Regression Modelling in `R`

Chris Mainey

Senior Data Scientist
University Hospitals Birmingham
NHS Foundation Trust

chris.mainey@uhb.nhs.uk
@chrismainey

Workshop Overview

- Correlation
- Linear Regression
  - Specifying a model with `lm`
  - Interpreting the model output
  - Assessing model fit
- Multiple Regression
- Prediction
- Generalized Linear Models using `glm`
  - Logistic Regression

Mixture of theory, examples and practical exercises

HED

Relationships between variables

If two variables are related, we usually describe them as 'correlated'.

Usually interested in "strength" and "direction" of association

Two analysis techniques commonly used to investigate:

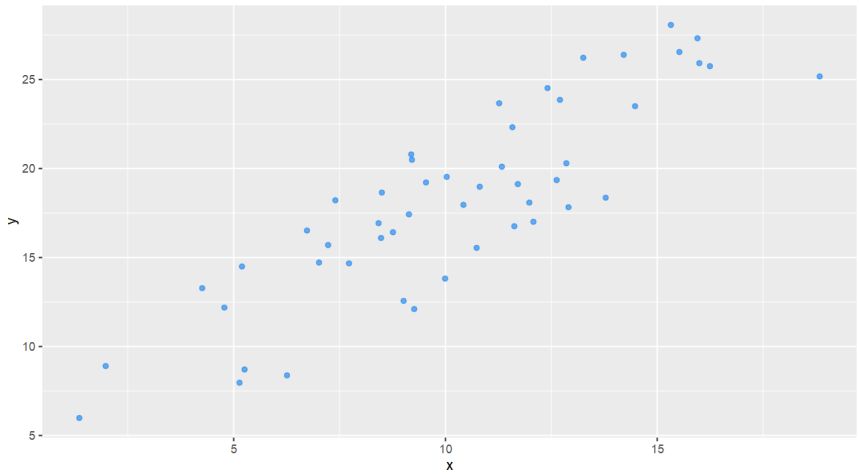
- *Correlation:* shows direction, and strength of association
- *Regression:* estimate how one variable: y changes when we observed changes in another: x (or more than one, x_i)

Sometimes the effects of other variables interact/mask this (*"confounding"*)

HED

Example:

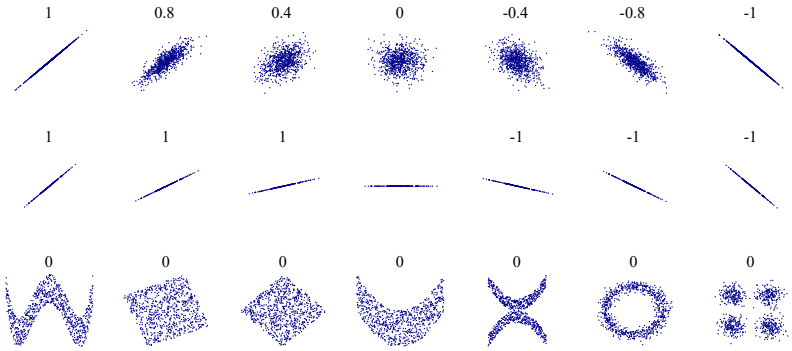
```
x <- rnorm(50, 10, 4)
y <- runif(50, min = 0.5, 10) + (1.25*x)

a<-ggplot(data.frame(x,y,z), aes(x=x,y=y))+
  geom_point(col="dodgerblue2", alpha=0.65)
a
```



HED

Correlation

- Measured with a correlation coefficient ('Pearson' is the most common)
- Range:
  - **-1 to 1:** Perfect negative to Perfect positive Correlation
  - **0:** No Correlation



Graphic from: Wikipedia: Correlation and dependence: By DenisBoigelot, https://commons.wikimedia.org/w/index.php?curid=15165296 [Accessed 24 Sept 2019]

HED

Correlation in R

Lets check the correlation in our generated data:

```
cor(x, y)
#> [1] 0.8650106

cor.test(x,y)
#>
#>     Pearson's product-moment correlation
#>
#> data:  x and y
#> t = 11.944, df = 48, p-value = 5.53e-16
#> alternative hypothesis: true correlation is not equal to 0
#> 95 percent confidence interval:
#>  0.7727115 0.9214882
#> sample estimates:
#>       cor
#> 0.8650106
```

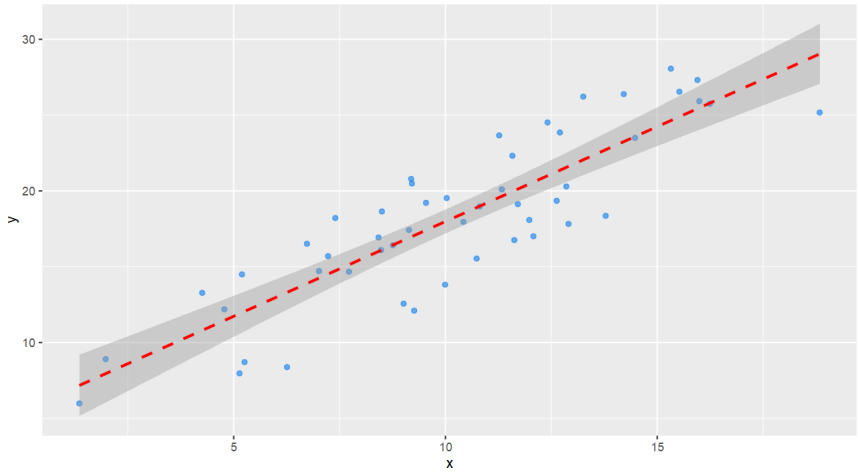
- `cor.test` is a correlation and a t-test.
- Different types of correlation coefficient, default is 'Pearson'
- Doesn't work for different distributions, data types or more variables

HED

Regression models (1)

Regression gives us more options than correlation:

```
z <- lm(y~x)
print(a<- a + geom_smooth(method="lm", col="red", linetype=2))
```



$$y = \alpha + \beta x + \epsilon$$

HED

Regression equation

$$y = \alpha + \beta_i x_i + \epsilon$$

- y - is our 'outcome', or 'dependent' variable
- α - is the 'intercept', the point where our line crosses y-axis
- β - is a coefficient (weight) applied to x
- x - is our 'predictor', or 'independent' variable
- i - is our index, we can have i predictor variables, each with a coefficient
- ϵ - is the remaining ('residual') error

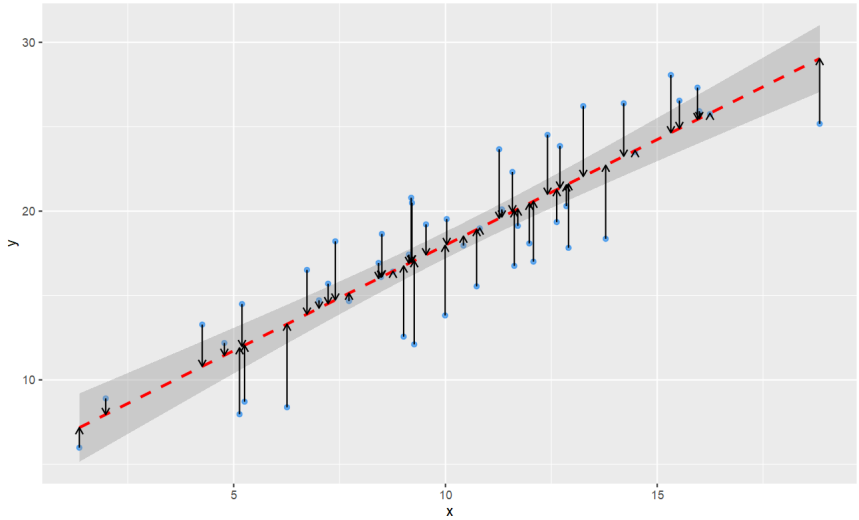
We are making some assumptions:

- Linear relations
- Data points are independent (not correlated)
- Normally distributed error
- Homoskedastic (error doesn't vary across the range)

Ordinary Least Squares 'OLS'

- 'Residual' distance between prediction and data point (ϵ).
- Sum would be zero, so we square and minimise *'sum of the squares'*

```
a + geom_segment(aes(xend=c(x), yend=c(z)), size=0.5, arrow=arrow(length=unit(0.2,"cm")))
```



Regression models (3)

- We created an `lm` object called `z`. You can then use that object with other functions like `summary`, `predict`, `plot` etc.

```
summary(z)
#>
#> Call:
#> lm(formula = y ~ x)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -4.9575 -2.2614  0.4444  2.4475  4.1663
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   5.4776     1.1386   4.811 1.53e-05 ***
#> x             1.2507     0.1047  11.944 5.53e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.77 on 48 degrees of freedom
#> Multiple R-squared:  0.7482,    Adjusted R-squared:  0.743
#> F-statistic: 142.7 on 1 and 48 DF,  p-value: 5.53e-16
```

- We can test fit using f-tests, prediction error or the R^2 .
- R^2 is the proportion of variation in y, explained by x .

Interpretation

We can now read this as:

- *"For each increase of 1 in x, y increases by 1.25, starting at 5.48."*

A common addition is to **"mean-centre and scale"** our variables:

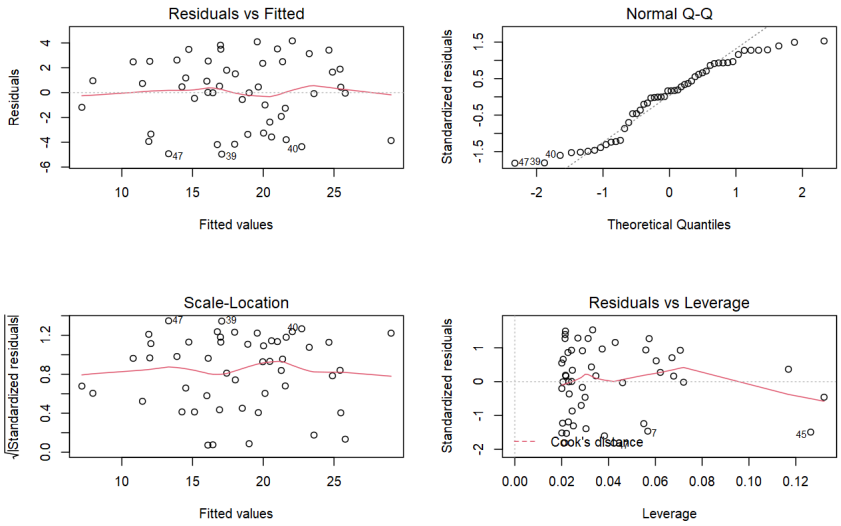
Original Scaled Interpretation

```
z <- lm(y~x)
summary(z)
#>
#> Call:
#> lm(formula = y ~ x)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -4.9575 -2.2614  0.4444  2.4475  4.1663
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   5.4776     1.1386   4.811 1.53e-05 ***
#> x             1.2507     0.1047  11.944 5.53e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.77 on 48 degrees of freedom
#> Multiple R-squared:  0.7482,    Adjusted R-squared:  0.743
#> F-statistic: 142.7 on 1 and 48 DF,  p-value: 5.53e-16
```


Regression diagnostics (5)

A common check is to plot residuals:

```
plot(z)
```



Exercise 1: Linear regression with a single predictor

More than one predictor?

Our plots in earlier slides make sense in 2 dimensions, but regression is not limited to this.

If we add more predictos, our interpretation of each coefficient becomes:

- *"The change in y whilst holding all others parameters constant"*

We can add more predictors with the `+`:

```
lm(y ~ x1 + x2 + x3 + xi)
```

Categorical variables

How do we enter categorical variables into a model?

- Models won't understand text, and numbers are numeric, so we use `factor` variables?

`Factors` are 'dummy coded:'

- *'pivotted' to binary columns*
- Contain a reference level: with categories: "A", "B" & "C", we get:

```
a<- data.frame(Category = factor(c("A", "B", "C", "C", "A", "B")))
model.matrix(~., a)
#>   (Intercept) CategoryB CategoryC
#> 1           1         0         0
#> 2           1         1         0
#> 3           1         0         1
#> 4           1         0         1
#> 5           1         0         0
#> 6           1         1         0
#> attr(,"assign")
#> [1] 0 1 1
#> attr(,"contrasts")
#> attr(,"contrasts")$Category
#> [1] "contr.treatment"
```


Exercise 2: Linear regression with multiple predictors

What about non-linear data?

- Data are not necessarily linear. Death is binary, LOS is a count etc.
- We can use the Generalized Linear Model (GLM):

$$g(\mu) = \alpha + \beta x$$

Where μ is the expectation of Y, and g is the link function

- The link function transforms the data before fitting a model
- Can't use OLS for this, so we use 'maximum-likelihood'
- Many of the methods for `lm` are common to `glm`, but we can't use R^2
- Other measures include AUC ('C-statistic'), and AIC or likelihood ratio tests.

Generalized Linear Models

- `family` argument was set to `binomial` for binary outcomes.

```
library(COUNT)
data(medpar)

glm_binomial <- glm(died ~ factor(age80) + los + factor(type), data=medpar, family="binomial")

sjPlot::tab_model(glm_binomial, show.df = TRUE, show.obs = TRUE, show.se = TRUE, show.r2 = FALSE)
```

			1=Died;0=Alive		
Predictors	*Odds Ratios*	*std. Error*	*CI*	*p*	*df*
(Intercept)	0.55	0.10	0.46 – 0.67	<0.001	Inf
factor(age80)1	1.93	0.13	1.50 – 2.48	<0.001	Inf
Length of Stay	0.96	0.01	0.95 – 0.98	<0.001	Inf
factor(type)2	1.52	0.14	1.14 – 2.02	0.004	Inf
factor(type)3	2.61	0.23	1.66 – 4.12	<0.001	Inf
Observations	1495				

```
ModelMetrics::auc(glm_binomial)
#> [1] 0.6372224
```

HED
Interactions

- 'Interactions' are where predictor variables affect each other.
- Can add these with `*` or `:` (check help for which to use)

```
glm_binomial2 <- glm(died ~ factor(age80) * los + factor(type), data=medpar, family="binomial")

sjPlot::tab_model(glm_binomial2, show.df = TRUE, show.obs = TRUE, show.se = TRUE, show.r2 = FALSE)
```

			1=Died;0=Alive		
Predictors	*Odds Ratios*	*std. Error*	*CI*	*p*	*df*
(Intercept)	0.57	0.10	0.47 – 0.70	<0.001	Inf
factor(age80)1	1.69	0.21	1.13 – 2.56	0.011	Inf
Length of Stay	0.96	0.01	0.94 – 0.98	<0.001	Inf
factor(type)2	1.52	0.14	1.14 – 2.01	0.004	Inf
factor(type)3	2.62	0.23	1.67 – 4.14	<0.001	Inf
factor(age80)1:los	1.01	0.02	0.98 – 1.05	0.419	Inf
Observations	1495				

```
ModelMetrics::auc(glm_binomial2)
#> [1] 0.6376572
```


HED
Interpretation

- Our model coefficients in `lm` were straight-forward multipliers
- `glm` is similar, but it is on the scale of the link-function.
  - log scale for `poisson` models, or logit (log odds) scale for `binomial`
- Common to transform output back to response scale
- giving Incident Rate Ratios for `poisson`, or Odds Ratios for `binomial`

```
cbind(Link=coef(glm_binomial2), Response=exp(coef(glm_binomial2)))
#>                              Link  Response
#> (Intercept)         -0.56160376 0.5702937
#> factor(age80)1       0.52537865 1.6910991
#> los                 -0.04073769 0.9600809
#> factor(type)2        0.41743940 1.5180694
#> factor(type)3        0.96477095 2.6241865
#> factor(age80)1:los   0.01450661 1.0146123
```

Exercise 3: Generalized Linear Model (GLM)

Prediction (1)

- We can then use our model to predict our expected Y:
- Need to decide what scale to predict on: `link` or `response`

```
library(dplyr)
medpar$preds <- predict(glm_binomial2, type="response")

top_n(medpar,5) %>% knitr::kable(format = "html")
```

	los	hmo	white	died	age80	type	type1	type2	type3	provnum	preds
558	1	0	1	1	1	3	0	0	1	030017	0.7114250
919	5	0	1	1	1	3	0	0	1	030061	0.6894160
1464	2	0	1	1	1	3	0	0	1	032000	0.7060100
1486	5	0	1	1	1	3	0	0	1	032002	0.6894160
1488	4	0	1	1	1	3	0	0	1	032002	0.6950045

Prediction (2)

- Lets see the 10 cases with the highest predicted risk of death:

```
medpar %>% arrange(desc(preds)) %>% top_n(10) %>% knitr::kable(format = "html")
#> Selecting by preds
```

	los	hmo	white	died	age80	type	type1	type2	type3	provnum	preds
558	1	0	1	1	1	3	0	0	1	030017	0.7114250
1464	2	0	1	1	1	3	0	0	1	032000	0.7060100
1488	4	0	1	1	1	3	0	0	1	032002	0.6950045
919	5	0	1	1	1	3	0	0	1	030061	0.6894160
1486	5	0	1	1	1	3	0	0	1	032002	0.6894160
955	6	0	1	0	1	3	0	0	1	030061	0.6837716
896	9	0	1	0	1	3	0	0	1	030061	0.6665153
941	9	0	1	0	1	3	0	0	1	030061	0.6665153
1084	10	0	1	1	1	3	0	0	1	030069	0.6606596
1482	11	0	1	1	1	3	0	0	1	032000	0.6547543

Exercise 4: Predicting from models

Summary

- Correlation shows the direction and strength of association
- Regression allows us to quantify the relationships
- We can use a single, or multiple, predictors
- Regression coefficients explain how much a change in x affects y
- R^2 is a common measure of in linear models, C-statistic/AUC/ROC in logistic models
- Generalized Linear Model (`glm`) allow linear models on a transformed scale, e.g. logistic regression for binary variables
- Interactions terms allow us to examine confounded predictors
- We can predict from our model objects, but must remember the link-function scale in `glm`

Exercise 5: Predicting 10-year CHD risk in Framingham data