503 HW1 Problem 3

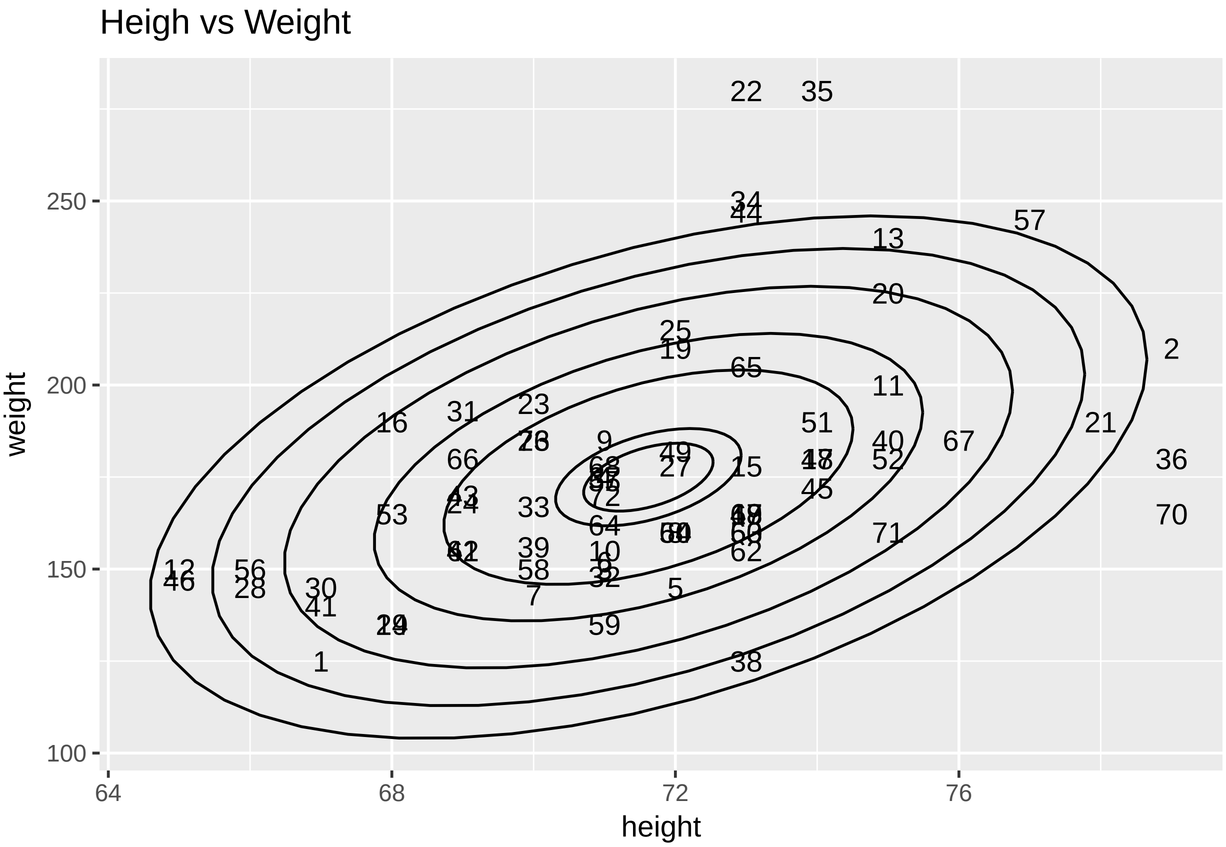
Katherine Wilkinson

1/24/2018

Using height and weight data for males and females, fit a 2-dim Gaussian to the male data, using the empirical mean and covariance. Plot your Gaussian distribution as an ellipse, superimposing on your scatter plot of data points, each which should be labeled by its index number.

(a) Original Data

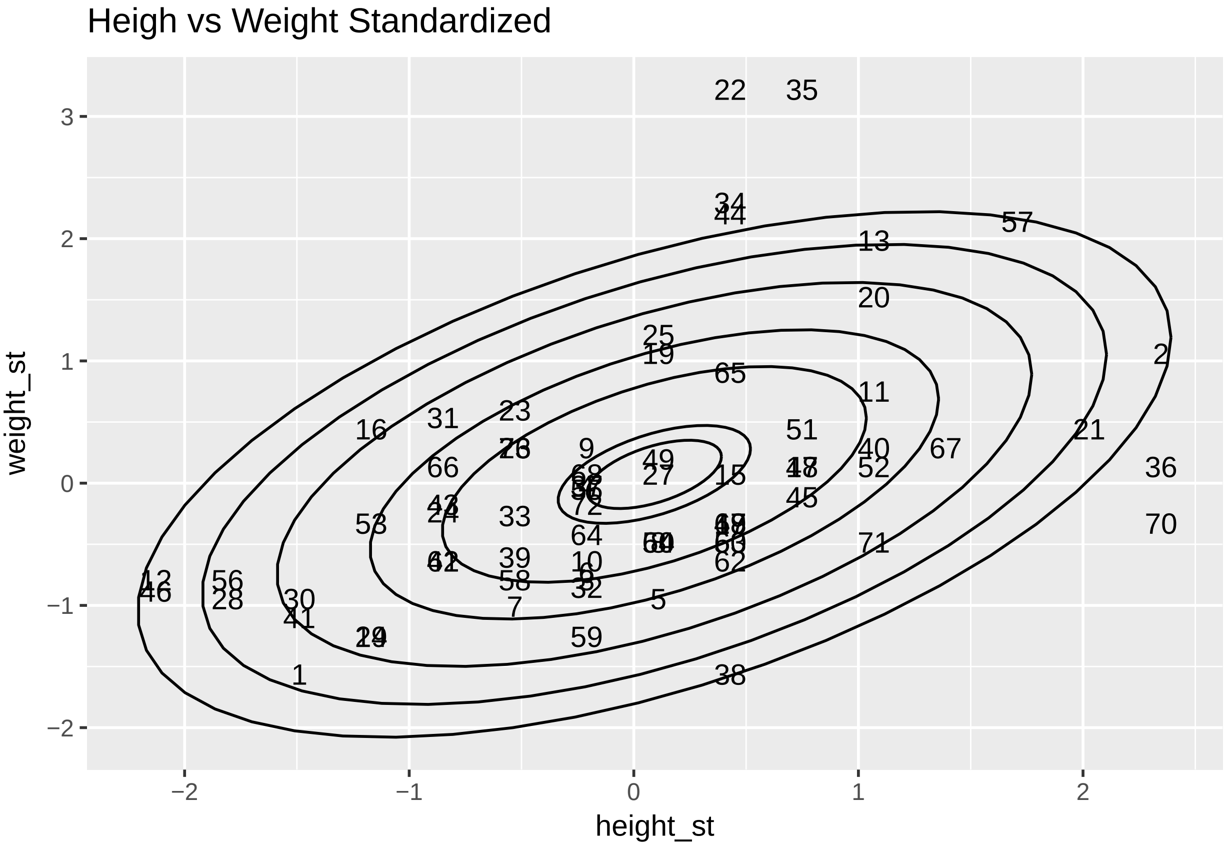
```
rm(list=ls())
setwd("/Users/maraudersmap/Documents/Machine-Learning-in-R/PCA")
library(dplyr)
library(ggplot2)

#Read in Data
data_hw = read.table('heightWeightData.txt', header = FALSE)

#rename Column Names
colnames(data_hw) <- c('gender', 'height','weight')


#Extract height/weight data corresponding to males
data_hwm <- data_hw %>% filter(gender == 1) %>% dplyr::select(height,weight)

#get mean and sigma values from data
hwm_mean <- apply(data_hwm, 2, mean)
hwm_sigma <- var(data_hwm)

library(MASS)

#generate random normal data from given mean and sigma
hw_bvn <- mvrnorm(210, hwm_mean, hwm_sigma)

#Use GGplot to get plot with sample data and ellipses based on generated normal data
ggplot(data = data_hwm,
       aes(x=height, y = weight)) +
  ggtitle('Heigh vs Weight')+
  geom_text(aes(label = rownames(data_hwm)))+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .05)+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .10)+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .40)+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .60)+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .80)+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .90)+
  stat_ellipse(data = data.frame(hw_bvn),
               level = .95)
```



The original data and Gaussian ellipses are centered around the mean of our data.

(b) Standardizing

```
data_hwm_st <- data_hwm %>%
  mutate(height_st = (height - hwm_mean[1])/sqrt(hwm_sigma[1])) %>%
  mutate(weight_st = (weight - hwm_mean[2])/sqrt(hwm_sigma[2,2])) %>%
  dplyr::select(height_st, weight_st)

#Once again get mean and variance of scaled data (should be 0 and 1)
hwm_mean_st <- apply(data_hwm_st, 2, mean)
hwm_sigma_st <- var(data_hwm_st)
hw_bvn_st <- mvrnorm(210, hwm_mean_st, hwm_sigma_st)

ggplot(data = data_hwm_st,
         aes(x=height_st, y = weight_st)) +
    ggtitle('Heigh vs Weight Standardized')+
    geom_text(aes(label = rownames(data_hwm_st)))+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .05)+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .10)+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .40)+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .60)+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .80)+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .90)+
    stat_ellipse(data = data.frame(hw_bvn_st),
                 level = .95)
```



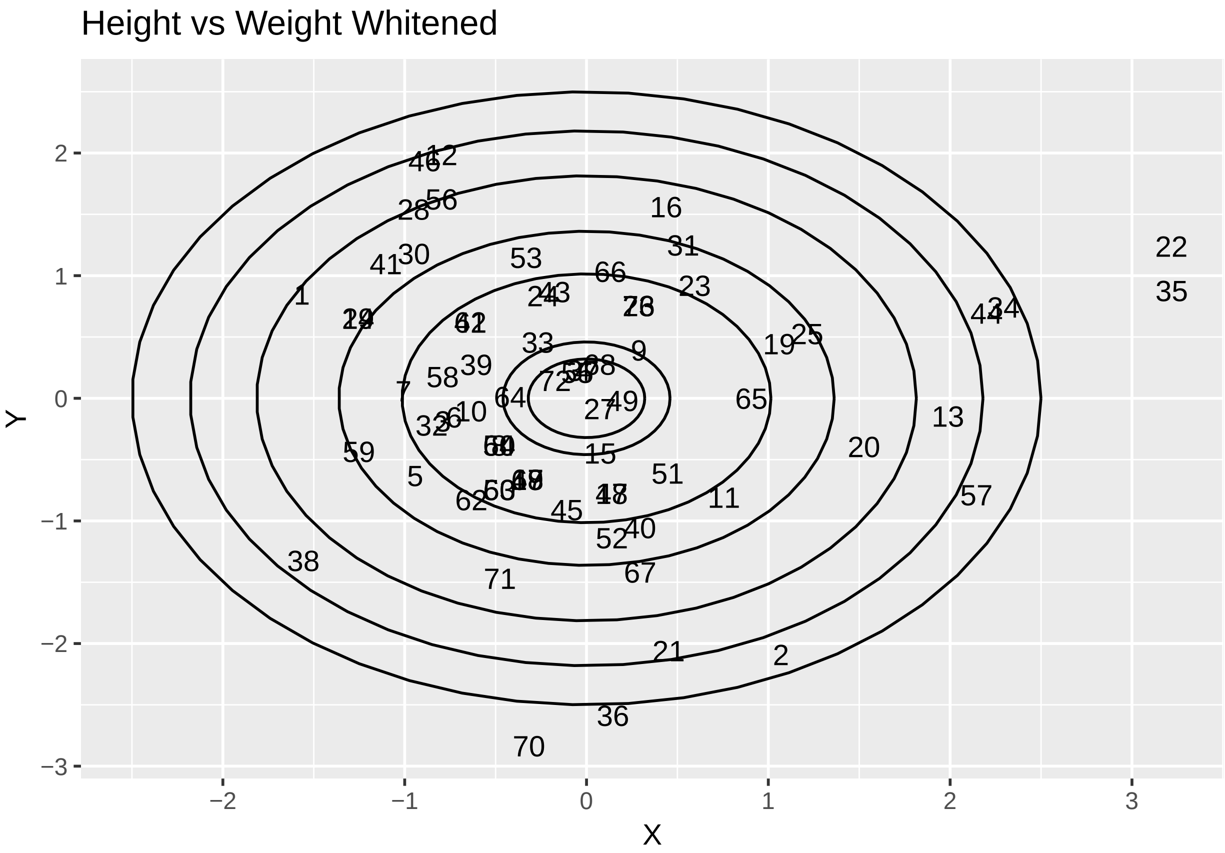
Our standardized data and ellipses are centered at 0.

(c) Whitening

Whitening data to ensure its empirical covariance matrix is proportional to the identiy matrix. Thus the data is uncorrelated and of equal variance along each dimension.

```
# Center data
data_hwms <- scale(data_hwm, scale = FALSE)

hwms_mean <- apply(data_hwms,2,mean)
hwms_sigma <- var(data_hwms)

eg <- eigen(hwms_sigma, symmetric = TRUE)
```

```r
#Get components
U <- eg$vectors
A <- eg$values

A <- A^(-1/2)

#Uncorrelate height and weight while creating variance 1

hwm_whiten <- A *t(U) %*%t(data_hwms)
hwm_whiten_m <- (t(hwm_whiten))
colnames(hwm_whiten_m) <- c('Height', 'Weight')
hwm_whiten <- as.data.frame(hwm_whiten_m)
colnames(hwm_whiten) <- c('X', 'Y')


ggplot(data = hwm_whiten,
       aes(x=X, y = Y)) +
  ggtitle('Height vs Weight Whitened')+
  geom_text(aes(label = rownames(hwm_whiten)))+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .05)+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .10)+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .40)+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .60)+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .80)+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .90)+
  stat_ellipse(data = data.frame(hwm_whiten),
               type = 'norm',
               level = .95)
```



Our whitened data has circular ellipses.