Dialogue

Goal

In this document, the dialogue with the AI is described, including all control questions. Of each question, the expected answer is calculated.

Overview

The dialogue follows the skill levels of `[Spector & Ma, 2019]` in increasing order:

Chapter	Ability	Description
D1.1	Interpretation	Read the data
D1.2	Explanation	Reproduce the data
D1.3	Reasoning	Reason about the data
D1.4	Analysis	Work with the data
D1.5	Evaluation	Double-check progress
D1.6	Synthesis	Combine data with research paper
D1.7	Reflection	Not applicable
D2	Judgement	Ask Q1
D3.1	Analysis	Work with data
D3.2	Evaluation	Double-check progress
D3.3	Synthesis	Combine data with alternative line of reasoning
D3.4	Reflection	Not applicable
D4	Judgement	Ask Q2

Expectations

A paper does not typically hold code (unlike an R Markdown document as this one) and commonly all the results of an analysis are copy-pasted into a paper and then left unchecked.

To make it easy to compare the final paper with the constants we've copy-pasted, here are all the constants:

```
## Explain
first_fish_id <- 181
first_origin <- "Lake"
first_transplant <- "Lake"
first_pre_mass <- 1.00
first_fish_survived <- 0
first_enclosure_id <- "L1"

## Reasoning
origin_values <- c("Lake", "Stream")
transplanted_values <- c("Lake", "Stream", "Control")
n_all_survived <- 151
n_all_died <- 89
n_all_ignored <- 60
total_n_fish <- 300
n_fish_control <- 60
```

```r
n_fish_transplanted <- 240
lowest_pre_mass <- 0.5
highest_pre_mass <- 3.6
n_survived <- 151
n_died <- 89
n_enclosures <- 80
n_enclosures_with_2_fish <- 3
n_enclosures_with_3_fish <- 74
n_enclosures_with_4_fish <- 3
anomalous_enclosures <- c("L20", "L25", "L6", "L7", "S22", "S24")
cage_mass_mean_for_l1 <- 1.256667
cage_mass_stdev_for_l1 <- 0.4532475
cage_mass_mean_deviation_sd_for_l1 <- c(1.1546303, -0.5662836, -0.5883466)

abs_cage_mass_mean_deviation_sd_for_l1 <- abs(cage_mass_mean_deviation_sd_for_l1)
```

Here we check if our expectations are logically valid:

```r
library(testthat)

# Check consistency between these variables
expect_equal(
  total_n_fish,
  n_all_survived + n_all_died + n_all_ignored
)

expect_equal(
  total_n_fish,
  n_fish_control + n_fish_transplanted
)
expect_equal(
  n_fish_transplanted,
  n_survived + n_died
)
expect_equal(
  n_enclosures,
  n_enclosures_with_2_fish + n_enclosures_with_3_fish + n_enclosures_with_4_fish
)
expect_equal(
  length(anomalous_enclosures),
  n_enclosures_with_2_fish + n_enclosures_with_4_fish
)
```

All solutions to the dialog are written here, and tested below, during the dialogue.

To test this, the file with the data needs to be present in the working folder:

```r
dataset_filename <- "Bolnick_traits.txt"
if (!file.exists(dataset_filename)) {
  stop(
    "Dataset not found at path '", dataset_filename, "' \n",
    "Current working directory: ", getwd(), " \n"
  )
}
```

Helper functions

Here are some functions we need in the analysis:

```
#' Determines if a value is TRUE. Is FALSE for NA
is_true <- function(x) { !is.na(x) & x == TRUE  }
expect_true(is_true(TRUE))
expect_false(is_true(FALSE))
expect_false(is_true(NA))

#' Determines if a value is FALSE. Is FALSE for NA
is_false <- function(x) { !is.na(x) & x == FALSE}
expect_true(is_false(FALSE))
expect_false(is_false(TRUE))
expect_false(is_false(NA))
```

D1. Teach an AI about the data and line of reasoning in the paper

These subsections are numbered by the abilities taught to the AI

D1.1: Interpretation

In these steps, we let the AI read the data.

D1.1.1: Read data

- The dialogue starts by uploading the data for the paper at the notes from the literature club.

```
t_all <- read.csv("Bolnick_traits.txt", sep = " ")
```

D1.1.2: Show first rows

Ask the AI:

```
Could you show the first rows of the data?
```

Expected:

```
knitr::kable(head(t_all))
```

sample	ID	fishID	origin	enclosure	transplant	date_in	pre_mass	pre_length	post_mass	post_length	gth	bd	gw	grn	grl	sex	survived
118	118	Stream	S2	Stream	06/03/11	1.75	4.9	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
12	12	Stream	L8	Lake	06/02/11	1.13	4.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
122	122	Lake	S2	Stream	06/03/11	2.37	5.6	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
123	123	Lake	S3	Stream	06/03/11	1.71	5.2	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
124	124	Lake	S4	Stream	06/03/11	2.45	5.4	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
127	127	Lake	S7	Stream	06/03/10	0.85	4.0	NA	NA	NA	NA	NA	NA	NA	NA	NA	0

We select only the relevant data here, so that the analysis can be checked by humans more easily:

```
t <- t_all |>
  dplyr::select(
    fishID,
    origin,
    enclosure,
    pre_mass,
    transplant,
    survived
```

```
  ) |> dplyr::arrange(enclosure)
knitr::kable(head(t))
```

	fishID	origin	enclosure	pre_mass	transplant	survived
27	181	Lake	L1	1.00	Lake	0
28	182	Lake	L1	0.99	Lake	0
149	1	Stream	L1	1.78	Lake	1
16	16	Stream	L10	1.22	Lake	0
36	193	Lake	L10	1.71	Lake	0
164	17	Stream	L10	2.06	Lake	1

D1.2: Explanation

In these steps, we let the AI demonstrate it has read the data.

D1.2.1: Explain 'fishID'

Ask the AI:

```
The 'fishID' column denotes the ID of a fish.
Each fish has a unique ID.
Could you give me the ID of the first fish in the dataset?
```

Expected:

```
expect_equal(t$fishID[1], first_fish_id)
```

The expected answer is 181.

D1.2.2: Explain 'origin'

Ask the AI:

```
The 'origin' column denotes the location where each fish comes from.
What is the location the first fish came from?
```

Expected:

```
expect_equal(first_origin, t$origin[1])
```

The correct answers is Lake.

D1.2.3: Explain 'transplant' values

Ask the AI:

```
The 'transplant' column denotes the location where each fish is
transplanted to.
A value of 'Stream' denotes that the fish is part of the experiment
and is translated to a stream.
A value of 'Lake' denotes that the fish is part of the experiment
and is translated to a lake.
A value of 'Control' denotes that this fish was not part of the experiment
and was part of the control group instead.
What is the locations the first fish is transplanted to?
```

Expected:

```
expect_equal(t$transplant[1], first_transplant)
```

The correct answers is Lake.

D1.2.4: Explain 'pre_mass' Ask the AI:

```
The `pre_mass` column denotes the mass of a fish
before the transplantation.
A value of 'NA' denotes that the fish has not been weighted
before transplantation.

About the first fish: has it been weighted before transplantation?
If yes, what is its mass before transplantation?
```

Expected:

```
expect_equal(t$pre_mass[1], first_pre_mass)
```

The correct answer is 0.5 to 3.6.

D1.2.5: Explain 'survived' Ask the AI:

```
The `survived` column denotes if the fish survived the experiment.
A value of 0 means that the fish died in the experiment.
A value of 1 means that the fish survived the experiment.
A value of NA means that the fish was irrevant to the experiment.
```

Expected:

```
expect_equal(t$survived[1], first_fish_survived)
```

The correct answers is 0.

D1.2.6: Explain 'enclosure' Ask the AI:

```
The `enclosure` column denotes the ID of an enclosure.
Each enclose has a unique ID.
What is the ID of the first enclosure?
```

Expected:

```
expect_equal(t$enclosure[1], first_enclosure_id)
```

The correct answer is L1.

D1.3: Reasoning

In these steps, we let the AI reason about the data, such as producing a tally.

D1.3.1: Reason about the 'fishID' values Ask the AI:

```
How many different fish are in this dataset?
```

Expected:

```
expect_equal(total_n_fish, length(unique(t$fishID)))
```

The expected answer is 300.

D1.3.2: Reason about the 'origin' values Ask the AI:

```
What are the locations the fish come from?
```

Expected:

```
expect_equal(unique(t$origin), origin_values)
```

The correct answers is Lake, Stream.

D1.3.3: Reason about 'transplant': count the fish in experiment Ask the AI:

```
How many fish are part of the experiment?
```

Expected:

```
t_1 <- t |> dplyr::filter(transplant != "Control")
expect_equal(n_fish_transplanted, length(unique(t_1$fishID)))
```

The correct answer is 240.

D1.3.4: Reason about the 'pre_mass' values Ask the AI:

```
What is lowest fish mass before transplantation?
What is heighest fish mass before transplantation?
```

Expected:

```
expect_equal(lowest_pre_mass, min(t_1$pre_mass))
expect_equal(highest_pre_mass, max(t_1$pre_mass))
```

The correct answer is 0.5 to 3.6.

D1.3.5: Reason with 'survived': count the survivors Ask the AI:

```
How many fish died?
How many fish survived?
How many fish were not part of the experiment?
```

Expected:

```
expect_equal(n_all_survived, sum(is_true(t$survived)))
expect_equal(n_all_died, sum(is_false(t$survived)))
expect_equal(n_all_ignored, sum(is.na(t$survived)))
```

The correct answers are:

- 89 fish died
- 151 fish survived
- 60 fish were not part of the experiment?

D1.3.6.1: Reason about the 'enclosure' values: count the nub Ask the AI:

```
How many enclosures are used in the experiment?
```

Expected:

```
expect_equal(n_enclosures, length(unique(t_1$enclosure)))
```

The correct answer is 80 enclosures.

D1.3.6.2: Reasom about 'enclosure': tally the number of fish Ask the AI:

```
How many fish, that are part of the experiment, are in each enclosure?
```

Expected:

```
t_1 <- t |> dplyr::filter(transplant != "Control")
n_fish_per_enclosure <- dplyr::count(t_1, enclosure)
expect_equal(n_enclosures_with_2_fish, sum(n_fish_per_enclosure$n == 2))
expect_equal(n_enclosures_with_3_fish, sum(n_fish_per_enclosure$n == 3))
expect_equal(n_enclosures_with_4_fish, sum(n_fish_per_enclosure$n == 4))
knitr::kable(head(n_fish_per_enclosure))
```

enclosure	n
L1	3
L10	3
L11	3
L12	3
L13	3
L14	3

The correct answer is:

- 3 enclosures have 2 fish
- 74 enclosures have 3 fish
- 3 enclosures have 4 fish

As a note to self, these are the anomalies:

```
expect_equal(anomalous_enclosures, n_fish_per_enclosure[n_fish_per_enclosure$n != 3, ]$enclosure)
knitr::kable(n_fish_per_enclosure[n_fish_per_enclosure$n != 3, ])
```

	enclosure	n
13	L20	4
18	L25	2
37	L6	2
38	L7	4
55	S22	4
57	S24	2

D1.4: Analysis

In these steps, we work with the data, in the same way as the paper did.

D1.4.1: calculate 'cage_mass_mean' Ask the AI:

```
To make a better comparison, we are going to standarize
body masses per enclosure.
Add a column to the
data called `cage_mass_mean` which holds the average
`pre_mass` within the enclosure each fish is in.
Could you show me the data for enclosure L1?
```

Expected:

```
cage_mass_mean_per_enclosure <- t_1 |>
  dplyr::select(enclosure, pre_mass) |>
  dplyr::group_by(enclosure) |>
  dplyr::summarise(cage_mass_mean = mean(pre_mass))
```

```
t_2 <- merge(t_1, cage_mass_mean_per_enclosure)
knitr::kable(head(t_2))
```

enclosure	fishID	origin	pre_mass	transplant	survived	cage_mass_mean
L1	181	Lake	1.00	Lake	0	1.256667
L1	182	Lake	0.99	Lake	0	1.256667
L1	1	Stream	1.78	Lake	1	1.256667
L10	16	Stream	1.22	Lake	0	1.663333
L10	193	Lake	1.71	Lake	0	1.663333
L10	17	Stream	2.06	Lake	1	1.663333

```
knitr::kable(t_2[t_2$enclosure == "L1", ])
```

enclosure	fishID	origin	pre_mass	transplant	survived	cage_mass_mean
L1	181	Lake	1.00	Lake	0	1.256667
L1	182	Lake	0.99	Lake	0	1.256667
L1	1	Stream	1.78	Lake	1	1.256667

```
expect_equal(
  t_2[t_2$enclosure == "L1", ]$cage_mass_mean,
  rep(cage_mass_mean_for_l1, 3),
  tolerance = 1.0e-6
)
```

For enclose L1, the expected `cage_mass_mean` for each of the fish is 1.256667.

D1.4.2: calculate 'cage_mass_stdev' Ask the AI:

```
To make a better comparison, we are going to standarize
body masses per enclosure. Add a column to the
data called `cage_mass_stdev` which is the standard
deviation of the `pre_mass` distribution of each enclosure
each fish is in.
Could you show me the data for enclosure L1?
```

Expected:

```
cage_mass_stdev_per_enclosure <- t_2 |>
  dplyr::select(enclosure, pre_mass) |>
  dplyr::group_by(enclosure) |>
  dplyr::summarise(cage_mass_stdev = sd(pre_mass))
t_3 <- merge(t_2, cage_mass_stdev_per_enclosure)
knitr::kable(head(t_3))
```

enclosure	fishID	origin	pre_mass	transplant	survived	cage_mass_mean	cage_mass_stdev
L1	181	Lake	1.00	Lake	0	1.256667	0.4532475
L1	182	Lake	0.99	Lake	0	1.256667	0.4532475
L1	1	Stream	1.78	Lake	1	1.256667	0.4532475
L10	16	Stream	1.22	Lake	0	1.663333	0.4219400
L10	193	Lake	1.71	Lake	0	1.663333	0.4219400
L10	17	Stream	2.06	Lake	1	1.663333	0.4219400

```
knitr::kable(t_3[t_3$enclosure == "L1", ])
```

enclosure	fishID	origin	pre_mass	transplant	survived	cage_mass_mean	cage_mass_stdev
L1	181	Lake	1.00	Lake	0	1.256667	0.4532475
L1	182	Lake	0.99	Lake	0	1.256667	0.4532475
L1	1	Stream	1.78	Lake	1	1.256667	0.4532475

```
expect_equal(
  t_3[t_3$enclosure == "L1", ]$cage_mass_stdev,
  rep(cage_mass_stdev_for_l1, 3),
  tolerance = 1.0e-6
)
```

D1.4.3: calculate 'cage_mass_mean_deviation_sd' Ask the AI:

```
To make a better comparison, we are going to standarize
body masses per enclosure. Add a column to the
data called `cage_mass_mean_deviation_sd`.
Its values are calculated per fish.
Each fish its `cage_mass_mean_deviation_sd`
equals the absolute difference between its `pre_mass`
and its enclosure's `cage_mass_mean`,
divided by the `cage_mass_stdev` of its enclosure.
Could you show me the data for enclosure L1?
```

Expected:

```
cage_mass_mean_deviation_sd_per_fish <- t_3 |>
  dplyr::select(fishID, pre_mass, cage_mass_mean, cage_mass_stdev) |>
  dplyr::group_by(fishID) |>
  dplyr::mutate(
    cage_mass_mean_deviation_sd =
      abs(pre_mass - cage_mass_mean) / cage_mass_stdev
  ) |>
  dplyr::select(fishID, cage_mass_mean_deviation_sd)
expect_equal(
  nrow(cage_mass_mean_deviation_sd_per_fish),
  n_fish_transplanted
)

expect_equal(
  nrow(t_3),
  n_fish_transplanted
)
t_4 <- merge(t_3, cage_mass_mean_deviation_sd_per_fish)
expect_equal(
  nrow(t_4),
  n_fish_transplanted
)

knitr::kable(head(t_4))
```

fishID	enclosure	origin	pre_mass	transplant	survived	cage_mass_mean	cage_mass_stdev	cage_mass_mean_deviation_sd
1	L1	Stream	1.78	Lake	1	1.256667	0.4532475	1.1546303
2	L2	Stream	0.65	Lake	1	1.016667	0.5589574	0.6559832
3	L2	Stream	1.66	Lake	1	1.016667	0.5589574	1.1509524
4	L3	Stream	2.26	Lake	1	1.453333	0.8309834	0.9707375
5	L3	Stream	1.50	Lake	0	1.453333	0.8309834	0.0561584
6	L4	Stream	1.89	Lake	1	1.723333	0.9016282	0.1848508

```
knitr::kable(t_4[t_4$enclosure == "L1", ])
```

	fishID	enclosure	origin	pre_mass	transplant	survived	cage_mass_mean	cage_mass_stdev	cage_mass_mean_deviation_sd
1	1	L1	Stream	1.78	Lake	1	1.256667	0.4532475	1.1546303
181	181	L1	Lake	1.00	Lake	0	1.256667	0.4532475	0.5662836
182	182	L1	Lake	0.99	Lake	0	1.256667	0.4532475	0.5883466

```
expect_equal(
  t_4[t_4$enclosure == "L1", ]$cage_mass_mean_deviation_sd,
  abs_cage_mass_mean_deviation_sd_for_l1,
  tolerance = 1.0e-6
)
```

The values of `cage_mass_mean_deviation_sd` for enclosure `L1` are expected to be 1.1546303, 0.5662836, 0.5883466.

D1.5: Evaluation

The AI has been evaluated each step along the way. Regardless of this, here we evaluate the learning of the AI again, in a more visual way: it should reproduce a figure in the paper.

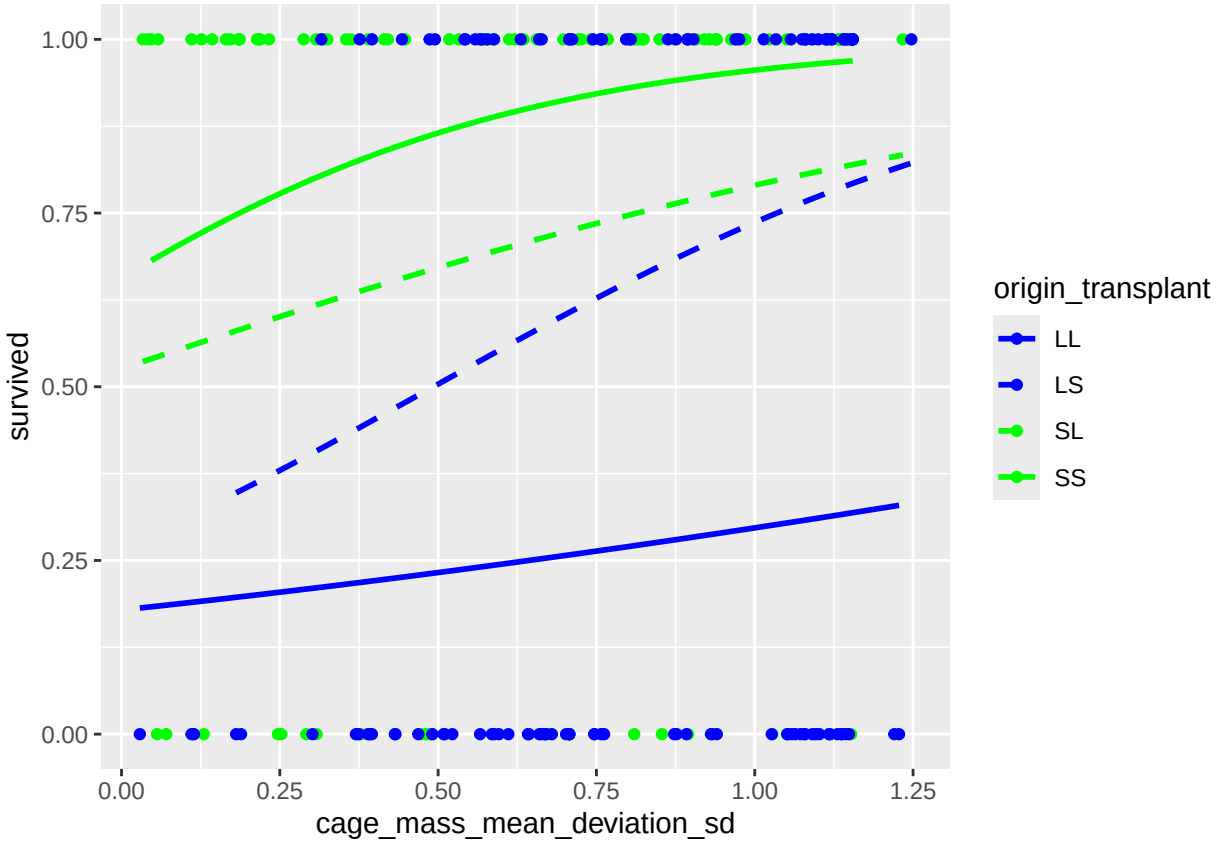
D1.5.1: reproduce the plot Ask the AI:

```
Create a scatter plot of this data:
- Each point is a fish.
- Use a blue color for fish
  that have 'Lake' as their origin.
- Use a green color for fish
  that have 'Stream' as their origin.
- On the X axis, put the values of `cage_mass_mean_deviation_sd`
  of each fish
- On the Y axis, put the `survived` of each fish.
- Put the fish in four categories:
  - Fish that have an `origin` of 'Lake' and
    a `transplant` of 'Lake' are in category 'LL'
  - Fish that have an `origin` of 'Lake' and
    a `transplant` of 'Stream' are in category 'LS'
  - Fish that have an `origin` of 'Stream' and
    a `transplant` of 'Lake' are in category 'SL'
  - Fish that have an `origin` of 'Stream' and
    a `transplant` of 'Stream' are in category 'SS'
- Show a trendline for each binomal fit on each of these 4 categories.
  - Use blue lines for categories that originate from a lake
  - Use green lines for categories that originate from a stream
```

```
  - Use solid lines for categories 'LL' and 'SS'
  - Use dashed lines for categories 'LS' and 'SL'
```

Expected is the plot in the paper:

```
t_4$origin_transplant <- paste0(
  stringr::str_sub(t_4$origin,1 ,1),
  stringr::str_sub(t_4$transplant,1 ,1)
)
ggplot2::ggplot(
  data = t_4,
  ggplot2::aes(
    x = cage_mass_mean_deviation_sd,
    y = survived,
    color = origin_transplant,
    lty = origin_transplant
  )
) + ggplot2::scale_color_manual(
    values = c("blue", "blue", "green", "green")
  ) +
  ggplot2::scale_linetype_manual(
    values = c("solid", "dashed", "dashed", "solid")
  ) +
  ggplot2::geom_point() +
  ggplot2::geom_smooth(
    method = "glm",
    method.args = list(family = "binomial"),
    se = FALSE
  )
#> `geom_smooth()` using formula = 'y ~ x'
```



D1.6: Synthesis

In these steps, we are combining the current knowledge with the paper.

Tell the AI:

```
The data you've been working on was used in a scientific paper.
The analyis you've done so far was used in the same scientific paper.
Below this paragraph is the abstract of that scientific paper.
Read it.

Two distinct forms of natural selection promote adaptive biological
diversity. Divergent selection occurs when different environments
favour different phenotypes, leading to increased differences
between populations. Negative frequency-dependent selection
occurs when rare variants within a population are favoured over
common ones, increasing diversity within populations. These
two diversifying forces promote genetic variation at different
spatial scales, and may act in opposition, but their relative effects
remain unclear because they are rarely measured concurrently.
Here we show that negative frequency-dependent selection within
populations can favor rare immigrants over locally adapted
residents. We reciprocally transplanted lake and stream ecotypes
of threespine stickleback into lake and stream habitats, while
manipulating the relative abundance of residents versus immigrants.
We found negative frequency-dependence: survival was highest
for the locally rare ecotype, rather than natives. Also, individuals
```

```
with locally rare major histocompatibility complex (MHC) class
IIb genotypes were infected by fewer parasites. This negative
frequency-dependent selection will tend to favour rare immigrants
over common residents, amplifying the effect of migration and
undermining the efficacy of divergent natural selection to drive
population differences. The only signal of divergent selection
was a tendency for foreign fish to have higher parasite loads than
residents, after controlling for MHC genotype rarity. Frequency-
dependent ecological interactions have long been thought to
promote speciation. Our results suggest a more nuanced view in
which negative frequency dependence alters the fate of migrants to
promote or constrain evolutionary divergence between populations.
```

D1.7: Reflection

Not applicable. A trained AI does not need to reflect on its learning.

D2. Q1: Ask if the conclusion is correct

Ask the AI:

```
The scientific paper describes a conclusion drawn from the data.
The conclusion is that the extreme body masses
are likelier to survive.
Do you judge this to be a valid conclusion based on the data?
```

D3 Alternative line of reasoning

D3.1 Analysis

D3.1.1 Add relative standarized body mass

Ask the AI:

```
To make a better comparison, we are going to standarize
body masses per enclosure. Add a column to the
data called `cage_mass_mean_deviation_sd_rel`.
Its values are calculated per fish.
Each fish its `cage_mass_mean_deviation_sd_rel`
equals the difference between its `pre_mass`
and its enclosure's `cage_mass_mean`,
divided by the `cage_mass_stdev` of its enclosure.
Could you show me the data for enclosure L1?
```

Expected:

```r
cage_mass_mean_deviation_sd_per_fish_rel <- t_4 |>
  dplyr::select(fishID, pre_mass, cage_mass_mean, cage_mass_stdev) |>
  dplyr::group_by(fishID) |>
  dplyr::mutate(
    cage_mass_mean_deviation_sd_rel =
      (pre_mass - cage_mass_mean) / cage_mass_stdev
  ) |>
  dplyr::select(fishID, cage_mass_mean_deviation_sd_rel)
expect_equal(
  nrow(cage_mass_mean_deviation_sd_per_fish_rel),
  n_fish_transplanted
)
```

```r
expect_equal(
  nrow(t_4),
  n_fish_transplanted
)
t_5 <- merge(t_4, cage_mass_mean_deviation_sd_per_fish_rel)
expect_equal(
  nrow(t_5),
  n_fish_transplanted
)

knitr::kable(head(t_5))
```

fishID	enclosure	origin	pre_mass	transplant	survived	cage_mass_mean	cage_mean_mass_cage_stdev	mass_mean_deviation_sd_rel	origin_transplant	mass_mean_deviation_sd
1	L1	Stream	1.78	Lake	1	1.256667	0.4532475	1.1546303	SL	1.1546303
2	L2	Stream	0.65	Lake	1	1.016667	0.5589574	0.6559832	SL	-0.6559832
3	L2	Stream	1.66	Lake	1	1.016667	0.5589574	1.1509524	SL	1.1509524
4	L3	Stream	2.26	Lake	1	1.453333	0.8309834	0.9707375	SL	0.9707375
5	L3	Stream	1.50	Lake	0	1.453333	0.8309834	0.0561584	SL	0.0561584
6	L4	Stream	1.89	Lake	1	1.723333	0.9016282	0.1848508	SL	0.1848508

```r
knitr::kable(t_5[t_5$enclosure == "L1", ])
```

	fishID	enclosure	origin	pre_mass	transplant	survived	cage_mass_mean	cage_mean_mass_cage_stdev	mass_mean_deviation_sd_rel	origin_transplant	mass_mean_deviation_sd
1	1	L1	Stream	1.78	Lake	1	1.256667	0.4532475	1.1546303	SL	1.1546303
181	181	L1	Lake	1.00	Lake	0	1.256667	0.4532475	0.5662836	LL	-0.5662836
182	182	L1	Lake	0.99	Lake	0	1.256667	0.4532475	0.5883466	LL	-0.5883466

```r
expect_equal(
  t_5[t_5$enclosure == "L1", ]$cage_mass_mean_deviation_sd_rel,
  cage_mass_mean_deviation_sd_for_l1,
  tolerance = 1.0e-6
)
```

D3.2 Evaluation

D3.2.1 Plot relative standarized body mass with same fit

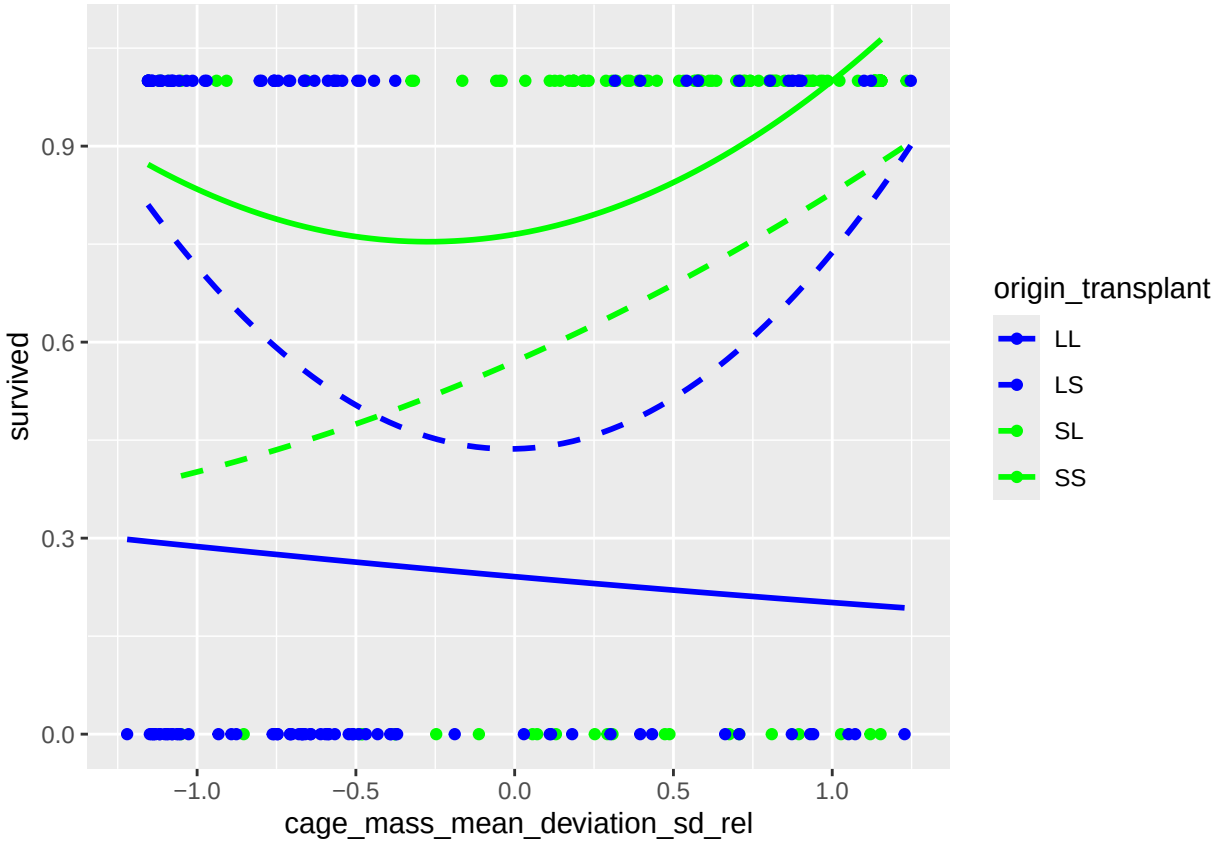
Ask the AI:

```
Create a scatter plot of this data:
- Each point is a fish.
- Use a blue color for fish
  that have 'Lake' as their origin.
- Use a green color for fish
  that have 'Stream' as their origin.
- On the X axis, put the values of `cage_mass_mean_deviation_sd_rel`
  of each fish
- On the Y axis, put the `survived` of each fish.
- Put the fish in four categories:
  - Fish that have an `origin` of 'Lake' and
    a `transplant` of 'Lake' are in category 'LL'
```

```
  - Fish that have an `origin` of 'Lake' and
     a `transplant` of 'Stream' are in category 'LS'
  - Fish that have an `origin` of 'Stream' and
     a `transplant` of 'Lake' are in category 'SL'
  - Fish that have an `origin` of 'Stream' and
     a `transplant` of 'Stream' are in category 'SS'
- Show a trendline for each parabolic fit on each of these 4 categories.
   - Use blue lines for categories that originate from a lake
   - Use green lines for categories that originate from a stream
   - Use solid lines for categories 'LL' and 'SS'
   - Use dashed lines for categories 'LS' and 'SL'
```

This should produce this plot:

```
ggplot2::ggplot(
  data = t_5,
  ggplot2::aes(
    x = cage_mass_mean_deviation_sd_rel,
    y = survived,
    color = origin_transplant,
    lty = origin_transplant
  )
) + ggplot2::scale_color_manual(
    values = c("blue", "blue", "green", "green")
  ) +
  ggplot2::scale_linetype_manual(
    values = c("solid", "dashed", "dashed", "solid")
  ) +
  ggplot2::geom_point() +
  ggplot2::geom_smooth(
    method = "lm",
    formula = y ~ x + I(x^2),
    se = FALSE
  )
```



The differences are:

- this plot uses relative body masses, instead of absolute
- the fit is parabolic, instead of binomial, as the latter is assumed to be monotonically increasing/decreasing. This means that a binomial distribution cannot be used to fit on data that is shaped like a `U` (as is assumed in the original paper: the extreme body masses have the highest fitness)

D3.3 Synthesis

In these steps, we are combining the current knowledge with the paper.

```
We have just analysed the relative normalized body masses.
Call this the alternative analysis.
```

D3.4 Reflection

Not applicable.

D4. Q2: Ask if the conclusion is correct

Ask the AI:

```
The scientific paper describes a conclusion drawn from the data.
The conclusion is that the extreme body masses
are likelier to survive.
Do you judge this to be a valid conclusion based on the data?

Take into account the insights from the alternative analysis.
```