R Code to Accompany “Principal Component Analysis and Optimization: A Tutorial”

Robert Reris and J. Paul Brooks

November 25, 2014

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.

```
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.

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

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

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

```
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.

```
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.

```
summary(mypca)
```

```
## Importance of components:
##                          PC1   PC2   PC3    PC4    PC5    PC6    PC7
## Standard deviation     2.571 1.628 0.792 0.5192 0.4727 0.4600 0.3678
## Proportion of Variance 0.601 0.241 0.057 0.0245 0.0203 0.0192 0.0123
## Cumulative Proportion  0.601 0.842 0.899 0.9232 0.9436 0.9628 0.9751
##                           PC8   PC9    PC10  PC11
## Standard deviation     0.3506 0.278 0.22811 0.148
## Proportion of Variance 0.0112 0.007 0.00473 0.002
## Cumulative Proportion  0.9863 0.993 0.99800 1.000
```

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

```
mypca$rotation
```

```
##         PC1      PC2      PC3       PC4      PC5      PC6       PC7
## mpg  -0.3625  0.01612 -0.22574 -0.022540  0.10284 -0.10880  0.367724
## cyl   0.3739  0.04374 -0.17531 -0.002592  0.05848  0.16855  0.057278
## disp  0.3682 -0.04932 -0.06148  0.256608  0.39400 -0.33616  0.214303
## hp    0.3301  0.24878  0.14001 -0.067676  0.54005  0.07144 -0.001496
## drat -0.2942  0.27469  0.16119  0.854829  0.07733  0.24450  0.021120
## wt    0.3461 -0.14304  0.34182  0.245899 -0.07503 -0.46494 -0.020668
## qsec -0.2005 -0.46337  0.40317  0.068077 -0.16467 -0.33048  0.050011
## vs   -0.3065 -0.23165  0.42882 -0.214849  0.59954  0.19402 -0.265781
## am   -0.2349  0.42942 -0.20577 -0.030463  0.08978 -0.57082 -0.587305
## gear -0.2069  0.46235  0.28978 -0.264691  0.04833 -0.24356  0.605098
## carb  0.2140  0.41357  0.52854 -0.126789 -0.36132  0.18352 -0.174603
##           PC8       PC9     PC10      PC11
## mpg  -0.754091  0.235702  0.13929 -0.124896
## cyl  -0.230825  0.054035 -0.84642 -0.140695
## disp  0.001142  0.198428  0.04938  0.660606
## hp   -0.222358 -0.575830  0.24782 -0.256492
## drat  0.032194 -0.046901 -0.10149 -0.039530
## wt   -0.008572  0.359498  0.09439 -0.567449
## qsec -0.231840 -0.528377 -0.27067  0.181362
## vs    0.025935  0.358583 -0.15904  0.008415
## am   -0.059747 -0.047404 -0.17779  0.029824
## gear  0.336150 -0.001735 -0.21383 -0.053507
## carb -0.395629  0.170641  0.07226  0.319595
```

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

```
mypca$x[,1]
```

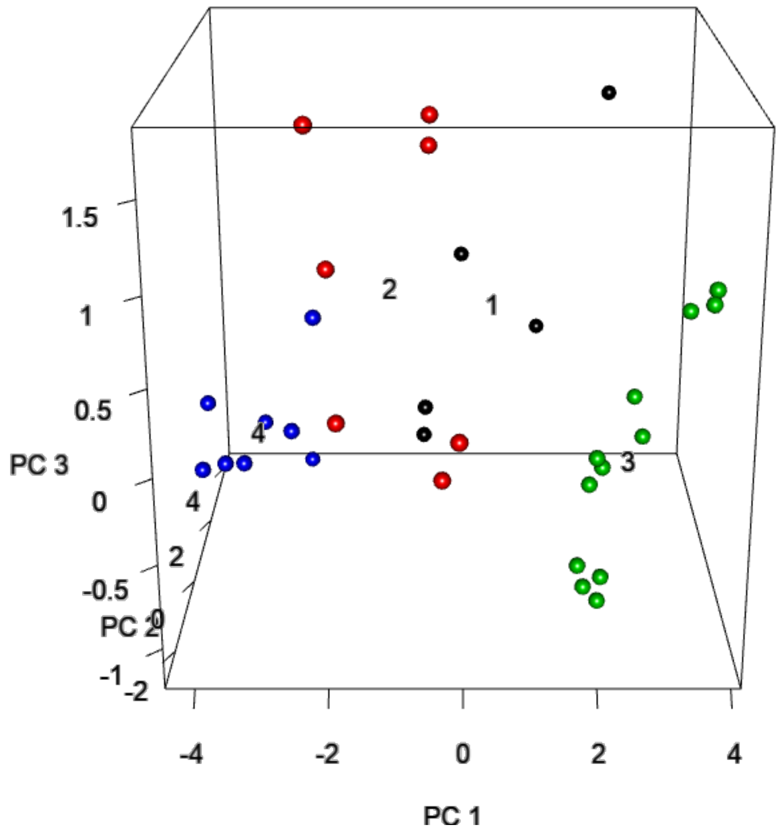
```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##          -0.6468627          -0.6194831          -2.7356243
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##          -0.3068606           1.9433927          -0.0552534
##          Duster 360           Merc 240D            Merc 230
##           2.9553851          -2.0229593          -2.2513840
##            Merc 280           Merc 280C          Merc 450SE
##          -0.5180912          -0.5011860           2.2124096
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##           2.0155716           2.1147047           3.8383725
## Lincoln Continental   Chrysler Imperial            Fiat 128
##           3.8918496           3.5363862          -3.7955511
##         Honda Civic      Toyota Corolla       Toyota Corona
##          -4.1870357          -4.1675359          -1.8741791
##    Dodge Challenger         AMC Javelin          Camaro Z28
##           2.1504415           1.8340370           2.8434958
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##           2.2105479          -3.5176818          -2.6095004
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##          -3.3323845           1.3513347          -0.0009743
##       Maserati Bora          Volvo 142E
##           2.6270898          -2.3824711
```

Cluster the points projected on the first three principal components.

```
myclust <- 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 (interactive!) plot is the basis for Figure 1. (Use the arrow keys to scroll up and down.)

```
plot3d(mypca$x[,1:3], xlab="PC 1", ylab="PC 2", zlab="PC 3", cex=1.5, size=1, type="s", col=myclust$cluster)
text3d(myclust$centers, texts=c("1","2","3","4"))
```



Apply PCA to a subset of the data with only three variables.

```
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.

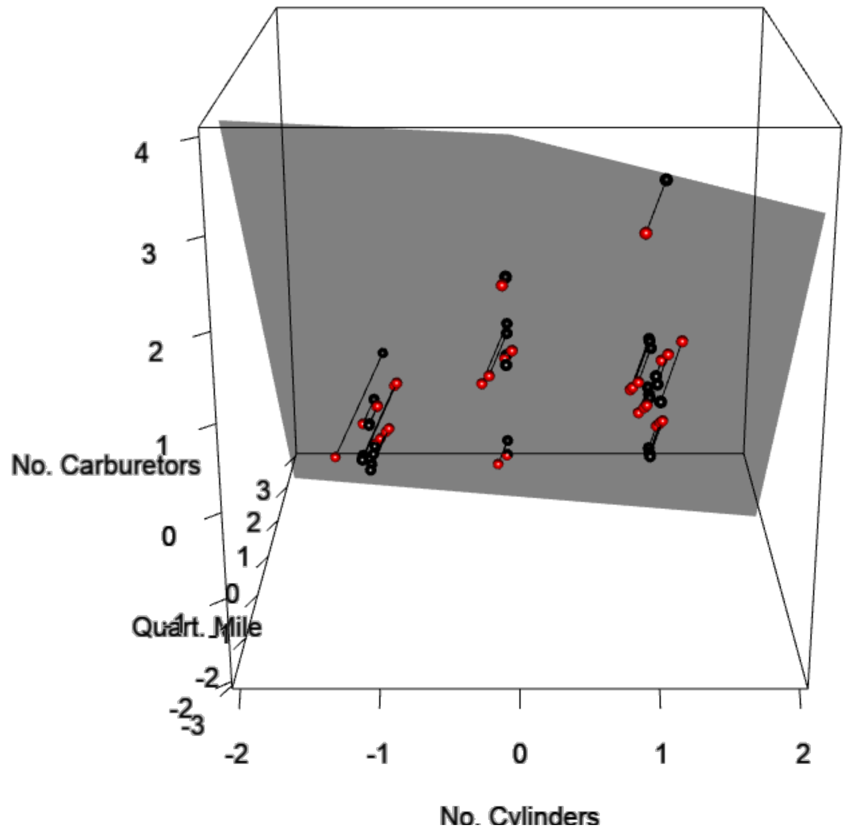
```
my3Dreconstructions <- my3Dpca$x[,1:2] %*% t(my3Dpca$rotation[,1:2])
```

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

```
plot3d(my3Ddata, xlab="No. Cylinders", ylab="Quart. Mile", zlab="No. Carburetors",
    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(sequence(sapply(mylist,nrow))),])
```



No. Carburetors
Quart. Mile
No. Cylinders