1.

(a) True $\hat{\beta}^{OLS} = (X^TX)^{-1}X^TY$ $E(\hat{\beta}^{OLS}) = (X^TX)^{-1}X^TX\beta^* = \beta^*$

(b) True bias-variance trade-off is shown is Q.3.4

Variance $= 62 \sum_{i=1}^{d} \frac{d_{ii}}{(d_{ii}+\lambda)^2}$ is monotonically decreasing, when $\lambda \uparrow$

and bias is monotonically increasing when $\lambda \uparrow$

(c) True ① $MSE(\hat{\beta}^{ols}) = Var(\hat{\beta}^{ols})$ Suppose ① = ② at some λ

② $MSE(\hat{\beta}_\lambda^{Lasso}) = Var(\hat{\beta}_\lambda^{Lasso}) + bias(\hat{\beta}_\lambda^{Lasso})^2$

$> Var + \Delta_V + B - \Delta_B$, where $\Delta_V < \Delta_B$

① > ②

(d) False $\hat{\beta}^{Lasso} = 0$ when $\lambda = \infty$

(e) False $\lambda > 0$, every solution $\hat{\beta}$ has the same L_1 norm.

(f) True $H = UDU^T$ $H^2 = UDU^T = H$

$(UDU^T)(UDU^T) = UD^2U^T = UDU^T$

$d_{ii}^2 = d_{ii}$

$d_{ii}(d_{ii}-1) = 0$ $d_{ii} = 1$ or $d_{ii} = 0$

(g) True A symmetric matrix with non-negative eigenvalues is a PSD

$(X(X^TX)^{-1}X^T)^T = X(X^TX)^{-1}X^T$ symmetric

$X(X^TX)^{-1}X^T \cdot X(X^TX)^{-1}X^T = X(X^TX)^{-1}X^T$ idempotent

H has only non-negative eigenvalues, so H is a PSD

(h) False $\text{trace}(Q) = \sum_{i=1}^{n}(I_{ii} - H_{ii}) = n - (d) = n-d$

$\text{trace}(H) = \text{trace}(X^TX(X^TX)^{-1}) = \text{trace}(I_d) = d$

2.

1. $\hat{\beta} = (X^TX)^{-1}X^TY$ $\quad \mathbb{E}(\hat{\beta}) = (X^TX)^{-1}X^T\mathbb{E}(Y)$

$= (X^TX)^{-1}X^TX\beta^*$

$= \beta^*$

$Cov(\hat{\beta}) = Cov[(X^TX)^{-1}X^TY] = (X^TX)^{-1}X^T\,Cov(Y)\,[(X^TX)^{-1}X^T]^T$

$= \sigma^2 (X^TX)^{-1}X^TX(X^TX)^{-1}$

$= \sigma^2(X^TX)^{-1}$

Since Y is Normal, $\hat{\beta}$ is a linear function of Y

so $\hat{\beta}$ is also normal $\hat{\beta} \sim N(\beta^*, \sigma^2(X^TX)^{-1})$

$\mathbb{E}(X\hat{\beta}) = X\mathbb{E}(\hat{\beta}) = X(X^TX)^{-1}X^TY = X(X^TX)^{-1}X^TX\beta^*$

$= X\beta^*$

2. $\mathbb{E}(RSS) = \mathbb{E}(\hat{e}^T\hat{e}) = \mathbb{E}(Y^T(I-H)^T(I-H)Y) = \mathbb{E}(e^T(I-H)e)$

Since $\hat{e} = (I-H)Y$

$= (I-H)(X\beta+e)$

$= (I-H)e$

$= \sum_{i,j}(I_{ij}-H_{ij})\mathbb{E}(e_ie_j)$

Since e_i e_j uncorrelated when $i \neq j$

$= \sum_{i=1}^{n}\sigma^2(I_i - h_{ii})$

$= \sigma^2(n - tr(H))$

$= \sigma^2(n-d)$

$tr(H) = tr(X(X^TX)^{-1}X^T)$

$= tr((X^TX)^{-1}X^TX)$

$= tr(I_d)$

$= d$

3.

We know OLS $\hat{\beta}$ is unbiased

let linear unbiased estimator $\bar{\beta} = DY$, $\bar{\beta}$ is OLS when $D = (X^TX)^{-1}X^T$

now let $D = A + (X^TX)^{-1}X^T$

$$E(\bar{\beta}) = E((A+(X^TX)^{-1}X^T)Y)$$
$$= E(AY) + \beta^*$$
$$= AX\beta^* + \beta^*$$

since $\bar{\beta}$ is unbiased, we need $AX = 0$

$$Cov(\bar{\beta}) = Cov((A+(X^TX)^{-1}X^T)Y)$$
$$= (A+(X^TX)^{-1}X^T)\,Cov(Y)\,(A+(X^TX)^{-1}X^T)^T$$
$$= \sigma^2(A+(X^TX)^{-1}X^T)(A^T + X(X^TX)^{-1})$$
$$= \sigma^2(AA^T + AX(X^TX)^{-1} + (X^TX)^{-1}X^TA^T + (X^TX)^{-1})$$
$$= \sigma^2 AA^T + \sigma^2(X^TX)^{-1}$$
$$= \sigma^2 AA^T + Cov(\hat{\beta}^{OLS})$$

we have $Cov(\bar{\beta}) \geq Cov(\hat{\beta}^{OLS})$

AA^T is a PSD

2.4

We train use $(X_1, y_1) \cdots (X_N, y_N)$ to get least square estimate $\hat{\beta}$

this means we minimize $R_{tr}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i)^2$ to get $\hat{\beta}$

When we predict, we use the data we've never seen, if we are lucky to have exactly same test data as traing data. We have $E(R_{tr}(\hat{\beta})) = E(R_{te}(\hat{\beta}))$

But usually we have new data we've never seen, so on average we'll predict worse in test data than in training data. That is $E(R_{tr}(\hat{\beta})) \leq E(R_{te}(\hat{\beta}))$

3.

1. $\min_{\theta} \|X\theta - y\|_2^2 + \lambda\|\theta\|_2^2$

$= \min_{\theta} \|X\theta - y\|_2^2 + \|\sqrt{\lambda}\theta + 0\|_2^2$

$= \min_{\theta} \|A\theta - Y\|_2^2$

let $A = \begin{pmatrix} X_{n\times d} \\ \sqrt{\lambda} I_{d\times d} \end{pmatrix}$ $Y = \begin{pmatrix} y \\ 0 \end{pmatrix}$

This is similar to LS, we can use normal equation to get $\hat{\theta}_\lambda$

$$\hat{\theta}_\lambda = (A^TA)^{-1}A^TY$$

$$= \left[(X^T \sqrt{\lambda} I)\cdot\begin{pmatrix} X \\ \sqrt{\lambda} I \end{pmatrix}\right]^{-1}\cdot(X^T \sqrt{\lambda} I)\begin{pmatrix} y \\ 0 \end{pmatrix}$$

$$= (X^TX + \lambda I)^{-1}X^Ty$$

X^TX is a PSD

λI is a PD for $\lambda > 0$

So $X^TX + \lambda I$ is a PD

and PD is always invertible

So we have a unique solution for any $\lambda > 0$

let A be PSD, then $x^TAx \geq 0$ for any vector x

let B be PD, the $x^TBx > 0$

$x^T(A+B)x = x^TAx + x^TBx > 0$

$\geq 0 \quad > 0$

So $A+B$ is a PD

3.2. $\mathbb{E}(\hat{\theta}_\lambda) = \mathbb{E}((X^TX+\lambda I)^{-1}X^Ty)$

$= (X^TX+\lambda I)^{-1}X^T\mathbb{E}(y)$

$= (X^TX+\lambda I)^{-1}X^TX\theta^*$

$= W_\lambda\theta^*$

$Cov(\hat{\theta}_\lambda) = (X^TX+\lambda I)^{-1}X^T Cov(y) X (X^TX+\lambda I)^{-1}$

$= \sigma^2 W_\lambda (X^TX+\lambda I)^{-1}$

Since $\hat{\theta}_\lambda$ is a linear function of y, and y is normal

So $\hat{\theta}_\lambda$ is normal as well.

$\hat{\theta}_\lambda \sim N(W_\lambda\theta^*, \sigma^2 W_\lambda (X^TX+\lambda I_d)^{-1})$

3.3

$$\|\mathbb{E}(\hat{\theta}_\lambda)-\theta^*\|_2^2 = \|W_\lambda\theta^* - \theta^*\|^2$$

$$= \|[(X^TX+\lambda I_d)^{-1}X^TX - I]\theta^*\|_2^2$$

use the face $(X^TX+\lambda I_d)$ is invertible

$$= \|[(X^TX+\lambda I_d)^{-1}X^TX - (X^TX+\lambda I_d)^{-1}(X^TX+\lambda I_d)]\theta^*\|_2^2$$

$$= \|-\lambda(X^TX+\lambda I_d)^{-1}\theta^*\|_2^2$$

$$= \lambda^2\|(UDU^T+\lambda UU^T)^{-1}\theta^*\|_2^2$$

$$= \lambda^2\|U(D+\lambda I)^{-1}U^T\theta^*\|_2^2$$

$$= \lambda^2 tr(\theta^{*T}U(D+\lambda I)^{-1}(D+\lambda I)^{-1}U^T\theta^*)$$

$$= \lambda^2 tr(\|(D+\lambda I)^{-1}U^T\theta^*\|_2^2)$$

$$= \sum_{i=1}^{d}\frac{\lambda^2}{(d_i+\lambda)^2}(u_i^T\cdot\theta^*)^2$$

where $U = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_d \\ | & & | \end{pmatrix}$

$$U^T = \begin{pmatrix} u_1^T \\ \vdots \\ u_d^T \end{pmatrix}$$

u_i^T is the ith column of U

$$\mathbb{E}\|\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)\|_2^2$$

$$= \mathbb{E}(tr((\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)^T(\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda))))$$

$$= \mathbb{E}(tr((\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)(\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)^T)))$$

$$= tr(\mathbb{E}((\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)(\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)^T)))$$

$$= tr(cov(\hat{\theta}_\lambda))$$

where $cov(\hat{\theta}_\lambda) = \sigma^2 W_\lambda(X^TX+\lambda I_d)^{-1}$

$$= \sigma^2(UDU^T+\lambda I_d)^{-1}UDU^T(UDU^T+\lambda I_d)^{-1}$$

$$= \sigma^2 U(D+\lambda I)^{-1}U^TUDU^TU(UDU^T+\lambda I_d)^{-1}U^T$$

$$= \sigma^2 U(D+\lambda I)^{-1}D(D+\lambda I)^{-1}U^T$$

$$= tr(\sigma^2(D+\lambda I)^{-1}D(D+\lambda I)^{-1})$$

$$= tr(\sigma^2(D+\lambda I)^{-2}D)$$

$$= \sum_{i=1}^{d}\sigma^2\cdot\frac{d_i}{(d_i+\lambda)^2}$$

3.4

$\lambda = 0$ squared bias $= 0$ scalar variance $= \sigma^2 \sum_{i=1}^{n} \frac{1}{d_i^2}$

$\lambda = \infty$ squared bias $= \sum_{i=1}^{d} \theta_i^{*2}$ scalar variance $= 0$

when bias is at its minimum, the variance reaches its maximum

When bias is at its maximum, the variance reaches its minimum.

and squared bias is a monotonic increasing function of λ for $\lambda \in [0, +\infty)$

scalar variance is a monotonic decreasing function of λ for $\lambda \in [0, +\infty)$

Variance $= \sigma^2 \sum_{i=1}^{d} \frac{1}{(d_i+\lambda)^2}$ as λ increases, variance decreases, bias increases

3.5

$$\text{trace}(M(\lambda)) = \text{trace}(\mathbb{E}[(\hat{\theta}_\lambda - \theta^*)(\hat{\theta}_\lambda - \theta^*)^T])$$

$$= \mathbb{E}(\text{trace}((\hat{\theta}_\lambda - \theta^*)(\hat{\theta}_\lambda - \theta^*)^T))$$

$$= \mathbb{E}(\text{trace}((\hat{\theta}_\lambda - \theta^*)^T(\hat{\theta}_\lambda - \theta^*)))$$

$$= \mathbb{E}(\|\hat{\theta}_\lambda - \theta^*\|_2^2)$$

$$\mathbb{E}(\|\hat{\theta}_\lambda - \theta^*\|_2^2) = \mathbb{E}(\|\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda) + \mathbb{E}(\hat{\theta}_\lambda) - \theta^*\|_2^2)$$

$$= \mathbb{E}(\|\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda)\|_2^2 + \mathbb{E}(\|\mathbb{E}(\hat{\theta}_\lambda) - \theta^*\|_2^2)$$

$$+ 2\mathbb{E}((\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda))^T \cdot (\mathbb{E}(\hat{\theta}_\lambda) - \theta^*)]$$

$$= \text{Scalar-variance} + \text{squared bias}$$

where $\mathbb{E}(\hat{\theta}_\lambda) - \theta^*$ is Constant

$$+ 2(\mathbb{E}(\hat{\theta}_\lambda) - \theta^*) \cdot \mathbb{E}((\hat{\theta}_\lambda - \mathbb{E}(\hat{\theta}_\lambda))^T)$$

$$= \text{Scalar-variance} + \text{Squared bias}$$

where $\mathbb{E}(\hat{\theta}_\lambda^T) - \mathbb{E}(\mathbb{E}(\hat{\theta}_\lambda)^T) = 0$

3.6. $\hat{\theta}^{OLS}$ is when $\lambda = 0$

$$\begin{aligned}\text{trace}(M(0)) &= \text{trace}(\mathbb{E}(\hat{\theta}^{OLS}-\theta^*)(\hat{\theta}^{OLS}-\theta^*)^T) \\ &= \mathbb{E}(\text{trace}((\hat{\theta}^{OLS}-\theta^*)^T(\hat{\theta}^{OLS}-\theta^*))) \\ &= \mathbb{E}(\|\hat{\theta}^{OLS}-\theta^*\|^2) \\ &= MSE(\hat{\theta}^{OLS})\end{aligned}$$

$MSE(\hat{\theta}^{OLS}) > MSE(\hat{\theta}_\lambda)$

$\Rightarrow$ $MSE(\hat{\theta}^{OLS}) - MSE(\hat{\theta}_\lambda) > 0$

$\Rightarrow$ $\text{trace}(M(0)) - \text{trace}(M(\lambda)) > 0$

$\Rightarrow$ $\text{trace}(M(0) - M(\lambda)) > 0$

We need to prove $M(0)-M(\lambda)$ is a PD matrix (symmetric ✓)

Since the trace of a matrix is equal to the sum of its eigenvalues, and a PD (symmetric) matrix has positive eigenvalues, so if $M(0)-M(\lambda)$ is a PD, then we can prove $\text{trace}(M(0)-M(\lambda)) > 0$

first derive $M(\lambda)$

$$\begin{aligned}M(\lambda) &= \mathbb{E}[(\hat{\theta}_\lambda-\theta^*)(\hat{\theta}_\lambda-\theta^*)^T] \\ &= \mathbb{E}[(\hat{\theta}_\lambda-\mathbb{E}(\hat{\theta}_\lambda)+\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)(\hat{\theta}_\lambda-\mathbb{E}(\hat{\theta}_\lambda)+\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)^T] \\ &= \mathbb{E}[(\hat{\theta}_\lambda-\mathbb{E}(\hat{\theta}_\lambda))(\hat{\theta}_\lambda-\mathbb{E}(\hat{\theta}_\lambda))^T] + \mathbb{E}[(\hat{\theta}_\lambda-\mathbb{E}(\hat{\theta}_\lambda))(\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)^T] \quad ① \\ &\quad + \mathbb{E}[(\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)(\hat{\theta}_\lambda-\mathbb{E}(\hat{\theta}_\lambda))^T] \quad ② + \mathbb{E}[(\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)(\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)^T]\end{aligned}$$

while for ① and ② $\mathbb{E}(\hat{\theta}_\lambda)-\theta^*$ is a constant, and $\mathbb{E}(X-\mathbb{E}(X)) = 0$

$$\begin{aligned}&= \text{Var}(\hat{\theta}_\lambda) + 0 + 0 + (\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)(\mathbb{E}(\hat{\theta}_\lambda)-\theta^*)^T \\ &= \sigma^2 W_\lambda (X^TX+\lambda I_d)^{-1} + (W_\lambda\theta^*-I_d)\theta^*\theta^{*T}(W_\lambda-I_d)^T\end{aligned}$$

next page

$M(0) - M(\lambda)$ when $\lambda = 0$ $W_\lambda = I$

$= \underbrace{\sigma^2 (X^TX)^{-1} - \sigma^2 W_\lambda (X^TX + \lambda I_d)^{-1}}_{①} + \underbrace{(W_\lambda \theta^* - I_d)\theta^* \theta^{*T} (W_\lambda - I_d)^T}_{②}$

$① = \sigma^2 (W_\lambda W_\lambda^{-1} (X^TX)^{-1} (W_\lambda^T)^{-1} W_\lambda^T - W_\lambda (X^TX)^{-1} W_\lambda^T)$

$= \sigma^2 W_\lambda (W_\lambda^{-1} (X^TX)^{-1} (W_\lambda^T)^{-1} - (X^TX)^{-1}) W_\lambda^T$ let $S = (X^TX)$

$= \sigma^2 W_\lambda (S^{-1}(S+\lambda I) S^{-1} (S+\lambda I) S^{-1} - S^{-1}) W_\lambda^T$

$= \sigma^2 W_\lambda (S^{-1} (SS^{-1}S + SS^{-1}\lambda + \lambda S^{-1} S + \lambda S^{-1} \lambda) S^{-1} - S^{-1}) W_\lambda^T$

$= \sigma^2 W_\lambda (S^{-1} (\lambda (2I + \lambda S^{-1}) S^{-1}) W_\lambda^T$

$= \sigma^2 W_\lambda (2\lambda (X^TX)^{-2} + \lambda^2 (X^TX)^{-3}) W_\lambda^T$

$= \sigma^2 (X^TX + \lambda I_d)^{-1} (2\lambda I_d + \lambda^2 (X^TX)^{-1}) (X^TX + \lambda I_d)^{-1}$

② use the fact $W_\lambda - I_d = -\lambda (X^TX + \lambda I_d)^{-1}$

Because $(X^TX + \lambda I)^{-1} X^TX - I_d = (X^TX + \lambda I)^{-1} X^TX - (X^TX + \lambda I)^{-1} (X^TX + \lambda I)$

$= (X^TX + \lambda I)^{-1} (-\lambda I)$

$= -\lambda (X^TX + \lambda I_d)^{-1}$

$② = -\lambda (X^TX + \lambda I_d)^{-1} \theta^* \cdot \theta^{*T} (-\lambda (X^TX + \lambda I_d)^{-1})$

$= \lambda^2 (X^TX + \lambda I_d)^{-1} \theta^* \theta^{*T} (X^TX + \lambda I_d)^{-1}$

next page

$M(0) - M(\lambda)$

$= ① - ②$

$= \lambda(X^TX+\lambda I_d)^{-1}\underbrace{[2\delta^2 I_d + \delta^2\lambda(X^TX)^{-1} - \lambda\theta^*\theta^{*T}]}_{B}\underbrace{(X^TX+\lambda I_d)^{-1}}_{A}$

A is invertible and $A^T = ((X^TX+\lambda I_d)^{-1})^T = (X^TX+\lambda I_d)^{-1} = A$

We have $M(0) - M(\lambda) = \lambda A^T B A$

This is a note:

We only need B to be PD

because let z be any non-zero vector in $\mathbb{R}^n$

$z^TA^TBAz = (Az)^TBAz$

$= w^TBw > 0$ iff B is PD

Since A is invertible $Az \neq 0$

$B = 2\delta^2 I_d + \delta^2\lambda(X^TX)^{-1} - \lambda\theta^*\theta^{*T}$

X^TX is symmetric for any non-zero vector $z \in \mathbb{R}^n$

$z^TX^TXz = (Xz)^T(Xz) = \|Xz\|_2^2 \geq 0$

Thus X^TX is a PSD, this means $(X^TX)^{-1}$ is a PSD

PSD + PD = PD, so we need $2\delta^2 I_d - \lambda\theta^*\theta^{*T}$ to be a PD

Similarly let z be any non-zero vector $\in \mathbb{R}^n$

$2\delta^2 z^T I_d z - \lambda z^T\theta^*\theta^{*T}z$

$= 2\delta^2\|z\|_2^2 - \lambda(\theta^{*T}z)^T\theta^{*T}z > 0$ note $\theta^{*T}z$ is a scalar

$\lambda < \frac{2\delta^2\|z\|_2^2}{\|\theta^{*T}z\|_2^2} = \frac{2\delta^2\|z\|_2^2}{\|\theta^*\|_2^2\|z\|_2^2} = \frac{2\delta^2}{\|\theta^*\|_2^2}$

Conclude $MSE(\hat{\theta}_\lambda) < MSE(\theta^{OLS})$ for $\lambda \in (0, \frac{2\delta^2}{\|\theta^*\|_2^2})$

4.1

① $f(\lambda x+(1-\lambda)y) = \frac{L}{2}(\lambda x+(1-\lambda)y)^2 = \frac{L}{2}(\lambda^2x^2+2\lambda x(1-\lambda)y+(1-\lambda)^2y^2)$

② $\lambda f(x)+(1-\lambda)f(y) = \lambda\frac{L}{2}x^2+(1-\lambda)\frac{L}{2}y^2$

To prove convexity of $f(x)=\frac{L}{2}x^2$

let ② > ①

$$\lambda f(x)+(1-\lambda)f(y)-f(\lambda x+(1-\lambda)y) = \lambda(1-\lambda)\frac{L}{2}(x^2+y^2-2xy)$$

$$= \lambda(1-\lambda)\frac{L}{2}(x-y)^2 \geq 0$$

Since $\lambda \in [0,1]$, $\frac{L}{2} \geq 0$

Thus $f(x)$ is convex

$\nabla_x f(x) = Lx$ gradient update

$$x_{t+1} = x_t - \gamma L x_t$$

$$= x_t(1-\gamma L)$$

$$= x_{t-1}(1-\gamma L)(1-\gamma L)$$

$$= x_{(0)}(1-\gamma L)^{t+1}$$

4.2

when $\gamma = \frac{1}{L}$, $x_t = x_{(0)}(1-\frac{1}{L}L)^t$ $x_{(1)}$ becomes 0 right away

$= 0$

when $\gamma = \frac{2}{L}$, $x_t = x_{(0)}(1-\frac{2}{L}\cdot L)^t$

$= x_{(0)}(-1)^t$ x_t jumps between $\{-x_0, x_0\}$

$\gamma \in (0, \frac{2}{L})$ where $\gamma \neq \frac{1}{L}$

$\gamma L \in (0,1) \cup (1,2)$

$1-\gamma L \in (-1,0) \cup (0,1)$

We have $x_t = x_{(0)}(1-\gamma L)^t$

Since $1-\gamma L$ shrink $x_{(0)}$ as t increases

we get a convergence

let $|x_t - x_*| \leq \varepsilon$ x_* is a point where $f(x)$ gets optimum (gradient is zero)

$$|x_t| - |x_*| \leq |x_t - x_*| \leq \varepsilon$$

$$|x_{(0)}(1-\gamma L)^t| \leq \varepsilon + |x_*|$$

$$|(1-\gamma t)^t| \leq \frac{\varepsilon + |x_*|}{|x_{(0)}|}$$

$$t\log(|1-\gamma t|) \leq \log\left(\frac{\varepsilon + |x_*|}{|x_{(0)}|}\right)$$

$x_* = 0$ in this case

$$t \leq \frac{\log\left(\frac{\varepsilon}{|x_{(0)}|}\right)}{\log(|1-\gamma t|)}$$

4.3

graphically $f(x) = \frac{1}{2}(x-c)^2$ is a horizontally shifted $f(x) = \frac{1}{2}x^2$

so we'll get the same steps

4.4. $\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} L & 0 \\ 0 & m \end{pmatrix}$

let $z \in \mathbb{R}^n$ $z \neq 0$, $z^T \begin{pmatrix} L & 0 \\ 0 & m \end{pmatrix} z = (z_1, z_2) \begin{pmatrix} L & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$

$= Lz_1^2 + mz_2^2 > 0$ $\nabla^2 f(x)$ is PD

Thus $f(x)$ is convex

$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} Lx_1 \\ mx_2 \end{pmatrix}$

$x^t = x^{t-1} - \gamma \nabla f(x)$

$= \begin{pmatrix} x_1^{t-1} \\ x_2^{t-1} \end{pmatrix} - \gamma \begin{pmatrix} Lx_1^{t-1} \\ mx_2^{t-1} \end{pmatrix}$

$= \begin{pmatrix} x_1^{t-1}(1-\gamma L) \\ x_2^{t-1}(1-\gamma m) \end{pmatrix}$

$= \begin{pmatrix} x_1^{t-2}(1-\gamma L)^2 \\ x_2^{t-2}(1-\gamma m)^2 \end{pmatrix}$

$\vdots$

$= \begin{pmatrix} x_1^0 (1-\gamma L)^t \\ x_2^0 (1-\gamma m)^t \end{pmatrix}$

$= \begin{pmatrix} (1-\gamma L)^2 & 0 \\ 0 & (1-\gamma m)^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$

5.1 when $d > n$, $\max(\text{rank}(X)) = n$

while the number of columns are $d > n$

X is not full rank, so X is not invertible

X^TX is also not invertible. Thus OLS is not unique.

X: $n \times p$

$\text{rank}(X) = \text{rank}(X^TX) \leq n < p$

$$f(\theta) = \frac{1}{2}\|X\theta - Y\|_2^2$$

$$\nabla_\theta f(\theta) = X^T(X\theta - Y) = 0$$

$$X^T(X\hat{\theta}_{OLS} - Y) = 0$$

$$X^T \cdot \hat{e} = 0$$

residuals are still orthogonal to span(X) and their sum is 0

5.2 ridge solution is unique, because $X^TX + \lambda I_d$ is a PD.

X^TX is PSD, λI_d is PD for $\lambda > 0$, so $X^TX + \lambda I_d$ is PD

PD is invertible, so $\hat{\beta}_{(\lambda)} = (X^TX + \lambda I_d)^{-1}X^TY$ has a unique solution

5.3

$\lim_{\lambda \to 0} (X^TX + \lambda I_d)^{-1}X^TY = \hat{\beta}_{OLS}$ and this OLS is not unique

```
## Warning: package 'glmnet' was built under R version 3.4.4

## Warning: package 'Matrix' was built under R version 3.4.4

## Warning: package 'ggplot2' was built under R version 3.4.4

## Warning: package 'caret' was built under R version 3.4.4
```

#5.4

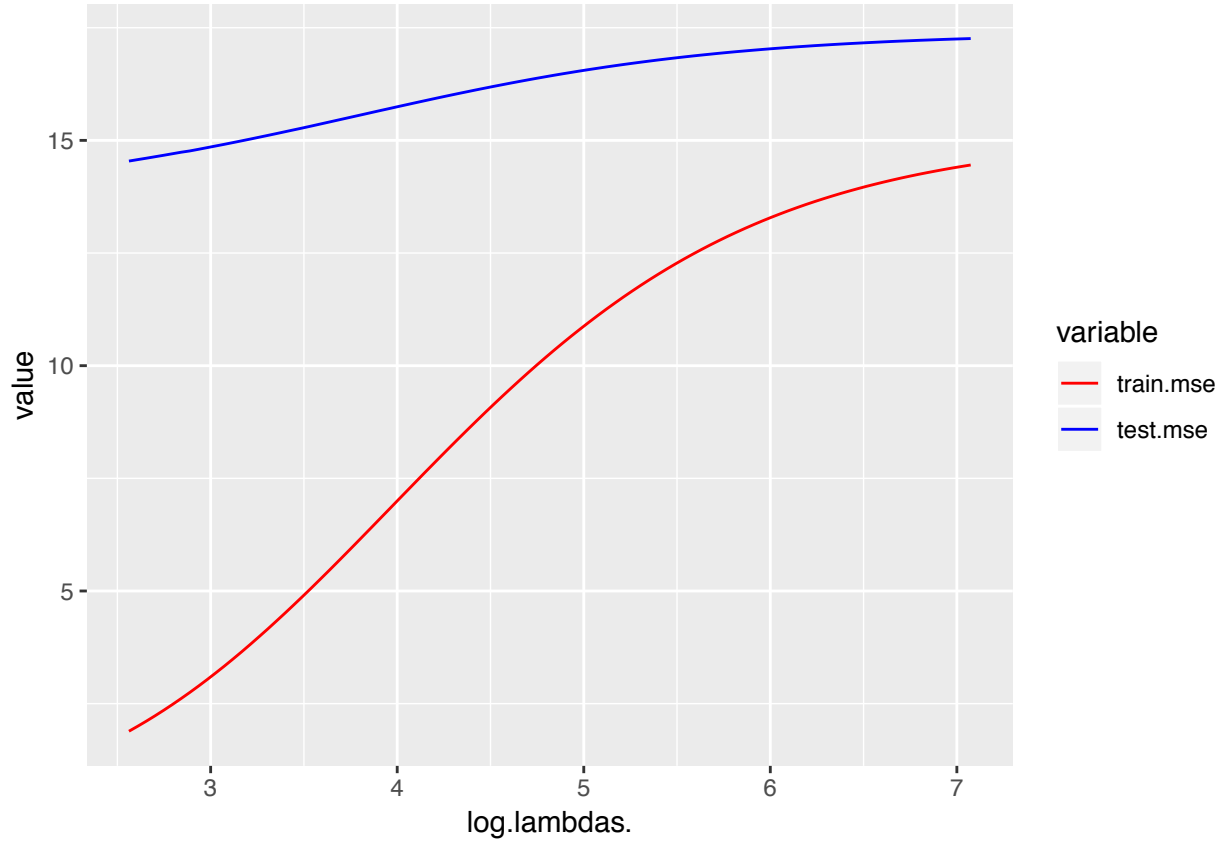
```
n=1000
d=5000
x = matrix(rnorm(5000*1000, mean = 0, sd = 1),nrow = 1000)
err = rnorm(1000,0,sqrt(0.25))
beta = c(rep(1,15),rep(0,4985))
y = x %*% beta + err
```

#5.5

```
training = x[1:800,]
training.label = y[1:800]
test = x[801:1000,]
test.label = y[801:1000]
```

#5.6

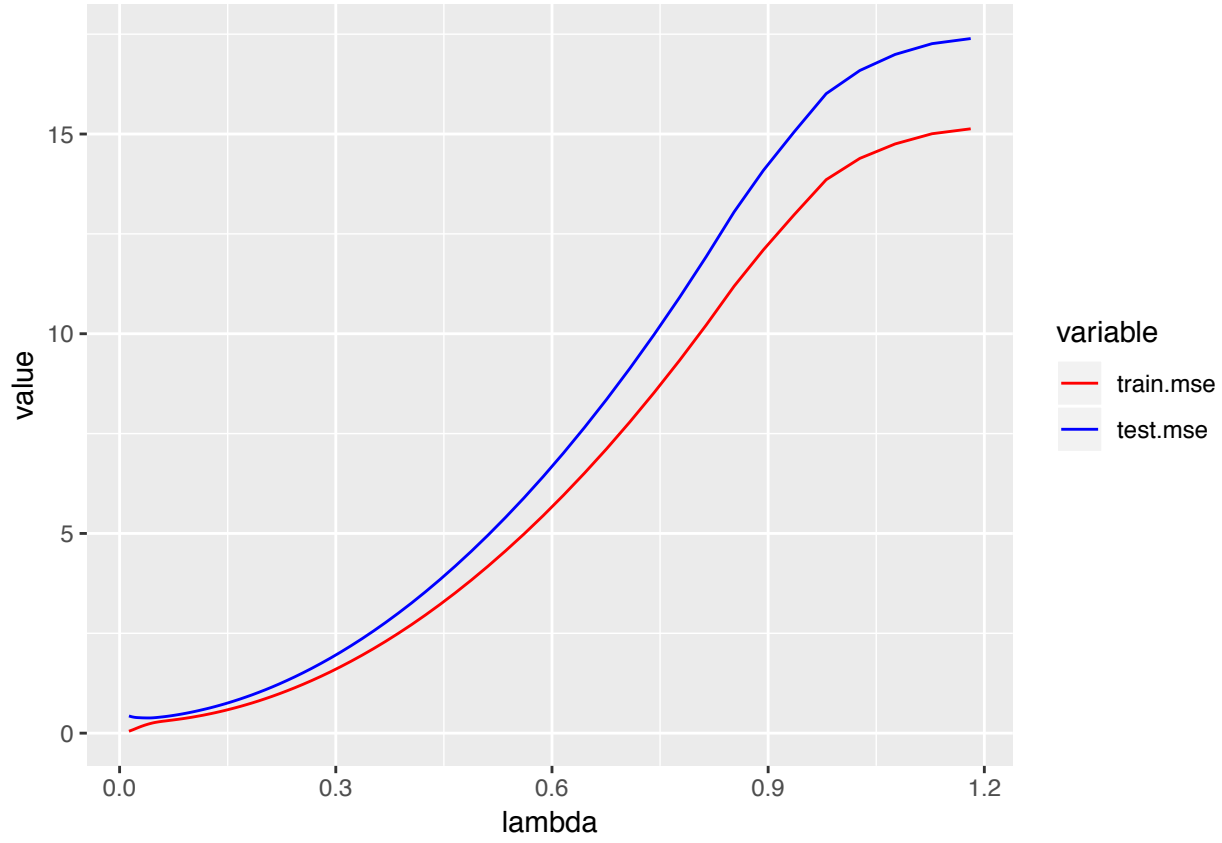
```
mod = cv.glmnet(training, training.label, alpha=0, type.measure = "mse")

lambdas = mod$lambda
training.error = rep(0,length(lambdas))
test.error = rep(0,length(lambdas))

for (i in 1:length(lambdas)) {
  fit.ridge= glmnet(training, training.label, alpha=0, lambda = lambdas[i])
  training.error[i] = mean((training.label - predict(fit.ridge, training))^2)
  test.error[i] = mean((test.label - predict(fit.ridge, test))^2)
}

dat <- data.frame("log(lambdas)" = log(lambdas), train.mse = training.error, test.mse = test.error)
dat.m <- melt(dat, id.vars = "log.lambdas.")
ggplot(dat.m, aes(log.lambdas., value, colour = variable)) +
  geom_line() +
  scale_colour_manual(values = c("red", "blue"))
```




```
ridge.mse = test.error[which(test.error == min(test.error))]
ridge.lambda.min = lambdas[which(test.error == min(test.error))]
```

#5.7

```
mod = cv.glmnet(training, training.label, alpha=1, type.measure = "mse")

lambdas = mod$lambda
training.error = rep(0,length(lambdas))
test.error = rep(0,length(lambdas))

for (i in 1:length(lambdas)) {
  fit.lasso= glmnet(training, training.label, alpha=1, lambda = lambdas[i])
  training.error[i] = mean((training.label - predict(fit.lasso, training))^2)
  test.error[i] = mean((test.label - predict(fit.lasso, test))^2)
}

dat <- data.frame(lambda = lambdas, train.mse = training.error, test.mse = test.error)
dat.m <- melt(dat, id.vars = "lambda")
ggplot(dat.m, aes(lambda, value, colour = variable)) +
  geom_line() +
  scale_colour_manual(values = c("red", "blue"))
```




```
lasso.mse = test.error[which(test.error == min(test.error))]
lasso.lambda.min = lambdas[which(test.error == min(test.error))]
```

#5.8

```
ridge.lambda.min
```

```
## [1] 12.96238
```

```
ridge.mse
```

```
## [1] 14.54114
```

```
lasso.lambda.min
```

```
## [1] 0.03958523
```

```
lasso.mse
```

```
## [1] 0.3805818
```

#6

#6.1-6.2

```
MSE = function(y, x, beta){
  sum((y-(x %*% beta))^2)/length(y)
}
R2 = function(y, x, beta){
  cor(y, x %*% beta)^2
}
```

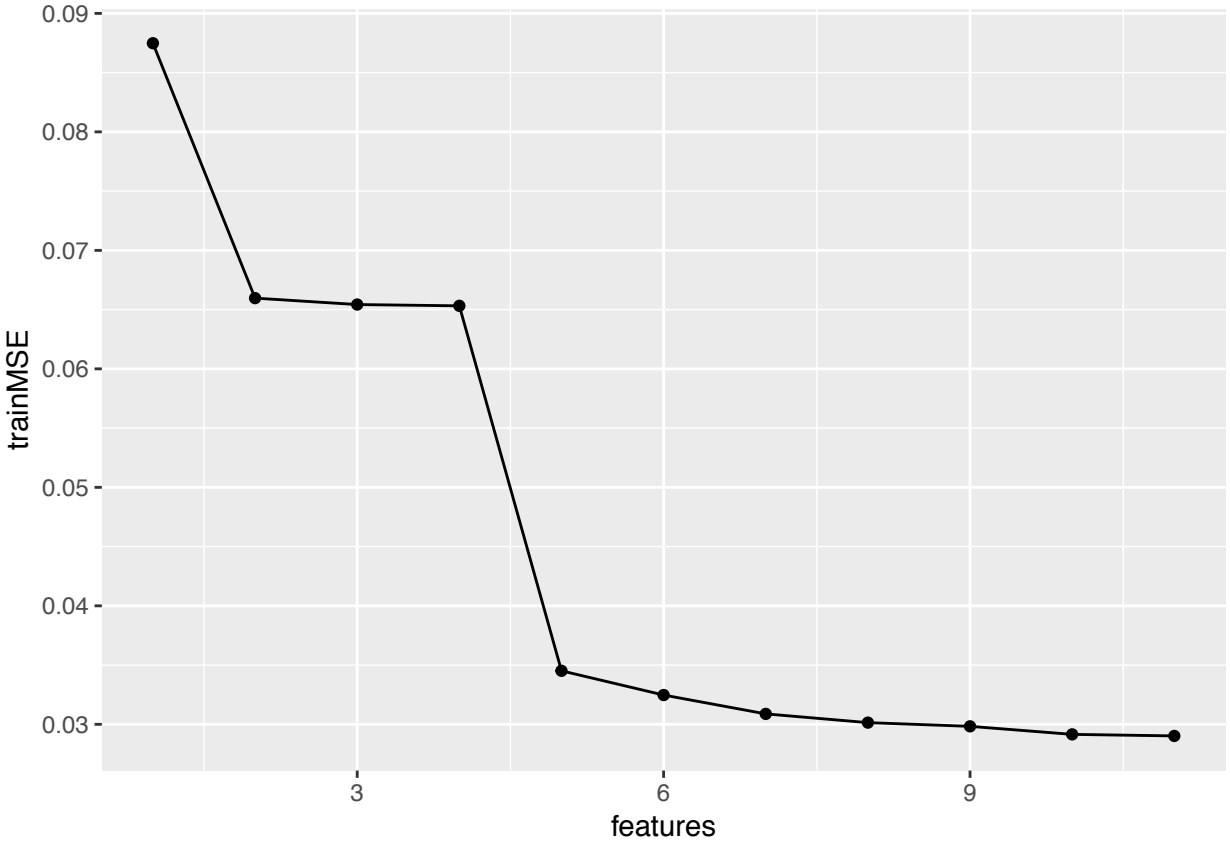
#6.3

```
test.mse= rep(0, 11)
modelQuality = matrix(rep(0,22),nrow=11)
modelQualityRImp =matrix(rep(0,22),nrow=11)
for (i in 1:length(Xnames)) {
  formula = as.formula(paste("log(SalePrice + 1) ~ ", paste(Xnames[1:i], collapse= "+")))
  lmod = lm(formula, data = AmesTinyTrain)
  training.label = log(AmesTinyTrain$SalePrice+1)
  mse = MSE(training.label, model.matrix(lmod),lmod$coefficients)
  r2 = R2(training.label,model.matrix(lmod),lmod$coefficients)
  modelQuality[i,1] = mse
  modelQuality[i,2] = r2
  r.out = summary(lmod)
  modelQualityRImp[i,1] = sum(r.out$residuals^2)/nrow(model.matrix(lmod))
  modelQualityRImp[i,2] = r.out$r.squared

  test.label = log(AmesTinyTest$SalePrice+1)
  test.mse[i] = sum((test.label-predict(lmod, AmesTinyTest))^2)/nrow(AmesTinyTest)
}

modelQuality = as.data.frame(modelQuality)
modelQualityRImp = as.data.frame(modelQualityRImp)
colnames(modelQuality) <- c("MSE", "R2")
colnames(modelQualityRImp) <- c("MSE", "R2")
modelQuality
```

```
##           MSE        R2
## 1  0.08748009 0.4745942
## 2  0.06596899 0.6037900
## 3  0.06542675 0.6070466
## 4  0.06531829 0.6076980
## 5  0.03451592 0.7926972
## 6  0.03247054 0.8049818
## 7  0.03087920 0.8145394
## 8  0.03014235 0.8189649
## 9  0.02982983 0.8208419
## 10 0.02915284 0.8249079
## 11 0.02902088 0.8257005
```

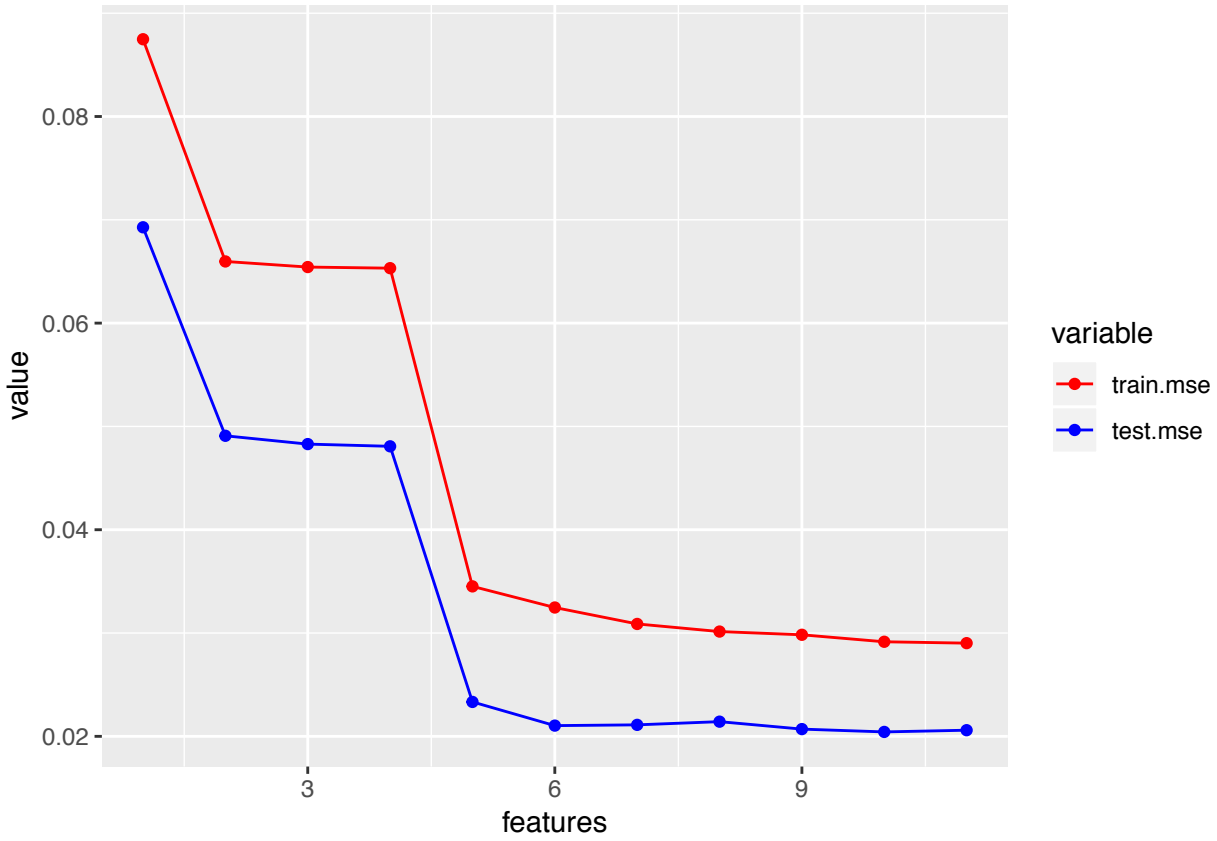
```
modelQualityRImp
```

```
##           MSE        R2
## 1  0.08748009 0.4745942
## 2  0.06596899 0.6037900
## 3  0.06542675 0.6070466
## 4  0.06531829 0.6076980
## 5  0.03451592 0.7926972
## 6  0.03247054 0.8049818
## 7  0.03087920 0.8145394
## 8  0.03014235 0.8189649
## 9  0.02982983 0.8208419
## 10 0.02915284 0.8249079
## 11 0.02902088 0.8257005
```



#6.4

the training MSE is decreasing as the number of predictors increase

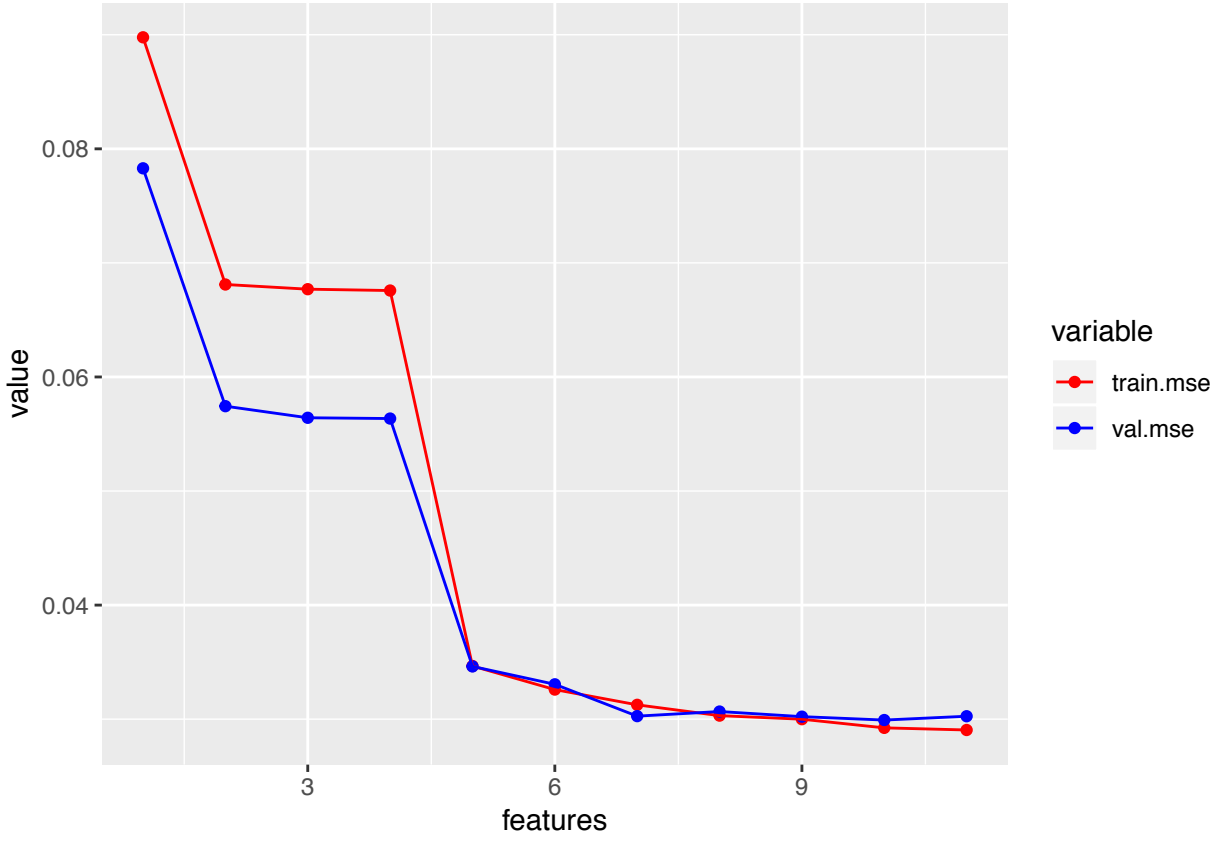


#6.5

test MSE is decreseing in [1:10] and starts increasing at 11 and the 10th model gives the lowest test MSE

#6.2.2

```
modelQualitySingleVal = matrix(rep(0,22),nrow=11)
for (i in 1:length(Xnames)) {
  formula = as.formula(paste("log(SalePrice + 1) ~ ", paste(Xnames[1:i], collapse= "+")))
  lmod = lm(formula, data = AmesTinyActTrain)
  training.label = log(AmesTinyActTrain$SalePrice+1)
  mse = MSE(training.label, model.matrix(lmod),lmod$coefficients)
  modelQualitySingleVal[i,1] = mse

  val.label = log(AmesTinyActVal$SalePrice+1)
  modelQualitySingleVal[i, 2] = sum((val.label-predict(lmod, AmesTinyActVal))^2)/nrow(AmesTinyActVal)
}

dat <- data.frame(features = 1:length(Xnames), train.mse = modelQualitySingleVal[,1], val.mse = modelQua
dat.m <- melt(dat, id.vars = "features")
ggplot(dat.m, aes(features, value, colour = variable)) +
  geom_point() + geom_line(aes(features, value, colour = variable))+
  scale_colour_manual(values = c("red", "blue"))
```




```
which(modelQualitySingleVal[,2] == min(modelQualitySingleVal[,2]))
```

```
## [1] 10
```

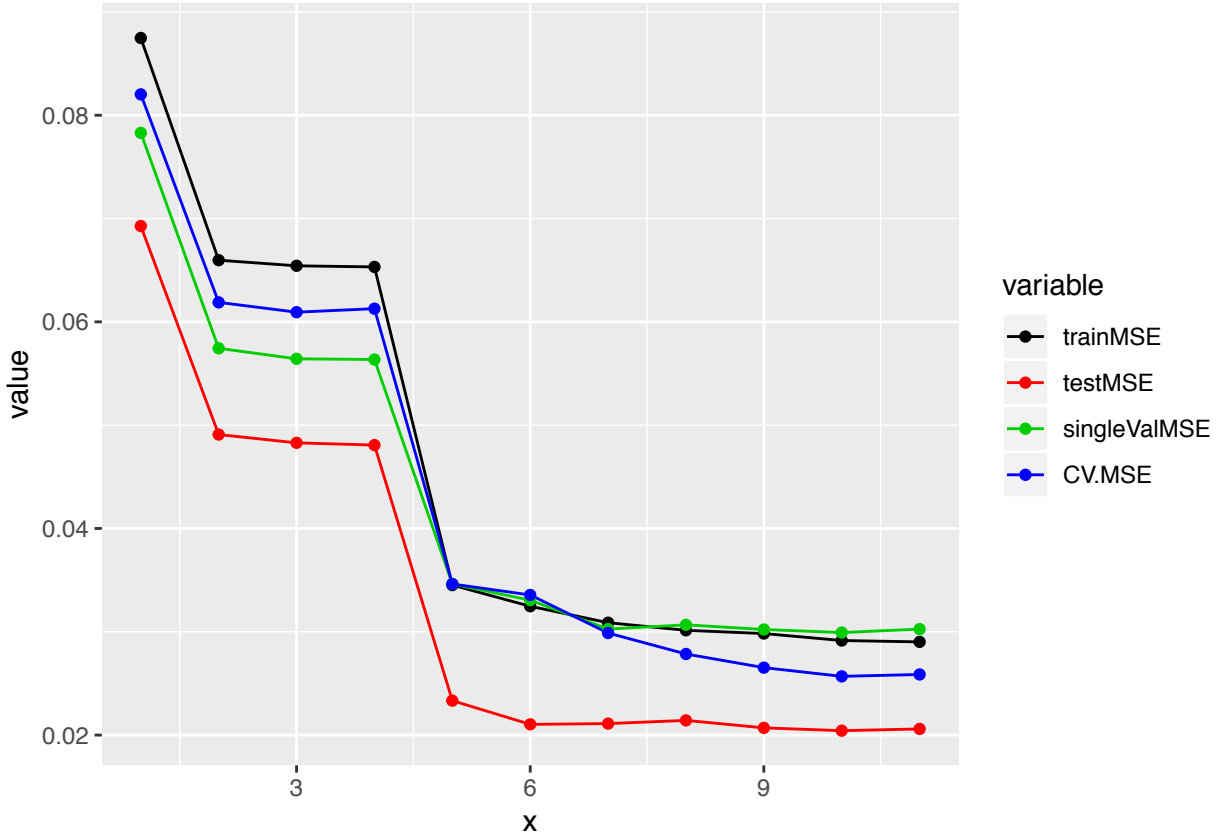
the 10th model gives the lowest validation MSE

#6.3.1

```
set.seed(123)
folds <- createFolds(AmesTinyTrain$SalePrice, k = 5)
cvMSE = matrix(rep(0,55),nrow=11)

for (i in 1:length(Xnames)) {
  formula = as.formula(paste("log(SalePrice + 1) ~ ", paste(Xnames[1:i], collapse= "+")))
  for (f in 1:5) {
    train = AmesTinyTrain[-folds[[f]], ]
    lmod = lm(formula, data = train)
    test = AmesTinyTrain[folds[[1]], ]
    pred<- predict(lmod, test)
    true_y<- log(test$SalePrice + 1)
    mse1 = 1/length(folds[[1]]) * sum((pred-true_y)^2)
    cvMSE[i,f] = mse1
  }
}
```

#6.3.2

```
dat <- data.frame(x = 1:length(Xnames), trainMSE = modelQuality$MSE, testMSE = test.mse, singleValMSE =
dat.m <- melt(dat, id.vars = "x")
ggplot(dat.m, aes(x, value, colour = variable)) +
  geom_line() + geom_point(aes(x, value, colour = variable))+
  scale_colour_manual(values = 1:5)
```




```
#10th model
# which(dat$CV.MSE == min(dat$CV.MSE))
```

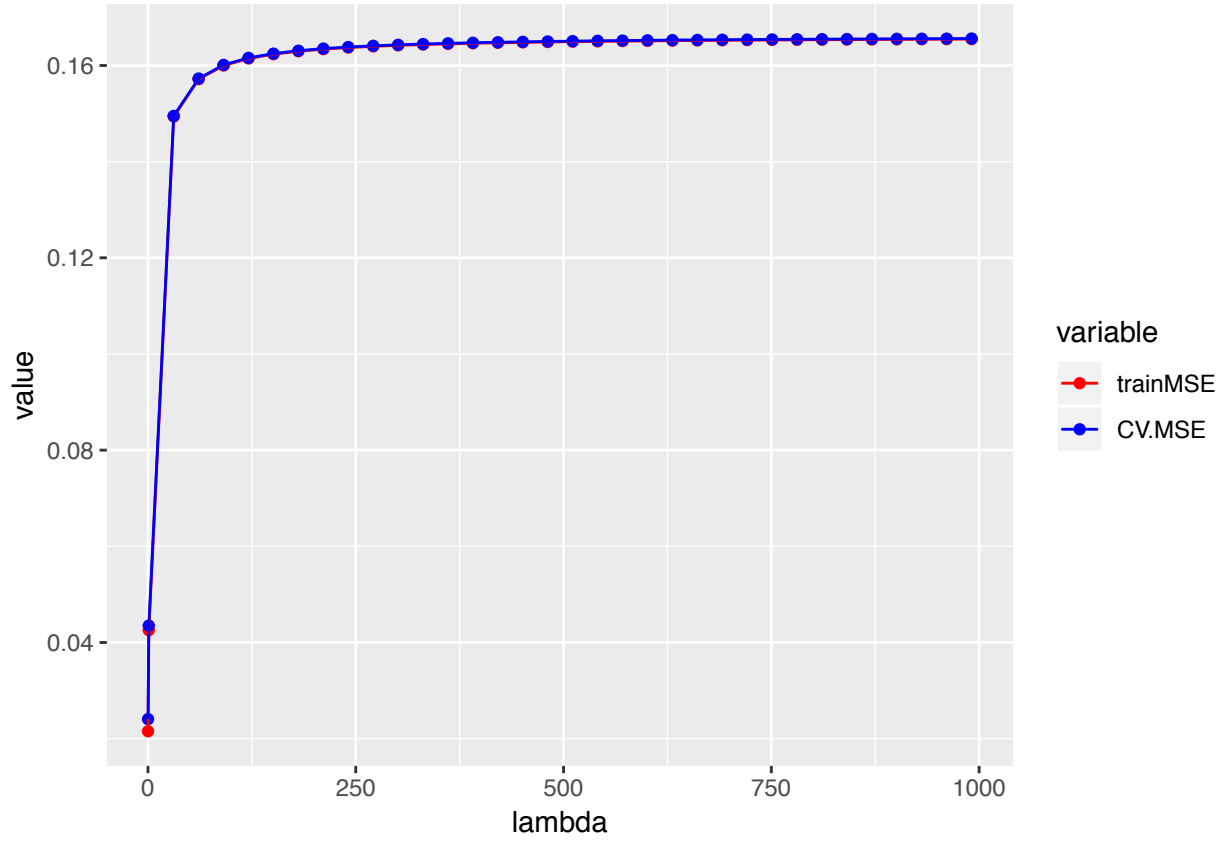
the 10th model gives the lowest CV-MSE

#6.4.1

```
train = AmesTiny[setdiff(names(AmesTiny), c("SalePrice"))]

x_train <- model.matrix( ~ .-1, train)
train.label = log(AmesTiny$SalePrice + 1)
mod.ridge = glmnet(x_train, log(AmesTiny$SalePrice + 1), alpha = 0, lambda = 1)
```

#6.4.2

```
lambdas = c(0.1, seq(1,1000,30))

training.error = rep(0,length(lambdas))
test.error = rep(0,length(lambdas))

for (i in 1:length(lambdas)) {
  fit.ridge= glmnet(x_train, train.label, alpha=0, lambda = lambdas[i])
  training.error[i] = mean((train.label - predict(fit.ridge, x_train))^2)
}

cvmod = cv.glmnet(x_train, train.label, alpha=0, lambda = lambdas, type.measure = 'mse',nfolds = 5)

dat <- data.frame(lambda = lambdas, trainMSE = training.error, CV.MSE = rev(cvmod$cvm))
dat.m <- melt(dat, id.vars = "lambda")
ggplot(dat.m, aes(lambda, value, colour = variable)) +
  geom_point() + geom_line(aes(lambda, value, colour = variable))+
  scale_colour_manual(values = c("red", "blue"))
```



when $\lambda = 0.1$, we have both minimum training MSE and CV-MSE