Hypothesis Testing With Python

True Difference or Noise?

0.72

0.76

Which is better?

Noise?

That's a question.

Mosky

- Python Charmer at Pinkoi.
- Has spoken at: PyCons in TW, MY, KR, JP, SG, HK, COSCUPs, and TEDx, etc.
- Countless hours on teaching Python.
- Own the Python packages like ZIPCodeTW.
- http://mosky.tw/

Outline

- Introduction
- Welch's t-test
- Chi-squared test
- Power analysis
- More tests
- Complete steps
- Theory
  - P-value & α
  - Raw effect size, β, sample Size
  - Actual negative rate, inverse α, inverse β

The PDF, Notebooks, and Packages

- The PDF and notebooks are available on https://github.com/moskytw/hypothesis-testing-with-python .
- The packages:
  - `$ pip3 install jupyter numpy scipy sympy matplotlib ipython pandas seaborn statsmodels scikit-learn`

    Or:
  - `> conda install jupyter numpy scipy sympy matplotlib ipython pandas seaborn statsmodels scikit-learn`

To buy, or not to buy

- Going to buy a bulb on an online store.
- If see 10/100 bad reviews? Hmm …
- If see 5/100 bad reviews? Good to buy.
- If see 1/100 bad reviews? Good to buy.

- Going to buy a notebook computer on an online store.
- If see 10/100 bad reviews? Hmm ...
- If see 5/100 bad reviews? Hmm ...
- If see 1/100 bad reviews? Maybe good enough.
- Context matters.

Build our "bad reviews" in statistics

- Build a statistical model by a hypothesis.
  - "The means of two populations are equal."
  - $\equiv \mathrm{E}[X] = \mathrm{E}[Y]$
- Put the data into the model, get a probability, *p-value*.
  - "Given the model, the probability to observe the data."
- If see *p-value* $= 0.10$?
- If see *p-value* $= 0.05$?
- If see *p-value* $= 0.01$?
- Decide by your context.

Equal or not

- If the hypothesis contains "equal":
  - Can build a model directly, like the previous slide.
  - Called a **null** hypothesis.
- If the hypothesis contains "not equal":
  - Can build a model by negating it.
  - Called an **alternative** hypothesis.
- **P-value**: given a **null**, the probability to observe the data.

The threshold

- *a*: significance level, 0.05 usually, or decided by context.
- If *p-value* $<$ *a*:
  - Can reject the null, i.e., can reject the equal.
  - Can accept the alternative, i.e., can accept the not-equal.
- If *p-value* $\geq$ *a*:
  - ~~Can accept the null, i.e., can accept the equal.~~
    - “Given the null, the probability of the data is 6%.”
  - Can't reject the null.
  - Can't accept the alternative.
  - We may investigate further.

Formats suggested by APA and NEJM

p-value & α	Wording	Summary
p-value < 0.001	Very significant	***
p-value < 0.01	Very significant	**
p-value < 0.05	Significant	*
p-value ≥ 0.05	Not significant	ns

- Many researchers suggest to report **without** formatting.
  - Since the largely misunderstandings:
    - Misunderstandings of p-values – Wikipedia
    - Scientists rise up against statistical significance – Natural
      - “We are not calling for a ban on P values. Nor are we saying they cannot be used as a decision criterion in certain specialized applications.”
      - “We are calling for a stop to the use of P values in the conventional, dichotomous way — to decide whether a result refutes or supports a scientific hypothesis.”

Define assumptions

- The hypothesis testing:
  - Suitable to answer a yes–no question:
    - "Means or medians of two populations are equal?"
      - E.g., "The order counts of A and B are equal?"
    - "Proportions of two populations are equal?"
      - E.g., "The conversion rates of A and B are equal?"

- “Poor or non-poor marriage has different affair times?”
- “Poor or non-poor marriage has different affair proportion?”
- “Occupations have different affair times?”
- “Occupations have different affair proportion?”

Validate assumptions

- Collect data ...
  - The "Fair" dataset:
    - Fair, Ray. 1978. "A Theory of Extramarital Affairs," Journal of Political Economy, February, 45-61.
    - A dataset from 1970s.
    - Rows: 6,366
    - Columns: (next slide)
- The full version of the analysis steps: http://bit.ly/analysis-steps .

1. *rate_marriage*: 1~5; very poor, poor, fair, good, very good.
2. *age*
3. *yrs_married*
4. *children*: number of children.
5. *religious*: 1~4; not, mildly, fairly, strongly.
6. *educ*: 9, 12, 14, 16, 17, 20; grade school, some college, college graduate, some graduate school, advanced degree.
7. *occupation*: 1, 2, 3, 4, 5, 6; student, farming-like, white-colloar, teacher-like, business-like, professional with advanced degree.
8. *occupation_husb*
9. *affairs*: n times of extramarital affairs per year since marriage.


JESUS SAV
Fill up with the Holy Ghost & Fire

Summary of the tests today

	Non-poor	Poor	Uplift	P-value	
Times	0.64	1.52	+138%	< 0.001 ***	#1
Prop.	30%	66%	+120%	< 0.001 ***	#2

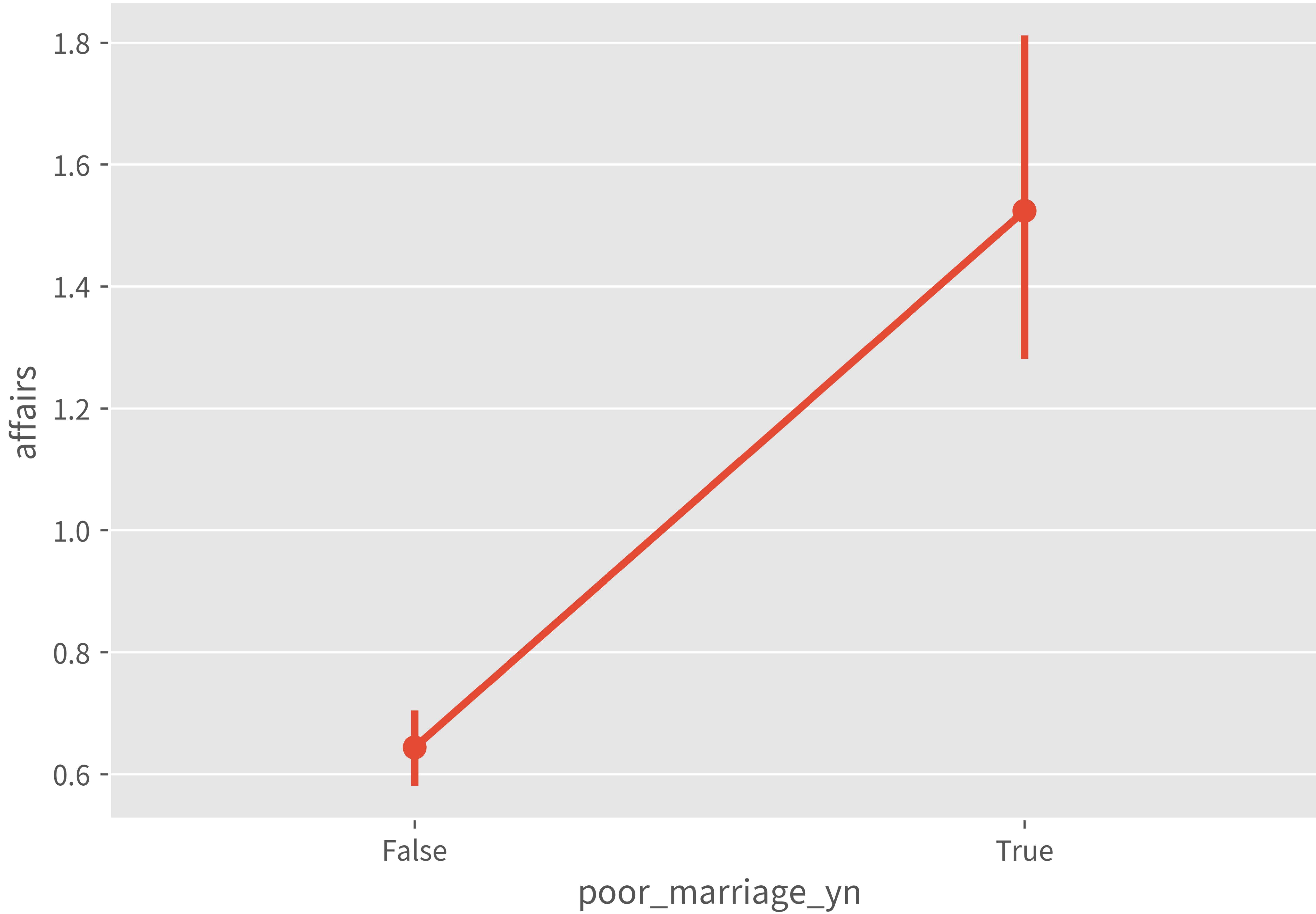
	Farming-like	White-colloar	Uplift	P-value	
Times	0.72	0.76	+6%	0.698 ns	#3
Prop.	29%	35%	+21%	0.004 **	#4

#1 Welch's t-test

	count	mean	std
poor_marriage_yn			
False	5919.0	0.643549	2.116982
True	447.0	1.524038	3.015937

```
p-value: 2.7446844166802127e-09
```

- Preprocess:
  - Group into poor or not.
- Describe.
- Test:
  - Assume the affair times are equal, the probability to observe it: super low.
- So, we accept the times are not equal at 1% significance level.
  - Non-poor: 0.64
  - Poor: 1.52



1.8
1.6
1.4
1.2
1.0
0.8
0.6
affairs
False
True
poor_marriage_yn

```python
import scipy as sp
import statsmodels.api as sm
import seaborn as sns


print(sm.datasets.fair.SOURCE,
      sm.datasets.fair.NOTE)


# -> Pandas's Dataframe
df_fair = sm.datasets.fair.load_pandas().data


df = df_fair
# 2: poor
# 3: fair
df = df.assign(poor_marriage_yn
               =(df.rate_marriage <= 2))
df_fair_1 = df
```

```python
df = df_fair_1

display(df
        .groupby('poor_marriage_yn')
        .affairs
        .describe())

a = df[df.poor_marriage_yn].affairs
b = df[~df.poor_marriage_yn].affairs

# ttest_ind(...) === Student's t-test
# ttest_ind(..., equal_var=False) === Welch's t-test
print('p-value:',
      sp.stats.ttest_ind(a, b, equal_var=False)[1])
```

```
df = df_fair_1
sns.pointplot(x=df.poor_marriage_yn,
              y=df.affairs)
```

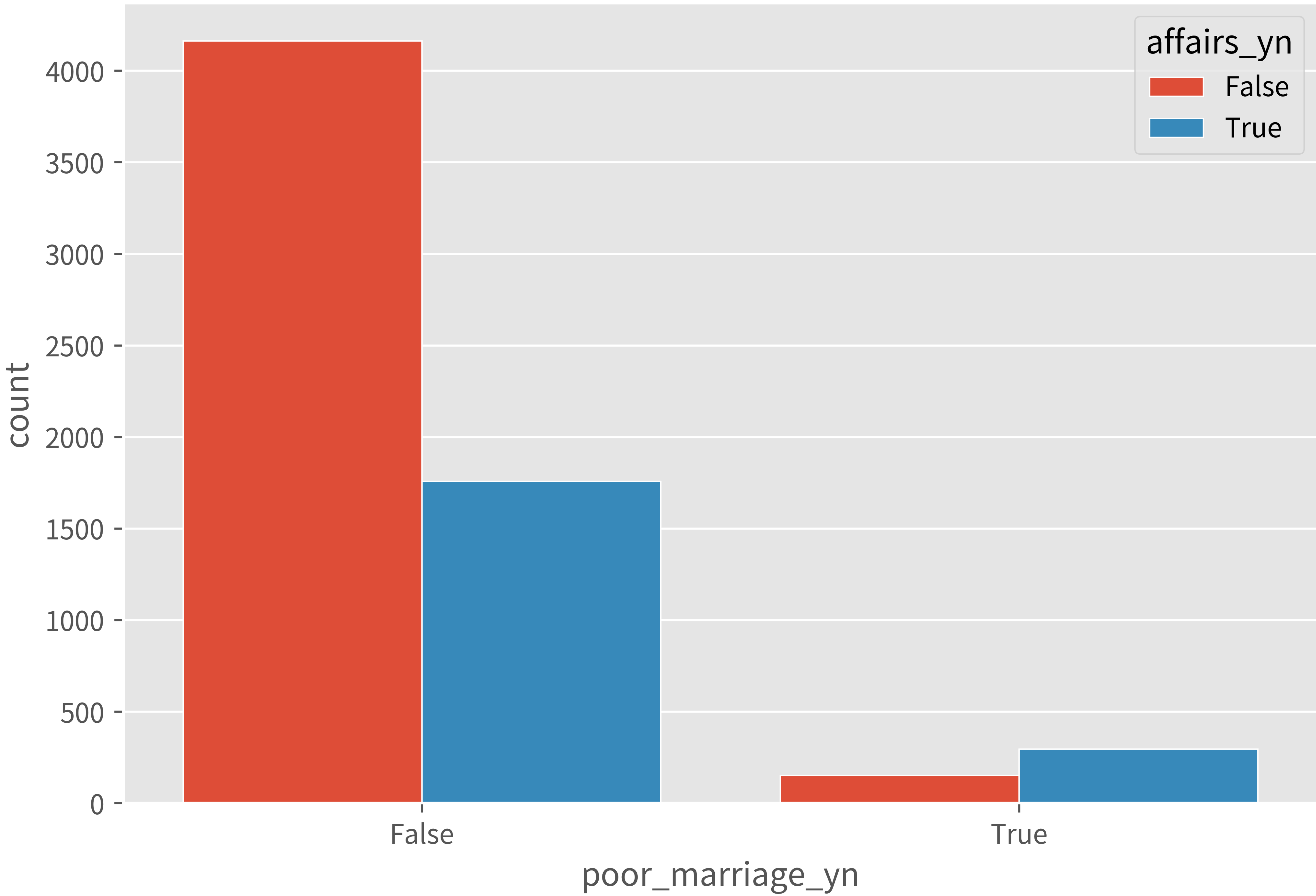
#2 Chi-squared test

affairs_yn	False	True
poor_marriage_yn		
False	4161	1758
True	152	295

affairs_yn	False	True
poor_marriage_yn		
False	0.702990	0.297010
True	0.340045	0.659955

```
p-value: 1.9460298519537103e-56
```

- Preprocess:
  - Add “affairs > 0” as true.
  - Group into poor or not.
- Describe.
- Test:
  - Assume the affair proportions are equal, the probability to observe it: super low.
- So, we accept the proportions are not equal at 1% significance level.
  - Non-poor: 30%
  - Poor: 66%


affairs_yn
False
True
count
4000
3500
3000
2500
2000
1500
1000
500
0
False
True
poor_marriage_yn

```
df = df_fair
# 2: poor
# 3: fair
df = df.assign(poor_marriage_yn
               =(df.rate_marriage <= 2),
               affairs_yn=(df.affairs > 0))
df_fair_3 = df
```

```python
df = df_fair_3

df = (df
      .groupby(['poor_marriage_yn', 'affairs_yn'])
      [['affairs']]
      .count()
      .unstack()
      .droplevel(axis=1, level=0))

df_pct = df.apply(axis=1, func=lambda r: r/r.sum())

display(df, df_pct)

print('p-value:',
      sp.stats.chi2_contingency(
          df,
          correction=False
      )[1])
```

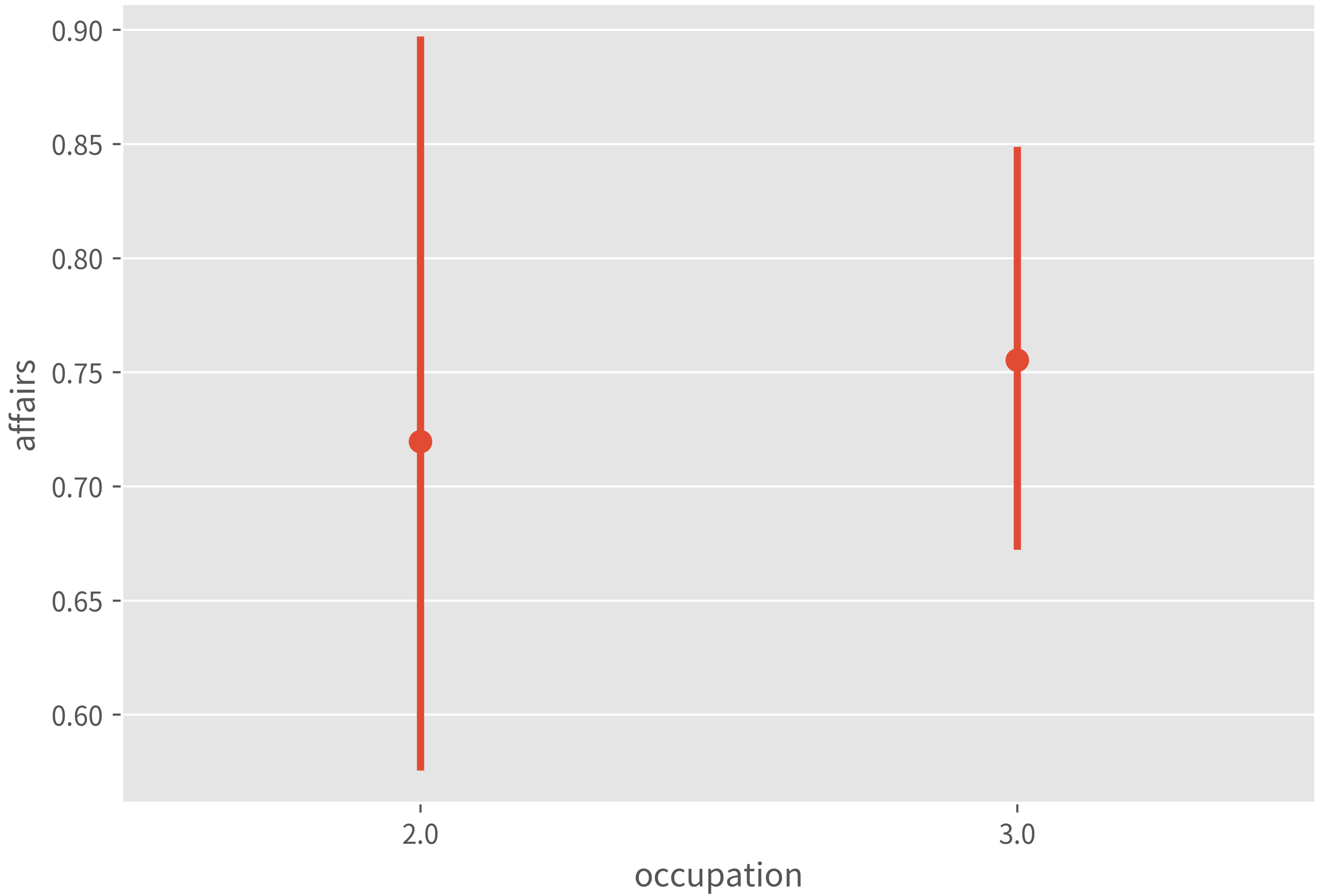
```python
df = df_fair_3
sns.countplot(data=df,
              x='poor_marriage_yn', hue='affairs_yn',
              saturation=0.95, edgecolor='white')
```

#3 Welch's t-test

occupation	count	mean	std
2.0	859.0	0.719556	2.375644
3.0	2783.0	0.755248	2.305594

```
p-value: 0.698381462473247
```

- Preprocess:
  - Select the two occupations.
  - Group by the occupations.
- Describe.
- Test:
  - Assume the affair times are equal, the probability to observe it: 70%.
- So, we can't accept the times are not equal at 1% significance level.
  - Farming-like: 0.72
  - White-colloar: 0.76

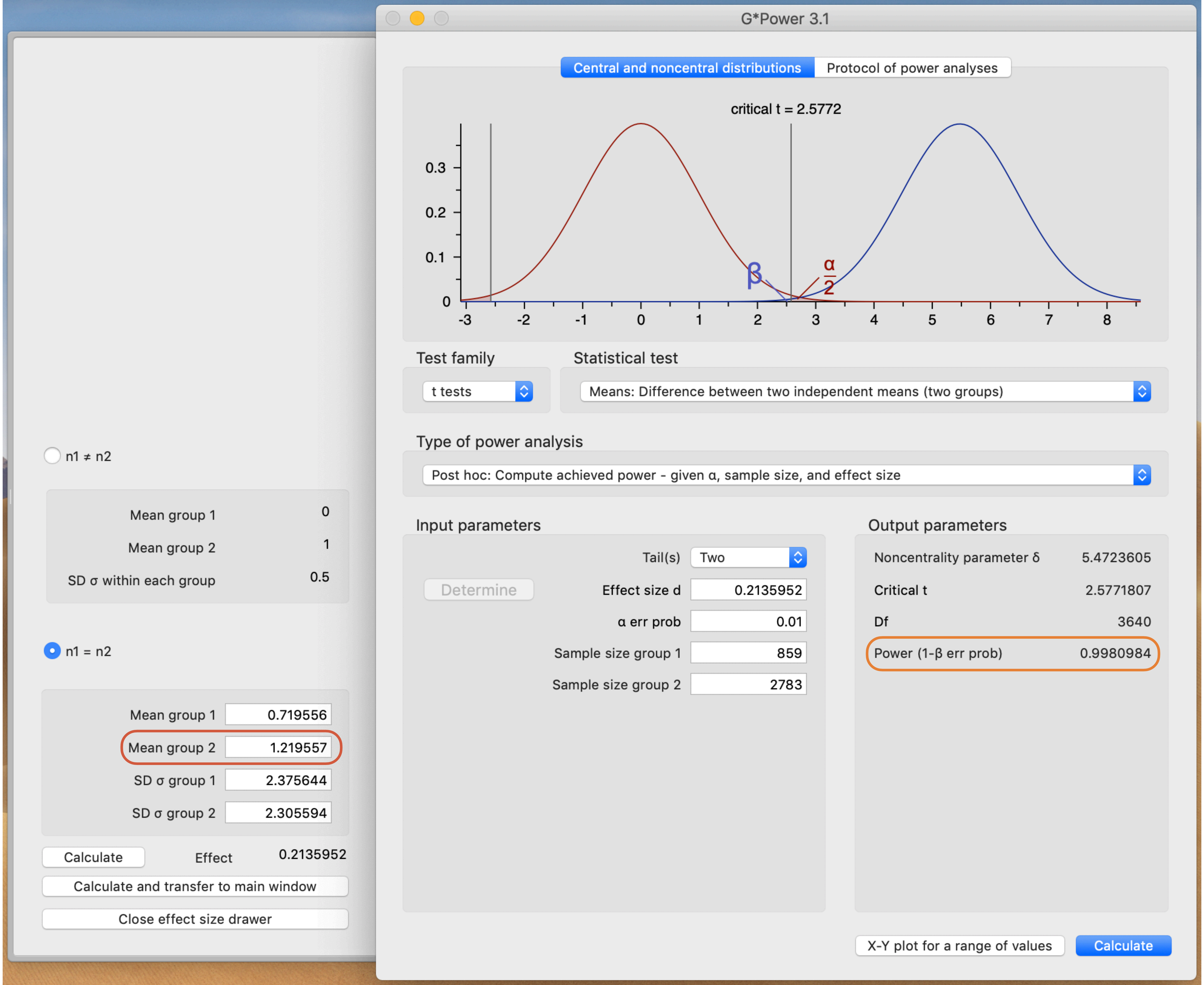

0.90
0.85
0.80
0.75
0.70
0.65
0.60
affairs
2.0
3.0
occupation

```python
df = df_fair
# 2: farming-like
# 3: white-colloar
df = df[df.occupation.isin([2, 3])]
df_fair_2 = df


df = df_fair_2

display(df
        .groupby('occupation')
        .affairs
        .describe())

a = df[df.occupation == 2].affairs
b = df[df.occupation == 3].affairs

print('p-value:',
      sp.stats.ttest_ind(a, b, equal_var=False)[1])
```

```python
df = df_fair_2
sns.pointplot(x=df.occupation,
              y=df.affairs,
              join=False)


print('p-value:',
      sp.stats.ttest_ind([1, 2, 3, 4, 5, 6],
                         [1, 2, 3, 4, 5, 60],
                         equal_var=False)[1])
```

If there is a true difference, can we detect it?

- To detect $\geq$ 0.5 times difference at 1% significance level:
  - *raw effect size* $= 0.5$
  - $a = 0.01$
- Use G*Power or StatsModels:
  - *power* $= 0.9981$
- If there is a 0.5 times difference and the given significance level, we can detect it 99.81% of the time. It's good.
- So, we accept the times are equal or the difference < 0.5.
- If *power* is low, relax *effect size*, *a*, or collect a larger sample.


G*Power 3.1
Central and noncentral distributions
Protocol of power analyses
critical t = 2.5772
β
α
2
Test family
t tests
Statistical test
Means: Difference between two independent means (two groups)
Type of power analysis
Post hoc: Compute achieved power - given α, sample size, and effect size
Input parameters
Tail(s) Two
Determine
Effect size d 0.2135952
α err prob 0.01
Sample size group 1 859
Sample size group 2 2783
Output parameters
Noncentrality parameter δ 5.4723605
Critical t 2.5771807
Df 3640
Power (1-β err prob) 0.9980984
X-Y plot for a range of values
Calculate
n1 ≠ n2
Mean group 1 0
Mean group 2 1
SD σ within each group 0.5
n1 = n2
Mean group 1 0.719556
Mean group 2 1.219557
SD σ group 1 2.375644
SD σ group 2 2.305594
Calculate
Effect 0.2135952
Calculate and transfer to main window
Close effect size drawer

The similar concepts

		Statistics	Understandable
α	= 1 - confidence level	✔	
power	= 1 - β	✔	✔
β	= 1 - power		
confidence level	= 1 - α		✔

		Statistics	Understandable
“reject null”	≡ “accept alter.”	✓	
“accept alter.”	≡ “reject null”		✓
“can't reject null”	≡ “investigate further”	✓	
“investigate further”	≡ “can't reject null”		✓

Power analysis

- $f(a, \text{raw effect size}, \text{power}) = \text{sample size}$
- Before collecting data:
  - Define *a, raw effect size, power* to calculate required *sample size*.
- After test:
  - If $\text{p-value} < a$, good to say there is a difference.
  - If $\text{p-value} \geq a$, or closes to *a*, may investigate the *power*.
- The *a, raw effect size, power* here are "to-achieve", not "observed".
- 2×2 chi-squared test ≡ two-proportion z-test. [ref]
  - The power analysis of two-proportion z-test is much easier.

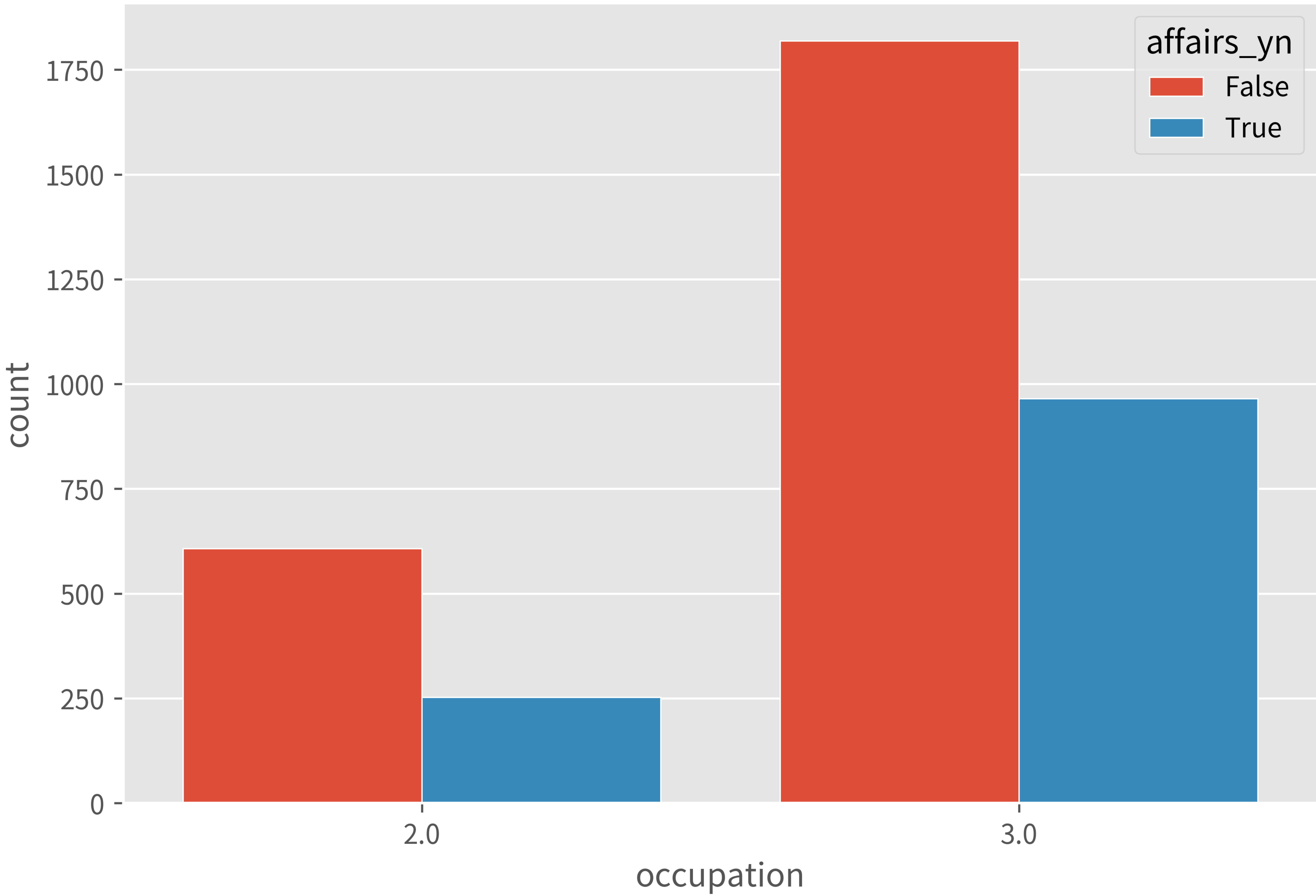
affairs_yn	False	True
occupation		
2.0	607	252
3.0	1818	965

affairs_yn	False	True
occupation		
2.0	0.706636	0.293364
3.0	0.653252	0.346748

```
p-value: 0.0037369587127306517
```

#4 Chi-squared test

- Preprocess:
  - Add “affairs > 0” as true.
  - Select the two occupations.
  - Group by the occupations.
- Describe.
- Test:
  - Assume the affair proportions are equal, the probability to observe it: 0.4%.
- So, we accept the proportions are not equal at 1% significance level:
  - Farming-like: 29%
  - White-colloar: 35%



affairs_yn
False
True
count
1750
1500
1250
1000
750
500
250
0
2.0
3.0
occupation

```python
df = df_fair_3
# 2: farming-like
# 3: white-colloar
df = df[df.occupation.isin([2, 3])]
df_fair_4 = df
```

```
df = df_fair_4

df = (df
      .groupby(['occupation', 'affairs_yn'])
      [['affairs']]
      .count()
      .unstack()
      .droplevel(axis=1, level=0))

df_pct = df.apply(axis=1, func=lambda r: r/r.sum())

display(df, df_pct)

print('p-value:',
      sp.stats.chi2_contingency(
          df,
          correction=False
      )[1])
```

```python
df = df_fair_4
sns.countplot(data=df,
              x='occupation', hue='affairs_yn',
              saturation=0.95, edgecolor='white')


print('p-value:',
      sp.stats.chi2_contingency(
          [[607, 252],
           [1818, 965]],
          correction=False
      )[1])
```

The mini cheat sheet

- If testing proportions, chi-squared test.
- If testing medians, Mann–Whitney U test.
- If testing means, Welch's t-test.

The cheat sheet

- If testing homogeneity:
  - If total sample size < 1000, or
    more than 20% of cells have expected frequencies < 5, Fisher's exact test.
  - Else, **chi-squared test**, or 2×2 chi-squared test ≡ two-proportion z-test.
- If testing equality:
  - If median is better, don't want to trim outliers,
    variable is ordinal, or any group size ≤ 20:
    - If groups are paired, Wilcoxon signed-rank test.
    - If groups are independent, Mann–Whitney U test.
  - Else:
    - If groups are paired, Paired Student's t-test.
    - If groups are independent, **Welch's t-test**, not Student's.

Why Welch's t-test, not Student's t-test?

- Student's t-test assumed the two populations have the same variance, which may not be true in most cases.
- Welch's t-test relaxed this assumption without side effects.
- So, just use Welch's t-test directly. [ref]

- More cheat sheets:
  - Selecting Commonly Used Statistical Tests – Bates College
  - Choosing a statistical test – HBS
- References:
  - Fisher's exact test of independence – HBS
  - Statistical notes for clinical researchers – Restor Dent Endod
  - Nonparametric Test and Parametric Test – Minitab
  - Dependent t-test for paired samples – Student's t-test – Wikipedia

Complete steps

1. Decide what test.
2. Decide *a*, *raw effect size*, *power* to achieve.
3. Calculate *sample size*.
4. Still collect a sample as large as possible.
5. Test.
6. Investigate *power* if need.
7. Report fully, not only significant or not.
    - Means, confidence intervals, p-values, research design, etc.

Keep learning

- Seeing Theory
- Statistics – SciPy Tutorial
- StatsModels
- Biological Statistics
- Research Design

Recap

- The null hypothesis is the one which states “equal”.
- The *p-value* is:
  - Given **null**, the probability to observe the data.
  - “How compatible the null hypothesis and the data are.”
- The Welch's t-test and chi-squared test.
- The power analysis to calculate sample size or power.
- Report fully, not only significant or not.
- Let's evaluate hypotheses efficiently! 📊

P-value & α

Theory

Seeing is believing

- *p-value* = 0.0027 ($<$ 0.01)
  - 👏🏻👏🏻👏🏻
- *p-value* = 0.0271 (0.01–0.05)
  - 👏🏻❓👏🏻❓❓❓
- *p-value* = 0.2718 ($\geq$ 0.05)
  - ❓❓❓❓❓❓
- *appendixes/theory_01_how_tests_work.ipynb*

Confusion matrix, where $A = 00_2 = C[0, 0]$

	predicted negative AC	predicted positive BD
actual negative AB	true negative A	false positive B
actual positive CD	false negative C	true positive D

False positive rate = P(BD|AB) = B/AB = 4/(96+4) = 4/100

	predicted negative AC	predicted positive BD
actual negative AB	96 A	4 B
actual positive CD	9 C	41 D

α = P(reject null|null) = P(predicted positive|actual negative)

	predicted negative AC	predicted positive BD
actual negative AB	true negative A	false positive B
actual positive CD	false negative C	true positive D

Predefined acceptable confusion matrix

	predicted negative AC	predicted positive BD
actual negative AB	true negative A	false positive B
actual positive CD	false negative C	true positive D

False positive, p-value, and α

false positive rate	Calculated with the actual answer.
p-value	Calculated false positive rate by a null hypothesis.
α	Predefined acceptable false positive rate.

Raw effect size, β, sample size

Theory

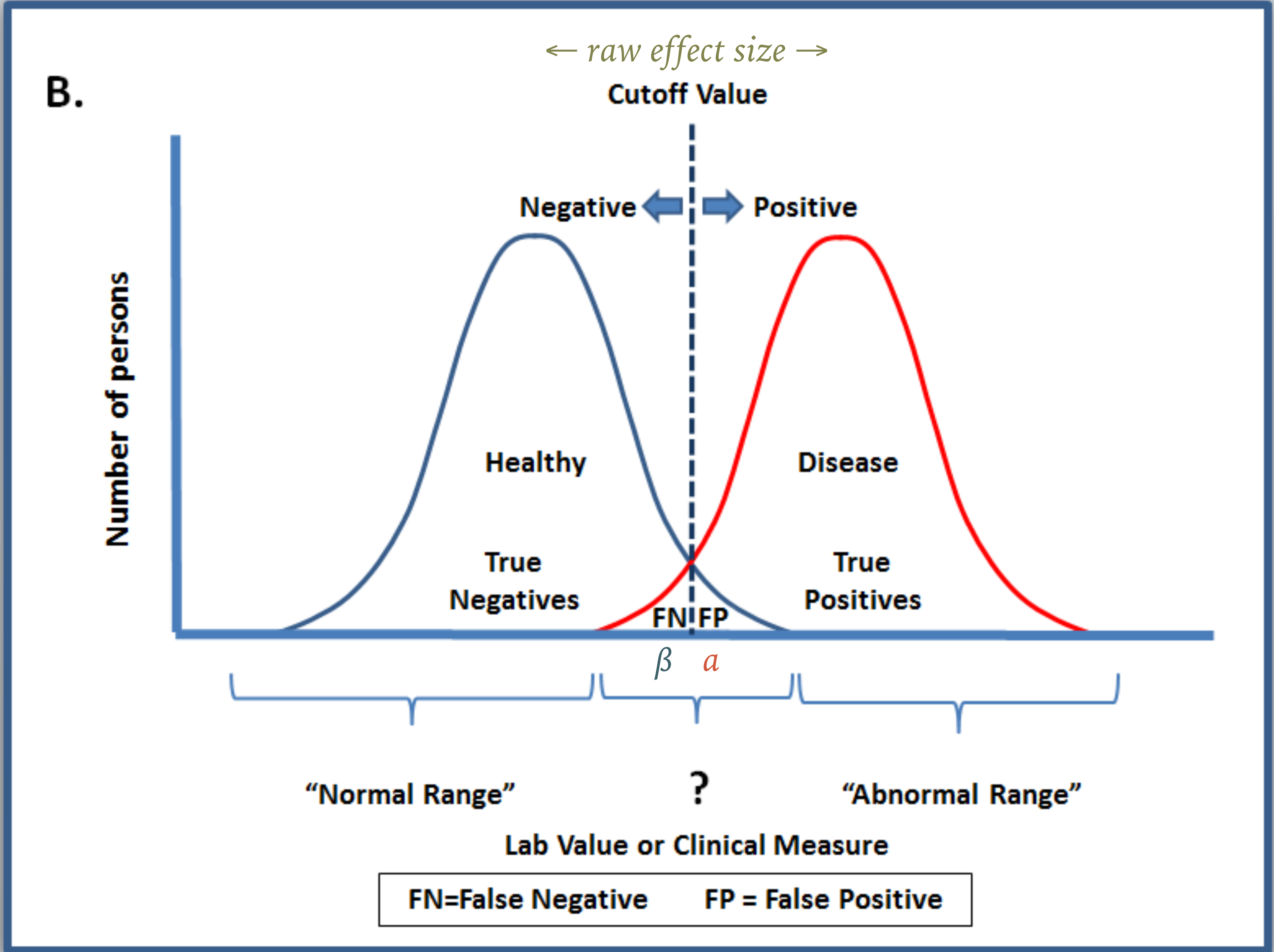
The elements of a complete test

1. The null hypothesis, data, p-value, α.
2. The raw effect size, β, sample size.
3. The false negative rate, inverse α, inverse β.

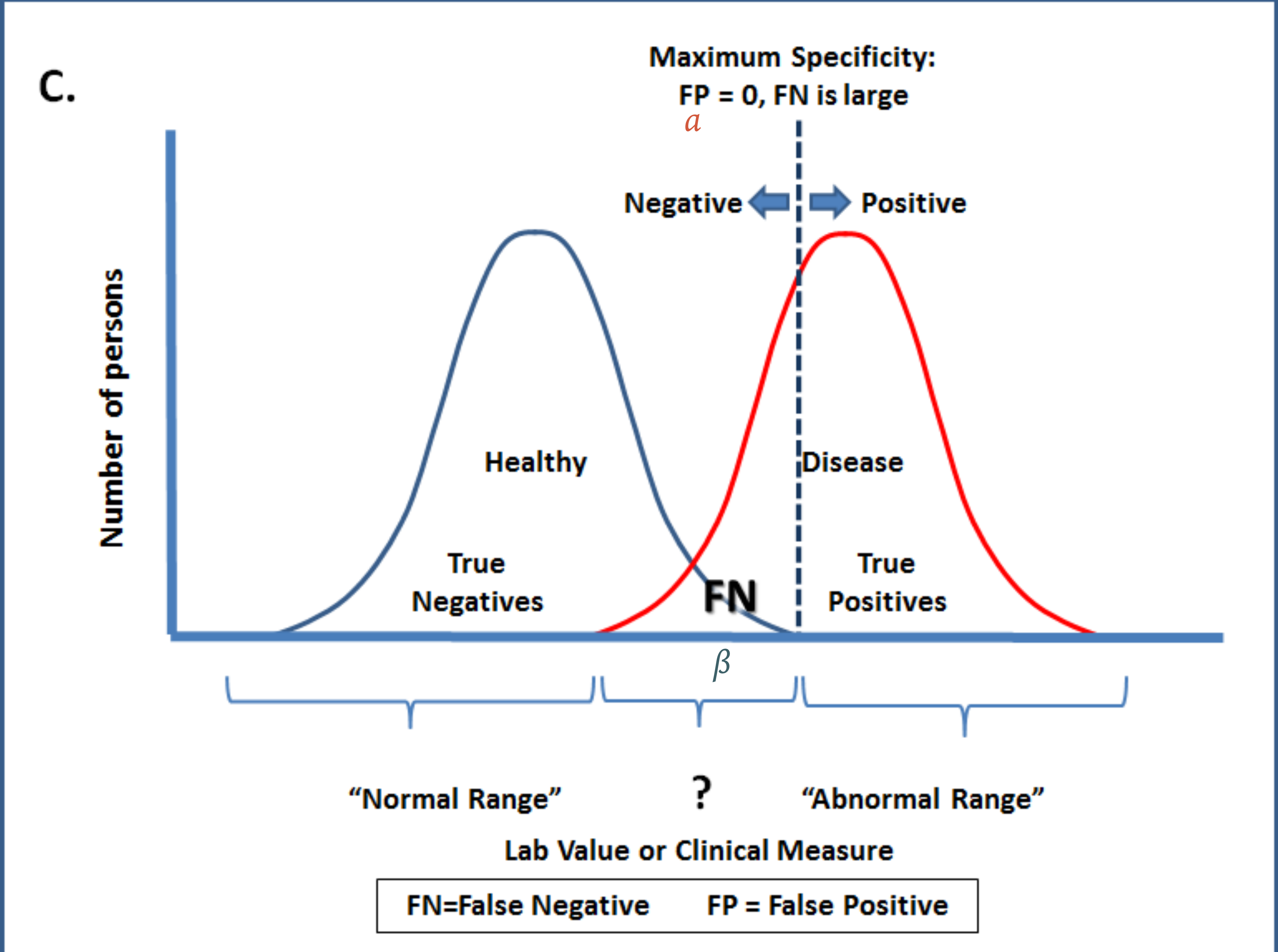
- Will introduce them by the confusion matrix.

Raw effect size, and β

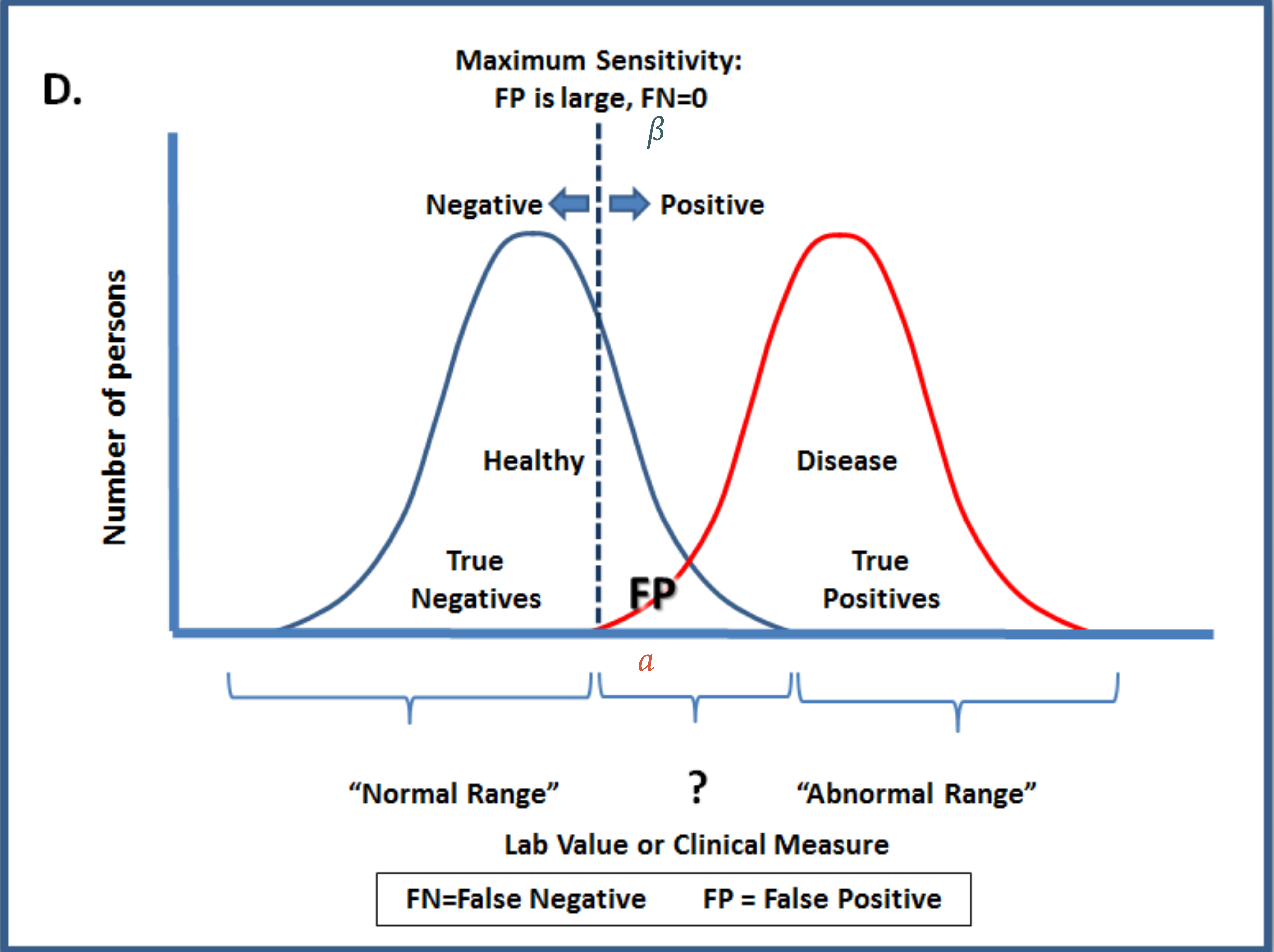
- DSM5: The case for double standards – James Coplan, M.D.
    - The figures explain *a*, *raw effect size*, and *β* perfectly.
    - “FP”: *a*
    - “The distance between the means”: *raw effect size*
    - “FN”: *β*



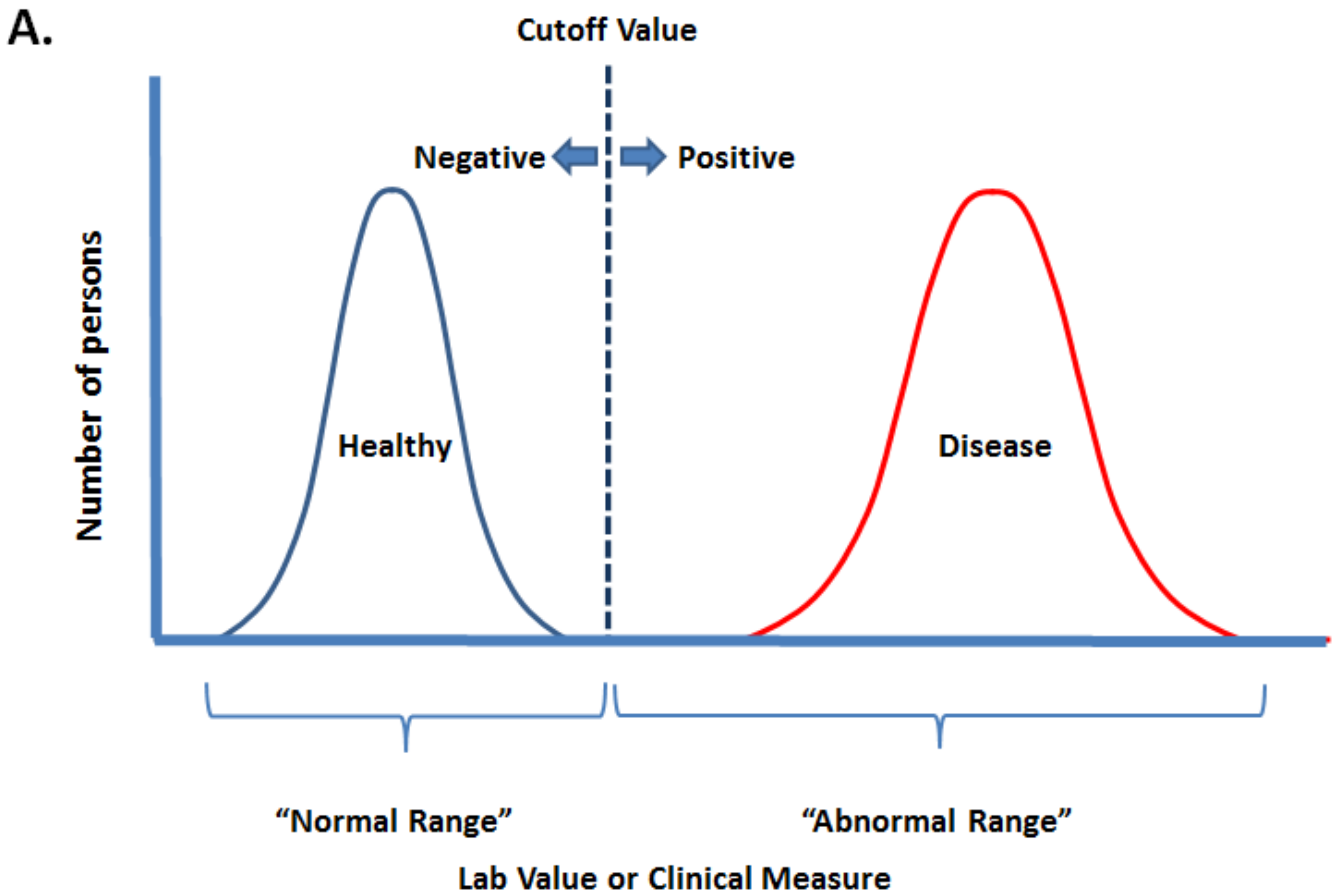
← raw effect size →
B.
Cutoff Value
Negative
Positive
Number of persons
Healthy
Disease
True
Negatives
True
Positives
FN
FP
β
a
"Normal Range"
?
"Abnormal Range"
Lab Value or Clinical Measure
FN=False Negative
FP = False Positive



C.
Maximum Specificity:
FP = 0, FN is large
a
Negative
Positive
Number of persons
Healthy
Disease
True
Negatives
FN
True
Positives
β
"Normal Range"
?
"Abnormal Range"
Lab Value or Clinical Measure
FN=False Negative
FP = False Positive


D.
Maximum Sensitivity:
FP is large, FN=0
β
Negative
Positive
Number of persons
Healthy
Disease
True
Negatives
FP
True
Positives
a
"Normal Range"
?
"Abnormal Range"
Lab Value or Clinical Measure
FN=False Negative
FP = False Positive

sample size ↑


A.
Cutoff Value
Negative
Positive
Number of persons
Healthy
Disease
"Normal Range"
"Abnormal Range"
Lab Value or Clinical Measure

$\beta = P(AC|CD) = C/CD$

	predicted negative AC	predicted positive BD
actual negative AB	true negative A	false positive B
actual positive CD	false negative C	true positive D

- Given α, raw effect size, β, get the sample size.
- Given α, raw effect size, sample size, get the β.
- Increase sample size to decrease α, β, or raw effect size.

Actual negative rate, inverse α, inverse β

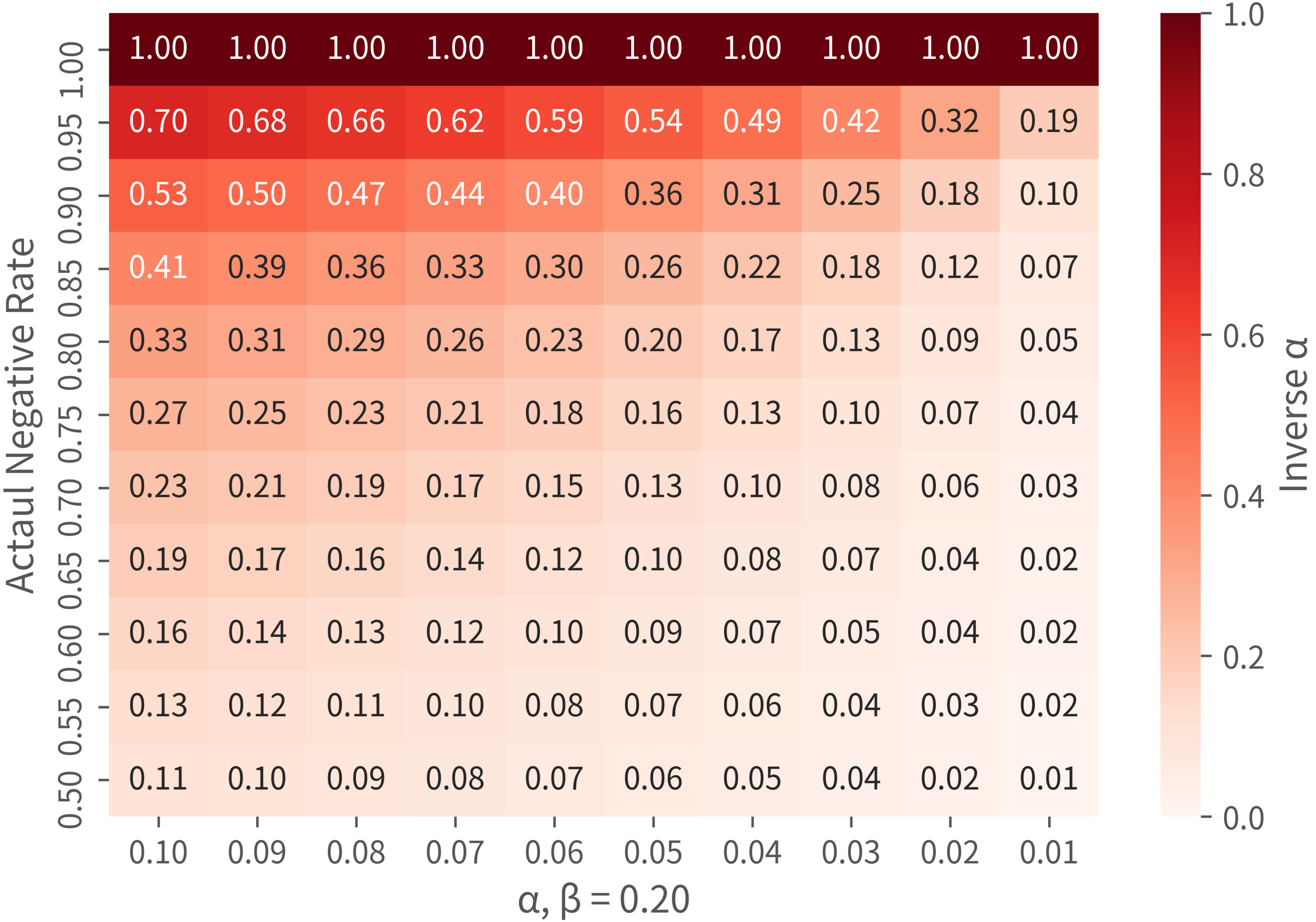
Theory

Inverse α = P(AB|BD) = B/BD

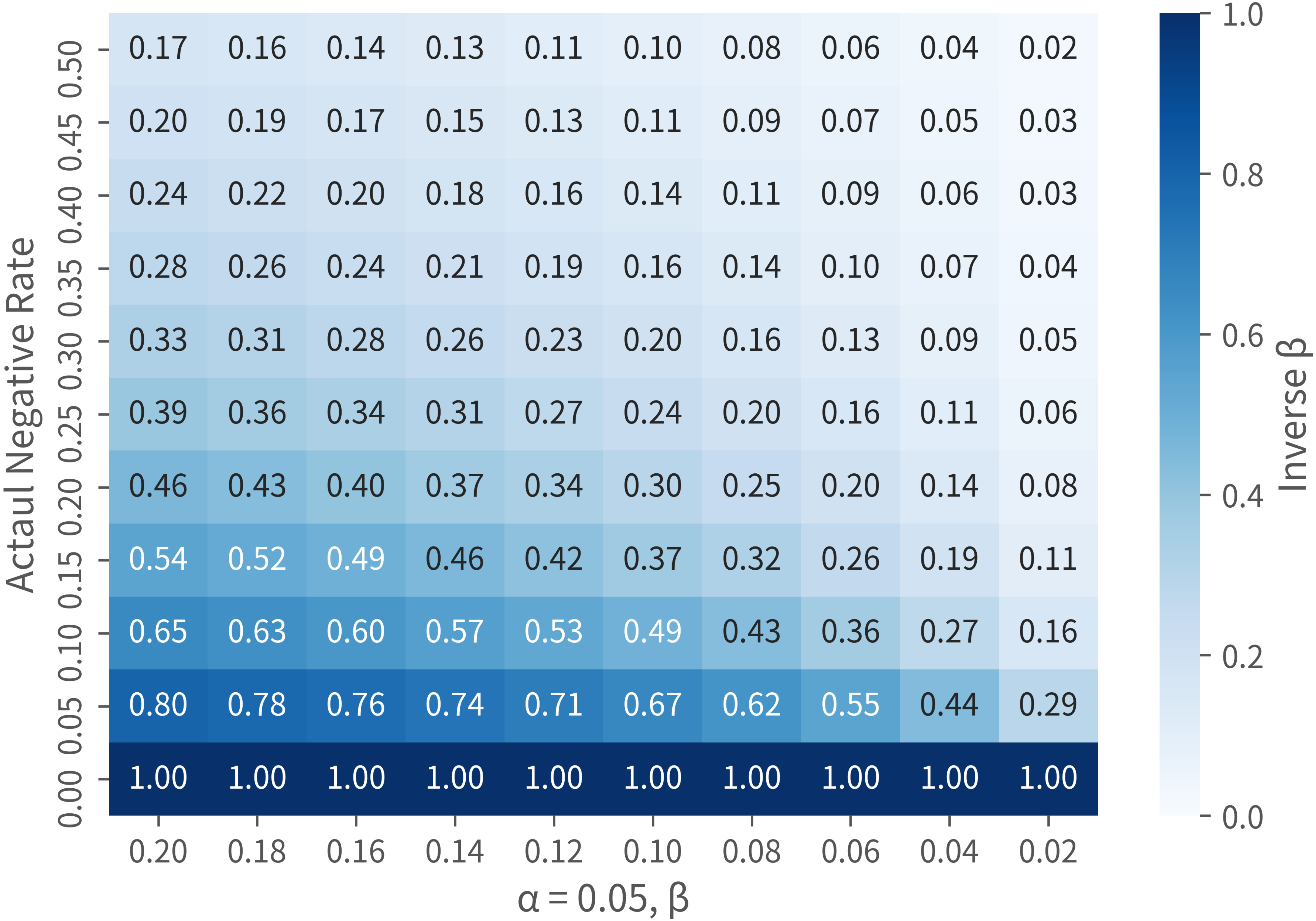
	predicted negative AC	predicted positive BD
actual negative AB	true negative A	false positive B
actual positive CD	false negative C	true positive D

Inverse β = P(CD|AC) = C/AC

	predicted negative AC	predicted positive BD
actual negative AB	true negative A	false positive B
actual positive CD	false negative C	true positive D



Actaul Negative Rate
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
0.70 0.68 0.66 0.62 0.59 0.54 0.49 0.42 0.32 0.19
0.53 0.50 0.47 0.44 0.40 0.36 0.31 0.25 0.18 0.10
0.41 0.39 0.36 0.33 0.30 0.26 0.22 0.18 0.12 0.07
0.33 0.31 0.29 0.26 0.23 0.20 0.17 0.13 0.09 0.05
0.27 0.25 0.23 0.21 0.18 0.16 0.13 0.10 0.07 0.04
0.23 0.21 0.19 0.17 0.15 0.13 0.10 0.08 0.06 0.03
0.19 0.17 0.16 0.14 0.12 0.10 0.08 0.07 0.04 0.02
0.16 0.14 0.13 0.12 0.10 0.09 0.07 0.05 0.04 0.02
0.13 0.12 0.11 0.10 0.08 0.07 0.06 0.04 0.03 0.02
0.11 0.10 0.09 0.08 0.07 0.06 0.05 0.04 0.02 0.01
0.10 0.09 0.08 0.07 0.06 0.05 0.04 0.03 0.02 0.01
α, β = 0.20
Inverse α
1.0
0.8
0.6
0.4
0.2
0.0


Actaul Negative Rate
0.50 0.45 0.40 0.35 0.30 0.25 0.20 0.15 0.10 0.05 0.00
0.17 0.16 0.14 0.13 0.11 0.10 0.08 0.06 0.04 0.02
0.20 0.19 0.17 0.15 0.13 0.11 0.09 0.07 0.05 0.03
0.24 0.22 0.20 0.18 0.16 0.14 0.11 0.09 0.06 0.03
0.28 0.26 0.24 0.21 0.19 0.16 0.14 0.10 0.07 0.04
0.33 0.31 0.28 0.26 0.23 0.20 0.16 0.13 0.09 0.05
0.39 0.36 0.34 0.31 0.27 0.24 0.20 0.16 0.11 0.06
0.46 0.43 0.40 0.37 0.34 0.30 0.25 0.20 0.14 0.08
0.54 0.52 0.49 0.46 0.42 0.37 0.32 0.26 0.19 0.11
0.65 0.63 0.60 0.57 0.53 0.49 0.43 0.36 0.27 0.16
0.80 0.78 0.76 0.74 0.71 0.67 0.62 0.55 0.44 0.29
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
0.20 0.18 0.16 0.14 0.12 0.10 0.08 0.06 0.04 0.02
α = 0.05, β
1.0 0.8 0.6 0.4 0.2 0.0
Inverse β

Rates in predefined acceptable confusion matrix

	=	=	= predefined
α	B/AB	significance level type I error rate	false positive rate
β	C/CD	type II error rate	false negative rate
inverse α	B/BD		false discovery rate
inverse β	C/AC		false omission rate
confidence level	A/AB	1-α	specificity
power	D/CD	1-β	sensitivity recall

Rates in confusion matrix

	=	=	= observed
false positive rate	B/AB		α
false negative rate	C/CD		β
false discovery rate	B/BD		inverse α
false omission rate	C/AC		inverse β
actual negative rate	AB/ABCD		
sensitivity	D/CD	recall	power
specificity	A/AB		confidence level
precision	D/BD		inverse power
recall	D/CD	sensitivity	power

- *appendixes/theory_02_complete_a_test.ipynb*
- *appendixes/theory_03_figures.ipynb*
- That's all. 📚