hw1

January 23, 2024

[ ]:

```
import pandas as pd

df = pd.read_excel("DataBrain.xlsx", header=[0, 1, 2, 3])
```

[ ]:

```
df = df.drop([0] + list(range(len(df)-4, len(df))))
df = df.dropna(axis=1)
df
```

[ ]:

```
   Budday17                                                                  \
     CX-ten          CX-com          CX-shr          CR-ten          CR-com   
     lambda       P  lambda       P   gamma       P  lambda       P  lambda   
        [-]   [kPa]     [-]   [kPa]     [-]   [kPa]     [-]   [kPa]     [-]   
1          1  0.0000  1.0000  0.0000  0.0000  0.0000  1.0000  0.0000  1.0000   
2     1.0063  0.0251  0.9938 -0.0308  0.0125  0.0147  1.0063  0.0157  0.9938   
3     1.0125  0.0462  0.9875 -0.0659  0.0250  0.0294  1.0125  0.0235  0.9875   
4     1.0188  0.0666  0.9812 -0.1040  0.0375  0.0486  1.0188  0.0345  0.9812   
5      1.025  0.0838  0.9750 -0.1479  0.0500  0.0633  1.0250  0.0423  0.9750   
6     1.0312  0.1010  0.9688 -0.1908  0.0625  0.0814  1.0312  0.0509  0.9688   
7     1.0375  0.1175  0.9625 -0.2375  0.0750  0.0983  1.0375  0.0572  0.9625   
8     1.0437  0.1324  0.9563 -0.2920  0.0875  0.1186  1.0437  0.0642  0.9563   
9       1.05  0.1488  0.9500 -0.3504  0.1000  0.1412  1.0500  0.0721  0.9500   
10    1.0562  0.1661  0.9437 -0.4127  0.1125  0.1649  1.0562  0.0791  0.9437   
11    1.0625  0.1856  0.9375 -0.4866  0.1250  0.1942  1.0625  0.0869  0.9375   
12    1.0688  0.2091  0.9313 -0.5684  0.1375  0.2292  1.0688  0.0940  0.9313   
13     1.075  0.2366  0.9250 -0.6579  0.1500  0.2698  1.0750  0.1050  0.9250   
14    1.0813  0.2710  0.9187 -0.7630  0.1625  0.3227  1.0813  0.1151  0.9187   
15    1.0875  0.3125  0.9125 -0.8837  0.1750  0.3791  1.0875  0.1292  0.9125   
16    1.0938  0.3650  0.9062 -1.0005  0.1875  0.4557  1.0938  0.1418  0.9062   
17       1.1  0.4151  0.9000 -1.1484  0.2000  0.5435  1.1000  0.1582  0.9000   

            …                                                          \
            …  BG-com          BG-shr          CC-ten          CC-com   
         P  …  lambda       P   gamma       P  lambda       P  lambda   
     [kPa]  …     [-]   [kPa]     [-]   [kPa]     [-]   [kPa]     [-]   
1   0.0000  …  1.0000  0.0000  0.0000  0.0000  1.0000  0.0000  1.0000   
2  -0.0193  …  0.9938 -0.0174  0.0125  0.0070  1.0063  0.0078  0.9938   
3  -0.0387  …  0.9875 -0.0358  0.0250  0.0140  1.0125  0.0149  0.9875   
4  -0.0543  …  0.9812 -0.0534  0.0375  0.0210  1.0188  0.0196  0.9812   
```

```
5  -0.0800  …  0.9750 -0.0778  0.0500  0.0305  1.0250  0.0251  0.9750
6  -0.1040  …  0.9688 -0.1021  0.0625  0.0397  1.0312  0.0298  0.9688
7  -0.1305  …  0.9625 -0.1265  0.0750  0.0488  1.0375  0.0337  0.9625
8  -0.1674  …  0.9563 -0.1479  0.0875  0.0579  1.0437  0.0376  0.9563
9  -0.2024  …  0.9500 -0.1752  0.1000  0.0703  1.0500  0.0415  0.9500
10 -0.2453  …  0.9437 -0.2102  0.1125  0.0805  1.0562  0.0454  0.9437
11 -0.2959  …  0.9375 -0.2414  0.1250  0.0930  1.0625  0.0486  0.9375
12 -0.3543  …  0.9313 -0.2842  0.1375  0.1088  1.0688  0.0533  0.9313
13 -0.4127  …  0.9250 -0.3270  0.1500  0.1257  1.0750  0.0580  0.9250
14 -0.4827  …  0.9187 -0.3776  0.1625  0.1449  1.0813  0.0634  0.9187
15 -0.5723  …  0.9125 -0.4321  0.1750  0.1686  1.0875  0.0697  0.9125
16 -0.6657  …  0.9062 -0.4905  0.1875  0.1969  1.0938  0.0775  0.9062
17 -0.7591  …  0.9000 -0.5528  0.2000  0.2262  1.1000  0.0862  0.9000


           CC-shr
         P  gamma       P
     [kPa]    [-]   [kPa]
1   0.0000 0.0000  0.0000
2  -0.0096 0.0125  0.0036
3  -0.0164 0.0250  0.0072
4  -0.0300 0.0375  0.0109
5  -0.0427 0.0500  0.0170
6  -0.0564 0.0625  0.0217
7  -0.0730 0.0750  0.0319
8  -0.0895 0.0875  0.0342
9  -0.1051 0.1000  0.0422
10 -0.1363 0.1125  0.0468
11 -0.1596 0.1250  0.0558
12 -0.1946 0.1375  0.0627
13 -0.2297 0.1500  0.0751
14 -0.2764 0.1625  0.0853
15 -0.3270 0.1750  0.1011
16 -0.3854 0.1875  0.1192
17 -0.4555 0.2000  0.1429

[17 rows x 24 columns]
```

```
[ ]: df_cr = df.iloc[:, 6:12]
```

```
[ ]: df_cr
```

```
[ ]:   Budday17
         CR-ten          CR-com          CR-shr
         lambda       P  lambda       P   gamma       P
            [-]   [kPa]     [-]   [kPa]     [-]   [kPa]
   1     1.0000  0.0000  1.0000  0.0000  0.0000  0.0000
```

```
2    1.0063  0.0157  0.9938 -0.0193  0.0125  0.0079
3    1.0125  0.0235  0.9875 -0.0387  0.0250  0.0159
4    1.0188  0.0345  0.9812 -0.0543  0.0375  0.0238
5    1.0250  0.0423  0.9750 -0.0800  0.0500  0.0318
6    1.0312  0.0509  0.9688 -0.1040  0.0625  0.0409
7    1.0375  0.0572  0.9625 -0.1305  0.0750  0.0488
8    1.0437  0.0642  0.9563 -0.1674  0.0875  0.0601
9    1.0500  0.0721  0.9500 -0.2024  0.1000  0.0681
10   1.0562  0.0791  0.9437 -0.2453  0.1125  0.0817
11   1.0625  0.0869  0.9375 -0.2959  0.1250  0.0964
12   1.0688  0.0940  0.9313 -0.3543  0.1375  0.1133
13   1.0750  0.1050  0.9250 -0.4127  0.1500  0.1347
14   1.0813  0.1151  0.9187 -0.4827  0.1625  0.1596
15   1.0875  0.1292  0.9125 -0.5723  0.1750  0.1878
16   1.0938  0.1418  0.9062 -0.6657  0.1875  0.2227
17   1.1000  0.1582  0.9000 -0.7591  0.2000  0.2611
```

```
[ ]: df_cxr_reg = df_cr.copy()
     columns = [
         ('Budday17', 'CR-ten', 'epsilon', '[-]'),
         ('Budday17', 'CR-ten', 'sigma', '[kPa]'),
         ('Budday17', 'CR-com', 'epsilon', '[-]'),
         ('Budday17', 'CR-com', 'sigma', '[kPa]'),
         ('Budday17', 'CR-shr', 'gamma', '[-]'),
         ('Budday17', 'CR-shr', 'tau', '[kPa]')
     ]
     df_cxr_reg.columns=pd.MultiIndex.from_tuples(columns)
     df_cxr_reg.iloc[:, [0, 2]] = df_cxr_reg.iloc[:, [0, 2]] - 1
     df_cxr_reg
```

```
[ ]:    Budday17
          CR-ten          CR-com          CR-shr
         epsilon   sigma epsilon   sigma   gamma     tau
             [-]   [kPa]     [-]   [kPa]     [-]   [kPa]
     1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
     2    0.0063  0.0157 -0.0062 -0.0193  0.0125  0.0079
     3    0.0125  0.0235 -0.0125 -0.0387  0.0250  0.0159
     4    0.0188  0.0345 -0.0188 -0.0543  0.0375  0.0238
     5    0.0250  0.0423 -0.0250 -0.0800  0.0500  0.0318
     6    0.0312  0.0509 -0.0312 -0.1040  0.0625  0.0409
     7    0.0375  0.0572 -0.0375 -0.1305  0.0750  0.0488
     8    0.0437  0.0642 -0.0437 -0.1674  0.0875  0.0601
     9    0.0500  0.0721 -0.0500 -0.2024  0.1000  0.0681
     10   0.0562  0.0791 -0.0563 -0.2453  0.1125  0.0817
     11   0.0625  0.0869 -0.0625 -0.2959  0.1250  0.0964
     12   0.0688  0.0940 -0.0687 -0.3543  0.1375  0.1133
     13   0.0750  0.1050 -0.0750 -0.4127  0.1500  0.1347
```

```
14   0.0813  0.1151 -0.0813 -0.4827  0.1625  0.1596
15   0.0875  0.1292 -0.0875 -0.5723  0.1750  0.1878
16   0.0938  0.1418 -0.0938 -0.6657  0.1875  0.2227
17   0.1000  0.1582 -0.1000 -0.7591  0.2000  0.2611
```

[ ]:

```
from sklearn.linear_model import LinearRegression

tension = LinearRegression()
tension.fit(df_cxr_reg.iloc[:, [0]], df_cxr_reg.iloc[:, [1]])

compression = LinearRegression()
compression.fit(df_cxr_reg.iloc[:, [2]], df_cxr_reg.iloc[:, [3]])

shear = LinearRegression()
shear.fit(df_cxr_reg.iloc[:, [4]], df_cxr_reg.iloc[:, [5]])

Eten = tension.coef_[0][0]
Ecom = compression.coef_[0][0]
mu = shear.coef_[0][0]

print(f"Eten = {Eten:.3f}")
print(f"Ecom = {Ecom:.3f}")
print(f"mu = {mu:.3f}")
```

```
Eten = 1.427
Ecom = 7.226
mu = 1.186
```

[ ]:

```
nu = 0.5
Eshr = 3 * mu * (1 + nu)
print(f"Eshr = {Eshr:.3f}")
```

```
Eshr = 5.337
```

[ ]:

```
import numpy as np

mean_elastic_modulus = np.mean([Eten, Ecom, Eshr])
print(f"mean_elastic_modulus = {mean_elastic_modulus:.3f}")
```

```
mean_elastic_modulus = 4.663
```

[ ]:

```
tension_r2 = tension.score(df_cxr_reg.iloc[:, [0]], df_cxr_reg.iloc[:, [1]])
compression_r2 = compression.score(df_cxr_reg.iloc[:, [2]], df_cxr_reg.iloc[:,␣
 ↪[3]])
shear_r2 = shear.score(df_cxr_reg.iloc[:, [4]], df_cxr_reg.iloc[:, [5]])

print(f"R2 score for tension: {tension_r2:.3f}")
print(f"R2 score for compression: {compression_r2:.3f}")
print(f"R2 score for shear: {shear_r2:.3f}")
```

```
R2 score for tension: 0.988
R2 score for compression: 0.930
R2 score for shear: 0.920
```

[ ]:

```python
from sklearn.metrics import mean_squared_error

tension_mse = mean_squared_error(df_cxr_reg.iloc[:, [1]], tension.
  ↪predict(df_cxr_reg.iloc[:, [0]]))
compression_mse = mean_squared_error(df_cxr_reg.iloc[:, [3]], compression.
  ↪predict(df_cxr_reg.iloc[:, [2]]))
shear_mse = mean_squared_error(df_cxr_reg.iloc[:, [5]], shear.
  ↪predict(df_cxr_reg.iloc[:, [4]]))

print(f"MSE for tension: {tension_mse:.6f}")
print(f"MSE for compression: {compression_mse:.6f}")
print(f"MSE for shear: {shear_mse:.6f}")
```

```
MSE for tension: 0.000024
MSE for compression: 0.003673
MSE for shear: 0.000460
```

[ ]: