Back to Back NBA Games

Nicholas Capofari

December 1, 2015

Contents

Research Question

Are NBA teams affected by playing games on back to back days?
Basketball is a demanding sport. Each NBA team, on average, runs over a combined 16 miles per game. Due to scheduling demands, teams are sometimes forced to play on back to back days. For example, in this upcoming season, teams will play anywhere from 14 to 20 games the day after playing a game the previous day. If playing games on back to back days has an adverse effect on a team's chance to win, then teams with more back-to-backs are at an unfair disadvantage. When playoff positions are determined by 1 or 2 games, the scheduling decisions made during the off-season could end up costing teams millions of dollars in revenue. My goal for this project is to determine if such a disadvantage exists.

If there is a disadvantage I would like to use what we have learned in chapter 8 of our text book and apply multiple regression techniques to see if I can develop a model to predict when teams will win or lose their back to back contests.

This is my first data project and I find the work very exciting. At the same time I understand the limits of what I am able to do. I will do my best to highlight any errors I commit during this process.

Data Collection

Basketball-Reference.com is a great website that warehouses a vast amount of NBA data. For example, using the website you can access every box score of an NBA game since 1954. I will extract the data from this site.

I will store this data in a data frame. I will also try to extract any information that may be helpful later in my exploration. While conducting the data collection, it became apparent to me that the amount of data I would need to make a proper model is outside the scope of what I am currently capable of (due to time and/or skill restrictions). I would like to revisit this topic later on in my academic career to see how I have progressed.

```
library(XML)
library(RCurl)
library(dplyr)
library(tidyr)
```

```
library(lubridate)
library(stringr)
library(scales)
library(ggplot2)
library(gridExtra)
```

A team's average age may play a part in determining how the team performs playing back to back games.

```
years <- c(2006:2015)
#create data frame to store team average age
team_ages <- data.frame()
for(year in years){
  age_url <- getURL(sprintf("http://www.basketball-reference.com/leagues/NBA_%s.html", year))
  age_parsed <- htmlParse(age_url, encoding="UTF-8")
  age_df <- data.frame(readHTMLTable(age_parsed)[22])
  temp_df <- data.frame(yr=year,
                        team=str_replace_all(age_df[,2],"\\*",""),
                        age=age_df[,3])
  team_ages <- data.frame(rbind(team_ages, temp_df))
}

#helper function to find a team's average age for a season
#parameters are t (full team name) and y (year)
age_func <- function(t, y){
  temp <- subset(team_ages, team_ages$yr==y & team_ages$team==t)[,3]
  age <- as.numeric(as.character(temp))
  return(age)
}
```

A team's opponent's winning % will also be a factor.

```
team_wins <- data.frame()
for(year in years){
  opp_win_url <- getURL(sprintf("http://www.basketball-reference.com/leagues/NBA_%s_standings.html", yea
  ow_parsed <- htmlParse(opp_win_url, encoding="UTF-8")
  ow_df <- data.frame(readHTMLTable(ow_parsed)[4], stringsAsFactors = FALSE)
  ow_df <- ow_df[,2:3]
  ow_df <- separate(ow_df, expanded.standings.Overall,
                    into = c("wins", "losses"), "-")
  temp_df <- data.frame(yr=year,
                        team=ow_df$expanded.standings.Team,
                        wins=ow_df$wins,
                        losses=ow_df$losses)
  team_wins <- data.frame(rbind(team_wins, temp_df))
}
team_wins$prop <- as.numeric(as.character(team_wins$wins))/
  (as.numeric(as.character(team_wins$wins))+as.numeric(as.character(team_wins$losses)))

#helper function to find opponent's win%
#parameters are oppisng team name and year
opp_win_func <- function(opp_team, y){
  w_prop <- subset(team_wins, team_wins$yr==y &
                     team_wins$team==opp_team)[,5]
```

```
  return(as.numeric(unlist(w_prop)))
}
```

Now, to collect our data for all teams since the 2005-2006 season.

```
#retrieve team abbreviations of all active NBA teams
franchise_url <- getURL("http://www.basketball-reference.com/about/franchises.html")
franchise_parsed <- htmlParse(franchise_url, encoding = "UTF-8")
franchise_names <- data.frame(readHTMLTable(franchise_parsed),
                              stringsAsFactors = FALSE)
franchise_names <- subset(franchise_names,
                          str_detect(franchise_names$NULL.Last.Year,
                                     "^(a|2)"))
franchise_names$NULL.Team.Name = str_replace_all(
  franchise_names$NULL.Team.Name, "NO", "New Orleans")
franchise_names$NULL.Team.Name = str_replace_all(
  franchise_names$NULL.Team.Name, "Supers", "SuperS")
```

```
#create an empty data frame that will store the data
#of every back to back game played in the NBA since
#the 2005-2006 season
back_to_backs <- data.frame()
all_teams <- paste(team_wins$yr, team_wins$team, sep="")
for(each_team in all_teams){
    #for error checking if loop does not complete
    write(each_team, "each_team_error.txt")
    #get abbreviation of team
    full_team_name <- str_extract(each_team, "\\D{4}(.+)")
    team <- as.character(subset(franchise_names,
                                franchise_names$NULL.Team.Name ==
                                  full_team_name)[,2])
    year <- as.integer(str_extract(each_team, "^\\d{4}"))
    #base url for our data
    base_url <- "http://www.basketball-reference.com/teams/%s/%s_games.html"
    #insert year and team abbreviation
    my_url <- sprintf(base_url, team, year)
    team_url <- getURL(my_url)
    #if url is bad, move on
    if(team_url == ""){ next }
    team_parsed <- htmlParse(team_url, encoding = "UTF-8")
    team_scores <- data.frame(readHTMLTable(
        team_parsed)[1], stringsAsFactors = FALSE)
    #remove header rows
    team_scores <- team_scores[team_scores$teams_games.G != 'G', ]
    #streak should represent win/lose streak
    #going into a game - not after the game
    temp <- team_scores$teams_games.Streak
    temp <- c("W 0", as.character(temp[1:length(temp)-1]))
    #change streak to an integer ex. L 3 = -3
    temp <- str_replace(temp, "L ", "-")
    temp <- str_replace(temp, "W ", "")
    team_scores$teams_games.Streak <- as.integer(temp)
    #keep track of rest days between games
```

```
team_scores <- team_scores %>%
           mutate(days_off=-1+difftime(
                    mdy(str_sub(
                      teams_games.Date,6)),
                    mdy(str_sub(
                      lag(teams_games.Date,1),6)),
                    units="days"))
#using 60 as the number of rest days between seasons
team_scores[is.na(team_scores)] <- 60
team_scores$team <- team
team_scores$full_team_name <- full_team_name
team_scores$season <- paste(year-1, year, sep = "")
team_scores <- team_scores %>%
              mutate(games=max(as.numeric(levels(
                      teams_games.G))[teams_games.G]),
                    wins=max(as.numeric(levels(
                      teams_games.W))[teams_games.W]),
                    losses=games-wins,
                    win_prop=wins/(wins+losses),
                    h_wins=sum(teams_games..3 == ""
                            & teams_games..4 == "W"),
                    h_losses=sum(teams_games..3 == ""
                            & teams_games..4 == "L"),
                    h_win_prop=h_wins/(h_wins+h_losses),
                    a_wins=wins-h_wins,
                    a_losses=losses-h_losses,
                    a_win_prop=a_wins/(a_wins+a_losses),
                    c_wins=cumsum(teams_games..4 == "W"),
                    c_losses=cumsum(teams_games..4 == "L"),
                    c_win_prop=c_wins/(c_wins+c_losses),
                    back_to_back=ifelse(days_off == 0,
                                        "TRUE", "FALSE"),
                    bb_g_number=cumsum(back_to_back == TRUE),
                    bb_g_total=sum(back_to_back == TRUE),
                    bb_wins=sum(back_to_back == TRUE
                            & teams_games..4 == "W"),
                    bb_losses=sum(back_to_back == TRUE
                            & teams_games..4 == "L"),
                    bb_win_prop=bb_wins/(bb_wins+bb_losses),
                    bb_loc=paste(
                      ifelse(lag(teams_games..3)=="@","a","h"),
                      ifelse(teams_games..3=="@","a","h"), sep=""),
                    prev_win=ifelse(
                      lag(teams_games..4)=="W",TRUE,FALSE))
#keep only the back to back games
team_scores <- team_scores[team_scores$back_to_back == TRUE, ]
#add opponent winning %
team_scores$opp_win_prop=unlist(lapply(as.character(
  team_scores$teams_games.Opponent), opp_win_func, year))
#add back to back home away wins and losses
team_scores <- team_scores %>% mutate(
    win_prop_diff=win_prop-opp_win_prop,
    avg_age=age_func(full_team_name, year),
```

```
        day=substr(teams_games.Date,1,2),
        time=as.integer(substr(teams_games.,1,1)),
        bb_h_wins=sum(str_detect(bb_loc, "^.h$")
                      & teams_games..4 == "W"),
        bb_h_losses=sum(str_detect(bb_loc, "^.h$")
                      & teams_games..4 == "L"),
        bb_h_win_prop=bb_h_wins/(bb_h_wins+bb_h_losses),
        bb_a_wins=bb_wins-bb_h_wins,
        bb_a_losses=bb_losses-bb_h_losses,
        bb_a_win_prop=bb_a_wins/(bb_a_wins+bb_a_losses))
    #add data frame to master data frame
    back_to_backs <- rbind(back_to_backs, team_scores)
}
#remove unused columns
back_to_backs <- select(back_to_backs, -c(4,5,12,13,15))
#rename certain columns
colnames(back_to_backs)[1:10] <- c("g_number","date", "long_time",
                        "home_away", "opponent", "result", "ot",
                        "tm_points", "opp_points", "streak")
#place team and season first
back_to_backs <- back_to_backs[, c(12:15,1:11,16:46)]
#save a local copy of data frame
write.csv(back_to_backs, file = "NBA_back_to_backs.csv")
```

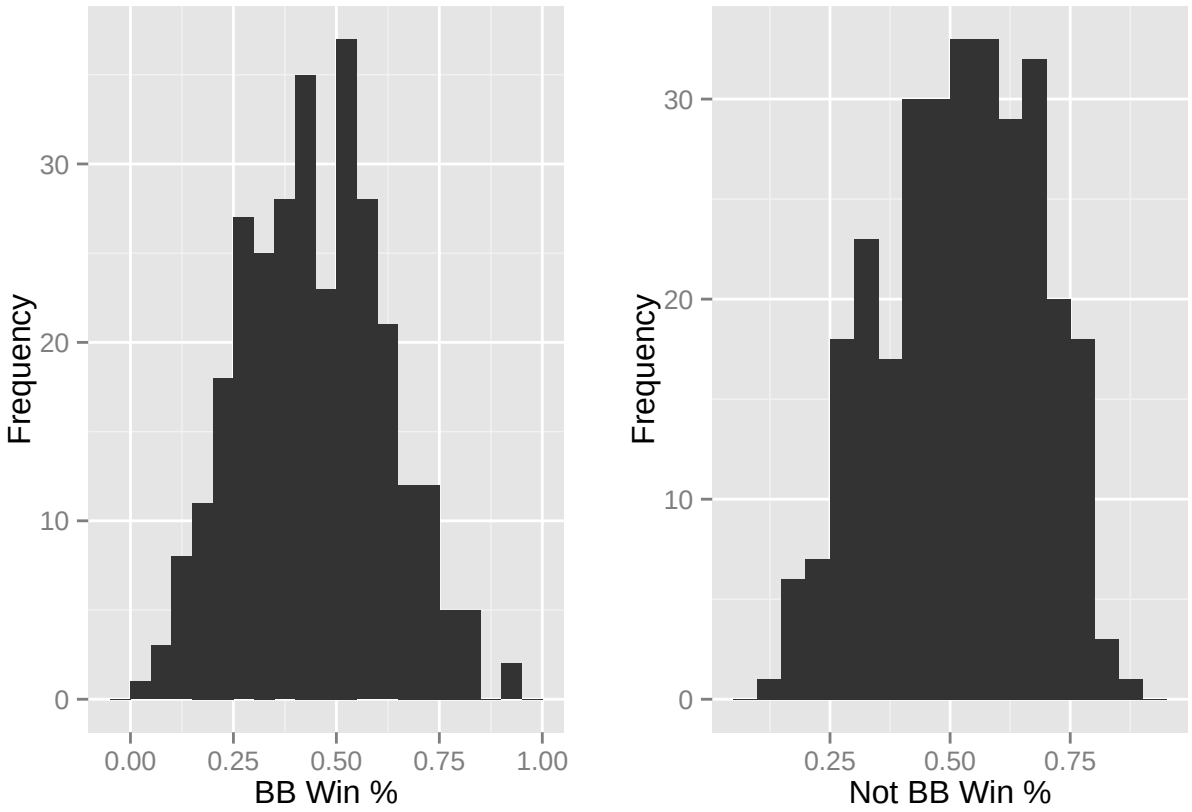
Hypothesis Testing

Is there convincing evidence that NBA teams have a different winning percentage playing the 2nd game of back to back games?

```
#each team for each season
team_bb <- back_to_backs[,c(1:4,16:24,30:33,38,41:46)]
team_bb <- unique(team_bb)
#separate back to back wins and losses
team_bb <- team_bb %>% mutate(win_prop_not_bb=(wins-bb_wins)/(games-bb_g_total))
```

Before we conduct the hypothesis test, we must first check the diagnostics and see if out data fits the requirements of performing a t-test. Each observation is independent and it represents less than 10% of all NBA games. Below are the distributions of the variables that we are investigating. Both come from a near normal distribution.

```
plot1 <- qplot(bb_win_prop, data=team_bb,
      geom="histogram", binwidth=.05) + xlab("BB Win %") + ylab("Frequency")
plot2 <- qplot(win_prop_not_bb, data=team_bb,
      geom="histogram", binwidth=.05) + xlab("Not BB Win %") + ylab("Frequency")
grid.arrange(plot1, plot2, ncol=2)
```



The null hypothesis represents that there is no difference between winning percentages for NBA teams playing the second game of back to back games.

H_o: There is no difference between winning percentage for NBA teams playing the second game of back to back games.
H_a: There is some difference between the winning percentages.

```
ttest <- t.test(team_bb$bb_win_prop, team_bb$win_prop_not_bb)
ttest
```

```
##
##  Welch Two Sample t-test
##
## data:  team_bb$bb_win_prop and team_bb$win_prop_not_bb
## t = -5.6317, df = 596.32, p-value = 2.75e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.10300500 -0.04973888
## sample estimates:
## mean of x mean of y
## 0.4420958 0.5184678
```

Since the p-value is less than 0.01, we have significant evidence to reject the NULL hypothesis and accept the alternative. There is a difference between winning percentages for NBA teams playing the second game of back to back games.

```
h <- as.numeric((table(back_to_backs$home_away))[1])
a <- as.numeric((table(back_to_backs$home_away))[2])
```

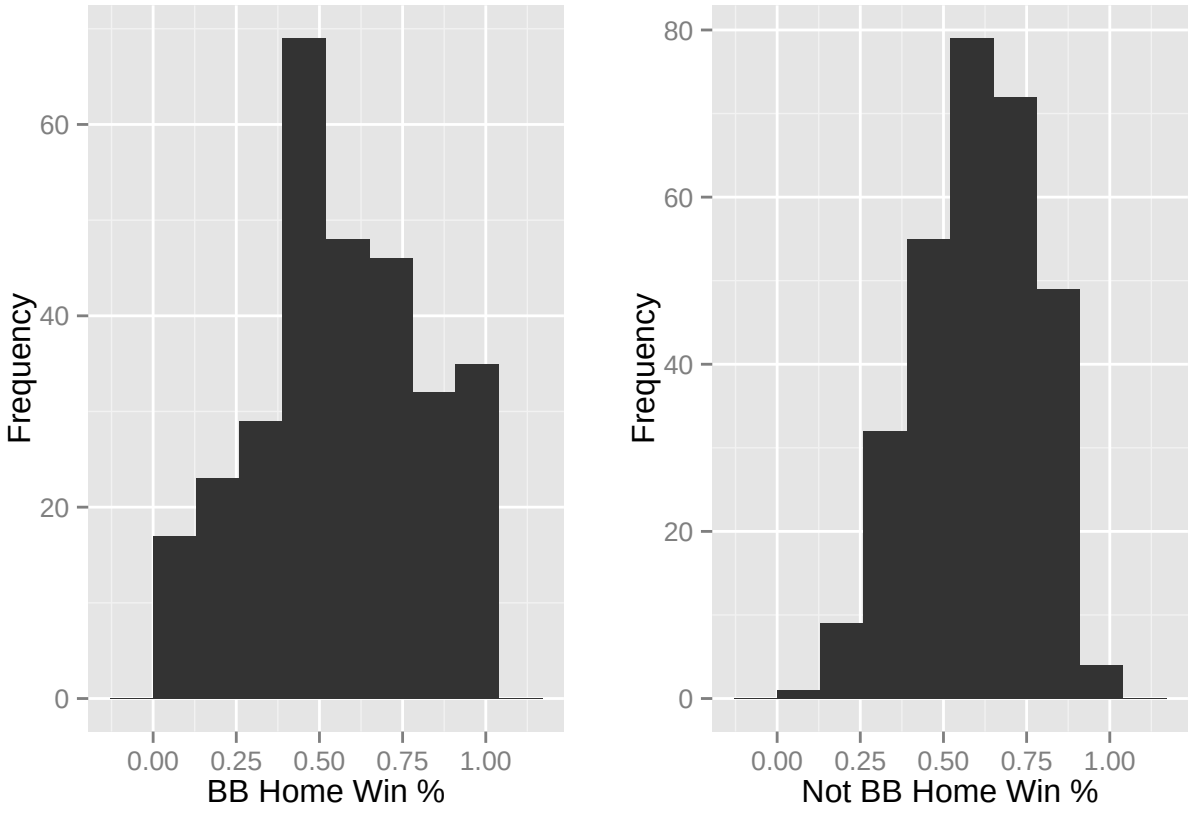
This conclusion is misleading. `67.9%` of the games in question are played on the road. Wee need to separate home and road records to really see if there is a significant relationship.

Is there convincing evidence that NBA teams have a different winning percentage playing the 2nd game of back to back games when they play at home?

```
#separate home back to back wins and losses
team_bb <- team_bb %>% mutate(win_prop_home_not_bb=(h_wins-bb_h_wins)/
                                  (h_wins+h_losses-(bb_h_wins+bb_h_losses)))
```

Before we conduct the hypothesis test, we must first check the diagnostics and see if out data fits the requirements of performing a t-test. Each observation is independent and it represents less than 10% of the population. Below are the distributions of the variables that we are investigating. Both come from a near normal distribution.

```
plot1 <- qplot(bb_h_win_prop, data=team_bb,
      geom="histogram", binwidth=.13) + xlab("BB Home Win %") + ylab("Frequency")
plot2 <- qplot(win_prop_home_not_bb, data=team_bb,
      geom="histogram", binwidth=.13) + xlab("Not BB Home Win %") + ylab("Frequency")
grid.arrange(plot1, plot2, ncol=2)
```



The null hypothesis represents that there is no difference between home winning percentages for NBA teams playing the second game of back to back games at home.

H_o: There is no difference between home winning percentage for NBA teams playing the second game of back to back games at home.
H_a: There is some difference between the winning percentages.

```
ttest_home <- t.test(team_bb$bb_h_win_prop, team_bb$win_prop_home_not_bb)
ttest_home
```

```
##
##  Welch Two Sample t-test
##
## data:  team_bb$bb_h_win_prop and team_bb$win_prop_home_not_bb
## t = -1.8716, df = 520.28, p-value = 0.06182
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.069959059  0.001694134
## sample estimates:
## mean of x mean of y
## 0.5658783 0.6000107
```

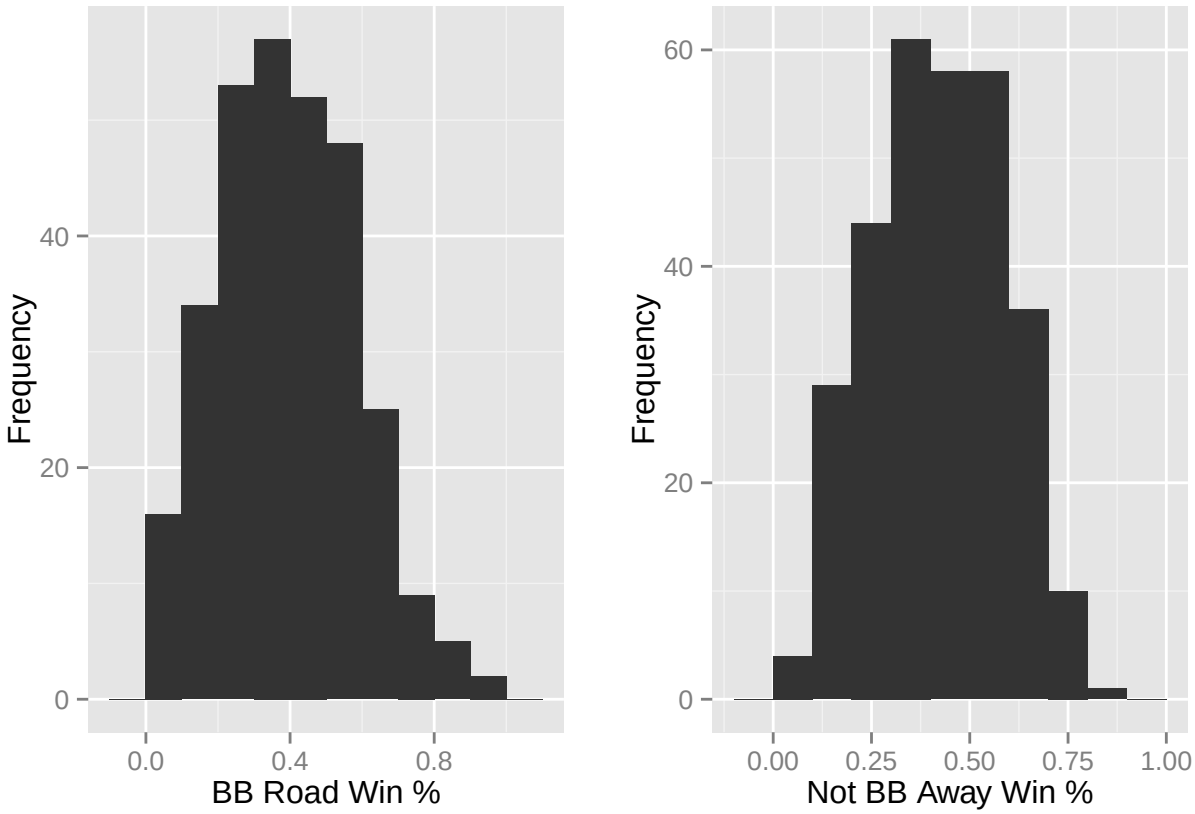
Since the p-value is greater than 0.05, we do not have significant evidence to reject the NULL hypothesis and accept the alternative. There is no significant difference between home winning percentages for NBA teams playing the second game of back to back games at home.

Is there convincing evidence that NBA teams have a different winning percentage playing the 2nd game of back to back games when they play on the road?

```
#separate away back to back wins and losses
team_bb <- team_bb %>% mutate(win_prop_away_not_bb=(a_wins-bb_a_wins)/
                                (a_wins+a_losses-(bb_a_wins+bb_a_losses)))
```

Once again, we must perform diagnostic tests before we conduct hypothesis testing. Each observation is independent and it represents less than 10% of all NBA games in question. Below are the distributions of the variables that we are investigating. Both come from a near normal distribution.

```
plot1 <- qplot(bb_a_win_prop, data=team_bb,
      geom="histogram", binwidth=.10) + xlab("BB Road Win %") + ylab("Frequency")
plot2 <- qplot(win_prop_away_not_bb, data=team_bb,
      geom="histogram", binwidth=.10) + xlab("Not BB Away Win %") + ylab("Frequency")
grid.arrange(plot1, plot2, ncol=2)
```



The null hypothesis represents that there is no difference between road winning percentages for NBA teams playing the second game of back to back games on the road.

H_o: There is no difference between road winning percentage for NBA teams playing the second game of back to back games on the road.
H_a: There is some difference between the winning percentages.

```
ttest_away <- t.test(team_bb$bb_a_win_prop, team_bb$win_prop_away_not_bb)
ttest_away
```

```
##
##  Welch Two Sample t-test
##
## data:  team_bb$bb_a_win_prop and team_bb$win_prop_away_not_bb
## t = -2.1926, df = 593.26, p-value = 0.02872
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.058799675 -0.003234057
## sample estimates:
## mean of x mean of y
## 0.3838059 0.4148228
```

Since the p-value is smaller than 0.05, we do have significant evidence to reject the NULL hypothesis and accept the alternative. There is a significant difference between road winning percentages for NBA teams playing the second game of back to back games on the road.

Model Construction

Since there is a significant difference in average winning percentages for NBA teams playing the second game of back to back games on the road, I will use logistic regression to build a model to predict if a team will win or lose that game. Logistic regression is a tool for building models when there is a categorical response variable with two levels. I will choose to use logit transformation to ensure that my results are between 0 and 1.

```
away_bb <- subset(back_to_backs, back_to_backs$home_away=="@")
full_model_away <- glm(away_bb$result ~ away_bb$avg_age +
                         away_bb$a_win_prop + away_bb$day +
                         away_bb$time + away_bb$bb_loc +
                         away_bb$streak + away_bb$prev_win +
                         away_bb$win_prop_diff, family=binomial("logit"))
summary(full_model_away)
```

```
##
## Call:
## glm(formula = away_bb$result ~ away_bb$avg_age + away_bb$a_win_prop +
##     away_bb$day + away_bb$time + away_bb$bb_loc + away_bb$streak +
##     away_bb$prev_win + away_bb$win_prop_diff, family = binomial("logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1849  -0.8827  -0.5262   0.9999   2.4995
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.21549    0.74452  -0.289  0.77225
## away_bb$avg_age       -0.04717    0.02645  -1.783  0.07451 .
## away_bb$a_win_prop     1.03592    0.38008   2.726  0.00642 **
## away_bb$dayMo          0.31521    0.18254   1.727  0.08421 .
## away_bb$daySa          0.24805    0.16557   1.498  0.13409
## away_bb$daySu          0.18617    0.22007   0.846  0.39756
## away_bb$dayTh          0.13307    0.21459   0.620  0.53518
## away_bb$dayTu          0.28883    0.23014   1.255  0.20948
## away_bb$dayWe          0.13741    0.16587   0.828  0.40742
## away_bb$time           0.02481    0.04212   0.589  0.55585
## away_bb$bb_locha      -0.05325    0.07874  -0.676  0.49880
## away_bb$streak        -0.04620    0.01871  -2.470  0.01352 *
## away_bb$prev_winTRUE   0.20691    0.11569   1.788  0.07370 .
## away_bb$win_prop_diff  4.77410    0.25578  18.665  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4828.4  on 3623  degrees of freedom
## Residual deviance: 4036.9  on 3610  degrees of freedom
##   (379 observations deleted due to missingness)
## AIC: 4064.9
##
## Number of Fisher Scoring iterations: 4
```

Trimming the variables whose p-value > 0.05 will yield a different model. These are the variables I removed, first to last:

- bb_loc
- time
- prev_win
- day

```
model_away <- glm(away_bb$result ~ away_bb$avg_age +
                    away_bb$a_win_prop +
                    away_bb$streak +
                    away_bb$win_prop_diff,
                  family=binomial("logit"))
summary(model_away)
```

```
##
## Call:
## glm(formula = away_bb$result ~ away_bb$avg_age + away_bb$a_win_prop +
##     away_bb$streak + away_bb$win_prop_diff, family = binomial("logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1952  -0.8833  -0.5282   0.9959   2.5773
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.41406    0.61676   0.671  0.50200
## away_bb$avg_age       -0.05437    0.02480  -2.193  0.02833 *
## away_bb$a_win_prop     1.05425    0.35740   2.950  0.00318 **
## away_bb$streak        -0.02858    0.01246  -2.295  0.02176 *
## away_bb$win_prop_diff  4.78583    0.24211  19.767  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5328.0  on 4002  degrees of freedom
## Residual deviance: 4468.4  on 3998  degrees of freedom
## AIC: 4478.4
##
## Number of Fisher Scoring iterations: 4
```

The final equation to model the probability of a road team winning a game of a back to back is:

$$log(\frac{p_i}{1-p_i}) = 0.414 - 0.054 \times avg\_age + 1.054 \times a\_win\_prop - 0.029 \times streak + 4.786 \times win\_prop\_diff$$

There are two key conditions for fitting a logistic regression model. First, each predictor is linearly related to logit(p_i) if all other predictors are held constant. The second condition is that each outcome is independent of all other outcomes.

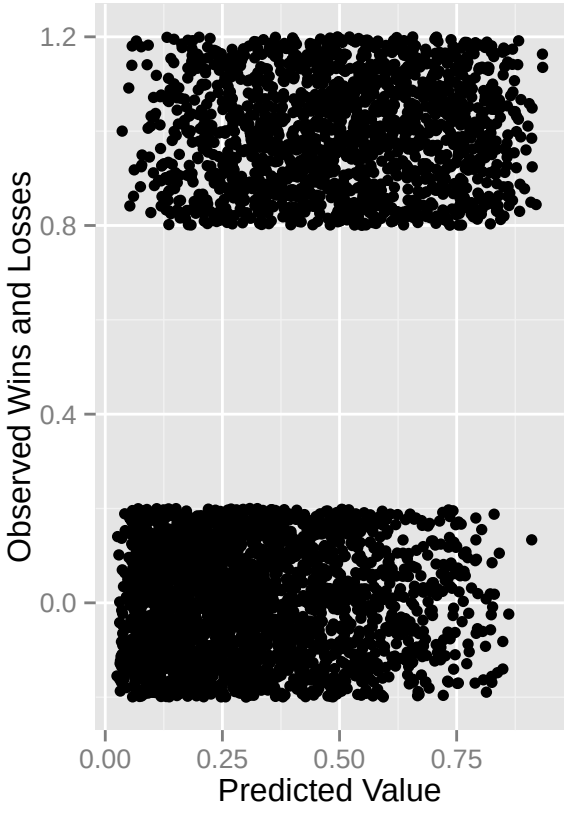
Checking the first condition:

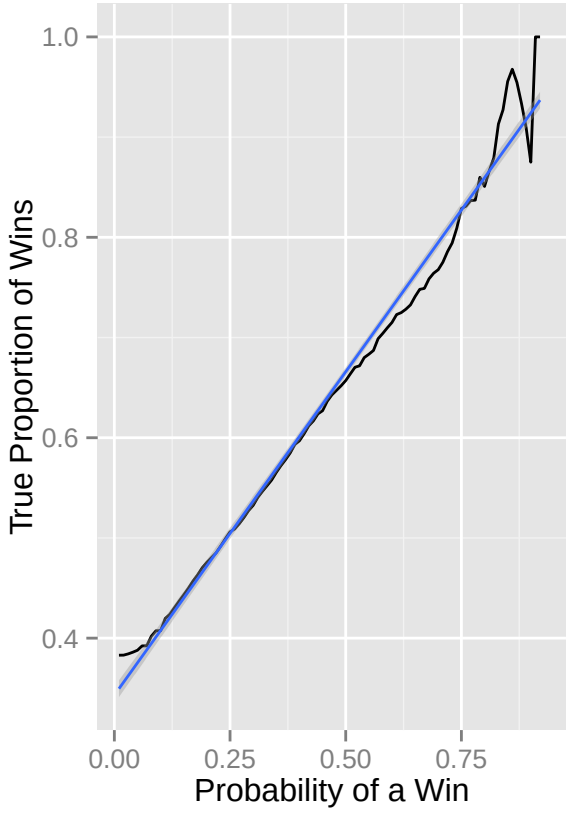
```
p <- ggplot(model_away, aes(model_away$fitted.values, jitter(model_away$y)))
p <- p + xlab("Predicted Value") + ylab("Observed Wins and Losses") + geom_point()
```

I am not sure how to correctly use natural splines.

```
check <- data.frame("true_outcome"=model_away$y,
                    "probability"=round(model_away$fitted.values,2))
#find the number of wins above a given probability
w <- function(p){
  r <- nrow(subset(check, check$probability > p & check$true_outcome == 1))
  return(r)
}
#find the number of wins above a given probability
l <- function(p){
  r <- nrow(subset(check, check$probability > p & check$true_outcome == 0))
  return(r)
}

my_df <- data.frame("prob_greater_than"=seq(.01,max(model_away$fitted.values),.01))
my_df$wins <- unlist(lapply(my_df$prob_greater_than, w))
my_df$losses <- unlist(lapply(my_df$prob_greater_than, l))
my_df <- my_df %>% mutate(prop=wins/(wins+losses))
p1 <- ggplot(my_df, aes(prob_greater_than, prop))
p1 <- p1 + xlab("Probability of a Win") + ylab("True Proportion of Wins") + geom_line()
p1 <- p1 + geom_smooth(method="glm")
grid.arrange(p, p1, ncol=2)
```

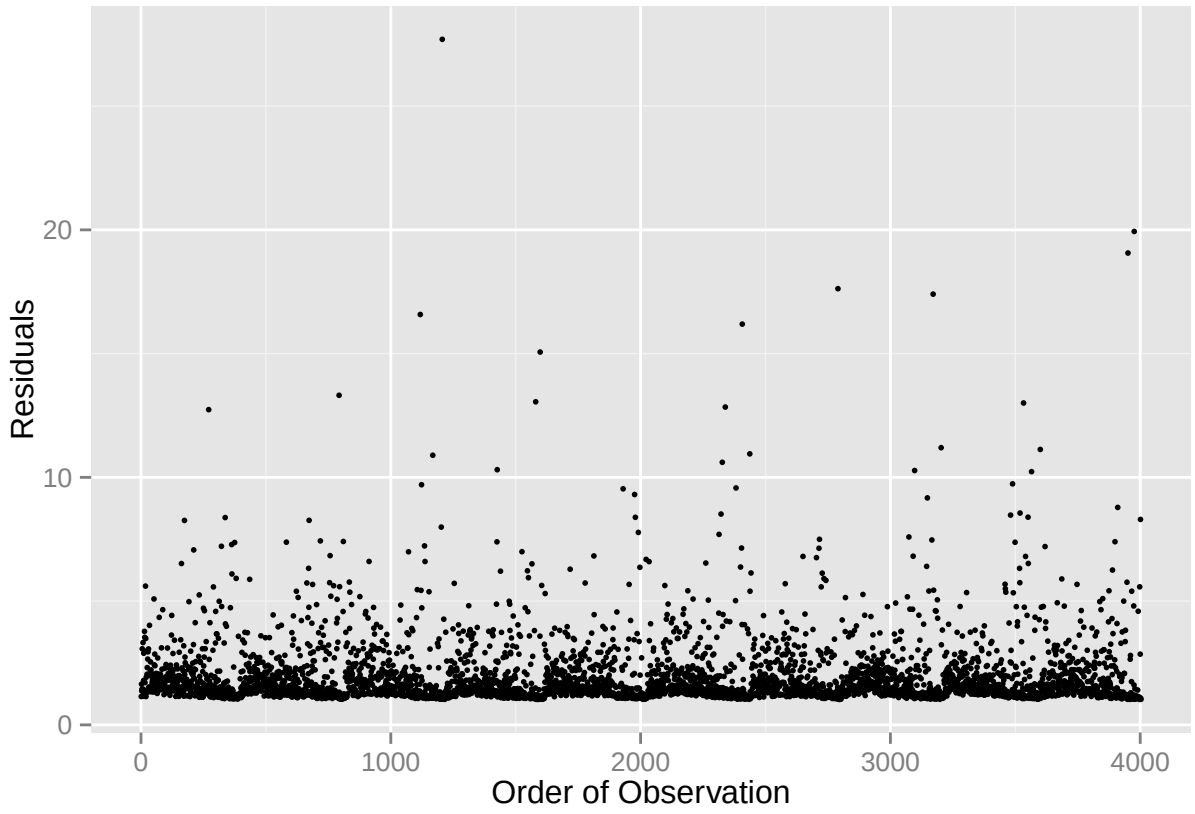




The variability to the right is due to the lack of observations with predicted values > 0.9.

Checking the second condition:

```
x <- c(1:length(model_away$residuals))
p2 <- ggplot(model_away, aes(1:length(model_away$residuals),abs(model_away$residuals)))
p2 <- p2 + xlab("Order of Observation") +
  ylab("Residuals")+ geom_point(size=1)
p2
```



Plotting the residuals in the order that their observations were collected helps identify connections between successive observations. The wave like pattern is present because the model uses the end of year records and not current records. As the season progresses the model becomes more efficient.

Model Inadequacies

My model is lacking in a myriad of different ways. The most glaring error is that using the end of season records is inappropriate. Going into any game of the season it is impossible to know a team's final winning percentage. There are other factors that should be taken into account when determining the probability of an NBA team winning a game, many more than the ones I use.

Warriors @ Nets

I have selfish reasons for doing this project. This Sunday, December 6th, I have Nets tickets. They are playing the Golden State Warriors. The Warriors this season have not lost (I am writing this on 12/2 - there are 2 more games for the Warriors between now and Sunday). The Nets have not played very well. It is the 2nd

game of a back to back on the road for the Warriors. If we apply the model, let's see what the probability of a Warriors win is:

```
w_avg_age <- 27.3
#if they beat the hornets and raptors
w_a_win_prop <- 1
w_streak <- 21
w_win_prop_diff <- 1-(6/13) #if they beat the knicks 12/4
p <- model_away$coefficients[1] +
                   model_away$coefficients[2]*w_avg_age +
                   model_away$coefficients[3]*w_a_win_prop +
                   model_away$coefficients[4]*w_streak +
                   model_away$coefficients[5]*w_win_prop_diff
w_win <- exp(p)/(1+exp(p))
```

```
away_bb$point_diff <- as.numeric(as.character(away_bb$tm_points)) -
  as.numeric(as.character(away_bb$opp_points))
model_away_points <- lm(away_bb$point_diff ~ away_bb$avg_age +
                          away_bb$a_win_prop + away_bb$streak + away_bb$win_prop_diff)
summary(model_away_points)
```

```
##
## Call:
## lm(formula = away_bb$point_diff ~ away_bb$avg_age + away_bb$a_win_prop +
##     away_bb$streak + away_bb$win_prop_diff)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.212  -7.569   0.288   7.492  46.674
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.36682    3.21491   0.425  0.67075
## away_bb$avg_age         -0.28901    0.12892  -2.242  0.02503 *
## away_bb$a_win_prop       5.60149    1.87100   2.994  0.00277 **
## away_bb$streak          -0.13311    0.06289  -2.117  0.03436 *
## away_bb$win_prop_diff 26.70868    1.13937  23.442  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.73 on 3998 degrees of freedom
## Multiple R-squared:  0.2178, Adjusted R-squared:  0.217
## F-statistic: 278.3 on 4 and 3998 DF,  p-value: < 2.2e-16
```

```
w_spread <- model_away_points$coefficients[1] +
                   model_away$coefficients[2]*w_avg_age +
                   model_away$coefficients[3]*w_a_win_prop +
                   model_away$coefficients[4]*w_streak +
                   model_away$coefficients[5]*w_win_prop_diff
```

My model gives the Warriors probability of a win as `0.88`. The point differential model predicts that the Warriors will win by `2.9` points. The smart money will be on the Nets to cover.

Citations

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Duncan Temple Lang and the CRAN Team (2015). XML: Tools for Parsing and Generating XML Within R and S-Plus. R package version 3.98-1.3. http://CRAN.R-project.org/package=XML

Hadley Wickham and Romain Francois (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.3. http://CRAN.R-project.org/package=dplyr

Hadley Wickham (2015). tidyr: Easily Tidy Data with `spread()` and `gather()` Functions. R package version 0.3.1. http://CRAN.R-project.org/package=tidyr

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL http://www.jstatsoft.org/v40/i03/.

Hadley Wickham (2015). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.0.0. http://CRAN.R-project.org/package=stringr

Hadley Wickham (2015). scales: Scale Functions for Visualization. R package version 0.3.0. http://CRAN.R-project.org/package=scales H. Wickham. ggplot2: elegant graphics for data analysis. Springer New York, 2009.