DaigleHomework1.R

daiglechris

Sun Sep 23 15:23:43 2018

Chris Daigle

Homework 1

Exercise: 2

```
#
# 1 ####
# Download the housing dataset from https://www.kaggle.com/harlfoxem/housesalesprediction and run a reg

rm(list = ls())

setwd('~/Git/MachineLearningAndBigDataWithR/Data')

# dataurl  = "https://www.kaggle.com/harlfoxem/housesalesprediction/downloads/kc_house_data.csv/1"
dataName <- 'kcHouseData.csv'
# download.file(dataurl, destfile = dataName)
data <- read.csv(dataName, stringsAsFactors = FALSE)
str(data)
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

```
# price: price of home
# bedrooms: number of bedrooms
# floors: number of floors
# sqft_living: square footage of living space
```

```r
# yr_built

lm.fit <-
  lm(price ~ sqft_living + bedrooms + floors + yr_built, data = data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + bedrooms + floors + yr_built,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1857482  -130242   -16610    99627  3972920
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.810e+06  1.285e+05   45.21   <2e-16 ***
## sqft_living  3.304e+02  2.387e+00  138.42   <2e-16 ***
## bedrooms    -5.937e+04  2.211e+03  -26.86   <2e-16 ***
## floors       7.283e+04  3.672e+03   19.84   <2e-16 ***
## yr_built    -2.976e+03  6.658e+01  -44.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 246700 on 21608 degrees of freedom
## Multiple R-squared:  0.5486, Adjusted R-squared:  0.5485
## F-statistic:  6564 on 4 and 21608 DF,  p-value: < 2.2e-16
```

```r
names(lm.fit)
```

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

```r
# 2 ####
# Build a model to predict the housing price given characteristics of a house in the dataset.
# Consider the following predictors: season, sqft_living, yr_built, interaction of sqft_living yr_built
#     Create a variable "season" which equals
#     "Winter" if a house was sold in Jan, Feb, Mar, Dec.
#     "Spring" if it was sold in Apr, May, Jun
#     "Summer" if it was sold in Jul, Aug
#     "Fall" if it was sold in Sep, Oct, Nov.
# Do you find any seasonality in housing price?

data <-
  transform(
    data,
    Year = substr(date, 1, 4),
    Month = substr(date, 5, 6),
    Day = substr(date, 7, 8)
  )

data$cleanDate <-
  as.Date(paste0(data$Year, '-', data$Month, '-', data$Day))
```

```r
data$season[data$Month == '01' |
              data$Month == '02' |
              data$Month == '03' | data$Month == '12'] <- 'Winter'
data$season[data$Month == '04' |
              data$Month == '05' | data$Month == '06'] <- 'Spring'
data$season[data$Month == '07' | data$Month == '08'] <- 'Summer'
data$season[data$Month == '09' |
              data$Month == '10' | data$Month == '11'] <- 'Fall'

lm.fit1 <-
  lm(price ~ season + sqft_living + yr_built + sqft_living * yr_built + waterfront,
     data = data)
summary(lm.fit1)
```

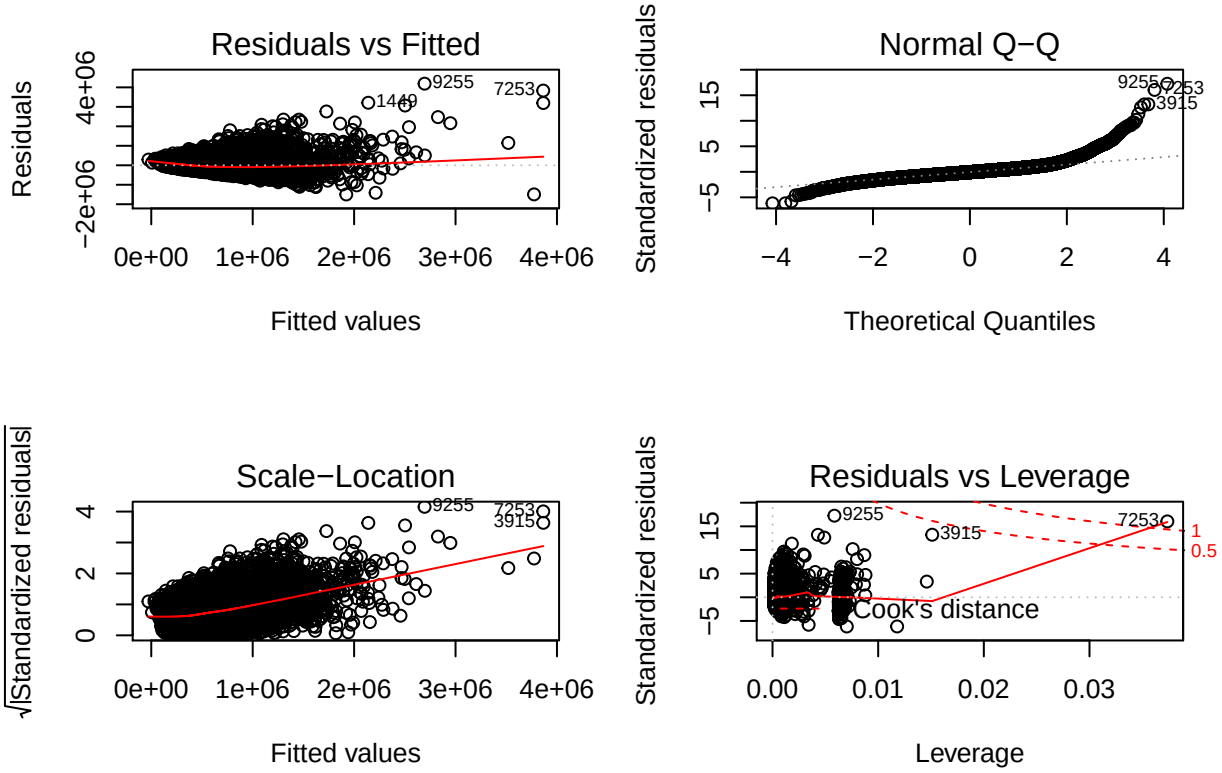
```
##
## Call:
## lm(formula = price ~ season + sqft_living + yr_built + sqft_living *
##     yr_built + waterfront, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1497733  -133514   -17225   107931  4190203
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.744e+06  2.686e+05  10.216  < 2e-16 ***
## seasonSpring            2.182e+04  4.519e+03   4.828 1.39e-06 ***
## seasonSummer            5.037e+03  5.102e+03   0.987    0.324
## seasonWinter            7.067e+03  4.730e+03   1.494    0.135
## sqft_living             1.055e+03  1.222e+02   8.632  < 2e-16 ***
## yr_built               -1.438e+03  1.364e+02 -10.543  < 2e-16 ***
## waterfront              7.828e+05  1.930e+04  40.556  < 2e-16 ***
## sqft_living:yr_built -3.841e-01  6.179e-02  -6.216 5.20e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 243600 on 21605 degrees of freedom
## Multiple R-squared:  0.5598, Adjusted R-squared:  0.5597
## F-statistic:  3925 on 7 and 21605 DF,  p-value: < 2.2e-16
```

```r
# Results
# Spring: 2.182e+04 = 2.182*(10^4) = 21820
# Summer: 5.037e+03 = 5.037*(10^3) = 5037
# Winter: 7.067e+03 = 7.067*(10^3) = 7067
# sqft_living: 1.055e+03 = 1.055*(10^3) = 1055
# yr_built: -1.438e+03 = -1.438*(10^3) = -1438
# waterfront: 7.828e+05 = 7.828*(10^5) = 782800
# sqft_living*yr_built: -3.841-e01 = -3.841*(10^(-1)) = -0.3841

# I find seasonality in housing price - it can be seen that selling in different
# seasons contributes differently to the housing price with Spring showing
# economically significant results ($21,820) as well as statistically
# significant ones (approximately zero p-value).
```

```
# 3 #### Do you find any nonlinearity or heteroskedasticity? YES. BREUSCH PAGAN
# TEST and NOTE FUNNEL LIKE RESIDUALS
#What is the problem if the error term is heteroskedastic? PREDICTION
#How can you address these problems (if you have here)?
# TRANSFORM DEPENDENT/RESPONSE VARIABLE, WHITE STANDARD ERRORS

par(mfrow = c(2, 2))
plot(lm.fit1)
```




```
par(mfrow = c(1, 1))
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(lm.fit1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm.fit1
## BP = 2882.7, df = 7, p-value < 2.2e-16
```

```
# The Breusch-Pagan test reveals that the null hypothesis of homoskedasticity is
# rejected, the p-value is 2.2e-16 (or incredbly near zero). We can also see the
```

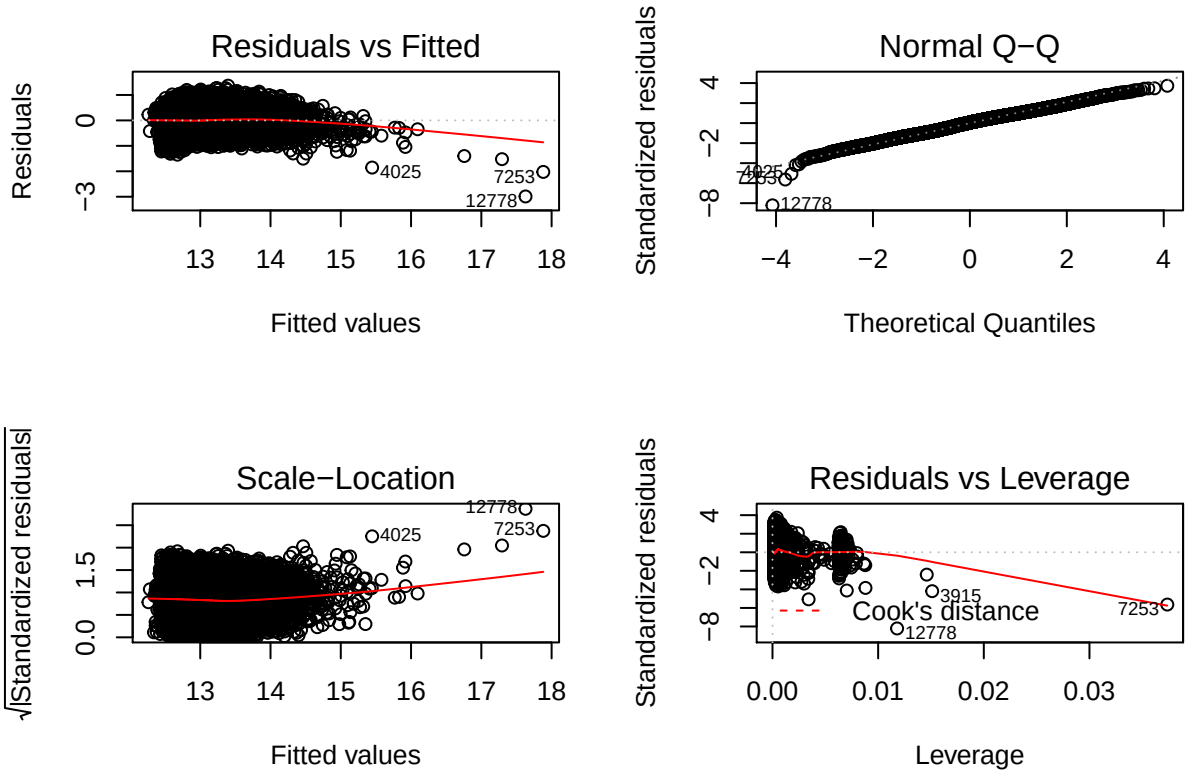
```r
# presence of a funnel type shape in the residuals plot and we can see the
# magnitude of the residuals tends to increase with the fitted values.

# Our ability to predict is diminished because the the variances of the error
# terms are non-constant, or put differently, the magnitude of the error terms
# changes with values of the independent variable(s).

library(sandwich)
lm.fit2 <-
  lm(log(price) ~ season + sqft_living + yr_built + sqft_living * yr_built + waterfront,
     data = data)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = log(price) ~ season + sqft_living + yr_built + sqft_living *
##     yr_built + waterfront, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.98925 -0.26650  0.02222  0.25836  1.36546
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.457e+01  4.036e-01  36.095  < 2e-16 ***
## seasonSpring            4.374e-02  6.792e-03   6.440 1.22e-10 ***
## seasonSummer            1.388e-02  7.668e-03   1.810   0.0704 .
## seasonWinter            5.554e-03  7.110e-03   0.781   0.4347
## sqft_living             1.875e-03  1.836e-04  10.210  < 2e-16 ***
## yr_built               -1.228e-03  2.050e-04  -5.990 2.13e-09 ***
## waterfront              5.709e-01  2.901e-02  19.679  < 2e-16 ***
## sqft_living:yr_built -7.358e-07  9.287e-08  -7.923 2.44e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3662 on 21605 degrees of freedom
## Multiple R-squared:  0.5169, Adjusted R-squared:  0.5167
## F-statistic:  3302 on 7 and 21605 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2, 2))
plot(lm.fit2)
```




```
par(mfrow = c(1, 1))
bptest(lm.fit2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm.fit2
## BP = 405.83, df = 7, p-value < 2.2e-16
```

```
# Employing the White Standard Errors
coeftest(lm.fit2, vcov = vcovHC(lm.fit2, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                           Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)            1.4569e+01  5.8635e-01 24.8476 < 2.2e-16 ***
## seasonSpring           4.3740e-02  6.7353e-03  6.4941 8.533e-11 ***
## seasonSummer           1.3876e-02  7.4945e-03  1.8515   0.06411 .
## seasonWinter           5.5539e-03  7.1871e-03  0.7728   0.43968
## sqft_living            1.8749e-03  2.9145e-04  6.4329 1.278e-10 ***
## yr_built              -1.2281e-03  2.9739e-04 -4.1296 3.647e-05 ***
## waterfront             5.7092e-01  3.4235e-02 16.6764 < 2.2e-16 ***
## sqft_living:yr_built -7.3577e-07  1.4719e-07 -4.9988 5.815e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(lm.fit1, vcov = vcovHC(lm.fit1, type = "HC1"))
```

```
##
## t test of coefficients:
```

```
##
##                           Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)           2.7436e+06  8.2806e+05  3.3133 0.0009234 ***
## seasonSpring          2.1817e+04  4.6100e+03  4.7325 2.231e-06 ***
## seasonSummer          5.0372e+03  5.0463e+03  0.9982 0.3181896
## seasonWinter          7.0668e+03  4.7111e+03  1.5000 0.1336213
## sqft_living           1.0546e+03  4.5669e+02  2.3092 0.0209438 *
## yr_built             -1.4382e+03  4.1915e+02 -3.4313 0.0006019 ***
## waterfront            7.8283e+05  5.8637e+04 13.3504 < 2.2e-16 ***
## sqft_living:yr_built -3.8407e-01  2.3043e-01 -1.6668 0.0955699 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# We can transform the response variable, Y, by predicting on the square-root
# or, more commonly employed because of ease of intepretation, logarithmic which
# demonstrates an elasticity (the responsiveness of the variable to the
# independent variables). Applying the log transform diminishes the amount of
# funneling as described in the residuals, but it is not completely gone. We can
# see from the Breusch-Pagan test that the null hypothesis of homoskedasticity
# continues to be rejected. We can also use White Standard Errors.

# 4 ####
# Conduct an F test for the following hypotheses.
# H0: there is no seasonality on the housing price. H1: H0 is not true.
# THE NULL HYPOTHESIS FAILS TO HOLD

yHatU <-
  lm(price ~ season + sqft_living + yr_built + sqft_living * yr_built + waterfront,
     data = data)
yHatR <-
  lm(price ~ sqft_living + yr_built + sqft_living * yr_built + waterfront,
     data = data)
ssr_u <- sum((fitted(yHatU) - data$price) ^ 2)
ssr_r <- sum((fitted(yHatR) - data$price) ^ 2)
f <- ((ssr_r - ssr_u) / 3) / (ssr_u / (21613 - 5 - 1))
var.test(yHatU, yHatR, alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
## data:  yHatU and yHatR
## F = 0.99889, num df = 21605, denom df = 21608, p-value = 0.9348
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9726002 1.0258839
## sample estimates:
## ratio of variances
##          0.9988868
```

```
# Min's solution
# SSR <- sum((yHatU$residuals)^2)
# SSRR <- sum((yHatR$residuals)^2)
# FSTAT <- ((SSRR - SSR)/3)/(SSR/(dim(data)[1] - 7 - 1))
# qf(0.95, df1 = 3, df2 = dim(data)[1] - 7 - 1)
```

```r
# 5 ####
# Predict the housing price when season = spring, sqft_living=2500, yr_built=2000, waterfront=0:
# PREDICTION RESULT: $594406.1
# 95% CONFIDENCE INTERVAL: ($589801.9, $599010.4)
lm.fit <- lm(price ~ sqft_living + yr_built + waterfront, data = data)
predict(
  object = lm.fit,
  newdata = data.frame(
    season = "Spring",
    sqft_living = 2500,
    yr_built = 2000,
    waterfront = 0
  ),
  interval = "prediction"
)
```

```
##        fit      lwr     upr
## 1 594406.1 116197.2 1072615
```

```r
predict(
  object = lm.fit,
  newdata = data.frame(
    season = "Spring",
    sqft_living = 2500,
    yr_built = 2000,
    waterfront = 0
  ),
  interval = "confidence"
)
```

```
##        fit      lwr      upr
## 1 594406.1 589801.9 599010.4
```

```r
predict(
  object = lm.fit,
  newdata = data.frame(
    season = "Spring",
    sqft_living = 2500,
    yr_built = 2000,
    waterfront = 0
  )
)
```

```
##        1
## 594406.1
```