Statistics

Irfan Kanat

11/04/2015

In this section, I will try to provide an introduction to using two simple statistical models in R: regression and logistic regression.

Regression

If your dependent variable is continuous you can simply use regression.

For this demonstration, I will use the same Motor Trends dataset I used in Visualization section.

```
data(mtcars) # Get the data
?mtcars # Help on dataset
```

We will use lm() function to fit regular regression.

```
?lm
```

Below I declare a model where I use horse power, cylinders, and transmission type to estimate gas milage. Pay attention to model specification:

```
mpg ~ hp + cyl + am
```

Here the left hand side of the tilde is the dependent variable. and the right hand side has all the predictors we use separated by plus signs.

```
# Fit
reg_0 <- lm( mpg ~ hp + cyl + am, data = mtcars)
summary(reg_0)
```

```
##
## Call:
## lm(formula = mpg ~ hp + cyl + am, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.864 -1.811 -0.158  1.492  6.013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.88834    2.78422  11.094 9.27e-12 ***
## hp          -0.03688    0.01452  -2.540  0.01693 *
## cyl         -1.12721    0.63417  -1.777  0.08636 .
## am           3.90428    1.29659   3.011  0.00546 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.807 on 28 degrees of freedom
## Multiple R-squared:  0.8041, Adjusted R-squared:  0.7831
## F-statistic: 38.32 on 3 and 28 DF,  p-value: 4.791e-10
```

Look at the R-squared value to see how much variance is explained by the model, the more the better.

You can access estimated values as follows. I used a head function to limit the output.

```
head(reg_0$fitted.values)
```

```
##         Mazda RX4     Mazda RX4 Wag        Datsun 710    Hornet 4 Drive
##          23.97302          23.97302          26.85433          20.06874
## Hornet Sportabout           Valiant
##          15.41740          20.25312
```

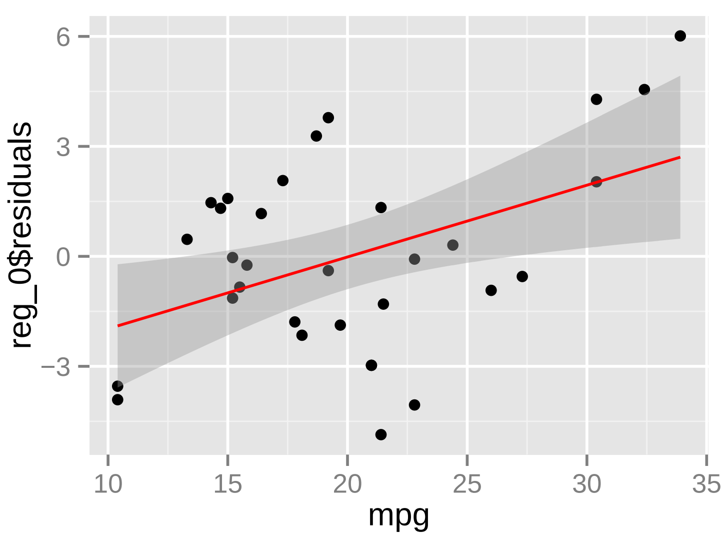
You can use the fitted model to predict new datasets. Here I am modifying Datsun710 to see how the gas milage may have been influenced if the car was automatic instead of manual transmission.

```
newCar <- mtcars[3,] # 3rd observation is Datsun 710
newCar$am <- 0 # What if it was automatic?
predict(reg_0, newdata = newCar) # Estimate went down by 4 miles
```

```
## Datsun 710
##   22.95005
```

One way to see how your model did is to plot residuals. Ideally the residuals should be close to 0 and randomly distributed. If you see a pattern, it indicates misspecification.

```
library(ggplot2)
# Plot the fitted values against real values
qplot(data=mtcars, x = mpg, y = reg_0$residuals) +
  stat_smooth(method = "lm", col = "red")
```



```r
# Are the residuals normally distributed?
shapiro.test(reg_0$residuals) # yes
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg_0$residuals
## W = 0.98366, p-value = 0.8961
```

Comparing models. If you are using the same dataset, and just adding or removing variables to a model. You can compare models with a likelihood ratio test or an F test. Anova facilitates comparison of simple regression models.

```r
# Add variable wt
reg_1 <- lm( mpg ~ hp + cyl + am + wt, mtcars)

# Aikikae Information Criteria
# AIC lower the better
AIC(reg_0)
```

```
## [1] 162.5849
```

```r
AIC(reg_1)
```

```
## [1] 156.2536
```

```r
# Compare
anova(reg_0, reg_1) # models are significantly different
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ hp + cyl + am
## Model 2: mpg ~ hp + cyl + am + wt
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     28 220.55
## 2     27 170.00  1    50.555 8.0295 0.008603 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Logistic Regression

Let us change gears and try to predict a binary variable. For this purpose we will use the logistic regression with a binomial link function. The model estimates the probability of Y=1.

Let us stick to the mtcars dataset and try to figure out if a car is automatic or manual based on predictors.

We will use glm function.

```r
?glm
```

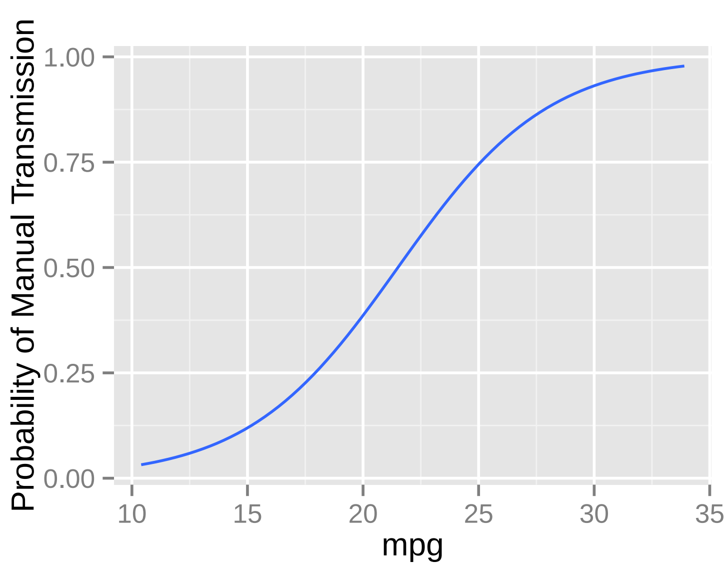
Let us fit the model

```r
logit_2 <- glm(am ~ mpg + drat + cyl, data = mtcars, family='binomial')
summary(logit_2)
```

```
##
## Call:
## glm(formula = am ~ mpg + drat + cyl, family = "binomial", data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.58367  -0.31020  -0.03757   0.17972   1.75395
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -49.4548    24.1280  -2.050   0.0404 *
## mpg           0.6378     0.4266   1.495   0.1349
## drat          7.2595     3.2702   2.220   0.0264 *
## cyl           1.6115     1.0801   1.492   0.1357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.23  on 31  degrees of freedom
## Residual deviance: 17.03  on 28  degrees of freedom
## AIC: 25.03
##
## Number of Fisher Scoring iterations: 7
```

Visualize the results.

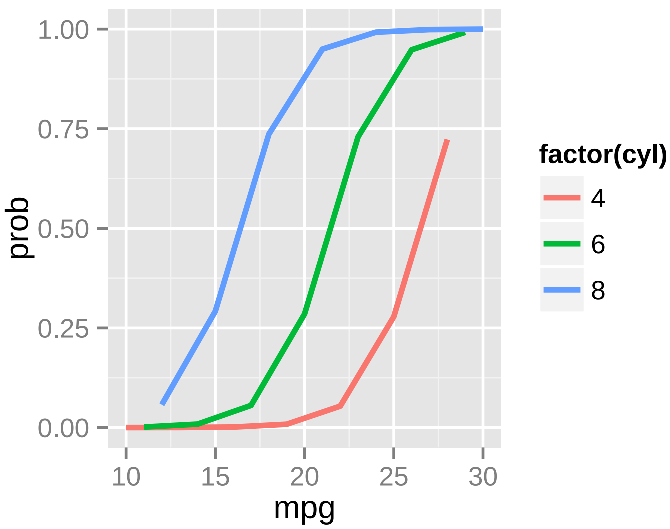
```r
ggplot(mtcars, aes(x = mpg, y = am)) +
    stat_smooth(method="glm", family="binomial", se=FALSE)+
# Bonus: rename the y axis label
        ylab('Probability of Manual Transmission')
```



How about plotting results for number of cylinders? We will need to process the data a little bit.

```r
# Create a new dataset with varying number of cylinders and other variables fixed at mean levels.
mtcars2<-data.frame(mpg = rep(10:30, 3),drat = mean(mtcars$drat), disp = mean(mtcars$disp), cyl = rep(c
# Predict probability of new data
mtcars2$prob<-predict(logit_2, newdata=mtcars2, type = "response")

# Plot the results
ggplot(mtcars2, aes(x=mpg, y=prob)) +
  geom_line(aes(colour = factor(cyl)), size = 1)
```



Diagnostics with logistic regression.

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
# Let us compare predicted values to real values
mtcars$prob <- predict(logit_2, type="response")
# Prevalence of Manual Transmission
mean(mtcars$am)
```

```
## [1] 0.40625
```

```r
# Create predict variable
mtcars$pred <- 0
# If probability is greater than .6 (1-prevalence), set prediction to 1
mtcars[mtcars$prob>.6, 'pred'] <- 1

# Confusion Matrix
confusionMatrix(table(mtcars[,c("am", "pred")]))
```

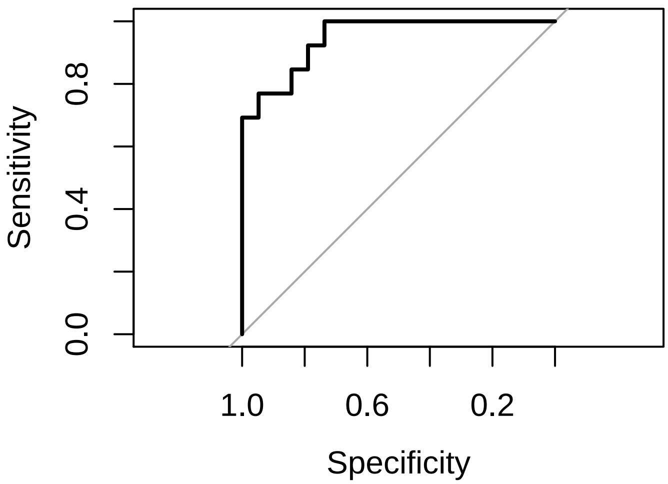
```
## Confusion Matrix and Statistics
##
##    pred
```

```
## am   0  1
##   0 18  1
##   1  3 10
##
##                Accuracy : 0.875
##                  95% CI : (0.7101, 0.9649)
##     No Information Rate : 0.6562
##     P-Value [Acc > NIR] : 0.005004
##
##                   Kappa : 0.7344
##  Mcnemar's Test P-Value : 0.617075
##
##             Sensitivity : 0.8571
##             Specificity : 0.9091
##          Pos Pred Value : 0.9474
##          Neg Pred Value : 0.7692
##              Prevalence : 0.6562
##          Detection Rate : 0.5625
##    Detection Prevalence : 0.5938
##       Balanced Accuracy : 0.8831
##
##        'Positive' Class : 0
##
```

```r
## ROC CURVE
# Load the necessary library
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
# Calculate the ROC curve using the predicted probability vs actual values
logit_2_roc <- roc(am~prob, mtcars)
# Plot ROC curve
plot(logit_2_roc)
```




```
##
## Call:
## roc.formula(formula = am ~ prob, data = mtcars)
##
## Data: prob in 19 controls (am 0) < 13 cases (am 1).
## Area under the curve: 0.9474
```