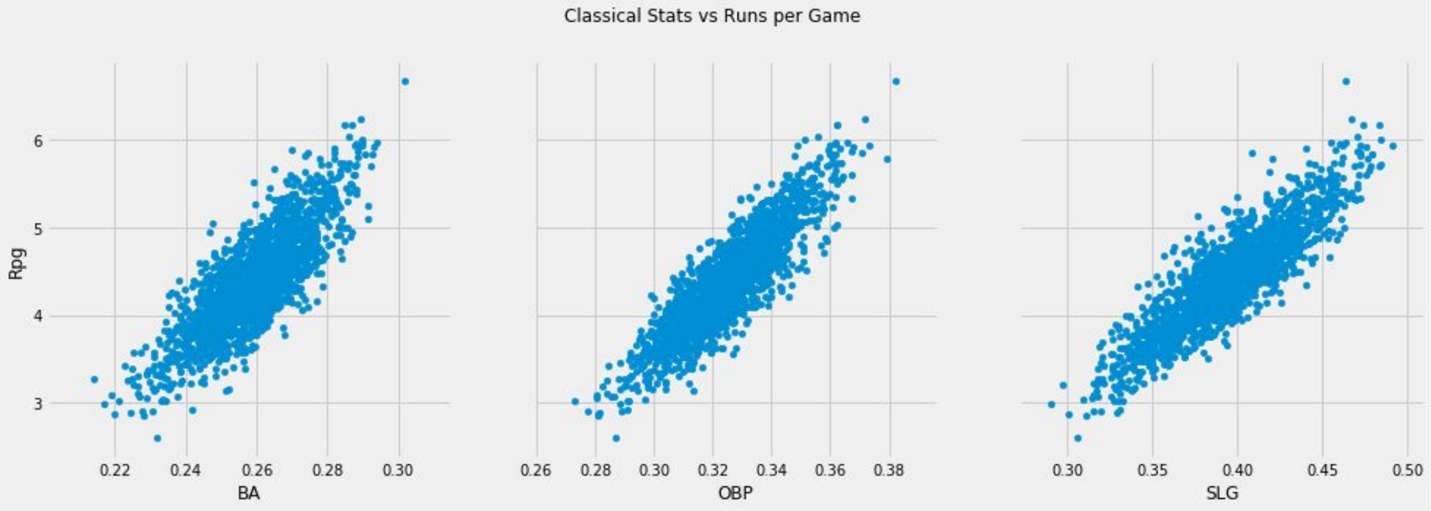
LS88: Sports Analytics

Run Expectancy

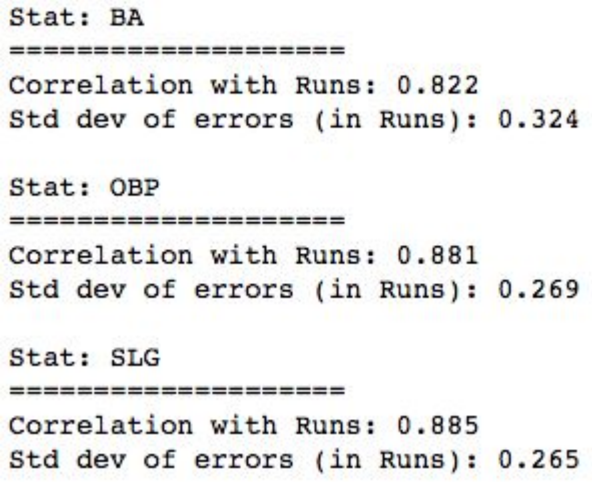
Classical Stats and Run Scoring

We know BA, OBP, and SLG are correlated/associated with run scoring



Classical Stats and Run Scoring

We know the weights matter



```
Stat: BA
====================
Correlation with Runs: 0.822
Std dev of errors (in Runs): 0.324

Stat: OBP
====================
Correlation with Runs: 0.881
Std dev of errors (in Runs): 0.269

Stat: SLG
====================
Correlation with Runs: 0.885
Std dev of errors (in Runs): 0.265
```

Classical Stats and Run Scoring

- How can we evaluate events that don’t directly score runs all the time?
  - A single with no one on
- We “know” these events must increase our likelihood of scoring runs
  - Getting on base means you’re 90’ closer to scoring

What We Expect from Plate Appearances

- You can’t score negative runs, so every PA is worth something positive
  - How much? Total Runs / Total PA. Typically about .1 Runs per PA
- What happens when you make an out?
  - You miss out on 0.1 Runs per PA by adding anything
  - And you shorten the inning for everyone else!
    4.5 Runs per Game / 27 Outs per Game = ~.17 Runs per Out
  - *Decreases* the expected number of runs we score *below* the average
  - The cost of shortening the game is like an *opportunity cost*

What We Expect from Plate Appearances

- You can’t score negative runs, so every PA is worth something positive
  - How much? Total Runs / Total PA. Typically about .1 Runs per PA
- What happens when you single?
  - You didn’t score a run but you give future hitters a chance to knock you in
  - Your value is tied up in the run scoring process: the interaction between hitters getting on base and hitters advancing runners
  - *Increases* the expected number of runs we score *above* the average

What We Expect from Plate Appearances

We don’t tick back runs but relative to expectation, ~0.1 Runs per each PA, outs are negative and singles are positive

Why Run Expectancy?

- We need to create a grounding through expectation
  - Given a situation, how many runs do we expect to score?
- Our world is always about forming expectations
  - We can’t rely on intuition or nothing alone
  - We need to rely on model/data to form expectations

Why Run Expectancy?

- We can evaluate based on changes in expectation
  - Increasing how many runs we expect to score is good
  - Decreasing is bad
- Evaluation was easy-ish for an out. And we know a Home Run scores at least 1 run
  - But we’re clueless right now about other events!

Why Run Expectancy?

We *need* a baseline expectation

We can’t evaluate who’s good/bad without a baseline

➔ Our method is to ignore quality/strength of players
    - We want the expected performance under completely average circumstances
    - Good players will eventually reveal themselves as above expectation.

What is Run Expectancy?

Intuition:

A batter walks up to the plate with 1 out and runners on 1st and 2nd

Starting from this PA, how many runs do we *expect* the team to score until the end of the inning?

1 run? 2 runs? 3 runs?

If we have this number, that would be the *Run Expectancy* value

Game States

A *Game State* is a description of the game that succinctly describes relevant details about the action.

Our baseball Game State: a baserunner situation and a number of outs

Examples:

Runner on 1st, 1 out
Runners on 2nd and 3rd, no out
Bases Loaded, 2 out

Game States

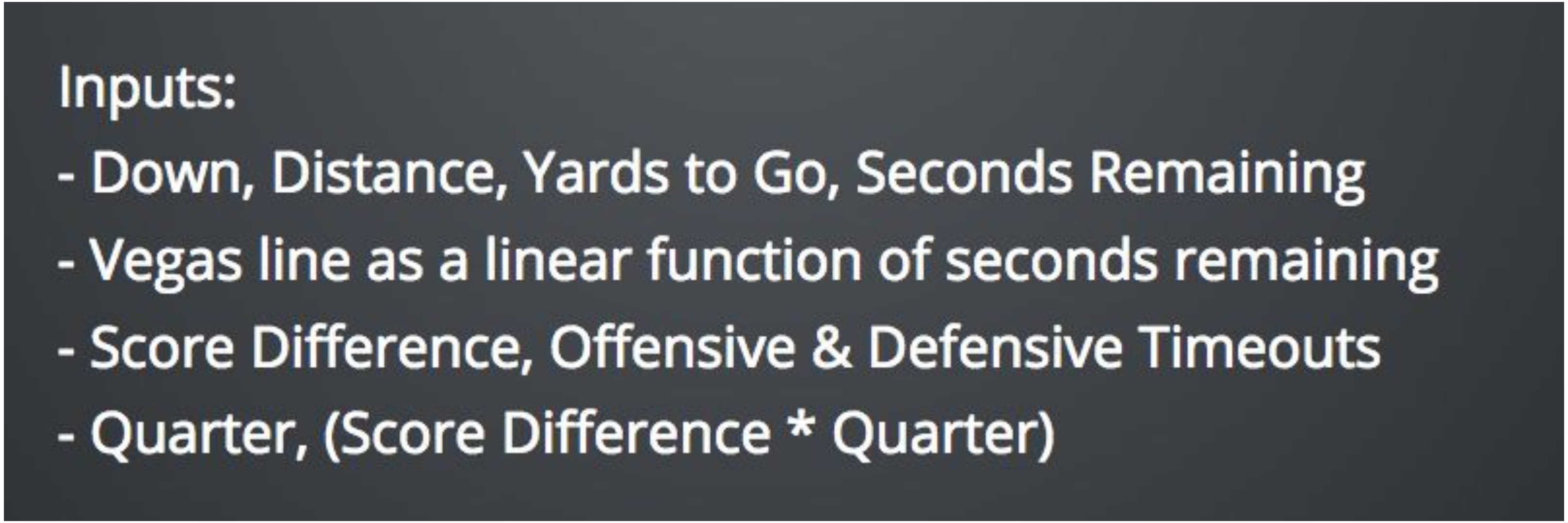
- We’re starting with a simple state for baseball: base runners and outs
- More complex states for baseball:
  - Base runners, outs, inning, score
  - Count, players

Game States in Other Sports

- ➔ Football: down, distance, yard line, time, quarter, score, timeouts remaining
- ➔ Basketball: possessing team, players on court, time, quarter, score, fouls, timeouts
- ➔ Basketball: precise spatial location of players from player tracking

Game States in Other Sports

Game State variables for the NYT 4th Down Bot (computes win probability)



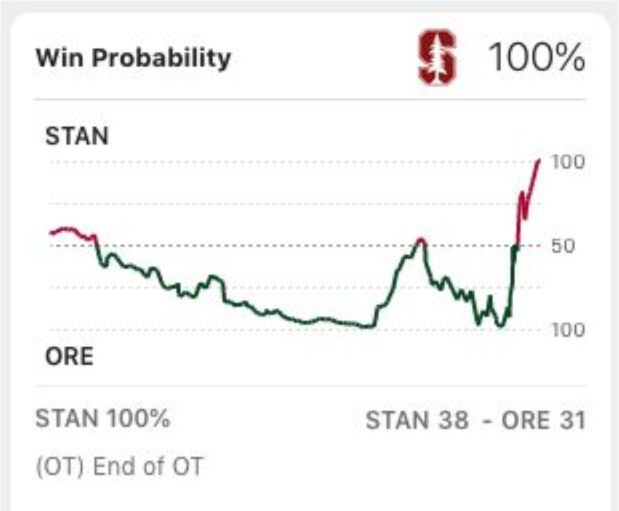
Inputs:

- Down, Distance, Yards to Go, Seconds Remaining
- Vegas line as a linear function of seconds remaining
- Score Difference, Offensive & Defensive Timeouts
- Quarter, (Score Difference * Quarter)

http://slides.com/treycausey/4thdownbot#/12

Game States in Other Sports

- Game State → Win Probability



Win Probability 100%

STAN 100%

STAN 38 - ORE 31

(OT) End of OT

Game States and Run Scoring

No runners on, no outs:

This is usually the start of the inning. This is a rather neutral state where we should expect to score about Runs / Game

Bases loaded, no outs:

This is an amazing position to be in. Our team should expect to score a lot of runs from this and after.

No runners on, two outs:

We have nothing. We're likely to be out in this next plate appearance.

Runs in Remainder of Inning

From the game state, we need how many runs were scored in that PA and after until the end of the inning

Why does this make sense?

- The game state serves as a launch point to score runs in remainder of inning
- More valuable game state ➡ tend to score more runs in remainder of inning
  - Bases Loaded, no outs vs Bases Empty, 2 outs

Computing the Run Expectancy Matrix

- ➔ Group and average
- ➔ For each state *S*:

$$\text{Run Expectancy for State } S = \frac{1}{\#\text{ of PAs in State } S} \sum_{\text{PAs in State } S} (\text{Runs in Rem. of Inn.})$$

Example: Bases Loaded, No Outs

- ➔ 10 PAs with Runs ROI: 4, 0, 3, 0, 4, 4, 0, 3, 4, 5
- ➔ Run Expectancy would be 2.7 Runs

Computing the Run Expectancy Matrix

We use the Retrosheet play-by-play data:

1. Every plate appearance in the season is tracked
2. We can pull out the base runner situation, the outs, and the runs ROI

- ➔ Over all the PAs, we group by state and average.
- ➔ The values for the 24 different combinations of runners and outs are called the *Run Expectancy Matrix*

Computing the Run Expectancy Matrix

Here is an example of a Run Expectancy Matrix from 2001

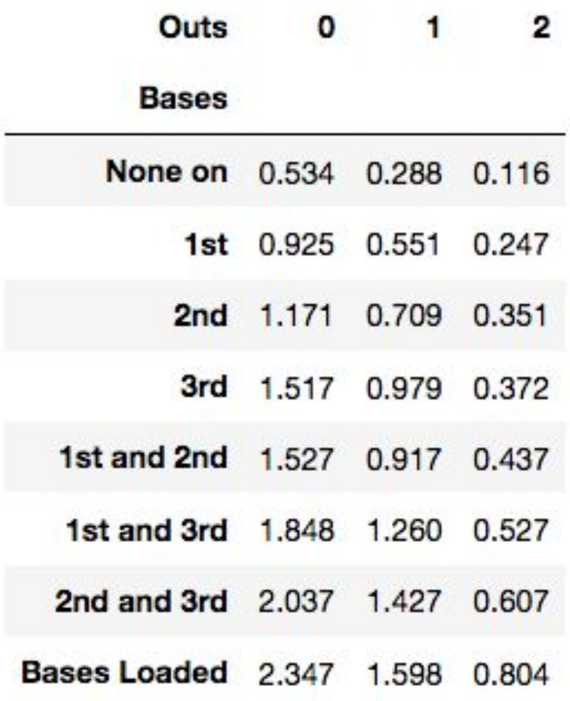
Notice:

➔ In terms of RE, an out varies in value

➔ A walk varies in value

None on to 1st vs. 2nd and 3rd to loaded

➔ Sacrifice fly: 2nd vs 3rd



Outs	0	1	2
Bases			
None on	0.534	0.288	0.116
1st	0.925	0.551	0.247
2nd	1.171	0.709	0.351
3rd	1.517	0.979	0.372
1st and 2nd	1.527	0.917	0.437
1st and 3rd	1.848	1.260	0.527
2nd and 3rd	2.037	1.427	0.607
Bases Loaded	2.347	1.598	0.804

Why Runs Remainder of Inning?

From the game state, we need how many runs were scored in that PA and all subsequent PAs until the end of the inning

- ➔ We treat PAs with the same state as equivalent whether our inning got to the state immediately or after a long series PAs

  Runner on 1st, 1 out can be reached many, many different ways

- ➔ Once we're in the state, our inning can play out similarly to other PAs in that same state

  For those interested (and can discuss further), we're assuming *Markov property* and *Stationarity*

Interpreting the Run Expectancy Matrix

What does it tell us?

Expected number of runs scored under average circumstances

➔ Expected: Repeat situation a lot and runs scored will concentrate around the value

➔ Average Circumstances: average hitters, pitchers, defenders, inning, etc. We averaged over all teams/players/circumstances

Interpreting the Run Expectancy Matrix

Scoring potential: high run expectancy = high potential

Event valuation

- ➔ No runners on, no outs. Batter gets a single. Now there's a runner on first, still no outs. What was the single worth in terms of Run Expectancy?
- ➔ 1 out, runner on first. In Run Expectancy, what can you gain by stealing second? What can you lose?

We can track player performances relative to average: good players will boost Run Expectancy

Run Expectancy Demo

Let’s do a demo.

RE24

RE24

RE24 is defined as:

$$RE24 = \text{Run Potential End State} - \text{Run Potential pre event} + \text{Runs Scored}$$

Ex:

➔ Runner on first, 1 out:

RE = 0.551

➔ Batter walks, so it’s now runners on 1sts and 2nd with 1 out:

New RE = 0.917

➔ RE24 = 0.917 - 0.551 + 0 = +0.366

The walk didn’t score a run but it was worth something!

RE24

RE24 is defined as:

$$RE24 = \text{Run Potential End State} - \text{Run Potential pre event} + \text{Runs Scored}$$

Ex:

➔ Bases loaded, no outs

RE = 2.347

➔ Grand slam. Bases empty, no outs.

New RE = 0.534

➔ RE24 = 0.534 - 2.347 + 4 = +2.187

The grand slam scored 4 runs, but high RE discounts it to just above 2!

RE24

Key Points:

- Use RE as change in run potential
- Gives a value for every plate appearance event
- For each batter, add up the RE24 values: Total RE24
  - A context-dependent, count stat which is *WAY* better than RBI

RE24

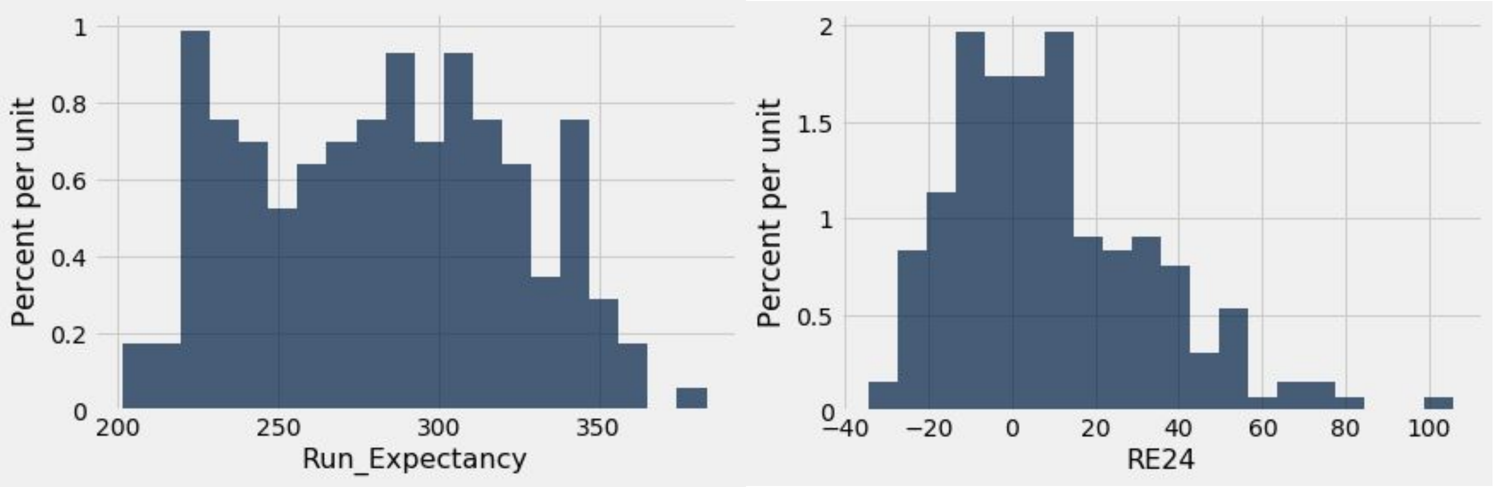
- ➔ What does it tell us? Contextualized run production
- ➔ It’s a value metric: retrospective of what the play’s value was
- ➔ Total RE24 is a player’s total value contribution

Some features:

- ➔ Partially contextualized: accounts for run scoring potential (number of outs/runners)
- ➔ Still situational: all singles with bases empty worth less than all singles with bases loaded
- ➔ Production: events that do not score runs still “produce”. Production is now captured

RE24

RE and RE24 among qualified hitters (≥ 400 PA) for 2001



RE24 Demo

Let’s do another demo!

Run Expectancy Summary

Some key takeaways:

- Without context for a situation, we cannot gauge performance
- We rely on lots of events to build estimates of RE
- Remove situational context
- Discount production based on expected outcome
    - Credit only above or below expectation