

Data Club Fall 24 Surveying Project

January 4, 2025

```
[221]: import numpy as np
from datascience import *

import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import warnings
warnings.simplefilter('ignore', FutureWarning)

from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets
```

```
[222]: #ALPHAS = Survey about Transferring
#SIGMAS = Survey about Academic Programs
```

1 Survey Topic: Transferring

```
[223]: AlphasTable = Table().read_table("AlphasResults.csv")
AlphasTable = AlphasTable.drop("Timestamp")
AlphasTable
```

```
[223]: Select all of the following activities you are currently involved in. | Do you
want to transfer to a 4 year institution? | Do you want to transfer to an out-
of-state institution? | Which type of institution do you want to transfer to? |
After how many years at Community College do you hope to transfer (enter just
the number)
Community Service/Volunteer Work, Paid Work/Internship, Sports, I ... | Yes
| No | Public University
(UC's, CSU's, etc) | 2
Sports, Instruments, Hobbies, Personal Project/Business, School L ... | Yes
| No | Private University
| 1
Community Service/Volunteer Work, Paid Work/Internship, Sports, I ... | Yes
| No | Public University
(UC's, CSU's, etc) | 2
```

```

Community Service/Volunteer Work, Educational Preparation Program ... | Yes
| No | Public University
(UC's, CSU's, etc) | 2
Community Service/Volunteer Work | Yes
| Yes | Public University
(UC's, CSU's, etc) | 2
Community Service/Volunteer Work, Paid Work/Internship, Sports, I ... | Yes
| No | Public University
(UC's, CSU's, etc) | 2
Paid Work/Internship, Educational Preparation Program | Yes
| No | Public University
(UC's, CSU's, etc) | 2
Paid Work/Internship, Sports, Instruments, Hobbies | Yes
| No | Public University
(UC's, CSU's, etc) | 2
Community Service/Volunteer Work, Paid Work/Internship | Yes
| No | Public University
(UC's, CSU's, etc) | 2
Community Service/Volunteer Work, Paid Work/Internship, Sports, I ... | Yes
| No | Public University
(UC's, CSU's, etc) | 2
... (97 rows omitted)

```

2 Survey Topic: Academic Program

```

[224]: SigmasTable = Table().read_table("SigmasResults.csv")
SigmasTable = SigmasTable.drop("Timestamp")
SigmasTable

```

```

[224]: What is your major at Cypress College? | Which program is your major aligned
with? | Which program do you believe to be the hardest? | Which program do you
believe to be the easiest?

```

```

Nursing | Health Science
| Science, Engineering, Math | Social Sciences
Biology | Science, Engineering, Math
| Science, Engineering, Math | Visual & Performing Arts
Biology | Health Science
| Health Science | Social Sciences
Civil Engineering | Science, Engineering, Math
| Science, Engineering, Math | Social Sciences
Comp Sci | Science, Engineering, Math
| Science, Engineering, Math | Visual & Performing Arts
Engineering | Science, Engineering, Math
| Science, Engineering, Math | Social Sciences
Biology | Health Science
| Science, Engineering, Math | Language Arts

```

Data Science	Science, Engineering, Math
Science, Engineering, Math	Language Arts
Biology-premed	Science, Engineering, Math
Science, Engineering, Math	Visual & Performing Arts
Dental Assistant	Health Science
Business & Computer Information Systems	Visual & Performing Arts
... (111 rows omitted)	

All of the Academic Programs at Cypress College: Business & Computer Information Systems, Career Technical Education, Health Science, Kinesiology, Language Arts, Science, Engineering, Math, Social Sciences, Visual & Performing Arts

```
[225]: # Null Hypothesis: There is no tendency for students to choose their own
      ↪ academic program as either the easiest or hardest.
      #Any difference in proportions are due to chance alone

      # Alternative Hypothesis: There is tendency for students to select their own
      ↪ academic program as either the easiest or hardest.
      # Thus, their opinions may be biased.
```

```
[226]: def yourMajorIs (yourProg, hardestProg, easiestProg):
      '''If the student selected their own major program as the hardest program,
      ↪ return True'''
      if (yourProg == hardestProg):
          return "hardest"
      if (yourProg == easiestProg):
          return "easiest"
      else:
          return "neither"
```

```
[227]: yourMajor = SigmasTable.apply(yourMajorIs, "Which program is your major aligned
      ↪ with?", "Which program do you believe to be the hardest?", "Which program do
      ↪ you believe to be the easiest?")
```

```
[228]: SigmasTable = SigmasTable.with_column("Your Major Is", yourMajor)
      SigmasTable
```

```
[228]: What is your major at Cypress College? | Which program is your major aligned
      with? | Which program do you believe to be the hardest? | Which program do you
      believe to be the easiest? | Your Major Is
      Nursing | Health Science
      | Science, Engineering, Math | Social Sciences
      | neither
      Biology | Science, Engineering, Math
      | Science, Engineering, Math | Visual & Performing Arts
      | hardest
      Biology | Health Science
```

Health Science	Social Sciences
hardest	
Civil Engineering	Science, Engineering, Math
Science, Engineering, Math	Social Sciences
hardest	
Comp Sci	Science, Engineering, Math
Science, Engineering, Math	Visual & Performing Arts
hardest	
Engineering	Science, Engineering, Math
Science, Engineering, Math	Social Sciences
hardest	
Biology	Health Science
Science, Engineering, Math	Language Arts
neither	
Data Science	Science, Engineering, Math
Science, Engineering, Math	Language Arts
hardest	
Biology-premed	Science, Engineering, Math
Science, Engineering, Math	Visual & Performing Arts
hardest	
Dental Assistant	Health Science
Business & Computer Information Systems	Visual & Performing Arts
neither	
... (111 rows omitted)	

```
[229]: hardestProportion = SigmasTable.where("Your Major Is", are.equal_to("hardest")).
      ↪ num_rows/SigmasTable.num_rows
hardestProportion
easiestProportion = SigmasTable.where("Your Major Is", are.equal_to("easiest")).
      ↪ num_rows/SigmasTable.num_rows
easiestProportion
neitherProportion = SigmasTable.where("Your Major Is", are.equal_to("neither")).
      ↪ num_rows/SigmasTable.num_rows
neitherProportion
observedProportions = make_array(hardestProportion, easiestProportion,
      ↪ neitherProportion)
observedProportions
```

```
[229]: array([ 0.46280992,  0.04958678,  0.48760331])
```

3 HARDEST

```
[231]: # Test Statistic: Proportion of students who chose their academic program as
      ↪ the hardest program
```

```
[232]: #Observed Test Statistic:  
observedTestStatistic = observedProportions.item(0)  
observedTestStatistic
```

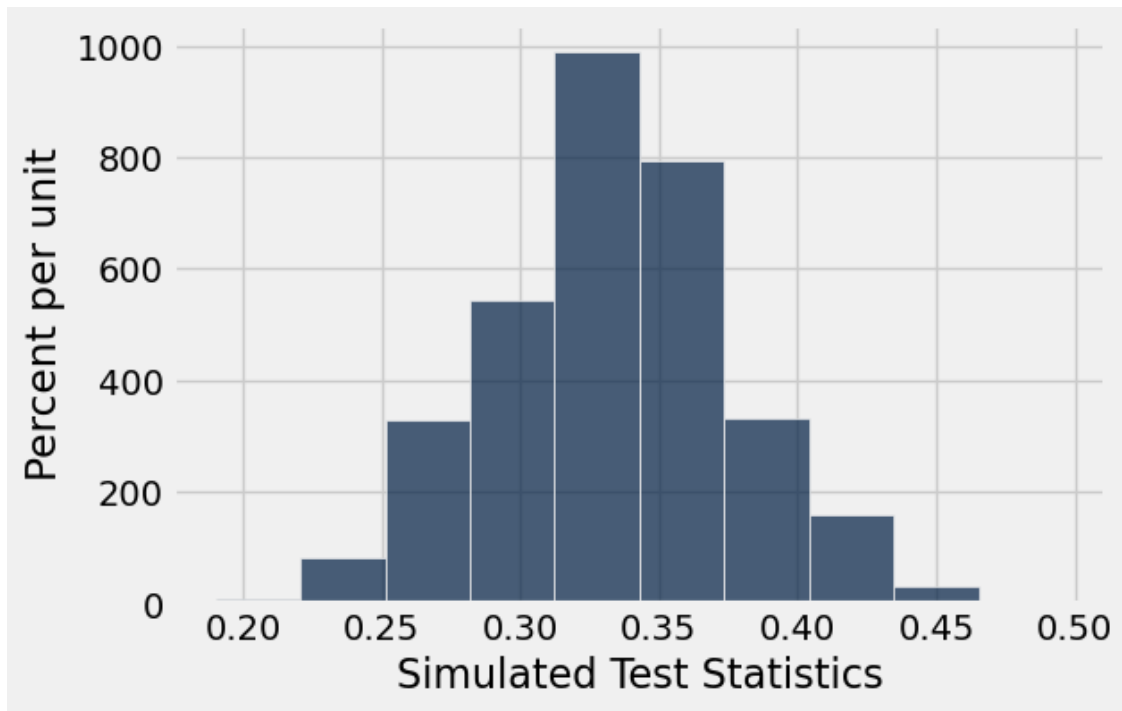
```
[232]: 0.4628099173553719
```

```
[254]: #We are simulating 10,000 trials of sampling from the modelProportions which is  
↳under the null hypothesis. This will serve as  
#the simulated test statistics  
modelProportions = make_array((1/3), (1/3), (1/3))  
appendedArrayHardest = make_array()  
appendedArrayEasiest = make_array()  
  
for i in np.arange(10000):  
    oneSimulationHardest = sample_proportions(121, modelProportions).item(0)  
    oneSimulationEasiest = sample_proportions(121, modelProportions).item(1)  
    appendedArrayHardest = np.append(appendedArrayHardest, oneSimulationHardest)  
    appendedArrayEasiest = np.append(appendedArrayEasiest, oneSimulationEasiest)
```

```
[234]: # Simulated Test Statistics:  
myTable = Table().with_column("Simulated Test Statistics", appendedArrayHardest)  
myTable
```

```
[234]: Simulated Test Statistics  
0.330579  
0.363636  
0.338843  
0.272727  
0.305785  
0.322314  
0.289256  
0.338843  
0.280992  
0.330579  
... (9990 rows omitted)
```

```
[235]: myTable.hist()
```



```
[236]: # P Value for Hardest:
```

```
[237]: pValueHardest = (np.count_nonzero(myTable.column("Simulated Test Statistics")
    ↳ >= observedTestStatistic)/10000) * 100
pValueHardest
```

```
[237]: 0.19
```

```
[245]: # 0.19% is less than the conventional P Cutoff value of 1%/0.5%. Thus, we have
    ↳ significant evidence for the alternative hypothesis.
```

4 EASIEST

```
[240]: #Test Statistic: Proportion of students who chose their academic program as the
    ↳ easiest program
```

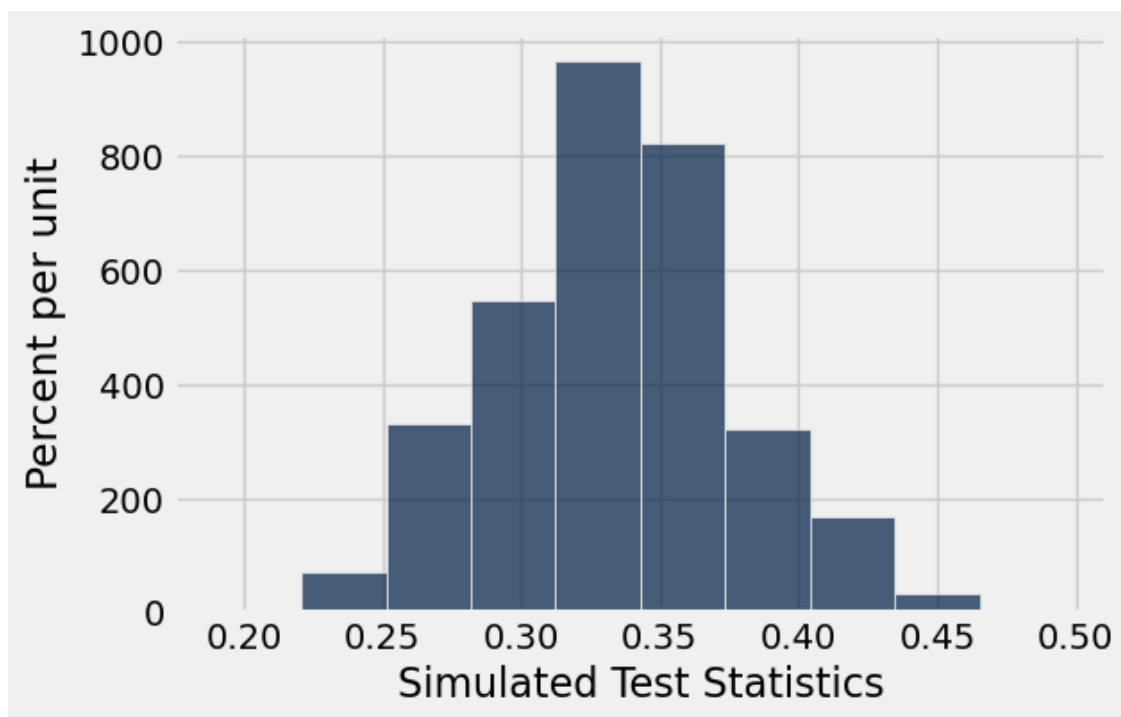
```
[241]: #Observed Test Statistic:
observedTestStatistic = observedProportions.item(1)
observedTestStatistic
```

```
[241]: 0.049586776859504134
```

```
[242]: #Simulated Test Statistics:  
myTable = Table().with_column("Simulated Test Statistics", appendedArrayEasiest)  
myTable
```

```
[242]: Simulated Test Statistics  
0.31405  
0.322314  
0.31405  
0.330579  
0.31405  
0.338843  
0.363636  
0.206612  
0.429752  
0.280992  
... (9990 rows omitted)
```

```
[243]: myTable.hist()
```



```
[253]: #P Value for Easiest is unnecessary as it can be seen from the histogram above,  
      ↪ that the observed test statistic (0.0496 approx.)  
      #is nowhere within the range of the simulated test statistics
```

5 Conclusion

Conclusion: With `pValueHardest` being 0.19 (less than 0.5; 0.5% p-cutoff value), we are 99.5% confident that there is tendency for students to select their own academic program as either the easiest or hardest (alternative hypothesis). If there was no tendency whatsoever, there would be an approximate distribution of 0.33, 0.33, 0.33 for the `observedProportions` which are the proportions of students that claimed their major to be the hardest, easiest, neither (exact order). Instead, `observedProportions` had a distribution of 0.46280992, 0.04958678, 0.48760331. According to our P-Value, these proportions were not due to chance. Right away, we can see that way more students (approx 46.28% of them) chose their own academic program as the hardest while way less students (approx. 4.96% of them) chose their own academic program as the easiest. Thus, students at Cypress College are more likely to say that their own program is the hardest (or simply neither easiest/hardest) than say that it is the easiest. In conclusion, asking a student about what they believe to be easiest/hardest academic program can result in a biased response.

```
[252]: hardest = SigmasTable.where("Your Major Is", are.equal_to("hardest"))
SEM = (hardest.where("Which program is your major aligned with?", are.
    ↪equal_to("Science, Engineering, Math")).num_rows)/hardest.num_rows
HS = (hardest.where("Which program is your major aligned with?", are.
    ↪equal_to("Health Science")).num_rows)/hardest.num_rows
SEM, HS
```

```
[252]: (0.6428571428571429, 0.30357142857142855)
```

Bias in Survey Collection: Students in the academic programs “Science, Engineering, Math” and “Health Science” were overrepresented. This could explain the significant difference in `observedProportions`: These two academic programs are widely recognized as the hardest out of the others. To support this belief, the above cell shows that approx. 64.29% of those that claimed their own program to be the hardest were in “Stem, Engineering, Math” while 30.36% were in “Health Science”.