\} & \sigma\{b_0, b_1\} & \dots & \sigma\{b_0, b_{p-1}\} \\ \sigma\{b_1, b_0\} & \sigma^2\{b_1\} & \dots & \sigma\{b_1, b_{p-1}\} \\ \vdots & \vdots & & \vdots \\ \sigma\{b_{p-1}, b_0\} & \sigma\{b_{p-1}, b_1\} & \dots & \sigma^2\{b_{p-1}\} \end{bmatrix}$$

can be determined by

$$\boldsymbol{\sigma^2}\{\mathbf{b}\} = \sigma^2(\mathbf{X'X})^{-1}.$$

Note that this equation, confirms the issues we saw with multicollinearity in the simulations from Sections 4.1 and 4.3. The spread of the sampling distribution of the parameter estimates depends on $(\mathbf{X'X})^{-1}$, since this matrix will be nearly singular when the covariates are highly correlated. Finally, the estimated variance-covariance matrix is given by

$$\mathbf{s}^2\{\mathbf{b}\} = MSE(\mathbf{X'X})^{-1}.$$

[KNN04]

5.3 Eigenvalues and Eigenvectors

Let's first consider the definition of eigenvalues and eigenvectors from Chapter 5 of Lay's *Linear Algebra* text [Lay06].

Definition 5.1. An **eigenvector** of an $n \times n$ matrix A is a nonzero vector $\mathbf{x}$ such that $A\mathbf{x} = \lambda\mathbf{x}$ for some scalar λ. A scalar λ is called an eigenvalue of A if there is a nontrivial solution $\mathbf{x}$ of $A\mathbf{x} = \lambda\mathbf{x}$; such an $\mathbf{x}$ is called an *eigenvector corresponding to* λ.

For example, let A=$\begin{bmatrix} 2 & 7 \\ -1 & -6 \end{bmatrix}$. Then an eigenvector of A would be $\mathbf{u}$=$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and it's corresponding eigenvalue would be $\lambda = -5$, since

$$A\mathbf{u} = \begin{bmatrix} 2 & 7 \\ -1 & -6 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -5 \end{bmatrix} = -5 \begin{bmatrix} -1 \\ 1 \end{bmatrix} = -5\mathbf{u}.$$

A couple of important theorems pertaining to eigenvalues and eigenvectors that may be used with Principal Component Analysis are as follows:

Theorem 5.1. If $\mathbf{v}_1$,...,$\mathbf{v}_r$ are eigenvectors that correspond to distinct eigenvalues $\lambda_1, ..., \lambda_r$ of an $n \times n$ matrix A, then the set $\{\mathbf{v}_1,...,\mathbf{v}_r\}$ is linearly independent.

Theorem 5.2. An $n \times n$ matrix A is diagonalizable if and only if A has n linearly independent eigenvectors.

Theorem 5.3. Let A be a real 2×2 matrix with a complex eigenvalue $\lambda = a - bi (b \neq 0)$ and an associated eigenvector $\mathbf{v}$ in C^2. Then

$$A = PCP^{-1}, \quad \text{where} \quad P = [Re\mathbf{v} \;\; Im\mathbf{v}] \quad \text{and} \quad C = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}.$$

[Lay06]

5.4 Spectral Decomposition

The *spectrum* of a matrix $\mathbf{A}$ is the set of eigenvalues of $\mathbf{A}$. A theorem describing the spectrum and some of its properties follows:

Theorem 5.4. An $n \times n$ symmetric matrix $\mathbf{A}$ has the following properties:

1. **A** has n real eigenvalues, counting multiplicities.
2. The dimension of the eigenspace for each eigenvalue λ equals the multiplicity of λ as a root of the characteristic equation.
3. The eigenspaces are mutually orthogonal, in the sense that eigenvectors corresponding to different eigenvalues are orthogonal.
4. **A** is orthogonally diagonalizable.

Suppose that $\mathbf{A}=\mathbf{PDP'}$, where the columns of **P** are orthonormal eigenvectors $\boldsymbol{u_1}, ..., \boldsymbol{u_n}$ of **A** and $\lambda_1, ..., \lambda_n$ represent the eigenvalues of **A**. Then

$$\mathbf{A} = \mathbf{PDP'} = \begin{bmatrix} \boldsymbol{u_1} & \dots & \boldsymbol{u_n} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} \boldsymbol{u_1}' \\ \boldsymbol{u_2}' \\ \vdots \\ \boldsymbol{u_n}' \end{bmatrix}.$$

Using matrix multiplication between **P** and **D**, we obtain

$$\mathbf{PDP'} = \begin{bmatrix} \lambda_1 \boldsymbol{u_1} & \dots & \lambda_n \boldsymbol{u_n} \end{bmatrix} \begin{bmatrix} \boldsymbol{u_1}' \\ \vdots \\ \boldsymbol{u_n}' \end{bmatrix}$$

which leads us to the following representation.

The *spectral decomposition* of **A** can be represented using the following equation:

$$\mathbf{A} = \lambda_1 \boldsymbol{u_1} \boldsymbol{u_1}' + \lambda_2 \boldsymbol{u_2} \boldsymbol{u_2}' ... + \lambda_n \boldsymbol{u_n} \boldsymbol{u_n}'.$$

This representation breaks **A** into projection matrices based on the spectrum (or eigenvalues) or **A**. [Lay06]

5.4.1 Example

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 4 & 0 \\ 1 & 0 & 4 \end{bmatrix}.$$

The eigenvalues of **A** are $\lambda_1 = 5$, $\lambda_2 = 4$, $\lambda_3 = 2$, and the eigenvectors of **A** are $\boldsymbol{v_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $\boldsymbol{v_2} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$, and $\boldsymbol{v_3} = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$, respectively.

Making these eigenvectors orthonormal, we get

$$\frac{\boldsymbol{v_1}}{\|\boldsymbol{v_1}\|} = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} \frac{\boldsymbol{v_2}}{\|\boldsymbol{v_2}\|} = \begin{bmatrix} 0 \\ \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \frac{\boldsymbol{v_3}}{\|\boldsymbol{v_3}\|} = \begin{bmatrix} \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \end{bmatrix}.$$

So our **P** and **P'** matrices for the formula $\mathbf{A} = \mathbf{PDP'}$ are

$$\mathbf{P} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix} \mathbf{P'} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix}.$$

Now denote the columns of P by $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ and $\boldsymbol{u_3}$, respectively. Since we have all of the components needed for spectral decomposition, we can use the formula

$$\mathbf{A} = \lambda_1 \boldsymbol{u_1} \boldsymbol{u_1}' + \lambda_2 \boldsymbol{u_2} \boldsymbol{u_2}' ... + \lambda_n \boldsymbol{u_n} \boldsymbol{u_n}'$$

to write the spectral decomposition for **A** as

$$\mathbf{A} = 5\boldsymbol{u_1}\boldsymbol{u_1}' + 4\boldsymbol{u_2}\boldsymbol{u_2}' + 2\boldsymbol{u_3}\boldsymbol{u_3}'.$$

[Lay06]

6 Principal Component Analysis

In order to avoid the problems we've seen in previous examples regarding multicollinearity and predicting values, we can use a process called principal component analysis.
This process is a dimension reduction tool used to reduce a large set of correlated predictor variables to a smaller, less correlated set, called principal components, that still contains most of the information in the larger set.
The first principal component contains as much of the variability in the data as possible, and the principal components following the first, account for remaining variability as much as they possibly can. The analysis is usually performed on a square symmetric matrix, such as the covariance matrix which was explained in Section 5.2.

Definition 6.1. The **principal components** for a set of vectors are a set of linear combinations of the vectors, chosen so that this captures the most information in a smaller subset of vectors.

Even though this method may seem like a foolproof way to handle problems that multicollinearity causes, there is no guarantee that the new dimensions are interpretable after dimension reduction. Sometimes, when a variable is left out, important information and variance of the data is also removed so we aren't able to estimate parameters accurately.

6.1 Determining Principal Components: Method One

Suppose we have a random vector **X** with p components, and **X** has population variance-covariance matrix

$$\sigma^2(\mathbf{X}) = \sum = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1p}\} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2p} \\ \vdots & \vdots & & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_p^2 \end{bmatrix}.$$

The first goal is to look for a linear function $\alpha_1' x$ of the elements of x having maximum variance

and where α_1 is a vector of p constants, so $\alpha_1' x = \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1p}x_p = \sum_{j=1}^{p} \alpha_{1j}x_j$. We then continue this process until the kth $\alpha_k' x$ for the kth principal component. Up to p principal components can be found, but usually most of the variation in x will be accounted for by k principal components, with $k < p$. [Jol02]

6.2 Determining Principal Components: Method Two

A large part of determining principal components is manipulating the X matrix. Here, we will start with the X matrix of data and nine predictor variables and walk through the process. We assume first that X has n rows and $k+1$ columns, and the first column is the one-vector with the next k columns are x_1 through x_k. We also need the columns to be centered to have mean zero. Unfortunately with most data, this is not the case and we need to subtract out the mean of each column to obtain mean zero for each variable. We then have

$$X'X = \begin{pmatrix} n & 0 \\ 0 & A \end{pmatrix}.$$

Taking the eigenvalues and eigenvectors of A, we let $V = diag(\lambda_1, ..., \lambda_k)$ where $diag()$ denotes the diagonal matrix, and U have $u_1, u_2, ..., u_k$ as columns. Using the spectral decomposition of A, we get $U'\bar{X}'\bar{X}U = A$.
Finally, let $P = \bar{X}U$, where each column of P, p_i, is a linear combination of the columns of X and we call the p_i's principal components of the predictor variables. Since these principal components are linear combinations of the covariates, they are in the column space spanned by the covariates.

6.3 Example: Finding and Interpreting Principal Components

We will introduce a new data set for this example called Protein Consumption in Europe. [Web73] The predictor variables in this example represent nine different food groups, namely: Red Meat, White Meat, Eggs, Milk, Fish, Cereal, Starch, Nuts, and Fruits/Vegetables. These nine variables measure protein consumption in 25 different European countries. We will only be working with the predictor variables here to demonstrate how to find principal components using matrix representation.
We will use Method Two to manipulate the X matrix in R to determine the principal components for this data. So we start with X as a 25×9 matrix, with the covariates in the same order as listed above. We won't walk through each step of the method, since it is mostly just manipulating matrices until we reach the principal component matrix. Below is the output from our R code looking at the matrix U, with it's columns as the eigenvectors from the A matrix.

```
##             [,1]        [,2]        [,3]         [,4]        [,5]
##  [1,] -0.3026094  0.05625165  0.29757957 -0.646476536 -0.32216008
##  [2,] -0.3105562  0.23685334 -0.62389724  0.036992271  0.30016494
##  [3,] -0.4266785  0.03533576 -0.18152828 -0.313163873 -0.07911048
##  [4,] -0.3777273  0.18458877  0.38565773  0.003318279  0.20041361
##  [5,] -0.1356499 -0.64681970  0.32127431  0.215955001  0.29003065
##  [6,]  0.4377434  0.23348508 -0.09591750  0.006204117 -0.23816783
##  [7,] -0.2972477 -0.35282564 -0.24297503  0.336684733 -0.73597332
##  [8,]  0.4203344 -0.14331056  0.05438778 -0.330287545 -0.15053689
```

```
##  [9,]  0.1104199 -0.53619004 -0.40755612 -0.462055746  0.23351666
##              [,6]        [,7]        [,8]       [,9]
##  [1,] -0.45986989  0.15033385 -0.01985770 -0.2459995
##  [2,] -0.12100707 -0.01966356 -0.02787648 -0.5923966
##  [3,]  0.36124872 -0.44327151 -0.49120023  0.3333861
##  [4,]  0.61843780  0.46209500  0.08142193 -0.1780841
##  [5,] -0.13679059 -0.10639350 -0.44873197 -0.3128262
##  [6,]  0.08075842  0.40496408 -0.70299504 -0.1522596
##  [7,]  0.14766670  0.15275311  0.11453956 -0.1218582
##  [8,]  0.44701001 -0.40726235  0.18379989 -0.5182749
##  [9,]  0.11854972  0.44997782  0.09196337  0.2029503
```

For the first principal component, in the first column of U, we see that Cereal and Nuts have the largest positive values, and RedMeat, WhiteMeat, Eggs and Milk all have large negative values. So we could interpret this as an "overall protein consumption" variable, meaning that people who eat a lot of all food categories will have high protein intake. In the second principal component we see that WhiteMeat and Cereal have higher positive values, and Fish, Starch and Fruits/Vegetables have higher negative values. Since the negative values have more weight than the positive values, we will consider these values. We can call this the "Fish, Starch and Fruits/Vegetables" variable, meaning when people eat only foods from these categories, then their protein intake may suffer. Looking at the third principal component, we see that WhiteMeat and Fruits/Vegetables have high negative values, while RedMeat, Milk and Fish all have positive values that are significant. This indicates that the amount of RedMeat, Milk and Fish that people eat positively affects their protein intake. [Web73]

6.4 Example: Principal Component Analysis and Linear Regression

Introducing a new example, we will be examining the response variable body fat, and the predictor variables defined below.

FAT: based on the immersion method, expressed as a percent of total weight
SKIN: a measure of the triceps skinfold thickness in millimeters
THIGH: thigh circumference in centimeters
ARM: mid-arm circumference in centimeters

The data is based on 20 female subjects, between the ages of 25-34 years. [Jol02]

We should notice intuitively, that as any of the predictor variables increase, we should also see an increase in the response variable. Here is the beginning $\mathbf{X}$ design matrix with the columns as SKIN, THIGH, ARM, and FAT, from left to right.

$$\mathbf{X} = \begin{bmatrix} 19.5 & 43.1 & 29.1 & 11.9 \\ 24.7 & 49.8 & 28.2 & 22.8 \\ 30.7 & 51.9 & 37.0 & 18.7 \\ 29.8 & 54.3 & 31.1 & 20.1 \\ 19.1 & 42.2 & 30.9 & 12.9 \\ 25.6 & 53.9 & 23.7 & 21.7 \\ 31.4 & 58.5 & 27.6 & 27.1 \\ 27.9 & 52.1 & 30.6 & 25.4 \\ 22.1 & 49.9 & 23.2 & 21.3 \\ 25.5 & 53.5 & 24.8 & 19.3 \\ 31.1 & 56.6 & 30.0 & 25.4 \\ 30.4 & 56.7 & 28.3 & 27.2 \\ 18.7 & 46.5 & 23.0 & 11.7 \\ 19.7 & 44.2 & 28.6 & 17.8 \\ 14.6 & 42.7 & 21.3 & 12.8 \\ 29.5 & 54.4 & 30.1 & 23.9 \\ 27.7 & 55.3 & 25.7 & 22.6 \\ 30.2 & 58.6 & 24.6 & 25.4 \\ 22.7 & 48.2 & 27.1 & 14.8 \\ 25.2 & 51.0 & 27.5 & 21.1 \end{bmatrix}.$$

We will be using the steps of finding principal components based on Method Two in Section 6.2. All of the matrix manipulation steps are included in the R code and we will look at the R outputs of the values and matrices we need.

We will first analyze the original, full model,

$$\text{FAT} = \beta_0 + \beta_1 \text{SKIN} + \beta_2 \text{THIGH} + \beta_3 \text{ARM} + \varepsilon.$$

Regressing the data onto FAT, we obtain:

$$\text{FAT} = 20.20 + 4.33\text{SKIN} - 2.86\text{THIGH} - 2.18\text{ARM} + \varepsilon.$$

Looking at the β values for this model, we see that two of them are negative values. Based on the original analysis of the data, we shouldn't expect any negative relationship Using the same matrix manipulation process from Section 6.3, we can obtain the eigenvector matrix $\mathbf{U}$ displayed below:

$$\mathbf{U} = \begin{bmatrix} 0.695 & 0.050 & 0.718 \\ 0.629 & 0.441 & 0.640 \\ 0.348 & 0.896 & 0.274 \end{bmatrix}.$$

Using the columns of the eigenvector matrix $\mathbf{U}$, we can create three principal components that are linear combinations of SKIN, THIGH, and ARM.

$$p_1 = 0.695SKIN + 0.629THIGH + 0.348ARM$$
$$p_2 = -0.050SKIN - 0.441THIGH + 0.896ARM$$
$$p_3 = -0.718SKIN + 0.640THIGH + 0.274ARM$$

We regress p_1, p_2 and p_3 onto FAT, just as we would normally regress covariates. We then obtain the following summary of the model.

```
##
## Call:
## lm(formula = y ~ p1 + p2 + p3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7263 -1.6111  0.3923  1.4656  4.1277
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.1950     0.5545  36.418  < 2e-16 ***
## p1            12.7286     1.7261   7.374 1.57e-06 ***
## p2            -7.2340     2.5682  -2.817   0.0124 *
## p3          -119.3463    91.9924  -1.297   0.2129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.48 on 16 degrees of freedom
## Multiple R-squared:  0.8014,Adjusted R-squared:  0.7641
## F-statistic: 21.52 on 3 and 16 DF,  p-value: 7.343e-06
```

Looking at the p-value for the third principal component, we see that it is significantly higher than the other p-values. This indicates, that this third principal component isn't as significant in the estimation of parameters as the other two. So we will remove this principal component and transform back to the original data using the following process.

We will use this formula:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{pc}} = \boldsymbol{U}\hat{\boldsymbol{\gamma}}_{\boldsymbol{pc}},$$

where $\hat{\boldsymbol{\gamma}}_{\boldsymbol{pc}}$ is the $\boldsymbol{U}$ matrix, with the principal component p_3 replaced with 0's. So in this case,

$$\hat{\boldsymbol{\gamma}}_{\boldsymbol{pc}} = \begin{bmatrix} 0.695 & 0.050 & 0 \\ 0.629 & 0.441 & 0 \\ 0.348 & 0.896 & 0 \end{bmatrix}.$$

Thus, we obtain $\hat{\boldsymbol{\beta}}_{\boldsymbol{pc}}$, which will generally not contain zero elements, because elimination of variables in the orthogonal model does not result in elimination of variables in the original model.

So, after transforming back to the original data, we will create a new, full model once again. The model is as follows:

$$FAT = -17.309 + 0.483SKIN + 0.479THIGH + 0.037ARM + \varepsilon.$$

Notice that with this new model, all of the parameter estimates are positive, which is what we expected originally. The process of analyzing and removing principal components in the model has worked, and we have found a model and parameter estimates that correctly summarize the data.

6.5 Other Uses for Principal Component Analysis

Principal component analysis is used in almost all scientific disciplines to analyze data. For example, in neuroscience, this method is used to detect coordinated activities of large neuron ensembles. Some of the other disciplines that use principal component analysis include biology, forestry, business, chemistry, criminology, educational research, psychology, sociology, and sports for example. There are also extensions of this process such as Correspondence Analysis and Multiple Factor Analysis.

7 Conclusion

Highly correlated covariates in a multiple regression model cause many issues with analysis of data. Realize that principal component analysis cannot always fix the parameter estimation problems caused by multicollinearity, but this process is often effective. Not only is PCA used in statistics but also in many other disciplines and real world applications.

Acknowledgement

I would like to acknowledge Professor Stacy Edmondson for her assistance and guidance throughout the writing of this report and presentation.
I would also like to acknowledge Professor Albert Schueller for leading the Senior Project course, editing drafts, and organizing presentation practice.

References

[Web73] A. Weber. *Agrarpolitik im Spannungsfeld der internationalen Ernaehrungspolitik*. 1973. URL: `http://dasl.datadesk.com/data/view/93`.

[Jol02] I.T. Jolliffe. *Principal Component Analysis Springer Series in Statistics*. Springer Science Business Media, 2002.

[Hei+04] Christiaan Heij et al. *Econometric Methods with Applications in Business and Economics*. 2004.

[KNN04] Michael H. Kutner, Christopher J. Nachtsheim, and John Neter. *Applied Linear Regression Models*. New York, NY: McGraw-Hill/Irwin, 2004.

[Lay06] David C. Lay. *Linear Algebra and its applications*. Boston, MA: Pearson Education, Inc., 2006.