

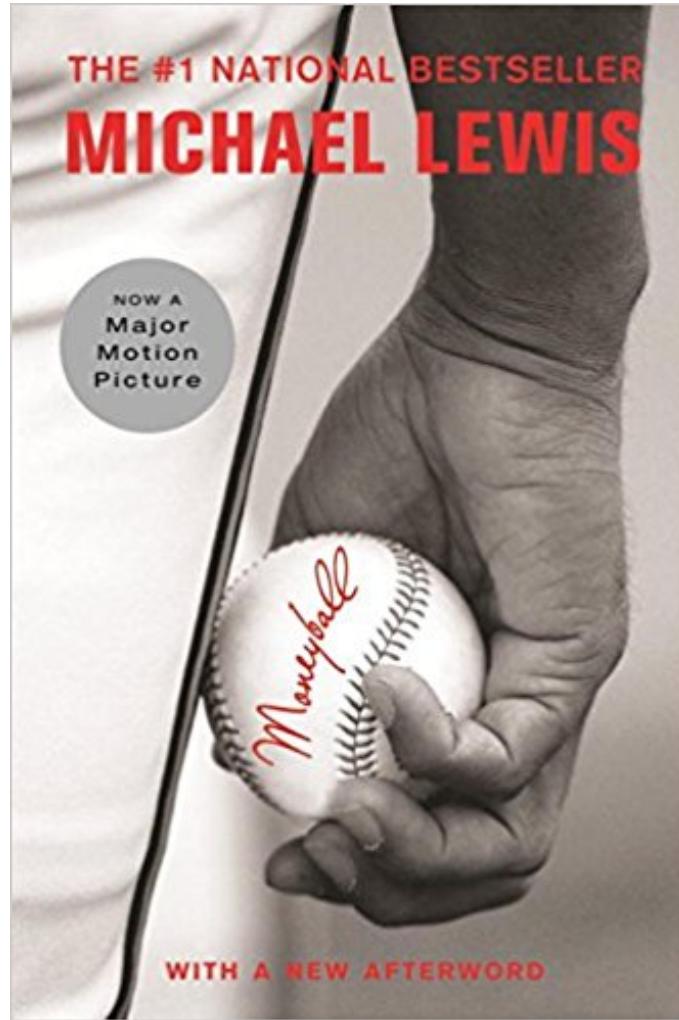
An R Engine for Real-Time Sports Analytics

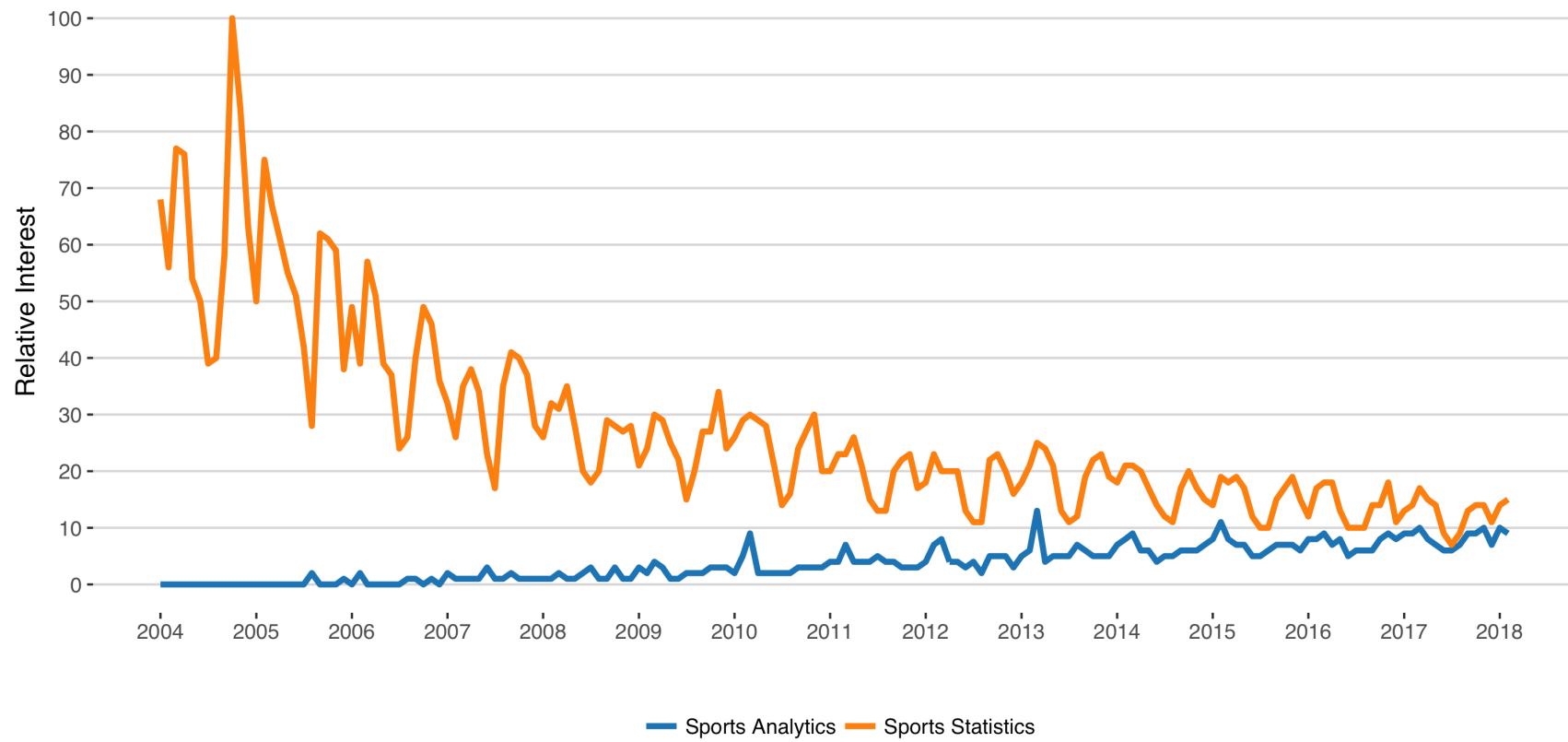
Stephanie Kovalchik

satRday 2018, Cape Town

About Me

- Senior Data Scientist for Tennis Australia's Game Insight Group
- Research Fellow at Victoria University
- Tennis Blogger at on-the-t.com
- @StatsOnTheT







Source: 2018 MIT Sloan Sports Analytics Conference

Tennis Analytics



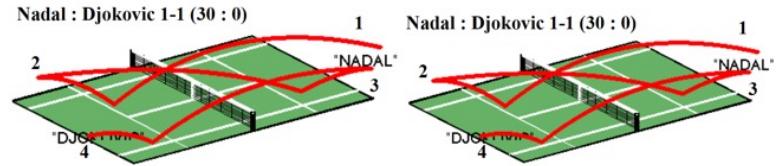
Source: 2012 MIT Sloan Sports Analytics Conference



"The Thin Edge of the Wedge": Accurately Predicting Shot Outcomes in Tennis using Style and Context Priors

Paper Track: Other Sports (Tennis)

Paper ID: 1475



Source: 2016 MIT Sloan
Sports Analytics Conference

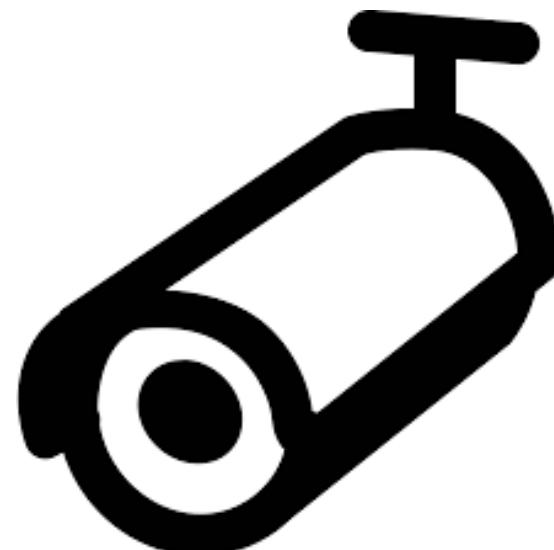


GIG Mission

Revolutionise tennis through data science.

GIG Data

Data Sources



Web Data



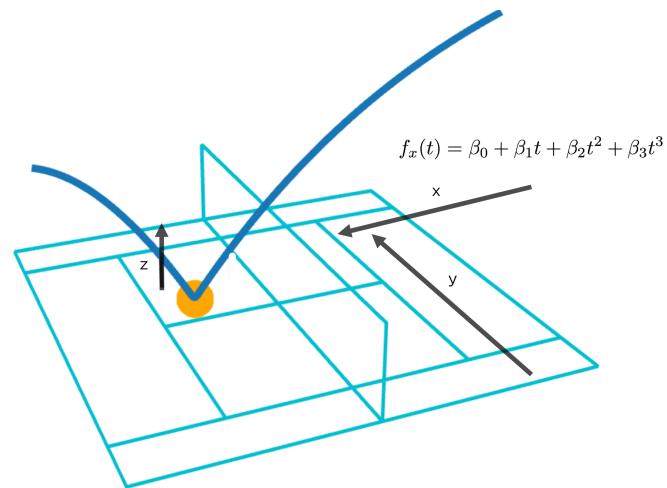
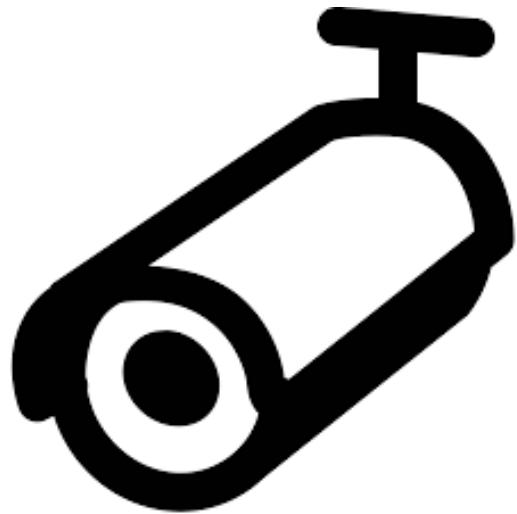
- Shot, point, and match-level performance data
- All pro matches since Open Era
- Automated Web scraping
- Stored in SQL database
- Main sources:
 - flashscore.com

Web Data



github.com/skoval/deuce

Tracking Data



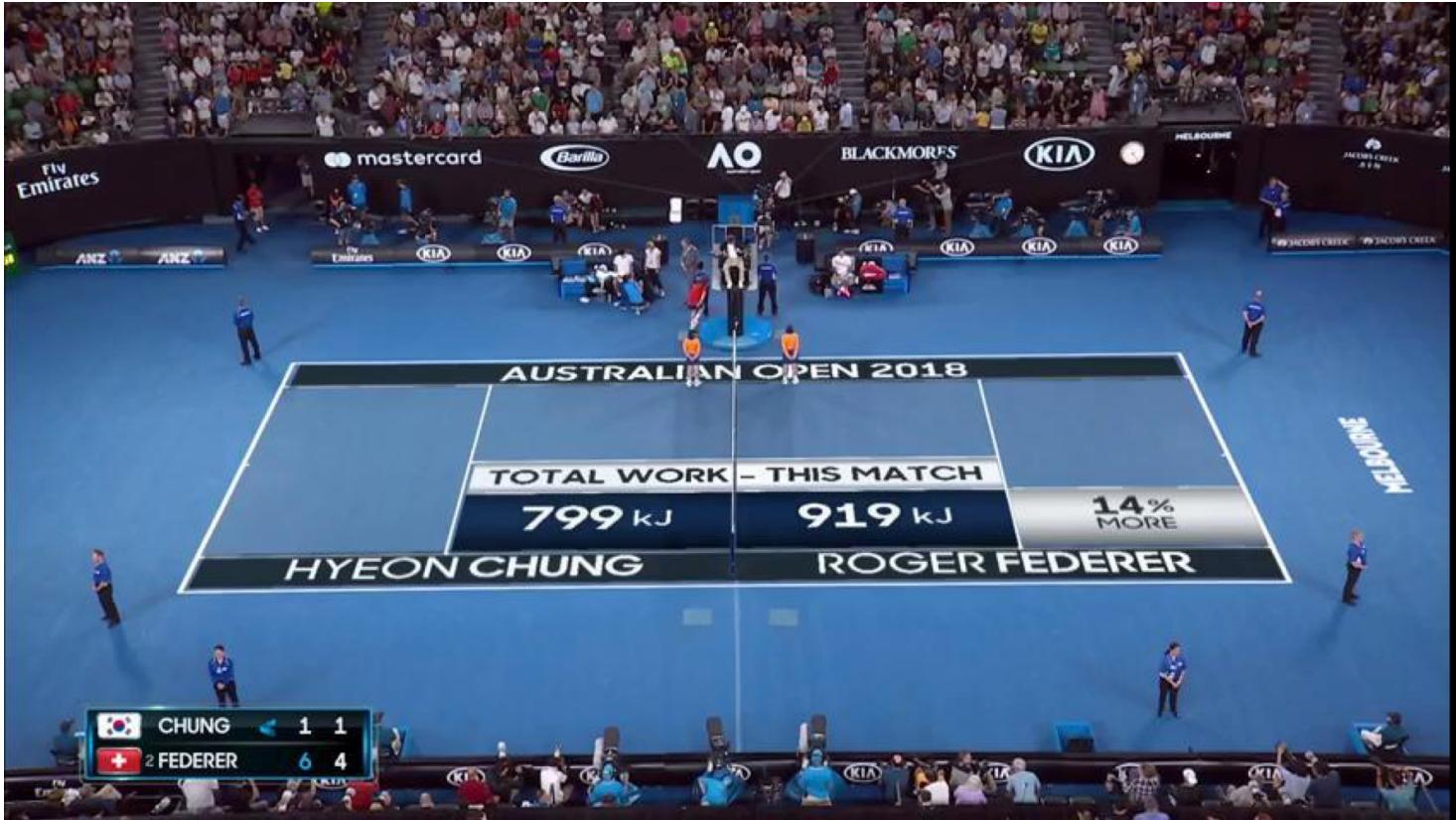
Tracking Data



- Proprietary
- Ball and player trajectories
- Australian Open Series, 2012 to Present
- Real-time collection since 2017
- Stored in SQL database

Event Operations





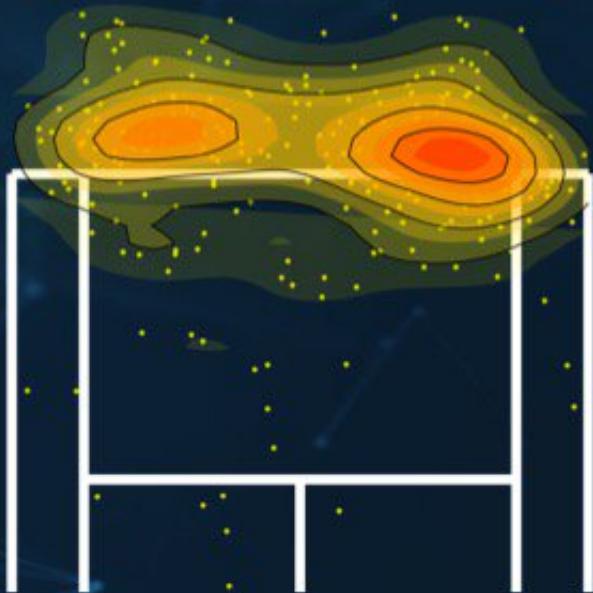


MEN'S FINAL 2018 #AUSOPEN

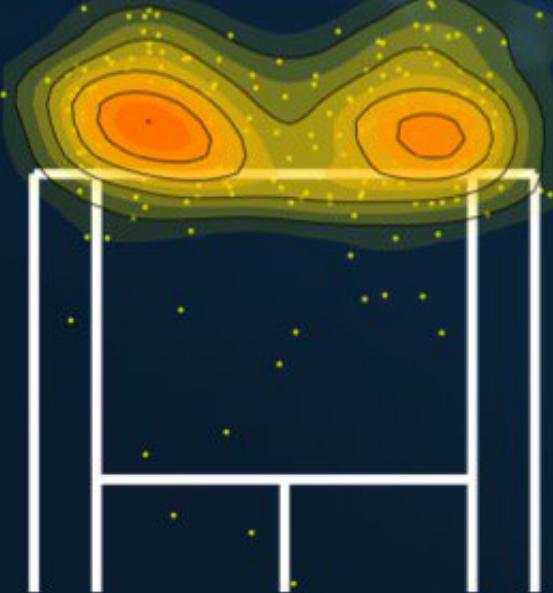


POSITIONING DURING RALLIES

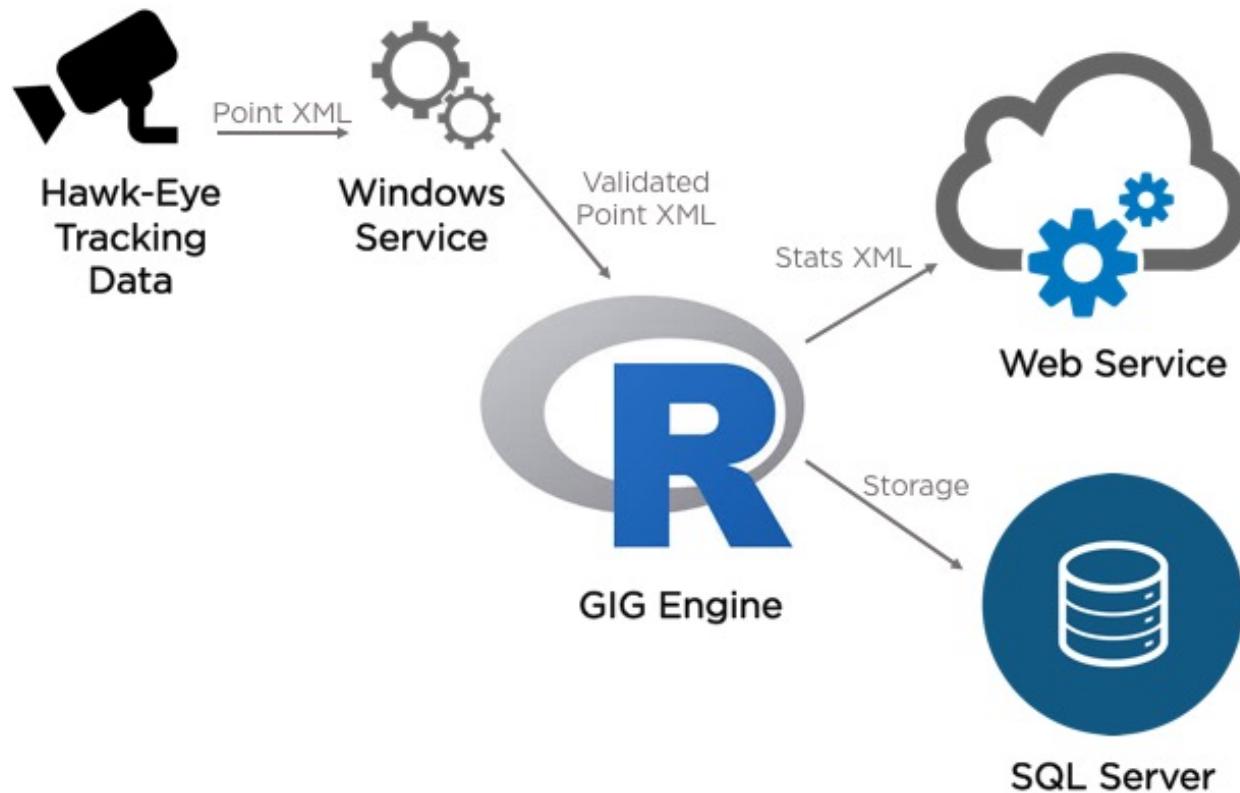
CILIC



FEDERER



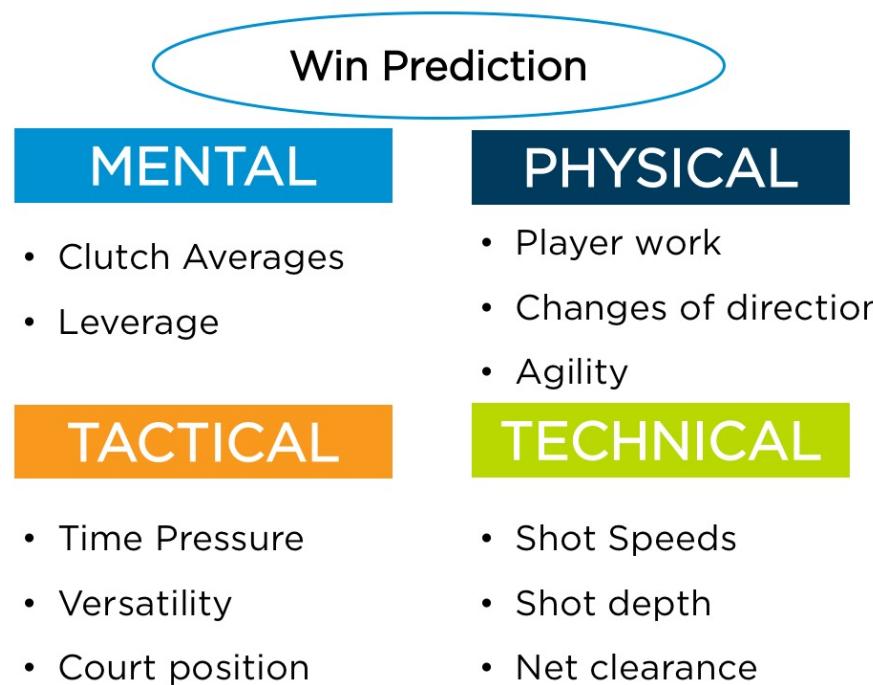
How We Get There...



Source XML: What We Get

Stats XML: What We Return

We derive point-level statistics on four dimensions of tennis using our own R package called the **GIGEngine**.



Demo: Tour GIGEngine

Modular Code

Modular Code

- You get some idea from the package layout that we break down our processes into small steps

Modular Code

- You get some idea from the package layout that we break down our processes into small steps
- This is a key design principle to our code

Modular Code

- You get some idea from the package layout that we break down our processes into small steps
- This is a key design principle to our code
- You can see this at work even better if we look at the main function that creates our XML output

Demo: Generating Output

Demo: Wrapper for Automation

Automation: Rscript

- Real-time can't happen manually!
- Thankfully, R has a number of tools for command-line processing
- **Rscript** is one of these
- It's a linux/unix command to invoke R scripts that comes with base R



Automation: Example

```
# Run file
library(GIGEngine)

args <- commandArgs(trailingOnly = TRUE)

xml.file <- args[1]
out.file <- args[2]
data.path <- args[3]
match.code <- args[4]

make_xml(
  xml.file = xml.file,
  out.file = out.file,
  data.path = data.path,
  match.code = match.code
)
```

Automation: Example

```
# Run file
library(GIGEngine)

args <- commandArgs(trailingOnly = TRUE)

xml.file <- args[1]
out.file <- args[2]
data.path <- args[3]
match.code <- args[4]

make_xml(
  xml.file = xml.file,
  out.file = out.file,
  data.path = data.path,
  match.code = match.code
)
```

```
Rscript --vanilla run.R <xml.file> <out.path> <storage.path>
<matchcode>
```

Demo: Automation

Delivering Output: Web Service

MatchServices

Click [here](#) for a complete list of operations.

GetMatchFeed

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
vendorID:	<input type="text"/>
password:	<input type="text"/>
matchCode:	<input type="text" value="MS701"/>

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /MatchServices.asmx HTTP/1.1
Host: services.tennis.com.au
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://webservice.aoapplications.net/GetMatchFeed"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMatchFeed xmlns="http://webservice.aoapplications.net">
      <vendorID>string</vendorID>
      <password>string</password>
      <matchCode>string</matchCode>
    </GetMatchFeed>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMatchFeedResponse xmlns="http://webservice.aoapplications.net">
      <GetMatchFeedResult>xml</GetMatchFeedResult>
    </GetMatchFeedResponse>
  </soap:Body>
</soap:Envelope>
```

Output XML

```
▼<Tournament MsgID="Match" Name="Australian Open" Year="2018" Today="2018-01-28">
  ▼<Match ID="MS701" CurrentSet="5">
    ▼<ClutchPoint>
      ▼<Team Number="1" SetsWon="2">
        ▼<Players>
          <Player ID="" ShortName="CILIC"/>
        ▼</Players>
        ▼<Scores GameScore="00" Points="4">
          ▼<Score Set="5">
            <SglStats ServerWonPct="63.5" ReturnerWonPct="28.9" FSPct="61.5" FSPTs="91" FSPTsWon="64" FSPTsWonPct="70.3" SSPTs="57" SSPTsWon="30" SSPTsWonPct="52.6"
              TotAces="16" DF="4" FSRetPts="85" SSRetPts="57" FSRetPtsWon="17" SSRetPtsWon="24" BrkPts="9" BrkPtsWon="2" TotPtsWon="135"/>
            </Score>
          ▼</Scores>
          <Point CumulativeClutchAverageServerWonPt="64" CumulativeClutchAverageReturnerWonPt="31.4" CumulativeClutchAverageTotPtsWon="47.5"
            CumulativeClutchAverageFSPTs="60.8" CumulativeClutchAverageBrkPtsWon="17.8" CumulativeClutchAverageBrkPts="17.3" CumulativeClutchAverageDF="2.5"
            CumulativeClutchAverageTotAces="8.6" CumulativeClutchAverageFSRetPtsWon="24.4" CumulativeClutchAverageFSPTsWon="70" CumulativeClutchAverageSSRetPtsWon="41"
            CumulativeClutchAverageSSPTsWon="54.8" PlayerWinProb="10.7" PlayerLastWinProb="1" PlayerBaselineWinProb="14.9" Swing="9.7" HighSwing="0"
            PlayerWinProbIfPointWon="14.4" PlayerWinProbIfPointLost="9.9" Leverage="4.5" HighLeverage="0"/>
        ▼</Team>
      ▼<Team Number="2" SetsWon="2">
        ▼<Players>
          <Player ID="" ShortName="FEDERER"/>
        ▼</Players>
        ▼<Scores GameScore="40" Points="4">
          ▼<Score Set="5">
            <SglStats ServerWonPct="71.1" ReturnerWonPct="36.5" FSPct="59.9" FSPTs="85" FSPTsWon="68" FSPTsWonPct="80" SSPTs="57" SSPTsWon="33" SSPTsWonPct="57.9"
              TotAces="26" DF="4" FSRetPts="91" SSRetPts="57" FSRetPtsWon="27" SSRetPtsWon="27" BrkPts="13" BrkPtsWon="6" TotPtsWon="155"/>
            </Score>
          ▼</Scores>
          <Point CumulativeClutchAverageServerWonPt="68.6" CumulativeClutchAverageReturnerWonPt="36" CumulativeClutchAverageTotPtsWon="52.5"
            CumulativeClutchAverageFSPTs="57.6" CumulativeClutchAverageBrkPtsWon="48.7" CumulativeClutchAverageBrkPts="18.2" CumulativeClutchAverageDF="3.1"
            CumulativeClutchAverageTotAces="17.2" CumulativeClutchAverageFSRetPtsWon="30" CumulativeClutchAverageFSPTsWon="75.6" CumulativeClutchAverageSSRetPtsWon="45.2"
            CumulativeClutchAverageSSPTsWon="59" PlayerWinProb="89.3" PlayerLastWinProb="99" PlayerBaselineWinProb="85.1" Swing="-9.7" HighSwing="0"
            PlayerWinProbIfPointWon="90.1" PlayerWinProbIfPointLost="85.6" Leverage="4.5" HighLeverage="0"/>
        ▼</Team>
      ▼</ClutchPoint>
    ▼<ClutchSetSummary>
      ▼<Players>
        <Player ID="" ShortName="CILIC"/>
      ▼</Players>
      ▼<Score Set="0">
        <SglStats ServerWonPct="63.5" ReturnerWonPct="28.9" FSPct="61.5" FSPTs="91" FSPTsWon="64" FSPTsWonPct="70.3" SSPTs="57" SSPTsWon="30" SSPTsWonPct="52.6"
          TotAces="16" DF="4" FSRetPts="85" SSRetPts="57" FSRetPtsWon="17" SSRetPtsWon="24" BrkPts="9" BrkPtsWon="2" TotPtsWon="135"/>
      ▼</Score>
      <Point CumulativeClutchAverageServerWonPt="64" CumulativeClutchAverageReturnerWonPt="31.4" CumulativeClutchAverageTotPtsWon="47.5"
        CumulativeClutchAverageFSPTs="60.8" CumulativeClutchAverageBrkPtsWon="17.8" CumulativeClutchAverageBrkPts="17.3" CumulativeClutchAverageDF="2.5"
        CumulativeClutchAverageTotAces="8.6" CumulativeClutchAverageFSRetPtsWon="24.4" CumulativeClutchAverageFSPTsWon="70" CumulativeClutchAverageSSRetPtsWon="41"
```

GIG Web Service

Tips: R Software for Real Time

Tips: R Software for Real Time

- Think modularly!

Tips: R Software for Real Time

- Think modularly!
- Profile bottlenecks using `lineprof` package

Tips: R Software for Real Time

- Think modularly!
- Profile bottlenecks using `lineprof` package
- Embrace the tidyverse

Tips: R Software for Real Time

- Think modularly!
- Profile bottlenecks using `lineprof` package
- Embrace the tidyverse
- Further optimize with:
 - Fast handling of `data.frames` (e.g. `datatable`)
 - Parallel processing (`parallel`, `foreach`, etc.)
 - Write bottlenecks in C++ and use in R through `Rcpp`

GIG Reports



The following GIG statistics are averages over the rounds of the current Australian Open.

Full Report

GIG Reports

- Driven by R Markdown
- Automated with Rscript
- Multiple Reports for Media:
 - Player Profiles
 - Match Previews
 - Match Reviews

Template

```
--  
params:  
  player: "DJOKOVIC"  
  opponent: "NADAL"  
  mens: TRUE  
  player_title: "Novak Djokovic"  
  opponent_title: "Rafael Nadal"  
  player_img: "djokovic.jpg"  
  opponent_img: "nadal.jpg"  
  stats: "~/Software/GIGEngine/output/mens/stats.RData"  
  times: "~/Software/GIGEngine/output/mens/times.RData"  
output:  
  html_document:  
    css: /Users/skovalchik/Software/GIGEngine/reports/nobreaks_v4.css  
--  
  
<!-- Header -->  
<div>  
    
</div>
```

Demo: Match Report

Match Report Script

```
library(rmarkdown)

options(warn = -1)

args <- commandArgs(trailingOnly = TRUE)

arg1 <- args[[1]]

arg2 <- args[[2]]

arg3 <- args[[3]]

arg4 <- args[[4]]

rmarkdown::render("~/Software/GIGEngine/reports/matchreports_short/match_review.Rmd")
```

Demo: Report Automation

Tips: Real-time Rmarkdown

Tips: Real-time Rmarkdown

- Think of your document as a function

Tips: Real-time Rmarkdown

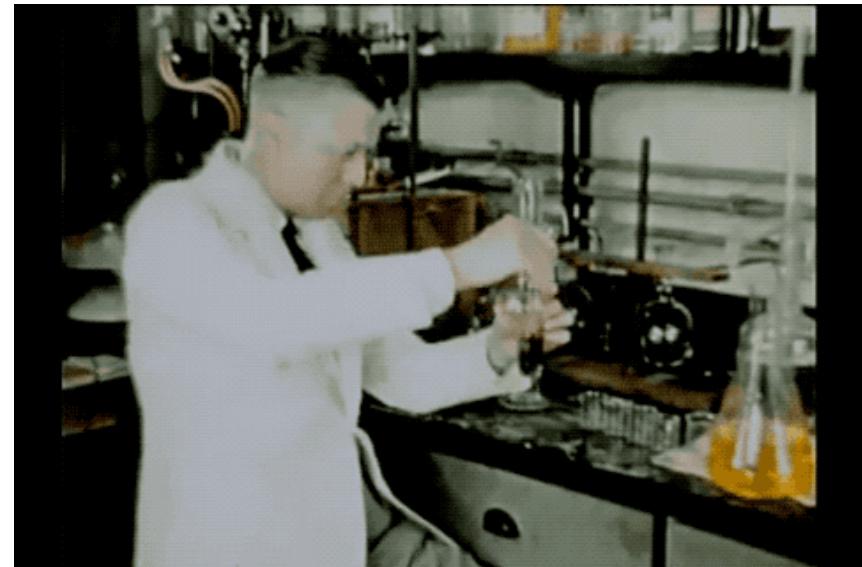
- Think of your document as a function
- Make your summaries & plots generalizable

Tips: Real-time Rmarkdown

- Think of your document as a function
- Make your summaries & plots generalizable
- Separate content from style with a CSS style sheet

Making an Application Robust

- Test, test, test!
- Anticipate how things can go wrong and build that into your testing
- If you are not sure what could go wrong, run a representative group of input through the process and document what goes wrong



Resources

- **Rscript**: <https://stat.ethz.ch/R-manual/R-devel/library/utils/html/Rscript.html>
- **lineprof**
- **datatable**
- Code Optimization: <http://adv-r.had.co.nz/Performance.html>

Stephanie Kovalchik

Sports Data Scientist

[@StatsOnTheT](https://twitter.com/StatsOnTheT)

