

APPLICATIONS



OF DATA SCIENCE

Intro to Web Scraping

Applications of Data Science - Class Bonus

Giora Simchoni

gsimchoni@gmail.com and add #dsapps in subject

Stat. and OR Department, TAU

2022-12-26

APPLICATIONS



OF DATA SCIENCE

The Three Rules of Web Scraping

APPLICATIONS



OF DATA SCIENCE

Rule 1: Do you *really* need web scraping?

There are data APIs for just about anything, you know...

The image is a collage of nine logos, each representing a different data API or service:

- CHUCKNORRIS.IO**: A logo featuring a cartoon character of Chuck Norris wearing a cowboy hat and holding a rifle, with the text "CHUCKNORRIS.IO" below it.
- PokeAPI**: A logo for "The RESTful Pokémon API" with the tagline "Serving over 60,000,000 API calls each month!"
- Harvard Art Museums**: Logos for the Fogg Museum, Busch-Reisinger Museum, and Arthur M. Sackler Museum.
- FBI MOST WANTED**: The iconic black and white "FBI MOST WANTED" poster.
- FLUTRACK**: A logo with the word "FLUTRACK" in red and a small icon.
- ALPHA VANTAGE**: A logo with the words "ALPHA VANTAGE" in large white letters against a dark background with a circuit board pattern.
- Spotify**: The Spotify logo.
- RainViewer**: The RainViewer logo.
- Clinical Table Search Service**: A logo for the U.S. National Library of Medicine's Clinical Table Search Service, featuring the NIH logo and the text "A web API service for use with autocomplete-lhc & LHC-Forms".

APPLICATIONS



R API Packages

Many of them already accessible with a R/Python package...

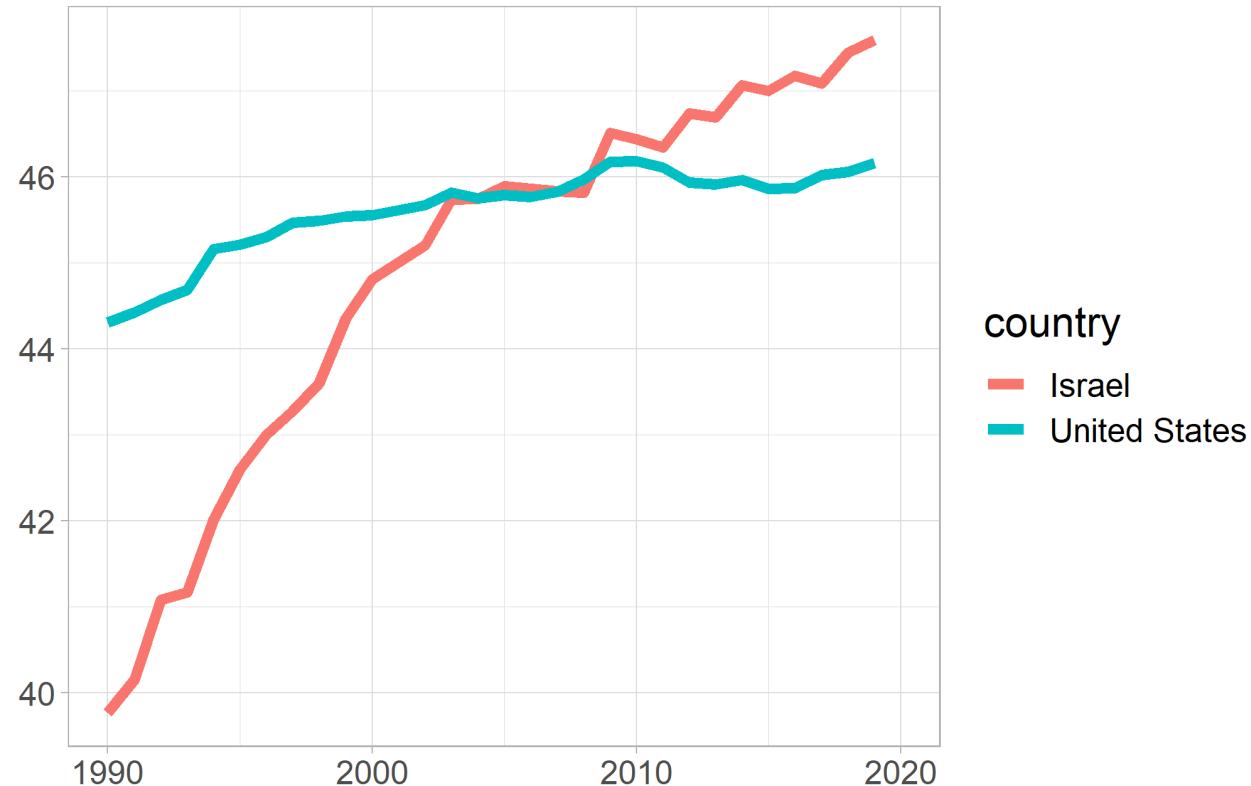
```
library(wbstats)

female_labor <- wb_data(
  indicator = c("women_lab_share" = "SL.TLF.TOTL.FE.ZS"),
  start_date = 1990,
  end_date = 2020
)

female_labor %>%
  filter(country %in% c("Israel", "United States")) %>%
  ggplot(aes(date, women_lab_share, color = country)) +
  geom_line(lwd = 2) +
  labs(title = "Share of women in labor force") +
  theme_light() +
  theme(text = element_text(size=16))
```

From: <https://cfss.uchicago.edu/notes/application-program-interface/>

% of women in labor force



country

- Israel
- United States

The datapasta package

My gift to you.



Rule 2: Learn some HTML first!

HTML is a set (or tree) of *elements*, marked by *HTML tags*:

```
<!DOCTYPE html>
<html>
<head>
<title>My Awesome Webpage</title>
</head>
<body>

<h1>My Superb Heading</h1>
<p>I am going to be a web designer.</p>

</body>
</html>
```

My Superb Heading

I am going to be a web designer.

- First children in the tree: header and body
- View any page's HTML (on chrome) with right-click and "View page source" (or Ctrl + U)

Useful elements and attributes to know

- <p> for paragraph </p>
- <h1> for headings </h1>
-
, <hr> for breaks
- for links
- <i> For bold, italic etc. </i>
-
- <p style="color:DodgerBlue;"> for font color </p>

HTML Tables

A big thing when it comes to data as you can imagine...

```
<!DOCTYPE html>
<html>
<body>

<h2>Basic HTML Table</h2>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Britney</td>
    <td>Spears</td>
    <td>39</td>
  </tr>
  <tr>
    <td>Christina</td>
    <td>Aguillera</td>
    <td>40</td>
  </tr>
  <tr>
    <td>Beyonce</td>
    <td>Knowles</td>
    <td>39</td>
  </tr>
</table>

</body>
</html>
```

Basic HTML Table

Firstname	Lastname	Age
Britney	Spears	39
Christina	Aguillera	40
Beyonce	Knowles	39

HTML Classes

A class attribute is defined in a style sheet, lets you repeat a style.

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
    background-color: tomato;
    color: white;
    border: 2px solid black;
    margin: 20px;
    padding: 20px;
}
</style>
</head>
<body>

<div class="city">
<h2>London</h2>
<p>London is the capital of England.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital of France.</p>
</div>

<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

The image displays three separate orange rectangular boxes, each containing a city name in bold and a descriptive sentence below it. The first box contains 'London' and 'London is the capital of England.' The second box contains 'Paris' and 'Paris is the capital of France.' The third box contains 'Tokyo' and 'Tokyo is the capital of Japan.'

City	Description
London	London is the capital of England.
Paris	Paris is the capital of France.
Tokyo	Tokyo is the capital of Japan.

Rule 3: Be polite!

With great power comes great responsibility.

See e.g. the [polite](#) package.



rvest

APPLICATIONS



OF DATA SCIENCE

read_html()

You're now a NLP expert, and you've just developed a SOTA Q&A model. How would you demonstrate your model's performance?

How about [triviaquestionsnow.com?](https://www.triviaquestionsnow.com/)

Let's scrape a few Q&As. Politely.

```
library(rvest)  
  
url <- "https://www.triviaquestionsnow.com/for/sports-trivia"  
  
html_obj <- read_html(url)
```

`read_html()` is usually where you'd start. What did you get?

```
class(html_obj)  
  
## [1] "xml_document" "xml_node"
```

View page source

With time, you'll become friendly with this weird object. Right now it is better than...

```
1 <!DOCTYPE html>
2 <html lang="en" ng-app="triviaApp">
3 <head>
4   <meta charset="utf-8" />
5   <meta http-equiv="x-ua-compatible" content="ie=edge" />
6   <link rel="icon" href="https://www.triviaquestionsnow.com/favicon.ico" type="image/x-icon" />
7   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8   <meta name="description" content="Want to put your sports knowledge to the test? triviaquestionsnow is the right place for you. Work your brain and have fun with our Sports Trivia questions." />
9   <meta name="keywords" content="Sports Trivia, Sports Trivia questions" />
10
11  <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png" />
12  <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png" />
13  <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png" />
14  <link rel="manifest" href="/site.webmanifest" />
15  <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5" />
16  <meta name="msapplication-TileColor" content="#da532c" />
17  <meta name="theme-color" content="#ffffff" />
18
19  <link rel="canonical" href="https://www.triviaquestionsnow.com/for/sports-trivia" />
20
21  <meta property="fb:app_id" content="329788460996850" />
22  <meta property="og:url" content="https://www.triviaquestionsnow.com/for/sports-trivia" />
23  <meta property="og:type" content="website" />
24    <meta property="og:title" content="Sports Trivia Questions and Answers" />
25      <meta property="og:description" content="Want to put your sports knowledge to the test? triviaquestionsnow is the right place for you. Work your brain and have fun with our Sports Trivia questions." />
26      <meta property="og:image" content="https://www.triviaquestionsnow.com/trivia-questions-and-answers.jpg" />
27    <title>Sports Trivia Questions and Answers | TQNK</title>
28
29  <!-- Fonts -->
30  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css" integrity="sha384-XdYbMnZ/QJNlO34U2DzIuq/lnZoWZuPj7LmVvZDyQF0XuHfC9k" />
31  <link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700,300" rel="stylesheet" type="text/css" />
32  <link href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-icons.css" rel="stylesheet" />
33
34  <!-- Styles -->
35  <link href="https://www.triviaquestionsnow.com/css/all.css?v=1" rel="stylesheet" />
```

`html_children()` and `html_node()`

Our tree has two children: head and body

```
html_obj %>% html_children()
```

```
## {xml_nodeset (2)}
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=...
## [2] <body>\n      <div class="title-bar" data-responsive-toggle="nav" data...
```

Again notice the object returned might not be familiar
("xml_nodeset")

And each of the children has children of its own:

```
html_obj %>% html_node("body") %>% html_children()
```

```
## {xml_nodeset (6)}
## [1] <div class="title-bar" data-responsive-toggle="nav" data-hide-for="m...
## [2] <div id="nav" class="top-bar">\n      <div class="row">\n...
## [3] <div class="wrap bg-grey-light t-pad-20 b-pad-20">\n      <div...
```

html_nodes()

Usually we'd figure out a rule and want a list of all relevant nodes:

```
html_obj %>% html_nodes("img")
```

```
## {xml_nodeset (8)}
## [1] % html_nodes("a")
```

```
## {xml_nodeset (44)}
## [1] <a href="/">\n                               
## [2] <a href="https://www.triviaquestionsnow.com" class="no-pad">\n                               
## [3] <a href="https://www.triviaquestionsnow.com/easy-trivia-questions">\n                               
## [4] <a href="https://www.triviaquestionsnow.com/for/sports-trivia">Sports\n                               
## [5] <a href="https://www.triviaquestionsnow.com/for/music-trivia">Music\n                               
```

html_attrs()

Getting a specific attribute from those nodes:

```
html_obj %>% html_nodes("img") %>% html_attr("src")
```

```
## [1] "https://www.triviaquestionsnow.com/img/trivia-questions.png"
## [2] "https://www.triviaquestionsnow.com/img/trivia-questions.png"
## [3] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1"
## [4] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1"
## [5] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1"
## [6] "https://www.triviaquestionsnow.com/img/category/360x130/apologetics"
## [7] "https://www.triviaquestionsnow.com/img/category/360x130/-category-1"
## [8] "https://www.triviaquestionsnow.com/img/category/360x130/-category-2"
```

```
html_obj %>% html_nodes("a") %>% html_attr("href")
```

```
## [1] "/"
## [2] "https://www.triviaquestionsnow.com"
## [3] "https://www.triviaquestionsnow.com/easy-trivia-questions"
## [4] "https://www.triviaquestionsnow.com/for/sports-trivia"
## [5] "https://www.triviaquestionsnow.com/for/music-trivia"
## [6] "https://www.triviaquestionsnow.com/for/math-trivia"
## [7] "https://www.triviaquestionsnow.com/categories"
```

`html_text()`

Getting the text from whatever set of elements we have:

```
html_obj %>% html_nodes("a") %>% html_text()
```

```
## [1] "\n"
## [2] "\n"
## [3] "Easy Trivia"
## [4] "Sports Trivia"
## [5] "Music Trivia"
## [6] "Math Trivia"
## [7] "Categories"
## [8] "All Trivia"
## [9] "\n"
## [10] "Show Answer"
## [11] "\n"
## [12] "Show Answer"
## [13] "\n"
## [14] "Show Answer"
## [15] "\n"
## [16] "Show Answer"
## [17] "\n"
## [18] "Show Answer"
## [19] "\n"
## [20] "Show Answer"
```

The popular 1973 "Battle of the Sexes" was a t

Which NBA personality came under fire in 2016 fo

Who did Andy Murray defeat in the semi-finals to

Who were the victors of the women's doubles of t

The 2000 French Open men's singles Championship .

During the entire 2017 season, Patriots' wide re

How to get to those questions? Option 1

Look at the page source, get some identifier yourself (class, ID, link)

Who was the only person to have won a Super Bowl as a player, as an assistant coach and as a head coach?

VeryHard

Show Answer

```
<div class="question callout" ng-controller="QuestionController as question">
<input type="hidden" ng-model="question.data.id" ng-init="question.data.id='1443'">
<span class="float-right light-grey bold l-cush-10">VeryHard</span>
<h3 class="fs-1 bold">
<a href="https://www.triviaquestionsnow.com/question/person-to-win-super-bowl-as-player-as-assistant-coach-and-a
Who was the only person to have won a Super Bowl as a player, as an assistant coach and as a head coach?
</h3>
<div class="t-pad-10">
<a href="#" class="click-to-show bold" ng-click="question.clickShow($event)">Show Answer</a>
<div class="clearfix"></div>
<p class="answer hide b-cush-0 t-cush-10" ng-class="{ 'answer': true, 'hide': !(question.showAnswer) }">
<input type="hidden" name="question_position_1443" value="1">
<input type="hidden" ng-model="question.data.id" ng-init="question.data.id='1443'">
<input type="hidden" ng-model="question.data.answer" ng-init="question.data.answer='mike-ditka'">
<strong>Answer: </strong>Mike Ditka</em>
<br />
<span class="dark-grey"><!--<strong>More information:</strong>-->Ditka won as a player with the Cowboys
</p>
</div>
</div>
```

```
html_obj %>% html_nodes(".question") %>% .[[1]]
```

```
## {html_node}
## <div class="question callout" ng-controller="QuestionController as quest.
## [1] <input type="hidden" ng-model="question.data.id" ng-init="question.d
## [2] <span class="float-right light-grey bold l-cush-10">Hard</span>
## [3] <h3 class="fs-1 bold">\n                                <a href="https://www.triviaquest.
## [4] <div class="t-pad-10">\n                                <a href="#" class="click-to-show
```

After some trial and error...

```
html_obj %>%
  html_nodes(".question") %>%
  html_nodes(".fs-1") %>%
  html_text() %>%
  str_trim()

## [1] "The popular 1973 \"Battle of the Sexes\" was a tennis match between"
## [2] "Which NBA personality came under fire in 2016 for using the word \
## [3] "Who did Andy Murray defeat in the semi-finals to qualify for the 2017"
## [4] "Who were the victors of the women's doubles of the 2008 Australian Open?"
## [5] "The 2000 French Open men's singles Championship was won by whom?"
## [6] "During the entire 2017 season, Patriots' wide receiver Julian Edelman had the most receiving yards in the NFL. Who was he?"
## [7] "What is the average total length of a NFL football game?"
## [8] "Which of the following golf clubs is not located in Canada?"
## [9] "In what year did Vince Lombardi, the man whom the trophy awarded to the Super Bowl champion is named after, die?"
## [10] "Which Country won the 2018 Fed Cup?"
```

How to get to those questions? Option 2

SelectorGadget!



From here it's a function fest!

```
extract_questions_and_answers_from_page <- function(url) {  
  html_obj <- read_html(url)  
  levels <- html_obj %>% html_nodes(".question") %>%  
    html_nodes(".l-cush-10") %>% html_text()  
  questions <- html_obj %>% html_nodes(".question") %>%  
    html_nodes(".fs-1") %>% html_text() %>% str_trim()  
  answers <- html_obj %>% html_nodes(".question") %>%  
    html_nodes(".answer") %>% html_text() %>%  
    str_extract(., "Answer:.*") %>% str_replace("Answer: ", "")  
  tibble(level = levels, question = questions, answer = answers)  
}  
  
extract_questions_and_answers_from_page(url)  
  
## # A tibble: 10 × 3  
##   level    question  
##   <chr>    <chr>  
## 1 Easy     Which of the following are not types of tennis court?  
## 2 Easy     What country was host to the 1930 inaugural FIFA Football Worl.  
## 3 Easy     Which Country won the 2015 Davis Cup?  
## 4 Medium   Who won the title for the men's singles of the 2011 Australian.  
## 5 Medium   Which tennis player won the 2016 Laurens World Sports Award?  
## 6 Easy     In what year were women first allowed to participate in the mo.  
## 7 Easy     Which country lost to Croatia in the 2018 Davis Cup final matc.  
## 8 Medium   Where did the event of curling originate?
```

Pagination

```
create_page_url <- function(topic, page_num) {  
  str_c("https://www.triviaquestionsnow.com/for/", topic, "-trivia")  
}  
  
extract_multiple_pages_single_topic <- function(topic, n = 5) {  
  cat(topic, "\n")  
  res <- map_dfr(  
    1:n,  
    function(i) {  
      cat(" ", i)  
      url <- create_page_url(topic, i)  
      extract_questions_and_answers_from_page(url)  
    }  
  )  
  res$topic <- topic  
  cat("\n")  
  res  
}
```

```
extract_multiple_pages_single_topic("sports")
```

```
## # A tibble: 50 × 4
##   level    question      answer
##   <chr>    <chr>        <chr>
## 1 VeryHard "In 1948, which NBA basketball team did the Harlem Glo... Minnesota
## 2 Medium   "The Jacksonville Jaguars and Carolina Panthers entered ... 1995
## 3 Hard     "An automatic progression by a player to the next stag... Bye
## 4 VeryHard "In 2016, Giants' wide receiver Odell Beckham, Jr. applic... Code
## 5 Hard     "Which NBA player broke the record for most points sco... Jere
## 6 Medium   "Before relocating to Foxborough, Massachusetts, what ... Boston
## 7 Easy     "What is the term for the historic jerseys today worn ... Throw
## 8 Medium   "Who served as the starting center of the Golden State... Andre
## 9 Easy     "In what year was the 4 minute mile achieved?"           1954
## 10 Hard    "Who was the first tennis player to complete a \"Grand... Don
## # ... with 40 more rows
```

Magic!

```
topics <- c("sports", "kids", "science", "bible", "food-drink", "video-games")  
  
df_all <- map_dfr(  
  topics,  
  extract_multiple_pages_single_topic  
)  
  
df_all %>% count(topic)  
  
## # A tibble: 8 × 2  
##   topic          n  
##   <chr>     <int>  
## 1 bible        50  
## 2 food-drink   50  
## 3 geography    50  
## 4 history      50  
## 5 kids         50  
## 6 science      50  
## 7 sports       50  
## 8 video-games  50
```

BeautifulSoup

APPLICATIONS



OF DATA SCIENCE

Almost always start with

```
import requests
from bs4 import BeautifulSoup

html_obj = requests.get('https://en.wikipedia.org/wiki/List_of_The
soup = BeautifulSoup(html_obj.content, 'html.parser')
type(soup)

## <class 'bs4.BeautifulSoup'>
```

This object has all sorts of attributes and methods:

```
soup.get_text()
soup.prettify()
soup.attrs
soup.children
soup.title
```

`find()` a tag, `find_all()`

```
link_objs = soup.find_all('a', href=True)
type(link_objs)

## <class 'bs4.element.ResultSet'>

type(link_objs[3])

## <class 'bs4.element.Tag'>

link_objs[3].text

## 'reality television'

link_objs[3].attrs

## {'href': '/wiki/Reality_television', 'title': 'Reality television'}
```

See the actual link in the [page](#).

Getting that table

```
table = soup.find('table', attrs={'class':'wikitable'})
table_body = table.find('tbody')

rows = table_body.find_all('tr')
print(len(rows))

## 155

print(rows[0])

## <tr>
## <th rowspan="2">Installment
## </th>
## <th rowspan="2">Housewives
## </th>
## <th rowspan="2">First season<br/>starred
## </th>
## <th rowspan="2">Last season<br/>starred
## </th>
## <th colspan="3">Number of seasons
## </th></tr>
```

Getting a Housewife name

```
<tr>
<td><span data-sort-value="De La Rosa, Jo&nbsp;!">Jo De La Rosa</span>
</td>
<td>1
</td>
<td>2
</td>
<td>2
</td>
<td>0
</td>
<td>2
</td></tr>
<tr>
<td><span data-sort-value="Gunvalson, Vicki&nbsp;!"><a href="/wiki/Vicki_Gunvalson" title="Vicki Gunvalson">Vicki Gunvalson</a></span>
</td>
<td>1
</td>
<td>13
```

```
import re

print(rows[3].find('span', attrs = {'data-sort-value': re.compile
```



```
## <span data-sort-value="De La Rosa, Jo !">Jo De La Rosa</span>
```

Getting only HWives with Wiki pages

```
housewives_with_links = []
for row in rows:
    housewife = row.find('span',
        attrs = {'data-sort-value': re.compile(r'.*')})
    if housewife is not None:
        link = housewife.find('a')
        if link is not None:
            housewives_with_links.append((housewife.text, link['href']))

import pandas as pd

h_df = pd.DataFrame(housewives_with_links, columns=['name', 'link'])

h_df.head()
```

```
##                                name                      link
## 0      Vicki Gunvalson          /wiki/Vicki_Gunvalson
## 1      Jeana Keough           /wiki/Jeana_Keough
## 2      Heather Dubrow         /wiki/Heather_Dubrow
## 3  Shannon Storms Beador   /wiki/Shannon_Storms_Beador
## 4      Meghan King            /wiki/Meghan_King
```

(Though if your table is simple, try:)

```
l = pd.read_html(html_obj.text)  
l[0].head()
```

```
##      Installment    Housewives ... Number of seasons  
##      Installment    Housewives ...                   Friend Guest  
## 0  Orange County  Kimberly Bryant ...                      0   3  
## 1  Orange County    Jo De La Rosa ...                      0   2  
## 2  Orange County    Vicki Gunvalson ...                   1   0  
## 3  Orange County     Jeana Keough ...                   1   5  
## 4  Orange County    Lauri Peterson ...                   1   1  
##  
## [5 rows x 7 columns]
```

Following HWives Links

```
def get_housewife_img_ref(housewife_link):
    html_obj = requests.get('https://en.wikipedia.org' + housewife_link)
    soup = BeautifulSoup(html_obj.content, 'html.parser')
    infobox = soup.find('table', attrs = {'class': 'vcard'})
    if infobox is not None:
        img_obj = infobox.find('img', src=True)
        if img_obj is not None:
            return img_obj['src']
    return None

h_df['img_ref'] = h_df['link'].apply(get_housewife_img_ref)
h_df.dropna(inplace=True)

h_df.head()
```

```
##                                     name ...
## 0          Vicki Gunvalson ...
## 1          Luann de Lesseps ...
## 2          Bethenny Frankel ...
## 3  Kelly Killoren Bensimon ...
## 4          Carole Radziwill ...
##
## [5 rows x 3 columns]
```

Downloading HWives Images

```
def make_img_filename(hf_name):
    return 'data/housewives/' + hf_name.lower().strip(',.-').replace(' ', '_')

def download_hw_img(hf_name, hf_img_ref):
    img_file = make_img_filename(hf_name)
    img_data = requests.get('http:' + hf_img_ref).content
    with open(img_file, 'wb') as handler:
        handler.write(img_data)

h_df.apply(lambda row: download_hw_img(row['name'], row['img_ref'])
           , axis=1)
```