Load required libraries. knitr is necessary for producing this html document. rgl is necessary for 3-dimensional plots. Set the random number generator seed, so we get the same results from clustering each time.

```r
library(knitr)
library(rgl)
set.seed(123456)
knit_hooks$set(webgl = hook_webgl)
cat('<script type="text/javascript">', readLines(system.file('WebGL', 'CanvasMatrix.js', package = 'rgl')), '</script>', sep = '\n')
```

Load the Motor Trend data and take a look at the first few rows.

```r
data(mtcars)
head(mtcars)
```

Center and scale the data by subtracting out column means and dividing by the standard deviations of the columns.

```r
mydata <- scale(mtcars, center=T, scale=T)
```

Apply principal component analysis (PCA). The data are already centered and scaled, so there is no need for prcomp() to do it again. The option retx=T indicates that we will get the scores.

```r
mypca <- prcomp(mydata, center=F, scale=F, retx=T)
```

View the principal component standard deviations, variance explained, and cumulative variance explained for the principal components. The information for Table 1 comes from this view.

```r
summary(mypca)
```

View the principal component loadings. The information for Table 2 comes from this view.

```r
mypca$rotation
```
## Principal Component Analysis

### Table 3: Principal Component Scores

<table>
<thead>
<tr>
<th>Variable</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>-0.3625</td>
<td>0.01612</td>
<td>-0.22574</td>
<td>-0.022540</td>
<td>0.10284</td>
<td>-0.10880</td>
<td>0.367724</td>
</tr>
<tr>
<td>cyl</td>
<td>0.3739</td>
<td>0.04374</td>
<td>-0.17531</td>
<td>-0.002592</td>
<td>0.05848</td>
<td>0.16855</td>
<td>0.057278</td>
</tr>
<tr>
<td>disp</td>
<td>0.3682</td>
<td>-0.04932</td>
<td>-0.06148</td>
<td>0.256608</td>
<td>0.39400</td>
<td>-0.33616</td>
<td>0.214303</td>
</tr>
<tr>
<td>hp</td>
<td>0.3301</td>
<td>0.24878</td>
<td>0.14001</td>
<td>-0.067676</td>
<td>0.54005</td>
<td>0.07144</td>
<td>-0.001496</td>
</tr>
<tr>
<td>drat</td>
<td>-0.2942</td>
<td>0.27469</td>
<td>0.16119</td>
<td>0.854829</td>
<td>0.07733</td>
<td>0.24450</td>
<td>0.021120</td>
</tr>
<tr>
<td>wt</td>
<td>0.3461</td>
<td>-0.14304</td>
<td>0.34182</td>
<td>0.245899</td>
<td>-0.07503</td>
<td>-0.46494</td>
<td>-0.020668</td>
</tr>
<tr>
<td>qsec</td>
<td>-0.2005</td>
<td>-0.46337</td>
<td>0.40317</td>
<td>0.068077</td>
<td>-0.16467</td>
<td>-0.33048</td>
<td>0.050011</td>
</tr>
<tr>
<td>vs</td>
<td>-0.3065</td>
<td>-0.23165</td>
<td>0.42882</td>
<td>-0.214849</td>
<td>0.59954</td>
<td>0.19402</td>
<td>-0.265781</td>
</tr>
<tr>
<td>am</td>
<td>-0.2349</td>
<td>0.42942</td>
<td>-0.20577</td>
<td>-0.030463</td>
<td>0.08978</td>
<td>-0.57082</td>
<td>-0.587305</td>
</tr>
<tr>
<td>gear</td>
<td>-0.2069</td>
<td>0.46235</td>
<td>0.28978</td>
<td>-0.264691</td>
<td>0.04833</td>
<td>-0.24356</td>
<td>0.605098</td>
</tr>
<tr>
<td>carb</td>
<td>0.2140</td>
<td>0.41357</td>
<td>0.52854</td>
<td>-0.126789</td>
<td>-0.36132</td>
<td>0.18352</td>
<td>-0.174603</td>
</tr>
</tbody>
</table>

### Table 4: Principal Component Scores (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>PC8</th>
<th>PC9</th>
<th>PC10</th>
<th>PC11</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>-0.754091</td>
<td>0.235702</td>
<td>0.13929</td>
<td>-0.124896</td>
</tr>
<tr>
<td>cyl</td>
<td>-0.230825</td>
<td>0.054035</td>
<td>-0.84642</td>
<td>-0.140695</td>
</tr>
<tr>
<td>disp</td>
<td>0.001142</td>
<td>0.198428</td>
<td>0.04938</td>
<td>0.660606</td>
</tr>
<tr>
<td>hp</td>
<td>-0.222358</td>
<td>-0.575830</td>
<td>0.24782</td>
<td>-0.256492</td>
</tr>
<tr>
<td>drat</td>
<td>0.032194</td>
<td>-0.046901</td>
<td>-0.10149</td>
<td>-0.039530</td>
</tr>
<tr>
<td>wt</td>
<td>-0.008572</td>
<td>0.359498</td>
<td>0.09439</td>
<td>-0.567449</td>
</tr>
<tr>
<td>qsec</td>
<td>-0.231840</td>
<td>-0.528377</td>
<td>-0.27067</td>
<td>0.181362</td>
</tr>
<tr>
<td>vs</td>
<td>0.025935</td>
<td>0.358583</td>
<td>-0.15904</td>
<td>0.008415</td>
</tr>
<tr>
<td>am</td>
<td>-0.059747</td>
<td>-0.047404</td>
<td>-0.17779</td>
<td>0.029824</td>
</tr>
<tr>
<td>gear</td>
<td>0.336150</td>
<td>-0.001735</td>
<td>-0.21383</td>
<td>-0.053507</td>
</tr>
<tr>
<td>carb</td>
<td>-0.395629</td>
<td>0.170641</td>
<td>0.07226</td>
<td>0.319595</td>
</tr>
</tbody>
</table>

View the principal component scores on the first principal component. The information for Table 3 comes from this view.

```r
comal <- kmeans(mypca$x[,1:10], centers=4, nstart=100)
```

Cluster the points projected on the first three principal components.

```r
comal <- kmeans(mypca$x[,1:3], centers=4, nstart=100)
```

Plot the points projected into the best-fitting three-dimensional subspace, and color the points according to cluster membership. This is the basis for Figure 1. (Use the arrow keys to scroll up and down.)

```r
plot3d(mypca$x[,1:3], xlab="PC 1", ylab="PC 2", zlab="PC 3", cex=1.5, size=1, type="s", col=comal$cluster)
text3d(comal$centers, texts=c("1","2","3","4"))
```
Apply PCA to a subset of the data with only three variables.

```r
my3Ddata <- mydata[,c("cyl", "qsec", "carb")]
my3Dpca <- prcomp(my3Ddata, center=F, scale=F, retx=T)
```

Find the projected points in terms of the original coordinates by multiplying the first two columns of the scores matrix by the transpose of the first two columns of the rotation matrix. This plot is the basis for Figure 4.

```r
my3Dreconstructions <- my3Dpca$x[,1:2] %*% t(my3Dpca$rotation[,1:2])
```

Plot the original points, their (reconstructed) projections in the best-fitting two-dimensional subspace, and line segments between them.

```r
       xlim=c(-2,2), ylim = c(-3,3), zlim = c(-2,4), cex=1.5, size=1, type="s")
planes3d(my3Dpca$rotation[,3], alpha=.5)

plot3d(my3Dreconstructions, col="red", cex=1.5, size=1, add=TRUE,type="s")

mylist <- list(my3Ddata, my3Dreconstructions)
segments3d(do.call("rbind", mylist)[order(sapply(mylist,nrow))],]
```