Linear Regression

Caitlin Simopoulos

2017-05-17

Linear regression

Let's do some "simple" analysis, starting with linear regression.

Quick review...

Linear regression is a *supervised* machine learning technique. The model *learns* from known data in order to predict from currently unknown data. Researchers also use this technique to determine trends in their data.

In short, linear regression explains the relationship between dependent (Y) and independent (X) variables (*i.e.* how does the values of Y change when X varies and other independent variables remain the same?).

Simply, linear regression takes the linear equation of:

$$Y = mX + b$$

But we're going to use it more like:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \epsilon_i$$

where:
$\hat{y}_i$ = predicted response for expeimental unit i
x_i = predictor (or independent variable) of experimental unit i
$\hat{\beta}_0$ = is expected value when $x_i = 0$ (intercept)
$\hat{\beta}_1$ = slope
ϵ = some error, because models are never perfect!

β_0 and β_1 are unknown, but are estimated from our training data! To do this, we must determine a line of best fit in some data that minimizes error.

An example: Does head size predict brain size?

Let's use some real data to create a linear regession, and use it to make some predictions. We've already imported this data... but let's re-do it.

```
library(data.table)
# we've already loaded this package...
head_brain <- fread('http://www.stat.ufl.edu/~winner/data/brainhead.dat')
head(head_brain)
```

```
##    V1 V2   V3   V4
## 1:  1  1 4512 1530
## 2:  1  1 3738 1297
## 3:  1  1 4261 1335
## 4:  1  1 3777 1282
## 5:  1  1 4177 1590
```

```
## 6:  1  1 3585 1300
```

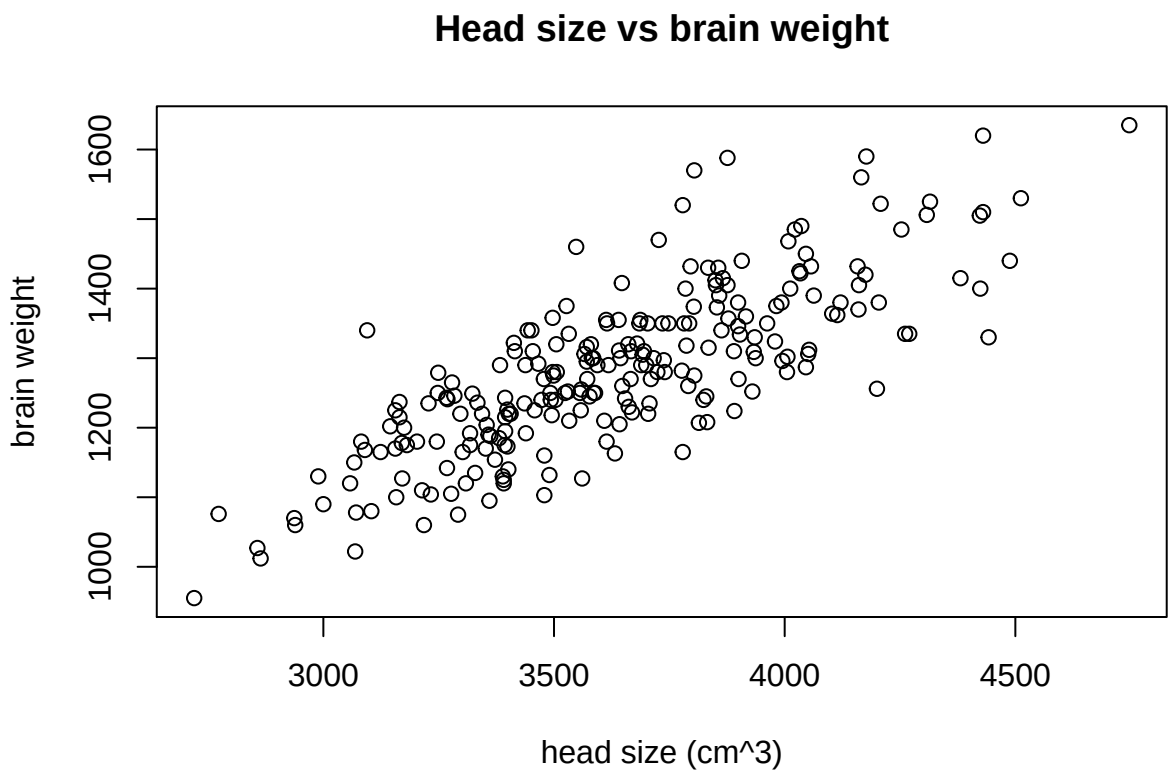
Columns are:

1. gender (1 = male, 2 = female)
2. age (1 = 20-46, 2 = 46+)
3. head size (cm^3)
4. brain weight (g)

```
# name columns to reflect what they are
colnames(head_brain) = c("gender", "age", "head", "brain")
```

Ok, let's look at the data to see if a linear model looks reasonable...

```
#attach(head_brain)
plot(head_brain$head, head_brain$brain,
     main="Head size vs brain weight", xlab = "head size (cm^3)", ylab = "brain weight")
```



Looks pretty reasonable to me!! OK, let's estimate some coefficients!

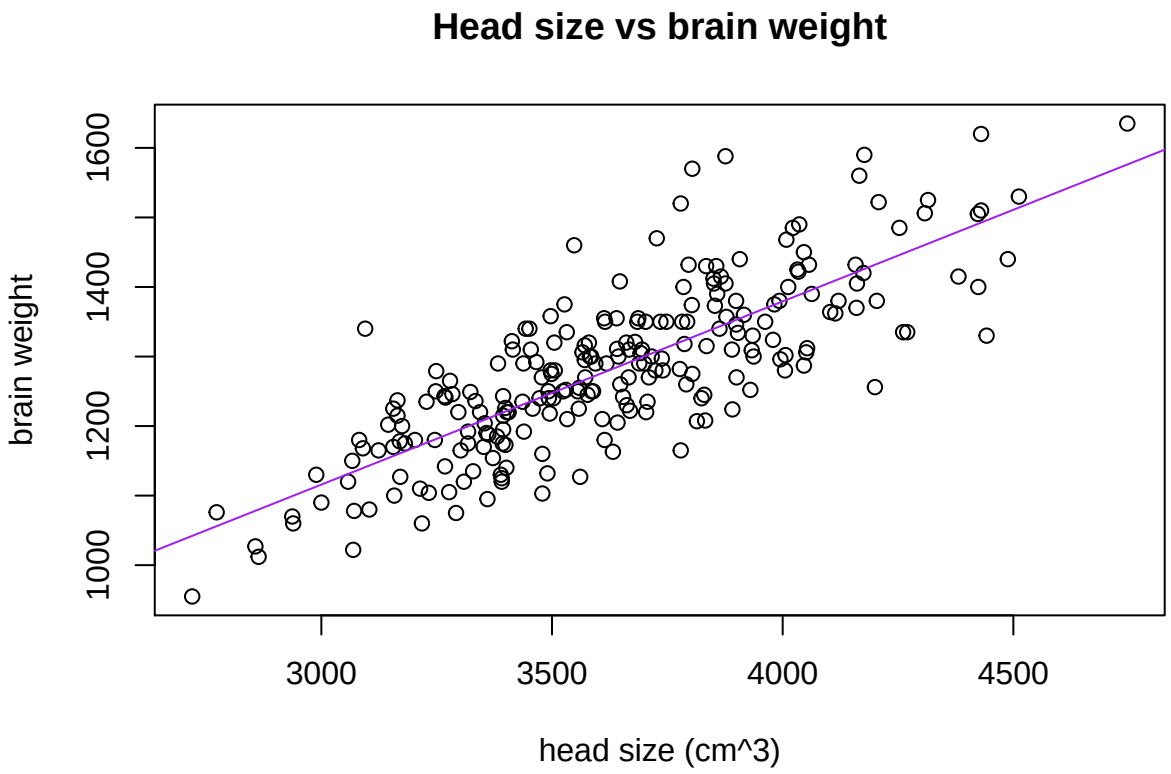
In this example let's see how well we can predict brain weight from head size.

```
lm1 = lm(brain ~ head, data=head_brain)
summary(lm1)
```

```
##
## Call:
## lm(formula = brain ~ head, data = head_brain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175.98  -49.76   -1.76   46.60  242.34
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.57342   47.14085   6.906 4.61e-11 ***
## head          0.26343    0.01291  20.409  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.43 on 235 degrees of freedom
## Multiple R-squared:  0.6393, Adjusted R-squared:  0.6378
## F-statistic: 416.5 on 1 and 235 DF,  p-value: < 2.2e-16
```

```
plot(head_brain$head, head_brain$brain, main="Head size vs brain weight", xlab = "head size (cm^3)", yla
abline(lm1, col="purple")
```



Our summary output gives us a lot of information:

1. Info about distrubution of residuals (errors)
2. The estimates of our coefficients
3. Standard error of coefficient estimates
   - square root of the variance
   - $\sqrt{\sigma^2}$
4. t-values (coefficient estimates/standard error)
5. R^2 = (want closer to 1)
6. p-values (yuck!)
   - testing if your coefficient = 0

Instead of p-values, you should use confidence intervals

```
confint(lm1, level=0.95) #95% confidence interval
```

```
##                   2.5 %      97.5 %
## (Intercept) 232.7007553 418.4460868
```

```
## head          0.2380003   0.2888584
# we are 95% confident that the intercept coeffifient estimate ranges between 232.7 and 418.4
#
```

This is great and all... but we have more information than what we've given our model, like gender and age categories of the individuals. Does this information help with the analysis?

Multiple linear regression

We can include our gender and age information in our anaylsis after tweaking the data just a bit. While we can input our data as-is, it's better to get into the habit of changing binary variables to 0 and 1 — our data is currently 1 and 2. We can use the very useful ifelse() function.

```
head_brain$gender = as.factor(ifelse(head_brain$gender == 1, 1, 0))
# if head_brain$gender[i] == 1
#     head_brain$gender[i] == 1
# else
#     head_brain$gender[i] == 0
head_brain$age = as.factor(ifelse(head_brain$age == 1, 1, 0))
```

Let's see if age category has an effect on predicting brain weight from head size...

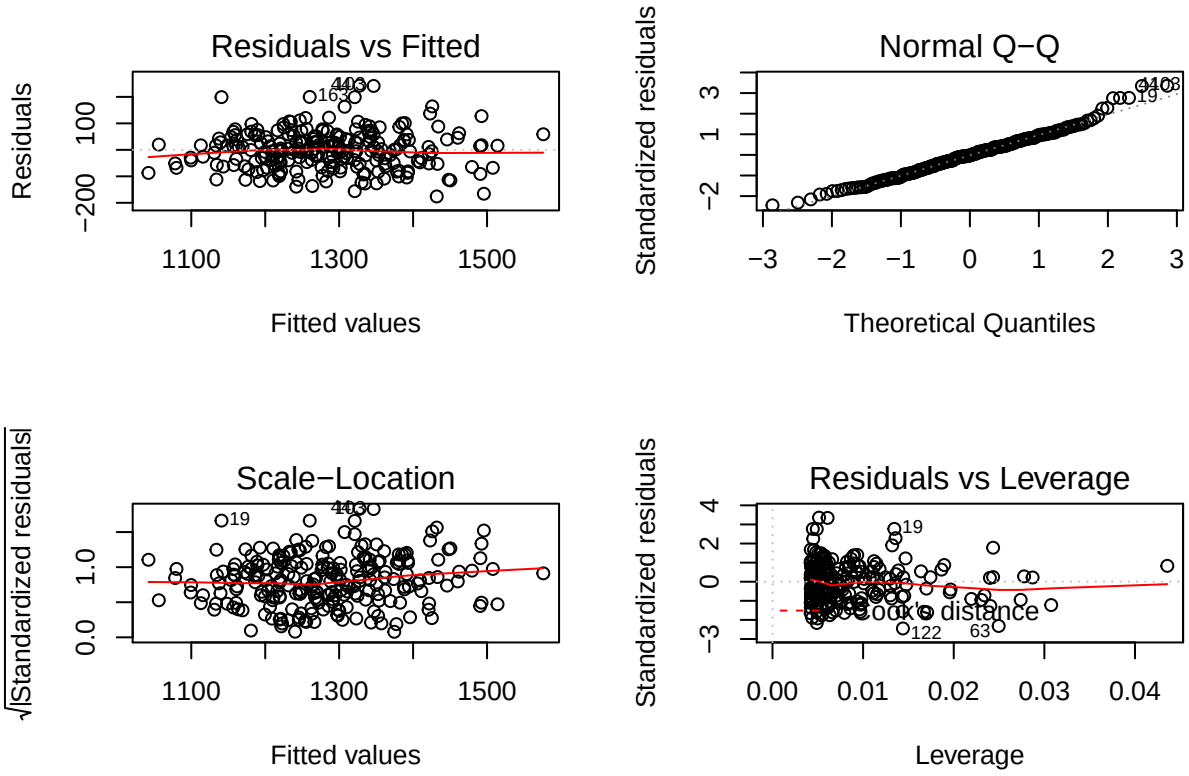
We have some options on possible formulas for the linear regression: * brain ~ head + age * $\hat{brain}_i = \hat{\beta}_0 + \hat{\beta}_1 head_i + \hat{\beta}_2 age_i + \epsilon_i$ * brain ~ head:age * $\hat{brain}_i = \hat{\beta}_0 + \hat{\beta}_2 age_i head_i + \epsilon_i$ * brain ~ head$age = brain \sim head + age + head\ x\ age$ $\hat{brain}_i = \hat{\beta}_0 + \hat{\beta}_1 head_i + \hat{\beta}_2 age_i + \hat{\beta}_2 age_i head_i + \epsilon_i$

```
lm2 = lm(brain ~ head + age, data=head_brain)
lm3 = lm(brain ~ head:age, data=head_brain)
lm4 = lm(brain ~ head * age, data=head_brain)
```

Exercise How do these models perform individually? How do these models perform when compared to each other?

Do these models meet assumptions of the linear model?

```
par(mfrow=c(2,2))
plot(lm1)
```



Residuals vs. Fitted

Looking for residuals to be randomly distributed around 0.

Q-Q plot

Q-Q plot can be used to look for normality. We want the points to follow the diagonal line.

Scale-Location plot

Similar to residuals vs fitted.

Residuals vs. Leverage plot

Was our model driven by outliers? If it was, Cook's distance lines would be visible (they aren't right now), and data points driving the model would be labeled. If this happens, try removing the outliers.

PS. http://data.library.virginia.edu/diagnostic-plots/ is helpful for interpretting these plots.

Predictions

OK cool..but how do we predict from these linear models?

Let's say we have some new head size data, and we want to use it to predict brain sizes.

```
new_data = data.frame(gender = as.factor(c(1,1,0,0)), age =as.factor(c(0,1,0,1)), head = c(2265, 999, 2
print(new_data)
```

```
##   gender age head
## 1      1   0 2265
## 2      1   1  999
## 3      0   0 2775
## 4      0   1 9275
```

```
lm1_pred <- predict(lm1,newdata=new_data,
                    interval="confidence", level=.95)
lm2_pred <- predict(lm2,newdata=new_data,
                    interval="confidence", level=.95)
lm3_pred <- predict(lm3,newdata=new_data,
                    interval="confidence", level=.95)
lm4_pred<- predict(lm4,newdata=new_data,
                   interval="confidence", level=.95)
print(lm1_pred)
```

```
##         fit       lwr       upr
## 1  922.2409  886.2160  958.2658
## 2  588.7393  521.0959  656.3827
## 3 1056.5898 1032.8614 1080.3183
## 4 2768.8805 2625.1358 2912.6252
```

```
print(lm2_pred)
```

```
##         fit       lwr       upr
## 1  916.7127  880.6385  952.7868
## 2  607.7288  538.5166  676.9410
## 3 1049.5364 1025.1693 1073.9036
## 4 2763.1201 2620.4437 2905.7964
```

```
print(lm3_pred)
```

```
##         fit       lwr       upr
## 1  920.3607  884.5438  956.1775
## 2  598.9382  531.0790  666.7974
## 3 1052.0020 1028.0432 1075.9607
## 4 2779.5253 2636.4323 2922.6182
```

```
print(lm4_pred)
```

```
##         fit       lwr       upr
## 1  899.4405  851.5487  947.3323
## 2  647.6267  547.1953  748.0581
## 3 1038.8714 1007.6941 1070.0486
## 4 2679.6397 2470.9700 2888.3094
```

Now, we've used our model to predict! Very cool! But, the predictions from the models vary... so how do we know which predictions are better? We should evaluate our models using our training data.

Exercise How can you evaluate models using training data? Hint: you might need to subset the training data....

Acknowledgements

This document was inspired by: this tutorial