R in an Hour

R Language and Ecosystem

What are we going to cover

- Highlights of R language, a bit of history and its characteristics
- Great uses to consider R language and eco-system
- Demo/Walkthrough of Real R projects
- We WILL NOT cover deep language syntax - sorry engineers, separate Talk ;)

- We don't have enough time in an hour to cover the R language core feature set.
- The objective of this talk is to showcase the capabilities and use of the R ecosystem

R Highlights and Characteristics…

R - What is it? Where did it come from?

- Historically a DSL for Statistic analysis and reporting
- Based from S language
    - S in 1970s
    - R dialect of S in late 1990's
- Implemented in C and FORTAN
- Heavy adoption in Data Science in recent years
- Heavy community support to extend R core with packages - CRAN
- Competition from Python

R - Language Characteristics

- Dynamic Typing
- Interpreted
- Functional, with some OO littered in
- Declarative vs Imperative – emphasis on set operations, versus scalar
- Potentially large learning curve for developers with OO or Imperative Language Backgrounds

R - Foundational Basics

- Vectors

```
someVector <- c(1,2,3)
```

- Matrix

```
someMatrix <- c(1,2,3,4,5,6, nrow=2,ncol=3)
```

- List

```
someList <- list(someMatrix,someVector)
```

Vectors, Matrices & Lists are the foundational building blocks in R. Scalars are available, but take a back seat in declarative, set based language like R.

R - Data Frame (Table)

- In R we can create tables (similar to tables in RDBMS)
- Create a Data Frame (Table)

```
library(data.table)


set.seed(100)


dt <- data.table(id = c(1:100),
                 value = c(sample(1:1000,100)))
```

The Data Frame (or Data Table) closely resembled the table in Relational Databases.

The Data Table is an extended CRAN package implemented in C for raw performance, and adds convenience methods on top of base R.

R - Accessing element from data table

- To access an element, we can use “subsetting”

- By Row:

```
dt[2]
```

```
##    id value
## 1:  2   258
```

- Columns by name:

```
dt$value[2]
```

```
## [1] 258
```

R - Filtering data tables

- Let's filter table rows with ID of two via "subsetting"
- Subsetting is R core feature to select

```
dt[dt$id == 2]
```

```
##    id value
## 1:  2   258
```

- Or, we can use the dplyr (or plyr) filter method…

```
library(dplyr)
filter(dt,id == 2)
```

```
##    id value
## 1:  2   258
```

R - Chaining

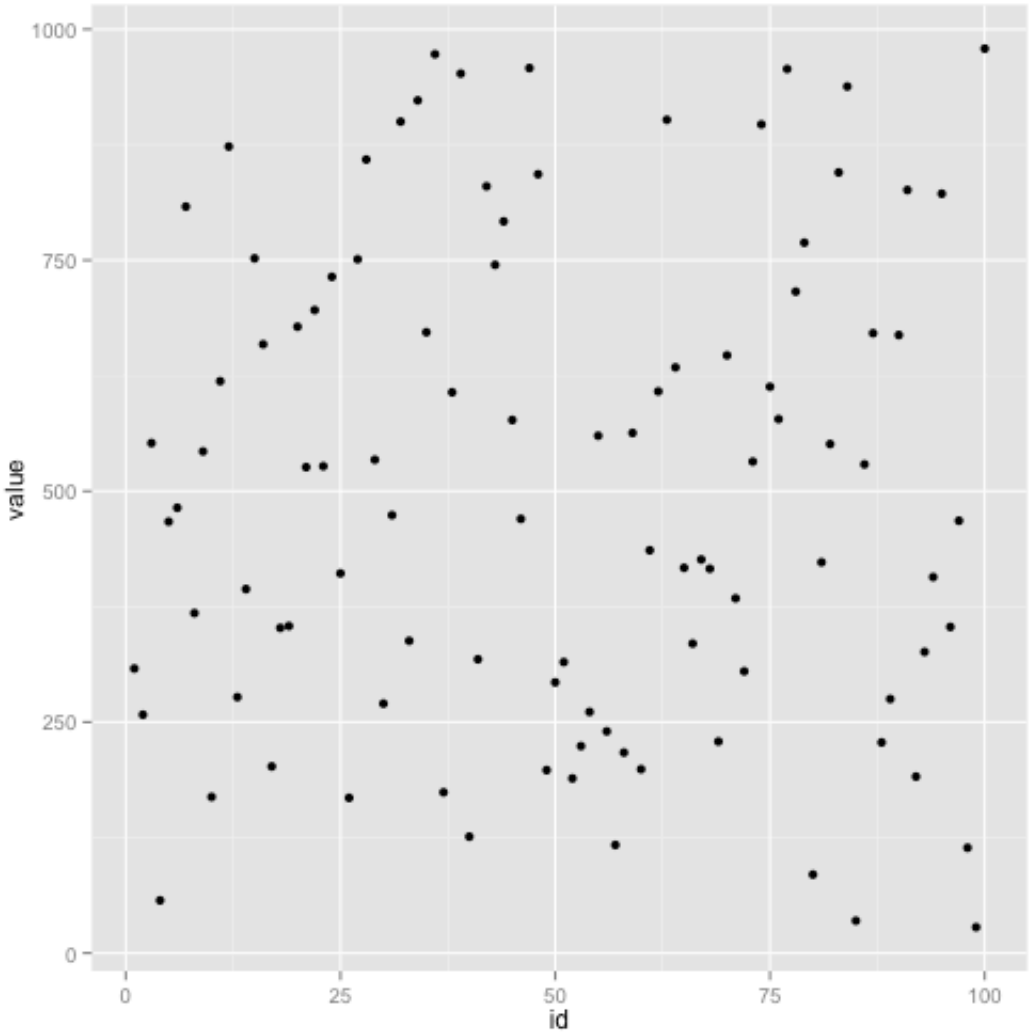
- Chain several aggregrate operations

```
filter(dt, id > 10) %>%
        summarize(count = n(),
                  mean = mean(value),
                  median = quantile(value,0.50),
                  ninetyPercent = quantile(value,0.90),
                  total = sum(value))
```

```
##    count     mean median ninetyPercent total
## 1:    90 513.8444  526.5         897.3 46246
```

R - Plotting

```
library(ggplot2)
p <- qplot(x=id, y=value, data=dt)
print(p)
```



plot of chunk example.plot

R - Key Language Takeaway | Avoid Looping Constructs!

- R Language Performance - traditional procedural looping constructs not as efficient as set based constructs
- Much effort by community to build performant set based packages, such as `dplyr` and `data.table`

R Uses and Ecosystem…

What is it good for?

- Statistics/Numerical analysis and Reporting
- Exploratory Analysis
- Lightweight Reporting
- Lightweight Presentations (like this one)

What is is NOT good for?

- Systems Programming
- Complex Reporting

Ecosystem

- Open source Development Community - R Core, CRAN
- Commercial Contributors - RStudio, Revolution Analytics
- Affliation with Open Source Agile tools - GitHub, RPubs, Markdown

RStudio

- IDE
- Interactive Reporting Server

Markdown | Simplistic plain text Markup language

- No tags (like in JSON, HTML, XML)
- Awesome for short documents, wikis – Github supports natively

```
# Level 1 Heading

## Level 2 Heading

### Level 3 Heading

- Unordered List Item (bullet)

1. Order List Item (bullet)
```

```
[Some Link Title](http://somehypertextlink.com)
```

Markdown | Rendering

Level 3 Heading

- Unordered List Item (bullet)
- Order List Item (bullet)

Some Link Title

RMarkdown & Literal Programming

- **Literal Programming**: Documenting logical flow in plain English, embedding code supporting the flow
- RStudio supports Literal Programming
- RMarkdown is implementation of Markdown, allowing:
  - Embed of dynamic R code
  - Output of different formats, such as HTML, PDF, etc
- This presentation was implemented with RMarkdown, using Google ioslide output format.
- RStudio supports interactive reporting via RMarkdown and Shiny framework

RPubs

- Community Web Server for Publishing results
- Single click publish from RStudio for RMarkdown Content
- Free Edition is restrictive for full (public access)

Walkthrough of Reproducible Research Project

Walkthrough of HAR Extract Report

Bye!

References

- R Language: http://www.r-project.org/
- RStudio: http://www.rstudio.com/
- CRAN: http://cran.r-project.org/
- Markdown: http://daringfireball.net/projects/markdown/
- RMarkdown: http://rmarkdown.rstudio.com/
- ioslides: http://rmarkdown.rstudio.com/ioslides_presentation_format.html#presenter-mode

- RPubs: http://rpubs.com/
- Revolution Analytics: http://www.revolutionanalytics.com/