Baseball World Series Champion

Contents

Introduction **1**

Exercices **2**

1. Limiting to Teams Making the Playoffs 2

Problem 1.1 2

Problem 1.2 3

Problem 1.3 3

Problem 1.4 4

2. Adding an Important Predictor 4

Problem 2.1 4

Problem 2.2 5

Problem 2.3 5

Problem 2.4 6

3. Bivariate Models for Predicting World Series Winner 6

Problem 3.1 6

Problem 3.2 6

4. Multivariate Models for Predicting World Series Winner 13

Problem 4.1 13

Problem 4.2 13

Problem 4.3 14

Introduction

Last week, in the Moneyball lecture, we discussed how regular season performance is not strongly correlated with winning the World Series in baseball. In this homework question, we'll use the same data to investigate how well we can predict the World Series winner at the beginning of the playoffs.

To begin, load the dataset baseball.csv into R using the read.csv function, and call the data frame "baseball". This is the same data file we used during the Moneyball lecture, and the data comes from Baseball-Reference.com.

As a reminder, this dataset contains data concerning a baseball team's performance in a given year. It has the following variables:

- ***Team*** : A code for the name of the team
- ***League*** : The Major League Baseball league the team belongs to, either AL (American League) or NL (National League)
- ***Year*** : The year of the corresponding record
- ***RS*** : The number of runs scored by the team in that year
- ***RA*** : The number of runs allowed by the team in that year
- ***W*** : The number of regular season wins by the team in that year
- ***OBP*** : The on-base percentage of the team in that year
- ***SLG*** : The slugging percentage of the team in that year
- ***BA*** : The batting average of the team in that year Playoffs: Whether the team made the playoffs in that year (1 for yes, 0 for no)
- ***RankSeason*** : Among the playoff teams in that year, the ranking of their regular season records (1 is best)
- ***RankPlayoffs*** : Among the playoff teams in that year, how well they fared in the playoffs. The team winning the World Series gets a RankPlayoffs of 1.
- ***G*** : The number of games a team played in that year
- ***OOBP*** : The team's opponents' on-base percentage in that year
- ***OSLG*** : The team's opponents' slugging percentage in that year

Exercices

1. Limiting to Teams Making the Playoffs

Problem 1.1

Each row in the baseball dataset represents a team in a particular year.

How many team/year pairs are there in the whole dataset?

```
## 'data.frame':    1232 obs. of  15 variables:
##  $ Team        : chr  "ARI" "ATL" "BAL" "BOS" ...
##  $ League      : chr  "NL" "NL" "AL" "AL" ...
##  $ Year        : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
##  $ RS          : int  734 700 712 734 613 748 669 667 758 726 ...
##  $ RA          : int  688 600 705 806 759 676 588 845 890 670 ...
##  $ W           : int  81 94 93 69 61 85 97 68 64 88 ...
##  $ OBP         : num  0.328 0.32 0.311 0.315 0.302 0.318 0.315 0.324 0.33 0.335 ...
##  $ SLG         : num  0.418 0.389 0.417 0.415 0.378 0.422 0.411 0.381 0.436 0.422 ...
##  $ BA          : num  0.259 0.247 0.247 0.26 0.24 0.255 0.251 0.251 0.274 0.268 ...
##  $ Playoffs    : int  0 1 1 0 0 0 1 0 0 1 ...
##  $ RankSeason  : int  NA 4 5 NA NA NA 2 NA NA 6 ...
##  $ RankPlayoffs: int  NA 5 4 NA NA NA 4 NA NA 2 ...
##  $ G           : int  162 162 162 162 162 162 162 162 162 162 ...
##  $ OOBP        : num  0.317 0.306 0.315 0.331 0.335 0.319 0.305 0.336 0.357 0.314 ...
##  $ OSLG        : num  0.415 0.378 0.403 0.428 0.424 0.405 0.39 0.43 0.47 0.402 ...
```

```
##      Team              League              Year            RS
##  Length:1232        Length:1232        Min.   :1962   Min.   : 463.0
##  Class :character   Class :character   1st Qu.:1977   1st Qu.: 652.0
##  Mode  :character   Mode  :character   Median :1989   Median : 711.0
##                                        Mean   :1989   Mean   : 715.1
##                                        3rd Qu.:2002   3rd Qu.: 775.0
```

```
##                                       Max.   :2012   Max.   :1009.0
##
##        RA               W              OBP              SLG
##  Min.   : 472.0   Min.   : 40.0   Min.   :0.2770   Min.   :0.3010
##  1st Qu.: 649.8   1st Qu.: 73.0   1st Qu.:0.3170   1st Qu.:0.3750
##  Median : 709.0   Median : 81.0   Median :0.3260   Median :0.3960
##  Mean   : 715.1   Mean   : 80.9   Mean   :0.3263   Mean   :0.3973
##  3rd Qu.: 774.2   3rd Qu.: 89.0   3rd Qu.:0.3370   3rd Qu.:0.4210
##  Max.   :1103.0   Max.   :116.0   Max.   :0.3730   Max.   :0.4910
##
##        BA            Playoffs        RankSeason     RankPlayoffs
##  Min.   :0.2140   Min.   :0.0000   Min.   :1.000   Min.   :1.000
##  1st Qu.:0.2510   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:2.000
##  Median :0.2600   Median :0.0000   Median :3.000   Median :3.000
##  Mean   :0.2593   Mean   :0.1981   Mean   :3.123   Mean   :2.717
##  3rd Qu.:0.2680   3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :0.2940   Max.   :1.0000   Max.   :8.000   Max.   :5.000
##                                    NA's   :988     NA's   :988
##        G              OOBP             OSLG
##  Min.   :158.0   Min.   :0.2940   Min.   :0.3460
##  1st Qu.:162.0   1st Qu.:0.3210   1st Qu.:0.4010
##  Median :162.0   Median :0.3310   Median :0.4190
##  Mean   :161.9   Mean   :0.3323   Mean   :0.4197
##  3rd Qu.:162.0   3rd Qu.:0.3430   3rd Qu.:0.4380
##  Max.   :165.0   Max.   :0.3840   Max.   :0.4990
##                  NA's   :812      NA's   :812
```

Answer : 1232

Explanation :
You can read the dataset into R by using the following command:

both show that there are 1232 team/year pairs.

Problem 1.2

Though the dataset contains data from 1962 until 2012, we removed several years with shorter-than-usual seasons.
Using the table() function, identify the total number of years included in this dataset.

```
## [1] 47
```

Answer : 47

Explanation :

contains 47 years (1972, 1981, 1994, and 1995 are missing). You can count the number of years in the table, or the command

directly provides the answer.

Problem 1.3

Because we're only analyzing teams that made the playoffs, use the subset() function to replace baseball with a data frame limited to teams that made the playoffs (so your subsetted data frame should still be

called “baseball”).
How many team/year pairs are included in the new dataset?

```
## [1] 244
```

Answer : 244

Explanation :

functions can be used to identify that 244 team/year pairs remain.

Problem 1.4

Through the years, different numbers of teams have been invited to the playoffs.
Which of the following has been the number of teams making the playoffs in some season? Select all that apply.

```
##
## 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1973 1974 1975 1976 1977 1978
##    2    2    2    2    2    2    2    4    4    4    4    4    4    4    4    4
## 1979 1980 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1996 1997
##    4    4    4    4    4    4    4    4    4    4    4    4    4    4    8    8
## 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
##    8    8    8    8    8    8    8    8    8    8    8    8    8    8   10
```

Answer :

1. **2**
2. **4**
3. 6
4. **8**
5. **10**
6. 12

Explanation :
Using

we can see at least one season had 2, 4, 8, and 10 contenders. A fancier approach would be to use

2. Adding an Important Predictor

Problem 2.1

It’s much harder to win the World Series if there are 10 teams competing for the championship versus just two. Therefore, we will add the predictor variable NumCompetitors to the baseball data frame. NumCompetitors will contain the number of total teams making the playoffs in the year of a particular team/year pair. For instance, NumCompetitors should be 2 for the 1962 New York Yankees, but it should be 8 for the 1998 Boston Red Sox.

We start by storing the output of the table() function that counts the number of playoff teams from each year:

You can output the table with the following command:

We will use this stored table to look up the number of teams in the playoffs in the year of each team/year pair.

Just as we can use the names() function to get the names of a data frame's columns, we can use it to get the names of the entries in a table.
What best describes the output of names(PlayoffTable)?

```
##  chr [1:47] "1962" "1963" "1964" "1965" "1966" "1967" "1968" "1969" "1970" ...
```

Answer :

1. Vector of years stored as numbers (type num)
2. **Vector of years stored as strings (type chr)**
3. Vector of frequencies stored as numbers (type num)
4. Vector of frequencies stored as strings (type chr)

Explanation :
From the call

we see PlayoffTable has names of type chr, which are the years of the teams in the dataset.

Problem 2.2

Given a vector of names, the table will return a vector of frequencies.
Which function call returns the number of playoff teams in 1990 and 2001?
(HINT: If you are not sure how these commands work, go ahead and try them out in your R console!)

Answer :

1. PlayoffTable(1990, 2001)
2. PlayoffTable(c(1990, 2001))
3. PlayoffTable("1990", "2001")
4. PlayoffTable(c("1990", "2001"))
5. PlayoffTable[1990, 2001]
6. PlayoffTable[c(1990, 2001)]
7. PlayoffTable["1990", "2001"]
8. **PlayoffTable[c("1990", "2001")]**

Explanation :
Because PlayoffTable is an object and not a function, we look up elements in it with square brackets instead of parentheses. We build the vector of years to be passed with the c() function. Because the names of PlayoffTable are strings and not numbers, we need to pass "1990" and "2001".

Problem 2.3

Putting it all together, we want to look up the number of teams in the playoffs for each team/year pair in the dataset, and store it as a new variable named NumCompetitors in the baseball data frame.
While of the following function calls accomplishes this?
(HINT: Test out the functions if you are not sure what they do.)

Answer :

1. baseball$NumCompetitors = PlayoffTable(baseball$Year)

2. baseball$NumCompetitors = PlayoffTable[baseball$Year]
3. baseball$NumCompetitors = PlayoffTable(as.character(baseball$Year))
4. **baseball$NumCompetitors = PlayoffTable[as.character(baseball$Year)]**
   Explanation :
   Because PlayoffTable is an object and not a function, we look up elements in it with square brackets instead of parentheses. as.character() is needed to convert the Year variable in the dataset to a string, which we know from the previous parts is needed to look up elements in a table. If you're not sure what a function does, remember you can look it up with the ? function. For instance, you could type ?as.character to look up information about as.character.

Problem 2.4

Add the NumCompetitors variable to your baseball data frame.
How many playoff team/year pairs are there in our dataset from years where 8 teams were invited to the playoffs?

```
## [1] 128
```

Answer : 128

Explanation :
You can add the NumCompetitors variable to the baseball data frame with the following command:

Then you can obtain the number of team/year pairs with 8 teams in the playoffs by running

3. Bivariate Models for Predicting World Series Winner

Problem 3.1

In this problem, we seek to predict whether a team won the World Series; in our dataset this is denoted with a RankPlayoffs value of 1. Add a variable named WorldSeries to the baseball data frame, by typing the following command in your R console:

WorldSeries takes value 1 if a team won the World Series in the indicated year and a 0 otherwise.
How many observations do we have in our dataset where a team did NOT win the World Series?

```
## [1] 197
```

Answer : 197

Explanation :
You can create the WorldSeries variable by running the command:

Then, if you create the table:

You can see that there are 197 teams that did not win the World Series.

Problem 3.2

When we're not sure which of our variables are useful in predicting a particular outcome, it's often helpful to build bivariate models, which are models that predict the outcome using a single independent variable.
Which of the following variables is a significant predictor of the WorldSeries variable in a bivariate logistic regression model?

To determine significance, remember to look at the stars in the summary output of the model. We'll define an independent variable as significant if there is at least one star at the end of the coefficients row for that variable (this is equivalent to the probability column having a value smaller than 0.05). Note that you have to build 12 models to answer this question! Use the entire dataset baseball to build the models.
(Select all that apply.)

```
## [[1]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0297  -0.6797  -0.5435  -0.4648   2.1504
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 72.23602   22.64409    3.19  0.00142 **
## Year        -0.03700    0.01138   -3.25  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 228.35  on 242  degrees of freedom
## AIC: 232.35
##
## Number of Fisher Scoring iterations: 4
##
##
## [[2]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8254  -0.6819  -0.6363  -0.5561   2.0308
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.661226   1.636494   0.404    0.686
## RS          -0.002681   0.002098  -1.278    0.201
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 237.45  on 242  degrees of freedom
## AIC: 241.45
##
## Number of Fisher Scoring iterations: 4
```

```
##
##
## [[3]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9749  -0.6883  -0.6118  -0.4746   2.1577
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.888174   1.483831   1.272   0.2032
## RA          -0.005053   0.002273  -2.223   0.0262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 233.88  on 242  degrees of freedom
## AIC: 237.88
##
## Number of Fisher Scoring iterations: 4
##
##
## [[4]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0623  -0.6777  -0.6117  -0.5367   2.1254
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.85568    2.87620  -2.384   0.0171 *
## W            0.05671    0.02988   1.898   0.0577 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 235.51  on 242  degrees of freedom
## AIC: 239.51
##
## Number of Fisher Scoring iterations: 4
##
##
```

```
## [[5]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8071  -0.6749  -0.6365  -0.5797   1.9753
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.741      3.989   0.687    0.492
## OBP          -12.402     11.865  -1.045    0.296
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 238.02  on 242  degrees of freedom
## AIC: 242.02
##
## Number of Fisher Scoring iterations: 4
##
##
## [[6]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9498  -0.6953  -0.6088  -0.5197   2.1136
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.200      2.358   1.357   0.1748
## SLG          -11.130      5.689  -1.956   0.0504 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 235.23  on 242  degrees of freedom
## AIC: 239.23
##
## Number of Fisher Scoring iterations: 4
##
##
## [[7]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
```

```
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6797  -0.6592  -0.6513  -0.6389   1.8431
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6392     3.8988  -0.164    0.870
## BA           -2.9765    14.6123  -0.204    0.839
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 239.08  on 242  degrees of freedom
## AIC: 243.08
##
## Number of Fisher Scoring iterations: 4
##
##
## [[8]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7805  -0.7131  -0.5918  -0.4882   2.1781
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8256     0.3268  -2.527   0.0115 *
## RankSeason   -0.2069     0.1027  -2.016   0.0438 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 234.75  on 242  degrees of freedom
## AIC: 238.75
##
## Number of Fisher Scoring iterations: 4
##
##
## [[9]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -0.5318  -0.5176  -0.5106  -0.5023   2.0697
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9306     8.3728  -0.111    0.912
## OOBP         -3.2233    26.0587  -0.124    0.902
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 84.926  on 113  degrees of freedom
## Residual deviance: 84.910  on 112  degrees of freedom
##   (130 observations deleted due to missingness)
## AIC: 88.91
##
## Number of Fisher Scoring iterations: 4
##
##
## [[10]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5610  -0.5209  -0.5088  -0.4902   2.1268
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.08725    6.07285  -0.014    0.989
## OSLG        -4.65992   15.06881  -0.309    0.757
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 84.926  on 113  degrees of freedom
## Residual deviance: 84.830  on 112  degrees of freedom
##   (130 observations deleted due to missingness)
## AIC: 88.83
##
## Number of Fisher Scoring iterations: 4
##
##
## [[11]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9871  -0.8017  -0.5089  -0.5089   2.2643
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)     0.03868    0.43750   0.088 0.929559
## NumCompetitors -0.25220    0.07422  -3.398 0.000678 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.96  on 242  degrees of freedom
## AIC: 230.96
##
## Number of Fisher Scoring iterations: 4
##
##
## [[12]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnam[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6772  -0.6772  -0.6306  -0.6306   1.8509
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3558     0.2243  -6.045  1.5e-09 ***
## LeagueNL     -0.1583     0.3252  -0.487    0.626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 238.88  on 242  degrees of freedom
## AIC: 242.88
##
## Number of Fisher Scoring iterations: 4
```

Answer :

1. **Year**
2. RS
3. **RA**
4. W
5. OBP
6. SLG
7. BA
8. **RankSeason**
9. OOBP
10. OSLG
11. **NumCompetitors**
12. League

Explanation :
The results come from building each bivariate model and looking at its summary. For instance, the result for the variable Year can be obtained by running

You can save time on repeated model building by using the up arrow in your R terminal. The W and SLG variables were both nearly significant, with p = 0.0577 and 0.0504, respectively.

4. Multivariate Models for Predicting World Series Winner

Problem 4.1

In this section, we'll consider multivariate models that combine the variables we found to be significant in bivariate models. Build a model using all of the variables that you found to be significant in the bivariate models.
How many variables are significant in the combined model?

```
##
## Call:
## glm(formula = WorldSeries ~ Year + RA + RankSeason + NumCompetitors,
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0336  -0.7689  -0.5139  -0.4583   2.2195
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    12.5874376 53.6474210   0.235    0.814
## Year           -0.0061425  0.0274665  -0.224    0.823
## RA             -0.0008238  0.0027391  -0.301    0.764
## RankSeason     -0.0685046  0.1203459  -0.569    0.569
## NumCompetitors -0.1794264  0.1815933  -0.988    0.323
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.37  on 239  degrees of freedom
## AIC: 236.37
##
## Number of Fisher Scoring iterations: 4
```

Answer : 0

Explanation :
You can create a model with all of the significant variables from the bivariate models (Year, RA, RankSeason, and NumCompetitors) by using the following command:

Looking at summary(LogModel), you can see that none of the variables are significant in the multivariate model!

Problem 4.2

Often, variables that were significant in bivariate models are no longer significant in multivariate analysis due to correlation between the variables.

Which of the following variable pairs have a high degree of correlation (a correlation greater than 0.8 or less than -0.8)?
Select all that apply.

```
##                     Year        RA RankSeason NumCompetitors
## Year           1.0000000 0.4762422  0.3852191      0.9139548
## RA             0.4762422 1.0000000  0.3991413      0.5136769
## RankSeason     0.3852191 0.3991413  1.0000000      0.4247393
## NumCompetitors 0.9139548 0.5136769  0.4247393      1.0000000
```

Answer :

1. Year/RA
2. Year/RankSeason
3. **Year/NumCompetitors**
4. RA/RankSeason
5. RA/NumCompetitors
6. RankSeason/NumCompetitors

Explanation :
To test the correlation between two variables, use a command like

While every pair was at least moderately correlated, the only strongly correlated pair was Year/NumCompetitors, with correlation coefficient 0.914. As a shortcut, you can compute all pair-wise correlations between these variables with:

Problem 4.3

Build all six of the two variable models listed in the previous problem. Together with the four bivariate models, you should have 10 different logistic regression models.
Which model has the best AIC value (the minimum AIC value)?

```
## [[1]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0297  -0.6797  -0.5435  -0.4648   2.1504
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 72.23602   22.64409    3.19  0.00142 **
## Year        -0.03700    0.01138   -3.25  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
```

```
## Residual deviance: 228.35  on 242  degrees of freedom
## AIC: 232.35
##
## Number of Fisher Scoring iterations: 4
##
##
## [[2]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9749  -0.6883  -0.6118  -0.4746   2.1577
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.888174   1.483831   1.272   0.2032
## RA          -0.005053   0.002273  -2.223   0.0262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 233.88  on 242  degrees of freedom
## AIC: 237.88
##
## Number of Fisher Scoring iterations: 4
##
##
## [[3]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7805  -0.7131  -0.5918  -0.4882   2.1781
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8256     0.3268  -2.527   0.0115 *
## RankSeason   -0.2069     0.1027  -2.016   0.0438 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 234.75  on 242  degrees of freedom
## AIC: 238.75
```

```
##
## Number of Fisher Scoring iterations: 4
##
##
## [[4]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9871  -0.8017  -0.5089  -0.5089   2.2643
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.03868    0.43750   0.088 0.929559
## NumCompetitors -0.25220    0.07422  -3.398 0.000678 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.96  on 242  degrees of freedom
## AIC: 230.96
##
## Number of Fisher Scoring iterations: 4
##
##
## [[5]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0402  -0.6878  -0.5298  -0.4785   2.1370
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 63.610741  25.654830   2.479   0.0132 *
## Year        -0.032084   0.013323  -2.408   0.0160 *
## RA          -0.001766   0.002585  -0.683   0.4945
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 227.88  on 241  degrees of freedom
## AIC: 233.88
##
```

```
## Number of Fisher Scoring iterations: 4
##
##
## [[6]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0560  -0.6957  -0.5379  -0.4528   2.2673
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 63.64855   24.37063   2.612  0.00901 **
## Year        -0.03254    0.01231  -2.643  0.00822 **
## RankSeason  -0.10064    0.11352  -0.887  0.37534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 227.55  on 241  degrees of freedom
## AIC: 233.55
##
## Number of Fisher Scoring iterations: 4
##
##
## [[7]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0050  -0.7823  -0.5115  -0.4970   2.2552
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    13.350467  53.481896   0.250    0.803
## Year           -0.006802   0.027328  -0.249    0.803
## NumCompetitors -0.212610   0.175520  -1.211    0.226
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.90  on 241  degrees of freedom
## AIC: 232.9
##
## Number of Fisher Scoring iterations: 4
##
```

```
##
## [[8]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9374  -0.6933  -0.5936  -0.4564   2.1979
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.487461   1.506143   0.988    0.323
## RA          -0.003815   0.002441  -1.563    0.118
## RankSeason  -0.140824   0.110908  -1.270    0.204
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 232.22  on 241  degrees of freedom
## AIC: 238.22
##
## Number of Fisher Scoring iterations: 4
##
##
## [[9]]
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0433  -0.7826  -0.5133  -0.4701   2.2208
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.716895   1.528736   0.469  0.63911
## RA             -0.001233   0.002661  -0.463  0.64313
## NumCompetitors -0.229385   0.088399  -2.595  0.00946 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.74  on 241  degrees of freedom
## AIC: 232.74
##
## Number of Fisher Scoring iterations: 4
##
##
## [[10]]
```

```
##
## Call:
## glm(formula = paste("WorldSeries", baseballvarnamBI[x], sep = " ~ "),
##     family = binomial, data = baseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0090  -0.7592  -0.5204  -0.4501   2.2562
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.12277    0.45737   0.268  0.78837
## RankSeason     -0.07697    0.11711  -0.657  0.51102
## NumCompetitors -0.22784    0.08201  -2.778  0.00546 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.52  on 241  degrees of freedom
## AIC: 232.52
##
## Number of Fisher Scoring iterations: 4
```

Answer :

1. Year
2. RA
3. RankSeason
4. **NumCompetitors**
5. Year/RA
6. Year/RankSeason
7. Year/NumCompetitors
8. RA/RankSeason
9. RA/NumCompetitors
10. RankSeason/NumCompetitors

Explanation :
The two-variable models can be built with the following commands:

None of the models with two independent variables had both variables significant, so none seem promising as compared to a simple bivariate model. Indeed the model with the lowest AIC value is the model with just NumCompetitors as the independent variable.

This seems to confirm the claim made by Billy Beane in Moneyball that all that matters in the Playoffs is luck, since NumCompetitors has nothing to do with the quality of the teams!