

## Úloha D

Do triedy `Model` sme pridali metódu na výpočet R-kvadrátu podľa definície :

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

```
def r2(self) -> float:

    if len(self.beta) == 0:
        print('Model is not solved')
        return 0.0

    y_hat = self.beta[0] + np.dot(self.x_vect.transpose(), self.beta[1:])
    y_mean = np.mean(self.y)

    res1 = 0
    res2 = 0

    for i in range(len(self.y)):
        res1 += (self.y[i] - y_hat[i]) ** 2
        res2 += (self.y[i] - y_mean) ** 2

    result = 1 - (res1 / res2)
    return result
```

Implementujeme metódu na dátach `A04wine.csv` :

- načítame dáta pomocou `pandas`, rozdelíme ich do premenných (rovnako ako v predošlých úlohách)

```
data = pd.read_csv('data/A04wine.csv')
y = data['Price']
x = data[['WinterRain', 'AGST', 'HarvestRain', 'Age', 'FrancePop']]
x = x.to_numpy().transpose()
```

- naimportujeme predom zadefinované `L1` a `LInf` modely
- zostavíme LP problémy cez obe modely pre načítané dáta a vyriešime ich

```
# utilize developed L1 and LInf regression model classes
l1_model = L1Model(y, x)
linf_model = LInfModel(y, x)

# solve LP problems
l1_model.solve()
linf_model.solve()
```

- po vyriešení zavoláme na dané modely metódu `r2()`, čím získame príslušné R-kvadráty

```
# calculate R-squared coefficient
print(f'R-squared for L1 regression on wine data: {l1_model.r2()}')
print(f'R-squared for LInf regression on wine data: {linf_model.r2()}')
```

Podľa modelov sme dostali nasledujúce výsledky:

$$R_{l_1}^2 \approx 0.78813$$

$$R_{l_\infty}^2 \approx 0.80649$$

Z čoho vieme povedať, že pre dané dáta je pre predikovanie o trochu presnejšia Chebysheva norma.