Estimating Impacts of Policy Changes, Part 4

Will Doyle

2022-07-12

Once we have a model that can predict an outcome, we can also run some simulations using our existing data to understand what would happen if we implemented different policies. In this class we're going to go over how to generate predictions using hypothetical data at different levels of complexity. This is helpful to understand the implications of different models. It's particularly helpful in the context of logistic regression, as the coefficients from a logistic regression are difficult to understand on their own. Once we feel like we have a handle on the relationship between the variables and the outcome, we can try changing policies to see what the new predictions might look like.

We'll start by loading in our standard libraries, plus the `modelr` library, and getting the data set up.

```
library(modelr)
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'
```

```
library(tidyverse)
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'hms'
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
library(tidymodels)
```

```
## Warning: package 'recipes' was built under R version 4.1.2
```

```
ad<-read_rds("admit_data.rds")
```

Data Wrangling

```
ad<-ad%>%
  mutate(yield_f=as_factor(ifelse(yield==1,"Yes","No")))%>%
  mutate(yield_f=relevel(yield_f,ref="No"))%>%
    mutate(sat=sat/100,
         income=income/1000,
         distance=distance/1000,
         net_price=net_price/1000)
```

Settting the Model

Today I'm going to use a more full specified logit model, without feature selection. We'll include most of the variables in the dataset, then fit the model and take a look at the coefficients.

```
logit_mod<-logistic_reg()%>%
  set_engine("glm")%>%
  set_mode("classification")
```

Formula and Recipe

```
admit_formula <- as.formula(
                  "yield_f~
                          legacy+
                          visit+
                          registered+
                          sent_scores+
                          sat+
                          income+
                          gpa+
                          distance+
                          net_price")

admit_recipe<-recipe(admit_formula,ad)%>%
  step_log(distance)
```

Workflow

```
ad_wf<-workflow()%>%
  add_model(logit_mod)%>%
  add_recipe(admit_recipe)
```

Fit the Model

```
ad_result<-ad_wf%>%
  fit(ad)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
tidy(ad_result)
```

```
## # A tibble: 10 x 5
##    term        estimate std.error statistic  p.value
##    <chr>          <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept) -10.5      1.55       -6.75  1.52e-11
##  2 legacy        0.514    0.154       3.34  8.43e- 4
##  3 visit         0.295    0.137       2.16  3.08e- 2
##  4 registered    0.469    0.132       3.54  3.98e- 4
##  5 sent_scores   0.758    0.180       4.20  2.65e- 5
##  6 sat          -0.0375   0.127      -0.296 7.67e- 1
##  7 income        0.0523   0.00315    16.6   4.35e-62
##  8 gpa           1.81     0.340       5.33  9.91e- 8
##  9 distance     -0.405    0.0688     -5.88  4.07e- 9
## 10 net_price    -0.0559   0.00764    -7.31  2.67e-13
```

Current Institutional Policies

(This is a recap from lecture 1)

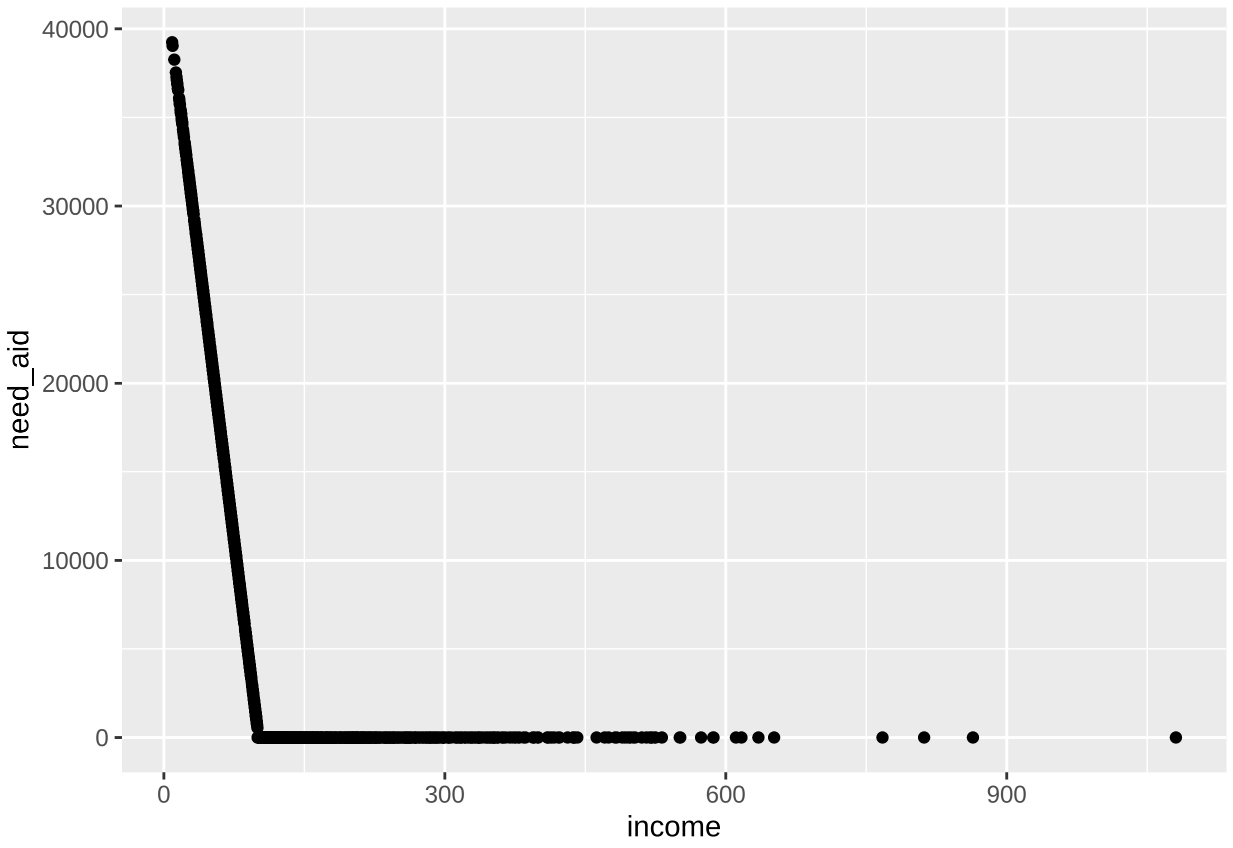
Essentially every private college engages heavily in tuition discounting. This has two basic forms: need-based aid and merit-based aid. Need-based aid is provided on the basis of income, typically with some kind of income cap. Merit-based aid is based on demonstrated academic qualifications, again usually with some kind of minimum. Here's this institution's current policies.

The institution is currently awarding need-based aid for families making less than $100,0000 on the following formula:

$need_aid = 500 + (income/1000 - 100) - 425$

Translated, this means for every $1,000 the family makes less than $100,000 the student receives an additional 425 dollars. So for a family making $50,000, need-based aid will be $500 + (50,000/1000 - 100) * -425 = 500 + (-50 * -425)$=$21,750. Need based aid is capped at total tuition.

```
ad%>%
  ggplot(aes(x=income,y=need_aid))+
  geom_point()
```

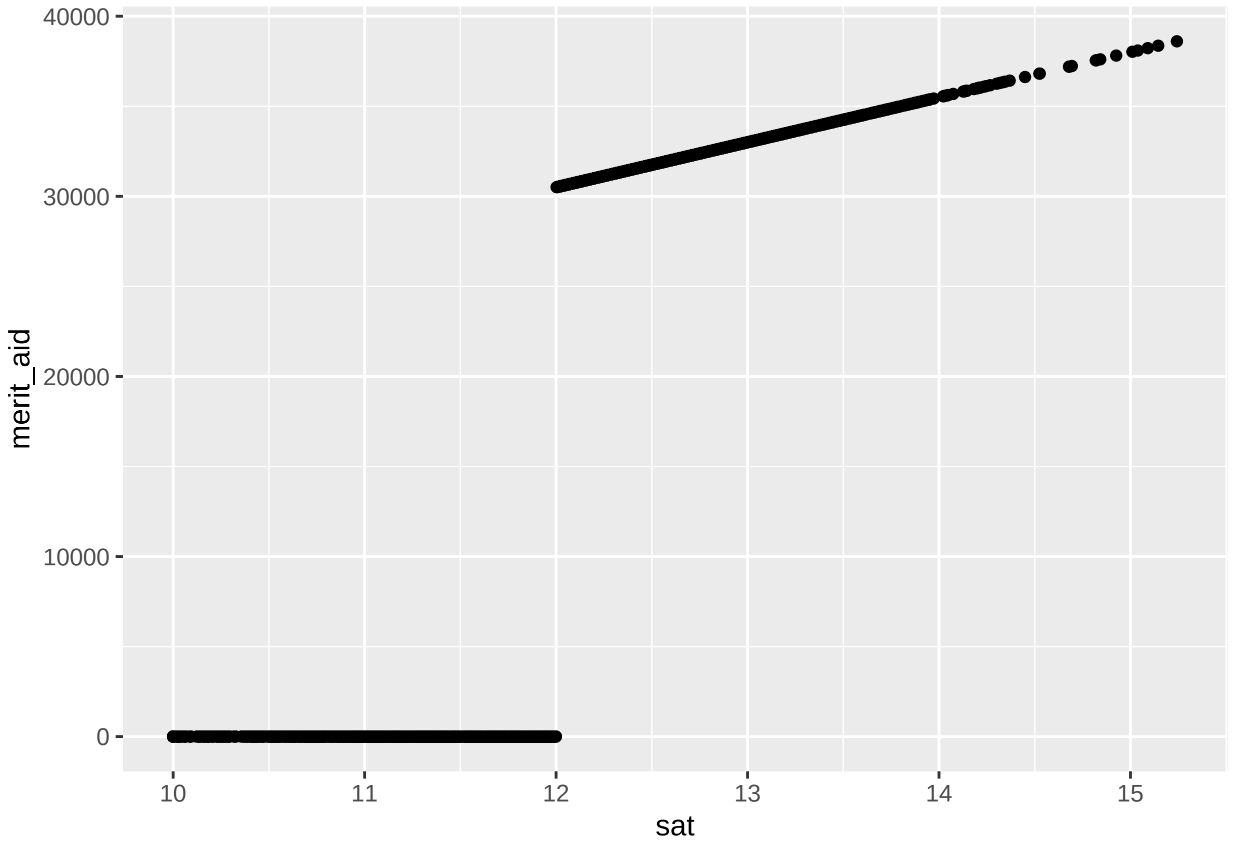


The institution is currently awarding merit-based aid for students with SAT scores above 1250 on the following formula:

$merit_aid = 500 + (sat/10250)$

Translated, this means that for every 10 points in SAT scores above 1250, the student will receive an additional $250. So for a student with 1400 SAT, merit based aid will be : $500 + (1400/10 * 250) = 500 + 140 * 250$

```
ad%>%
  ggplot(aes(x=sat,y=merit_aid))+
  geom_point()
```



Probability for a Specific Case

The first thing we'll do is to generate a predictede probability of yield for a particular type of case. Let's take a look at the predicted probability of enrolling for an individual who:

- Is not a legacy (legacy=0)
- Visited Campus (visit=1)
- Registered on the college website (register=1)
- Did not send scores in advance of applying (sent_scores=1)
- Has an SAT score of 1400 (sat=14)
- Has a family income of $95,000 (income=95)
- Has a GPA of 3.9 (gpa=3.9)

- Lives 100 miles from campus (distance=.1)
- Net Price of 6,875

According to our policy, someone with this set of characteristics would receive

$need_aid = 500 + (income/1000 - 100) - 425$= 2625 in need-based aid, and

$merit_aid = 500 + (sat/10250)$= 35,500 in merit based aid, so their net price would be:

45000-2625-35,500= 6,875

First we'll access the model results using `extract_fit_parsnip`

```
ad_fit<-ad_result%>%extract_fit_parsnip(ad_result)
```

Then we'll use the `data_grid` command. This command allows us to specify values of the covariates in a model so that we can then use these to get a predicted probability from the model.

```
hypo_data <- ad %>%
  data_grid(legacy=0,
            visit=1,
            registered=1,
            sent_scores=1,
            sat=14,
            income=95,
            gpa=3.9,
            distance=.1,
            net_price=6.875
            )
```

```
ad_result%>%
  predict(hypo_data,type="prob")%>%
  bind_cols(hypo_data)
```

```
## # A tibble: 1 x 11
##   .pred_No .pred_Yes legacy visit registered sent_scores   sat income   gpa
##      <dbl>     <dbl>  <dbl> <dbl>      <dbl>       <dbl> <dbl>  <dbl> <dbl>
## 1   0.0427     0.957      0     1          1           1    14     95   3.9
## # ... with 2 more variables: distance <dbl>, net_price <dbl>
```

Quick Exercise: What's the probability that the same student would attend if we increased their net price by 10k?

```
# INSERT CODE HERE
```

Change in Probability for Two Cases

With the data_grid command, we can ask for results back for many combinations of values. Let's check to see what happens in the above case if we compare individuals who did and did not send their scores.

```
hypo_data <- ad %>%
  data_grid(legacy=0,
            visit=1,
            registered=1,
            sent_scores=c(0,1),
            sat=14,
            income=95,
            gpa=3.9,
            distance=.1,
            net_price=6.875
            )
```

```
ad_result%>%
  predict(hypo_data,type="prob")%>%
  bind_cols(hypo_data)
```

```
## # A tibble: 2 x 11
##   .pred_No .pred_Yes legacy visit registered sent_scores   sat income   gpa
##      <dbl>     <dbl>  <dbl> <dbl>      <dbl>       <dbl> <dbl>  <dbl> <dbl>
## 1   0.0869     0.913      0     1           1            0    14      95   3.9
## 2   0.0427     0.957      0     1           1            1    14      95   3.9
## # ... with 2 more variables: distance <dbl>, net_price <dbl>
```

Quick Exercise: What's the difference in probability for students who did and didn't visit campus

```
# INSERT CODE HERE
```

Using Default Values

One really powerful aspect of `data_grid` is that we don't have to specify the value of every variable. Instead, we can use supply the model to data_grid and it will use the default values. Let's say we just want a prediction for someone with mean or modal value for all of the characteristics. We can use data_grid with the model argument to get this:

```
hypo_data <- ad %>%
  data_grid(.model=ad_fit$fit)
```

```
ad_result%>%
  predict(hypo_data,type="prob")%>%
  bind_cols(hypo_data)
```

```
## # A tibble: 1 x 11
##   .pred_No .pred_Yes legacy visit registered sent_scores   sat income   gpa
##      <dbl>     <dbl>  <dbl> <dbl>      <dbl>       <dbl> <dbl>  <dbl> <dbl>
## 1    0.145     0.855      0     0          1           0  12.0   99.7  3.79
## # ... with 2 more variables: distance <dbl>, net_price <dbl>
```

Quick Exercise: generate probabilities for individuals who are both at the mean or mode of all variables, but have a gpa of 3.5 and 3.9.

```
hypo_data <- ad %>%
  data_grid(.model=ad_fit$fit,
            gpa=c(3.5,3.9))

ad_result%>%
  predict(hypo_data,type="prob")%>%
  bind_cols(hypo_data)
```

```
## # A tibble: 2 x 11
##   .pred_No .pred_Yes   gpa legacy visit registered sent_scores   sat income
##      <dbl>     <dbl> <dbl>  <dbl> <dbl>      <dbl>       <dbl> <dbl>  <dbl>
## 1    0.223     0.777   3.5      0     0          1           0  12.0   99.7
## 2    0.122     0.878   3.9      0     0          1           0  12.0   99.7
## # ... with 2 more variables: distance <dbl>, net_price <dbl>
```

Probabilities Across a Range of Cases

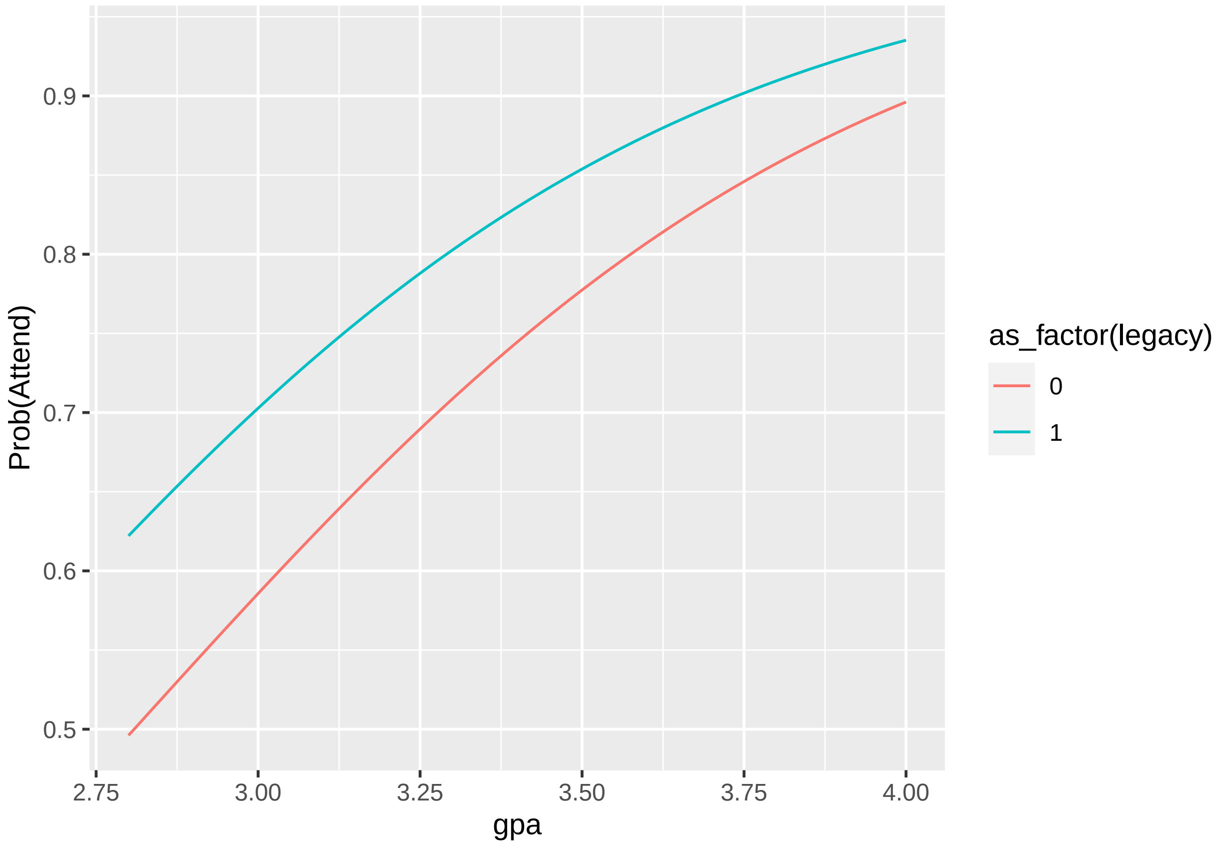
The other thing we can do is to generate probabilities for a range of a continuous variable. The `seq_range` variable allows us to go from the minimum to the maximum of a given variable in a specified number of steps. Below I go from the minimum gpa to the maximum gpa in 100 steps, for both legacy and non-legacy students, with all other values held at their mean or mode.

```
hypo_data <- ad %>%
  data_grid(gpa = seq_range(gpa, n = 100),
            legacy=c(0,1),
            .model=ad_fit$fit)
```

We can then plot that data as follows:

```
plot_data<-ad_result%>%
  predict(hypo_data,type="prob")%>%
  bind_cols(hypo_data)

plot_data%>%
ggplot(aes(x=gpa,y=.pred_Yes,color=as_factor(legacy)))+
  geom_line()+
  ylab("Prob(Attend)")
```



Quick Exercise: Plot the impact of changing distance across its range for those who have and haven't visited campus

```
# INSERT CODE HERE
```

The above tools can allow us to think through the substantive importance of changes in the different variables. This is an important step in these kinds of models, where the coefficients aren't directly interpretable.

Changing policy

AS we think about changing policy for the campus, we need to answer a series of questions about the key characteristics of the campus right now. Remember that for your assignment we want you to do the the four following things:

The college president and the board of trustees have two strategic goals:

1. Increase the average SAT score to 1300
2. Admit at least 200 more students with incomes less than $50,000
3. Don't allow tuition revenues from first-year students to drop to less than $30 million
4. Keep the number of enrolled students between 1,450 and 1,550

So, what's the average SAT score?

```
ad%>%
  filter(yield==1)%>%
  summarize(mean(sat))
```

```
##   mean(sat)
## 1  12.25941
```

Second, how many currently enrolled students come from families that make less than $50,000 a year?

```
ad%>%
  filter(yield==1,income<50)%>%
  count()
```

```
##    n
## 1 77
```

Someone alert Raj Chetty!

Next how much does the campus collect from first-year students in tuition revenue?

```
ad%>%
  filter(yield==1)%>%
  summarize(dollar(sum(net_price)))
```

```
##   dollar(sum(net_price))
## 1             $30,674.15
```

And how many students currently yield?

```
ad%>%
  filter(yield==1)%>%
  count()
```

```
##      n
## 1 1466
```

What we want to do now is to think about how changing policies might affect all four of these summary measures. Let's think about changing our financial aid policy for students who sent scores.

```
ad_result%>%
  predict(ad)%>%
  bind_cols(ad)%>%
  group_by(sent_scores,.pred_class)%>%
  count()
```

```
## # A tibble: 4 x 3
## # Groups:   sent_scores, .pred_class [4]
##   sent_scores .pred_class     n
##         <dbl> <fct>       <int>
## 1           0 No            621
## 2           0 Yes          1099
## 3           1 No             74
## 4           1 Yes           356
```

The model currently predicts that 356 of the 430 students who sent scores will enroll. What happens if we increase their net price by 5,000? I'm going to create a new version of our dataset, with net price increased by 5,000 (5) for everyone who sent scores.

```
ad_np<-ad%>%
  mutate(net_price=ifelse(sent_scores==1,
                                net_price+5,
                                net_price))
```

Then I can generate predictions from the dataset, and create a new variable for .pred_Class which will classify our variable. Please not that this uses a threshold of .5 by default.

```
ad_np<-ad_result%>%
  predict(ad_np)%>%
  bind_cols(ad_np)
```

How many students who sent scores are now predicted to yield?

```
ad_np%>%
  group_by(sent_scores,.pred_class)%>%
  count()
```

```
## # A tibble: 4 x 3
## # Groups:   sent_scores, .pred_class [4]
##   sent_scores .pred_class     n
##         <dbl> <fct>       <int>
## 1           0 No            621
## 2           0 Yes          1099
## 3           1 No             96
## 4           1 Yes           334
```

So, we lost about 22 students by increasing the net price. On the other hand, we charged more to the students who did attend.

```
ad_np%>%
  filter(.pred_class=="Yes")%>%
  summarize(dollar(sum(net_price)))
```

```
## # A tibble: 1 x 1
##   `dollar(sum(net_price))`
##   <chr>
## 1 $31,137.57
```

Total revenues went up!

And what happened to overall enrollment?

```
ad_np%>%
  group_by(.pred_class)%>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   .pred_class [2]
##   .pred_class     n
##   <fct>       <int>
## 1 No            717
## 2 Yes          1433
```

So, this policy decreased enrollment slightly, but raised 500,000 dollars. Worth it?

Quick Exercise: Decrease prices for everyone who lives more than 1500 miles away by 10,000 and summarize the impacts on the campus.

```
# INSERT CODE HERE
```