Assignment 2: Analyze a Linear Model

Penny Kahn


```
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(DT))
suppressPackageStartupMessages(library(here))
suppressPackageStartupMessages(library(MuMIn))
suppressPackageStartupMessages(library(cowplot))
```

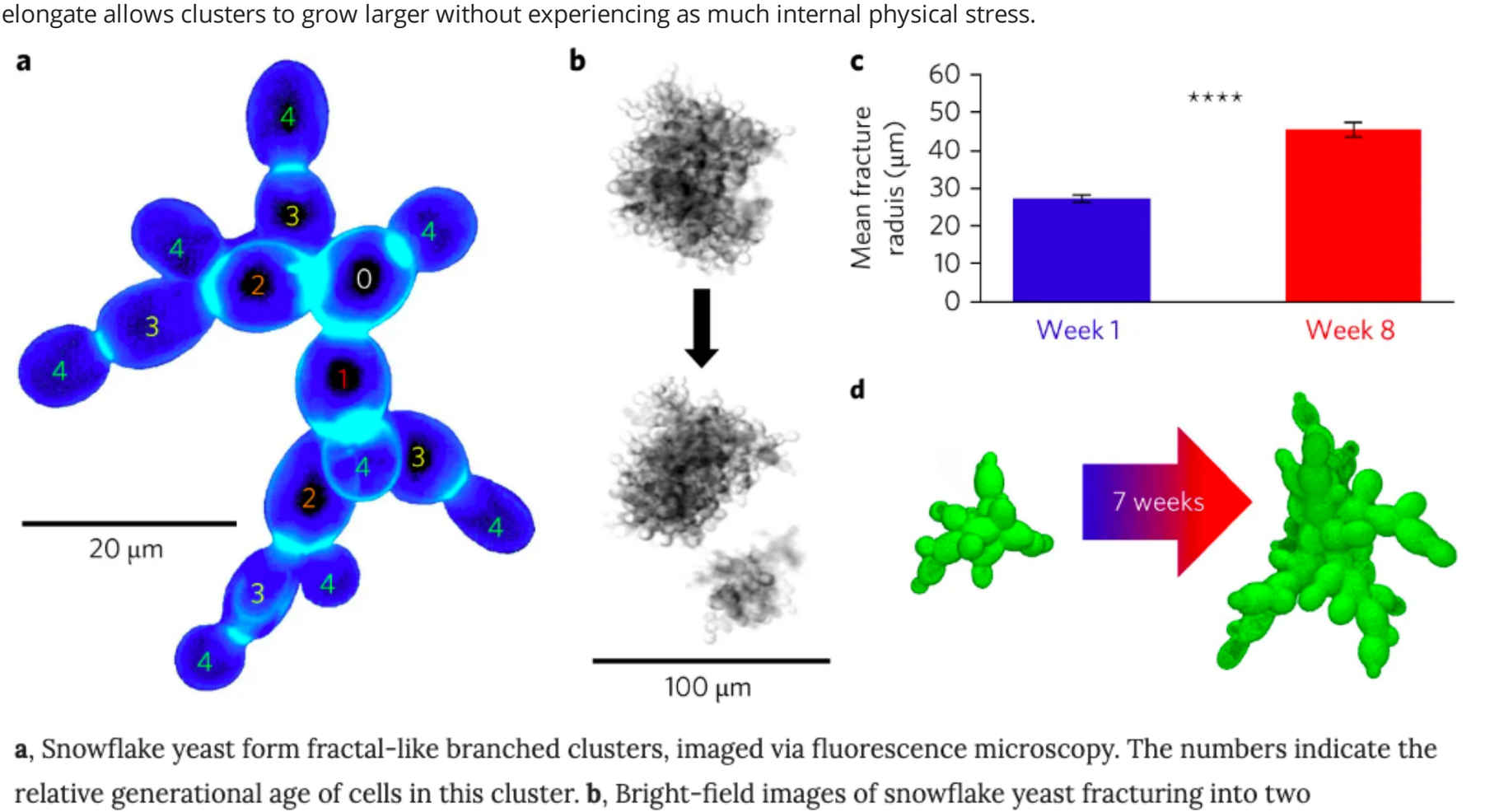
Introduction

Jacobeen, S., Pentz, J.T., Graba, E.C., Brandys, C.G., Ratcliff, W.C., Yunker, P.J. 2018. Cellular packing, mechanical stress and the evolution of multicellularity. *Nature Physics* 14, 286–290.

Link to paper on nature.com

The paper I have chosen is from a group at the Georgia Institute of Technology that studies the evolution of multicellularity and multicellular complexity using a yeast model. They have experimentally evolved several independent populations of multicellular yeast (aka "snowflake yeast") from a commonly used unicellular lab strain. Their experimental evolution method involves selecting individual units (i.e. clusters) that settle out of liquid media most rapidly, and consequently only the largest clusters survive each selection event. Over many generations of this selection regime, cluster size has increased despite physical challenges associated with cellular packing and mechanical stress.

The study I have chosen invesitgates the factors that enable this system to overcome physical challenges which constrain cluster size. For example, as the cluster is growing, cells in the center are dividing and become overcrowded. The cells push against each other and cause the cluster to fracture. The authors have found that one mechanism for growing larger clusters by avoiding fracture is to increase cell size. Cluster size scales with cell size, so there are fewer cell divisions needed to achieve a large size. Another way to pack tightly is to increase cell aspect ratio - hot dog shapes are easier to pack tightly than spheres. Making cells larger and more elongate allows clusters to grow larger without experiencing as much internal physical stress.



a, Snowflake yeast form fractal-like branched clusters, imaged via fluorescence microscopy. The numbers indicate the relative generational age of cells in this cluster. **b**, Bright-field images of snowflake yeast fracturing into two independently viable clusters. **c**, Over seven weeks (~291 generations) of selection for large size, snowflake yeast clusters increase their average maximum radius by a factor of 1.7. **d**, Three-dimensional confocal images show that week-8 snowflake yeast (right) contain a greater number of larger, more elongate cells than week-1 (left). The error bars in **c** denote one standard error of the mean; $^{****}P < 0.0001$.

Describe the Data

The data I will use are from a plot showing the relationship between number of cells and cluster size at two time points in the evolution experiment. The response variable is number of cells in a cluster (continuous, fixed), and the explanatory variables are cluster size (continuous, fixed) and time point (categorical, fixed). Each individual cluster has a measurement for size (radius, μm) and number of cells, as well as a time point (either Week 1 or Week 8). All individuals were measured from the same population which experienced daily selection events for large size. Let's look at the data to get an idea of the patterns.

First I'll read the data into a table called "snowflake" using the here() function from the "here" package. This allows the code to work for someone who has cloned the whole R project to a directory of their choosing anywhere on their local system.

```
snowflake <- read_csv(here::here("Assignment_2", "mod_attempt2.csv"))
```

```
## Parsed with column specification:
## cols(
##   radius = col_double(),
##   no_cells = col_double(),
##   week = col_character()
## )
```

Let's take a look at the data using the datatable() function from the "DT" package. This prints a nice looking, interactive table.

```
datatable(snowflake)
```

Show 10 entries Search:

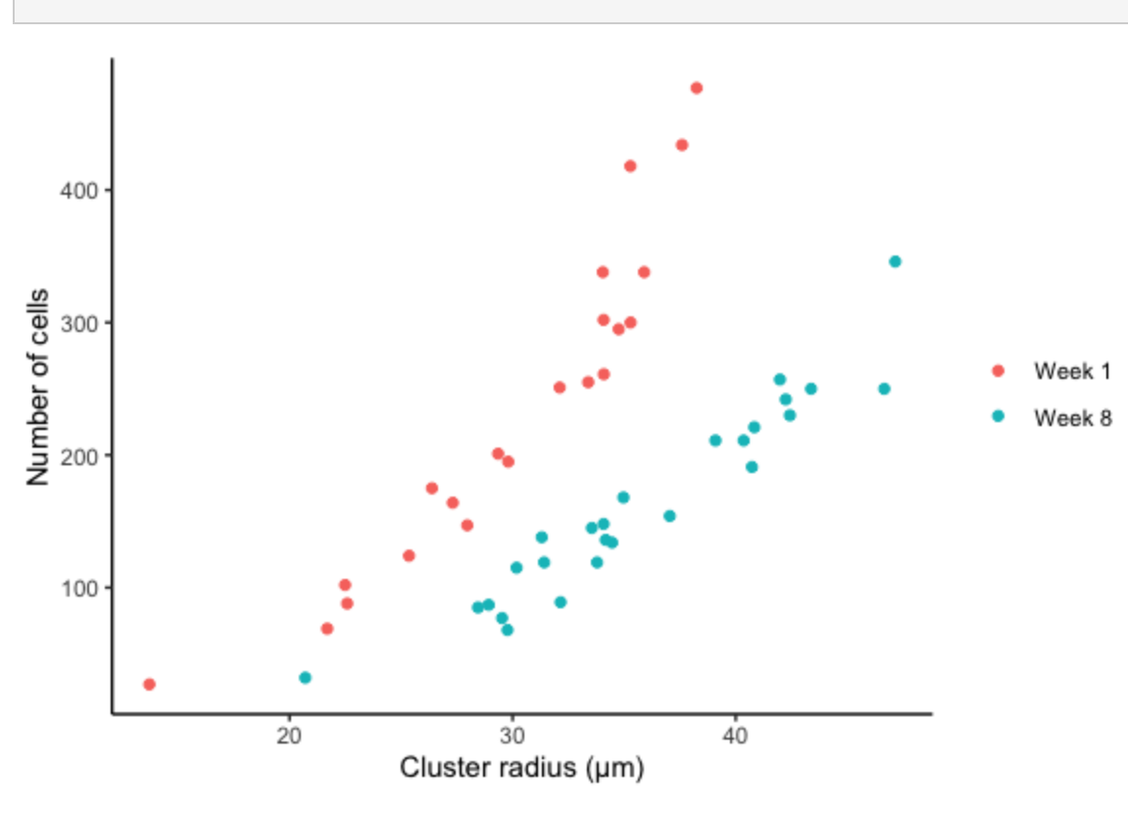
	radius	no_cells	week
1	13.71572611	27	Week 1
2	21.69013628	69	Week 1
3	22.48987348	102	Week 1
4	22.58168368	88	Week 1
5	25.35716802	124	Week 1
6	26.38453564	175	Week 1
7	27.31516746	164	Week 1
8	27.9658254	147	Week 1
9	29.34870614	201	Week 1
10	29.80074802	195	Week 1

Showing 1 to 10 of 47 entries Previous 1 2 3 4 5 Next

We'll visualize it further with a scatter plot. The different colors indicate time point.

```
plot <- snowflake %>%
  ggplot(aes(x = radius, y = no_cells)) +
  geom_point(aes(color = week)) +
  labs(x = "Cluster radius (μm)", y = "Number of cells") +
  theme_classic() +
  theme(legend.title = element_blank())

plot
```



Patterns

The first (perhaps obvious) pattern is that as cluster size increases, so does the number of cells within a cluster. This makes sense because cell division (and failure to separate mother and daughter) is how an individual grows. A more meaningful pattern we can see from this graph is that the distribution has shifted toward a larger cluster size from week 1 to week 8, indicating that cluster size has in fact increased over the course of the evolution experiment. Most importantly for the context of the study, we see that for each cluster radius, the week 8 individuals have a lower number of cells than the week 1 individuals. Remember, this is because the cells themselves are increasing in size and aspect ratio.

Parameters

A linear model fit to these data will provide us with estimates for slope and intercepts of lines that fit the data depending on the settings of the model.

- For a model with one explanatory and one response variable, we will receive an intercept and slope.
- For a model with two explanatory variables we will receive an intercept and a slope for one line, and a value for the difference in intercept of another line. So we will get two parallel lines with different intercepts.
- For a model with an interaction between two explanatory variables, we will receive an intercept and slope for one line, and values for the differences in the intercept and slope of the other line. So we will get two lines with different slopes and intercepts.
- If we ask for a model with a second degree polynomial (y = b0 + b1X + b2X^2), we will receive an estimate for the intercept (b0), the coefficient for the first term (b1), and the coefficient for the second term (b2). If we include an interaction with another explanatory variable, we will also receive all the estimates for differences in those three parameters for the second line.

Hypotheses

1. There is a positive linear relationship between size and cell number in general.
2. The group means at each time point are actually different from one another (there should be a different intercept for each time point).
3. There is an interaction between time point and cluster size (there should be a different slope and intercept for each time point).
4. Adding a second degree polynomial factor will fit the data better than a straight line.

Fit a Linear Model

I don't think this is necessary for the assignment, but I'm going to fit a few models (instead of just one) with increasing complexity to show how different models fit the data. I think it's useful to show how each added term affects the model, and discuss the changing implications for interpretation of the data.

The simplest linear model

This first model will be the simplest. It will look at the overall relationship between cluster size (explanatory) and number of cells in a cluster (response). The output we get from this model will give us estimates for slope and intercept of one line (y = b0 + b1*X) that fits the entire dataset.

```
mod_1 <- lm(no_cells ~ radius, data = snowflake)
summary(mod_1)
```

```
##
## Call:
## lm(formula = no_cells ~ radius, data = snowflake)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -96.25 -57.46 -39.16  43.86 233.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -123.364     59.832  -2.062    0.045 *
## radius         9.599      1.765   5.439 2.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 82.55 on 45 degrees of freedom
## Multiple R-squared:  0.3966, Adjusted R-squared:  0.3832
## F-statistic: 29.58 on 1 and 45 DF,  p-value: 2.106e-06
```

The (Intercept) estimate gives the intercept of the line, but it's not very biologically relevant as it is not possible to have a cluster size of 0 μm. In fact a single yeast cell has a radius of about 2.5 to 5 μm. This value determines the line's position on the y-axis. The model picks a position that will minimize residuals. In this case, for the purpose of the model, when cluster radius is 0 the number of cells is -123.364.

The "radius" estimate gives us the slope of the line. In this simplest model, with every 1 μm increase in cluster radius, there is a cell number increase of 9.599. The sign of the value demonstrates a positive relationship between cluster size and number of cells in a cluster, which again should be obvious.

I'll perform an ANOVA to test our first model against a null hypothesis of no relationship.

```
anova(mod_1)
```

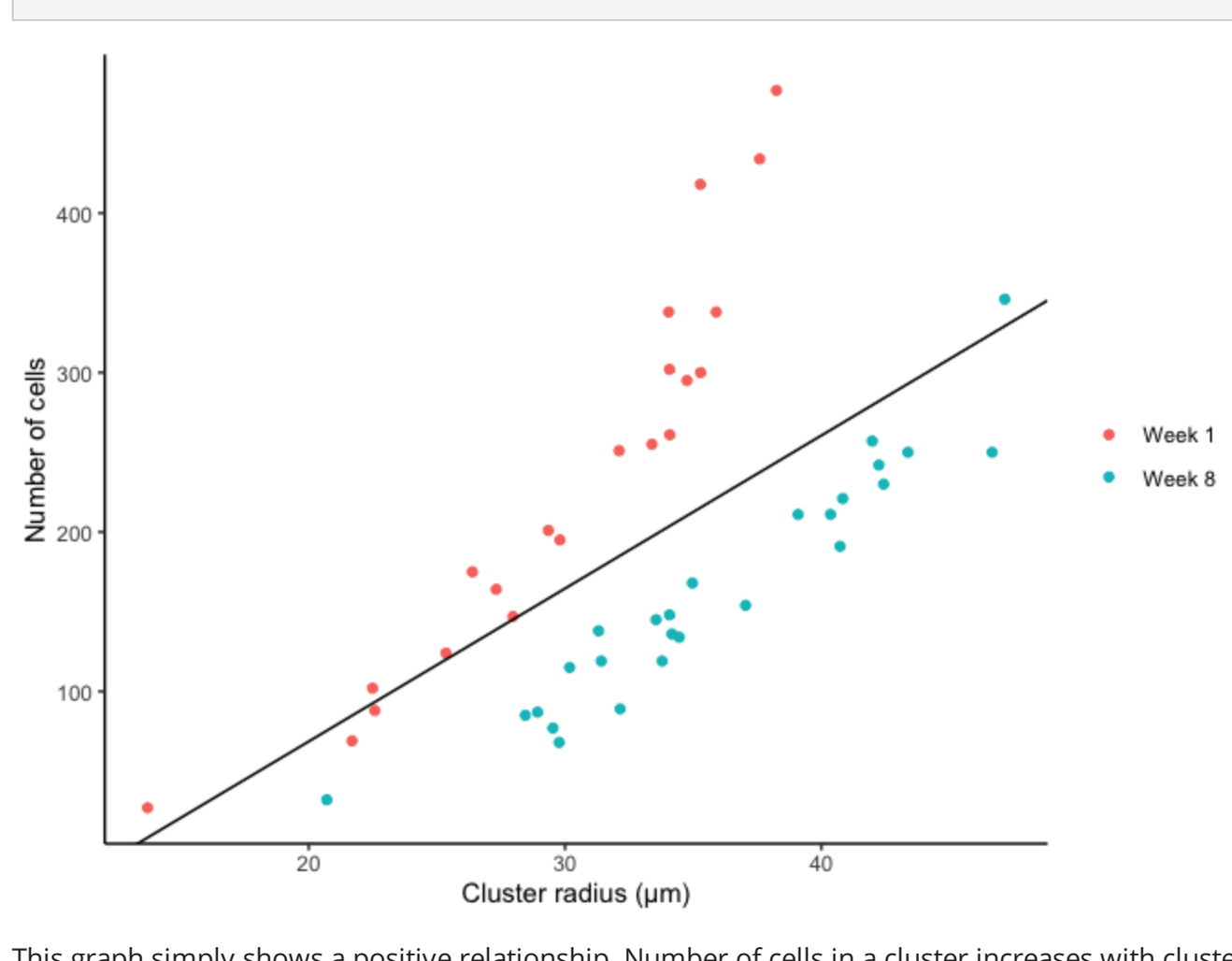
```
## Analysis of Variance Table
##
## Response: no_cells
##           Df Sum Sq Mean Sq F value    Pr(>F)
## radius     1 201593  201593  29.583 2.106e-06 ***
## Residuals 45 306654    6815
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This ANOVA provides us with a p-value for comparing the model we have created (mod1) with a null model which assumes no positive or negative relationship between the variables. The null model tries to fit the data to the average response. The ANOVA yields a p-value of 2.106e-06, so based on p-value alone our simple model fits the data better than the null model. We can conclude there is a relationship between cluster size and number of cells.

Now let's visualize the fit of the model to the data. I'll use the geom_abline() function to easily specify intercept and slope, which I have taken from the model output.

```
plot1 <- plot +
  geom_abline(intercept = -123.364, slope = 9.599)

plot1
```



This graph simply shows a positive relationship. Number of cells in a cluster increases with cluster radius. We can see the line doesn't fit the data perfectly. It mostly goes down the middle of two groups: samples from week 1 and week 8. I think it's easy to see that there should be separate lines for each time point.

Adding time point variable

Now I'll add in the categorical explanatory variable of time point. We should still see a positive relationship for both groups within the explanatory variable "week", but they will have different intercepts, and therefore different positions in graphical space.

```
mod_2 <- lm(no_cells ~ radius + week, data = snowflake)
summary(mod_2)
```

```
##
## Call:
## lm(formula = no_cells ~ radius + week, data = snowflake)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -68.16 -26.56  -9.20  18.48 124.06
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -192.6535    30.7060  -6.274 1.33e-07 ***
## radius        14.2644     0.9757  14.620  < 2e-16 ***
## weekWeek 8  -154.7785    13.3895 -11.560 6.36e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.55 on 44 degrees of freedom
## Multiple R-squared:  0.8505, Adjusted R-squared:  0.8437
## F-statistic: 125.2 on 2 and 44 DF,  p-value: < 2.2e-16
```

Now we have three coefficient estimates. In the order given in the output, we have the intercept for week 1, the slope for both weeks, and a difference of intercept for week 8. We can use the values from this output to make two regression equations - one for each week. The intercepts for the lines are -192.6535 for week 1 and (-192.6535 - 154.7785) or -347.432 for week 8. This estimates that for a given cluster radius there will be ~155 cells fewer in week 8 than week 1.

We see that the slope has increased from the first model (14.2644 > 9.599). Mod2 predicts that for a 1 μm increase in cluster radius, there is a cell number increase of 9.599. This increase in slope is because the relationship in the first model was being obscured by the difference in distribution along the x-axis. Also, the adjusted r-squared value has increased from the first to the second model (0.8437 > 0.3832) indicating a tighter fit of the data to the lines.

I'll test if the second model which includes the explanatory variable of "week" is significantly different from the first using an ANOVA again.

```
anova(mod_2, mod_1)
```

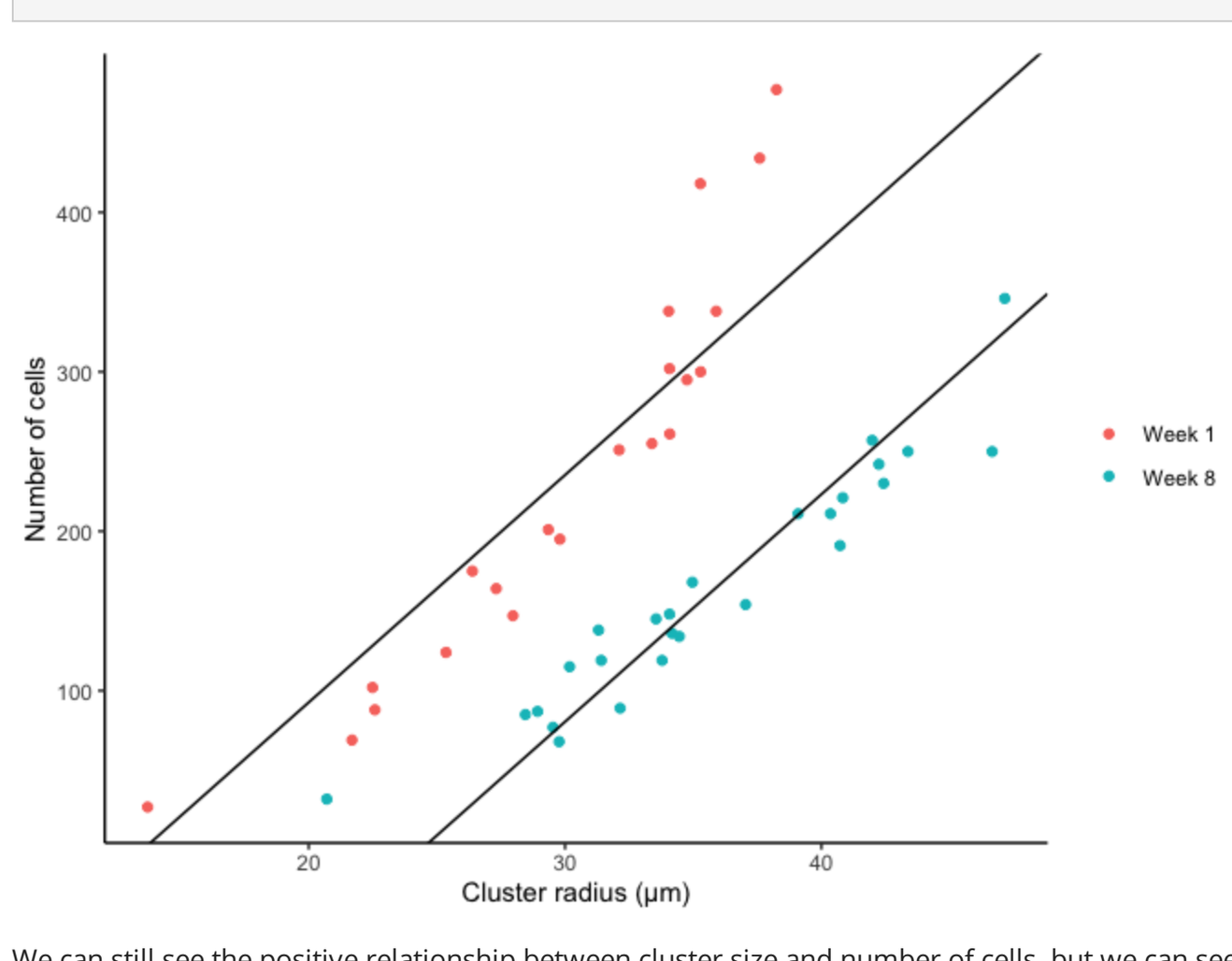
```
## Analysis of Variance Table
##
## Model 1: no_cells ~ radius + week
## Model 2: no_cells ~ radius
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     44  75962
## 2     45 306654 -1   -230692 133.63 6.358e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This ANOVA is comparing a model that includes time point as an explanatory variable as well as cluster size to the first model which only includes cluster size. Since we have a significant p-value we can conclude that there is a difference in group mean between the two weeks, and the model we choose should reflect that. In other words, using week and cluster size to predict number of cells in a cluster will yield a more accurate prediction than using cluster size alone.

We'll visualize the model fit to the data using the same method as before. I'm adding the first and third coefficient estimates for for the intercept of the second line.

```
plot2 <- plot +
  geom_abline(intercept = -192.6535, slope = 14.2644) +
  geom_abline(intercept = (-192.6535 - 154.7785), slope = 14.2644)

plot2
```



We can still see the positive relationship between cluster size and number of cells, but we can see that over time (from week 1 to week 8) the distribution has shifted right on the x-axis and down on the y-axis, that is, the cluster size has increased overall with a fewer number of cells at each size.

These lines appear to fit the data better than a single line for both weeks, but it might be better to have different slopes for each week, so I'll add an interaction term between week and cluster size.

Interaction component

I'll add an interaction between week and radius by putting an asterisk between them instead of a plus sign.

```
mod_3 <- lm(no_cells ~ radius * week, data = snowflake)
summary(mod_3)
```

```
##
## Call:
## lm(formula = no_cells ~ radius * week, data = snowflake)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.736 -22.746  -5.028   9.483  90.868
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -314.641     38.029  -8.274 1.95e-10 ***
## radius              18.322      1.239  14.787  < 2e-16 ***
## weekWeek 8          80.791     55.040   1.468    0.149
## radius:weekWeek 8   -7.235      1.655  -4.373 7.66e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.97 on 43 degrees of freedom
## Multiple R-squared:  0.8965, Adjusted R-squared:  0.8893
## F-statistic: 124.2 on 3 and 43 DF,  p-value: < 2.2e-16
```

In the output of the interaction model we have four coefficient estimates. In order they are: the intercept of the week 1 line, the slope of the week 1 line, the difference in intercept of the week 8 line, and the difference in slope of the week 8 line.

Since the slope is different for each line, the difference in intercept is less biologically interpretable on its own. The models estimate that at a cluster size of 0 μm, the number of cells will be -314.641 in week 1 (meaningless) and (-314.641 + 80.791) or -233.85 in week 8 (also meaningless). Since the slopes are different, this difference in cell number between the two weeks at a given cluster size will change as a function of the slopes.

The interpretation of the slopes is more biologically relevant. In week 1, with every cluster size increase of 1 μm, there will be 18.322 more cells. In week 8, with an increase of 1 μm there will be (18.322 - 7.235) or 11.087 more cells. Fewer cells are needed to generate the same size increase.

Again we can see that the R-squared value has increased from mod2 (0.8437) to mod3 (0.8893).

Now we'll use an ANOVA to test if week and cluster size interact significantly.

```
anova(mod_3, mod_2)
```

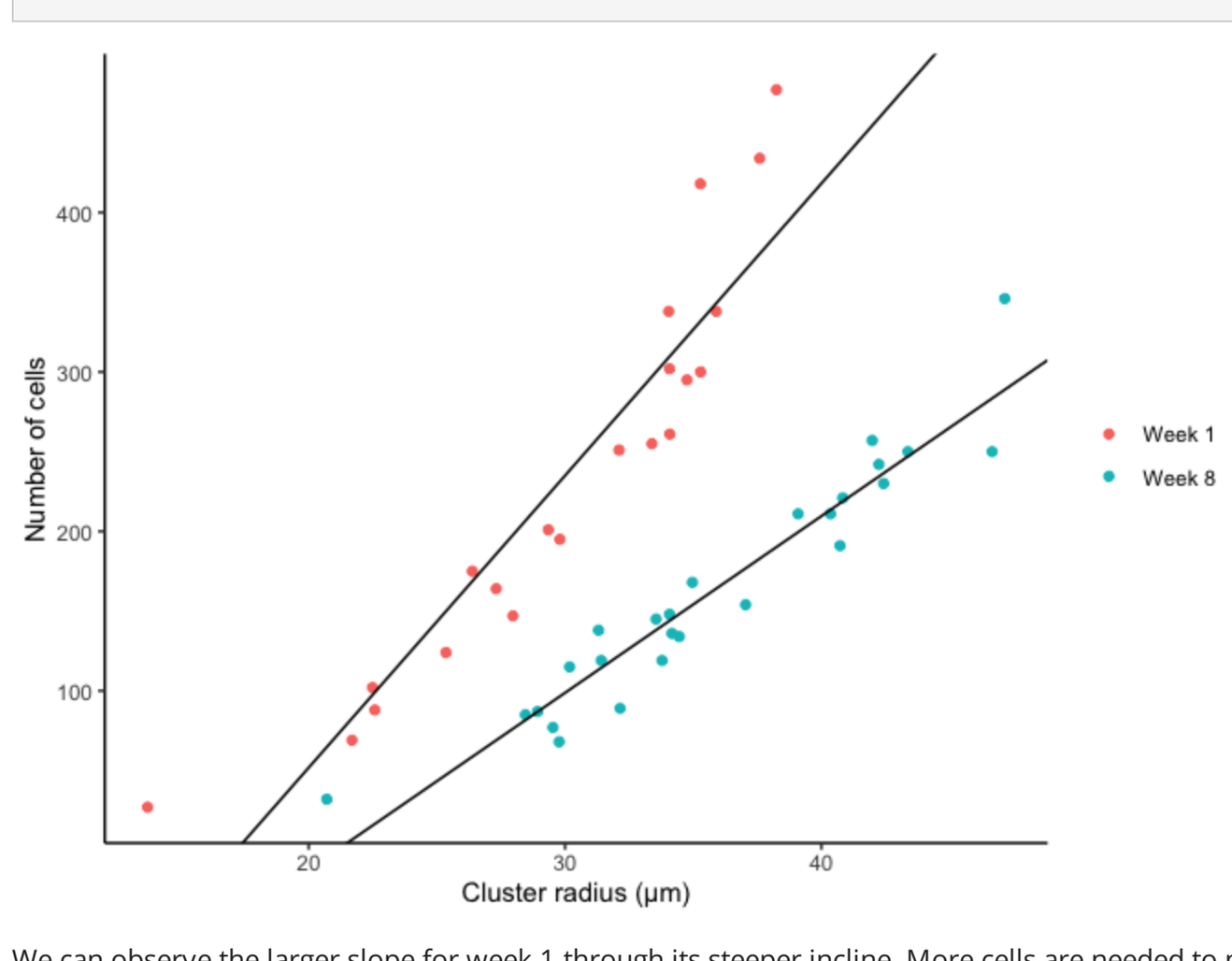
```
## Analysis of Variance Table
##
## Model 1: no_cells ~ radius * week
## Model 2: no_cells ~ radius + week
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     43 52582
## 2     44 75962 -1    -23380 19.119 7.659e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This significant p-value indicates that there is an interaction with week and there should be different slopes for each line. The trends for how clusters grow at each time point are significantly different. We can more accurately predict number of cells per cluster by using different slopes and intercepts for each week instead of just having two parallel lines.

To add the lines to this plot, I'm using geom_abline() again, but now I also add the difference between slopes to the second line.

```
plot3 <- plot +
  geom_abline(intercept = -314.641, slope = 18.322) +
  geom_abline(intercept = (-314.641 + 80.791), slope = (18.322 - 7.235))

plot3
```



We can observe the larger slope for week 1 through its steeper incline. More cells are needed to produce the same size increase in week 1 than in week 8. In week 8 cells are larger, so each addition of a cell will provide more volume to the cluster than a tiny week 1 cell will.

Polynomial factor

The data don't look very linear because the nature of cluster growth is not linear. With each generation time (i.e. every time a yeast cell divides) every cell within the cluster gets a new daughter (the ones on the inside and the ones on the outside). The larger a cluster grows, the less effect one division cycle has on the size of the overall cluster, which we can see especially well in the week 1 sample. Even as number of cells increases from ~250 to almost 500, there is little increase in cluster size. So I think a model that includes a second degree polynomial will fit the data better than straight lines.

```
mod_4 <- lm(no_cells ~ poly(radius, 2, raw = TRUE) * week, data = snowflake)
summary(mod_4)
```

```
##
## Call:
## lm(formula = no_cells ~ poly(radius, 2, raw = TRUE) * week, data = snowflake)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -49.01 -15.49   0.25  14.75  74.16
##
## Coefficients:
##                                         Estimate Std. Error t value
## (Intercept)                             197.9767    93.6365   2.114
## poly(radius, 2, raw = TRUE)1            -21.4283     6.9801  -3.070
## poly(radius, 2, raw = TRUE)2              0.7246     0.1261   5.746
## weekWeek 8                             -210.8983   160.9878  -1.310
## poly(radius, 2, raw = TRUE)1:weekWeek 8  19.6616    10.2200   1.924
## poly(radius, 2, raw = TRUE)2:weekWeek 8  -0.5435     0.1638  -3.317
##                                        Pr(>|t|)
## (Intercept)                             0.04061 *
## poly(radius, 2, raw = TRUE)1            0.00379 **
## poly(radius, 2, raw = TRUE)2           9.95e-07 ***
## weekWeek 8                              0.19748
## poly(radius, 2, raw = TRUE)1:weekWeek 8  0.06134 .
## poly(radius, 2, raw = TRUE)2:weekWeek 8  0.00191 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.13 on 41 degrees of freedom
## Multiple R-squared:  0.9449, Adjusted R-squared:  0.9382
## F-statistic: 140.7 on 5 and 41 DF,  p-value: < 2.2e-16
```

There's a lot of coefficient estimates in this one. Basically, there's an estimate for b0, b1, and b2 for week 1 as well as the differences in all those coefficients for the week 8 function. And those get plugged into the quadratic function y = b0 + b1X + b2X^2. I was having a hard time interpreting the biological significance of polynomial parameter estimates, but then I found this from Stimson, J.A., Carmines, E.G., Zeller, R.A. 1978. Interpreting Polynomial Regression. *Sociological Methods & Research* 6, 515-524:

> *"What meaning can be attributed to the individual coefficients in polynomial regression equations? Unfortunately, these coefficients cannot be easily or readily interpreted, partly because they are noncomparable. For example, by definition Bl and B2 measure the change in Y associated with each unit change of X or X2, respectively, controlling for the effects of the other."*

So I'm going to leave it at that. Just as a note, I'll mention that the positive sign of the coefficient b2 means the graph is convex (u-shaped). b2 also tells us about the steepness of the curvature. Since b2 is larger in the week 1 function, the u shape will be steeper than in week 8. Apparently b1 gives the rate of change when x is equal to zero.

Anyway… let's move onto the ANOVA to test if this quadratic function is a better model than straight lines.

```
anova(mod_4, mod_3)
```

```
## Analysis of Variance Table
##
## Model 1: no_cells ~ poly(radius, 2, raw = TRUE) * week
## Model 2: no_cells ~ radius * week
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     41 27994
## 2     43 52582 -2    -24588 18.006 2.441e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

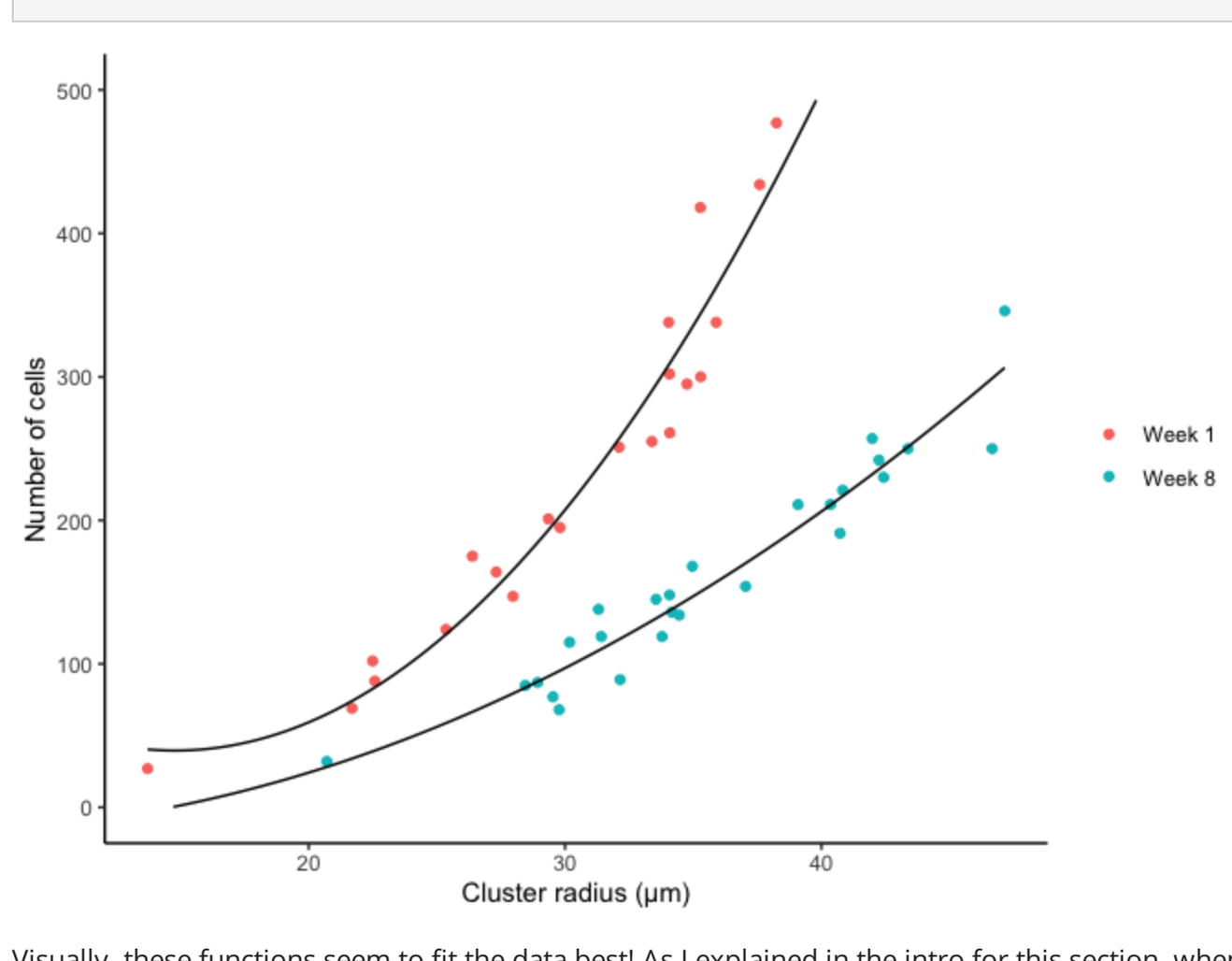
Indeed, the p-value is significant. The polynomial model fits the data data significantly better than one that produces linear equations.

Here I'm writing the two functions, one for each week, using the coefficient estimates from the summary of mod4. I'll use those set functions to graph their lines in ggplot. I'm limiting the y-axis to 500 because otherwise it shows the line up to y = 800 even though there are no data associated with that range.

```
poly1 <- function(x) 197.9767 - 21.4283 * x + 0.7246 * x^2
poly2 <- function(x) (197.9767 - 210.8983) + (-21.4283 + 19.6616) * x + (0.7246 - 0.5435) * x^2
```

```
plot4 <- plot +
  stat_function(method = lm, fun = poly1) +
  stat_function(method = lm, fun = poly2) +
  ylim(0, 500)

plot4
```

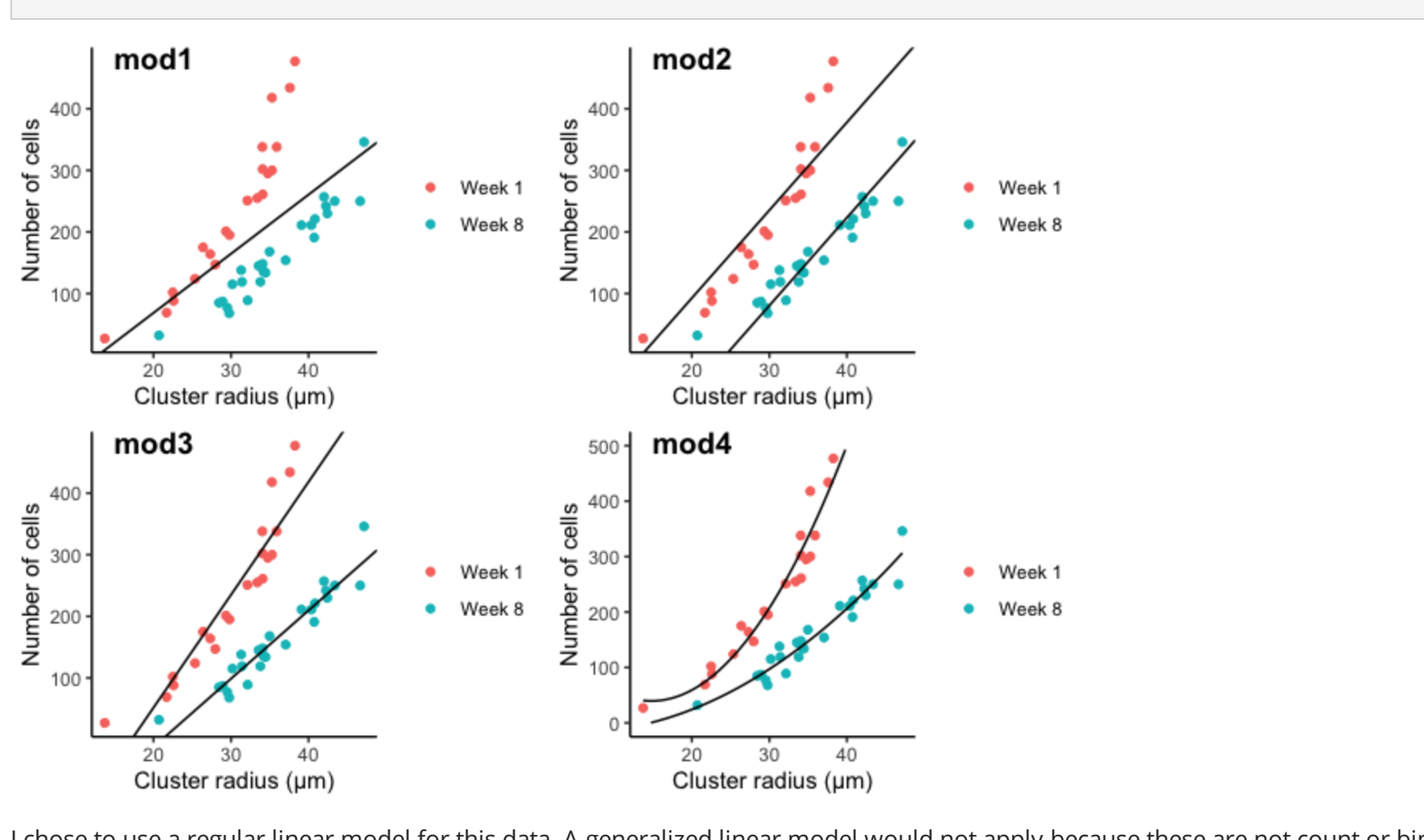


Visually, these functions seem to fit the data best! As I explained in the intro for this section, when a cluster is very small, the addition of a single cells or a small number of cells can have a big impact on cluster size. But when a cluster is large, new cells are more concentrated in the interior of the cluster, becoming more tightly packed, and not contributing as much to cluster size. Since cells are becoming more elongate over the course of the evolution experiment, they can pack more easily, feel less internal stress, and don't reach a growth block as quickly as the clusters from week 1.

Conclusions

Let's look at all the models side-by-side for easy comparison. I am doing this with the plot_grid() function from the "cowplot" package.

```
plot_grid(plot1, plot2, plot3, plot4, labels = c("mod1", "mod2", "mod3", "mod4"), hjust = -1.3)
```



I chose to use a regular linear model for this data. A generalized linear model would not apply because these are not count or binary data. There are also no random effects because each data point represents an individual from the same population, and different individuals are measured at each time point. All the variables are accounted for, and none of them produce a random effect.

Overall conclusions

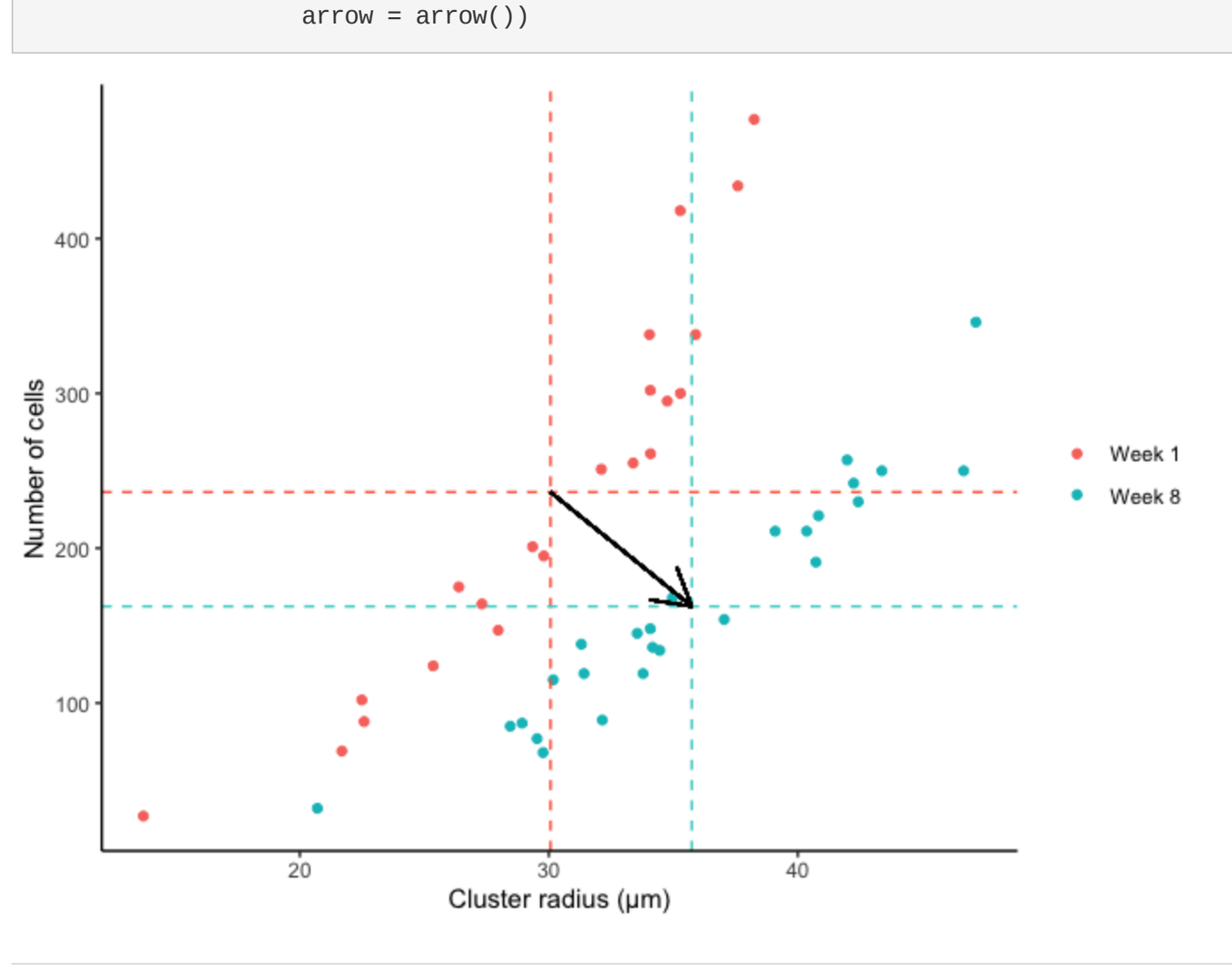
Internal mechanical stress is a physical challenge for growing large size in clonal cellular clusters. The evolution of snowflake yeast size has been experimentally manipulated by imposing daily selection for large size. Over time, snowflake yeast have been able to overcome certain physical challenges to prevent cluster fracture while maintaining large cluster size. They do so by increasing cell size within the cluster as well as making cells more elongate to reduce internal stress.

When we observe the characteristics of clusters from early and later in the evolution experiment we notice several patterns. By week 8 of the experiment clusters have gotten significantly larger, which we proved by showing that mod2 was significantly different from mod1. Additionally, week 8 clusters require fewer cells to grow the same amount as a week 1 cluster because each cell adds significantly more volume. We showed this by testing mod3 against mod 2. Finally, because cells are becoming more elongate over time, the internal stress is reduced and the week 8 clusters can continue to add more cells and grow even larger without being constrained to a certain size like the week 1 clusters.

Here is a graph with the mean number of cells and cluster size marked with a dashed line for each group (time point). This clearly demonstrates that although cluster size is increasing, number of cells per cluster is decreasing, and we can attribute this to the adaptations of larger and more elongate cells.

```
week1 <- snowflake %>%
  filter(week == "Week 1")
week8 <- snowflake %>%
  filter(week == "Week 8")

plot +
  geom_vline(xintercept = mean(week1$radius), linetype="dashed", color = "tomato") +
  geom_vline(xintercept = mean(week8$radius), linetype="dashed", color = "mediumturquoise") +
  geom_hline(yintercept = mean(week1$no_cells), linetype="dashed", color = "tomato") +
  geom_hline(yintercept = mean(week8$no_cells), linetype="dashed", color = "mediumturquoise") +
  geom_segment(aes(x = mean(week1$radius), y = mean(week1$no_cells),
                   xend = mean(week8$radius), yend = mean(week8$no_cells)),
                   arrow = arrow())
```



I made four progressively more complex models, and the last, most complex one turned out to be the best according to the p-values. Higher complexity doesn't always mean a better model according the Akaike Information Criterion, which penalizes model complexity in balance with rewarding good model fit. Just for fun I'll perform some data dredging using the dredge() function from the "MuMIn" package. It will give us the AIC for each possible model within the parameter space I set.

```
full <- lm(no_cells ~(.)^2, data = snowflake, na.action = na.fail)
dredge <- dredge(full, rank = "AICc")
```

```
## Fixed term is "(Intercept)"
```

```
dredge
```

```
## Global model call: lm(formula = no_cells ~ (.)^2, data = snowflake, na.action = na.fail)
## ---
## Model selection table
##     (Int)    rds wek rds:wek df   logLik  AICc delta weight
## 8 -314.6 18.320   +       +  5 -231.660 474.8  0.00  0.999
## 4 -192.7 14.260   +          4 -240.304 489.6 14.78  0.001
## 2 -123.4  9.599              3 -273.098 552.8 77.97  0.000
## 3  236.2          +          3 -281.846 570.3 95.47  0.000
## 1  195.4                     2 -284.972 574.2 99.43  0.000
## Models ranked by AICc(x)
```

In fact, the model with the most estimated terms (df = 5) has the lowest AIC and highest weight. Our analysis is supported!