Metrics of successful sites and companies

February 6, 2017

```
In [1]: #First we import the libraries we will need
        import urllib
        import urllib2
        import time
        import os
        from bs4 import BeautifulSoup
        import re
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [2]: #First of all we need to find all the name of the sites that belong to fort
        #The information needed from the below link
        url = "http://www.zyxware.com/articles/4344/list-of-fortune-500-companies-a
        list_company_number =[]
        list_company_name = []
        list_company_website = []
```

```
In [3]: #In order to extract the needed informations we will create 3 lists. The fi
        #second one will contain the name of the company and the 3rd one will conta
        #For achieving this purpose we will create a funstion that will in its turn
        #In order to know if the function worked we will ask it to return the first
        def websites (url):
            from time import time # I used it to see how much time it does to run t
            start = time ()
            browser = urllib2.build_opener()
            browser.addheaders = [('User-agent', 'Mozilla/5.0')]
            response = browser.open(url)# this might throw an exception if somethin
            myHTML = response.read()
            soup = BeautifulSoup(myHTML,"lxml")
            o = 0
            td_list =[]
            for row2 in soup.html.body.findAll('td'):
                td_list.insert(o, row2)
                o = o + 1
            a = 0
            b = 1
            c = 2
            list_numbering = 0
            for i in range (0,500):
                num = str(td_list[a])
                company = str(td_list[b])
                site = str(td_list[c])
                c_num = re.findall('>(.+?)</td>',num)
                c_num = str(c_num[0])
                c_name = re.findall('>(.+?)</td>',company)
                c_name = str(c_name[0])
                c_site = re.findall('">(.+?)</a>',site)
                c_site = str(c_site[0])
                list_company_number.insert(list_numbering,c_num)
                list_company_name.insert(list_numbering,c_name)
                list_company_website.insert(list_numbering,c_site)
                a = a + 3
                b = b + 3
                c = c + 3
                list_numbering =  list_numbering + 1
            end = time ()
            duration = round (end - start, 1)
            minutes = round (duration /60, 1)
            print 'The lists are ready in ', duration, ' seconds'
            print 'The lists are ready in ', minutes, ' minutes'
```

```
In [4]: # After creating the function we should now test that it actually works co
        websites (url)
```

```
The lists are ready in  1.2  seconds
The lists are ready in  0.0  minutes
```

```
In [5]: #Try to validate each page url #pip install validators
        import validators
        nv = 0
        for num in range(len(list_company_website)):
            line = 'http://' + str(list_company_website[num])
            x = validators.url(line)
            if x != True:
                nv = nv +1
        print "The validation is complete! There were" , nv, "not valid pages"
```

```
The validation is complete! There were 0 not valid pages
```

```
In [6]: list500_sites = []
        list500_names = []
        list500_num = []
        list500_url = []
```

```
In [7]: #def list_company_HTML (list_company_website,list_company_name,start,end):
        import time
        browser2 = urllib2.build_opener()
        browser2.addheaders = [('User-agent', 'Mozilla/5.0')]
        for i in range (0,500):
            k = str(i + 1)
            lc = str(list_company_website[i])
            lc = lc.replace("'","")
            lc = lc.replace("[","")
            lc = lc.replace("]","")
            lcn = str(list_company_name[i])
            lcn = lcn.replace("'","")
            lcn = lcn.replace("[","")
            lcn = lcn.replace("]","")
            url2= 'http://' + lc
            list500_names.insert(i,lcn)
            list500_url.insert(i,lc)
            list500_num.insert(i,k)
            if i == 118 or i == 464 or i == 70:
                #These sites have a problem and the whole code is stacking
                #when I run it so we will thing of this site as a not downloadable
                list500_sites.insert(i,0)
                print ("The site " + str(i) + " has NOT been downloaded!")
            else:
                #an exception might be thrown, so the code should be in a try-excep
                try:
                    response2=browser2.open(url2)
                    print ("The site " + str(i) + " has been downloaded!")
                except Exception: # this describes what to do if an exception is th
                    list500_sites.insert(i,0)
                    print ("The site " + str(i) + " has NOT been downloaded from ex
                    continue
                    #if it goes into to exception it does not continue below
                myHTML2=response2.read()
                list500_sites.insert(i,myHTML2)
                #wait for 2 seconds
                time.sleep(2)
```

```
The site 0 has been downloaded!
The site 1 has been downloaded!
The site 2 has been downloaded!
The site 3 has been downloaded!
The site 4 has been downloaded!
The site 5 has been downloaded!
The site 6 has been downloaded!
The site 7 has been downloaded!
The site 8 has been downloaded!
The site 9 has been downloaded!
The site 10 has been downloaded!
The site 11 has been downloaded!
The site 12 has been downloaded!
The site 13 has been downloaded!
The site 14 has been downloaded!
The site 15 has NOT been downloaded from exception!
The site 16 has been downloaded!
The site 17 has been downloaded!
The site 18 has been downloaded!
The site 19 has been downloaded!
The site 20 has been downloaded!
The site 21 has been downloaded!
The site 22 has been downloaded!
The site 23 has been downloaded!
The site 24 has been downloaded!
The site 25 has been downloaded!
The site 26 has been downloaded!
The site 27 has been downloaded!
The site 28 has been downloaded!
The site 29 has been downloaded!
The site 30 has been downloaded!
The site 31 has been downloaded!
The site 32 has been downloaded!
The site 33 has been downloaded!
The site 34 has been downloaded!
The site 35 has been downloaded!
The site 36 has been downloaded!
The site 37 has been downloaded!
The site 38 has been downloaded!
The site 39 has been downloaded!
The site 40 has been downloaded!
The site 41 has been downloaded!
The site 42 has been downloaded!
The site 43 has been downloaded!
The site 44 has been downloaded!
The site 45 has been downloaded!
The site 46 has been downloaded!
The site 47 has been downloaded!
The site 48 has been downloaded!
The site 49 has been downloaded!
The site 50 has been downloaded!
The site 51 has been downloaded!
The site 52 has been downloaded!
The site 53 has been downloaded!
The site 54 has been downloaded!
The site 55 has been downloaded!
The site 56 has been downloaded!
The site 57 has been downloaded!
The site 58 has been downloaded!
The site 59 has been downloaded!
The site 60 has been downloaded!
The site 61 has been downloaded!
The site 62 has NOT been downloaded from exception!
The site 63 has been downloaded!
The site 64 has been downloaded!
The site 65 has been downloaded!
The site 66 has been downloaded!
The site 67 has been downloaded!
The site 68 has been downloaded!
The site 69 has been downloaded!
The site 70 has NOT been downloaded!
The site 71 has been downloaded!
The site 72 has been downloaded!
The site 73 has been downloaded!
The site 74 has been downloaded!
The site 75 has been downloaded!
The site 76 has been downloaded!
The site 77 has been downloaded!
The site 78 has been downloaded!
The site 79 has been downloaded!
The site 80 has been downloaded!
The site 81 has been downloaded!
The site 82 has been downloaded!
The site 83 has been downloaded!
The site 84 has been downloaded!
The site 85 has been downloaded!
The site 86 has been downloaded!
The site 87 has been downloaded!
The site 88 has been downloaded!
The site 89 has been downloaded!
The site 90 has NOT been downloaded from exception!
The site 91 has been downloaded!
The site 92 has been downloaded!
The site 93 has been downloaded!
The site 94 has been downloaded!
The site 95 has been downloaded!
The site 96 has been downloaded!
The site 97 has NOT been downloaded from exception!
The site 98 has been downloaded!
The site 99 has been downloaded!
The site 100 has been downloaded!
The site 101 has been downloaded!
The site 102 has been downloaded!
The site 103 has been downloaded!
The site 104 has been downloaded!
The site 105 has been downloaded!
The site 106 has been downloaded!
The site 107 has been downloaded!
The site 108 has been downloaded!
The site 109 has been downloaded!
The site 110 has been downloaded!
The site 111 has been downloaded!
The site 112 has been downloaded!
The site 113 has been downloaded!
The site 114 has been downloaded!
The site 115 has been downloaded!
The site 116 has been downloaded!
The site 117 has been downloaded!
The site 118 has NOT been downloaded!
The site 119 has been downloaded!
The site 120 has been downloaded!
The site 121 has been downloaded!
The site 122 has been downloaded!
The site 123 has been downloaded!
The site 124 has been downloaded!
The site 125 has been downloaded!
The site 126 has been downloaded!
The site 127 has been downloaded!
The site 128 has been downloaded!
The site 129 has been downloaded!
The site 130 has been downloaded!
The site 131 has been downloaded!
The site 132 has been downloaded!
The site 133 has been downloaded!
The site 134 has been downloaded!
The site 135 has NOT been downloaded from exception!
The site 136 has been downloaded!
The site 137 has been downloaded!
The site 138 has been downloaded!
The site 139 has been downloaded!
The site 140 has been downloaded!
The site 141 has NOT been downloaded from exception!
The site 142 has been downloaded!
The site 143 has been downloaded!
The site 144 has been downloaded!
The site 145 has been downloaded!
The site 146 has been downloaded!
The site 147 has been downloaded!
The site 148 has been downloaded!
The site 149 has been downloaded!
The site 150 has been downloaded!
The site 151 has been downloaded!
The site 152 has been downloaded!
The site 153 has been downloaded!
The site 154 has been downloaded!
The site 155 has been downloaded!
The site 156 has been downloaded!
The site 157 has been downloaded!
The site 158 has been downloaded!
The site 159 has been downloaded!
The site 160 has been downloaded!
The site 161 has NOT been downloaded from exception!
The site 162 has been downloaded!
The site 163 has been downloaded!
The site 164 has NOT been downloaded from exception!
The site 165 has been downloaded!
The site 166 has been downloaded!
The site 167 has been downloaded!
The site 168 has been downloaded!
The site 169 has been downloaded!
The site 170 has been downloaded!
The site 171 has been downloaded!
The site 172 has been downloaded!
The site 173 has been downloaded!
The site 174 has been downloaded!
The site 175 has been downloaded!
The site 176 has been downloaded!
The site 177 has been downloaded!
The site 178 has been downloaded!
The site 179 has been downloaded!
The site 180 has been downloaded!
The site 181 has been downloaded!
The site 182 has been downloaded!
The site 183 has been downloaded!
The site 184 has been downloaded!
The site 185 has been downloaded!
The site 186 has been downloaded!
The site 187 has been downloaded!
The site 188 has been downloaded!
The site 189 has been downloaded!
The site 190 has been downloaded!
The site 191 has been downloaded!
The site 192 has been downloaded!
The site 193 has been downloaded!
The site 194 has been downloaded!
The site 195 has NOT been downloaded from exception!
The site 196 has been downloaded!
The site 197 has been downloaded!
The site 198 has been downloaded!
The site 199 has been downloaded!
The site 200 has been downloaded!
The site 201 has been downloaded!
The site 202 has been downloaded!
The site 203 has been downloaded!
The site 204 has been downloaded!
The site 205 has been downloaded!
The site 206 has been downloaded!
The site 207 has been downloaded!
The site 208 has been downloaded!
The site 209 has been downloaded!
The site 210 has been downloaded!
The site 211 has been downloaded!
The site 212 has been downloaded!
The site 213 has been downloaded!
The site 214 has been downloaded!
The site 215 has been downloaded!
The site 216 has NOT been downloaded from exception!
The site 217 has been downloaded!
The site 218 has been downloaded!
The site 219 has been downloaded!
The site 220 has been downloaded!
The site 221 has been downloaded!
The site 222 has been downloaded!
The site 223 has been downloaded!
The site 224 has been downloaded!
The site 225 has been downloaded!
The site 226 has been downloaded!
The site 227 has been downloaded!
The site 228 has NOT been downloaded from exception!
The site 229 has been downloaded!
The site 230 has been downloaded!
The site 231 has been downloaded!
The site 232 has been downloaded!
The site 233 has been downloaded!
The site 234 has been downloaded!
The site 235 has been downloaded!
The site 236 has been downloaded!
The site 237 has been downloaded!
The site 238 has been downloaded!
The site 239 has NOT been downloaded from exception!
The site 240 has been downloaded!
The site 241 has been downloaded!
The site 242 has NOT been downloaded from exception!
The site 243 has been downloaded!
The site 244 has been downloaded!
The site 245 has been downloaded!
The site 246 has been downloaded!
The site 247 has been downloaded!
The site 248 has been downloaded!
The site 249 has been downloaded!
The site 250 has been downloaded!
The site 251 has been downloaded!
The site 252 has been downloaded!
The site 253 has been downloaded!
The site 254 has been downloaded!
The site 255 has been downloaded!
The site 256 has been downloaded!
The site 257 has been downloaded!
The site 258 has been downloaded!
The site 259 has been downloaded!
The site 260 has been downloaded!
The site 261 has been downloaded!
The site 262 has been downloaded!
The site 263 has been downloaded!
The site 264 has been downloaded!
The site 265 has been downloaded!
The site 266 has been downloaded!
The site 267 has been downloaded!
The site 268 has been downloaded!
The site 269 has been downloaded!
The site 270 has been downloaded!
The site 271 has been downloaded!
The site 272 has been downloaded!
The site 273 has been downloaded!
The site 274 has been downloaded!
The site 275 has NOT been downloaded from exception!
The site 276 has been downloaded!
The site 277 has been downloaded!
The site 278 has been downloaded!
The site 279 has been downloaded!
The site 280 has been downloaded!
The site 281 has been downloaded!
The site 282 has been downloaded!
The site 283 has been downloaded!
The site 284 has been downloaded!
The site 285 has been downloaded!
The site 286 has been downloaded!
The site 287 has been downloaded!
The site 288 has been downloaded!
The site 289 has been downloaded!
The site 290 has been downloaded!
The site 291 has been downloaded!
The site 292 has been downloaded!
The site 293 has been downloaded!
The site 294 has been downloaded!
The site 295 has been downloaded!
The site 296 has been downloaded!
The site 297 has been downloaded!
The site 298 has been downloaded!
The site 299 has been downloaded!
The site 300 has been downloaded!
The site 301 has been downloaded!
The site 302 has been downloaded!
The site 303 has been downloaded!
The site 304 has been downloaded!
The site 305 has been downloaded!
The site 306 has NOT been downloaded from exception!
The site 307 has been downloaded!
The site 308 has been downloaded!
The site 309 has been downloaded!
The site 310 has been downloaded!
The site 311 has been downloaded!
The site 312 has been downloaded!
The site 313 has been downloaded!
The site 314 has been downloaded!
The site 315 has been downloaded!
The site 316 has been downloaded!
The site 317 has been downloaded!
The site 318 has been downloaded!
The site 319 has been downloaded!
The site 320 has been downloaded!
The site 321 has been downloaded!
The site 322 has been downloaded!
The site 323 has been downloaded!
The site 324 has been downloaded!
The site 325 has been downloaded!
The site 326 has NOT been downloaded from exception!
The site 327 has been downloaded!
The site 328 has been downloaded!
The site 329 has been downloaded!
The site 330 has been downloaded!
The site 331 has been downloaded!
The site 332 has been downloaded!
The site 333 has been downloaded!
The site 334 has been downloaded!
The site 335 has been downloaded!
The site 336 has been downloaded!
The site 337 has been downloaded!
The site 338 has been downloaded!
The site 339 has been downloaded!
The site 340 has been downloaded!
The site 341 has been downloaded!
The site 342 has been downloaded!
The site 343 has been downloaded!
The site 344 has been downloaded!
The site 345 has been downloaded!
The site 346 has been downloaded!
The site 347 has been downloaded!
The site 348 has been downloaded!
The site 349 has been downloaded!
The site 350 has been downloaded!
The site 351 has been downloaded!
The site 352 has been downloaded!
The site 353 has been downloaded!
The site 354 has been downloaded!
The site 355 has been downloaded!
The site 356 has been downloaded!
The site 357 has been downloaded!
The site 358 has been downloaded!
The site 359 has been downloaded!
The site 360 has been downloaded!
The site 361 has been downloaded!
The site 362 has been downloaded!
The site 363 has NOT been downloaded from exception!
The site 364 has been downloaded!
The site 365 has been downloaded!
The site 366 has been downloaded!
The site 367 has been downloaded!
The site 368 has been downloaded!
The site 369 has been downloaded!
The site 370 has been downloaded!
The site 371 has been downloaded!
The site 372 has been downloaded!
The site 373 has been downloaded!
The site 374 has been downloaded!
The site 375 has been downloaded!
The site 376 has been downloaded!
The site 377 has been downloaded!
The site 378 has been downloaded!
The site 379 has been downloaded!
The site 380 has been downloaded!
The site 381 has been downloaded!
The site 382 has been downloaded!
The site 383 has been downloaded!
The site 384 has been downloaded!
The site 385 has been downloaded!
The site 386 has been downloaded!
The site 387 has been downloaded!
The site 388 has been downloaded!
The site 389 has been downloaded!
The site 390 has been downloaded!
The site 391 has been downloaded!
The site 392 has been downloaded!
The site 393 has been downloaded!
The site 394 has been downloaded!
The site 395 has been downloaded!
The site 396 has been downloaded!
The site 397 has NOT been downloaded from exception!
The site 398 has been downloaded!
The site 399 has been downloaded!
The site 400 has been downloaded!
The site 401 has been downloaded!
The site 402 has been downloaded!
The site 403 has been downloaded!
The site 404 has been downloaded!
The site 405 has been downloaded!
The site 406 has been downloaded!
The site 407 has been downloaded!
The site 408 has been downloaded!
The site 409 has been downloaded!
The site 410 has been downloaded!
The site 411 has been downloaded!
The site 412 has been downloaded!
The site 413 has been downloaded!
The site 414 has NOT been downloaded from exception!
The site 415 has been downloaded!
The site 416 has been downloaded!
The site 417 has been downloaded!
The site 418 has been downloaded!
The site 419 has been downloaded!
The site 420 has been downloaded!
The site 421 has been downloaded!
The site 422 has been downloaded!
The site 423 has been downloaded!
The site 424 has been downloaded!
The site 425 has been downloaded!
The site 426 has been downloaded!
The site 427 has been downloaded!
The site 428 has been downloaded!
The site 429 has been downloaded!
The site 430 has been downloaded!
The site 431 has been downloaded!
The site 432 has been downloaded!
The site 433 has been downloaded!
The site 434 has been downloaded!
The site 435 has been downloaded!
The site 436 has been downloaded!
The site 437 has been downloaded!
The site 438 has been downloaded!
The site 439 has been downloaded!
The site 440 has been downloaded!
The site 441 has NOT been downloaded from exception!
The site 442 has been downloaded!
The site 443 has been downloaded!
The site 444 has been downloaded!
The site 445 has been downloaded!
The site 446 has been downloaded!
The site 447 has been downloaded!
The site 448 has been downloaded!
The site 449 has been downloaded!
The site 450 has been downloaded!
The site 451 has been downloaded!
The site 452 has been downloaded!
The site 453 has been downloaded!
The site 454 has been downloaded!
The site 455 has been downloaded!
The site 456 has been downloaded!
The site 457 has been downloaded!
The site 458 has been downloaded!
The site 459 has been downloaded!
The site 460 has been downloaded!
The site 461 has been downloaded!
The site 462 has been downloaded!
The site 463 has been downloaded!
The site 464 has NOT been downloaded!
The site 465 has been downloaded!
The site 466 has been downloaded!
The site 467 has been downloaded!
The site 468 has been downloaded!
The site 469 has been downloaded!
The site 470 has been downloaded!
The site 471 has been downloaded!
The site 472 has been downloaded!
The site 473 has been downloaded!
The site 474 has been downloaded!
The site 475 has been downloaded!
The site 476 has been downloaded!
The site 477 has been downloaded!
The site 478 has been downloaded!
The site 479 has been downloaded!
The site 480 has been downloaded!
The site 481 has been downloaded!
The site 482 has been downloaded!
The site 483 has been downloaded!
The site 484 has been downloaded!
The site 485 has been downloaded!
The site 486 has been downloaded!
The site 487 has been downloaded!
The site 488 has been downloaded!
The site 489 has been downloaded!
The site 490 has been downloaded!
The site 491 has been downloaded!
The site 492 has been downloaded!
The site 493 has been downloaded!
The site 494 has been downloaded!
The site 495 has been downloaded!
The site 496 has been downloaded!
The site 497 has been downloaded!
The site 498 has been downloaded!
The site 499 has been downloaded!
```

```
In [8]: #As we can see there is one site that hasn't been downloaded in order
        #to keep track of the sites that we could not download
        #we will create a new list that we will keep them all together there
        not_d = []
        not_d_n = []
        num = []
        def not_downloadables (list500_names,list500_sites):
            met = 0
            for i in range(len(list500_names)):
                if list500_sites[i] == 0:
                    ct = list500_names[i]
                    not_d.insert(met,ct)
                    not_d_n.insert(met,str(i))
                    num.insert(met,met)
                    met = met + 1
```

```
In [9]: #Now we will run the function to see which sites havent been downloaded
        not_downloadables (list500_names,list500_sites)
        d = {'company' : pd.Series(not_d, index=[num]),
             'number' : pd.Series(not_d_n, index=[num])}
        nd = pd.DataFrame(d)
        nd
```

```
Out[9]:                         company number
        0                     Fannie Mae     15
        1                   HCA Holdings     62
        2                       Best Buy     70
        3                           Nike     90
        4                        Tesoro      97
        5              Arrow Electronics    118
        6                     AutoNation    135
        7             Southwest Airlines    141
        8                       Southern    161
        9         American Electric Power    164
        10                  Office Depot    195
```

```
11                   PBF Energy    216
12          Consolidated Edison    228
13                  Toys “R” Us    239
14           Dominion Resources    242
15              Global Partners    275
16              PayPal Holdings    306
17                   News Corp.    326
18                     Williams    363
19       Auto-Owners Insurance    397
20               Tractor Supply    414
21  Old Republic International    441
22             St. Jude Medical    464
```

```
In [10]: empty=[]
         keyf = []
         flesch = []
         sentence =[]
         word = []
         unique_w = []
```

```
In [11]: import time # I used it to see how much time it does to run the function
         for num in range(0,500):
             site = list500_sites[num]
             line = list500_url[num]
             url_check = "http://www.webpagefx.com/tools/read-able/check.php?tab=Te
             browser = urllib2.build_opener()
             browser.addheaders = [('User-agent', 'Mozilla/5.0')]
             if site == 0 or num == 107:
                 print("Site", str(num), "is not validated from sites")
                 flesch.insert(num,"n/a")
                 sentence.insert(num,"n/a")
                 word.insert(num,"n/a")
                 unique_w.insert(num,"n/a")
             else:
                 try:
                     response = browser.open(url_check)
                 except Exception:
                     flesch.insert(num,"n/a")
                     sentence.insert(num,"n/a")
                     word.insert(num,"n/a")
                     unique_w.insert(num,"n/a")
                     print("Site", str(num), "is not validated from check")
                     continue
                 html_r = response.read()
                 check = str(html_r)
                 if check != empty:
                         soup = BeautifulSoup(check,"lxml")
                         o = 0
```

```
keyf = []
for row in soup.html.body.findAll('tr'):
    keyf.insert(o,row)
    o = o + 1
if keyf != empty:
        print("Site", str(num), "is validated")
        #Flesh measurement
        if keyf[0] != empty:
            readability = str(keyf[0])
            split1 = readability.split('>')
            readability2 = str(split1[4])
            split2 = readability2.split('<')
            readability3 = str(split2[0])
            flesch.insert(num,readability3)
        else:
            flesch.insert(num,"n/a")
            sentence.insert(num,"n/a")
            word.insert(num,"n/a")
            unique_w.insert(num,"n/a")
        #Number of sentences
        if keyf[6] != empty:
            sentences = str(keyf[6])
            splil = sentences.split('>')
            sentences2 = str(splil[4])
            spli2 = sentences2.split('<')
            sentences3 = str(spli2[0])
            sentence.insert(num,sentences3)
        else:
            flesch.insert(num,"n/a")
            sentence.insert(num,"n/a")
            word.insert(num,"n/a")
            unique_w.insert(num,"n/a")
        #Number of words
        if keyf[7] != empty:
            words = str(keyf[7])
            spl1 = words.split('>')
            words2 = str(spl1[4])
            spl2 = words2.split('<')
            words3 = str(spl2[0])
            word.insert(num,words3)
        else:
            flesch.insert(num,"n/a")
            sentence.insert(num,"n/a")
            word.insert(num,"n/a")
            unique_w.insert(num,"n/a")
        #No. of complex words
        if keyf[7] != empty:
            unique_ws = str(keyf[8])
```

```
                                     sp1 = unique_ws.split('>')
                                     unique_ws2 = str(sp1[4])
                                     sp2 = unique_ws2.split('<')
                                     unique_ws3 = str(sp2[0])
                                     unique_w.insert(num,unique_ws3)
                                 else:
                                     flesch.insert(num,"n/a")
                                     sentence.insert(num,"n/a")
                                     word.insert(num,"n/a")
                                     unique_w.insert(num,"n/a")
                         else:
                                 print("Site", str(num), "is not validated from che
                                 flesch.insert(num,"n/a")
                                 sentence.insert(num,"n/a")
                                 word.insert(num,"n/a")
                                 unique_w.insert(num,"n/a")
             time.sleep(2)
```

```
('Site', '0', 'is not validated from check 2')
('Site', '1', 'is not validated from check 2')
('Site', '2', 'is validated')
('Site', '3', 'is validated')
('Site', '4', 'is validated')
('Site', '5', 'is validated')
('Site', '6', 'is validated')
('Site', '7', 'is validated')
('Site', '8', 'is validated')
('Site', '9', 'is validated')
('Site', '10', 'is validated')
('Site', '11', 'is not validated from check 2')
('Site', '12', 'is validated')
('Site', '13', 'is validated')
('Site', '14', 'is validated')
('Site', '15', 'is not validated from sites')
('Site', '16', 'is validated')
('Site', '17', 'is validated')
('Site', '18', 'is validated')
('Site', '19', 'is validated')
('Site', '20', 'is validated')
('Site', '21', 'is validated')
('Site', '22', 'is validated')
('Site', '23', 'is validated')
('Site', '24', 'is validated')
('Site', '25', 'is validated')
('Site', '26', 'is validated')
('Site', '27', 'is validated')
('Site', '28', 'is validated')
('Site', '29', 'is validated')
```

```
('Site', '30', 'is validated')
('Site', '31', 'is validated')
('Site', '32', 'is validated')
('Site', '33', 'is not validated from check 2')
('Site', '34', 'is validated')
('Site', '35', 'is validated')
('Site', '36', 'is validated')
('Site', '37', 'is not validated from check 2')
('Site', '38', 'is validated')
('Site', '39', 'is validated')
('Site', '40', 'is validated')
('Site', '41', 'is validated')
('Site', '42', 'is validated')
('Site', '43', 'is validated')
('Site', '44', 'is validated')
('Site', '45', 'is validated')
('Site', '46', 'is validated')
('Site', '47', 'is validated')
('Site', '48', 'is validated')
('Site', '49', 'is validated')
('Site', '50', 'is validated')
('Site', '51', 'is validated')
('Site', '52', 'is validated')
('Site', '53', 'is validated')
('Site', '54', 'is validated')
('Site', '55', 'is validated')
('Site', '56', 'is validated')
('Site', '57', 'is validated')
('Site', '58', 'is not validated from check 2')
('Site', '59', 'is validated')
('Site', '60', 'is validated')
('Site', '61', 'is validated')
('Site', '62', 'is not validated from sites')
('Site', '63', 'is validated')
('Site', '64', 'is validated')
('Site', '65', 'is validated')
('Site', '66', 'is validated')
('Site', '67', 'is not validated from check 2')
('Site', '68', 'is validated')
('Site', '69', 'is validated')
('Site', '70', 'is not validated from sites')
('Site', '71', 'is validated')
('Site', '72', 'is validated')
('Site', '73', 'is validated')
('Site', '74', 'is validated')
('Site', '75', 'is validated')
('Site', '76', 'is validated')
('Site', '77', 'is validated')
```

```
('Site', '78', 'is validated')
('Site', '79', 'is validated')
('Site', '80', 'is validated')
('Site', '81', 'is validated')
('Site', '82', 'is not validated from check 2')
('Site', '83', 'is validated')
('Site', '84', 'is validated')
('Site', '85', 'is validated')
('Site', '86', 'is validated')
('Site', '87', 'is validated')
('Site', '88', 'is validated')
('Site', '89', 'is validated')
('Site', '90', 'is not validated from sites')
('Site', '91', 'is validated')
('Site', '92', 'is validated')
('Site', '93', 'is validated')
('Site', '94', 'is validated')
('Site', '95', 'is validated')
('Site', '96', 'is validated')
('Site', '97', 'is not validated from sites')
('Site', '98', 'is validated')
('Site', '99', 'is validated')
('Site', '100', 'is validated')
('Site', '101', 'is validated')
('Site', '102', 'is validated')
('Site', '103', 'is validated')
('Site', '104', 'is validated')
('Site', '105', 'is validated')
('Site', '106', 'is not validated from check')
('Site', '107', 'is not validated from sites')
('Site', '108', 'is validated')
('Site', '109', 'is validated')
('Site', '110', 'is validated')
('Site', '111', 'is validated')
('Site', '112', 'is validated')
('Site', '113', 'is validated')
('Site', '114', 'is validated')
('Site', '115', 'is validated')
('Site', '116', 'is validated')
('Site', '117', 'is validated')
('Site', '118', 'is not validated from sites')
('Site', '119', 'is validated')
('Site', '120', 'is validated')
('Site', '121', 'is validated')
('Site', '122', 'is validated')
('Site', '123', 'is validated')
('Site', '124', 'is validated')
('Site', '125', 'is not validated from check 2')
```

```
('Site', '126', 'is validated')
('Site', '127', 'is validated')
('Site', '128', 'is validated')
('Site', '129', 'is validated')
('Site', '130', 'is validated')
('Site', '131', 'is validated')
('Site', '132', 'is validated')
('Site', '133', 'is validated')
('Site', '134', 'is validated')
('Site', '135', 'is not validated from sites')
('Site', '136', 'is validated')
('Site', '137', 'is validated')
('Site', '138', 'is validated')
('Site', '139', 'is validated')
('Site', '140', 'is validated')
('Site', '141', 'is not validated from sites')
('Site', '142', 'is validated')
('Site', '143', 'is validated')
('Site', '144', 'is validated')
('Site', '145', 'is validated')
('Site', '146', 'is validated')
('Site', '147', 'is not validated from check 2')
('Site', '148', 'is validated')
('Site', '149', 'is validated')
('Site', '150', 'is validated')
('Site', '151', 'is validated')
('Site', '152', 'is validated')
('Site', '153', 'is validated')
('Site', '154', 'is validated')
('Site', '155', 'is not validated from check 2')
('Site', '156', 'is validated')
('Site', '157', 'is validated')
('Site', '158', 'is validated')
('Site', '159', 'is validated')
('Site', '160', 'is validated')
('Site', '161', 'is not validated from sites')
('Site', '162', 'is validated')
('Site', '163', 'is validated')
('Site', '164', 'is not validated from sites')
('Site', '165', 'is validated')
('Site', '166', 'is not validated from check 2')
('Site', '167', 'is validated')
('Site', '168', 'is validated')
('Site', '169', 'is validated')
('Site', '170', 'is validated')
('Site', '171', 'is not validated from check 2')
('Site', '172', 'is validated')
('Site', '173', 'is validated')
```

```
('Site', '174', 'is validated')
('Site', '175', 'is validated')
('Site', '176', 'is validated')
('Site', '177', 'is validated')
('Site', '178', 'is validated')
('Site', '179', 'is not validated from check 2')
('Site', '180', 'is validated')
('Site', '181', 'is validated')
('Site', '182', 'is validated')
('Site', '183', 'is validated')
('Site', '184', 'is validated')
('Site', '185', 'is validated')
('Site', '186', 'is validated')
('Site', '187', 'is validated')
('Site', '188', 'is validated')
('Site', '189', 'is validated')
('Site', '190', 'is validated')
('Site', '191', 'is validated')
('Site', '192', 'is validated')
('Site', '193', 'is validated')
('Site', '194', 'is validated')
('Site', '195', 'is not validated from sites')
('Site', '196', 'is validated')
('Site', '197', 'is validated')
('Site', '198', 'is validated')
('Site', '199', 'is validated')
('Site', '200', 'is validated')
('Site', '201', 'is validated')
('Site', '202', 'is validated')
('Site', '203', 'is validated')
('Site', '204', 'is validated')
('Site', '205', 'is validated')
('Site', '206', 'is validated')
('Site', '207', 'is validated')
('Site', '208', 'is validated')
('Site', '209', 'is not validated from check 2')
('Site', '210', 'is validated')
('Site', '211', 'is not validated from check 2')
('Site', '212', 'is validated')
('Site', '213', 'is validated')
('Site', '214', 'is validated')
('Site', '215', 'is validated')
('Site', '216', 'is not validated from sites')
('Site', '217', 'is validated')
('Site', '218', 'is validated')
('Site', '219', 'is validated')
('Site', '220', 'is validated')
('Site', '221', 'is validated')
```

```
('Site', '222', 'is validated')
('Site', '223', 'is validated')
('Site', '224', 'is validated')
('Site', '225', 'is not validated from check 2')
('Site', '226', 'is validated')
('Site', '227', 'is validated')
('Site', '228', 'is not validated from sites')
('Site', '229', 'is validated')
('Site', '230', 'is validated')
('Site', '231', 'is validated')
('Site', '232', 'is validated')
('Site', '233', 'is validated')
('Site', '234', 'is validated')
('Site', '235', 'is validated')
('Site', '236', 'is validated')
('Site', '237', 'is validated')
('Site', '238', 'is validated')
('Site', '239', 'is not validated from sites')
('Site', '240', 'is validated')
('Site', '241', 'is validated')
('Site', '242', 'is not validated from sites')
('Site', '243', 'is validated')
('Site', '244', 'is validated')
('Site', '245', 'is validated')
('Site', '246', 'is validated')
('Site', '247', 'is validated')
('Site', '248', 'is validated')
('Site', '249', 'is validated')
('Site', '250', 'is validated')
('Site', '251', 'is validated')
('Site', '252', 'is validated')
('Site', '253', 'is validated')
('Site', '254', 'is validated')
('Site', '255', 'is validated')
('Site', '256', 'is validated')
('Site', '257', 'is validated')
('Site', '258', 'is validated')
('Site', '259', 'is validated')
('Site', '260', 'is validated')
('Site', '261', 'is validated')
('Site', '262', 'is validated')
('Site', '263', 'is validated')
('Site', '264', 'is validated')
('Site', '265', 'is validated')
('Site', '266', 'is validated')
('Site', '267', 'is validated')
('Site', '268', 'is validated')
('Site', '269', 'is validated')
```

```
('Site', '270', 'is validated')
('Site', '271', 'is validated')
('Site', '272', 'is not validated from check 2')
('Site', '273', 'is validated')
('Site', '274', 'is validated')
('Site', '275', 'is not validated from sites')
('Site', '276', 'is validated')
('Site', '277', 'is validated')
('Site', '278', 'is validated')
('Site', '279', 'is validated')
('Site', '280', 'is validated')
('Site', '281', 'is validated')
('Site', '282', 'is not validated from check 2')
('Site', '283', 'is validated')
('Site', '284', 'is validated')
('Site', '285', 'is validated')
('Site', '286', 'is validated')
('Site', '287', 'is validated')
('Site', '288', 'is validated')
('Site', '289', 'is validated')
('Site', '290', 'is validated')
('Site', '291', 'is validated')
('Site', '292', 'is validated')
('Site', '293', 'is validated')
('Site', '294', 'is validated')
('Site', '295', 'is validated')
('Site', '296', 'is validated')
('Site', '297', 'is not validated from check 2')
('Site', '298', 'is validated')
('Site', '299', 'is validated')
('Site', '300', 'is validated')
('Site', '301', 'is validated')
('Site', '302', 'is validated')
('Site', '303', 'is validated')
('Site', '304', 'is validated')
('Site', '305', 'is validated')
('Site', '306', 'is not validated from sites')
('Site', '307', 'is validated')
('Site', '308', 'is validated')
('Site', '309', 'is validated')
('Site', '310', 'is validated')
('Site', '311', 'is validated')
('Site', '312', 'is validated')
('Site', '313', 'is validated')
('Site', '314', 'is validated')
('Site', '315', 'is validated')
('Site', '316', 'is validated')
('Site', '317', 'is validated')
```

```
('Site', '318', 'is validated')
('Site', '319', 'is validated')
('Site', '320', 'is validated')
('Site', '321', 'is validated')
('Site', '322', 'is validated')
('Site', '323', 'is validated')
('Site', '324', 'is validated')
('Site', '325', 'is validated')
('Site', '326', 'is not validated from sites')
('Site', '327', 'is validated')
('Site', '328', 'is validated')
('Site', '329', 'is validated')
('Site', '330', 'is validated')
('Site', '331', 'is validated')
('Site', '332', 'is validated')
('Site', '333', 'is validated')
('Site', '334', 'is validated')
('Site', '335', 'is validated')
('Site', '336', 'is validated')
('Site', '337', 'is validated')
('Site', '338', 'is validated')
('Site', '339', 'is validated')
('Site', '340', 'is validated')
('Site', '341', 'is validated')
('Site', '342', 'is validated')
('Site', '343', 'is validated')
('Site', '344', 'is validated')
('Site', '345', 'is validated')
('Site', '346', 'is validated')
('Site', '347', 'is validated')
('Site', '348', 'is not validated from check 2')
('Site', '349', 'is validated')
('Site', '350', 'is validated')
('Site', '351', 'is validated')
('Site', '352', 'is validated')
('Site', '353', 'is validated')
('Site', '354', 'is validated')
('Site', '355', 'is validated')
('Site', '356', 'is validated')
('Site', '357', 'is validated')
('Site', '358', 'is validated')
('Site', '359', 'is validated')
('Site', '360', 'is validated')
('Site', '361', 'is validated')
('Site', '362', 'is validated')
('Site', '363', 'is not validated from sites')
('Site', '364', 'is validated')
('Site', '365', 'is validated')
```

```
('Site', '366', 'is not validated from check 2')
('Site', '367', 'is validated')
('Site', '368', 'is validated')
('Site', '369', 'is not validated from check 2')
('Site', '370', 'is validated')
('Site', '371', 'is validated')
('Site', '372', 'is validated')
('Site', '373', 'is validated')
('Site', '374', 'is validated')
('Site', '375', 'is not validated from check 2')
('Site', '376', 'is validated')
('Site', '377', 'is validated')
('Site', '378', 'is validated')
('Site', '379', 'is validated')
('Site', '380', 'is validated')
('Site', '381', 'is validated')
('Site', '382', 'is validated')
('Site', '383', 'is not validated from check 2')
('Site', '384', 'is not validated from check 2')
('Site', '385', 'is validated')
('Site', '386', 'is validated')
('Site', '387', 'is validated')
('Site', '388', 'is validated')
('Site', '389', 'is not validated from check 2')
('Site', '390', 'is validated')
('Site', '391', 'is validated')
('Site', '392', 'is validated')
('Site', '393', 'is validated')
('Site', '394', 'is validated')
('Site', '395', 'is validated')
('Site', '396', 'is validated')
('Site', '397', 'is not validated from sites')
('Site', '398', 'is validated')
('Site', '399', 'is validated')
('Site', '400', 'is validated')
('Site', '401', 'is validated')
('Site', '402', 'is not validated from check 2')
('Site', '403', 'is validated')
('Site', '404', 'is not validated from check 2')
('Site', '405', 'is validated')
('Site', '406', 'is validated')
('Site', '407', 'is not validated from check 2')
('Site', '408', 'is validated')
('Site', '409', 'is validated')
('Site', '410', 'is validated')
('Site', '411', 'is validated')
('Site', '412', 'is validated')
('Site', '413', 'is validated')
```

```
('Site', '414', 'is not validated from sites')
('Site', '415', 'is validated')
('Site', '416', 'is validated')
('Site', '417', 'is validated')
('Site', '418', 'is validated')
('Site', '419', 'is validated')
('Site', '420', 'is validated')
('Site', '421', 'is validated')
('Site', '422', 'is validated')
('Site', '423', 'is validated')
('Site', '424', 'is validated')
('Site', '425', 'is validated')
('Site', '426', 'is validated')
('Site', '427', 'is validated')
('Site', '428', 'is validated')
('Site', '429', 'is validated')
('Site', '430', 'is validated')
('Site', '431', 'is validated')
('Site', '432', 'is not validated from check 2')
('Site', '433', 'is validated')
('Site', '434', 'is validated')
('Site', '435', 'is validated')
('Site', '436', 'is validated')
('Site', '437', 'is validated')
('Site', '438', 'is not validated from check 2')
('Site', '439', 'is validated')
('Site', '440', 'is validated')
('Site', '441', 'is not validated from sites')
('Site', '442', 'is validated')
('Site', '443', 'is not validated from check 2')
('Site', '444', 'is validated')
('Site', '445', 'is validated')
('Site', '446', 'is validated')
('Site', '447', 'is not validated from check 2')
('Site', '448', 'is validated')
('Site', '449', 'is validated')
('Site', '450', 'is validated')
('Site', '451', 'is validated')
('Site', '452', 'is validated')
('Site', '453', 'is validated')
('Site', '454', 'is validated')
('Site', '455', 'is validated')
('Site', '456', 'is validated')
('Site', '457', 'is validated')
('Site', '458', 'is validated')
('Site', '459', 'is validated')
('Site', '460', 'is validated')
('Site', '461', 'is validated')
```

```
('Site', '462', 'is validated')
('Site', '463', 'is validated')
('Site', '464', 'is not validated from sites')
('Site', '465', 'is validated')
('Site', '466', 'is not validated from check 2')
('Site', '467', 'is validated')
('Site', '468', 'is not validated from check 2')
('Site', '469', 'is validated')
('Site', '470', 'is validated')
('Site', '471', 'is validated')
('Site', '472', 'is validated')
('Site', '473', 'is validated')
('Site', '474', 'is validated')
('Site', '475', 'is validated')
('Site', '476', 'is validated')
('Site', '477', 'is validated')
('Site', '478', 'is validated')
('Site', '479', 'is validated')
('Site', '480', 'is validated')
('Site', '481', 'is validated')
('Site', '482', 'is validated')
('Site', '483', 'is validated')
('Site', '484', 'is validated')
('Site', '485', 'is validated')
('Site', '486', 'is validated')
('Site', '487', 'is validated')
('Site', '488', 'is validated')
('Site', '489', 'is validated')
('Site', '490', 'is validated')
('Site', '491', 'is validated')
('Site', '492', 'is validated')
('Site', '493', 'is validated')
('Site', '494', 'is validated')
('Site', '495', 'is validated')
('Site', '496', 'is validated')
('Site', '497', 'is validated')
('Site', '498', 'is validated')
('Site', '499', 'is validated')
```

```
In [12]: readability = []
```

```
In [13]: def readable (flesch):
             for i in range (len(flesch)):
                     f_n = flesch[i]
                     if f_n == "n/a":
                         readability.insert(i,"n/a")
                     else:
                         a = int(float(f_n))
                         if a > 90:
                             readability.insert(i,"Very easy")
                         elif a > 80:
                             readability.insert(i,"Easy")
                         elif a > 70:
                             readability.insert(i,"Fairly easy")
                         elif a > 60:
                             readability.insert(i,"Standard")
                         elif a > 50:
                             readability.insert(i,"Fairly difficult")
                         elif a > 30:
                             readability.insert(i,"Difficult")
                         else:
                             readability.insert(i,"Very Confusing")
             print "The function is completed!"
```

```
In [14]: readable (flesch)
```

```
The function is completed!
```

```
In [15]: d1 = {'company' : pd.Series(list500_names, index=[list500_num]),
               'url' : pd.Series(list500_url, index=[list500_num]),
               'Readability' : pd.Series(readability, index=[list500_num]),
               'Flesh_Mesaure' : pd.Series(flesch,index=[list500_num]),
         'Sentences' : pd.Series(sentence, index=[list500_num]),
         'Words' : pd.Series(word, index=[list500_num]),
         'Unique words' : pd.Series(unique_w, index=[list500_num])}
         fre = pd.DataFrame(d1)
         fre #we see the first 3 in the data frame
```

```
Out[15]:     Flesh_Mesaure       Readability Sentences Unique words Words  \
         1             n/a               n/a       n/a          n/a   n/a
         2             n/a               n/a       n/a          n/a   n/a
         3            59.7  Fairly difficult       119           25   279
         4            55.6  Fairly difficult        27           39   197
         5              25    Very Confusing       229          295   799
         6            53.4  Fairly difficult        37           59   326
         7              35         Difficult       183          249   818
         8            32.8         Difficult       231           74   600
         9            82.1              Easy       258           80   760
         10           71.5       Fairly easy       374          176  1267
         11           51.2  Fairly difficult        80           43   192
         12            n/a               n/a       n/a          n/a   n/a
         13           46.1         Difficult        55           13    89
         14           20.4    Very Confusing       348          311   926
         15           70.3          Standard        20            4    43
         16            n/a               n/a       n/a          n/a   n/a
```

```
17           43.5         Difficult        74          118   500
18           73.3       Fairly easy        11           10    56
19             37         Difficult       223          121   584
20           57.9  Fairly difficult       273          149   763
21           39.5         Difficult       482          289  1316
22           47.5         Difficult        76           86   322
23           45.4         Difficult       291          244  1343
24           53.7  Fairly difficult       183          116   545
25           42.7         Difficult         1            4    25
26           50.4         Difficult       290          227  1534
27             59  Fairly difficult       346          237  1515
28           65.3          Standard       254           56   723
29           66.7          Standard         5            4    29
30           47.8         Difficult        99           81   383
..            ...               ...       ...          ...   ...
471           5.2    Very Confusing       170          175   417
472       -3422.4    Very Confusing         2            2     7
473          36.5         Difficult       477          206  1347
474          39.9         Difficult        32           28   105
475          42.8         Difficult        92           88   363
476          47.4         Difficult       126          126   529
477          57.6  Fairly difficult        66           28   189
478          46.8         Difficult        79          108   447
479          54.6  Fairly difficult        89           87   465
480          32.9         Difficult       172          125   578
481          41.9         Difficult       281          107   725
482          36.2         Difficult       654          818  2929
483          36.7         Difficult       122          132   437
484            65          Standard       140           27   359
485           5.5    Very Confusing       641          580  1534
486          58.8  Fairly difficult       199          208  1059
487            52  Fairly difficult       232          132   673
488            58  Fairly difficult       106           70   359
489          40.3         Difficult        89           73   321
490          38.8         Difficult       216          302  1232
491          33.1         Difficult       337          143   651
492          36.5         Difficult       131          124   414
493            21    Very Confusing       366          452  1312
494          42.9         Difficult        54          109   435
495          65.1          Standard       459           70  1059
496          47.1         Difficult       129           84   380
497          45.5         Difficult       325          333  1529
498          64.5          Standard        47           39   252
499          53.1  Fairly difficult       115          151   669
500            81              Easy        33           10   113

                                    company                                 ur
1                                   Walmart                 www.walmart.co
```

```
2                               Exxon Mobil              www.exxonmobil.co
3                                     Apple                   www.apple.co
4                        Berkshire Hathaway       www.berkshirehathaway.co
5                                  McKesson                www.mckesson.co
6                        UnitedHealth Group       www.unitedhealthgroup.co
7                                CVS Health               www.cvshealth.co
8                            General Motors                      www.gm.co
9                                Ford Motor                    www.ford.co
10                                 AT&amp;T                     www.att.co
11                         General Electric                      www.ge.co
12                        AmerisourceBergen       www.amerisourcebergen.co
13                                  Verizon                 www.verizon.co
14                                  Chevron                 www.chevron.co
15                                   Costco                  www.costco.co
16                               Fannie Mae               www.fanniemae.co
17                                   Kroger             www.thekrogerco.co
18                               Amazon.com                  www.amazon.co
19                 Walgreens Boots Alliance  www.walgreensbootsalliance.co
20                                       HP                      www.hp.co
21                          Cardinal Health                www.cardinal.co
22                  Express Scripts Holding         www.express-scripts.co
23                        J.P. Morgan Chase           www.jpmorganchase.co
24                                   Boeing                  www.boeing.co
25                                Microsoft               www.microsoft.co
26                    Bank of America Corp.           www.bankofamerica.co
27                              Wells Fargo              www.wellsfargo.co
28                               Home Depot               www.homedepot.co
29                                Citigroup               www.citigroup.co
30                              Phillips 66              www.phillips66.co
..                                      ...                             ..
471                      Arthur J. Gallagher                     www.ajg.co
472               Host Hotels &amp; Resorts              www.hosthotels.co
473                                  Ashland                 www.ashland.co
474                      Insight Enterprises                 www.insight.co
475                   Energy Future Holdings    www.energyfutureholdings.co
476                                   Markel              www.markelcorp.co
477                                Essendant               www.essendant.co
478                                CH2M Hill                    www.ch2m.co
479  Western &amp; Southern Financial Group         www.westernsouthern.co
480                            Owens Corning           www.owenscorning.co
481                            S&amp;P Global                www.spglobal.co
482                  Raymond James Financial           www.raymondjames.co
483                                 NiSource                www.nisource.co
484                                   Airgas                  www.airgas.co
485                           ABM Industries                     www.abm.co
486                 Citizens Financial Group            www.citizensbank.co
487              Booz Allen Hamilton Holding              www.boozallen.co
488                     Simon Property Group                   www.simon.co
```

```
489                                  Domtar                   www.domtar.co
490                        Rockwell Collins          www.rockwellcollins.co
491                            Lam Research              www.lamresearch.co
492                                  Fiserv                   www.fiserv.co
493                          Spectra Energy            www.spectraenergy.co
494                                 Navient                  www.navient.co
495                                Big Lots                  www.biglots.co
496            Telephone &amp; Data Systems                   www.tdsinc.co
497                First American Financial                  www.firstam.co
498                                     NVR                   www.nvrinc.co
499                    Cincinnati Financial                   www.cinfin.co
500                       Burlington Stores         www.burlingtonstores.co

[500 rows x 7 columns]
```

```
In [16]: #Retreiving the social media from each site
         #First create empty lists for the ones that
         #we will need to calculate
         sm_f = []
         sm_t = []
         sm_i = []
         sm_p = []
         sm_y = []
         sm_l = []
         sm_nm = []
         nm = []
         sm_url = []
```

```
In [17]: #Then create a function that will feel in those
         #lists so as to make the data frame later on
         def socialmedia (list500_sites,list500_names,list500_url):
             from time import time
             # I used it to see how much time it does to run the function
             start = time ()
             for i in range(len(list500_names)):
                     myHTML = list500_sites[i]
                     sm = ['facebook.com','twitter.com',
                           'instagram.com','pinterest.com',
                           'youtube.com','linkedin.com']
                     if myHTML == 0:
                         sm_nm.insert(i,list500_names[i])
                         nm.insert(i,i)
                         sm_url.insert(i,list500_url[i])
                         sm_f.insert(i,'n/a')
                         sm_t.insert(i,'n/a')
                         sm_i.insert(i,'n/a')
                         sm_p.insert(i,'n/a')
                         sm_y.insert(i,'n/a')
```

```
                sm_l.insert(i,'n/a')
            else:
                for index in range(len(sm)):
                    x = sm[index]
                    social = re.findall(x,myHTML)
                    if (len(social) > 0):
                        if x == 'facebook.com':
                            answerf = 'TRUE'
                        if x == 'twitter.com':
                            answert = 'TRUE'
                        if x == 'instagram.com':
                            answeri = 'TRUE'
                        if x == 'pinterest.com':
                            answerp = 'TRUE'
                        if x == 'youtube.com':
                            answery = 'TRUE'
                        if x =='linkedin.com':
                            answerl = 'TRUE'
                    else:
                         if x == 'facebook.com':
                            answerf = 'FALSE'
                         if x == 'twitter.com':
                            answert = 'FALSE'
                         if x == 'instagram.com':
                            answeri = 'FALSE'
                         if x == 'pinterest.com':
                            answerp = 'FALSE'
                         if x == 'youtube.com':
                            answery = 'FALSE'
                         if x =='linkedin.com':
                            answerl = 'FALSE'
                sm_nm.insert(i,list500_names[i])
                nm.insert(i,i)
                sm_url.insert(i,list500_url[i])
                sm_f.insert(i,answerf)
                sm_t.insert(i,answert)
                sm_i.insert(i,answeri)
                sm_p.insert(i,answerp)
                sm_y.insert(i,answery)
                sm_l.insert(i,answerl)
    end = time ()
    duration = round (end - start, 3)
    minutes = round (duration /60, 1)
    print 'The lists are completed in ', minutes, ' minutes'
    print 'The lists are ready in ', duration, ' seconds'
```

```
In [18]: #Now we will run the function for the 25 first sites for starters
         socialmedia (list500_sites,list500_names,list500_url)
```

```
The lists are completed in  0.0  minutes
The lists are ready in  0.26  seconds
```

```
In [19]: #Finally we create the data frame with the elements we found
         d2 = {'company' : pd.Series(sm_nm, index=[nm]),
              'facebook' : pd.Series(sm_f, index=[nm]),
               'twitter' : pd.Series(sm_t, index=[nm]),
              'instagram' : pd.Series(sm_i, index=[nm]),
               'pinterest' : pd.Series(sm_p, index=[nm]),
              'youtube' : pd.Series(sm_y, index=[nm]),
               'linkedin' : pd.Series(sm_l, index=[nm]),}
         social_media = pd.DataFrame(d2)
         social_media.tail(3) #we see the first 3 in the data frame
```

```
Out[19]:                    company facebook instagram linkedin pinterest twitter  \
         497                    NVR     TRUE      TRUE    FALSE      TRUE    TRUE
         498  Cincinnati Financial     TRUE     FALSE    FALSE     FALSE   FALSE
         499     Burlington Stores     TRUE      TRUE    FALSE      TRUE    TRUE

             youtube
         497    TRUE
         498   FALSE
         499    TRUE
```

```
In [20]: #Create the lists we will need for the data frame
         l_nm = []
         l_ex = []
         l_in = []
         l_t = []
         nm = []
         l_url = []
```

```
In [21]: #create the function that will calculate the different type of links
         def links (list500_sites,list500_names,list500_url):
             from time import time
             # I used it to see how much time it does to run the function
             start = time ()
             for num in range(len(list500_names)):
                     myHTML = list500_sites[num]
                     if myHTML == 0:
                         l_nm.insert(num,list500_names[num])
                         l_ex.insert(num,'n/a')
                         l_t.insert(num,'n/a')
                         l_in.insert(num,'n/a')
                         nm.insert(num,num)
                     else:
                         href = re.findall('href',myHTML)
                         external = re.findall('href="https:',myHTML)
                         ex = (len(external))
                         alllinks = (len(href))
                         internal =  (len(href) - len(external))
                         l_nm.insert(num,list500_names[num])
                         l_ex.insert(num,ex)
                         l_t.insert(num,alllinks)
                         l_in.insert(num,internal)
                         nm.insert(num,num)
             end = time ()
             duration = round (end - start, 3)
             minutes = round (duration /60, 1)
             print 'The lists are ready in ', minutes, ' minutes'
             print 'The lists are ready in ', duration, ' seconds'
```

```
In [22]: #Run the function in order to find the external,
         #internal and total links of each site
         #For now we are running for the first 25 sites only
         links (list500_sites,list500_names,list500_url)
```

```
The lists are ready in  0.0  minutes
The lists are ready in  0.085  seconds
```

```
In [23]: #Create a dataframe so as to be able to see
         #the results of the function we run
         d3 = {'company' : pd.Series(l_nm, index=[nm]),
               'external' : pd.Series(l_ex, index=[nm]),
               'internal' : pd.Series(l_in, index=[nm]),
              'total links' : pd.Series(l_t, index=[nm])}
         sites_links = pd.DataFrame(d3)
         sites_links.tail(3) #we see the first 3 in the data frame
```

```
Out[23]:                   company external internal total links
         497                   NVR        5       29          34
         498  Cincinnati Financial        3       74          77
         499     Burlington Stores       16      149         165
```

```
In [24]: #The initial lists we will need in order
         #to calculate the loading time
         lt_nm = []
         lt_time = []
         nm = []
         lt_url = []
```

```
In [25]: #the function that will calculate the loading time
         def loadtime (list_company_website,list500_names,list500_url):
             from time import time
             browser2 = urllib2.build_opener()
             browser2.addheaders = [('User-agent', 'Mozilla/5.0')]
```

```
    for num in range(len(list500_names)):
        lc = str(list_company_website[num])
        lc = lc.replace("'","")
        lc = lc.replace("[","")
        lc = lc.replace("]","")
        url2 = 'http://' + lc
        if num == 118 or num == 464 or num == 70:
            #The site 118(119) has a problem and the whole code
            #is stacking when I run it so we will thing of this
            #site as a not downloadable
            lt_nm.insert(num,list500_names[num])
            lt_time.insert(num,'n/a')
            nm.insert(num,num)
            lt_url.insert(num,list500_url[num])
        else:
            try:
                response2 = browser2.open(url2)
            except Exception:
                lt_time.insert(num,'n/a')
                lt_nm.insert(num,list500_names[num])
                nm.insert(num,num)
                print ("The site " + str(num)+ " has NOT been loaded!")
                continue
            start_time = time()
            myHTML2 = response2.read()
            end_time = time()
            response2.close()
            l_t = round(end_time-start_time, 3)
            #in order to be more readable we rounded the time
            loadt = str(l_t)
            lt_nm.insert(num,list500_names[num])
            lt_time.insert(num,loadt)
            nm.insert(num,num)
            lt_url.insert(num,list500_url[num])
            #print ("The site " + str(num) + " has been loaded!")
    print "The function is completed!"
```

```
In [26]: #running the function for the first 25 sites
         loadtime (list_company_website,list500_names,list500_url)
```

```
The site 15 has NOT been loaded!
The site 62 has NOT been loaded!
The site 90 has NOT been loaded!
The site 97 has NOT been loaded!
The site 135 has NOT been loaded!
The site 141 has NOT been loaded!
The site 161 has NOT been loaded!
The site 164 has NOT been loaded!
The site 195 has NOT been loaded!
The site 216 has NOT been loaded!
The site 228 has NOT been loaded!
The site 239 has NOT been loaded!
The site 242 has NOT been loaded!
The site 275 has NOT been loaded!
The site 306 has NOT been loaded!
The site 326 has NOT been loaded!
The site 363 has NOT been loaded!
The site 397 has NOT been loaded!
The site 414 has NOT been loaded!
The site 441 has NOT been loaded!
The function is completed!
```

```
In [27]: #creating the data frame with the loading times
         d4 = {'company' : pd.Series(lt_nm, index=[nm]),
               'loading time' : pd.Series(lt_time, index=[nm])}
         loading_time = pd.DataFrame(d4)
         loading_time.head(3) #we see the first 3 in the data frame
```

```
Out[27]:        company  loading time
         0      Walmart         0.212
         1  Exxon Mobil         3.447
         2        Apple         0.023
```

```
In [28]: #Find out how many and what type of images each site has
         #first we create the initially empty lists
         p_p = []
         p_d = []
         p_jpg = []
         p_jpeg = []
         p_gif = []
         p_tif = []
         p_tiff = []
         p_bmp = []
         p_jpe = []
         p_nm = []
         p_tt =[]
         nm = []
         p_url = []
```

```
In [29]: #Then we create the function that will explore
         #the html pages and search for the images
         def images (list500_sites,list500_names,list500_url):
             from time import time # I used it to see
             #how much time it does to run the function
             start = time ()
             for num in range(len(list500_names)):
```

```
myHTML = list500_sites[num]
image = ['.png','.dib','.jpg','.jpeg',
         '.bmp','.jpe','.gif','.tif','.tiff']
totalnumber = 0
if myHTML == 0:
    p_nm.insert(num,list500_names[num])
    p_p.insert(num,'n/a')
    p_d.insert(num,'n/a')
    p_jpg.insert(num,'n/a')
    p_jpeg.insert(num,'n/a')
    p_gif.insert(num,'n/a')
    p_tif.insert(num,'n/a')
    p_tiff.insert(num,'n/a')
    p_bmp.insert(num,'n/a')
    p_jpe.insert(num,'n/a')
    p_tt.insert(num,'n/a')
    nm.insert(num,num)
    p_url.insert(num,list500_url[num])
else:
    for index in range(len(image)):
        x = image[index]
        photo = re.findall(x,myHTML)
        if x == '.png':
            p = str (len(photo))
        if x == '.dib':
            d = str (len(photo))
        if x == '.jpg':
            jpg = str (len(photo))
        if x == '.jpeg':
            jpeg = str (len(photo))
        if x == '.gif':
            gif = str (len(photo))
        if x == '.tif':
            tif = str (len(photo))
        if x == '.tiff':
            tiff = str (len(photo))
        if x == '.bmp':
            bmp = str (len(photo))
        if x == '.jpe':
            jpe = str (len(photo))
        totalnumber = len(photo) + totalnumber
    total = str (totalnumber)
    p_nm.insert(num,list500_names[num])
    p_p.insert(num,p)
    p_d.insert(num,d)
    p_jpg.insert(num,jpg)
    p_jpeg.insert(num,jpeg)
    p_gif.insert(num,gif)
```

```
                         p_tif.insert(num,tif)
                         p_tiff.insert(num,tiff)
                         p_bmp.insert(num,bmp)
                         p_jpe.insert(num,jpe)
                         p_tt.insert(num,total)
                         nm.insert(num,num)
                         p_url.insert(num,list500_url[num])
             end = time ()
             duration = round (end - start, 3)
             minutes = round (duration /60, 1)
             print 'The lists are ready in ', minutes, ' minutes'
             print 'The lists are ready in ', duration, ' seconds'
```

```
In [30]: #Then we run the function for the first 20 sites for now
         images (list500_sites,list500_names,list500_url)
```

```
The lists are ready in  0.1  minutes
The lists are ready in  3.627  seconds
```

```
In [31]: #Finally we create a dataframe in order to see the results of the function
         d5 = {'company' : pd.Series(p_nm, index=[nm]),
               '.png' : pd.Series(p_p, index=[nm]),
               '.dib' : pd.Series(p_d, index=[nm]),
               '.jpg' : pd.Series(p_jpg, index=[nm]),
               '.jpeg' : pd.Series(p_jpeg, index=[nm]),
               '.bmp' : pd.Series(p_bmp, index=[nm]),
               '.jpe' : pd.Series(p_jpe, index=[nm]),
               '.gif' : pd.Series(p_gif, index=[nm]),
               '.tif' : pd.Series(p_tif, index=[nm]),
               '.tiff' : pd.Series(p_tiff, index=[nm]),
               'total images' : pd.Series(p_tt, index=[nm])}
         images_types = pd.DataFrame(d5)
         images_types.head(3) #we see the first 3 in the data frame
```

```
Out[31]:   .bmp .dib .gif .jpe .jpeg .jpg .png .tif .tiff      company total images
         0    0    0   29  134   134   94   42    7     0      Walmart          440
         1    0    0    1    0     0   17    2    4     0  Exxon Mobil           24
         2    0    0    1    0     0    0    2    0     0        Apple             3
```

```
In [32]: #Now we will find the different dimensions that each site uses
         #initially we create the empty lists we will need
         nm = []
         s_comp = []
         s_dimensions = []
         s_times = []
         s_tt_dif_dim = []
         ht = [] #list of different heights in each case
         wt = [] #list of different widths in each case
         h_w = [] # combinations of height and width
         dif_size = []
         un_size = []
         s_url = []
```

```
In [33]: #With the below function we will gather
         #in a variable all the different dimensions
         #and in another one all the times that each
         #dimension occures for each html code
         def find_dif_sizes (list_company_website,list500_names,list500_url):
             from time import time # I used it to see how much time it does to run
             start = time ()
             for num in range(len(list500_names)):
                     nm.insert(num,num)
                     s_comp.insert(num,list500_names[num])
                     s_url.insert(num,list500_url[num])
                     myHTML = list500_sites[num]
                     if myHTML == 0:
                         s_dimensions.insert(num,0)
                         s_times.insert(num,0)
                     else:
                         soup = BeautifulSoup(myHTML, "lxml")
                         # we create 2 local variables so as to gather the
                         #different dimensions and occurencies  of each page sepera
                         s_dimensions_local = []
                         s_times_local = []
                         hw = 0
                         # we use it for the lists of height and width
                         # find all the img in the first site html.Since in some
                         #cases either the height or the width is missing we would
                         #like to keep only the ones that have both dimensions
                         for tag in soup.find_all('img'):
                             h = tag.attrs.get('height', None)
                             w = tag.attrs.get('width', None)
                             #we use if to check which ones have both
                             if h != None:
                                 if w != None:
                                     ht.insert(hw,h)
                                     wt.insert(hw,w)
                                     hw = hw + 1
                         hw2 = 0
                         for l in range(len(ht)):
                             h_w_c = ht[l] + 'x' + wt[l]
                             #we create a str with the form (300x300)
                             #so as to be more easily to read later on
                             h_w.insert(hw2,h_w_c)
                             #we put it in a new list
                             hw2 = hw2 + 1
```

```
if h_w == []:#we check if there are not any dimensions ava
    nm.insert(num,num)
    s_comp.insert(num,list500_names[num])
    s_dimensions.insert(num,0)
    s_times.insert(num,0)
if h_w != []:#now we continue with the cases
    #where the dimensions are indeed available
    from collections import Counter
    hw_unique = Counter(h_w)
    hw_unique2 = str(hw_unique)
    #the unique different dimensions for the specific site
    #Due to the fact that we are talking about
    #a list we have to split the parts we need
    split1 = hw_unique2.split('{')
    a = split1[1]
    split2 = a.split('}')
    b = split2[0]
    split3 = b.split(',')
    finalsplit = []
    fs = []
    z = 0
    m = 1
    j = 0
    z1 = 0
    m1 = 1
    #each of the items in split3 has a form '300x300 : 15
    #and in order to create the dataframe we have
    #to split this form and keep the informations in diffe
    for numb in split3:
        oldstring = numb
        newstring = oldstring.replace("'", "")
        new = newstring.replace("'","")
        string = new.replace(" ","")
        finalstring = string.split(':')
        #the finalstring is a list that contains the dimen
        #and the occurencies in order toseperate in differ
        #lists we create an additional loop
        for xx in range(len(finalstring)):
            ax = finalstring[xx]
            if 'x' in ax:
                s_dimensions_local.insert(z1,finalstring[x
                z1 = z1 + 1
            else:
                s_times_local.insert(m1,finalstring[xx])
                m1 = m1 + 1
    #Now we can add to the lists the parts we created so a
    #to have them all gathered together
    s_dimensions.insert(num,s_dimensions_local)
```

```
                             s_times.insert(num,s_times_local)
             end = time ()
             duration = round (end - start, 3)
             minutes = round (duration /60, 1)
             print 'The lists are ready in ', minutes, ' minutes'
             print 'The lists are ready in ', duration, ' seconds'
```

```
In [34]: #Run the function for the first 20 sites
         find_dif_sizes (list500_sites,list500_names,list500_url)
```

```
The lists are ready in  1.4  minutes
The lists are ready in  81.676  seconds
```

```
In [35]: #Find the unique different image dimensions and put them on a list
         def unique_dif_sizes (s_dimensions,list500_names):
             ds = 0
             for num in range(len(list500_names)):
                 asw = s_dimensions[num]
                 if asw != 0 :
                     for s in range(len(asw)):
                         ss = asw[s]
                         dif_size.insert(ds,ss)
                         ds = ds + 1
             dsu = 0
             for i in dif_size:
                 if i not in un_size:
                     un_size.insert(dsu,i)
                     dsu = dsu + 1
```

```
In [36]: #Run the function unique_dif_sizes
         unique_dif_sizes (s_dimensions,list500_names)
```

```
In [37]: #The lists we will need for the next function
         t_f_s = []
         ttf = []
         nm = []
         com = []
```

```
In [38]: #Function in order to check whether or not each
         #company has these dimensions
         def dimensions_per_company (un_size,list500_names):
             from time import time
             # I used it to see how much time it does to run the function
             start = time ()
             #t_f_s.insert(0,un_size)
             #ttf.insert(0,t_f_s)
             for num in range(len(list500_names)):
                 #print(str(num))
                 s1a = s_dimensions[num]
                 #dimensions of site num
                 where = [] #empty list
                 wh = 0
                 haveornot = []
                 for er in range (len(un_size)):
                     if s1a != 0 :
                         for sizea in s1a:
                             if sizea == un_size[er]:
                                 where.insert(wh,str(er))
                                 wh = wh +1
                                 break
                     if str(er) in where:
                         haveornot.insert(er,True)
                     else:
                         haveornot.insert(er,False)

                 t_f_s.insert(num,haveornot)
                 ttf.insert(num,t_f_s)
                 nm.insert(num,num)
                 com.insert(num,list500_names[num])
             end = time ()
             duration = round (end - start, 3)
             minutes = round (duration /60, 1)
             print 'The lists are ready in ', minutes, ' minutes'
             print 'The lists are ready in ', duration, ' seconds'
```

```
In [39]: #Run the function dimensions_per_company
         dimensions_per_company (un_size,list500_names)
```

```
The lists are ready in  0.1  minutes
The lists are ready in  3.429  seconds
```

```
In [40]: #Create an initial dataframe where we will add the sizes later on
         d6 = {'company' : pd.Series(com, index=[nm])}
         sizess = pd.DataFrame(d6)
         sizess.head(3)
```

```
Out[40]:        company
         0      Walmart
         1  Exxon Mobil
         2        Apple
```

```
In [41]: #Now we want to break the variable t_f_s
         #in order to add the columns to the dataframe
         #Finally we create the data frame with the elements we found
         def final_dimensions_dataframe (un_size,t_f_s,list500_names):
             from time import time
```

```
    # I used it to see how much time it does to run the function
    start = time ()
    for q in range(len(un_size)):
        names = un_size[q]
        var = []
        for num in range(len(list500_names)):
            a = t_f_s[num]
            var.insert(num,a[q])
        sizess[names] = pd.Series(var, index=sizess.index)
    end = time ()
    duration = round (end - start, 3)
    minutes = round (duration /60, 1)
    print 'The lists are ready in ', minutes, ' minutes'
    print 'The lists are ready in ', duration, ' seconds'
```

```
In [42]: #Run the function final_dimensions_dataframe
         final_dimensions_dataframe (un_size,t_f_s,list500_names)
```

```
The lists are ready in  0.0  minutes
The lists are ready in  0.301  seconds
```

```
In [43]: sizess.tail(3)
```

```
Out[43]:                     company 144x144 15x75  8x15 44x556   1x1 800x1200 autox1
         497                     NVR    True  True  True   True  True     True      T
         498   Cincinnati Financial     True  True  True   True  True     True      T
         499      Burlington Stores     True  True  True   True  True     True      T

            24pxx133px 21pxx173px  ...   318x460 370x630 75x171 105x530 781x1800
         497       True       True  ...     False   False  False   False    False
         498       True       True  ...      True    True   True    True     True
         499       True       True  ...      True    True   True    True     True

            50x100 360x1306 306x1306 338x1306 82x136
         497  False    False    False    False  False
         498   True    False    False    False  False
         499   True     True     True     True   True

         [3 rows x 694 columns]
```

```
In [44]: #In order to validate the html code we will use the w3 validator
         #We will validate each url and then we will open the url of the validatio
         #so as to extract the errors, the info warnings and the non-document-erro
         #First we create the empty lists we would use later on
         num_errors = []
         num_info_warnings = []
         num_non_doc = []
         nm = []
         num_open_page = []
         empty = ""

In [45]: #Then we create the function that will pull the informations we want
         def html_validation (list500_url,list500_names):
             from time import time # I used it to see how much time it does to run
             start = time ()
             for num in range(len(list500_names)):
                 line = list500_url[num]
                 url_check = "https://validator.w3.org/nu/?doc=https://" + line
                 browser = urllib2.build_opener()
                 browser.addheaders = [('User-agent', 'Mozilla/5.0')]
                 response = browser.open(url_check)
                 html_check = response.read()
                 html_check
                 check = str(html_check)
                 er = 0
                 err = 0
                 errr = 0
                 e = False
                 if check != empty:
                     e = True
                     soup = BeautifulSoup(check,"lxml")
                     o = 0
                     keyf = []
                     for row in soup.html.body.findAll('div'):
                         keyf.insert(o,row)
                         o = o + 1
                     #print(len(keyf),list500_url[num], "site number: ", str(num))
                     if len(keyf) != 0:
                             keyfin = str(keyf[2])
                             #the elements we need is in the 2nd div of the code
                             dol= re.findall('class="error"',keyfin)
                             er = er + len(dol)
                             doll= re.findall('class="info warning"'
                                              ,keyfin)
                             err = err + len(doll)
                             dolll= re.findall('class="non-document-error io"'
                                               ,keyfin)
                             errr = errr + len(dolll)
                 num_errors.insert(num,er)
                 num_info_warnings.insert(num,err)
                 num_non_doc.insert(num,errr)
                 nm.insert(num,num)
                 num_open_page.insert(num,e)
             end = time ()
             duration = round (end - start, 3)
             minutes = round (duration /60, 1)
             print 'The lists are ready in ', minutes, ' minutes'
```

```
In [46]: #Now we will run the function we created
         html_validation (list500_url,list500_names)

The lists are ready in  36.8  minutes
```

```
In [47]: #After the checks we will create the dataframe with the informations we wa
         d8 = {'company' : pd.Series(list500_names, index=[nm]),
               'The_page_opened' : pd.Series(num_open_page, index=[nm])
               ,'number_of_errors' : pd.Series(num_errors, index=[nm]),
               'number_of_warning' : pd.Series(num_info_warnings, index=[nm])
               ,'non-document-error' : pd.Series(num_non_doc, index=[nm])}
         html_val = pd.DataFrame(d8)
         html_val.head(3)
```

```
Out[47]:   The_page_opened      company  non-document-error  number_of_errors  \
         0             True      Walmart                   0               814
         1             True  Exxon Mobil                   0                55
         2             True        Apple                   0                16

            number_of_warning
         0                  1
         1                 29
         2                  7
```

```
In [48]: #The next step is to take some informations from the fortune 500 site for
         #In order to achieve that we should open the pages for each one of the sit
         #Since there is a pattern in the way the pages are named it shouldn't be c
         #Firstly we should create the pattern with which we will download the page
         #By running the code we can see that the names of each comany are not
         #written exactly as we have saved them
         #So we do need to alter the names first in order for the below function to
```

```
In [49]: #creating a new list with alterations in order for the names
         #to match the ones that fortune 500 uses so that we can download the html
         list_company_name_new = []
         for num in range (0,500):
             cn = list_company_name[num]
             cn = cn.replace(" ", "-")
             cn = cn.replace("&", "")
             cn = cn.replace("’", "")
             cn = cn.replace(".", "-")
             cn = cn.replace("amp;", "")
             company = cn.lower()
             list_company_name_new.insert(num,cn)
```

```
In [50]: fortune_pages = []
         def fortune500 (list_company_name_new):
             from time import time # I used it to see how much time it does to run
             start = time ()
             for num3 in range (0,500):
                 i = str (num3 +1)
                 companyname =  list_company_name_new[num3]
                 browser = urllib2.build_opener()
                 #because i work from different computers with different
                 #pyhton version some commands are not recognizable in each versio
                 browser.addheaders = [('User-agent', 'Mozilla/5.0')]
                 site_fortune = "http://beta.fortune.com/fortune500/"+companyname+'
                 page_fortune = browser.open(site_fortune)
                 html_fortune = page_fortune.read()
                 #print("fortune page for company: ", list_company_name_new[num3],
                 fortune_pages.insert(num3, html_fortune)
             end = time ()
             duration = round (end - start, 3)
             minutes = round (duration /60, 1)
             print 'The lists are ready in ', minutes, ' minutes'
             print 'The lists are ready in ', duration, ' seconds'

In [51]: #Run the function we created
         fortune500 (list_company_name_new)

The lists are ready in  20.1  minutes
The lists are ready in  1208.919  seconds


In [52]: #Now that we have opened the url we are going to extract
         #some informations that we need from them
         #In order to do that initially we have to create
         #the variables we will need
         keyf =[]
         per =[]
         rev_dol = []
         rev_per = []
         prof_dol = []
         prof_per = []
         assets_dol = []
         assets_per = []
         tse_dol = []
         tse_per = []
         mar_dol = []
         mar_per = []
         market = []
         nm = []
         ln = []
         urln = []
         empty = []
```

```
In [53]: def fortune_metrics (list_company_name,list_company_website):
             x = 0
             for n in range (0,500):   #we put 25 for testing
                 nm.insert(x,x)
                 ln.insert(x,list_company_name[n])
                 urln.insert(x,list_company_website[n])
                 files = fortune_pages[x]
                 soup = BeautifulSoup(files,"lxml")
                 o=0
                 for row in soup.html.body.findAll('tbody'):
                     keyf.insert(o,row)
                     o=o+1
                 keyfin = keyf[0]
                 #the elements we need is in the first tbody of the code
                 data = keyfin.findAll('td')

                 one = str(data[0])
                 # revenue
                 two = str(data[1])
                 # revenue in dollars we need to extract this
                 revdol= re.findall('>\$(.+?)</td>',two)
                 #we keep only the numbers
                 if revdol[0] != empty:
                     w = revdol[0]
                     a = w.replace("[", "")
                     r = a.replace("]","")
                     rev_dol.insert(x,r)
                 else:
                     rev_dol.insert(x,'not available')
                 tria = str(data[2])
                 # revenue in percentage we need to extract this as well
                 revper= re.findall('>(.+?)%</td>',tria)
                 #we keep only the numbers
                 if revper != empty:
                     w = revper[0]
                     a = w.replace("[", "")
                     r1 = a.replace("]","")
                     rev_per.insert(x,r1)
                 else:
                     rev_per.insert(x,'not available')
                 four = str(data[3])   # profit
                 five = str(data[4])
                 # profit in dollars we need to extract this
                 profdol= re.findall('>\$(.+?)</td>',five)
                 #we keep only the numbers
                 if profdol != empty:
                     w = profdol[0]
                     a = w.replace("[", "")
```

```
    p = a.replace("]","")
    prof_dol.insert(x,p)
else:
    prof_dol.insert(x,'not available')
six = str(data[5])
# profit in percentage we need to extract this as well
profper = re.findall('>(.+?)%</td>',six)
#we keep only the numbers
if profper != empty:
    w = profper[0]
    a = w.replace("[", "")
    p1 = a.replace("]","")
    prof_per.insert(x,p1)
else:
    prof_per.insert(x,'not available')
seven = str(data[6]) #assets
eight = str(data[7]) #assets in dollars we need to extract this
assetsdol= re.findall('>\$(.+?)</td>',eight)
#we keep only the numbers
if assetsdol != empty:
    w = assetsdol[0]
    a = w.replace("[", "")
    ass = a.replace("]","")
    assets_dol.insert(x,ass)
else:
    assets_dol.insert(x,'not available')
ten = str(data[9]) #Total Stockholder Equity ($M)
eleven = str(data[10])
#Total Stockholder Equity ($M) in dollars we need to extract this
tsedol= re.findall('>\$(.+?)</td>',eleven)
#we keep only the numbers
if tsedol != empty:
    w = tsedol[0]
    a = w.replace("[", "")
    ts = a.replace("]","")
    tse_dol.insert(x,ts)
else:
    tse_dol.insert(x,'not available')
thirteen = str(data[12]) # market value
fourteen = str(data[13])
# market value in dollars we need to extract this
mardol= re.findall('>\$(.+?)</td>',fourteen)
#we keep only the numbers
if mardol != empty:
    w = mardol[0]
    a = w.replace("[", "")
    mar = a.replace("]","")
    mar_dol.insert(x,mar)
```

```
                 else:
                     mar_dol.insert(x,'not available')
                 x = x + 1
             print "The function is complete!"

In [54]: fortune_metrics (list_company_name,list_company_website)

The function is complete!


In [55]: d9 = {'company' : pd.Series(ln, index=[nm]),
               'Revenues $' : pd.Series(rev_dol, index=[nm]),
               'Revenues %' : pd.Series(rev_per, index=[nm]),
               'Assets $' : pd.Series(assets_dol, index=[nm]),
               'Total Stockholder Equity $' : pd.Series(tse_dol, index=[nm]),
               'Market value $' : pd.Series(mar_dol, index=[nm])}
         fort500 = pd.DataFrame(d9)
         fort500.head(3)

Out[55]:   Assets $ Market value $ Revenues $ Revenues % Total Stockholder Equity $
         0  199,581        215,356    482,130       -0.7                    80,546
         1  336,758        347,129    246,204      -35.6                   170,811
         2  290,479        604,304    233,715       27.9                   119,355

                company
         0      Walmart
         1  Exxon Mobil
         2        Apple

In [56]: result = pd.merge(fort500, html_val, how='inner', on=['company', 'company'
         result2 = pd.merge(social_media, fre, how='inner', on=['company', 'company
         result3 = pd.merge(sites_links, sizess, how='inner', on=['company', 'compa
         result4 = pd.merge(images_types, loading_time, how='inner', on=['company',
         result5 = pd.merge(result,result2 , how='inner', on=['company', 'company']
         result6 = pd.merge(result3, result4, how='inner', on=['company', 'company'
         final = pd.merge(result5, result6, how='inner', on=['company', 'company'])
         final.head(3)

Out[56]:   Assets $ Market value $ Revenues $ Revenues % Total Stockholder Equity $
         0  199,581        215,356    482,130       -0.7                    80,546
         1  336,758        347,129    246,204      -35.6                   170,811
         2  290,479        604,304    233,715       27.9                   119,355

                company The_page_opened  non-document-error  number_of_errors  \
         0      Walmart            True                   0               814
         1  Exxon Mobil            True                   0                55
         2        Apple            True                   0                16

            number_of_warning     ...      .dib .gif .jpe .jpeg .jpg .png .tif .ti
         0                    1     ...         0    29   134    134    94    42     7
         1                   29     ...         0     1     0      0    17     2     4
         2                    7     ...         0     1     0      0     0     2     0

           total images  loading time
         0          440         0.212
         1           24         3.447
         2            3         0.023

         [3 rows x 729 columns]

In [57]: final.to_csv('total_500_new.csv', sep=';')

In [58]: data500 = pd.read_csv("total_500_new.csv", sep=';')

In [59]: data500.head(3)

Out[59]:    Unnamed: 0 Assets $ Market value $ Revenues $ Revenues %  \
         0           0  199,581        215,356    482,130       -0.7
         1           1  336,758        347,129    246,204      -35.6
         2           2  290,479        604,304    233,715       27.9

           Total Stockholder Equity $      company The_page_opened  non-document-er
         0                     80,546      Walmart            True
         1                    170,811  Exxon Mobil            True
         2                    119,355        Apple            True

            number_of_errors     ...       .dib .gif .jpe .jpeg .jpg .png .tif .tif
         0               814     ...          0   29  134   134   94   42    7
         1                55     ...          0    1    0     0   17    2    4
         2                16     ...          0    1    0     0    0    2    0

           total images  loading time
         0          440         0.212
         1           24         3.447
         2            3         0.023

         [3 rows x 730 columns]

In [ ]:
```