DFIR Redefined:

Deeper **F**unctionality for **I**nvestigators with **R**

SecureWorld Seattle 2017

Eric Kapfhammer
Russ McRee

HolisticInfoSec™

"To competently perform rectifying security service, two critical incident response elements are necessary:
information & **organization**."
~ Robert E. Davis

DFIR as we know it…

Digital **F**orensics &

Incident **R**esponse

- 38% reported an **increase in the number of hours devoted to incident response**
- 42% reported an **increase in the volume of incident response data collected**
- 39% indicated an **increase in the volume of security alerts**

"It's just not mathematically possible for companies to hire a large enough staff to investigate tens of thousands of alerts per month, nor would it make sense."
~Nathan Burke

From Secure360.org The Challenges of Incident Response, Nov 2016

The Show Must Go On!

2017 SANS Incident Response Survey (Matt Bromiley, June 2017)

- Survey shows that IR teams are:
  - Detecting the attackers faster than before, with a drastic improvement in dwell time
  - Containing incidents more rapidly
  - **Relying more on in-house detection and remediation mechanisms**

DFIR redefined…

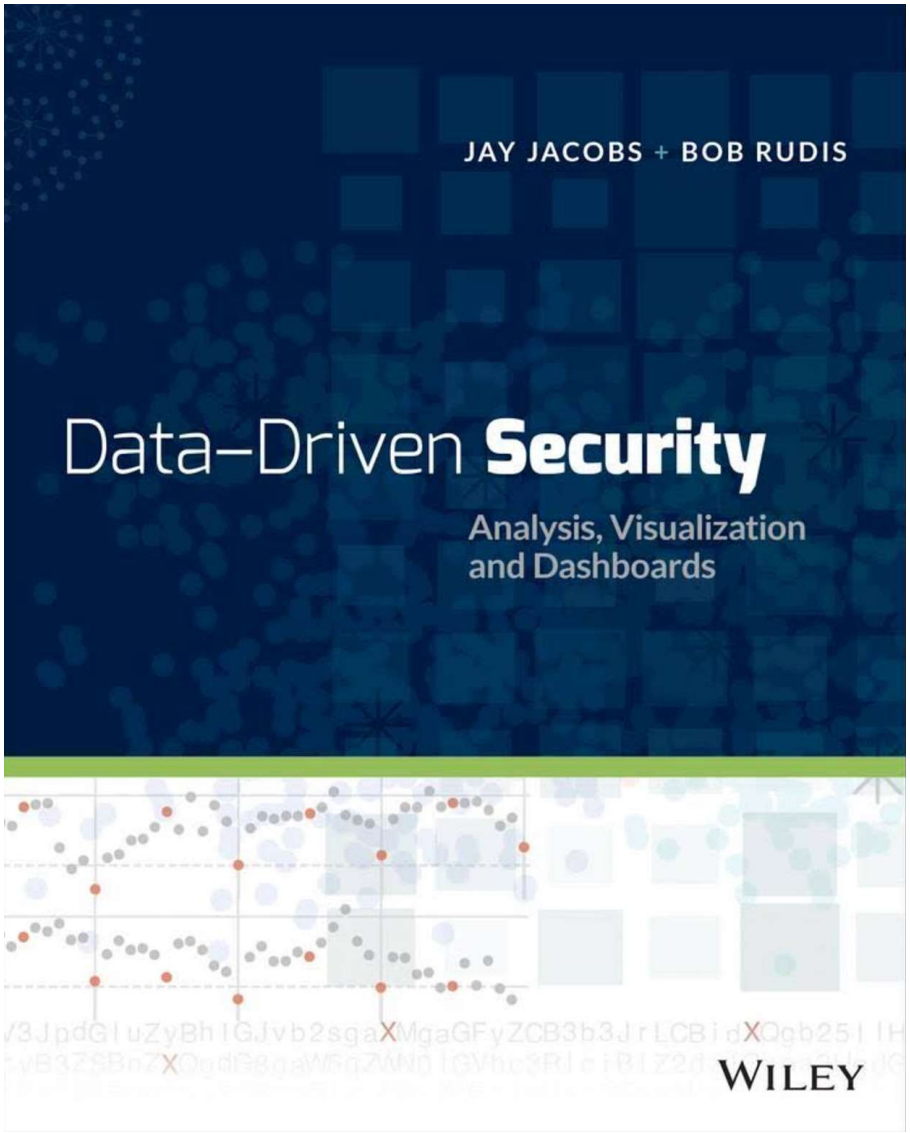
Deeper **F**unctionality for **I**nvestigators with **R**

- Incident responders and investigators **need all the help they can get**
- What concepts & methods further enable handlers & investigators as they continue to strive for **faster detection** and containment?
- **Data science** & **visualization** sure can't hurt
- How can we be more **creative** to achieve "deeper functionality"?

DFIR redefined…

Deeper **F**unctionality for **I**nvestigators with **R**

- "In God we trust. **All others must bring data.**" ~William E. Deming



What is R, besides a pirate's favorite programming language?

R

- "**100% focused & built for statistical data analysis & visualization**"
- "Makes it remarkably simple to run extensive statistical analysis on your data & then generate informative & appealing visualizations with just a few lines of code"
- Interface with data via file ingestion, database connection, APIs
- Benefit from a wide range of packages & strong community investment

Sympathy For The Part-time R User

- "Not all R users consider themselves to be expert programmers (many are happy calling themselves **analysts**)."
- "R is often used in collaborative projects where there are varying levels of programming expertise."
- I propose that this represents the vast majority of us in this room
  - Not expert programmers, data scientists, or statisticians
  - Analysts re-using code for our own purposes, red or blue

DFIR Redefined Scenarios

- Have you been pwned?
- Visualization for malicious Windows Event Id sequences
- How do your potential attackers *feel*, or can you identify an attacker via sentiment analysis?
- Fast Frugal Trees (decision trees) for prioritizing criticality
- Time Series Regression for user logons at volume

Have you been pwned?

Scenario: Assess organizational exposure as the result of data breaches (Any Equifax customers in the house?)

- Troy Hunt's *';--have i been been pwned* is a gold mine
  - https://haveibeenpwned.com
- API allows the list of pwned accounts (email addresses and usernames) to be quickly searched via a RESTful service
  - v2 of API returns a significant amount of additional data over v1
  - **Breaches**
    - all breaches for an account, all breached sites in the system, a single breached site
  - **Pastes**
    - all pastes for an account
  - **Pwned Passwords**

Have I Been Pwned?

Bindings for the 'HaveIBeenPwned.com' Data Breach API

Steph Locke's HIBPwned package for Troy Hunt's Have I Been Pwned

Documentation for package 'HIBPwned' version 0.1.6

- DESCRIPTION file.

Help Pages

HIBPwned-package	HIBPwned
account_breaches	Search for data breaches associated with one or more accounts.
breached_site	Get a specific breached site, based in breach name (not domain)
breached_sites	Get all (nonsensitive) breached sites in HIBP
data_classes	Get list of classes of data that have been exposed in breaches
GETcontent	Used to extract content from a GET request and collapse to a data.frame
HIBPwned	HIBPwned
HIBP_headers	Construct headers for a HIBP request. [Optional] Change the agent to make the request bespoke to you.
pastes	Search for cases where an email address has been included in a paste

Has Russ Been Pwned?

HIBPwned

In [8]:
```
library("HIBPwned")

# Has Russ been pwned?
account_breaches(c("rmcree@yahoo.com","holisticinfosec@gmail.com","russ@holisticinfosec.org"))
breached_site("LinkedIn")
pastes("russ@holisticinfosec.org")
```

$`rmcree@yahoo.com`

Title	Name	Domain	BreachDate	AddedDate	ModifiedDate	PwnCount	Description	DataClas
Adobe	Adobe	adobe.com	2013-10-04	2013-12-04T00:00:00Z	2013-12-04T00:00:00Z	152445165	In October 2013, 153 million Adobe accounts were breached with each containing an internal ID, username, email, encrypted password and a password hint in plain text. The password cryptography was poorly done and many were quickly resolved back to plain text. The unencrypted hints also disclosed much about the passwords adding further to the risk that hundreds of millions of Adobe customers already faced.	Em address Passwo hint Passwords , Usernam
Last.fm	Lastfm	last.fm	2012-03-22	2016-09-20T20:00:49Z	2016-09-20T20:00:49Z	37217682	In March 2012, the music website Last.fm was hacked and 43 million user accounts were exposed. Whilst <a href="http://www.last.fm/passwordsecurity" target="_blank" rel="noopener">Last.fm knew of an incident back in	Em addresse Passwords , Usernam

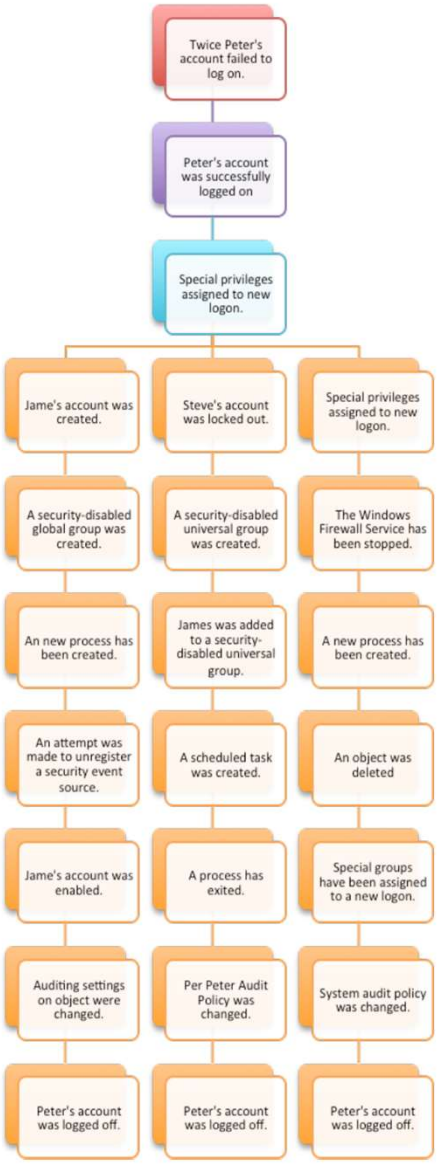
Malicious Event Sequence Visualization

Scenario: Malicious event or process sequences

- Sequenced visualization opportunities
  - Windows Events by Event ID
  - Hypothetical scenario for this visualization:
    - Multiple failed logon attempts (4625) followed by successful logon (4624), then various malicious sequences
  - Fantastic reference paper:
    - Intrusion Detection Using Indicators of Compromise Based on Best Practices and Windows Event Logs
- Additional opportunities
  - Processes by parent/child

HolisticInfoSec™

Malicious Event Sequence Visualization



Malicious Event Sequence Visualization

- Sequenced data after parsed and counted

	A	B
1	4625-4625-4624-4672-4720-4728-4668-4905-4722-4907-4634	45
2	4625-4625-4624-4672-4740-4759-4761-4698-4689-4912-4634	23
3	4625-4625-4624-4672-4672-5025-4688-4660-4694-4719-4634	15
4	4625-4625-4624-4964-4767-4760-4758-4757-4753-4750-4743-4740	34
5	4625-4625-4624-5025-5034-4950-4949	220
6	4625-4625-4624-4649-4912-4618	108
7	4625-4625-4624-4907-4660-4670-4691	427
8	4625-4625-4624-4672-4720-4728-4668-4905-4722-4907-4634	19
9	4625-4625-4624-4672-4740-4759-4761-4698-4689-4912-4634	62
10	4625-4625-4624-4672-4672-5025-4688-4660-4694-4719-4634	78
11	4625-4625-4624-106	2430
12	4625-4625-4624-2004	431
13	4625-4625-4624-2005	235
14	4625-4625-4624-2006	116

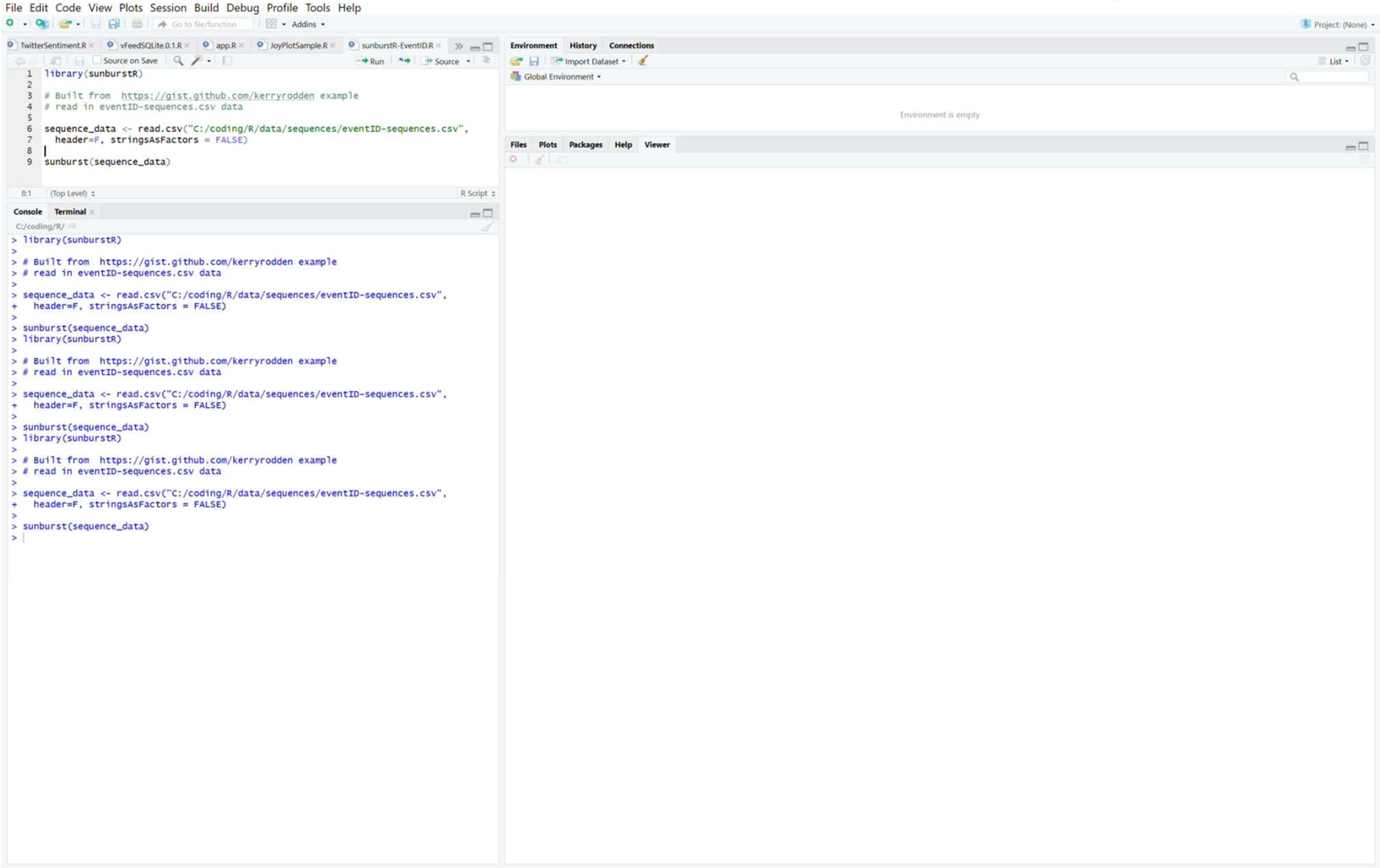
- Run through three line of R code

```
library(sunburstR)

# Built from  https://gist.github.com/kerryrodden example
# read in eventID-sequences.csv data

sequence_data <- read.csv("C:/coding/R/data/sequences/eventID-sequences.csv",
  header=F, stringsAsFactors = FALSE)

sunburst(sequence_data)
```


HolisticInfoSec™

Malicious Event Sequence Visualization

Sunburst visualization can be included in dashboards via Shiny or PowerBI



Assessing emotional valence (sentiment)

- Do certain adversaries or adversarial communities use social media?
  - **Yes**
- As such, can social media serve as an early warning system, if not an actual sensor?
  - **Yes**
- Are certain adversaries, at times, so unaware of OpSec on social media that you can actually locate them or correlate against other geo data?
  - **Yes**

Assessing emotional valence (sentiment)

- **twitteR** & **rtweet** (Jeff Gentry) : Interface with Twitter API
  - twitteR: provides access to the Twitter API. Most functionality of the API is supported, with a bias towards API calls that are more useful in data analysis as opposed to daily interaction.
  - Rtweet: R client for interacting with Twitter's REST & stream API's.
  - Code & concepts here are drawn directly from Michael Levy, PhD UC Davis: Playing With Twitter

Assessing emotional valence (sentiment)

Scenario: DDoS attacks from hacktivist or chaos groups

- Attacker groups often use associated hashtags and handles
  - The minions that want to be *part of* often retweet & use the hashtag
- Individual attackers either freely give themselves away, or often become easily identifiable or associated, via Twitter
- A walkthrough of analysis techniques that may help identify or better understand the motives of certain adversaries & adversary groups
  - Won't use actual adversary handles, used a DDoS news cycle and journalist/bloggers as exemplars

Assessing emotional valence (sentiment)

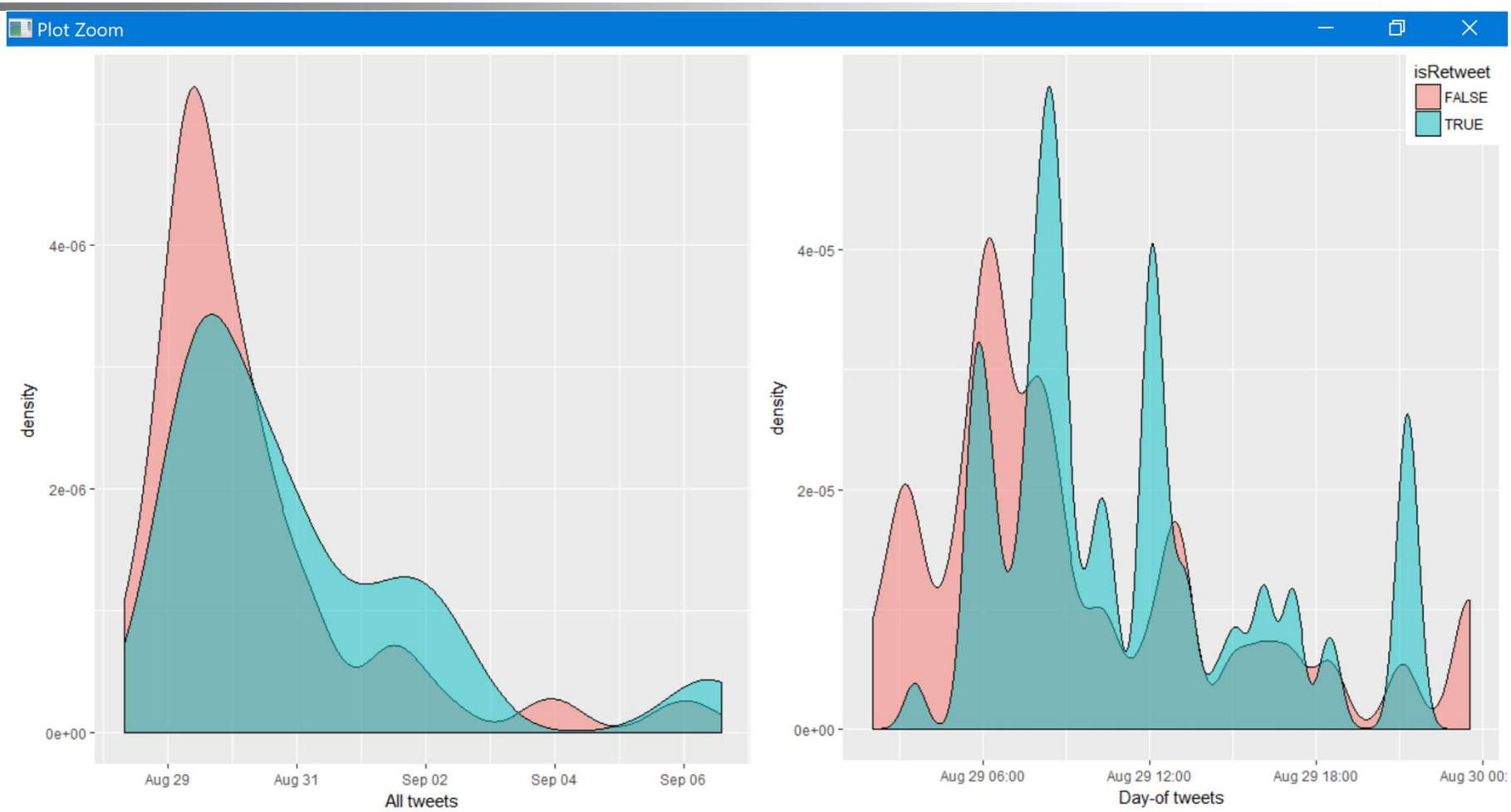
Example: Following the **WireX botnet**, comprised mainly of Android mobile devices utilized to launch a high-impact DDoS extortion campaign against multiple organizations in the travel and hospitality sector, August 2017

Started with three related hashtags:

#DDOS #Android #WireX

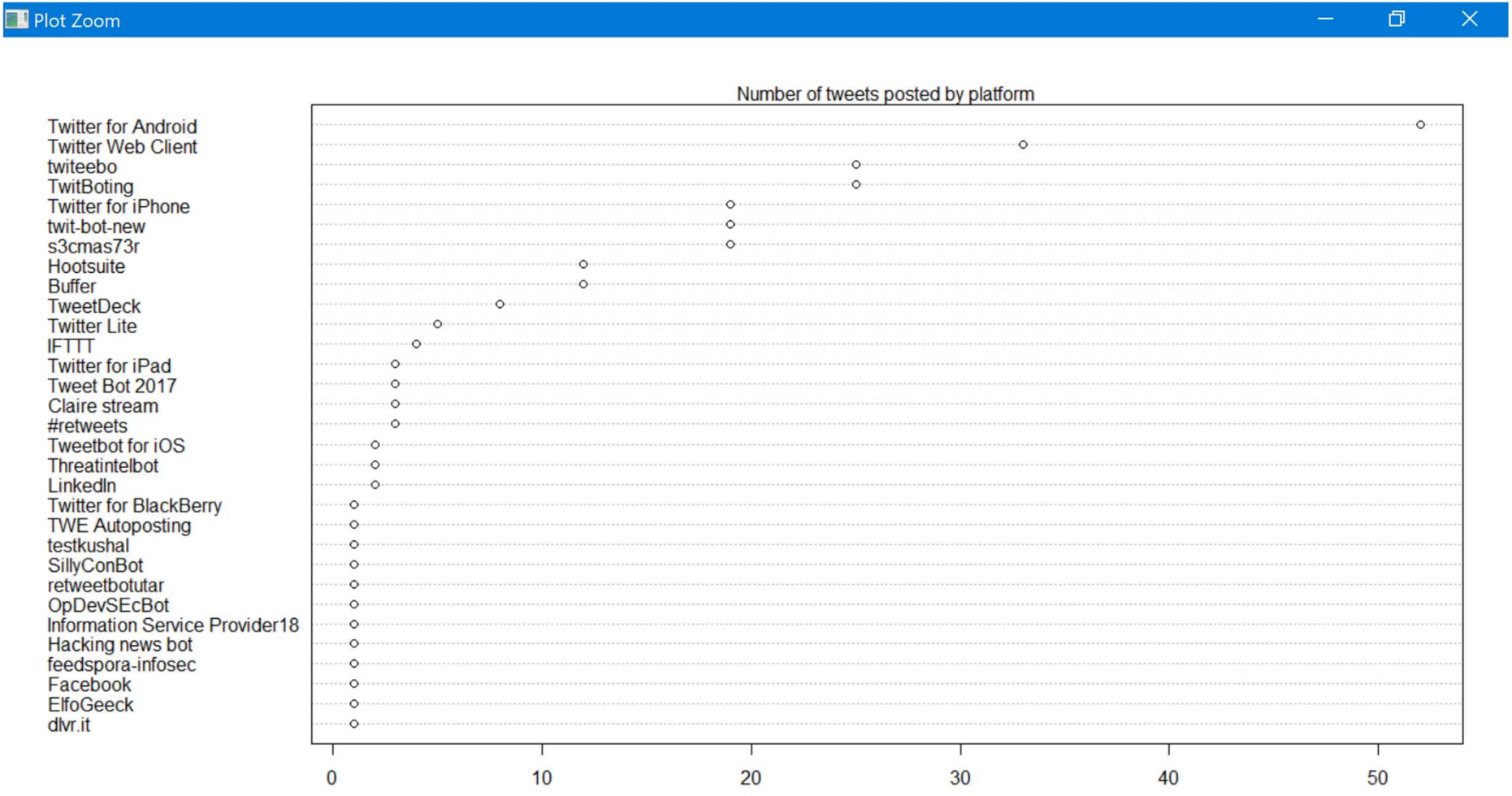
HolisticInfoSec™

Tweets by Day & by Time of Day




```
tw = searchTwitter('#DDOS AND #Android AND #WireX', n = 1000, since = '2017-08-27')
d = twListToDF(tw)
write.csv(d, 'C:/coding/R/data/TwitterSentiment/DDOS.csv')
# d = read.csv('C:/coding/R/data/TwitterSentiment/DDOS.csv')

dlcreated = with_tz(dlcreated, 'America/Los_Angeles')

timeDist = ggplot(d, aes(created)) +
  geom_density(aes(fill = isRetweet), alpha = .5) +
  scale_fill_discrete(guide = 'none') +
  xlab('All tweets')

# Zoom in on day of month
dayOf = filter(d, mday(created) == 29)
timeDistDayOf = ggplot(dayOf, aes(created)) +
  geom_density(aes(fill = isRetweet), adjust = .25, alpha = .5) +
  theme(legend.justification = c(1, 1), legend.position = c(1, 1)) +
  xlab('Day-of tweets')
cowplot::plot_grid(timeDist, timeDistDayOf)
```

Tweets Posted By Platform




```
#Number of tweets posted by platform

par(mar = c(3, 3, 3, 2))
d$statusSource = substr(d$statusSource,
                        regexpr('>', d$statusSource) + 1,
                        regexpr('</a>', d$statusSource) - 1)
dotchart(sort(table(d$statusSource)))
mtext('Number of tweets posted by platform')
```

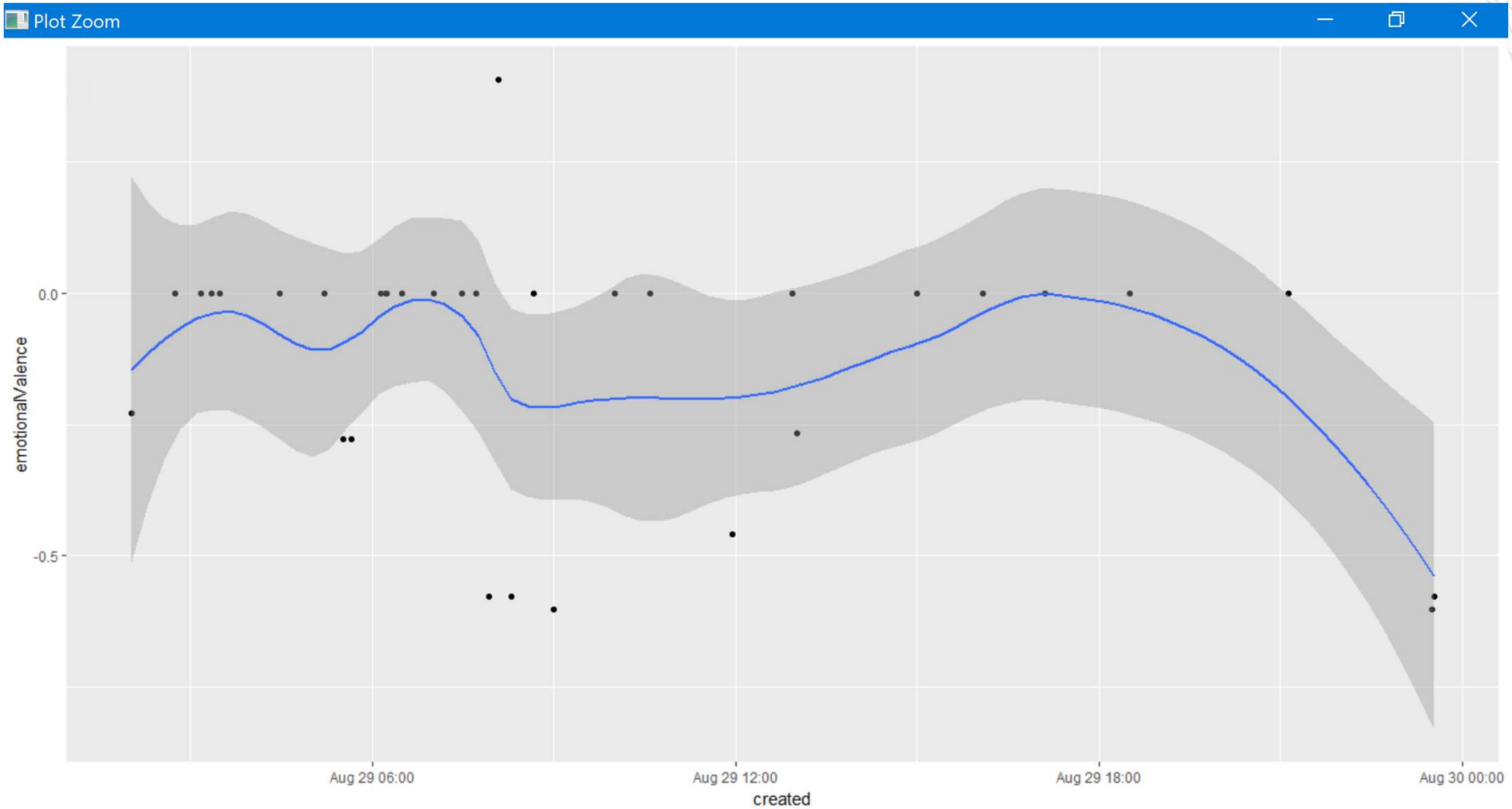
Emotional Valence – Most & Least Positive Tweets

Console Terminal

C:/coding/R/

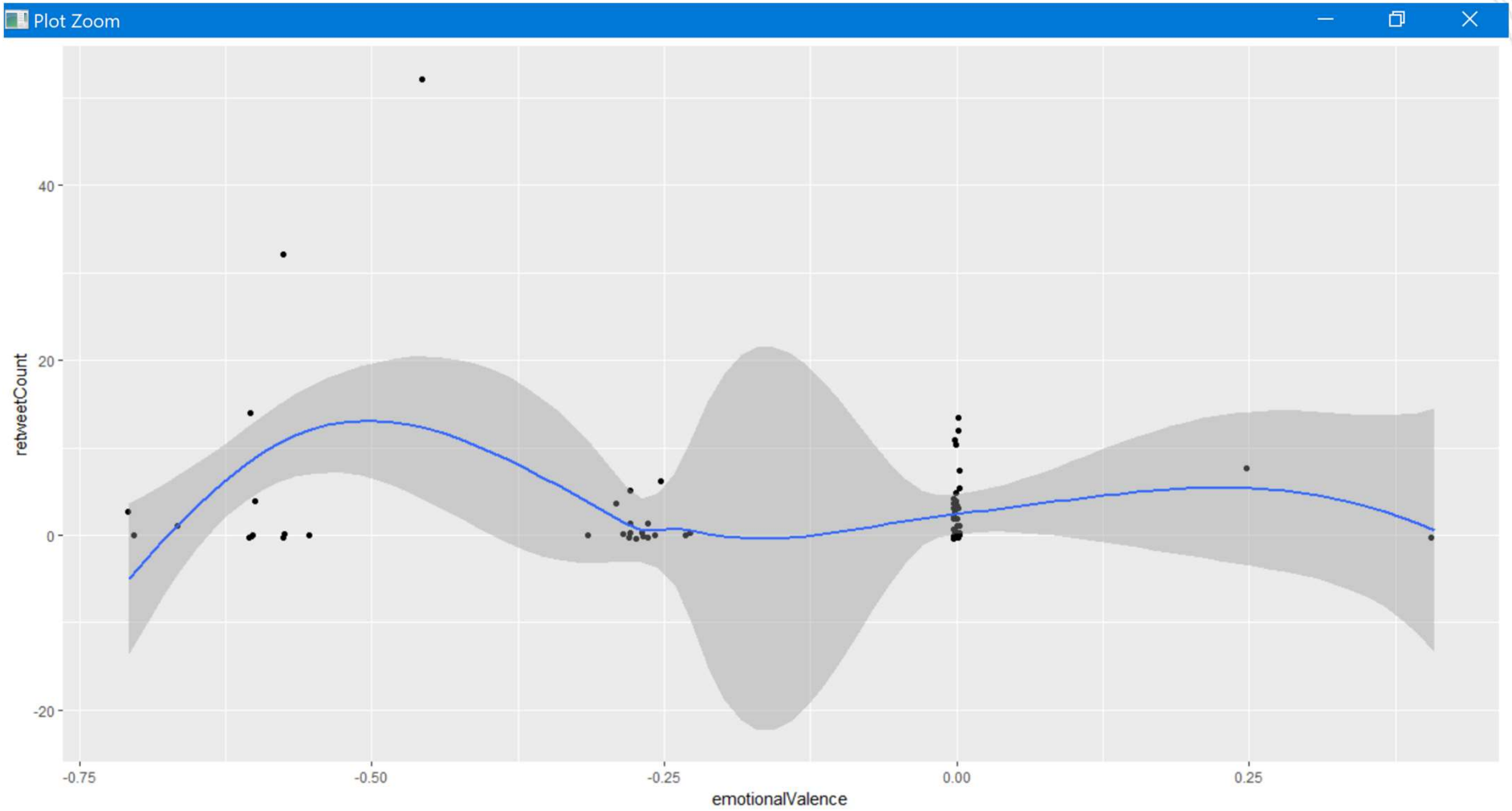
```
> # As reality check, what are the most and least positive tweets
> orig$text[which.max(orig$emotionalValence)]
[1] "A bunch of Internet tech companies had to work together to clean up #WireX #Android #DDoS #botnet\n
https://t.co/KyR5NLU72R\n#iat"
> orig$text[which.min(orig$emotionalValence)]
[1] "Dangerous #WireX #Android #DDoS #Botnet Killed by #SecurityGiants https://t.co/2GckGJQXKK"
>
```

How does emotional valence change over the day?




```
# How does emotionalValence change over the day?
filter(orig, mday(created) == 29) %>%
  ggplot(aes(created, emotionalValence)) +
  geom_point() +
  geom_smooth(span = .5)
```

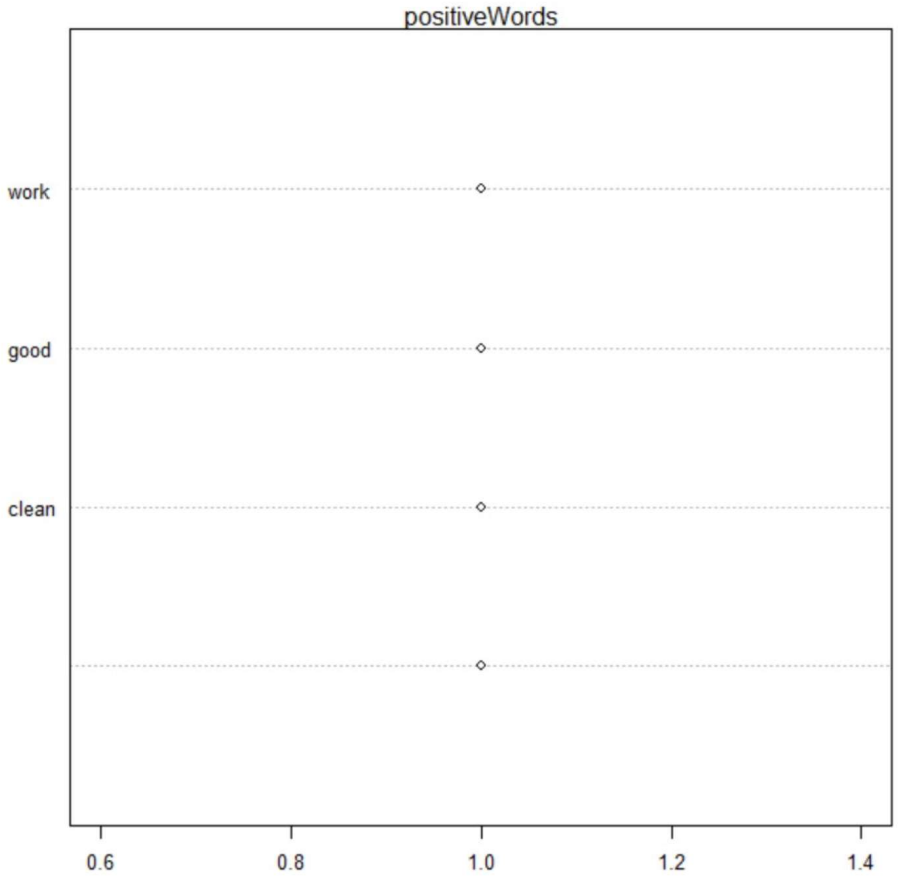
Do more positive (happier) tweets get retweeted more?

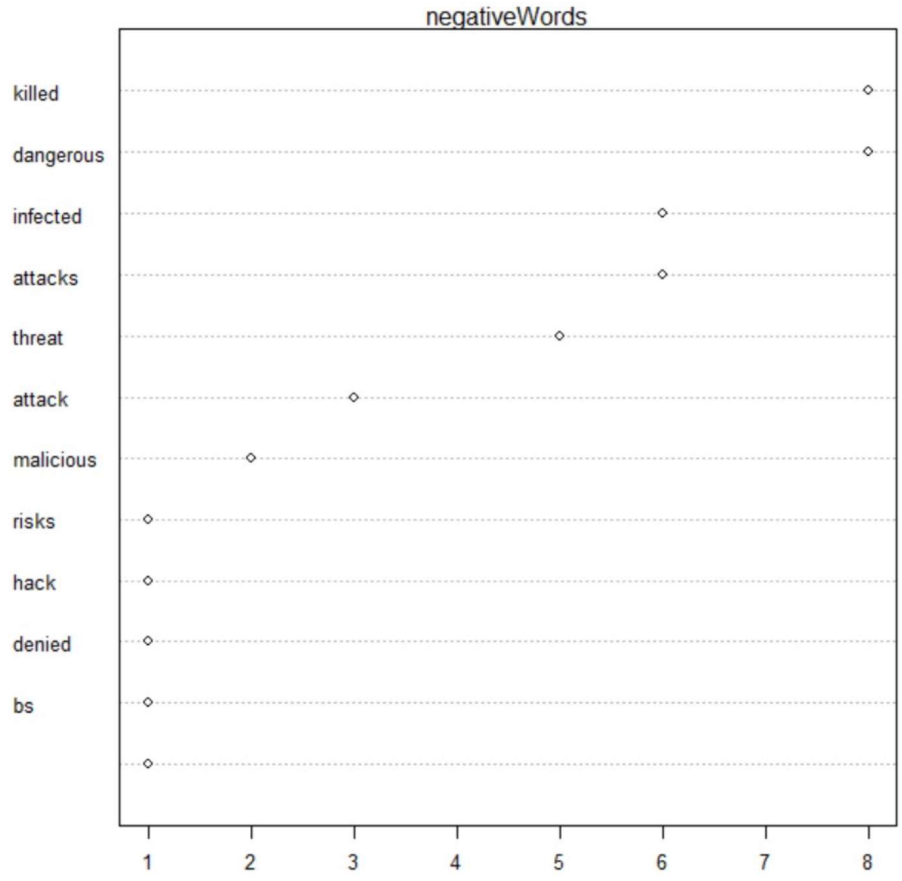



```
# Do happier tweets get retweeted more?
ggplot(orig, aes(x = emotionalValence, y = retweetCount)) +
  geom_point(position = 'jitter') +
  geom_smooth()
```

HolisticInfoSec™

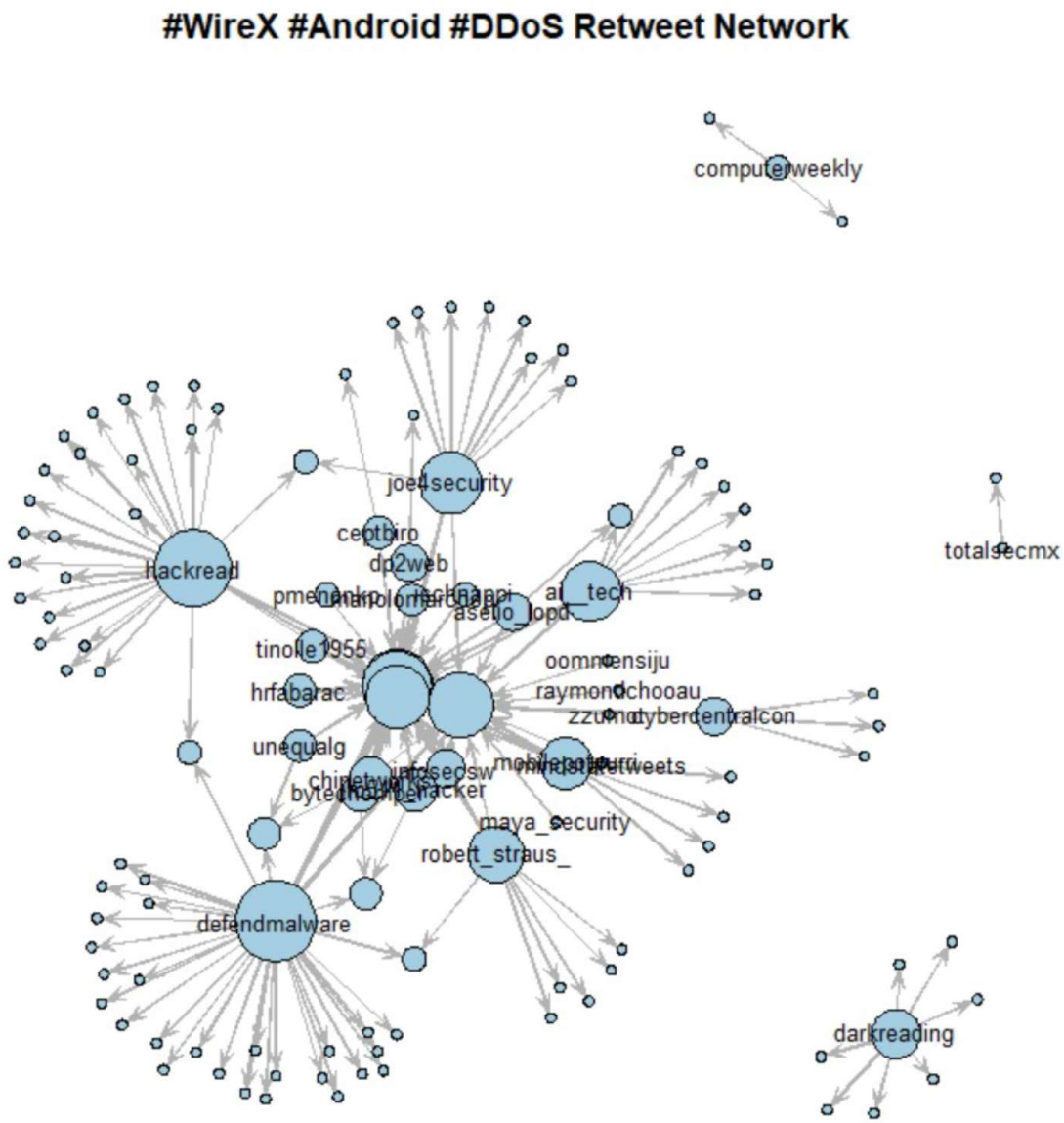
Positive and Negative Words





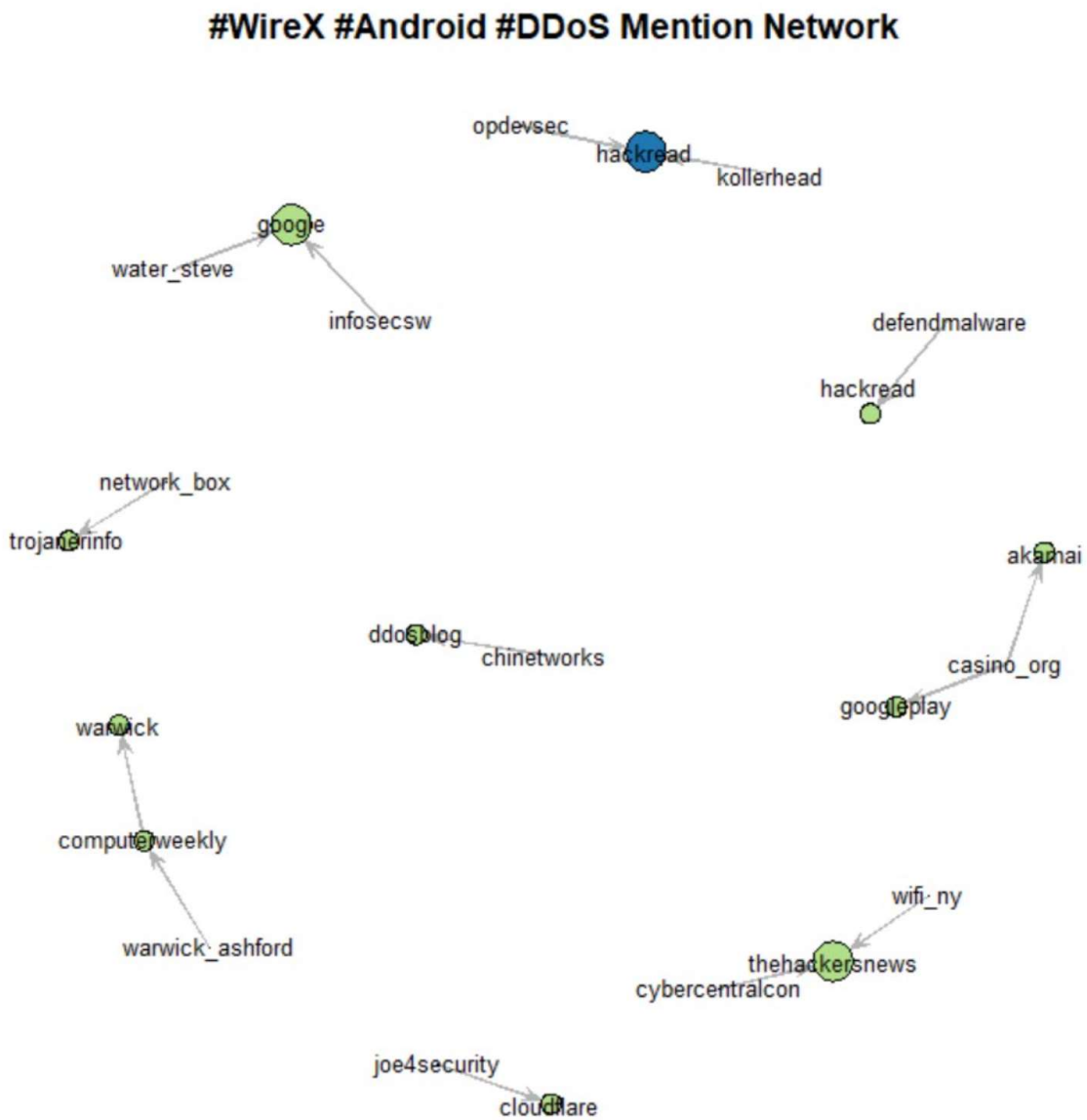

```
# Emotional Content

polWordTables =
  sapply(pol, function(p) {
    words = c(positivewords = paste(p[[1]]$pos.words[[1]], collapse = ' '),
              negativewords = paste(p[[1]]$neg.words[[1]], collapse = ' '))
    gsub('-', '', words)  # Get rid of nothing found's "-"
  }) %>%
  apply(1, paste, collapse = ' ') %>%
  stripWhitespace() %>%
  strsplit(' ') %>%
  sapply(table)

par(mfrow = c(1, 2))
invisible(
  lapply(1:2, function(i) {
    dotchart(sort(polWordTables[[i]]), cex = .8)
    mtext(names(polWordTables)[i])
  }))
```

Who is retweeting who?




```
# Who's retweeting whom?
el = as.data.frame(cbind(sender = tolower(rt$sender),
                         receiver = tolower(rt$screenName)))
el = count(el, sender, receiver)
rtnet = network(el, matrix.type = 'edgelist', directed = TRUE,
                ignore.eval = FALSE, names.eval = 'num')

# Get names of only those who were retweeted to keep labeling reasonable
vlabs = rtnet %v% 'vertex.names'
vlabs[degree(rtnet, cmode = 'outdegree') == 0] = NA

par(mar = c(0, 0, 3, 0))
plot(rtnet, label = vlabs, label.pos = 5, label.cex = .8,
     vertex.cex = log(degree(rtnet)) + .5, vertex.col = col3[1],
     edge.lwd = 'num', edge.col = 'gray70', main = '#WireX #Android #DDoS Retweet Network')
```

Who is mentioning who?




```
# Make an edge from the tweeter to the mentioned, for each mention
mentionEL =
  lapply(seq_along(origtext), function(i) {
    # If the tweet didn't have a mention, don't make edges
    if(mentioned[[i]] == '')
      return(NULL)
    # Otherwise, loop over each person mentioned, make an edge, and rbind them
    lapply(mentioned[[i]], function(m)
      c(sender = origscreenName[i], receiver = m)) %>%
      do.call(rbind, .) %>% as.data.frame()
  }) %>%
  do.call(rbind, .) %>%
  count(tolower(sender), tolower(receiver))
# Make the network
mentionNet = network(mentionEL, matrix.type = 'edgelist', directed = TRUE,
                     ignore.eval = FALSE, names.eval = 'num')
# Color mention entities
vCol = rep(col3[3], network.size(mentionNet))
tweeters = c('joe4security', 'HackRead', 'defendmalware', 'AI__TECH',
             'Robert_Straus_', 'ChiNetworks', 'MobilePotpurri', 'dp2web',
             'Unequal9', 'ComputerWeekly', 'infosecsw')
vCol[(mentionNet %v% 'vertex.names') %in% tweeters] = col3[1]
vCol[mentionNet %v% 'vertex.names' == 'hackread'] = col3[2]
plot(mentionNet, displaylabels = TRUE, label.pos = 5, label.cex = .8,
     vertex.cex = degree(mentionNet, cmode = 'indegree'), vertex.col = vCol,
     edge.lwd = 'num', edge.col = 'gray70', main = '#wirex #Android #DDoS Mention Network')
```

Geolocation

```
#Lat Long for Milan, IT (@HackRead HQ)
tweets <- searchTwitter("@joe4security OR @HackRead OR @defendmalware OR @AI__TECH OR @Robert_S
                        n=100, lang="en", geocode='45.4642,9.1900,5mi', since="2017-08-27")
tweets_geolocated.df <- twListToDF(tweets)
write.csv(tweets_geolocated.df, "tweets_geolocated.csv")
```

text	created	screenName
Ocado launches voice-driven shopping app for Amazon Alexa https://t.co/YLOi2uQayb via @computerweekly	01-09-17 16:17	AlePanni
Two-thirds of UK firms hiring for #GDPR by @Warwick_Ashford on @computerweekly https://t.co/rlfI6SGDey https://t.co/k7YLKzEsFp	01-09-17 13:43	BandWinBW
Gartner hype cycle - #AI and Digital Platforms rank among the top 'megatrends' https://t.co/nZY1ULtzks Analyst news via @ComputerWeekly	31-08-17 05:03	mpasculli
Danske Bank taps into the cognitive #capabilities of IBM Watson https://t.co/ymxVLi7ifK via @computerweekly	30-08-17 10:59	dimperios

Alessio Panni
@AlePanni

Director of Business Development & Solutions - cloud, digital & innovation - @ Engineering D.HUB - tweets are my own - Love cycling, music & feeling free!

Milan

Joined August 2012

Edoardo Facchini
@BandWinBW

Dad, no profit volunteer, I train customers. I help companies to understand their customers needs, satisfy them and to present their solutions.

Milan, Lombardy

bandwin.net

Joined December 2014

Marco Pasculli
@mpasculli

Milan, Lombardy

Joined August 2010

Gianluca D'Imperio
@dimperios

#Compliance specialist, founding Member of International #RegTech Association. Passionate about #Fintech, #PayTech #blockchain #AI. Tweets are my own

Milano

Making fast, good decisions with FFTrees

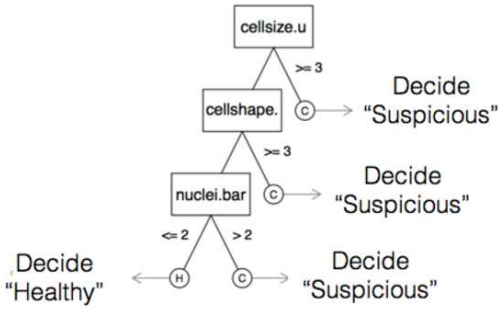
- Fast-and-frugal trees (FFTs) are simple algorithms that facilitate efficient & accurate decisions based on limited information.
- Nathaniel D. Phillips, PhD created FFTrees to allow anyone to easily create, visualize & evaluate FFTs

"…[W]e are suspicious of rapid cognition. We live in a world that assumes that the quality of a decision is directly related to the time and effort that went into making it." ~ Malcolm Gladwell

Logistic Regression vs **Fast and Frugal Trees (FFTrees)**

$$p(Suspicious) = \frac{1}{1 + e^{-(-10.1 + 0.54 \times x_1 - 0.01 \times x_2 + \ldots)}}$$

$x_1 = thickness$
$x_2 = cellsize.unif$
$x_3 = cellshape.unif$
$x_4 = adhesion$
$x_5 = epithelial$
$x_6 = nuclei.bare$
$x_7 = chromatin$
…



Making fast, good decisions with FFTrees

Scenario: After a breach, and subsequent risk assessment that generated a ton of CVSS data, make a fast decision about what treatments to apply first

- Because everyone *loves* CVSS

Making fast, good decisions with FFTrees

The data looks less than compelling…

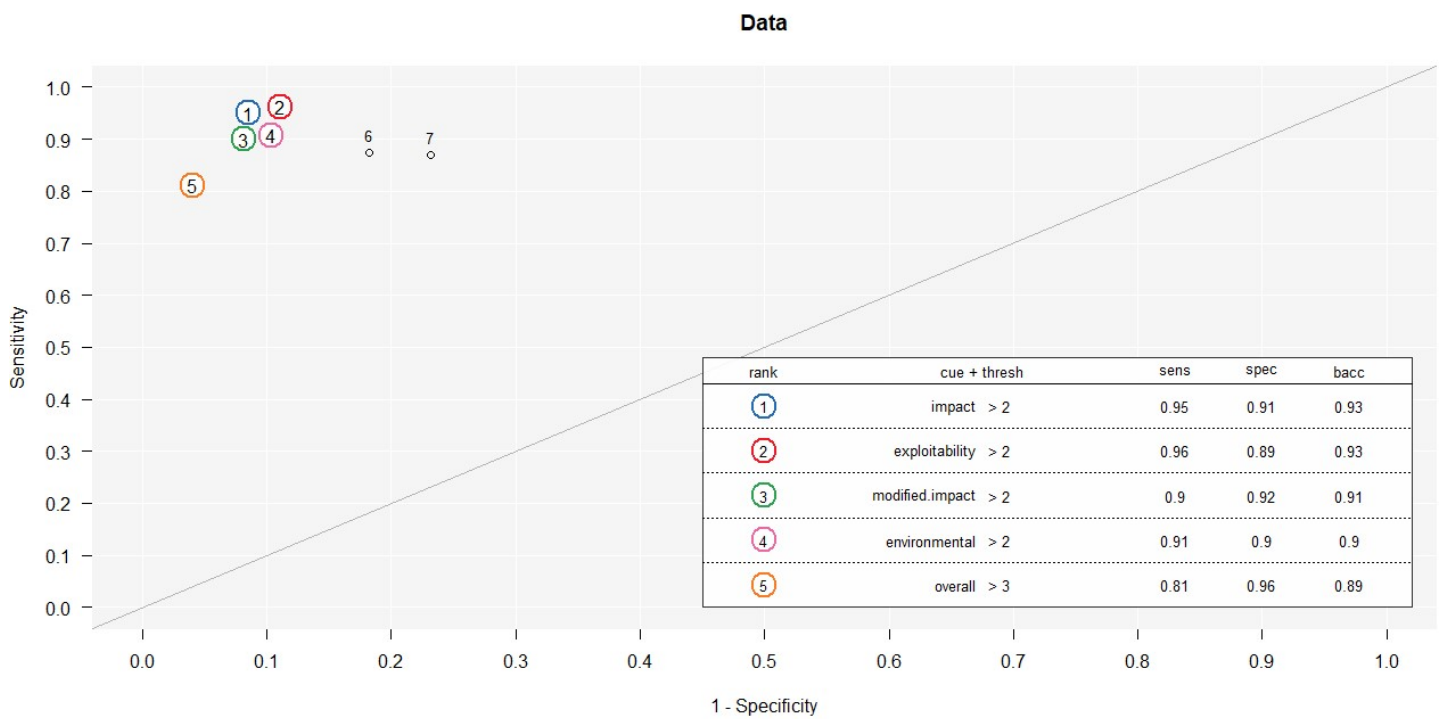
base	impact	exploitability	temporal	environmental	modified.impact	overall	critical
5	1	1	1	2	1	3	FALSE
5	4	4	5	7	10	3	FALSE
3	1	1	1	2	2	3	FALSE
6	8	8	1	3	4	3	FALSE
4	1	1	3	2	1	3	FALSE
8	10	10	8	7	10	9	TRUE
1	1	1	1	2	10	3	FALSE
2	1	2	1	2	1	3	FALSE
2	1	1	1	2	1	1	FALSE
4	2	1	1	2	1	2	FALSE
1	1	1	1	1	1	3	FALSE
2	1	1	1	2	1	2	FALSE

…does the decision model agree with the risk assessor?

```
library("FFTrees")
cvss <- read.csv("c:/coding/r/FFTrees/CVSS.csv")
cvss.fft <- FFTrees(formula = critical ~., data = cvss)

plot(cvss.fft, what = "cues")

plot(cvss.fft,
     main = "CVSS FFT",
     decision.names = c("Non-Critical", "Critical"))
```


Making fast, good decisions with FFTrees

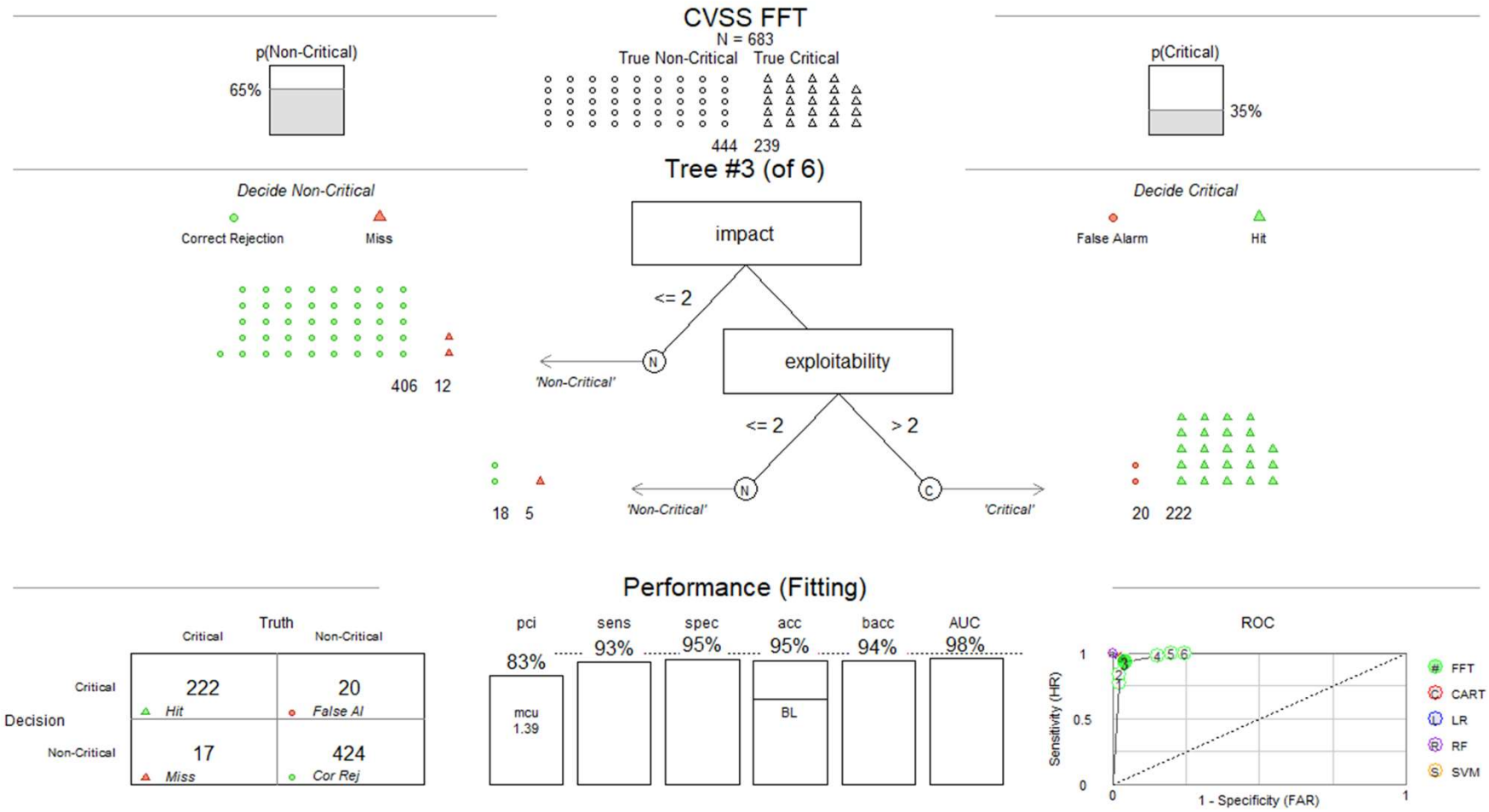
- Data points for decision model are base, impact, exploitability, temporal, environmental, modified impact & overall
- Decision model selects **impact** and **exploitability** based on *cues*, as they are best positioned away from the AUC on the ROC



rank	cue + thresh	sens	spec	bacc
1	impact > 2	0.95	0.91	0.93
2	exploitability > 2	0.96	0.89	0.93
3	modified.impact > 2	0.9	0.92	0.91
4	environmental > 2	0.91	0.9	0.9
5	overall > 3	0.81	0.96	0.89

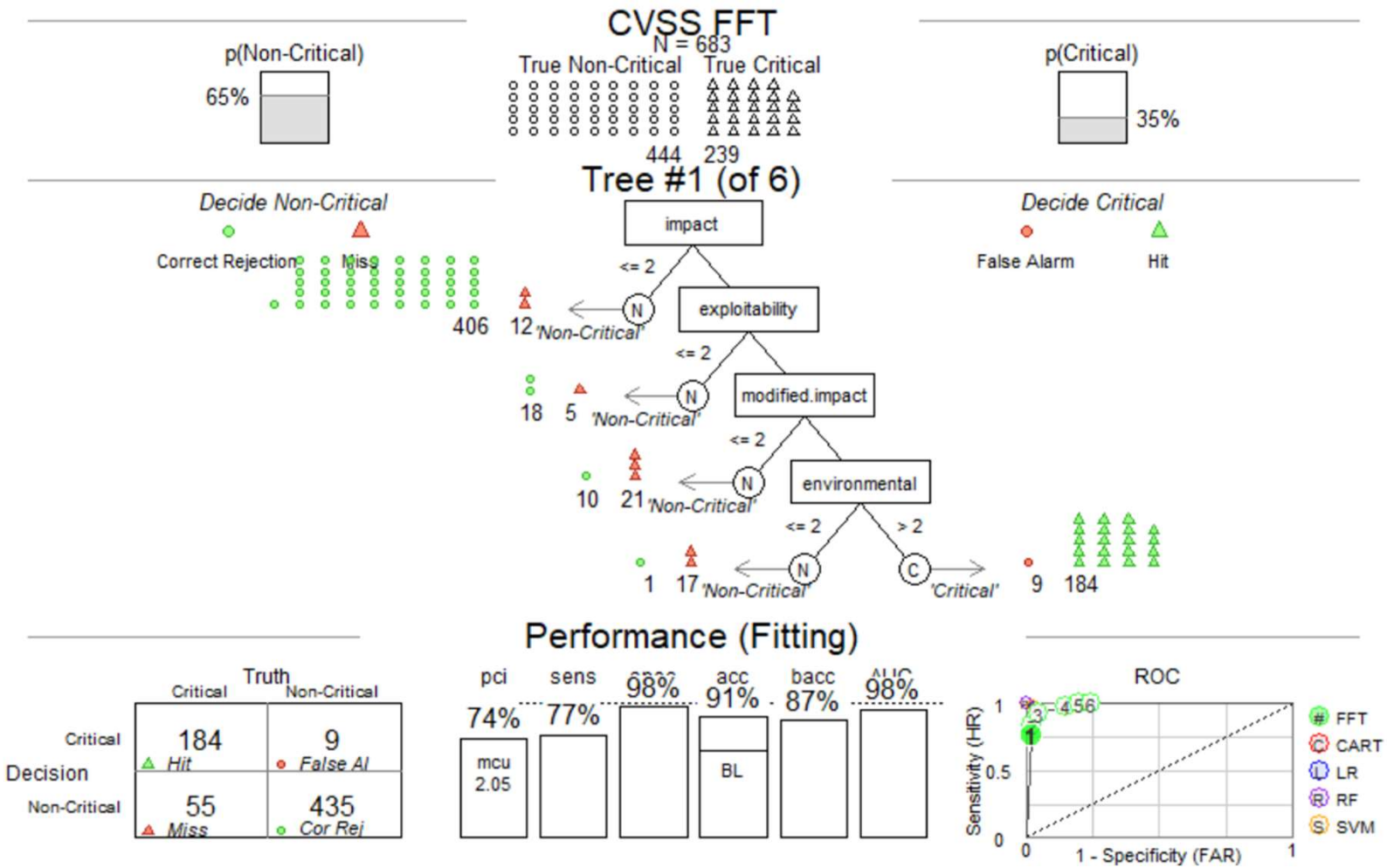
Making fast, good decisions with FFTrees

- The tree made its decision where **impact** & **exploitability** with scores equal or less than 2 were non-critical and exploitability greater than 2 was labeled critical



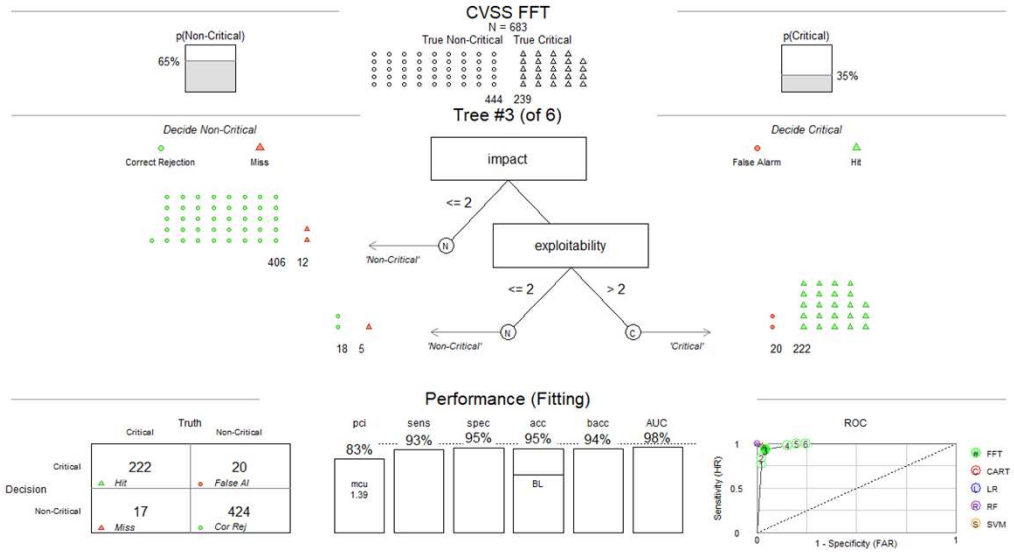
Making fast, good decisions with FFTrees

- The FFTrees function automatically builds several versions of the same general tree that make different error trade-offs.



Making fast, good decisions with FFTrees

- **Outcome:** The tree returns 222 critical findings to prioritize vs 239 determined by the risk assessor.



- **Summary:** Fast frugal trees make very fast decisions on 1 to 5 pieces of information & *ignore* all other information. In other words, FFTrees are *noncompensatory*, once they make a decision based on a few pieces of information, no additional information changes the decision.

Time Series Regression on 4624s

Scenario: Monitor user logon anomalies in high volume environments with Time Series Regression (TSR)

- Windows Event ID 4624: An account was successfully logged on
- Typically one of the top 5 events in terms of volume in the environment
- Has multiple types (e.g. Network, Service, RemoteInteractive)
- User accounts will begin to show patterns over time, in aggregate
- Seasonality: day of week, patch cycles,
- Trend: volume of logons increasing/decreasing over time
- Noise: randomness

- … candidate for behavioral detection based on established patterns

Approaches

Z-Score:

- Set a threshold based on how many standard deviations away from the average count over a given period of time
- Simple! -> the higher the value, the greater the degree of "anomalousness"

Time Series Regression (TSR):

- Statistical method for predicting a future response based on the response history (known as autoregressive dynamics) & the transfer of dynamics from relevant predictors
- Understand & predict the behavior of dynamic systems from experimental or observational data
- Commonly used for modeling & forecasting of economic, financial & biological systems

Time Series Regression on 4624s

- How to spot the anomaly in a sea of logon data?
  - "Triple Exponential Smoothing (Holt-Winters method) is one of many algorithms used to forecast data points in a series, provided that the series is **"seasonal", i.e. repetitive over some period**."
  - Winters improved on Holts double exponential smoothing by adding seasonality in 1960 and published **Forecasting sales by exponentially weighted moving averages**
  - We can detect when **a user's account exhibits changes in their seasonality** as it relates to a confidence interval established (learned) over time

https://grisha.org/blog/2016/01/29/triple-exponential-smoothing-forecasting/

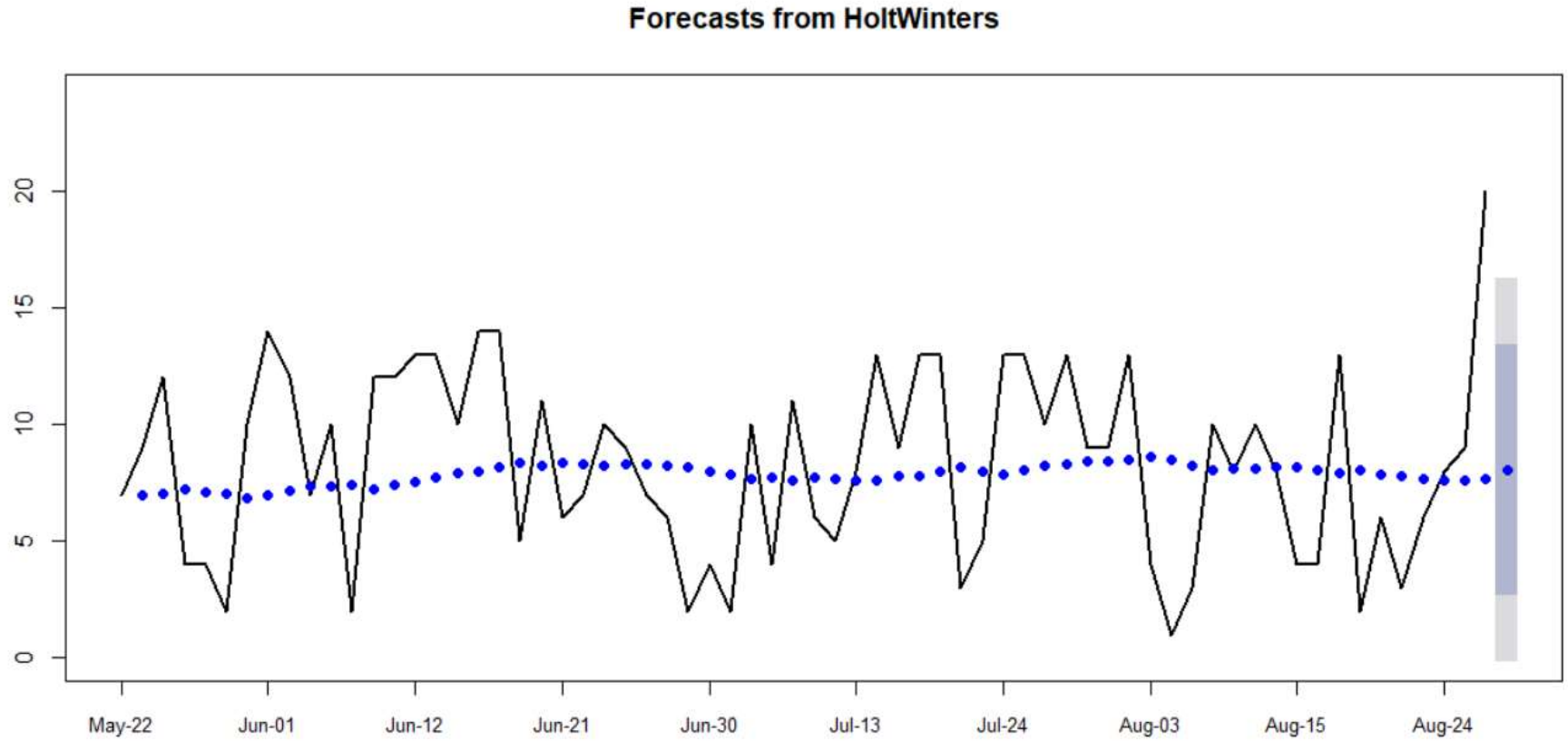
Time Series Regression on 4624s

Example: User DARPA-549521, in the SUPERSECURE domain, 90 days of aggregate 4624 Type 10 events by day

date	destinationUserName	destinationNtDomain	logonTypeCode	externalId	eventCount
22-05-17	DARPA-549521	SUPERSECURE	10	4624	7
23-05-17	DARPA-549521	SUPERSECURE	10	4624	9
24-05-17	DARPA-549521	SUPERSECURE	10	4624	12
25-05-17	DARPA-549521	SUPERSECURE	10	4624	4
26-05-17	DARPA-549521	SUPERSECURE	10	4624	4
30-05-17	DARPA-549521	SUPERSECURE	10	4624	2
31-05-17	DARPA-549521	SUPERSECURE	10	4624	10
01-06-17	DARPA-549521	SUPERSECURE	10	4624	14
02-06-17	DARPA-549521	SUPERSECURE	10	4624	12
05-06-17	DARPA-549521	SUPERSECURE	10	4624	7
06-06-17	DARPA-549521	SUPERSECURE	10	4624	10
07-06-17	DARPA-549521	SUPERSECURE	10	4624	2
08-06-17	DARPA-549521	SUPERSECURE	10	4624	12
09-06-17	DARPA-549521	SUPERSECURE	10	4624	12
12-06-17	DARPA-549521	SUPERSECURE	10	4624	13
13-06-17	DARPA-549521	SUPERSECURE	10	4624	13
14-06-17	DARPA-549521	SUPERSECURE	10	4624	10
15-06-17	DARPA-549521	SUPERSECURE	10	4624	14
16-06-17	DARPA-549521	SUPERSECURE	10	4624	14
19-06-17	DARPA-549521	SUPERSECURE	10	4624	5
20-06-17	DARPA-549521	SUPERSECURE	10	4624	11
21-06-17	DARPA-549521	SUPERSECURE	10	4624	6
22-06-17	DARPA-549521	SUPERSECURE	10	4624	7
23-06-17	DARPA-549521	SUPERSECURE	10	4624	10

Time Series Regression on 4624s

- 210 lines of R, including comments, log read, file output & graphing via *two* TSR methods gives us:



index	destinationUserName	destinationNtDomain	logonTypeCode	externalId	date	eventCount	Threshold	IsException
1	DARPA-549521	SUPERSECURE	10	4624	27-08-17	20	19	1

Next level

Seasonal and Trend Decomposition using Loess (STL)

- Handles any type of seasonality ~ can change over time
- Smoothness of the trend-cycle can also be controlled by the user
- Robust to outliers

https://www.otexts.org/fpp/6/5

Classification and Regression Trees (CART)

- Supervised learning approach: teach trees to classify anomaly / non-anomaly
- Unsupervised learning approach: focus on top-day hold-out and error check

Neural Networks

- LSTM / Multiple time series in combination

https://blog.statsbot.co/time-series-anomaly-detection-algorithms-1cef5519aef2

Be brave, be creative, go forth…

Go do: As you & your teams continue to…

- Detect attackers faster
- Containing incidents more rapidly
- Use in-house detection & remediation mechanisms

…add elements of **data science** and **visualization**, R & Python are well supported & broadly used for this mission

https://github.com/holisticinfosec/DFIR

Questions? Comments?

You can have data without information, but you cannot have information without data.
~ Daniel Keys Moran

russ@holisticinfosec.org
rmcree@microsoft.com
@holisticinfosec

