Sports Analytics in Python

A tour of the hobbyist's playground

July 13, 2016

Table of contents

Introduction

Bio

Who am I?

- Currently: Analyst at Google
- Recently: Data Scientist/Statistician at Capital One
- Before that: Got a couple of degrees in Economics

Sports questions I've focused on:

- Evaluating goalie quality in hockey
- Predicting basketball games and betting on them
- Taking advantage of my friends in fantasy hockey

I've started writing about sports on `oddacious.github.io`

Opinions expressed herein are in my own, and do not represent my employer's viewpoints on sports analytics...or other topics

What is sports analytics?

Simple Definition

The application of quantitative methods to the realm of sports

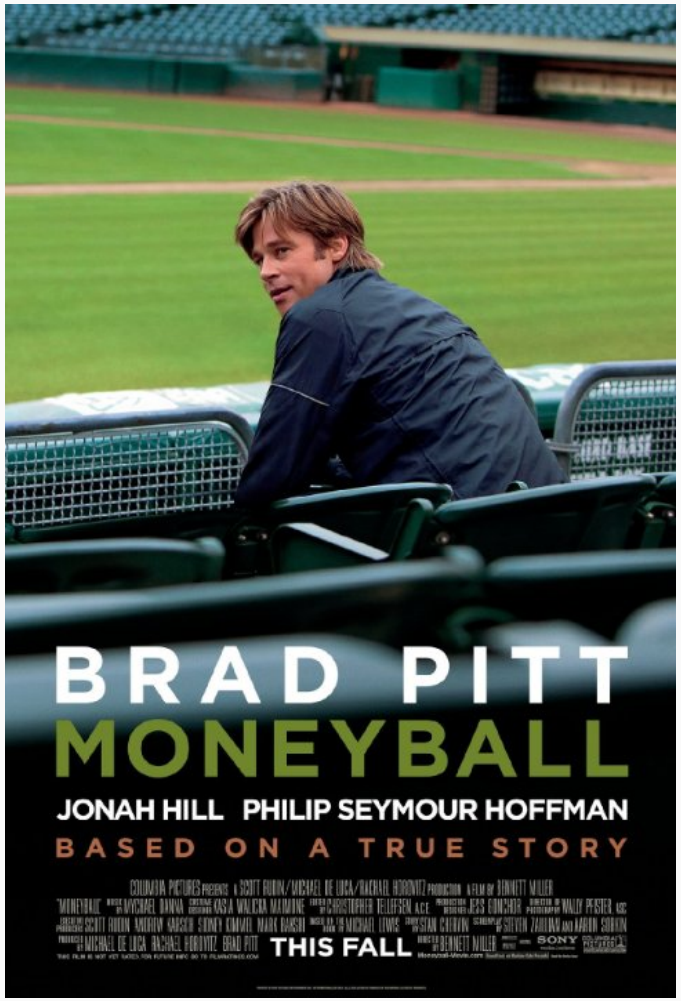
But why?

Data collection is ever-expanding and so is the set of people who can use it

Who is it for?

- *Managers* who want to use their budget wisely
- *Coaches* who want to play the appropriate players
- *Scouts* who want to evaluate across leagues
- *Trainers* who want to prevent injuries
- *Players* who want to understand their opponents
- *Reporters* who want to explain victory and defeat
- *Academics* who want to be less bored in grad school
- *Bettors* who want avoid losing money
- *Fans* who want to argue
- *Nerds* who want to be right

Analytics are great if you're this guy



…Or if you're this guy

Why Python?

It’s not the only option out there

With other audiences, Python might not always be the self-evident choice

Browsing my sports code I found a few file extensions...

It’s not the only option out there

With other audiences, Python might not always be the self-evident choice

Browsing my sports code I found a few file extensions...

- .py

It’s not the only option out there

With other audiences, Python might not always be the self-evident choice

Browsing my sports code I found a few file extensions...

- .py
- .R

It’s not the only option out there

With other audiences, Python might not always be the self-evident choice

Browsing my sports code I found a few file extensions...

- .py
- .R
- .stata

It’s not the only option out there

With other audiences, Python might not always be the self-evident choice

Browsing my sports code I found a few file extensions...

- .py
- .R
- .stata
- .sas

It’s not the only option out there

With other audiences, Python might not always be the self-evident choice

Browsing my sports code I found a few file extensions...

- .py
- .R
- .stata
- .sas
- .pl

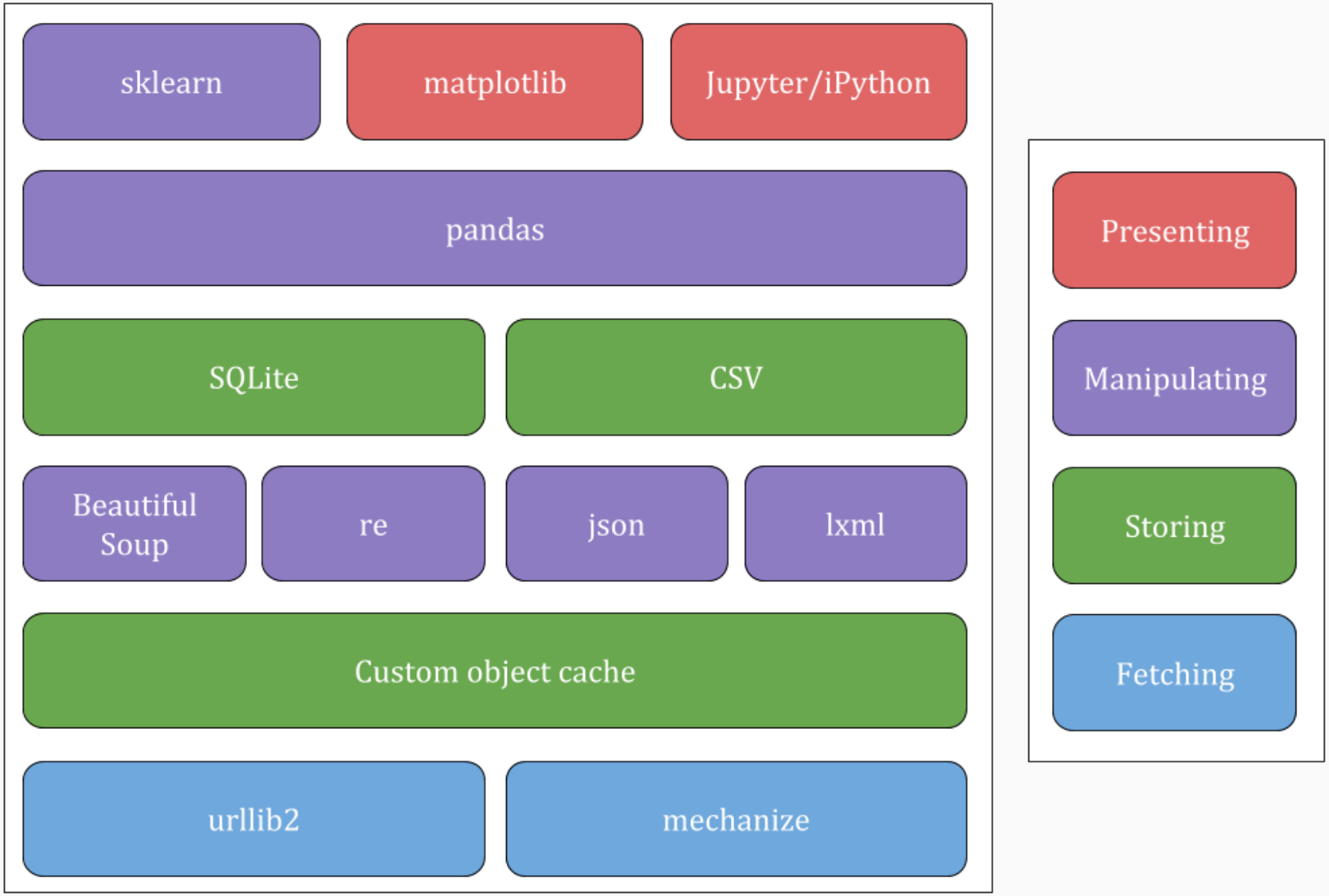
Python is now my end-to-end choice for this type of work

Python is a strong choice across the board, with no specific weakness in sports analytics

Some key reasons I now use Python heavily:

- Vibrant community
- Easier coding leads to faster insights
- Superb packages for getting, parsing, and storing data
- A growing *and consistent* data analysis codebase
- Your data will be ugly. Your code doesn't have to be.

Typical technologies and/or packages



Where Hockey Data Comes From

The data in hockey is limited but growing

What we have

- We have the occurrence of *events* (goals, shots, faceoffs, blocks, hits, penalties, and giveaways), when they happened, and where they happened
- We know who was on the ice at any given time

What I would like to have

- Where everybody else on the ice was when events happened
- Where everybody was in between the events
- Where the puck was during and between the events

Your data may come from…

Beautiful JSON


```
      "player" : {
        "id" : 8469466,
        "fullName" : "Ales Hemsky",
        "link" : "/api/v1/people/8469466"
      },
      "playerType" : "Hittee"
    } ],
    "result" : {
      "event" : "Hit",
      "eventCode" : "PIT41",
      "eventTypeId" : "HIT",
      "description" : "Simon Despres hit Ales Hemsky"
    },
    "about" : {
      "eventIdx" : 63,
      "eventId" : 41,
      "period" : 1,
      "periodType" : "REGULAR",
      "ordinalNum" : "1st",
      "periodTime" : "13:43",
      "dateTime" : "2014-04-14T00:06:06Z",
      "goals" : {
        "away" : 0,
        "home" : 0
      }
    },
```

Your data may come from...

Parsable XML

VISITOR

1

Play By Play

Saturday, October 10, 2015
Attendance 17,125 at Gila River Arena
Start 7:15 MST; End 9:45 MST
Game 0029
Final

HOME

2

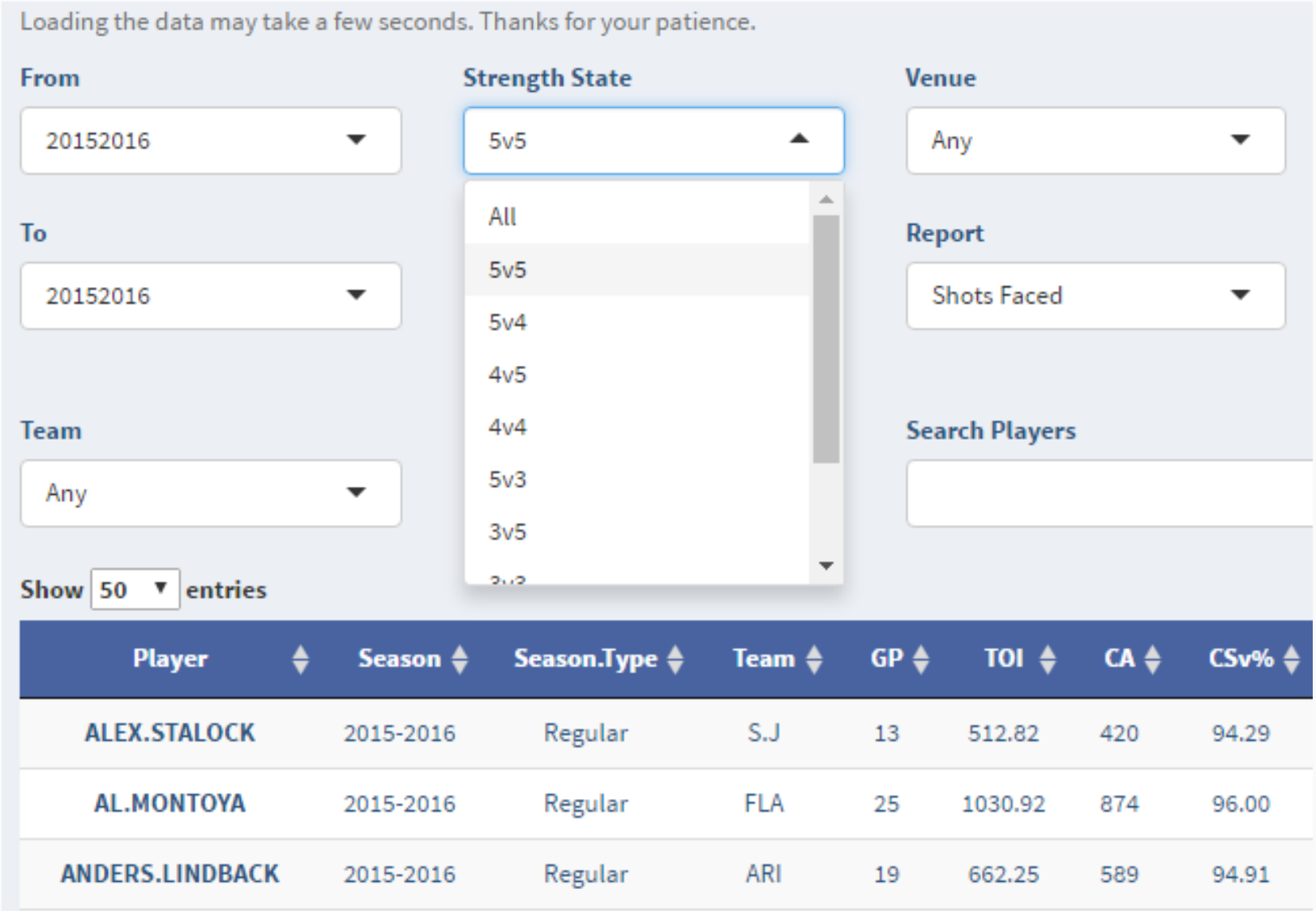
PITTSBURGH PENGUINS
Game 2 Away Game 2

ARIZONA COYOTES
Game 2 Home Game 1

#	Per	Str	Time: Elapsed Game	Event	Description	PIT On Ice
1	1		0:00 20:00	PSTR	Period Start- Local time: 7:15 MST	81 87 14 28 58 29 16 C C L D D G C
2	1	EV	0:00 20:00	FAC	PIT won Neu. Zone - PIT #87 CROSBY vs ARI #50 VERMETTE	81 87 14 28 58 29 16 C C L D D G C
3	1	EV	0:08 19:52	HIT	ARI #16 DOMI HIT PIT #14 KUNITZ, Off. Zone	81 87 14 28 58 29 16 C C L D D G C
4	1	EV	0:10 19:50	BLOCK	ARI #50 VERMETTE BLOCKED BY PIT #58 LETANG, Wrist, Def. Zone	81 87 14 28 58 29 16 C C L D D G C
5	1	EV	0:29 19:31	GIVE	PIT GIVEAWAY - #28 COLE, Def. Zone	81 87 14 28 58 29 16 C C L D D G C
6	1	EV	0:42 19:18	BLOCK	ARI #5 MURPHY BLOCKED BY PIT #14 KUNITZ, Wrist, Def. Zone	81 87 14 28 58 29 16 C C L D D G C

Your data may come from…

Interactive webpages



Your data may come from...

But not yet persistent player/ball tracking, which exists in the NBA

Quantifying Goaltenders

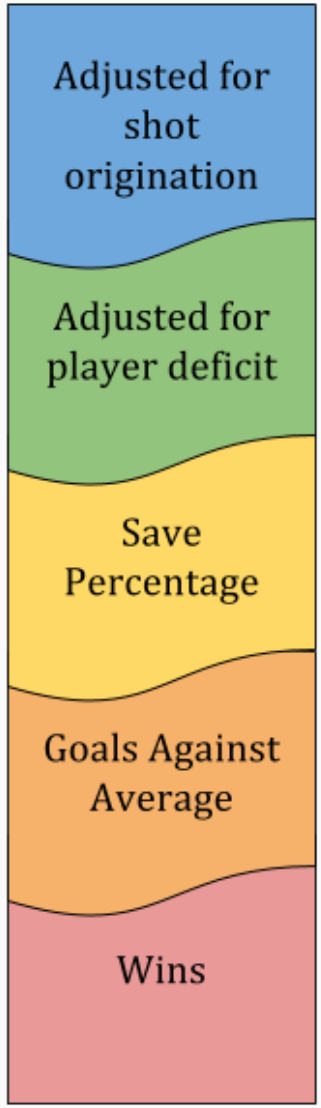
Measuring goalies is hard

Goalies are tough to evaluate

- It's tough to separate goalies from team effects, because they are always on the ice
- We have a small sample of goals

We can adjust for randomness and the information we can identify

- Goalies don't control the offense side of wins*
- Goalies don't control how many shots they face*
- Goalies don't control the quality of those shots*



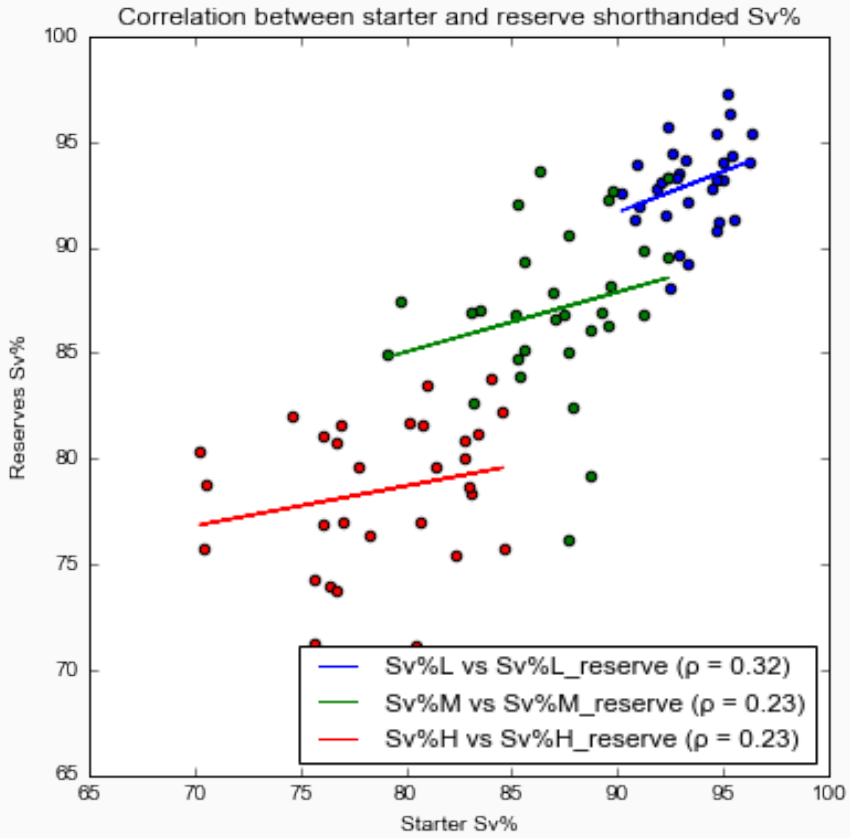
*Not entirely true, but the magnitudes are probably debatable

Problem 1: Team effects

```
stats['adj'] = stats['SvPct_L'] * stats['League_ShPct_L']
             + stats['SvPct_M'] * stats['League_ShPct_M']
             + stats['SvPct_H'] * stats['League_ShPct_H']
```

The question is still not solved

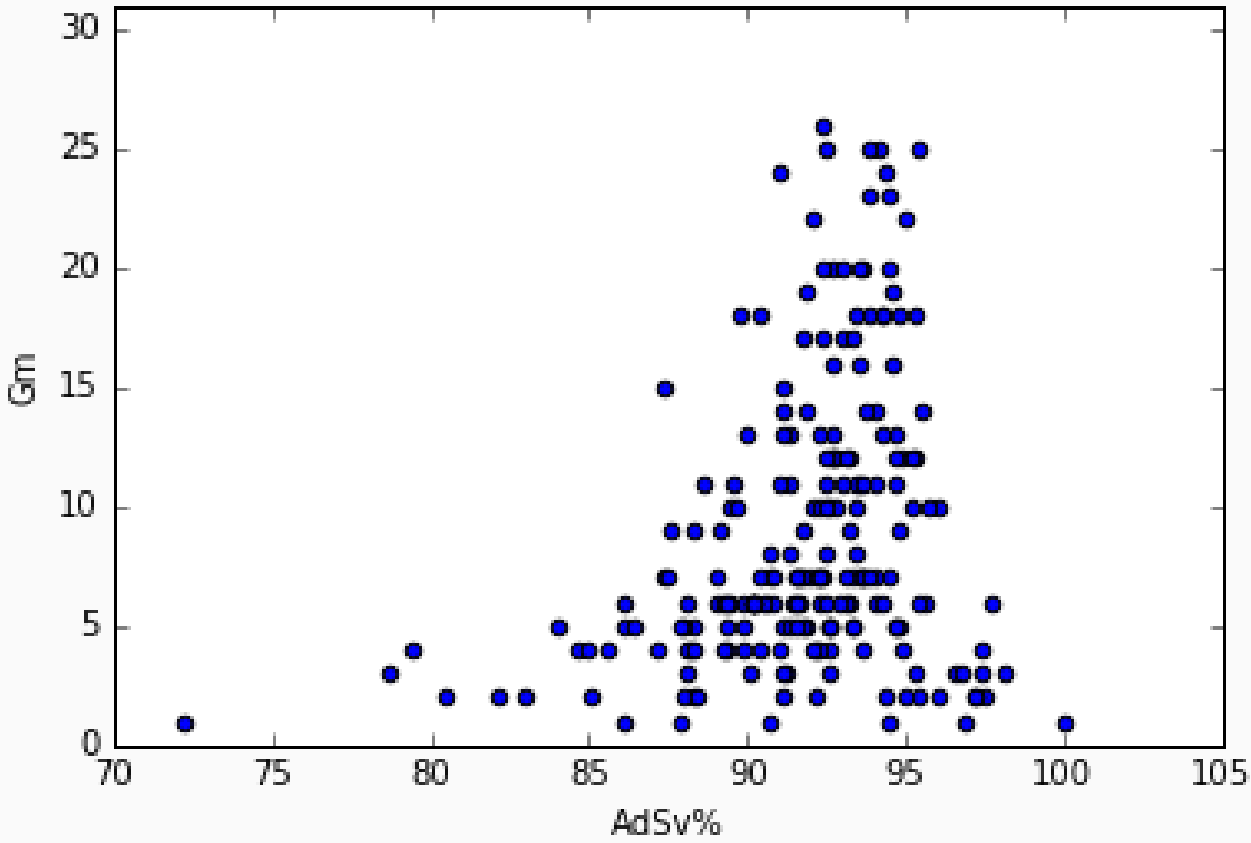
The presence of high correlation between goalies and their backups suggests that adjusting for both shot location and situation does not entirely isolate team effects*



*Alternative: The teams who find the best starters also find the best backups

Problem 2: Small sample size

For evaluating playoff performance (or backup goalies during the regular season) there is little sample size, especially if we want to adjust for every scenario



Problem 2: Small sample size

Method 1: Bayesian updating of binomial data

$$\alpha = \text{prior expectation of shots}$$

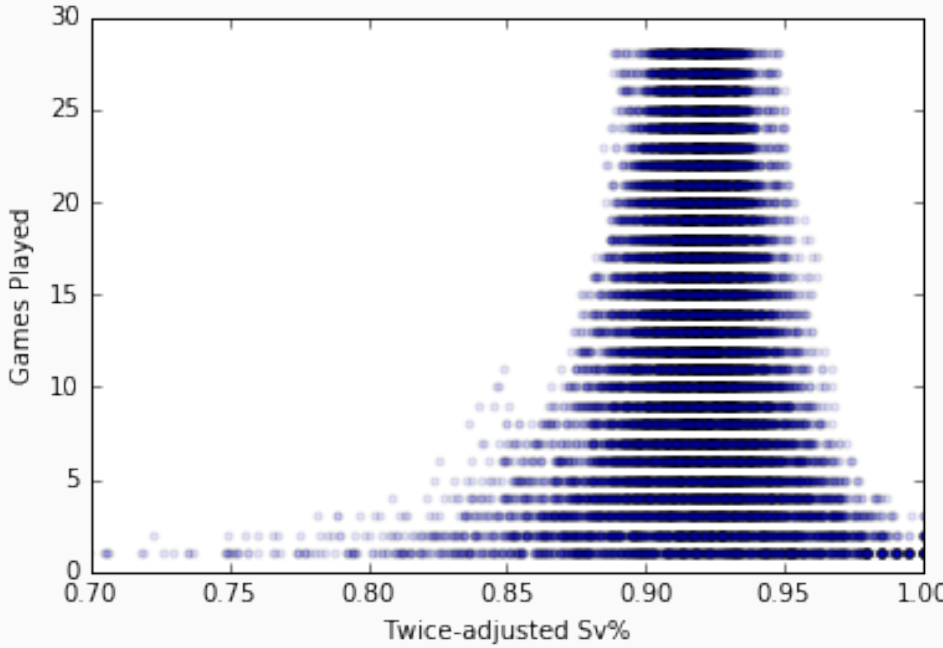
$$\beta = \text{prior expectation of goals}$$

$$\text{posterior} \sim \mathit{Beta}(\alpha, \beta)$$

$$\text{posterior mean} = \frac{\alpha + \mathit{saves}}{\alpha + \beta + \mathit{shots}}$$

I do this across zones and situations and use the league average as my prior, with a weight for sensitivity to the prior

Method 2: Empirical distribution to establish variance



Note that this is not a bootstrap distribution. I used the regular season to generate streaks from continuous actual games

Problem 2: Small sample size

```
def find_threshold(self, num_games, value):
    """Find where "value" would rank, within streaks of "num_games" played"""

    sample = self._streak_data

    if sample is None:
        raise LookupError, "Call build_streaks() before find_threshold()"

    if num_games not in sample:
        raise IndexError, "Streaks of {} games not found".format(num_games)

    items = len(sample[num_games])

    lower_bound = 1.0*sum(i >= value for i in sample[num_games])/items
    upper_bound = 1.0*sum(i > value for i in sample[num_games])/items

    return 1 - (lower_bound + upper_bound)/2
```

Fantasy Hockey

Putting the game in game theory

Understand your problem domain

Figure out what really matters. Restrictions on your team, how the scoring works, the key dynamics of gameplay, etc.

Predict what matters

Predict every stat (goals, hits, etc) for every player through (automatic) LASSO models*

Develop an optimization criteria

Team average across categories, standardized, weighted by predictability of the stat and diminishing in distance from average

Develop an optimization strategy

Think in terms of a **Bayesian Nash Equilibrium**, and model how my opponents draft

*LASSO is by no means the only (or even best) option. I used it to limit overfit.

Simplified example of fitting LASSO models

```
from sklearn import linear_model
import numpy as np

# data imported elsewhere
data = data.loc[numpy.isfinite(data['lag_gp_1'])]
clf = linear_model.LassoCV()

for stat in ["g", "a", "hits", "blocks", "ppp", "shg", "sog", "fow"]:
    res = clf.fit(data[predictors], data[stat])
    print "Results for target \"{}\"".format(stat)
    print "{0:<20}{1:.3f}".format("Intercept", res.intercept_)
    for index in range(len(predictors)):
        if abs(res.coef_[index]) > 0.01:
            print "{0:<20}{1:.3f}".format(predictors[index],
                                          res.coef_[index].item())
```

Thank you for your time!

Questions?