Data Technologies Group 1 Update 11/18/2021

- Tyler Antoloci
- Ragan Astle
- Nate Nellis
- McKade Thomas

Image Processing

- All ads have a grid like layout that facilitates cropping
- May attempt to filter all non-black colors to help remove product images
- Front Page advertises dollars off, while rest of the ad advertises % off.

```r
library(tesseract)
library(magick)
library(magrittr)
library(imagefx)
library(pdftools)
library(magick)

ad_test <- image_read("C:/Users/n8nel/Downloads/2752687-1200-100000.jpg")
text_ad_test <- ocr(ad_test)
cat(text_ad_test)

a1 <- image_crop(ad_test, geometry = geometry_area(width = 300,
                                                   height = 150,
                                                   x_off = 50,
                                                   y_off = 590))
a2 <- image_crop(ad_test, geometry = geometry_area(width = 300,
                                                   height = 125,
                                                   x_off = 50,
                                                   y_off = 975))
a3 <- image_crop(ad_test, geometry = geometry_area(width = 300,
                                                   height = 125,
                                                   x_off = 50,
                                                   y_off = 1325))
b1 <- image_crop(ad_test, geometry = geometry_area(width = 275,
                                                   height = 150,
                                                   x_off = 330,
                                                   y_off = 590))
b2 <- image_crop(ad_test, geometry = geometry_area(width = 300,
                                                   height = 125,
                                                   x_off = 330,
                                                   y_off = 960))
b3 <- image_crop(ad_test, geometry = geometry_area(width = 300,
                                                   height = 130,
                                                   x_off = 330,
                                                   y_off = 1310))
c1 <- image_crop(ad_test, geometry = geometry_area(width = 275,
                                                   height = 150,
                                                   x_off = 600,
                                                   y_off = 575))
c2 <- image_crop(ad_test, geometry = geometry_area(width = 275,
                                                   height = 125,
                                                   x_off = 620,
                                                   y_off = 975))
c3 <- image_crop(ad_test, geometry = geometry_area(width = 300,
                                                   height = 150,
                                                   x_off = 620,
                                                   y_off = 1300))
d1 <- image_crop(ad_test, geometry = geometry_area(width = 275,
```

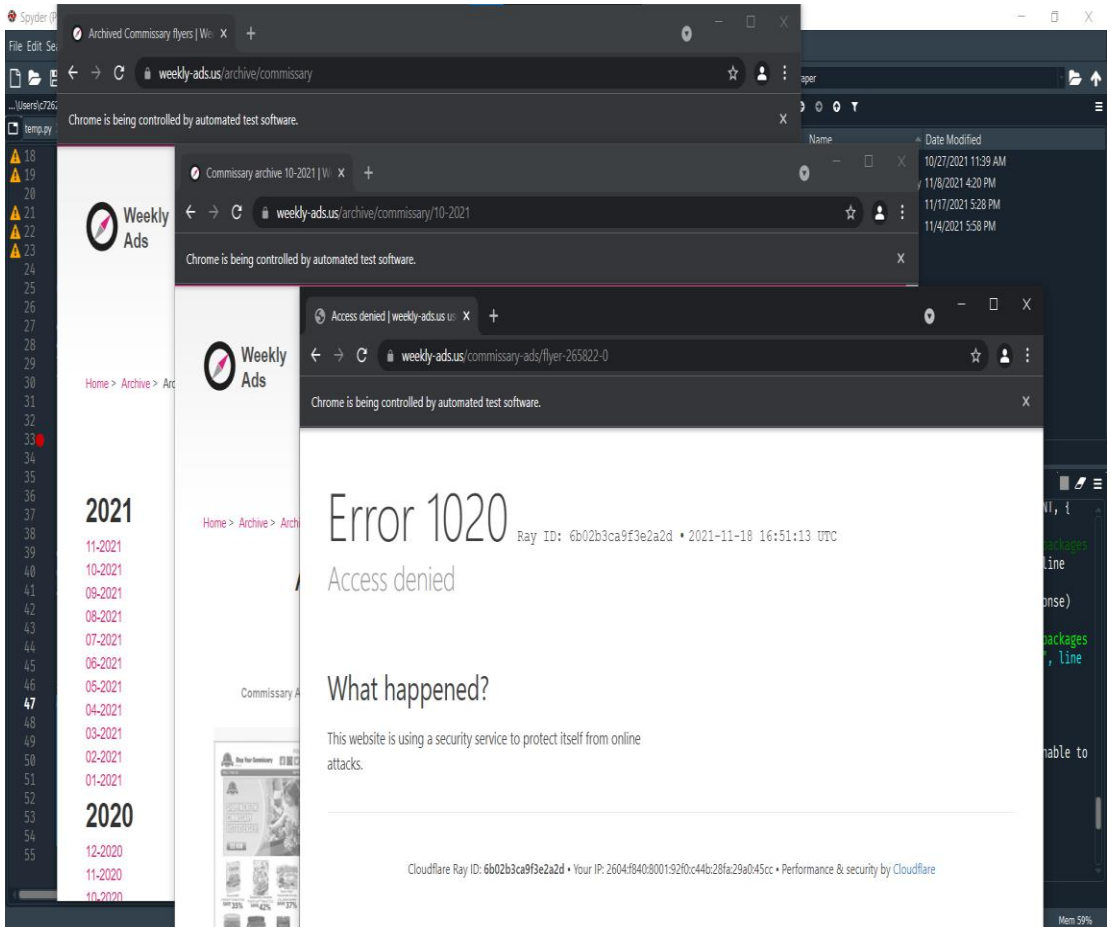
```r
image_write(ad_test, "2752687-1200-100000.pdf", format = "pdf")

# based on https://www.r-bloggers.com/2017/08/tesseract-and-magick-high-qu

ad_textProcessed <- ad_test %>%
  image_resize("2000") %>%
  image_convert(colorspace = "gray") %>%
  image_trim()

ad_textOCR <- ad_textProcessed %>%
  image_ocr()

cat(ad_textOCR)


ad_textOCRData <- ad_textProcessed %>%
  image_ocr_data()

print(ad_textOCRData, n = 10)


### - this one is the best

library(pdftools)

ad_textOCR2 <- pdf_ocr_text("2752687-1200-100000.pdf")

cat(ad_textOCR2)
```

Image Processing Code

Data Collection

- Weekly-ads.us continues to thwart our efforts at scraping.
- Have successfully downloaded single jpeg with script, but implementing loops results in ip blocking.
- Using multiple web browsers helped but didn’t get us all the way.
- Will be attempting to automate tor to randomize ip address throughout scraping process.

```python
url = "https://weekly-ads.us/archive/commissary"

driver = webdriver.Chrome() # link to web driver connected to chromium
driver.get(url)  # access site
links = driver.find_elements_by_tag_name('a')
link_urls = [] # initialize list of urls to target
regex = "^\d{2}-\d{4}$" # regex pattern for mm-yyyy date format

for link in links: # check each link in list of all links
    if bool(search(regex, link.text)): # If link follows date format
        link_urls.append(link.get_attribute('href')) # Then add url to url list



driver2 = webdriver.Chrome() # initializing multiple drivers prevents IP blocking
driver2.get(link_urls[1]) # follow first link that follows date format
ad_links = driver2.find_elements_by_xpath("//div[@class = 'leaflet-detail']/a[starts-with(@href, '/commissary-ads/flyer-')][1]")

for link in ad_links:
    link.get_attribute('href')

# Single Ad Multiple Pages for each page
driver3 = webdriver.Chrome()
image_urls = [] # initialize list of url's leading to jpegs to be downloaded in bash

for link in ad_links:
    driver3.implicitly_wait(0.5) #rapid requests trigger IP blocking, wait .5 seconds
    driver3.get(link.get_attribute('href')) # retrieve url to jpeg
    driver3.implicitly_wait(0.5)
    image_urls.append(driver3.find_element_by_id("leaflet").get_attribute('src'))# ad to jpeg's url to list

```



Data Collection Code