R Lab 4

Jared Andreatta

2025-04-01

3.6.4 Interaction terms

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.4.3
```

```
attach(Boston)

# Here is syntax for using interaction terms in a regression. We use lstat*age, which is
# shorthand for lstat+age+lstat:age, where lstat:age is the actual interaction term. This
# is an easy way to include interaction terms in a regression.
fit_interact <- lm(medv ~ lstat*age, data=Boston)

summary(fit_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
```

3.6.5 Nonlinear Transformations of Predictors

lm() function can accomodate nonlinear tranformations. Given X, we can create X^2 using the I() function.

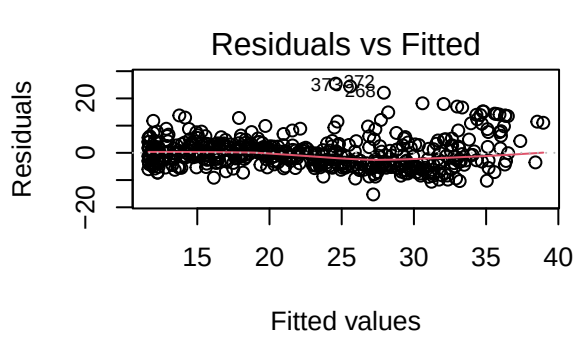
```
# Be sure to include normal lstat term
fit_nonlinear_transform <- lm(medv ~ lstat + I(lstat^2))
# Statistically significant of lstat^2 indicates that the quadratic term is an improvement
summary(fit_nonlinear_transform)
```

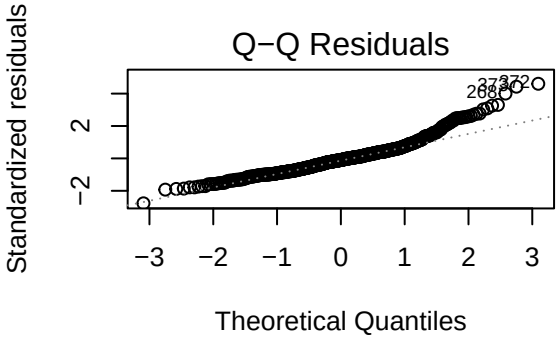
```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## 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
```

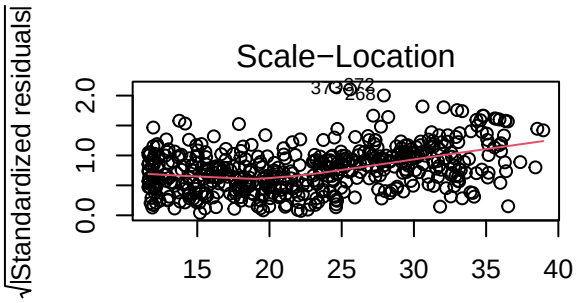
```
# Next, we use the anova() function
fit_lm <- lm(medv~lstat) # Regular linear model
# ANOVA table indicates that regression with quadratic term is superior
anova(fit_lm, fit_nonlinear_transform)
```

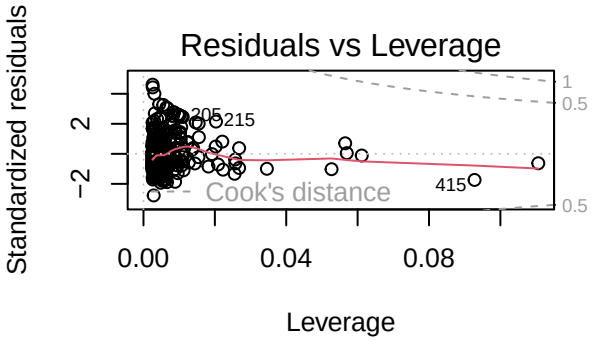
```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Plotting
par(mfrow = c(2, 2))
plot(fit_nonlinear_transform)
```










```
# Higher order polynomials #
# We can use the poly() function in our model for higher order polynomials'
fit_poly5 <- lm(medv ~ poly(lstat,5))
summary(fit_poly5) # Statistically significant for 5th degree polynomial estimation
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 5))
##
## 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
```

```
# We can also easily do a log tranform
fit_log <- lm(medv~log(rm))
summary(fit_log)
```

```
##
## Call:
## lm(formula = medv ~ log(rm))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.487  -2.875  -0.104   2.837  39.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -76.488      5.028  -15.21   <2e-16 ***
## log(rm)       54.055      2.739   19.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.915 on 504 degrees of freedom
## Multiple R-squared:  0.4358, Adjusted R-squared:  0.4347
## F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

3.6.6 Qualitative Predictors

```
# The Carseats dataset includes qualitative predictors
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
## 6 10.81       124    113          13        501    72       Bad  78        16
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

```
# Regression with all predictors and Income:Advertising and Price:Age interaction terms
lm.fit <- lm(Sales ~ . + Income:Advertising + Price:Age,
 data = Carseats)
summary(lm.fit)
```

```
##
## 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
```

```r
# contrasts generates dummy variables for qualitative vars
# This creates a "Good" variable if it is Good, and 0 if otherwise,
# a "Medium" variable that is 1 if medium, 0 if otherwise,
# and a bad shelving location corresponds to when both of these dummies are 0.
contrasts(Carseats$ShelveLoc)
```

```
##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```