THE UNIVERSITY OF TEXAS AT AUSTIN
McCOMBS SCHOOL OF BUSINESS

Model building: problems and fixing them

Lecture 14

STA 371G

Today's data set

We're going to look at a data set of newly hired managers:

- Salary (response)
- Manager rating
- Years of experience
- Years since graduation
- Origin (internal or external hire)

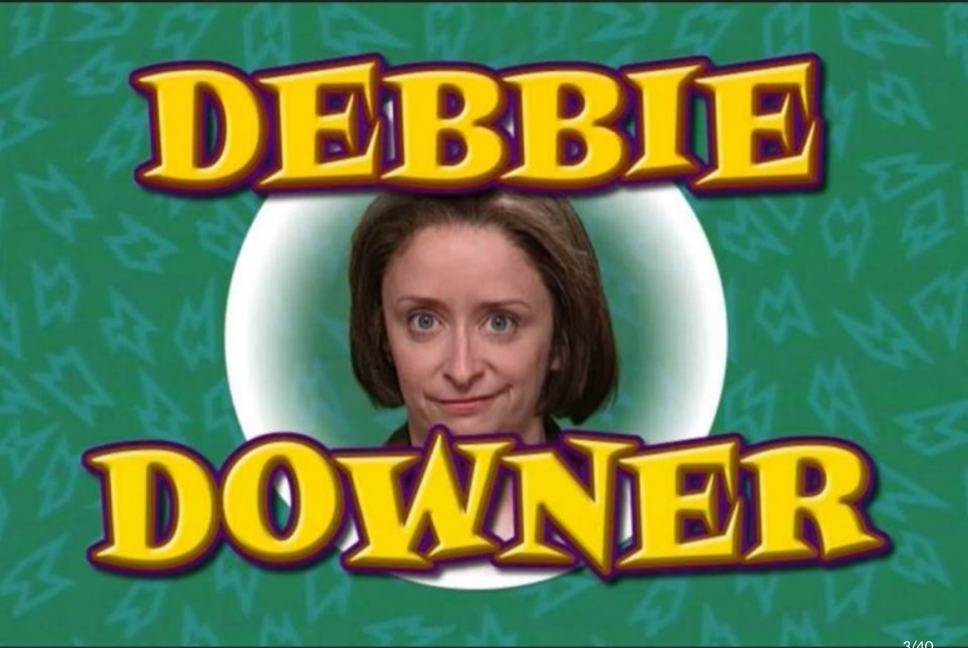
Data issues

Data scientists report that they spend 70% of their time on obtaining and cleaning the data. Only 30% is for statistical analysis.

Data issues

Data scientists report that they spend 70% of their time on obtaining and cleaning the data. Only 30% is for statistical analysis.

Never run a regression without exploring and cleaning the data first!


DEBBIE
DOWNER

The most common issues:

1. Outliers
2. Missing data
3. Multicollinearity
4. Highly influential points
5. Handling nonlinearity

1. Outliers
2. Missing data
3. Multicollinearity
4. Highly influential points
5. Handling nonlinearity

Exploring the data: Outliers

Boxplots are commonly used to find cases that might be outliers.

Let's start by looking at the Salary column.

```
boxplot(manager$Salary, xlab="Salary", horizontal=T)
```

Exploring the data: outliers

If a case is shown as an outlier on the boxplot (i.e., 1.5 IQR above Q3 or 1.5 IQR below Q1):

- It might be an error.
- It might represent a missing value or other situation. (Consult the codebook for the data set, if there is one!)
- It might be a true outlier.

Exploring the data: Outliers

```
subset(manager, Salary > 200)

    Salary MngrRating YearsExp YrsSinceGrad   Origin
146    511        6.1        2            2 Internal

subset(manager, Salary < 0)

    Salary MngrRating YearsExp YrsSinceGrad   Origin
121    -66        5.7        1            2 Internal
```

Exploring the data: Outliers

```
subset(manager, Salary > 200)

    Salary MngrRating YearsExp YrsSinceGrad   Origin
146    511        6.1        2            2 Internal

subset(manager, Salary < 0)

    Salary MngrRating YearsExp YrsSinceGrad   Origin
121    -66        5.7        1            2 Internal
```

We can deal with outliers in two ways.

- If the result of **errors in the data**, we can try to correct or omit.

Exploring the data: Outliers

```
subset(manager, Salary > 200)

    Salary MngrRating YearsExp YrsSinceGrad   Origin
146    511        6.1        2            2 Internal

subset(manager, Salary < 0)

    Salary MngrRating YearsExp YrsSinceGrad   Origin
121    -66        5.7        1            2 Internal
```

We can deal with outliers in two ways.

- If the result of **errors in the data**, we can try to correct or omit.
- If not, consider omitting, but report on them separately.

Exploring the data: Outliers

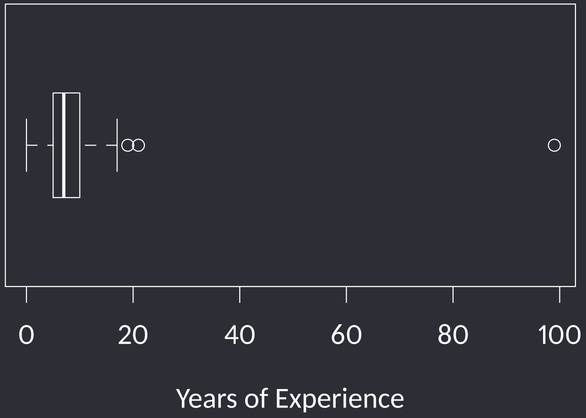
Let's omit the outliers by creating a new data set `mclean` that consists of the subset of the data where the salary is between $0 and $200,000.

```
mclean <- subset(manager, Salary > 0 & Salary < 200)
```

We'll use `mclean` for our analysis, but we won't destroy the original data set!

Exploring the data: Outliers

```
boxplot(mclean$YearsExp, xlab="Years of Experience",
  horizontal=T)
```



Exploring the data: Outliers

99 must be a code for missing entry in the Years of Experience variable!

Exploring the data: Outliers

99 must be a code for missing entry in the Years of Experience variable!

Let’s label all 99s as `NA` (“not available” — R’s code for missing data).

Exploring the data: Outliers

99 must be a code for missing entry in the Years of Experience variable!

Let's label all 99s as NA ("not available" — R's code for missing data).

```
mclean$YearsExp[mclean$YearsExp == 99] <- NA
```

1. Outliers
2. Missing data
3. Multicollinearity
4. Highly influential points
5. Handling nonlinearity

Exploring the data: Missing entries

Let's see if we have other missing data.

```
mclean[!complete.cases(mclean),]

    Salary MngrRating YearsExp YrsSinceGrad   Origin
103     75         NA        8            8 Internal
110     81         NA        9            9 External
124     73        5.9       NA            7 External
154     49        8.0        1            1     <NA>
```

Exploring the data: Missing entries

Let's see if we have other missing data.

```
mclean[!complete.cases(mclean),]

    Salary MngrRating YearsExp YrsSinceGrad   Origin
103     75         NA        8            8 Internal
110     81         NA        9            9 External
124     73        5.9       NA            7 External
154     49        8.0        1            1     <NA>
```

This isn't surprising—it is very common to have missing entries in your data. (The comma is needed so that we capture the full row.)

Exploring the data: Missing entries

There are two ways of dealing with missing data:

- Omit the rows that have missing entries in it.
- Try to predict values to fill the missing entries.

Omitting data is the easiest, but often not the best way, because you lose all the other information available in the same row.

Exploring the data: Missing entries

What should we replace the `NA`s in the Manager Rating and Years of Experience columns with?

Exploring the data: Missing entries

What should we replace the NAs in the Manager Rating and Years of Experience columns with?

The simplest way would be to use the averages in the respective columns.

```
mclean$MngrRating[is.na(mclean$MngrRating)] <-
  mean(mclean$MngrRating, na.rm=T)

mclean$YearsExp[is.na(mclean$YearsExp)] <-
  mean(mclean$YearsExp, na.rm=T)
```

Exploring the data: Missing entries

What should we replace the NAs in the Manager Rating and Years of Experience columns with?

The simplest way would be to use the averages in the respective columns.

```
mclean$MngrRating[is.na(mclean$MngrRating)] <-
  mean(mclean$MngrRating, na.rm=T)

mclean$YearsExp[is.na(mclean$YearsExp)] <-
  mean(mclean$YearsExp, na.rm=T)
```

A smarter and more advanced way is to predict the missing data from the other data (using regression!).

Exploring the data: Missing entries

What about the missing data for categorical variables?

Exploring the data: Missing entries

What about the missing data for categorical variables? Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

Exploring the data: Missing entries

What about the missing data for categorical variables? Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

This removes all the rows that contain missing entries (only the Origin column has missing entries in this case.)

Exploring the data: Missing entries

What about the missing data for categorical variables? Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

This removes all the rows that contain missing entries (only the Origin column has missing entries in this case.)

We could also predict the missing entries, or treat the missing entries as a seperate level (e.g. "Unknown").

Exploring the data: Missing entries

- While dealing with the missing data, we assume that the data is “Missing Completely at Random” (MCAR).

Exploring the data: Missing entries

- While dealing with the missing data, we assume that the data is "Missing Completely at Random" (MCAR).
- If this assumption does not hold (e.g. if the missing data mostly belongs to external hires), the model will be biased.

Exploring the data: Missing entries

- While dealing with the missing data, we assume that the data is "Missing Completely at Random" (MCAR).
- If this assumption does not hold (e.g. if the missing data mostly belongs to external hires), the model will be biased.
- Making predictions for missing data based on available data reinforces the existing relationships between variables, so impacts the standard error.

Exploring the data: Missing entries

- While dealing with the missing data, we assume that the data is "Missing Completely at Random" (MCAR).
- If this assumption does not hold (e.g. if the missing data mostly belongs to external hires), the model will be biased.
- Making predictions for missing data based on available data reinforces the existing relationships between variables, so impacts the standard error.
- If a lot of data is missing (e.g. more than 5%) for a particular variable, you may have to discard the whole column.

1. Outliers
2. Missing data
3. Multicollinearity
4. Highly influential points
5. Handling nonlinearity

Exploring the data: Multicollinearity

Multicollinearity exists whenever 2+ predictors in a regression model are moderately or highly correlated.

Exploring the data: Multicollinearity

Multicollinearity exists whenever 2+ predictors in a regression model are moderately or highly correlated.

- If two predictors X_1 and X_2 are highly correlated, it is hard to estimate the effect of changing X_1 while keeping X_2 constant.

Exploring the data: Multicollinearity

Multicollinearity exists whenever 2+ predictors in a regression model are moderately or highly correlated.

- If two predictors X_1 and X_2 are highly correlated, it is hard to estimate the effect of changing X_1 while keeping X_2 constant.
- This means we will have large standard errors, and large p-values, for X_1 and/or X_2.

Exploring the data: Multicollinearity

Multicollinearity exists whenever 2+ predictors in a regression model are moderately or highly correlated.

- If two predictors X_1 and X_2 are highly correlated, it is hard to estimate the effect of changing X_1 while keeping X_2 constant.
- This means we will have large standard errors, and large p-values, for X_1 and/or X_2.
- This does not mean there isn't a relationship between X_1 and Y, or X_2 and Y – it just means we can't pin down that relationship, because of the correlation!

Exploring the data: Multicollinearity

Multicollinearity exists whenever 2+ predictors in a regression model are moderately or highly correlated.

- If two predictors X_1 and X_2 are highly correlated, it is hard to estimate the effect of changing X_1 while keeping X_2 constant.
- This means we will have large standard errors, and large p-values, for X_1 and/or X_2.
- This does not mean there isn't a relationship between X_1 and Y, or X_2 and Y – it just means we can't pin down that relationship, because of the correlation!

Exploring the data: Multicollinearity

Multicollinearity exists whenever 2+ predictors in a regression model are moderately or highly correlated.

- If two predictors X_1 and X_2 are highly correlated, it is hard to estimate the effect of changing X_1 while keeping X_2 constant.
- This means we will have large standard errors, and large p-values, for X_1 and/or X_2.
- This does not mean there isn't a relationship between X_1 and Y, or X_2 and Y – it just means we can't pin down that relationship, because of the correlation!

Correlation between the response and the predictors is good, but correlation between the predictors is not!

Exploring the data: Multicollinearity

We want to avoid multicollinearity in our models!

Exploring the data: Multicollinearity

We want to avoid multicollinearity in our models!

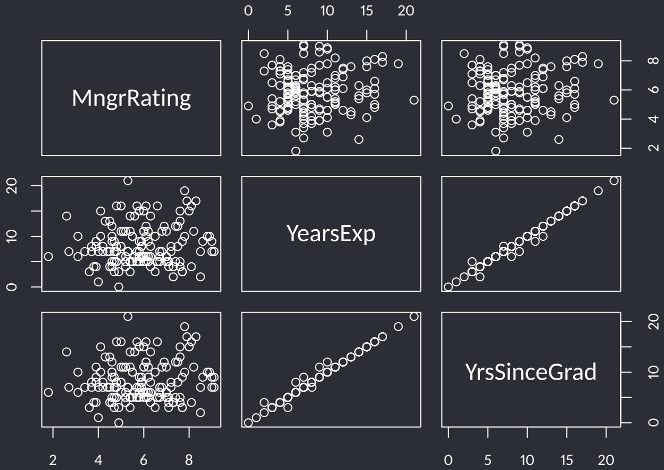
- Any conclusions based on the p-values, coefficients, and confidence intervals of the highly correlated variables will be unreliable.

Exploring the data: Multicollinearity

We want to avoid multicollinearity in our models!

- Any conclusions based on the p-values, coefficients, and confidence intervals of the highly correlated variables will be unreliable.
- These statistics will not be stable: adding new data or predictors to the model could drastically change them.

```r
pairs(~ MngrRating + YearsExp + YrsSinceGrad, data=mclean)
```



```r
model <- lm(Salary ~ MngrRating + YearsExp + YrsSinceGrad + Origin,
           data=mclean)
summary(model)
```

```
Call:
lm(formula = Salary ~ MngrRating + YearsExp + YrsSinceGrad +
    Origin, data = mclean)

Residuals:
     Min       1Q   Median       3Q      Max
-19.7766  -4.2842  -0.2906   3.3266  28.2773

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     54.1521     2.6071  20.771  < 2e-16 ***
MngrRating       4.5147     0.3997  11.296  < 2e-16 ***
YearsExp        -1.5262     1.3790  -1.107 0.270203
YrsSinceGrad     0.7692     1.3833   0.556 0.578976
OriginInternal  -4.7314     1.3878  -3.409 0.000838 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.838 on 149 degrees of freedom
Multiple R-squared:  0.6065,Adjusted R-squared:  0.596
F-statistic: 57.42 on 4 and 149 DF,  p-value: < 2.2e-16
```

Exploring the data: Multicollinearity

One way to see if two variables are collinear is to check the correlation between the two:

```
cor(mclean$YearsExp, mclean$YrsSinceGrad)

[1] 0.9947616
```

Exploring the data: Multicollinearity

One way to see if two variables are collinear is to check the correlation between the two:

```
cor(mclean$YearsExp, mclean$YrsSinceGrad)

[1] 0.9947616
```

Any correlation ≥ 0.95 is definitely a problem, but smaller correlations could be problematic too.

Exploring the data: Multicollinearity

A better way to check multicollinearity is using Variance Inflation Factors (VIF).

- The VIF is

$$\text{VIF}(\beta_j) = \frac{1}{1 - R_j^2},$$

  where R_j^2 is the R^2 in a regression predicting X variable j from the other X variables.

Exploring the data: Multicollinearity

A better way to check multicollinearity is using Variance Inflation Factors (VIF).

- The VIF is
$$\text{VIF}(\beta_j) = \frac{1}{1 - R_j^2},$$
where R_j^2 is the R^2 in a regression predicting X variable j from the other X variables.
- $\text{VIF}(\beta_j) = 0$ when $R_j^2 = 0$; i.e., the jth predictor variable is completely independent from the others.

Exploring the data: Multicollinearity

A better way to check multicollinearity is using Variance Inflation Factors (VIF).

- The VIF is

$$\mathrm{VIF}(\beta_j) = \frac{1}{1 - R_j^2},$$

  where R_j^2 is the R^2 in a regression predicting X variable j from the other X variables.
- $\mathrm{VIF}(\beta_j) = 0$ when $R_j^2 = 0$; i.e., the jth predictor variable is completely independent from the others.
- $\mathrm{VIF}(\beta_j)$ increases as R_j^2 does, and is ∞ when there is perfect multicollinearity; i.e., when X_j is perfectly predictable from the other X variables.

Exploring the data: Multicollinearity

```
library(car)
vif(model)

  MngrRating     YearsExp YrsSinceGrad       Origin
    1.136002    95.954255    97.011260     1.540448
```

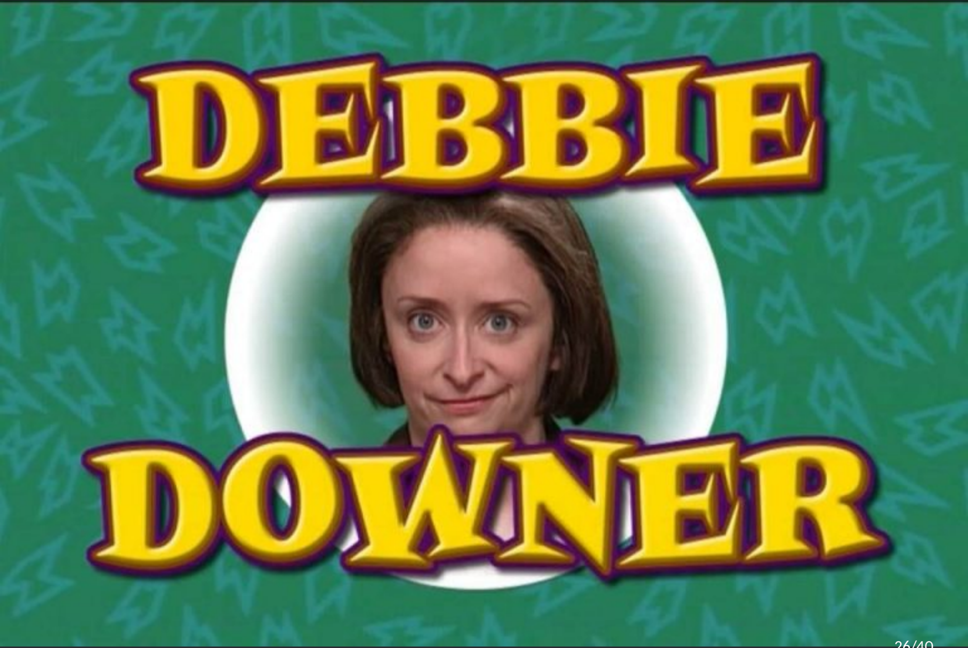
Predictors with VIF > 5 indicate multicollinearity.

Exploring the data: Multicollinearity

```
library(car)
vif(model)

  MngrRating     YearsExp YrsSinceGrad       Origin
    1.136002    95.954255    97.011260     1.540448
```

Predictors with VIF > 5 indicate multicollinearity.

Remember: Multicollinearity could exist between more than two predictors (this is why there are only $n - 1$ dummy variables for a categorical variable with n values).


DEBBIE
DOWNER

Dealing with multicollinearity

There are two general strategies for dealing with multicollinearity:

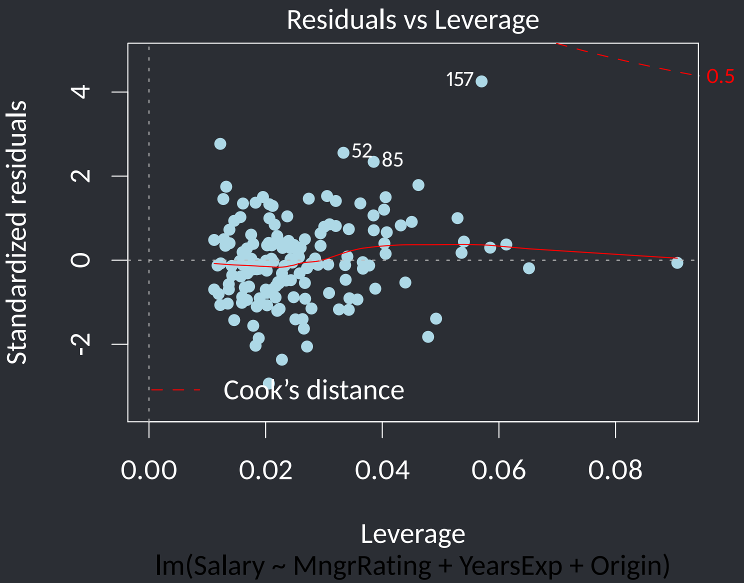
- Drop a variable with a high VIF factor. (Just like we drop one of the dummy variables when putting a categorical variable in the model!)
- Combine the variables that correlate into a composite variable.

```r
model2 <- lm(Salary ~ MngrRating + YearsExp + Origin, data=mclean)
summary(model2)
```

```
Call:
lm(formula = Salary ~ MngrRating + YearsExp + Origin, data = mclean)

Residuals:
     Min       1Q   Median       3Q      Max 
-19.8115  -4.3474  -0.3964   3.3358  28.1801 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     54.1080     2.5999  20.812  < 2e-16 ***
MngrRating       4.5309     0.3977  11.394  < 2e-16 ***
YearsExp        -0.7651     0.1687  -4.534 1.18e-05 ***
OriginInternal  -4.6467     1.3762  -3.376 0.000935 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.823 on 150 degrees of freedom
Multiple R-squared:  0.6057,Adjusted R-squared:  0.5978 
F-statistic: 76.82 on 3 and 150 DF,  p-value: < 2.2e-16
```

1. Outliers
2. Missing data
3. Multicollinearity
4. **Highly influential points**
5. Handling nonlinearity

Finding highly influential points



Outliers among the residuals

Let's look at row 157:

```
manager[157,]
```

```
    Salary MngrRating YearsExp YrsSinceGrad   Origin
157     95          4        1            1 Internal
```

Someone with only 1 year of experience and a poor rating is hired as manager at $95K!

Outliers among the residuals

Let's look at row 157:

```
manager[157,]

    Salary MngrRating YearsExp YrsSinceGrad   Origin
157     95          4        1            1 Internal
```

Someone with only 1 year of experience and a poor rating is hired as manager at $95K!

If you decide that this is an anomaly (e.g. the CEO's son was promoted!) that you don't want to include in your analysis, omit that row and report on it separately in your conclusions.

Influential cases

- The Residuals vs Leverage plot tells about potentially influential.

Influential cases

- The Residuals vs Leverage plot tells about potentially influential.
- A **high-leverage case** is one that has an unusual combination of predictor values.

Influential cases

- The Residuals vs Leverage plot tells about potentially influential.
- A **high-leverage case** is one that has an unusual combination of predictor values.
- An **influential case** is a high-leverage case that also has a high residual: it could change your β values significantly when excluded from your analysis, i.e., it does not follow the overall trend.

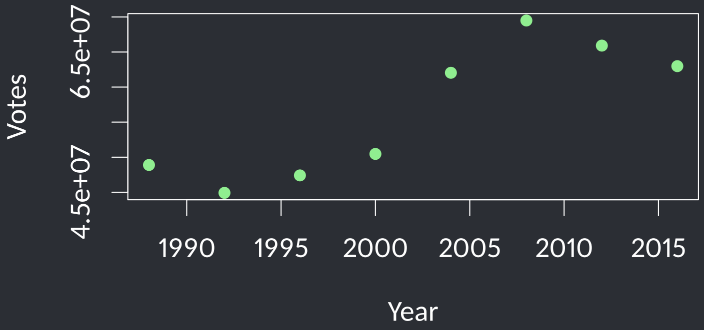
Influential cases

- The Residuals vs Leverage plot tells about potentially influential.
- A **high-leverage case** is one that has an unusual combination of predictor values.
- An **influential case** is a high-leverage case that also has a high residual: it could change your β values significantly when excluded from your analysis, i.e., it does not follow the overall trend.
- Look for the cases on the upper/lower right corners (beyond the dashed curves).

1. Outliers
2. Missing data
3. Multicollinearity
4. Highly influential points
5. Handling nonlinearity

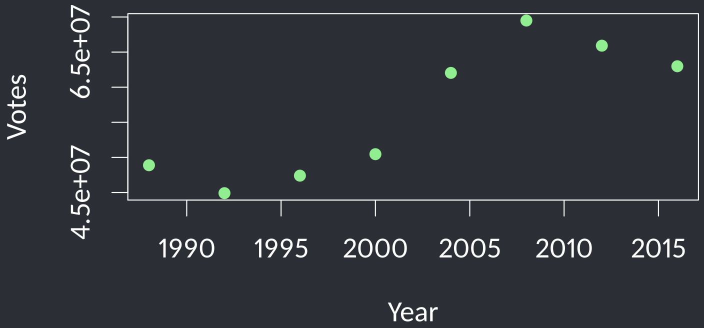
Let’s look at the total number of votes the winning candidate for U.S. President has won since 1988:

```
plot(Votes ~ Year, data=elections, pch=16, col="lightgreen"
```



Let's look at the total number of votes the winning candidate for U.S. President has won since 1988:

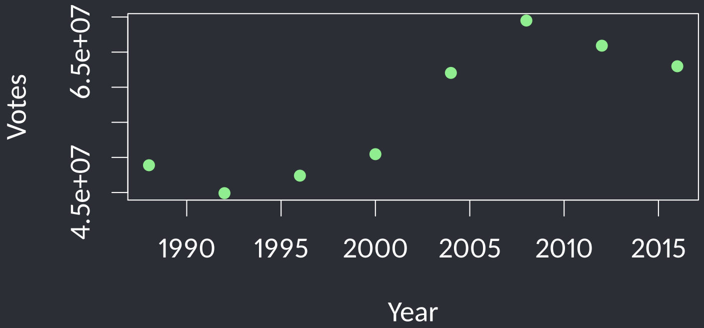
```
plot(Votes ~ Year, data=elections, pch=16, col="lightgreen"
```



Is a line a good fit for this data?

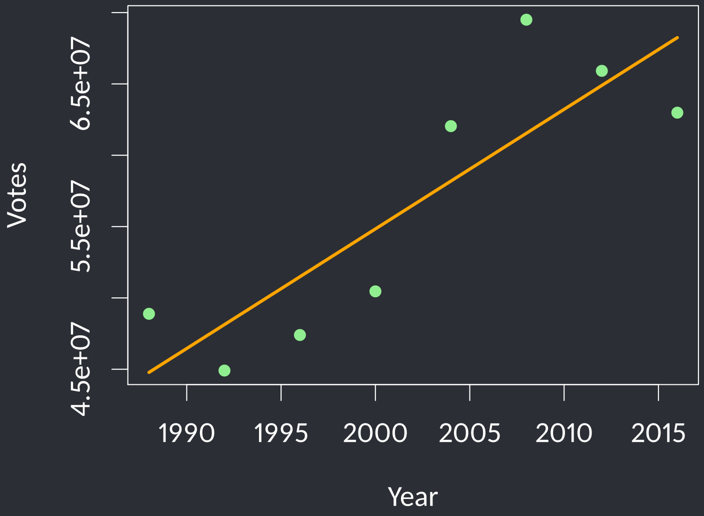
Let's look at the total number of votes the winning candidate for U.S. President has won since 1988:

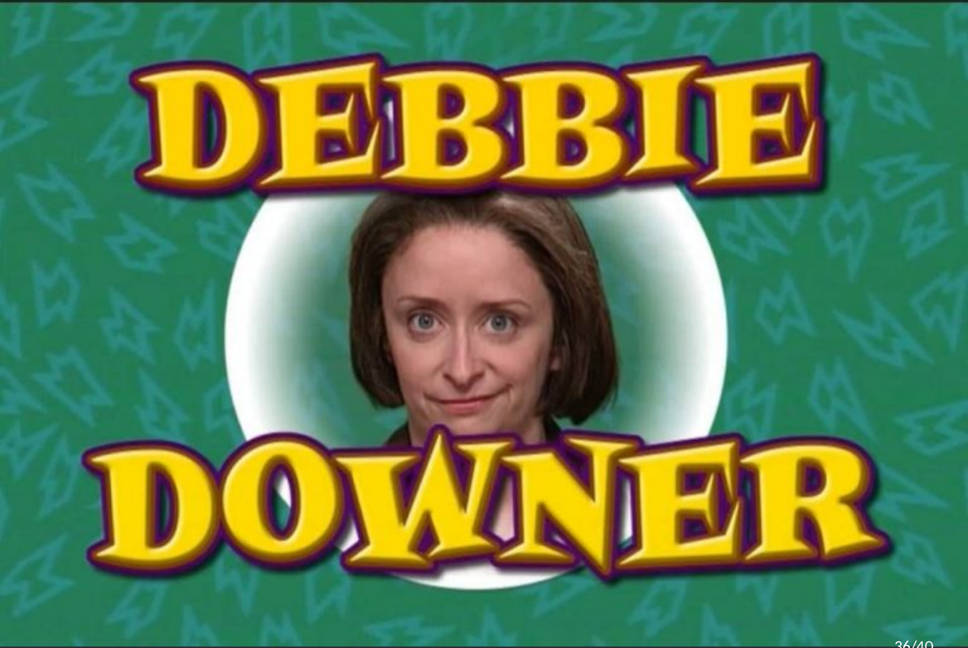
```
plot(Votes ~ Year, data=elections, pch=16, col="lightgreen"
```



Is a line a good fit for this data? Is there any kind of transformation of either X or Y that can fix this nonlinearity?

```r
elections$Time <- elections$Year - 1988
model1 <- lm(Votes ~ Time, data=elections)
```




DEBBIE
DOWNER

Let's try to fit a polynomial!

- Think of a transformation of X or Y as fitting a curve to the data; for example, $X \to \log X$ fits a logarithmic curve to the data.

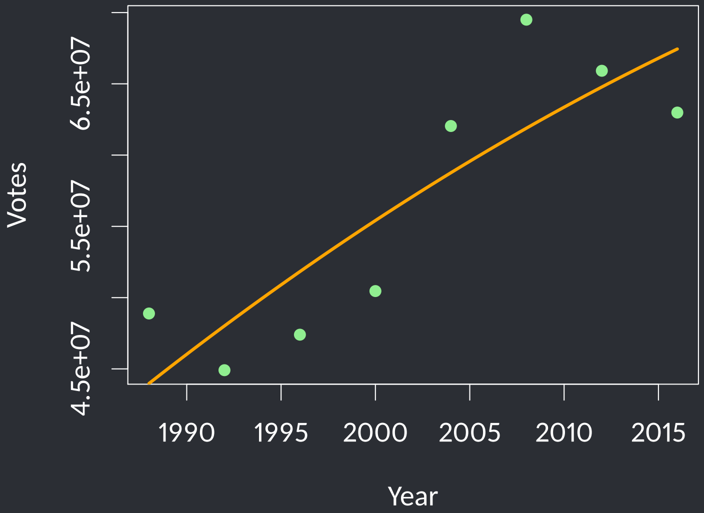
Let's try to fit a polynomial!

- Think of a transformation of X or Y as fitting a curve to the data; for example, $X \to \log X$ fits a logarithmic curve to the data.
- A polynomial is a function of the form

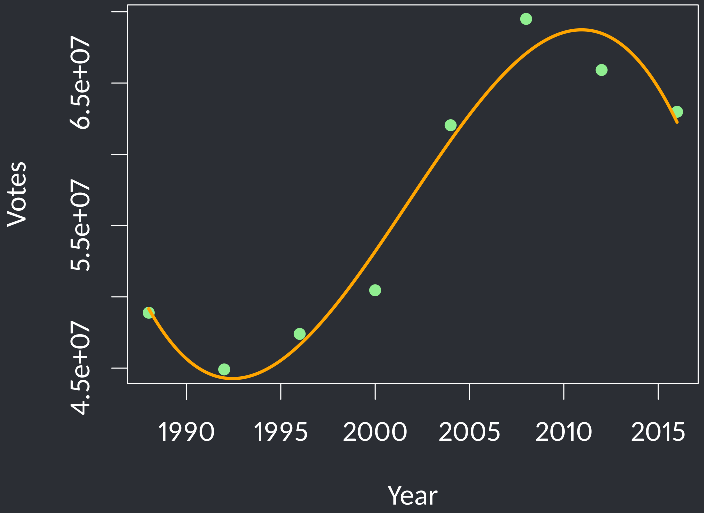
$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n,$$

for some n. We can fit a polynomial curve to our data by just adding the higher order X^k terms as predictor variables to our regression model!

```r
elections$TimeSquared <- elections$Time^2
model2 <- lm(Votes ~ Time + TimeSquared, data=elections)
```



```r
elections$TimeCubed <- elections$Time^3
model3 <- lm(Votes ~ Time + TimeSquared +
                              TimeCubed, data=elections)
```



Nonlinearity

- Just like model selection with any other variable: use the statistical significance of the highest order term, and changes in R^2, to determine how many powers you should add.
- Just like interactions, you should have a good reason to try a polynomial model.
- If you include a power X^k, you should also include $X, X^2, \ldots, X^{k-1}$, even if they are not statistically significant.
- Be particularly careful with extrapolation when using a polynomial model!