Does NFL Spread Betting Obey the Efficient Market Hypothesis?

Johannes Harkins

May 2013

Abstract

In this paper I examine the possibility that NFL spread betting does not obey the strong form of the Efficient Market Hypothesis. Building off of Gray and Gray, I use a probit model to examine whether or not home-team advantage and underdog factors are accurately captured by odds makers' lines, and add team strength variables into the regression. I find that although chi-square tests reject the null hypothesis that NFL spread betting obeys the Efficient Market Hypothesis, the information not captured by the pre-game line is not important enough to exploit the market for real gains.

1 Introduction

In this project, I attempt to identify market inefficiencies in the NFL betting market, by examining the simple case of spread betting. For each NFL game, bookmakers set a spread or line that attempts to estimate by how much one team (usually the favored team) will beat the other. If the favored team wins by anything less than this margin, they have failed to beat the spread. For example, if someone bets on the Seattle Seahawks when are favored to beat the San Francisco 49ers by a margin of 5 points, and the final score is 35-10 in favor of Seattle, then Seattle has beaten the spread, and the bettor wins the bet. In most places where NFL betting is legal, a winning bet of $11 returns only $10 (the extra dollar is collected by the betting house to ensure a profit). Thus a bettor must in 52.38% of their bets in order to make a profit. A feature of this type of betting is that, if the spread set by sports books is accurately capturing all information, then any given team should have a 50% chance of beating the spread. This is equivalent to saying that the NFL point-spread betting obeys the strong form of the efficient market hypothesis. There are certain phenomena that are claimed to be useful in predicting whether or not teams will "beat" spreads set by bookmakers. . A common cliche in sports is the so-called "home-team advantage," which posits that teams perform better in home games than they otherwise would. Another is the idea that perhaps book makers overestimate the performance of favored teams, that is, they consistently set the spread too high. I attempt to examine both of these ideas in testing the efficiency of point-spread betting as a market.

2 Data

My data come from the website of Warren Repole, and were checked for correct scores and approximately accurate lines. The dataset tracks information for each NFL game since the 1978 season on the pre-game spread (in terms of the home team) and predicted total score of the game, the teams playing, and the actual score. Each observation represents a single matchup of NFL teams. This results in a total of 8174 observations over the 1978-2012 seasons. I generated variables for the actual score difference, as well as generating variables for all of these statistics using the favorite as the team to which the spread refers. The variables included in the dataset were:

```
. describe date visteam visscore hometeam homescore line totline spr_actual

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
date            str10  %10s                   Game Day dd/mm/yyyy
visteam         str20  %20s                   Visiting Team
visscore        byte   %10.0g                 Visitor´s Score
hometeam        str20  %20s                   Home Team
homescore       byte   %10.0g                 Home Score
line            double %10.0g                 Pre-Game Spread
totline         double %10.0g                 Pre-Game Total Score Line
spr_actual      float  %9.0g                  Home Score minus Visitor´s Score
```

In order to test the ideas that the bookmakers don't properly factor in home team advantage and over-value favorites, it is instructive to generate a few variables in order to compare them to the line set by book makers. First, I generated a fav variable for the favored team in each matchup. I then generated $line\_fav$ and spr_fav which correspond to the line set by the odds makers and the actual point difference of the game, but with the odds from the perspective of the favorite.

```
. summarize line spr_actual line_fav spr_fav

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        line |      8174    2.492904    5.891091      -18.5       24.5
  spr_actual |      8174    2.752997     14.6047        -51         59
    line_fav |      8174    5.397235    3.433158          0       24.5
     spr_fav |      8174    5.246758    13.90487        -45         59
```

These statistics show that the home team is winning, on average, by a larger margin than the spread predicts, and underdogs are losing by less than the spread predicts. I examine these potential market inefficiencies in the next section.

3 Model

In order to examine whether or not the spread accurately captures all available information, I begin by using the basic probit model in the 1997 paper by Phillip K. Gray Stephen F. Gray, who examined a similar question. This model estimates whether or not being the home team or underdog has an effect on the probability of the team beating the spread. I generated a random 0 or 1 for each matchup by creating a new variable and assigning it to the results of this random generation using the stata command **rbinomial(8174,0.5)**. I then created a new variable for a random team to which the spread refers, by using the home team if the random number was 0 and the away team otherwise. Thus the reference team should beat the spread about 50% of the time, and no information should be helpful in predicting whether or not this team beat the spread. I generated variables *beat_rand* for whether or not the favorite beat the spread (whether or not they lost by less than predicted or won by more) and ran a probit regression of *beat_rand* on the dummy variables for the reference team being the home team or underdog. The probit regression uses a maximum likelihood method to estimate the effect of each variable on the probability that the reference team beats the spread given information about that team. The matrix for the values of the independent variables is written as in the simple case of regression:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1_1} & x_{2_1} & \dots & x_{q_1} \\ \vdots & \ddots & \vdots & & \vdots \\ 1 & x_{1_i} & x_{2_i} & \dots & x_{q_i} \\ \vdots & & \vdots & \ddots & \vdots \\ 1 & x_{1_N} & x_{2_N} & \dots & x_{q_N} \end{bmatrix} \tag{1}$$

Similarly, the column vector for the coefficients is written:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \tag{2}$$

The linear probability model would be written:

$$\Pr(\mathbf{Y} = 1 \,|\, \mathbf{X}) = \mathbf{X}\boldsymbol{\beta} \tag{3}$$

Where $\mathbf{Y}$ is the dependent variable indicating whether or not the random reference team beat the spread in each matchup. The probit model however, assumes the functional form of

$$\Pr(\mathbf{Y} = 1 \,|\, \mathbf{X}) = \phi(\mathbf{Z}) \tag{4}$$

With ϕ being the cumulative distribution function of the normal distribution and $\mathbf{Z} = \mathbf{X}\boldsymbol{\beta}$. In my first probit regression, z is specified with relatively few regressors as:

$$z_i = \beta_0 + \beta_1 home\_yn_i + \beta_2 und\_yn_i \tag{5}$$

In order for the NFL betting market to satisfy the Efficient Market Hypothesis, no variable can have a significant effect on whether or not the reference team beats the spread, since, if the pre-game line captured all the information. The results of the probit regression are shown in Table 1: The results of this regression indicate that the effect of the reference team being the underdog on whether or not the team beats the spread is statistically significant , and the effect of being the home team is not. There is reason to believe, then, that there is a market inefficiency in NFL betting. The coefficients themselves, however, are not immediately interpretable, since they represent the marginal effect of a change in x_i

Table 1: Results of probit regression with simple model

	(1)
VARIABLES	beat_rand
home_yn	0.0302
	(0.0298)
und_yn	0.1000***
	(0.0298)
Constant	-0.115***
	(0.0282)
Observations	8,174

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

on z_i rather than $\Pr(y_i = 1)$. Using the **dprobit** command on the same variables returns coefficients that represent the estimated marginal effects of the variables on $\Pr(y_i = 1)$, which is more readily interpretable. The results of the **dprobit** regression, shown in Table 2, indicate that being the underdog increases the reference team's chances of beating the spread by around 3.98%, which is significant at the .001 level. The (same) chi-square statistic of .0035 reported by both regressions rejects at the 5% significance level the null hypothesis that neither variable has an effect. Since the Efficient Market Hypothesis states that the pre-game spread contains all available information on the outcome, to reject the null hypothesis than all of the coefficients are zero is to reject the null hypothesis that NFL betting obeys the Efficient Market Hypothesis. This result confirms the result in Gray and Gray, whose data go through 1994. The coefficients also have the corresponding signs and similar magnitudes. Here, goodness-of-fit is of less concern than the ability to correctly predict outcomes using the model, so the diagnostic I performed for this model was to test the number of predictions correctly made by the model about whether or not the reference team would beat the spread. In order to asses the accuracy of the model's predictions, I generated a variable for the predicted probability of beating the spread using **predict pr_br**. Then, I took all values of *pr_br* greater than .5 and considered them a prediction of 1, and anything under .5 to be a prediction of 0. Finally I generated a dummy variable indicating if my model had predicted correctly or not, and summarized it to obtain its mean. The resulting mean of .5198 indicates that the model correctly predicts if a team will beat the spread 51.98% of the time. This is better predictive power than guessing, but not by much, and it doesn't meet the threshold required to break even. In order to try to test the possibility that other

Table 2: Marginal effects for simple probit model

	(1)
VARIABLES	beat_rand
home_yn	0.0120
	(0.0119)
und_yn	0.0398***
	(0.0118)
Observations	8,174

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

factors may influence the probability that a team beats the spread, I generated a variable for the pre-game point differential of the reference team (their points for minus their points against for all games up to that point) as well as the same statistic for the team opposing the reference team.[1] This is a potential measure of team strength that could indicate if sports books are improperly accounting for team strength, and if these measures are a reliable basis for a betting strategy. Maintaining the probit functional form and adding the new variables resulted in the following model:

$$z_i = \beta_0 + \beta_1 home\_yn_i + \beta_2 und\_yn_i + \beta_3 rpd_i + \beta_4 opd_i \tag{6}$$

Where rid is the reference team's point differential going into the game in question, and pod is the opponent's point differential to date. For this model, I opted to drop the first game

[1] See appendix

of every season, since the beginning point differentials are always 0, and give no predictive power. I performed probit and dprobit regressions using the new model, as well as the old model for comparison over the same observations. The outputs of the dprobit regressions are shown in Table 3. As the table shows, the effects of these two team strength variables on the probability of a team beating the spread are insignificant, but not as close to zero as they look, since a good team can finish a season with a +200 point differential, which equates to little less than a percent change in probability. This leads to the conclusion that perhaps, at

Table 3: Results of dprobit regression on first two models

	(1)	(2)
VARIABLES	beat_rand	beat_rand
home_yn	0.0103	0.00991
	(0.0123)	(0.0129)
und_yn	0.0397***	0.0383**
	(0.0122)	(0.0153)
rpd		-8.82e-05
		(0.000109)
opd		-6.35e-05
		(0.000109)
Observations	7,654	7,654

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

least by this measure of team strength, sports books do a pretty good job of capturing the relative strengths of teams in setting the betting line. I also experimented with the addition of interaction variables, running the same regression with the following model:

$$\begin{aligned} z_i = & \beta_0 + \beta_1 home\_yn_i + \beta_2 und\_yn_i + \beta_3(home\_yn_i \times und\_yn_i) + \beta_4 rpd_i + \beta_5 opd_i \\ & + \beta_6(home\_yn_i \times rpd_i) + \beta_7(home\_yn_i \times rpd_i) + \beta_8(und\_yn_i \times rpd_i) + \beta_9(und\_yn_i \times opd_i) \end{aligned} \tag{7}$$

Table 4 shows the comparison of the three models. There are some significant coefficients

Table 4: Comparison of 3 probit model specifications

VARIABLES	(1) beat_rand	(2) beat_rand	(3) beat_rand
home_yn	0.0103	0.00991	0.0384*
	(0.0123)	(0.0129)	(0.0214)
und_yn	0.0397***	0.0383**	0.0671***
	(0.0122)	(0.0153)	(0.0230)
int_homeund			-0.0562*
			(0.0339)
rpd		-8.82e-05	-0.000106
		(0.000109)	(0.000228)
opd		-6.35e-05	-0.000453**
		(0.000109)	(0.000216)
int_homerpd			0.000118
			(0.000237)
int_homeopd			0.000546**
			(0.000232)
int_undrpd			-8.92e-05
			(0.000237)
int_undopd			0.000234
			(0.000233)
Observations	7,654	7,654	7,654

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

for the third model. The interaction term for *opd* and *home_yn*, as well as the term interacting *home_yn* and *und_yn* are significant. The effects here are somewhat complicated to break down. Th third term being significantly negative implies that perhaps there is some sort of confounding factor between the home-team advantage and over-valued favorites. That is, odds makers don't properly take into account home teams or underdogs in setting the line, but they do a better job of accounting for underdog home teams. This actually lines up with a sports cliche often spoken along the lines of "maybe they can do it at home..", or "this is a tough place to play...". The coefficients on *opd* and *int_homeopd* are significant and opposite signs, which means that, given that the reference team is the favorite, the opposing team's point differential drops the probability of the reference team beating the spread by about 4% for every 100 positive points, and that when the reference team is at home, the effect of the opposing team's point differential is nearly cancelled out. I performed similar generation of variables, using the predicted probabilities two models, that allowed me to calculate success of predictions. Summarizing the three variables indicating a correct prediction by each model gives the proportion predicted correctly by this model as the mean, as shown below.

Table 5: Proportion predicted correctly by 3 models

Variable	**Mean**	**Std. Dev.**
nocorrect	0.519	0.5
nocorrect2	0.519	0.5
nocorrect3	0.523	0.5
N	7654	

The end result is that no model predicts correctly often enough to break even given the proportion needed as a result of the 11 for 10 rule. There is however, the possibility of modifying the betting strategy, while still using the models described above. Say instead of betting in favor of the reference team when the model predicts the probability to be above .5 and betting against otherwise, you picked a different cutoff, or even a range in which not to bet. I attempted to implement some of these strategies on the sample to test whether or not these would be more successful than the simple strategy to always bet based on the .5 cutoff mark. I selected 3 strategies using either my first or last model from above and implemented them using .do files[2] to generate variables that recorded whether or not a bet won. A comparison of the models is shown in Table 6 using summary statistics of generated variables indicating bets won, lost and not placed using different strategies. The first was to

Table 6: Comparison of betting strategies with win percentage as mean

Variable	**Mean**	**Std. Dev.**	**N**
win1_1	0.518	0.5	7654
win1_3	0.524	0.499	7654
win2_1	0.496	0.5	1243
win2_3	0.541	0.498	1743
win3_3	0.553	0.497	1959

simply use the mean of the predicted probability of beating the spread as the cutoff point. I did this for both models, and found that the more third, more complex model, did slightly better, and was technically turning a profit, winning 52.42 % of its bets. Next I tried using cutoff points for betting. I created a variable based on the estimated probability of beating the spread being over .50 and less than .425. Using the first model, this resulted in a wining percentage of .4963, but the second model resulted in a winning percentage of .5410. It's worth noting that using these cutoffs makes the number of observations different for each variable representing whether or not the bet won, and I considered a reasonable number of bets to be a required category for a good strategy, since the margins were so thin, and a

[2] See appendix

betting strategy that only bets once a season isn't much of a realistic strategy at all. Finally, I tried to fine tune the cutoff strategy using the third model, and came up with a winning percentage of .5533 by implementing a strategy using the third model where the bettor bets against the reference team if the model predicts lower than a .465 probability of beating the spread and bets for the reference team if the model predicts a .520 or higher probability of beating the spread, placing no bet otherwise. Though the winning percentage on this strategy is better than the rest, it only bets ,on average, on around a quarter of the games in an NFL season. This means the expected value of this strategy for $10 bets throughout the course of the season is only $40.26. [3] Placing larger bets could increase the reward, but puts you at risk to potentially lose more.

4 Conclusion

Using different specifications of a probit model to estimate the effect different variables on a random team beating the spread, I was able to reflect at the 5% level the null hypothesis that NFL spread betting obeys the strong form of the Efficient Market Hypothesis. However, using the predictions of these models is several different betting strategies, I was unable to come up with a specification for the model, coupled with a betting strategy, that exploited the inefficiency for any kind of serious gain. Given the $10 for $11 rule, the margins for profit from betting strategies quickly become insignificant. However, I do think that there's

[3]Dividing the number of observations by the total number of games gets the proportion of games that bets are placed on with this strategy: $1959/7654 = .2560$. We can use this proportion to calculate that someone employing this strategy would bet on in a 16 game NFL season with 32 teams: $.2560 \times 256$ total games = about 65 games per season. Since the expected value of betting using this strategy is $10 \times .5533 - 11 \times (1 - .5533) = .6129$ dollars per game, the expected value of an NFL season spent betting this way is $\$.6129 \times 65 = \$40.26$

room for more exploration in the area. I found creating .do files for variables that changed for each team over the course of a season to be somewhat cumbersome and extremely time consuming, but given more time and willingness to take on some of the more tedious tasks involved in generating these variables, I would have liked to perhaps include some more measures of team strength, including average margin of victory, and more advanced football metrics, like the DVOA statistic from FootballOutsiders.com. It would also be interesting to see if parity measures enacted by the NFL to make the league more competitive, or time trends with rule changes have changed the ability of odds makers to correctly set the lines, or the predictive power of different variables.

Appendix

.do files

1. Generating rpd and opd:
   In order to generate running point totals for each team for each season, I used the following code in a Stata .do file:

```
qui {
forvalues i =1/39 {
local pd = 0
forvalues j = 673/896 {
if rteam_`i´[`j´] == 1 {

  if hteam_`i´[`j´] == 1 {
    replace rpd = `pd´ in `j´
    local pd = `pd´ + spr_actual[`j´]
      }
  else{
    replace rpd = `pd´ in `j´
    local pd = `pd´ - spr_actual[`j´] // arithmetic looks different if refteam == visteam
      }

}
else{
  if hteam_`i´[`j´] == 1 {
    local pd = `pd´ + spr_actual[`j´]
      }
  else if vteam_`i´[`j´] ==1 {
    local pd = `pd´ - spr_actual[`j´]
      }
   else{
      }
}
}
}
}
```

   The top loop over i executed for each team that existed at some point during the sample (in some cases it will do nothing since those teams doesn't appear in many seasons). Then the point differential resets to zero for the beginning of each season in the 3rd line. The range of values for j delimits a season(in this case the 1981 season). I had to do each season by hand, replacing the values delimiting the season. I'm sure there's a more elegant way to do it, but by the time I had this figured out, I was far enough in to go with it. This loop walks through each individual observation and replaces the value of *rpd* contingent on some conditions below that line. In the 5th line, *rteam_1* - *rteam_39* are indicator variables for which team the reference team is (similarly there are home and visiting team indicators used). I probably could have used a categorical variable instead. *pd* is a macro used as a running count of the point differential. In the first "**if hteam...**" clause and the else clause below, the order of the replace command and the adjustment of pd matters, since I wanted *rpd* to indicate a team's point differential prior to the matchup (observable for betting purposes). The code in the more general else loop in the bottom half is necessary to keep track of a team's point differential from weeks when they are not the reference team. The following observations from the full dataset mark the beginning of seasons: (1, 225, 449, 673, 897, 1023, 1247, 1471, 1695, 1919, 2087, 2311, 2535, 2759, 2983, 3207, 3431, 3655, 3895, 4135, 4375, 4615, 4863, 5111, 5359, 5615, 5871, 6127, 6383, 6639, 6895, 7151, 7407, 7663, 7919) I modified the code slightly and followed a similarly tedious method of going through individual seasons in order to generate *opd*. The .do file for that variable generation is shown below:

```
qui {
forvalues i =1/39 {
```

```
local pd = 0
forvalues j = 673/896 {
if rteam_`i´[`j´] == 1 {
  if hteam_`i´[`j´] == 1 {
    local pd = `pd´ + spr_actual[`j´]
     }
  else if vteam_`i´[`j´] == 1{
    local pd = `pd´ - spr_actual[`j´]
     }

}
else{
  if hteam_`i´[`j´] == 1 {
    replace opd = `pd´ in `j´
    local pd = `pd´ + spr_actual[`j´]
     }
  else if vteam_`i´[`j´] ==1 {
    replace opd = `pd´ in `j´
    local pd = `pd´ - spr_actual[`j´]
     }
   else{
     }
}
}
}
}
```

2. Implementing Betting Strategies:

   (a) Strategy 1_1
   Strategy 1 using model 1:

   ```
   gen bet1_1 = 0
   replace bet1_1 = 1 if  (p_br >= .4799)
   gen win1_1 = 0
   replace win1_1 = 1 if (bet1_1 == beat_rand)
   summarize win1_1
   ```

   (b) Strategy 1_3:
   Strategy 1 using model 3:

   ```
   gen bet1_3 = 0
   replace bet1_3 = 1 if  (pr_br3 >= .4796)
   gen win1_3 = 0
   replace win1_3 = 1 if (bet1_3 == beat_rand)
   summarize win1_3
   ```

   (c) Strategy 2_1
   Strategy 2 using model 1:

   ```
   gen bet2_1 = 0
   replace bet2_1 = 1 if  pr_br >=.5
   replace bet2_1 = -1 if pr_br <=.425
   gen win2_1 = 1 if ((bet2_1 == -1 & beat_rand == 0) | (bet2_1 == 1 & beat_rand == 1))
   replace win2_1 = 0 if ((bet2_1 == 1 & beat_rand ==0) |(bet2_1 == 1 & beat_rand == 0))
   summarize win2_1
   ```

   (d) Strategy 2_3
   Strategy 2 using model 3: gen bet2_3 = 0 replace bet2_3 = 1 if pr_br3 ¿=.5 replace bet2_3 = -1 if pr_br3 ¡=.425 gen win2_3 = 1 if ((bet2_3 == -1 & beat_rand == 0) — (bet2_3 == 1 & beat_rand == 1)) replace win2_3 = 0 if ((bet2_3 == 1 & beat_rand ==0) —(bet2_3 == 1 & beat_rand == 0)) summarize win2_3

   (e) Strategy 3_3
   Strategy 3 (using model 3):

```
gen bet3_3 = 0
replace bet3_3 = 1 if  pr_br3 >=.520
replace bet3_3 = -1 if pr_br3 <=.465
gen win3_3 = 1 if ((bet3_3 == -1 & beat_rand == 0) | (bet3_3 == 1 & beat_rand == 1))
replace win3_3 = 0 if ((bet3_3 == -1 & beat_rand ==1) |(bet3_3 == 1 & beat_rand == 0))
summarize win3_3
```

These methods all generated variables that had either 1's 0's or missing values, where a 1 corresponded to a bet won on that matchup, 0 to a lost bet, and a missing value to a bet not placed. The *pr_br* and *pr_br*3 variables are the probabilities of beating the spread predicted by each model and were generated using the **predict** command after each model's probit regression.

References

P.K. Gray, S.F. Gray, "Testing Market Efficiency: Evidence From the NFL Sports Betting Market." *The Journal of Finance* 52.4 (2001) 1725-1737.

Sunshine Forecast (2012). Regular Season Games Scores and Lines: NFL 2012. Retrieved 4/2/2013 from http://www.repole.com/sun4cast/stats/nfl2012lines.csv

Sunshine Forecast (2012). Regular Season Games Scores and Lines: NFL All Years from 1978 through 2011. Retrieved 4/2/2013 from http://www.repole.com/sun4cast/stats/nfl2012lines.csv