

Sports and statistics

Lecture 6: NFL, expected points, game theory

Goals

- i) Expected points
- ii) NFL game theory & decision making

Tools

- i) Expected value
- ii) Review of logistic regression

From last week:

```
fit_2 <- glm(Success ~ Distance + Grass + Year + GameMinute,  
             data = nfl_kick, family = "binomial")
```

```
tidy(fit_2)
```

```
A tibble: 5 x 5
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	-104.	17.3	-6.02	1.78e- 9
Distance	-0.105	0.00318	-33.0	1.84e-238
GrassTRUE	-0.155	0.0549	-2.82	4.75e- 3
Year	0.0548	0.00863	6.35	2.14e- 10
GameMinute	0.000289	0.00163	0.177	8.59e- 1

Summary: odds ratios, z-test statistics, predicted probabilities

Link to expected points:

Transform logistic regression model into probabilities

- `predict()` function (or by hand)
- Example:

$P(\text{Success} = 1 \mid \text{Dist} = 40, \text{Grass} = \text{TRUE}, \text{Yr} = 2010, \text{GM} = 10) = 0.82$

- Expected points: $E(X_i) = x_i * p_i$

$3 * 0.82 = 2.46$ points

Interpretation:

Link to expected points:

So kicking a field goal attempt is worth 2.46 points.
What else?

Other possible decisions:

- Go for it
- Punt it?

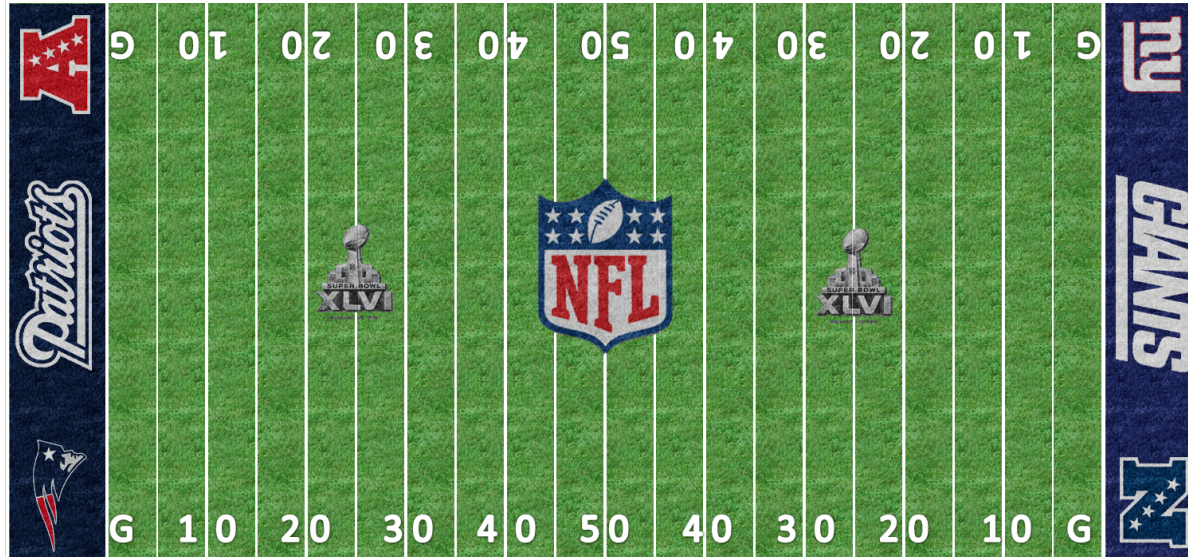
All possible outcomes:

- Successful fourth down conversion
- Successful field goal
- Missed fourth down conversion
- Missed field goal
- Disaster plays

Formal definition:

“Given any combination of down, yards to go, and distance from the end zone, the expected value of the points from that position are equal to the average of every next score from that position.” - Causey

Function of at least 3 play characteristics: down, distance, yard line.



Some probability:

Expectation of random variable X

$$E(x) = \sum_{i=1} x_i p_i = x_1 p_1 + x_2 p_2 + \dots$$

x	7	3	-2	-3	-7
p	0.4	0.2	<0.01	0.2	0.2

$$E(X) = 7(0.4) + 3(0.2) - 3(0.2) - 7(0.2) = 1.4$$

Expected value calculations

Point totals fixed – stated in terms of offense

- -7, -3, -2, 0, 2, 3, 7

Probabilities are the probability that each point total is the next one scored

- Conditional on **down, distance, yard line**
- Ex: First and 10 from own 20-yard line

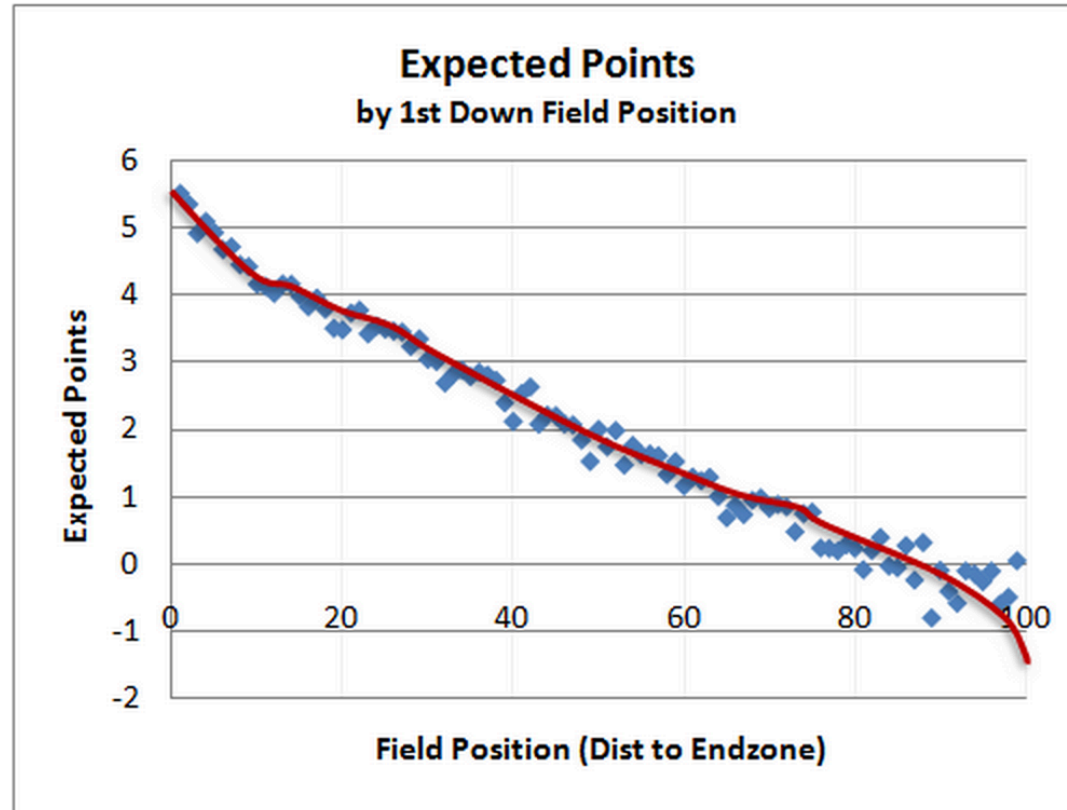
Expected value estimates

TABLE I
THE EXPECTED POINT VALUES OF POSSESSION OF THE FOOTBALL WITH FIRST
DOWN AND TEN YARDS TO GO FOR VARIOUS TEN-YARD STRIPS

Center of the ten-yard strip (yards from the target goal line): X	Expected point value: $E(X)$
95	-1.245
85	-0.637
75	+0.236
65	0.923
55	1.538
45	2.392
35	3.167
25	3.681
15	4.572
5	6.041

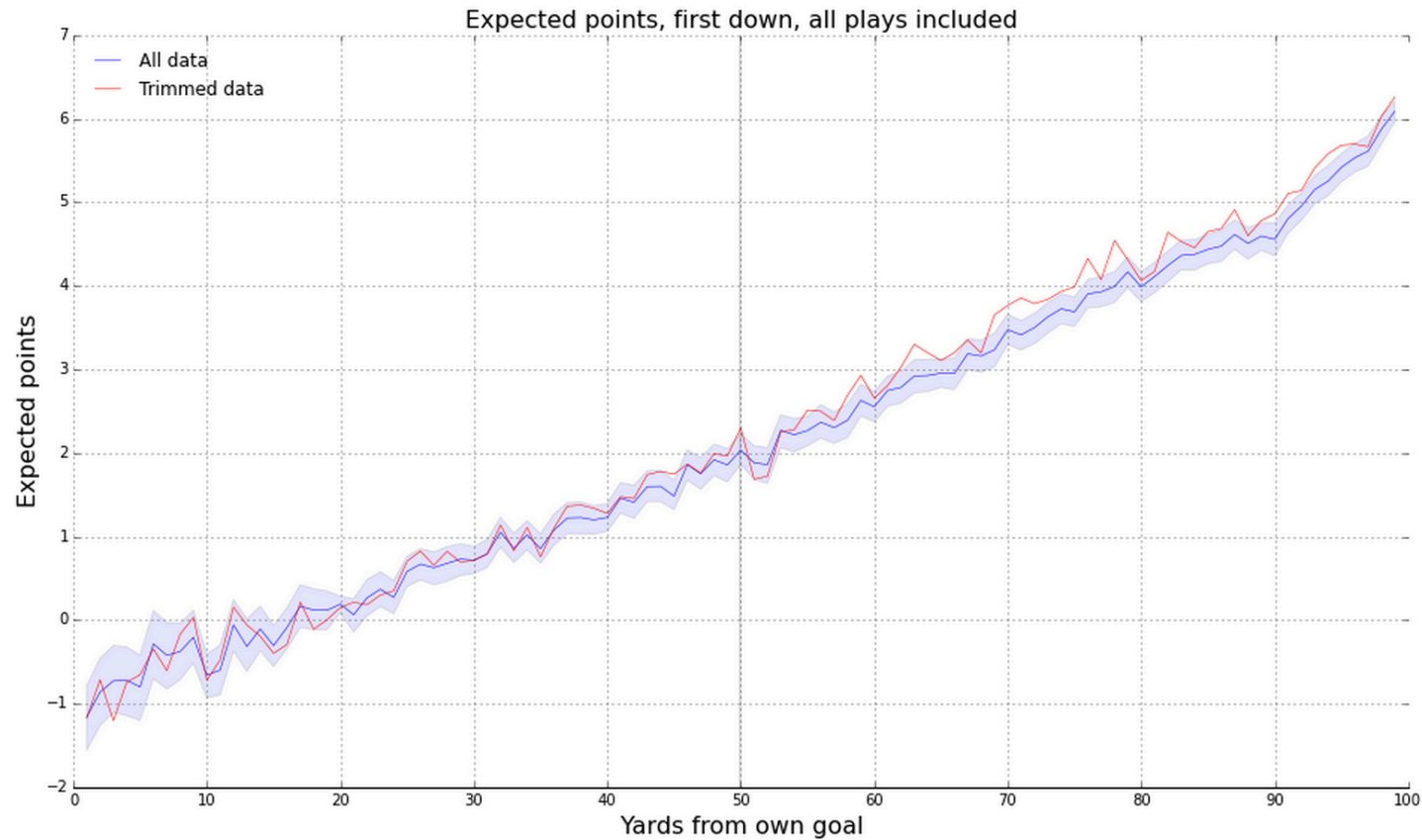
-[Carter/Machol](#), 1971

Expected value estimates



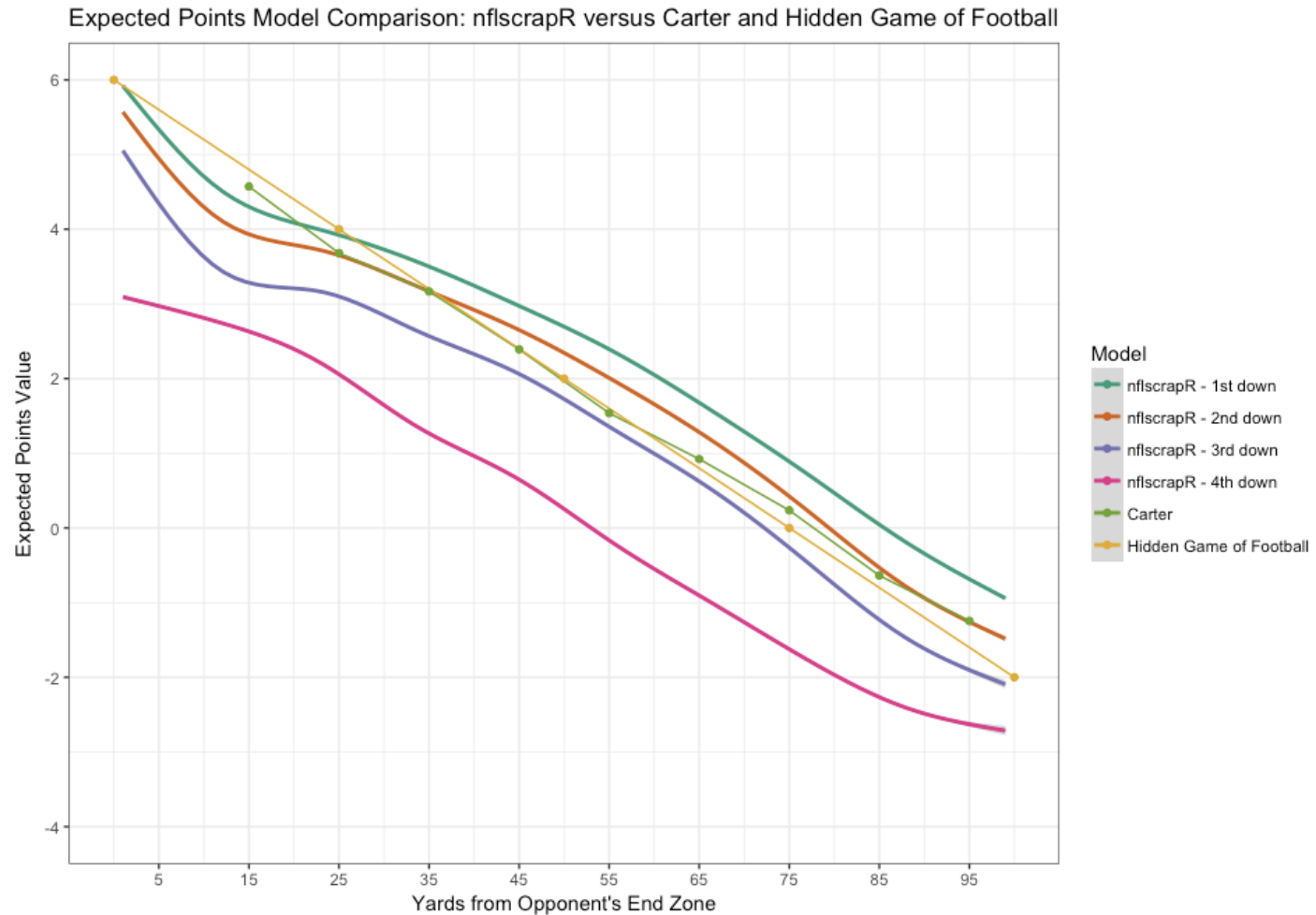
-[Burke](#), 2010

Expected value estimates



-[Trey Causey](#), 2015

Expected value estimates



-Yurko et al, 2019

Expected points and NFL decisions

Example:

- 4th - 2 from opponents 23-yard, $E(X) = 2.1$

How to calculate probabilities?

1:

2:

How to improve calculations?

1:

2:

Expected points and NFL decisions

- 4th - 2 from opponents 23-yard line

```
> round(tab.Fourth.2, 2)
```

	Decision	
pts.next	Go	for it Kick
-8	0.00	0.00
-7	0.09	0.06
-6	0.00	0.00
-3	0.07	0.05
2	0.01	0.00
3	0.23	0.81
6	0.04	0.00
7	0.37	0.04
8	0.01	0.00
<NA>	0.19	0.02

Go for it:

- $E(X) = 2.81$

Kick it:

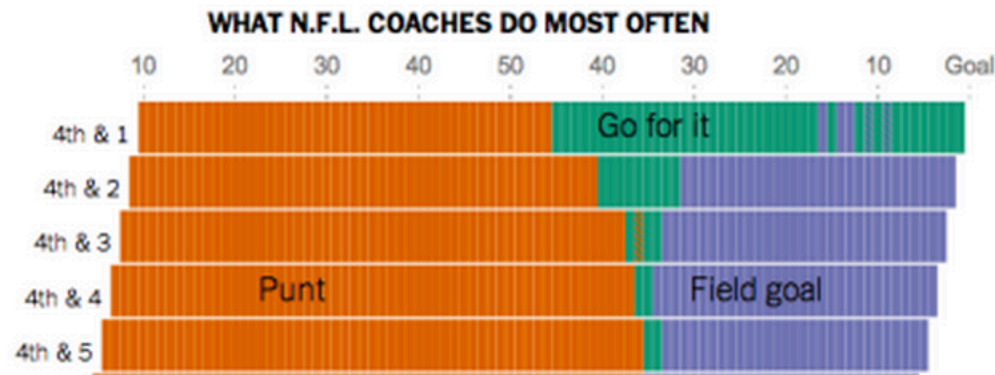
- $E(X) = 2.14$

What do coaches do?

Expected points and NFL decisions

Fourth and 2, opponents 23 yard line

- Going for it ~ 2.8 expected points
- Field goal ~ 2.1 expected points



- In reality, coaches kick field goal roughly twice as often in this example

Expected points and NFL decisions

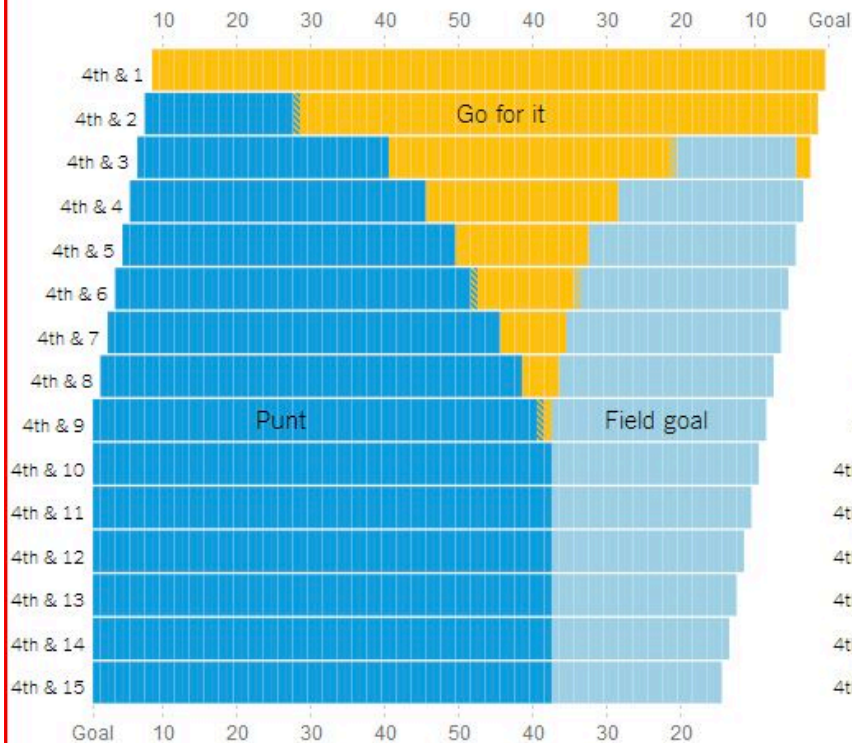
Details

- Quarters 1/3 & closer games receive priority
- Extensions: Expected points added
 - Ex 1: Kicker value
 - Ex 2: Play value
 - Ex 3: Play choice
 - Win probability models
- Weaknesses: Not all points created equally

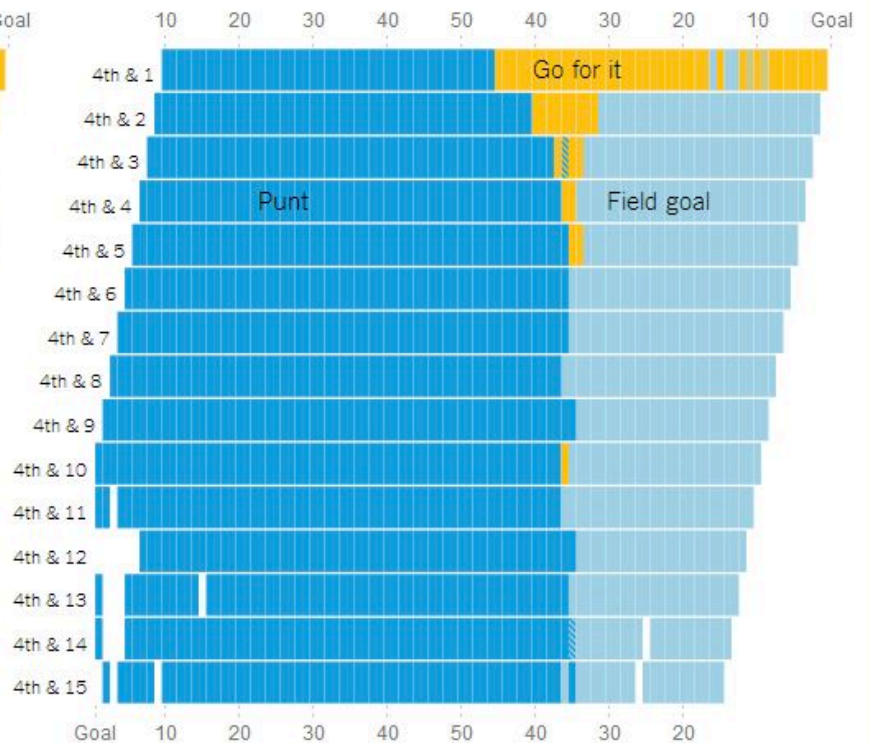
NFL Decisions

SEPT. 4, 2014

WHAT NYT 4TH DOWN BOT RECOMMENDS ON 4TH DOWN



WHAT N.F.L. COACHES DO MOST OFTEN



NFL Decisions

Reasons for varying strategy

- 1) Minimax
- 2) Prospect theory
- 3) Risk aversion

NFL Decisions: Minimax

Minimax: decision that minimizes possible loss in *worst case scenario*

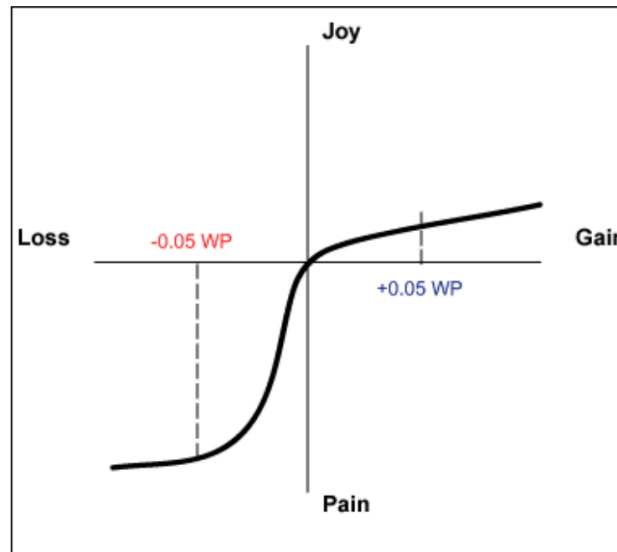
Ex: 4th and 2 from opponents 23

```
> round(tab.Fourth.2, 2)
```

	Decision	
pts.next	Go for it	Kick
-8	0.00	0.00
-7	0.09	0.06
-6	0.00	0.00
-3	0.07	0.05
2	0.01	0.00
3	0.23	0.81
6	0.04	0.00
7	0.37	0.04
8	0.01	0.00
<NA>	0.19	0.02

NFL Decisions: Prospect Theory

Prospect theory: humans make decisions based on value of losses and gains, and not on final outcome – and fear of losses outweighs equivalent gains.



Ex: Run versus pass

NFL Decisions: Risk aversion

Risk adverse: reluctance to accept bargain in favor of decision with more certain – but possibly lower - payoff

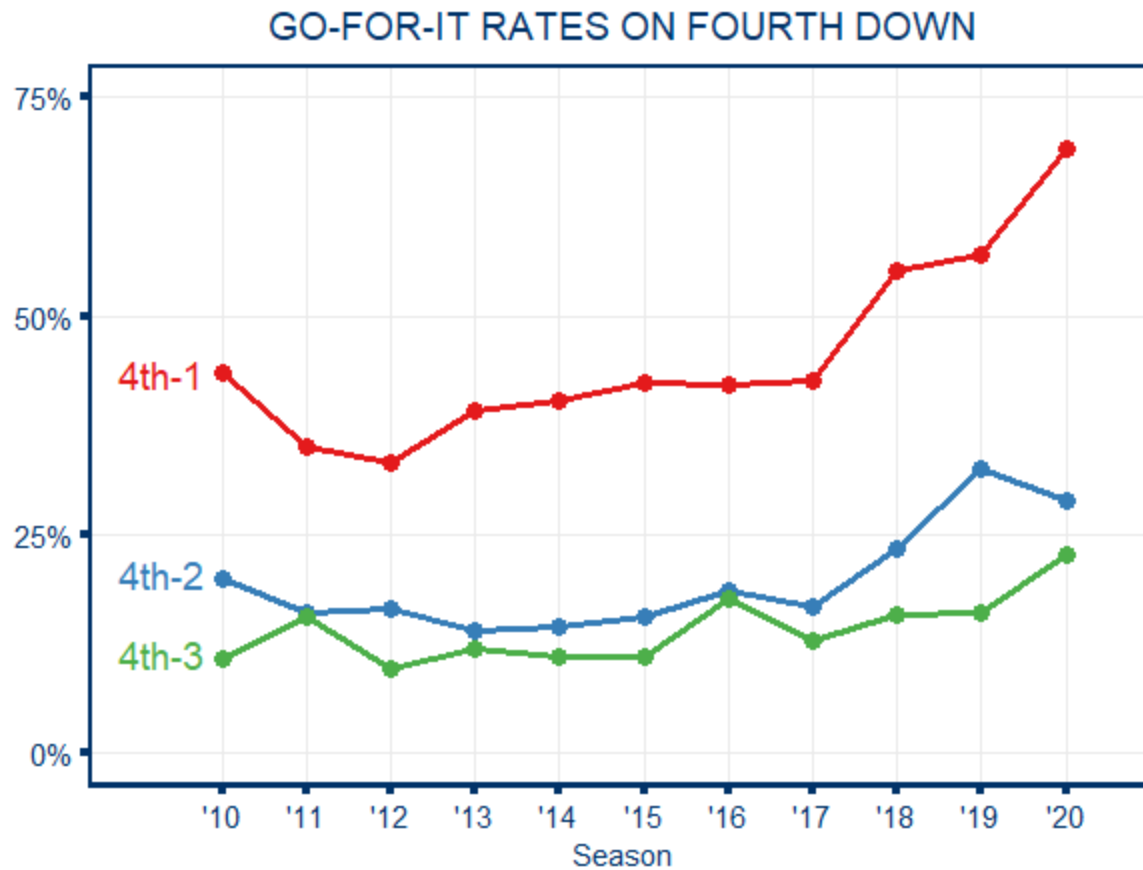
"Had we done that [gone for it] after what we had done to get down there and [not scored a touchdown], I can imagine what the critique would have been today about the play call." – Brian Billick

"You guys might very well be right that we're calling something too conservative in that situation. But what you guys don't understand is that if I make a call that's viewed to be controversial by the fans and the owner, and I fall, I lose my job" – Marvin Lewis

[What was Mike Smith thinking](#)

Ex: 4th and 2 from opponents 23

NFL Decisions:



<https://twitter.com/StatsbyLopez/status/1179426045432344576/photo/1>

Decisions in other sports

- 1) Minimax
- 2) Prospect theory
- 3) Risk aversion

nflscrapR/nflfastR

Introducing the `nflscrapR` Package

This package was built to allow R users to utilize and analyze data from the National Football League (NFL) API. The functions in this package allow users to perform analysis at the play and game levels on single games and entire seasons. By parsing the play-by-play data recorded by the NFL, this package allows NFL data enthusiasts to examine each facet of the game at a more insightful level. The creation of this package puts granular data into the hands of any `R` user with an interest in performing analysis and digging up insights about the game of American football. With open-source data, the development of reproducible advanced NFL metrics can occur at a more rapid pace and lead to growing the football analytics community.

Note: Data is only available after 2009... for now

nflscrapR/nflfastR

README.md

nflfastR

CRAN 3.0.0 build passing R-CMD-check passing Follow @nflfastR 4.8k

`nflfastR` is a set of functions to efficiently scrape NFL play-by-play data. `nflfastR` expands upon the features of `nflscrapR`:

- The package contains NFL play-by-play data back to 1999
- As suggested by the package name, it obtains games **much** faster
- Includes completion probability (`cp`), completion percentage over expected (`cpoe`), and expected yards after the catch (`xyac_epa` and `xyac_mean_yardage`) in play-by-play going back to 2006
- Includes drive information, including drive starting position and drive result
- Includes series information, including series number and series success
- Hosts [a repository of play-by-play data going back to 1999](#) for very quick access
- Features models for Expected Points, Win Probability, Completion Probability, and Yards After the Catch (see section below)
- Includes a function `update_db()` that creates and updates a database

We owe a debt of gratitude to the original `nflscrapR` team, Maksim Horowitz, Ronald Yurko, and Samuel Ventura, without whose contributions and inspiration this package would not exist.



nflfastR

```
library(nflfastR)
pbp_20 <- readRDS(url("https://raw.githubusercontent.com/guga31bb/nflfastR-data/master/data/play_by_play_2020.rds"))
dim(pbp_20)
[1] 11472    340
```

```
pbp_20 %>% select(game_id, posteam, play_id, down, ydstogo, yardline_100, home_score, away_score, desc, epa) %>%
  sample_n(4)
```

A tibble: 4 x 10

game_id	posteam	play_id	down	ydstogo	yardline_100	home_score	away_score	desc	epa
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>	<chr>	<dbl>
2020_03_W~	WAS	2790	1	10	29	34	20	(5:45) (No Huddle, Shotgun) 24~	0.799
2020_04_J~	JAX	78	2	6	71	33	25	(14:25) 30-J.Robinson left guar~	0.840
2020_02_M~	IND	1795	2	5	11	28	11	(2:00) 63-D.Pinter reported in ~	-0.612
2020_03_K~	KC	3891	2	7	66	20	34	(4:46) (Shotgun) 25-C.Edwards-H~	-0.563