Class 4: Linear Regression

BUS 696

Prof. Jonathan Hersh

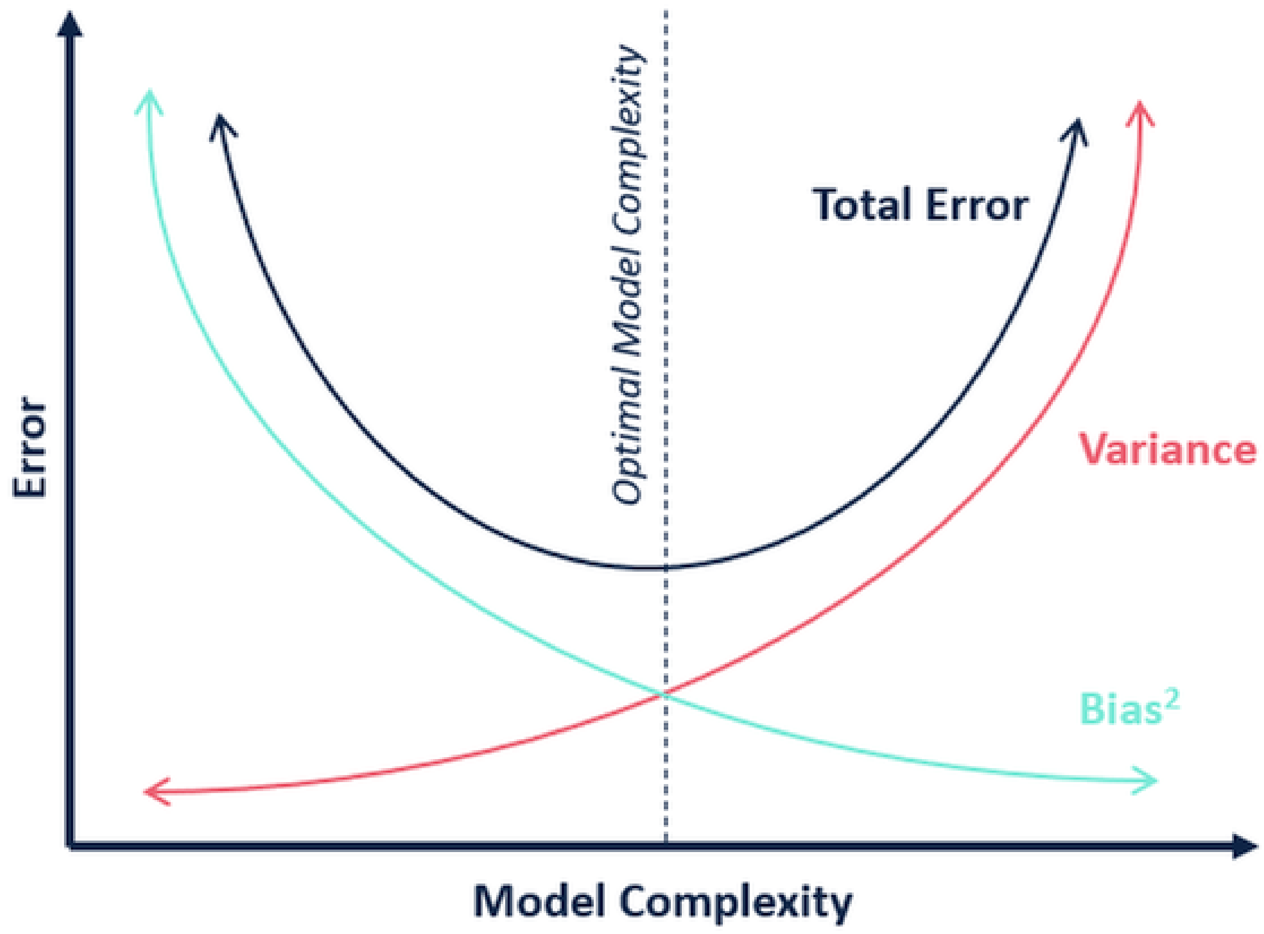
Class 4: Announcements

1. Problem Set 1 Posted, Due Sept 25
    - Any Qs?
2. Office Hours
    1. TA: Wed: 12-1, Th 5-6
    2. Instructor: M: 11-12, W 5-6
3. Post October 5th: Hybrid Online/In person?

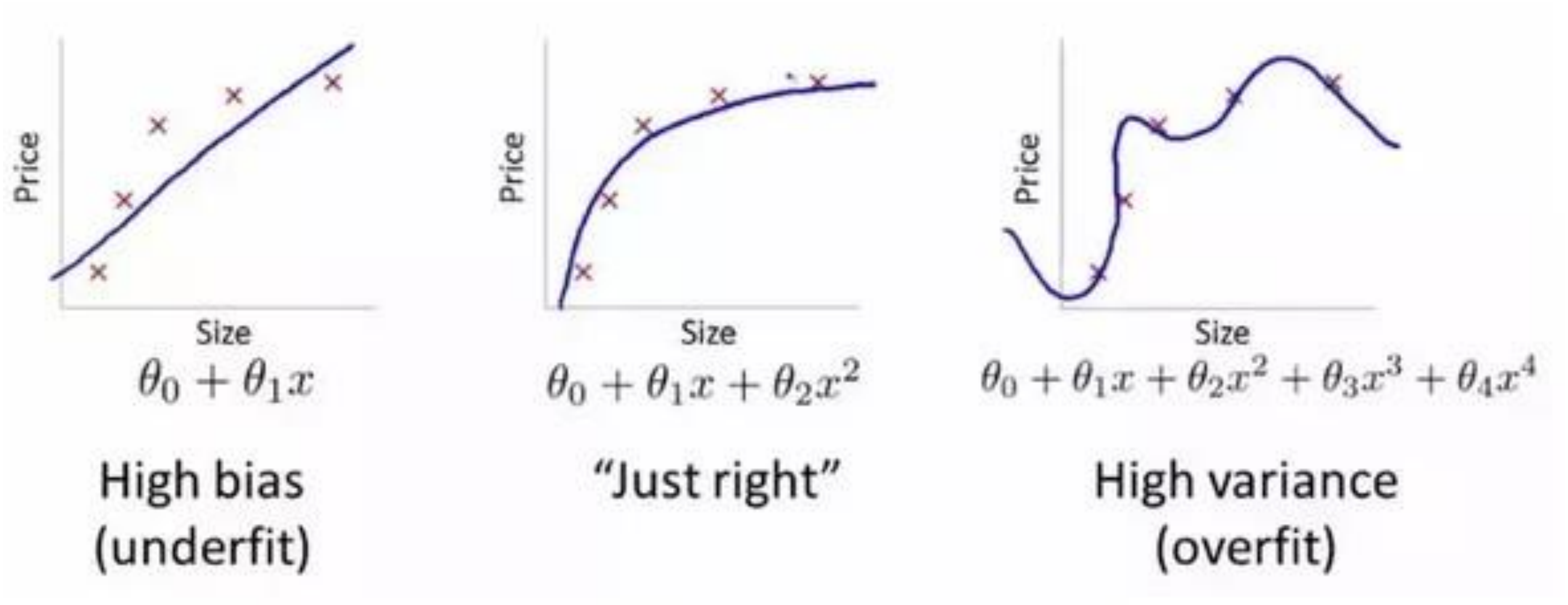
Class 4: Outline

1. Last Class Review:
   - Bias, Variance, Overfit, Underfit, Mean Squared Error
2. Linear Regression Review
3. Estimating Linear Models in R
4. Interpreting Linear Model Coefficients
5. Regression Lab 1
6. Inference/Hypothesis Testing in Linear Models
7. Discrete/Qualitative Independent Variables
8. Model Evaluation
   - Predicted/True Plots, RMSE, and R-Squared
9. Regression Lab 2

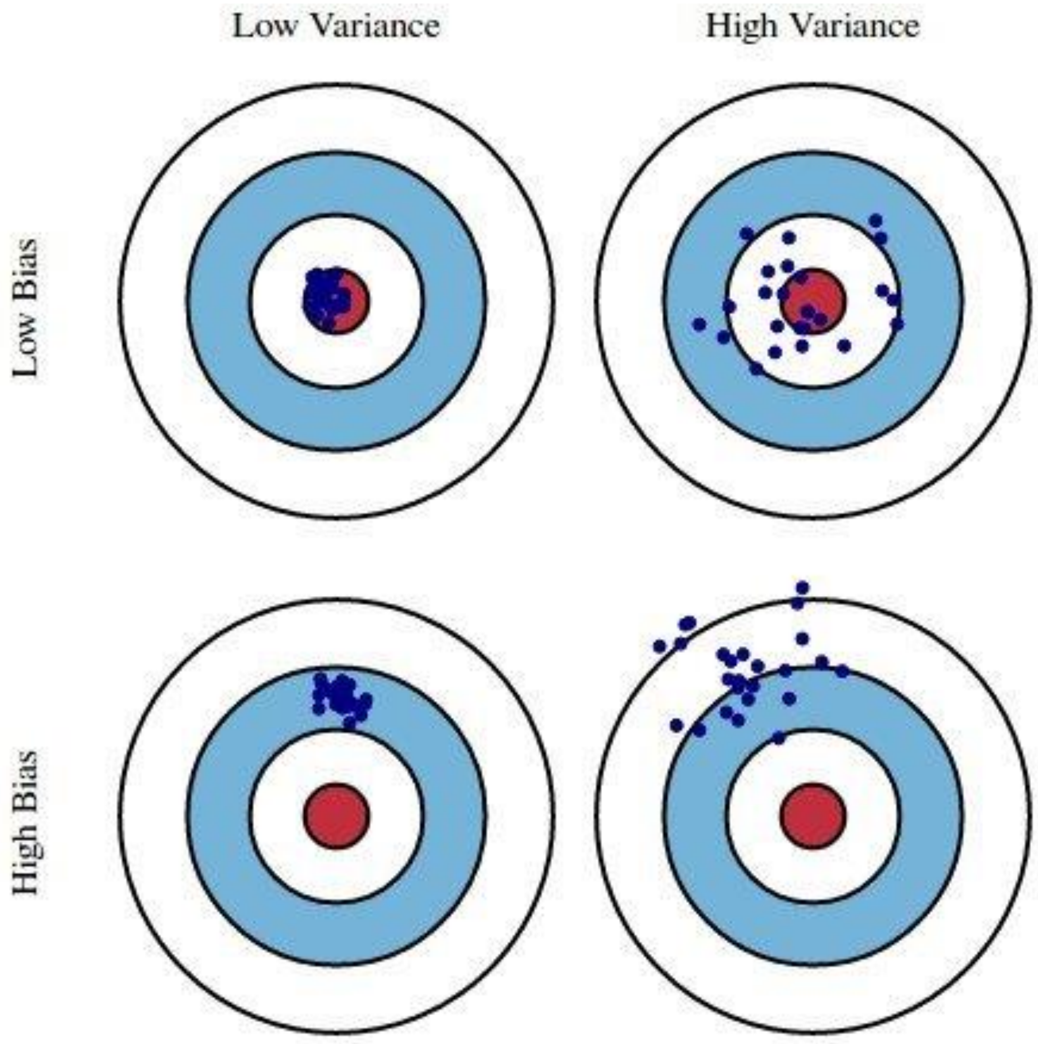
Key: Finding Optimal Model Complexity



Optimal Model Complexity: Neither Underfit Nor Overfit



Bias-Variance Tradeoff



How to Judge Difference Between True Data And Model? Loss Function aka Distance Metric

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2$$

$\sum$ means we add up anything with i, starting at $i = 1$ to $i = N$ (all obs)

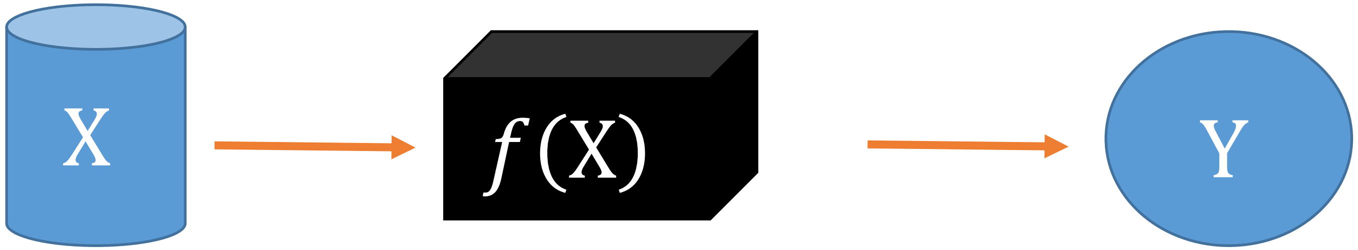
Note:
1. Large differences penalized more than small distances
2. Positive vs negative errors equally penalized

Mean Squared Error (MSE)

y_i	$\hat{y}_i$	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$
5	5	0	0
5	7	-2	$(-2)^2$=4
9	8	1	1^2=1
10	1	9	9^2=81
13	13	0	0

Recipes for learning $f(X)$: Ordinary Linear Models

$$Y = f(X) + \epsilon$$



Ordinary Linear Models

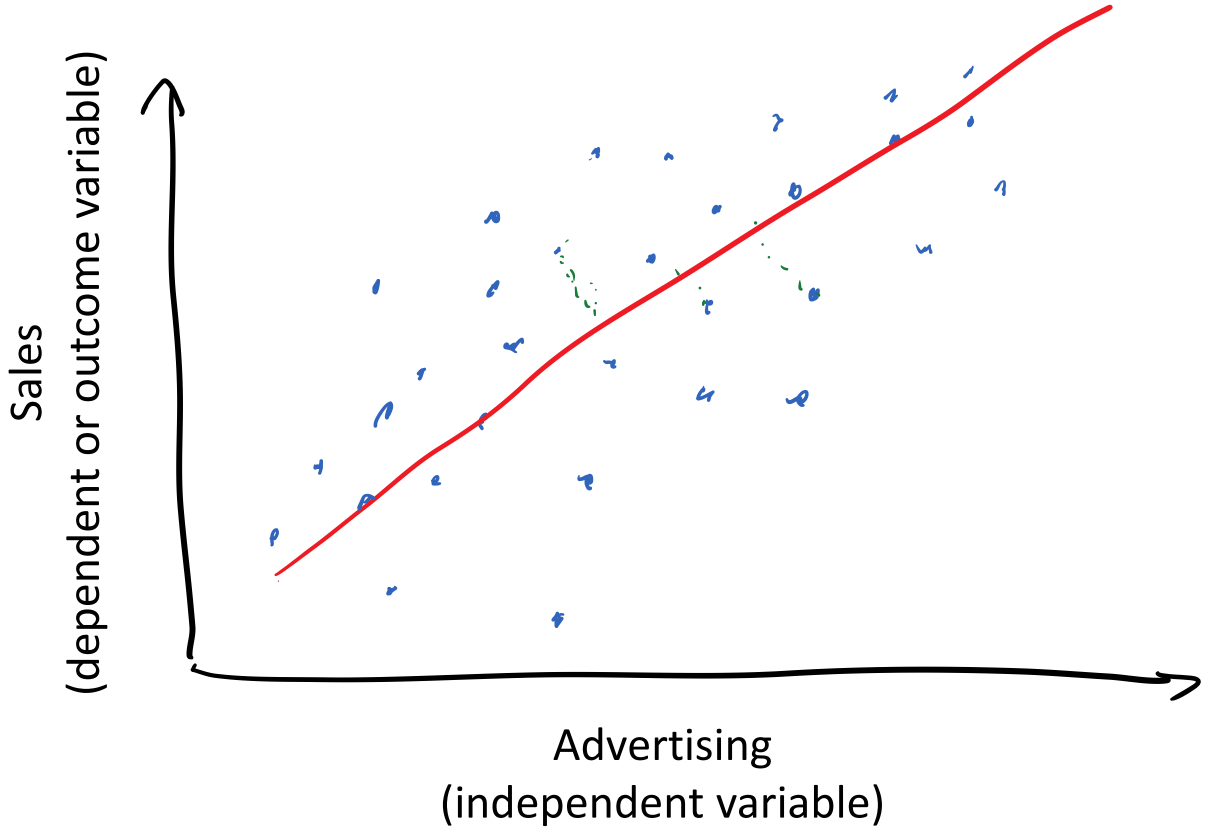
$$f(X) = \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_k \cdot x_k$$

OLS: Only allows linear combinations of Xs

Class 7: Outline

1. Review Bias, Variance, Overfit, Underfit
2. **Linear Regression Review**
3. Estimating Linear Models in R
4. Interpreting Model Coefficients
5. Regression Lab

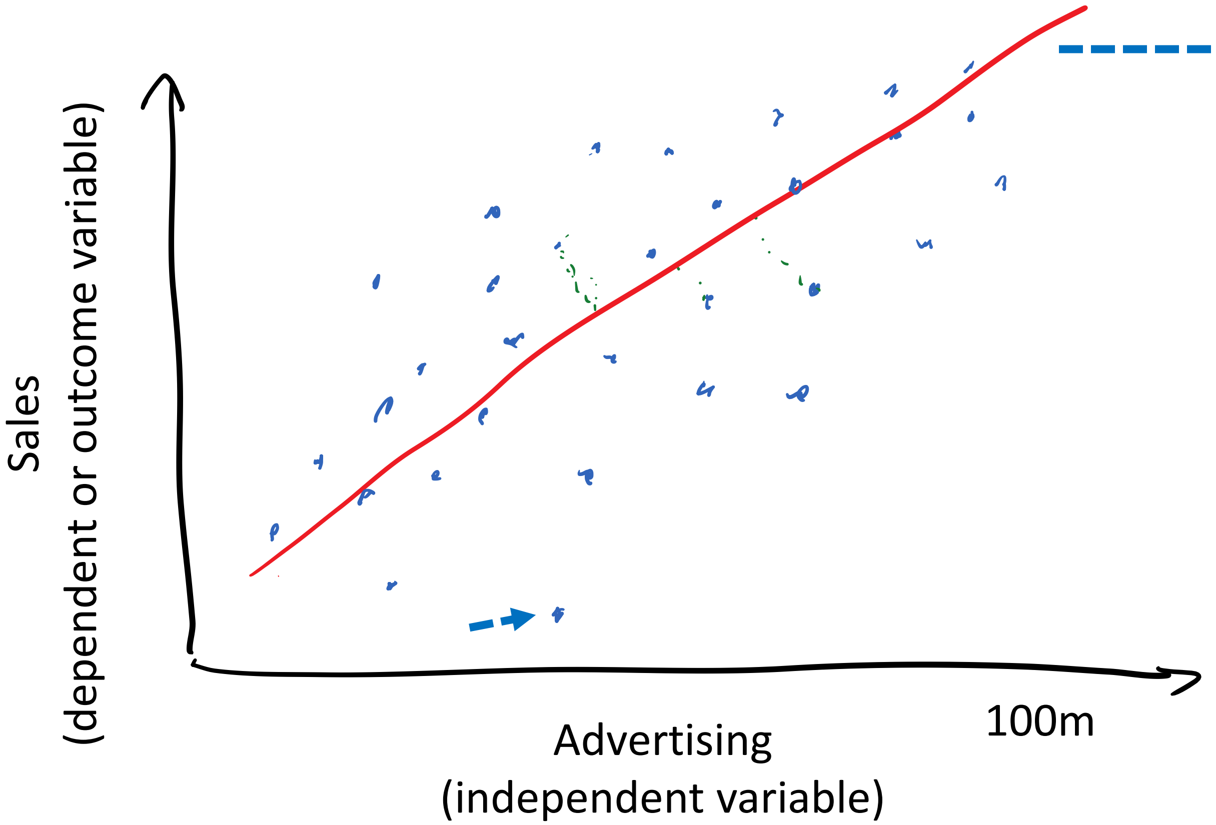
What is Linear Regression?



Regression: statistical process of estimating relationship between an <u>outcome</u> and and one or more <u>predictors</u> or <u>independent variables</u>

Linear Regression: restricting relationship between predictors and outcome to be linear

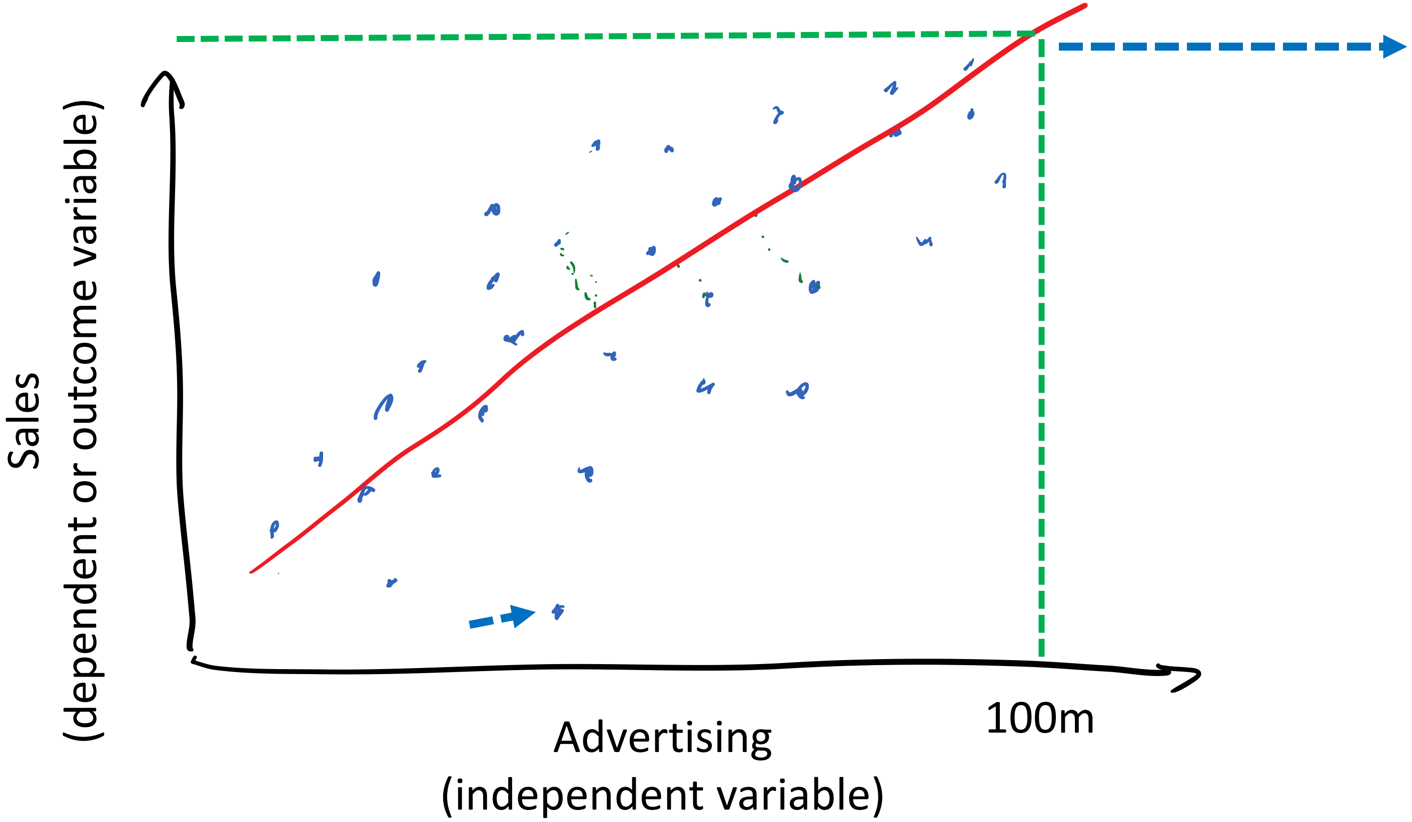
Linear Regression Equation



$$y_i = \beta_0 + \beta_1 x_{Advertising} + \epsilon_i$$

Red line “explains” the data the best.

Predictions from Linear Regression



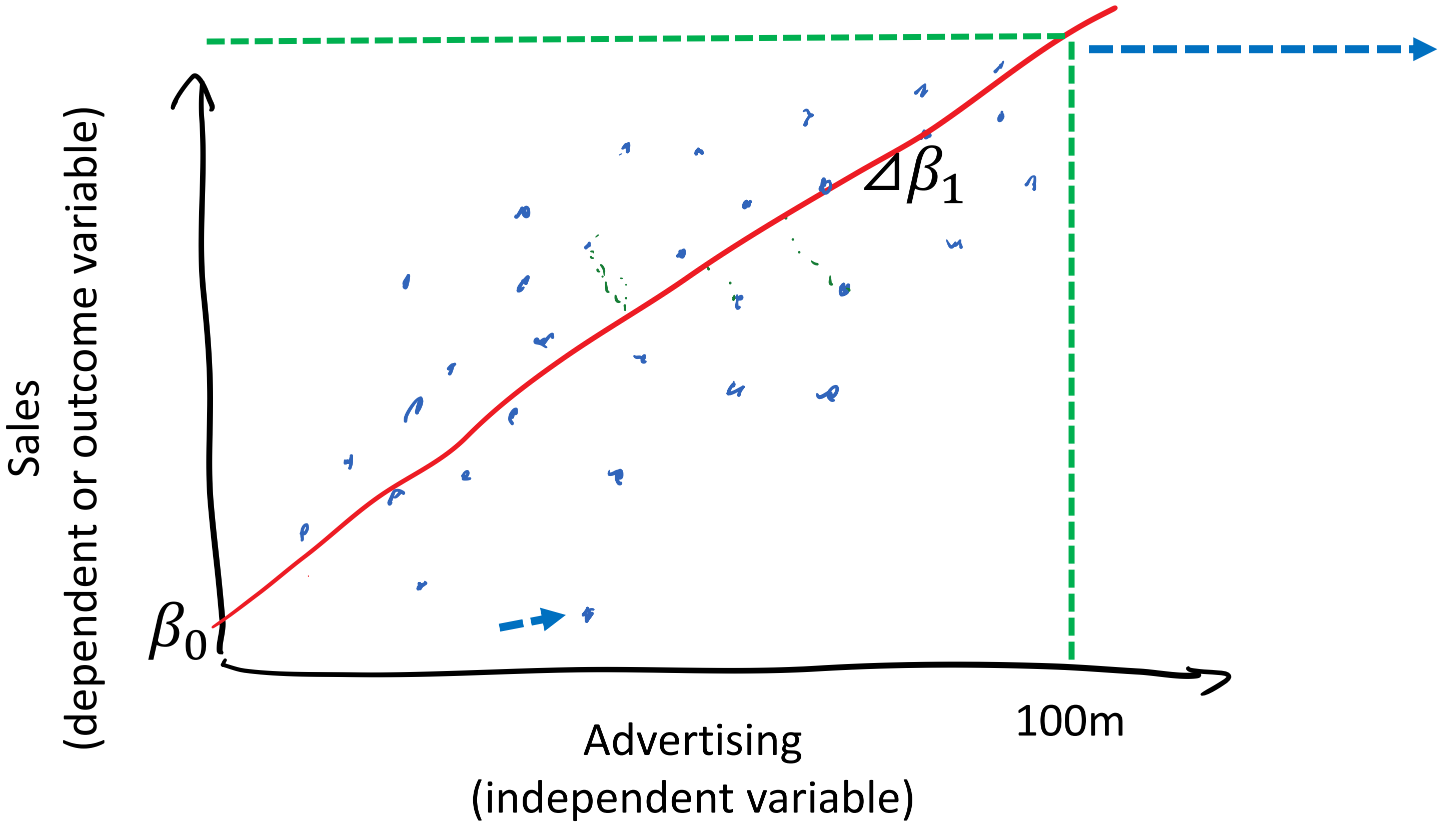
$$y_i = \beta_0 + \beta_1 x_{Advertising} + \epsilon_i$$

Suppose we spend 100m on advertising?
What's our expected sales?

$$? = \widehat{\beta_0} + \widehat{\beta_1} 100m$$

"Hat", e.g. $\widehat{\beta_0}$, means we've estimated this relationship from data.

Predictions from Linear Regression



$$y_i = \beta_0 + \beta_1 x_{Advertising} + \epsilon_i$$

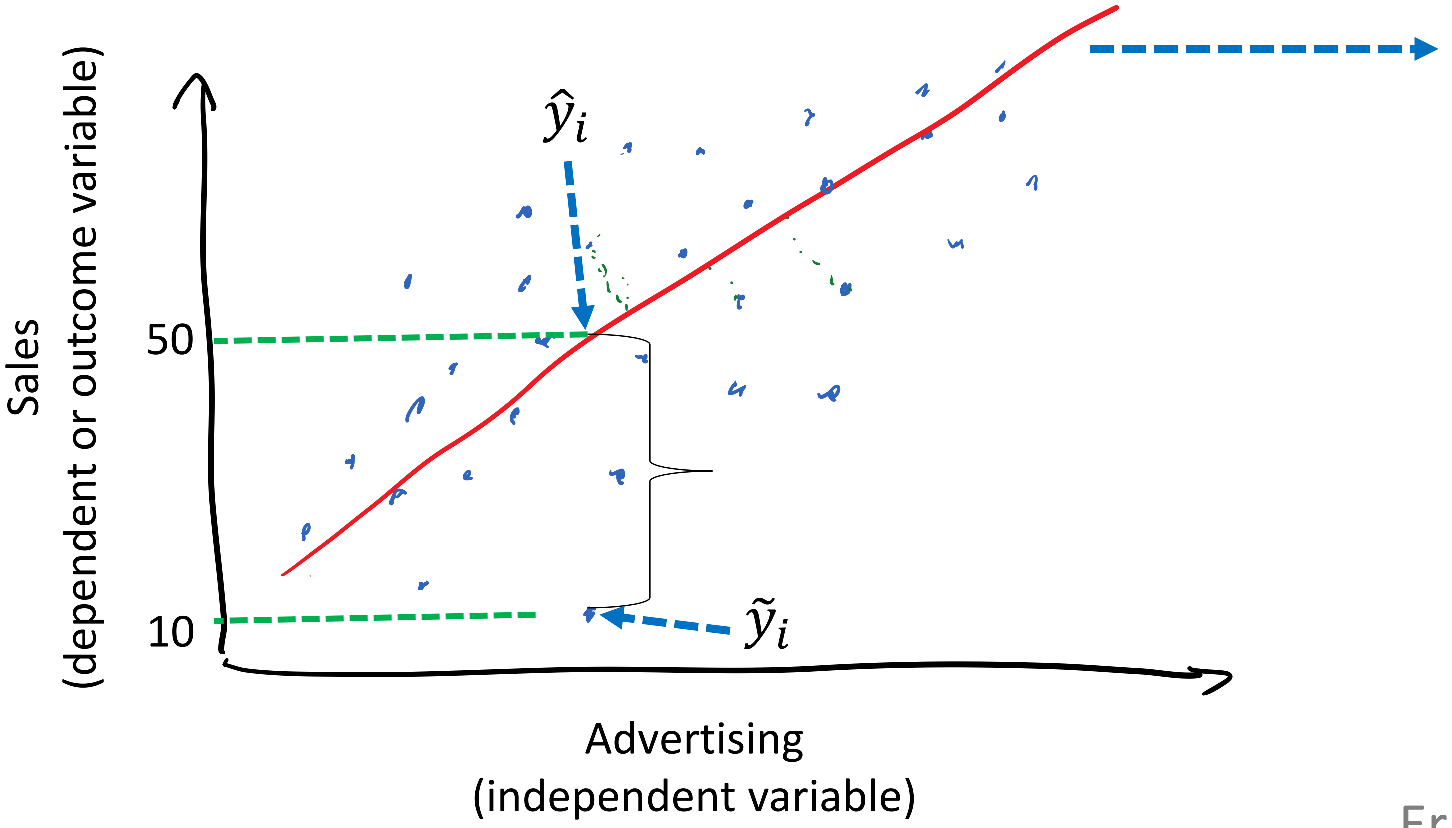
Suppose we spend 100m on advertising?
What's our expected sales?

$$? = \widehat{\beta_0} \;\; + \widehat{\beta_1} 100m$$

$$? = 10 \;\; + 1 * 100$$
$$110 = 10 \;\; + 1 * 100$$

"Hat", e.g. $\widehat{\beta_0}$, means we've estimated this relationship from data.

Measuring Errors



$$y_i = \beta_0 + \beta_1 x_{Advertising} + \epsilon_i$$

Errors: $\epsilon_i = y_i - \hat{y}_i$

Error: $\hat{\epsilon}_i = 10 - 50 = -40$

Errors are the difference between what we predict ($\hat{y}_i$) and the actual values (y_i).

Class 4: Outline

1. Last Class Review:
   - Bias, Variance, Overfit, Underfit, Mean Squared Error
2. Linear Regression Review
3. Estimating Linear Models in R
4. Interpreting Linear Model Coefficients
5. Regression Lab 1
6. Inference/Hypothesis Testing in Linear Models
7. Discrete/Qualitative Independent Variables
8. Model Evaluation
   - Predicted/True Plots, RMSE, and R-Squared
9. Regression Lab 2

Model Formulas in R

- Formulas in R start with the dependent variable on the left hand side (LHS)
- Followed by ”~” tilde
- Then all dependent variables separated by plus signs

```
>
>
>
> data(mpg)
> hwy ~ year + displ + cyl
hwy ~ year + displ + cyl
>
```

- **The above translates to a regression equation of:**
- $\boldsymbol{hwy = \beta_0 + \beta_1 \cdot year + \beta_2 \cdot displ + \beta_3 \cdot cyl}$

Estimating Linear Models Using lm()

- Estimate a linear model using the 'lm()' function in R
- We must pass the dataset on which to estimate our model
- Then we store the regression model as 'mod1' (or whatever name you like
- Summary() outputs a summary of the estimated model

```r
# estimate a linear model with displacement, and
# cycl on the RHS, and hwy as the
# development variable (LHS)
# Use the 'mpg' dataframe to estimate the model
# and store the regression equation as 'mod1'
mod1 <- lm(hwy ~ displ + cyl,
           data = mpg)

# print out a summary of the linear model
summary(mod1)

# or just view the whole "list" object of
# the model results
str(mod1)
```

Viewing Regression Output Using “Summary”

```
> summary(mod1)

Call:
lm(formula = hwy ~ displ + cyl, data = mpg)

Residuals:
    Min      1Q  Median      3Q     Max
-7.5098 -2.1953 -0.2049  1.9023 14.9223

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept)  38.2162     1.0481  36.461 < 0.0000000000000002 ***
displ        -1.9599     0.5194  -3.773             0.000205 ***
cyl          -1.3537     0.4164  -3.251             0.001323 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.759 on 231 degrees of freedom
Multiple R-squared:  0.6049,    Adjusted R-squared:  0.6014
F-statistic: 176.8 on 2 and 231 DF,  p-value: < 0.00000000000000022
```

Coefficient standard errors

Estimated Coefficients or “betas”

Independent variables

Coefficient T-Statistic

P-values for coefficients

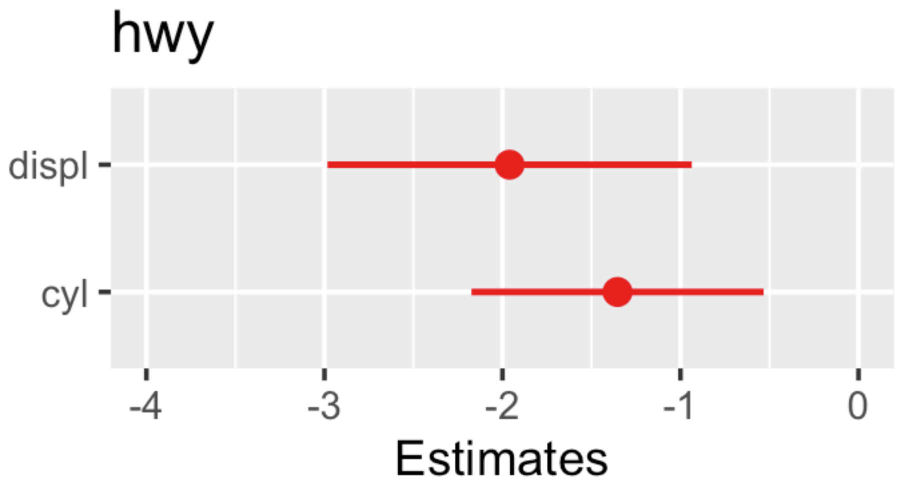
R^2, or “coefficient of determination” (model fit)

Making “Pretty” Version of Regression Output Table

```r
# install.packages('sjPlot')
library('sjPlot')
# output a prettier table of results
# looks very nice in RMarkdown!
tab_model(mod1)

# output a plot of regression coefficients
plot_model(mod1)

# output a table of nice coefficients
tidy(mod1)
```

	hwy		
Predictors	*Estimates*	*CI*	*p*
(Intercept)	38.22	36.15 – 40.28	**<0.001**
displ	-1.96	-2.98 – -0.94	**<0.001**
cyl	-1.35	-2.17 – -0.53	**0.001**
Observations	234		
R^2 / R^2 adjusted	0.605 / 0.601		

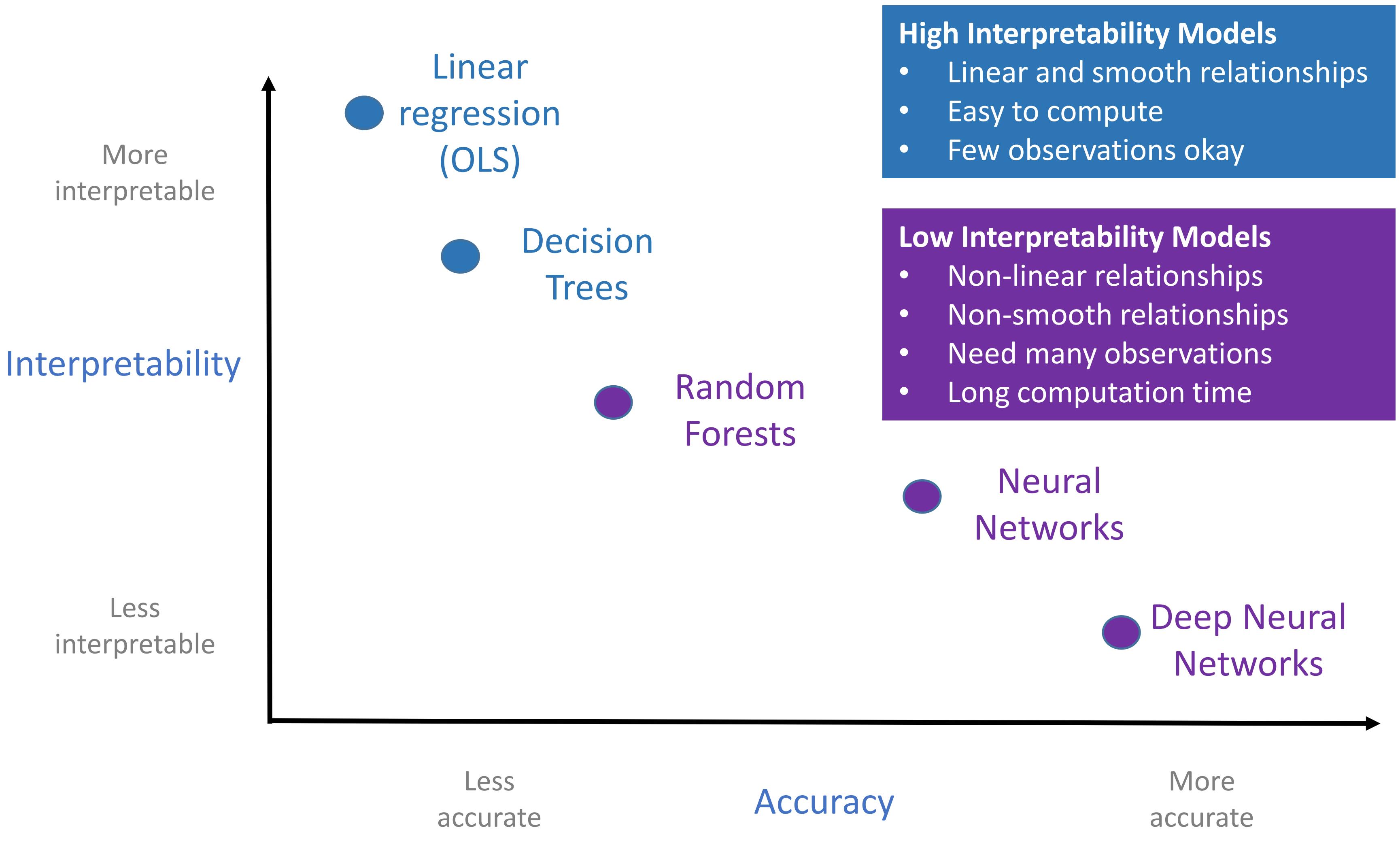
```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    38.2      1.05       36.5 8.57e-98
2 displ          -1.96     0.519      -3.77 2.05e- 4
3 cyl            -1.35     0.416      -3.25 1.32e- 3
>
```



Class 4: Outline

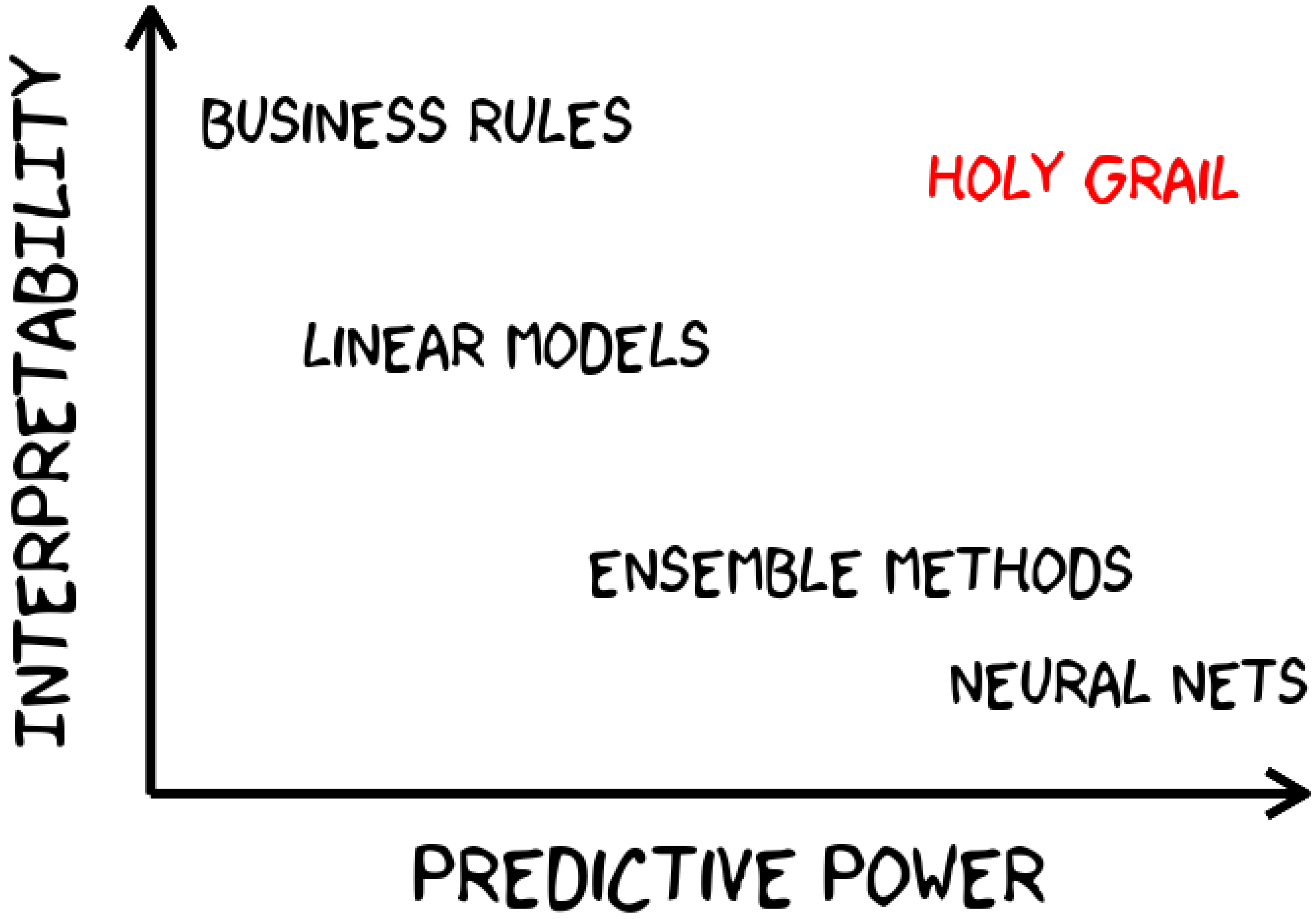
1. Last Class Review:
   - Bias, Variance, Overfit, Underfit, Mean Squared Error
2. Linear Regression Review
3. Estimating Linear Models in R
4. Interpreting Linear Model Coefficients
5. Regression Lab 1
6. Inference/Hypothesis Testing in Linear Models
7. Discrete/Qualitative Independent Variables
8. Model Evaluation
   - Predicted/True Plots, RMSE, and R-Squared
9. Regression Lab 2

What Is Model Interpretability?



- **Model interpretability:**
  - “the degree to which a human can understand the cause of a decision” (Miller, 2017)
- The higher the interpretability, the easier it is for someone to comprehend why a decision has been made

Of Course We Care About Both!



Why Do We Care About Model Interpretability?

1. Strengthen Trust and Transparency

- People trust things they can understand, and don’t trust things they don’t (5G)

2. Explain decisions

- An interpretable model allows humans to understand the proposed decision, and diagnose and analyzed the solution

3. Regulatory Requirements

- Certain regulatory schemes (GDPR, Anti-Discrimination) require transparency.

4. Improve the models

- Interpretability ensures the model is right or wrong for the right reasons. Interpretability offers new feature engineering and helps debugging.

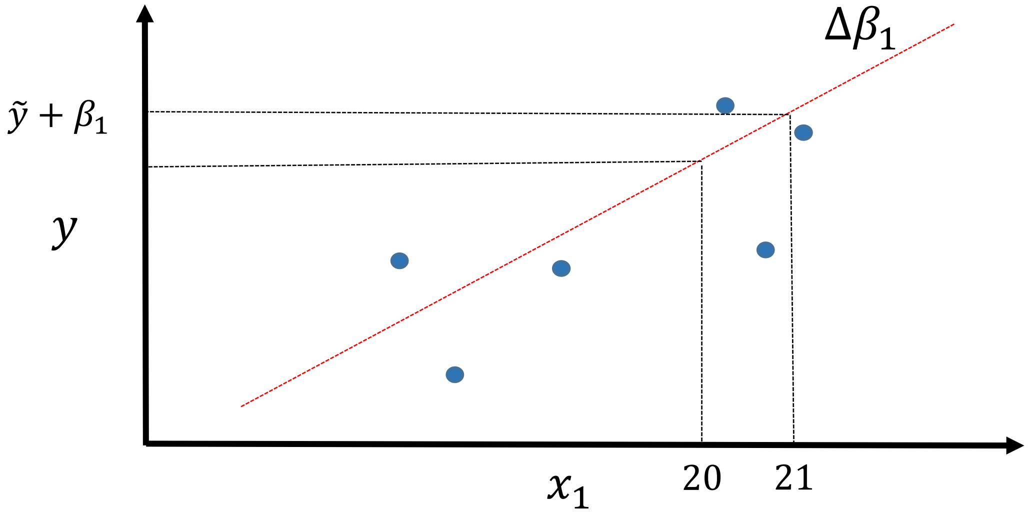
Interpreting Linear Model Coefficients

- $\boldsymbol{\beta_1}$ mathematically explains how y changes when we increase x_1 by one unit
- Suppose we change x_1 by one unit of x_1. By how much does y change?
- Well, it changes by exactly $\boldsymbol{\beta_1}$

$$y = \beta_0 + \boldsymbol{\beta_1} \cdot x_1 \ + \cdots + \beta_k \cdot x_k$$

$$? = \beta_0 + \boldsymbol{\beta_1} \cdot (x_1 + 1) \ + \cdots + \beta_k \cdot x_k$$

$$\tilde{y} + \beta_1 = \beta_0 + \boldsymbol{\beta_1} \cdot (x_1 + 1) \ + \cdots + \beta_k \cdot x_k$$



Interpreting Linear Coefficients In Words

- **Communicating effect of coefficient**

  Increasing displacement by **one liter** (communicate units!) **decreases highway mile per gallon** (y variable) by **1.96 miles per gallon** **holding all else (cyl) fixed**

  - **X-variable**
  - **X-variable units**
  - **Direction (pos/neg)**
  - **Y-variable (outcome)**
  - **Estimated coefficient (magnitude)**
  - **Y-units**

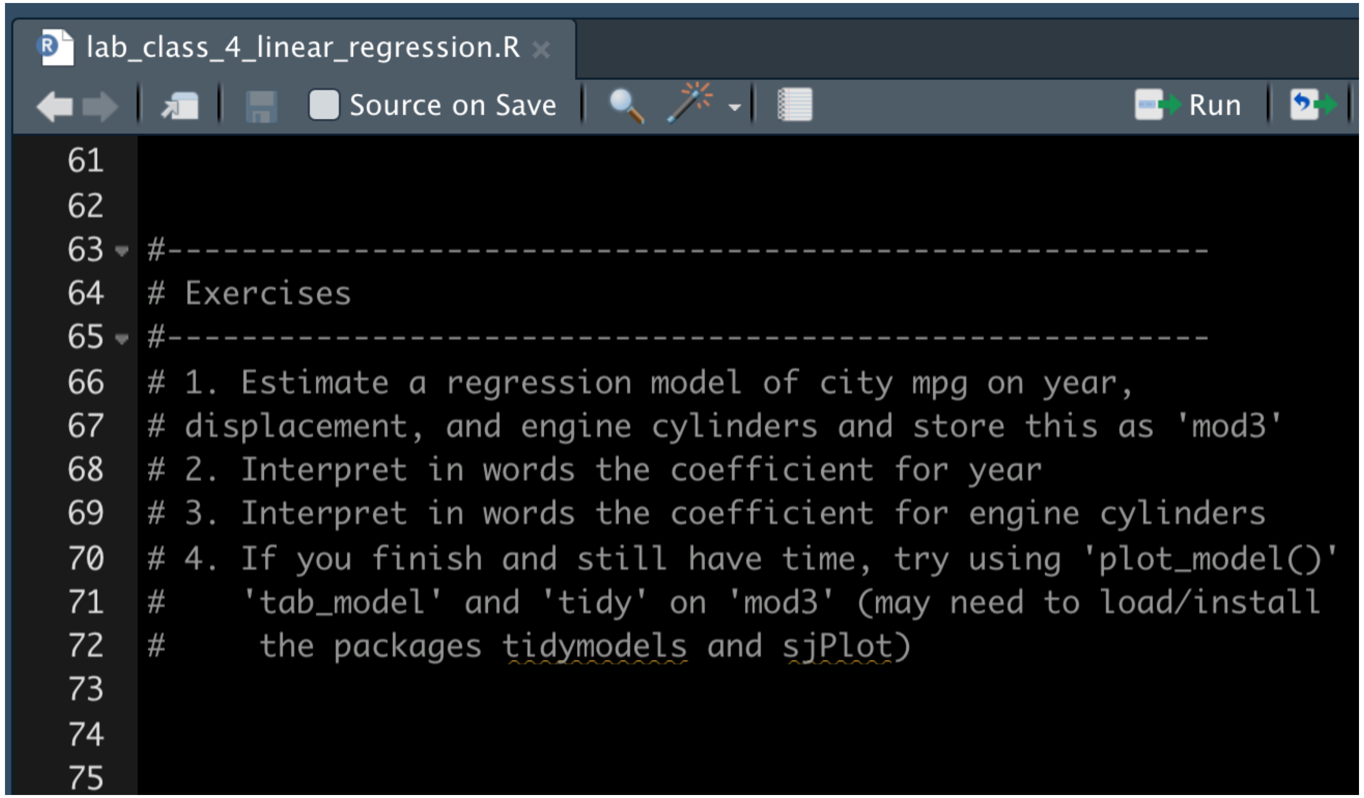
```
> summary(mod1)

Call:
lm(formula = hwy ~ displ + cyl, data = mpg)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.5098 -2.1953 -0.2049  1.9023 14.9223 

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)    
(Intercept)  38.2162     1.0481  36.461 < 0.0000000000000002 ***
displ        -1.9599     0.5194  -3.773             0.000205 ***
cyl          -1.3537     0.4164  -3.251             0.001323 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.759 on 231 degrees of freedom
Multiple R-squared:  0.6049,	Adjusted R-squared:  0.6014 
F-statistic: 176.8 on 2 and 231 DF,  p-value: < 0.00000000000000022
```

DO NOT JUST SAY WHEN X GOES UP Y GOES UP
THIS IS OBVIOUS AND YOU WILL GET FIRED

Class 4: Outline

1. Last Class Review:
   - Bias, Variance, Overfit, Underfit, Mean Squared Error
2. Linear Regression Review
3. Estimating Linear Models in R
4. Interpreting Linear Model Coefficients
5. **Regression Lab 1**
6. Inference/Hypothesis Testing in Linear Models
7. Discrete/Qualitative Independent Variables
8. Model Evaluation
   - Predicted/True Plots, RMSE, and R-Squared
9. Regression Lab 2

Class 4 Lab Part 1

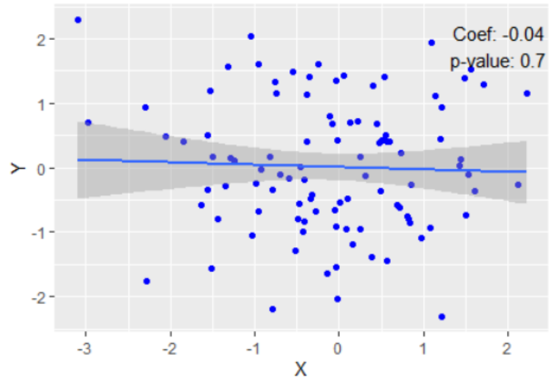


1. Estimate a regression model of city mpg on year, displacement, and engine cylinders and store this as 'mod3'
2. Interpret in words the coefficient for year
3. Interpret in words the coefficient for engine cylinders
4. If you finish and still have time, try using 'plot_model()' 'tab_model' and 'tidy' on 'mod3' (may need to load/install the packages tidymodels and sjPlot)

Hypothesis Test for Coefficients

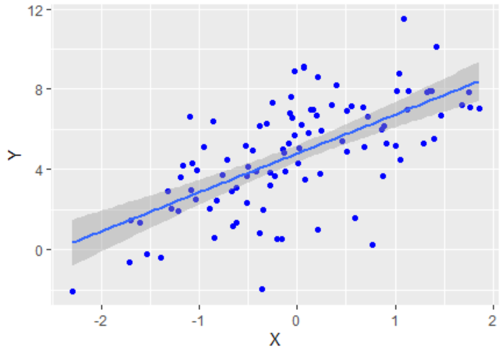
Null Hypothesis ($\boldsymbol{H_0}$):

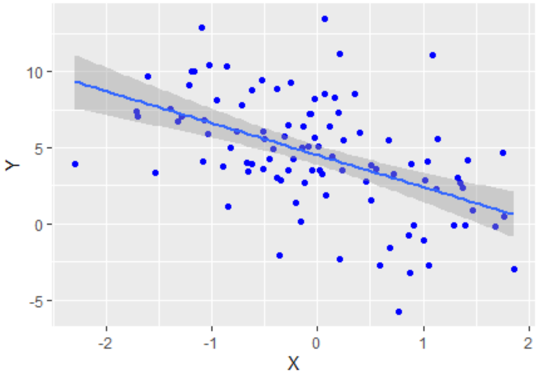
- There is no linear relationship between X and Y
- $\beta = 0$



Alternative Hypothesis ($\boldsymbol{H_1}$)

- There is some linear relationship between X and Y

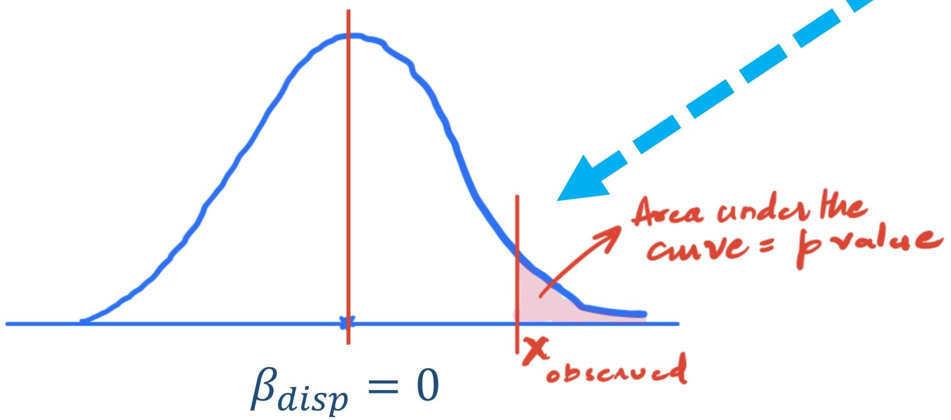




We either **reject the null hypothesis** or **fail to reject the null hypothesis**
Based on a chosen critical value of alpha

What Do p-values Measure?

```
> summary(mod1)

Call:
lm(formula = hwy ~ displ + cyl, data = mpg)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.5098 -2.1953 -0.2049  1.9023 14.9223 

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)    
(Intercept)  38.2162     1.0481  36.461 < 0.0000000000000002 ***
displ        -1.9599     0.5194  -3.773             0.000205 ***
cyl          -1.3537     0.4164  -3.251             0.001323 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.759 on 231 degrees of freedom
Multiple R-squared:  0.6049,	Adjusted R-squared:  0.6014 
F-statistic: 176.8 on 2 and 231 DF,  p-value: < 0.00000000000000022
```



P-value tells us the likelihood that, if the null hypothesis were true, we would receive a result as extreme as the one seen

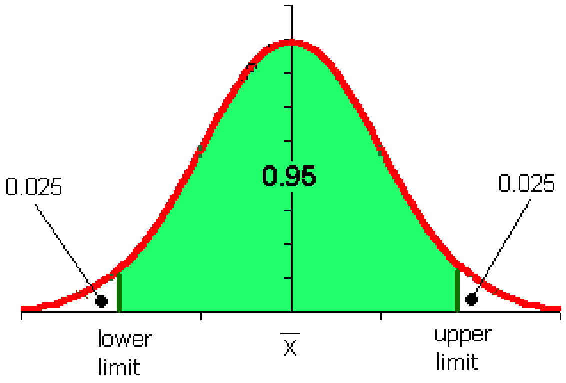
P-value for β_{disp} of 0.00205 say – assuming the null hypothesis of $\beta_{disp} = 0$ (flat slope) is actually true – we would see a coefficient as extreme as $\beta_{disp} = -1.9599$ 0.2% of the time.

We either **reject the null hypothesis** or **fail to reject the null hypothesis**
Based on a chosen critical value of alpha **(e.g. alpha = 0.05)**
~= incorrectly reject the null hypothesis 5% of the time even though null is true

What About Standard Error, T-Statistic and Confidence Interval?

```
> #------------------------------------------------------------
> # install package
> # install.packages('moderndive')
> library('moderndive')
> get_regression_table(mod1)
# A tibble: 3 x 7
  term      estimate std_error statistic p_value lower_ci upper_ci
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept    38.2      1.05      36.5    0        36.2    40.3
2 displ        -1.96     0.519     -3.77   0        -2.98   -0.936
3 cyl          -1.35     0.416     -3.25   0.001    -2.17   -0.533
```

- **Standard error** tells us the estimated standard deviation of the coefficient (the amount it varies across cases)
- ~= Measure of precision of estimate of coefficient
- Smaller SE relative to coefficient = more precise



$$t = \frac{\hat{\beta}_1 - 0}{SE\left(\hat{\beta}_1\right)}$$

- **Confidence Interval** for $\hat{\beta}$ has a x% probability of containing the true value of β
- E.g. 95% confidence interval contains β with prob 95%

- **T-stat of coefficient** is a transformation of the estimated coefficient divided by the standard error (or precision of estimate)
- Large t-stat in abs value -> big effect size

Measures of Overall Model Fit: F-Stat and R Squared

```
> summary(mod1)

Call:
lm(formula = hwy ~ displ + cyl, data = mpg)

Residuals:
    Min      1Q  Median      3Q     Max
-7.5098 -2.1953 -0.2049  1.9023 14.9223

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept)  38.2162     1.0481  36.461 < 0.0000000000000002 ***
displ        -1.9599     0.5194  -3.773             0.000205 ***
cyl          -1.3537     0.4164  -3.251             0.001323 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.759 on 231 degrees of freedom
Multiple R-squared:  0.6049,	Adjusted R-squared:  0.6014
F-statistic: 176.8 on 2 and 231 DF,  p-value: < 0.00000000000000022
```

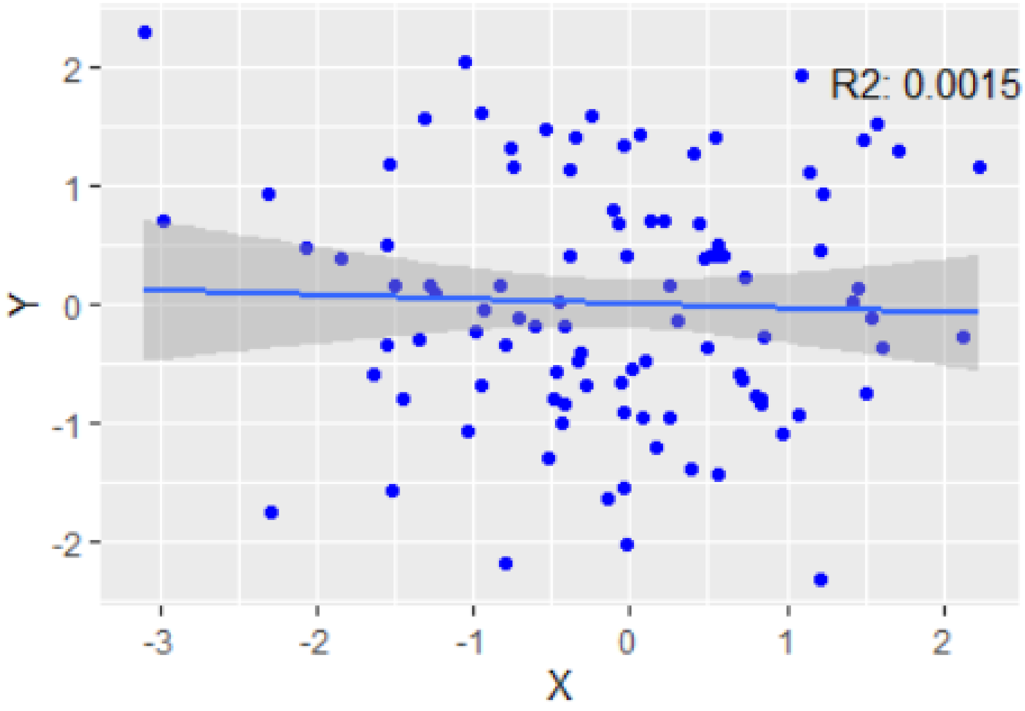
- **F-statistic** tells us whether all of the variables do better than a model with just an intercept
- Null = no effect of all variables except intercept.
- Almost always reject null. Outdated statistic.

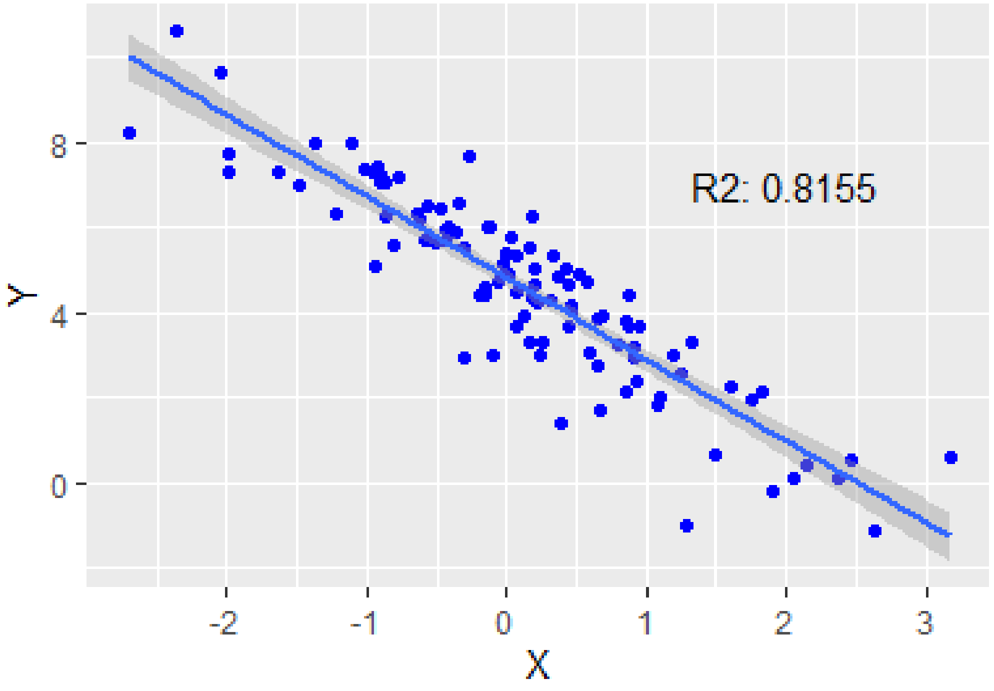
$$R^2 = \frac{TSS - RSS}{TSS} = \frac{TSS}{TSS} - \frac{RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- **R^2 or "Coefficient of Determination"**
- Measures fraction model explains of variation in outcome (y)
- $R^2 \in [0,1]$.
  - 1 = Explain all variation in y
  - 0 = Explain none of the variation
- Higher R^2 better prediction model
- Use adjusted (adjusts for extra variables)

High Versus Low R2





Class 4: Outline

1. Last Class Review:
    - Bias, Variance, Overfit, Underfit, Mean Squared Error
2. Linear Regression Review
3. Estimating Linear Models in R
4. Interpreting Linear Model Coefficients
5. Regression Lab 1
6. Inference/Hypothesis Testing in Linear Models
7. **Discrete/Qualitative Independent Variables**
8. Model Evaluation
    - Predicted/True Plots, RMSE, and R-Squared
9. Regression Lab 2

Factors: Like Strings But Better!

```
> DF <- data.frame(y = rnorm(5),
+                  x1 = 1:5,
+                  x2 = c("A","B","B","A","C"))
> head(DF)
            y x1 x2
1 -0.03030868  1  A
2  0.69707469  2  B
3 -0.93332824  3  B
4  1.35858876  4  A
5 -1.13368597  5  C
```

```
> DF <- DF %>%
+   mutate(x2 = as.factor(x2))
> glimpse(DF)
Rows: 5
Columns: 3
$ y  <dbl> -0.03030868, 0.69707469,
$ x1 <int> 1, 2, 3, 4, 5
$ x2 <fct> A, B, B, A, C
```

```
> fct_unique(DF$x2)
[1] A B C
Levels: A B C
> fct_count(DF$x2)
# A tibble: 3 x 2
  f         n
  <fct> <int>
1 A         2
2 B         2
3 C         1
```

- **Factors hold string values efficiently**
- Instead of holding a character string, it just holds a number and has a key which associates each number with a unique value of the string
- If we convert the character string x2 to a factor, we see A = 1, B = 2, C = 3, etc
- Will say more about working w/ factors but know that the package forcats is your friend

Incorporating Qualitative/Discrete Information Into Regressions

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_{state} \cdot x_{state}?$$

```
> model.matrix(~ x1 + x2,
+              data = DF)
  (Intercept) x1 x2B x2C
1           1  1   0   0
2           1  2   1   0
3           1  3   1   0
4           1  4   0   0
5           1  5   0   1
```

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_{stateB} \cdot StateB + \beta_{stateC} \cdot StateC?$$

- **Suppose A = State A, B = State B, etc.**
- How do include state into a regression?

- **model.matrix() function shows how we convert a factor to a regression matrix**
- Each "level" of the factor gets it own column, and a binary indicator of whether that level is true for that observation

- **Columns x2B and x2C are called "dummy" variables**
- Aka "one hot encoding" in machine learning

Why Does Every Factor Level Not Get Its Own Dummy Variable?

Intercept	Y	x1	x2_A	X2_B	X2_C
1	0.4	1	1	0	0
1	-0.5	2	0	1	0
1	-0.3	3	0	1	0
1	0.1	4	1	0	0
1	-0.8	5	0	0	1

x2
"A"
"B"
"B"
"A"
"C"

Why? Because estimates are computed as

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

Linear algebra requires $(X^T X)^{-1}$ to be full column rank i.e. each column of X must be linearly independent.

Intercept + X2_A + X_B + X2_C only "span" 3 dimensions

Excluded Base Level of Factor Becomes Base Level for Interpretation

		β_{x1}	β_{x2_A}	β_{x2_B}
Intercept	**Y**	**x1**	**x2_A**	**X2_B**
1	0.4	1	1	0
1	-0.5	2	0	1
1	-0.3	3	0	1
1	0.1	4	1	0
1	-0.8	5	0	0

Interpreting Dummy Variable Coefficients:

- β_{x2_A}: We estimate y will increase by β_{x2_A} *if it is of category A* relative to category C
- β_{x2_B}: We estimate y will increase by β_{x2_B} *if it is of category B* relative to category C

Binary/Dummy Variable coefficients can ONLY be interpreted relative to each other

Left out category (i.e. no column) is comparison category

Side Note, This is How Wage Discrimination Regressions Are Performed

$x_i = 1$, if female

$x_i = 0$, if male

$y_i =$ credit card balance

$$y_i = \beta_0 + \beta_1 \cdot x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 \cdot x_i + \epsilon_i & i = female \\ \beta_0 + \epsilon_i & i = male \end{cases}$$

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	509.80	33.13	15.389	< 0.0001
`gender[Female]`	19.73	46.05	0.429	0.6690

TABLE 3.7. *Least squares coefficient estimates associated with the regression of* `balance` *onto* `gender` *in the* `Credit` *data set. The linear model is given in (3.27). That is, gender is encoded as a dummy variable, as in (3.26).*

Interpreting Binary/Dummy Coefficients With mpg Dataset

```
> mpg <- mpg %>%
+   mutate(class = factor(class))
> mod2 <- lm(hwy ~ displ + class,
+            data = mpg)
> summary(mod2)

Call:
lm(formula = hwy ~ displ + class, data = mpg)

Residuals:
   Min      1Q Median      3Q     Max 
-5.572 -1.569 -0.245  1.355 14.724 

Coefficients:
                Estimate Std. Error t value              Pr(>|t|)    
(Intercept)      38.9533     1.7976  21.669 < 0.0000000000000002 ***
displ            -2.2976     0.2132 -10.778 < 0.0000000000000002 ***
classcompact     -5.3122     1.5283  -3.476             0.000610 ***
classmidsize     -4.9471     1.4722  -3.360             0.000914 ***
classminivan     -8.7986     1.5939  -5.520 0.0000000926135694 72 ***
classpickup     -11.9232     1.3687  -8.711 0.0000000000000000646 ***
classsubcompact  -4.6988     1.5097  -3.112             0.002095 ** 
classsuv        -10.5851     1.3268  -7.978 0.0000000000000074281 ***
```

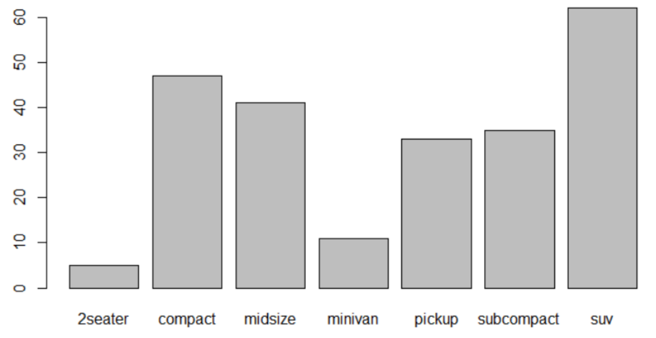
```
> levels(mpg$class)
[1] "2seater"    "compact"    "midsize"    "minivan"    "pickup"    
    "subcompact" "suv"
```

- $\beta_{compact}$: Holding engine size (displacement) fixed we estimated a compact car gets 5.31 *worse highway miles per gallon* relative to the excluded category!

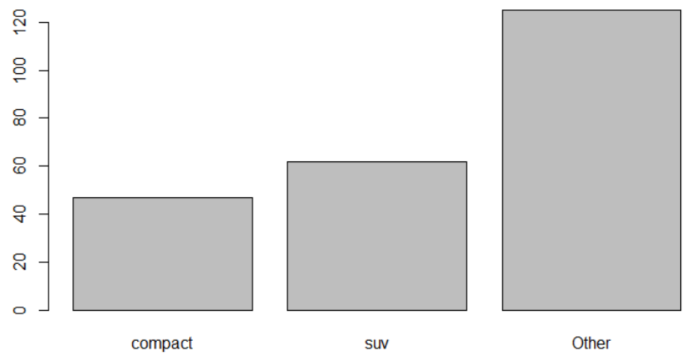
- What is the excluded category?

- By default it's the first level of the factor, here "2seater"

Changing Factor Levels with fct_lump




```
>
>
> mpg <- mpg %>%
+   mutate(class_lump = fct_lump(class, n = 2))
> levels(mpg$class_lump)
[1] "compact" "suv"     "Other"
> plot(mpg$class_lump)
```



- Suppose we want to change the levels of a factor?
- Many functions in ‘forcats’ to do this but fct_lump is useful.
  - n = 2 specifies how many explicit factors we want
- Here we’ve only given explicit labels to “compact” and “suv”. Every other level is placed into “other” category

Estimating Model With Simplified Factor Levels

```
Call:
lm(formula = hwy ~ displ + class_lump, data = mpg)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.4807 -2.3191 -0.2518  1.7201 15.3142 

Coefficients:
                Estimate Std. Error t value             Pr(>|t|)    
(Intercept)      35.1127     0.7402  47.435 < 0.0000000000000002 ***
displ            -2.9304     0.2224 -13.178 < 0.0000000000000002 ***
class_lumpsuv    -3.9243     0.8472  -4.632           0.00000605 ***
class_lumpOther  -0.8590     0.6668  -1.288                0.199    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.631 on 230 degrees of freedom
Multiple R-squared:  0.633,	Adjusted R-squared:  0.6282 
F-statistic: 132.2 on 3 and 230 DF,  p-value: < 0.00000000000000022
```

- What is the reference category?

```
> levels(mpg$class_lump)
[1] "compact" "suv"     "Other"  
>
```

- Change reference category with “relevel()”

```
Call:
lm(formula = hwy ~ displ + relevel(class_lump, ref = "Other"), 
    data = mpg)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.4807 -2.3191 -0.2518  1.7201 15.3142 

Coefficients:
                                          Estimate Std. Error t value             Pr(>|t|)
(Intercept)                                34.2536     0.8258  41.480 < 0.0000000000000002
displ                                      -2.9304     0.2224 -13.178 < 0.0000000000000002
relevel(class_lump, ref = "Other")compact   0.8590     0.6668   1.288                0.199
relevel(class_lump, ref = "Other")suv      -3.0653     0.6098  -5.027             0.000001
                                             
(Intercept)                               ***
displ                                     ***
relevel(class_lump, ref = "Other")compact    
relevel(class_lump, ref = "Other")suv     ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.631 on 230 degrees of freedom
Multiple R-squared:  0.633,	Adjusted R-squared:  0.6282 
F-statistic: 132.2 on 3 and 230 DF,  p-value: < 0.00000000000000022
```

Factor: Switches, Continuous: Sliders

$$hwy_i = \beta_0 + \beta_1 x_{budget} + \beta_2 \cdot x_{imdb\_score} + \epsilon_i$$

$$gross_i$$
$$= \beta_0 + \beta_1 x_{budget} + \beta_2 \cdot x_{imdb_{score}} + \beta_3$$
$$\cdot MichaelBay + \beta_4 \cdot StevenSpielberg + \epsilon_i$$

Source: https://twitter.com/andrewheiss/status/1171084259660107777?s=20

Factor Variable As Switches, Continuous Variables As Sliders

$$hwy_i = \beta_0 + \beta_1 x_{displ} + \epsilon_i$$

$$hwy_i \\ = \beta_0 + \beta_1 x_{compact} + \beta_2 \cdot x_{suv} + \epsilon_i$$

$$hwy_i \\ = \beta_0 + \beta_1 x_{compact} + \beta_2 \cdot x_{suv} \\ + \beta_3 \cdot x_{displ} + \epsilon_i$$

Class 4: Outline

1. Last Class Review:
   - Bias, Variance, Overfit, Underfit, Mean Squared Error
2. Linear Regression Review
3. Estimating Linear Models in R
4. Interpreting Linear Model Coefficients
5. Regression Lab 1
6. Inference/Hypothesis Testing in Linear Models
7. Discrete/Qualitative Independent Variables
8. **Model Evaluation**
   - **Predicted/True Plots, RMSE, and R-Squared**
9. Regression Lab 2

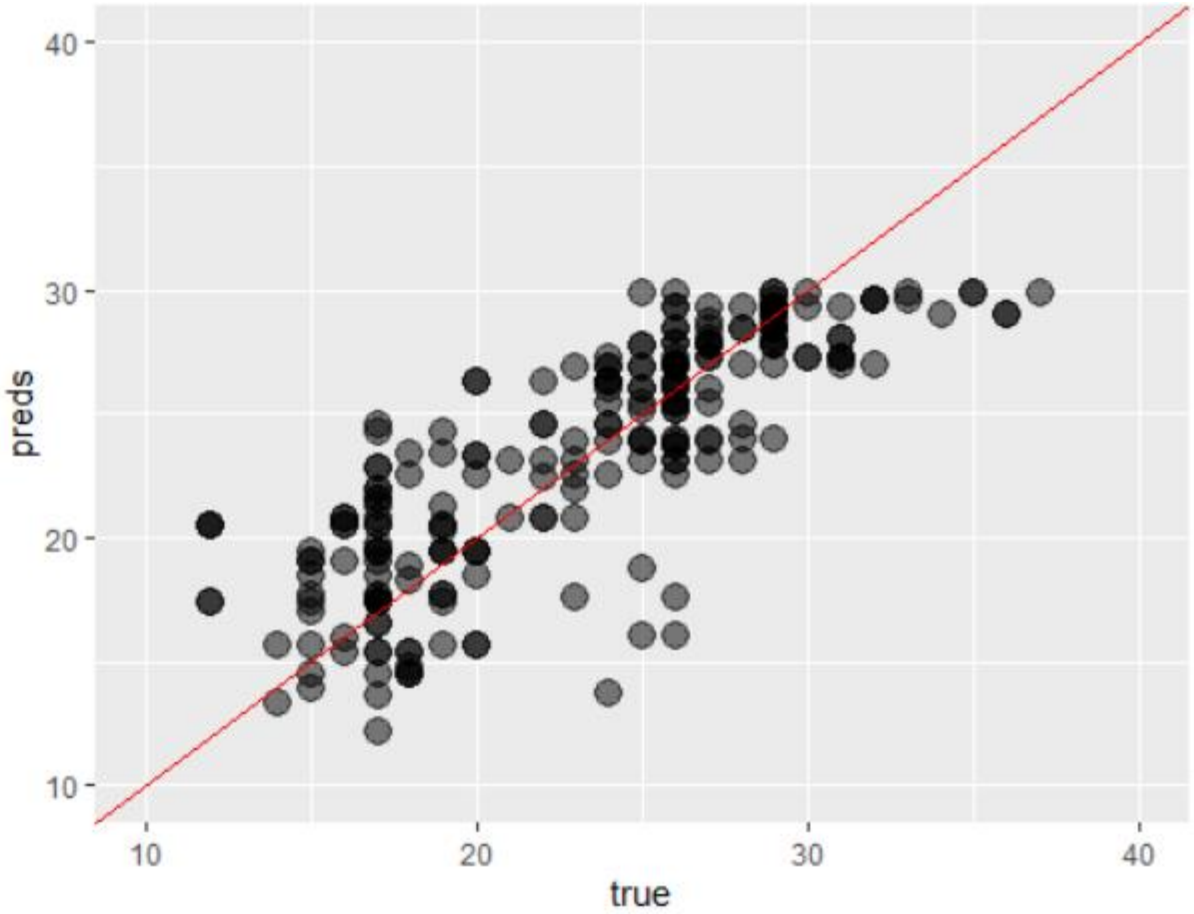
Predict() Function to Generate Model Predictions

```r
#----------------------------------------------------------
# Generate model predictions using "predict" function
#----------------------------------------------------------
# predict on the same data
preds <- predict(mod4)

# can also predict on a new dataset!
preds_new <- predict(mod,
                     newdata = newX)
```

```
>
> resids <- mod4$residuals
> resids <- mpg$hwy - preds
> mean(resids)
[1] 0.000000000000001447624
>
```

- Use the predict() function to generate model predictions using a trained/estimated model
- Note, we can also give it a new dataset (same Xs) with which to generate new predictions
- To generate residuals (true – predicted or $\hat{\epsilon}_i = y_i - \hat{y}_i$) some functions provide these for us, but we can calculate them ourselves
- Residuals (in-sample or on training set) are mean zero on average

Predicted True Plots



- Generally it's a good idea to plot your predictions against the actual values to see how your model performs

- Red 45 degree line = if prediction were perfect

```
# combine preds and resids into a data frame
results <- data.frame(
  preds = preds,
  true = mpg$hwy,
  resids = resids
)
ggplot(results,
       aes(x = true, y = preds)) +
  geom_point(alpha = 1/2, size = 4) +
  geom_abline(color = "red") +
  xlim(10,40) + ylim(10,40)
```

Class 4 Lab 2 (Time Permitting)

1. # 1. Use the mutate and the as.factor() functions to create a factor variable from the drv variable
2. Estimate a regression model predicting highway mpg as a function of displacement, year, and factor drive style (drv)
3. Interpret the coefficient on 'drvf'
4. Generate predictions and residuals for this model
5. Plot the model predictions against the true values