Secure Systems Engineering and Management

A Data-driven Approach

Security analytics: Intro to data analysis

Hypothesis Testing, Effect Sizes, and Regression

Michael Hicks
UPenn CIS 7000-003
Spring 2026

Lecture 1: Hypothesis Testing, Non-Parametric Tests, and Effect Sizes

Lecture 1 Overview

Topics:

- Quick level set
- The logic of hypothesis testing
- Chi-square test of independence
- Student’s t-test (parametric)
- Mann-Whitney U test (non-parametric)
- Effect sizes (Cohen’s d, Vargha-Delaney A)
- Bootstrapped confidence intervals
- Common pitfalls

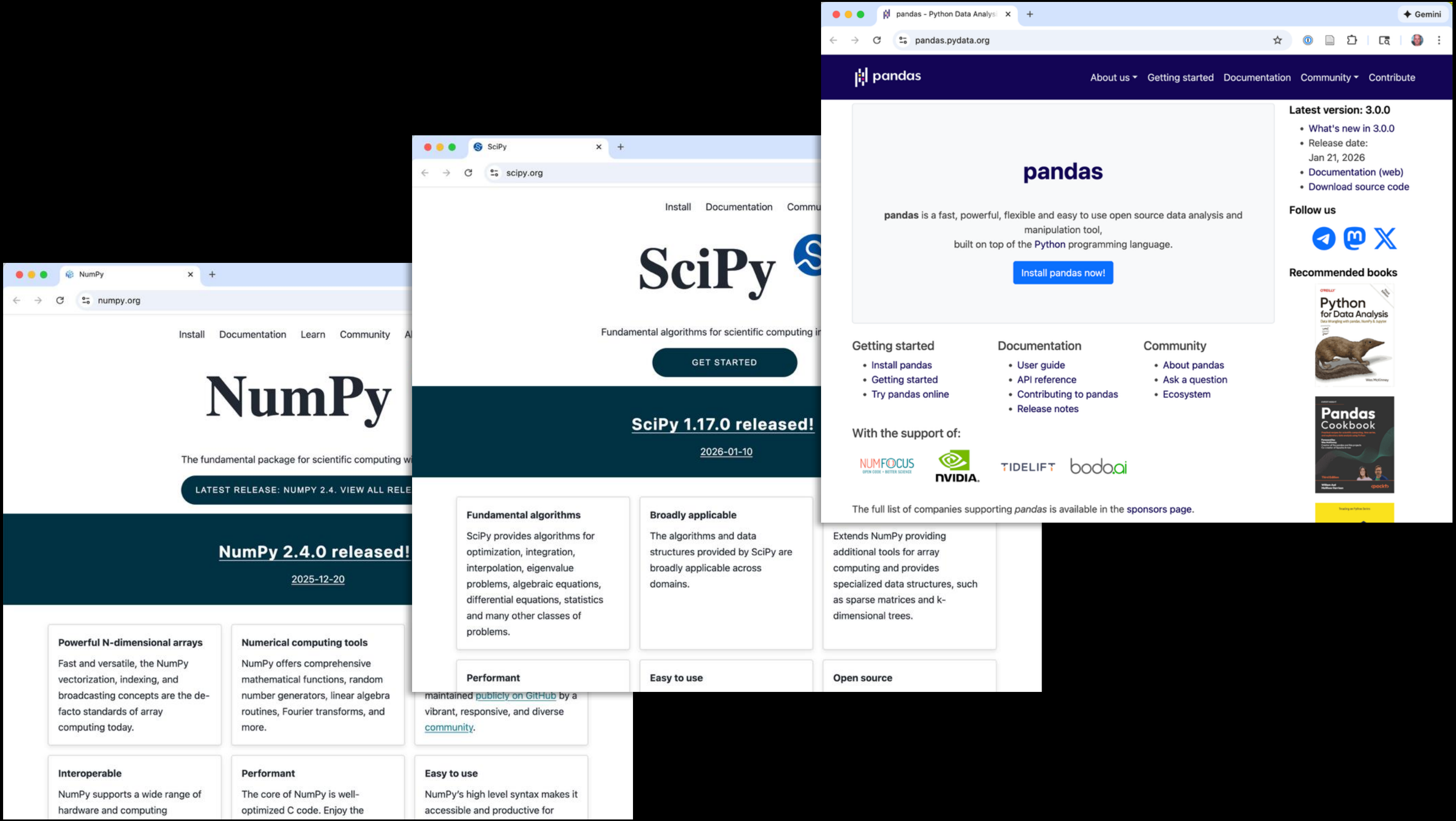
Sample Data for This Lecture

We'll use synthetic vulnerability data throughout:

```
sample_vuln_data.csv (n = 2,000)
├── cve_id              # CVE identifier
├── pub_year            # Publication year (2018-2024)
├── cwe_category        # Memory, InputValidation, Crypto, Auth, Other
├── cvss_base           # CVSS score (0-10)
├── impact              # Impact subscore
├── exploitability      # Exploitability subscore
├── severity            # Low, Medium, High, Critical
└── in_kev              # TRUE if actively exploited
```

Loading the Sample Data in R

```r
# Load the sample vulnerability data
data <- read.csv("sample_vuln_data.csv")

# Convert categorical variables to factors
data$cwe_category <- factor(data$cwe_category)
data$severity <- factor(data$severity,
                        levels = c("Low", "Medium",
"High", "Critical"),
                        ordered = TRUE)

# Quick check
str(data)
summary(data$cvss_base)
table(data$in_kev)
```



NumPy
numpy.org
Install
Documentation
Learn
Community
NumPy
The fundamental package for scientific computing wi
LATEST RELEASE: NUMPY 2.4. VIEW ALL RELE
NumPy 2.4.0 released!
2025-12-20
Powerful N-dimensional arrays
Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.
Numerical computing tools
NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.
maintained publicly on GitHub by a vibrant, responsive, and diverse community.
Interoperable
NumPy supports a wide range of hardware and computing
Performant
The core of NumPy is well-optimized C code. Enjoy the
Easy to use
NumPy's high level syntax makes it accessible and productive for
SciPy
scipy.org
Install
Documentation
Commu
SciPy
Fundamental algorithms for scientific computing i
GET STARTED
SciPy 1.17.0 released!
2026-01-10
Fundamental algorithms
SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and many other classes of problems.
Broadly applicable
The algorithms and data structures provided by SciPy are broadly applicable across domains.
Extends NumPy providing additional tools for array computing and provides specialized data structures, such as sparse matrices and k-dimensional trees.
Performant
Easy to use
Open source
pandas - Python Data Analys
Gemini
pandas.pydata.org
pandas
About us
Getting started
Documentation
Community
Contribute
pandas
pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,
built on top of the Python programming language.
Install pandas now!
Latest version: 3.0.0
What's new in 3.0.0
Release date:
Jan 21, 2026
Documentation (web)
Download source code
Follow us
Recommended books
Getting started
Install pandas
Getting started
Try pandas online
Documentation
User guide
API reference
Contributing to pandas
Release notes
Community
About pandas
Ask a question
Ecosystem
With the support of:
NUMFOCUS
NVIDIA.
TIDELIFT
bodo.ai
The full list of companies supporting pandas is available in the sponsors page.
Python for Data Analysis
Pandas Cookbook

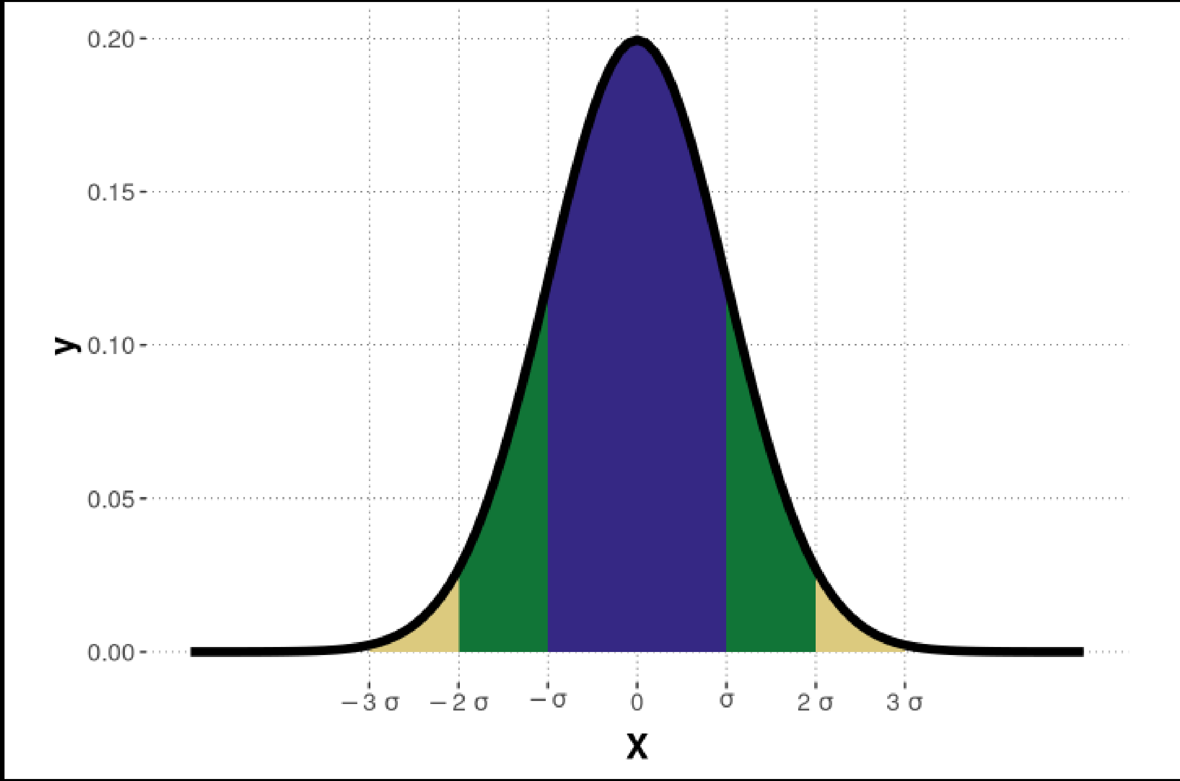
Loading the Sample Data in Python

```python
import pandas as pd
import numpy as np

# Load the sample vulnerability data
data = pd.read_csv("sample_vuln_data.csv")

# Convert to categorical (optional but good practice)
data['cwe_category'] = pd.Categorical(data['cwe_category'])
data['severity'] = pd.Categorical(
    data['severity'],
    categories=["Low", "Medium", "High", "Critical"],
    ordered=True
)

# Quick check
print(data.info())
print(data['cvss_base'].describe())
print(data['in_kev'].value_counts())
```

Level setting: Terms you know

- Empirical data
  - Observational vs. experimental
- Analysis
  - Explanation vs. prediction
- Variable
  - nominal / categorical / binary vs. ordinal vs. metric
- Distribution and sample
- Central tendency
  - Mean / average μ, median, mode
- Dispersion
  - Variance, standard deviation σ, quantiles

Our friend: The normal distribution



Part 1: Hypothesis Testing

Why Hypothesis Testing?

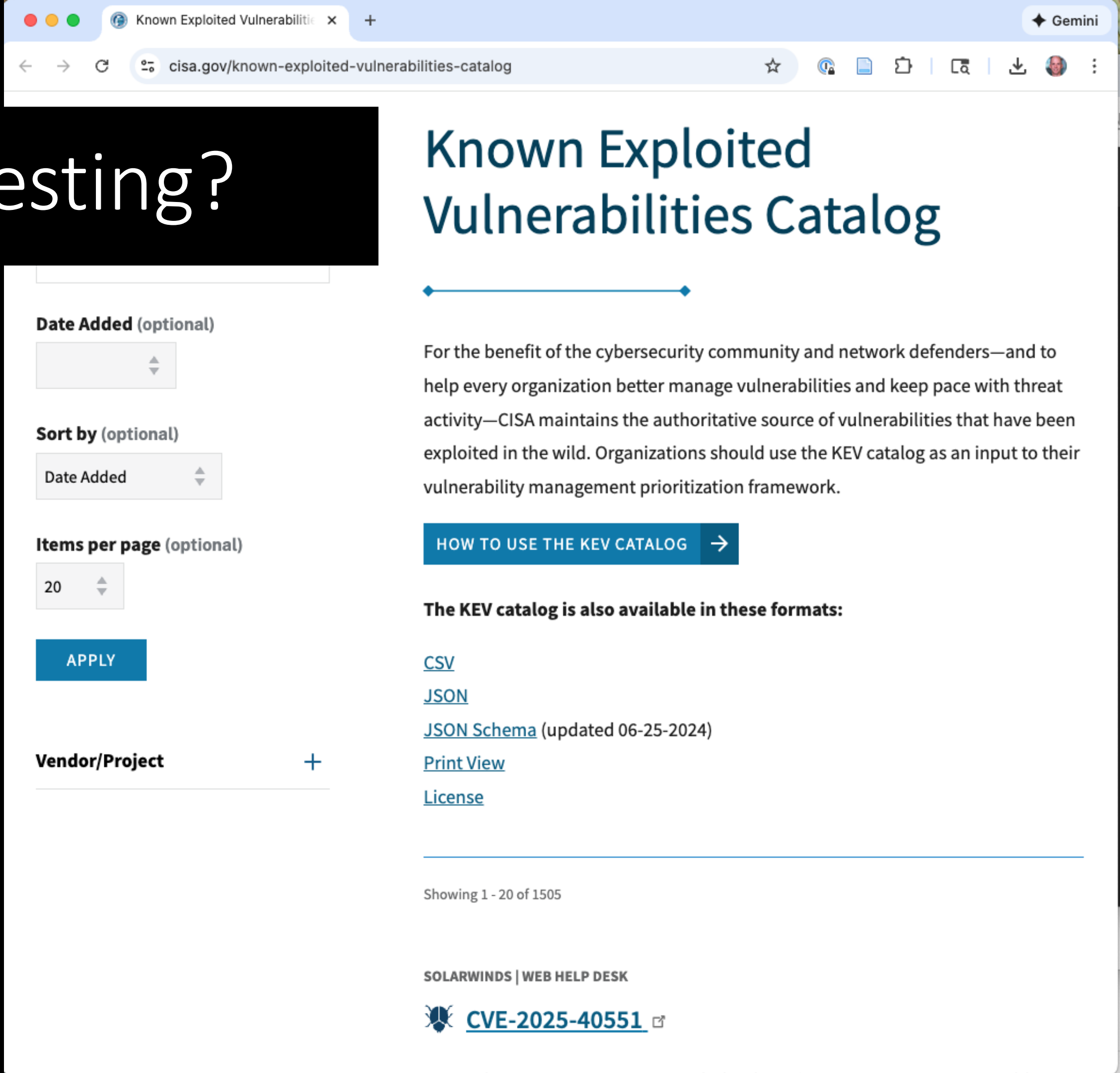
The fundamental problem

How do we distinguish signal from noise in our data?

Suppose

- Exploited vulnerabilities have mean CVSS = 6.26
- Non-exploited vulnerabilities have mean CVSS = 5.22

Is this a real difference, or just random variation?


Known Exploited Vulnerabiliti
Gemini
cisa.gov/known-exploited-vulnerabilities-catalog

Date Added (optional)

Sort by (optional)

Date Added

Items per page (optional)

20

APPLY

Vendor/Project

Known Exploited Vulnerabilities Catalog

For the benefit of the cybersecurity community and network defenders—and to help every organization better manage vulnerabilities and keep pace with threat activity—CISA maintains the authoritative source of vulnerabilities that have been exploited in the wild. Organizations should use the KEV catalog as an input to their vulnerability management prioritization framework.

HOW TO USE THE KEV CATALOG

The KEV catalog is also available in these formats:

CSV
JSON
JSON Schema (updated 06-25-2024)
Print View
License

Showing 1 - 20 of 1505

SOLARWINDS | WEB HELP DESK

CVE-2025-40551

Hypotheses

Null hypothesis (H_0): *The default assumption*

- Usually “no effect” or “no difference”
- Example: “CVSS scores are the same for exploited and non-exploited vulnerabilities”

Alternative hypothesis (H_1): *What we’re testing for*

- Example: “CVSS scores differ between exploited and non-exploited vulnerabilities”

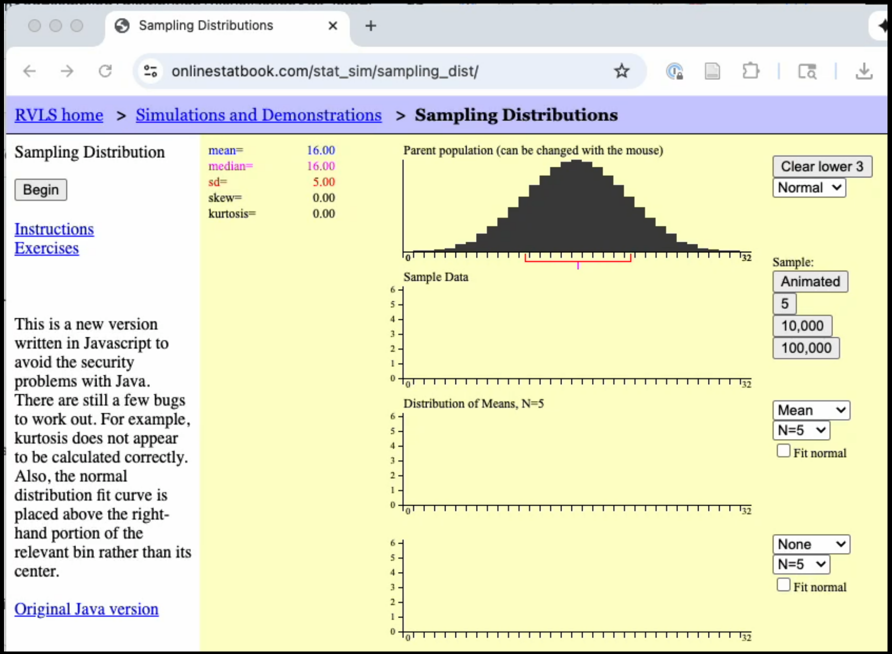
Test Statistics and Sampling Distributions

Test statistic

An assessment of our experimental data, as it relates to H_0

Sampling distribution

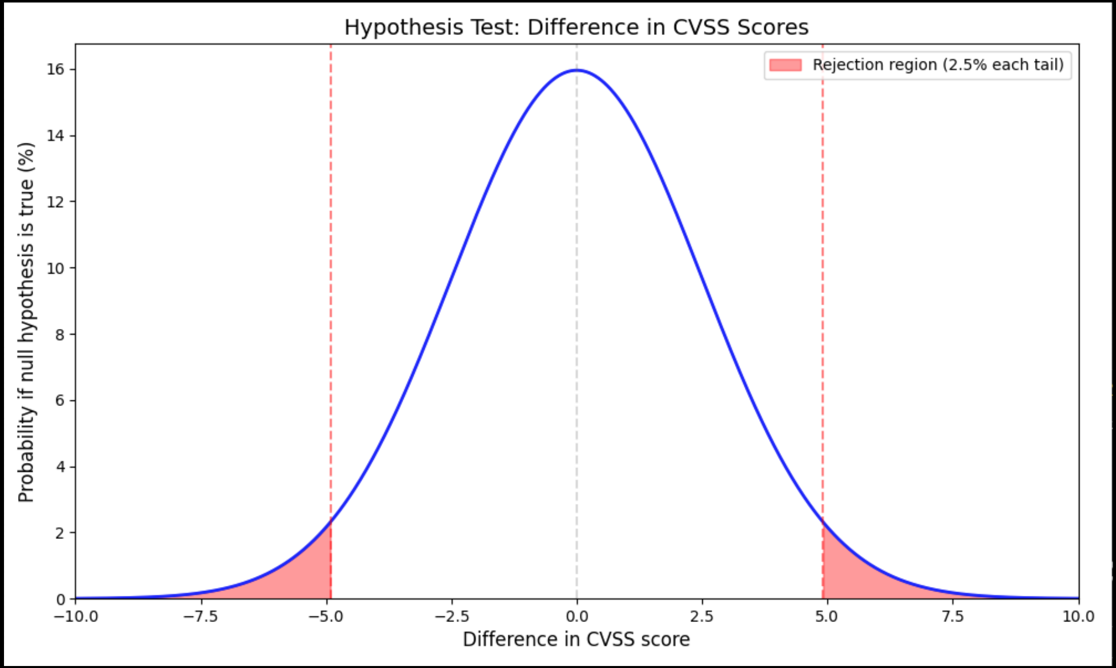
The distribution of the test statistic *if we repeated the experiment many times*



The Frequentist Framework

Core question

What would we expect to see if there were no real effect?



If the observed data would be *very unusual* under the "no effect" assumption, we have evidence against that assumption. (Note: graph is notional, not based on analysis.)

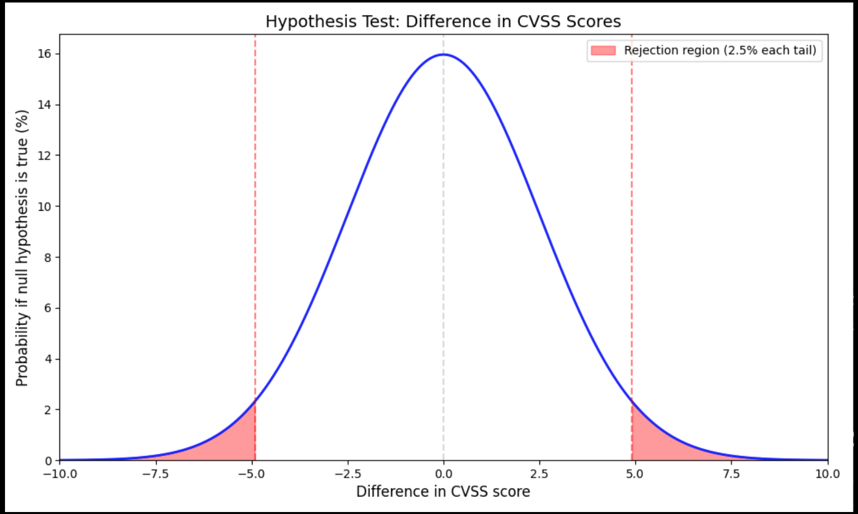
The p-value

Definition:

The probability of observing data as extreme as (or more extreme than) ours, *if H_0 were true*

Correct interpretation:

p-value = 0.05: "If there were truly no difference in CVSS scores, there is only a 5% chance of seeing a difference this large."



p-value Misinterpretations

Common mistakes

- ❌ “The probability that H_0 is true is p”
- ❌ “The probability that H_1 is true is 1 – p”
- ❌ “A significant result means the effect is large”
- ❌ “A non-significant result means there’s no effect”

Tang et al. found 26% of SOUPS papers had interpretation errors like these

Misuse, Misreporting, Misinterpretation of Statistical Methods in Usable Privacy and Security Papers

Jenny Tang
Carnegie Mellon University

Lujo Bauer
Carnegie Mellon University

Nicolas Christin
Carnegie Mellon University

Abstract

Null hypothesis significance testing (NHST) is commonly used in quantitative usable privacy and security studies. Many papers use results from statistical tests to assert whether effects or differences exist depending on the resulting *p*-value. We conduct a systematic review of papers published in 10 editions of the Symposium on Usable Privacy and Security over a span of 20 years to evaluate the field’s use of NHST. We code statistical tests for potential statistical validity, reporting, or interpretation issues that may undermine assertions made in the 121 papers that use NHST. Most problematically, tests in 23% of papers inadequately account for non-independence between samples, leading to potentially invalid claims. 58% of papers lack information to verify whether an assertion is supported, such as imprecisely specifying the statistical test conducted. Many papers contain more minor statistical issues or report statistics in ways that deviate from best practice. We conclude with recommendations for statistical reporting and statistical thinking in the field.

1 Introduction

Statistical methods are often used in human-computer interaction research to support assertions about the presence (or absence) of an effect of scientific significance (e.g., some magnitude of difference) accompanied by a measure of statistic significance. Indeed, one of the most common refrains in statistical analysis is that a result is significant because the “*p*-value” is less than a given threshold, e.g., $p < 0.05$. Despite over half a century of criticism, null hypothesis significance testing (NHST, also known as statistical significance testing)—that is, methods using *p*-values from inferential statistical tests as evidence to reject a null hypothesis—remains the dominant form of statistical analysis and evaluation [17]. However, simply dichotimizing results into “significant” and “non-significant” through their associated *p*-values without reporting other information is not in itself sufficient to convey the scientific importance of the claims, nor the richness and complexity of data collected from human subjects. This reliance on *p*-values to support assertions sometimes leads other information vital to understanding statistical and scientific significance to be omitted.

As a result, complete reliance on *p*-values is increasingly frowned upon, with some journals banning the reporting of *p*-values altogether [75, 81]. Most other current guidance is less drastic, and recommends using statistical hypothesis testing as a starting point and providing sufficient context (such as effect sizes, confidence intervals, and underlying data) to convey the scientific significance of the claims [2, 13, 49, 59, 80, 81]. We use this guidance to evaluate whether the scientific assertions made on the basis of NHST in usable privacy and security (UPS) are accompanied by sufficient reporting for readers to validate whether these assertions are supported by the information present in the paper. We focus on UPS as it is still a fairly young area, with evolving standards, features a considerable amount of quantitative research, and errors or misinterpretations can be detrimental to user safety in the digital world and beyond.

Prior work has also examined the transparency, reporting, and validity of statistical methods in HCI and various subfields [16, 25, 36, 51, 62, 66, 77]. However, the evaluations in these works typically focus on evaluating whether *p*-values are accurately computed or on whether there may be false negatives (such as due to lack of power) or false positives (such as from inaccurately reported *p*-values).

In this work, we look beyond statistical significance to examine statistical validity (whether the chosen test is suitable for the data or whether it may produce spurious results), reporting transparency and completeness (whether the reported

USENIX Symposium on Usable Privacy and Security (SOUPS) 2025.
August 10–12, 2025, Seattle, WA, United States.

Significance Threshold (α)

Convention: $\alpha = 0.05$

What this means

- We reject H_0 if $p < \alpha$
- We accept a 5% risk of *false positives*

Type I error: Rejecting H_0 when it's actually true (*false positive*)

Type II error: Failing to reject H_0 when it's actually false (*false negative*)

Part 2: Chi-Square Test of Independence

When to Use Chi-Square

Purpose: Test (non)independence of two categorical variables

Example: Is vulnerability **severity category** (Low/Medium/High/Critical) independent of **CWE category** (Memory/Crypto/Input Validation/…)?

	Memory	Crypto	Input Val
Low	?	?	?
Medium	?	?	?
High	?	?	?
Critical	?	?	?

Building a Contingency Table

Observed counts

	Memory	Crypto	Input Val	Row Total
Low	?	?	?	245
Medium	?	?	?	350
High	?	?	?	355
Critical	?	?	?	200
Col Total	350	400	400	**1150**

Expected Counts Under Independence

If no association exists (i.e., *independent*)

Expected count = (Row total × Column total) / Grand total

E(Low, Memory) = **(245 / 1150)** × 350 = 74.6

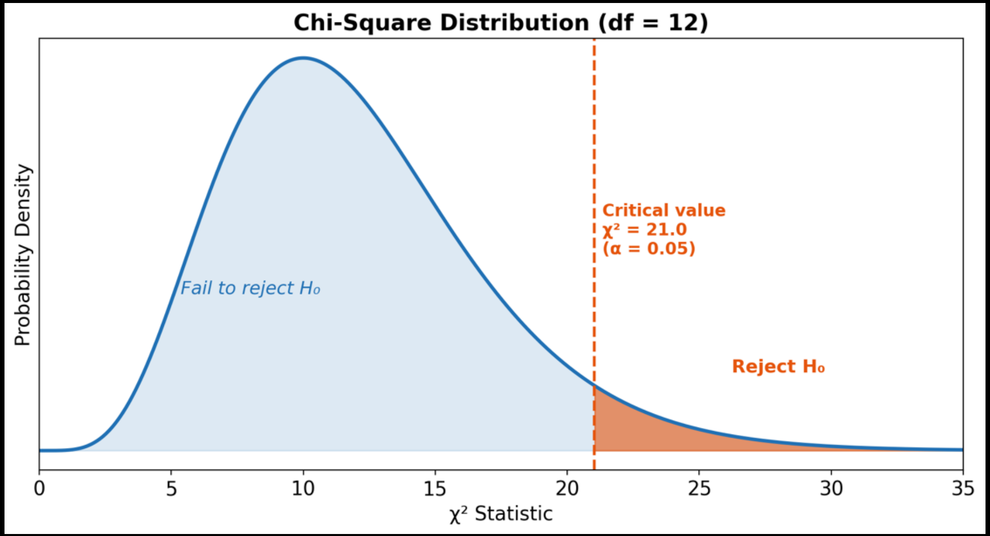
	Memory	Crypto	Input Val	Row Total
Low	45	120	80	245
Medium	90	150	110	350
High	130	85	140	355
Critical	85	45	70	200
Col Total	350	400	400	**1150**

Under independence, we'd expect ~75, but we observed only 45.

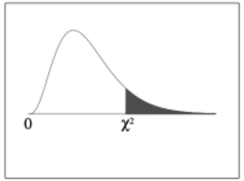
The Chi-Square Test Statistic

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

- Sum over all cells in the table
- Large differences between O and E → large χ^2
- Compare to χ^2 distribution with df = (rows − 1)(cols − 1)



Chi-Square Distribution Table



The shaded area is equal to α for $\chi^2 = \chi^2_\alpha$.

df	$\chi^2_{.995}$	$\chi^2_{.990}$	$\chi^2_{.975}$	$\chi^2_{.950}$	$\chi^2_{.900}$	$\chi^2_{.100}$	$\chi^2_{.050}$	$\chi^2_{.025}$	$\chi^2_{.010}$	$\chi^2_{.005}$
1	0.000	0.000	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401
22	8.643	9.542	10.982	12.338	14.041	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.559
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.879	14.573	16.151	18.114	36.741	40.113	43.195	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.121	14.256	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.336
30	13.787	14.953	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169

Chi-Square in Python

```
from scipy.stats import chi2_contingency

# Create contingency table
cont_table = pd.crosstab(data['severity'], data['cwe_category'])

# Run chi-square test
chi2, p_value, dof, expected = chi2_contingency(cont_table)

# Print results
print(f"χ² = {chi2:.2f}, df = {dof}, p = {p_value:.4f}")

# Calculate standardized residuals manually
std_residuals = (cont_table - expected) / np.sqrt(expected)
print(std_residuals)
```

Running Chi-square on Sample Data

Expected output on sample_vuln_data.csv:

```
χ² = 110.53, df = 12, p = 0.000000000000000
```

The chi-square test examines whether severity and CWE category are independent.

- **Null hypothesis**: Severity distribution identical for all CWE categories
- **Result**: We reject H_0 ($p < 0.001$) — there is a significant association

Interpreting Results: Where Is the Association?

A significant χ^2 tells you *that* there's an association, not *where*.

Standardized residuals: $(O - E) / \sqrt{E}$

Interpretation	Meaning
Residual > +2	More than expected (overrepresented)
Residual < −2	Fewer than expected (underrepresented)

Sample data associations

Pattern	Residual	Meaning
Memory + High	+5.37	Far **more high-severity memory bugs** than expected
Memory + Low	-4.73	Far **fewer low-severity memory bugs** than expected
Other + Low	+4.44	**More low-severity “Other” bugs** than expected
Crypto + Low	+3.42	**More low-severity crypto bugs** than expected

Chi-Square Assumptions

1. **Independence:** Each observation is independent
2. **Expected count rule:** Most cells should have $E \geq 5$
3. **Categorical data:** Both variables must be categorical

⚠️ **Warning for large samples:**

With thousands of vulnerabilities, even trivial associations are “significant”

→ Always report effect sizes!

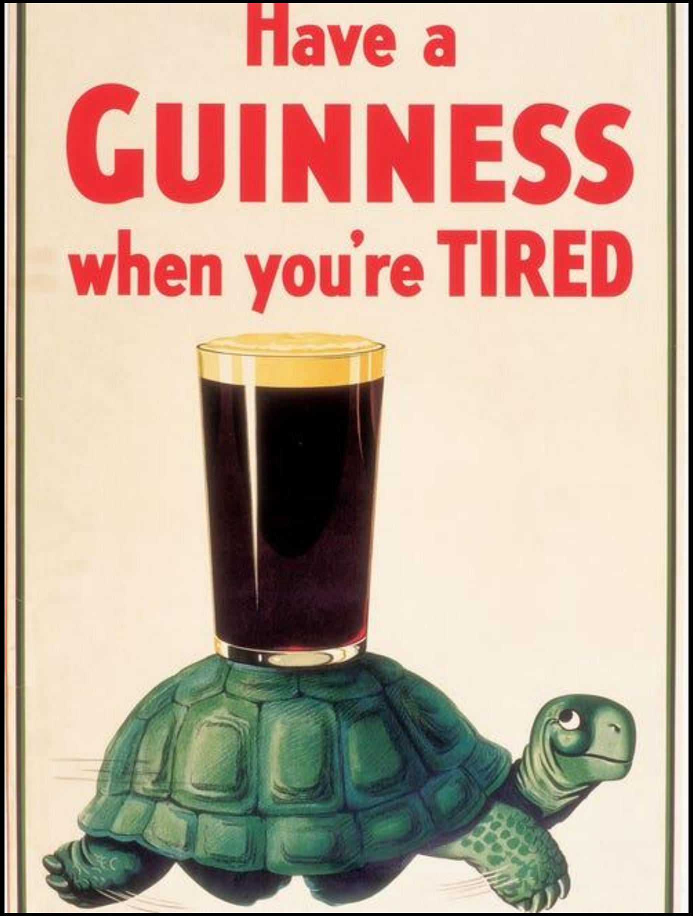
Part 3: Comparing Two Groups — t-test and Mann-Whitney U

The Student's t-Test

Purpose: Test whether the means of two groups differ significantly

The question we're asking:
Do exploited vulnerabilities have different CVSS scores (on average) than non-exploited vulnerabilities?

H_0: μ_exploited = μ_not_exploited
H_1: μ_exploited ≠ μ_not_exploited



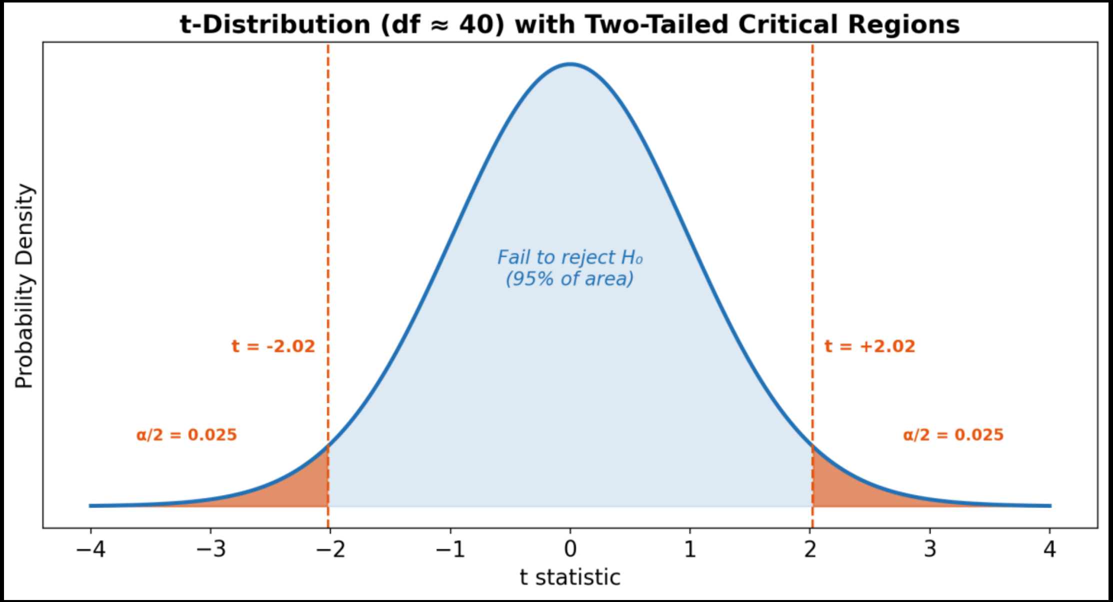
t-Test: How It Works

Test statistic:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{SE_{difference}}$$

where SE depends on the pooled standard deviation and sample sizes

Under H_0: t follows a t-distribution with df ≈ $n_1 + n_2 - 2$



t-Test Assumptions

1. **Independence:** Observations are independent
2. **Normality:** Data in each group is normally distributed
3. **Equal variance:** Both groups have similar variance (for standard t-test)

How important are these?

- Independence: **Critical** — violations cause serious problems
- Normality: Less critical with large samples (Central Limit Theorem)
- Equal variance: Use Welch's t-test to relax this assumption

t-Test in Python

```
from scipy.stats import ttest_ind

# Separate CVSS scores by exploitation status
exploited = data[data['in_kev'] == True]['cvss_base']
not_exploited = data[data['in_kev'] == False]['cvss_base']

# Welch's t-test (equal_var=False is safer)
t_stat, p_value = ttest_ind(exploited, not_exploited,
equal_var=False)

# View results
print(f"t = {t_stat:.3f}, p = {p_value:.4f}")
print(f"Mean (exploited): {exploited.mean():.2f}")
print(f"Mean (not exploited): {not_exploited.mean():.2f}")
print(f"Difference: {exploited.mean() -
not_exploited.mean():.2f}")
```

Running t-Test on Sample Data

Expected output on sample_vuln_data.csv:

```
t = 5.098, p = 0.00001
Mean (exploited): 6.26
Mean (not exploited): 5.22
Difference: 1.04
```

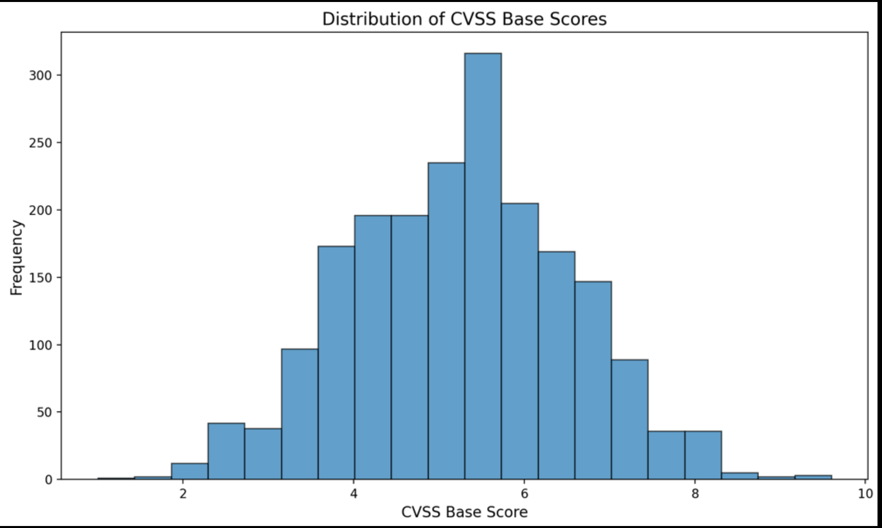
Interpretation:

Exploited vulnerabilities have significantly higher CVSS scores (M = 6.26) than non-exploited ones (M = 5.22), $t(37.5) = 5.098$, $p < 0.00001$.

Checking Normality

Visual checks:

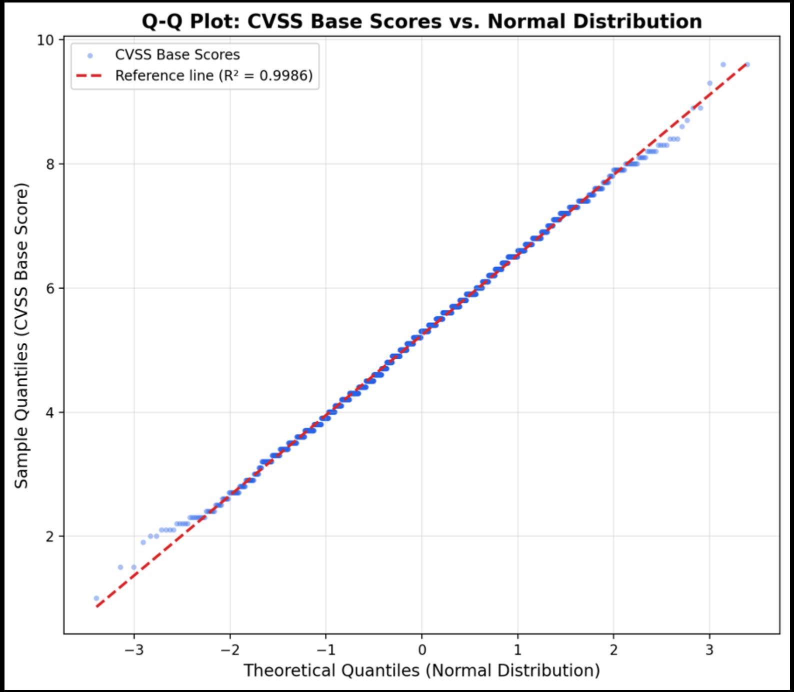
- Histogram — is it roughly bell-shaped?



Checking Normality

Visual checks:

- Histogram — is it roughly bell-shaped?
- Q-Q plot — do points follow the diagonal?

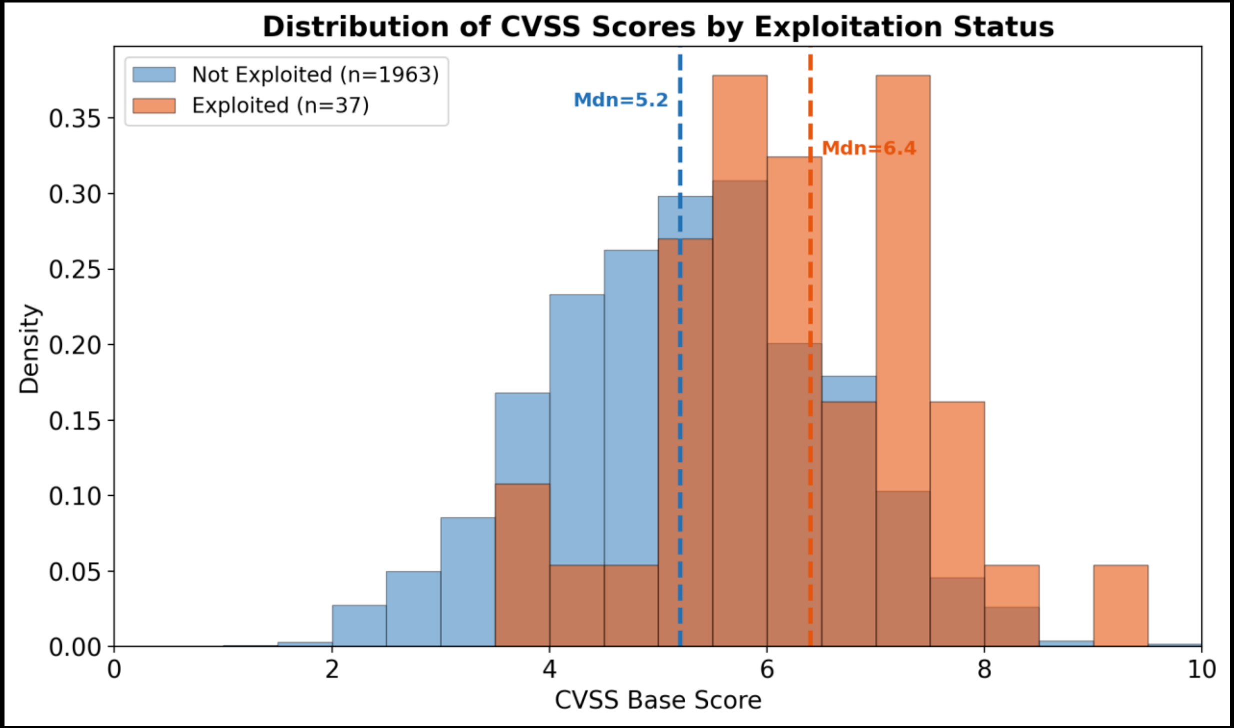


Statistical tests: Shapiro-Wilk test (but sensitive with large n)

The Shapiro-Wilk test shows:

- Statistic: 0.9985 (suggests normality)
- p-value: 0.062 (rejects NH – appear normal)

Data visualized, according to KEV status



Parametric vs. Non-Parametric Tests

Criterion	Parametric (t-test)
Assumption	Normal distribution (or n is large)
Data type	Continuous, interval
Sensitivity	More powerful if assumptions met
Measures	Compares means

Use t test when you can, Mann-Whitney when you must

Mann-Whitney U Test

Also called: Wilcoxon rank-sum test

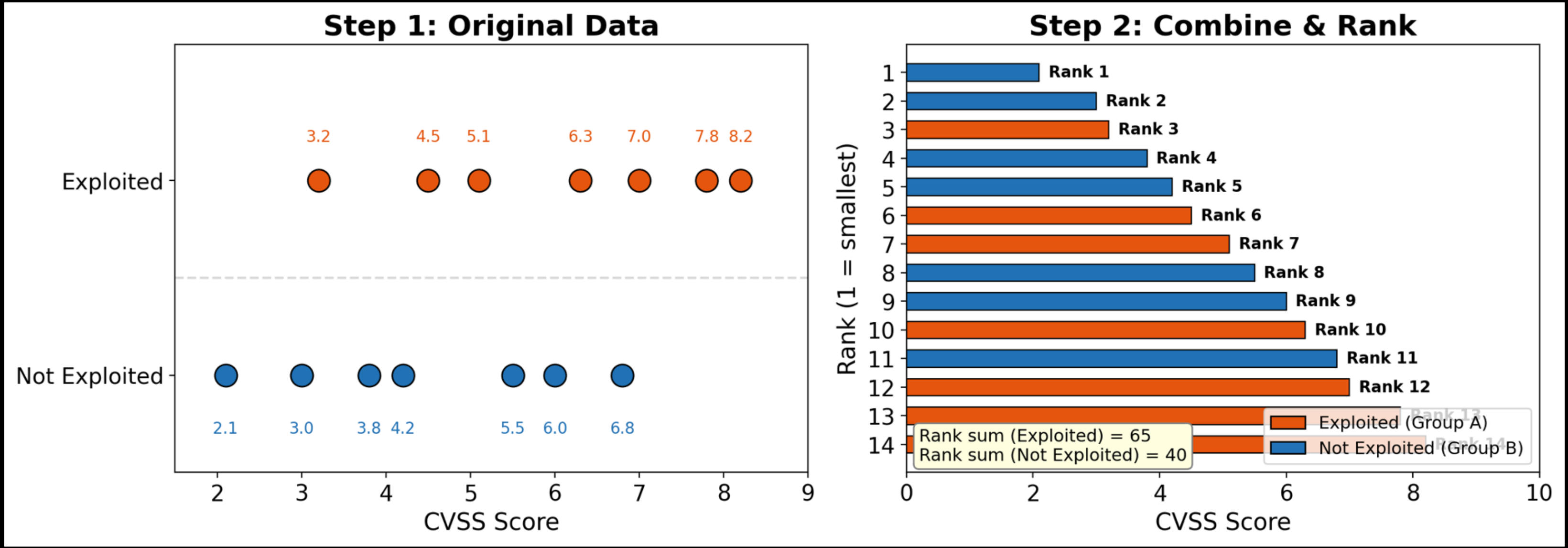
Purpose: Test whether one group tends to have larger values than another

How it works:

1. Combine both samples and rank all values (1 = smallest)
2. Sum the ranks for each group
3. The U statistic measures overlap between groups

H_0: The distributions are identical
H_1: One group tends to have larger values

Mann-Whitney: Visual Intuition



High overlap → U statistic near expected value → large p

Low overlap → U statistic far from expected → small p

Mann-Whitney in Python

```python
from scipy.stats import mannwhitneyu

# Separate groups
exploited = data[data['in_kev'] == True]['cvss_base']
not_exploited = data[data['in_kev'] == False]['cvss_base']

# Mann-Whitney U test
u_stat, p_value = mannwhitneyu(exploited, not_exploited,
                               alternative='two-sided')

# View results
print(f"U = {u_stat:.0f}, p = {p_value:.4f}")
print(f"Median (exploited): {exploited.median():.2f}")
print(f"Median (not exploited):
{not_exploited.median():.2f}")
```

Running Both Tests on Sample Data

t-test:

```
t = 5.098, p < 0.00001
Mean difference = 1.04
```

Mann-Whitney U:

```
U = 52580, p < 0.000003
Median (exploited) = 6.40, Median (not exploited) = 5.2
```

Both agree: Strong evidence that exploited vulnerabilities have higher CVSS scores

⚠️ Pitfall: Unclear Test Specification

Ambiguous: “We used a Wilcoxon test”

This could mean:

- **Mann-Whitney U** (Wilcoxon rank-sum) — independent samples
- **Wilcoxon signed-rank** — paired samples

Clear: “We used a Mann-Whitney U test (Wilcoxon rank-sum) to compare CVSS scores between exploited and non-exploited vulnerabilities.”

Part 4: Effect Sizes

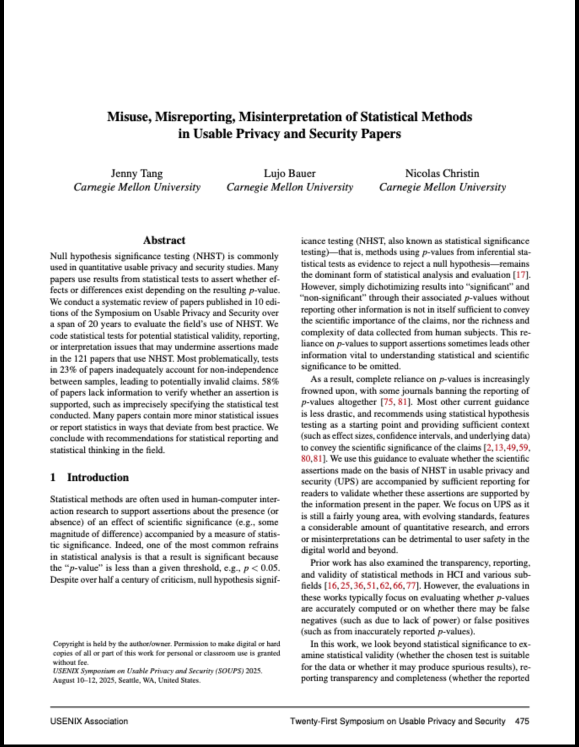
Why p-values Are Not Enough

The problem: With large samples, even trivial differences become “significant”

Example: With 200,000+ CVEs, a difference of **0.1 CVSS points** might suggest a $p < 0.001$ statistically significant different frequency of exploitability

Is that difference *practically meaningful* for security prioritization?

80% of papers had incomplete scientific significance reporting



Misuse, Misreporting, Misinterpretation of Statistical Methods in Usable Privacy and Security Papers

Jenny Tang
Carnegie Mellon University

Lujo Bauer
Carnegie Mellon University

Nicolas Christin
Carnegie Mellon University

Abstract

Null hypothesis significance testing (NHST) is commonly used in quantitative usable privacy and security studies. Many papers use results from statistical tests to assert whether effects or differences exist depending on the resulting *p*-value. We conduct a systematic review of papers published in 10 editions of the Symposium on Usable Privacy and Security over a span of 20 years to evaluate the field's use of NHST. We code statistical tests for potential statistical validity, reporting, or interpretation issues that may undermine assertions made in the 121 papers that use NHST. Most problematically, tests in 23% of papers inadequately account for non-independence between samples, leading to potentially invalid claims. 58% of papers lack information to verify whether an assertion is supported, such as imprecisely specifying the statistical test conducted. Many papers contain more minor statistical issues or report statistics in ways that deviate from best practice. We conclude with recommendations for statistical reporting and statistical thinking in the field.

1 Introduction

Statistical methods are often used in human-computer interaction research to support assertions about the presence (or absence) of an effect of scientific significance (e.g., some magnitude of difference) accompanied by a measure of statistic significance. Indeed, one of the most common refrains in statistical analysis is that a result is significant because the “*p*-value” is less than a given threshold, e.g., $p < 0.05$. Despite over half a century of criticism, null hypothesis significance testing (NHST, also known as statistical significance testing)—that is, methods using *p*-values from inferential statistical tests as evidence to reject a null hypothesis—remains the dominant form of statistical analysis and evaluation [17]. However, simply dichotimizing results into “significant” and “non-significant” through their associated *p*-values without reporting other information is not in itself sufficient to convey the scientific importance of the claims, nor the richness and complexity of data collected from human subjects. This reliance on *p*-values to support assertions sometimes leads other information vital to understanding statistical and scientific significance to be omitted.

As a result, complete reliance on *p*-values is increasingly frowned upon, with some journals banning the reporting of *p*-values altogether [75, 81]. Most other current guidance is less drastic, and recommends using statistical hypothesis testing as a starting point and providing sufficient context (such as effect sizes, confidence intervals, and underlying data) to convey the scientific significance of the claims [2,13,49,59,80,81]. We use this guidance to evaluate whether the scientific assertions made on the basis of NHST in usable privacy and security (UPS) are accompanied by sufficient reporting for readers to validate whether these assertions are supported by the information present in the paper. We focus on UPS as it is still a fairly young area, with evolving standards, features a considerable amount of quantitative research, and errors or misinterpretations can be detrimental to user safety in the digital world and beyond.

Prior work has also examined the transparency, reporting, and validity of statistical methods in HCI and various subfields [16,25,36,51,62,66,77]. However, the evaluations in these works typically focus on evaluating whether *p*-values are accurately computed or on whether there may be false negatives (such as due to lack of power) or false positives (such as from inaccurately reported *p*-values).

In this work, we look beyond statistical significance to examine statistical validity (whether the chosen test is suitable for the data or whether it may produce spurious results), reporting transparency and completeness (whether the reported

USENIX Symposium on Usable Privacy and Security (SOUPS) 2025.
August 10–12, 2025, Seattle, WA, United States.

Effect Sizes: The Solution

Effect size
A standardized measure of the *magnitude* of a difference or association

Two key effect sizes for comparing groups

Effect Size	Use Case
Cohen's d	Parametric (with t-test)
Vargha-Delaney A	Non-parametric (with Mann-Whitney)

Cohen’s d (Parametric Effect Size)

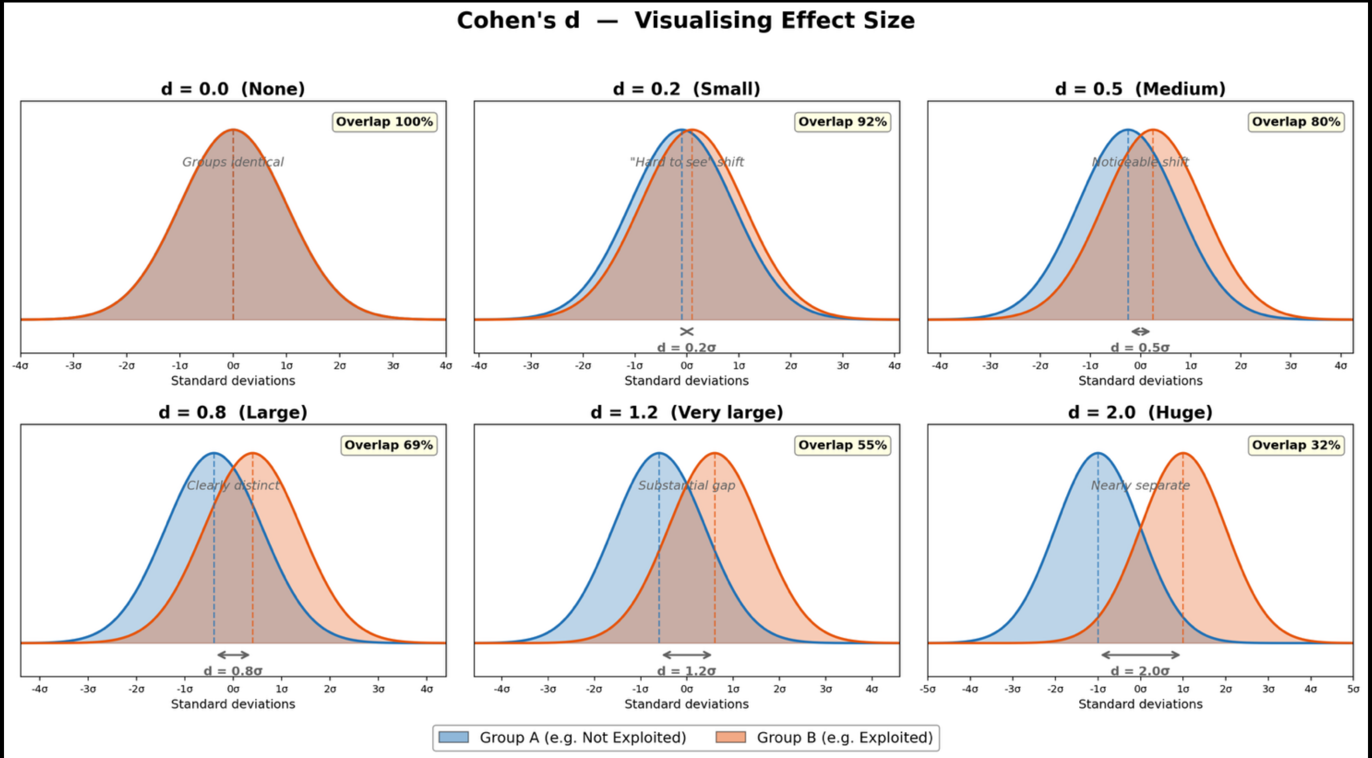
Formula

$$d = \frac{\bar{X}_1 - \bar{X}_2}{SD_{pooled}}$$

Interpretation: How many standard deviations apart are the means?

	d
0.2	Small
0.5	Medium
0.8	Large

Cohen’s d: Visual



Cohen’s d in Python

```
import pingouin as pg  # Install: pip install pingouin

# Compute Cohen's d
d_value = pg.compute_effsize(exploited, not_exploited,
eftype='cohen')
print(f"Cohen's d = {d_value:.2f}")

# Interpretation
if abs(d_value) < 0.2:
    magnitude = "negligible"
elif abs(d_value) < 0.5:
    magnitude = "small"
elif abs(d_value) < 0.8:
    magnitude = "medium"
else:
    magnitude = "large"
print(f"Magnitude: {magnitude}")
```

Vargha-Delaney A (Non-Parametric Effect Size)

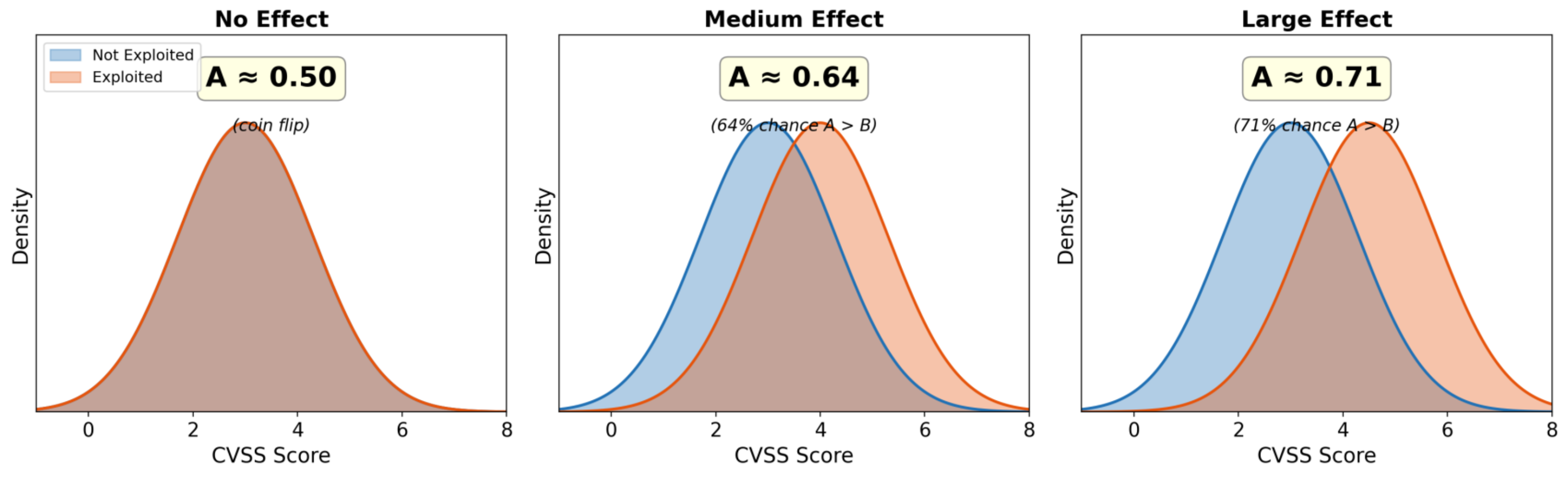
What it measures: The probability that a randomly selected value from group A exceeds a randomly selected value from group B

Interpretation

A value	Meaning
0.50	No difference (coin flip)
0.56	Small effect
0.64	Medium effect
0.71	Large effect
→ 1.0	A always exceeds B

Vargha-Delaney A: Visual

A = 0.50: Complete overlap **A = 0.64:** Moderate separation
A = 0.85: Clear separation



Vargha-Delaney A in Python

```
import pingouin as pg

# Method 1: Get A directly from Mann-Whitney test
mw_result = pg.mwu(exploited, not_exploited,
alternative='two-sided')
print(mw_result)
# Look at the 'CLES' column — this is Vargha-Delaney A

# Method 2: Compute manually (CLES = Common Language
Effect Size)
# A = U / (n1 * n2) where U is Mann-Whitney U statistic
from scipy.stats import mannwhitneyu
u_stat, _ = mannwhitneyu(exploited, not_exploited)
n1, n2 = len(exploited), len(not_exploited)
vd_a = u_stat / (n1 * n2)
print(f"Vargha-Delaney A = {vd_a:.2f}")
```

Running Effect Sizes on Sample Data

Cohen’s d (parametric):

```
Cohen's d = 0.81
Magnitude: large
```

Vargha-Delaney A (non-parametric):

```
Vargha-Delaney A = 0.72 (large effect)
```

Interpretation:

Both effect sizes indicate a **large** effect. Exploited vulnerabilities have substantially higher CVSS scores than non-exploited ones.

⚠️ Pitfall: Conflating Statistical and Practical Significance

Scenario

With $n = 100{,}000$ vulnerabilities:

- Mean CVSS (exploited) = 7.15
- Mean CVSS (non-exploited) = 7.05
- $p < 0.001$, $d = 0.08$

Statistically significant? Yes

Practically significant? Probably not!

Part 5: Bootstrapped Confidence Intervals

Confidence Intervals

Problem: We are computing a value on a sample from a broader population. How close is our estimate to the true value?

Solution: Confidence interval (CI)

The CI is a range; the *confidence level* (e.g., 95%) of it indicates how often the true value falls within the CI over repeated sampling (i.e., in the sampling distribution)

Challenge: How to compute CI?

The Bootstrap Idea

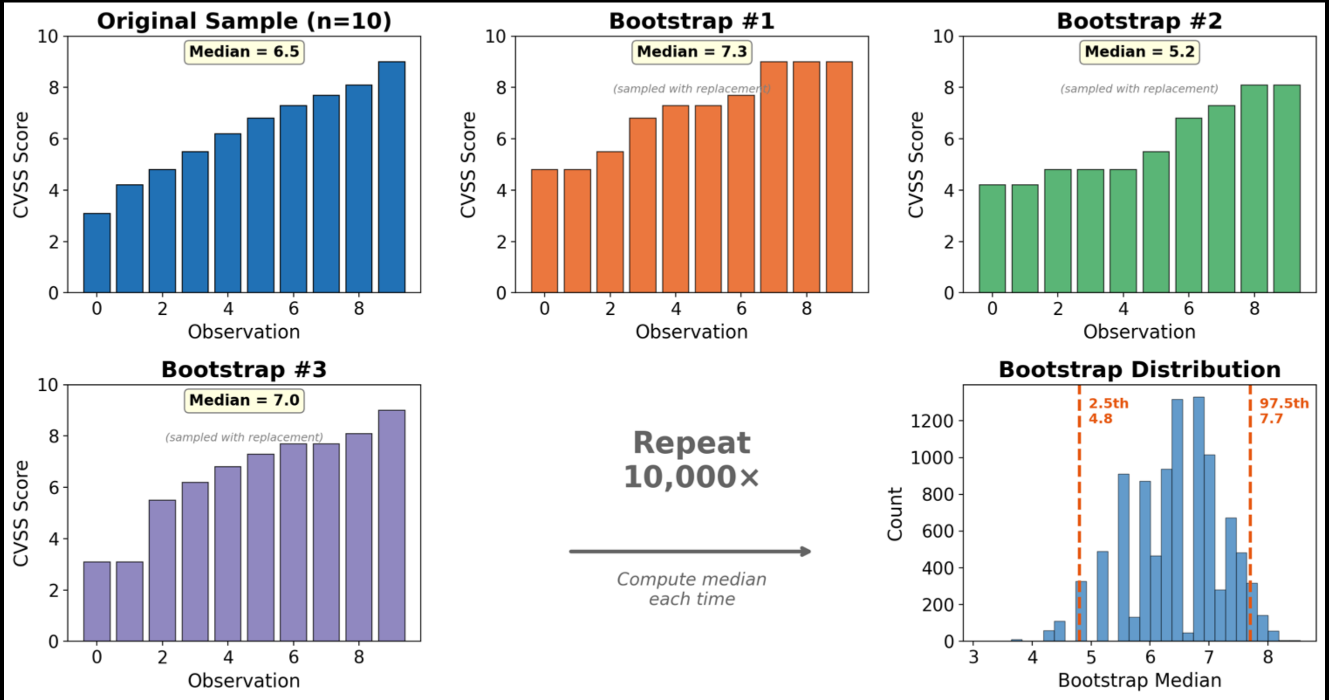
Problem: We want a confidence interval for a statistic (e.g., median difference), but we don't know its sampling distribution

Solution: Simulate the sampling distribution by resampling our data

Bootstrap Procedure

1. Draw a sample of size n *with replacement* from your data (which itself has n elements)
2. Compute the statistic of interest (e.g., median difference)
3. Repeat 10,000 times
4. Use the 2.5th and 97.5th percentiles as the 95% CI

Bootstrap Procedure



Why Bootstrap?

- **No distributional assumptions** — works for any statistic
- **Works for complex statistics** — medians, ratios, custom quantities
- **Intuitive interpretation** — “we’re 95% confident the true value lies in this range”

Bootstrap in Python

```python
import numpy as np

def bootstrap_median_diff(data, n_boot=10000):
    """Bootstrap 95% CI for median difference."""
    # Separate groups
    exploited = data[data['in_kev'] == True]['cvss_base'].values
    not_exploited = data[data['in_kev'] == False]['cvss_base'].values

    # Store bootstrap statistics
    diffs = []
    for _ in range(n_boot):
        # Resample each group with replacement
        e_sample = np.random.choice(exploited, size=len(exploited), replace=True)
        n_sample = np.random.choice(not_exploited, size=len(not_exploited), replace=True)
        # Compute median difference
        diffs.append(np.median(e_sample) - np.median(n_sample))

    # Return 2.5th and 97.5th percentiles
    return np.percentile(diffs, [2.5, 97.5])

ci = bootstrap_median_diff(data)
print(f"95% CI for median difference: [{ci[0]:.2f}, {ci[1]:.2f}]")
```

Running Bootstrap on Sample Data

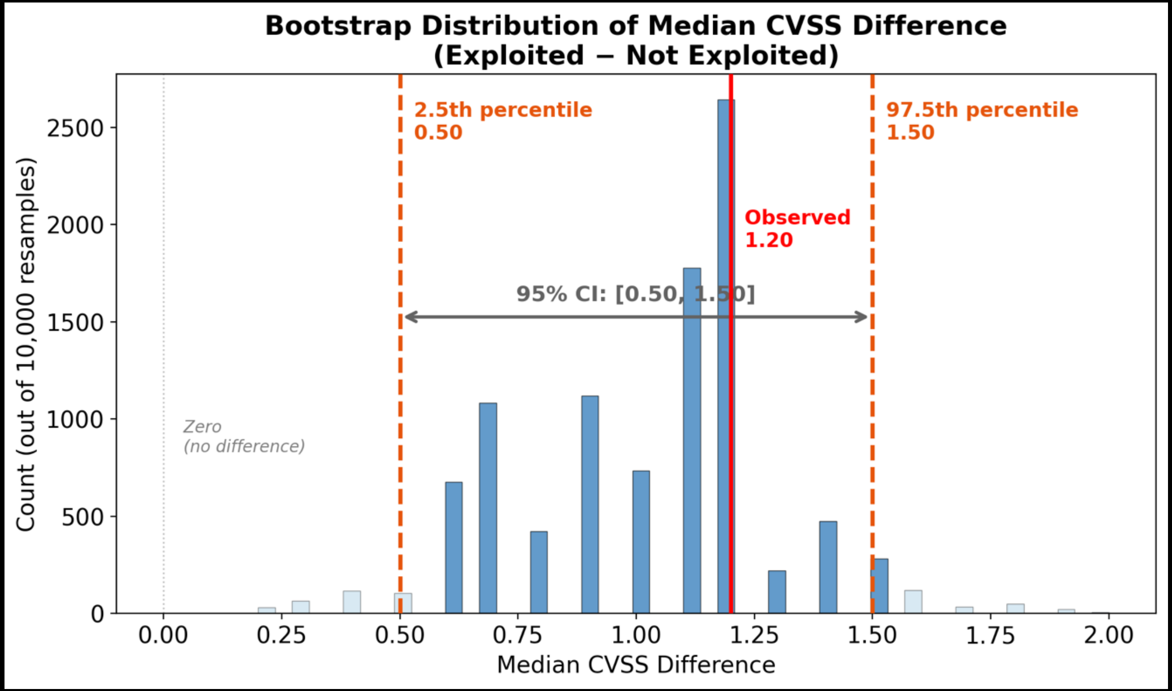
```
Bootstrap 95% CI for median difference: [0.50, 1.50]
```

Interpretation

"The median CVSS of exploited vulnerabilities is 0.90 points higher than non-exploited vulnerabilities, 95% CI [0.50, 1.50]."

The CI doesn't include zero → significant difference in medians.

Bootstrap on our data, visualized



Part 6: Common Pitfalls

Tang et al. Findings

- **97%** of papers had at least one statistical issue
- **23%** had incorrect tests (e.g., non-independence violations)
- **86%** had incomplete statistical significance reporting
- **80%** had incomplete practical significance reporting
- **26%** had misinterpretations

Misuse, Misreporting, Misinterpretation of Statistical Methods in Usable Privacy and Security Papers

Jenny Tang, *Carnegie Mellon University*
Lujo Bauer, *Carnegie Mellon University*
Nicolas Christin, *Carnegie Mellon University*

Abstract

Null hypothesis significance testing (NHST) is commonly used in quantitative usable privacy and security studies. Many papers use results from statistical tests to assert whether effects or differences exist depending on the resulting p-value. We conduct a systematic review of papers published in 10 editions of the Symposium on Usable Privacy and Security over a span of 20 years to evaluate the field's use of NHST. We code statistical tests for potential statistical validity, reporting, or interpretation issues that may undermine assertions made in the 121 papers that use NHST. Most problematically, tests in 23% of papers inadequately account for non-independence between samples, leading to potentially invalid claims. 58% of papers lack information to verify whether an assertion is supported, such as imprecisely specifying the statistical test conducted. Many papers contain more minor statistical issues or report statistics in ways that deviate from best practice. We conclude with recommendations for statistical reporting and statistical thinking in the field.

1 Introduction

Statistical methods are often used in human-computer interaction research to support assertions about the presence (or absence) of an effect of scientific significance (e.g., some magnitude of difference) accompanied by a measure of statistic significance. Indeed, one of the most common refrains in statistical analysis is that a result is significant because the "p-value" is less than a given threshold, e.g., $p < 0.05$. Despite over half a century of criticism, null hypothesis significance testing (NHST, also known as statistical significance testing)—that is, methods using p-values from inferential statistical tests as evidence to reject a null hypothesis—remains the dominant form of statistical analysis and evaluation [17]. However, simply dichotimizing results into "significant" and "non-significant" through their associated p-values without reporting other information is not in itself sufficient to convey the scientific importance of the claims, nor the richness and complexity of data collected from human subjects. This reliance on p-values to support assertions sometimes leads other information vital to understanding statistical and scientific significance to be omitted.

As a result, complete reliance on p-values is increasingly frowned upon, with some journals banning the reporting of p-values altogether [75, 81]. Most other current guidance is less drastic, and recommends using statistical hypothesis testing as a starting point and providing sufficient context (such as effect sizes, confidence intervals, and underlying data) to convey the scientific significance of the claims [2,13,49,59,80,81]. We use this guidance to evaluate whether the scientific assertions made on the basis of NHST in usable privacy and security (UPS) are accompanied by sufficient reporting for readers to validate whether these assertions are supported by the information present in the paper. We focus on UPS as it is still a fairly young area, with evolving standards, features a considerable amount of quantitative research, and errors or misinterpretations can be detrimental to user safety in the digital world and beyond.

Prior work has also examined the transparency, reporting, and validity of statistical methods in HCI and various subfields [16,25,36,51,62,66,77]. However, the evaluations in these works typically focus on evaluating whether p-values are accurately computed or on whether there may be false negatives (such as due to lack of power) or false positives (such as from inaccurately reported p-values).

In this work, we look beyond statistical significance to examine statistical validity (whether the chosen test is suitable for the data or whether it may produce spurious results), reporting transparency and completeness (whether the reported

USENIX Symposium on Usable Privacy and Security (SOUPS) 2025. August 10–12, 2025, Seattle, WA, United States.

The Multiple Comparisons Problem

The problem: At $\alpha = 0.05$, expect 1 false positive per 20 tests *by chance*

Example

Testing whether CVSS differences across 10 CWE categories = 45 pairwise comparisons

Expected false positives by chance: ~2-3

Solutions: Bonferroni Correction

Bonferroni: Divide α by the number of tests

$$\alpha_{adjusted} = \frac{0.05}{k}$$

For 10 tests: $\alpha_{adjusted}$ = 0.005

Pros: Simple, conservative
Cons: Very conservative — increases false negatives

Solutions: Benjamini-Hochberg (FDR)

FDR (False Discovery Rate): Controls the expected *proportion* of false positives among rejected hypotheses

Procedure:

1. Order p-values smallest to largest
2. Compare each p-value to (rank / k) × α
3. Reject all hypotheses up to the largest one that passes

Less conservative than Bonferroni — better for exploratory analysis

Multiple Comparisons in Python

```python
from statsmodels.stats.multitest import multipletests
import numpy as np

# P-values from multiple tests
p_values = np.array([0.001, 0.01, 0.03, 0.04, 0.08, 0.12])

# Bonferroni correction
reject_bonf, p_bonf, _, _ = multipletests(p_values, method='bonferroni')

# Benjamini-Hochberg (FDR) correction
reject_fdr, p_fdr, _, _ = multipletests(p_values, method='fdr_bh')

# Display results
for i, p in enumerate(p_values):
    print(f"p={p:.3f} -> Bonf: {p_bonf[i]:.3f} (sig={reject_bonf[i]}), "
          f"FDR: {p_fdr[i]:.3f} (sig={reject_fdr[i]})")
```

⚠️ Pitfall: Ignoring Non-Independence

The problem: Most tests assume independent observations

Common violations in security research:

- Multiple vulnerabilities from the same vendor
- Multiple CVEs from the same software product
- Vulnerabilities discovered by the same researcher

Ask yourself: “Could any two data points be more similar to each other than to a random pair?”

⚠️ Pitfall: Reporting Only p-values

Insufficient: "There was a significant difference ($p = 0.02$)."

Complete reporting includes:

1. The exact test name
2. Test statistic and degrees of freedom
3. Exact p-value (or $p < 0.001$)
4. Effect size
5. Descriptive statistics for each group

Complete Reporting Example

Bad:

“There was a significant difference ($p = 0.02$).”

Good:

“Exploited vulnerabilities had significantly higher CVSS scores (Mdn = 6.30, IQR = 1.85) than non-exploited vulnerabilities (Mdn = 5.30, IQR = 2.40), Mann-Whitney $U = 56{,}789$, $p < 0.001$, Vargha-Delaney $A = 0.74$ (large effect).”

Lecture 1 Checklist

Statistical Validity:

- ☐ Is my test appropriate for my data type?
- ☐ Have I accounted for non-independence?
- ☐ Am I using paired tests for paired data?

Multiple Comparisons:

- ☐ Have I corrected for multiple comparisons?

Reporting:

- ☐ Test name, statistic, df, p-value?
- ☐ Effect size?
- ☐ Descriptive statistics with variability?

Lecture 1 Summary

Concept	Key Takeaway	Project Use
p-values	P(data | H_0), not P(H_0 is true)	
Chi-square	Association between categorical variables	Severity × CWE
t-test	Parametric comparison of means	When data is normal
Mann-Whitney U	Non-parametric group comparison	Exploited vs. not
Cohen's d	Parametric effect size	With t-test
Vargha-Delaney A	Non-parametric effect size	With Mann-Whitney
Bootstrapping	CIs without assumptions	Median differences
Multiple comparisons	Correct when running many tests	Post-hoc tests

Recommended Readings

Primary Textbook (Franke):

- Section 16.2 — p-values
- Section 16.6.1 — Chi-square
- Section 12.1 — Linear regression
- Section 15.2 — Logistic regression

Secondary Textbook (Seltman):

- Chapter 6.2 — Hypothesis testing
- Chapter 9 — Linear regression
- Chapter 16.2-16.3 — Chi-square and logistic

Misuse, Misreporting, Misinterpretation of Statistical Methods in Usable Privacy and Security Papers

Jenny Tang
Carnegie Mellon University

Lujo Bauer
Carnegie Mellon University

Nicolas Christin
Carnegie Mellon University

Abstract

Null hypothesis significance testing (NHST) is commonly used in quantitative usable privacy and security studies. Many papers use results from statistical tests to assert whether effects or differences exist depending on the resulting *p*-value. We conduct a systematic review of papers published in 10 editions of the Symposium on Usable Privacy and Security over a span of 20 years to evaluate the field's use of NHST. We code statistical tests for potential statistical validity, reporting, or interpretation issues that may undermine assertions made in the 121 papers that use NHST. Most problematically, tests in 23% of papers inadequately account for non-independence between samples, leading to potentially invalid claims. 58% of papers lack information to verify whether an assertion is supported, such as imprecisely specifying the statistical test conducted. Many papers contain more minor statistical issues or report statistics in ways that deviate from best practice. We conclude with recommendations for statistical reporting and statistical thinking in the field.

1 Introduction

Statistical methods are often used in human-computer interaction research to support assertions about the presence (or absence) of an effect of scientific significance (e.g., some magnitude of difference) accompanied by a measure of statistic significance. Indeed, one of the most common refrains in statistical analysis is that a result is significant because the "*p*-value" is less than a given threshold, e.g., $p < 0.05$. Despite over half a century of criticism, null hypothesis significance testing (NHST, also known as statistical significance testing)—that is, methods using *p*-values from inferential statistical tests as evidence to reject a null hypothesis—remains the dominant form of statistical analysis and evaluation [17]. However, simply dichotimizing results into "significant" and "non-significant" through their associated *p*-values without reporting other information is not in itself sufficient to convey the scientific importance of the claims, nor the richness and complexity of data collected from human subjects. This reliance on *p*-values to support assertions sometimes leads other information vital to understanding statistical and scientific significance to be omitted.

As a result, complete reliance on *p*-values is increasingly frowned upon, with some journals banning the reporting of *p*-values altogether [75, 81]. Most other current guidance is less drastic, and recommends using statistical hypothesis testing as a starting point and providing sufficient context (such as effect sizes, confidence intervals, and underlying data) to convey the scientific significance of the claims [2,13,49,59,80,81]. We use this guidance to evaluate whether the scientific assertions made on the basis of NHST in usable privacy and security (UPS) are accompanied by sufficient reporting for readers to validate whether these assertions are supported by the information present in the paper. We focus on UPS as it is still a fairly young area, with evolving standards, features a considerable amount of quantitative research, and errors or misinterpretations can be detrimental to user safety in the digital world and beyond.

Prior work has also examined the transparency, reporting, and validity of statistical methods in HCI and various subfields [16,25,36,51,62,66,77]. However, the evaluations in these works typically focus on evaluating whether *p*-values are accurately computed or on whether there may be false negatives (such as due to lack of power) or false positives (such as from inaccurately reported *p*-values).

In this work, we look beyond statistical significance to examine statistical validity (whether the chosen test is suitable for the data or whether it may produce spurious results), reporting transparency and completeness (whether the reported

USENIX Symposium on Usable Privacy and Security (SOUPS) 2025.
August 10–12, 2025, Seattle, WA, United States.