```
library(dplyr)

rladies_global %>%
  filter(city == 'London')
```

# R-Ladies
# Lightening Talks

# Presentations

- Theatre ticket sales analysis in R- Agnes Salanki
- What can we map- Annabel St John- Lyle
- Hacking antibiotic resistance- Victoria Butt
- Getting started with tidy eval- Nic Crane
- What should I have for lunch- Emma Vestesson

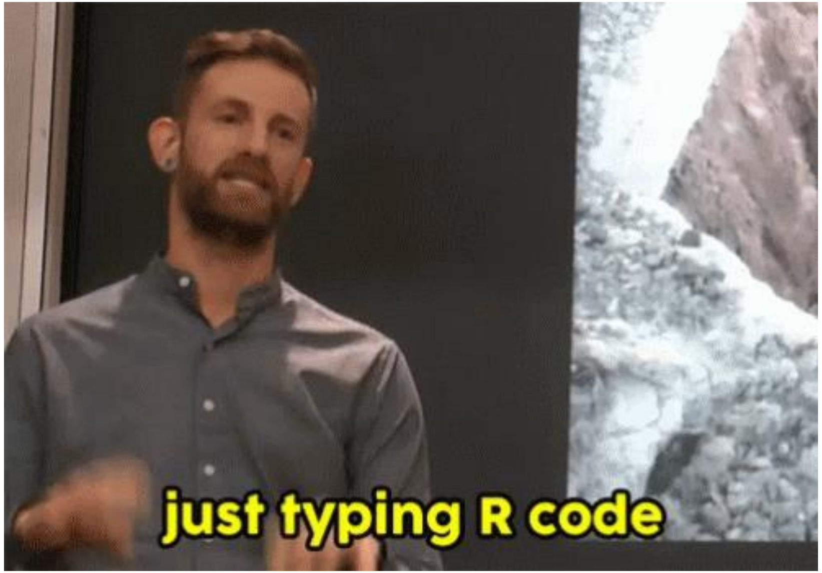
```r
library(dplyr)

rladies_global %>%
  filter(city == ‘London')
```

R-Ladies

# “...all the men and women merely players” – **Theatre ticket sales analysis in R**

# Who am I?

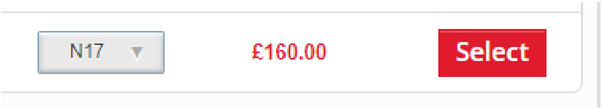
By day: CRM Analyst @ Hotels.com, crunching numbers



By night: theatre enthusiast, (not) going to plays

# My theatre experience in London

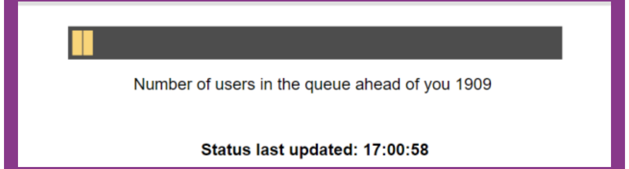
## Either too expensive



N17 ▾

£160.00

## Or too popular



Number of users in the queue ahead of you 1909

**Status last updated: 17:00:58**

## Or too hard to understand

**Workaround: go to a play every time you are in Budapest!**

**Task: to predict which are the plays which will not be sold too quickly?**

# Approach

1. Scrape the website periodically to figure out which play is sold out when
2. Collect information about the plays to have some features to work with
3. Build a simple model on April + validate it on June

# 1 Scrape the website periodically

```
library(rvest)
```

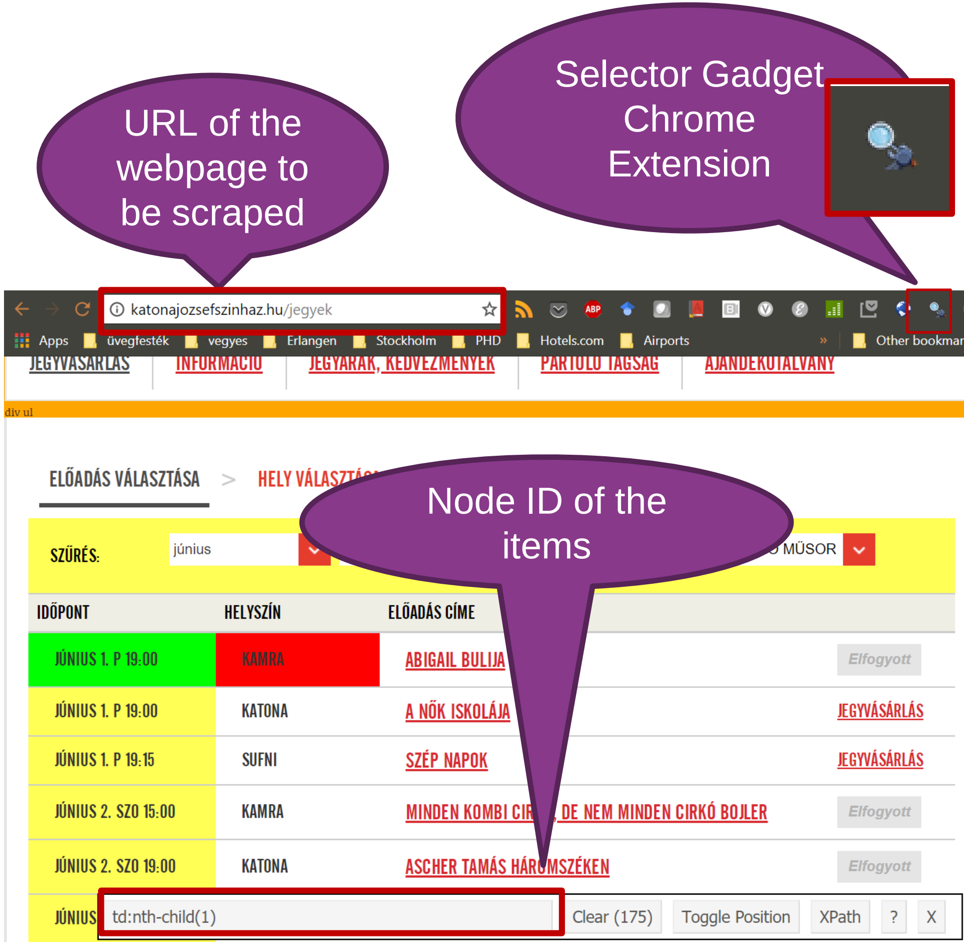
✓ Load the `rvest` package

✓ Define the root URL

```
url_root <-
    "http://katonajozsefszinhaz.hu/jeg
    yek"
webpage_root <- read_html(url_root)

for(i in 1:length(plays_titles)) {
```

✓ Extract a particular field

```
current_date <-
    html_text(html_node(webpage_root,

    paste0('tr:nth-child(', i,')
    td')))

  current_location <-
    html_text(html_node(webpage_root,

    paste0('tr:nth-child(', i, ')
    td:nth-child(2)')))
  …
```



URL of the webpage to be scraped

Selector Gadget Chrome Extension

Node ID of the items

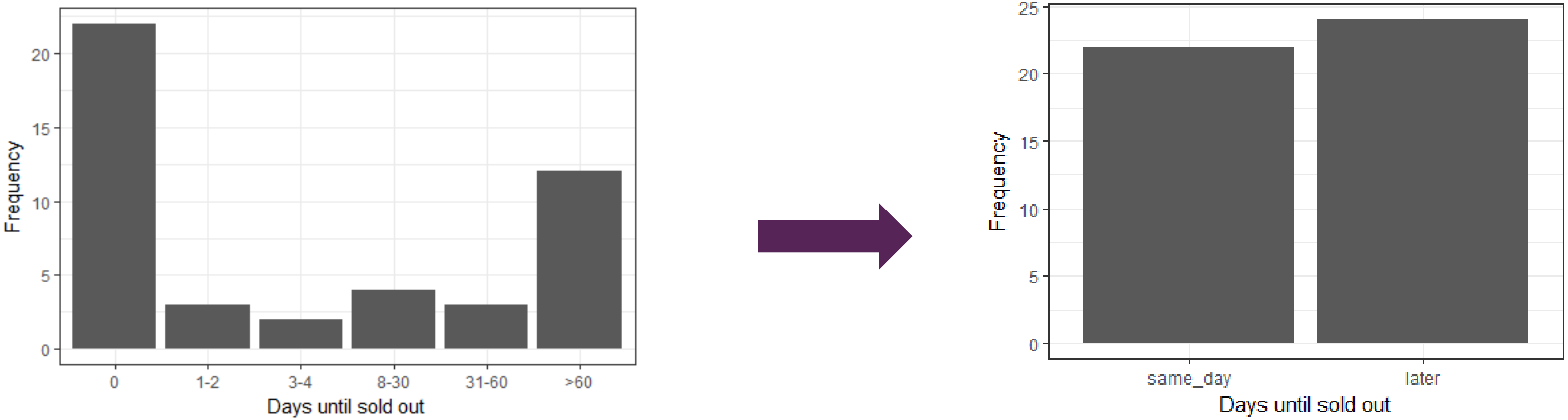
katonajozsefszinhaz.hu/jegyek

| IDŐPONT | HELYSZÍN | ELŐADÁS CÍME | |
|---|---|---|---|
| JÚNIUS 1. P 19:00 | KAMRA | ABIGAIL BULIJA | Elfogyott |
| JÚNIUS 1. P 19:00 | KATONA | A NŐK ISKOLÁJA | JEGYVÁSÁRLÁS |
| JÚNIUS 1. P 19:15 | SUFNI | SZÉP NAPOK | JEGYVÁSÁRLÁS |
| JÚNIUS 2. SZO 15:00 | KAMRA | MINDEN KOMBI CIR , DE NEM MINDEN CIRKÓ BOJLER | Elfogyott |
| JÚNIUS 2. SZO 19:00 | KATONA | ASCHER TAMÁS HÁR MSZÉKEN | Elfogyott |
| JÚNIUS | | | |

td:nth-child(1)

# 2 Feature engineering

- ✓ Artists (actors, directors, etc.)
- ✓ Number of events per month
- ✓ Location
- ✓ Price of the cheapest and most expensive ticket
- ✓ **Target: # days until sold out → sold out the same day or later**



# 3 Model building

```
library(h2o)
h2o.init()
```

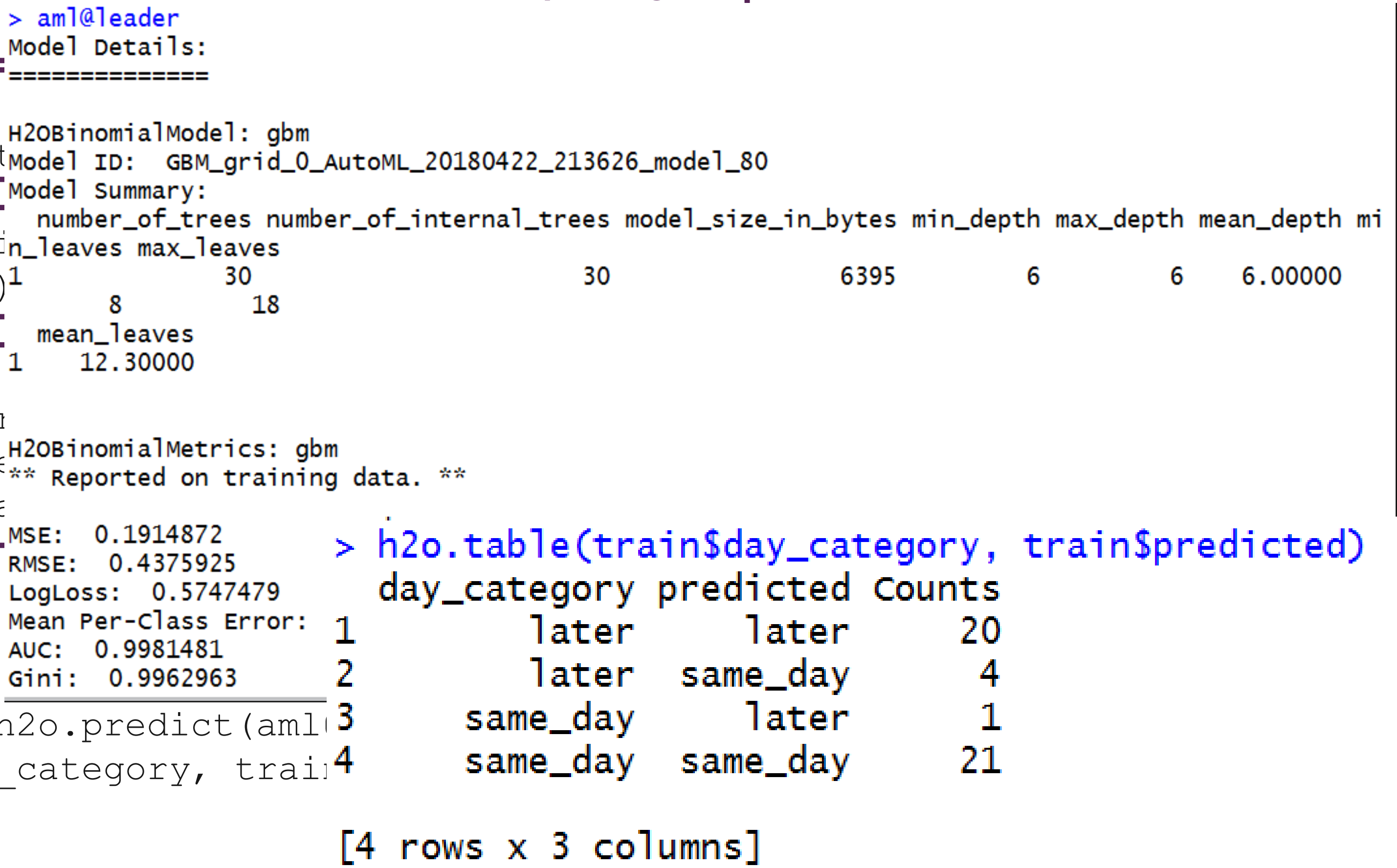
✓ Load the h2o package

```
y <- "day_category"
x <- setdiff(names(t
```

```
train <- as.h2o(trai
test <- as.h2o(test)
```

```
aml <- h2o.automl(x
                  tr
                  le
                  ma
```

```
aml@leaderboard
aml@leader

train$predicted <- h2o.predict(aml
h2o.table(train$day_category, trai
```



```
> aml@leader
Model Details:
==============

H2OBinomialModel: gbm
Model ID:  GBM_grid_0_AutoML_20180422_213626_model_80
Model Summary:
  number_of_trees number_of_internal_trees model_size_in_bytes min_depth max_depth mean_depth mi
n_leaves max_leaves
1              30                       30                6395         6         6    6.00000
       8         18
  mean_leaves
1    12.30000


H2OBinomialMetrics: gbm
** Reported on training data. **

MSE:  0.1914872
RMSE:  0.4375925
LogLoss:  0.5747479
Mean Per-Class Error:
AUC:  0.9981481
Gini:  0.9962963
```

```
> h2o.table(train$day_category, train$predicted)
  day_category predicted Counts
1        later     later     20
2        later  same_day      4
3     same_day     later      1
4     same_day  same_day     21

[4 rows x 3 columns]
```

# Conclusions and acknowledgements

- Location matters (smaller stage with 100 seats gets sold out quicker)
- Number of events matters? (the more frequent is the play, the more quickly it gets sold out) → caused the most false positives in the validation phase
- Only one actor which seems to “sell” the play

- The same play was popular in April, not so much in June

- Blog post on rvest (found on R-Ladies slack ☺): https://www.analyticsvidhya.com/blog/2017/03/beginners-guide-on-web-scraping-in-r-using-rvest-with-hands-on-knowledge/
- H2O AutoML webinar (by our own R-Lady Erin LeDell): https://www.youtube.com/watch?v=j6rqrEYQNdo
- gitHub repo for the code: https://github.com/salankia/Random-R-code-snippets/tree/master/theatre%20modeling

```
library(dplyr)

rladies_global %>%
  filter(city == 'London')
```

# WHAT CAN WE MAP?

# Hi!

## I’m Annabel

and I’m going to show you some cool packages for geospatial analysis.

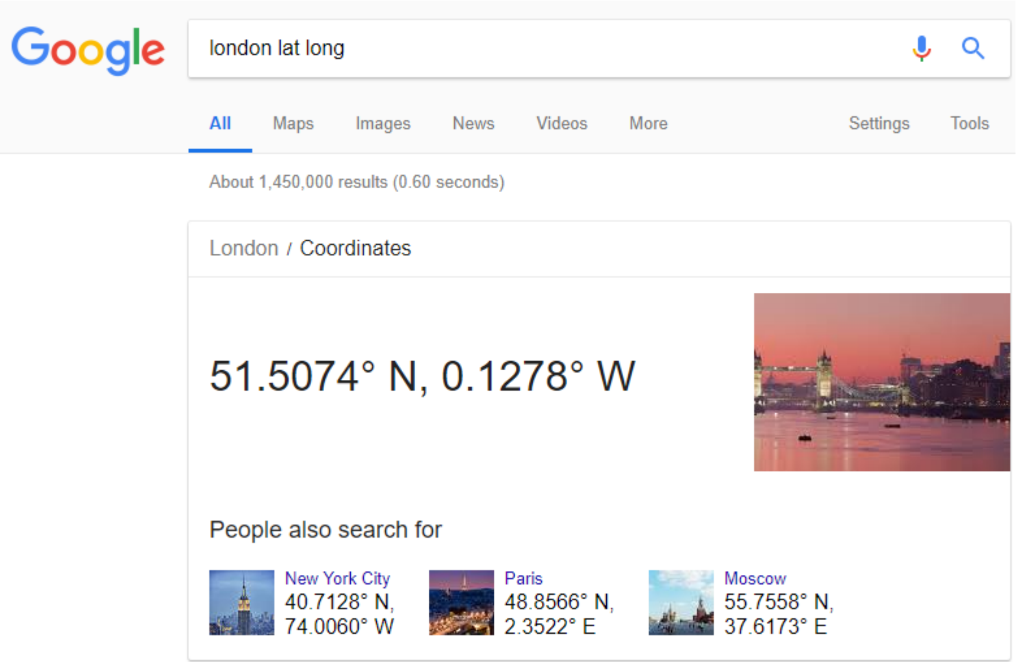
# 1.

# Geospatial Analysis

The Basics

# Lats & Longs

- Every place on earth can be described with a latitude and longitude
- Your data set needs to have one too



london lat long

About 1,450,000 results (0.60 seconds)

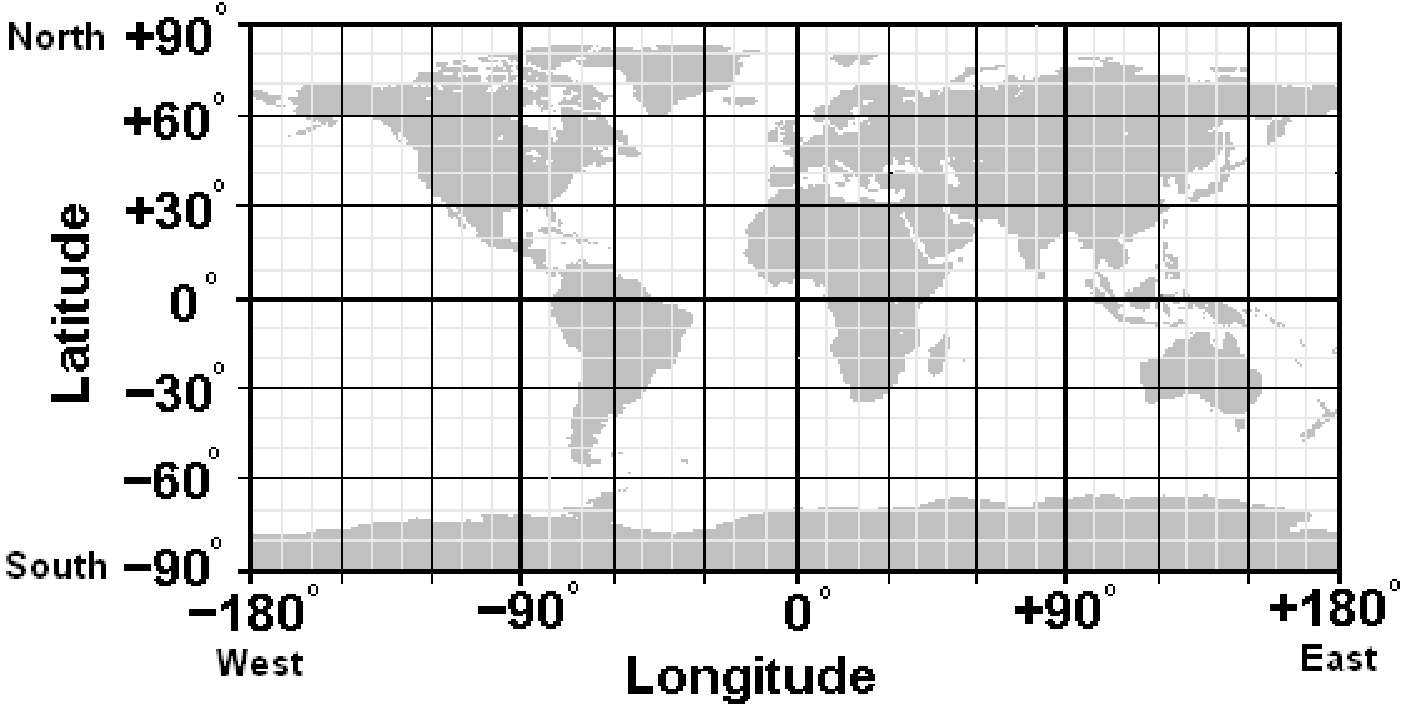
London / Coordinates

51.5074° N, 0.1278° W

People also search for

| New York City | Paris | Moscow |
|---|---|---|
| 40.7128° N, 74.0060° W | 48.8566° N, 2.3522° E | 55.7558° N, 37.6173° E |

# Lats & Longs



www.satsig.net

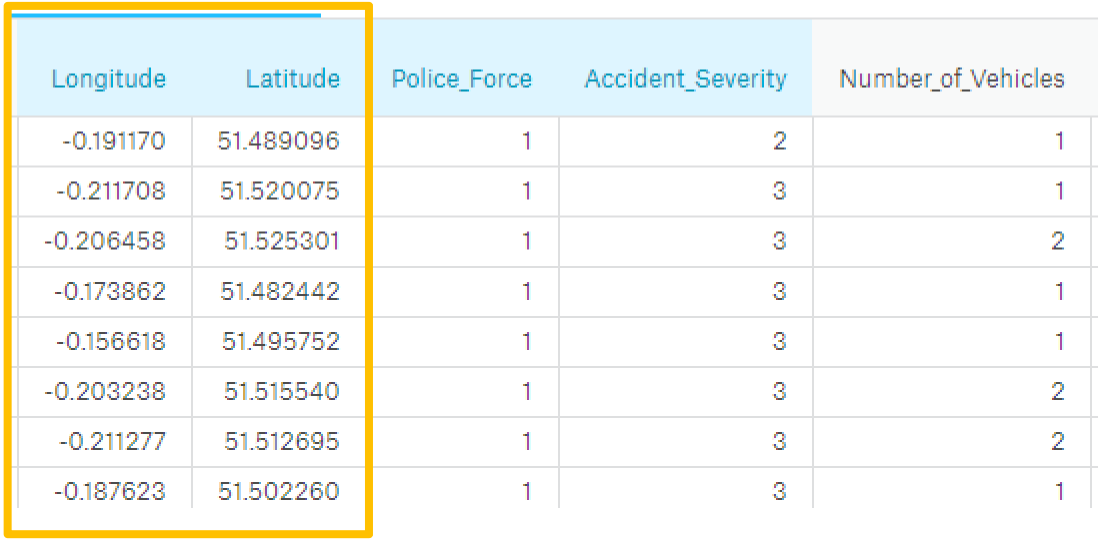
# 2.
# Maps with ggplot and ggmap

# Dataset

- Car Accidents in the UK January 2015

- Published by Department for Transport

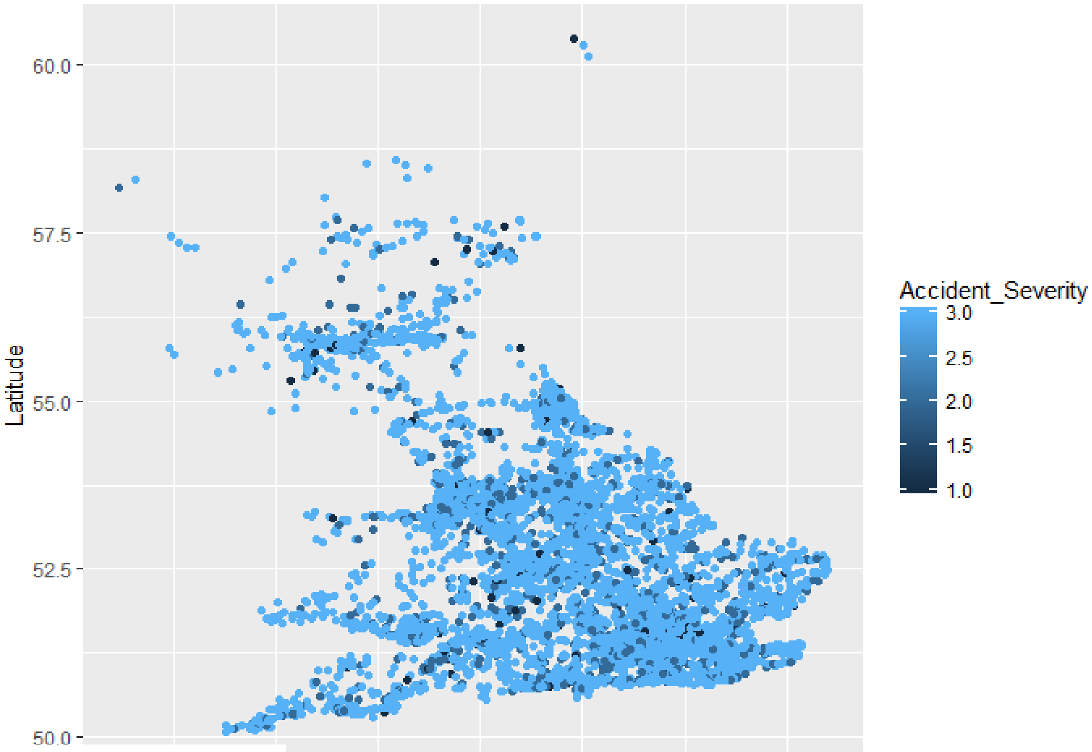
- **Lat & Long** for each accident + 28 features

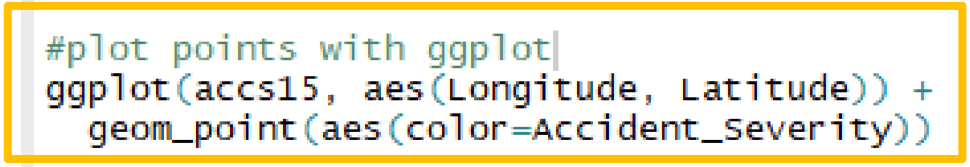


| Longitude | Latitude | Police_Force | Accident_Severity | Number_of_Vehicles |
|---|---|---|---|---|
| -0.191170 | 51.489096 | 1 | 2 | 1 |
| -0.211708 | 51.520075 | 1 | 3 | 1 |
| -0.206458 | 51.525301 | 1 | 3 | 2 |
| -0.173862 | 51.482442 | 1 | 3 | 1 |
| -0.156618 | 51.495752 | 1 | 3 | 1 |
| -0.203238 | 51.515540 | 1 | 3 | 2 |
| -0.211277 | 51.512695 | 1 | 3 | 2 |
| -0.187623 | 51.502260 | 1 | 3 | 1 |

# ggplot

- Plots your data
- Doesn't superimpose it on a map

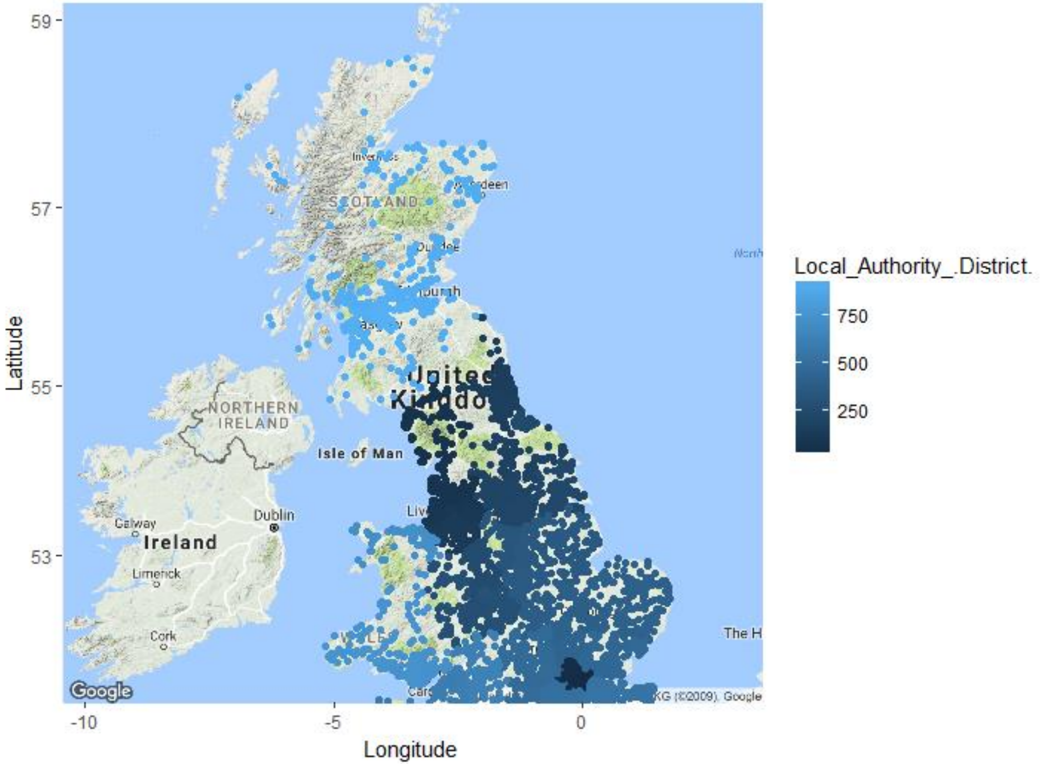


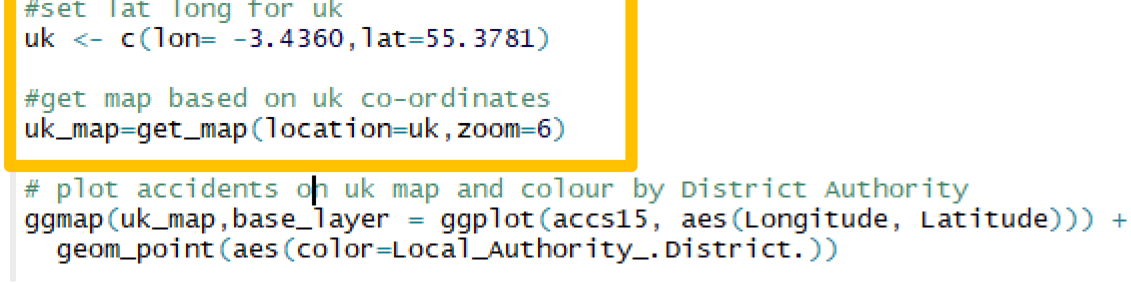


```
#plot points with ggplot
ggplot(accs15, aes(Longitude, Latitude)) +
  geom_point(aes(color=Accident_Severity))
```

# ggmap & ggplot

- ggmap generates a map as the first layer in your visualisation
- Set a co-ordinate
- Set a zoom level
- Get map

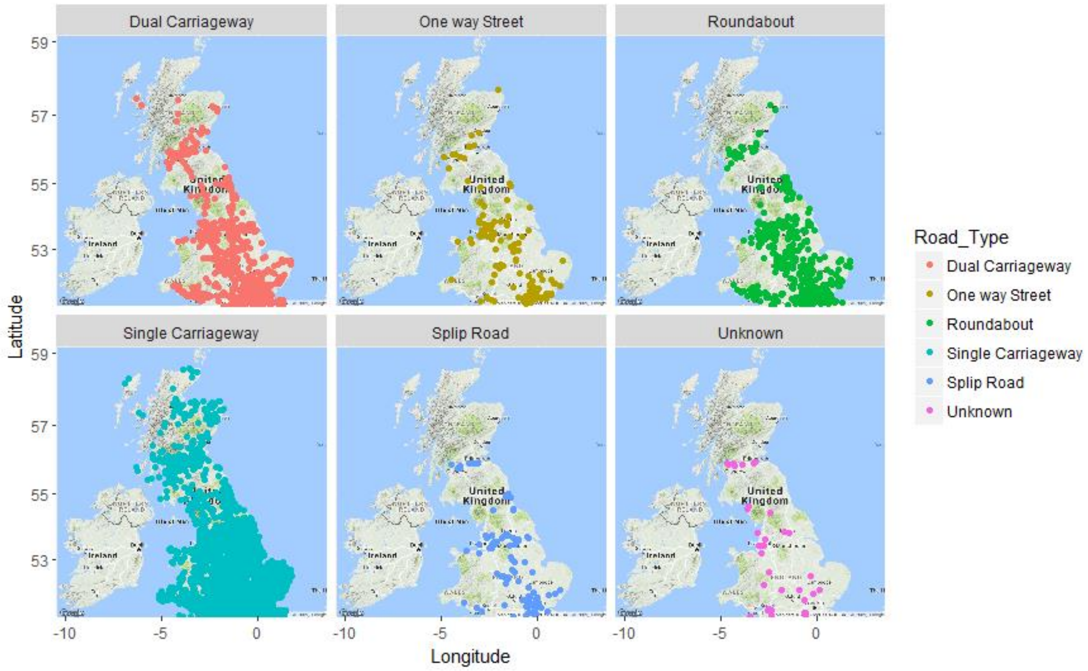




```
#set lat long for uk
uk <- c(lon= -3.4360,lat=55.3781)

#get map based on uk co-ordinates
uk_map=get_map(location=uk,zoom=6)

# plot accidents on uk map and colour by District Authority
ggmap(uk_map,base_layer = ggplot(accs15, aes(Longitude, Latitude)) +
  geom_point(aes(color=Local_Authority_.District.))
```

# Facets!

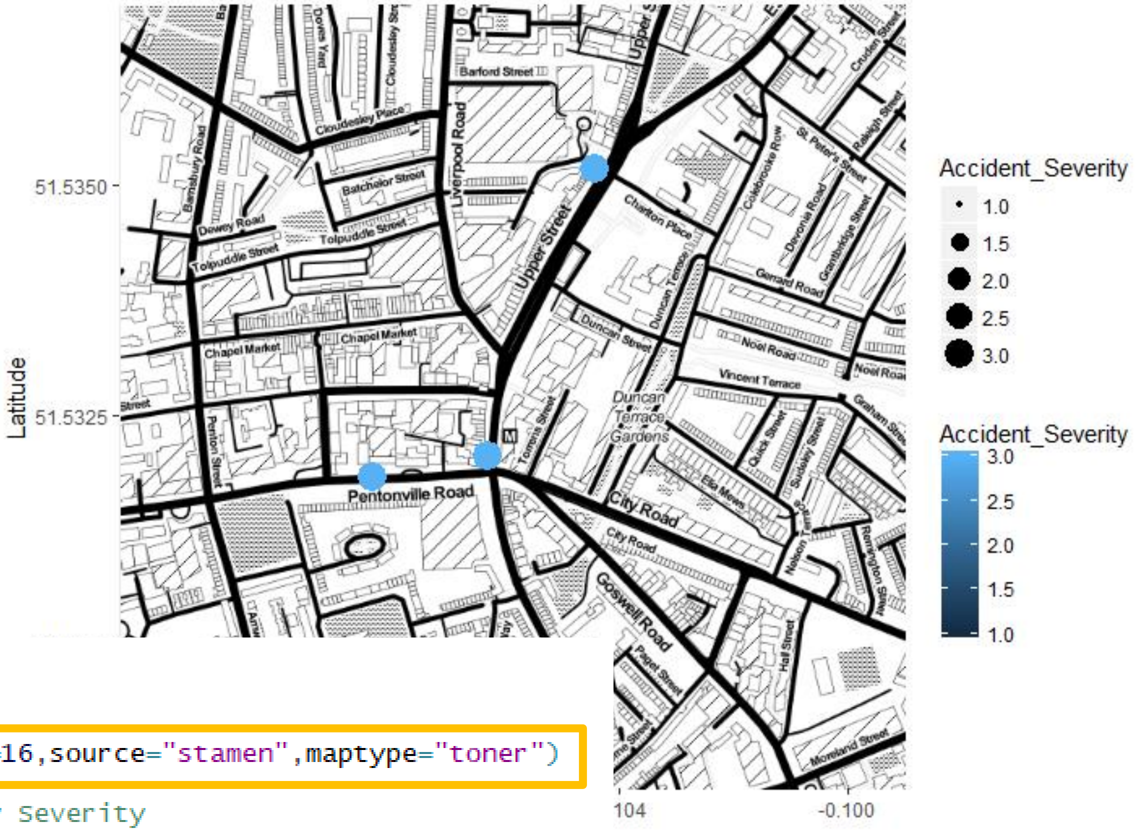
- Facet your maps with one extra line



```r
# plot uk map by road type
ggmap(uk_map,base_layer = ggplot(accs15, aes(Longitude, Latitude))) +
  geom_point(aes(color=Road_Type)) +
  facet_wrap(~ Road_Type)
```

# Settings

- Zoom & change map backgrounds!

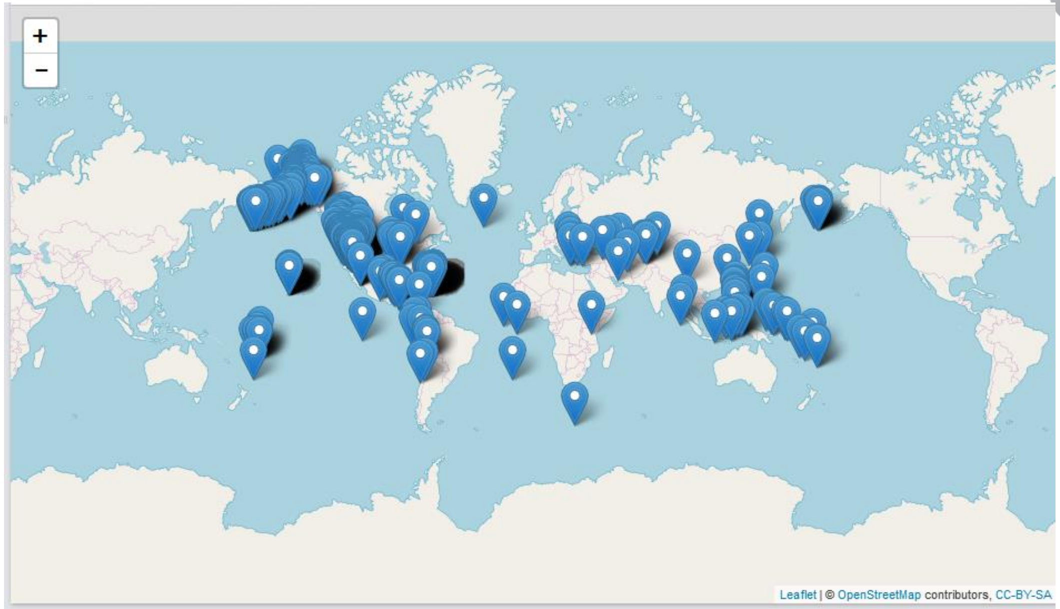


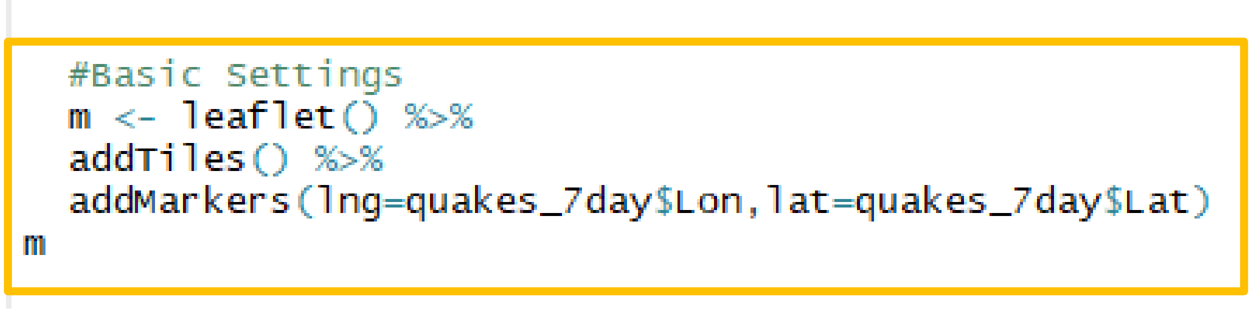
```
#set lat long for london (angel)
london <- c(lon= - 0.1059,lat=51.5327)

london_map=get_map(location=london,zoom=16,source="stamen",maptype="toner")

# plot accidents on uk map and colour by Severity
ggmap(london_map,base_layer = ggplot(accs15, aes(Longitude, Latitude))) +
  geom_point(aes(color=Accident_Severity,size=Accident_Severity))
```

# 2.
# Interactive maps with Leaflet

# Leaflet Maps

- Add Tiles
- Add Markers

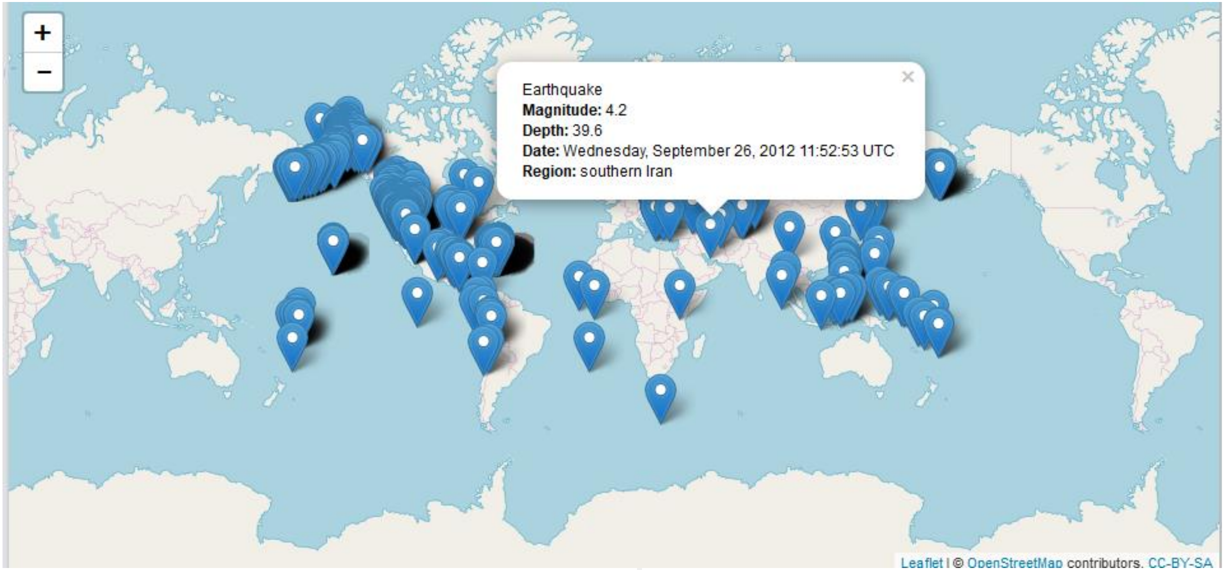




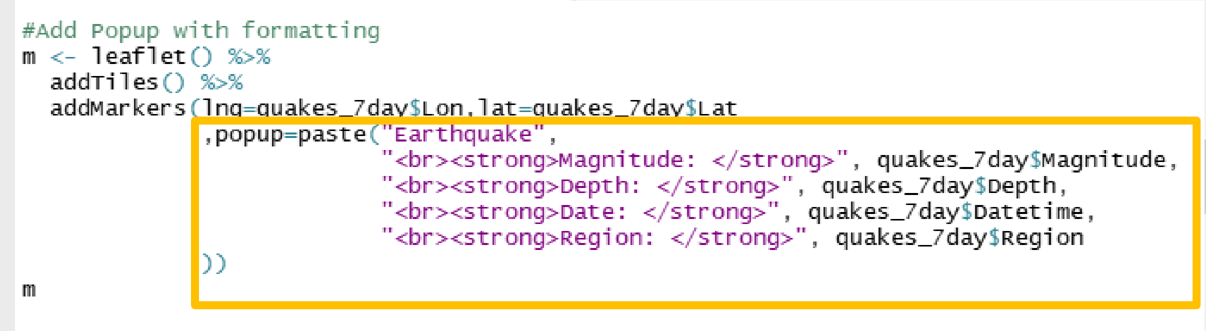
```
  #Basic Settings
  m <- leaflet() %>%
  addTiles() %>%
  addMarkers(lng=quakes_7day$Lon,lat=quakes_7day$Lat)
m
```

# Add Pop-up

- Customise Pop-up text



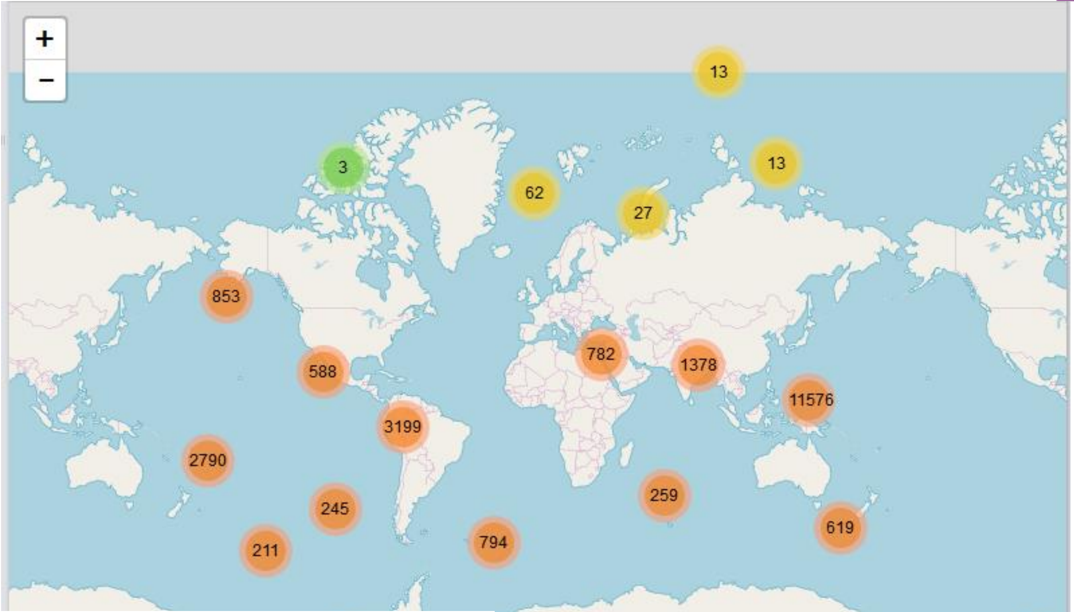
Earthquake
**Magnitude:** 4.2
**Depth:** 39.6
**Date:** Wednesday, September 26, 2012 11:52:53 UTC
**Region:** southern Iran

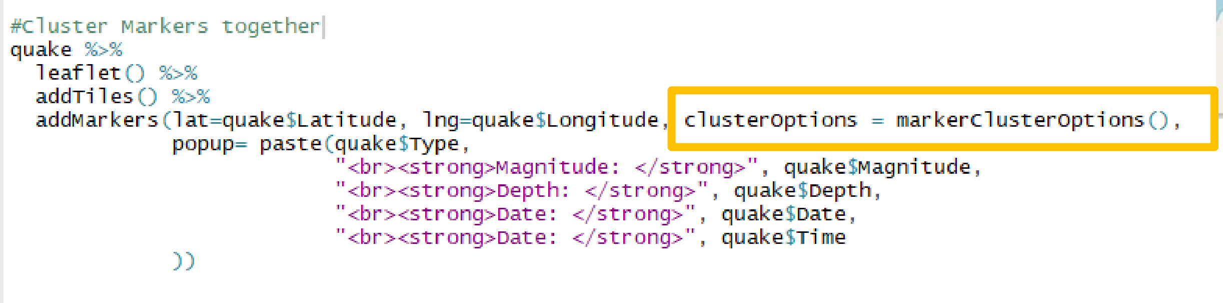


```
#Add Popup with formatting
m <- leaflet() %>%
  addTiles() %>%
  addMarkers(lng=quakes_7day$Lon,lat=quakes_7day$Lat
                ,popup=paste("Earthquake",
                             "<br><strong>Magnitude: </strong>", quakes_7day$Magnitude,
                             "<br><strong>Depth: </strong>", quakes_7day$Depth,
                             "<br><strong>Date: </strong>", quakes_7day$Datetime,
                             "<br><strong>Region: </strong>", quakes_7day$Region
                ))
m
```

# Cluster Markers

- When you have too many data points

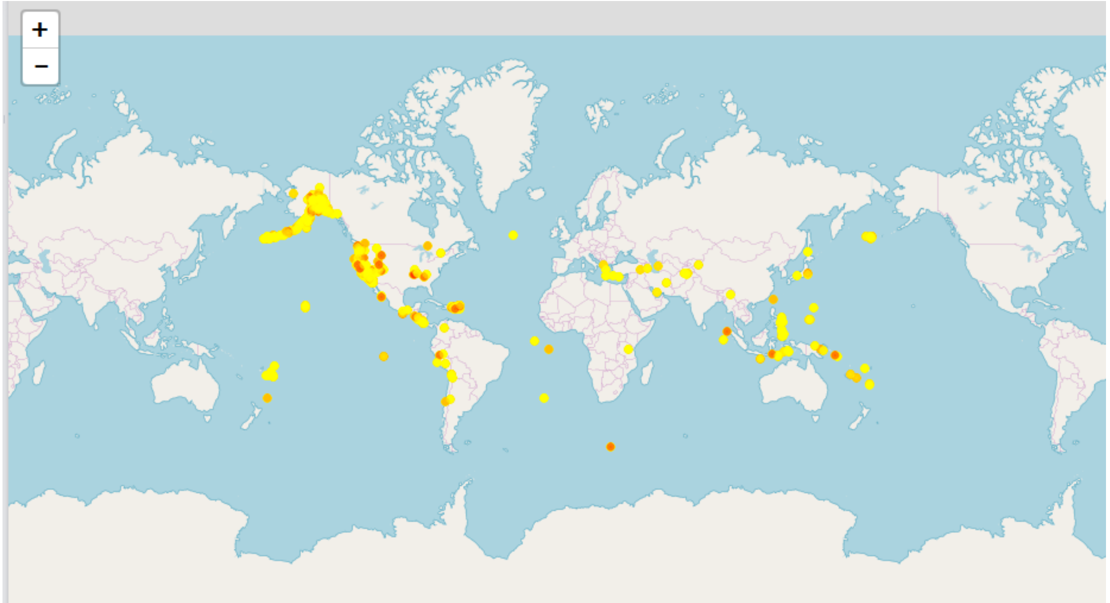


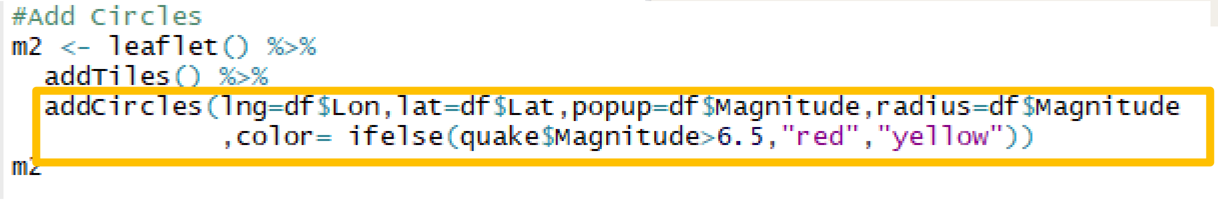


```
#Cluster Markers together
quake %>%
  leaflet() %>%
  addTiles() %>%
  addMarkers(lat=quake$Latitude, lng=quake$Longitude, clusterOptions = markerClusterOptions(),
             popup= paste(quake$Type,
                          "<br><strong>Magnitude: </strong>", quake$Magnitude,
                          "<br><strong>Depth: </strong>", quake$Depth,
                          "<br><strong>Date: </strong>", quake$Date,
                          "<br><strong>Date: </strong>", quake$Time
             ))
```

# Circles & Colors

- Plot circles instead of markers

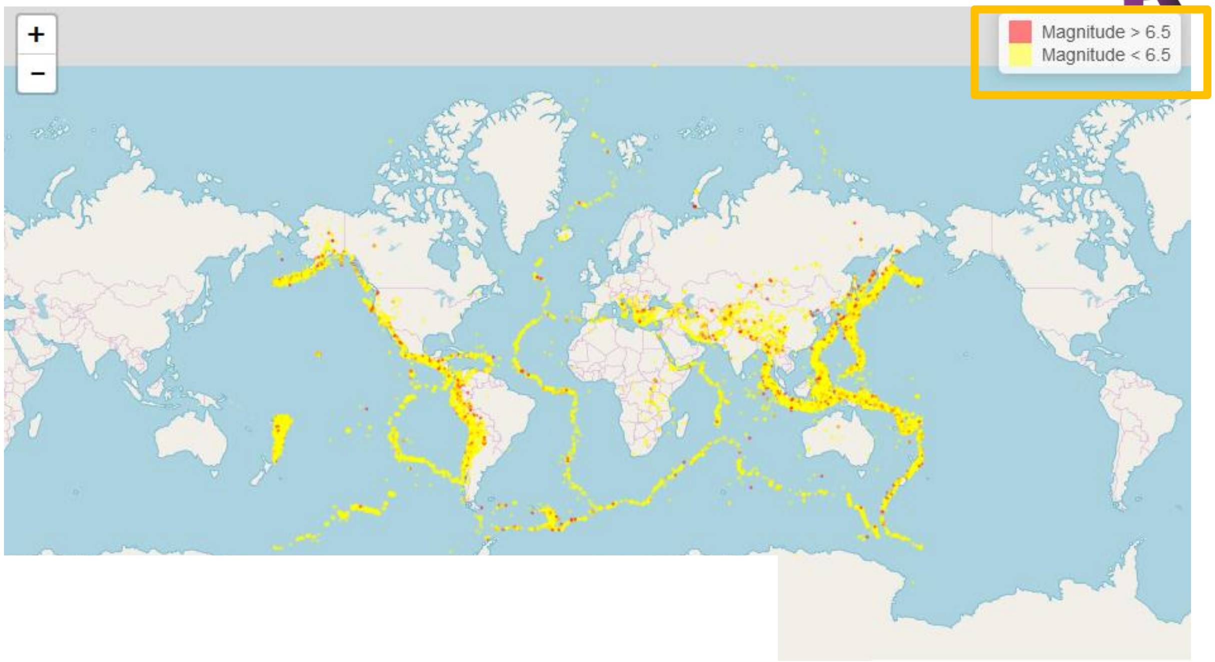




```
#Add Circles
m2 <- leaflet() %>%
  addTiles() %>%
  addCircles(lng=df$Lon,lat=df$Lat,popup=df$Magnitude,radius=df$Magnitude
             ,color= ifelse(quake$Magnitude>6.5,"red","yellow"))
m2
```

# Legend

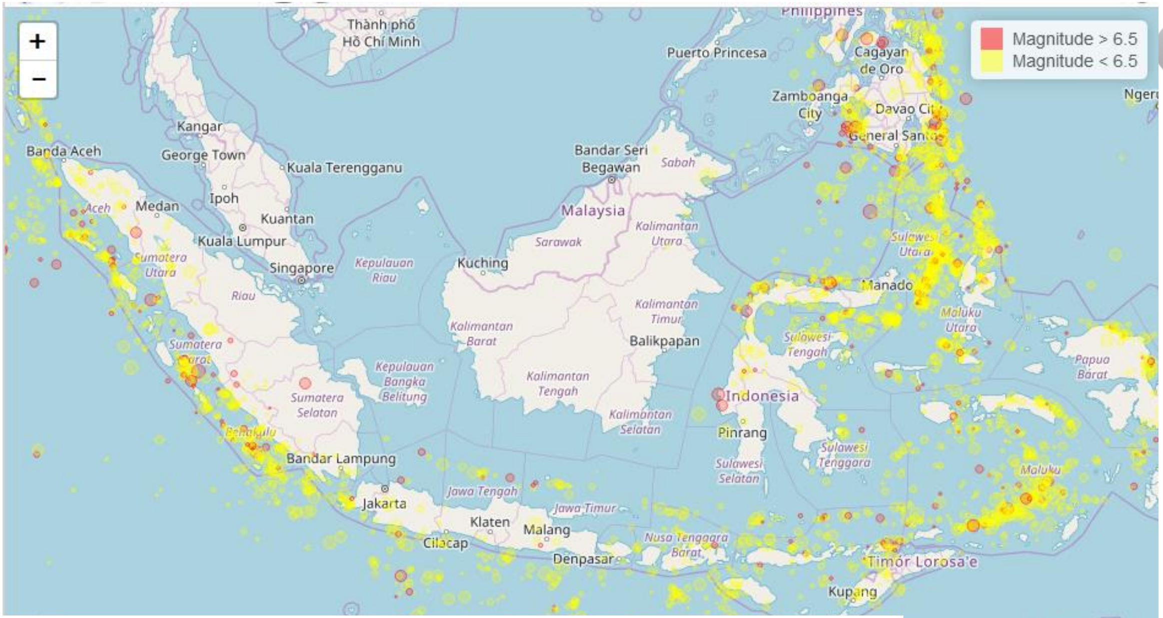
- Add Legend
- Set standard size for points



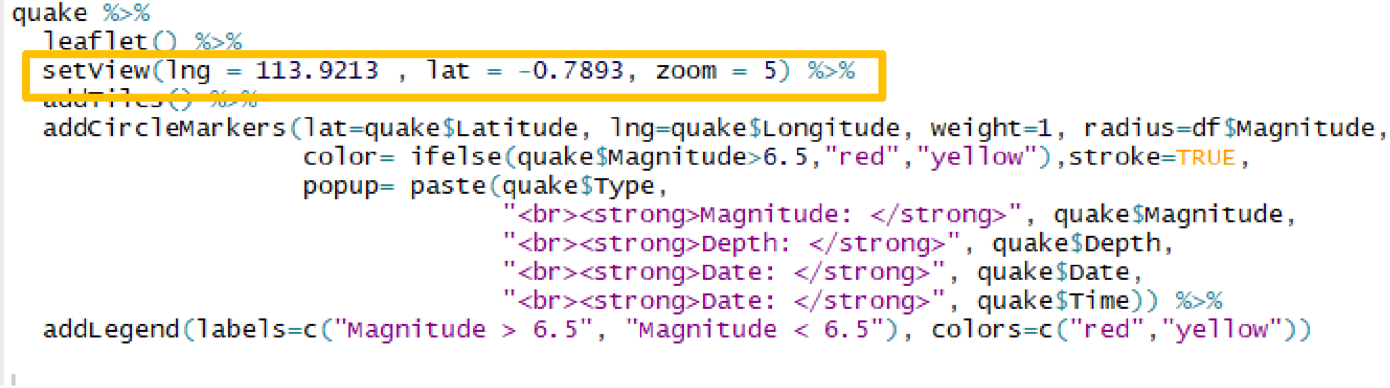
```
#Add Legend and more points
quake %>%
  leaflet() %>%
  addTiles() %>%
  addCircleMarkers(lat=quake$Latitude, lng=quake$Longitude, weight=1, radius=1,
                   color= ifelse(quake$Magnitude>6.5,"red","yellow"),stroke=TRUE,
                   popup= paste(quake$Type,
                                "<br><strong>Magnitude: </strong>", quake$Magnitude,
                                "<br><strong>Depth: </strong>", quake$Depth,
                                "<br><strong>Date: </strong>", quake$Date,
                                "<br><strong>Date: </strong>", quake$Time)) %>%
  addLegend(labels=c("Magnitude > 6.5", "Magnitude < 6.5"), colors=c("red","yellow"))
```

# Set Zoom

- When you want to view a specific region



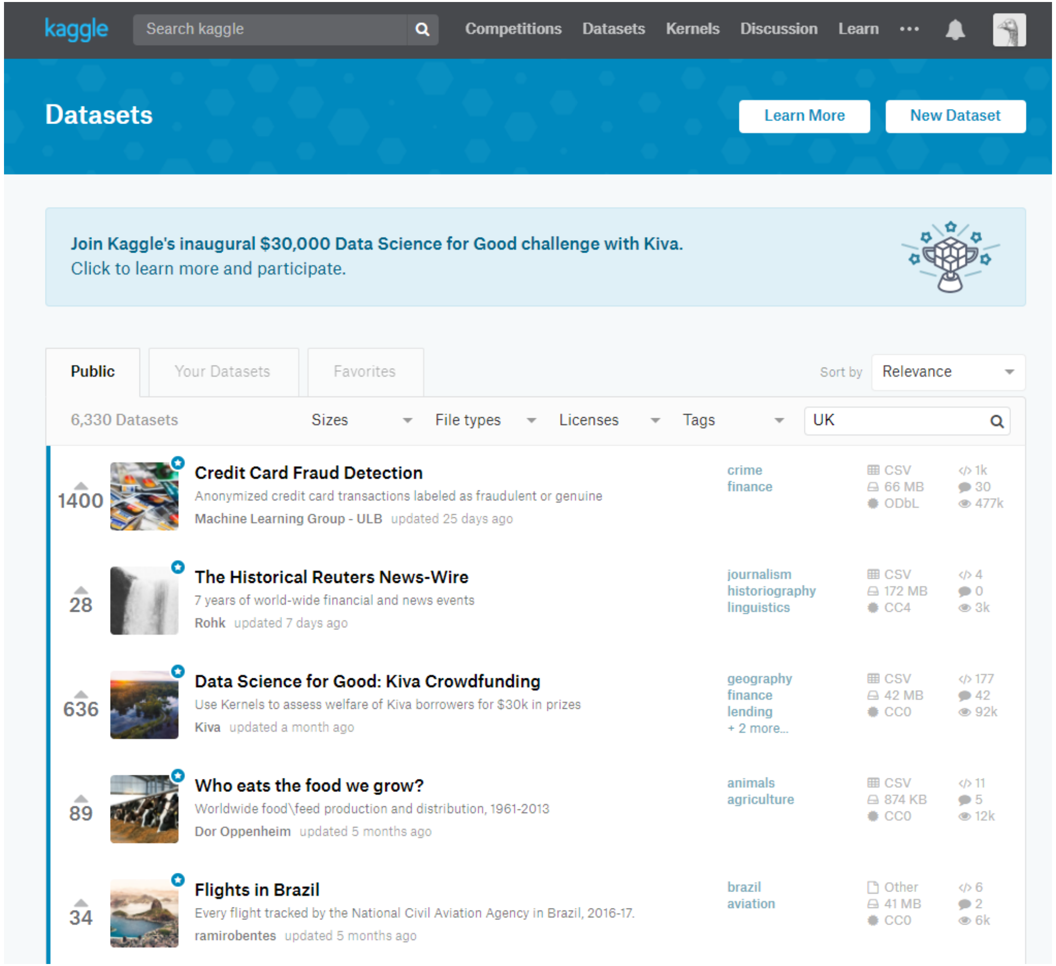
© OpenStreetMap contributors, CC-BY-SA



```
quake %>%
  leaflet() %>%
  setView(lng = 113.9213 , lat = -0.7893, zoom = 5) %>%
  addTiles() %>%
  addCircleMarkers(lat=quake$Latitude, lng=quake$Longitude, weight=1, radius=df$Magnitude,
                   color= ifelse(quake$Magnitude>6.5,"red","yellow"),stroke=TRUE,
                   popup= paste(quake$Type,
                                "<br><strong>Magnitude: </strong>", quake$Magnitude,
                                "<br><strong>Depth: </strong>", quake$Depth,
                                "<br><strong>Date: </strong>", quake$Date,
                                "<br><strong>Date: </strong>", quake$Time)) %>%
  addLegend(labels=c("Magnitude > 6.5", "Magnitude < 6.5"), colors=c("red","yellow"))
```

# Data Sets

- Kaggle.com/datasets is a great resource



kaggle Search kaggle Competitions Datasets Kernels Discussion Learn •••

Datasets

Learn More New Dataset

Join Kaggle's inaugural $30,000 Data Science for Good challenge with Kiva.
Click to learn more and participate.

Public | Your Datasets | Favorites

Sort by Relevance

6,330 Datasets Sizes File types Licenses Tags UK

1400 **Credit Card Fraud Detection**
Anonymized credit card transactions labeled as fraudulent or genuine
Machine Learning Group - ULB updated 25 days ago
crime
finance
CSV 66 MB ODbL
1k 30 477k

28 **The Historical Reuters News-Wire**
7 years of world-wide financial and news events
Rohk updated 7 days ago
journalism
historiography
linguistics
CSV 172 MB CC4
4 0 3k

636 **Data Science for Good: Kiva Crowdfunding**
Use Kernels to assess welfare of Kiva borrowers for $30k in prizes
Kiva updated a month ago
geography
finance
lending
+ 2 more...
CSV 42 MB CC0
177 42 92k

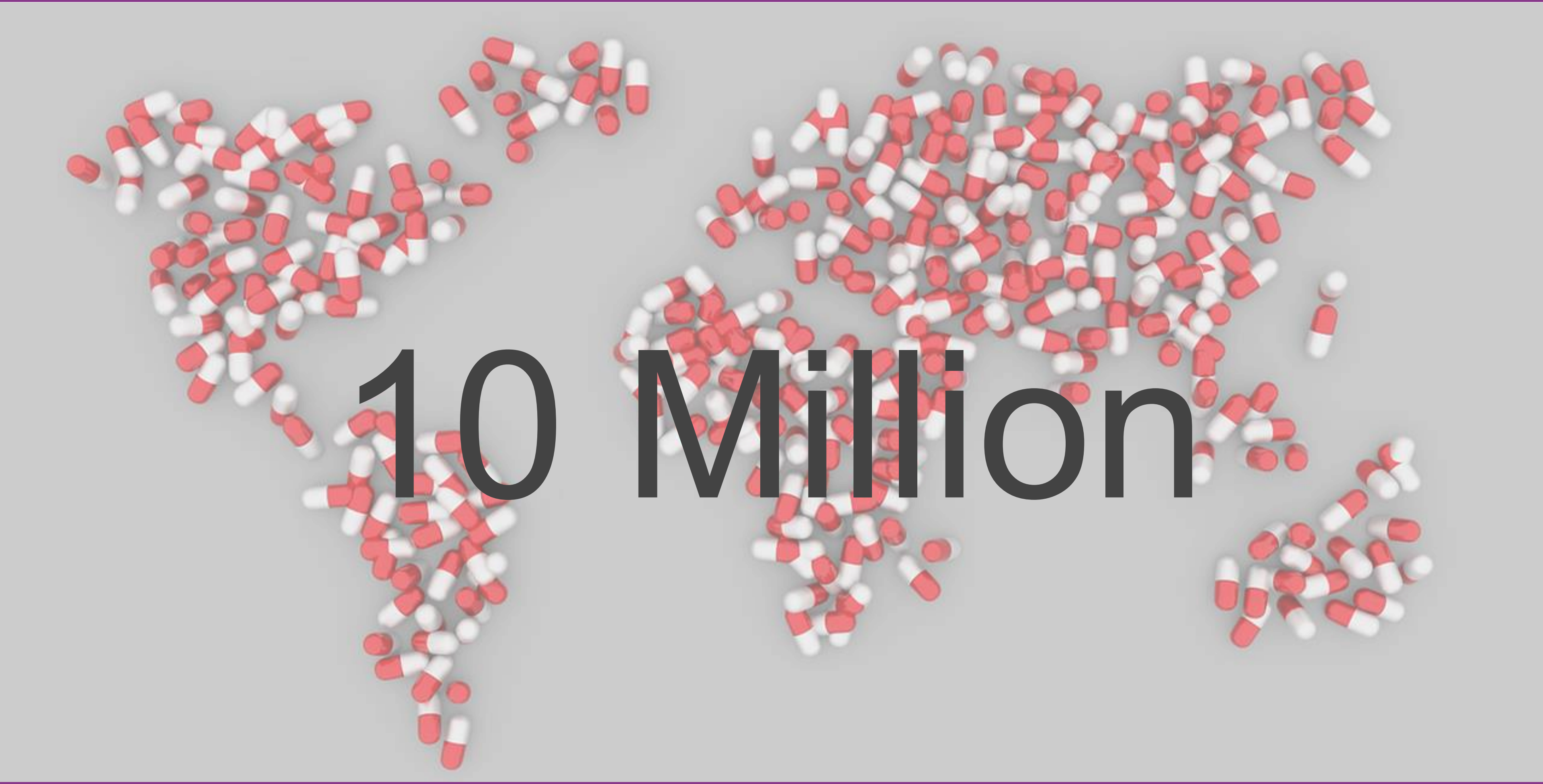
89 **Who eats the food we grow?**
Worldwide food\feed production and distribution, 1961-2013
Dor Oppenheim updated 5 months ago
animals
agriculture
CSV 874 KB CC0
11 5 12k

34 **Flights in Brazil**
Every flight tracked by the National Civil Aviation Agency in Brazil, 2016-17.
ramirobentes updated 5 months ago
brazil
aviation
Other 41 MB CC0
6 2 6k

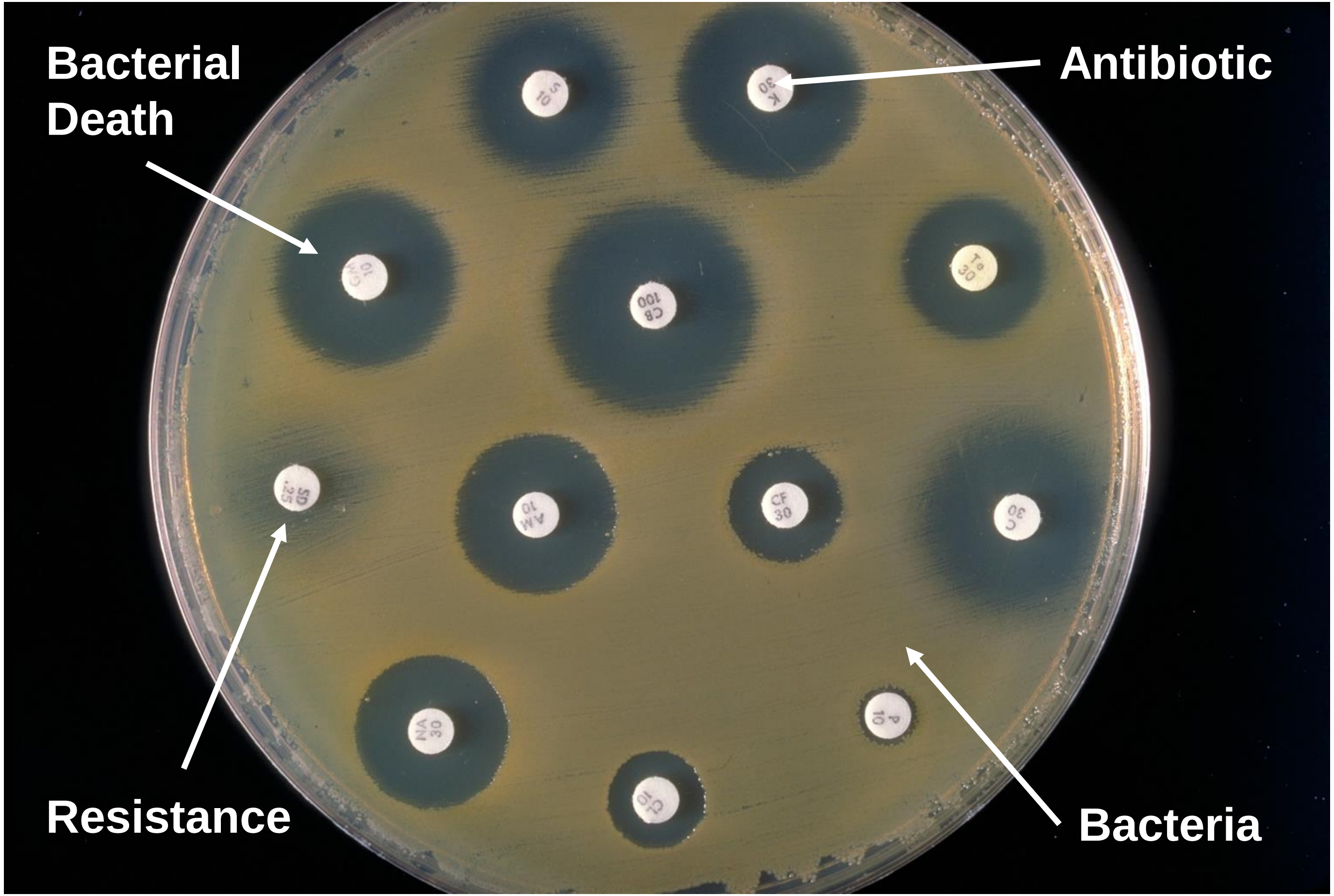
R-Ladies

# Hacking Antibiotic Resistance

**Victoria Butt**
**Bioinformatics PhD Student, King’s College London**



# 10 Million

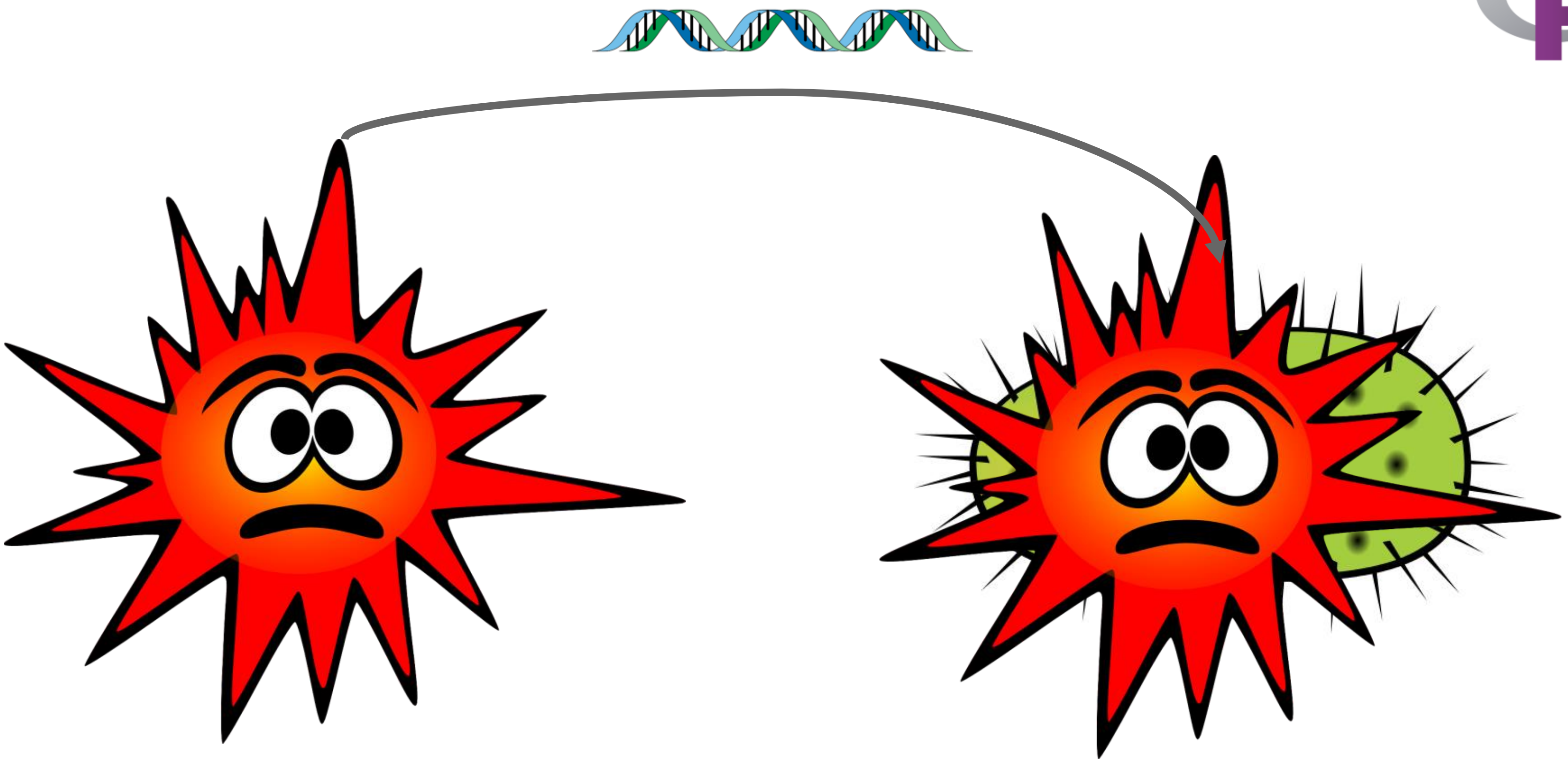


Bacterial Death

Antibiotic

Resistance

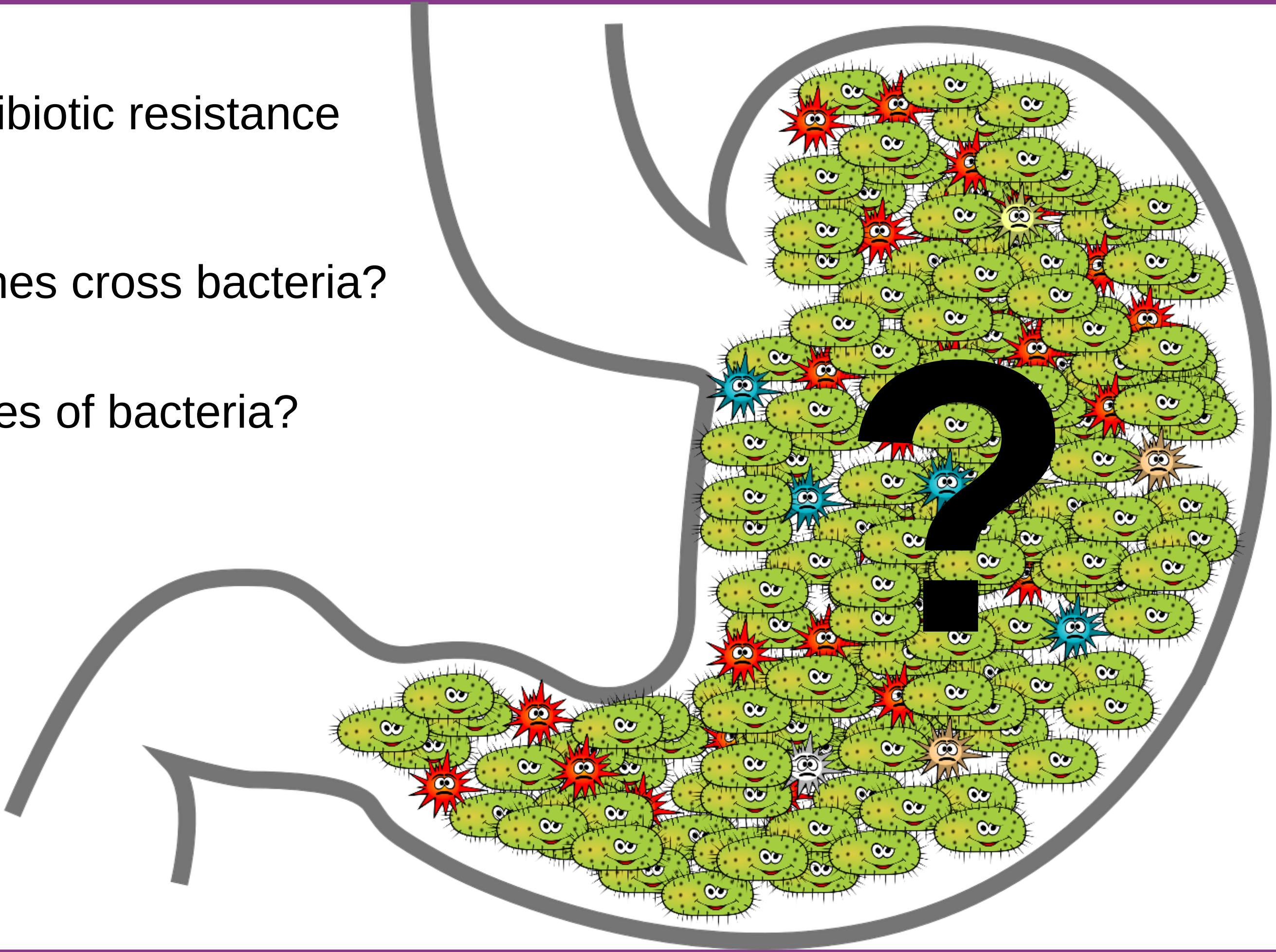
Bacteria

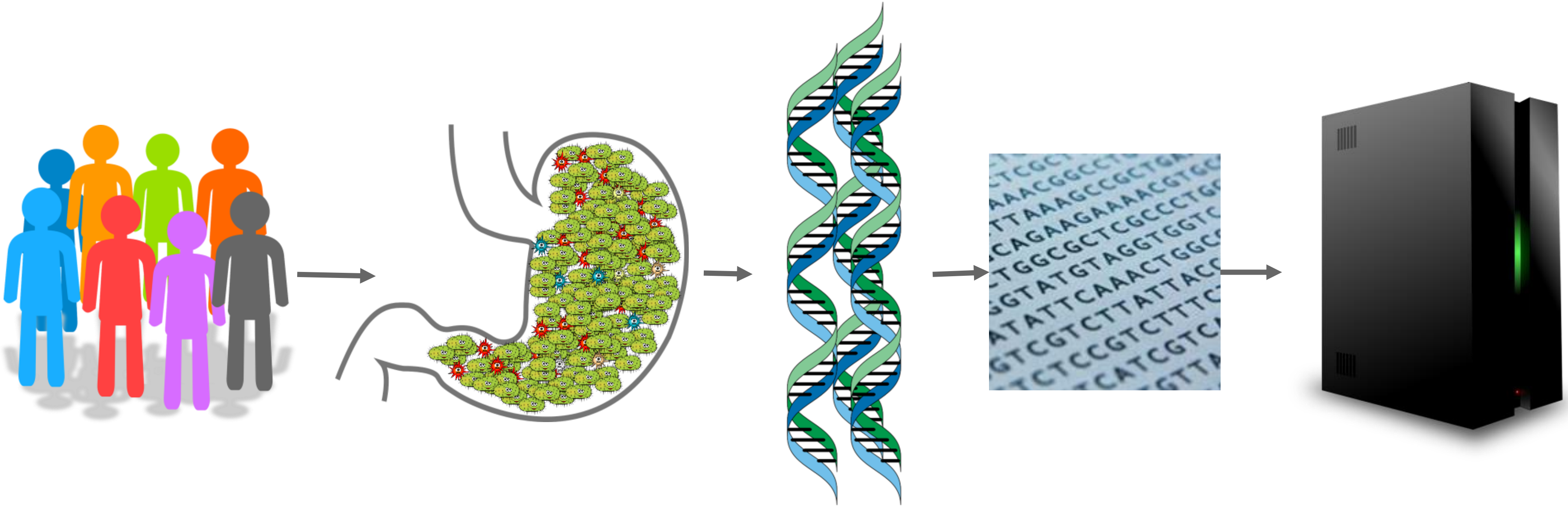


What antibiotic resistance genes?

What genes cross bacteria?

What types of bacteria?




CTCGCC
AAACGGCCGCTGA
TTAAAGCCGAAAACGTGC
CAGAAGAAAACTCGCCCTG
TGGCGCTCGCCCTGGTC
GGTATGTAGGTGGTC
ATATTCAAACTGGC
GTCGTCTTATTACC
TCTCCCGTCTTTTC
CATCGTCA
GTTA

map

full_join

# Thank you for listening!

@ResearchersCode

meetup.com/researchers-code

```r
library(dplyr)

rladies_global %>%
  filter(city == ‘London')
```

# GETTING STARTED WITH TIDY EVAL

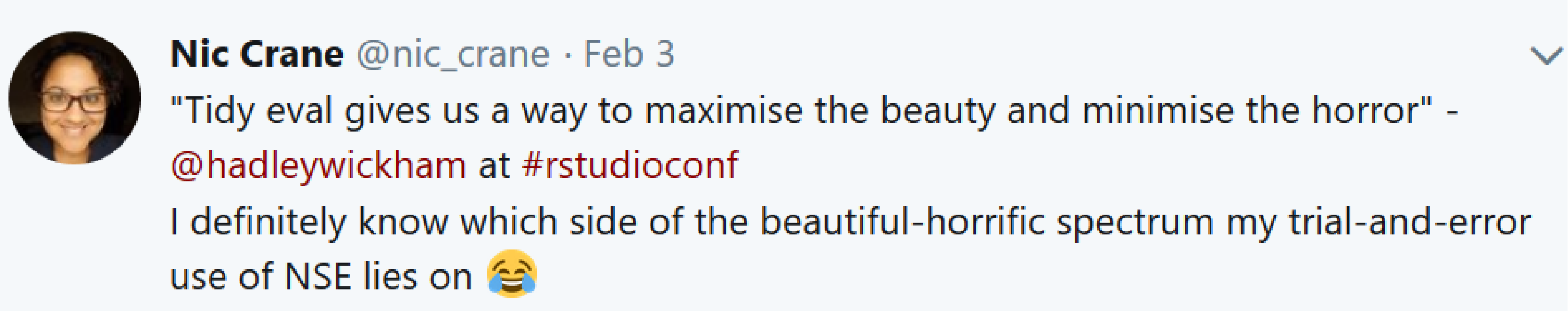
# What is tidy eval and why should I care?

# You should care
## if:

- You write your own R functions
- You want to use functions from dplyr (and tidyr) inside your functions

# You should care
# if:

- This is also you



**Nic Crane** @nic_crane · Feb 3

"Tidy eval gives us a way to maximise the beauty and minimise the horror" - @hadleywickham at #rstudioconf

I definitely know which side of the beautiful-horrific spectrum my trial-and-error use of NSE lies on 😂

# Simple?

```
> library(dplyr)

> wrangle_data <- function(data, column, val){

  data %>%
    select(column) %>%
    filter(column == val)

}

> wrangle_data(iris, "Species", "versicolor")
```

# Nope!

```
[1] Species
<0 rows> (or 0-length row.names)
```

☹

> *“Most dplyr functions use non-standard evaluation (NSE). This is a catch-all term that means they don’t follow the usual R rules of evaluation. Instead, they capture the expression that you typed and evaluate it in a custom way.*

# Fix

```
library(rlang)
wrangle_data <- function(data, column, val){

  data %>%
    select(!!sym(column)) %>%
    filter(!!sym(column) == val)

}
```

# Hooray!

```
> wrangle_data(iris, "Species", "versicolor")

Species
1   versicolor
2   versicolor
3   versicolor
4   versicolor
5   versicolor
6   versicolor
```

# Huh?

## !!

- “Bang bang”
- Overrides dplyr’s “special” behaviour

*“The !! operator unquotes its argument. It gets evaluated immediately in the surrounding context.”*

```r
library(rlang)
wrangle_data <- function(data, column, val){

  data %>%
    select(!!sym(column)) %>%
    filter(!!sym(column) == val)

}
```

# Huh?

## sym

- Converts to a symbol
- “Species” -> Species

# Other important tidy eval functions

# Other important functions & concepts

| Concept | Key functions | Guide |
|---|---|---|
| Writing your own dplyr-style functions | `enquo()` | https://bit.ly/2JcG4oJ |
| Quasiquotation (theory) | `sym(), !!` | https://bit.ly/2HQFQnO |
| Debugging your tidy eval code | `qq_show()` | |

# Other resources

| Resource | URL |
|---|---|
| RStudio tidy eval webinar | https://www.rstudio.com/resources/webinars/tidy-eval/ |
| dplyr programming vignette | https://dplyr.tidyverse.org/articles/programming.html |
| Edwin Thoen - dplyr recipes | https://edwinth.github.io/blog/dplyr-recipes/ |
| Nic Crane – tidy eval posts | https://thisisnic.github.io/tags/tidyeval/ |

```
library(dplyr)

rladies_global %>%
  filter(city == ’London')
```

# What should I have for lunch 🐟🍒🍔?

# Hello!

## I am Emma

You can find me at @gummifot or emmavestesson.com

# Me at 12.30

## But where to go for lunch?

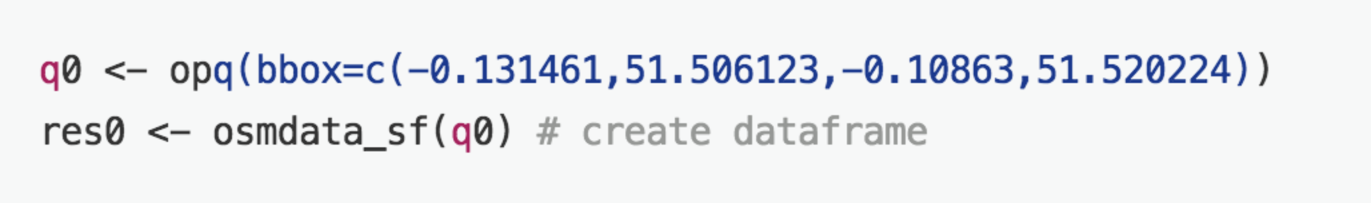
🍣 🍔 🌯 ⍰

# Solution

- Use R to help me a pick a restaurant at random!

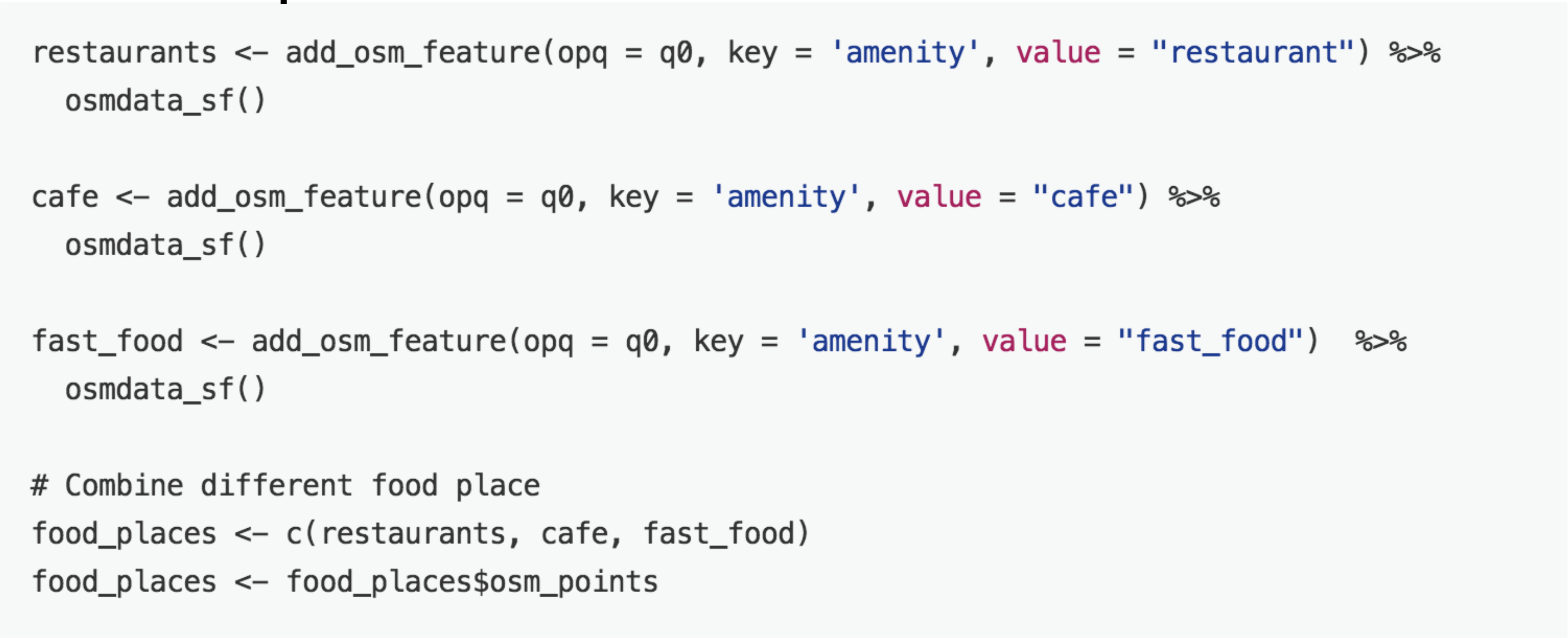
- osmdata – package to access open street map data
- sf – package to work with spatial data
- leaflet – package to build interactive maps
- shiny – adds the reactive part

## Get the data for area



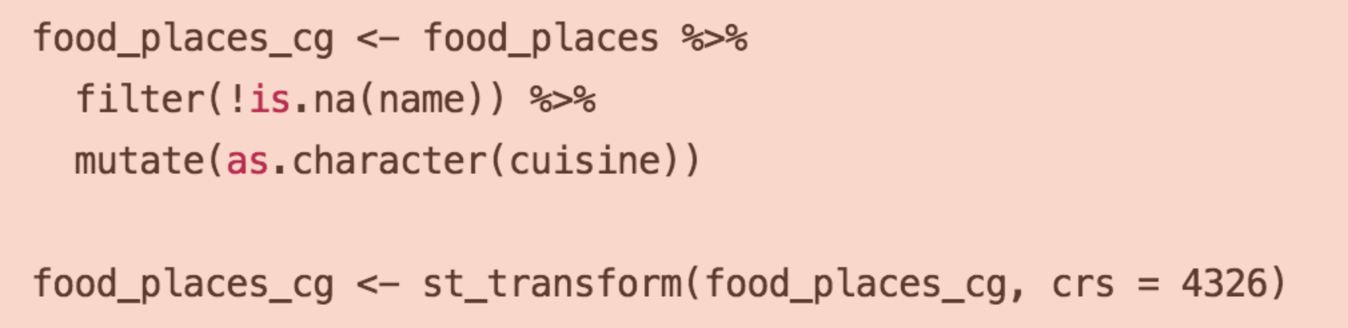
```r
q0 <- opq(bbox=c(-0.131461,51.506123,-0.10863,51.520224))
res0 <- osmdata_sf(q0) # create dataframe
```

## Pick certain parts of the data



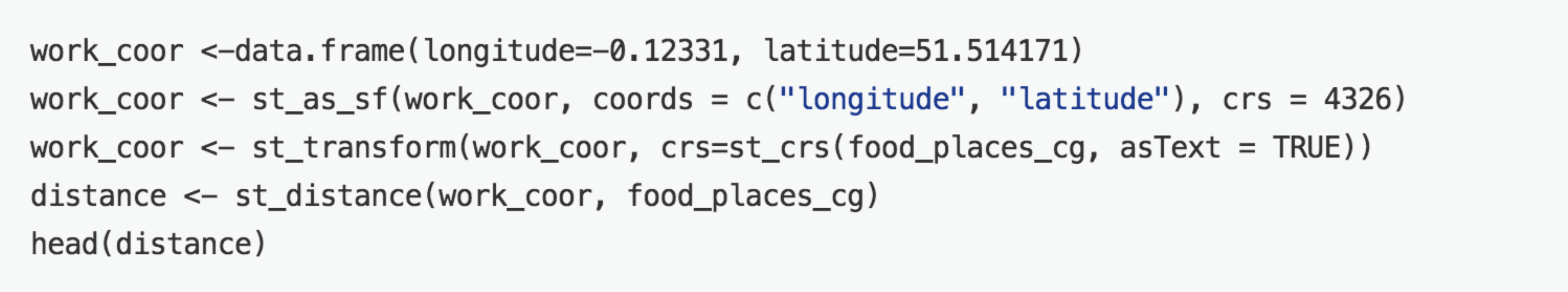
```r
restaurants <- add_osm_feature(opq = q0, key = 'amenity', value = "restaurant") %>%
  osmdata_sf()

cafe <- add_osm_feature(opq = q0, key = 'amenity', value = "cafe") %>%
  osmdata_sf()

fast_food <- add_osm_feature(opq = q0, key = 'amenity', value = "fast_food")  %>%
  osmdata_sf()

# Combine different food place
food_places <- c(restaurants, cafe, fast_food)
food_places <- food_places$osm_points
```

## Clean the data



```
food_places_cg <- food_places %>%
  filter(!is.na(name)) %>%
  mutate(as.character(cuisine))

food_places_cg <- st_transform(food_places_cg, crs = 4326)
```

## Calculate the distance



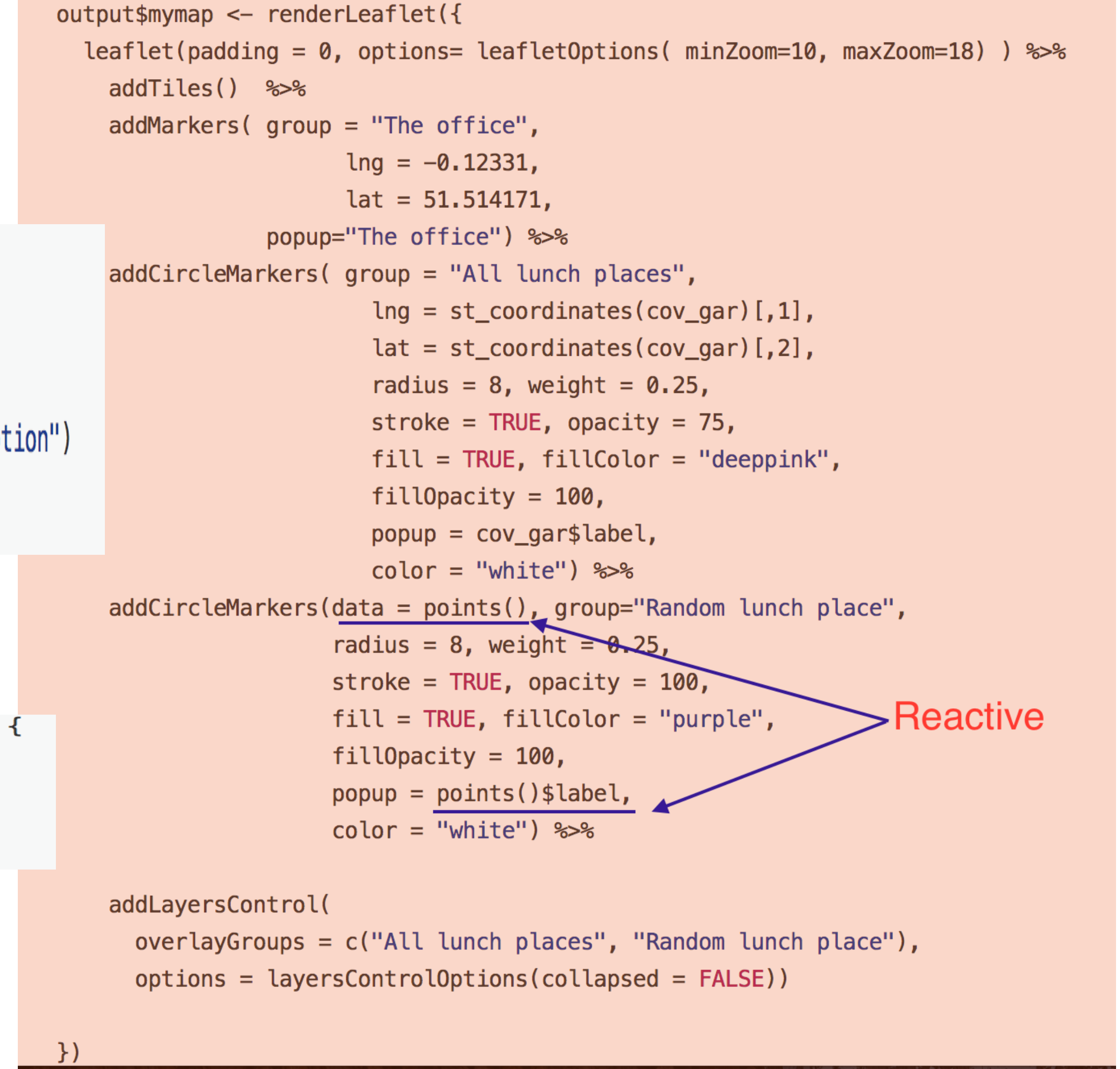
```
work_coor <-data.frame(longitude=-0.12331, latitude=51.514171)
work_coor <- st_as_sf(work_coor, coords = c("longitude", "latitude"), crs = 4326)
work_coor <- st_transform(work_coor, crs=st_crs(food_places_cg, asText = TRUE))
distance <- st_distance(work_coor, food_places_cg)
head(distance)
```

## User interface function

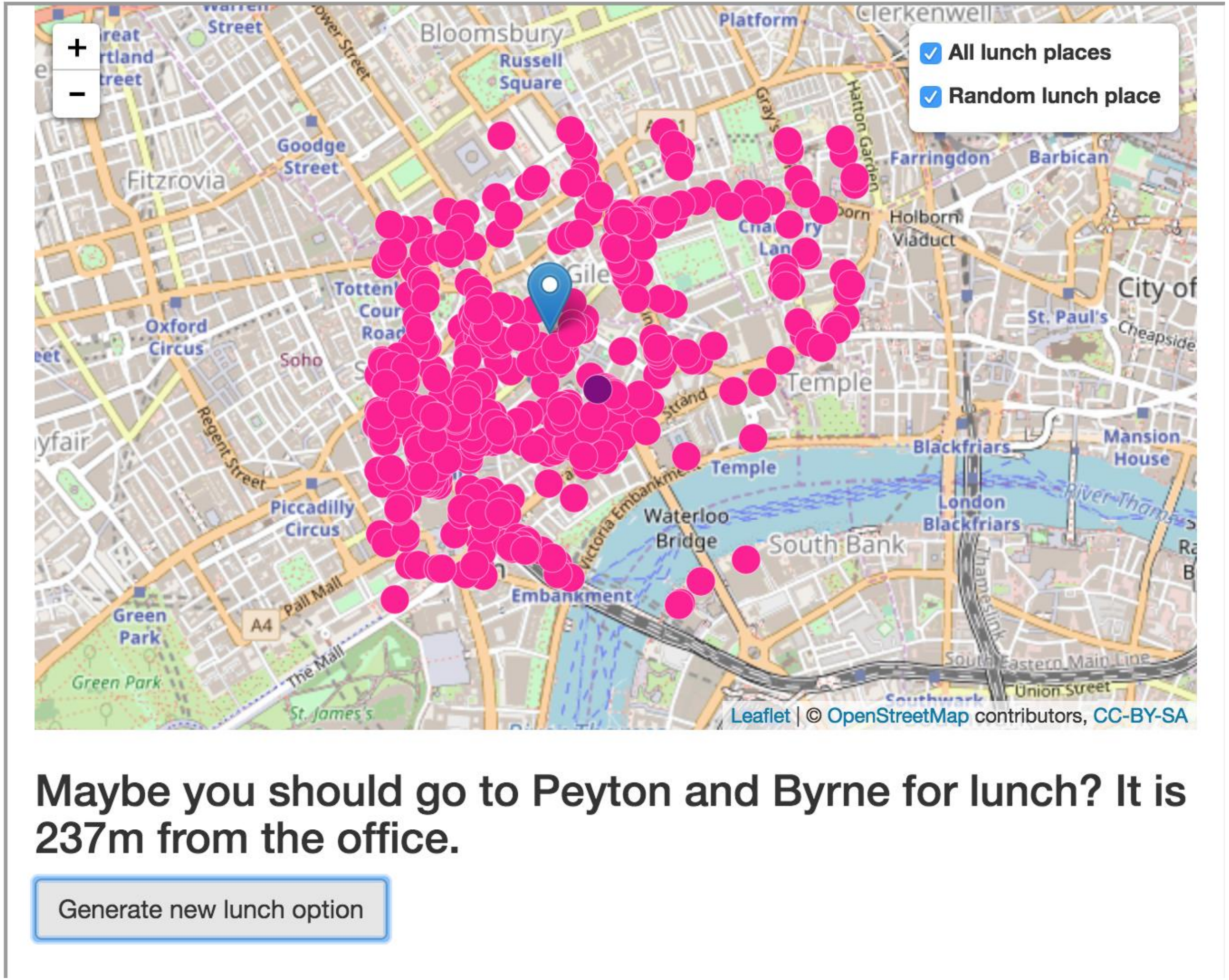
```
ui <- fluidPage(
  leafletOutput("mymap"),
  h3(textOutput("selected_var")),
  actionButton("recalc", "Generate new lunch option")
)
```

## Server function

```
points <- eventReactive(input$recalc, {

  sample_n(cov_gar,1)
}, ignoreNULL = FALSE)
```



```
output$mymap <- renderLeaflet({
  leaflet(padding = 0, options= leafletOptions( minZoom=10, maxZoom=18) ) %>%
    addTiles()  %>%
    addMarkers( group = "The office",
                      lng = -0.12331,
                      lat = 51.514171,
                popup="The office") %>%
    addCircleMarkers( group = "All lunch places",
                      lng = st_coordinates(cov_gar)[,1],
                      lat = st_coordinates(cov_gar)[,2],
                      radius = 8, weight = 0.25,
                      stroke = TRUE, opacity = 75,
                      fill = TRUE, fillColor = "deeppink",
                      fillOpacity = 100,
                      popup = cov_gar$label,
                      color = "white") %>%
    addCircleMarkers(data = points(), group="Random lunch place",
                    radius = 8, weight = 0.25,
                    stroke = TRUE, opacity = 100,
                    fill = TRUE, fillColor = "purple",
                    fillOpacity = 100,
                    popup = points()$label,
                    color = "white") %>%

    addLayersControl(
      overlayGroups = c("All lunch places", "Random lunch place"),
      options = layersControlOptions(collapsed = FALSE))

})
```



Maybe you should go to Peyton and Byrne for lunch? It is 237m from the office.

Full code: https://emmavestesson.netlify.com/2018/02/what-should-i-have-for-lunch/

# R-Ladies London
## Upcoming Events

Data in London Town [May 29]

Shiny workshop [TBC ~June 20]

Also on the conference circuit-

eRum Budapest [May 14-16]

useR! Brizzie [Jul 10-13]

EARL [Sep 11-13]