Multivariable Linear Regression

NCAA Men's Basketball Power Scores

We are given a dataset of the game results of each game in 2022–2023

Season	WLoc	WTeamName	LTeamName	ScoreDiff	WScore	LScore
<dbl>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
2023	H	DePaul	Loyola MD	6	72	66
2023	H	Duke	Jacksonville	27	71	44
2023	A	Evansville	Miami OH	-4	78	74
2023	A	FL Gulf Coast	USC	-13	74	61
2023	H	Florida	Stony Brook	36	81	45
2023	H	Florida Intl	Houston Chr	11	77	66

i = index of the game

$H(i)$ = home team index in game i

$A(i)$ = away team

y_i = score diff

Suppose each team j has a latent (unobserved) power score β_j and we model the outcome of a game (score diff) by

$$y_i = \beta_{H(i)} - \beta_{A(i)} + \beta_0 + \varepsilon_i$$

ε_i is noise $\mathbb{E}\varepsilon_i = 0$ independent

β_0 is home court advantage

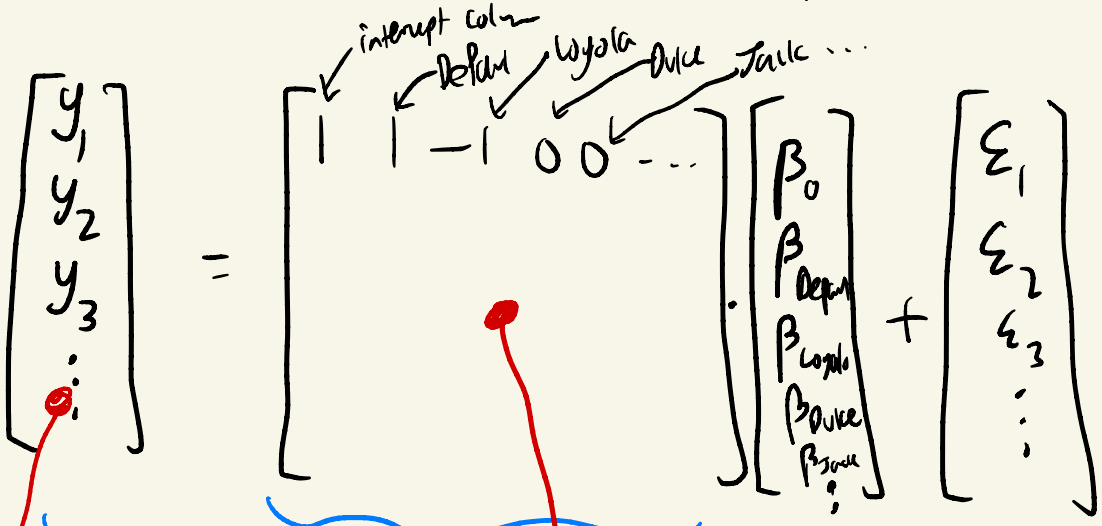
$$y_i = X_i \cdot \beta + \varepsilon_i$$

$i=1,$ $y_1 = \beta_0 + \beta_{Defun} - \beta_{Loyola} + \varepsilon_1$

$y_2 = \beta_0 + \beta_{Duke} - \beta_{Jack} + \varepsilon_2$

$\vdots$ $\qquad\qquad$ $\vdots$

In matrix-vector form:



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

outcomes (data)

scheduling matrix
↳ data

```
> df_ncaamb2[1:5,]
# A tibble: 5 × 7
  Season WTeamName   LTeamName    WScore LScore WLoc  ScoreDiff
   <dbl> <chr>       <chr>         <dbl>  <dbl> <chr>      <dbl>
1   2023 Abilene Chr Jackson St       65     56 H              9
2   2023 Akron       S Dakota St      81     80 H              1
3   2023 Alabama     Longwood         75     54 H             21
4   2023 Arizona     Nicholls St     117     75 H             42
5   2023 Arizona St  Tarleton St      62     59 H              3
> X[1:5,c(1:5,131)]
     (Intercept) Abilene Chr Air Force Akron Alabama Jackson St
[1,]           1           1         0     0       0          -1
[2,]           1           0         0     1       0           0
[3,]           1           0         0     0       1           0
[4,]           1           0         0     0       0           0
[5,]           1           0         0     0       0           0
```

How do we estimate the coefficients (the power scales) β from the data (X, y)?

In Simple Linear regression, we estimated (β_0, β_1) by minimizing RSS and we do the same thing here

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\, RSS(\beta) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - (\beta_{H(i)} - \beta_{A(i)} + \beta_0)\right)^2$$

would like to use the X matrix

$$\vec{\hat{\beta}} = \underset{\vec{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \vec{X}_{i,\bullet}\vec{\beta}\right)^2$$

i^{th} Row of X

$$(X\beta)_i = x_i \beta$$

$$x_i^T = [1 \quad 1 \quad -1]$$

1 → intercept; 1 → H(i); −1 → A(i)

Set gradient equal to 0 and solve

$$\nabla_\beta RSS(\beta) = 0 \quad \text{and solve}$$

$$\boxed{\hat{\beta} = (X^T X)^{-1} X^T y}$$

$X_{n \times K}$

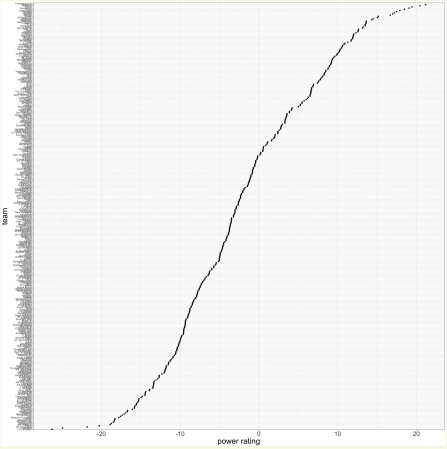
$$\vec{y} = X\vec{\beta} + \vec{\varepsilon}$$

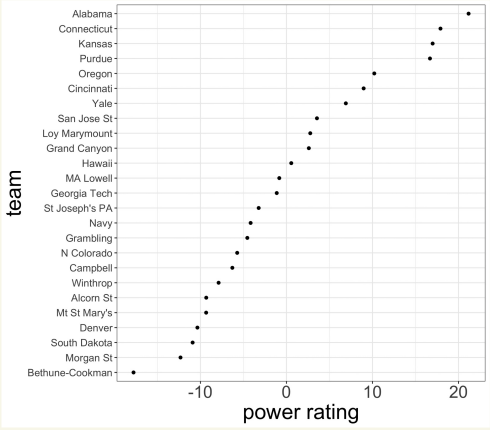
$$X^T y = X^T X \beta$$

$$(X^T X)^{-1} X^T y = \beta$$

So for our NCAA basketball power study model $y = X \cdot \beta + \varepsilon$ and how we can estimate β.

```
### get power ratings using multivariable linear regression
power_ratings_model = lm(df_ncaamb2$ScoreDiff ~ X + 0)
power_ratings = power_ratings_model$coefficients
```






```
> 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
```

Intercept $\hat{\beta}_0 = 2$

Being at home increases the predicted score diff by 2 pts above being on a neutral court.

Expected Points in American Football

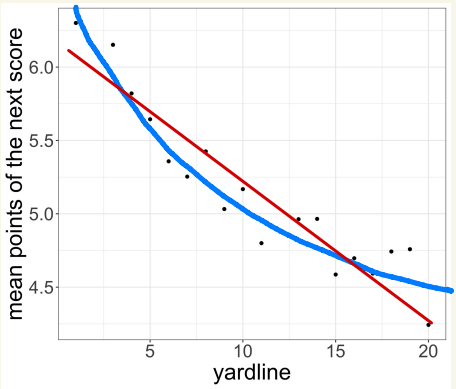
We are given a dataset of NFL plays

- i = index i^{th} play
- ydl_i = yardline
- y_i = net points of the next score in the half $\in \{7, 3, 2, 0, -2, -3, -7\}$

Define expected point (EP) by a model

$$\mathbb{E}[y_i] = f(ydl_i)$$

Plot



might not be linear...

empirical mean

Begin with just the redzone for simplicity.

How can we use linear regression to capture a relationship that is not linear?

data transformation

Linear model $y_i = \beta_0 + \beta_1 \cdot ydl_i + \varepsilon_i$ (y_i: pts)

Quadratic model $y_i = \beta_0 + \beta_1 \cdot ydl_i + \beta_2 \cdot ydl_i^2 + \varepsilon_i$

$\beta_2 \cdot ydl_i^2 = \beta_2 \cdot X_{2i}$

use multivariable regression with new features that we created transformed from the original features

$\beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \dots$

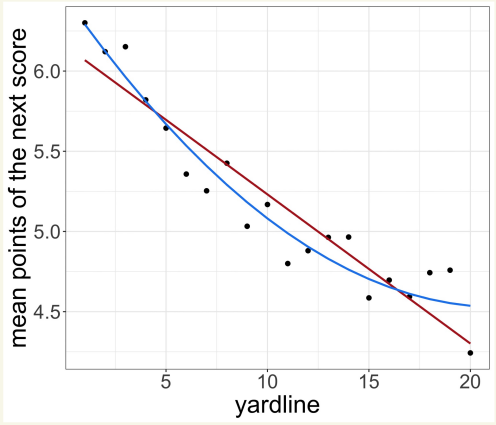
$\hat{\beta} = (X^T X)^{-1} X^T y$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & ydl_1 & ydl_1^2 \\ 1 & ydl_2 & ydl_2^2 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$\vec{y} = X \cdot \vec{\beta} + \vec{\varepsilon}$

```
> m_ep_linear = lm(data=D3r, pts_next_score ~ yardline_100)
> m_ep_linear

Call:
lm(formula = pts_next_score ~ yardline_100, data = D3r)

Coefficients:
 (Intercept)  yardline_100
     6.16098      -0.09299

> ### quadratic model
> m_ep_quad = lm(data=D3r, pts_next_score ~ yardline_100 + I(yardline_100^2))
> m_ep_quad

Call:
lm(formula = pts_next_score ~ yardline_100 + I(yardline_100^2),
    data = D3r)

Coefficients:
      (Intercept)       yardline_100  I(yardline_100^2)
         6.467712          -0.180798           0.004212
```

$y \sim ydl + ydl^2$



NFL Draft Expected Value Curves

We are given a dataset of NFL draft picks

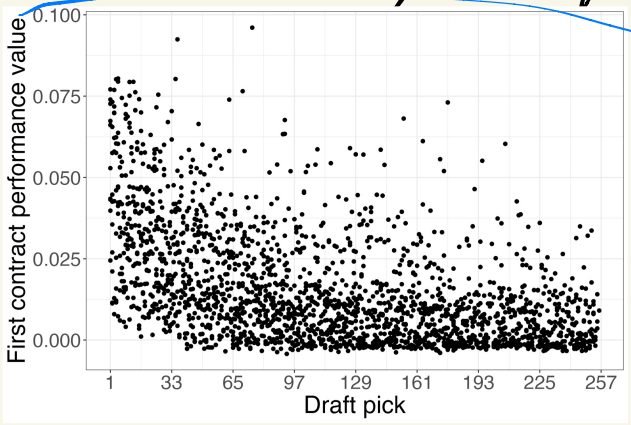
- i = i^{th} draft pick in our dataset
- X_i = draft pick number $\in \{1, 2, .., 256\}$
- y_i = player who was drafted's first contract performance "value"

player_id	player_name	year	t	draft_pos	firstContractPerformanceValue
40688	A.J. Bouye	2013	1	NA	1.659893e-02
42410	A.J. Cann	2015	1	67	3.541377e-02
35558	A.J. Edds	2011	1	119	-7.510774e-04
37077	A.J. Green	2011	1	4	8.018761e-02
30819	A.J. Hawk	2006	1	5	4.689117e-02
35863	A.J. Jefferson	2010	1	NA	4.419914e-03
38560	A.J. Jenkins	2012	1	30	5.734494e-03
40096	A.J. Klein	2013	1	148	1.305431e-02
30972	A.J. Nicholson	2006	1	157	-2.287106e-03

Model $\mathbb{E}[y_i] = f(x_i)$

ignore position
ignore everything else

Before specifying a specific model, plot



$

salary cap

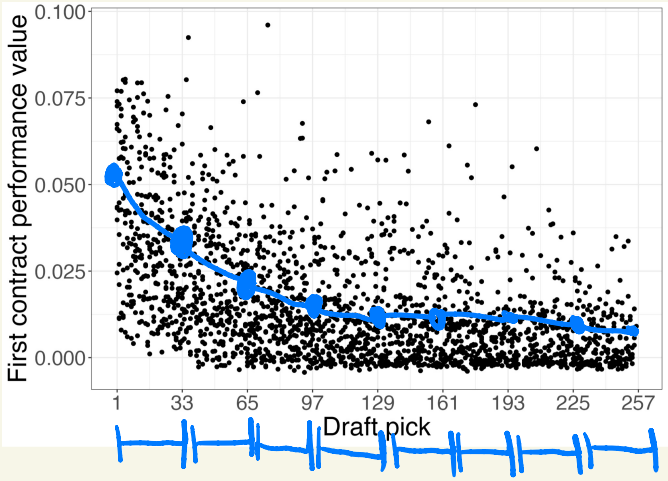
the draft is very noisy.
the expected value curve is nonlinear.

Convex: the dropoff in value b/t picks t and t+1 decreases as t increases

Spline:

to model a general nonlinear shape, using a spline often works.

To fit a spline, you fit a separate polynomial (usually a cubic) to different subsections of the data



a spline also forces the curve to be smooth, even at the knots:

at each knot, we mandate that the curve has

- the same left y value and right y value
- the same left derivative and right derivative
- the same left 2nd derivative and right 2nd derivative

Suppose we fit a cubic spline with one knot at $x = K$
(e.g. $K = 129$ in middle of draft)
we model $y_i = f(x_i | \beta) + \varepsilon_i$
where f is a spline and β are the spline parameters,

$$f(x|\beta) = \begin{cases} \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 & \text{if } x \le K \\ \beta_4 + \beta_5 x + \beta_6 x^2 + \beta_7 x^3 & \text{if } x > K \end{cases}$$

Such that
$$\begin{cases} \lim_{x \to K^-} f(x|\beta) = \lim_{x \to K^+} f(x|\beta) \\ \lim_{x \to K^-} f'(x|\beta) = \lim_{x \to K^+} f'(x|\beta) \\ \lim_{x \to K^-} f''(x|\beta) = \lim_{x \to K^+} f''(x|\beta) \end{cases}$$

enforce
$$\begin{cases} \beta_0 + \beta_1 K + \beta_2 K^2 + \beta_3 K^3 = \beta_4 + \beta_5 K + \beta_6 K^2 + \beta_7 K^3 \\ \beta_1 + 2\beta_2 K + 3\beta_3 K^2 = \beta_5 + 2\beta_6 K + 3\beta_7 K^2 \\ 2\beta_2 + 6\beta_3 K = 2\beta_6 + 6\beta_7 K \end{cases}$$

$$\beta_7 = (2\beta_2 + 6\beta_3 K - 2\beta_6) / (6K)$$

β_7 is dead

each knot in a cubic spline "kills" 3 parameters

with 1 knot, we need to estimate 5 parameters

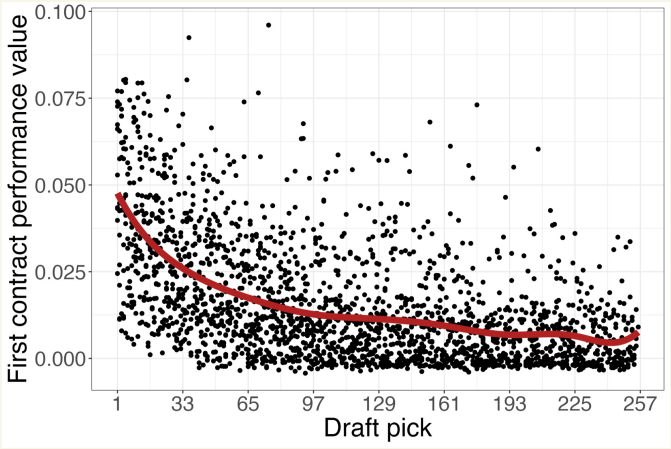
$$f(x|\beta) = \begin{cases} \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 & \text{if } x \le K \\ \beta_4 + \beta_5 x + \beta_6 x^2 + \beta_7 x^3 & \text{if } x > K \end{cases}$$

$$\gamma_1 \tilde{x}_{1i} + \gamma_2 \tilde{x}_{2i} + \gamma_3 \tilde{x}_{3i} + \gamma_4 \tilde{x}_{4i} + \gamma_5 \tilde{x}_{5i}$$

↳ the spline basis

```
draft_model1 = lm(
  firstContractPerformanceValue ~ splines::bs(draft_pos, degree=3, knots=seq(33,32*8,by=32)),
  data=df_draft
)
draft_model1
```

$y \sim \text{Spline}(X, \text{deg}=3, \text{knots} = (33, 65, \ldots))$




```
draft_model2 = lm(
  firstContractPerformanceValue ~ splines::bs(draft_pos, degree=3, df=5),
  data=df_draft
)
draft_model2
```

degrees of freedom