# Statistical inference and linear models

December 5, 2018

# General approach

1. Scientific question

# General approach

1. Scientific question

↓

2. Model and Statistical question

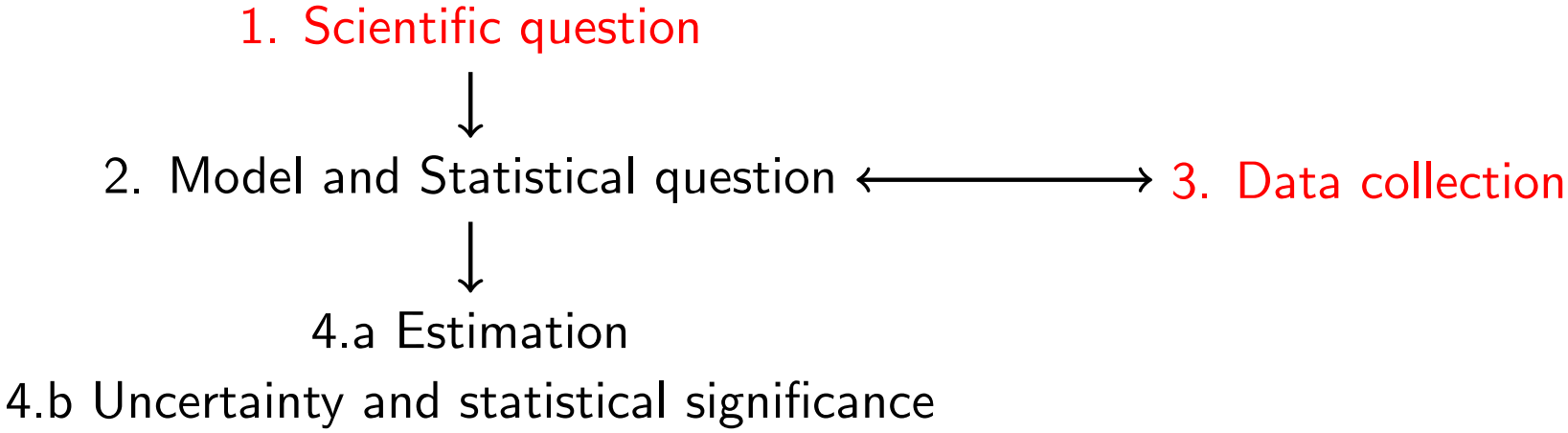
# General approach

1. Scientific question

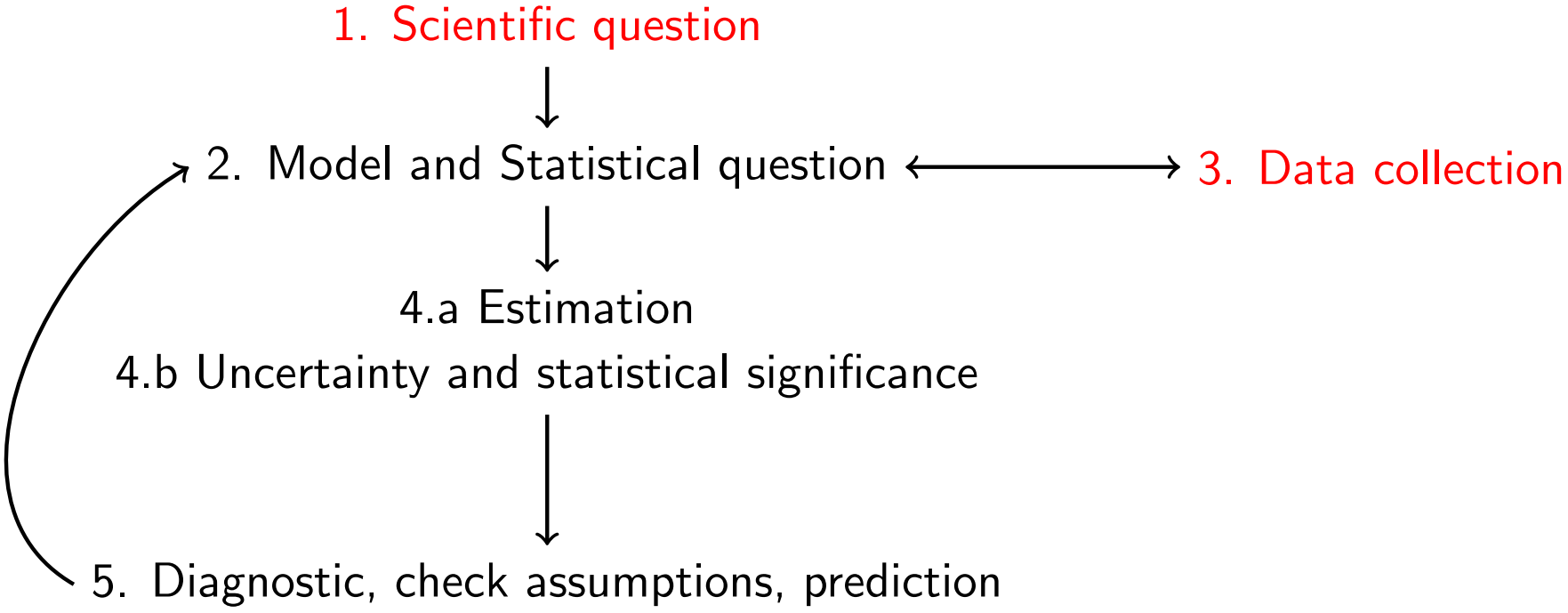
↓

2. Model and Statistical question ⟷ 3. Data collection

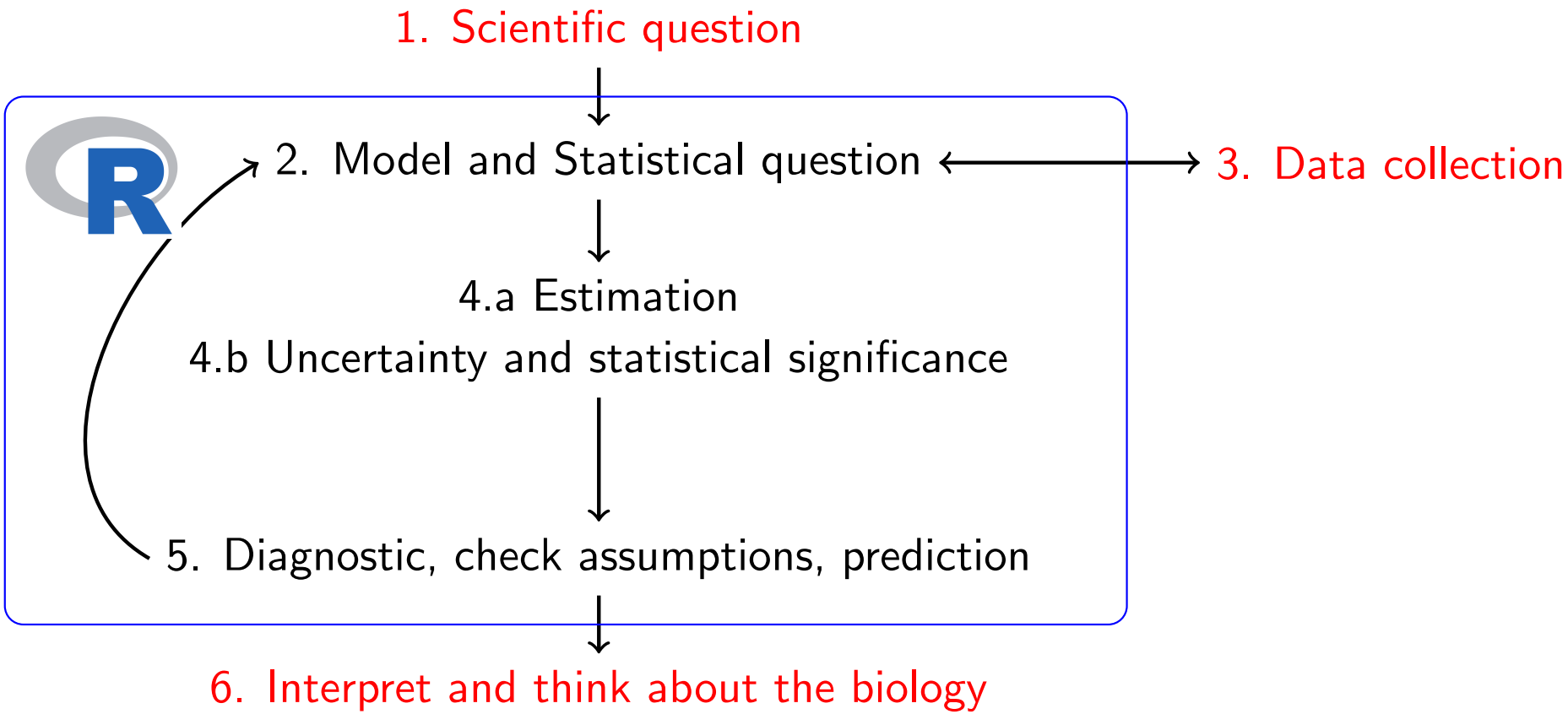
# General approach



# General approach



# General approach



# Iris t.test

# Iris t.test

```
data("iris")
```

```
str(iris)
plot(iris)
```

1. Scientific question: Are the taxa "setosa" and "versicolor" different species?

# Iris t.test

```
data("iris")
```

```
str(iris)
plot(iris)
```

1. Scientific question: Are the taxa "setosa" and "versicolor" different species?
2. Model and stat question:
   - Model:
     - ⋆ There is an intrinsic/expected sepal length value for a species; an individual value is the sum of this expectation and a random Gaussian deviation.
     - ⋆ $y_i = \mu_{species_i} + \epsilon_i$ with $\epsilon \sim N(0, \sigma^2)$
     - ⋆ t-test

# Iris t.test

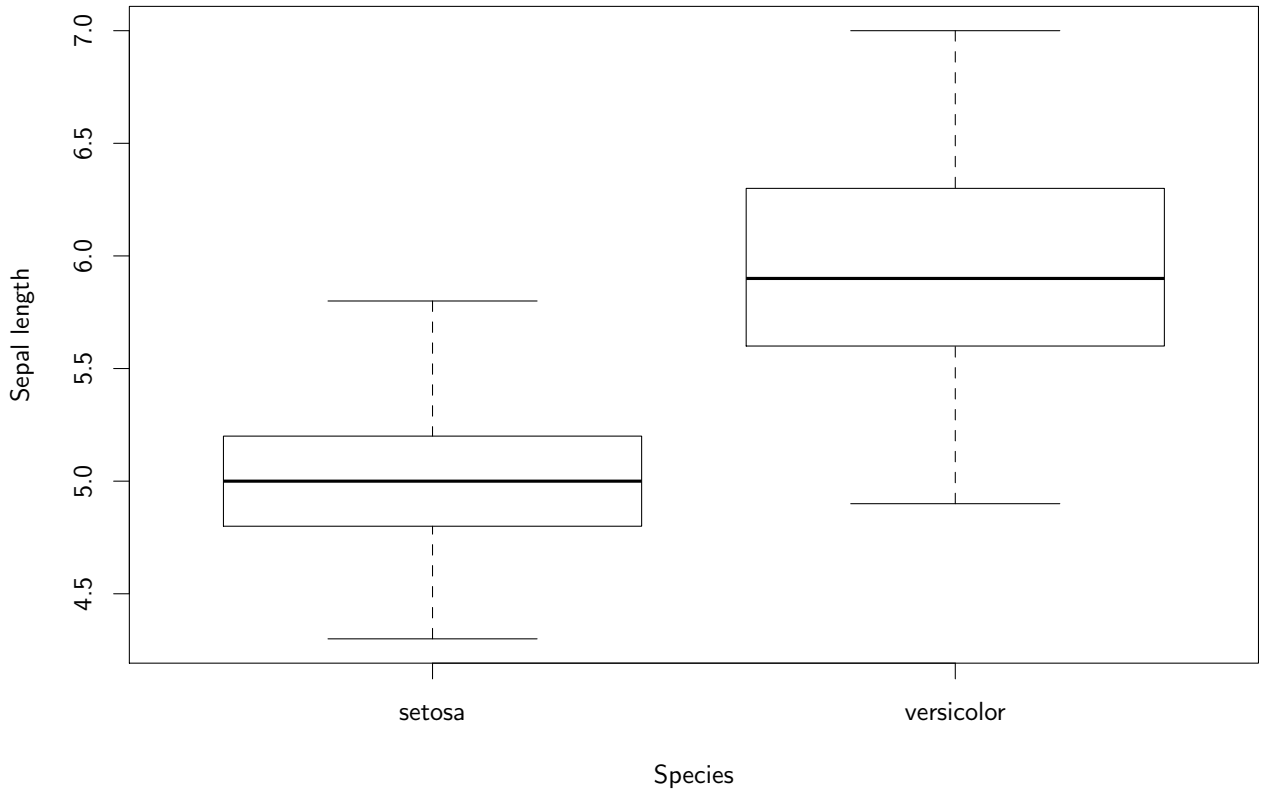
```
data("iris")
```

```
str(iris)
plot(iris)
```

1. Scientific question: Are the taxa "setosa" and "versicolor" different species?
2. Model and stat question:
   - Model:
     - ⋆ There is an intrinsic/expected sepal length value for a species; an individual value is the sum of this expectation and a random Gaussian deviation.
     - ⋆ $y_i = \mu_{species_i} + \epsilon_i$ with $\epsilon \sim N(0, \sigma^2)$
     - ⋆ t-test
   - Statistical question:
     - ⋆ Does sepal length **differ significantly** between the two taxa **in our sample**?
     - ⋆ Is the observed difference between taxa likely if both taxa have the same intrinsic/expected value?
3. Data collection

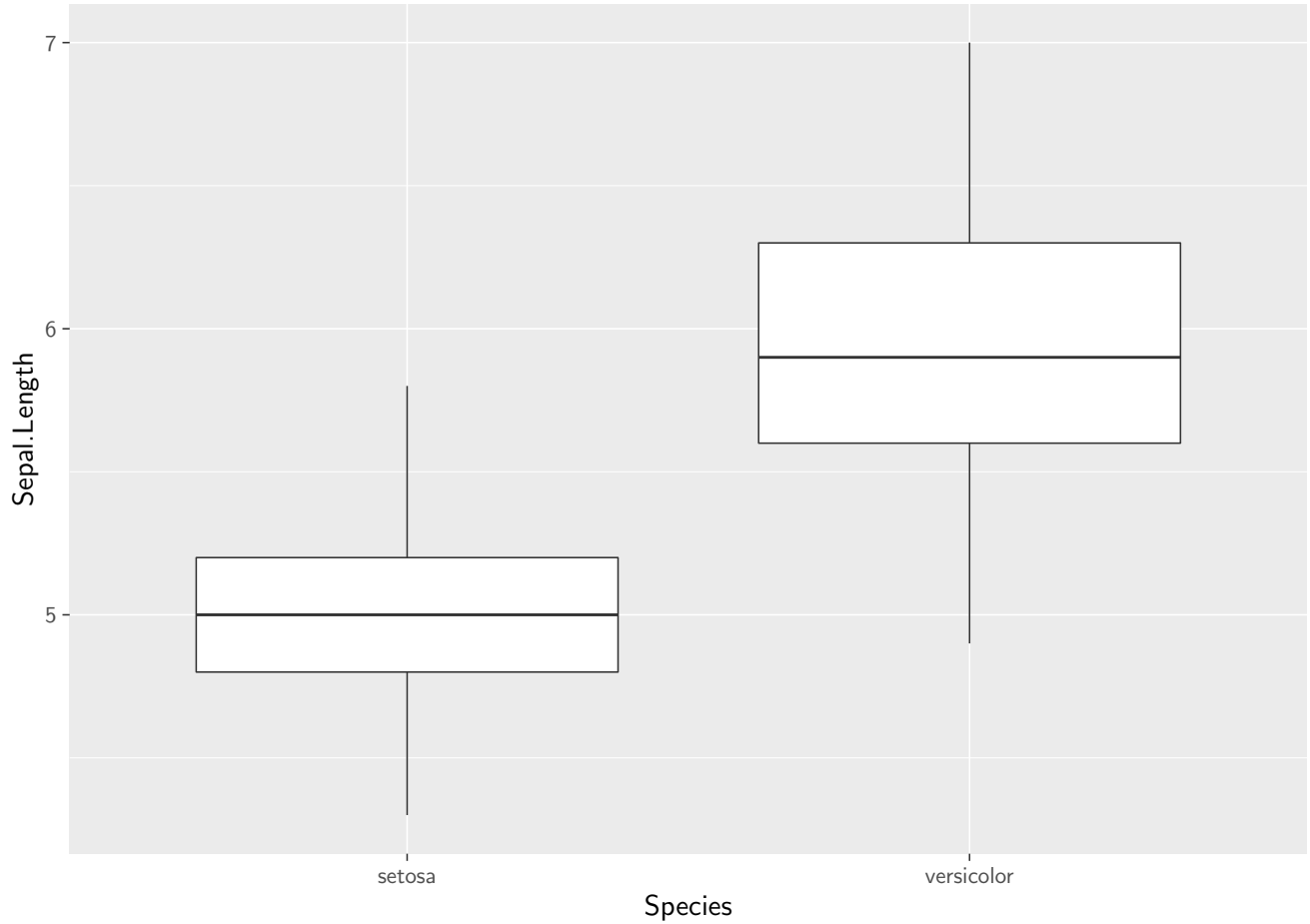
# Iris t.test, visualization

```
boxplot(Sepal.Length ~ Species,
        data = iris[iris$Species %in% c("setosa","versicolor"),],
        drop = TRUE, ylab="Sepal length", xlab="Species")
```



# Iris t.test, visualization

```
library(ggplot2)
  ggplot( iris[iris$Species %in% c("setosa","versicolor"),],
          aes(x=Species, y=Sepal.Length)) + geom_boxplot()
```



# Iris t.test

One t-test for sepal length between *setosa* and *versicolor*:

```
  t.test(x = iris$Sepal.Length[iris$Species == "setosa"],
         y = iris$Sepal.Length[iris$Species == "versicolor"])


Welch Two Sample t-test

data:  iris$Sepal.Length[iris$Species == "setosa"] and iris$Sepal.Le
t = -10.521, df = 86.538, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.1057074 -0.7542926
sample estimates:
mean of x mean of y
    5.006     5.936
```

# When do we know it is different?

4. Statistical estimation
   - a Estimation
     - ★ Cannot know true difference $\mu_{species_1} - \mu_{species_2}$
     - ★ Estimated difference $= \text{Mean}_1 - \text{Mean}_2$
     - ★ Estimated difference contains random variation
   - b Quantify uncertainty / Statistical significance
     - ★ $t = \frac{\text{Mean}_1 - \text{Mean}_2}{\text{Variation}} \frac{\sqrt{\text{Sample Size}}}{\sqrt{2}}$
     - ★ We know exactly how t is distributed when $\mu_{species_1} - \mu_{species_2} = 0$
     - ★ Hence we know probability of $\geq t$ if $\mu_{species_1} - \mu_{species_2} = 0$ ($p$-value)
     - ★ Can derive confidence interval and standard error

# When do we know it is different?

4. Statistical estimation
   - a Estimation
     - ★ Cannot know true difference $\mu_{species_1} - \mu_{species_2}$
     - ★ Estimated difference $= \text{Mean}_1 - \text{Mean}_2$
     - ★ Estimated difference contains random variation
   - b Quantify uncertainty / Statistical significance
     - ★ $t = \frac{\text{Mean}_1 - \text{Mean}_2}{\text{Variation}} \frac{\sqrt{\text{Sample Size}}}{\sqrt{2}}$
     - ★ We know exactly how t is distributed when $\mu_{species_1} - \mu_{species_2} = 0$
     - ★ Hence we know probability of $\geq t$ if $\mu_{species_1} - \mu_{species_2} = 0$ ($p$-value)
     - ★ Can derive confidence interval and standard error

Less uncertainty with

- Larger absolute difference
- Smaller variability
- Larger sample size

# When do we know it is different? Simulations

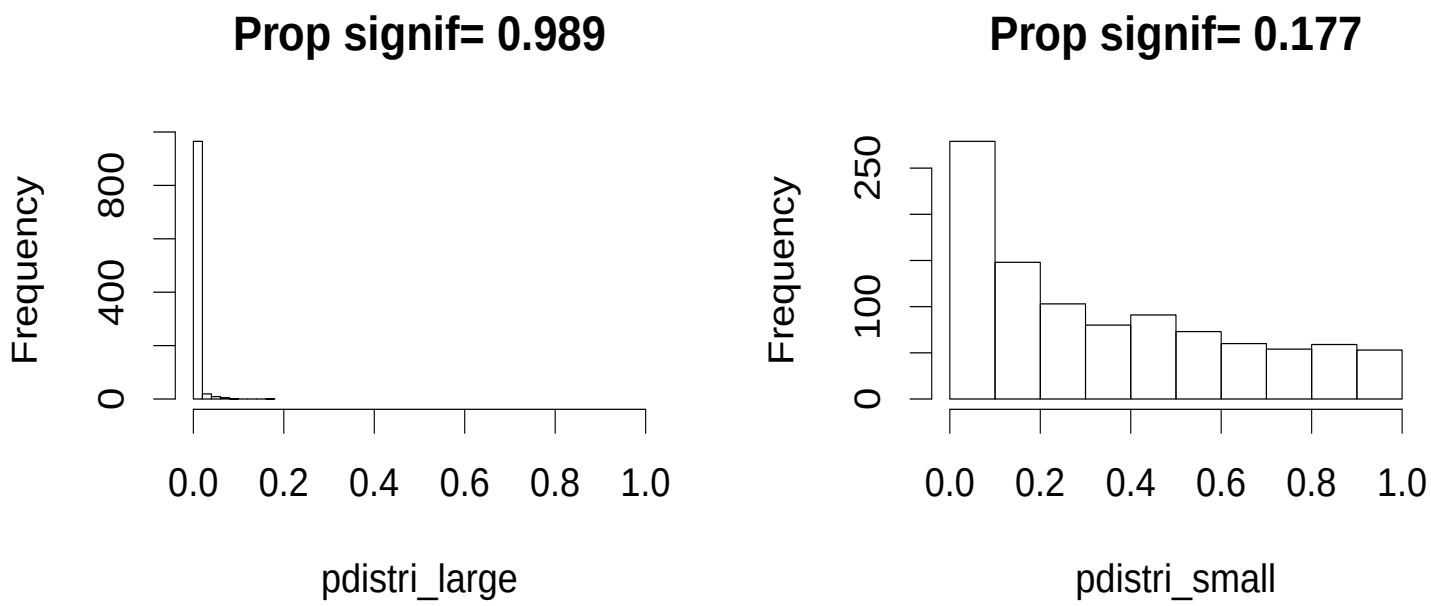
# When do we know it is different? Simulations

## 1. Larger absolute difference

```
nbsim <- 1000
pdistri_large <- vector(length = nbsim)
pdistri_small <- vector(length = nbsim)
for (i in 1:nbsim)
  {
  x1 <- rnorm(n = 10, mean = 2, sd = 1)
  x2 <- rnorm(n = 10, mean = 4, sd = 1) #large diff
  x3 <- rnorm(n = 10, mean = 2.5, sd = 1) #small diff
  out_large <- t.test(x1, x2)
  out_small <- t.test(x1, x3)
  pdistri_large[i]<-out_large$p.value
  pdistri_small[i]<-out_small$p.value
}
```

# When do we know it is different?

```
par(mfrow=c(1,2), cex=2)
hist(pdistri_large, xlim=c(0,1),
     main=paste("Prop signif=",mean(pdistri_large<0.05)))
hist(pdistri_small, xlim=c(0,1),
     main=paste("Prop signif=",mean(pdistri_small<0.05)))
```




```
par(mfrow=c(1,1))
```

# When do we know it is different? Try it!

**Exercise**

Check the effect of smaller variability and/or larger sample size.

# By the way, what are these p-values?

*Probability for a summary statistic to be greater or equal to the observed summary statistic,* **when the null-hypothesis of a given statistical model is true.**

# By the way, what are these p-values?

*Probability for a summary statistic to be greater or equal to the observed summary statistic,* **when the null-hypothesis of a given statistical model is true.**

## Properties

- Depends on the null-hypothesis ($H_0$) of a given model with assumptions
- Uniform distribution under $H_0$ ...
- ... hence proportion(significance under $H_0$) = significance threshold

# By the way, what are these p-values?

*Probability for a summary statistic to be greater or equal to the observed summary statistic,* **when the null-hypothesis of a given statistical model is true.**

## Properties

- Depends on the null-hypothesis ($H_0$) of a given model with assumptions
- Uniform distribution under $H_0$ ...
- ... hence proportion(significance under $H_0$) = significance threshold

**NB: Focus on *p*-value criticized, but common and they are no more evil than other misused statistics!**

# Column selection

```
lm1   <- lm(y ~ x1 + x2)
lm2   <- lm(y ~ 1)
lm3   <- lm(y ~ x1*x2)
lm3   <- lm(y ~ x2)
lm4   <- lm(y ~ x1)
...

plot(lm1)
plot(lm2)
...
```

# Column selection

You changed the name to be more explicit

```
lmadd  <- lm(y ~ x1 + x2)
lmnull <- lm(y ~ 1)
lmff   <- lm(y ~ x1*x2)
lmx2   <- lm(y ~ x2)
lmx1   <- lm(y ~ x1)
...

plot(lm1)
plot(lm2)
...
```

What a pain to repeat the changes in `plot()`!

# Column selection

You changed the name to be more explicit

```
lmadd  <- lm(y ~ x1 + x2)
lmnull <- lm(y ~ 1)
lmff   <- lm(y ~ x1*x2)
lmx2   <- lm(y ~ x2)
lmx1   <- lm(y ~ x1)
...

plot(lm1)
plot(lm2)
...
```

What a pain to repeat the changes in `plot()`!

`Alt + click`

# Column selection

What if my code was not well aligned??

```
lmadd <- lm(y ~ x1 + x2)
lmnull <- lm(y ~ 1)
lmff <- lm(y ~ x1*x2)
lmx2<- lm(y ~ x2)
lmx1  <- lm(y ~ x1)
...

plot(lm1)
plot(lm2)
...
```

# Column selection

What if my code was not well aligned??

```
lmadd <- lm(y ~ x1 + x2)
lmnull <- lm(y ~ 1)
lmff <- lm(y ~ x1*x2)
lmx2<- lm(y ~ x2)
lmx1  <- lm(y ~ x1)
...

plot(lm1)
plot(lm2)
...
```

`Ctrl + Alt + clicks` creates multiple cursors
then `Shift + Home` and `Ctrl + C`

# Shortcuts

`Alt + Shift + K`

# A small example

Animal behavior in response to weather
Load data:

```
getwd()
setwd()
```

```
dat.behav <- read.csv(file = "Data/datbehav.csv") # path to file
```

# A small example

Animal behavior in response to weather
Load data:

```
getwd()
setwd()
```

```
dat.behav <- read.csv(file = "Data/datbehav.csv") # path to file
```

STEP 1: have a look at your data

```
str(dat.behav)
summary(dat.behav)
plot(dat.behav)
```

# t-test

```
fitstudent <- t.test(x = dat.behav$activity[dat.behav$weather==
                                            "rainy"],
                     y = dat.behav$activity[dat.behav$weather==
                                            "sunny"],
                     var.equal = TRUE)
print(fitstudent)
```

```

Two Sample t-test

data:  dat.behav$activity[dat.behav$weather == "rainy"] and dat.beha
t = 3.2752, df = 33, p-value = 0.002485
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.6138373 2.6270325
sample estimates:
mean of x mean of y
 6.781476  5.161041
```

# t-test, graphically

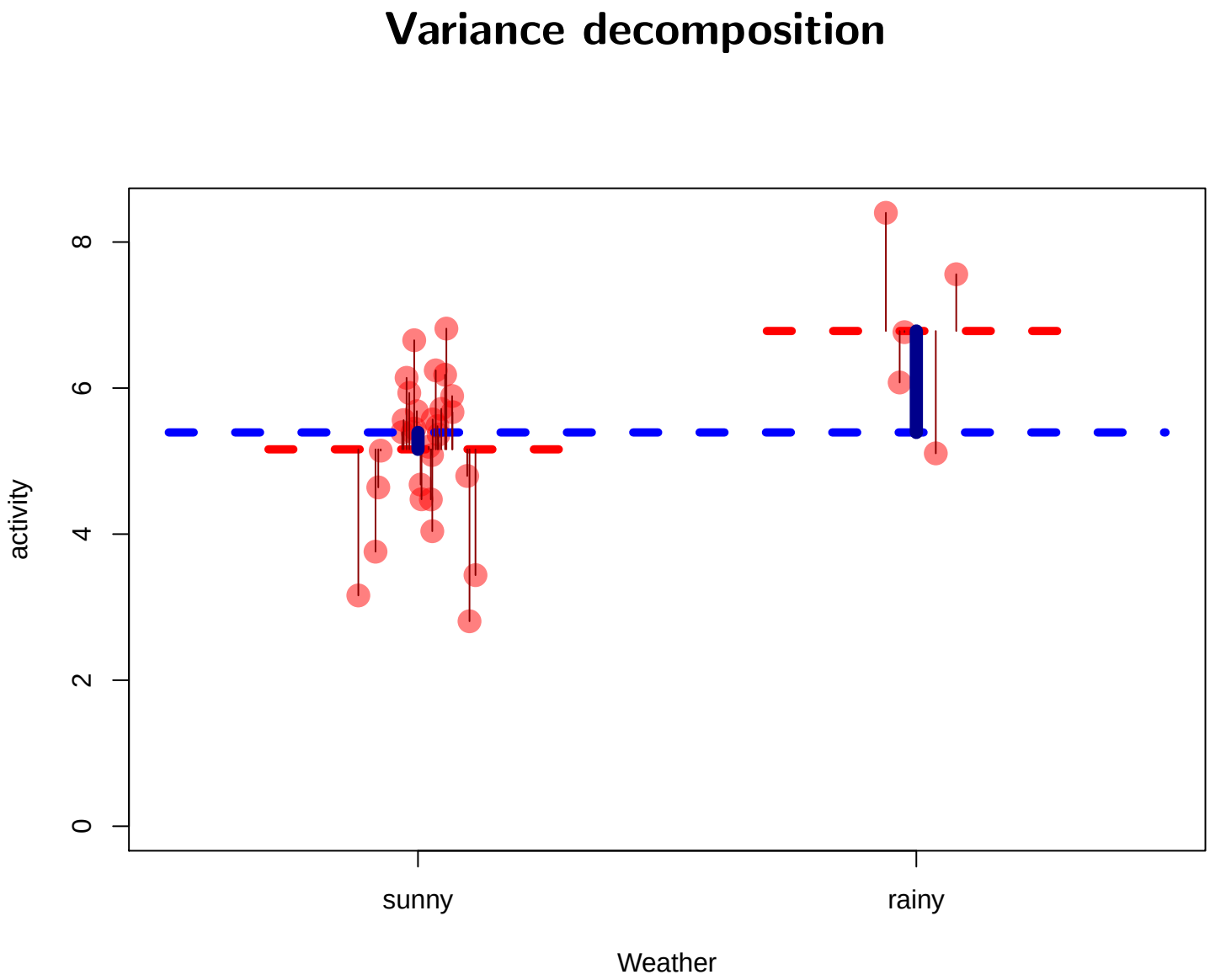
# ANOVA

1. What is the fastest way to get row averages in a data-frame?
2. Create a function called colVars, like colMeans but for variance
3. Create nice plots to visualize iris data (ideally journal-quality)

```
fitanova <- aov(data = dat.behav, formula = activity ~ weather)
summary(fitanova)

            Df Sum Sq Mean Sq F value  Pr(>F)
weather      1  11.25  11.253   10.73 0.00248 **
Residuals   33  34.62   1.049
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# ANOVA, graphically



# Linear regression

```
fitlm <- lm(data = dat.behav, formula = activity ~ weather)
summary(fitlm)


Call:
lm(formula = activity ~ weather, data = dat.behav)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3547 -0.6028  0.2346  0.6419  1.6534

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.7815     0.4581  14.805 3.94e-16 ***
weathersunny  -1.6204     0.4948  -3.275  0.00248 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.024 on 33 degrees of freedom
```

# Regression, graphically

# NB: aov() vs. anova()

```
aov(data = dat.behav, formula = activity ~ weather)
anova(fitlm)
```

# All is one. . .

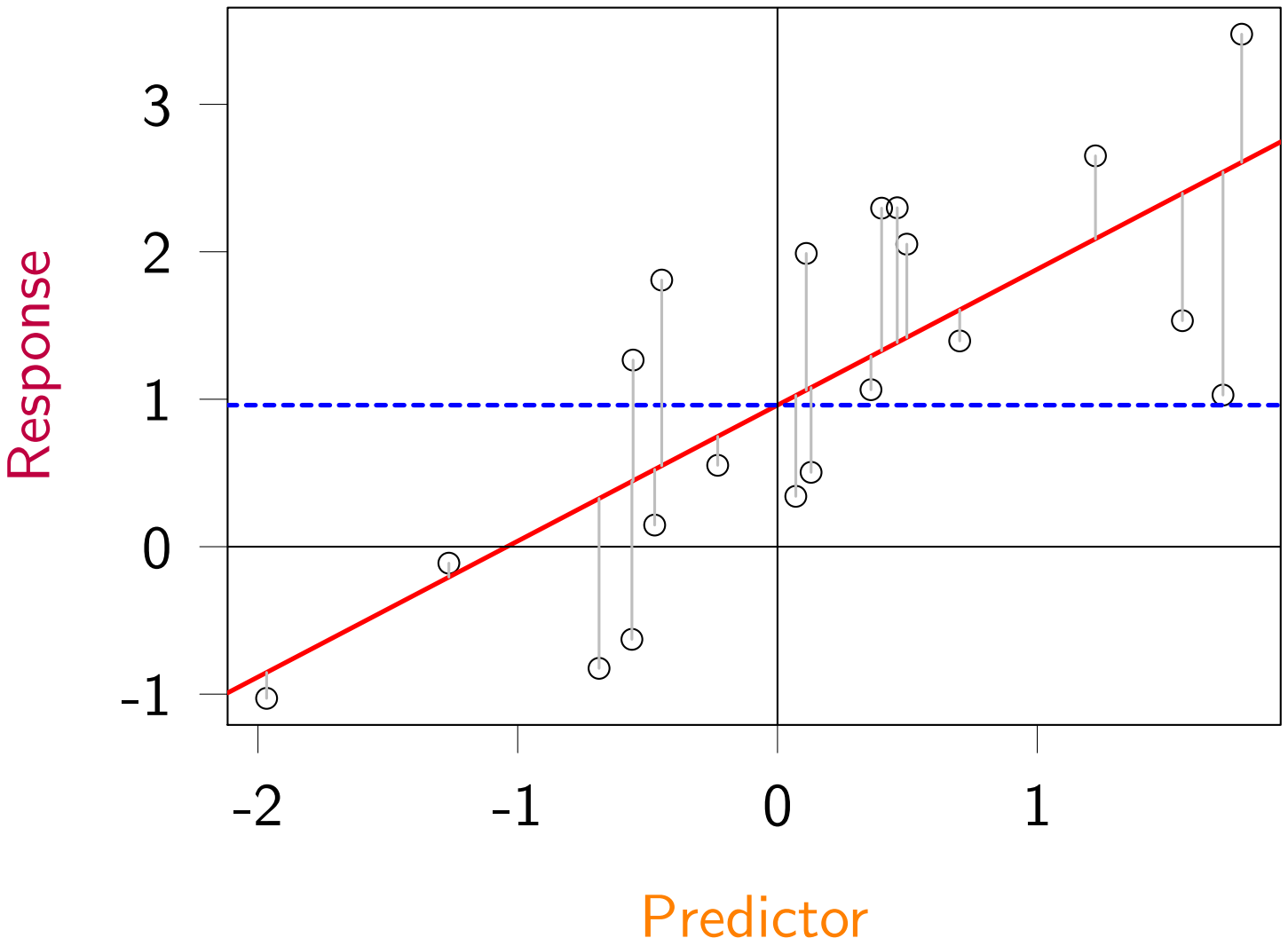
# All is one...

## ...but `lm()` rules!

- t-test, ANOVA, regression and others can be mathematically equivalent
- In R, `lm()` and related functions can do them all...
- ...and much more!

# A simple linear model

**Response** = **Intercept** + **Slope** × **Predictor** + **Error**



# A simple linear model

**Response** = **Intercept** + **Slope** × **Predictor** + **Error**

**In R:**

```
lm(response ~ 1 + predictor1 + predictor2, data=data)
  # equivalent to
lm(response ~ predictor1 + predictor2, data=data)
```

- Intercept can be explicit or implicit
- Can remove intercept with `...~ 0 + ...`
- Error is implicit
- Feed the option `data=` to keep code short, reliable and flexible
- Order of predictors do not matter

# Interpretation

```
  Ans <- read.csv(file = "Anscombe.csv")

Warning in file(file, "rt"):  cannot open file 'Anscombe.csv':
No such file or directory
Error in file(file, "rt"):  cannot open the connection
```

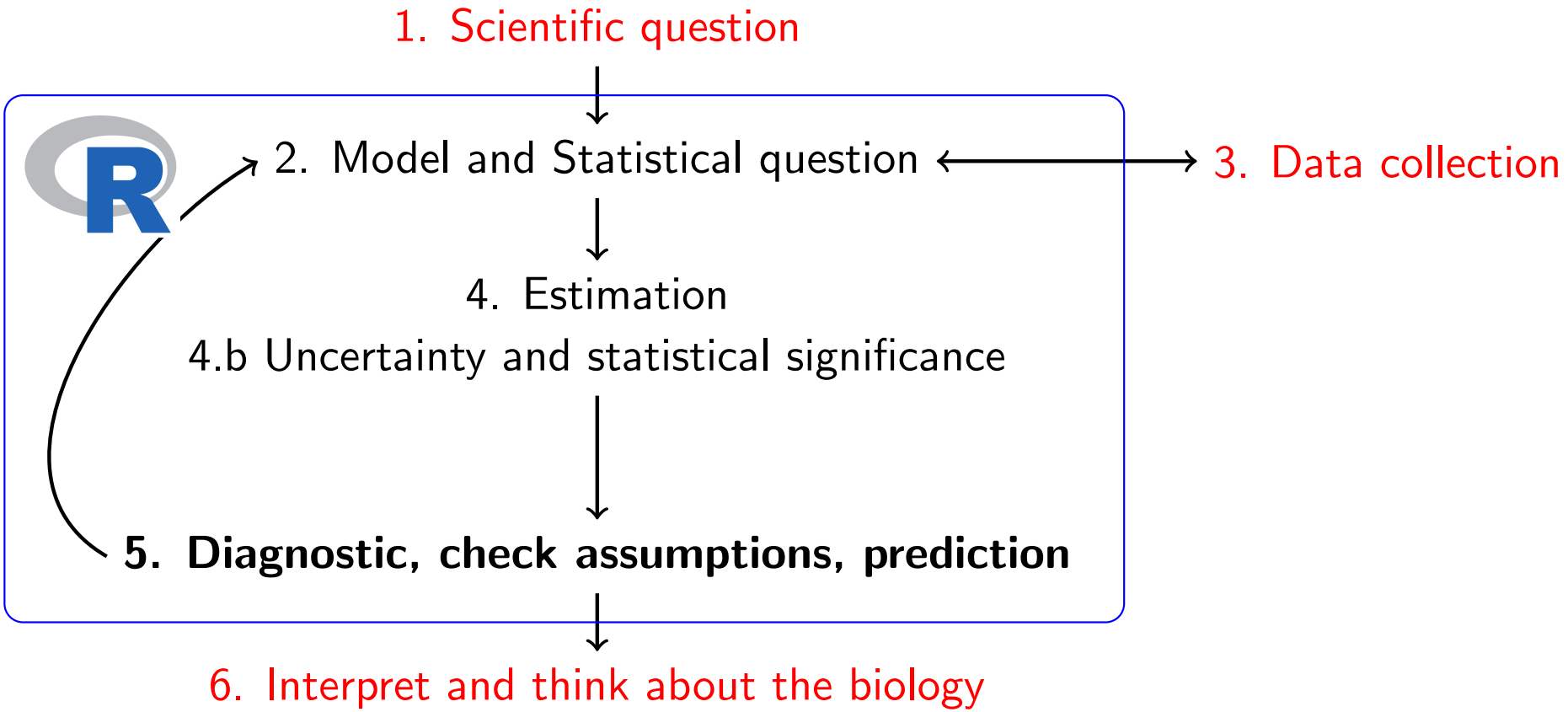
```
  lm1 <- lm(y ~ x , data=Ans[Ans$distri==1,])
  summary(lm1)
  plot(Ans$x[Ans$distri==1], Ans$y[Ans$distri==1],
       xlim=c(0,15), ylim=c(0,12))
  abline(lm1)
```

# Interpretation

## lm vs. plot

- Fit a linear model $y \sim x$for each of the four "distri"
- Plot the relationship $y \sim x$ for each of the four "distri"
- Can we trust these models? For what? *I expect more than "it's all bullshit"*

# General approach



# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )
  *Risk: biologically meaningless*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )
  *Risk: biologically meaningless*
- Predictor not perfectly correlated
  *Risk: Model won't run, unstable convergence, or huge SE*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )
  *Risk: biologically meaningless*
- Predictor not perfectly correlated
  *Risk: Model won't run, unstable convergence, or huge SE*
- Little error in predictors
  *Risk: bias estimates (underestimate with Gaussian error)*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )
  *Risk: biologically meaningless*
- Predictor not perfectly correlated
  *Risk: Model won't run, unstable convergence, or huge SE*
- Little error in predictors
  *Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution
  *Risk: Poor predictions*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )
  *Risk: biologically meaningless*
- Predictor not perfectly correlated
  *Risk: Model won't run, unstable convergence, or huge SE*
- Little error in predictors
  *Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution
  *Risk: Poor predictions*
- Homoscedasticity (constant error variance)
  *Risk: Over-optimistic uncertainty, unreliable predictions*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )
  *Risk: biologically meaningless*
- Predictor not perfectly correlated
  *Risk: Model won't run, unstable convergence, or huge SE*
- Little error in predictors
  *Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution
  *Risk: Poor predictions*
- Homoscedasticity (constant error variance)
  *Risk: Over-optimistic uncertainty, unreliable predictions*
- Independence of error
  *Risk: Bias and over-optimistic uncertainty*

# Diagnostic: summary and plot

```
 lm1 <- lm(y ~ x , data=Ans[Ans$distri==1,])
```

```
Error in is.data.frame(data):  object 'Ans' not found
```

```
 lm2 <- lm(y ~ x , data=Ans[Ans$distri==2,])
```

```
Error in is.data.frame(data):  object 'Ans' not found
```

```
 summary(lm1)
 par(mfrow=c(2,2))
 plot(lm1)
```

```
  summary(lm2)
 plot(lm2)
  par(mfrow=c(1,1))
```

# Diagnostic: prediction

```
pred2 <- predict(lm2, se.fit = TRUE, interval = "confidence")
```

```
Error in predict(lm2, se.fit = TRUE, interval = "confidence"):
object 'lm2' not found
```

```
pred2 <- cbind(Ans[Ans$distri==2,], pred2)
```

```
Error in cbind(Ans[Ans$distri == 2, ], pred2):  object 'Ans' not
found
```

# Diagnostic: prediction

```
pred2 <- predict(lm2, se.fit = TRUE, interval = "confidence")
```

```
Error in predict(lm2, se.fit = TRUE, interval = "confidence"):
object 'lm2' not found
```

```
pred2 <- cbind(Ans[Ans$distri==2,], pred2)
```

```
Error in cbind(Ans[Ans$distri == 2, ], pred2):  object 'Ans' not
found
```

```
Error in plot(pred2$x, pred2$y, ylim = c(0, 12), col = "red", pch
= 16, :  object 'pred2' not found
Error in lines(pred2$x, pred2$fit.fit):  object 'pred2' not found
Error in arrows(x0 = pred2$x, y0 = pred2$fit.lwr, y1 =
pred2$fit.upr, :  object 'pred2' not found
Error in strwidth(legend, units = "user", cex = cex, font =
text.font):  plot.new has not been called yet
```

# Practice lm() with parasites

## What explains variation in parasitic load?

You collected ecto-parasites on some furry large mammals at three locations. Parasites break easily when we collect them and are impossible to count, so we decide to measure parasitic load as their mass. **Why do some mammals have larger parasitic load?**

# Practice lm() with parasites

## What explains variation in parasitic load?

You collected ecto-parasites on some furry large mammals at three locations. Parasites break easily when we collect them and are impossible to count, so we decide to measure parasitic load as their mass. **Why do some mammals have larger parasitic load?**

- Load the `Para.csv` data (don't forget: str(), summary(), plot()...)
- Model `Parasite_Mass` using `lm()`
- Find what variables predict `Parasite_Mass`
- How good are your models? Assumptions? Prediction?
- What biological interpretation can you imagine?

# Extra exercises

## General R coding

1. What is the fastest way to get row averages in a data-frame?
2. Create a function called colVars, like colMeans but for variance
3. Create nice plots to visualize iris data (ideally journal-quality)

## Linear models

1. Load Cdata.csv, fit models of y predited by x1 and x2, or x2 and x3. Something is weird, what is going on? What to do?
2. For model that can be fitted with t.test, aov, and lm, is one of the function faster?
3. Write your own code to obtain a prediction from a lm (that is, a simpler version of the predict function), with confidence interval. (extra toughness: do it using the matrix formulation of the analytical solution to a linear model)