**Zadanie 1: False Discovery** W danych symulujących 1000 szeregów czasowych poniżej około 4.5% trendów, wykrywanych przez test nachylenia regresji, raportowanych jest błędnie (przypadkowe p-value). Użyj korekty dla False Discovery Rate tak by wykryć trendy z błędem false-positive około 1% lub mniejszym, zademonstruj to w macierzy (confusion matrix). Skomentuj wyniki, czy redukcja “false positive” ma efekty uboczne?

from scipy import stats

import matplotlib.pyplot as plt

import numpy as np

R = 1000

N = 10

Y = stats.norm(0,0.1).rvs(size=(R,N))

Y[:100] = np.linspace(0,1,N) + stats.norm(0,0.2).rvs(size=(100,N)) # 10% of trends are real

x = np.linspace(0,1,10)

pvals = np.array([stats.linregress(x,y).pvalue for y in Y])

extreme\_id = pvals[100:].argmin()

plt.plot(x,Y[extreme\_id],label='samples')

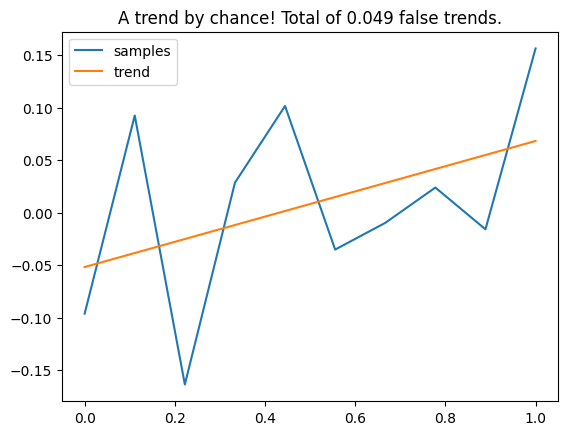
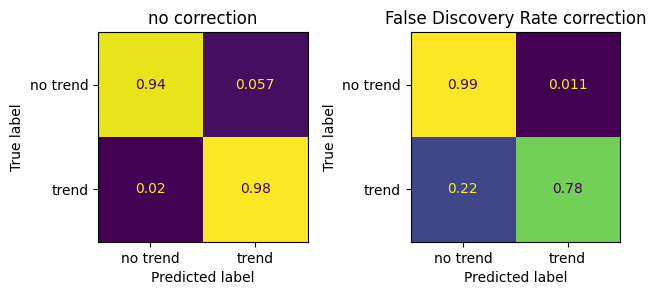
res = stats.linregress(x,Y[extreme\_id])

plt.plot(x,x\*res.slope+res.intercept,label='trend')

plt.title(f"A trend by chance! Total of {(pvals<0.05).mean()-0.1:.3f} false trends.")

plt.legend()

plt.show()

****

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import ConfusionMatrixDisplay, confusion\_matrix

import matplotlib.pyplot as plt

fig,axs = plt.subplots(1,2)

trend\_predicted = pvals < 0.05

trend\_true = np.arange(len(pvals)) < 100

ax = axs[0]

cm = confusion\_matrix(trend\_true, trend\_predicted, normalize='true')

ConfusionMatrixDisplay(cm, display\_labels=['no trend','trend']).plot(ax=axs[0],colorbar=False)

ax.set\_title("no correction")

from statsmodels.stats.multitest import multipletests

#FIXME: use multipletests to correct FDR values as in the notebook, pvals\_corr = multipletests(...)

trend\_predicted = pvals\_corr < 0.05

trend\_true = np.arange(len(pvals)) < 100

ax = axs[1]

cm = confusion\_matrix(trend\_true, trend\_predicted, normalize='true')

ConfusionMatrixDisplay(cm, display\_labels=['no trend','trend']).plot(ax=ax,colorbar=False)

ax.set\_title("False Discovery Rate correction")

plt.tight\_layout()

plt.show()

**Zadanie 2: CI a siła testu** Losujemy próbę wielkości N z rozkładu normalnego o nieznanej średniej <mu> i wariancji 1. Chcemy testować czy mu=0 (null) czy raczej |mu|>=0.30 (alternatywa). Jak używając przedziału CI 95% dla średniej (patrz notebook 2) zbudować test dla którego oba błędy (typu I i typu II) są mniejsze niż 5%? Jakie N wystarczy? **Zadanie 3: Istotność vs efekt**. Artykuł [Sample size, power and effect size revisited: simplified and practical approaches in pre-clinical, clinical and laboratory studies](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7745163/) traktuje o różnicy pomiędzy statystyczną istotnością wykazania różnicy (significance) a biznesową użytecznością odkrytej różnicy (impact, effect size). Na przykład, zmiana cholesterolu o co najmniej 0.5mmol/L może być uznana za godną uwagi w kontekście badań klinicznych.

Significance vs effect size


W badaniu “[Changes in Blood Lipid Levels After a Digitally Enabled Cardiometabolic Preventive Health Program: Pre-Post Study in an Adult Dutch General Population Cohort](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8987960/)” poniżej podano średni poziom cholesterolu wraz z odchyleniem standardowym (w formacie mean (sd)). Oszacuj 95% CI dla średniego poziomu cholesterolu obu grupach. Czy zachodzą istotne różnice pomiędzy grupami? Czy przekraczają 0.5 mmol / L (\*można w uproszczeniu porównać na podstawie końców przedziałów)?

|  | **Baseline (N=348)** | **Cardiometabolic intervention (N=104)** | **P -value** |
| --- | --- | --- | --- |
| **Characteristic** |  |  |  |
| **Age (years), mean (SD)** | 44.6 (11.1) | 49.4 (9.0) | .02 |
| **Sex (female), n (%)** | 195 (56.0%) | 38 (36.5%) | .09 |
| **Weight (kg), mean (SD)** | 77.2 (14.4) | 83.5 (12.2) | .02 |
| **BMI (kg/m2), mean (SD)** | 25.0 (4.7) | 26.6 (2.9) | .01 |
| **Body fat percentage, mean (SD)** | 24.9 (9.8) | 26.7 (7.7) | .26 |
| **Total cholesterol level (mmol/L), mean (SD)** | 5.10 (1.06) | 6.00 (1.01) | <.001 |

**Zadanie 4: Wizualizacja struktur**. Zademonstruj jak użyć metody UMAP do pokazania struktury klastrów w wybranych przez siebie danych (łatwo znaleźć na OpenML). Przykład:

#%pip install openml --quiet

#%pip install umap-learn --quiet

import sklearn

import umap

import matplotlib.pyplot as plt

import openml

import seaborn as sns

# get dataset from OpenML

dataset = openml.datasets.get\_dataset('WineDataset') # find an interesting dataset

X, y, \_, \_ = dataset.get\_data(target="Wine") # figure out the right target

# use UMAP for powerful dimension-reduction and visualization!

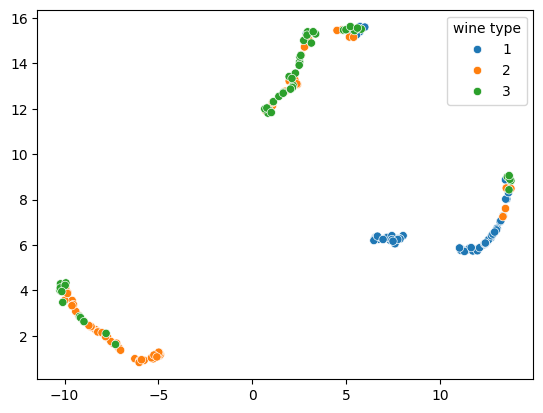
X\_embed = umap.UMAP(random\_state=42).fit\_transform(X)

fig = sns.scatterplot(x=X\_embed[:, 0], y=X\_embed[:, 1], hue=y.astype(str)) # each class gets its own color

fig.legend(title="wine type")

plt.show()

# check for cool patterns!

 **Zadanie 5 Hyperparameter Tunning** Exercise 15 z Notebooka 4