Today

- Questions from homework (brief)?
- Linear regression reflection
- Regression workflow algorithm
- Repetition structures:
  - counter control using for

Linear regression reflection

- What was the goal of your analysis?
  - what question were you asking?
- How did the analysis answer the question?
  - what output (e.g. numerical, graphical) answered the question?

Your examples

Your question	Your output
• Predict plant height from permafrost thickness	• Model summary, p-value, R^2, plot
• Predict hybrid index of birds from elevation	• as above
• Does the time for a squirrel to husk a cone depend on the width of the cone?	• as above
• Does Iris sepal length depend on species?	• p-value and pairwise differences

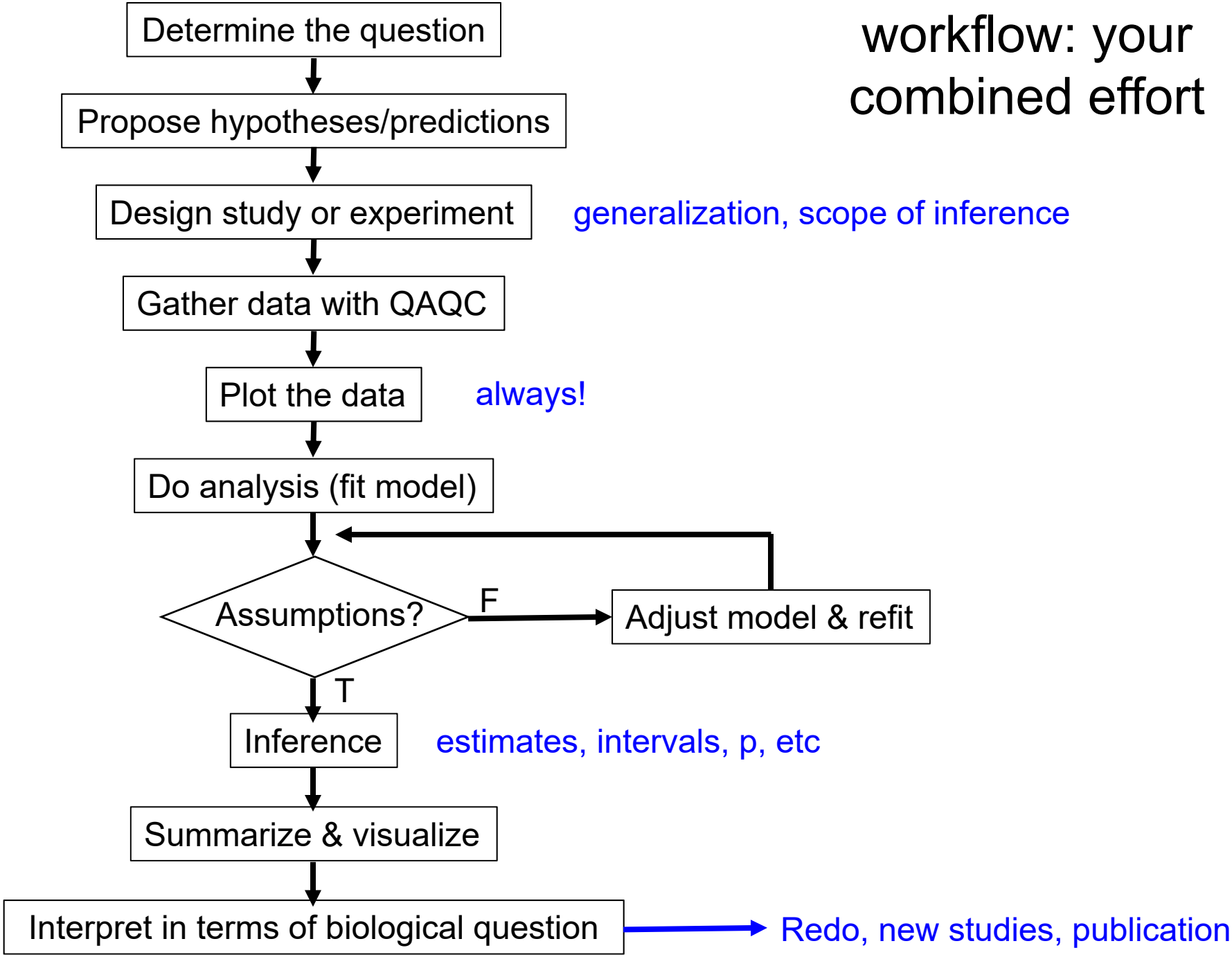
Goals of your lm analyses

- Estimation (various quantities) 0
  - what is the state of the world?
- Is there an effect? (H test) 8 Unlearn!
  - ... but there is always an effect, just small
  - causation (mechanism) vs association
- Prediction 2
  - what will be the state of the world given some other information we know?

Workflow algorithm

- Combine your efforts to make a workflow algorithm for linear regression
  - diagram it, e.g. flowchart
- What did you learn as a group?
  - things that not everyone had

workflow: your combined effort



Workflow algorithms in DS

NSF master data science algorithm

```
Plan for data
Acquire data
Manage data
Analyze data
Infer from data
Report about data
```

R: `for` repetition structure

Most programming languages have a specialized structure for counter-controlled repetition (usually called "for")

```
for ( i in starti:endi ) {
    expression
}
```

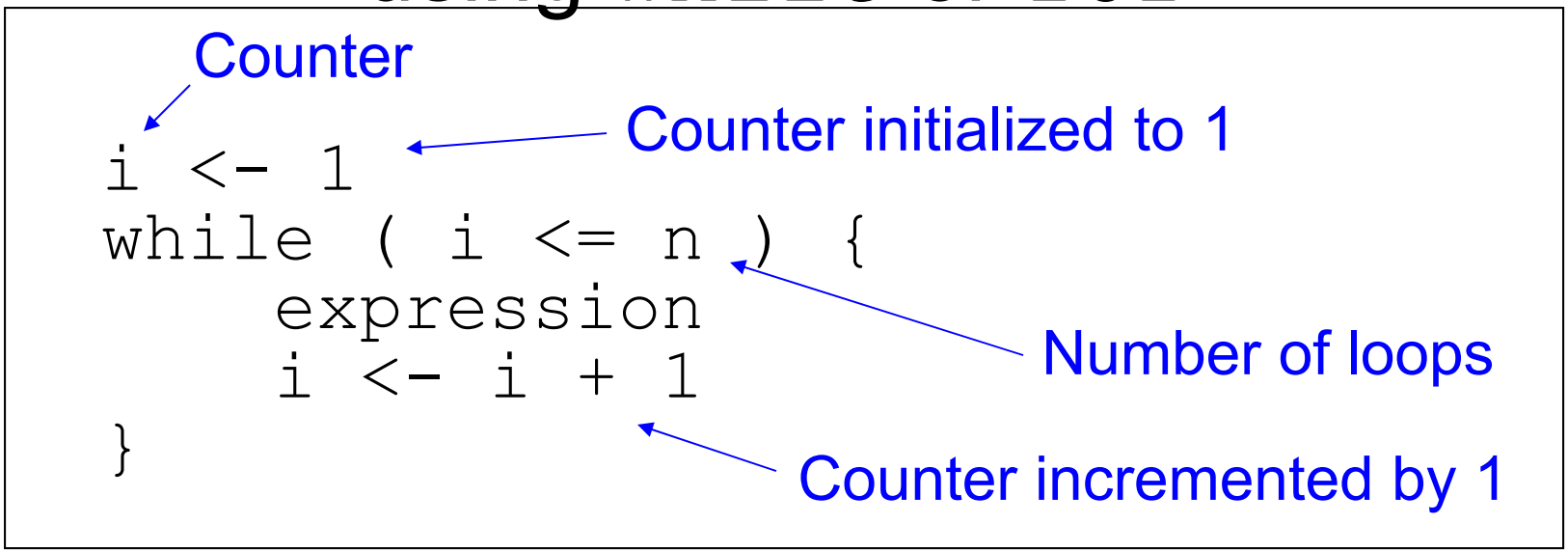
R: `for` repetition structure

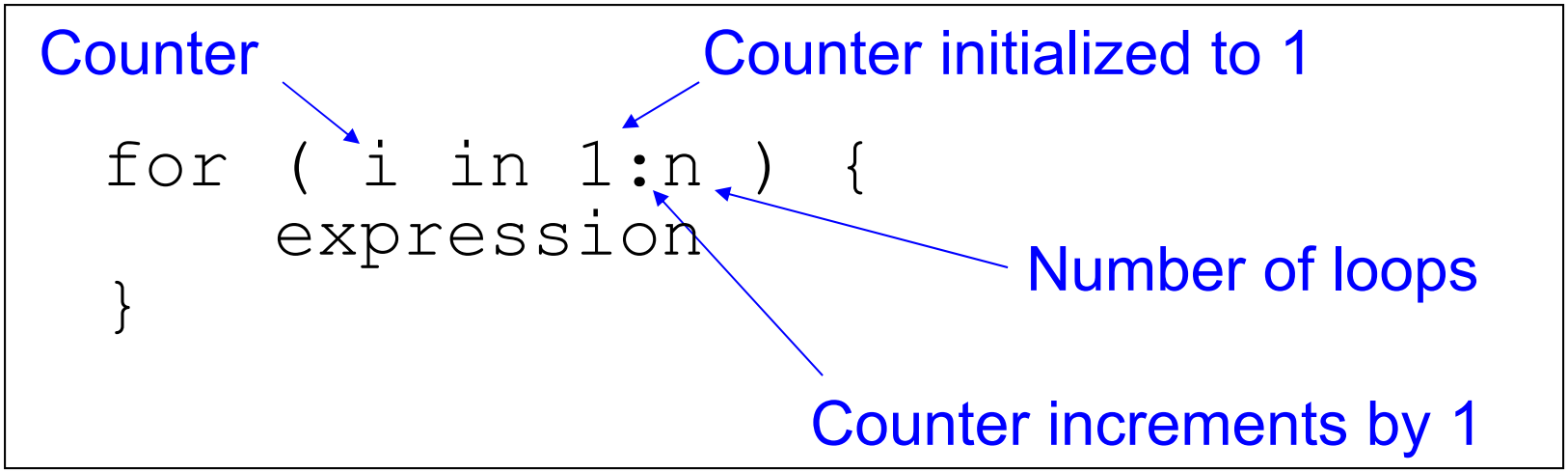
Example

```
for ( i in 1:10 ) {
    j <- i * 2
    print(j)
}
```

What does this do?

The 4 components of counter control using `while` or `for`





R: `for` repetition structure

Correct

```
for ( i in 1:n ) {
    expression
}
```

Incorrect

```
i <- 1
for ( i in 1:n ) {
    expression
    i <- i + 1
}
```

```
# Finds the number (y) that is the zth power of x

# Initialize parameters
x <- 3.2     #Any real number
z <- 2       #Any integer > 0

# Initialize working variables
y <- 1
counter <- 1

# Processing phase
while ( counter <= z ) {
    y <- y * x
    counter <- counter + 1
}

# Termination phase
y
```

This code uses a while structure to do counter controlled repetition. Modify it to use a for counter-control structure instead.

Increment variation

```
for ( i in seq(0, n, 2) ) {
    expression
}
```

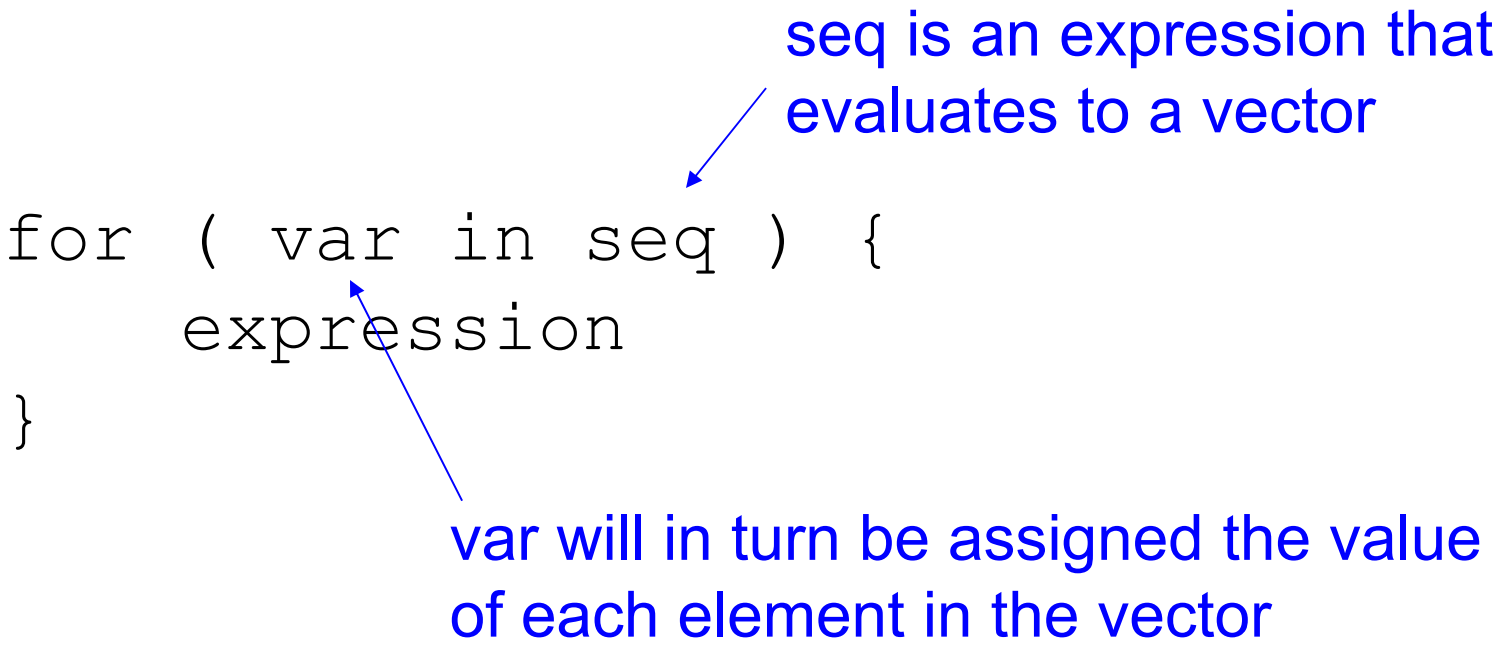
```
i <- 0
while ( i <= n ) {
    expression
    i <- i + 2
}
```

Counter initialized to 0

Counter incremented by 2

R: `for` is vector controlled

R's for structure is actually vector controlled repetition, a special case of counter controlled repetition



Any vector will do!

R: `for` is vector controlled

Example

```
a <- c(0.51,0.57,0.09,1.02,1.10)
for ( number in a ) {
    print(number * 2)
}
```

What does this do?