STAT 32950 Assignment 7

Bin Yu

May 19, 2025

Question 1

(a)

The R code to generate the data and fit the least-squares model is:

```
set.seed(42)

x1 <- rnorm(30)
x2 <- x1 + rnorm(30, sd = 0.01)
Y  <- rnorm(30, mean = 3 + x1 + x2)

ls_model <- lm(Y ~ x1 + x2)
summary(ls_model)
```

Output:

```
    Call:
lm(formula = Y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.4559 -0.4919  0.1632  0.5268  1.3745 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3.1721     0.1413  22.450   <2e-16 ***
x1           23.1768    13.8529   1.673    0.106    
x2          -21.3004    13.8761  -1.535    0.136    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7682 on 27 degrees of freedom
Multiple R-squared:  0.9136, Adjusted R-squared:  0.9072 
F-statistic: 142.8 on 2 and 27 DF,  p-value: 4.383e-15
```

The output gives estimated coefficients

$$\hat{\beta}_0 = 3.1721, \quad \hat{\beta}_1 = 23.1768, \quad \hat{\beta}_2 = -21.3004.$$

Hence the fitted model is

$$E(Y \mid x_1, x_2) = \hat{\beta}_0 + \hat{\beta}_1\, x_1 + \hat{\beta}_2\, x_2 = 3.1721 + 23.1768\, x_1 - 21.3004\, x_2.$$

(b)

The data were generated from the true model

$$Y \;=\; \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \;=\; 3 + 1 \cdot x_1 + 1 \cdot x_2 + \varepsilon,$$

so the true parameter values are

$$\beta_0^{\text{true}} = 3, \quad \beta_1^{\text{true}} = 1, \quad \beta_2^{\text{true}} = 1.$$

The least-squares estimates obtained in part (a) are

$$\hat{\beta}_0 = 3.1721, \quad \hat{\beta}_1 = 23.1768, \quad \hat{\beta}_2 = -21.3004.$$

Clearly $\hat{\beta}_1$ and $\hat{\beta}_2$ are far from the true value 1.

This poor performance is due to the near-perfect collinearity of the predictors ($x_2 \approx x_1$), which makes the design matrix almost singular and inflates the variance of the ordinary least squares estimators.

In fact, although the sum

$$\hat{\beta}_1 + \hat{\beta}_2 \approx 23.1768 - 21.3004 = 1.8764$$

is close to the true sum $1 + 1 = 2$, the individual slope estimates are highly unstable and not reliable under this multicollinearity.

(c)

Use the following R code to generate the RSS:

R Code:

```
rss_ls <- sum(resid(ls_model)^2)

y_hat_true <- 3 + 1 * x1 + 1 * x2
rss_true  <- sum((Y - y_hat_true)^2)

rss_ls
rss_true
```

Results:

$$\text{RSS}_{\text{LS}} = 15.93167, \qquad \text{RSS}_{\text{true}} = 18.88215.$$

Although these two quantities differ slightly, they are of the same order of magnitude and relatively close. In fact, the least-squares (LS) fit achieves a lower RSS than the true model on this particular data set, because LS chooses $\hat{\beta}$ to minimize the observed residuals, including fitting the realized noise—whereas the true parameters $(3, 1, 1)$ are not tailored to this sample's noise realization.

Why bad estimates can yield good predictions: The hat-matrix criterion cares only about $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i}$, not the individual $\hat{\beta}_j$. Here x_1 and x_2 are nearly collinear ($x_2 \approx x_1$), so the design matrix X is almost rank-deficient. Its column space is effectively one-dimensional, spanned by $x_1 + x_2$. Consequently, only the sum $\hat{\beta}_1 + \hat{\beta}_2$ is well-determined by the data; many pairs (β_1, β_2) with $\beta_1 + \beta_2 \approx 2$ yield essentially the same fitted values.

In our LS fit,

$$\hat{\beta}_1 + \hat{\beta}_2 = 23.1768 - 21.3004 = 1.8764 \approx 1 + 1 = 2,$$

so the model predicts $Y_i \approx 3 + 2\,x_{1i}$ (or equivalently $3 + 2\,x_{2i}$), which is very close to the true linear predictor. Thus, despite wildly incorrect individual slopes, the LS model captures the only identifiable direction $x_1 + x_2$ and so achieves low RSS and good predictive performance.

(d)

R Code:

```
library(MASS)
ridge_model <- lm.ridge(Y ~ x1 + x2, lambda = 1)
coef(ridge_model)
```

Output:

$$\begin{array}{lccc} \text{Coefficients at } \lambda = 1: & (\mathit{Intercept}) & x_1 & x_2 \\ & 3.1987525 & 0.9627315 & 0.9194409 \end{array}$$

The fitted ridge regression model is

$$E(Y \mid x_1, x_2) = \hat{\beta}_0^{\text{ridge}} + \hat{\beta}_1^{\text{ridge}}\,x_1 + \hat{\beta}_2^{\text{ridge}}\,x_2 = 3.1988 + 0.9627\,x_1 + 0.9194\,x_2.$$

The true parameters are $\beta_0 = 3$, $\beta_1 = 1$, $\beta_2 = 1$. Compared to the ordinary least squares estimates $(23.18, -21.30)$ from part (a), the ridge estimates $(0.9627,\ 0.9194)$ are much closer to the true slopes.

(e)

The LS estimator $(\hat{\beta}_0, \hat{\beta})$ minimizes the residual sum of squares

$$L_{\text{LS}}(\beta_0, \beta) \;=\; \sum_{i=1}^{n} \bigl(y_i - \beta_0 - x_{1i}\,\beta_1 - x_{2i}\,\beta_2\bigr)^2.$$

The ridge estimator $(\hat{\beta}_0, \hat{\beta})$ minimizes a penalized objective

$$L_{\text{ridge}}(\beta_0, \beta) \;=\; \sum_{i=1}^{n} \bigl(y_i - \beta_0 - x_{1i}\,\beta_1 - x_{2i}\,\beta_2\bigr)^2 \;+\; \lambda\bigl(\beta_1^2 + \beta_2^2\bigr),$$

with $\lambda = 1$ in our example.

Comparison:

- LS estimates (from part (a)): $\hat{\beta}_1 = 23.1768$, $\hat{\beta}_2 = -21.3004$, wildly far from the true slopes $(1, 1)$.
- Ridge estimates (with $\lambda = 1$, part (d)): $\hat{\beta}_1 = 0.9627$, $\hat{\beta}_2 = 0.9194$, much closer to $(1, 1)$.

Bias of the Ridge Estimator

The ridge estimator

$$\hat{\beta}^{\text{ridge}} = \left(X^\top X + \lambda I\right)^{-1} X^\top y$$

is generally biased, since

$$E[\hat{\beta}^{\text{ridge}}] = \left(X^\top X + \lambda I\right)^{-1} X^\top X\, \beta \neq \beta.$$

However, by adding the penalty matrix λI, $X^\top X + \lambda I$ is always invertible, even under exact collinearity. Its eigenvalues are $d_j + \lambda$, where d_j are those of $X^\top X$, so the amplification factors $1/(d_j + \lambda)$ remain bounded. With a suitably small λ, the induced bias is modest, while the reduction in variance—especially along directions with small d_j—can be dramatic.

Effect of the Ridge Penalty

By minimizing

$$\sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|^2,$$

ridge regression shrinks each coefficient toward zero. In the presence of near-collinearity ($x_2 \approx x_1$), ordinary least squares inflates variance along the nearly singular direction, whereas ridge suppresses the unstable (small-eigenvalue) components of β. It achieves a favorable bias–variance trade-off, recovering the true linear combination $x_1 + x_2$ and yielding slope estimates that are both stable and close to the true values.

Overall, ridge regression trades a small bias for a large variance reduction, often resulting in a lower mean squared error than OLS when predictors are nearly collinear.

Question 2

(a)

We fit a LASSO model for predicting `medv` using the first 300 observations as a training set and the remaining 206 as a calibration (validation) set. We use 10-fold cross-validation on the training data to select λ.

R Code

```
# Split data into training (first 300) and calibration (remaining 206)
Tdata <- Boston[1:300, ]
Cdata <- Boston[301:506, ]

X_train <- as.matrix(Tdata[, 1:13])
Y_train <- Tdata[, 14]

X_cal   <- as.matrix(Cdata[, 1:13])
Y_cal   <- Cdata[, 14]

set.seed(42)
cv_lasso <- cv.glmnet(
  X_train, Y_train
)

plot(cv_lasso)
```

```
lambda_min <- cv_lasso$lambda.min
lambda_1se <- cv_lasso$lambda.1se

print(lambda_min)
print(lambda_1se)
# Coefficients at _min and at _1se
coef_min <- coef(cv_lasso, s = "lambda.min")
coef_1se <- coef(cv_lasso, s = "lambda.1se")

print(coef_min)
print(coef_1se)

# Predict on calibration set using _min
Y_pred <- predict(cv_lasso, s = lambda_min, newx = X_cal)

# Compute calibration MSE
mse_cal <- mean((Y_cal - Y_pred)^2)
cat("Calibration MSE at _min:", round(mse_cal, 4), "\n")

Y_pred_1se <- predict(cv_lasso, s = lambda_1se, newx = X_cal)

# Compute calibration MSE
mse_1se <- mean((Y_cal - Y_pred_1se)^2)
cat("Calibration MSE at _1se:", round(mse_1se, 4), "\n")
```

Output:

```
> print(lambda_min)
[1] 0.02999934
> print(lambda_1se)
[1] 0.3369898

print(coef_min)
14 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept) -14.260071684
crim          0.900845410
zn            0.010063604
indus         0.002493744
chas          0.608781556
nox          -5.916962318
rm            9.166661636
age          -0.045510324
dis          -0.897515408
rad           0.105383202
tax          -0.013146679
ptratio      -0.630258858
black         0.015592280
lstat        -0.110318451
> print(coef_1se)
14 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept) -19.934523590
crim          .
zn            .
```

```
indus         .
chas          .
nox           .
rm            9.148319539
age          -0.016705129
dis          -0.222292832
rad           .
tax          -0.008258467
ptratio      -0.560766697
black         0.006476342
lstat        -0.124114819

Calibration MSE at _min: 228.6519

Calibration MSE at _1se: 63.3177
```

Results

$$\lambda_{\min} = 0.02999934, \qquad \lambda_{\text{1se}} = 0.3369898.$$

- At $\lambda_{\min}$, all 13 predictors enter the model:

  $(\hat{\beta}_0, \hat{\beta}_{\text{crim}}, \ldots, \hat{\beta}_{\text{lstat}}) = (-14.2601,\ 0.9008,\ 0.0101,\ 0.0025,\ 0.6088,\ -5.9170,\ 9.1667,\ -0.0455,\ -0.8975,\ 0.1054,\ -0.0131,$

- At the more conservative λ_{1se}, only 7 predictors remain nonzero:

  $$\{\texttt{rm}, \texttt{age}, \texttt{dis}, \texttt{tax}, \texttt{ptratio}, \texttt{black}, \texttt{lstat}\}.$$

Validation Performance

$$\text{MSE}_{\text{cal}}(\lambda_{\min}) = 228.65, \qquad \text{MSE}_{\text{cal}}(\lambda_{\text{1se}}) = 63.32.$$

Interpretation

- Although $\lambda_{\min}$ minimizes the cross-validation error on the training folds, it leads to a model that overfits the training noise, resulting in very high MSE on the calibration set.
- The "1-se" rule (λ_{1se}) selects a larger penalty, producing a sparser model that aggressively shrinks small coefficients to zero. This increases bias modestly but dramatically reduces variance. The more regularized model at λ_{1se} achieves far better predictive accuracy on unseen data (MSE reduced from 228.65 to 63.32).

Thus, if we restrict our evaluation to the model fit (as measured by calibration MSE) and the parameter estimates (their stability and sparsity):

The $\lambda_{\min}$ model, despite minimizing CV error on the training folds, performs poorly out-of-sample (MSE = 228.65) and yields unstable, nonzero estimates for all predictors. The λ_{1se} model, by contrast, achieves substantially better calibration fit (MSE = 63.32), selects only seven key variables, and produces much more stable coefficient estimates.

Hence, using the 1-SE rule strikes the best balance between bias and variance, indicating that when predicting, the important factors are rm, age, dis, tax, ptratio, black and lstat.

(b)

Using the following R code to compare LASSO with OLS regression model:

R Code:

```
# Ordinary least squares on training set
df_train <- as.data.frame(Tdata)
ols_model <- lm(medv ˜ ., data = df_train)
ols_coefs <- coef(ols_model)
print(ols_coefs)

# Calibration MSE for OLS
df_cal <- as.data.frame(Cdata)
Y_pred_ols <- predict(ols_model, newdata = df_cal)
mse_ols <- mean((Y_cal - Y_pred_ols)ˆ2)
cat("Calibration MSE (OLS):", round(mse_ols,4), "\n")

# Calibration MSE for LASSO at lambda.min and lambda.1se
cat("Calibration MSE (LASSO, lambda.min):", round(mse_cal,4), "\n")
cat("Calibration MSE (LASSO, lambda.1se):", round(mse_1se,4), "\n")

# LASSO coefficient paths
par(mfrow=c(1,1))
plot(log(cv_lasso$lambda), as.matrix(coef(cv_lasso))[-1, ], type="l",
     xlab="log(lambda)", ylab="Coefficient",
     main="LASSO Coefficient Paths")
```

The coefficient of OLS regression:

```
(Intercept)        crim          zn       indus        chas         nox          rm
-12.39946856  1.20125417  0.01484179  0.02355189  0.60206504 -8.82764179  9.13062230
         age         dis         rad         tax     ptratio       black       lstat
 -0.04735918 -1.01328587  0.16786579 -0.01456734 -0.64183404  0.01677883 -0.10976398
```

Results:

$$\text{Calibration MSE (OLS)} = 366.0655,$$
$$\text{Calibration MSE (LASSO, } \lambda_{\text{min}}) = 228.6519,$$
$$\text{Calibration MSE (LASSO, } \lambda_{\text{1se}}) = 63.3177.$$

Commentary: The OLS model, which uses all 13 predictors without shrinkage, achieves a high calibration MSE of 366.07, indicating severe overfitting to the training noise under multicollinearity. The LASSO model at λ_{min} improves calibration error (MSE = 228.65) by regularizing coefficients, but still retains all variables. The more conservative LASSO choice λ_{1se} yields the lowest MSE (63.32) by selecting only 7 predictors and shrinking the rest to zero, dramatically reducing variance and overfitting.

In sum, LASSO with an appropriate penalty markedly outperforms OLS in out-of-sample prediction and produces a sparse, more interpretable model.

Therefore, regularization via LASSO substantially improves out-of-sample prediction compared to ordinary least squares. While OLS overfits and yields a high validation error (MSE = 366), LASSO at λ_{min} already reduces error (MSE = 229) by shrinking coefficients, and the 1-SE rule further refines the model to just seven variables, achieving the lowest MSE (= 63). Thus, LASSO not only enhances predictive accuracy but also yields a parsimonious and interpretable set of predictors.

Question 3

(a)

We perform PCA on the eight hearing-loss measurements, centering and scaling each variable.

R code:

```
data <- read.csv("hearlossData.csv", header=FALSE)
colnames(data) <- c("Left5c","Left1k","Left2k","Left4k",
                    "Right5c","Right1k","Right2k","Right4k")

pca_hearloss <- prcomp(data, center=TRUE, scale.=TRUE)

summary(pca_hearloss)

print(pca_hearloss$rotation)
```

The output is:

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
Standard deviation     1.9822 1.2721 0.9876 0.68321 0.58317 0.56204 0.44734 0.39303
Proportion of Variance 0.4911 0.2023 0.1219 0.05835 0.04251 0.03949 0.02501 0.01931
Cumulative Proportion  0.4911 0.6934 0.8153 0.87368 0.91619 0.95568 0.98069 1.00000


               PC1        PC2         PC3        PC4         PC5         PC6         PC7
Left5c  -0.4010948 -0.3169638  0.15815686  0.3277576 -0.02313643 -0.44590406  0.32925525
Left1k  -0.4209908 -0.2254640 -0.05196134  0.4816310  0.37922678  0.06745818 -0.03312121
Left2k  -0.3663748  0.2385933 -0.47029298  0.2824293 -0.43924664  0.06379987 -0.52551666
Left4k  -0.2808559  0.4741545  0.42950248  0.1610807 -0.35031958  0.41692698  0.42694396
Right5c -0.3432510 -0.3860197  0.25931925 -0.4876003 -0.49750307 -0.19477711 -0.15935065
Right1k -0.4114209 -0.2317725 -0.02885395 -0.3723164  0.35131760  0.61363774 -0.08367787
Right2k -0.3115483  0.3170590 -0.56293305 -0.3914171  0.11078565 -0.26503010  0.47781584
Right4k -0.2542212  0.5135121  0.42622285 -0.1590982  0.39595899 -0.36604656 -0.41393534
                PC8
Left5c   0.54629986
Left1k  -0.62273890
Left2k   0.18634686
Left4k  -0.08393472
Right5c -0.34253018
Right1k  0.36136545
Right2k -0.14658750
Right4k  0.05082062
```

The first two principal components explain about 69.34% of the total variance (49.11% by PC1 and 20.23% by PC2), suggesting a strong low-dimensional structure. PC1 has roughly equal negative loadings on all eight variables (range -0.28 to -0.42), indicating it might capture an overall "hearing-loss severity" factor.

PC2 contrasts high-frequency loss (positive loadings on Left4k, Right4k, etc.) against low-frequency loss (negative loadings on Left5c, Right5c), accounting for about 20% of variability. Subsequent components (PC3–PC8) each explain only 6–12% or less, and their loadings mix frequencies in more complex ways, so interpretation beyond PC2 is less clear. The scree plot (not shown) would display an "elbow" at PC2 or PC3, reinforcing that the first two or three components captures most variation.

(b)

We first note from the standard PCA that the first two components explain approximately $0.4911 + 0.2023 = 0.6934$ (69.34%) of the total variance, so we set $K = 2$.

R code:

```
X <- scale(data)

# Sparse PCA with 2 components, each with 3 nonzero loadings
library(elasticnet)
set.seed(2025)
spca.res <- spca(X,
                 K      = 2,
                 type   = "predictor",
                 sparse = "varnum",
                 para   = c(3, 3))

print(spca.res$loadings)

# Compute proportion of variance explained (PVE)
scores     <- X %*% spca.res$loadings
p          <- ncol(X)
var_scores <- apply(scores, 2, var)
pve        <- var_scores / p
print(pve)
```

Output:

```
            PC1        PC2
Left5c  -0.5689839 0.00000000
Left1k  -0.5687330 0.00000000
Left2k   0.0000000 0.01364663
Left4k   0.0000000 0.70915622
Right5c  0.0000000 0.00000000
Right1k -0.5939699 0.00000000
Right2k  0.0000000 0.00000000
Right4k  0.0000000 0.70491931

PC1       PC2
0.3018235 0.2155123
```

Interpretation The two sparse principal components capture distinct frequency-specific patterns in the hearing-loss data, together explaining approximately 30.2% (PC1) and 21.6% (PC2) of the total variance, which is lower than original PCA. **PC1** has nonzero loadings only on Left5c, Left1k, and Right1k, indicating that these three variables are driven factors that represent an overall low-frequency hearing-loss factor. In contrast, **PC2** has nonzero loadings only on Left4k and Right4k, highlighting a high-frequency hearing-loss component. By focusing on just three variables per component, this sparse decomposition isolates the most important frequency bands—low vs. high—that drive variation in hearing loss among these 39-year-old males.

Question 4

(a)

We use the following R code to perform PCA **R Code:**

```
X <- read.table("tableICA", header = TRUE)  # or header = FALSE if no header

# Perform PCA (center and scale)
pca_res <- prcomp(X, center = TRUE, scale. = TRUE)

print(pca_res$rotation)

screeplot(pca_res, type = "lines",
          main = "Scree Plot of PCA on tableICA")

# Project observations onto the first two PCs
scores <- pca_res$x[, 1:2]
plot(scores,
     xlab = "PC1",
     ylab = "PC2",
     main = "Observations in First Two Principal Components",
     pch = 19,
     col = "blue")
abline(h = 0, v = 0, lty = 2, col = "gray")
```

Output:

```
          PC1        PC2        PC3
V1 -0.3177192 0.92163571  0.2228052
V2  0.6938867 0.06585576  0.7170664
V3  0.6462010 0.38242729 -0.6604344
```

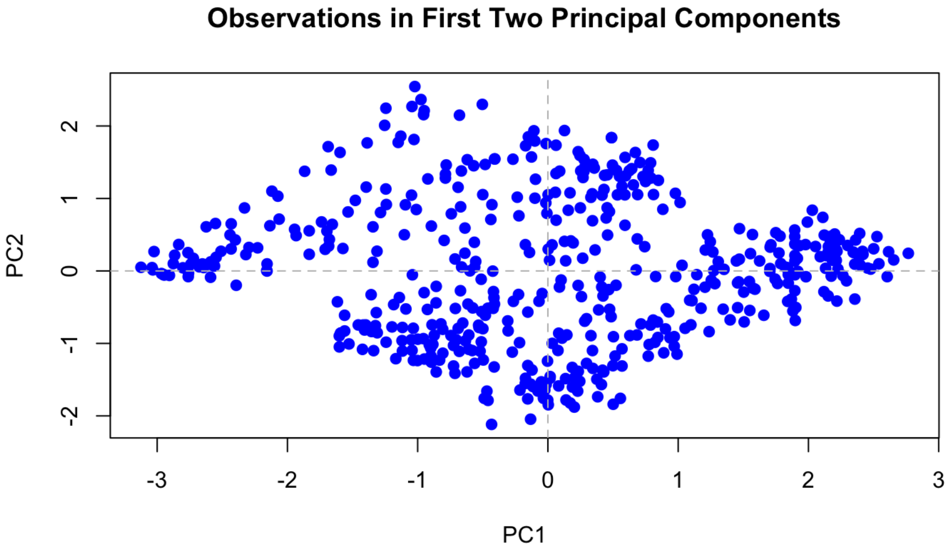

Figure 1: Observations in First Two Principal Components

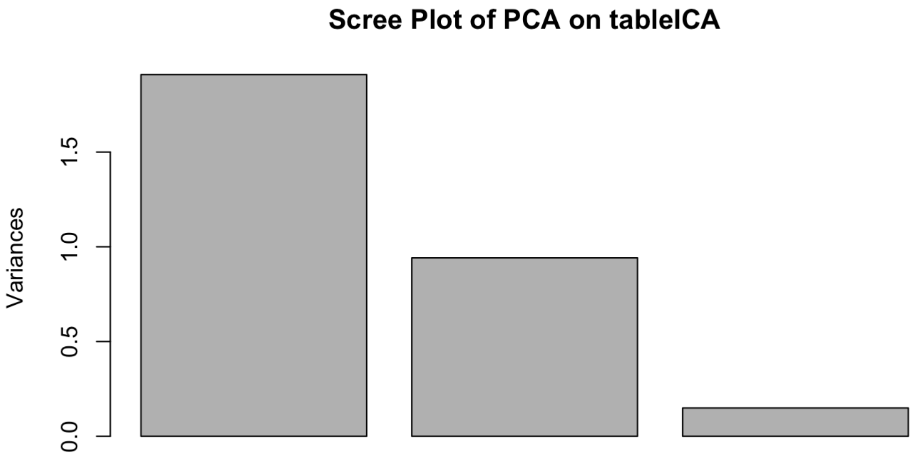

Figure 2: Scree Plot

Comments on PCA Results:

The scree plot shows a steep drop from PC1 to PC2 and then from PC2 to PC3, indicating that the first two components capture most of the variability. PC1 explains the largest part of the variance (highest eigenvalue) and PC2 the next most; subsequent components contribute markedly less.

PC1 has strong positive loadings on $V2$ (0.6939) and $V3$ (0.6462) and a moderate negative loading on $V1$ (–0.3177). Thus PC1 represents a contrast in which $V2$ and $V3$ move together against $V1$. **PC2** is dominated by $V1$ (0.9216) with much smaller contributions from $V2$ and $V3$. Therefore PC2 mainly captures variability unique to $V1$.

The scatterplot of the first two principal component scores reveals a continuous, lens-shaped distribution rather than distinct clusters. Points span the full range of PC1 (approximately -3 to 3) with no clear gaps or separated groups, indicating that there are no discrete subpopulations in these two dimensions. The vertical spread along PC2 is also relatively uniform, suggesting that PC2 captures a secondary gradient of variation orthogonal to PC1. A few points at the extremes of PC1 or PC2 hint at mild outliers, but overall the pattern supports a single, continuous structure in the data rather than categorical grouping.

It is worth noting that in the PC1–PC2 scatterplot, the marginal distributions of both PC1 and PC2 scores appear approximately Gaussian. This reflects PCA's inherent "Gaussianization" effect: by projecting onto uncorrelated directions (linear combinations of many variables), the resulting component scores often exhibit near-normal behavior, in line with the Central Limit Theorem.

(b)

We apply ICA to recover three latent sources from the observed mixtures.

R Code:

```
library(fastICA)
set.seed(123)
```

```
ica_res <- fastICA(X, n.comp = 3, fun = "logcosh")

S <- ica_res$S  # n × 3 matrix

# Plot the three ICs as a function of observation index
par(mfrow=c(3,1), mar=c(4,4,2,1))
plot(S[,1], type="l", main="ICA Component 1", xlab="Obs", ylab="IC1")
plot(S[,2], type="l", main="ICA Component 2", xlab="Obs", ylab="IC2")
plot(S[,3], type="l", main="ICA Component 3", xlab="Obs", ylab="IC3")
```

Output:

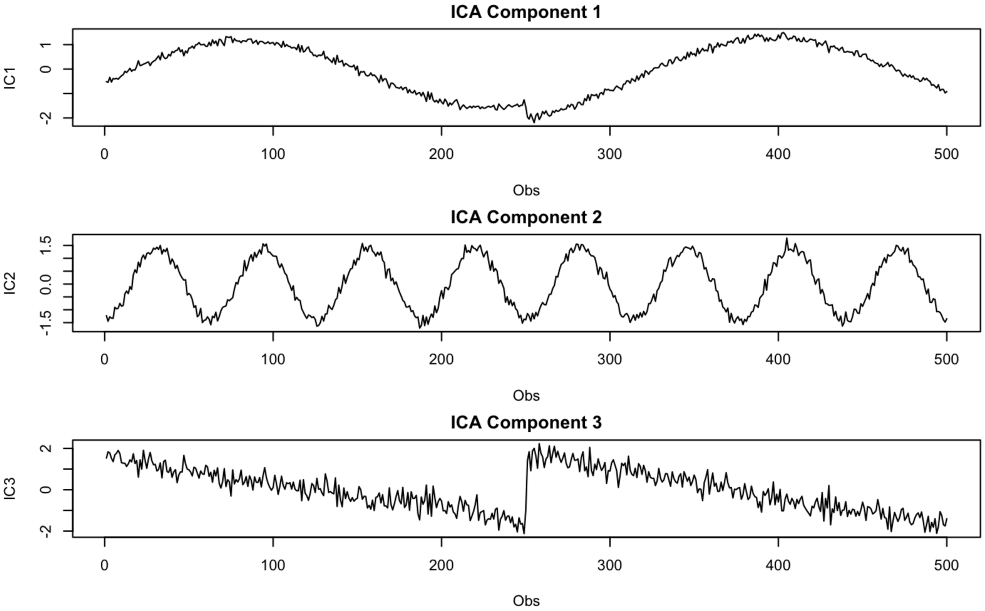

Figure 3: ICA Components

Interpretation:

- **IC1** shows a slowly varying, near–sinusoidal trend with amplitude modulation, suggesting one smooth source.
- **IC2** exhibits a clear high-frequency oscillation (multiple cycles), indicating a periodic source.
- **IC3** displays a piecewise-constant structure with a sharp jump around observation 250, indicating a block-step signal.
- Unlike PCA, which would yield approximately Gaussian components, ICA has separated three non-Gaussian, statistically independent signals—smooth trend, periodic oscillation, and abrupt change—that were linearly mixed in the observed data.

(c)

In order to compare the results using PCA and ICA, we first plot the signal recovered via PCA:

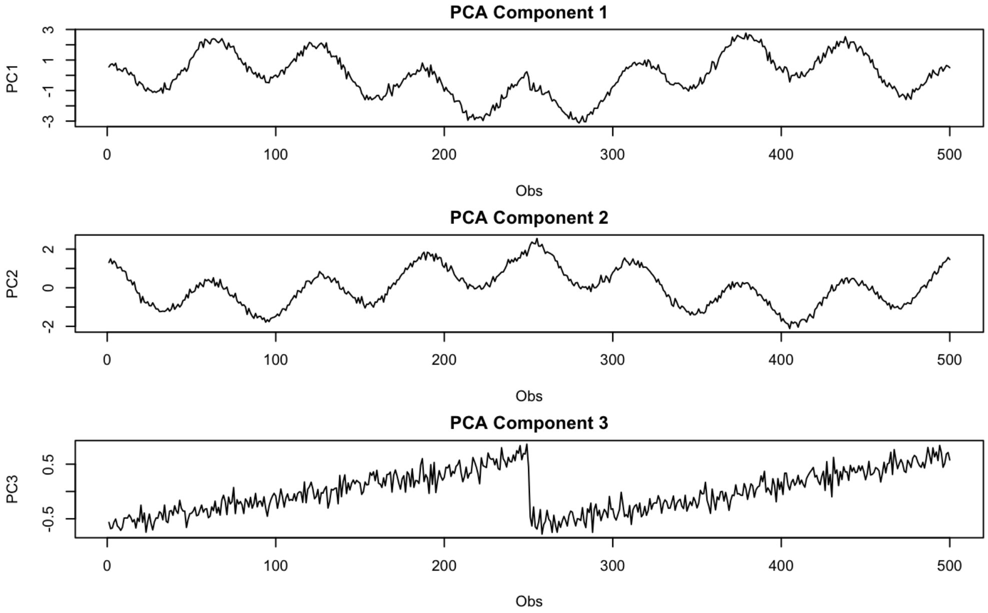

Figure 4: PCA Components

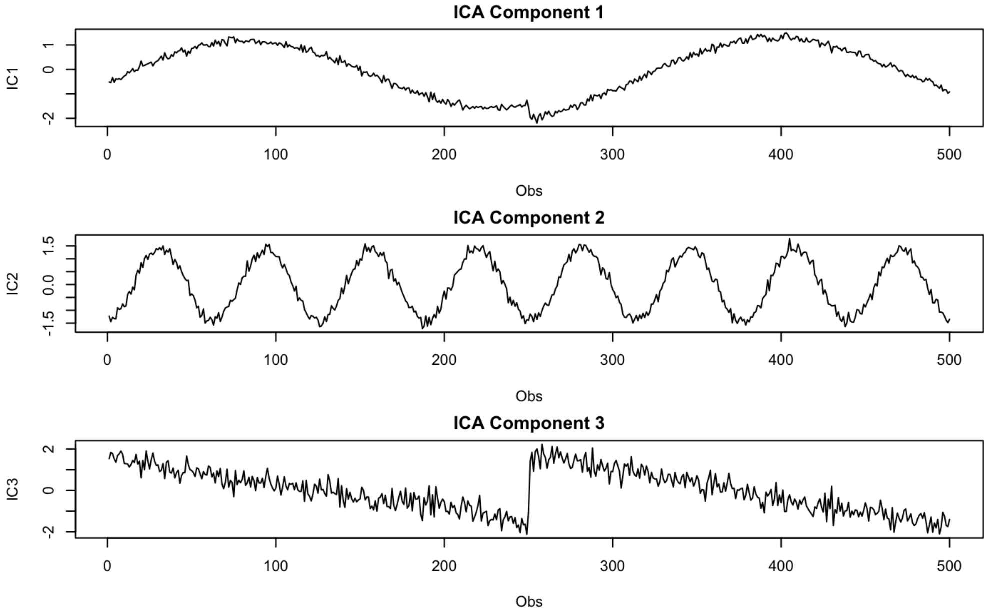

Figure 5: ICA Components

PCA Components (Part a):

- The first three principal components (PCs) each capture maximal variance under the orthogonality constraint.
- In the time-series plots, PC1 and PC2 both appear as smoothed, approximately sinusoidal waves (albeit with different phases and amplitudes), and PC3 shows a gradual trend with a fall near the midpoint.
- All PCs together explain decreasing proportions of total variance, but none strictly correspond to a single underlying signal. Instead, they remain linear mixtures of the true sources.

ICA Components (Part b):

- The three independent components (ICs) recovered by ICA are clearly non-Gaussian and statistically independent.
- IC1 exhibits a smooth, low-frequency oscillation; IC2 shows a higher-frequency periodic waveform; IC3 displays a step-change pattern at the same midpoint where PC3 falls.
- Each IC corresponds closely to one of the original latent signals.

Key Differences and Comments:

- **Orthogonality vs. Independence:** PCA enforces orthogonal (uncorrelated) directions and maximizes variance, but does not exploit higher-order statistics—hence its components remain Gaussian-like mixtures. ICA, by maximizing non-Gaussianity, achieves statistical independence and can recover the true source signals.
- **Interpretability:** PCA components are less interpretable as they combine multiple underlying patterns. In contrast, ICA components each reflect a distinct physical process (trend, periodicity, abrupt change), making them more meaningful.
- **Usage Recommendation:** When the goal is to identify and separate independent non-Gaussian sources, ICA is the appropriate choice.

Question 5

(a)

Let $X \sim \mathcal{N}(\mu, \sigma^2)$ have density

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\frac{(x-\mu)^2}{2\sigma^2}\Bigr).$$

The differential entropy is

$$H(X) = -\int_{-\infty}^{\infty} f(x)\, \log f(x)\, dx.$$

Substitute $f(x)$:

$$\begin{aligned} H(X) &= -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\tfrac{(x-\mu)^2}{2\sigma^2}\Bigr) \log\Bigl[\tfrac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\tfrac{(x-\mu)^2}{2\sigma^2}\Bigr)\Bigr]\, dx \\ &= -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\tfrac{(x-\mu)^2}{2\sigma^2}\Bigr) \Bigl[-\tfrac{1}{2}\log(2\pi\sigma^2) - \tfrac{(x-\mu)^2}{2\sigma^2}\Bigr]\, dx. \end{aligned}$$

Note that the integral of the density is 1, and

$$\operatorname{Var}(X) \;=\; E\bigl[(X-\mu)^2\bigr] \;=\; \int_{-\infty}^{\infty} (x-\mu)^2\, f(x)\, \mathrm{d}x$$

Thus,

$$\begin{aligned}
H(X) &= -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\frac{(x-\mu)^2}{2\sigma^2}\Bigr) \Bigl[-\tfrac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}\Bigr]\, dx \\
&= \frac{1}{2}\log(2\pi\sigma^2) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\frac{(x-\mu)^2}{2\sigma^2}\Bigr)\, dx \;+\; \frac{1}{2\sigma^2}\int_{-\infty}^{\infty} (x-\mu)^2 \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Bigl(-\frac{(x-\mu)^2}{2\sigma^2}\Bigr)\, dx \\
&= \frac{1}{2}\log(2\pi\sigma^2)\cdot 1 \;+\; \frac{1}{2\sigma^2}\cdot\sigma^2 \\
&= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2} \\
&= \log\bigl(\sqrt{2\pi\sigma^2}\bigr) + \log\bigl(\sqrt{e}\bigr) \\
&= \log\bigl(\sigma\sqrt{2\pi}\bigr) + \log\bigl(\sqrt{e}\bigr) \\
&= \log\bigl(\sigma\sqrt{2\pi e}\bigr).
\end{aligned}$$

Thus, univariate normal random variable has the entropy:

$$H(X) = \log\bigl(\sigma\sqrt{2\pi e}\bigr).$$

(b)

Let X be a continuous random variable with mean 0, variance σ^2 and density $f(x)$. Let

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\Bigl(-\frac{x^2}{2\sigma^2}\Bigr)$$

be the $\mathcal{N}(0,\sigma^2)$ density. Then

$$\begin{aligned}
-\int_{-\infty}^{\infty} f(x)\,\log\bigl(\phi(x)\bigr)\,\mathrm{d}x &= -\int_{-\infty}^{\infty} f(x)\,\log\Bigl[\tfrac{1}{\sigma\sqrt{2\pi}}\exp\bigl(-\tfrac{x^2}{2\sigma^2}\bigr)\Bigr]\,\mathrm{d}x \\
&= -\int_{-\infty}^{\infty} f(x)\,\Bigl[-\tfrac{x^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})\Bigr]\,\mathrm{d}x \\
&= \int_{-\infty}^{\infty} f(x)\,\frac{x^2}{2\sigma^2}\,\mathrm{d}x \;+\; \log(\sigma\sqrt{2\pi})\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x \\
&= \frac{1}{2\sigma^2}\,E[X^2] + \log(\sigma\sqrt{2\pi})\cdot 1 \\
&= \frac{1}{2\sigma^2}\bigl(\mathrm{Var}(X) + (E[X])^2\bigr) + \log(\sigma\sqrt{2\pi}) \\
&= \frac{1}{2\sigma^2}(\sigma^2 + 0) + \log(\sigma\sqrt{2\pi}) \quad (\text{since } E[X]=0,\ \mathrm{Var}(X)=\sigma^2) \\
&= \tfrac{1}{2} + \log(\sigma\sqrt{2\pi}) \\
&= \tfrac{1}{2} + \tfrac{1}{2}\log(2\pi\sigma^2) \\
&= \log\bigl(e^{1/2}(2\pi\sigma^2)^{1/2}\bigr) \\
&= \log\bigl(\sigma\sqrt{2\pi e}\bigr).
\end{aligned}$$

Thus

$$-\int_R f(x)\,\log\phi(x)\,\mathrm{d}x = \log\bigl(\sigma\sqrt{2\pi e}\bigr).$$

(c)

Let X be any continuous random variable on R with density $f(x)$, mean zero and variance σ^2.

Define its differential entropy

$$H(X) \; = \; -\int_{-\infty}^{\infty} f(x)\, \log f(x)\, \mathrm{d}x.$$

Let

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\Bigl(-\frac{x^2}{2\sigma^2}\Bigr)$$

be the density of $\mathcal{N}(0, \sigma^2)$, whose entropy we know is

$$H(\phi) = \log\bigl(\sigma\sqrt{2\pi e}\bigr).$$

From (b), we already have:

$$H(\phi) = -\int_R f(x)\, \log\phi(x)\, \mathrm{d}x = \log\bigl(\sigma\sqrt{2\pi e}\bigr).$$

Consider

$$\begin{aligned} H(\phi) - H(X) = \quad & -\int_{-\infty}^{\infty} f(x)\, \log\phi(x)\, \mathrm{d}x \; - \; \Bigl[-\int_{-\infty}^{\infty} f(x)\, \log f(x)\, \mathrm{d}x\Bigr] \\ = & \int_{-\infty}^{\infty} f(x)\, \log\frac{f(x)}{\phi(x)}\, \mathrm{d}x \end{aligned}$$

We need to prove that

$$\int_{-\infty}^{\infty} f(x)\, \log\frac{f(x)}{\phi(x)}\, \mathrm{d}x \; \geq \; 0,$$

So that $H(\phi) \geq H(X)$

Since log is a concave function, $-\log$ is convex. Moreover,

$$\int_{-\infty}^{\infty} \phi(x)\, \mathrm{d}x = \int_{-\infty}^{\infty} f(x)\, \mathrm{d}x = 1.$$

We apply Jensen's inequality to the convex function $g(u) = -\log u$. Define the random variable

$$Y \; = \; \frac{\phi(X)}{f(X)}$$

under the distribution f.

$$\begin{aligned} E_f[Y] &= \int_{-\infty}^{\infty} Y(x)\, f(x)\, \mathrm{d}x \quad && \text{(definition of expectation under density } f\text{)} \\ &= \int_{-\infty}^{\infty} \frac{\phi(x)}{f(x)}\, f(x)\, \mathrm{d}x \\ &= \int_{-\infty}^{\infty} \phi(x)\, \mathrm{d}x \\ &= 1 && \text{(since } \phi(x) \text{ is a probability density).} \end{aligned}$$

Then

$$E_f[Y] = \int f(x)\, \frac{\phi(x)}{f(x)}\, \mathrm{d}x = \int \phi(x)\, \mathrm{d}x = 1.$$

By Jensen's inequality,

$$g\bigl(E_f[Y]\bigr) \; \leq \; E_f\bigl[g(Y)\bigr],$$

i.e.

$$-\log\bigl(E_f[Y]\bigr) \;\le\; E_f\bigl[-\log Y\bigr].$$

Since $E_f[Y] = 1$,

$$0 \;\le\; \int_{-\infty}^{\infty} f(x)\,\bigl[-\log(\tfrac{\phi(x)}{f(x)})\bigr]\,\mathrm{d}x = \;-\int f(x)\,\log\phi(x)\,\mathrm{d}x \;+\; \int f(x)\,\log f(x)\,\mathrm{d}x.$$

$$-\int f(x)\,\log\phi(x)\,\mathrm{d}x \;\ge\; -\int f(x)\,\log f(x)\,\mathrm{d}x$$

$$H(\phi) \;\ge\; H(f).$$

To prove equality holds if and only if X is of normal distribution, we need to prove that:

$$H(\phi) - H(X) = \int f(x)\,\log\frac{f(x)}{\phi(x)}\,\mathrm{d}x = 0.$$

Thus we must have

$$\frac{f(x)}{\phi(x)} = C \quad \text{for almost every } x.$$

Hence

$$f(x) = C\,\phi(x).$$

Integrating both sides over R gives

$$1 = \int f(x)\,\mathrm{d}x = C\int \phi(x)\,\mathrm{d}x = C\;\cdot 1,$$

so $C = 1$.

Therefore

$$f(x) = \phi(x) \quad \text{a.e.},$$

i.e. X has the $\mathcal{N}(0,\sigma^2)$ density.

Conversely, if X is normal then $f = \phi$ and $h(f) = h(\phi)$.

This completes the proof that

$$h(f) \le h(\phi),$$

with equality if and only if $X \sim \mathcal{N}(0,\sigma^2)$.

(d)

Let X_1, X_2 be continuous on R with $E[X_i] = 0$ and $\mathrm{Var}(X_i) = \sigma_i^2$. Define

$$Y = X_1 + X_2.$$

Then

$$\mathrm{Var}(Y) = \mathrm{Var}(X_1) + \mathrm{Var}(X_2) + 2\,\mathrm{Cov}(X_1,X_2) \le \sigma_1^2 + \sigma_2^2 + 2\,\sigma_1\sigma_2 = (\sigma_1+\sigma_2)^2,$$

Where we used the fact that, for two random variables U and V:

$$U = X_1 - E[X_1], \quad V = X_2 - E[X_2],$$

$$\begin{aligned} \bigl|\mathrm{Cov}(X_1,X_2)\bigr| &= \bigl|E[UV]\bigr| \\ &\le \sqrt{E[U^2]\,E[V^2]} \quad \text{(by Cauchy–Schwarz)} \\ &= \sqrt{\mathrm{Var}(X_1)\,\mathrm{Var}(X_2)} \\ &= \sigma_1\,\sigma_2. \end{aligned}$$

By the maximum-entropy property of the normal distribution, for any continuous Z with variance V,

$$H(Z) \;\le\; \tfrac{1}{2}\log\bigl(2\pi e\, V\bigr).$$

Applied to Y, we get

$$H(Y) \;\le\; \tfrac{1}{2}\log\bigl(2\pi e\,\mathrm{Var}(Y)\bigr) \;\le\; \tfrac{1}{2}\log\bigl(2\pi e\,(\sigma_1+\sigma_2)^2\bigr) = \log\bigl((\sigma_1+\sigma_2)\sqrt{2\pi e}\bigr).$$

Hence

$$\max_{X_1,X_2} H(Y) = \log\bigl((\sigma_1+\sigma_2)\sqrt{2\pi e}\bigr).$$

Equality requires both

1. $\mathrm{Var}(Y)$ is maximal, i.e. $\mathrm{Cov}(X_1,X_2)=\sigma_1\sigma_2$, and

2. Y is Gaussian.

First, we have

$$\mathrm{Corr}(X_1,X_2) = \frac{\mathrm{Cov}(X_1,X_2)}{\sigma_1\sigma_2} = 1 \quad\Longrightarrow\quad X_2 = \frac{\sigma_2}{\sigma_1}\,X_1$$

Therefore

$$Y = X_1+X_2 = \Bigl(1+\frac{\sigma_2}{\sigma_1}\Bigr)\,X_1 \;\Longrightarrow\; X_1 = \frac{\sigma_1}{\sigma_1+\sigma_2}\,Y.$$

Since Y is Gaussian and X_1 is a non-singular linear function of Y, it follows that

$$X_1 \;\sim\; \mathcal{N}\Bigl(0,\,\bigl(\tfrac{\sigma_1}{\sigma_1+\sigma_2}\bigr)^2(\sigma_1+\sigma_2)^2\Bigr) = \mathcal{N}(0,\sigma_1^2).$$

Finally,

$$X_2 = \frac{\sigma_2}{\sigma_1}\,X_1$$

is again a linear transform of a Gaussian, hence

$$X_2 \sim \mathcal{N}(0,\sigma_2^2).$$

Thus X_1 and X_2 must both be Gaussian.

Thus we start by choosing

$$X_1 \sim \mathcal{N}(0,\sigma_1^2), \quad X_2 = \frac{\sigma_2}{\sigma_1}\,X_1 \sim \mathcal{N}(0,\sigma_2^2).$$

Since X_1 is normal, X_2 is also normal with

$$E[X_2] = \frac{\sigma_2}{\sigma_1}\,E[X_1] = 0, \qquad \mathrm{Var}(X_2) = \Bigl(\frac{\sigma_2}{\sigma_1}\Bigr)^2\mathrm{Var}(X_1) = \sigma_2^2.$$

Their covariance is

$$\mathrm{Cov}(X_1,X_2) = E\bigl[X_1\cdot\tfrac{\sigma_2}{\sigma_1}X_1\bigr] = \frac{\sigma_2}{\sigma_1}\,E[X_1^2] = \sigma_1\,\sigma_2,$$

so $\mathrm{Corr}(X_1,X_2)=1$.

Hence

$$Y = X_1+X_2 = \Bigl(1+\tfrac{\sigma_2}{\sigma_1}\Bigr)\,X_1 \;\sim\; \mathcal{N}\bigl(0,(\sigma_1+\sigma_2)^2\bigr),$$

which indeed has $\mathrm{Var}(Y)=(\sigma_1+\sigma_2)^2$, the maximum possible, and thus

$$H(Y) = \tfrac{1}{2}\log\bigl(2\pi e(\sigma_1+\sigma_2)^2\bigr) = \log\bigl((\sigma_1+\sigma_2)\sqrt{2\pi e}\bigr).$$

These hold exactly when (X_1,X_2) is jointly normal with perfect positive correlation. Equivalently one may take

$$X_1 \sim \mathcal{N}(0,\sigma_1^2), \quad X_2 = \frac{\sigma_2}{\sigma_1}\,X_1,$$

so that $X_2 \sim \mathcal{N}(0, \sigma_2^2)$, $\mathrm{Cov}(X_1, X_2) = \sigma_1\sigma_2$, and

$$Y = X_1 + X_2 \sim \mathcal{N}\big(0, (\sigma_1 + \sigma_2)^2\big),$$

which achieves $H(Y) = \log\big((\sigma_1 + \sigma_2)\sqrt{2\pi e}\big)$.

Therefore, we should choose:

$$X_1 \sim \mathcal{N}(0, \sigma_1^2), \quad X_2 = \frac{\sigma_2}{\sigma_1} X_1 \sim \mathcal{N}(0, \sigma_2^2).$$

Question 6

(a)

$$E[X] = \begin{pmatrix} E[X_1] \\ \vdots \\ E[X_p] \end{pmatrix} = \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_p \end{pmatrix} \in R^{p\times 1}.$$

(b)

Since $X = (X_1, \ldots, X_p)^\top$ has independent components, for $i \neq j$, independence gives

$$\mathrm{Cov}[X_i X_j] = 0.$$

Thus,

$$\mathrm{Cov}(X) = \begin{pmatrix} \mathrm{Var}(X_1) & 0 & \cdots & 0 \\ 0 & \mathrm{Var}(X_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{Var}(X_p) \end{pmatrix} = \begin{pmatrix} \nu_1(1-\nu_1) & 0 & \cdots & 0 \\ 0 & \nu_2(1-\nu_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \nu_p(1-\nu_p) \end{pmatrix} \in R^{p\times p}.$$

(c)

(i)

By the law of total expectation,

$$\mu \;=\; E[Y] = E\big[E[Y \mid C]\big] = \sum_{c=1}^{K} \Pr(C = c)\, E[Y \mid C = c].$$

Since conditional on $C = c$, Y has mean $\mu_c = (\mu_{c1}, \ldots, \mu_{cp})^\top$, we get

$$\mu = \sum_{c=1}^{K} \pi_c\, \mu_c = \begin{pmatrix} \sum_{c=1}^{K} \pi_c\, \mu_{c1} \\ \vdots \\ \sum_{c=1}^{K} \pi_c\, \mu_{cp} \end{pmatrix} \in R^{p\times 1}.$$

(ii)

Let $y = (y_1, \dots, y_p)^\top \in \{0,1\}^p$ be an observed vector. Conditional on $C = c$, each component Y_i is an independent Bernoulli(μ_{ci}). Hence for each i:

$$P(Y_i = y_i \mid C = c) = \begin{cases} \mu_{ci}, & y_i = 1, \\ 1 - \mu_{ci}, & y_i = 0. \end{cases}$$

By independence,

$$\begin{aligned} P(Y = y \mid C = c) &= \prod_{i=1}^{p} P(Y_i = y_i \mid C = c) \\ &= \prod_{i=1}^{p} \begin{cases} \mu_{ci}, & y_i = 1, \\ 1 - \mu_{ci}, & y_i = 0, \end{cases} \\ &= \prod_{i=1}^{p} \mu_{ci}^{y_i} \, (1 - \mu_{ci})^{1 - y_i}. \end{aligned}$$

(iii)

By the law of total probability,

$$P(Y = y \mid \pi, \{\mu_c\}) = \sum_{c=1}^{K} P(C = c) \, P(Y = y \mid C = c) = \sum_{c=1}^{K} \pi_c \, P(Y = y \mid C = c).$$

Since $P(Y = y \mid C = c) = \prod_{i=1}^{p} \mu_{ci}^{y_i} (1 - \mu_{ci})^{1 - y_i}$, we get

$$P(Y = y \mid \pi, \mu_1, \dots, \mu_K) = \sum_{c=1}^{K} \pi_c \prod_{i=1}^{p} \mu_{ci}^{y_i} \, (1 - \mu_{ci})^{1 - y_i}.$$

(iv)

Use the law of total covariance for a random vector Y and discrete variable C:

$$\mathrm{Cov}(Y) = E\big[\mathrm{Cov}(Y \mid C)\big] \; + \; \mathrm{Cov}\big(E[Y \mid C]\big).$$

Here $C \in \{1, \dots, K\}$ with $\Pr(C = c) = \pi_c$, and conditional on $C = c$,

$$E[Y \mid C = c] = \mu_c, \qquad \mathrm{Cov}(Y \mid C = c) = \Sigma_c.$$

Moreover, from part (c-i) the marginal mean is $\mu = E[Y] = \displaystyle\sum_{c=1}^{K} \pi_c \, \mu_c$.

Thus

$$\begin{aligned} E\big[\mathrm{Cov}(Y \mid C)\big] &= \sum_{c=1}^{K} \pi_c \, \Sigma_c \;=\; \sum_{c=1}^{K} \pi_c \, \mathrm{diag}\big(\mu_{c1}(1 - \mu_{c1}), \dots, \mu_{cp}(1 - \mu_{cp})\big) \\ &= \mathrm{diag}\Big(\sum_{c=1}^{K} \pi_c \, \mu_{c1}(1 - \mu_{c1}), \; \dots, \; \sum_{c=1}^{K} \pi_c \, \mu_{cp}(1 - \mu_{cp})\Big). \end{aligned}$$

For $\mathrm{Cov}\big(E[Y \mid C]\big)$, we first have:

$$E[\mu_C] = \sum_{c=1}^{K} \Pr(C = c) \, \mu_c = \sum_{c=1}^{K} \pi_c \, \mu_c = \mu.$$

Then,

$$\begin{aligned}\operatorname{Cov}\bigl(E[Y \mid C]\bigr) &= \operatorname{Cov}(\mu_C)\\ &= E\bigl[(\mu_C - E[\mu_C])(\mu_C - E[\mu_C])^\top\bigr]\\ &= E\bigl[\mu_C\mu_C^\top - \mu_C\, E[\mu_C]^\top - E[\mu_C]\,\mu_C^\top + E[\mu_C]\,E[\mu_C]^\top\bigr]\\ &= E[\mu_C\mu_C^\top] - E[\mu_C]\,E[\mu_C]^\top\\ &= \sum_{c=1}^{K}\pi_c\,\mu_c\mu_c^\top \;-\; \mu\,\mu^\top,\end{aligned}$$

where $\mu = E[\mu_C] = \sum_{c=1}^{K}\pi_c\,\mu_c$.

Thus,

$$\begin{aligned}\operatorname{Cov}(Y) &= \sum_{c=1}^{K}\pi_c\operatorname{diag}\bigl(\mu_{c1}(1-\mu_{c1}),\dots,\mu_{cp}(1-\mu_{cp})\bigr) \;+\; \sum_{c=1}^{K}\pi_c\,\mu_c\mu_c^\top \;-\; \mu\,\mu^\top\\ &= \operatorname{diag}\Bigl(\sum_{c=1}^{K}\pi_c\,\mu_{c1}(1-\mu_{c1}),\;\dots,\;\sum_{c=1}^{K}\pi_c\,\mu_{cp}(1-\mu_{cp})\Bigr) \;+\; \sum_{c=1}^{K}\pi_c\,\mu_c\mu_c^\top \;-\; \mu\,\mu^\top\\ &= \sum_{c=1}^{K}\pi_c\Bigl[\operatorname{diag}\bigl(\mu_{c1}(1-\mu_{c1}),\dots,\mu_{cp}(1-\mu_{cp})\bigr) +\; \mu_c\mu_c^\top\Bigr] \;-\; \mu\,\mu^\top.\end{aligned}$$

Alternatively,

$$\begin{aligned}\sum_{c=1}^{K}\pi_c\,\mu_c\mu_c^\top \;-\; \mu\mu^\top &= \sum_{c=1}^{K}\pi_c\,\mu_c\mu_c^\top - \;\Bigl(\sum_{c=1}^{K}\pi_c\Bigr)\mu\mu^\top\\ &= \sum_{c=1}^{K}\pi_c\,\mu_c\mu_c^\top \;-\; \sum_{c=1}^{K}\pi_c\,\mu\mu^\top\\ &= \sum_{c=1}^{K}\pi_c\bigl(\mu_c\mu_c^\top - \mu\mu^\top\bigr)\\ &= \sum_{c=1}^{K}\pi_c\Bigl[\mu_c\mu_c^\top - \mu_c\mu^\top + \mu_c\mu^\top - \mu\mu^\top\Bigr] \quad (\text{add and subtract } \mu_c\mu^\top)\end{aligned}$$

Note that:

$$\sum_{c=1}^{K}\pi_c\,(\mu_c-\mu)(\mu_c-\mu)^\top = \sum_{c=1}^{K}\pi_c\bigl(\mu_c\mu_c^\top - \mu_c\mu^\top - \mu\,\mu_c^\top + \mu\mu^\top\bigr)$$

Thus,

$$\sum_{c=1}^{K}\pi_c\Bigl[\mu_c\mu_c^\top - \mu_c\mu^\top + \mu_c\mu^\top - \mu\mu^\top\Bigr] = \sum_{c=1}^{K}\pi_c\Bigl[(\mu_c-\mu)(\mu_c-\mu)^\top\Bigr]$$

Thus, $Cov(Y)$ could also be written as:

$$\operatorname{Cov}(Y) = \sum_{c=1}^{K}\pi_c\,\Sigma_c + \sum_{c=1}^{K}\pi_c\,(\mu_c-\mu)(\mu_c-\mu)^\top = \sum_{c=1}^{K}\pi_c\Bigl[\operatorname{diag}\bigl(\mu_{c1}(1-\mu_{c1}),\dots,\mu_{cp}(1-\mu_{cp})\bigr)\Bigr]+\sum_{c=1}^{K}\pi_c\,(\mu_c-\mu)(\mu_c-\mu)^\top$$

Therefore,

$$\operatorname{Cov}(Y) = \sum_{c=1}^{K}\pi_c\Bigl[\operatorname{diag}\bigl(\mu_{c1}(1-\mu_{c1}),\dots,\mu_{cp}(1-\mu_{cp})\bigr)\Bigr] + \sum_{c=1}^{K}\pi_c\,(\mu_c-\mu)(\mu_c-\mu)^\top$$

or

$$\mathrm{Cov}(Y) = \sum_{c=1}^{K} \pi_c \Big[\mathrm{diag}\big(\mu_{c1}(1-\mu_{c1}), \ldots, \mu_{cp}(1-\mu_{cp})\big) + \ \mu_c \mu_c^\top\Big] \ - \ \mu\, \mu^\top.$$

(d)

(i)

Let $y^{(i)} = (y_{i1}, \ldots, y_{ip})^\top$, $i = 1, \ldots, n$, be independent observations. The likelihood of the mixture model is

$$\begin{aligned} L(\{\mu_c\}, \pi \mid Y) &= \prod_{i=1}^{n} P\big(Y = y^{(i)} \mid \{\mu_c\}, \pi\big) \\ &= \prod_{i=1}^{n} \sum_{c=1}^{K} \pi_c \, P\big(Y = y^{(i)} \mid C = c\big) \\ &= \prod_{i=1}^{n} \sum_{c=1}^{K} \pi_c \prod_{j=1}^{p} \mu_{cj}^{\,y_{ij}} \, (1-\mu_{cj})^{\,1-y_{ij}}. \end{aligned}$$

(ii)

Take $p = 3$, $n = 4$, and invent the following non-degenerate binary observations:

$$Y \ = \ \big\{y^{(i)}\big\}_{i=1}^{4} \ = \ \begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \\ y_{41} & y_{42} & y_{43} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

(iii)

The mixture-Bernoulli likelihood from (d-i) specializes to

$$L(\{\mu_c\}, \pi \mid Y) = \prod_{i=1}^{4} \sum_{c=1}^{K} \pi_c \prod_{j=1}^{3} \mu_{cj}^{\,y_{ij}} \, (1-\mu_{cj})^{\,1-y_{ij}}.$$

Substituting each $y^{(i)}$ gives

$$\begin{aligned} L = &\Big[\sum_{c=1}^{K} \pi_c \, \mu_{c1}(1-\mu_{c2})\mu_{c3}\Big] \ \times \ \Big[\sum_{c=1}^{K} \pi_c \, (1-\mu_{c1})\mu_{c2}(1-\mu_{c3})\Big] \\ &\times \Big[\sum_{c=1}^{K} \pi_c \, \mu_{c1}\mu_{c2}\mu_{c3}\Big] \ \times \ \Big[\sum_{c=1}^{K} \pi_c \, (1-\mu_{c1})(1-\mu_{c2})\mu_{c3}\Big]. \end{aligned}$$