Regression, Part 2: Multivariable Linear Regression

Q Devise power scores for college basketball teams which take into account strength of schedule and the score differential of each game and home court advantage.

Variables

i is the index of the i^{th} game

team indices $\{1, \dots, N\}$

Y_i is the score differential of game i
(Home score minus away score)

$\beta_{H(i)}$ is the (unknown) power score of the Home team $H(i)$ of game i

$\beta_{A(i)}$ is same for away team $A(i)$

Model

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

mean-zero noise ε_i
so, $\mathbb{E}\varepsilon_i = 0$

2023 NCAA Mens Basketball DataFrame

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

$$Y_1 = \beta_0 + \beta_{DePaul} - \beta_{Loyola} + \varepsilon_1$$

$$Y_2 = \beta_0 + \beta_{Duke} - \beta_{Jacksonville} + \varepsilon_2$$

$$Y_3 = \beta_0 + \beta_{Miami} - \beta_{Evansville} + \varepsilon_3$$

$$\vdots$$

In Matrix-Vector FORM:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{array}{c} \begin{array}{ccccccc} \text{intercept} & \text{DePaul} & \text{Loyola} & \text{Duke} & \text{Jacksonville} & \text{Miami} & \text{Evansville} \end{array} \\ \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 0 & 1 & -1 & \\ & & & \vdots & & & & \end{bmatrix} \end{array} \begin{bmatrix} \beta_0 \\ \beta_{DePaul} \\ \beta_{Loyola} \\ \beta_{Duke} \\ \beta_{Jacksonville} \\ \vdots \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \end{bmatrix}$$

Vectorized Model

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

$\vec{y} = (y_1, \ldots, y_n)$ vector of observed score differentials

$\vec{\beta} = (\beta_0, \beta_{\text{depaul}}, \beta_{\text{Loyola}}, \ldots)$ vector of unknown power ratings, to be estimated

$\vec{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$ vector of mean zero noise $\mathbb{E}\varepsilon_i = 0$

$X =$ Scheduling matrix so,

$j=1$, $X_{i1} = 1$ (intercept term)

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

$$= \begin{cases} 1 & \text{if home team in game } i \text{ is team } j-1 \\ -1 & \text{if away team in game } i \text{ is team } j-1 \\ 0 & \text{else} \end{cases}$$

```
> 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
>
```

Remove the vector superscripts for simplicity: $\boxed{y = X\beta + \varepsilon}$

How do we estimate the coefficients (e.g., the power ratings) β from observed data (X, y)?

Recall that in simple linear regression, we estimated (β_0, β_1) by minimizing the Residual Sum of Squares.

Similarly, in multivariable linear regression we minimize the RSS,

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \; RSS(\beta)$$

$$= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - x_i^T \beta\right)^2$$

where x_i is the i^{th} row of X

and $x_i^T\beta = x_i \cdot \beta = x_{i1}\beta_0 + x_{i2}\beta_1 + \dots + x_{i(k+1)}\beta_k$

$$= \sum_{j=0}^{k} x_{i,j+1}\beta_j$$ is the dot product

$$= \underset{\beta}{\operatorname{argmin}} \; (y - X\beta)^T (y - X\beta)$$

in matrix form

$$= \underset{\beta}{\operatorname{argmin}} \; y^T y - 2\beta^T X^T y + \beta^T X^T X \beta$$

Multivariable Calulus: set the gradient equal to 0. The gradient is the analog of the derivative.

Gradient $\nabla_\beta f(\beta) = \nabla_\beta f(\beta_0, \dots, \beta_K)$

$$= \left(\frac{\partial f}{\partial \beta_0}, \dots, \frac{\partial f}{\partial \beta_K} \right)$$

is the vector of partial derivatives.

$$0 = \nabla_\beta RSS(\beta)$$

$$= \nabla_\beta \left(y^T y - 2\beta^T X^T y + \beta^T X^T X \beta \right)$$

$$= -2X^T y + 2(X^T X)\beta$$

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

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

This is the matrix form of linear multivariable regression!

HW suppose $K=1$ so $X = \begin{bmatrix} 1 & x_{11} \\ 1 & x_{21} \\ \vdots & \vdots \\ 1 & x_{n1} \end{bmatrix}$

and $\beta = (\beta_0, \beta_1)$; this is simple linear regression.

Show that $\hat{\beta} = (X^T X)^{-1} X^T y$ matches our previous simple linear regression formular for β_0, β_1.

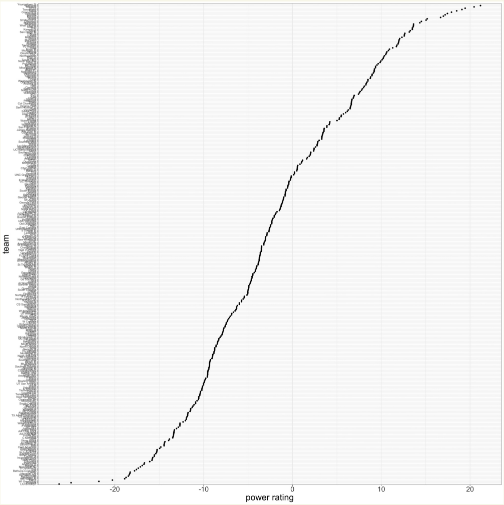
So, for our NCAA Basketball power Ratings model $y = X\beta + \varepsilon$, we now know how to estimate $\hat{\beta}$.

Let's Run the computation and see what it says!

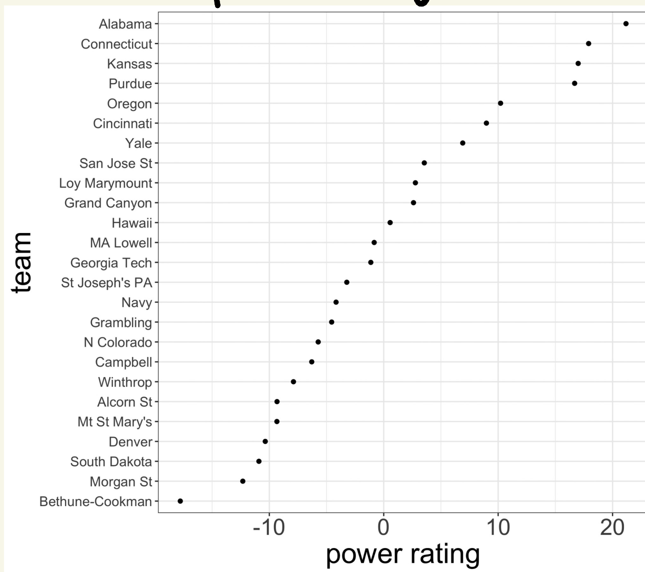
```
### get power ratings using multivariable linear regression
power_ratings_model = lm(df_ncaamb2$ScoreDiff ~ X + 0)
power_ratings = power_ratings_model$coefficients
```

Intercept $\hat{\beta}_0 = 2 \rightarrow$ Home Court Advantage!

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 × 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 × 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
```

* Biggest problem with this power rating method: only works well if there is a "path" from each team to each other team. Isolated groups of teams means we can't compare teams across groups. This is the problem with comparing international soccer teams in different leagues using power scores.

Q Use the power scores to simulate the NCAA tournament.

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

$$\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

Since $\mathbb{E}\varepsilon_i = 0$ still, we can use the same linear regression process to estimate β.

But now, we need to also estimate σ^2, the (unknown) variance of the noise.

* Assume we have $\hat{\sigma}^2$, an estimate of σ^2:

```
> sigma(power_ratings_model)
[1] 10.93223
```

* Then we can simulate the Result (score differential) of a basketball game by sampling $\varepsilon_i \sim N(0, \hat{\sigma}^2)$

and then computing $y_i = \hat{\beta}_0 + \hat{\beta}_{H(i)} - \hat{\beta}_{A(i)} + \varepsilon_i$.

* We can then simulate the March Madness tournament by simulating each of the 65 games.

* How to estimate σ^2:

Error $\varepsilon_i = y_i - x_i^T \beta$

Residual $\hat{\varepsilon}_i = y_i - x_i^T \hat{\beta}$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{i=1}^{n} \hat{\varepsilon}_i^2$$

K = # columns of X excluding intercept

Thm $\mathbb{E}[\hat{\sigma}^2] = \sigma^2$ (unbiased)

HW Prove it. OR ask me to do it.

HW OR, Prove it in the case of simple linear regression $K=1$

HW Create power scores and then simulate the March Madness tournament.

Q Predict 400 meter dash time from a database of previous races, which includes runner names.

Variables i = index of i^{th} runner-race in the dataset

Y_i = race time of i^{th} runner-race

$X_{i0} = 1$

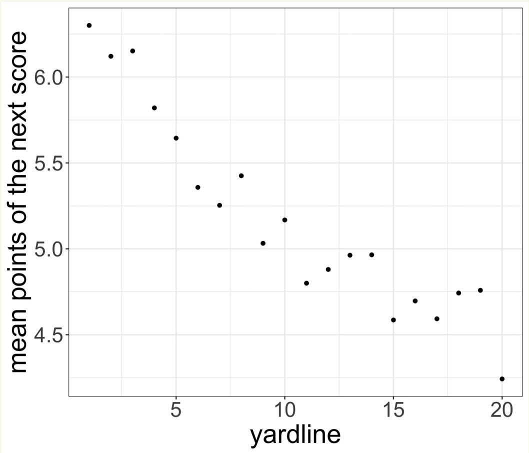
$X_{ik} = 1$ if player k is the runner in the i^{th} row, else 0

Model $Y_i = \beta_0 + \sum_{j=1}^{K} X_{ij}\beta_j + \varepsilon_i$, $\mathbb{E}\varepsilon_i = 0$

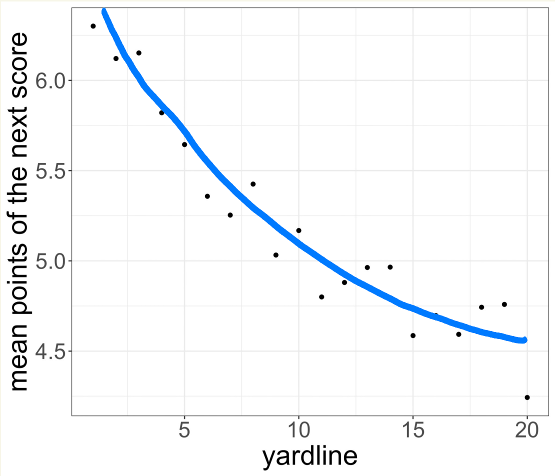
We now know how to estimate using linear regression, $\hat{\beta} = (X^T X)^{-1} X^T y$.

Q Estimate the expected points of the next score in a half in American football as a function of yardline, in the Redzone.

Generally, it is smart to begin with plotting



The Relationship looks quadratic, not linear



How can we use linear regression to capture a nonlinear relationship??

Data transformations!

<u>Variables</u> i = index of the i^{th} play in our dataset

Y_i = points of the next score in the half after play i relative to the team with possession (a number in $\{7,-7,3,-3,2,-2,0\}$)

X_i = yardline of play i (X_i yards from opponent's endzone)

<u>Linear Model</u> $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$

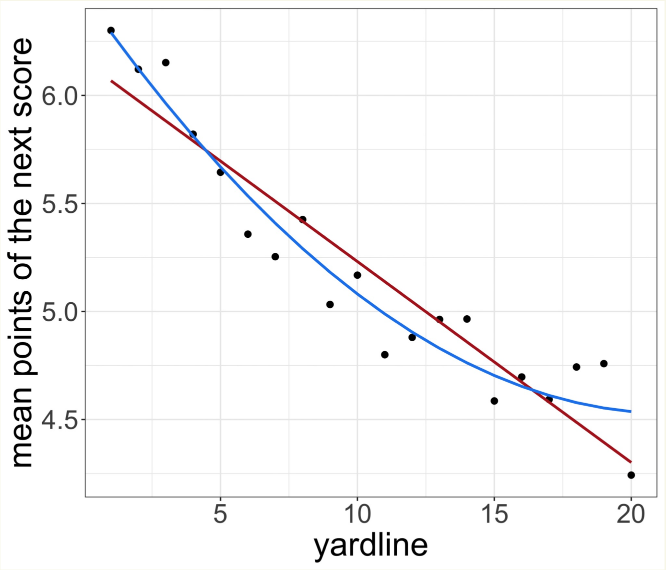
where $\mathbb{E}\varepsilon_i = 0$ (mean zero)

<u>Quadratic Model</u> $Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$

where $\mathbb{E}\varepsilon_i = 0$

The 2^{nd} model has an additional parameter, β_2.
Next time: how to estimate a regression model with $K \geq 3$ parameters.

```
> 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
```



Quadratic model looks better.

HW Predict an NFL player's 2^{nd} contract value (a proxy for his on field value) as a function of his draft position.

This relationship should be highly nonlinear!! Compare a linear model to a nonlinear model (say, a quartic polynomial), using linear regression to fit both.

What fits better?

If we had more time: Splines

Model any continuous shape