Finished - FT - date and title

August 18, 2017

1 Scraping data from Financial Times

- This data was scraped august 18, 2017, 12:00.
- It is the result of a search on 'Bitcoin'

1.1 Preparation

```
In [1]: # Imports data for our webscra
        import requests
        from bs4 import BeautifulSoup
```

1.2 Find article-titles

1.2.1 Find kkk (all-title-list)

```
In [21]: #This function makes a list with all pages containing search results.
         def next_page():
             url = 'http://www.ft.com/search?q=bitcoin&page='
             FTpages= []
             for i in range(0,47):
                 nextpage =  url + str(i)
                 FTpages.append(nextpage)
             return FTpages


         # Call funktion and subscribe list from function to variable
         FTpages = next_page()


         titles=[]
         # Choose one page-http: adress
         for pagenr in FTpages:
         # pick text from page-http:
             response = requests.get(pagenr)
         # Pick the text in another format
             soup = BeautifulSoup(response.text,"lxml" )
         # Finds substring of HTML including the title
             page = soup.find_all('a', attrs={'class':'js-teaser-heading-link'})
```

```
# makes a list with all links (and some other text) included
    for links in page:
        titles.append(str(links))
# extract only the links from the titles-list and name the all-title list - kkk.
kkk=[i.split('>\n\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t',1)[1].split('\n\t\t\t\t\t\t\t\t\
```

1.2.2 Remove unicode from kkk (all-title-list)

```
In [22]: # First unicode-characters removed
         kkk.index("You say you want a revolution\u2009.\u2009.\u2009.\u2009")
         kkk.remove("You say you want a revolution\u2009.\u2009.\u2009.\u2009")
         kkk.insert(787,"You say you want a revolution")

         # Second unicode-character removed
         kkk.index("Similar immunity of Bitcoin\u200a and\u200a gold")
         kkk.remove('Similar immunity of Bitcoin\u200a and\u200a gold')
         kkk.insert(900,"Similar immunity of Bitcoin and gold")
```

1.2.3 Save all-title-list as csv.file

```
In [ ]: with open("titel1.csv", "w") as out_file:
            for i in range(len(kkk)):
                out_string = ""
                out_string += str(kkk[i])
                out_string += '\n'
                out_file.write(out_string)
```

1.3 Find article-dates

1.3.1 Find jjj(all-dates-list)

```
In [24]: # Call funktion and subscribe list from function to variable
         FTpages = next_page()

         dates=[]
         for pagenr in FTpages:
         # pick up text from the first page of the Bitcoin search
             response = requests.get(pagenr)
         # Pick the text in another format
             soup = BeautifulSoup(response.text,"lxml" )
         # Finds substring of HTML including the date
             page = soup.find_all('div', attrs={'class':'stream-card__date'})
         # makes a list with all dates (and some other text) included
             for links in page:
                 dates.append(str(links))
         # extract only the date from the dates-list and name the all-date-list - jjj.
         jjj=[i.split('000Z">',1)[1].split('</time>',1)[0] for i in dates]
         o
```

```
['Friday, 18 August, 2017', 'Friday, 18 August, 2017', 'Thursday, 17 August, 2017', 'Thursday, 1
```

1.3.2 Save all-dates-list as csv.file

```
In [ ]: with open("dato1.csv", "w") as out_file:
            for i in range(len(jjj)):
                out_string = ""
                out_string += str(jjj[i])
                out_string += '\n'
                out_file.write(out_string)
```

1.4 Concluding remarks - Datascrape Financial Times

- In order to have fully matched title and dates, we have made a few manually changes to the csv-files and merged them into one common file.
- From this point we will just import the data into two lists and make a dataframe.