kaggle Search Competitions Datasets Kernels Discussion Learn

In [1]:

```
## Importing packages

# This R environment comes with all of CRAN and many other helpful packages preinstalled.
# You can see which packages are installed by checking out the kaggle/rstats docker image:
# https://github.com/kaggle/docker-rstats

library(tidyverse) # metapackage with lots of helpful functions

## Running code

# In a notebook, you can run a single code cell by clicking in the cell and then hitting
# the blue arrow to the left, or by clicking in the cell and pressing Shift+Enter. In a script,
# you can run code by highlighting the code you want to run and then clicking the blue arrow
# at the bottom of this window.

## Reading in files

# You can access files from datasets you've added to this kernel in the "../input/" directory.
# You can see the files added to this kernel by running the code below.

list.files(path = "../input")

## Saving data

# If you save any files or images, these will be put in the "output" directory. You
# can see the output directory by com
# can see the output directory by com
mitting and running your kernel (usin
g the
# Commit & Run button) and then check
ing out the compiled version of your
 kernel.
```

```
── Attaching packages ──
────────────────────────
───────────── tidyverse
1.2.1 ──
✔ ggplot2 3.0.0.9000
✔ purrr   0.2.5
✔ tibble  1.4.2
✔ dplyr   0.7.6
✔ tidyr   0.8.1
✔ stringr 1.3.1
✔ readr   1.2.0
✔ forcats 0.3.0
── Conflicts ───────────
────────────────────────
─────── tidyverse_confli
cts() ──
✖ dplyr::filter() masks
stats::filter()
✖ dplyr::lag()    masks
stats::lag()
```

In [2]:

```
#
# https://youtu.be/5ONFqIk3RFg
#
```

In [3]:

```
#
# MASS: Support Functions and Dataset
s for Venables and Ripley's MASS
# ISLR: Data for an Introduction to S
tatistical Learning with Applications
 in R
#
```

```
library(MASS)
library(ISLR)
```

```
Attaching package: ‘MAS
S’

The following object is
masked from ‘package:dpl
yr’:

    select
```

In [4]:

```
# see name of the variables
names(Boston)
```

'crim' 'zn' 'indus' 'chas' 'nox'
'rm' 'age' 'dis' 'rad' 'tax'
'ptratio' 'black' 'lstat' 'medv'

In [5]:

```
# And so if you want more detail, yo
u're going to ask
# for help on Boston.
?Boston
# It's got 506 rows and 14 columns.
```

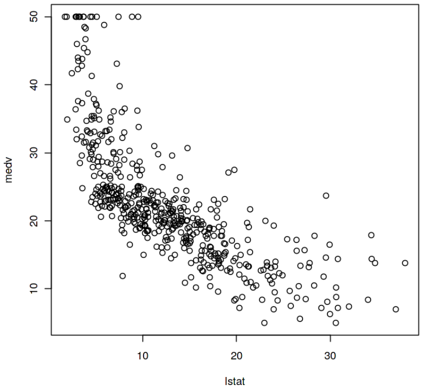
In [6]:

```
### Simple linear regression
```

In [7]:

```
# plot some variables: medv and lstat
# the response is medv (on vertical a
xis)
# find these variables in data set Bo
ston
```

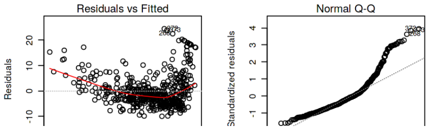
```
# plot(Boston$lstat,Boston$medv)
# or
plot(medv~lstat,Boston)
```

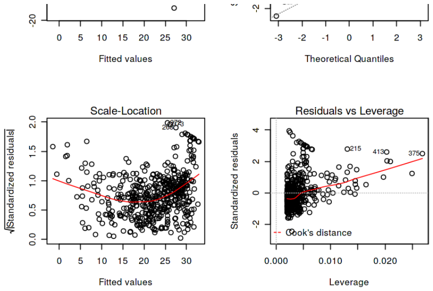


In [8]:

```
# medv : response
# ~ : is modeled as
# lstat: single predictor
fit1=lm(medv~lstat,data=Boston)
fit1
# you can see its a negative relation
ship
# it gives you a brif summary

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

```
Call:
lm(formula = medv ~ lsta
t, data = Boston)

Coefficients:
(Intercept)        lstat

      34.55        -0.95
```





In [9]:

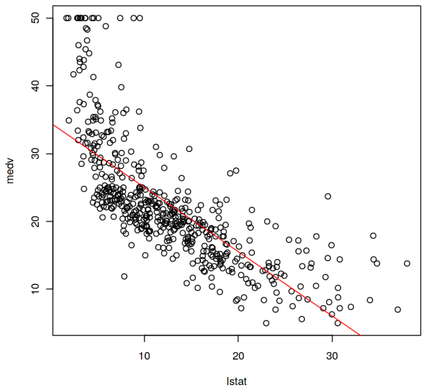
```
# get more detail of summary
summary(fit1)
# both of intercept and lstat are sig
nificant
```

```
Call:
lm(formula = medv ~ lsta
t, data = Boston)

Residuals:
    Min      1Q  Median
     3Q     Max
-15.168  -3.990  -1.318
  2.034  24.500

Coefficients:
            Estimate St
d. 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:  0.5
```

```
441,    Adjusted R-squar
ed:  0.5432
F-statistic: 601.6 on 1
and 504 DF,  p-value: <
2.2e-16
```

In [10]:

```
# add linear model line to the plot
# need to plot before abline
plot(medv~lstat,Boston)
abline(fit1,col="red")
#?abline
```



In [11]:

```
names(fit1)
```

'coefficients' 'residuals' 'effects'
'rank' 'fitted.values' 'assign' 'qr'
'df.residual' 'xlevels' 'call' 'terms'
'model'

In [12]:

```
# find the confident interval for the
 fit
confint(fit1)
#?confint
```

	2.5 %	97.5 %
(Intercept)	33.448457	35.6592247
lstat	-1.026148	-0.8739505

In [13]:

```
# The predict function is another one of these methods
# where we can use to query a linear model fit.
# In this case, we're going to predict with three new values
# for lstat, or three particular values, five, 10, and 15.
# 3 And we're going to not only ask for predictions, we're
# going to ask for a confidence interval.
# So those are additional arguments to predict.
predict(fit1,data.frame(lstat=c(5,10,15)),interval="confidence")
# And when we do that, we get the fit at those three values,
# and then the lower confidence interval, and the upper
# confidence band.

# https://stackoverflow.com/questions/38109501/how-does-predict-lm-compute-confidence-interval-and-prediction-interval
```

	fit	lwr	upr
1	29.80359	29.00741	30.59978
2	25.05335	24.47413	25.63256
3	20.30310	19.73159	20.87461

3.R_Linear Regression in R

R notebook using data from no data sources · 5 views ·

Edit tags

0

Access

Edit

```
### Multiple linear regression
```

Version 5
5 commits

Notebook

Data

Output

Log

Comments

In [15]:

```
# we wanna fit lstat and age
# we seperate variables with +
fit2=lm(medv~lstat+age,data=Boston)
summary(fit2)
# And age is also significant, quite
 strongly so, but not as significant
 lstat.
# One of the things down below is the
 r squared, which we talked about as
 well, for the model.
# Remember, r squared, it's the highe
r the better. It's a percentage of va
riance explained.
```

```
Call:
lm(formula = medv ~ lsta
t + age, data = Boston)

Residuals:
    Min      1Q  Median
     3Q     Max
-15.981  -3.978  -1.283
  1.968  23.158

Coefficients:
            Estimate St
d. 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
**
```

Notebook Data Output Log Comments

```
' ' 0.1 ' ' 1
```

```
Residual standard error:
6.173 on 503 degrees of
freedom
Multiple R-squared:  0.5
513,    Adjusted R-squar
ed:  0.5495
F-statistic:   309 on 2
and 503 DF,  p-value: <
2.2e-16
```

In [16]:

```
# And ~. means is that we're supposed
 to use all
# the other variables in the Boston d
ata frame except medv,
# which is the response, and all the
 others will be predictors.
fit3=lm(medv~.,Boston)
summary(fit3)
# Age, now, is no longer significant.
# So age, when it was in the model ju
st with lstat, was
# significant. But now it's in the mo
del with all these other predictors.
# And it's no longer significant. Wha
t that means is there's basically a l
ot of other
# predictors that are very correlated
 with age.
# And in the presence of them, age is
 no longer required.
```

```
Call:
lm(formula = medv ~ ., d
ata = Boston)

Residuals:
    Min      1Q  Median
     3Q     Max
-15.595  -2.730  -0.518
  1.777  26.199
```

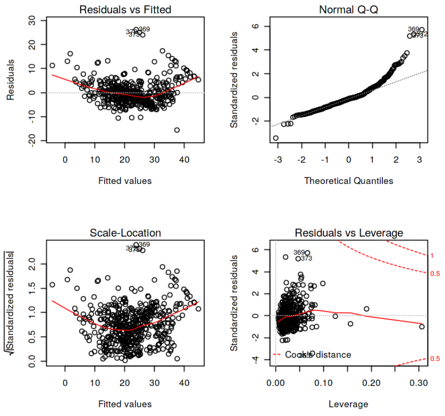
```
Coefficients:
              Estimate S
td. Error t value Pr(>|t
|)
(Intercept)  3.646e+01
5.103e+00   7.144 3.28e-
12 ***
crim        -1.080e-01
3.286e-02  -3.287 0.0010
87 **
zn           4.642e-02
1.373e-02   3.382 0.0007
78 ***
indus        2.056e-02
6.150e-02   0.334 0.7382
88
chas         2.687e+00
8.616e-01   3.118 0.0019
25 **
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 0.9582
29
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 0.0011
12 **
ptratio     -9.527e-01
1.308e-01  -7.283 1.31e-
12 ***
black        9.312e-03
2.686e-03   3.467 0.0005
73 ***
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:  0.7
406,    Adjusted R-squar
ed:  0.7338
F-statistic: 108.1 on 13
and 492 DF,  p-value: <
2.2e-16
```

In [17]:

```
# You can plot linear models.
# I made a two by two layout, because
 I know that four plots
par(mfrow=c(2,2))
plot(fit3)

# The first one is the residuals agai
nst the fitted values.
# The vector fitted values is just a
 single vector.
# So we can plot the residuals agains
t that.
# And the reason we do that is we are
 looking for non-linearities.
# And we kind of know there's a non-l
inearity in this one.
# We saw that in the very first plot.
 And by the curve in the residuals he
re, we can see
# that the model is not quite capturi
ng everything that's going on.
# There seems to be some non-linearit
y.

# This lower left one, is the square
 root of the absolute standardized re
siduals.
```

```
# One plots this to see, perhaps, if the variance is changing with the mean or the fit.
# In this case, it looks like there may be some relationship there.
# But that could be a result of a non-linearity that we seem
# to have missed in the model.
```



In [18]:

```
# Twiddle means--nothing on the left means we're going to
# use the same response,
# . means whatever the model was in fit3, That's replaced in dot.
# And minus age means we want to remove age.
# And minus indus, we want to remove indus as well.
# So this will fit the model with those two variables removed, all the others in.
fit4=update(fit3,~.-age-indus)
summary(fit4)
# And now everything that's left in the model appears to be significant.
```

```
Call:
lm(formula = medv ~ crim
+ zn + chas + nox + rm +
dis + rad +
    tax + ptratio + blac
k + lstat, data = Bosto
n)

Residuals:
     Min       1Q   Medi
an       3Q      Max
-15.5984  -2.7386  -0.50
46   1.7273  26.2373

Coefficients:
              Estimate S
td. Error t value Pr(>|t
|)
(Intercept)  36.341145
5.067492   7.171 2.73e-1
2 ***
crim         -0.108413
0.032779  -3.307 0.00101
0 **
zn            0.045845
0.013523   3.390 0.00075
4 ***
chas          2.718716
0.854240   3.183 0.00155
1 **
nox         -17.376023
3.535243  -4.915 1.21e-0
6 ***
rm            3.801579
0.406316   9.356  < 2e-1
6 ***
dis          -1.492711
0.185731  -8.037 6.84e-1
5 ***
rad           0.299608
0.063402   4.726 3.00e-0
6 ***
tax          -0.011778
0.003372  -3.493 0.00052
1 ***
ptratio      -0.946525
0.129066  -7.334 9.24e-1
3 ***
black         0.009291
0.002674   3.475 0.00055
```

```
7 ***
lstat        -0.522553
0.047424 -11.019  < 2e-1
6 ***
---
Signif. codes:  0 ‘***’
0.001 ‘**’ 0.01 ‘*’ 0.05
‘.’ 0.1 ‘ ’ 1

Residual standard error:
4.736 on 494 degrees of
freedom
Multiple R-squared:  0.7
406,    Adjusted R-squar
ed:  0.7348
F-statistic: 128.2 on 11
and 494 DF,  p-value: <
2.2e-16
```

In [19]:

```
# ### Nonlinear terms and Interaction
s
```

In [20]:

```
# The first thing we'll do is make a
 fit where we put an interaction
# between lstat and age. And that we
 do with a star, sort of like multipl
y.
# But in this formula language, it me
ans an interaction.
fit5=lm(medv~lstat*age,Boston)
summary(fit5)

# So that star in the formula means t
hat we're going to have
# main effects for each and the inter
action.
# And the pure interaction is indicat
ed by a colon.
# And while the main effect for age i
s not significant here,
# the interaction is somewhat signifi
cant.

# https://stackoverflow.com/question
s/24192428/what-does-the-capital-lett
er-i-in-r-linear-regression-formula-m
ean
# ?formula
```

```
Call:
lm(formula = medv ~ lsta
t * age, data = Boston)

Residuals:
    Min      1Q  Median
     3Q     Max
-15.806  -4.045  -1.333
  2.085  27.552

Coefficients:
              Estimate S
td. 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   0.97
11
lstat:age    0.0041560
0.0018518   2.244   0.02
52 *
---
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:  0.5
557,    Adjusted R-squar
```

```
ed:  0.5531
F-statistic: 209.3 on 3
and 502 DF,  p-value: <
2.2e-16
```

In [21]:

```
# And we saw that there was a non-lin
ear looking scatter plot
# between medv and lstat. And so here
 we explicitly put in a quadratic ter
m.

# And there's two things going on her
e.
# (1) the quadratic we indicate by ls
tat power two.
#     But power has a meaning in this
 formula language.
#     And so if you want it to mean a
ctually just raise lstat to
#     the power of two, we protect it
 with this identity function.
#     So the formula language doesn't
 dig inside this identity function.
#
# (2) we've put two commands in one l
ine, which you can do in R.
#     But you have to separate them w
ith a semi-colon.
#     So you can have as many command
s in one line as you like,
#     but separate them with semi-col
ons.
fit6=lm(medv~lstat +I(lstat^2),Boston
); summary(fit6)

# And sure enough, no surprise, both
 coefficients are strongly
# significant, the linear and the qua
dratic.
```

```
Call:
lm(formula = medv ~ lsta
t + I(lstat^2), data = B
oston)

Residuals:
     Min       1Q   Medi
an       3Q      Max
-15.2834  -3.8313  -0.52
95   2.3095  25.4148

Coefficients:
             Estimate St
d. Error t value Pr(>|t
|)
(Intercept) 42.862007
0.872084   49.15   <2e-1
6 ***
lstat       -2.332821
0.123803  -18.84   <2e-1
6 ***
I(lstat^2)   0.043547
0.003745   11.63   <2e-1
6 ***
---
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:  0.6
407,    Adjusted R-squar
ed:  0.6393
F-statistic: 448.5 on 2
and 503 DF,  p-value: <
2.2e-16
```

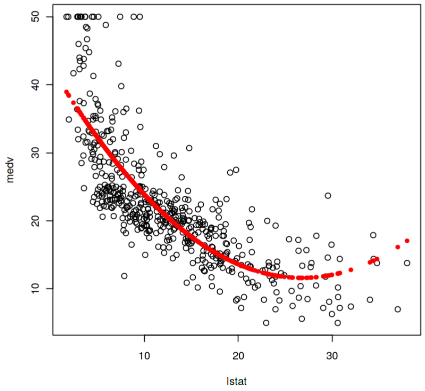
In [22]:

```
# attach: That means that the named v
ariables in Boston are
# available in our data space.
attach(Boston)
```

In [23]:

In [23]:

```r
# Now, we can't use abline anymore, because that only
# works when you've got a straight line fit.

# we use points:
# And the first argument is lstat itself.
# The second argument are the fitted values from fit6.
# That was our quadratic fit.
# So the fitted values are for each value of lstat, it's the
# fitted value from the model.
par(mfrow=c(1,1))
plot(medv~lstat)
points(lstat,fitted(fit6),col="red",pch=20)

# And the pch, which is the plotting character, is to be 20.
```

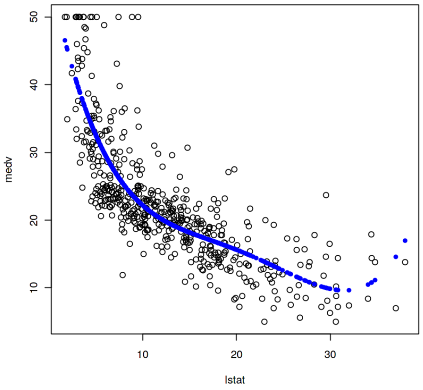


In [24]:

```r
# we are going to fit medv as a polynomial of degree four in lstat.
par(mfrow=c(1,1))
plot(medv~lstat)
fit7=lm(medv~poly(lstat,4))
fit7
points(lstat,fitted(fit7),col="blue",pch=20)
```

```
pch=20)
#?points
#?fitted
# And you can see that the fourth deg
ree polynomial is
# getting a little bit too wiggly.
# It's starting to over-fit the data
 a little bit,
```

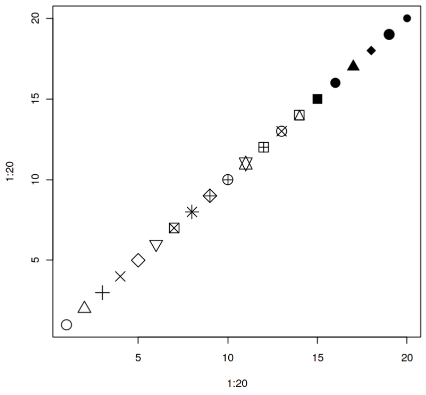
```
Call:
lm(formula = medv ~ poly
(lstat, 4))

Coefficients:
    (Intercept)  poly(ls
tat, 4)1  poly(lstat, 4)
2  poly(lstat, 4)3
          22.53
-152.46            64.23
           -27.05
poly(lstat, 4)4
          25.45
```



In [25]:

```
# Let's have a look at what plotting
 characters are available.
# So here's a simple way of seeing th
em all; plot one to
# 20 and plotting characters one to 2
0.
# We can see the whole lot. And there
 you see them.
```

```
plot(1:20,1:20,pch=1:20,cex=2)
```



In [26]:

```
###Qualitative predictors
```

In [27]:

```
# So the command fix is a way of throwing up an editor in R.
#fix(Carseats)
head(Carseats,n=10)
# its studies on children's carseats
```

Sales	CompPrice	Income	Advertisi
9.50	138	73	11
11.22	111	48	16
10.06	113	35	10
7.40	117	100	4
4.15	141	64	3
10.81	124	113	13
6.63	115	105	0
11.85	136	81	15
6.54	132	110	0
4.69	132	113	0

In [28]:

```
names(Carseats)
```

'Sales' 'CompPrice' 'Income'
'Advertising' 'Population' 'Price'
'ShelveLoc' 'Age' 'Education'
'Urban' 'US'

In [29]:

```
summary(Carseats)
```

```
     Sales          Comp
Price       Income
  Advertising
 Min.   : 0.000   Min.
: 77   Min.   : 21.00
Min.   : 0.000
 1st Qu.: 5.390   1st Q
u.:115   1st Qu.: 42.75
  1st Qu.: 0.000
 Median : 7.490   Median
:125   Median : 69.00
Median : 5.000
 Mean   : 7.496   Mean
:125   Mean   : 68.66
Mean   : 6.635
 3rd Qu.: 9.320   3rd Q
u.:135   3rd Qu.: 91.00
  3rd Qu.:12.000
 Max.   :16.270   Max.
:175   Max.   :120.00
Max.   :29.000
   Population        Pri
ce        ShelveLoc
   Age          Educatio
n
 Min.   : 10.0   Min.
: 24.0   Bad   : 96   Mi
n.   :25.00   Min.   :1
0.0
 1st Qu.:139.0   1st Q
u.:100.0   Good   : 85
1st Qu.:39.75   1st Qu.:
12.0
```

```
 Median :272.0   Median 
:117.0   Medium:219   Me
dian :54.50   Median :1
4.0  
 Mean   :264.8   Mean   
:115.8                 Me
an   :53.32   Mean   :1
3.9  
 3rd Qu.:398.5   3rd Q
u.:131.0                 
3rd Qu.:66.00   3rd Qu.:
16.0  
 Max.   :509.0   Max.   
:191.0                 Ma
x.   :80.00   Max.   :1
8.0  
 Urban       US     
 No :118   No :142  
 Yes:282   Yes:258  
```

In [30]:

```
# Sales~.  : It means everything in t
he frame but sales.
# Plus we're going to add in interact
ions between income, and advertising,

# and age, and price.
fit1=lm(Sales~.+Income:Advertising+Ag
e:Price,Carseats)
summary(fit1)

# And income and advertising appears
 to be strongly
# significant.But price and age is no
t.
```

```
Call:
lm(formula = Sales ~ . + 
Income:Advertising + Ag
e:Price, data = Carseat
s)

Residuals:
    Min      1Q  Median
     3Q     Max
-2.9208 -0.7503  0.0177
0.6754  3.3413

Coefficients:
                     Est
imate Std. Error t value
Pr(>|t|)
(Intercept)         6.57
55654  1.0087470   6.519
2.22e-10 ***
CompPrice           0.09
29371  0.0041183  22.567
  < 2e-16 ***
Income              0.01
08940  0.0026044   4.183
3.57e-05 ***
Advertising         0.07
02462  0.0226091   3.107
0.002030 **
Population          0.00
01592  0.0003679   0.433
0.665330
Price              -0.10
08064  0.0074399 -13.549
  < 2e-16 ***
ShelveLocGood       4.84
86762  0.1528378  31.724
  < 2e-16 ***
ShelveLocMedium     1.95
32620  0.1257682  15.531
  < 2e-16 ***
Age                -0.05
79466  0.0159506  -3.633
0.000318 ***
Education          -0.02
08525  0.0196131  -1.063
0.288361
UrbanYes            0.14
01597  0.1124019   1.247
0.213171
```

```
USYes              -0.15
75571  0.1489234  -1.058
0.290729
Income:Advertising  0.00
07510  0.0002784   2.698
0.007290 **
Price:Age           0.00
01068  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:  0.8
761,    Adjusted R-squar
ed:  0.8719
F-statistic:   210 on 13
and 386 DF,  p-value: <
2.2e-16
```

In [31]:

```
# ShelveLoc was a qualitative variabl
e.
# If you look at contrasts function,
 it shows you how R
# will code that variable when it's p
ut in a linear model.
#
# And in this case, it's a three-leve
l factor.
# And so it puts in two dummy variabl
```

Did you find this Kernel useful?
Show your appreciation with an upvote

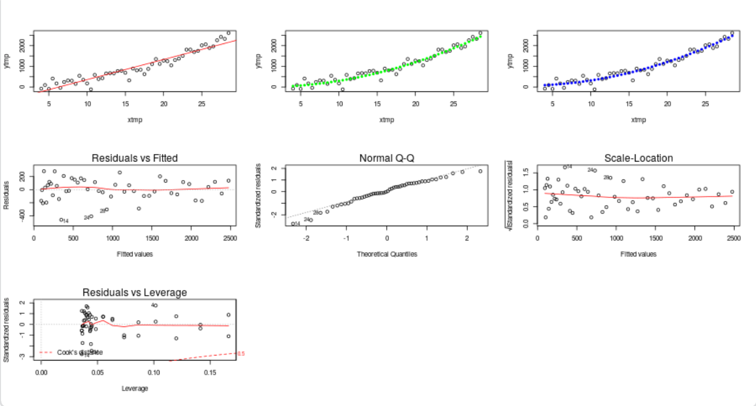
▲ 0

Data

No Data Sources

Edit to add a new Data Source

Output Visualizations



Run Info

Succeeded	True	Run Time	12.4 seconds
Exit Code	0	Queue Time	0 seconds
Docker Image Name	kaggle/rstats(Dockerfile)		
Timeout Exceeded	False	Output Size	0
		Used All Space	False
Failure Message			

Log

Download Log

```
Time   Line #  Log Message
2.7s        1  [NbConvertApp] Converting
               notebook script.irnb to html
5.3s        2  [NbConvertApp] Executing
               notebook with kernel: ir
12.0s       3  [NbConvertApp] Support files
               will be in
               __results___files/
```

```
                    [NbConvertApp] Making
                    directory __results___files
12.0s         4     [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
12.0s         5     [NbConvertApp] Making
                    directory __results___files
                    [NbConvertApp] Making
                    directory __results___files
12.0s         6     [NbConvertApp] Writing
                    331141 bytes to
                    __results__.html
12.0s         7
12.0s         9     Complete. Exited with code
                    0.
```

Comments (0)

Click here to enter a comment...

Our Team Terms Privacy Contact/Support