# Friday, Feb 25

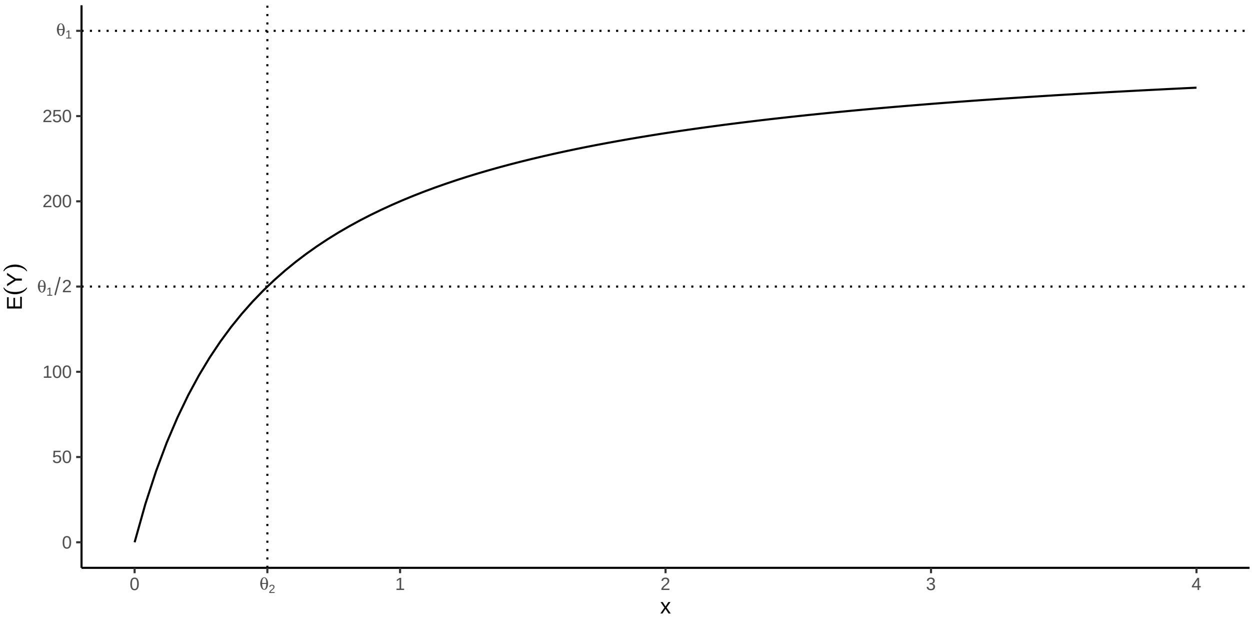
## The Michaelis-Menten Model

The Michaelis-Menten model is perhaps the quintessential example of an application of nonlinear regression. It is from biochemistry and concerns the relationship between the (expected) rate of an enzymatic reaction to the concentration of an enzymatic substrate (i.e., the material of the reaction). As a nonlinear regression model the Michaelis-Menten model can be written as

$$E(Y) = \frac{\theta_1 x}{\theta_2 + x},$$

where $Y$ is the reaction rate and $x$ is the substrate concentration.[1]

The two parameters of this model, $\theta_1$ and $\theta_2$, are interpretable in terms of the relationship between the expected reaction rate and substrate concentration. The $\theta_1$ parameter is the maximum expected reaction rate (i.e., the upper asymptote as $x \to \infty$), and $\theta_2$ is the value of $x$ at which the reaction rate is half of $\theta_1$ (i.e., a "half-life" parameter) so smaller values of $\theta_2$ mean that the curve is approaching $\theta_1$ "faster" as $x$ increases.[2] Note also that if $x = 0$ then $E(Y) = 0$ so the curve is constrained to have an "intercept" of zero, which makes sense in the context of enzyme kinetics. The plot below shows an example of the model where $\theta_1 = 300$ and $\theta_2 = 0.5$.



To estimate $\theta_1$ and $\theta_2$ the typical method is to conduct a series of assays, varying substrate concentration and recording the reaction rate at each concentration, and then using nonlinear regression to estimate $\theta_1$ and $\theta_2$. In the following problems you will be using data in the data frame `Puromycin`. It is included with R so there is no package to load. These data are from a study that observed reaction rates at several substrate

[1] See the Wikipedia entry on Michaelis-Menten for details if you are interested. A related model is the Beverton-Holt population dynamics model that is frequently used in fisheries research.

[2] The interpretation of $\theta_1$ can be seen by taking the limit of $\theta_1 x/(\theta_2 + x)$ as $x \to \infty$, and the interpretation of $\theta_2$ can be shown by solving $\theta_1/2 = \theta_1 x/(\theta_2 + x)$ for $x$.

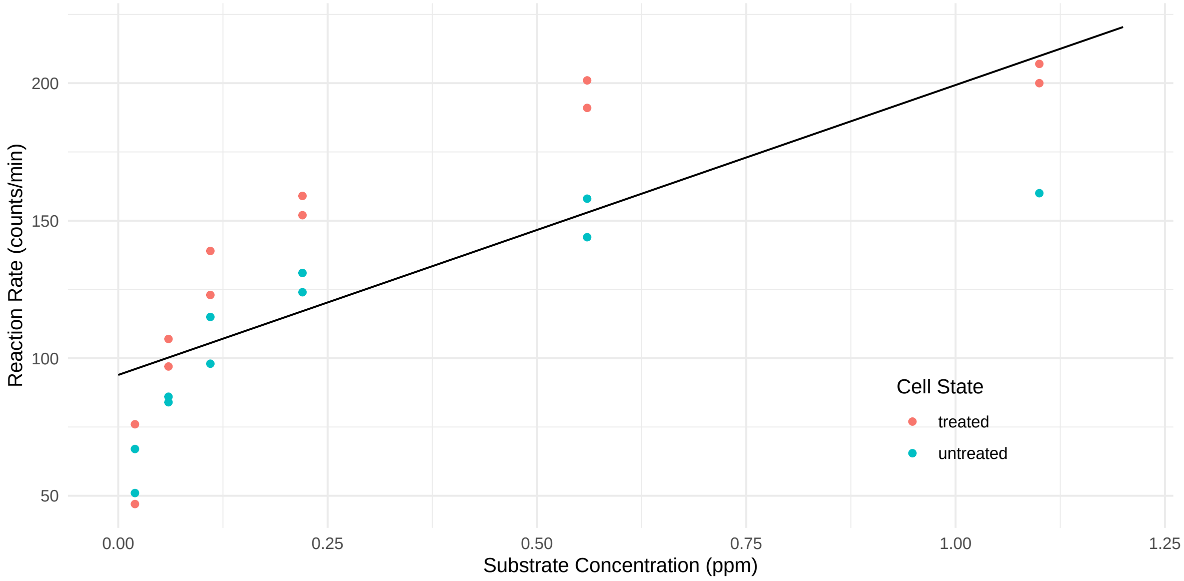
concentrations, but also for cells that were treated with puromycin (an antibiotic). Before starting you can familiarize yourself with the data by simply typing `Puromycin` at the console prompt (it is not a large data set).

1. To get started we will first ignore the experimental manipulation of treated cells with puromycin. The R code below will estimate the *linear* model $E(Y_i) = \beta_0 + \beta_1 x_i$ using `nls` and plot this model with the data.

```
library(ggplot2)
m <- nls(rate ~ b0 + b1 * conc, start = c(b0 = 0, b1 = 0), data = Puromycin)

d <- expand.grid(conc = seq(0, 1.2, length = 100), state = c("treated","untreated"))
d$yhat <- predict(m, newdata = d)

p <- ggplot(Puromycin, aes(x = conc, y = rate)) +
  geom_point(aes(color = state)) + geom_line(aes(y = yhat), data = d) +
  labs(x = "Substrate Concentration (ppm)",
    y = "Reaction Rate (counts/min)", color = "Cell State") +
  theme_minimal() + theme(legend.position = c(0.8,0.2))
plot(p)
```



Clearly the linear model is not a good model for the data. Cut-and-paste the R code above and modify it to replace the linear model with the Michaelis-Menten model (ignoring the experimental manipulation for now). To specify starting values for $\theta_1$ and $\theta_2$, look at the plot of the data and try to guess their approximate values. Remember that $\theta_1$ is the asymptote and $\theta_2$ is the concentration at which the (expected) reaction rate is half way between zero and the asymptote.

2. Now consider a linear model that will assume a linear relationship between reaction rate and concentration that is different for treated versus untreated cells (i.e., an "interaction" between substrate concentration and cell state).

```
m <- lm(rate ~ state + conc + conc:state, data = Puromycin)
cbind(summary(m)$coefficients, confint(m))
```

```
                    Estimate Std. Error t value  Pr(>|t|)  2.5 % 97.5 %
(Intercept)           103.49      10.53   9.832 6.914e-09  81.46 125.52
stateuntreated        -17.45      15.06  -1.158 2.611e-01 -48.98  14.08
```

```
conc                    110.42       20.46   5.397 3.301e-05   67.60  153.24
stateuntreated:conc     -21.08       32.69  -0.645 5.266e-01  -89.50   47.33
```

From `summary` we can see that the model can be written as

$$E(Y_i) = \beta_0 + \beta_1 d_i + \beta_2 x_i + \beta_3 d_i x_i,$$

where $x_i$ represents concentration (`conc`) and $d_i$ is an indicator variable for when the treatment (`state`) is "untreated" so that

$$d_i = \begin{cases} 1, & \text{if the } i\text{-th observation is of untreated cells,} \\ 0, & \text{otherwise.} \end{cases}$$

This model can be written case-wise as

$$E(Y_i) = \begin{cases} \beta_0 + \beta_2 x_i, & \text{if the } i\text{-th observation is of treated cells,} \\ \beta_0 + \beta_1 + (\beta_2 + \beta_3) x_i, & \text{if the } i\text{-th observation is of untreated cells.} \end{cases}$$

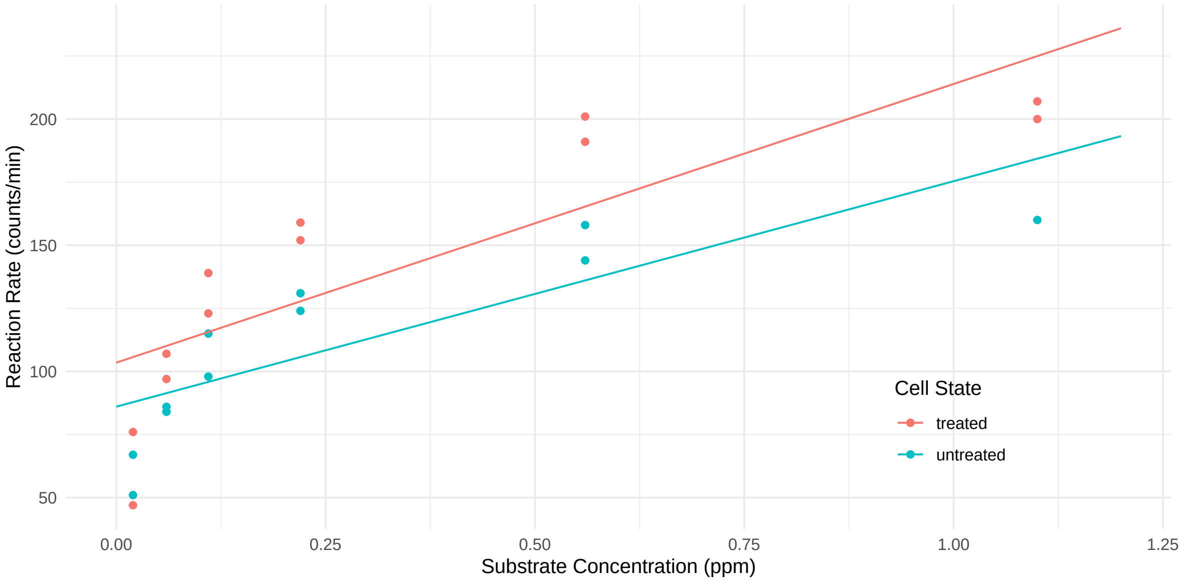
We can replicate this model using the `nls` function by specifying an indicator variable in the model formula.

```
m <- nls(rate ~ b0 + b1*(state == "untreated") +
  b2*conc + b3*(state == "untreated")*conc,
  data = Puromycin, start = list(b0 = 0, b1 = 0, b2 = 0, b3 = 0))
summary(m)$coefficients
```

```
   Estimate Std. Error t value  Pr(>|t|)
b0   103.49      10.53   9.832 6.914e-09
b1   -17.45      15.06  -1.158 2.611e-01
b2   110.42      20.46   5.397 3.301e-05
b3   -21.08      32.69  -0.645 5.266e-01
```

```
d <- expand.grid(conc = seq(0, 1.2, length = 100), state = c("treated","untreated"))
d$yhat <- predict(m, newdata = d)

p <- ggplot(Puromycin, aes(x = conc, y = rate, color = state)) +
  geom_point() + geom_line(aes(y = yhat), data = d) +
  labs(x = "Substrate Concentration (ppm)",
    y = "Reaction Rate (counts/min)", color = "Cell State") +
  theme_minimal() + theme(legend.position = c(0.8,0.2))
plot(p)
```



Since there are only two levels of `state` we could also use the `ifelse` function to produce the same results as using the indicator variable.

```
m <- nls(rate ~ ifelse(state == "treated", b0 + b2*conc, b0 + b1 + (b2 + b3)*conc),
  data = Puromycin, start = list(b0 = 0, b1 = 0, b2 = 0, b3 = 0))
summary(m)$coefficients
```

```
   Estimate Std. Error t value  Pr(>|t|)
b0   103.49      10.53   9.832 6.914e-09
b1   -17.45      15.06  -1.158 2.611e-01
b2   110.42      20.46   5.397 3.301e-05
b3   -21.08      32.69  -0.645 5.266e-01
```

Obviously this is a poor model for the `Puromycin` data since expected reaction rate does not appear to be a linear function of substrate concentration. Instead we would like to have a model where there the Michaelis-Menten model describes the relationship between the expected reaction rate and substrate concentration, but *differently for each state* so that the $\theta_1$ and $\theta_2$ parameters can depend on the state. Estimate this model using the `nls` function, noting that there are many different ways that this model could be parameterized. Plot the model as well with the raw data.

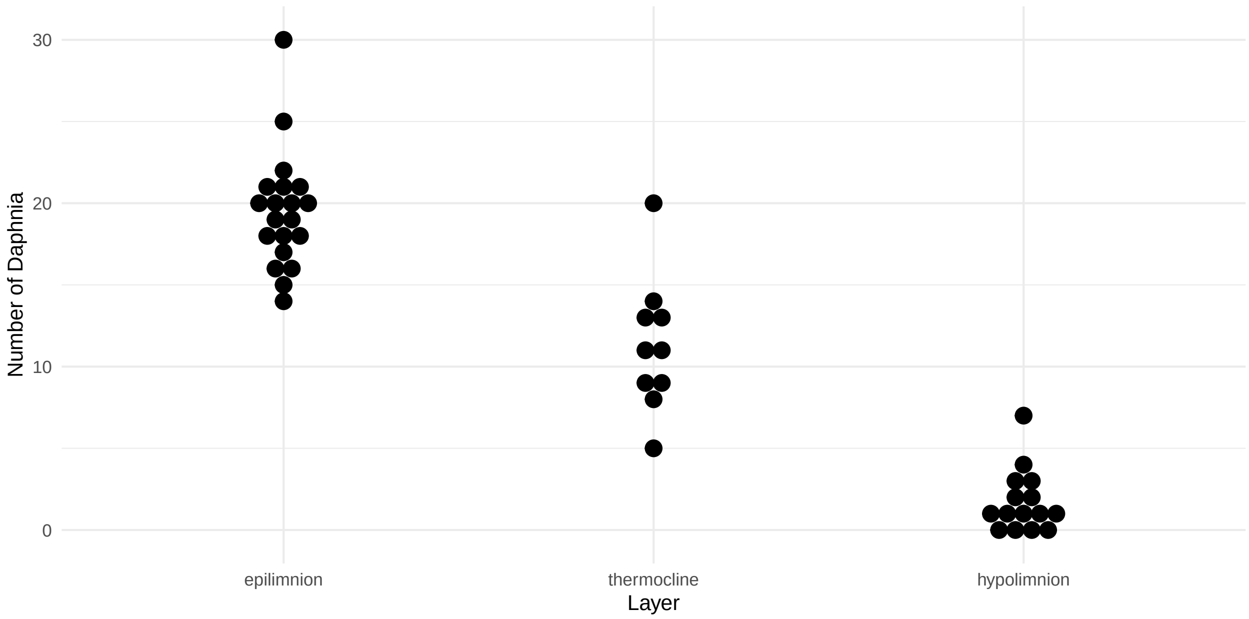
3. There are several potentially useful inferences we might make here. One is the values of the two parameters for the Michaelis-Menten model *for each state*. Another is the difference in the parameters *between states*. Depending on how the model was parameterized in the previous question, some of these could be found simply by using `summary` and `confint`, while others might require using `lincon` unless the model is reparameterized. Produce estimates, standard errors, and confidence intervals for the six quantities described above.

## Heteroscedasticity in the Daphnia Data

An example from you first homework assignment featured data from a survey of water fleas where the number of water fleas were counted in one liter samples of water taken from three different layers of a lake.

```
library(trtools) # for daphniastrat
library(ggplot2)
p <- ggplot(daphniastrat, aes(x = layer, y = count)) +
  geom_dotplot(binaxis = "y", binwidth = 1, stackdir = "center") +
  labs(x = "Layer", y = "Number of Daphnia") + theme_minimal()
```

```
plot(p)
```



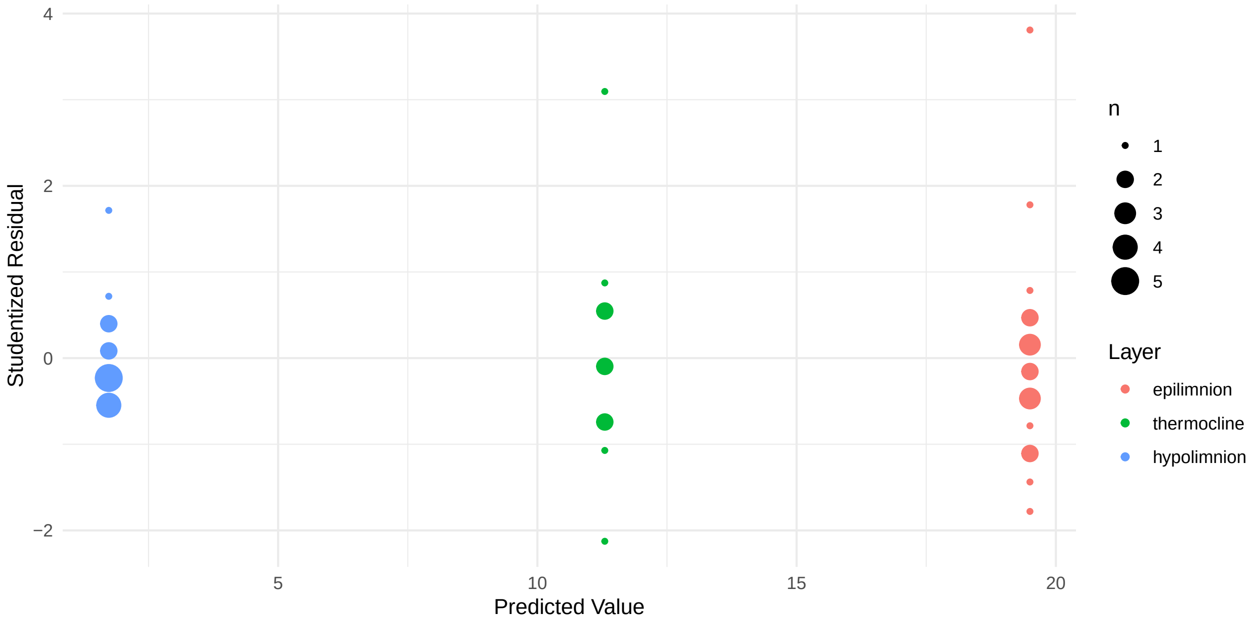
We used the following linear model for these data.

```
m <- lm(count ~ layer, data = daphniastrat)
summary(m)$coefficients
```

```
                  Estimate Std. Error t value  Pr(>|t|)
(Intercept)          19.50     0.7271  26.820 4.727e-28
layerthermocline     -8.20     1.2593  -6.512 7.293e-08
layerhypolimnion    -17.77     1.1106 -15.997 1.784e-19
```

But heteroscedasticity is evident in the plot of the raw data above and also in a plot of the studentized residuals.

```
daphniastrat$yhat <- predict(m)
daphniastrat$rest <- rstudent(m)
p <- ggplot(daphniastrat, aes(x = yhat, y = rest, color = layer)) +
  geom_count() + theme_minimal() +
  labs(x = "Predicted Value", y = "Studentized Residual", color = "Layer")
plot(p)
```



Note that `geom_count` is a variant of `geom_point` that makes the point size proportional to the number of points at a particular position, which is useful here.

Heteroscedasticty is quite common when the response variable is a count. Typically the variance of the counts increases with the expected count. Here we will consider a couple of ways of dealing with this heteroscedasticty.

1. We might assume that the variance varies by layer with the expected count, so that

$$\operatorname{Var}(Y_i) = \begin{cases} \sigma_e^2, & \text{if the } i\text{-th observation is from the epilimnion layer,} \\ \sigma_t^2, & \text{if the } i\text{-th observation is from the thermocline layer,} \\ \sigma_h^2, & \text{if the } i\text{-th observation is from the hypolimnion layer.} \end{cases}$$

   If so, then our weights should be specified such that

$$w_i \propto \begin{cases} 1/\sigma_e^2, & \text{if the } i\text{-th observation is from the epilimnion layer,} \\ 1/\sigma_t^2, & \text{if the } i\text{-th observation is from the thermocline layer,} \\ 1/\sigma_h^2, & \text{if the } i\text{-th observation is from the hypolimnion layer.} \end{cases}$$

   We do not know $\sigma_e^2$, $\sigma_t^2$, and $\sigma_h^2$, but they could be *estimated* using the sample variances $s_e^2$, $s_t^2$, and $s_h^2$, respectively, which we can easily compute since we have multiple observations from each layer. Estimate the model using weighted least squares with weights computed as described above. The sample standard deviations can be computed and used to add weights to the data frame using functions from the **dplyr** package as demonstrated with the `CancerSurvival` data in lecture on February 18th.

2. Assuming again that the variance of the counts vary by layer so that

$$\operatorname{Var}(Y_i) = \begin{cases} \sigma_e^2, & \text{if the } i\text{-th observation is from the epilimnion layer,} \\ \sigma_t^2, & \text{if the } i\text{-th observation is from the thermocline layer,} \\ \sigma_h^2, & \text{if the } i\text{-th observation is from the hypolimnion layer,} \end{cases}$$

   another approach would be to use a *parametric model* that estimates the usual parameters of the regression model (i.e., $\beta_0$, $\beta_1$, and $\beta_3$) and the three variances above *simultaneously*. Do this using the `gls` function from the **nlme** package as demonstrated in lecture on February 23rd.

3. With counts it is often assumed that the variance is proportional to the expected response — i.e., $\operatorname{Var}(Y_i) \propto E(Y_i)$. Assuming that is true here, estimate the model using *iteratively weighted least squares* as was demonstrated in lecture on February 23rd.