# Contents

# 1 ./covid19.py

```python

from fit_linear_reggression import fit_linear_reggression


import numpy as np
from copy import deepcopy
import pandas as pd
from plotnine import *
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D


if __name__ == "__main__":
        _path = "covid19_israel.csv"
        dataset_covid = pd.read_csv(_path)
        dataset_covid["log_detected"] = np.log( dataset_covid["detected"])
        X, Y = dataset_covid["day_num"], dataset_covid["log_detected"]
        X = np.array([X])
        W, S = fit_linear_reggression( X.transpose(), Y )
        gplot = ggplot(dataset_covid, aes(x='day_num' , y='log_detected')) +\
        geom_point()+ geom_abline(slope =W[0]) + ggtitle('log detected as function of the dayes')

        ggsave(gplot, "log_fig.png")
        X, Y = dataset_covid["day_num"], dataset_covid["detected"]
        Z = np.e ** (W[0]*X)

        gplot = ggplot( dataset_covid , aes(x='day_num' , y='detected')) +\
        geom_point()+\
        geom_line(aes(y='Z')) + geom_abline(slope =W[0]) + ggtitle('detected as function of the dayes')
        ggsave(gplot, "fig.png")
```

# תרגיל 2

29 באפריל 2020

דויד פונרובסקי 208504050

## 1 תרגיל 1:

נדרש להוכיח ש $\ker\left(X^T\right) = \ker\left(XX^T\right)$. קל ליראות ש $\ker\left(X^T\right) \subset \ker\left(XX^T\right)$ נוכיח את הכיוון השני. נתבונן ב$u$ כך ש $u \in \ker\left(XX^T\right)$. מכאן

$$XX^Tu = 0 \Rightarrow u^TXX^Tu = \left(X^Tu\right)^T X^Tu = \left\langle X^Tu, X^Tu\right\rangle = 0$$
$$\Rightarrow X^Tu = 0 \Rightarrow u \in \ker\left(X^T\right) \Rightarrow \ker\left(X^T\right) = \ker\left(XX^T\right)$$

## 2 תרגיל 2:

צ"ל $Im\left(A^T\right) = (\ker A)^\perp$. כיוון ראשון, ניתבונן ב $b \in Im\left(A^T\right)$ וב $w \in \ker(A)$

$$b \in Im\left(A^T\right) \Rightarrow \exists x : A^Tx = b \Rightarrow \langle b, w\rangle = \left\langle A^Tx, w\right\rangle =$$
$$= \langle x, Aw\rangle = 0 \Rightarrow Im\left(A^T\right) \subset (\ker A)^\perp$$

כיוון שני $x \in (\ker A)^\perp$ מכאן שלכל $Av = AA^Tv = \vec{0} \Rightarrow \langle x, v\rangle = 0$. ניתבונן בבסיס $B = \{b_i\} \bigcup \{b_i'\}$ כך ש- $b_i' \in \ker A$. את $x$ ניתן להציג ע"י $x = \sum \mu_i b_i + \sum 0 \cdot b_i'$ נסמן את וקטור המקדמים ב $\mu$. וניתן לרשום $x = \mu b^T$. מאחר ו $B$ הוא בסיס אז

$$span\{b_i\} \bigcap span\{b_i'\} = \emptyset \Rightarrow span\{b_i\} \bigcap \ker(A) = \emptyset$$
$$\Rightarrow \mu b^T \notin \ker A \overset{(1)}{\Rightarrow} \mu b^T \notin \ker A^TA$$

את $(1)$ קיבלנו מהסעיף הקודם. מכאן נקבל שמעל המרחב $span(B_2) = span(\{b_i\})$, המטריצה $\left[A^TA\right]_{B_2} : span(B_2) \to Im(A)$ היא חח"ע והפיכה (ריבועית עם גרעין ריק מעל התת מרחב). ולכן

$$x = \mu b^T = \left[A^TA\right]_{B_2}\left[\left(A^TA\right)\right]_{B_2}^{-1}(\mu) = \left(A^TA \cdot P_{B_2}\right)\left[\left(A^TA\right)\right]_{B_2}^{-1}(\mu)$$

כאשר $P_{B_2}$ היא מטריצת המעבר אל בסיס $B_2$. מצאנו וקטור $y$ כך ש $A^Ty = x$. $\Leftarrow x \in Im\left(A^T\right)$.

## 3 תרגיל 3:

$X$ אינה הפיכה, צ"ל ל $X^Tw = y$ יש אינסוף פתרונות אמ"מ $y \perp \ker(X)$. הוכחה :
ראשית מהסעיף הקודם אנו יודעים כי

$$Im\left(X^T\right) = (\ker X)^\perp \Rightarrow y \in Im\left(X^T\right) \Leftrightarrow y_\perp \ker X$$

ולכן אנו יודעים כי קיים לפחות פתרון אחד $w$. נראה כי $w + v$ הוא גם פתרון לכל $v \in \ker X^T$ :

$$X^T(w+v) = X^Tw + X^Tv = X^Tw = y$$

מכאן שאם $X$ היא לא הפיכה אז גם $X^T$ לא הפיכה ולכן $\ker X^T is\ a\ subsapce$ כלומר קיימים אין סוף $v$ים השייכים ל $\ker X^T$ ולכן יש אינסוף פתרונות מהצורה $w + v$.
כיוון שני, נניח שיש אין סוף פתרונות מהצורה $X^Tw = y$ ונתבונן ב $v$ כך ש $Xv = 0$ מכאן

$$\langle y, v\rangle = \left\langle X^Tw, v\right\rangle = \langle w, Xv\rangle = 0$$

כמובן שזה נכון לכל $v \in \ker X$ מכאן ש $y \in (\ker X)^\perp$.

## 4 תרגיל 4:

צ"ל להראות כי $XX^T\omega = Xy$ יש אינסוף פתרונות או פיתרון יחיד. נראה כי $Xy \perp \ker\left(XX^T\right)$. לפי סעיפים קודמים $\ker\left(XX^T\right) = \ker\left(X^T\right)$ ולכן מספיק להראות כי $Xy \perp \ker\left(X^T\right)$, אם $y \notin \ker X$ אז

$$Xy \in Im\left(X\right) = \left(\ker X^T\right)^{\perp} \Rightarrow Xy \perp \ker\left(X^T\right)$$

מקרה שני, אם $y \in \ker X$ אז כל פתרון ב $\omega \in \ker X^T$ יפתור, וכמובן שאם $\ker X \neq \emptyset$ אז $X$ לא הפיכה מכאן ש $X^T$ גם לא הפיכה ולכן גם $\ker X^T$ אינו ריק. כלומר ניתן למצוא $\omega \in \ker X^T$ במקרה זה.

## 5 תרגיל 5:

### 5.1 $(a)$

נראה כי $P = \sum v_i v_i^T$ סימטרית: $P^T = \left(\sum v_i v_i^T\right)^T = \sum\left(v_i v_i^T\right)^T = \sum v_i v_i^T = P$.

### 5.2 $(b)$

נסמן ב $\{e_i\}$ וקטורים שנמצאים בבסיס ל $\mathbb{R}^d/V$. כמובן ש $span\left(\{e_i\}\bigcup\{v_i\}\right) = \mathbb{R}^d$, ניתבנן ב $u \in \mathbb{R}^d$ ונפרוש אותו $u = \sum \alpha_i v_i + \sum \beta_i e_i$

$$\begin{aligned}
Pu &= \sum_i v_i v_i^T\left(\sum_j \alpha_j v_j + \sum_j \beta_j e_j\right) = \lambda u \\
&\Rightarrow v_i v_i^T\left(\sum_j \alpha_j v_j + \sum_j \beta_j e_j\right) = \lambda \alpha_i v_i \\
&= \sum_j \alpha_j v_i\left(v_i^T v_j\right) + v_i v_i^T \sum_j \beta_j e_j = \sum_j \alpha_j v_i \left\langle v_i, v_j\right\rangle + v_i v_i^T \sum_j \beta_j e_j = \\
&= \alpha_i v_i + v_i v_i^T \sum_j \beta_j e_j = \lambda \alpha_i v_i
\end{aligned}$$

כאשר המעבר האחרון תקף כי לבסיס אורטונורמלי מתקיים $\left\langle v_i, v_j\right\rangle = \delta_{i,j}$. ולכן נידרוש כי $v_i v_i^T \sum_j \beta_j e_j = 0$ אבל $e_j$ הוא בסיס בת"ל ולכן בהכרח $\beta_j = 0$ וכמובן $\lambda = 1$.
אם אנו מסתכלים על קורדינטה המתאימה ל $e_i$ כלומר :

$$\left(Pu\right)_k = \left(\sum_i v_i v_i^T\left(\sum_j \alpha_j v_j + \sum_j \beta_j e_j\right)\right)_{(k)} = \lambda \beta_k e_k$$

אז מאחר והקורדינטה ה$k$ית (יצוג ב $\{e_i\}\bigcup\{v_i\}$) של $P$ היא $0$ נקבל כי $0 = \lambda \beta_k e_k$ כלומר כך או כך $\beta_k = 0$.

### 5.3 $(c)$

צ"ל לכל $u \in V$ מתקיים $Pu = u$. מאחר ו $u \in V$ ניתן לפרוש אותו בהמצאות $\{v_j\}$ בלבד, ולכן נקבל כי

$$Pu = P\left(\sum \alpha_i v_i\right) = \sum \lambda_i \alpha_i v_i = \sum \alpha_i v_i = u$$

### 5.4 $(d)$

צ"ל $P^2 = P$. נחשב :

$$\begin{aligned}
P^2 &= \sum_{i,j} v_i v_i^T v_j v_j^T = \sum_{i,j} v_i \left\langle v_i, v_j\right\rangle v_j^T = \sum_{i,j} v_i \delta_{i,j} v_j^T = \\
&= \sum_i v_i v_i^T = P
\end{aligned}$$

### 5.5 $(e)$

ישירות מהסעיף הקודם :

$$(\mathbb{I}-P)\,P = P - P^2 = P - P = 0$$

## 6 תרגיל 6:

צ"ל $\left(XX^T\right)^{-1} = U\left(\Sigma\Sigma^T\right)^{-1}U^T$ נחשב:

$$\begin{aligned}\left(XX^T\right)^{-1} &= \left(\left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T\right)^{-1} = \left(U\Sigma V^T V\Sigma^T U^T\right)^{-1} = \\ &= \left(U\Sigma\Sigma^T U^T\right)^{-1} = \left(\left(U^T\right)^{-1}\left(\Sigma\Sigma^T\right)^{-1}U^{-1}\right) = U\left(\Sigma\Sigma^T\right)^{-1}U^T\end{aligned}$$

כאשר השתמשתי בכך ש $UU^T = VV^T = \mathbb{I}$ (אורטונורמליות). מהנחה כי $XX^T$ הפיכה, כל הע"ע שלה שונים מ0 ולכן $\Sigma^{-1} = \Sigma^{\dagger}$ מוגדר. מכאן נקבל כי

$$\begin{aligned}\left(XX^T\right)^{-1}X &= U\left(\Sigma\Sigma^T\right)^{-1}U^T U\Sigma V^T = U\Sigma^{-1}\Sigma^{-1}\Sigma V^T = \\ U\Sigma^{-1}V^T &= \left(V\Sigma^{-1}U^T\right)^T = \left(X^{\dagger}\right)^T\end{aligned}$$

## 7 תרגיל 7:

צ"ל ש $XX^T$ הפיכה אמ"מ $span\left\{x_1,..,x_m\right\} = \mathbb{R}^d$. כיוון ראשון, $x_1,..,x_m$ הם העמודות של $X$ ולכן אם

$$span\left\{x_1,..,x_m\right\} = \mathbb{R}^d \Leftrightarrow Im\left(X^T\right) = \mathbb{R}^d \Leftrightarrow \ker\left(X^T\right) = \emptyset$$

אבל בסעיפים קודמים הראנו כי $\ker\left(X^T\right) = \ker\left(XX^T\right)$ ולכן

$$XX^T\, is\, invertible \Leftrightarrow \ker\left(XX^T\right) = \emptyset \Leftrightarrow span\left\{x_1,..,x_m\right\} = \mathbb{R}^d$$

## 8 תרגיל 8:

קודם כל נראה כי $\hat{w} = X^{T\dagger}y$ הוא פיתרון.

$$XX^T\hat{w} = XX^TX^{T\dagger}y = U\Sigma V^T V\Sigma^T U^T\left(U\Sigma^{\dagger}V^T\right)y = U\Sigma V^T y = Xy$$

נשים לב שבבסיס הוע"ע של $XX^T$ יש ל$\hat{w}$ פתרון יחיד לכל קורדינטה ב$\hat{w}_i$. מכאן שלכל $\bar{w} = \hat{w} + \vec{\varepsilon}$ מתקיים כי $\vec{\varepsilon} \notin span\left(\hat{w}_i\right)$ ולכן

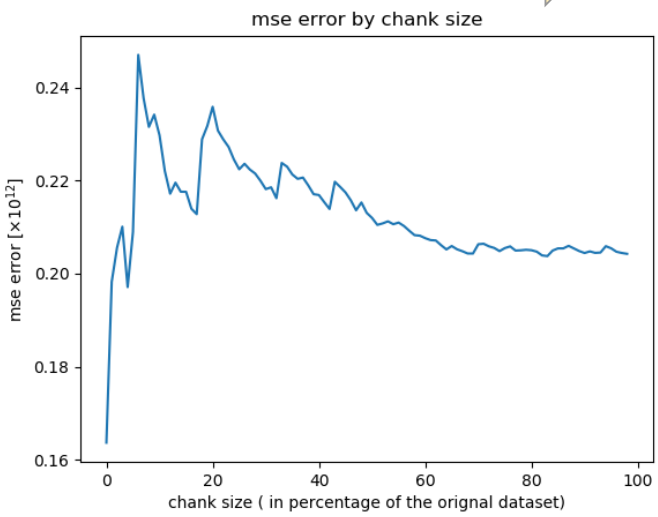
$$\begin{aligned}||\bar{w}|| = ||U^T\bar{w}|| = ||\left[\bar{w}\right]_U|| = ||\left[\hat{w}+\vec{\varepsilon}\right]_U|| = ||\left[\hat{w}\right]_U|| + ||\left[\vec{\varepsilon}\right]_U|| \geq \\ \geq ||\left[\hat{w}\right]_U|| = ||U^T\hat{w}|| = ||\hat{w}||\end{aligned}$$

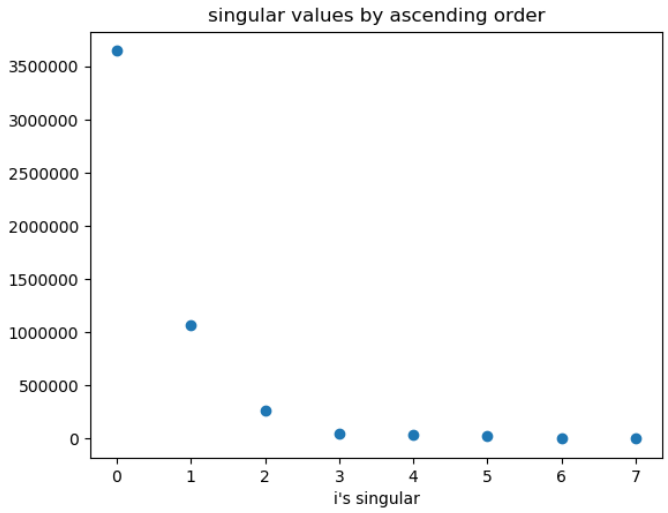
# 9 רגרסיה, מחירי הדירות.

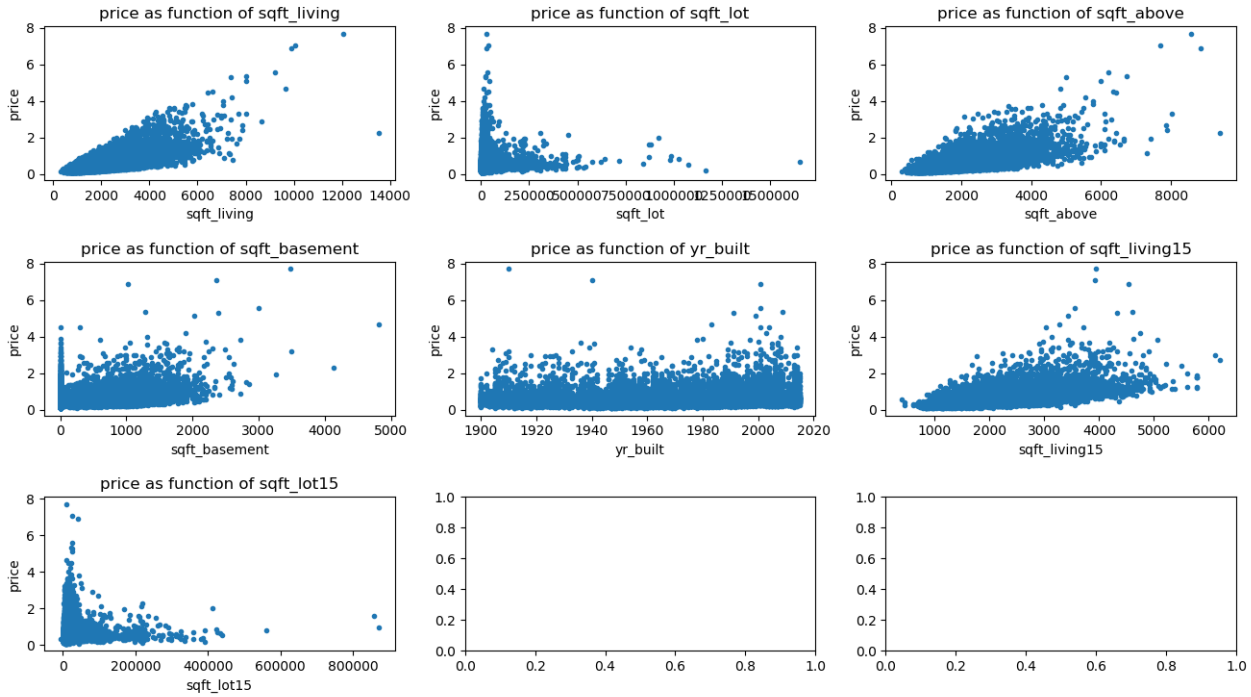
ערכים שהורדתי לחלוטין : $['date','zipcode','yr_renovated','price','lat','long']$ ערכים אלו לא הראו שום הגיון בבדיקת הקורלציה. קטגוריות : $['view','waterfront','bedrooms','grade','floors','condition','bathrooms']$.

ערכים אלו התחלקו לטווח ערכים סופי. בסך הכל הגעתי לטווח שגיאת $mse$ של $0.2 \times 10^{12}$ כלומר $std$ של בערך $\sim 0.2 \cdot 10^6$. כלומר שיערוך של שגיאה ממוצעת $\pm$ 200 אלף לדירה נחשב לדעתי לגיטמי לחלוטין (למרות שאולי בדולרים, זה כבר לא כל כך הגיוני). ניתן לראות כי יש 3 עע"מ מכרעים יחידים ( את כל השאר ניתן להזניח ) ואחד מהם הוא ההוזה $(bias)$.

6.1



מטריצות מקדמי המתאם :

$$p\left(price, sqft-living\right) = \begin{bmatrix} 2.5\cdot 10^{3} & 7.02\cdot 10^{-1} \\ 7.02\cdot 10^{-1} & 3.9\cdot 10^{-4} \end{bmatrix}$$

$$p\left(price, sqft-lot\right) = \begin{bmatrix} 1.1\cdot 10^{5} & 8.9\cdot 10^{-2} \\ 8.9\cdot 10^{-2} & 8.7\cdot 10^{-6} \end{bmatrix}$$

$$p\left(price, sqft-above\right) = \begin{bmatrix} 6.6\cdot 10^{-1} & 4.43\cdot 10^{-4} \\ 1.2\cdot 10^{3} & 3.2\cdot 10^{-1} \end{bmatrix}$$

$$p\left(price, sqft-basement\right) = \begin{bmatrix} 3.2\cdot 10^{-1} & 8.29\cdot 10^{-4} \\ 8.02\cdot 10 & 5.4\cdot 10^{-2} \end{bmatrix}$$

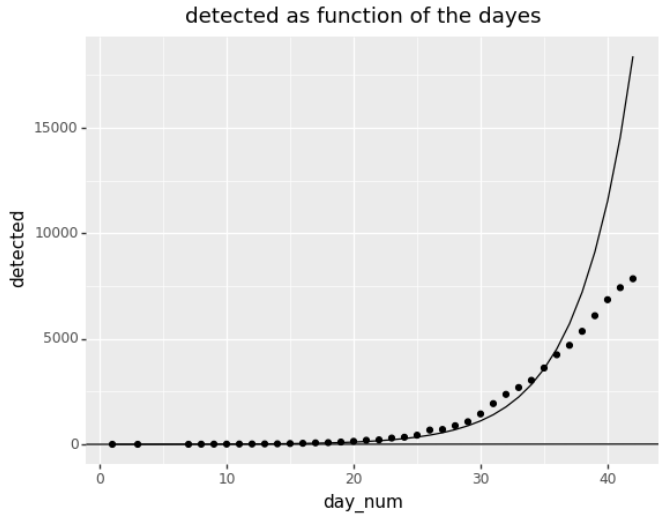
$$p\left(price, yr-built\right) = \begin{bmatrix} 5.4\cdot 10^{-2} & 1.25\cdot 10^{-2} \\ 7.02\cdot 10^{-1} & 3.9\cdot 10^{-4} \end{bmatrix}$$

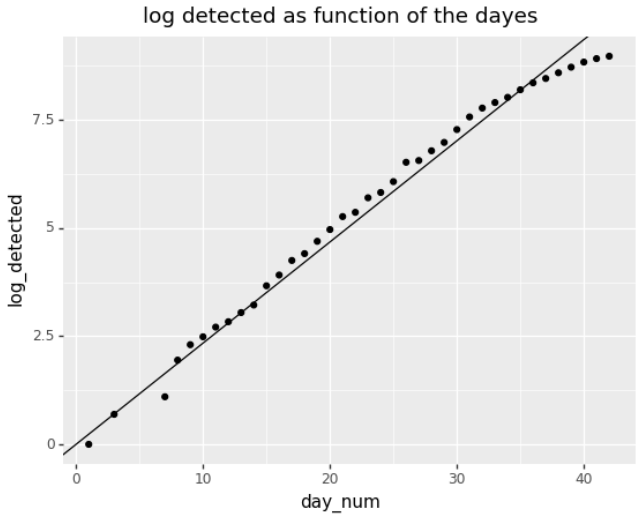
$$p\left(price, sqft-living15\right) = \begin{bmatrix} 2.5\cdot 10^{3} & 7.02\cdot 10^{-1} \\ 8.26\cdot 10^{-2} & 1.3\cdot 10^{-5} \end{bmatrix}$$

$$p\left(price, sqft-living15\right) = \begin{bmatrix} 7.43\cdot 10^{4} & 8.2\cdot 10^{-2} \\ 8.26\cdot 10^{-2} & 1.3\cdot 10^{-5} \end{bmatrix}$$

## 10 רגרסיה קורנה :

דומה לשאלה הקודמת.




## 11 תרגיל 10:

במקרה פונקציית המחיר היא $L_{exp} = \frac{1}{2}\left(e^{wX} - Y\right)^2$ ולכן נקבל כי

$$\nabla L_{exp} = X\left(e^{wX} - Y\right) = 0 \Rightarrow w = X^{\dagger T}\log Y$$

8.1

# 3 ./fit linear reggression.py

```python
import numpy as np
from copy import deepcopy
import pandas as pd
from plotnine import *
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D


def expand_bias(design_matrix):
    _shape = design_matrix.shape
    _ones = np.ones( (_shape[0], _shape[1] + 1) )
    _ones[:,:-1] = design_matrix
    return _ones

def fit_linear_reggression(design_matrix, respone_vec, bias=False):
    if bias:
        design_matrix = expand_bias( design_matrix )
        print(design_matrix)

    u, s, vh = np.linalg.svd(design_matrix)
    s_dagger = deepcopy( s )
    mask = s != 0
    s_dagger[ mask ] = 1 / s [ mask ]
    def diag_mul( _diag, _vec):
        return _diag * _vec[:_diag.shape[0]]
    t =  vh.transpose() @ diag_mul( s_dagger ,u.transpose() @ respone_vec )
    return t, s

def predict( _design_matrix, _coefficients, bias = False):
    if bias:
        _design_matrix = expand_bias( _design_matrix )
    return _design_matrix @ _coefficients

def mse(predicvec, responevec ):
    return np.linalg.norm(responevec - predicvec) ** 2 / responevec.shape[0]

def load_data(_path):

    def remove_end_cases(_frame):
        return _frame[  _frame['price'] > 0  ]

    _dataset_prepoc = remove_end_cases( pd.read_csv(_path) )
    target = 'price'
    prices = _dataset_prepoc[target]
    droped_fe = [ 'date', 'zipcode', 'yr_renovated', 'price', 'lat', 'long']
    categorical = ['view' , 'waterfront', 'bedrooms', 'grade', 'floors', 'condition', 'bathrooms']
    cat = pd.DataFrame ( { 'id' : _dataset_prepoc['id'] } , pd.get_dummies(_dataset_prepoc[categorical].astype('category') )
    _dataset_prepoc = _dataset_prepoc.drop([target , 'id'] + categorical + droped_fe, axis=1)
    #_dataset_prepoc = pd.merge( _dataset_prepoc , cat, left_on='id', right_on='id')
    return _dataset_prepoc, prices

def plot_singular_values( singulars ):
    singulars.sort()
    plt.scatter( list(range(len(singulars))), singulars[::-1])
    plt.title("singular values by ascending order")
    plt.xlabel("i's singular")
    plt.ylabel('value')
    plt.savefig("plot_singular_values.png")
    plt.show()
```

```python



def feature_evaluation():
    pass

def q1():

    design_matrix_frame, respone_vec_frame = load_data( "./kc_house_data.csv" )
    design_matrix , respone_vec = design_matrix_frame.to_numpy() , respone_vec_frame.to_numpy()

    respone_vec /= 10**6


    batchsize = int(len(respone_vec)/4)

    indices = np.random.randint(len(respone_vec), size=batchsize)
    W, S = fit_linear_reggression( design_matrix[:][indices], respone_vec[indices], bias=True )
    plot_singular_values( S )



    mask = np.ones(len(respone_vec) , bool)
    mask[indices] = False


    cases_design_matrix = design_matrix[:][mask]
    cases_prices = respone_vec[mask]
    precent = int(len(cases_prices) / 100)
    mses = []

    print(W)

    for i in range(1, 100 ):
        mses.append( mse(predict(design_matrix[:][:i*precent], W , bias=True), cases_prices[:i*precent]))

    print(mses)
    plt.plot( mses )
    plt.title("mse error by chank size")
    plt.xlabel("chank size ( in percentage of the orignal dataset)")
    plt.ylabel('mse error $ [\\times 10^{12}] $ ')
    plt.savefig("q1.png")
    plt.show()
    #respone_vec
    fig, ax = plt.subplots(3, 3)
    for i, _frame in enumerate(design_matrix_frame):
        ax[ int(i /3), i%3 ].scatter( design_matrix_frame[_frame], respone_vec, marker='.'  )
        ax[ int(i /3), i%3 ].set_title( 'price as function of {0}'.format(_frame))
        ax[ int(i /3), i%3 ].set_xlabel(_frame)
        ax[ int(i /3), i%3 ].set_ylabel('price')

        print(  np.cov(  design_matrix_frame[_frame], respone_vec)  /\
         (np.std( design_matrix_frame[_frame]) * np.std(respone_vec)))
    # design_matrix_frame.plot(subplots=True, layout=(4,5))
    plt.show()


if __name__ == '__main__':
    q1()

    _path = "covid19_israel.csv"
    dataset_covid = pd.read_csv(_path)
    dataset_covid["log_detected"] = np.log( dataset_covid["detected"])
    X, Y = dataset_covid["day_num"], dataset_covid["log_detected"]
    X = np.array([X])
    W, S = fit_linear_reggression( X.transpose(), Y )
    gplot = ggplot(dataset_covid, aes(x='day_num' , y='log_detected')) +\
    geom_point()+ geom_abline(slope =W[0]) + ggtitle('log detected as function of the dayes')
```

```
ggsave(gplot, "log_fig.png")
X, Y = dataset_covid["day_num"], dataset_covid["detected"]
Z = np.e ** (W[0]*X)

gplot = ggplot( dataset_covid , aes(x='day_num' , y='detected')) +\
geom_point()+\
geom_line(aes(y='Z')) + geom_abline(slope =W[0]) + ggtitle('detected as function of the dayes')
ggsave(gplot, "fig.png")
```

# Index of comments