Building a Set of Candidate Models

2020-02-07

Preparation

In this set of notes, you will learn about using information criteria to select a model from a set of candidate models. To do so, we will use the *usnews.csv* dataset (see the data codebook) to examine the factors that underlie the ratings our academic peers give to graduate programs of education.

```
# Load libraries
library(broom)
library(corrr)
library(educate) #Need version 0.1.0.1
library(patchwork)
library(skimr)
library(tidyverse)

# Import data
usnews = read_csv("~/Documents/github/epsy-8252/data/usnews.csv")

# View data
head(usnews)
```

```
# A tibble: 6 x 13
   rank school score  peer expert_score gre_verbal gre_quant doc_accept
  <dbl> <chr>  <dbl> <dbl>        <dbl>      <dbl>     <dbl>      <dbl>
1     1 Harva~   100   4.4          4.6        163       159        4.5
2     2 Stanf~    99   4.6          4.8        162       160        6.1
3     3 Unive~    96   4.2          4.3        156       152       29.1
4     3 Unive~    96   4.1          4.5        163       157        5
5     3 Unive~    96   4.3          4.5        155       153       26.1
6     6 Johns~    95   4.1          4.1        164       162       27.4
# ... with 5 more variables: phd_student_faculty_ratio <dbl>,
#   phd_granted_per_faculty <dbl>, funded_research <dbl>,
#   funded_research_per_faculty <dbl>, enroll <dbl>
```

To gather the peer assessment data, *U.S. News* asked deans, program directors and senior faculty to judge the academic quality of programs in their field on a scale of 1 (marginal) to 5 (outstanding). Based on the substantive literature we have three **scientific working hypotheses** about how academics perceive and, ultimatley rate, graduate programs:

- **H1:** Student-related factors drive the perceived academic quality of graduate programs in education.
- **H2:** Faculty-related factors drive the perceived academic quality of graduate programs in education.
- **H3:** Institution-related factors drive the perceived academic quality of graduate programs in education.

We need to translate these working hypotheses into statistical models that we can then fit to a set of data. The models are only proxies for the working hypotheses. However, that being said, the validity of using the models as proxies is dependent on whether we have measured well, whether the translation makes substantive sense given the literature base, etc. Here is how we are measuring the different attributes:

- The student-related factors we will use are GRE scores.
- The faculty-related factors we will use are funded research (per faculty member) and the number of Ph.D. graduates (per faculty member).
- The institution-related factors we will use are the acceptance rate of Ph.D. students, the Ph.D. student-to-faculty ratio, and the size of the program.

Model-Building

Before we begin the exploratory analysis associated with model-building, it is worth noting that there are missing data in the dataset. Here we use the `skim()` function from the **skimr** package to summarize the data (see documentation here). (Note that if you are trying to use this function in a RMarkdown document, you may get an error. Instead, you may need to use the `skim_without_charts()` function and immediately pipe into the `kable()` function.) Below I only report part of the `skim()` output.

```
usnews %>%
  skim()
```

```
Number of rows             129
Number of columns          13
_______________________

Variable type: character
  skim_variable n_missing complete_rate   min   max empty n_unique whitespace
1 school                0             1    13    48     0      129          0
_______________________

Variable type: numeric
   skim_variable              n_missing complete_rate   mean     sd    p0    p25   p50    p75   p100
 1 rank                               0             1   63.2   36.0     1    32    62     93    120
 2 score                              0             1   55.3   17.1    38    42    51     63    100
 3 peer                              0            1     3.29   0.484   2.5   2.9   3.2   3.6    4.6
 4 expert_score                      0            1     3.64   0.475   2.4   3.3   3.6   4      4.8
 5 gre_verbal                         6         0.953 155.      3.73  148   152   154    156    166
 6 gre_quant                          6         0.953 151.      4.47  142   148   150    153    167
 7 doc_accept                         0             1   41.4   21.2     4.5  25.8   39.8   54.1  100
 8 phd_student_faculty_ratio          0            1    2.91   1.74   0     1.7   2.6    3.7   11.7
 9 phd_granted_per_faculty            1          0.992   0.745  0.789  0     0.4   0.6    0.9    8.4
10 funded_research                   2          0.984  13.8    15.0    0.1   3.75  8     18.1   78.6
11 funded_research_per_faculty        2          0.984 226.    226.      2.9  78.4 163.   267.  1239.
12 enroll                             0             1  954.    658.     29   556   835   1281   4892
```

This is a problem when we are comparing models that use different variables as the observations used to fit one model will be different than the observations used to fit another model. Since we are going to be using a likelihood-based method of comparing the models, this is problematic. Remember, likelihood-based methods find us the most likely model given a set of data. If the datasets used are different, we won't know whether a model with a higher likelihood is truly more likely or is more likely because of the dataset used.

To alleviate this problem, we will eliminate any observations (rows in the dataset) that have missing data. This is called *listwise* or *row-wise* deletion. Any analyses performed on the remaining data constitute a *complete-cases* analysis, since these cases have no missing data. To select the complete cases, we will use the `drop_na()` function from the **tidyverse** package.

```r
# Drop rows with missing data
educ = usnews %>%
  drop_na()

# Check resulting data
educ %>%
  skim()
```

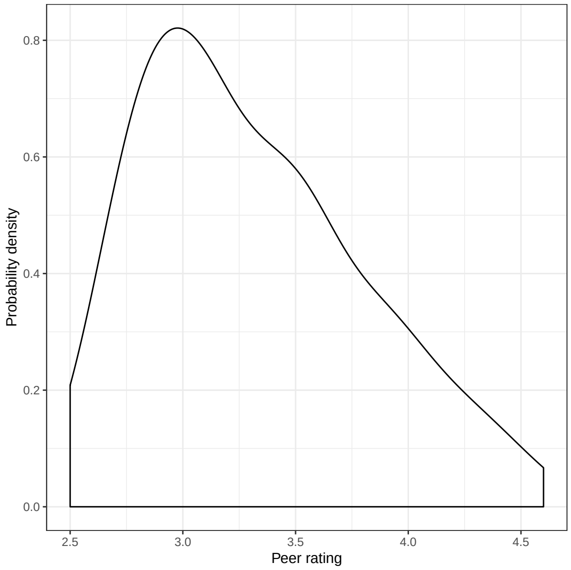
```
Number of rows             122
Number of columns          13
________________________

Variable type: character
  skim_variable n_missing complete_rate   min   max empty n_unique whitespace
1 school                0             1    13    48     0      122          0

________________________

Variable type: numeric
   skim_variable             n_missing complete_rate    mean      sd    p0    p25     p50     p75   p100
 1 rank                              0             1   60.5    35.1      1    31.2    59.5     89     120
 2 score                             0             1   56.2    17.1     38    43      51.5     63.8   100
 3 peer                              0             1    3.31    0.489   2.5   2.9     3.2      3.6    4.6
 4 expert_score                      0             1    3.66    0.479   2.4   3.3     3.6      4      4.8
 5 gre_verbal                        0             1  155.      3.71  148   152.    154      156     166
 6 gre_quant                         0             1  151.      4.42  142   148     150      153     167
 7 doc_accept                        0             1   40.1    20.2     4.5  25.5    38.6     51.6    92.7
 8 phd_student_faculty_ratio         0             1    2.94    1.75    0     1.7     2.7      3.77   11.7
 9 phd_granted_per_faculty           0             1    0.758   0.806   0     0.4     0.650    0.9     8.4
10 funded_research                   0             1   14.3    15.2     0.1   3.85    8.5     18.8    78.6
11 funded_research_per_faculty       0             1  229.    230.      2.9  77.8   161.     283.   1239.
12 enroll                            0             1  970.    665.     29   562.    842.    1312.   4892
```

After selecting the complete-cases, the usable, analytic sample size is $n = 122$. Seven observations (5.4%) were eliminated from the original sample because of missing data.

Exploration of the Outcome

The outcome variable we will use in each of the models is peer rating (`peer`). This variable can theoretically vary from 1 to 5, but in our sample has only ranges from 2.5 to 4.6. The density plot indicates that this variable is right-skewed. This may foreshadow problems meeting the normality assumption and we subsequently may consider log-transforming this variable.

Figure 1
Density Plot of the Outcome Variable



Below we show scatterplots of the outcome (peer ratings) versus each of the predictors we are considering in the three scientific models. (Syntax is not shown.) The loess smoother is also displayed in each plot.

Figure 2
Scatterplots of Peer Ratings versus the Student-Related Factors

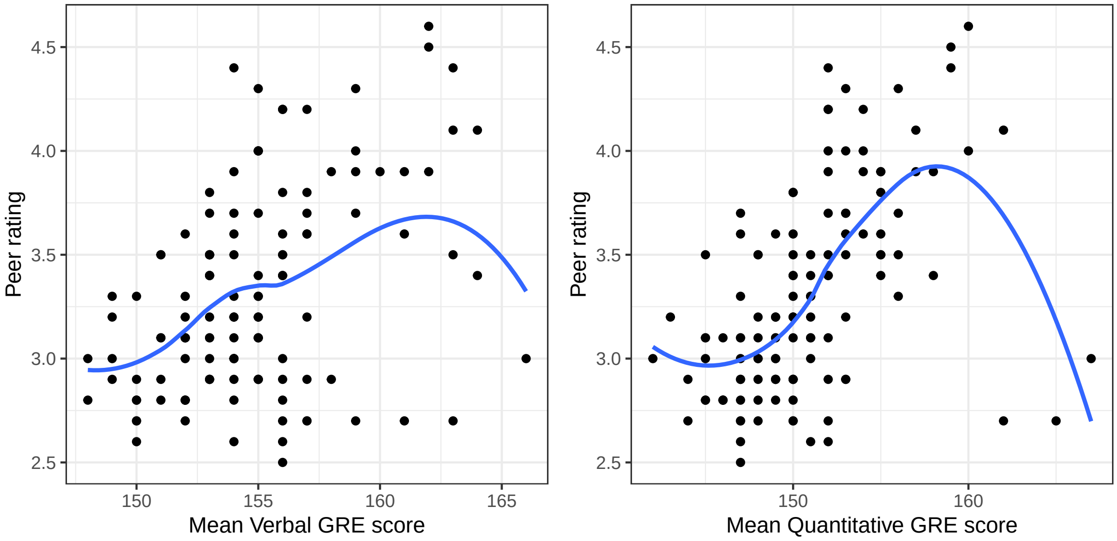


Figure 3

Scatterplots of Peer Ratings versus the Faculty-Related Factors

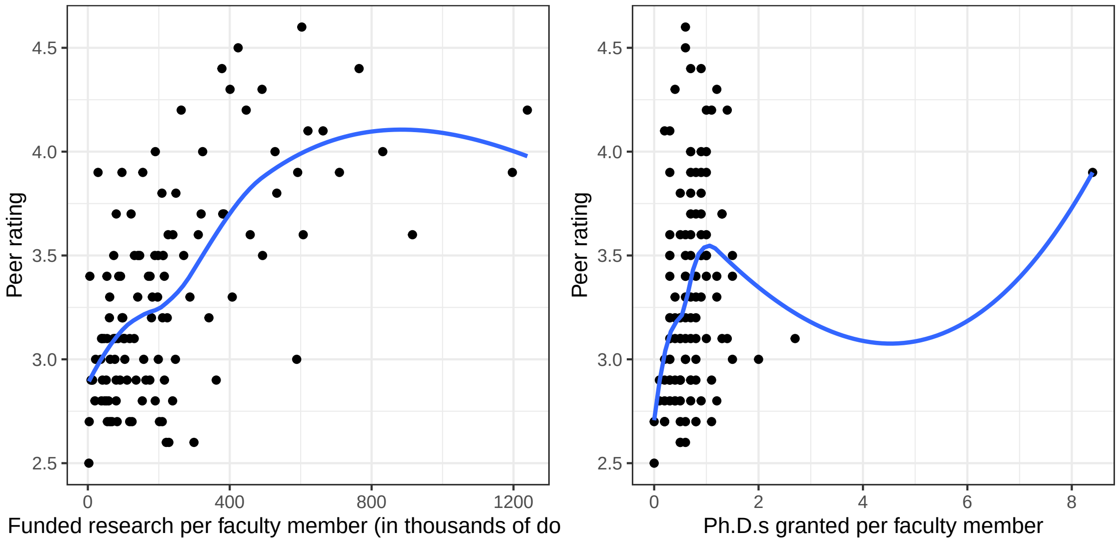
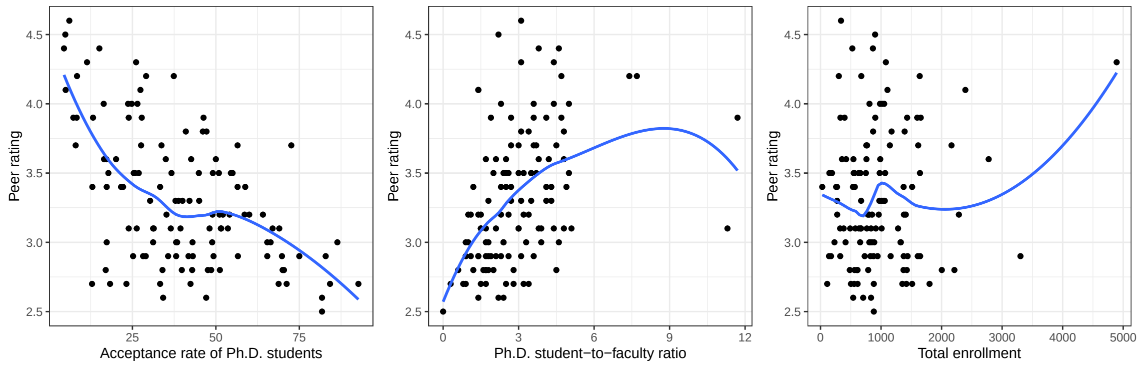


Figure 4

Scatterplots of Peer Ratings versus the Institution-Related Factors



Almost all of these plots show curvilinear patterns, some of which can be alleviated by log-transforming the outcome. Remember, log-transforming the outcome will also help with violations of homoskedasticity. Since we want to be able to compare the models at the end of the analysis, we NEED to use the same outcome in each of the models. Given the initial right-skewed nature of the outcome distribution and the evidence from the scatterplots, we will log-transform peer ratings and use that outcome in each model we fit.

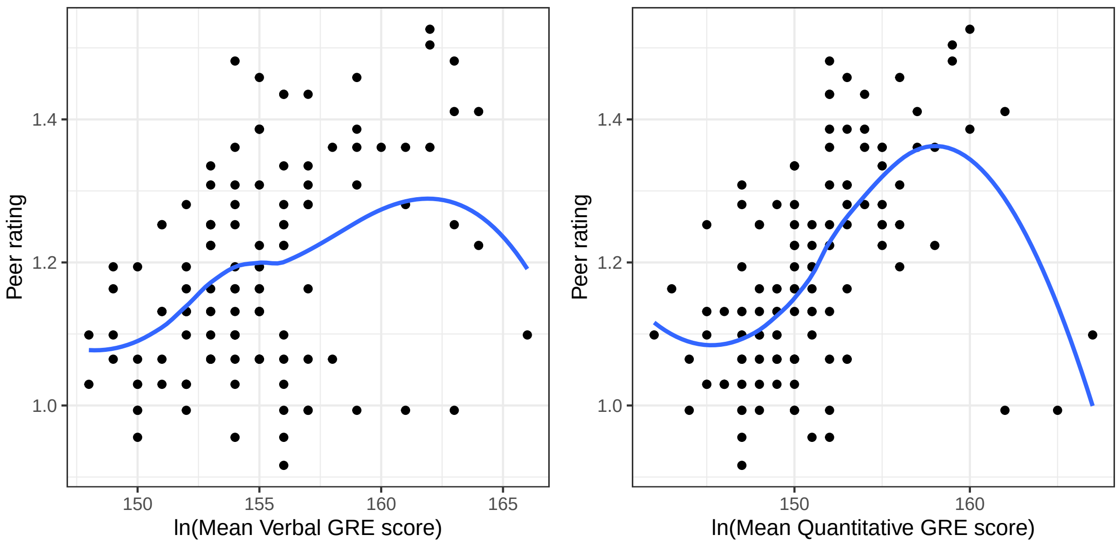
```
# Create log-transformed peer ratings
educ = educ %>%
  mutate(
    Lpeer = log(peer)
    )
```

Building the Student-Related Factors Model

To determine which of the student-related factors to include in the model, we will examine the scatterplots of each predictor against the log-transformed peer ratings and also examine the correlation matrix of the outcome and student-related predictors.

Figure 5

Scatterplots of the Log-Transformed Peer Ratings versus the Student-Related Factors



Note. The loess smoother is also displayed in each plot.

```
educ %>%
  select(Lpeer, gre_verbal, gre_quant) %>%
  correlate()
```

```
# A tibble: 3 x 4
  rowname     Lpeer gre_verbal gre_quant
  <chr>       <dbl>      <dbl>     <dbl>
1 Lpeer      NA          0.408     0.478
2 gre_verbal  0.408     NA         0.808
3 gre_quant   0.478      0.808    NA
```

Not surprisingly, the mean GRE verbal and GRE quantitative scores are highly correlated. Since including highly correlated predictors in a model can lead to unstable estimates, we will drop one of the predictors from the model. Empirically, the quantitative GRE scores seem more highly correlated with the outcome, so we will drop the GRE verbal scores from the model.

Focusing on the scatterplot of the GRE quantitative scores, the relationship with the log-transformed peer ratings looks curvilinear (non-monotonic). The empirical relationship seems to change direction twice, indicating that log-transformed peer ratings may be a cubic-function of the quantitative GRE scores.

```
# Fit cubic model
lm.1 = lm(Lpeer ~ 1 + gre_quant + I(gre_quant^2) + I(gre_quant^3), data = educ)
```

```r
# Model-level output
glance(lm.1)
```

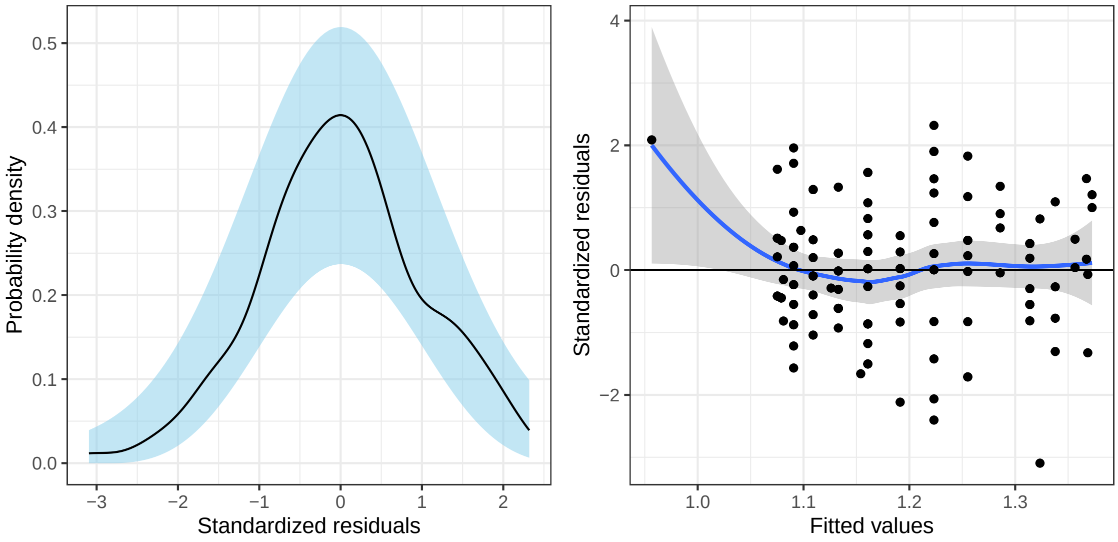
```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.409         0.394 0.112      27.2 1.96e-13     4   95.9 -182. -168.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```r
# Coefficient-level output
tidy(lm.1)
```

```
# A tibble: 4 x 5
  term             estimate   std.error statistic   p.value
  <chr>               <dbl>       <dbl>     <dbl>     <dbl>
1 (Intercept)    780.       176.             4.43 0.0000210
2 gre_quant      -15.4        3.43          -4.49 0.0000165
3 I(gre_quant^2)   0.102      0.0223         4.55 0.0000129
4 I(gre_quant^3)  -0.000223   0.0000483     -4.61 0.0000102
```

```r
# Obtain residuals
out_1 = augment(lm.1)

# Examine residuals
p1 = ggplot(data = out_1, aes(x = .std.resid)) +
  stat_density_confidence(model = "normal") +
  stat_density(geom = "line") +
  theme_bw() +
  xlab("Standardized residuals") +
  ylab("Probability density")

p2 = ggplot(data = out_1, aes(x = .fitted, y = .std.resid)) +
  geom_smooth(se = TRUE) +
  geom_hline(yintercept = 0) +
  geom_point() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")

p1 + p2
```

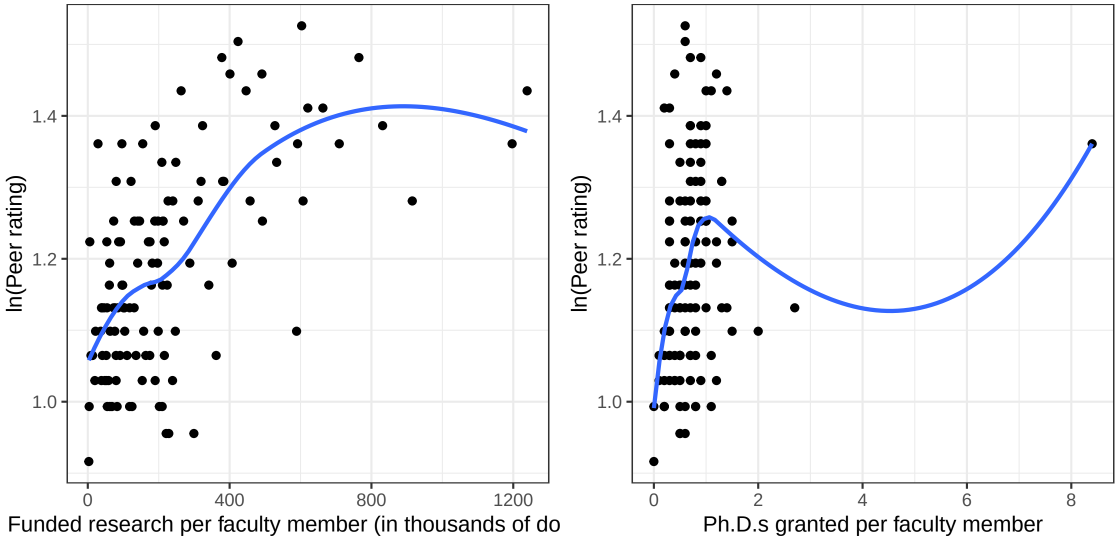
Figure 6
Residual Plots for the Student-Related Factors Fitted Model



Building the Faculty-Related Factors Model

To determine which of the faculty-related factors to include in the model, we will examine the scatterplots of each predictor against the log-transformed peer ratings and also examine the correlation matrix of the outcome and faculty-related predictors.

Figure 7
Scatterplots of the Log-Transformed Peer Ratings versus the Faculty-Related Factors



Note. The loess smoother is also displayed in each plot.

```
educ %>%
  select(Lpeer, funded_research_per_faculty, phd_granted_per_faculty) %>%
  correlate()
```

```
# A tibble: 3 x 4
  rowname                    Lpeer funded_research_per_facu~ phd_granted_per_facu~
  <chr>                      <dbl>                     <dbl>                 <dbl>
1 Lpeer                     NA                         0.597                 0.217
2 funded_research_per_fa~  0.597                     NA                     0.403
3 phd_granted_per_faculty  0.217                     0.403                NA
```

The two predictors are moderately correlated with each other and both are correlated with the outcome. The scatterplot of peer ratings versus funded research suggest a monotonic curvilinear relationship. The Rule of the Bulge indicates that log-transforming the predictor may help linearize this relationship. The scatterplot of peer ratings versus number of Ph.D.s granted suggests that the distribution of the predictor is right-skewed with a potential outlying observation. This relationship may also benefit from log-transforming the predictor.

Before log-transforming these predictors, it is a good idea to check the distributions for zero or negative values.

```
educ %>%
  select(funded_research_per_faculty, phd_granted_per_faculty) %>%
  skim()
```

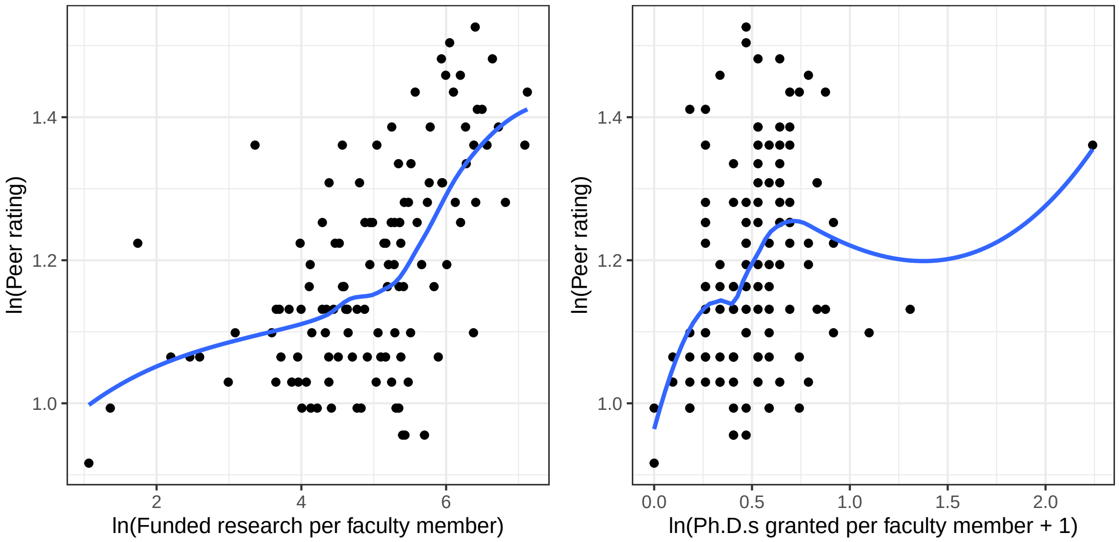
```
  skim_variable              n_missing complete_rate    mean      sd    p0   p25     p50   p75   p100
1 funded_research_per_faculty         0             1 229.    230.      2.9  77.8 161.    283.  1239.
2 phd_granted_per_faculty            0             1   0.758   0.806   0     0.4   0.650   0.9     8.4
```

The summary values for the `phd_granted_per_faculty` predictor indicates that there are some schools that the minimum value for this predictor is 0 (`p0`). Before we transform using a log-transformation, we need to make it so the smallest value in the predictor is 1, since the log of 0 (and any negative values) is undefined. To do this we will add some number (in our case 1) to each value for `phd_granted_per_faculty` prior to taking the log.

```
#Create log of the faculty-related predictors
educ = educ %>%
  mutate(
    Lfunded_research_per_faculty = log(funded_research_per_faculty),
    Lphd_granted_per_faculty = log(phd_granted_per_faculty + 1)
  )
```

Figure 8

Scatterplots of the Log-Transformed Peer Ratings versus the Log-Transformed Faculty-Related Factors



Although this helped, it did not “cure” the nonlinearity. We might want to further include a quadratic term for each of the predictors. To evaluate this, we will fit the model that includes the linear and quadratic log-transformed predictors and examine the coefficient-level output and residuals.

```
# Fit model
lm.2 = lm(Lpeer ~ 1 + Lfunded_research_per_faculty + I(Lfunded_research_per_faculty^2) + Lphd_granted_per_faculty

# Model-level output
glance(lm.2)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.431         0.412 0.110      22.2 1.22e-13     5   98.2 -184. -168.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
# Coefficient-level output
tidy(lm.2)
```

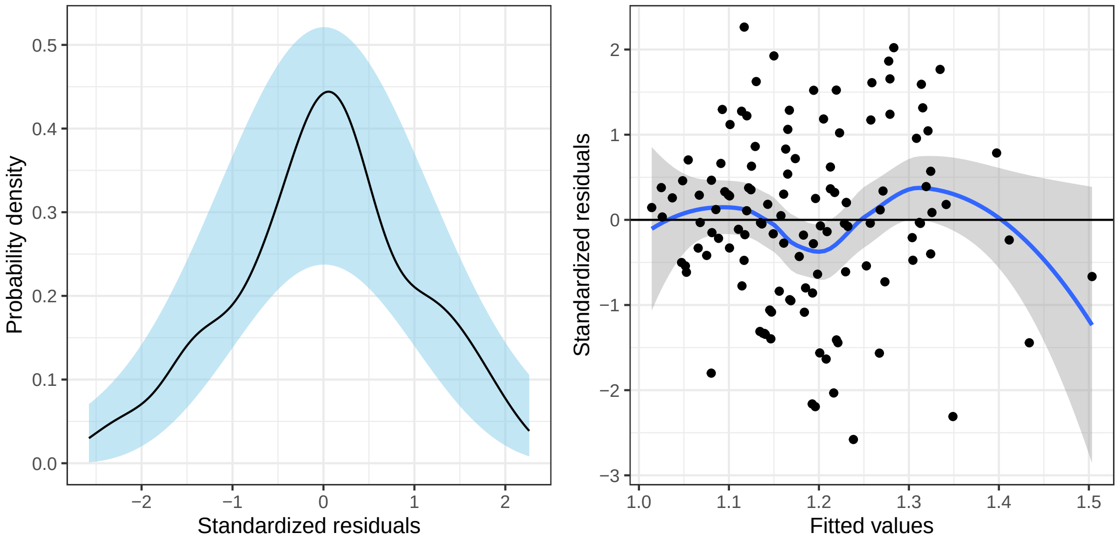
```
# A tibble: 5 x 5
  term                              estimate std.error statistic  p.value
  <chr>                                <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)                         1.21     0.106       11.5  6.76e-21
2 Lfunded_research_per_faculty       -0.151    0.0509      -2.96 3.70e- 3
3 I(Lfunded_research_per_faculty^2)   0.0238   0.00547      4.35 2.87e- 5
4 Lphd_granted_per_faculty            0.300    0.0948       3.16 1.99e- 3
5 I(Lphd_granted_per_faculty^2)      -0.139    0.0503      -2.77 6.58e- 3
```

```
# Obtain residuals
out_2 = augment(lm.2)
```

```
# Examine residuals
p1 = ggplot(data = out_2, aes(x = .std.resid)) +
  stat_density_confidence(model = "normal") +
  stat_density(geom = "line") +
  theme_bw() +
  xlab("Standardized residuals") +
  ylab("Probability density")

p2 = ggplot(data = out_2, aes(x = .fitted, y = .std.resid)) +
  geom_smooth(se = TRUE) +
  geom_hline(yintercept = 0) +
  geom_point() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")

p1 + p2
```

Figure 9

Residual Plots for the Faculty-Related Factors Fitted Model

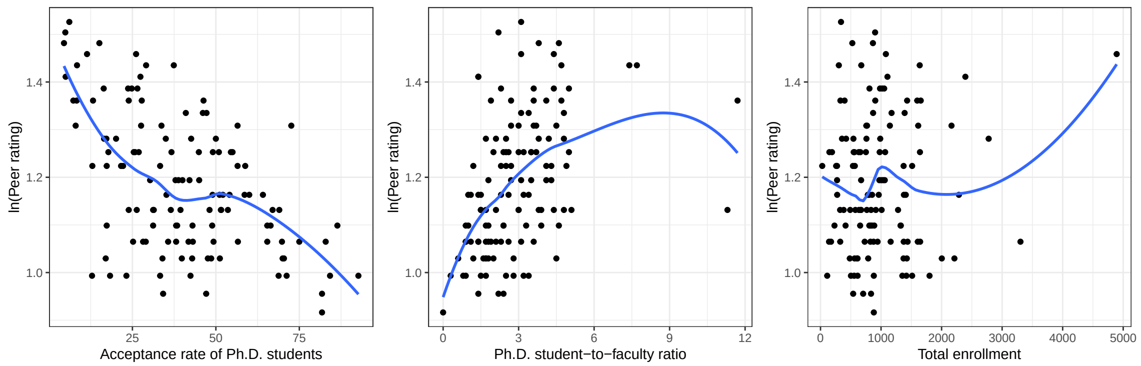


Building the Institution-Related Factors Model

To determine which of the institution-related factors to include in the model, we will examine the scatterplots of each predictor against the log-transformed peer ratings and also examine the correlation matrix of the outcome and institution-related predictors.

Figure 10

Scatterplots of the Log-Transformed Peer Ratings versus the Institution-Related Factors



Note. The loess smoother is also displayed in each plot.

```
educ %>%
  select(Lpeer, doc_accept, phd_student_faculty_ratio, enroll) %>%
  correlate()
```

```
# A tibble: 4 x 5
  rowname                     Lpeer doc_accept phd_student_faculty_rat~   enroll
  <chr>                       <dbl>      <dbl>                    <dbl>    <dbl>
1 Lpeer                     NA          -0.534                  0.423    0.0964
2 doc_accept                -0.534      NA                     -0.235   -0.0256
3 phd_student_faculty_ratio  0.423      -0.235                 NA        0.00450
4 enroll                     0.0964     -0.0256                 0.00450 NA
```

The three predictors are mostly uncorrelated with each other and all are correlated with the outcome, albeit enrollment is weakly correlated with peer ratings. Two of the three scatterplots suggest curvilinear relationships although with different functional forms—Ph.D. student-to-faculty ratio and total enrollment. The Rule of the Bulge indicates that log-transforming the Ph.D. student-to-faculty ratio predictor, and including quadratic may help linearize this relationship. The scatterplot of peer ratings versus total enrollment suggests that the distribution of the predictor is right-skewed with a potential outlying observations. This relationship may also benefit from log-transforming the predictor. Lastly, it is unclear whether any additional transformation or polynomial terms are necessary for modeling the relationship with doctoral acceptance rate; to double-check this we will also log-transform the total enrollment predictor.

As before, prior to log-transforming any predictors, it is a good idea to check the distributions for zero or negative values.

```
educ %>%
  select(doc_accept, phd_student_faculty_ratio, enroll) %>%
  skim()
```

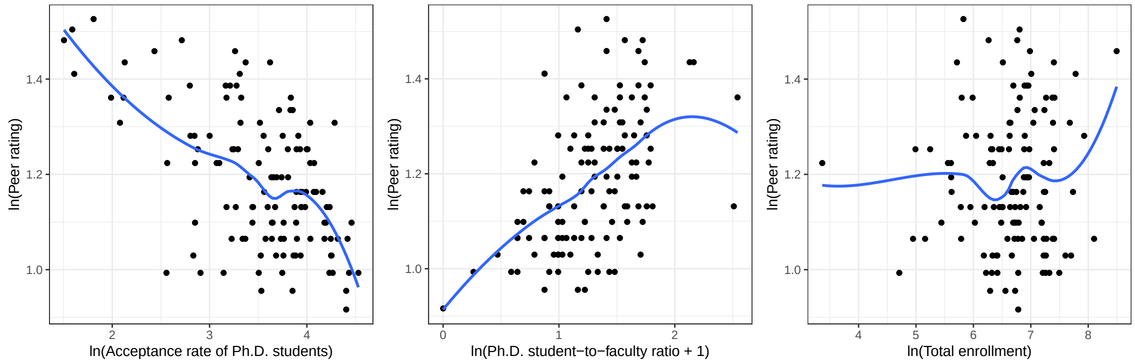
```
  skim_variable             n_missing complete_rate   mean     sd    p0   p25   p50     p75    p100
1 doc_accept                        0             1  40.1   20.2    4.5  25.5  38.6   51.6     92.7
2 phd_student_faculty_ratio         0             1   2.94   1.75   0     1.7   2.7    3.77    11.7
3 enroll                            0             1 970.    665.    29   562.  842.  1312.   4892
```

We will need to add one to every value of the `phd_student_faculty_ratio` predictor (so that the minimum value becomes 1) prior to the log-transformation.

```
#Create log of the institution-related predictors
educ = educ %>%
  mutate(
    Ldoc_accept = log(doc_accept),
    Lphd_student_faculty_ratio = log(phd_student_faculty_ratio + 1),
    Lenroll = log(enroll)
  )
```

Figure 11

Scatterplots of the Log-Transformed Peer Ratings versus the Log-Transformed Institution-Related Factors



Note. The loess smoother is also displayed in each plot.

The scatterplots indicate that all three relationships were satisfactorily linearized. After fitting the institution-related factors model we will further examine the coefficient-level output and residuals.

```
# Fit model
lm.3 = lm(Lpeer ~ 1 + Ldoc_accept + Lenroll + Lphd_student_faculty_ratio, data = educ)

# Model-level output
glance(lm.3)
```

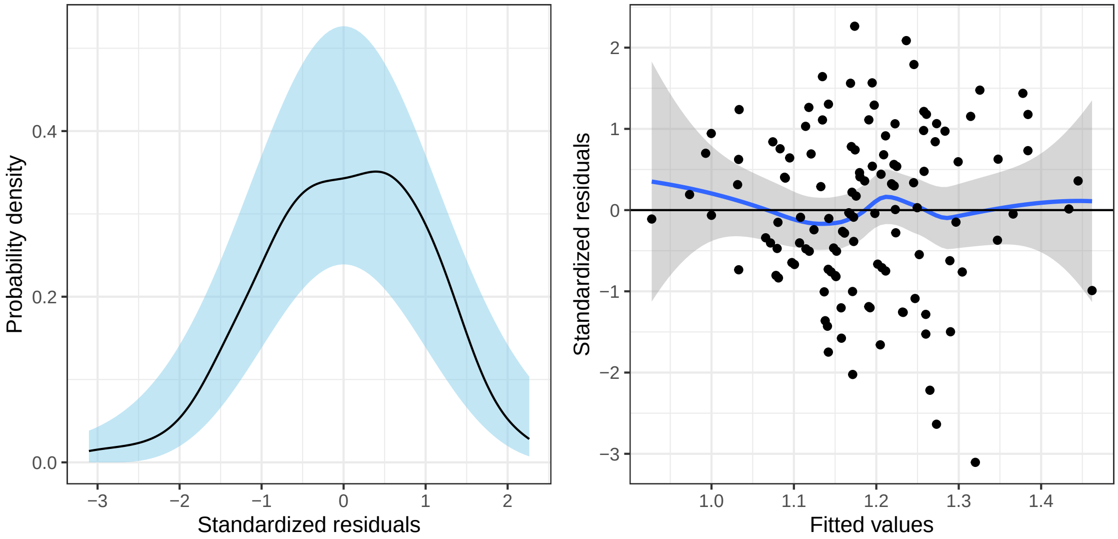
```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.460         0.446 0.107      33.5 9.68e-16     4   101. -193. -179.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
# Coefficient-level output
tidy(lm.3)
```

```
# A tibble: 4 x 5
  term                       estimate std.error statistic  p.value
  <chr>                         <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)                  1.31      0.108      12.1  1.96e-22
2 Ldoc_accept                 -0.109     0.0156     -6.97 1.95e-10
3 Lenroll                      0.0145    0.0136      1.07 2.87e- 1
4 Lphd_student_faculty_ratio   0.129     0.0247      5.20 8.42e- 7
```

```
# Obtain residuals
out_3 = augment(lm.3)

# Examine residuals
p1 = ggplot(data = out_3, aes(x = .std.resid)) +
  stat_density_confidence(model = "normal") +
  stat_density(geom = "line") +
  theme_bw() +
  xlab("Standardized residuals") +
  ylab("Probability density")

p2 = ggplot(data = out_3, aes(x = .fitted, y = .std.resid)) +
  geom_smooth(se = TRUE) +
  geom_hline(yintercept = 0) +
  geom_point() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")

p1 + p2
```

Figure 12
Residual Plots for the Institution-Related Factors Fitted Model



Note that although the coefficient-level output indicates that the log-transformed enrollment predictor may be unnecessary ($p = 0.287$), we will retain this predictor in the model as it was suggested by the substantive literature.

Candidate Statistical Models

Now that we have settled on the functional form for each of the three proposed models, we can write out the statistical models associated with the scientific hypotheses. These models using regression notation are:

Model 1

$$\text{Peer Rating}_i = \beta_0 + \beta_1(\text{GREQ}_i) + \beta_2(\text{GREQ}_i^2) + \beta_3(\text{GREQ}_i^3) + \epsilon_i$$

Model 2

$$\text{Peer Rating}_i = \beta_0 + \beta_1(\text{Funded research}_i) + \beta_2(\text{Funded research}_i^2) + \beta_3(\text{PhDs granted}_i) + \beta_4(\text{PhDs granted}_i^2) + \epsilon_i$$

Model 3

$$\text{Peer Rating}_i = \beta_0 + \beta_1(\text{PhD acceptance rate}_i) + \beta_2(\text{PhD student-to-faculty ratio}_i) + \beta_3(\text{Enrollment}_i) + \epsilon_i$$

where peer rating, funded research, Ph.D.s granted, Ph.D. acceptance rate, enrollment, and Ph.D. student-to-faculty ratio have all been log-transformed. We will also consider a fourth model that omits enrollment from the institution-related factors model.