Logistic Regression

Jesse Keränen

11/14/2022

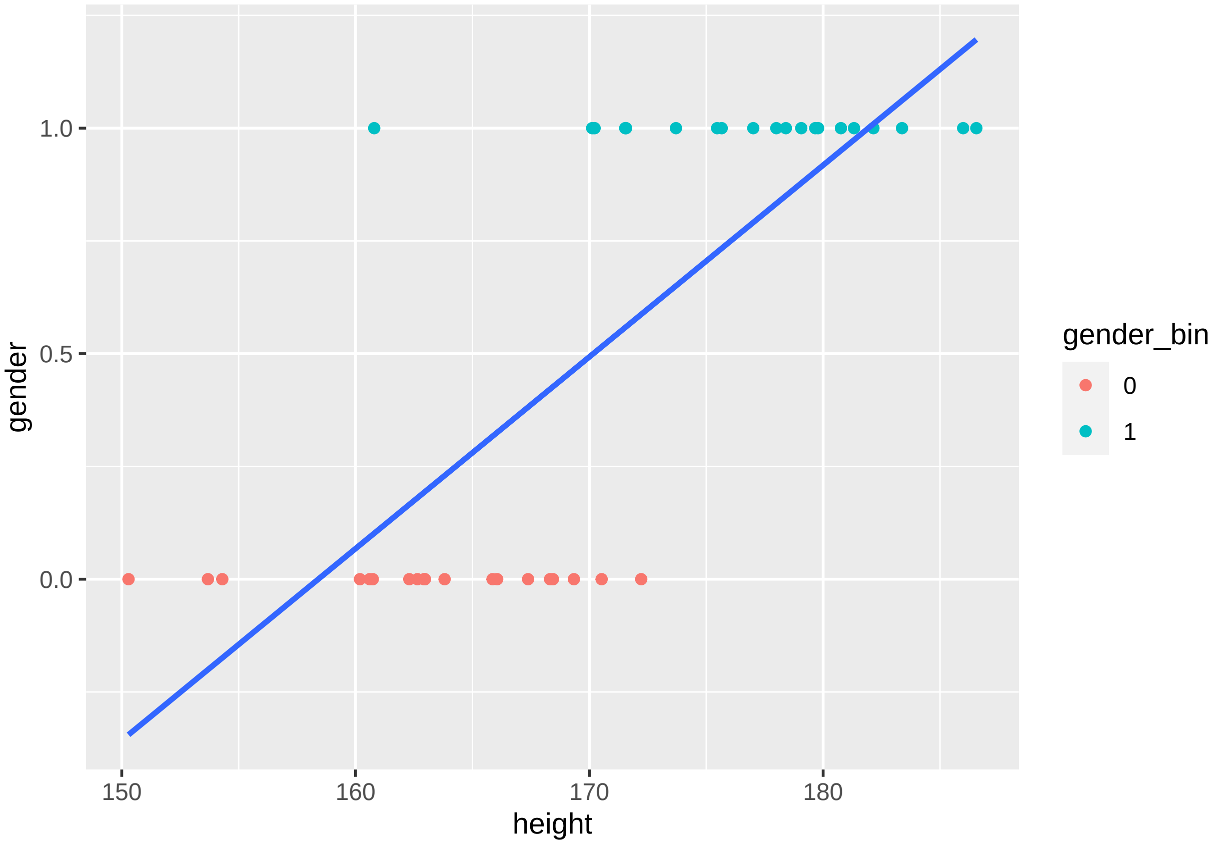
Prologue

Purpose of this file is to help me understand basic idea of logistic regression. Sometimes, instead of just looking at the mathematical formulas, I like to try to understand new concepts using example data and applying new method to that. In this file I try to go through process of logistic regression using sample data.

Data

To make this more interesting I need somewhat reasonable data, but discovering something from the data is not goal of this document. That's why I don't spend time looking for a suitable data set, but I create my own. I expect height of the females and males to be normally distributed. Then I quickly looked from internet means and standard deviations for both of these variables. Then I just simply generate sample set of heights for 20 individuals for both genders using R's rnorm function. Finally I store new created height values with gender denoted by binary variable to a data table.

```r
set.seed(1)
n <- 20
male_height <- rnorm(n, 175.77, 6.76)
female_height <- rnorm(n, 163.32, 6.55)

dt <- data.table(height = c(male_height, female_height),
                 gender_bin = as.factor(c(rep(1, n), rep(0, n))),
                 gender = c(rep(1, n), rep(0, n)))

ggplot(dt, aes(height, gender)) +
  geom_point(aes(color = gender_bin)) + geom_smooth(method = "lm", se = F)
```



We can see from the plot that gender clearly have different means, but there are still overlapping observations. That is exactly what we wanted.

Probability , odds and logit

As far as I have understood we can think of Y axis values in above plot as probability of observing different genders. If height of a individual is high there is high possibility that he is male. If height of the observation is low there is high possibility that she is female. That would then mean that there is low possibility of that individual being male. We would like to fit a line to the our data as we do in OLS which would show us our estimate of for probability of a persons gender for all heights. Since our response variable is binary straight line wouldn't fit too well to our data. Instead we can see that line with kind of s-shape would be more suitable.

To accomplish that we need to introduce couple of new measures. First one would be odds. Odds are rather simply to calculate.

$$Odds = \frac{p(x)}{1-p(x)}$$

Odds are basically probability of an event divided by one - probability of an event. Odds can get values between zero and positive infinity, where as probabilities are bound by zero and one. For the next equation I didn't find a proof, but I guess it is so well know fact that I take it as given. Odds equal exponent of a linear function.

$$\frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 X}$$

Which indicates that logarithm of odds, called log-odds or logit, can be presented by linear function. Log-odds can vary from minus infinity to positive infinity.

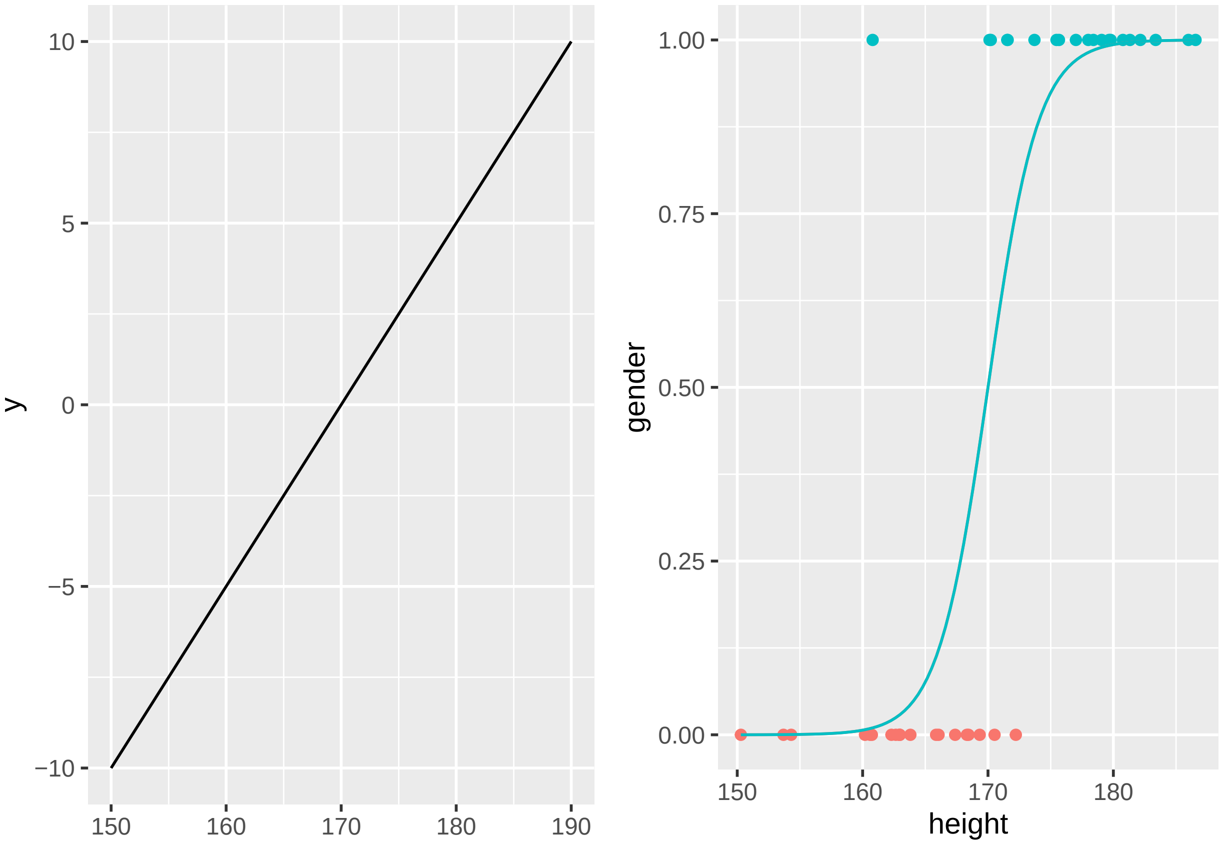
$$ln(\frac{p(x)}{1-p(x)}) = \beta_0 + \beta_1 X$$

Now that we know formula for odds, we can easily derive function for probabilities.

$$p(x) = (1 - p(x)e^{\beta_0+\beta_1 X}$$
$$p(x) = e^{\beta_0+\beta_1 X} - p(x)e^{\beta_0+\beta_1 X}$$
$$p(x) + p(x)e^{\beta_0+\beta_1 X} = e^{\beta_0+\beta_1 X}$$
$$(1 + e^{\beta_0+\beta_1 X})p(x) = e^{\beta_0+\beta_1 X}$$
$$p(x) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

At this point I am not yet interested in how we can find the line that best presents the probabilities. In this case (because observations are ones zeros, which lead to infinite logo-dds) we have to use likelihood function and maximize that. Next I try to clarify likelihood function calculation with my initial guess for the line. My first guess is that β_0 equals -85 and β_1 equals 0.5.

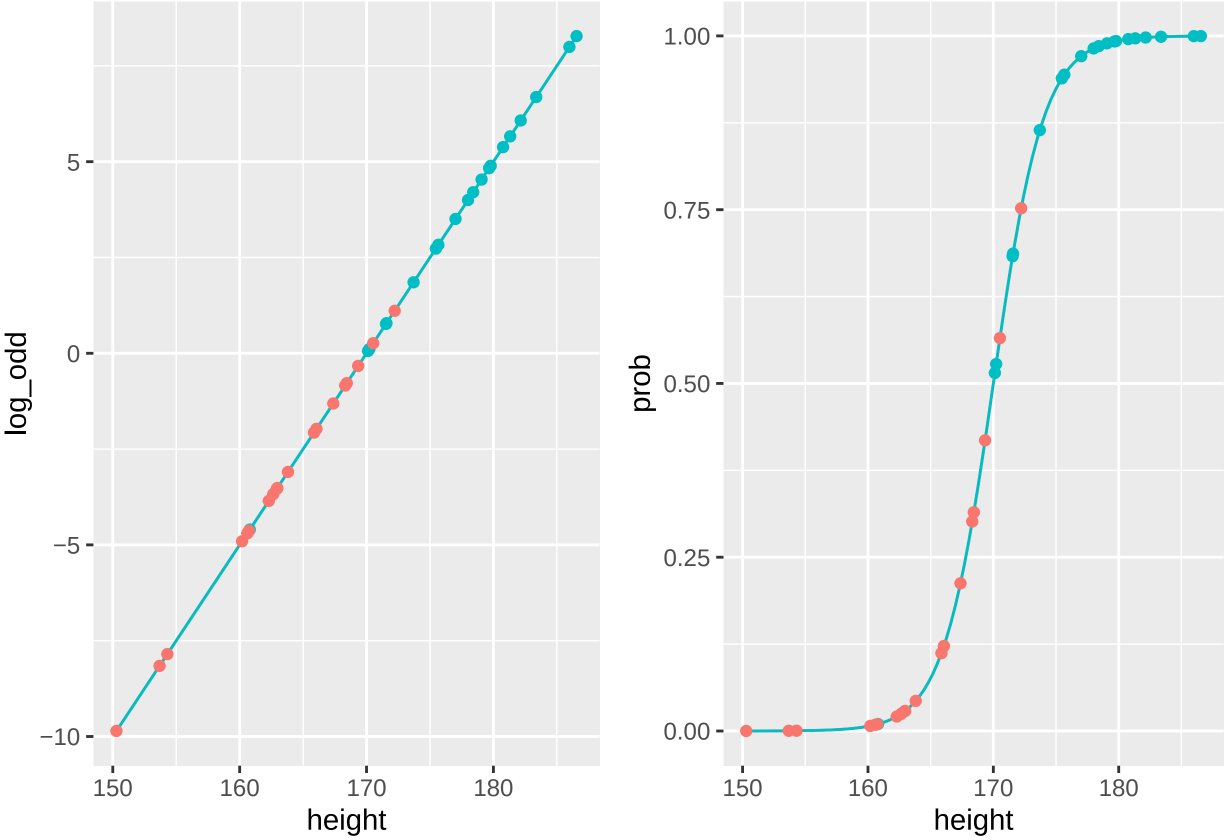
```r
a <- ggplot(dt) + geom_function(fun = function(x) -85 + 0.5*x) + xlim(min=150, max=190)

b <- ggplot(dt, aes(height, gender, color = gender_bin)) +
  geom_point() +
  geom_function(fun = function(x) exp(-85 + 0.5*x)/(1 + exp(-85 + 0.5*x))) +
  theme(legend.position = "none")

cowplot::plot_grid(a, b)
```



First I have plotted different logit as function of heights. We can see the linear relationship. Next probabilities are plotted as function of height using formula derived above with out parameter values for betas. We can see that line fit data quite nicely.

When we are looking for our optimal line, values that we are changing are betas. Since there is no closed form solution like for the OLS, we have to use numeric methods to find optimal one. This means that we need a objective function that tells us how good our current solution is compared to previous ones. We are going to use likelihood function. First we want to project our data to our current line. That way we can calculate logit and probit values.

```r
dt[, prob := exp(-85 + 0.5*height)/(1+exp(-85 + 0.5*height))]

dt[, log_odd := log(prob/(1-prob))]

c <- ggplot(dt, aes(height, log_odd, color = gender_bin)) +
  geom_function(fun = function(x) -85 + 0.5*x) + geom_point() +
  theme(legend.position = "none")

d <- ggplot(dt, aes(height, prob, color = gender_bin)) +
  geom_function(fun = function(x) exp(-85 + 0.5*x)/(1 + exp(-85 + 0.5*x))) +
  geom_point() + theme(legend.position = "none")

cowplot::plot_grid(c, d)
```



Using probit values we can easily calculate value for likelihood function.

$$likelihood = \prod_{i=1} p^{y_i}(1-p)^{1-y_i}$$

Which is just product of all observations probit (or one minus probit for observations with dependent value of zero) values. We can see that since we want to maximize our objective function optimizer has incentive to set line so that observations with observed dependent value of one have high probit value and observations with observed dependent value of zero have low probit value. To obtain some characteristics that will make our optimization easier we are going to use logarithmic version of the likelihood function.

$$ln(likelihood) = \sum_{i=1} y_i ln(p_i) + (1 - y_i) ln(1 - p_i)$$

```
dt[, l := prob^gender*(1 - prob)^(1 - gender)]
dt[, ll := gender*log(prob) + (1 - gender)*log(1 - prob)]

likelihood <- prod(dt$l)
log_likelihood <- sum(dt$ll)
log_likelihood
```

```
## [1] -11.23466
```

We can see that with our initial guess log likelihood function value would be -11.23. Let's see whether we can improve that.

Optimization

Now that we know what we are actually calculating we can move to optimization. Our optimization algorithm goes as follows. First we calculate gradient of our objective function. In our case log likelihood function. Then we will move in direction of the gradient by step length α, which we can also optimize. We repeat this process until the change in our objective function is small enough. We can decide this preciseness. After we have found this point we know that it is optimal since log likelihood function is convex function. This is called gradient ascent method. So, let's try to derive gradient of our log likelihood function.

$$\frac{\partial LL(\beta)}{\partial \beta_j} = \frac{\partial LL(\beta)}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \beta_j}$$

$$LL(\beta) = y\ ln(p) + (1 - y) ln(1 - p)$$

$$\frac{\partial LL(\beta)}{\partial p} = y \cdot \frac{1}{p} - 1 \cdot (1 - y) \cdot \frac{1}{1-p} = \frac{y}{p} - \frac{1-y}{1-p}$$

$$p = \sigma(z) \text{ where } \sigma \text{ denotes exponential and } z \text{ denotes } X\beta$$

$$\sigma(z) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$\frac{\partial p}{\partial z} = \frac{\partial \sigma(z)}{\partial z} = \frac{\partial}{\partial z} \cdot \sigma(z) = \frac{\partial}{\partial z} \cdot \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$\frac{\partial p}{\partial z} = \frac{(1+e^{\beta_0+\beta_1 x}) \cdot e^{\beta_0+\beta_1 x} + e^{\beta_0+\beta_1 x} \cdot e^{\beta_0+\beta_1 x}}{(1+e^{\beta_0+\beta_1 x})^2} = \frac{(1+e^{\beta_0+\beta_1 x}) \cdot e^{\beta_0+\beta_1 x} + e^{(\beta_0+\beta_1 x)+(\beta_0+\beta_1 x)}}{(1+e^{\beta_0+\beta_1 x})^2} =$$

$$\frac{(1+e^{\beta_0+\beta_1 x}) \cdot e^{\beta_0+\beta_1 x} + e^{2(\beta_0+\beta_1 x)}}{(1+e^{\beta_0+\beta_1 x})^2} = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} - \frac{e^{2(\beta_0+\beta_1 x)}}{(1+e^{\beta_0+\beta_1 x})^2}$$

$$\text{since}$$

$$e^{2(\beta_0+\beta_1 x)} = e^{(\beta_0+\beta_1 x)^2}$$

$$\text{we can write}$$

$$\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} - \left(\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}\right)^2$$

$$\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} - \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \cdot \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$\left(1 - \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}\right) \cdot \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$\text{finally}$$

$$\frac{\partial z}{\partial \beta_j} = x_j$$

$$\frac{\partial LL(\beta)}{\partial \beta_j} = (\frac{y}{p} - \frac{1-y}{1-p}) \cdot (1 - \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}) \cdot \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}) \cdot x_j$$

$$\frac{\partial LL(\beta)}{\partial \beta_j} = (\frac{y}{p} - \frac{1-y}{1-p}) \cdot (1-p) \cdot p \cdot x_j$$

$$(\frac{y \cdot (1-p)p}{p} - \frac{(1-y) \cdot (1-p)p}{1-p}) \cdot x_j$$

$$(y(1-p) - (1-y)p) \cdot x_j$$

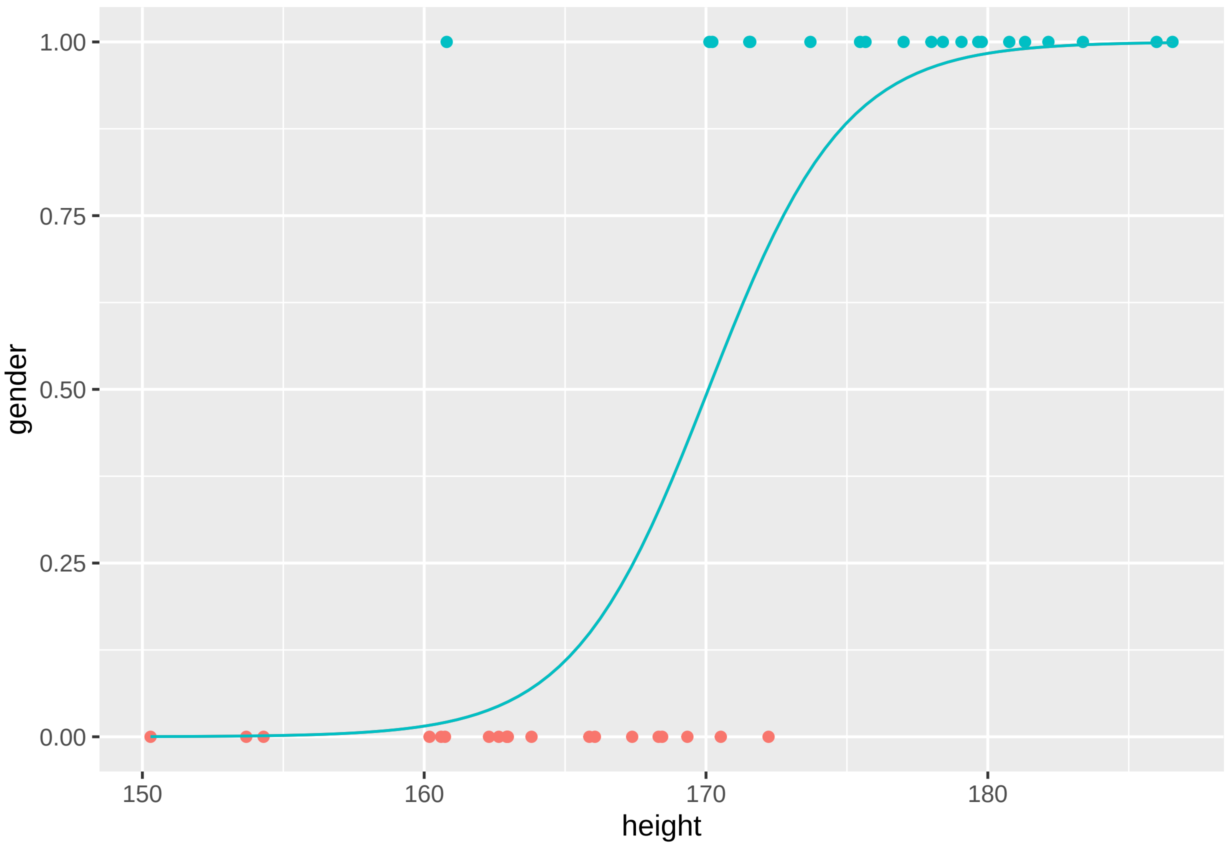
$$(y - yp - p + yp) \cdot x_j = (y-p) \cdot x_j = (y - \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}})x_j$$

```r
alpha <- 0.01

gradient <- function(dt, beta_0, beta_1) {
  probability(dt, beta_0, beta_1)
  beta_0 = beta_0 + alpha * dt$gender * (dt$gender - dt$prob)
  beta_1 = beta_1 + alpha * dt$gender * (dt$gender - dt$prob)
}

probability <- function(dt, beta_0, beta_1) {
  dt[, prob := exp(beta_0 + beta_1*height)/(1 + exp(beta_0 + beta_1*height))]
}
```

```r
ab <- glm(gender ~ height, data=dt, family=binomial (link=logit))
summary(ab)
```

```
##
## Call:
## glm(formula = gender ~ height, family = binomial(link = logit),
##     data = dt)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.56718  -0.32065   0.01172   0.25730   2.77639
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -70.2073    22.8825  -3.068  0.00215 **
## height        0.4128     0.1347   3.065  0.00217 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 55.452  on 39  degrees of freedom
## Residual deviance: 22.117  on 38  degrees of freedom
## AIC: 26.117
##
## Number of Fisher Scoring iterations: 6
```

```
ggplot(dt, aes(height, gender, color = gender_bin)) +
  geom_point() + geom_function(fun = function(x) exp(ab$coefficients[1] +
  ab$coefficients[2]*x)/(1 + exp(ab$coefficients[1] + ab$coefficients[2]*x))) +
  theme(legend.position = "none")
```



```
dt[, prob_opt := exp(ab$coefficients[1] +
                       ab$coefficients[2]*height)/(1+exp(ab$coefficients[1] +
                       ab$coefficients[2]*height))]

dt[, ll_opt := gender*log(prob_opt) + (1 - gender)*log(1 - prob_opt)]

log_likelihood_opt <- sum(dt$ll_opt)
```