Chapter 3

Wenduo Wang

July 20, 2016

Problem 15

(a) Let's create a simple linear regression model for each variable against `crim`, and plot the fitted line and actual data points.

```
library(MASS)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
summary(Boston)
```

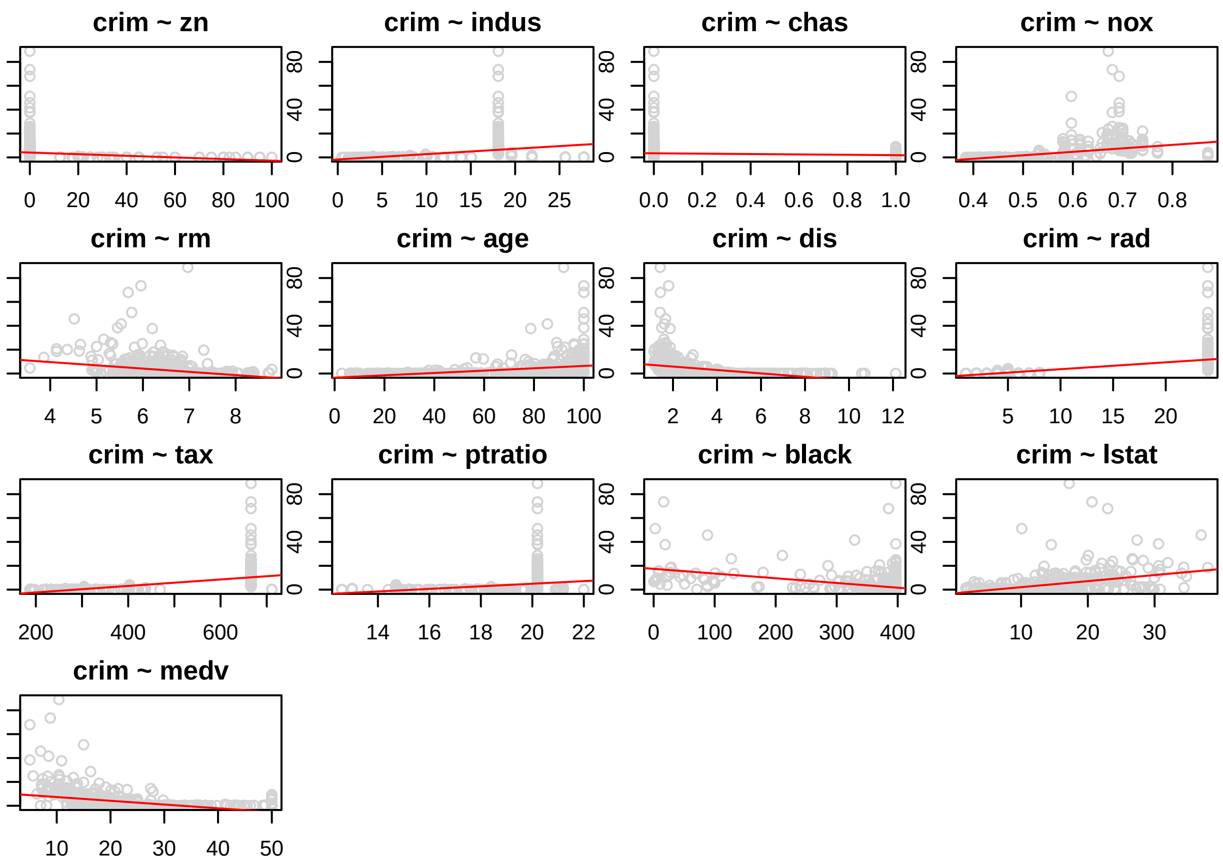
```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax            ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
```

```
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
row_no <- nrow(Boston)
training_rows <- sample(1:row_no, round(row_no*0.8))
training_set <- Boston[training_rows,]
test_set <- setdiff(Boston, training_set)
confint_df <- data.frame()
par(mfrow=c(4, 4), mar=c(2, 1, 2, 1))
for (i in c(1:ncol(Boston))){
    if (colnames(Boston)[i]=="crim"){
        next
    }
    formula <- as.formula(paste("crim~", colnames(Boston)[i]))
    lm_model <- lm(formula, data=training_set)
    plot(y=Boston$crim, x=Boston[,i], main=paste("crim ~", colnames(Boston)[i]), pch=1, col="lightgray"
    abline(lm_model$coef[1], lm_model$coef[2], col="red")
    confint_df <- rbind(confint_df, confint(lm_model)[2,])
}
print(dim(confint_df))
```

```
## [1] 13  2
```

```r
colnames(confint_df) <- c("2.5%", "97.5%")
row.names(confint_df) <- colnames(Boston)[colnames(Boston)!="crim"]
```



The linear model does not fit the data very well according to the plots. To further assess the correlation, let's have a closer look at the coefficients of these models

```
print(confint_df)
```

```
##                 2.5%       97.5%
## zn      -0.09978712 -0.03629300
## indus    0.35075153  0.54349442
## chas    -4.45344789  1.31968722
## nox     23.43371414 34.92301306
## rm      -3.76183094 -1.70335440
## age      0.07339008  0.12283516
## dis     -1.76437154 -1.10192356
## rad      0.51023851  0.63694442
## tax      0.02368028  0.03056465
## ptratio  0.75472376  1.41538217
## black   -0.04663265 -0.03230110
## lstat    0.41742728  0.60046478
## medv    -0.38693914 -0.24015501
```

The above code lists the 95% confindence intervals of coefficients for all the predictors with respect to each linear model. As seen from the result, the confident interval of the coefficient of `chas` contains zero, which means this predictor is probably not correlated with `crim`. Meanwhile, other predictors mostly have very small coefficients close to 0, with the exception of `nox`. So up to now, it appears `nox` is most likely to be a true predictor of `crim`.

(b, c) Now let's create a linear model of `crim` on all predictors, and see their coefficients' confident intervals.

```
lm_model_multiple <- lm(crim~., data=training_set)
print(confint(lm_model_multiple))
```

```
##                     2.5 %       97.5 %
## (Intercept)   9.404950972 35.702878811
## zn            0.001047215  0.068501244
## indus        -0.204294871  0.084131872
## chas         -2.940821917  1.285599365
## nox         -21.005655173 -1.877279566
## rm           -1.519116452  0.689520830
## age          -0.026899700  0.037253442
## dis          -1.346192070 -0.335770956
## rad           0.395640741  0.697757396
## tax          -0.012391003  0.005086045
## ptratio      -0.619402923  0.064379780
## black        -0.021762478 -0.008529096
## lstat         0.016897864  0.284913651
## medv         -0.203570814  0.014282966
```

When all the predictors are simultaneously fitted against `crim`, the result turns out very different from the previous single linear regression models. First, the previously strong predictor `nox` is now largely irrelevant. Actually in this case only `zn`, `dis` and `rad` show a significant correlation with `crim`, which is equivalent to rejecting the null hypothesis that H_0: β_j=0.

(d) To inspect if there is a correlation in the form of

$$crim = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$$

we create a linear regression model for each predictor against `crim`, in the format of `lm(crim ~ x + x^2 + x^3, data=training_set)`, and then inspect the model's coefficient confidence interval.

```r
for (i in c(1:ncol(Boston))){
    if (colnames(Boston)[i]=="crim"){
        next
    }
    variable <- colnames(Boston)[i]
    formula <- as.formula(paste("crim ~ ", variable, " + I(", variable, "^2) + I(", variable, "^3)", sep
    cat("For predictor:", colnames(Boston)[i], "\n")
    pn_model_3 <- lm(formula, data=training_set)
    print(confint(pn_model_3))
    cat("\n")
}
```

```
## For predictor: zn
##                     2.5 %        97.5 %
## (Intercept)  3.5533001006  5.197546e+00
## zn          -0.5196096588 -7.803779e-02
## I(zn^2)     -0.0021720212  1.391353e-02
## I(zn^3)     -0.0001014014  3.224474e-05
##
## For predictor: indus
##                    2.5 %       97.5 %
## (Intercept)  0.391330097  6.239078503
## indus       -2.685001506 -0.893511532
## I(indus^2)   0.155593977  0.301056126
## I(indus^3)  -0.008044261 -0.004544068
##
## For predictor: chas
##                 2.5 %   97.5 %
## (Intercept)  2.664197 4.182167
## chas        -4.453448 1.319687
## I(chas^2)          NA       NA
## I(chas^3)          NA       NA
##
## For predictor: nox
##                  2.5 %    97.5 %
## (Intercept)   147.6178  275.7463
## nox         -1486.9010 -838.2087
## I(nox^2)     1511.8891 2577.8705
## I(nox^3)    -1418.1884 -849.0716
##
## For predictor: rm
##                   2.5 %      97.5 %
## (Intercept) -80.7109547 170.5773557
## rm          -67.7797830  54.0186195
## I(rm^2)     -10.2243771   9.2529553
## I(rm^3)      -0.4307944   0.5945633
##
## For predictor: age
##                     2.5 %        97.5 %
## (Intercept) -8.631029e+00  2.3468435006
## age         -4.979534e-02  0.6847119371
## I(age^2)    -1.508205e-02 -0.0008160914
## I(age^3)     1.886983e-05  0.0001013002
##
```

```
## For predictor: dis
##                   2.5 %      97.5 %
## (Intercept)  21.5532006 31.00554751
## dis         -16.6257134 -9.98588028
## I(dis^2)      1.3996896  2.71442511
## I(dis^3)     -0.1360522 -0.05901924
##
## For predictor: rad
##                     2.5 %     97.5 %
## (Intercept) -4.509184613 3.26886254
## rad         -1.463976103 2.47426726
## I(rad^2)    -0.351821493 0.20504198
## I(rad^3)    -0.005450938 0.01161793
##
## For predictor: tax
##                     2.5 %       97.5 %
## (Intercept) -5.526624e-01 4.111674e+01
## tax         -3.333708e-01 5.064702e-03
## I(tax^2)    -3.548431e-05 8.222803e-04
## I(tax^3)    -5.856848e-07 8.185100e-08
##
## For predictor: ptratio
##                     2.5 %        97.5 %
## (Intercept)    84.4322179 714.020219126
## ptratio      -123.8048203 -12.790916447
## I(ptratio^2)    0.5742606   7.035196057
## I(ptratio^3)   -0.1308020  -0.006619314
##
## For predictor: black
##                     2.5 %        97.5 %
## (Intercept)  1.597005e+01  2.461270e+01
## black       -2.256656e-01 -1.168007e-02
## I(black^2)  -2.077354e-04  9.265579e-04
## I(black^3)  -1.275102e-06  3.794562e-07
##
## For predictor: lstat
##                     2.5 %       97.5 %
## (Intercept) -3.191763274 4.6627000338
## lstat       -1.162318655 0.6690502389
## I(lstat^2)  -0.023840354 0.0964932535
## I(lstat^3)  -0.001577993 0.0007024436
##
## For predictor: medv
##                     2.5 %        97.5 %
## (Intercept) 40.980170226 54.3410554538
## medv        -5.352398110 -3.6473551239
## I(medv^2)    0.102337286  0.1693242916
## I(medv^3)   -0.001685639 -0.0008953778
```

As seen from the output confidence intervals, it is first noticed that `chas^2` and `chas^3` do not have coefficients. That is due to that `chas` is a binary variable, whose squre or cube is essentially itself, so in this case `chas`, `chas^2` and `chas^3` are linearly related, and therefore the latter two polynomial terms are not fitted in the model. `chas` put aside, several polynomial terms exhibit correlation with `crim`, whose 95% coefficient confidence intervals exclude zero. For example, `nox^2` and `nox^3`.