Admistrivia

- proposals due tomorrow and homework on friday

Story: Regression to the mean

- Stephen Stiegler (you may of heard of his father)
- “The history of statistics in 1933.”
- Reviews book by Secrist
  - Most winners lose in second round
  - Good companies go bad
  - Everything is falling apart
  - We should do something about it!
– Moral decay!

- Steve argues that this is a water shed year for statistics since this book was eveserated in two journals that are alive and kicking today.
- Idea called “regression to the mean.” Known to Galton
- Read Steve’s paper

**Three distances**

- regression: up and down
- reverse regression: left and right
- errors in variables: closest distance (called SD line)
- conversion factor:
  - slope = correlation * ratio of standard deviations
  - So to turn a regression around, you keep the correlation the same, just change the ratio.

**Where PCA got started: IQ**

- Spearman (research done during 1904-1920)
  - rank correlation
– psychology, “G” in IQ
– came up with factor analysis: basically the same idea as PCA. We will use it for discussion of ideas.

IQ: background

• Spearman found grades in various classes correlated
• conjectured there was one thing that did it all
• called G

What are the class weights?

• classes: Math, English, history, PhEd
• weights: \(0.3 \text{ M} + 0.4 \text{ E} + 0.2 \text{ H} + 0 \text{ PhEd}\)
• tells how much each class is related to IQ
• FACT: reaction time is related, motor speed isn’t. Takes care to distinguish these two in tests.

What are the individual scores?

• Each student has \(g = 0.3 \text{ M} + 0.4 \text{ E} + 0.2 \text{ H} + 0 \text{ PhEd}\)
Putting them together

Instead of storing all grades, we can store two things:

- student scores: g
- class loading: (.3,.4,.2,0)
- Compute estimated grade in math by: g * .3
- Compute estimated grade in PhEd by: g * 0
- So we get nothing out of PhEd

Using it for PCA’s

- Suppose we want to find components $C$ for a dataset $X$
- If we found it, we could write each $X_{ij} \approx \beta_i C_j$.
- Looks like a regression!
- fit the betas:
  $$X_{ij} = \beta_i C_j + \sigma \epsilon_{ij}$$
  In R notation: $X \sim C$
- Where do the $C$’c come from? Fit it the $\beta$’s:
  $$X_{ij} = C_j \beta_i + \text{noise}$$
  In R notation: $X \sim \beta$
- Both easy! So use “fixed point idea.” Just iterate.
What do we get?

- We find a $C$ which is a linear combination of the $X$’s and it is close to all of them. This is exactly what we want.

- We can now take ALL these residuals and repeat the process. This gets a 2nd principle component.

- Continuing in this fashion we can get more and more.
Look at the .R file for details on how we read in the federalist papers. It is just like before. For those seeing the Rnw source—you can look at the commands below.

The components themselves

We used the exact same script that Kory used for the homework to read in the variables and name them. Now let’s make up our principle components.

```r
> few.variables <- c(11:85)
> variables <- c(11:200)
> pcs <- prcomp(federal[,variables])$rotation

> federal$pc1 <- as.matrix(federal[,variables]) %*% pcs[,1]
> colors <- c("red", "white", "black", "gray", "blue")[federal$author]
> plot(federal$pc1, pch=shapes , col=colors)
```

The pcs variable is a matrix. So we need to do some matrix tricks to generate the actual first principle compoent.

```r
> federal$pc1 <- as.matrix(federal[,variables]) %*% pcs[,1]
> colors <- c("red", "white", "black", "gray", "blue")[federal$author]
> plot(federal$pc1, pch=shapes , col=colors)
```

The 2nd PC (loadings):
> federal$pc2 <- as.matrix(federal[,variables]) %*% pcs[,2]
> federal$pc3 <- as.matrix(federal[,variables]) %*% pcs[,3]
> colors <- c("red","white","black","gray","blue")[federal$author]
> plot(federal$pc2, pch=shapes , col=colors)
>

The weights themselves (PC1):

> i <- (1:length(pcs[,1]))
> plot(pcs[,1],i,col="red",pch=16)
With word labels

```r
> n <- substr(rownames(pcs),6,100)
> plot(pcs[,1],i,pch=16,col="red",ylim=c(0,50))
> text(pcs[,1],i+1,n)
> `}
```
The weights themselves (PC2):

```r
> plot(pcs[,2],i,pch=16,col="red",ylim=c(0,50))
> text(pcs[,2],i+1,n)
> }
```
Predictions

Finally, let’s predict using the first principle component:

```r
> regr <- lm(federal$Y ~ federal$pc1)
> predictions <- predict(regr, newdata=federal)
> plot(predictions ~ federal$number, pch=16, col=colors)
> abline(.5,0)
> ```
Or using the first 3 principle components:

```r
> regr <- lm(federal$Y ~ federal$pc1 + federal$pc2 + federal$pc3)
> predictions <- predict(regr,newdata=federal)
> plot(predictions ~ federal$number , pch=16 , col=colors)
> abline(.5,0)
> 
```
With 9 PC's:

```r
> federal$pc4 <- as.matrix(federal[,variables]) %*% pcs[,4]
> federal$pc5 <- as.matrix(federal[,variables]) %*% pcs[,5]
> federal$pc6 <- as.matrix(federal[,variables]) %*% pcs[,6]
> federal$pc7 <- as.matrix(federal[,variables]) %*% pcs[,7]
> federal$pc8 <- as.matrix(federal[,variables]) %*% pcs[,8]
> federal$pc9 <- as.matrix(federal[,variables]) %*% pcs[,9]
> regr <- lm(federal$Y ~ federal$pc1 + federal$pc2 + federal$pc3 + federal$pc4 + federal$pc5 + federal$pc6 + federal$pc7 + federal$pc8 + federal$pc9)
```
> predictions <- predict(regr,newdata=federal)
> plot(predictions ~ federal$number , pch=16 , col=colors)
> abline(.5,0)
>
Why bother?

What if we used all the PCs? We get our usual complete disaster of over fitting.