JODATUT2019 Datan jalostaminen

Nimi: Petri Salminen
Opnum: 243840
Email: petri.salminen@tuni.fi

Jäimmme viimeksi siihen, että saimme datan todistettavasti ladattua jupyteriin. Nyt voimme ladata pandasin ynnä muut kumppanit ja alkaa tutkimaan dataa ja jalostamaan sitä parempaan muotoon.

In [1]:
```
import pandas
```

Nyt kun pandas on ladattu, voimme lukea csv-tiedoston. `sep` tarkoittaa erotinmerkkiä tiedostossa

In [2]:
```
df = pandas.read_csv('data/weblog.csv', sep=',')
```

In [3]:
```
df.head()
```

Out[3]:

	IP	Time	URL	Staus
0	10.128.2.1	[29/Nov/2017:06:58:55	GET /login.php HTTP/1.1	200
1	10.128.2.1	[29/Nov/2017:06:59:02	POST /process.php HTTP/1.1	302
2	10.128.2.1	[29/Nov/2017:06:59:03	GET /home.php HTTP/1.1	200
3	10.131.2.1	[29/Nov/2017:06:59:04	GET /js/vendor/moment.min.js HTTP/1.1	200
4	10.130.2.1	[29/Nov/2017:06:59:06	GET /bootstrap-3.3.7/js/bootstrap.js HTTP/1.1	200

Näyttää hyvältä! Katsotaan, onko data puutteellista

In [4]:
```
df.isna().sum()
```

Out[4]:
```
IP       0
Time     0
URL      0
Staus    0
dtype: int64
```

Näyttää hyvältä! Katsotaan, mihin muotoon pandas on lukenut tiedot

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16007 entries, 0 to 16006
Data columns (total 4 columns):
IP       16007 non-null object
Time     16007 non-null object
URL      16007 non-null object
Staus    16007 non-null object
dtypes: object(4)
memory usage: 500.3+ KB
```

Näköjään kaikki on luettu objecteiksi (merkkijono). Muutetaan Staus-kolumnin nimi Statukseksi ja koitetaan muuntaa tyyppi numeroksi.

In [6]:
```
df.rename(columns={'Staus':'Status'}, inplace=True)

#df = df.astype({"IP":object, "Time":object, "URL":object, "Status":int}) # Tämä meni mönkään, koska siellä oli jotain likaista
df.groupby('Status').size()
```

Out[6]:
```
Status
200             11330
2017]               7
2018]              28
206                52
302              3498
304               658
404               251
Aborted             4
Assertion           4
No                167
Segmentation        1
dumped              5
found               2
dtype: int64
```

In [7]:
```
df = df[df["Status"].apply(lambda x: x.isnumeric())] # Heitä roskiin ylimääräiset
df = df.astype({"IP":object, "Time":object, "URL":object, "Status":int}) # Nyt tämä toimii :)
```

Tehdään sama IP-sarakkeelle. IPv4-osoitteen pystyy muuttamaan numeroksi. Ohjeet esimerkiksi ohjeessa "How to convert ip addresses to decimal format (https://itstillworks.com/convert-ip-addresses-decimal-format-7611714.html)".

In [8]:
```
def ip_to_int(ip):
    ip_a = ip.split('.')
    return int(ip_a[0])*256**3+int(ip_a[1])*256**2+int(ip_a[2])*256**1+int(ip_a[3])

df['IP'] = df['IP'].apply(lambda x: ip_to_int(x))
```

In [9]:
```
df.head()
```

Out[9]:

	IP	Time	URL	Status
0	176161281	[29/Nov/2017:06:58:55	GET /login.php HTTP/1.1	200
1	176161281	[29/Nov/2017:06:59:02	POST /process.php HTTP/1.1	302
2	176161281	[29/Nov/2017:06:59:03	GET /home.php HTTP/1.1	200
3	176357889	[29/Nov/2017:06:59:04	GET /js/vendor/moment.min.js HTTP/1.1	200
4	176292353	[29/Nov/2017:06:59:06	GET /bootstrap-3.3.7/js/bootstrap.js HTTP/1.1	200

Dodiin! Seuraavaksi käykäämme Time kimppuun. Kyseessä on Datetime-muuttuja. Kokeillaan muuttaa sitä suoraan.

In [10]:
```
df['Time'] = pandas.to_datetime(df['Time'], format="[%d/%b/%Y:%H:%M:%S")
```

In [11]:
```
df.head()
```

Out[11]:

	IP	Time	URL	Status
0	176161281	2017-11-29 06:58:55	GET /login.php HTTP/1.1	200
1	176161281	2017-11-29 06:59:02	POST /process.php HTTP/1.1	302
2	176161281	2017-11-29 06:59:03	GET /home.php HTTP/1.1	200
3	176357889	2017-11-29 06:59:04	GET /js/vendor/moment.min.js HTTP/1.1	200
4	176292353	2017-11-29 06:59:06	GET /bootstrap-3.3.7/js/bootstrap.js HTTP/1.1	200

Mahtavaa! Nyt vielä jos URL-kolumnin saisi parempaan kuntoon. Muodostetaan siitä useampi sarake.

In [12]:
```
df['RequestType'], df['URL'], df['HttpVersion'] = df['URL'].str.split(' ', 2).s
tr
df['HttpVersion'] = df['HttpVersion'].apply(lambda x: x.split("/")[1])
```

In [13]:
```
df.head()
```

Out[13]:

	IP	Time	URL	Status	RequestType	HttpVersion
0	176161281	2017-11-29 06:58:55	/login.php	200	GET	1.1
1	176161281	2017-11-29 06:59:02	/process.php	302	POST	1.1
2	176161281	2017-11-29 06:59:03	/home.php	200	GET	1.1
3	176357889	2017-11-29 06:59:04	/js/vendor/moment.min.js	200	GET	1.1
4	176292353	2017-11-29 06:59:06	/bootstrap-3.3.7/js/bootstrap.js	200	GET	1.1

Nyt data alkaa näyttämään siltä, että sitä voisi hyödyntää analyysissä! WOOHOO! Kirjoitetaan data uuteen tiedostoon, jotta tätä siivousta ei tarvitse missään nimessä tehdä uudestaan.

In [14]:
```
df.to_csv('data/weblog_clean.csv', sep=",")
```

Hyödylliset linkit

- Pandasin koko dokumentaatio (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html#pandas.read_csv). Esimerkkinä read_csv
- "How to convert ip addresses to decimal format (https://itstillworks.com/convert-ip-addresses-decimal-format-7611714.html)

Erityisen helpot ja hankalat asiat

ohjeessa

- Hankalin asia se, että koko ajan oli takaraivossa ajatus siitä, että teen turhaa työtä tätä näin siivoessani. Kumminkin loppujen lopuksi kaikki vaikuttaa hyvältä :)
- Jupyter + pandas sopi tähän erittäin hienosti!

In [ ]: