# Chapter 6: Workload Characterization Techniques

Nicolas Kolling Ribas

### 6.1 The CPU time and disk I/O's of seven programs are shown in Table 6.7. Determine the equation for principal factors.

Load the dataset

```
dataset <- read.csv("table-6.7.csv")
print(dataset)
```

```
##   Program.Name    Function CPU.Time I.O.s
## 1          TKB      Linker       14  2735
## 2          MAC   Assembler       13   253
## 3        COBOL    Compiler        8    27
## 4        BASIC    Compiler        6    27
## 5       Pascal    Compiler        6    12
## 6          EDT Text editor        4    91
## 7          SOS Text editor        1    33
```

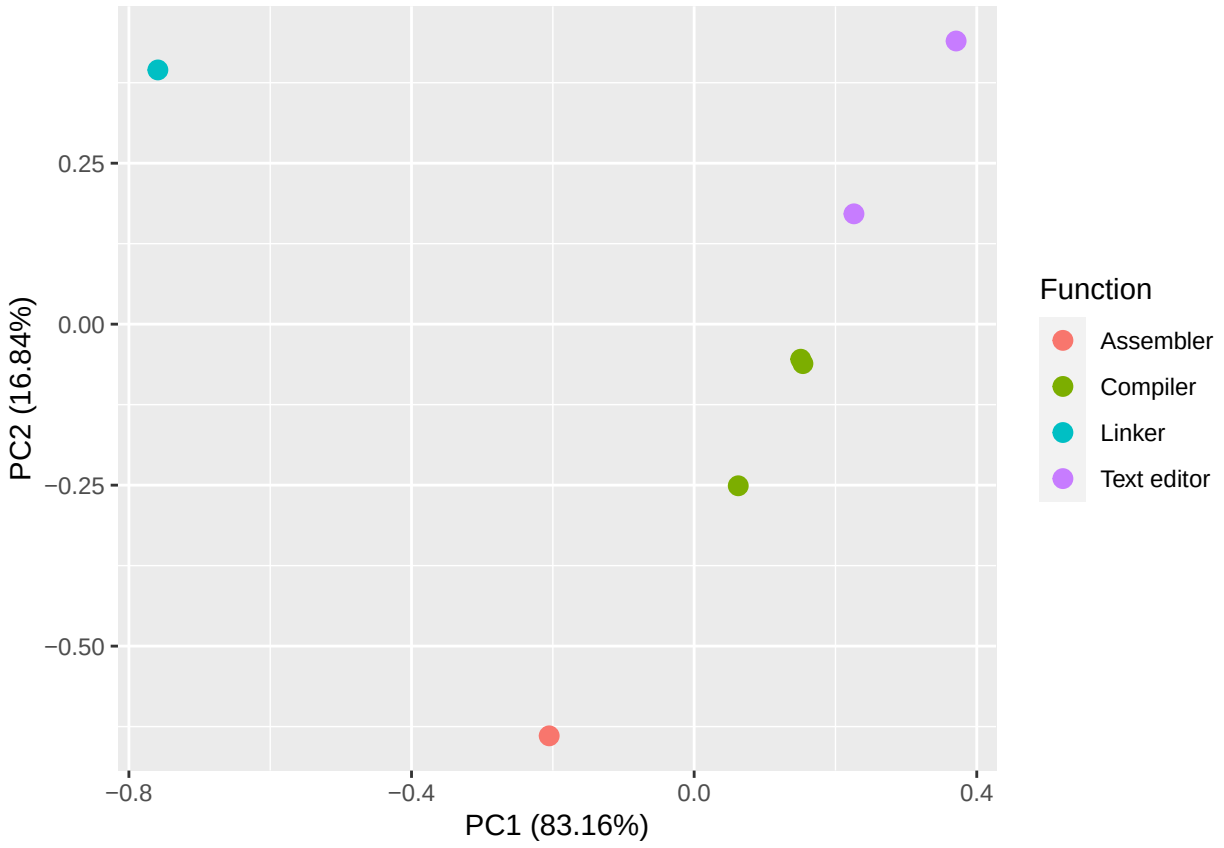
Run principal-component analysis algorithm

```
pca_result <- prcomp(dataset[3:4], scale = TRUE)
print(pca_result)
```

```
## Standard deviations (1, .., p=2):
## [1] 1.2896552 0.5803357
##
## Rotation (n x k) = (2 x 2):
##                 PC1        PC2
## CPU.Time -0.7071068 -0.7071068
## I.O.s    -0.7071068  0.7071068
```

Plot the values of principal factors

```
library(ggfortify)
autoplot(pca_result, data = dataset, colour = 'Function', size = 3)
```




### 6.2 Using a spanning-tree algorithm for cluster analysis, prepare a dendrogram for the data shown in Table 6.7. Interpret the result of your analysis.

Make program name the row name, remove "Function" collumn

```
library(tibble)
dataset <- column_to_rownames(dataset, var = "Program.Name")[,-1]
print(dataset)
```

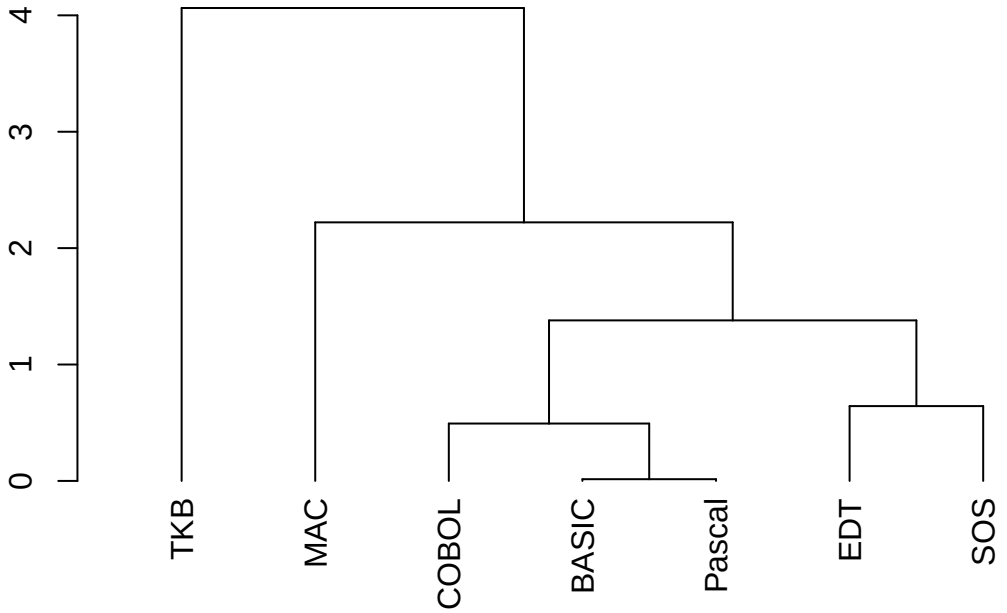
```
##         CPU.Time I.O.s
## TKB           14  2735
## MAC           13   253
## COBOL          8    27
## BASIC          6    27
## Pascal         6    12
## EDT            4    91
## SOS            1    33
```

Compute the intercluster distance matrix and do the agglomerative hierarchical clustering

```
dist <- dist(scale(dataset), method = "euclidean")
clust <- hclust(dist, method = "ward.D2")
```

Plot the dendrogram

```
dend <- as.dendrogram(clust)
plot(dend)
```


0
1
2
3
4
TKB
MAC
COBOL
BASIC
Pascal
EDT
SOS