Stats216: Session 2

Linear Regression Analysis of NCAA Basketball Data

In this in-class session, we will analyze a data set containing the outcomes of every game in the 2012-2013 regular season, and the postseason NCAA tournament. There are 5541 games and 347 teams. Our goals will be to:

- Estimate the quality of each team, to obtain objective rankings
- Explain the outcomes in the regular season and test hypotheses that may interest us
- Predict the winner and margin of victory in future games

1. Loading in the data

First, read in the files `games.csv` and `teams.csv` into the data frames `games` and `teams`. We will load them "as is" (i.e. strings not converted to factors).

```
games <- read.csv("http://statweb.stanford.edu/~jgorham/games.csv",as.is=TRUE)
teams <- read.csv("http://statweb.stanford.edu/~jgorham/teams.csv",as.is=TRUE)
```

The `games` data has an entry for each game played, and the `teams` data has an entry for each Division 1 team (there are a few non-D1 teams represented in the `games` data).

```
head(games)[, !(names(games) %in% c('neutralLocation', 'gameType'))]
```

```
##         date                          home                  away
## 1 2012-11-09            albany-great-danes        duquesne-dukes
## 2 2012-11-11           ohio-state-buckeyes    albany-great-danes
## 3 2012-11-13            washington-huskies    albany-great-danes
## 4 2012-11-17            albany-great-danes        umkc-kangaroos
## 5 2012-11-18            albany-great-danes loyola-(md)-greyhounds
## 6 2012-11-20 south-carolina-state-bulldogs    albany-great-danes
##   homeScore awayScore
## 1        69        66
## 2        82        60
## 3        62        63
## 4        62        59
## 5        64        67
## 6        55        83
```

Two columns have been omitted above: the column `neutralLocation` is 1 if the game was not a real home game for the nominal home team, and 0 otherwise. `gameType` is `REG` for regular-season games, `NCAA` for NCAA tournament games, and `POST` for other postseason games (not the NCAA tournament).

The `teams` data has an entry for each team coding its name, conference, whether it made the NCAA tournament, and its AP and USA Today ranks at the end of the regular season (1-25 or `NA` if unranked).

```
head(teams)
```

```
##                          team   conference inTourney apRank usaTodayRank
## 1      stony-brook-seawolves america-east         0     NA           NA
## 2         vermont-catamounts america-east         0     NA           NA
## 3 boston-university-terriers america-east         0     NA           NA
## 4             hartford-hawks america-east         0     NA           NA
## 5         albany-great-danes america-east         1     NA           NA
## 6          maine-black-bears america-east         0     NA           NA
```

Finally, we make one vector containing all the team names, because the three columns do not perfectly overlap:

```
all.teams <- sort(unique(c(teams$team,games$home,games$away)))
```

2. How to Rank the Teams?

Now, spend a few minutes to come up with a way to rank the teams, based on all of the regular-season games. Try your method, or a couple of different methods, and compare the ranks you get to the official end-season rankings. Which rankings do you find more credible?

[*Bonus Question*] Think about the following bonus question if you have extra time. I will not be going over the answers to bonus questions in class, but you can ask the circulating staff.

- Does your method account for a team's strength of schedule (the quality of a teams' opponents)? If not, how might you modify it so that it does?

Answers:

One simple way to rank teams is by comparing their average score to their opponents' average score.

```
## Function to compute a team's total margin of victory
total.margin <- function(team) {
  with(games,
       sum(homeScore[home==team])
       + sum(awayScore[away==team])
       - sum(homeScore[away==team])
       - sum(awayScore[home==team]))
  }
## Function to compute the number of games a team played in
number.games <- function(team) {
  with(games, sum(home==team) + sum(away==team))
}
## Compute total margin and number of games for each team
margins <- sapply(teams$team, total.margin)
number.games <- sapply(teams$team, number.games)
## check: make sure names line up
mean(names(margins) == names(number.games))
```

```
## [1] 1
```

```
## compute average
margin.per.game <- margins / number.games
```

```r
rank.table <- cbind("Margin (Avg)" = margin.per.game,
                    "Margin Rank"  = rank(-margin.per.game,ties="min"),
                    "AP Rank"      = teams$apRank,
                    "USAT Rank"    = teams$usaTodayRank)

margin.top25 <- order(margin.per.game,decreasing=TRUE)[1:25]
rank.table[margin.top25,]
```

```
##                                Margin (Avg) Margin Rank AP Rank USAT Rank
## florida-gators                    17.027027           1      14        12
## indiana-hoosiers                  16.527778           2       4         5
## gonzaga-bulldogs                  16.294118           3       1         1
## louisville-cardinals              15.675000           4       2         2
## kansas-jayhawks                   13.243243           5       3         3
## pittsburgh-panthers               13.181818           6      20        22
## weber-state-wildcats              12.972222           7      NA        NA
## middle-tennessee-blue-raiders     12.676471           8      NA        NA
## virginia-commonwealth-rams        12.416667           9      NA        23
## michigan-wolverines               11.923077          10      10        11
## duke-blue-devils                  11.888889          11       6         7
## syracuse-orange                   11.725000          12      16        18
## stephen-f.-austin-lumberjacks     11.448276          13      NA        NA
## belmont-bruins                    11.322581          14      NA        NA
## creighton-bluejays                11.277778          15      22        21
## saint-mary's-gaels                11.205882          16      NA        25
## davidson-wildcats                 11.176471          17      NA        NA
## ohio-state-buckeyes               10.864865          18       7         6
## ole-miss-rebels                   10.277778          19      NA        NA
## saint-louis-billikens             10.028571          20      13        13
## arizona-wildcats                   9.828571          21      21        20
## stony-brook-seawolves              9.718750          22      NA        NA
## memphis-tigers                     9.555556          23      19        15
## missouri-tigers                    9.411765          24      NA        NA
## north-dakota-state-bison           9.218750          25      NA        NA
```

Some of the highly-ranked teams look about right (Louisville, Indiana, Gonzaga), but some, like Weber State, who went 26-6 in the Big Sky conference and are ranked 7th, appear to be rather overrated by this metric.

Looking at the page for Weber State's 2012-2013 season, it appears that they benefited from an overall easy schedule with no ranked opponents, including one 65-point victory against the Southwest Minnesota State Mustang.

In the next section, we will see how to modify this method to account for strength of schedule.

3. A Linear Regression Model for Ranking Teams

Statistical modeling is a powerful tool for learning from data. Our meta-strategy is to define some statistical model whose parameters correspond to whatever quantities we want to estimate.

Our response variable for today is the margin of victory (or defeat) for the home team in a particular game. That is, define

$$y_i = (\text{home score} - \text{away score}) \text{ in game } i \tag{1}$$

Now, we want to define a linear regression model that *explains* the response, y_i, in terms of both teams' merits. The simplest such model will look something like

$$y_i = \text{quality of home}(i) - \text{quality of away}(i) + \text{noise} \tag{2}$$

where home(i) and away(i) are the home and away teams for game i.

To formulate this model as a linear regression in standard form, we need to come up with a definition for the predictors x and coefficients β so that estimating β amounts to estimating the quality of each team. That is, we want a definition for x_{ij} and β_j for which

$$y_i = \sum_j x_{ij}\beta_j + \varepsilon_i \tag{3}$$

Now, with your group, try to formulate our model as a linear regression. How many predictor variables are there? How many coefficients? How is x_{ij} defined?

[*Bonus Questions*] If you have time, consider the following questions:

- In our new model, suppose that a small high school plays only one game, against Weber State, and loses by a very wide margin. Can you prove that this game will have *no effect* whatsoever on the fitted values of β?
- (*Challenging*) Our model now controls for the quality of a team's opponents. As a result, I claim that scheduling many difficult opponents (or many easy opponents) will neither hurt nor help a team's ranking. Can you back up this claim in a precise mathematical way?

Answers:

Define one predictor variable for each team, which is a sort of "signed dummy variable." For game i and team j, let

$$x_{ij} = \begin{cases} +1 & j \text{ is home}(i) \\ -1 & j \text{ is away}(i) \\ 0 & j \text{ didn't play} \end{cases} \tag{4}$$

For example, if game i consists of team 1 visiting team 2, then $x_i = (-1, 1, 0, 0, \ldots, 0)$.

Now we can check that

$$\sum_j x_{ij}\beta_j = \beta_{\text{home}(i)} - \beta_{\text{away}(i)} \tag{5}$$

so the coefficient β_j corresponds exactly to the quality of team j in our model.

We can generate y and X as follows:

```r
y <- with(games, homeScore-awayScore)

## Construct a data frame of the right dimensions, with all zeros
X0 <- as.data.frame(matrix(0,nrow(games),length(all.teams)))
names(X0) <- all.teams

## Fill in the columns, one by one
for(tm in all.teams) {
  X0[[tm]] <- 1*(games$home==tm) - 1*(games$away==tm)
}
```

Now, we are in good shape, because we can use all our tools from linear regression to estimate β, carry out hypothesis tests, construct confidence intervals, etc.

4. An Identifiability Problem

When we fit our model, we will ask R to find the best-fitting β vector. There is a small problem, however: for any candidate value of β, there are infinitely many other values $\tilde{\beta}$ that make **exactly** the same predictions. So the "best β" is not uniquely defined.

For any constant c, suppose that I redefine $\tilde{\beta}_j = \beta_j + c$. Then for every game i,

$$\tilde{\beta}_{\text{home}(i)} - \tilde{\beta}_{\text{away}(i)} = \beta_{\text{home}(i)} - \beta_{\text{away}(i)} \tag{6}$$

so the distribution of y is identical for parameters $\tilde{\beta}$ and β, no matter what c is. We can never distinguish these two models from each other, because the models make identical predictions no matter what. In statistical lingo, this is called an *identifiability* problem. It very often arises with dummy variables.

To fix it, we need to add some linear constraint on β to resolve the ambiguity. For example $\sum_j \beta_j = 0$, or we can pick some "special" baseline team j and require that $\beta_j = 0$. We will use the latter strategy, and we will take Stanford's team as the baseline. Now, with your team, figure out how to modify the X matrix to implement this.

(Actually, `lm` is smart enough to fix this automatically for you by arbitrarily picking one team to be the baseline. But let's not blindly rely on that)

[*Bonus Questions*] If you have time, consider the following questions:

- Suppose that we had chosen a different team as our baseline.
  - How would the estimates be different?
  - Would we obtain identical rankings?
  - Would we obtain identical standard errors?
- Under what circumstances would we still have an identifiability problem even after constraining $\beta_{\text{Stanford}} = 0$?

Answers:

We can effectively force $\beta_j = 0$ for the j corresponding to Stanford by eliminating that column from the predictor matrix.

```r
X <- X0[,names(X0) != "stanford-cardinal"]
reg.season.games <- which(games$gameType=="REG")
```

Now, let's fit our model. There is no intercept in the model, so we explicitly exclude it from the formula.

```r
mod <- lm(y ~ 0 + ., data=X, subset=reg.season.games)
head(coef(summary(mod)))
```

```
##                           Estimate Std. Error     t value     Pr(>|t|)
## `air-force-falcons`      -6.687934   2.897768  -2.3079603 2.104205e-02
## `akron-zips`             -2.557678   2.841221  -0.9002039 3.680552e-01
## `alabama-a&m-bulldogs`  -30.590013   2.908702 -10.5167213 1.338001e-25
## `alabama-crimson-tide`   -2.851010   2.802443  -1.0173304 3.090456e-01
## `alabama-state-hornets` -29.958427   2.849296 -10.5143267 1.371671e-25
## `albany-great-danes`    -13.334555   2.818343  -4.7313458 2.291935e-06
```

```
summary(mod)$r.squared
```

```
## [1] 0.551235
```

It looks like our model explains about half of the variability in basketball scores.

5. Interpreting the Model

Next, let's try to interpret the model that we just fit. With your group, take a few moments to answer the following questions:

5.1. Based on this model, what would be a reasonable point spread if Alabama (`alabama-crimson-tide`) played against Air Force?

5.2. Can we be confident that Stanford is better than Alabama? Better than Air Force?

5.3. How can we test whether Alabama is better than Air Force?

5.4. Does the dataset and model support the notion of home field advantage? How many points per game is it? Is it statistically significant?

Answers:

5.1. If Alabama were the home team, then the expected score difference is

$$\hat{\beta}_{\text{Alabama}} - \hat{\beta}_{\text{Air Force}} \tag{7}$$

So we can answer the question by comparing their coefficients:

```
coef(mod)["`alabama-crimson-tide`"] - coef(mod)["`air-force-falcons`"]
```

```
## `alabama-crimson-tide`
##               3.836924
```

so a fair point spread would be about 3.84 in favor of Alabama.

5.2. The hypothesis that Stanford is better than Alabama is equivalent to the hypothesis that $\beta_{\text{Alabama}} < 0$. If we use a two-sided test, our t statistic needs to be bigger than 2 in absolute value. It is not for Alabama, but it is for Air Force; hence the answers are "no" and "yes."

5.3. We cannot answer this question based only on the coefficient table printed above. To test the null hypothesis that

$$\theta = \beta_{\text{Alabama}} - \beta_{\text{Air Force}} = 0 \tag{8}$$

we would need to know the standard error of

$$\hat{\theta} = \hat{\beta}_{\text{Alabama}} - \hat{\beta}_{\text{Air Force}} \tag{9}$$

The easiest way to find this out would be to simply re-fit the model with Alabama (or Air Force) as the baseline.

5.4 We can assess home field advantage by including another column, which is 1 if and only if that game was played in a non-neutral location:

```
homeAdv <- 1 - games$neutralLocation
Xh <- cbind(homeAdv=homeAdv, X)
homeAdv.mod <- lm(y ~ 0 + ., data=Xh, subset=reg.season.games)
head(coef(summary(homeAdv.mod)), 1)
```

```
##         Estimate Std. Error  t value      Pr(>|t|)
## homeAdv 3.528499  0.1568782 22.49197 8.654579e-107
```

It looks like home field advantage is worth about 3.5 points per match and is clearly significant.

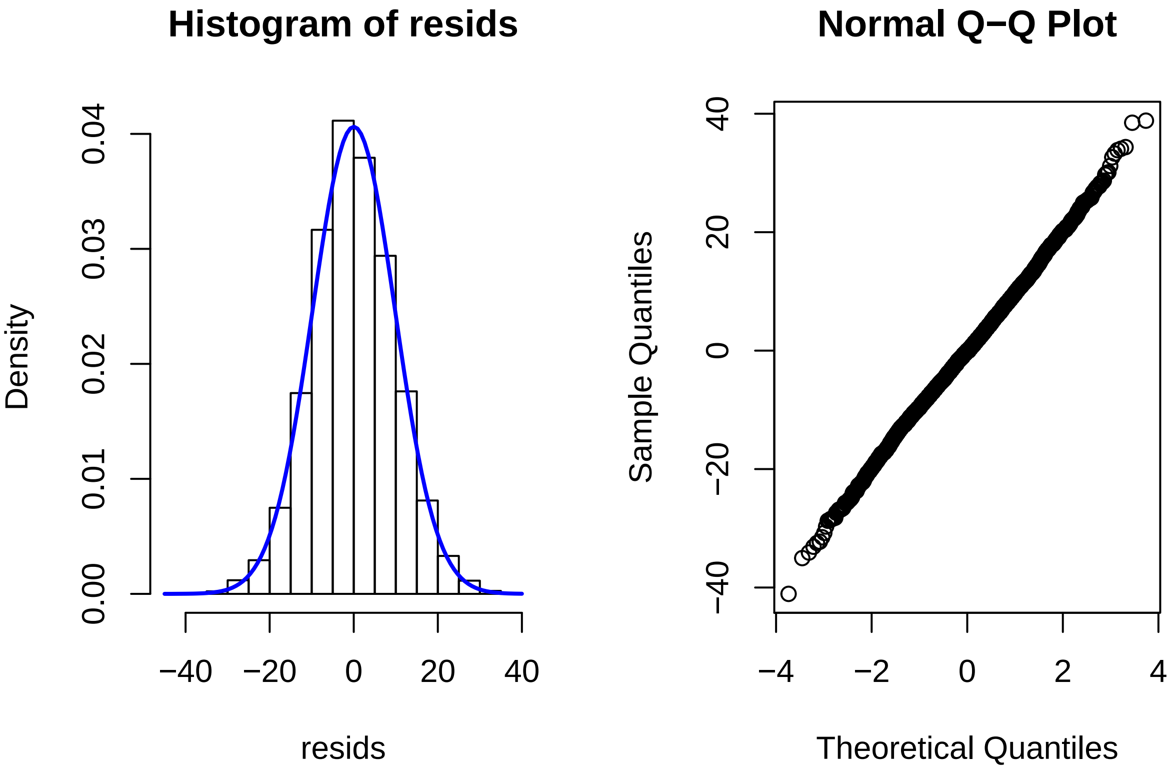
6. Predicting Wins and Losses

If we wanted to run a bookie business, we would not only have to set point spreads, but also set odds in advance of each game. In this section we will see that our model can give us odds as well as point spreads.

We made an assumption that the errors were normal, so we should also check whether the residuals are normal. Let's check out the residuals from the last model (with home-court advantage) and see if they look reasonably normal.

```
homeAdv.coef <- coef(homeAdv.mod)[paste("`",teams$team,"`",sep="")]
names(homeAdv.coef) <- teams$team

resids <- homeAdv.mod$resid
par(mfrow=c(1,2))
hist(resids,freq=FALSE)
curve(dnorm(x,mean(resids),sd(resids)),add=TRUE,col="blue",lwd=2)
qqnorm(resids)
```



Since our normal-errors assumption passes this basic sanity check, let's take our model seriously. Assuming the errors are truly normal, how can we use our model to predict the win/loss outcome of a particular game?

[*Bonus Questions*] If you have time, consider the following questions:

- Wichita State (`wichita-state-schockers` to us) made an improbable run in the NCAA tournament, beating a #1 seed (Gonzaga) and a #2 seed (Ohio State), and eventually losing to another #1 seed (Louisville) in the Final Four. According to the model we just fit, just *how* improbable was Wichita State's run? That is, what was the probability they would have beaten the first four opponents that they faced?
- The actual tournament winner was `louisville-cardinals`. Given the six opponents they faced, what was the probability they would win the tournament?

You can use this function to get data in a slightly more manageable form:

```
schedule <- function(team, game.type) {
  home.sch <- with(games, games[home==team & gameType==game.type,c(1,3,4,5)])
  away.sch <- with(games, games[away==team & gameType==game.type,c(1,2,5,4)])
  names(home.sch) <- names(away.sch) <- c("date","opponent","score","oppoScore")
  sch <- rbind(home.sch,away.sch)

  sch$margin <- with(sch, score-oppoScore)
  sch$oppoQuality <- homeAdv.coef[as.character(sch$opponent)]

  sch <- sch[order(sch$date),]
  rownames(sch) <- NULL
  return(sch)
}
schedule("wichita-state-shockers","NCAA")
```

```
##         date             opponent score oppoScore margin oppoQuality
## 1 2013-03-21  pittsburgh-panthers    73        55     18   7.3562577
## 2 2013-03-23     gonzaga-bulldogs    76        70      6   9.9875221
## 3 2013-03-28    la-salle-explorers    72        58     14  -0.7725049
## 4 2013-03-30  ohio-state-buckeyes    70        66      4   7.8965499
## 5 2013-04-06 louisville-cardinals    68        72     -4  12.0810415
```

Answers:

Our model says that $y_i \sim N(\mu_i, \sigma^2)$, where σ^2 is the noise variance and μ_i is the home-away quality difference (possibly adjusted for home-court advantage).

To calculate the probability the home team loses ($y_i < 0$), we can exploit the fact that $(y_i - \mu_i)/\sigma \sim N(0, 1)$. If Φ denotes the cumulative distribution function of a $N(0, 1)$ variable, then:

$$\begin{aligned} P(y_i < 0) &= P\left(\frac{y_i - \mu_i}{\sigma} < -\mu_i/\sigma\right) && (10) \\ &= \Phi(-\mu_i/\sigma) && (11) \end{aligned}$$

We can estimate σ by calculating the residual variance (technically we should make an adjustment for degrees of freedom but let's ignore that).

```
(sigma <- sd(resids))
```

```
## [1] 9.821427
```

So, for example, we can compute the probability that Stanford would beat Berkeley (a slightly worse team, according to our model), at Berkeley:

```
mu <- coef(homeAdv.mod)["homeAdv"] + coef(homeAdv.mod)["`california-golden-bears`"] - 0
pnorm(-mu/sigma)
```

```
##   homeAdv
## 0.4204275
```

According to our model, that game would be roughly a toss-up.

Next, let's find out how improbable Wichita State's run was. Note that all the tournament games were played at neutral locations.

```
wst.schedule <- schedule("wichita-state-shockers","NCAA")
mu <- wst.schedule$oppoQuality - homeAdv.coef["wichita-state-shockers"]
names(mu) <- wst.schedule$opponent
(p.wst.win <- pnorm(-mu/sigma))
```

```
##  pittsburgh-panthers     gonzaga-bulldogs   la-salle-explorers
##            0.2915299            0.2070124            0.6097760
##  ohio-state-buckeyes louisville-cardinals
##            0.2729443            0.1515074
```

So the probability that Wichita State would have won its first four games (conditional on the opponents it faced) is about 1%.

```
prod(p.wst.win[1:4])
```

```
## [1] 0.0100444
```

By contrast, the probability of Louisville winning the whole tournament (conditional on their opponents) was about 25%:

```
lou.schedule <- schedule("louisville-cardinals","NCAA")
mu <- lou.schedule$oppoQuality - homeAdv.coef["louisville-cardinals"]
prod(pnorm(-mu/sigma))
```

```
## [1] 0.2509396
```

7. Model Rankings vs. Official Rankings

Finally, let's inspect the rankings given by our model.

```
rank.table <- cbind("Model Score" = homeAdv.coef,
                    "Model Rank"  = rank(-homeAdv.coef,ties="min"),
                    "AP Rank"     = teams$apRank,
                    "USAT Rank"   = teams$usaTodayRank)
rank.table[order(homeAdv.coef,decreasing=TRUE)[1:25],]
```

```
##                            Model Score Model Rank AP Rank USAT Rank
## indiana-hoosiers             13.255228          1       4         5
## florida-gators               12.638550          2      14        12
## louisville-cardinals         12.081041          3       2         2
## gonzaga-bulldogs              9.987522          4       1         1
## duke-blue-devils              9.393678          5       6         7
## kansas-jayhawks               8.631876          6       3         3
## ohio-state-buckeyes           7.896550          7       7         6
## michigan-wolverines           7.505373          8      10        11
## pittsburgh-panthers           7.356258          9      20        22
## syracuse-orange               7.069567         10      16        18
## wisconsin-badgers             6.831338         11      18        17
## michigan-state-spartans       6.176226         12       9         9
## creighton-bluejays            5.283583         13      22        21
## miami-(fl)-hurricanes         5.191585         14       5         4
## virginia-commonwealth-rams    4.957174         15      NA        23
## arizona-wildcats              4.684883         16      21        20
## minnesota-golden-gophers      4.421583         17      NA        NA
## georgetown-hoyas              4.258965         18       8         8
## missouri-tigers               3.962353         19      NA        NA
## oklahoma-state-cowboys        3.690893         20      17        19
## saint-mary's-gaels            3.372377         21      NA        25
## colorado-state-rams           3.152205         22      NA        NA
## new-mexico-lobos              3.004408         23      11        10
## north-carolina-tar-heels      2.715207         24      NA        NA
## ole-miss-rebels               2.644823         25      NA        NA
```

Our rankings still differ significantly from the official rankings on several teams. To take one particularly glaring example, our model is highly confident that `florida-gators` is an elite team, despite its relatively low ranking by the press. By contrast, our model doesn't think too much of `miami-(fl)-hurricanes` despite the opinion of the press that the 'Canes were elite.

See if you can figure out why our model might beg to differ with the press.

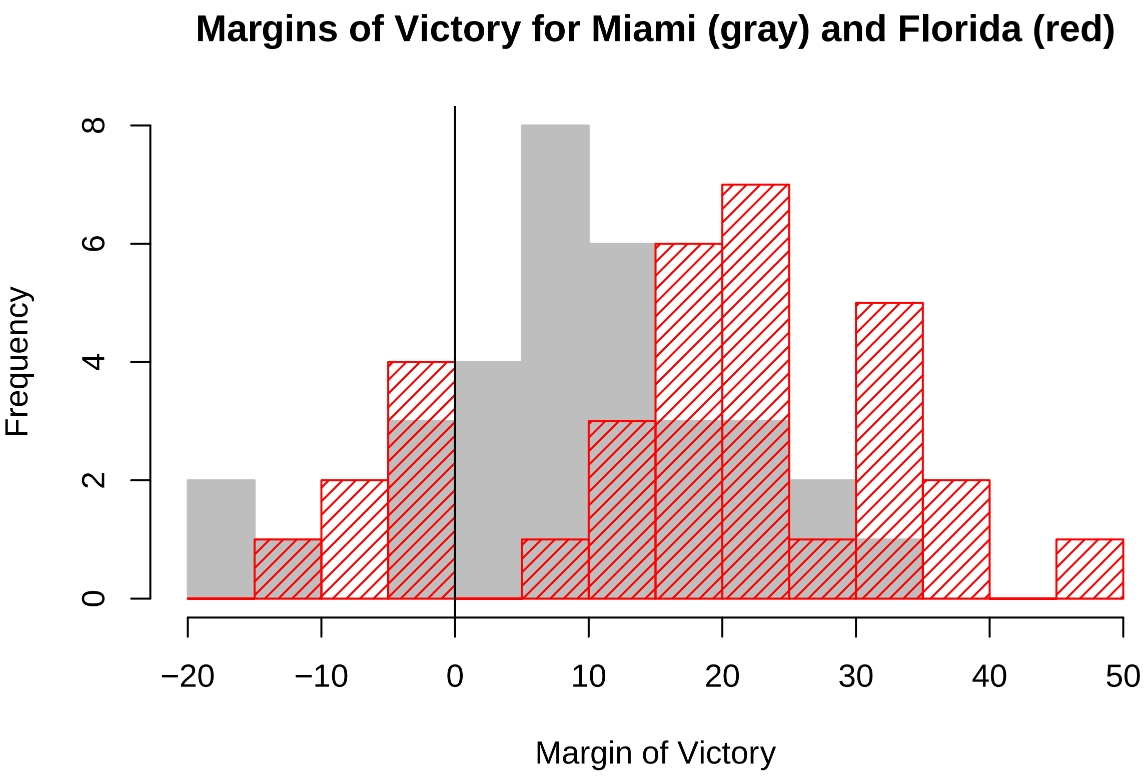
Answers:

From the schedules above, we see that Florida's 7 losses are all close, with only one loss by more than 6 points, whereas all of their 26 wins are by at least 10 points.

Miami has a better overall win-loss record (27-6) against a somewhat stronger group of opponents, and finished the season strong by winning the ACC tournament. However, 10 of their wins were by less than 10 points, and they lost three games by wide margins.

Below, we plot each team's margin of victory distribution for its 33 regular season games.

```r
hist(schedule("miami-(fl)-hurricanes", "REG")$margin,
     col="gray",border="gray",breaks=seq(-20,50,5),
     xlab="Margin of Victory",
     main="Margins of Victory for Miami (gray) and Florida (red)",)
hist(schedule("florida-gators", "REG")$margin,add=TRUE,
     border="red",breaks=seq(-20,50,5),col="red",density=20)
abline(v=0)
```



Miami did have slightly better opponents, but not by much

```
mean(schedule("miami-(fl)-hurricanes", "REG")$oppoQuality)
```

```
## [1] -3.836991
```

```
mean(schedule("florida-gators", "REG")$oppoQuality)
```

```
## [1] -4.873147
```

Notice that in our model, going from a -1 margin of victory to a +1 margin has the same impact on a team's rating is going from +6 to +8, or from +40 to +42. So, we might be giving Florida a little too much credit for running up the score against its opponents.

Conversely, sports journalists typically care a lot about whether a team wins or loses, because they believe that scoring a buzzer-beater to win a game proves that a team possesses the "heart of a champion." This elusive quality is thought to outweigh the more prosaic ability to score a lot of points.

8. The Value of Nerlens Noel

Nerlens Noel was a star center for the University of Kentucky (`kentucky-wildcats` in our data set). He was projected to be the #1 overall pick in the 2013 NBA draft before he tore his ACL in a February 12 game (and was eventually the #6 overall pick despite his injury).

Statistical analysts employed by sports teams are especially interested in evaluating individual players. There are various methods, usually requiring granular minute-by-minute data on scoring and substitutions. We do not have such granular data at our disposal, but for an injured player like Noel, we can estimate how much better Kentucky was with him than without him.

Re-fit your model with one additional predictor variable, to get an estimate and confidence interval for Noel's contribution to Kentucky (i.e., how much better Kentucky is with him than without him).

[*Bonus Questions*] If you have time, consider the following questions:

- Give an estimate and confidence interval for Kentucky's quality with Noel, and for its quality without Noel. Where would Kentucky be ranked if Noel had not been injured?
- Suppose that instead of the `games` data, we had a data point for every one-minute long period during the course of every game. During each minute, suppose we also had a record of which five players were playing for each team, as well as the number of points scored by each team during that minute (for simplicity, assume substitutions only happen in between the one-minute periods). Can you come up with a model, similar to the one we fit today, that we could use to estimate each *player's* quality?

Answers:

Let n_i be +1 if Noel played for the home team (i.e., all UK home games before February 12), -1 if he played for the away team (UK away games before Feb 12), and 0 if he did not play. Then we can simply fit the model:

$$y_i = \beta_{-1} n_i + \beta_0 h_i + \beta_{\text{home}(i)} - \beta_{\text{away}(i)} + \varepsilon_i \tag{12}$$

```
nerlens <- rep(0,nrow(games))
nerlens[games$home=="kentucky-wildcats" & games$date<"2013-02-12"] <- 1
nerlens[games$away=="kentucky-wildcats" & games$date<"2013-02-12"] <- -1
nerlens.mod <- lm(y ~ 0 + nerlens + ., data=Xh, subset=reg.season.games)
head(coef(summary(nerlens.mod)))
```

```
##                           Estimate Std. Error     t value      Pr(>|t|)
## nerlens                 11.4503609  4.1033060   2.7905208  5.282328e-03
## homeAdv                  3.5264822  0.1567733  22.4941548 8.294412e-107
## `air-force-falcons`     -5.3069737  2.7602474  -1.9226442  5.458163e-02
## `akron-zips`            -0.9369568  2.7066503  -0.3461684  7.292308e-01
## `alabama-a&m-bulldogs` -27.1891732  2.7742456  -9.8005644  1.784244e-22
## `alabama-crimson-tide`  -2.6585470  2.6688528  -0.9961385  3.192312e-01
```

```
nerlens.estimate <- coef(nerlens.mod)["nerlens"]
nerlens.se <- coef(summary(nerlens.mod))["nerlens",2]
(nerlens.CI <- c("CI lower"=nerlens.estimate - 2*nerlens.se,
                 "CI upper"=nerlens.estimate + 2*nerlens.se))
```

```
## CI lower.nerlens CI upper.nerlens
##         3.243749        19.656973
```

So Noel was worth about 11 points to Kentucky, but we cannot place too much confidence in this figure.

```
coef(nerlens.mod)[c("nerlens","`kentucky-wildcats`")]
```

```
##             nerlens `kentucky-wildcats`
##           11.450361           -5.988489
```

If we have minute-level data, then we can assign a coefficient β_j to each *player*. Then, letting y_i be the home-away scoring margin for minute i, we can set

$$X_{ij} = \begin{cases} +1 & j \text{ played for the home team in minute } i \\ -1 & j \text{ played for the away team in minute } i \\ 0 & j \text{ didn't play in minute } i \end{cases} \tag{13}$$

This is known as the *adjusted plus/minus* model, and it is one of the tools used by statisticians in the NBA to evaluate players.

9. A Non-Regression Based Ranking Approach

There are many other ways to rank teams; we'll see one more that can be cast as a random walk on a graph.

We setup the problem as follows. Place M counters initially on each team. We'll then complete 100 rounds of the following procedure:

1. Randomly permute all the teams
2. For each team X, grab all their counters and then for each counter:

- Select a random game that X played in and then flip a coin that comes up heads with probability p. If the coin is heads, put the counter on the winning team (otherwise move it to the losing team).

After many rounds have been completed, the teams with the most counters are deemed the highest ranked.

Let's see an example to make things clear. Suppose that we only had three teams–A, B and C–with 2, 0, and 1 counters respectively. Let's also say there were three games played, A beat B, A beat C, and B beat C. Then if we were to update the counters in one round, we'd start with A. Since A has two counters, we'd grab two random games; suppose they were both the game with B. Since A won that game, we'd flip two coins that come up heads with probability p. Say one came up heads and the other tails; this means we'd put one counter with A and one with B for the next round.

Since B doesn't have any counters in this round, we don't have any work. Finally, as C has one counter, we'd randomly choose a game they played; let's pick the game with B. Since B won the game, we'd move to counter to B with probability p. Let's say we do; then the counters for the next round would be A with 1, B with 2, and C with 0.

Try coding this up, using $p = 0.8$, $M = 100$, and 50 rounds cycling through all the teams. Then see if you can compare these to the previously used rankings.

Answers:

First, let's set the parameters and write a helper function to do most of the work (i.e. Step 2 above). Given a current set of counters, we'll write a function that takes a team and number of counters they currently have, and then redistributes them according to the games they played and coin flips.

```
p <- 0.8
M <- 100
counters <- rep(M, length(all.teams))
names(counters) <- all.teams

move.counters <- function(team, num.counters) {
    team.games <- subset(games, home == team | away == team)
    num.games <- nrow(team.games)
```

```r
    games.to.move.idx <- sample(1:num.games, num.counters, replace=TRUE)
    games.to.move <- team.games[games.to.move.idx, ]
    # now get the winners
    home.won <- with(games.to.move, homeScore - awayScore) > 0
    coin.flips <- sample(c(0,1), num.counters, replace=TRUE, prob=c(p, 1-p))
    # the coin flips only change the winner if 1
    move.to.home <- (home.won + coin.flips) %% 2

    next.counters <- c(
        games.to.move[move.to.home == 1, 'home'],
        games.to.move[move.to.home == 0, 'away']
    )
    table(next.counters)
}
```

Now, let's write the main for loop that moves the counters. We'll check the results to make sure it looks sane.

```r
set.seed(7)
for (round in 1:50) {
    # initialize the new counters after this round is complete
    new.counters <- rep(0, length(all.teams))
    names(new.counters) <- all.teams
    perm.teams <- all.teams[sample(length(all.teams))]
    for (team in perm.teams) {
        num.counters <- counters[team]
        moved.counters <- moveCounters(team, num.counters)
        # move the counters!
        new.counters[names(moved.counters)] <- (
            new.counters[names(moved.counters)] +
            moved.counters
        )
    }
    counters <- new.counters
}

counters <- sort(counters, decreasing=TRUE)
head(as.data.frame(counters),10)
```

```
##                          counters
## louisville-cardinals          632
## duke-blue-devils              548
## michigan-wolverines           477
## syracuse-orange               461
## gonzaga-bulldogs              447
## kansas-jayhawks               438
## ohio-state-buckeyes           437
## indiana-hoosiers              436
## miami-(fl)-hurricanes         434
## michigan-state-spartans       391
```

Cool, things look good. Let's finally compare them to previous ranks we used.

```r
# now put in the rank table
counter.rank <- rank(-counters,ties="min")
rank.table.idx <- sapply(row.names(rank.table), function (team) {
    which(team == names(counter.rank))
})
rank.table.idx <- unlist(rank.table.idx)
rank.table <- cbind(rank.table,
                    'Counter Rank'=counter.rank[rank.table.idx])
head(rank.table[order(rank.table[,'AP Rank']),], 25)
```

```
##                           Model Score Model Rank AP Rank USAT Rank
## gonzaga-bulldogs            9.9875221          4       1         1
## louisville-cardinals       12.0810415          3       2         2
## kansas-jayhawks             8.6318763          6       3         3
## indiana-hoosiers           13.2552278          1       4         5
## miami-(fl)-hurricanes       5.1915853         14       5         4
## duke-blue-devils            9.3936777          5       6         7
## ohio-state-buckeyes         7.8965499          7       7         6
## georgetown-hoyas            4.2589645         18       8         8
## michigan-state-spartans     6.1762258         12       9         9
## michigan-wolverines         7.5053732          8      10        11
## new-mexico-lobos            3.0044080         23      11        10
## kansas-state-wildcats       1.6525402         36      12        14
## saint-louis-billikens       2.6372882         26      13        13
## florida-gators             12.6385504          2      14        12
## marquette-golden-eagles     2.2985477         31      15        16
## syracuse-orange             7.0695669         10      16        18
## oklahoma-state-cowboys      3.6908930         20      17        19
## wisconsin-badgers           6.8313376         11      18        17
## memphis-tigers              1.0143519         42      19        15
## pittsburgh-panthers         7.3562577          9      20        22
## arizona-wildcats            4.6848829         16      21        20
## creighton-bluejays          5.2835829         13      22        21
## notre-dame-fighting-irish   1.9049621         35      23        NA
## ucla-bruins                 1.0059603         43      24        NA
## oregon-ducks                0.6666812         45      25        24
##                           Counter Rank
## gonzaga-bulldogs                     5
## louisville-cardinals                 1
## kansas-jayhawks                      6
## indiana-hoosiers                     8
## miami-(fl)-hurricanes                9
## duke-blue-devils                     2
## ohio-state-buckeyes                  7
## georgetown-hoyas                    13
## michigan-state-spartans             10
## michigan-wolverines                  3
## new-mexico-lobos                    11
## kansas-state-wildcats               25
## saint-louis-billikens               17
## florida-gators                      15
## marquette-golden-eagles             14
## syracuse-orange                      4
## oklahoma-state-cowboys              35
## wisconsin-badgers                   18
## memphis-tigers                      22
## pittsburgh-panthers                 29
## arizona-wildcats                    20
## creighton-bluejays                  16
## notre-dame-fighting-irish           21
## ucla-bruins                         31
## oregon-ducks                        19
```

We see that these rankings look relatively closer to the AP Rank than the margin based rankings. In fact, only the Oklahoma State Cowboys, the Pittsburgh Panthers, and the UCLA Bruins show up in the AP top 25 but are not the counter-based rankings.