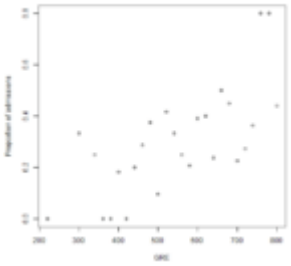
# Logistic Regression Exercises



In the exercises below we cover some material on logistic regression in R. Logistic regression is used when our response variable is binary (or dichotomous), i.e., takes on two categories (usually 'yes' and 'no'), and we usually have one continuous predictor variable.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

We will be using the dataset at http://www.ats.ucla.edu/stat/data/binary.csv, which includes the variables admit (whether a student was admitted to the school), gre (the student's GRE score), gpa (the student's GPA), and rank (school's prestige; 1=highest, 4=lowest).

First, let's obtain our data:

```
admissions <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")
```

**Exercise 1**

Suppose that we would like to predict admission based on a student's GRE score, using a logistic regression model. We can check whether a logistic regression model is a good idea by plotting proportions of admissions (by value of GRE) vs. our predictor variable. Compute these proportions.

**Exercise 2**
Create the plot mentioned in Exercise 1.

**Exercise 3**
Fit the model specified in Exercise 1.

**Exercise 4**
Extract the model coefficients.

**Exercise 5**
What is the estimated increase in the odds of admission for each additional 100 points on GRE score?

**Exercise 6**
Give a 95% confidence interval for the estimate in Exercise 5. Interpret.

**Exercise 7**
Use the value of residual deviance to test the fit of the model.

**Exercise 8**
Plot the fitted logistic regression curve (i.e., probability of admission vs. GRE).

**Exercise 9**
Extract the predicted values.

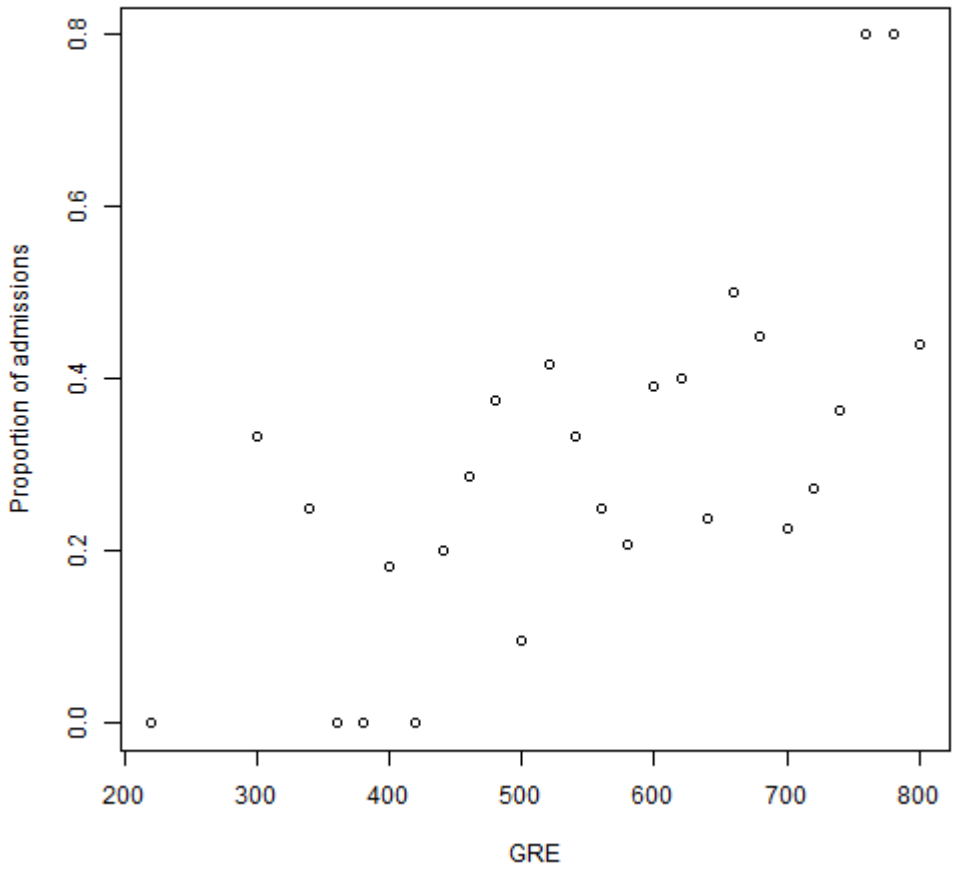
**Exercise 10**
Predict the probability of admission for a student whose GRE score is 580.

---

# Logistic Regression Solutions (part-1)

Below are the solutions to these exercises on Logistic Regression (part 1).

```
admissions <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")


####################
#                  #
#    Exercise 1    #
#                  #
####################
props <- aggregate(admit~gre,data=admissions,FUN=function(x) sum(x)/length(x))


####################
#                  #
#    Exercise 2    #
#                  #
####################
plot(props[,2] ~props[,1],xlab="GRE",ylab="Proportion of admissions")
```

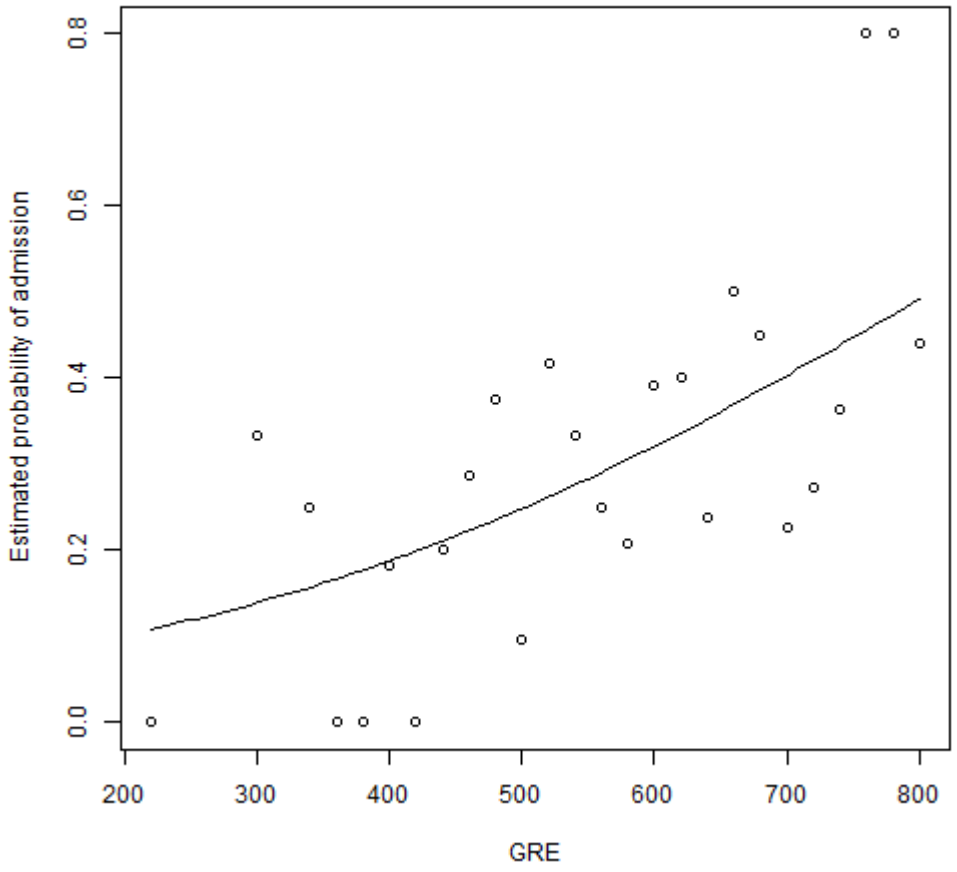



```
#We see that the pattern of proportions can potentially be a
good fit to the common "S"-shaped logistic regression curve
(except for the two outliers in the top right).

#####################
#                   #
#    Exercise 3     #
#                   #
#####################
m1 <- glm(admit~gre,data=admissions,family=binomial())
summary(m1)


##
## Call:
## glm(formula = admit ~ gre, family = binomial(), data =
admissions)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.1623  -0.9052  -0.7547   1.3486   1.9879
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.901344   0.606038  -4.787 1.69e-06 ***
## gre          0.003582   0.000986   3.633  0.00028 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 486.06  on 398  degrees of freedom
## AIC: 490.06
##
## Number of Fisher Scoring iterations: 4

####################
#                  #
#    Exercise 4    #
#                  #
####################
coefs <- coef(m1)
coefs

##  (Intercept)          gre
## -2.901344270  0.003582212

####################
#                  #
#    Exercise 5    #
#                  #
####################
exp(100*coefs[2])

##      gre
## 1.430782

#The odds of admission increase by about 43%
```

```
####################
#                  #
#    Exercise 6    #
#                  #
####################
exp(100*confint(m1)[2,])

##    2.5 %   97.5 %
## 1.182932 1.742420

#We are 95% confident that the odds of admission increase
between about 18% and 74%.

####################
#                  #
#    Exercise 7    #
#                  #
####################
anova(m1, test="Chisq")

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: admit
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                    399     499.98
## gre   1    13.92       398     486.06 0.0001907 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

#Our model is significantly better than the intercept-only
model.

####################
```

```
#                    #
#    Exercise 8      #
#                    #
#####################
plot(props[,2]        ~        props[,1],data        =
admissions,xlab="GRE",ylab="Estimated   probability   of
admission")
curve(predict(m1,data.frame(gre=x),type="response"),add=TRUE)
```




```
#####################
#                    #
#    Exercise 9      #
#                    #
#####################
predicted <- predict(m1, type="response")


#####################
#                    #
#    Exercise 10     #
```

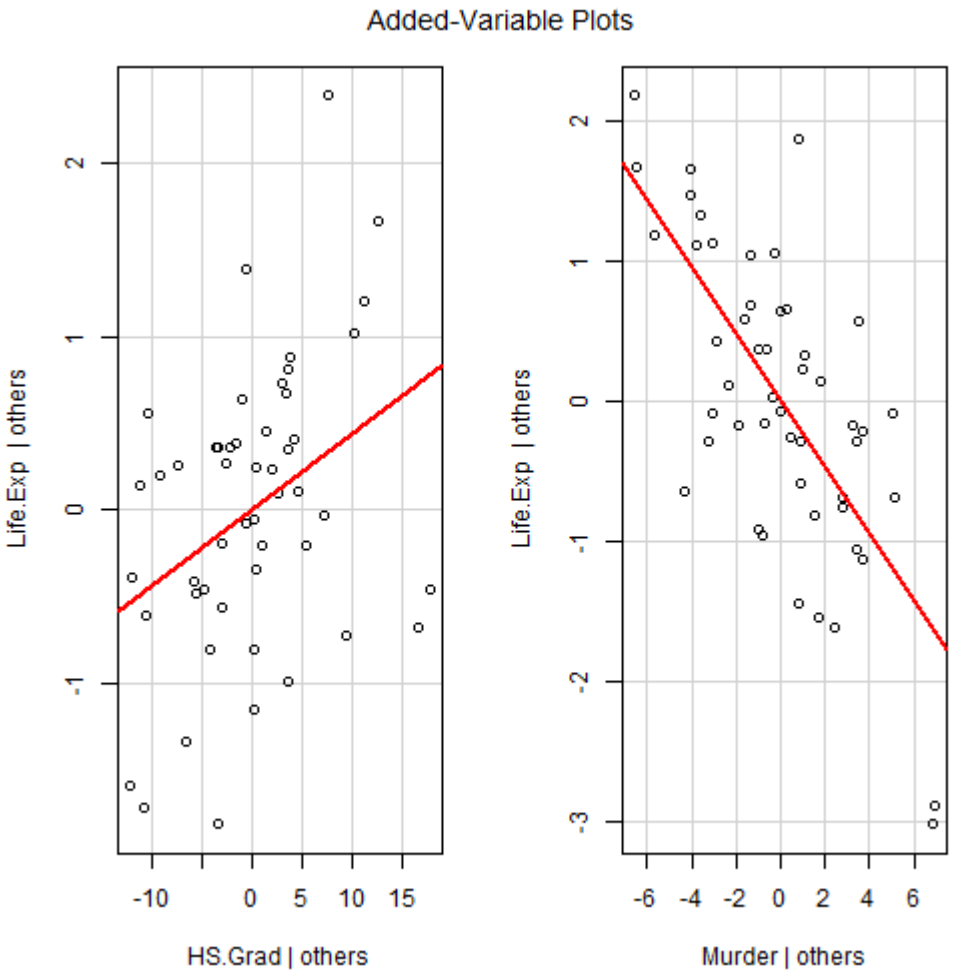
```
#                   #
#####################
predict(m1, newdata = data.frame(gre=580), type = "response")

##        1
## 0.304987
```

---

# Multiple Regression (Part 3) Diagnostics – Solutions

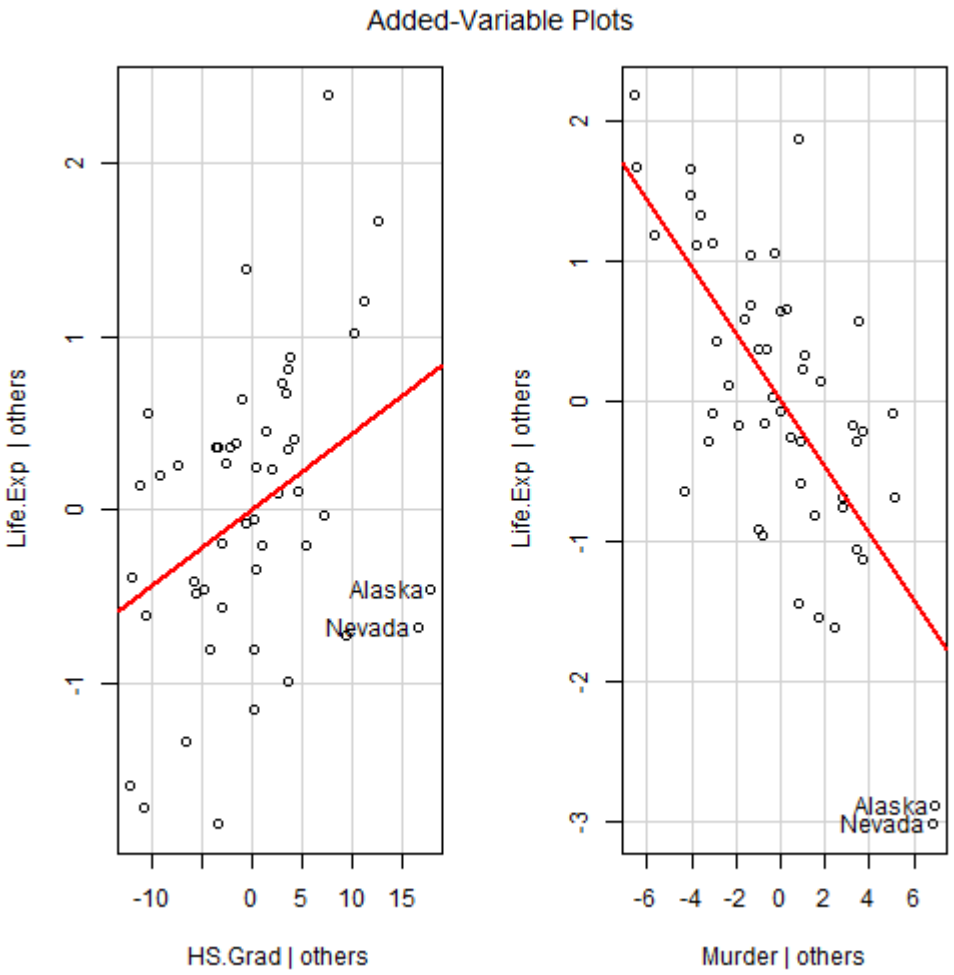
Below are the solutions to these exercises on Multiple Regression (part 3).

```
data(state)
state77 <- as.data.frame(state.x77)
names(state77)[4] <- "Life.Exp"
names(state77)[6] <- "HS.Grad"


#####################
#                   #
#    Exercise 1     #
#                   #
#####################
#a.
library(car)
m1 <- lm(Life.Exp ~ HS.Grad+Murder, data=state77)
avPlots(m1)
```

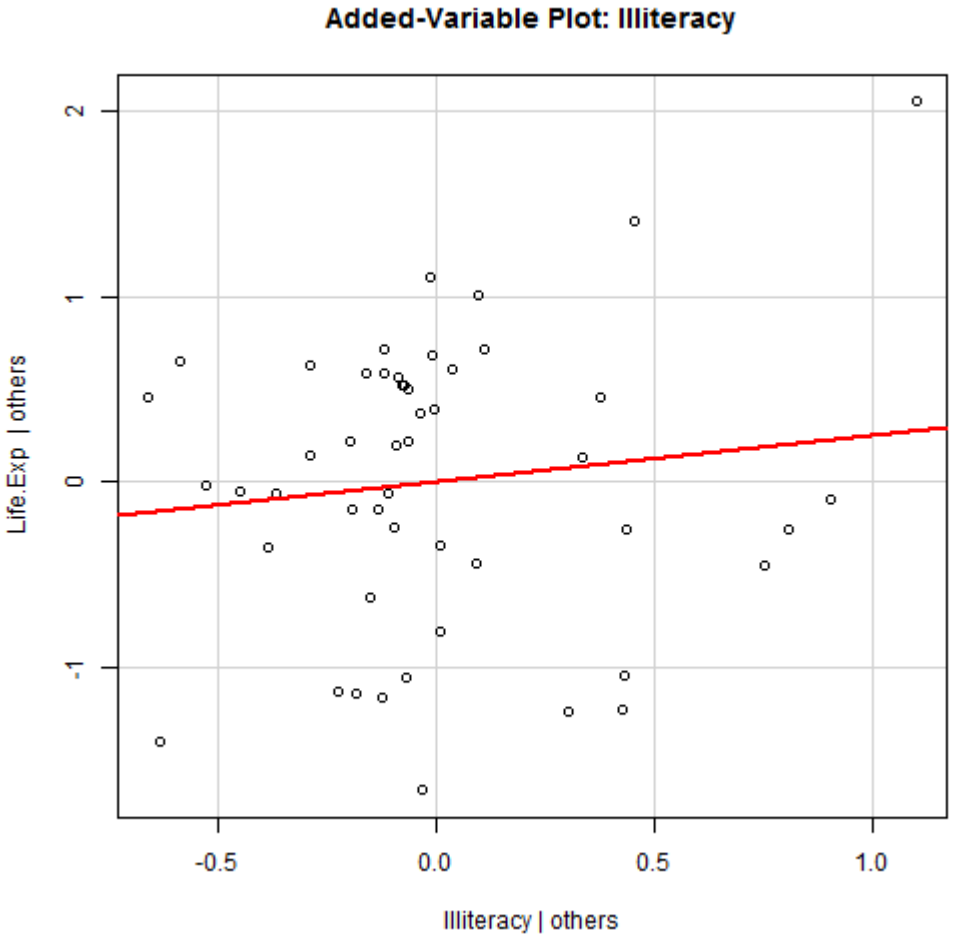



#Note that the slope of the line is positive in the HS.Grad plot, and negative in the Murder plot, as expected.

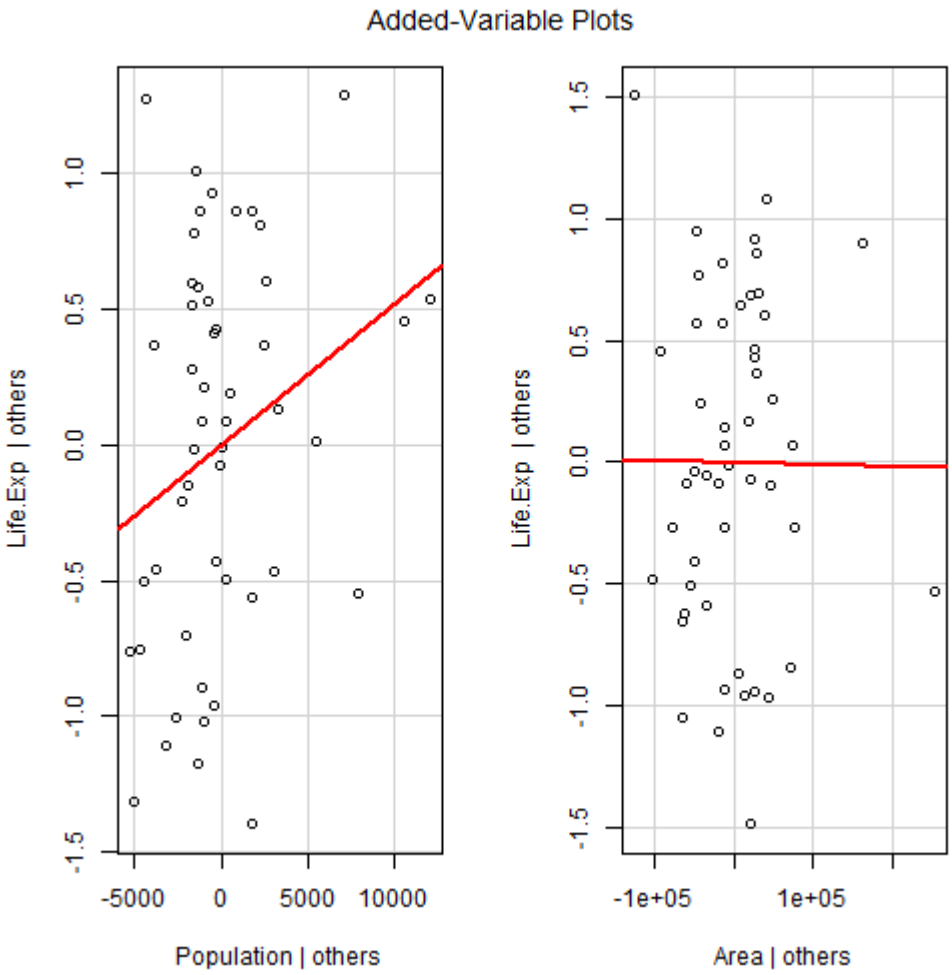
#b.
avPlots(m1,id.method=list("mahal"),id.n=2)



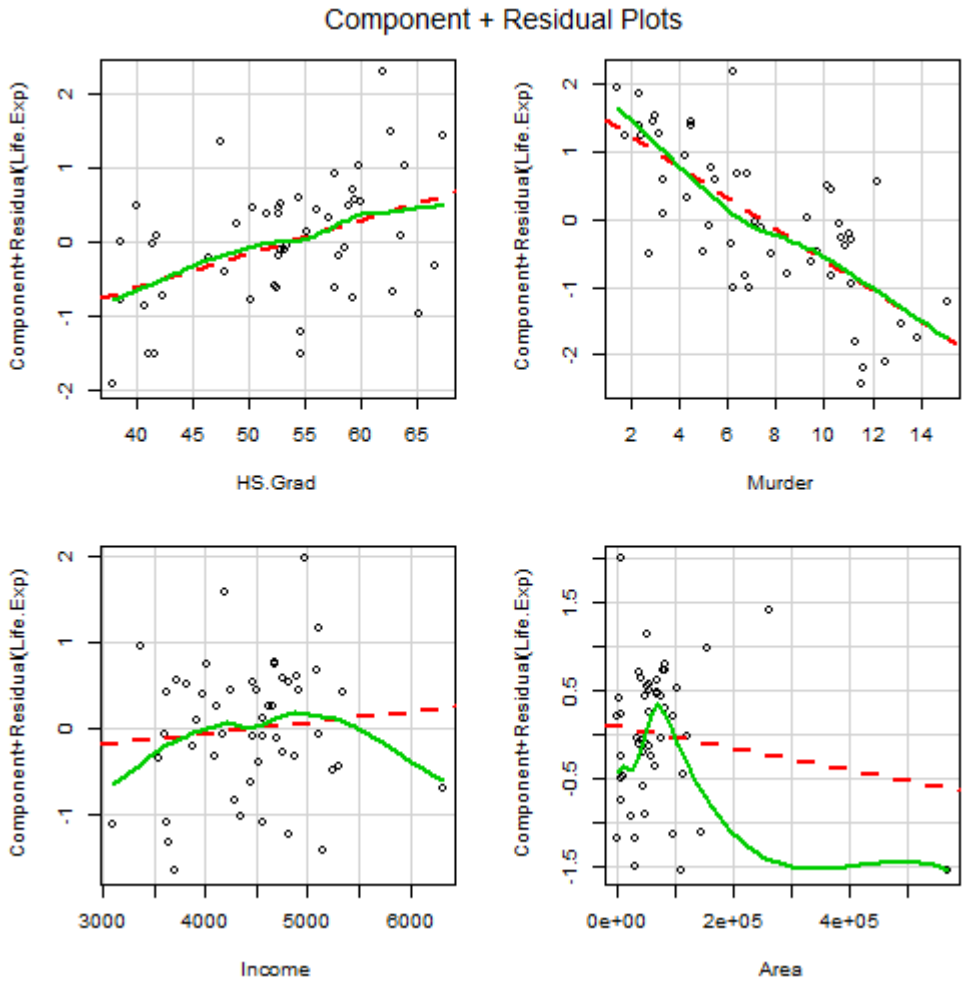

```
#####################
#                   #
#    Exercise 2     #
#                   #
#####################
#a.
with(state77,avPlot(lm(Life.Exp                          ~
HS.Grad+Murder+Illiteracy),variable=Illiteracy))
```



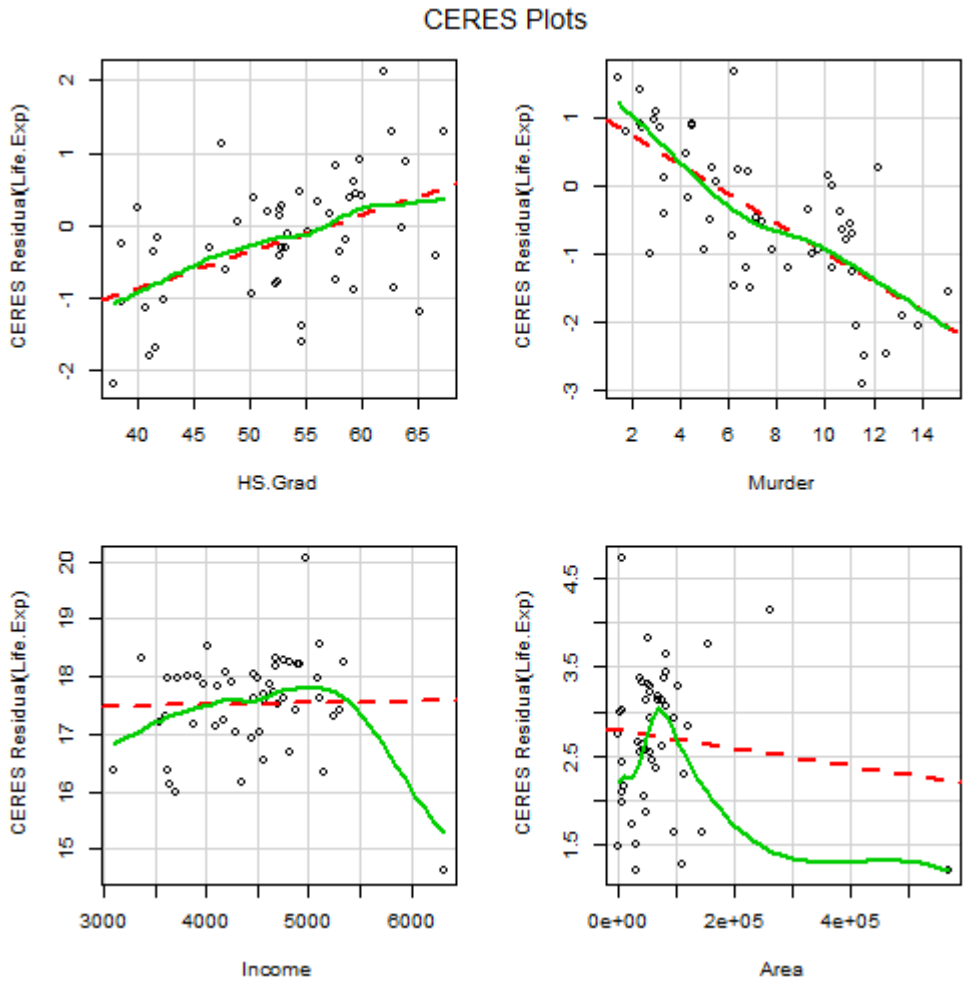

```
#Note that the slope is positive, contrary to what is expected

#b.
avPlots(lm(Life.Exp ~ .,data=state77), terms= ~ Population+Area)
```




```
#####################
#                   #
#    Exercise 3     #
#                   #
#####################
crPlots(lm(Life.Exp                                    ~
HS.Grad+Murder+Income+Area,data=state77))
```




```
#We see that there seems to be a problem with linearity for
Income and Area (which could be due to the outlier in the
lower right corner in both plots).


#####################
#                   #
#    Exercise 4     #
#                   #
#####################
ceresPlots(lm(Life.Exp                                    ~
HS.Grad+Murder+Income+Area,data=state77))
```




```
#Here, there is not much difference with the plots in Exercise 3 (although, in general, CERES plots are "less prone to leakage of nonlinearity among the predictors.")


#####################
#                   #
#    Exercise 5     #
#                   #
#####################
vif(lm(Life.Exp ~ .,data=state77))

## Population     Income Illiteracy     Murder    HS.Grad
Frost
##   1.499915   1.992680   4.403151   2.616472   3.134887
2.358206
##       Area
##   1.789764
```

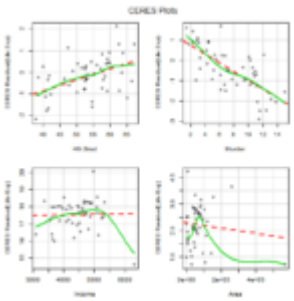
```
#Some authors advocate that a vif>2.5 is a cause for concern, while others mention vif>4 or vif>10. According to these criteria, Illiteracy, Murder, and HS.Grad are the most problematic (in the presence of all the other predictors).


####################
#                  #
#    Exercise 6    #
#                  #
####################
library(lmtest)
bptest(m1)

##
##      studentized Breusch-Pagan test
##
## data:  m1
## BP = 2.9728, df = 2, p-value = 0.2262

#There is no evidence of heteroscedasticity (of the type that depends on a linear combination of the predictors).


####################
#                  #
#    Exercise 7    #
#                  #
####################
ncvTest(m1)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.01065067    Df = 1     p = 0.9178026

#Note that the results are different to Exercise 6 because bptest (by default) uses studentized residuals (which is preferred for robustness) and assumes the error variance depends on a linear combination of the predictors, whereas ncvTest (by default) uses regular residuals and assumes the
```

```
error variance depends on the fitted values.
#ncvTest(m1) is equivalent to bptest(m1,varformula= ~
m1$fitted,studentize=F,data=state77)


####################
#                  #
#    Exercise 8    #
#                  #
####################
bptest(m1,varformula=                                ~
I(HS.Grad^2)+I(Murder^2)+HS.Grad*Murder,data=state77)

##
##      studentized Breusch-Pagan test
##
## data:  m1
## BP = 6.7384, df = 5, p-value = 0.2408

####################
#                  #
#    Exercise 9    #
#                  #
####################
#a.
ks.test(m1$residuals,"pnorm")

##
##      One-sample Kolmogorov-Smirnov test
##
## data:  m1$residuals
## D = 0.15546, p-value = 0.1603
## alternative hypothesis: two-sided

#There is no evidence that the residuals are not Normal.

#b.
shapiro.test(m1$residuals)

##
```

```
##      Shapiro-Wilk normality test
##
## data:  m1$residuals
## W = 0.96961, p-value = 0.2231

#Again, there is no evidence of nonnormality.


####################
#                  #
#    Exercise 10   #
#                  #
####################
durbinWatsonTest(m1)

##  lag Autocorrelation D-W Statistic p-value
##    1      0.04919151        1.8495   0.582
##  Alternative hypothesis: rho != 0

#There is no evidence of lag-1 autocorrelation in the
residuals.
```

---

# Multiple Regression (Part 3) Diagnostics



In the exercises below we cover some more material on multiple regression diagnostics in R. This includes added variable

(partial-regression) plots, component+residual (partial-residual) plots, CERES plots, VIF values, tests for heteroscedasticity (nonconstant variance), tests for Normality, and a test for autocorrelation of residuals. These are perhaps not as common as what we have seen in Multiple Regression (Part 2), but their aid in investigating our model's assumptions is valuable.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Multiple Regression (Part 2) Diagnostics can be found here.

As usual, we will be using the dataset state.x77, which is part of the state datasets available in R. (Additional information about the dataset can be obtained by running help(state.x77).)

First, please run the following code to obtain and format the data as usual:

```
data(state)
state77 <- as.data.frame(state.x77)
names(state77)[4] <- "Life.Exp"
names(state77)[6] <- "HS.Grad"
```

**Exercise 1**

For the model with Life.Exp as dependent variable, and HS.Grad and Murder as predictors, suppose we would like to study the marginal effect of each predictor variable, given that the other predictor is in the model.

a. Use a function from the car package to obtain added-variable (partial regression) plots for this purpose.

b. Re-create the added-variable plots from part a., labeling the two most influential points in the plots (according to Mahalanobis distance).

**Learn more** about multiple linear regression in the online course Linear regression in R for Data Scientists. In this course you will learn how to:

- Model basic and complex real world problem using linear regression
- Understand when models are performing poorly and correct it
- Design complex models for hierarchical data
- And much more

**Exercise 2**

a. Illiteracy is highly correlated with both HS.Grad and Murder. To illustrate problems that occur when multicollinearity exists, suppose we would like to study the marginal effect of Illiteracy (only), given that HS.Grad and Murder are in the model. Use a function from the car package to get the relevant added-variable plot.

b. From the correlation matrix in the previous Exercise Set, we know that Population and Area are the least strongly correlated variables with Life.Exp. Create added-variable plots for each of these two variables, given that all other six variables are in the model.

**Exercise 3**

Consider the model with HS.Grad, Murder, Income, and Area as predictors. Create component+residual (partial-residual) plots for this model.

**Exercise 4**

Create CERES plots for the model in Exercise 3.

**Exercise 5**

As an illustration of high collinearities, compute VIF (Variance Inflation Factor) values for a model with Life.Exp as the response, that includes all the variables as predictors. Which variables seem to be causing the most

problems?

**Exercise 6**
Using a function from the package lmtest, conduct a Breusch-Pagan test for heteroscedasticity (non-constant variance) for the model in Exercise 1.

**Exercise 7**
Re-do the test in the previous exercise by using a function from the car package.

**Exercise 8**
The test in Exercise 6 (and 7) is for linear forms of heteroscedasticity. To test for nonlinear heteroscedasticity (e.g., "bowtie-shape" in a residual plot), conduct White's test.

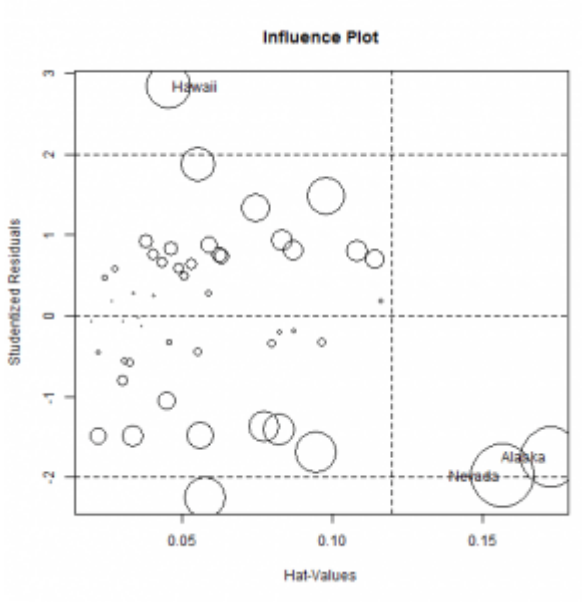
**Exercise 9**
a. Conduct the Kolmogorov-Smirnov normality test for the residuals from the model in Exercise 1.
b. Now conduct the Shapiro-Wilk normality test.
Note: More Normality tests can be found in the nortest package.

**Exercise 10**
For illustration purposes only, conduct the Durbin-Watson test for autocorrelation in residuals. (NOTE: This test is ONLY appropriate when the response variable is a time series, or somehow time-related (e.g., ordered by data collection time.))

---

# Multiple Regression (Part 2)

# – Diagnostics




Multiple Regression is one of the most widely used methods in statistical modelling. However, despite its many benefits, it is oftentimes used without checking the underlying assumptions. This can lead to results which can be misleading or even completely wrong. Therefore, applying diagnostics to detect any strong violations of the assumptions is important. In the exercises below we cover some material on multiple regression diagnostics in R.

Answers to the exercises are available here.

If you obtain a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Multiple Regression (Part 1) can be found here.

We will be using the dataset state.x77, which is part of the state datasets available in R. (Additional information about the dataset can be obtained by running help(state.x77).)

**Exercise 1**
a. Load the state datasets.
b. Convert the state.x77 dataset to a dataframe.
c. Rename the Life Exp variable to Life.Exp, and HS Grad to HS.Grad. (This avoids problems with referring to these variables when specifying a model.)
d. Produce the correlation matrix.
e. Create a scatterplot matrix for the variables Life.Exp, HS.Grad, Murder, and Frost.

**Exercise 2**

a. Fit the model with Life.Exp as dependent variable, and HS.Grad and Murder as predictors.
b. Obtain the residuals.
c. Obtain the fitted values.

**Exercise 3**

a. Create a residual plot (residuals vs. fitted values).
b. Create the same residual plot using the plot command on the lm object from Exercise 2.

**Learn more** about multiple linear regression in the online courses Linear regression in R for Data Scientists, Statistics with R – advanced level, and Linear Regression and Modeling.

**Exercise 4**

Create plots of the residuals vs. each of the predictor variables.

**Exercise 5**

a. Create a Normality plot.
b. Create the same plot using the plot command on the lm object from Exercise 2.

**Exercise 6**

a. Obtain the studentized residuals.
b. Does there appear to be any outliers?

**Exercise 7**

a. Obtain the leverage value for each observation and plot them.
b. Obtain the conventional threshold for leverage values. Are any observations influential?

**Exercise 8**

a. Obtain DFFITS values.
b. Obtain the conventional threshold. Are any observations influential?

c. Obtain DFBETAS values.
d. Obtain the conventional threshold. Are any observations influential?

**Exercise 9**
a. Obtain Cook's distance values and plot them.
b. Obtain the same plot using the plot command on the lm object from Exercise 2.
c. Obtain the threshold value. Are any observations influential?

**Exercise 10**
Create the Influence Plot using a function from the car package.

---

# Multiple Regression (Part 2) Diagnostics – Solutions

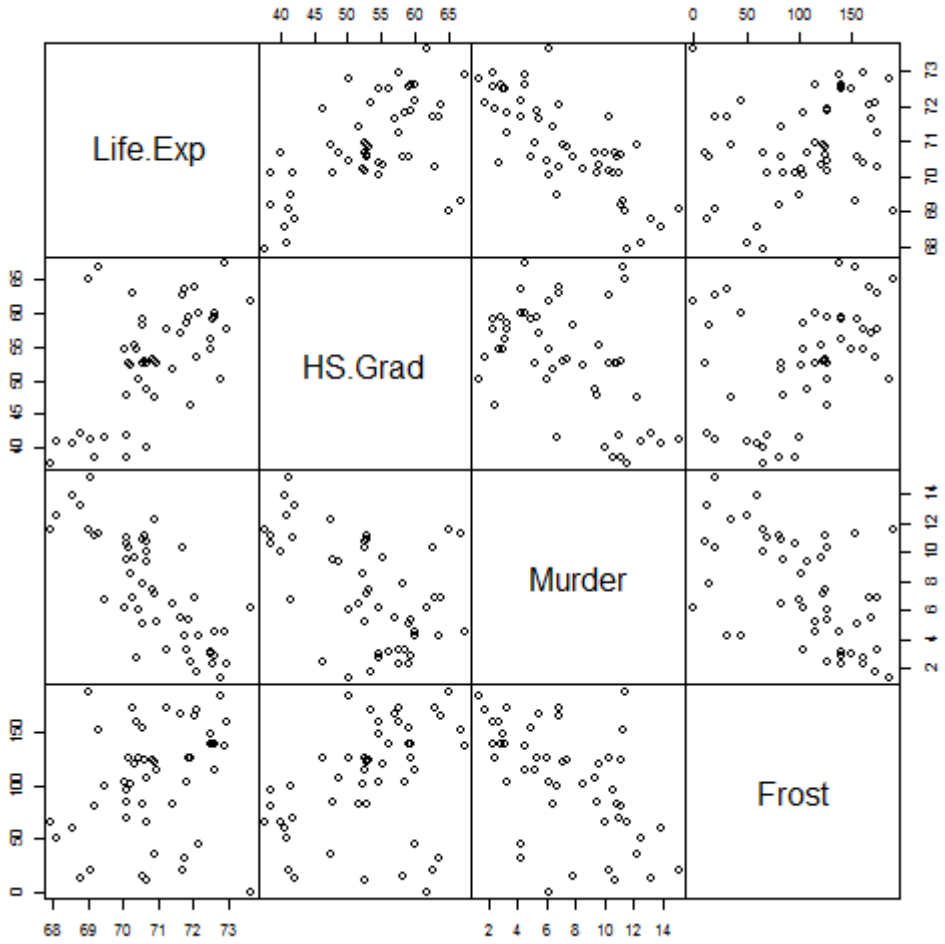
Below are the solutions to these exercises on Multiple Regression (part 2).

**Learn more** about multiple linear regression in the online courses Linear regression in R for Data Scientists, Statistics with R – advanced level, and Linear Regression and Modeling.

```
####################
#                  #
#    Exercise 1    #
#                  #
####################
#a.
data(state)
```

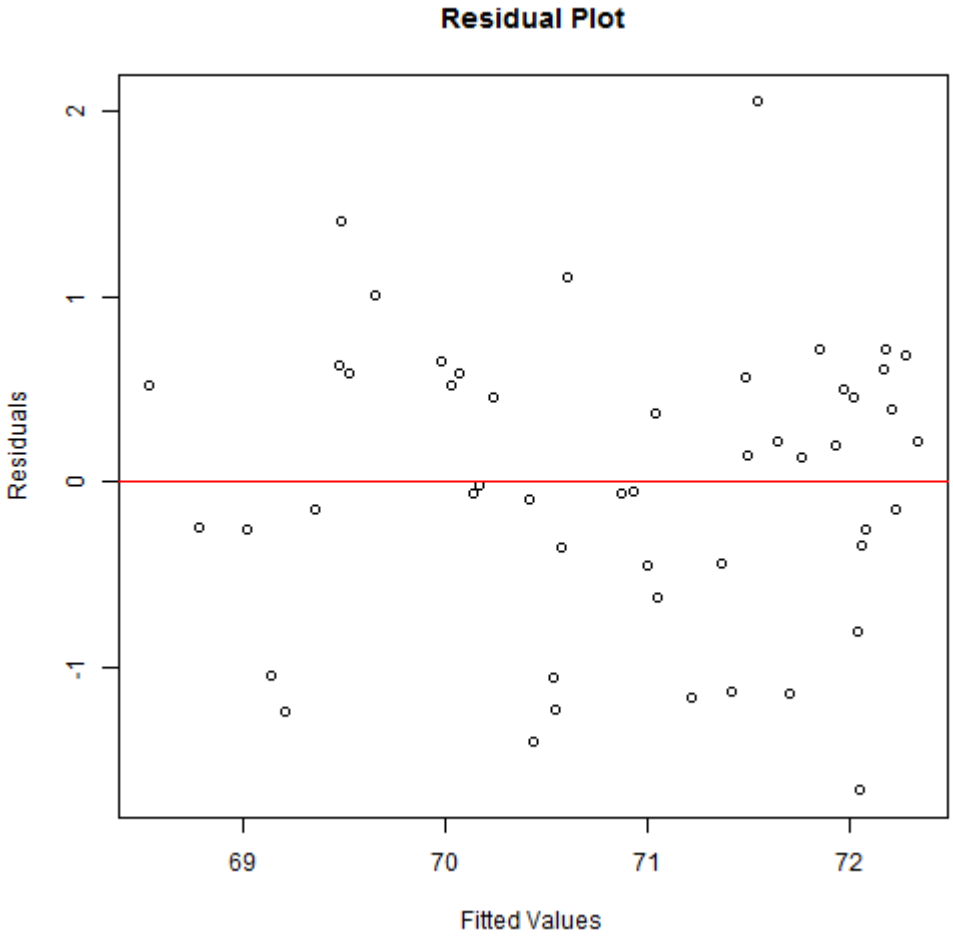
```
#b.
state77 <- as.data.frame(state.x77)

#c.
names(state77)[4] <- "Life.Exp"
names(state77)[6] <- "HS.Grad"

#d.
round(cor(state77),3) #displays correlations to 3 decimal
places

##             Population Income Illiteracy Life.Exp Murder
HS.Grad  Frost
## Population      1.000  0.208      0.108   -0.068  0.344
-0.098 -0.332
## Income          0.208  1.000     -0.437    0.340 -0.230
0.620  0.226
## Illiteracy      0.108 -0.437      1.000   -0.588  0.703
-0.657 -0.672
## Life.Exp       -0.068  0.340     -0.588    1.000 -0.781
0.582  0.262
## Murder          0.344 -0.230      0.703   -0.781  1.000
-0.488 -0.539
## HS.Grad        -0.098  0.620     -0.657    0.582 -0.488
1.000  0.367
## Frost          -0.332  0.226     -0.672    0.262 -0.539
0.367  1.000
## Area            0.023  0.363      0.077   -0.107  0.228
0.334  0.059
##              Area
## Population  0.023
## Income      0.363
## Illiteracy  0.077
## Life.Exp   -0.107
## Murder      0.228
## HS.Grad     0.334
## Frost       0.059
## Area        1.000

#e.
pairs(~ Life.Exp + HS.Grad + Murder + Frost, data=state77,
```

```
gap=)
```

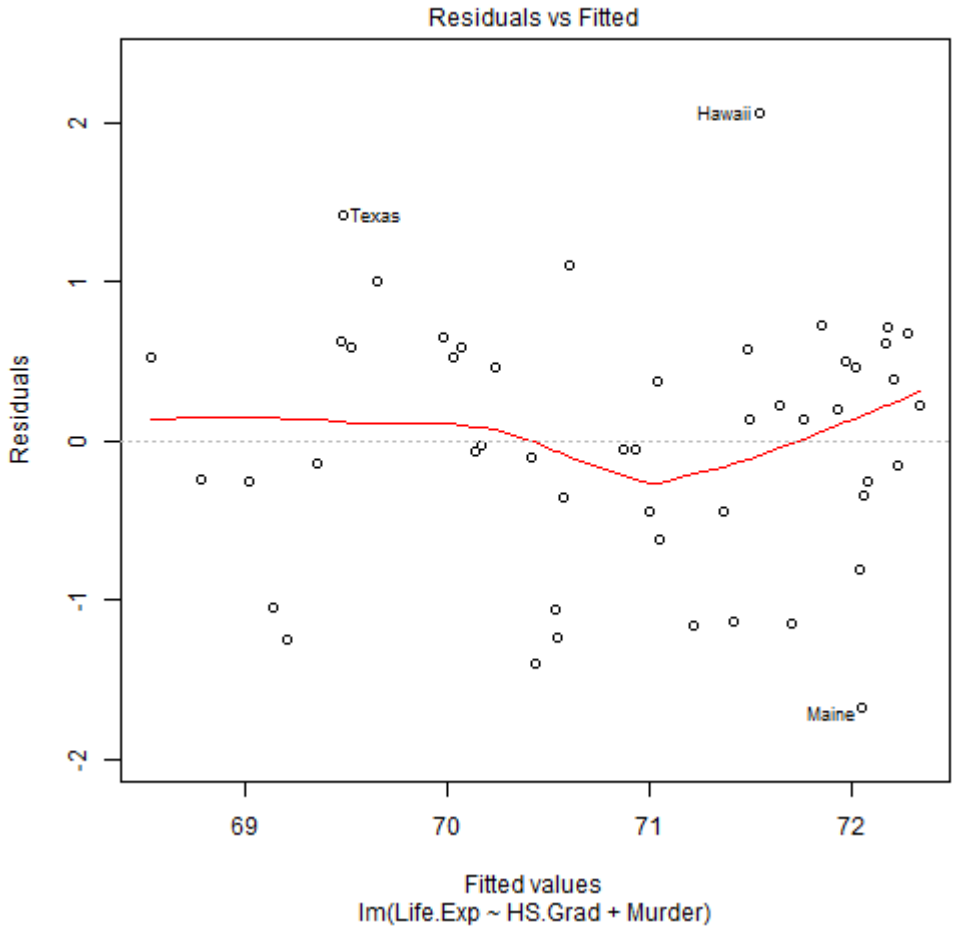



```
####################
#                  #
#    Exercise 2    #
#                  #
####################
#a.
model <- lm(Life.Exp ~ HS.Grad + Murder, data=state77)
summary(model)


##
## Call:
## lm(formula = Life.Exp ~ HS.Grad + Murder, data = state77)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66758 -0.41801  0.05602  0.55913  2.05625
##
```

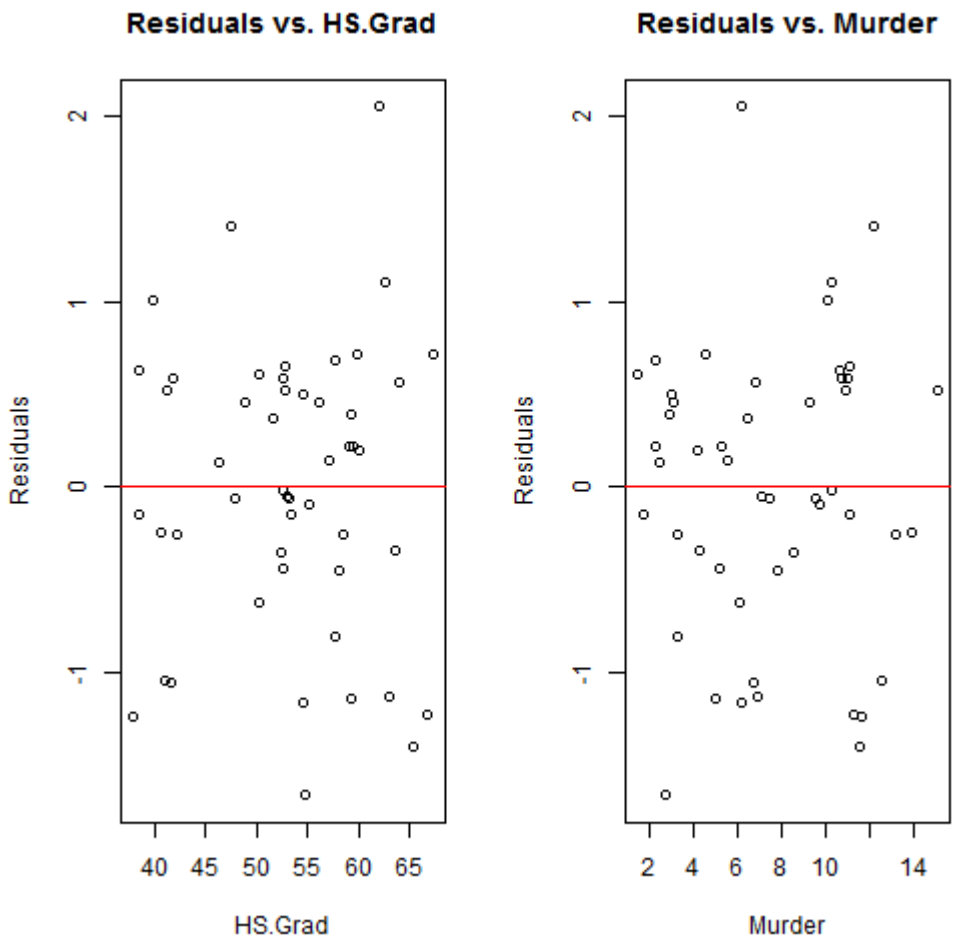
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 70.29708    1.01567  69.213  < 2e-16 ***
## HS.Grad      0.04389    0.01613   2.721  0.00909 **
## Murder      -0.23709    0.03529  -6.719 2.18e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.7959 on 47 degrees of freedom
## Multiple R-squared:  0.6628, Adjusted R-squared:  0.6485
## F-statistic:  46.2 on 2 and 47 DF,  p-value: 8.016e-12


#b.
resids <- model$residuals

#c.
fitted <- model$fitted.values

####################
#                  #
#    Exercise 3    #
#                  #
####################
#a.
plot(fitted,resids,main="Residual     Plot",xlab="Fitted
Values",ylab="Residuals")
abline(h=,col="red")
```



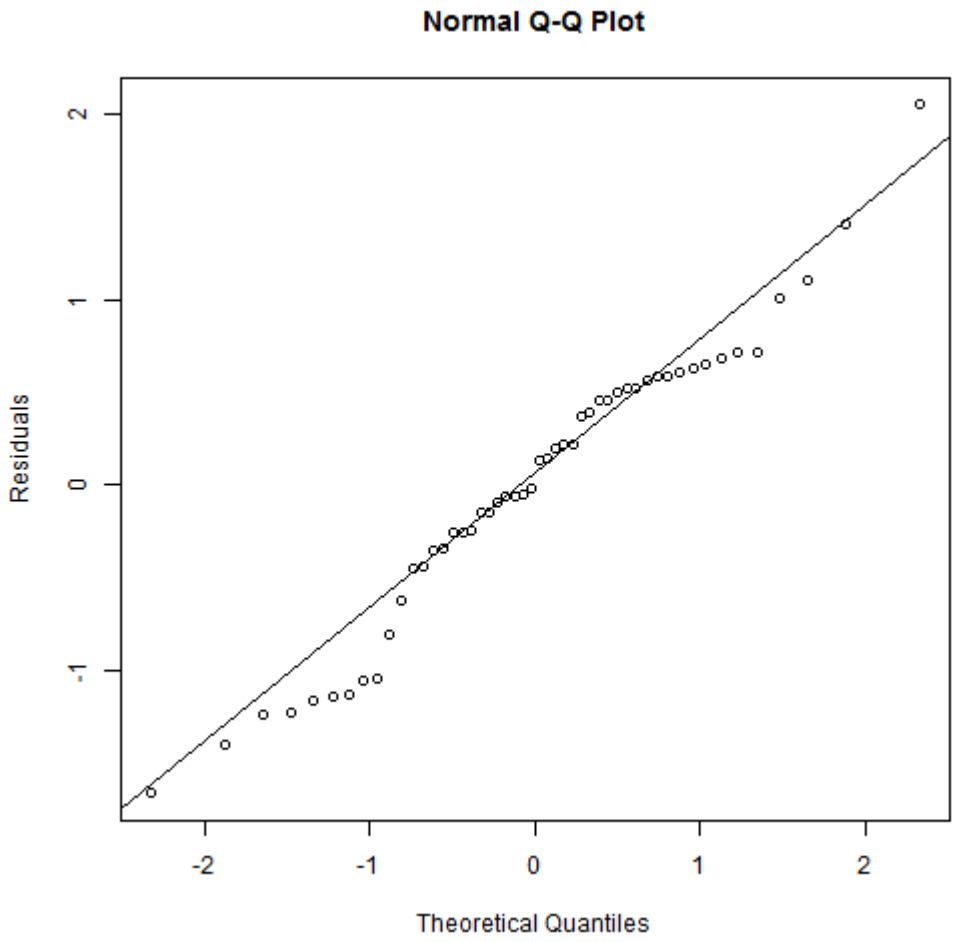

```
#b.
plot(model,which=1)
```



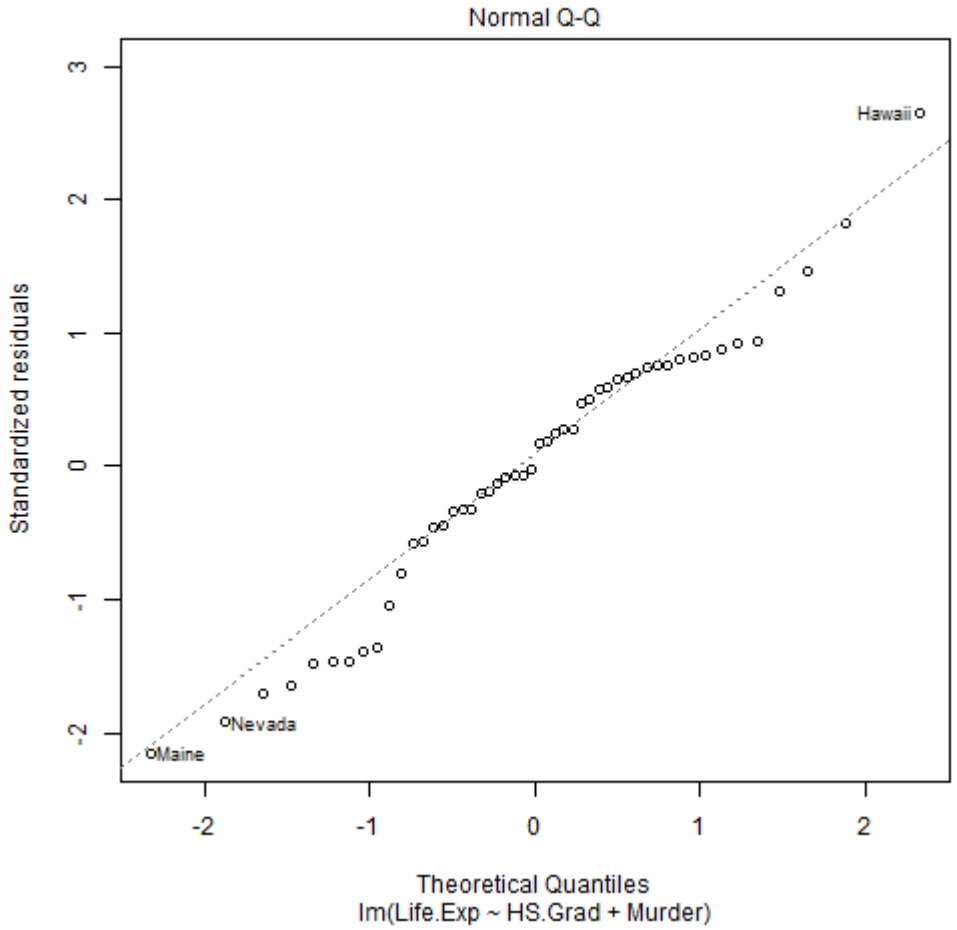

```
#####################
#                   #
#    Exercise 4     #
#                   #
#####################
par(mfrow=c(1,2)) # draw the two plots side by side

plot(state77$HS.Grad,resids,main="Residuals vs.
HS.Grad",xlab="HS.Grad",ylab="Residuals")
abline(h=,col="red")
plot(state77$Murder,resids,main="Residuals vs.
Murder",xlab="Murder",ylab="Residuals")
abline(h=,col="red")
```




```
par(mfrow=c(1,1)) # restore to the default

#####################
#                   #
#    Exercise 5     #
#                   #
#####################
#a.
qqnorm(resids,ylab="Residuals")
qqline(resids)
```

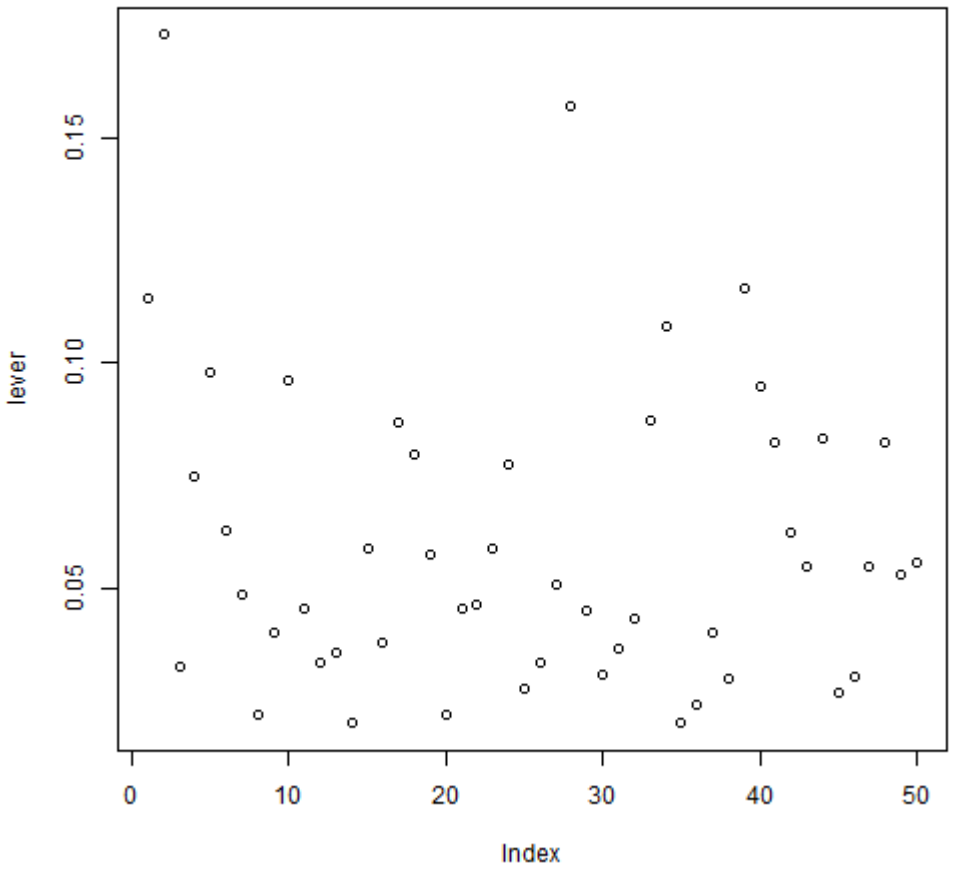



```
#b.
plot(model,which=2)
```




```
#####################
#                   #
#    Exercise 6     #
#                   #
#####################
#a.
stzed <- rstudent(model)

#b.
stzed[abs(stzed) > 2]

##    Hawaii     Maine
##  2.835488 -2.249583

#####################
#                   #
#    Exercise 7     #
#                   #
#####################
```

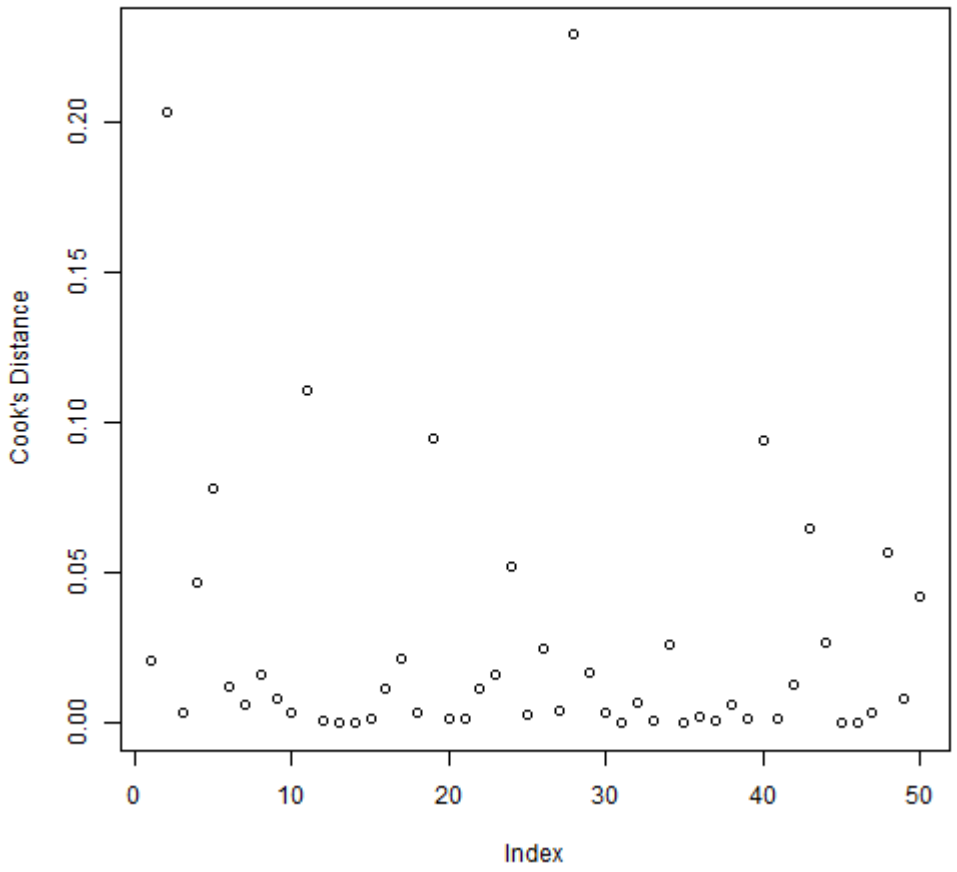
```
#a.
lever <- hat(model.matrix(model))
plot(lever)
```



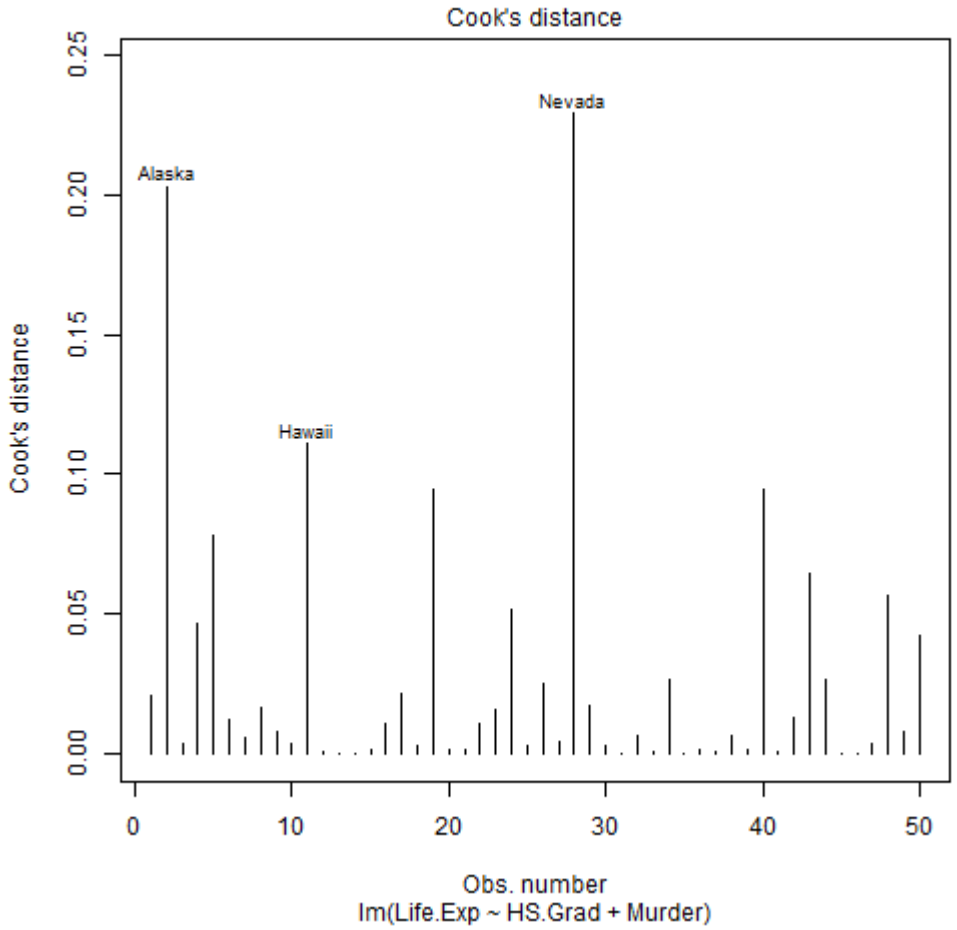

```
#b.
#obtain the threshold
thresh2 <- 2*length(model$coefficients)/length(lever)

#print leverage values above threshold
lever[lever > thresh2]

## [1] 0.1728282 0.1571342

#print corresponding state names
rownames(state77)[which(lever > thresh2)]

## [1] "Alaska" "Nevada"

#####################
```

```
#                  #
#    Exercise 8    #
#                  #
####################
#a.
dffits1 <- dffits(model)

#b.
thresh3 <- 2*sqrt(length(model$coefficients)/length(dffits1))
dffits1[dffits1 > thresh3]


##    Hawaii
## 0.6182642


#c.
dfbetas1 <- dfbetas(model)

#d.
thresh4 <- 2/sqrt(length(dfbetas1[,1]))
dfbetas1[dfbetas1[,1] > thresh4,1]  #for intercept


##    Alaska    Nevada
## 0.7036719 0.7400875

dfbetas1[dfbetas1[,2] > thresh4,2]  #for HS.Grad

##     California         Hawaii South Carolina  West Virginia
##      0.3993425      0.4430609      0.3855697      0.3613108

dfbetas1[dfbetas1[,3] > thresh4,3]  #for Murder

## California      Maine      Texas
##  0.3491024  0.4441167  0.3038958

####################
#                  #
#    Exercise 9    #
#                  #
####################
```

```
#a.
cooksd <- cooks.distance(model)
plot(cooksd,ylab="Cook's Distance")
```



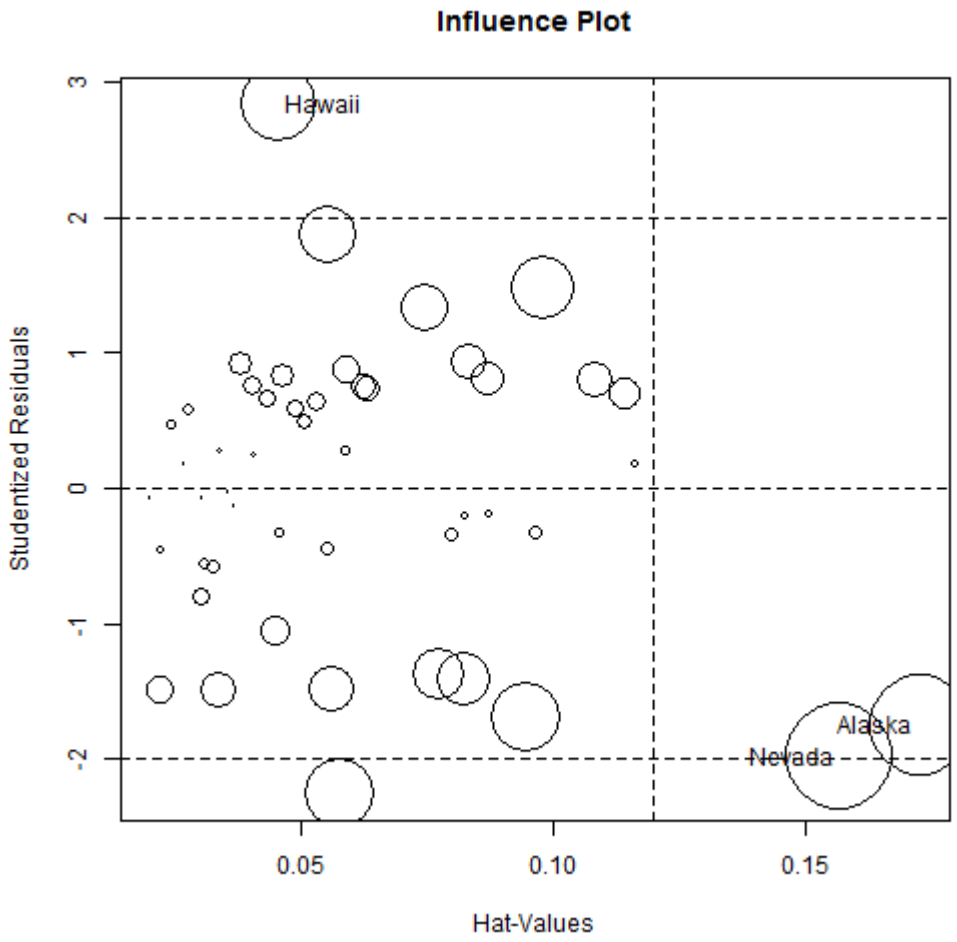

```
#b.
plot(model,which=4)
```




```
#c.
thresh <- 4/length(resids)
cooksd[cooksd > thresh]
```

```
##         Alaska         Hawaii          Maine         Nevada
South Carolina
##     0.20282640     0.11081779     0.09477421     0.22879279
0.09423789
```

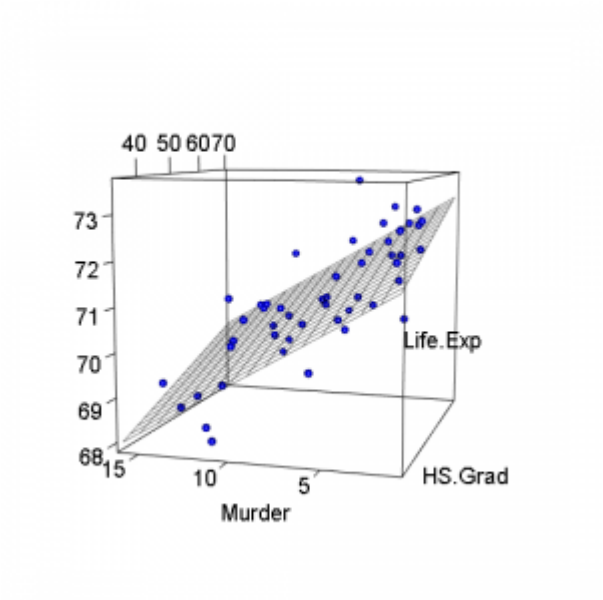
```
#####################
#                   #
#    Exercise 10    #
#                   #
#####################
library(car)
influencePlot(model, main="Influence Plot")
```




```
##           StudRes        Hat     CookD
## Alaska  -1.743144 0.17282816 0.2028264
## Hawaii   2.835488 0.04538585 0.1108178
## Nevada  -1.977284 0.15713419 0.2287928
```

---

# Multiple Regression (Part 1)




In the exercises below we cover some material on multiple regression in R.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

We will be using the dataset state.x77, which is part of the state datasets available in R. (Additional information about the dataset can be obtained by running help(state.x77).)

**Exercise 1**

a. Load the state datasets.
b. Convert the state.x77 dataset to a dataframe.
c. Rename the Life Exp variable to Life.Exp, and HS Grad to HS.Grad. (This avoids problems with referring to these variables when specifying a model.)

**Exercise 2**
Suppose we wanted to enter all the variables in a first-order linear regression model with Life Expectancy as the dependent variable. Fit this model.

**Exercise 3**

Suppose we wanted to remove the Income, Illiteracy, and Area variables from the model in Exercise 2. Use the update

function to fit this model.

**Learn more** about multiple linear regression in the online course Linear regression in R for Data Scientists. In this course you will learn how to:

- Model basic and complex real world problem using linear regression
- Understand when models are performing poorly and correct it
- Design complex models for hierarchical data
- And much more

**Exercise 4**
Let's assume that we have settled on a model that has HS.Grad and Murder as predictors. Fit this model.

**Exercise 5**
Add an interaction term to the model in Exercise 4 (3 different ways).

**Exercise 6**
For this and the remaining exercises in this set we will use the model from Exercise 4.

Obtain 95% confidence intervals for the coefficients of the two predictor variables.

**Exercise 7**
Predict the Life Expectancy for a state where 55% of the population are High School graduates, and the murder rate is 8 per 100,000.

**Exercise 8**

Obtain a 98% confidence interval for the mean Life Expectancy in a state where 55% of the population are High School graduates, and the murder rate is 8 per 100,000.

**Exercise 9**

Obtain a 98% confidence interval for the Life Expectancy of a person living in a state where 55% of the population are High School graduates, and the murder rate is 8 per 100,000.

**Exercise 10**

Since our model only has two predictor variables, we can generate a 3D plot of our data and the fitted regression plane. Create this plot.

---

# Multiple Regression (Part 1) Solutions

Below are the solutions to these exercises on Multiple Regression (part 1).

**Learn more** about multiple linear regression in the online course Linear regression in R for Data Scientists. In this course you will learn how to:

- Model basic and complex real world problem using linear regression
- Understand when models are performing poorly and correct it
- Design complex models for hierarchical data
- And much more

```
#####################
#                   #
#    Exercise 1     #
```

```
#                   #
#####################
#a.
data(state)

#b.
state77 <- as.data.frame(state.x77)

#c.
names(state77)[4] <- "Life.Exp"
names(state77)[6] <- "HS.Grad"


#####################
#                   #
#    Exercise 2     #
#                   #
#####################
model <- lm(Life.Exp ~ ., data=state77) #the '.' means 'all'
summary(model)

##
## Call:
## lm(formula = Life.Exp ~ ., data = state77)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586  < 2e-16 ***
## Population   5.180e-05  2.919e-05   1.775   0.0832 .
## Income      -2.180e-05  2.444e-04  -0.089   0.9293
## Illiteracy   3.382e-02  3.663e-01   0.092   0.9269
## Murder      -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## HS.Grad      4.893e-02  2.332e-02   2.098   0.0420 *
## Frost       -5.735e-03  3.143e-03  -1.825   0.0752 .
## Area        -7.383e-08  1.668e-06  -0.044   0.9649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

```
' 1
##
## Residual standard error: 0.7448 on 42 degrees of freedom
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10

####################
#                  #
#    Exercise 3    #
#                  #
####################
model2 <- update(model, . ~ . -Income -Illiteracy -Area) #the
'.' means 'same as in original model'
summary(model2)

##
## Call:
## lm(formula = Life.Exp ~ Population + Murder + HS.Grad +
Frost,
##     data = state77)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47095 -0.53464 -0.03701  0.57621  1.50683
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.103e+01  9.529e-01  74.542  < 2e-16 ***
## Population   5.014e-05  2.512e-05   1.996  0.05201 .
## Murder      -3.001e-01  3.661e-02  -8.199 1.77e-10 ***
## HS.Grad      4.658e-02  1.483e-02   3.142  0.00297 **
## Frost       -5.943e-03  2.421e-03  -2.455  0.01802 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.7197 on 45 degrees of freedom
## Multiple R-squared:  0.736,  Adjusted R-squared:  0.7126
## F-statistic: 31.37 on 4 and 45 DF,  p-value: 1.696e-12

####################
```

```
#                   #
#    Exercise 4     #
#                   #
#####################
model3 <- lm(Life.Exp ~ HS.Grad + Murder, data=state77)
summary(model3)

##
## Call:
## lm(formula = Life.Exp ~ HS.Grad + Murder, data = state77)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66758 -0.41801  0.05602  0.55913  2.05625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 70.29708    1.01567  69.213  < 2e-16 ***
## HS.Grad      0.04389    0.01613   2.721  0.00909 **
## Murder      -0.23709    0.03529  -6.719 2.18e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.7959 on 47 degrees of freedom
## Multiple R-squared:  0.6628, Adjusted R-squared:  0.6485
## F-statistic:  46.2 on 2 and 47 DF,  p-value: 8.016e-12

#####################
#                   #
#    Exercise 5     #
#                   #
#####################
model4 <- lm(Life.Exp ~ HS.Grad + Murder + HS.Grad:Murder,
data=state77)
summary(model4)

##
## Call:
## lm(formula = Life.Exp ~ HS.Grad + Murder + HS.Grad:Murder,
data = state77)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66077 -0.43846  0.06362  0.52665  1.99416
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    67.831203   2.530131  26.809   <2e-16 ***
## HS.Grad         0.089368   0.045684   1.956   0.0565 .
## Murder          0.023510   0.247487   0.095   0.9247
## HS.Grad:Murder -0.004959   0.004661  -1.064   0.2930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.7948 on 46 degrees of freedom
## Multiple R-squared:  0.6709, Adjusted R-squared:  0.6495
## F-statistic: 31.26 on 3 and 46 DF,  p-value: 3.592e-11
```

```
model4 <- lm(Life.Exp ~ HS.Grad*Murder, data=state77)
summary(model4)
```

```
##
## Call:
## lm(formula = Life.Exp ~ HS.Grad * Murder, data = state77)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66077 -0.43846  0.06362  0.52665  1.99416
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    67.831203   2.530131  26.809   <2e-16 ***
## HS.Grad         0.089368   0.045684   1.956   0.0565 .
## Murder          0.023510   0.247487   0.095   0.9247
## HS.Grad:Murder -0.004959   0.004661  -1.064   0.2930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.7948 on 46 degrees of freedom
```

```
## Multiple R-squared:  0.6709, Adjusted R-squared:  0.6495
## F-statistic: 31.26 on 3 and 46 DF,  p-value: 3.592e-11

model4 <- lm(Life.Exp ~ (HS.Grad+Murder)^2, data=state77)
summary(model4)

##
## Call:
## lm(formula = Life.Exp ~ (HS.Grad + Murder)^2, data =
state77)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66077 -0.43846  0.06362  0.52665  1.99416
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    67.831203   2.530131  26.809   <2e-16 ***
## HS.Grad         0.089368   0.045684   1.956   0.0565 .
## Murder          0.023510   0.247487   0.095   0.9247
## HS.Grad:Murder -0.004959   0.004661  -1.064   0.2930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 0.7948 on 46 degrees of freedom
## Multiple R-squared:  0.6709, Adjusted R-squared:  0.6495
## F-statistic: 31.26 on 3 and 46 DF,  p-value: 3.592e-11

####################
#                  #
#    Exercise 6    #
#                  #
####################
confint(model3, level=0.95)

##                   2.5 %      97.5 %
## (Intercept) 68.25382379 72.34034424
## HS.Grad      0.01144419  0.07633041
## Murder      -0.30807483 -0.16610536
```

```
####################
#                  #
#    Exercise 7    #
#                  #
####################
predict(model3,data.frame(HS.Grad=55,Murder=8))

##        1
## 70.81416

####################
#                  #
#    Exercise 8    #
#                  #
####################
predict(model3,data.frame(HS.Grad=55,Murder=8),interval="confi
dence",level=0.98)

##        fit      lwr     upr
## 1 70.81416 70.52183 71.1065

####################
#                  #
#    Exercise 9    #
#                  #
####################
predict(model3,data.frame(HS.Grad=55,Murder=8),interval="predi
ction",level=0.98)

##        fit      lwr      upr
## 1 70.81416 68.87527 72.75306

####################
#                  #
#    Exercise 10   #
#                  #
####################
library(rgl)

plotdat                                                    <-
```

```
expand.grid(HS.Grad=seq(34,70,by=2),Murder=seq(1,16,by=1))
plotdat$pred1 <- predict(model3,newdata=plotdat)
with(state77,plot3d(HS.Grad,Murder,Life.Exp,    col="blue",
size=1,  type="s"))
with(plotdat,surface3d(unique(HS.Grad),unique(Murder),pred1,
        alpha=0.5,front="line", back="line"))
```