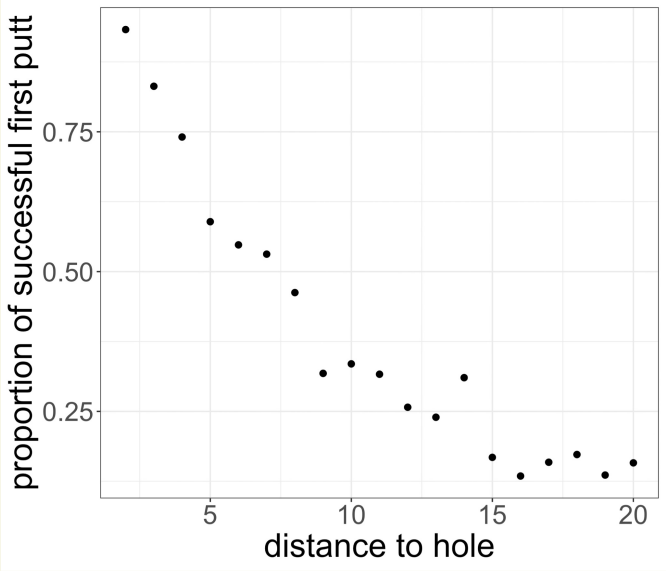
Regression, Part 3: Logistic Regression

Q Predict the probability 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.

Visualize



What do you notice?

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

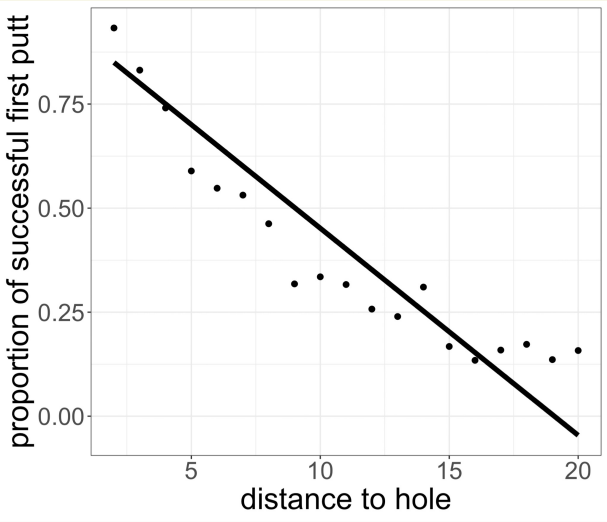
$Y_i = 1$ if putt is sunk, else 0

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.

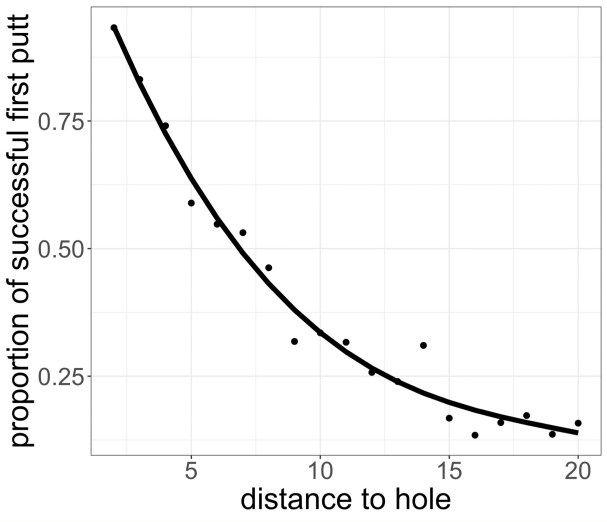


Not a great fit.

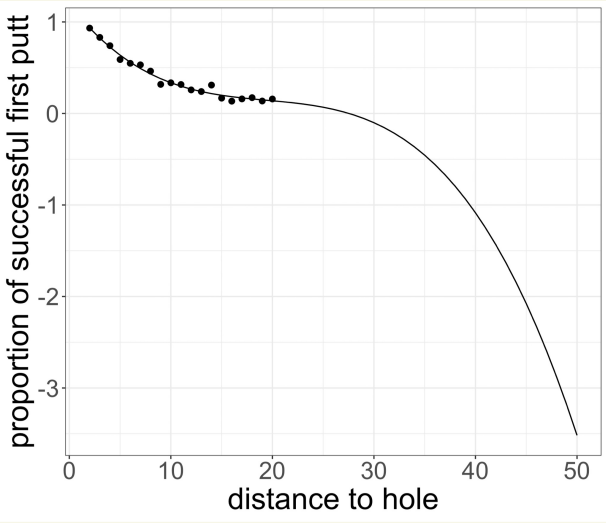
Model (Cubic Regression)

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

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 be in $[0,1]$ but ordinary linear Regression does not guarantee this.

Linear Regression $\vec{y} = X\vec{\beta} + \vec{\varepsilon}$

$\vec{y}$ is a vector of Real numbers

This model does not force $\vec{x}_i^T \vec{\beta}$ to be Bounded!

Idea Force our probability predictions to be bounded in $[0,1]$.

Recall $Y_i = \begin{cases} 1 & \text{if } i^{th} \text{ putt is made} \\ 0 & \text{if } i^{th} \text{ putt is missed} \end{cases}$

So, Y_i is a binary variable.

We are interested in $\mathbb{P}(Y_i = 1)$, so lets just model this probability directly!

Logistic Regression

Model

$$\log\left(\frac{\mathbb{P}(Y_i=1)}{1-\mathbb{P}(Y_i=1)}\right) = \beta_0 + \beta_1 X_i$$

Equivalently,

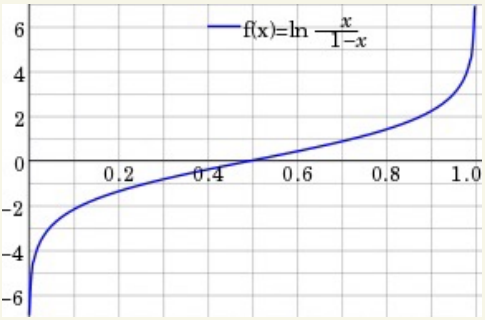
$$\mathbb{P}(Y_i=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_i)}}$$

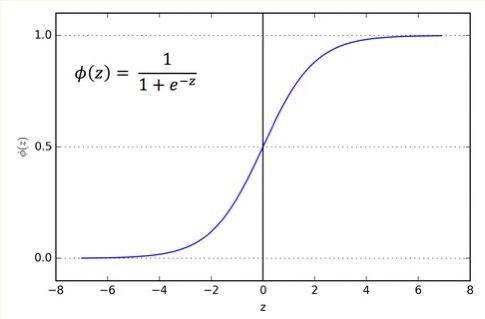
logit function

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

logistic function

$$\text{logistic}(z) = \frac{1}{1+e^{-z}}$$





logit $(0,1) \to \mathbb{R}$

logistic $\mathbb{R} \to (0,1)$

logit, logistic are inverse functions

* we want to model $p = \mathbb{P}(Y=1)$
to do so, we simply map $p \in [0,1]$
into $\text{logit}\left(\frac{p}{1-p}\right) \in \mathbb{R}$, and then
perform linear regression on the logit scale.

Note $\frac{p}{1-p}$ is the odds ratio, or the odds

$\log\left(\frac{p}{1-p}\right)$ is the log odds

So, we model the log odds as a linear function.

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
minimizing 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 $p_i = \mathbb{P}(y_i = 1 | x_i, \beta) = \frac{1}{1 + e^{-x_i^T \beta}}$

- 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 corresponds to a low p_i

* Let's minimize logloss:

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

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

Now,

Let $\phi(z) = \frac{1}{1+e^{-z}} = \text{logistic}(z)$

Then $\frac{d}{dz}\phi(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \phi(z)(1-\phi(z))$

$$\nabla_\beta p_i = \nabla_\beta \left(\frac{1}{1+e^{-x_i^T\beta}}\right) = \nabla_\beta \phi(x_i^T\beta)$$

$$= \phi(x_i^T\beta)(1-\phi(x_i^T\beta))\ \vec{x}_i$$

by Chain Rule

$$= p_i(1-p_i)\vec{x}_i$$

Hence

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

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

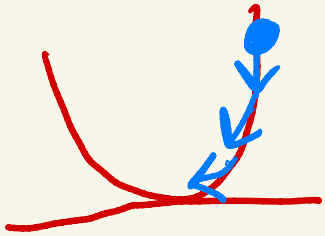
$$= -\frac{1}{n} \sum_{i=1}^{n} \left[y_i (1-p_i) x_i - (1-y_i) p_i x_i \right]$$

$$= -\frac{1}{n} \sum_{i=1}^{n} (y_i - p_i) x_i$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \left(y_i - \phi(x_i^T \beta) x_i \right)$$

■

* Setting $\nabla_\beta L(\beta) = 0$ has no known closed form solution.
So, use Newton Raphson or Gradient descent to approximate $\hat{\beta} = \underset{\beta}{\operatorname{argmin}} L(\beta)$.



Gradient Descent

Iterate until convergence of $\vec{\beta}$,
i.e. until $\|\vec{\beta}^{(t)} - \vec{\beta}^{(t+1)}\| < \varepsilon$:

$$\vec{\beta}^{(t+1)} = \vec{\beta}^{(t)} + K \cdot \sum_{i=1}^{n} \left(y_i - \phi(x_i^T \beta)\right) \vec{x}_i$$

K = learning Rate

Anyone Recognize K ??

* $ELO^{(t+1)} = ELO^{(t)} + K\left(\mathbb{1}(win) - \hat{P}(win)\right)$

→ ELO is one iteration of gradient descent!
will discuss later today.

So, now we can estimate β in logistic regression!! Done automatically in R.

✳ Recall:

Variables

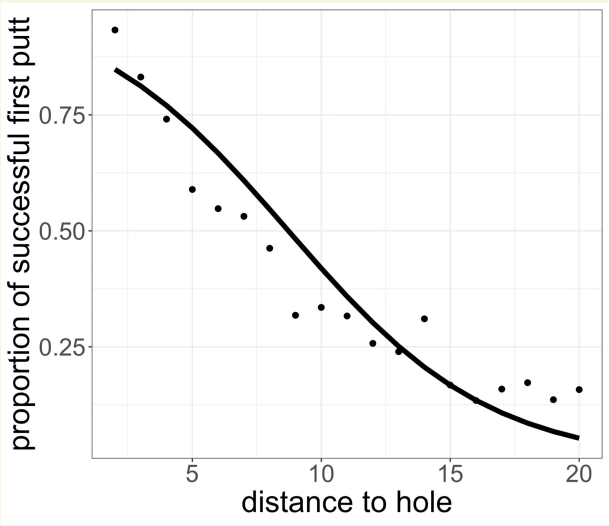
i = index of i^{th} putt in our dataset

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

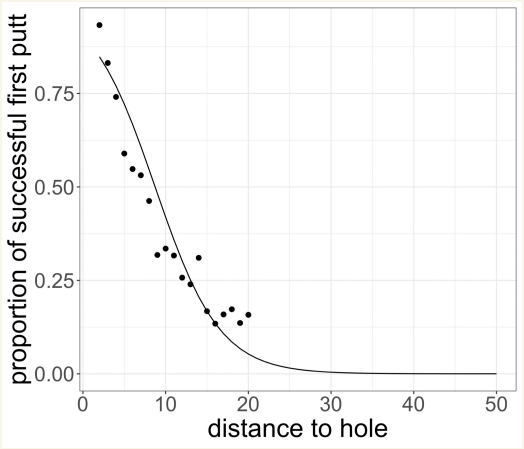
X_i = distance to hole of i^{th} putt

Logistic Regression Model

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



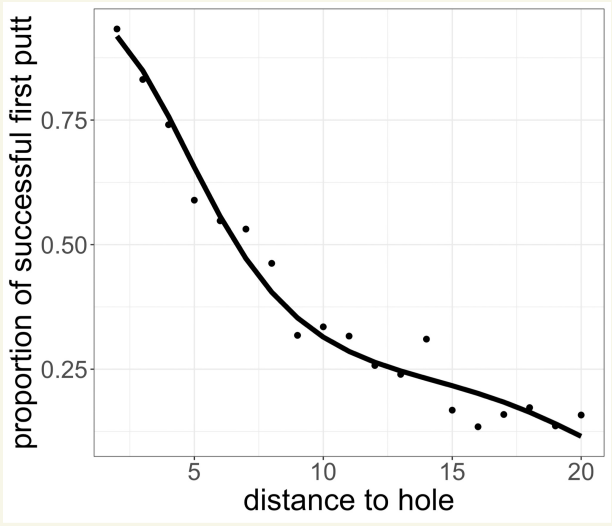
Looks decent!



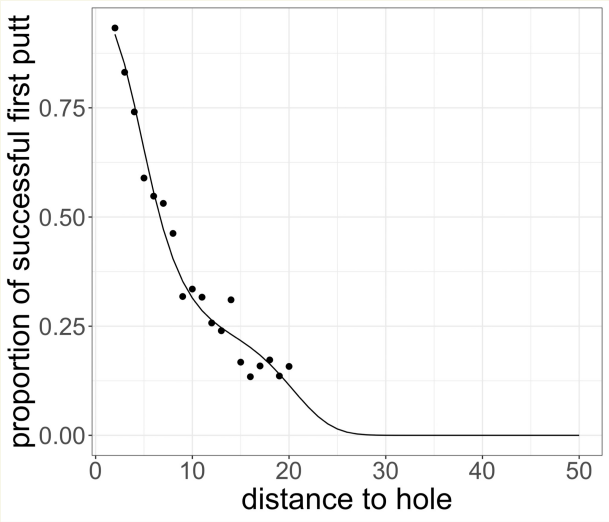
It extrapolates well because we forced our predictions to lie in $[0,1]$.

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

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



Solid!



this is good enough.

Q Create NCAA Mens Basketball Power Ratings which adjust for strength of schedule and Home Court by accounting for who beat whom, but ignoring score differential.

Bradley-Terry Logistic Regression Model

β_i = team i's (unknown) power rating
β_j = team j's (unknown) power rating

$$P_{ij} = P(\text{team } i \text{ beats team } j) = \frac{e^{\beta_i}}{e^{\beta_i} + e^{\beta_j}}$$

$$= \frac{1}{1 + e^{-(\beta_i - \beta_j)}}$$

* Add in Home court advantage:

i = Home team
j = away team
β_0 = Home court advantage

$$P_{ij} = \frac{1}{1 + e^{-(\beta_0 + \beta_i - \beta_j)}}$$

Data i is the index of the i^{th} match

$H(i)$ is the home team of match i

$A(i)$ is the away team of match i

y_i is 1 if home team wins, else 0

Scheduling matrix X, as before, is

$X_{i1} = 1$ (intercept)

$j > 1$, $X_{ij} = X[\text{row } i, \text{ column } j]$

$= X[\text{match } i, \text{ team } j-1]$

$$= \begin{cases} 1 & \text{if } H(i) = j-1 \\ -1 & \text{if } A(i) = j-1 \\ 0 & \text{else} \end{cases}$$

Model

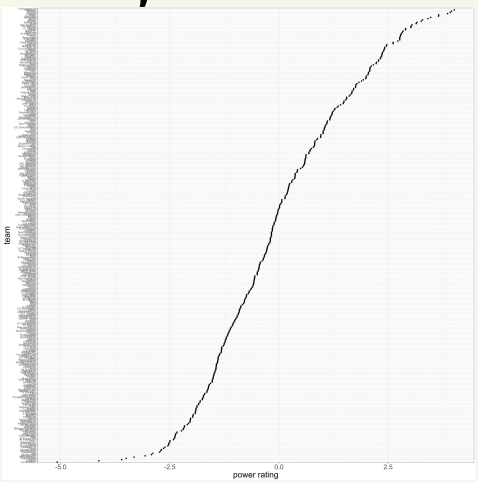
$$\mathbb{P}(y_i = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_{H(i)} - \beta_{A(i)})}}$$

$$= \frac{1}{1 + e^{-X_i^T \beta}}$$

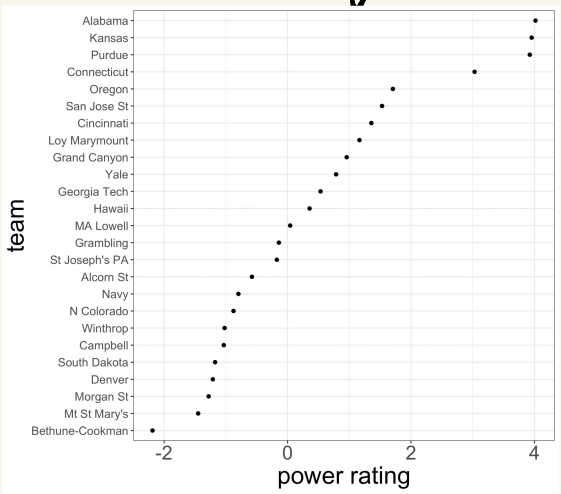
Run logistic regression using data (X, y).
Estimated coefficients β are our power scores!

```
### get power ratings using Bradley Terry (logistic regression)
bradley_terry = glm(df_ncaamb2$Win ~ X + 0, family="binomial")
power_ratings = bradley_terry$coefficients
```

Too many teams to see.



Some Power Ratings:




```
> tibble(teams=colnames(X), power_ratings=unname(power_ratings)) %>%
+   drop_na() %>%
+   arrange(power_ratings) %>%
+   head(5)
# A tibble: 5 x 2
  teams        power_ratings
  <chr>                <dbl>
1 LIU Brooklyn         -5.08
2 Hartford             -4.13
3 IUPUI                -3.60
4 Presbyterian         -3.50
5 WI Green Bay         -3.32
> tibble(teams=colnames(X), power_ratings=unname(power_ratings)) %>%
+   drop_na() %>%
+   arrange(-power_ratings) %>%
+   head(5)
# A tibble: 5 x 2
  teams    power_ratings
  <chr>            <dbl>
1 Alabama           4.02
2 Kansas            3.95
3 Purdue            3.92
4 Houston           3.86
5 Texas             3.66
```

Intercept = .23
Home court Advantage

Recall the power ratings which used linear regression on score differential:

```
> tibble(teams=colnames(X), power_ratings=unname(power_ratings)) %>%
+   drop_na() %>%
+   arrange(power_ratings) %>%
+   head(5)
# A tibble: 5 x 2
  teams        power_ratings
  <chr>                <dbl>
1 LIU Brooklyn         -26.3
2 Hartford             -24.9
3 WI Green Bay         -21.8
4 IUPUI                -20.3
5 MS Valley St         -18.9
> tibble(teams=colnames(X), power_ratings=unname(power_ratings)) %>%
+   drop_na() %>%
+   arrange(-power_ratings) %>%
+   head(5)
# A tibble: 5 x 2
  teams     power_ratings
  <chr>             <dbl>
1 Alabama            21.2
2 Houston            20.5
3 UCLA               19.4
4 Tennessee          19.1
5 Texas              18.5
```

Why are they different?

Takeaway

Use linear regression to predict a Real number.
Use logistic regression to predict a probability in $[0,1]$.

- Bradley-Terry model uses logistic Regression to create power ratings $\hat{\beta}$,

$$P(y_i = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_{H(i)} - \beta_{A(i)})}}$$

- Use gradient descent or something similar to solve for β

- ELO has the same model, but uses <u>one</u> iteration of gradient descent to update $\hat{\beta}^{(t+1)}$ after every game,

$$\vec{\beta}^{(t+1)} = \vec{\beta}^{(t)} + K \cdot \sum_{i=1}^{n} \left(y_i - \phi(x_i^T \beta)\right) \vec{x}_i$$

HW Implement ELO on our NCAA Mens Basketball dataset and compare the results to Bradley Terry.