PCA & FA of Satellite Data

Greg Johnson

The data are the set of 2512 pixels (with surrounding pixels completely contained in the image) that make up a 82×100 pixel sub-area of a scene. The 36 predictors are the 8-bit binary words for each pixel and its surrounding eight pixels in terms of the 4 spectral bands ($4 \times 9 = 36$).

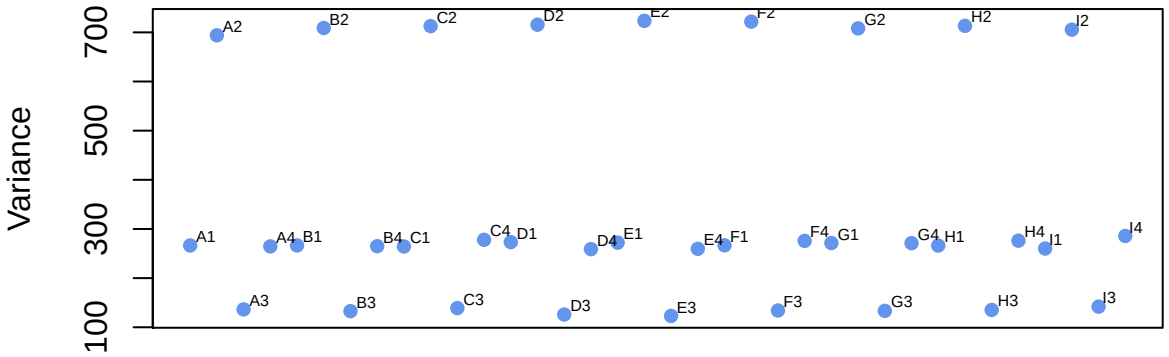
```
satellite<-read.table("data/landsat.txt",header=TRUE)
```

Principal Component Analysis

PCA of S Versus R

PCA of unstandardized variables (i.e PCA of Σ) is not equivalent to PCA of standardized variables (i.e. PCA of ρ). In fact the principal components derived in one situation are not a simple function of those derived in the other. When there is a disparity in the scales of the variables, the variances differ those with larger variances dominate (have larger coefficients for) the principal components.

```
plot(sapply(satellite[, -37], var), pch = 16, col = "cornflowerblue", xlab = "",
    ylab = "Variance", main = "Variances of Predictors", xaxt = "n")
textxy(1:36, sapply(satellite[, -37], var), names(satellite)[-37])
```



As we can see from the above plot, the predictors from the second spectral band; those from the third band; and those from the first and second spectral bands differ greatly in their variances.

We can also see this in the differences between principal component coefficients for the unstandardized variables and standardized variables:

```
PC1Sloadings<-princomp(satellite[,-37],cor=FALSE)[["loadings"]][,1] #1st PC of Sigma
PC1Rloadings<-princomp(satellite[,-37],cor=TRUE)[["loadings"]][,1] #1st PC of R
round(cbind(PC1Sloadings,PC1Rloadings),2)
```

```
##    PC1Sloadings PC1Rloadings
## A1        -0.15        -0.19
## A2        -0.27        -0.21
## A3         0.00         0.03
## A4         0.11         0.17
## B1        -0.15        -0.20
## B2        -0.27        -0.21
```

```
## B3         0.00         0.04
## B4         0.11         0.17
## C1        -0.15        -0.19
## C2        -0.27        -0.20
## C3         0.00         0.04
## C4         0.11         0.16
## D1        -0.15        -0.20
## D2        -0.28        -0.21
## D3         0.00         0.04
## D4         0.12         0.18
## E1        -0.15        -0.20
## E2        -0.29        -0.21
## E3         0.00         0.04
## E4         0.12         0.18
## F1        -0.15        -0.20
## F2        -0.28        -0.21
## F3         0.00         0.04
## F4         0.12         0.17
## G1        -0.15        -0.19
## G2        -0.27        -0.20
## G3         0.00         0.03
## G4         0.11         0.16
## H1        -0.15        -0.19
## H2        -0.28        -0.21
## H3         0.00         0.04
## H4         0.12         0.17
## I1        -0.15        -0.19
## I2        -0.27        -0.21
## I3         0.00         0.04
## I4         0.11         0.17
```

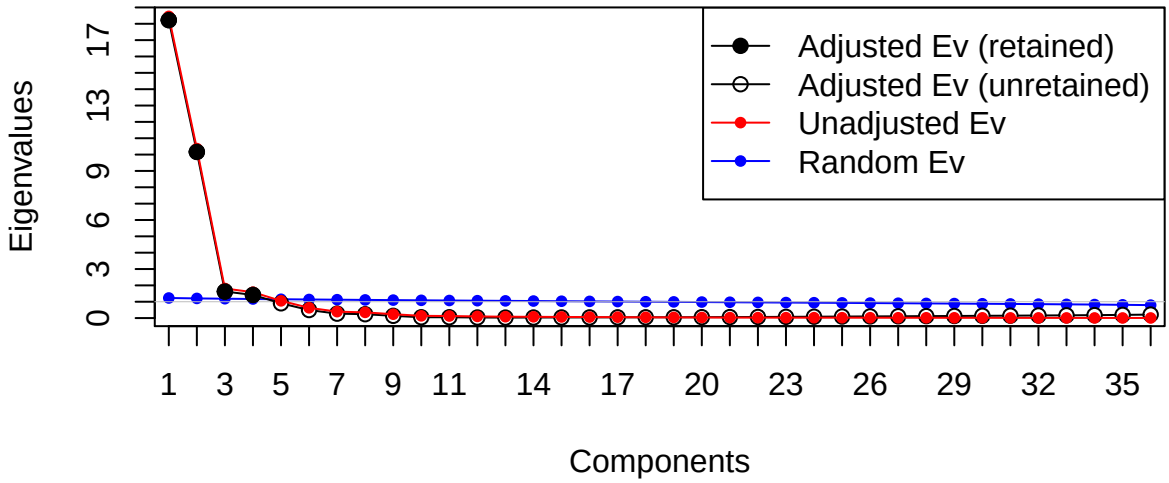
Notice that when applying PCA to Σ, the large variance of predictors from the second spectral band gives them much larger coefficients than predictors from the other bands. When applying PCA to ρ, the variances are standardized so the predictors from the second spectral band no longer dominate; now predictors from the first spectral band have comparably large coefficients even though they originally had smaller variances.

Perform PCA Model Selection and Interpret the Final Model

Principal Component Analysis model selection essentially just comes down to finding the correct number of principal components to retain. There are a number of ways of going about this. From a theoretical standpoint, the best modern methods is Horn's Parallel Analysis (HPA). From a pragmatic standpoint, we have a lot of pixel data that we want to minimize as much as possible in the interest of database storage whilst keeping the fidelity of the original image. Thus the best criterion for the application may be the cumulative amount of variation explained. We will pursue both avenues.

```
paran(satellite[,-37],quietly=TRUE,graph=TRUE)
```

```
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
```



HPA adjusts observed eigenvalues for random variation by generating uncorrelated data and using these "null eigenvalues" as corrections. HPA suggests retention of four principal components.

If we instead look at proportion of variance explained:

```
round(eigen(cor(satellite[,-37]))$values/36,2) #prop. of var.
```

```
##  [1] 0.51 0.29 0.05 0.04 0.03 0.02 0.01 0.01 0.01 0.00 0.00 0.00 0.00 0.00
## [15] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## [29] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
round(cumsum(eigen(cor(satellite[,-37]))$values/36),2) #cum. prop. of var.
```

```
##  [1] 0.51 0.80 0.85 0.89 0.92 0.94 0.95 0.96 0.97 0.97 0.98 0.98 0.98 0.98
## [15] 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 1.00 1.00 1.00 1.00 1.00
## [29] 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

The first four eigenvalues explain 89% of the variance which is a significant portion of the original variance. If we follow the suggestion of HPA, we at least know that a significant portion of the original variance will be captured by the 4 principal component solution. By that same token, the first 2 principal components account for 80% of the data so if our goal is to preserve the original variation in the data but reduce the number of variables we need to store as much as possible, it's arguably better to store two variables with 80% fidelity than four variables with 89% fidelity.

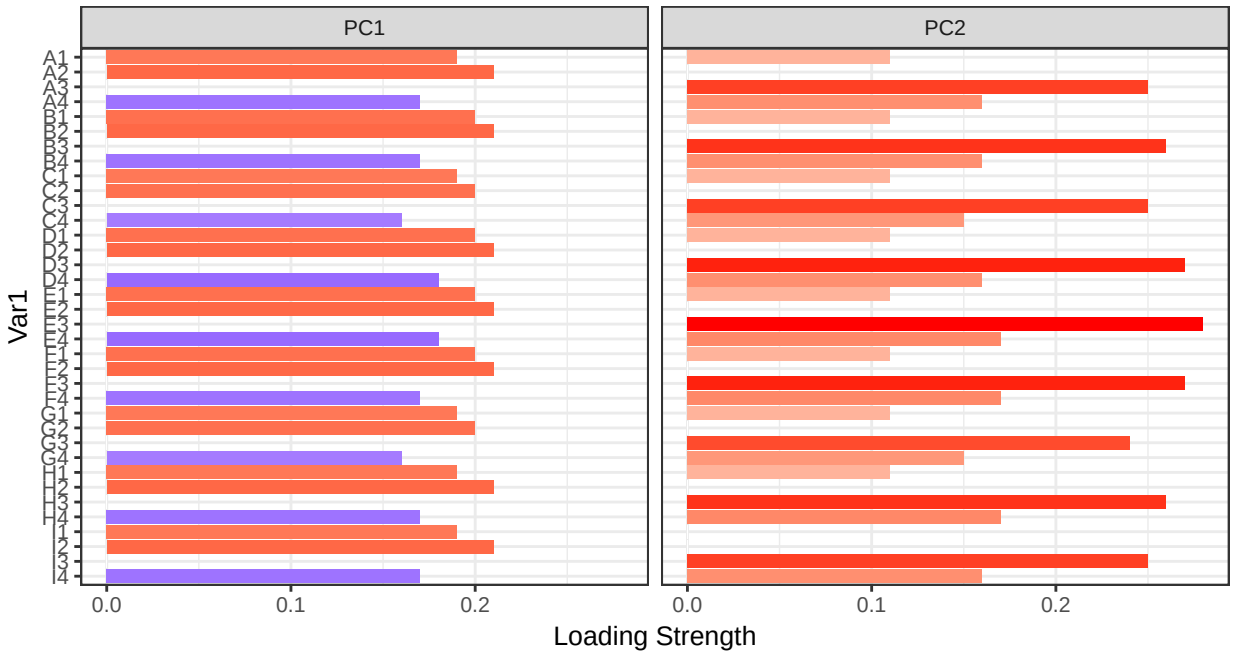
So we fit a PCA with 2 components:

```
pcafit<-princomp(satellite[,-37],cor=TRUE)
load<-round(eigen(cor(satellite[,-37]))$vectors[,1:2],2)
rownames(load)<-names(satellite)[-37]
colnames(load)<-c("PC1","PC2")
load[abs(load)<=.1]<-0#using 0.1 as the cutoff
```

The first principal component appears to be the difference between the average of the spectrum 1 and spectrum 2 predictors, and spectrum 4 predictors. The second PC is a weighted average between the spectrum 1,3, and 4 predictors. The following heat map of loadings helps us visualize the contributions of the different spectral predictors to the two different principal components.

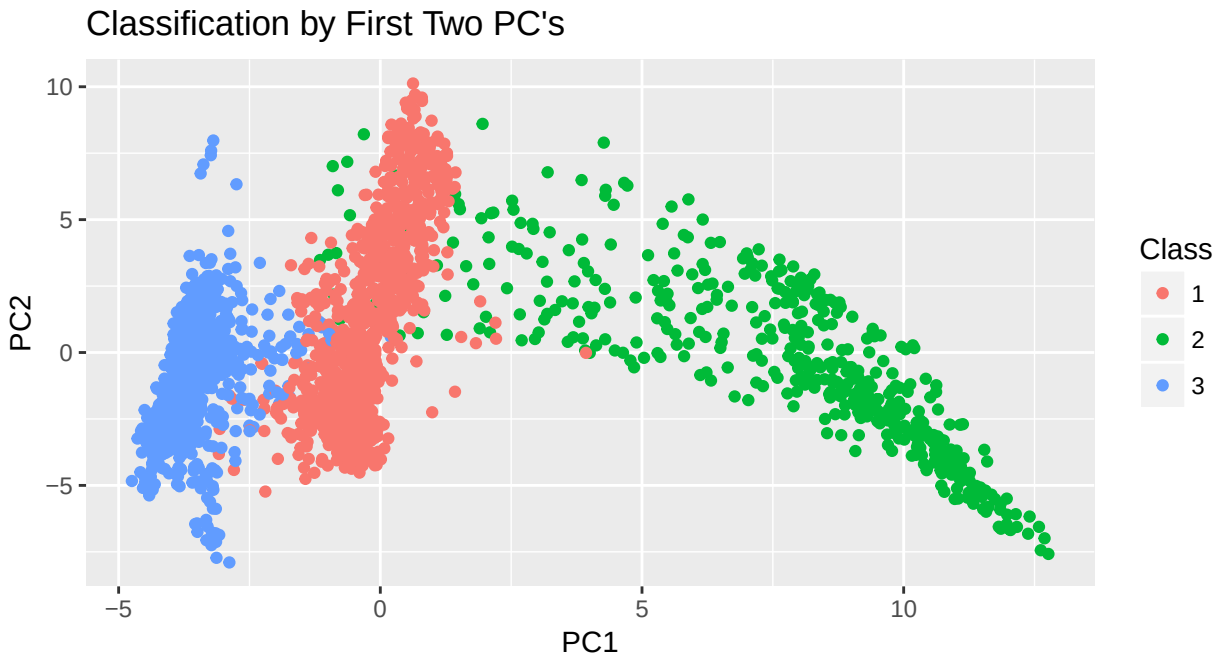
```
ggplot(melt(load[36:1,]), aes(Var1, abs(value), fill=value)) +
  facet_wrap(~ Var2, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
```

```
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-and-white theme with set font size
```



Fidelity of Two Component PCA in Terms of Classification

```
pcscores<-pcafit$scores[,1:2]
pcscores<-as.data.frame(pcscores)
pcscores[["class"]]<-factor(satellite$class)
names(pcscores)<-c("PC1","PC2","Class")
ggplot(aes(x=PC1,y=PC2,col=Class),data=pcscores)+geom_point()+labs(xlab="Principal Component 1",ylab="P
```



It looks like we can retain only 2 principal components (80% of the variance in the original 36 predictors) and still get nice separation of the three classes.

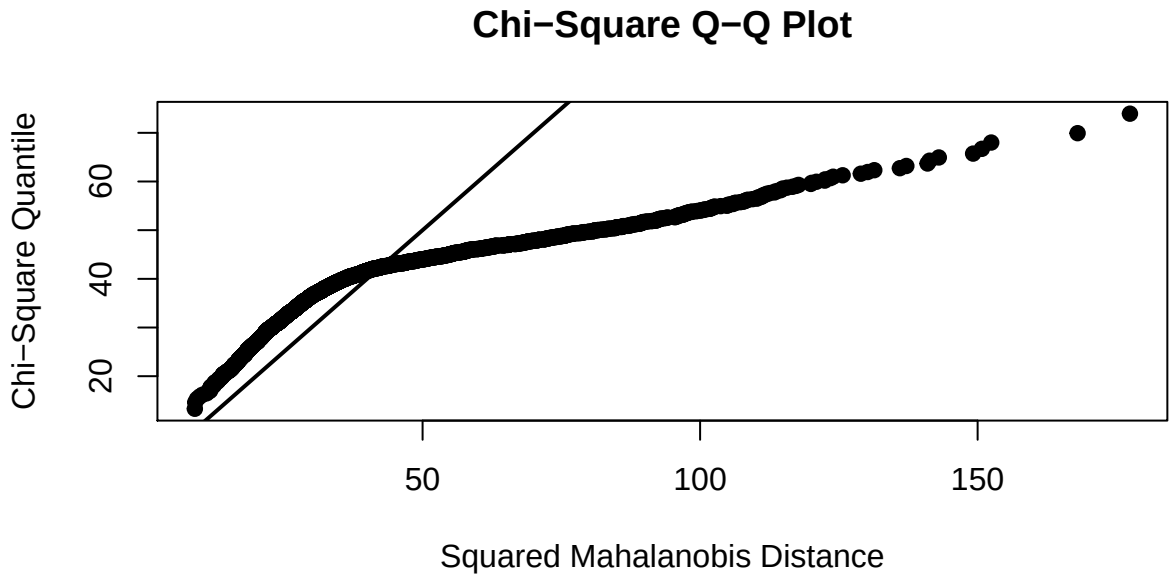
Factor Analysis

Perform FA Model Selection and Interpret the Final Model

Factor Analysis model selection comes down to selection of estimation method, number of factors, and rotation.

Estimation Method. Maximum likelihood is the preferred method since it comes with goodness-of-fit indices and significance tests of loadings. However it requires multivariate normal of the common and unique factors which we test by checking the multivariate normality of our data:

```
mardiaTest(satellite[,-37],qqplot=TRUE)
```



```
##    Mardia's Multivariate Normality Test
## ---------------------------------------
##    data : satellite[, -37]
##
##    g1p            : 118.2489
##    chi.skew       : 49506.88
##    p.value.skew   : 0
##
##    g2p            : 1780.419
##    z.kurtosis     : 197.588
##    p.value.kurt   : 0
##
##    chi.small.skew : 49569.2
##    p.value.small  : 0
##
##    Result          : Data are not multivariate normal.
## ---------------------------------------
```

Our data are decidedly non-normal. So we will pursue the classic Principal Component method of Factor Analysis.

Number of Factors. Horn's Parallel Analysis applies to Factor Analysis as well. The eigenvalues don't change between PCA and FA so HPA's recommendation stays the same: 4 factors. As with PCA, we will stick with 2 factors.

Rotation. We will utilize a Varimax (orthogonal) rotation to attempt to achieve some sort of simple structure for interpretation purposes.

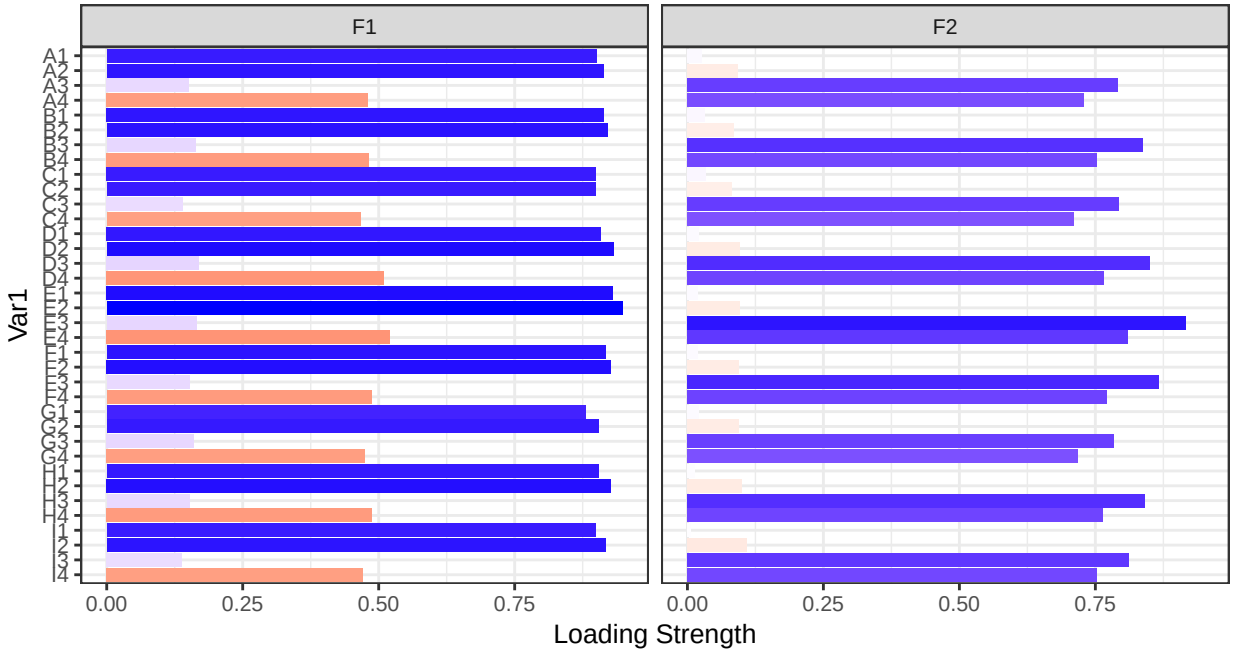
```
require(robustfa)
fafit<-factorScorePca(scale(satellite[,-37]),factors=2,rotation="varimax",scoresMethod="regression")
load<-matrix(as.numeric(loadings(fafit)),36,2,dimnames=list(names(satellite)[-37],c("F1","F2")))
load
```

```
##            F1           F2
## A1  0.9003283  0.025985149
## A2  0.9129493 -0.092953683
## A3  0.1515057  0.790827388
## A4 -0.4804339  0.729732991
## B1  0.9139195  0.032021250
## B2  0.9203699 -0.085245528
## B3  0.1631173  0.837495587
## B4 -0.4825812  0.752151836
## C1  0.8991148  0.034394474
## C2  0.8985726 -0.081700152
## C3  0.1406913  0.792498062
## C4 -0.4662511  0.710907726
## D1  0.9091276  0.021942587
## D2  0.9319498 -0.096979479
## D3  0.1700608  0.850726730
## D4 -0.5099322  0.765958294
## E1  0.9305071  0.018726181
## E2  0.9483405 -0.096855385
## E3  0.1653828  0.916445878
## E4 -0.5207761  0.809598207
## F1  0.9169559  0.018671637
## F2  0.9271769 -0.094956436
```

```
## F3  0.1520328  0.867334369
## F4 -0.4863795  0.771219304
## G1  0.8804019  0.020441137
## G2  0.9044902 -0.095139272
## G3  0.1605013  0.784293312
## G4 -0.4752708  0.718787275
## H1  0.9043772  0.013512863
## H2  0.9270195 -0.101073667
## H3  0.1519308  0.841945532
## H4 -0.4878754  0.763097741
## I1  0.8988323  0.006752744
## I2  0.9167576 -0.108576021
## I3  0.1386101  0.812399415
## I4 -0.4711699  0.752714716
```

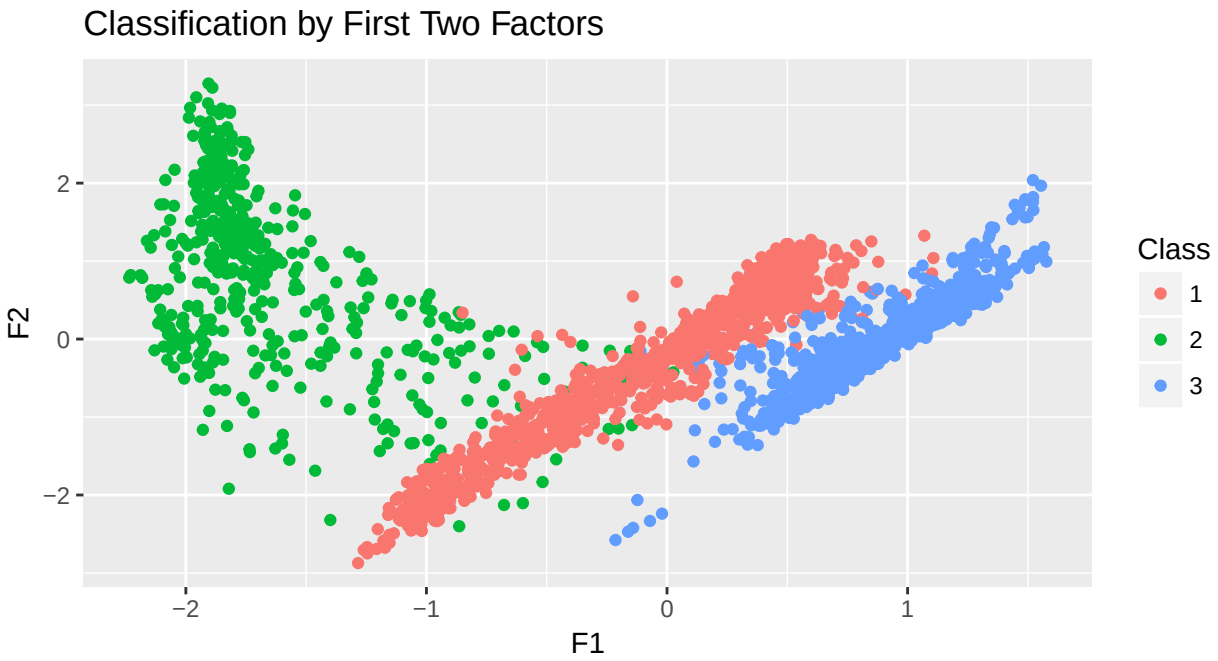
The predictors from the first two spectra have strong positive loadings on the first factor; those from the fourth spectra have moderate negative loadings. The predictors from the third and fourth spectra have strong positive loadings on the second factor.

```
ggplot(melt(load[36:1,]), aes(Var1, abs(value), fill=value)) +
  facet_wrap(~ Var2, nrow=1) + #place the factors in separate facets
  geom_bar(stat="identity") + #make the bars
  coord_flip() + #flip the axes so the test names can be horizontal
  #define the fill color gradient: blue=positive, red=negative
  scale_fill_gradient2(name = "Loading",
                       high = "blue", mid = "white", low = "red",
                       midpoint=0, guide=F) +
  ylab("Loading Strength") + #improve y-axis label
  theme_bw(base_size=10) #use a black-and-white theme with set font size
```



Fidelity of Two-Factor FA in Terms of Classification

```
fascores<-fafit$scores[,1:2]
fascores<-as.data.frame(fascores)
fascores[["class"]]<-factor(satellite$class)
names(fascores)<-c("F1","F2","Class")
ggplot(aes(x=F1,y=F2,col=Class),data=fascores)+geom_point()+labs(xlab="Factor 1",ylab="Factor 2",title=
```



The first two factor scores do a relatively clean job of separating the three classess. The second class shows a little bit of overlap with the first class but overall the classification power, visually, appears to be high.

Comparison of PCA and FA solutions

The first principal component and factor are similar in the relative contributions from the various predictors however the signs are opposite. This is find considering the sign of the PCA solution is non-unique.

The big difference comes from the second principal component and factor - the second spectrum predictors have a strong contribution to the former, but not to the latter! This is most likely due to the non-uniqueness of the factor solution. Because of the indeterminacy of the factor solution (it's only determined up to an orthogonal rotation), we can rotate the factors however we want to satisfy what the textbook calls a "Wow" factor. In short, the discrepancy between PCA and FA for the second PC/factor is probably due to the Varimax rotation we applied to FA.