# Cluster Analysis II

Mike Strube

October 29, 2018

## 1 Preliminaries

***In this section, the RStudio workspace and console panes are cleared of old output, variables, and other miscellaneous debris. Packages are loaded and any required data files are retrieved.***

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
    fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; important for ANOVAs.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(123)
library(knitr)
```

```
library(psych)
library(ggplot2)

##
## Attaching package:  'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha

library(MASS)
library(sciplot)
library(ggplot2)
library(vegan)

## Warning:  package 'vegan' was built under R version 3.5.1
## Loading required package:  permute
## Warning:  package 'permute' was built under R version 3.5.1
## Loading required package:  lattice
## This is vegan 2.5-2

library(smacof)
```

```
## Warning:  package 'smacof' was built under R version 3.5.1
## Loading required package:  plotrix
##
## Attaching package:  'plotrix'
## The following object is masked from 'package:psych':
##
##    rescale
##
## Attaching package:  'smacof'
## The following object is masked from 'package:base':
##
##    transform

library(ape)
library(ade4)

## Warning:  package 'ade4' was built under R version 3.5.1

library(scatterplot3d)
library(cluster)
library(factoextra)

## Warning:  package 'factoextra' was built under R version 3.5.1
## Welcome!  Related Books:  `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library(ggdendro)

## Warning:  package 'ggdendro' was built under R version 3.5.1

library(plyr)
library(fpc)
```

# 2 Old Data

***Clustering methods can be applied to the same kind of data that are examined using MDS. A proximity matrix can be used as input and the clusters identified using any of the methods.***

***Car rankings***
***Country attributes***
***President rankings***

## 2.1 Data Entry

```
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")
Cars <- read.table("cars_means_with_rover.csv", sep = ",", header = TRUE)
row.names(Cars) <- Cars$Car
Cars_Names <- as.vector(Cars[, 1])
Cars_Matrix <- as.matrix(Cars[, 2:ncol(Cars)])
# The cars data are mean ratings along multiple scales, all in the
# same metric. They can be converted to Euclidean distances.
Cars_Dist <- dist(Cars_Matrix, method = "euclidean")

Presidents <- read.table("presidents.csv", sep = ",", header = TRUE)
Presidents <- as.data.frame(Presidents)
row.names(Presidents) <- Presidents$President

# The euclidean distances are created from the ranks.
Presidents_Dist <- dist(Presidents[, 2:ncol(Presidents)], method = "euclidean",
    diag = TRUE)

# If the ratings are provided on quite different scales, then they
# should be standardized before distances are calculated. Here is
# way to do that and modify the names if files are later combined.
# Standardization is not needed here because the data are ranks
# and so all scales have identical standard deviations.
Presidents_Z <- scale(Presidents[, 2:ncol(Presidents)])
Presidents_Z <- as.data.frame(Presidents_Z)
names(Presidents_Z) <- paste(names(Presidents[-1]), "_Z", sep = "")
Presidents_Dist_Z <- dist(scale(Presidents[, 2:ncol(Presidents)]),
    method = "euclidean", diag = TRUE)
# Presidents_Dist_Z

# Raw ranks can be converted to normalized ranks as follows. This
# can be useful if there are missing data and thus different
# numbers of objects ranked across scales. This step is also
# unnecessary for for the current data because all objects were
# ranked for all scales.
Presidents_r <- Presidents[, 2:ncol(Presidents)] - 1
Presidents_NR <- matrix(NA, ncol = (length(Presidents_r[1, ])), nrow = length(Presidents_r[,
    1]))
for (j in seq(1, ncol(Presidents_r))) {
    for (i in seq(1, nrow(Presidents_r))) {
        Presidents_NR[i, j] <- Presidents_r[i, j]/(42)
```

```
    }
}

Presidents_NR <- as.data.frame(Presidents_NR)
names(Presidents_NR) <- paste(names(Presidents[-1]), "_NR", sep = "")
Presidents_Dist_NR <- dist(Presidents_NR, method = "euclidean", diag = TRUE)

Presidents_All <- cbind(Presidents, Presidents_Z, Presidents_NR)
# cor(Presidents_All[-1],,use='pairwise.complete.obs')

cor(Presidents[, 2:11])

##          PP     CL     EM     MA     IR     AS     RC    VSA
## PP  1.0000 0.9186 0.8686 0.7448 0.7440 0.7159 0.8127 0.9281
## CL  0.9186 1.0000 0.9002 0.8043 0.8706 0.7945 0.8408 0.9106
## EM  0.8686 0.9002 1.0000 0.7545 0.7981 0.8163 0.7753 0.8807
## MA  0.7448 0.8043 0.7545 1.0000 0.7419 0.7346 0.7085 0.8217
## IR  0.7440 0.8706 0.7981 0.7419 1.0000 0.7596 0.7066 0.7735
## AS  0.7159 0.7945 0.8163 0.7346 0.7596 1.0000 0.8025 0.7915
## RC  0.8127 0.8408 0.7753 0.7085 0.7066 0.8025 1.0000 0.8043
## VSA 0.9281 0.9106 0.8807 0.8217 0.7735 0.7915 0.8043 1.0000
## PEJ 0.5512 0.5817 0.6616 0.6392 0.5159 0.5667 0.5435 0.6222
## PCT 0.9230 0.9641 0.9166 0.8664 0.8443 0.8303 0.8774 0.9449
##         PEJ    PCT
## PP  0.5512 0.9230
## CL  0.5817 0.9641
## EM  0.6616 0.9166
## MA  0.6392 0.8664
## IR  0.5159 0.8443
## AS  0.5667 0.8303
## RC  0.5435 0.8774
## VSA 0.6222 0.9449
## PEJ 1.0000 0.6237
## PCT 0.6237 1.0000

Trump <- read.table("trump.csv", sep = ",", header = TRUE)
Trump <- as.data.frame(Trump)
row.names(Trump) <- Trump$Country
Trump_Dist <- dist(Trump[, 2:ncol(Trump)], method = "euclidean", diag = TRUE)
```

## 2.2 Ward's Method

*Ward's method usually provides a good solution, so we'll give that a try. The dissimilarity between two clusters (A and B) is the loss of information from joining the clusters, measured by the increase in error sum of squares.*

*The sum of squares for a cluster is the sum of squared deviations of each case from the centroid for the cluster. The error sum of squares is the total of these for all clusters. The two clusters among all possible combinations that have the minimum increase in error sum of squares are joined.*

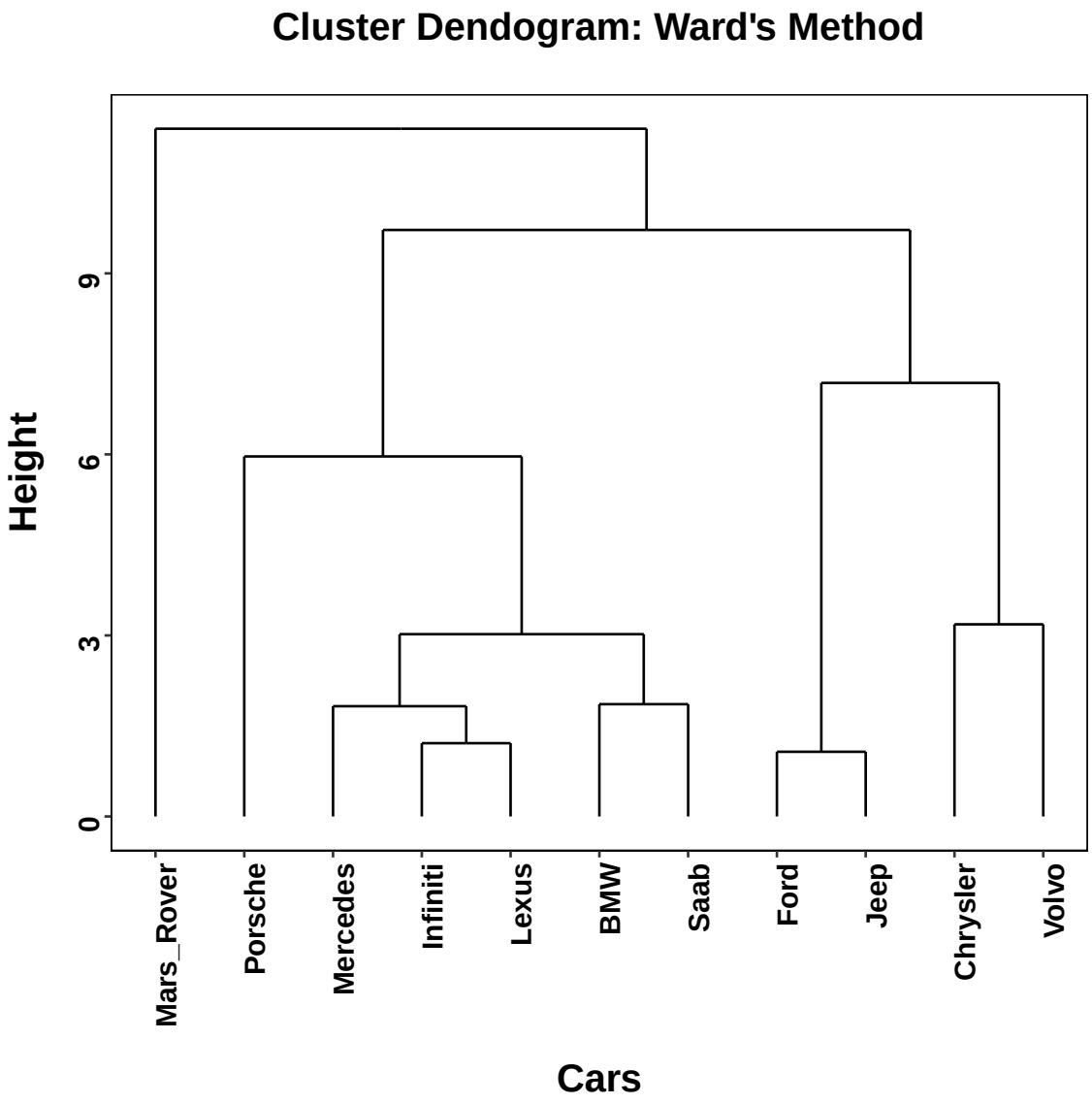
*Two versions are available. The Ward D method will produce the traditional Ward solution, but only if squared Euclidean distances are used. the Ward D2 method will produce the traditional Ward solution starting from Euclidean distances.*

*Note that Ward's method traditionally is described as requiring squared Euclidean distances, but this requirement is often relaxed, apparently with little effect. We will use it here, where it is is most strongly justified for the country rating data, somewhat less so for the president data, and least so for the car data. An active area of work is the impact on clustering of different metric with different clustering methods.*

### 2.2.1 Car Data

```
Cars_Wards <- hclust(Cars_Dist, method = "ward.D2")
```

```
ggdendrogram(Cars_Wards, theme_dendro = FALSE, size = 4) + xlab("Cars") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16, angle = 90), axis.line.x = element_blank(),
    axis.line.y = element_blank(), plot.title = element_text(size = 16,
        face = "bold", margin = margin(0, 0, 20, 0), hjust = 0.5),
    panel.background = element_rect(fill = "white", linetype = 1,
        color = "black"), panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "white"), plot.margin = unit(c(1,
        1, 1, 1), "cm"), legend.position = "bottom", legend.title = element_blank()) +
    ggtitle("Cluster Dendogram: Ward's Method")
```

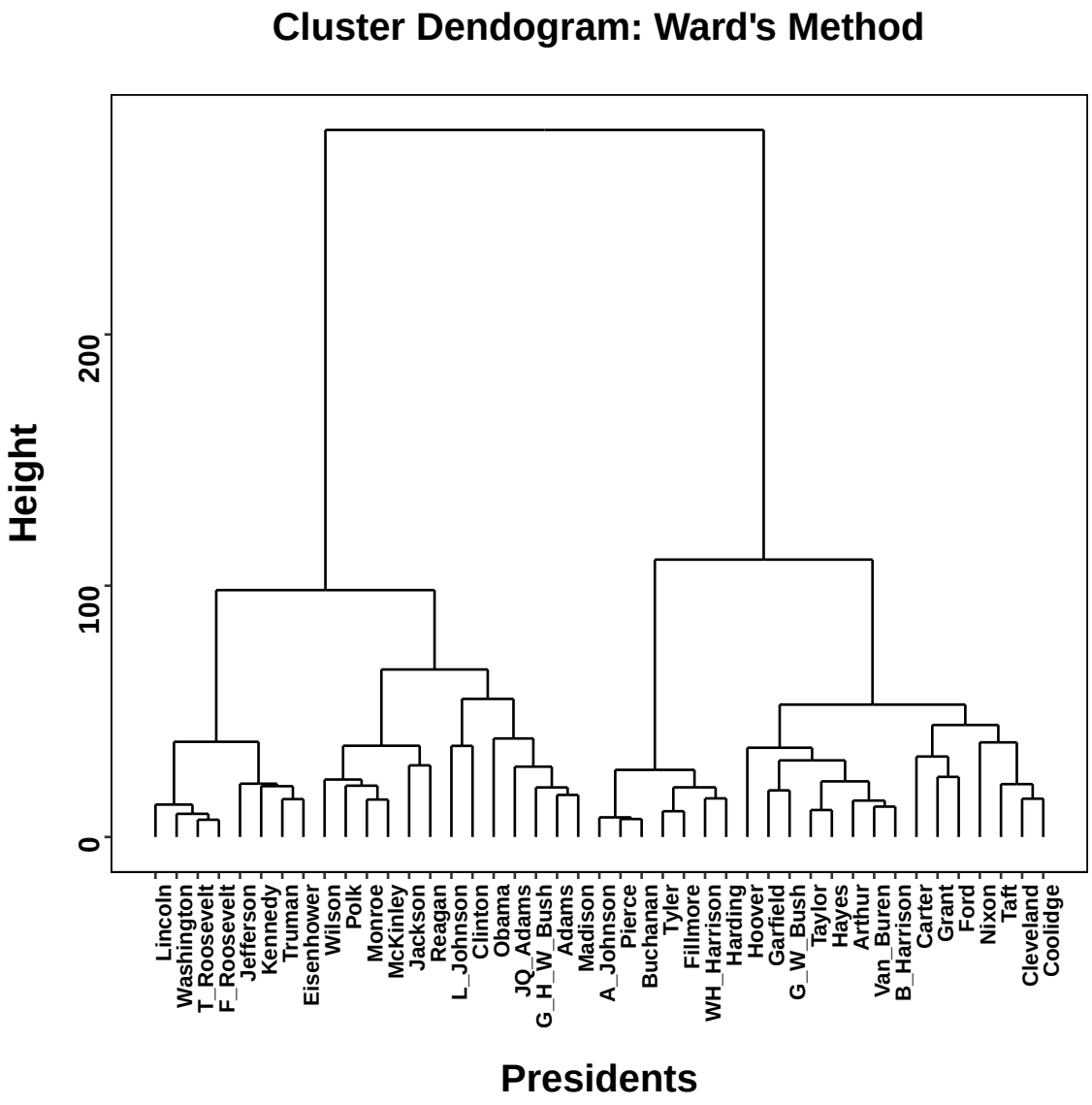



### 2.2.2 President Data

```
Presidents_Wards <- hclust(Presidents_Dist, method = "ward.D2")
```

```
ggdendrogram(Presidents_Wards, theme_dendro = FALSE, size = 4) + xlab("Presidents") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 9, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16, angle = 90), axis.line.x = element_blank(),
    axis.line.y = element_blank(), plot.title = element_text(size = 16,
```

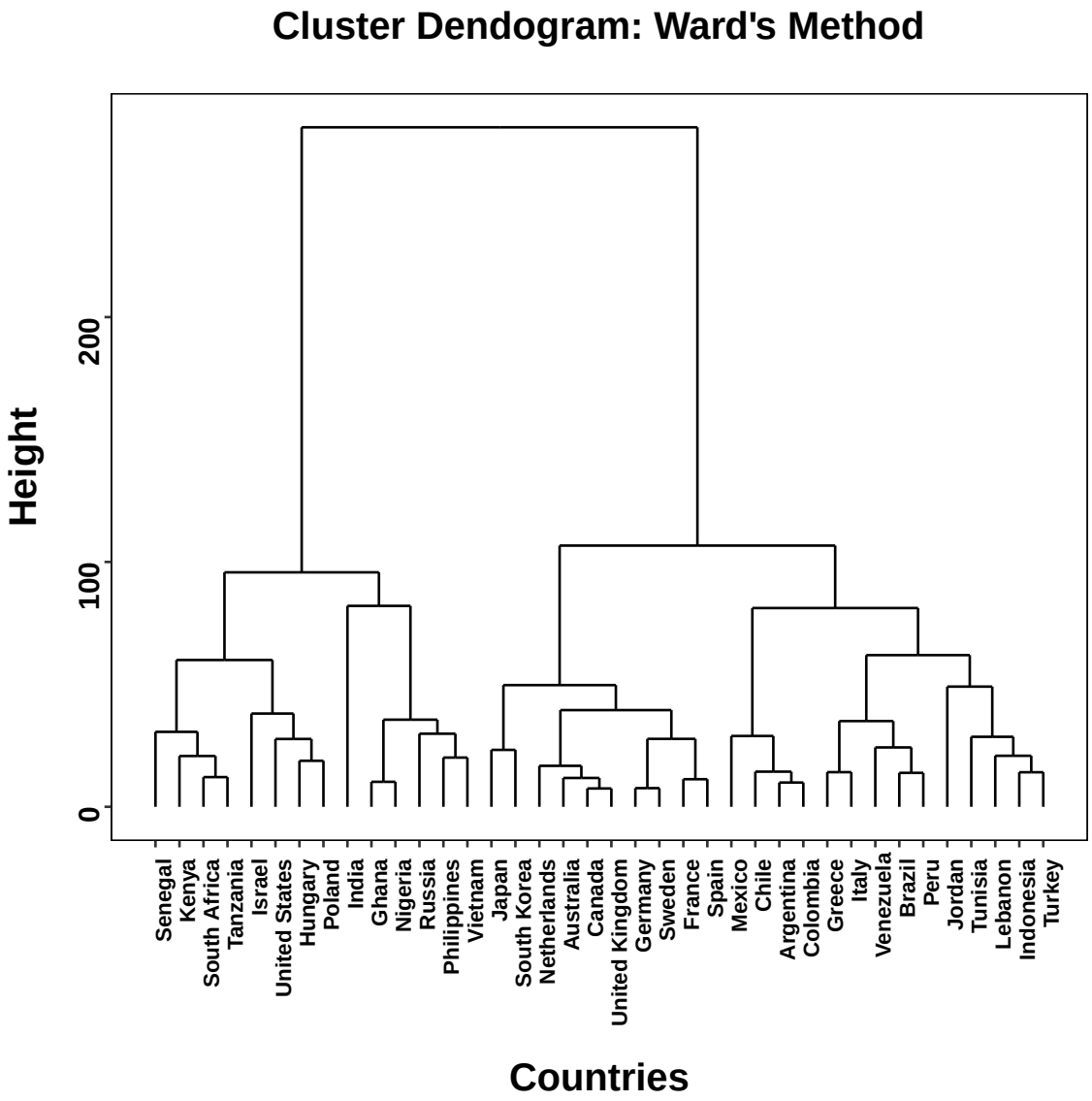
```
    face = "bold", margin = margin(0, 0, 20, 0), hjust = 0.5),
panel.background = element_rect(fill = "white", linetype = 1,
    color = "black"), panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
plot.background = element_rect(fill = "white"), plot.margin = unit(c(1,
    1, 1, 1), "cm"), legend.position = "bottom", legend.title = element_blank()) +
ggtitle("Cluster Dendogram: Ward's Method")
```




### 2.2.3 Trump Data

```
Trump_Wards <- hclust(Trump_Dist, method = "ward.D2")
```

```
ggdendrogram(Trump_Wards, theme_dendro = FALSE, size = 4) + xlab("Countries") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 9, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16, angle = 90), axis.line.x = element_blank(),
    axis.line.y = element_blank(), plot.title = element_text(size = 16,
        face = "bold", margin = margin(0, 0, 20, 0), hjust = 0.5),
    panel.background = element_rect(fill = "white", linetype = 1,
        color = "black"), panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "white"), plot.margin = unit(c(1,
        1, 1, 1), "cm"), legend.position = "bottom", legend.title = element_blank()) +
    ggtitle("Cluster Dendogram: Ward's Method")
```



# 3 K-Means Clustering

*The partitioning procedure known as K-Means clustering attempts to form clusters that have the smallest possible within-cluster variances. The partitioning approach to finding clusters begins with specification of the number of clusters desired (K) and "seed" values for the initial cluster centroids. Then, cases are assigned to clusters so that the sum of the squared distances from cases to cluster centroids are minimized. Cases are reassigned until no further reduction in the sum of squared deviations is found. The K-Means clustering procedure is similar to Ward's method, but is not a hierarchical approach. In Ward's method, when cases are joined in a cluster they cannot later separate and join different clusters. Reassignment is possible in K-Means clustering.*

## 3.1 Iris Data

*We'll begin by analyzing the iris data. The kmeans( ) function in the basic stats available when R starts up is a good option for most problems. It requires the raw data matrix, with the objects to be clustered on the rows. If the variables used to measure the objects are not in the same scale, then they should be standardized first. Because of the nature of the method, the data are assumed to be at least interval level.*

### 3.1.1 Data

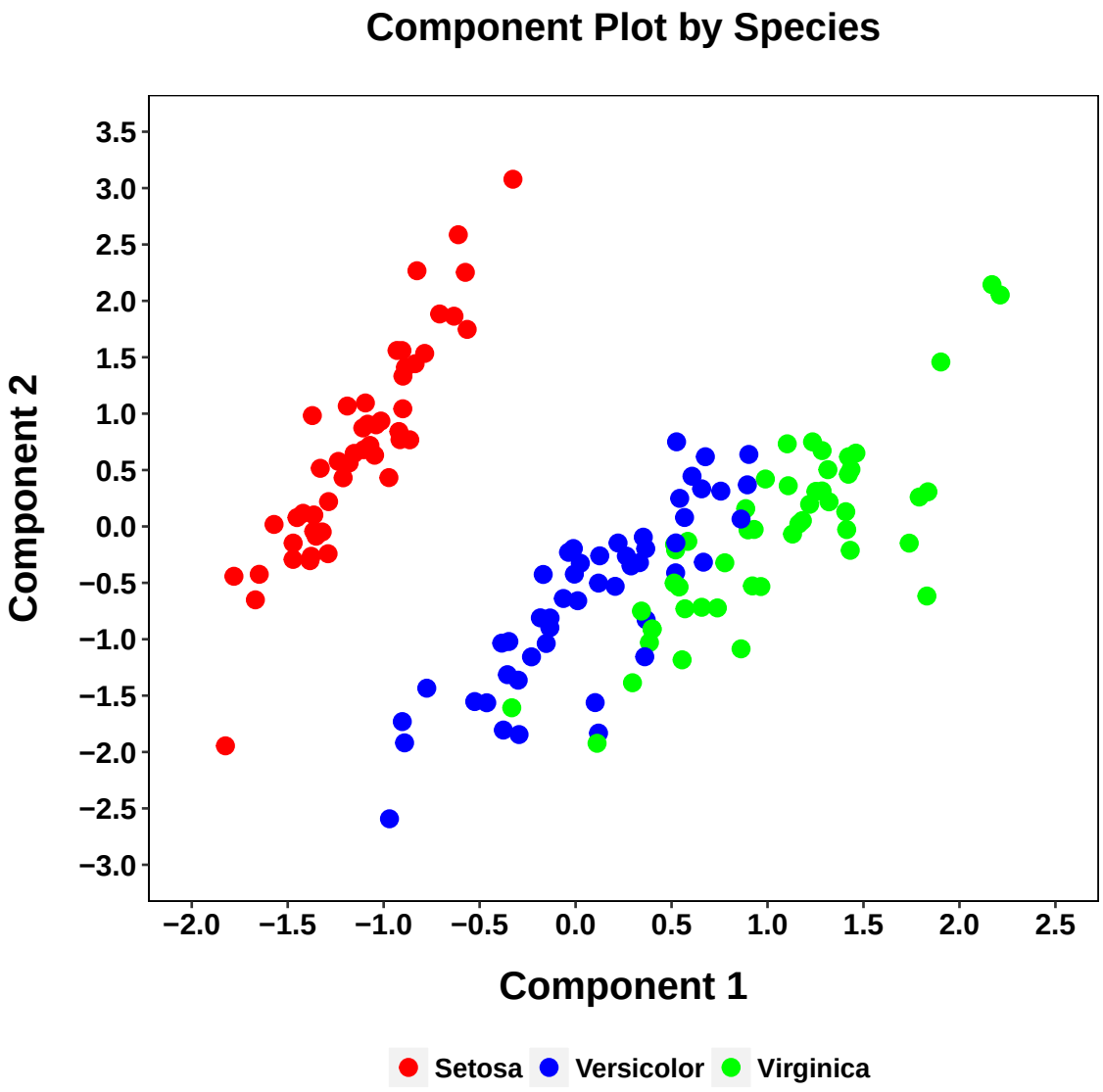
```
# Get the use data from the working directory.
Iris <- read.table("iris.csv", sep = ",", header = TRUE)
Iris <- as.data.frame(Iris)
Iris$Species[Iris$Species == "1"] <- "Setosa"
Iris$Species[Iris$Species == "2"] <- "Versicolor"
Iris$Species[Iris$Species == "3"] <- "Virginica"
```

### 3.1.2 Dimensional Plot

```
# Use PCA to show potential clustering along two dimensions.
PCA <- principal(Iris[, 1:4], nfactors = 2, rotate = "varimax", scores = TRUE)
PCA

## Principal Components Analysis
## Call: principal(r = Iris[, 1:4], nfactors = 2, rotate = "varimax",
##     scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                RC1   RC2   h2     u2 com
## Sepal_Length  0.96  0.05 0.92 0.0774 1.0
## Sepal_Width  -0.14  0.98 0.99 0.0091 1.0
## Petal_Length  0.94 -0.30 0.98 0.0163 1.2
## Petal_Width   0.93 -0.26 0.94 0.0647 1.2
##
##                        RC1  RC2
## SS loadings           2.70 1.13
## Proportion Var        0.68 0.28
## Cumulative Var        0.68 0.96
## Proportion Explained  0.71 0.29
## Cumulative Proportion 0.71 1.00
##
```

```
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.03
##  with the empirical chi square  1.72  with prob <  NA
##
## Fit based upon off diagonal values = 1

Iris <- cbind(Iris, PCA$scores)
```

```
ggplot(Iris, aes(x = RC1, y = RC2, color = factor(Species))) + geom_point(shape = 19,
    size = 3) + scale_color_manual(values = c("red", "blue", "green")) +
    scale_y_continuous(breaks = c(seq(-3, 3.5, 0.5))) + scale_x_continuous(breaks = c(seq(-2,
    2.5, 0.5))) + coord_cartesian(xlim = c(-2, 2.5), ylim = c(-3,
    3.5)) + xlab("Component 1") + ylab("Component 2") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Component Plot by Species")
```




### 3.1.3 Basic Function

```
# The minimum specification is the data source and number of
# clusters desired.
Iris_K <- kmeans(Iris[, 1:4], centers = 3, iter.max = 1000, nstart = 10)
Iris_K$cluster

##   [1] 3 2 1 2 1 3 2 1 1 3 1 1 2 1 2 1 2 3 1 2 2 1 2 2 1 3 2 1 1 1
##  [31] 3 1 1 1 2 3 3 1 2 3 2 3 1 3 2 1 3 1 1 2 3 3 1 3 3 3 2 2 3 3
##  [61] 3 1 1 3 3 1 1 3 3 1 1 3 3 2 2 1 1 2 3 3 2 2 1 2 1 1 1 3 3 2
##  [91] 1 3 1 1 1 3 3 1 1 1 3 3 2 1 2 1 3 3 1 1 2 2 3 1 1 3 1 1 1 1
## [121] 1 1 2 2 3 3 2 2 1 1 2 2 2 1 3 3 3 1 3 3 1 1 1 3 3 3 1 2 2 3

Iris_K$centers
```

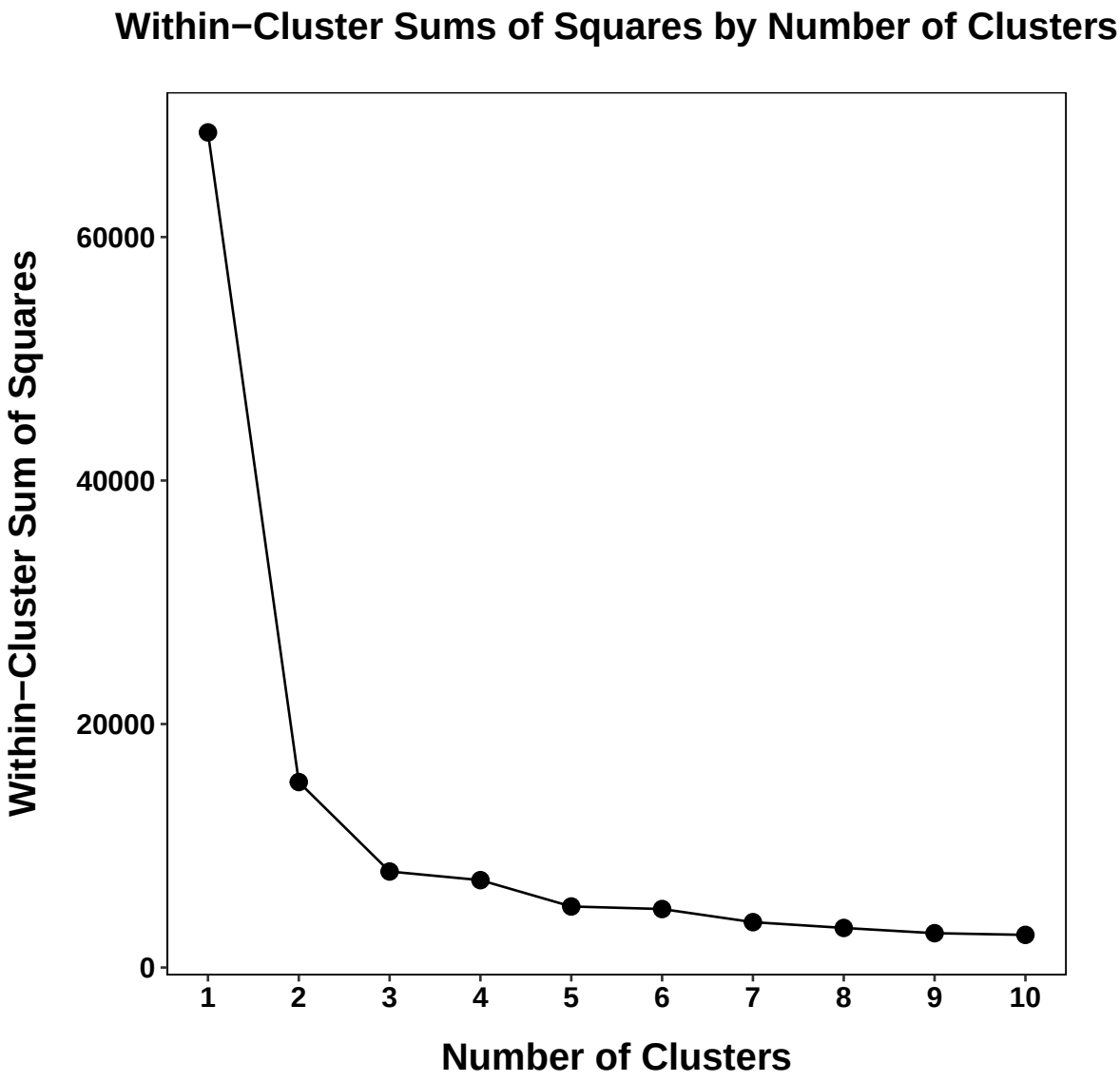
```
##   Sepal_Length Sepal_Width Petal_Length Petal_Width
## 1        59.02       27.48        43.94       14.34
## 2        68.50       30.74        57.42       20.71
## 3        50.06       34.28        14.62        2.46

Iris_K$totss

## [1] 68137

Iris_K$tot.withinss

## [1] 7885

Iris_K$betweenss

## [1] 60252

Iris_K$withinss

## [1] 3982 2388 1515

Iris_K$size

## [1] 62 38 50

Iris_Class <- as.data.frame(cbind(Iris_K$cluster, Iris$Species))
names(Iris_Class) <- c("Cluster", "Species")
table(Iris_Class$Species, Iris_Class$Cluster)

##
##                1  2  3
##   Setosa       0  0 50
##   Versicolor  48  2  0
##   Virginica   14 36  0

Iris_New <- as.data.frame(Iris_K$cluster)
names(Iris_New) <- c("Cluster")
Iris_New <- as.data.frame(cbind(Iris, Iris_New))
```

### 3.1.4 Plot of Within-Cluster Sums of Squares

***The method attempts to minimize the within-cluster sums of squares. This can be used to help identify the optimal number of clusters. For different numbers of clusters, a point may be found, after which little improvement in the solution occurs. Similar to a scree plot, the point at which the plot of within-cluster sums of squares reaches a discernible floor can be used as the optimal number of clusters.***

```
wssplot <- function(data, nc = 15, seed = 1234) {
    wss <- (nrow(data - 1)) * sum(apply(data, 2, var))
    for (i in 2:nc) {
        set.seed(seed)
        wss[i] <- sum(kmeans(data, centers = i)$withinss)
    }
    plot_data <- cbind(wss, seq(1, nc, 1))
    plot_data <- as.data.frame(plot_data)
```

```
    names(plot_data) <- c("wss", "nc")
    ggplot(plot_data, aes(x = nc, y = wss)) + geom_point(shape = 19,
        size = 3) + geom_line() + scale_x_continuous(breaks = c(seq(1,
        nc, 1))) + xlab("Number of Clusters") + ylab("Within-Cluster Sum of Squares") +
        theme(text = element_text(size = 14, family = "sans", color = "black",
            face = "bold"), axis.text.y = element_text(colour = "black",
            size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
            size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
            0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
            15, 0, 0), size = 16), axis.line.x = element_blank(),
            axis.line.y = element_blank(), plot.title = element_text(size = 16,
                face = "bold", margin = margin(0, 0, 20, 0), hjust = 0.5),
            panel.background = element_rect(fill = "white", linetype = 1,
                color = "black"), panel.grid.major = element_blank(),
            panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
            plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
            legend.title = element_blank()) + ggtitle("Within-Cluster Sums of Squares by Number of Clust
}
wssplot(Iris[, 1:4], nc = 10)
```

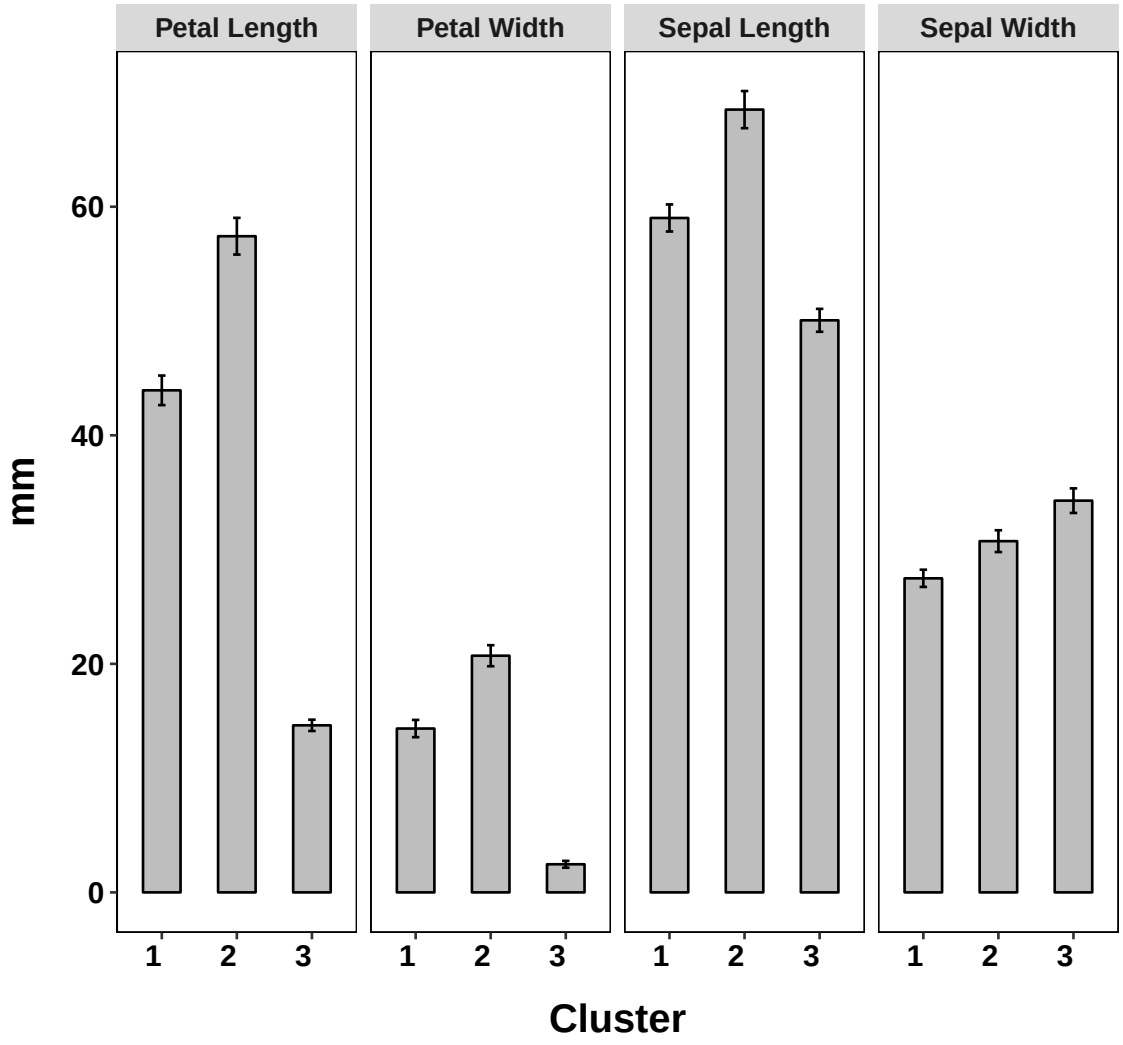



### 3.1.5 Plot of Cluster Means on Original Variables

*Once the clusters are identified, the means on the original variables can be plotted.*

```
SE_SL <- ddply(Iris_New, ~Cluster, summarise, se = se(Sepal_Length))
Means_SL <- ddply(Iris_New, ~Cluster, summarise, mean = mean(Sepal_Length))
N_SL <- table(Iris_New$Cluster)
SL <- cbind(Means_SL, SE_SL$se, N_SL)
SL <- as.data.frame(SL)
SL <- SL[-4]
names(SL) <- c("Cluster", "Mean", "SE", "N")

SE_SW <- ddply(Iris_New, ~Cluster, summarise, se = se(Sepal_Width))
Means_SW <- ddply(Iris_New, ~Cluster, summarise, mean = mean(Sepal_Width))
```

```
N_SW <- table(Iris_New$Cluster)
SW <- cbind(Means_SW, SE_SW$se, N_SW)
SW <- as.data.frame(SW)
SW <- SW[-4]
names(SW) <- c("Cluster", "Mean", "SE", "N")

SE_PL <- ddply(Iris_New, ~Cluster, summarise, se = se(Petal_Length))
Means_PL <- ddply(Iris_New, ~Cluster, summarise, mean = mean(Petal_Length))
N_PL <- table(Iris_New$Cluster)
PL <- cbind(Means_PL, SE_PL$se, N_PL)
PL <- as.data.frame(PL)
PL <- PL[-4]
names(PL) <- c("Cluster", "Mean", "SE", "N")

SE_PW <- ddply(Iris_New, ~Cluster, summarise, se = se(Petal_Width))
Means_PW <- ddply(Iris_New, ~Cluster, summarise, mean = mean(Petal_Width))
N_PW <- table(Iris_New$Cluster)
PW <- cbind(Means_PW, SE_PW$se, N_PW)
PW <- as.data.frame(PW)
PW <- PW[-4]
names(PW) <- c("Cluster", "Mean", "SE", "N")

plot_data <- rbind(SL, SW, PL, PW)
plot_data$Feature <- c(rep("Sepal Length", 3), rep("Sepal Width",
    3), rep("Petal Length", 3), rep("Petal Width", 3))
plot_data$Feature <- factor(plot_data$Feature)
plot_data$CI_L <- plot_data$Mean - plot_data$SE * qt(0.975, df = plot_data$N -
    1)
plot_data$CI_U <- plot_data$Mean + plot_data$SE * qt(0.975, df = plot_data$N -
    1)
plot_data$Cluster <- factor(plot_data$Cluster)

p <- ggplot(plot_data, aes(x = Cluster, y = Mean)) + geom_bar(position = position_dodge(),
    stat = "identity", color = "black", width = 0.5, fill = "grey") +
    geom_errorbar(aes(ymin = CI_L, ymax = CI_U), width = 0.1, position = position_dodge(0.5)) +
    xlab("Cluster") + ylab("mm") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 1), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Sepal and Petal Size By Cluster")
p + facet_grid(~Feature)
```




```
summary(aov(Iris_New$Sepal_Length ~ as.factor(Iris_New$Cluster)))

##                              Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Iris_New$Cluster)   2   7378    3689     191 <2e-16
## Residuals                   147   2839      19

summary(aov(Iris_New$Sepal_Width ~ as.factor(Iris_New$Cluster)))

##                              Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Iris_New$Cluster)   2   1280     640    60.6 <2e-16
## Residuals                   147   1551      11

summary(aov(Iris_New$Petal_Length ~ as.factor(Iris_New$Cluster)))

##                              Df Sum Sq Mean Sq F value Pr(>F)
```

```
## as.factor(Iris_New$Cluster)   2  43822   21911    1234 <2e-16
## Residuals                   147   2611      18
```

```
summary(aov(Iris_New$Petal_Width ~ as.factor(Iris_New$Cluster)))
```

```
##                              Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Iris_New$Cluster)   2   7773    3886     646 <2e-16
## Residuals                   147    884       6
```

### 3.1.6 Alternative Method

***A more general approach is to cluster around "medoids." A medoid is the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. In other words, it is the most centrally located object in the cluster. The pam( ) function in the cluster library can perform this calculation. In the following, a dissimilarity matrix based on Euclidean distance is created on the fly and used as input. The cluster information that is provided includes (from the documentation) the cardinality of the cluster (number of observations), the maximal and average dissimilarity between the observations in the cluster and the cluster's medoid, the diameter of the cluster (maximal dissimilarity between two observations of the cluster), and the separation of the cluster (minimal dissimilarity between an observation of the cluster and an observation of another cluster).***

```
Iris_P <- pam(dist(Iris[, 1:4], method = "euclidean"), k = 3, diss = TRUE)
attributes(Iris_P)
```

```
## $names
## [1] "medoids"    "id.med"     "clustering" "objective"
## [5] "isolation"  "clusinfo"   "silinfo"    "diss"
## [9] "call"
##
## $class
## [1] "pam"       "partition"
```

```
Iris_P$clustering
```

```
##   [1] 1 2 3 2 3 1 2 3 3 1 3 3 2 3 2 3 2 1 3 2 2 3 2 2 3 1 2 3 3 3
##  [31] 1 3 3 3 2 1 1 3 2 1 2 1 3 1 2 3 1 3 3 2 1 1 3 1 1 1 2 2 1 1
##  [61] 1 3 3 1 1 3 3 1 1 3 3 1 1 2 2 3 3 2 1 1 2 2 3 2 3 3 3 1 1 2
##  [91] 3 1 3 3 3 1 1 3 3 3 1 1 2 3 2 3 1 1 3 3 2 2 1 3 3 1 3 3 3 3
## [121] 3 3 2 2 1 1 2 2 3 3 2 2 2 3 1 1 1 3 1 1 3 3 3 1 1 1 3 2 2 1
```

```
Iris_P$clusinfo
```

```
##      size max_diss av_diss diameter separation
## [1,]   50    12.37   4.846    24.29     16.401
## [2,]   38    17.23   7.260    24.19      2.646
## [3,]   62    18.38   7.470    26.78      2.646
```

```
Iris_Class <- as.data.frame(cbind(Iris_P$clustering, Iris$Species))
names(Iris_Class) <- c("Cluster", "Species")
table(Iris_Class$Species, Iris_Class$Cluster)
```

```
##
##              1  2  3
```

```
##   Setosa     50  0  0
##   Versicolor  0  2 48
##   Virginica   0 36 14

# The following table compares the clustering done by pam( ) and
# that done by kmeans( ).
table(Iris_K$cluster, Iris_P$clustering)

##
##      1  2  3
##   1  0  0 62
##   2  0 38  0
##   3 50  0  0
```

## 3.2 Right Wing Data

***A sample of 150 people were surveyed concerning their opinions about four controversial issues. On a 10-point rating scale, ranging from Completely Disapprove (1) to Completely Approve (10), the respondents rated their opinions of:***

***Gun Control***
***Prayer in the Schools***
***Death Penalty***
***Same Sex Marriage***

***The sample also reported their annual income and their number of years of education. The role of socioeconomic status in shaping opinions on controversial topics was the goal of the study.***

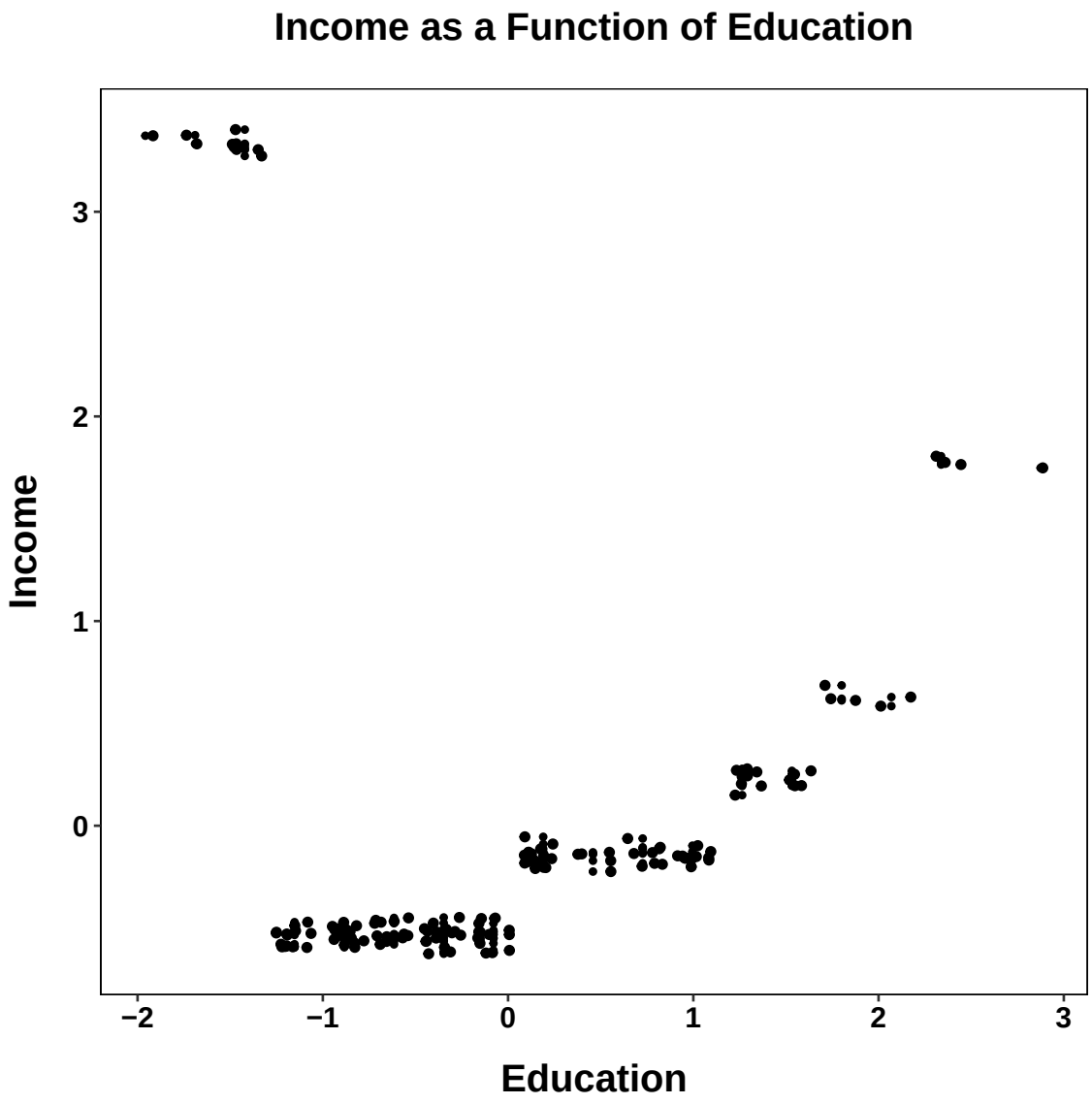
### 3.2.1 Data

```
# Get the use data from the working directory.
RW_Data <- read.table("right_wing_data.csv", sep = ",", header = TRUE)
RW <- as.data.frame(scale(RW_Data))
```

### 3.2.2 Relationship Between Income and Education

***The relationship between income and education is unusual and suggests subgroups may exist in the data. Cluster analysis can help identify them.***

```
plot_data <- RW

ggplot(plot_data, aes(x = educate, y = income)) + geom_point(shape = 19,
    size = 1) + geom_jitter() + xlab("Education") + ylab("Income") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Income as a Function of Education")
```




### 3.2.3 Ward's Method

***A hierarchical cluster analysis using Ward's method suggests from 2 to 6 clusters in the sample.***

```
RW_HC <- hclust(dist(RW[, 1:6], method = "euclidean"), method = "ward.D2")

RW_Clusters_H <- as.data.frame(cutree(RW_HC, k = 6))
names(RW_Clusters_H) <- c("Cluster_H")
RW_New_HC <- as.data.frame(cbind(RW, RW_Clusters_H))
aggregate(RW_New_HC, by = list(RW_New_HC$Cluster), mean)

##   Group.1 educate   income     gun  prayer   death samesex
## 1       1 -1.5286  3.33375 -0.7145  0.7445  1.0163 -0.8991
```

```
## 2       2 -0.7215 -0.55074 -1.0690 -0.2478 -0.3747 -0.8877
## 3       3 -0.5486 -0.52267 -0.2513  0.7326  0.9530 -0.3638
## 4       4  0.3998 -0.17050  0.1827 -0.8026 -0.8877  0.1952
## 5       5  1.0489  0.06428  1.0839 -0.4503 -0.5356  1.2516
## 6       6  2.4718  1.77354  1.2292  1.9747  1.4809 -0.4295
##   Cluster_H
## 1         1
## 2         2
## 3         3
## 4         4
## 5         5
## 6         6
```

```
summary(aov(RW_New_HC$educate ~ as.factor(RW_New_HC$Cluster_H)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_HC$Cluster_H)   5  124.7   24.94     148 <2e-16
## Residuals                      144   24.3    0.17
```

```
summary(aov(RW_New_HC$income ~ as.factor(RW_New_HC$Cluster_H)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_HC$Cluster_H)   5  145.5   29.10    1194 <2e-16
## Residuals                      144    3.5    0.02
```

```
summary(aov(RW_New_HC$gun ~ as.factor(RW_New_HC$Cluster_H)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_HC$Cluster_H)   5   99.1   19.83    57.3 <2e-16
## Residuals                      144   49.9    0.35
```

```
summary(aov(RW_New_HC$prayer ~ as.factor(RW_New_HC$Cluster_H)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_HC$Cluster_H)   5   67.6   13.51    23.9 <2e-16
## Residuals                      144   81.4    0.57
```

```
summary(aov(RW_New_HC$death ~ as.factor(RW_New_HC$Cluster_H)))
```

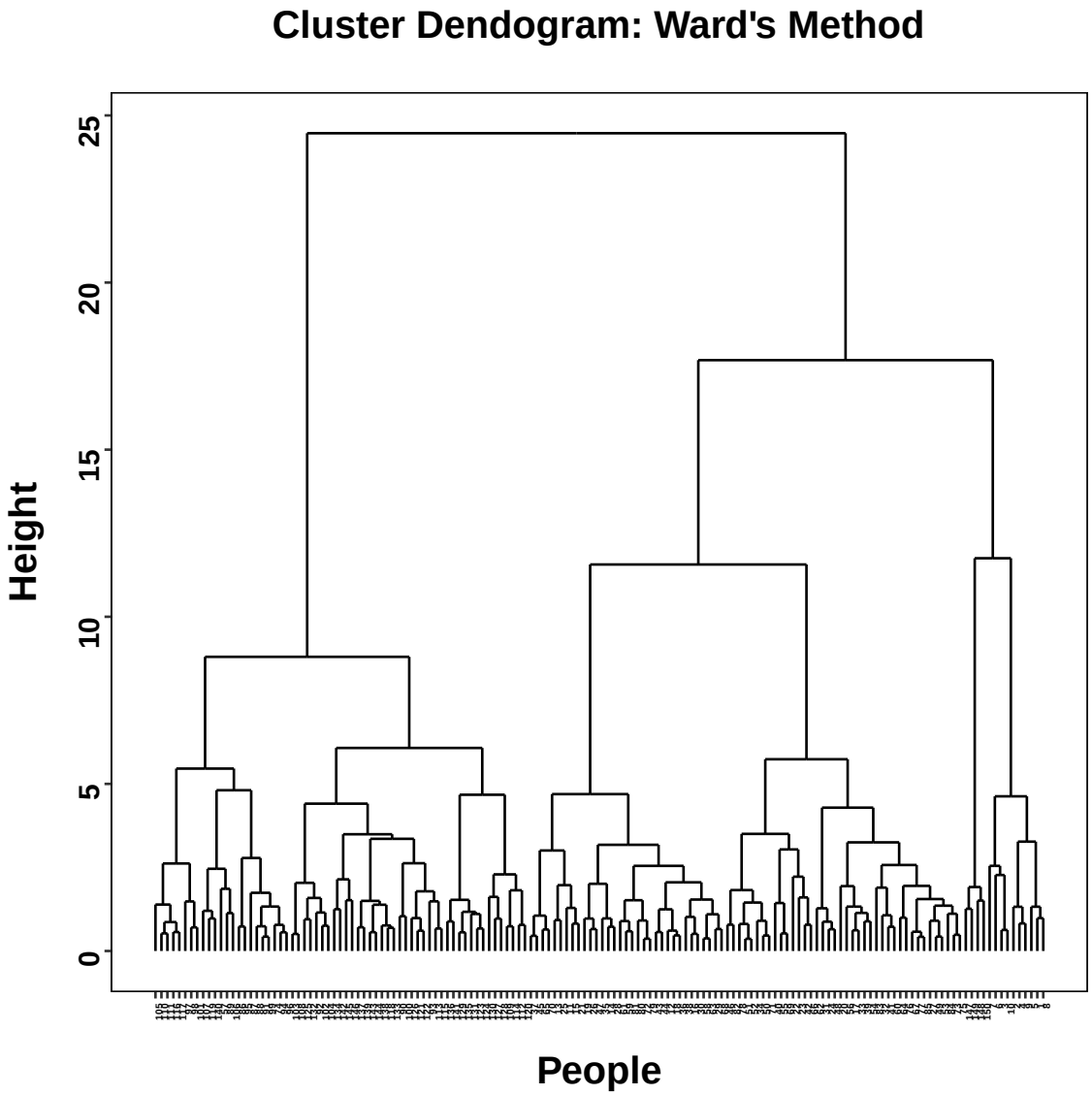
```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_HC$Cluster_H)   5   89.7   17.93    43.5 <2e-16
## Residuals                      144   59.3    0.41
```

```
summary(aov(RW_New_HC$samesex ~ as.factor(RW_New_HC$Cluster_H)))
```

```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_HC$Cluster_H)   5  103.7   20.73    65.8 <2e-16
## Residuals                      144   45.3    0.31
```

```
ggdendrogram(RW_HC, theme_dendro = FALSE, size = 4) + xlab("People") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 4, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
```

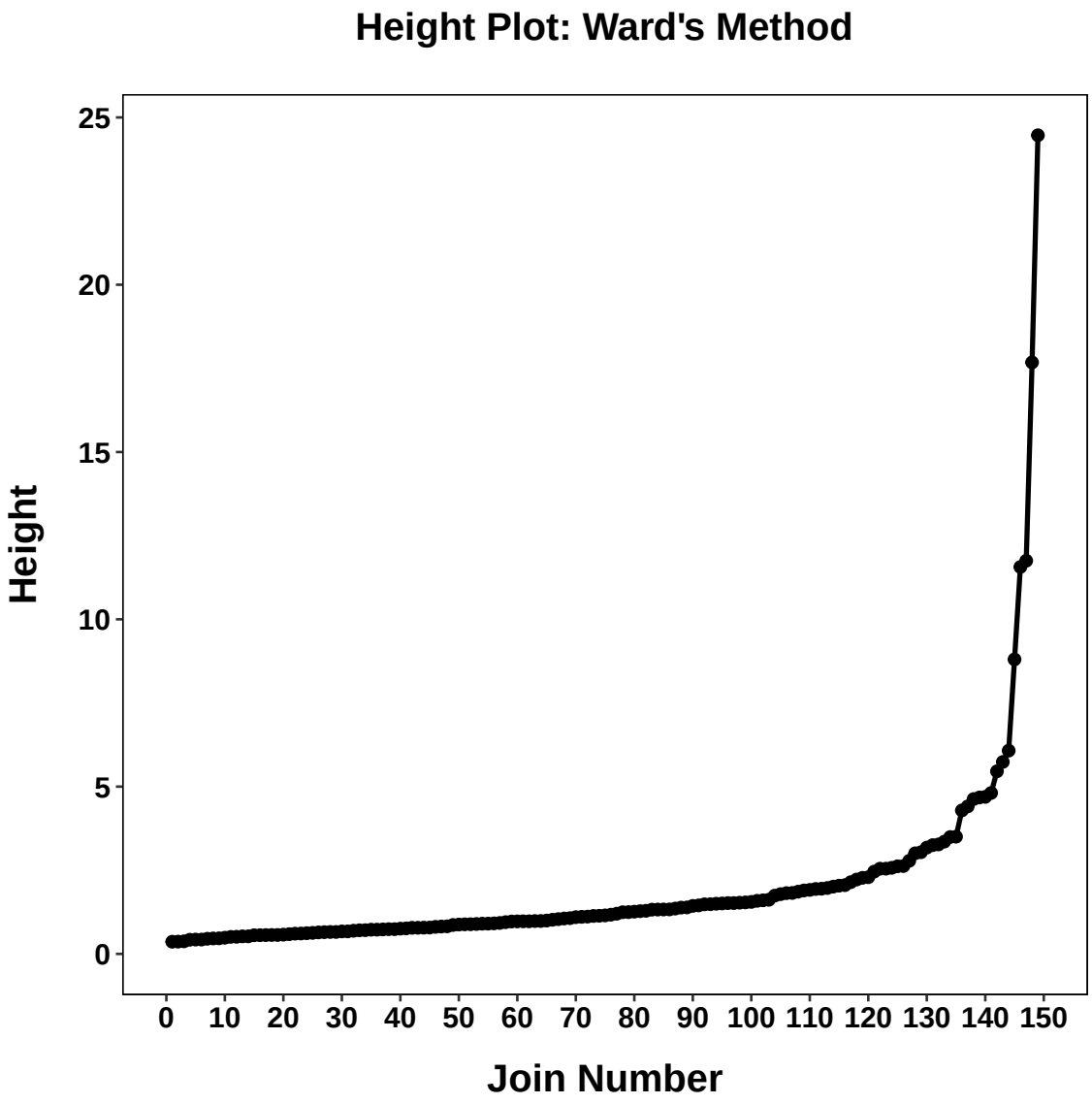
```
    15, 0, 0), size = 16, angle = 90), axis.line.x = element_blank(),
    axis.line.y = element_blank(), plot.title = element_text(size = 16,
        face = "bold", margin = margin(0, 0, 20, 0), hjust = 0.5),
    panel.background = element_rect(fill = "white", linetype = 1,
        color = "black"), panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "white"), plot.margin = unit(c(1,
        1, 1, 1), "cm"), legend.position = "bottom", legend.title = element_blank()) +
    ggtitle("Cluster Dendogram: Ward's Method")
```




#### 3.2.4 Height Plot

*A plot of the height at which joining occurs can sometimes provide insight into the number of clusters that best simplifies the data; 6 cluster appears to be best.*

```
H <- matrix(RW_HC$height, nrow = 149)
plot_data <- cbind(seq(1, 149, 1), H)
plot_data <- as.data.frame(plot_data)
names(plot_data) <- c("Join", "Height")

ggplot(plot_data, aes(x = Join, y = Height)) + geom_point(shape = 19,
    size = 2, color = "black", na.rm = TRUE) + geom_line(size = 1) +
    scale_x_continuous(breaks = c(seq(0, 150, 10))) + coord_cartesian(xlim = c(0,
    150), ylim = c(0, max(plot_data$Height))) + xlab("Join Number") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Height Plot: Ward's Method")
```



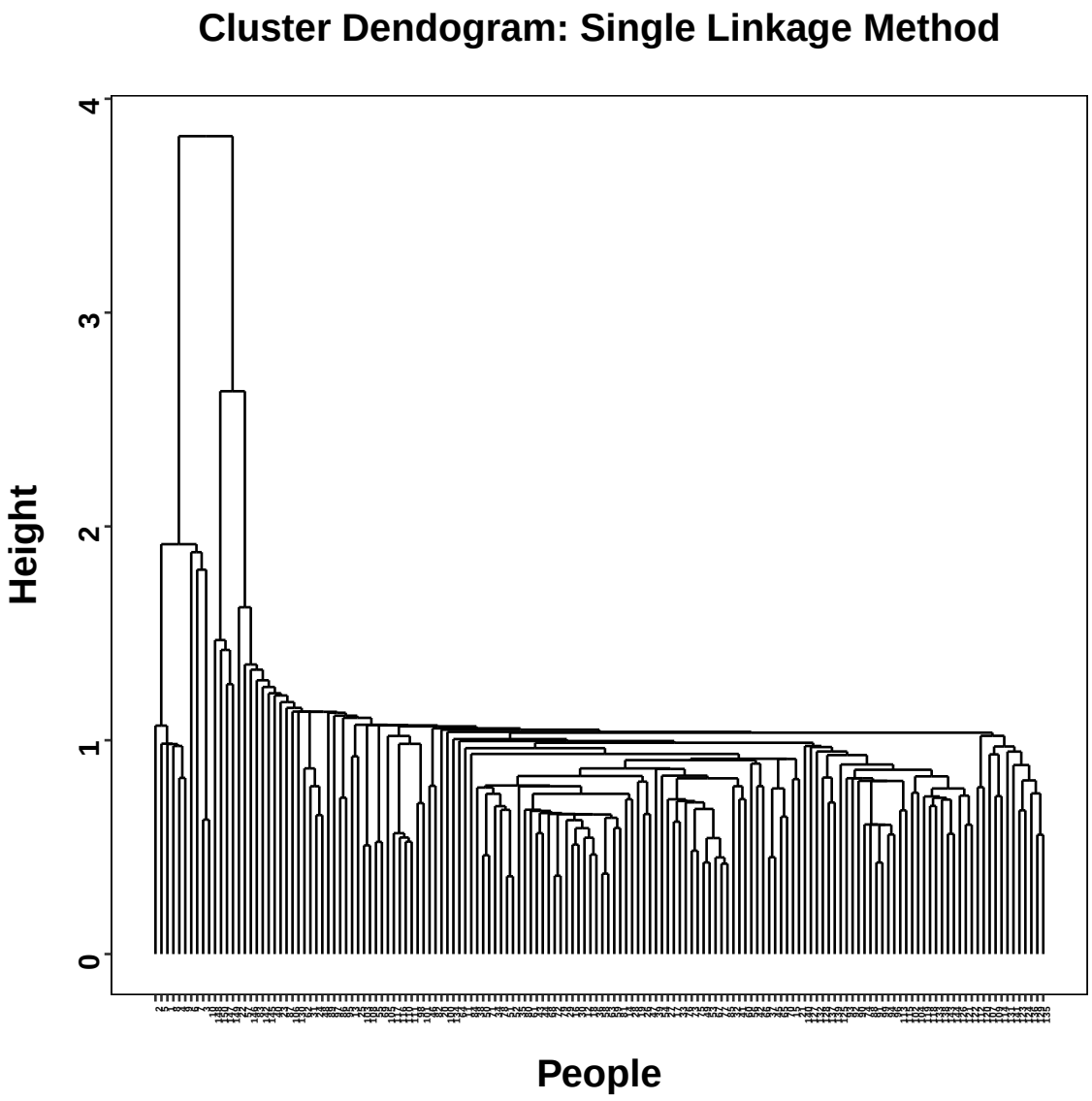

### 3.2.5 Single Linkage Method

*The single linkage method does not work particularly well with these data, exhibiting substantial chaining that is typical when clusters are not circular or spherical.*

```
RW_SL <- hclust(dist(RW[, 1:6], method = "euclidean"), method = "single")
```

```
ggdendrogram(RW_SL, theme_dendro = FALSE, size = 4) + xlab("People") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 4, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
```

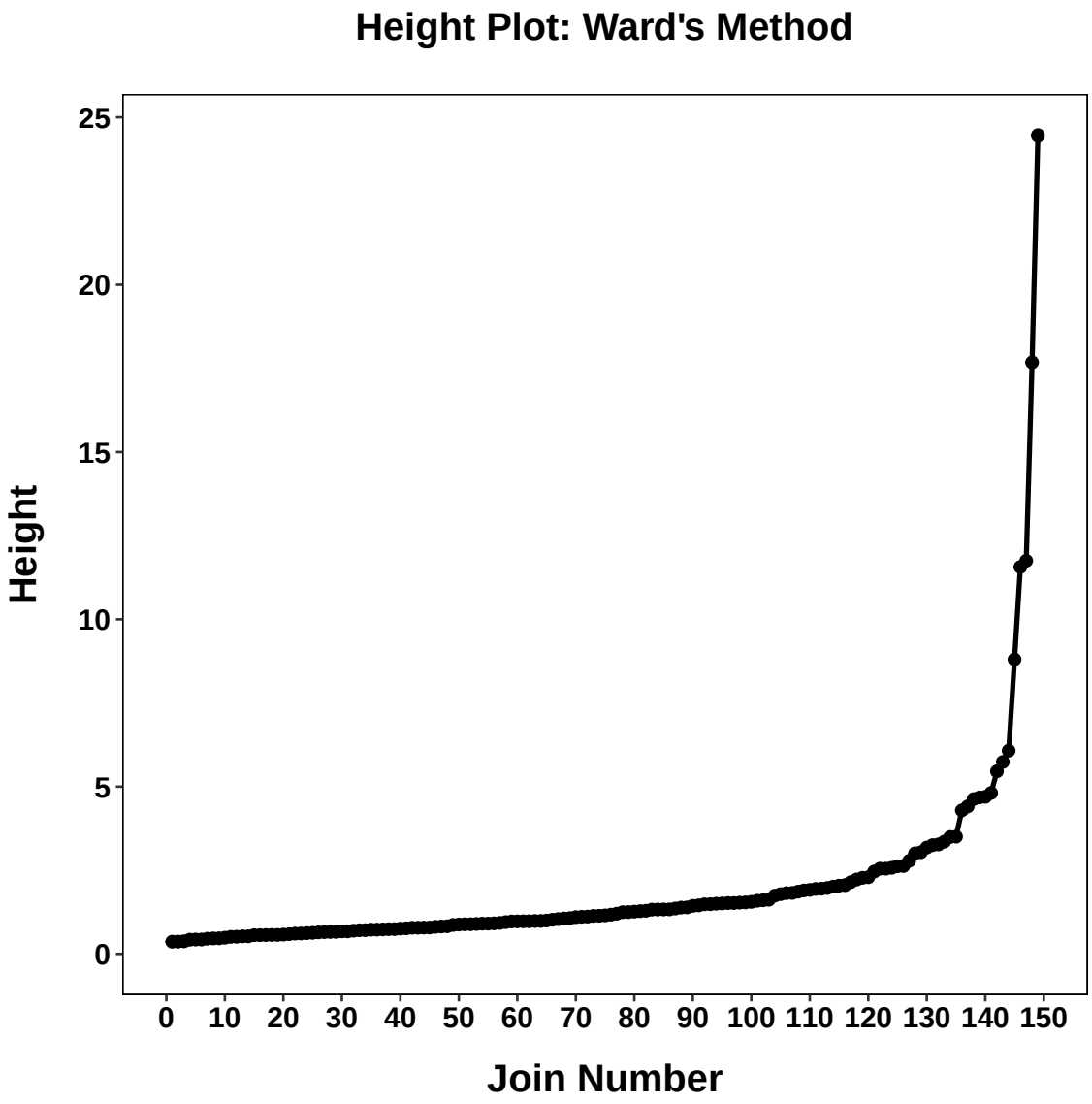
```
    15, 0, 0), size = 16, angle = 90), axis.line.x = element_blank(),
    axis.line.y = element_blank(), plot.title = element_text(size = 16,
        face = "bold", margin = margin(0, 0, 20, 0), hjust = 0.5),
    panel.background = element_rect(fill = "white", linetype = 1,
        color = "black"), panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "white"), plot.margin = unit(c(1,
        1, 1, 1), "cm"), legend.position = "bottom", legend.title = element_blank()) +
    ggtitle("Cluster Dendogram: Single Linkage Method")
```




### 3.2.6 Height Plot

*A plot of the height at which joining occurs can sometimes provide insight into the number of clusters that best simplifies the data.*

```r
H <- matrix(RW_HC$height, nrow = 149)
plot_data <- cbind(seq(1, 149, 1), H)
plot_data <- as.data.frame(plot_data)
names(plot_data) <- c("Join", "Height")

ggplot(plot_data, aes(x = Join, y = Height)) + geom_point(shape = 19,
    size = 2, color = "black", na.rm = TRUE) + geom_line(size = 1) +
    scale_x_continuous(breaks = c(seq(0, 150, 10))) + coord_cartesian(xlim = c(0,
    150), ylim = c(0, max(plot_data$Height))) + xlab("Join Number") +
    ylab("Height") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Height Plot: Ward's Method")
```




### 3.2.7 K-Means Clustering

*We'll take a look at 6 clusters and see how they compare to the hierarchical method and also determine if they help with the income and education data.*

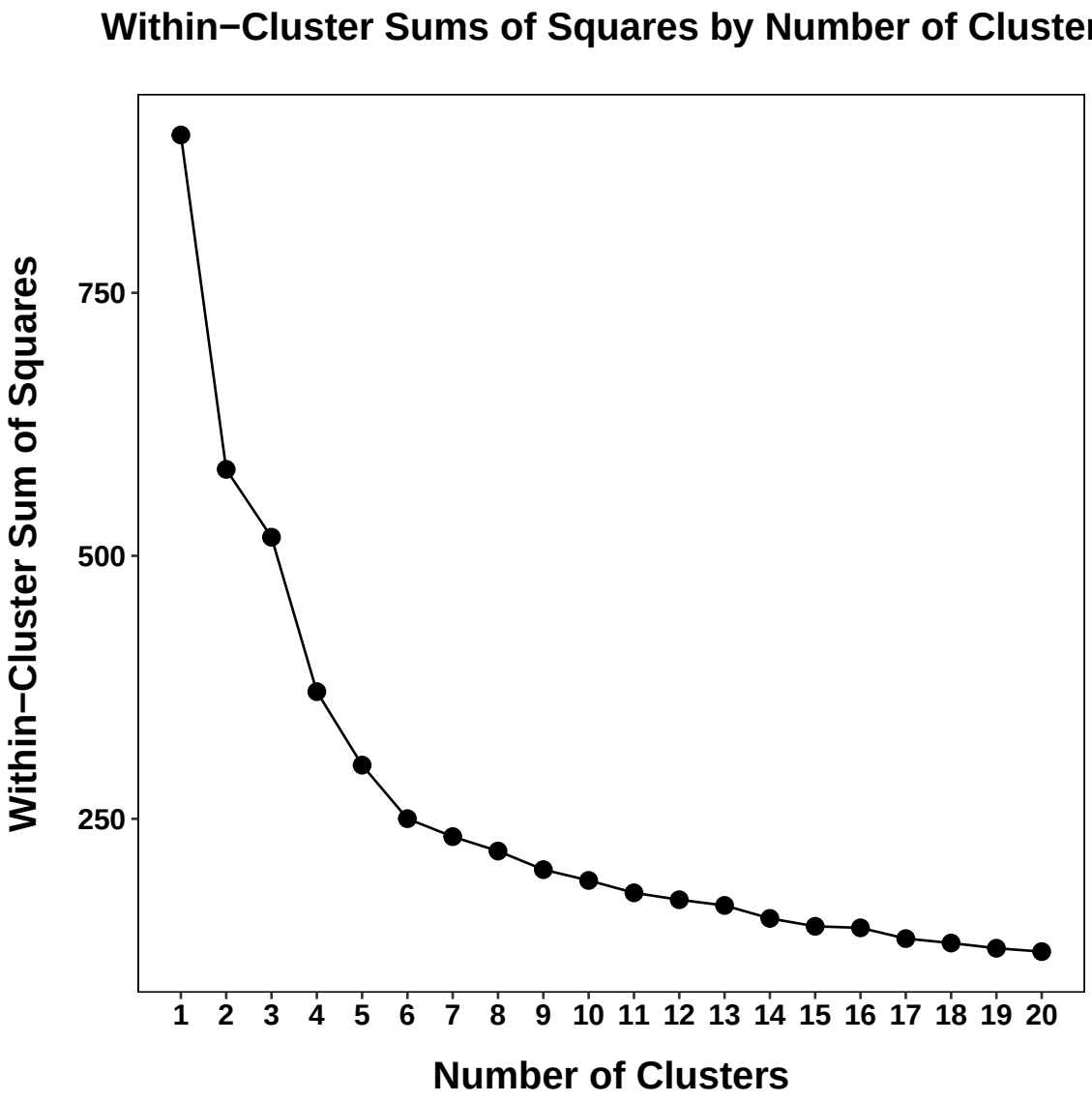
```
RW_K <- kmeans(RW[, 1:6], centers = 6, iter.max = 1000, nstart = 10)
RW_K$centers

##   educate   income     gun   prayer   death samesex
## 1 -0.6520 -0.53117 -1.0759 -0.25048 -0.3681 -0.7874
## 2 -0.5419 -0.52031 -0.1332  0.72080  0.9424 -0.4107
## 3 -1.5286  3.33375 -0.7145  0.74446  1.0163 -0.8991
## 4  1.1115  0.09967  1.3200 -0.09148 -0.5743  1.2798
## 5  2.4718  1.77354  1.2292  1.97471  1.4809 -0.4295
```

```
## 6  0.6156 -0.10194  0.2677 -1.10908 -0.7908  0.6165

RW_Clusters_K <- as.data.frame(RW_K$cluster)
names(RW_Clusters_K) <- c("Cluster_K")
RW_New_K <- as.data.frame(cbind(RW, RW_Clusters_K))
aggregate(RW_New_K, by = list(RW_New_K$Cluster), mean)

##   Group.1 educate   income     gun   prayer   death samesex
## 1       1 -0.6520 -0.53117 -1.0759 -0.25048 -0.3681 -0.7874
## 2       2 -0.5419 -0.52031 -0.1332  0.72080  0.9424 -0.4107
## 3       3 -1.5286  3.33375 -0.7145  0.74446  1.0163 -0.8991
## 4       4  1.1115  0.09967  1.3200 -0.09148 -0.5743  1.2798
## 5       5  2.4718  1.77354  1.2292  1.97471  1.4809 -0.4295
## 6       6  0.6156 -0.10194  0.2677 -1.10908 -0.7908  0.6165
##   Cluster_K
## 1         1
## 2         2
## 3         3
## 4         4
## 5         5
## 6         6

summary(aov(RW_New_K$educate ~ as.factor(RW_New_K$Cluster_K)))

##                               Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_K$Cluster_K)   5  123.3   24.67     138 <2e-16
## Residuals                     144   25.7    0.18

summary(aov(RW_New_K$income ~ as.factor(RW_New_K$Cluster_K)))

##                               Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_K$Cluster_K)   5  145.6   29.12    1229 <2e-16
## Residuals                     144    3.4    0.02

summary(aov(RW_New_K$gun ~ as.factor(RW_New_K$Cluster_K)))

##                               Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_K$Cluster_K)   5    109   21.81    78.6 <2e-16
## Residuals                     144     40    0.28

summary(aov(RW_New_K$prayer ~ as.factor(RW_New_K$Cluster_K)))

##                               Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_K$Cluster_K)   5   80.2   16.03    33.5 <2e-16
## Residuals                     144   68.8    0.48

summary(aov(RW_New_K$death ~ as.factor(RW_New_K$Cluster_K)))

##                               Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_K$Cluster_K)   5   87.7   17.53    41.2 <2e-16
## Residuals                     144   61.3    0.43

summary(aov(RW_New_K$samesex ~ as.factor(RW_New_K$Cluster_K)))

##                               Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(RW_New_K$Cluster_K)   5   98.7   19.73    56.5 <2e-16
## Residuals                     144   50.3    0.35
```

### 3.2.8 Plot of Within-Cluster Sums of Squares

*A plot of the within-cluster sums of squares for different numbers of clusters with the K-Means method can be used as well to determine a good choice for number of clusters. Here too 6 clusters appears to be the best solution.*

```
wssplot(RW[, 1:6], nc = 20)
```

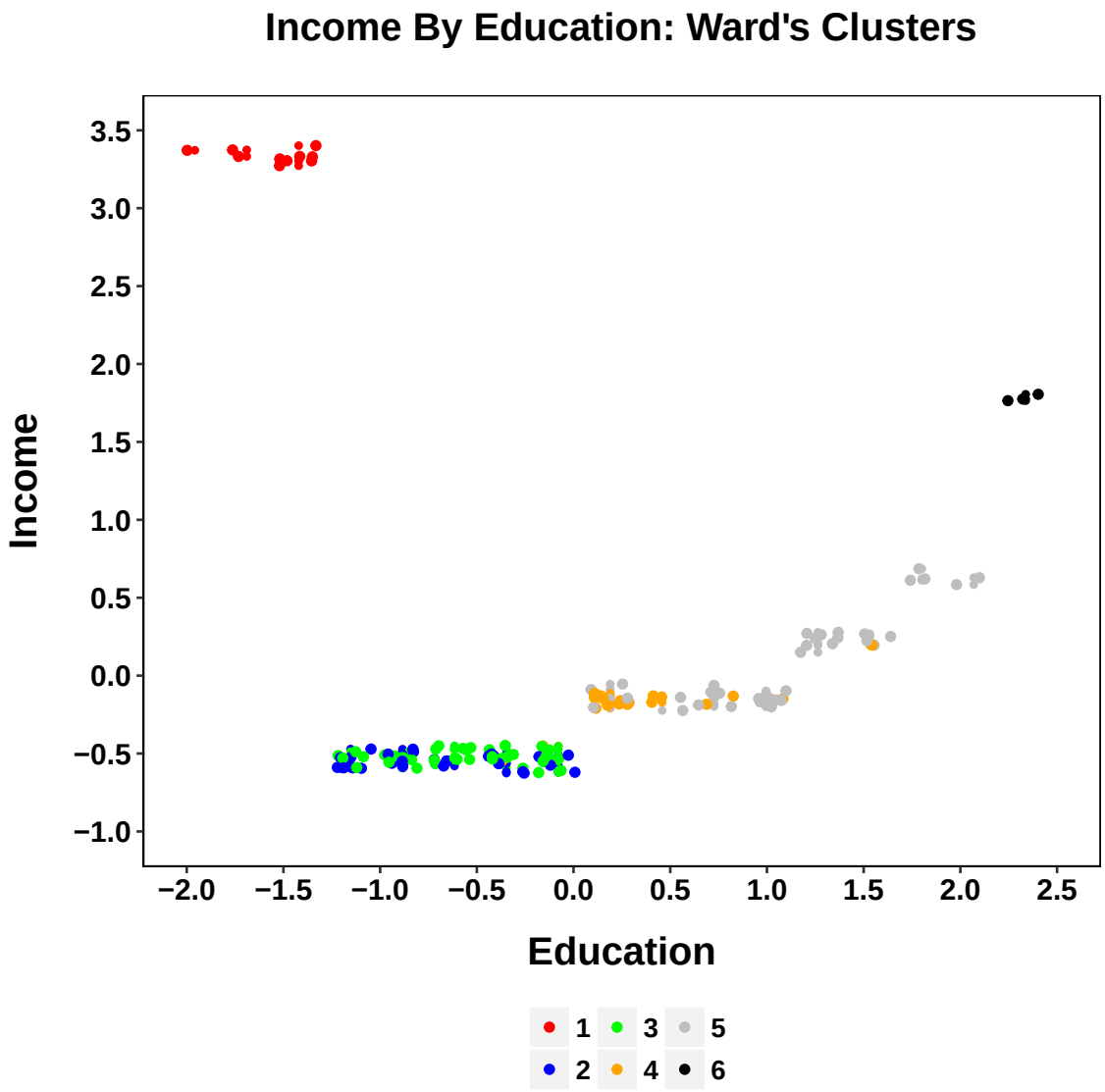



### 3.2.9 How Well Do the Methods Agree?

```
RW_New <- as.data.frame(cbind(RW, RW_Clusters_H, RW_Clusters_K))
Cross_T <- table(RW_New$Cluster_H, RW_New$Cluster_K)
```

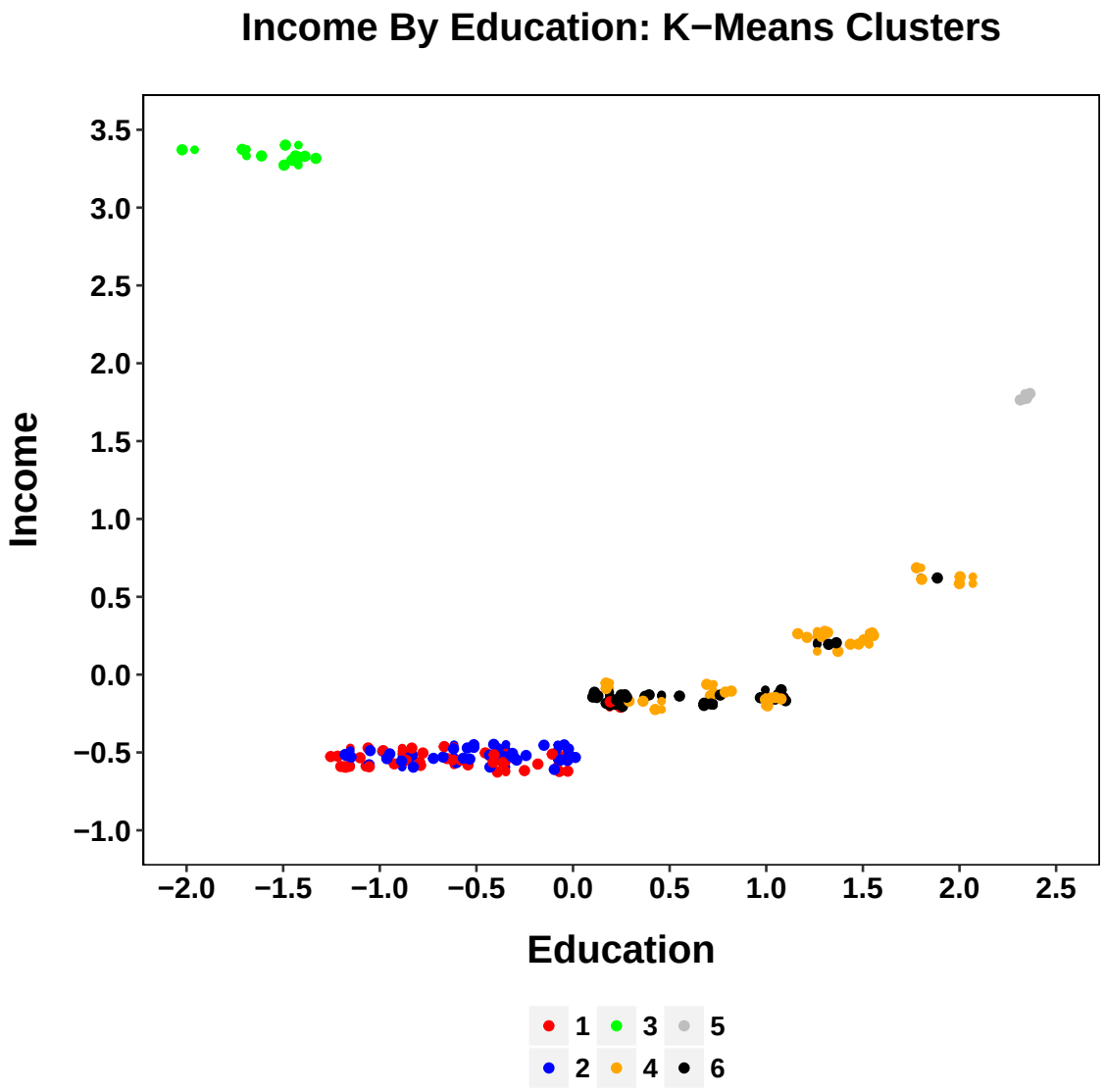
| | | K-Means Clusters | | | | | |
|---|---|---|---|---|---|---|---|
| | | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 | Cluster 6 |
| Ward's Clusters | Cluster 1 | 0 | 0 | 10 | 0 | 0 | 0 |
| | Cluster 2 | 32 | 1 | 0 | 0 | 0 | 0 |
| | Cluster 3 | 3 | 37 | 0 | 0 | 0 | 0 |
| | Cluster 4 | 2 | 2 | 0 | 2 | 0 | 17 |
| | Cluster 5 | 0 | 0 | 0 | 28 | 0 | 12 |
| | Cluster 6 | 0 | 0 | 0 | 0 | 4 | 0 |

```
ggplot(RW_New_HC, aes(x = educate, y = income, color = factor(Cluster_H))) +
    geom_point(shape = 19, size = 1) + geom_jitter() + scale_color_manual(values = c("red",
    "blue", "green", "orange", "gray", "black")) + scale_y_continuous(breaks = c(seq(-1,
    3.5, 0.5))) + scale_x_continuous(breaks = c(seq(-2, 2.5, 0.5))) +
    coord_cartesian(xlim = c(-2, 2.5), ylim = c(-1, 3.5)) + xlab("Education") +
    ylab("Income") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Income By Education: Ward's Clusters")
```



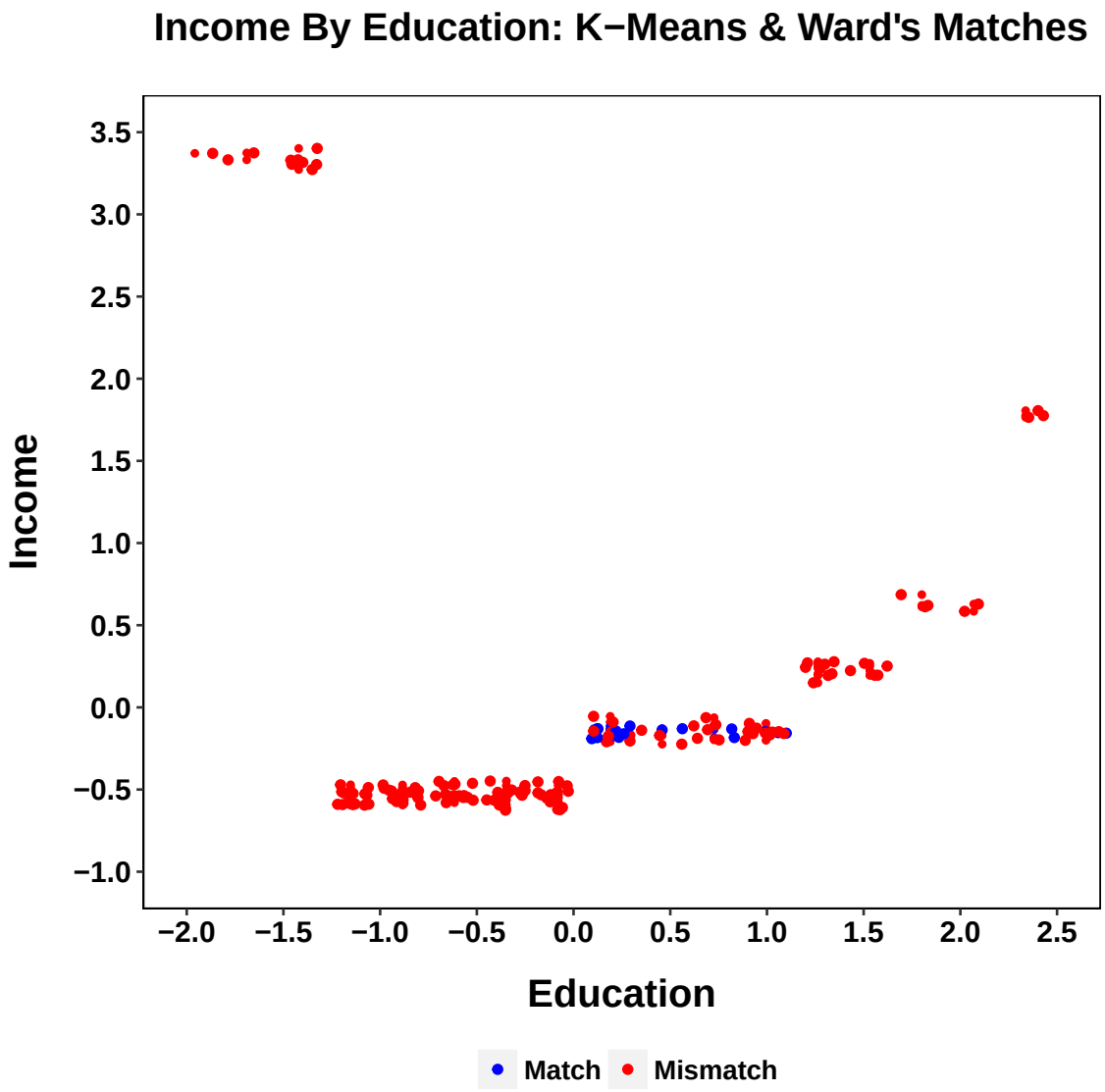

```
ggplot(RW_New_K, aes(x = educate, y = income, color = factor(Cluster_K))) +
    geom_point(shape = 19, size = 1) + geom_jitter() + scale_color_manual(values = c("red",
    "blue", "green", "orange", "gray", "black")) + scale_y_continuous(breaks = c(seq(-1,
    3.5, 0.5))) + scale_x_continuous(breaks = c(seq(-2, 2.5, 0.5))) +
    coord_cartesian(xlim = c(-2, 2.5), ylim = c(-1, 3.5)) + xlab("Education") +
    ylab("Income") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
```

```
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Income By Education: K-Means Clusters")
```




```
RW_New_K$Cluster_HC <- RW_New_HC$Cluster_H
for (i in seq(1, length(RW_New_K[, 1]))) {
    RW_New_K[i, "Match"] <- "Mismatch"
    if (RW_New_K[i, "Cluster_HC"] == 6 & RW_New_K[i, "Cluster_K"] ==
        1) {
        RW_New_K[i, "Match"] <- "Match"
    }
    if (RW_New_K[i, "Cluster_HC"] == 5 & RW_New_K[i, "Cluster_K"] ==
        2) {
```

```
        RW_New_K[i, "Match"] <- "Match"
    }
    if (RW_New_K[i, "Cluster_HC"] == 3 & RW_New_K[i, "Cluster_K"] ==
        3) {
        RW_New_K[i, "Match"] <- "Match"
    }
    if (RW_New_K[i, "Cluster_HC"] == 2 & RW_New_K[i, "Cluster_K"] ==
        4) {
        RW_New_K[i, "Match"] <- "Match"
    }
    if (RW_New_K[i, "Cluster_HC"] == 1 & RW_New_K[i, "Cluster_K"] ==
        5) {
        RW_New_K[i, "Match"] <- "Match"
    }
    if (RW_New_K[i, "Cluster_HC"] == 4 & RW_New_K[i, "Cluster_K"] ==
        6) {
        RW_New_K[i, "Match"] <- "Match"
    }
}

ggplot(RW_New_K, aes(x = educate, y = income, color = factor(Match))) +
    geom_point(shape = 19, size = 1) + geom_jitter() + scale_color_manual(values = c("blue",
    "red")) + scale_y_continuous(breaks = c(seq(-1, 3.5, 0.5))) +
    scale_x_continuous(breaks = c(seq(-2, 2.5, 0.5))) + coord_cartesian(xlim = c(-2,
    2.5), ylim = c(-1, 3.5)) + xlab("Education") + ylab("Income") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Income By Education: K-Means & Ward's Matches")
```

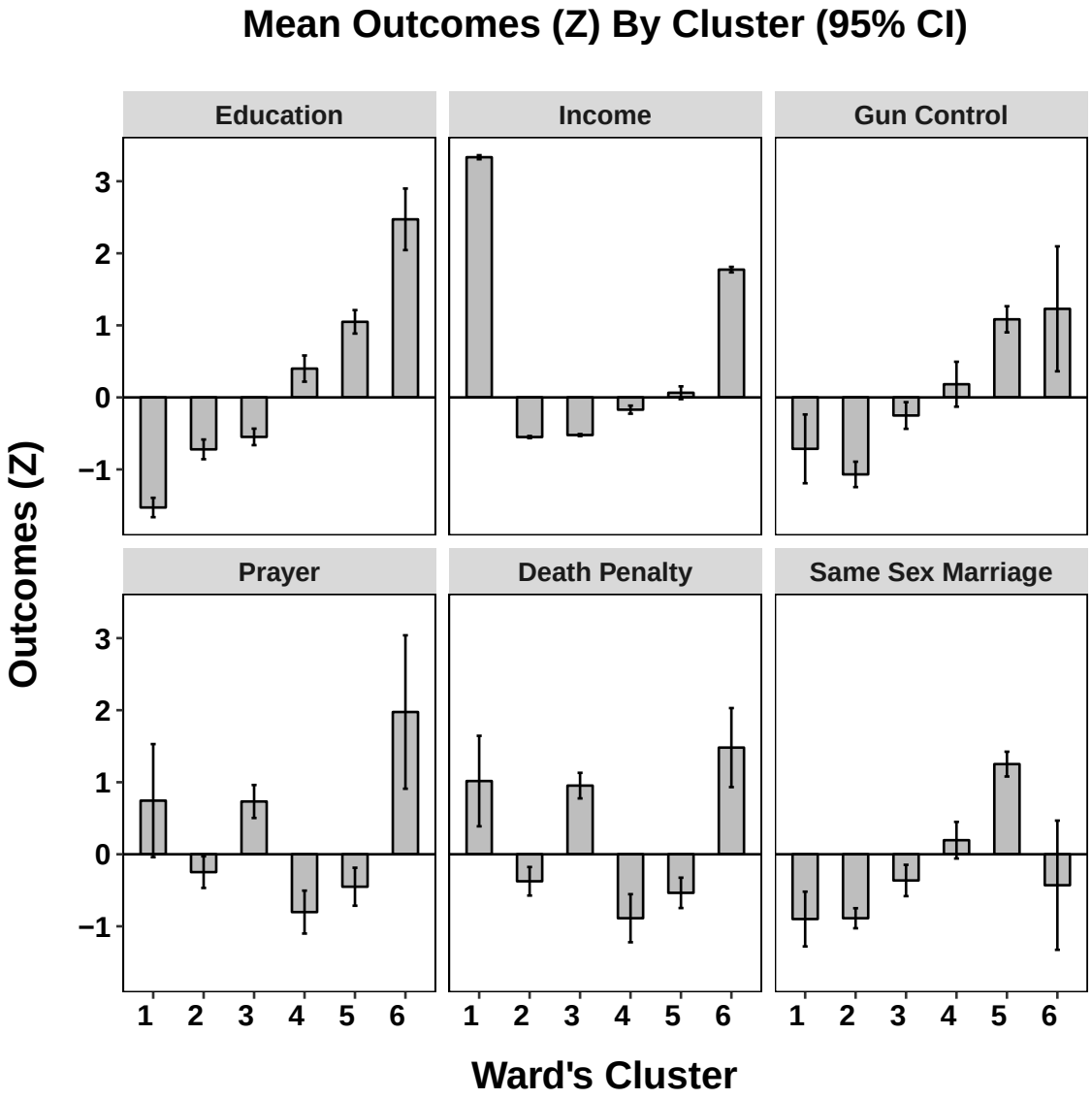



### 3.2.10 Plot of Cluster Means on Original Variables

```
SE_educate <- ddply(RW_New_HC, ~Cluster_H, summarise, se = se(educate))
Means_educate <- ddply(RW_New_HC, ~Cluster_H, summarise, mean = mean(educate))
N_educate <- table(RW_New_HC$Cluster_H)
educate <- cbind(Means_educate, SE_educate$se, N_educate)
educate <- as.data.frame(educate)
educate <- educate[-4]
names(educate) <- c("Cluster", "Mean", "SE", "N")

SE_income <- ddply(RW_New_HC, ~Cluster_H, summarise, se = se(income))
Means_income <- ddply(RW_New_HC, ~Cluster_H, summarise, mean = mean(income))
N_income <- table(RW_New_HC$Cluster_H)
```

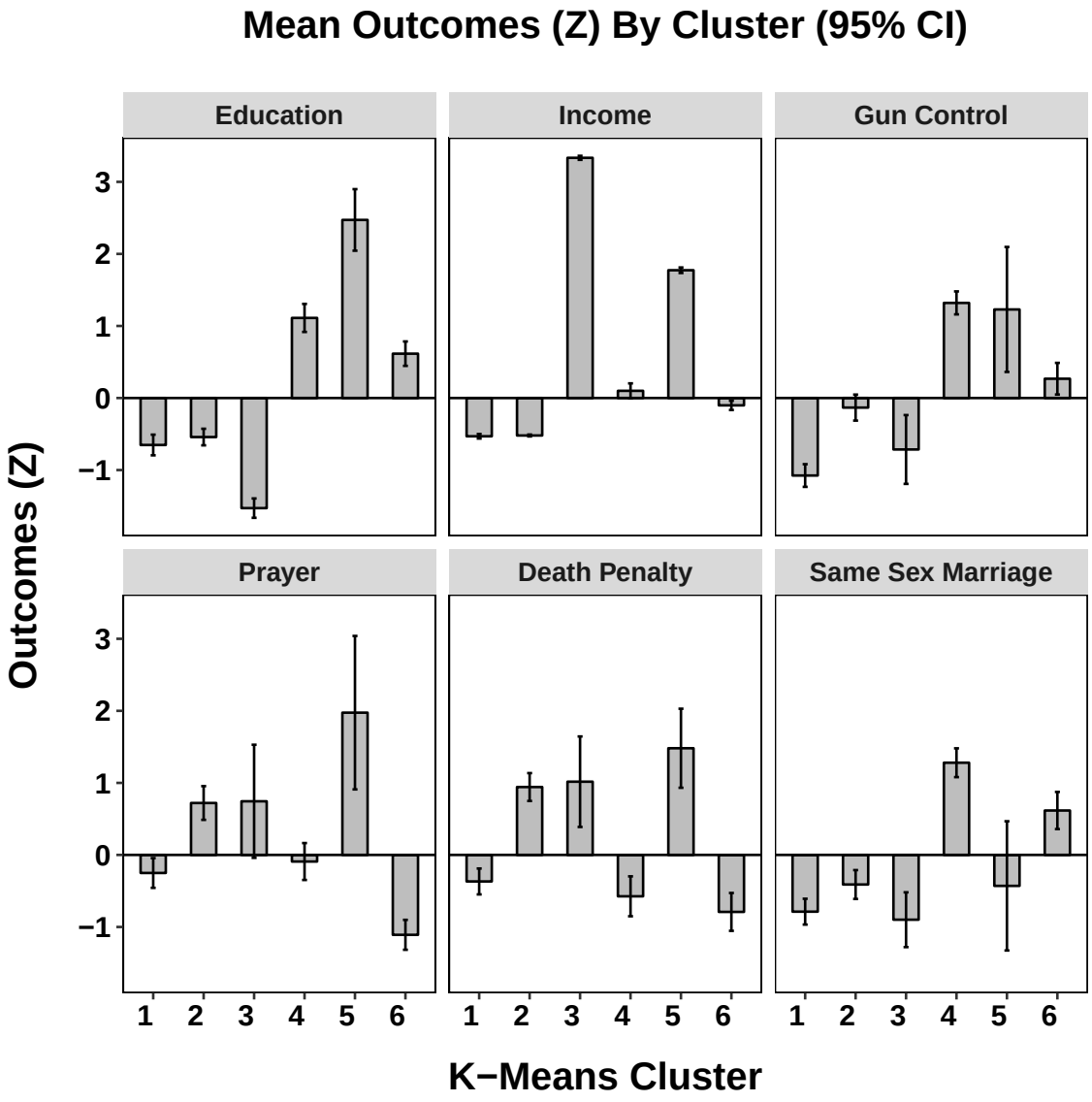
```
income <- cbind(Means_income, SE_income$se, N_income)
income <- as.data.frame(income)
income <- income[-4]
names(income) <- c("Cluster", "Mean", "SE", "N")

SE_gun <- ddply(RW_New_HC, ~Cluster_H, summarise, se = se(gun))
Means_gun <- ddply(RW_New_HC, ~Cluster_H, summarise, mean = mean(gun))
N_gun <- table(RW_New_HC$Cluster_H)
gun <- cbind(Means_gun, SE_gun$se, N_gun)
gun <- as.data.frame(gun)
gun <- gun[-4]
names(gun) <- c("Cluster", "Mean", "SE", "N")

SE_prayer <- ddply(RW_New_HC, ~Cluster_H, summarise, se = se(prayer))
Means_prayer <- ddply(RW_New_HC, ~Cluster_H, summarise, mean = mean(prayer))
N_prayer <- table(RW_New_HC$Cluster_H)
prayer <- cbind(Means_prayer, SE_prayer$se, N_prayer)
prayer <- as.data.frame(prayer)
prayer <- prayer[-4]
names(prayer) <- c("Cluster", "Mean", "SE", "N")

SE_death <- ddply(RW_New_HC, ~Cluster_H, summarise, se = se(death))
Means_death <- ddply(RW_New_HC, ~Cluster_H, summarise, mean = mean(death))
N_death <- table(RW_New_HC$Cluster_H)
death <- cbind(Means_death, SE_death$se, N_death)
death <- as.data.frame(death)
death <- death[-4]
names(death) <- c("Cluster", "Mean", "SE", "N")

SE_samesex <- ddply(RW_New_HC, ~Cluster_H, summarise, se = se(samesex))
Means_samesex <- ddply(RW_New_HC, ~Cluster_H, summarise, mean = mean(samesex))
N_samesex <- table(RW_New_HC$Cluster_H)
samesex <- cbind(Means_samesex, SE_samesex$se, N_samesex)
samesex <- as.data.frame(samesex)
samesex <- samesex[-4]
names(samesex) <- c("Cluster", "Mean", "SE", "N")

plot_data <- rbind(educate, income, gun, prayer, death, samesex)
plot_data$Feature <- c(rep("Education", 6), rep("Income", 6), rep("Gun Control",
    6), rep("Prayer", 6), rep("Death Penalty", 6), rep("Same Sex Marriage",
    6))
plot_data$Feature <- factor(plot_data$Feature, levels = c("Education",
    "Income", "Gun Control", "Prayer", "Death Penalty", "Same Sex Marriage"))
plot_data$CI_L <- plot_data$Mean - plot_data$SE * qt(0.975, df = plot_data$N -
    1)
plot_data$CI_U <- plot_data$Mean + plot_data$SE * qt(0.975, df = plot_data$N -
    1)
plot_data$Cluster <- factor(plot_data$Cluster)

p <- ggplot(plot_data, aes(x = Cluster, y = Mean)) + geom_bar(position = position_dodge(),
    stat = "identity", color = "black", width = 0.5, fill = "grey") +
    geom_errorbar(aes(ymin = CI_L, ymax = CI_U), width = 0.1, position = position_dodge(0.5)) +
    xlab("Ward's Cluster") + ylab("Outcomes (Z)") + theme(text = element_text(size = 14,
```

```
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 1), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + geom_hline(yintercept = 0) +
    ggtitle("Mean Outcomes (Z) By Cluster (95% CI)")
p + facet_wrap(~Feature, ncol = 3)
```



```
SE_educate <- ddply(RW_New_K, ~Cluster_K, summarise, se = se(educate))
Means_educate <- ddply(RW_New_K, ~Cluster_K, summarise, mean = mean(educate))
N_educate <- table(RW_New_K$Cluster_K)
educate <- cbind(Means_educate, SE_educate$se, N_educate)
educate <- as.data.frame(educate)
educate <- educate[-4]
names(educate) <- c("Cluster", "Mean", "SE", "N")

SE_income <- ddply(RW_New_K, ~Cluster_K, summarise, se = se(income))
Means_income <- ddply(RW_New_K, ~Cluster_K, summarise, mean = mean(income))
N_income <- table(RW_New_K$Cluster_K)
income <- cbind(Means_income, SE_income$se, N_income)
income <- as.data.frame(income)
income <- income[-4]
names(income) <- c("Cluster", "Mean", "SE", "N")

SE_gun <- ddply(RW_New_K, ~Cluster_K, summarise, se = se(gun))
Means_gun <- ddply(RW_New_K, ~Cluster_K, summarise, mean = mean(gun))
N_gun <- table(RW_New_K$Cluster_K)
gun <- cbind(Means_gun, SE_gun$se, N_gun)
gun <- as.data.frame(gun)
gun <- gun[-4]
names(gun) <- c("Cluster", "Mean", "SE", "N")

SE_prayer <- ddply(RW_New_K, ~Cluster_K, summarise, se = se(prayer))
Means_prayer <- ddply(RW_New_K, ~Cluster_K, summarise, mean = mean(prayer))
N_prayer <- table(RW_New_K$Cluster_K)
prayer <- cbind(Means_prayer, SE_prayer$se, N_prayer)
prayer <- as.data.frame(prayer)
prayer <- prayer[-4]
names(prayer) <- c("Cluster", "Mean", "SE", "N")

SE_death <- ddply(RW_New_K, ~Cluster_K, summarise, se = se(death))
Means_death <- ddply(RW_New_K, ~Cluster_K, summarise, mean = mean(death))
N_death <- table(RW_New_K$Cluster_K)
death <- cbind(Means_death, SE_death$se, N_death)
death <- as.data.frame(death)
death <- death[-4]
names(death) <- c("Cluster", "Mean", "SE", "N")

SE_samesex <- ddply(RW_New_K, ~Cluster_K, summarise, se = se(samesex))
Means_samesex <- ddply(RW_New_K, ~Cluster_K, summarise, mean = mean(samesex))
N_samesex <- table(RW_New_K$Cluster_K)
samesex <- cbind(Means_samesex, SE_samesex$se, N_samesex)
samesex <- as.data.frame(samesex)
samesex <- samesex[-4]
names(samesex) <- c("Cluster", "Mean", "SE", "N")

plot_data <- rbind(educate, income, gun, prayer, death, samesex)
plot_data$Feature <- c(rep("Education", 6), rep("Income", 6), rep("Gun Control",
    6), rep("Prayer", 6), rep("Death Penalty", 6), rep("Same Sex Marriage",
    6))
plot_data$Feature <- factor(plot_data$Feature, levels = c("Education",
```

```
    "Income", "Gun Control", "Prayer", "Death Penalty", "Same Sex Marriage"))
plot_data$CI_L <- plot_data$Mean - plot_data$SE * qt(0.975, df = plot_data$N -
    1)
plot_data$CI_U <- plot_data$Mean + plot_data$SE * qt(0.975, df = plot_data$N -
    1)
plot_data$Cluster <- factor(plot_data$Cluster)

p <- ggplot(plot_data, aes(x = Cluster, y = Mean)) + geom_bar(position = position_dodge(),
    stat = "identity", color = "black", width = 0.5, fill = "grey") +
    geom_errorbar(aes(ymin = CI_L, ymax = CI_U), width = 0.1, position = position_dodge(0.5)) +
    xlab("K-Means Cluster") + ylab("Outcomes (Z)") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0, hjust = 1), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + geom_hline(yintercept = 0) +
    ggtitle("Mean Outcomes (Z) By Cluster (95% CI)")
p + facet_wrap(~Feature, ncol = 3)
```


Mean Outcomes (Z) By Cluster (95% CI)
Education
Income
Gun Control
Prayer
Death Penalty
Same Sex Marriage
Outcomes (Z)
3
2
1
0
−1
1
2
3
4
5
6
K−Means Cluster