Regression modelling part 2

1 Regression modelling with one numerical and one categorical explanatory variable

Let's expand upon what we learned last week by revisiting the instructor evaluation data set `evals`. In Week 3 you were tasked with examining the relationship between teaching score (`score`) and age (`age`). Now, let's also introduce the additional (binary) categorical explanatory variable gender (`gender`). That is, we we will be examining:

- the teaching score (`score`) as our outcome variable y;
- age (`age`) as our numerical explanatory variable x_1; and
- gender (`gender`) as our categorical explanatory variable x_2.

The data can be downloaded below:

You can download today's session R script below:

1.1 Exploratory data analysis

Before we begin, lets load the following packages into R:

```
library(tidyverse)     # Data wrangling
library(ggplot2)       # Data visualization
library(performance)   # Model assessment
library(skimr)         # Exploratory analysis
library(sjPlot)        # Plot and tables for linear models
library(broom)         # Linear model tidy summaries
```

Now, lets start by subsetting the `evals` data set so that we only have the variables we are interested in, that is, `score`, `age` and `gender`.

> **Note**
>
> It is best to give your new data set a different name than `evals` as to not overwrite the original `evals` data set. Your new data set should look like the one below.

First, we read the data using the `read.csv()` function with `stringsAsFactors = TRUE` to automatically convert categorical variables into factors:

```
evals <- read.csv("evals.csv",stringsAsFactors = T)
eval.score <- evals %>%
  dplyr::select(c(score,age,gender))
```

> **Task 1**
>
> You can also view your data set using the `glimpse` function, or by opening a spreadsheet view in RStudio using the `View` function. Use the `skim` function to obtain some summary statistics from our data.

Click here to see the solution

```
eval.score %>%
  skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	463
Number of columns	3
Column type frequency:	
factor	1
numeric	2
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	mal: 268, fem: 195

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
score	0	1	4.17	0.54	2.3	3.8	4.3	4.6	5	□□□□□
age	0	1	48.37	9.80	29.0	42.0	48.0	57.0	73	□□□□□

Now, let's compute the correlation between our outcome variable `score` and our numerical explanatory variable `age`:

```
eval.score %>%
  summarise(rho = cor(score,age))
```

```
       rho
1 -0.107032
```

Question

Why do we not include the categorical variable `gender` when calculating the correlation?
Answer
The correlation coefficient only exists between numerical variables, which is why we do not include our categorical variable `gender`.

Furthermore, we can obtain the correlation for each gender as follows:

```
eval.score %>%
  summarise(rho = cor(score,age),
```

```
            .by = gender)
```

```
  gender         rho
1 female -0.26517575
2   male -0.07645422
```

From this we can tell that the negative linear association between age and teaching score appears to be larger for women than it does for men, i.e. the teaching score of women drops faster with age.

We can now visualize our data by producing a scatterplot, where seeing as we have the categorical variable `gender`, we shall plot the points using different colours for each gender:

```
ggplot(eval.score,
       aes(x = age, y = score, color = gender)) +
  geom_jitter() +
  labs(x = "Age", y = "Teaching Score", color = "Gender")
```

Figure 1: Instructor evaluation scores by age and gender. The points have been jittered.

> **Note**
>
> The above code has jittered the points, however, this is not necessary and `geom_point` would suffice. To plot separate points by gender we simply add the `color` argument to the `aes` function and pass to it `gender`.

From the scatterplot we can see that there are very few women over the age of 60 in our data set and that the variability for women seems to be slighlty larger than for men.

1.2 Multiple regression: parallel slopes model

Here, we shall begin by fitting what is referred to as a parallel regression lines model. This model implies that the slope of relationship between teaching score (`score`) and age (`age`) is the same for both males and females, with only the intercept of the regression lines changing. Hence, our parallel regression lines model is given as:

$$\begin{aligned} y_i &= \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \epsilon_i \\ &= \alpha + \beta_{\text{age}} \cdot \text{age} + \beta_{\text{male}} \cdot \mathbb{I}_{\text{male}}(x) + \epsilon_i, \end{aligned}$$

where

- α is the intercept of the regression line for females;
- β_{age} is the slope of the regression line for both males and females;
- β_{male} is the additional term added to α to get the intercept of the regression line for males; and
- $\mathbb{I}_{\text{male}}(x)$ is an indicator function such that

$$\mathbb{I}_{\text{male}}(x) = \begin{cases} 1 & \text{if gender } x \text{ is male,} \\ 0 & \text{Otherwise.} \end{cases}$$

We can fit the parallel regression lines model as follows:

```
par.model <- lm(score ~ age + gender, data = eval.score)
tab_model(par.model,show.ci = F)
```

	score	
Predictors	Estimates	p
(Intercept)	4.48	**<0.001**
age	-0.01	**0.001**
gender [male]	0.19	**<0.001**
Observations	463	
R^2 / R^2 adjusted	0.039 / 0.035	

Hence, the regression line for females is given by:

$$\widehat{\text{score}} = 4.48 - 0.01 \cdot \text{age},$$

while the regression line for males is given by:

$$\widehat{\text{score}} = 4.48 - 0.01 \cdot \text{age} + 0.191 = 4.671 - 0.009 \cdot \text{age}.$$

Now, let's superimpose our parallel regression lines onto the scatterplot of teaching score against age. To do so we will use the `plot_model` function from the `sjPlot` library as follows:

```
plot_model(model = par.model,                    # The fitted model
           type="pred",                  # type of plot
           terms = c("age","gender"), # terms to include in the plot
```

```
            grid=T,                    # split into a grid
            show.data = T,             # show observations
            jitter=T,                  # jitter the points
            ci.lvl = NA,               # show/hide confidence intervals
            title="Parallel Regression Model fitted lines") # Plot title
```

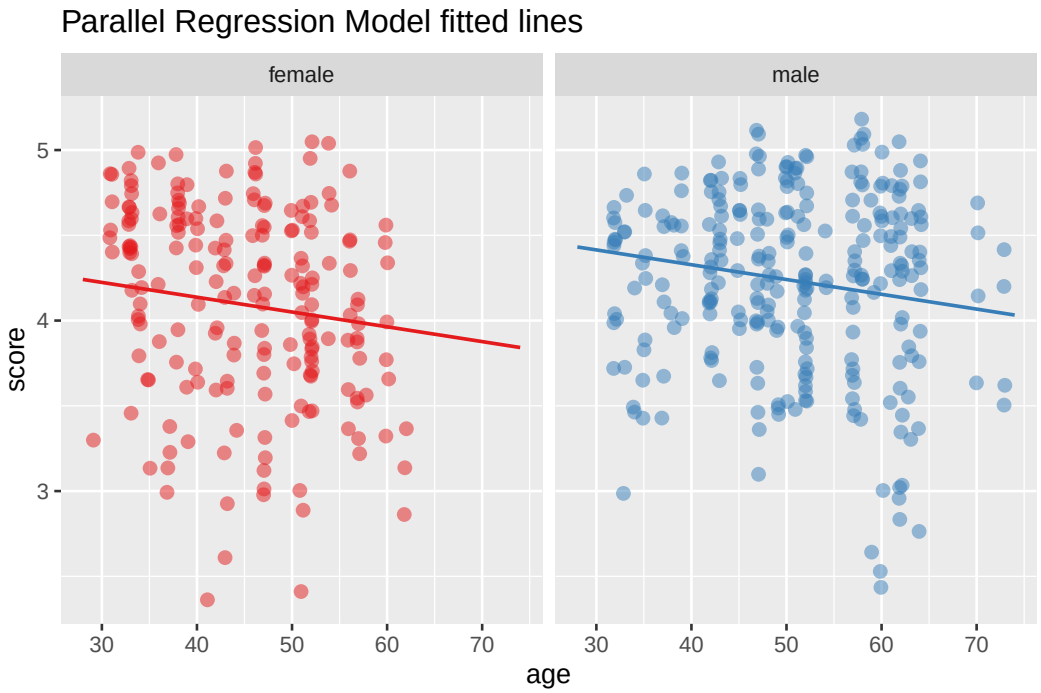


Figure 2: Instructor evaluation scores by age and gender with parallel regression lines superimposed.

Here is a short description of the `plot_model` function arguments we have used:

- `model` is the fitted `lm`-class model.
- `type = "pred"` plot the predicted values for specific model terms.
- `terms = c("age","gender")` the terms to be plotted
- `grid=T` logical argument to plot the different fitted lines for each group (i.e., gender) on different panels
- `show.data=T` and `jitter=T` logical arguments to show our observations and jitter the data points.
- `ci.lvl` set to `NA` to hide the confidence bands (we will talk more about this later)
- `title` title for the plot.

From the parallel regression lines model both males and females have the same slope, that is, the associated effect of age on teaching score is the same for both men and women. Hence, for every one year increase in age, there is an associated decrease in teaching score of 0.009. However, male instructors have a higher intercept term, that is, there is a vertical bump in the regression line for males in teaching scores. This is linked to the average difference in teaching scores that males obtain relative to females.

1.3 Multiple regression: interaction model

There is an *interaction effect* if the associated effect of one variable depends on the value of another variable. For example, the effect of age here will depend on whether the instructor is male or female, that is, the effect of age on teaching scores will differ by gender. The interaction model can be written as:

$$\begin{aligned} y_i &= \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i \\ &= \alpha + \beta_{\text{age}} \cdot \text{age} + \beta_{\text{male}} \cdot \mathbb{I}_{\text{male}}(x) + \beta_{\text{age, male}} \cdot \text{age} \cdot \mathbb{I}_{\text{male}}(x) + \epsilon_i, \end{aligned}$$

where $\beta_{\text{age, male}} \cdot \text{age} \cdot \mathbb{I}_{\text{male}}(x)$ corresponds to the interaction term.

In order to fit an interaction term within our regression model we replace the + sign with the $*$ sign as follows:

```
int.model <-lm(score ~ age * gender, data = eval.score)
tab_model(int.model,show.ci = F)
```

	score	
Predictors	Estimates	p
(Intercept)	4.88	**<0.001**
age	-0.02	**<0.001**
gender [male]	-0.45	0.094
age × gender [male]	0.01	**0.015**
Observations	463	
R^2 / R^2 adjusted	0.051 / 0.045	

Hence, the regression line for females is given by:

$$\widehat{\text{score}} = 4.88 - 0.018 \cdot \text{age},$$

while the regression line for males is given by:

$$\widehat{\text{score}} = 4.88 - 0.018 \cdot \text{age} - 0.446 + 0.014 \cdot \text{age} = 4.434 - 0.004 \cdot \text{age}.$$

> Question
>
> How do they compare with the teaching score values from the parallel regression lines model?
> Answer
> Here, we can see that, although the intercept for male instructors may be lower, the associated average decrease in teaching score with age (0.004) is not as severe as it is for female instructors (0.018).

We can plot the fitted model as we did before with the `plot_model` function. Lets try it now without the panel option (i.e., set `grid = F`) and selecting our color scheme:

```
plot_model(model = int.model,          # The fitted intertaction model
           type=  "pred",               # type of plot
           terms = c("age","gender"), # terms to include in the plot
           grid=F,                    # split into a grid
           show.data = T,             # show observations
```

```
            jitter=T,                  # jitter the points
            ci.lvl = NA,               # show/hide confidence intervals
            title ="Gender-age interaction model fitted lines", # Plot title
            colors = c("purple", "orange")) # color scheme
```

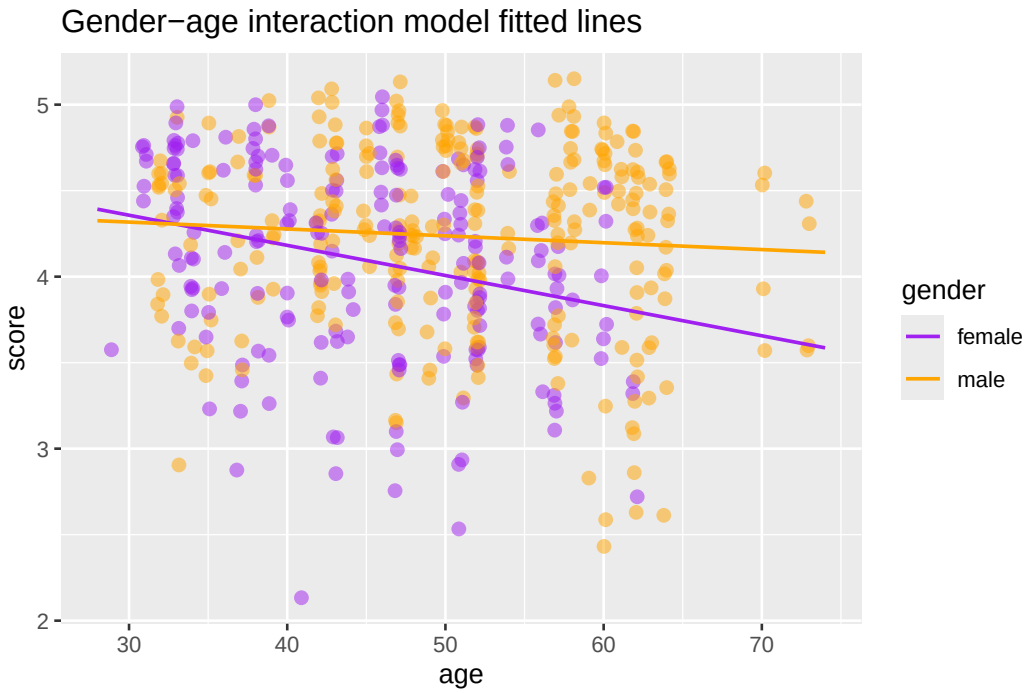


Figure 3: Instructor evaluation scores by age and gender with gender-varying regression lines superimposed.

1.4 Assessing model fit

Now we have to assess the fit of the model by looking at plots of the residuals. We shall do this for the interaction model. First, we need to obtain the fitted values and residuals from the interaction model as follows:

```
int.model_output <-  eval.score %>%
  mutate(score_hat  = int.model$fitted.values,
         residual = int.model$residuals)
```

Let's start by looking at a scatterplot of the residuals against the explanatory variable by gender:

```
ggplot(int.model_output, aes(x = age, y = residual)) +
  geom_point() +
  labs(x = "age", y = "Residual") +
  geom_hline(yintercept = 0, col = "blue", linewidth = 1) +
  facet_wrap(~ gender)
```

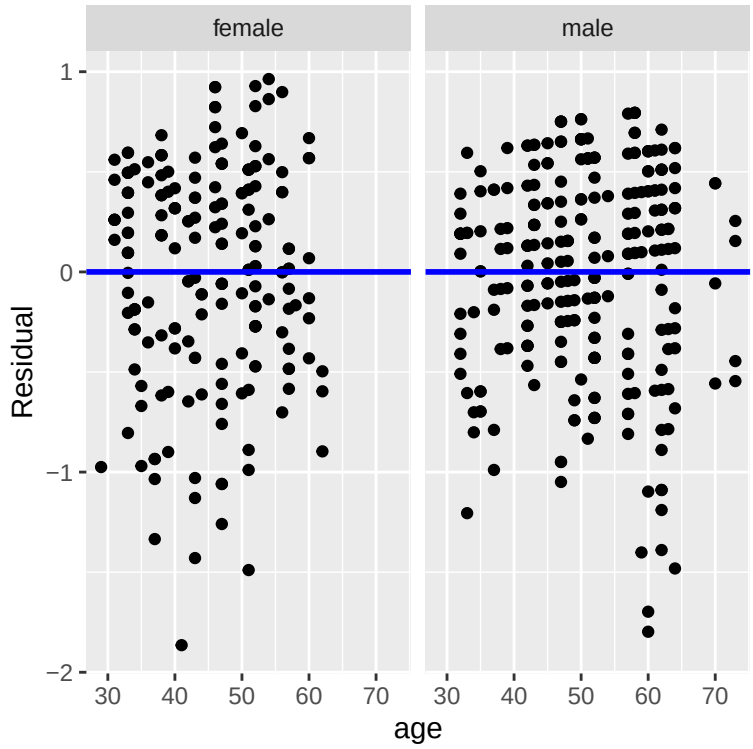
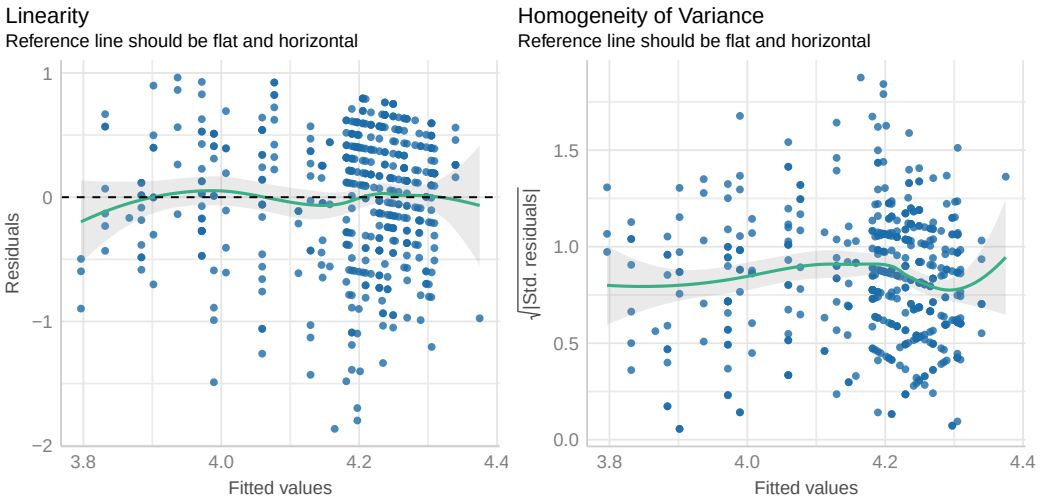


Figure 4: Residuals vs the explanatory variable age by gender.

Now, we can plot the residuals against the fitted values using either the `check_model` function from the `performance` package or the `plot_model()` from `sjPlot` by setting `type="diag"`:

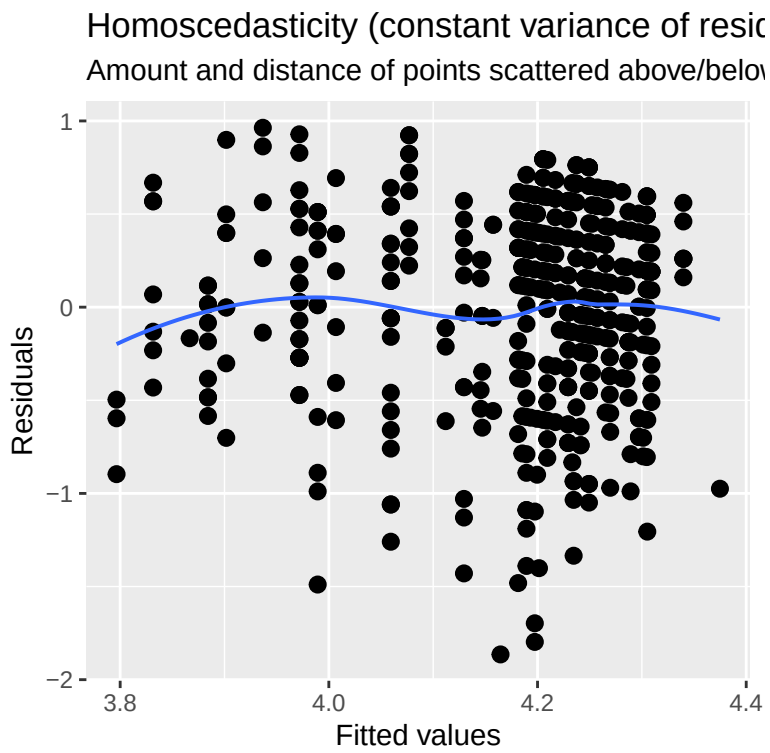
2 Using `check_model`

```
check_model(int.model,check = c("linearity","homogeneity"))
```



3 Using `plot_model`

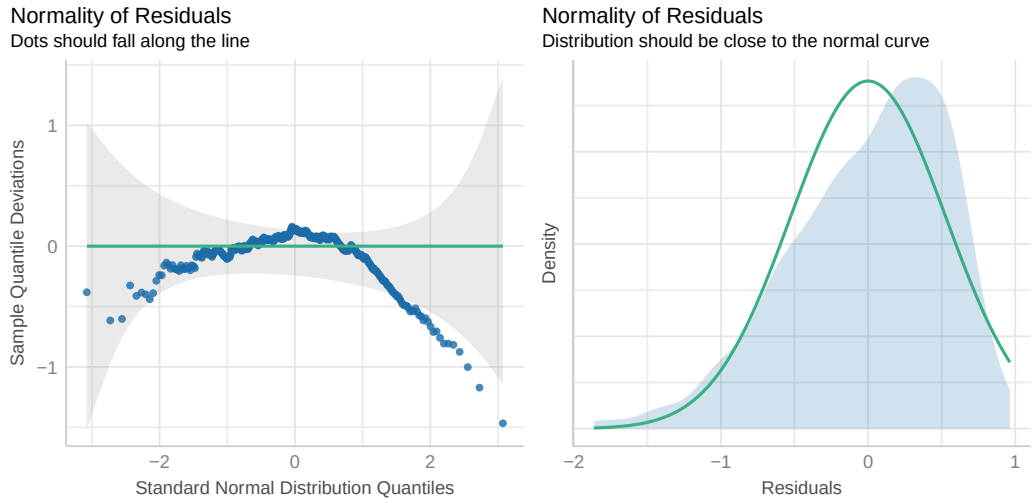
```
int.model_diag <- plot_model(int.model,type = "diag")
int.model_diag[[4]]
```



Finally, let's plot histograms of the residuals and QQ-plots to assess whether they are normally distributed with mean zero:

4 Using `check_model`

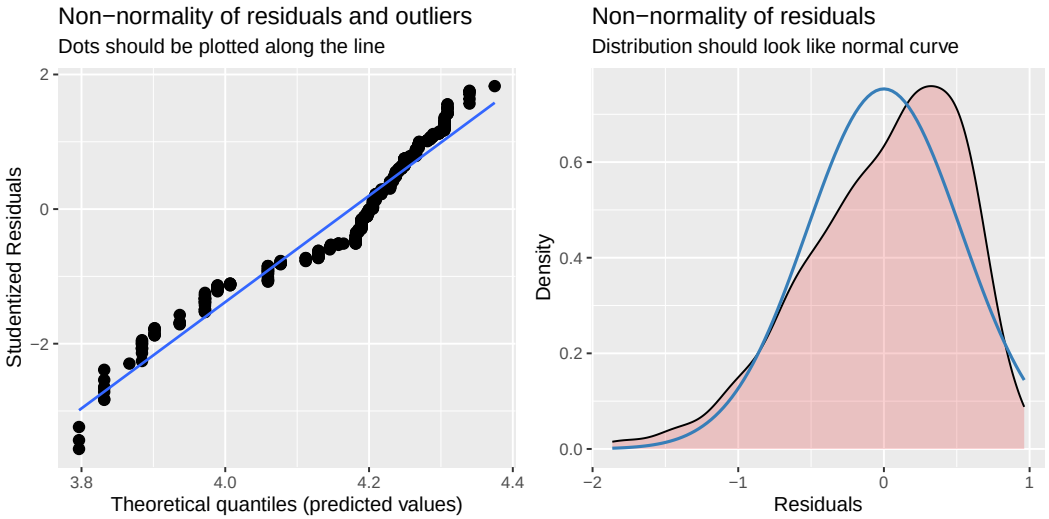
```
check_model(int.model,check = c("qq","normality"))
```



5 Using `plot_model`

```
library(gridExtra) # to arrange the plots side-by-side
int.model_diag <- plot_model(int.model,type = "diag")

gridExtra::grid.arrange(int.model_diag[[2]], # qqplot
                        int.model_diag[[3]], # histogram
                        ncol=2) # plot them side by side
```



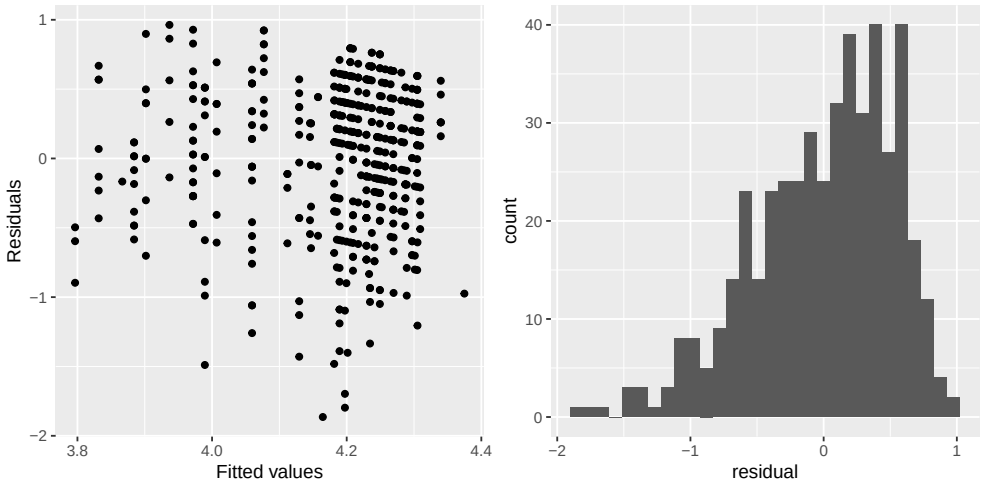
Task 2

Using `ggplot` to produce manually:

1. a scatter plot of the residuals vs. fitted values
2. histogram of the residuals to assess normality

Take a hint
The `int.model_output` data we produced above contains all the information you need. Recall that a scatterplots and histograms can be produced with `geom_point` and `geom_histogram` layers respectively.
Click here to see the solution

```
p1 <- ggplot(int.model_output,
             aes(y=residual,x=score_hat))+
  geom_point()+
  labs(y="Residuals",x="Fitted values")
p2 <- ggplot(int.model_output,
             aes(x=residual))+
  geom_histogram()
gridExtra::grid.arrange(p1, # resid. vs. fitted
                        p2, # histogram
                        ncol=2)
```



6 Sample statistics

This week, and in previous weeks, we have seen many examples of calculating *sample statistics* such as means, percentiles, standard deviations and regression coefficients. These *sample statistics* are used as *point estimates* of *population parameters* which describe the *population* from which the *sample* of data was taken. That last sentence assumes you're familiar with concepts and terminology about sampling (e.g. from the *Statistical Inference* course) so here is a summary of some key terms:

1. **Population**: The population is a set of N observations of interest.
2. **Population parameter**: A population parameter is a numerical summary value about the population. In most settings, this is a value that's unknown and you wish you knew it.

3. **Census**: An exhaustive enumeration/counting of all observations in the population in order to compute the population parameter's numerical value *exactly*.

    - When N is small, a census is feasible. However, when N is large, a census can get very expensive, either in terms of time, energy, or money.

4. **Sampling**: Collecting a sample of size n of observations from the population. Typically the sample size n is much smaller than the population size N, thereby making sampling a much cheaper procedure than a census.

    - It is important to remember that the lowercase n corresponds to the sample size and uppercase N corresponds to the population size, thus $n \leq N$.

5. **Point estimates/sample statistics**: A summary statistic based on the sample of size n that *estimates* the unknown population parameter.

6. **Representative sampling**: A sample is said be a *representative sample* if it "looks like the population". In other words, the sample's characteristics are a good representation of the population's characteristics.

7. **Generalisability**: We say a sample is *generalisable* if any results based on the sample can generalise to the population.

8. **Bias**: In a statistical sense, we say *bias* occurs if certain observations in a population have a higher chance of being sampled than others. We say a sampling procedure is *unbiased* if every observation in a population had an equal chance of being sampled.

9. **Random sampling**: We say a sampling procedure is *random* if we sample randomly from the population in an unbiased fashion.

6.1 Inference using sample statistics

The table below lists a variety of contexts where sample statistics can be used to estimate population parameters. In all 6 cases, the point estimate/sample statistic *estimates* the unknown population parameter. It does so by computing summary statistics based on a sample of size n. We'll cover Scenarios 5 and 6, namely construct CIs for the parameters in simple and multiple linear regression models. We will consider CIs based on theoretical results when standard assumptions hold, although sampling procedures such as bootstrap also exist. We will also consider how to use CIs for variable selection and finish by considering a model selection strategy based on objective measures for model comparisons.

Table 1: Scenarios of sample statistics for inference.

Scenario	Population Parameter	Population Notation	Sample Statistic	Sample Notation
1	Population proportion	p	Sample proportion	$\hat{p}$
2	Population mean	μ	Sample mean	$\bar{x}$
3	Diff.in pop. props	$p_1 - p_2$	Diff. in sample props	$\hat{p}_1 - \hat{p}_2$
4	Diff. in pop. means	$\mu_1 - \mu_2$	Diff. in sample means	$\bar{x}_1 - \bar{x}_2$
5	Pop. intercept	β_0	Sample intercept	$\hat{\beta}_0$ or b_0

Scenario	Population Parameter	Population Notation	Sample Statistic	Sample Notation
6	Pop. slope	β_1	Sample slope	$\hat{\beta}_1$ or b_1

In reality, we don't have access to the population parameter values (if we did, why would we need to estimate them?) we only have a single sample of data from a larger population. We'd like to be able to make some reasonable guesses about population parameters using that single sample to create a range of plausible values for a population parameter. This range of plausible values is known as a **confidence interval**. The confidence intervals we will see this week are calculated using the theoretical results based on the standard assumptions that you will have seen in *Regression Modelling*.

7 Constructing confidence intervals

A **confidence interval** gives a range of plausible values for a population parameter. It depends on a specified *confidence level* with higher confidence levels corresponding to wider confidence intervals and lower confidence levels corresponding to narrower confidence intervals. Common confidence levels include 90%, 95%, and 99%.

Confidence intervals are simple to define and play an important role in the sciences and any field that uses data. You can think of a confidence interval as playing the role of a net when fishing. Instead of just trying to catch a fish with a single spear (estimating an unknown parameter by using a single point estimate/sample statistic), we can use a net to try to provide a range of possible locations for the fish (use a range of possible values based around our sample statistic to make a plausible guess as to the location of the parameter).

7.1 Confidence Intervals for Regression Parameters

To illustrate this, let's have another look at teaching evaluations data `evals` with the SLR model with `age` as the the single explanatory variable and the instructors' evaluation `scores` as the response variable. This data and the fitted model are shown here.

```
slr.model <- lm(score ~ age, data = evals)
```

```
(Intercept)          age
4.461932354 -0.005938225
```

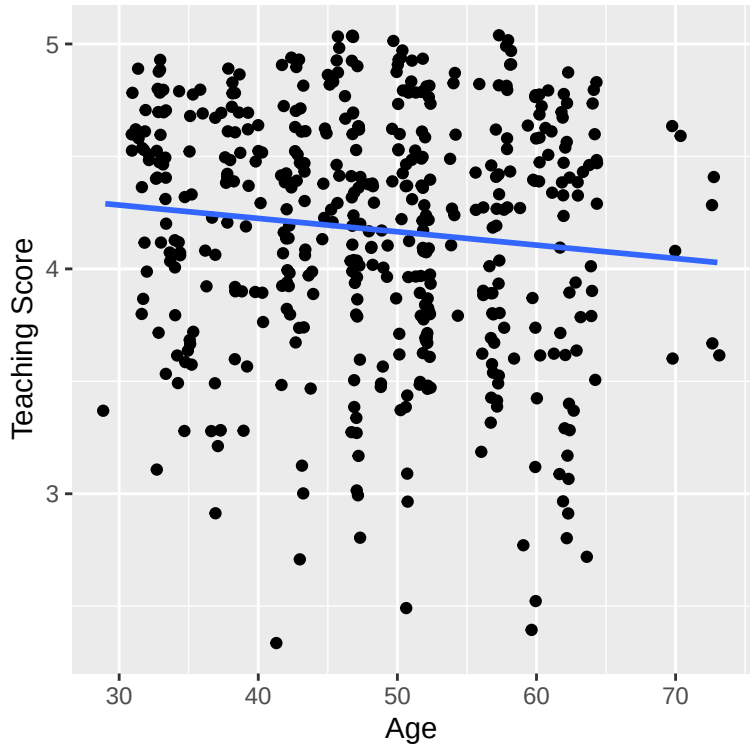


Figure 5: SLR model applied to the teaching evaluation Data.

The point estimate of the slope parameter here is $\hat{\beta} =$ -0.006. But what about the uncertainty of our estimation? Well we can construct confidence interval for this.

Simple linear regression (SLR)

> **Re-cap on SLR**
>
> Recall that for a simple linear regression model we have:
>
> - Var$(\hat{\beta}_0) = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]$ and $se(\hat{\beta}_0) = \sqrt{\hat{\sigma^2}\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]}$
> - Var$(\hat{\beta}_1) = \frac{\sigma^2}{S_{xx}}$ and $se(\hat{\beta}_1) = \frac{\hat{\sigma^2}}{S_{xx}}$
>
> Where $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$ and $\hat{\sigma^2} = \frac{\sum_i y_i - \hat{y}}{n-2}$. Then, we get the following pivotal quantities:
>
> $$\frac{\hat{\beta}_j - \beta_j}{se(\hat{\beta}_j)} \sim t_{n-2}$$
>
> for $j \in \{0, 1\}$.

A $100(1-\alpha)$% confidence interval for the intercept and slope is given by:

- $\hat{\beta}_0 \pm t_{\alpha/2,n-2} \times se(\hat{\beta}_0)$
- $\hat{\beta}_1 \pm t_{\alpha/2,n-2} \times se(\hat{\beta}_1)$

where $t_{\alpha/2,n-2}$ is the critical t-value for a 95% confidence interval (note that sometimes a Gaussian approximation is used such that $\hat{\beta}_j \pm 1.96 \times se(\hat{\beta}_j)$ - this assumes σ^2 is known).

A confidence interval gives a range of plausible values for a population parameter.

We can therefore use the confidence interval for β_1 to state a range of plausible values and, just as usefully, what values are **not** plausible. The most common value to compare the confidence interval of β_1 with is 0 (zero), since $\beta_1 = 0$ says there is *no* (linear) relationship between the response variable (y) and the explanatory variable (x). Therefore, if 0 lies within the confidence interval for β_1 then there is insufficient evidence of a linear relationship between y and x. However, if 0 **does not** lie within the confidence interval, then we conclude that β_1 is significantly different from zero and therefore that there is evidence of a linear relationship between y and x.

Let's use the confidence interval based on theoretical results for the slope parameter in the SLR model applied to the teacher evaluation scores with `age` as the the single explanatory variable and the instructors' evaluation `score`s as the outcome variable.

The `tab_model` function allow us to print the $(1-\alpha)$% confidence intervals for our parameters using the argument `show.ci = (1-alpha)`. Additionally, we can print the estimator standard error by setting `show.se=T`):

```
tab_model(slr.model,show.se = T,show.ci = 0.95)
```

	score			
Predictors	Estimates	std. Error	CI	p
(Intercept)	4.46	0.13	4.21 – 4.71	**<0.001**
age	-0.01	0.00	-0.01 – -0.00	**0.021**
Observations	463			
R^2 / R^2 adjusted	0.011 / 0.009			

This will give you a nice looking html table. However, if you prefer your output to be a `data.frame` object, you can use the `tidy` function from the `broom` package instead (you can print the CI using `conf.int=T`):

```
broom::tidy(slr.model,conf.int=T,conf.level = 0.95)
```

```
# A tibble: 2 x 7
  term        estimate std.error statistic   p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)  4.46      0.127       35.2  1.05e-132   4.21    4.71
2 age         -0.00594   0.00257     -2.31 2.13e-  2  -0.0110 -0.000890
```

Then we can plot our fitted SLR using the `plot_model` function:

```
plot_model(model = slr.model,
           terms = c("age"),
           type  = "pred",
           title = "Fitted Linear regression model",
           show.data = T,
           jitter = T)
```

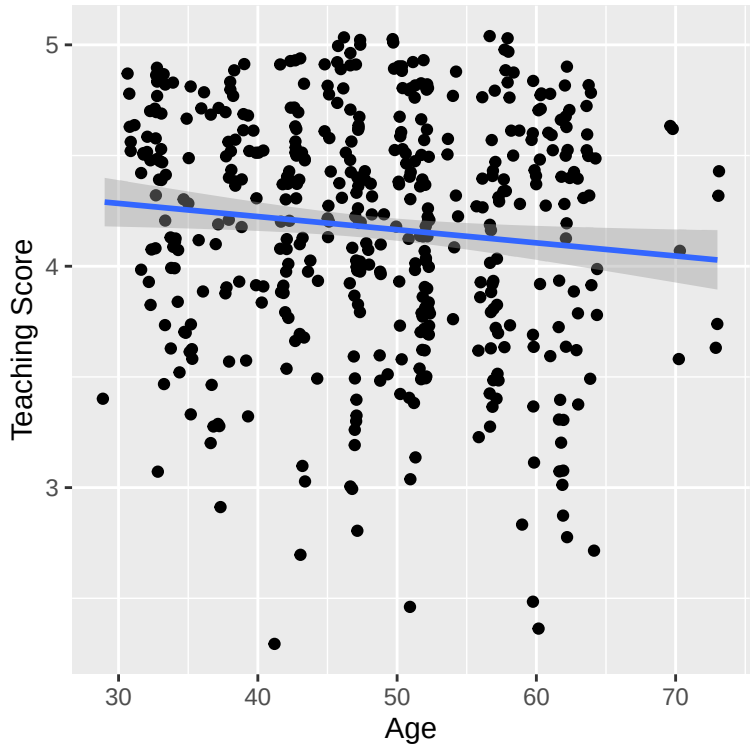
Figure 6: SLR model applied to the teaching evaluation Data.

Task 3

Use `ggplot` to reproduce the plot above. You may achieve this by adding a `geom_smooth()` layer.
Hint
Chose `method ="lm"` as argument of the `geom_smooth()` layer. You may also chance the confidence level via the `level` argument.
Click here to see the solution

```
ggplot(evals, aes(x = age, y = score)) +
  geom_jitter() +
  labs(x = "Age", y = "Teaching Score") +
 geom_smooth(method = "lm",level=.95)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Multiple Regression confidence intervals

Let's continue with the teaching evaluations data by looking into the multiple regression models we have fitted with one numerical and one categorical explanatory variable . In these models:

- y: response variable of instructor evaluation `score`
- explanatory variables
  - x_1: numerical explanatory variable of `age`
  - x_2: categorical explanatory variable of `gender`

First, recall that we had two competing potential models to explain professors' teaching evaluation scores:

1. Model 1 (Equation 1): Parallel lines model (no interaction term) - both male and female professors have the same slope describing the associated effect of age on teaching score
2. Model 2 (Equation 1): Interaction model - allowing for male and female professors to have different slopes describing the associated effect of age on teaching score

Let's recall the output of these regression models:

```
tab_model(par.model,
          int.model,
          collapse.ci = T,
          show.stat = T,
          show.se = T,
          dv.labels = c("parallel lines model","interaction model") )
```

	parallel lines model	interaction model

Predictors	Estimates	std. Error	Statistic	p	Estimates	std. Error	Statistic	p
(Intercept)	4.48 (4.24 – 4.73)	0.13	35.79	**<0.001**	4.88 (4.48 – 5.29)	0.21	23.80	**<0.001**
age	-0.01 (-0.01 – -0.00)	0.00	-3.28	**0.001**	-0.02 (-0.03 – -0.01)	0.00	-3.92	**<0.001**
gender [male]	0.19 (0.09 – 0.29)	0.05	3.63	**<0.001**	-0.45 (-0.97 – 0.08)	0.27	-1.68	0.094
age × gender [male]					0.01 (0.00 – 0.02)	0.01	2.45	**0.015**
Observations	463				463			
R^2 / R^2 adjusted	0.039 / 0.035				0.051 / 0.045			

Notice that, together with the estimated parameter values, the tables we produced now include other information about each estimated parameter in the model, namely:

- **std_error**: the standard error of each parameter estimate (set `show.se = T`);
- **statistic**: the test statistic value used to test the null hypothesis that the population parameter is zero (set `show.stat = T`);
- **p_value**: the p value associated with the test statistic under the null hypothesis; and
- **lower_ci** and **upper_ci**: the lower and upper bounds of the 95% confidence interval for the population parameter (the options `collapse.ci =T` can be used to parse the CI next to the parameter estimates)

These values are calculated using the theoretical results based on the standard assumptions that you will have seen in *Regression Modelling*.

8 Variable selection using confidence intervals

When there is more than one explanatory variable in a model, the parameter associated with each explanatory variable is interpreted as the change in the mean response based on a 1-unit change in the corresponding explanatory variable **keeping all other variables held constant**. Therefore, care must be taken when interpreting the confidence intervals of each parameter by acknowledging that each are plausible values **conditional on all the other explanatory variables in the model**.

Because of the interdependence between the parameter estimates and the variables included in the model, choosing which variables to include in the model is a rather complex task. We will introduce some of the ideas in the simple case where we have 2 potential explanatory variables (x_1 and x_2) and use confidence intervals to decide which variables will be useful in predicting the response variable.

One approach is to consider a hierarchy of models:

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i}$$

$$y_i = \alpha + \beta_1 x_{1i} \qquad\qquad y_i = \alpha + \beta_2 x_{2i}$$

$$y_i = \alpha$$

Within this structure we might take a top-down approach:

1. Fit the most general model, i.e. $y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i}$ since we believe this is likely to provide a good description of the data
2. Construct confidence intervals for β_1 and β_2
   (a) If both intervals exclude 0 then retain the model with both x_1 and x_2.
   (b) If the interval for β_1 contains 0 but that for β_2 does not, fit the model with x_2 alone.
   (c) If the interval for β_2 contains 0 but that for β_1 does not, fit the model with x_1 alone.
   (d) If both intervals include 0 it may still be that a model with one variable is useful. In this case the two models with the single variables should be fitted and intervals for β_1 and β_2 constructed and compared with 0.

If we have only a few explanatory variables, then an extension of the strategy outlined above would be effective, i.e. start with the full model and simplify by removing terms until no further terms can be removed. When the number of explanatory variables is large the problem becomes more difficult. We will consider this more challenging situation in the next section.

Recall that as well as `age` and `gender`, there is also a potential explanatory variable `bty_avg` in the `evals` data, i.e. the numerical variable of the average beauty score from a panel of six students' scores between 1 and 10. We can fit the multiple regression model with the two continuous explanatory variables `age` and `bty_avg` as follows:

```
mlr.model <- lm(score ~ age + bty_avg, data = evals)
tab_model(mlr.model)
```

Table 9: Estimates from the MLR model with `age` and `bty_avg`.

		score	
Predictors	Estimates	CI	p
(Intercept)	4.05	3.72 – 4.39	**<0.001**
age	-0.00	-0.01 – 0.00	0.251
bty avg	0.06	0.03 – 0.09	**<0.001**
Observations	463		
R^2 / R^2 adjusted	0.038 / 0.034		

Based on this output, we can say that after accounting for the mean beauty score or holding it constant in our model, the variable age is not a significant linear predictor for the teacher's score and can therefore be removed from it. But what about if we have multiple candidate models or many potential variables?

9 Model comparisons using objective criteria

As was noted in the last section, when the number of potential explanatory variables is large the problem of selecting which variables to include in the final model becomes more difficult. The selection of a final regression model always involves a compromise:

- Predictive accuracy (improved by including more predictor/explanatory variables)
- Interpretability (achieved by having less predictor/explanatory variables)

There are many objective criteria for comparing different models applied to the same data set. All of them trade off the two objectives above, i.e. fit to the data against complexity. Common examples include:

1. The R^2_{adj} values, i.e. the proportion of the total variation of the response variable explained by the models.

$$R^2_{adj} = 1 - \frac{RSS/(n-p-1)}{SST/(n-1)} = 100 \times \left[1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2/(n-p-1)}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2/(n-1)}\right]$$

   - where
     - n is the sample size
     - p is the number of parameters in the model
     - RSS is the residual sum of squares from the fitted model
     - SST is the total sum of squares around the mean response.
   - R^2_{adj} values are used for nested models, i.e. where one model is a particular case of the other

2. Akaike's Information Criteria (AIC)

$$AIC = -2 \cdot \text{log-likeihood} + 2p = n \ln\left(\frac{RSS}{n}\right) + 2p$$

   - A value based on the maximum likelihood function of the parameters in the fitted model penalised by the number of parameters in the model
   - It be used to compare any models fitted to the same response variable
   - The smaller the AIC the 'better' the model, i.e. no distributional results are employed to assess differences
   - See the `stepAIC` function from the `MASS` library that was mention in Week 6.

3. Bayesian Information Criteria

$$BIC = -2 \cdot \text{log-likeihood} + \ln(n)p$$

A popular data analysis strategy that can be adopted is to calculate R^2_{adj}, AIC and BIC and compare the models which **minimise** the AIC and BIC with the model that **maximises** the R^2_{adj}.

To illustrate this, let's return to the `evals` data and the MLR on the teaching evaluation score `score` with the two continuous explanatory variables `age` and `bty_avg` and compare this with two SLR models using `bty_avg` or `age` as predictors only.

```
mlr.model <- lm(score ~ age + bty_avg, data = evals)
slr.model_bty <- lm(score ~ bty_avg, data = evals)
slr.model_age <- lm(score ~ age , data = evals)

tab_model(mlr.model,slr.model_bty,slr.model_age,
          show.aic = T,
          collapse.ci = T)
```

Table 10: Model comparisson for a multiple linear regression model agianst two nested simple linear regression models.

	score		score		score	
Predictors	Estimates	p	Estimates	p	Estimates	p
(Intercept)	4.05 (3.72 – 4.39)	**<0.001**	3.88 (3.73 – 4.03)	**<0.001**	4.46 (4.21 – 4.71)	**<0.001**
age	-0.00 (-0.01 – 0.00)	0.251			-0.01 (-0.01 – -0.00)	**0.021**
bty avg	0.06 (0.03 – 0.09)	**<0.001**	0.07 (0.03 – 0.10)	**<0.001**		
Observations	463		463		463	
R^2 / R^2 adjusted	0.038 / 0.034		0.035 / 0.033		0.011 / 0.009	
AIC	739.119		738.445		749.616	

According to the AIC values shown in Table 10 , the simple linear model with beauty average as only predictor is preferred.

Note that we can also use the `glance` function in the `broom` package (not to be confused with the `glimpse` function from the `dplyr` package) to compute several information criteria metrics:

```
broom::glance(mlr.model)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
1    0.0378        0.0336 0.535      9.03 0.000142     2  -366.  739.  756.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
broom::glance(slr.model_bty)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
1    0.0350        0.0329 0.535      16.7 0.0000508     1  -366.  738.  751.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Both BIC and AIC suggest that the simple linear model with beauty score as a predictor provides the best fit t the data.

> **Note**
>
> The `tab_model` function returns a corrected AIC for transformed response-values, and thus, results might differ from those obtained through `glance`.

10 A final word on model selection

A great deal of care should be taken in selecting predictor/explanatory variables for a model because the values of the regression coefficients depend upon the variables included in the model. Therefore, the predictors included and the order in which they are entered into the model can have great impact. In an ideal world, predictors should be selected based on past research and new predictors should be added to existing models based on the theoretical importance of the variables. One thing not to do is select hundreds of random predictors, bung them all into a regression analysis and hope for the best.

But in practice there are automatic strategies, such as **Stepwise** and **Best Subsets** regression, based on systematically searching through the entire list of variables not in the current model to make decisions on whether each should be included. These strategies need to be handled with care, and a proper discussion of them is beyond this course. Our best strategy is a mixture of judgement on what variables should be included as potential explanatory variables, together with an interval estimation and hypothesis testing strategy for assessing these. The judgement should be made in the light of advice from the problem context.

Golden rule for modelling

> The key to modelling data is to only use the objective measures as a rough guide. In the end the choice of model will involve your own judgement. You have to be able to defend why you chose a particular model.