Homework 2

Ze Yang (zey@andrew.cmu.edu)

Due Thursday, November 9 at 3:00 PM

You should submit the Rmd and pdf file for Question 1. You should also submit a pdf file with your responses to Questions 2 through 4. There is nothing wrong with handwritten solutions; I am not asking you to learn Latex to complete the homework.

Please do not submit photos of your homework. Scanners are available for your use.

Question 1:

Run a **both** K-means and hierarchical clustering algorithm on the yield curve shift data that was considered back when PCA was introduced. Discuss anything interesting that you find. You are free to make decisions regarding settings to the algorithms as you see fit.

```
# Reproducibility
set.seed(42)


# download data
fullYCweb =read_html("https://goo.gl/j97141")
tvdnodes =html_nodes(fullYCweb, ".text_view_data")
tableelements =html_text(tvdnodes)
tableelements[grep("N/A", tableelements)] = NA
tableelements =html_text(tvdnodes)

YCdata =matrix(tableelements, ncol=12,byrow=TRUE)
YCdata =data.frame(YCdata, stringsAsFactors=FALSE)
names(YCdata) =c("Date","1mo","3mo","6mo","1yr",
                 "2yr","3yr","5yr","7yr","10yr",
                 "20yr","30yr")
rates.names = c("1mo","3mo","6mo","1yr",
                "2yr","3yr","5yr","7yr","10yr",
                "20yr","30yr")
YCdata$Date =as.Date(YCdata$Date,format="%m/%d/%y")
YCdata[,2:12] =apply(YCdata[,2:12],2,as.numeric)
# only consider data after 2010-01-01
YCrates = YCdata[YCdata$Date > "2010-01-01", -1]
YCrates = YCrates[-which(apply(YCrates,1,sum) == 0),]
# compute rates shift
YCshifts =apply(YCrates,2,diff)
```

```r
# construct rates shift dataframe
df.yc = as.data.frame(YCshifts[complete.cases(YCshifts),])
df.yc.dates = YCdata[row.names(df.yc),]$Date
df.yc$Date = df.yc.dates
str(df.yc)
```

```
## 'data.frame':    1964 obs. of  12 variables:
##  $ 1mo : num  -0.02 0 -0.01 0 -0.01 0.01 0 0 0.01 0 ...
##  $ 3mo : num  -0.01 -0.01 -0.01 0 -0.01 0.01 0.01 -0.01 0.01 0 ...
##  $ 6mo : num  -0.01 -0.02 0.01 -0.01 -0.02 ...
##  $ 1yr : num  -0.04 -0.01 0 -0.03 -0.02 ...
##  $ 2yr : num  -0.08 0 0.02 -0.07 -0.01 ...
##  $ 3yr : num  -0.09 0.03 0.02 -0.06 -0.01 ...
##  $ 5yr : num  -0.09 0.04 0.02 -0.05 0.01 ...
##  $ 7yr : num  -0.08 0.05 0 -0.02 0.01 ...
##  $ 10yr: num  -0.08 0.08 0 -0.02 0.02 ...
##  $ 20yr: num  -0.06 0.09 -0.01 -0.01 0.03 ...
##  $ 30yr: num  -0.06 0.11 -0.01 0.01 0.04 ...
##  $ Date: Date, format: "2010-01-05" "2010-01-06" ...
```

```r
# scaling & assign kmeans cluster
df.yc.scale = apply(df.yc[1:11], 1, scale)
km = kmeans(t(df.yc.scale), centers=5, nstart=10)
df.yc$cluster.km = factor(km$cluster)
```

```r
# calculate distance, then use diffusion map to get a 2-dimension view
distance = dist(t(df.yc.scale))
diffusion.map = diffuse(distance, eps.val=50, t=10)
```

```
## Performing eigendecomposition
## Computing Diffusion Coordinates
## Used default value: 2 dimensions
## Elapsed time: 6.491 seconds
```

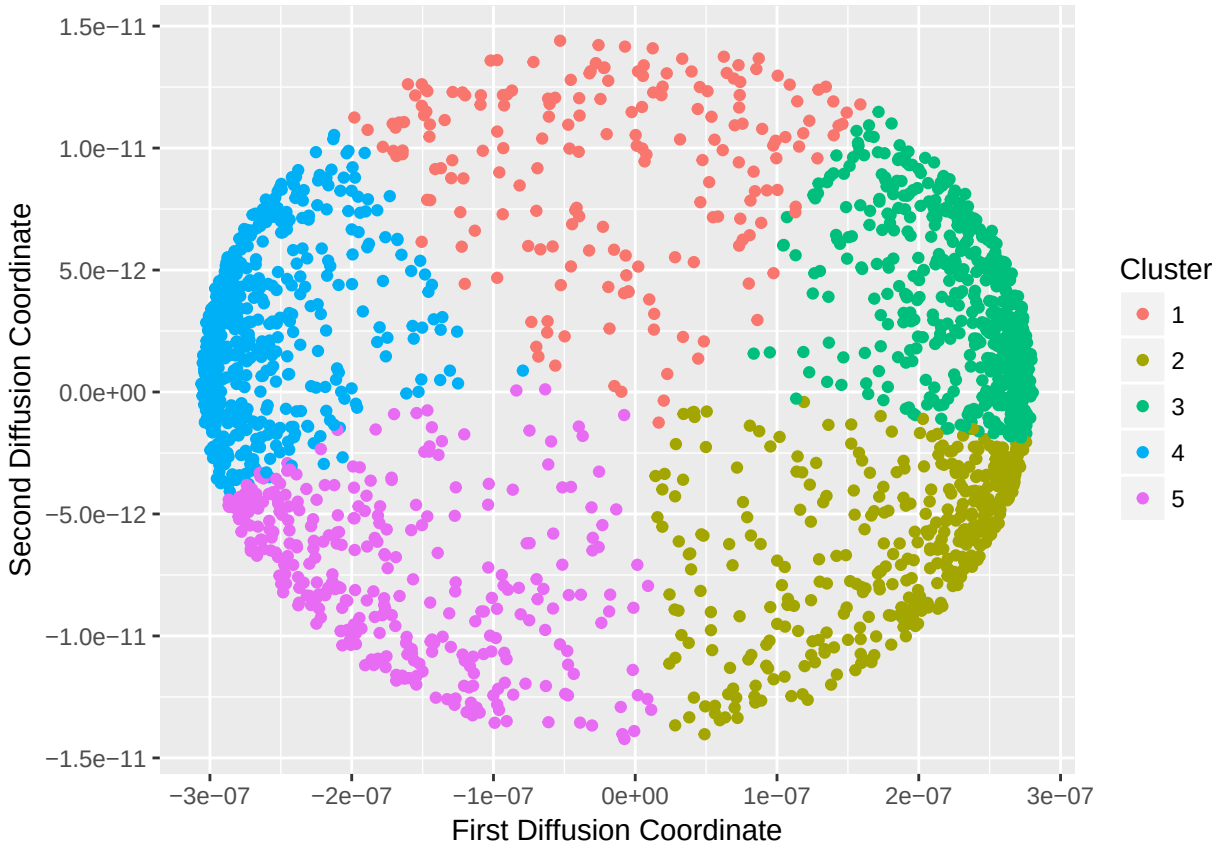
```r
df.yc$dmap1 = diffusion.map$X[,1]
df.yc$dmap2 = diffusion.map$X[,2]

# assign an id to each observation
# also assign another id when sorted by clusters
df.yc$id = seq(nrow(df.yc))
df.yc$id.sort = df.yc$id[order(df.yc$cluster.km)]
df.yc$id.sort = order(df.yc$id.sort)

tmp.df = cbind(df.yc[,1:11], df.yc$id, df.yc$id.sort)
tmp.df.scale = as.data.frame(cbind(t(df.yc.scale), df.yc$id, df.yc$id.sort))
colnames(tmp.df) = c(rates.names, 'id', 'id.sort')
colnames(tmp.df.scale) = c(rates.names, 'id', 'id.sort')
```

```
df.yc.melt = melt(tmp.df, id.vars=c("id", "id.sort"))
df.yc.scale.melt = melt(tmp.df.scale, id.vars=c("id", "id.sort"))
```

```
# plot 2-dimensional representation with cluster coloring
ggplot(df.yc,aes(x=dmap1,y=dmap2,color=cluster.km)) +
  geom_point() +
  labs(x="First Diffusion Coordinate",
       y="Second Diffusion Coordinate", color="Cluster")
```

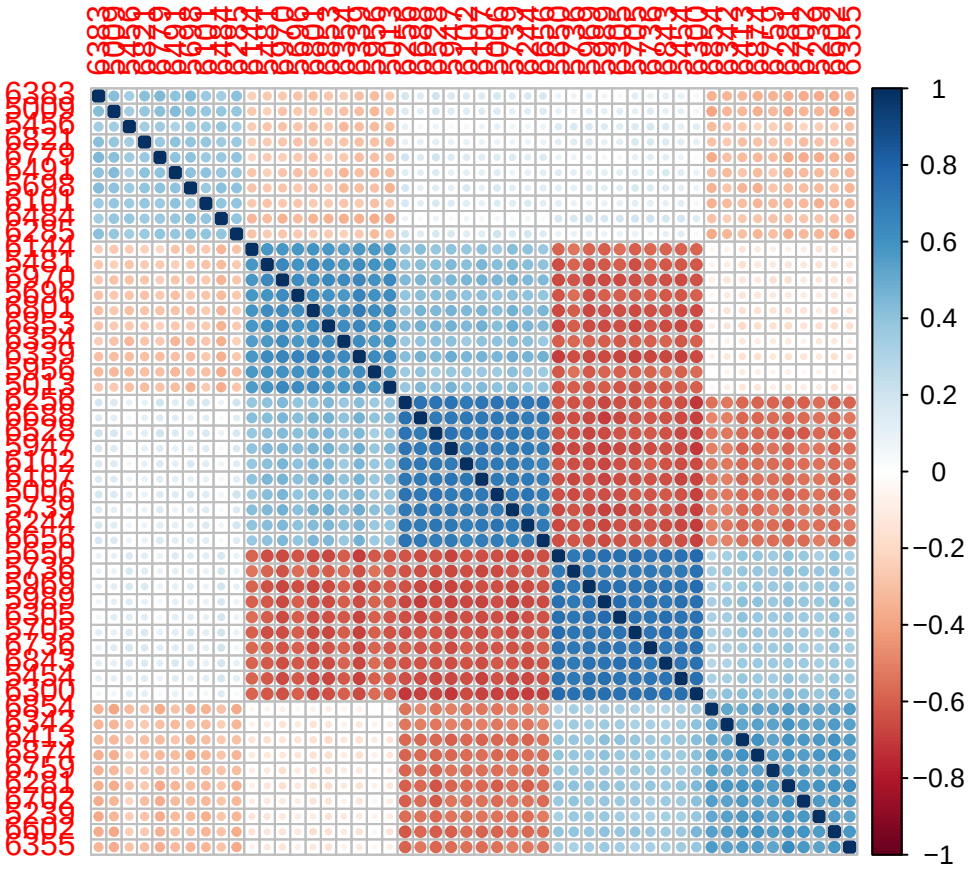



```
# draw sample from every classes
km.sample = function(df.yc, group, size=20) {
  g = df.yc[df.yc$cluster.km==group, 1:11]
  g.sample = g[sample(nrow(g), size),]
  return(g.sample)
}

# And examine the correlation plot of the sample
avg.cor.mat = matrix(rep(0, 50*50), 50, 50)
N = 100
for (i in 1:N) {
  rnd.sample = t(df.yc[sample(nrow(df.yc), 40), 1:11])
  yc.sample.1 = t(km.sample(df.yc, 1, 10))
  yc.sample.2 = t(km.sample(df.yc, 2, 10))
```

```
  yc.sample.3 = t(km.sample(df.yc, 3, 10))
  yc.sample.4 = t(km.sample(df.yc, 4, 10))
  yc.sample.5 = t(km.sample(df.yc, 5, 10))
  yc.sample = cbind(yc.sample.1, yc.sample.2,
                    yc.sample.3, yc.sample.4, yc.sample.5)
  cor.mat = cor(yc.sample)
  avg.cor.mat = avg.cor.mat + cor.mat/N
}


corrplot(avg.cor.mat)
```



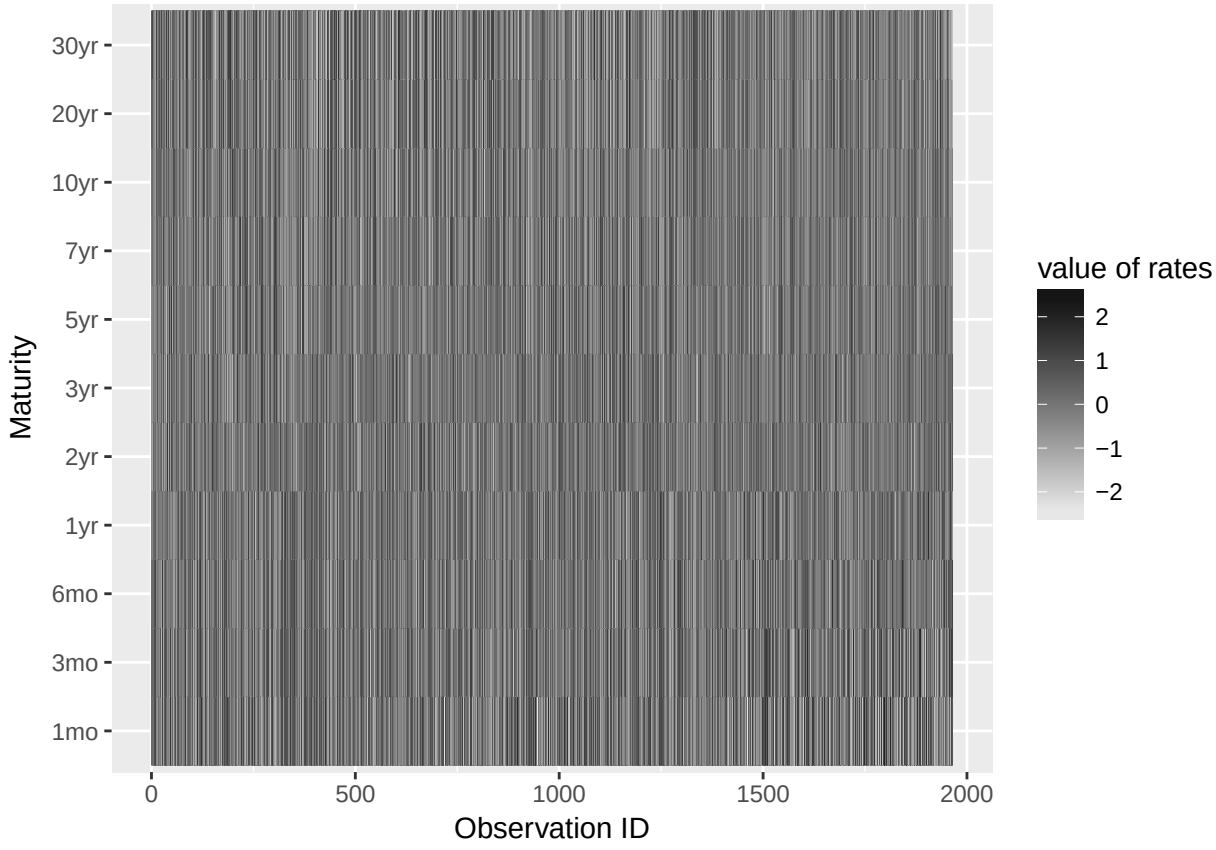
Discussion:

- The plot above is a visualization of the correlation matrix of the yield curves within our sample. (see library `corrplot`)
- We draw 10 observations within each cluster, and align them together. Hence we get a 50-by-50 correlation matrix, which can also be viewed at a 5-by-5 matrix of "blocks", Where each block is a 10-by-10 matrix.
- Clearly, the 5 blocks along the diagonal of the bigger correlation matrix represents the correlation matrix within every single group.
- We run a Monte-Carlo simulation by draw 100 samples, for each sample we calculate the correlation matrix. Finally we look at the average of these correlation matrices, and

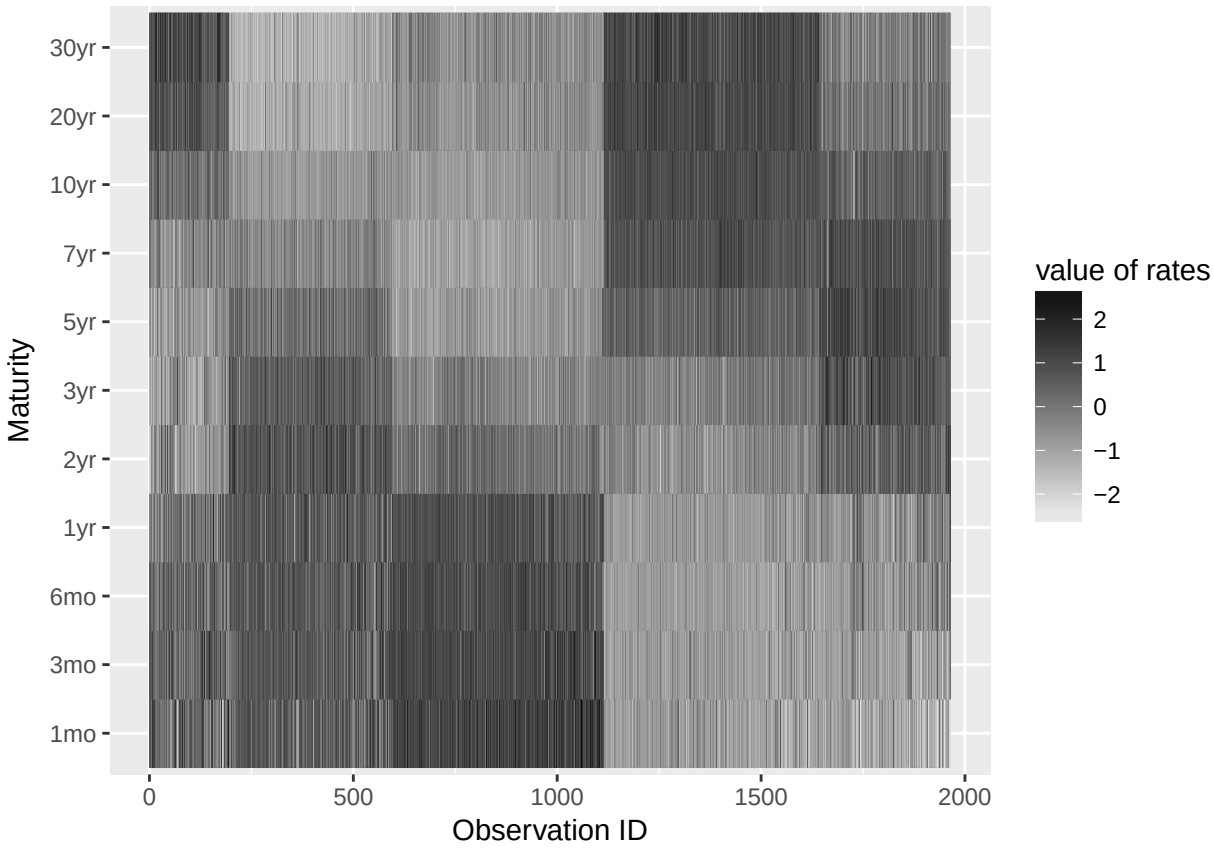
use this as a crude estimate of `correlation` among our 5 groups.

- Indeed, we can observe that the correlation is higher along the diagonal (the blocks along the diagonal), especially for group 2, 3, and 4.
- Therefore, we can conclude that kmeans cluster captures the similarities between the shape of the yield curves, as measured by the corrlation coefficient.

```
# plot the heatmap of original dataset
ggplot(df.yc.scale.melt,
       aes(x=id, y=variable)) +
  geom_tile(aes(colour=value)) +
  scale_colour_gradient(low = "white", high = "black") +
  labs(x="Observation ID",
       y="Maturity", color="value of rates")
```



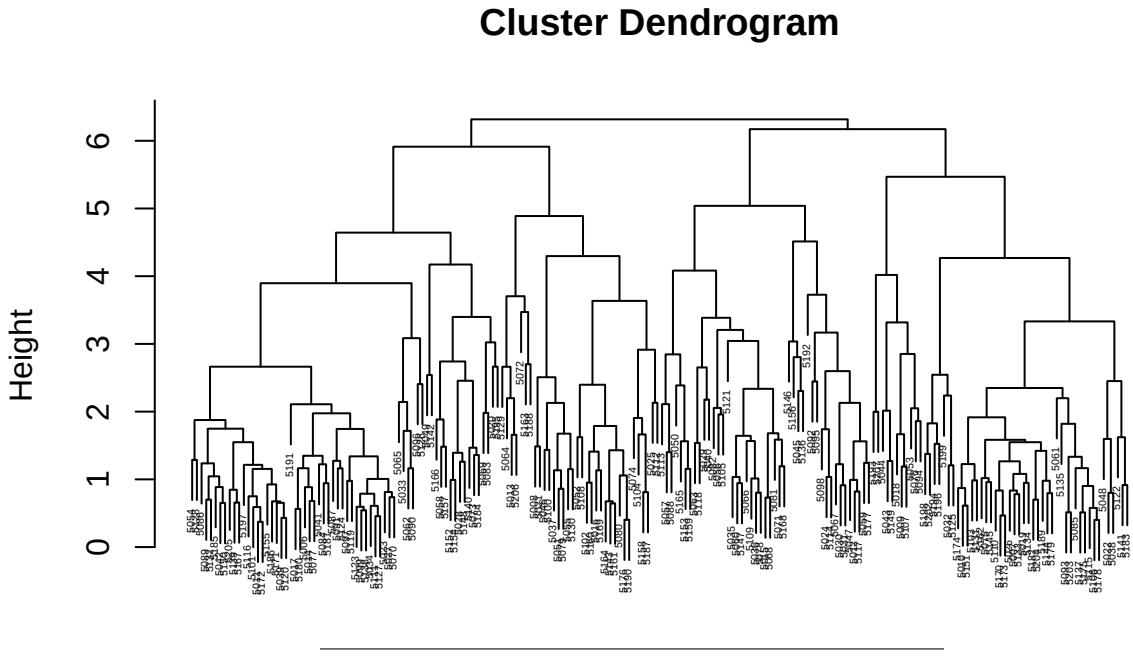

```
# plot the heatmap of the dataset with kmeans grouping
ggplot(df.yc.scale.melt,
       aes(x=id.sort, y=variable)) +
  geom_tile(aes(colour=value)) +
  scale_colour_gradient(low = "white", high = "black") +
  labs(x="Observation ID",
       y="Maturity", color="value of rates")
```



Discussion (2):

- Now we look at the dataset again by the heatmap plot (see the two plots above).
- The row of the heatmap represents a certain maturity, while the column represents observation. In the first plot, we order the observations by their initial id, in the second one, we order such that observations from same cluster are aligned together.
- Indeed, the first heatmap looks quite random.
- In the second heatmap, we can observe patterns. In particular, yield curves within same cluster are more similar to each other. We can also observe the shape of yield curve by looking at the color of heatmap: dark color means high yield, light color means low yield. We may roughtly conclude that:
- Yield curves in the 1st cluster are likely to be of "U-shape"
- Yield curves in the 2nd cluster are likely to be between 3rd and 1st.
- Yield curves in the 3rd cluster are likely to be "downward sloping".
- Yield curves in the 4th cluster are likely to be "upward sloping".
- Yield curves in the 5th cluster are likely to be of "Inverse-U-shape".

```
# hierarchical clustering
hc = hclust(dist(t(df.yc.scale[,1:200])), method = "complete")
plot(hc, cex=0.35, sub="", xlab="")
```



Question 2:

Suppose that $X_1, X_2, \ldots, X_n$ are iid with the Gamma(α, β) distribution. Determine the method of moments estimators for α and β.

Comment: This is an important case, because the "standard" approach to constructing estimators, maximum likelihood, does not admit a closed form for the estimators for α and β.

Solution:

Since $\{X_i\}_1^n \sim$ i.i.d $\Gamma(\alpha, \beta)$, we can find the *population moments* from the table of common distributions' sheet in probability course.

$$\begin{aligned} \mathbb{E}[X] &= \alpha\beta \\ \mathbb{V}\mathrm{ar}[X] &= \alpha\beta^2 \\ \mathbb{E}\left[X^2\right] &= \mathbb{V}\mathrm{ar}[X] + \mathbb{E}[X]^2 = \alpha\beta^2 + \alpha^2\beta^2 \end{aligned} \tag{1}$$

Denote M_1, M_2 the first and second sample moments:

$$\begin{aligned} M_1 &= \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \\ M_2 &= \frac{1}{n}\sum_{i=1}^{n} X_i^2 \end{aligned} \tag{2}$$

Following the method of moments, we align polulation moments to sample moments:

$$\begin{cases} \alpha\beta = M_1 \\ \alpha\beta^2 + \alpha^2\beta^2 = M_2 \end{cases} \tag{3}$$

Rewrite the second equation as $\alpha\beta(\beta + \alpha\beta) = M_1(\beta + M_1)$, we have $M_1(\beta + M_1) = M_2$. Hence

$$\hat{\beta}_{mm} = \frac{M_2}{M_1} - M_1 = \frac{M_2 - M_1^2}{M_1} \tag{4}$$

Note that $\sum_{i=1}^n X_i = n\bar{X} = \sum_{i=1}^n \bar{X}$:

$$\begin{aligned} M_2 - M_1^2 &= \frac{1}{n}\sum_{i=1}^n X_i^2 - \bar{X}^2 = \frac{1}{n}\left(\sum_{i=1}^n X_i^2 - n\bar{X}^2\right) \\ &= \frac{1}{n}\left(\sum_{i=1}^n X_i^2 - 2n\bar{X}^2 + n\bar{X}^2\right) \\ &= \frac{1}{n}\left(\sum_{i=1}^n X_i^2 - 2\bar{X}\sum_{i=1}^n X_i + \sum_{i=1}^n \bar{X}^2\right) \\ &= \frac{1}{n}\sum_{i=1}^n \left(X_i^2 - 2\bar{X}X_i + \bar{X}^2\right) \\ &= \frac{1}{n}\sum_{i=1}^n \left(X_i - \bar{X}\right)^2 = s_X^2 \quad (\textit{sample variance}) \end{aligned} \tag{5}$$

Therefore, $\hat{\beta}_{mm} = \frac{s_X^2}{M_1}$, and $\hat{\alpha}_{mm} = \frac{M_1}{\hat{\beta}_{mm}} = \frac{M_1^2}{s_X^2}$. In summary:

$$\hat{\alpha}_{mm} = \frac{M_1^2}{s_X^2} = \frac{\bar{X}^2}{\frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})^2} = \frac{n\bar{X}^2}{\sum_{i=1}^n (X_i - \bar{X})^2} \tag{6}$$

$$\hat{\beta}_{mm} = \frac{s_X^2}{M_1} = \frac{\frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})^2}{\bar{X}} = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n\bar{X}} \tag{7}$$

Question 3:

We discussed the following result in lecture, now it is time to prove it. Define

$$\text{Mean Squared Error of estimator } \widehat{\theta} = \text{MSE}\left(\widehat{\theta}\right) = E\left[\left(\widehat{\theta} - \theta\right)^2\right]$$

Show that

$$\text{MSE}\left(\widehat{\theta}\right) = V(\widehat{\theta}) + \text{bias}^2(\widehat{\theta})$$

where

$$\text{bias}(\widehat{\theta}) = E(\widehat{\theta}) - \theta.$$

Hint: Start by writing

$$E\left[\left(\widehat{\theta} - \theta\right)^2\right] = E\left[\left(\left(\widehat{\theta} - E\left(\widehat{\theta}\right)\right) + \left(E\left(\widehat{\theta}\right) - \theta\right)\right)^2\right]$$

Note: In general, the expression for MSE($\widehat{\theta}$) will be a function of θ.

Proof:

$$\begin{aligned} MSE(\hat{\theta}) &= \mathbb{E}\left[(\hat{\theta}-\theta)^2\right] = \mathbb{E}\left[\Big((\hat{\theta}-\mathbb{E}[\hat{\theta}])+(\mathbb{E}[\hat{\theta}]-\theta)\Big)^2\right] \\ &= \mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right] + \mathbb{E}\left[2(\hat{\theta}-\mathbb{E}[\hat{\theta}])(\mathbb{E}[\hat{\theta}]-\theta)\right] + \mathbb{E}\left[(\mathbb{E}[\hat{\theta}]-\theta)^2\right] \quad (\dagger) \end{aligned} \tag{8}$$

Note that the only random variable in the formula above is $\hat{\theta} \setminus \mathbb{E}[\hat{\theta}], \theta$ are deterministic numbers, so is $(\mathbb{E}[\hat{\theta}]-\theta)$, hence it can be taken out of the expectations.

$$\begin{aligned} (\dagger) &= \mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right] + 2(\mathbb{E}[\hat{\theta}]-\theta)\mathbb{E}\left[\hat{\theta}-\mathbb{E}[\hat{\theta}]\right] + (\mathbb{E}[\hat{\theta}]-\theta)^2 \\ &= \mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right] + 2(\mathbb{E}[\hat{\theta}]-\theta)\cdot 0 + (\mathbb{E}[\hat{\theta}]-\theta)^2 \\ &= \mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right] + (\mathbb{E}[\hat{\theta}]-\theta)^2 \\ &= \mathbb{V}\text{ar}[\hat{\theta}] + \text{Bias}^2(\hat{\theta}) \end{aligned} \tag{9}$$

By defintion of bias and variance.

Question 4:

Suppose that $X_1, X_2, \ldots, X_n$ are independent random variables such that $\mu = E(X_i)$, i.e., each of the random variables has the same mean. But, $V(X_i) = \sigma_i^2$ is not necessarily constant across the random variables. It does hold that $0 < \sigma_i^2 < \infty$ for all i. You can assume that the σ_i^2 are known, but μ is unknown.

Consider an estimator of the form

$$\widehat{\mu} = \sum_{i=1}^{n} w_i X_i$$

where $0 \leq w_i \leq 1$.

a. Show that the estimator $\widehat{\mu}$ is an unbiased estimator of μ if $\sum_i w_i = 1$.
b. What value should be used for the w_i in order to keep $\widehat{\mu}$ unbiased but minimize the MSE?

Guidance: You should be able to show that

$$V(\widehat{\mu}) = \sum_{i=1}^{n} w_i^2 \sigma_i^2,$$

but in order to force the estimator to be unbiased, you need to incorporate the constraint that $\sum_i w_i = 1$. One way to do this is to replace w_n with

$$1 - \sum_{i=1}^{n-1} w_i.$$

Then, take the derivative of the variance with respect to w_i for $1 \leq i \leq n-1$ and set those equal to zero.

Proof:

(a) Since we have $\mathbb{E}[X_i] = \mu$ for each of $X_i, i = 1, 2, ..., n$:

$$\mathbb{E}[\hat{\mu}] = \mathbb{E}\left[\sum_{i=1}^{n} w_i X_i\right] = \sum_{i=1}^{n} w_i \mathbb{E}[X_i] = \mu \sum_{i=1}^{n} w_i \tag{10}$$

Hence $\hat{\mu}$ is unbiased $\iff \sum_{i=1}^{n} w_i = 1$. We write this in vector form: $\mathbf{1}^\top \boldsymbol{w} = 1.\backslash \sim \backslash$ **(b)** Let $\boldsymbol{w} := (w_1 \ \ ... \ \ w_n)^\top$, $\boldsymbol{X} := (X_1 \ \ ... \ \ X_n)^\top$,

$$\boldsymbol{\Sigma_X} := \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix}$$

Since $\hat{\mu}$ is an unbiased estimator,

$$\begin{aligned} MSE(\hat{\mu}) &= \mathbb{V}\text{ar}[\hat{\mu}] = \mathbb{V}\text{ar}[\boldsymbol{w}^\top \boldsymbol{X}] = \boldsymbol{w}^\top \boldsymbol{\Sigma_X} \boldsymbol{w} \\ &= \sum_{i=1}^{n} w_i^2 \sigma_i^2 \end{aligned} \tag{11}$$

We impose the constraint $\sum_{i=1}^{n} w_i = 1$, i.e. we solve the quadratic programming problem with linear constraint:

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{\Sigma_X} \boldsymbol{w} \\ \text{over} & x \in \mathbb{R}^n \\ \text{subject to} & \mathbf{1}^\top \boldsymbol{w} = 1 \end{array} \tag{12}$$

The Lagrangian dual of this QP is

$$\mathcal{L}(\boldsymbol{w}, \lambda) = \boldsymbol{w}^\top \boldsymbol{\Sigma_X} \boldsymbol{w} - \lambda(\mathbf{1}^\top \boldsymbol{w} - 1) \tag{13}$$

Take the first order condition:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}(\boldsymbol{w}, \lambda) = 2\boldsymbol{\Sigma_X} \boldsymbol{w} - \lambda \mathbf{1} = 0 \tag{14}$$

$$\Rightarrow \qquad \boldsymbol{w} = \frac{\lambda}{2} \boldsymbol{\Sigma_X}^{-1} \mathbf{1} = \frac{\lambda}{2} \begin{pmatrix} \frac{1}{\sigma_1^2} \\ \frac{1}{\sigma_2^2} \\ \vdots \\ \frac{1}{\sigma_n^2} \end{pmatrix} \tag{15}$$

Then plug into the constraint $\mathbf{1}^\top \boldsymbol{w} = 1 \;\Rightarrow$

$$\frac{\lambda}{2}\sum_{i=1}^{n}\frac{1}{\sigma_i^2} = 1 \quad \Rightarrow \quad \frac{\lambda}{2} = \frac{1}{\sum_{i=1}^{n}\frac{1}{\sigma_i^2}} \tag{16}$$

Hence:

$$\boldsymbol{w} = \frac{1}{\sum_{i=1}^{n}\frac{1}{\sigma_i^2}}\begin{pmatrix}\frac{1}{\sigma_1^2}\\ \frac{1}{\sigma_2^2}\\ \vdots \\ \frac{1}{\sigma_n^2}\end{pmatrix} = \frac{1}{\mathrm{tr}(\boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1})}\boldsymbol{\Sigma}_{\boldsymbol{X}}^{-1}\mathbf{1} \tag{17}$$

With

$$w_i = \frac{1}{\sum_{k=1}^{n}\frac{1}{\sigma_k^2}}\frac{1}{\sigma_i^2}, \quad i = 1, 2, ..., n$$