

# nflMarkov: k of n

July 30, 2014

## Contents

<b>1</b>	<b>1 of n: a basic markov chain</b>	<b>2</b>
<b>2</b>	<b>2 of n: expectation values of a basic markov chain</b>	<b>3</b>
<b>3</b>	<b>3 of n: accounting for turnovers</b>	<b>6</b>
<b>4</b>	<b>4 of n: the yards gained distributions (transition matrix)</b>	<b>8</b>
<b>5</b>	<b>4 of n: the parameterization of the transition matrix</b>	<b>9</b>

# 1 1 of n: a basic markov chain

This is the first in a series of posts describing my work on a Markov chain tool to analyze football. The code is available from github, <https://github.com/bdilday/nflMarkov>

Before talking about football, I want to talk about a less complicated problem. It is probably worth saying at this point that I am not a mathematician, nor a computer scientist. What I am is an astronomer and astrophysicist, so obviously I know some math and some programming, but at the same time I'm not necessarily an expert on Markov chains. As with many things, what I know is what I've had to pick up in order to work on research problems, so what I have to say may be obvious, or naive, or even flat-out wrong. So if that seems to be the case, now you know the reason. In any case, writing this out is in large part for my own sake, just to see it all layed out and to help myself clarify my thoughts on the problem; as well as to hopefully be useful people who read this without much knowledge of Markov chains coming in.

The first simple problem I want to talk about, which is pretty much the simplest non-trivial Markov chain problem I can think of, is this; imagine a random walk where there are 5 possible states which I will call  $s_{-2}, s_{-1}, s_0, s_{+1}, s_{+2}$ . In each step of the process, the walker can move one step to the right, which happens with probability  $p$ , or one step to the left, which happens with probability  $q=1-p$ . If the walker lands at position  $+2$  or  $-2$  they stay there. Here is what the transition matrix looks like,

$$T_{ij} = \begin{pmatrix} 1 & 1-p & 0 & 0 & 0 \\ 0 & 0 & 1-p & 0 & 0 \\ 0 & p & 0 & 1-p & 0 \\ 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & p & 1 \end{pmatrix}$$

If I iterate once, i.e., compute  $T^2 = T_{ik}T_{kj}$ , I get,

$$T^2 = \begin{pmatrix} 1 & 1-p & (1-p)^2 & 0 & 0 \\ 0 & (1-p)p & 0 & (1-p)^2 & 0 \\ 0 & 0 & 2(1-p)p & 0 & 0 \\ 0 & p^2 & 0 & (1-p)p & 0 \\ 0 & 0 & p^2 & p & 1 \end{pmatrix}$$

Three times,

$$T^3 = \begin{pmatrix} 1 & p(1-p)^2 - p + 1 & (1-p)^2 & (1-p)^3 & 0 \\ 0 & 0 & 2(1-p)^2p & 0 & 0 \\ 0 & 2(1-p)p^2 & 0 & 2(1-p)^2p & 0 \\ 0 & 0 & 2(1-p)p^2 & 0 & 0 \\ 0 & p^3 & p^2 & (1-p)p^2 + p & 1 \end{pmatrix}$$

Four times,

$$T^4 = \begin{pmatrix} 1 & p(1-p)^2 - p + 1 & p(1-p)^3 + (p(1-p)^2 - p + 1)(1-p) & (1-p)^3 & 0 \\ 0 & 2(1-p)^2p^2 & 0 & 2(1-p)^3p & 0 \\ 0 & 0 & 4(1-p)^2p^2 & 0 & 0 \\ 0 & 2(1-p)p^3 & 0 & 2(1-p)^2p^2 & 0 \\ 0 & p^3 & (1-p)p^3 + ((1-p)p^2 + p)p & (1-p)p^2 + p & 1 \end{pmatrix}$$

You can stare at these and start to see the structure of how you move from state to state as you iterate the Markov chain, but I think the main usefulness is just to illustrate that repeated applications of the transition matrix mix things up in a deterministic and straight-forward (if tedious to enumerate) way. For a simple problem like this you could probably even come up with a closed form solution.

Now let me put in some numbers and keep iterating the transition matrix until it converges. lets try  $p=1-p=0.5$ . If I iterate 100 times, and Chop (i.e. round to zero in Mathematica nomenclature) values less than  $1 \times 10^{-6}$ , I get,

$$T^{100} \approx T^\infty = \begin{pmatrix} 1. & 0.75 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0.75 & 1. \end{pmatrix}$$

So what exactly does this mean? It means if I start in the state  $s_{-2}$ , I end in the state  $s_{-2}$  with probability 1. If I start in the state  $s_{-1}$  I end in the state  $s_{-2}$  with probability 0.75 and  $s_{+2}$  with probability 0.25. If I start in the state  $s_0$ , I end in the state  $s_{-2}$  with probability 0.5 and  $s_{+2}$  with probability 0.5. In other words, reading down each column (j) tells me the probability to end in the state (i) after 100 (which may as well be infinity) transitions.

If I use  $p=0.501$  instead, this is what I get,

$$\begin{pmatrix} 1. & 0.748498 & 0.498 & 0.248502 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.251502 & 0.502 & 0.751498 & 1. \end{pmatrix}$$

If I use  $p=0.99$ , I get,

$$\begin{pmatrix} 1. & 0.010101 & 0.00010202 & 1.02 \times 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.989899 & 0.999898 & 0.999999 & 1. \end{pmatrix}$$

So this first post just shows that you can set up a Markov chain that models a random walk and make the states at the end "sinks" (or roach motels; once you enter, you never leave), and after iterating enough times, you will always end up in one of the sinks. In the football application, the sinks are going to be scoring events. So we can imagine that making it all the way to the right is like scoring a touchdown, all the way to the left, like giving up a safety. In the next part I'll look at the expectation values associated with such a Markov chain.

## 2 2 of n: expectation values of a basic markov chain

In part 1 I talked about a simple random walk in 1-dimension, where the states all the way to the left and all the way to the right are sinks (or roach motels). The next step from that is to ask what is the expectation value associated with each state? I mentioned that we could think of the state  $s_{+2}$  as scoring a touchdown (+6 or +7 or +8 points) and  $s_{-2}$  as giving up a safety (-2 points), but to illustrate the problem, it is convenient to assume that landing at location +2 is worth +1 point and at location -2, -1 point.

As a concrete example, let me take  $p=0.6$  and ask what the expectation values are, given that I start in state  $s_{-1}$ ,  $s_0$ , or  $s_{+1}$ . Iterating the transition matrix to convergence, as I did in part 1, I get,

$$T^\infty = \begin{pmatrix} 1. & 0.584615 & 0.307692 & 0.123077 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.415385 & 0.692308 & 0.876923 & 1. \end{pmatrix}$$

This says that if I start at state  $s_0$ , I end in state  $s_{-2}$  with probability 0.308 and  $s_{+2}$  with probability 0.692. If these are worth -1 and +1 points, respectively, then the expectation value for points will be

$e_0 = 0.692 - 0.308 = 0.384$ . In a similar way I can see that the expectation values for states  $s_{-1}$  and  $s_{+1}$  are  $e_{-1} = 0.415 - 0.585 = -0.169$  and  $0.877 - 0.123 = 0.754$ , respectively. So this give me the answer using the first method.

The question I want to address now is, other than iterating to convergence, how do we go from the transition matrix (which is relatively easy to code up for football) to the expectation value (which is what we really want to know)? Logically, the expectation value,  $e_0$ , when starting from the state  $s_0$ , should be,

$$e_0 = e_{-2}p_{-2,0} + e_{-1}p_{-1,0} + e_0p_{0,0} + e_{+1}p_{+1,0} + e_{+2}p_{+2,0}$$

in words, its the expectation value of the subsequent state in the chain (for example,  $e_{-1}$ ), multiplied by the probability to transition to that state (for example  $p_{-1,0}$ ). Inspection shows that this says (using Einstein notation for matrix/vector multiplication),

$$e_i = e_k T_{k,i}$$

or, in vector notation,

$$\vec{e} = \vec{e} \vec{T},$$

or

$$\vec{e} (\vec{T} - \vec{I}) = 0, \text{ where } \vec{I} \text{ is the identity matrix.}$$

for the particular transition matrix we are discussing, we have,

$$\begin{pmatrix} e_{-2} \\ e_{-1} \\ e_0 \\ e_{+1} \\ e_{+2} \end{pmatrix} = \begin{pmatrix} e_{-2} \\ e_{-1} \\ e_0 \\ e_{+1} \\ e_{+2} \end{pmatrix} \times \begin{pmatrix} 1 & 1-p & 0 & 0 & 0 \\ 0 & 0 & 1-p & 0 & 0 \\ 0 & p & 0 & 1-p & 0 \\ 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & p & 1 \end{pmatrix}$$

Because  $e_m T_{m,n} = T_{m,n} e_m = T'_{n,m} e_m$ , where  $T'$  is the transpose of  $T$ , I can also write this as,

$$\vec{e} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1-p & 0 & p & 0 & 0 \\ 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 1-p & 0 & p \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \vec{e}$$

But the values  $e_{-2}$  and  $e_{+2}$  aren't really variables, i.e. quantities that need to take on particular values for the set of equations to hold; they are parameters such that their values are arbitrary, and once chosen, impact the expectation values for the non-sink states.

Let me denote by (s, t) states that are **sinks**, and (n,m) states that are **NOT** sinks. What I really want to know are the set of values  $e_n$ , i.e. the expectation values for states that are **not** sinks. So I can write,

$$e_n = e_s T_{s,n} + e_m T_{m,n}$$

Rearranging this equation gives,

$$e_m (T_{m,n} - \delta_{m,n}) = -e_s T_{s,n}$$

or in vector notation,

$$\vec{e}^n (\vec{T}^n - \vec{1}) = -\vec{e}^s \vec{T}^s$$

Taking the transpose of both sides I get,

$$(\vec{T}^n - \vec{1})' \vec{e}^n = -\vec{T}'^s \vec{e}^s$$

Note that the superscripts here denote sink or non-sink and **do not** denote exponentiation. Also, let  $S$  denote the number of sinks and  $N$  the number of non-sinks; then the dimensions of  $e^n, e^s, T'^n$ , and  $T'^s$  are  $N \times 1, S \times 1, N \times N$ , and  $N \times S$ , respectively.

For the particular problem we've been discussing, this can be expanded as,

$$\begin{pmatrix} 0 & p & 0 \\ 1-p & 0 & p \\ 0 & 1-p & 0 \end{pmatrix} \times \begin{pmatrix} e_{-1} \\ e_0 \\ e_{+1} \end{pmatrix} = - \begin{pmatrix} 1-p & 0 \\ 0 & 0 \\ 0 & p \end{pmatrix} \times \begin{pmatrix} e_{-2} \\ e_{+2} \end{pmatrix}$$

or

$$\begin{pmatrix} 0 & p & 0 \\ 1-p & 0 & p \\ 0 & 1-p & 0 \end{pmatrix} \times \begin{pmatrix} e_{-1} \\ e_0 \\ e_{+1} \end{pmatrix} = - \begin{pmatrix} e_{-2}(1-p) \\ 0 \\ e_{+2}p \end{pmatrix}$$

Note that once I choose values for  $e_{-2}$  and  $e_{+2}$ , the the right hand side is just a constant vector.

$$\begin{pmatrix} 0 & p & 0 \\ 1-p & 0 & p \\ 0 & 1-p & 0 \end{pmatrix} \times \begin{pmatrix} e_{-1} \\ e_0 \\ e_{+1} \end{pmatrix} = \begin{pmatrix} (1-p) \\ 0 \\ -p \end{pmatrix}$$

The formal solution is,

$$\begin{aligned} & ((\vec{T}^n - \vec{1})')^{-1} \cdot (-\vec{T}'^s \cdot \vec{e}^s) \\ &= \begin{pmatrix} \frac{-e_{-2}p^3 + e_{+2}p^3 + 2e_{-2}p^2 - 2e_{-2}p + e_{-2}}{2p^2 - 2p + 1} \\ \frac{e_{-2}(p-1)^2 + e_{+2}p^2}{2p^2 - 2p + 1} \\ \frac{e_{+2}p(p^2 - p + 1) - e_{-2}(p-1)^3}{2p^2 - 2p + 1} \end{pmatrix} \end{aligned}$$

so if I set  $e_{-2} = -1$  and  $e_{+2} = +1$ , then I get,

$$\begin{pmatrix} \frac{2p^3 - 2p^2 + 2p - 1}{2p^2 - 2p + 1} \\ \frac{p^2 - (p-1)^2}{2p^2 - 2p + 1} \\ \frac{(p-1)^3 + p(p^2 - p + 1)}{2p^2 - 2p + 1} \end{pmatrix}$$

and then if I further set  $p = 0.6$ , then I get,

$$\begin{pmatrix} -0.169231 \\ 0.384615 \\ 0.753846 \end{pmatrix}$$

precisely as what I got from iterating the transition matrix until it converges and then reading off the columns. One last thing to note, is that the equation,

$$\vec{e} \cdot \vec{T} = \vec{e}$$

from above implies that (if I multiply by  $T$  from the right on both sides),

$$\vec{e} \cdot \vec{T}^2 = \vec{e} \cdot \vec{T},$$

but  $\vec{e} \cdot \vec{T} = \vec{e}$ , so

$$\vec{e} \cdot \vec{T}^2 = \vec{e},$$

and,

$$\vec{e} \cdot \vec{T}^\infty = \vec{e},$$

which is exactly what we got by iterating the transition matrix to convergence.

### 3 3 of n: accounting for turnovers

In parts 1 and 2 I described a problem that could be modeled as a Markov chain and that roughly corresponds to the scoring of touchdowns and safetys. Now I will discuss introducing the concept of a turnover. In the original problem, the alker had a probability to move right of  $p$ , and left of  $(1-p)$ . I will now model the case where a “turnover” occurs with probability  $a$ . This reverses right and left, so that the walker moves left with probability  $p$  and right with probability  $(1-p)$ . The most straight-forward way to model this augmented problem is to double the size of the number of possible states by including a flag that tells you whether the walker is moving “right” (as in the original problem) or left (after reversing directions). I think it’s convenient to refer to a state as  $s_x^y$ , where  $x$  refers to location and  $y$  to direction. As a concrete example, the state 1 position over from the left sink, and moving to the right can be denoted as  $s_{-1}^+$ , the state one step over from the left sink and moving to the left as  $s_{-1}^-$ . Additionally, the values of the sinks get transposed, so that  $e_{+2}^+ = -e_{-2}^-$  (think a touchdown scored in the + direction is +7, a touchdown in the - direction is -7) and  $e_{-2}^+ = e_{+2}^-$  (a safety given up in the + direction is worth -2, in the - direction +2). Combining the effects of stepping and turning over, results in a transition matrix that looks like this,

$$\begin{pmatrix} 1 & (1-a)(1-p) & 0 & 0 & 0 & 0 & a(1-p) & 0 & 0 & 0 \\ 0 & 0 & (1-a)(1-p) & 0 & 0 & 0 & 0 & a(1-p) & 0 & 0 \\ 0 & (1-a)p & 0 & (1-a)(1-p) & 0 & 0 & ap & 0 & a(1-p) & 0 \\ 0 & 0 & (1-a)p & 0 & 0 & 0 & 0 & ap & 0 & 0 \\ 0 & 0 & 0 & (1-a)p & 1 & 0 & 0 & 0 & ap & 0 \\ 0 & ap & 0 & 0 & 0 & 1 & (1-a)p & 0 & 0 & 0 \\ 0 & 0 & ap & 0 & 0 & 0 & 0 & (1-a)p & 0 & 0 \\ 0 & a(1-p) & 0 & ap & 0 & 0 & (1-a)(1-p) & 0 & (1-a)p & 0 \\ 0 & 0 & a(1-p) & 0 & 0 & 0 & 0 & (1-a)(1-p) & 0 & 0 \\ 0 & 0 & 0 & a(1-p) & 0 & 0 & 0 & 0 & (1-a)(1-p) & 1 \end{pmatrix}$$

So, for example, the walker moves from state  $s_{-1}^+$  (the 2nd column) to the state  $s_{-2}^+$  with probability  $(1-a)(1-p)$ , the state  $s_0^+$  with probability  $(1-a)p$ , the state  $s_{-2}^-$  with probability  $ap$ , the state  $s_{-2}^-$  with probability  $a(1-p)$ . Note that when  $a = 0$ , “moving to the right” (the top-left 5x5) separates from “moving to the left” (the bottom-right 5x5). Specifically,

$$\begin{pmatrix} 1 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 1-p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & 1 \end{pmatrix}$$

With this formulation, everything I talked about in parts 1 and 2 carries over, i.e., the expectation values can be determined by iterating the transition matrix until it converges, or by separating out the sink and non-sink states and explicitly solving the system of equations for the non-sink expectation values. If I take  $p=1-p=0.5$ , and  $a=0$ , the solution is,

$$\begin{pmatrix} e_{-1}^+ \\ e_0^+ \\ e_{+1}^+ \\ e_{-1}^- \\ e_0^- \\ e_{+1}^- \end{pmatrix} = \begin{pmatrix} -0.5 \\ 0 \\ 0.5 \\ -0.5 \\ 0 \\ 0.5 \end{pmatrix}$$

If I take  $p=0.6$ , and  $a=0$ , the solution is,

$$\begin{pmatrix} e_{-1}^+ \\ e_0^+ \\ e_{+1}^+ \\ e_{-1}^- \\ e_0^- \\ e_{+1}^- \end{pmatrix} = \begin{pmatrix} -0.169231 \\ 0.384615 \\ 0.753846 \\ -0.753846 \\ -0.384615 \\ 0.169231 \end{pmatrix}$$

Ok, so these make sense based on the results from parts 1 and 2. If I take  $p=0.5$  and vary the value of  $a$ , then the result is as shown in Fig. 1. In other words, it makes no difference if I reverse direction or not. This makes sense because either way I am moving left or right with equal probability, and moreover, I have defined the points for landing at location +2 (a “touchdown”) to be equal to the points for landing in location -2 (a “safety”).

If I set  $p=0.6$  and vary  $a$ , the results are as shown in Fig. 2. And, if I set  $p=0.9$  and vary  $a$ , the results are as in Fig. 3.

As a test case to investigate in more detail, let me choose  $p = 0.6, a = 0.1$ . In that case the solution is,

$$\begin{pmatrix} e_{-1}^+ \\ e_0^+ \\ e_{+1}^+ \\ e_{-1}^- \\ e_0^- \\ e_{+1}^- \end{pmatrix} = \begin{pmatrix} -0.316552 \\ 0.206897 \\ 0.642069 \\ -0.642069 \\ -0.206897 \\ 0.316552 \end{pmatrix}$$

An important point to note is that the solution has a lot of symmetry. In other words, if you flip the direction you’re moving and also flip the starting point, then the expectation value is equal in magnitude and opposite in sign. So, for example, starting from  $s_{-1}$  and moving right is the negative of starting from  $s_{+1}$  and moving left. I can write this more concisely as,  $e_i^- = -e_{2m-i}^+$ , where  $m$  stands for “middle”. In this case  $m = 0$ , and

$$e_{-1}^- = -e_{+1}^+ e_0^- = -e_0^+ e_{+1}^- = -e_{-1}^+$$

Moreover, the transition probabilities are symmetric if we flip directions and flip position around the middle, that is,  $T_{i,j}^{+,-} = T_{2m-i,2m-j}^{+,-}$ . We can use the symmetry of the expectation values and transition probabilities to avoid the doubling of the state space. To repeat the general equation for expectation value in terms of transition probabilities (Einstein notation),

$$\begin{aligned} e_j^+ &= e_i^+ T_{i,j}^+ \\ &= e_i^+ T_{i,j}^{+,+} + e_i^- T_{i,j}^{-,+} \end{aligned}$$

With the assumption that a “turnover” occurs with probability  $a$ , and the relations described above, I can write,

$$e_j^+ = (1-a)e_i^+ T_{i,j}^{+,+} - ae_{2m-i}^- T_{2m-i,2m-j}^{-,-}$$

So if I generate a matrix,  $U$ , with elements  $U_{i,j} = (1-a)T_{i,j}^+ + aT_{i,2m-j}^-$ , then I can write the vector equation  $\vec{e} = \vec{e}U$ . Everything discussed in part 2 regarding solving this then carries over. As a concrete example, here is what the  $5 \times 5$  matrix  $U$  looks like for this problem,

$$\begin{pmatrix} 1-a & (1-a)(1-p) & 0 & -a(1-p) & -a \\ 0 & 0 & (1-a)(1-p) - a(1-p) & 0 & 0 \\ 0 & (1-a)p - a(1-p) & 0 & (1-a)(1-p) - ap & 0 \\ 0 & 0 & (1-a)p - ap & 0 & 0 \\ -a & -ap & 0 & (1-a)p & 1-a \end{pmatrix}$$

Finally, since this equation is identical to the one in part 2, with  $U$  replacing  $T$ , the transition matrix without considering turnovers, it follows that a 2nd way to determine the expectation values is to iterate  $U$  until it converges, and read off the columns, as in part 1. In the football application, where the transition matrix has (depending on the parameterization) about 8000 elements, iterating may be a better choice, computationally.

## 4 4 of n: the yards gained distributions (transition matrix)

In football, a basic state consists of a set of down-distance-yardline values. One could include score differential, or time I suppose, but I’m not considering those here. The transition matrix can be built by using the yards-gained distribution, along with the probabilities to run a play as opposed to punting or attempting a field goal. So the main questions I want to address here are,

- what are the probabilities for passing versus running versus kicking?
- given that a play is run, what is the yards gained distribution?

I am ignoring context-specific distinctions such as time on the clock, score-differential, weather... This is very much in the same vein as something like RE24 (from the more developed field of baseball analytics) in that it accounts for game state, but is otherwise context neutral.

With that being said, the dependence of the yards-gained distribution on down, distance, and yardline are not at all obvious to me. So my starting point was to grab some data and start slicing and dicing and making some graphs. To clarify, yardline is encoded as “yards-from-own-goal”, abbreviated yfog, where 1 means you are backed up against your end zone, and 99 means you are 1 yard away from scoring a touchdown.

- In between, say yfog=20 and yfog=75, mean yards gained doesn’t depend strongly on field position (Fig. 4).



- For plays originating between yfog=20 and yfog=75, mean yards gained doesn't depend strongly on yards-to-go (Fig. 5).
- The fraction of plays that are passes depends strongly on down and distance, and weakly on field position (Figs. 6 & 7).
- On fourth down, the probabilities to punt versus try a field goal versus go for it vary rapidly as a function of yards-to-go and yards-from-own-goal, particularly for yfog  $\sim$  50 to yfog  $\sim$  80 (Figs. 8, 9, and 10).

So now the question is, how can we model the yards-gained distribution? Since mean yards gained doesn't depend too strongly on down, distance, and field position, it is instructive to pool a bunch of states, and look in more detail at the distribution to get a feel for what it looks like. In Fig. 11, I show distributions for passes (left) and rushes (right), for all first-down plays originating between yfog=20 and yfog=75.

After playing around with some functions, the best general agreement I could find came from the following,

$$p(y) = A \frac{e^{(y-y_0)/\sigma_1}}{1+e^{(y-y_0)(\sigma_1+\sigma_2)/(\sigma_1\sigma_2)}} + G e^{-(y-g_0)^2/(2\sigma_g^2)}.$$

I refer to this as a “Bazin plus Gauss” function; Bazin because I first encountered the first term in a paper by Bazin, et al, that used it to model supernova light curves (<http://arxiv.org/abs/1109.0948>), and Gauss for obvious reasons. The first (Bazin) term basically stitches together two exponentials at the location  $y_0$ . For  $y \ll y_0$ , it looks like a rising exponential with a scale factor  $\sigma_1$ , and for  $y \gg y_0$ , like a declining exponential with a scale factor  $\sigma_2$ . The Gaussian part describes being sacked, and  $G$  is identically 0 for rushes.

Using this functional form, I use the function minimization package `pyminuit` to determine the maximum likelihood values for the parameters,  $y_0, \sigma_1, \sigma_2$  for rushes, and additionally  $G/A, g_0$ , and  $\sigma_g$  for passes. For rushes, typical values might be  $y_0 \sim 1, \sigma_1 \sim 1.5, \sigma_2 \sim 3.5$ . For passes,  $y_0 \sim 4.5, \sigma_1 \sim 1.8, \sigma_2 \sim 8.0, G/A \sim 0.12, g_0 \sim -6.5$ , and  $\sigma_g \sim 3.0$ . Figs. 12 & 13 compare the model to the empirical distribution for 1st and 10 plays from the 20.

In the next section I will describe how the model is implemented in code.

## 5 4 of n: the parameterization of the transition matrix

In part 3, I described modeling the yards-gained distribution, including the probabilities to kick versus run a play. Here I will describe how these are controlled in the Markov chain computer program.

The program reads a parameter file which has a generic form of

```
parameter-name down ytg-min ytg-max yfog param-value
```

For a given down and distance range (yards-to-go, or ytg), the param values are stored as a function of yards-from-own-goal. Then, during run time, the parameter value is set using linear interpolation over yfog. This gives a very flexible input scheme. As a concrete example, here are example parameter values for “going for it on 4th”,

```
4thGoForItProb 4 1 1 0 0
4thGoForItProb 4 1 1 20 0
4thGoForItProb 4 1 1 50 0.24
4thGoForItProb 4 1 1 60 0.78
4thGoForItProb 4 1 1 70 0.75
4thGoForItProb 4 1 1 85 0.42
4thGoForItProb 4 1 1 100 0.50
```

```

4thGoForItProb 4 2 4 0 0
4thGoForItProb 4 2 4 20 0
4thGoForItProb 4 2 4 40 0.07
4thGoForItProb 4 2 4 50 0.10
4thGoForItProb 4 2 4 60 0.45
4thGoForItProb 4 2 4 80 0.14
4thGoForItProb 4 2 4 100 0.14

```

This says that, when its 4th and between 1 and 1 to go (in other words precisely 1), if yfog is between 0 and 20, then interpolate between 0 and 0. In other words the probability to go for it is 0. If yfog is between 20 and 50, interpolate between 0 and 0.24. So for example, half way in between, when yfog=35, the probability would be 0.12. On the other hand, if yards-to-go is between 2 and 4 (inclusive), then the probability at yfog=35 will be the slope,  $(0.07 - 0.00)/(40 - 20)$ , times  $35 - 20 = 15$ , plus the initial value at yfog = 20, which is 0. That is,  $0 + 0.07/20 * 15 = 0.0525$ . If down is set to 0, it means the parameters hold for downs 1-4. The parameters that are controllable are,

```

intProb: interception probability
fumProb: fumble probability
incompleteProb: incomplection probability
yardsDistParsRush: parameters of the Bazin-Gauss function for rushes (G identically 0)
yardsDistParsPass: parameters of the Bazin-Gauss function for passes
4thGoForItProb: probability to go for it on 4th
4thFgProb: probability to go for a field goal on 4th
FgMakeProb: probability to make a field goal
passProb: given that you run a play, the probability to pass

```

The final inputs are a model name, which determines the output data file name, and a flag to say whether the model is empirical. In that case, the parameters controlling the yards-gained distributions are ignored and the distributions are instead read in empirically.

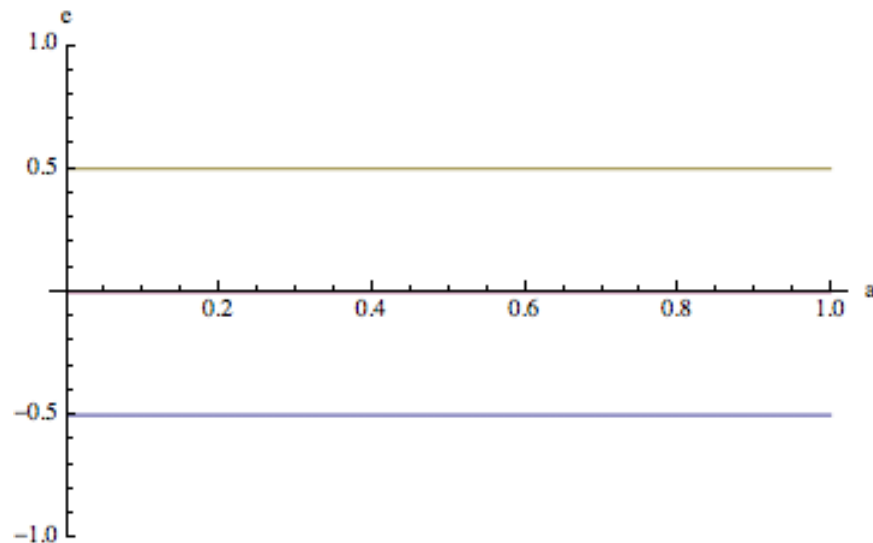


Figure 1: points

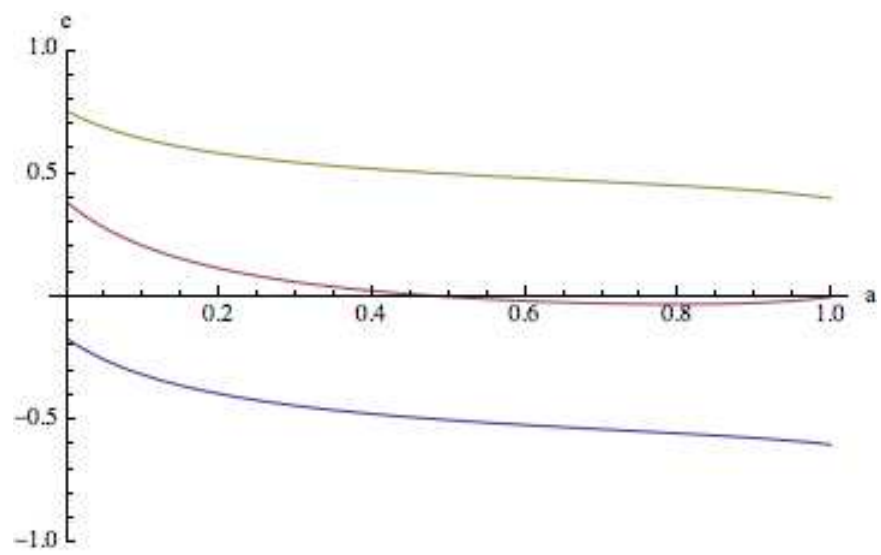


Figure 2: points

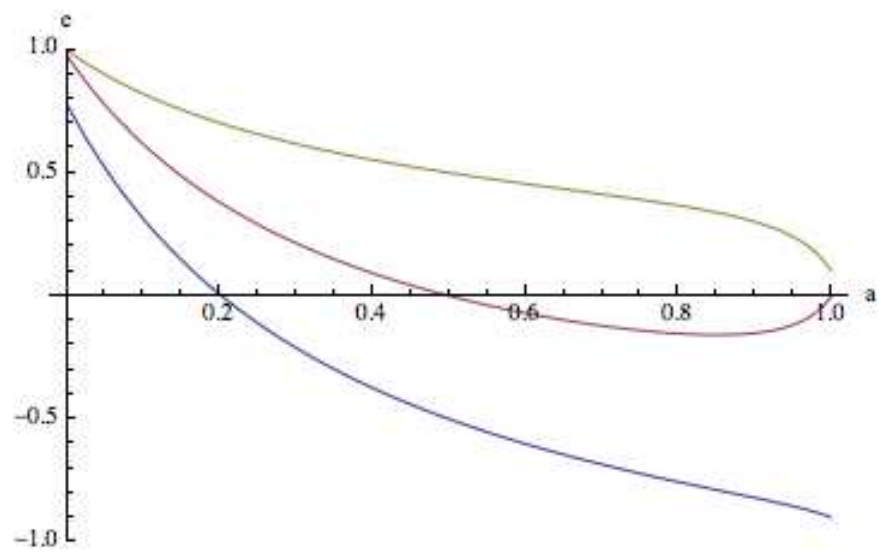


Figure 3: points

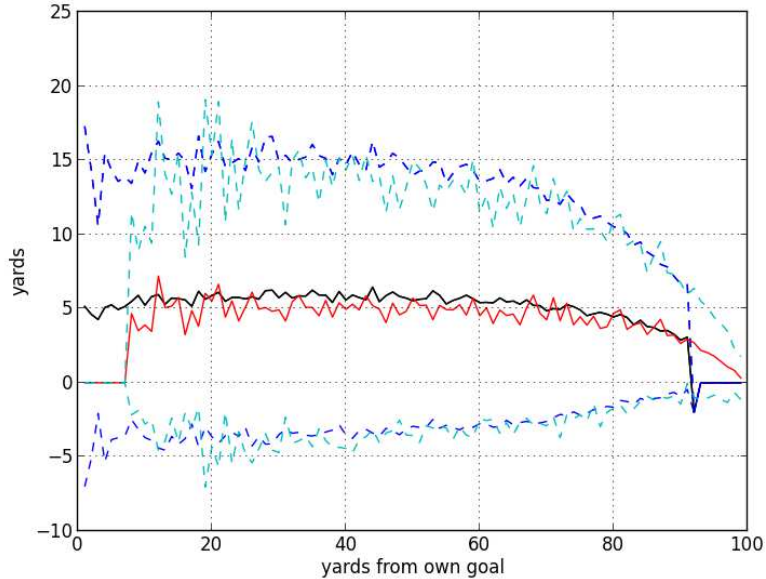


Figure 4: mean yards per play as a function of field position. black is 1st and 10 or more, red is 3rd and short, specifically 3 or less. dashed lines show  $\pm 1$  standard deviation. what I take away from this is that the standard deviation is so large that you don't have to worry about variation that depends on field position as long as you're at 75 yards or less.

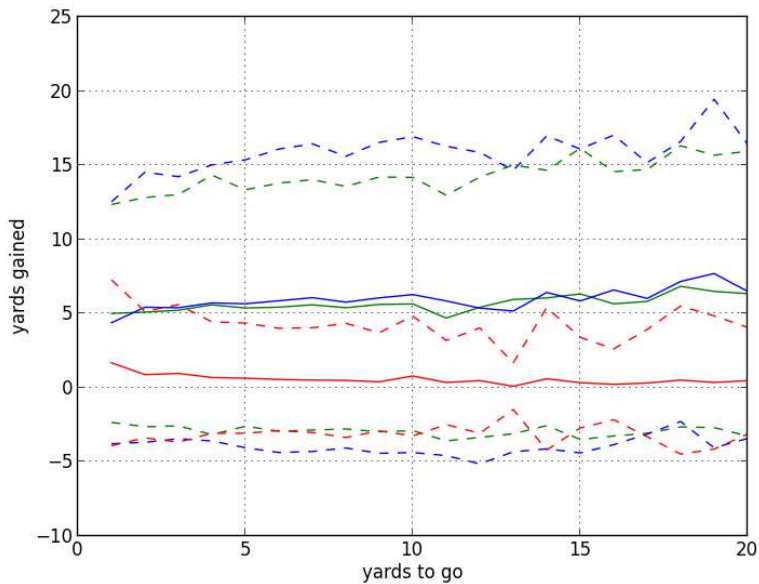


Figure 5: (green, blue, red) = (2nd, 3rd, 4th) downs. The only 1st down values that matter are 1st and 5, 1st and 10 and first and 15, which are 1 and 5 :  $5.27 \pm 8.63$  1 and 10; :  $5.71 \pm 9.00$  1 and 15; :  $6.15 \pm 8.94$

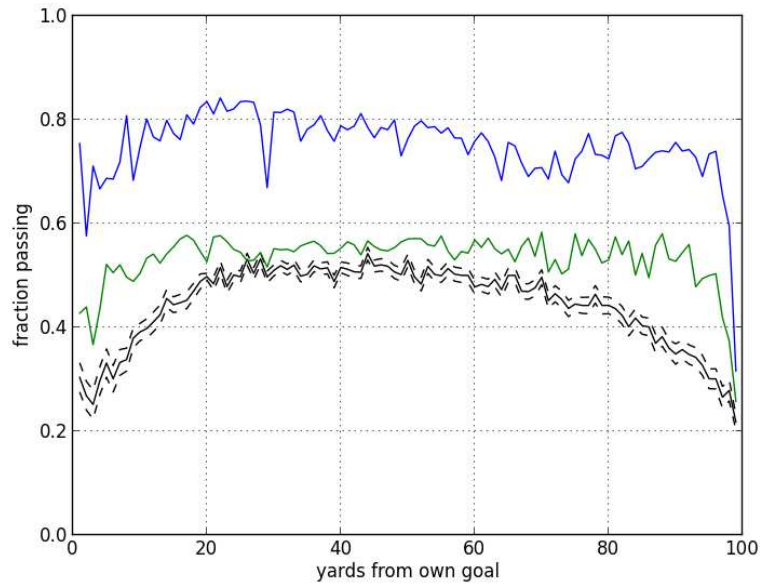


Figure 6: here is fraction of plays that are passes by yard from own goal. black, green, blue are 1st, 2nd 3rd downs. the point at yfog=29 is interesting; if you are 3rd down at the 29, you probably got there starting from 1st and 10 at the 20 (a touchback) so you are 3rd and 1 and you rush more often. its so logical!

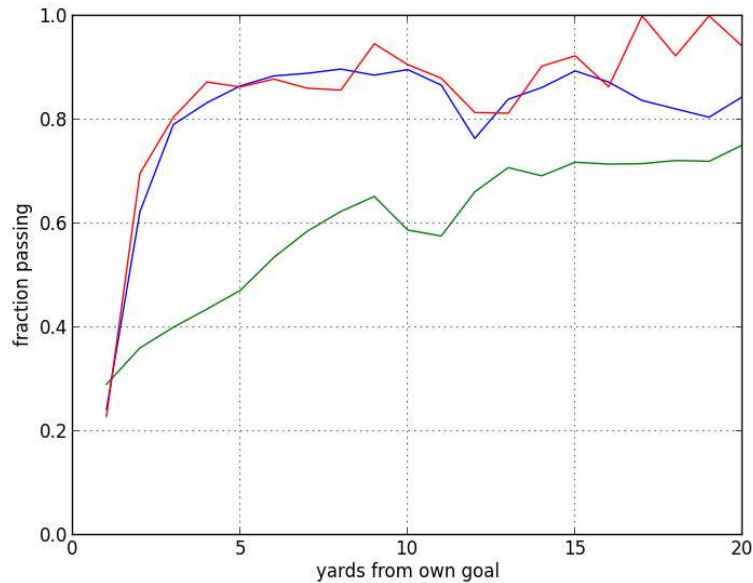


Figure 7: here is fraction of plays that are passes (given that its either a pass or a rush and originates between yfog=20 and yfog=75), based on distance to a first down. green, blue, red are 2nd, 3rd, 4th downs. 1st and 10 is 49.83% pass (with N=88547).

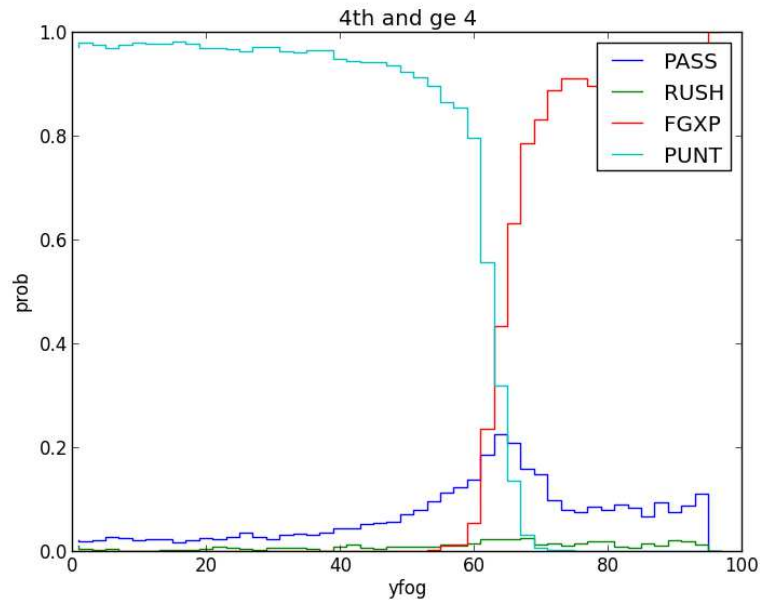


Figure 8: Probabilities to attempt (pass, rush, field goal, punt) for 4th and  $\geq 4$

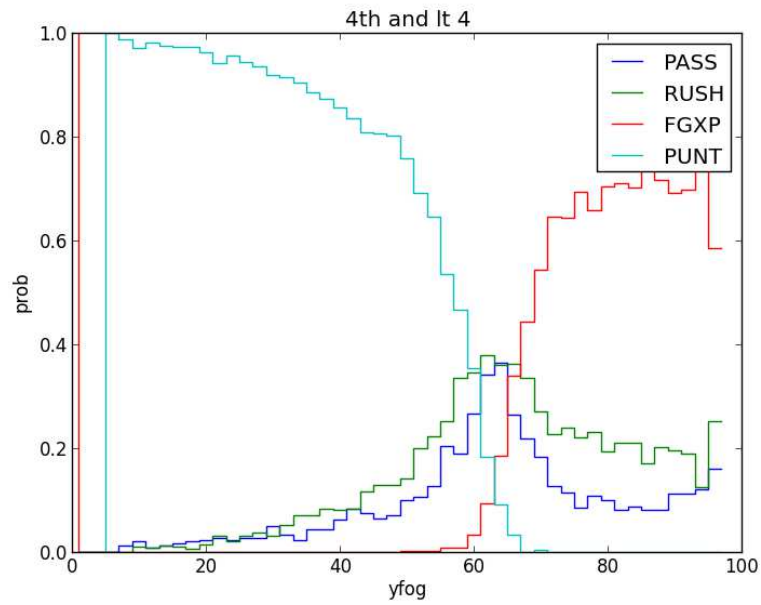


Figure 9: Probabilities to attempt (pass, rush, field goal, punt) for 4th and  $< 4$

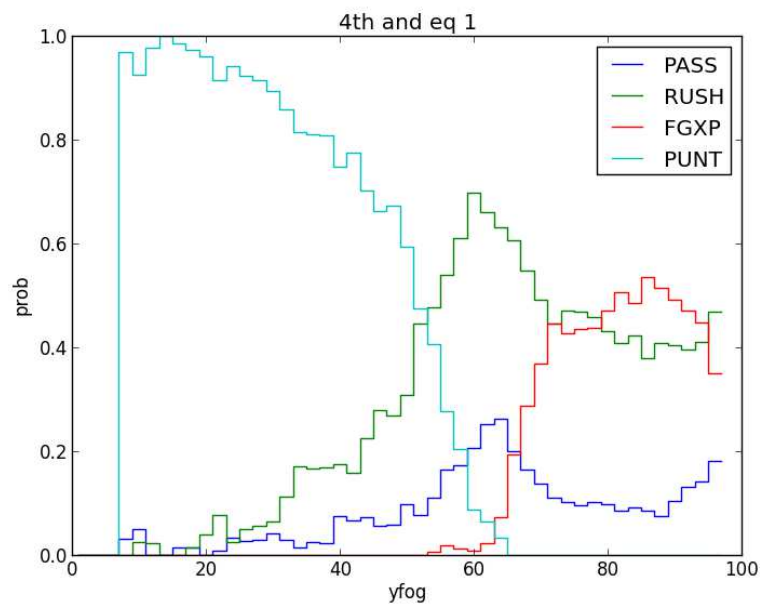


Figure 10: Probabilities to attempt (pass, rush, field goal, punt) for 4th and 1

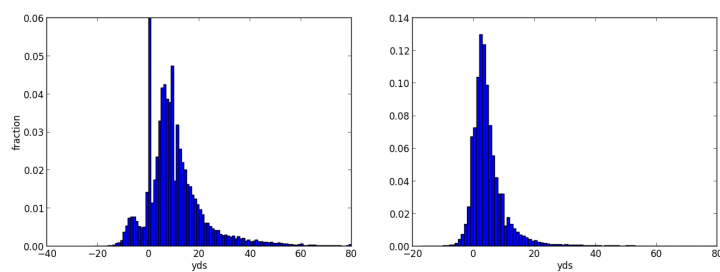


Figure 11: distributions of yards gained for all 1st downs that originate between 20 and 75, left panel is passes (35% or so gain 0 yards), right is rushes.

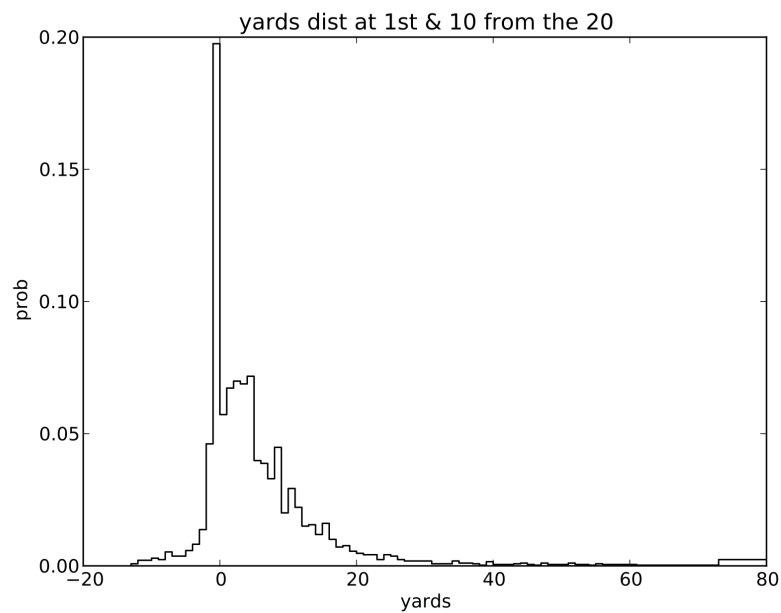


Figure 12: distributions of yards gained for all 1st downs that originate between 20 and 75, left panel is passes (35% or so gain 0 yards), right is rushes.

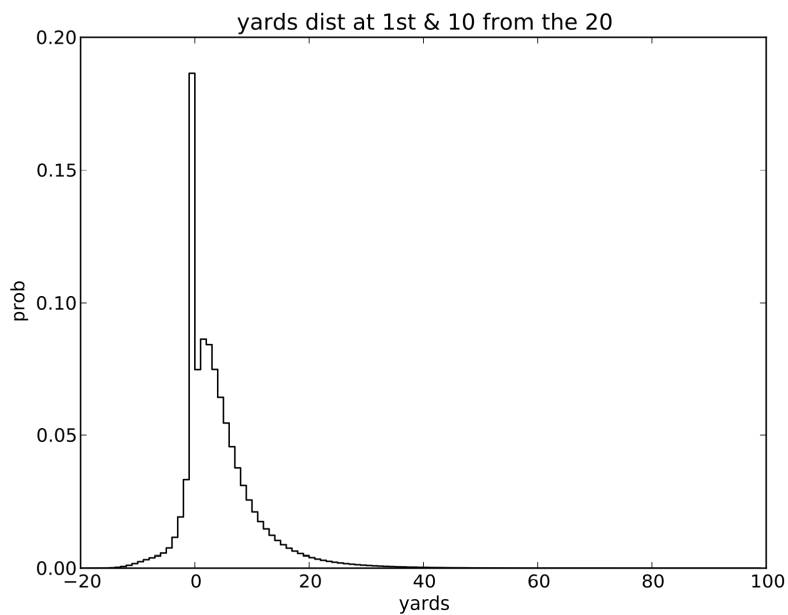


Figure 13: distributions of yards gained for all 1st downs that originate between 20 and 75, left panel is passes (35% or so gain 0 yards), right is rushes.