Multiple Linear Regression

Nate Wells

Math 141, 4/30/21

Outline

In this lecture, we will. . .

- Quantify variance in a linear model using the correlation coefficient
- Discuss metrics for selecting the "best" model
- Describe the forward-selection and backward-elimination procedures for model selection

Section 1

Multiple Linear Regression

Multiple Regression Model

In a **multiple linear regression model** (MLR), we express the response variable Y as a linear combination of k explanatory variables $X_1, X_2, \ldots, X_k$:

$$\hat{Y} = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \cdots + \beta_k \cdot X_k$$

Multiple Regression Model

In a **multiple linear regression model** (MLR), we express the response variable Y as a linear combination of k explanatory variables $X_1, X_2, \ldots, X_k$:

$$\hat{Y} = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \cdots + \beta_k \cdot X_k$$

We use the following R code to fit and summarize a linear model:

```
mod<-lm(Y ~ X1 + X2 + X3, data = my_data)
get_regression_table(mod)
```

```
## # A tibble: 4 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept     3.26     7.94       0.41   0.686  -13.3     19.8
## 2 X1           -1.24     0.313     -3.95   0.001   -1.89    -0.584
## 3 X2            2.68     1.94       1.38   0.182   -1.36     6.72
## 4 X3            3.20     0.397      8.06   0        2.37     4.02
```

Multiple Regression Model

In a **multiple linear regression model** (MLR), we express the response variable Y as a linear combination of k explanatory variables $X_1, X_2, \ldots, X_k$:

$$\hat{Y} = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \cdots + \beta_k \cdot X_k$$

We use the following R code to fit and summarize a linear model:

```
mod<-lm(Y ~ X1 + X2 + X3, data = my_data)
get_regression_table(mod)
```

```
## # A tibble: 4 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept     3.26     7.94       0.41   0.686   -13.3    19.8
## 2 X1           -1.24     0.313     -3.95   0.001    -1.89   -0.584
## 3 X2            2.68     1.94       1.38   0.182    -1.36    6.72
## 4 X3            3.20     0.397      8.06   0         2.37    4.02
```

- Which gives us our linear regression formula:

$$\hat{Y} = 3.26 - 1.24 \cdot X_1 + 2.68 \cdot X_2 + 3.2 \cdot X_3$$

How Strong is a Linear Model?

For SLR, we used the correlation coefficient R to assess model strength.

How Strong is a Linear Model?

For SLR, we used the correlation coefficient R to assess model strength.

- The value R^2 has utility too! It represents the percentage of variability in values of the response variable just due to variability in explanatory variable.

How Strong is a Linear Model?

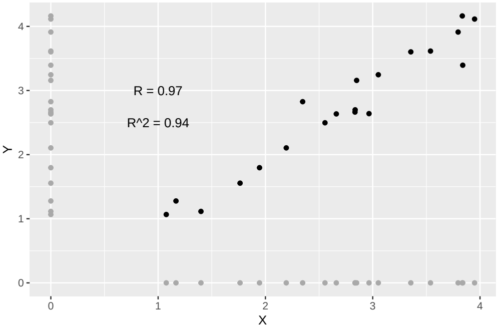
For SLR, we used the correlation coefficient R to assess model strength.

- The value R^2 has utility too! It represents the percentage of variability in values of the response variable just due to variability in explanatory variable.
  - If $R \approx \pm 1$, then $R^2 \approx 1$: nearly all the variability in response is due to variability in the explanatory variable.

How Strong is a Linear Model?

For SLR, we used the correlation coefficient R to assess model strength.

- The value R^2 has utility too! It represents the percentage of variability in values of the response variable just due to variability in explanatory variable.
  - If $R \approx \pm 1$, then $R^2 \approx 1$: nearly all the variability in response is due to variability in the explanatory variable.



Model Strength for MLR

We can also compute R^2 for MLR. In particular,

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in outcomes}} = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)}$$

Model Strength for MLR

We can also compute R^2 for MLR. In particular,

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in outcomes}} = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)}$$

- Usually, we use software to compute

Model Strength for MLR

We can also compute R^2 for MLR. In particular,

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in outcomes}} = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)}$$

- Usually, we use software to compute

```
mod<-lm(Y ~ X1 + X2 + X3, data = my_data)
get_regression_summaries(mod)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.798         0.769  17.0  4.13  4.50      27.6       0     3    25
```

Model Strength for MLR

We can also compute R^2 for MLR. In particular,

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in outcomes}} = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)}$$

- Usually, we use software to compute

```
mod<-lm(Y ~ X1 + X2 + X3, data = my_data)
get_regression_summaries(mod)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.798         0.769  17.0  4.13  4.50      27.6       0     3    25
```

- But it turns out this formula gives a **biased** estimate of the variability in the *population* explained by the model.

Model Strength for MLR

We can also compute R^2 for MLR. In particular,

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in outcomes}} = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)}$$

- Usually, we use software to compute

```
mod<-lm(Y ~ X1 + X2 + X3, data = my_data)
get_regression_summaries(mod)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.798         0.769  17.0  4.13  4.50      27.6       0     3    25
```

- But it turns out this formula gives a **biased** estimate of the variability in the *population* explained by the model.
- Instead, we use the adjusted R:

$$R^2 = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)} \cdot \frac{n-1}{n-k-1}$$

Model Strength for MLR

We can also compute R^2 for MLR. In particular,

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in outcomes}} = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)}$$

- Usually, we use software to compute

```
mod<-lm(Y ~ X1 + X2 + X3, data = my_data)
get_regression_summaries(mod)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.798         0.769  17.0  4.13  4.50      27.6       0     3    25
```

- But it turns out this formula gives a **biased** estimate of the variability in the *population* explained by the model.
- Instead, we use the adjusted R:

$$R^2 = 1 - \frac{\mathrm{Var}(e_i)}{\mathrm{Var}(y_i)} \cdot \frac{n-1}{n-k-1}$$

- This adjusted R^2 is usually a bit smaller than R^2, and the difference decreases as n gets large.

Section 2

Model Building

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).
- Let $I_{\text{Sophomore}}, I_{\text{Junior}}, I_{\text{Senior}}$ be the indicator functions for the respective levels.

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).
- Let $I_{\text{Sophomore}}, I_{\text{Junior}}, I_{\text{Senior}}$ be the indicator functions for the respective levels.
  - That is, $I_{\text{Sophomore}}(x) = 1$ if the observation x is a first year, and 0 otherwise.

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).
- Let $I_{\text{Sophomore}}, I_{\text{Junior}}, I_{\text{Senior}}$ be the indicator functions for the respective levels.
  - That is, $I_{\text{Sophomore}}(x) = 1$ if the observation x is a first year, and 0 otherwise.
- An MLR model could be

$$\hat{Y} = 34.2 + 0.6 \cdot X_1 + 0.9 \cdot I_{\text{Sophomore}} - 3.6 \cdot I_{\text{Junior}} - 0.6 \cdot I_{\text{Senior}}$$

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).
- Let $I_{\text{Sophomore}}, I_{\text{Junior}}, I_{\text{Senior}}$ be the indicator functions for the respective levels.
  - That is, $I_{\text{Sophomore}}(x) = 1$ if the observation x is a first year, and 0 otherwise.
- An MLR model could be

$$\hat{Y} = 34.2 + 0.6 \cdot X_1 + 0.9 \cdot I_{\text{Sophomore}} - 3.6 \cdot I_{\text{Junior}} - 0.6 \cdot I_{\text{Senior}}$$

  - To predict your final exam score, start with 34.2 points, add 60% of your 1st midterm score, and then add 0.9 points if you are a sophomore, subtract 3.6 points if you are a junior, or subtract 0.6 point if you are a senior.

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).
- Let $I_{\text{Sophomore}}, I_{\text{Junior}}, I_{\text{Senior}}$ be the indicator functions for the respective levels.
    - That is, $I_{\text{Sophomore}}(x) = 1$ if the observation x is a first year, and 0 otherwise.
- An MLR model could be

$$\hat{Y} = 34.2 + 0.6 \cdot X_1 + 0.9 \cdot I_{\text{Sophomore}} - 3.6 \cdot I_{\text{Junior}} - 0.6 \cdot I_{\text{Senior}}$$

    - To predict your final exam score, start with 34.2 points, add 60% of your 1st midterm score, and then add 0.9 points if you are a sophomore, subtract 3.6 points if you are a junior, or subtract 0.6 point if you are a senior.
- Why no indicator for first-years?

Modeling Exam Grades

Suppose we want to fit a model that predicts final exam score Y as a function of 1st midterm score X_1 and year in school X_2.

- Note that both Y and X_1 are quantitative, but X_2 is categorical with 4 levels (First-year, Sophomore, Junior, Senior).
- Let $I_{\text{Sophomore}}, I_{\text{Junior}}, I_{\text{Senior}}$ be the indicator functions for the respective levels.
  - That is, $I_{\text{Sophomore}}(x) = 1$ if the observation x is a first year, and 0 otherwise.
- An MLR model could be

$$\hat{Y} = 34.2 + 0.6 \cdot X_1 + 0.9 \cdot I_{\text{Sophomore}} - 3.6 \cdot I_{\text{Junior}} - 0.6 \cdot I_{\text{Senior}}$$

  - To predict your final exam score, start with 34.2 points, add 60% of your 1st midterm score, and then add 0.9 points if you are a sophomore, subtract 3.6 points if you are a junior, or subtract 0.6 point if you are a senior.
- Why no indicator for first-years?
  - If you aren't a sophomore, junior, or senior, you must be a first-year.

Data Exploration

Midterm scores, Final score, and year are recorded for 50 (fictitious) intro stat students

Data Exploration

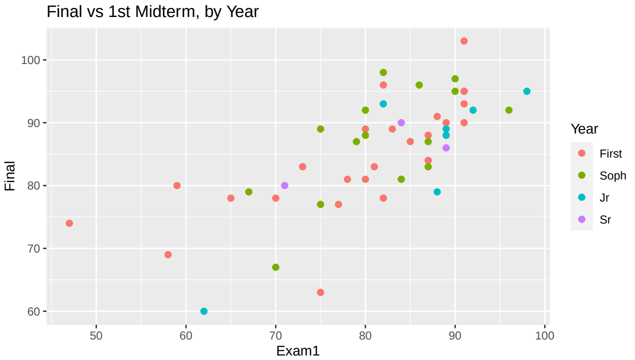
Midterm scores, Final score, and year are recorded for 50 (fictitious) intro stat students

```
##   Exam1 Exam2 Final  Year
## 1    73    82    83 First
## 2    87    90    83  Soph
## 3    89    89    86    Sr
## 4    58    65    69 First
## 5    80    77    88  Soph
```

Data Exploration

Midterm scores, Final score, and year are recorded for 50 (fictitious) intro stat students

```
##   Exam1 Exam2 Final  Year
## 1    73    82    83 First
## 2    87    90    83  Soph
## 3    89    89    86    Sr
## 4    58    65    69 First
## 5    80    77    88  Soph
```



Model Fitting

Using the `lm` function, we create a linear model for Final score as a function of 1st Midterm score and Year:

```
mod_mt_year<-lm(Final ~ Exam1 + Year, data = Grades)
```

Model Fitting

Using the `lm` function, we create a linear model for Final score as a function of 1st Midterm score and Year:

```
mod_mt_year<-lm(Final ~ Exam1 + Year, data = Grades)
```

And we examine the model using the `get_regression_table` function

```
get_regression_table(mod_mt_year)
```

```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2        7.25     4.72    0        19.6     48.8
## 2 Exam1        0.636      0.09     7.06    0         0.455    0.817
## 3 YearSoph     0.929      2.12     0.438   0.664    -3.35     5.20
## 4 YearJr      -3.58       2.82    -1.27    0.212    -9.26     2.11
## 5 YearSr      -0.598      3.95    -0.151   0.88     -8.56     7.36
```

Model Fitting

Using the `lm` function, we create a linear model for Final score as a function of 1st Midterm score and Year:

```
mod_mt_year<-lm(Final ~ Exam1 + Year, data = Grades)
```

And we examine the model using the `get_regression_table` function

```
get_regression_table(mod_mt_year)
```

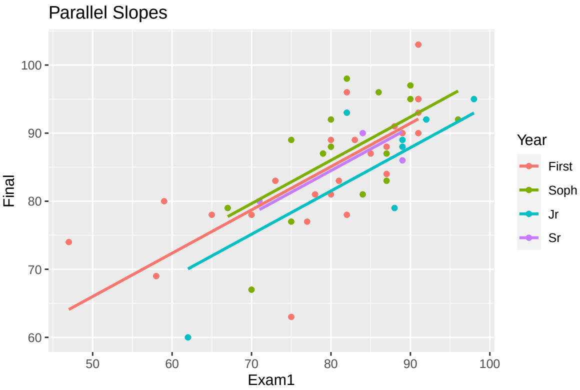
```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2        7.25     4.72    0        19.6     48.8
## 2 Exam1        0.636      0.09     7.06    0         0.455    0.817
## 3 YearSoph     0.929      2.12     0.438   0.664    -3.35     5.20
## 4 YearJr      -3.58       2.82    -1.27    0.212    -9.26     2.11
## 5 YearSr      -0.598      3.95    -0.151   0.88     -8.56     7.36
```

From the table, our regression equation is

$$\hat{Y} = 34.2 + 0.6 \cdot X_1 + 0.9 \cdot I_{\text{Sophomore}} - 3.6 \cdot I_{\text{Junior}} - 0.6 \cdot I_{\text{Senior}}$$

Graph of Parallel Slopes Model

```
ggplot(Grades, aes( x = Exam1, y = Final, color = Year))+
  geom_point()+
  labs(title = "Parallel Slopes")+
  geom_parallel_slopes(se = F) ### Note the different geom
```



Section 3

Model Selection

Model Selection

Does knowing a students year in school really add significant predictive power to the model?

Model Selection

Does knowing a students year in school really add significant predictive power to the model?

```
get_regression_table(mod_mt_year)
```

```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2        7.25     4.72    0        19.6     48.8
## 2 Exam1        0.636      0.09     7.06    0         0.455    0.817
## 3 YearSoph     0.929      2.12     0.438   0.664    -3.35     5.20
## 4 YearJr      -3.58       2.82    -1.27    0.212    -9.26     2.11
## 5 YearSr      -0.598      3.95    -0.151   0.88     -8.56     7.36
```

Model Selection

Does knowing a students year in school really add significant predictive power to the model?

```
get_regression_table(mod_mt_year)
```

```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2        7.25     4.72    0        19.6     48.8
## 2 Exam1        0.636      0.09     7.06    0         0.455    0.817
## 3 YearSoph     0.929      2.12     0.438   0.664    -3.35     5.20
## 4 YearJr      -3.58       2.82    -1.27    0.212    -9.26     2.11
## 5 YearSr      -0.598      3.95    -0.151   0.88     -8.56     7.36
```

- In most cases, changing year in school changes exam score by less than 1 point.

Model Selection

Does knowing a students year in school really add significant predictive power to the model?

```
get_regression_table(mod_mt_year)
```

```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2       7.25      4.72    0        19.6     48.8
## 2 Exam1        0.636     0.09      7.06    0         0.455    0.817
## 3 YearSoph     0.929     2.12      0.438   0.664    -3.35     5.20
## 4 YearJr      -3.58      2.82     -1.27    0.212    -9.26     2.11
## 5 YearSr      -0.598     3.95     -0.151   0.88     -8.56     7.36
```

- In most cases, changing year in school changes exam score by less than 1 point.
- And for seniors, sample size should be a concern ($n = 3$)

Model Selection

Does knowing a students year in school really add significant predictive power to the model?

```
get_regression_table(mod_mt_year)
```

```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2        7.25     4.72    0        19.6     48.8
## 2 Exam1        0.636      0.09     7.06    0         0.455    0.817
## 3 YearSoph     0.929      2.12     0.438   0.664    -3.35     5.20
## 4 YearJr      -3.58       2.82    -1.27    0.212    -9.26     2.11
## 5 YearSr      -0.598      3.95    -0.151   0.88     -8.56     7.36
```

- In most cases, changing year in school changes exam score by less than 1 point.
- And for seniors, sample size should be a concern ($n = 3$)
- Using a *t*-test against the null hypothesis that the true coefficient is 0, we see that none of sophomore, junior or senior dummy variables are significant at the $\alpha = 0.05$ level

Model Selection

Does knowing a students year in school really add significant predictive power to the model?

```
get_regression_table(mod_mt_year)
```

```
## # A tibble: 5 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   34.2       7.25      4.72    0       19.6     48.8
## 2 Exam1        0.636     0.09      7.06    0        0.455    0.817
## 3 YearSoph     0.929     2.12      0.438   0.664   -3.35     5.20
## 4 YearJr      -3.58      2.82     -1.27    0.212   -9.26     2.11
## 5 YearSr      -0.598     3.95     -0.151   0.88    -8.56     7.36
```

- In most cases, changing year in school changes exam score by less than 1 point.
- And for seniors, sample size should be a concern ($n = 3$)
- Using a *t*-test against the null hypothesis that the true coefficient is 0, we see that none of sophomore, junior or senior dummy variables are significant at the $\alpha = 0.05$ level
    - It is plausible that there truly is no difference in scores between years, and any observed difference is just due to random chance.

Model Selection, cont'd

- On the other hand, we do have data on year in school, so why not use it?

Model Selection, cont'd

- On the other hand, we do have data on year in school, so why not use it?
- We also have data on 2nd exam, so why not include it as well?

Model Selection, cont'd

- On the other hand, we do have data on year in school, so why not use it?
- We also have data on 2nd exam, so why not include it as well?
- A regression model which includes all measured variables is called the **full model**

Model Selection, cont'd

- On the other hand, we do have data on year in school, so why not use it?
- We also have data on 2nd exam, so why not include it as well?
- A regression model which includes all measured variables is called the **full model**

```
mod_full<-lm(Final ~ Exam1 + Exam2 + Year, data = Grades)
get_regression_table(mod_full)
```

```
## # A tibble: 6 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   31.0      8.19       3.78    0        14.5     47.5
## 2 Exam1        0.511    0.173      2.96    0.005     0.163    0.858
## 3 Exam2        0.162    0.19       0.853   0.398    -0.221    0.546
## 4 YearSoph     0.421    2.21       0.19    0.85     -4.04     4.88
## 5 YearJr      -3.24     2.86      -1.14    0.262    -9.00     2.52
## 6 YearSr      -0.654    3.96      -0.165   0.87     -8.64     7.33
```

Model Selection, cont'd

- On the other hand, we do have data on year in school, so why not use it?
- We also have data on 2nd exam, so why not include it as well?
- A regression model which includes all measured variables is called the **full model**

```
mod_full<-lm(Final ~ Exam1 + Exam2 + Year, data = Grades)
get_regression_table(mod_full)
```

```
## # A tibble: 6 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   31.0      8.19      3.78     0       14.5     47.5
## 2 Exam1        0.511    0.173     2.96     0.005    0.163    0.858
## 3 Exam2        0.162    0.19      0.853    0.398   -0.221    0.546
## 4 YearSoph     0.421    2.21      0.19     0.85    -4.04     4.88
## 5 YearJr      -3.24     2.86     -1.14     0.262   -9.00     2.52
## 6 YearSr      -0.654    3.96     -0.165    0.87    -8.64     7.33
```

- Why don't we always use the full model?

Occam's Razor

"Numquam ponenda est pluralitas sine necessitate."

Plurality must never be posited without necessity

— William of Ockham, c. 1300

Occam's Razor

"Numquam ponenda est pluralitas sine necessitate."

Plurality must never be posited without necessity

— William of Ockham, c. 1300

- All else held equal, a simpler model makes better predictions.

Occam's Razor

"Numquam ponenda est pluralitas sine necessitate."

Plurality must never be posited without necessity

— William of Ockham, c. 1300

- All else held equal, a simpler model makes better predictions.
- Adding additional variables to a model increases the likelihood that the model fits to particular features of the sample, rather than general trends in the population.

Occam's Razor

"Numquam ponenda est pluralitas sine necessitate."

Plurality must never be posited without necessity

— William of Ockham, c. 1300

- All else held equal, a simpler model makes better predictions.
- Adding additional variables to a model increases the likelihood that the model fits to particular features of the sample, rather than general trends in the population.
- On the other hand, failing to include important variables may lead to missing relevant relations

Occam's Razor

"Numquam ponenda est pluralitas sine necessitate."

Plurality must never be posited without necessity

— William of Ockham, c. 1300

- All else held equal, a simpler model makes better predictions.
- Adding additional variables to a model increases the likelihood that the model fits to particular features of the sample, rather than general trends in the population.
- On the other hand, failing to include important variables may lead to missing relevant relations
- In statistical/machine learning, this is oft referred to as the *Bias-Variance trade-off*

Selection Criteria

There are several numbers we can use to assess the strength of a model:

1. Individual p-values
2. R^2
3. Residual standard errors
4. Overall model p-value
5. F-statistic from ANOVA
6. Adjusted R^2

Selection Criteria

There are several numbers we can use to assess the strength of a model:

1. Individual p-values
2. R^2
3. Residual standard errors
4. Overall model p-value
5. F-statistic from ANOVA
6. Adjusted R^2

Some numbers lead to decreased Bias at the cost of increased Variance. Others do the opposite. Some are relatively balanced.

Selection Criteria

There are several numbers we can use to assess the strength of a model:

1. Individual p-values
2. R^2
3. Residual standard errors
4. Overall model p-value
5. F-statistic from ANOVA
6. Adjusted R^2

Some numbers lead to decreased Bias at the cost of increased Variance. Others do the opposite. Some are relatively balanced.

- Choices are usually discipline specific, and the particular trade-offs are discussed in advanced statistics and statistical learning courses (like Math 243!)

Selection Criteria

There are several numbers we can use to assess the strength of a model:

1. Individual p-values
2. R^2
3. Residual standard errors
4. Overall model p-value
5. F-statistic from ANOVA
6. Adjusted R^2

Some numbers lead to decreased Bias at the cost of increased Variance. Others do the opposite. Some are relatively balanced.

- Choices are usually discipline specific, and the particular trade-offs are discussed in advanced statistics and statistical learning courses (like Math 243!)

We'll focus on individual P-values and adjusted R^2

Section 4

Selection Strategies

Backward-Elimination

- One of the most common model selection techniques is **backward-elimination**:

Backward-Elimination

- One of the most common model selection techniques is **backward-elimination**:
  - Begin with the full model (with all predictors)
  - Eliminate the predictor with greatest p-value larger than desired significance level
  - Refit the model with remaining predictors and repeat until all are significant

Backward-Elimination on Exam Scores I

```
mod_full<-lm(Final ~ Exam1 + Exam2 + Year, data = Grades)
get_regression_table(mod_full)
```

```
## # A tibble: 6 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   31.0       8.19      3.78    0        14.5     47.5
## 2 Exam1        0.511     0.173     2.96    0.005     0.163    0.858
## 3 Exam2        0.162     0.19      0.853   0.398    -0.221    0.546
## 4 YearSoph     0.421     2.21      0.19    0.85     -4.04     4.88
## 5 YearJr      -3.24      2.86     -1.14    0.262    -9.00     2.52
## 6 YearSr      -0.654     3.96     -0.165   0.87     -8.64     7.33
```

```
get_regression_summaries(mod_full)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.541         0.489  36.9  6.08  6.48      10.4       0     5    50
```

Backward-Elimination on Exam Scores I

```
mod_full<-lm(Final ~ Exam1 + Exam2 + Year, data = Grades)
get_regression_table(mod_full)
```

```
## # A tibble: 6 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   31.0       8.19      3.78    0        14.5     47.5
## 2 Exam1        0.511     0.173     2.96    0.005     0.163    0.858
## 3 Exam2        0.162     0.19      0.853   0.398    -0.221    0.546
## 4 YearSoph     0.421     2.21      0.19    0.85     -4.04     4.88
## 5 YearJr      -3.24      2.86     -1.14    0.262    -9.00     2.52
## 6 YearSr      -0.654     3.96     -0.165   0.87     -8.64     7.33
```

```
get_regression_summaries(mod_full)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.541         0.489  36.9  6.08  6.48      10.4       0     5    50
```

- The p-values for each year dummy variable are larger than 0.05, so we eliminate year from our model.

Backward-Elimination on Exam Scores I

```
mod_full<-lm(Final ~ Exam1 + Exam2 + Year, data = Grades)
get_regression_table(mod_full)
```

```
## # A tibble: 6 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   31.0       8.19      3.78    0        14.5     47.5
## 2 Exam1        0.511     0.173     2.96    0.005     0.163    0.858
## 3 Exam2        0.162     0.19      0.853   0.398    -0.221    0.546
## 4 YearSoph     0.421     2.21      0.19    0.85     -4.04     4.88
## 5 YearJr      -3.24      2.86     -1.14    0.262    -9.00     2.52
## 6 YearSr      -0.654     3.96     -0.165   0.87     -8.64     7.33
```

```
get_regression_summaries(mod_full)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.541         0.489  36.9  6.08  6.48      10.4       0     5    50
```

- The p-values for each year dummy variable are larger than 0.05, so we eliminate year from our model.
    - Note: Including categorical variables is “all-or-nothing”; either we include all levels of the variable or we include none. If at least 1 level is significant, we’ll leave all in the model.

Backward-Elimination on Exam Scores II

Let's fit with just the 2 exam scores:

Backward-Elimination on Exam Scores II

Let's fit with just the 2 exam scores:

```
mod_no_year<-lm(Final ~ Exam1 + Exam2 , data = Grades)
get_regression_table(mod_no_year)
```

```
## # 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   30.9       8.00       3.86   0        14.8     47.0
## 2 Exam1        0.447     0.161      2.78   0.008     0.124    0.77
## 3 Exam2        0.221     0.176      1.26   0.215    -0.133    0.575
```

- Note that the estimates changed in the reduced model.

Backward-Elimination on Exam Scores II

Let's fit with just the 2 exam scores:

```
mod_no_year<-lm(Final ~ Exam1 + Exam2 , data = Grades)
get_regression_table(mod_no_year)
```

```
## # 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   30.9       8.00       3.86   0        14.8     47.0
## 2 Exam1        0.447     0.161      2.78   0.008     0.124    0.77
## 3 Exam2        0.221     0.176      1.26   0.215    -0.133    0.575
```

- Note that the estimates changed in the reduced model.
- The p-values for the `Exam2` variable is larger than 0.05, so we eliminate `Exam2`

Backward-Elimination on Exam Scores II

- But before we create a new model, let's consider R^2:

Backward-Elimination on Exam Scores II

- But before we create a new model, let's consider R^2:

```
get_regression_summaries(mod_full)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.541         0.489  36.9  6.08  6.48      10.4       0     5    50
```

```
get_regression_summaries(mod_no_year)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.525         0.505  38.2  6.18  6.38      26.0       0     2    50
```

Backward-Elimination on Exam Scores II

- But before we create a new model, let's consider R^2:

```
get_regression_summaries(mod_full)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.541         0.489  36.9  6.08  6.48      10.4       0     5    50
```

```
get_regression_summaries(mod_no_year)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.525         0.505  38.2  6.18  6.38      26.0       0     2    50
```

- Note that while R^2 decreased from the full model to the reduced model, adjusted R^2 actually increased!

Backward-Elimination on Exam Scores II

- But before we create a new model, let's consider R^2:

```
get_regression_summaries(mod_full)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.541         0.489  36.9  6.08  6.48      10.4       0     5    50
```

```
get_regression_summaries(mod_no_year)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.525         0.505  38.2  6.18  6.38      26.0       0     2    50
```

- Note that while R^2 decreased from the full model to the reduced model, adjusted R^2 actually increased!
- Recall that adjusted R^2 penalizes R^2 by the number of variables in the model.

Backward-Elimination on Exam Scores III

Let's fit the model with just `Exam1`

Backward-Elimination on Exam Scores III

Let's fit the model with just Exam1

```
mod_exam1<-lm(Final ~ Exam1  , data = Grades)
get_regression_table(mod_exam1)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   35.5       7.14       4.97       0   21.2     49.9
## 2 Exam1        0.617     0.087      7.06       0    0.441    0.792
```

```
get_regression_summaries(mod_exam1)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.509         0.499  39.5  6.29  6.42      49.8       0     1    50
```

Backward-Elimination on Exam Scores III

Let's fit the model with just Exam1

```
mod_exam1<-lm(Final ~ Exam1  , data = Grades)
get_regression_table(mod_exam1)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   35.5       7.14       4.97       0   21.2     49.9
## 2 Exam1        0.617     0.087      7.06       0    0.441    0.792
```

```
get_regression_summaries(mod_exam1)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.509         0.499  39.5  6.29  6.42      49.8       0     1    50
```

- All remaining variables are significant, so this is the model we use.

Backward-Elimination on Exam Scores IV

Out of curiosity, what would the model with Score $\sim$ Exam2 look like?

```
mod_exam2<-lm(Final ~ Exam2 , data = Grades)
get_regression_table(mod_exam2)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   32.9       8.51       3.86       0   15.8     50.0
## 2 Exam2        0.633     0.102      6.23       0    0.429    0.838
```

Backward-Elimination on Exam Scores IV

Out of curiosity, what would the model with Score ∼ Exam2 look like?

```
mod_exam2<-lm(Final ~ Exam2 , data = Grades)
get_regression_table(mod_exam2)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   32.9       8.51       3.86       0   15.8     50.0
## 2 Exam2        0.633     0.102      6.23       0    0.429    0.838
```

```
get_regression_table(mod_exam1)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   35.5       7.14       4.97       0   21.2     49.9
## 2 Exam1        0.617     0.087      7.06       0    0.441    0.792
```

Backward-Elimination on Exam Scores IV

Out of curiosity, what would the model with Score ~ Exam2 look like?

```
mod_exam2<-lm(Final ~ Exam2 , data = Grades)
get_regression_table(mod_exam2)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   32.9       8.51       3.86       0   15.8     50.0
## 2 Exam2        0.633     0.102      6.23       0    0.429    0.838
```

```
get_regression_table(mod_exam1)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept   35.5       7.14       4.97       0   21.2     49.9
## 2 Exam1        0.617     0.087      7.06       0    0.441    0.792
```

Backward-Elimination on Exam Scores V

- Why eliminate Exam 2 if it is a significant predictor of Final score?

Backward-Elimination on Exam Scores V

- Why eliminate Exam 2 if it is a significant predictor of Final score?
  - While each exam, on its own, is a good predictor of the final score, but if exam1 is already in the model, exam2 becomes unnecessary and redundant.

Backward-Elimination on Exam Scores V

- Why eliminate Exam 2 if it is a significant predictor of Final score?
  - While each exam, on its own, is a good predictor of the final score, but if exam1 is already in the model, exam2 becomes unnecessary and redundant.

```
get_correlation(data = Grades, Exam1 ~ Exam2)
```

```
##    cor
## 1 0.84
```

Backward-Elimination on Exam Scores V

- Why eliminate Exam 2 if it is a significant predictor of Final score?
  - While each exam, on its own, is a good predictor of the final score, but if exam1 is already in the model, exam2 becomes unnecessary and redundant.

```
get_correlation(data = Grades, Exam1 ~ Exam2)
```

```
##    cor
## 1 0.84
```

- So which model should we go with?

Backward-Elimination on Exam Scores V

- Why eliminate Exam 2 if it is a significant predictor of Final score?
  - While each exam, on its own, is a good predictor of the final score, but if exam1 is already in the model, exam2 becomes unnecessary and redundant.

```
get_correlation(data = Grades, Exam1 ~ Exam2)
```

```
##    cor
## 1 0.84
```

- So which model should we go with?

```
get_regression_summaries(mod_exam1)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.509         0.499  39.5  6.29  6.42      49.8       0     1    50
```

```
get_regression_summaries(mod_exam2)
```

```
## # A tibble: 1 x 9
##   r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
##       <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
## 1     0.447         0.435  44.5  6.67  6.81      38.8       0     1    50
```

Forward-selection

- The second most common model selection techniques is **forward-selection**:

Forward-selection

- The second most common model selection techniques is **forward-selection**:
  - Begin with a model with no predictors
  - For each possible predictor, create a model with that predictor added.
  - Pick the predictor model where the added predictor had the smallest significant p-value.
  - Repeat the previous 2 steps until no added predictors have significant p-values.

Forward-selection

- The second most common model selection techniques is **forward-selection**:
  - Begin with a model with no predictors
  - For each possible predictor, create a model with that predictor added.
  - Pick the predictor model where the added predictor had the smallest significant p-value.
  - Repeat the previous 2 steps until no added predictors have significant p-values.
- There is no guarantee that forward-selection and backward-elimination will reach the same model.

Forward-selection

- The second most common model selection techniques is **forward-selection**:
  - Begin with a model with no predictors
  - For each possible predictor, create a model with that predictor added.
  - Pick the predictor model where the added predictor had the smallest significant p-value.
  - Repeat the previous 2 steps until no added predictors have significant p-values.
- There is no guarantee that forward-selection and backward-elimination will reach the same model.
- Usually, we just use one selection method. Since backward-elimination requires fewer steps, it is often used.