Homework 6

Joe Diaz

10/1/2019

Question 9.1

Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA. (Note that to first scale the data, you can include scale. = TRUE to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

Let's read in the table and run pca.

```
df <- read.table("uscrime.txt", stringsAsFactors = FALSE, header = TRUE)
pca <- prcomp(df[,1:15], scale. = TRUE)

components <- pca$x[,1:4] # store key columns as a dataframe
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.56729 0.55444 0.48493 0.44708 0.41915 0.35804
## Proportion of Variance 0.02145 0.02049 0.01568 0.01333 0.01171 0.00855
## Cumulative Proportion  0.92142 0.94191 0.95759 0.97091 0.98263 0.99117
##                           PC13   PC14    PC15
## Standard deviation     0.26333 0.2418 0.06793
## Proportion of Variance 0.00462 0.0039 0.00031
## Cumulative Proportion  0.99579 0.9997 1.00000
```

build linear model on pca

```
crime_components <- cbind(components, df[,16])
model <- lm(V5~., data = as.data.frame(crime_components))
#summary(model)

intercept <- model$coefficients[1]
coeff <- model$coefficients[2:5]
```

reverse pca

```
alphas <- pca$rotation[,1:4] %*% coeff


origAlpha <- alphas/sapply(df[,1:15],sd)

m <- sapply(df[,1:15],mean)
std <- sapply(df[,1:15],sd)
```

```
intercept - sum(coeff*m/std)
```

```
## Warning in coeff * m: longer object length is not a multiple of shorter
## object length
```

```
## (Intercept)
##   -760.9287
```

```
origIntercept<- sum(coeff*sapply(df[,1:15],mean)/sapply(df[,1:15],sd))
```

```
## Warning in coeff * sapply(df[, 1:15], mean): longer object length is not a
## multiple of shorter object length
```

generate predictions on original alphas and betas

```
preds <- as.matrix(df[,1:15]) %*% origAlpha + origIntercept



sse = sum((preds - df[,16])^2)
tot_ss = sum((df[,16] - mean(df[,16]))^2)



r2 <- 1 - sse/tot_ss

r2
```

```
## [1] 0.3091106
```

achieved r2 of .31