

# **Regression: modelling**

ENVX2001 Applied Statistical Methods

**Liana Pozza**

Feb 2026

**Welcome to regression modelling!**

## About me

- Research topics: spatial modelling and mapping, precision agriculture, winter grains
- Timeline at USYD
  - BSc (Hons) in Agricultural Science
  - PhD in Digital Agriculture
  - Postdoc in Spatial Modelling
  - Associate Lecturer in Agricultural Data Science



Figure 1: Faba beans at Trangie

## **Learning Outcomes**

- LO1. demonstrate proficiency in designing sample schemes and analysing data from them using R
- LO2. describe and identify the basic features of an experimental design; replicate, treatment structure and blocking structure
- LO3. demonstrate proficiency in the use or the statistical programming language R to** an ANOVA and **fit regression models to experimental data**
- LO4. demonstrate proficiency in the use or the statistical programming language R to use multivariate methods to find patterns in data
- LO5. interpret the output and understand conceptually how its derived of a regression**, ANOVA and multivariate analysis that have been calculated by R
- LO6. write statistical and modelling results as part of a scientific report
- LO7. appraise the validity of statistical analyses used publications.

## Refresher from ENVX1002

- Regression modelling is for one *continuous numerical* response ( $y$ ) and one or more *numerical* predictors ( $x_1, x_2, x_n$ )
- Can be for linear or nonlinear relationships – focus on linear in ENVX2001
- To help us:
  - Understand the relationship between variables
  - Predict new values of  $y$  based on  $x$
  - Test hypotheses about the relationship between variables
- Fit a ‘line of best fit’ that minimises the sum of the squared residuals (least-squares)

## Workflow

### 1. Model development

- Explore: visualise, summarise
- Model: fit, check assumptions, interpret – (transform, repeat).
- Transform predictors

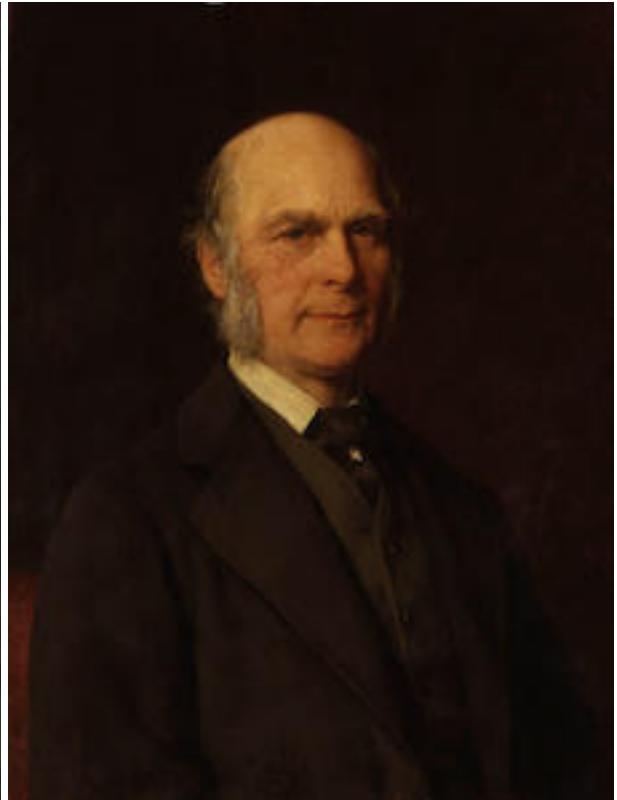
### 2. Variable selection

- VIF: remove predictors with high variance inflation factor
- Model selection: stepwise selection, AIC, principle of parsimony, assumption checks

### 3. Predictive modelling

- Predict: Use the model to predict new data
- Validate: Evaluate the model's performance

## Brief history



Adrien-Marie Legendre, Carl Friedrich Gauss, Francis Galton

**i** Note

Many other people contributed to the development of regression analysis, but these three are the most well-known.

## Brief history

- **Method of least squares** first theorised by Adrien-Marie Legendre in 1805
- **Technique of least squares** first used by Carl Friedrich Gauss in 1809 (to fit a parabola to the orbit of the asteroid Ceres)
- **Model fitting** first published by Francis Galton in 1886 (predicting the height of a child from the height of the parents)

# Simple linear regression

| An example with Galton's data: parent and child heights.

## Example: child vs parent height

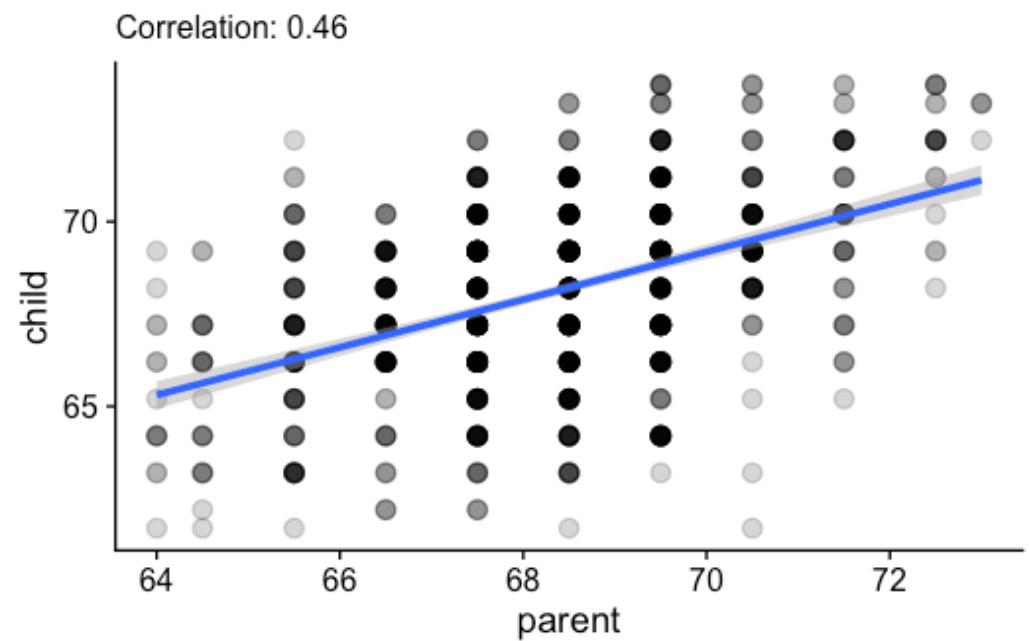
Galton, F. (1886). Regression Towards Mediocrity in Hereditary Stature Journal of the Anthropological Institute, 15, 246-263

```
library(HistData)
data(Galton)
str(Galton)
```

```
'data.frame': 928 obs. of 2 variables:
 $ parent: num 70.5 68.5 65.5 64.5 64 67.5 67.5
 67.5 66.5 66.5 ...
 $ child : num 61.7 61.7 61.7 61.7 61.7 62.2
 62.2 62.2 62.2 62.2 ...
```

```
ggplot(Galton, aes(x = parent, y = child)) +
  geom_point(alpha = .2, size = 3) +
  geom_smooth(method = "lm") +
  labs(subtitle = paste("Correlation:",
  round(cor(Galton$parent, Galton$child), 2)))
```

- 928 children of 205 pairs of parents
- Average height of both parents and their child's height measured in inches
- Size classes were binned (hence data looks discrete)



## Defining a linear relationship

- Pearson correlation coefficient ( $r$ ) measures the linear correlation between two variables (ranges from  $-1$  to  $1$ )
- Useful for distinguishing *strength* (weak/moderate/strong) and *direction* (positive/negative) of the association
- Does not distinguish different *patterns* – i.e. is the relationship actually linear?

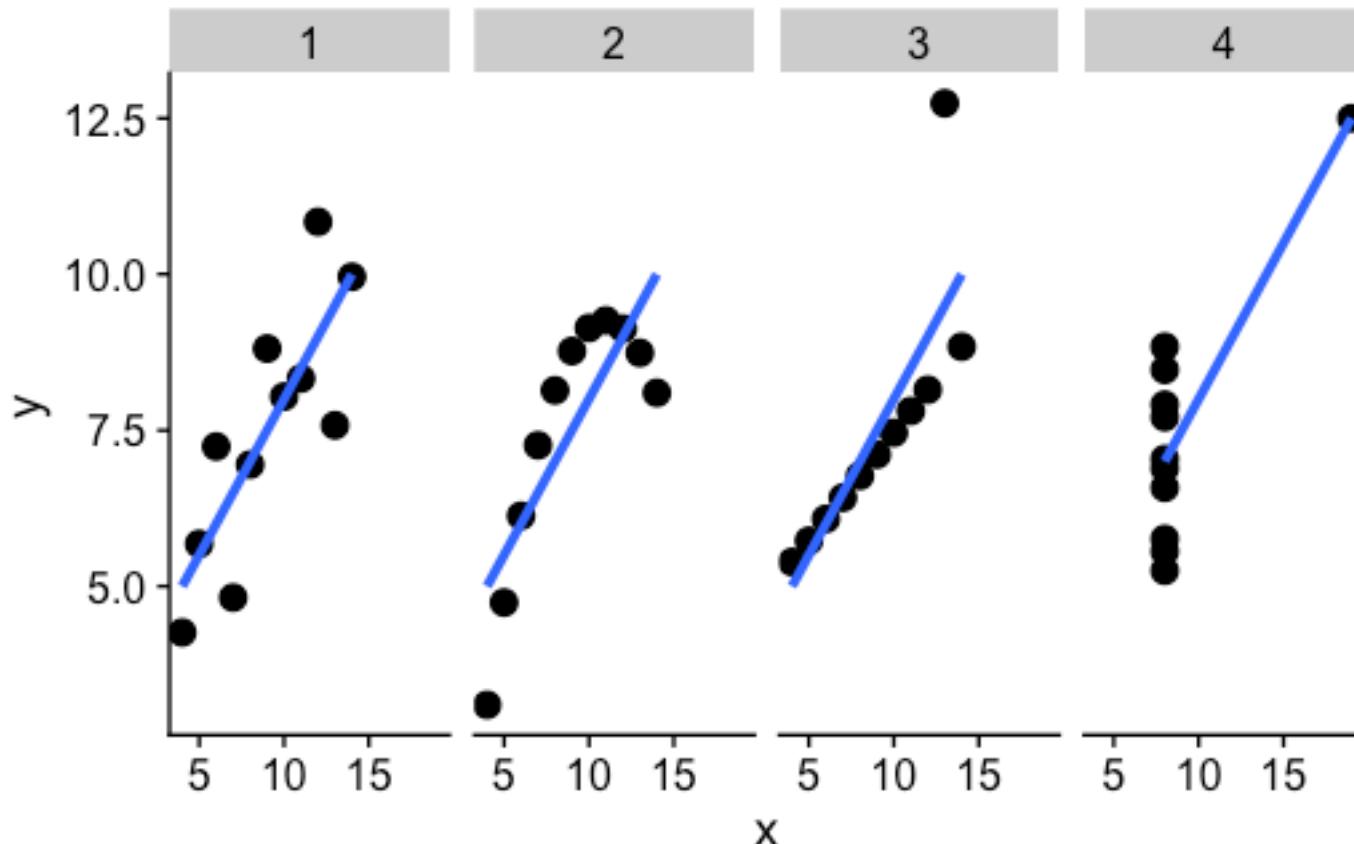
$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

```
cor(Galton$parent, Galton$child) |> round(2)
```

```
[1] 0.46
```

## Anscombe's quartet

```
library(tidyverse)
anscombe %>%
  pivot_longer(everything(), cols_vary = "slowest",
  names_to = c(".value", "set"), names_pattern = "(.)(.)") %>%
  ggplot(aes(x = x, y = y)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~set, ncol = 4)
```



All of these data have a correlation coefficient of about 0.8 – always visualise your data.

## Simple linear regression model

We want to predict a response  $Y$  based on a predictor  $x$  for  $i$  number of observations:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where

$$\epsilon_i \sim N(0, \sigma^2)$$

- $Y_i$ , the *response*, is an observed value of the dependent variable.
- $\beta_0$ , the *constant*, is the population intercept and is **fixed**.
- $\beta_1$  is the population *slope* parameter, and like  $\beta_0$ , is also **fixed**.
- $\epsilon_i$  is the error associated with predictions of  $y_i$ , and unlike  $\beta_0$  or  $\beta_1$ , it is *not fixed*.

Because  $\epsilon_i$  is the only part of the equation that is not fixed, we associate it with the **residual error** (*observed – predicted*). It would also cover other aspects of error (e.g. sampling error, parallax error) but these are hard to discern.

## Fitting the model

- $\hat{y}_i$  is the predicted value of  $y_i$ :

$$\hat{y}_i = \beta_0 + \beta_1 x_i$$

- The *residual* is the difference between the observed value of the response and the predicted value:

$$\hat{\epsilon}_i = y_i - \hat{y}_i$$

- Therefore:

$$\hat{\epsilon}_i = y_i - (\beta_0 + \beta_1 x_i)$$

- We use the **method of least squares** and minimise the sum of the squared residuals (SS):

$$\sum_{i=1}^n \hat{\epsilon}_i^2 = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$

Finding the minimum SS requires solving the following problem:

$$\operatorname{argmin}_{\beta_0, \beta_1} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$

We can find  $\beta_0$  and  $\beta_1$  **analytically**. We first find  $\beta_1$ :

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{Cov(x, y)}{Var(x)} = \frac{SS_{xy}}{SS_{xx}}$$

And then substitute  $\beta_1$  into the equation for  $\beta_0$ :

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

## Numerical fitting

Computers use “random guesses” to find set of parameters that minimises objective function (SS) – more computationally efficient and applies beyond linear regression.

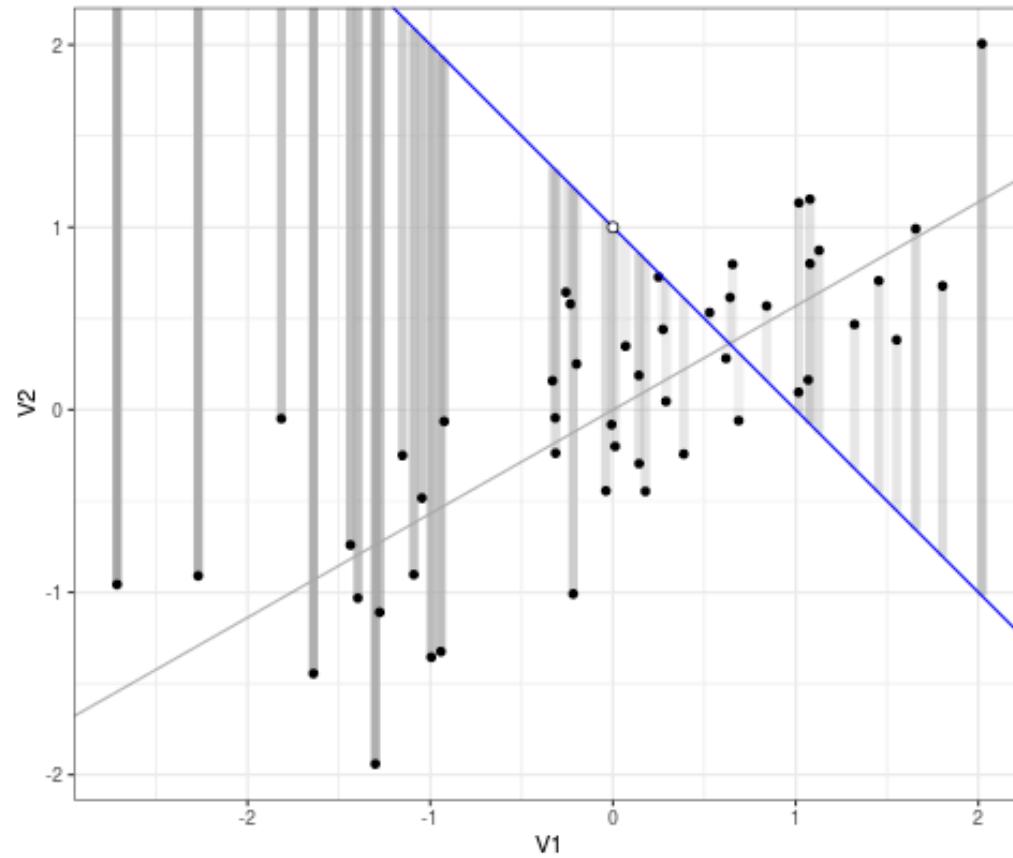


Figure 1: source

## Fitting a model in R is easy with `lm()`

```
fit ← lm(child ~ parent, data = Galton)
```

That's it – the model has been fitted.

**But** there is a process similar to HATPC (hypothesis, assumptions, test, p-value, conclusions).

## Define the hypothesis

$$H_0 : \beta_1 = 0$$

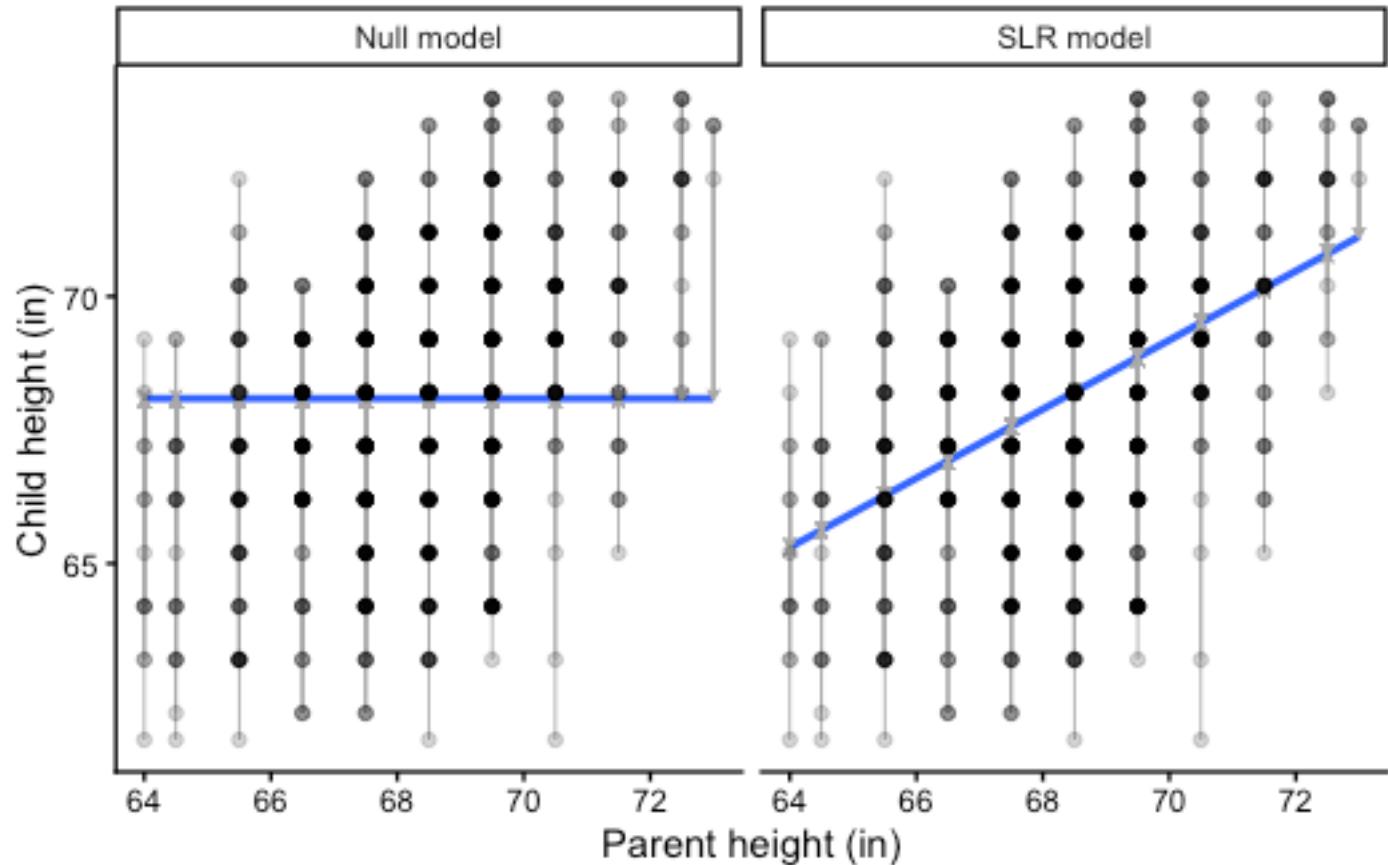
$$H_1 : \beta_1 \neq 0$$

The null model is a line with no slope (i.e. flat or horizontal) at the mean of the child height ( $\bar{y} = 68$  inches).

```
library(dplyr)
null_model ← Galton %>%
  lm(child ~ 1, data = .) %>%
  broom::augment(Galton)
lin_model ← Galton %>%
  lm(child ~ parent, data = .) %>%
  broom::augment(Galton)
models ← bind_rows(null_model, lin_model) %>%
  mutate(model = rep(c("Null model", "SLR model"), each = nrow(Galton)))

ggplot(data = models, aes(x = parent, y = child)) +
  geom_smooth(
    data = filter(models, model == "Null model"),
```

```
method = "lm", se = FALSE, formula = y ~ 1, size = 1
) +
geom_smooth(
  data = filter(models, model == "SLR model"),
  method = "lm", se = FALSE, formula = y ~ x, size = 1
) +
geom_segment(
  aes(xend = parent, yend = .fitted),
  arrow = arrow(length = unit(0.1, "cm")),
  size = 0.3, color = "darkgray"
) +
geom_point(alpha = .2) +
facet_wrap(~model) +
xlab("Parent height (in)") +
ylab("Child height (in)") +
theme_classic()
```



## Assumptions

The data **must** meet certain criteria, which we often call *assumptions*. They can be remembered using **LINE**:

- **L**inearity. The relationship between  $y$  and  $x$  is linear.
- **I**ndependence. The errors  $\epsilon$  are independent.
- **N**ormal. The errors  $\epsilon$  are normally distributed.
- **E**qual Variance. At each value of  $x$ , the variance of  $y$  is the same i.e. homoskedasticity, or constant variance.



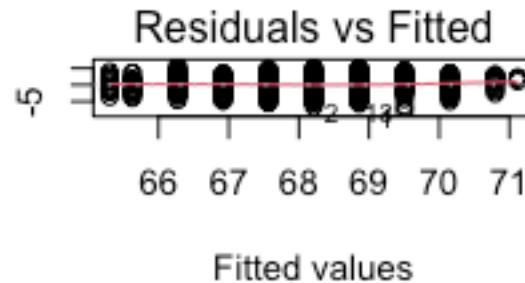
Tip

All but the independence assumption can be assessed using diagnostic plots.

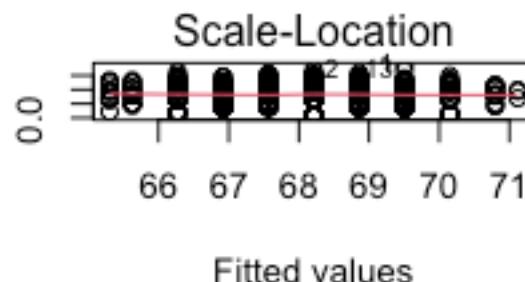
## Assumptions with base R `plot()`

```
par(mfrow= c(2, 2)) # plots combined into 2x2 grid
plot(fit)
```

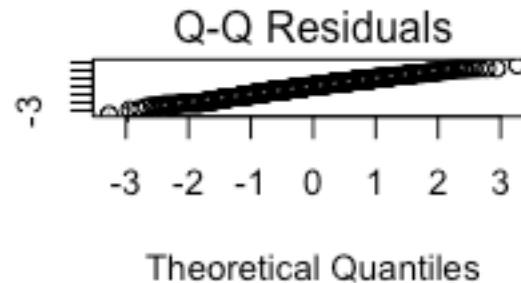
Residuals



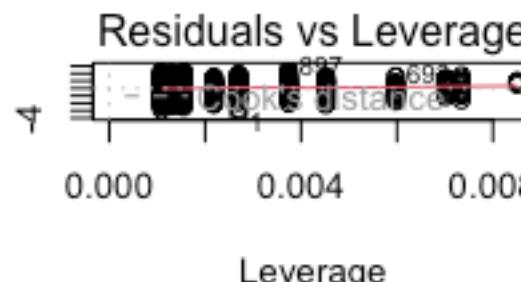
$\sqrt{|\text{Standardized residuals}|}$



Standardized residuals

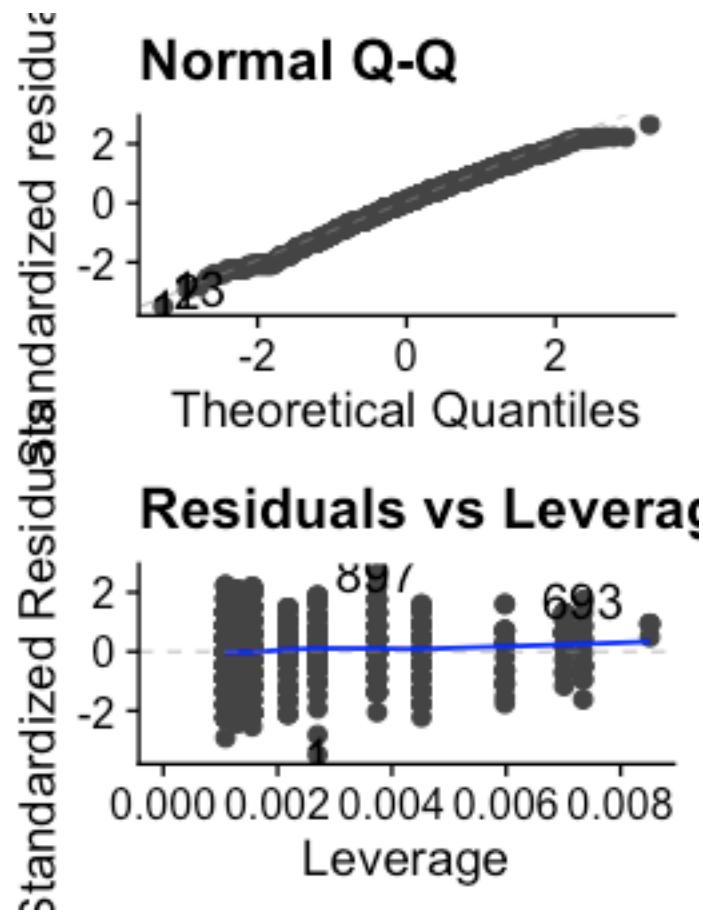
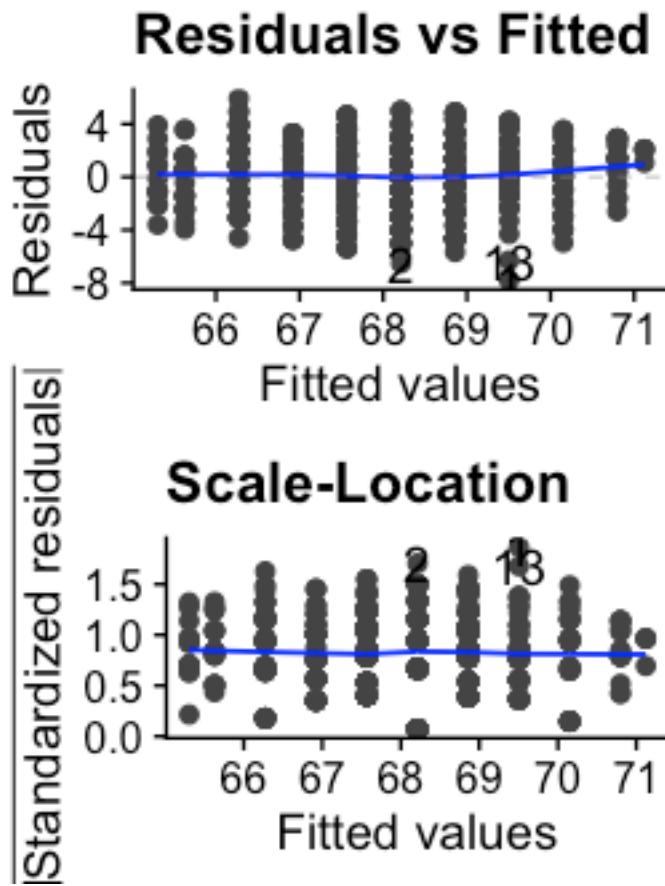


Standardized residuals



## Assumptions with `ggfortify` package and `autoplot()`

```
library(ggfortify)  
autoplot(fit)
```



## Assumptions using `performance`

(Also provides a guide on what to check for in the assumption plot)

```
library(performance)
performance::check_model(fit) # check all assumptions
performance::check_model(fit, check = c("linearity", "qq", "homogeneity", "outliers")) # check
specific assumptions
```

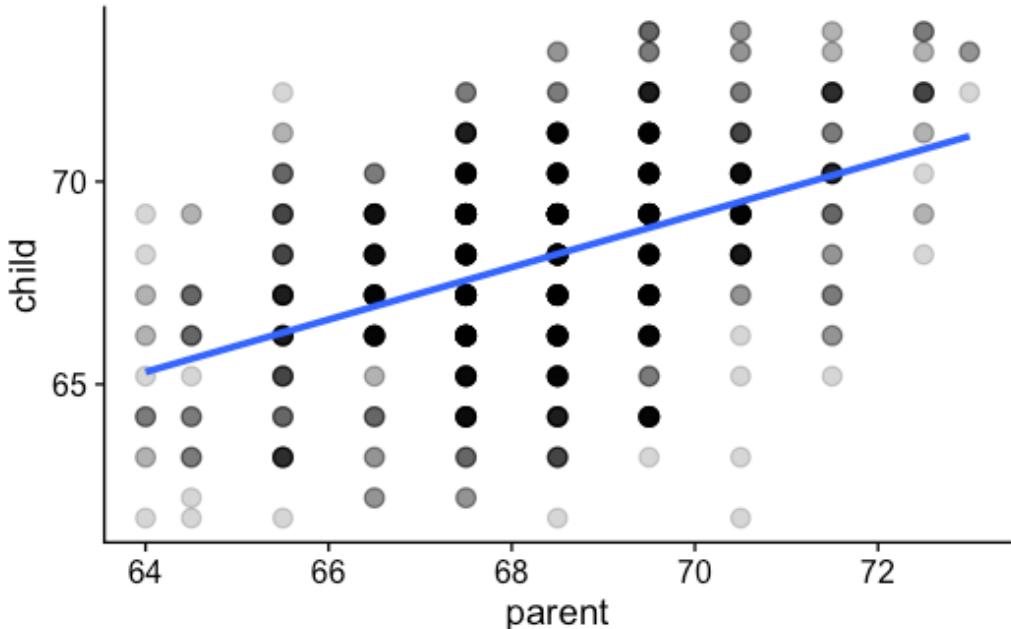
## Assumption: Linearity

Prior knowledge and visual inspection comes into play. Does the relationship look approximately linear?

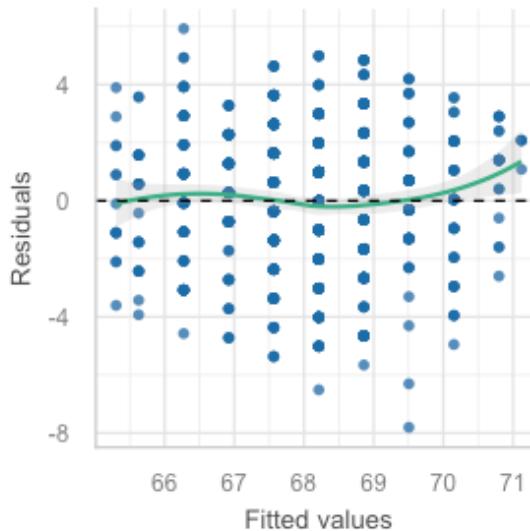
```
ggplot(Galton, aes(x = parent, y = child)) +  
  geom_point(alpha = .2, size = 3) +  
  geom_smooth(method = "lm", se = FALSE)
```

The linearity assumption can be checked again by looking at a plot of the residuals against  $x$  (i.e. `parent` height).

```
performance::check_model(fit, check =  
  "linearity")
```



**Linearity**  
Reference line should be flat and horizontal



- Where the green reference line is  $> 0$ , the model *underestimates*, and where it is  $< 0$ , it *overestimates*.
- If the linearity assumption is **violated**, we should not be fitting a linear model – transform or use a nonlinear model.

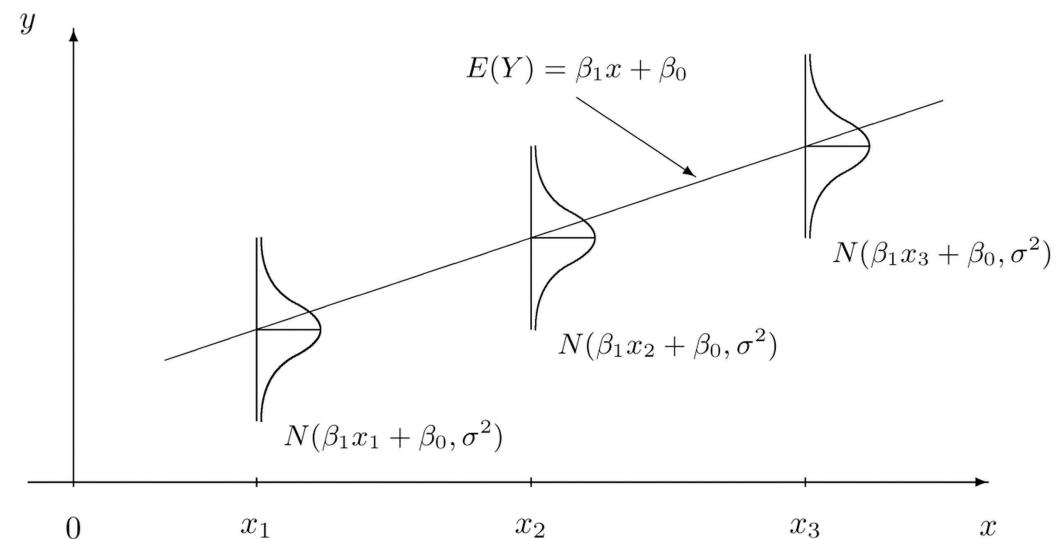
## Assumption: Independence

This assumption is addressed during experimental design, but issues like correlation between errors and patterns occurring due to time are possible if:

- Observations of the same subject are related i.e. **multicollinearity**
- Time-series data, if the same subjects are sampled i.e. **autocorrelation**

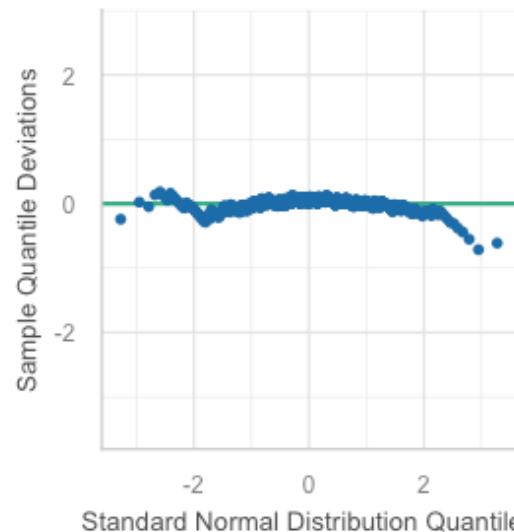
## Assumption: Normality

For a given value of  $x$ , the residuals should be normally distributed. In a scatterplot of  $x$  and  $y$ , the points would appear evenly distributed (linear and no fanning).

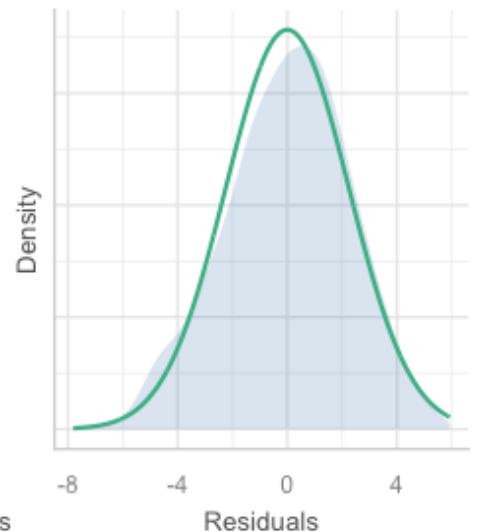


```
performance::check_model(fit, check =  
c("normality", "qq"))
```

Normality of Residuals  
Dots should fall along the line



Normality of Residuals  
Distribution should be close to the normal



- How to interpret a QQ plot

- QQ plot interpretation

## Assessing normality using residuals

- **Light-tailed**: small variance in residuals, resulting in a narrow distribution
- **Heavy-tailed**: many extreme positive and negative residuals, resulting in a wide distribution
- **Left-skewed** ( $n$  shape): more data falls to the left of the mean
- **Right-skewed** ( $u$  shape): more data falls to the right of the mean

Heavy-tailed, left-skewed.

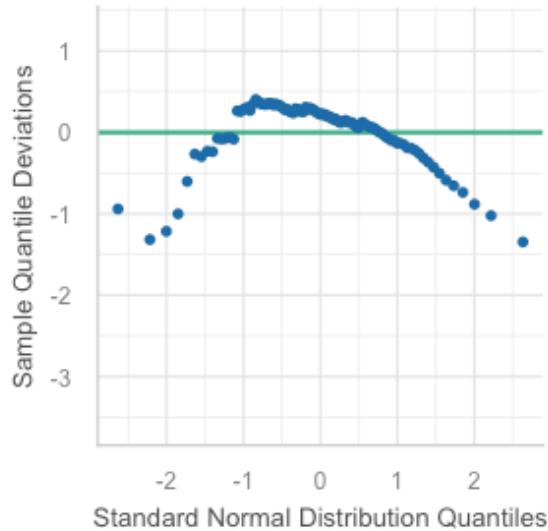
```
set.seed(1028)
x ← rnorm(100)
y ← 2 + 5 * x + rchisq(100, df = 3) * -1
df ← data.frame(x, y)
performance::check_model(lm(y ~ x, data = df),
  check = c(c("qq")))
```

Light-tailed, right-skewed.

```
set.seed(1028)
x ← rnorm(100)
y ← 2 + 5 * x + rnbinom(100, 10, .5)
df ← data.frame(x, y)
performance::check_model(lm(y ~ x, data = df),
  check = c(c("qq")))
```

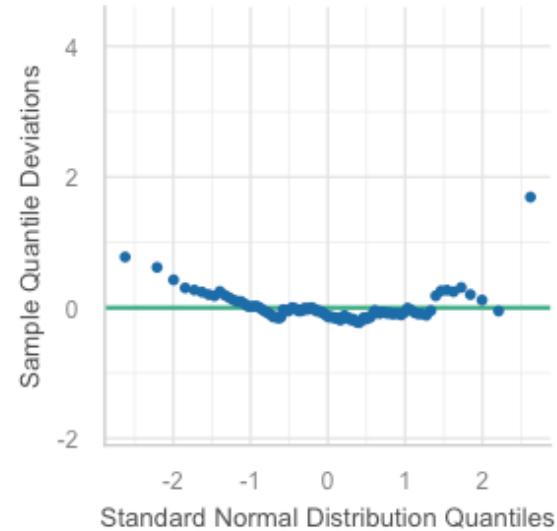
### Normality of Residuals

Dots should fall along the line



### Normality of Residuals

Dots should fall along the line



## Asumption: Equal variances

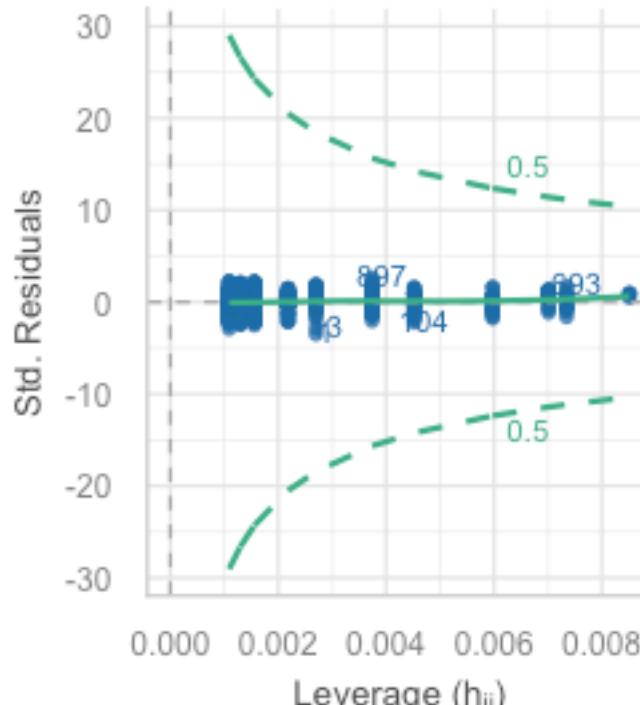
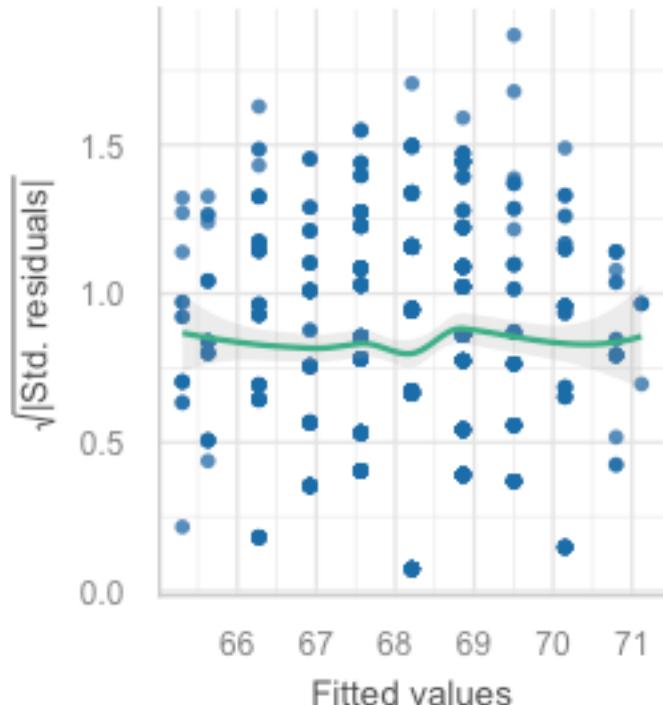
```
performance::check_model(fit, check = c("homogeneity", "outliers"))
```

## Homogeneity of Variance

Reference line should be flat and horizontal

## Influential Observations

Points should be inside the contour lines



Outliers are not a strict assumption, but they will affect the model fit.

## What is a standardised residual?

- The standardised residual is the residual divided by the standard error of the residual (normalised).

$$\text{Standardised residual} = \frac{\text{Residual}}{\text{Standard error of the residual}}$$

- The *mean* of the residuals is 0 in linear regression
- A standardised residual of 2 or above suggests the point is an outlier (far from the regression line)
- Spread should be random i.e. no pattern (fanning, W), which indicates **equal variances**

# Model Fit

How well does our fitted model represent the relationship between the variables?

## ANOVA and linear regression

ANOVA is a variation of linear regression – both partition variance into sum of squares for residuals (variance explained) and sum of squares for error (variance not explained) aka **the components of the F-statistic**.

### ANOVA Output

```
fit ← lm(child ~ parent, data = Galton)
anova(fit)
```

#### Analysis of Variance Table

	Response: child				
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
parent	1	1236.9	1236.93	246.84	< 2.2e-16
	**				
Residuals	926	4640.3			
	5.01				
---					

- `parent Sum Sq`: the variation that `parent` explains in the `child` variable
- `Residuals Mean Sq`: variation (per degree of freedom) that the model does not explain
- The `F-value` is the ratio, i.e. does `parent` explain enough variation in `child` to be considered significant?

$$\text{F-value} = \frac{\text{parent Sum Sq}}{\text{Residuals Mean Sq}} = \frac{1236.9}{5.01} = 246.84$$

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05  
'.' 0.1 ' ' 1
```

## Regression Output

```
fit ← lm(child ~ parent, data = Galton)  
summary(fit)  
  
# F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

## ANOVA and linear regression

ANOVA is a variation of linear regression – both partition variance into sum of squares for residuals (variance explained) and sum of squares for error (variance not explained) aka **the components of the F-statistic**.

### ANOVA Output

The ANOVA suggests that the main effect of parent is statistically significant and large ( $F(1, 926) = 246.84, p < .001$ )

### Regression Output

We fitted a linear model (estimated using OLS) to predict child with parent (formula:  $\text{child} \sim \text{parent}$ ). The model explains a statistically significant and moderate proportion of variance ( $R^2 = 0.21, F(1, 926) = 246.84, p < .001$ ). Within this model, the effect of parent is statistically significant and positive ( $\beta_1 = 0.65, 95\% \text{ CI } [0.57, 0.73], t(926) = 15.71, p < .001$ ).

#### i Note

For simple linear regression, the significance of the predictor (i.e. `child`) is the same as the model significance.

# Interpret output

| Model fit and predictions

## Model fit

```
summary(fit)
```

Call:

```
lm(formula = child ~ parent, data = Galton)
```

Residuals:

Min	1Q	Median	3Q	Max
-7.8050	-1.3661	0.0487	1.6339	5.9264

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	23.94153	2.81088	8.517	<2e-16 **
parent	0.64629	0.04114	15.711	<2e-16 **
---				

$$\widehat{\text{child}} = 23.9 + 0.65 \cdot \text{parent}$$

For every unit change in parent (i.e. 1 inch), we expect a 0.65 unit change in child.

How much variation is explained?  $R^2 = 0.21 = 21\%$

- **Multiple R<sup>2</sup>**: proportion of variance in the response variable explained by the model.
- **Adjusted R<sup>2</sup>**: as above but adjusted for the number of predictors in the model.
  - ▶ For multiple linear regression
  - ▶ It only increases if the new term improves the model more than would be expected by chance
  - ▶ *Always lower than multiple R<sup>2</sup>*

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05  
'.' 0.1 ' ' 1
```

```
Residual standard error: 2.239 on 926 degrees of  
freedom
```

```
Multiple R-squared:  0.2105,    Adjusted R-  
squared:  0.2096
```

```
F-statistic: 246.8 on 1 and 926 DF,  p-value: <  
2.2e-16
```

## Making predictions

What is the predicted child height for a parent height of 70 inches?

```
child ← 23.9 + 0.65 * 70  
child
```

```
[1] 69.4
```

We use `predict()` to make predictions – it takes in the `lm()` model, recreates the equation and applies it to new data.

```
predict(fit, data.frame(parent = 70)) # using 70 as this is the value we want to sub in and predict
```

```
1  
69.18187
```

### i Note

How good is our prediction actually? What if we had more parents and children, would the equation still hold up? We cover this in Week 9.

# Transformations

What if assumptions are not met, or we want to improve the model?

## What if assumptions are not met?

### Violations of...

- **Linearity** can cause systematically wrong predictions
- **Homoskedasticity** makes it difficult to estimate “true” standard deviation of errors (i.e. noisy estimates)
- **Normality** can compromise inferences and hypothesis testing

## How do we solve these problems?

- Use less restrictive (but more complicated) models, e.g. generalised linear models, non-parametric techniques (ENVX3002)
- Perform variance corrections (complicated)
- **Transform the response variable ( $y$ )** to stabilise variance and correct normality
- **Transform the predictor variable ( $x$ )** if issues still exist in the diagnostics

**i** Note

We can also perform transformations to improve the model fit, but **beware of overfitting** – we want to make reasonable predictions, not fit the data!

## Example: air quality

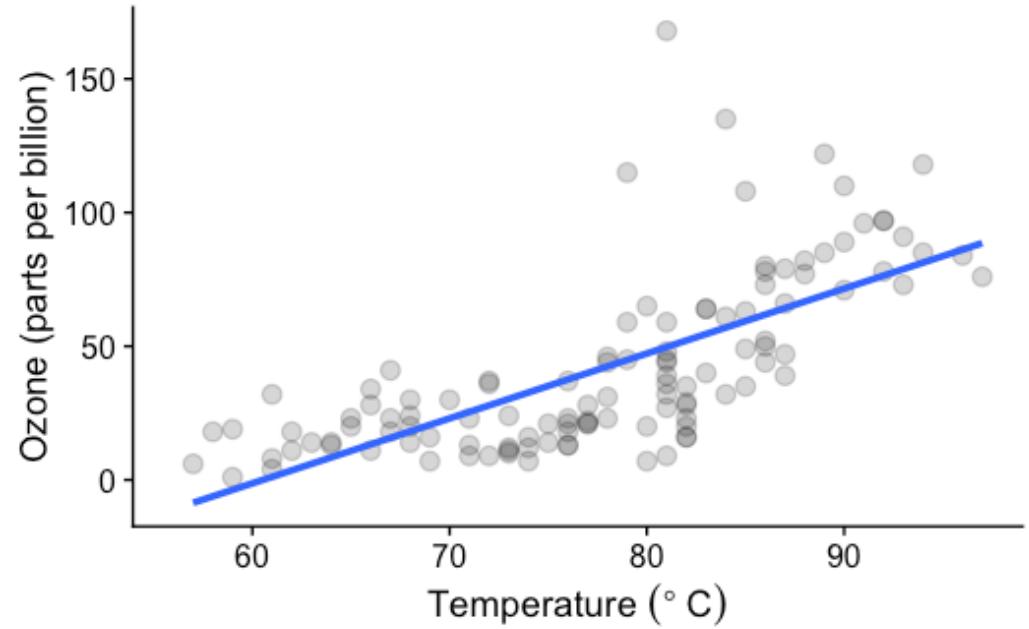
Daily air quality measurements in New York, May to September 1973.

```
str(airquality)
```

```
'data.frame': 153 obs. of 6 variables:  
 $ Ozone : int 41 36 12 18 NA 28 23 19 8  
NA ...  
 $ Solar.R: int 190 118 149 313 NA NA 299 99 19  
194 ...  
 $ Wind   : num 7.4 8 12.6 11.5 14.3 14.9 8.6  
13.8 20.1 8.6 ...  
 $ Temp   : int 67 72 74 62 56 66 65 59 61  
69 ...  
 $ Month  : int 5 5 5 5 5 5 5 5 5 5 ...  
 $ Day    : int 1 2 3 4 5 6 7 8 9 10 ...
```

We start with one variable: is ozone concentration influenced by temperature?

```
ggplot(airquality, aes(x = Temp, y = Ozone)) +  
  geom_point(alpha = .2, size = 3) +  
  labs(  
    x = expression("Temperature " ( degree~C)),  
    y = "Ozone (parts per billion)") +  
  geom_smooth(method = "lm", se = FALSE)
```

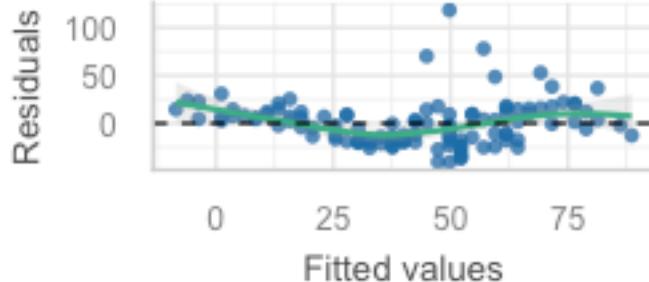


## Assumption checks

```
fit ← lm(Ozone ~ Temp, data = airquality)
performance::check_model(fit, check = c("linearity", "qq", "homogeneity", "outliers")) # check
specific assumptions
```

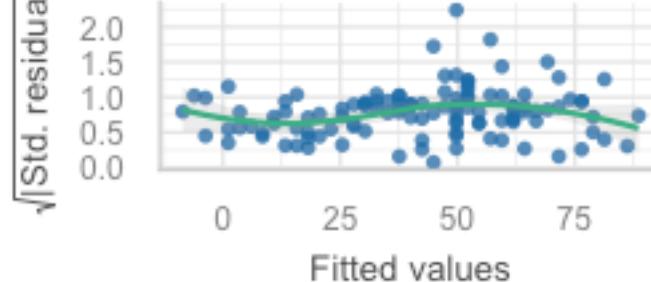
### Linearity

Reference line should be flat and horizontal



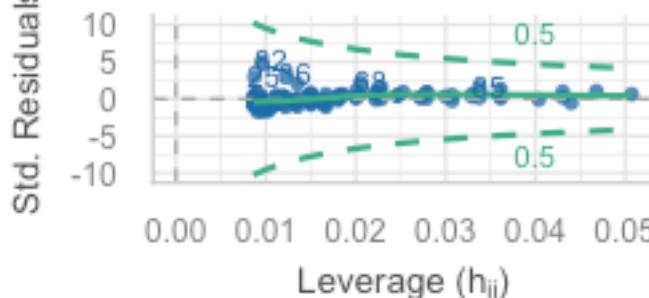
### Homogeneity of Variance

Reference line should be flat and horizontal



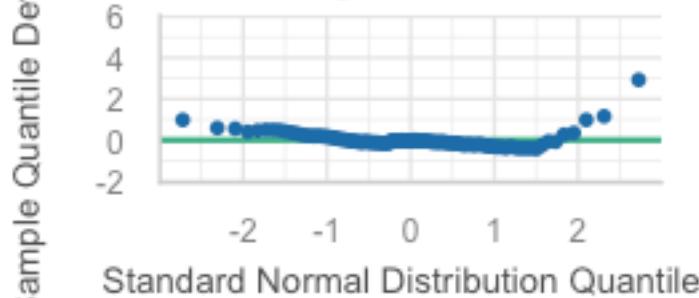
### Influential Observations

Points should be inside the contour lines



### Normality of Residuals

Dots should fall along the line



Is a simple linear model appropriate? *Depends on your threshold for what is acceptable.*

## Backtransforming – FYI

A log transformation (natural or a base) is relatively easy to back-transform.

$$\log(\widehat{Ozone}) = -1.8380 + 0.0675 \times Temp$$

$$\widehat{Ozone} = e^{-1.8380 + 0.0675 \times Temp} = e^{-1.8380} \times e^{0.0675 \times Temp}$$

But given we are focused on a 1-unit change of `Temp`,  $\widehat{Ozone}$  changes by  $e^{0.0675} = 1.07$  **times**.

If this had been a `sqrt()` transformation...

$$\sqrt{\widehat{Ozone}} = -1.8380 + 0.0675 \times Temp$$

$$\widehat{Ozone} = (-1.8380 + 0.0675 \times Temp)^2 = 3.3782 - (0.2481 \times Temp) + (0.0675 \times Temp)^2$$

## Interpreting log transformations – FYI

- Log-linear:  $\text{Log}(Y) = \beta_0 + \beta_1 x$ 
  - An increase of  $x$  by 1 unit corresponds to a  $\beta_1$  unit increase in  $\log(Y)$
  - An increase of  $x$  by 1 unit corresponds to approximately a  $\beta_1 \times 100\%$  increase in  $Y$
- Linear-log:  $Y = \beta_0 + \beta_1 \log(x)$ 
  - An increase of 1% in  $x$  corresponds to a  $\frac{\beta_1}{100}$  increase in  $Y$
- Log-log:  $\text{Log}(Y) = \beta_0 + \beta_1 \log(x)$ 
  - An increase of 1% in  $x$  corresponds to a  $\beta_1\%$  increase in  $Y$

## Percent change with $\ln$ transformation – FYI

Interpreting as a percent change can be more meaningful - it can be done with any log transformation (substitute  $e$  below for 10 or any other base), but the **quick approximation only works with natural log transformations**.

If  $y$  has been transformed with a natural log ( $\log(y)$ ), for a one-unit increase in  $x$  the **percent change in  $y$**  (not  $\log(y)$ ) is calculated with:

$$\Delta y\% = 100 \cdot (e^{\beta_1} - 1)$$

If  $\beta_1$  is small (i.e.  $-0.25 < \beta_1 < 0.25$ ), then:  $e^{\beta_1} \approx 1 + \beta_1$ . So  $\Delta y\% \approx 100 \cdot \beta_1$ .

$\beta$	Exact $(e^\beta - 1)\%$	Approximate $100 \cdot \beta$
-0.25	-22.13	-25
-0.1	-9.52	-10
0.01	1.01	1
0.1	10.52	10
0.25	28.41	25
0.5	64.87	50
2	638.91	200

- **$y$  transformed:** a one-unit increase in  $x$  is *approximately* a  $\beta_1\%$  change in  $y$ .
- **$x$  transformed:** a 1% increase in  $x$  is *approximately* a  $0.01 \cdot \beta_1$  change in  $y$ .
- **Both  $x$  and  $y$  transformed:** a 1% increase in  $x$  is *approximately* a  $\beta_1\%$  change in  $y$ .

## Transforming Ozone

Let's transform Ozone using the natural log (`log()`).

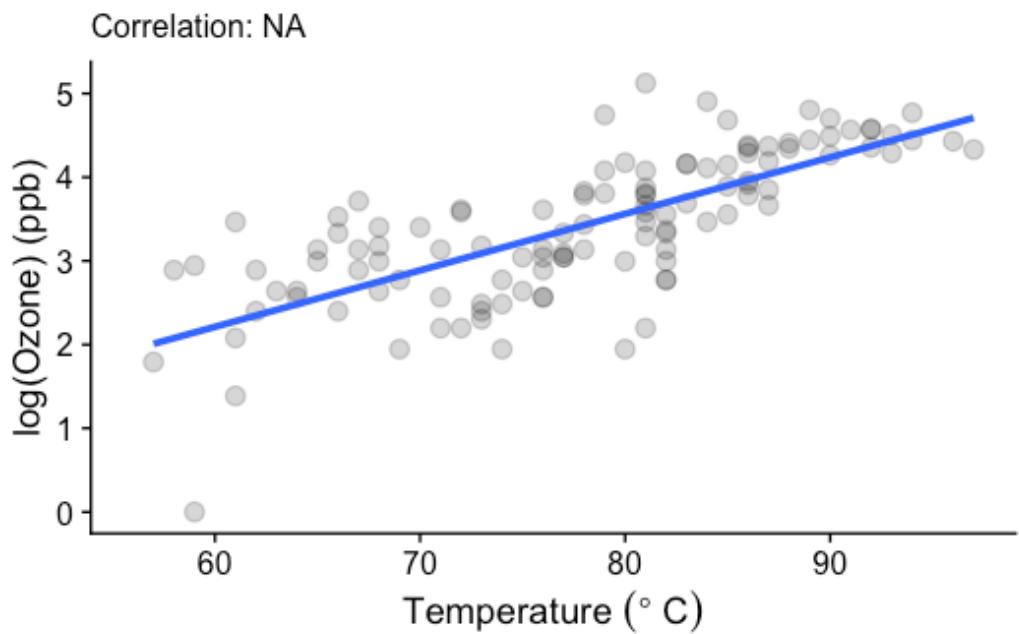
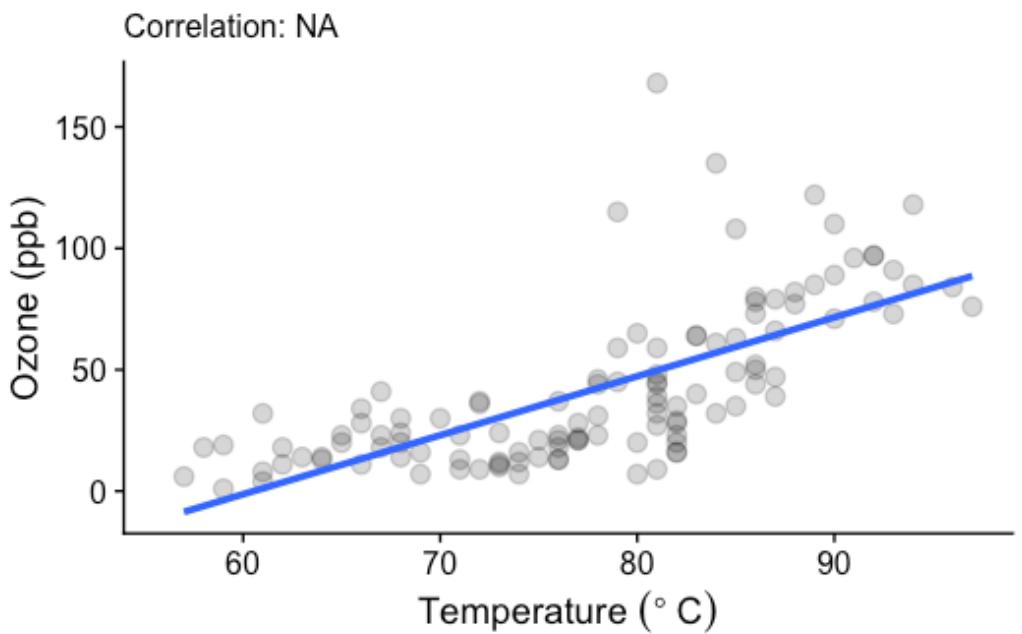
```
fit_log ← lm(log(Ozone) ~ Temp, data = airquality)
```

### Before

```
ggplot(airquality, aes(x = Temp, y = Ozone)) +  
  geom_point(alpha = .2, size = 3) +  
  labs(  
    x = expression("Temperature " ( degree~C)),  
    y = "Ozone (ppb)") +  
  geom_smooth(method = "lm", se = FALSE) +  
  labs(subtitle = paste("Correlation:",  
    round(cor(airquality$Temp, airquality$Ozone),  
    2)))
```

### After

```
ggplot(airquality, aes(x = Temp, y =  
  log(Ozone))) +  
  geom_point(alpha = .2, size = 3) +  
  labs(  
    x = expression("Temperature " ( degree~C)),  
    y = "log(Ozone) (ppb)") +  
  geom_smooth(method = "lm", se = FALSE) +  
  labs(subtitle = paste("Correlation:",  
    round(cor(airquality$Temp,  
    log(airquality$Ozone)), 2)))
```



## Assumption: Linearity

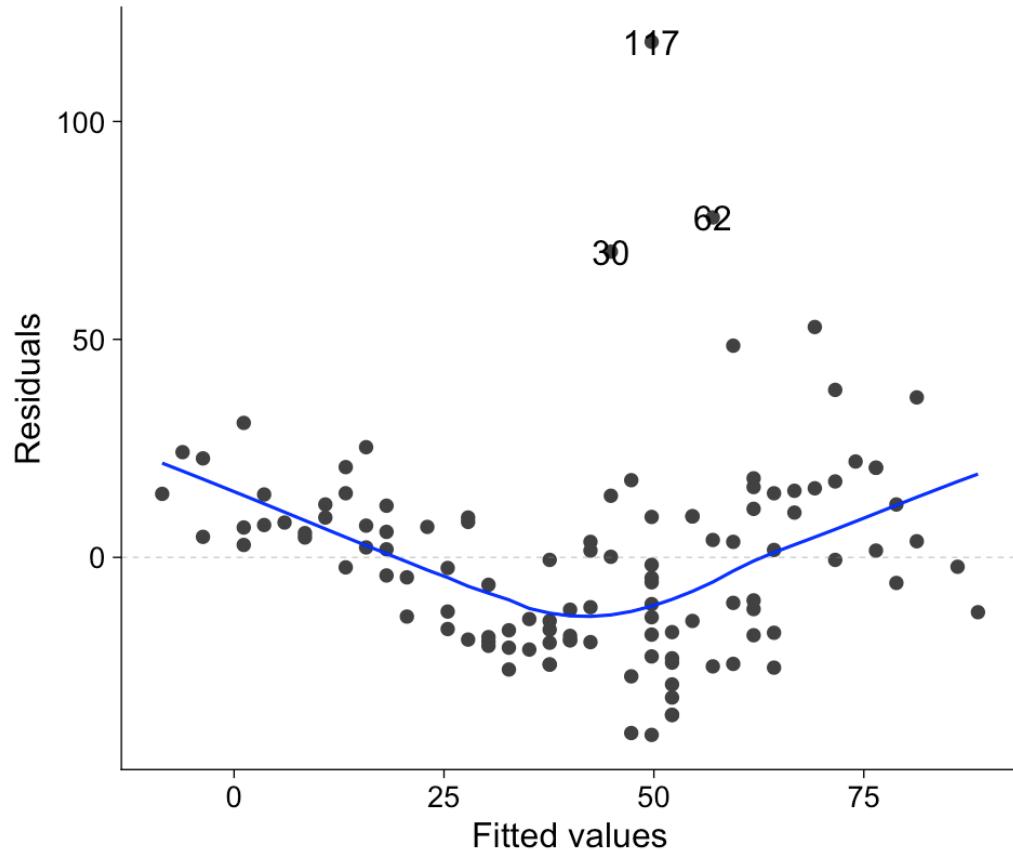
Before

```
autoplot(fit, 1, ncol = 1) +  
cowplot::theme_cowplot(font_size = 24)
```

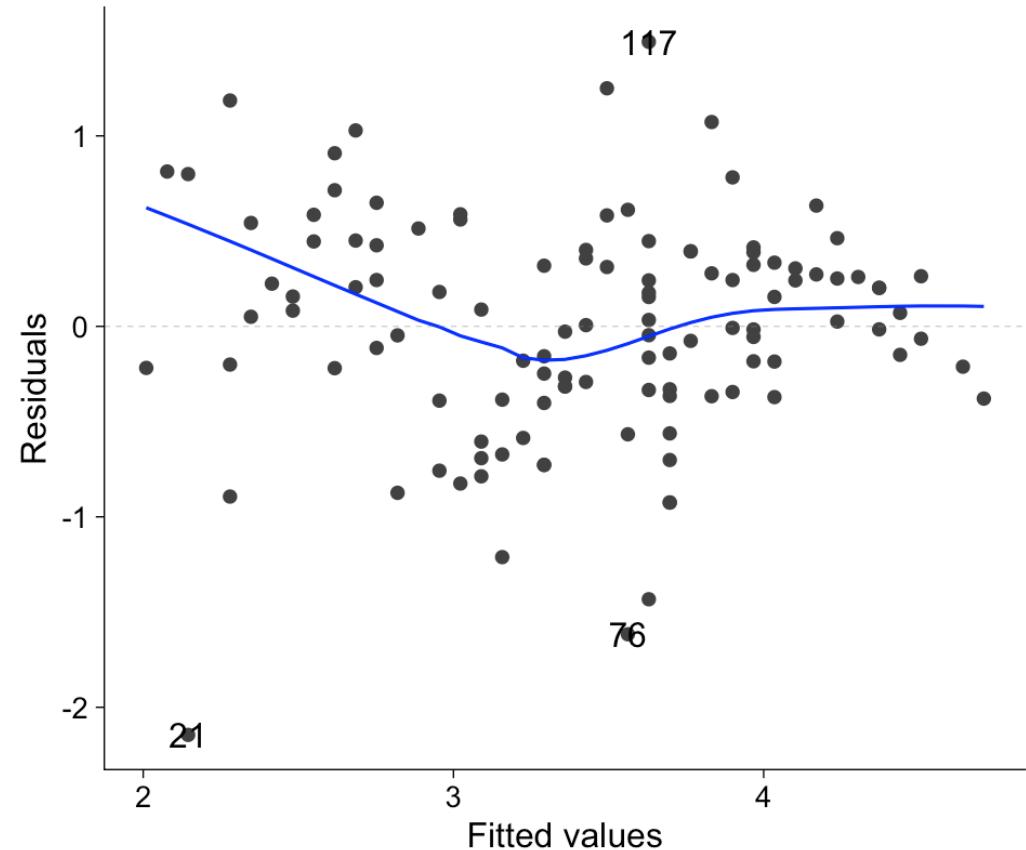
After

```
autoplot(fit_log, 1, ncol = 1) +  
cowplot::theme_cowplot(font_size = 24)
```

### Residuals vs Fitted



### Residuals vs Fitted



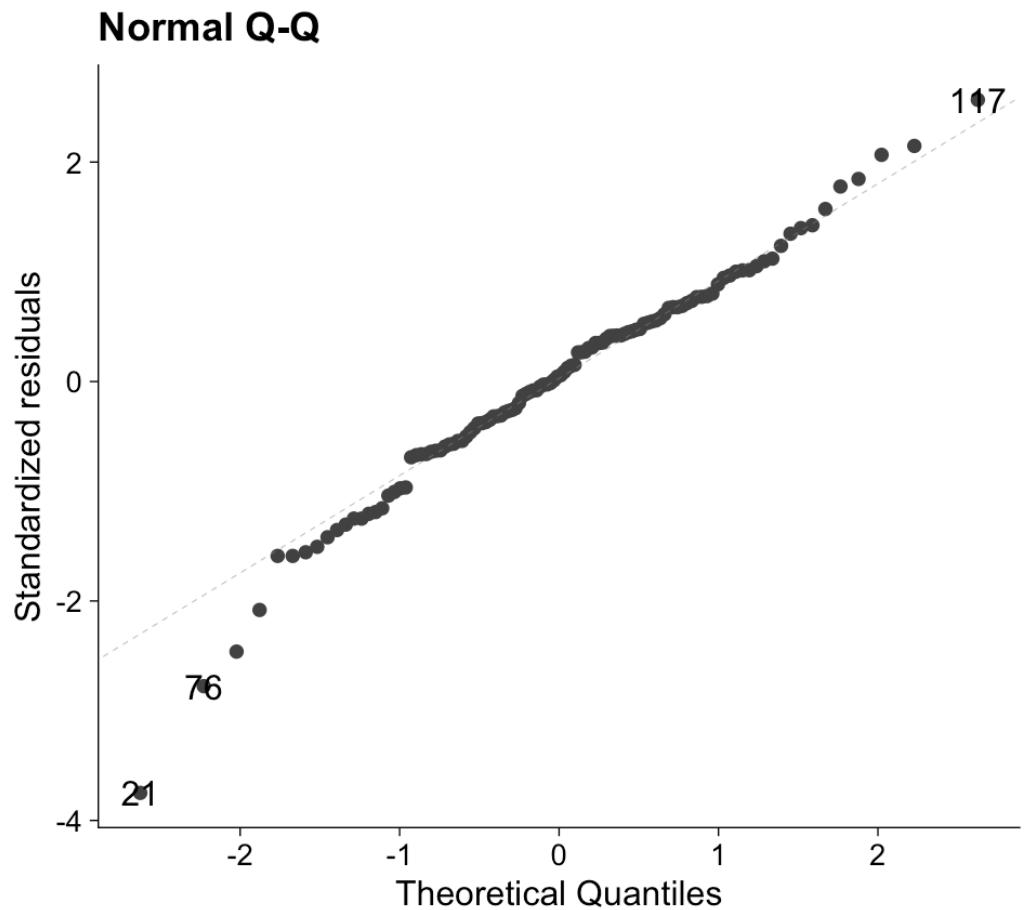
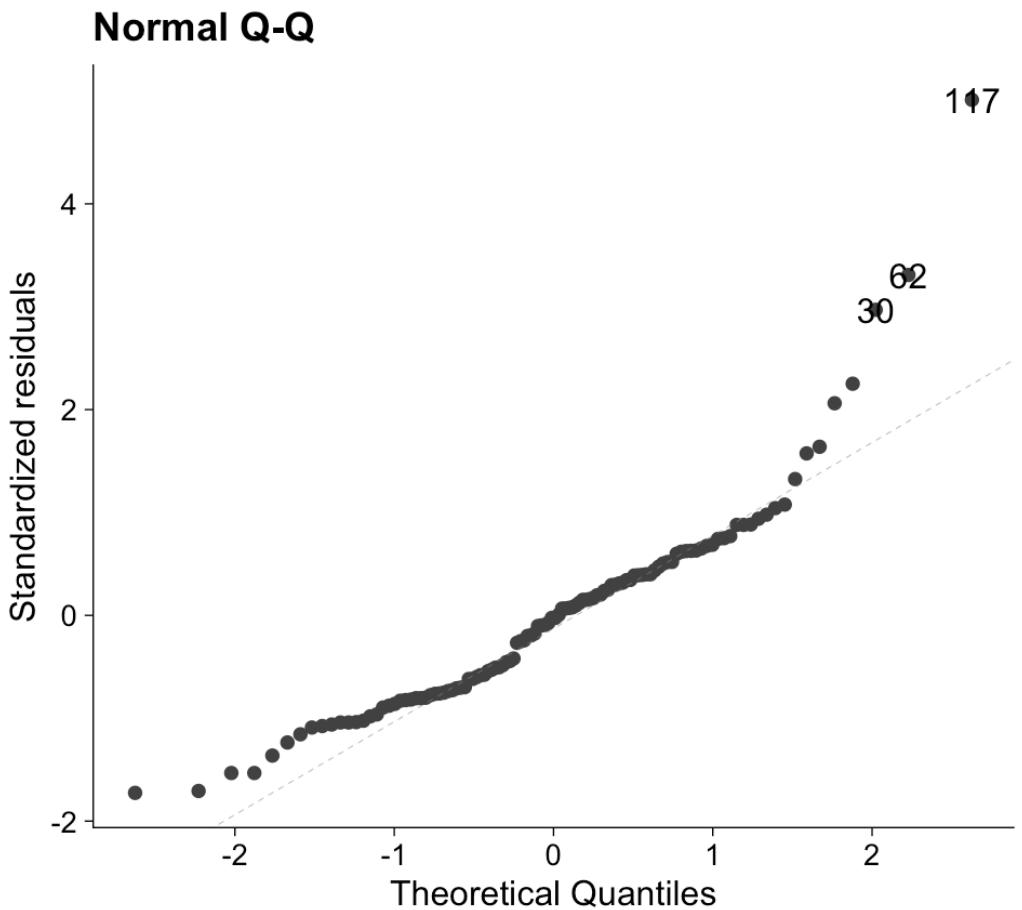
## Assumption: Normality

### Before

```
autoplot(fit, 2, ncol = 1) +  
cowplot::theme_cowplot(font_size = 24)
```

### After

```
autoplot(fit_log, 2, ncol = 1) +  
cowplot::theme_cowplot(font_size = 24)
```



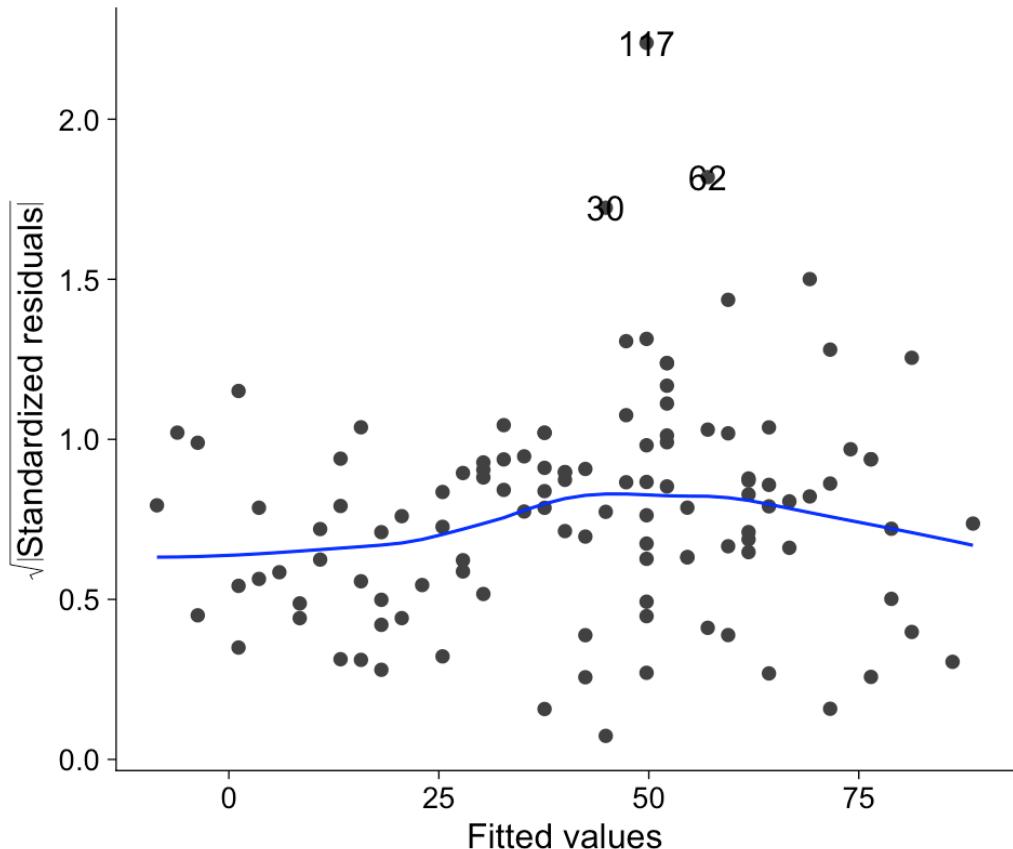
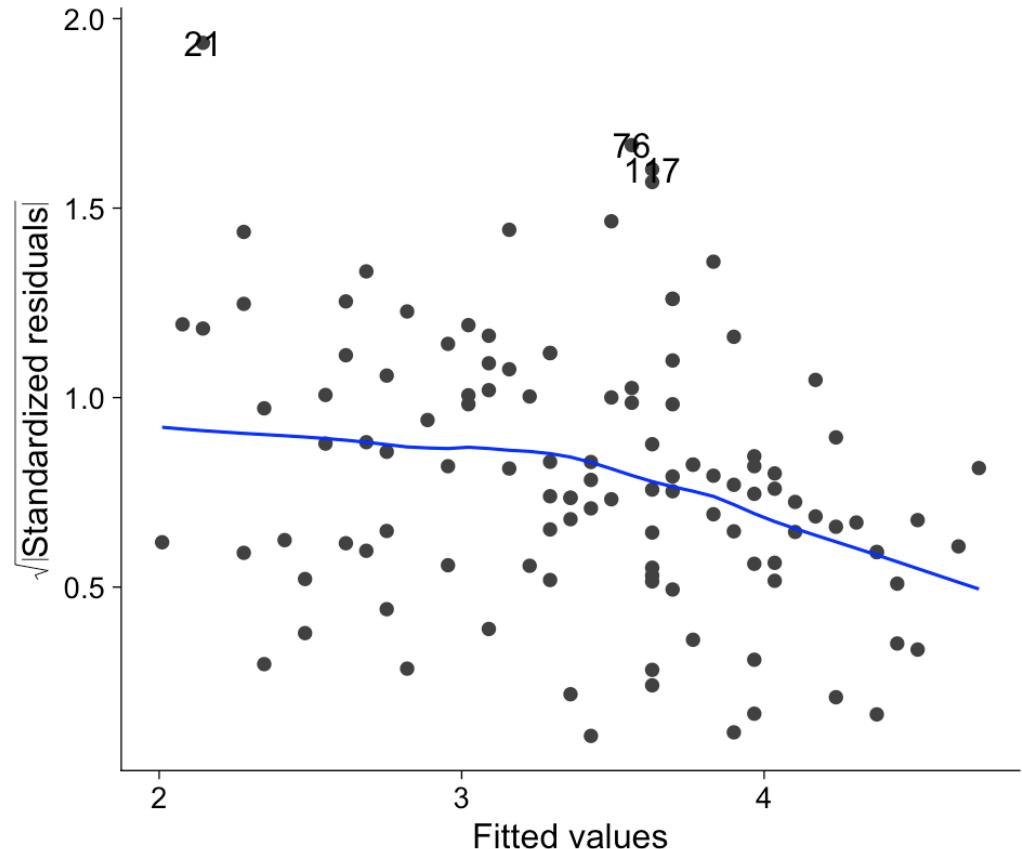
## Assumption: Equal variances

Before

```
autoplot(fit, 3, ncol = 1) +  
cowplot::theme_cowplot(font_size = 24)
```

After

```
autoplot(fit_log, 3, ncol = 1) +  
cowplot::theme_cowplot(font_size = 24)
```

**Scale-Location****Scale-Location**

## Is transforming better?

### Before

```
summary(fit)
```

Call:  
lm(formula = Ozone ~ Temp, data = airquality)

Residuals:

Min	1Q	Median	3Q	Max
-40.729	-17.409	-0.587	11.306	118.271

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-146.9955	18.2872	-8.038 9.37e-13 ***	
Temp	2.4287	0.2331	10.418 < 2e-16 ***	

### After

```
summary(fit_log)
```

Call:  
lm(formula = log(Ozone) ~ Temp, data = airquality)

Residuals:

Min	1Q	Median	3Q	Max
-2.14469	-0.33095	0.02961	0.36507	1.49421

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-1.83797	0.45100	-4.075	8.53e-05 ***
Temp	0.06750	0.00575	11.741	< 2e-16

```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
'.' 0.1 ' ' 1

Residual standard error: 23.71 on 114 degrees of
freedom
(37 observations deleted due to missingness)
Multiple R-squared:  0.4877,    Adjusted R-
squared:  0.4832
F-statistic: 108.5 on 1 and 114 DF,  p-value: <
2.2e-16

```

```

***  

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
'.' 0.1 ' ' 1

Residual standard error: 0.5848 on 114 degrees
of freedom
(37 observations deleted due to missingness)
Multiple R-squared:  0.5473,    Adjusted R-
squared:  0.5434
F-statistic: 137.8 on 1 and 114 DF,  p-value: <
2.2e-16

```

The transformed model model equation is:

$$\widehat{\log(Ozone)} = -1.8380 + 0.0675 \times Temp$$

A 1 degree ( $^{\circ}\text{F}$ ) increase in temperature is associated with a:

- 0.0675 increase in `log(Ozone)` concentration
- $e^{0.0675} = 1.07$  times increase in `Ozone` concentration

- Approximately a 6.75% increase in Ozone concentration

## Multiple linear regression

### The MLR model

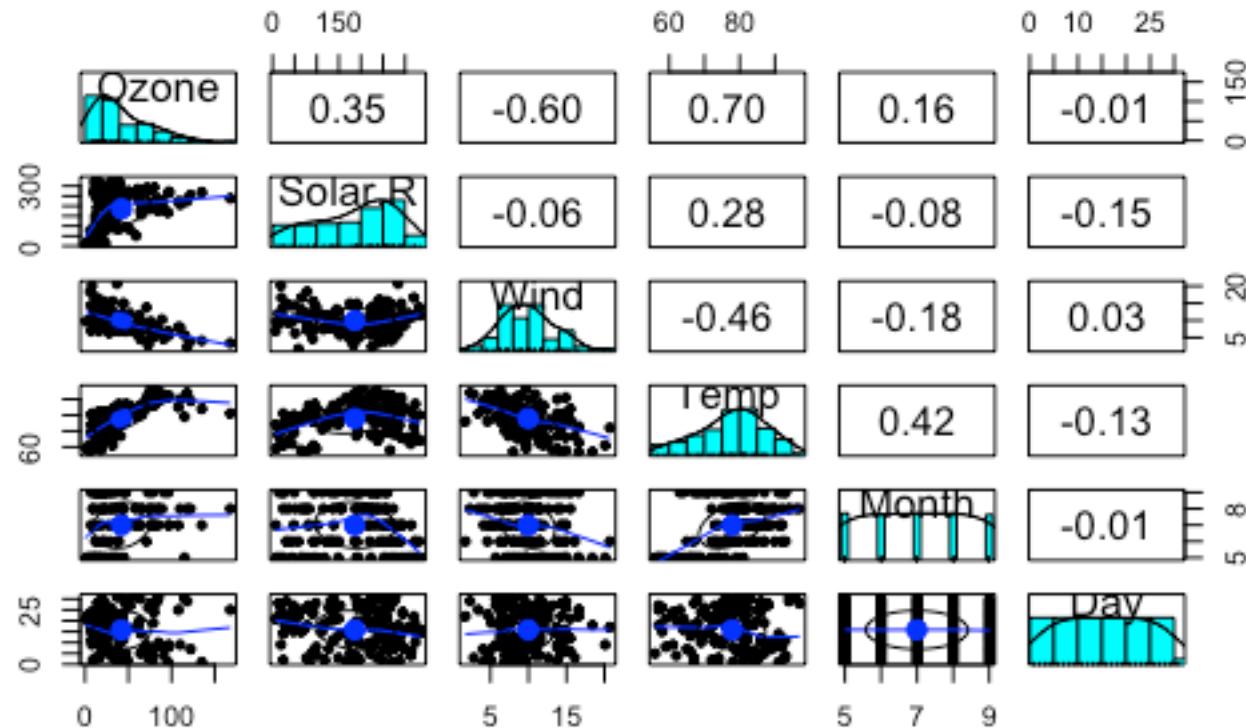
$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \epsilon$$

where

- a response variable ( $Y$ ) which we wish to predict using predictor variables ( $x_k$ )
- $\beta_0$  is the y-intercept
- $\beta_k$  is the partial regression coefficient associated with the  $k^{th}$  predictor variable
- $\epsilon$  is error and  $\epsilon \sim N(0, \sigma^2)$

## Can we use more predictors?

```
psych::pairs.panels(airquality)
```



Can we improve the current model by adding *wind* and *solar radiation* as additional predictors?

**Can we use more predictors?**

**From:**

$$\log(\text{size})_i = \beta_0 + \beta_1 \text{Temp}_i + \epsilon_i$$

**To:**

$$\log(\text{size})_i = \beta_0 + \beta_1 \text{Temp}_i + \beta_2 \text{Solar.R}_i + \beta_3 \text{Wind}_i + \epsilon_i$$

## Can we use more predictors?

$$\log(\text{size})_i = \beta_0 + \beta_1 \text{Temp}_i + \beta_2 \text{Solar.R}_i + \beta_3 \text{Wind}_i + \epsilon_i$$

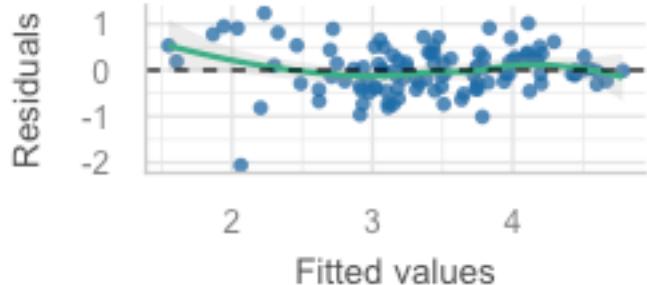
```
multi_fit ← lm(log(Ozone) ~ Temp + Solar.R + Wind, data = airquality)
```

## Assumptions

```
performance::check_model(multi_fit, check = c("linearity", "qq", "homogeneity", "outliers")) #  
check specific assumptions
```

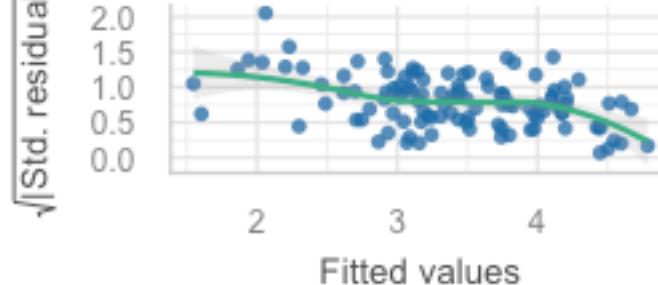
### Linearity

Reference line should be flat and horizontal



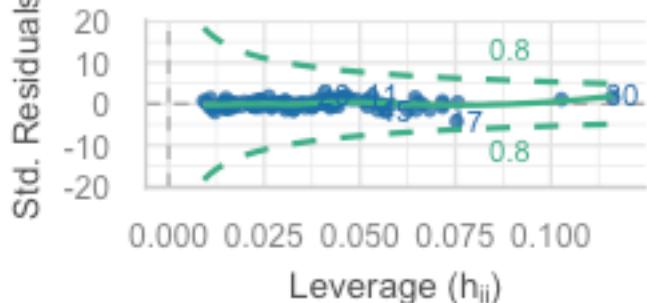
### Homogeneity of Variance

Reference line should be flat and horizontal



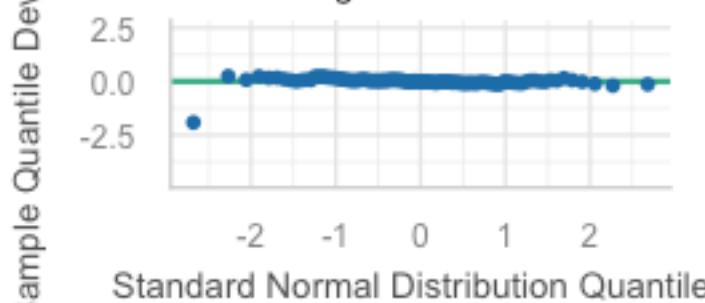
### Influential Observations

Points should be inside the contour lines



### Normality of Residuals

Bots should fall along the line



There is one additional assumption for multiple linear regression. **Collinearity** is when two or more predictors are very highly correlated. If the predictors are basically identical, the model cannot distinguish how much variability each explains. (Correlations in previous slides look fine).

## Hypothesis

For multiple linear regression, there are two hypothesis tests:

- Individual predictors, where the significance of each predictor is tested via t-tests

$$H_0 : \beta_k = 0$$

$$H_1 : \beta_k \neq 0$$

- The overall model, which is tested with an F-test (to get F-stat).  $H_0$  is an intercept-only model (i.e. the mean), so if at least one predictor is useful, the model is better than the intercept-only model.

$$H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$$

$$H_1 : \text{At least one } \beta_k \neq 0$$

## Model Fit

```
summary(multi_fit)
```

Call:

```
lm(formula = log(Ozone) ~ Temp + Solar.R + Wind, data = airquality)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.06193	-0.29970	-0.00231	0.30756	1.23578

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.2621323	0.5535669	-0.474	0.636798
Temp	0.0491711	0.0060875	8.077	1.07e-12 ***
Solar.R	0.0025152	0.0005567	4.518	1.62e-05 ***
Wind	-0.0615625	0.0157130	-3.918	0.000158 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
Residual standard error: 0.5086 on 107 degrees of freedom  
(42 observations deleted due to missingness)
```

```
Multiple R-squared:  0.6644,    Adjusted R-squared:  0.655  
F-statistic: 70.62 on 3 and 107 DF,  p-value: < 2.2e-16
```

Model equation:

$$\widehat{\log(Ozone)} = -0.262 + 0.0492 \cdot Temp + 0.00252 \cdot Solar.R - 0.0616 \cdot Wind$$

## Interpretation

$$\widehat{\log(Ozone)} = -0.262 + 0.0492 \cdot Temp + 0.00252 \cdot Solar.R - 0.0616 \cdot Wind$$

### Holding all other variables constant:

- A one degree ( $^{\circ}\text{F}$ ) increase in `Temp` is associated with a 4.9% increase in `Ozone` concentration.
- A one unit increase in `Solar.R` is associated with a 0.25% increase in `Ozone` concentration.
- A one unit increase in `Wind` is associated with a 6.2% decrease in `Ozone` concentration.

Automating extracting the model equation into latex using `extract_eq()` from the package `equatiomatic`:

```
equatiomatic::extract_eq(multi_fit, use_coefs = TRUE, coef_digits = 3) |> print()
```

```
$$
\operatorname{\widehat{log(Ozone)}} = -0.262 + 0.049(\operatorname{Temp}) +
0.003(\operatorname{Solar.R}) - 0.062(\operatorname{Wind})
$$
```

## Is MLR model better?

```
sjPlot::tab_model(fit_log, multi_fit, digits = 4, show.ci = FALSE)
```

	log(Ozone)		log(Ozone)	
Predictors	Estimates	p	Estimates	p
(Intercept)	-1.8380	<0.001	-0.2621	0.637
Temp	0.0675	<0.001	0.0492	<0.001
Solar R			0.0025	<0.001
Wind			-0.0616	<0.001
Observations	116		111	
R <sup>2</sup> / R <sup>2</sup> adjusted	0.547 / 0.543		0.664 / 0.655	

- The adjusted  $R^2$  is higher for the MLR model...
- Interpretation of  $R^2$  is the same as for simple linear regression: how much of the variation in the response variable is explained by the model
- **Are all the variables/predictors needed?** (next week)

# Summing up

- We fit a simple linear model to represent a linear relationship between two variables
  - used method of least squares to find the best fitting line
  - model equation;  $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ 
    - Multiple linear regression
      - Do more predictors improve model fit?
      - $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \epsilon$
- Hypothesis testing with linear models + Is our model the best representation of the relationship?
- Check assumptions to understand the validity of the model
  - collinearity, linearity, independence, normality, equal variance (CLINE)
- Transformations to meet assumptions and improve model fit
- Interpreting model output + F-test via ANOVA and summary(), R<sup>2</sup>

## **Next lecture: Variable selection**

We will discuss how to select the best subset of predictors for a model.

# Thanks!

Questions? Comments?

Slides made with Quarto