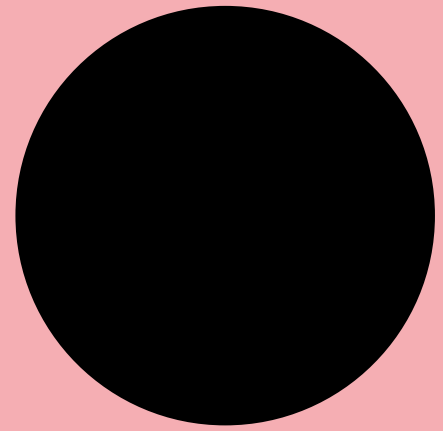
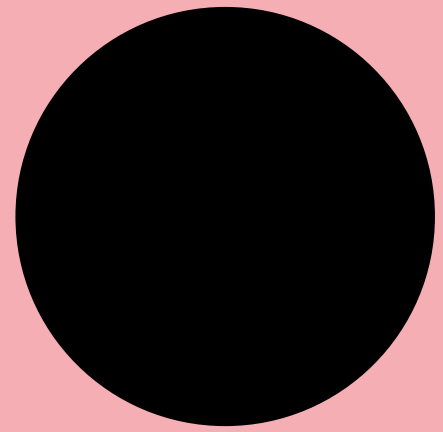
Web Scraping in Market Research





Web scraping and automating the boring

Savanta Services:

Working with clients across their brand & project lifecycle

1

Devising customer centric strategies

- Buyer journey mapping
- Persona & segmentation development

2

Launching category leading products & services

- New product & service development
- Pricing strategy

3

Building powerful brands & marcomms

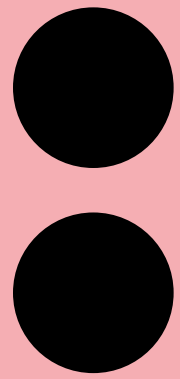
- Brand development & tracking
- Marketing optimisation
- Thought-leadership
- Research-led PR

4

Inspiring customer loyalty

- Optimise the customer experience
- Enhance customer satisfaction

Market Research has a problem…



Houston, we have a problem…

Problem 1

- **“If I’d asked people what they wanted, they would have said a faster horse”-Henry Ford**
- Although respondents can have the best intentions, Research is still susceptible to problems such as:
  - Over stating
  - Flatlining
  - Misinterpreting questions
- Some of these can be helped with good survey design, discussion guides etc. but some are more challenging to overcome:
  - People are often very good at saying **why they <u>think</u>** they behaved a certain way, but it can often be a **false justification**
  - A lot of research is based on **retrospection**
  - Choice based questioning can either be overwhelming with choice ***or*** don’t contain the choice a respondent would actually select.
- One way we could tackle this is by **natural observation,** and fortunately the internet has this in abundance (and free to take)

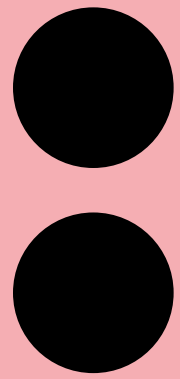
Houston, we have a problem...

Problem 2

- **Two of the most time-consuming tasks in the Market Research industry is Desk Research and Quality Control**
- The need for additional resources and information can happen at any point during a project lifespan:
  - Rationale support
  - Survey Design
  - Investigating the reasons behind a particular outcome.
- Currently this is done manually by using google to find sources and then a lot of copying and pasting to gather information.

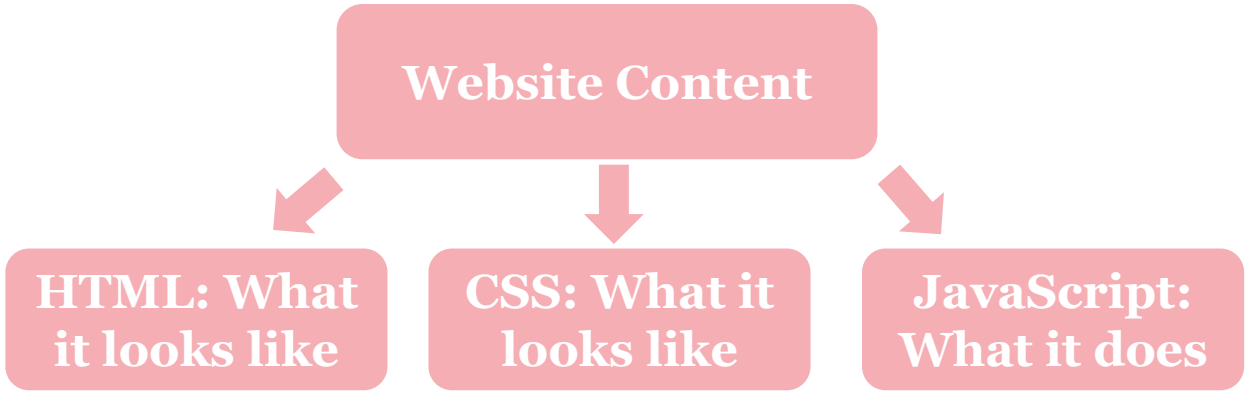
- Dependent on length and complexity, an online survey could need testing anywhere between 1 hour to 2-3 days.
  - Even then, it is almost impossible to test every permutation without some form of random data generation.
  - After a long period of testing it becomes harder to spot mistakes or potential issues.

Using R for Web Scraping



Web Scraping

What is Web Scraping



All websites have the same components an these are **accessible** and **viewable** (to a degree) using F12 or SelectorGadget. This means that if you know how:

- You can extract the components
  - By extracting the components you have access to the content both visible and invisible
    - Content= Data, images, text, widgets, URLs.

Imagine having to perform tasks for problem 1 and problem 2 manually:

- Time Consuming
- Boring

Web Scraping

What Packages do I need.

Rvest: https://cran.r-project.org/web/packages/rvest/rvest.pdf

- **What does it do:** Makes it easier to extract html from a webpage.
- **What you need:** List of URLS to extract information from and node/ CSS selector.
- **Why I like it:** I find it easier to extract and create a dataframe based on my query.
- **Exclusions:** Rvest can't call any Javascript function e.g. clicking or scrolling.

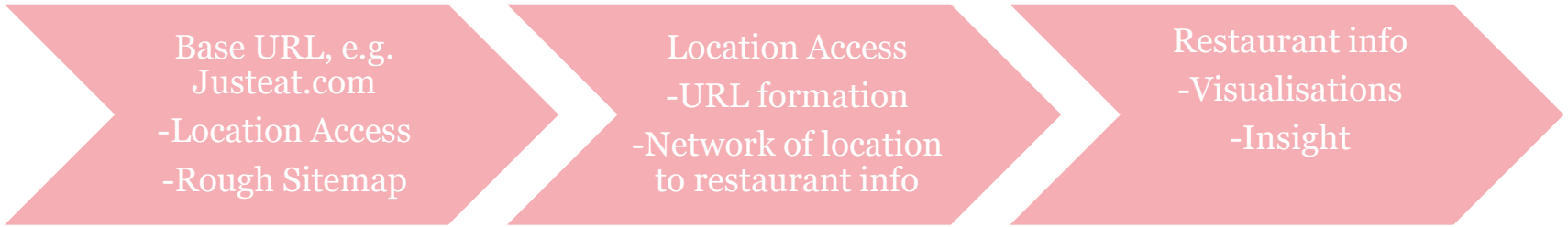
Rselenium: https://cran.r-project.org/web/packages/RSelenium/readme/README.html

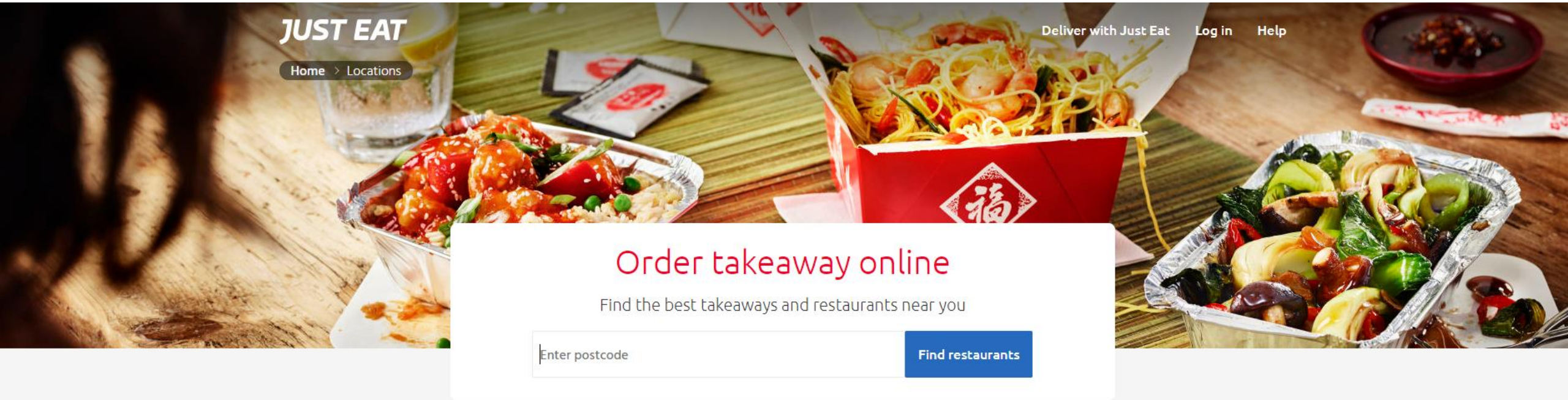
- **What does it do:** Automates browsers. Has numerous applications e.g. scraping, web-testing.
- **What you need:** A list of search terms/ a rough framework of what you want your driver to do.
- **Why I like it**: By setting up the webdriver you can observe your machine performing tasks without the need to click them yourself. It will extract pretty much anything from a webpage.
- **Exclusions:** Can be tricky to run, especially is you don't factor in page loading time for your extraction to take place.

Web Scraping
Rvest Thought Process..

Let's see how we can tackle problem 1...

- **Goal**: In our survey, we found that a lot of people from London preferred takeaways over cooking for themselves. We want to know what takeaways are on offer in this area.
- **Information we have:** A list of London postcodes.
- **Sources we can use:** Just Eat, Deliveroo, UberEats

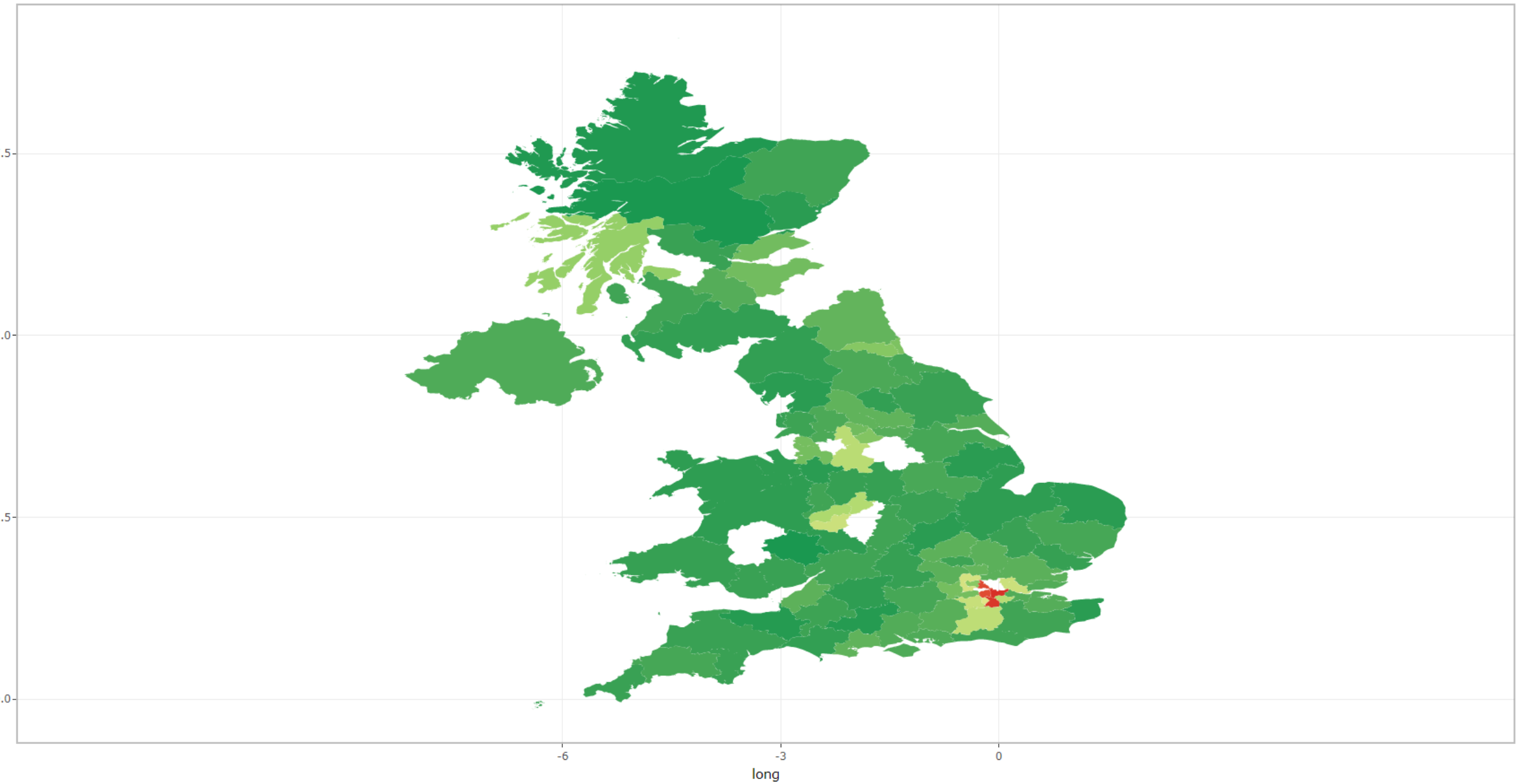



JUST EAT
Home › Locations
Deliver with Just Eat
Log in
Help
Order takeaway online
Find the best takeaways and restaurants near you
Enter postcode
Find restaurants

```r
url1="https://www.just-eat.co.uk/takeaway/"
page1<-read_html(url1) %>% html_nodes('.grouped-link-list__link') %>% html_attr('href')
page1[grep("https://www.just-eat.co.uk",page1,invert = T)]<-paste0("https://www.just-eat.co.uk",page1)
subreg<-page1 %>% map(read_html)
page2<-lapply(page1, . %>% read_html() %>% html_nodes('.grouped-link-list__link') %>% html_attr('href'))
names(page2)<-page1
Areas<-do.call(cbind,page2)

for(i in 1:ncol(Areas)){
  Areas[,i][duplicated(Areas[,i])] <- NA
}
AreaTable<-as.data.frame(Areas)
AreaTable<- AreaTable[!apply(AreaTable == "", 1, all),]
AreaList<-as.data.frame(colnames(AreaTable))
```

```
AreaMelt<-melt(as.matrix(AreaTable))
AreaMelt<-as.data.frame(AreaMelt)
AreaMelt<-AreaMelt[!is.na(AreaMelt$value),]
AreaMelt<-AreaMelt[grep("http",AreaMelt$value),]
AreaMelt$value<-as.character(AreaMelt$value)
AreaMelt$Var2<-gsub("https://www.just-eat.co.uk/takeaway/","",AreaMelt$Var2)
AreaMelt$value<-trimws(AreaMelt$value)
AreaMelt<-AreaMelt[!duplicated(AreaMelt$value),]

URLS1<-AreaMelt$value[1:500]
URLS1<-trimws(URLS1)
```

```r
html<-URLS1 %>% map(read_html)
html2<-URLS2 %>% map(read_html)
html3<-URLS3 %>% map(read_html)
names(html)<- URLS1
names(html2)<- URLS2
names(html3)<- URLS3



df <- html %>%
  map(html_nodes, '.u-clearfix') %>%
  map_df(map_df,
         ~list(

           Name = .x %>% html_node('.c-listing-item-title') %>% html_text(trim = TRUE),
           Cuisine = .x %>% html_node('.c-listing-item-info .c-badge--noPad') %>% html_text(trim = TRUE) ,
           Rating = .x %>% html_node(".c-listing-item-rating") %>% html_text(trim=TRUE)  %>% gsub('\\s+', ' ', .),
           Promo = .x %>% html_node(".c-listing-item-promo-text") %>% html_text(trim=TRUE)  %>% gsub('\\s+', ' ', .),
           link = .x %>% html_attr("href")
         ),
         .id = 'Area ID')
```

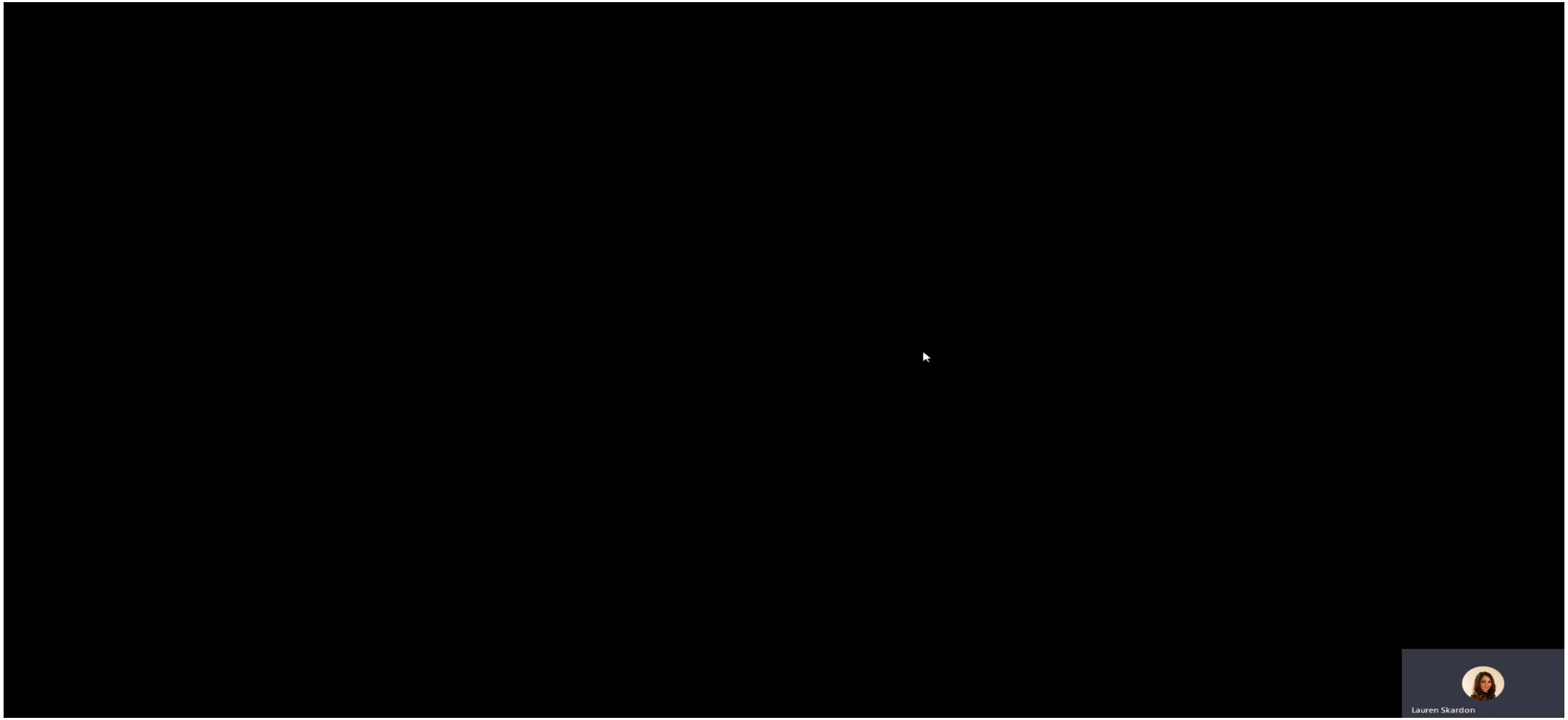

.5
.0
.5
.0
-6
-3
0
long
n
150
100
50

Web Scraping
Selenium in action.

Let’s see how we can tackle problem 2…

- **Goal**: We have a survey going live and not enough time to manually test and no Random Data generator to check base sizing and routing..

Web Scraping
Selenium in action.



Web Scraping
What else can we do?

The demos showed the basics of both packages. But using the basics we can build things that can save businesses a lot of time and money.

- Create indexing for brand availability across the world
- Find and tabulate hotel information e.g. Rating, location and special offers on a global scale.
- Log into CRM management tools to access each project, extract information and download materials.
- Compiling menus from various restaurants.
- Mapping cuisine availability in the UK
- Extracting reviews to perform sentiment analysis and compare discourse vs demographics/ review type.

Basically if it is on the internet, there's a very good chance we can scrape it or use Selenium to automate, removing the time and cost associated with manual research.

Web Scraping
How this can be used in the future of Market Research

We've only explored the tip of the iceberg…

- As supplement to surveys e.g. explaining survey data.
- Monitoring brand attitude, prevalence and forecast as an early warning system.
- Replacing some desk research
- Building hypotheses to input into research/ proposals
- Cutting down questionnaire length by replacing survey questions
- Replacing some qual components e.g. interviewing, open ends, exploratory work
- Visual representation of the current market trends
- Chart the internet (within reason)
- Create dashboards and user interfaces so brands can view and manage their online presence.

FAQ

How easy is this to do?

- As much information as possible e.g. exact account info, data source makes data easier to locate
- Design input is helpful
- Scope dependent

What are the issues that could arise?

- Incomplete data
- Restricted data e.g. Facebook, direct Google scrape.
- Not enough data

How much does it cost?

- Time e.g. For **just** follower counts over time Est 1 day (based on 1,000 brands -finding accounts), 1 day to compile. 1 day for sentiment analysis (one brand) vs. 3-5 days in desk research.
- Companies that specialize in Web scraping generally charge around £200 for initial set up and £100 per link. If you need to scrape 1,000 links, you are saving your business £1,200

Any Questions?