22-01-24-lab3

January 22, 2024

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[3]:
```
from urllib.request import urlopen
url = "https://matplotlib.org/cheatsheets/"
page = urlopen(url)
html_bytes = page.read()
html_bytes.decode("utf-8")
```

[3]:
```
'\n<!DOCTYPE html>\n<html>\n<head>\n<meta charset="utf-8" />\n<meta
name="viewport" content="width=device-width, initial-scale=1.0" /><meta
name="generator" content="Docutils 0.17.1: http://docutils.sourceforge.net/"
/>\n<title>Matplotlib cheatsheets &#8212; Visualization with
Python</title>\n<link
href="_static/styles/theme.css?digest=1999514e3f237ded88cf"
rel="stylesheet">\n<link href="_static/styles/pydata-sphinx-
theme.css?digest=1999514e3f237ded88cf" rel="stylesheet">\n<link rel="stylesheet"
href="_static/vendor/fontawesome/5.13.0/css/all.min.css">\n<link rel="preload"
as="font" type="font/woff2" crossorigin
href="_static/vendor/fontawesome/5.13.0/webfonts/fa-solid-900.woff2">\n<link
rel="preload" as="font" type="font/woff2" crossorigin
href="_static/vendor/fontawesome/5.13.0/webfonts/fa-brands-400.woff2">\n<link
rel="stylesheet" type="text/css" href="_static/pygments.css" />\n<link
rel="stylesheet" type="text/css" href="_static/css/style.css" />\n<link
rel="stylesheet" type="text/css" href="_static/css/normalize.css" />\n<link
rel="stylesheet" type="text/css" href="_static/css/landing.css" />\n<link
rel="preload" as="script" href="_static/scripts/pydata-sphinx-
theme.js?digest=1999514e3f237ded88cf">\n<script data-url_root="./"
id="documentation_options"
src="_static/documentation_options.js"></script>\n<script
src="_static/jquery.js"></script>\n<script
src="_static/underscore.js"></script>\n<script
src="_static/doctools.js"></script>\n<link rel="shortcut icon"
href="_static/favicon.ico" />\n<link rel="index" title="Index"
href="genindex.html" />\n<link rel="search" title="Search" href="search.html"
/>\n<meta name="viewport" content="width=device-width, initial-scale=1"
```

/>\n<meta name="docsearch:language" content="None">\n\n</head>\n<body data-spy="scroll" data-target="#bd-toc-nav" data-offset="80">\n \n<nav class="navbar navbar-light navbar-expand-lg bg-light fixed-top bd-navbar" id="navbar-main"> \n \n\n<img src="_static/images/logo2.svg" class="logo" alt="logo">\n\n \n<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar-collapsible" aria-controls="navbar-collapsible" aria-expanded="false" aria-label="Toggle navigation">\n\n</button>\n \n \n \n<ul id="navbar-main-elements" class="navbar-nav">\n<li class="nav-item">\nPlot types\n
\n<li class="nav-item">\nExamples\n
\n<li class="nav-item">\nTutorials\n
\n<li class="nav-item">\nReference\n
\n<li class="nav-item">\nUsage guide\n
\n<li class="nav-item">\nDevelop\n
\n<li class="nav-item">\nRelease notes\n
\n
\n \n \n \n \n<ul id="navbar-icon-links" class="navbar-nav" aria-label="Icon Links">\n<li class="nav-item">\n\n\n<label class="sr-only">gitter</label>\n\n
\n<li class="nav-item">\n\n\n<label class="sr-only">discourse</label>\n\n
\n<li class="nav-item">\n\n\n<label class="sr-only">GitHub</label>\n\n
\n<li class="nav-item">\n\n\n<label class="sr-only">twitter</label>\n\n
\n
\n \n \n \n \n</nav>\n \n \n \n \n \n \n On this page\n \n<nav id="bd-toc-nav">\n<ul class="visible

nav section-nav flex-column">\n<li class="toc-h2 nav-item toc-entry">\n\nCheatsheets\n\n
\n<li class="toc-h2 nav-item toc-entry">\n\nHandouts\n\n
\n<li class="toc-h2 nav-item toc-entry">\n\nContribute\n\n
\n
\n</nav>\n \n \n \n \n<main class="col-12 col-md-11 col-xl-8 py-md-5 pl-md-5 pr-md-4 bd-content" role="main">\n \n<section id="matplotlib-cheatsheets-and-handouts">\n<h1>Matplotlib cheatsheets and handouts¶</h1>\n<section id="cheatsheets">\n<h2>Cheatsheets¶</h2>\n \n \n\n \n \n\n \n \n Cheatsheets [pdf] \n</section>\n<section id="handouts">\n<h2>Handouts¶</h2>\n \n \n\n Beginner [pdf] \n \n \n\n Intermediate [pdf] \n \n \n\n Tips [pdf] \n \n \n</section>\n<section id="contribute">\n<h2>Contribute¶</h2>\n Issues, suggestions, or pull-requests gratefully accepted at\nmatplotlib/cheatsheets \n</section>\n</section>\n \n\n \n \n</main>\n \n \n<script src="_static/scripts/pydata-sphinx-theme.js?digest=1999514e3f237ded88cf"></script>\n<footer class="footer mt-5 mt-md-0">\n \n \n<p

```
class="copyright">\n&copy; Copyright 2012 - 2023 The Matplotlib development
team.<br>\n</p>\n</div>\n<div class="footer-item">\n<p class="sphinx-
version">\nCreated using <a href="http://sphinx-doc.org/">Sphinx</a>
4.4.0.<br>\n</p>\n</div>\n</div>\n</footer>\n<script defer src="https://static.c
loudflareinsights.com/beacon.min.js/v84a3a4012de94ce1a686ba8c167c359c16969738933
17" integrity="sha512-euoFGowhlaLqXsPWQ48qSkBSCFs3DPRyiwVu3FjR96cMPx+Fr+gpWRhIaf
cHwqwCqWS42RZhIudOvEI+Ckf6MA==" data-cf-beacon=\'{"rayId":"8497fccb8c5f7932","b"
:1,"version":"2024.1.0","token":"0175964edbd040b6ac0bff692bcb22ec"}\'
crossorigin="anonymous"></script>\n</body>\n</html>'
```

[4]:
```
def get_attr(s1,attr):
    r = 0
    for i in s1:
        if attr in s1:
            x = s1.find(attr)
            r += 1
            s1 = s1[x+3:]
    return r
```

0.1 pyton package – Communications

[5]:
```
from urllib.request import urlopen
url = "https://github.com/pyrogram/pyrogram/"
page = urlopen(url)
html_bytes = page.read()
s1 = html_bytes.decode("utf-8")
a1 = get_attr(s1,"src")
a2 = get_attr(s1,"img")
a3 = get_attr(s1,"href")
a = ["communication",a1,a2,a3]
```

[6]:
```
a
```

[6]:
```
['communication', 108, 25, 174]
```

0.2 python package – Artistic Software Category

[7]:
```
from urllib.request import urlopen
url = "https://github.com/lace/vg"
page = urlopen(url)
html_bytes = page.read()
s1 = html_bytes.decode("utf-8")
a1 = get_attr(s1,"src")
a2 = get_attr(s1,"img")
a3 = get_attr(s1,"href")
b = ["Artistic Soft",a1,a2,a3]
```

[8]:
```
b
```

[8]:
```
['Artistic Soft', 101, 20, 153]
```

0.3 python package – Adaptive Technologies

[9]:
```
from urllib.request import urlopen
url = "https://github.com/NapoII/logNow"
page = urlopen(url)
html_bytes = page.read()
s1 = html_bytes.decode("utf-8")
a1 = get_attr(s1,"src")
a2 = get_attr(s1,"img")
a3 = get_attr(s1,"href")
c = ["Adaptive Tech",a1,a2,a3]
```

[10]:
```
c
```

[10]:
```
['Adaptive Tech', 103, 26, 133]
```

[11]:
```
x = [a,b,c]
df = pd.DataFrame(x,columns=["Category","src","images","href"])
```

0.4 plotting

[12]:
```
df
```

[12]:

	Category	src	images	href
0	communication	108	25	174
1	Artistic Soft	101	20	153
2	Adaptive Tech	103	26	133

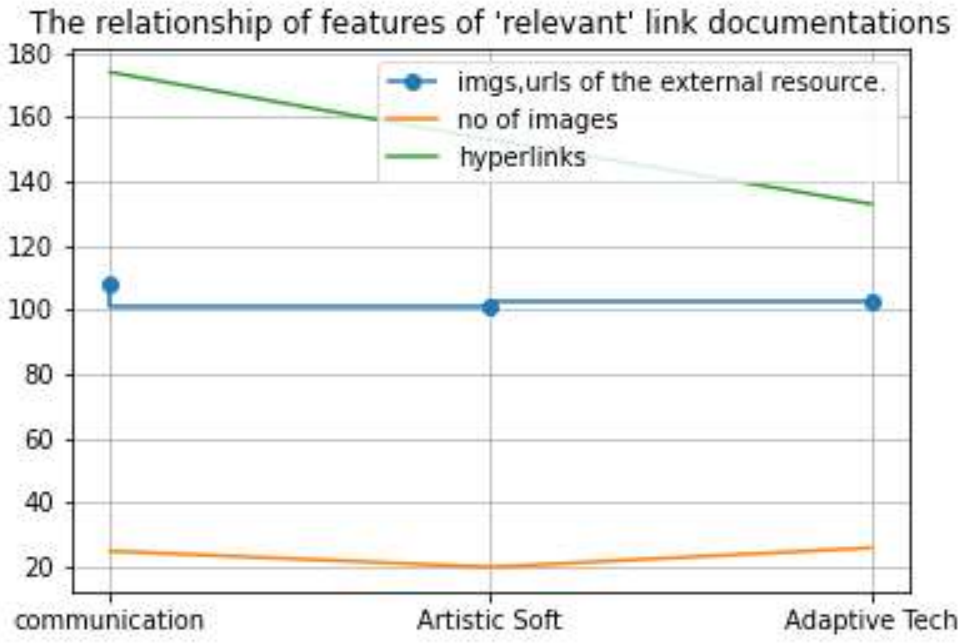
[49]:
```
fig, ax = plt.subplots()

# plotting all 3 variables
ax.plot(df["Category"],df["src"],marker='o',drawstyle = 'steps',solid_capstyle␣
 ↪= "butt", label="imgs,urls of the external resource.")
ax.plot(df["Category"],df["images"],label="no of images")
ax.plot(df["Category"],df["href"],label="hyperlinks")

# Title it
ax.set_title("The relationship of features of 'relevant' link documentations")

# Legend for identification
ax.legend()
ax.grid('on')
```

```
plt.savefig("plots/one.png")
```

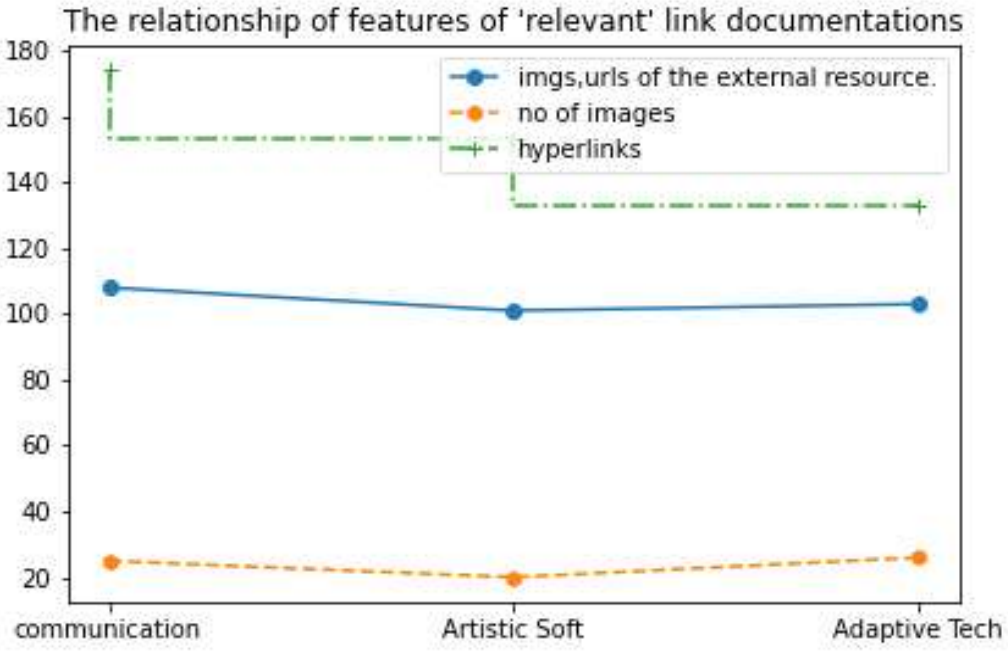
[47]:
```
fig, ax = plt.subplots()

# plotting all 3 variables
ax.plot(df["Category"],df["src"],marker='o',linestyle = 'solid',solid_capstyle␣
 ↪= "butt", label="imgs,urls of the external resource.")
ax.plot(df["Category"],df["images"],marker='h',linestyle =␣
 ↪'dashed',solid_capstyle = "butt",label="no of images")
ax.plot(df["Category"],df["href"],marker='+',linestyle = 'dashdot', drawstyle =␣
 ↪'steps',solid_capstyle = "butt",label="hyperlinks")

# Title it
ax.set_title("The relationship of features of 'relevant' link documentations")

# Legend for identification
ax.legend()
plt.tight_layout()
plt.savefig("plots/two.png")
```

[ ]:
```

```

Results of running 1st cell



Results of running 2nd cell


The relationship of features of 'relevant' link documentations
180
160
140
120
100
80
60
40
20
imgs,urls of the external resource.
no of images
hyperlinks
communication
Artistic Soft
Adaptive Tech