Sabermetrics with R
Type of X-informatics: Sports Informatics

Team: Suganya Varadarajan, Bharati Patidar, Sunil Agrawal
Suganya Varadarajan - svaradar@umail.iu.edu
Bharati Patidar - bpatidar@umail.iu.edu
Sunil Agrawal - suniagra@umail.iu.edu

Background

Baseball sport is predominantly built around individuals than a team effort. We can use individual player's data to analyze how a particular player performs and his contribution to the team. PITCHf/x is a pitch tracking system, by Sportsvision company and it is installed in every MLB stadium. It tracks the velocity, movement, release point, spin, and location of every pitch thrown in a game. This data is publicly available since 2007 and this data is very precise such that it can be analyzed and used for predictions and make decisions on a players or a game play.

Problem Definition

Goal

Our project is to analyze PITCHf/x baseball data, and other baseball source raw data and produce informative charts and graphs using R programming about player's performances. The data set covers for July to Sep 2015 top 5 pitchers for analysis. The key baseball metrics used in the project are FIP (Field independent pitching), and ERA (Earned Run Average). This project also analyzes the pitchball types used by 2015 top 5 pitchers.

Definition of FIP: It is a measurement of a pitcher's performance that strips out the role of defense, luck, and sequencing, making it a more stable indicator of how a pitcher actually performed over a given period of time than a runs allowed based statistic that would be highly dependent on the quality of defense played behind him, for exam. Overall FIP captures most pitchers' true performance quite well.
Formula: The formula is (HR*13+(BB+HBP-IBB)*3-K*2)/IP, plus a league-specific factor (usually around 3.2) to round out the number to an equivalent ERA number.

Home runs times 13 plus three times walks plus hit batsmen minus two times strikeouts, divided by innings pitched, plus a league constant to assure that the league's average FIP equals the league's average ERA

Definition: This baseball statistic represents the amount of earned runs, on average, that a pitcher allows for every nine innings he pitches. A pitcher's ERA is one of the most fundamental stats used to compare pitchers. ERA is calculated by dividing the pitcher's earned runs allowed by the total number of innings pitched and multiplying that total by 9.

Formula: Earned Run Average = 9 x (Earned Runs Allowed/Innings Pitched)

Tasks involved

1. Project Environment setup:
   - R environment setup – Install R and R studio for IDE
   - Deploy R packages needed for analysis:

     ```
     #Install these packages
     install.packages("xlsx")
     install.packages("plyr")
     install.packages("cowplot")
     install.packages("pitchRx")
     ```

   -
     ```
     #Load all the packages
     library(pitchRx)
     library(dplyr)
     library(plyr)
     library(RSQLite)
     library(ggplot2)
     library(grid)
     library(gridExtra)
     require(cowplot)
     library(xlsx)
     library(data.table)
     ```

2. Deliverable 1 Tasks:

   a. Set up game database connection and load PITCHf/x data into R data frames:

   ```
   my_db <- src_sqlite("GamedayDB.sqlite3", create = TRUE)
   scrape(start = "2015-07-01", end = "2015-07-01",suffix = "inning/inning_all.xml",
   connect = my_db$con)
   ```

   b. For each of the above 5 pitchers, the various pitch types used in the games for #July 2015 – Aug 2015 are analyzed and their accuracy is measured in the chart.

   - Data table: What types of pitches did they throw in the games? Field pitch_type in PITCHf/x data provide this information.
   - Data table: What were the outcomes of these pitches? Example: Field des in PITCHf/x data provides this information.
   - Plot faceted graph: What were the locations of these pitches ?

   The PITCHf/x data classifies the pitches as CH(changeup), #CU(Curve ball), FA (Fast ball), FC(Clutter), FF(Four-seam fastball), PO(Pitchout),SL(Slider) etc. It uses the field des in PITCHf/x data to determine the type of pitches the pitchers had thrown, and the outcome of pitchers are shown. For all the 5 pitchers, the charts are shown side-by-side for comparison.

3. Deliverable 2 Tasks:
   a. For Field independent metric (FIP) and Earned Run average (ERA), the project uses raw data from ESPN MLB team website for July 2015 – Sep 2015:
      Sources:
      http://espn.go.com/mlb/team/stats/pitching/_/name/chw/chicago-white-sox
      http://espn.go.com/mlb/team/stats/pitching/_/name/wsh/split/45/washington-nationals
   - Load this spreadsheet data (xlsx) into R data frames
     FIPstats<-read.xlsx("Pitcher2015FIP.xlsx", 1)
   - The raw data has ERA already calculated and the project uses this value for comparison with FIP metric.
   - Calculate FIP metric for July,Aug, Sep for top 5 pitchers of 2015:
     - Clayton Kershaw
     - David Price
     - Jake Arrieta
     - Max Scherzer
     - Chris Sale
     - Plot the graphs – FIP and ERA for side-by-side comparison

       According to Fangraphs (Source: http://www.fangraphs.com/library/pitching/fip/)
       FIP rating is based on the following values:

Rating	FIP
Excellent	2.90
Great	3.20
Above Average	3.50
Average	3.80
Below Average	4.10
Poor	4.40
Awful	4.70

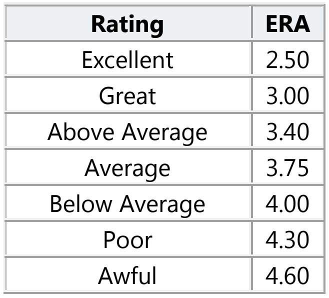
4. Deliverable 3 Tasks:
   ERA(Earned run Average) baseball metric: The project analyzes 2015 World Series team NY Mets and compare ERA metrics for their pitchers.

   For creativity, background image of NY Mets is added to chart.
   ESPN MLB team raw data is extracted for NY Mets pitchers for overall regular season and stored in R data frame as follows:
   Source: http://espn.go.com/mlb/team/stats/pitching/_/name/nym/new-york-mets

   NYMetsPitchers <- c("Bartolo Colon","Hansel Robles","Noah Syndergaard","Alex Torres","Logan Verrett","Matt Harvey","Sean Gilmartin","Jacob deGrom")
   NYMetsERA <-c(4.16,3.67,3.24,3.15,3.03,2.71,2.67,2.54)

   Using R ggplot, plot ERA for these pitchers and compare.

According to Fangraphs (Source: http://www.fangraphs.com/library/pitching/era/), ERA rating is as follows:



Rating	ERA
Excellent	2.50
Great	3.00
Above Average	3.40
Average	3.75
Below Average	4.00
Poor	4.30
Awful	4.60

Software used and technical awareness

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and Mac OS. R has built-in packages for doing analysis and plotting rich graphs which is otherwise tedious with other tools such as MS Excel. CRAN (The Comprehensive R Archive Network) is a network of ftp/web servers that has up-to-date R versions software and documentation . CRAN is used to install the following R packages needed for Project setup:

```
    #Install these packages
    install.packages("xlsx")
    install.packages("plyr")
    install.packages("cowplot")
    install.packages("pitchRx")

-   #Load all the packages
    library(pitchRx)
    library(dplyr)
    library(plyr)
    library(RSQLite)
    library(ggplot2)
    library(grid)
    library(gridExtra)
    require(cowplot)
    library(xlsx)
    library(data.table)
```

R is good for statistical analysis, and at the same time to extract massive amount of data, it takes lot of memory for processing and hence it is dependent on the type of resource(machine environment) being used. Our project did not use H2O which is a Java Virtual Machine that is optimized for doing “in memory” processing of distributed, parallel machine learning algorithms on clusters. It could help in loading the data in parallel.

Deliverable task 1:

Pitchrx R package provides tools for collecting MLB Gameday data and visualizing.
Its scrape() function is used to collect all PITCHf/x data recorded on a particular day or series of days..

```
my_db <- src_sqlite("GamedayDB.sqlite3", create = TRUE)
scrape(start = "2015-07-01", end = "2015-08-30",suffix = "inning/inning_all.xml", connect = my_db$con)
```

The tables pitch and atbat are used to get all pitch and batting related information.
The pitch data fields that are used in this project for analysis are:
(Source: https://fastballs.wordpress.com/2007/08/02/glossary-of-the-gameday-pitch-fields)

- des: a brief text description of the result of the pitch: Ball; Ball In Dirt; Called Strike; Foul; Foul (Runner Going); Foul Tip; Hit by Pitch; In play, no out; In play, out(s); In play, run(s); Intent Ball; Pitchout; Swinging Strike; Swinging Strike (Blocked).
- type: a one-letter abbreviation for the result of the pitch: B, ball; S, strike (including fouls); X, in play.

```
IUpitch11 <-select(tbl(my_db,"pitch"),pitch_type, px, pz, des, num, gameday_link,inning,type)
IUatbat11 <-select(tbl(my_db,"atbat"),batter,pitcher_name, batter_name, num, gameday_link, event, stand,inning,b,s)
```

R data frame IUdata is constructed by joining pitch and atbat ofPITCHF/x data as follows:

```
IUdata <- collect(inner_join(IUpitch11, IUatbat11))
```

From this data frame, pitcher specifc data is extracted and analyzed. For example, here is the code snippet that uses Jake Arrieta’s specific game day records which gives details of pitch type he threw and their outcomes.

```
#Jake Arrietta
#What types of pitches did Jake Arrietta throw in this game?
IUJake <- IUdata[IUdata$pitcher_name %in% jake,]
with(IUJake, table(pitch_type))
```

Once this is extracted, his statistics are plotted in the graph using R package strikeFX:

```
g0<-strikeFX(IUJake,  point.alpha = 1 ,
          layer=facet_wrap(~pitch_type, ncol=2)) +  ggtitle("Jake Arrieta") +
  theme(legend.position = "right", legend.direction = "vertical") +
  theme_bw()
```

Similarly, all the remaining 4 pitchers are analyzed and finally, all the pitchers graphs are grouped and shown as a single graph using R package cowplot and its method plot _grid:

```
plot_grid(g0,g1,g2,g3,g4,  align='h')
```

Deliverable task 2 and task 3:

ESPN MLB game data is retrieved as spreadsheet and loaded into R dataframe. The raw data is shown below that has all the data neeed to calculate FIP (Field independent pitching) and ERA is provided. FIP metric is calculated in R as follows:

```
#Calculate FIP by month
FIPStatsTable$FIPNumerator = ((13*FIPStatsTable$HR) + (3*(FIPStatsTable$BB + FIPStatsTable$H)) -(2*(FIPStatsTable$SO)) )
FIPStatsTable$FIPDenominator    =                    (FIPStatsTable$IP + 3.2)
FIPStatsTable$FIP = FIPStatsTable$FIPNumerator/FIPStatsTable$FIPDenominator
FIPStatsTable$month<-factor(FIPStatsTable$month)
```

Pitcher	IP	H	ER	HR	BB	SO	K/9	P/GS	WHIP	ERA	month	year	Monthname
Max Scherzer	28	33	20	7	7	37	11.89	671.8	1.43	6.43	8	2015	August
David Price	43.1	34	11	3	8	50	10.38	194.5	0.97	2.28	8	2015	August
Chris Sale	33.2	20	12	3	10	52	13.9	664.6	0.89	3.21	8	2015	August
Jake Arrieta	42.1	19	2	0	10	43	9.14	573	0.69	0.43	8	2015	August
Clayton Kershaw	45	29	7	2	6	51	10.2	565.3	0.78	1.4	8	2015	August
Max Scherzer	39.2	33	15	6	5	42	9.53	559.8	0.96	3.4	7	2015	July
David Price	0	0	0	0	0	0	0	0	0	0	7	2015	July
Chris Sale	34.1	42	16	4	3	36	9.44	664.6	1.31	4.19	7	2015	July
Jake Arrieta	42.2	28	9	1	11	44	9.28	573	0.91	1.9	7	2015	July
Clayton Kershaw	33	19	1	0	2	45	12.27	848	0.64	0.27	7	2015	July
Max Scherzer	41.2	37	14	7	8	50	10.8	559.8	1.08	3.02	9	2015	September
David Price	31	23	8	1	10	37	10.74	233.4	1.06	2.32	9	2015	September
Chris Sale	30.1	40	17	7	6	38	11.27	664.6	1.52	5.04	9	2015	September
Jake Arrieta	40	20	2	1	4	39	8.78	687.6	0.6	0.45	9	2015	September
Clayton Kershaw	44	27	9	2	9	58	11.86	565.3	0.82	1.84	9	2015	September

Once R data frame FIPStatsTable is updated with FIP metric calculated values, all the 5 pitchers FIP and ERA are shown side-by-side in the graph using ggplot and finally placed together using plot_grid().

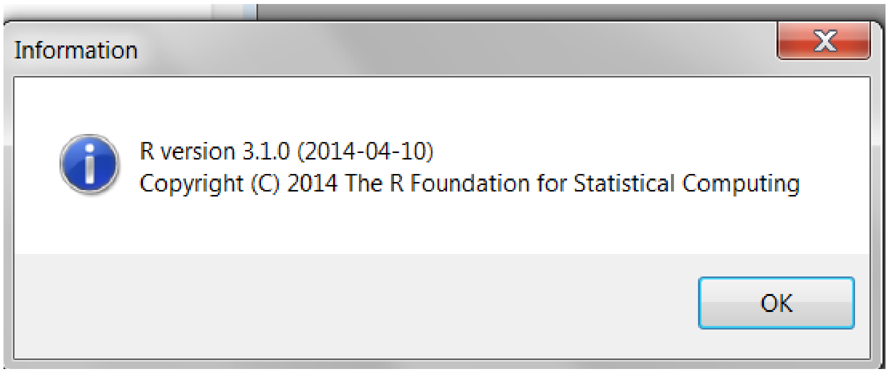
```
#Plot the graphs
g7 <- ggplot(data=FIPStatsTable, aes(x=month, y=abs(FIP), group = Pitcher, color = Pitcher)) +
  geom_line(size=2) +
  geom_point( size=4, shape=21, fill="white")+ggtitle("2015 FIP Comparison for July-Sep ")+
  scale_y_continuous("FIP",limits=c(0,9))
```

```
g8 <- ggplot(data=FIPStatsTable, aes(x=month, y=abs(ERA), group = Pitcher, color = Pitcher))
+
  geom_line(size=2) +
  geom_point( size=4, shape=21, fill="white")+ggtitle("2015 ERA Comparison for July-Sep ")+
  scale_y_continuous("ERA",limits=c(0,9))

plot_grid(g7, g8, align="h")plot_grid(g7, g8, align="h")
```

Produce Results

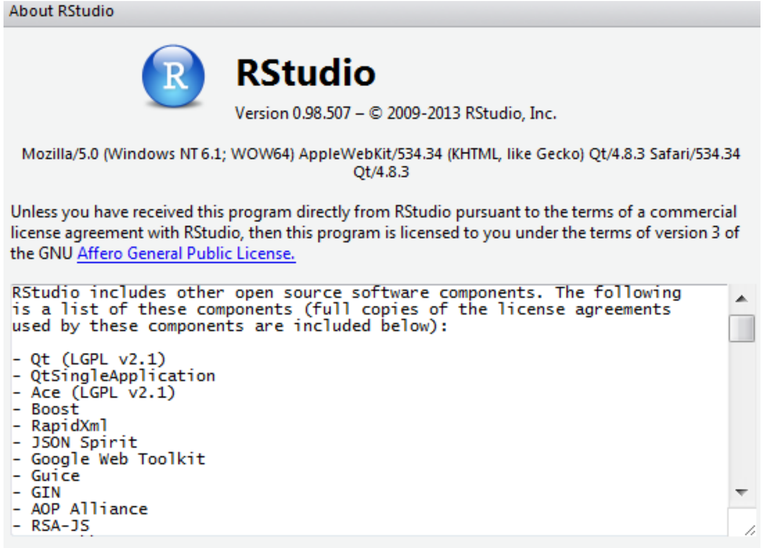
Software installation

R version: 3.1.0



R version 3.1.0 (2014-04-10)
Copyright (C) 2014 The R Foundation for Statistical Computing

R Studio IDE :



Version 0.98.507 – © 2009-2013 RStudio, Inc.

Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.34 (KHTML, like Gecko) Qt/4.8.3 Safari/534.34 Qt/4.8.3

Unless you have received this program directly from RStudio pursuant to the terms of a commercial license agreement with RStudio, then this program is licensed to you under the terms of version 3 of the GNU Affero General Public License.

```
RStudio includes other open source software components. The following
is a list of these components (full copies of the license agreements
used by these components are included below):

- Qt (LGPL v2.1)
- QtSingleApplication
- Ace (LGPL v2.1)
- Boost
- RapidXml
- JSON Spirit
- Google Web Toolkit
- Guice
- GIN
- AOP Alliance
- RSA-JS
```

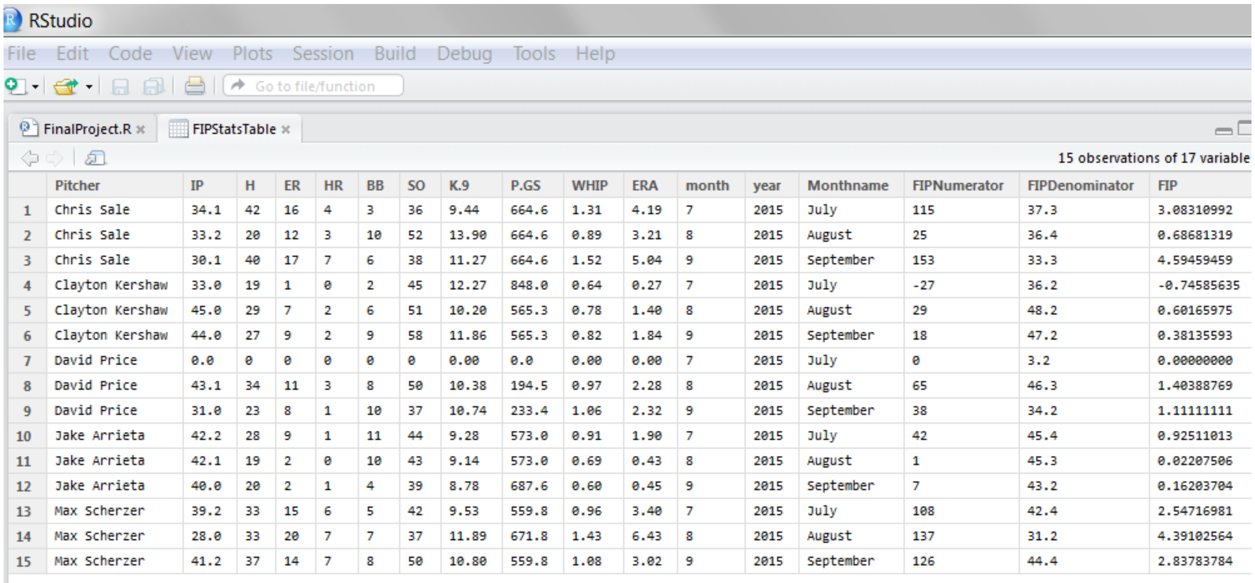
Proofs of R Packages installation

```
> install.packages("xlsx")
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/xlsx_0.5.7.zip'
Content type 'application/zip' length 400756 bytes (391 Kb)
opened URL
downloaded 391 Kb

package ‘xlsx’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Suganya\AppData\Local\Temp\RtmpAh4eKv\downloaded_packages
> install.packages("plyr")
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/plyr_1.8.3.zip'
Content type 'application/zip' length 1114002 bytes (1.1 Mb)
opened URL
downloaded 1.1 Mb

package ‘plyr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Suganya\AppData\Local\Temp\RtmpAh4eKv\downloaded_packages
> install.packages("cowplot")
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/cowplot_0.4.0.zip
'
Content type 'application/zip' length 987311 bytes (964 Kb)
opened URL
downloaded 964 Kb

package ‘cowplot’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Suganya\AppData\Local\Temp\RtmpAh4eKv\downloaded_packages
> install.packages("pitchRx")
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/pitchRx_1.8.1.zip
'
Content type 'application/zip' length 991369 bytes (968 Kb)
opened URL
downloaded 968 Kb

package ‘pitchRx’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Suganya\AppData\Local\Temp\RtmpAh4eKv\downloaded_packages

> library(pitchRx)
Loading required package: ggplot2
Warning messages:
1: package ‘pitchRx’ was built under R version 3.1.3
2: package ‘ggplot2’ was built under R version 3.1.3
> library(plyr)
Warning message:
package ‘plyr’ was built under R version 3.1.3
> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:plyr’:

    arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

The following object is masked from ‘package:stats’:
    filter

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

Warning message:
package ‘dplyr’ was built under R version 3.1.1
> library(RSQLite)
Loading required package: DBI
Warning messages:
1: package ‘RSQLite’ was built under R version 3.1.3
2: package ‘DBI’ was built under R version 3.1.1
> library(ggplot2)
> library(grid)
> library(gridExtra)
Warning message:
package ‘gridExtra’ was built under R version 3.1.2
> require(cowplot)
Loading required package: cowplot

Attaching package: ‘cowplot’

The following object is masked from ‘package:ggplot2’:

    ggsave

Warning message:
package ‘cowplot’ was built under R version 3.1.3
> library(xlsx)
Loading required package: rJava
Loading required package: xlsxjars
Warning messages:
1: package ‘xlsx’ was built under R version 3.1.3
2: package ‘rJava’ was built under R version 3.1.1
3: package ‘xlsxjars’ was built under R version 3.1.1
> library(data.table)
data.table 1.9.2  For help type: help("data.table")

Attaching package: ‘data.table’

The following objects are masked from ‘package:dplyr’:

    between, last

Warning message:
package ‘data.table’ was built under R version 3.1.1
```

Proof of Results in texts, figures, graphs or tables:

IUdata – R dataframe that holds all PITCHf/x data set for July 2015 – Aug 2015 all gameday data.

FIP metric that is calculated and ERA stored in R data frame for July 2015 – Sep 2015 for to 2015 5 pitchers in regular season:

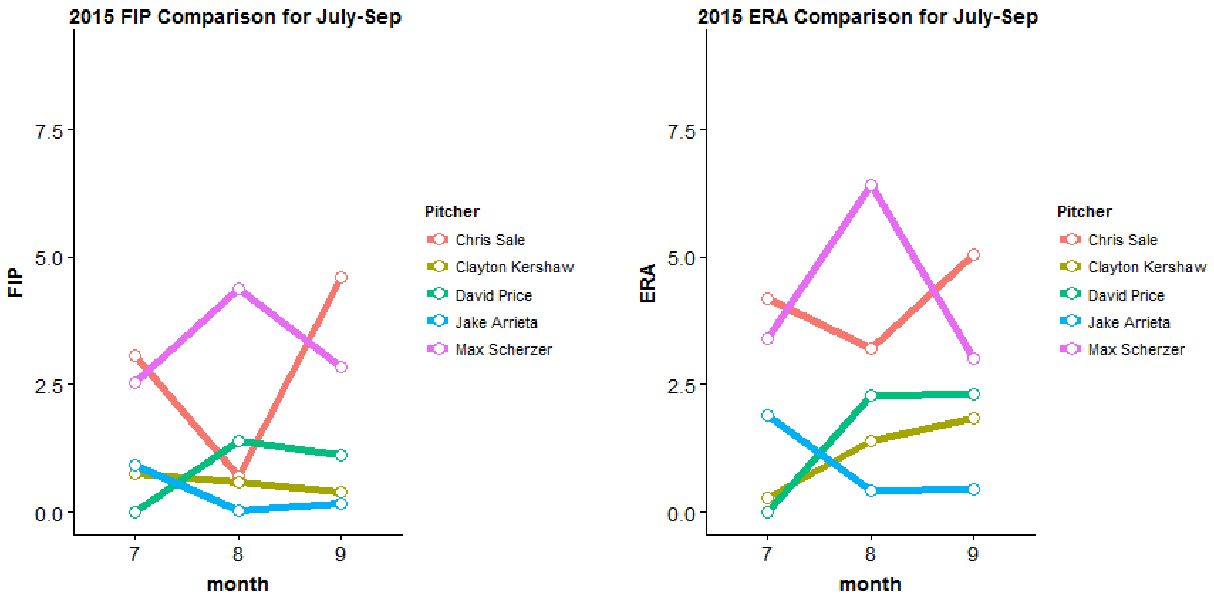


15 observations of 17 variable

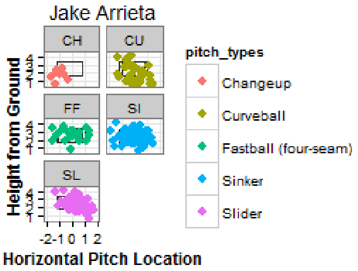
	Pitcher	IP	H	ER	HR	BB	SO	K.9	P.GS	WHIP	ERA	month	year	Monthname	FIPNumerator	FIPDenominator	FIP
1	Chris Sale	34.1	42	16	4	3	36	9.44	664.6	1.31	4.19	7	2015	July	115	37.3	3.08310992
2	Chris Sale	33.2	20	12	3	10	52	13.90	664.6	0.89	3.21	8	2015	August	25	36.4	0.68681319
3	Chris Sale	30.1	40	17	7	6	38	11.27	664.6	1.52	5.04	9	2015	September	153	33.3	4.59459459
4	Clayton Kershaw	33.0	19	1	0	2	45	12.27	848.0	0.64	0.27	7	2015	July	-27	36.2	-0.74585635
5	Clayton Kershaw	45.0	29	7	2	6	51	10.20	565.3	0.78	1.40	8	2015	August	29	48.2	0.60165975
6	Clayton Kershaw	44.0	27	9	2	9	58	11.86	565.3	0.82	1.84	9	2015	September	18	47.2	0.38135593
7	David Price	0.0	0	0	0	0	0	0.00	0.0	0.00	0.00	7	2015	July	0	3.2	0.00000000
8	David Price	43.1	34	11	3	8	50	10.38	194.5	0.97	2.28	8	2015	August	65	46.3	1.40388769
9	David Price	31.0	23	8	1	10	37	10.74	233.4	1.06	2.32	9	2015	September	38	34.2	1.11111111
10	Jake Arrieta	42.2	28	9	1	11	44	9.28	573.0	0.91	1.90	7	2015	July	42	45.4	0.92511013
11	Jake Arrieta	42.1	19	2	0	10	43	9.14	573.0	0.69	0.43	8	2015	August	1	45.3	0.02207506
12	Jake Arrieta	40.0	20	2	1	4	39	8.78	687.6	0.60	0.45	9	2015	September	7	43.2	0.16203704
13	Max Scherzer	39.2	33	15	6	5	42	9.53	559.8	0.96	3.40	7	2015	July	108	42.4	2.54716981
14	Max Scherzer	28.0	33	20	7	7	37	11.89	671.8	1.43	6.43	8	2015	August	137	31.2	4.39102564
15	Max Scherzer	41.2	37	14	7	8	50	10.80	559.8	1.08	3.02	9	2015	September	126	44.4	2.83783784

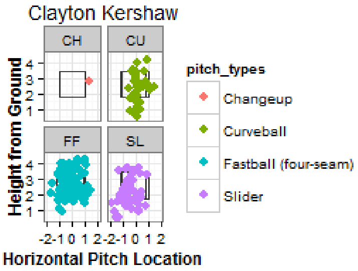
Results

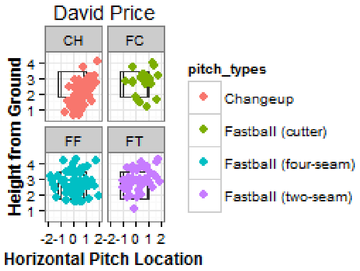
Deliverable 1:

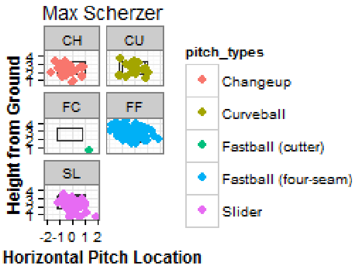


Deliverable 2:









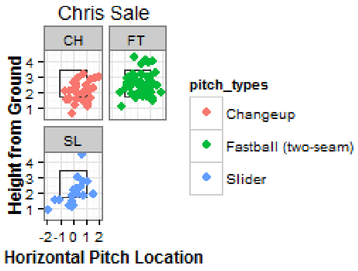
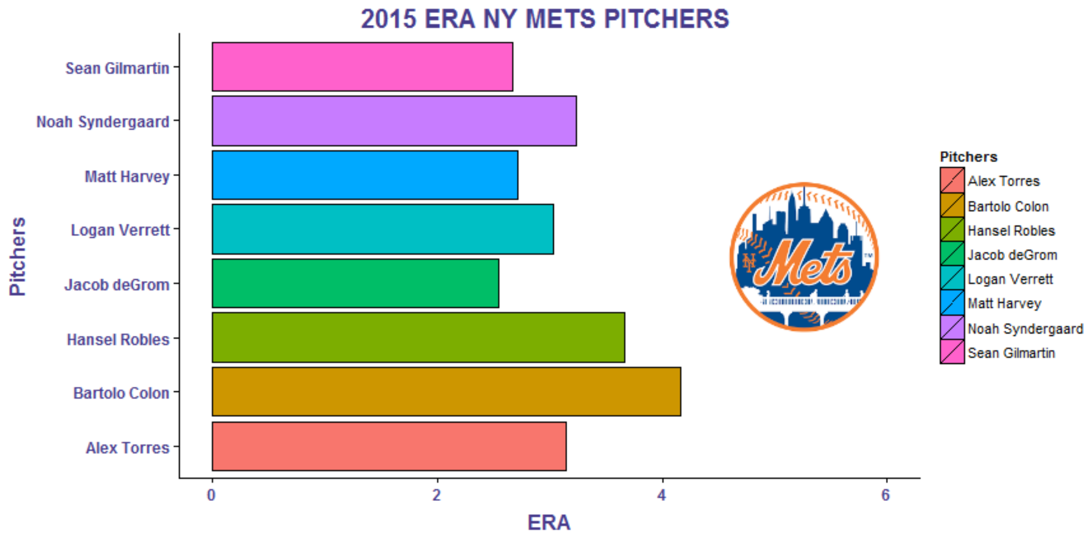


Table data for the above chart:

The above data in raw table format from R:

```
>  with(IUJake, table(des, pitch_type))
                           pitch_type
des                          CH CU FF SI SL
  Ball                        3 17  9 26 15
  Ball In Dirt                1  2  0  0  0
  Called Strike               1  4  6 15  7
  Foul                        1  3  4 11 12
  Foul (Runner Going)         0  0  0  0  1
  Hit By Pitch                0  0  0  0  1
  In play, no out             0  1  1  5  2
  In play, out(s)             3  5  0 10 10
  In play, run(s)             0  0  0  1  2
  Swinging Strike             1  6  2  5  7
  Swinging Strike (Blocked)   0  1  0  0  0
> with(IUClayton, table(pitch_type))
pitch_type
 CH  CU  FF  SL 
  1  41 127  56 
> with(IUClayton, table(des, pitch_type))
                           pitch_type
des                          CH CU FF SL
  Ball                        1 10 28 17
  Ball In Dirt                0  3  1  4
  Called Strike               0 12 28  5
  Foul                        0  3 29  6
  Foul Tip                    0  0  2  0
  Hit By Pitch                0  0  1  0
  In play, no out             0  1  8  3
  In play, out(s)             0  8 16  4
  In play, run(s)             0  0  1  0
  Swinging Strike             0  3 13 16
  Swinging Strike (Blocked)   0  1  0  1
> with(IUDavid, table(des, pitch_type))
                           pitch_type
des                          CH FC FF FT KC
  Ball                       16 10 19 16 12
  Ball In Dirt                1  0  0  0  0
  Called Strike               6  8 12 12  5
  Foul                        8  4 22  8  0
  Foul (Runner Going)        1  0  0  0  0
  In play, no out            4  2  2  1  2
  In play, out(s)           16  0 10  1  1
  In play, run(s)            0  0  2  1  0
  Swinging Strike           14  1  4  5  3
  Swinging Strike (Blocked)  2  0  0  0  0
> with(IUMax, table(des, pitch_type))
                           pitch_type
des                          CH CU FC FF SL
  Ball                       6  6  1 23 10
  Ball In Dirt               0  0  0  0  1
  Called Strike              1  4  0 14  8
  Foul                       4  2  0 34  6
  Foul Bunt                  0  0  0  2  0
  Foul Tip                   1  0  0  4  0
  Hit By Pitch               1  0  0  0  0
  In play, no out            1  0  0  3  2
  In play, out(s)            8  4  0  6  6
  In play, run(s)            1  1  0  3  2
  Swinging Strike            3  2  0 10  4
  Swinging Strike (Blocked)  2  0  0  0  0
> with(IUChris, table(des, pitch_type))
                 pitch_type
des                 CH FT SL
  Ball              11 11  8
  Called Strike      7  9  5
  Foul               4 12  3
  Foul Tip           1  0  0
  Hit By Pitch       0  0  1
  In play, no out    1  2  1
  In play, out(s)    6 12  1
  In play, run(s)    0  2  0
  Swinging Strike    6  5  0
```

Deliverable 3:



Future improvements:

1. R is good for statistical analysis, and at the same time to extract massive amount of data, it takes lot of memory for processing and hence it is dependent on the type of resource(machine environment) being used. Our project did not use H2O which is a Java Virtual Machine that is optimized for doing "in memory" processing of distributed, parallel machine learning algorithms on clusters. It could help in loading the data in parallel.
2. Due to time constraints, only charts were able to be produced. In future, this can be extended to publish using RPubs and make use of R Shiny package to do some interactive visualization.

Reproducibility

Documentation

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and Mac OS.

- R version: 3.1.0 and R Studio version: 0.98.507 or higher

- Github Source code Repository for the project can be accessed from: https://github.com/Gr8Dev/Sabermetrics-with-R

- Readme.md - NOTE: Change to the directory location you want to get Github files from. If you place github files in c://Sabermetrics-with-R, this script will work as-is.

  Using Git Bash, change the working directory to c:/
  cd c:/

  Git environment setup (using Gitbash)
  git init

  This will create a new directory called Sabermetrics-with-R
  git clone https://github.com/Gr8Dev/Sabermetrics-with-R

  cd Sabermetrics-with-R
  The following files will be seen in this directory:
  newyorkmets.png
  FinalProject.R
  PitchFX2015.xls
  FinalProject.R
  3 sample deliverables on running FinalProject.R

  Open RStudio
  Make sure you set using Tools -> Global options, Default working directory
  is set to C://Sabermetrics-with-R or any desired location that you placed the files.

  Open FinalProject.R in Console , press Ctrl-A and Run them as a batch or run individual line.

How to run the software:

Install R 3.1.0 – Download R for windows from https://cran.r-project.org/bin/windows/base/old/3.1.0/

Install R Studio - https://www.rstudio.com/products/rstudio/download/

Open R Studio and install the following R necessary packages needed for the project and load the library:

```
#Install these packages
install.packages("xlsx")
install.packages("plyr")
install.packages("cowplot")
install.packages("pitchRx")


#Load all the packages
library(pitchRx)
library(dplyr)
library(plyr)
library(RSQLite)
library(ggplot2)
library(grid)
library(gridExtra)
require(cowplot)
library(xlsx)
library(data.table)
```

Get the source code from Github (Copy and paste the code) in R studio

https://github.com/Gr8Dev/Sabermetrics-with-R

Select all the lines and click run. The game data for 2 months took approximately 5 minutes initially to gather and the other steps should finish quickly. So, there is an overall 10min required to run the program or it could be faster on powerful machines..

Dataset Used

The dataset used for this project is PITCHf/x.

PITCHf/x is a pitch tracking system, created by Sportvision, and is installed in every MLB stadium since around 2006. This system tracks the velocity, movement, release point, spin, and pitch location for every pitch thrown in baseball, allowing pitches and pitchers to be analyzed and compared at a detailed level. The PITCHf/x data classifies the pitches as CH(changeup), #CU(Curve ball), FA (Fast ball), FC(Clutter), FF(Four-seam fastball), PO(Pitchout),SL(Slider) etc. It uses the field des in PITCHf/x data to determine the type of pitches the pitchers had thrown, and the outcome of pitchers are gathered. The data set covers for July to Sep 2015 top 5 pitchers for analysis.

Bonus

The design of deliverables task and charting is original as we proposed.

Work cited

ESPN. ESPN Internet Ventures. Web. 27 Oct. 2015.

"Exploring Baseball Data with R." Exploring Baseball Data with R. Web. 27 Oct. 2015.

"FIP | FanGraphs Sabermetrics Library." FIP | FanGraphs Sabermetrics Library. Web. 27 Oct. 2015.

"FIP | FanGraphs Sabermetrics Library." FIP | FanGraphs Sabermetrics Library. Web. 27 Oct. 2015.

"Major League Leaderboards » 2015 » Batters » Dashboard | FanGraphs Baseball." Major League Leaderboards » 2015 » Batters » Dashboard | FanGraphs Baseball. Web. 27 Oct. 2015.

"2009 NL Playoff Preview: Pitching Rotation - Graph of the Day." Beyond the Box Score. 24 Sept. 2009. Web. 27 Oct. 2015

http://espn.go.com/mlb/team/stats/pitching/_/name/chw/chicago-white-sox
http://espn.go.com/mlb/team/stats/pitching/_/name/wsh/split/45/washington-nationals

http://www.fangraphs.com/library/pitching/fip/

http://blog.revolutionanalytics.com/2014/04/a-dive-into-h2o.html
https://fastballs.wordpress.com/2007/08/02/glossary-of-the-gameday-pitch-fields/
http://www.fangraphs.com/library/misc/pitch-fx/