

USING R FOR DISEASE SURVEILLANCE, ENABLING CITIZEN DATA SCIENCE & CYBERSECURITY+SAFETY TIPS FOR SCIENTISTS

ISDS R Group for Biosurveillance
2018-01-24

I'd like to thank everyone for attending the webinar today. I went through the recordings of previous webinars and you've had some great presenters on some really interesting topics (I especially liked Shirin Glander's talk on machine learning models).

ABOUT://ME

- 20+ years in cybersecurity
- Author of 70+ R packages (a dozen or so on CRAN)
- Contributor to rOpenSci
- Co-author of Data-Driven Security (Wiley Press)
- Chief Data Scientist, Rapid7
- Former chief researcher & team lead, Verizon Data Breach Investigations Report



You got most of this in the bio. The most important thing on here is the Captain America shield. If you ever need to find me, just look for the shield.

DISCLAIMER: I'm not an epidemiologist.

I don't even play one on TV like this gentleman.



So, by now you may be wondering what prompted a cybersecurity professional to create a package that enables reproducible research with and dashboard creation of CDC influenza surveillance data. Hopefully I can help answer that question and also introduce you to some of the core features of the package.



What is cdcfluvie?

The "U.S." Center for Disease Control (CDC) maintains a portal (<http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>) for accessing state, regional and national influenza statistics as well as mortality surveillance data. The web interface makes it difficult and time-consuming to select and retrieve influenza data. Tools are provided to access the data provided by the portal's underlying API.

Maintainer: Bob Rudis <bob@rudis>

Author(s): Bob Rudis^{*} (0000-0001-5670-2640), Craig McGowan^{*} (0000-0002-6298-0185)

Install package and any missing dependencies by running this line in your R console:
`install.packages("cdcfluvie")`

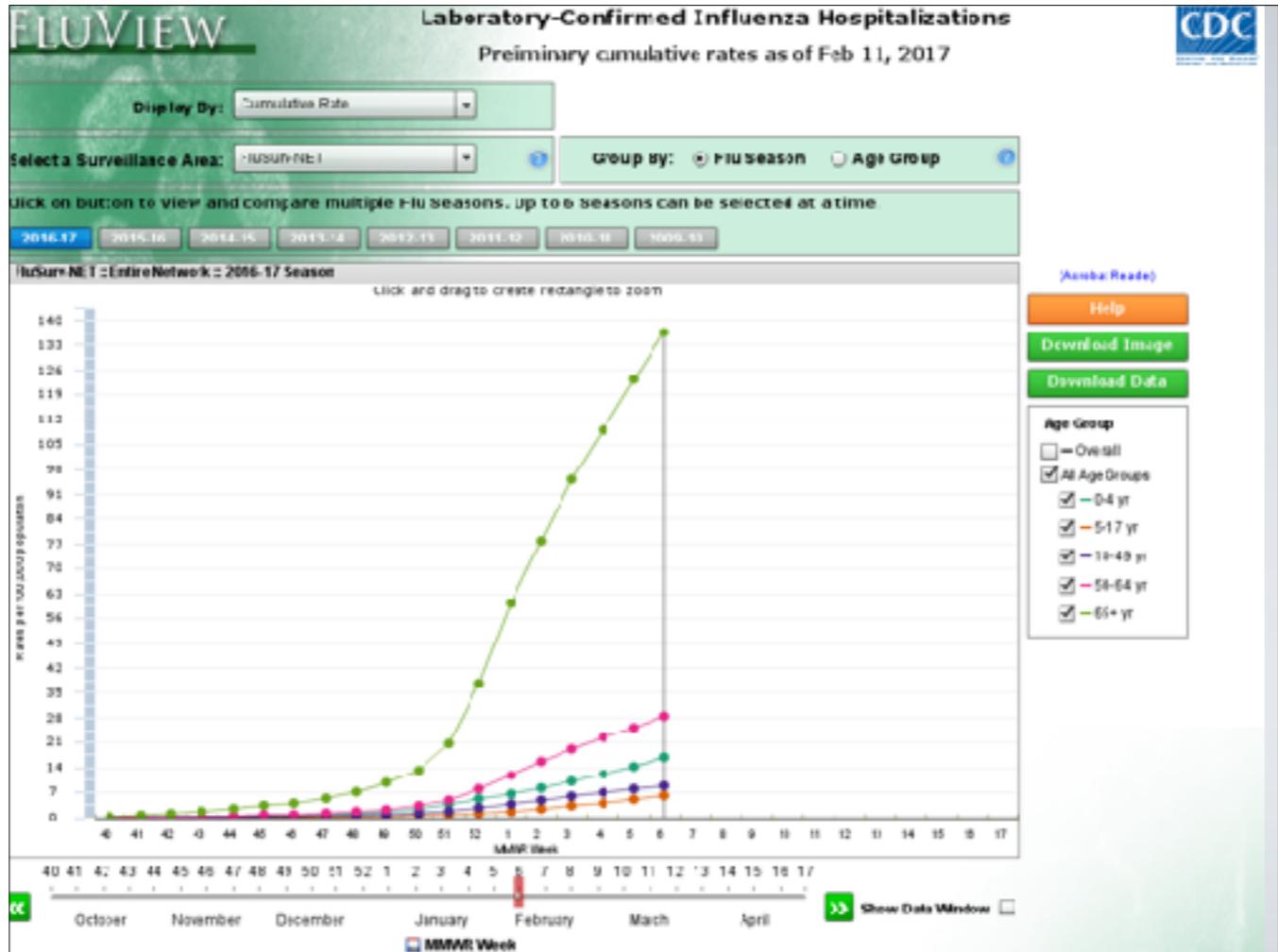
	Dependencies Table	Dependencies Graph	Reverse Table
Depends			
Imports	readr , dplyr , http , purrr , xml2 , MMWtweek , sf , units , jsonlite		
Suggests	testthat , corr		
Enhances			
Linking to			

cdcfluvie Documentation
 Manual: [cdcfluvie.pdf](#)
 Vignettes: None available.

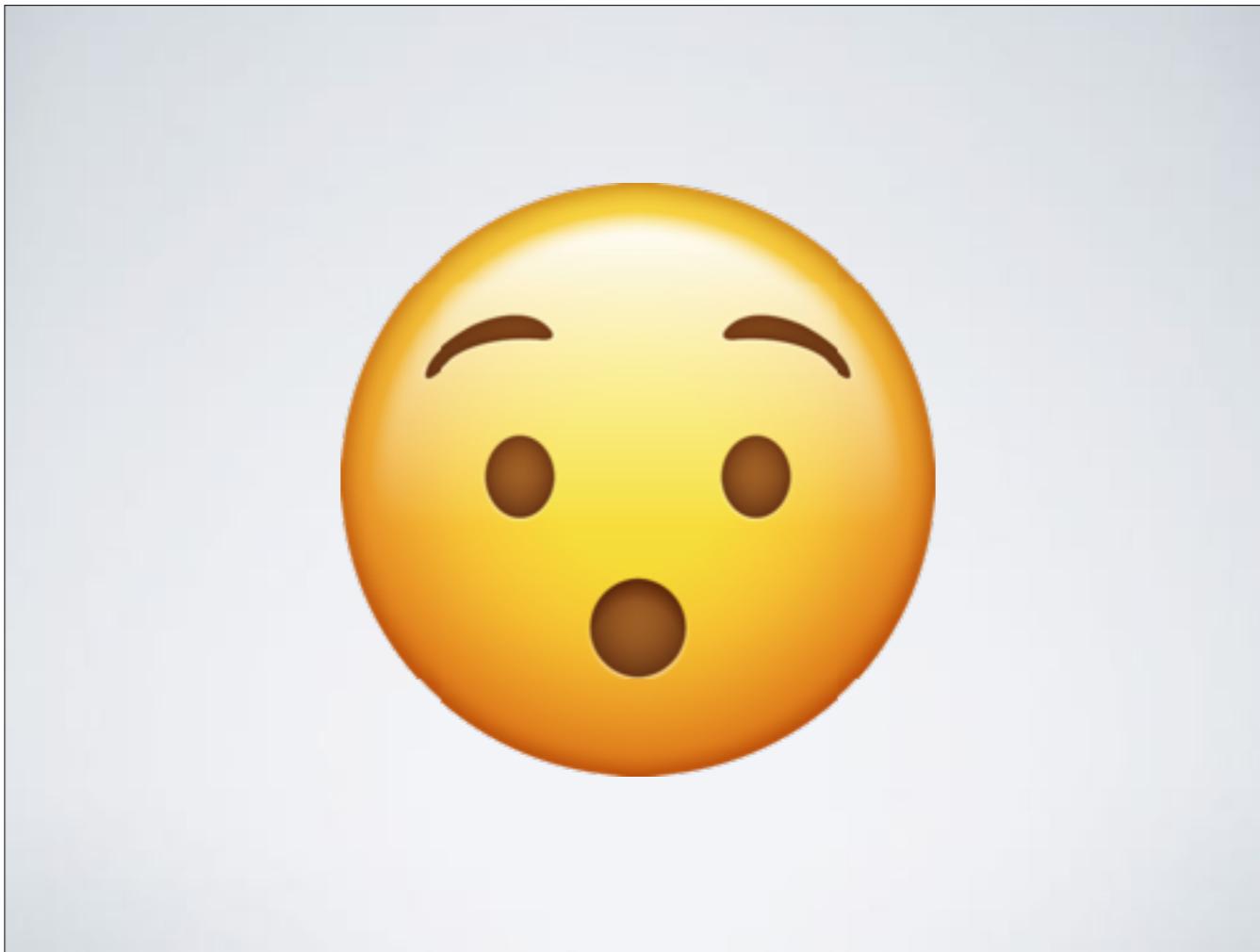
cdcfluvie
 Q

Package	cdcfluvie
Materials	
URL	https://github.com/rbrmstr/cdcfluvie
Task Views	
version	0.7.0
Published	2017-11-16
License	MIT + file LICENSE
Bug Reports	https://github.com/rbrmstr/cdcfluvie issues
System Requirements	
Needs Compilation	False
Citation	
CRAN Checks	cdcfluvie check results
Package Source	cdcfluvie_0.7.0.tar.gz

It's very likely that all of you are intimately familiar with the CDC's FluView portal. They provide a wealth of information — and some questionable visualizations — on influenza and other influenza like illnesses. The main focus of the `cdcfluvie` R package is to provide a streamlined query interface to all the data behind the FluView portal but there are also tools to work with some of the geo-data and it has sibling packages that make it straightforward to quickly produce visualizations from the data.



If you are familiar with CDC FluView, you likely remember the old interface. It was based on Adobe Flash and I was trying to help a non-cyber colleague script getting to the underlying data. Rather than just make a one-off script, I decided to dive into the hidden API and turn it into a full-featured R package. I'll talk about that more in a bit, but I have a bit of a package addiction when it comes to R. Packages are the easiest way to bundle up operations and documentation and a good package enables the user to forget about the mechanics of data access and just focus on the analysis. So, while I'm not an epidemiologist, I am fairly adept at dissecting hidden APIs (hacking experience comes in handy in many ways) and making them useful. I thought I was making one person's life easier.



But, it seems others figured out that I made the package (which surprised me since it's fairly niche package)

The screenshot shows a WordPress blog post titled "Retrieving ILI from cdcfluvie". The post is authored by Tung, published on November 6, 2017, and has 0 comments. The content of the post is a GitHub Gist containing R code. The code is used to update the package's public ILI data (ili.RData) weekly. It includes details about the package version (0.5.0), the CDC FluView API URL, and a BibTeX entry for Latent Dirichlet Allocation. The R code also specifies the author as Bob Rudis, the year as 2016, and the license as MIT-BSD. A note at the bottom states that the citation information was auto-generated. The post is part of the "resources" category.

UMN SPH EnHS Real Time Flu Forecast

Home About Us Model Descriptions ENVI Updates

Retrieving ILI from cdcfluvie

Tung November 6, 2017 0 comments resources

```
# To cite package 'cdcfluvie' in publications, use:  
#  
# Bob Rudis (2016). cdcfluvie: Retrieving U.S. Flu Season Data from  
# the CDC FluView API. R package version 0.5.0.  
# http://github.com/kkromer/cdcfluvie  
#  
# A BibTeX entry for Latent Dirichlet Allocation  
#  
# @Misc{Rudis2016,  
#   title = {cdcfluvie: Retrieve U.S. Flu Season Data from the  
#           CDC FluView API},  
#   author = {Bob Rudis},  
#   year = {2016},  
#   note = {R package version 0.5.0},  
#   url = {https://github.com/kkromer/cdcfluvie},  
# }  
#  
# ATTENTION: This citation information has been auto-generated  
# from
```

CONNECT WITH US

CALENDAR

NOVEMBER 2017

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

AUTHORS

The folks at UMN use it for their real-time dashboard (totally no pressure to ensure the package works. nope, not at all)

FluSight 2016-17

- [Home](#)
- [Forecasts](#)
- [National Forecasts](#)
- [Region 1 Forecasts](#)
- [Region 2 Forecasts](#)
- [Region 3 Forecasts](#)
- [Region 4 Forecasts](#)
- [Region 5 Forecasts](#)
- [Region 6 Forecasts](#)
- [Region 7 Forecasts](#)
- [Region 8 Forecasts](#)
- [Region 9 Forecasts](#)
- [Region 10 Forecasts](#)
- [Data](#)
- [Evaluation](#)
- [Guidance Documents](#)
- [Submit](#)

FluSight: National Influenza Forecasting

NOTE: Forecasting for the 2016-17 season has concluded. Forecasting will resume in November 2017.

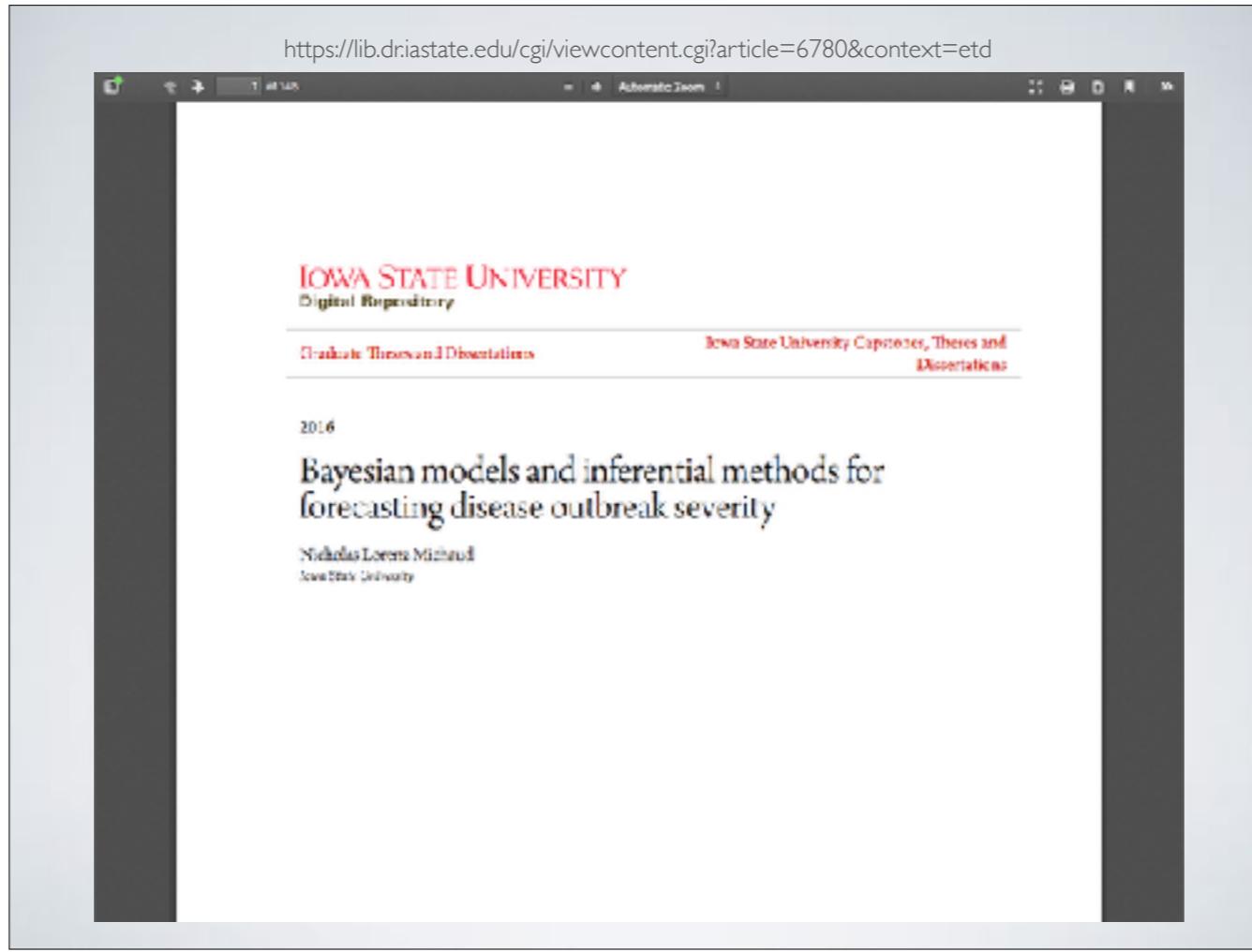
Influenza (flu) is a respiratory virus that can result in illness ranging from mild to severe. Each year millions of people get sick with influenza, hundreds of thousands are hospitalized and thousands or tens of thousands die from flu. Tracking flu activity to inform prevention measures is an important public health function that is currently performed by CDC's flu surveillance system, which can lag behind real-time flu activity. But what if it were possible to predict flu activity accurately weeks or months in advance for multiple locations? While this is not currently possible, the goal of forecasting is to produce more timely and accurate tracking that results in better tools for large-scale interventions, informed public health actions, and accurate media/sector communications, disease prevention and control. The potential benefits of flu forecasting are significant.

Since 2015, the Influenza Division at the Centers for Disease Control and Prevention has worked with external researchers to improve the science and usability of influenza forecasts by identifying seasonal influenza prediction challenges. This work includes defining prediction targets, facilitating data access, establishing evaluation metrics to assess accuracy, and developing forecast visualizations.

Twenty-one research teams have developed different flu forecasting models and are providing flu activity forecasts to CDC for the 2016/17 influenza season. This beta website houses the weekly influenza activity forecasts provided by the various research teams. It is important to note that these are not CDC forecasts and that the forecasts on this website are not endorsed by CDC. These forecasts are based on different models, can vary significantly, and may be inaccurate.

Interested in participating in the challenge? Please email flusec@cdc.gov for more information.

I also discovered it's suggested by (but not endorsed or affiliated with) the Epidemic Prediction Initiative (if you click on Data you'll see instructions)



There are a few citations of it (but papers don't really cite R packages as much as they should) so I have no real idea the extent of the usage.



At this point, you're likely thinking "Just show me the package". So let's go!

Retrieve ILINet Surveillance Data

Description

The CDC FluView Portal provides in-season and past seasons' national, regional, and state-level outpatient illness and viral surveillance data from both ILINet (Influenza-like Illness Surveillance Network) and WHO/NREVSS (National Respiratory and Enteric Virus Surveillance System).

Usage

```
ilinet(region = c("national", "hhs", "census", "state"), years = NULL)
```

Arguments

region	one of "national", "hhs", "census", or "state"
years	a vector of years to retrieve data for (i.e. 2014 for CDC flu season 2014-2015). CDC has data for this API going back to 1997. Default value (NULL) means retrieve all years. NOTE: if you happen to specify a 2-digit season value (i.e. 57 == 2017-2018) the function is smart enough to retrieve by season ID vs convert that to a year.

```
# stable / CRAN  
install.packages(cdcfluvview)  
  
# bleeding edge  
devtools::install_github("hrbrmstr/cdcfluvview")
```

```
install.packages(ggalt)
install.packages(statebins)
install.packages(hrbrthemes)
```

these are other packages of mine that I'll be using in some of the examples (I'm not going to overwhelm you with code). They're all on CRAN but each has some cutting edge functionality on GitHub (I also try to not overwhelm CRAN folks with package updates since they're volunteers). There's a link to a GitHub repository for this talk that has all of the packages being used in it, so there's no need to frantically copy everything from the screen.

```
library(ggalt)
library(cdcfluview)
library(hrbrthemes)
library(tidyverse)

ili <- ilinet("national", years = 2013:2018)

## Observations: 224
## Variables: 16
## $ region_type      <chr> "National", "National", "National", "National...
## $ region           <chr> "National", "National", "National", "National...
## $ year              <int> 2013, 2013, 2013, 2013, 2013, 2013, 201...
## $ week              <int> 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5...
## $ weighted_ili     <dbl> 1.15746, 1.27184, 1.31623, 1.37444, 1.46781, ...
## $ unweighted_ili   <dbl> 1.13249, 1.21121, 1.25682, 1.37097, 1.42956, ...
## $ age_0_4           <dbl> 2974, 3276, 3483, 3930, 4045, 4434, 4643, 483...
## $ age_25_49         <dbl> 1840, 1924, 2010, 2292, 2354, 2511, 2716, 308...
## $ age_25_64         <dbl> NA, N...
## $ age_5_24           <dbl> 3769, 3925, 3880, 4484, 4642, 5062, 5000, 557...
## $ age_50_64          <dbl> 638, 762, 785, 816, 930, 953, 990, 1083, 1059...
## $ age_65             <dbl> 456, 533, 500, 514, 531, 538, 578, 583, 639, ...
## $ ilitotal           <dbl> 9677, 10420, 10658, 12036, 12502, 13498, 1392...
## $ num_of_providers  <dbl> 1960, 1990, 2018, 2027, 2028, 2017, 2054, 201...
## $ total_patients    <dbl> 854487, 860298, 848016, 877917, 874537, 87884...
## $ week_start         <date> 2013-10-07, 2013-10-14, 2013-10-21, 2013-10-...
```

```
hhs <- ilinet("hhs", years = 2013:2018)
```

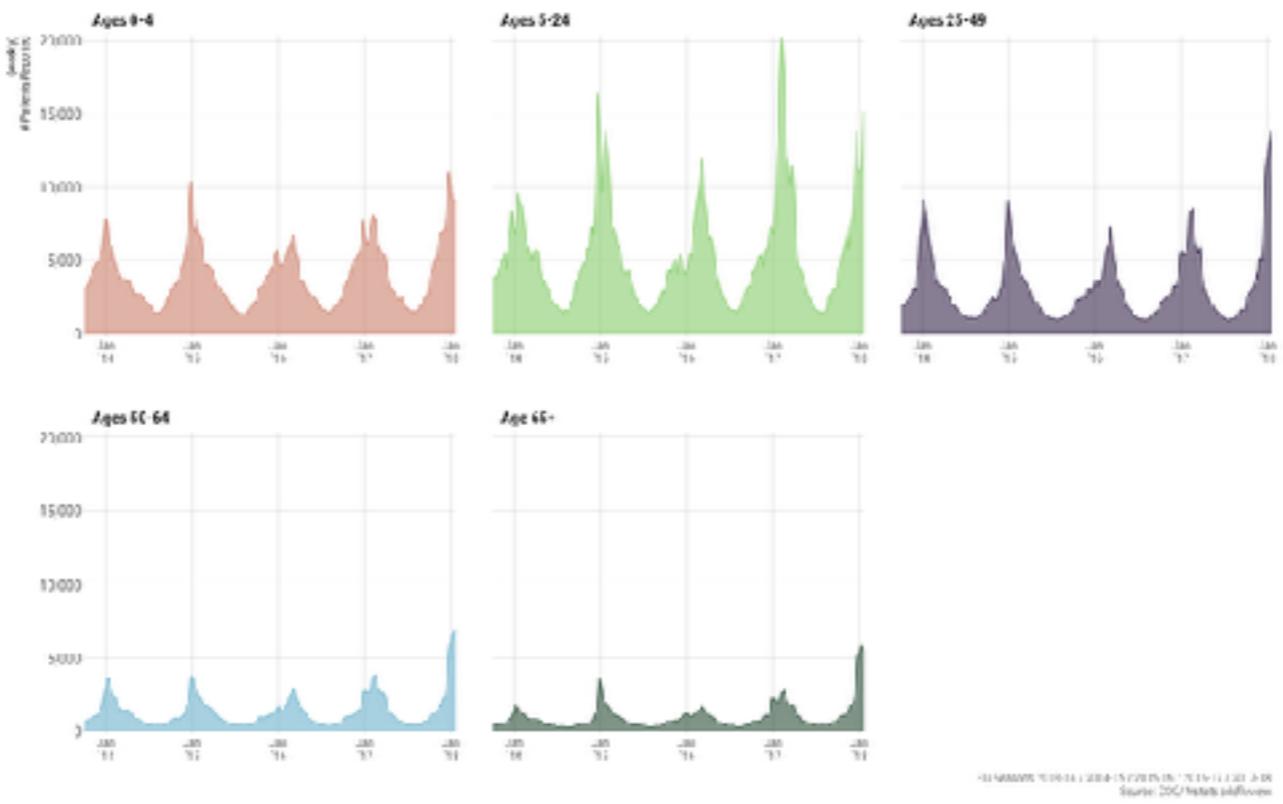
```
Observations: 2,240
Variables: 16
$ region_type      <chr> "HHS Regions", "HHS Regions", "HHS Regions", "HH...
$ region           <fctr> Region 1, Region 2, Region 3, Region 4, Region ...
$ year              <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, ...
$ week              <int> 40, 40, 40, 40, 40, 40, 40, 40, 40, 41, 41, ...
$ weighted_ili     <dbl> 0.621395, 1.607510, 0.926323, 0.943459, 0.885838...
$ unweighted_ili   <dbl> 0.615189, 1.431470, 1.094000, 1.020040, 1.139130...
$ age_0_4           <dbl> 117, 632, 401, 432, 402, 585, 48, 111, 234, 12, ...
$ age_25_49          <dbl> 44, 321, 293, 249, 207, 418, 39, 74, 178, 17, 39...
$ age_25_64          <dbl> NA, ...
$ age_5_24            <dbl> 168, 644, 465, 666, 435, 709, 71, 178, 399, 34, ...
$ age_50_64          <dbl> 19, 120, 75, 62, 110, 120, 17, 28, 79, 8, 15, 12...
$ age_65              <dbl> 20, 75, 55, 44, 76, 78, 7, 17, 82, 2, 16, 68, 46...
$ ilitotal            <dbl> 368, 1792, 1289, 1453, 1230, 1910, 182, 408, 972...
$ num_of_providers    <dbl> 176, 222, 233, 322, 270, 235, 90, 119, 246, 47, ...
$ total_patients      <dbl> 59819, 125186, 117824, 142445, 107977, 91046, 53...
$ week_start          <date> 2013-10-07, 2013-10-07, 2013-10-07, 2013-10-07, ...
```

```
update_geom_font_defaults(font_rc)
theme_set(theme_ipsum_rc(grid = "XY", strip_text_face = "bold"))

select(ili, week_start, starts_with("age")) %>%
  select(-age_25_64) %>%
  gather(group, ct, -week_start) %>%
  mutate(group = factor(group, levels = c("age_0_4", "age_5_24", "age_25_49", "age_50_64", "age_65"),
                        labels = c("Ages 0-4", "Ages 5-24", "Ages 25-49", "Ages 50-64", "Age 65+"))) %>%
  ggplot(aes(week_start, ct, group = group)) +
  stat_xspline(geom = "area", aes(color = group), fill = group, size = 2/5, alpha = 2/3) +
  scale_x_date(expand = c(0,0), date_labels = "%b\n`%y") +
  scale_y_comma() +
  scale_color_ipsum() +
  scale_fill_ipsum() +
  labs(
    x = NULL, y = "(weekly)\n# Patients Reported",
    title="Weekly reported Influenza-Like Illness (ILI) – U.S./National by Age Group",
    subtitle="All age groups except 5-24 are reporting larger number of cases this season than the previous four seasons",
    caption="Flu Seasons 2013-14 / 2014-15 / 2015-16 / 2016-17 / 2017-18\nSource: CDC/#rstudio cdcfluview"
  ) +
  facet_wrap(~group, scales = "free_x", nrow = 2) +
  theme(axis.text.x = element_text(size = 9)) +
  theme(legend.position = "none")
```

Weekly reported Influenza-Like Illness (ILI) – U.S./National by Age Group

All age groups except 5-24 are reporting larger numbers of cases this season than the previous four seasons.



```
ggplot(hhs, aes(week_start, weighted_ili, group=region)) +
  stat_xspline(geom="area", aes(color=region, fill=region),
               size=2/5, alpha=2/3) +
  geom_smooth(se=FALSE, size=1, color="#2b2b2b") +
  scale_x_date(expand=c(0,0), date_labels="%b\n`%y") +
  scale_y_comma() +
  ggthemes::scale_color_tableau() +
  ggthemes::scale_fill_tableau() +
  labs(
    x=NULL, title="Weighted ILI by HHS Region (2013/4-2017/8)",
    caption="Source: CDC/#rstats cdcfluvview"
  ) +
  facet_wrap(~region, scales="free_x", nrow=2) +
  theme(axis.text.x=element_text(size=9)) +
  theme(legend.position="none")
```



```
library(statebins)

flu <- ili_weekly_activity_indicators(2017)

filter(flu, weekend == last(weekend)) %>%
  statebins(state_col = "statename",
            value_col = "activity_level",
            round = TRUE,
            ggplot2_scale_function =
              viridis::scale_fill_viridis,
            name = "ILI Activity Level") +
  labs(title = sprintf(
    "U.S. ILI Weekly Activity : Week Ending %s / 2017-18
Season",
    last(flu$weekend))
  ) +
  theme_statebins(base_family = "Roboto Condensed")
```

U.S. ILI Weekly Activity : Week Ending 2018-01-13 / 2017-18 Season



ILI Activity Level
0.0 2.5 5.0 7.5 10.0

```
library(statebins)
library(magick)
library(tidyverse)

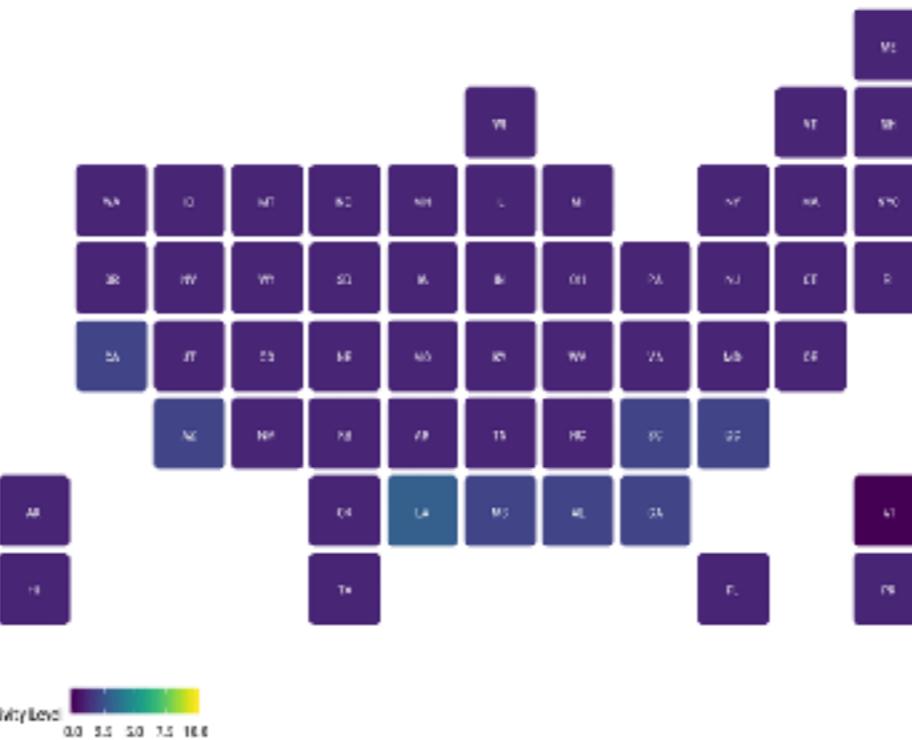
flu <- ili_weekly_activity_indicators(2017)

frames <- image_graph(width=1800, height=1200, res=144)

arrange(flu, weekend) %>%
  pull(weekend) %>%
  unique() %>%
  map(~{
    filter(flu, weekend == .x) %>%
      statebins(state_col = "statename", value_col = "activity_level",
                round = TRUE,
                ggplot2_scale_function = viridis::scale_fill_viridis,
                limits=c(0,10),
                name = "ILI Activity Level ") +
    labs(title = sprintf("U.S. ILI Weekly Activity : Week Ending %s / 2017-18", .x)) +
    theme_statebins(base_family = "Roboto Condensed") -> gg
    print(gg)
  }) -> y

gif <- image_animate(frames, 1)
image_write(gif, "fluvview.gif")
```

U.S. ILI Weekly Activity : Week Ending 2017-10-07 / 2017-18



age_group_distribution	Age Group Distribution of Influenza Positive Tests Reported by Public Health Laboratories
cdc_basemap	Retrieve CDC U.S. Basemaps
census_regions	Census Region Table
geographic_spread	State and Territorial Epidemiologists Reports of Geographic Spread of Influenza
get_weekly_flu_report	Retrieves (high-level) weekly (XML) influenza surveillance report from the CDC
hhs_regions	HHS Region Table
hospitalizations	Laboratory-Confirmed Influenza Hospitalizations
ilinet	Retrieve ILINet Surveillance Data
ili_weekly_activity_indicators	Retrieve weekly state-level ILI indicators per-state for a given season
mmwrid_map	MMWR ID to Calendar Mappings
mmwr_week	Convert a Date to an MMWR day+week+year
mmwr_weekday	Convert a Date to an MMWR weekday
mmwr_week_to_date	Convert an MMWR year+week or year+week+day to a Date object
pi_mortality	Pneumonia and Influenza Mortality Surveillance
state_data_providers	Retrieve metadata about U.S. State CDC Provider Data
surveillance_areas	Retrieve a list of valid sub-regions for each surveillance area.
who_nrevss	Retrieve WHO/NREVSS Surveillance Data

There are a few more functions and built-in data sources that map directly to the areas on the CDC FluView portal. If I can make any of the interfaces, documentation any better or add some functionality that would help you out, don't hesitate to file an issue on GitHub.

WHY CDCFLUVIEW?

I mentioned the history of how FluView was created earlier. But, really. Why would I spend time creating this thing that requires care and feeding? And it most certainly does. Craig McGowan (a CDC researcher) notified me about the migration from Flash to something more open and safe last year and took a stab at ensuring all the functions worked after the cutover. I went back and reorganized the package interface to better map to the CDC's site and to also make the data a tad more useful. But, then again, why?



Initially, one of the drivers was to help someone not have to deal with the Flash interface. As a cybersecurity researcher, Flash is absolutely anathema. It's insecure, it destroys the semantic web and can be used to hide data. It's nothing but evil. A big motivator was to help ensure scientists and researchers could use a non-Flash-enabled browser. As you'll see in a bit, I'm a bit protective of the data science community as they're some of the most vulnerable users out there.

REPRODUCIBILITY

But, a bigger reason was reproducibility. What do I mean by reproducibility?

https://scholarlykitchen.sspnet.org/2017/05/24/reproducible-research-just-not-reproducible/

The Scholarly Kitchen is a blog by the Society for Scholarly Publishing. The post, titled "Reproducible Research, Just Not Reproducible By You," discusses the challenges of reproducibility in research. It features a cartoon illustration of a complex laboratory setup.

**THE SCHOLARLY
kitchen**
What's New and Cooking In Scholarly Publishing

ABOUT ARCHIVES COLLECTIONS CHEFS PODCAST FOLLOW

Reproducible Research, Just Not Reproducible By You

By DAVID COOPER | MAY 24, 2017 | 10 COMMENTS

DATA PUBLISHING | RESEARCH | TECHNOLOGY

174 2000

We tend to think of research as either being reproducible and thus valid, or irreproducible, and questionable. That sort of binary thinking is problematic, because there's a large body of research that's entirely accurate but not easily reproducible. Do we need a new term for research fall into this in-between zone?

At the recent [STM Annual Meeting in Washington](#), Masha Pitkänen, founder and CEO of the [Journal of Visualized Experiments](#) (JVE) gave a talk about the growing role present in life science in scientific reproducibility. Enormous amounts of effort, money, and regulation have been put toward opening up the often behind-the-scenes experiments. But very little attention seems to have been directed toward the protocols and methodologies used to collect these data.



It means that someone can take your code and run it and get the same results (for the same data). Or be able to use it with new data but with consistent assumptions and diagnostic output. My own discipline is terrible about reproducible research, but then again, green text on black background blog posts count as research papers in cybersecurity, so it's no real surprise.



This is one area on the current CDC FluView portal. Folks are encouraged by the CDC to go there to get data they need.



This does not
truly enable
either speed or
reproducibility

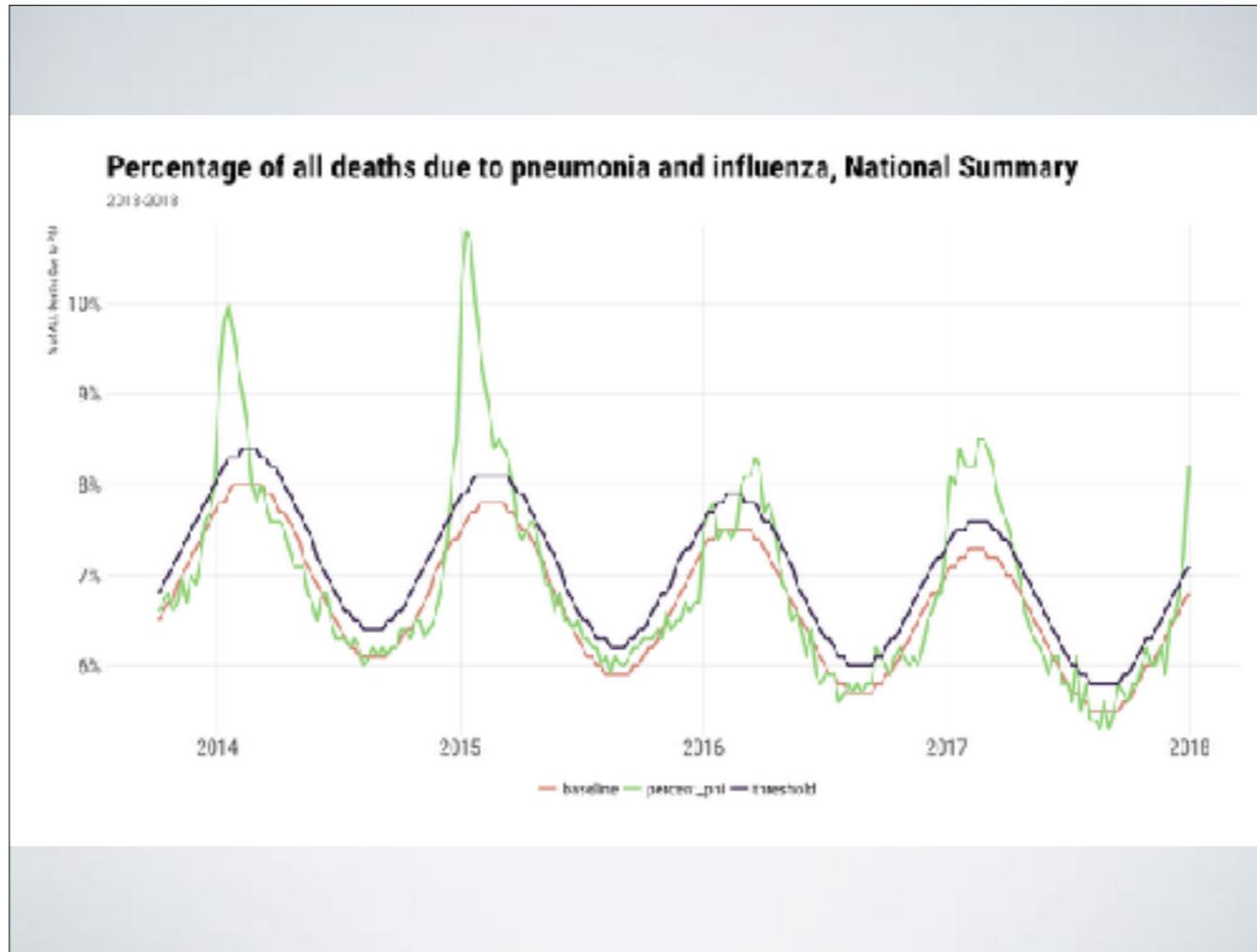
But, they were before the site was in HTML and the interface did change, so all the manual instructions for "go here and click X" were wrong and any automation based on instrumenting a browser also stopped working. Sure, my package had to be modified slightly, but the users had no real disruption in the API (I kept the old functionality even as I introduced a new, better one).

And, any workflow that involves a human means you can't really do:

```
mort <- pi_mortality(years=2013:2018)

select(mort, wk_end, baseline, threshold, percent_pni) %>%
gather(measure, value, -wk_end) %>%
ggplot(aes(wk_end, value, group=measure)) +
geom_line(aes(color=measure)) +
labs(
  x=NULL, y="% of ALL Deaths Due to P&I",
  title="Percentage of all deaths due to pneumonia and influenza, National Summary",
  subtitle="2013-2018"
) +
theme_ipsum_rc(grid="XY") +
scale_color_ipsum(name=NULL) +
scale_y_percent() +
theme(legend.position="bottom") +
theme(legend.direction="horizontal")
```

this. And run it every week to get



this. It's not reproducible. And if you manually download the file to the wrong place, things just stop working. I could go on for a while abt reproducibility, but let's keep moving.

ENABLING CITIZEN DATA SCIENCE

Another "cause" / driver for me is to ensure there's data literacy in the general public and enable+enable "Citizen Data Science"

https://www.ngdata.com/the-rise-of-the-citizen-data-scientist/

THE RISE OF THE CITIZEN DATA SCIENTIST

Last update: January 19, 2018 · Lukas Mergen · My blog · My no comments



We have written a few times about the data scientist profession here in this space (and by the way, we're hiring for that role as we speak). The role of data scientist is unique in any organization. As we say in our own job description, the data scientist's function is at the core of most successful big data work. Data scientists have the critical responsibility of humanizing organizational data to help businesses better understand their consumers. In short, they serve as the interface between data and action, responsible for creating and executing strategies that will enable companies to gain a deeper and more nuanced understanding of their users.



Requirements for being a data scientist are pretty rigorous, and truly qualified candidates are few and far between. I wrote about that subject for *Wired* last year, and you can read it [here](#). Not everyone has the technical background, the personal skills and the business acumen to be able to pull it off.

But recently, I attended Gartner's Business Intelligence & Analytics Summit in Las Vegas where I kept hearing the same phrase over and over again – the citizen data scientist. The idea here is that tools and technology have advanced to a point where everyday users within an organization can leverage them to perform analytic

We're producing scads of data in virtually every field and publishing it, but a good chunk of the general public has no idea how to consume it or work with it. I and my students (I teach data science in community colleges ... more on that in a bit, too) knew the flu season was getting worse before it made headlines ... because we were looking at the data. I've got my own personal dashboard and this package is part of it (partly so I can monitor for changes in the CDC's hidden API).

We really need to find ways of getting the public more interested in this type of data and other data if we're going to have an informed society who can put checks and balances on things and also not just take the word of the media (or, even government) for things.

With the current anti-data/anti-science makeup of the U.S. government, more and more data sources are going away, so having interfaces like cdcfluvie also helps in archiving this data. But, I'm almost on a soapbox and shld back away slowly and move on.

ASA Community

HOME UNITS/STANDARDS COMMUNITIES - INDUSTRIES - CONFERENCE - CHAPTERS - NEWSLETTER BLOGS - MEMBER - PARTNERS -

SEARCH 

Data Science/Analytics Courses/Programs in Two-Year Colleges

By Steve Pierson posted 04-25-2017 21:50

Recommend

The ASA has been seeking to help two-year colleges in their deliberations to offer data science/analytics courses or start data science/analytics programs. As part of this effort, we compiled an informal and incomplete list of such programs. The recent query to ASA Connect spurred me to share our list and also ask for help to make it more complete.

The NCFI has funded an ASA proposal to host the [Two-Year College Data Science Summit](#), May 9-10, 2018 in Washington, DC. Please visit the website to learn more.

The data science/analytics courses or programs at two-year colleges of which we are aware are the following. Please let me know what we're missing and I'll update this list.

- Butte-Hill Community College, MT: Data Management (First Year) Certificate Program
- Community College, PA: Data Analyst/Certificate
- Community College of Allegheny County, PA: Data Analytics Technology Associate of Science
- Great Bay Community College, NH: Certificate in Practical Data Science
- Johnson County Community College, KS: Data Science, Data Analytics Certificate
- Manchester Community College, NH: Applied Data Analytics Certificate
- Montgomery College Data Science Program, MD: Its Dolegnywa Curriculum Committee has approved four data science courses, the first to start the fall.
- Nashua Community College, NH: Foundations in Data Analytics
- Normandale Community College, MN: Data Management & Analysis
- Roanoke College, NC: Data Analytics TAACCT
- Wake Tech Community College, NC: Associate in Applied Science in Business Analytics
- Windham Community College, VT: Applied Data & Data Certificate

You might also be interested in these other data science/analytics resources:

- [ASA Statement on Data Science](#) on ASA Executive Director Paul Wiegert's blog (warning: long post)
- ASA's new [Data Science Working Group](#) (August 2015 Annual News, July 2015 Annual News)
- ASA's new [Master's Data Science program](#) ([apply to master's now](#); more ASA coming in June 2017 issue)
- ASA's new [Community College statistics initiatives](#) ([apply to initiative now](#))
- [Universities and Colleges Creating New Undergraduate Statistics Data Science Programs](#), November 2014 ASA Community blog entry
- [The Emergence of Master's in Analytics and Data Science Programs](#), August 2016 ASA Community blog entry

See [other ASA Science Policy programs](#). For ASA science policy updates, follow @asa_science on Twitter.

3 comments 0 likes

Permalink

<http://community.amstat.org/blogs/steve-pierson/2017/04/25/data-science-analytics-coursesprograms-in-two-year-colleges>

http://community.amstat.org/blogs/steve-pierson/2017/04/25/data-science-analytics-coursesprograms-in-two-year-colleges

I helped develop a data science program for the college system of New Hampshire and am involved with other efforts at the ASA on developing a core, standard curriculum for 2 year colleges to get students through either a certificate program in data science or a full associates degree. This is designed so that the average citizen can come in and leave with data literacy and data science skills and also to give 4-year and beyond students a leg up when they take their biology, ecology, sociology or other major courses. A friendly interface to CDC surveillance data truly makes the topic far more approachable.

WHY **R** FOR ~~DISEASE-SURVEILLANCE~~ **EVERYTHING?**

The package and the courses I teach are all in R. But why R?

**Deep
down you
already
know the
truth.**



Packages are one big area that make R great for data science. Sure, Python has orders of magnitude more overall packages, but R's packages are usually written by practitioners who know what others need and try to ensure the focus is on data quality and also work to make them easy to use.

The screenshot shows a web browser displaying a research article from the **ONLINE JOURNAL OF PUBLIC HEALTH INFORMATICS**. The article is titled **Detection of Outbreak Signals Using R**. The authors listed are Steve J. Ricci, George Tambelkou, Richard DeCoste, Arash Paydar, Brian J. Johnson, and Jennifer L. Clark. The journal issue is **Br. J. Public Health Inform.**, 2014, 3(1), e9. The PMID is 25648284. The article is dated 2014 Mar 29. The abstract discusses the development of a statistically rigorous automated process for weekly community disease report analysis to improve the speed and accuracy of outbreak detection. The methods section notes the use of a 10-year community-wide disease database (DCI-01) to model the background frequency of reported diseases. The results section indicates that the system can detect outbreaks with strong statistical power. The conclusion section states that the system can generate alerts for "alarm triggers" in epidemiological surveillance. The right sidebar includes links for **Homestatic**, **Share** (Facebook, Twitter, Google+), **Save Item** (Add to Favorites), and **Recent Activity**.

You've got many R packages for disease surveillance.

 **Temporal and Spatio-Temporal Monitoring and Modeling of Epidemic Phenomena**

Statistical methods for the modeling and monitoring of time series of counts, proportions and categorical data, as well as for the modeling of continuous-time point processes of epidemic phenomena. The package includes methods for an alertation detection in count data time series from public health surveillance of rare events/phenomena, for application in such areas as medical vigilance, environmental monitoring, quality engineering, econometrics or social sciences. The package implements many typical surveillance procedures such as the Poisson GLR, Poisson-Gaussian, EWMA, CUSUM, InflUMLM method of Hensler and Paul (2008) [doi:10.1214/07-EI015](#), a novel CUSUM approach combining logistic and environmental logistic modeling it also included. The package contains several functions to test for the ability to simulate count data, and to visualize the results of the monitoring in a temporal, spatial or spatio-temporal fashion. A recent overview of the available methods can be seen is given by Salmeron et al. (2011) [Gutiérrez-Peña et al. \(2011\)](#). For the retrospective analysis of epidemic spread, the package provides three discrete-epidemic modeling frameworks: likelihood maximization, Bayesian inference, and simulation. The MCMC estimates models for multivariate count time series following Paul and Held (2011) [doi:10.1080/10618600.2011.557744](#) and Meyer and Held (2011) [doi:10.1080/10618600.2011.557745](#). The MCMC module is the same module for discrete time series as a time event history or a fixed population, e.g. epidemics detected or measured, as an alternative package is proposed by Höhle (2009) [doi:10.1080/20930850.200902501](#). In addition, estimates self-exciting point process models for a spatio-temporal point pattern of infectious events, e.g., time-stamped georeferenced surveillance data as proposed by Meyer et al. (2012) [doi:10.1111/j.1465-3516.2011.00684.x](#). A recent overview of the implemented space-time modeling frameworks for epidemic phenomena is given by Meyer et al. (2012) [doi:10.1080/10618600.2012.670381](#).

Maintainer: Sebastian Meyer s.meyer@fau.de
 Contributors: Michael Höhle*, Alexander Paul*, Stephan Wald*, Rainer Ruckenstein*, Philipp Camphausen*, Matthias Hofmann*, Christian Lang*, Jana Meissner*, Stefan Schmid*, Daniel Salje*, Stephan Stellmacher*, Michael Wirsching*, Stefan Werner*, Elisa Zampi*
 *A few code segments are modified versions of code from Baier et al.

 Install package and any missing dependencies by running this line in your R console:
`install.packages("surveillance")`

Dependencies	Dep. lib. reqs. R 3.0.2	Reverse Table
Imports	stats, graphics, grid	
Imports	spatstat, rgeos, time, tcltk, nortest	
Suggests	Trajopt, Intervals, geosphere, rcolorbrewer, lattice, rgeos, gridExtra, water, raster, RMySQL, RSQLite, DBI, RPostgreSQL, MySQL, dplyr, DBI, RMySQL, RSQLite, RPostgreSQL, gridExtra, RMySQL, RSQLite	
References		
Linking		
CRAN checks		
Package source		

Task View	surveillance
Tasks	High-level Disease Early Detection
Task View	Time Series, Time Series, spatial statistics, Environmental
Version	1.12.0
Published	2017-12-09
License	GPL-3
Tags	spatio-temporal
System requirements	
Needs compilation	True
Changelog	
CRAN checks	surveillance check results
Package source	surveillance_1.12.0.tar.gz

Some don't even try to have clever names (many R packages try to be clever name-wise)

CONNECTIVITY

- Spark
- Apache Drill
- Every database (without needing to know SQL)
- Every file format

But, there's also connectivity.

R was meant to be extensible and you can call python from it or use any (and I mean any) database or data format. It has support and with a friendly interface.

VISUALIZATIONS

- ggplot2
- htmlwidgets (**plotly**)
- Shiny
- (ugh) Tableau
- Many others

I tend to overemphasize the vis side of data science. One reason for that is that cybersecurity folks don't trust statistics and are still pretty wary about machine learning. We're still in the "counting things" phase (it's really a protoscience). All of the visualizations I showed today were in ggplot2. It's by far the best static vis system. Period.

But, there's a dynamic side to R and htmlwidgets enable easy creation of shareable interactive visualizations.

Shiny is a web app framework for R that makes it straightforward (I say "easy" deliberately and I mean straightforward here...it's got a learning curve) to create high quality apps backed by your amazing predictive models.



Likely the best reason to consider R for anything (but especially for disease surveillance) is the community. I work with many of the rOpenSci folks and they're amazing.

rOpenSci Packages

Our packages are carefully vetted, staff and community curated R software tools that lower barriers to working with scientific data sources and data that support research applications on the web. Read our [Blog](#) to learn how to use open source packages or contribute to their improvement.

[Browse our `bioconductor` and `usebio` repos](#)

Curious about contributing your package? See [Onboarding](#) for details. We make our [volunteers](#) review packages submitted to our open peer review process.

FILTERS
 All Analytics Data Processing Tools Visualization Databases Generics Web
 Image Processing Infrastructure Computing Infrastructure Security Textmining Quantitative Methods

Type to search...

PACKAGE	MAIN AUTHOR	DESCRIPTION	DRIVER
active	Jeroen Ooms	R Client for ETIF/NCI17 Protocol	CRAN 
agent	Jeroen Ooms	Encrypted KeyValue Store for Sensitive Data	CRAN 
ain	Scott Chamberlain	R wrapper to (unofficial) ZFS plugin ecosystem (by Plan9 ZFS author, additional layers built on this and work out of the box (CrossRef, DataCite Publishers, and the Public Knowledge Project (PKP))	CRAN 
arviz	Jérôme Duriez	Bayesian Inference Posterior Summaries	CRAN 

If you don't know about rOpenSci, definitely give them a look. They have scads of packages. Tons of learning materials. A discussion forum and more. They are the smartest and truly nicest folks I know.



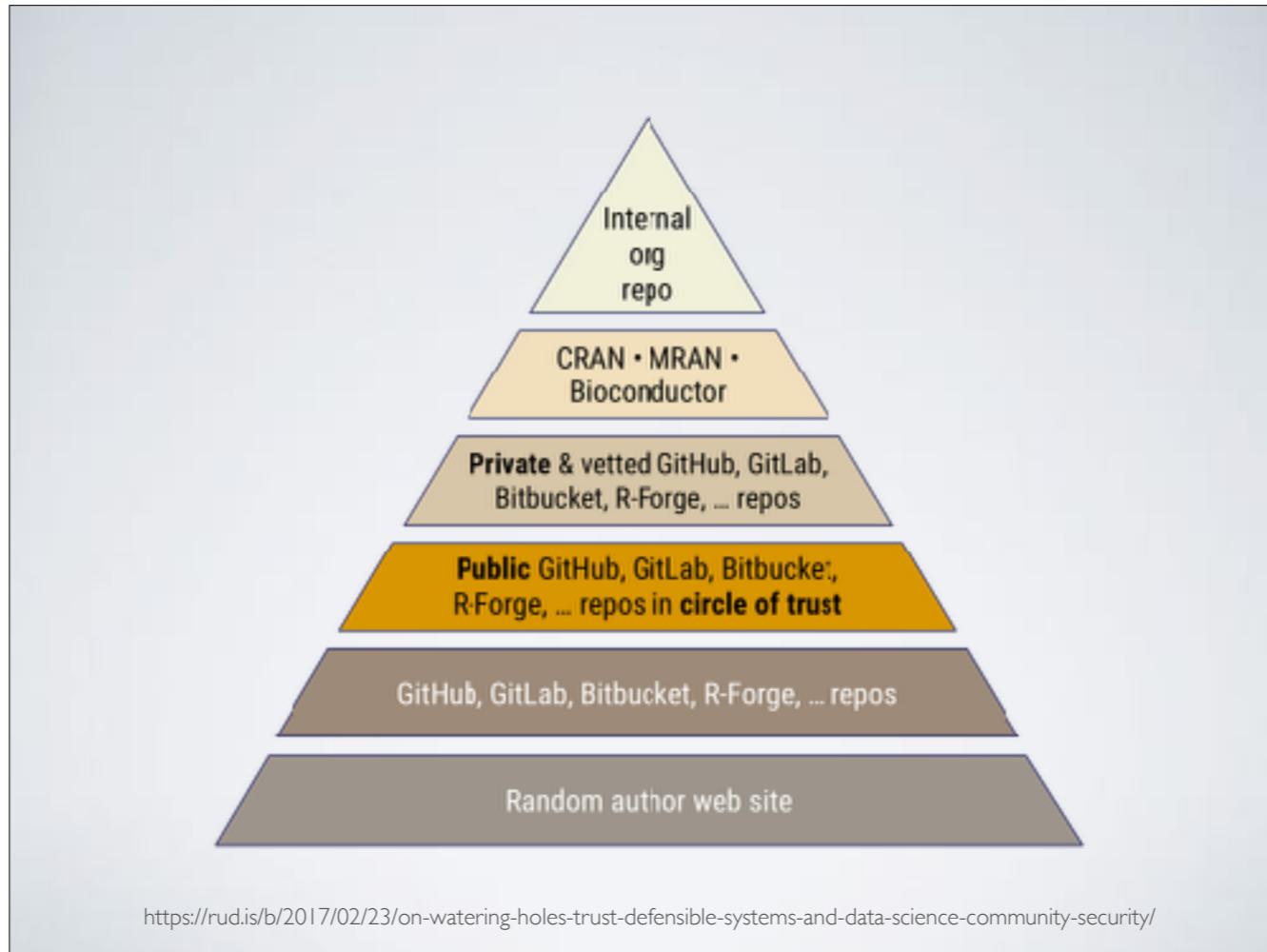
Time's closing in, but I can't leave you without talking about security and safety when it comes to data science.



When you download an app, package or even visit a link on your workstation, you're opening up you and your computer to harm.

I'm not suggesting you go live in a cave in Maine (though it is nice up here!)

But, you should be aware of the potential safety & security issues whenever you decided to load up some new "data science" thing.

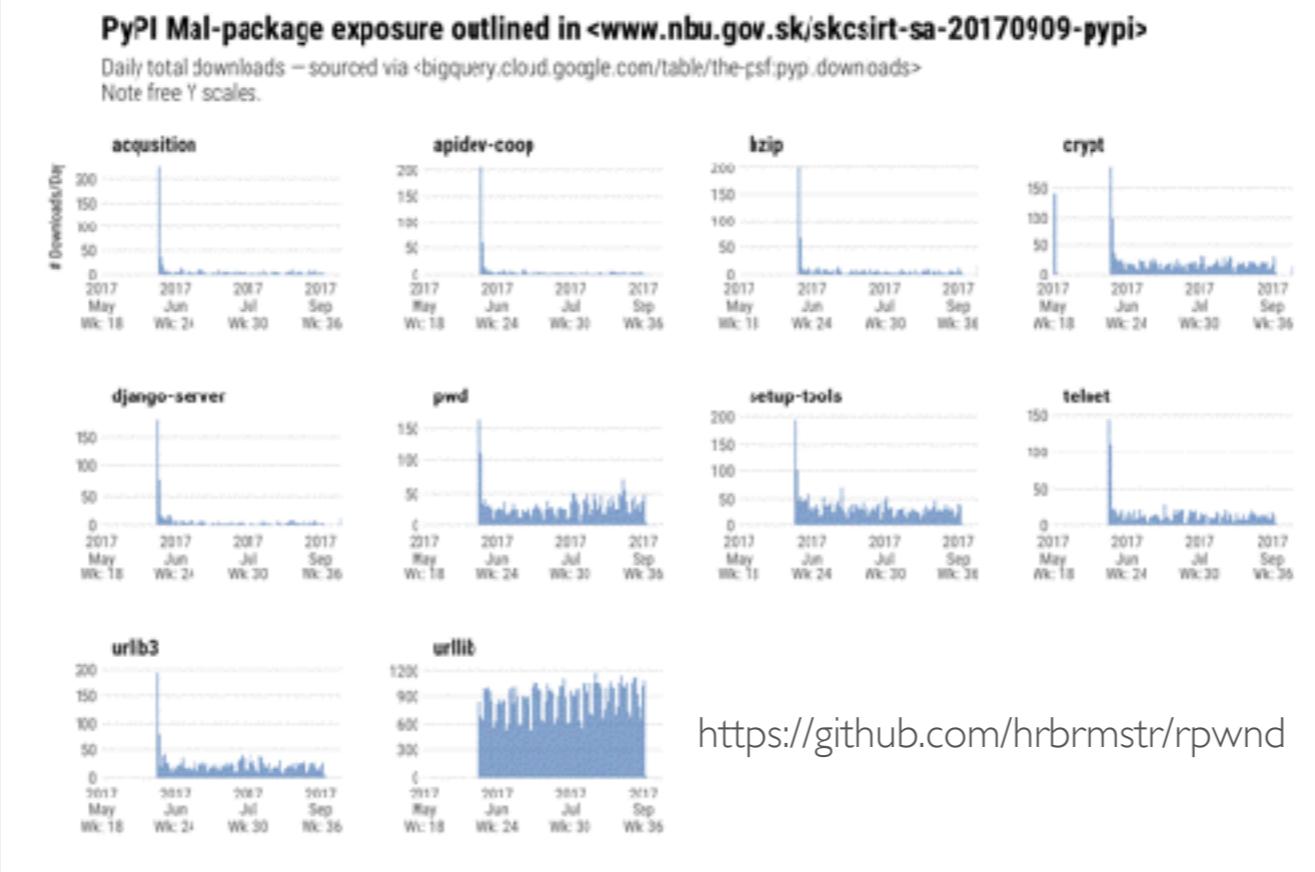


I put a blog post together last year that talks about this in detail, so I'll just hit the highlights. First, you really need to be careful about what you decided to load up in R (or Python, etc). This is my trust pyramid for packages. Stuff at work or home is super gd. CRAN & MRAN (which is fed from CRAN) are good too. I should really remove Bioconductor from that row, though. It's a hot mess.

If I know the folks, private/vetted repos are ok as well.

Things on github that are by contributors I know and especially signed (ask me about that if you want to know more) are ok, too, but I'm very wary of anything outside my circle of trust and read through every line before using (even the C/C++ code).

Generic GitHub repos or random code on web sites are right out unless you read every line and know what you're doing.



Think it can't happen? The Python packages listed here were all "hacked" last year. The graphs show (free Y scales) the number of weekly downloads of them. Sure, these aren't "data sci" packages (well the URL ones could be used by some) but these are the known ones. PyPI security & integrity is laughable, especially when compared to CRAN.

What could a malicious package do? Hit the URL at the bottom to see more (and, trying it out will have a bigger impact than if I just tell you).

I could go into much more on this, but the core message is that you are a target. Whether it's organized crime, nation-states (even your own) or some rogue organization, they all know they can collect or corrupt data to cause whatever end they do. So, while you also need to worry about data quality, you also need to be concerned with what you click, what you use and how much trust you're putting in those things.

- bob@rud.is
- bob_rudis@rapid7.com
- @hrbrmstr
- <https://rud.is/b/>
- <https://github.com/hrbrmstr/>
- <https://github.com/hrbrmstr/2018-01-24-isds>



I've covered a wide array of topics and this presentation and the R code in it + all the links are in the github repo at the bottom.

If you're on Twitter, I do regular "PSAs" about important security topics you need to know about (like the latest vulns/attacks and how to protect yourself).

But, don't think twice about e-mailing or @'ing me. I'd love to help you be more secure or help ensure I make things that are useful to you.

With that I thank you for your time and have just a few minutes left for more questions.