Assignment1

Michael Nefiodovas(22969312) Carmen Leong(22789943)
Nicholas Choong(21980614)

Question 1

(a) What does reproducible calculations and reproducible simulations refer to and when or why should the calculations/simulations be reproducible?

Reproducible calculation means that the result of the calculation will always be the same when given the same input to the calculation. The ability to reproduce simulations refers to the ability for a two separate users to run a simulation study and receive the same results given that they configure the model with the same parameters. Because the random number generators are deterministic, using and sharing fixed seed values with others will allow them to get the same results.

These are important in situations where you want to share your results with others since they may want to verify your results. If they re-run your code, they can see the results you generated. Reproduciblity is also helpful in debugging programs written to do simulation.

(b) When is it a sensible strategy to use the Gaussian model in a simulation? When is it not and why? (Hint. Your answer could contain an illustrative example.)

It is sensible to use a gaussian given that the random data you're modelling can be reasonably expected to be drawn from a gaussian distribution.

This occurs in 2 main circumstances:

1. When the data being modelled natrually comes from a gaussian (rare) - e.g. assuming gaussian noise on observations.
2. The data you're modelling comes from realisations of sums/means of other random variables. Because of the central limit theorem, we know that the distribution of the sample mean is asymptotically normal. Example: looking at voter preferences in different districts (each district's "mean" outlook on a topic is a sample mean of individual results).

If your support is not all real numbers, then gaussian model is wrong (e.g. wait times in a queueing model). Gaussian distributions also have extremly flat tails, so they often underestimate "rare" events - therefore it might not be a good idea to model systems which have infrequent shocks (e.g. stock prices) with Gaussians. Gaussians are also unimodal & symmetric which may not correpsond to the system you are modelling.

(c) Consider a random sample of observations X = [X1, . . . , Xn]. Why would we expect that, typically, the variables of the sample X are correlated, but the principal component scores obtained from these variables are not?

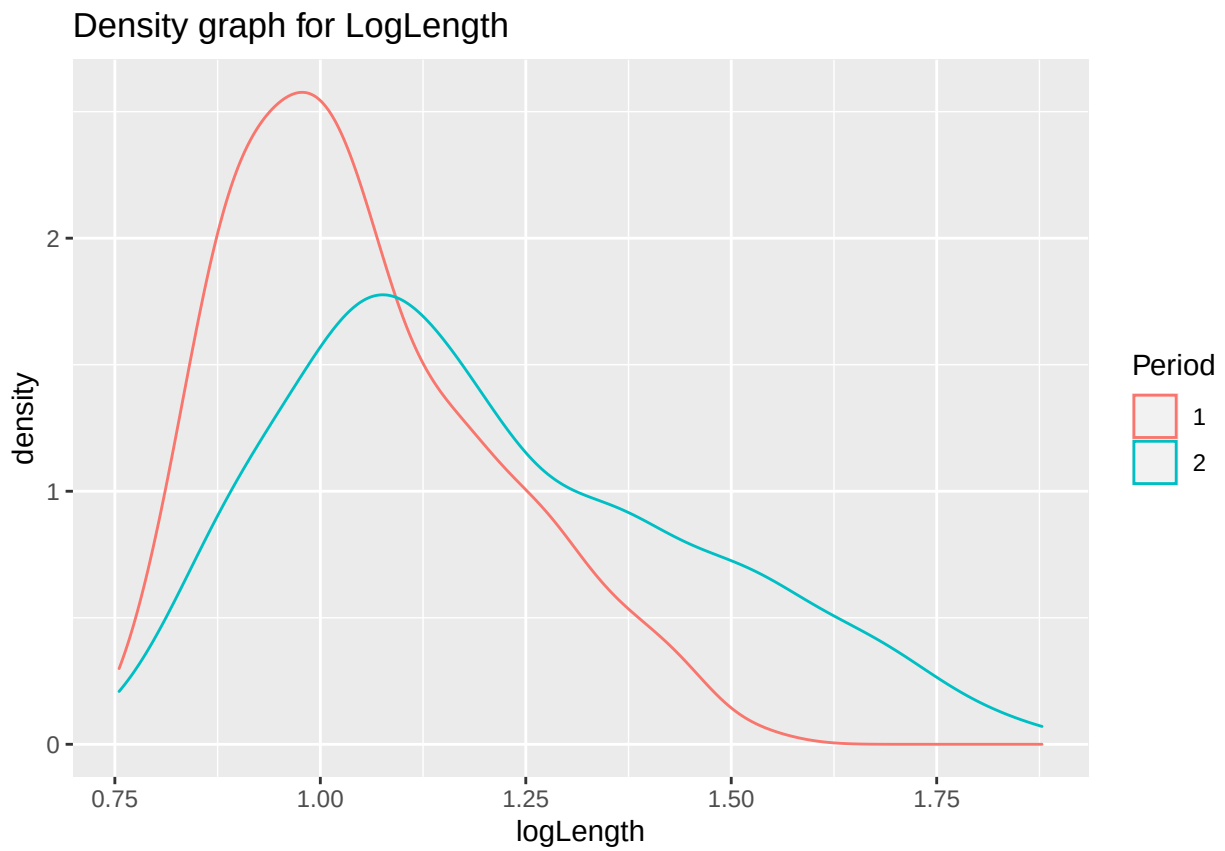
Given an observation Xi, the entries within this observation are likely to be correlated in general. This is because usually an observation is a measurement of some system, likely many aspects of a system are interrelated. For example, if a measurement was of a particular person's health vitals, having no pulse will likely be correlated with other negative health effects (e.g. low blood pressure).

Each "coordinate" in a principal component is orthogonal and the kth pc vector points in the a direction of kth most variance. Because of the orthognonality and because we have centered our data, the resulting PC vectors will be uncorrelated.

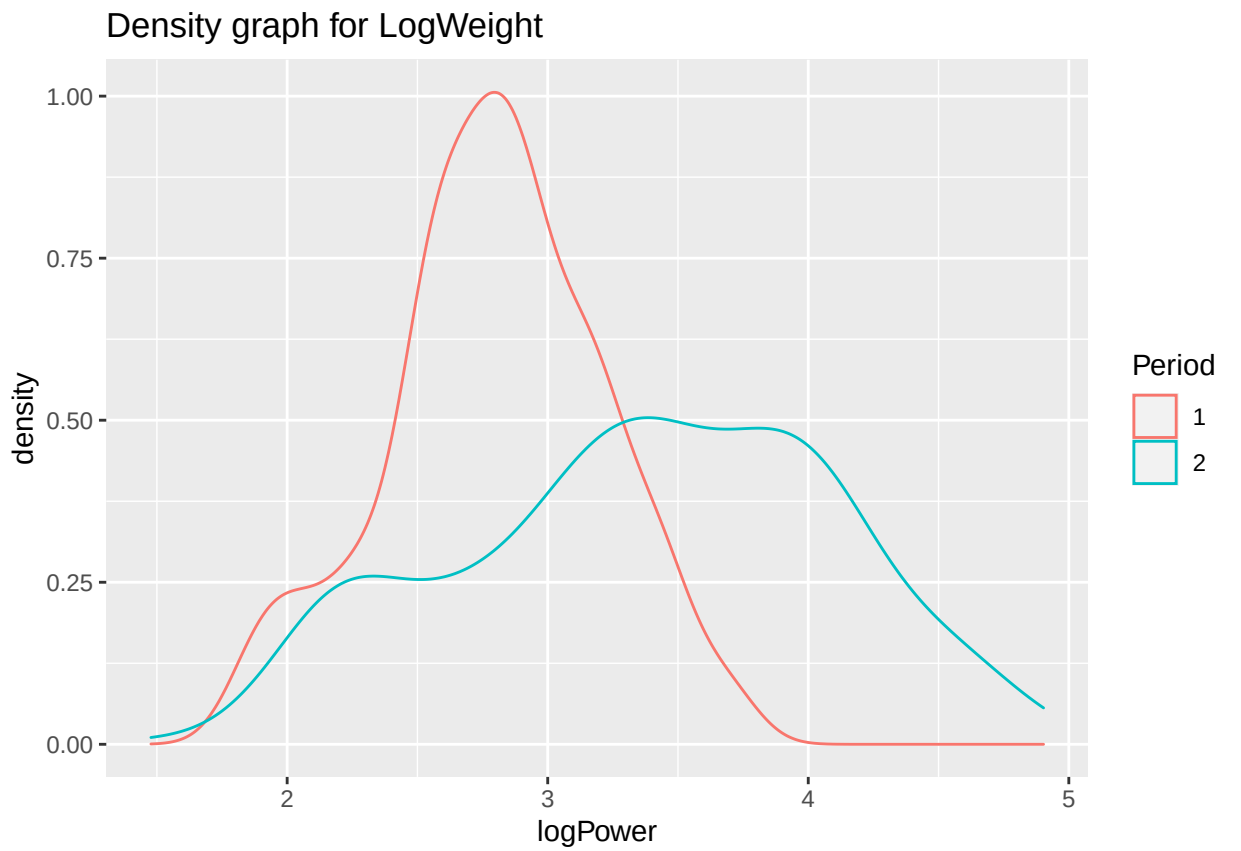
Question 2

(a) Show smoothed histograms of logLength and logPower separately for the two periods. Comment on the shapes of the histograms and how the change over time affects this shape.

```
df <- read.csv("aircraft.csv")
df0 <- data.frame(
  Year = df$Year,
  Period = factor(df$Period),
  logPower = log10(df$Power),
  logSpan = log10(df$Span),
  logLength = log10(df$Length),
  logWeight = log10(df$Weight),
  logSpeed = log10(df$Speed),
  logRange = log10(df$Range)
)

df0 <- within(df0, Period[Year <= 42] <- 1)
df0 <- within(df0, Period[Year > 42] <- 2)

ggplot(df0, (aes(logLength, group = Period, colour = Period))) +
  geom_density() + ggtitle("Density graph for LogLength")
```




```
ggplot(df0, (aes(logPower, group = Period, colour = Period))) +
  geom_density() + ggtitle("Density graph for LogWeight")
```



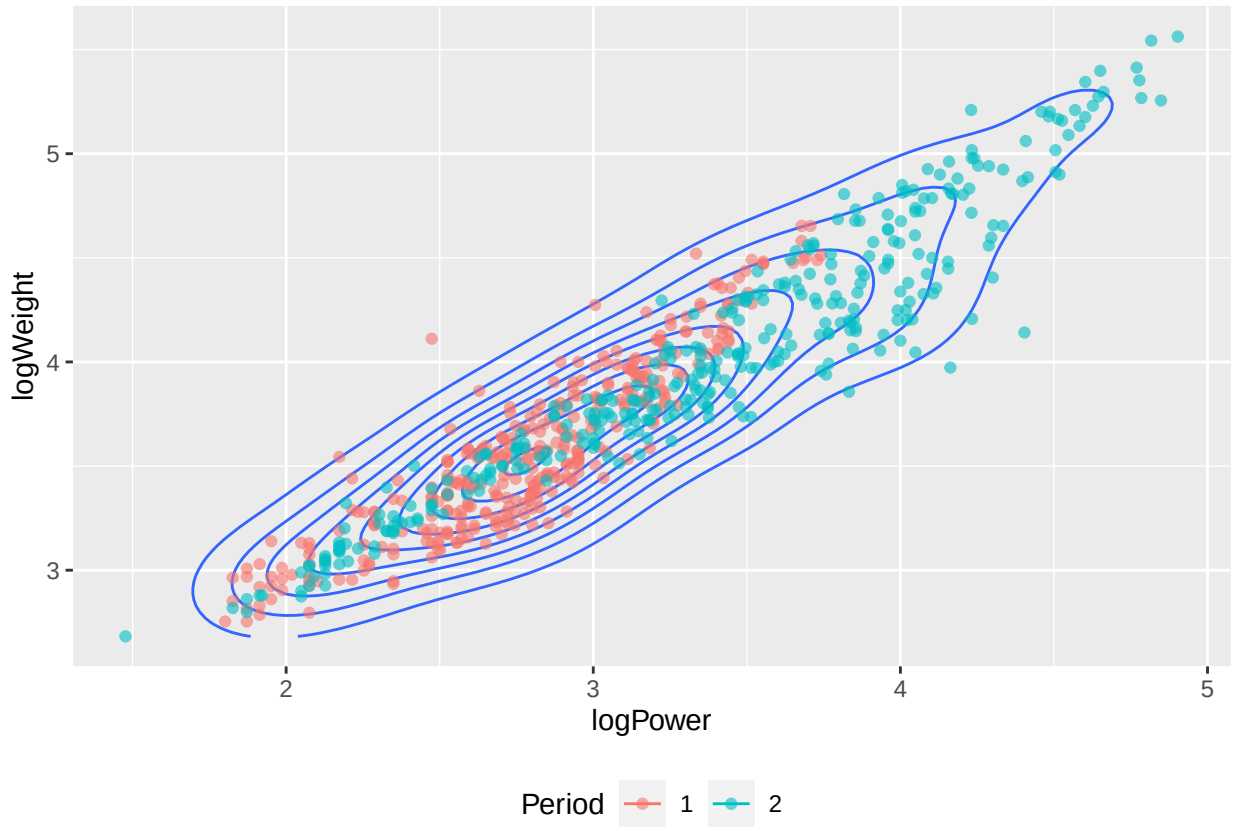
The logLength density graphs for both periods are skewed to the right and are unimodal. However, there are some changes over time on the logLength density graph. The density graph for period 2 is less skewed to the right and is more distributed across the x-axis in the positive direction when compared to the density graph for period 1.

On the other hand, the logPower density graph for period 1 is not skewed and is unimodal. However, the logPower density graph for period 2 is skewed to the left and is bimodal.

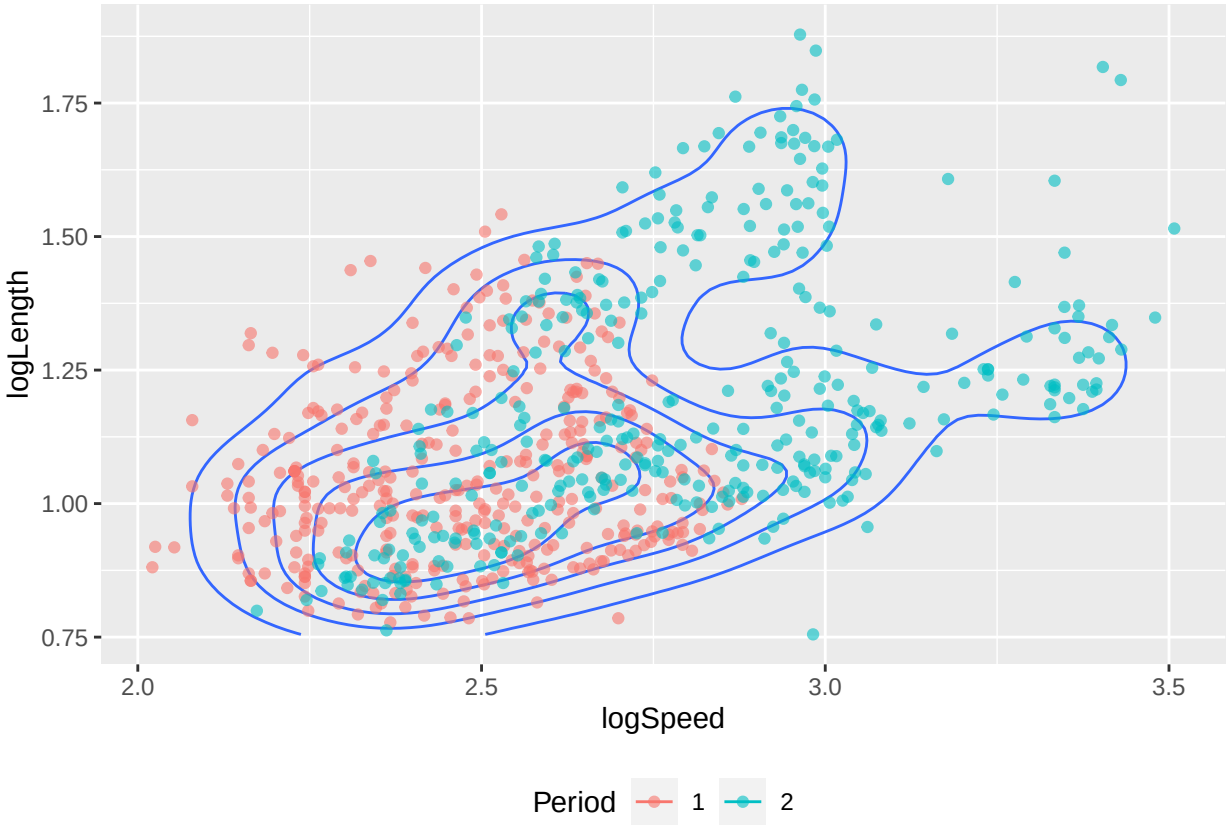
It can also be seen that the range of both logPower and logLength increases over time as shown from the scale on the x-axis.

(b) Construct contour plots of the 2D smoothed histograms of the pairs (logPower, logWeight) and (logSpeed, logLength). Describe the shapes of the density plot and discuss how they change over time.

```
ggplot(df0, aes(logPower, logWeight)) +
  geom_density_2d() +
  geom_point(aes(colour = Period), alpha = 0.6) +
  theme(legend.position = "bottom")
```




```
ggplot(df0, aes(logSpeed, logLength)) +
  geom_density_2d() +
  geom_point(aes(colour = Period), alpha = 0.6) +
  theme(legend.position = "bottom")
```



For the first density plot which represent the 2D smoothed histograms of the pairs (logPower, logWeight). We can see that the graph is unimodel and is skewed in the direction of positive correlation for both periods, indicating a strong positive correlation between logspeed and logweight. It can also be seen that this correlation became stronger over time. The graph also shows that the plots of period 2 are distributed across larger range of logPower and logWeight as compared to period 1, indicating that aircraft are designed to be more powerful and heavier over time.

For the second density plot which represent the 2D smoothed histograms of the pairs (logSpeed, logLength), there's a very low correlation between logSpeed and logLength. We can see that the pattern of the graph is also different for the two periods. For period 1, the graph is more concentrated and can be seen as a unimodel graph. However, for period 2, the graph is more distributed and can be considered as a multimodel graph. It can be understood that, over the time, aircrafts are designed to be more specialised and are of more variation in speed and length, such as faster aircraft or longer aircraft.

(c) For which pair of variables would you expect the largest change in correlation or shape of their density over time and why?

I would expect the pair of variable logSpeed, logLength to have the largest change in correlation or shape of their density over time. The density plot for the pairs (logPower, logWeight) shows consistent and strong correlation between power and weight over the two periods; this may be because more power is required for heavier airplanes. However, the density plot for the pairs (logSpeed, logLength) shows that the correlation between speed and length changes over time as we can see that the graph changes from unimodel to multimodel over the two periods, most possibly because there are little relationship between the length and speed of airplanes.

Question 3

Loading data

Period 1

```
log_aircraft1 <- df0[df0$Period == 1, 3:8]
tail(log_aircraft1)
str(log_aircraft1)
```

Period 2

```
log_aircraft2 <- df0[df0$Period == 2, 3:8]
head(log_aircraft2)
str(log_aircraft2)
```

(a) Separately for the two periods selected in Q2, carry out a principal component analysis using `prcomp` based on the logged data (without scaling).

Centering the data for pca.

Period 1

```
log_aircraft1_pr <- prcomp(log_aircraft1, center = T, scale = F)
log_aircraft1_pr
```

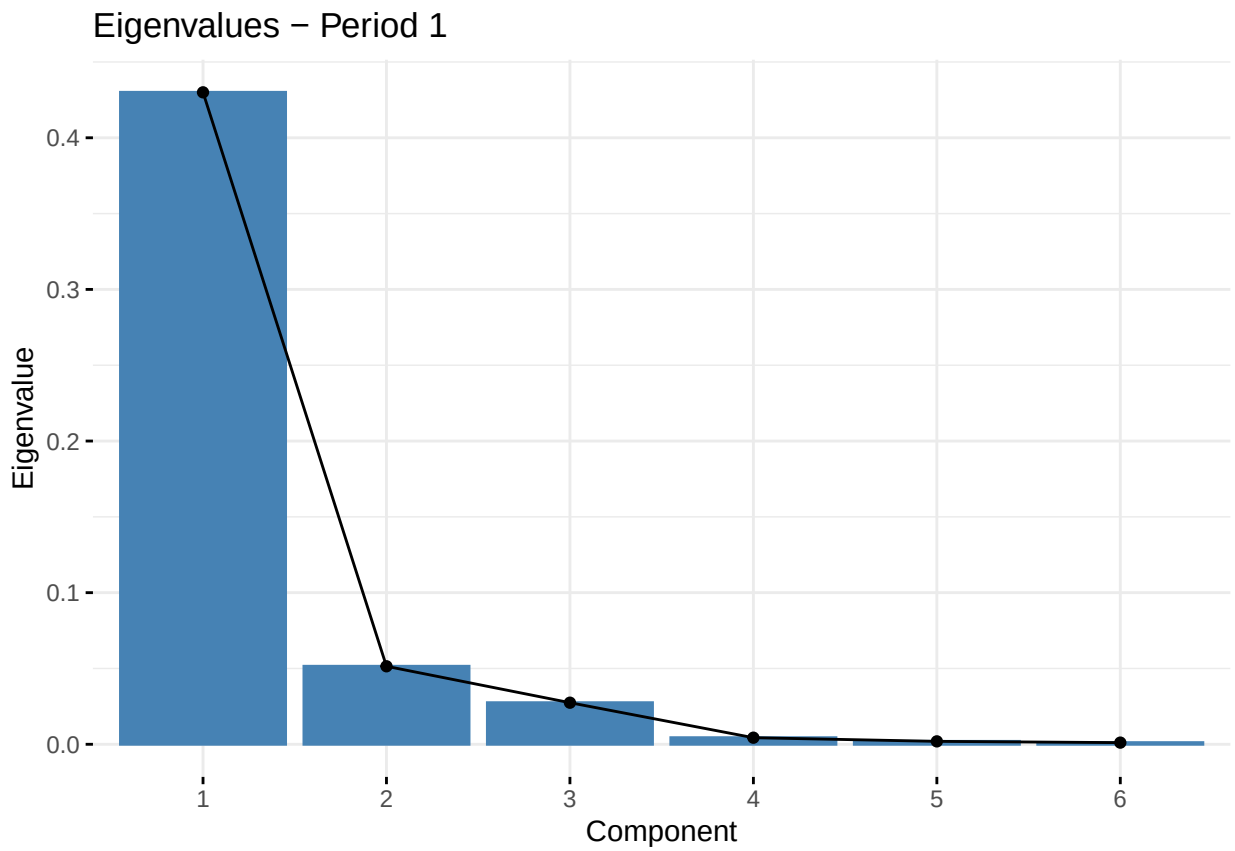
Period 2

```
log_aircraft2_pr <- prcomp(log_aircraft2, center = T, scale = F)
log_aircraft2_pr
```

(b) Show eigenvalues plots for each of the two periods. Interpret the results.

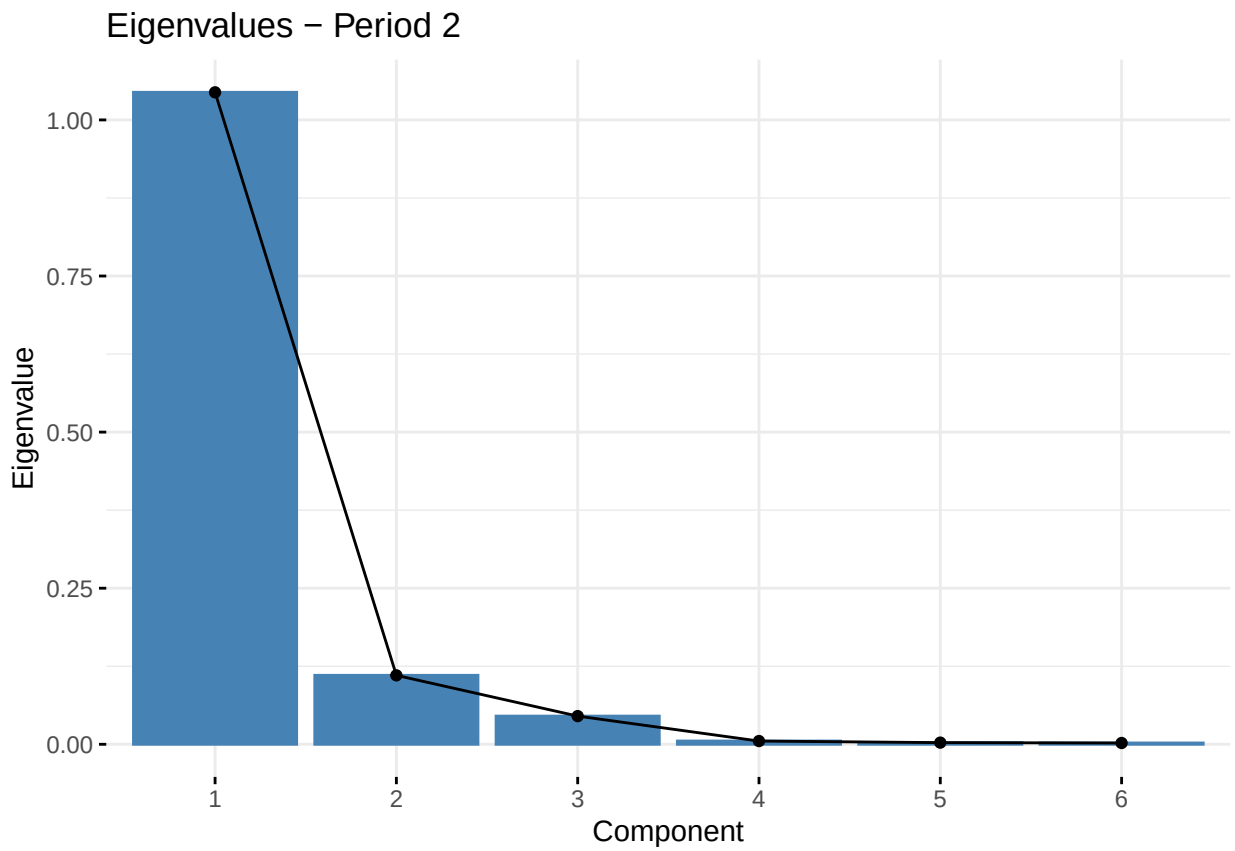
Period 1

```
fviz_screeplot(log_aircraft1_pr, choice = "eigenvalue") +
  ggtitle("Eigenvalues - Period 1") +
  xlab("Component")
```



Period 2

```
fviz_screeplot(log_aircraft2_pr, choice = "eigenvalue") +
  ggtitle("Eigenvalues - Period 2") +
  xlab("Component")
```



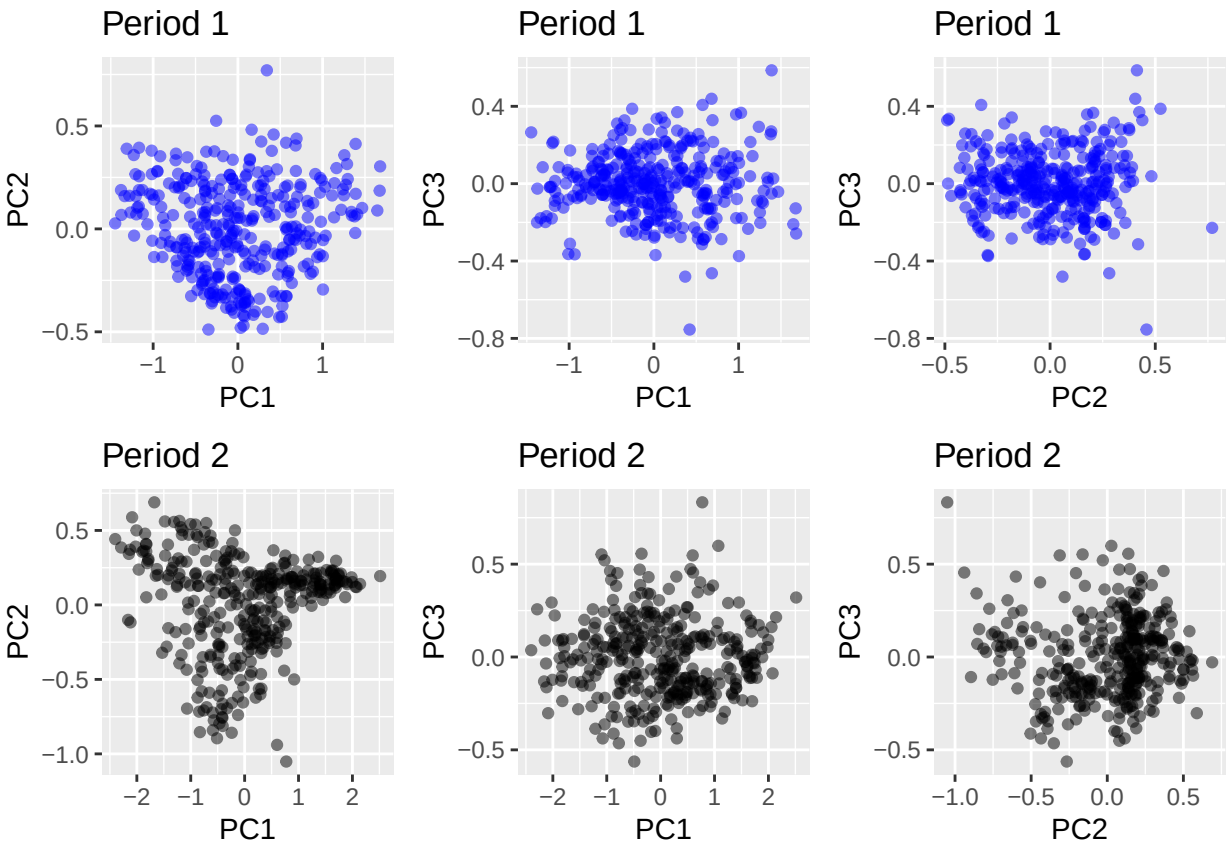
The figures above show that the first components of both plots are greater than 35. These principal components explain 99% of the variation in both periods. The figures also show a kink at the second principal component as we can see that the eigenvalues start to form a horizontal straight line after the second principal component. The first eigenvalue increased by approximately two times in the second period.

(c) Show score plots for the first three PCs for each period. Comment on the results.

Plotting PC2 against PC1 for each period.

```
pc_scores1 <- data.frame(log_aircraft1_pr$x)

pc1pc21 <- ggplot(pc_scores1, aes(x = PC1, y = PC2)) +
  geom_point(alpha = 0.5, colour = "blue") +
  ggtitle("Period 1")

pc1pc31 <- ggplot(pc_scores1, aes(x = PC1, y = PC3)) +
  geom_point(alpha = 0.5, colour = "blue") +
  ggtitle("Period 1")

pc2pc31 <- ggplot(pc_scores1, aes(x = PC2, y = PC3)) +
  geom_point(alpha = 0.5, colour = "blue") +
  ggtitle("Period 1")
```

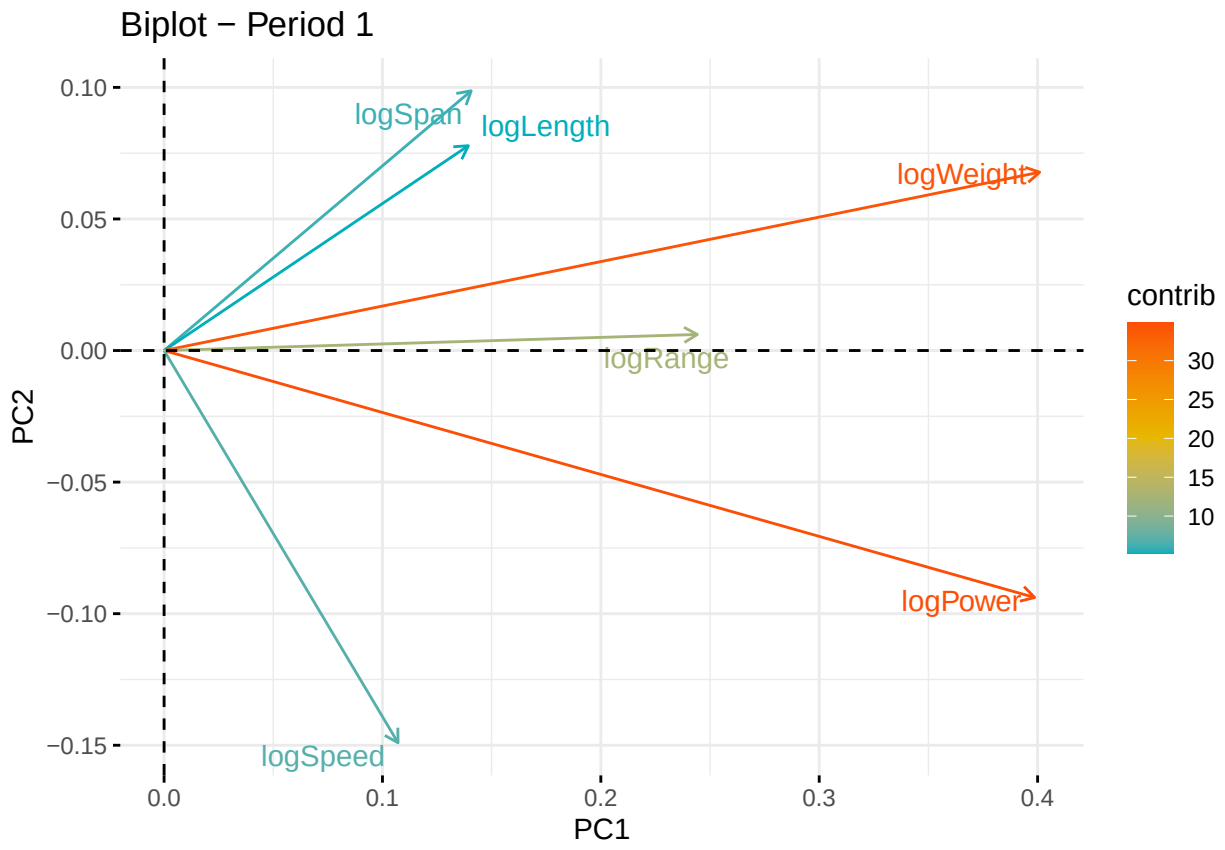
```
pc_scores2 <- data.frame(log_aircraft2_pr$x)

pc1pc22 <- ggplot(pc_scores2, aes(x = PC1, y = PC2)) +
  geom_point(alpha = 0.5) +
  ggtitle("Period 2")

pc1pc32 <- ggplot(pc_scores2, aes(x = PC1, y = PC3)) +
  geom_point(alpha = 0.5) +
  ggtitle("Period 2")

pc2pc32 <- ggplot(pc_scores2, aes(x = PC2, y = PC3)) +
  geom_point(alpha = 0.5) +
  ggtitle("Period 2")


figure <- ggarrange(pc1pc21, pc1pc31, pc2pc31, pc1pc22, pc1pc32, pc2pc32,
  ncol = 3, nrow = 2
)
figure
```



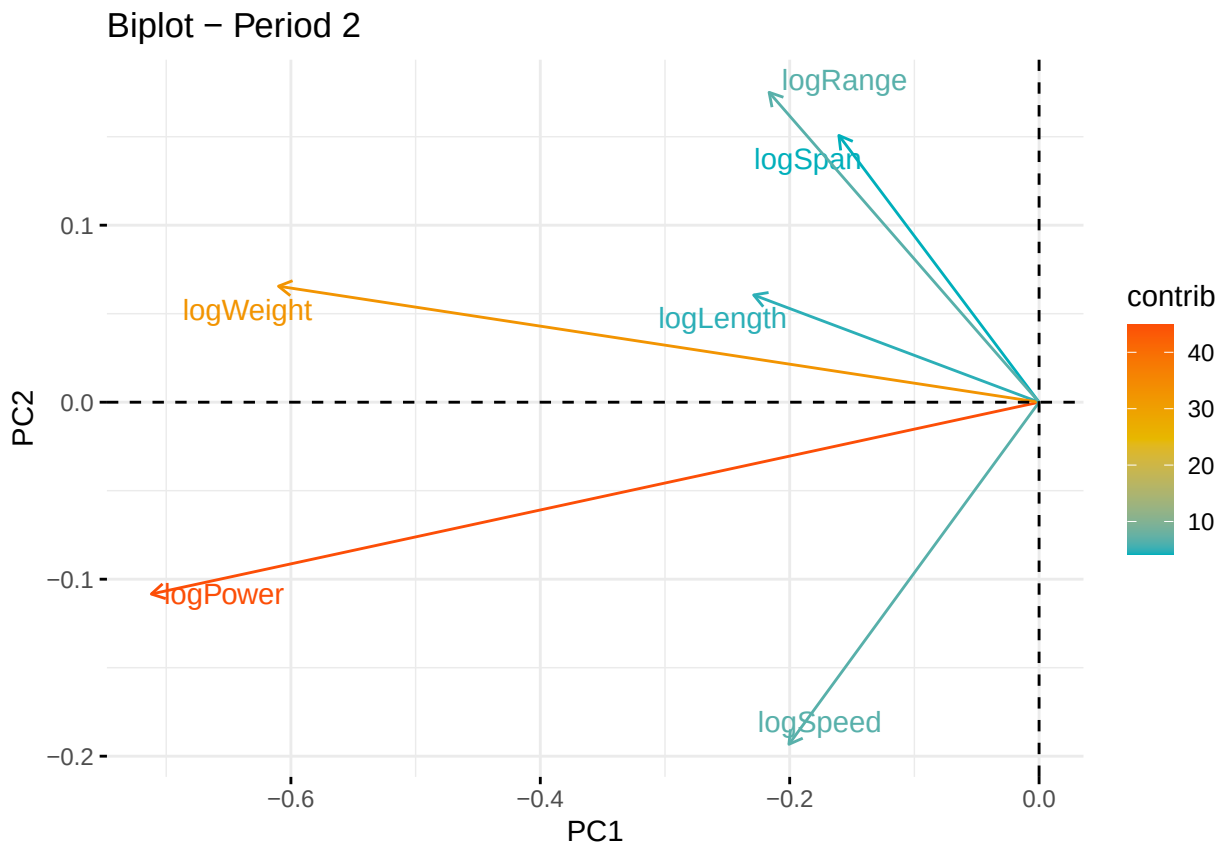
The above figure shows six score plots. The three plots at the top are the score plots for period 1, and the three at the bottom are for period 2. The range of values is larger in period 2 than in period 1, which also means that the observations are more spread out in period 2. We can see that the observations are more concentrated on the point of origin in period 1 than in period 2. The PC1/PC2 plots look like the letter T, and the shape looks more distinctive in period 2.

(d) Which logged variable contributes most to PC1 for each period? Does this change across the two periods? Comments on the results.

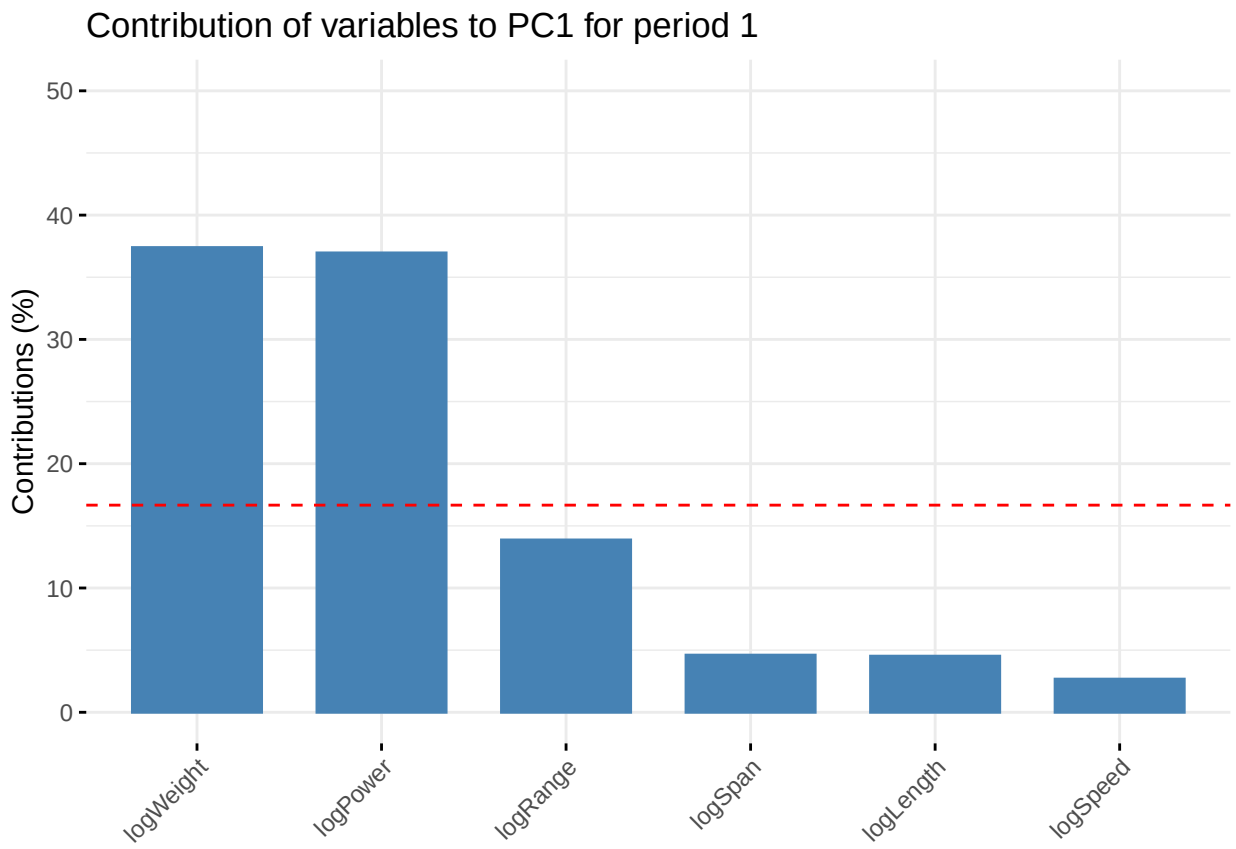
```
fviz_pca_var(log_aircraft1_pr,
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE # Avoid text overlapping
) +
  labs(x = "PC1", y = "PC2") +
  ggtitle("Biplot - Period 1")
```



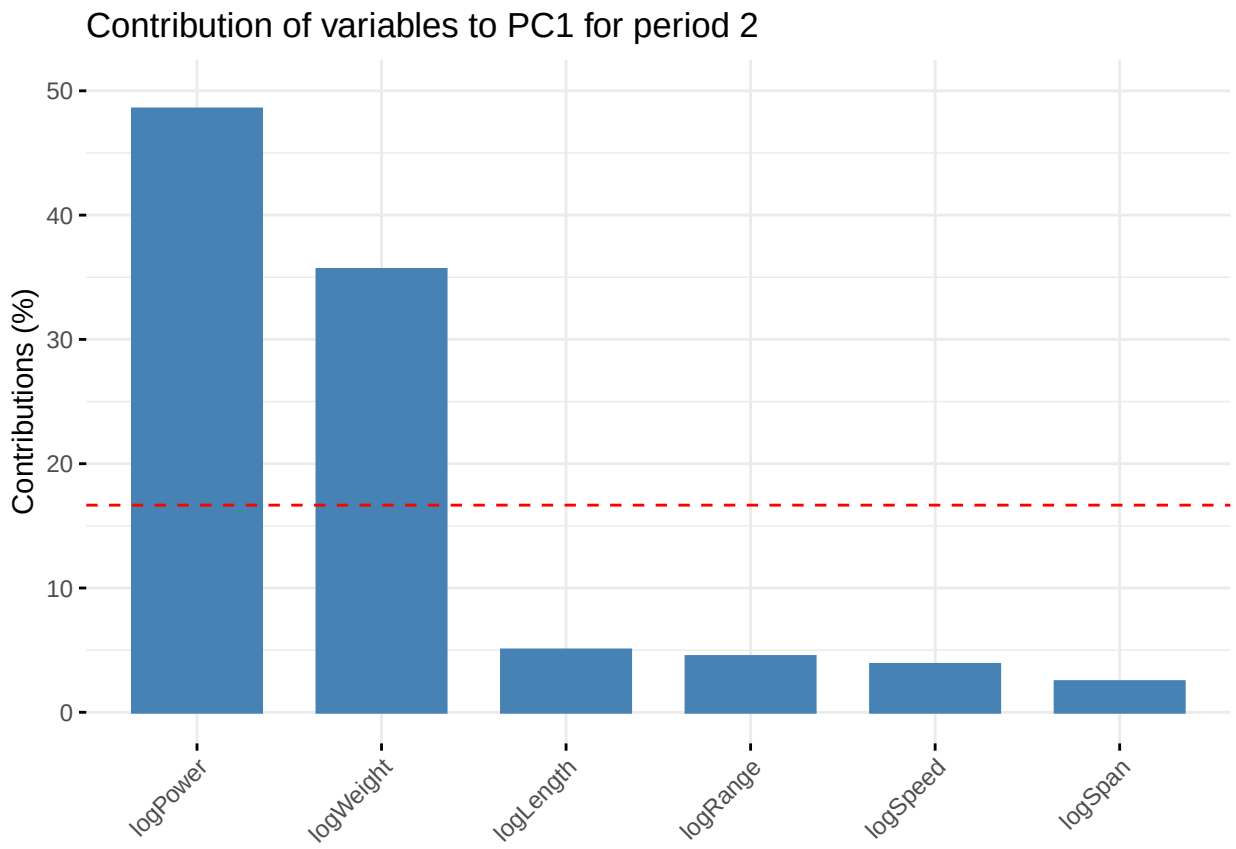

```
fviz_pca_var(log_aircraft2_pr,
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE # Avoid text overlapping
) +
  labs(x = "PC1", y = "PC2") +
  ggtitle("Biplot - Period 2")
```




```
fviz_contrib(log_aircraft1_pr, choice = "var", axes = 1) +
  ggtitle("Contribution of variables to PC1 for period 1") +
  ylim(0, 50)
```




```
fviz_contrib(log_aircraft2_pr, choice = "var", axes = 1) +
  ggtitle("Contribution of variables to PC1 for period 2") +
  ylim(0, 50)
```




```
log_aircraft1_pr$rotation[, "PC1"] # Period 1
```

```
##  logPower   logSpan logLength logWeight  logSpeed  logRange
## 0.6079327 0.2144726 0.2124951 0.6114775 0.1633870 0.3723778
```

```
log_aircraft2_pr$rotation[, "PC1"] # Period 2
```

```
##   logPower    logSpan  logLength  logWeight   logSpeed   logRange
## -0.6967009 -0.1571265 -0.2240067 -0.5969116 -0.1963363 -0.2118695
```

logWeight contributed the most in the first period, whereas logPower contributed the most in the second. The contribution from logPower increased drastically in period 2, and logRange has lesser contributions in period 2 than in period 1. The contributions from the last four variables seem insignificant compared to those from logWeight and logPower.

(e) Based on your analysis, discuss the main changes that have occurred over time.

Weight and power are strongly positively correlated and are likely to increase even more. The correlation between speed and range decreases over time and is now uncorrelated. This might be due to aircraft being more specialised for speed or range. Companies may have shifted away from increasing the range as aircraft can only travel so far before reaching the opposite side of Earth.

Question 4

The data set `ass2pop.csv` is available in the LMS folder 'Data sets'. It contains the means and covariance matrices corresponding to two populations. The first and second column of ass2pop.csv are the means μ_1 and μ_2 of the first and second population respectively; columns 3:22 correspond to the covariance matrix Σ_1 of the first population, and the remaining columns correspond to the covariance matrix Σ_2 of the second populations. In this question we generate random samples from these populations as described below.

Normal Distribution Analysis

```
set.seed(2733)
```

```
ass2pop <- read.csv("ass2pop.csv", header = FALSE)
mean1 <- ass2pop$V1
mean2 <- ass2pop$V2

cov1 <- data.matrix(subset(ass2pop, select = V3:V22))
cov2 <- data.matrix(subset(ass2pop, select = V23:V42))

print(dim(cov1))
```

```
## [1] 20 20
```

(a) Read the data into R. What is the dimension of the covariance matrix Σ_1?

The covariance matrix is 20×20.

```
set.seed(27312)
N1 <- 250
N2 <- 150
eig <- list()

samples1 <- rmvnorm(N1, mean = mean1, sigma = cov1)
samples2 <- rmvnorm(N2, mean = mean2, sigma = cov2)
samples <- rbind(samples1, samples2)
samplecov <- cov(samples)
print(round(samplecov, 2))
eigenresult <- eigen(samplecov)
vals <- list(eigenresult$values)
eig <- append(eig, vals)
save(vals, file = "./eigens/normal_eigen/1.RData")

for (i in 2:50) {
  samples1 <- rmvnorm(N1, mean = mean1, sigma = cov1)
  samples2 <- rmvnorm(N2, mean = mean2, sigma = cov2)
  samples <- rbind(samples1, samples2)
  samplecov <- cov(samples)
  eigenresult <- eigen(samplecov)
  vals <- list(eigenresult$values)
  eig <- append(eig, vals)
```

```
  filename <- paste("./eigens/normal_eigen/", toString(i), ".RData", sep = "")
  save(vals, file = filename)
}

mat <- t(do.call("cbind", eig))
meaneig <- colMeans(mat)

m <- as.data.frame(meaneig)
m["x"] <- 1:20
colnames(m) <- c("value", "Var1")
m["Var2"] <- rep(0, 20)

data <- melt(t(mat))
p <- data %>%
  ggplot(aes(x = Var1, y = value, group = Var2)) +
  geom_line(alpha = 0.1) +
  geom_line(data = m, colour = "red")

print(p)
```

```
print("----")
print(eig)
print(round(meaneig, 3))
```

(b) Generate 250 random samples from the Gaussian distribution$N \sim (\mu_1, \Sigma_1)$ and 150 samples from the Gaussian distribution $N \sim (\mu_2, \Sigma_2)$. What is the size of the data matrix consisting of these random samples? Calculate the sample covariance matrix S of the random samples, and find eigenvalues of S. Save the vector of eigenvalues into a file for later analysis.

What is the size of the data matrix consisting of these random samples?
First sample: 250 x 20
Second sample: 150 x 20
Therefore, in total, our sample is: 400 x 20

Calculate the sample covariance matrix S of the random samples:

Done in code

(d) Calculate the mean vector of eigenvalues over the 50 repetitions and list/print this mean vector.

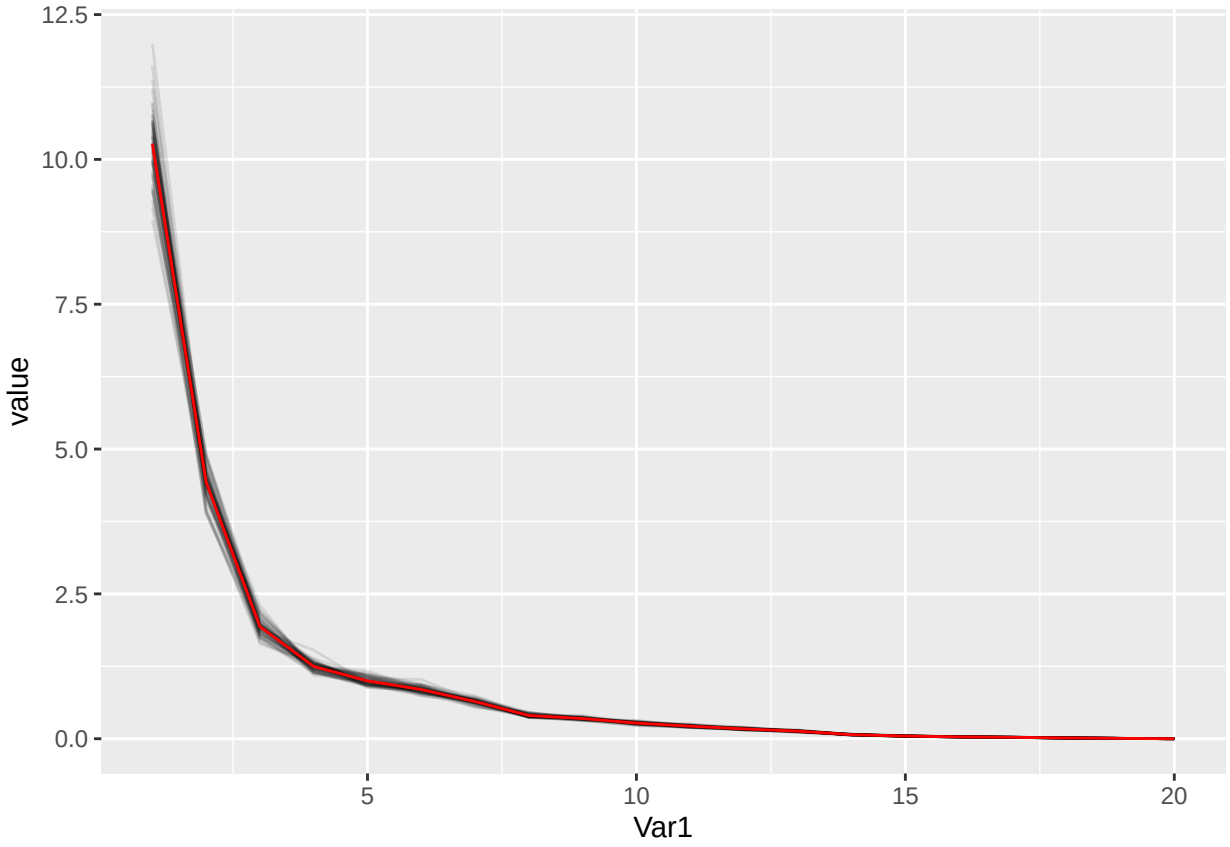
```
print(round(meaneig, 3))
```

```
##  [1] 10.272  4.427  1.935  1.248  0.998  0.849  0.642  0.401  0.348  0.270
## [11]  0.216  0.170  0.130  0.070  0.046  0.032  0.024  0.013  0.006  0.000
```

(e) Display the 50 vectors of eigenvalues and their mean vector in an eigenvalue or scree plot. How similar are these eigenvalue plots? Where does the largest deviation from the mean vector occur?

They are very similar, the largest deviation occurs in the first PC which has a range of about 9-11.

```
print(p)
```



Student-t Distribution Analysis

```
set.seed(27312)
N1 <- 250
N2 <- 150
df1 <- 10
df2 <- 3
eig <- list()
for (i in 1:50) {
  scale1 <- ((df1 - 2) / df1) * cov1
  scale2 <- ((df2 - 2) / df2) * cov2
  samples1 <- mean1 + rmvt(N1, sigma = scale1, df = df1)
  samples2 <- mean2 + rmvt(N2, sigma = scale2, df = df2)
  samples <- rbind(samples1, samples2)
  samplecov <- cov(samples)
```

```
  eigenresult <- eigen(samplecov)
  vals <- list(eigenresult$values)
  filename <- paste("./eigens/t_eigen/", toString(i), ".RData", sep = "")
  save(vals, file = filename)
  eig <- append(eig, vals)
}

mat <- t(do.call("cbind", eig))
meaneig1 <- colMeans(mat)

m <- as.data.frame(meaneig1)
m["x"] <- 1:20
colnames(m) <- c("value", "Var1")
m["Var2"] <- rep(0, 20)

data <- melt(t(mat))
p1 <- data %>%
  ggplot(aes(x = Var1, y = value, group = Var2)) +
  geom_line(alpha = 0.1) +
  geom_line(data = m, colour = "red")

print(p1)
```

```
print(round(meaneig1, 3))
```

(f) Repeat parts (b) to (e) with 250 samples from the t-distribution $t_{10}(\mu_1, \Sigma_{01})$ and 150 samples from t-distribution $t_3(\mu_2, \Sigma_{02})$. (Hint. Σ_{0k} is the scale matrix which is obtained from the covariance matrix Σ_k using the following relationship $\Sigma_k = \frac{\nu}{\nu-2}\Sigma_{0k}$, with ν the degree of freedom of the t-distribution and $k = 1$ and 2 here.)

Calculate the mean vector of eigenvalues over the 50 repetitions and list/print this mean vector.

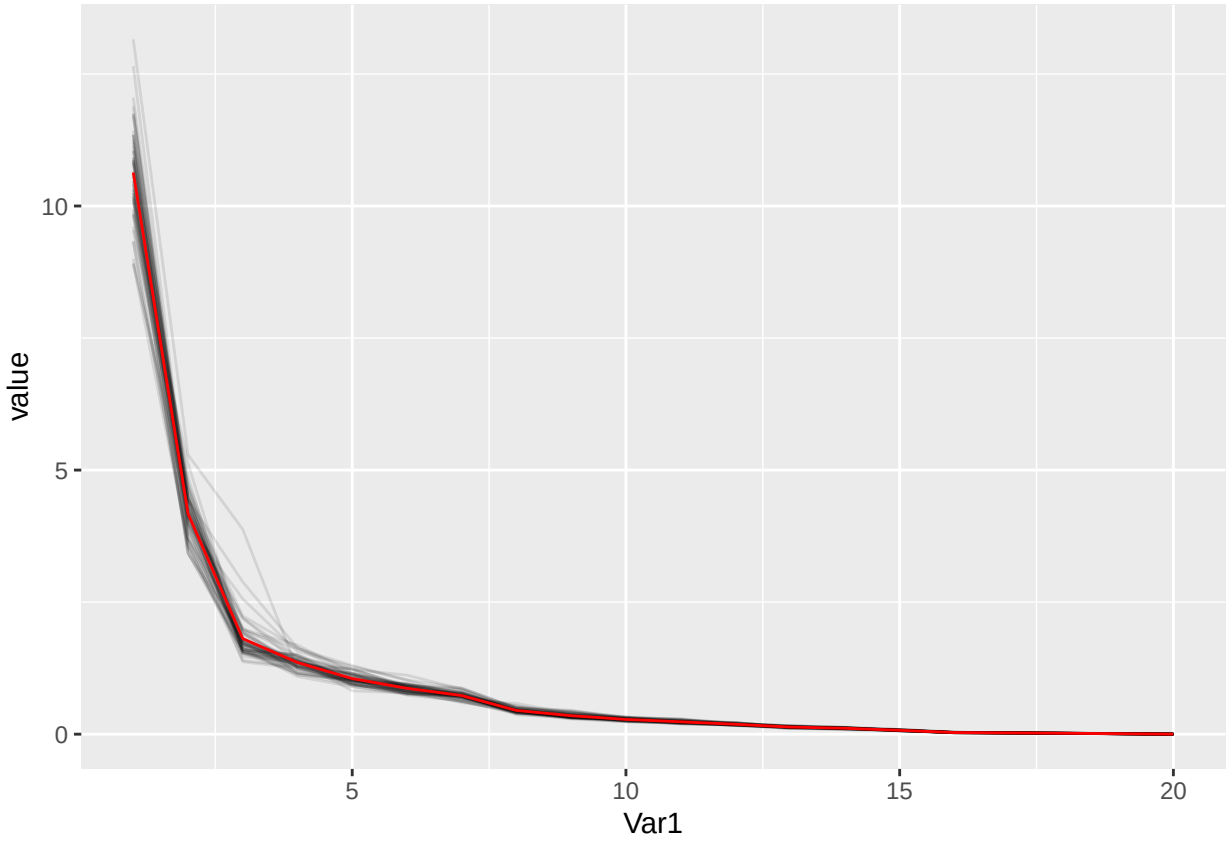
```
print(round(meaneig1, 3))
```

```
##  [1] 10.634  4.176  1.808  1.357  1.052  0.869  0.731  0.449  0.350  0.280
## [11]  0.234  0.187  0.135  0.112  0.073  0.031  0.024  0.017  0.009  0.000
```

Display the 50 vectors of eigenvalues and their mean vector in an eigenvalue or scree plot. How similar are these eigenvalue plots? Where does the largest deviation from the mean vector occur?

The largest deviation occurs in the first PC which is approximately 4. It's varies quite a lot (range is about 8-35).

```
print(p1)
```



(g) Compare the results of the two different simulations and comment on interesting findings and differences between them. Why do we expect differences between the pairs of simulations?

The t-distribution has much more variance in the generated scree plots. Additionally, the t-distribution has much more of the total variance captured by the first eigenvalue whereas the normal distribution has a shallower gradient. In other words, the first mean eigenvalue in the normal is smaller than the first mean eigenvalue in the t-distribution case.

We expect to see differences because the two distributions are different and hence the covariances will be different. Additionally, t-distributions have much fatter tails than normals so we expect more "outliers" and thus more total variance.