# Úloha C

V tejto úlohe sa snažíme predikovať kvalitu vína, inšpirovaní prístupom Orleya Ashenfeltera k predikcii cien vína z Bordeaux.

Využívame dáta zo súboru `A04wine.csv` a aplikujeme modely $L^1$ a $L^\infty$ z úlohy A. Budeme využívať podobný postup ako v B. Na implementáciu formulovaných LP úloh využívame opäť:

- `pandas` - načítanie dát z `csv` súboru
- `numpy` - tvorenie matíc a vektorov
- `scipy.optimize` - implementovaný LP solver

Vyberieme z dát dané nezávislé premenné `x` a závislú premennú `y`

```
y = data['Price']
x = data[['WinterRain','AGST', 'HarvestRain', 'Age', 'FrancePop']]
```

Zistíme počet premenných (plus 1 pre intercept)

```
numberOfVariablesBeta = x.shape[1] + 1
```

Vytvoríme potrebné štruktúry pre zostavenie modelu normy `l1`

```
c = np.array([0]*numberOfVariablesBeta + [1] * len(x.values)) #
                                        Objective function coefficients

ALeft = np.matrix([ [1] * len(x.values)]).transpose() # Coefficients for
                                         beta0
ARigth = np.matrix(x.values) # Coefficients for other independent
                                         variables beta
A = np.block([ALeft, ARigth]) # Concenate coefficients of variables into
                                         one matrix

I = np.identity(len(x.values)) # Identity matrix for intercept term
```

Naformulujeme problém a vyriešime pomocou `scipy.optimize.linprog`

```
A_ub = np.block([[-A, -I], [A, -I]])
b_ub = np.concatenate([-y, y])

solve = linprog(c, A_ub, b_ub, bounds = [(None,None)]*
                                        numberOfVariablesBeta +[(0, None)])
```

Po vyriešení vyberieme z riešenia koeficienty

```
betas = solve.x[:numberOfVariablesBeta]
```

Čo nám dá:

$$\beta_0^{(1)} \approx -8.8801 \cdot 10^{-1},\ \beta_1^{(1)} \approx 1.5793 \cdot 10^{-3},\ \beta_2^{(1)} \approx 5.2130 \cdot 10^{-1}$$
$$\beta_3^{(1)} \approx -4.5137 \cdot 10^{-3},\ \beta_4^{(1)} \approx 1.1300 \cdot 10^{-2},\ \beta_5^{(1)} \approx -2.2111 \cdot 10^{-5}$$

Ďalej zostrojíme relevantné štruktúry a naformulujeme LP pre $L^\infty$ normu:

```
c_inf = np.array([0]*numberOfVariablesBeta + [1]) # Koeficienty cielovej
                                                  funkcie
A_inf = np.block([ALeft, ARigth]) # Koeficienty pre nezavisle premenne
                                  pre l_inf normu
i_inf = np.array([ [1] * len(x.values)]).transpose() # Koeficienty pre
                                                     intercept term pre l_inf normu

A_ub_inf = np.block([[-A_inf,-i_inf],[A_inf,-i_inf]])
b_ub_inf = np.concatenate([-y, y])
```

Vyriešime aj tento problém pomocou `scipy.optimize.linprog()` pre $L^\infty$ normu a vyberieme $\beta$ koeficienty:

```
solve_inf = linprog(c_inf, A_ub_inf, b_ub_inf, bounds=[(None,None)]*
                                                numberOfVariablesBeta+[(0,None)])
betas_inf = solve_inf.x[:numberOfVariablesBeta]
```

Po čom dostaneme:

$$\beta_0^{(\infty)} \approx 3.4841,\ \beta_1^{(\infty)} \approx 8.3399 \cdot 10^{-4},\ \beta_2^{(\infty)} \approx 6.0027 \cdot 10^{-1}$$
$$\beta_3^{(\infty)} \approx -3.3416 \cdot 10^{-3},\ \beta_4^{(\infty)} \approx -2.3036 \cdot 10^{-2},\ \beta_5^{(\infty)} \approx -1.1958 \cdot 10^{-4}$$