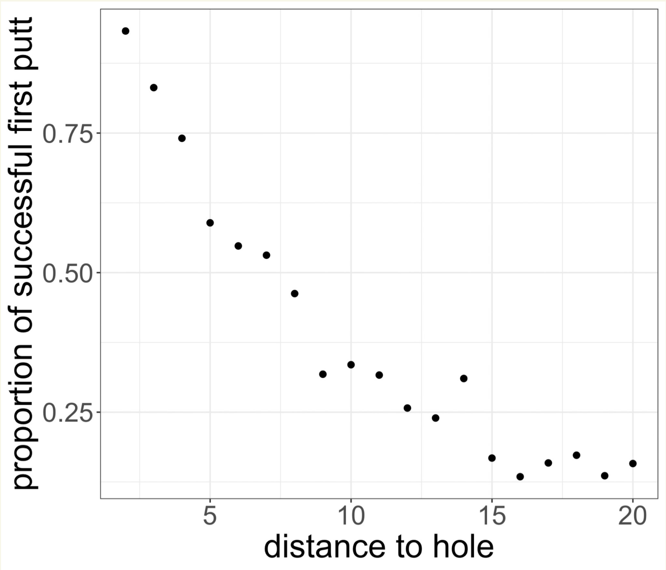
Logistic Regression

Q Predict the <u>probability</u> that a putt is sunk as a function of distance to hole.

Dataset of 5,988 putts from Columbia including distance to hole and whether the putt was sunk or not.

<u>Visualize</u>



What do you notice?

Variables i = index of i^{th} putt in our dataset

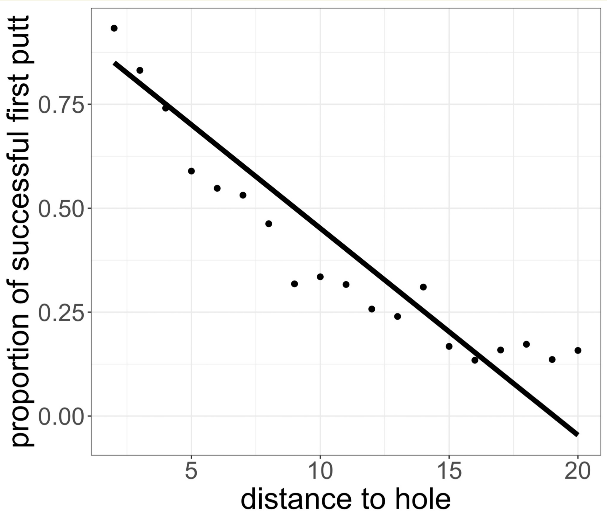
$Y_i = 1$ if putt is sunk, else 0

X_i = distance to hole of i^{th} putt

Model 1 (Linear Regression)

$$\begin{cases} Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \\ \text{mean zero Noise } \mathbb{E}\varepsilon_i = 0 \end{cases}$$

We know how to estimate β_0, β_1.

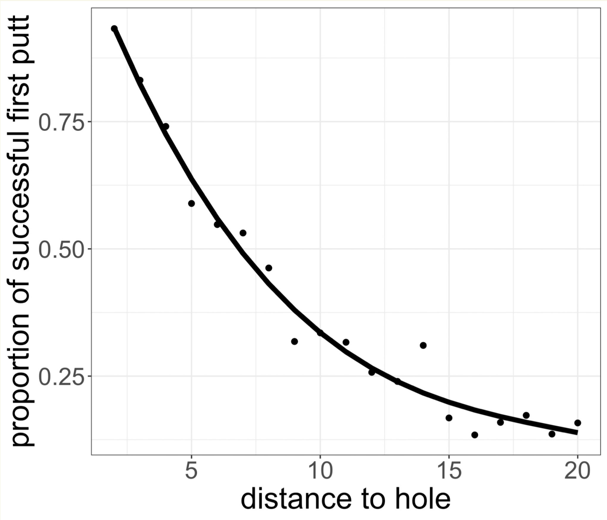


Not a great fit.

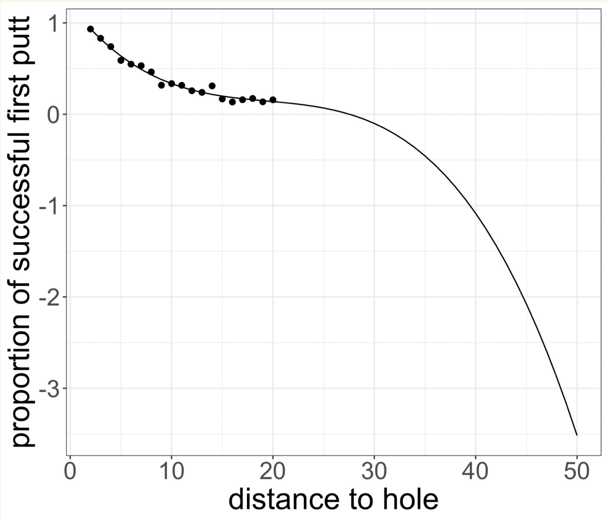
Model (Cubic Regression)

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X^2 + \beta_3 X_i^3 + \varepsilon_i$$

We know how to estimate $(\beta_0, \beta_1, \beta_2, \beta_3)$!



Fit looks good when $X_i \in [0, 20]$

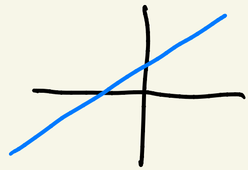


Not able to extrapolate when $X_i > 20$ ...

Problem The probability of an event must lie in $[0,1]$, ordinary linear regression does not guarantee this

Idea Force our predictions $\hat{Y}_i$ to lie in $[0,1]$

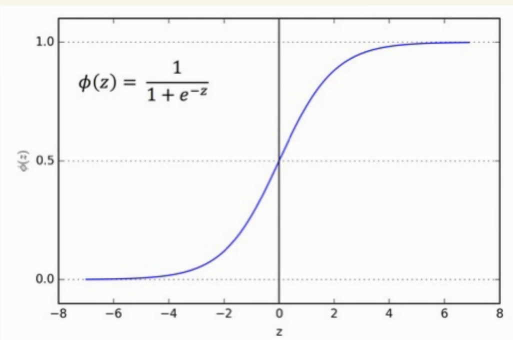
OLR: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$



Squishification Function

↳ Takes a number in $(-\infty, \infty)$ and squishes it into $[0,1]$

$$\boxed{\text{Logistic}(z) = \frac{1}{1+e^{-z}}} = \text{Sigmoid}(z) = \sigma(z)$$



Before: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$

Now: $\hat{y}_i = \text{Logistic}(\hat{\beta}_0 + \hat{\beta}_1 x_i)$

Model the probability directly,

$$p_i = P(y_i = 1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_i)}}$$

Logistic Regression **Model**

$$\boxed{\begin{aligned} p_i &= P(y_i = 1) = \frac{1}{1+e^{-(x_i^T\beta)}} \\ y_i &\sim \text{Bernoulli}(p_i) = \begin{cases} 1 & \text{w.p. } p_i \\ 0 & \text{w.p. } 1-p_i \end{cases} \end{aligned}}$$

Q Our data is in terms of $Y_i \in \{0,1\}$ and X_i, not $\{p_i\}$.
So, how do we estimate $\vec{\beta}$ in logistic Regression?

* In linear regression, we estimate β by minimiting the Residual Sum of Squares RSS (e.g. the squared error),

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - x_i^T\beta)^2$$

* In logistic Regression, we estimate β by minimizing the log loss, i.e. the cross entropy loss,

$$L(\beta) = \frac{-1}{n} \sum_{i=1}^{n} y_i \log P_i + (1-y_i) \log (1-P_i)$$

where $P_i = \mathbb{P}(y_i = 1 / x_i, \beta) = \dfrac{1}{1+e^{-x_i^T \beta}}$

- If $y_i = 1$ then $y_i \log P_i + (1-y_i) \log(1-P_i) = \log P_i$
  - If $P_i \approx 1$ then $\log P_i$ high, so $-\log P_i$ low, so $L(\beta)$ low
  - If $P_i \approx 0$ then $\log P_i$ low, so $-\log P_i$ low, so $L(\beta)$ high

- Similarly, if $y_i = 0$ then $y_i \log P_i + (1-y_i) \log(1-P_i) = \log(1-P_i)$ and a low loss correspond to a low P_i

- Set gradient of the loss function equal to 0 and solve (see Math HW):

$$\nabla_\beta L(\beta) = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \nabla_\beta \log P_i + (1-y_i) \nabla_\beta \log (1-P_i) \right]$$

Set $\nabla_\beta L(\beta) = 0$ and solve for β (iteratively, e.g. by gradient descent)

Golf:

Variables

i = index of i^{th} putt in our dataset

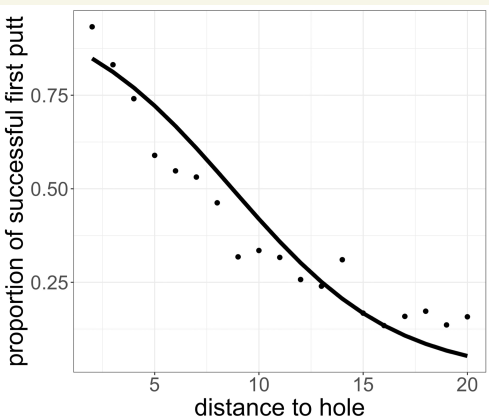
Y_i = 1 if putt is sunk, else 0

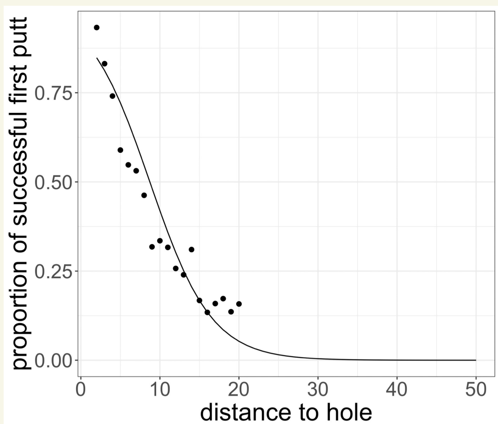
X_i = distance to hole of i^{th} putt

Logistic Regression Model

$$\mathbb{P}(Y_i=1) = \frac{1}{1+e^{-\beta_0+\beta_1 X_i}}$$

```
m3 = glm(y ~ dist_to_hole, data=putt_df_1, family="binomial")
m3
```



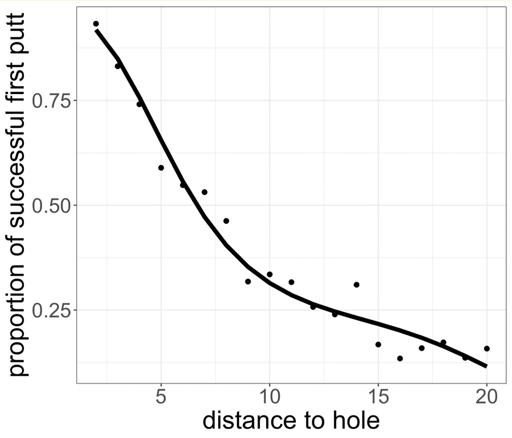


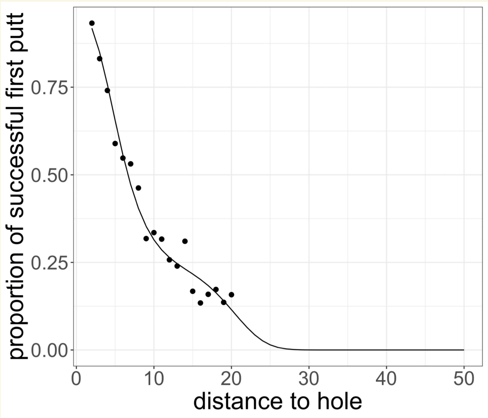
It extrapolates well because we forced our predictions to lie in [0,1].

* We can do better by modeling the log odds as a cubic,

$$\mathbb{P}(Y_i=1) = \frac{1}{1+e^{-\beta_0+\beta_1 X_i+\beta_2 X_i^2+\beta_3 X_i^3}}$$

```
m4 = glm(y ~ poly(dist_to_hole,3), data=putt_df_1, family="binomial")
```





Takeaway

Use linear regression to predict a Real number.

Use logistic regression to predict a probability in [0,1].