Stat 897 Spring 2017 Data Analysis Assignment 3

Penn State

Due September 10, 2017

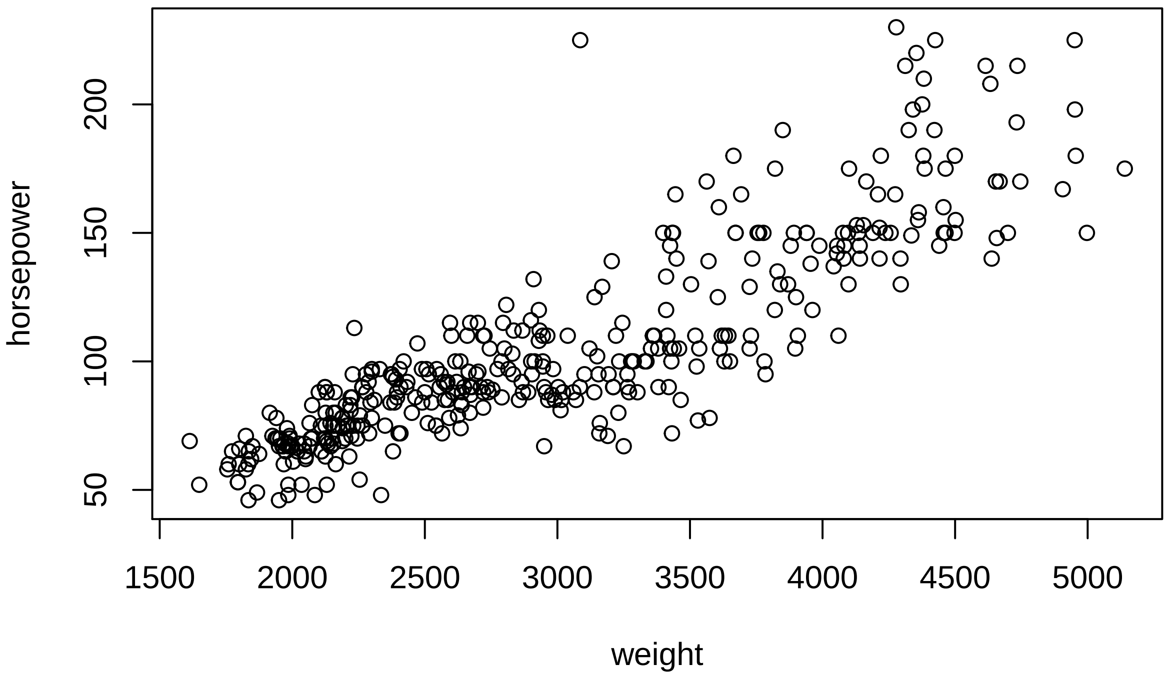
The goal of this DA assignment will be to familiarize you with linear models and assessing models, as well as beginning to think about model selection.

1. (a) Using the dataset `Auto` from the ISLR package produce simple summaries of the variables in the data. Plot a couple variables against each other where you think one may be a good predictor of the other.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.2.5
```

```
attach(Auto)
plot(weight, horsepower)
```



(b) Fit a simple linear model using the two variables you plotted and produce a summary of the model.

```
simple_lm = lm(horsepower ~ weight)
summary(simple_lm)
```

```
##
## Call:
## lm(formula = horsepower ~ weight)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.272 -12.285  -0.557   9.063 116.283
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.183485   3.570431  -3.412 0.000712 ***
## weight        0.039177   0.001153  33.972  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.37 on 390 degrees of freedom
## Multiple R-squared:  0.7474, Adjusted R-squared:  0.7468
## F-statistic:  1154 on 1 and 390 DF,  p-value: < 2.2e-16
```

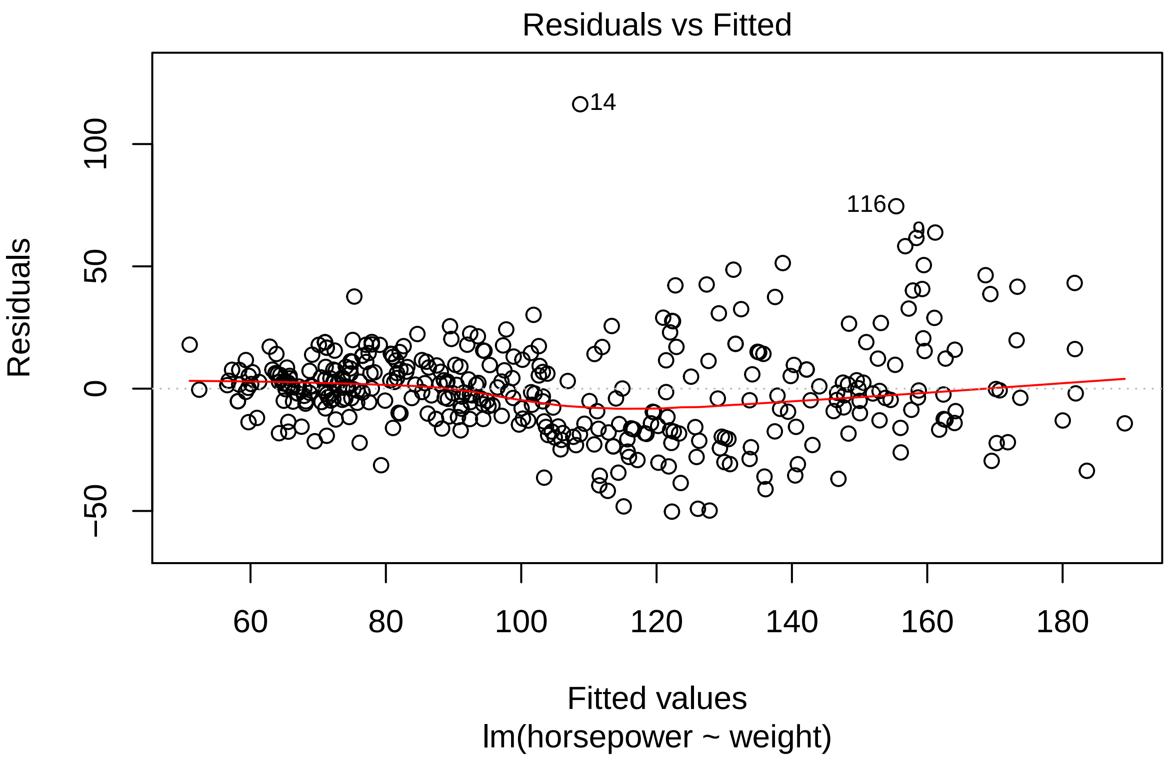
(c) Give a 95% confidence interval for the coefficients. Do you think variables are related? Why or why not?

```r
confint(simple_lm)
```

```
##                    2.5 %      97.5 %
## (Intercept) -19.20318628 -5.16378313
## weight        0.03690973  0.04144431
```

(d) Plot the residuals from your model against the fitted values and comment on anything that looks unusual. (Hint: use the plot.lm function with which = 1.)

```r
plot(simple_lm, which = 1)
```



There seem to be a few outliers and non-constant variance of the residuals.

(e) How might you improve your model? (e.g. transformation or addition of a variable).

Add cylinders as another predictor, since I would expect this to also be related to the horsepower

2. (a) Load (or install) the `mlbench` library and upload the `BostonHousing` data. Produce a summary of the variable `medv`.

```
library(mlbench)
data("BostonHousing")
summary(BostonHousing$medv)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   17.02   21.20   22.53   25.00   50.00
```

(b) Using `medv` as your response variable fit a linear regression model with all othe variables as predictors. Compute the training MSE.

```
set.seed(10412)
housing_lm = glm(medv ~ ., data = BostonHousing)
mean(residuals(housing_lm) ^ 2)
```

```
## [1] 21.89483
```

(c) Now find a good model for predicting `medv`. Explain your process in choosing the model and why it is a good prediction model. Feel free to use any number of the other variables in the data as predictors.

```r
## First let's look at the linear model using all the predictors.
summary(housing_lm)
```

```
##
## Call:
## glm(formula = medv ~ ., data = BostonHousing)
##
## Deviance 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
## chas1        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 ***
## b            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
##
## (Dispersion parameter for gaussian family taken to be 22.51785)
##
##     Null deviance: 42716  on 505  degrees of freedom
## Residual deviance: 11079  on 492  degrees of freedom
## AIC: 3027.6
##
## Number of Fisher Scoring iterations: 2
```

```r
## check AIC
housing_lm$aic
```

```
## [1] 3027.609
```

```r
## check test MSE
library(boot)
```

```
## Warning: package 'boot' was built under R version 3.2.3
```

```r
cv.glm(BostonHousing, housing_lm, K = 10)$delta[1]
```

```
## [1] 23.62048
```

```r
## let's try removing indus and age as predictors
housing_lm2 = glm(medv ~ . -(age + indus), data = BostonHousing)
summary(housing_lm2)
```

```
##
## Call:
## glm(formula = medv ~ . - (age + indus), data = BostonHousing)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -15.5984   -2.7386   -0.5046    1.7273   26.2373
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## zn            0.045845   0.013523   3.390 0.000754 ***
## chas1         2.718716   0.854240   3.183 0.001551 **
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## b             0.009291   0.002674   3.475 0.000557 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 22.43191)
##
##     Null deviance: 42716  on 505  degrees of freedom
## Residual deviance: 11081  on 494  degrees of freedom
## AIC: 3023.7
##
## Number of Fisher Scoring iterations: 2
```

```r
## check R-squared
housing_lm2$aic
```

```
## [1] 3023.726
```

```r
## check test MSE
cv.glm(BostonHousing, housing_lm2, K = 10)$delta[1]
```

```
## [1] 23.5062
```

The second model improves on the first both in AIC and test MSE.

```r
## now let's try removing crim and chas as predictors
housing_lm3 = glm(medv ~ . -(age + indus + crim + chas), data = BostonHousing)
summary(housing_lm3)
```

```
##
## Call:
## glm(formula = medv ~ . - (age + indus + crim + chas), data = BostonHousing)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -12.8917   -2.7329   -0.4988    1.8547   26.6433
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.459724   5.158054   6.875 1.87e-11 ***
## zn            0.041396   0.013737   3.013 0.002715 **
## nox         -15.502932   3.583879  -4.326 1.84e-05 ***
## rm            3.879580   0.414180   9.367  < 2e-16 ***
## dis          -1.451648   0.187926  -7.725 6.26e-14 ***
## rad           0.252412   0.061778   4.086 5.12e-05 ***
## tax          -0.012360   0.003427  -3.606 0.000342 ***
## ptratio      -0.968703   0.131248  -7.381 6.69e-13 ***
## b             0.010842   0.002705   4.008 7.06e-05 ***
## lstat        -0.555124   0.047699 -11.638  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 23.3487)
##
##     Null deviance: 42716  on 505  degrees of freedom
## Residual deviance: 11581  on 496  degrees of freedom
## AIC: 3042
##
## Number of Fisher Scoring iterations: 2
```

```r
## check R-squared
housing_lm3$aic
```

```
## [1] 3042.039
```

```r
## check test MSE
cv.glm(BostonHousing, housing_lm3, K = 10)$delta[1]
```

```
## [1] 23.94409
```

The third model performs worse than the second model on both AIC and test MSE, so we will choose the second model as our model.

(d) Now let's briefly consider a comparison of neural networks with linear regression. We will use the `nnet` package and the function `nnet`. Using the same data fit a neural network with `medv` as response and all other variables as predictors. (Note: for the neural network you need to scale the response variable so that all values are between 0 and 1. An easy way to do this is by dividing by the maximum value. When you predict values remember to restore the original scale.) Compute the training MSE and compare it to the MSE from part (b).

```r
##
## model neural network
##
require(nnet)
```

```
## Loading required package: nnet
```

```
## Warning: package 'nnet' was built under R version 3.2.3
```

```r
# scale inputs: divide by 50 to get 0-1 range
nnet_fit = nnet(medv / 50 ~ ., data = BostonHousing, size = 2, decay = 5e-04)
```

```
## # weights:  31
## initial  value 19.587375
## iter  10 value 17.087482
## iter  20 value 16.690149
## iter  30 value 13.415803
## iter  40 value 9.997862
## iter  50 value 7.613471
## iter  60 value 5.488038
## iter  70 value 4.065195
## iter  80 value 3.567584
## iter  90 value 3.421313
## iter 100 value 3.329528
## final  value 3.329528
## stopped after 100 iterations
```

```r
# multiply 50 to restore original scale
nnet_predict = predict(nnet_fit) * 50

# mean squared error
mean((nnet_predict - BostonHousing$medv) ^ 2)
```

```
## [1] 16.20199
```

The training error for the neural net is significantly lower than using the linear regression.

Using the same model you chose in part (c), fit a neural network. Compare the training MSEs between the two.

```r
nnet_fit2 = nnet(medv / 50 ~ . -(age + indus), data = BostonHousing, size = 2, decay = 5e-04)
```

```
## # weights:  27
## initial  value 19.946635
## iter  10 value 15.178243
## iter  20 value 14.623152
## iter  30 value 13.613098
## iter  40 value 11.696458
## iter  50 value 9.000962
## iter  60 value 6.237568
## iter  70 value 4.846673
## iter  80 value 3.757777
## iter  90 value 3.637886
## iter 100 value 3.554981
## final  value 3.554981
## stopped after 100 iterations
```

```r
nnet_predict2 = predict(nnet_fit2) * 50

mean((nnet_predict2 - BostonHousing$medv) ^ 2)
```

```
## [1] 17.45821
```

The training error from the neural net is still lower than from the linear model, though it is higher than the other neural network.

Finally submit BOTH your .rmd file and the resulting .pdf file with Canvas as Data Analysis Assignment 3.