ISL Chapter 3 Exercises

Liz

4/13/2021

Contents

Chapter 3 Exercises

Name	Content
TYPE	notes
BOOK	An Introduction to Statistical Learning
AUTHORS	Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani
PUBLISHER	Springer

```
# the data from the book can be downloaded using
# install.packages('ISLR').  It's then loaded using the line
# below. The MASS dataset is preloaded into R.
library(MASS)
library(ISLR)
library(car)  # needed for the VIF
## Loading required package: carData
library(interactions)  # needed for interaction graph
```

Simple Linear Models

The examples below use the Boston data set from the MASS library. It includes the median house value (*medv*) for 506 neighborhoods around Boston. It also includes a lot of predictors including, *age*, *rm*, and *lstat* which are the age of the house, average number of rooms, and the percent of households with low income status.

The $lm()$ function is used to fit a simple linear model. It takes the form $lm(y \sim x, data = dataset)$ where y is the response (here, *medv*) and x is the predictor (here, *lstat*). $data = dataset$ is the dataset where R should look for the variables of interest (*medv* and *lstat* are in the **Boston** dataset.)

```

lm.fit = lm(medv ~ lstat, data = Boston)  # we don't have to name the model lm.fit,
# it can be named anything, but convention says it's best to
# name it something useful
lm.fit  # calling lm.fit will print out some basic information about our model.
##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
```

```
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

The intercept is β_0 and the slope is β_1 from our linear regression equation: $y = \beta_0 + \beta_1 X + \epsilon$.

In this case, our equation would be $medv = 34.55 - 0.95 \times lstat + \epsilon$

To get more detailed information, we use the $summary()$ function.

```
summary(lm.fit)
##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

How to interpret linear regression output:

1. the *call* shows us the linear model we fit.
   - in this case, we're regressing median house value (*medv*) onto the percent of households with low income status (*lstat*) using the **Boston** dataset
2. the *residuals* section is the five number summary of the model's *residuals*.
   - The *residuals* show the different between the actual median home value (*medv*) and the predicted median home value ($\hat{medv}$).
   - Typically, this information is more useful when plotted (below).
3. The $coefficients$ are the different values for β_0 and β_1.
   - The first row (*intercept*) is for β_0 (the median house value when we consider the average percent households with a low income status).
     - The first number *intercept* and *estimate* is **34.55384**.
       - The median house value is 34.5538 when the percent of households with low income status is 0. Since the *medv* column is reported in thousands, the median house value when there are no households with low income status is $34,556.84.
     - The second number *intercept* and *Std. Error* is **0.56263**.
       - *Std. Error* means *Standard Error*
       - Our model will be off by **0.56263** if we used the intercept to predict everything.
     - The third number *intercept* and *t-value* is **61.41**
       - This is the standardized estimate of the mean.
       - Usually, the higher the number, the better.
     - The fourth number *intercept* and *p-value* is **<2e-16**
       - Area more extreme than the t that you found.
       - Probability you didn't find something by random chance given the null hypothesis is true.
       - The probability that our results are due to random chance is pretty much 0.

      - *note* <2e-16 is the lowest number that R can list.
  - The second row (*lstat*) is our β_1 (the effect that a one percent increase in households with low income status (*lstat*) has on median home value (*medv*)).
    - The first number *lstat* and *estimate* is **-0.95005.**
      - The slope term is saying that for every 1 percent increase in households with low income status, the median house value *decreases* by 0.95005
      - This is the size of effect that *lstat* has on *medv*
    - The second number *lstat* and *Std. Error* is **0.0387**.
      - This measures the average amount that *lstat* varies from the actual average value of *medv*.
      - The lower the number relative to *lstat* the better.
      - Here, the *medv* based on low income status households varies by roughly 4%.
      - This is a measure of how real the effect is.
    - The third number *lstat* and *t-value* is **-24.5**.
      - This is a measure of how many standard deviations our *lstat* estimate is from 0.
      - We want this to be as far away from 0 as possible, because that allows us to reject the null hypothesis.
      - Here, it's pretty far from zero and negatively correlated. That means that as *lstat* goes up, *medv* goes down.
      - This is a measure of how real the effect is.
    - The fourth number *lstat* and $p - value$ is **<2e-16**
      - It means the same as it does for the intercept.
      - Small values for slope and coefficient p-values indicate we can reject the null hypothesis. There is a relationship between *medv* and *lstat*.
      - This is a measure of how real the effect is.

- The $Significance\ Codes$ tell you at what significance your *p-values* are reported at. Three stars represents highly significant p-values.
- The *Residual Standard Error* is **6.216 on 504 degrees of freedom**.
  - This measures the quality of our fit.
  - It is the average amount that the response will deviate from the true regression line.
  - Here, *medv* will deviate from the true regression line by approximately 6,216 dollars (since *medv* is given in thousands)
  - This is always in whatever units Y is. So if our response was in feet, the RSE would also be in feet.
  - *degrees of freedom* is the number of data points used to estimate the parameters minus the parameters used (or restrictions placed.)
    - Here we have 506 data points with 2 restrictions (*medv* and *lstat*), therefore 504 degrees of freedom.
- $Multiple\ R - Squared$ is **0.5441**.
  - This is a standardized estimate of how well our model is fitting the data. It will always be between 0 and 1.
  - A little more than half (54%) of the variance in median home value (*medv*) is explained by households with low income status (*lstat*).
- The $Adjusted\ R - Squared$ is **0.5432**
  - Since the *multiple R-Squared* will always increase as more variables are included, so the *adjusted R-Squared* is preferred.
  - Here, it is essentially the same at 54% of the variance in *medv* being explained by *lstat*.
- The $F - Statistic$ is **602 on 504 DF** with a *p-value* of **2e-16**
  - This is an indication of whether there is a relationship between *lstat* and *medv*.
  - The further away from 1 the better.
  - If you have a lot of data points, the *f-statistic* only needs to be a little away from 1.
  - If you have few data points, the *f-statistic* needs to be larger to determine whether there is a relationship between the predictor and response.
  - Here, we have 506 data points, and the *f-statistic* is fairly far away from 1, so there is probably a

relationship between *lstat* and *medv*.

You can use the *names()* function to find out what else is stored in the model

```
names(lm.fit)
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

You can extract these values in the same way that you would variables, *e.g.,* lm.fit$coefficients. It's better to use the *coef()* function.

```
coef(lm.fit)
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

This gives us the coefficient estimates for the slope and intercept, which are also included in the summary above.

You can use the *confint()* function to get the confidence interval for the coefficient estimates.

```
confint(lm.fit)
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

1. The confidence intervals tell us that we're 95% sure that in the absence of low income households (*lstat*), the median home value (*medv* will be between $33,448 and $35,659.
2. The bottom two numbers tell us that for each 1% increase in low income households (*lstat*), there will be a corresponding *decrease* in median home value (*medv*) between $873 and $1020.

We can predict the confidence levels based on a value for *lstat* using the *predict()* function.

```
predict(lm.fit, data.frame(lstat=c(5,10,15)), # lm.fit is the linear model.
         #data.frame() tells r to create
         #a dataframe object that includes the lstat percentages for 5, 10, and 15%.
         interval = "confidence") #interval=confidence returns the confidence intervals.
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

1. When the neighborhood is 5% low income households (*lstat*), we're 95% sure that the median home value (*medv*) will be between $20,007 and $30,600.
2. When the neighborhood is 10% low income households (*lstat*), we're 95% sure that the median home value (*medv*) will be between $24,470 and $25,633.
3. When the neighborhood is 10% low income households (*lstat*), we're 95% sure that the median home value (*medv*) will be between $19,731 and $20,874.

If we change interval="confidence" to interval="prediction" we can see the prediction interval which reflects the uncertainty around a single value. + The prediction interval will usually be wider than the confidence intervals.

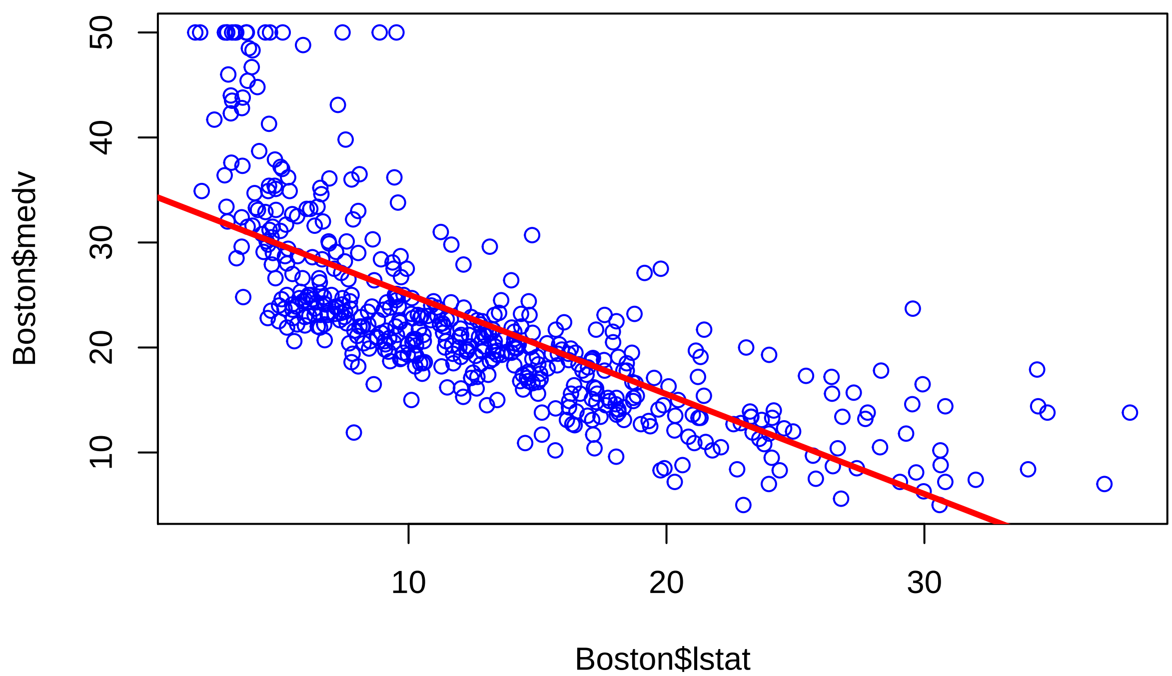
```
predict(lm.fit, data.frame(lstat = c(5, 10, 15)), interval = "prediction")
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

The confidence and prediction intervals are centered around the same point (25.05, first column, second number) but the prediction intervals are much wider.

- For example, for a *lstat* of 10, the prediction interval is 95% sure the median house value (*medv*) is between $12,828 and $37,280

plotting the regression line

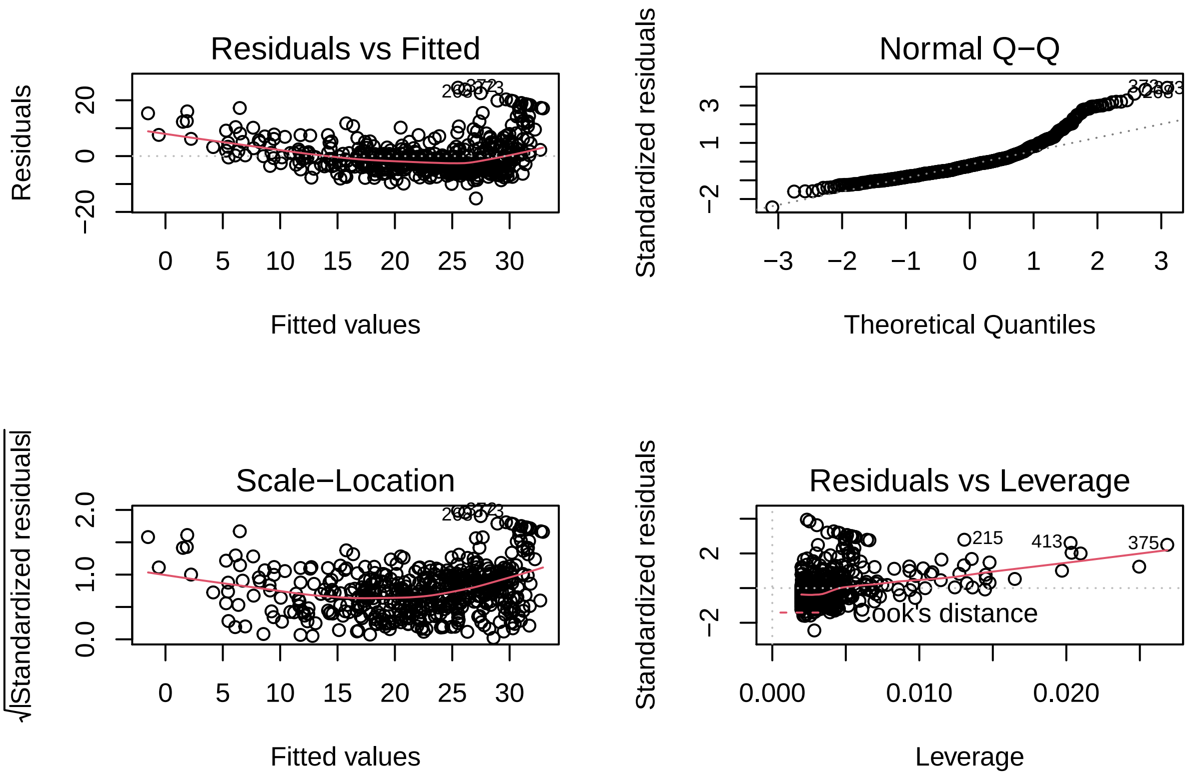
```
plot(Boston$lstat, Boston$medv, col = "blue")
# this creates a scatterplot of low income status vs.median
# house value this adds the straight line through the plot
# where a is the intercept and b is the slope.
abline(lm.fit, lwd = 3, col = "red")
```



Plotting the residuals will help us find out if there are problems with our model. These problems include non-linearity of the data, correlation of error terms, non-constant variance of error terms, identifying any outliers, identifying high leverage points, and determining if there is collinearity.

Four diagnostic plots are automatically included by using the *plot()* function. However, we want to display them in a two-by-two grid. We use the *par()* function to do so.

```
par(mfrow = c(2, 2))  # mfrow tells the par function to make a 2x2 grid
plot(lm.fit)
```



How to interpret these plots

1. Top left (Residual v. Fitted): This graph shows there might be some non-linearity in the data. Ideally, you will see little evidence of a pattern. The points will "bounce randomly" around the horizontal line and no one point will stand out. Since our points follow the red line, it suggests non-linearity of the data.

2. Top right (Normal Q-Q): This is a quantile-quantile plot (Q-Q plot). It helps us determine if the data follows a theoretical distribution (*e.g.* normal or exponential). If the data come from the same distribution (*e.g.* if the fitted and real values are both normally distributed) we should see a fairly straight line. Here, it's clear that they follow the same distribution because the line is fairly straight.

3. Bottom left (Scale-Location): This is similar to the Residual v. Fitted plot, but makes it easier to determine if there is homoskedasticity (if the errors are the same across all values of the independent variable). There are two things to check for.

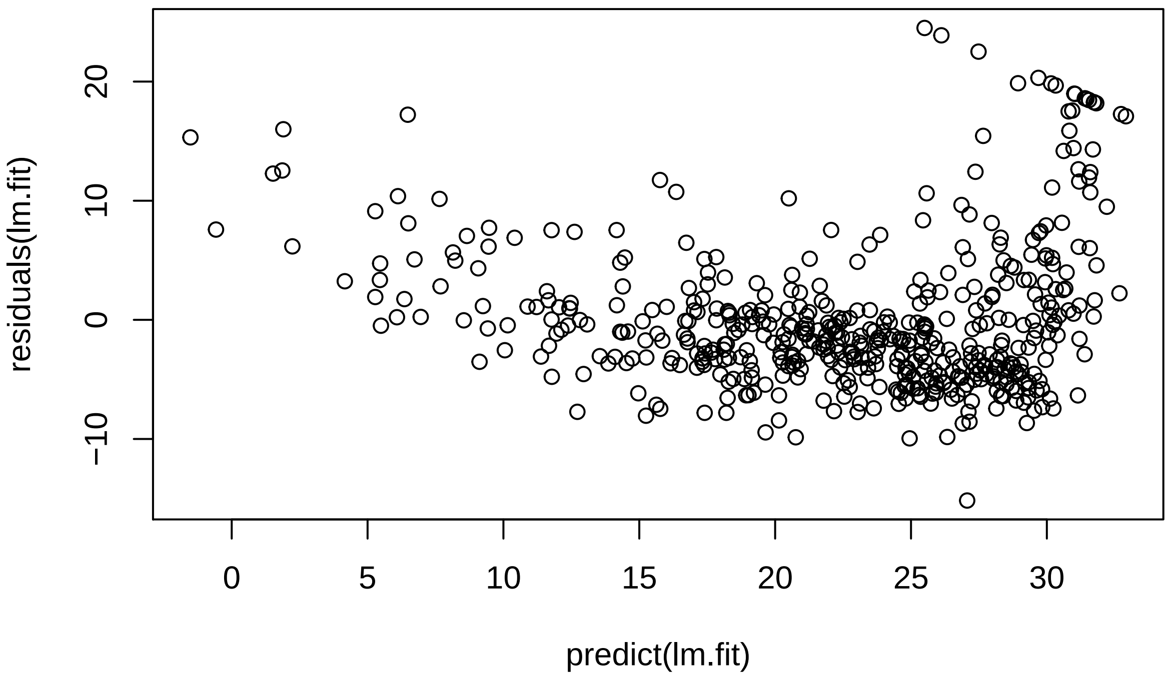
   - the red line is approximately horizontal. This means that the average magnitude of the standardized residuals isn't changing much.
   - the distribution of the points don't vary much around the red line.

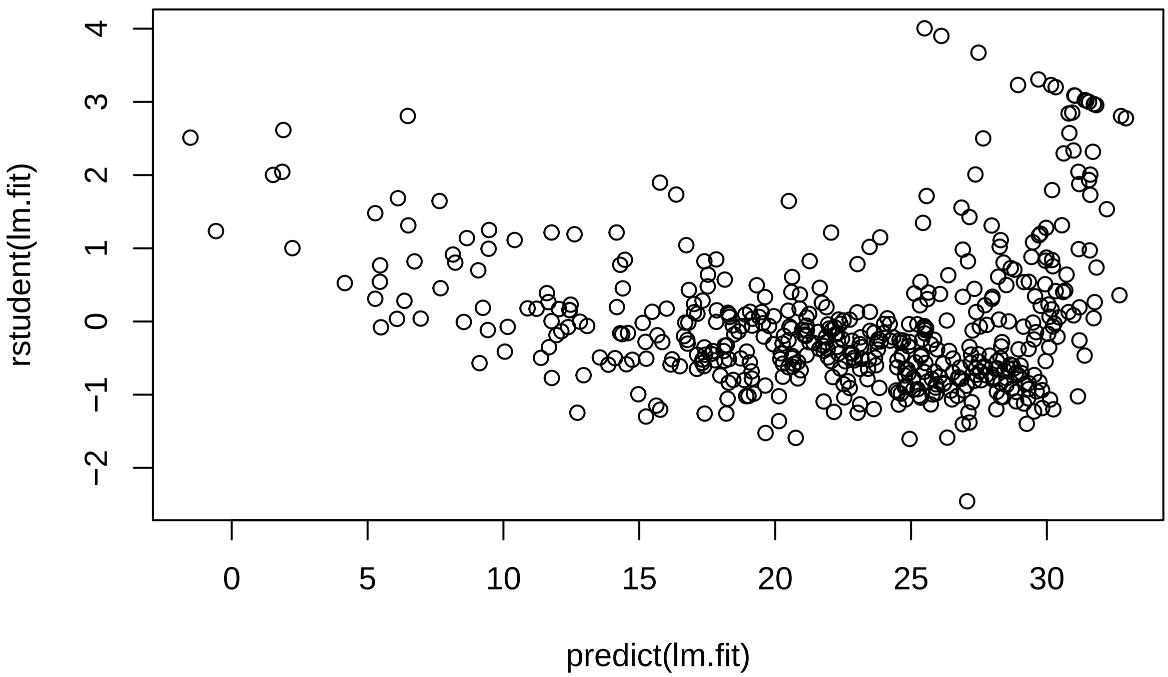
   Here, there is more evidence of non-linearity. The magnitude is lowest around 0 and higher around 20-30.

4. Bottom right (Residuals vs. Leverage): This plot is used to determine if there are outliers in the data. Here, there are several outliers which are far away from the other points. Cook's distance helps us determine if any points have high-leverage (so deleting them would change the model significantly). One point falls outside of Cook's distance here and is said to have high leverage.

Alternatively, using the *residuals()* function will output the residuals from the linear regression model. I did not include this here because it takes up a lot of space. You can also plot the residuals against the fitted values using the plot function combined with the *residuals()* and *rstudent()* functions. *rstudent()* returns the studentized residuals.

```
plot(predict(lm.fit), residuals(lm.fit))
plot(predict(lm.fit), rstudent(lm.fit))
```



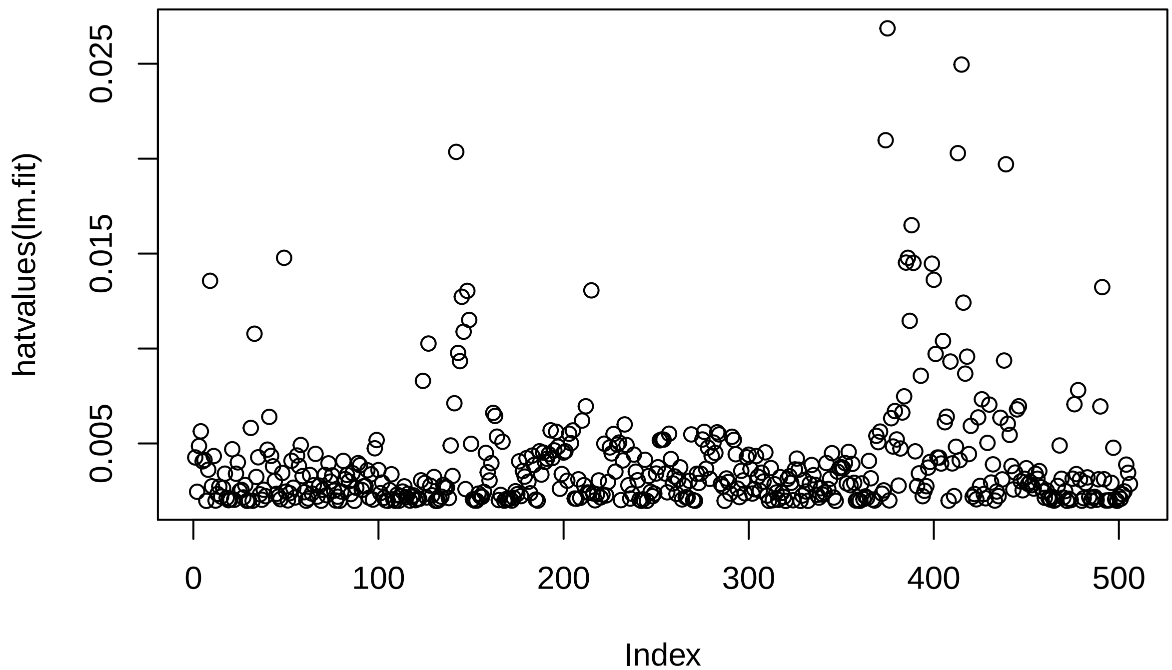


How to read the residual (& studentized residual) v. predicted values plot.

1. These plots are read in a similar manner to the residual v. fitted value above. You want to make sure that the points are not following any sort of pattern. Here they skew towards the 20-30 range which suggests some non-linearity.

Leverage statistics can be computed using the *hatvalues()* function.

```
plot(hatvalues(lm.fit))
which.max(hatvalues(lm.fit))
## 375
## 375
```



How to interpret a hatvalue() plot:

This shows the distribution of the data with particular focus on points that may have high leverage. The points that are far away from the main group may have high leverage.

which.max() identifies the index of the largest element. This tells us that the observation with the largest leverage statistic is 375.

Multiple Linear Regression

The *lm()* function can be used to fit multiple linear models. It takes the form $lm(y\tilde{x}1 + x2 + x3, data = data)$

- $x1 + x2 + x3$ are the three predictors. This can be two to any number of predictors.
- data = data is where the data can be found. For this exercise, it would read data=Boston since we're using the Boston data set again.

```
lm.mult = lm(medv ~ lstat + age, data = Boston)  # This looks at median home value
# based on the number of low income households in the area
# and average age of the houses.
summary(lm.mult)
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

How to interpret the multiple linear regression output

1. The call and residuals are interpreted in the same way that the simple linear regression model are.
2. The *intercept* and *estimate* is **33.22276**.
   - This is the median home value (*medv*) when *both* low income households *and* the average age of the house are 0.
   - Since *medv* is in \$1,000s, the median home value in a neighborhood with there are brand new houses and no low income households is \$33,222.
3. The *intercept* and *Std. Error (Standard Error)*
4. The *intercept* and the *t-value* is **-21.416**.
   - This is the hypothesis test for $lstat = 0$
5. The *intercept* and the *Pr(>|t|)*
6. The *lstat* and the *estimate* is **-1.03207**
   - This is the effect in *medv* for a one percent increase in low income households *lstat* while controlling for *age*.
   - Here, a 1% increase in *lstat* results in a \$1,032 *decrease* in *medv*.
7. The *age* and the *estimate* is **0.03454**
   - This is the effect in *medv* for a one year increase in average of houses *age* when controlling for *lstat*.
   - Here, a 1 age increase in *age* results in approximately a \$35 increase in *medv*
8. The *standard error*, *t-value*, and *p-value* are all interpreted in the same way.
   - For example, for *lstat* the *standard error* is **0.04819**
     - that means our model will be off by **0.73085**
     - it is in the units of the the response (here, *medv*)
     - So we're off by about 0.73085*1000 $\approx$ \$731
   - For example, for *lstat* the *t-value* is **-21.416** with a *p-value* of **<2e-16**.
     - This is a measure of how many standard deviations our *lstat* estimate is from 0.
     - We want this to be as far away from 0 as possible, because that allows us to reject the null hypothesis.
     - Here, it's pretty far from zero and negatively correlated. That means that as *lstat* goes up, *medv* goes down while controlling for *age*.
     - This is a measure of how real the effect is.
     - the *p-value* is interpreted in the same manner as above.
9. The *significance codes* are interpreted in the same wasy as in a simple linear regression.
10. The *residual standard error* is interpreted in the same way as in a simple linear regression.
    - Here, *medv* will deviate from the true regression line by approximately \$6,173 (since *medv* is given in thousands)
11. The *multiple R-squared* shows us that approximately 55% of the variation in median home value can be explained by *lstat* and *age*.
    - By adding age, we can explain approximately 10% more variation than with *lstat* alone.
12. The *F-statistic* and associated *p-value* are **309 on 2 and 503 degrees of freedom** and *2.2e-16*.
    - The f-statistic is a test of the null hypothesis
    - here, the null hypothesis is that *lstat* and *age* are not related to *medv*.

- it is pretty far away from 0 with high significance, so we can reject the null hypothesis.

In the Boston data set there are 13 variables. Adding them each by using $+x1 + ... + x13$ would be tedious. To look at the regression output for all the variables at once we use the notation ˜.

```
lm.all = lm(medv ~ ., data = Boston)  # This looks at median home value based on
# the number of low income households in the area and average
# age of the houses.
summary(lm.all)
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

How to interpret a multiple regression model with all variables + It is the same as the above two interpretations. + remember that each coefficient is interpreted as if all other variables are controlled for.

You can pull out individual components using the $ $ $ operator.

```
summary(lm.all)$r.sq  # this pulls out the R Squared Value
## [1] 0.7406427
```

How to interpret the R-Squared value

1. The R^2 value is **0.7406427**
   - The R^2 will always lie between 0 and 1.
   - A larger number is good, it means that more of the variance is explained by the predictors.
   - Approximately 74% of the variance in *medv* is explained by all of the variables.
   - R^2 will always increase when more variables are added, so we focus on the adjusted R^2 in multiple regression models which is included in the *summary()* above.

```
summary(lm.all)$sigma  # this pulls out the Residual Standard Error
## [1] 4.745298
```

How to interpret the Residual Standard Error value

1. The *RSE* is **4.745298**
   - it is the residual variation
   - it represents the average of the observations points around the fitted regression line.
   - it's an absolute measure of patterns in the data that can't be explained by the model.
   - a small *RSE* means that the model fits the data pretty well.
   - Whether or not this is a good value depends on the problem context. Since *RSE* is measured in Y units, this means our model is off by about $4,745.

The *Variance Influence Factor (VIF)* can be used to determine if there is multicollinearity. Base R does not have a function to do this, so we can install the *car* package. See the top of this document on how to install packages in R.

```
library(car)
vif(lm.all)
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.792192 2.298758 3.991596 1.073995 4.393720 1.933744 3.100826 3.955945
##      rad      tax  ptratio    black    lstat
## 7.484496 9.008554 1.799084 1.348521 2.941491
```

How to interpret VIF:

1. The *VIF* ranges from 1 up. Generally, this means:
   - 1 = not correlated
   - Between 1 and 5 = moderately correlated
   - Greater than 5 = highly correlated.
2. It is what percentage the variance is inflated for each coefficient.
3. The greater the VIF, the less reliable your regression results are going to be.

You can also run regressions for all variables except one. Above, age has a high p-value, so we might want to exclude it from our regression.

```
lm.exception = lm(medv ~ . - age, data = Boston)  # This looks at median home value
# based on the number of low income households in the area
# and average age of the houses.
summary(lm.exception)
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
```

```
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

The interpretation is the same.

Interaction Terms

Including $lsat : age$ in our lm() call tells R to include an interaction term between $lstat$ and age. + Using $lstat * age$ tells R to simultaneously include $lstat$, age, and the interaction term between $lstat \times age$ as predictors.

```
lm.interact = lm(medv ~ lstat * age, data = Boston)
summary(lm.interact)
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

How to interpret a linear regression model with interaction terms

1. The *call* and *residuals* are interpreted in in the same way as before.
2. The *intercept* and *estimate* is **36.0885359**
   - this is our **baseline**
   - This is the *medv* under control conditions, that is the median value of the house that is brand new and is in a neighborhood with no low income households.
3. The *lstat* and *estimate* is **-1.3921168**
   - This is the expected decrease in *medv* when the age of the house is controlled for.
   - We expect that the *medv* will decrease by *1,392* with every one percent increase in low income households when compared to the baseline.
4. The *age* and *estimate* is **-0.0007209**
   - This is the expected increase in *medv* when low income households is controlled for.

  - We expect that the *medv* will decrease by \$0.72 for every year the house is standing.
5. The *lstat:age* is **0.0041560**
   - This is the interaction for *lstat* and *age* on *medv*.
   - With a *p-value* of **0.0252** the interaction effect **is** statistically significant.
     - This suggests that the estimated difference in *medv* between two houses whose ages differ by one year is equal to about 0.4%.
     - Since it is statistically significant, we can determine that there is an interaction effect between *lstat* and *age*, but it is difficult to interpret what *0.0041560* number means. The best way to determine the effect of an interaction term is to plot it.

```
# I'll use the interplot package for this graph. See the top of the document
library(interplot)
## Loading required package: ggplot2
## Loading required package: abind
## Loading required package: arm
## Loading required package: Matrix
## Loading required package: lme4
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
##
## arm (Version 1.11-2, built: 2020-7-27)
## Working directory is G:/My Drive/Github/statistics/ISL Machine Learning/exercises
##
## Attaching package: 'arm'
## The following object is masked from 'package:car':
##
##     logit
interplot(m=lm.interact, var1="lstat", var2="age") +
   # m = model, var1 is the main predictor, var2 is the interaction effect
   xlab("Age") + # this labels the x axis
   ylab("Low Income Household Status") + # label the y axis
   ggtitle("Estimated Coefficient for Low Income Household Status
           \n on Median Home Value by Average of the House") + # this labels the whole graph
   theme(plot.title = element_text(hjust = 0.5)) # this centers the plot title.
```

Here we see that as age increases, the percentage of households that are low income goes up.

Non-linear Transformations of the Predictors

We can make non-linear transformations in our $lm()$ call by using the function $I()$ inside the regression model. This is because the ˆ symbol has special meaning inside $lm()$ and $I()$ tells $lm()$ to ignore that meaning.

```
lm.nonlinear = lm(medv ~ lstat + I(lstat^2), data = Boston)
summary(lm.nonlinear)
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
```

```
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

This regresses *medv* onto *lstat* and $lstat^2$

The near-zero *p-value* of the quadratic term $I(lstat^2)$ suggests that this might lead to an improved model.

Using the *anova*() function will tell us the extent to which the quadratic measure fits the data better.

```
anova(lm.fit, lm.nonlinear)
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

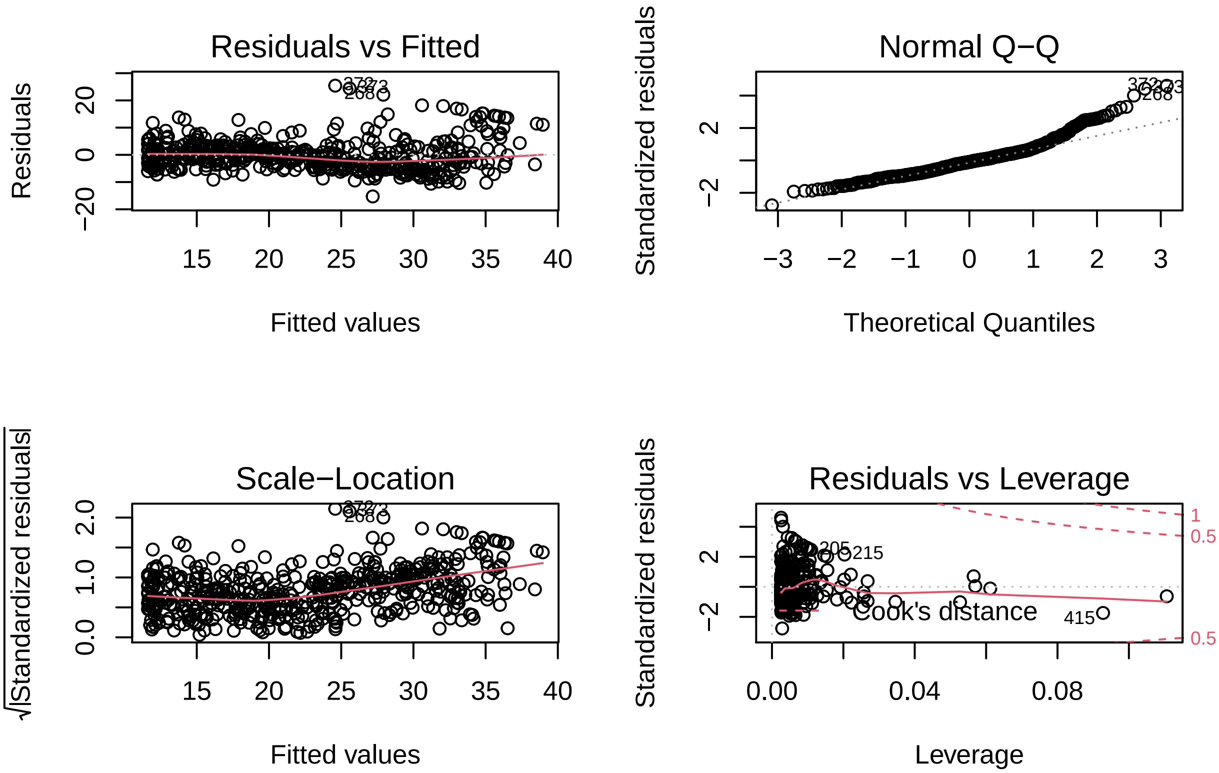
The *anova*() function performs a hypothesis test comparing two models. The null hypothesis is that both models fit the data equally well. The alternative hypothesis is that the model with the non-linear transformation fits the data better.

How to read the Anova() output

1. The first column is the *degrees of freedom*.
2. The second column is the RSS. Since the RSS for model two ($lstat^2$) is smaller, it indicates that the non-linear model may be a better fit.

When we plotted the residuals above, there was evidence for non-linearity. When we plot the residuals for the $lstat^2$ model, we can see how the residuals show little discernible pattern (which is what we want).

```
par(mfrow = c(2, 2))
plot(lm.nonlinear)
```



You can use $I(\hat{}3)$ to create a cubic fit, but a better option is to use the $poly()$ function.

The code below creates a fifth order polynomial fit and will show the RSS for $lstat^1$ to $lstat^5$.

```
lm.poly = lm(medv ~ poly(lstat, 5), data = Boston)
summary(lm.poly)
##
## Call:
## lm(formula = medv ~ poly(lstat, 5), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5433  -3.1039  -0.7052   2.0844  27.1153
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
## poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
## poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
## poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
## poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
## poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.215 on 500 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6785
```

```
## F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

Here, the *multiple R-squared* is **0.6817** which shows that the fit up to the 5th polynomial leads to improvement in the model fit. Additionally, the *f-statistic* is **214.2** which is pretty far from zero and is statistically significant with a p-value close to zero, **2.2e-16**.

Qualitative Predictors

A new data set is used for the following examples. It is the *Carseats* data in the *ISLR* library. See the top of this document for how to install and load the ISLR package.

```

carseats <- Carseats
names(carseats)
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

ShelveLoc is where the carseat is located on the store shelf and takes three values: *Bad, Medium, and Good*. R will create dummy variables automatically for this kind of qualitative data.

```
lm.carseats = lm(Sales ~ . + Income:Advertising + Price:Age,
    data = carseats)
summary(lm.carseats)
##
## Call:
## lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9208 -0.7503  0.0177  0.6754  3.3413
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
## CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
## Income              0.0108940  0.0026044   4.183 3.57e-05 ***
## Advertising         0.0702462  0.0226091   3.107 0.002030 **
## Population          0.0001592  0.0003679   0.433 0.665330
## Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
## ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
## ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
## Age                -0.0579466  0.0159506  -3.633 0.000318 ***
## Education          -0.0208525  0.0196131  -1.063 0.288361
## UrbanYes            0.1401597  0.1124019   1.247 0.213171
## USYes              -0.1575571  0.1489234  -1.058 0.290729
## Income:Advertising  0.0007510  0.0002784   2.698 0.007290 **
## Price:Age           0.0001068  0.0001333   0.801 0.423812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.011 on 386 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8719
## F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

Here, the outcome of interest is *Sales*. We're regressing *Sales* onto all the predictors in the *carseats* dataset and including the interactions between *Income* and *Advertising* and *Price* and *Age*.

By using *contrasts()* we can see what coding scheme R used for *ShelveLoc*

```
contrasts(carseats$ShelveLoc)
##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```

How to interpret the contrasts() output: Each of the three values *Good, Medium, Bad* are coded as dummy variables.

1. If the carseat is in a *Good* location, it is coded with a 1. 0 otherwise.
2. If the carseat is in a *Medium* location, it is coded with a 1. 0 otherwise.
3. If the carseat is in a *Bad* location, it is coded with two zeros.

Looking at the regression output, the *estimate* for *ShelveLocGood* is **4.8486762** which indicates that a good shelving location is associated with higher sales than a bad shelving location. *ShelveLocMedium* is also positively associated with more sales with an *estimate* of **1.9532620** than a bad shelving location, but will have less sales than a good shelving location.