

# **PSC4375: Linear Regression**

## **Week 5: Lecture 10 (& 11)**

Prof. Weldzius

Villanova University

Slides Updated: 2025-02-19

# Can betting markets help us predict elections?

- Data from an online betting company Intrade
- People trade contracts such as “Obama to win the electoral votes in Florida”
- Market prices of each contract fluctuate based on its sales
- Why might we expect betting markets like Intrade to accurately predict outcomes of elections?

# Linear Regression: Prediction using bivariate relationships

- Goal: what's our best guess about  $Y_i$  if we know what  $X_i$  is?
  - What's our best guess about election margins if we know the market's margins?
- Terminology:
  - **Dependent/outcome variable**: what we want to predict (election margin).
  - **Independent/explanatory variable**: what we're using to predict (market margin).

# We'll use two datasets: intrade08.csv & pres08.csv

Name	Description
day	Date of the session
statename	Full name of each state (including DC in 2008)
state	Abbreviation of each state (including DC in 2008)
PriceD	Closing price (predicted vote share) of Democratic Nominee's market
PriceR	Closing price (predicted vote share) of Republican Nominee's market
VolumeD	Total session trades of Democratic Party Nominee's market
VolumeR	Total session trades of Republican Party Nominee's market

- intrade08.csv: Each row represents daily trading information about the contracts for either the Democratic or Republican Party nominee's victory in a particular state.

# Presidential voting data from 2008

Name	Description
<code>state.name</code>	Full name of state (only in pres2008)
<code>state</code>	Two letter state abbreviation
<code>Obama</code>	Vote percentage for Obama
<code>McCain</code>	Vote percentage for McCain
<code>EV</code>	Number of electoral college votes for this state

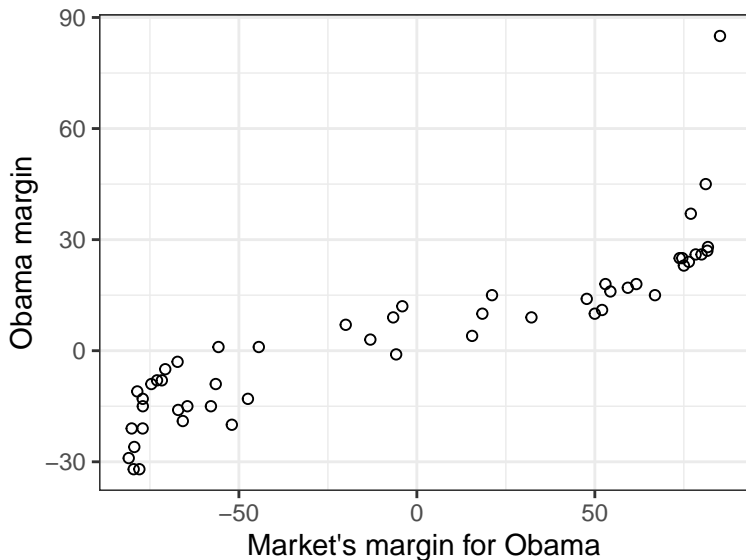
# Predicting Elections Using Betting Markets and Linear Models

- Load the data

```
library(tidyverse)
intrade08 <- read.csv("../data/intrade08.csv")
pres08 <- read.csv("../data/pres08.csv")

## merge datasets and calculate margins for DV and IV
intresults08 <- inner_join(intrade08,pres08) %>%
  mutate(obama.intmarg = PriceD - PriceR,
         obama.actmarg = Obama - McCain)
```

# Plot bivariate relationship



# Using a line to predict

- Prediction: for any value of  $X$ , what's the best guess about  $Y$ ?
  - Need a function  $y = f(x)$  that maps values of  $X$  into predictions.
  - **Machine learning**: fancy ways to determine  $f(x)$
- Simplest possible way to relate two variables: a line.

$$y = mx + b$$

- Problem: for any line we draw, not all the data is on the line.
  - Some points will be above the line, some below.
  - Need a way to account for **chance variation** away from the line.



# Linear Regression Model

- Model for the line of best fit

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \times X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

- **Coefficients/parameters** ( $\alpha, \beta$ ): true unknown intercept/slope of the line of best fit
- **Chance error** ( $\epsilon_i$ ): accounts for the fact that the line doesn't perfectly fit the data.
  - Each observation allowed to be off the regression line
  - Chance errors are 0 on average
- Useful fiction: this model represents the **data generating process**
  - George Box: “all models are wrong, some are useful”

# Interpreting the Regression Line

$$Y_i = \alpha + \beta \times X_i + \epsilon_i$$

- **Intercept** *alpha*: average value of  $Y$  when  $X$  is 0
  - Average Obama margin when market's margin is 0.
- **Slope** *beta*: average change in  $Y$  when  $X$  increases by one unit
  - Average increase in Obama margin for each additional margin increase by the market.
- But we don't know  $\alpha$  or  $\beta$ . How can we estimate them? Next time. . .
  - Or now if we still have time!

# Linear Regression Model (skip if same day)

- Model for the line of best fit

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \times X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

- **Coefficients/parameters**  $(\alpha, \beta)$ : true unknown intercept/slope of the line of best fit
- **Chance error**  $(\epsilon_i)$ : accounts for the fact that the line doesn't perfectly fit the data.
  - Each observation allowed to be off the regression line
  - Chance errors are 0 on average

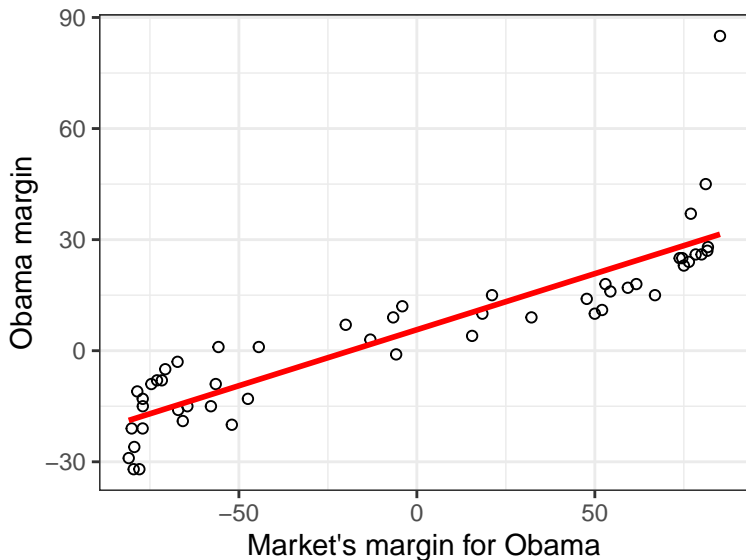
# Estimate coefficients

- Parameters:  $\alpha, \beta$ 
  - Unknown features of the data-generating process.
  - Chance error makes these impossible to observe directly.
- Estimates:  $\hat{\alpha}, \hat{\beta}$ 
  - An estimate is our best guess about some parameter.
- Regression line:

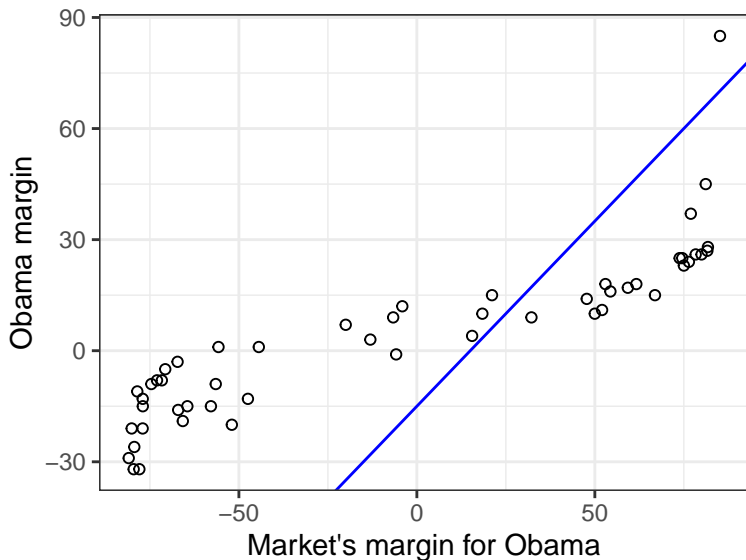
$$\hat{Y} = \hat{\alpha} + \hat{\beta} * x$$

- Average value of  $Y$  when  $X$  is  $x$
- Represents the best guess or **predicted value** of the outcome at  $x$ .

# Line of best fit



# Why not this line?



# Least squares

- How do we figure out the best line to draw?
  - **Fitted/predicted value** for each observation:  $\hat{Y} = \hat{\alpha} + \hat{\beta} \times X_i$
  - **Residual/prediction error**:  $\hat{\epsilon}_i = Y_i - \hat{Y}$
- Get these estimates by the **least squares method**
- Minimize the **sum of the squared residuals** (SSR):

$$SSR = \sum_{i=1}^n \hat{\epsilon}_i^2 = \sum_{i=1}^n (Y_i - \hat{\alpha} - \hat{\beta}X_i)^2$$

- Finds the line that minimizes the magnitude of the prediction errors!

# Linear Regression in R

- R will calculate least squares line for a data set using `lm()`
  - Syntax: `lm(y ~ x, data = mydata)`
  - `y` is the name of the dependent variable
  - `x` is the name of the independent variable
  - `mydata` is the data.frame where they live



# Linear Regression in R

```
fit <- lm(obama.actmarg ~ obama.intmarg, data = intresults08)
fit

##
## Call:
## lm(formula = obama.actmarg ~ obama.intmarg, data = intresults08)
##
## Coefficients:
##      (Intercept)      obama.intmarg
##           5.5681              0.2799
```

# Coefficients and fitted values

- Use `coef()` to extract estimated coefficients:

```
coef(fit)
```

```
##      (Intercept) obama.intmarg  
##      5.5681423      0.2799326
```

- R can show you each of the fitted values as well:

```
head(fitted(fit))
```

```
##           1           2           3           4           5           6  
## 5.568142 5.568142 5.568142 5.568142 5.568142 5.568142
```

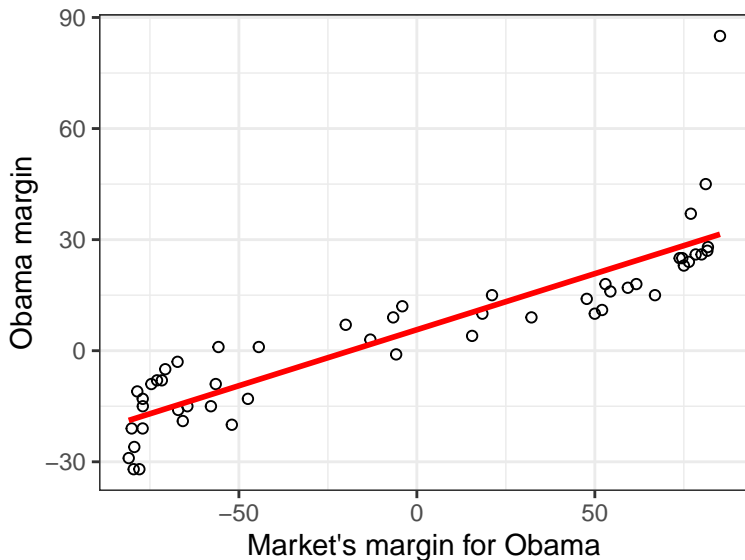
# Properties of least squares

- Least squares line always goes through  $(\bar{X}, \bar{Y})$
- Estimated slope is related to correlation:

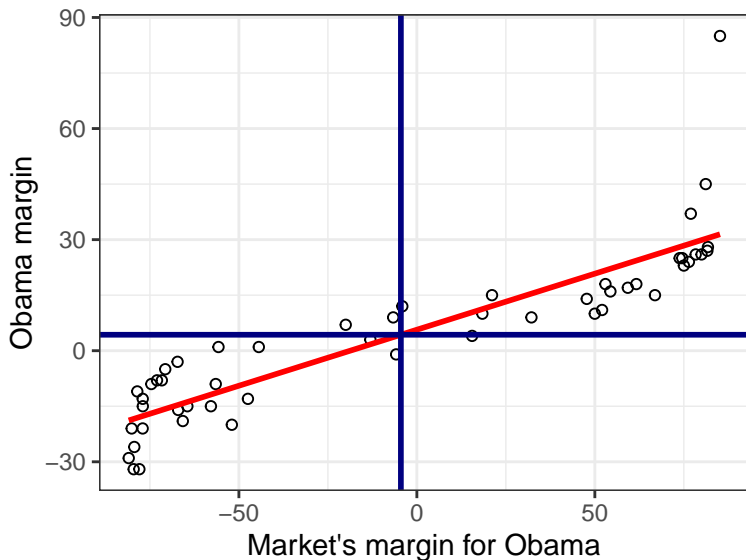
$$\hat{\beta} = (\text{correlation of } X \text{ and } Y) \times \frac{\text{SD of } Y}{\text{SD of } X}$$

- Mean of residuals is always 0

# Visual components of least squares



# Visual components of least squares



# Visual components of least squares

