

ODREĐIVANJE I FILTRACIJA POŽELJNIH PODATAKA

Analizirat ću podatke: zone_temperature, local_switch i zone_fan_speed

```
In [13]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
path = '/Users/mateotoic/Desktop/ProjektR/Godina/2022/zones_20_year
zone = pd.read_csv(path)
```

```
In [14]: zcop = zone.copy()
zcop['timestamp'] = pd.to_datetime(zcop['timestamp'])

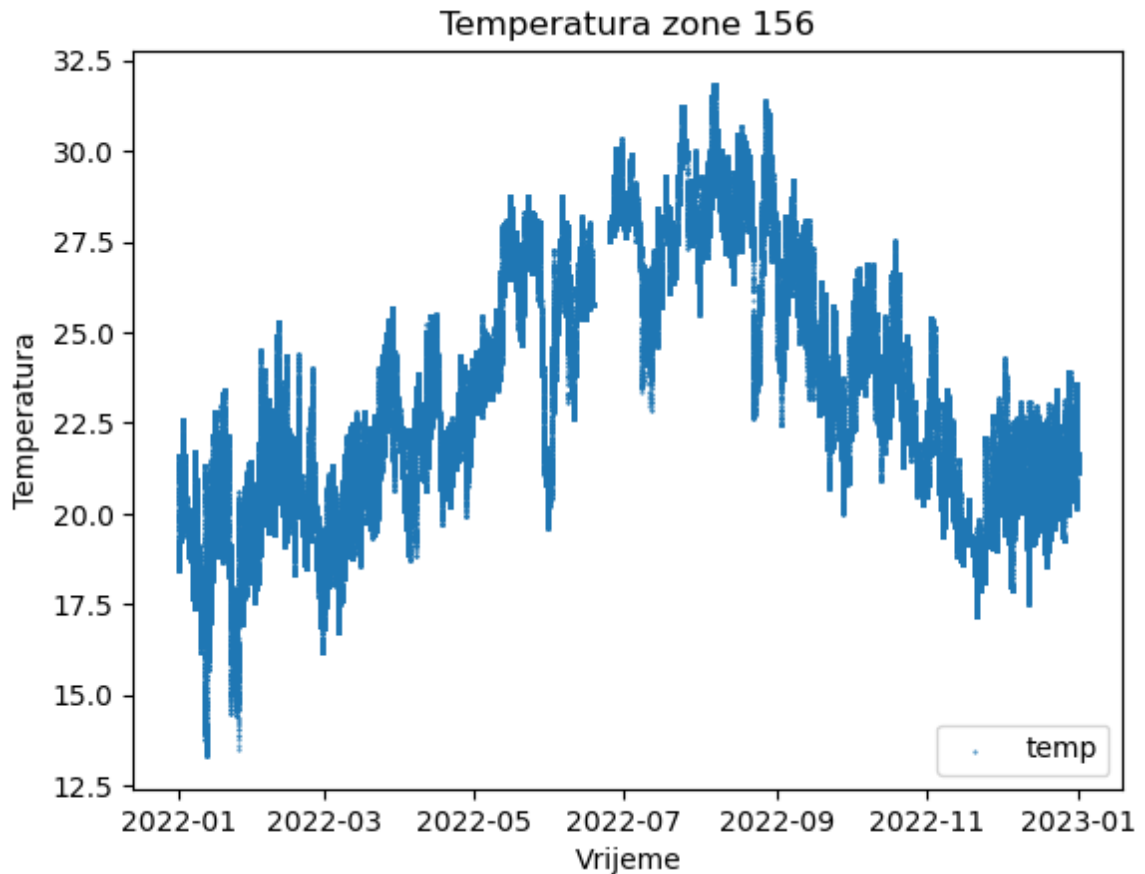
zone_sort = zcop.sort_values(by='timestamp')
```

Izdvojiti ću jednu zonu koja će mi služiti kao primjer ali isto vrijedi i za sve ostale.

```
In [25]: zona156 = zone_sort[zone_sort["zone_id"] == 156]
```

Temperatura

```
In [27]: plt.scatter(zona156.timestamp, zona156.zone_temperature, s=0.2, label="Vrijeme")
plt.xlabel("Vrijeme")
plt.ylabel("Temperatura")
plt.title("Temperatura zone 156")
plt.legend(loc="lower right")
plt.show()
```



Da bih odredio ispravnost podataka temperature koristit ću 3-sigma pravilo

```
In [42]: mean_temp = zona156['zone_temperature'].mean()
std_temp = zona156['zone_temperature'].std()

donja_granica = mean_temp - 3 * std_temp
gornja_granica = mean_temp + 3 * std_temp

prihvatljivi = zona156[(zona156['zone_temperature'] >= donja_granica
                        (zona156['zone_temperature'] <= gornja_granica))

odbaceni = zona156[(zona156['zone_temperature'] < donja_granica) |
                   (zona156['zone_temperature'] > gornja_granica)]

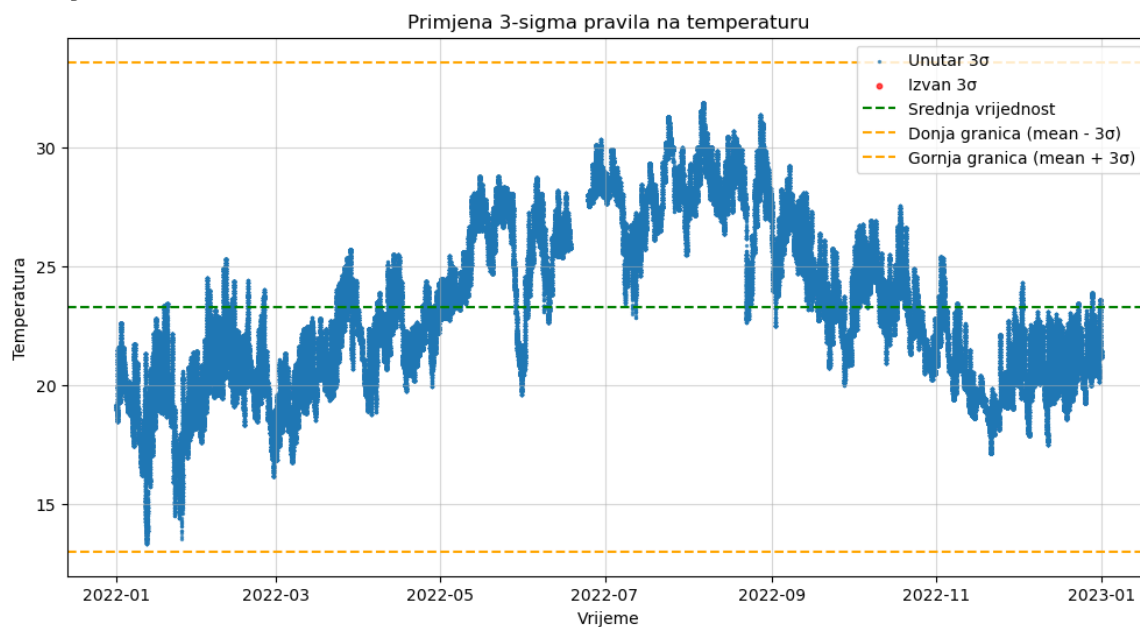
print(f"Srednja vrijednost: {mean_temp}")
print(f"Standardna devijacija: {std_temp}")
print(f"Interval (mean ± 3σ): [{donja_granica}, {gornja_granica}]")
print(f"Broj podataka unutar intervala: {len(zona156)}")
print(f"Broj outliera: {len(odbaceni)}")

plt.figure(figsize=(12,6))
plt.scatter(zona156['timestamp'], zona156['zone_temperature'], label='temp')
plt.scatter(odbaceni['timestamp'], odbaceni['zone_temperature'], label='outliers')

plt.axhline(mean_temp, color='green', linestyle='--', label="Srednja vrijednost")
plt.axhline(donja_granica, color='orange', linestyle='--', label="Donja granica")
plt.axhline(gornja_granica, color='orange', linestyle='--', label="Gornja granica")
plt.title("Primjena 3-sigma pravila na temperaturu")
plt.xlabel("Vrijeme")
plt.ylabel("Temperatura")
```

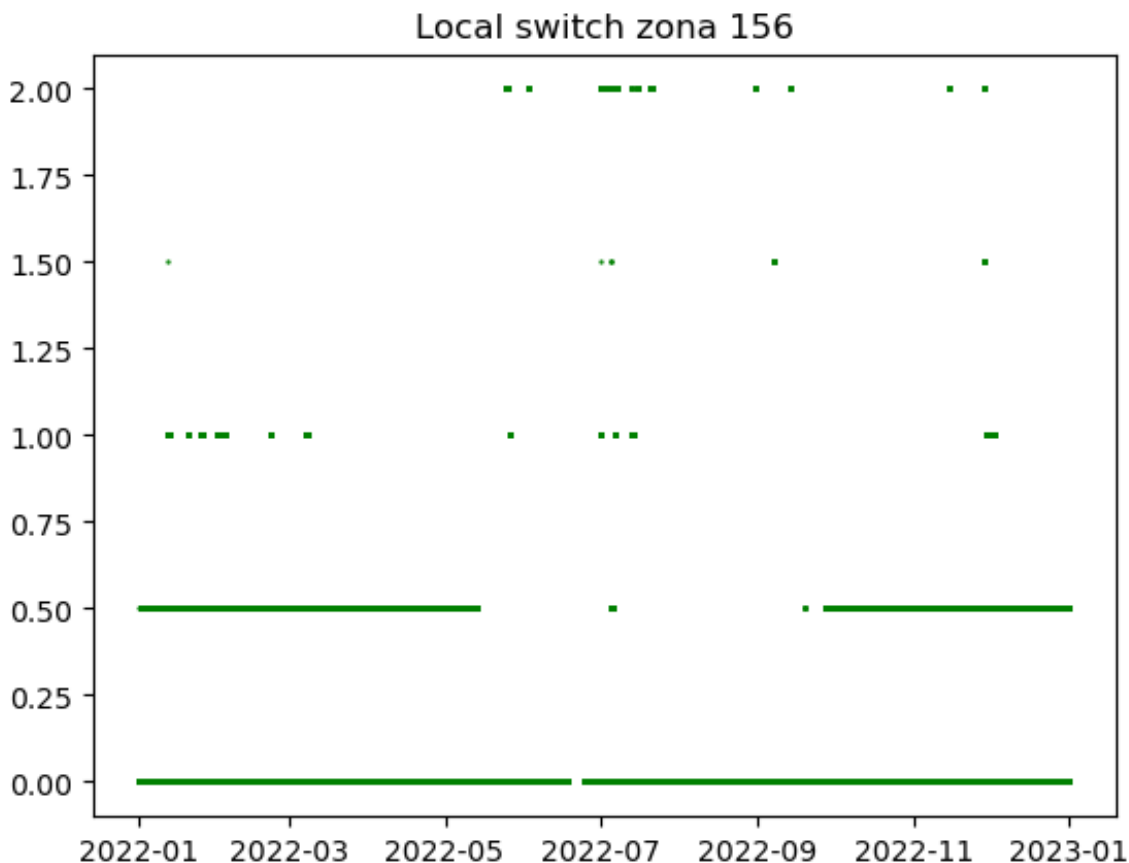
```
plt.legend()  
plt.grid(alpha=0.5)  
  
plt.show()
```

Srednja vrijednost: 23.284099032328886
Standardna devijacija: 3.4283890730685767
Interval (mean \pm 3 σ): [12.998931813123155, 33.569266251534614]
Broj podataka unutar intervala: 514741
Broj outliera: 0



Local switch

```
In [47]: plt.scatter(zona156.timestamp, zona156.local_switch, s=0.4, color=''  
plt.title("Local switch zona 156")  
plt.show()
```



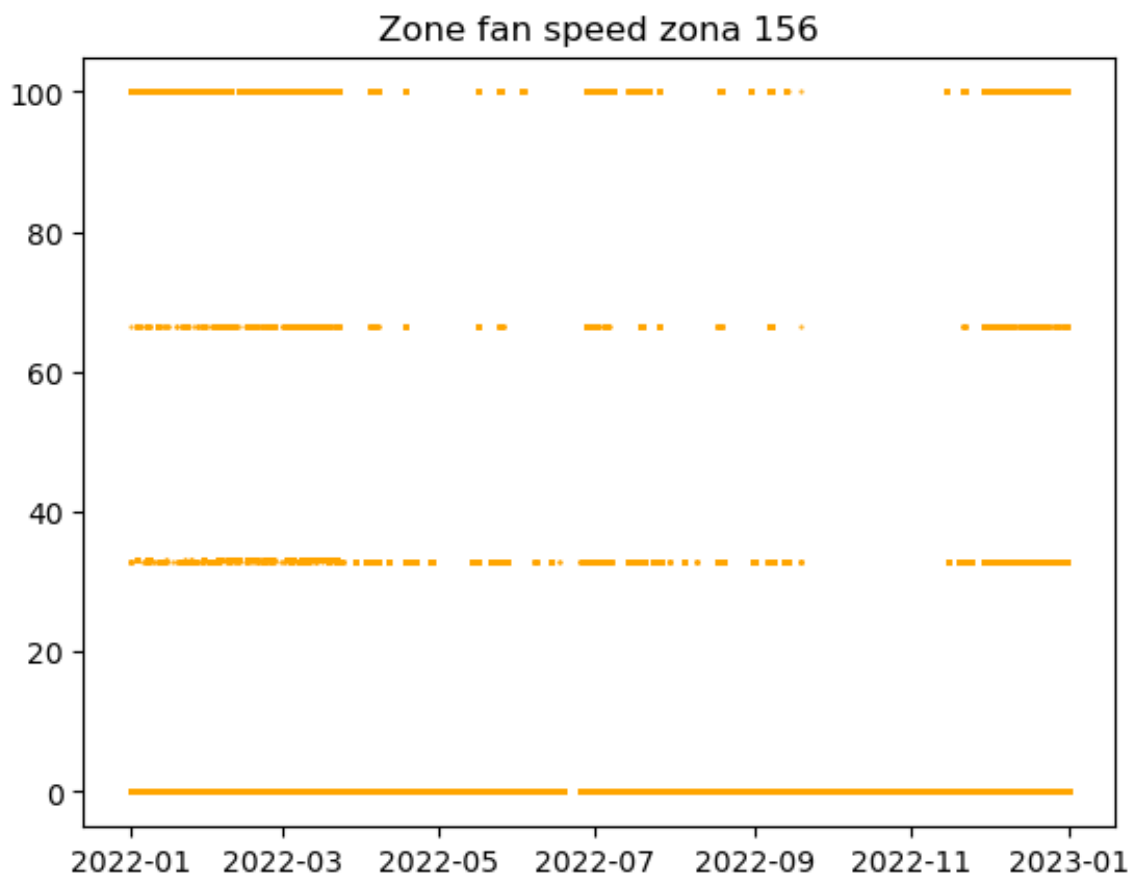
Da bih odredio ispravnost podataka local_switch provjerit ću pripadaju li predefiniranom skupu vrijednosti koje mogu poprimiti (0, 0.5, 1, 1.5, 2)

```
In [94]: dopusteni = [0, 0.5, 1, 1.5, 2]
dobri = zona156[zona156["local_switch"].isin(dopusteni)]
losi = zona156[~zona156["local_switch"].isin(dopusteni)]
print(f"Od {len(zona156)} podataka dobrih ima {len(dobri)}, a losih
```

Od 514741 podataka dobrih ima 514741, a losih 0

Zone_fan_speed

```
In [77]: plt.scatter(zona156.timestamp, zona156.zone_fan_speed, s=0.3, color
plt.title("Zone fan speed zona 156")
plt.show()
```



Svi podaci bi trebali pripadati skupu (0, 33, 33.3, 66.5, 100) pa ću to provjeriti na isti način kao i za local_switch

```
In [92]: dopusteni = [0, 33, 33.3, 66.5, 100]
dobri = zona156[zona156["zone_fan_speed"].isin(dopusteni)]
losi = zona156[~zona156["zone_fan_speed"].isin(dopusteni)]
print(f"Od {len(zona156)} podataka dobrih ima {len(dobri)}, a losih
```

Od 514741 podataka dobrih ima 514741, a losih 0