6.439 Problem Set 1

Xudong Sun

1.1 The Salk Vaccine Field Trial

(a)

The differences between the two experiments are as follows: compared to the randomized controlled double-blind experiment, 1) the experimenters in the NFIP study were not chosen randomly but according to their grade, therefore it was not randomized; 2) the NFIP study chose the grade 2 to be the experiment group and then asked for consent therefore the vaccine group and no vaccine group cannot be compared directly; 3) the control group in the NFIP study did not get the "placebo vaccine" such as salt injection, and everyone knows who actually got the treatment, therefore the study was not blind.

(b)

The polio rate data from the randomized controlled double-blind experiment. (28 compared to 71)

(c)

As polio is an infectious disease, people who thought they had a higher risk are more likely to accept the experiment. For example, they might live in an endemic area, have immune deficiency, or suffer from malnutrition[1]. If so, the no-consent group's polio rate is expected be lower than the population average. In the NFIP study, it's also possible that people of different age have different polio rate.

(d)

Yes. The study can be biased in both ways. It could exaggerate the effect of the vaccine because we cannot distinguish the true effect of the vaccine from the placebo effect. It could also understate the effect. If the experiment is not blind, the behavior of the vaccine group and no-vaccine group will be different regarding the prevention of the disease. The no-vaccine group knows they are not treated so they might try other ways to prevent the disease, or simply be more cautious, which can also decrease the polio rate.

(e)

They were more likely wrong. The polio rate difference between consent and the no-consent group was very low (highly possible to be statistically insignificant, if we can do a hypothesis test given enough samples), which means joining the experiment wasn't "more dangerous." More importantly, the polio rate difference between the vaccine group and control group was a strong proof that the vaccine was effective.

[1] From Wikipedia, https://en.wikipedia.org/wiki/Poliomyelitis#Prevention

6.439 Problem Set 1

Xudong Sun

1.2 NASA Compton Gamma Ray Observatory Data

(a)

The number of gamma-rays in a given time interval $X \sim Poisson(\lambda)$, λ is the emission rate. Gamma rays are received independently of one another at a given rate.

(b)

To determine whether X follows a Poisson distribution, the null and alternative hypothesis are as follows:

$$H_0: X\ follows\ a\ Poisson\ distribution\ with\ a\ constant\ \lambda$$

$$H_A: each\ interval\ has\ a\ different\ \lambda$$

Use likelihood ratio test.

(c)

Calculate the MLE $\hat{\lambda}$ for $X \sim Poisson(\lambda)$, The likelihood function is:

$$L(X_1, X_2, \dots, X_n; \lambda) = \prod_{i=1}^{n} e^{-\lambda} \cdot \frac{\lambda^{x_i}}{X_i!}$$

The log-likelihood:

$$\mathcal{L}(X_1, X_2, \dots, X_n; \lambda) = -n\lambda + \log(\lambda) \cdot \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} log(X_i!)$$

The first derivative:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -n + \frac{1}{\lambda} \cdot \sum_{i=1}^{n} X_i = 0$$

$$\hat{\lambda} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i$$

The MLE is the average gamma-rays in a given time interval. In our data, the time intervals are not equal, therefore:

$$\hat{\lambda} = \frac{\sum counts}{\sum second} = 0.00388$$

(d)

The alternative hypothesis is that λ is not constant, which means each interval i has its own λ_i. Similar to (c), we can calculate the MLE for each interval:

$$\widehat{\lambda_i} = X_i$$

For the intervals in which $X_i = 0$, we think the interval has a Poisson(0) distribution. The likelihood for these intervals is 1.

The test statistic of the likelihood test:

$$\Lambda(\lambda) = -2\log\left(\frac{L(X_1, X_2, \dots, X_n; \lambda)}{L(X_1, X_2, \dots, X_n; \lambda_1, \lambda_2, \dots, \lambda_n)}\right)$$

$$= -2\log\left(\frac{\prod_{i=1}^{n} e^{-\lambda} \cdot \frac{\lambda^{x_i}}{X_i!}}{\prod_{i=1}^{n} e^{-\lambda_i} \cdot \frac{\lambda_i^{x_i}}{X_i!}}\right), \lambda_i > 0$$

(f)

(g)

1.3 P-values

I don't think it is a good idea just to ban the p-value from scientific papers, neither did the author of the article. In general, merely banning p-values does not solve any problem related to it, and what we need is just a bit more caution.

Firstly, the motivation of banning p-values is reasonable. As the article stated, the misunderstanding and misuse of p-values and null hypothesis significance testing has led to much confusion and even doubt about scientific conclusions. In scientific conclusions and business or policy decisions, reducing data analysis to achieving p-value thresholds can lead to wrong conclusions and bad decisions. By itself, a p-value does not provide a good measure of evidence regarding a model or hypothesis without context or other evidence, which researchers usually leave out.

However, it is still not wise to simply ban the p-value. The p-value is just an indicator summarizing the incompatibility between a particular set of data and a proposed model for the data. The problem is not that the p-value is "wrong" but it is misused and misunderstood without many contextual factors that is necessary for data analysis and scientific inference.

In sum, the p-value should be used with more caution. Researchers and audience should always use or interpret the p-value correctly.

1.4 Detecting Leukemia Types

For each gene, we use t-test on its expression level data of ALL and AML patients.

The number of genes that are associated with the different tumor types are as follows:

	Number of genes
Uncorrected p-values	1045

Holm-Bonferroni correction	98
Benjamini-Hochberg correction	681

As the expression level data of ALL and AML patients have different sample size, we try Welch's t-test in case that the variance are unequal between two groups.

	Number of genes
Uncorrected p-values	1078
Holm-Bonferroni correction	103
Benjamini-Hochberg correction	695

The Python code are as follows:

```python
# bonferroni correction
def holm_bonferroni(pvals, alpha=0.05):
    m, pvals = len(pvals), np.asarray(pvals)
    ind = np.argsort(pvals)
    test = [p > alpha/(m+1-k) for k, p in enumerate(pvals[ind])]
    significant = np.zeros(np.shape(pvals), dtype='bool')
    significant[ind[0:m-np.sum(test)]] = True
    return significant

# Benjamini-Hochberg procedure
def BH(pvals, q=0.05):
    m = len(pvals)
    significant = np.zeros(m, dtype='bool')
    sort_ind = np.argsort(pvals).astype(int)+1 # sort the p-values

    for i in range(1,m+1): #i = the individual p-value's rank
        if pvals[sort_ind[i]] < (i)*q/m:
            significant[sort_ind[i]-1] = True # record the significant index
    return significant
```

6.439 Problem Set 1

Xudong Sun

1.6 Regression and Gradient Descent

(a)

$$\hat{\beta} = (X^T X)^{-1} X^T Y = \begin{bmatrix} 1.9296 \\ 1.2640 \\ -4.5980 \end{bmatrix}$$

(b)

The Python code for gradient descent are as follows:

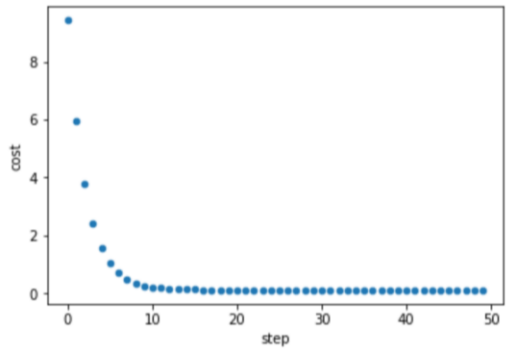
```
def gradientDescent(X, Y, beta_0, alpha, t):
    m, n = X.shape # m is number of cases, n is the number of variables
    cost = pd.DataFrame(np.zeros([t,2]))
    cost.columns = ['step','cost']
    beta = beta_0

    for i in range(t):
        # vectorized gradient: X'*(Y-X*beta)
        res = Y- np.matmul(X, beta)
        beta = beta + 2 * alpha * (1/m) * np.matmul(np.transpose(X), res)
        # calculate the cost base on current beta
        cost['step'][i] = i
        cost['cost'][i] = calCost(X, Y, beta)

    cost.plot(kind = 'scatter', x = 'step',y = 'cost')
    return beta, cost

def calCost(X, Y, beta):
    m, n = X.shape
    # vectorized cost: (X*beta - Y)'(X*beta - Y)
    residual = Y- np.matmul(X, beta)
    return (1/(2*m))*np.matmul(np.transpose(residual), residual)
```

Try different initial value and α:

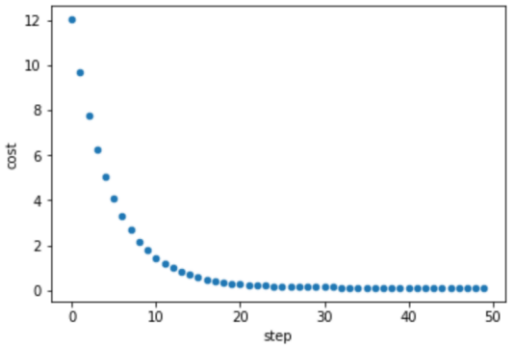
1. $\beta_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}, \alpha = 0.1, t = 50$

6.439 Problem Set 1

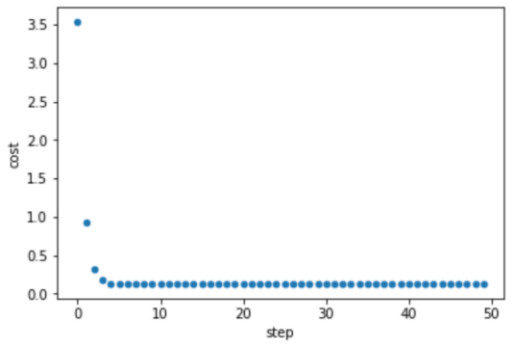
Xudong Sun



2. $\beta_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}, \alpha = 0.05, t = 50$



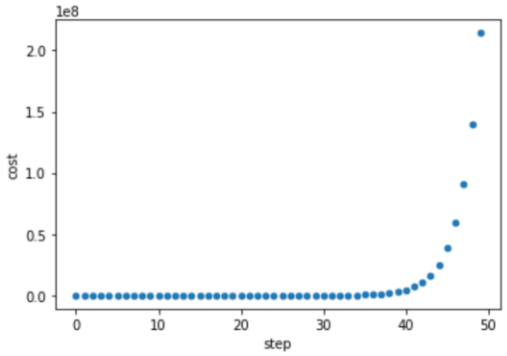
3. $\beta_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}, \alpha = 0.25, t = 50$



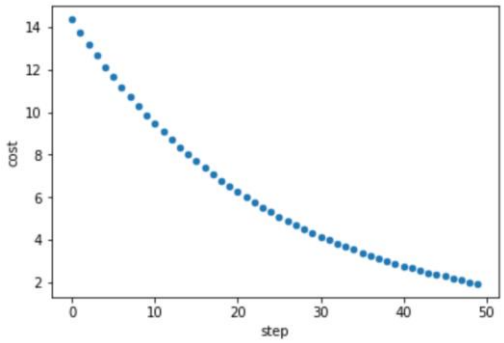
4. $\beta_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}, \alpha = 0.8, t = 50$

6.439 Problem Set 1

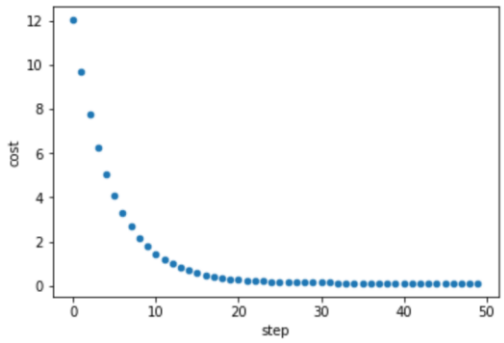
Xudong Sun



5. $\beta_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}, \alpha = 0.01, t = 50$



6. $\beta_0 = random\ number\ from\ (0,1), \alpha = 0.05, t = 50$



The optimal step size can be decided by setting a threshold ε. Theoretically, the optimal step size and threshold can be decided by finding bounds. Find constant L such that for all points u:

$$f(u) \leq f(w) + \langle \nabla f(w), u - w \rangle + \frac{L}{2}\|u - w\|^2$$

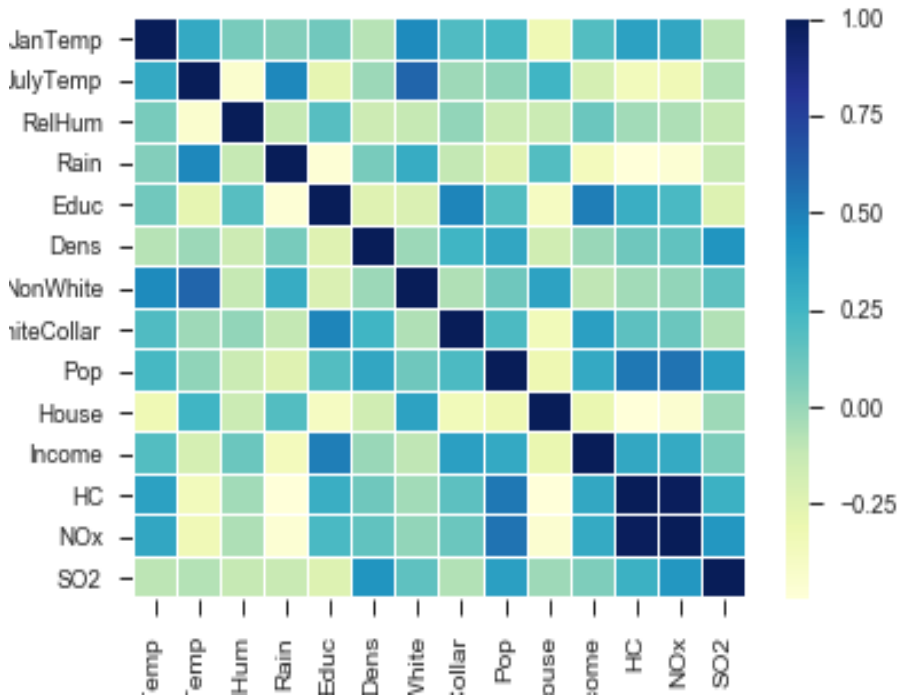
The step size $\alpha_t = \frac{1}{L}$ gives convergence, $\varepsilon = \left(1 - \frac{m}{L}\right)^t (f\left(w^{(0)} - f(w^*)\right)$ need $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ iterations for $f\left(w^{(t)}\right) - f(w^*) \leq \varepsilon$.

6.439 Problem Set 1

Xudong Sun

(c)

We firstly check the correlation matrix of all explanatory variables:

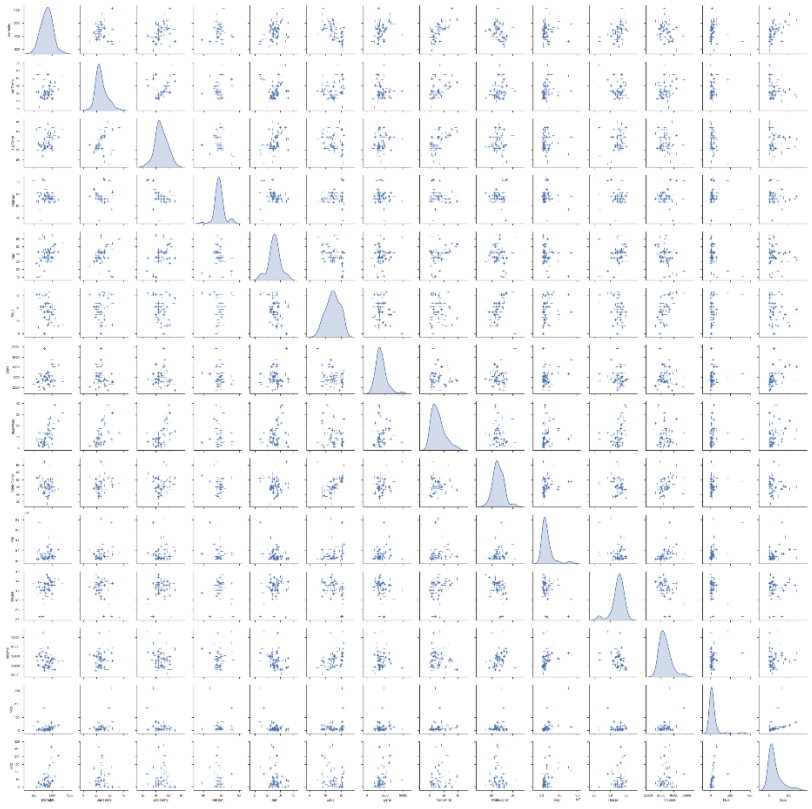


We found that the HC and NOx are highly correlated so one of them need to be removed from our model.

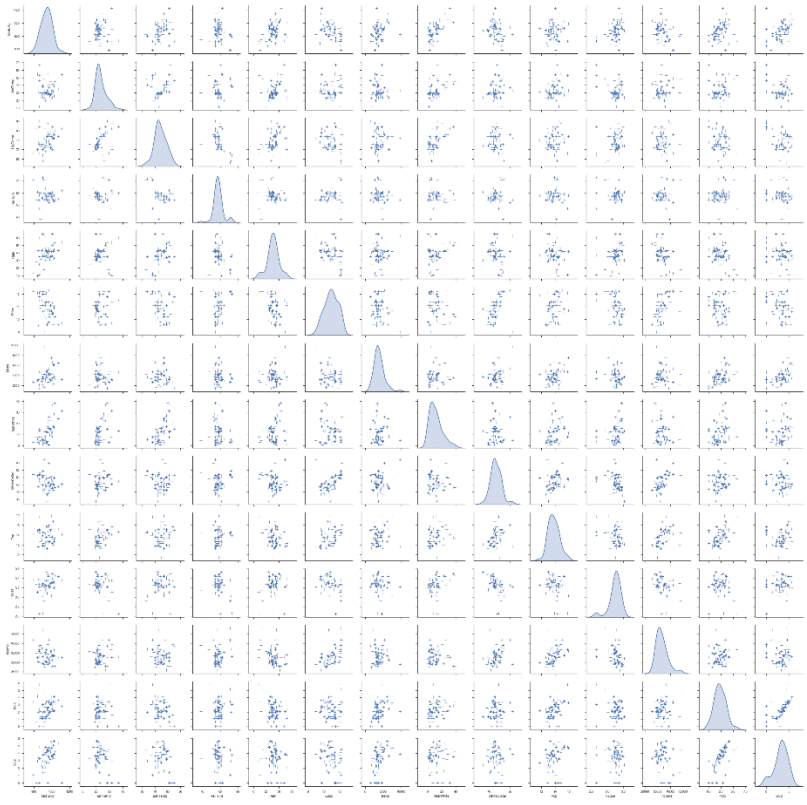
Then, we investigate the histograms and scatterplots of each variable/variable pairs. From the plot, we think SO2, NOx, and Pop need log-transformation.

6.439 Problem Set 1

Xudong Sun



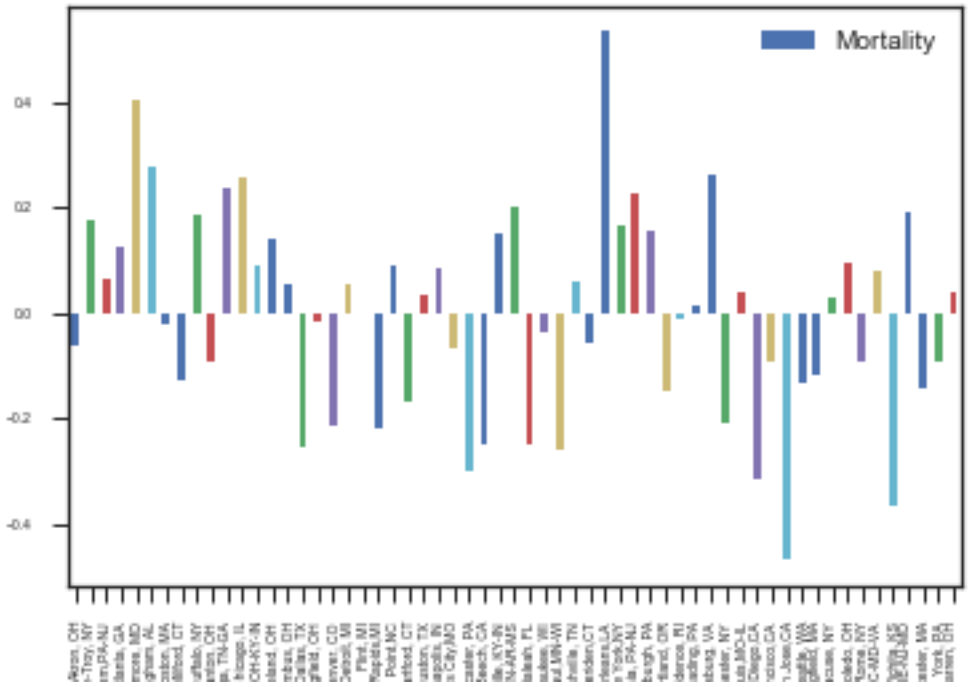
After log-transformation:



6.439 Problem Set 1

Xudong Sun

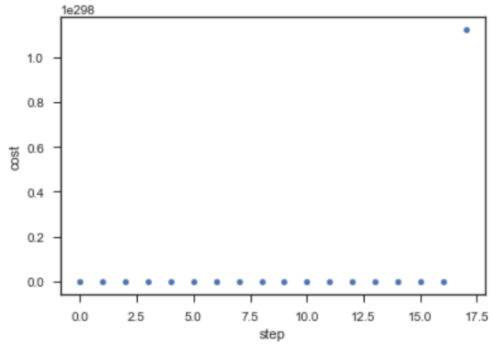
We then normalized all numeric variables. Plot the mortality rate of all cities:



New Orleans, LA, has the highest mortality while San Jose has the lowest.

(d) (e)

Run gradient descent on raw data:

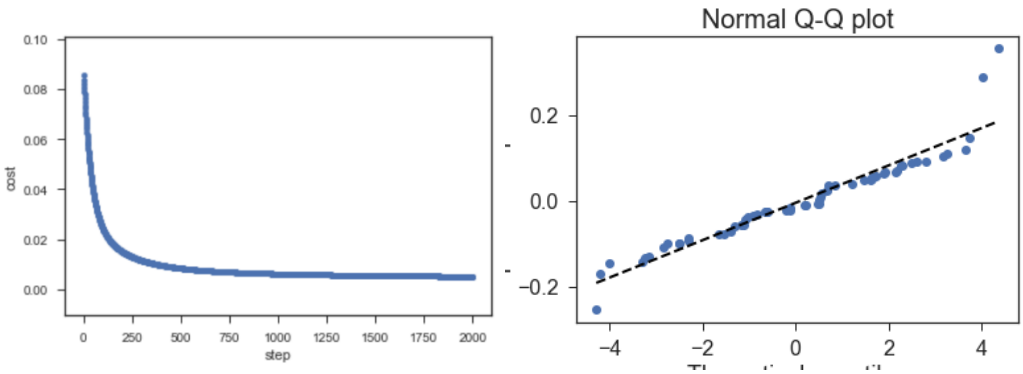


The gradient descent does not converge.

Run gradient descent on transformed data:

6.439 Problem Set 1

Xudong Sun



The gradient descent successfully converged. We plot the q-q plot and the residuals are normally distributed.

(f) The function to minimize:

$$f(\beta) = \sum_i log(1 + exp(-y_i \beta^T x_i))$$

The derivative of $f(\beta)$:

$$\nabla_\beta f(\beta) = \sum_i \nabla_\beta \log(1 + exp(-y_i \beta^T x_i))$$

$$= \sum_i \frac{1}{1 + \exp(-y_i \beta^T x_i)} \cdot \exp(-y_i \beta^T x_i) \cdot (-y_i x_i)$$

In each step of the gradient descent:

$$\beta := \beta - \alpha \cdot \nabla_\beta f(\beta)$$

1.7 Computational Aspects of Regression

(a)

To store the matrix X, we need $10^8 \times 200 \times 64/10^9 = 1280$gb of memory. It's (currently, in 2018) impossible to store or perform matrix calculations on common computers.

(b)

Use Stochastic Gradient Descent (SGD) or mini-batch gradient descent by doing 1 iteration of gradient descent on 1 data entry or a small batch of data. Thus, we can store the data on cloud and extract small amount of data each time.

(c)

The maximum rank of $X^T X$ is the number of rows, n, when the number of variables p is much larger than n, there will be many local optimums.