An Overview of R

Nathan Eastwood

04/03/2019

Let’s Connect!

nathaneastwood

@nathaneastwood_

nathaneastwood

http://nathaneastwood.github.io (http://nathaneastwood.github.io)

About Me

Education

University	Degree	Grade
Plymouth University	BSc Mathematics and Statistics	1^{st} Class (Hons)
Sheffield University	MSc Statistics	Distinction

Employment History

Company	Title	Time
Nuffield Health Foundation	Data Analyst	Summer 2011
C3 Resources	Data Analyst	Jun. 2012 - Jan. 2013
Plymouth University	Data Scientist	Feb. 2013 - Feb. 2016
Mango Solutions	Data Science Consultant	Mar. 2016 - May 2018
iotec Global	Senior Data Scientist	May 2018 - Aug. 2018
Equiniti Data	Senior Data Scientist	Aug. 2018 - Present

Clients Worked With

- Ministry of Defence
- Public Health England
- Office for National Statistics
- HAYS Recruitment
- NATO
- Direct Line Group
- and many more!

Key Skills

Language	Years of Experience
R	9 years
Python	4 years
SQL	7 years
LaTeX	7 years
bash	6 years
git	7 years

My Type of Work

- Software development
- Building data analysis pipelines
- Data analysis
- Machine learning
- Statistical modelling
- Data visualisation

A Brief History of R

The Birth of R

- Before R, there was S
- S was created at Bell Labs in 1976 by John Chambers
- S Plus was developed by TIBCO Software in 1988
- R was created in 1992 by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand
- The first stable release was in 1995
- R is now maintained by the R Development Core Team

CRAN

- The Comprehensive R Archive Network (CRAN) is where you will find the majority (over 13,500) supplementary packages
- You can find additional packages on Bioconductor, Omegahat, GitHub, etc.

R and Other Languages

- R is written with a combination of C and Fortran
- More advanced users can write C, C++, .NET, Java, Python, JavaScript and Go code and call it directly from R

The Birth of Data Science

- R has become increasingly popular. As of February 2019, R ranks 15th in the TIOBE index[1], a measure of popularity of programming languages
- 90% of the world’s data was created in the last two years[2]

[1]: https://www.tiobe.com/tiobe-index/
[2]: https://www.mediapost.com/publications/article/291358/90-of-todays-data-created-in-two-years.html

RStudio

- RStudio are a company which develop the fantastic (and free) R IDE of the same name
- The initial release was in 2011
- RStudio and its team have contributed to many open source packages including
  - Tidyverse – R packages for data science, including `ggplot2`, `dplyr`, `tidyr`, and `purrr`
  - `Shiny` – An interactive web technology
  - `RMarkdown` – Insert R code into markdown documents
  - `knitr` – Dynamic reports combining R, TeX, Markdown & HTML
  - `packrat` – Package dependency tool
  - `devtools` – Package development tool

What Can R Do?

Interpreted Language

```
2 + 2
```

```
[1] 4
```

Data Manipulation

readr

```
qof_data <- read_csv(
  here(
    'data',
    'QOF 2017-18- Prevalence, achievements and exceptions at CCG level.csv'
  )
)
```

readr

```
qof_data
```

dplyr

dplyr

- dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges
    - `mutate()` adds new variables that are functions of existing variables
    - `select()` picks variables based on their names.
    - `filter()` picks cases based on their values.
    - `summarise()` reduces multiple values down to a single summary.
    - `arrange()` changes the ordering of the rows.
- These all combine naturally with `group_by()` which allows you to perform any operation “by group”

Renaming Columns

```
qof_data <-
  qof_data %>%
  rename_all(funs(gsub("\\(|\\)|\\+", "", gsub(" |-", "_", tolower(.)))))
qof_data
```

Selecting Columns

```
alt_data <-
  qof_data %>%
  select(
    ccg_name, total_exceptions_2016_2017, total_exceptions_2017_2018,
    patients_receiving_intervention_per_cent
  )
alt_data
```

```
# A tibble: 194 x 4
   ccg_name         total_exceptions_2… total_exceptions_… patients_receiving_i…
   <chr>                          <dbl>              <dbl>                 <dbl>
 1 NHS AIREDALE, W…                 328                342                  89.5
 2 NHS BARNSLEY CCG                 227                241                  91.9
 3 NHS BASSETLAW C…                  71                 58                  90.9
 4 NHS BRADFORD DI…                 519                480                  89.7
 5 NHS CALDERDALE …                 181                177                  90.8
 6 NHS BRADFORD CI…                 110                264                  93.8
 7 NHS DONCASTER C…                 482                572                  91.1
 8 NHS EAST RIDING…                 481                539                  90.4
 9 NHS GREATER HUD…                 494                492                  91.5
10 NHS HAMBLETON, …                 118                165                  91.3
# … with 184 more rows
```

Filtering Columns

```
alt_data <-
  alt_data %>%
  filter(ccg_name %in% c(
     "NHS KERNOW CCG", "NHS NORTHERN, EASTERN AND WESTERN DEVON CCG",
     "NHS SOUTH DEVON AND TORBAY CCG"
    )
  )
alt_data
```

```
# A tibble: 3 x 4
  ccg_name          total_exceptions_2… total_exceptions_… patients_receiving_i…
  <chr>                           <dbl>              <dbl>                 <dbl>
1 NHS KERNOW CCG                   1941               2183                  90.8
2 NHS NORTHERN, EA…                2050               2885                  90.7
3 NHS SOUTH DEVON …                 785                855                  89.7
```

Sorting Columns

```
alt_data <-
  alt_data %>%
  arrange(patients_receiving_intervention_per_cent)
alt_data
```

```
# A tibble: 3 x 4
  ccg_name          total_exceptions_2… total_exceptions_… patients_receiving_i…
  <chr>                           <dbl>              <dbl>                 <dbl>
1 NHS SOUTH DEVON …                 785                855                  89.7
2 NHS NORTHERN, EA…                2050               2885                  90.7
3 NHS KERNOW CCG                   1941               2183                  90.8
```

Mutating Columns

```
alt_data <-
  alt_data %>%
  mutate(
    exception_rate_yoy_change = (total_exceptions_2016_2017 / total_exceptions_2017_2018) * 100
  )
alt_data
```

```
# A tibble: 3 x 5
  ccg_name  total_exceptions… total_exception… patients_receiv… exception_rate_…
  <chr>                 <dbl>            <dbl>            <dbl>            <dbl>
1 NHS SOUT…               785              855             89.7             91.8
2 NHS NORT…              2050             2885             90.7             71.1
3 NHS KERN…              1941             2183             90.8             88.9
```

dplyr Chaining

```
qof_data %>%
  filter(ccg_name %in% c(
    "NHS KERNOW CCG", "NHS NORTHERN, EASTERN AND WESTERN DEVON CCG",
    "NHS SOUTH DEVON AND TORBAY CCG"
  )
) %>%
  select(
    ccg_name, total_exceptions_2016_2017, total_exceptions_2017_2018,
    patients_receiving_intervention_per_cent
  ) %>%
  arrange(patients_receiving_intervention_per_cent) %>%
  mutate(
    exception_rate_yoy_change = (total_exceptions_2016_2017 / total_exceptions_2017_2018) * 100
  )
```

```
# A tibble: 3 x 5
  ccg_name  total_exceptions… total_exception… patients_receiv… exception_rate_…
  <chr>                 <dbl>            <dbl>            <dbl>            <dbl>
1 NHS SOUT…               785              855             89.7             91.8
2 NHS NORT…              2050             2885             90.7             71.1
3 NHS KERN…              1941             2183             90.8             88.9
```

Summarising Data

```
qof_data %>%
  summarise(
    mean_prip = mean(patients_receiving_intervention_per_cent),
    max_prip = max(patients_receiving_intervention_per_cent),
    min_prip = min(patients_receiving_intervention_per_cent)
  ) %>%
  mutate(range_prip = max_prip - min_prip)
```

```
# A tibble: 1 x 4
  mean_prip max_prip min_prip range_prip
      <dbl>    <dbl>    <dbl>      <dbl>
1      90.6     94.0     82.6       11.3
```

Grouping Data

```
qof_data %>%
  group_by(sub_region_name) %>%
  summarise(mean_prip = mean(patients_receiving_intervention_per_cent))
```

```
# A tibble: 16 x 2
   sub_region_name                                        mean_prip
   <chr>                                                      <dbl>
 1 LONDON NORTH EAST AND CENTRAL                               91.6
 2 LONDON NORTH WEST                                           88.0
 3 LONDON SOUTH                                                88.7
 4 NHS ENGLAND CENTRAL MIDLANDS                                91.2
 5 NHS ENGLAND CHESHIRE AND MERSEYSIDE                         91.1
 6 NHS ENGLAND CUMBRIA AND NORTH EAST                          90.8
 7 NHS ENGLAND EAST                                            90.9
 8 NHS ENGLAND GREATER MANCHESTER                              90.9
 9 NHS ENGLAND HAMPSHIRE, ISLE OF WIGHT AND THAMES VALLEY      89.7
10 NHS ENGLAND KENT, SURREY AND SUSSEX                         90.3
# … with 6 more rows
```

Plotting

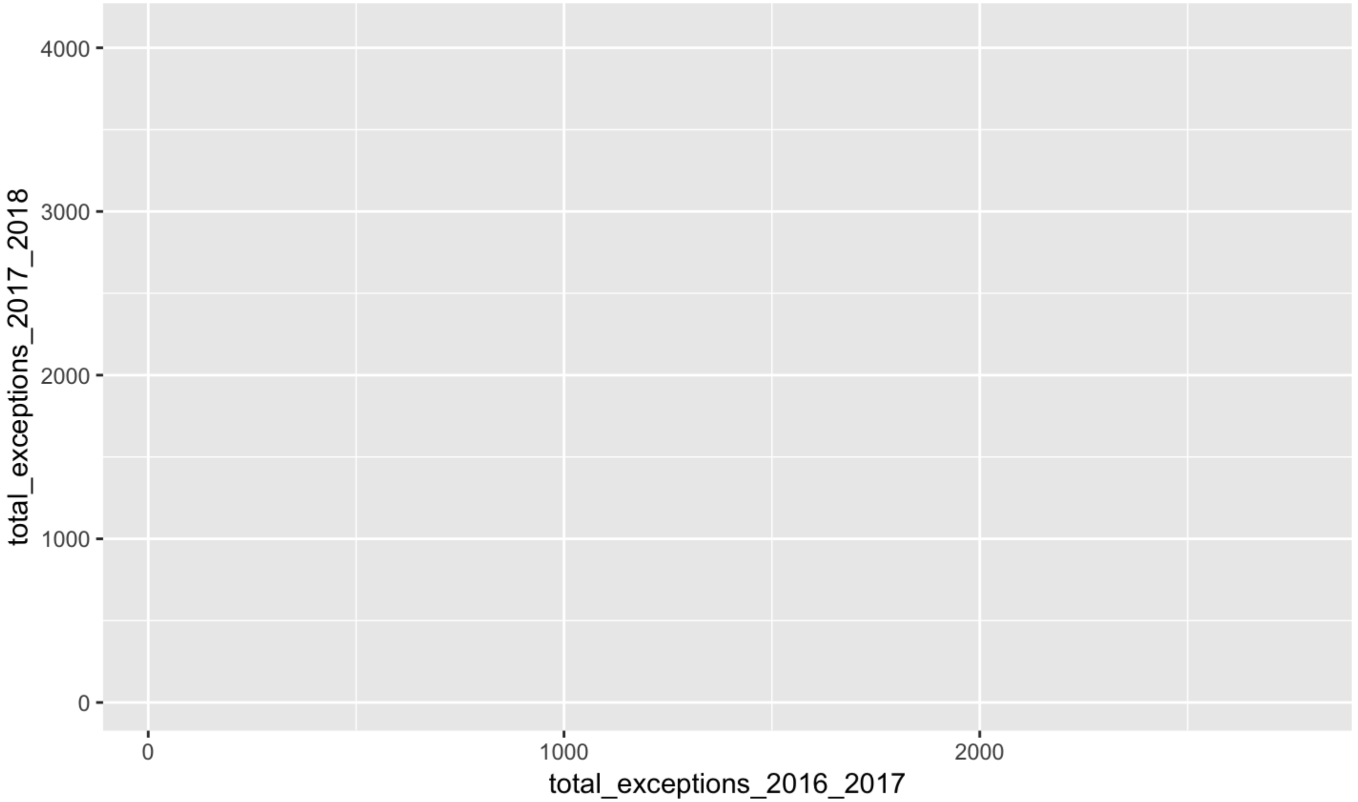
History of Plotting in R

- R comes with the base `graphics` package
- Later the `lattice` package and `grid` package were developed
- `ggplot2` was developed on top of `grid` and is developed on the theory of the "grammar of graphics"

ggplot2

```
p <- ggplot(
  data = qof_data,
  aes(x = total_exceptions_2016_2017, y = total_exceptions_2017_2018)
)
```

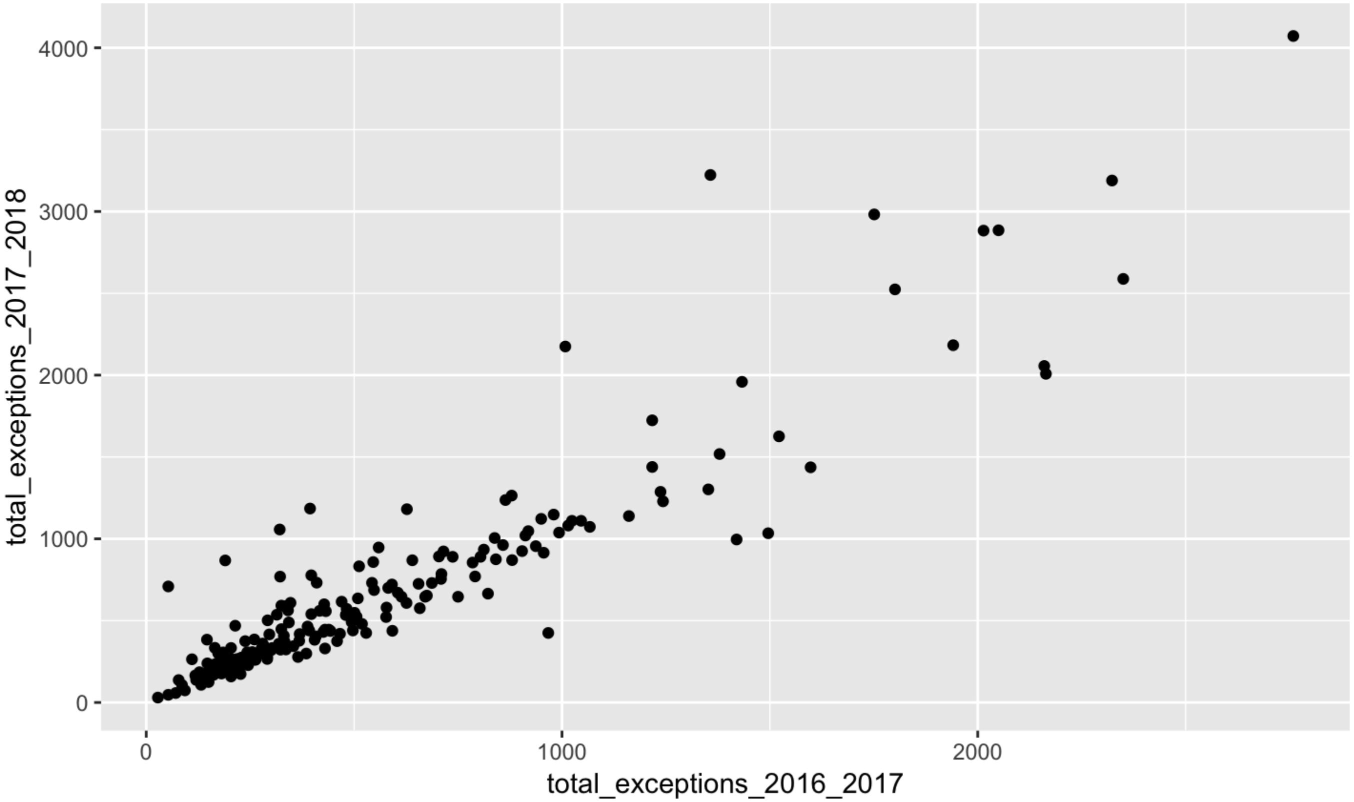
ggplot2


4000
3000
2000
1000
0
total_exceptions_2017_2018
0
1000
2000
total_exceptions_2016_2017

Points

```
p <- p +
  geom_point()
```

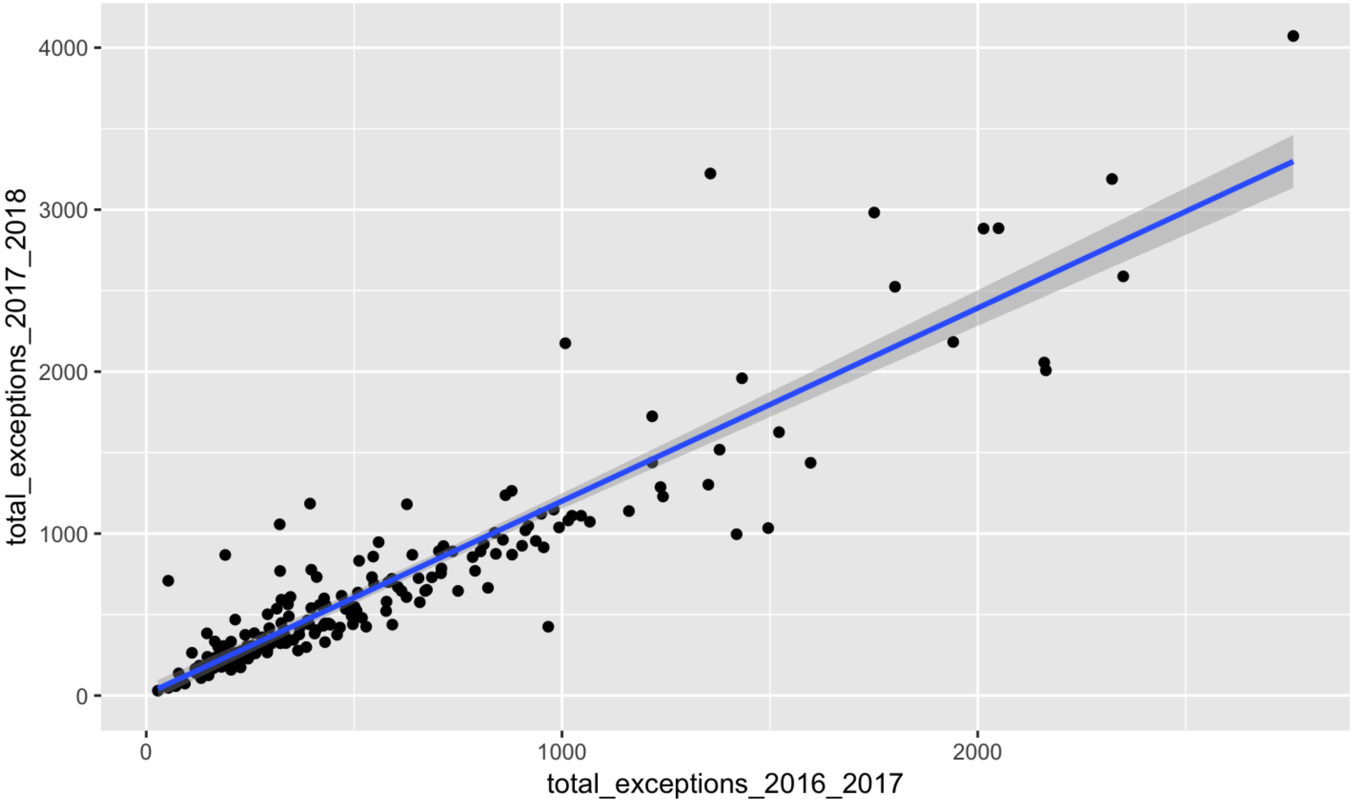
Points


4000
3000
2000
1000
0
total_exceptions_2017_2018
0
1000
2000
total_exceptions_2016_2017

Lines

```
p <- p +
  geom_smooth(method = "lm")
```

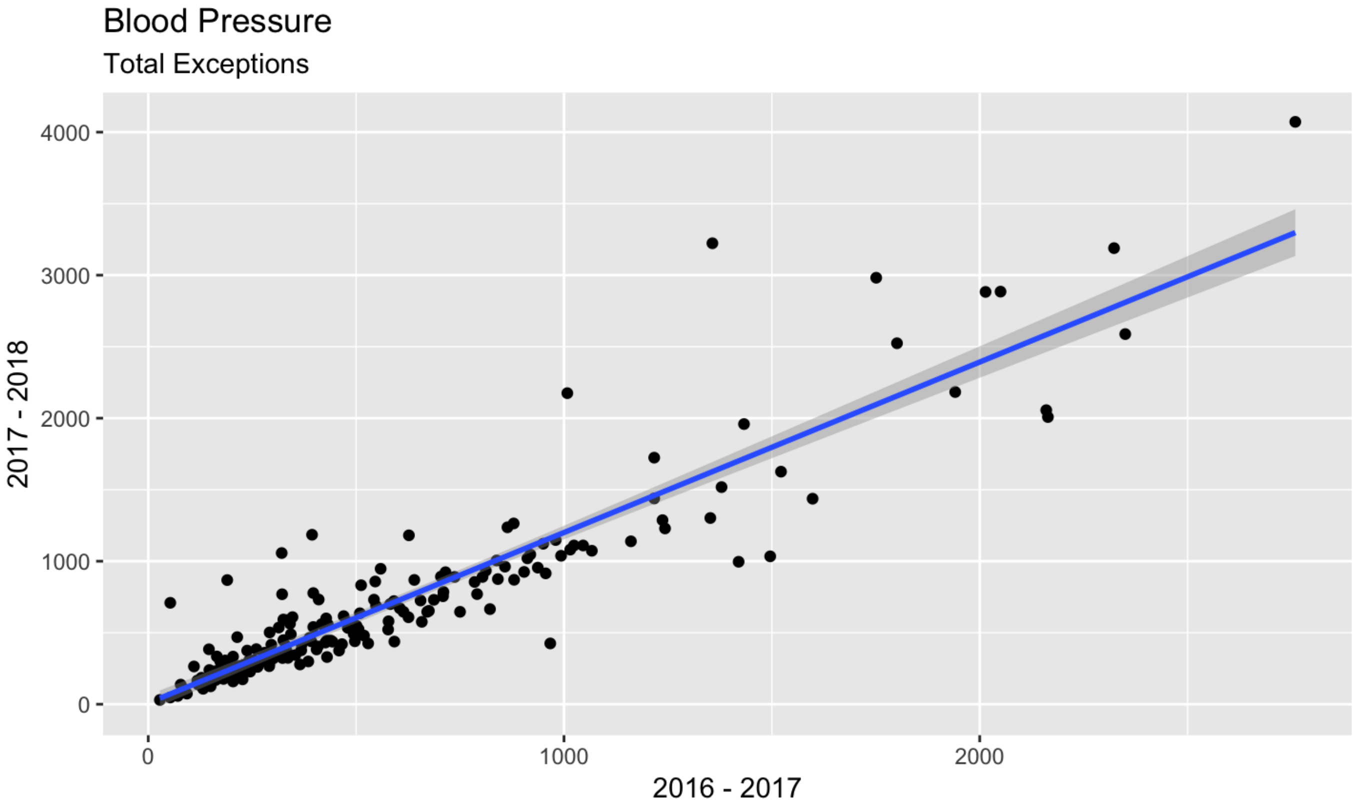
Lines


4000
3000
2000
1000
0
total_exceptions_2017_2018
0
1000
2000
total_exceptions_2016_2017

Labels

```
p <-
  p +
  labs(
    x = "2016 - 2017", y = "2017 - 2018",
    title = "Blood Pressure", subtitle = "Total Exceptions"
  )
```

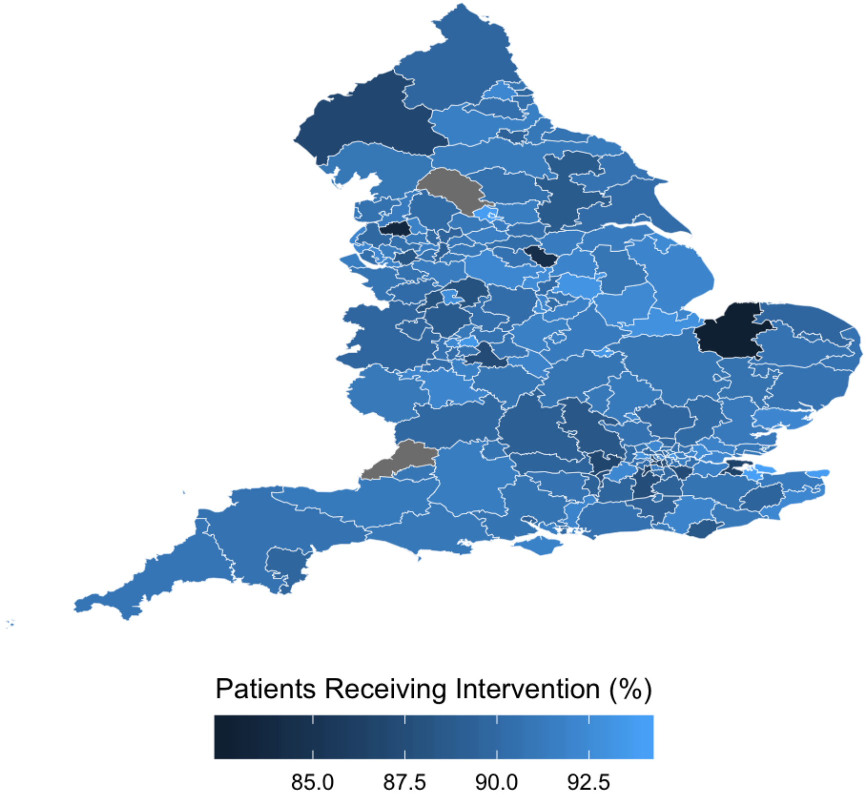
Labels


Blood Pressure
Total Exceptions
4000
3000
2000
1000
0
2017 - 2018
0
1000
2000
2016 - 2017

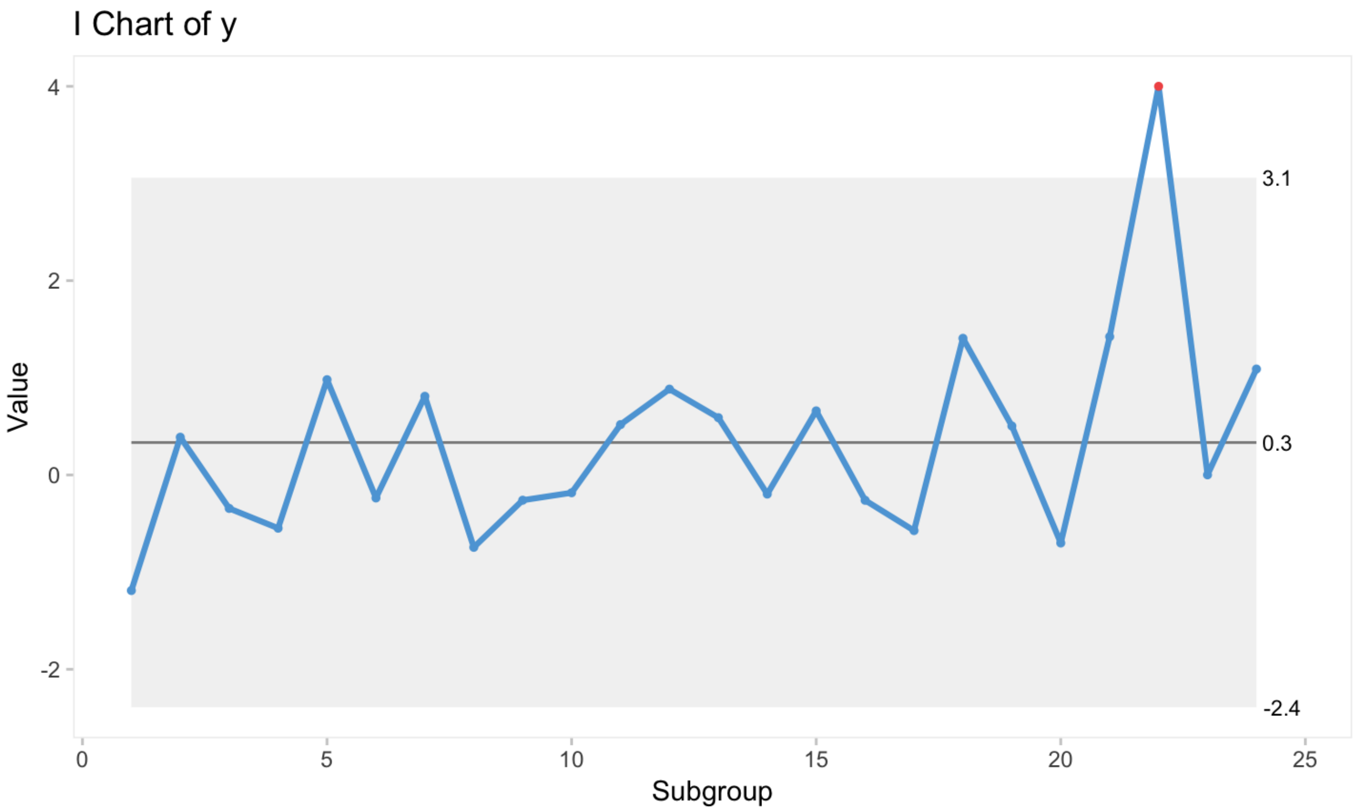
ggplot2 Extensions

- There are many extension packages that have been written for `ggplot2`
- These packages are showcased at https://www.ggplot2-exts.org/ (https://www.ggplot2-exts.org/)

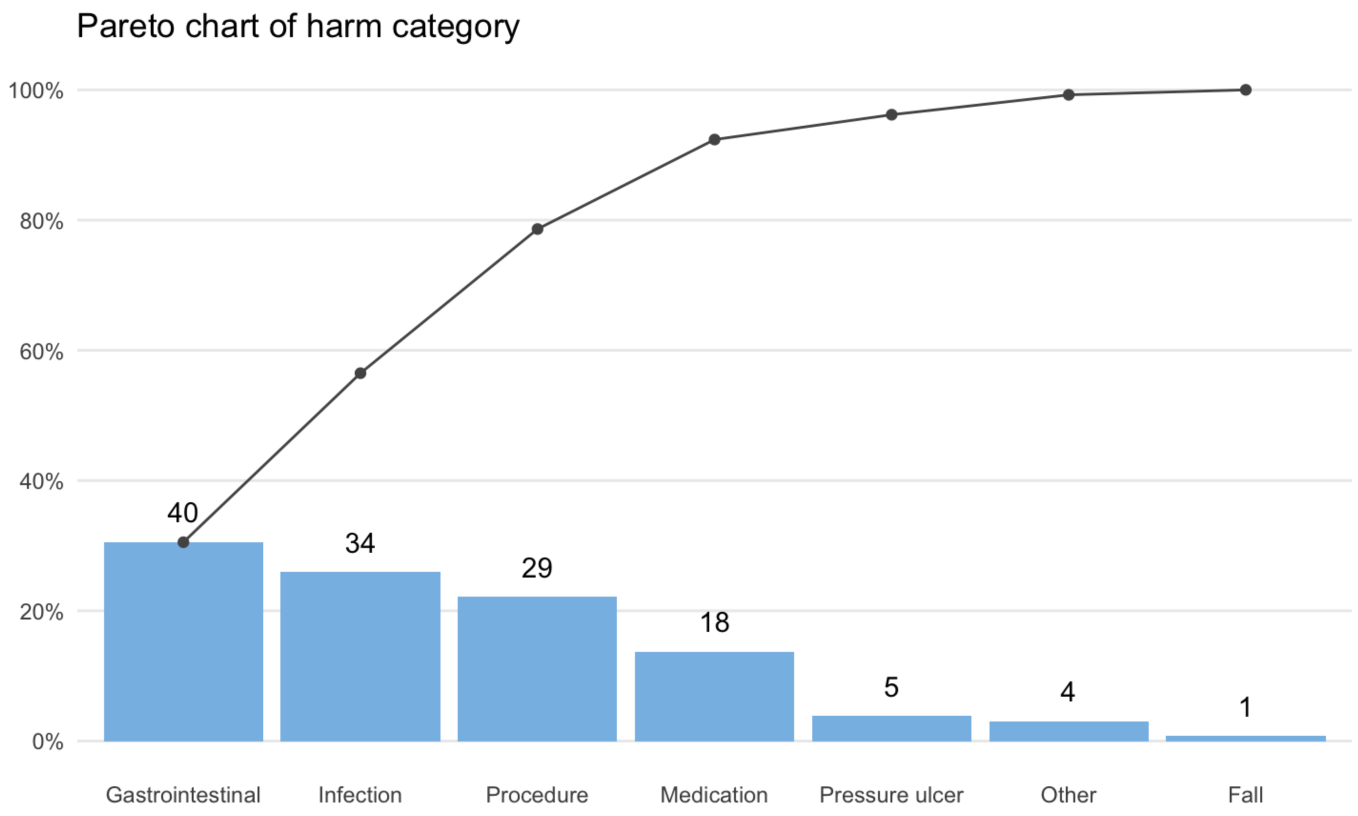
ggplot2 Map



qicharts2



Pareto charts



Statistical Programming

Statistical Programming

- R is first and foremost designed as a statistical programming language
- It can do all sorts of statistical analyses, from t-tests to machine learning
- There are hundreds of advanced, production quality packages for statistical analysis such as `caret` and `mlr`

Linear Regression Example

```
lm1 <- lm(total_exceptions_2016_2017 ~ total_exceptions_2017_2018, data = qof_data)
summary(lm1)
```

So R Is Just a Statistical Programming Language?

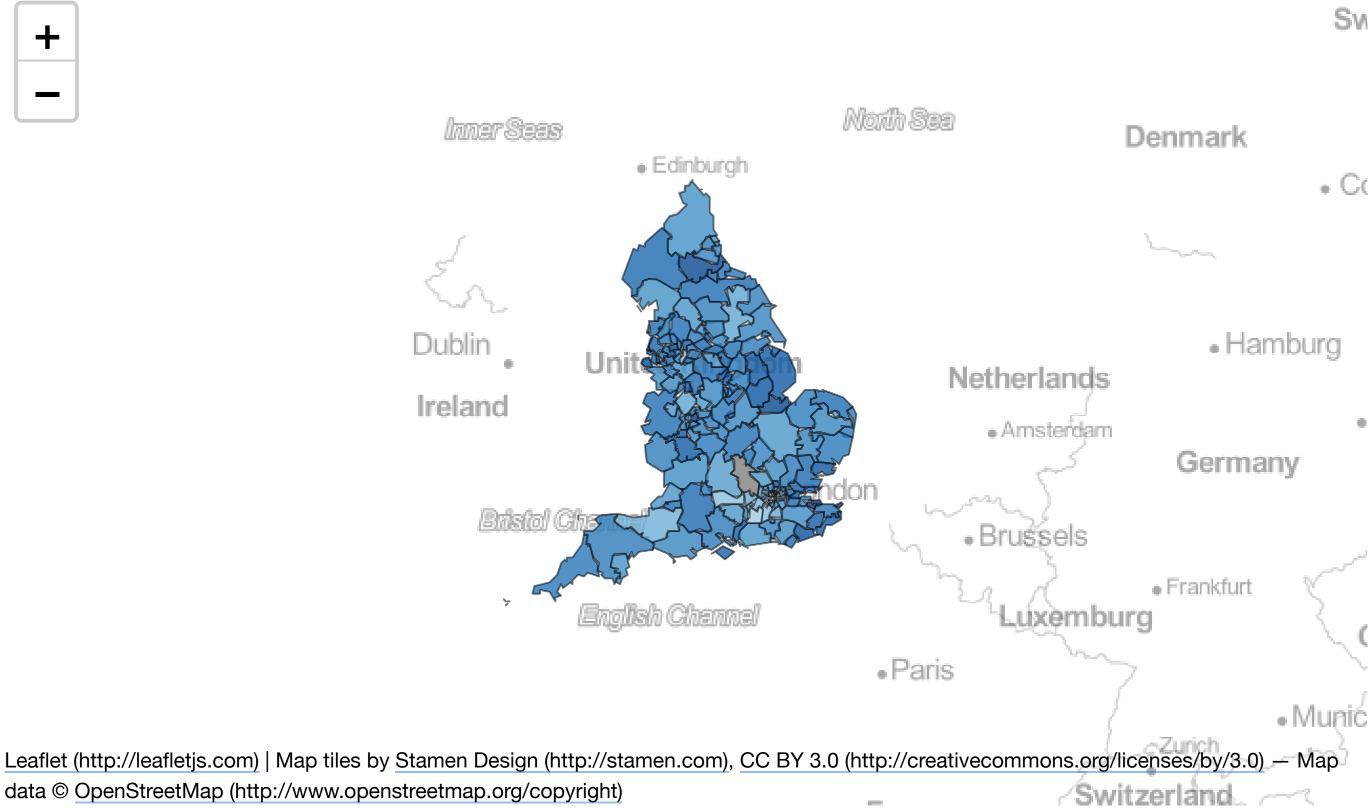
Not Quite!

- R can…
  - Make interactive graphics
  - Make web applications
  - Produce professional, word, pdf, excel, etc. documents
  - Communicate with other languages such as Java, C++, Python and Go!

Interactive Plotting

- You can create interactive graphics using `htmlwidgets`
- `htmlwidgets` provides a wrapper for JavaScript libraries
- You can see `htmlwidgets` examples at http://www.htmlwidgets.org/showcase_leaflet.html (http://www.htmlwidgets.org/showcase_leaflet.html)

Leaflet example

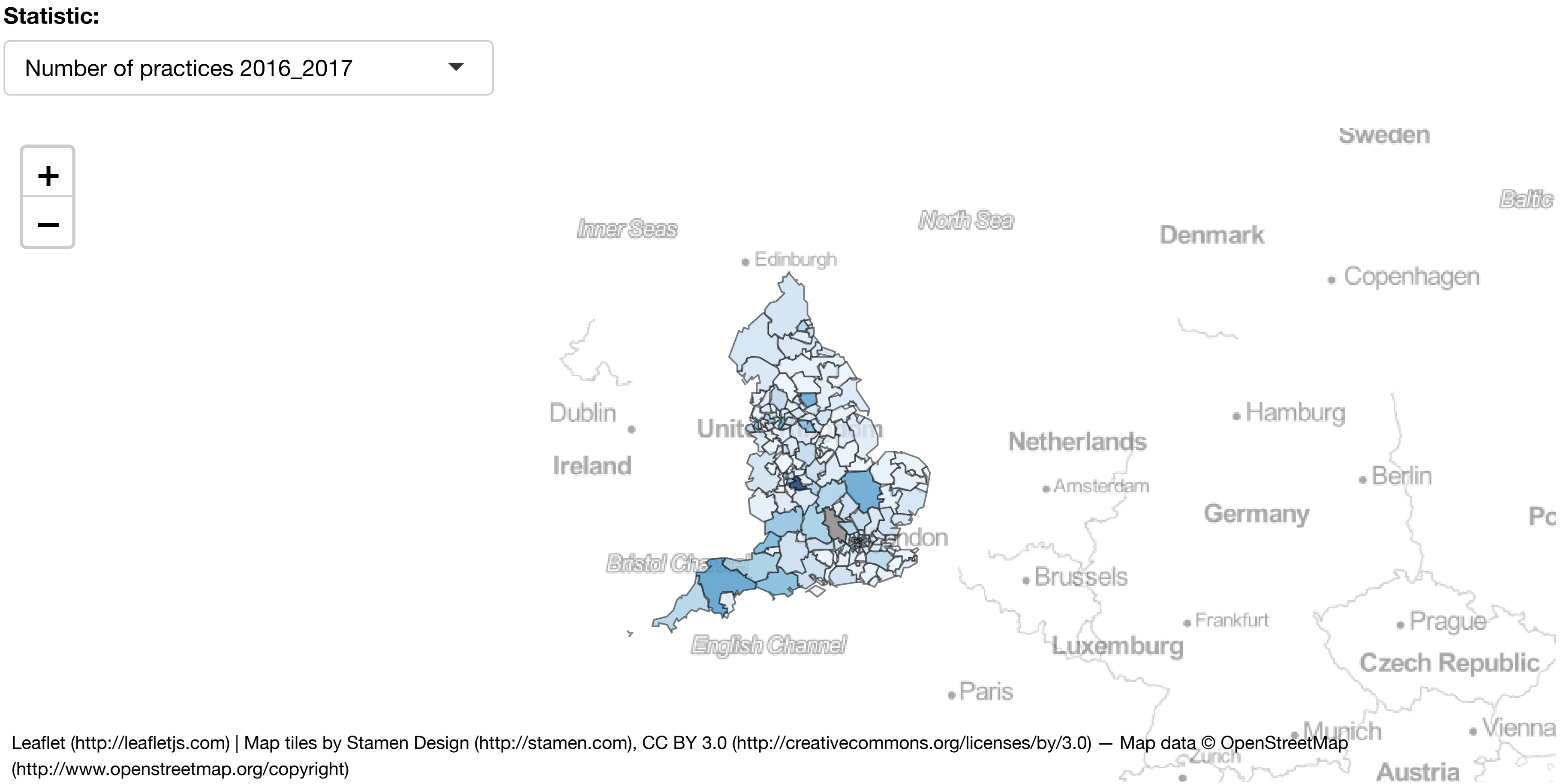


Leaflet (http://leafletjs.com) | Map tiles by Stamen Design (http://stamen.com), CC BY 3.0 (http://creativecommons.org/licenses/by/3.0) — Map data © OpenStreetMap (http://www.openstreetmap.org/copyright)

Application Building

- It is possible to build interactive web applications in R by using `shiny`
- `shiny` wraps HTML, JavaScript and CSS into an R package which you call using only R code
- You can use custom HTML, JavaScript and CSS to extend `shiny`
- There are lots of `shiny` application examples at https://shiny.rstudio.com/gallery/ (https://shiny.rstudio.com/gallery/)

Shiny Example



Statistic:

Number of practices 2016_2017

Leaflet (http://leafletjs.com) | Map tiles by Stamen Design (http://stamen.com), CC BY 3.0 (http://creativecommons.org/licenses/by/3.0) — Map data © OpenStreetMap (http://www.openstreetmap.org/copyright)

Documents

- You can build many documents from R using `Rmarkdown`
- This is an `Rmarkdown` presentation!
- Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents (and more!)
- When you **knit** together an `Rmarkdown` document, a document will be generated that includes both content as well as the output of any embedded R code “chunks” within the document
- For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com)

R can…

- Read in data
- Clean and manipulate data
- Produce complex graphics
- Perform statistical analyses
- Create interactive graphics
- Create websites and web applications
- Produce documents of many formats

Any Questions?

nathaneastwood

@nathaneastwood_

nathaneastwood

http://nathaneastwood.github.io (http://nathaneastwood.github.io)