1. Consider Section 1.4 of the typed outline (concerning the variance-bias trade-off in prediction). Suppose that in a very simple problem with \( p = 1 \), the distribution \( P \) for the random pair \((x, y)\) is specified by

\[
   x \sim U(0, 1) \quad \text{and} \quad y \mid x \sim N(x^2, (1 + x))
\]

\((1 + x)\) is the conditional variance of the output). Further, consider two possible sets of functions \( S = \{g\} \) for use in creating predictors of \( y \), namely

1. \( S_1 = \{g \mid g(x) = a + bx \text{ for real numbers } a, b\} \), and

2. \( S_2 = \left\{ g \mid g(x) = \sum_{j=1}^{10} a_j I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right] \text{ for real number } a_j \right\} \)

Training data are \( N \) pairs \((x_i, y_i)\) iid \( P \). Suppose that the fitting of elements of these sets is done by

1. OLS (simple linear regression) in the case of \( S_1 \), and

2. according to

\[
   \hat{a}_j = \begin{cases} 
   \bar{y} & \text{if no } x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right] \\
   \frac{1}{\# x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right]} \sum_{i \text{ with } x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right]} y_i & \text{otherwise}
   \end{cases}
\]

in the case of \( S_2 \) to produce predictors \( \hat{f}_1 \) and \( \hat{f}_2 \).

**a)** Find (analytically) the functions \( g^* \) for the two cases. Use them to find the two expected squared model biases \( \mathbb{E}^x(\mathbb{E}[y \mid x] - g^*(x))^2 \). How do these compare for the two cases?

---

Note that \( \mathbb{E}[y \mid x] = x^2 \), and I need to find \( g^* = \arg \min_{g \in S} \mathbb{E}^x(E[y|x] - g(x))^2 \).

1. **(2 pts)** For \( S_1 \), I need to solve \( a \) and \( b \) such that \( \mathbb{E}^x(x^2 - (a + bx))^2 \) is minimized where \( x \sim U(0, 1) \). Let

\[
   h(a, b) \equiv \mathbb{E}^x(x^2 - (a + bx))^2 = \int_0^1 (x^2 - (a + bx))^2 \, dx = a^2 + \frac{1}{3}b^2 + \frac{1}{5} - \frac{2}{3}a - \frac{1}{2}b + ab.
\]
Set $\frac{\partial h}{\partial a} = 2a - \frac{2}{3} + b = 0$ and $\frac{\partial h}{\partial b} = \frac{2}{3}b - \frac{1}{2} + a = 0$, I get $a = -\frac{1}{6}$ and $b = 1$, and the determine of the second derivative is $\frac{1}{3} > 0$. So, $g_1^*(x) = -\frac{1}{6} + x$ and $\mathbb{E}^x(\mathbb{E}[y \mid x] - g_1^*(x))^2 = \frac{1}{180} \approx 0.0056$.

2. (2 pts) For $S_2$, I need to solve $a_j$ such that $\mathbb{E}^x \left( x^2 - \sum_{j=1}^{10} a_j I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right] \right)^2$ is minimized where $x \sim U(0, 1)$. This is equivalent to minimize $\int_{(j-1)/10}^{j/10} (x^2 - a_j)^2 dx$ for each $j$. Let

$$h(a_j) = \int_{(j-1)/10}^{j/10} (x^2 - a_j)^2 dx = a_j^2 \left( \frac{j}{10} - \frac{j-1}{10} \right) - \frac{2}{3}a_j((\frac{j}{10})^3 - (\frac{j-1}{10})^3) + \frac{1}{5}((\frac{j}{10})^5 - (\frac{j-1}{10})^5).$$

Set $\frac{\partial h}{\partial a_j} = 2a_j \left( \frac{j}{10} - \frac{j-1}{10} \right) - \frac{2}{3}a_j((\frac{j}{10})^3 - (\frac{j-1}{10})^3) = 0$, I get $a_j = \frac{10}{3}((\frac{j}{10})^3 - (\frac{j-1}{10})^3)$ and $\frac{\partial^2 h}{\partial a_j^2} = \frac{1}{5} > 0$. So, $g_2^*(x) = \frac{10}{3} \sum_{j=1}^{10} \left( (\frac{j}{10})^3 - (\frac{j-1}{10})^3 \right) I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right]$ and $\mathbb{E}^x(\mathbb{E}[y \mid x] - g_2^*(x))^2 \approx 0.0011$.

(1 pt) The term $\mathbb{E}^x(\mathbb{E}[y \mid x] - g^*(x))^2$ is the average squared model bias and ideally $g^*(x)$ gives the smallest bias to the conditional mean function $\mathbb{E}[y \mid x]$ among all other elements of $S$. From the above calculation, the $g^*$ of $S_2$ has smaller bias than the one of $S_1$ on average. (0.0011 < 0.0056)

b) For the second case, find an analytical form for $\mathbb{E}^T \hat{f}_2$ and then for the average squared estimation bias $\mathbb{E}^x(\mathbb{E}^T \hat{f}_2(x) - g_2^*(x))^2$. (Hints: What is the conditional distribution of the $y_i$ given that no $x_i \in (\frac{i-1}{10}, \frac{i}{10}]$? What is the conditional mean of $y$ given that $x \in (\frac{i-1}{10}, \frac{i}{10}]$?)

Note that the question gave the predictor $\hat{f}_2(x) = \sum_{j=1}^{10} \hat{a}_j I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right]$ where $x$ is a new observation differ to the training set implicit in the notation of $\hat{f}_2$. Let $R_j = (\frac{i-1}{10}, \frac{i}{10}]$ for $j = 1, \ldots, 10$. 

---

2011-01-27

Homework 1

STAT602X

Keys

R Output

```
R> j <- 1:10
R> a.j <- 10 / 3 * ((j / 10)^3 - ((j - 1) / 10)^3)
R> h.a.j <- a.j^2 * ((j / 10)^3 - ((j - 1) / 10)^3) + 2 / 3 * a.j * ((j / 10)^3 - ((j - 1) / 10)^3) + 1/5 * ((j/10)^5 - ((j - 1)/10)^5)
R> j <- 1:10
R> sum(h.a.j)
[1] 0.00110889
```

(1 pt) The term $\mathbb{E}^x(\mathbb{E}[y \mid x] - g^*(x))^2$ is the average squared model bias and ideally $g^*(x)$ gives the smallest bias to the conditional mean function $\mathbb{E}[y \mid x]$ among all other elements of $S$. From the above calculation, the $g^*$ of $S_2$ has smaller bias than the one of $S_1$ on average. (0.0011 < 0.0056)
(i) For $E^T \hat{f}_2$, we define an indicator $I_j = I(\text{no } x_i \in R_j)$ which is a Bernoulli random variable, 1 with probability $\left( \frac{9}{10} \right)^N$, and 0 with probability $1 - \left( \frac{9}{10} \right)^N$. In the following, I use a notation $f(\cdots)$ for the pdf or pmf depending on the arguments.

1. (2 pts) For case $I_j = 1$, I have

\[
E^T (y_i \mid I_j = 1) = \int \int y_i f(y_i, x \mid I_j = 1) dy_i dx
\]

\[
= \int_0^1 \int_{-\infty}^\infty y_i f(y_i \mid x, I_j = 1) f(x \mid I_j = 1) dy_i dx_i
\]

\[
= \int_0^1 f(x_i \mid I_j = 1) \int_{-\infty}^\infty y_i f(y_i \mid x_i) dy_i dx_i
\]

\[
= \int_{[0,1]\setminus R_j} \frac{10}{9} x_i^2 dx_i
\]

\[
= \frac{10}{27} \left( 1 - \left( \frac{4}{10} \right)^3 + \left( \frac{4}{10} \right)^3 \right).
\]

Then, $E^T (y \mid I_j = 1) = \frac{1}{N} \sum_{i=1}^N E^T (y_i \mid I_j = 1) = \frac{10}{27} \left( 1 - \left( \frac{4}{10} \right)^3 + \left( \frac{4}{10} \right)^3 \right)$.

2. (2 pts) For case $I_j = 0$, I have

\[
E (y \mid I_j = 0) = \int \int y f(y, x \mid I_j = 0) dy dx
\]

\[
= \int_0^1 \int_{-\infty}^\infty y f(y \mid x, I_j = 0) f(x \mid I_j = 0) dy dx
\]

\[
= \int_0^1 f(x \mid I_j = 0) \int_{-\infty}^\infty y f(y \mid x) dy dx
\]

\[
= \int_{R_j} 10 x^2 dx
\]

\[
= \frac{10}{3} \left( \left( \frac{4}{10} \right)^3 - \left( \frac{4}{10} \right)^3 \right).
\]

Then, $E^T \left( \frac{1}{\# x_i \in R_j} \sum_{x_i \in R_j} y_i \mid I_j = 0 \right) = \frac{1}{\# x_i \in R_j} \sum_{x_i \in R_j} E^T (y_i \mid I_j = 0) = E^T (y \mid I_j = 0) = \frac{10}{3} \left( \left( \frac{4}{10} \right)^3 - \left( \frac{4}{10} \right)^3 \right)$.

(1 pts) Combining all above, for all $j$,

\[
E^T (\hat{a}_j) = P^T (I_j = 1) E^T (\hat{a}_j \mid I_j = 1) + P^T (I_j = 0) E^T (\hat{a}_j \mid I_j = 0)
\]

\[
= \left( \frac{9}{10} \right)^N \frac{10}{27} \left( 1 - \left( \frac{4}{10} \right)^3 + \left( \frac{4}{10} \right)^3 \right) + \left( 1 - \left( \frac{9}{10} \right)^N \right) \frac{10}{3} \left( \left( \frac{4}{10} \right)^3 - \left( \frac{4}{10} \right)^3 \right).
\]

Then, $E^T \hat{f}_2 (x) = \sum_{j=1}^{10} E^T (\hat{a}_j) I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right]$.

(ii) (2 pts) Let $I_j (x) = I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right] \sim\text{ Bernoulli} \left( \frac{1}{10} \right)$. From the answer of the part a),
we have $g_2^*(x) = \frac{10}{3} \sum_{j=1}^{10} \left( \left( \frac{j}{10} \right)^3 - \left( \frac{j-1}{10} \right)^3 \right) I_j(x)$, so for $N = 50,$

$$E^x((E^T \hat{f}_2(x) - g_2^*(x))^2) = E^x \left[ \sum_{j=1}^{10} \left( \left( \frac{j}{10} \right)^3 - \left( \frac{j-1}{10} \right)^3 \right) I_j(x) \right]^2$$

$$= \sum_{j=1}^{10} E^x \left[ \left( \left( \frac{j}{10} \right)^3 - \left( \frac{j-1}{10} \right)^3 \right) I_j(x) \right]^2$$

$$= 2.878469 \times 10^{-6}$$

where $E^x I_j(x)^2 = \frac{1}{10}$ and $E^x I_j(x) I_k(x) = 0$ for all $j \neq k.$

R Output

```r
> j <- 1:10
> N <- 50
> p.N <- (9 / 10)^N
> d.rl <- (j / 10)^3 - ((j - 1) / 10)^3
> # E.a.j <- (p.N * (1 - d.rl) + (1 - p.N) * d.rl) / 3
> sum((E.a.j - 10 / 3 * d.rl)^2 / 10)
[1] 2.878469e-06
```

c) For the first case, simulate at least 1000 training data sets of size $N = 50$ and do OLS on each one to get corresponding $\hat{f}$'s. Average those to get an approximation for $E^T \hat{f}_1.$ (If you can do this analytically, so much the better!) Use this approximation and analytical calculation to find the average squared estimation bias $E^x(E^T \hat{f}_1(x) - g_1^*(x))^2$ for this case.

(2 pts) Let $E^T \hat{f}_1(x) \equiv \tilde{a} + \tilde{b}x$ where $\tilde{a} = \frac{1}{1000} \sum_{k=1}^{1000} \alpha_k^{OLS} \approx -0.1661305$ and $\tilde{b} = \frac{1}{1000} \sum_{k=1}^{1000} \beta_k^{OLS} \approx 0.9989991$ are empirically estimated from simulations. From the answer of the part a), we have the analytic solution $g_1^*(x) = -\frac{1}{6} + x.$ Then,

$$E^x(E^T \hat{f}_1(x) - g_1^*(x))^2 = E^x \left( (\tilde{a} + \tilde{b}x) - (-\frac{1}{6} + x) \right)^2$$

$$= (\tilde{a} + \frac{1}{6})^2 + 2(\tilde{a} + \frac{1}{6})(\tilde{b} - 1)E^x x + (\tilde{b} - 1)^2 E^x x^2$$

$(\because E^x x = 1/2, E^x x^2 = 1/3)$

$$\approx 8.47644 \times 10^{-8}$$

R Output

```r
R> set.seed(60210113)
R> B <- 10000
R> N <- 50
```
R> ret <- matrix(0, nrow = B, ncol = 2)
R> for(b in 1:B){
  + data.simulated <- generate.pair.xy(N)
  + ret[b, ] <- lm(y ~ x, data = data.simulated)$coef
+ } 
R> (ret <- colMeans(ret))
[1] -0.1661305 0.9989991
[1] 8.47644e-08

d) How do your answers for b) and c) compare for a training set of size $N = 50$?

(1 pts) The quantity $E^x(\mathbb{E}^T \hat{f}(x) - g^*(x))^2$ measures how well the estimation method is between the predictor and the best estimating function $g^*(x)$ among all other elements of $\mathcal{S}$. $E^x(\mathbb{E}^T \hat{f}_1(x) - g_1^*(x))^2 \approx 8.47644 \times 10^{-8}$ is smaller than $E^x(\mathbb{E}^T \hat{f}_2(x) - g_2^*(x))^2 \approx 2.878469 \times 10^{-6}$ indicating that the estimation method (OLS) for $\mathcal{S}_1$ yields smaller difference than the method (empirical average on 10 equal intervals) for $\mathcal{S}_2$ when sample size $N = 50$. When the sample size is small, this will lead to large variation for $E^x(\mathbb{E}^T \hat{f}_1(x) - g_1^*(x))^2$. Also, the number of simulation affects the result, then the conclusion may differ.

e) (This may not be trivial, I haven’t completely thought it through.) Use whatever combination of analytical calculation, numerical analysis, and simulation you need to use (at every turn preferring analytics to numerics to simulation) to find the expected prediction variances $E^x\text{Var}^T(\hat{f}(x))$ for the two cases for training set size $N = 50$.

1. (2 pts) For $\hat{f}_1(x)$, I can generate $B = 1000$ training sets and each has size $N = 50$. I apply OLS to all training sets to obtain 1000 $a$’s and $b$’s estimations, then estimate $\text{Var}^T(\hat{a}), \text{Var}^T(\hat{b})x$ and $\text{Cov}^T(\hat{a}, \hat{b})$ by Monte Carlo. Therefore,

$$E^x\text{Var}^T(\hat{f}_1(x)) = E^x(\text{Var}^T(\hat{a}) + \text{Var}^T(\hat{b})x^2 + 2x\text{Cov}^T(\hat{a}, \hat{b}))$$

$$= \text{Var}^T(\hat{a}) + \frac{1}{3}\text{Var}^T(\hat{b}) + \text{Cov}^T(\hat{a}, \hat{b})$$

where $E^xx = 1/2$ and $E^xx^2 = 1/3$. From the simulation output, I get $E^x\text{Var}^T(\hat{f}_1(x)) \approx 0.06052453$.

2. (2 pts)
For \( \hat{f}_2(x) \), I can apply similar steps as \( \hat{f}_1 \). Let \( I_j(x) = I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right] \) which is constant to the measure of \( T \), so

\[
\mathbb{E}^x \text{Var}^T(\hat{f}_2(x)) = \mathbb{E}^x \text{Var}^T(\sum_{j=1}^{10} \hat{a}_j I_j(x))
\]
\[
= \mathbb{E}^x \left( \sum_{j=1}^{10} I_j(x)^2 \text{Var}^T(\hat{a}_j) + 2 \sum_{j<k} I_j(x) I_k(x) \text{Cov}^T(\hat{a}_j, \hat{a}_k) \right)
\]
\[(I_j(x) \sim \text{Bernoulli}(1/10)) \Rightarrow \text{Var}^T(\hat{a}_j) = \frac{1}{10} \sum_{j=1}^{10} \text{Var}^T(\hat{a}_j)
\]

Therefore, I only need the Monte Carlo to estimate \( \text{Var}^T(\hat{a}_j) \) for all \( j \). From the simulation output, I get \( \mathbb{E}^x \text{Var}^T(\hat{f}_2(x)) \approx 0.3777918 \).

\[
\text{R Output}
\]
\[
\text{R> generate.pair.xy <- function(N)}\{\text{+ x <- runif(N)}\}
\text{+ y <- rnorm(N, mean = x^2, sd = sqrt(1 + x))}\text{+ data.frame(x = x, y = y)}\text{+}\} \text{R> estimate.a.j <- function(da.org){}\text{+ a.j <- rep(mean(da.org$y), 10)}\text{+ for(j in 1:10){}\text{+ id <- da.org$x > (j - 1) / 10 & da.org$x <= j / 10}\text{+ if(any(id)){}\text{+ a.j[j] <- mean(da.org$y[id])}\text{+}\} }\text{+ a.j}\text{+}\} \text{R> set.seed(60210115)} \text{R> B <- 1000} \text{R> N <- 50} \text{R> ret.f.1 <- matrix(0, nrow = B, ncol = 2)} \text{R> ret.f.2 <- matrix(0, nrow = B, ncol = 10)} \text{R> for(b in 1:B){}\text{+ data.simulated <- generate.pair.xy(N)}\text{+ ret.f.1[b, ] <- lm(y ~ x, data = data.simulated)$coef}\text{+ ret.f.2[b, ] <- estimate.a.j(data.simulated)}\text{+}\} \text{R> var(ret.f.1[, 1]) + var(ret.f.1[, 2]) / 3 + cov(ret.f.1[, 1], ret.f.1[, 2])}\text{[1] 0.06052453} \text{R> mean(apply(ret.f.2, 2, var))}\text{[1] 0.3777918}
\]

f) In sum, which of the two predictors here has the best value of Err for \( N = 50 \)?
(1 pts) From the lecture note, I have

\[
\text{Err} \equiv \mathbb{E}^x \text{Err}(x) = \mathbb{E}^x \text{Var}^T \hat{f}(x) + \mathbb{E}^x (\mathbb{E}^T \hat{f}(x) - \mathbb{E}[y \mid x])^2 + \mathbb{E}^x \text{Var} y \mid x].
\]

The forth term \( \mathbb{E}^x \text{Var} y \mid x \) is the same for two cases, then compare the sum of the first three terms is sufficient. From the previous calculation, I have

\[
\text{Err}_1 \approx 0.06062453 + 8.47644 \times 10^{-8} + 0.0056 + c = 0.06622461 + c
\]

and

\[
\text{Err}_2 \approx 0.377918 + 2.878469 \times 10^{-6} + 0.001108889 + c = 0.3789036 + c
\]

where \( c = 1.5 \) is a constant. The errors are dominated by the first term \( \mathbb{E}^x \text{Var}^T \hat{f}(x) \), and the \( \text{Err}_1 < \text{Err}_2 \) shows that the set of \( S_1 \) with OLS estimation provides a better prediction and on average has smaller error than the set of \( S_2 \) with empirical average estimation on 10 equal intervals.

2. Two files provided at http://thirteen-01.stat.iastate.edu/snoweye/stat602x/ with respectively 50 and then 1000 pairs \((x_i, y_i)\) were generated according to \( P \) in Problem 1. Use 10-fold cross validation to see which of the two predictors in Problem 1 looks most likely to be effective. (The data sets are not sorted, so you may treat successively numbered groups of 1/10th of the training cases as your \( K = 10 \) randomly created pieces of the training set.)

(10 pts) The two datasets were originally generated by

R Code

```r
set.seed(6021012)
generate.pair.xy <- function(N){
  x <- runif(N)
  y <- rnorm(N, mean = x^2, sd = sqrt(1 + x))
  data.frame(x = x, y = y)
}
data.hw1.Q2.set1 <- generate.pair.xy(50)
data.hw1.Q2.set2 <- generate.pair.xy(1000)
write.table(data.hw1.Q2.set1, file = "data.hw1.Q2.set1.txt", quote = FALSE, sep = "\t", row.names = FALSE)
write.table(data.hw1.Q2.set2, file = "data.hw1.Q2.set2.txt", quote = FALSE, sep = "\t", row.names = FALSE)
```

I use 10-fold cross-validation to compute "cross-validation error", given by the lecture
note,

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} \left( \hat{f}^{k(i)}(x_i) - y_i \right)^2$$

where $k(i)$ is the index of the piece $T_{k}$ training set. From the output given in the following, for $N = 50$, the

$\text{err}_1 = 1.911028$ and $\text{err}_2 = 2.776087$.

For $N = 1000$, the

$\text{err}_1 = 1.545203$ and $\text{err}_2 = 1.562407$.

The errors decrease as $N$ increases and the case $S_1$ is better choice. As the same size increases, the OLS results $S_1$ may be slightly better than the results of the empirical average $calS_2$ after $N = 10000$, and this conclusion may depend on the number of partitions on $[0, 1]$.

10–fold cross validateion

<table>
<thead>
<tr>
<th>N</th>
<th>$\text{err}_1$</th>
<th>$\text{err}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>1.911028</td>
<td>2.776087</td>
</tr>
<tr>
<td>1000</td>
<td>1.545203</td>
<td>1.562407</td>
</tr>
<tr>
<td>5000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R Output

```r
R> estimate.a.j <- function(da.org){
+   a.j <- rep(mean(da.org$y), 10)
+   for(j in 1:10){
+     id <- da.org$x > (j - 1) / 10 & da.org$x <= j / 10
+     if(any(id)){
+       a.j[j] <- mean(da.org$y[id])
+     }
+   }
```

...
3. (This is another calculation intended to provide intuition in the direction that shows data in \( \mathcal{Y}^p \) are necessarily sparse.) Let \( F_p(t) \) and \( f_p(t) \) be respectively the \( \chi^2_p \) cdf and pdf. Consider the MVN\(_p(0, I)\) distribution and \( Z_1, Z_2, \ldots, Z_N \) iid with this distribution. With

\[
M = \min\{\|Z_i\| \mid i = 1, 2, \ldots, N\}
\]

Write out a one-dimensional integral involving \( F_p(t) \) and \( f_p(t) \) giving \( \mathbb{E}M \). Evaluate this mean for \( N = 100 \) and \( p = 1, 5, 10, \) and 20 either numerically using simulation.
(10 pts) Given the question setting, $||Z_i||^2 \sim \chi^2_p$ has cdf $F_p(t)$ and pdf $f_p(t)$ where $t > 0$ is an observed value of $||Z_i||^2$. From the order statistics, the pdf of $M$ is $f_M(m) = Nf_p(m^2)(1 - F_p(m^2))^{N-1}|J|$ where $|J| = 2m$. Hence, the mean of $M$ is

$$
E_M = \int_0^\infty 2m^2Nf_p(m^2)(1 - F_p(m^2))^{N-1}dm.
$$

This can be done by Monte Carlo or numerical intergration.

```
R> set.seed(6021013)
R> B <- 1000
R> DF <- c(1, 5, 10, 20, 50)
R> ret.100 <- matrix(0, nrow = B, ncol = length(DF))
R> for(df in 1:length(DF)){
+   for(b in 1:B){
+     ret.100[b, df] <- sqrt(min(rchisq(100, DF[df])))
+   }
+ }
R> ret.5000 <- matrix(0, nrow = B, ncol = length(DF))
R> for(df in 1:length(DF)){
+   for(b in 1:B){
+     ret.5000[b, df] <- sqrt(min(rchisq(5000, DF[df])))
+   }
+ }
R> data.frame(p = DF, mean.100 = colMeans(ret.100), mean.5000 = colMeans(ret.5000))
p  mean.100  mean.5000
1 1 0.01220739 0.0002436844
2 5 0.67931648 0.3014548833
3 10 1.52105314 0.9698842615
4 20 2.75772516 2.1245447477
5 50 5.32947136 4.6139640221
```

From the output, we can see that the mean of the minimum distance of $N = 100$ observations to the orange is increased as $p$ increases. The mean of $p = 20$ is almost 50 times than the mean of $p = 1$. This quantity $E_M$ should go to 0 as $N \to \infty$, and from the output it probably needs more samples as $p$ increases. $\mathbb{R}^p$ is sparse.

Total: 40 pts (Q1: 20, Q2: 10, Q3: 10). Any resonable solutions are acceptable. Some results may be different due to Monte Carlo simulation.