Data-Cleaning

Abisai Lujan

2025-01-25

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
GameStats <- read.csv("GameStats.csv")
```

Display column names, variable types, and preview of the data

```
names(GameStats)
```

```
##  [1] "School"    "Date"      "G."        "X"         "Opponent"  "X.1"
##  [7] "PassCmp"   "PassAtt"   "PassPct"   "PassYds"   "PassTD"    "RushAtt"
## [13] "RushYds"   "RushAvg"   "RushTD"    "XPM"       "XPA"       "XPPercent"
## [19] "FGM"       "FGA"       "FGPercent" "KickPts"   "Fum"       "Int"
## [25] "TotalTO"
```

Data Dictionary

G: The nth game played overall (not just conference)

X: stats for this school

X.1: if empty, the School played at home stadium, otherwise @ means it was an away game

PassCmp: Number of passes completed

PassAtt: Total number of passes

PassPct: (PassCmp / PassAtt). Pass completion percentage

PassYds: Distance, in yards, covered from completed passes

PassTD: Number of passes that resulted in a touch down

RushAtt: Total number of times the team attempted to run the ball.

RushYds: Total yards team gained from the rushes made.

RushAvg: (RushYDs / RushAtt). Total yardage gained from all rushes divided by total number of times team attempted to rush; Average of how many yards were gained per rush attempt.

RushTD: The number of times the team rushed and successfully made a touchdown.

XPM: Extra points made after touchdowns (i.e., rushing TDs, passing TDs, etc)

XPA: Extra points attempted after touchdowns

XPPercent: (XPM/XPA) Percent of extra points after touchdowns successfully made

FGM: Number of field goals successfully made

FGA: Total number of attempted field goal shots

FGPercent: Percent of field goal shots completed

Kickpts:

Fum:

Int: Number of opponent passes that team intercepted

Observe variable types.

```
(str(GameStats, give.attr=F))
```

```
## 'data.frame':    7360 obs. of  25 variables:
##  $ School   : chr  "Georgia State" "Michigan State" "Oregon State" "SMU" ...
##  $ Date     : chr  "9/4/2015" "9/4/2015" "9/4/2015" "9/4/2015" ...
##  $ G.       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ X        : chr  "" "@" "" "" ...
##  $ Opponent : chr  "Charlotte" "Western Michigan" "Weber State" "Baylor" ...
##  $ X.1      : chr  "L" "W" "W" "L" ...
##  $ PassCmp  : int  25 15 12 16 10 20 33 22 34 29 ...
##  $ PassAtt  : int  43 31 22 24 20 35 50 36 53 42 ...
##  $ PassPct  : num  58.1 48.4 54.5 66.7 50 57.1 66 61.1 64.2 69 ...
##  $ PassYds  : int  299 256 110 166 114 150 365 229 338 265 ...
##  $ PassTD   : int  2 2 2 2 2 0 2 4 1 0 ...
##  $ RushAtt  : int  26 40 56 54 58 22 23 38 42 28 ...
##  $ RushYds  : int  93 196 281 203 312 29 18 163 229 78 ...
##  $ RushAvg  : num  3.6 4.9 5 3.8 5.4 1.3 0.8 4.3 5.5 2.8 ...
##  $ RushTD   : int  0 3 0 1 2 0 0 0 5 1 ...
##  $ XPM      : int  2 4 2 3 6 1 3 6 6 1 ...
##  $ XPA      : int  2 4 2 3 6 1 3 6 6 1 ...
##  $ XPPercent: num  100 100 100 100 100 100 100 100 100 100 ...
##  $ FGM      : int  2 1 4 0 1 2 1 0 2 2 ...
##  $ FGA      : int  3 1 4 0 2 3 2 0 2 2 ...
##  $ FGPercent: num  66.7 100 100 NA 50 66.7 50 NA 100 100 ...
##  $ KickPts  : int  8 7 14 3 9 7 6 6 12 7 ...
##  $ Fum      : int  2 1 0 0 0 0 0 1 0 0 ...
##  $ Int      : int  1 0 1 2 1 1 2 0 1 1 ...
##  $ TotalTO  : int  3 1 1 2 1 1 2 1 1 1 ...

## NULL
```

```r
head(GameStats)
```

```
##           School     Date G. X         Opponent X.1 PassCmp PassAtt PassPct
## 1  Georgia State 9/4/2015  1          Charlotte   L      25      43    58.1
## 2 Michigan State 9/4/2015  1 @ Western Michigan   W      15      31    48.4
## 3   Oregon State 9/4/2015  1        Weber State   W      12      22    54.5
## 4            SMU 9/4/2015  1            Baylor   L      16      24    66.7
## 5       Syracuse 9/4/2015  1       Rhode Island   W      10      20    50.0
## 6     Washington 9/4/2015  1 @      Boise State   L      20      35    57.1
##   PassYds PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA
## 1     299      2      26      93     3.6      0   2   2       100   2   3
## 2     256      2      40     196     4.9      3   4   4       100   1   1
## 3     110      2      56     281     5.0      0   2   2       100   4   4
## 4     166      2      54     203     3.8      1   3   3       100   0   0
## 5     114      2      58     312     5.4      2   6   6       100   1   2
## 6     150      0      22      29     1.3      0   1   1       100   2   3
##   FGPercent KickPts Fum Int TotalTO
## 1      66.7       8   2   1       3
## 2     100.0       7   1   0       1
## 3     100.0      14   0   1       1
## 4        NA       3   0   2       2
## 5      50.0       9   0   1       1
## 6      66.7       7   0   1       1
```

Convert Date into numeric type

```r
#Change date format
GameStats$Date=as.Date(GameStats$Date, format="%m/%d/%Y")
#Numeric format by removing hyphen separators
GameStats$Date=as.numeric(gsub("-","",GameStats$Date))
head(GameStats)
```

```
##           School     Date G. X         Opponent X.1 PassCmp PassAtt PassPct
## 1  Georgia State 20150904  1          Charlotte   L      25      43    58.1
## 2 Michigan State 20150904  1 @ Western Michigan   W      15      31    48.4
## 3   Oregon State 20150904  1        Weber State   W      12      22    54.5
## 4            SMU 20150904  1            Baylor   L      16      24    66.7
## 5       Syracuse 20150904  1       Rhode Island   W      10      20    50.0
## 6     Washington 20150904  1 @      Boise State   L      20      35    57.1
##   PassYds PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA
## 1     299      2      26      93     3.6      0   2   2       100   2   3
## 2     256      2      40     196     4.9      3   4   4       100   1   1
## 3     110      2      56     281     5.0      0   2   2       100   4   4
## 4     166      2      54     203     3.8      1   3   3       100   0   0
## 5     114      2      58     312     5.4      2   6   6       100   1   2
## 6     150      0      22      29     1.3      0   1   1       100   2   3
##   FGPercent KickPts Fum Int TotalTO
## 1      66.7       8   2   1       3
## 2     100.0       7   1   0       1
## 3     100.0      14   0   1       1
## 4        NA       3   0   2       2
## 5      50.0       9   0   1       1
## 6      66.7       7   0   1       1
```

Date range of the games

```
paste("Start:", min(GameStats$Date), ", End:", max(GameStats$Date))
```

```
## [1] "Start: 20150903 , End: 20191017"
```

Summary

```
#lets take a look at the summary and distribution of the variables
summary(GameStats)
```

```
##     School               Date                G.              X
##  Length:7360        Min.   :20150903   Min.   : 1.000   Length:7360
##  Class :character   1st Qu.:20160910   1st Qu.: 3.000   Class :character
##  Mode  :character   Median :20170923   Median : 6.000   Mode  :character
##                     Mean   :20168804   Mean   : 6.536
##                     3rd Qu.:20180929   3rd Qu.:10.000
##                     Max.   :20191017   Max.   :15.000
##
##    Opponent             X.1               PassCmp         PassAtt
##  Length:7360        Length:7360        Min.   : 0.00   Min.   : 0.00
##  Class :character   Class :character   1st Qu.:14.00   1st Qu.:25.00
##  Mode  :character   Mode  :character   Median :18.00   Median :31.00
##                                        Mean   :18.72   Mean   :31.47
##                                        3rd Qu.:23.00   3rd Qu.:38.00
##                                        Max.   :58.00   Max.   :88.00
##
##     PassPct          PassYds          PassTD         RushAtt
##  Min.   :  0.00   Min.   :  0.0   Min.   :0.000   Min.   : 8.00
##  1st Qu.: 51.50   1st Qu.:161.0   1st Qu.:1.000   1st Qu.:32.00
##  Median : 59.50   Median :225.0   Median :1.000   Median :38.00
##  Mean   : 59.07   Mean   :232.4   Mean   :1.687   Mean   :39.21
##  3rd Qu.: 66.70   3rd Qu.:296.0   3rd Qu.:3.000   3rd Qu.:46.00
##  Max.   :100.00   Max.   :734.0   Max.   :9.000   Max.   :89.00
##  NA's   :1
##     RushYds          RushAvg          RushTD           XPM
##  Min.   :-73.0   Min.   :-2.900   Min.   :0.000   Min.   : 0.00
##  1st Qu.:108.0   1st Qu.: 3.100   1st Qu.:1.000   1st Qu.: 2.00
##  Median :165.0   Median : 4.250   Median :2.000   Median : 3.00
##  Mean   :177.5   Mean   : 4.364   Mean   :1.819   Mean   : 3.45
##  3rd Qu.:234.0   3rd Qu.: 5.500   3rd Qu.:3.000   3rd Qu.: 5.00
##  Max.   :651.0   Max.   :13.700   Max.   :9.000   Max.   :11.00
##
##       XPA           XPPercent          FGM             FGA
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.000   Min.   :0.000
##  1st Qu.: 2.000   1st Qu.:100.00   1st Qu.:0.000   1st Qu.:1.000
##  Median : 3.000   Median :100.00   Median :1.000   Median :1.000
##  Mean   : 3.563   Mean   : 96.46   Mean   :1.133   Mean   :1.535
##  3rd Qu.: 5.000   3rd Qu.:100.00   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :11.000   Max.   :100.00   Max.   :7.000   Max.   :7.000
##                   NA's   :388
##    FGPercent         KickPts            Fum              Int
##  Min.   :  0.00   Min.   : 0.000   Min.   :0.0000   Min.   :0.0000
```

```
##  1st Qu.: 50.00   1st Qu.: 4.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :100.00   Median : 7.000   Median :0.0000   Median :1.0000
##  Mean   : 73.18   Mean   : 6.849   Mean   :0.6709   Mean   :0.8541
##  3rd Qu.:100.00   3rd Qu.: 9.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :100.00   Max.   :28.000   Max.   :5.0000   Max.   :6.0000
##  NA's   :1478
##     TotalTO
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.525
##  3rd Qu.:2.000
##  Max.   :8.000
##
```

It is important to note that there are games with incomplete pairs of stats for the home and away team. That is, a game lacks stats for either team. It is also important to note the date of the game, as the choice of home stadium can alter between different games of the same pairing.

Therefore, any game on the same date must have a row for both teams (i.e., there will be two rows where each team will be listed under `School`)

Lets observe a complete pairing. Georgia State played (at its home stadium) against Charlotte on 9/4/2015. We can see Georgia's performance stats below:

```
#Select all rows where the School is Georgia State, the Opponent is Charlotte, and the date is 9/4/15
home_gs_v_ch <- GameStats[GameStats$School=="Georgia State"
                          & GameStats$Opponent=="Charlotte"
                          & GameStats$Date==20150904,]
home_gs_v_ch
```

```
##          School     Date G. X  Opponent X.1 PassCmp PassAtt PassPct PassYds
## 1 Georgia State 20150904  1   Charlotte   L      25      43    58.1     299
##   PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA FGPercent
## 1      2      26      93     3.6      0   2   2       100   2   3      66.7
##   KickPts Fum Int TotalTO
## 1       8   2   1       3
```

We can see Charlotte's (the away team) performance stats for the same game (9/4/2015) below:

```
away_ch_v_gs <- GameStats[GameStats$School=="Charlotte"
                          & GameStats$Opponent=="Georgia State"
                          & GameStats$Date==20150904,]
away_ch_v_gs
```

```
##        School     Date G. X      Opponent X.1 PassCmp PassAtt PassPct PassYds
## 135 Charlotte 20150904  1 @ Georgia State   W      19      32    59.4     244
##     PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA FGPercent
## 135      1      54     164       3      0   2   2       100   3   3       100
##     KickPts Fum Int TotalTO
## 135      11   2   2       4
```

We may also see that the away (@) symbol is correctly present in the 4th column of Charlotte's performance stats.

Lets observe a couple of cases where a team's stats are incomplete for a game:

```
smu_stats_v_b <- GameStats[GameStats$School=="SMU"
                           & GameStats$Opponent=="Baylor"
                           & GameStats$Date==20150904,]
smu_stats_v_b
```

```
##   School     Date G. X Opponent X.1 PassCmp PassAtt PassPct PassYds PassTD
## 4    SMU 20150904  1     Baylor   L      16      24    66.7     166      2
##   RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA FGPercent KickPts
## 4      54     203     3.8      1   3   3       100   0   0        NA       3
##   Fum Int TotalTO
## 4   0   2       2
```

```
bay_stats_v_smu <- GameStats[GameStats$School=="Baylor"
                             & GameStats$Opponent=="SMU"
                             & GameStats$Date==20150904,]
bay_stats_v_smu
```

```
##  [1] School    Date      G.        X         Opponent  X.1       PassCmp
##  [8] PassAtt   PassPct   PassYds   PassTD    RushAtt   RushYds   RushAvg
## [15] RushTD    XPM       XPA       XPPercent FGM       FGA       FGPercent
## [22] KickPts   Fum       Int       TotalTO
## <0 rows> (or 0-length row.names)
```

Here we can see that SMU's opponent Baylor does not have any stats for their game on 9/4/2015.

Baylor does have performance stats for games with other teams, we can see all those below:

```
#Select all of Baylor's performance stats for any games they played on any date
bay_stats <- GameStats[GameStats$School=="Baylor",]
head(bay_stats)
```

```
##     School     Date G. X           Opponent X.1 PassCmp PassAtt PassPct PassYds
## 133 Baylor 20150904  1 @ Southern Methodist   W      17      32    53.1     423
## 331 Baylor 20150912  2               Lamar   W      26      38    68.4     373
## 498 Baylor 20150926  3                Rice   W      19      25    76.0     366
## 581 Baylor 20151003  4 N         Texas Tech   W      16      24    66.7     312
## 668 Baylor 20151010  5 @            Kansas   W      27      37    73.0     363
## 754 Baylor 20151017  6        West Virginia   W      21      34    61.8     389
##     PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA FGPercent
## 133      6      37     300     8.1      2   8   8       100   0   0        NA
## 331      4      56     412     7.4      5   7   7       100   1   1       100
## 498      7      53     427     8.1      3  10  10       100   0   0        NA
## 581      4      52     368     7.1      5   9   9       100   0   0        NA
## 668      5      45     281     6.2      3   9   9       100   1   2        50
## 754      6      50     304     6.1      2   8   8       100   2   2       100
##     KickPts Fum Int TotalTO
## 133       8   0   1       1
## 331      10   1   3       4
## 498      10   1   0       1
## 581       9   0   1       1
## 668      12   1   0       1
## 754      14   0   0       0
```

Baylor is present in both the `School` and `Opponent` lists.

Therefore, to identify complete pairs, both teams should have their own row, where they are listed as `School` and `Opponent`, and have the same date.

First, lets assure that a team is present in both the GameStats `School` and `Opponent` lists.

```
#List of all teams in the School column
schs <- unique(GameStats$School)
schs[1:10]
```

```
##  [1] "Georgia State"    "Michigan State"   "Oregon State"     "SMU"
##  [5] "Syracuse"         "Washington"       "Western Michigan" "Arizona"
##  [9] "Ball State"       "Central Michigan"
```

```
#List of all teams in the Opponent column
opps <- unique(GameStats$Opponent)
opps[1:10]
```

```
##  [1] "Charlotte"                   "Western Michigan"
##  [3] "Weber State"                 "Baylor"
##  [5] "Rhode Island"                "Boise State"
##  [7] "Michigan State"              "Texas-San Antonio"
##  [9] "Virginia Military Institute" "Oklahoma State"
```

Take the intersection of the two lists

```
fully_present_teams <- intersect(schs, opps)
fully_present_teams[1:10]
```

```
##  [1] "Georgia State"    "Michigan State"   "Oregon State"     "Syracuse"
##  [5] "Washington"       "Western Michigan" "Arizona"          "Ball State"
##  [9] "Central Michigan" "Colorado"
```

Lets test if schools we know to be fully present are in this new list. For example Baylor, Georgia State, and Charlotte

```
c("Baylor", "Georgia State", "Charlotte") %in% fully_present_teams
```

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

```
#Also check if schools we know to not be fully present are in this list
c("SMU", "Pitt") %in% fully_present_teams
```

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

Now it appears we have all teams are fully present and valid, so I will filter their respective rows into a new data set.

```
valid_GS <- GameStats %>%
  filter(School %in% fully_present_teams
         & Opponent %in% fully_present_teams)
head(valid_GS)
```

```
##             School     Date G. X         Opponent X.1 PassCmp PassAtt PassPct
## 1    Georgia State 20150904  1          Charlotte   L      25      43    58.1
## 2   Michigan State 20150904  1 @ Western Michigan   W      15      31    48.4
## 3       Washington 20150904  1 @      Boise State   L      20      35    57.1
## 4 Western Michigan 20150904  1     Michigan State   L      33      50    66.0
## 5 Central Michigan 20150903  1     Oklahoma State   L      29      42    69.0
## 6         Colorado 20150903  1 @           Hawaii   L      23      40    57.5
##   PassYds PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA
## 1     299      2      26      93     3.6      0   2   2       100   2   3
## 2     256      2      40     196     4.9      3   4   4       100   1   1
## 3     150      0      22      29     1.3      0   1   1       100   2   3
## 4     365      2      23      18     0.8      0   3   3       100   1   2
## 5     265      0      28      78     2.8      1   1   1       100   2   2
## 6     156      0      53     215     4.1      2   2   2       100   2   2
##   FGPercent KickPts Fum Int TotalTO
## 1      66.7       8   2   1       3
## 2     100.0       7   1   0       1
## 3      66.7       7   0   1       1
## 4      50.0       6   0   2       2
## 5     100.0       7   0   1       1
## 6     100.0       8   2   1       3
```

Lets see which teams in the original `School` list are not in the valid data set's `School` list

```
#Which elements of x are not in y
setdiff(GameStats$School, valid_GS$School)
```

```
##  [1] "SMU"      "UCF"      "UTSA"     "Ole Miss" "Pitt"     "UNLV"
##  [7] "USC"      "UTEP"     "LSU"      "UAB"
```

How much have we reduced the orginal data set of game stats?

```
(1- (nrow(valid_GS) / nrow(GameStats))) *100
```

```
## [1] 20.02717
```

We have removed ~20% of the games from the original data set but considering the sample size and the sake of correct pairing, it should be permissible.

Merge each game's stats into 1 rows First I believe it would be efficient to split the valid game stats into two sets: home teams and away teams.

Home teams are those who do not have the @ X-value

```
home_teams <- valid_GS %>%
  filter(X != "@")
head(home_teams)
```

```
##             School     Date G. X        Opponent X.1 PassCmp PassAtt PassPct
## 1    Georgia State 20150904  1         Charlotte   L      25      43    58.1
## 2 Western Michigan 20150904  1    Michigan State   L      33      50    66.0
## 3 Central Michigan 20150903  1    Oklahoma State   L      29      42    69.0
## 4           Hawaii 20150903  1          Colorado   W      19      38    50.0
## 5            Idaho 20150903  1              Ohio   L      36      48    75.0
## 6        Minnesota 20150903  1   Texas Christian   L      19      35    54.3
##   PassYds PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA
## 1     299      2      26      93     3.6      0   2   2       100   2   3
## 2     365      2      23      18     0.8      0   3   3       100   1   2
## 3     265      0      28      78     2.8      1   1   1       100   2   2
## 4     202      3      34     100     2.9      0   2   2       100   2   2
## 5     297      1      28     100     3.6      2   2   2       100   2   2
## 6     197      1      39     144     3.7      1   2   2       100   1   1
##   FGPercent KickPts Fum Int TotalTO
## 1      66.7       8   2   1       3
## 2      50.0       6   0   2       2
## 3     100.0       7   0   1       1
## 4     100.0       8   0   2       2
## 5     100.0       8   1   2       3
## 6     100.0       5   2   0       2
```

Away teams are those who do not have the " " X-value

```
away_teams <- valid_GS %>%
  filter(X != "")
head(away_teams)
```

```
##           School     Date G. X         Opponent X.1 PassCmp PassAtt PassPct
## 1 Michigan State 20150904  1 @ Western Michigan   W      15      31    48.4
## 2     Washington 20150904  1 @      Boise State   L      20      35    57.1
## 3       Colorado 20150903  1 @           Hawaii   L      23      40    57.5
## 4           Duke 20150903  1 @           Tulane   W      29      44    65.9
## 5       Michigan 20150903  1 @             Utah   L      27      43    62.8
## 6 North Carolina 20150903  1 N   South Carolina   L      19      31    61.3
##   PassYds PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA
## 1     256      2      40     196     4.9      3   4   4       100   1   1
## 2     150      0      22      29     1.3      0   1   1       100   2   3
## 3     156      0      53     215     4.1      2   2   2       100   2   2
## 4     324      2      49     206     4.2      1   4   4       100   3   3
## 5     279      2      29      76     2.6      0   2   2       100   1   2
## 6     232      1      32     208     6.5      0   1   1       100   2   2
##   FGPercent KickPts Fum Int TotalTO
## 1     100.0       7   1   0       1
## 2      66.7       7   0   1       1
## 3     100.0       8   2   1       3
## 4     100.0      13   2   0       2
## 5      50.0       5   0   3       3
## 6     100.0       7   0   3       3
```

There are 8 more rows in home_teams than in away_teams. After manually searching, the extra rows' indicies are (1468,2733,2186, 225, 2881,759, 12, 2951): (Baylor vs Liberty), (Florida vs Idaho), (fresno state vs idaho), (Georgia State vs Liberty), (Penn State vs Idaho), (Virginia Tech vs Liberty), (West Virginia vs Liberty), (Wyoming vs Idaho)

```
home_teams[c(1468,2733,2186, 225, 2881,759, 12, 2951),1:6]
```

```
##             School     Date G. X Opponent X.1
## 1468        Baylor 20170902  1    Liberty   L
## 2733       Florida 20181117 11      Idaho   W
## 2186  Fresno State 20180901  1      Idaho   W
## 225  Georgia State 20151003  4    Liberty   L
## 2881    Penn State 20190831  1      Idaho   W
## 759  Virginia Tech 20160903  1    Liberty   W
## 12   West Virginia 20150912  2    Liberty   W
## 2951       Wyoming 20190914  3      Idaho   W
```

Lets remove these extra rows to prepare for merging

```
home_teams2 <- home_teams[-c(1468,2733,2186, 225, 2881,759, 12, 2951),]
```

```
head(home_teams2)[, 1:6]
```

```
##             School     Date G. X        Opponent X.1
## 1    Georgia State 20150904  1         Charlotte   L
## 2 Western Michigan 20150904  1    Michigan State   L
## 3 Central Michigan 20150903  1    Oklahoma State   L
## 4           Hawaii 20150903  1          Colorado   W
## 5            Idaho 20150903  1              Ohio   L
## 6        Minnesota 20150903  1   Texas Christian   L
```

Next I aim to reorder the home and away team sets to pair them correctly. For home teams I will reorder the rows by X, School, then Date. This way we will see the home teams in alphabetical order, their opponent (away) team, and the dates they played in order.

```
home_reorder <- home_teams2[order(home_teams2$X, home_teams2$School, home_teams2$Date),]
head(home_reorder)[,1:6]
```

```
##        School     Date G. X       Opponent X.1
## 111 Air Force 20150912  2   San Jose State   W
## 253 Air Force 20151010  5          Wyoming   W
## 415 Air Force 20151024  7     Fresno State   W
## 490 Air Force 20151107  9             Army   W
## 521 Air Force 20151114 10       Utah State   W
## 830 Air Force 20160910  2    Georgia State   W
```

For away teams, I will order by X, the opponent (home) teams, then the date of the games.

```
away_reorder <- away_teams[order(away_teams$X, away_teams$Opponent, away_teams$Date),]
head(away_reorder)[,1:6]
```

```
##             School     Date G. X  Opponent X.1
## 67  San Jose State 20150912  2 @ Air Force   L
## 207        Wyoming 20151010  6 @ Air Force   L
## 339   Fresno State 20151024  8 @ Air Force   L
## 493           Army 20151107  9 @ Air Force   L
## 479     Utah State 20151114 10 @ Air Force   L
## 771  Georgia State 20160910  2 @ Air Force   L
```

Lets see how the rows line up

```
nrow(home_reorder)-nrow(away_reorder)
```

```
## [1] 0
```

```
nrow(home_reorder) - sum(home_reorder$Date == away_reorder$Date)
```

```
## [1] 0
```

All rows line up and are of equal length

Now lets merge the data sets

```
GameStats_merged <- cbind(home_reorder, away_reorder)
head(GameStats_merged)[,1:6]
```

```
##        School     Date G. X       Opponent X.1
## 111 Air Force 20150912  2   San Jose State   W
## 253 Air Force 20151010  5          Wyoming   W
## 415 Air Force 20151024  7     Fresno State   W
## 490 Air Force 20151107  9             Army   W
## 521 Air Force 20151114 10       Utah State   W
## 830 Air Force 20160910  2    Georgia State   W
```

Removing neutral field games Rows with X="N" represent neutral field games, which I will remove from the merged data set.

```
#First rename the X columns to X.Home and X.Away respectively
colnames(GameStats_merged)[4] = "X.Home"
colnames(GameStats_merged)[29] = "X.Away"
(colnames(GameStats_merged))
```

```
##  [1] "School"    "Date"      "G."        "X.Home"    "Opponent"  "X.1"
##  [7] "PassCmp"   "PassAtt"   "PassPct"   "PassYds"   "PassTD"    "RushAtt"
## [13] "RushYds"   "RushAvg"   "RushTD"    "XPM"       "XPA"       "XPPercent"
## [19] "FGM"       "FGA"       "FGPercent" "KickPts"   "Fum"       "Int"
## [25] "TotalTO"   "School"    "Date"      "G."        "X.Away"    "Opponent"
## [31] "X.1"       "PassCmp"   "PassAtt"   "PassPct"   "PassYds"   "PassTD"
## [37] "RushAtt"   "RushYds"   "RushAvg"   "RushTD"    "XPM"       "XPA"
## [43] "XPPercent" "FGM"       "FGA"       "FGPercent" "KickPts"   "Fum"
## [49] "Int"       "TotalTO"
```

```
#Determine which home team rows played on neutral fields
head(GameStats_merged[GameStats_merged$X.Home=="N",])[1:5]
```

```
##         School     Date G. X.Home      Opponent
## 663  Air Force 20151229 14      N    California
## 1011 Air Force 20161015  6      N    New Mexico
## 1371 Air Force 20161230 13      N South Alabama
## 691      Akron 20151222 13      N    Utah State
## 2143     Akron 20171202 13      N        Toledo
## 20     Alabama 20150905  1      N     Wisconsin
```

```r
head(which(GameStats_merged$X.Home=="N"))
```

```
## [1] 2706 2707 2708 2709 2710 2711
```

```r
#Determine which Away team rows played on neutral fields
head(GameStats_merged[GameStats_merged$X.Away=="N",])[1:5]
```

```
##         School     Date G. X.Home      Opponent
## 663  Air Force 20151229 14      N    California
## 1011 Air Force 20161015  6      N    New Mexico
## 1371 Air Force 20161230 13      N South Alabama
## 691      Akron 20151222 13      N    Utah State
## 2143     Akron 20171202 13      N        Toledo
## 20     Alabama 20150905  1      N     Wisconsin
```

```r
head(which(GameStats_merged$X.Away == "N"))
```

```
## [1] 2706 2707 2708 2709 2710 2711
```

```r
#Determine if the indices for the home team and away team rows on neutral fields
#match up
setdiff(which(GameStats_merged$X.Away=="N"),which(GameStats_merged$X.Home == "N"))
```

```
## integer(0)
```

```r
#Filter out games played on neutral fields
GSM_no_nf <- GameStats_merged[-which(GameStats_merged$X.Away == "N"),]
head(GSM_no_nf)
```

```
##        School     Date G. X.Home       Opponent X.1 PassCmp PassAtt PassPct
## 111 Air Force 20150912  2        San Jose State   W       3      11    27.3
## 253 Air Force 20151010  5              Wyoming   W       5      10    50.0
## 415 Air Force 20151024  7          Fresno State   W       6      11    54.5
## 490 Air Force 20151107  9                 Army   W       7      10    70.0
## 521 Air Force 20151114 10            Utah State   W      11      17    64.7
## 830 Air Force 20160910  2         Georgia State   W       3       9    33.3
##     PassYds PassTD RushAtt RushYds RushAvg RushTD XPM XPA XPPercent FGM FGA
## 111      24      0      69     428     6.2      5   4   5        80   1   1
## 253      80      1      58     299     5.2      3   4   4       100   1   2
## 415     128      1      79     458     5.8      5   6   6       100   0   1
## 490     156      2      47     196     4.2      0   2   2       100   2   2
## 521     271      1      64     309     4.8      4   5   5       100   0   0
## 830      67      0      83     464     5.6      5   6   6       100   2   2
##     FGPercent KickPts Fum Int TotalTO         School     Date G. X.Away
## 111       100       7   0   1       1 San Jose State 20150912  2      @
## 253        50       7   1   1       2        Wyoming 20151010  6      @
## 415         0       6   1   0       1   Fresno State 20151024  8      @
## 490       100       8   0   0       0           Army 20151107  9      @
## 521        NA       5   0   0       0     Utah State 20151114 10      @
## 830       100      12   0   0       0  Georgia State 20160910  2      @
```

```
##      Opponent X.1 PassCmp PassAtt PassPct PassYds PassTD RushAtt RushYds
## 111 Air Force   L      18      33    54.5     140      1      20     150
## 253 Air Force   L      15      29    51.7     192      2      35     115
## 415 Air Force   L      14      39    35.9     177      0      17     134
## 490 Air Force   L       2       8    25.0      45      0      44     124
## 521 Air Force   L      25      47    53.2     364      4      23      75
## 830 Air Force   L       9      27    33.3     142      1      14      27
##     RushAvg RushTD XPM XPA XPPercent FGM FGA FGPercent KickPts Fum Int TotalTO
## 111     7.5      1   1   2        50   1   2        50       4   0   2       2
## 253     3.3      0   2   2       100   1   1       100       5   2   2       4
## 415     7.9      2   2   2       100   0   0        NA       2   0   1       1
## 490     2.8      0   0   0        NA   1   1       100       3   0   0       0
## 521     3.3      0   4   4       100   0   0        NA       4   0   1       1
## 830     1.9      1   2   2       100   0   1         0       2   0   0       0
```

Change variables First three variable names: Date, Home, Away.

```
#I have to rename duplicate column names before I can select/modify new data set
names(GSM_no_nf)[1] = "Home"
names(GSM_no_nf)[26] = "Away"
head(GSM_no_nf)[1:5]
```

```
##          Home     Date G. X.Home       Opponent
## 111 Air Force 20150912  2        San Jose State
## 253 Air Force 20151010  5               Wyoming
## 415 Air Force 20151024  7          Fresno State
## 490 Air Force 20151107  9                 Army
## 521 Air Force 20151114 10            Utah State
## 830 Air Force 20160910  2         Georgia State
```

```
GSM_3vars <- GSM_no_nf[c(2, 1, 26)]
head(GSM_3vars)
```

```
##         Date      Home           Away
## 111 20150912 Air Force San Jose State
## 253 20151010 Air Force        Wyoming
## 415 20151024 Air Force   Fresno State
## 490 20151107 Air Force           Army
## 521 20151114 Air Force     Utah State
## 830 20160910 Air Force  Georgia State
```

`HomeWins` variable

I will make a loop to convert Home team's X.1 into binary 1,0

```
home_outcomes <- GSM_no_nf[6]$X.1

for(i in 1:length(home_outcomes)) {
  if(home_outcomes[i] == "W") {
    home_outcomes[i] = 1
  }
  else if(home_outcomes[i] == "L") {
```

```
    home_outcomes[i] = 0
  }
}
```

```
#Bind the column of binary outcomes to the right side of the new data set
GSM4 <- cbind(GSM_3vars, HomeWins = home_outcomes)
GSM4[1:8,]
```

```
##          Date      Home           Away HomeWins
## 111  20150912 Air Force San Jose State        1
## 253  20151010 Air Force        Wyoming        1
## 415  20151024 Air Force   Fresno State        1
## 490  20151107 Air Force           Army        1
## 521  20151114 Air Force     Utah State        1
## 830  20160910 Air Force  Georgia State        1
## 921  20161001 Air Force           Navy        1
## 1052 20161022 Air Force         Hawaii        0
```

Rename team statistics appropriately

```
#Column binding the appropriate columns with the appropriate names
GSM5 <- cbind(GSM4,
              HPassCmp = GSM_no_nf[,7],
              APassCmp = GSM_no_nf[,32],

              HPassAtt = GSM_no_nf[,8],
              APassAtt = GSM_no_nf[,33],

              HPassPct = GSM_no_nf[,9],
              APassPct = GSM_no_nf[,34],

              HPassYds = GSM_no_nf[,10],
              APassYds = GSM_no_nf[,35],

              HPassTD = GSM_no_nf[,11],
              APassTD = GSM_no_nf[,36],

              HRushAtt = GSM_no_nf[,12],
              ARushAtt = GSM_no_nf[,37],

              HRushYds = GSM_no_nf[,13],
              ARushYds = GSM_no_nf[,38],

              HRushAvg = GSM_no_nf[,14],
              ARushAvg = GSM_no_nf[,39],

              HRushTD = GSM_no_nf[,15],
              ARushTD = GSM_no_nf[,40],

              HXPM = GSM_no_nf[,16],
              AXPM = GSM_no_nf[,41],

              HXPA = GSM_no_nf[,17],
```

```
              AXPA = GSM_no_nf[,42],

              HXPPercent = GSM_no_nf[,18],
              AXPPercent = GSM_no_nf[,43],

              HFGM = GSM_no_nf[,19],
              AFGM = GSM_no_nf[,44],

              HFGA = GSM_no_nf[,20],
              AFGA = GSM_no_nf[,45],

              HFGPercent = GSM_no_nf[,21],
              AFGPercent = GSM_no_nf[,46],

              HKickPts = GSM_no_nf[,22],
              AKickPts = GSM_no_nf[,47],

              HFum =  GSM_no_nf[,23],
              AFum =  GSM_no_nf[,48],

              HInt =  GSM_no_nf[,24],
              AInt =  GSM_no_nf[,49],

              HTotalTO = GSM_no_nf[,25],
              ATotalTO = GSM_no_nf[,50]
              )
```

```
head(GSM5)[1:8]
```

```
##         Date      Home           Away HomeWins HPassCmp APassCmp HPassAtt
## 111 20150912 Air Force San Jose State        1        3       18       11
## 253 20151010 Air Force        Wyoming        1        5       15       10
## 415 20151024 Air Force   Fresno State        1        6       14       11
## 490 20151107 Air Force           Army        1        7        2       10
## 521 20151114 Air Force     Utah State        1       11       25       17
## 830 20160910 Air Force  Georgia State        1        3        9        9
##     APassAtt
## 111       33
## 253       29
## 415       39
## 490        8
## 521       47
## 830       27
```

Also check for NA values

```
#colSums function loops through each column and sums its number of NA values
na_counts <- colSums(is.na(GSM5))
na_counts
```

```
##       Date       Home       Away   HomeWins   HPassCmp   APassCmp   HPassAtt
##          0          0          0          0          0          0          0
```

```
##   APassAtt   HPassPct   APassPct   HPassYds   APassYds    HPassTD    APassTD
##          0          0          1          0          0          0          0
##   HRushAtt   ARushAtt   HRushYds   ARushYds   HRushAvg   ARushAvg    HRushTD
##          0          0          0          0          0          0          0
##    ARushTD       HXPM       AXPM       HXPA       AXPA HXPPercent AXPPercent
##          0          0          0          0          0        114        202
##       HFGM       AFGM       HFGA       AFGA HFGPercent AFGPercent   HKickPts
##          0          0          0          0        519        568          0
##   AKickPts       HFum       AFum       HInt       AInt   HTotalTO   ATotalTO
##          0          0          0          0          0          0          0
```

```
GSM5[which(is.na(GSM5$APassPct)),1:10]
```

```
##          Date      Home Away HomeWins HPassCmp APassCmp HPassAtt APassAtt
## 1940 20171104 Air Force Army        0        6        0       13        0
##      HPassPct APassPct
## 1940     46.2       NA
```

Since the NA value in row 12 column APassPct results from a divide by 0 error in (APassComp/APassAtt), I will replace it with 0

```
GSM5[which(is.na(GSM5$APassPct)),"APassPct"] = 0
```

Looks like all the stats that involve percentages are the columns with NA values, if it's a dive by 0 problem I will also convert those NAs to 0.

HXPPercent = (HXPM / HXPA)

```
hxpp_nas <- which(is.na(GSM5$HXPPercent))
head(GSM5[hxpp_nas,c("HXPM", "HXPA", "HXPPercent")])
```

```
##      HXPM HXPA HXPPercent
## 1940    0    0         NA
## 211     0    0         NA
## 1012    0    0         NA
## 2721    0    0         NA
## 3141    0    0         NA
## 2418    0    0         NA
```

```
GSM5[hxpp_nas,"HXPPercent"]=0
head(GSM5[hxpp_nas,c("HXPM", "HXPA", "HXPPercent")])
```

```
##      HXPM HXPA HXPPercent
## 1940    0    0          0
## 211     0    0          0
## 1012    0    0          0
## 2721    0    0          0
## 3141    0    0          0
## 2418    0    0          0
```

AXPPercent = (AXPM / AXPA)

```
axpp_nas <- which(is.na(GSM5$AXPPercent))
GSM5[axpp_nas,"AXPPercent"]=0
head(GSM5[axpp_nas,c("AXPM", "AXPA", "AXPPercent")])
```

```
##      AXPM AXPA AXPPercent
## 490     0    0          0
## 544     0    0          0
## 1708    0    0          0
## 177     0    0          0
## 894     0    0          0
## 922     0    0          0
```

HFGPercent = (HFGM / HFGA)

```
hfgp_nas <- which(is.na(GSM5$HFGPercent))
GSM5[hfgp_nas,"HFGPercent"]=0
head(GSM5[hfgp_nas, c("HFGM","HFGA", "HFGPercent" ), ])
```

```
##      HFGM HFGA HFGPercent
## 521     0    0          0
## 1245    0    0          0
## 1972    0    0          0
## 2419    0    0          0
## 2687    0    0          0
## 3013    0    0          0
```

AFGPercent = (AFGM / AFGA)

```
afgp_nas <- which(is.na(GSM5$AFGPercent))
GSM5[afgp_nas,"AFGPercent"]=0
head(GSM5[afgp_nas, c("AFGM","AFGA", "AFGPercent" ), ])
```

```
##      AFGM AFGA AFGPercent
## 415     0    0          0
## 521     0    0          0
## 921     0    0          0
## 1972    0    0          0
## 2037    0    0          0
## 2718    0    0          0
```

Check for any remaining NA values

```
colSums(is.na(GSM5))
```

```
##       Date       Home       Away   HomeWins   HPassCmp   APassCmp   HPassAtt
##          0          0          0          0          0          0          0
##   APassAtt   HPassPct   APassPct   HPassYds   APassYds    HPassTD    APassTD
##          0          0          0          0          0          0          0
##   HRushAtt   ARushAtt   HRushYds   ARushYds   HRushAvg   ARushAvg    HRushTD
##          0          0          0          0          0          0          0
```

```
##    ARushTD       HXPM       AXPM       HXPA       AXPA HXPPercent AXPPercent
##          0          0          0          0          0          0          0
##       HFGM       AFGM       HFGA       AFGA HFGPercent AFGPercent   HKickPts
##          0          0          0          0          0          0          0
##   AKickPts       HFum       AFum       HInt       AInt   HTotalTO   ATotalTO
##          0          0          0          0          0          0          0
```

Create new csv for cleaned data

```
write.csv(GSM5, "CleanedGamesStats.csv")
```

```
head(GSM5)
```

```
##         Date      Home           Away HomeWins HPassCmp APassCmp HPassAtt
## 111 20150912 Air Force San Jose State        1        3       18        11
## 253 20151010 Air Force        Wyoming        1        5       15        10
## 415 20151024 Air Force   Fresno State        1        6       14        11
## 490 20151107 Air Force           Army        1        7        2        10
## 521 20151114 Air Force     Utah State        1       11       25        17
## 830 20160910 Air Force  Georgia State        1        3        9         9
##     APassAtt HPassPct APassPct HPassYds APassYds HPassTD APassTD HRushAtt
## 111       33     27.3     54.5       24      140       0       1       69
## 253       29     50.0     51.7       80      192       1       2       58
## 415       39     54.5     35.9      128      177       1       0       79
## 490        8     70.0     25.0      156       45       2       0       47
## 521       47     64.7     53.2      271      364       1       4       64
## 830       27     33.3     33.3       67      142       0       1       83
##     ARushAtt HRushYds ARushYds HRushAvg ARushAvg HRushTD ARushTD HXPM AXPM HXPA
## 111       20      428      150      6.2      7.5       5       1    4    1    5
## 253       35      299      115      5.2      3.3       3       0    4    2    4
## 415       17      458      134      5.8      7.9       5       2    6    2    6
## 490       44      196      124      4.2      2.8       0       0    2    0    2
## 521       23      309       75      4.8      3.3       4       0    5    4    5
## 830       14      464       27      5.6      1.9       5       1    6    2    6
##     AXPA HXPPercent AXPPercent HFGM AFGM HFGA AFGA HFGPercent AFGPercent
## 111    2         80         50    1    1    1    2        100         50
## 253    2        100        100    1    1    2    1         50        100
## 415    2        100        100    0    0    1    0          0          0
## 490    0        100          0    2    1    2    1        100        100
## 521    4        100        100    0    0    0    0          0          0
## 830    2        100        100    2    0    2    1        100          0
##     HKickPts AKickPts HFum AFum HInt AInt HTotalTO ATotalTO
## 111        7        4    0    0    1    2        1        2
## 253        7        5    1    2    1    2        2        4
## 415        6        2    1    0    0    1        1        1
## 490        8        3    0    0    0    0        0        0
## 521        5        4    0    0    0    1        0        1
## 830       12        2    0    0    0    0        0        0
```

Data Visualization I will create visualizations for the ACC conference, which include the following schools:

```r
acc_schools <- c("Clemson", "Duke", "North Carolina", "North Carolina State", "Wake Forest", "Virginia"
```

```r
Cleaned_ACC_Col <- GSM5 %>%
  mutate(
    HomeConference = ifelse(Home %in% acc_schools, "ACC", "Other")) %>%
  select(Date, Home, HomeConference, Away, everything())
```

```r
acc_games <- Cleaned_ACC_Col %>%
  filter(HomeConference == "ACC")
```

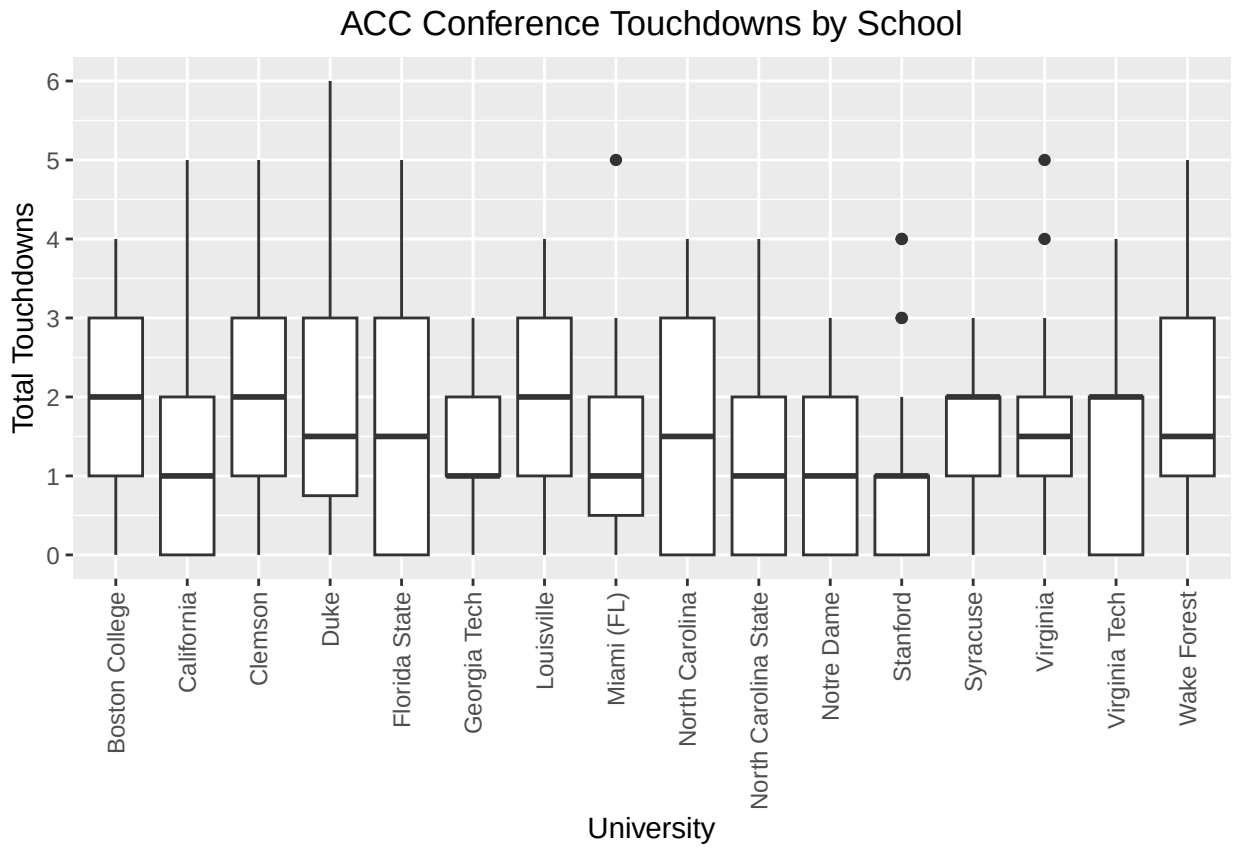
```r
head(acc_games)
```

```
##       Date           Home HomeConference                 Away HomeWins HPassCmp
## 1 20150918 Boston College            ACC        Florida State        0        5
## 2 20150926 Boston College            ACC    Northern Illinois        1        5
## 3 20151010 Boston College            ACC          Wake Forest        0        6
## 4 20151031 Boston College            ACC        Virginia Tech        0        8
## 5 20151107 Boston College            ACC North Carolina State        0       23
## 6 20161001 Boston College            ACC              Buffalo        1       15
##   APassCmp HPassAtt APassAtt HPassPct APassPct HPassYds APassYds HPassTD
## 1       15       15       24     33.3     62.5       56      119       0
## 2       11       14       25     35.7     44.0       92       81       1
## 3       11       20       25     30.0     44.0       74      109       0
## 4       16       21       23     38.1     69.6      143      194       0
## 5       14       37       27     62.2     51.9      257      212       1
## 6       10       27       23     55.6     43.5      258       41       2
##   APassTD HRushAtt ARushAtt HRushYds ARushYds HRushAvg ARushAvg HRushTD ARushTD
## 1       1       43       33      139       98      3.2      3.0       0       0
## 2       0       63       31      234       72      3.7      2.3       1       1
## 3       0       54       28      196       33      3.6      1.2       0       0
## 4       1       35       46       75       81      2.1      1.8       1       0
## 5       1       34       33       28      139      0.8      4.2       0       2
## 6       0       57       19      142       26      2.5      1.4       3       0
##   HXPM AXPM HXPA AXPA HXPPercent AXPPercent HFGM AFGM HFGA AFGA HFGPercent
## 1    0    2    0    2          0        100    0    0    0    1          0
## 2    2    2    2    2        100        100    1    0    1    1        100
## 3    0    0    0    0          0          0    0    1    2    2          0
## 4    1    2    1    2        100        100    1    4    2    4         50
## 5    0    3    0    3          0        100    0    1    0    1          0
## 6    5    0    5    0        100          0    0    1    0    1          0
##   AFGPercent HKickPts AKickPts HFum AFum HInt AInt HTotalTO ATotalTO
## 1          0        0        2    1    0    1    0        2        0
## 2          0        5        2    0    1    1    1        1        2
## 3         50        0        3    3    1    1    1        4        2
## 4        100        4       14    3    1    1    1        4        2
## 5        100        0        6    1    1    3    1        4        2
## 6        100        5        3    2    1    0    0        2        1
```

```r
library(dplyr)
library(ggplot2)
```

Boxplot of each university's touchdowns in the ACC Conference

```
ggplot(acc_games, aes(x = Home, y=HTotalTO)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, vjust=0.5, hjust=1),
        plot.title = element_text(hjust = 0.5)) +
  labs(title = "ACC Conference Touchdowns by School",
       x = "University",
       y = "Total Touchdowns") +
  scale_y_continuous(breaks = seq(0, 7, by = 1))
```



Create new variables. *DiffPassCmp*: How the home team compares in number of passes completed against the away team (home passes - away passes)

DiffXPM: How the home team compares in number of extra points made after touchdowns against the away team (home extra points - away extra points)

DiffTotalTO: How the home team compares in the number of total touch downs against the away team (HTotalTO - ATotalTO)

```
acc_games <- acc_games %>%
  mutate(DiffPassCmp = HPassCmp - APassCmp,
         DiffXPM = HXPM - AXPM,
         DiffTotalTO = HTotalTO - ATotalTO) %>%
  select(c(colnames(acc_games)[1:7],
           DiffPassCmp,
           colnames(acc_games)[8:25],
           DiffXPM,
           everything(),
```

```
          DiffTotalTO
          ))
head(acc_games)
```

```
##       Date           Home HomeConference                Away HomeWins HPassCmp
## 1 20150918 Boston College            ACC        Florida State        0        5
## 2 20150926 Boston College            ACC    Northern Illinois        1        5
## 3 20151010 Boston College            ACC          Wake Forest        0        6
## 4 20151031 Boston College            ACC        Virginia Tech        0        8
## 5 20151107 Boston College            ACC North Carolina State        0       23
## 6 20161001 Boston College            ACC              Buffalo        1       15
##   APassCmp DiffPassCmp HPassAtt APassAtt HPassPct APassPct HPassYds APassYds
## 1       15         -10       15       24     33.3     62.5       56      119
## 2       11          -6       14       25     35.7     44.0       92       81
## 3       11          -5       20       25     30.0     44.0       74      109
## 4       16          -8       21       23     38.1     69.6      143      194
## 5       14           9       37       27     62.2     51.9      257      212
## 6       10           5       27       23     55.6     43.5      258       41
##   HPassTD APassTD HRushAtt ARushAtt HRushYds ARushYds HRushAvg ARushAvg HRushTD
## 1       0       1       43       33      139       98      3.2      3.0       0
## 2       1       0       63       31      234       72      3.7      2.3       1
## 3       0       0       54       28      196       33      3.6      1.2       0
## 4       0       1       35       46       75       81      2.1      1.8       1
## 5       1       1       34       33       28      139      0.8      4.2       0
## 6       2       0       57       19      142       26      2.5      1.4       3
##   ARushTD HXPM AXPM DiffXPM HXPA AXPA HXPPercent AXPPercent HFGM AFGM HFGA AFGA
## 1       0    0    2      -2    0    2          0        100    0    0    0    1
## 2       1    2    2       0    2    2        100        100    1    0    1    1
## 3       0    0    0       0    0    0          0          0    0    1    2    2
## 4       0    1    2      -1    1    2        100        100    1    4    2    4
## 5       2    0    3      -3    0    3          0        100    0    1    0    1
## 6       0    5    0       5    5    0        100          0    0    1    0    1
##   HFGPercent AFGPercent HKickPts AKickPts HFum AFum HInt AInt HTotalTO ATotalTO
## 1          0          0        0        2    1    0    1    0        2        0
## 2        100          0        5        2    0    1    1    1        1        2
## 3          0         50        0        3    3    1    1    1        4        2
## 4         50        100        4       14    3    1    1    1        4        2
## 5          0        100        0        6    1    1    3    1        4        2
## 6          0        100        5        3    2    1    0    0        2        1
##   DiffTotalTO
## 1           2
## 2          -1
## 3           2
## 4           2
## 5           2
## 6           1
```

Scatter plot and linear regression model predicting the Home Team's Total Touchdown differential by the Home Team's Passes Completed differential

```
#Linear regression model
pass_to_mod <- lm(DiffTotalTO ~ DiffPassCmp, acc_games)
#Scatter plot
```

```
plot(main = "Home Team Performance in Passes Completed & Total Touchdowns",
     x = acc_games$DiffPassCmp,
     y = acc_games$DiffTotalTO,
     col = c("red", "blue"),
     pch = 16,
     xlab = "Passes Completed Differential",
     ylab = "Total Touchdowns Differential",
     ylim = c(-8,10))
legend("topleft",
       legend =
         c("Home Passes Completed - Away Passes Completed",
           "Home Total Touchdowns - Away Total Touchdowns"),
       col = c("red", "blue"),
       pch = 16)
axis(side = 2, at = seq(-10, 10), by = 1)
```

```
## Warning in axis(side = 2, at = seq(-10, 10), by = 1): "by" is not a graphical
## parameter
```

```
#Fit linear model
abline(pass_to_mod, lwd=1.5)
```