What is A/B testing?

A/B TESTING IN PYTHON

Moe Lotfy, PhD

Principal Data Science Manager

Intro to A/B testing

- **An A/B test is...**
  - **an experiment designed to test which version is better**
  - **based on metric(s):** signup rate, average sales per user, etc.
  - **using random assignment and analyzing results**

To A/B test or not to test?

Good use of A/B testing:

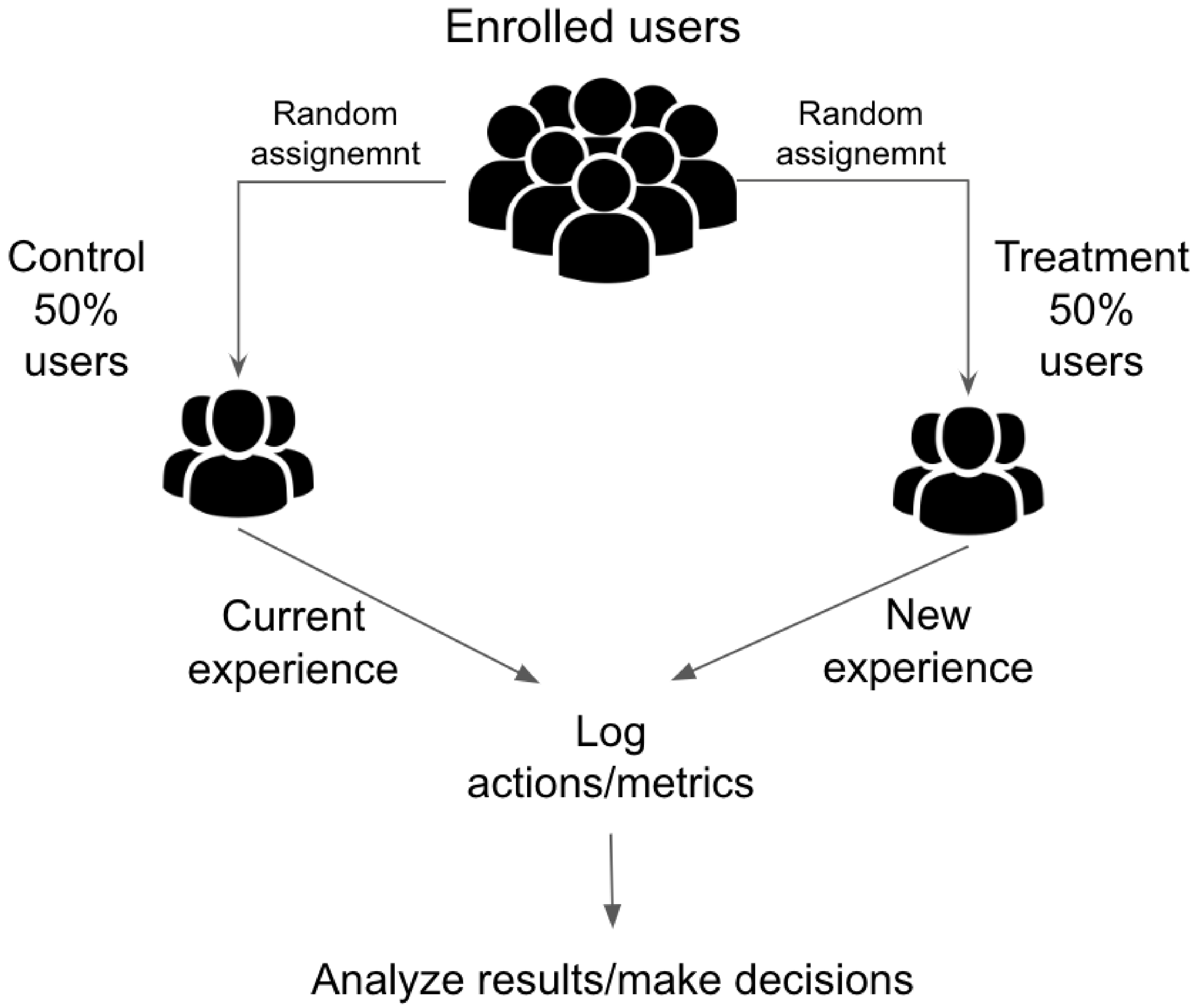
- Optimizing conversion rates
- Releasing new app features
- Evaluating incremental effects of ads
- Assessing the impact of drug trials

Do not A/B test if:

- No sufficient traffic/"small" sample size
- No clear logical hypothesis
- Ethical considerations
- High opportunity cost

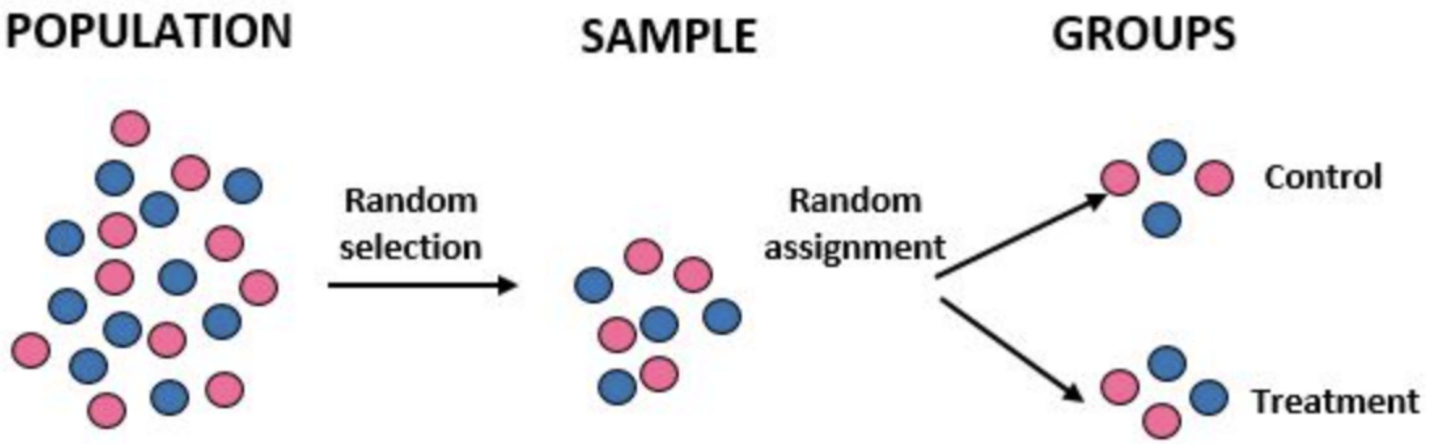
A/B testing fundamental steps

1. Specify the goal and designs/experiences
2. Randomly sample users for enrollment
3. Randomly assign users to:
    - control variant: current state
    - treatment/test variant(s): new design
4. Log user actions and compute metrics
5. Test for statistically significant differences



Value of randomization

- Generalizability and representativeness
- Minimizing bias between groups
- Establishing causality by isolating treatment effect



[1] https://www.statology.org/random-selection-vs-random-assignment/

Python example of random assignment

```
checkout.info()
```

```
RangeIndex: 9000 entries, 0 to 8999
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
 0   user_id        9000 non-null   int64
 1   checkout_page  9000 non-null   object
 2   order_value    7605 non-null   float64
 3   purchased      9000 non-null   float64
 4   gender         9000 non-null   object
 5   browser        9000 non-null   object
dtypes: float64(2), int64(1), object(3)
memory usage: 422.0+ KB
```

Python example of random assignment

```
checkout['gender'].value_counts(normalize=True)
```

```
F    0.507556
M    0.492444
Name: gender, dtype: float64
```

```
sample_df = checkout.sample(n=3000)
sample_df['gender'].value_counts(normalize=True)
```

```
M    0.506333
F    0.493667
Name: gender, dtype: float64
```

Python example of random assignment

```
checkout.groupby('checkout_page')['gender'].value_counts(normalize=True)
```

```
checkout_page  gender
A              M         0.505000
               F         0.495000
B              F         0.507333
               M         0.492667
C              F         0.520333
               M         0.479667
Name: gender, dtype: float64
```

Let's practice!

A/B TESTING IN PYTHON

Why run experiments?

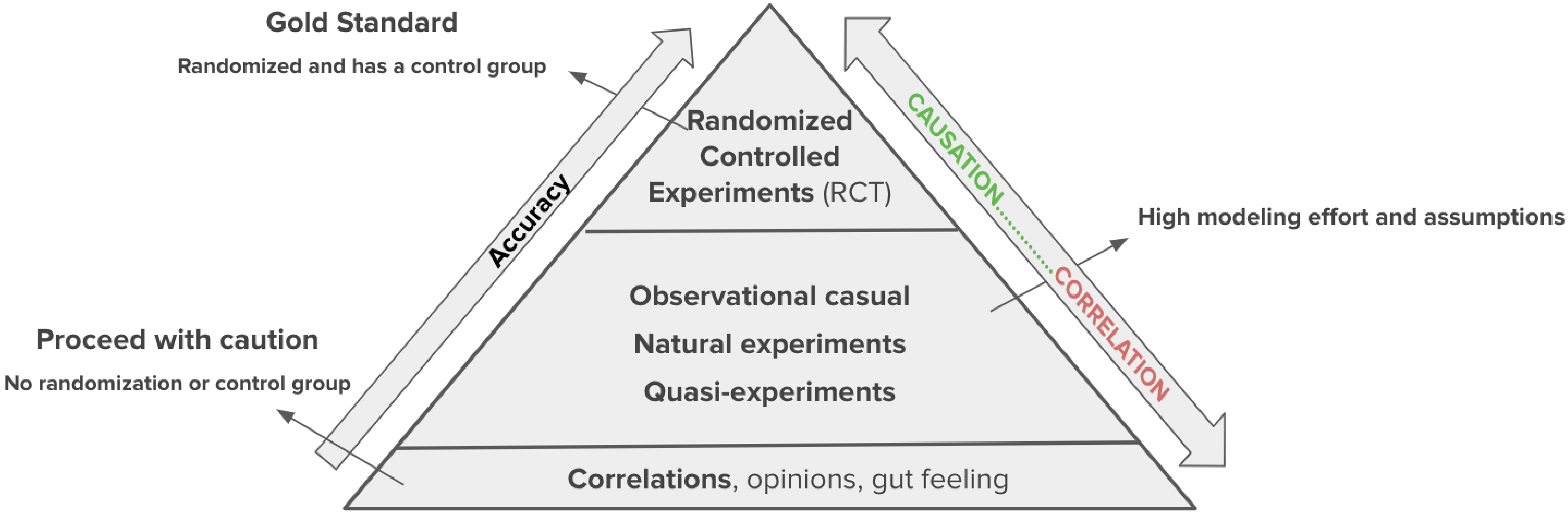
A/B TESTING IN PYTHON

Moe Lotfy, PhD
Principal Data Science Manager

The value of A/B testing

- Reduce uncertainty around the impact of new designs and features
- Decision-making --> scientific, evidence-based - not intuition
- Generous value for the investment: simple changes lead to major wins
- Continuous optimization at the mature stage of the business
- Correlation does not imply causation

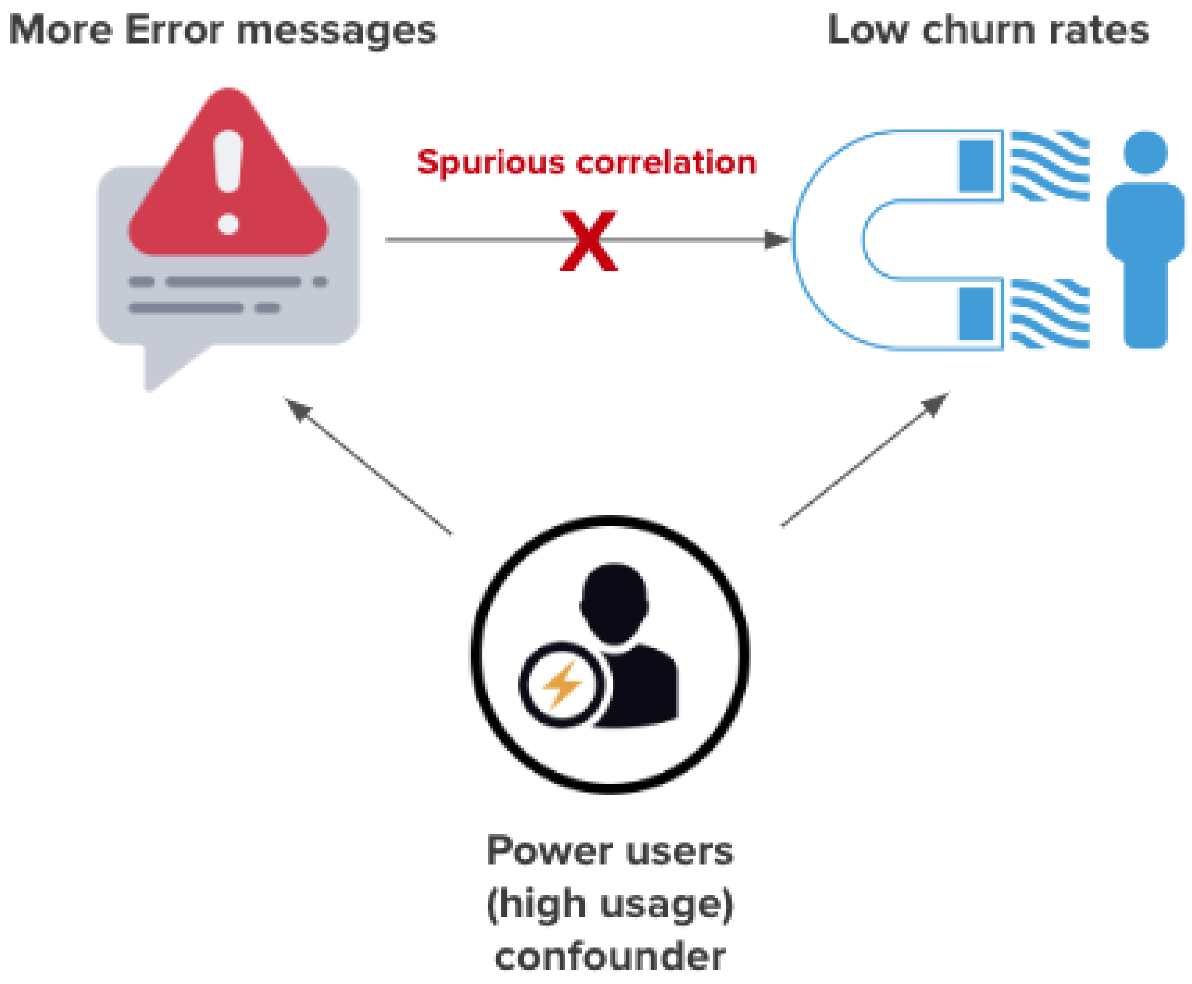
Hierarchy of evidence



[1] https://jamanetwork.com/journals/jama/article-abstract/392650

Do error messages reduce churn?

- Microsoft Office 365 spurious correlation example:[1]



  - **Spurious correlation:** a strong correlation that appears to be causal but is not.

[1] Kohavi, Ron,Tang, Diane,Xu, Ya. Trustworthy Online Controlled Experiments. Cambridge University Press.

Pearson's correlation coefficient

- A score that measures the strength of a linear relationship between two variables.
- r>0: positive correlation
- r = 0: neutral correlation
- r<0: negative correlation
- **Pearson's correlation coefficient (r) formula:**

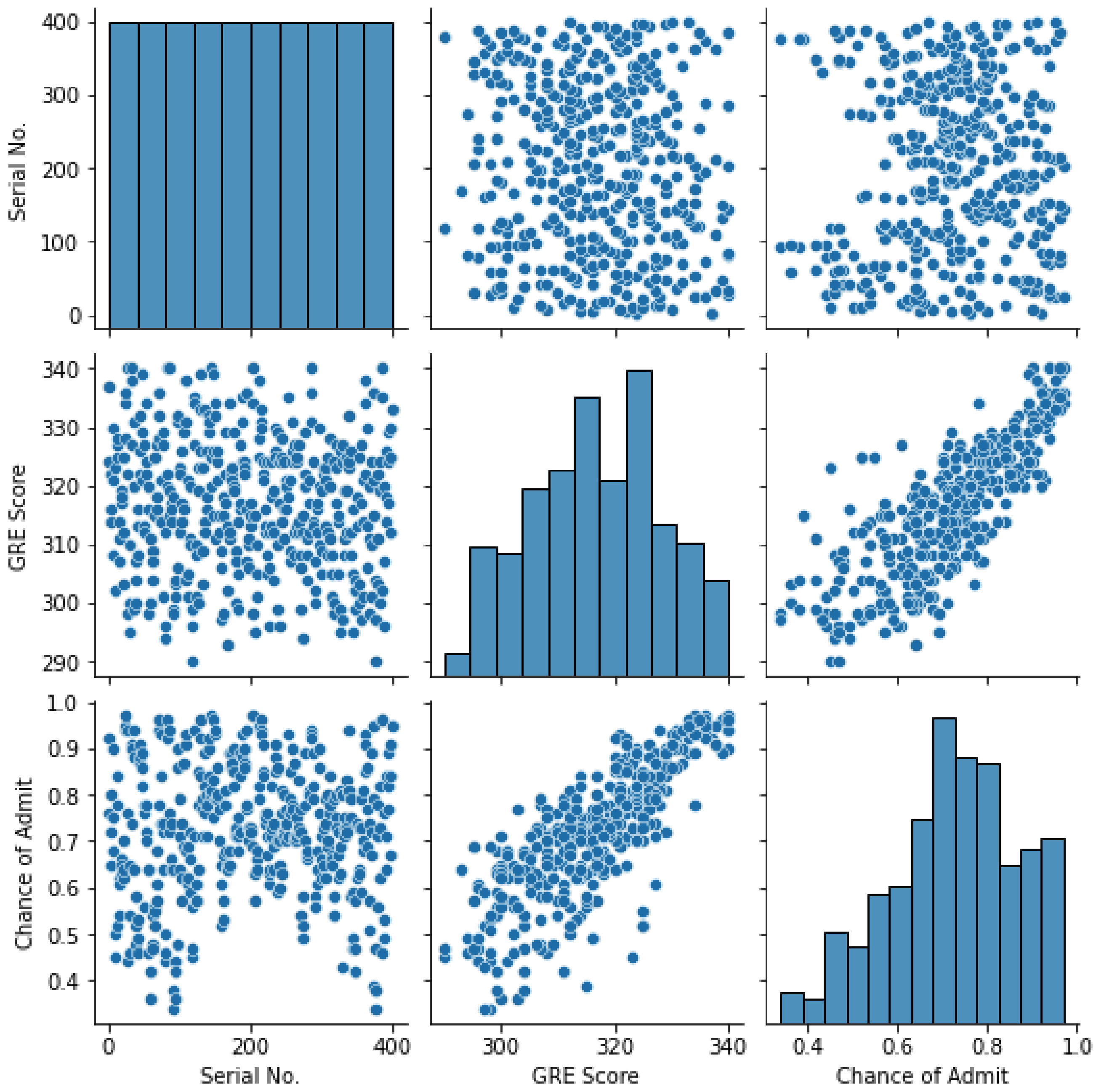
$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

- Assumes: Normal distribution and Linearity

Correlations visual inspection

```
# Import visualization library seaborn
import seaborn as sns

# Create pairplots
sns.pairplot(admissions[['Serial No.',\
      'GRE Score', 'Chance of Admit']])
```



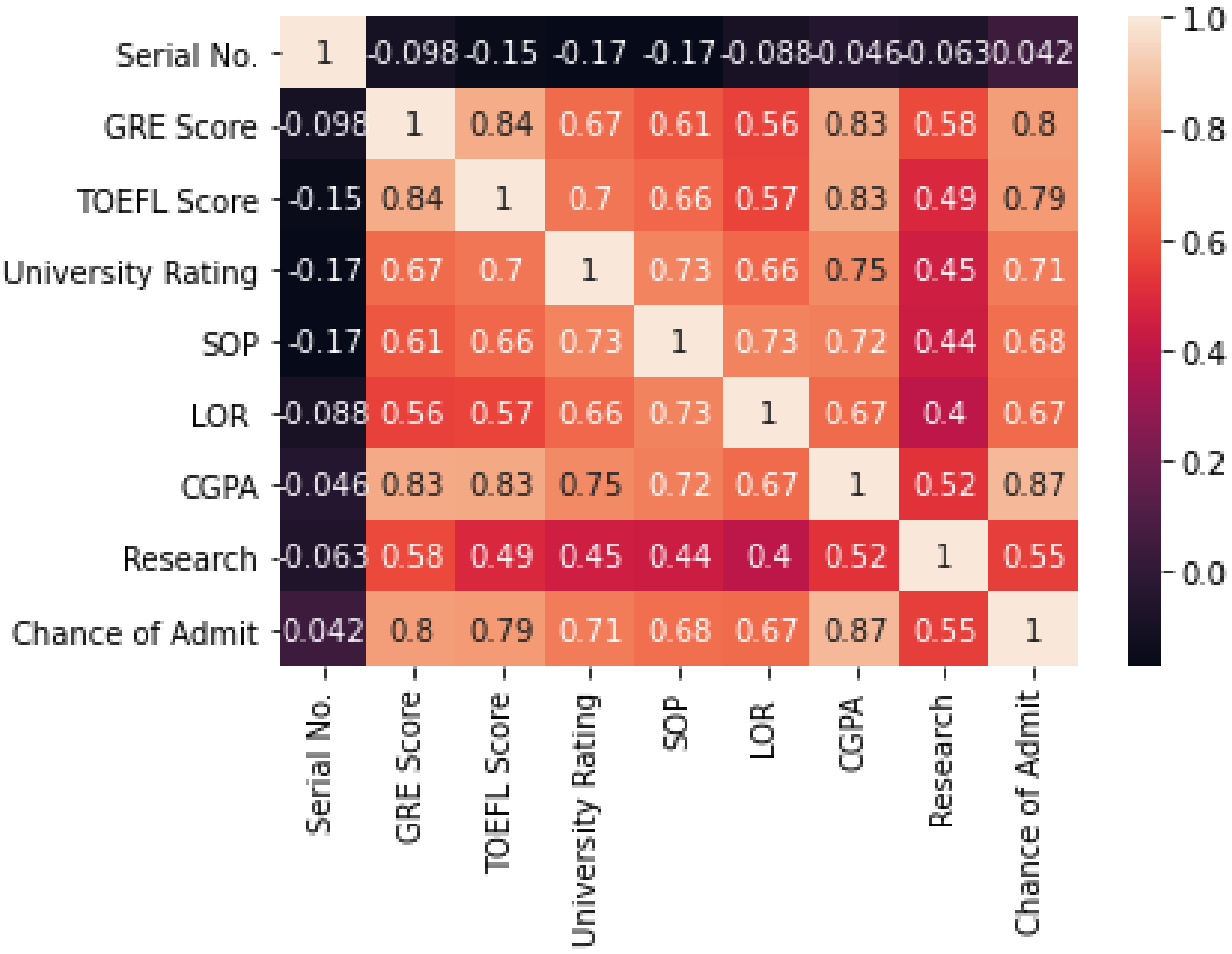
Pearson correlation heatmap

```python
# Import visualization library seaborn
import seaborn as sns

# Print Pearson correlation coefficient
print(admissions['GRE Score']\
    .corr(admissions['Chance of Admit']))
```

```
0.802610459590350З
```

```python
# Plot correlations heatmap
sns.heatmap(admissions.corr(),annot=True)
```



Let's practice!

A/B TESTING IN PYTHON

Metrics design and estimation

A/B TESTING IN PYTHON

Moe Lotfy, PhD
Principal Data Science Manager

Types of metrics

- **Primary (goal/north-star):**
  - Best describes the success of the business or mission
- **Granular metrics:**
  - Best explain users' behavior
  - More sensitive and actionable
  - Signup rate:
    - = (clicks/visitors) X (signups/clicks)
- **Instrumentation/guardrail metrics:**
  - Outside the scope of this course

- Signup rate
- Daily active users
- Average sales per user
- Average listening time per user

Types of metrics

Quantitative categorization

- **Means/percentiles:** average sales, median time on page
- **Proportions:**
  - Signup rate: signups/total visitors
  - Page abandonment rate: page abandoners/total visitors
- **Ratios:**
  - Click-through-rate(CTR): clicks/page visits or clicks/ad impressions
  - Revenue per session
- **Metrics can be combined to form a more comprehensive success/failure criteria**

Metrics requirements

- **Stable/robust** against the unimportant differences
- **Sensitive** to the important changes
- **Measurable** within logging limitations
- **Non-gameable**
  - Bright colors
  - Time on page

Python metrics estimation

```
checkout.groupby('gender')['purchased'].mean()
```

```
gender
F    0.908056
M    0.780009
Name: purchased, dtype: float64
```

```
checkout[(checkout['browser']=='chrome')|(checkout['browser']=='safari')]\
    .groupby('gender')['order_value'].mean()
```

```
gender
F    29.814161
M    30.383431
Name: order_value, dtype: float64
```

Python metrics estimation

```
checkout.groupby('browser')[['order_value', 'purchased']].mean()
```

```
         order_value  purchased
browser
chrome     30.016625   0.839088
firefox    29.887491   0.851725
safari     30.119808   0.844337
```

Let's practice!

A/B TESTING IN PYTHON