



**Tyler Gunnell**

**MBA 614**

**VBA Final Project**

**4/13/2011**

## Executive Summary

Fantasy sports gaming is tremendously popular. In fact, in 2007, 29.9 million people aged 12 and older in the United States and Canada played fantasy sports<sup>1</sup>. I have played fantasy basketball for over ten years and enjoy the competition, researching players, and following player statistics. Because player stats are so carefully tracked and widely available on the internet, I realized that instead of randomly picking players each week to put into my lineup I could use these stats to estimate how players would perform on a weekly basis. In the past I have tried to put together complex spreadsheets and searched the internet for hours to find the information I needed. It always took a lot of time and I eventually stopped doing it because of the time commitment.

The model I have built utilizes player statistics from the internet that will predict the most likely outcome of a weekly matchup. The user will open the spreadsheet, enter their team information as well as opponent team information into user forms, click a button to download the latest player statistics as well as weekly team schedules from the internet, and then click a button to calculate the projected score totals for both fantasy teams. These totals, as well as the final score, will then be shown to the user. If unsatisfied with the projected score, the user could then adjust their team lineup in the user form to see if a better outcome could be achieved.

## Implementation Documentation

When the user first opens the spreadsheet they will see the Fantasy Forecaster Dashboard with multiple buttons. The user then has the option to update any or all of the information and calculate the projected score for the week. The “Main” sheet is the only relevant sheet to the user and the other sheets are simply used to store the data.

YrMoDay Next Fri Last Fri  
Friday Date: **110325** 110401 110325

**Your Team**

**Opponent**

**Refresh Data**

**Calculate Score**

**FG % FT% 3PTM PTS REB AST ST BLK TO**

**Your Team**

**Opponent**

**Final Score**

Main Team Info Data Schedule Week

<sup>1</sup> [http://en.wikipedia.org/wiki/Fantasy\\_sport](http://en.wikipedia.org/wiki/Fantasy_sport)

## Web Queries:

The first step in the project was to identify websites that contained the information I needed for the model. I found all the player stats I needed from Yahoo Sports and the weekly NBA team schedules from ESPN. Using a standard web query I was able to download all relevant statistics from Yahoo and put them into a sheet labeled "Data". The ESPN site was a little trickier to handle. Every Friday a columnist will update the team schedules for the upcoming week of games for the season. After this is done, the web address changes to the Friday date of when this update was completed. To solve this problem I have a cell on the "Main" sheet where the user will input the most recent Friday date. Then, when the macro runs, this date is put into the VBA code of the web address and the correct schedule is found and the information downloaded.

### User Input of Friday date on "Main" Sheet:

	YrMoDay	Next Fri	Last Fri
Friday Date:	110325	110401	110325

### VBA Code:

```
With ActiveSheet.QueryTables.Add(Connection:= _
    "URL;http://sports.espn.go.com/fantasy/basketball/fba/story?page=nbaforecaster" & Range("A1").value _
    , Destination:=Range("$A$2"))
```

This information is then put into the sheet labeled "Schedule". There was one other problem I encountered once the schedule had been downloaded from the web. The number of games for each team was formatted in such a way that Excel treated it as a date. So as part of the sub procedure I utilized the "Text to Columns" feature to get the correct formatting.

```
Range("I2:I31").Select
Selection.TextToColumns Destination:=Range("I2"), DataType:=xlFixedWidth, _
OtherChar:="/", FieldInfo:=Array(Array(0, 1), Array(1, 1)), _
TrailingMinusNumbers:=True
Selection.NumberFormat = "0"
```

## Sorting the Data:

The next step was to pull out all of the relevant data from these internet pages and sort it appropriately. Each player name as it came in from Yahoo included a single space before the name started. So in order to delete the single space before the player name I created a sub procedure that would select the entire column and remove any spaces.

```
Sub Trim()

Dim r As Range
Set r = Intersect(Range("A1").EntireColumn, ActiveSheet.UsedRange)
r.value = Evaluate("IF(ROW(" & r.Address & "),IF(" & r.Address & "<>\"\"\",TRIM(" & r.Address & "),\"\"\"))")

End Sub
```

Due to the significant number of total NBA players, using the “Find” feature in a macro (like our Fallen Angel assignment) would not have been an efficient way to handle the data. In order to correctly link the player information in the “Team Info” sheet I used the Excel VLOOKUP function to bring in player statistics. All player information downloaded from Yahoo Sports was put into the “Data” sheet and I gave it a named range so that the VLOOKUP function would be able to find the relevant player stats.

J3		fx		=VLOOKUP(\$B3,Data,6,FALSE)					
A	B	C	F	G	H	I	J	K	L
Your Team							FG		
		Team	GP	Min		M	A	Pct	
PG	Kyle Lowry	4 HOU	71	33.8		4.6	10.7	43.3	
SG	Carlos Delfino	4 MIL	42	33.5		4.4	11.0	40.2	
G	Josh Smith	3 ATL	73	34.9		6.5	13.7	47.7	

## User Forms:

After having all applicable data successfully brought into the worksheet, the focus was now on the user being able to input their team and opponent player information. I created two separate user forms that were almost identical, the only difference being one for the user’s team and the other for their opponent’s team. I linked each user form to its own button on the “Main” sheet. The user form shows the position, player name, team, and number of games for the week.

## Buttons:

Your Team

Opponent

## User Team Entry Form:

Team Entry Form

YOUR TEAM

	PLAYER NAME	TEAM	GAMES
PG:	Kyle Lowry	HOU	4
SG:	Carlos Delfino	MIL	4
G:	Josh Smith	ATL	3
SF:	Nene Hilario	DEN	3
PF:	Al Horford	ATL	3
F:	Russell Westbrook	OKC	4
C:	Dorell Wright	GS	3
Util 1:	Chris Paul	NO	3
Util 2:	Jason Richardson	ORL	4
Total Games			31
Bench 1:	Carlos Boozer	CHI	4
Bench 2:	Danilo Gallinari	DEN	3
Bench 3:	Marc Gasol	MEM	3

Close

Save

## Opponent Team Entry Form:

Opponent Entry Form

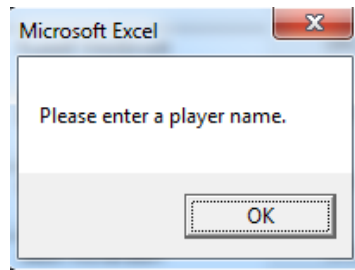
OPPONENT

	PLAYER NAME	TEAM	GAMES
PG:	Chauncey Billups	NY	3
SG:	Jordan Crawford	WAS	4
G:	Carmelo Anthony	NY	3
SF:	Dirk Nowitzki	DAL	4
PF:	Samuel Dalembert	SAC	4
F:	Darren Collison	IND	4
C:	Gerald Wallace	POR	4
Util 1:	J.J. Hickson	CLE	4
Util 2:	Channing Frye	PHO	4
Total Games			34
Bench 1:	Paul Millsap	UTA	3
Bench 2:	Greg Monroe	DET	3
Bench 3:	DeMar DeRozan	TOR	3

Close

Save

The team player information is stored in the “Team Information” sheet and as explained previously, all player stats are linked correctly through the VLOOKUP function. If no player name is entered for a position a message box will appear informing the user to enter one.



## Score Calculation:

The last step in the process was to find a way to calculate the value of each fantasy team's weekly total in each category and then tell the user the projected final score of the matchup. For each category a "W" will be shown for a Win or an "L" for a Loss. The highest number in eight of the nine scoring categories will return a "W" (the last category "TO" is the opposite with the lowest number getting a "W") I started by giving each category a named range in order to shorten the amount of code I would need to write.

```
Set games = Sheets("Team Info").Range("C3:C14")
Set fgM = Sheets("Team Info").Range("J3:J14")
Set fgA = Sheets("Team Info").Range("K3:K14")
Set ftM = Sheets("Team Info").Range("T3:T14")
Set ftA = Sheets("Team Info").Range("U3:U14")
Set threePt = Sheets("Team Info").Range("O3:O14")
Set pnts = Sheets("Team Info").Range("AI3:AI14")
Set reb = Sheets("Team Info").Range("AA3:AA14")
Set assists = Sheets("Team Info").Range("AD3:AD14")
Set steals = Sheets("Team Info").Range("AF3:AF14")
Set blk = Sheets("Team Info").Range("AG3:AG14")
Set turnOv = Sheets("Team Info").Range("AE3:AE14")
```

With these named ranges I was then able to calculate the values for each category and team.

```
Sheets("Main").Range("J4").value = Application.WorksheetFunction.SumProduct(games.value, threePt)
Sheets("Main").Range("K4").value = Application.WorksheetFunction.SumProduct(games.value, pnts)
Sheets("Main").Range("L4").value = Application.WorksheetFunction.SumProduct(games.value, reb)
Sheets("Main").Range("M4").value = Application.WorksheetFunction.SumProduct(games.value, assists)
Sheets("Main").Range("N4").value = Application.WorksheetFunction.SumProduct(games.value, steals)
Sheets("Main").Range("O4").value = Application.WorksheetFunction.SumProduct(games.value, blk)
Sheets("Main").Range("P4").value = Application.WorksheetFunction.SumProduct(games.value, turnOv)
```

The values are then displayed on the "Main" sheet of the workbook.

	<b>FG %</b>	<b>FT%</b>	<b>3PTM</b>	<b>PTS</b>	<b>REB</b>	<b>AST</b>	<b>ST</b>	<b>BLK</b>	<b>TO</b>
<b>Your Team</b>	0.467	0.797	39.2	496.6	174.1	141.6	45.0	17.5	63.4
<b>Opponent</b>	0.452	0.818	31.2	511.2	201.6	85.8	27.7	22.1	67.9

The second part of the code will then compare these values and return a “W” or “L” for each category.

```
Sub winOrloss()
    Dim i As Long
    Dim begCel As Long
    Dim tvar1 As Double
    Dim tvar2 As Double

    i = 1
    begCel = 8

    Do While i <= 8
        tvar1 = Sheets("Main").Cells(4, begCel)
        tvar2 = Sheets("Main").Cells(6, begCel)

        If tvar1 > tvar2 Then
            Sheets("Main").Cells(8, begCel) = "W"
        Else
            Sheets("Main").Cells(8, begCel) = "L"
        End If

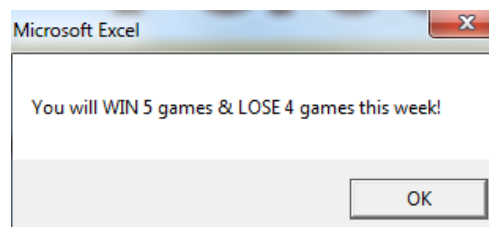
        tvar1 = 0
        tvar2 = 0
        begCel = begCel + 1
        i = i + 1
    Loop

    ' this is for the Turnovers column Only
    If Sheets("Main").Range("P4") > Sheets("Main").Range("P6") Then
        Sheets("Main").Range("P8") = "L"
    Else
        Sheets("Main").Range("P8") = "W"
    End If
End Sub
```

Displayed on the “Main” sheet:

<b>Final Score</b>	W	L	W	L	L	W	W	L	W
--------------------	---	---	---	---	---	---	---	---	---

A message box then appears to tell the user the projected final score for the week.



## Learning and Conceptual Difficulties:

I ran into various difficulties while putting this project together. The first difficulty was finding websites that had the needed information without requiring a password. The intent

of my project was to create a model that allows anyone to use it without having to log-in to a specific fantasy website. I found two different websites that fulfilled these criteria and the needs of the project, but they displayed the information differently. For example, on ESPN the team information is referred to by the team name while on Yahoo Sports the team information is referred to by the city where the team is located. To overcome this I created a table that converted abbreviated team city names to the full team names. By doing this the necessary information could be used from the two different websites.

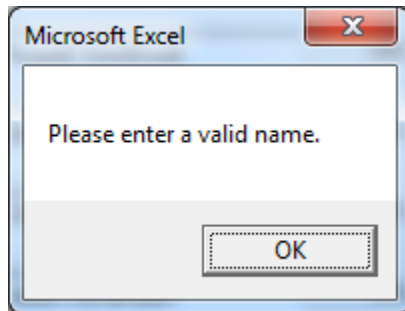
Originally I had intended to create a drop down list for each player position so that the user could select their players from a drop down list instead of having to type in each player's name. However, in order to get all player information from one web page I had to do a query of all NBA players on Yahoo Sports. By doing this, I lost all associated player position information when the data is brought into the model. I decided that this was a better solution than having to download multiple web pages into the model and sorting all the data. Had I done this it would have created additional problems because many players are listed as eligible to play multiple positions thereby producing duplicate entries.

Another difficulty I had was in getting the data to be calculated correctly. After trial and error and searching the internet I was able to figure out that I could utilize the SumProduct Excel function in my VBA code to do exactly what I wanted. Also, there are three players (Bench players) whose stats won't be used in calculating the final scores. I wanted to show these players in the user form with the number of games they had that week so that the user could decide accordingly who to play, while also not allowing their stats to be used in the calculation of the weekly final score.

The last issue that was difficult to solve was the spelling of player names. Each player name had to be spelled exactly as it appeared on the Yahoo Sports website or it would cause a run time error in the VBA code. I tried numerous things to try and solve the issue and was finally able to do it with the following code.

```
Sub checkNamePG()  
  
    Dim r As Integer  
    Dim found As Boolean  
  
    found = False  
  
    For r = 175 To 630  
  
        If frmYourTeam.txtPG.value = Sheets("Data").Cells(r, 1).value Then  
  
            found = True  
  
        End If  
  
    Next r  
  
    If found = False Then  
  
        MsgBox ("Please enter a valid name.")  
  
    End If  
  
End Sub
```

I had to replicate this code for each position in order to get it to work the way I wanted. There is probably a better or more efficient way to do this, but this was the solution I was able to come up with. If the player name is not spelled correctly the user will see a message box telling them to go back and enter a valid name.



## Conclusion

This project was a great learning experience for me in using the concepts I learned in class to create a model that is useful to me. It has allowed me to shorten a task that used to take several hours a week, to now only a matter of minutes. The model is also accessible to anyone no matter what fantasy website they play on. I have sent my completed project to multiple friends to test and solicit their feedback. All of their responses have been very positive on how easy it is to use and how quickly they are able to see their matchup results.

I know there are many things I still don't understand or feel comfortable with in VBA, but I do think I will be able to take the things I have learned in this class to automate more tasks and be more efficient. One scarce resource everyone has is their time and as Benjamin Franklin famously said, "Time is money." Hopefully the VBA skills I have learned will help me in my career to become more efficient and successful.