Will Vucevic Score a Double Double?

Amanda

February 12, 2021

Contents

Model to predict whether the Magic's Nikola Vucevic will score a double double in a given game.

0.1 Importing Data

Statistics for individual seasons that listed every NBA game Nikola Vucevic played in were obtained from Basketball Reference.

- **Data Source:** https://www.basketball-reference.com/players/v/vucevni01.html
- **Seasons:** 10 Seasons total
- **First Season:** 2011-2012
- **Last Season:** 2020-2021 incomplete as games are in progress
- **Naming Convention:** 2021 indicates the 2020-2021 season following Basketball Reference naming convention

0.2 Merge & Clean Data

0.2.1 Data Frame Created

- **Vucevic:**: Includes Stats for every game Vucevic played from 2011 through 2021-02-12. Each row of data represents a single game.

0.2.2 Variables Created

- **Vucevic$TeamGameSeason:** Indicates what game it is in the season for Vucevic's team
- **Vucevic$PlayerGameSeason:** Indicates how many games Vucevic has played in that season
- **Vucevic$PlayerGameCareer:** Indicates how many games Vucevic has played in his entire NBA career
- **Vucevic$Player:** Identifies Vucevic as the player. Future benefit if combined with other datasets.

The variables in the dataset include:

```
##  [1] "TeamGameSeason"   "PlayerGameSeason" "Date"             "Age"
##  [5] "Team"             "Location"         "Opponent"         "GameStarted"
##  [9] "MinsPlayed"       "FG"               "FGA"              "FG%"
## [13] "3P"               "3PA"              "3P%"              "FT"
## [17] "FTA"              "FT%"              "ORB"              "DRB"
## [21] "TRB"              "AST"              "STL"              "BLK"
## [25] "TOV"              "PF"               "PTS"              "GmSc"
## [29] "+/-"              "Season"           "PlayerGameCareer" "Player"
## [33] "Minutes"          "Seconds"          "WinLoss"
```

0.3 Merge Teams that Have Moved

Vucevic has played against 32 NBA teams in his career. With only 30 teams in the league this has occurred as teams have moved and changed names. For example, the New Jersey Nets became the Brooklyn Nets in 2012. To keep a consistent 30 NBA teams we will use 'BRK' to indicate games played against in both locations.

0.3.1 BRK Team will Include:

- NJN New Jersey Nets 1977 to 2012
- BRK Brooklyn Nets 2012 - Present

Similarly the Charlotte Hornets moved from Charlotte to New Orleans to become the New Orleans Pelicans in 2002. In 2004 the NBA established the Charlotte Bobcats as an expansion team. The Charlotte Bobcats changed their name to Charlotte Hornets in 2014. Honoring Charlotte's history all games played in Charlotte will be included in the CHA Team.

0.3.2 CHO Team will Include:

- CHA was used for the Charlotte Bobcats from 2004 to 2014
- CHO is used for the Charlotte Hornets from 2014 to present

The updated count of games Vucevic has played against every NBA team is:

```
##
## ATL BOS BRK CHA CHI CHO CLE DAL DEN DET GSW HOU IND LAC LAL MEM MIA MIL MIN NJN
##  30  27  28   0  26  26  26  15  14  26  16  14  29  16  13  16  30  28  16   0
## NOH NOP NYK OKC ORL PHI PHO POR SAC SAS TOR UTA WAS
##   2  13  28  18   1  27  16  15  13  17  31  17  30
```

0.4 Create Double Double Variable

0.4.1 Variable Created

- **DoubleDouble:** Indicates for each game whether Vucevic scored a double double.

Where a double double is defined as scoring 10 or more in two categories of either points, rebounds, assists, steals or blocks. Typically the most common combinations are 10 or more points and 10 or more rebounds. Followed by 10 or more points and 10 or more assists.

```
##
## FALSE  TRUE
##   342   282
```

```
##      TRUE
## 0.4519231
```

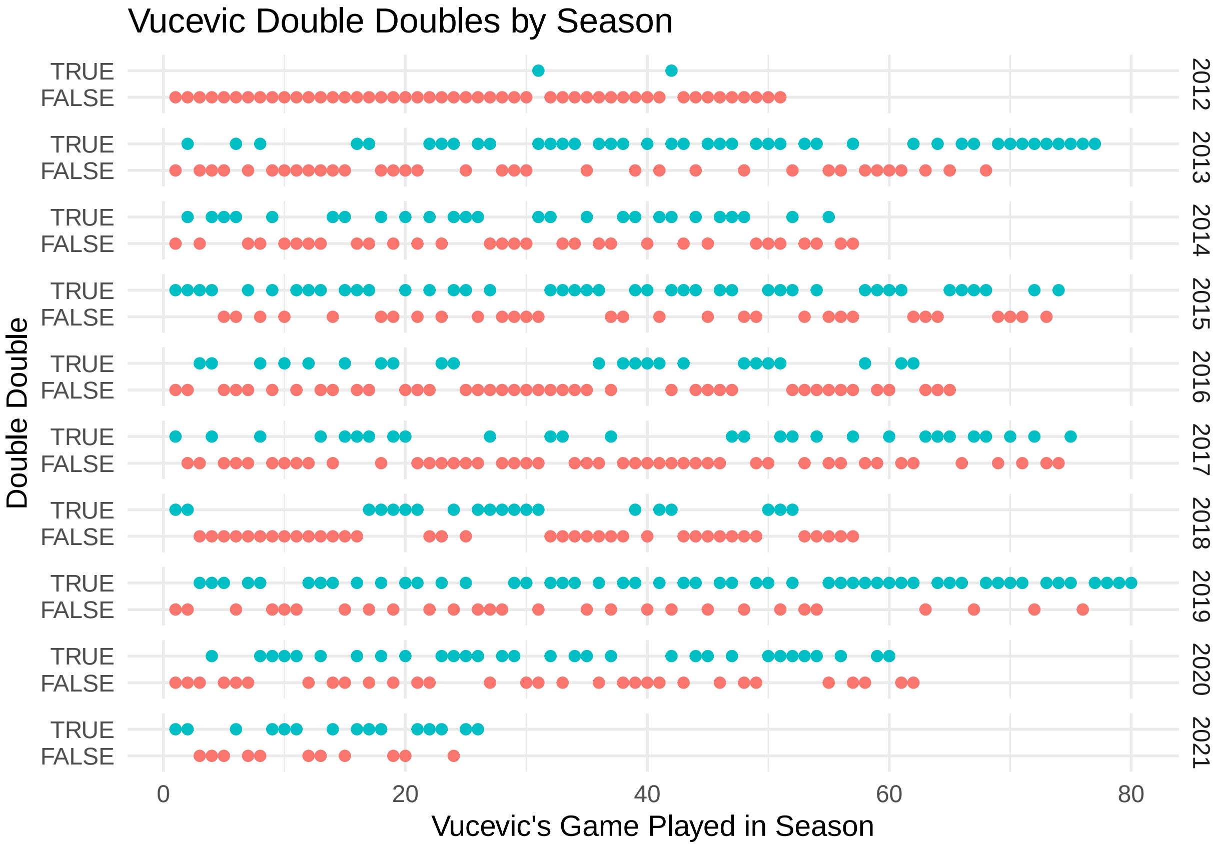
0.5 Create Data Dictionary

Please see the data dictionary for information on all the variables included in the dataset.

```
##    ColNumber     VariableName  DataType
## 1          1   TeamGameSeason   integer
## 2          2 PlayerGameSeason   integer
## 3          3             Date      Date
## 4          4              Age character
## 5          5             Team    factor
## 6          6         Location    factor
## 7          7         Opponent    factor
## 8          8      GameStarted    factor
## 9          9       MinsPlayed character
## 10        10               FG   integer
## 11        11              FGA   integer
## 12        12              FG%   numeric
## 13        13               3P   integer
## 14        14              3PA   integer
## 15        15              3P%   numeric
## 16        16               FT   integer
## 17        17              FTA   integer
## 18        18              FT%   numeric
## 19        19              ORB   integer
## 20        20              DRB   integer
## 21        21              TRB   integer
## 22        22              AST   integer
## 23        23              STL   integer
## 24         24              BLK   integer
## 25         25              TOV   integer
## 26         26               PF   integer
## 27         27              PTS   integer
## 28         28             GmSc   numeric
## 29         29              +/-   integer
## 30         30           Season   numeric
## 31         31 PlayerGameCareer   integer
## 32         32           Player character
## 33         33          Minutes character
## 34         34          Seconds character
## 35         35          WinLoss    factor
## 36         36     DoubleDouble   logical
## 37         37     TripleDouble   logical
```

0.6 Response Variable Visualization

By visualizing Vucevic's Double Double's overtime we can see that as he has progressed in his career he has increasingly achieved a Double Double.



## Modeling Process, Sampling

- **Vucevic:** Observations of all games played
- **VucevicTrain:** 70% of the observations in the Vucevic dataset
- **VucevicTest:** 30% of the observations in the Vucevic dataset

To provide an accurate understanding of the our final optimal model the **Vucevic** dataframe was split into both a training and testing dataset. The training set will be used to train our algorithms and compare models. The testing set will be reserved to make an unbiased assessment of the model's performance.

A 70/30% split was used across 627 observations in the original Vucevic dataset.

A stratified sampling approach was taken to ensure a balanced representation of the response distribution in both the training and testing datasets. In other words we used a similar percent of games where Vucevic scored a Double Double in both the training and testing data.

The breakdown of Double Double's on the **VucevicTrain** dataset:

```
##
##     FALSE      TRUE
## 0.5479744 0.4520256
```

The breakdown of Double Double's on the **VucevicTest** dataset:

```
##
##     FALSE      TRUE
## 0.5483871 0.4516129
```

0.7 Logistic Regression

Logistic regression is used when the response variable is binary. In this case Vucevic either scored a Double Double or he did not.

0.8 Model 1: Logistic Regression Location

For our first model we explore if Vucevic is more likely to score a Double Double depending on if the game is at home or away.

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic p.value
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   -0.184      0.133   -1.39     0.166
## 2 LocationHome  -0.0167     0.186   -0.0902   0.928
```

- **Intercept:** -0.183922837867232
- **Coefficient, Home:** -0.016747857062125
- **P-Value, Home:** 0.928102289648827

Based on our analysis we predict that Vucevic is just as likely to score a Double Double whether at Home or Away. With the high p-value it can also be determined that the location of the game is not statistically significant in determining whether Vucevic will score a Double Double.

0.9 Model 2: Logistic Regression Opponent

Predicts if Vucevic will score a Double Double depending on the opposing team.

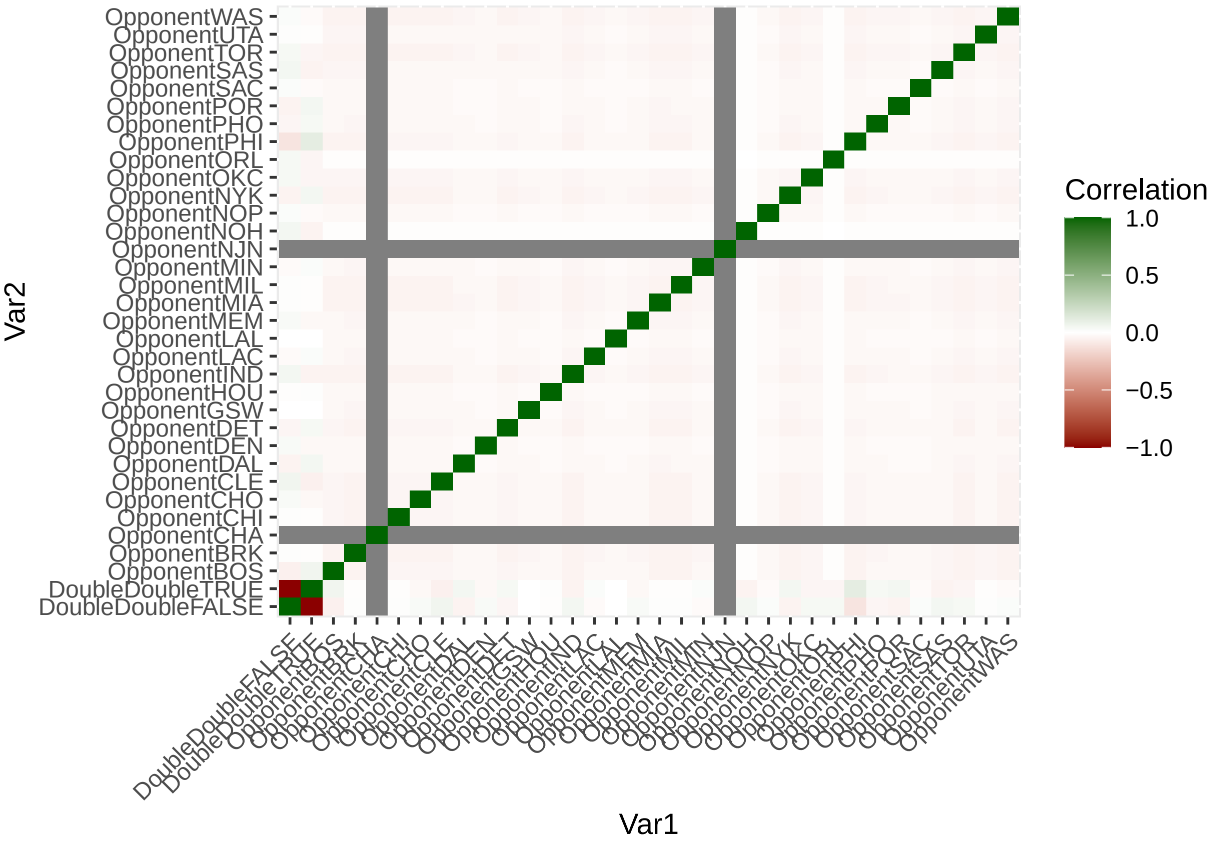
0.9.1 Correlation Matrix: Model 2, Logistical Regression Opponent

A correlation Matrix is used to visualize how each Team impacts Vucevic's likelihood to score a Double Double.

```
VucevicMatrix <- model.matrix(~ 0 + DoubleDouble + Opponent, data = Vucevic)

cor.mat <- round(cor(VucevicMatrix), 2)
```

```
## Warning in stats::cor(x, ...): the standard deviation is zero
```

```
melted.cor.mat <- melt(cor.mat)

ggplot(melted.cor.mat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(x=Var1, y=Var2, label = " ")) +
  scale_fill_gradient2(low = "darkred", high = "darkgreen", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, size = 9, hjust = 1))
```



Interpreting the correlation matrix graphic:

- Grey lines for CHA and NJN indicating there are no values.
- Dark green boxes, indicating perfect positive correlation, for every instance when one variable equals another variable.

- Dark red boxes, indicating perfect negative correlation, when DoubleDoubleFalse equals DoubleDoubleTrue.
- Faint red boxes everywhere else.

It appears that there is little correlation between the team Vucevic played and the likelihood he would score a double double. We can use a Logistical regression model to confirm the numeric value of the correlation using a coefficient estimate, standard error and p-value.

An initial interpretation of the coefficient for each team would indicate that when Vucevic is playing against teams with a positive coefficient value he is more likely to score a Double Double.

```
LogisticRegOpponent <- glm(DoubleDouble ~ Opponent, family = "binomial", data = VucevicTrain)
LogisticRegOpponent
```

```
##
## Call:  glm(formula = DoubleDouble ~ Opponent, family = "binomial", data = VucevicTrain)
##
## Coefficients:
## (Intercept)  OpponentBOS  OpponentBRK  OpponentCHI  OpponentCHO  OpponentCLE
##  -4.771e-15    2.624e-01   -2.007e-01   -4.855e-01   -5.596e-01   -5.596e-01
## OpponentDAL  OpponentDEN  OpponentDET  OpponentGSW  OpponentHOU  OpponentIND
##   2.231e-01   -1.504e+00   -3.365e-01   -4.055e-01    1.823e-01   -6.931e-01
## OpponentLAC  OpponentLAL  OpponentMEM  OpponentMIA  OpponentMIL  OpponentMIN
##   1.542e-01    4.804e-15   -2.877e-01    5.726e-15   -4.418e-01    5.513e-15
## OpponentNOP  OpponentNYK  OpponentOKC  OpponentORL  OpponentPHI  OpponentPHO
##   6.931e-01    3.677e-01   -4.055e-01   -1.357e+01    1.030e+00    2.877e-01
## OpponentPOR  OpponentSAC  OpponentSAS  OpponentTOR  OpponentUTA  OpponentWAS
##   1.823e-01    1.639e-15   -6.931e-01   -8.109e-01   -3.365e-01   -2.877e-01
##
## Degrees of Freedom: 468 Total (i.e. Null);  439 Residual
## Null Deviance:       645.8
## Residual Deviance: 620.5     AIC: 680.5
```

```
#list(LogisticRegOpponent$coefficients > 0)
```

0.10 Further Interpertation: Model 2, Logistical Regression Opponent

```
TidyLogisticRegOpponent <- tidy(LogisticRegOpponent)
head(TidyLogisticRegOpponent)
```

```
## # A tibble: 6 x 5
##   term         estimate std.error statistic p.value
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept) -4.77e-15     0.426 -1.12e-14   1.00
## 2 OpponentBOS  2.62e- 1     0.599  4.38e- 1   0.661
## 3 OpponentBRK -2.01e- 1     0.620 -3.24e- 1   0.746
## 4 OpponentCHI -4.86e- 1     0.619 -7.84e- 1   0.433
## 5 OpponentCHO -5.60e- 1     0.615 -9.10e- 1   0.363
## 6 OpponentCLE -5.60e- 1     0.615 -9.10e- 1   0.363
```

0.10.1 Analyzing Boston

Starting with Boston (BOS), the coefficient is positive which would generally indicate that Vucevic is more likely to score a double double when they are the opponent. Further interpretation of our model, shows a large standard error, creating a wide confidence range, and high p-value.

- **Intercept:** -4.77121051439687e-15
- **Coefficient, Boston:** 0.262364264467496
- **P-Value, Boston:** 0.661357106709575

These numbers all indicate that our current Logistical Regression model is not very accurate in predicting if Vucevic will score a Double Double when he plays against Boston. Similar data is seen for all NBA teams.

0.11 Improving the Model: Model 3, Logistical Regression Multiple

One method to improve model accuracy is to incorporate multiple variables when making a prediction. The following methods were utilized to determine which variables to include:

1. Remove variables that create left side leakage.
2. Remove variables that have no meaning for our model
3. Remove variables that are statistically similar to each other.
4. Create new variables.

0.11.1 Left Side Leakage

When predicting if Vucevic will have a Double Double we need to make sure that our prediction is not based on any of the variables that are used to calculate a Double Double.

In addition to removing variables for Points, Rebounds, Assists, Steals and Blocks, variables that are based in part on those variables will also need to be removed. For example Offensive Rebounds, Free Throws, and Free Throw Attempts.

```
#Create a backup of the Vucevic data frame just in case
VucevicBackup <- Vucevic

#Removing Variables from the Data Set Vucevic that cause left side leakage
Vucevic <- Vucevic %>% select(-FG, -FGA, -FT, -FTA, -ORB, -DRB, -TRB, -AST, -STL, -BLK, -PTS, -GmSc, -T
#Any variable names that start with a number, or have a % sign will need to be included in "" to remove
Vucevic <- Vucevic %>% select(-’FG%’, -’3P’, -’3PA’, -’3P%’, -’FT%’)
```

0.11.2 Removing Variables, No Meaning in Model

Since our model is to perdict whether Vucevic will score a Double Double, the variables that are only available after a game has been played need to be removed. They include: - TOV - PF - +/- - WinLoss

Other variables are very similar to each other and were removed: - Date (Similar to PlayerGameCareer) - Age (Similar to PlayerGameCareer) - MinsPlayed (Similar to Minutes) - Player (Only analyzing Vucevic) - Seconds (Similar to Minutes)

```r
#Remove any variables that would be calculated after a game is played
Vucevic <- Vucevic %>% select(-TOV, -PF, -"+/-", -WinLoss)

#Use summary to see if there are variables we can eliminate
summary(Vucevic)
```

```
##  TeamGameSeason  PlayerGameSeason      Date                 Age           
##  Min.   : 1.00   Min.   : 1.00    Min.   :2011-12-28   Length:624        
##  1st Qu.:17.00   1st Qu.:16.00    1st Qu.:2014-01-23   Class :character  
##  Median :37.00   Median :32.00    Median :2016-02-25   Mode  :character  
##  Mean   :38.41   Mean   :33.57    Mean   :2016-05-06                     
##  3rd Qu.:58.00   3rd Qu.:50.00    3rd Qu.:2018-11-09                     
##  Max.   :82.00   Max.   :80.00    Max.   :2021-02-11                     
##                                                                          
##   Team     Location      Opponent   GameStarted  MinsPlayed       
##  ORL:573   Away:313   TOR    : 31   0: 61       Length:624        
##  PHI: 51   Home:311   ATL    : 30   1:563       Class :character  
##                       MIA    : 30               Mode  :character  
##                       WAS    : 30                                 
##                       IND    : 29                                 
##                       BRK    : 28                                 
##                       (Other):446                                 
##      Season     PlayerGameCareer    Player             Minutes         
##  Min.   :2012   Min.   :  1.0    Length:624         Length:624        
##  1st Qu.:2014   1st Qu.:156.8    Class :character   Class :character  
##  Median :2016   Median :312.5    Mode  :character   Mode  :character  
##  Mean   :2016   Mean   :312.5                                         
##  3rd Qu.:2019   3rd Qu.:468.2                                         
##  Max.   :2021   Max.   :624.0                                         
##                                                                       
##    Seconds          DoubleDouble   
##  Length:624         Mode :logical  
##  Class :character   FALSE:342      
##  Mode  :character   TRUE :282      
##                                    
##                                    
##                                    
## 
```

```r
#Remove variables that contain extraneous information
Vucevic <- Vucevic %>% select(-Age, -MinsPlayed, -Player, -Seconds)

#Convert Variables to
Vucevic$Season <- as.factor(Vucevic$Season)
Vucevic$Minutes <- as.factor(Vucevic$Minutes)

#Check your work
summary(Vucevic)
```

```
##  TeamGameSeason  PlayerGameSeason      Date                 Team     Location  
##  Min.   : 1.00   Min.   : 1.00    Min.   :2011-12-28   ORL:573   Away:313  
##  1st Qu.:17.00   1st Qu.:16.00    1st Qu.:2014-01-23   PHI: 51   Home:311  
##  Median :37.00   Median :32.00    Median :2016-02-25                       
##  Mean   :38.41   Mean   :33.57    Mean   :2016-05-06
##  3rd Qu.:58.00   3rd Qu.:50.00    3rd Qu.:2018-11-09
##  Max.   :82.00   Max.   :80.00    Max.   :2021-02-11
##
##     Opponent   GameStarted     Season    PlayerGameCareer    Minutes
##  TOR    : 31   0: 61       2019    : 80   Min.   :  1.0    32     : 56
##  ATL    : 30   1:563       2013    : 77   1st Qu.:156.8    33     : 52
##  MIA    : 30               2017    : 75   Median :312.5    34     : 52
##  WAS    : 30               2015    : 74   Mean   :312.5    31     : 45
##  IND    : 29               2016    : 65   3rd Qu.:468.2    27     : 31
##  BRK    : 28               2020    : 62   Max.   :624.0    28     : 30
##  (Other):446               (Other):191                      (Other):358
##  DoubleDouble
##  Mode :logical
##  FALSE:342
##  TRUE :282
##
##
##
##
```

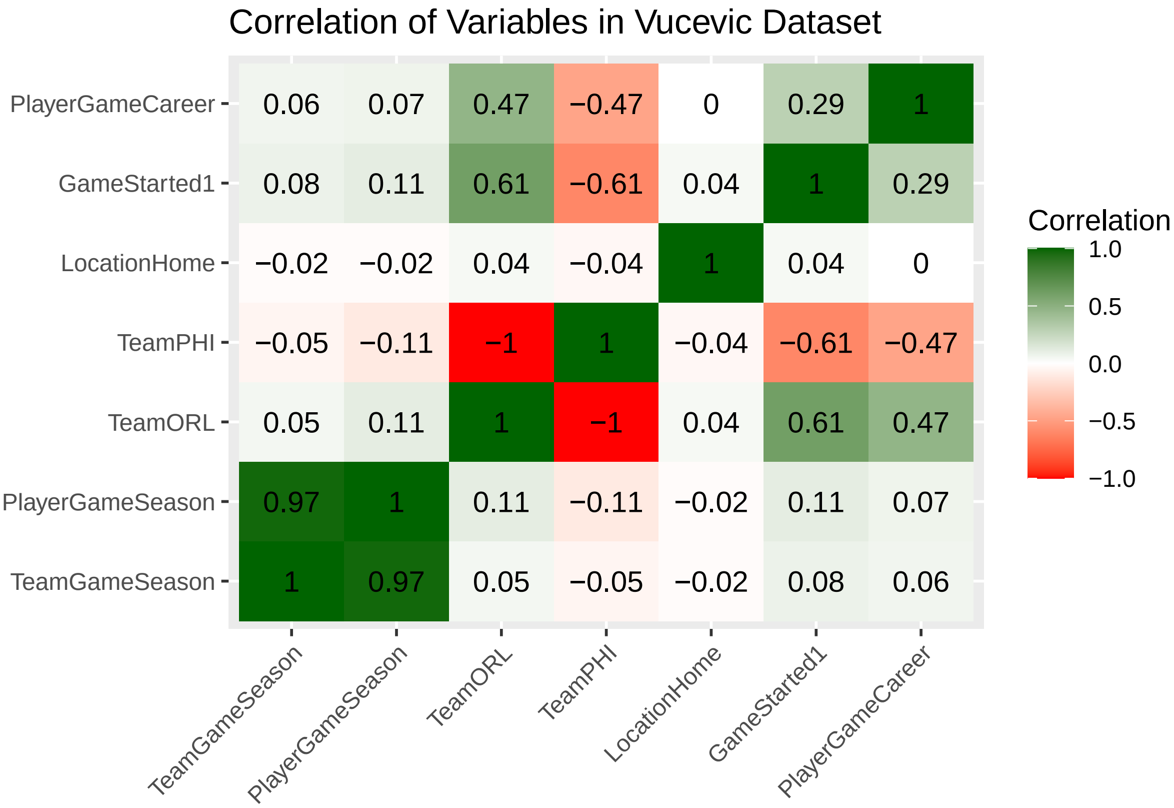
0.11.3 Remove Variables, Statistically Similar

By creating a correlation matrix on the remaining variables, we can determine if any of the variables are statistically similar to each other.

```
names(Vucevic)
```

```
##  [1] "TeamGameSeason"   "PlayerGameSeason" "Date"             "Team"
##  [5] "Location"         "Opponent"         "GameStarted"      "Season"
##  [9] "PlayerGameCareer" "Minutes"          "DoubleDouble"
```

```
VucevicMatrix <- model.matrix(~ 0 + TeamGameSeason + PlayerGameSeason + Team + Location + GameStarted +

cor.mat <- round(cor(VucevicMatrix), 2)
melted.cor.mat <- melt(cor.mat)

ggplot(melted.cor.mat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  ggtitle("Correlation of Variables in Vucevic Dataset") +
  xlab("") +
  ylab("") +
  geom_text(aes(x=Var1, y=Var2, label = value)) +
  scale_fill_gradient2(low = "red", high = "darkgreen", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, size = 9, hjust = 1))
```

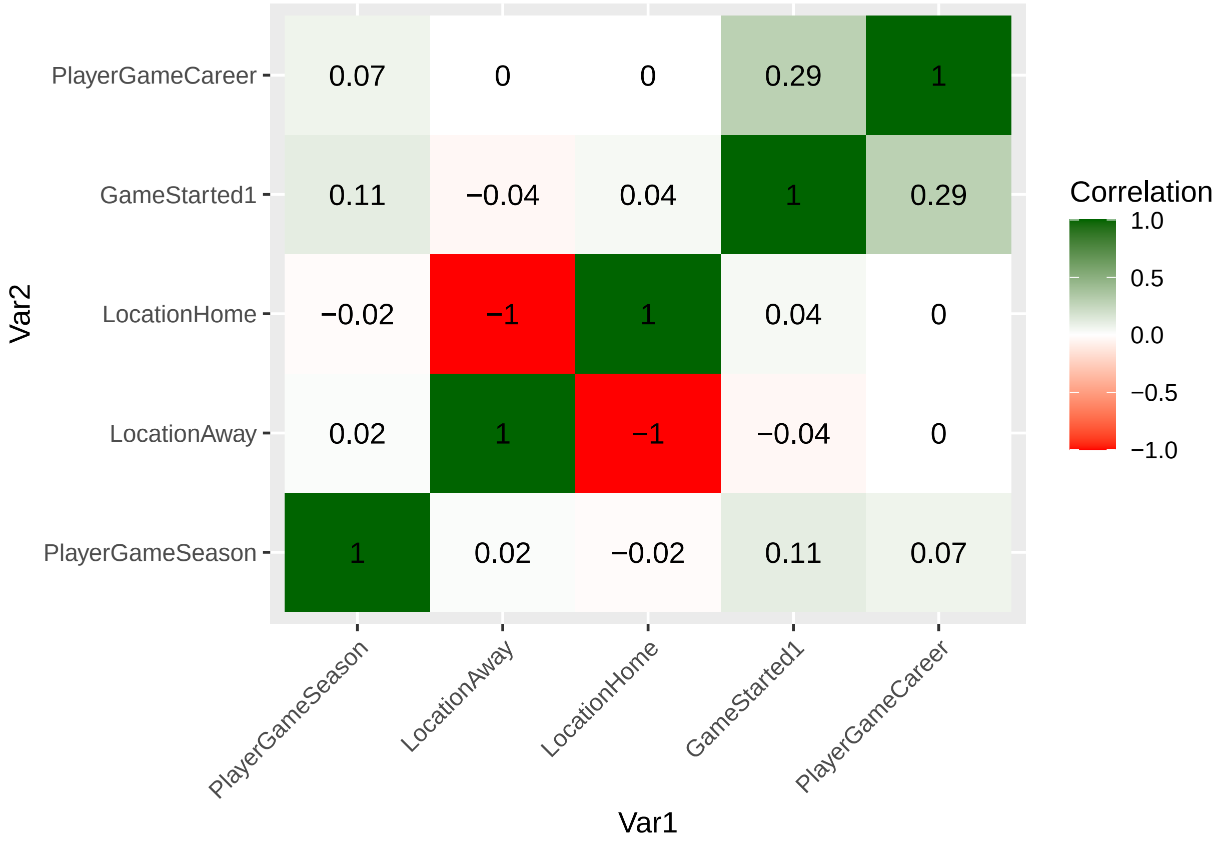


Both TeamPHI and TeamORL have an impact on multiple other variables. This is likely due to the fact that Vucevic only played for Philadelphia in his rookie year. As a rookie he was likely to start (GameStarted1), had played fewer games (PlayerGameCareer), and less likely to score a Double Double.

We also see a strong correlation between TeamGameSeason and PlayerGameSeason. PlayerGameSeason is calculated by the number of games Vucevic has played in a given season, while TeamGameSeason is calculated by the number of games his team has played in a given season. If Vucevic had long or frequent injuries these calculations would show more variation. Since they don't we can get ride of TeamGameSeason.

Once we remove TeamPHI, TeamORL and TeamGameSeason our Correlation Matrix shows:

```
VucevicMatrix <- model.matrix(~ 0 + PlayerGameSeason + Location + GameStarted + PlayerGameCareer, data

cor.mat <- round(cor(VucevicMatrix), 2)
melted.cor.mat <- melt(cor.mat)

ggplot(melted.cor.mat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(x=Var1, y=Var2, label = value)) +
  scale_fill_gradient2(low = "red", high = "darkgreen", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, size = 9, hjust = 1))
```



### Create New Variables

Based on our above findings we are now left with the following variables: - PlayerGameSeason - Location - GameStarted - PlayerGameCareer

Additional variables may provide better insights: - **Back to Back Games:** created using a mutate() and lag()
- **Conference:** - **Time Zone:** Created using an ifelse()

```
#Create New Variable Back to Back
Vucevic <- Vucevic %>%
  arrange(Vucevic$Date) %>%
  mutate(DaysSinceLastGame = Vucevic$Date - lag(Vucevic$Date))
Vucevic$BackToBack <- ifelse(Vucevic$DaysSinceLastGame == 1, TRUE, FALSE)
Vucevic <- Vucevic %>% select(-Date, -DaysSinceLastGame)

#Create New Variable Conference
Vucevic$Conference <-  ifelse(Vucevic$Opponent == "ATL", "Eastern",
                        ifelse(Vucevic$Opponent == "BOS", "Eastern",
                        ifelse(Vucevic$Opponent == "BRK", "Eastern",
                        ifelse(Vucevic$Opponent == "CHA", "Eastern",
                        ifelse(Vucevic$Opponent == "CHO", "Eastern",
                        ifelse(Vucevic$Opponent == "CHI", "Eastern",
                        ifelse(Vucevic$Opponent == "CLE", "Eastern",
                        ifelse(Vucevic$Opponent == "DAL", "Western",
                        ifelse(Vucevic$Opponent == "DEN", "Western",
                        ifelse(Vucevic$Opponent == "DET", "Eastern",
                        ifelse(Vucevic$Opponent == "GSW", "Western",
```

```r
                        ifelse(Vucevic$Opponent == "HOU", "Western",
                        ifelse(Vucevic$Opponent == "IND", "Eastern",
                        ifelse(Vucevic$Opponent == "LAC", "Western",
                        ifelse(Vucevic$Opponent == "LAL", "Western",
                        ifelse(Vucevic$Opponent == "MEM", "Western",
                        ifelse(Vucevic$Opponent == "MIA", "Eastern",
                        ifelse(Vucevic$Opponent == "MIL", "Eastern",
                        ifelse(Vucevic$Opponent == "MIN", "Western",
                        ifelse(Vucevic$Opponent == "NJN", "Eastern",
                        ifelse(Vucevic$Opponent == "NOH", "Western",
                        ifelse(Vucevic$Opponent == "NOP", "Western",
                        ifelse(Vucevic$Opponent == "NYK", "Eastern",
                        ifelse(Vucevic$Opponent == "OKC", "Western",
                        ifelse(Vucevic$Opponent == "ORL", "Eastern",
                        ifelse(Vucevic$Opponent == "PHI", "Eastern",
                        ifelse(Vucevic$Opponent == "PHO", "Western",
                        ifelse(Vucevic$Opponent == "POR", "Western",
                        ifelse(Vucevic$Opponent == "SAC", "Western",
                        ifelse(Vucevic$Opponent == "SAS", "Western",
                        ifelse(Vucevic$Opponent == "TOR", "Eastern",
                        ifelse(Vucevic$Opponent == "UTA", "Western",
                        ifelse(Vucevic$Opponent == "WAS", "Eastern",
                               "Eastern")))))))))))))))))))))))))))))))))

#Create New Variable Time Zone
Vucevic$TimeZone <- ifelse(Vucevic$Location == "Home", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "ATL", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "BOS", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "BRK", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "CHA", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "CHO", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "CHI", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "CLE", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "DAL", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "DEN", "Mountain",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "DET", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "GSW", "Pacific",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "HOU", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "IND", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "LAC", "Pacific",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "LAL", "Pacific",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "MEM", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "MIA", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "MIL", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "MIN", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "NJN", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "NOH", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "NOP", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "NYK", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "OKC", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "ORL", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "PHI", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "PHO", "Mountain",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "POR", "Pacific",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "SAC", "Pacific",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "SAS", "Central",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "TOR", "Eastern",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "UTA", "Mountain",
                        ifelse(Vucevic$Location == "Away" & Vucevic$Opponent == "WAS", "Eastern",
                               "Eastern"))))))))))))))))))))))))))))))))))

summary(Vucevic)
```

```
##  TeamGameSeason  PlayerGameSeason  Team     Location      Opponent
##  Min.   : 1.00   Min.   : 1.00    ORL:573   Away:313   TOR    : 31
##  1st Qu.:17.00   1st Qu.:16.00    PHI: 51   Home:311   ATL    : 30
##  Median :37.00   Median :32.00                         MIA    : 30
##  Mean   :38.41   Mean   :33.57                         WAS    : 30
##  3rd Qu.:58.00   3rd Qu.:50.00                         IND    : 29
##  Max.   :82.00   Max.   :80.00                         BRK    : 28
##                                                        (Other):446
##  GameStarted     Season    PlayerGameCareer    Minutes    DoubleDouble
##  0: 61       2019   : 80   Min.   :  1.0    32     : 56   Mode :logical
##  1:563       2013   : 77   1st Qu.:156.8    33     : 52   FALSE:342
##              2017   : 75   Median :312.5    34     : 52   TRUE :282
##              2015   : 74   Mean   :312.5    31     : 45
##              2016   : 65   3rd Qu.:468.2    27     : 31
##              2020   : 62   Max.   :624.0    28     : 30
##              (Other):191                    (Other):358
##  BackToBack       Conference          TimeZone
##  Mode :logical   Length:624         Length:624
##  FALSE:509       Class :character   Class :character
##  TRUE :114       Mode  :character   Mode  :character
##  NA’s :1
##
##
##
```

After creating the new variables we need to transform them to the correct data type.

- *BackToBack:* Logical
- *Conference:* Factor
- *TimeZone:* Factor

```
Vucevic$Conference <- as.factor(Vucevic$Conference)
Vucevic$TimeZone <- as.factor(Vucevic$TimeZone)
summary(Vucevic)
```

```
##  TeamGameSeason  PlayerGameSeason  Team     Location      Opponent
##  Min.   : 1.00   Min.   : 1.00    ORL:573   Away:313   TOR    : 31
##  1st Qu.:17.00   1st Qu.:16.00    PHI: 51   Home:311   ATL    : 30
##  Median :37.00   Median :32.00                         MIA    : 30
##  Mean   :38.41   Mean   :33.57                         WAS    : 30
##  3rd Qu.:58.00   3rd Qu.:50.00                         IND    : 29
##  Max.   :82.00   Max.   :80.00                         BRK    : 28
##                                                         (Other):446
##  GameStarted     Season    PlayerGameCareer    Minutes    DoubleDouble
##  0: 61       2019   : 80   Min.   :  1.0    32     : 56   Mode :logical
##  1:563       2013   : 77   1st Qu.:156.8    33     : 52   FALSE:342
##              2017   : 75   Median :312.5    34     : 52   TRUE :282
##              2015   : 74   Mean   :312.5    31     : 45
##              2016   : 65   3rd Qu.:468.2    27     : 31
##              2020   : 62   Max.   :624.0    28     : 30
##              (Other):191                    (Other):358
##  BackToBack        Conference      TimeZone
##  Mode :logical   Eastern:393   Central : 85
##  FALSE:509       Western:231   Eastern :481
##  TRUE :114                     Mountain: 24
##  NA’s :1                       Pacific : 34
##
##
##
```

0.12 Test Model 3, Multiple Logistical Regression

0.12.1 Sampling: Model 3, Multiple Logistical Regression

After narrowing the **Vucevic** data set we need to split it into a Training and Testing set before running our model.

Both the training and testing sets have a similar percentage of Double Double responses.

```
#Stratified resampling using the rsample package
set.seed(123)
#initate the split of the UnorderFactos dataset, sampling based on the response variable DoubleDouble
VucevicSplit <- initial_split(Vucevic, prob = 0.7, strata = "DoubleDouble")
VucevicTrain <- training(VucevicSplit)
VucevicTest <- testing(VucevicSplit)

#Shows the response (Double Double) ratio for Training and Testing data
table(VucevicTrain$DoubleDouble) %>% prop.table()
```

```
##
##     FALSE      TRUE
## 0.5479744 0.4520256
```

```
table(VucevicTest$DoubleDouble) %>% prop.table()
```

```
##
##     FALSE      TRUE
## 0.5483871 0.4516129
```

The remaining variables available for our model in the Vucevic dataset include:

```
names(Vucevic)
```

```
##  [1] "TeamGameSeason"   "PlayerGameSeason" "Team"             "Location"
##  [5] "Opponent"         "GameStarted"      "Season"           "PlayerGameCareer"
##  [9] "Minutes"          "DoubleDouble"     "BackToBack"       "Conference"
## [13] "TimeZone"
```

From previous models we already know that both Location and Opponent have little statistical value in determining if Vucevic will score a Double Double, so we exclude these from our model.

```
LogisticRegMultiple <- glm(DoubleDouble ~ TeamGameSeason +GameStarted +Season +PlayerGameCareer +Minutes

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 58 x 5
##    term           estimate std.error statistic p.value
##    <chr>             <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept)    -18.3    6523.      -0.00280   0.998
##  2 TeamGameSeason  -0.0397    0.0287  -1.38      0.167
##  3 GameStarted1    -0.626     0.663   -0.944     0.345
##  4 Season2013      -2.63      2.26    -1.16      0.244
##  5 Season2014      -6.04      4.35    -1.39      0.166
##  6 Season2015      -8.86      6.42    -1.38      0.167
##  7 Season2016     -13.0       8.89    -1.46      0.145
##  8 Season2017     -15.6      11.1     -1.40      0.163
##  9 Season2018     -19.3      13.4     -1.44      0.150
## 10 Season2019     -20.9      15.6     -1.34      0.180
## # ... with 48 more rows
```

0.12.2 Narrowing Variables Model 3, Multiple Logistical Regression

From the output above we see that the model p-value is almost 1, and the minutes played levels are also almost 1. For our next interation of the Multiple Logistical Regression model we remove the variable Minutes from the model.

```
LogisticRegMultiple <- glm(DoubleDouble ~ TeamGameSeason +GameStarted +Season +PlayerGameCareer +BackToB

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 18 x 5
##    term              estimate std.error statistic p.value
##    <chr>                <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept)        -2.54      0.826     -3.08  0.00210
##  2 TeamGameSeason     -0.0512    0.0249    -2.05  0.0403
##  3 GameStarted1       -0.305     0.524     -0.582 0.560
##  4 Season2013         -0.702     1.89      -0.370 0.711
##  5 Season2014         -5.52      3.73      -1.48  0.139
##  6 Season2015         -9.38      5.53      -1.69  0.0901
##  7 Season2016        -15.2       7.67      -1.98  0.0478
##  8 Season2017        -19.7       9.61      -2.05  0.0407
##  9 Season2018        -24.5      11.6       -2.12  0.0344
## 10 Season2019        -27.4      13.5       -2.04  0.0418
## 11 Season2020        -33.1      15.6       -2.12  0.0337
## 12 Season2021        -36.8      17.5       -2.10  0.0357
## 13 PlayerGameCareer    0.0670    0.0291     2.30  0.0212
## 14 BackToBackTRUE     -0.299     0.269     -1.11  0.267
## 15 ConferenceWestern   0.0490    0.239      0.205 0.838
## 16 TimeZoneEastern    -0.178     0.324     -0.551 0.582
## 17 TimeZoneMountain   -0.417     0.609     -0.685 0.494
## 18 TimeZonePacific    -0.0964    0.543     -0.178 0.859
```

The model p-value is 0.00210318137428919 when examining variables TeamGameSeason, GameStarted, Season, PlayerGameCareer, BacktoBack, Conference, and TimeZone.

```
LogisticRegMultiple <- glm(DoubleDouble ~ TeamGameSeason +GameStarted +PlayerGameCareer +BackToBack +Co

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 9 x 5
##   term              estimate std.error statistic p.value
##   <chr>                <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)       -1.47      0.512       -2.86  0.00418
## 2 TeamGameSeason     0.00358   0.00402      0.891 0.373
## 3 GameStarted1       1.15      0.398        2.88  0.00393
## 4 PlayerGameCareer   0.000830  0.000554     1.50  0.134
## 5 BackToBackTRUE    -0.319     0.256       -1.25  0.213
## 6 ConferenceWestern  0.0751    0.227        0.330 0.741
## 7 TimeZoneEastern   -0.145     0.307       -0.472 0.637
## 8 TimeZoneMountain  -0.626     0.585       -1.07  0.284
## 9 TimeZonePacific   -0.226     0.518       -0.437 0.662
```

- **The model p-value is 0.00417626474544997** when examining variables TeamGameSeason, GameStarted, PlayerGameCareer, BacktoBack, Conference, and TimeZone.

```
LogisticRegMultiple <- glm(DoubleDouble ~ TeamGameSeason +GameStarted +PlayerGameCareer +BackToBack +Ti

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 8 x 5
##   term             estimate std.error statistic p.value
##   <chr>               <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)      -1.41      0.485       -2.91  0.00360
## 2 TeamGameSeason    0.00351   0.00401      0.875 0.381
## 3 GameStarted1      1.15      0.398        2.90  0.00378
## 4 PlayerGameCareer  0.000823  0.000553     1.49  0.137
## 5 BackToBackTRUE   -0.330     0.253       -1.30  0.193
## 6 TimeZoneEastern  -0.177     0.291       -0.607 0.544
## 7 TimeZoneMountain -0.602     0.580       -1.04  0.300
## 8 TimeZonePacific  -0.201     0.512       -0.393 0.695
```

- **The model p-value is 0.00359655327765949** when examining variables TeamGameSeason, GameStarted, PlayerGameCareer, BacktoBack, and TimeZone.

```
LogisticRegMultiple <- glm(DoubleDouble ~ TeamGameSeason +GameStarted +PlayerGameCareer +BackToBack, fa

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 5 x 5
##   term              estimate std.error statistic  p.value
##   <chr>                <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      -1.58      0.413       -3.83  0.000129
## 2 TeamGameSeason    0.00373   0.00400      0.931 0.352
## 3 GameStarted1      1.14      0.397        2.88  0.00404
## 4 PlayerGameCareer  0.000809  0.000552     1.46  0.143
## 5 BackToBackTRUE   -0.323     0.251       -1.29  0.198
```

- **The model p-value is 0.000129487665953061** when examining variables TeamGameSeason, GameStarted, PlayerGameCareer, and BacktoBack.

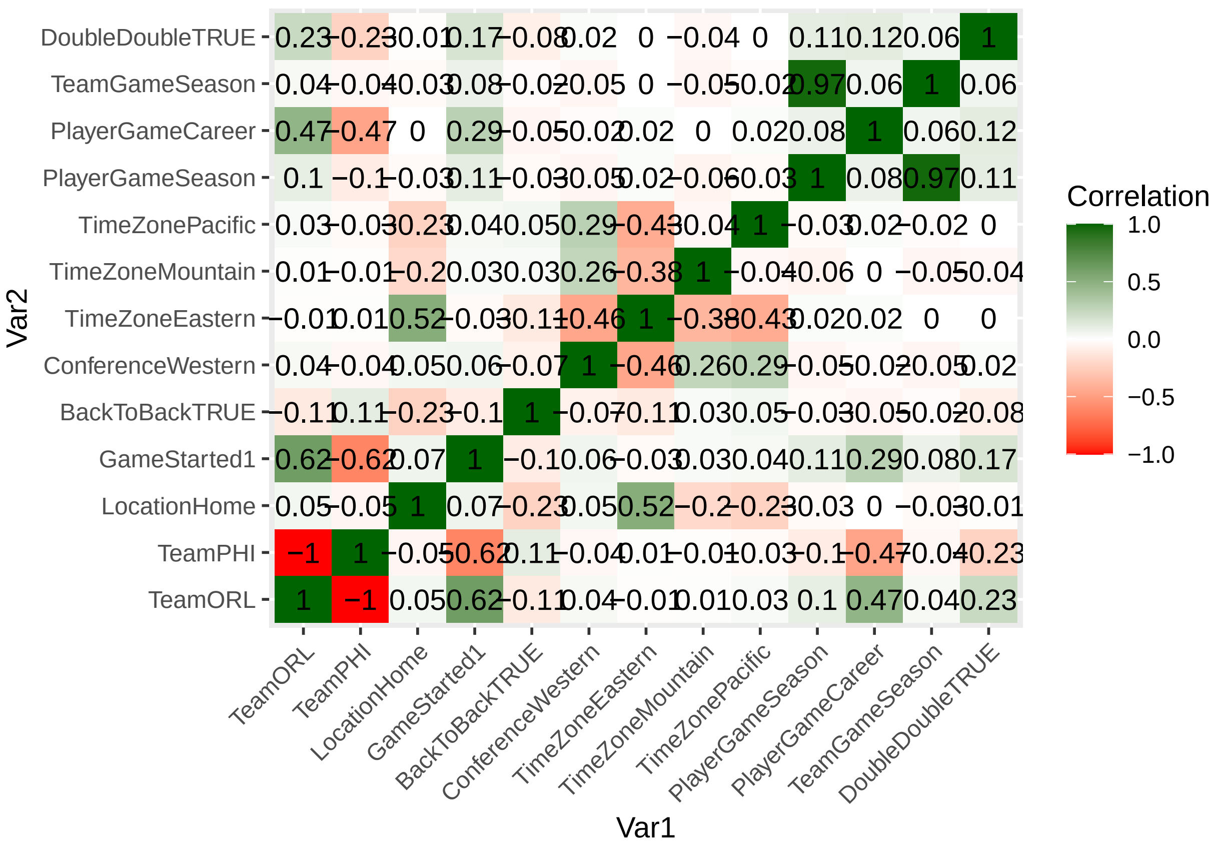
```
LogisticRegMultiple <- glm(DoubleDouble ~ BackToBack +GameStarted +PlayerGameSeason, family = "binomial

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 4 x 5
##   term             estimate std.error statistic   p.value
##   <chr>               <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)      -1.58      0.401       -3.94 0.0000830
## 2 BackToBackTRUE   -0.325     0.251       -1.29 0.196
## 3 GameStarted1      1.24      0.387        3.19 0.00140
## 4 PlayerGameSeason  0.00923   0.00464      1.99 0.0468
```

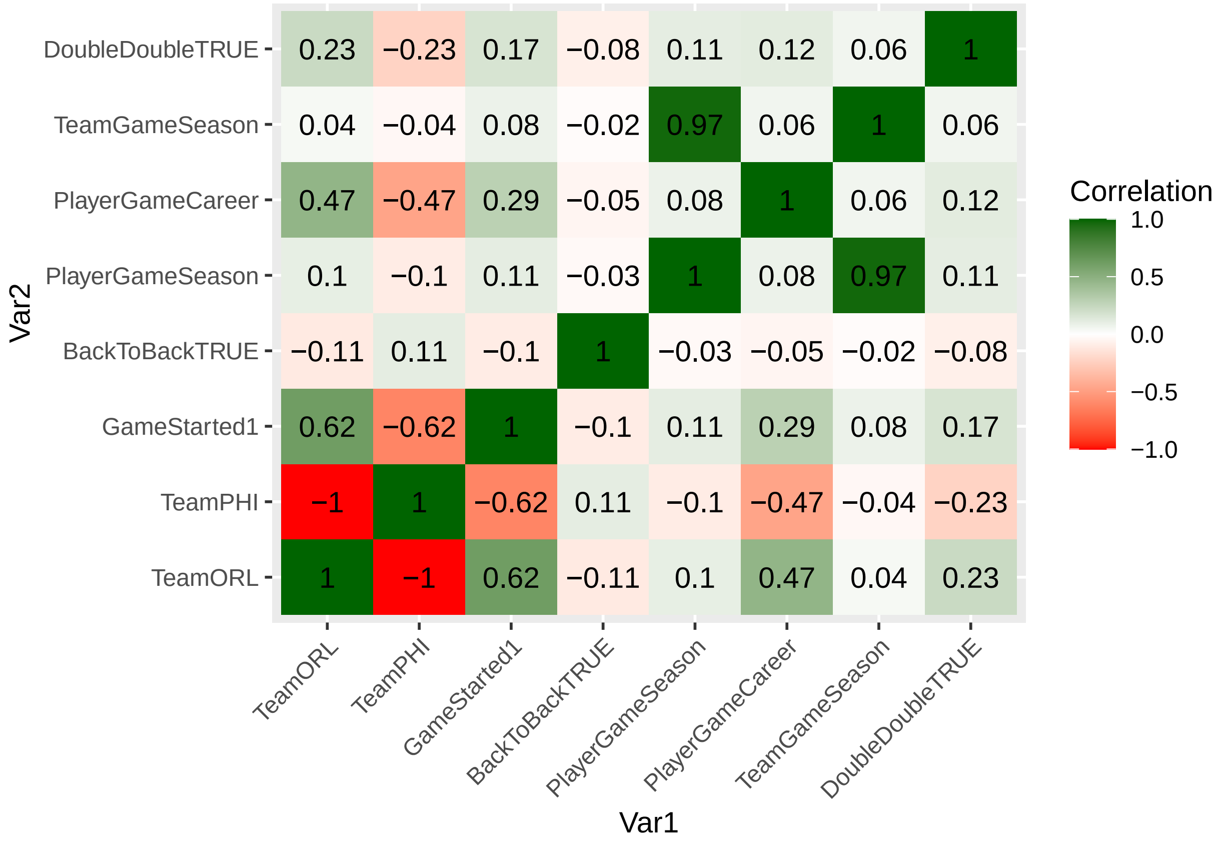
- **The model p-value is 8.3034562763722e-05** when examining variables GameStarted, PlayerGameSeason, and BacktoBack.

Heading

```
VucevicMatrix <- model.matrix(~ 0 +Team +Location +GameStarted +BackToBack +Conference +TimeZone +Playe

cor.mat <- round(cor(VucevicMatrix), 2)
melted.cor.mat <- melt(cor.mat)

ggplot(melted.cor.mat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(x=Var1, y=Var2, label = value)) +
  scale_fill_gradient2(low = "red", high = "darkgreen", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, size = 9, hjust = 1))
```



```
VucevicMatrix <- model.matrix(~ 0 +Team +GameStarted +BackToBack +PlayerGameSeason +PlayerGameCareer +Te

cor.mat <- round(cor(VucevicMatrix), 2)
melted.cor.mat <- melt(cor.mat)

ggplot(melted.cor.mat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(x=Var1, y=Var2, label = value)) +
  scale_fill_gradient2(low = "red", high = "darkgreen", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, size = 9, hjust = 1))
```

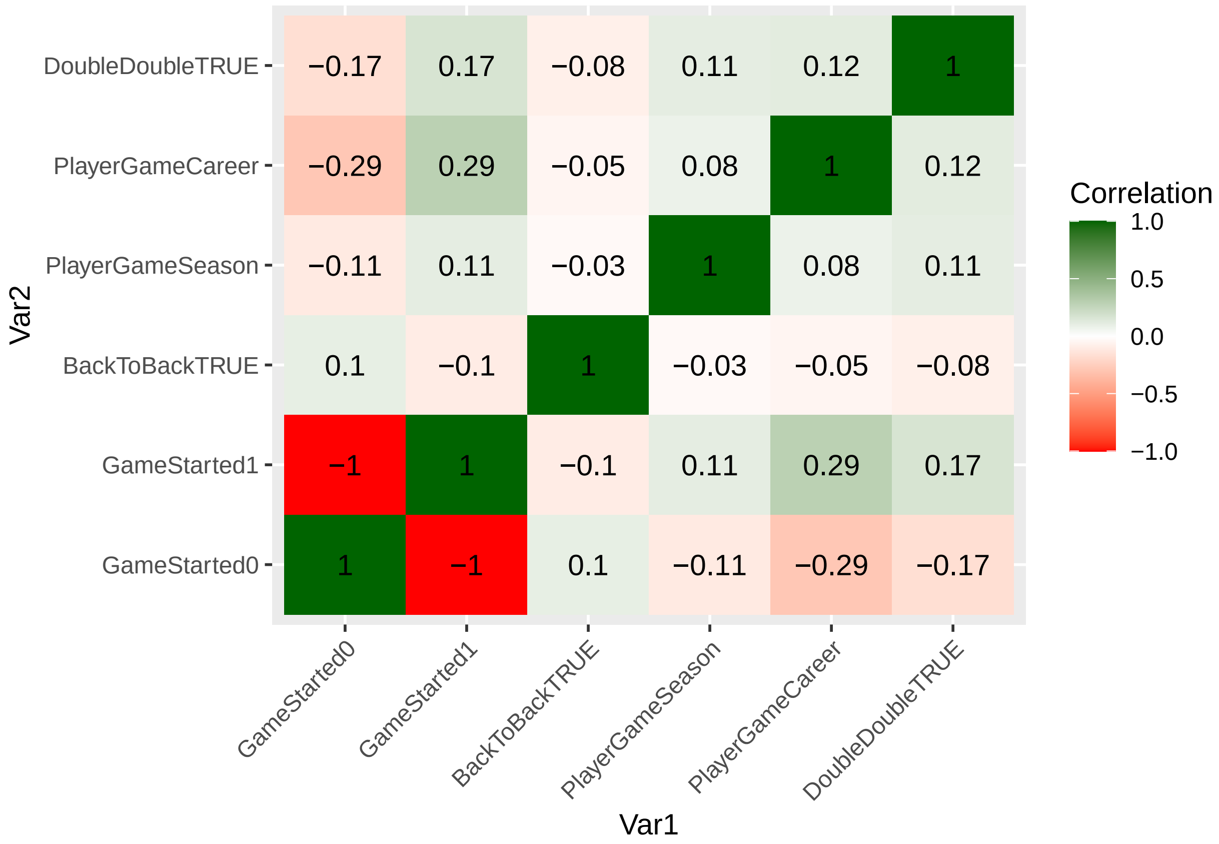


Var2	TeamORL	TeamPHI	GameStarted1	BackToBackTRUE	PlayerGameSeason	PlayerGameCareer	TeamGameSeason	DoubleDoubleTRUE
DoubleDoubleTRUE	0.23	−0.23	0.17	−0.08	0.11	0.12	0.06	1
TeamGameSeason	0.04	−0.04	0.08	−0.02	0.97	0.06	1	0.06
PlayerGameCareer	0.47	−0.47	0.29	−0.05	0.08	1	0.06	0.12
PlayerGameSeason	0.1	−0.1	0.11	−0.03	1	0.08	0.97	0.11
BackToBackTRUE	−0.11	0.11	−0.1	1	−0.03	−0.05	−0.02	−0.08
GameStarted1	0.62	−0.62	1	−0.1	0.11	0.29	0.08	0.17
TeamPHI	−1	1	−0.62	0.11	−0.1	−0.47	−0.04	−0.23
TeamORL	1	−1	0.62	−0.11	0.1	0.47	0.04	0.23

LogisticRegMultiple <- glm(DoubleDouble ~ Team +BackToBack +GameStarted +PlayerGameSeason , family = “binomial”, data = VucevicTrain)

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple) TidyLogisticRegMultiple

```
VucevicMatrix <- model.matrix(~ 0 +GameStarted +BackToBack +PlayerGameSeason +PlayerGameCareer +DoubleDo

cor.mat <- round(cor(VucevicMatrix), 2)
melted.cor.mat <- melt(cor.mat)

ggplot(melted.cor.mat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(x=Var1, y=Var2, label = value)) +
  scale_fill_gradient2(low = "red", high = "darkgreen", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme(axis.text.x = element_text(angle = 45, size = 9, hjust = 1))
```



## Multiple Logistic Regression

```
LogisticRegMultiple <- glm(DoubleDouble ~ +GameStarted +BackToBack +PlayerGameSeason +PlayerGameCareer,

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 5 x 5
##   term               estimate std.error statistic   p.value
##   <chr>                 <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)       -1.70      0.412        -4.13 0.0000360
## 2 GameStarted1       1.11      0.398         2.78 0.00546
## 3 BackToBackTRUE    -0.320     0.252        -1.27 0.204
## 4 PlayerGameSeason   0.00901   0.00466       1.93 0.0532
## 5 PlayerGameCareer   0.000788  0.000554      1.42 0.155
```

```
LogisticRegMultiple <- glm(DoubleDouble ~ +GameStarted +PlayerGameSeason, family = "binomial", data = V

TidyLogisticRegMultiple <- tidy(LogisticRegMultiple)
TidyLogisticRegMultiple
```

```
## # A tibble: 3 x 5
##   term              estimate std.error statistic   p.value
##   <chr>                <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)       -1.69      0.393       -4.31 0.0000160
## 2 GameStarted1       1.29      0.385        3.36 0.000784
## 3 PlayerGameSeason   0.00939   0.00463      2.03 0.0426
```

0.13 K-fold Cross Validation

Resampling Using k-fold cross validation

Since the sample size of our total dataset is only 621 games, validation with a single testing dataset could vary significantly based on the specific games in the 30% held for the testing datset.

k-fold cross validation is a resampling method that randomly divides the training data into k groups. The model is fit on k-1 folds (aka. groups) and then the group that was left out is used to test performance. Which means the model is tested k times and the average k test error is the cross validation estimate.

k-folds: 10

```
#Convert DoubleDouble column to factor if necessary
Vucevic$DoubleDouble <- as.factor(Vucevic$DoubleDouble)
class(Vucevic$DoubleDouble)
```

```
## [1] "factor"
```

```
#K-fold Cross Validation Outside of a Model
set.seed(123)
cv_model1 <- train(
  DoubleDouble ~ BackToBack +GameStarted +PlayerGameSeason,
  data = Vucevic,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10),
  na.action = na.exclude)

cv_model1
```

```
## Generalized Linear Model
##
## 624 samples
##   3 predictor
##   2 classes: 'FALSE', 'TRUE'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 560, 561, 561, 561, 561, 561, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.5729135  0.122902
```

0.13.1 Model 2, Cross Validation

```
#K-fold Cross Validation Outside of a Model
set.seed(123)
cv_model2 <- train(
  DoubleDouble ~ BackToBack +GameStarted +PlayerGameSeason +PlayerGameCareer,
  data = Vucevic,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10),
  na.action = na.exclude)

cv_model2
```

```
## Generalized Linear Model
##
## 624 samples
##   4 predictor
##   2 classes: 'FALSE', 'TRUE'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 560, 561, 561, 561, 561, 561, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.581106  0.1429171
```

0.13.2 Cross Validation Model 3

```
#K-fold Cross Validation Outside of a Model
set.seed(123)
cv_model3 <- train(
  DoubleDouble ~ TeamGameSeason +PlayerGameSeason +GameStarted +PlayerGameCareer +BackToBack,
  data = Vucevic,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10),
  na.action = na.exclude)

cv_model3
```

```
## Generalized Linear Model
##
## 624 samples
##   5 predictor
##   2 classes: 'FALSE', 'TRUE'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 560, 561, 561, 561, 561, 561, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.5953917  0.1761245
```

0.13.3 Compare Multiple Cross Validation Models

```
summary(
  resamples(
    list(
      model1 = cv_model1,
      model2 = cv_model2,
      model3 = cv_model3)))$statistics$Accuracy
```

```
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## model1 0.5000000 0.5341142 0.5645161 0.5729135 0.6104711 0.6612903    0
## model2 0.5079365 0.5341142 0.5967742 0.5811060 0.5967742 0.6666667    0
## model3 0.5000000 0.5541475 0.5887097 0.5953917 0.6386329 0.7096774    0
```

Looking at the mean of the 3 models the best preformance is from Model 3 with a 59% accuracy.

0.14 Create Confusion Matrix

```
Vucevic <- na.omit(Vucevic)

# predict class
pred_class <- predict(cv_model3, Vucevic, )

# create confusion matrix
confusionMatrix(
  data = relevel(pred_class, ref = "TRUE"),
  reference = relevel(Vucevic$DoubleDouble, ref = "TRUE"))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction TRUE FALSE
##      TRUE   147   106
##      FALSE  135   235
##
##                Accuracy : 0.6132
##                  95% CI : (0.5737, 0.6516)
##     No Information Rate : 0.5474
##     P-Value [Acc > NIR] : 0.0005254
##
##                   Kappa : 0.2123
##
##  Mcnemar's Test P-Value : 0.0712880
##
##             Sensitivity : 0.5213
##             Specificity : 0.6891
##          Pos Pred Value : 0.5810
##          Neg Pred Value : 0.6351
##              Prevalence : 0.4526
##          Detection Rate : 0.2360
##    Detection Prevalence : 0.4061
##       Balanced Accuracy : 0.6052
##
##        'Positive' Class : TRUE
##
```