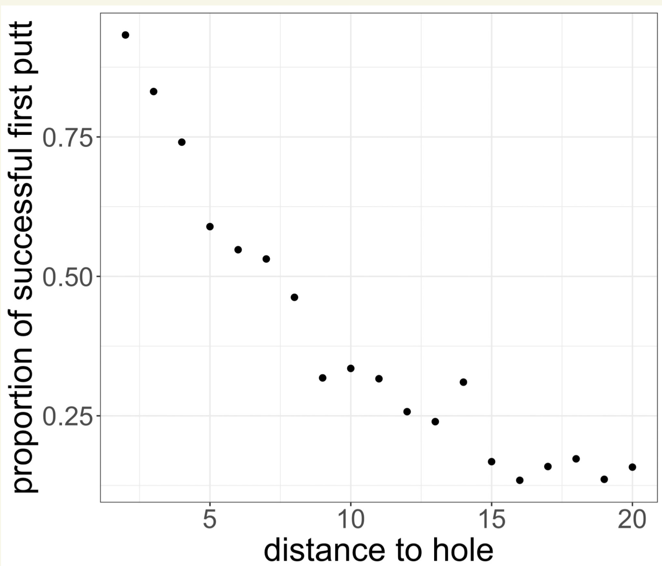
Regression, Part 3: Logistic Regression

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?

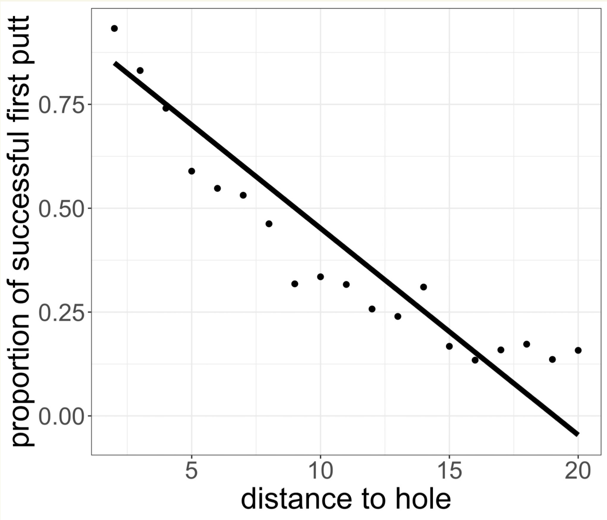
Variables

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.

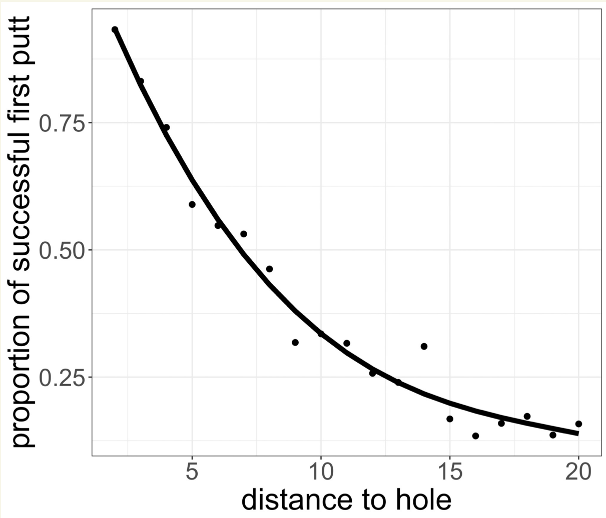


Not a great fit.

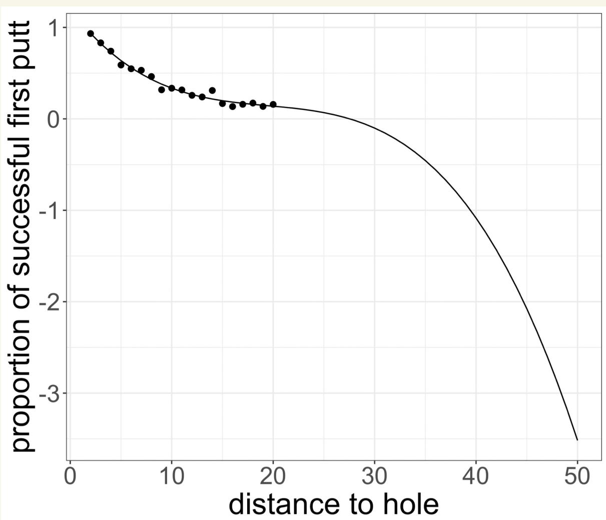
Model (Cubic Regression)

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

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



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

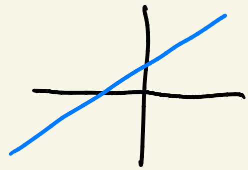


Not able to extrapolate when $X_i > 20$ ...

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

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

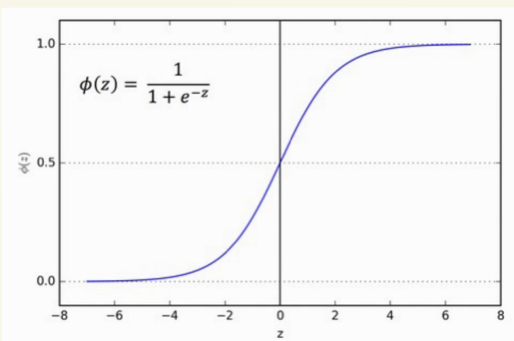
OLR: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$



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 <u>probability</u> directly,

$$\hat{p}_i = \hat{P}(y_i = 1) = \frac{1}{1+e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_i)}}$$

Logistic Regression <u>Model</u>

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

$$y_i \sim \text{Bernoulli}(p_i) = \begin{cases} 1 & \text{w.p. } p_i \\ 0 & \text{w.p. } 1-p_i \end{cases}$$

Before estimating the coefficients β, let's look at another example.

<u>Q</u> 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 Power Scores

Logistic Regression Power Scores.

Schedule matrix X from yesterday

game i, Home team $H(i)$, Away team $A(i)$,

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

Outcomes win/loss y

$$y_i = \begin{cases} 1 & \text{if } H(i) \text{ wins} \\ 0 & \text{if } H(i) \text{ loses} \end{cases}$$

Model

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

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

$y_i \sim \text{Bernoulli}(p_i)$

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 <u>log loss</u>, i.e. the <u>cross entropy loss</u>,

$$\boxed{\begin{aligned} & L(\beta) = \frac{-1}{n}\sum_{i=1}^{n} y_i \log P_i + (1-y_i)\log(1-P_i) \\ & \text{where } P_i = \mathbb{P}(y_i = 1 \mid x_i, \beta) = \frac{1}{1+e^{-x_i^T\beta}} \end{aligned}}$$

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

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

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

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

by Chain Rule

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

Hence

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

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

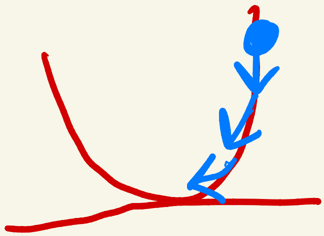
$$= -\frac{1}{n}\sum_{i=1}^{n}\left[y_i(1-p_i)X_i - (1-y_i)\,p_i X_i\right]$$

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

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

$\blacksquare$

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



Gradient Descent

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

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

K = learning Rate

Anyone Recognize K ?? $\boxed{\text{ELO}}$

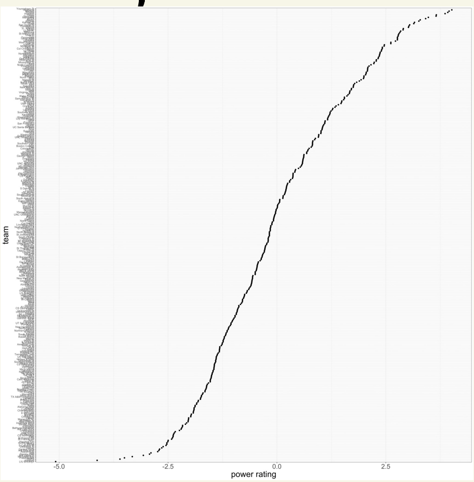
$$* \quad \text{ELO}^{(t+1)} = \text{ELO}^{(t)} + K\left(\mathbb{1}(\text{win}) - \mathbb{P}(\text{win})\right)$$

One iteration of gradient descent in logistic Regression
for Bradley Terry Power scores
is one ELO update!

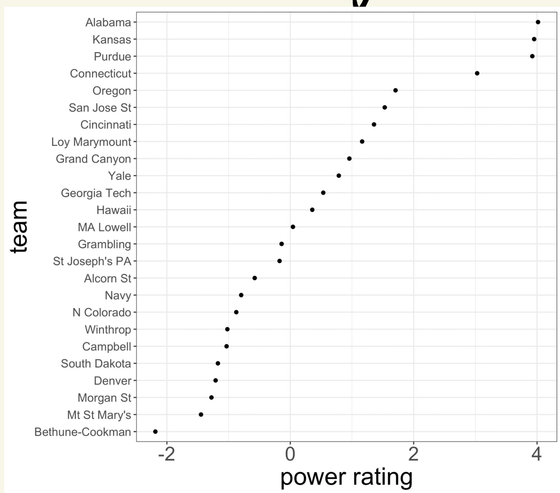
So, now we can estimate β in logistic Regression!!
Done automatically in R.
Estimated coefficients β are our Power scores!

```r
### 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:




```r
> 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

* Back to Golf:

Variables

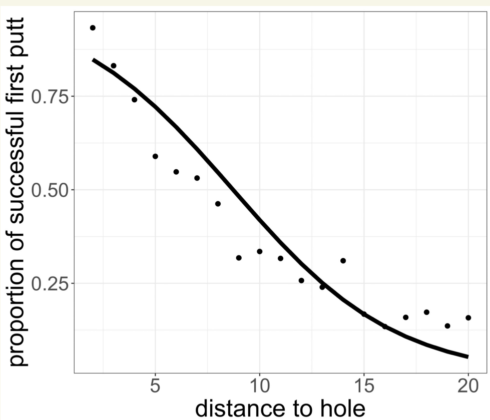
i = index of i^{th} putt in our dataset

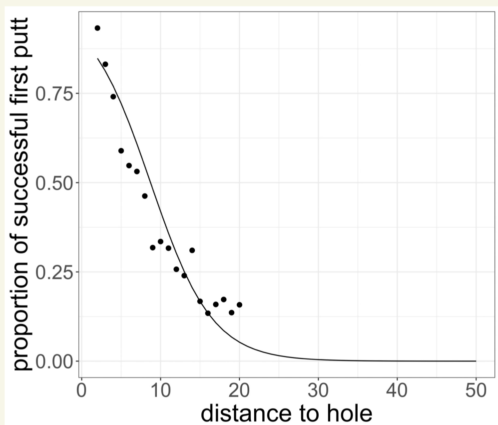
Y_i = 1 if putt is sunk, else 0

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

Logistic Regression Model

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

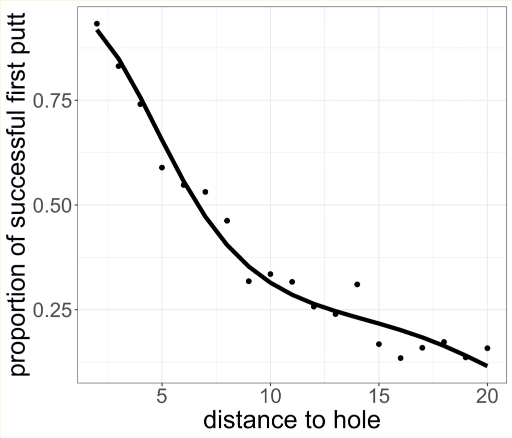


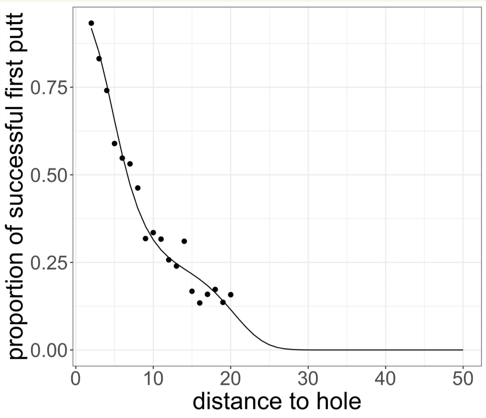


It extrapolates well because we forced our predictions to lie in [0,1].

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

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





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

Takeaway

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