removing_outliers

January 11, 2021

[3]:
```
from helpers import pandas_helper as pdh
import matplotlib.pyplot as plt

corr = 'BMR008'

df20 = pdh.read_csv_activpal20_with_activiteiten(corr)
walking_df = df20.loc[df20['pal_activity_name'] == 'lopen']
print(walking_df.describe().T)
```

```
                count        mean        std   min    25%    50%    75%    max
pal_accX       6000.0   58.576333  33.554421   1.0   36.0   56.0   76.0  192.0
pal_accY       6000.0  138.978500  23.743020  56.0  123.0  138.0  153.0  235.0
pal_accZ       6000.0  130.536333  29.986912  22.0  116.0  128.0  141.0  253.0
pal_activity   6000.0    2.000000   0.000000   2.0    2.0    2.0    2.0    2.0
```

[25]:
```
w = walking_df.pal_accX
delta_x = w[1:].to_numpy() - w[:-1].to_numpy()
```

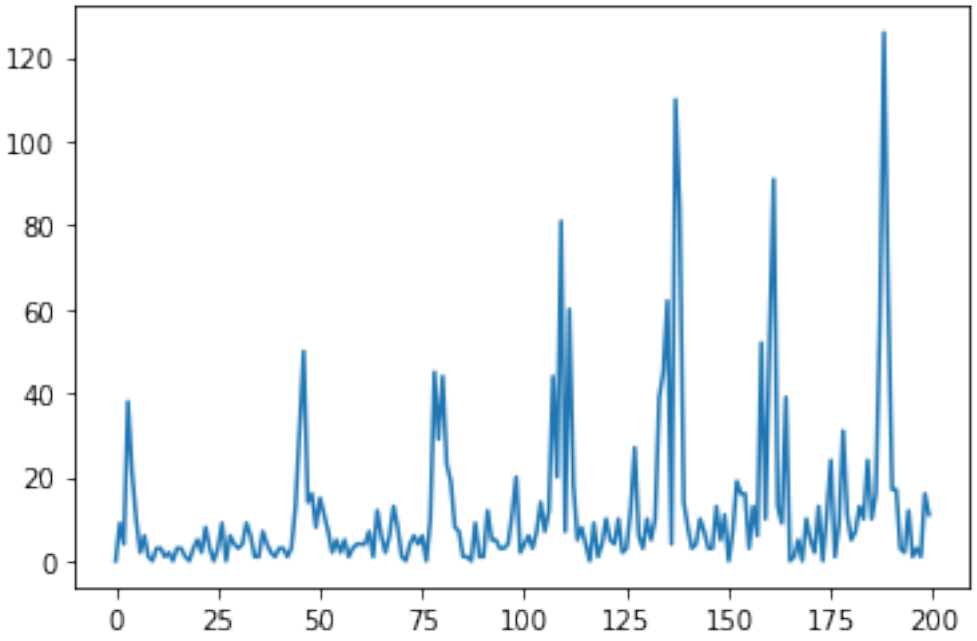
[29]:
```
import numpy as np
```

[32]:
```
sum(np.abs(delta_x))/len(delta_x)
```

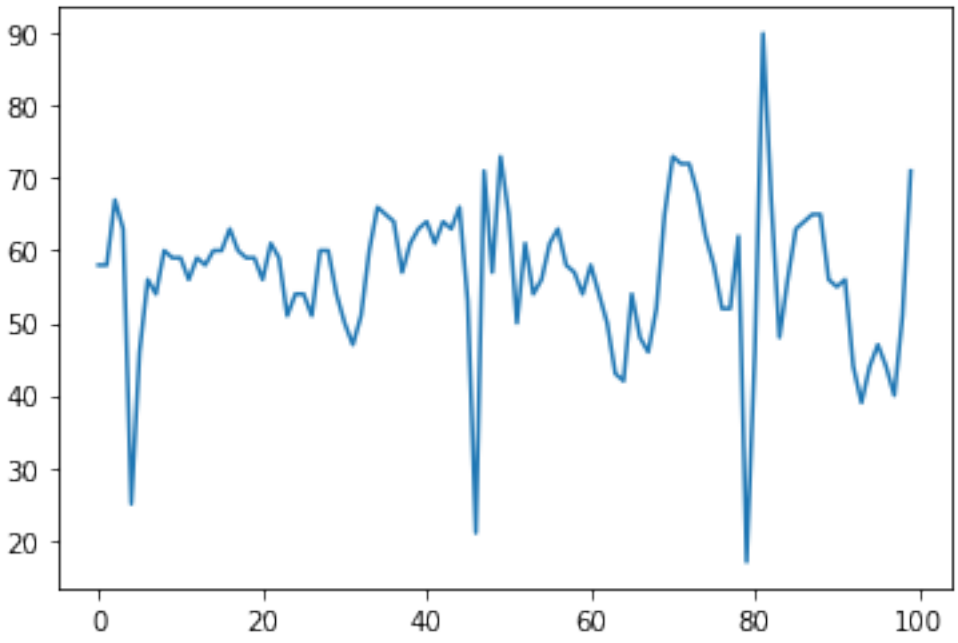
[32]:
```
30.12485414235706
```

[30]:
```
plt.plot(np.abs(delta_x[:200]))
```

[30]:
```
[<matplotlib.lines.Line2D at 0x7f519c3e3470>]
```

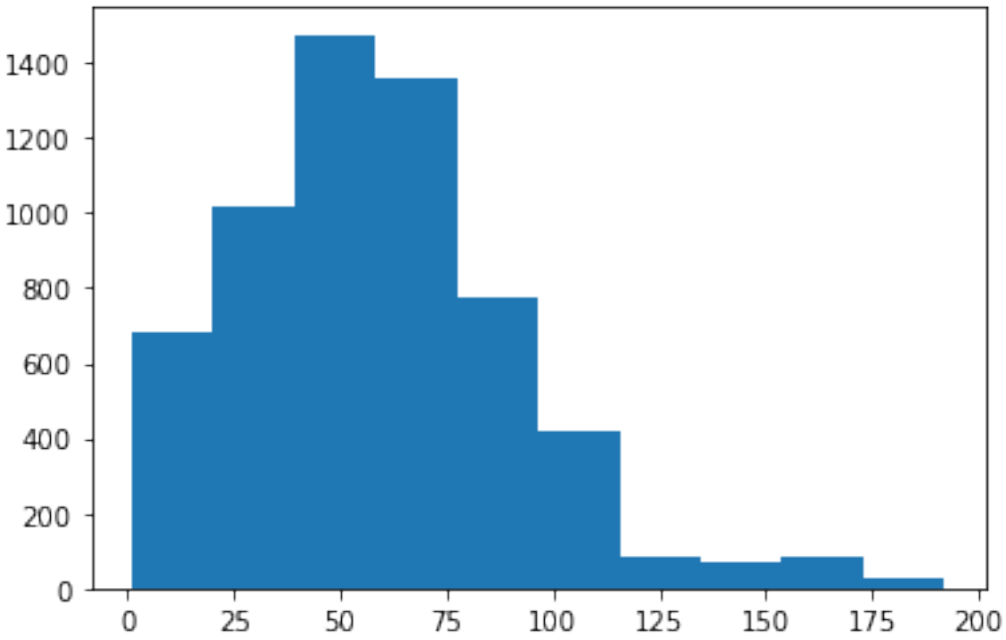


[11]:
```
plt.plot(range(100), list(walking_df['pal_accX'][:100]));
```



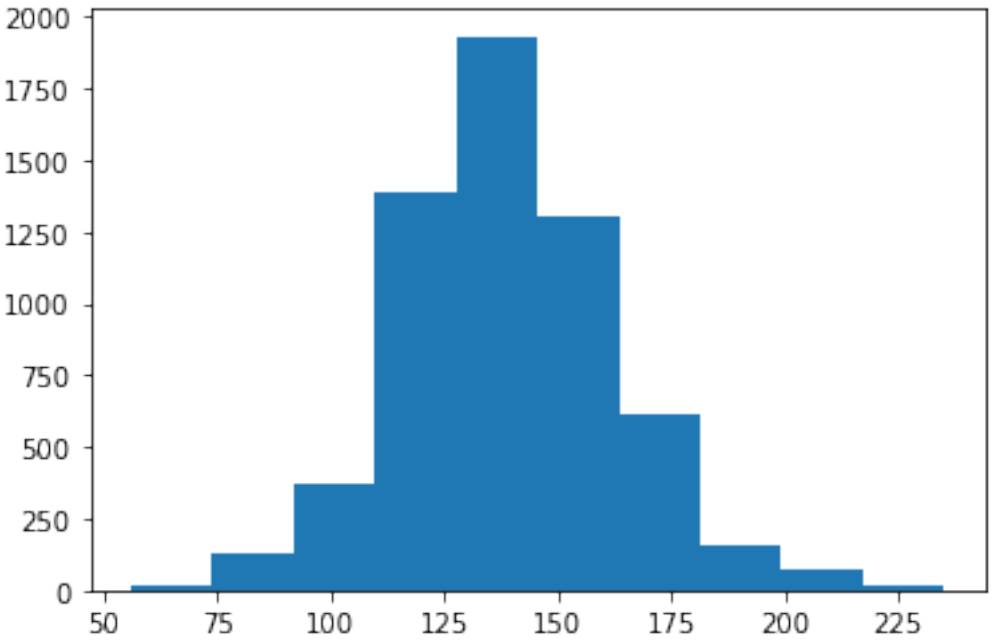
[198]:
```
plt.hist(walking_df['pal_accX'])
```

[198]:
```
(array([ 682., 1018., 1475., 1362.,  778.,  420.,   81.,   71.,   84.,
          29.]),
 array([  1. ,  20.1,  39.2,  58.3,  77.4,  96.5, 115.6, 134.7, 153.8,
        172.9, 192. ]),
 <a list of 10 Patch objects>)
```



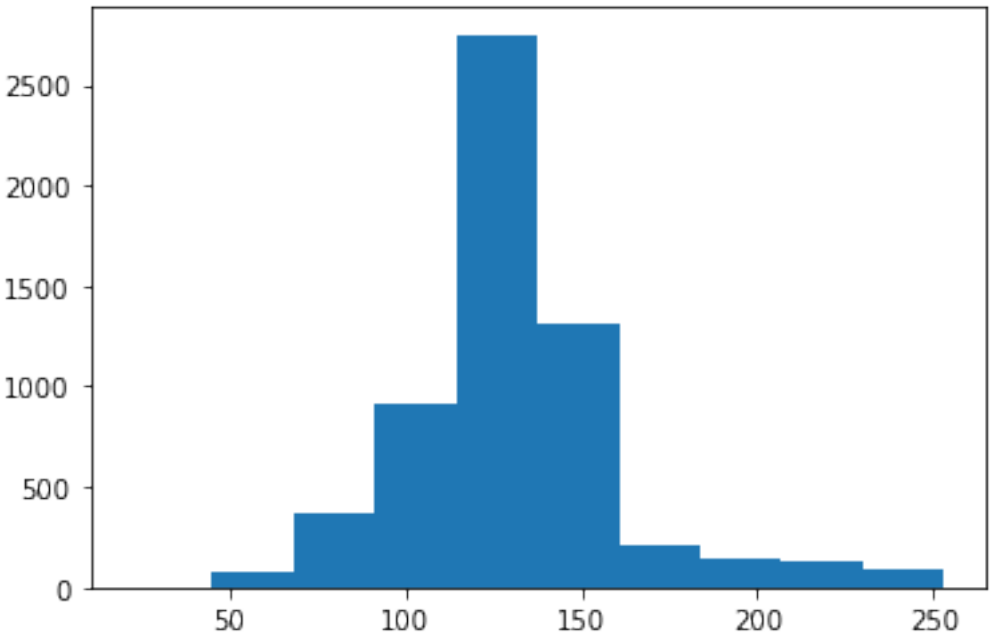
[199]:
```
plt.hist(walking_df['pal_accY'])
```

[199]:
```
(array([  17.,  125.,  367., 1392., 1932., 1303.,  611.,  161.,   76.,
          16.]),
 array([ 56. ,  73.9,  91.8, 109.7, 127.6, 145.5, 163.4, 181.3, 199.2,
        217.1, 235. ]),
 <a list of 10 Patch objects>)
```



[200]:
```
plt.hist(walking_df['pal_accZ'])
```

[200]:
```
(array([   3.,   81.,  366.,  911., 2750., 1317.,  211.,  139.,  134.,
         88.]),
 array([ 22. ,  45.1,  68.2,  91.3, 114.4, 137.5, 160.6, 183.7, 206.8,
        229.9, 253. ]),
 <a list of 10 Patch objects>)
```

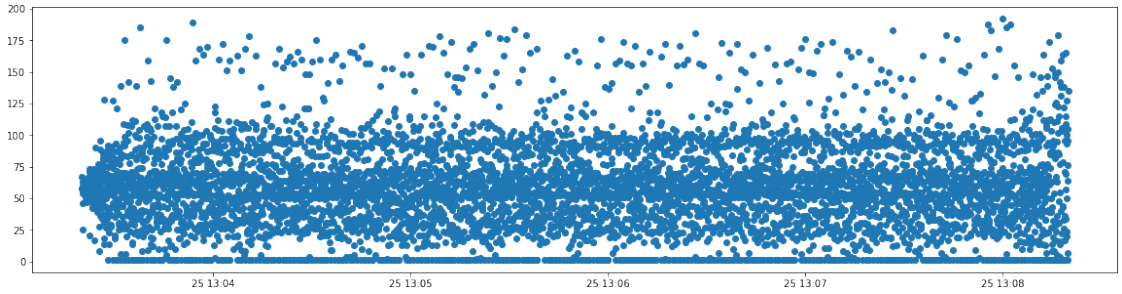


[15]:
```
columns = ['pal_accX','pal_accY','pal_accZ']


def remove_outliers(df, column, low_boundary = 0.01, high_boundary = 0.99,␣
 ↪inclusive = False):
    q_low = df[column].quantile(low_boundary)
    q_hi  = df[column].quantile(high_boundary)

    if not inclusive:
        return df[(df[column] < q_hi) & (df[column] > q_low)]
    else:
        return df[(df[column] <= q_hi) & (df[column] >= q_low)]


new_df = remove_outliers(walking_df, columns[0])

print (new_df.describe().T)
```

	count	mean	std	min	25%	50%	75%	max
pal_accX	5545.0	61.431199	28.806339	2.0	41.0	58.0	78.0	164.0
pal_accY	5545.0	139.198016	23.486167	63.0	124.0	138.0	153.0	235.0
pal_accZ	5545.0	128.884220	25.304522	22.0	116.0	128.0	140.0	253.0
pal_activity	5545.0	2.000000	0.000000	2.0	2.0	2.0	2.0	2.0

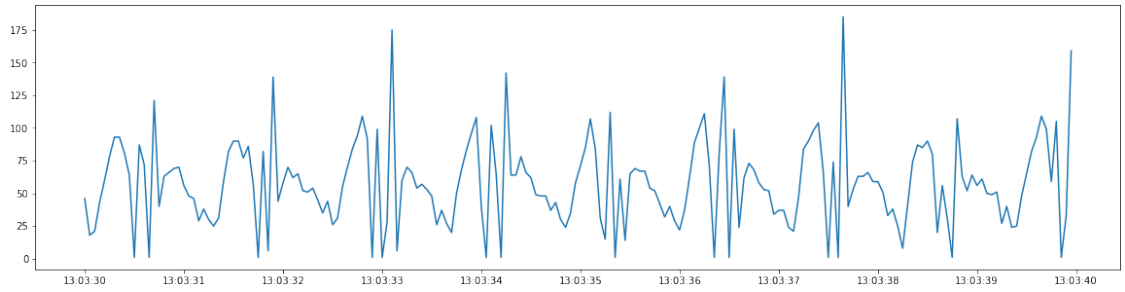
[16]:
```
plt.figure(figsize=(20,5))
plt.scatter(walking_df.index, walking_df['pal_accX'])
```

[16]: <matplotlib.collections.PathCollection at 0x7f51a2208710>



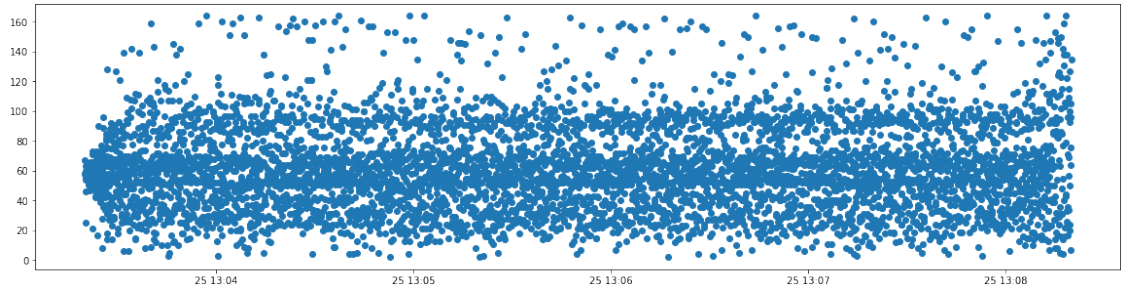
[21]:
```
plt.figure(figsize=(20,5))
plt.plot(walking_df['pal_accX'][200:400])
```

[21]: [<matplotlib.lines.Line2D at 0x7f51a208ae10>]



[17]:
```
plt.figure(figsize=(20,5))
plt.scatter(new_df.index, new_df['pal_accX'])
```

[17]: <matplotlib.collections.PathCollection at 0x7f51a21ea3c8>



[189]:
```
print(new_df)
```

```
                           pal_accX  pal_accY  pal_accZ  pal_activity  \
pal_time
2019-09-25 13:03:20.000001        58       146       157             2
2019-09-25 13:03:20.050002        58       146       157             2
2019-09-25 13:03:20.100001        67       130       137             2
2019-09-25 13:03:20.150001        63       133       131             2
2019-09-25 13:03:20.200000        25       130       144             2
...                              ...       ...       ...           ...
2019-09-25 13:08:19.700001        96        99       123             2
2019-09-25 13:08:19.800001         7       114       132             2
2019-09-25 13:08:19.850000        64       147       147             2
2019-09-25 13:08:19.900000        76       150       124             2
2019-09-25 13:08:19.950000       135       116       128             2

                           pal_activity_name
pal_time
2019-09-25 13:03:20.000001             lopen
2019-09-25 13:03:20.050002             lopen
```

```
2019-09-25 13:03:20.100001            lopen
2019-09-25 13:03:20.150001            lopen
2019-09-25 13:03:20.200000            lopen
…                                     …
2019-09-25 13:08:19.700001            lopen
2019-09-25 13:08:19.800001            lopen
2019-09-25 13:08:19.850000            lopen
2019-09-25 13:08:19.900000            lopen
2019-09-25 13:08:19.950000            lopen

[5604 rows x 5 columns]
```

[ ]: