The Generalised Linear Model (1)

PSM 2

Bennett Kleinberg

15 Jan 2019

Welcome

Probability, Statistics & Modeling II

Lecture 2

What question do you have?

Today

- Modelling data
- Regression in general
- Linear regression
  - simple
  - multiple
- Effects in regression analysis
- Why the GLM?

Modelling data

Overall aim: make inference from sample to population.

- make assumptions about data generation process
- model specifies the data by variables

Modelling data

- Predictions
- Relationships (extraction information)

Modelling data

Case for today

Dataset 1: Terror data ("Trial and Terror dataset")

```r
load('./data/terror_data.RData')

names(terror_data)
```

```
## [1] "firstName"      "lastName"       "gender"         "case_informant
## [5] "case_sting"     "sentence"
```

```r
head(terror_data)
```

```
##             firstName    lastName gender case_informant case_sting sente
## 3             Mubarak       Hamed   male          false      false
## 11              Tarek       Makki   male          false      false
## 20        Jalal Sadat    Moheisen   male           true       true
## 21 Thirunavukkarasu Varatharasa   male           true       true
## 22           Reinhard       Rusli   male           true       true
## 23     Syed Mustajab        Shah   male           true       true
```

```r
dim(terror_data)
```

```
## [1] 471   6
```

Case for today

Dataset 2: Mass Shootings in detail (Stanford Mass Shootings in America dataset)

```r
load('./data/mass_shootings_detailed.RData')

names(smsd)
```

```
##  [1] "caseid"         "n_fatal"        "n_injured"      "date"
##  [5] "day"            "age"            "gender"         "n_guns"
##  [9] "school_related" "mental_illness"
```

```r
head(smsd)
```

```
##   caseid n_fatal n_injured       date      day age gender n_guns
## 1      1      16        32   8/1/1966   Monday  20   Male      8
## 2      2       5         1 11/12/1966 Saturday  11   Male      1
## 3      3       9        13   12/31/72   Sunday  17   Male      3
## 4      4       1         3    1/17/74 Thursday   3   Male      3
## 5      5       3         7   12/30/74   Monday   8   Male      3
## 7      7       7         2    7/12/76   Monday  34   Male      1
##   school_related mental_illness
## 1            Yes            Yes
## 2            Yes            Yes
## 3             No            Yes
## 4            Yes            Yes
## 5            Yes             No
## 7            Yes            Yes
```

```r
dim(smsd)
```

```
## [1] 182  10
```

Core idea of regression

- Model a relationship between an outcome variable and predictor variable(s)
- Find relationships in data
- Make predictions for new data

Core idea of regression

Aim: find a line that simplifies the data

Why linear?

- Simplest-model principle
- Many relationships approximate linearity
- Non-linear relationships are often linear after transformation

Regression formalised

```
Y = a + b*X + E
```

Regression formalised

- The dependent variable `Y`
- The predictor variable `X`
- The intercept `a` (= the value of `Y` if `X` is `0`)
- The slope `b` (= the change in `Y` for every unit change in `X`)
- The error term `E` (= the difference between the predicted value and the observed value)

Regression formalised

```
Y = a + b*X + E
```

Note: **linear relationship**

Regression assumptions

1. Linear relationship
2. Little multicollinearity
3. Residuals i.i.d. (independently, identically distributed)
    - `E ~ i.i.d. N(0, sd)`

Your shooter model

Modelling the no. of fatalities

```
victims = intercept + slope*number_of_guns
```

- more guns –> more victims?
- baseline victims –> 3

```
pred.victims = 3 + 1.5*smsd$n_guns
```

Your shooter model

```r
head(smsd, 1)
```

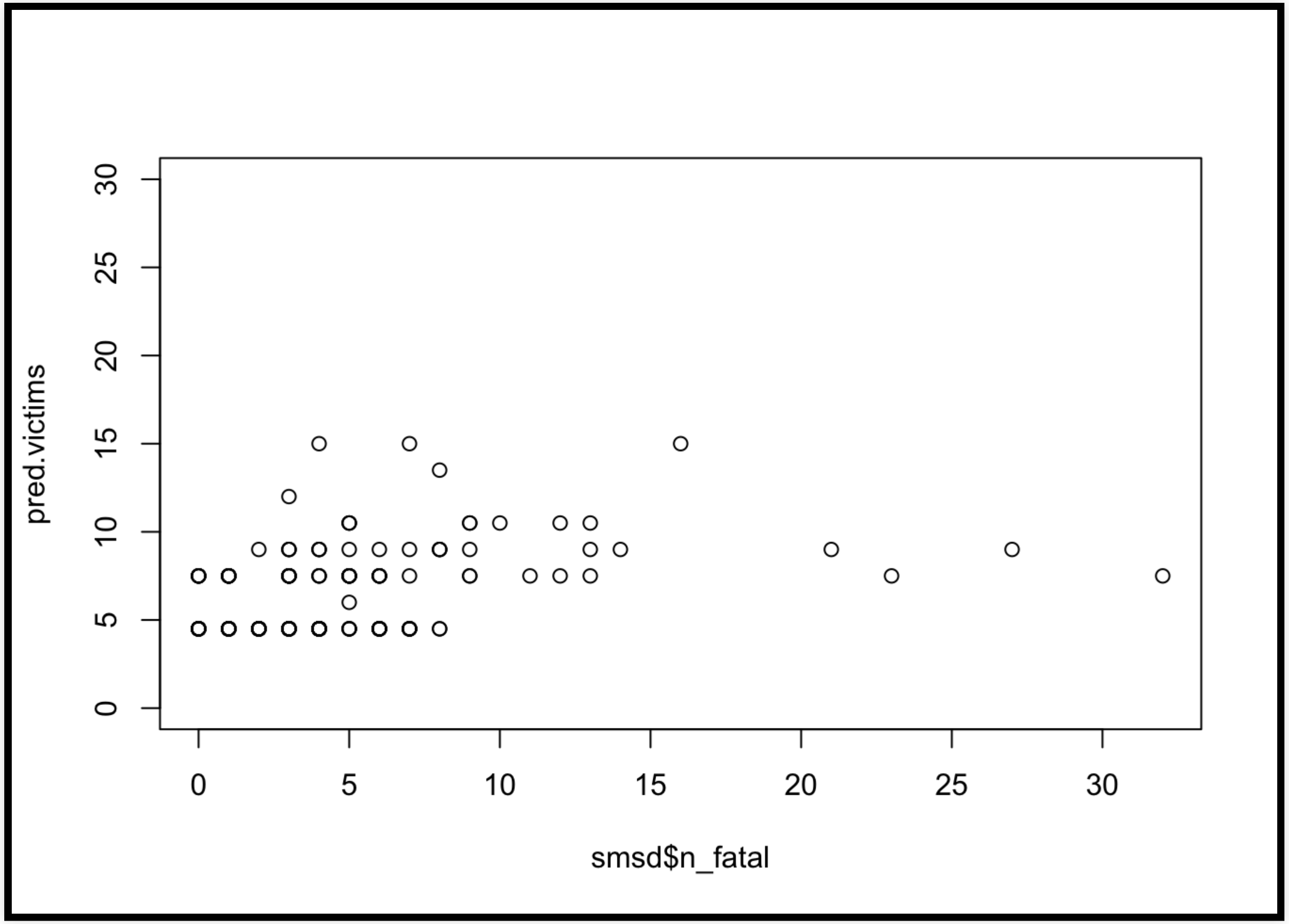
```
##   caseid n_fatal n_injured     date     day age gender n_guns
## 1      1      16        32 8/1/1966 Monday  20   Male      8
##   school_related mental_illness
## 1            Yes            Yes
```

```r
case_1 = 3+1.5*8
case_1
```

```
## [1] 15
```

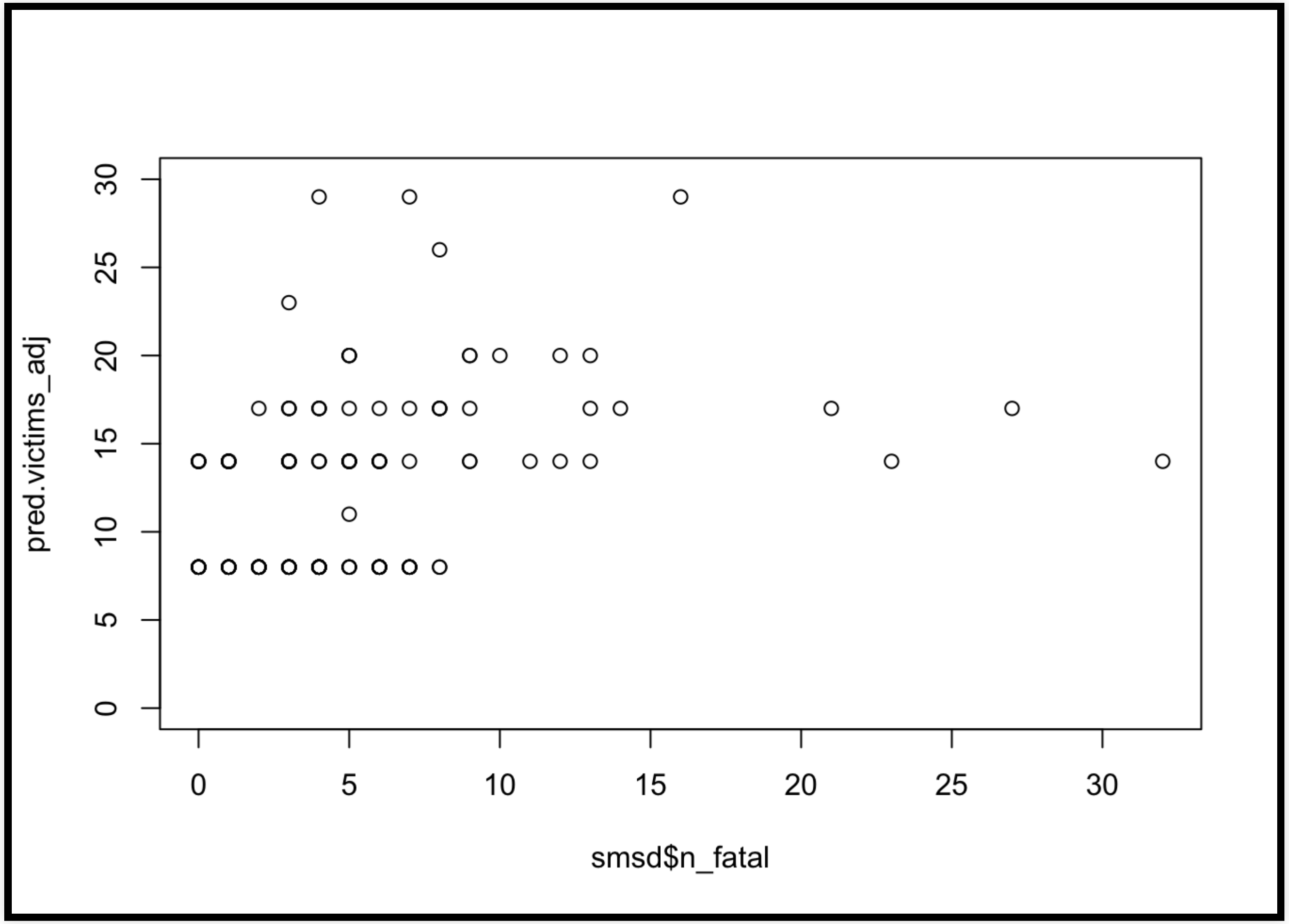
Your shooter model

```
plot(smsd$n_fatal, pred.victims, ylim=c(0,30))
```



Maybe adjust the model?

```
pred.victims_adj = 5 + 3*smsd$n_guns
plot(smsd$n_fatal, pred.victims_adj, ylim=c(0,30))
```



Shooter model

An empirical approach:

- let the model parameters be estimated from the data
- you ~~specify~~ build the model
- linearity in parameters

Linearity in parameters

```
Y = a + b*X + E
```

Linear because: `Y = a + b`

Modelling syntax in R

OK, let's model the data then…

R syntax for modelling:

- Model formula approach
- Use the `~` to say "explained through…"
- Left side: outcome variable (dependent variable)
- Right side: the model that explains the outcome variable

The shooter model

```r
shooter_model_1 = lm(formula = smsd$n_fatal ~ smsd$n_guns)
shooter_model_1
```

```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$n_guns)
##
## Coefficients:
## (Intercept)  smsd$n_guns
##       2.087        1.105
```

Understanding the model

```
shooter_model_1
```

```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$n_guns)
##
## Coefficients:
## (Intercept)  smsd$n_guns
##       2.087        1.105
```

The model equation therefore is:

```
n_fatal = 2.087 + 1.105*n_guns
```

More model info

```r
summary(shooter_model_1)
```

```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$n_guns)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9250 -2.4012 -0.4012  1.2440 26.5988
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0870     0.5268   3.962 0.000107 ***
## smsd$n_guns   1.1047     0.1981   5.577 8.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.148 on 180 degrees of freedom
## Multiple R-squared:  0.1473, Adjusted R-squared:  0.1426
## F-statistic:  31.1 on 1 and 180 DF,  p-value: 8.847e-08
```

- Statistically significant intercept
- Statistically signiifcant predictor `n_guns`

Using the model

```
n_fatal = 2.087 + 1.105*n_guns
```

So we can make predictions, right?

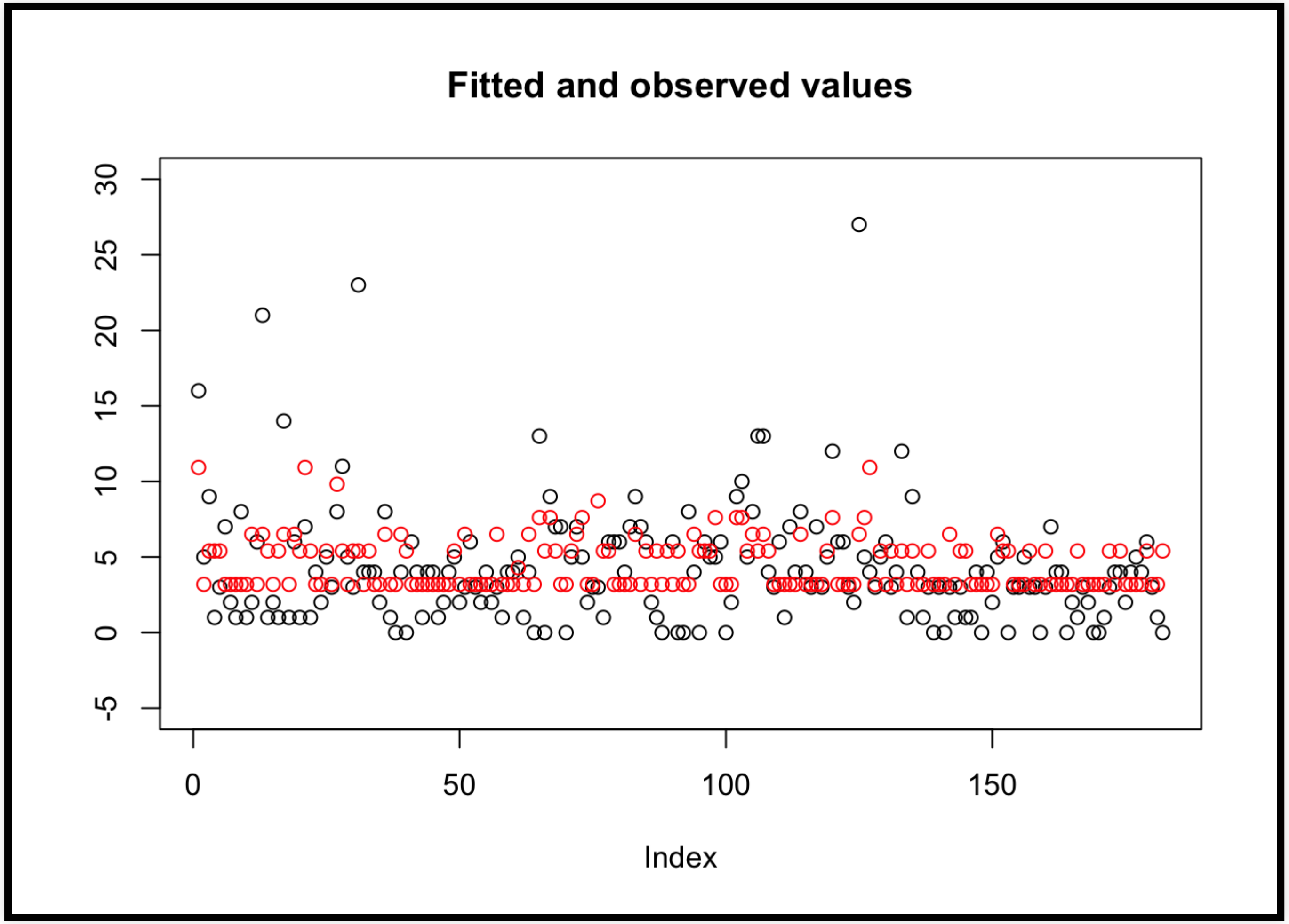
Predictions with the model

Have a look at he model object:

```
shooter_model_1$fitted.values
```

```
##         1         2         3         4         5         6         7 
## 10.924961  3.191753  5.401241  5.401241  5.401241  3.191753  3.191753 
##         8         9        10        11        12        13        14 
##  3.191753  3.191753  3.191753  6.505985  3.191753  6.505985  5.401241 
##        15        16        17        18        19        20        21 
##  3.191753  5.401241  6.505985  3.191753  6.505985  5.401241 10.924961 
##        22        23        24        25        26        27        28 
##  5.401241  3.191753  3.191753  5.401241  3.191753  9.820217  5.401241 
##        29        30        31        32        33        34        35 
##  3.191753  5.401241  5.401241  3.191753  5.401241  3.191753  3.191753 
##        36        37        38        39        40        41        42 
##  6.505985  3.191753  3.191753  6.505985  5.401241  3.191753  3.191753 
##        43        44        45        46        47        48        49 
##  3.191753  3.191753  3.191753  3.191753  3.191753  3.191753  5.401241 
##        50        51        52        53        54        55        56 
##  3.191753  6.505985  3.191753  3.191753  3.191753  3.191753  3.191753 
##        57        58        59        60        61        62        63 
##  6.505985  3.191753  3.191753  3.191753  4.296497  3.191753  6.505985
```

```
{plot(smsd$n_fatal, main="Fitted and observed values", ylab="", ylim=c(-5
points(shooter_model_1$fitted.values, col='red')}
```



What about the error term?

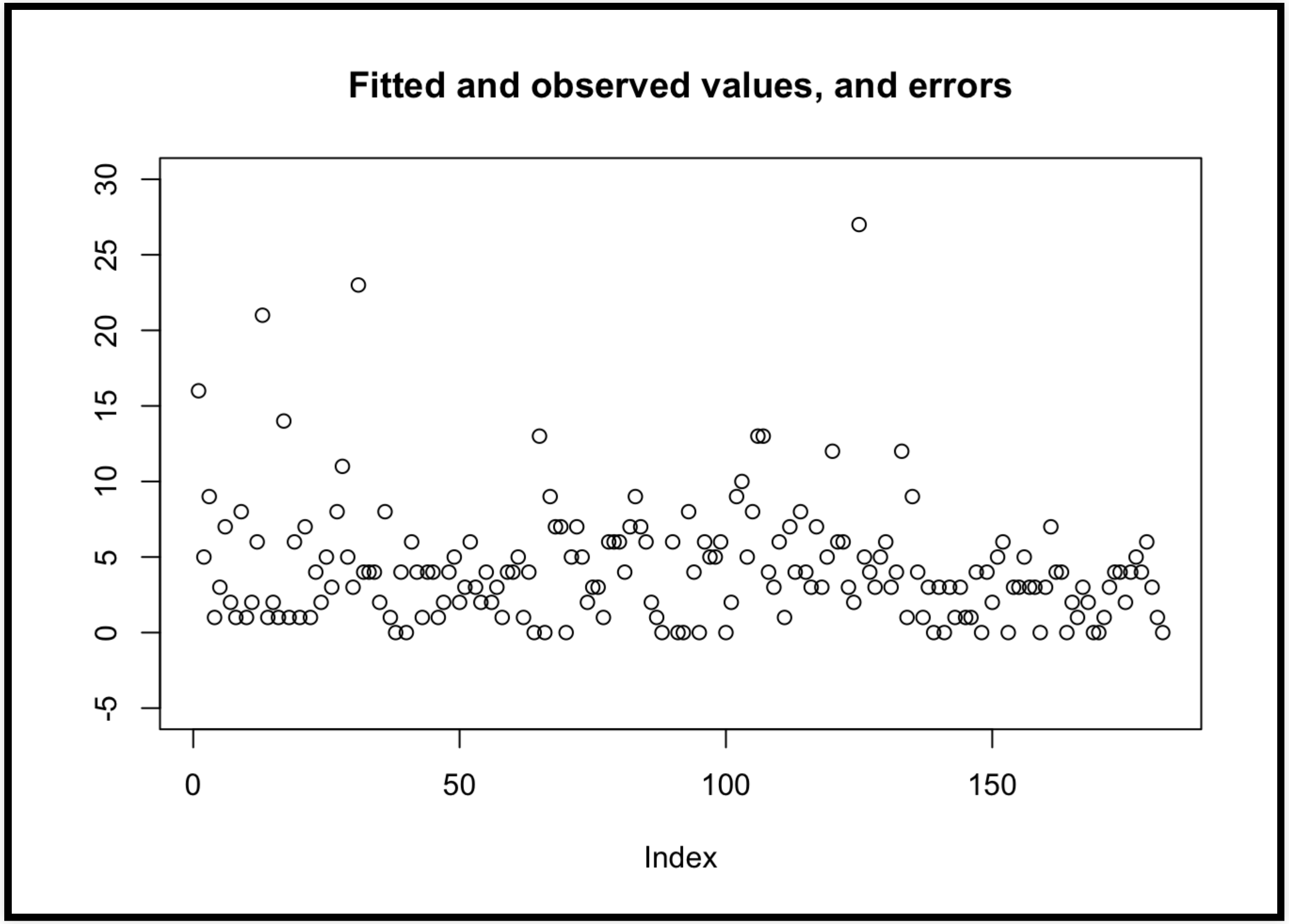
```r
head(shooter_model_1$residuals, 10)
```

```
##         1         2         3         4         5         6         7 
##  5.075039  1.808247  3.598759 -4.401241 -2.401241  3.808247 -1.191753 
##         8         9        10 
## -2.191753  4.808247 -2.191753 
```

Relationships between observed values, fitted values and errors?

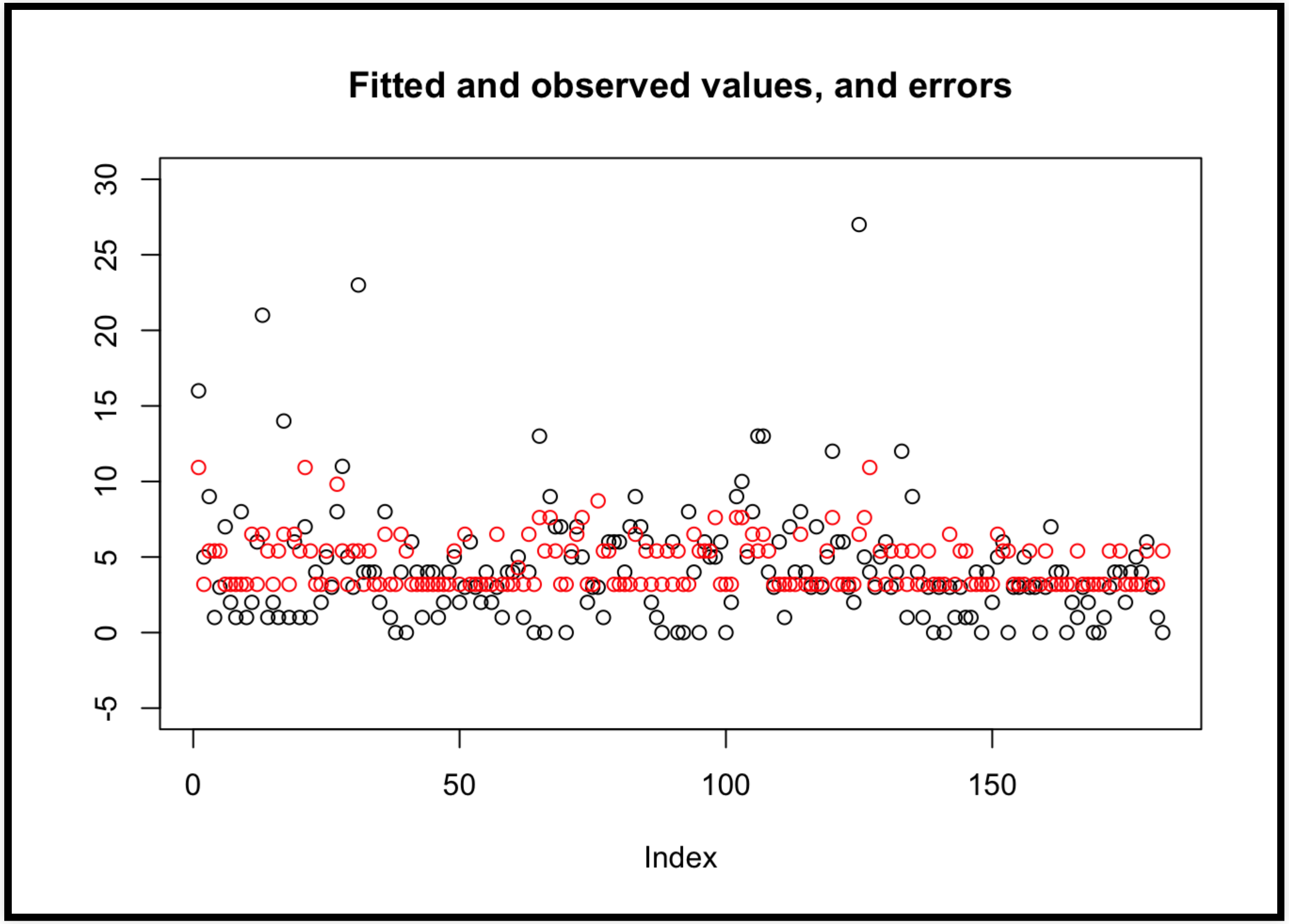
Observed values:

```
{plot(smsd$n_fatal, main="Fitted and observed values, and errors", ylab=
```

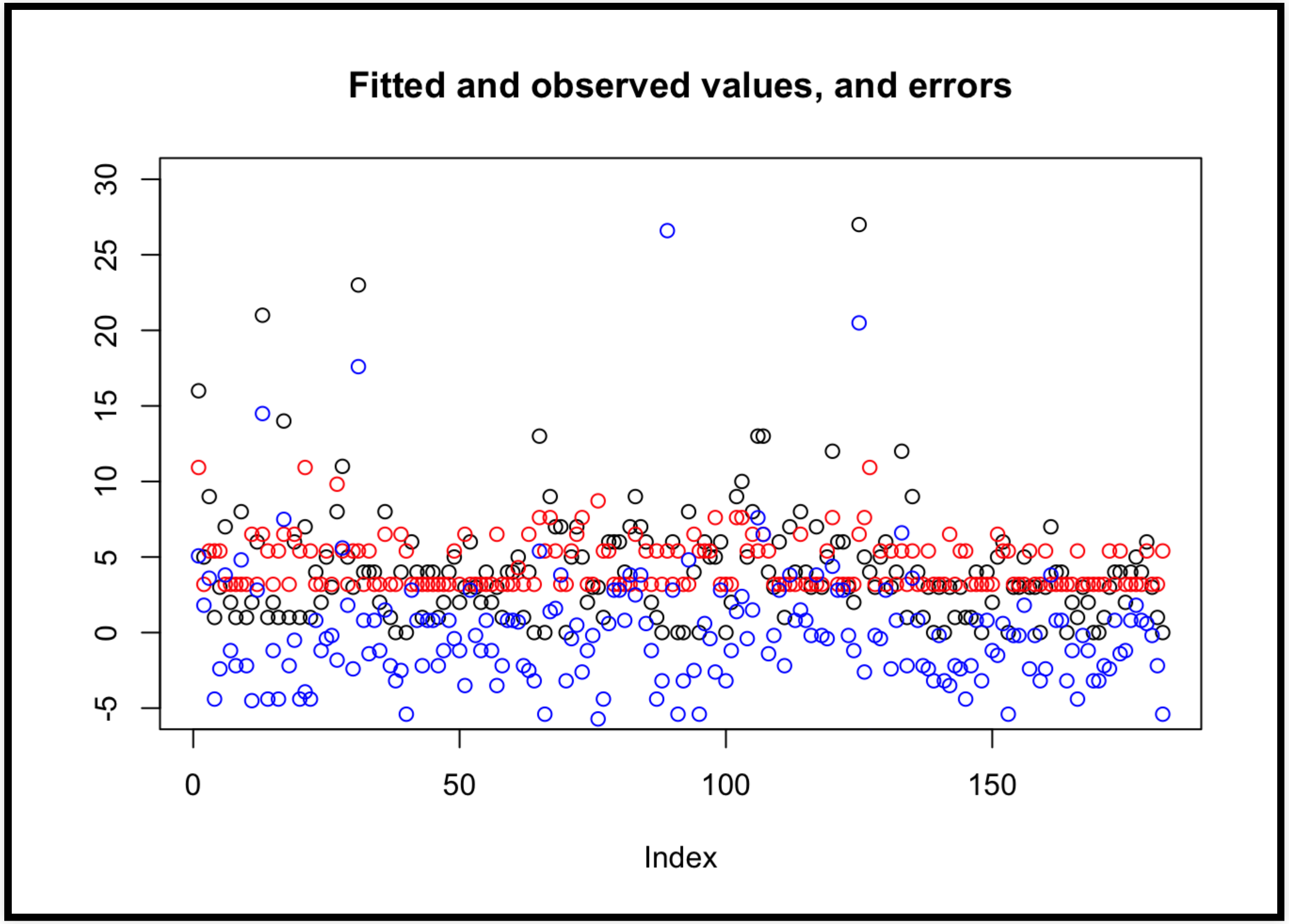


Observed + fitted values

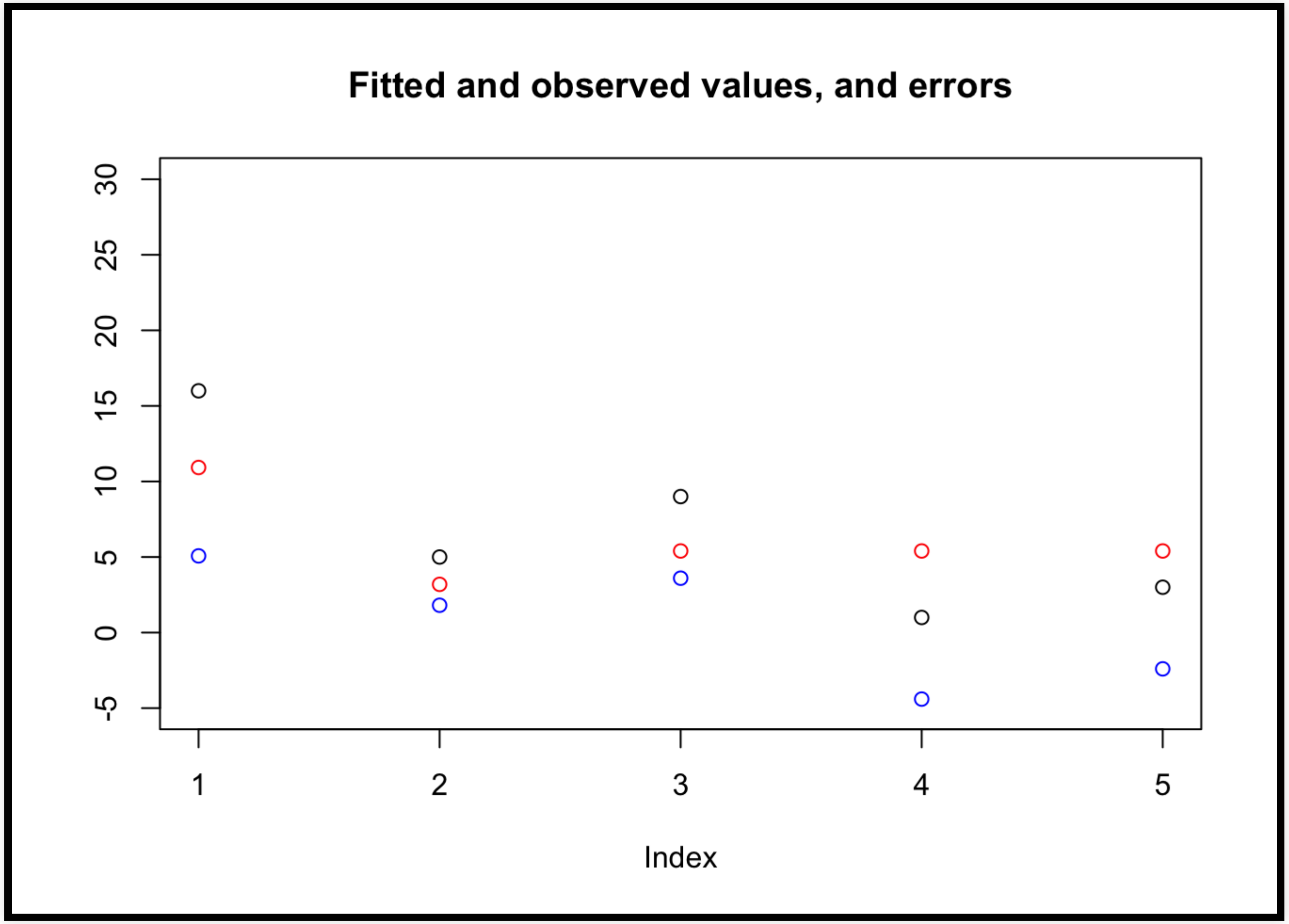
```
{plot(smsd$n_fatal, main="Fitted and observed values, and errors", ylab=
points(shooter_model_1$fitted.values, col='red')}
```



```
{plot(smsd$n_fatal, main="Fitted and observed values, and errors", ylab='
points(shooter_model_1$fitted.values, col='red')
points(shooter_model_1$residuals, col='blue')}
```



```
{plot(smsd$n_fatal[1:5], main="Fitted and observed values, and errors", y
points(shooter_model_1$fitted.values[1:5], col='red')
points(shooter_model_1$residuals[1:5], col='blue')}
```



Understanding residuals

If:

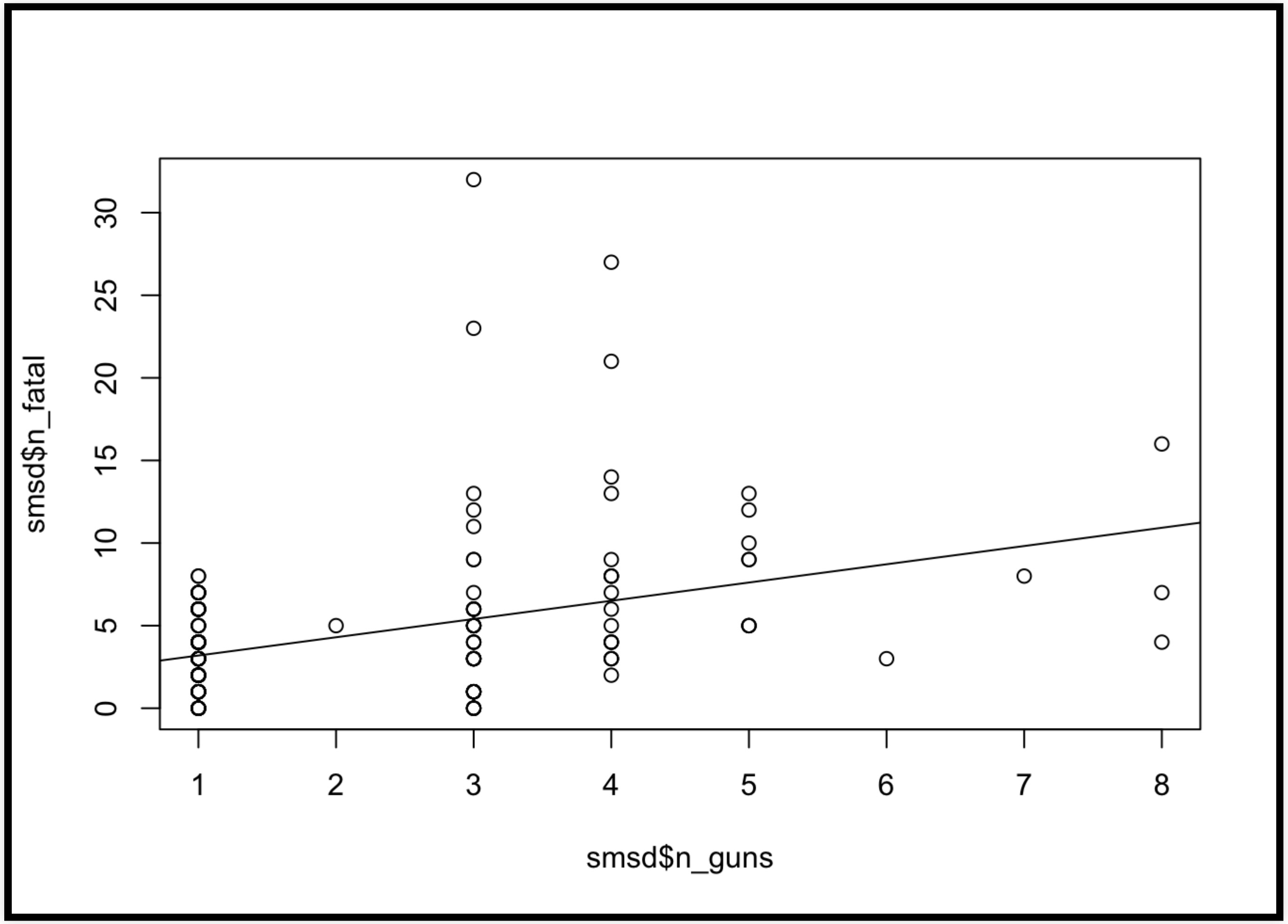
```
residual = observed - predicted
```

… then: What is the sum of residuals?

Thinking of the model graphically

Aim: find best fitting line

```
{plot(smsd$n_guns,smsd$n_fatal)
abline(shooter_model_1)}
```



smsd$n_fatal
0
5
10
15
20
25
30
1
2
3
4
5
6
7
8
smsd$n_guns

Check

```
smsd$fitted_values = shooter_model_1$fitted.values
smsd$residuals = shooter_model_1$residuals

smsd[smsd$n_guns == 6, ]
```

```
##    caseid n_fatal n_injured     date    day age gender n_guns
## 82     82       3         0 10/28/02 Monday  38   Male      6
##    school_related mental_illness fitted_values residuals
## 82            Yes             No      8.715473 -5.715473
```

What is the sum of residuals?

```r
sum(smsd$residuals)
```

```
## [1] -2.292611e-14
```

So how to tell how good the model is?

Sum of squares

```
sum(shooter_model_1$residuals^2)
```

```
## [1] 3096.339
```

Hence the name: OLS regression –> Ordinary Least Squares!

But:

… this is a shitty model!

```
victims = intercept + slope*number_of_guns
```

Adding predictors to the model

- Simple regression
  - one outcome variable
  - one predictor variable
  - one slope for the predictor variable
  - intercept
- Multiple regression
  - one outcome variable
  - **multiple** predictor variables
  - one slope for **each** predictor
  - intercept

General formula:

```
Y = b_0 + b_1*X1 + b_2*X2 + b_3*X3 ... b_i*Xi
```

Let’s add terms to out model:

Conceptual:

```
victims = b_0 + b_1*number_of_guns + b_2*mental_illness
```

What will this mean for the model’s fit?

Adding terms to the model in R

```
shooter_model_2 = lm(formula = smsd$n_fatal ~ smsd$n_guns + smsd$mental_i
shooter_model_2
```

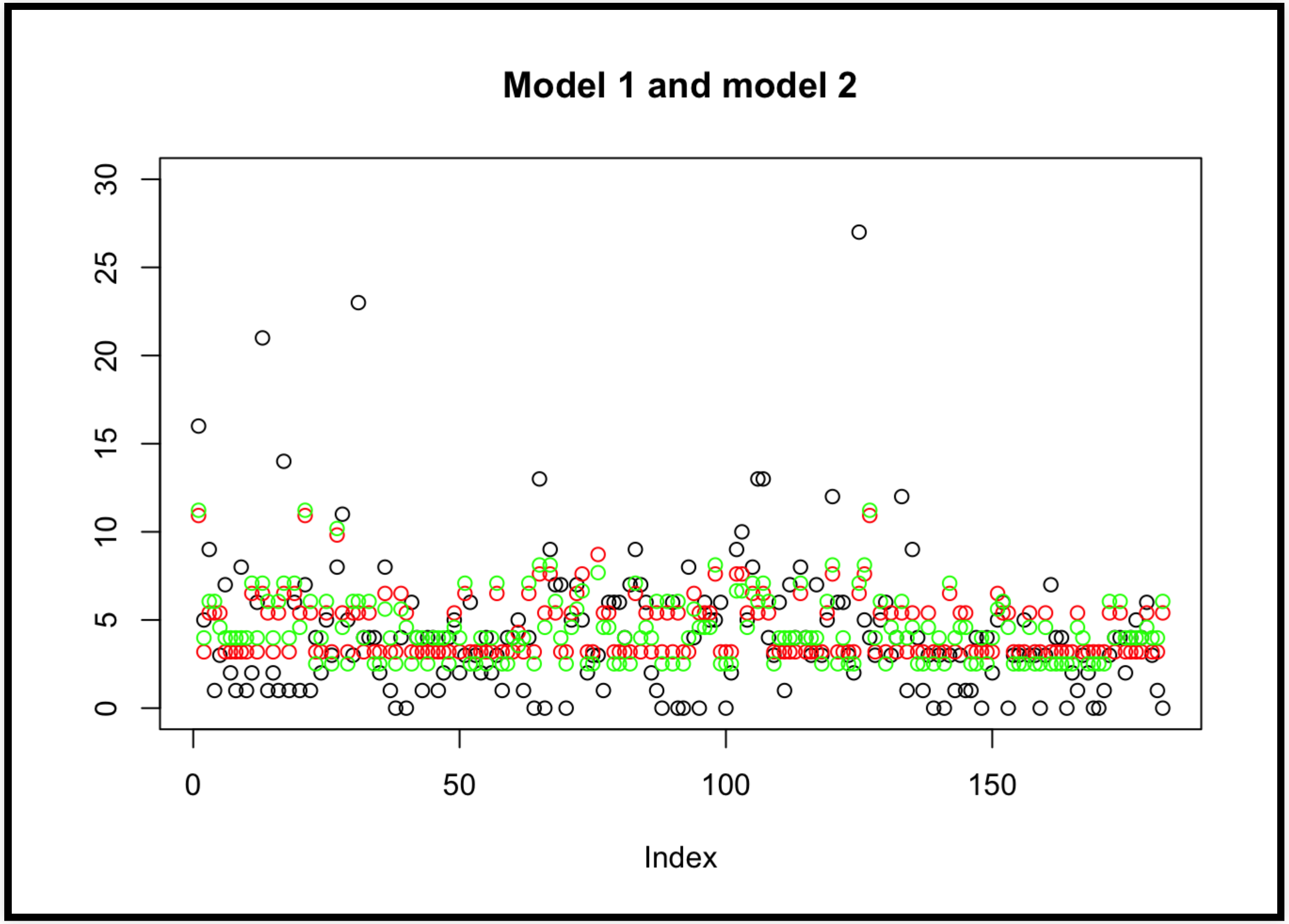
```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$n_guns + smsd$mental_illness)
##
## Coefficients:
##            (Intercept)            smsd$n_guns  smsd$mental_illnessYes
##                  1.480                  1.034                   1.471
```

->

```
n_fatal = 1.48 + 1.034*n_guns + 1.471*mentall_illness
```

Look at the predictions

```
{plot(smsd$n_fatal, main="Model 1 and model 2", ylab="", ylim=c(0, 30))
points(shooter_model_1$fitted.values, col='red')
points(shooter_model_2$fitted.values, col='green')}
```



Model 1 vs model 2

Shooter model 1:

```r
summary(shooter_model_1)
```

```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$n_guns)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9250 -2.4012 -0.4012  1.2440 26.5988
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0870     0.5268   3.962 0.000107 ***
## smsd$n_guns   1.1047     0.1981   5.577 8.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.148 on 180 degrees of freedom
## Multiple R-squared:  0.1473, Adjusted R-squared:  0.1426
## F-statistic:  31.1 on 1 and 180 DF,  p-value: 8.847e-08
```

Model 1 vs model 2

Shooter model 2:

```r
summary(shooter_model_2)
```

```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$n_guns + smsd$mental_illness)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.224 -2.514 -0.514  1.486 25.947
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.4797     0.5785   2.558   0.0114 *
## smsd$n_guns              1.0342     0.1977   5.230 4.69e-07 ***
## smsd$mental_illnessYes   1.4706     0.6141   2.395   0.0177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.094 on 179 degrees of freedom
## Multiple R-squared:  0.1738,  Adjusted R-squared:  0.1646
```

Comparing the models?

If all residuals sum to zero?

```
sum(shooter_model_1$residuals^2)
```

```
## [1] 3096.339
```

```
sum(shooter_model_2$residuals^2)
```

```
## [1] 3000.22
```

Remember: what does the 2nd model do?

Yet another model:

```
smsd = smsd[smsd$school_related != 'Killed', ]
smsd = droplevels(smsd)
shooter_model_3 = lm(smsd$n_fatal ~ smsd$mental_illness + smsd$school_re
```

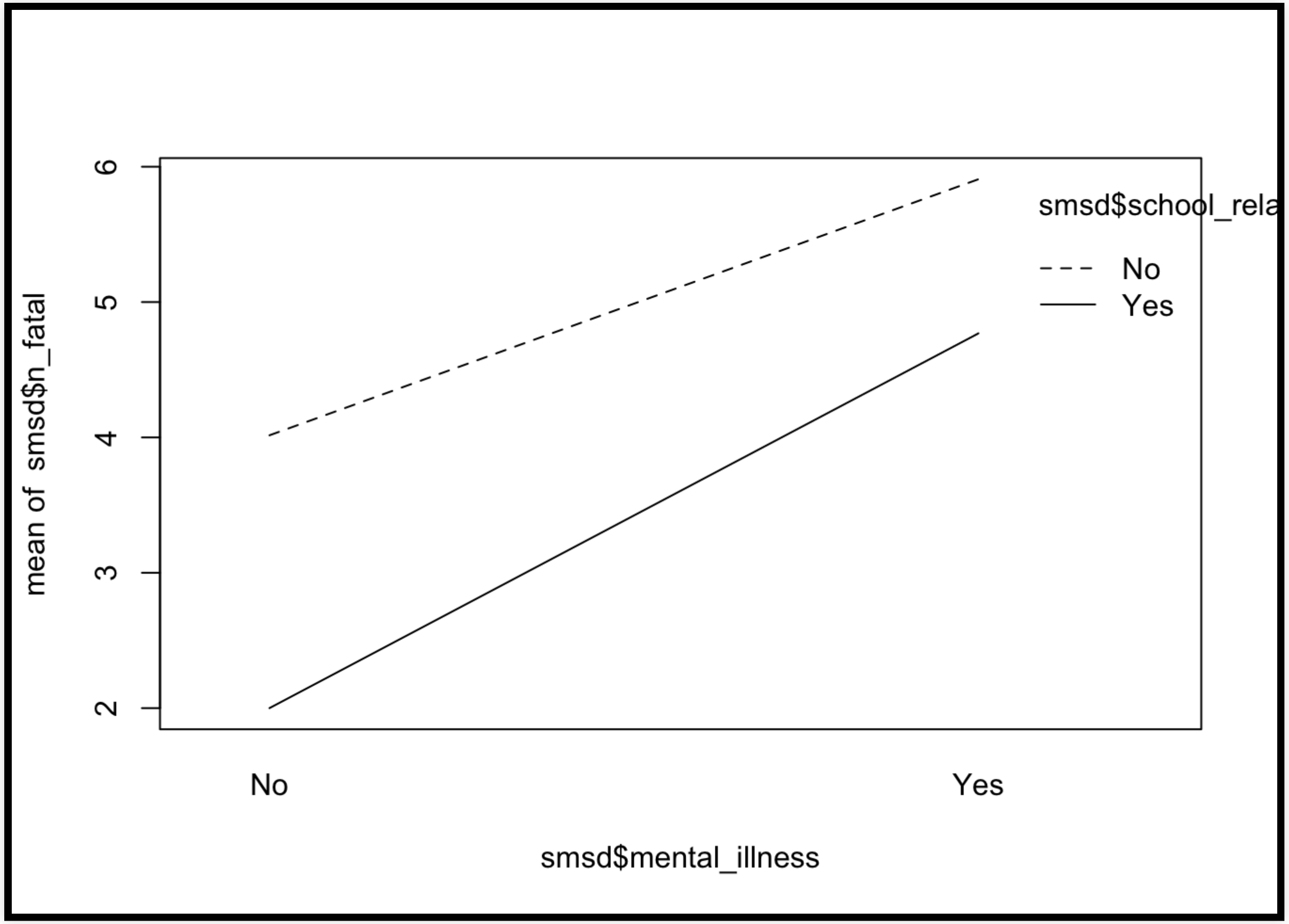
Model 3

```r
summary(shooter_model_3)
```

```
##
## Call:
## lm(formula = smsd$n_fatal ~ smsd$mental_illness + smsd$school_related
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0676 -2.5475 -0.8805  1.6396 27.4525
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.8805     0.4998   7.764 6.24e-13 ***
## smsd$mental_illnessYes   2.1871     0.6543   3.343  0.00101 **
## smsd$school_relatedYes  -1.5201     0.6865  -2.214  0.02808 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.347 on 178 degrees of freedom
## Multiple R-squared:  0.07356,    Adjusted R-squared:  0.06315
```

What does it do?

```
interaction.plot(smsd$mental_illness, smsd$school_related, smsd$n_fatal)
```



Main effects: effeect of one predictor variable on the outcome variable.

A new case: Trial and Terror Data

```r
names(terror_data)
```

```
## [1] "firstName"       "lastName"        "gender"          "case_informant
## [5] "case_sting"      "sentence"
```

Let’s start modelling

```
baseline_model = lm(terror_data$sentence ~ terror_data$gender)
```

Baseline model

```
summary(baseline_model)
```

```
##
## Call:
## lm(formula = terror_data$sentence ~ terror_data$gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -138.96  -97.96  -37.96   43.04 1007.50
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               151.963      6.762  22.474   <2e-16 ***
## terror_data$genderfemale  -41.463     24.457  -1.695   0.0907 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 141 on 469 degrees of freedom
## Multiple R-squared:  0.006091,   Adjusted R-squared:  0.003972
## F-statistic: 2.874 on 1 and 469 DF,  p-value: 0.09068
```

No effect!

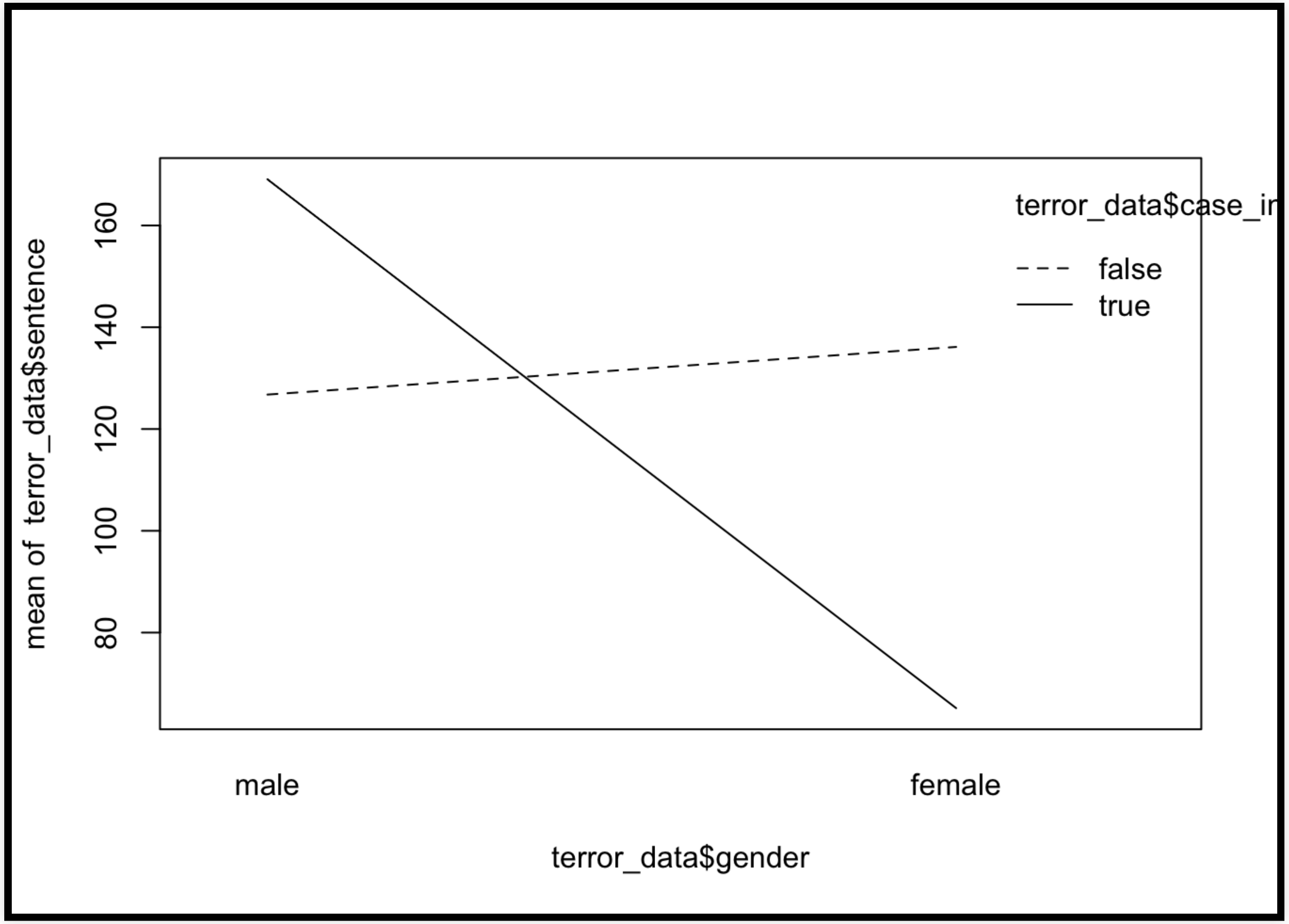
Add another variable

Extended model 1:

```
extended_model_1 = lm(terror_data$sentence ~ terror_data$gender + terror_
summary(extended_model_1)
```

```
##
## Call:
## lm(formula = terror_data$sentence ~ terror_data$gender + terror_data$c
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -152.72 -100.72  -39.72   62.78 1019.78
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     131.72      10.33  12.747   <2e-16 **
## terror_data$genderfemale        -33.50      24.51  -1.367   0.1723
## terror_data$case_informanttrue   34.00      13.18   2.579   0.0102 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.2 on 468 degrees of freedom
## Multiple R-squared:  0.02002,    Adjusted R-squared:  0.01583
```

```
interaction.plot(terror_data$gender, terror_data$case_informant, terror_
```



What’s going on??????

Interaction effects

Statistical interaction: effect of one predictor variable on the outcome variable **depends on another predictor variable**.

Adding interaction terms

```r
extended_model_2 = lm(terror_data$sentence ~ terror_data$gender + terror_
summary(extended_model_2)
```

```
##
## Call:
## lm(formula = terror_data$sentence ~ terror_data$gender + terror_data$c
##     terror_data$gender:terror_data$case_informant)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -156.08 -100.77  -39.08   64.88  981.87
##
## Coefficients:
##                                                     Estimate
## (Intercept)                                          126.767
## terror_data$genderfemale                               9.363
## terror_data$case_informanttrue                        42.318
## terror_data$genderfemale:terror_data$case_informanttrue -113.294
##                                                     Std. Error t v
## (Intercept)                                             10.521  12
## terror_data$genderfemale                                30.947   (
```

Looking at the numbers

Main effect of `case_informant`:

```
tapply(terror_data$sentence, list(terror_data$case_informant), mean)
```

```
##     false     true
## 127.8492 164.1176
```

Interpretation?

Looking at the numbers

Main effect of `gender`:

```
tapply(terror_data$sentence, list(terror_data$gender), mean)
```

```
##      male   female
## 151.9632 110.5000
```

Looking at the numbers

Interaction between `case_informant` and `gender`:

```
tapply(terror_data$sentence, list(terror_data$gender, terror_data$case_i
```

```
##              false      true
## male    126.7670 169.08494
## female 136.1304  65.15385
```

What if just want all terms in there?

- main effects
- interaction effects
- (higher order interactions)

Specify the full model with `*`

```
lm(terror_data$sentence ~ terror_data$gender + terror_data$case_informant
```

```
##
## Call:
## lm(formula = terror_data$sentence ~ terror_data$gender + terror_data$
##     terror_data$gender:terror_data$case_informant)
##
## Coefficients:
##                                              (Intercept)
##                                                  126.767
##                                  terror_data$genderfemale
##                                                    9.363
##                             terror_data$case_informanttrue
##                                                   42.318
## terror_data$genderfemale:terror_data$case_informanttrue
##                                                 -113.294
```

```
#identical to:

lm(terror_data$sentence ~ terror_data$gender*terror_data$case_informant)
```

```
##
## Call:
## lm(formula = terror_data$sentence ~ terror_data$gender * terror_data$c
##
## Coefficients:
##                                              (Intercept)
##                                                  126.767
##                                 terror_data$genderfemale
##                                                    9.363
##                            terror_data$case_informanttrue
##                                                   42.318
## terror_data$genderfemale:terror_data$case_informanttrue
##                                                 -113.294
```

Maybe we can optimise this?

What if you don't know what the 'ideal' model is?

Especially neat for predictive modelling

Back to the shooting data:

```
names(smsd)
```

```
##  [1] "caseid"         "n_fatal"        "n_injured"      "date"
##  [5] "day"            "age"            "gender"         "n_guns"
##  [9] "school_related" "mental_illness" "fitted_values"  "residuals"
```

Automated variable selection

1. Specify the complete model

```
complete_model = lm(n_fatal ~ gender*n_guns*mental_illness*school_relatec
```

4 predictor variables: how many terms in the model?

Automated variable selection

1. Specify the complete model

```
complete_model = lm(n_fatal ~ n_guns*mental_illness*school_related, data
```

2. Specify the null model

```
null_model = lm(n_fatal ~ 1, data = smsd)
```

3. Run model selection …

3 predictor variables: how many terms in the model?

- 1 intercept
- 3 main effects
- 3 2-way interactions
- 1 3-way interaction

Model selection

```r
summary(complete_model)
```

```
##
## Call:
## lm(formula = n_fatal ~ n_guns * mental_illness * school_related,
##     data = smsd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9592 -2.1233 -0.6777  1.2421 26.2074
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                               2.30041    0.93210   2.46
## n_guns                                    0.86436    0.39577   2.18
## mental_illnessYes                         1.47991    1.28991   1.14
## school_relatedYes                        -0.01274    1.70127  -0.00
## n_guns:mental_illnessYes                  0.03300    0.49495   0.06
## n_guns:school_relatedYes                 -1.02874    0.77367  -1.33
## mental_illnessYes:school_relatedYes      -3.41734    2.24208  -1.52
```

Model selection

```
summary(null_model)
```

```
##
## Call:
## lm(formula = n_fatal ~ 1, data = smsd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4751 -2.4751 -0.4751  1.5249 27.5249
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4751     0.3338    13.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.491 on 180 degrees of freedom
```

Model selection: backward

```r
step(complete_model, direction = 'backward')
```

```
## Start:  AIC=511.13
## n_fatal ~ n_guns * mental_illness * school_related
##
##                                        Df Sum of Sq    RSS    AIC
## <none>                                              2790.6 511.13
## - n_guns:mental_illness:school_related  1    72.456 2863.0 513.77
```

```
##
## Call:
## lm(formula = n_fatal ~ n_guns * mental_illness * school_related,
##     data = smsd)
##
## Coefficients:
##                                 (Intercept)
##                                     2.30041
##                                      n_guns
##                                     0.86436
##                           mental_illnessYes
##                                     1.47991
##                           school_relatedYes
##                                    -0.01274
##                    n_guns:mental_illnessYes
##                                     0.03300
##                    n_guns:school_relatedYes
##                                    -1.02874
```

Model selection: forward

```r
step(null_model, direction = 'forward'
     , scope=list(lower=null_model, upper=complete_model))
```

```
## Start:  AIC=544.78
## n_fatal ~ 1
##
##                    Df Sum of Sq    RSS    AIC
## + n_guns            1    535.46 3095.7 517.91
## + mental_illness    1    174.44 3456.7 537.87
## + school_related    1     55.94 3575.2 543.97
## <none>                          3631.1 544.78
##
## Step:  AIC=517.91
## n_fatal ~ n_guns
##
##                    Df Sum of Sq    RSS    AIC
## + mental_illness    1    95.460 3000.2 514.24
## + school_related    1    57.394 3038.3 516.52
## <none>                          3095.7 517.91
##
## Step:  AIC=514.24
```

```
##
## Call:
## lm(formula = n_fatal ~ n_guns + mental_illness + school_related +
##     n_guns:mental_illness, data = smsd)
```

```
##
## Coefficients:
##              (Intercept)                    n_guns
##                   2.7122                    0.6060
##        mental_illnessYes         school_relatedYes
##                   0.3975                   -1.5038
## n_guns:mental_illnessYes
##                   0.6247
```

Model selection: bi-directional

```
step(null_model, direction = 'both'
     , scope=list(upper=complete_model))
```

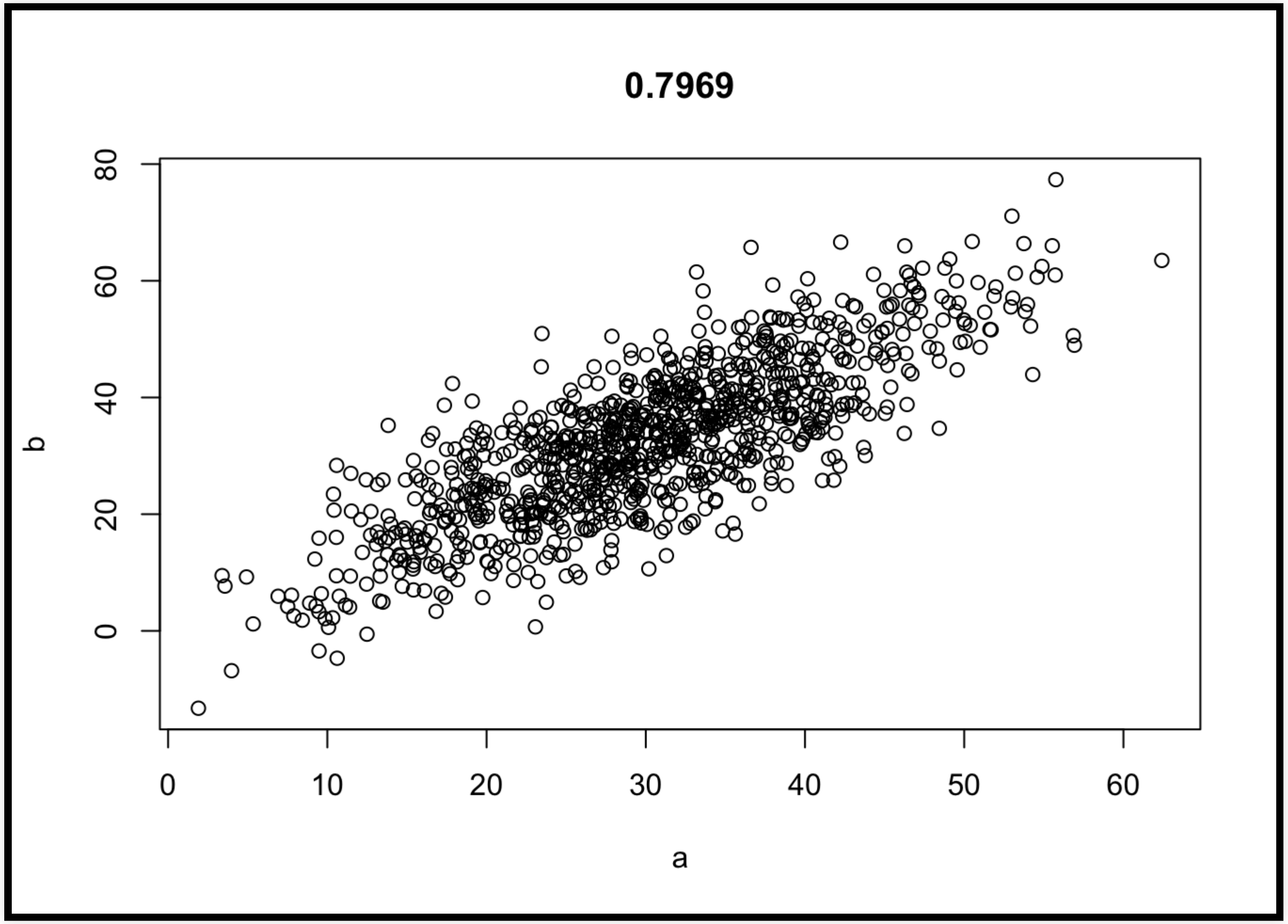
```
## Start:  AIC=544.78
## n_fatal ~ 1
##
##                   Df Sum of Sq    RSS    AIC
## + n_guns           1    535.46 3095.7 517.91
## + mental_illness   1    174.44 3456.7 537.87
## + school_related   1     55.94 3575.2 543.97
## <none>                         3631.1 544.78
##
## Step:  AIC=517.91
## n_fatal ~ n_guns
##
##                   Df Sum of Sq    RSS    AIC
## + mental_illness   1     95.46 3000.2 514.24
## + school_related   1     57.39 3038.3 516.52
## <none>                         3095.7 517.91
## - n_guns           1    535.46 3631.1 544.78
##
```

```
##
## Call:
## lm(formula = n_fatal ~ n_guns + mental_illness + school_related +
##     n_guns:mental_illness, data = smsd)
```

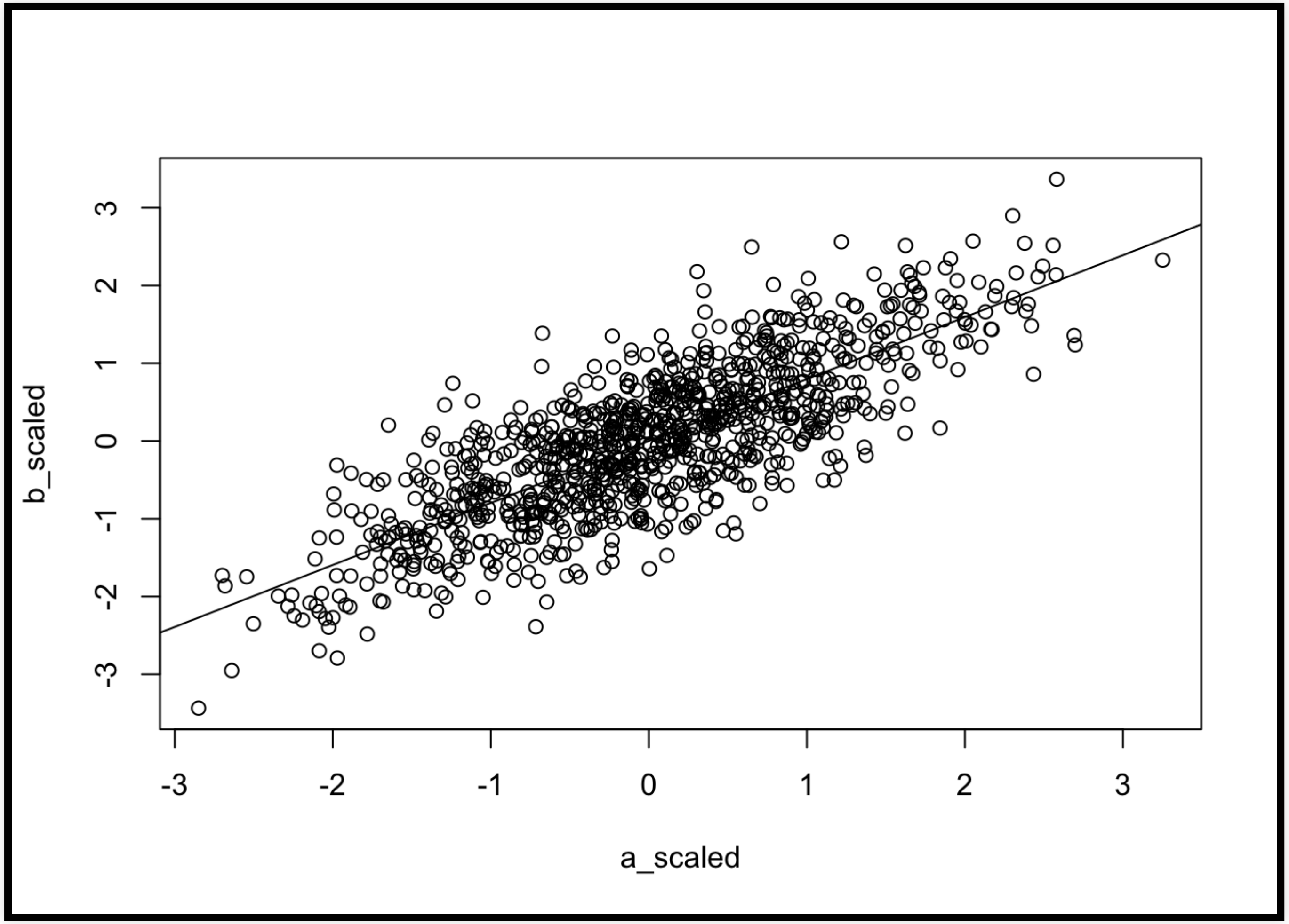
```
##
## Coefficients:
##            (Intercept)                   n_guns
##                 2.7122                   0.6060
##      mental_illnessYes        school_relatedYes
##                 0.3975                  -1.5038
## n_guns:mental_illnessYes
##                 0.6247
```

Limitations of linear regression?

```r
set.seed(123)
a = rnorm(1000, 30, 10)
b = a + rnorm(1000, 2, 8)
plot(a, b, main = round(cor(a, b), 4))
```


0.7969
80
60
40
20
0
b
0
10
20
30
40
50
60
a

```
a_scaled = scale(a)
b_scaled = scale(b)
{plot(a_scaled, b_scaled)
abline(lm(a_scaled ~ b_scaled))}
```



```r
lm(a_scaled ~ b_scaled - 1)
```

```
##
## Call:
## lm(formula = a_scaled ~ b_scaled - 1)
##
## Coefficients:
## b_scaled
##   0.7969
```

Limitations of linear regression?

- Correlation != causation
- **Continuous outcome variable**

Generalising the model

The Generalised Linear Model

Connections to machine learning

- Regression the best starting point
- Core difference: explanatory modelling vs predictive modelling
- More care against overfitting in predictive modelling
- Split the data

RECAP

- Simple regression with intercept, slope, error term
- Extended to multiple regression
- Main effects & interactions
- Model selection
- How to extend to other outcome variables?

Outlook

Next week

- More on the GLM
- Extended cases
- How good is the model?
- How does a model compare to another?

Homework

- Regression modelling in R

END