lab_39

June 6, 2020

1 Badanie właściwości ferroelektryków:

2 Opracacowanie wyników

[18]:
```
import pandas as pd
import numpy as np
import gc
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="darkgrid")
def convert_to_number(series,start=0):
    return series.loc[start:].replace(to_replace=r'[,]',value='.', regex=True).
 ↪astype(float)
```

2.1 Parsowanie danych i przygotowanie do dalszej obróbki

[20]:
```
df_const = pd.read_csv('../../data/39/data_39.csv')
df_const['t [°C]'] = convert_to_number(df_const['t [°C]'])
df_const['Xc [dz]'] = convert_to_number(df_const['Xc [dz]'])
df_const['Ys [dz]'] = convert_to_number(df_const['Ys [dz]'])
df_const['S [m^2]'] = convert_to_number(df_const['S [m^2]'])
df_const['d [m]'] = convert_to_number(df_const['d [m]'])
series = [df_const['t [°C]'], df_const['Xc [dz]'], df_const['Ys [dz]'] ]
df_var = pd.DataFrame().join(series, how='outer')
for s in series:
    df_const = df_const.drop(s.name,axis=1)
df_const = df_const.dropna()
df_const
```

[20]:

	R1 [Ω]	R2 [Ω]	S [m^2]	d [m]	C_0 [mF]	1 [V]	2 [mV]
0	3800000.0	26000.0	1.160000e-09	0.0022	1.0	5.0	20.0

[21]:
```
df_var['T [K]'] = df_var['t [°C]']+ 274.15
df_var.head()
```

[21]:

	t [°C]	Xc [dz]	Ys [dz]	T [K]
0	28.1	4.0	4.0	302.25
1	30.1	4.0	4.0	304.25
2	32.1	4.0	4.0	306.25
3	36.0	3.0	3.0	310.15
4	38.2	3.0	3.0	312.35

2.2 Wyliczanie Ec oraz Ps

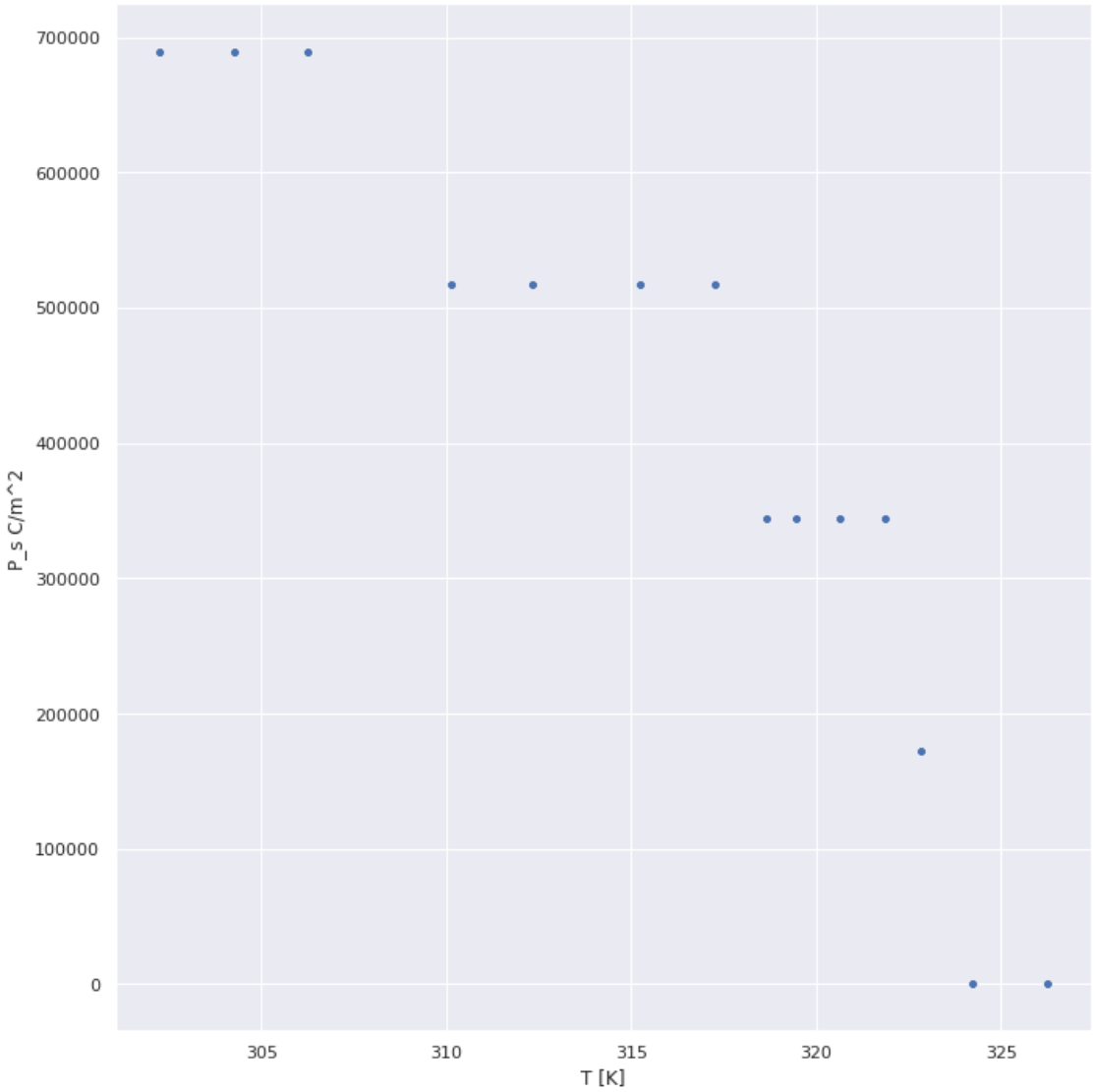
[28]:
```
df_const[' 2 [V]'] = df_const[' 2 [mV]'] / 1000
const =( df_const['R1 [Ω]'] + df_const['R2 [Ω]'])/(df_const[' 2␣
 ↪[V]']*df_const['R2 [Ω]']*df_const['d [m]'])
const = const[0]
df_var['E_c [V/m]'] = df_var['Xc [dz]'] * const
```

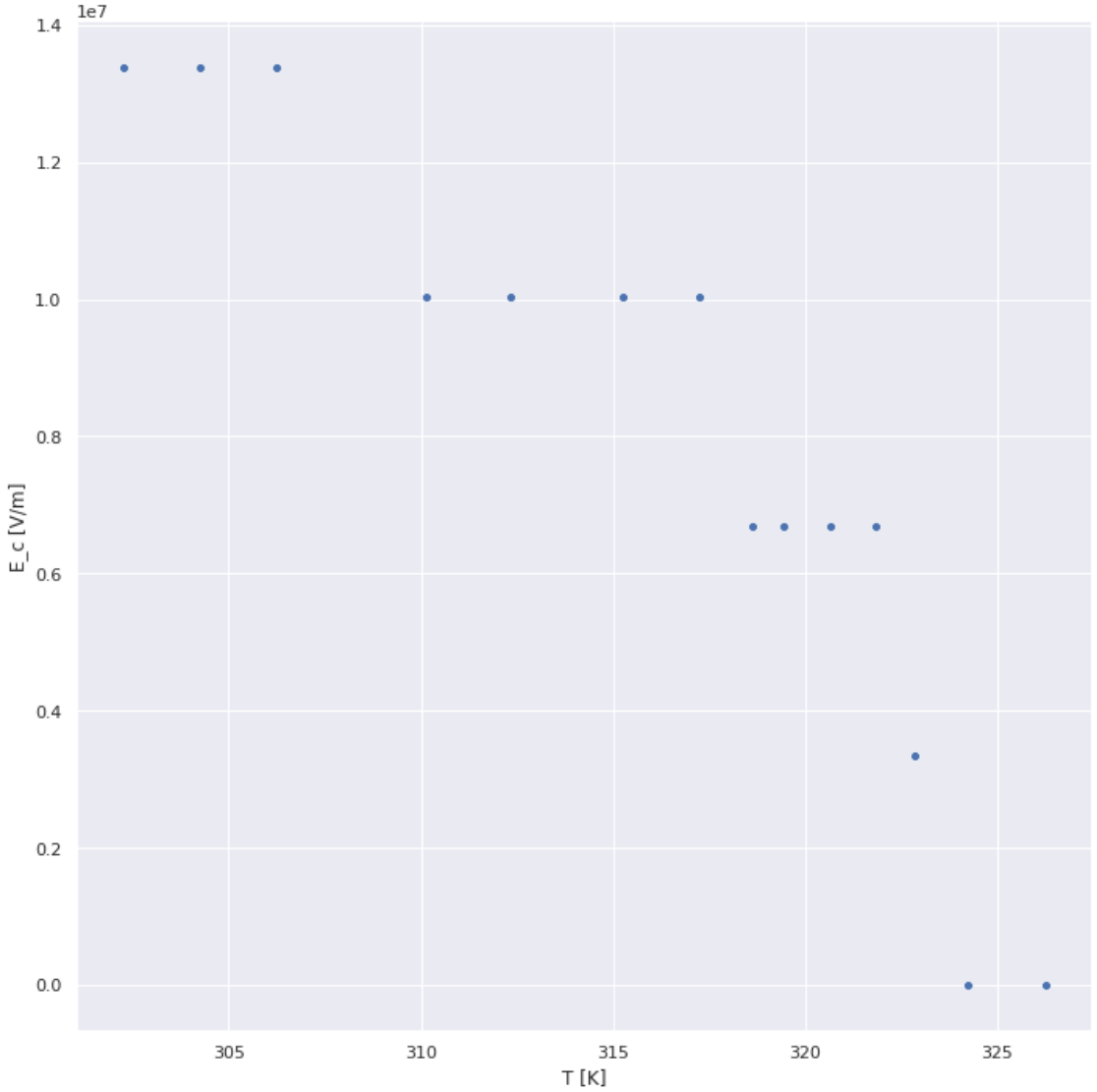
[28]:
```
0     1.337762e+07
1     1.337762e+07
2     1.337762e+07
3     1.003322e+07
4     1.003322e+07
5     1.003322e+07
6     1.003322e+07
7     6.688811e+06
8     6.688811e+06
9     6.688811e+06
10    6.688811e+06
11    3.344406e+06
12    0.000000e+00
13    0.000000e+00
Name: E_c [V/m], dtype: float64
```

[31]:
```
const = (df_const['C_0 [mF]']/1000)/(df_const['S [m^2]'] * df_const[' 1 [V]'])
const = const[0]
df_var['P_s C/m^2'] = df_var['Ys [dz]']*const
```

[35]:
```
sns.relplot(y='P_s C/m^2', x='T [K]', data=df_var, height=10)
```

[35]:
```
<seaborn.axisgrid.FacetGrid at 0x7fd2a6159cd0>
```


700000
600000
500000
400000
300000
200000
100000
0
P_s C/m^2
305
310
315
320
325
T [K]


1e7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
E_c [V/m]
305
310
315
320
325
T [K]

[ ]: