How to fit a model in R

Use the lm() function:

```
lm(response ~ explanatory, data = df)
```

You should always save the output of this to a variable so that you can use it later.

Once the model is constructed, you can:

- Get information about it with **`summary()`**
- Use **`abline()`** to add the line to a plot

Let's try it...

Confidence Interval

Recall confidence intervals:

point estimate ± critical value * standard error

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.20760    9.29990   0.990    0.332
Biological   0.90144    0.09633   9.358  1.2e-09
```

95% confidence interval for the slope:

$0.90144 \pm 2.06 \times 0.09633$

(0.703, 1.010)

2.06: t-statistic for 95% confidence, df = 25

Multiple Regression

Simple linear regression

2 variables: *y* and *x*

$y = \beta_0 + \beta_1 x$ + "error"

Multiple linear regression

2+ variables: y and x_1, x_2, …

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$ + "error"

Simple vs. Multiple Regression

weight ~ volume:

107.68 + 0.71 volume $R^2 = .80$

weight ~ cover:

796.43 - 168.30 cover $R^2 = .10$

weight ~ volume + cover:

197.96 + 0.72 volume - 184.05 cover $R^2 = .93$

Why aren't the volume and cover coefficients from the single variable models the same as those in the multiple variable model?

Statistical significance

```
lm(formula = Foster ~ Biological, data = twins)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.20760    9.29990   0.990    0.332
Biological   0.90144    0.09633   9.358  1.2e-09 ***
```

"Biological" is the coefficient b_1.

Pr(>|t|) is the p-value. It is much smaller than 0.05, so there is a significant relationship: one twin's IQ predicts the other.

"Significance" of the intercept

- Whether or not x, y are associated depends only on slope
- Intercept b_0 is determined by b_1, x, y, which are all analyzed separately

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.20760    9.29990   0.990    0.332
```

- Calculated with respect to null $b_0 = 0$, which may not be relevant

For all these reasons, we don't generally do inference on β_0

Multiple regression coefficients

```
lm(formula = weight ~ volume + cover, data = allbacks)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 197.96284   59.19274   3.344 0.005841
volume        0.71795    0.06153  11.669  6.6e-08
coverpb    -184.04727   40.49420  -4.545 0.000672

Residual standard error: 78.2 on 12 degrees of freedom
Multiple R-squared:  0.9275,  Adjusted R-squared:  0.9154
F-statistic: 76.73 on 2 and 12 DF,  p-value: 1.455e-07
```

Categorical variable coefficients

weight = 197.96 + 0.72 volume - 184.05 cover

For hardback:

weight = 197.96 + 0.72 volume - 184.05 × 0
= 197.96 + 0.72 volume

For paperback:

weight = 197.96 + 0.72 volume - 184.05 × 1
= 13.91 + 0.72 volume

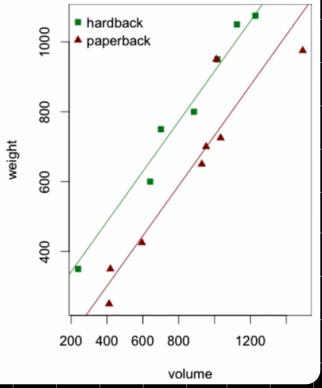
Visualizing multiple regression

With 1 numerical variable and 1 categorical variable, draw a line for each value of the categorical variable

$weight_{hb}$ =
197.96 + 0.72 volume

$weight_{pb}$ =
13.91 + 0.72 volume

With more variables, it can be difficult to visualize



Simple vs. Multiple Regression

Adding explanatory variables may:

Increase R^2

- Each explanatory variable explains a different part of the variance of the response variable

Change p-values

- Variable may be "explained away" by another
- Variable may be able to explain the data only after another variable explains some variance

Example: trans fat

Data: nutrition information on 111 Burger King menu items

Response variable: trans fat content (g)

Some possible explanatory variables:

- Calories (numerical)
- Total fat (g, numerical)
- Sugar (g, numerical)
- Does the item have meat? (categorical)
- Is the item a breakfast item? (categorical)

Should we use all these explanatory variables?

Why not include all variables?

It's usually little or no extra work to add variables; so why exclude any?

- Simpler models are easier to interpret
- More variables -> more random variation
- More variation means the estimates b_0, b_1, etc. will have more random error; so additional variables should contribute something useful

Step-wise regression

Backward stepwise regression with R^2

1. Build model with all variables, record its R^2
2. For each variable in the model
   - Build model missing that variable
3. Stop if all new R^2s are lower than the original
4. Otherwise, choose model with highest R^2
5. Repeat from step 1, using remaining variables

Alternatives:

- forward instead of backward
- other measures instead of R^2

Better fit vs. simpler model?

More variables will never decrease and will typically increase R^2.

So is a model with more variables a better fit? With standard R^2, we can't tell.

Adjusted R^2 allows for such comparisons

Better fit vs. simpler model?

More variables will never decrease and will typically increase R^2.

So is a model with more variables a better fit? With standard R^2, we can't tell.

Adjusted R^2 allows for such comparisons

$$R^2 = 1 - \frac{Var(e_i)}{Var(y_i)} \qquad R^2_{adj} = 1 - \frac{Var(e_i)}{Var(y_i)} \times \frac{n-1}{n-p-1}$$

Alternative model selection

Alternative models: penalize large coefficients

- Ridge regression (penalizes β_i^2)
- LASSO (penalizes $|\beta_i|$)
- These "shrink" coefficients (possibly to 0--meaning a variable is dropped)

Alternative criteria:

- "Information criteria" (e.g. AIC, BIC) evaluate models, penalize # of variables differently than adjusted R^2
- Some stepwise algorithms use these -- for instance, R's `step()` uses AIC by default

Model evaluation

In practice, the best way to evaluate a model is to see how it performs on "unknown" data

Overfitting: when your model fits the data you used to construct it very well, but makes bad predictions

Overfitting can be the result of using too many variables or a too sophisticated model

Model evaluation

It's important that we test a model on different data than we used to construct it

So, in deciding which variables to use:

- Split data into "training data" (used to fit the model) and "testing data" (used to evaluate)
- Fit the models on the "training data" and make predictions on the "testing data"
- Assess accuracy of predictions by calculating the sum of squared residuals

Logistic Regression

GLM for categorical response variables

$$\log_e(p / 1\text{-}p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$$

Logistic regression predicts probabilities:

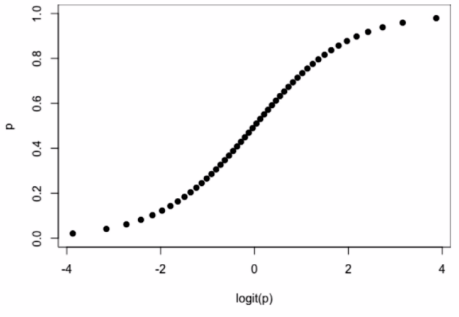
$$\hat{p} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots)}}$$

Interpretation:

- High $\hat{p}$ ⇒ outcome 1 is likely
- Low $\hat{p}$ ⇒ outcome 0 is likely

Transforming (0, 1) ⇔ (-∞,+∞)

$$logit(p_i) = log_e\left(\frac{p}{1-p}\right)$$



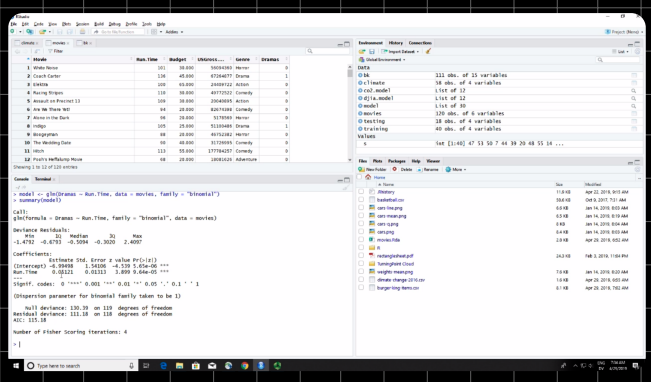
Interpretation of slopes

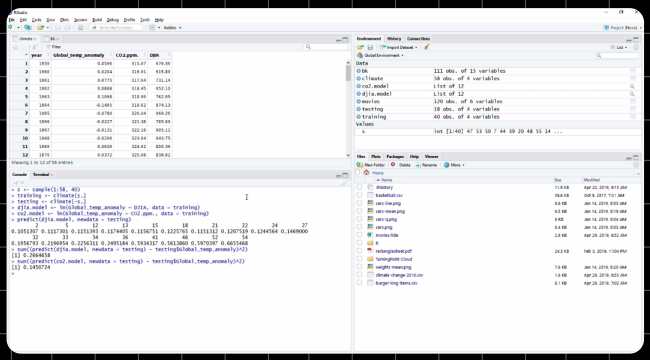
$\text{logit}(p_{\text{StatusSurvive}}) = \beta_0 + \beta_1 \text{ Age} + \beta_2 \text{ SexMale}$

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.23041    1.38686   2.329   0.0198 *
Age         -0.07820    0.03728  -2.097   0.0359 *
SexMale     -1.59729    0.75547  -2.114   0.0345 *
```

The correct interpretation of the Age slope is:

After controlling for sex, a person that is 1 year older is expected to have their odds of survival reduced by a factor of $e^{-0.07820} \approx 92\%$





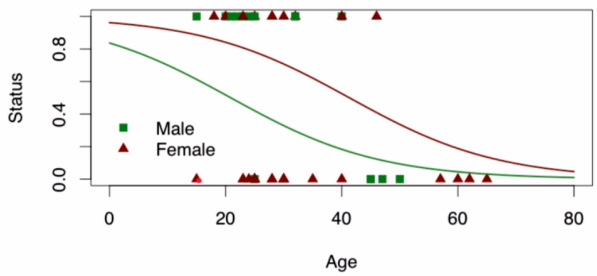
Predicting Survival

In 1846 the Donner and Reed families left Illinois for California by covered wagon. The group became stranded when the Sierra Nevada mountains were hit by heavy snows. By the time the last survivor was rescued on April 21, 1847, 40 of the 87 members had died from famine and exposure to extreme cold.

	Age	Sex	Status
1	23	Male	Died
2	40	Female	Survived
3	40	Male	Survived
4	30	Male	Died
5	28	Male	Died
6	40	Male	Died
7	45	Female	Died
8	62	Male	Died
9	65	Male	Died
10	45	Female	Died
...			

Logistic Regression for Survival

$$\hat{p}_{\text{status}} = \frac{1}{1 + e^{-(3.23 + -0.078 \times \text{Age} + -1.597 \times \text{Sex})}}$$



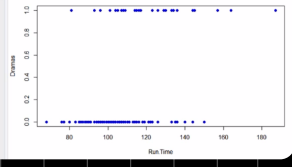
Calculating coefficients

```
glm(Status ~ Age + Sex, case2001,
   family = "binomial")
```

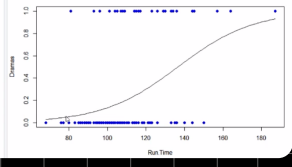
`glm()` calculates *generalized linear models*. Inputs

- Formula (same as in `lm()`)
- Data frame (same as in `lm()`)
- Model family: determines type of GLM
  - `gaussian` gives standard linear regression
  - `binomial` calculates logistic regression

```
> model <- glm(Dramas ~ Run.Time, data = movies, family = "binomial")
> summary(model)

Call:
glm(formula = Dramas ~ Run.Time, family = "binomial", data = movies)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4782  -0.6702  -0.5094  -0.3020   2.4097

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.994958   1.541436  -4.539 5.65e-06 ***
Run.Time     0.055121   0.011313   3.899 9.64e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 130.39  on 119  degrees of freedom
Residual deviance: 111.18  on 118  degrees of freedom
AIC: 115.18

Number of Fisher Scoring iterations: 4

> plot(Dramas ~Run.Time, data = movies, pch=19, col="blue")
```




```
> x <- seq(80, 180, 1)
> y <- 1 / (1 + exp(-(model$coefficients[1] + x * model$coefficients[2])))
> lines(x,y)
>
```




```
> y <- 1 / (1 + exp(-(model$coefficients[1] + x * model$coefficients[2])))
> lines(x,y)
> model <- step(glm(Dramas ~ Run.Time + Budget + USGross...., data = movies, family = "binomial"))
Start:  AIC=103.46
Dramas ~ Run.Time + Budget + USGross....

               Df Deviance    AIC
- USGross....  1   93.461 103.461
<none>             93.461 103.461
- Budget       1  106.922 112.922
- Run.Time     1  129.732 135.732

Step:  AIC=99.53
Dramas ~ Run.Time + Budget

               Df Deviance    AIC
<none>             97.531 99.531
- Budget       1  111.183 115.183
- Run.Time     1  129.718 133.718
> |
```