STAT3622 Assignment2

Havin Chung (3035729772)

Question 1

First, to remove the norm constraint $\|\boldsymbol{w}\| = 1$, we rescale the parameters. Define $\boldsymbol{w}' = \frac{\boldsymbol{w}}{M}$ and $b' = \frac{b}{M}$, where $M > 0$ since we are maximizing a positive margin. Since $\|\boldsymbol{w}\| = 1$, the norm of the rescaled weight vector is:

$$\|\boldsymbol{w}'\| = \left\|\frac{\boldsymbol{w}}{M}\right\| = \frac{\|\boldsymbol{w}\|}{M} = \frac{1}{M}$$

Substitute $\boldsymbol{w} = M\boldsymbol{w}'$ and $b = Mb'$ into the constraint:

$$y_i\left((M\boldsymbol{w}')^\top \boldsymbol{x}_i + Mb'\right) \geq M(1-\xi_i)$$

Divide both sides by M:

$$y_i\left(\boldsymbol{w}'^\top \boldsymbol{x}_i + b'\right) \geq 1 - \xi_i$$

For the slack variables, we define $\xi_i' = \xi_i$. $\sum_i \xi_i \leq C$ remains unchanged since $\xi_i' = \xi_i$. However, the objective changes to maximizing M is equivalent to minimizing $\frac{1}{M}$, which is $\|\boldsymbol{w}'\|$. Thus, the problem becomes like:

$$\min_{\boldsymbol{w}',b'} \|\boldsymbol{w}'\| \quad \text{subject to} \quad y_i(\boldsymbol{w}'^\top \boldsymbol{x}_i + b') \geq 1 - \xi_i', \quad \xi_i' \geq 0, \quad \sum_i \xi_i' \leq C$$

For the **hard-margin SVM**,where $\xi_i = 0$, can be simplified to:

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\| \quad \text{subject to} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$$

As $\boldsymbol{w}'$ and b' are just scaled versions of $\boldsymbol{w}$ and b.

For the **soft-margin SVM**, we can start with the standard form:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \xi_i \quad \text{subject to} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

The term $\frac{1}{2}\|\boldsymbol{w}\|^2$ comes from the fact that minimizing $\|\boldsymbol{w}\|$ is equivalent to minimizing $\|\boldsymbol{w}\|^2$, and the factor $\frac{1}{2}$ is a convention for optimization convenience.

The slack variable ξ_i measures the margin violation. If $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$, then $\xi_i = 0$. If $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) < 1$, then $\xi_i = 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)$. Which is exactly the hinge loss:

$$(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))_+ = \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))$$

Thus, $\xi_i = (1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))_+$ and substituting this into the objective:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i (1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))_+$$

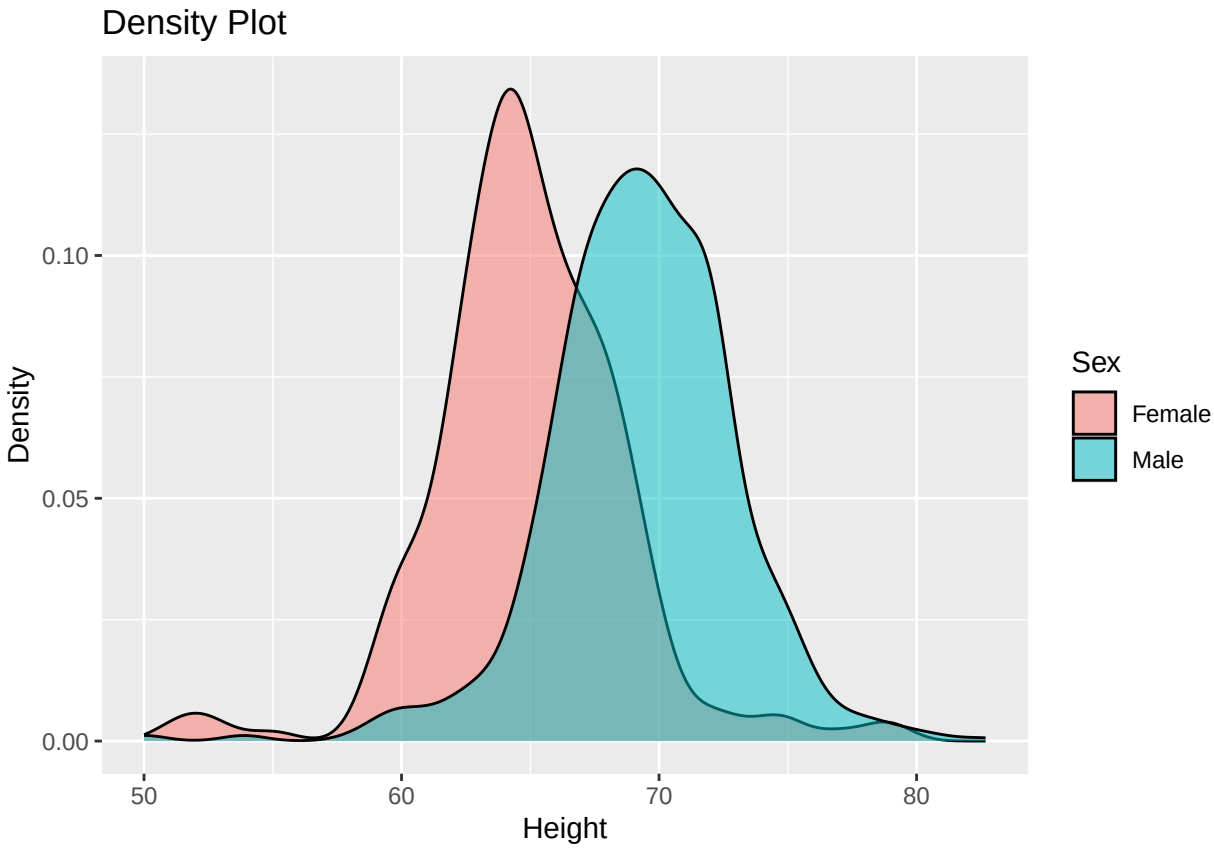
This matches the given final penalized form:

$$\min_{\boldsymbol{w},b} \sum_{i=1}^{n} (1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))_+ + \lambda\|\boldsymbol{w}\|^2$$

Question 2

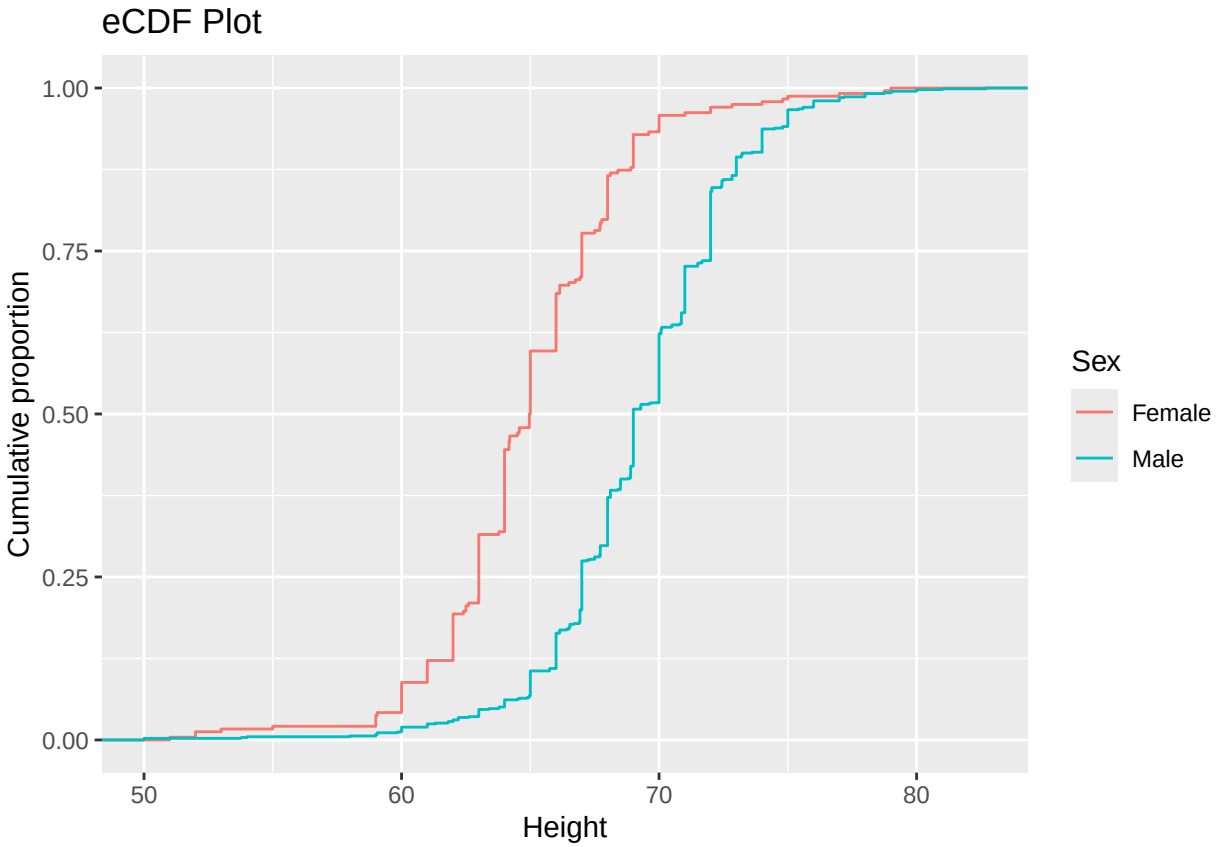
Density plot

```
data(heights)
ggplot(heights, aes(x = height, fill = sex)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot", x = "Height", y = "Density", fill = "Sex")
```



eCDF plot

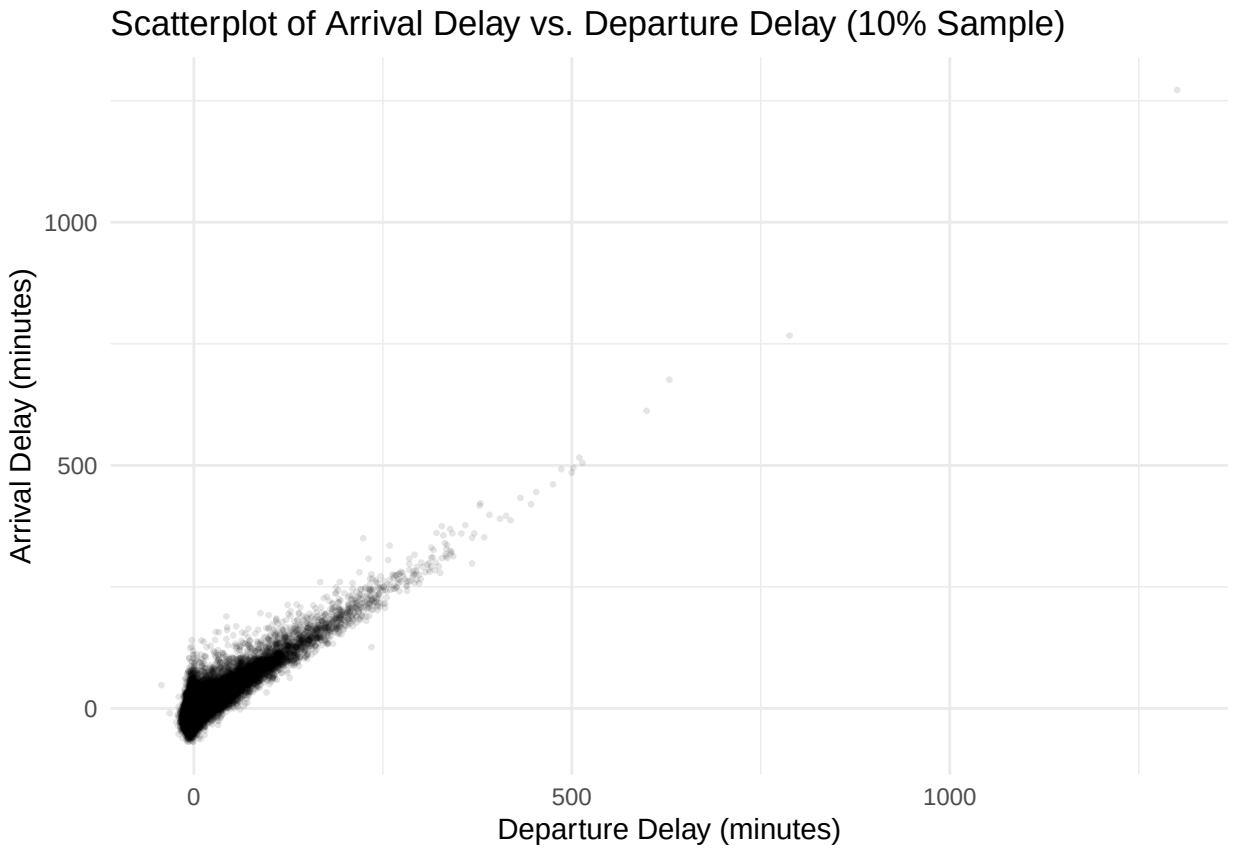
```
ggplot(heights, aes(x = height, color = sex)) +
  stat_ecdf() +
  labs(title = "eCDF Plot", x = "Height", y = "Cumulative proportion", color = "Sex")
```



- **Density Plot**: Emphasizes the shape, peak, and spread (variability) of the distributions, which helps compare the overall height profiles of males and females.
- **eCDF Plot**: Focuses on cumulative proportions and quantiles (e.g., median, where the curve hits 0.5), making it easier to compare the fraction of each group below a given height.

Question 3

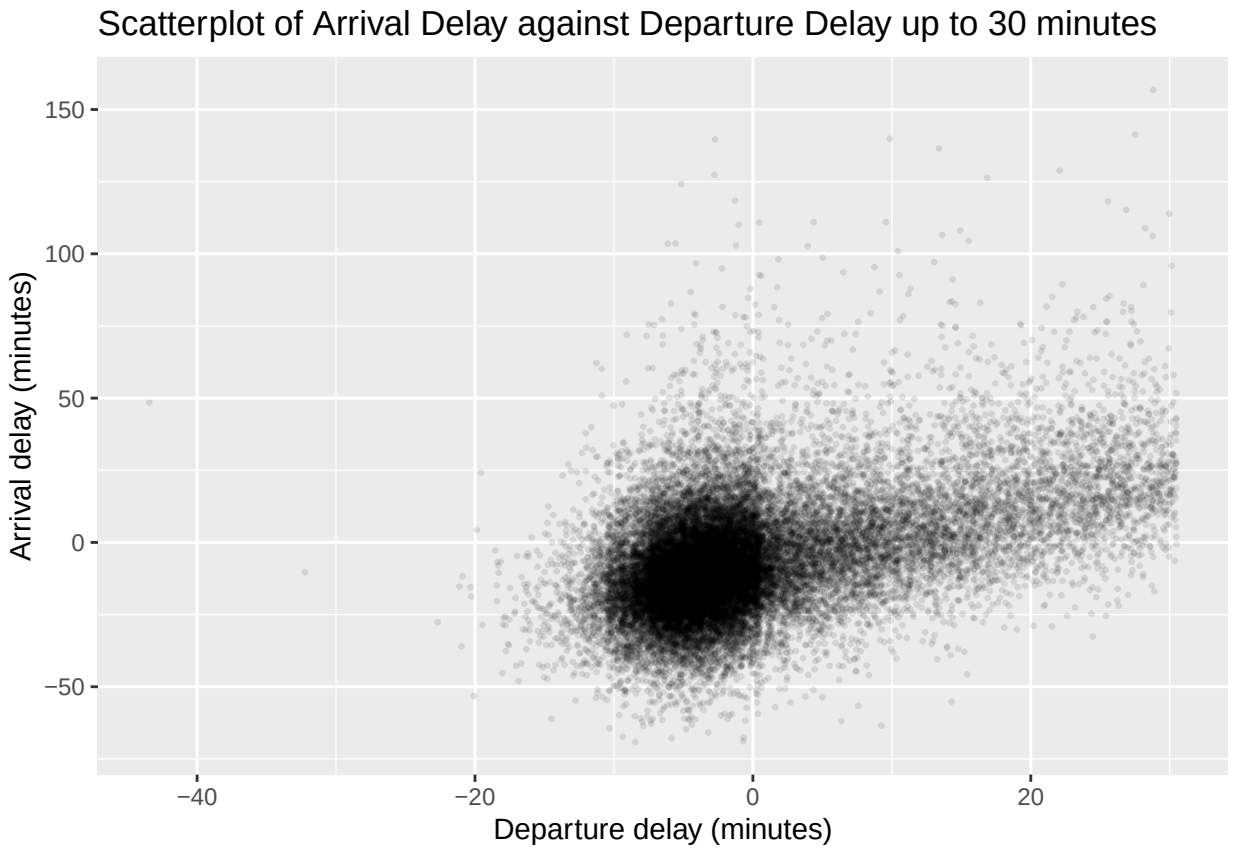
```r
flights_sample <- flights %>% na.omit() %>% sample_frac(0.1)

ggplot(flights_sample, aes(x = dep_delay, y = arr_delay)) +
  geom_point(size = 0.5, alpha = 0.1) +
  labs(title = "Scatterplot of Arrival Delay vs. Departure Delay (10% Sample)",
       x = "Departure Delay (minutes)",
       y = "Arrival Delay (minutes)") +
  theme_minimal()
```



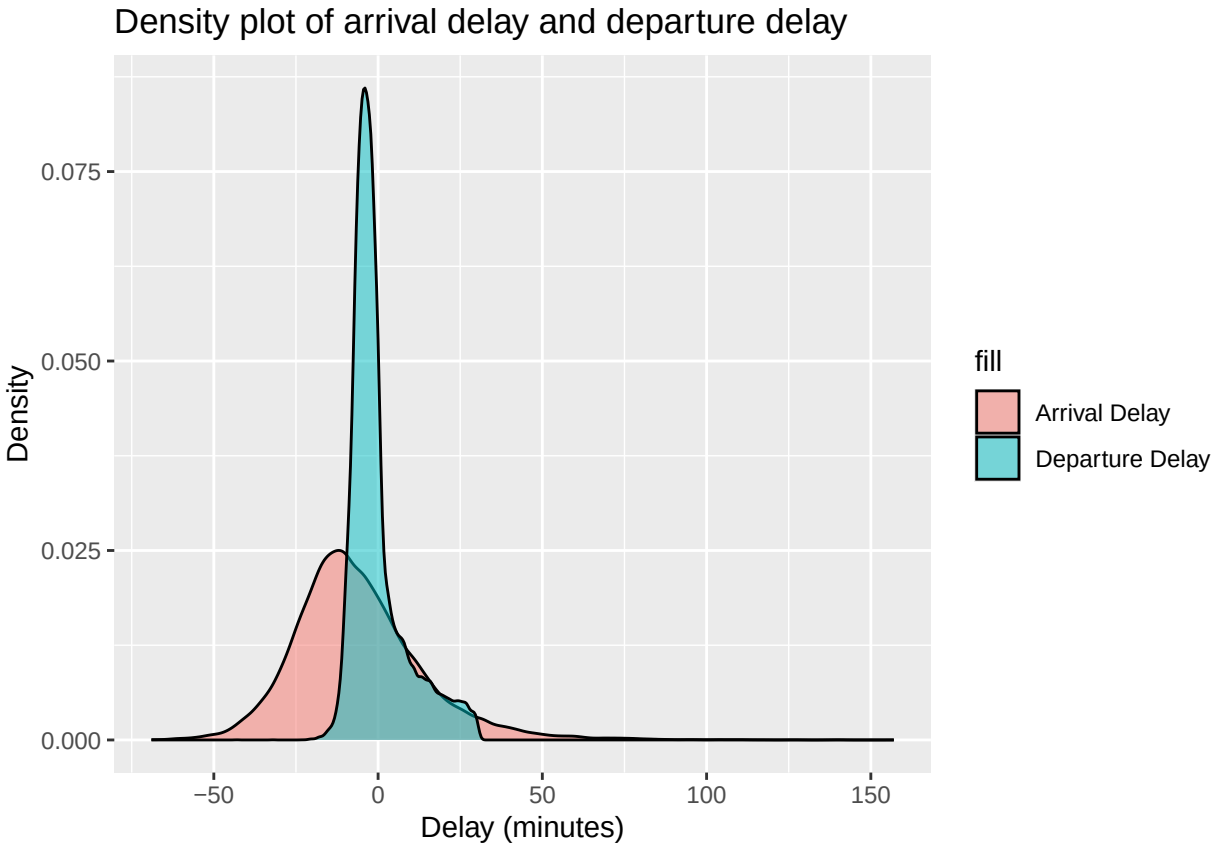
Comment:

The scatterplot shows a strong positive relationship between departure and arrival delays: flights with longer departure delays tend to have longer arrival delays. Most points cluster near the origin (small or no delays), but there's a clear linear trend along the line, indicating that departure delays often translate directly into arrival delays. Some flights have negative delays (early departures/arrivals), and there are outliers with very large delays (e.g., > 500 minutes).

```
new_flights_sample <- flights_sample %>% dplyr::filter(dep_delay <= 30)

p <- ggplot(new_flights_sample, aes(x = dep_delay, y = arr_delay)) +
  geom_point(size = 0.5, alpha = 0.1, position = position_jitter(width = 0.5, height = 0.5)) +
  labs(title = "Scatterplot of Arrival Delay against Departure Delay up to 30 minutes",
       x = "Departure delay (minutes)",
       y = "Arrival delay (minutes)")
p
```




```
ggplot(new_flights_sample, aes(x = dep_delay, fill = "Departure Delay")) +
  geom_density(aes(x = arr_delay, fill = "Arrival Delay"), alpha = 0.5) + geom_density(alpha = 0.5) +
  labs(title = "Density plot of arrival delay and departure delay", x = "Delay (minutes)", y = "Density
```



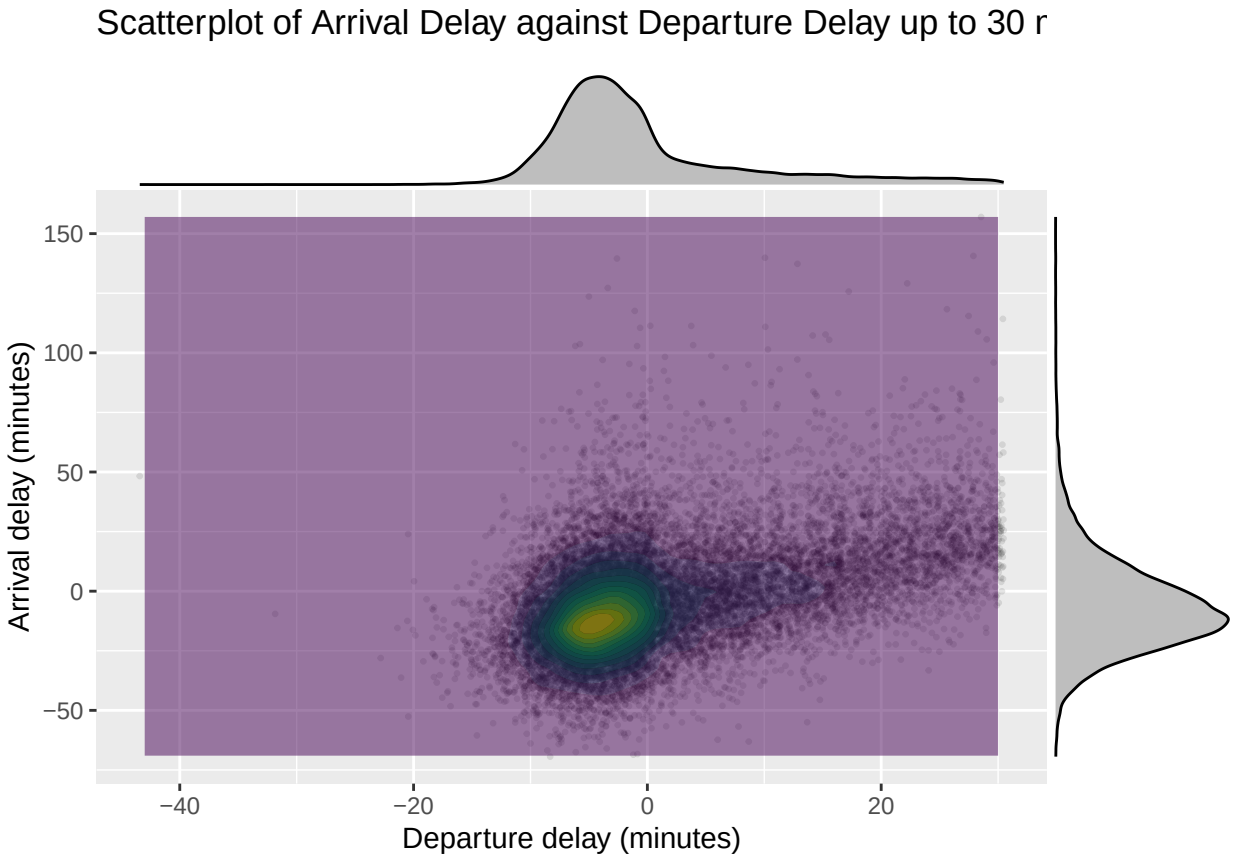
Comment:

The scatterplot shows a clear linear correlation between departure and arrival delays, with most data points clustering on the left, indicating that flights with departure delays of up to 30 minutes often have minimal delays or arrive early (negative delay values); to support this, I also created a density plot to visually represent the distribution of both delays.

```
p <- p + geom_density_2d_filled(aes(x = dep_delay, y = arr_delay),
                                contour = TRUE,
                                alpha = 0.5,
                                show.legend = FALSE)
```

```
## Warning: Duplicated aesthetics after name standardisation: contour
```

```
ggMarginal(p, type = "density", margins = "both", size = 5, fill = "gray")
```



Comment:

We can also see that most flights depart and arrive earlier than scheduled, with departure delays showing slightly more variation than arrival delays.

Question 4

Loading the data:

```
data <- read.table("big8.txt", header = TRUE, sep=' ')
```

(a)

(i) Ordinary least squares:

```
ols_model <- lm(sprtrn~ RETX, data = data)
summary(ols_model)
```

```
##
## Call:
## lm(formula = sprtrn ~ RETX, data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0284607 -0.0035358  0.0000985  0.0035024  0.0247446
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0002860  0.0001264   2.264   0.0237 *
## RETX        0.3290424  0.0102336  32.153   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005672 on 2014 degrees of freedom
## Multiple R-squared:  0.3392, Adjusted R-squared:  0.3389
## F-statistic:  1034 on 1 and 2014 DF,  p-value: < 2.2e-16
```

(ii) Lasso:

```
x <- as.matrix(data[, -c(1:3,11)])
target <- data$sprtrn

cv_lasso <- cv.glmnet(x, target, alpha = 1, nfolds = 10)

optimal_lambda_value <- cv_lasso$lambda.min

trained_lasso <- glmnet(x, target, alpha = 1, lambda = optimal_lambda_value)

coef(trained_lasso)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -6.713606e-06
## PRC          .
## RET          1.836943e-03
## RETX         1.980419e-03
## vwretd       1.260519e+00
## vwretx       2.686770e-06
## ewretd      -3.076492e-01
## ewretx       .
```

(iii) Ridge regression

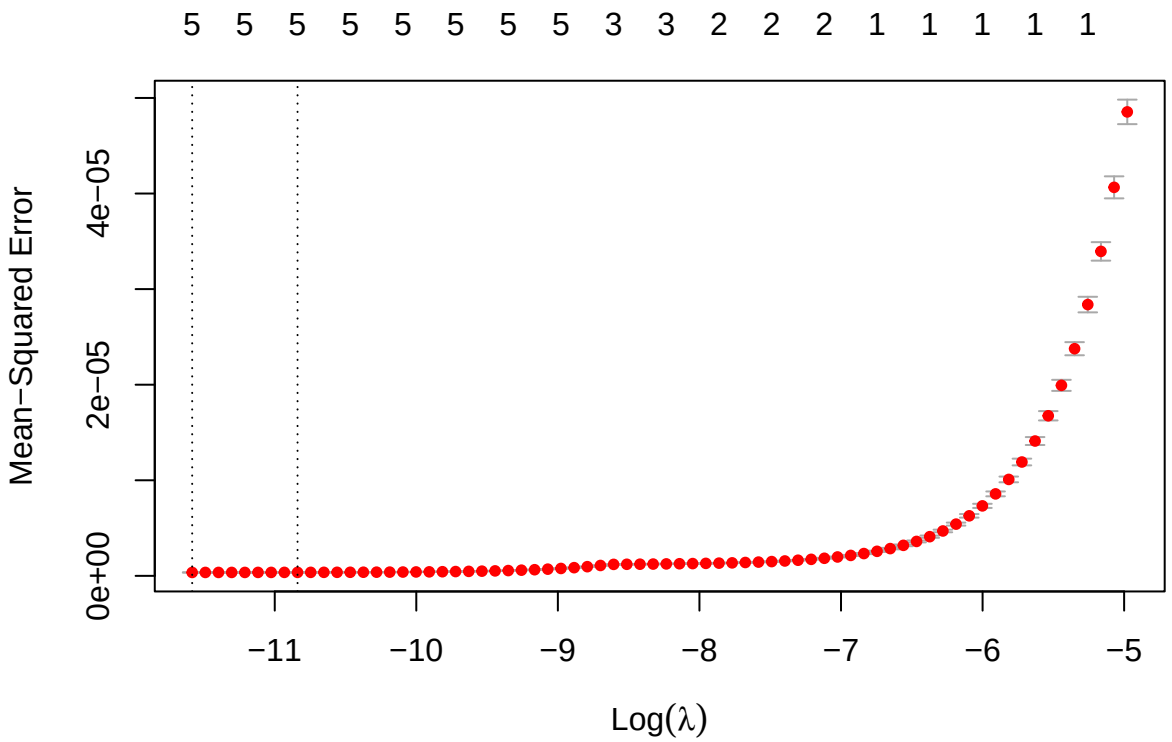
```
x <- as.matrix(data[, -c(1:3,11)])
target <- data$sprtrn

cv_ridge <- cv.glmnet(x, target, alpha = 0, nfolds = 10)

optimal_lambda_value <- cv_ridge$lambda.min
rr <- glmnet(x, target, alpha = 0, lambda = optimal_lambda_value)

coef(rr)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -4.553861e-05
## PRC         -2.268171e-07
## RET          1.820078e-02
## RETX         1.565071e-02
## vwretd       4.605573e-01
## vwretx       4.588090e-01
```

```
## ewretd      -1.110793e-02
## ewretx      -1.330238e-02
```

(b)

```
plot(cv_lasso)
```



The Lasso regression model yields an optimal lambda value of 0.0006889264, with the results indicating five non-zero coefficients. According to the output of -6.71360566391235e-06, 0, 0.00183694258669638, 0.00198041942348896, 1.26051851766217, 2.68676960064891e-06, -0.307649230659877, 0, the coefficients for *PRC* and *ewretx* are not shown, as Lasso regression likely shrunk them to zero due to their lack of significance in the model.

Question 5

```
nikkei_data <- read.table("nikkei_daily.txt", header = TRUE)
sp_data <- read.table("sp_daily.txt", header = TRUE)
ex_data <- read.table("ex_daily.txt", header = TRUE)
```

(a)

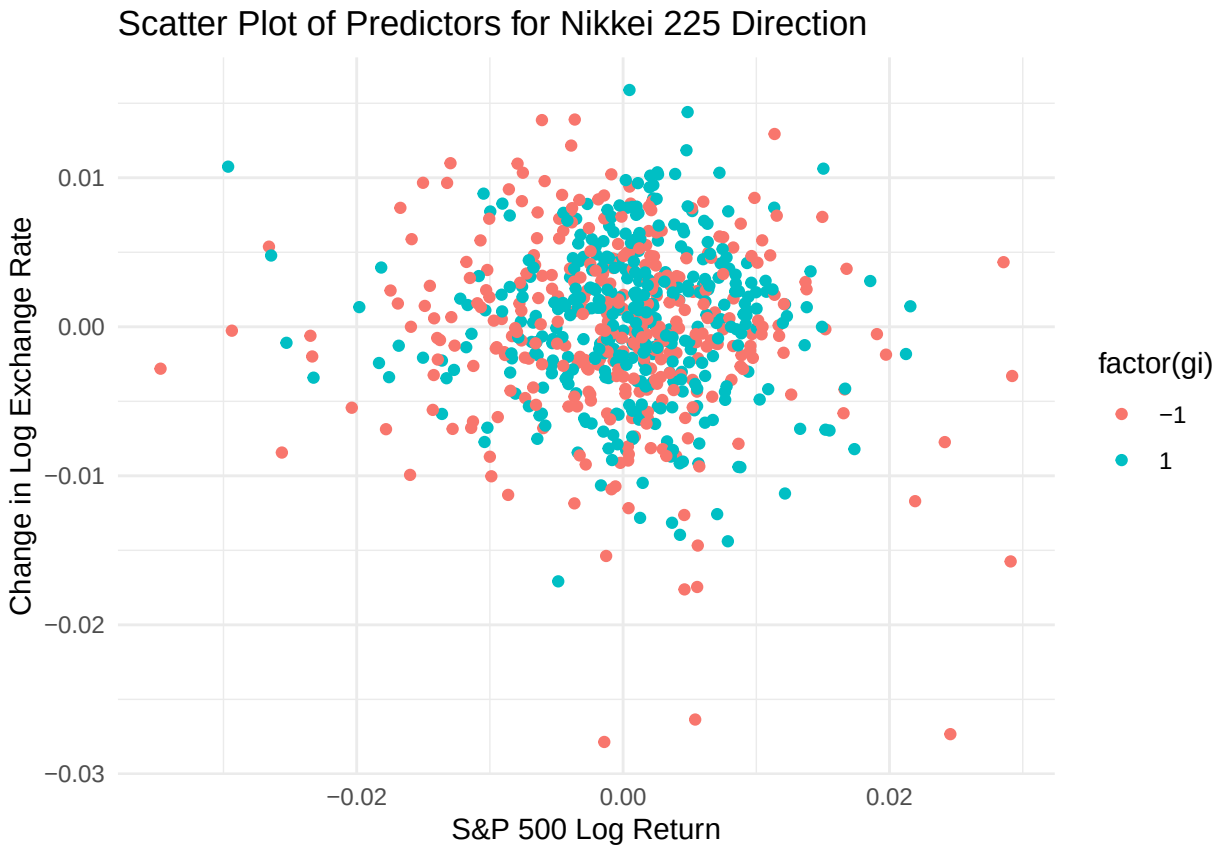
```
# Changing the format
nikkei_data$Date <- format(as.Date(nikkei_data$Date, format = "%d-%b-%Y"), "%Y%m%d")
ex_data$DATE <- format(as.Date(ex_data$DATE, format = "%Y-%m-%d"), "%Y%m%d")

# Output Variable
nikkei_data <- mutate(nikkei_data, gi = c(NA,
                                          ifelse((diff(nikkei_data$Value)) > 0, 1, -1)))
ex_data <- mutate(ex_data, xi2 = c(NA,
```

```
                                    diff(log(lag(ex_data$VALUE, 1)))))

# Merge
data <- merge(nikkei_data, sp_data, by.x="Date", by.y="caldt") %>%
  rename_at("sprtrn", ~"xi1")
data <- merge(data, ex_data, by.x="Date", by.y="DATE")
data <- data[,c("Date","gi", "xi1", "xi2")]
data <- data[data$Date >= "20050103" & data$Date <= "20071231", ]

head(data)
```

```
##         Date gi       xi1          xi2
## 942 20050104  1 -0.011671  0.001459783
## 943 20050105 -1 -0.003628  0.013906550
## 944 20050106  1  0.003506 -0.003073674
## 945 20050107 -1 -0.001431  0.008811474
## 946 20050111  1 -0.006100 -0.005830363
## 947 20050112 -1  0.003981 -0.008664731
```

(b)

```
ggplot(data, aes(x = xi1, y = xi2, color = factor(gi))) +
  geom_point() +
  labs(x = "S&P 500 Log Return",
       y = "Change in Log Exchange Rate",
       title = "Scatter Plot of Predictors for Nikkei 225 Direction") +
  theme_minimal()
```



(c)

(i) Logistic Regression

```
new_data <- data
new_data$gi[new_data$gi == -1] <- 0

lr <- glm(gi ~ xi1 + xi2, data = new_data, family = binomial)
summary(lr)
```

```
##
## Call:
## glm(formula = gi ~ xi1 + xi2, family = binomial, data = new_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.08635    0.07531   1.147   0.2516
## xi1         18.97404    9.69928   1.956   0.0504 .
## xi2         18.89066   13.70866   1.378   0.1682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 986.90  on 712  degrees of freedom
## Residual deviance: 981.47  on 710  degrees of freedom
## AIC: 987.47
```

```
##
## Number of Fisher Scoring iterations: 4
```

```
lr_pred <- predict(lr, type = "response")
lr_class <- ifelse(lr_pred > 0.5, 1, -1)

lr_error <- mean(lr_class != new_data$gi)
cat("Logistic Regression Classification Error Rate:", lr_error, "\n")
```

```
## Logistic Regression Classification Error Rate: 0.6002805
```

(ii) Linear Discriminant Analysis

```
lda_model <- lda(gi ~ xi1 + xi2, data = new_data)
lda_model
```

```
## Call:
## lda(gi ~ xi1 + xi2, data = new_data)
##
## Prior probabilities of groups:
##         0         1
## 0.4768583 0.5231417
##
## Group means:
##              xi1           xi2
## 0 -0.0003306912 -0.0001559156
## 1  0.0007712225  0.0003596026
##
## Coefficients of linear discriminants:
##          LD1
## xi1 108.1278
## xi2 107.8992
```

```
lda_pred <- predict(lda_model)
lda_class <- lda_pred$class

lda_error <- mean(lda_class != new_data$gi)
cat("LDA Classification Error Rate:", lda_error, "\n")
```

```
## LDA Classification Error Rate: 0.454418
```