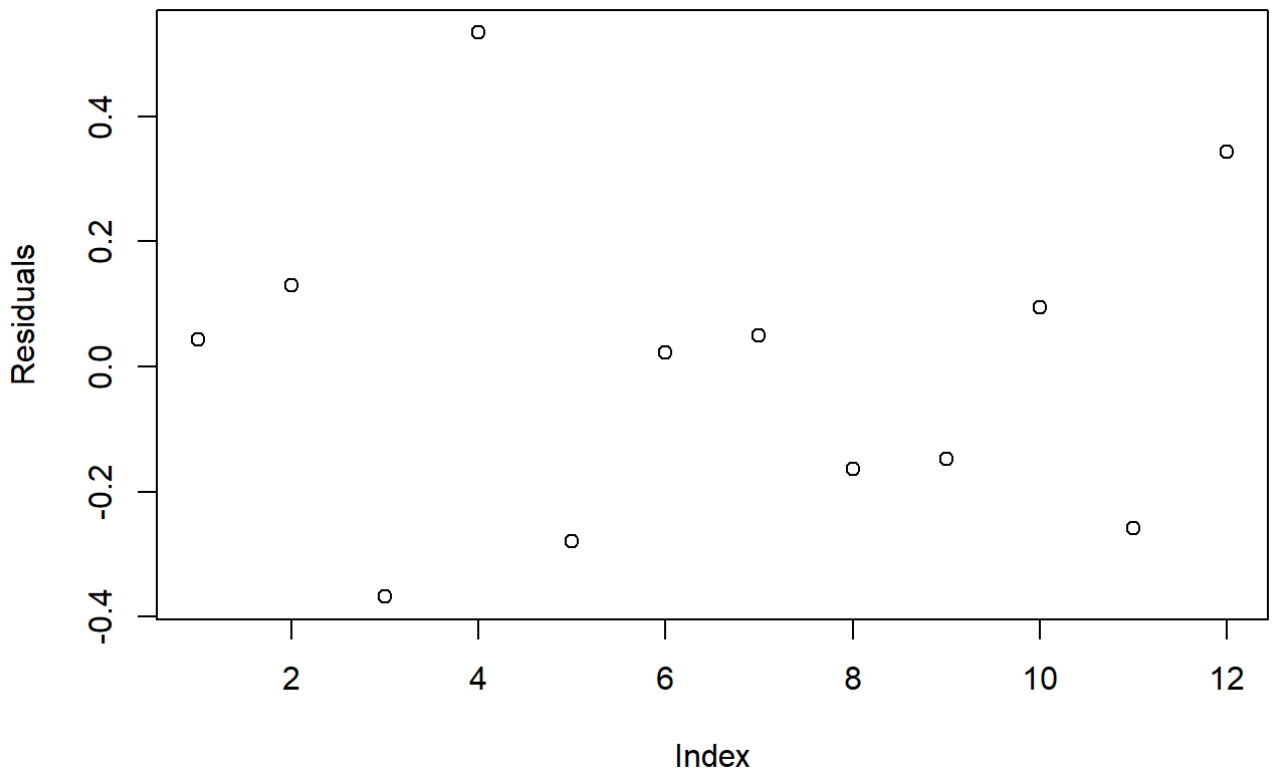
Tutorial 3-4 Exercise Solution

Regression:

```
library(readxl)
data_reg <- read_excel("./example_linear-reg_dataset1.xls", sheet = "Hoja3")
lmHeight = lm(height~age + playtime, data = data_reg)
summary(lmHeight)
```

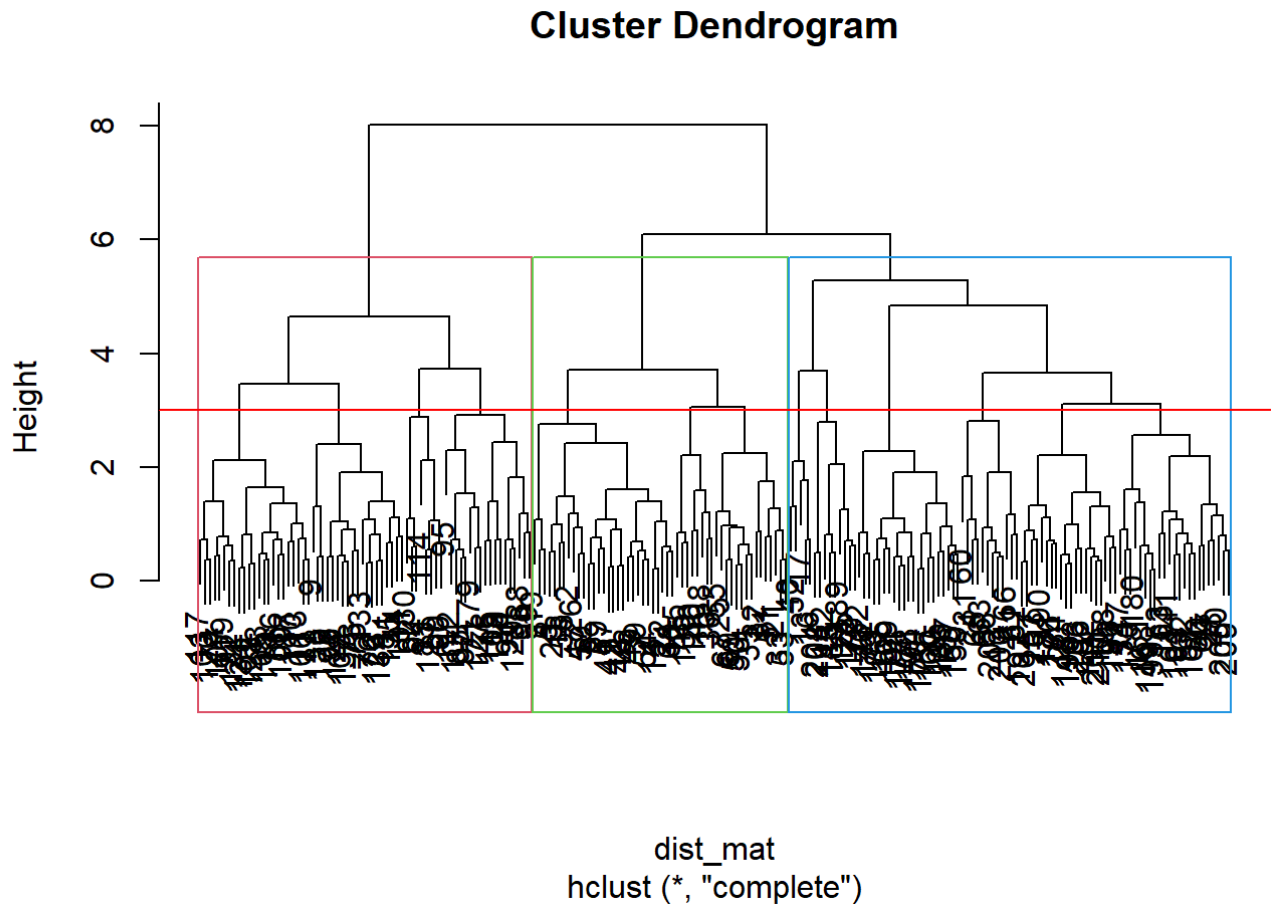
```
##
## Call:
## lm(formula = height ~ age + playtime, data = data_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36793 -0.18717  0.03282  0.10406  0.53356
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 56.48302    0.41013 137.720 2.85e-16 ***
## age          0.98463    0.02434  40.453 1.72e-11 ***
## playtime     0.01388    0.03276   0.424    0.682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2909 on 9 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9933
## F-statistic: 819.8 on 2 and 9 DF,  p-value: 6.564e-11
```

```
plot(lmHeight$residuals, ylab = "Residuals") #the residual plot should look random - should'nt show a pattern
```



Clustering:

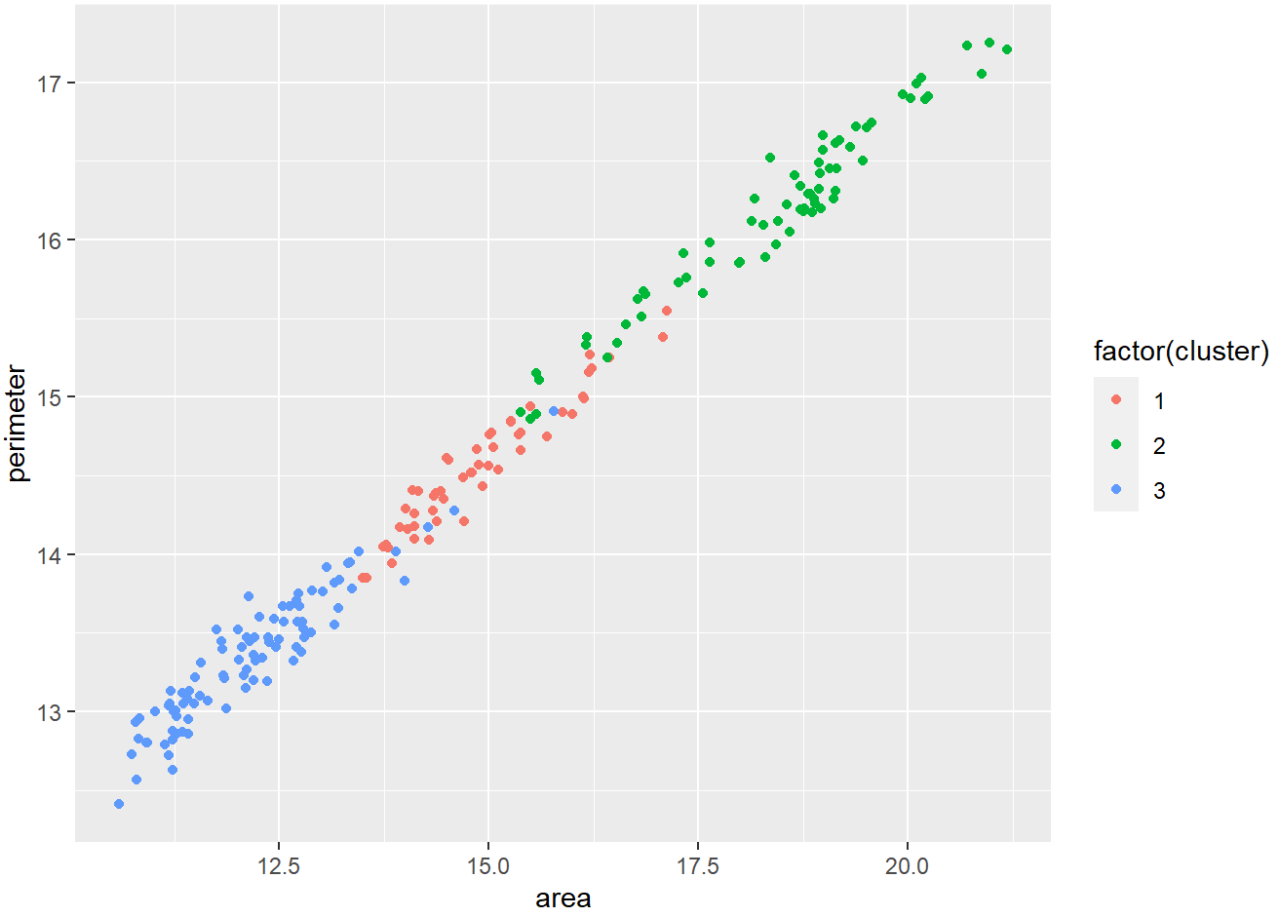
```
library(devtools)
library(ggbiplot)
library(dplyr)
library(ggplot2)

data_seeds <- read.csv("./seeds_dataset.txt" ,sep = '\t',header = FALSE)
feature_name <-
c('area','perimeter','compactness','length.of.kernel','width.of.kernal','asymmetry
.coefficient','length.of.kernel.groove','type.of.seed')
colnames(data_seeds) <- feature_name
seeds_label <- data_seeds$type.of.seed
data_seeds$type.of.seed <- NULL
data_seeds_norm <- as.data.frame(scale(data_seeds))
dist_mat <- dist(data_seeds_norm, method = 'euclidean')
hclust_complete <- hclust(dist_mat, method = 'complete')
cut_complete <- cutree(hclust_complete, k = 3)
plot(hclust_complete)
rect.hclust(hclust_complete, k = 3, border = 2:6)
abline(h = 3, col = 'red')
```




```
data_seeds_cl <- mutate(data_seeds, cluster = cut_complete)
count(data_seeds_cl,cluster)
```

```
##   cluster  n
## 1       1 52
## 2       2 68
## 3       3 90
```

```
ggplot(data_seeds_cl, aes(x=area, y = perimeter, color = factor(cluster))) +
geom_point()
```



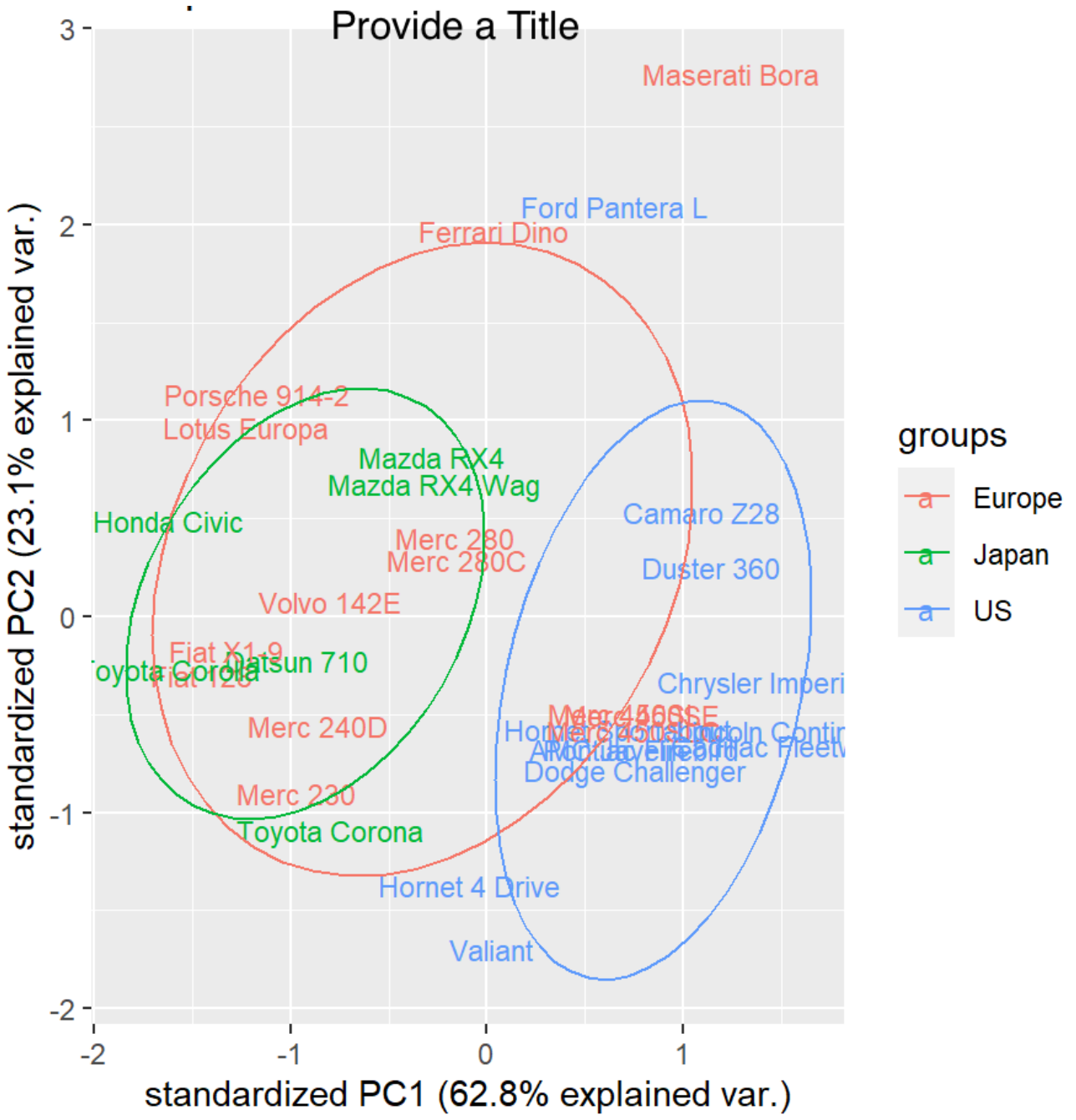

```
table(data_seeds_cl$cluster,seeds_label)
```

```
##    seeds_label
##       1  2  3
##   1 48  4  0
##   2  2 66  0
##   3 20  0 70
```

Conclusion: Hierarchical clustering treats each data point as a singleton cluster, and then successively merges clusters until all points have been merged into a single remaining cluster. In complete linkage hierarchical clustering, in each step two clusters with the smallest maximum pairwise distance are merged. Here, the distance between groups is defined as the distance between the most distant pair of objects, one from each group (think about this using the above figure). Whereas, the Average linkage method does not correspond to the idea of clusters as compact objects. Here, the distance between two clusters is defined as the average of distances between all pairs of objects, where each pair is made up of one object from each group.

Also, you needed to talk about the differences you observed between the two results from average and complete methods.

```
mtcars.pca <- prcomp(mtcars[,c(1:7,10,11)], center = TRUE,scale. = TRUE)
mtcars.country <- c(rep("Japan", 3), rep("US",4), rep("Europe", 7),rep("US",3),
"Europe", rep("Japan", 3), rep("US",4), rep("Europe", 3), "US", rep("Europe", 3))
ggbiplot(mtcars.pca,ellipse=TRUE, labels=rownames(mtcars), groups=mtcars.country,
var.axes = FALSE) + ggtitle("Provide a Title")
```



Classification:

```
require(ISLR)
glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3, data = Smarket, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3, family = binomial,
##     data = Smarket)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.384  -1.204   1.077   1.146   1.348
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.074230   0.056672   1.310    0.190
## Lag1        -0.071404   0.050102  -1.425    0.154
## Lag2        -0.044260   0.050019  -0.885    0.376
## Lag3         0.008873   0.049855   0.178    0.859
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1728.4  on 1246  degrees of freedom
## AIC: 1736.4
##
## Number of Fisher Scoring iterations: 3
```

```
glm.probs <- predict(glm.fit,type = "response")
glm.pred <- ifelse(glm.probs > 0.5, "Up", "Down")
attach(Smarket)
table(glm.pred,Direction)
```

```
##         Direction
## glm.pred Down  Up
##     Down  114  97
##     Up    488 551
```

```
mean(glm.pred == Direction)
```

```
## [1] 0.532
```

The classification rate is 53.2%, which is slightly better than the original model we tried in the classification tutorial.