

Introduction to statistics in R

Rémy Beugnon & Malte Jochum



- Rémy Beugnon

<https://remybeugnon.netlify.app>

@BeugnonRemy



- Malte Jochum

<http://maltejochum.de/>

@MalteJochum



In this lecture:

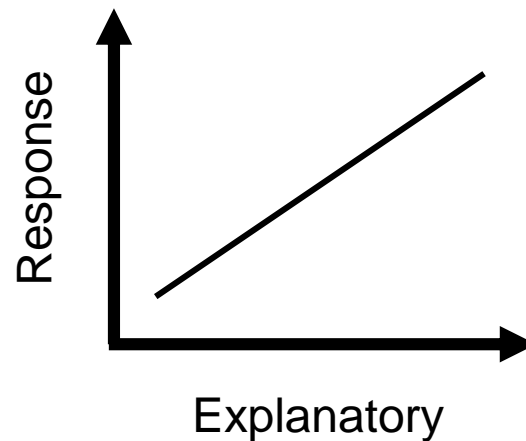
- 1. The stepwise process to analyze your data**
- 2. Application**
- 3. Practical on your own**
- 4. Conclusion**

In this lecture:

- 1. The stepwise process to analyze your data**

In this lecture:

- 1. The stepwise process to analyze your data**
Focus on linear models with continuous predictors.

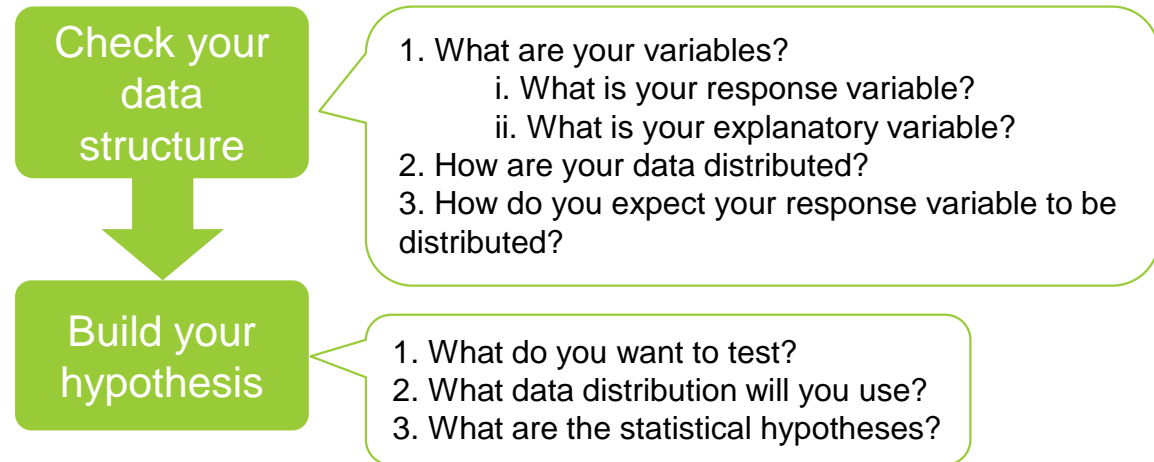


Steps to analyze your data

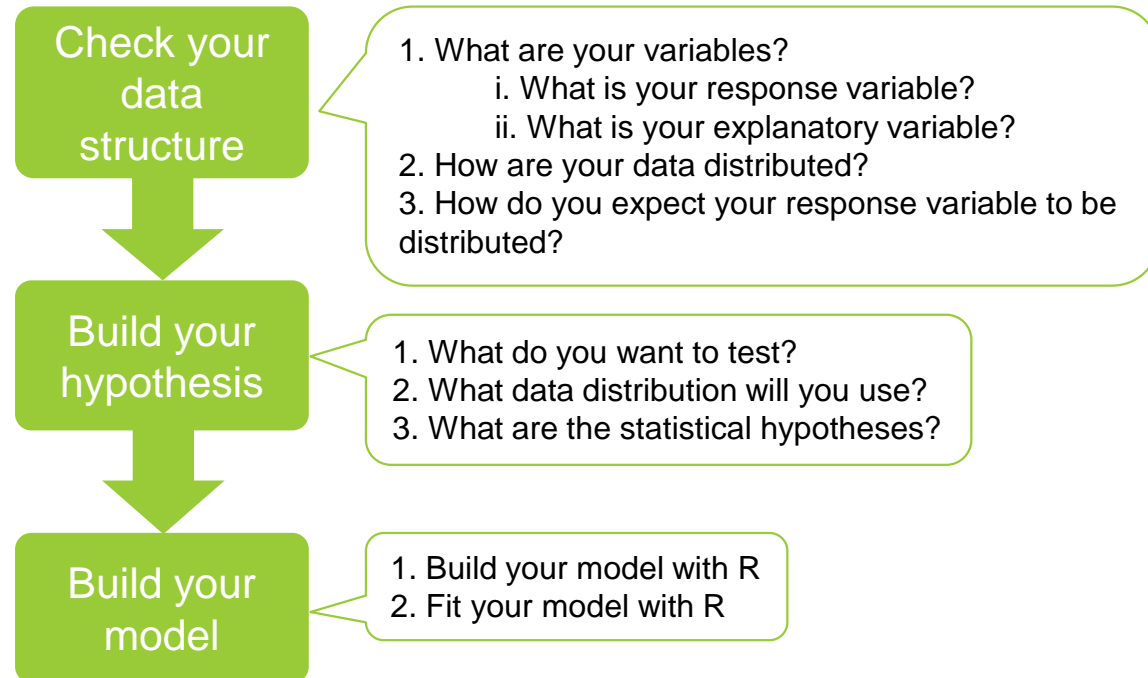
Check your
data
structure

1. What are your variables?
 - i. What is your response variable?
 - ii. What is your explanatory variable?
2. How are your data distributed?
3. How do you expect your response variable to be distributed?

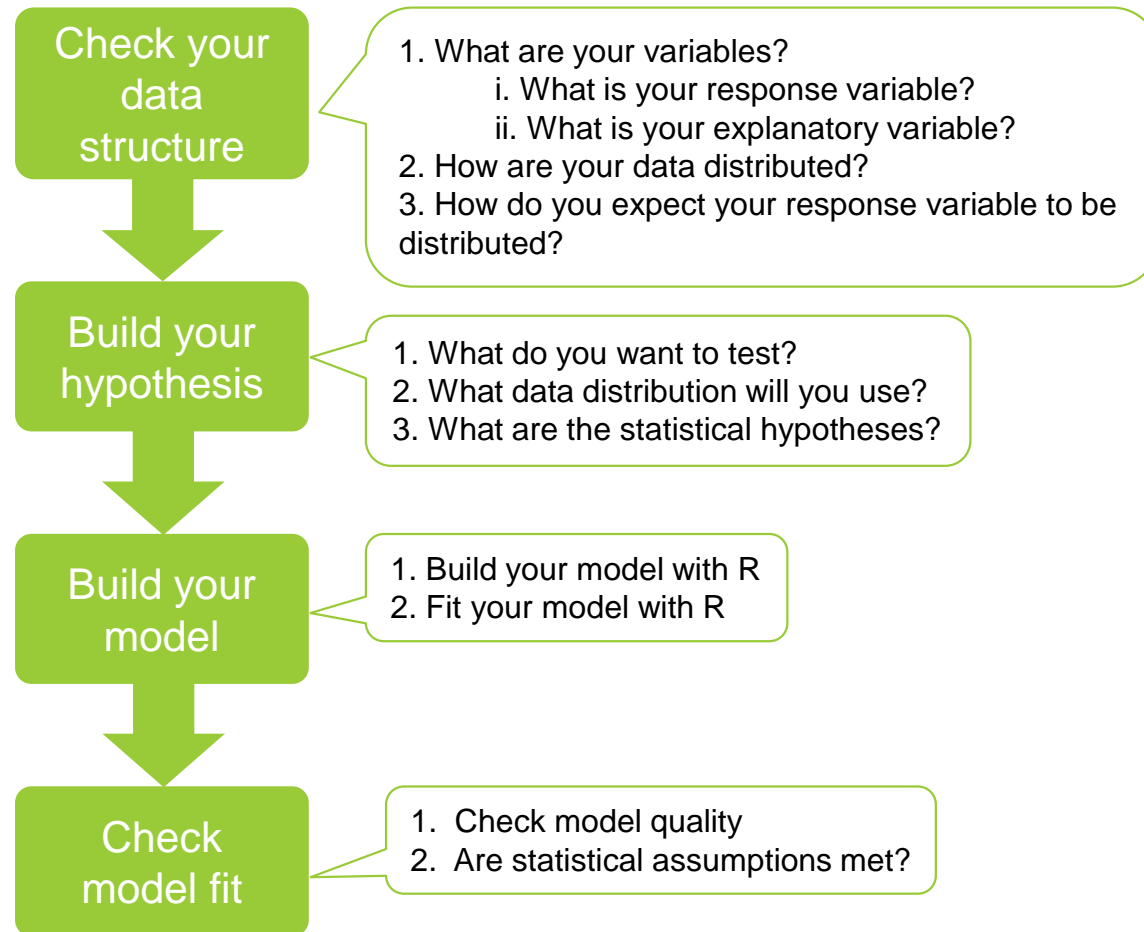
Steps to analyze your data



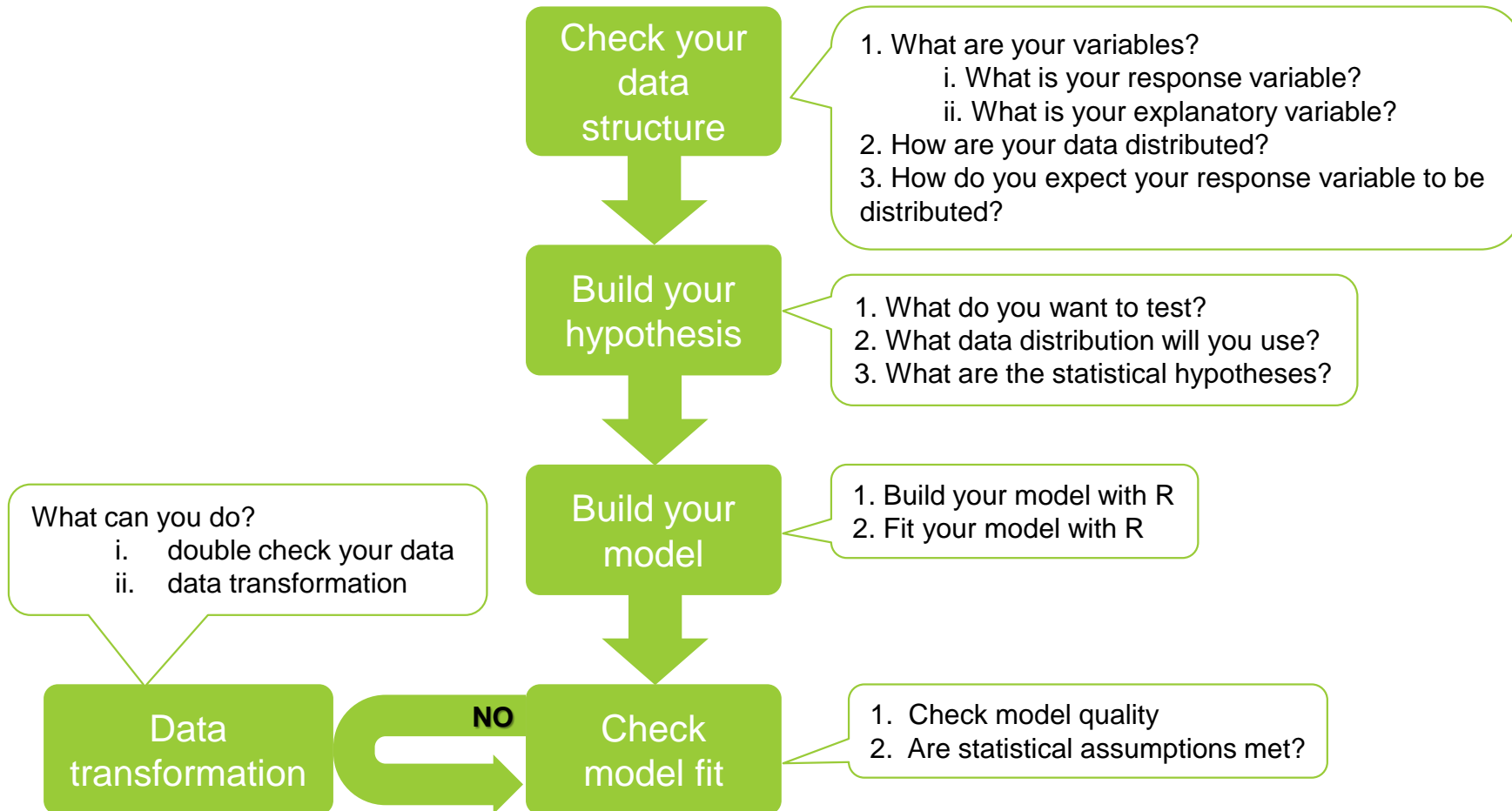
Steps to analyze your data



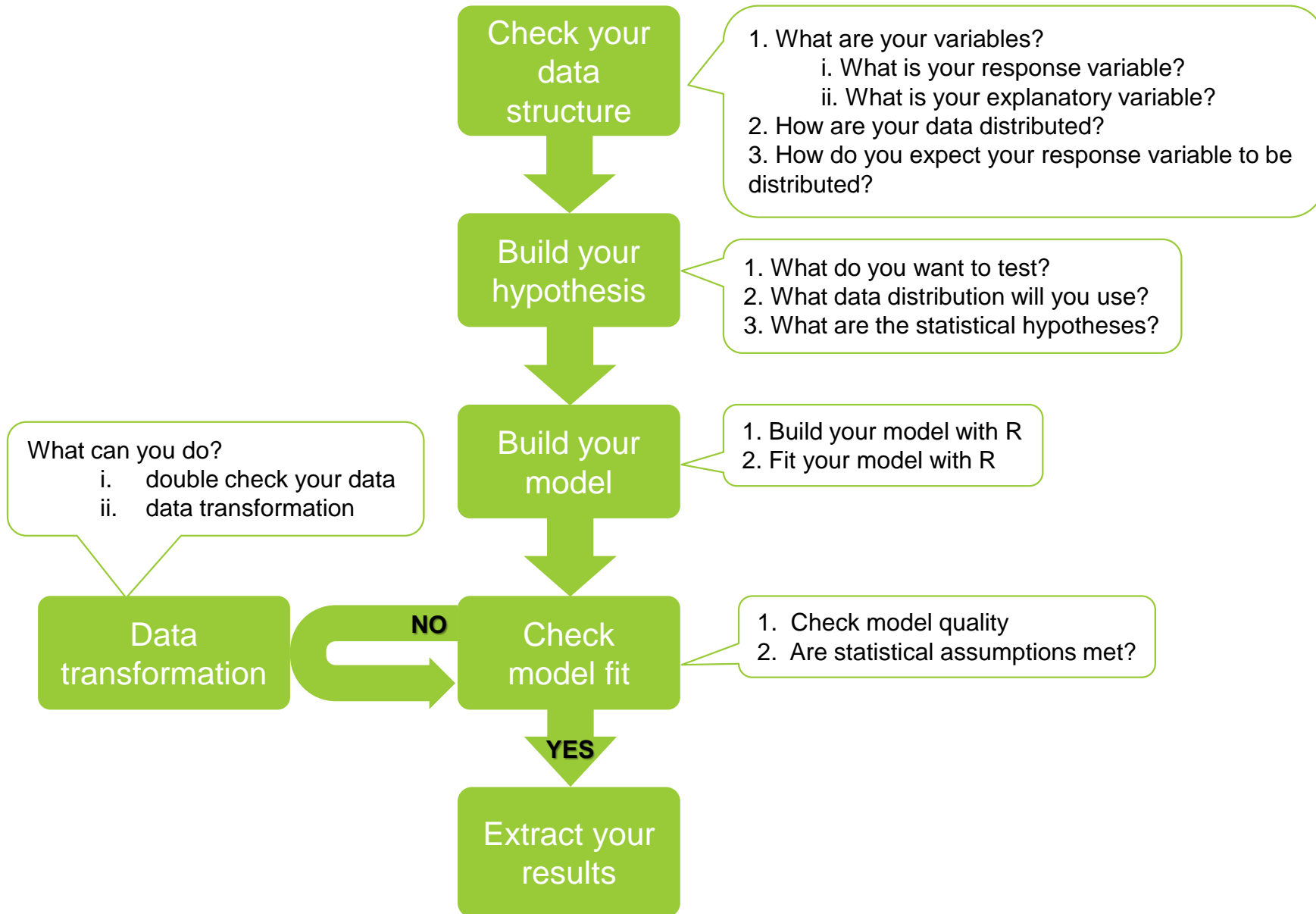
Steps to analyze your data



Steps to analyze your data



Steps to analyze your data



In this lecture:

- 1. The stepwise process to analyze your data**
- 2. Application**

Who to do that using RStudio

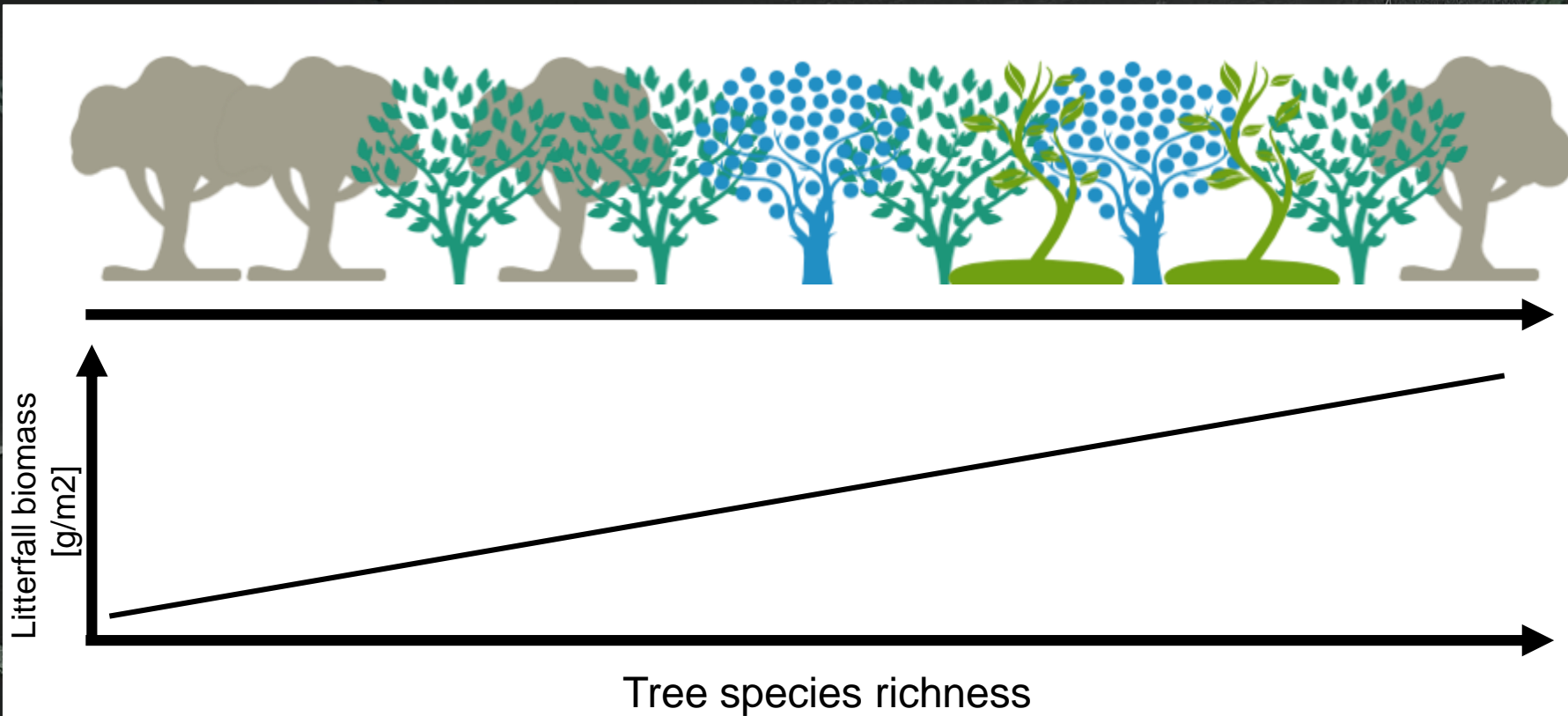
- **You need**

- RStudio
- R version 4.0 or higher
- The following packages:
 - Data handling: **dplyr**
 - Model quality checks: **performance** (needed with **see**)
 - Extract your results: **ggeffects**
 - Plot: **ggplot2** (join the course from Steph for more details)
- A dataset to analyze

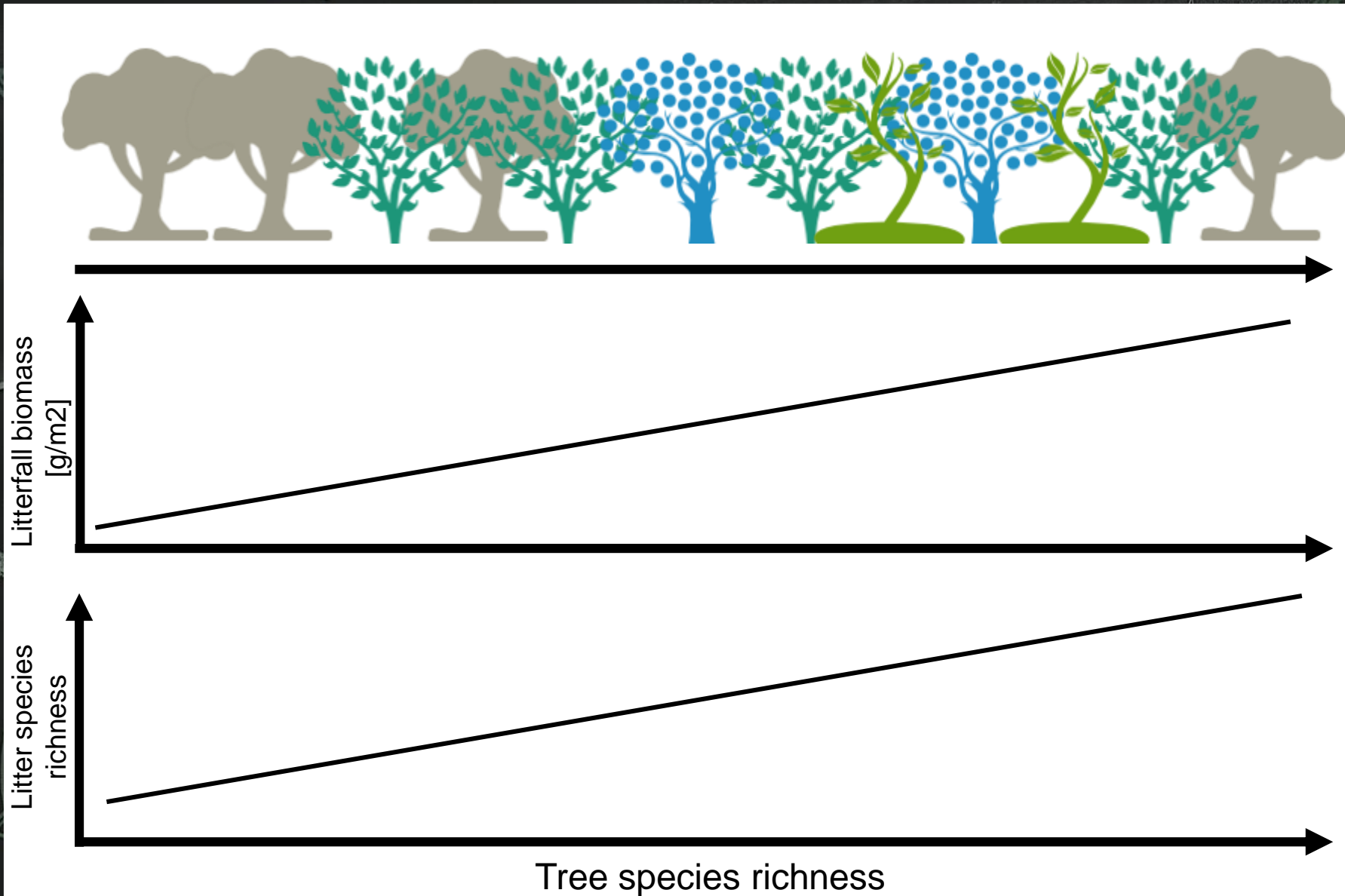
Example: tree diversity effect on litterfall and decomposition



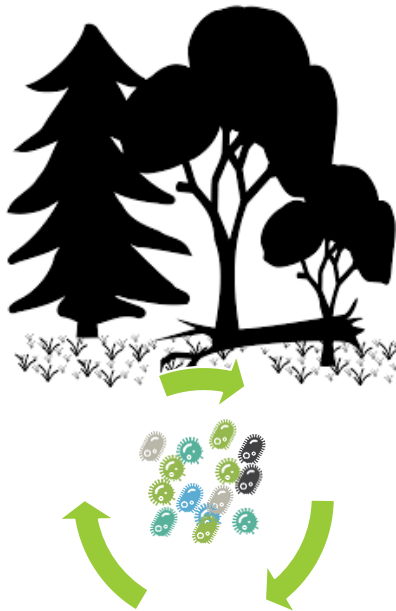
Example: tree diversity effect on litterfall and decomposition



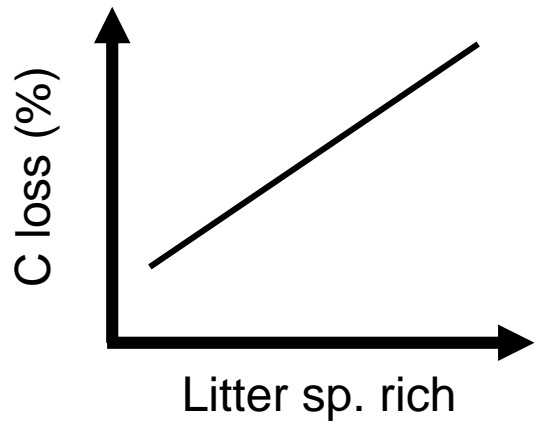
Example: tree diversity effect on litterfall and decomposition



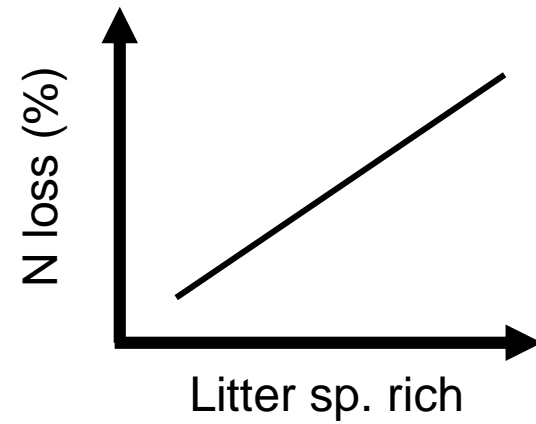
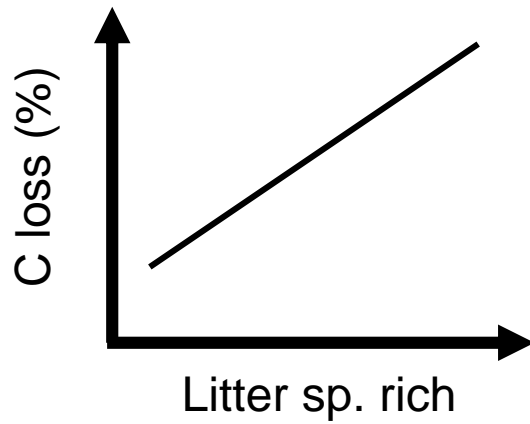
Example: tree diversity effect on litterfall and decomposition



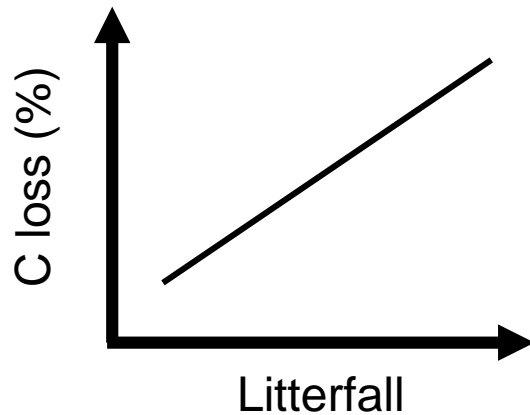
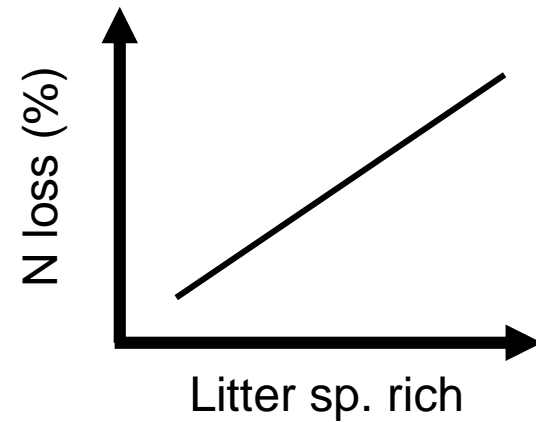
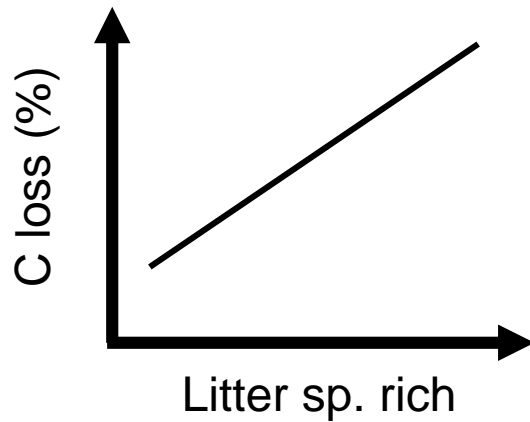
Example: tree diversity effect on litterfall and decomposition



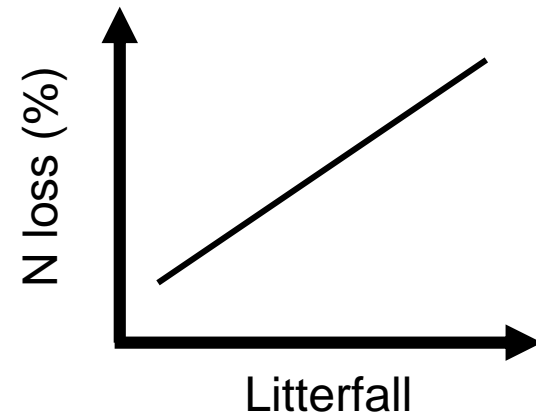
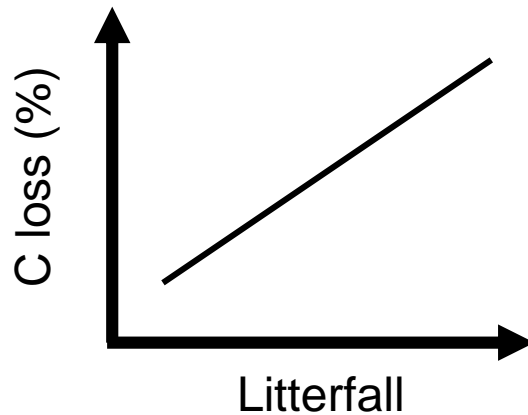
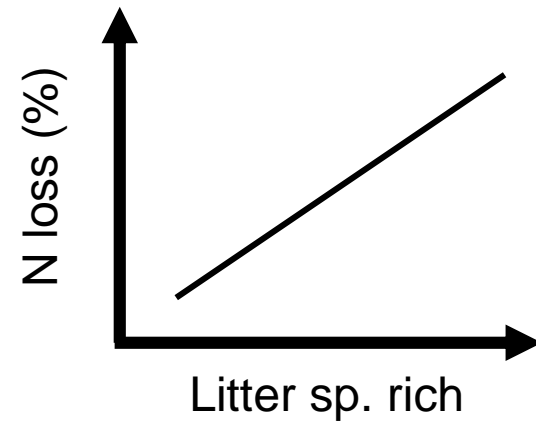
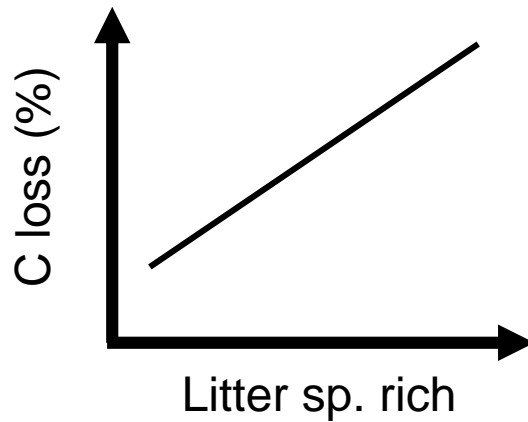
Example: tree diversity effect on litterfall and decomposition



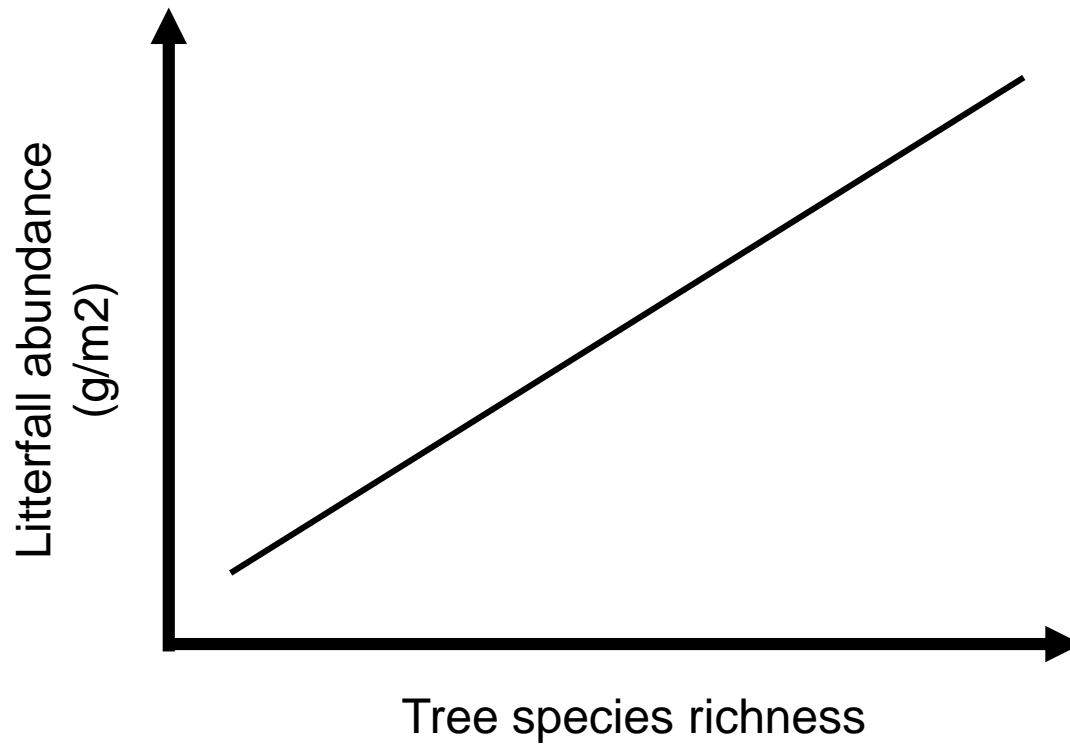
Example: tree diversity effect on litterfall and decomposition



Example: tree diversity effect on litterfall and decomposition



Example: tree diversity effect on litterfall abundance



Check your data structure

Check your
data
structure

1. What are your variables?
 - i. What is your response variable?
 - ii. What is your explanatory variable?
2. How are your data distributed?
3. How do you expect your response variable to be distributed?

Check your data structure

1. load your data in a dataset called df:

File type	R function [package]	Example
.csv	<code>read.csv(file = 'name.csv')</code>	<code>df = read.csv(file = "my-data.csv")</code>
.txt	<code>read.delim(file = 'name.txt')</code>	<code>df = read.txt(file = "my-data.txt")</code>
.xlsx	<code>read_xlsx(path = 'name.xlsx', sheet = "sheet.name")</code> [package: readxl]	<code>df = read_xlsx(path = "my-data.xlsx", sheet = "rawdata")</code>

Check your data structure

1. load your data in a dataset called df:

	TSP	litterfall	neigh.sp.rich
1	1-E34	73.98	1
2	10-G17	21.82	1
3	100-Q21	71.98	2
4	101-Q21	38.18	2
5	102-P26	66.06	2
6	103-P26	35.30	2
7	105-O6	22.71	3
8	11-O27	123.49	1
9	112-H31	147.98	2
10	113-H31	102.94	2
11	115-T17	292.50	2
12	116-I27	15.19	2
13	117-I27	22.12	3
14	118-I27	37.39	3
15	119-S18	91.97	2

Check your data structure

1. load your data in a dataset called df
2. what are your variables?

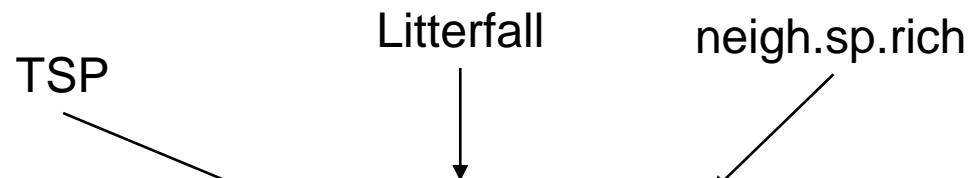


Diagram illustrating the variable labels and their corresponding columns in the dataset:

- TSP points to the first column (TSP).
- Litterfall points to the second column (litterfall).
- neigh.sp.rich points to the third column (neigh.sp.rich).

	TSP	litterfall	neigh.sp.rich
1	1-E34	73.98	1
2	10-G17	21.82	1
3	100-Q21	71.98	2
4	101-Q21	38.18	2
5	102-P26	66.06	2
6	103-P26	35.30	2
7	105-O6	22.71	3
8	11-O27	123.49	1
9	112-H31	147.98	2
10	113-H31	102.94	2
11	115-T17	292.50	2
12	116-I27	15.19	2
13	117-I27	22.12	3
14	118-I27	37.39	3
15	119-S18	91.97	2

Check your data structure

1. load your data in a dataset called df
2. what are your variables?

Variable name	Measure	Type	Expected range	Expected distribution
TSP	Sample name			
litterfall	Quantity of litter in gram fall on 1 m2			
neigh.sp.rich	Number of species in the surrounding			

Check your data structure

1. load your data in a dataset called df
2. what are your variables?

Variable name	Measure	Type	Expected range	Expected distribution
TSP	Sample name			
litterfall	Quantity of litter in gram fall on 1 m2			
neigh.sp.rich	Number of species in the surrounding			

str(df)

```
> str(df.fall)
'data.frame':  180 obs. of  3 variables:
 $ TSP      : chr  "1-E34" "10-G17" "100-Q21" "101-Q21" ...
 $ litterfall : num  74 21.8 72 38.2 66.1 ...
 $ neigh.sp.rich: int  1 1 2 2 2 2 3 1 2 2 ...
```

Check your data structure

1. load your data in a dataset called df
2. what are your variables?

Variable name	Measure	Type	Expected range	Expected distribution
TSP	Sample name	Character		
litterfall	Quantity of litter in gram fall on 1 m2	Numeric		
neigh.sp.rich	Number of species in the surrounding	Integer		

Check your data structure

1. load your data in a dataset called df
2. what are your variables?

Variable name	Measure	Type	Expected range	Expected distribution
TSP	Sample name	Character	All sample names	
litterfall	Quantity of litter in gram fall on 1 m2	Numeric	0 – 500 g/m2	
neigh.sp.rich	Number of species in the surrounding	Integer	[1;12]	

Check your data structure

1. load your data in a dataset called df
2. what are your variables?

Variable name	Measure	Type	Expected range	Expected distribution
TSP	Sample name	Character	All sample names	-
litterfall	Quantity of litter in gram fall on 1 m2	Numeric	0 – 500 g/m2	Normal
neigh.sp.rich	Number of species in the surrounding	Integer	[1;12]	-

Check your data structure

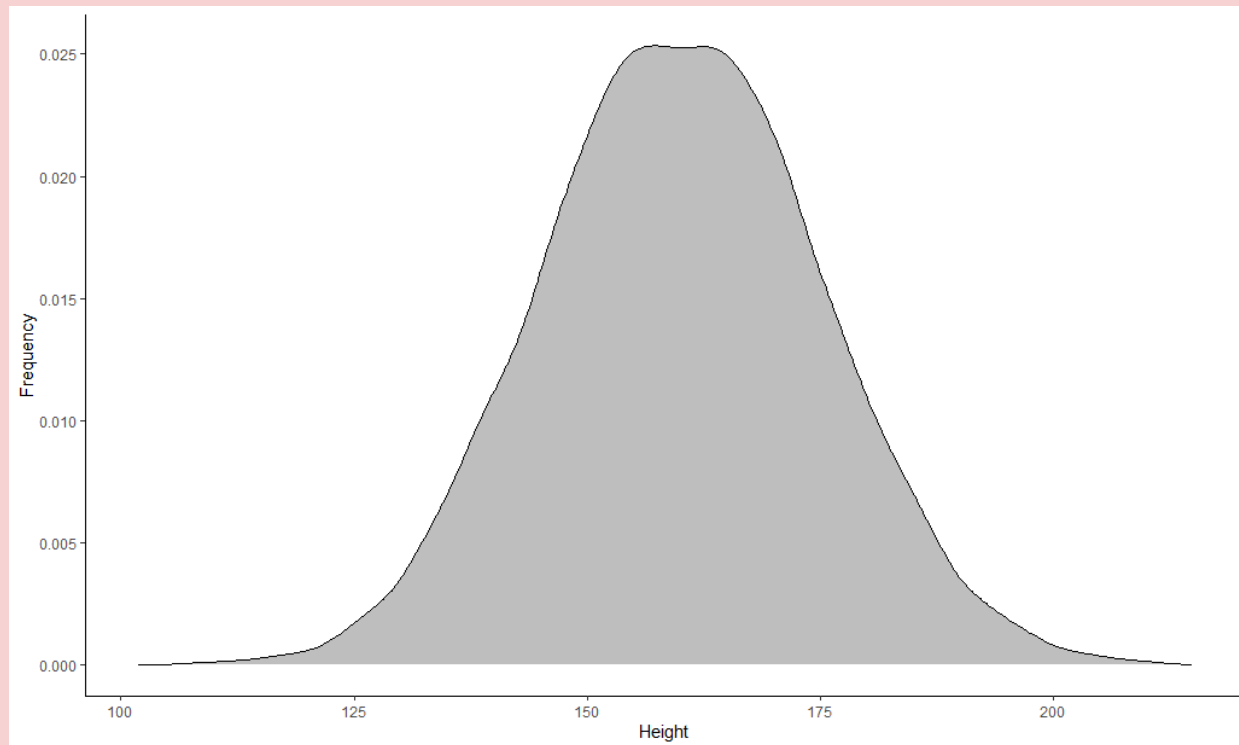
DANGER ZONE

Your data are not Normally distributed, your residuals should be!

Check your data structure

DANGER ZONE

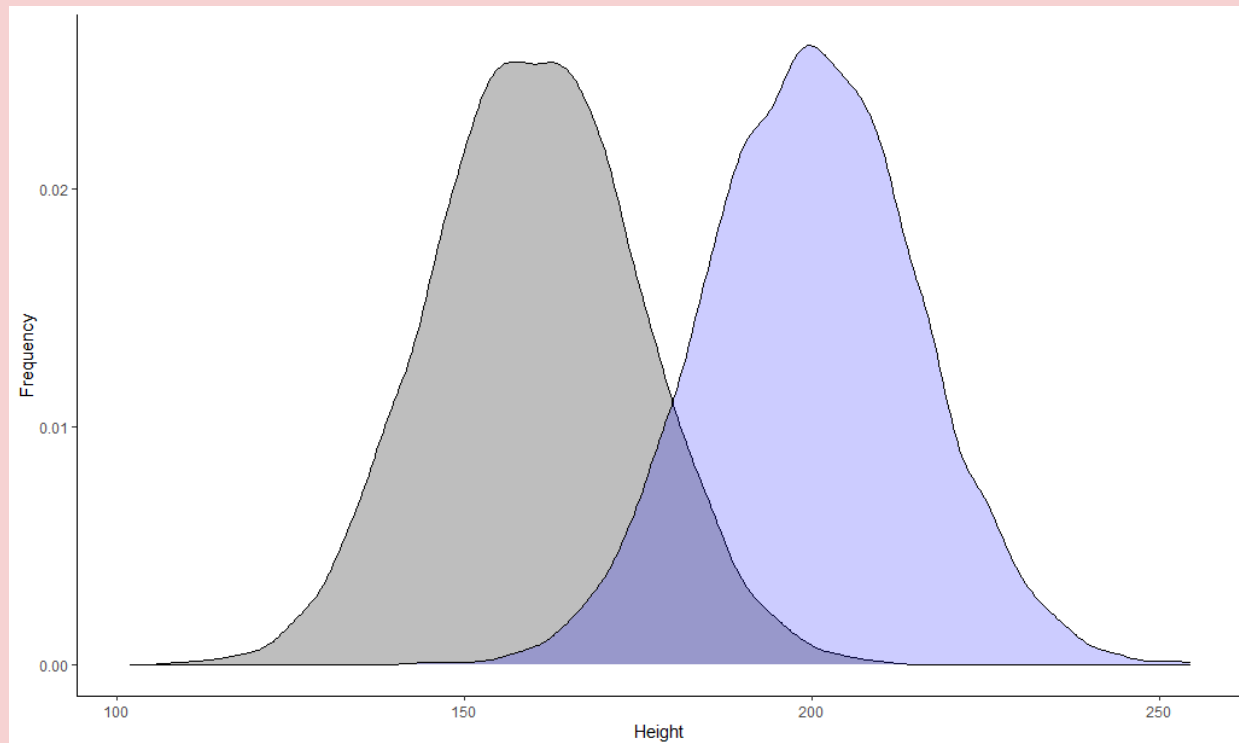
Your data are not Normally distributed, your residuals should be!
Let takes people height as example:



Check your data structure

DANGER ZONE

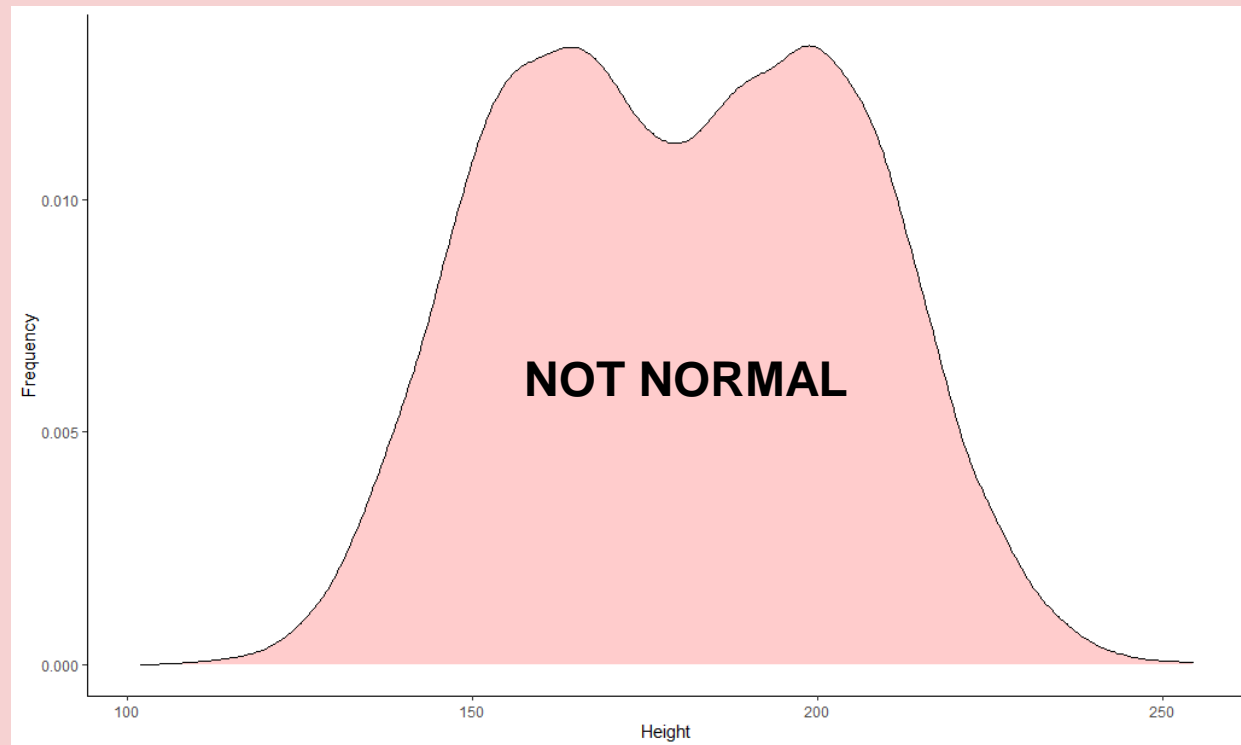
Your data are not Normally distributed, your residuals should be!
Let takes people height as example, **drinking your soup** makes you grow up



Check your data structure

DANGER ZONE

Your data are not Normally distributed, your residuals should be!
Let takes people height as example, drinking your soup makes you grow up



Check your data structure

DANGER ZONE

Height should follow a normal distribution
Therefore, your residuals should follow a normal distribution
Your population **DOES NOT** follow a normal distribution

(Same goes with other distribution types!)

Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Missing values

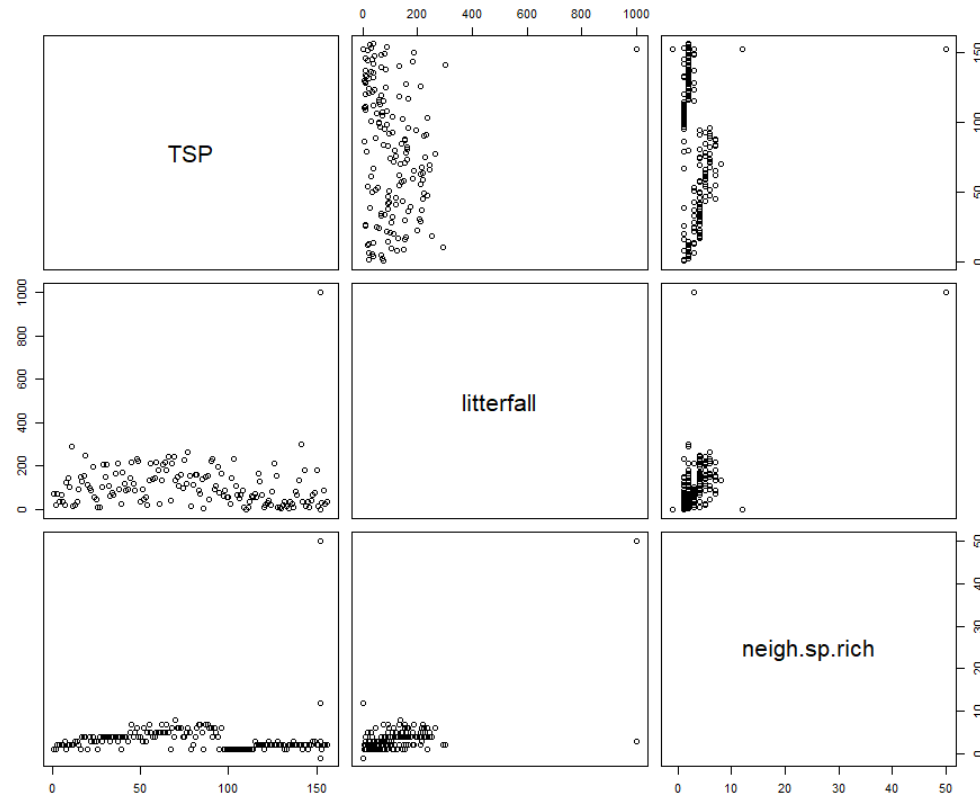
WARNING DANGER ZONE

Only keep complete rows:
`df = df[complete.cases(),]`

Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?

Quick and dirty
`plot(df)`

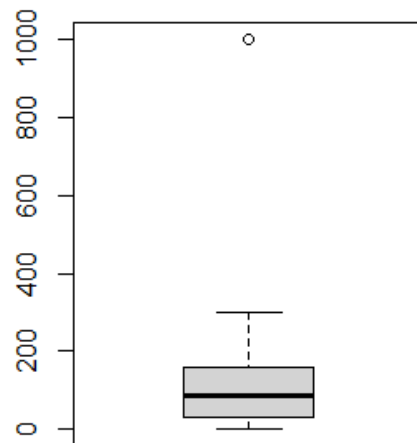


Check your data structure

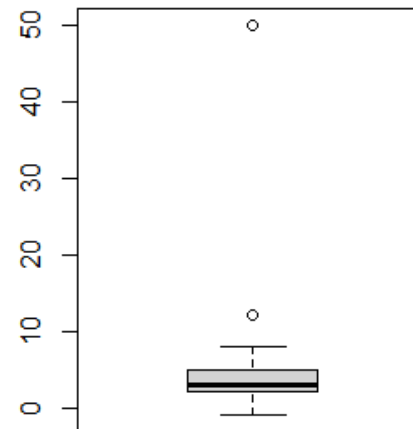
1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?

```
boxplot(df$litterfall)
```

Litterfall



neigh.sp.rich



Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Control data out of range:

```
df[df$litterfall<0 | df$litterfall>500,]
```

Conditions on rows

All columns

Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Control data out of range:

```
df[df$litterfall<0 | df$litterfall>500,]
```

	TSP	litterfall	neigh.sp.rich
170 outliers		1000	50
171 outliers		1000	50
172 outliers		1000	50
173 outliers		1000	50
174 outliers		1000	50
175 outliers		1000	3
176 outliers		1000	3
177 outliers		1000	3
178 outliers		1000	3
179 outliers		1000	3
180 outliers		1000	3

Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Control data out of range:

```
df[df$neigh.sp.rich<1 | df$neigh.sp.rich>12,]
```

	TSP	litterfall	neigh.sp.rich
165 outliers		1	-1
166 outliers		1	-1
167 outliers		1	-1
168 outliers		1	-1
169 outliers		1	-1
170 outliers		1000	50
171 outliers		1000	50
172 outliers		1000	50
173 outliers		1000	50
174 outliers		1000	50

Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Control data out of range
 2. Correct if typos or remove

Write the opposite conditional:

```
df[df$neigh.sp.rich>=1 & df$neigh.sp.rich<=12,]
```

Leave R to do it for you:

```
df[!(df$neigh.sp.rich<1 | df$neigh.sp.rich>12),]
```

Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Control data out of range
 2. Correct if typos or remove

WARNING DANGER ZONE

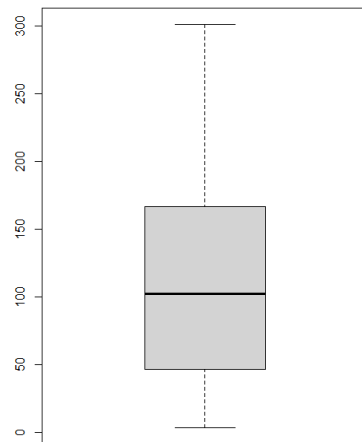
You will overwrite your data in r keep a safe copy
df.raw = df

```
df = df[!(df$neigh.sp.rich<1 | df$neigh.sp.rich>12),]  
df = df[!(df$litterfall<0 | df$litterfall>500),]
```

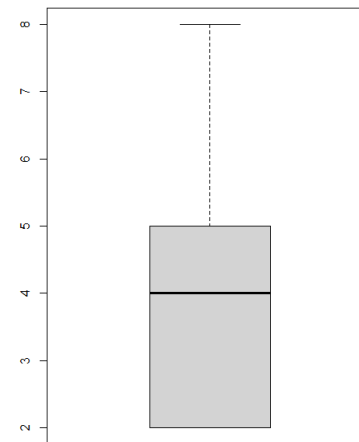
Check your data structure

1. load your data in a data called df
2. what are your variables?
3. how are your variables distributed?
 1. Control data out of range
 2. Correct if typos or remove

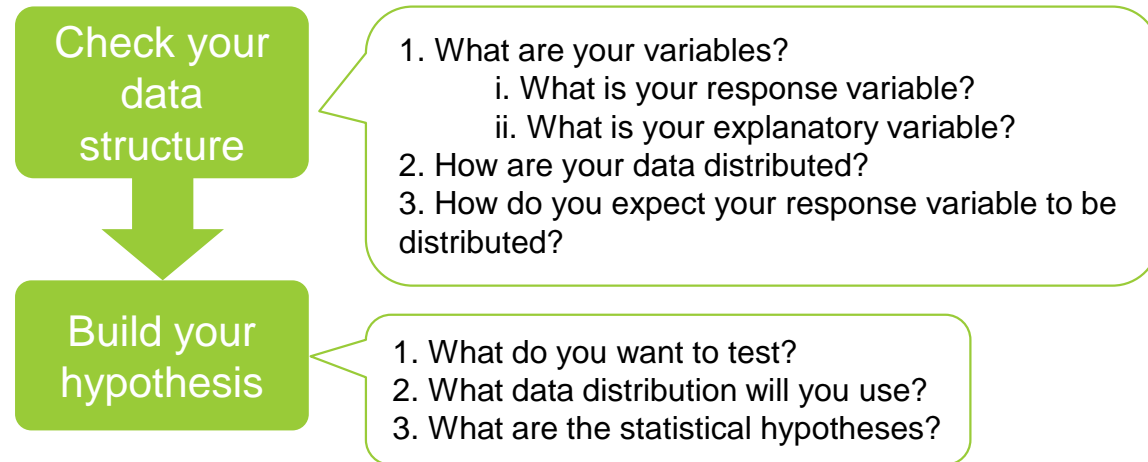
Litterfall



neigh.sp.rich

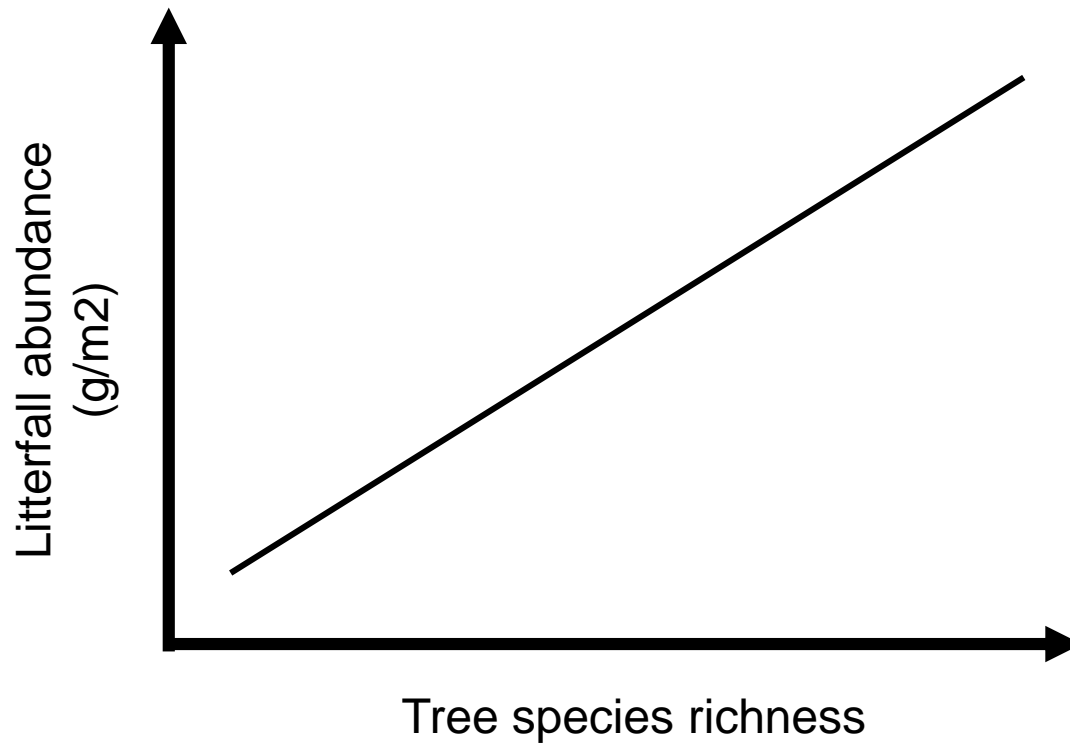


Build your hypothesis



Build your hypothesis

1. what do you want to test?



Build your hypothesis

1. what do you want to test?

Tree species richness increase litterfall

Build your hypothesis

1. what do you want to test?

Tree species richness increase litterfall

“litterfall” increase with “neigh.sp.rich”

Build your hypothesis

1. what do you want to test?

Tree species richness increase litterfall

“litterfall” increase with “neigh.sp.rich”

$$litterfall \sim \mu + \alpha \times neigh.sp.rich + \varepsilon$$

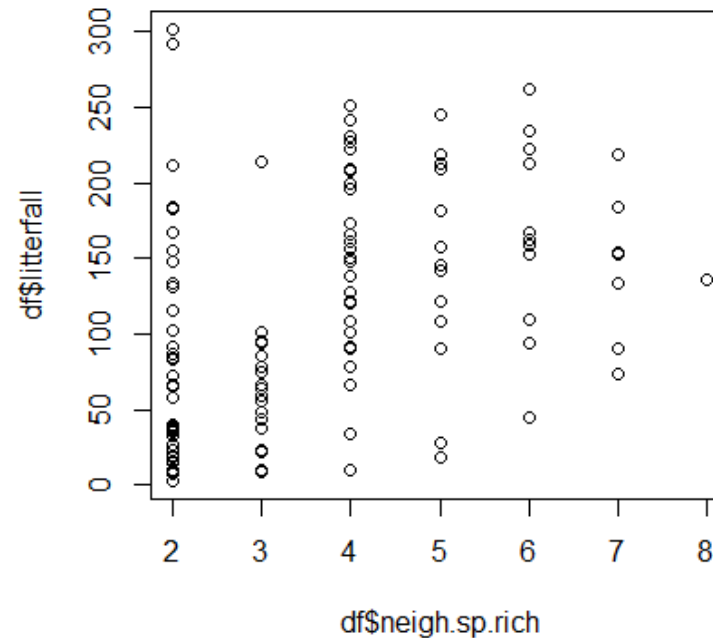
$$H_0: \alpha = 0, litterfall \sim \mu + \varepsilon$$

$$H_1: \alpha \neq 0, litterfall \sim \mu + \alpha \times neigh.sp.rich + \varepsilon$$

Build your hypothesis

1. what do you want to test?

take a look at your data: `plot(df$litterfall ~ df$neigh.sp.rich)`



Build your hypothesis

1. what do you want to test?
2. what distribution will you use? How do you expect your data to fall around your mean

$$litterfall \sim \mu + \alpha \times neigh.sp.rich + \varepsilon$$

Build your hypothesis

1. what do you want to test?
2. what distribution will you use? How do you expect your data to fall around your mean

$$litterfall \sim \mu + \alpha \times neigh.sp.rich + \varepsilon$$

$$\varepsilon \hookrightarrow N(0, \sigma)$$

Build your hypothesis

1. what do you want to test?
2. what distribution will you use?
3. what are your statistical hypotheses?

Build your hypothesis

1. what do you want to test?
2. what distribution will you use?
3. what are your statistical hypotheses?
 - i. Independence
 - ii. Random sampling
 - iii. Normally distributed error: $\varepsilon \hookrightarrow N(0, \sigma)$
 - iv. Equal variances (homoscedasticity)
 - v. Linearity
 - vi. Predictors are fixed

Build your hypothesis

1. what do you want to test?
2. what distribution will you use?
3. what are your statistical hypotheses?
most control by your experiment structure

i. Independence

ii. Random sampling

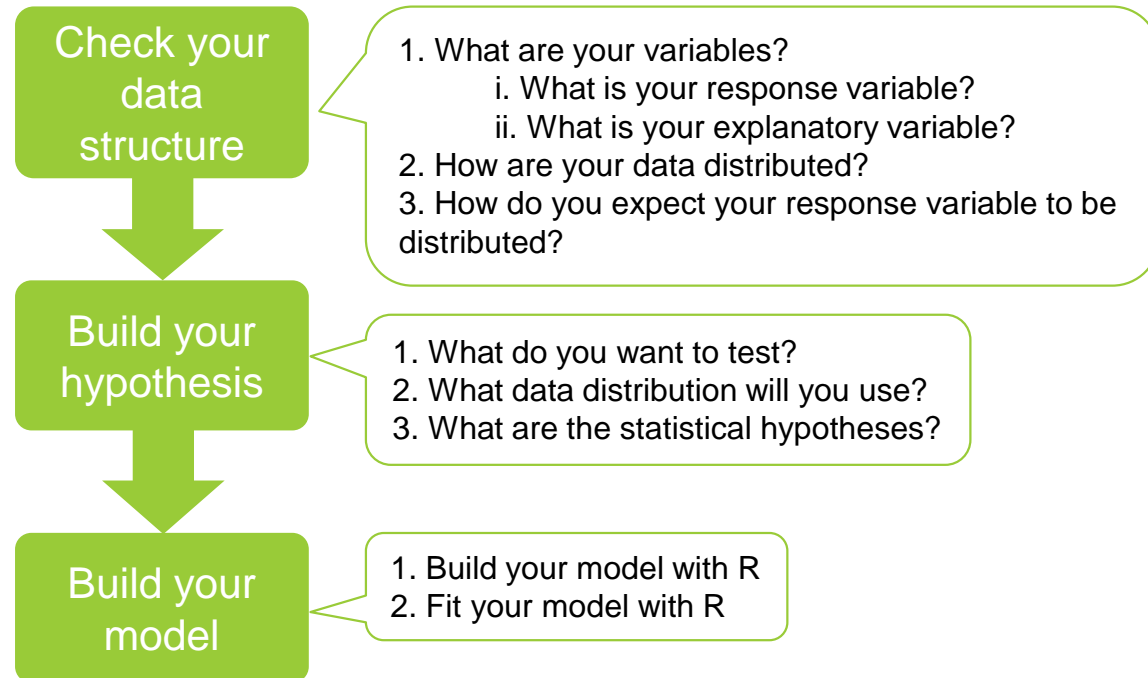
iii. Normally distributed error: $\varepsilon \hookrightarrow N(0, \sigma)$

iv. Equal variances (homoscedasticity)

v. Linearity

vi. Predictors are fixed

Build your model in R



Build your model in R

1. build your model

Build your model in R

1. build your model

Function: `lm()` (`glm()` for other residual distribution)

Build your model in R

1. build your model

Function: `lm()` (`glm()` for other residual distribution)

Formula: $y \sim x$

Build your model in R

1. build your model

Function: `lm()` (`glm()` for other residual distribution)

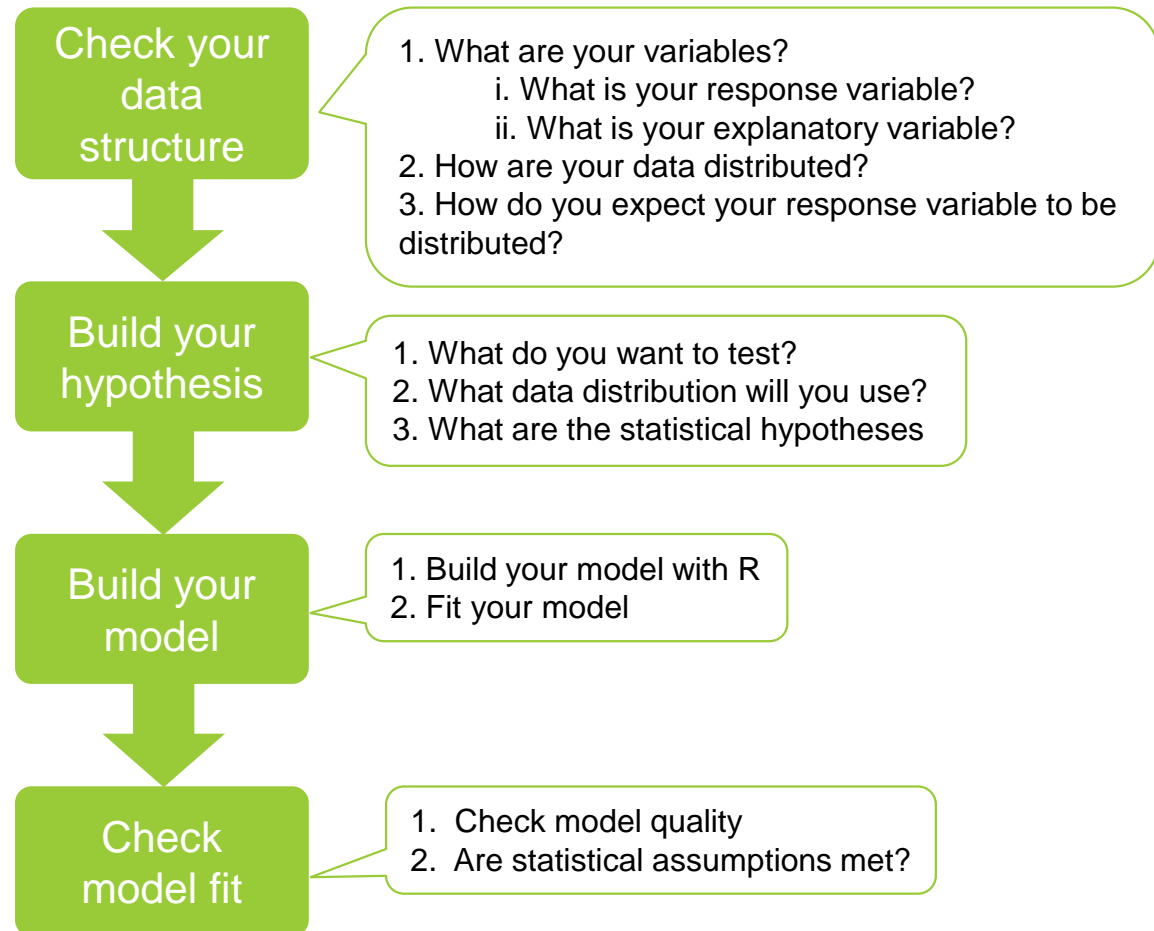
Formula: $y \sim x$

Together: `lm(formula = litterfall ~ neigh.sp.rich, data = df)`

2. fit the model to your data:

```
mod = lm(formula = litterfall ~ neigh.sp.rich, data = df)
```

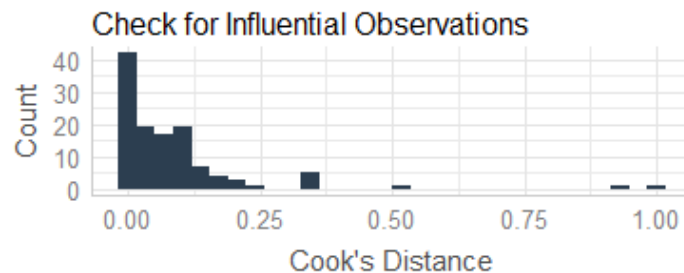
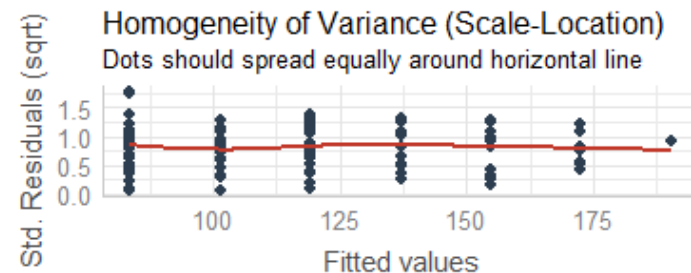
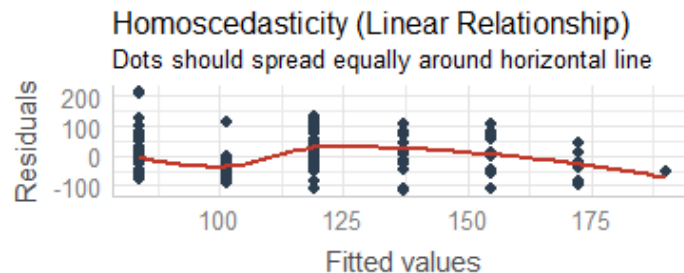
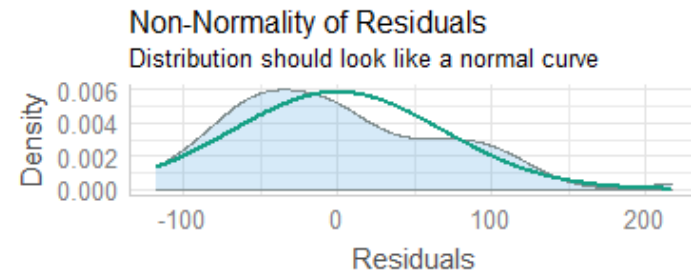
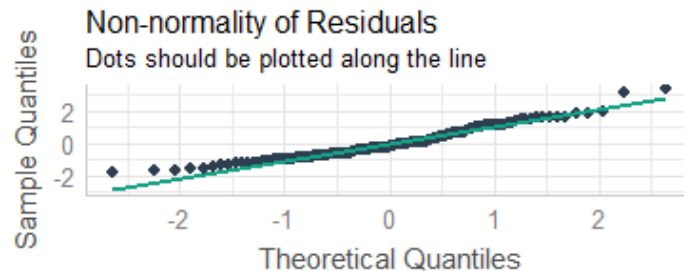
Check the model fit



Check the model fit

Check the model quality and the assumptions: the **performance** package

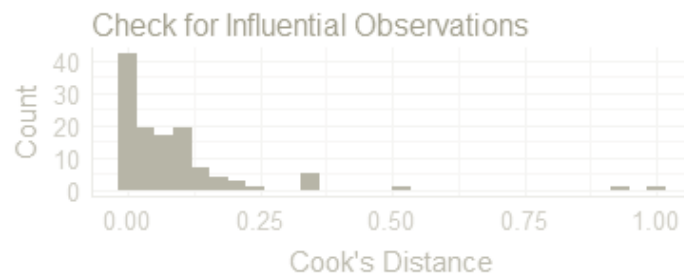
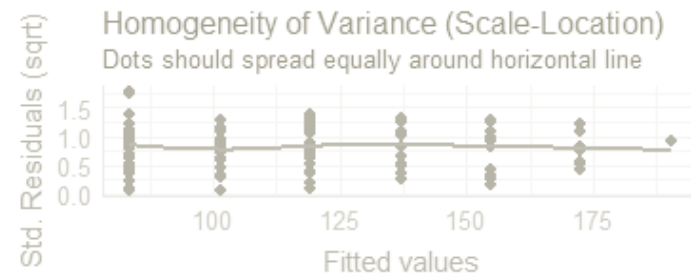
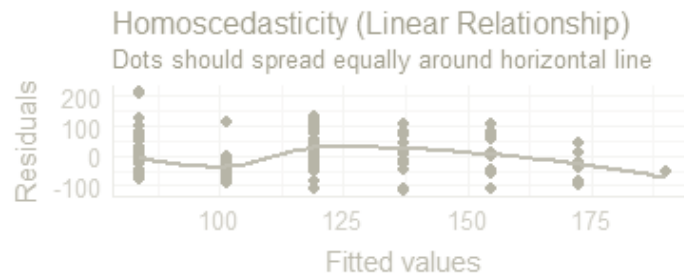
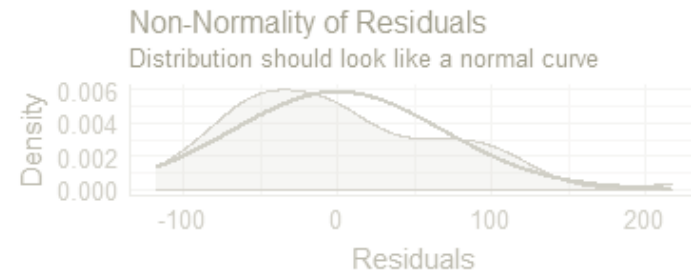
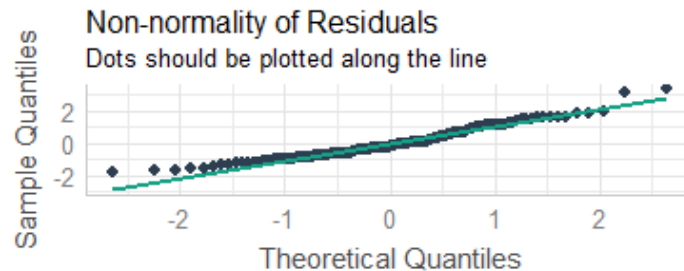
`check_model(mod)`



Check the model fit

Check the model quality and the assumptions: the **performance** package

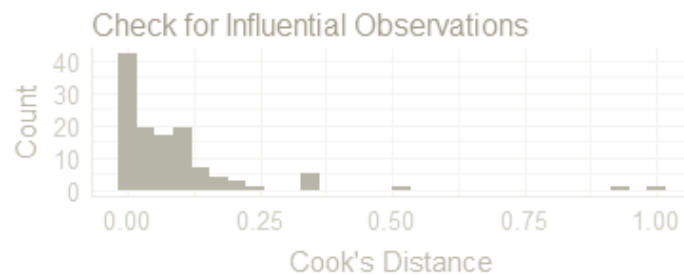
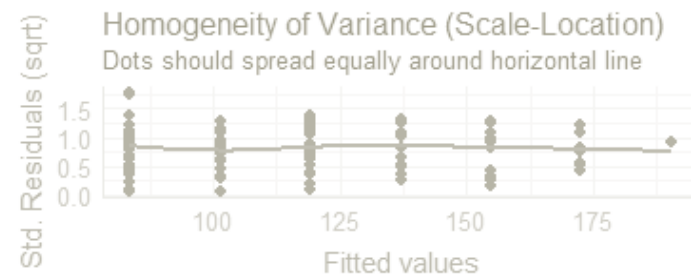
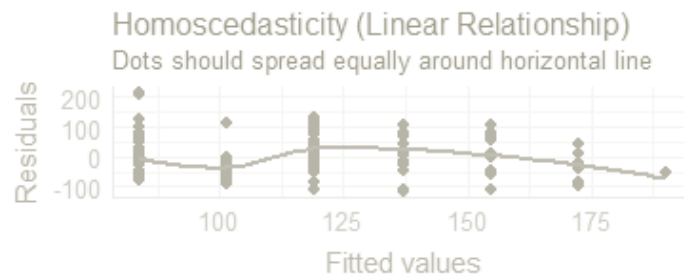
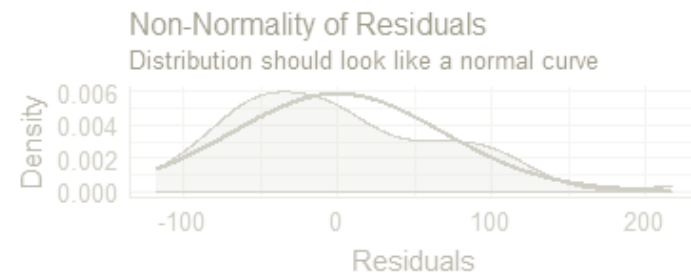
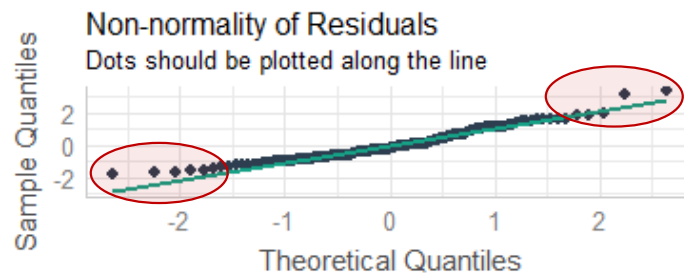
```
check_model(mod)
```



Check the model fit

Check the model quality and the assumptions: the **performance** package

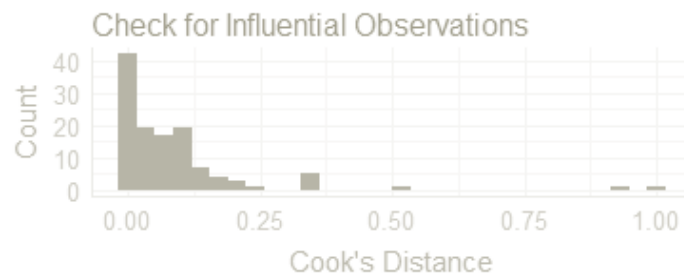
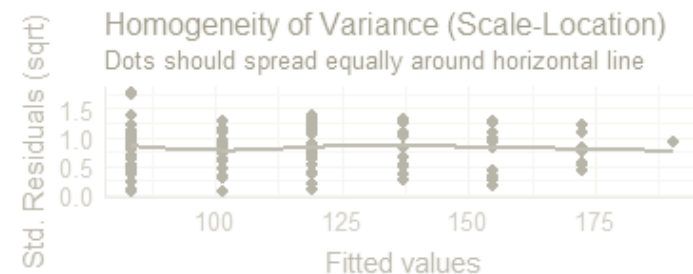
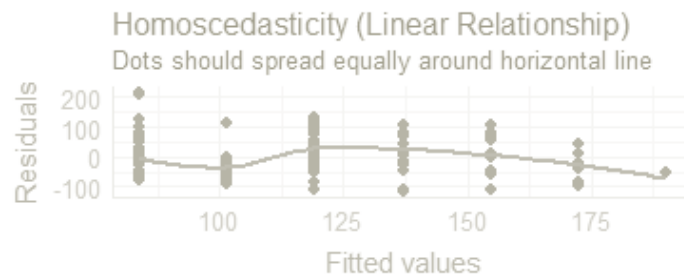
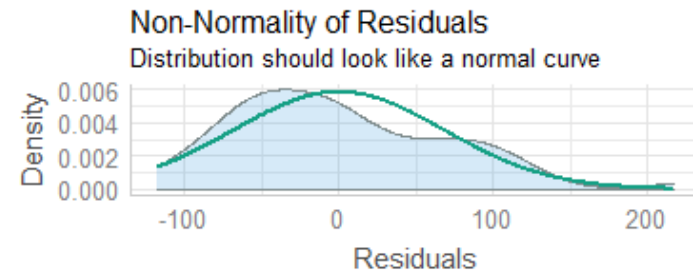
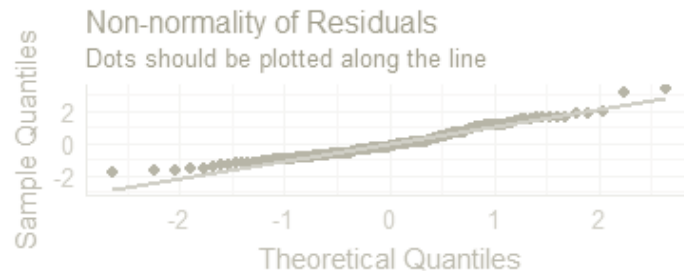
`check_model(mod)`



Check the model fit

Check the model quality and the assumptions: the **performance** package

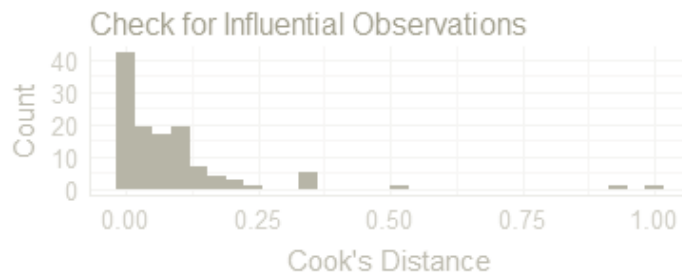
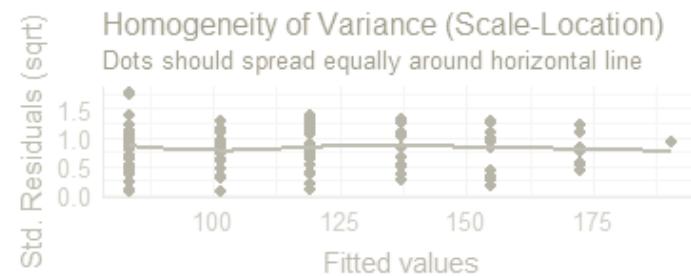
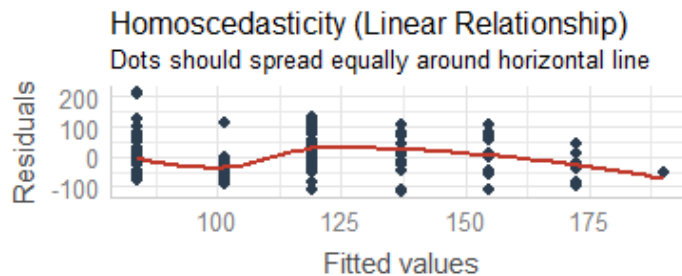
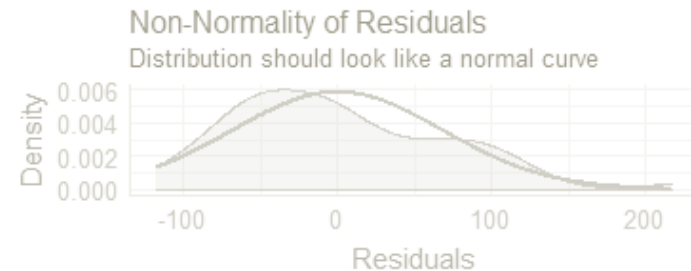
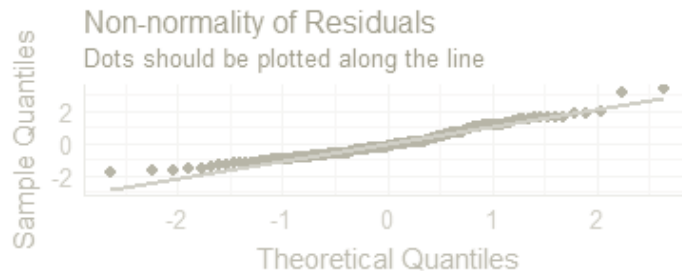
`check_model(mod)`



Check the model fit

Check the model quality and the assumptions: the **performance** package

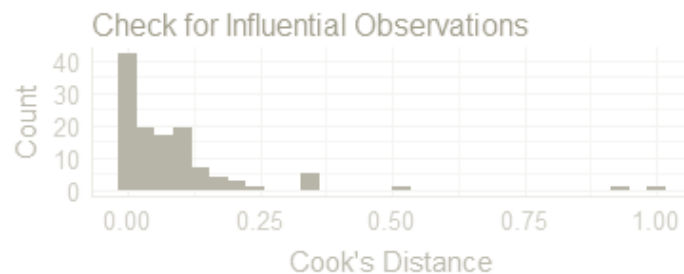
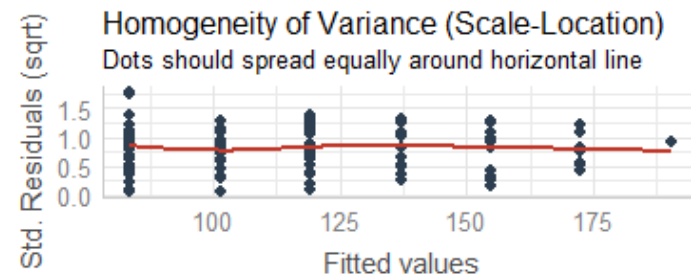
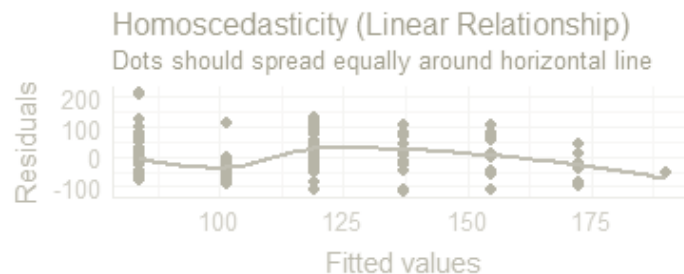
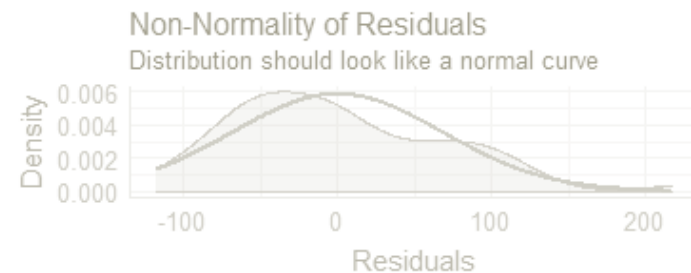
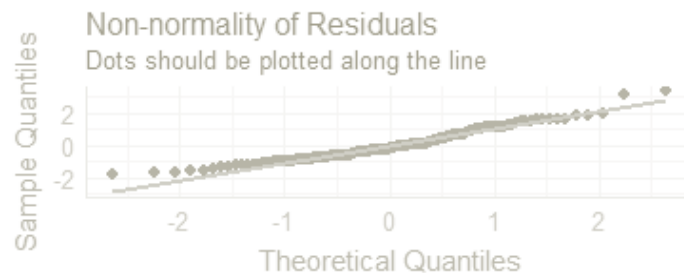
`check_model(mod)`



Check the model fit

Check the model quality and the assumptions: the **performance** package

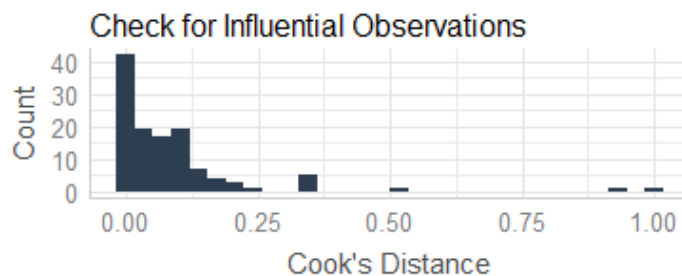
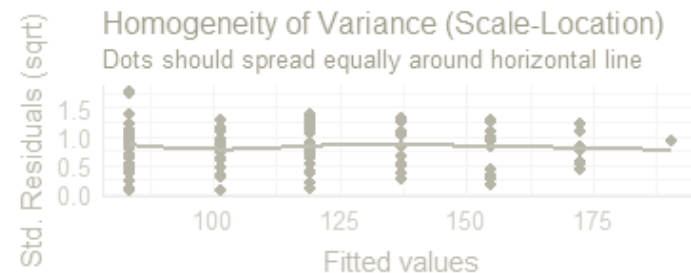
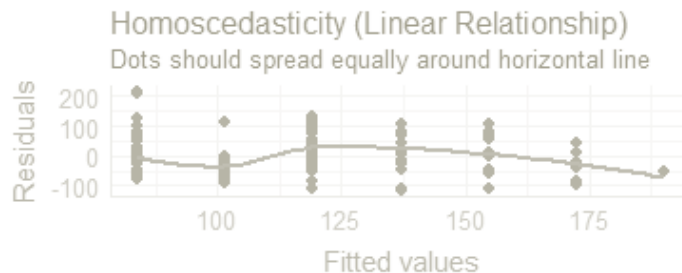
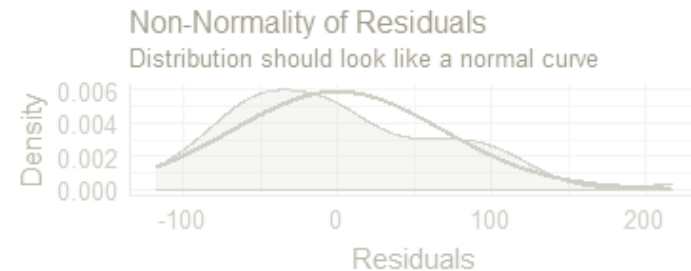
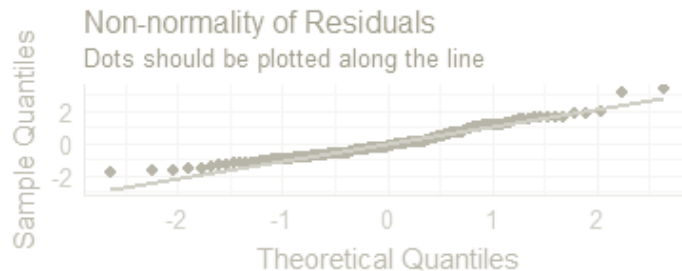
`check_model(mod)`



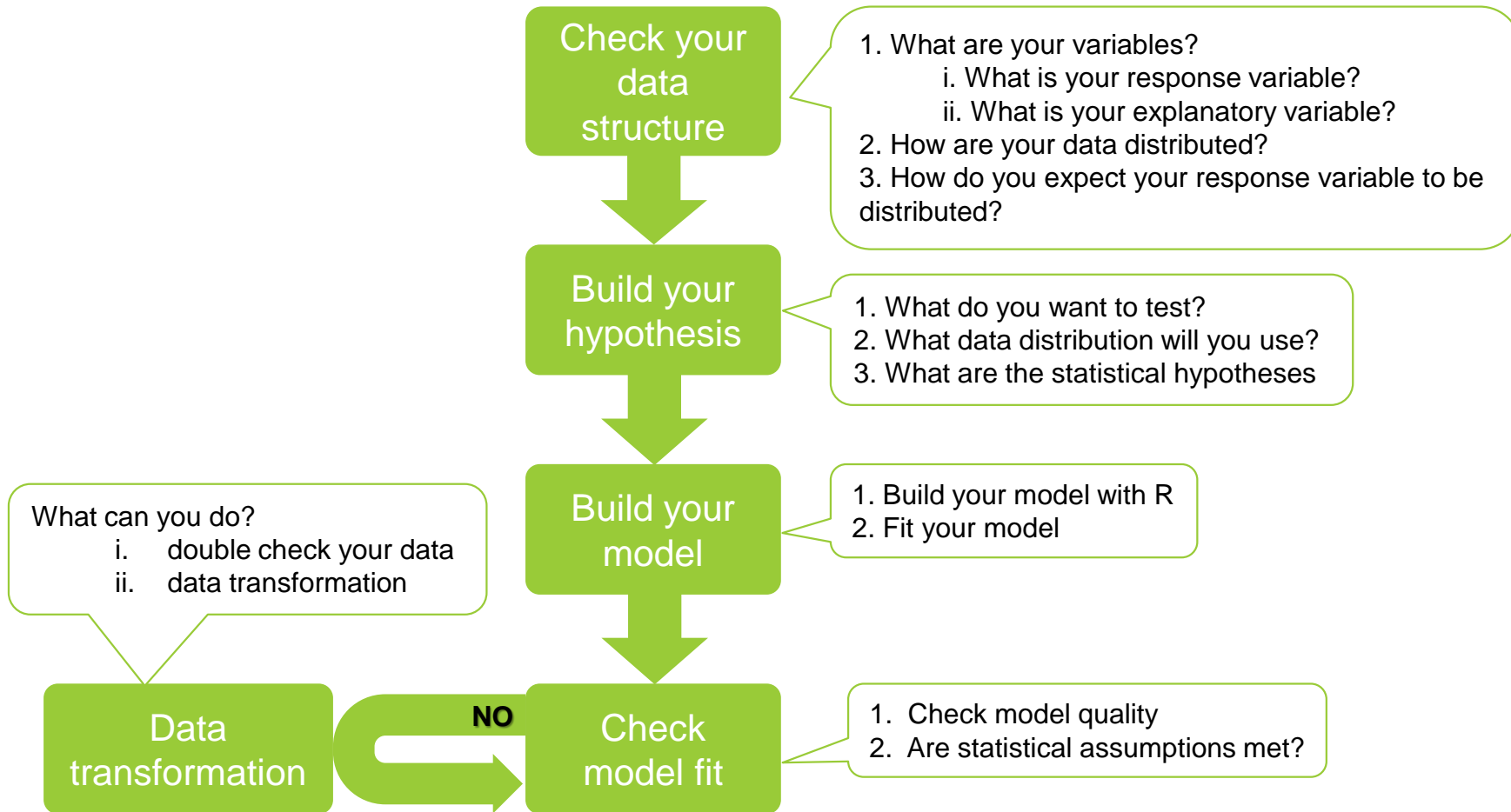
Check the model fit

Check the model quality and the assumptions: the **performance** package

`check_model(mod)`



Data transformation and outliers



Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: why?

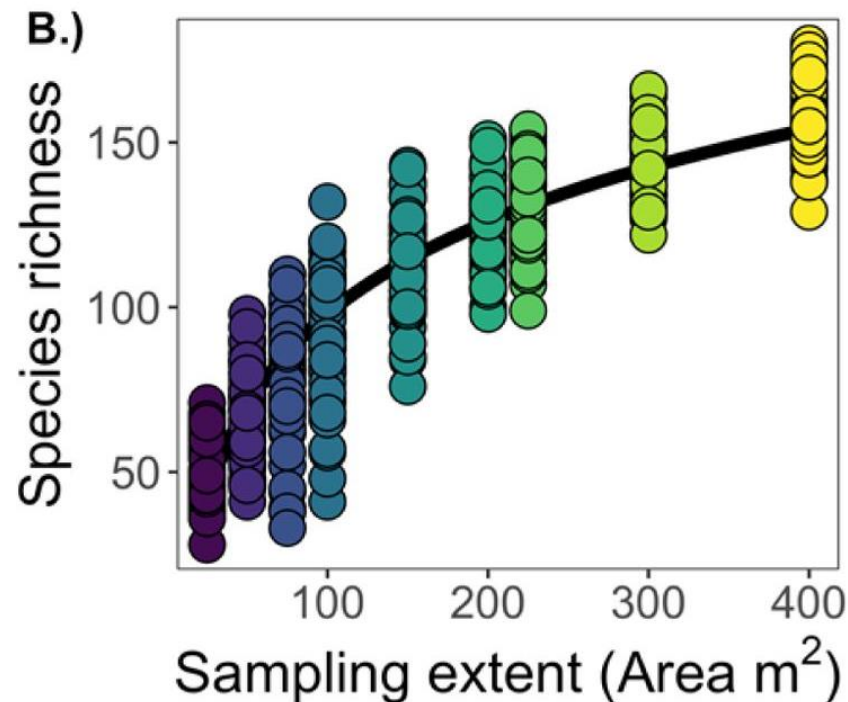
- to make linear non-linear things
- to make normal non-normal distribution

Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: why?

- to make linear non-linear things
- to make normal non-normal distribution

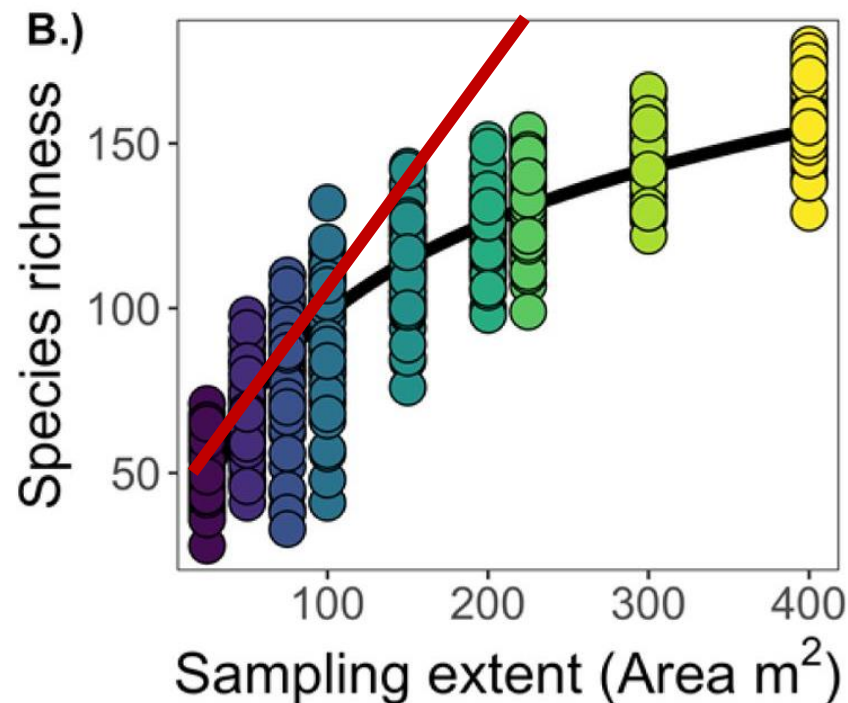


Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: why?

- to make linear non-linear things
- to make normal non-normal distribution

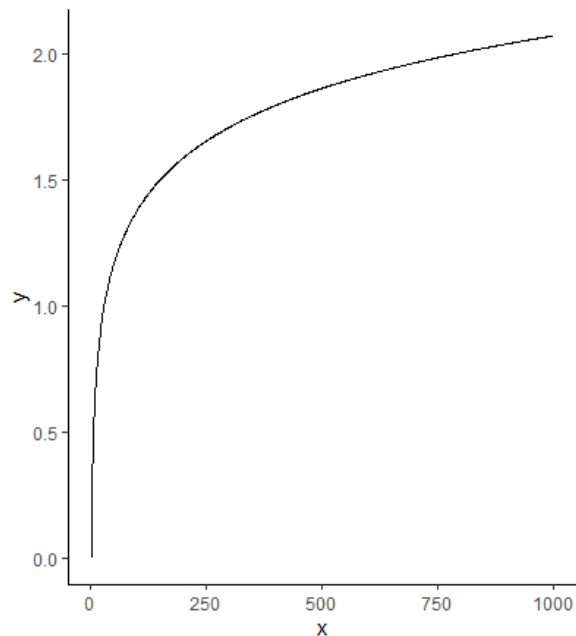


Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: why?

- to make linear non-linear things
- to make normal non-normal distribution

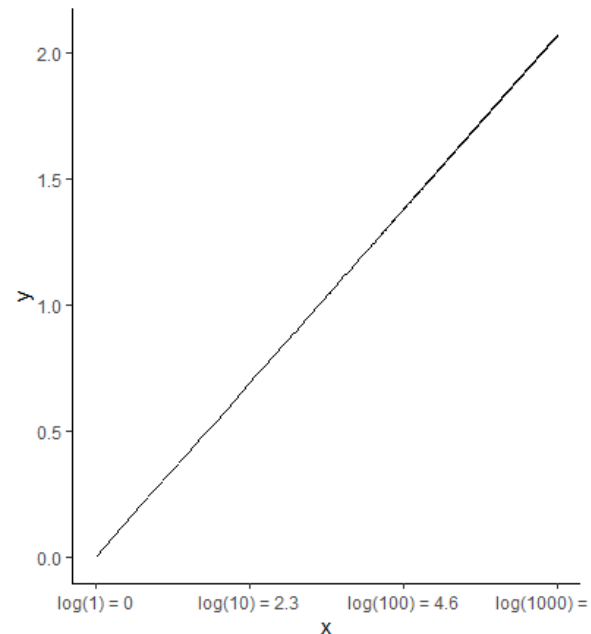
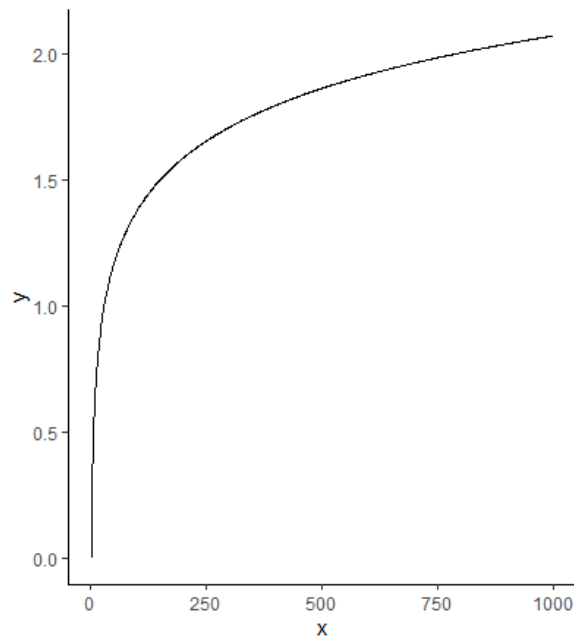


Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: why?

- to make linear non-linear things
- to make normal non-normal distribution

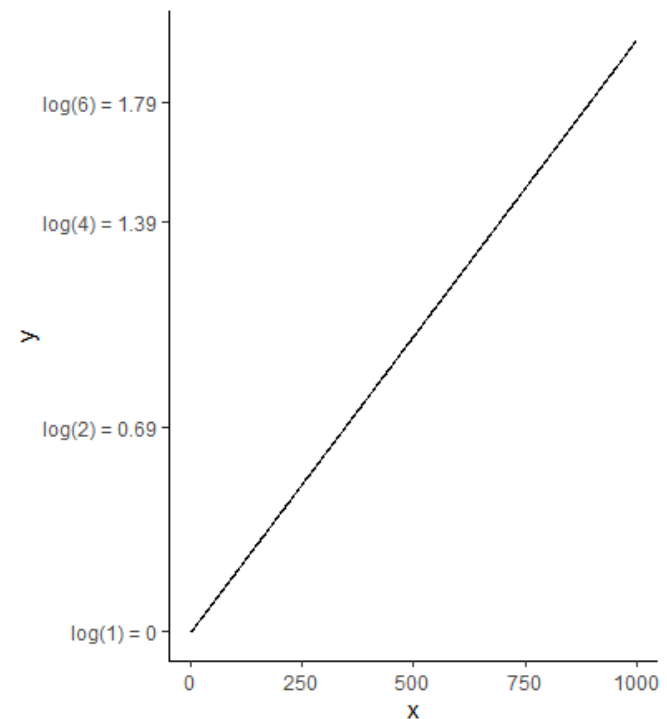
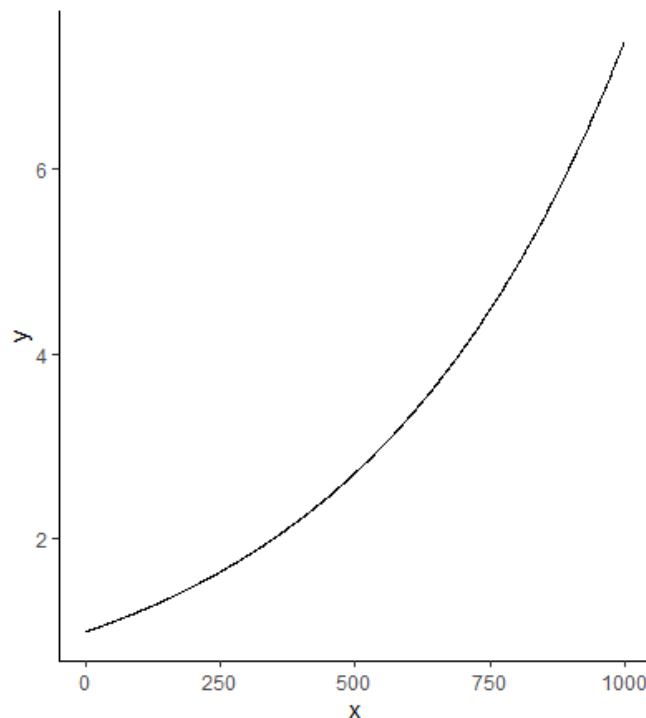


Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: why?

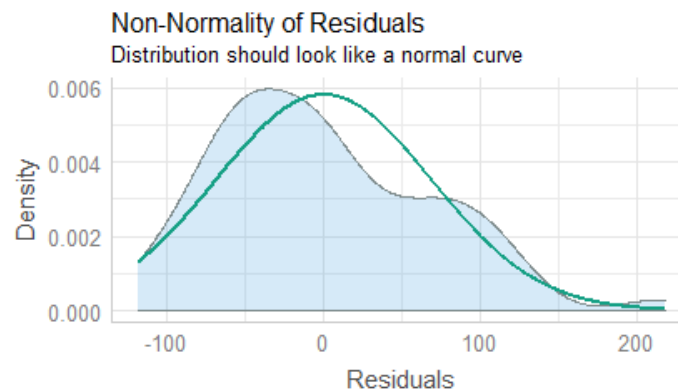
- to make linear non-linear things
- to make normal non-normal distribution



Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

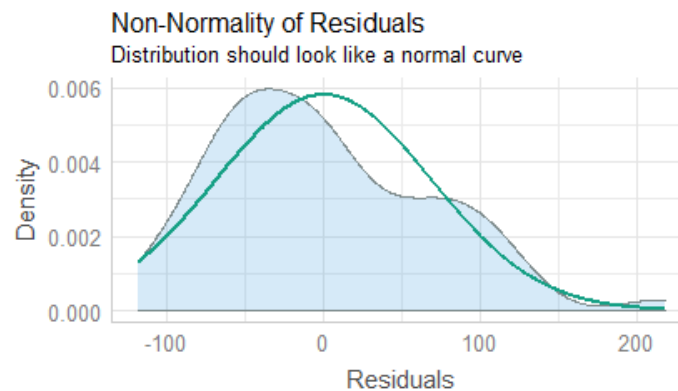
Data transformation:



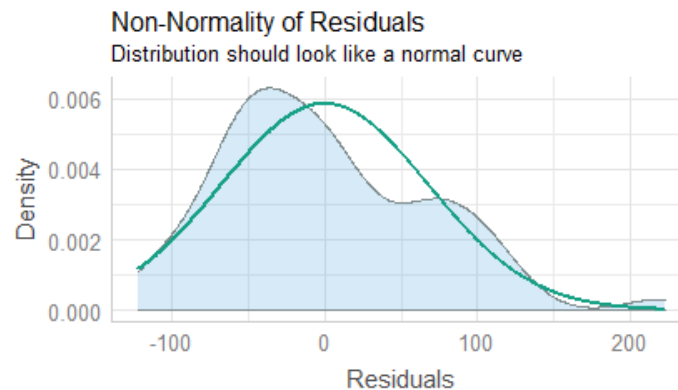
Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation:



log-transformation explanatory variable



Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: **log-transformation** explanatory variable

Compare the models quality: `compare_performance(mod, mod.log)`

```
# Comparison of Model Performance Indices
```

Name	Model	AIC	BIC	R2	R2 (adj.)	RMSE	Sigma
mod	lm	1360.645	1369.007	0.151	0.144	68.403	68.980
mod.2	lm	1358.819	1367.182	0.164	0.157	67.884	68.457

Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: **log-transformation** explanatory variable

Compare the models quality: `compare_performance(mod, mod.log)`

```
# Comparison of Model Performance Indices
```

Name	Model	AIC	BIC	R2	R2 (adj.)	RMSE	Sigma
mod	lm	1360.645	1369.007	0.151	0.144	68.403	68.980
mod.2	lm	1358.819	1367.182	0.164	0.157	67.884	68.457

AIC: fit quality –
weighted by the
number of variables

BIC: fit quality –
weighted by the
number of variables
and the sample size

Sigma: residual
standard error

RMSE: Mean Root
Standard Error –
standard error of the
residuals

R: fit quality – part of
variance explained

Data transformation and outliers

Check outliers with performance: `check_outliers(mod)`

Data transformation: **log-transformation** explanatory variable

Compare the models quality: `compare_performance(mod, mod.log)`

```
# Comparison of Model Performance Indices
```

Name	Model	AIC	BIC	R2	R2 (adj.)	RMSE	Sigma
mod	lm	1360.645	1369.007	0.151	0.144	68.403	68.980
mod.2	lm	1358.819	1367.182	0.164	0.157	67.884	68.457

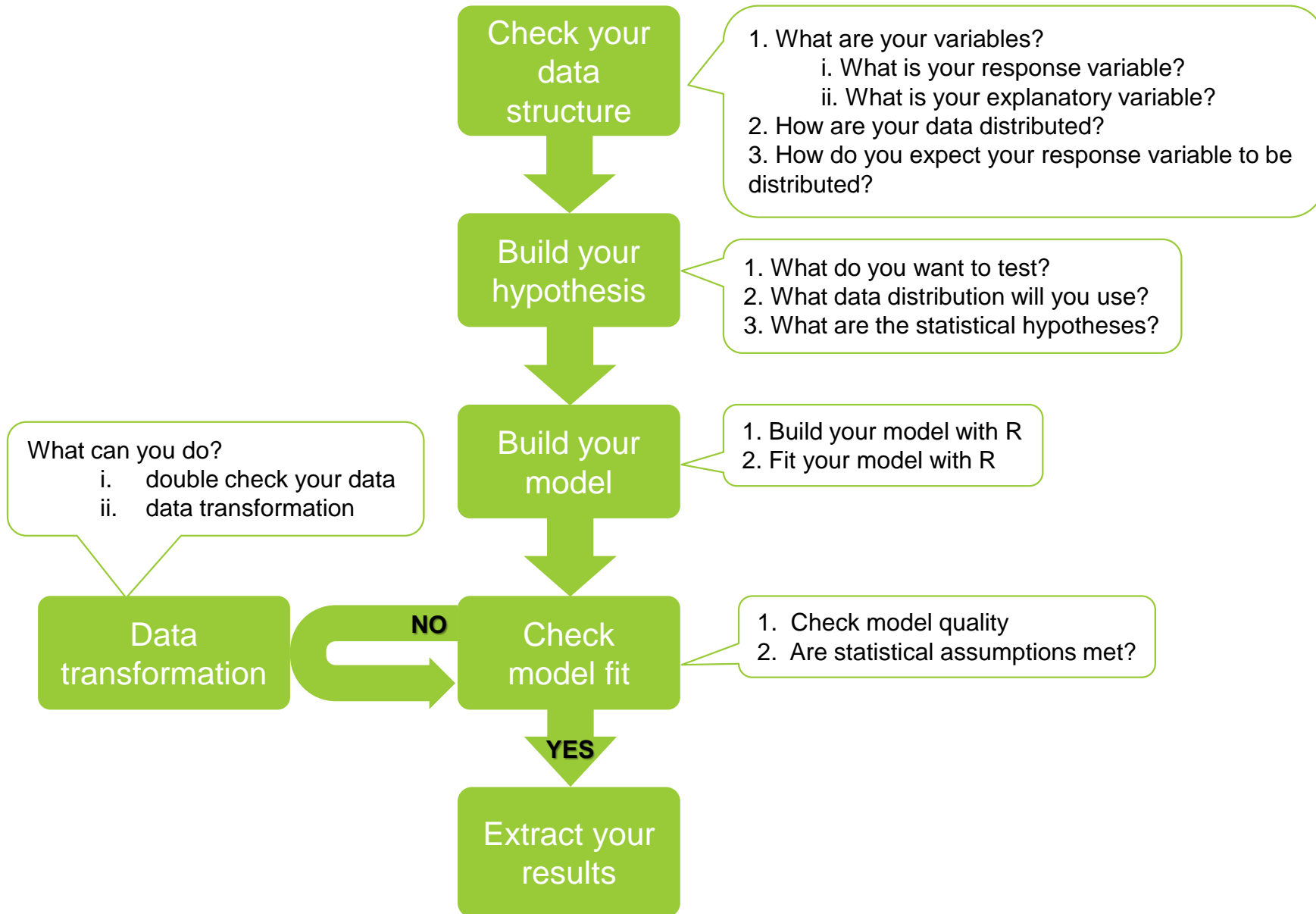
AIC: fit quality –
weighted by the
number of variables

$\Delta AIC > [2,8]$: the model are different

Can be completed by ANOVA (see following lecture)

R: fit quality – part of
variance explained

Extract your results



Extract your results

summary(mod)

$$\text{litterfall} \sim \mu + \alpha \times \log(\text{neigh.sp.rich}) + \varepsilon$$

```
> summary(mod.2)

Call:
lm(formula = "litterfall ~ log(neigh.sp.rich)", data = df.fall)

Residuals:
    Min       1Q   Median       3Q      Max
-118.83  -47.15  -13.37   38.85  213.10

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    50.852     9.339   5.445 2.02e-07 ***
log(neigh.sp.rich) 53.960     8.147   6.624 5.61e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 65.01 on 153 degrees of freedom
Multiple R-squared:  0.2228,    Adjusted R-squared:  0.2178 
F-statistic: 43.87 on 1 and 153 DF,  p-value: 5.614e-10
```

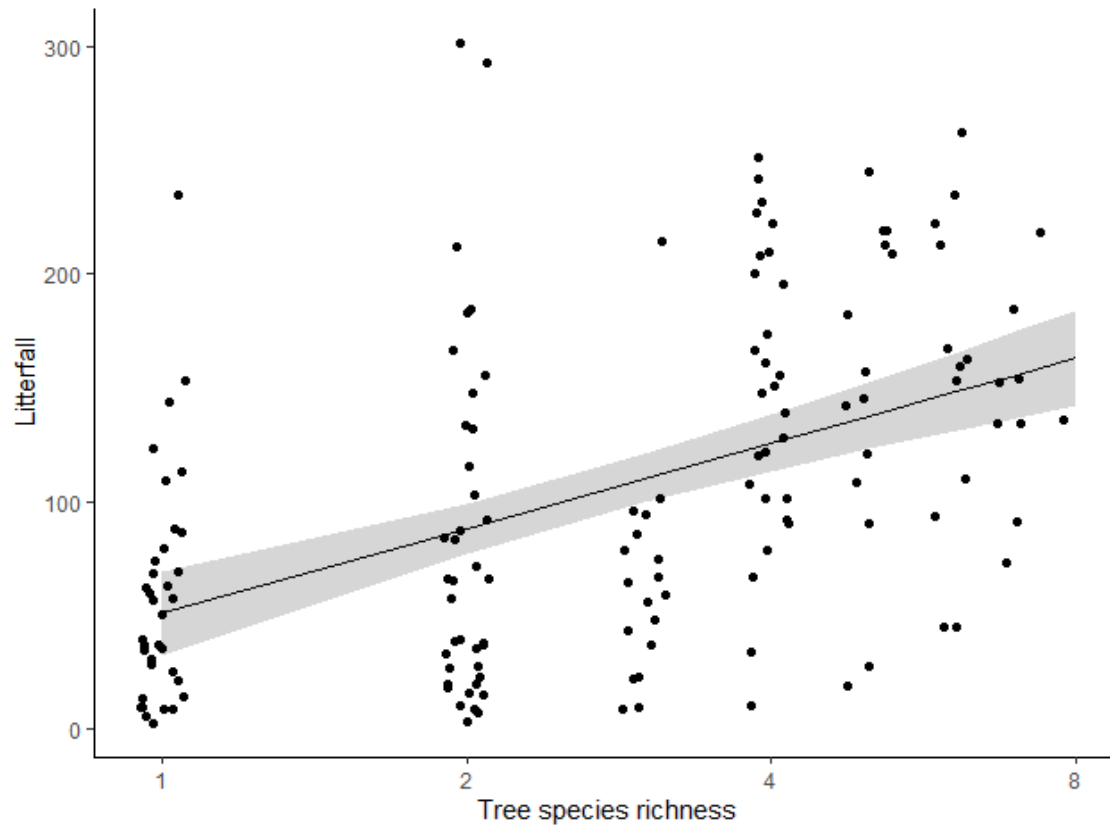
μ

α

Mean litterfall when diversity null = 50.852 +/- 18.304 g/m² (Estimate +/- 1.96 x SE)
Effect species richness = 53.960 +/- 15.958 g/m²/log(#species)

Extract your results

summary(mod)



Mean litterfall when diversity null = 50.852 ± 18.304 g/m² (Estimate $\pm 1.96 \times$ SE)
Effect species richness = 53.960 ± 15.958 g/m²/log(#species)

Extract your results

```
summary(mod)
```

DANGER ZONE: the factors



A



B



C



D

```
lm(formula = litterfall ~ species, data = df)
```

Extract your results

summary(mod)

DANGER ZONE: the factors



A



B



C



D

`lm(formula = litterfall ~ species, data = df)`

$litterfall \sim \alpha_A \times specie_A + \alpha_B \times specie_B + \alpha_C \times specie_C + \alpha_D \times specie_D + \varepsilon$

$specie_i$ is 0 or 1

Extract your results

```
summary(mod)
```

DANGER ZONE: the factors

```
Call:
lm(formula = "litterfall ~ specie", data = d.2)

Residuals:
    Min       1Q   Median       3Q      Max
-137.34  -46.82  -10.53   31.08  218.85

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    58.75      10.95   5.367 5.27e-07 ***
specieB        23.76      15.08   1.576  0.118
specieC        88.83      16.42   5.410 4.38e-07 ***
specieD        76.77      65.68   1.169  0.245
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 64.76 on 99 degrees of freedom
Multiple R-squared:  0.2392,    Adjusted R-squared:  0.2162
F-statistic: 10.38 on 3 and 99 DF,  p-value: 5.335e-06
```

$$litterfall \sim \alpha_A \times specie_A + \alpha_B \times specie_B + \alpha_C \times specie_C + \alpha_D \times specie_D + \varepsilon$$

Extract your results

summary(mod)

DANGER ZONE: the factors

Call: lm(formula = "litterfall ~ specie", data = d.2)					
Residuals:					
Min	1Q	Median	3Q	Max	
-137.34	-46.82	-10.53	31.08	218.85	
Coefficients:					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	58.75	10.95	5.367	5.27e-07 ***	
specieB	23.76	15.08	1.576	0.118	
specieC	88.83	16.42	5.410	4.38e-07 ***	
specieD	76.77	65.68	1.169	0.245	
--- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Residual standard error: 64.76 on 99 degrees of freedom Multiple R-squared: 0.2392, Adjusted R-squared: 0.2162 F-statistic: 10.38 on 3 and 99 DF, p-value: 5.335e-06					

$$\text{litterfall} \sim \alpha_A \times \text{specie}_A + \alpha_B \times \text{specie}_B + \alpha_C \times \text{specie}_C + \alpha_D \times \text{specie}_D + \varepsilon$$

If you like to test the differences between the different factors you need to do an ANOVA and a Tukey test

Extract your results

```
summary(mod)
```

DANGER ZONE: the factors

If you like to test the differences between the different factors you need to do an ANOVA and a Tukey test

```
mod = lm(formula = litterfall ~ species, data = df)
mod.aov = aov(mod)
TukeyHSD(mod.aov)
```

```
Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = .)

$specie
      diff      lwr      upr      p adj
B-A 23.76082 -15.64056  63.1622 0.3970308
C-A 88.83229  45.92617 131.7384 0.0000026
D-A 76.76800 -94.85645 248.3924 0.6477987
C-B 65.07147  23.15468 106.9883 0.0005684
D-B 53.00718 -118.37262 224.3870 0.8504196
D-C -12.06429 -184.28362 160.1551 0.9978085
```

Extract your results

summary(mod)

$$\text{litterfall} \sim \mu + \alpha \times \log(\text{neigh.sp.rich}) + \varepsilon$$

```
> summary(mod.2)

Call:
lm(formula = "litterfall ~ log(neigh.sp.rich)", data = df.fall)

Residuals:
    Min       1Q   Median       3Q      Max
-118.83  -47.15  -13.37   38.85  213.10

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    50.852     9.339   5.445 2.02e-07 ***
log(neigh.sp.rich) 53.960     8.147   6.624 5.61e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 65.01 on 153 degrees of freedom
Multiple R-squared:  0.2228,    Adjusted R-squared:  0.2178
F-statistic: 43.87 on 1 and 153 DF,  p-value: 5.614e-10
```

Extract your results

```
summary(mod)
```

Extract the coefficients: `summary(mod)$coefficients`

```
> summary(mod.2)$coefficients
              Estimate Std. Error t value    Pr(>|t|)
(Intercept)   50.85248    9.338548  5.445437 2.017066e-07
log(neigh.sp.rich) 53.95982    8.146519  6.623666 5.613835e-10
```

To extract the predictions from your models: `ggeffect` package
`pred = ggpredict(model = mod, terms = 'neigh.sp.rich')`

```
# Predicted values of litterfall
# x = neigh.sp.rich
```

x	Predicted	95% CI
1	50.85	[32.55, 69.16]
2	88.25	[77.23, 99.28]
3	110.13	[99.63, 120.64]
4	125.66	[113.28, 138.03]
5	137.70	[123.02, 152.38]
6	147.54	[130.65, 164.42]
7	155.85	[136.95, 174.76]
8	163.06	[142.33, 183.79]

In this lecture:

- 1. The stepwise process to analyses your data**
- 2. Application on an example with R**
- 3. Practical on your own**

Your time to play

