Problem Statement:

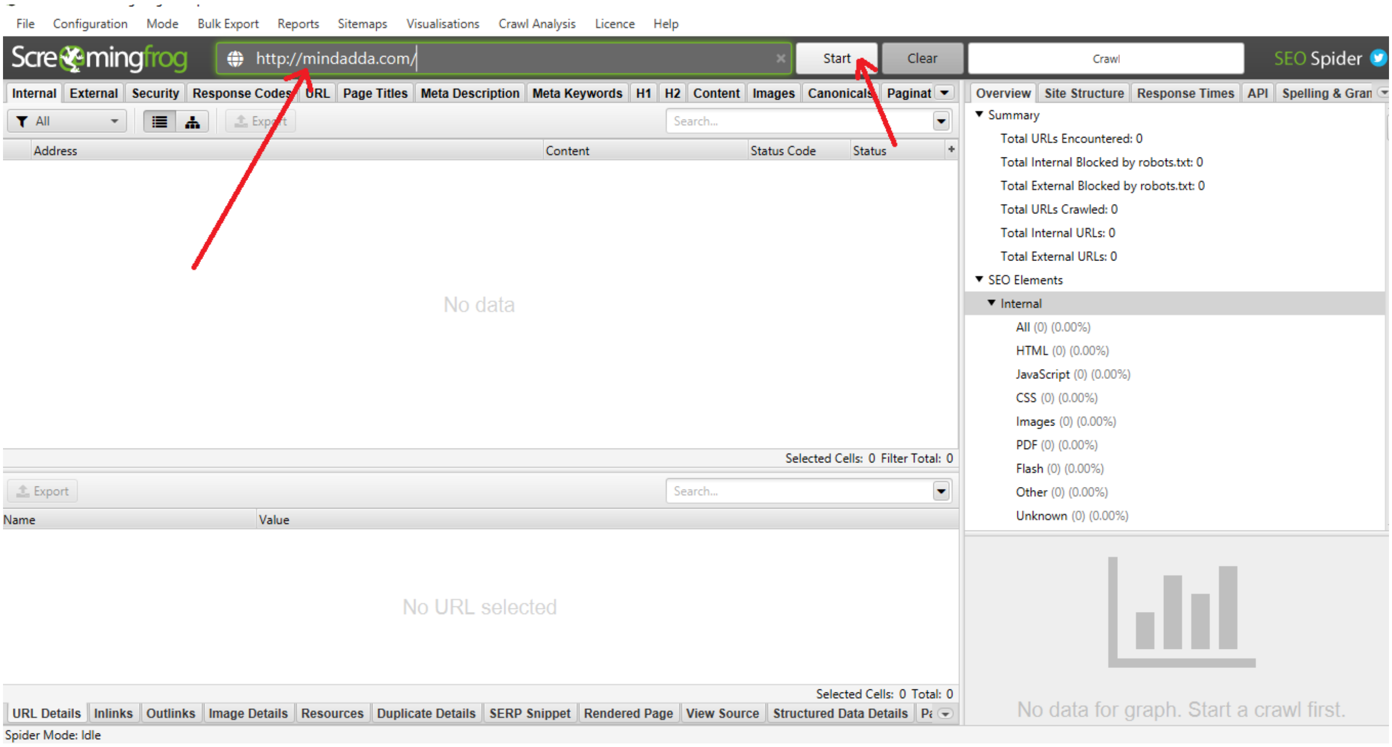
A comparative analysis between competitors that related to mindadda.

Assumption:

1. *BYJUS* - a well developed one.
2. *MIDBRAIN ACADEMY* - a moderate one.
3. *MINDADDA* - Starting company.

Data collection Process

Used a tool named '**Screaming frog SEO Spider**' to collect the data from each web page from the source main link that user provided at input box which highlighted in below image

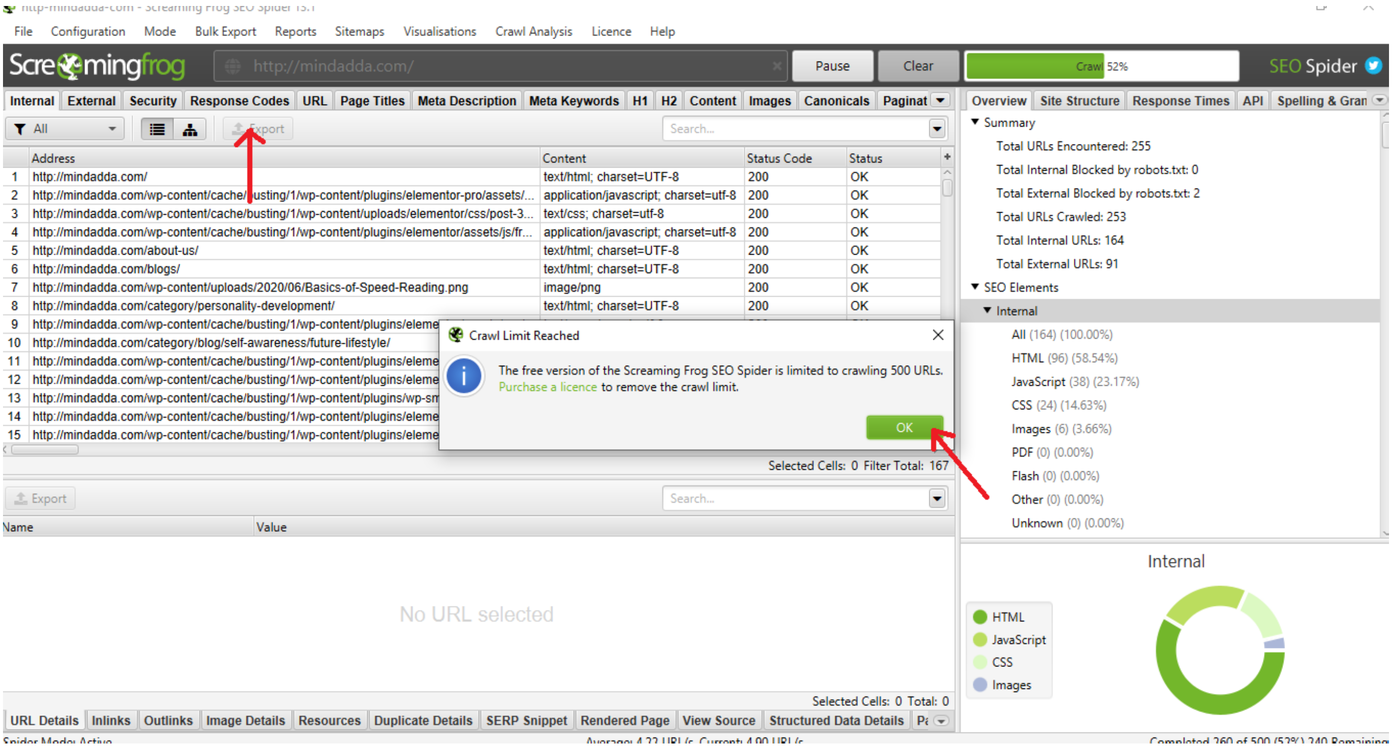


- Summary
  - Total URLs Encountered: 0
  - Total Internal Blocked by robots.txt: 0
  - Total External Blocked by robots.txt: 0
  - Total URLs Crawled: 0
  - Total Internal URLs: 0
  - Total External URLs: 0
- SEO Elements
  - Internal
    - All (0) (0.00%)
    - HTML (0) (0.00%)
    - JavaScript (0) (0.00%)
    - CSS (0) (0.00%)
    - Images (0) (0.00%)
    - PDF (0) (0.00%)
    - Flash (0) (0.00%)
    - Other (0) (0.00%)
    - Unknown (0) (0.00%)

After clicking on start, the crawling process will begin

From below image there are two points:

1. limitation of 500 urls scraping
2. Second thing is we can export in to csv file - which is the input for our analysis



	Address	Content	Status Code	Status
1	http://mindadda.com/	text/html; charset=UTF-8	200	OK
2	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/elementor-pro/assets/...	application/javascript; charset=utf-8	200	OK
3	http://mindadda.com/wp-content/cache/busting/1/wp-content/uploads/elementor/css/post-3...	text/css; charset=utf-8	200	OK
4	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/elementor/assets/js/fr...	application/javascript; charset=utf-8	200	OK
5	http://mindadda.com/about-us/	text/html; charset=UTF-8	200	OK
6	http://mindadda.com/blogs/	text/html; charset=UTF-8	200	OK
7	http://mindadda.com/wp-content/uploads/2020/06/Basics-of-Speed-Reading.png	image/png	200	OK
8	http://mindadda.com/category/personality-development/	text/html; charset=UTF-8	200	OK
9	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/eleme			
10	http://mindadda.com/category/blog/self-awareness/future-lifestyle/			
11	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/eleme			
12	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/eleme			
13	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/wp-sm			
14	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/eleme			
15	http://mindadda.com/wp-content/cache/busting/1/wp-content/plugins/eleme			

Crawl Limit Reached

The free version of the Screaming Frog SEO Spider is limited to crawling 500 URLs. Purchase a licence to remove the crawl limit.

OK

Selected Cells: 0 Filter Total: 167

Export

Name Value

No URL selected

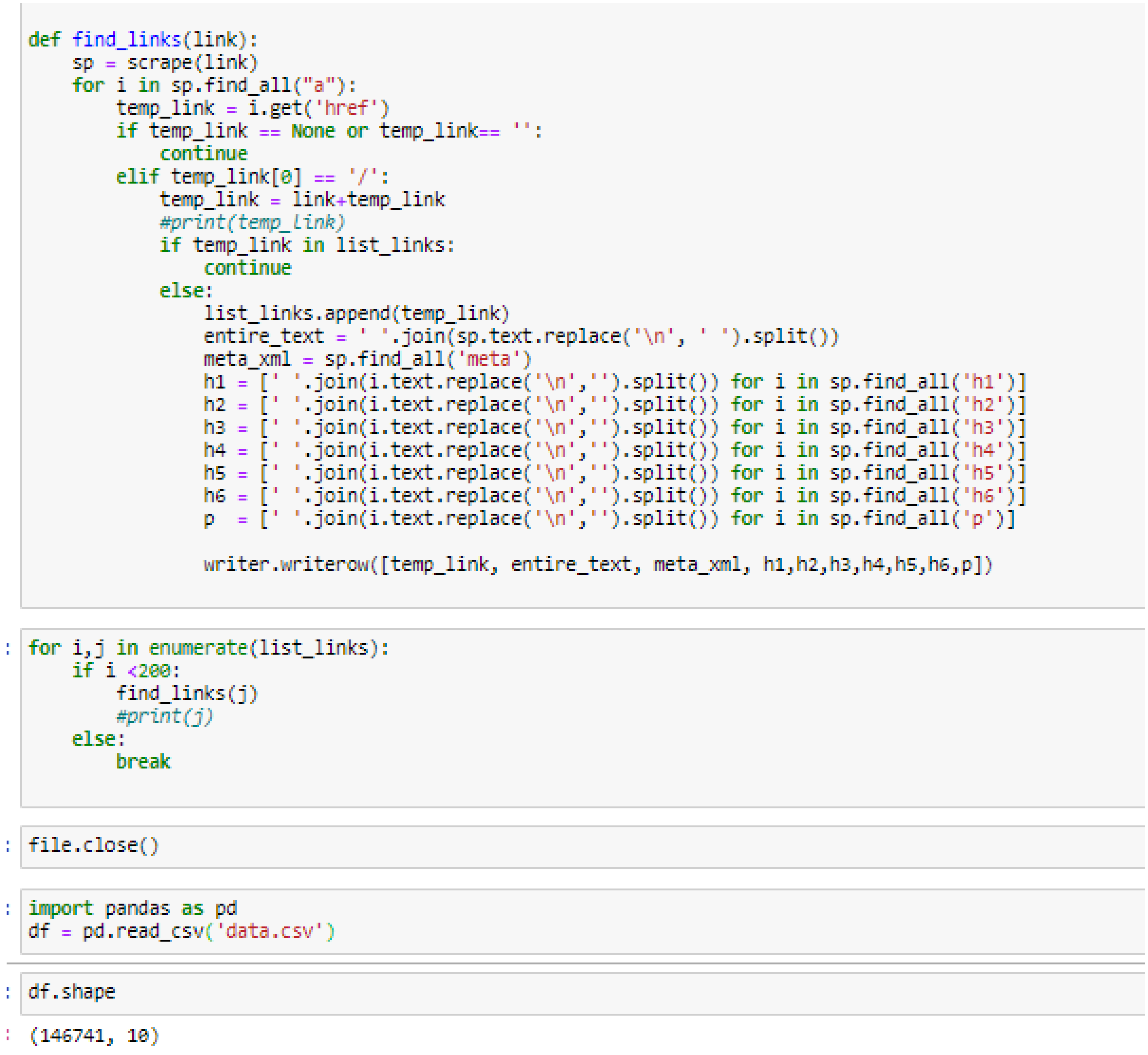
Selected Cells: 0 Total: 0

- Summary
  - Total URLs Encountered: 255
  - Total Internal Blocked by robots.txt: 0
  - Total External Blocked by robots.txt: 2
  - Total URLs Crawled: 253
  - Total Internal URLs: 164
  - Total External URLs: 91
- SEO Elements
  - Internal
    - All (164) (100.00%)
    - HTML (96) (58.54%)
    - JavaScript (38) (23.17%)
    - CSS (24) (14.63%)
    - Images (6) (3.66%)
    - PDF (0) (0.00%)
    - Flash (0) (0.00%)
    - Other (0) (0.00%)
    - Unknown (0) (0.00%)

Internal
HTML
JavaScript
CSS
Images
Spider Mode: Active
Average: 4.22 URL/s. Current: 4.90 URL/s.
Completed 260 of 500 (52%) 240 Remaining

For limitation of URLS extraction:

we developed our own python library to extract all urls contains in their website.



```
def find_links(link):
    sp = scrape(link)
    for i in sp.find_all("a"):
        temp_link = i.get('href')
        if temp_link == None or temp_link== '':
            continue
        elif temp_link[0] == '/':
            temp_link = link+temp_link
            #print(temp_link)
            if temp_link in list_links:
                continue
            else:
                list_links.append(temp_link)
                entire_text = ' '.join(sp.text.replace('\n', ' ').split())
                meta_xml = sp.find_all('meta')
                h1 = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('h1')]
                h2 = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('h2')]
                h3 = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('h3')]
                h4 = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('h4')]
                h5 = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('h5')]
                h6 = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('h6')]
                p  = [' '.join(i.text.replace('\n','').split()) for i in sp.find_all('p')]

                writer.writerow([temp_link, entire_text, meta_xml, h1,h2,h3,h4,h5,h6,p])
```

```
: for i,j in enumerate(list_links):
      if i <200:
          find_links(j)
          #print(j)
      else:
          break
```

```
: file.close()
```

```
: import pandas as pd
  df = pd.read_csv('data.csv')
```

```
: df.shape
```

```
: (146741, 10)
```

Data analysis

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [89]:

```
#BYJUS Dataset loading
df1 = pd.read_csv('byjus_data.csv')
df1.head(5)
```

Out[89]:

	Address	Content	Status Code	Status	Indexability	Indexability Status	Title 1
0	https://byjus.com/	text/html; charset=UTF-8	200	OK	Indexable	NaN	e Learning for Online Courses like UPSC K3, K..
1	https://byjus.com/commerce/ts-grewal-solutions/	text/html; charset=UTF-8	200	OK	Indexable	NaN	TS Grewal Solutions - Class 11th and 12th 20 Exam
2	https://byjus.com/ncert-solutions-class-9-scie...	text/html; charset=UTF-8	200	OK	Indexable	NaN	NCERT Solutions Class 1 Chemistry Chapter 4 S..
3	https://byjus.com/ncert-books/	text/html; charset=UTF-8	200	OK	Indexable	NaN	NCERT Book PDF Download For Free (2020-21 - C..
4	https://byjus.com/pseb/pseb-syllabus/	text/html; charset=UTF-8	200	OK	Indexable	NaN	Punjab School Education Board Syllabus Get t..

5 rows × 50 columns

In [90]:

```
df1.shape
```

Out[90]:

```
(481, 50)
```

In [91]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 481 entries, 0 to 480
Data columns (total 50 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Address                         481 non-null    object
 1   Content                         481 non-null    object
 2   Status Code                     481 non-null    int64
 3   Status                          481 non-null    object
 4   Indexability                    481 non-null    object
 5   Indexability Status             8 non-null      object
 6   Title 1                         473 non-null    object
 7   Title 1 Length                  481 non-null    int64
 8   Title 1 Pixel Width             481 non-null    int64
 9   Meta Description 1              471 non-null    object
 10  Meta Description 1 Length       481 non-null    int64
 11  Meta Description 1 Pixel Width  481 non-null    int64
 12  Meta Keyword 1                  0 non-null      float64
 13  Meta Keywords 1 Length          481 non-null    int64
 14  H1-1                            460 non-null    object
 15  H1-1 length                     481 non-null    int64
 16  H2-1                            380 non-null    object
 17  H2-1 length                     481 non-null    int64
 18  H2-2                            285 non-null    object
 19  H2-2 length                     481 non-null    int64
 20  Meta Robots 1                   471 non-null    object
 21  X-Robots-Tag 1                  0 non-null      float64
 22  Meta Refresh 1                  0 non-null      float64
 23  Canonical Link Element 1        473 non-null    object
 24  rel="next" 1                    0 non-null      float64
 25  rel="prev" 1                    0 non-null      float64
 26  HTTP rel="next" 1               0 non-null      float64
 27  HTTP rel="prev" 1               0 non-null      float64
 28  Size (bytes)                    481 non-null    int64
 29  Word Count                      481 non-null    int64
 30  Text Ratio                      481 non-null    float64
 31  Crawl Depth                     481 non-null    int64
 32  Link Score                      0 non-null      float64
 33  Inlinks                         481 non-null    int64
 34  Unique Inlinks                  481 non-null    int64
 35  % of Total                      481 non-null    float64
 36  Outlinks                        481 non-null    int64
 37  Unique Outlinks                 481 non-null    int64
 38  External Outlinks               481 non-null    int64
 39  Unique External Outlinks        481 non-null    int64
 40  Closest Similarity Match        0 non-null      float64
 41  No. Near Duplicates             0 non-null      float64
 42  Spelling Errors                 0 non-null      float64
 43  Grammar Errors                  0 non-null      float64
 44  Hash                            473 non-null    object
 45  Response Time                   481 non-null    float64
 46  Last Modified                   0 non-null      float64
 47  Redirect URL                    8 non-null      object
 48  Redirect Type                   8 non-null      object
 49  URL Encoded Address             481 non-null    object
dtypes: float64(16), int64(18), object(16)
memory usage: 188.0+ KB
```

In [92]:

```
df1.columns
```

Out[92]:

```
Index(['Address', 'Content', 'Status Code', 'Status', 'Indexability',
       'Indexability Status', 'Title 1', 'Title 1 Length',
       'Title 1 Pixel Width', 'Meta Description 1',
       'Meta Description 1 Length', 'Meta Description 1 Pixel Width',
       'Meta Keyword 1', 'Meta Keywords 1 Length', 'H1-1', 'H1-1 length',
       'H2-1', 'H2-1 length', 'H2-2', 'H2-2 length', 'Meta Robots 1',
       'X-Robots-Tag 1', 'Meta Refresh 1', 'Canonical Link Element 1',
       'rel="next" 1', 'rel="prev" 1', 'HTTP rel="next" 1',
       'HTTP rel="prev" 1', 'Size (bytes)', 'Word Count', 'Text Ratio',
       'Crawl Depth', 'Link Score', 'Inlinks', 'Unique Inlinks', '% of Tota
l',
       'Outlinks', 'Unique Outlinks', 'External Outlinks',
       'Unique External Outlinks', 'Closest Similarity Match',
       'No. Near Duplicates', 'Spelling Errors', 'Grammar Errors', 'Hash',
       'Response Time', 'Last Modified', 'Redirect URL', 'Redirect Type',
       'URL Encoded Address'],
      dtype='object')
```

In [95]:

```
df1.describe()
```

Out[95]:

	Status Code	Title 1 Length	Title 1 Pixel Width	Meta Description 1 Length	Meta Description 1 Pixel Width	Meta Keyword 1	Meta Keywords 1 Length	
count	481.000000	481.000000	481.000000	481.000000	481.000000	0.0	481.0	48
mean	201.696466	63.752599	601.056133	181.920998	1171.950104	NaN	0.0	34
std	13.058178	18.284645	173.301432	49.252094	316.576402	NaN	0.0	2
min	200.000000	0.000000	0.000000	0.000000	0.000000	NaN	0.0	(
25%	200.000000	55.000000	512.000000	159.000000	1027.000000	NaN	0.0	1
50%	200.000000	63.000000	594.000000	178.000000	1151.000000	NaN	0.0	3
75%	200.000000	73.000000	689.000000	201.000000	1302.000000	NaN	0.0	4
max	302.000000	164.000000	1584.000000	441.000000	2782.000000	NaN	0.0	14

8 rows × 34 columns

In [96]:

```
# Midbrain Academy dataset loading
df2 = pd.read_csv('midbrainacademy_data.csv')
df2.sample(5)
```

Out[96]:

	Address	Content	Status Code	Status	Indexability
260	https://midbrainacademy.in/midbrain-activation...	text/html	200	OK	Indexable
108	https://www.midbrainacademy.in/js/velidation.js	application/javascript	200	OK	Indexable
350	http://midbrainacademy.in/midbrain-activation-...	text/html	200	OK	Non-Indexable
416	http://midbrainacademy.in/privacy-policy.html	text/html	200	OK	Indexable
362	https://midbrainacademy.in/blog/nlp-training-f...	text/html	404	Not Found	Non-Indexable

5 rows × 50 columns

In [97]:

```
df2.shape
```

Out[97]:

```
(444, 50)
```

In [98]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 444 entries, 0 to 443
Data columns (total 50 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Address                         444 non-null    object
 1   Content                         444 non-null    object
 2   Status Code                     444 non-null    int64
 3   Status                          444 non-null    object
 4   Indexability                    444 non-null    object
 5   Indexability Status             99 non-null     object
 6   Title 1                         182 non-null    object
 7   Title 1 Length                  444 non-null    int64
 8   Title 1 Pixel Width             444 non-null    int64
 9   Meta Description 1              158 non-null    object
 10  Meta Description 1 Length       444 non-null    int64
 11  Meta Description 1 Pixel Width  444 non-null    int64
 12  Meta Keyword 1                  98 non-null     object
 13  Meta Keywords 1 Length          444 non-null    int64
 14  H1-1                            182 non-null    object
 15  H1-1 length                     444 non-null    int64
 16  H2-1                            128 non-null    object
 17  H2-1 length                     444 non-null    int64
 18  H2-2                            109 non-null    object
 19  H2-2 length                     444 non-null    int64
 20  Meta Robots 1                   176 non-null    object
 21  X-Robots-Tag 1                  0 non-null      float64
 22  Meta Refresh 1                  0 non-null      float64
 23  Canonical Link Element 1        0 non-null      float64
 24  rel="next" 1                    0 non-null      float64
 25  rel="prev" 1                    0 non-null      float64
 26  HTTP rel="next" 1               0 non-null      float64
 27  HTTP rel="prev" 1               0 non-null      float64
 28  Size (bytes)                    444 non-null    int64
 29  Word Count                      444 non-null    int64
 30  Text Ratio                      444 non-null    float64
 31  Crawl Depth                     444 non-null    int64
 32  Link Score                      0 non-null      float64
 33  Inlinks                         444 non-null    int64
 34  Unique Inlinks                  444 non-null    int64
 35  % of Total                      444 non-null    float64
 36  Outlinks                        444 non-null    int64
 37  Unique Outlinks                 444 non-null    int64
 38  External Outlinks               444 non-null    int64
 39  Unique External Outlinks        444 non-null    int64
 40  Closest Similarity Match        0 non-null      float64
 41  No. Near Duplicates             0 non-null      float64
 42  Spelling Errors                 0 non-null      float64
 43  Grammar Errors                  0 non-null      float64
 44  Hash                            182 non-null    object
 45  Response Time                   444 non-null    float64
 46  Last Modified                   427 non-null    object
 47  Redirect URL                    0 non-null      float64
 48  Redirect Type                   0 non-null      float64
 49  URL Encoded Address             444 non-null    object
dtypes: float64(17), int64(18), object(15)
memory usage: 173.6+ KB
```

In [99]:

```
df2.columns
```

Out[99]:

```
Index(['Address', 'Content', 'Status Code', 'Status', 'Indexability',
       'Indexability Status', 'Title 1', 'Title 1 Length',
       'Title 1 Pixel Width', 'Meta Description 1',
       'Meta Description 1 Length', 'Meta Description 1 Pixel Width',
       'Meta Keyword 1', 'Meta Keywords 1 Length', 'H1-1', 'H1-1 length',
       'H2-1', 'H2-1 length', 'H2-2', 'H2-2 length', 'Meta Robots 1',
       'X-Robots-Tag 1', 'Meta Refresh 1', 'Canonical Link Element 1',
       'rel="next" 1', 'rel="prev" 1', 'HTTP rel="next" 1',
       'HTTP rel="prev" 1', 'Size (bytes)', 'Word Count', 'Text Ratio',
       'Crawl Depth', 'Link Score', 'Inlinks', 'Unique Inlinks', '% of Tota
l',
       'Outlinks', 'Unique Outlinks', 'External Outlinks',
       'Unique External Outlinks', 'Closest Similarity Match',
       'No. Near Duplicates', 'Spelling Errors', 'Grammar Errors', 'Hash',
       'Response Time', 'Last Modified', 'Redirect URL', 'Redirect Type',
       'URL Encoded Address'],
      dtype='object')
```

In [100]:

```
df2.describe()
```

Out[100]:

	Status Code	Title 1 Length	Title 1 Pixel Width	Meta Description 1 Length	Meta Description 1 Pixel Width	Meta Keywords 1 Length	H1-1 length	
count	444.000000	444.000000	444.000000	444.000000	444.000000	444.000000	444.000000	4
mean	238.556306	19.081081	177.108108	61.036036	387.736486	23.869369	12.211712	
std	79.909367	24.989437	230.393533	84.273357	534.954234	47.639742	17.222623	
min	200.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	200.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	200.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	200.000000	46.000000	433.000000	161.000000	1016.000000	0.000000	23.000000	
max	404.000000	77.000000	727.000000	216.000000	1376.000000	158.000000	59.000000	

8 rows × 35 columns

In [101]:

```
#mindadda dataset loading
df3 = pd.read_csv('mindadda_data.csv')
df3.sample(5)
```

Out[101]:

	Address	Content	Status Code	Status	Indexability	Indexabilit Statu
19	http://mindadda.com/category/super-memory/	text/html; charset=UTF-8	200	OK	Indexable	NaI
167	http://mindadda.com/category/personality-devel...	text/html; charset=UTF-8	200	OK	Indexable	NaI
125	http://mindadda.com/tag/employee-skills/	text/html; charset=UTF-8	200	OK	Indexable	NaI
190	https://mindadda.com/tag/email-writing-skils/	text/html; charset=UTF-8	200	OK	Indexable	NaI
10	http://mindadda.com/terms-conditions/	text/html; charset=UTF-8	200	OK	Indexable	NaI

5 rows × 51 columns

In [102]:

```
df3.shape
```

Out[102]:

```
(237, 51)
```

In [103]:

```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 237 entries, 0 to 236
Data columns (total 51 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Address                         237 non-null    object 
 1   Content                         237 non-null    object 
 2   Status Code                     237 non-null    int64  
 3   Status                          237 non-null    object 
 4   Indexability                    237 non-null    object 
 5   Indexability Status             8 non-null      object 
 6   Title 1                         231 non-null    object 
 7   Title 1 Length                  237 non-null    int64  
 8   Title 1 Pixel Width             237 non-null    int64  
 9   Meta Description 1              80 non-null     object 
 10  Meta Description 1 Length       237 non-null    int64  
 11  Meta Description 1 Pixel Width  237 non-null    int64  
 12  Meta Keyword 1                  0 non-null      float64
 13  Meta Keywords 1 Length          237 non-null    int64  
 14  H1-1                            216 non-null    object 
 15  H1-1 length                     237 non-null    int64  
 16  H2-1                            224 non-null    object 
 17  H2-1 length                     237 non-null    int64  
 18  H2-2                            101 non-null    object 
 19  H2-2 length                     237 non-null    int64  
 20  Meta Robots 1                   231 non-null    object 
 21  Meta Robots 2                   231 non-null    object 
 22  X-Robots-Tag 1                  0 non-null      float64
 23  Meta Refresh 1                  0 non-null      float64
 24  Canonical Link Element 1        231 non-null    object 
 25  rel="next" 1                    50 non-null     object 
 26  rel="prev" 1                    25 non-null     object 
 27  HTTP rel="next" 1               0 non-null      float64
 28  HTTP rel="prev" 1               0 non-null      float64
 29  Size (bytes)                    237 non-null    int64  
 30  Word Count                      237 non-null    int64  
 31  Text Ratio                      237 non-null    float64
 32  Crawl Depth                     237 non-null    int64  
 33  Link Score                      0 non-null      float64
 34  Inlinks                         237 non-null    int64  
 35  Unique Inlinks                  237 non-null    int64  
 36  % of Total                      237 non-null    float64
 37  Outlinks                        237 non-null    int64  
 38  Unique Outlinks                 237 non-null    int64  
 39  External Outlinks               237 non-null    int64  
 40  Unique External Outlinks        237 non-null    int64  
 41  Closest Similarity Match        0 non-null      float64
 42  No. Near Duplicates             0 non-null      float64
 43  Spelling Errors                 0 non-null      float64
 44  Grammar Errors                  0 non-null      float64
 45  Hash                            231 non-null    object 
 46  Response Time                   237 non-null    float64
 47  Last Modified                   229 non-null    object 
 48  Redirect URL                    3 non-null      object 
 49  Redirect Type                   3 non-null      object 
 50  URL Encoded Address             237 non-null    object 
```

```
dtypes: float64(13), int64(18), object(20)
memory usage: 94.6+ KB
```

In [104]:

```
df3.columns
```

Out[104]:

```
Index(['Address', 'Content', 'Status Code', 'Status', 'Indexability',
       'Indexability Status', 'Title 1', 'Title 1 Length',
       'Title 1 Pixel Width', 'Meta Description 1',
       'Meta Description 1 Length', 'Meta Description 1 Pixel Width',
       'Meta Keyword 1', 'Meta Keywords 1 Length', 'H1-1', 'H1-1 length',
       'H2-1', 'H2-1 length', 'H2-2', 'H2-2 length', 'Meta Robots 1',
       'Meta Robots 2', 'X-Robots-Tag 1', 'Meta Refresh 1',
       'Canonical Link Element 1', 'rel="next" 1', 'rel="prev" 1',
       'HTTP rel="next" 1', 'HTTP rel="prev" 1', 'Size (bytes)', 'Word Coun
t',
       'Text Ratio', 'Crawl Depth', 'Link Score', 'Inlinks', 'Unique Inlink
s',
       '% of Total', 'Outlinks', 'Unique Outlinks', 'External Outlinks',
       'Unique External Outlinks', 'Closest Similarity Match',
       'No. Near Duplicates', 'Spelling Errors', 'Grammar Errors', 'Hash',
       'Response Time', 'Last Modified', 'Redirect URL', 'Redirect Type',
       'URL Encoded Address'],
      dtype='object')
```

In [105]:

```
df3.describe()
```

Out[105]:

	Status Code	Title 1 Length	Title 1 Pixel Width	Meta Description 1 Length	Meta Description 1 Pixel Width	Meta Keyword 1	Meta Keywords 1 Length	
count	237.000000	237.000000	237.000000	237.000000	237.000000	0.0	237.0	2
mean	203.860759	106.683544	991.497890	260.215190	1663.713080	NaN	0.0	
std	25.371678	38.753309	356.400016	507.043265	3217.122231	NaN	0.0	
min	200.000000	0.000000	0.000000	0.000000	0.000000	NaN	0.0	
25%	200.000000	112.000000	1047.000000	0.000000	0.000000	NaN	0.0	
50%	200.000000	121.000000	1131.000000	0.000000	0.000000	NaN	0.0	
75%	200.000000	126.000000	1168.000000	464.000000	2970.000000	NaN	0.0	
max	404.000000	166.000000	1532.000000	3503.000000	22069.000000	NaN	0.0	

8 rows × 31 columns

MISSING column compared to other two datasets

In [17]:

```
#particular in mindadda data that not present in byjus data and midbrain academy
[i for i in df3.columns if i not in df1.columns and i not in df2.columns]
```

Out[17]:

```
['Meta Robots 2']
```

In [107]:

```
for i in df3['Meta Robots 2']:
    print(i)
```

```
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:

-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:
-1
```

Picking important columns for further process

In [108]:

```
df1 = df1[['Address','Content', 'Status Code', 'Status','Title 1', 'Title 1 Length','Meta D
           'Meta Description 1 Length','Meta Keyword 1', 'Meta Keywords 1 Length', 'H1-1','
           'H2-2', 'H2-2 length', 'Word Count', 'Text Ratio', 'Crawl Depth', 'Link Score',
           'Unique Inlinks', '% of Total', 'Outlinks', 'Unique Outlinks', 'External Outlink
           'Unique External Outlinks', 'Closest Similarity Match','Response Time']]
```

In [109]:

```
df1.sample(5)
```

Out[109]:

	Address	Content	Status Code	Status	Title 1	Title 1 Length	Meta Description 1	D
381	https://byjus.com/mbose/	text/html; charset=UTF-8	200	OK	MBOSE: Meghalaya Board Of School Education Det...	59	MBOSE or Meghalaya Board Of School Education d...	
318	https://byjus.com/cbse/cbse-sample-papers-for-...	text/html; charset=UTF-8	200	OK	CBSE Sample Papers for Class 12 - Download PDF...	76	You can find the CBSE Sample Papers for Class ...	
102	https://byjus.com/byjus-classes/	text/html; charset=UTF-8	200	OK	BYJU'S Classes : Online Classes, Learning App ...	70	BYJU'S classes is a combination of online clas...	
271	https://byjus.com/jee/jee-main-question-paper/	text/html; charset=UTF-8	200	OK	JEE Main Question Papers - Download Past Quest...	73	JEE Main Question Papers - Students can downlo...	
414	https://byjus.com/ncert-exemplar-solutions-cla...	text/html; charset=UTF-8	200	OK	NCERT Exemplar Class 6 | Download Chapter-wise...	56	NCERT Exemplar Solutions Class 6 are available...	

5 rows × 27 columns

In [110]:

```
df2 = df2[['Address','Content', 'Status Code', 'Status','Title 1', 'Title 1 Length','Meta D
         'Meta Description 1 Length','Meta Keyword 1', 'Meta Keywords 1 Length', 'H1-1','
         'H2-2', 'H2-2 length', 'Word Count', 'Text Ratio', 'Crawl Depth', 'Link Score',
         'Unique Inlinks', '% of Total', 'Outlinks', 'Unique Outlinks', 'External Outlink
         'Unique External Outlinks', 'Closest Similarity Match','Response Time']]
```

In [111]:

```
df2.sample(5)
```

Out[111]:

	Address	Content	Status Code	Status	Title 1	Title 1 Length	Descri
416	http://midbrainacademy.in/privacy-policy.html	text/html	200	OK	Privacy Policy	14	V �RA ACADE commit re
417	https://midbrainacademy.in/blog/vedic-maths-fr...	text/html	404	Not Found	NaN	0	
253	http://midbrainacademy.in/	text/html	200	OK	Midbrain Activation Franchise | Best Mid Brain...	68	Mic Activ Franc Mic a
301	https://midbrainacademy.in/blog/nlp-training-f...	text/html	404	Not Found	NaN	0	
19	https://www.midbrainacademy.in/midbrain-activa...	text/html	200	OK	Midbrain Activation in Jabalpur - RAJMIN ACADEMY	48	Mic Activat Jab Call

5 rows × 27 columns

In [112]:

```
df3 = df3[['Address','Content', 'Status Code', 'Status','Title 1', 'Title 1 Length','Meta D
         'Meta Description 1 Length','Meta Keyword 1', 'Meta Keywords 1 Length', 'H1-1','
         'H2-2', 'H2-2 length', 'Word Count', 'Text Ratio', 'Crawl Depth', 'Link Score',
         'Unique Inlinks', '% of Total', 'Outlinks', 'Unique Outlinks', 'External Outlink
         'Unique External Outlinks', 'Closest Similarity Match','Response Time']]
```

In [113]:

```
df3.sample(5)
```

Out[113]:

	Address	Content	Status Code	Status	Title 1	Title 1 Length	I
60	https://mindadda.com/about-us/	text/html; charset=UTF-8	200	OK	About Us - MindAdda | Your Empowerment Partner	46	I
34	https://mindadda.com/blogs/	text/html; charset=UTF-8	200	OK	Blogs & Posts - MindAdda	24	I
147	http://mindadda.com/author/vijay/page/2/	text/html; charset=UTF-8	200	OK	Vijay venkat, Author at MindAdda - Your Empowe...	134	
98	http://mindadda.com/tag/super-memory/	text/html; charset=UTF-8	200	OK	super memory Archives | MindAdda - Your Empowe...	120	
232	http://mindadda.com/tag/business-statergy/	text/html; charset=UTF-8	200	OK	Business Statergy Archives | MindAdda - Your E...	125	

5 rows × 27 columns

Title tag analaysis

In [114]:

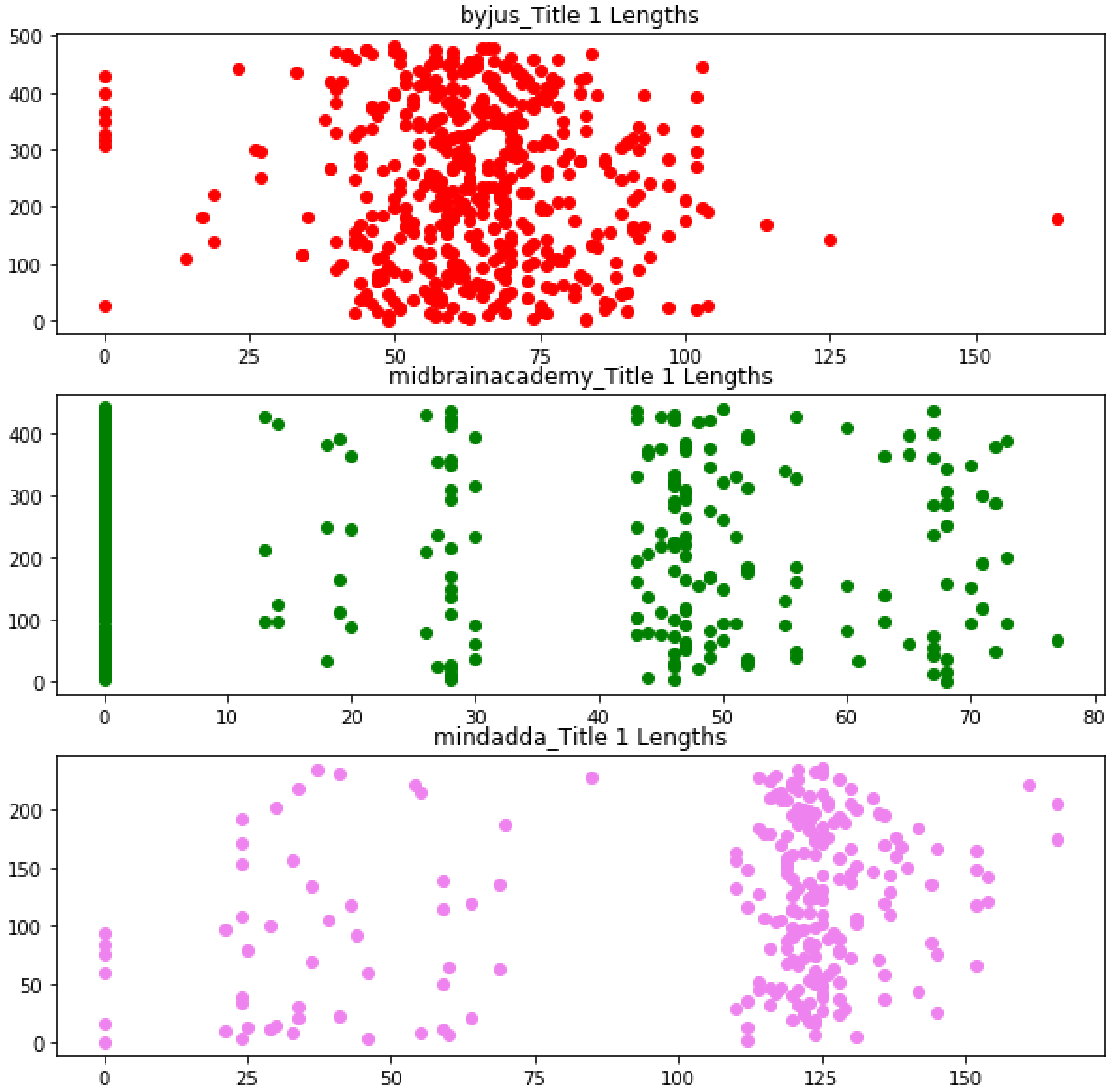
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['Title 1 Length'],range(0,len(df1['Title 1 Length'])),color='red')
ax1.set_title('byjus_Title 1 Lengths')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['Title 1 Length'],range(0,len(df2['Title 1 Length'])),color='green')
ax2.set_title('midbrainacademy_Title 1 Lengths')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['Title 1 Length'],range(0,len(df3['Title 1 Length'])),color='violet')
ax3.set_title('mindadda_Title 1 Lengths')

plt.show()
```



byjus_Title 1 Lengths

midbrainacademy_Title 1 Lengths

mindadda_Title 1 Lengths

In [115]:

```
print("\n*************** BYJUS Title tag ANALYSIS ***************\n\n")
print('\tTotal count of title rows:\t',df1['Title 1 Length'].count())
print('\n\tAverage of title length:\t',df1[df1['Title 1 Length']>0]['Title 1 Length'].mean(
print('\n\tminimum value of title length:\t',df1['Title 1 Length'].min())
print('\n\tmaximum value of title length:\t',df1['Title 1 Length'].max())
print('\n\n','*'*60)
```

```

*************** BYJUS Title tag ANALYSIS ***************


        Total count of title rows:       481

        Average of title length:         64.83086680761099

        minimum value of title length:   0

        maximum value of title length:   164


 ************************************************************
```

In [116]:

```
print("\n*************** MIDBRAINACADEMY Title tag ANALYSIS ***************\n\n")
print('\tTotal count of title rows:\t', df2['Title 1 Length'].count())
print('\n\tAverage of title length:\t', df2[df2['Title 1 Length']>0]['Title 1 Length'].mean
print('\n\tminimum value of title length:\t', df2['Title 1 Length'].min())
print('\n\tmaximum value of title length:\t', df2['Title 1 Length'].max())
print('\n\n','*'*60)
```

```

*************** MIDBRAINACADEMY Title tag ANALYSIS ***************


        Total count of title rows:       444

        Average of title length:         46.54945054945055

        minimum value of title length:   0

        maximum value of title length:   77


 ************************************************************
```

In [117]:

```
print("\n*************** MINDADDA Title tag ANALYSIS ***************\n\n")
print('\tTotal count of title rows:\t',df3['Title 1 Length'].count())
print('\n\tAverage of title length:\t',df3[df3['Title 1 Length']>0]['Title 1 Length'].mean(
print('\n\tminimum value of title length:\t',df3['Title 1 Length'].min())
print('\n\tmaximum value of title length:\t',df3['Title 1 Length'].max())
print('\n\n','*'*60)
```

```

*************** MINDADDA Title tag ANALYSIS ***************


        Total count of title rows:       237

        Average of title length:         109.45454545454545

        minimum value of title length:   0

        maximum value of title length:   166


 ************************************************************
```

Meta Description Analysis

In [118]:

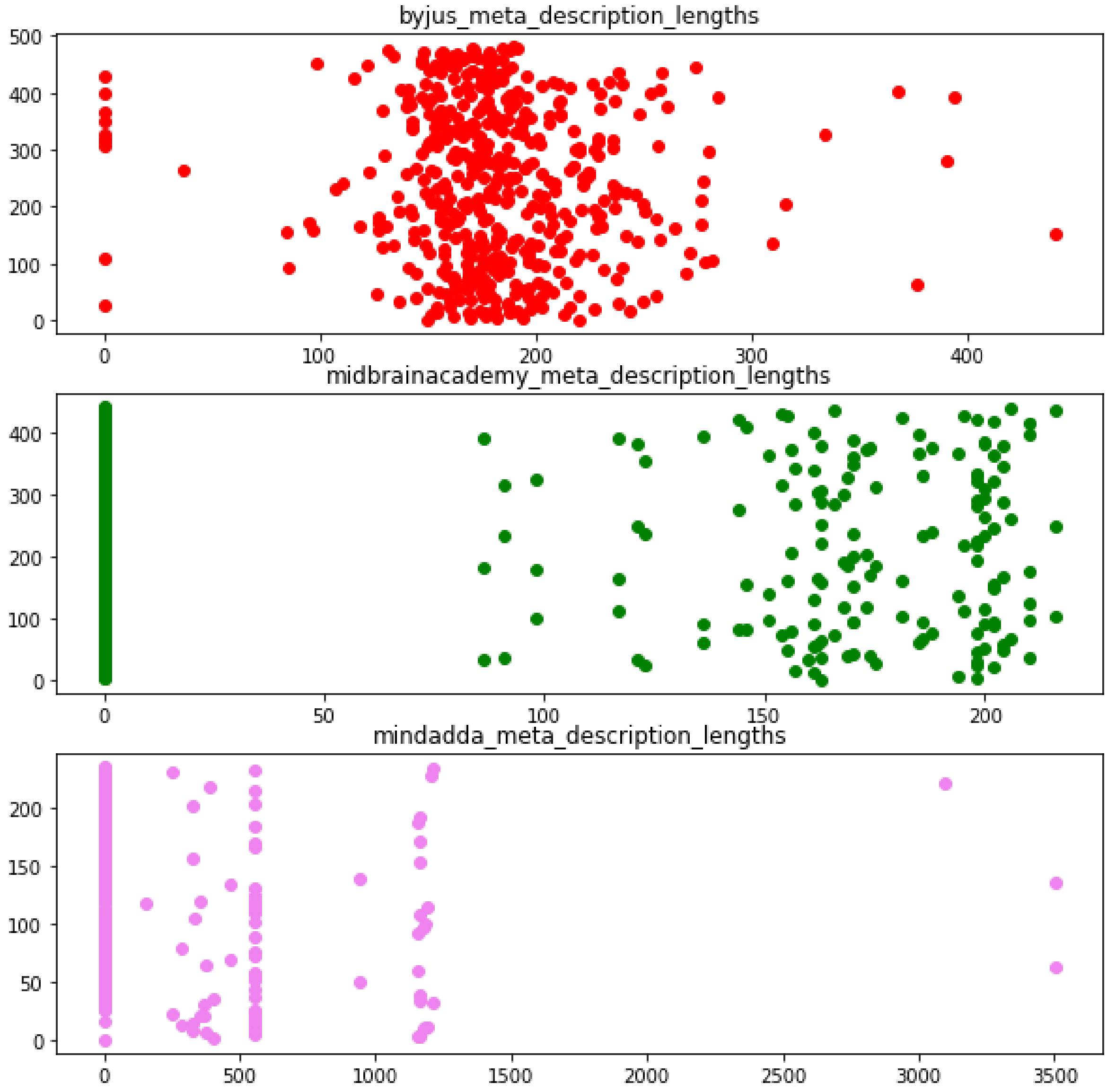
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['Meta Description 1 Length'],range(0,len(df1['Meta Description 1 Length'])))
ax1.set_title('byjus_meta_description_lengths')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['Meta Description 1 Length'],range(0,len(df2['Meta Description 1 Length'])))
ax2.set_title('midbrainacademy_meta_description_lengths')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['Meta Description 1 Length'],range(0,len(df3['Meta Description 1 Length'])))
ax3.set_title('mindadda_meta_description_lengths')

plt.show()
```



byjus_meta_description_lengths

midbrainacademy_meta_description_lengths

mindadda_meta_description_lengths

In [119]:

```
print("\n*************** BYJUS META DESCRIPTION ANALYSIS ***************\n\n")
print('\tTotal count of meta description rows:\t',df1['Meta Description 1 Length'].count())
print('\n\tAverage of meta description length:\t',df1[df1['Meta Description 1 Length']>0]['
print('\n\tminimum value of description length:\t',df1['Meta Description 1 Length'].min())
print('\n\tmaximum value of description length:\t',df1['Meta Description 1 Length'].max())
print('\n\n','*'*60)
```

```

*************** BYJUS META DESCRIPTION ANALYSIS ***************


        Total count of meta description rows:    481

        Average of meta description length:      185.78343949044586

        minimum value of description length:     0

        maximum value of description length:     441


 ************************************************************
```

In [120]:

```
print("\n*************** MIDBRAIN ACADEMY META DESCRIPTION ANALYSIS ***************\n\n")
print('\tTotal count of meta description rows:\t',df2['Meta Description 1 Length'].count())
print('\n\tAverage of meta description length:\t',df2[df2['Meta Description 1 Length']>0]['
print('\n\tminimum value of description length:\t',df2['Meta Description 1 Length'].min())
print('\n\tmaximum value of description length:\t',df2['Meta Description 1 Length'].max())
print('\n\n','*'*60)
```

```
*************** MIDBRAIN ACADEMY META DESCRIPTION ANALYSIS ***************


        Total count of meta description rows:   444

        Average of meta description length:     171.51898734177215

        minimum value of description length:    0

        maximum value of description length:    216


 ************************************************************
```

In [121]:

```
print("\n*************** MINDADDA META DESCRIPTION ANALYSIS ***************\n\n")
print('\tTotal count of meta description rows:\t',df3['Meta Description 1 Length'].count())
print('\n\tAverage of meta description length:\t',df3[df3['Meta Description 1 Length']>0]['
print('\n\tminimum value of description length:\t',df3['Meta Description 1 Length'].min())
print('\n\tmaximum value of description length:\t',df3['Meta Description 1 Length'].max())
print('\n\n','*'*60)
```

```
*************** MINDADDA META DESCRIPTION ANALYSIS ***************


        Total count of meta description rows:   237

        Average of meta description length:     770.8875

        minimum value of description length:    0

        maximum value of description length:    3503


 ************************************************************
```

Meta Keywords Analysis

In [122]:

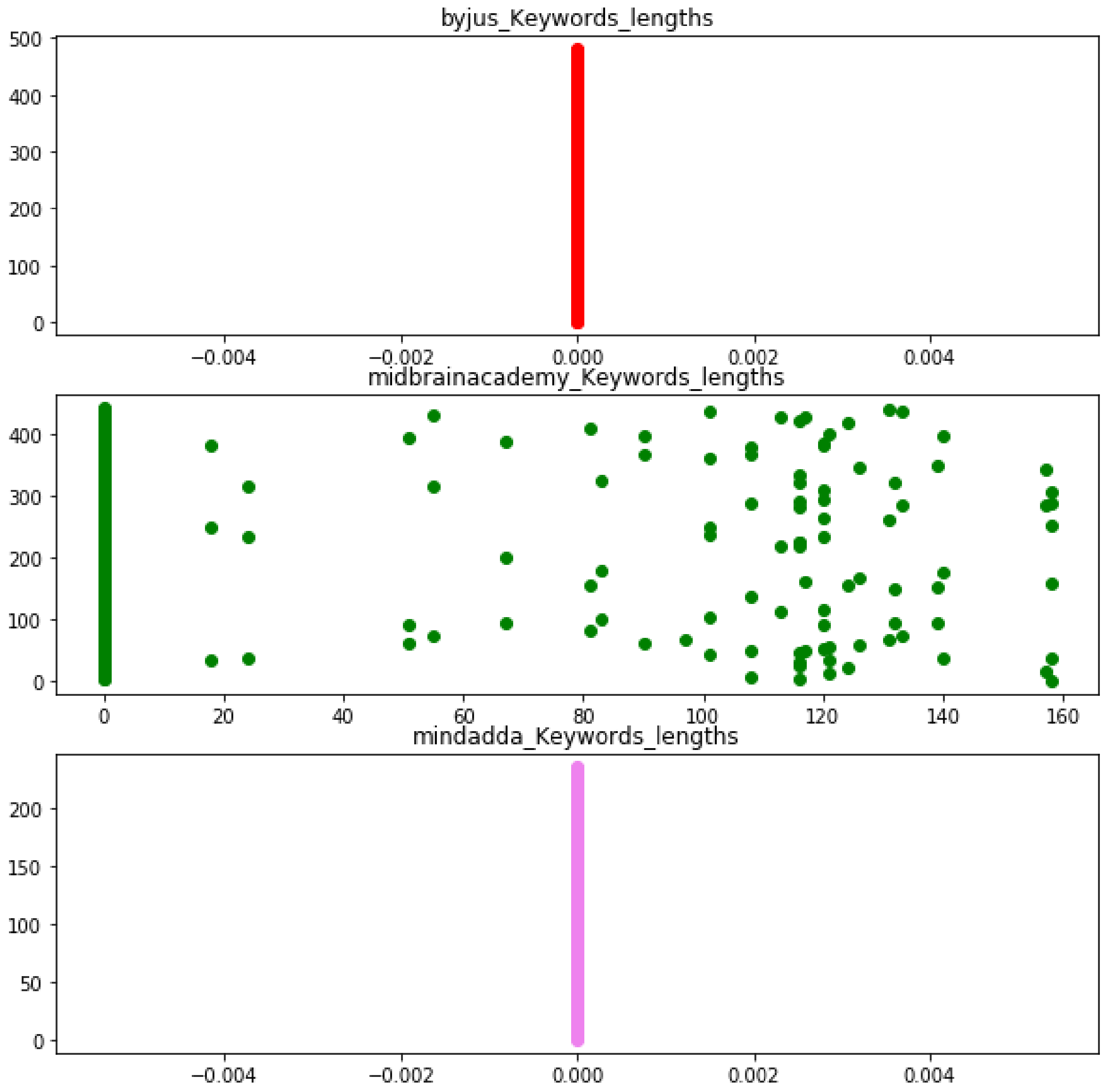
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['Meta Keywords 1 Length'],range(0,len(df1['Meta Keywords 1 Length'])),color
ax1.set_title('byjus_Keywords_lengths')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['Meta Keywords 1 Length'],range(0,len(df2['Meta Keywords 1 Length'])),color
ax2.set_title('midbrainacademy_Keywords_lengths')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['Meta Keywords 1 Length'],range(0,len(df3['Meta Keywords 1 Length'])),color
ax3.set_title('mindadda_Keywords_lengths')

plt.show()
```



In [123]:

```
df1['Meta Keywords 1 Length'].describe()
```

Out[123]:

```
count    481.0
mean       0.0
std        0.0
min        0.0
25%        0.0
50%        0.0
75%        0.0
max        0.0
Name: Meta Keywords 1 Length, dtype: float64
```

In [126]:

```
df2['Meta Keywords 1 Length'].describe()
```

Out[126]:

```
count    444.000000
mean      23.869369
std       47.639742
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max      158.000000
Name: Meta Keywords 1 Length, dtype: float64
```

In [127]:

```
df3['Meta Keywords 1 Length'].describe()
```

Out[127]:

```
count    237.0
mean       0.0
std        0.0
min        0.0
25%        0.0
50%        0.0
75%        0.0
max        0.0
Name: Meta Keywords 1 Length, dtype: float64
```

In [128]:

```
df2.sample(5)
```

Out[128]:

	Address	Content	Status Code	Status	Title
244	https://midbrainacademy.in/css/shortcodes.css	text/css	200	OK	Na
393	http://midbrainacademy.in/cancellation-policy....	text/html	200	OK	Cancellati Poli
254	https://midbrainacademy.in/js/jquery-easing-1....	application/javascript	200	OK	Na
301	https://midbrainacademy.in/blog/nlp-training-f...	text/html	404	Not Found	Na
45	https://www.midbrainacademy.in/banner/booklet.gif	image/gif	200	OK	Na

5 rows × 27 columns

H1 Tag Analysis

In [129]:

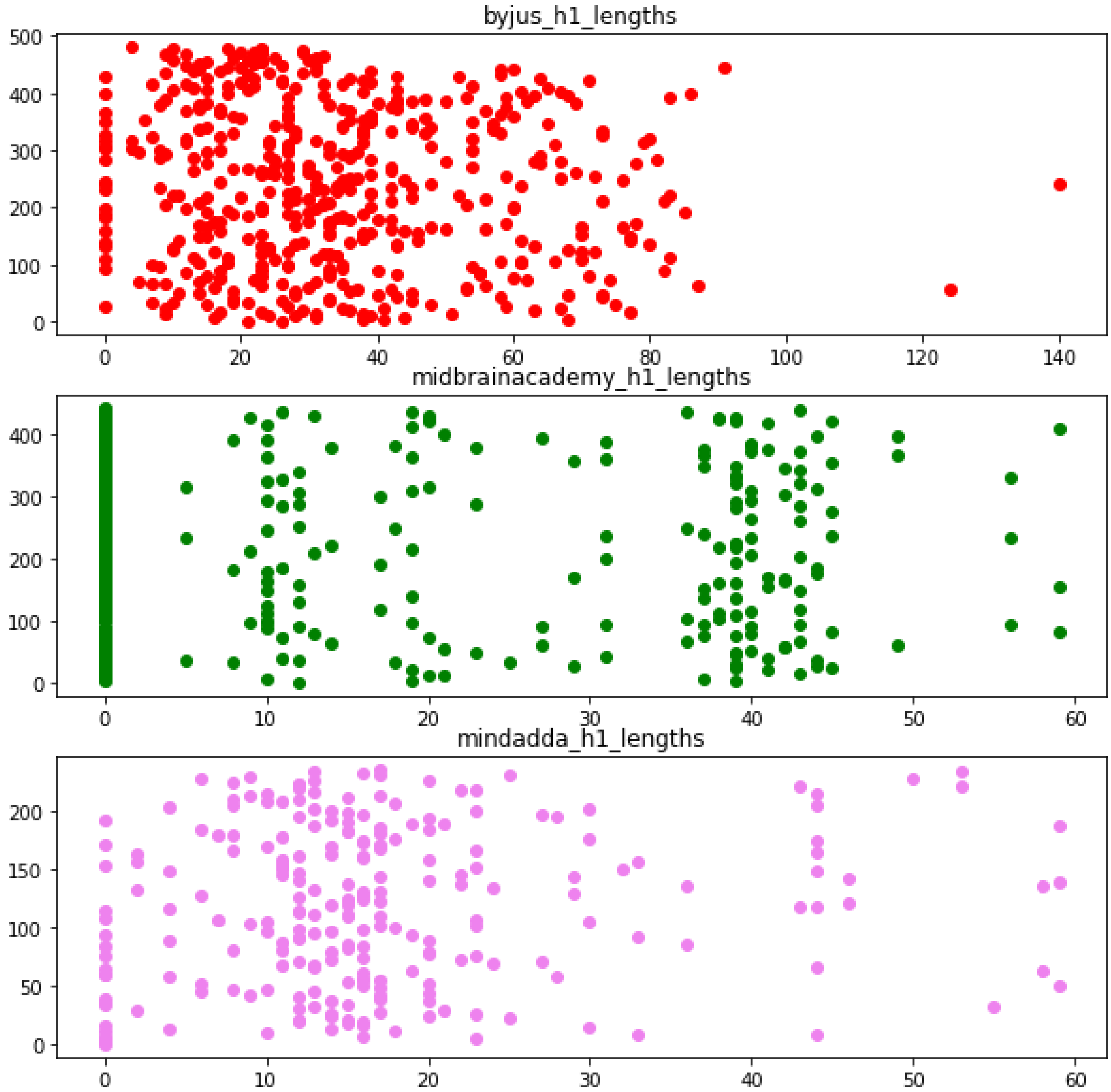
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['H1-1 length'],range(0,len(df1['H1-1 length'])),color='red')
ax1.set_title('byjus_h1_lengths')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['H1-1 length'],range(0,len(df2['H1-1 length'])),color='green')
ax2.set_title('midbrainacademy_h1_lengths')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['H1-1 length'],range(0,len(df3['H1-1 length'])),color='violet')
ax3.set_title('mindadda_h1_lengths')

plt.show()
```



In [130]:

```
print("\n*************** BYJUS H1 tag ANALYSIS ***************\n\n")
print('\tTotal count of H1 rows:\t\t',df1['H1-1 length'].count())
print('\n\tAverage of H1 length:\t\t',df1[df1['H1-1 length']>0]['H1-1 length'].mean())
print('\n\tminimum value of H1 length:\t',df1['H1-1 length'].min())
print('\n\tmaximum value of H1 length:\t',df1['H1-1 length'].max())
print('\n\n','*'*60)
```

```
*************** BYJUS H1 tag ANALYSIS ***************


        Total count of H1 rows:          481

        Average of H1 length:            35.72826086956522

        minimum value of H1 length:      0

        maximum value of H1 length:      140


 ************************************************************
```

In [131]:

```
print("\n*************** MIDBRAIN ACADEMY H1 tag ANALYSIS ***************\n\n")
print('\tTotal count of H1 rows:\t\t',df2['H1-1 length'].count())
print('\n\tAverage of H1 length:\t\t',df2[df2['H1-1 length']>0]['H1-1 length'].mean())
print('\n\tminimum value of H1 length:\t',df2['H1-1 length'].min())
print('\n\tmaximum value of H1 length:\t',df2['H1-1 length'].max())
print('\n\n','*'*60)
```

```

*************** MIDBRAIN ACADEMY H1 tag ANALYSIS ***************


        Total count of H1 rows:          444

        Average of H1 length:            29.791208791208792

        minimum value of H1 length:      0

        maximum value of H1 length:      59


 ************************************************************
```

In [132]:

```
print("\n*************** MINDADDA H1 tag ANALYSIS ***************\n\n")
print('\tTotal count of H1 rows:\t\t',df3['H1-1 length'].count())
print('\n\tAverage of H1 length:\t\t',df3[df3['H1-1 length']>0]['H1-1 length'].mean())
print('\n\tminimum value of H1 length:\t',df3['H1-1 length'].min())
print('\n\tmaximum value of H1 length:\t',df3['H1-1 length'].max())
print('\n\n','*'*60)
```

```

*************** MINDADDA H1 tag ANALYSIS ***************


        Total count of H1 rows:          237

        Average of H1 length:            18.875

        minimum value of H1 length:      0

        maximum value of H1 length:      59


 ************************************************************
```

H2 Tag Analysis

In [133]:

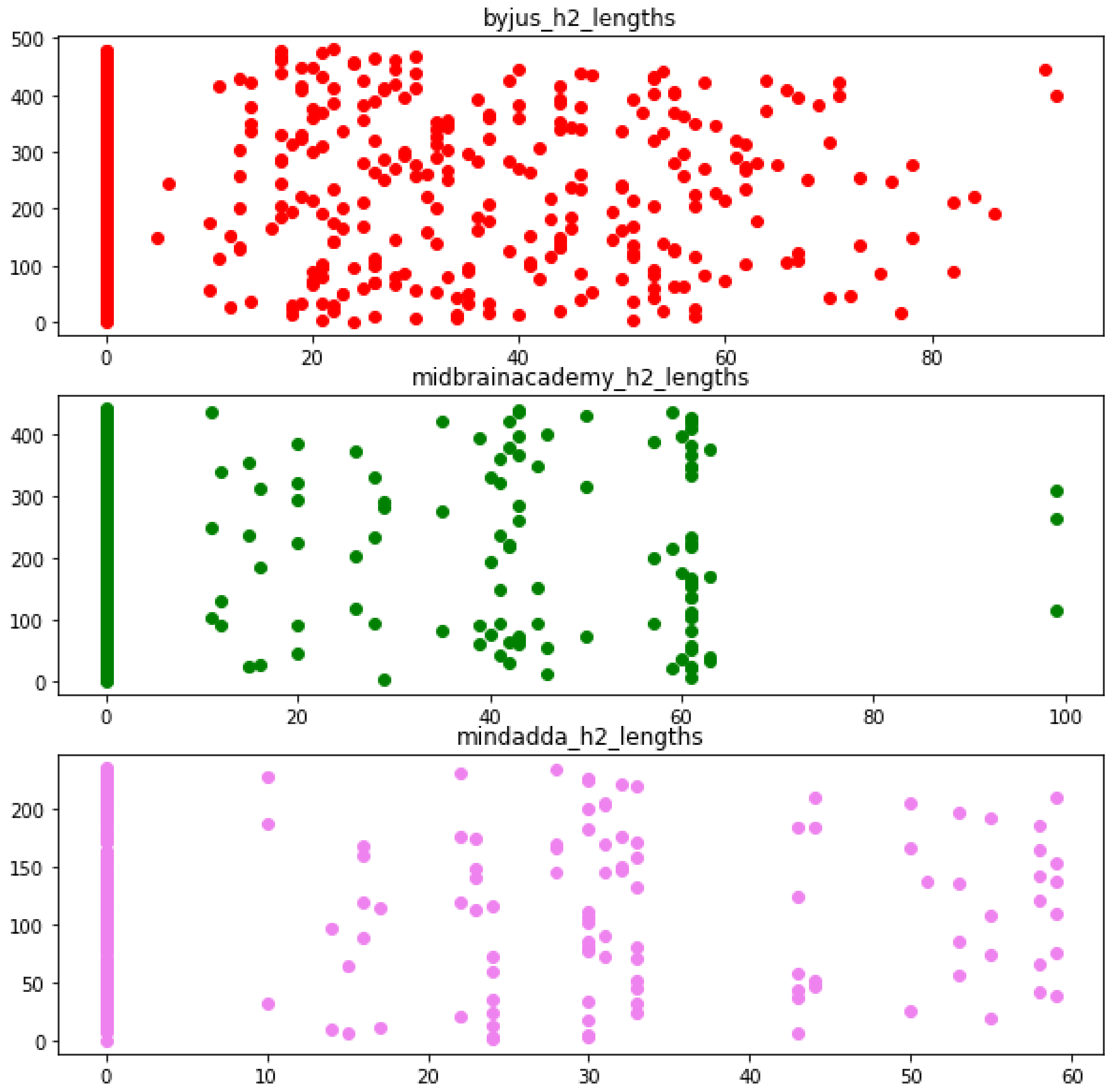
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['H2-2 length'],range(0,len(df1['H2-2 length'])),color='red')
ax1.set_title('byjus_h2_lengths')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['H2-2 length'],range(0,len(df2['H2-2 length'])),color='green')
ax2.set_title('midbrainacademy_h2_lengths')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['H2-2 length'],range(0,len(df3['H2-2 length'])),color='violet')
ax3.set_title('mindadda_h2_lengths')

plt.show()
```



In [134]:

```python
print("\n*************** BYJUS H2 tag ANALYSIS ***************\n\n")
print('\tTotal count of H2 rows:\t\t',df1['H2-2 length'].count())
print('\n\tAverage of H2 length:\t\t',df1[df1['H2-2 length']>0]['H2-2 length'].mean())
print('\n\tminimum value of H2 length:\t',df1['H2-2 length'].min())
print('\n\tmaximum value of H2 length:\t',df1['H2-2 length'].max())
print('\n\n','*'*60)
```

```
*************** BYJUS H2 tag ANALYSIS ***************


        Total count of H2 rows:          481

        Average of H2 length:            38.50877192982456

        minimum value of H2 length:      0

        maximum value of H2 length:      92


 ************************************************************
```

In [135]:

```python
print("\n*************** MIDBRAIN ACADEMY H2 tag ANALYSIS ***************\n\n")
print('\tTotal count of H2 rows:\t\t',df2['H2-2 length'].count())
print('\n\tAverage of H2 length:\t\t',df2[df2['H2-2 length']>0]['H2-2 length'].mean())
print('\n\tminimum value of H2 length:\t',df2['H2-2 length'].min())
print('\n\tmaximum value of H2 length:\t',df2['H2-2 length'].max())
print('\n\n','*'*60)
```

```
*************** MIDBRAIN ACADEMY H2 tag ANALYSIS ***************


        Total count of H2 rows:          444

        Average of H2 length:            45.0

        minimum value of H2 length:      0

        maximum value of H2 length:      99


 ************************************************************
```

In [136]:

```
print("\n*************** MINDADDA H2 tag ANALYSIS ***************\n\n")
print('\tTotal count of H2 rows:\t\t',df3['H2-2 length'].count())
print('\n\tAverage of H2 length:\t\t',df3[df3['H2-2 length']>0]['H2-2 length'].mean())
print('\n\tminimum value of H2 length:\t',df3['H2-2 length'].min())
print('\n\tmaximum value of H2 length:\t',df3['H2-2 length'].max())
print('\n\n','*'*60)
```

```
*************** MINDADDA H2 tag ANALYSIS ***************


        Total count of H2 rows:          237

        Average of H2 length:            34.68316831683168

        minimum value of H2 length:      0

        maximum value of H2 length:      59


 ************************************************************
```

In [137]:

```
df2.sample(5)
```

Out[137]:

	Address	Content	Status Code	Status	Title 1	Title 1 Length
102	https://www.midbrainacademy.in/midbrain-activa...	text/html	200	OK	Midbrain Activation Franchise Cost in Delhi	43
159	https://www.midbrainacademy.in/midbrain-activa...	text/html; charset=iso-8859-1	403	Forbidden	NaN	0
249	https://midbrainacademy.in/midbrain-activation...	text/html	200	OK	Midbrain Activation Franchise Cost in Delhi	43
361	http://midbrainacademy.in/css/retina.css	text/css	200	OK	NaN	0
252	https://midbrainacademy.in/css/flexslider.css	text/css	200	OK	NaN	0

5 rows × 27 columns

Inlinks Analysis

Inlinks = Links on other websites that send traffic to your site.

Inlinks – Links on other websites that send traffic to your site.

- Also called "backlinks".
- Traditionally Google has counted them as "votes" for your site, so that they help your site's Google rankings and traffic.

In [138]:

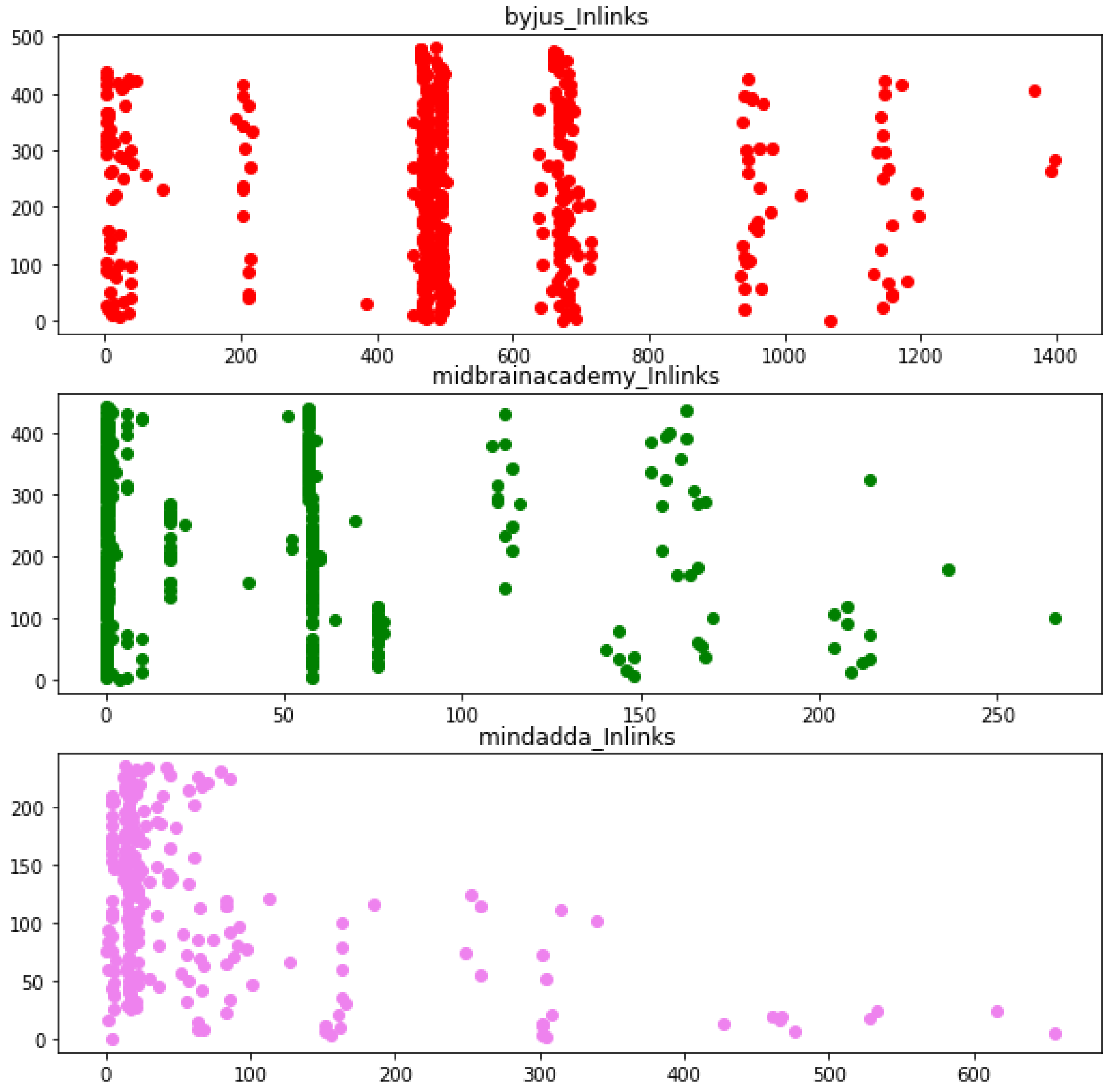
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['Inlinks'],range(0,len(df1['Inlinks'])),color='red')
ax1.set_title('byjus_Inlinks ')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['Inlinks'],range(0,len(df2['Inlinks'])),color='green')
ax2.set_title('midbrainacademy_Inlinks ')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['Inlinks'],range(0,len(df3['Inlinks'])),color='violet')
ax3.set_title('mindadda_Inlinks ')

plt.show()
```



In [139]:

```
print("\n*************** BYJUS Inlinks ANALYSIS ***************\n\n")
print('\tTotal count of Inlinks rows:\t',df1['Inlinks'].count())
print('\n\tAverage of Inlinks:\t\t',df1[df1['Inlinks']>0]['Inlinks'].mean())
print('\n\tminimum value of Inlinks:\t',df1['Inlinks'].min())
print('\n\tmaximum value of Inlinks:\t',df1['Inlinks'].max())
print('\n\n','*'*60)
```

```
*************** BYJUS Inlinks ANALYSIS ***************


        Total count of Inlinks rows:     481

        Average of Inlinks:              513.4968814968815

        minimum value of Inlinks:        1

        maximum value of Inlinks:        1397


 ************************************************************
```

In [140]:

```
print("\n*************** MIDBRAIN ACADEMY Inlinks ANALYSIS ***************\n\n")
print('\tTotal count of Inlinks rows:\t',df2['Inlinks'].count())
print('\n\tAverage of Inlinks:\t\t',df2[df2['Inlinks']>0]['Inlinks'].mean())
print('\n\tminimum value of Inlinks:\t',df2['Inlinks'].min())
print('\n\tmaximum value of Inlinks:\t',df2['Inlinks'].max())
print('\n\n','*'*60)
```

```
*************** MIDBRAIN ACADEMY Inlinks ANALYSIS ***************


        Total count of Inlinks rows:     444

        Average of Inlinks:              58.56678700361011

        minimum value of Inlinks:        0

        maximum value of Inlinks:        266


 ************************************************************
```

In [141]:

```
print("\n*************** MINDADDA Inlinks ANALYSIS ***************\n\n")
print('\tTotal count of Inlinks rows:\t',df3['Inlinks'].count())
print('\n\tAverage of Inlinks:\t\t',df3[df3['Inlinks']>0]['Inlinks'].mean())
print('\n\tminimum value of Inlinks:\t',df3['Inlinks'].min())
print('\n\tmaximum value of Inlinks:\t',df3['Inlinks'].max())
print('\n\n','*'*60)
```

```

*************** MINDADDA Inlinks ANALYSIS ***************


        Total count of Inlinks rows:     237

        Average of Inlinks:              67.0042372881356

        minimum value of Inlinks:        0

        maximum value of Inlinks:        656


 ************************************************************
```

In [142]:

```
df1.sample(5)
```

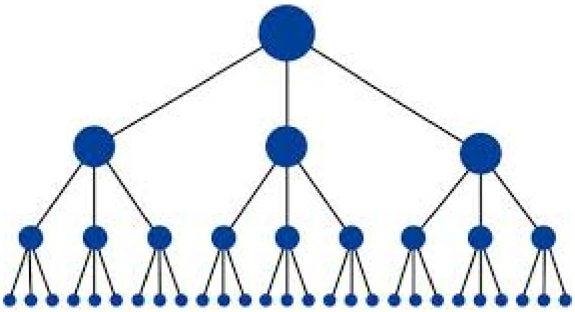
Out[142]:

	Address	Content	Status Code	Status	Title 1	Title 1 Length	Me Descriptic
379	https://byjus.com/maths/calculus/	text/html; charset=UTF-8	200	OK	Introduction to Calculus (Differential and Int...	61	Calculus is branch mathemati that deals
144	https://byjus.com/ncert-books-for-class-9/	text/html; charset=UTF-8	200	OK	NCERT Books for Class 9 All Subjects - Downloa...	56	NCEF Books f Class 9 a given he to help
143	https://byjus.com/ncert-solutions-class-10-sci...	text/html; charset=UTF-8	200	OK	NCERT Solutions Class 10 Science Chapter 12 El...	76	NCEF Solutioı Class ⸍ Scienı Chapter ⸍ El
347	https://byjus.com/ncert-solutions-class-9-math...	text/html; charset=UTF-8	200	OK	NCERT Solutions Class 9 Maths Chapter 15 Proba...	68	NCEF Solutioı Class Matl Chapter ⸍ Proba
219	https://byjus.com/ncert-solutions-class-10-sci...	text/html; charset=UTF-8	200	OK	NCERT Solutions Class 10 Science Chapter 5 Per...	92	Downloa free PDFs NCEF Solutioı Class 10

5 rows × 27 columns

Crawl Depth Analysis

Crawl depth is the extent to which a search engine indexes pages within a website.



In [143]:

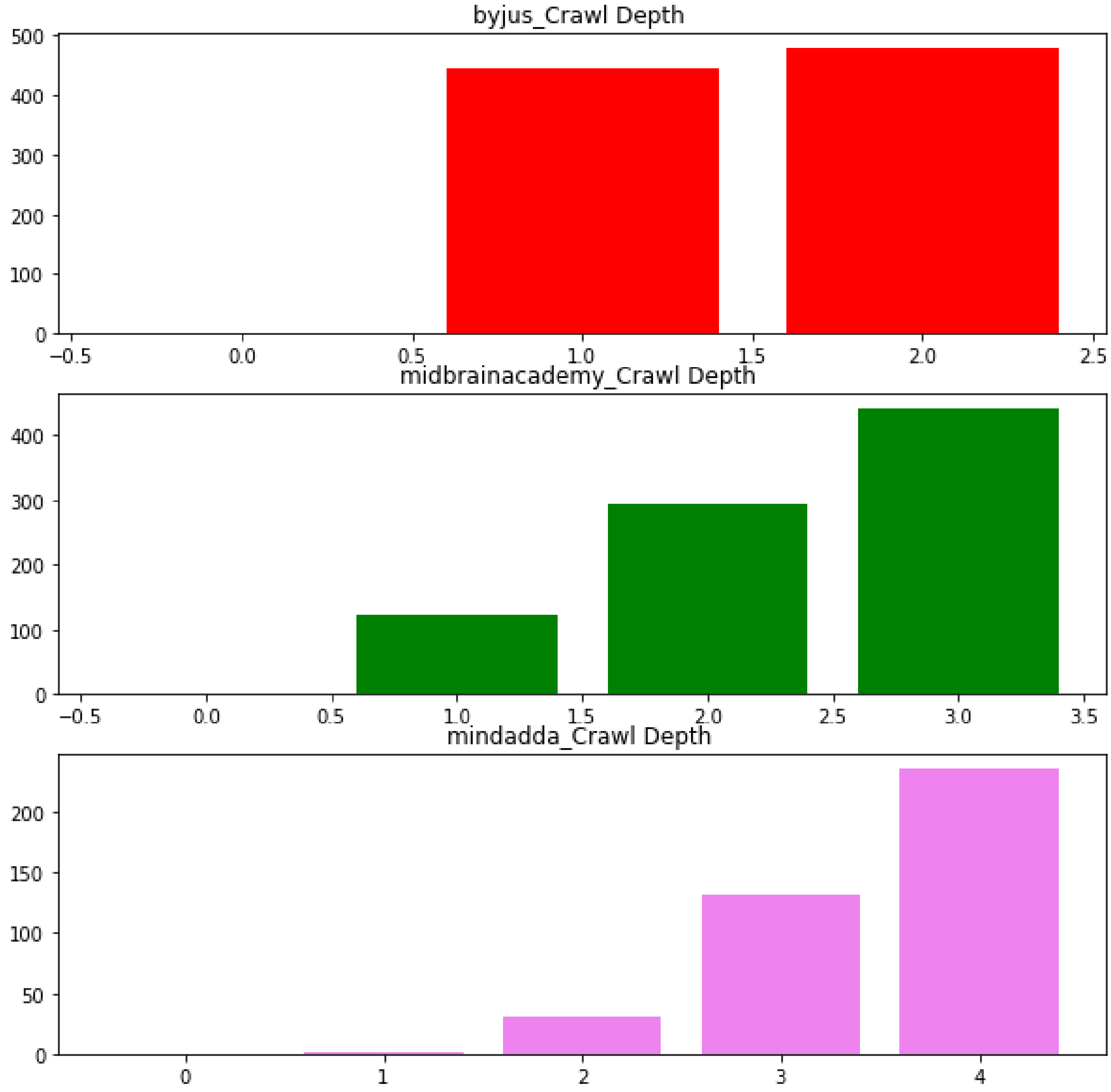
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.bar(df1['Crawl Depth'],range(0,len(df1['Crawl Depth'])),color='red')
ax1.set_title('byjus_Crawl Depth ')

ax2 = fig.add_subplot(312)
ax2.bar(df2['Crawl Depth'],range(0,len(df2['Crawl Depth'])),color='green')
ax2.set_title('midbrainacademy_Crawl Depth ')

ax3 = fig.add_subplot(313)
ax3.bar(df3['Crawl Depth'],range(0,len(df3['Crawl Depth'])),color='violet')
ax3.set_title('mindadda_Crawl Depth ')


plt.show()
```



In [144]:

```
print("\n*************** BYJUS Crawl Depth ANALYSIS ***************\n\n")
print('\tTotal count of Crawl Depth rows:\t',df1['Crawl Depth'].count())
print('\n\tAverage of Crawl Depth:\t\t\t',df1[df1['Crawl Depth']>0]['Crawl Depth'].mean())
print('\n\tminimum value of Crawl Depth:\t\t',df1['Crawl Depth'].min())
print('\n\tmaximum value of Crawl Depth:\t\t',df1['Crawl Depth'].max())
print('\n\n','*'*60)
```

```

*************** BYJUS Crawl Depth ANALYSIS ***************



        Total count of Crawl Depth rows:         481

        Average of Crawl Depth:                  1.0729166666666667

        minimum value of Crawl Depth:            0

        maximum value of Crawl Depth:            2


 ************************************************************
```

In [145]:

```
print("\n*************** MIDBRAIN ACADEMY Crawl Depth ANALYSIS ***************\n\n")
print('\tTotal count of Crawl Depth rows:\t',df2['Crawl Depth'].count())
print('\n\tAverage of Crawl Depth:\t\t\t',df2[df2['Crawl Depth']>0]['Crawl Depth'].mean())
print('\n\tminimum value of Crawl Depth:\t\t',df2['Crawl Depth'].min())
print('\n\tmaximum value of Crawl Depth:\t\t',df2['Crawl Depth'].max())
print('\n\n','*'*60)
```

```

*************** MIDBRAIN ACADEMY Crawl Depth ANALYSIS ***************



        Total count of Crawl Depth rows:         444

        Average of Crawl Depth:                  2.0632054176072234

        minimum value of Crawl Depth:            0

        maximum value of Crawl Depth:            3


 ************************************************************
```

In [146]:

```
print("\n*************** MINDADDA Crawl Depth ANALYSIS ***************\n\n")
print('\tTotal count of Crawl Depth rows:\t',df3['Crawl Depth'].count())
print('\n\tAverage of Crawl Depth:\t\t\t',df3[df3['Crawl Depth']>0]['Crawl Depth'].mean())
print('\n\tminimum value of Crawl Depth:\t\t',df3['Crawl Depth'].min())
print('\n\tmaximum value of Crawl Depth:\t\t',df3['Crawl Depth'].max())
print('\n\n','*'*60)
```

```
*************** MINDADDA Crawl Depth ANALYSIS ***************


        Total count of Crawl Depth rows:         237

        Average of Crawl Depth:                  3.330508474576271

        minimum value of Crawl Depth:            0

        maximum value of Crawl Depth:            4


 ************************************************************
```

In [147]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 444 entries, 0 to 443
Data columns (total 27 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Address                   444 non-null    object 
 1   Content                   444 non-null    object 
 2   Status Code               444 non-null    int64  
 3   Status                    444 non-null    object 
 4   Title 1                   182 non-null    object 
 5   Title 1 Length            444 non-null    int64  
 6   Meta Description 1        158 non-null    object 
 7   Meta Description 1 Length 444 non-null    int64  
 8   Meta Keyword 1            98 non-null     object 
 9   Meta Keywords 1 Length    444 non-null    int64  
 10  H1-1                      182 non-null    object 
 11  H1-1 length               444 non-null    int64  
 12  H2-2                      109 non-null    object 
 13  H2-2 length               444 non-null    int64  
 14  Word Count                444 non-null    int64  
 15  Text Ratio                444 non-null    float64
 16  Crawl Depth               444 non-null    int64  
 17  Link Score                0 non-null      float64
 18  Inlinks                   444 non-null    int64  
 19  Unique Inlinks            444 non-null    int64  
 20  % of Total                444 non-null    float64
 21  Outlinks                  444 non-null    int64  
 22  Unique Outlinks           444 non-null    int64  
 23  External Outlinks         444 non-null    int64  
 24  Unique External Outlinks  444 non-null    int64  
 25  Closest Similarity Match  0 non-null      float64
 26  Response Time             444 non-null    float64
dtypes: float64(5), int64(14), object(8)
memory usage: 93.8+ KB
```

Outlinks Analysis

Outlinks” shows the list of external URLs that were clicked by our visitors from our website pages.

In [148]:

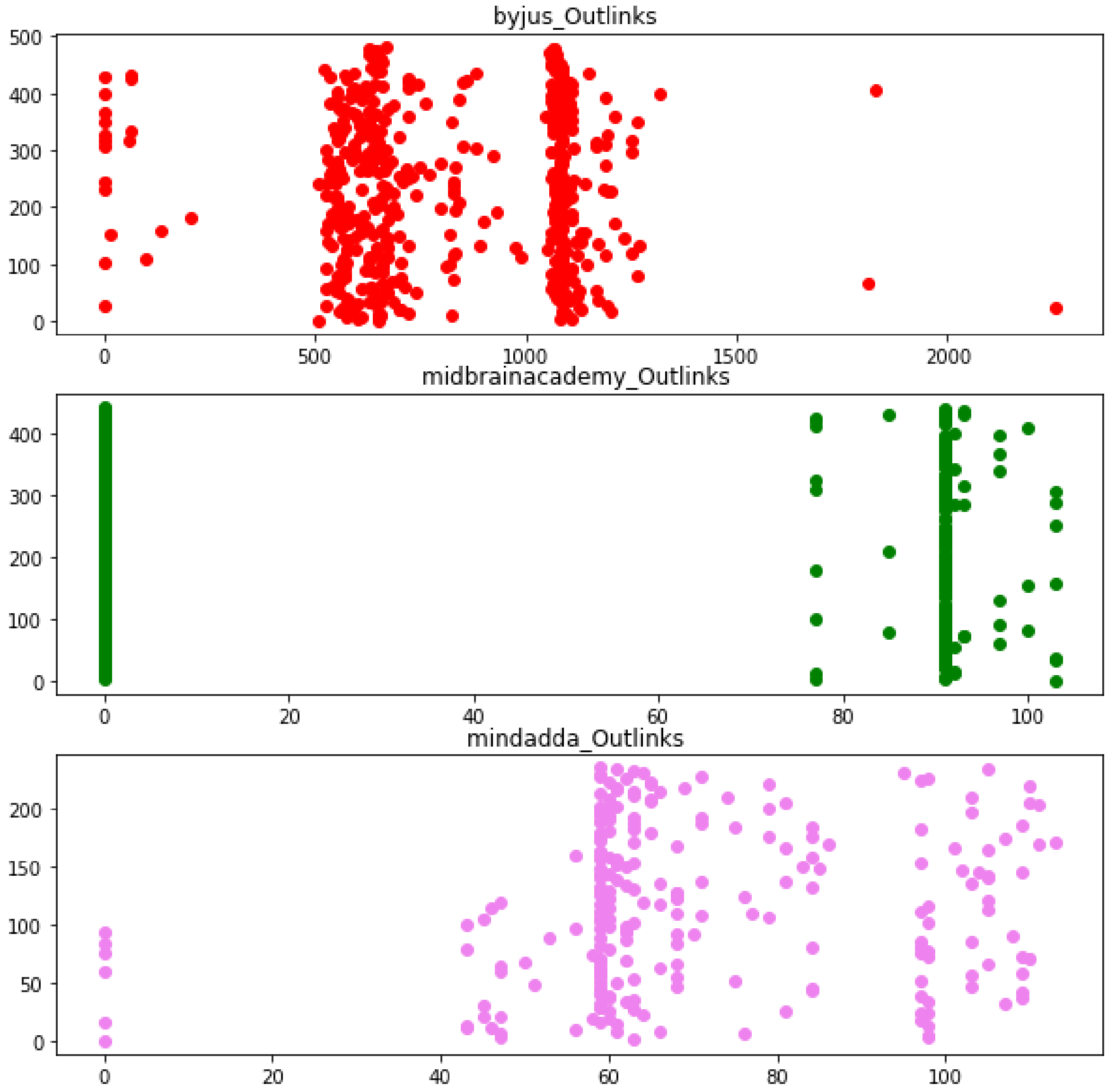
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['Outlinks'],range(0,len(df1['Outlinks'])),color='red')
ax1.set_title('byjus_Outlinks ')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['Outlinks'],range(0,len(df2['Outlinks'])),color='green')
ax2.set_title('midbrainacademy_Outlinks ')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['Outlinks'],range(0,len(df3['Outlinks'])),color='violet')
ax3.set_title('mindadda_Outlinks ')

plt.show()
```



In [149]:

```
print("\n*************** BYJUS Outlinks ANALYSIS ***************\n\n")
print('\tTotal count of Outlinks rows :\t',df1['Outlinks'].count())
print('\n\tAverage of Outlinks :\t\t',df1[df1['Outlinks']>0]['Outlinks'].mean())
print('\n\tminimum value of Outlinks :\t',df1['Outlinks'].min())
print('\n\tmaximum value of Outlinks :\t',df1['Outlinks'].max())
print('\n\n','*'*60)
```

```
*************** BYJUS Outlinks ANALYSIS ***************


        Total count of Outlinks rows :   481

        Average of Outlinks :            833.4861995753715

        minimum value of Outlinks :      0

        maximum value of Outlinks :      2255


 ************************************************************
```

In [150]:

```
print("\n*************** MIDBRAIN ACADEMY Outlinks ANALYSIS ***************\n\n")
print('\tTotal count of Outlinks rows:\t',df2['Outlinks'].count())
print('\n\tAverage of Outlinks :\t\t',df2[df2['Outlinks']>0]['Outlinks'].mean())
print('\n\tminimum value of Outlinks :\t',df2['Outlinks'].min())
print('\n\tmaximum value of Outlinks :\t',df2['Outlinks'].max())
print('\n\n','*'*60)
```

```
*************** MIDBRAIN ACADEMY Outlinks ANALYSIS ***************


        Total count of Outlinks rows:    444

        Average of Outlinks :            91.11731843575419

        minimum value of Outlinks :      0

        maximum value of Outlinks :      103


 ************************************************************
```

In [151]:

```
print("\n*************** MINDADDA Outlinks ANALYSIS ***************\n\n")
print('\tTotal count of Outlinks rows:\t',df3['Outlinks'].count())
print('\n\tAverage of Outlinks :\t\t',df3[df3['Outlinks']>0]['Outlinks'].mean())
print('\n\tminimum value of Outlinks :\t',df3['Outlinks'].min())
print('\n\tmaximum value of Outlinks :\t',df3['Outlinks'].max())
print('\n\n','*'*60)
```

```

*************** MINDADDA Outlinks ANALYSIS ***************


        Total count of Outlinks rows:    237

        Average of Outlinks :            71.91774891774892

        minimum value of Outlinks :      0

        maximum value of Outlinks :      113


 ************************************************************
```

In [152]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 481 entries, 0 to 480
Data columns (total 27 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Address                     481 non-null    object 
 1   Content                     481 non-null    object 
 2   Status Code                 481 non-null    int64  
 3   Status                      481 non-null    object 
 4   Title 1                     473 non-null    object 
 5   Title 1 Length              481 non-null    int64  
 6   Meta Description 1          471 non-null    object 
 7   Meta Description 1 Length   481 non-null    int64  
 8   Meta Keyword 1              0 non-null      float64
 9   Meta Keywords 1 Length      481 non-null    int64  
 10  H1-1                        460 non-null    object 
 11  H1-1 length                 481 non-null    int64  
 12  H2-2                        285 non-null    object 
 13  H2-2 length                 481 non-null    int64  
 14  Word Count                  481 non-null    int64  
 15  Text Ratio                  481 non-null    float64
 16  Crawl Depth                 481 non-null    int64  
 17  Link Score                  0 non-null      float64
 18  Inlinks                     481 non-null    int64  
 19  Unique Inlinks              481 non-null    int64  
 20  % of Total                  481 non-null    float64
 21  Outlinks                    481 non-null    int64  
 22  Unique Outlinks             481 non-null    int64  
 23  External Outlinks           481 non-null    int64  
 24  Unique External Outlinks    481 non-null    int64  
 25  Closest Similarity Match    0 non-null      float64
 26  Response Time               481 non-null    float64
dtypes: float64(6), int64(14), object(7)
memory usage: 101.6+ KB
```

Response Time Analysis

Response time : In website monitoring, response time is calculated as the total time taken to resolve the DNS, complete the TCP hand-shake, negotiate an SSL handshake (for HTTPS) and download the full HTML output from the server

What is a good response time for a website?

Average page size: 1.56MB; As you can see the location you are accessing the website plays a significant role in the response time, but the average is around 3.5 seconds. “The lower, the better” rule applies here as well, of course, but if your site is able to fully load under 2 seconds, that is a great result.

In [153]:

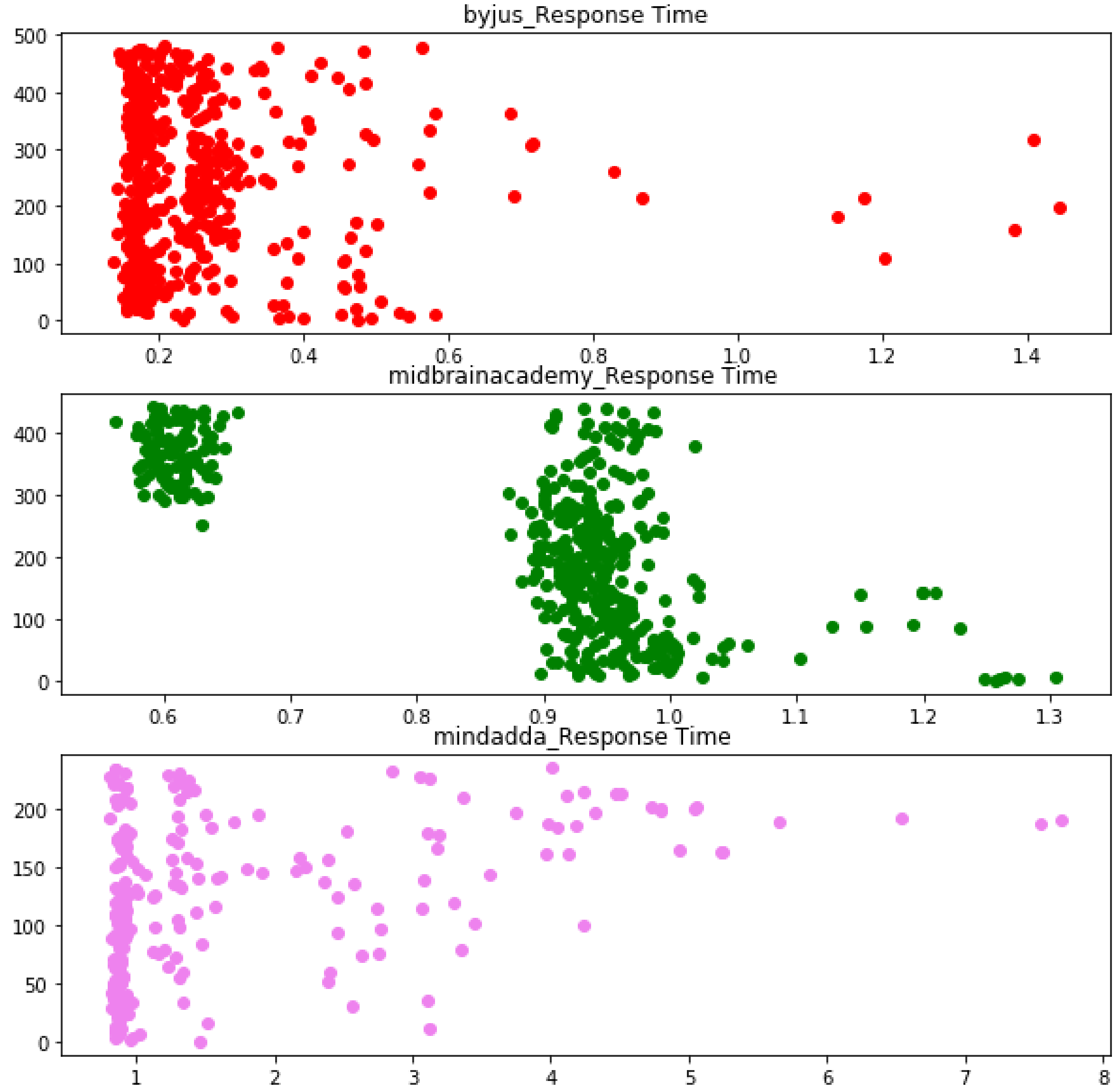
```
fig = plt.figure(figsize=(10,10))

ax1 = fig.add_subplot(311)
ax1.scatter(df1['Response Time'],range(0,len(df1['Response Time'])),color='red')
ax1.set_title('byjus_Response Time')

ax2 = fig.add_subplot(312)
ax2.scatter(df2['Response Time'],range(0,len(df2['Response Time'])),color='green')
ax2.set_title('midbrainacademy_Response Time ')

ax3 = fig.add_subplot(313)
ax3.scatter(df3['Response Time'],range(0,len(df3['Response Time'])),color='violet')
ax3.set_title('mindadda_Response Time ')

plt.show()
```



In [154]:

```
print("\n*************** BYJUS Response Time ANALYSIS ***************\n\n")
print('\tTotal count of Response Time rows:\t',df1['Response Time'].count())
print('\n\tAverage of Response Time :\t\t',df1[df1['Response Time']>0]['Response Time'].mea
print('\n\tminimum value of Response Time :\t',df1['Response Time'].min())
print('\n\tmaximum value of Response Time :\t',df1['Response Time'].max())
print('\n\n','*'*60)
```

```

*************** BYJUS Response Time ANALYSIS ***************


        Total count of Response Time rows:       481

        Average of Response Time :               0.25269854469854497

        minimum value of Response Time :         0.136

        maximum value of Response Time :         1.443


 ************************************************************
```

In [155]:

```
print("\n*************** MIDBRAIN ACADEMY Response Time ANALYSIS ***************\n\n")
print('\tTotal count of Response Time rows :\t',df2['Response Time'].count())
print('\n\tAverage of Response Time :\t\t',df2[df2['Response Time']>0]['Response Time'].mea
print('\n\tminimum value of Response Time :\t',df2['Response Time'].min())
print('\n\tmaximum value of Response Time :\t',df2['Response Time'].max())
print('\n\n','*'*60)
```

```

*************** MIDBRAIN ACADEMY Response Time ANALYSIS ***************


        Total count of Response Time rows :      444

        Average of Response Time :               0.8758671171171174

        minimum value of Response Time :         0.562

        maximum value of Response Time :         1.305


 ************************************************************
```

In [156]:

```
print("\n*************** MINDADDA Response Time ANALYSIS ***************\n\n")
print('\tTotal count of Response Time rows:\t',df3['Response Time'].count())
print('\n\tAverage of Response Time :\t\t',df3[df3['Response Time']>0]['Response Time'].mea
print('\n\tminimum value of Response Time :\t',df3['Response Time'].min())
print('\n\tmaximum value of Response Time :\t',df3['Response Time'].max())
print('\n\n','*'*60)
```

```
*************** MINDADDA Response Time ANALYSIS ***************


        Total count of Response Time rows:      237

        Average of Response Time :              1.6534177215189865

        minimum value of Response Time :        0.805

        maximum value of Response Time :        7.705


 ************************************************************
```

THANK YOU

-- Vamshikrishna Bandari

-- Mindadda - Machine Learning & Analytics Team