In [1]:

```
# data
import pandas as pd
import numpy as np

# Visualization
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import Markdown as mk, display

pd.options.mode.copy_on_write = True  # to avoid SettingWithCopyWarning,
```

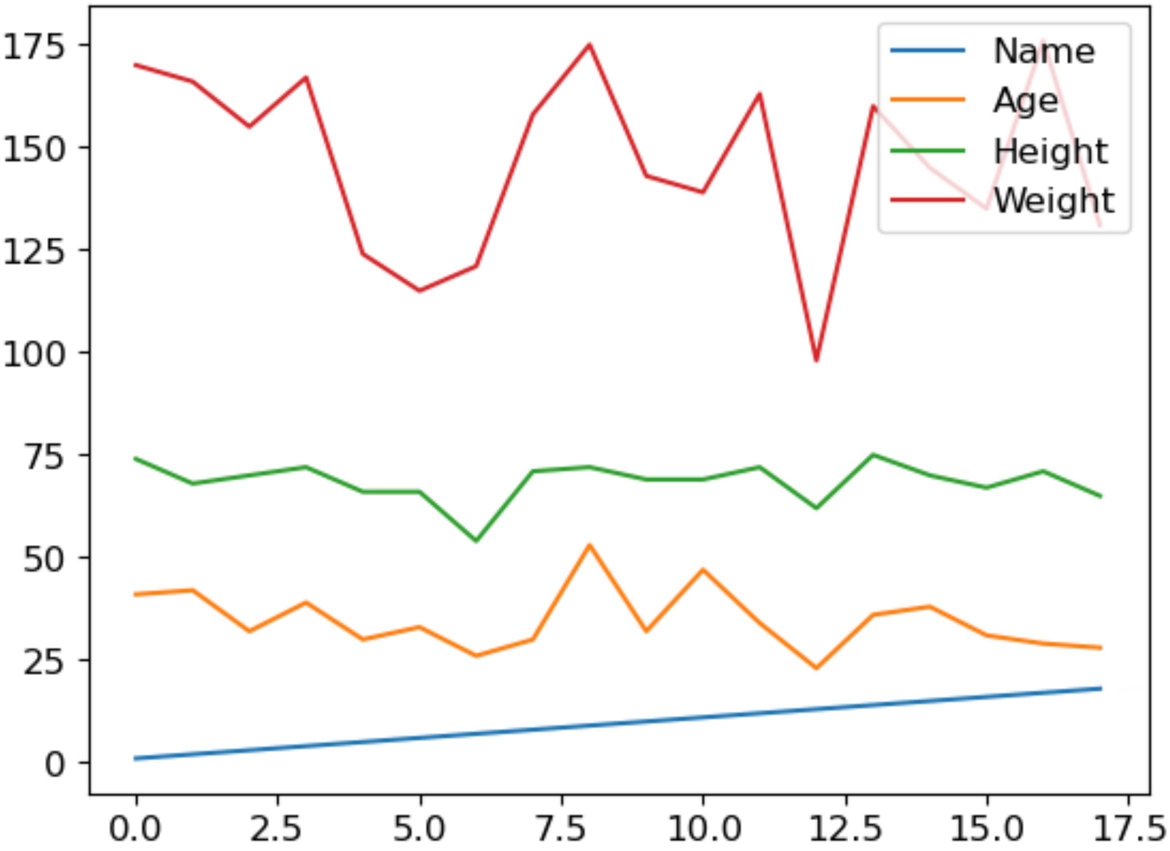
Exercise 2

In [134…

```
data = pd.read_csv("/Users/steenbender/Desktop/PhD/nanokemi/plot_i_python
data.plot()
```

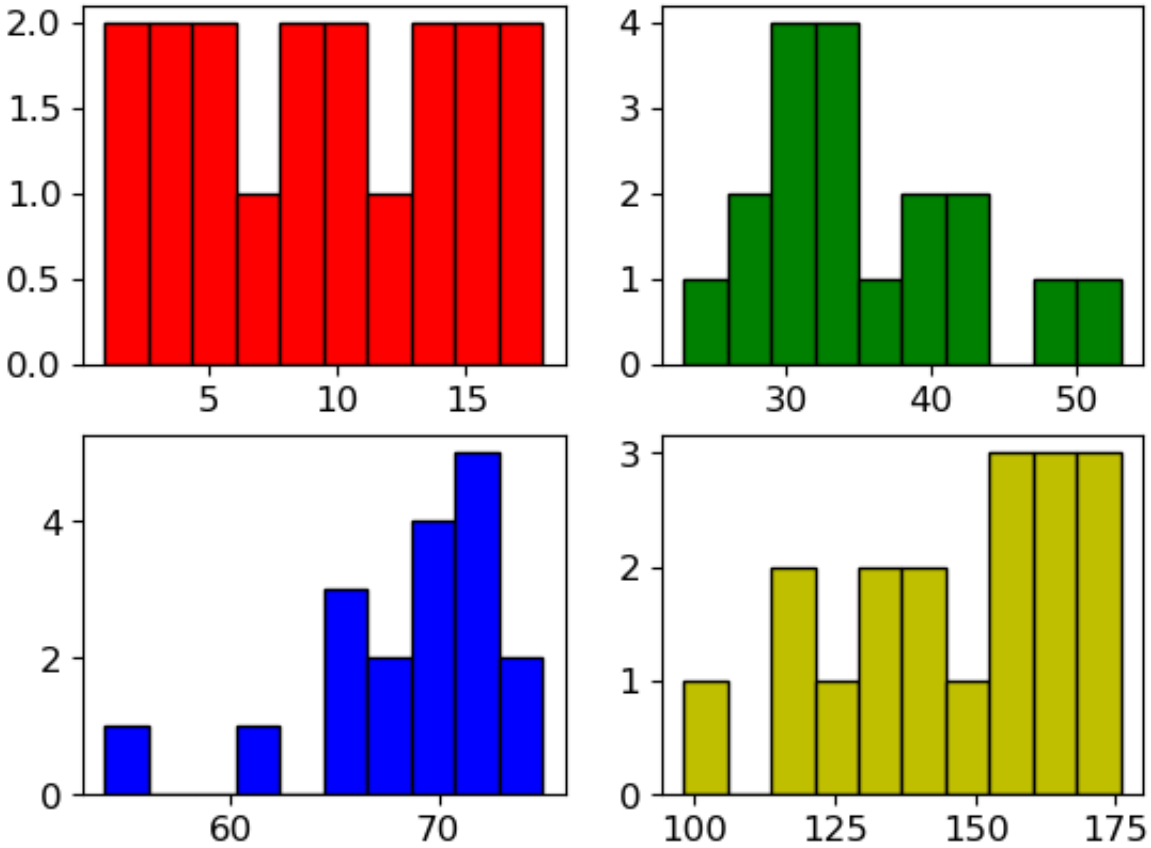
Out[134…

```
<Axes: >
```



In [135…

```
data_values = data[names].values
fig, ax = plt.subplots(2, 2)
ax = ax.flatten()
colors = ["r", "g", "b", "y"]
for i, dat in enumerate(data_values.T):
    ax[i].hist(dat, bins=10, color=colors[i], edgecolor="black")
```

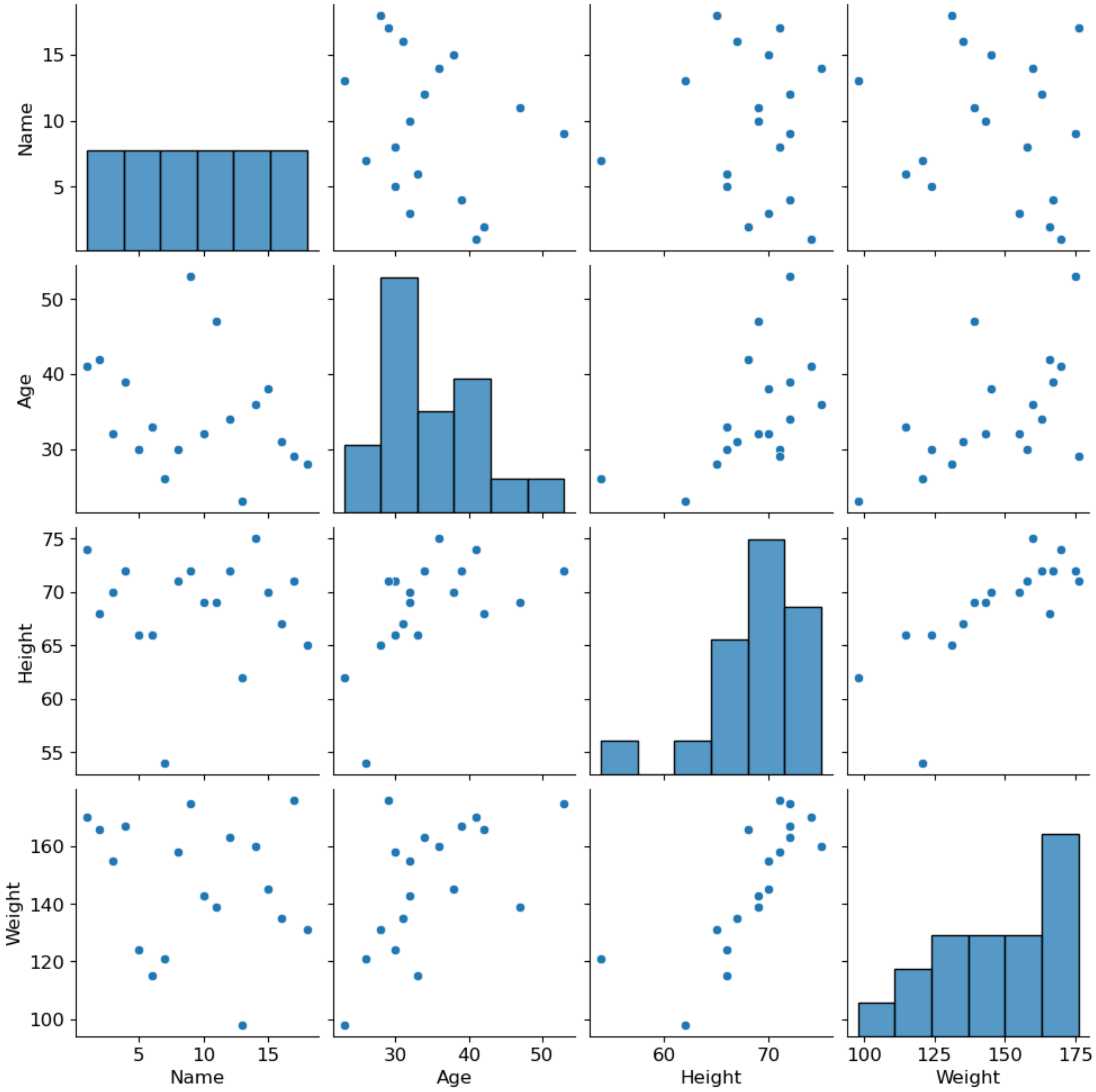


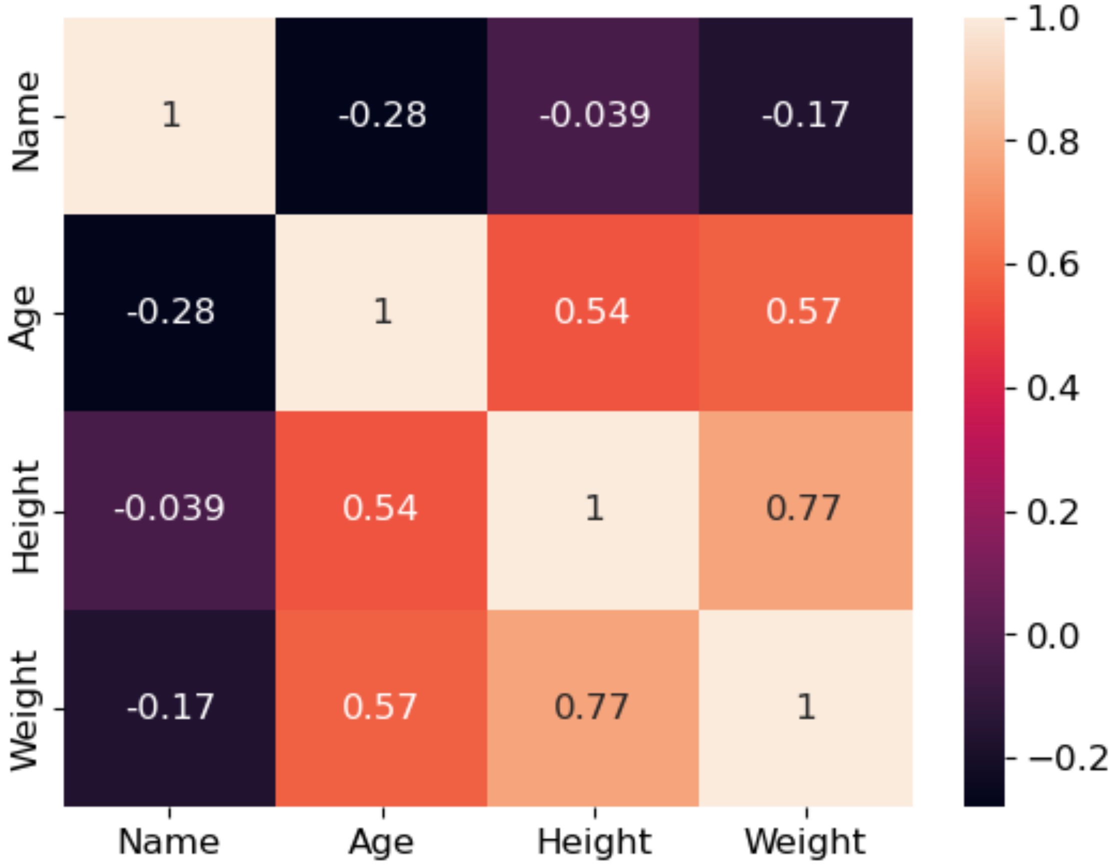
In [140…

```
names = ["Name", "Age", "Height", "Weight"]
corr = data[names].corr()
sns.pairplot(data[names])
fig, ax = plt.subplots()
sns.heatmap(corr, annot=True)
```

Out[140…

```
<Axes: >
```


Name
15
10
5
Age
50
40
30
Height
75
70
65
60
55
Weight
160
140
120
100
5
10
15
Name
30
40
50
Age
60
70
Height
100
125
150
175
Weight



Exercise 3

In [126…

```
data = pd.read_csv(
    "/Users/steenbender/Desktop/PhD/nanokemi/plot_i_python/example_1.csv"
    skiprows=17,
    skipfooter=9,
    engine="python",
)
data
```

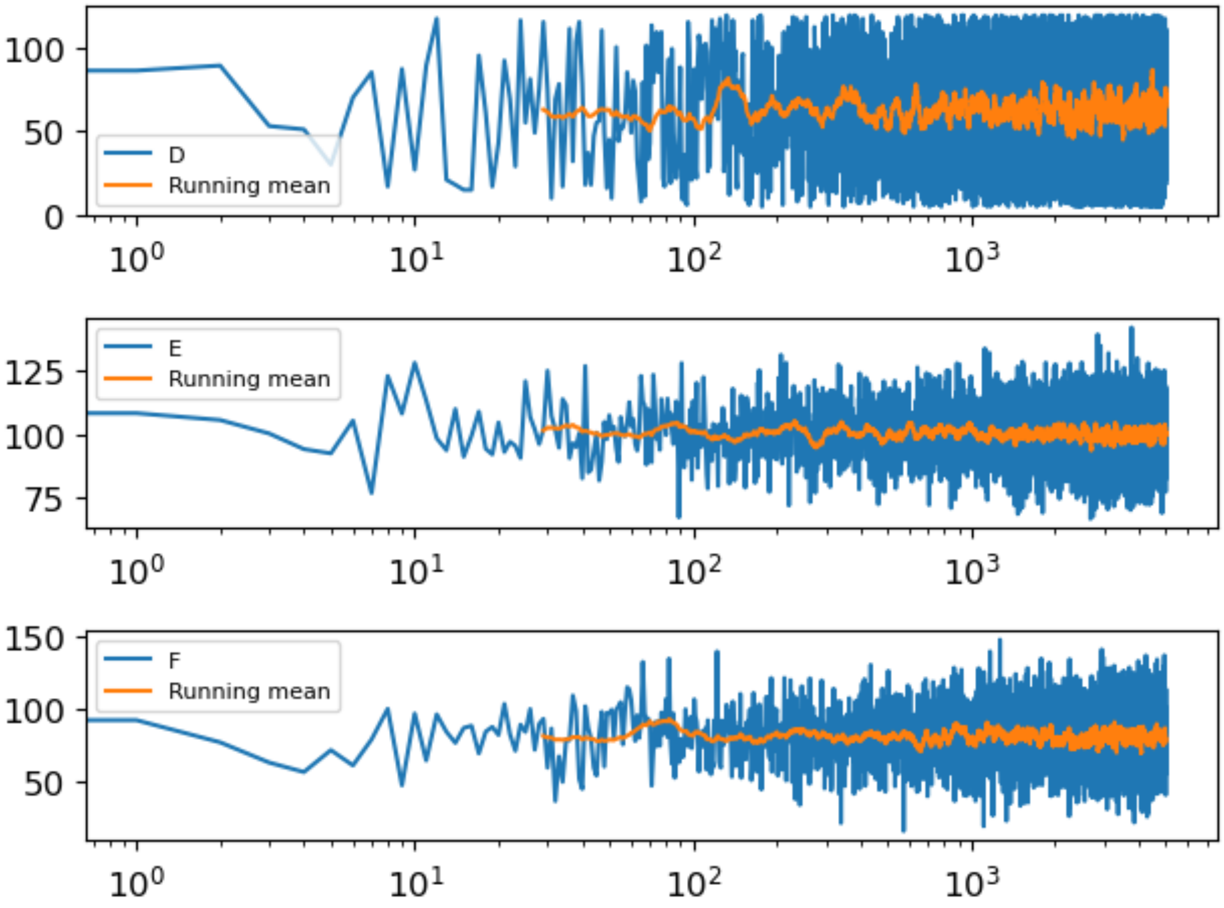
Out[126…

	Name	Sex	Age	Height	Weight
0	1	M	41	74	170
1	2	M	42	68	166
2	3	M	32	70	155
3	4	M	39	72	167
4	5	F	30	66	124
5	6	F	33	66	115
6	7	F	26	54	121
7	8	M	30	71	158
8	9	M	53	72	175
9	10	M	32	69	143
10	11	F	47	69	139
11	12	M	34	72	163
12	13	F	23	62	98
13	14	M	36	75	160
14	15	M	38	70	145
15	16	F	31	67	135
16	17	M	29	71	176
17	18	F	28	65	131

Exercise 4

In [128…

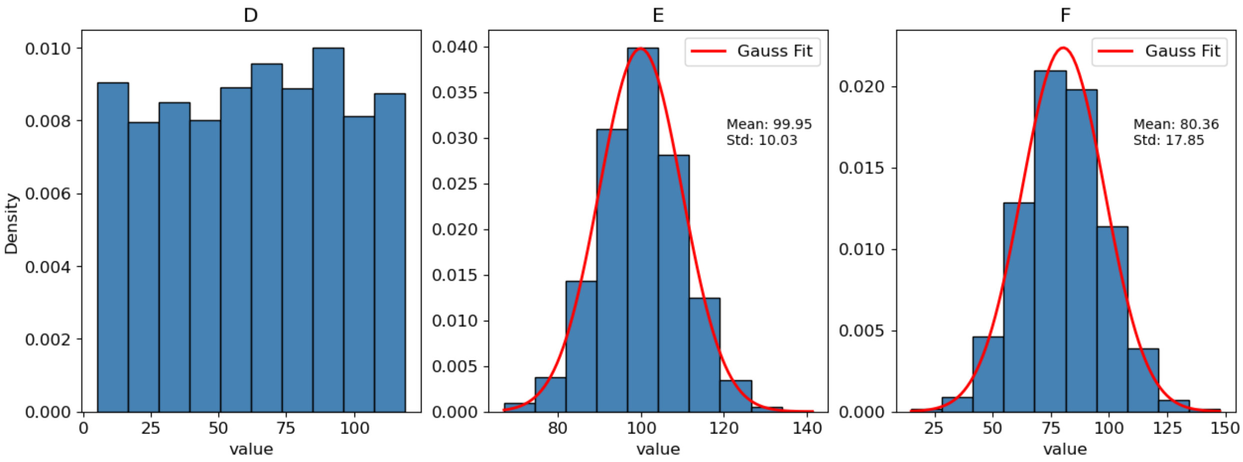
```
data = pd.read_csv(
    "/Users/steenbender/Desktop/PhD/nanokemi/plot_i_python/test_data2.csv
    index_col=0,
    skiprows=1,
    names=["D", "E", "F"],
)
running_mean = data.rolling(window=30).mean()
fig, ax = plt.subplots(3, 1)
ax = ax.flatten()
for i, col in enumerate(data.columns):
    ax[i].plot(data[col], label=col)
    ax[i].plot(running_mean[col], label="Running mean")
    ax[i].legend(fontsize=8)
    ax[i].set_xscale("log")
fig.tight_layout()
```



In [ ]:

```
# Fit a line to the data
from scipy import stats
from scipy.optimize import curve_fit


fig_hist, ax_hist = plt.subplots(1, 3, figsize=(15, 5))
for i, col in enumerate(data.columns):
    counts, bins, _ = ax_hist[i].hist(
        data[col], bins=10, color="steelblue", edgecolor="black", density
    )
    if i > 0:
        x_range = np.linspace(bins.min(), bins.max(), 100)
        fit = stats.norm.pdf(x_range, np.mean(data[col]), np.std(data[col
        text = f"Mean: {np.mean(data[col]):.2f}\nStd: {np.std(data[col]):
        ax_hist[i].plot(x_range, fit, label="Gauss Fit", lw=2, color="red
        ax_hist[i].text(0.7, 0.7, text, transform=ax_hist[i].transAxes, f
        ax_hist[i].legend()
    else:
        ax_hist[i].set_ylabel("Density")
    ax_hist[i].set_xlabel("value")
    ax_hist[i].set_title(col)
```

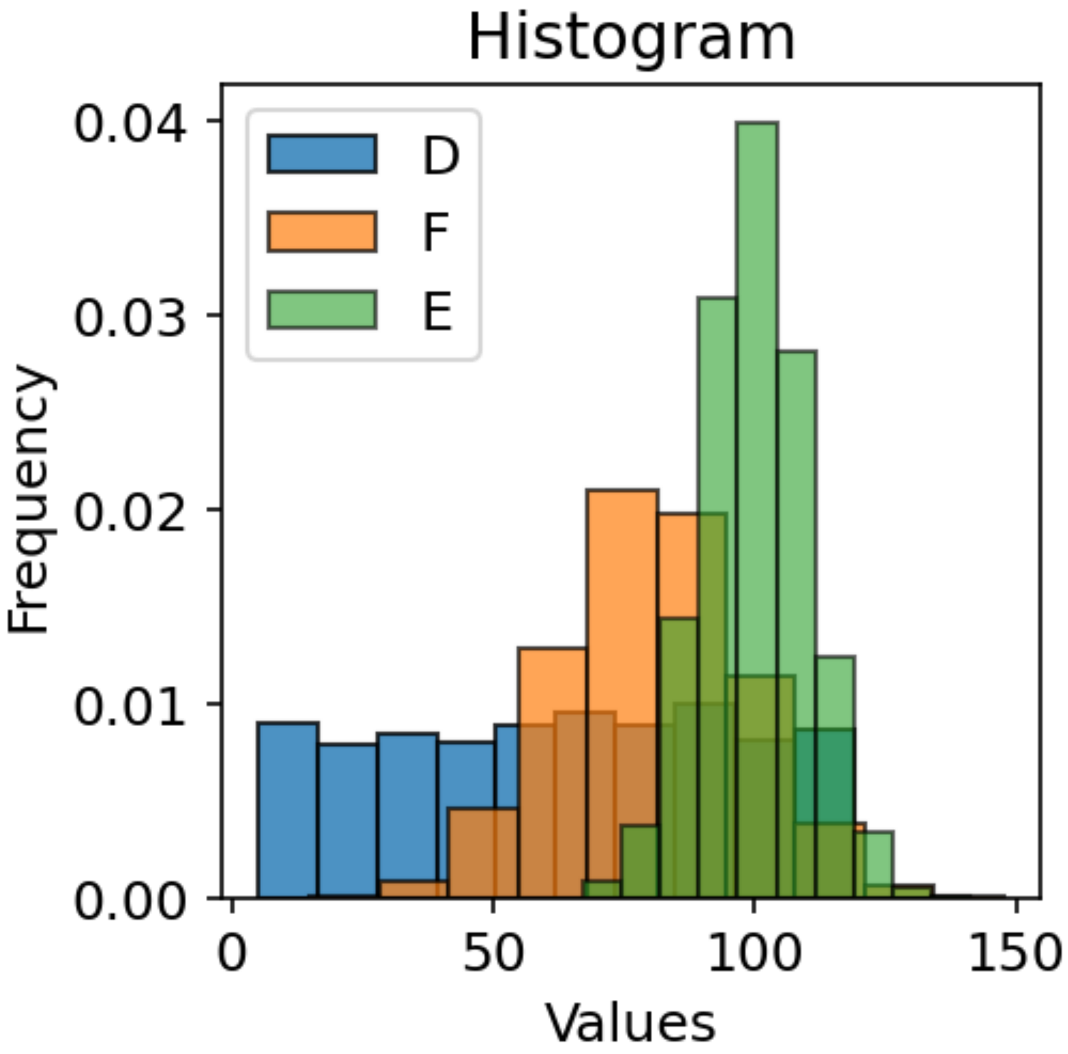


In [132…

```
plt.rcParams.update({"font.size": 12})
fig, ax = plt.subplots(figsize=(3.3, 3.3), dpi=150)
ax.hist(data.D, bins=10, edgecolor="black", alpha=0.8, density=True, labe
ax.hist(data.F, bins=10, edgecolor="black", alpha=0.7, density=True, labe
ax.hist(data.E, bins=10, edgecolor="black", alpha=0.6, density=True, labe
ax.legend()
ax.set(xlabel="Values", ylabel="Frequency", title="Histogram")
```

Out[132…

```
[Text(0.5, 0, 'Values'),
 Text(0, 0.5, 'Frequency'),
 Text(0.5, 1.0, 'Histogram')]
```



In [ ]:

```
test = pd.read_csv(
    "/Users/steenbender/Desktop/PhD/nanokemi/plot_i_python/All ligands an
    skiprows=46,
    sep="\t",
    decimal=",",
)
# test_col = test["A1"]
# split_row = test[test["A1"].isnull()]
# df_1 = test.iloc[: split_row.index[0], :]
# df_2 = test.iloc[split_row.index[0] + 2 :, :]
# df_1.loc[:, "Kinetic read"] = pd.to_timedelta(df_1["Kinetic read"]).dt.
# well_names = df_1.columns[1:]
# correct_values = df_1.loc[:, well_names].apply(lambda x: x.str.replace(
# df_1.loc[:, well_names] = correct_values
# df_1 = df_1.astype(float)
```

In [119…

```
test
```

Out[119…

	Kinetic read	A1	A2	A3	B1	B2	B3	C1	C2	C3	...	
0	0:00:04	0,340	0,348	0,318	0,221	0,397	0,224	0,296	0,315	0,262	...	0
1	0:00:11	0,341	0,346	0,322	0,233	0,400	0,232	0,298	0,315	0,265	...	0
2	0:00:18	0,343	0,345	0,324	0,242	0,402	0,241	0,299	0,315	0,266	...	0
3	0:00:25	0,344	0,346	0,325	0,250	0,404	0,247	0,300	0,315	0,268	...	0
4	0:00:32	0,346	0,347	0,326	0,257	0,406	0,254	0,301	0,315	0,269	...	0
...	...	...	...	...	...	...	...	...	...	...	...	
200	646	0,326	0,360	0,304	0,338	0,403	0,318	0,278	0,307	0,262	...	0
201	647	0,326	0,360	0,304	0,338	0,403	0,318	0,278	0,307	0,262	...	0
202	648	0,325	0,360	0,303	0,338	0,403	0,318	0,278	0,306	0,262	...	0
203	649	0,325	0,359	0,303	0,338	0,403	0,317	0,278	0,306	0,262	...	0
204	650	0,325	0,359	0,303	0,338	0,403	0,317	0,278	0,306	0,262	...	0

205 rows × 22 columns

In [ ]: