

# Analiza Kretanja Vozila

September 3, 2019

## 1 Analiza podataka kretanja vozila iz .csv filea

Demo kako uz pomoc pythona i *matplotlib* biblioteke za prikaz grafova

Prvo uvezemo potrebne biblioteke

```
[100]: import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
from numpy import genfromtxt, arange, sin, pi
from matplotlib import style
from matplotlib import dates as mpl_dates
import numpy as np
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
```

Unese se ime datoteke s podacima i mapiraju se polja sukladno zapisanome.

U ovom primjeru podaci su razdvojeni s znakom ',' ali cesti je slucaj kada su podaci odvojeni nekim drugim znakom te se to treba posebno naznaciti kako bi program znao granice izmedu polja.

```
[101]: filename='GPSLOG10.CSV'
#plt.style.use('ggplot')
data=pd.read_csv(filename, header=None, delimiter=',',
→names=['Sentence', 'Time', 'Validity', 'Latitude', 'NS', 'Longitude', 'EW', 'Speed',
→'Direction', 'Date', 'NA1', 'NA2', 'Checksum', 'Temperature'])
```

### 1.1 Line plot

Sada smo spremni za prikazati prikupljene podatke. Prvo mozemo prikazati jednostavan s/t graf - brzinu u vremenu. Kako je brzina zapisana u cvorovima, a mi je zelimo prikazati u km/h potrebno izvršiti konverziju. 1 nauticna milja odgovara 1.852 km.

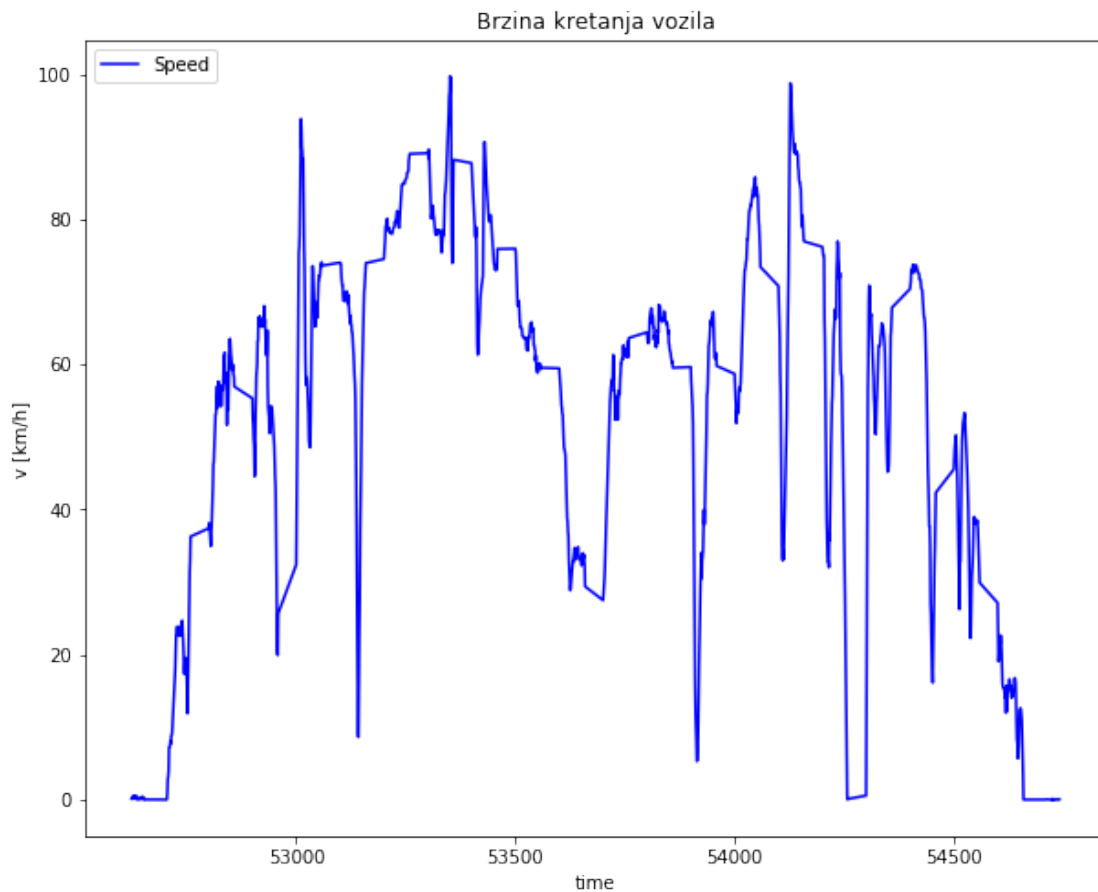
Svaki graf treba imati oznacene osi. S komandom plt.xlabel i ylabel oznacili smo osi grafa i analogno tome imenovan je i graf kako bi citatelj znao to graf predstavlja. Naravno, pojedinačni grafovi se mogu posebno spremati u visokoj rezoluciji i zeljenom formatu za kasniju upotrebu.

```
[102]: #otvori graf u novom prozoru
#%matplotlib qt
#plt.plot(data['Time'], data['Speed']*1.852, 'b-')
```

```
plt.plot(data['Time'],data['Speed']*1.852, 'b-')

plt.legend(loc='upper left')
plt.xlabel ('time')
plt.ylabel ('v [km/h]')
plt.title('Brzina kretanja vozila')
#plt.savefig('GrafKretanjaBrzineVozila.png',format='png', bbox_inches='tight',
→dpi=100)
```

[102]: Text(0.5, 1.0, 'Brzina kretanja vozila')



Dodatno se mogu izracunati i pogledati razni podaci koje nas zanimaju.

Ako npr. zelimo znati koja je bila maksimalna brzina kojom se vozilo kretalo to se moze vidjeti na sljedeci nacin:

[103]: `print('Maksimalna brzina = ',np.max(data['Speed']*1.852) , 'km/h')`

Maksimalna brzina = 99.73020000000001 km/h

Ako nas zanimaju podaci o temperaturi moguće je čak koristiti i ugrađene statističke funkcije za izracunati zeljene podatke

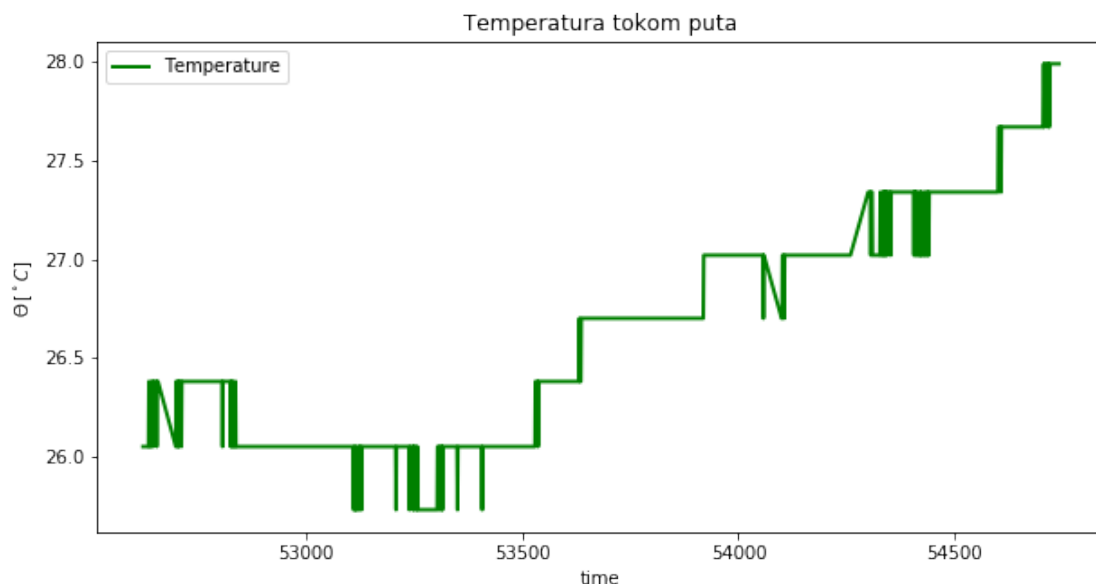
```
[104]: # prikazi graf inline
%matplotlib inline
plt.rcParams["figure.figsize"] = (10,5)
plt.plot(data['Time'],data['Temperature'], 'g', linewidth=2)

plt.legend(loc='upper left')
plt.xlabel ('time')
plt.ylabel ('$\\Theta \\, [^\\circ C]$')
plt.title('Temperatura tokom puta')

print ('Minimalna temperatura: ',np.min(data['Temperature']), 'řC')
print ('Maximalna temperatura: ',np.max(data['Temperature']), 'řC')
print ('Razlika temperature: ',np.max(data['Temperature']) - np.
    ↳min(data['Temperature']), 'řC')
print ('Razlika temperature: ', '{:.2f}'.format(np.ptp(data['Temperature'])), 'řC')
print ('Prosječna temperatura: ', '{:.2f}'.format(np.
    ↳mean(data['Temperature'])), 'řC ± ',
    '{:.2f}'.format(np.std(data['Temperature'])), 'řC')
```

```
Minimalna temperatura: 25.73 řC
Maximalna temperatura: 27.99 řC
Razlika temperature: 2.259999999999998 řC
Razlika temperature: 2.26 řC
Prosječna temperatura: 26.64 řC ± 0.57 řC
```

```
C:\Users\Kristijan\AppData\Roaming\Python\Python37\site-
packages\numpy\core\fromnumeric.py:2389: FutureWarning: Method .ptp is
deprecated and will be removed in a future version. Use numpy.ptp instead.
    return ptp(axis=axis, out=out, **kwargs)
```



Ako elimo plotati ove dvije veliine na istom grafu da vizualno utvrdimo postojanje korelacije moemo kreirati subplot

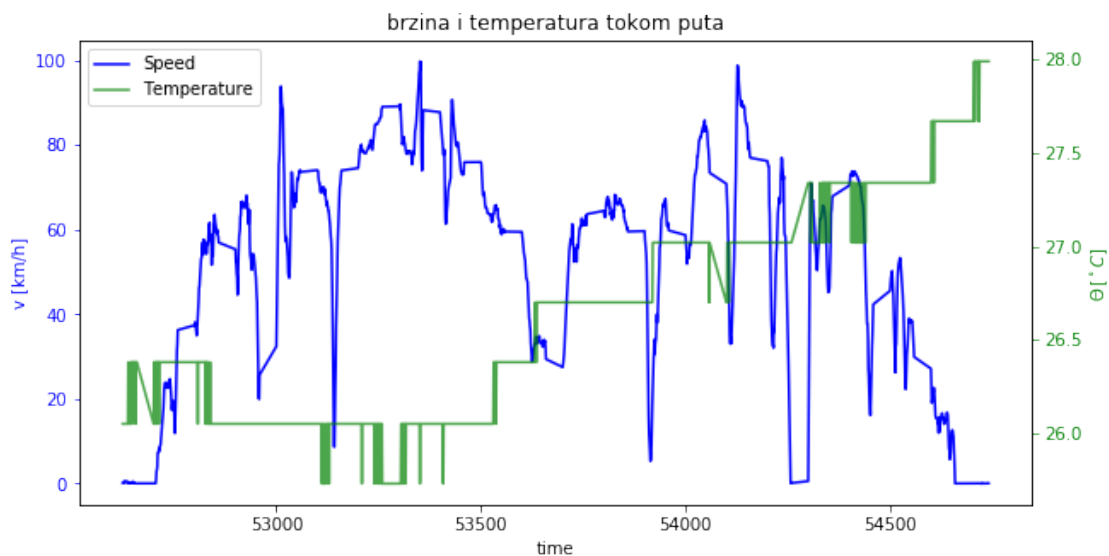
```
[105]: fig, ax1 = plt.subplots()
plt.title('brzina i temperatura tokom puta')

l1,=ax1.plot(data['Time'],data['Speed']*1.852, 'b-', label='Speed')
ax1.set_xlabel('time')
ax1.set_ylabel('v [km/h]', color='b')
ax1.tick_params('y', colors='b')

ax2 = ax1.twinx()
l2,=ax2.plot(data['Time'],data['Temperature'], 'g', alpha=0.7,
→label='Temperature')
ax2.set_ylabel('$\Theta$, [^\circ C]$', color='g')
ax2.tick_params('y', colors='g')

plt.legend([l1, l2],["Speed", "Temperature"])
```

[105]: <matplotlib.legend.Legend at 0x2ef94623828>



Uvidom u graf ne mozemo vizualno utvrditi postojanje zavisnosti jedve velicine o drugoj, ali se moze primjetiti trend porasta temperature s vremenom. Koji je tocan uzrok tome treba dodatno istraziti sto nije predmet ovog rada.

## 1.2 Scatter plot

Koristeci funkciju *scatter* koji za razliku od *line* ne spaja susjedne toke linijom moemo prikazati kretanje vozila u koordinatnom sustavu.

Dodavanjem vrijednosti parametru *c* (color) moemo prikazati i dodatnu dimenziju, a to je u ovom sluaju brzina kretanja prikazana putem obojene skale.

```
[107]: fig, ax = plt.subplots(1)

p=ax.scatter(data['Longitude'],data['Latitude'],c=data['Speed']*1.
→852,marker='o',cmap='jet')
plt.rcParams["figure.figsize"] = (10,8)
fig.colorbar(p)

plt.xlabel ('Longitude')
plt.ylabel ('Latitude')
plt.title('Pozicija i brzina objekta')
```

```
[107]: Text(0.5, 1.0, 'Pozicija i brzina objekta')
```

