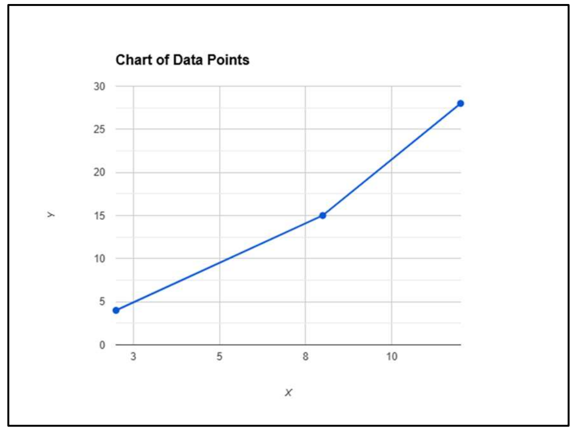
LASSO regression
(**L**east **A**bsolute **S**hrinkage and **S**election **O**perator):

- variation of **linear regression** with **penalty term** to enforce **sparsity** in the model
- **sparsity** (coefficients are **exactly** or **close to zero** = reducing the risk of **overfitting**)

Input data:

i	**x_i**	**$y_{i(target)}$**
0	2.5	14
1	3.5	18
2	8	22

q = 10
beta1 = 2



Sum of squared Residuals (SSR):

$$SSR = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1}$$

Mean squared error (MSE):

$$MSE = \frac{1}{n} SSR = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2}$$

Linear equation:

$$\hat{y}_i = \beta_1 . x_{ij} + \beta_0 \tag{3}$$

Combine equations = cost function:

- in Lasso regression, the **L1**-norm penalty is applied to all coefficients, except the intercept **β_0 !!!**
- where **λ** is a **damping factor**

$$J(\beta_0, \beta_1) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i - \left(\beta_1 . x_{ij} + \beta_0 \right) \right]^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{4}$$

Find global minimum of the function:

$$\frac{\partial J}{\partial \beta_0} = 0 \tag{5}$$

$$\frac{\partial J}{\partial \beta_1} = 0$$

- derivatives according to the coefficient **β_0**, regularization term L1 cannot be applied on interception
- we isolate **β_0**

$$\frac{\partial J}{\partial \beta_0} = \frac{2}{n}\sum_{i=1}^{n}(y_i - \beta_1.x_{i0} - \beta_0).(-1)$$

$$-\frac{2}{n}\sum_{i=1}^{n}(y_i - \beta_1.x_{i0} - \beta_0) = 0 \rightarrow \sum_{i=1}^{n}(y_i - \beta_1.x_{i0} - \beta_0) = 0$$

$$\sum_{i=1}^{n}(y_i - \beta_1.x_{i0}) - \beta_0.n = 0 \tag{6}$$

$$\beta_0 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_1.x_{i0})$$

- derivatives according to the coefficient **β_1**

$$\frac{\partial J}{\partial \beta_1} = \frac{2}{n}\sum_{i=1}^{n}(y_i - \beta_1.x_{i1} - \beta_0).(-x_{i1}) + \frac{\boldsymbol{\partial(\lambda|\beta_1|)}}{\boldsymbol{\partial\beta_1}} = 0$$

$$-\frac{2}{n}\sum_{i=1}^{n}x_{i1}(y_i - \beta_1.x_{i1} - \beta_0) + \lambda.sign(\beta_1) = 0 \tag{7}$$

$$\frac{2}{n}\sum_{i=1}^{n}x_{i1}(y_i - \beta_1.x_{i1} - \beta_0) = \lambda.sign(\beta_1)$$

- term in a bracket represents actually residuals:

$$r_i = y_i - \beta_1.x_{i1} - \beta_0$$

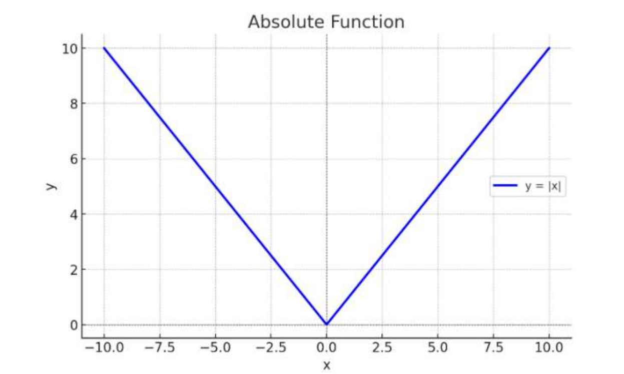
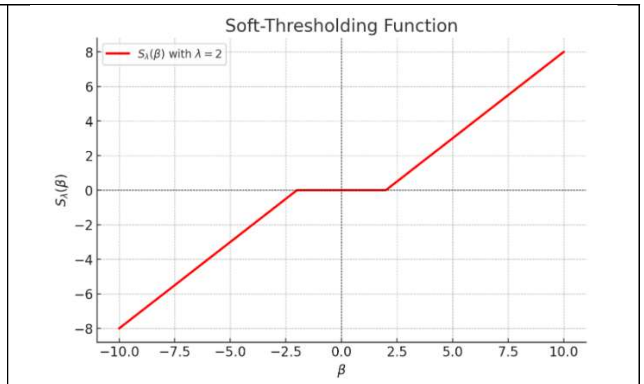
$$\frac{1}{n}\sum_{i=1}^{n}x_{i1}.r_i = \frac{\lambda}{2}.sign(\beta_1) \tag{8}$$

- because the the lambda is a shrinkage factor we can skip the value 2 and simply the equation:

$$\frac{1}{n}\sum_{i=1}^{n}x_{i1}.r_i = \lambda.sign(\beta_1) \tag{9}$$

Soft thresholding function:

- behavior of an absolute function => **derivative of the absolute function is not possible!!!**
- we need to apply soft thresholding function to overcome the fact we cannot derive absolute function

	
Absolute function	Soft-thresholding function

- soft thresholding function expression:
    - signum gives values (+1 or -1)
    - max chooses the max values between the **computed value** or **zero**

$$S_\lambda(z) = sign(z).max(|z| - \lambda, 0) \tag{9}$$

- final equation:

$$\beta_1 = S_\lambda(z)$$

$$\beta_1 = sign(z).max(|z| - \lambda, 0) \tag{8}$$

$$\beta_1 = sign\left(\frac{1}{n}\sum_{i=1}^{n} x_{i1}.r_i\right).max\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_{i1}.r_i\right| - \lambda, 0\right)$$

Conclusion-how to interpret the equations:

- if I have high coefficients or high input values (could be outliers) this cause low residuals
- if I have low residuals or lower than penalizing factor then the coefficient is decreased or set to zero if residuals are even lower than penalizing factor lambda