

ADVENTURES IN DATA MUNGING

Computers and Consumer Reports

GETTING THE DATA

- Selenium




- Scrapy Spider





- Selenium


- Site organization:


Ratings & recommended computers




Laptops (128)
Laptop reliability has been mostly undistinguished. Apple had the best technical support, so Apple owners are far more likely to have a positive tech-support experience than those with Windows computers. These **tests** are made by [Consumerreports](#).
[Recommended laptops](#) 
[Laptop Ratings](#)



Desktop computers (40)
Apple was the most reliable among desktop brands. It also had the best technical support, so Apple owners are far more likely to have a positive tech-support experience than those with Windows computers.
[Recommended desktop computers](#) 
[Desktop computer Ratings](#)



Chromebooks (19)
Recommended models are standout choices with high scores. They include CR Best Buys, which offer exceptional value. (Occasionally, high-scoring models are not recommended due to their Brand Repair History or other issues.) When narrowing your... [More](#)
[Recommended chromebooks](#) 
[Chromebook Ratings](#)

Level I:

Product Type

☒ Laptops
☐ Desktop computers
☐ Chromebooks




CR Recommends

☐ Recommended Only

Brand



☒ All
☐ Acer

I-30 of 174 Tested Models Shown Sort


Model Name		Price Range
	Acer Aspire E5-574-53QS Subscribe Now!	Price: \$390.00
	Acer Aspire ES1-571-P1MG Subscribe Now!	Price: \$315.00
	Acer Aspire One	Price: \$165.00


SITE LEVEL II

- Sometimes a Link is not just a link:
 - <https://www.consumerreports.org/products/laptop/acer-aspire-e5-574-53qs-385910/specs>
 - <https://www.consumerreports.org/products/laptop/acer-aspire-e5-574-53qs-385910/user-reviews>

Overview	Ratings 	Brand Reliability 	User Reviews	Specs
Specs				
Display size (in.)	15.6			
Operating system (as tested)	Windows 10 Home			
Processor	Intel Core i5-6200U 2.3GHz			
Memory (GB)	4			
Storage (GB)	1000			
Video memory (MB)	N/A			
LCD resolution	1366 x 768			
Video adapter	Intel HD Graphics 520			
Optical drive	CD/DVD burner			

Overview

Ratings 

Brand Reliability 

User Reviews

Specs

User Reviews

Write the First Review

Price & Shop

Amazon	\$459.99	In stock	<div>Shop</div> <div>At Amazon</div>
--------	----------	----------	--------------------------------------

Provided by eBay Commerce Network and Amazon, Consumer Reports Price & Shop makes it easy to find the right product from a variety of online retailers. Clicking "Shop" will take you to the retailer's website to shop for this product. Please note that Consumer Reports collects fees from both eBay Commerce Network and Amazon for referring users. We use 100% of these fees to fund our testing programs.

WHAT WORKS FOR ONE PART OF A SITE DOESN'T ALWAYS WORK FOR ANOTHER

Selenium

```
try:
    time.sleep(2)
    button.click()
    time.sleep(2)
except WebDriverException as wde:
    # path to exception: selenium.common.exceptions
    # this seemed to work in verizon test even though it does not make sense
    # if data is missing on real attempt, then we can look into wait / exception cases

    print("ButtonClick code Exception: You have encountered a WebDriverException:")
    print(wde)
    print(type(wde))
    index = index + 1
    continue
```



Scrapy Spider

```
specs = response.xpath('//div[@class="product-model-spec-container"]//div[@class="row"]')

for spec in specs:
    spec_value = spec.xpath('.//div[@class="col-lg-7 col-md-7 col-sm-6 col-xs-12 product-model-spec-item-value"]/text()').extract_first().strip()

    spec_label = spec.xpath('.//span[@class="product-model-spec-item-tooltip-text"]/strong/text()').extract_first().strip()
    spec_label = re.sub(r'[\s\(\)-]+', '_', spec_label)

    # model = response.xpath('//a[@class="product-model-eyebrow-link visible-xs"]/@data-modelname').extract_first()
    model = response.xpath('//div[@class="product-model-name-container"]/h1/text()').extract()[1].strip()
    # if model == "" or model == " " or model == np.nan:
    #     model = "__NULL__"

    # brand = response.xpath('//a[@class="product-model-eyebrow-link visible-xs"]/@data-brandname').extract_first()
    brand = response.xpath('//div[@class="product-model-name-container"]//strong/text()').extract_first()
    # if brand == "" or brand == " " or model == np.nan:
    #     brand = "__NULL__"

    prod_class = response.url.split('/')[4] # request holds original URL passed into this function
```

Selenium Again

```
for url in start_urls: # True to run, set to index < 10 or something to test
    try:
        driver.get(url)
        print("Scraping Page number " + str(index))
        print("Current url: " + url)
        button_xpath = '//*[@id="user-reviewsanchor"]'
        print("Attempting To Click Load Button: ", button_xpath)
        try:
            button = driver.find_element_by_xpath(button_xpath)
            button.click()
        except WebDriverException as wde:
            print("TabATagClick code Problem: You have encountered a WebDriverException")
            print(wde)
            print(type(wde))
            index = index + 1
            wde_count += 1. # simple count tells us how many times this happens
            continue

    wait_review = WebDriverWait(driver, 5)
    reviews = wait_review.until(EC.presence_of_all_elements_located((By.XPATH,
        '//div[@class="col-lg-12 col-md-12 col-sm-12 col-xs-12"]'))))
```



MODULAR ITERATIVE APPROACH

- Get All The Data Up Front
- Some Transformation with Scrapers for Some Known Patterns
- Create CSV Spreadsheet Each Step Of The Way
- Load into R or Jupyter Module – Whichever One “Feels Faster”
- Generate Sheet of Data
- Use Data In Analysis
- Go Back to Source and Do It All Again
- You Don’t Build Everything You Want But You Do Build Something
- Each Sheet Along The Way Allows You to Pick Up Where You Left Off Without Rerunning Code

[illegible]

Most Frequent Words - Positive Reviews

Word	Frequency
comput	44
use	40
work	39
laptop	33
back	26
keyboard	24
issu	24
hour	22
window	21
problem	20
will	20
time	19
get	18
screen	18
repair	18
buy	18
year	18
tri	18
key	17
support	17

WORD CLOUDS: REVIEW DATA

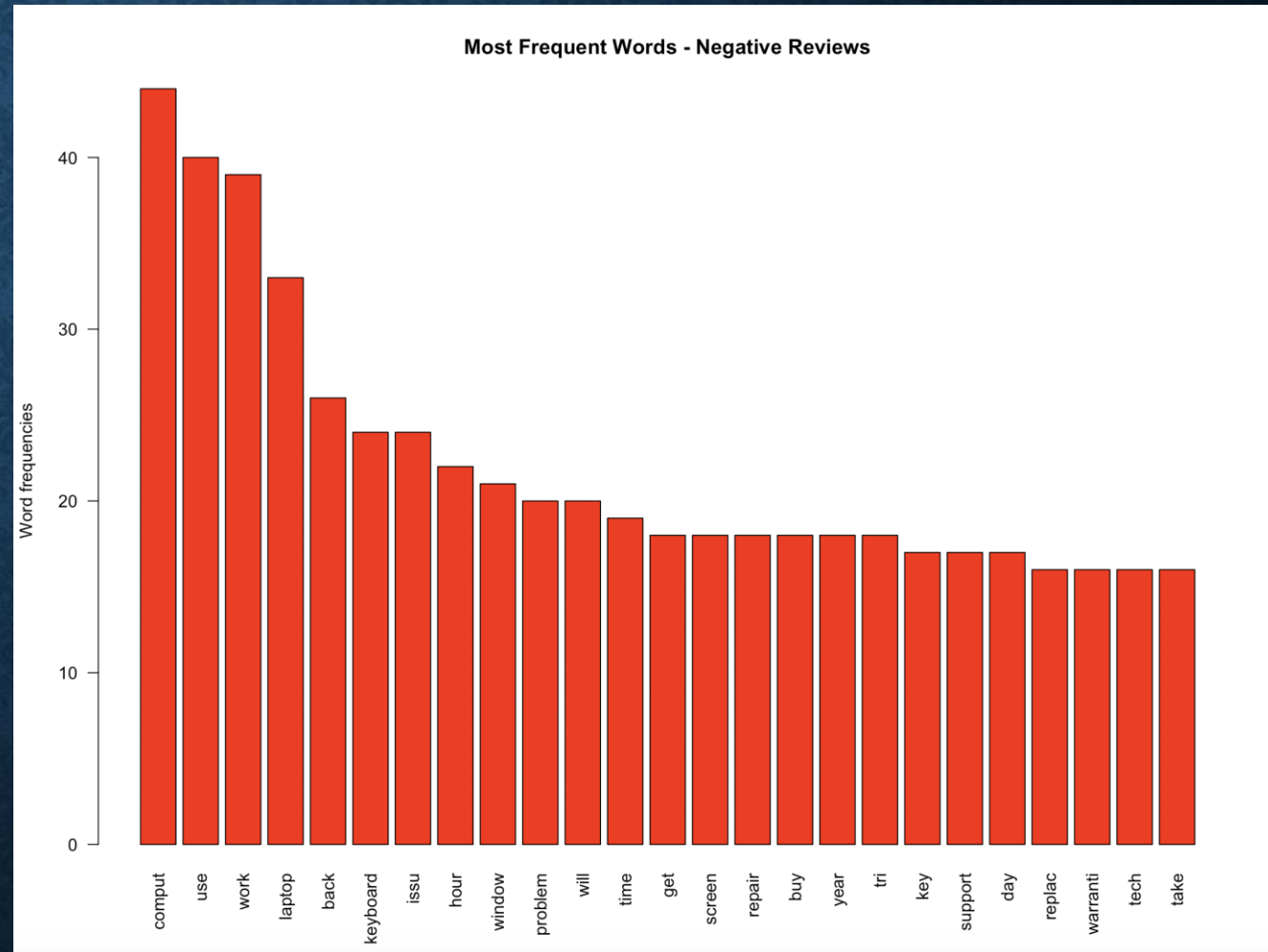
NEGATIVE REVIEWS

```
set.seed(1234)
wordcloud(words = d2$word, freq = d2$freq, min.freq = 1,
          max.words=74, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"), scale=c(4,.5))
...

```



Top 25 By Frequency:



[illegible]

Most Frequent Words - All Reviews

Word	Frequency (approx.)
use	265
comput	215
work	135
laptop	135
screen	98
keyboard	92
appl	80
window	75
get	74
like	74
buy	73
time	73
issu	70
year	69
great	66
also	66
will	65
one	64
display	64
problem	64
can	64
drive	61
back	61
key	61
hour	59

ANALYSIS OF SPECIFICATIONS WITH MATPLOTLIB AND SEABORN

- A Jupyter Notebook Saved to HTML provides a convenient way to tell this story:
[“TMWP CR Spec TableAnalysis1.html”](#)
- High Level – We Look At:
 - New Features – Coverage by Consumer Reports
 - Little Bit About RAM and Price
 - 53 Variables were collected for 268 Desktop, Laptop, and Chromebook Products were collected, but each one, even with work performed inline by scrapers, still required work to prepare it for analysis.