Problem 3: Linjär regression

Linjär regression är en metod för att modellera sambandet mellan en beroende variabel y och en eller flera oberoende variabler x genom en linjär funktion. Grundidén är att hitta den linje som bäst passar data genom att minimera summan av kvadrerade fel.
För enkel linjär regression har vi modellen:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, 2, \ldots, n \tag{1}$$

där β_0 är intercept, β_1 är lutningen, och ε_i är feltermer som antas vara oberoende och normalfördelade med väntevärde 0 och varians σ^2.
Minsta kvadratmetoden minimerar objektfunktionen:

$$Q(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{2}$$

Felet kvadreras för att göra alla avvikelser positiva (så att positiva och negativa fel inte tar ut varandra), ge större vikt åt stora fel än små, och göra funktionen matematiskt hanterbar med en unik minimumpunkt.
Genom att sätta partialderivatorna lika med noll får vi normalekvationerna, vilka kan lösas analytiskt för att ge skattningarna $\hat{\beta}_0$ och $\hat{\beta}_1$.
I vektorform kan modellen skrivas som:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3}$$

där $\mathbf{X}$ är designmatrisen, $\boldsymbol{\beta}$ är parametervektorn, och lösningen blir:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4}$$

Funktionen tools.regress

Filen `tools.py` innehåller funktionen `regress` som implementerar multipel linjär regression med minsta kvadratmetoden. Funktionen löser modellen:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{5}$$

där $\mathbf{y}$ är en vektor av observerade värden med form $(n,)$, $\mathbf{X}$ är en matris av regressorer med form (n, p), $\boldsymbol{\beta}$ är parametervektorn med form $(p,)$, och $\boldsymbol{\varepsilon}$ är en vektor av slumpmässiga fel med form $(n,)$.
Funktionen regress har följande egenskaper:

1. **QR-faktorisering:** Funktionen använder QR-faktorisering för att lösa normalekvationerna numeriskt stabilt, istället för att direkt invertera $\mathbf{X}^T\mathbf{X}$.

2. **NaN-hantering:** Alla rader som innehåller NaN-värden i antingen $\mathbf{X}$ eller $\mathbf{y}$ filtreras bort innan beräkningarna.

3. **Konfidensintervall:** Funktionen beräknar konfidensintervall för varje parameter baserat på t-fördelningen med $n - p$ frihetsgrader, där n är antalet observationer och p är antalet parametrar.

4. **Standardfel:** Standardfelet för varje parameter beräknas från diagonalen av $(\mathbf{R}^T\mathbf{R})^{-1}$, där $\mathbf{R}$ kommer från QR-faktoriseringen.