36-401 Class Notes
Data Reduction I: Principal Components

Problem: Many Collinear Variables

Problems with correlation and multicollinearity in the X's:

- More uncertainty (higher Std Err's) in estimation of individual coefficients;
- Interpretations based on changing one covariate at a time less meaningful;
- Model selection more complex; may result in many similarly good models.
- Horrid multicollinearity: "Error in lm.fit.qr(x, y): computed fit is singular, rank K"
- Bad multicollinearity: Correlation of *coefficients* greater than e.g. 0.9 in absolute value.

Example

The data are from an unpublished Master's Thesis in Civil Engineering by Carl Hoffstedt, as reported by Weisberg (1985, p. 206). They relate the accident report rate, in accidents per million vehicle miles (y) to 13 potential independent variables. The data include 39 sections of large highways in the state of Minnesota in 1973. The variables are:

```
Rate - 1973 accident rate per million vehicle miles
Len  - length of segment in miles
Adt  - average daily traffic count in thousands (estimated)
Trk  - truck volume as a percent of total volume
Sp   - speed limit (in 1973, before the 55 mph limits)
lw   - lane width in feet
shl  - width in feet of outer shoulder on the roadway
itg  - number of freeway-type interchanges per mile in the segment
sigs - number of signalized interchanges per mile in the segment
acpt - number of access points per mile in the segment
lan  - total number of lanes of traffic in both directions
type -  1 if Federal aid interstate highway,
        2 if principal arterial interstate highway,
        3 if minor arterial interstate highway,
        4 otherwise
```

(The full dataset is `highway.dat` on the class website).

We start with a naive regression analysis:

```
> high _ read.table("highway.dat",header=T)
> high[1:5,]
  Rate   Len Adt Trk Sp lw shl  itg sigs acpt lan type
1 4.58  4.99  69   8 55 12  10 1.20    0  4.6   8    1
2 2.86 16.11  73   8 60 12  10 1.43    0  4.4   4    1
3 3.02  9.75  49  10 60 12  10 1.54    0  4.7   4    1
4 2.29 10.65  61  13 65 12  10 0.94    0  3.8   6    1
5 1.61 20.01  28  12 70 12  10 0.65    0  2.2   4    1

> X _ high[,-1]
> y _ high[,1]

> rawlm _ lm(Rate ~ . , data=high)
> summary(rawlm)

Call: lm(formula = Rate ~ ., data = high)
Residuals:
    Min      1Q  Median     3Q   Max
 -1.867 -0.8856 0.09851 0.6906 2.887

Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept)  12.5190   6.5290     1.9175   0.0658
        Len  -0.0564   0.0328    -1.7222   0.0965
        Adt   0.0093   0.0326     0.2841   0.7785
        Trk  -0.1271   0.1057    -1.2026   0.2396
         Sp  -0.0685   0.0637    -1.0756   0.2916
         lw  -0.3433   0.4967    -0.6912   0.4954
        shl  -0.0947   0.1245    -0.7606   0.4535
        itg   0.2868   1.1426     0.2510   0.8037
       sigs   0.4908   0.3941     1.2453   0.2237
       acpt   0.0684   0.0241     2.8376   0.0085
        lan  -0.0131   0.2743    -0.0477   0.9623
       type   0.2338   0.4893     0.4778   0.6366

Residual standard error: 1.172 on 27 degrees of freedom
Multiple R-Squared: 0.7527
F-statistic: 7.472 on 11 and 27 degrees of freedom, the p-value is
    1.051e-05

[ ... ]
```

```
> collin(X,cutoff=0.7)  # R^2 for regressing each X on the other X's
Adt 0.902
Sp 0.739
shl 0.747
itg 0.836
lan 0.741
type 0.743
```

Clearly there is a lot of collinearity!

One approach might be to proceed with variable selection, either "by eye" using t-tests in the summary(lm()) output, or by using the BIC criterion in sum.step(stepwise()).

- t-statistics in the output only support including "Len" (length of hwy segment) and "acpt" (number of access points per mile). But with so much collinearity these results are hard to trust.
- sum.step(sw()) suggests:

```
> args(stepwise)
function(x, y, wt, intercept = T, tolerance = 1e-07, method = "efroymson",
        size.max = ncol(x), nbest = 3, f.crit = 2, xinclude, plot.rss = dev.cur(
        ) != 1, time = 0.02)
NULL
> sw _ stepwise(X,y,meth="exhaustive")
> args(sum.step)
function(step.rslt, data, y, sortcol = 1, sigfig = c(IC = 1, RSE = 1, R = 3),
        includeBIC = T, includeAIC = F, includeRSE = F, includeR2 = F,
        includeR2a = T, top = NULL, compact = F)
NULL
> sum.step(sw,high,"Rate",includeAIC=T,includeR2=T)
          BIC   AIC    R2   R2a Len Adt Trk Sp lw shl itg sigs acpt lan type
 3(#1) -66.0 -62.6 0.694 0.667   1   0   0  1  0   0   0    0    1   0    0
 4(#1) -66.1 -61.9 0.720 0.687   1   0   0  1  0   0   0    1    1   0    0
 4(#2) -66.3 -62.2 0.717 0.683   1   0   1  1  0   0   0    0    1   0    0
 3(#2) -66.3 -63.0 0.688 0.661   0   0   1  1  0   0   0    0    1   0    0
```

The variables in the best several models here are:

```
Len  - length of segment in miles
Trk  - truck volume as a percent of total volume
Sp   - speed limit (in 1973, before the 55 mph limits)
sigs - number of signalized interchanges per mile in the segment
acpt - number of access points per mile in the segment
```

but what are these variables telling us about the type of road that is accident-prone?

Principal Components

1. A way to reduce a large number of collinear variables to a few summary variables that contain "most of the information".
2. Can be used as a "transformation and variable selection" method for the X's in linear regression, logistic regression, etc.
3. Can be used as a way of "learning the structure" of the X's even if we won't use the results in a regression.
4. If we have several possible response variables (y's), can be used to "extract" a best summary response y^* from them.

We will illustrate 1, 2, 3 below…

- *Goal:*
  - Find the linear combination of X's with *the biggest variance*. This is the *First Principal Component (sometimes called "first factor" or F_1).*
  - Find the linear combination of X's *uncorrelated with* F_1 that has the *next biggest variance*. This is the *Second Principal Component (sometimes called "second factor" or F_2).*
  - Etc. etc.
- *Notes:*
  - The usual rule is not to work with the raw X's but rather to work with standardized variables: $Z_1 = (X_1 - \overline{X}_1)/sd_1$, $Z_2 = (X_2 - \overline{X}_2)/sd_2$, …
  - We don't want to learn about the means of the X's, only the variablility of the X's, so centering by $\overline{X}_k$ makes sense.
  - If we do not divide each X_k by sd_k, then F_1 will mostly be the original X with the largest variance, F_2, will mostly be the original X with the next largest variance, etc. These are usually harder to interpret, so we divide each X_k by its sd_k as wll.
- *The "Model" [really, a transformation]*: If X has K columns, we obtain $J \leq K$ principal components as follows:

$$Z_{ik} = \ell_{k1}F_{i1} + \ell_{k2}F_{i2} + \ldots + \ell_{kJ}F_{iJ}, \quad k = 1, \ldots, K$$

or in matrix notation

$$Z = FL^T$$

where $F = [F_{ij}]$ is the matrix of "component scores" and $L = [\ell_{kj}]$ is the matrix of "loadings". We can recover the original X's as follows:

$$X_{ik} = (\ell_{k1}F_{i1} + \ell_{k2}F_{i2} + \ldots + \ell_{kJ}F_{iJ}) \cdot sd_k + \overline{X}_k$$

- If $J = K$, the matrices X (raw data) and F (principal components) are just functions of each other.
- If $J < K$, the matrix F (principal components) contains less information than X (hopefully, the "essential" information in X).
- Often the principal components have "nice" interpretations, that can help us understand the structure of the data.
- In a regression problem, we would try to work the with principal components (columns of the F matrix), F_1, F_2, F_3, etc., instead of the original X's.

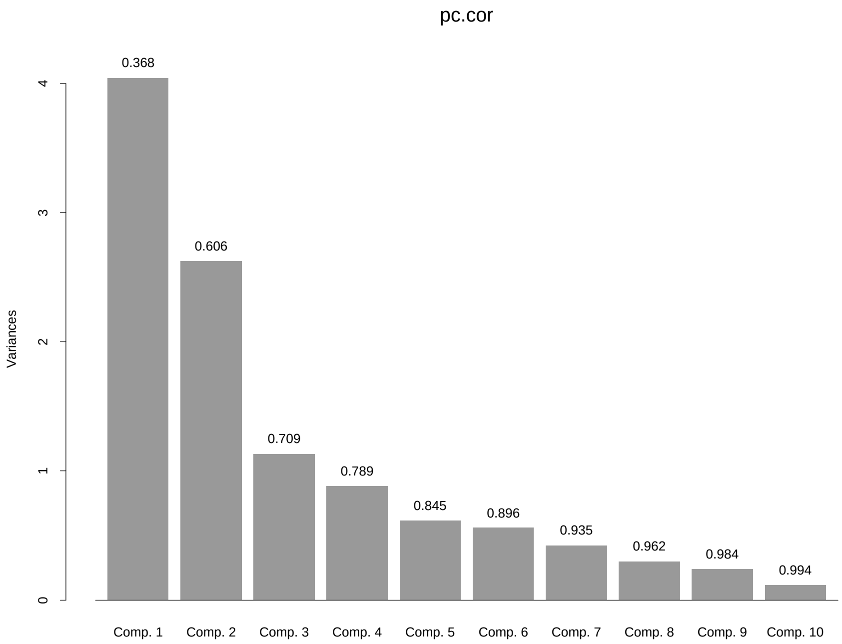
Example: Back to the highway data...

In doing a principal components analysis there are two important things to look at "first":

- A "scree plot", that shows how much variance in the X's is accounted for by each principal component (analogous to R^2 in regression).
- A listing of the loadings matrix, to see which X's are related to which F's. When printing the factor loadings it is useful to "blank out" the small ones so that we can see relationships more clearly.

First we consider the scree plot:

```
> high _ read.table("highway.dat",header=T)
> X _ high[,-1]
> y _ high[,1]

> pc.cor _ princomp(X,cor=T) # cor=T uses Z's.  It is possible to
>                            # use X's (cor=F) but we will not...
> screeplot(pc.cor)
```



We interpret the scree plot by looking for either

- An "elbow" that suggests that more principal components don't help much. This suggests we should keep F_1 (component 1) and F_2 (component 2), and probably F_3 (component 3) which is right at the elbow; or
- A point at which most (say, 90%) of the variance in the original data has been explained. This suggests we should keep F_1, F_2, F_3 and F_4.

Now we look at the loadings to see if there are nice relationships between the X's and the F's. This often helps us interpret the F's.

```
> print(loadings(pc.cor),cutoff=.1) # blanks all loadings below +/- 0.1
     Comp. 1 Comp. 2 Comp. 3 Comp. 4 Comp. 5 Comp. 6 Comp. 7 Comp. 8 Comp. 9
 Len  0.114   0.417  -0.372   0.181   0.497           0.562  -0.172   0.198
 Adt -0.446  -0.141  -0.170   0.223  -0.125                   0.108
 Trk          0.447  -0.191   0.414  -0.111  -0.553  -0.460   0.166  -0.120
  Sp -0.275   0.386   0.249  -0.310   0.140  -0.136   0.127   0.535
  lw         -0.134   0.769   0.421   0.254  -0.240   0.198
 shl -0.358   0.147          -0.606          -0.288          -0.172   0.102
 itg -0.415  -0.117  -0.169   0.286  -0.283   0.133   0.166   0.310   0.440
sigs         -0.466  -0.210           0.658  -0.154  -0.370   0.270   0.251
acpt  0.188  -0.405  -0.167          -0.277  -0.685   0.407
 lan -0.417  -0.150  -0.196           0.209           0.165          -0.790
```

```
type  0.441                                  0.132   0.226   0.666  -0.217

     Comp. 10 Comp. 11
 Len
 Adt  0.266    0.767
 Trk  0.137
  Sp -0.493    0.168
  lw  0.198
 shl  0.578   -0.152
 itg          -0.537
sigs
acpt -0.213
 lan          -0.248
type  0.483

> print(loadings(pc.cor),cutoff=.2) # blanks all loadings below +/- 0.2
     Comp. 1 Comp. 2 Comp. 3 Comp. 4 Comp. 5 Comp. 6 Comp. 7 Comp. 8 Comp. 9
 Len          0.417  -0.372           0.497           0.562
 Adt -0.446                   0.223
 Trk          0.447           0.414          -0.553  -0.460
  Sp -0.275   0.386   0.249  -0.310                           0.535
  lw                  0.769   0.421   0.254  -0.240
 shl -0.358                  -0.606          -0.288
 itg -0.415                   0.286  -0.283                   0.310   0.440
sigs         -0.466  -0.210           0.658          -0.370   0.270   0.251
acpt         -0.405                  -0.277  -0.685   0.407
 lan -0.417                           0.209                          -0.790
type  0.441                                           0.226   0.666  -0.217

     Comp. 10 Comp. 11
 Len
 Adt  0.266    0.767
 Trk
  Sp -0.493
  lw
 shl  0.578
 itg          -0.537
sigs
acpt -0.213
 lan          -0.248
type  0.483
```

```
> print(loadings(pc.cor),cutoff=.3) # # blanks all loadings below +/- 0.3
     Comp. 1 Comp. 2 Comp. 3 Comp. 4 Comp. 5 Comp. 6 Comp. 7 Comp. 8 Comp. 9
 Len          0.417  -0.372          0.497           0.562
 Adt -0.446
 Trk          0.447           0.414          -0.553  -0.460
  Sp          0.386          -0.310                          0.535
  lw                  0.769   0.421
 shl -0.358                  -0.606
 itg -0.415                                                  0.310   0.440
sigs         -0.466                  0.658          -0.370
acpt         -0.405                          -0.685   0.407
 lan -0.417                                                         -0.790
type  0.441                                                  0.666

     Comp. 10 Comp. 11
 Len
 Adt           0.767
 Trk
  Sp -0.493
  lw
 shl  0.578
 itg          -0.537
sigs
acpt
 lan
type  0.483
```

From this we can see that

- Component 1 is mostly related to:

  ```
  Adt  - average daily traffic count in thousands (estimated)
  shl  - width in feet of outer shoulder on the roadway
  itg  - number of freeway-type interchanges per mile in the segment
  lan  - total number of lanes of traffic in both directions
  type -  1 if Federal aid interstate highway,
          2 if principal arterial interstate highway,
          3 if minor arterial interstate highway,
          4 otherwise
  ```

  and seems to be related to whether the highway is a *major road*.

- Component 2 is mostly related to:

  ```
  Len  - length of segment in miles
  Trk  - truck volume as a percent of total volume
  Sp   - speed limit (in 1973, before the 55 mph limits)
  sigs - number of signalized interchanges per mile in the segment
  acpt - number of access points per mile in the segment
  ```

  and shares little in common with component 1! This component seems to be related to aspects of the highway that *slow down* traffic; it is also the same set of variables that BIC variable selection picked up.

- Component 3 is mostly related to:

  ```
  Len  - length of segment in miles
  lw   - lane width in feet
  ```

  Perhaps this component is related to whether this is an *open road* where visibility is good.

- Component 4 is mostly related to:

  ```
  Trk  - truck volume as a percent of total volume
  Sp   - speed limit (in 1973, before the 55 mph limits)
  lw   - lane width in feet
  shl  - width in feet of outer shoulder on the roadway
  ```

  This component may be related to *crowdedness* on the highway.

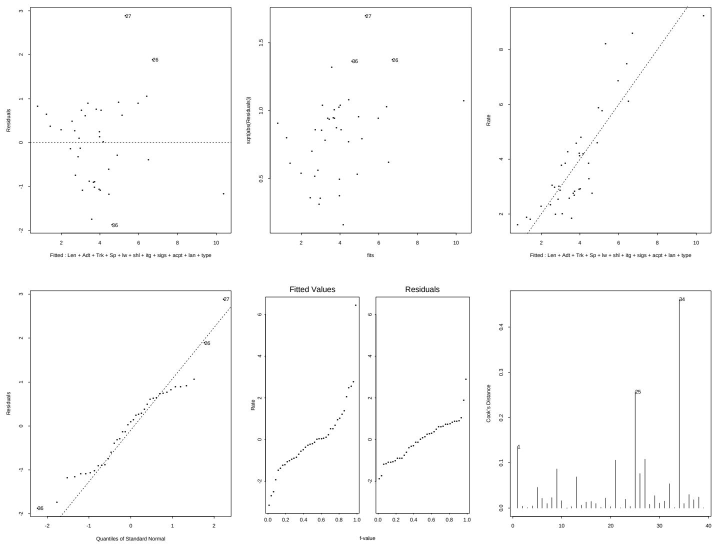
Let's give these components interpretable names, and re-do the regression analysis:

```
> dim(pc.cor$scores)  # this is where the factor scores F1, F2, ...  are kept
[1] 39 11
> major.road _   - pc.cor$scores[,1]
> slow.downs _   - pc.cor$scores[,2]
> open.road  _   - pc.cor$scores[,3]
> crowdedness _    pc.cor$scores[,4]

> attach(high)
> pclm _ lm(Rate ~ major.road + slow.downs + open.road + crowdedness)

> summary(pclm)

Call: lm(formula = Rate ~ major.road + slow.downs + open.road + crowdedness)
Residuals:
    Min      1Q Median     3Q   Max
 -1.826 -0.8812 0.2622 0.5912 3.236

Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept)   3.9333   0.1751    22.4637   0.0000
 major.road  -0.2711   0.0871    -3.1136   0.0037
 slow.downs   0.9628   0.1081     8.9078   0.0000
  open.road   0.2500   0.1645     1.5198   0.1378
crowdedness  -0.0058   0.1864    -0.0311   0.9754

Residual standard error: 1.093 on 34 degrees of freedom
Multiple R-Squared: 0.7288
F-statistic: 22.84 on 4 and 34 degrees of freedom, the p-value is 3.116e-09

Correlation of Coefficients:
            (Intercept) major.road slow.downs open.road
 major.road  0
 slow.downs  0           0
  open.road  0           0          0
crowdedness  0           0          0          0
```

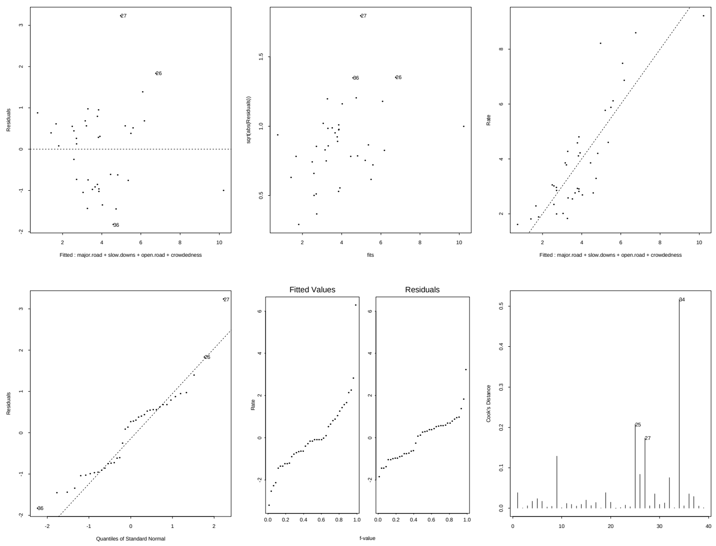
Note that

- The estimated β's have 0 correlation, because the components are themselves perfectly uncorrelated—no collinearity at all!

- It looks like only the first two components, major.road and slow.downs, were needed after all.
- We haven't lost much explanatory power from the original regression analysis! R^2 dropped only from 75% to 73diagnostic plots below suggest the same sorts of fit issues for both models.



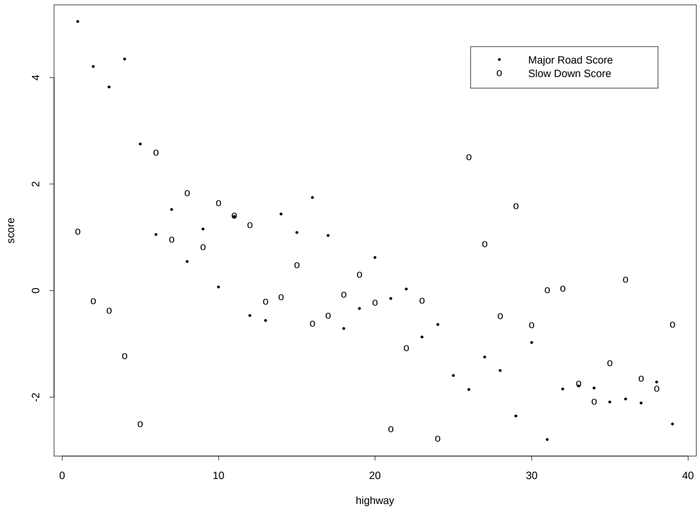
Diagnostic Plots from Raw Regression Analysis



Diagnostic Plots from Regression on Four Principal Components

Finally, we might want to rank highways on our principal component scores, since they seem to describe a lot of what might make highways accident-prone.

```
> plot(major.road,xlab="highway",ylab="score",pch="*")
> points(slow.downs,pch="o")
> legend(locator(),pch="*o",legend=c("Major Road Score", "Slow Down Score"))
```



Another tool for displaying these components is the *biplot*:

- Each observation is plotted at (x, y) = (major.road,slow.downs), labelled by its observation number;

- The *projection* of each original variable onto the (major.road,slow.downs) is plotted on the same plot, labelled by its name.