# Exercise 1
# Advanced Statistical Learning
# Summer semester 2025

A) (1 point) We want to work with a regression data set of size n and the quadratic
loss function
L (y, f (xlθ)) = (y–f (xlθ))2
.
First we consider a very simple constant model that does not take into account any
features in the data and outputs constant predictions.Write down the explicit form of the constant model f (xlθ) (i.e., in terms of coeffi-
cients and features).

A> Constant model $f(x|\theta)$ on the explicit form will be : $\boxed{f(x|\theta) = \theta}$ because it's constant and doesn't have any features, Therefore the intercept $\theta$ is the constant value of the model.

B) (2 points) Show that for the constant model, the optimal constant that optimizes
the empirical risk
1
n
n
i=1
L y(i), f (x(i)lθ)
induced by the quadratic loss function is the arithmetic mean of the responses.
Remember to show if it is a minimum or a maximum.

A> We have the empirical risk as average loss $\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - f(x^{(i)}|\theta))^2$ as $f(x^{(i)}|\theta)$ is constant, then $Risk(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)}-\theta)^2$

1) We do the $1^{st}$ derivative with respect to the model parameter $\theta$.

$$\frac{dR(\theta)}{d\theta} = \frac{d}{d\theta}\left(\frac{1}{n}\sum_{i=1}^{n}(y^{(i)}-\theta)^2\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{d}{d\theta}\left((y^{(i)}-\theta)^2\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{d}{d\theta}\left(y^{(i)2} - 2y^{(i)}\theta + \theta^2\right)$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}-2(y^{(i)}-\theta) = -\frac{2}{n}\sum_{i=1}^{n}(y^{(i)}-\theta)$$

2) Finding the critical point, $\frac{dR(\theta)}{d\theta} = 0$.

$$-\frac{2}{n}\sum_{i=1}^{n}(y^{(i)}-\theta)=0 \Rightarrow \sum_{i=1}^{n}(y^{(i)}-\theta)=0 \Rightarrow \sum_{i=1}^{n}y^{(i)}-n\theta=0 \Rightarrow n\theta=\sum_{i=1}^{n}y^{(i)}$$

$$\Rightarrow \boxed{\theta=\frac{1}{n}\sum_{i=1}^{n}y^{(i)}}$$

We arrived to the arithmetic mean of the responses (outputs = y).

3) To find minimum or maximum we need to calculate the second derivative of $R(\theta)$.

$$\frac{d}{d\theta}\left(-\frac{2}{n}\sum_{i=1}^{n}(y^{(i)}-\theta)\right)=-\frac{2}{n}\sum_{i=1}^{n}\frac{d}{d\theta}\left(y^{(i)}-\theta\right)=-\frac{2}{n}\sum_{i=1}^{n}-1 \Rightarrow 2>0.$$

As the second derivative is positive, the critical point is a minimum.

c) (1 point) Consider a data set with response vector y= {10, 26, 14, 7, 8}. Determine the optimal constant as derived in b). Construct a table that depicts the loss value for every observation when using this optimal constant. Then, visualize the loss function as a function of y with the optimal constant inserted. In your plot, depict the values from your table and highlight the optimal constant appropriately.

A) $y=\langle 10,26,14,7,8\rangle$

1) The optimal constant is the arithmetic mean of the responses.

$$\theta=\frac{1}{n}\sum_{i=1}^{n}y^{(i)}=\frac{1}{5}(10+26+14+7+8)=\frac{65}{5}=13$$

2) Using $\theta=13$ we are going to calculate the loss function for each response.

$(y^{(i)}-\theta)^2=L(y^{(i)},\theta)$

| $y^{(i)}$ | 10 | 26 | 14 | 7 | 8 |
|---|---|---|---|---|---|
| $(y^{(i)}-\theta)^2$ | 9 | 169 | 1 | 36 | 25 |

$L(y^{(i)},\theta)=L(y^{(i)},13)=\langle 9,169,1,36,25\rangle$

D) (2 points) Analternativetothequadraticlossfunctionistheabsolutelossfunction
L (y, f (x|θ)) = y(i) −f (x(i)|θ).
We consider a constant model again. The constant that then minimizes the empirical risk induced by the absolute loss function is the median of the response.
Repeat c) for the same data set, but with the absolute loss function and the corresponding optimal constant.
Which of the two loss functions would we use if we wanted to limit the influence of extreme values of the response on the resulting constant model? Give a reason for your answer.

A) 1) First of all we search for the median of response vector: $y = \langle 7, 8, 10, 14, 26 \rangle$

$\theta^* = 10$ (the median)

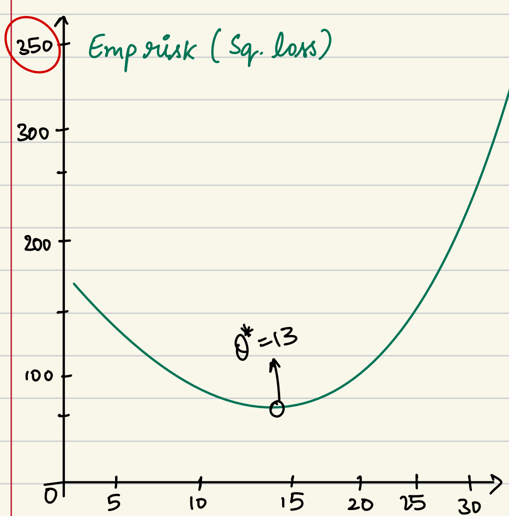
2) Let's calculate all losses with the absolute loss function,

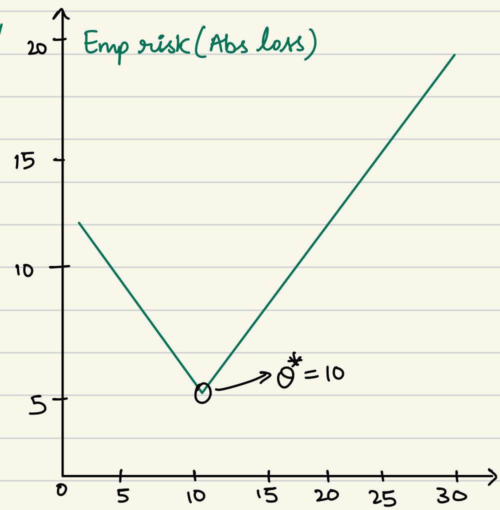
$L(y, f(x|\theta)) = |y^{(i)} - f(x^{(i)}|\theta)| = |y^{(i)} - \theta^*|$

| $y^{(i)}$ | 10 | 26 | 14 | 7 | 8 |
|---|---|---|---|---|---|
| $\lvert y^{(i)} - \theta^* \rvert$ | 0 | 16 | 4 | 3 | 2 |

$L(y^{(i)}, \theta^*) = L(y^{(i)}, 10) = \langle 0, 16, 4, 3, 2 \rangle$

If we want to limit the influences of extreme values, such as outliers, we will use this second loss function, the absolute loss. The reason why the quadratic loss function is more sensitive to these extreme values is because the squaring increases the weight of these particular responses, amplifying the error. However, this is a reason that can be easily seen theoretically, if an error is smaller than 1, $0 \leq$ error $< 1$ the quadratic loss will minimize the value. Otherwise the value will grow quadratically. On the other hand, the absolute loss function grows with a linear error.



If we compare both plots, there's a huge difference for when the values become extreme. The average loss function for squared loss arrives at 350 whereas for absolute loss arrives at 20.

Every point of the plot corresponds to the $\theta$ value vs empiric risk function.

$\theta = 13$ (optimal)

$$R(\theta) = R(13) = \frac{1}{N} \sum_{i=1}^{n} (y^{(i)} - 13)^2 = \frac{240}{5} = 48.$$ (13,48) in 1st plot.

$\theta = 10$ (optimal)

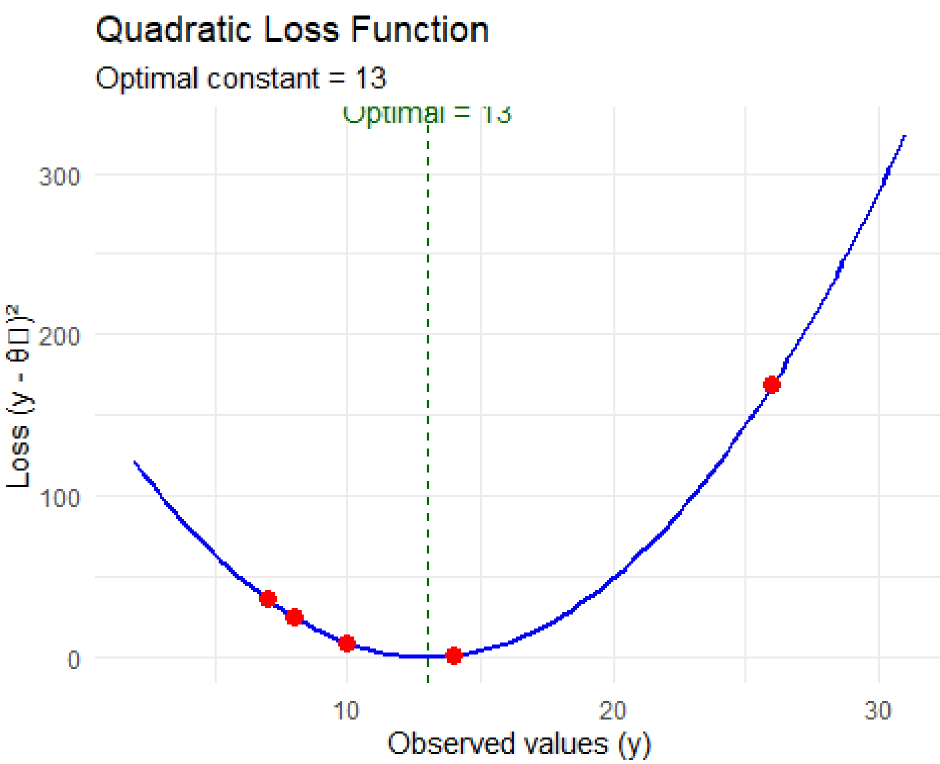
$$R(\theta) = R(10) = \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - 10)^2 = \frac{25}{5} = 5.$$ (10,5) in 2nd plot

# Exercises C and D

2025-04-15

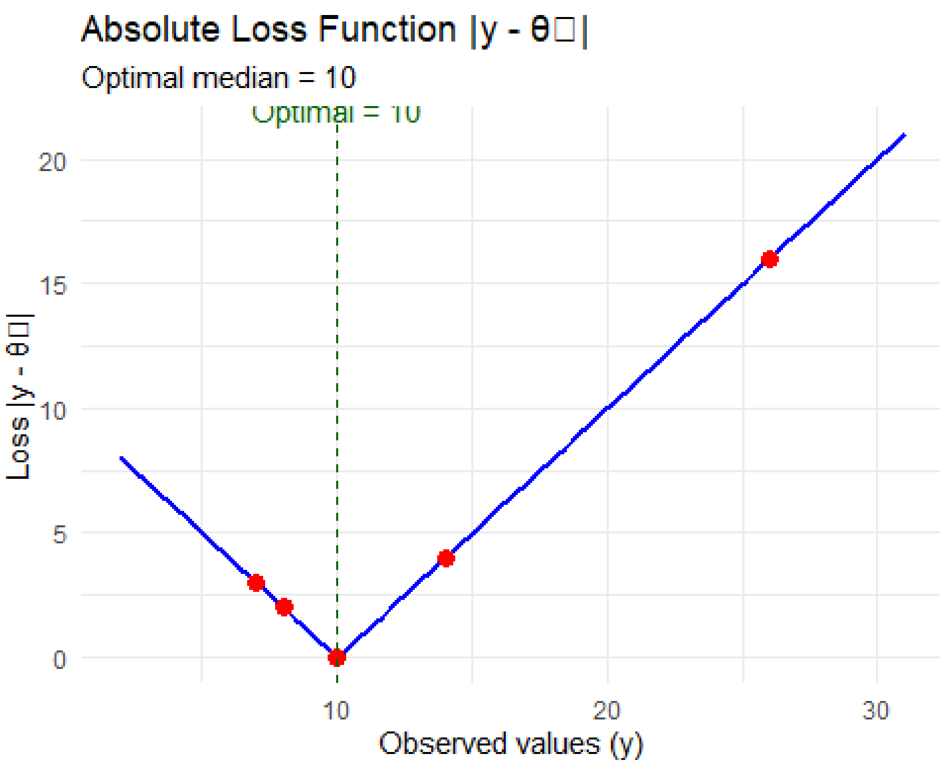
Square

```
## Optimal constant (mean): 13

## [1] "Loss table:"

##    y Loss
## 1 10    9
## 2 26  169
## 3 14    1
## 4  7   36
## 5  8   25
```




4

```
## Optimal constant (median): 10

## [1] "Absolute loss table:"

##    y Loss
## 1 10    0
## 2 26   16
## 3 14    4
## 4  7    3
## 5  8    2
```


Absolute Loss Function |y - θ□|
Optimal median = 10
Optimal = 10
Loss |y - θ□|
20
15
10
5
0
10
20
30
Observed values (y)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
y = np.array([10,26,14,7,8])
```

## c) Using the Quadratic Loss Function

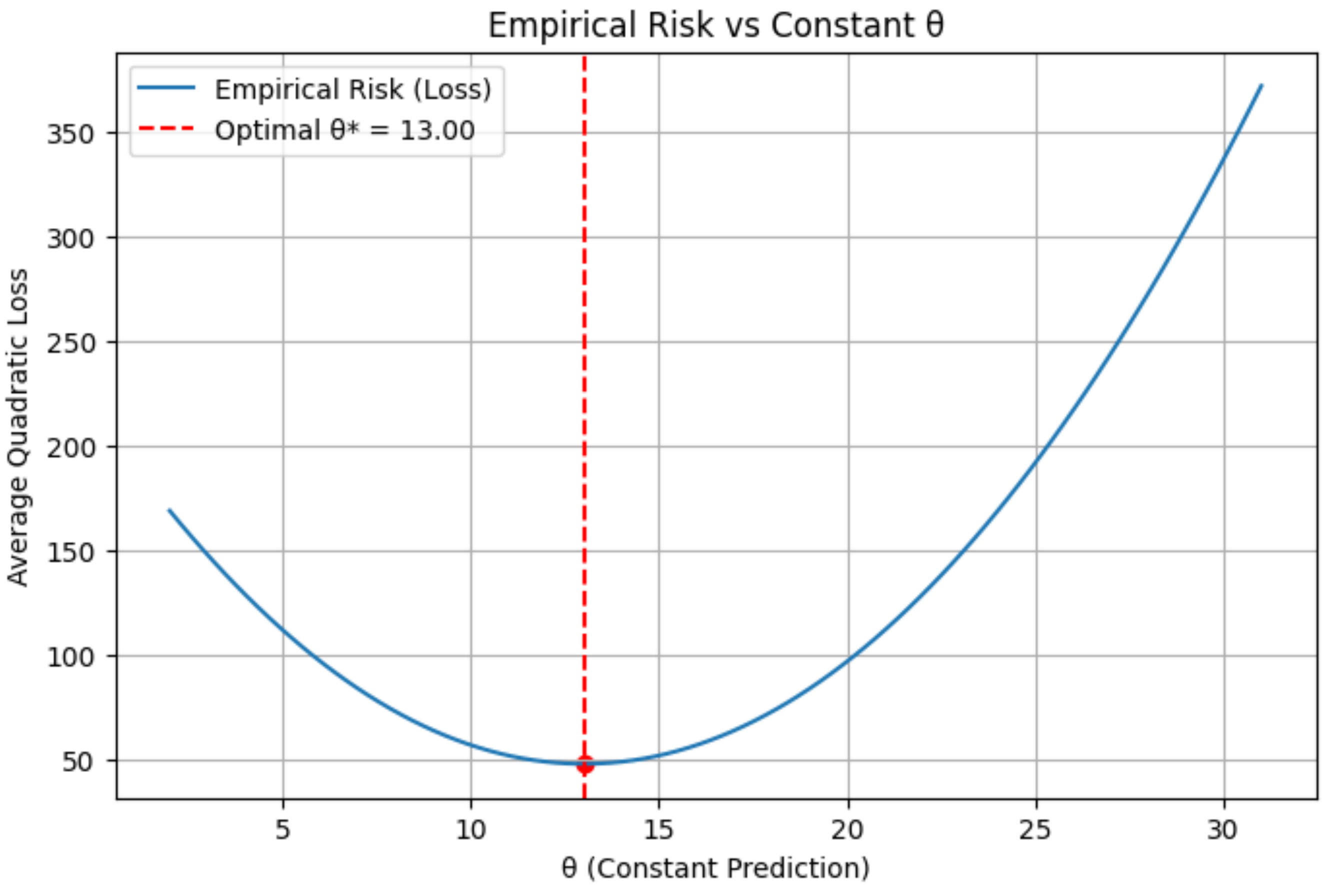
```
opt_const = np.mean(y)
print('The optimal constant value applying the arithmetic mean of the responses
```

```
The optimal constant value applying the arithmetic mean of the responses is
```

```
losses_array = (y-opt_const)**2

table = pd.DataFrame({
    'y_i': y,
    'Loss (y_i - theta*)^2': losses_array
})

print('Table of the y_i and each loss value:\n')
print(table)
```

```
Table of the y_i and each loss value:

   y_i  Loss (y_i - theta*)^2
0   10                    9.0
1   26                  169.0
2   14                    1.0
3    7                   36.0
4    8                   25.0
```

```
theta_vals = np.linspace(min(y)-5, max(y)+5, 100)
total_loss = [np.mean((y - theta) ** 2) for theta in theta_vals]

plt.figure(figsize=(8, 5))
plt.plot(theta_vals, total_loss, label='Empirical Risk (Loss)')
plt.axvline(opt_const, color='red', linestyle='--', label=f'Optimal θ* = {opt_co
plt.scatter([opt_const], [np.mean(losses_array)], color='red',marker='o', linewi
plt.title('Empirical Risk vs Constant θ')
plt.xlabel('θ (Constant Prediction)')
plt.ylabel('Average Quadratic Loss')
plt.legend()
plt.grid(True)
plt.show()
```



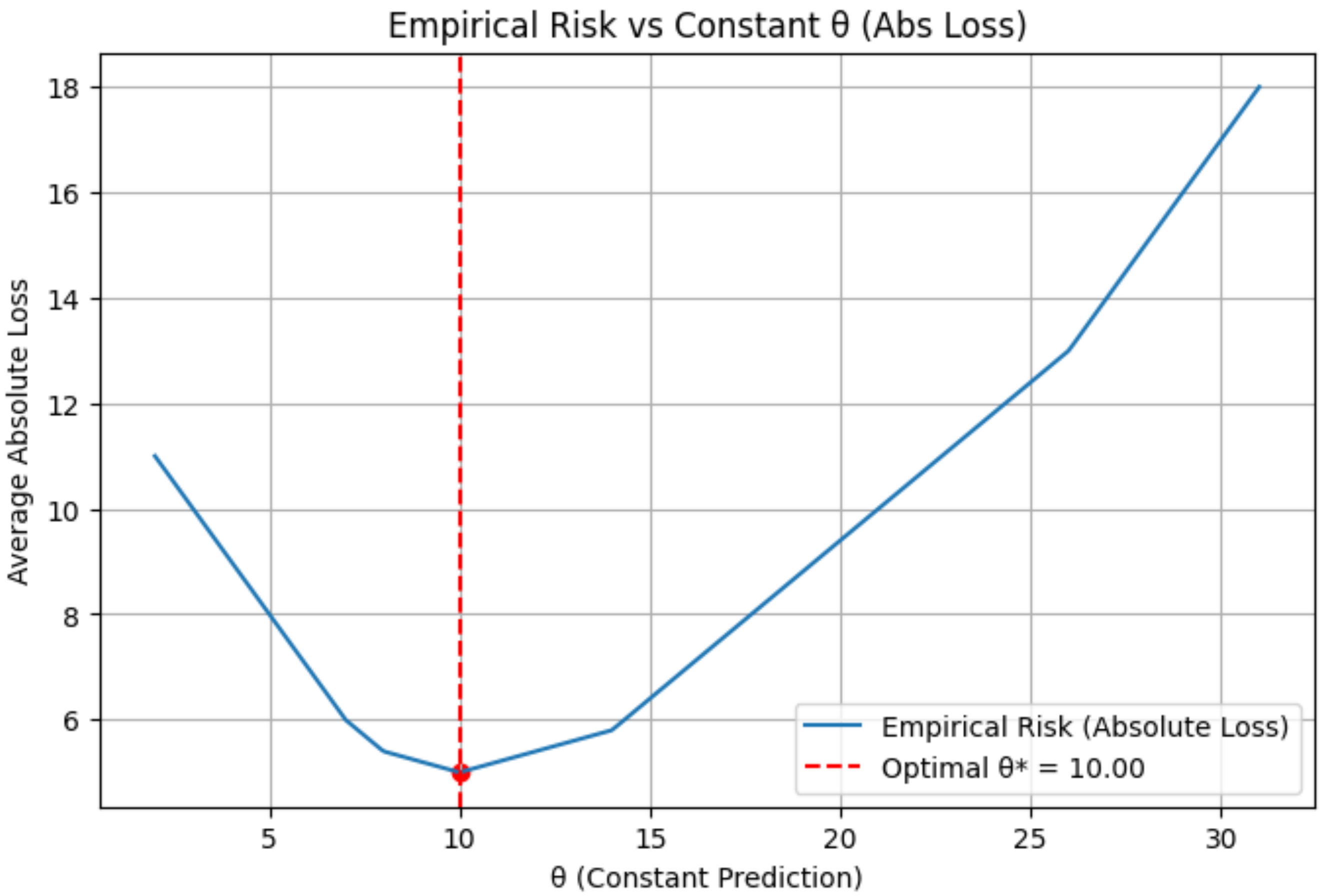

```
opt_const_abs = np.median(y)
print('The optimal constant value applying the median of the responses is: ', c
```

```
The optimal constant value applying the median of the responses is:  10.0
```

```
losses_array_abs = abs(y-opt_const_abs)

table = pd.DataFrame({
    'y_i': y,
    'Loss abs(y_i - theta*)': losses_array_abs
})

print('Table of the y_i and each loss value:\n')
print(table)
```

```
Table of the y_i and each loss value:

   y_i  Loss abs(y_i - theta*)
0   10                     0.0
1   26                    16.0
2   14                     4.0
3    7                     3.0
4    8                     2.0
```

```
theta_vals = np.linspace(min(y)-5, max(y)+5, 400)
total_loss = [np.mean(abs(y - theta)) for theta in theta_vals]

plt.figure(figsize=(8, 5))
plt.plot(theta_vals, total_loss, label='Empirical Risk (Absolute Loss)')
plt.axvline(opt_const_abs, color='red', linestyle='--', label=f'Optimal θ* = {c
plt.scatter([opt_const_abs], [np.mean(losses_array_abs)], color='red',marker='c
plt.title('Empirical Risk vs Constant θ (Abs Loss)')
plt.xlabel('θ (Constant Prediction)')
plt.ylabel('Average Absolute Loss')
plt.legend()
plt.grid(True)
plt.show()
```



E) Now we want to modify the model from a) and b) by adding four coefficients describing the effects of four features x1, x2, x3 and x4. What is the formal representation (hypothesis space) for this problem?

A> We are now modifying the model $f(x|\theta) = \theta$ by adding 4 features, $x = \{x_1, x_2, x_3, x_4\}^T \in \mathbb{R}^4$ where the linear regression model is:

$$f(x|\theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$\theta_0$ : intercept

The hypothesis space is: $\mathcal{H} = \{ f(x|\theta) = \theta^T \tilde{x} \mid \theta \in \mathbb{R}^5 \}$

where: $\tilde{x} = (1, x_1, x_2, x_3, x_4) \in \mathbb{R}^5$

$\theta^T = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)^T \in \mathbb{R}^5$

F) We want to use the quadratic loss function for the problem from e) again. No numerical optimization is needed to estimate the coefficient vector. Derivetheestimatorforθofthelinearregressionmodelyouspecifiedine). Proceed as follows: Write down the quadratic loss of the whole training data set (the RSS or SSE) and minimize it for θ.
Use the design matrix notation with included intercept (Slides 0: Notation and Definitions, page 6, right side). You may assume that the design matrix has full column rank. You may argue why the optimum is a minimum using the second order derivative, you don't have to prove it from scratch

A> We are now derivating the estimator for $\theta$ of the linear regression model from E).
We have $\tilde{x} = (1, x_1, x_2, x_3, x_4)^T$ and $\theta = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)^T$. The estimator is derived:

$$y = \tilde{x}\theta$$

$$L(\theta) = \|y - \tilde{x}\theta\|^2 = (y - \tilde{x}\theta)^T (y - \tilde{x}\theta)$$

$$RSS(\theta) = \sum_{i=1}^{n} (y^{(i)} - x^{(i)T}\theta^2)$$

To minimize the loss, we need to find the optimal $\hat{\theta}$ by minimizing: $\min_\theta \|y - \tilde{x}\theta\|^2$
calculating the $\nabla$ (gradient) with respect to $\theta$:

$$\nabla_\theta RSS(\theta) = \sum_{i=1}^{n} \underbrace{-2x^{(i)T}(y^{(i)} - x^{(i)T}\theta)}_{\text{vector of length 5}} = -2\tilde{x}^T(y - \tilde{x}\theta)$$

We set it to 0:

$$\nabla_\theta RSS(\theta) = 0 \Leftrightarrow -2\tilde{x}^T(y - \tilde{x}\theta) = 0 \Leftrightarrow \tilde{x}^T(y - \tilde{x}\theta) = 0 \Leftrightarrow \tilde{x}^T y = \tilde{x}^T\tilde{x}\theta \Leftrightarrow \boxed{\hat{\theta} = (\tilde{x}^T\tilde{x})^{-1} \tilde{x}^T y}$$

(G) Suppose that we are interested in the performance / accuracy / generalization error of the estimated model from f). Is it enough to evaluate the error rate of the model on the training dataset (i.e. training error rate)? Why/why not? Describe (briefly) in general how we would evaluate the model fitted in f) using concepts that you learned from the lecture

A> No, evaluating the performance/accuracy/generalization error of a model based on error rate is not enough as it shows how well the model fits on a dataset which it is already familiar with. We won't be able to predict its true accuracy or performance. To obtain so, we can divide the dataset into 3 parts namely: training set, validation set and test set. The model is first trained on the training set and then evaluated on validation set to tune and find the best configurations. After finalizing, it is then tested on the test set to find an unbiased estimate of the models performance.