Analysis with pandas

April 14, 2017

0.1 Anaylsis with Pandas

0.1.1 By: Justin Sallese

```
In [61]: pwd
```

```
Out[61]: '/Users/justinsalt/pcap'
```

```
In [62]: ls -la
```

```
total 111544
drwxr-xr-x    5 justinsalt  staff       170 14 Apr 15:48 ./
drwxr-xr-x+  74 justinsalt  staff      2516 14 Apr 15:57 ../
-rw-r--r--@  1 justinsalt  staff      6148 14 Apr 15:42 .DS_Store
-rw-r--r--   1 justinsalt  staff    915550 14 Apr 15:55 frame.len
-rw-r--r--   1 justinsalt  staff  56180821 14 Apr 14:58 nitroba.pcap
```

```
In [63]: ls -l nitroba.pcap
```

```
-rw-r--r--  1 justinsalt  staff  56180821 14 Apr 14:58 nitroba.pcap
```

```
In [64]: !tshark -n -r nitroba.pcap -T fields -Eheader=y -e frame.number
         -e frame.len > frame.len
```

The command above converts the pcap into a file with the only desired information

```
In [65]: !head -10 frame.len
```

```
frame.number        frame.len
1         70
2         70
3         1421
4         70
5         1284
6         70
7         70
8         70
9         78
```

The above command is also like the head() command in the R languange

```
In [66]: import pandas as pd
         df=pd.read_table("frame.len")
         df
```

```
Out[66]:        frame.number  frame.len
         0                 1         70
         1                 2         70
         2                 3       1421
         3                 4         70
         4                 5       1284
         5                 6         70
         6                 7         70
         7                 8         70
         8                 9         78
         9                10         78
         10               11        386
         11               12         78
         12               13         80
         13               14         80
         14               15         82
         15               16         78
         16               17         70
         17               18         70
         18               19         70
         19               20        172
         20               21         70
         21               22       1466
         22               23        392
         23               24         70
         24               25        209
         25               26         76
         26               27        111
         27               28        117
         28               29         70
         29               30         70
         ...             ...        ...
         94380         94381        397
         94381         94382        395
         94382         94383        399
         94383         94384        391
         94384         94385         70
         94385         94386         70
         94386         94387         70
         94387         94388         70
         94388         94389         64
         94389         94390         70
```

```
94390      94391      70
94391      94392      70
94392      94393      70
94393      94394      70
94394      94395     118
94395      94396      97
94396      94397      70
94397      94398      64
94398      94399     118
94399      94400     347
94400      94401     403
94401      94402     331
94402      94403     323
94403      94404     367
94404      94405     343
94405      94406     397
94406      94407     395
94407      94408     399
94408      94409     391
94409      94410      70

[94410 rows x 2 columns]
```

```
In [67]: df["frame.len"].describe()
```

```
Out[67]: count    94410.000000
         mean       579.072524
         std        625.671800
         min         42.000000
         25%         70.000000
         50%         86.000000
         75%       1466.000000
         max       1466.000000
         Name: frame.len, dtype: float64
```

I have found that the above command is also like the summary() command is the R language.

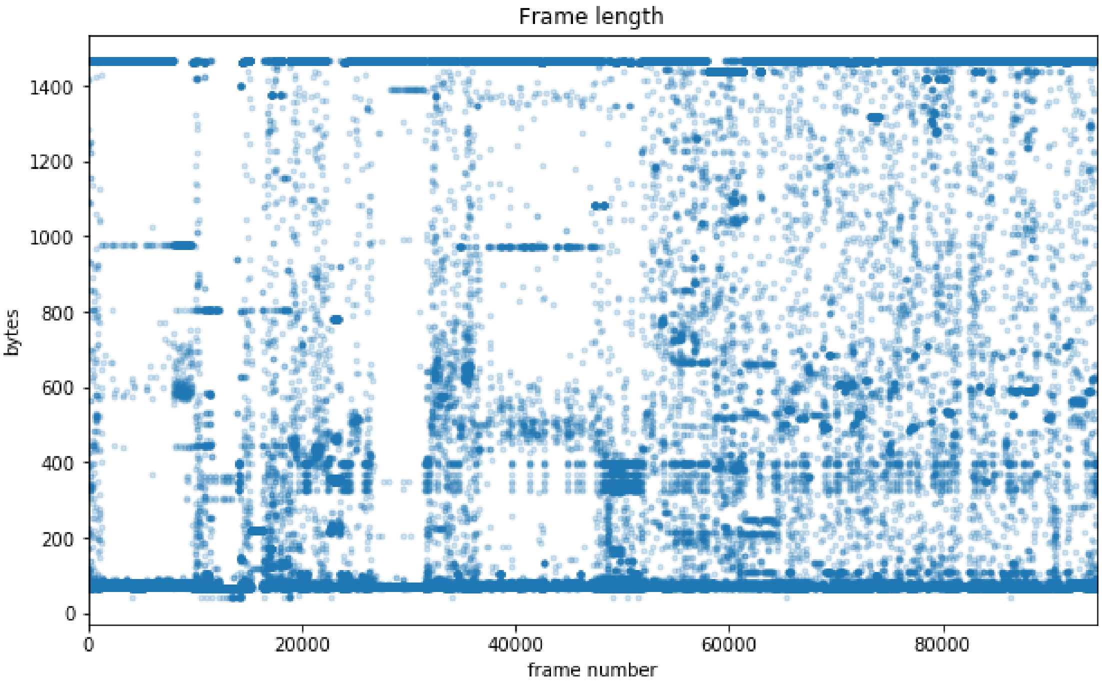
```
In [68]: %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [69]: figsize(10,6)
```

```
In [70]: df["frame.len"].plot(style=".", alpha=0.2)
         title("Frame length")
         ylabel("bytes")
         xlabel("frame number")
```

```
Out[70]: <matplotlib.text.Text at 0x1187cd358>
```



The plot above shows the distribution of the frame length of the packets within the provided data. As can be seen above the legth of packets are clusetered above 1400 and below 200 and some in the range of 400. The reason why there is not nuch analysis with pandas is because the main analysis is to be done with R.