The *R* library `MASS` includes the dataframe `Cars93`. It is a selection of 93 car models from the Consumer Reports (1993). It includes 26 variables, some of which are categorical. We consider predicting the city mileage of a car, `MPG.city`, based on the number of revolutions per minute at maximum horsepower, `RPM`, and the weight of the car, `Weight`. Consider the following steps.

1. Plot `MPG.city` versus `Weight`. Add the fitted line identifying outliers.

2. Fit a simple regression model `MPG.city` versus `Weight`. What is the Adj-R^2?

3. Fit a multiple regression model `MPG.city` versus `Weight` and `RPM`. What is the Adj-R^2?

4. Transform `RPM` into a factor. What is the base level?

5. Again Fit a multiple regression model `MPG.city` versus `Weight` and `RPM`. What is the Adj-R^2?

6. Use the coefficients Table to identify non-significant levels of factor `RPM`.

7. Add non-significant levels to the base level

8. Again Fit a multiple regression model `MPG.city` versus `Weight` and `RPM`. What is the Adj-R^2?

9. Plot `MPG.city` versus `Weight`. Add the fitted lines for each category identifying outliers.

```
# car.r
library(MASS)
d0 = Cars93
d1 = subset(d0,select=c("MPG.city","RPM","Weight"))

# scatterplot
plot(MPG.city~Weight,d1,pch=19,cex=0.6)
grid()

# model vs Weight
#=====================================================================
m0=lm(MPG.city~Weight,d1)
summary(m0)
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) 47.048353   1.679912   28.01   <2e-16 ***
# Weight      -0.008032   0.000537  -14.96   <2e-16 ***
# Residual standard error: 3.038 on 91 degrees of freedom
# Multiple R-squared:  0.7109,    Adjusted R-squared:  0.7077
# F-statistic: 223.8 on 1 and 91 DF,  p-value: < 2.2e-16

plot(MPG.city~Weight,d1,pch=19,cex=0.6)
text(MPG.city~Weight,d1,labels=rownames(d1),pos=1,cex=0.6)
abline(m1,lty=2)
grid()

# 39,42,83 outliers
d0[c(39,42,83),]

# model vs RPM+Weight
#=====================================================================
m1=lm(MPG.city~RPM+Weight,d0)
summary(m1)
#Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
#(Intercept)  4.688e+01  4.254e+00  11.020   <2e-16 ***
#RPM          2.582e-05  5.906e-04   0.044    0.965
#Weight      -8.021e-03  5.974e-04 -13.426   <2e-16 ***
#Residual standard error: 3.055 on 90 degrees of freedom
#Multiple R-squared:  0.7109,    Adjusted R-squared:  0.7045
#F-statistic: 110.6 on 2 and 90 DF,  p-value: < 2.2e-16

anova(m1)
#Analysis of Variance Table
#          Df  Sum Sq Mean Sq F value    Pr(>F)
#RPM        1  382.96  382.96   41.03 6.687e-09 ***
#Weight     1 1682.58 1682.58  180.27 < 2.2e-16 ***
#Residuals 90  840.03    9.33

# Adjusted R2 same as model with no RPM
# RPM p-values in Coeff Table, ANOVA Table
# Is RPM required in model?
```

```
# RPM as factor
#============================================================
d2 = d0
d2$RPM = as.factor(d2$RPM)
m2=lm(MPG.city~RPM+Weight,d2)
summary(m2)

# Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
#(Intercept) 47.0412933  2.8621954  16.435  < 2e-16 ***
#RPM4000      0.0342904  2.7698998   0.012 0.990159
#RPM4100     -2.9223233  3.1880935  -0.917 0.362573
#RPM4200     -0.7249827  2.6034637  -0.278 0.781498
#RPM4400     -1.3397479  3.1883602  -0.420 0.675664
#RPM4500      0.7186849  3.1926716   0.225 0.822573
#RPM4600     -1.5487233  2.5236976  -0.614 0.541479
#RPM4800     -0.9590356  2.3407744  -0.410 0.683307
#RPM5000     -1.0926181  2.3804357  -0.459 0.647699
#RPM5100     -4.3596932  3.2058604  -1.360 0.178349
#RPM5200     -1.7374400  2.3966732  -0.725 0.470977
#RPM5300      0.2620712  3.1884275   0.082 0.934734
#RPM5400     -0.3257535  2.5468986  -0.128 0.898604
#RPM5500     -1.3766630  2.4127084  -0.571 0.570160
#RPM5550     -3.2921644  3.2531383  -1.012 0.315127
#RPM5600      1.2049205  2.4703716   0.488 0.627297
#RPM5700      7.6789698  2.7990959   2.743 0.007768 **
#RPM5750     -5.0104987  3.2380632  -1.547 0.126415
#RPM5800     -2.6969918  2.5358764  -1.064 0.291302
#RPM5900     13.2127342  3.2469691   4.069 0.000125 ***
#RPM6000     -0.5621574  2.3544584  -0.239 0.812008
#RPM6200     -1.8352000  3.1930724  -0.575 0.567361
#RPM6300     -1.0297425  3.2185995  -0.320 0.749999
#RPM6500     -5.9714850  2.7955638  -2.136 0.036278 *
#Weight      -0.0077677  0.0004885 -15.900  < 2e-16 ***
#Residual standard error: 2.254 on 68 degrees of freedom
#Multiple R-squared:  0.8811,    Adjusted R-squared:  0.8391
#F-statistic: 20.99 on 24 and 68 DF,  p-value: < 2.2e-16

anova(m2)
# Analysis of Variance Table
#           Df  Sum Sq Mean Sq F value    Pr(>F)
# RPM       23 1275.19   55.44   10.91 6.716e-15 ***
# Weight     1 1284.81 1284.81  252.82 < 2.2e-16 ***
# Residuals 68  345.57    5.08

# Adjusted R2 improved
# some RPM levels non-significant
```

```
# add nonsig levels to base level
#================================================================

d3 = d2
d2$RPM
#  [1] 6300 5500 5500 5500 5700 5200 4800 4000 4800 4100 6000 5200 5200 4600 5200 4800 4000
# [18] 4200 5000 5300 5000 4800 6000 4800 4800 5000 4800 6000 6000 5800 5000 6500 4200 4600
# [35] 5500 4800 4800 4200 5700 5400 5800 5900 5600 5500 6000 5550 6000 6000 5200 6000 4400
# [52] 4600 5000 5500 5600 5000 6500 5100 5500 5750 3800 6000 6000 6000 5600 4800 5200 6000
# [69] 5200 4800 4800 5000 5600 5200 4600 5000 4800 6000 5000 5600 5200 5600 6000 5200 5400
# [86] 5400 5000 5500 4500 5800 5800 5400 6200
# 24 Levels: 3800 4000 4100 4200 4400 4500 4600 4800 5000 5100 5200 5300 5400 5500 ... 6500

table(d2$RPM)
# 3800 4000 4100 4200 4400 4500 4600 4800 5000 5100 5200 5300
#    1    2    1    3    1    1    4   13   10    1   10    1
# 5400 5500 5550 5600 5700 5750 5800 5900 6000 6200 6300 6500
#    4    8    1    6    2    1    4    1   14    1    1    2

# keep levels 5700, 5900, 6500
levels(d2$RPM)[c(17,20,24)]
# [1] "5700" "5900" "6500"

# set all other levels to 0
levels(d3$RPM)[-c(17,20,24)]="0"
d3$RPM
#  [1] 0    0    0    0    5700 0    0    0    0    0    0    0    0    0    0    0    0
# [18] 0    0    0    0    0    0    0    0    0    0    0    0    0    0    6500 0    0
# [35] 0    0    0    0    5700 0    0    5900 0    0    0    0    0    0    0    0    0
# [52] 0    0    0    0    0    6500 0    0    0    0    0    0    0    0    0    0    0
# [69] 0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
# [86] 0    0    0    0    0    0    0    0
# Levels: 0 5700 5900 6500


# model vs Weight+RPM(as a factor)
#================================================================

m3=lm(MPG.city~RPM+Weight,d3)
summary(m3)

# Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept) 45.4630971  1.2738638  35.689  < 2e-16 ***
# RPM5700      8.7948426  1.6131799   5.452 4.50e-07 ***
# RPM5900     14.3836351  2.2755915   6.321 1.04e-08 ***
# RPM6500     -4.8634115  1.6113206  -3.018  0.00333 **
# Weight      -0.0075944  0.0004038 -18.807  < 2e-16 ***
# Residual standard error: 2.243 on 88 degrees of freedom
# Multiple R-squared:  0.8477,    Adjusted R-squared:  0.8407
# F-statistic: 122.4 on 4 and 88 DF,  p-value: < 2.2e-16
```

```
anova(m3)
# Analysis of Variance Table
#           Df  Sum Sq Mean Sq F value    Pr(>F)
# RPM        3  683.98  227.99  45.328 < 2.2e-16 ***
# Weight     1 1778.97 1778.97 353.686 < 2.2e-16 ***
# Residuals 88  442.62    5.03

# Adjusted R2 about the same, but more simple model


# plot explaining the outliers
#=====================================================================

par(mfrow=c(1,1))

a = which(d3$RPM %in% c("5700","5900","6500"))
# [1]  5 32 39 42 57

aux0=rownames(d3[d3$RPM!="0",])  # "5"  "32" "39" "42" "57"
aux0=as.numeric(aux0)

# color for each point
aux  = as.numeric(d3$RPM)
# [1] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#[32] 4 1 1 1 1 1 1 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1
#[63] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

plot(MPG.city~Weight,d3,col=aux,pch=aux,cex=0.88)
text(MPG.city~Weight,d3,labels=ifelse(d3$RPM!="0",rownames(d3),""),pos=1,offset=0.5,cex=0.4,col=au

coef(m3)
#  (Intercept)      RPM5700      RPM5900      RPM6500       Weight
# 45.463097132  8.794842578 14.383635134 -4.863411485 -0.007594354

abline(m3$coef[1],m3$coef[5],col=1,lty=1,lwd=1.4)
abline(m3$coef[1]+m3$coef[2],m3$coef[5],col=2,lty=2,lwd=1.4)
abline(m3$coef[1]+m3$coef[3],m3$coef[5],col=3,lty=3,lwd=1.4)
abline(m3$coef[1]+m3$coef[4],m3$coef[5],col=4,lty=4,lwd=1.4)
grid()
legend("topright",c("0","5700","5900","6500"),lty=1:4,cex=0.6,col=c(1,2,3,4))

# originally  39,42,83 outliers
# actually 83 is the only outlier
# 5,39 belong to other population (RPM = 3700)
# 32,57 belong to RPM = 6500
# 42 belongs to RPM = 5900

# add common LS line
coef(m0)
#  (Intercept)       Weight
# 47.048353174 -0.008032392
abline(47.04835,-0.008,col="purple")
```

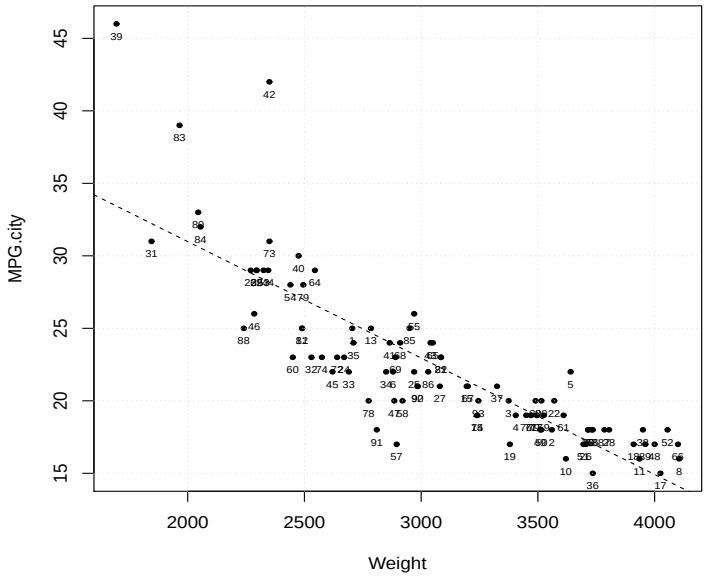

Figure 1: Scatterplot MPG.city versus Weight, with SLR fitted line

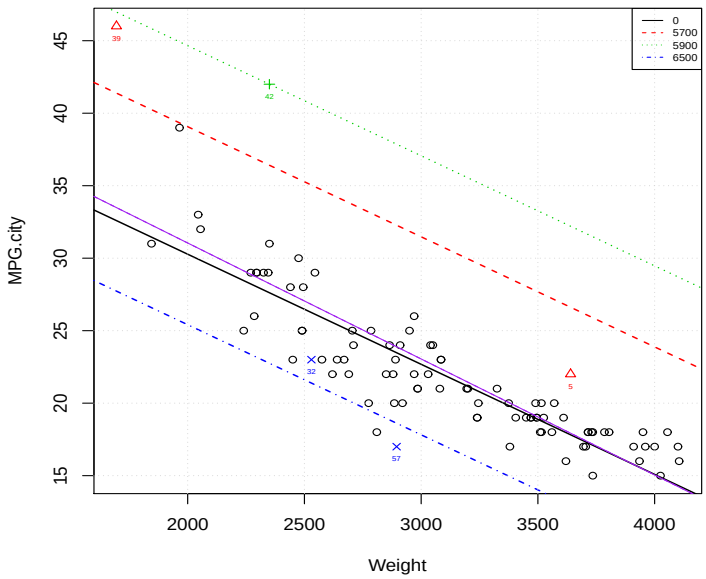

Figure 2: Scatterplot MPG.city versus Weight, with fitted lines for RPM categories