UNIVERSITY OF PADOVA
DEPARTMENT OF INFORMATION ENGINEERING

Biomedical Wearable Technologies
for Healthcare and Wellbeing

Quantitative Usability Evaluation

A.Y. 2023-2024
Giacomo Cappon

DEPARTMENT OF
INFORMATION
ENGINEERING
UNIVERSITY OF PADOVA

Usability

- **Usability** has an international standard definition in **ISO 9241-11**: the extent to which a product can be used by specified users to achieve specified goals with **effectiveness**, **efficiency**, and **satisfaction** in a **specified context of use**.

- Three separate components:
  - **Effectiveness**: whether people can actually complete their tasks and achieve their goals
  - **Efficiency**: the extent to which they expend resource in achieving their goals
  - **Satisfaction**: the level of comfort they experience in achieving those goals

- Thus, a system that lets people complete their tasks, but at the expense of considerable expenditure of time and effort and which was felt to be very unsatisfactory by all concerned, could not really be said to be usable.

- Seamlessly, a system which people enjoyed using but which didn’t allow them to complete any tasks and on which they spent a lot of unproductive time is not very usable.

Usability

- **Problem:** ISO 9241-11 makes the point that there are **no specific guidelines** on how to measure effectiveness, efficiency, and satisfaction since they depend on:
  - **Background/Experience** of the user
  - **Task** that the user must perform
  - **Environment** in which the task is performed

- There are generally two types of usability tests:
  - **Formative tests**
  - **Summative tests**

Formative tests

- **Formative** tests: they are the bulk of usability testing, and they have the aim of finding and fixing usability problems

- Data from formative tests take the form of problem descriptions and design recommendations

- Quantification can be made in terms of frequency and severity of an encountered problem, time to complete a task, completion rate of a task.

Summative tests

- **Summative tests** have the aim of describing the usability of an application using metrics. They are of two types: benchmark and comparative

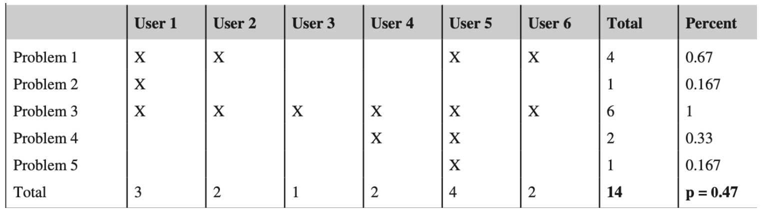
- **Benchmark Usability Tests**: aim to describe how usable an application is relative to a set of benchmark goals. They provide:
  - Input on what to fix in an interface
  - Baseline for the comparison of post-design changes

- **Comparative Usability Tests**: aim to compare usability of two or more applications or different versions of the same application
  - They allow to identify the best tool for the job from the point-of-view of usability

Metrics

- **Completion rates** are the most fundamental of usability metrics.
  - Typically, they are binary (i.e., 0 or 1)
  - You report rates as a percentage of user that complete a specific task.

- **UI problems**: if a user found a problem while attempting a task
  - Typically organized into lists with names, descriptions, and severity rating.
  - You report the occurrence rate of a problem in a UI problem matrix:



	User 1	**User 2**	**User 3**	**User 4**	**User 5**	**User 6**	**Total**	**Percent**
Problem 1	X	X			X	X	4	0.67
Problem 2	X						1	0.167
Problem 3	X	X	X	X	X	X	6	1
Problem 4				X	X		2	0.33
Problem 5					X		1	0.167
Total	3	2	1	2	4	2	**14**	**p = 0.47**

  - Using the matrix, and assigning an impact level (from 1 to 10) to each problem, you can create priorities for each problem on a scale from 1 to 100) with the formula

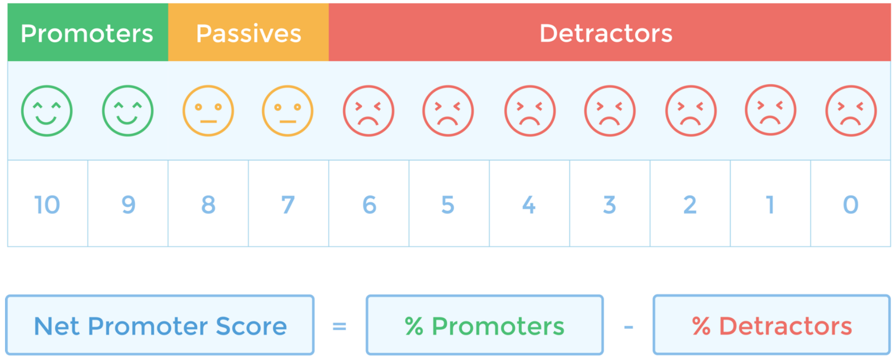
$$priority = (occurence\% * impact)/10$$

  - E.g., a problem with occurrence 80% and impact 3 has priority 24

Metrics

- **Net promoter score (NPS)**: is a score based on a single question on loyalty: “How likely it is that you will recommend this product to a friend or colleague on a scale from 0 to 10?”

- You are a:
  - **Promoter** if your response is 9 or 10
  - **Passive** if your response is 7 or 8
  - **Detractor** if your response is < 7



Promoters		Passives		Detractors						
10	9	8	7	6	5	4	3	2	1	0

Net Promoter Score = % Promoters - % Detractors

- NPS is computed subtracting the percentage of promoters with the percentage of detractors

- The final score goes from -100% to 100%

Metrics

- **Comments and open-ended data**: can be very heterogeneous and might include:
  - Reasons why users are promoters or detractors for an app
  - User insights from field studies
  - App complaints to calls to user service
  - Why a task was difficult

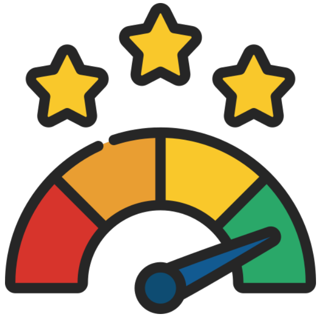
- **Requirement list**: features that a user identify as necessary in your app but it is still not implemented.

Satisfaction ratings

- **Satisfaction ratings**: obtained via standardized usability questionnaries.

- **Standardized usability questionnaires (SUQ)** consists of a collection of question items each associated to an underneath score. At the end, all questions are collected in a single value (i.e., satisfaction rating) that provide a standardized measure.

- The advantages of SUQ include:
  - **Objectivity**: independent verification of the measurement
  - **Replicability**: it is easy to replicate the studies of others
  - **Quantification**: results can be quantified via numbers and statistics
  - **Economy**: very cheap to reuse
  - **Communication**: results are very easy to communicate
  - **Generalization**: it is fundamental that the results are generalizable



Assessing the quality of a SUQ

- However, we cannot just put together a bunch of questions to create a good SUQ. Indeed, a good SUQ must be **reliable**, **valid**, **sensitive**.

- **Reliability**: consistency of measurement
  - Can be measured via the Cronbach's alpha. Can range between 0 (no reliability) and 1 (perfect reliability). Measures that can affect a person's future (e.g., college entrance test) should have a minimum alpha of 0.9. For other research or evaluations > 0.7 is acceptable.

- **Validity**: am I measuring what I am trying to measure?
  - Content validity
  - Criterion-related validity
  - Construct validity

- **Sensitivity**: it must be sensitive to experimental manipulations
  - For example, responses from participants who experience difficulties working with Product A but find Product B easy to use should reflect a statistically significant difference in the overall SEQ outcome
  - There is no direct measurement of sensitivity.

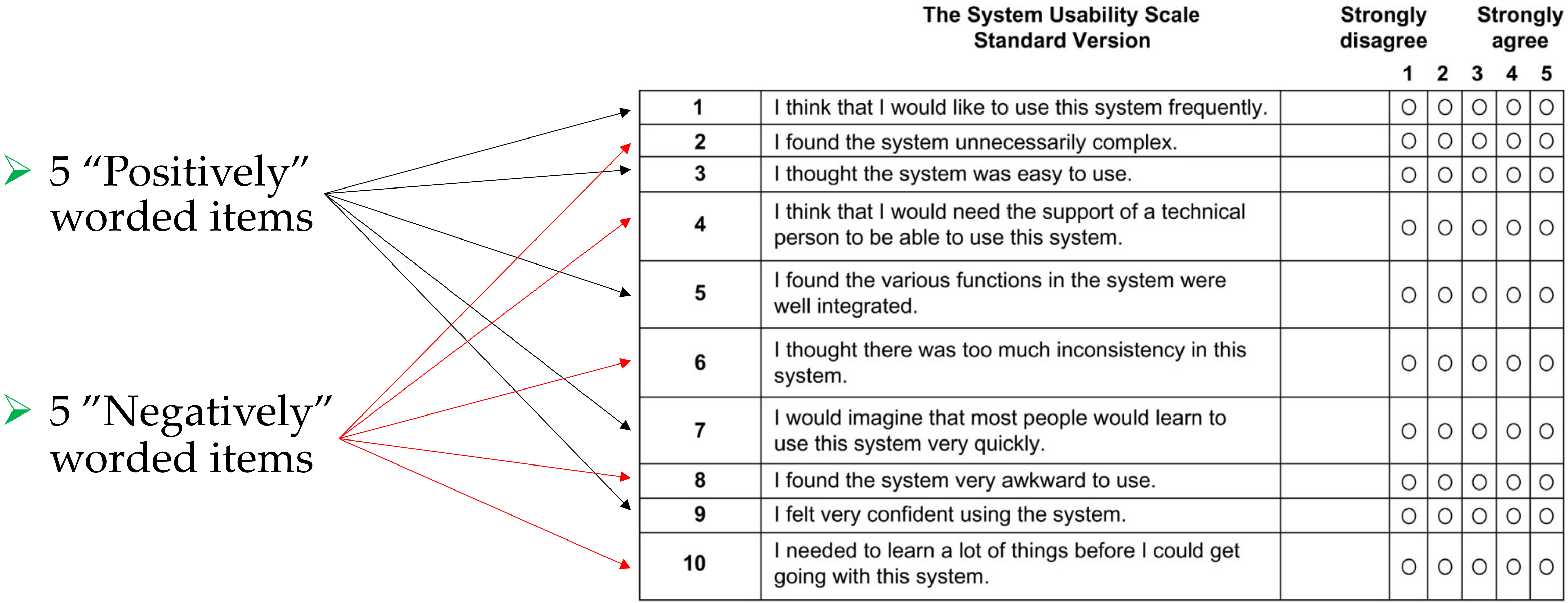
Available SUQ

- SUQ can be administered at the end of each task, i.e., **post-task**:
  - After-Scenario Questionnaire (ASQ) - Lewis, 1991
  - Expectation Ratings (ER) – Albert and Dixon, 2003
  - Usability Magnitude Estimation (UME) – McGee, 2004
  - Single Ease Question (SEQ) – Sauro, 2010
  - Subjective Mental Effort Question (SMEQ) – Sauro and Dumas, 2009

- …or at the end of the entire study, i.e., **post-study**:
  - Questionnaire for User Interaction Satisfaction (QUIS) – Chin et al., 1988
  - Software Usability Measurement Inventory (SUMI) – Kirakowski and Corbett, 1993
  - Post-Study System Usability Questionnaire (PSSUQ) – Lewis, 1995
  - **System Usability Scale (SUS) – Brooke, 1996**

- **General recommendation:**
  - Post-test: SEQ or SMEQ
  - Post-study: SUS

SUS



- 5 “Positively” worded items
- 5 ”Negatively” worded items

	The System Usability Scale Standard Version		Strongly disagree				Strongly agree
			1	2	3	4	5
1	I think that I would like to use this system frequently.		○	○	○	○	○
2	I found the system unnecessarily complex.		○	○	○	○	○
3	I thought the system was easy to use.		○	○	○	○	○
4	I think that I would need the support of a technical person to be able to use this system.		○	○	○	○	○
5	I found the various functions in the system were well integrated.		○	○	○	○	○
6	I thought there was too much inconsistency in this system.		○	○	○	○	○
7	I would imagine that most people would learn to use this system very quickly.		○	○	○	○	○
8	I found the system very awkward to use.		○	○	○	○	○
9	I felt very confident using the system.		○	○	○	○	○
10	I needed to learn a lot of things before I could get going with this system.		○	○	○	○	○

Evaluation of SUS

- The ten SUS items were selected from a pool of 50 potential items, based on the responses of 20 people who used the full set of items to rate two software systems, one of which was relatively easy to use, and the other relatively difficult.
- The items selected for the SUS were those that provided the strongest discrimination between the systems.
- In the original paper by Brooke (1996), he did not report any measures of reliability or validity, referring to the SUS as a quick and dirty usability scale.
- An early assessment of the SUS indicated a **reliability** of 0.85. More recent estimates using larger samples have consistently found its reliability to be at or just over **0.90**.

How SUS score works

- Each item's score contribution ranges from 0 to 4.
- For items 1, 3, 5, 7, and 9 (the positively worded items) the score contribution is the scale position minus 1.
- For items 2, 4, 6, 8, and 10 (the negatively worded items), the contribution is 5 minus the scale position.
- You then multiply the sum of the scores by 2.5 to obtain the overall value of SUS.

Psychometric evaluation of SUS

- According to an investigation of the psychometric properties of the SUS (from 2324 SUS questionnaires), **two factors** can be identified in the SUS questionnaire:
  - Items 1, 2, 3, 5, 6, 7, 8, and 9 quantify the "**Usable**" factor
  - Items 4 and 10 quantify the "**Learnable**" factor

- To make the Usable and Learnable scores comparable with the Overall SUS score so they also range from 0 to 100, just multiply their summed score contributions by 3.125 for Usable and 12.5 for Learnable.

- The two subscale reliabilities were 0.91 for Usable and 0.70 for Learnable.

Where did the 2.5, 3.125 and 12.5 multipliers come from?

- The standard SUS raw score contributions can range from 0 to 40 (ten items with five scale steps ranging from 0 to 4). To get the multiplier needed to increase the apparent range of the summed scale to 100, divide 100 by the maximum sum of 40, which equals 2.5:

  100/40 = 2.5

- Because the Usable subscale has eight items, its range for summed score contributions is 0-32, so its multiplier is:

  100/32 = 3.125

- Following the same process, the multiplier for the Learnable subscale is:

  100/8 = 12.5

Example – Overall SUS

- For items 1, 3, 5, 7, and 9 (the positively worded items) the score contribution is the scale position minus 1:

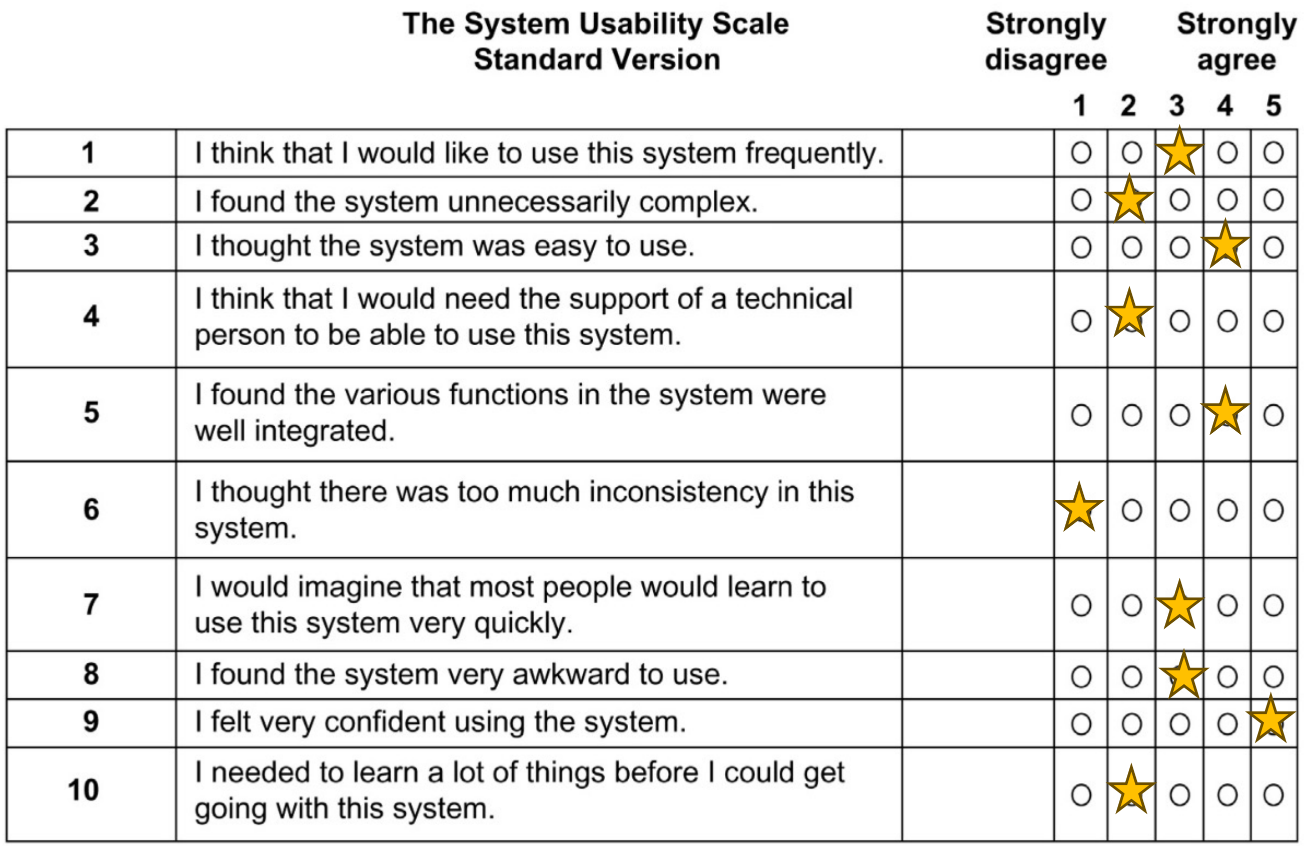
(3-1) + (4-1) + (4-1) + (3-1) + (5-1) = 16

- For items 2, 4, 6, 8, and 10 (the negatively worded items), the contribution is 5 minus the scale position:

(5-2) + (5-2) + (5-1) + (5-3) + (5-2) = 15

- You then multiply the sum of the scores by 2.5 to obtain the overall value of SUS:

(16 + 15) x 2.5 = **77.5**



The System Usability Scale Standard Version

Strongly disagree — Strongly agree

		1	2	3	4	5
1	I think that I would like to use this system frequently.	○	○	★	○	○
2	I found the system unnecessarily complex.	○	★	○	○	○
3	I thought the system was easy to use.	○	○	○	★	○
4	I think that I would need the support of a technical person to be able to use this system.	○	★	○	○	○
5	I found the various functions in the system were well integrated.	○	○	○	★	○
6	I thought there was too much inconsistency in this system.	★	○	○	○	○
7	I would imagine that most people would learn to use this system very quickly.	○	○	★	○	○
8	I found the system very awkward to use.	○	○	★	○	○
9	I felt very confident using the system.	○	○	○	○	★
10	I needed to learn a lot of things before I could get going with this system.	○	★	○	○	○

Example – Usable factor

- We just need to account for items 1, 2, 3, 5, 6, 7, 8, and 9

- For items 1, 3, 5, 7, and 9 the score contribution is the scale position minus 1:

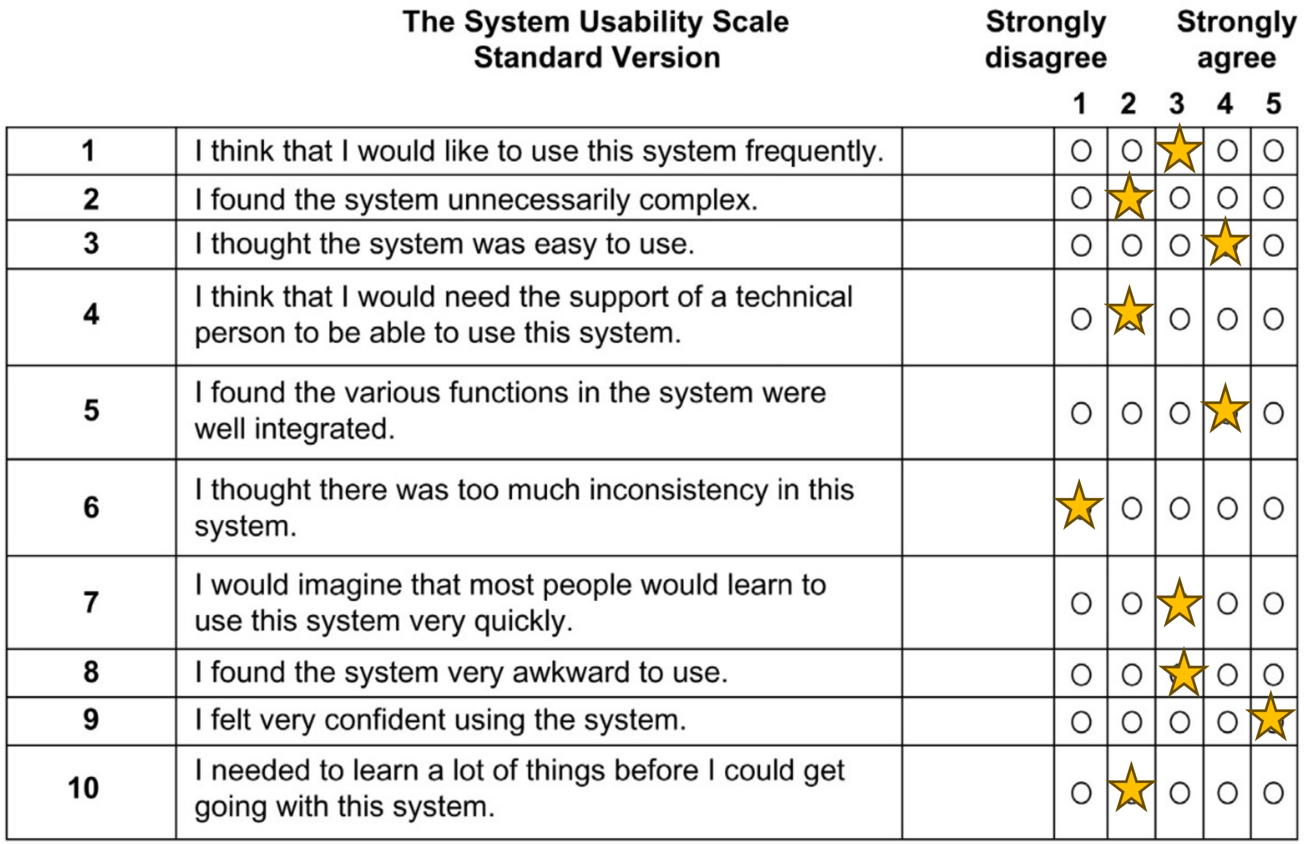
(3-1) + (4-1) + (4-1) + (3-1) + (5-1) = 16

- For items 2, 6, and 8 the contribution is 5 minus the scale position:

(5-2) + (5-1) + (5-3) = 9

- You then multiply the sum of the scores by 3.125 to obtain the Usable SUS score:

(16 + 9) x 2.5 = **78.125**



	The System Usability Scale Standard Version	**Strongly disagree**				**Strongly agree**
		1	**2**	**3**	**4**	**5**
1	I think that I would like to use this system frequently.	○	○	★	○	○
2	I found the system unnecessarily complex.	○	★	○	○	○
3	I thought the system was easy to use.	○	○	○	★	○
4	I think that I would need the support of a technical person to be able to use this system.	○	★	○	○	○
5	I found the various functions in the system were well integrated.	○	○	○	★	○
6	I thought there was too much inconsistency in this system.	★	○	○	○	○
7	I would imagine that most people would learn to use this system very quickly.	○	○	★	○	○
8	I found the system very awkward to use.	○	○	★	○	○
9	I felt very confident using the system.	○	○	○	○	★
10	I needed to learn a lot of things before I could get going with this system.	○	★	○	○	○

Example – Learnable factor

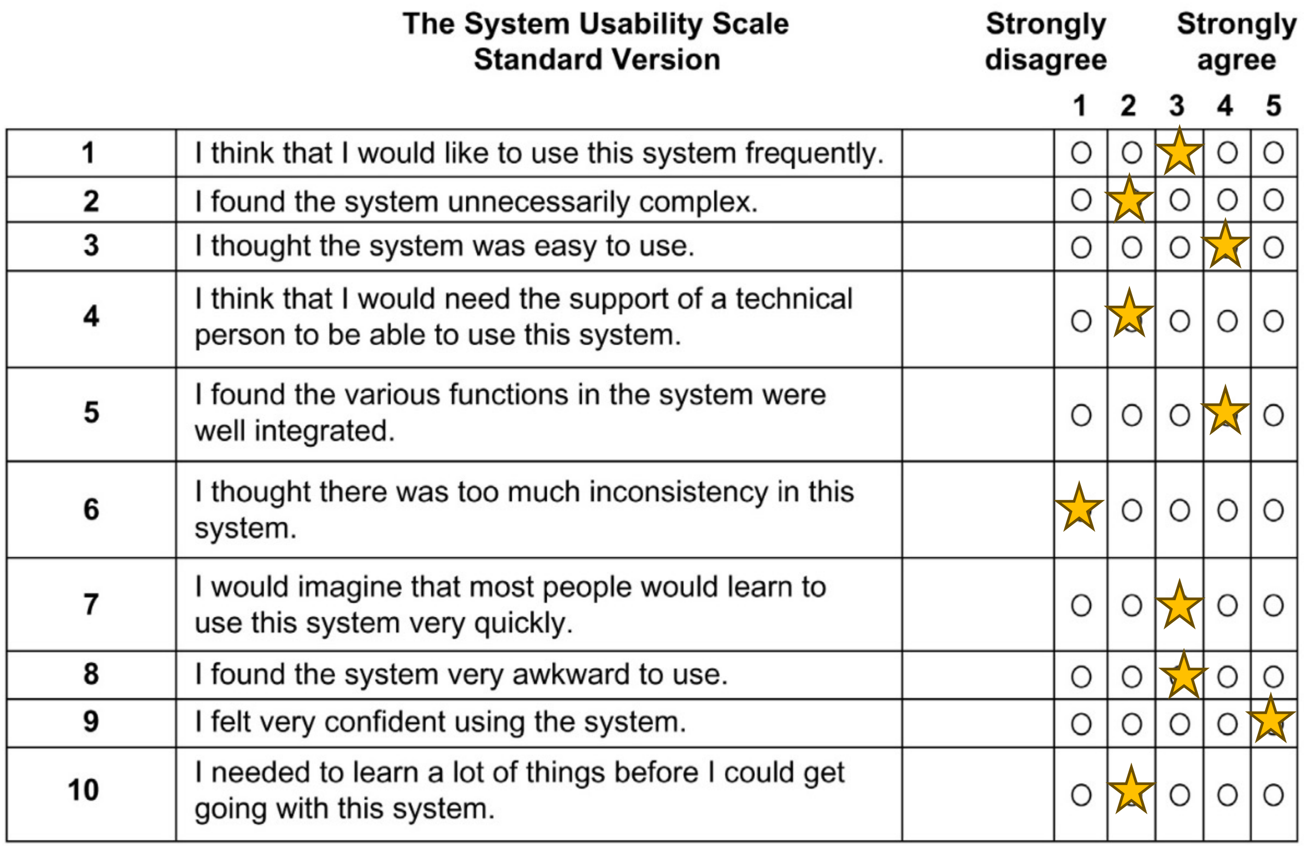
- We just need to account for items 4 and 10

- For items 4, and 10 the contribution is 5 minus the scale position:

(5-2) + (5-2) = 6

- You then multiply the sum of the scores by 12.5 to obtain the Leanable SUS score:

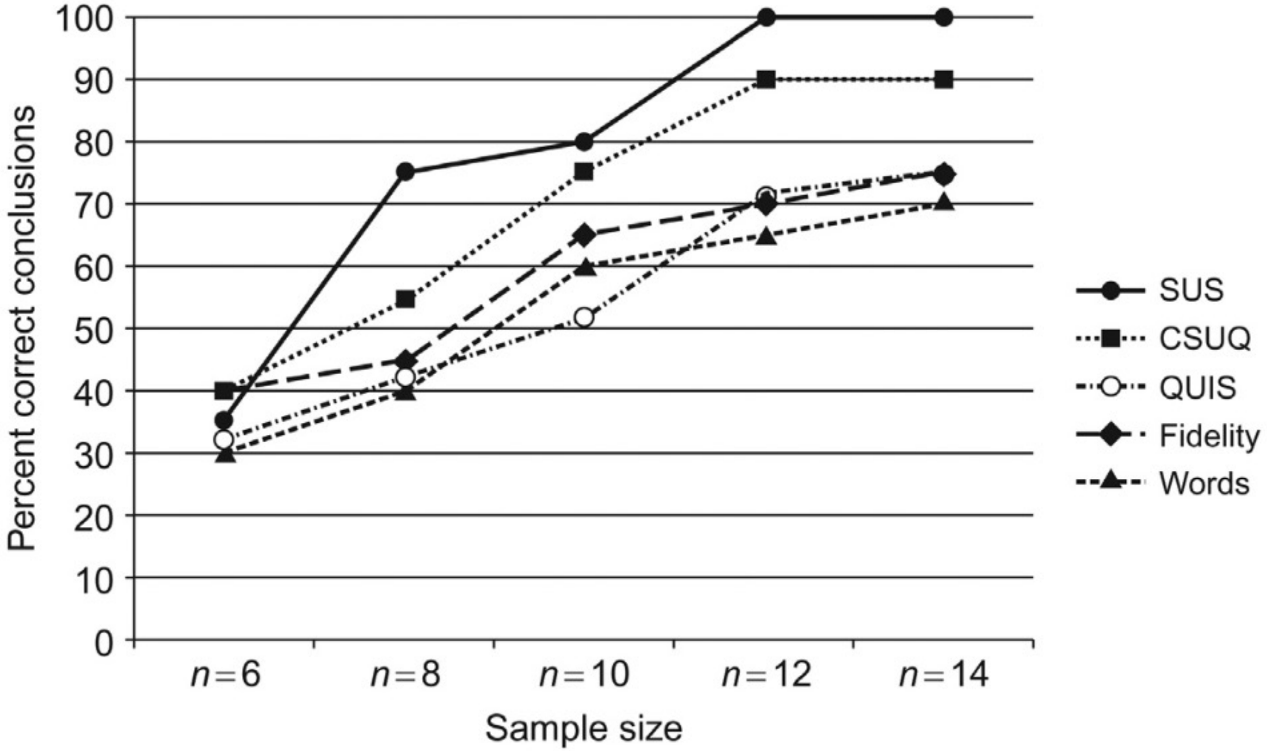
6 x 12.5 = **75**



	The System Usability Scale Standard Version		**Strongly disagree**				**Strongly agree**
			1	**2**	**3**	**4**	**5**
1	I think that I would like to use this system frequently.		○	○	★	○	○
2	I found the system unnecessarily complex.		○	★	○	○	○
3	I thought the system was easy to use.		○	○	○	★	○
4	I think that I would need the support of a technical person to be able to use this system.		○	★	○	○	○
5	I found the various functions in the system were well integrated.		○	○	○	★	○
6	I thought there was too much inconsistency in this system.		★	○	○	○	○
7	I would imagine that most people would learn to use this system very quickly.		○	○	★	○	○
8	I found the system very awkward to use.		○	○	★	○	○
9	I felt very confident using the system.		○	○	○	○	★
10	I needed to learn a lot of things before I could get going with this system.		○	★	○	○	○

How many people do I need?

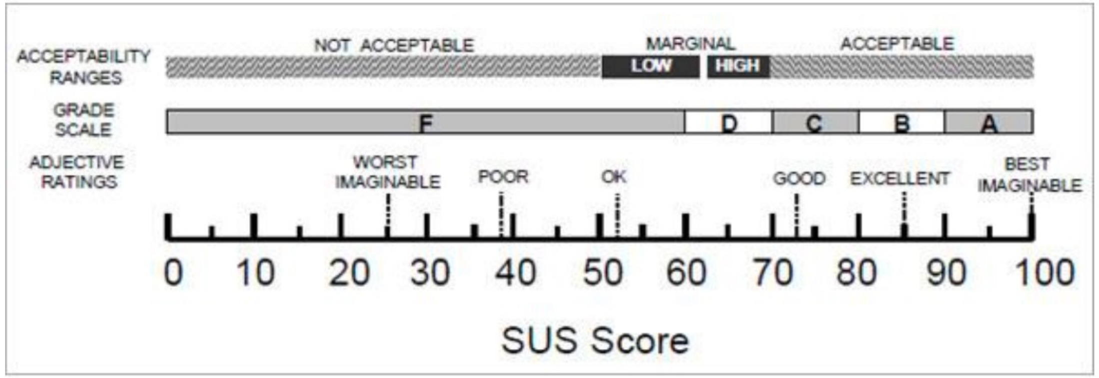
- To run a usability test using SUS you do not always need a lot of people
- According to a study of Tullis and Stetson, 2004 good results can be obtained with 8-12 people.
- **Remember**: your test user pool must be **representative** of the final user population.
- (Extreme) Example: If your app is intended for pregnant women, your pool cannot be made of men

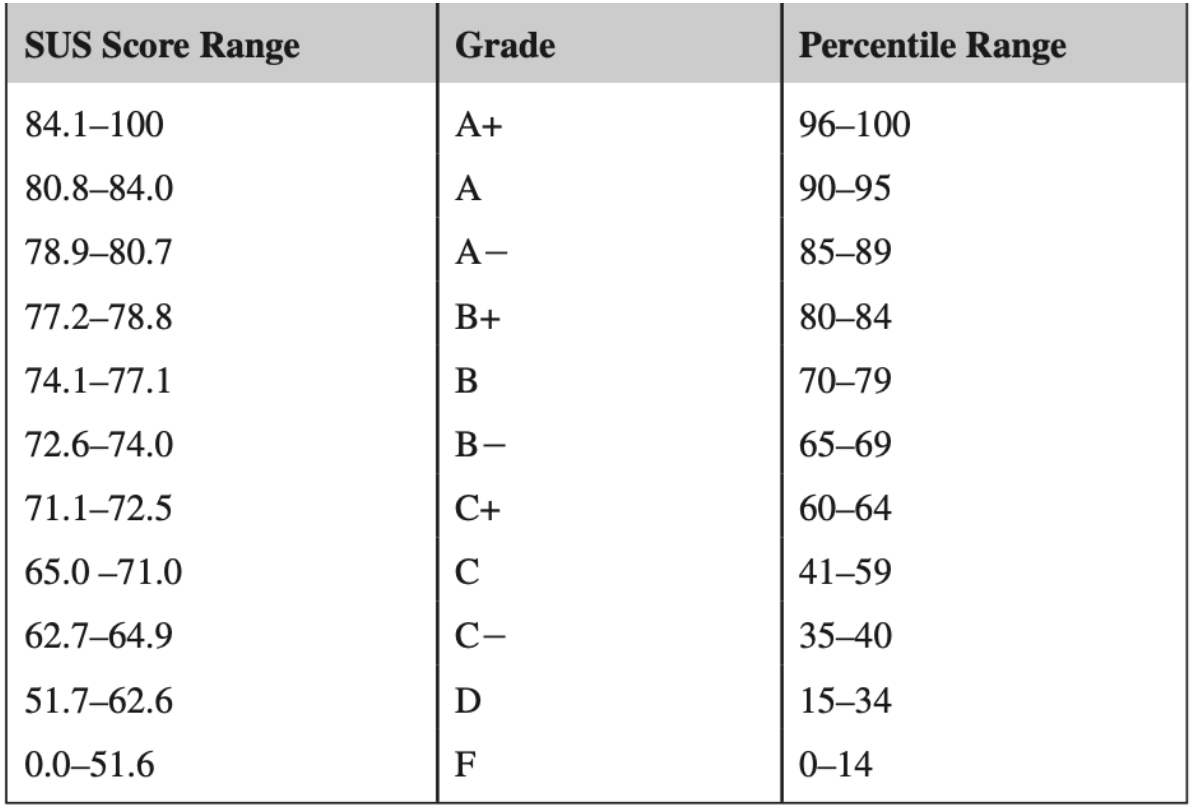


What is a good SUS?

- To be “good” SUS must be > 73

- Note: It has been showed that a SUS score of 82 (±5), also tend to be "Promoters.” according to NPS

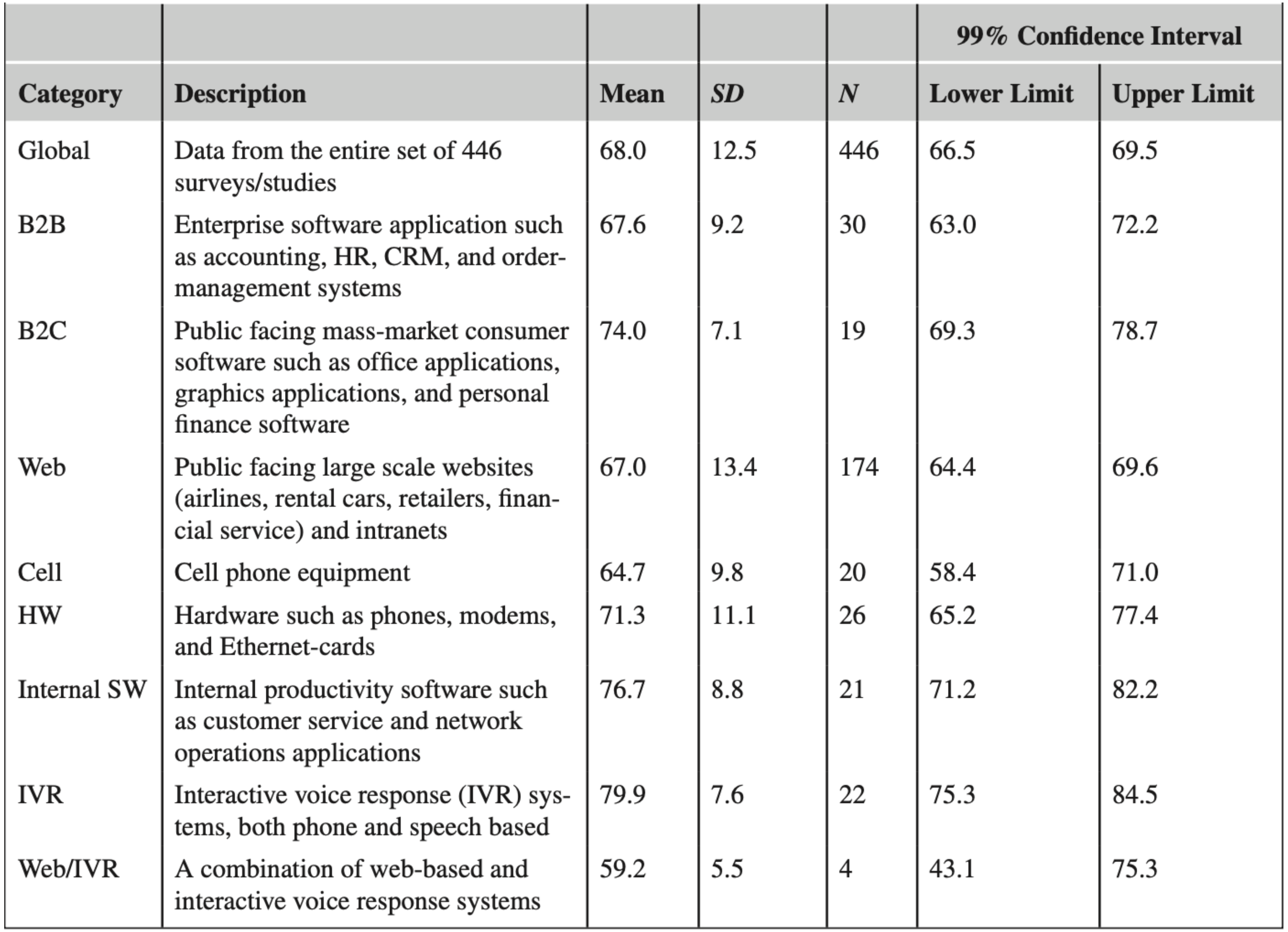




SUS Score Range	Grade	Percentile Range
84.1–100	A+	96–100
80.8–84.0	A	90–95
78.9–80.7	A−	85–89
77.2–78.8	B+	80–84
74.1–77.1	B	70–79
72.6–74.0	B−	65–69
71.1–72.5	C+	60–64
65.0 –71.0	C	41–59
62.7–64.9	C−	35–40
51.7–62.6	D	15–34
0.0–51.6	F	0–14

BONUS

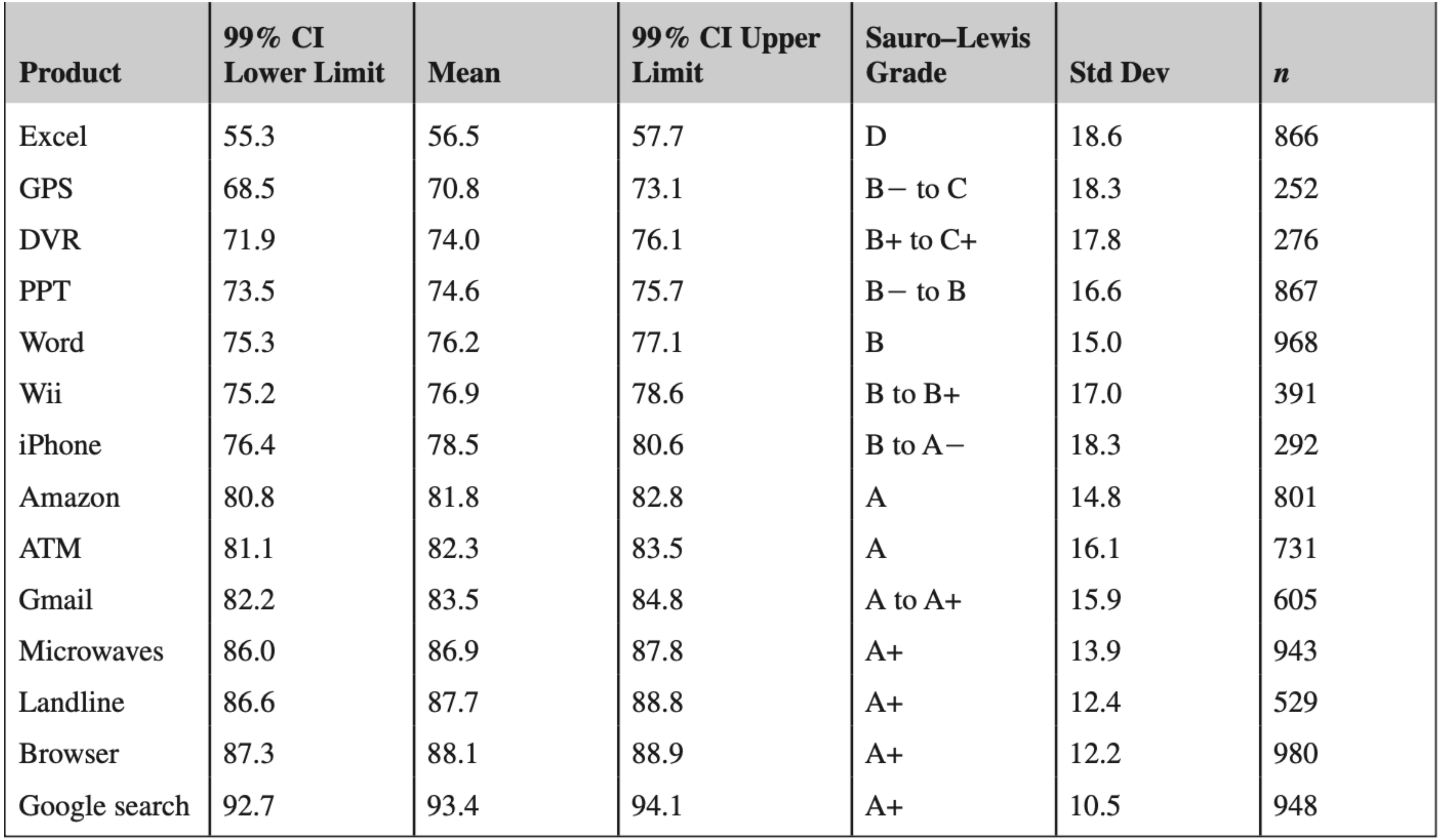
What is a good (interface specific) SUS?



					99% Confidence Interval	
Category	**Description**	**Mean**	***SD***	***N***	**Lower Limit**	**Upper Limit**
Global	Data from the entire set of 446 surveys/studies	68.0	12.5	446	66.5	69.5
B2B	Enterprise software application such as accounting, HR, CRM, and order-management systems	67.6	9.2	30	63.0	72.2
B2C	Public facing mass-market consumer software such as office applications, graphics applications, and personal finance software	74.0	7.1	19	69.3	78.7
Web	Public facing large scale websites (airlines, rental cars, retailers, financial service) and intranets	67.0	13.4	174	64.4	69.6
Cell	Cell phone equipment	64.7	9.8	20	58.4	71.0
HW	Hardware such as phones, modems, and Ethernet-cards	71.3	11.1	26	65.2	77.4
Internal SW	Internal productivity software such as customer service and network operations applications	76.7	8.8	21	71.2	82.2
IVR	Interactive voice response (IVR) systems, both phone and speech based	79.9	7.6	22	75.3	84.5
Web/IVR	A combination of web-based and interactive voice response systems	59.2	5.5	4	43.1	75.3

BONUS

How everyday products are rated using SUS?



Product	99% CI Lower Limit	Mean	99% CI Upper Limit	Sauro–Lewis Grade	Std Dev	*n*
Excel	55.3	56.5	57.7	D	18.6	866
GPS	68.5	70.8	73.1	B− to C	18.3	252
DVR	71.9	74.0	76.1	B+ to C+	17.8	276
PPT	73.5	74.6	75.7	B− to B	16.6	867
Word	75.3	76.2	77.1	B	15.0	968
Wii	75.2	76.9	78.6	B to B+	17.0	391
iPhone	76.4	78.5	80.6	B to A−	18.3	292
Amazon	80.8	81.8	82.8	A	14.8	801
ATM	81.1	82.3	83.5	A	16.1	731
Gmail	82.2	83.5	84.8	A to A+	15.9	605
Microwaves	86.0	86.9	87.8	A+	13.9	943
Landline	86.6	87.7	88.8	A+	12.4	529
Browser	87.3	88.1	88.9	A+	12.2	980
Google search	92.7	93.4	94.1	A+	10.5	948

Final notes on SUS

- Extensive analyses of SUS show that:
  - SUS is **reliable**. Users respond consistently to the scale items, and SUS has been shown to detect differences at smaller sample sizes than other questionnaires.
  - SUS is **valid**. That is, it measures what it purports to measure.
  - SUS is **not diagnostic**. That is, it does not tell you what makes a system usable or not.
  - SUS **scores are not percentages**, despite returning a value between 0 and 100. To understand how your product compares to others, you need to look at its percentile ranking.
  - SUS measures both **learnability** and **usability**.
  - SUS scores have a **modest correlation with task performance**, but it is not surprising that people's subjective assessments may not be consistent with whether or not they were successful using a system.

How precise is our estimate?

- In usability testing, like most applied research settings, we almost never have access to the entire user population: we have to rely on taking samples to estimate the unknown population values.

- The sample means and sample proportions (called statistics) are estimates of the values we really want.

- We need a way to know how good (precise) our estimates are.

- To do so, we construct a range of values that we think will have a specified chance of containing the unknown population parameter. These ranges are called **confidence intervals (CI)**.

Confidence interval (CI)

- CI quantifies twice the margin of error.
  - If you hear that 57% of likely voters approve a legislation with a margin of error ±3% then the CI is 6% wide, falling between 54% and 60% (57 – 3% and 57 + 3%).

- CI provides both a measure of **location** and **precision**.
  - That is, we can see that the average approval rating is around 57%. We can also see that this estimate is reasonably precise. If we want to know whether the majority of voters approve the legislation we can see that it is very unlikely that fewer than half the voters approve.

- CI is made of 3 components:
  - Confidence level
  - Variability
  - Sample size

Components of a CI

- **Confidence level**: the ”coverage” of a confidence interval
  - A confidence level of 95% (the typical value) means that if you were to sample from the same population 100 times, you’d expect the interval to contain the actual mean or proportion 95 times.

- **Variability**: If there is more variation in a population, each sample taken will fluctuate more and therefore create a wider CI.
  - The variability of the population is estimated using the **standard deviation** from the sample.

- **Sample size**: how many samples do you have
  - Without lowering the confidence level, the sample size is the only thing a researcher can control in affecting the width of a CI.
  - The **CI width and sample size have an inverse square root relationship**. This means if you want to cut your margin of error in half, you need to quadruple your sample size. For example, if your margin of error is ±20% at a sample size of 20 you’d need a sample size of approximately 80 to have a margin of error of ±10%.

Computing CI

- CI are computed differently according to your population pool and the “nature” of the data

- Here we will focus on computing CI for continuous data

- As a case of study, we will work on rating scales obtained from SUS

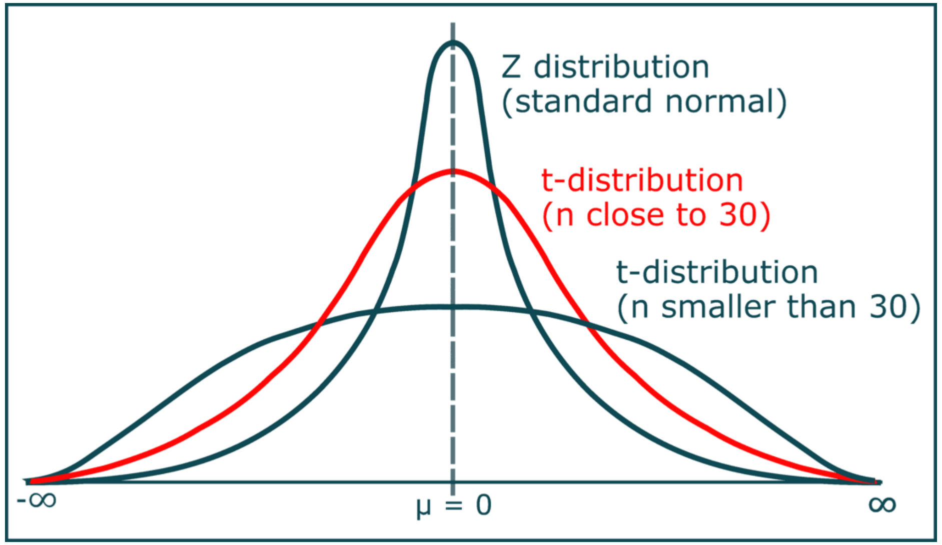
t-distribution

- The best approach for constructing a CI around numeric rating scales is to compute the mean and standard deviation of the responses then use the **t-distribution**.

- The t-distribution is like the standard normal distribution (also called z-distribution) except that it takes the sample size into account

- The t-distribution adjusts for how good our estimate is by making the intervals wider as the sample sizes get smaller.

- As the sample size increases (especially at or above a sample size of 30), the t-confidence interval converges on the normal z-confidence interval.

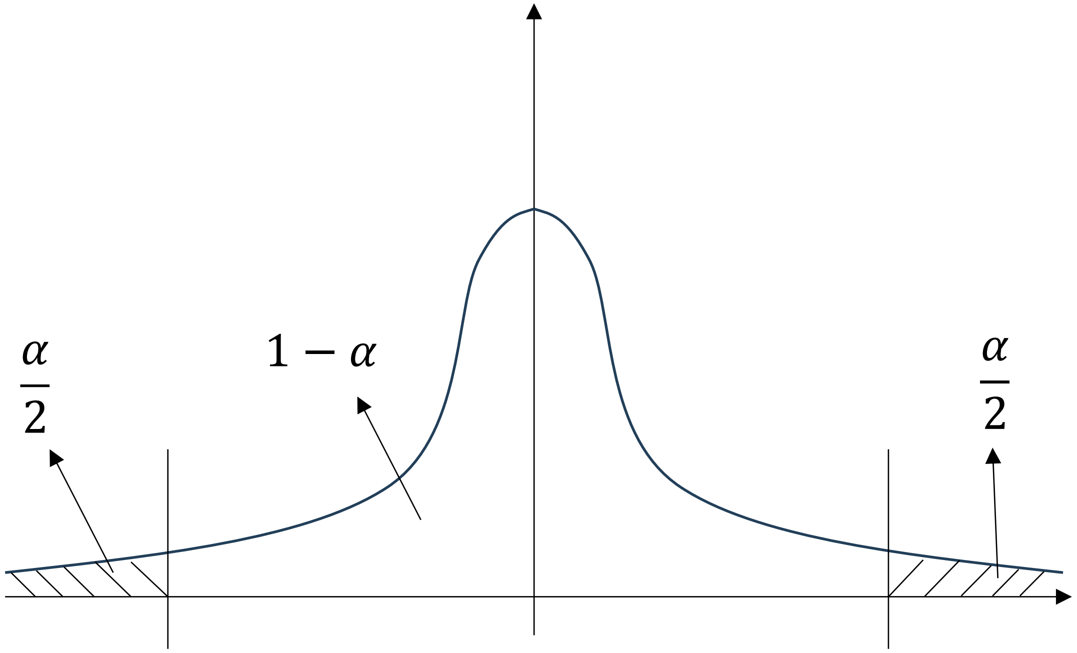


Computing CI with t

- The t-confidence interval takes the following form:

$$\bar{x} \pm t_{(\alpha,\mathrm{n}-1)} \frac{s}{\sqrt{n}}$$

- $\bar{x}$ is the sample mean,
- n is the sample size,
- s is the sample standard deviation, and
- $t_{(\alpha,\mathrm{n}-1)}$ is the critical value from the *t*-distribution for n−1 degrees of freedom and the specified confidence level $1-\alpha$
- α is called level of significance



- Note: usually you set α and n comes from the data

CI

- $t_{(\alpha,\mathrm{n}-1)}$ is usually obtained form a table like this.

- So, knowing the degrees of freedom we have and the significance level we want to achieve we just need to look up at the right value.

- Important: usually these tables are “one-sided”. So, we have to use $\frac{\alpha}{2}$ to obtain the right value

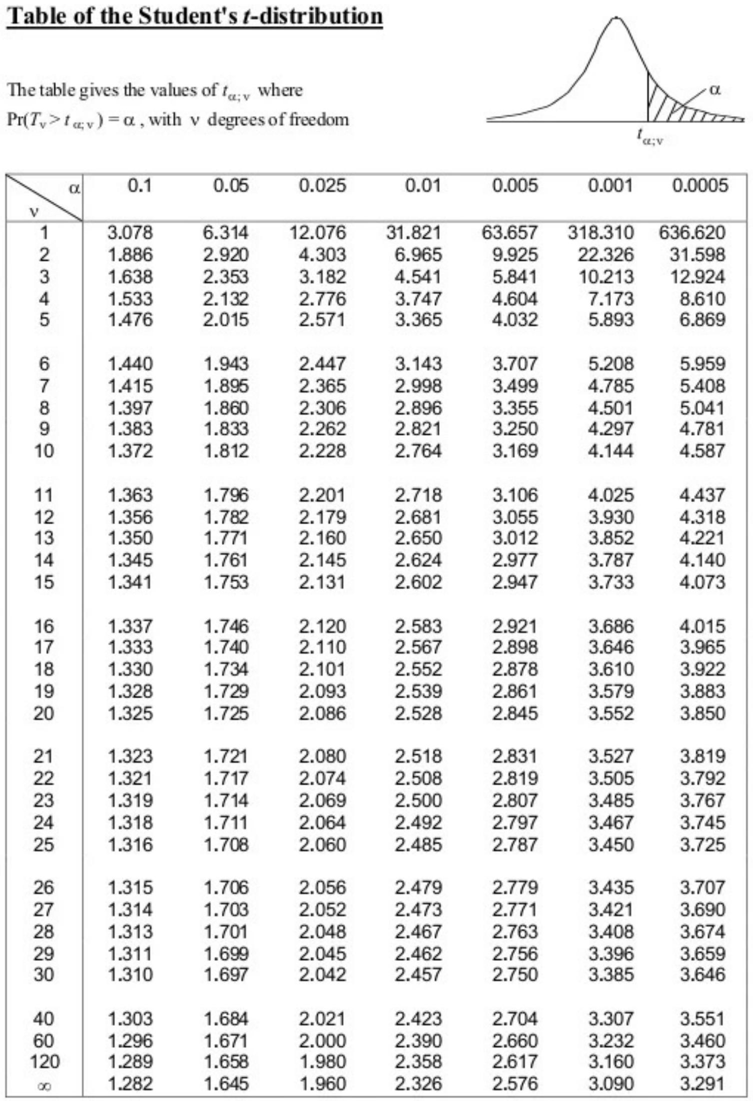


Table of the Student's *t*-distribution

The table gives the values of $t_{\alpha;\nu}$ where
$\Pr(T_\nu > t_{\alpha;\nu}) = \alpha$, with ν degrees of freedom

ν \ α	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	3.078	6.314	12.076	31.821	63.657	318.310	636.620
2	1.886	2.920	4.303	6.965	9.925	22.326	31.598
3	1.638	2.353	3.182	4.541	5.841	10.213	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	1.296	1.671	2.000	2.390	2.660	3.232	3.460
120	1.289	1.658	1.980	2.358	2.617	3.160	3.373
∞	1.282	1.645	1.960	2.326	2.576	3.090	3.291

Example

- For example, let's use the following scores from the System Usability Scale (SUS), collected when users rated the usability of an app:
  - 90, 80, 87, 95, 91
- From this data **we want to generate the CI at 95%.**
- It is easy to compute:
  - Mean: 88.60
  - Standard deviation: 5.59
  - Sample size: 5
- The CI at a confidence level of 95% results:

$$\bar{x} \pm t_{(\alpha,\mathrm{n}-1)} \frac{s}{\sqrt{n}} = 88.60 \pm t_{(0.95,4)} \frac{5.59}{\sqrt{5}} = 88.60 \pm 2.776 * 2.50 = \mathbf{88.60 \pm 6.94}$$

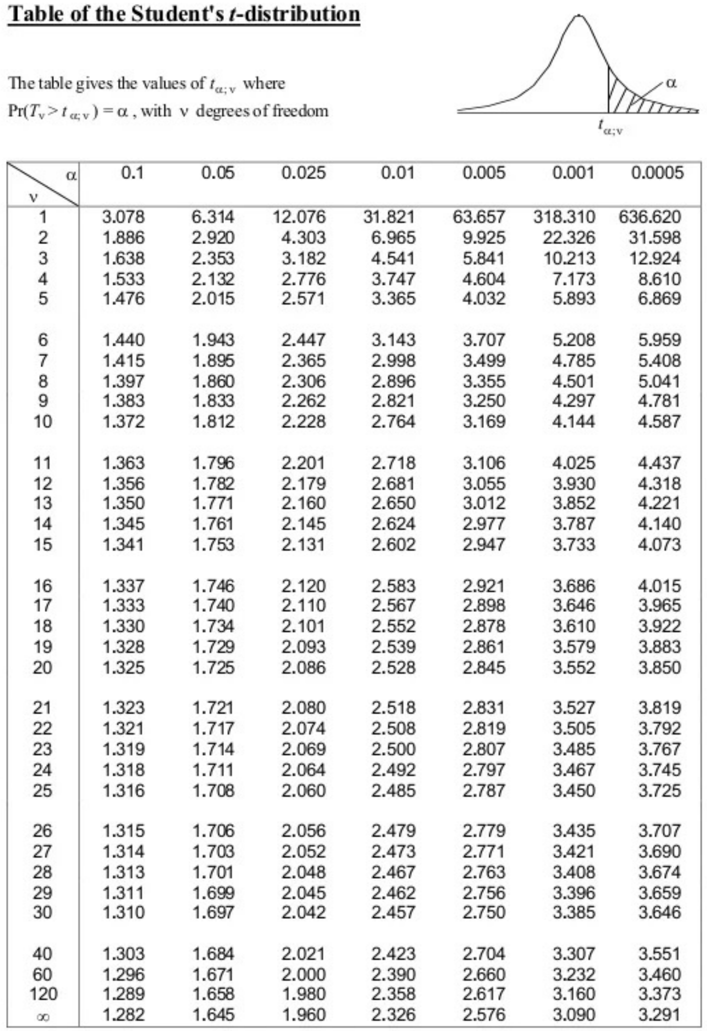


Table of the Student's t-distribution

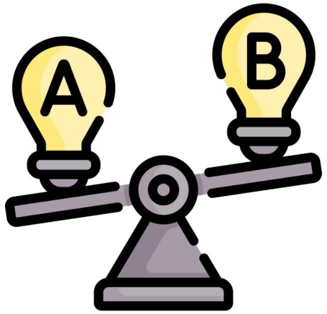
The table gives the values of $t_{\alpha;\nu}$ where $\Pr(T_\nu > t_{\alpha;\nu}) = \alpha$, with ν degrees of freedom

α ν	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	3.078	6.314	12.076	31.821	63.657	318.310	636.620
2	1.886	2.920	4.303	6.965	9.925	22.326	31.598
3	1.638	2.353	3.182	4.541	5.841	10.213	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	1.296	1.671	2.000	2.390	2.660	3.232	3.460
120	1.289	1.658	1.980	2.358	2.617	3.160	3.373
∞	1.282	1.645	1.960	2.326	2.576	3.090	3.291

This reads that we can be 95% confident that the true score is between 81.66 and 95.54

Comparing results

- What if we want to compare the usability of two products using SUS?



- Just because a sample of users from Product A has a higher average SUS score than a sample from Product B does not mean the average SUS score for all users is higher on Product A than Product B.

- Chance plays a role in every sample selection, and we need to account for that when comparing means.

- To determine whether SUS scores are **statistically** significantly different, you first need to identify whether the same users were used in each test (**within-subjects design**) or whether there was a different set of users tested on each product (**between-subjects design**).

Within-subjects comparison

- When the same users are in each test group you have removed a major source of variation between your sets of data.
- In such tests you should alternate which product users encounter first to minimize carry-over effects
  - If all users encounter Product A first, this runs the risk of unfairly biasing users either for or against Product A.

- To determine if the different scores are statistically different, we can use the **paired t-test**:

$$t = \frac{\bar{D}}{\frac{s_D}{\sqrt{n}}}$$

- $\bar{D}$ is the mean of the difference scores,
- s_D is the standard deviation of the difference scores,
- n is the sample size (the total number of users)
- t is the test statistic (look-up using the t-distribution based on the sample size).

Example

- For example, let's use the following SUS scores collected from 5 users rating app A and app B:
  - SUS of app A = 92, 77, 73, 98, 95
  - SUS of app B = 83, 79, 63, 90, 84
- It is easy to compute
  - Differences = 9, -2, 10, 8, 11
  - Mean of SUS A = 87
  - Mean of SUS B = 79.6
  - Mean of differences = 7.4
- It follows:

$$t = \frac{\overline{D}}{\frac{s_D}{\sqrt{n}}} = \frac{7.4}{\frac{5.46}{\sqrt{5}}} = \frac{7.4}{2.44} = 3.03$$

- This means that the usability of the two apps are: statistically different if we consider a level of significance greater than ~ 0.05

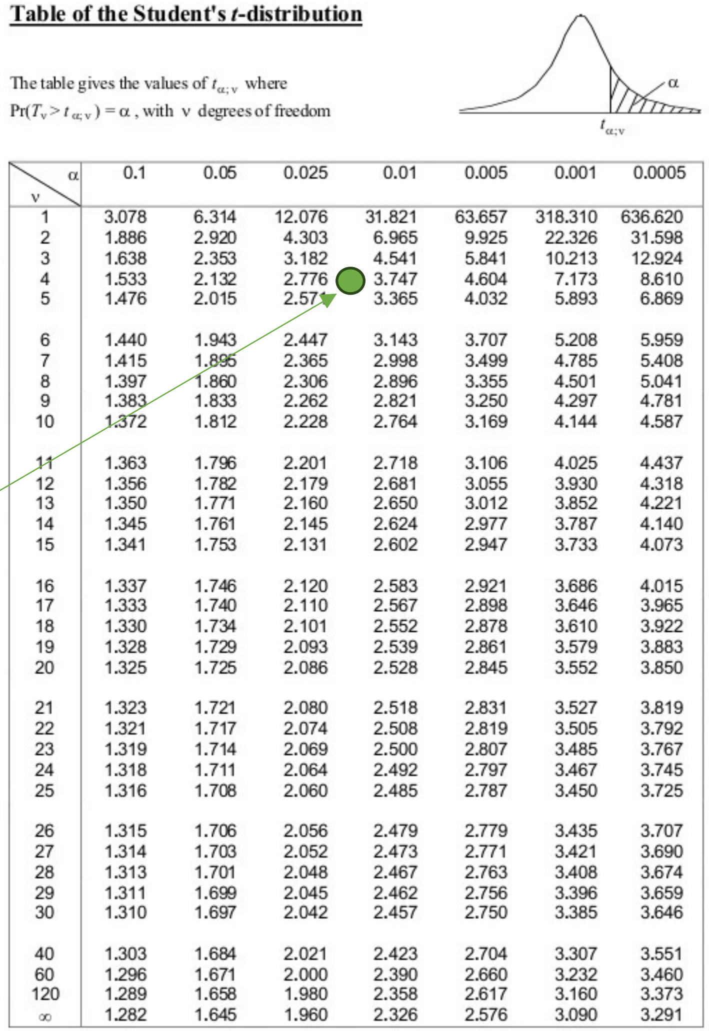


Table of the Student's *t*-distribution

The table gives the values of $t_{\alpha;\nu}$ where
$\Pr(T_\nu > t_{\alpha;\nu}) = \alpha$, with ν degrees of freedom

α ν	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	3.078	6.314	12.076	31.821	63.657	318.310	636.620
2	1.886	2.920	4.303	6.965	9.925	22.326	31.598
3	1.638	2.353	3.182	4.541	5.841	10.213	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	1.296	1.671	2.000	2.390	2.660	3.232	3.460
120	1.289	1.658	1.980	2.358	2.617	3.160	3.373
∞	1.282	1.645	1.960	2.326	2.576	3.090	3.291

Within-subjects comparison

- When a different set of users is tested on each product there is variation both between users and between designs.
- Any difference between the means must be tested to see whether it is greater than the variation between the different users.
- To determine whether there is a significant difference between means of independent samples of users, we use the **two-sample t-test** (also called t-test on independent means). It uses the following formula:

$$t = \frac{\widehat{x_1} - \widehat{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- $\widehat{x_1}$ and $\widehat{x_2}$ are means from samples 1 and 2
- s_1 and s_2 are standard deviations from samples 1 and 2
- n_1 and n_2 are the sample size from samples 1 and 2, and
- t is the test statistic (look-up using the t-distribution based on the sample size) with $n_1 + n_2$-2 degrees of freedom

Example

- For example, let’s use the following SUS scores collected from A users rating app A and other 5 users rating app B:
  - SUS of app A = 79, 76, 60, 86, 77, 90
  - SUS of app B = 92, 85, 80, 98, 95
- It is easy to compute:
  - Mean of SUS A = 78
  - Mean of SUS B = 90
  - Degree of freedom = 9
- It follows:

$$t = \frac{\widehat{x_1} - \widehat{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{78-90}{\sqrt{\frac{91.3}{6} + \frac{54.5}{5}}} = \frac{-12}{5.11} = -2.348 = 2.348$$

- The usability of app B is statistically better if we consider a level of significance (p-value) greater than ~ 0.05

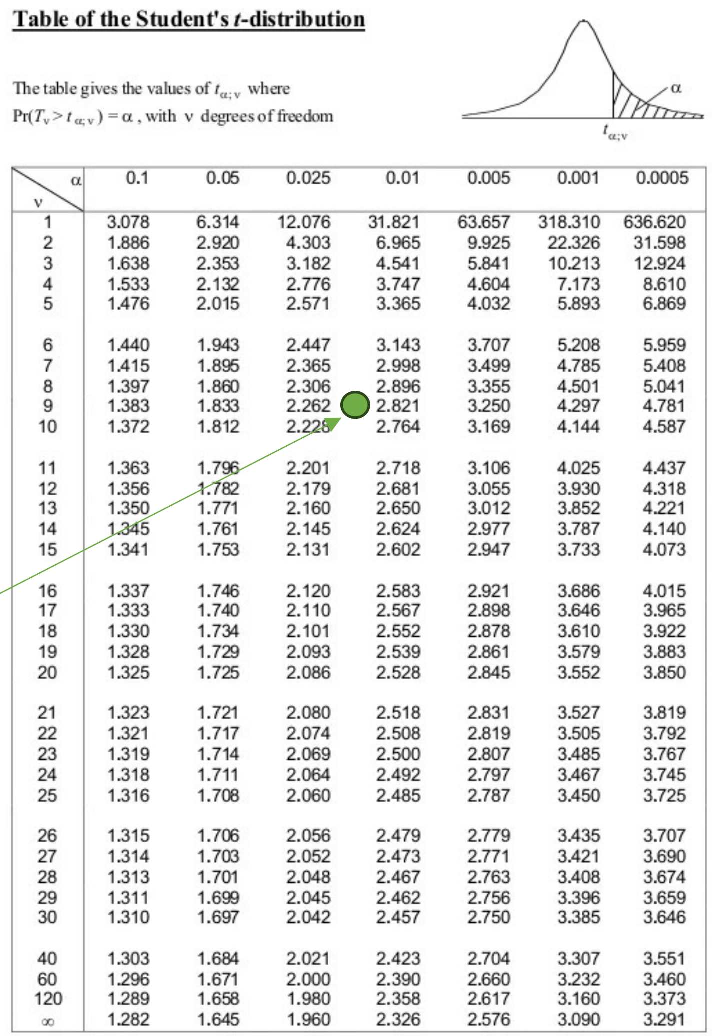


Table of the Student's *t*-distribution

The table gives the values of $t_{\alpha;\nu}$ where $\Pr(T_\nu > t_{\alpha;\nu}) = \alpha$, with ν degrees of freedom

α ν	0.1	0.05	0.025	0.01	0.005	0.001	0.0005
1	3.078	6.314	12.076	31.821	63.657	318.310	636.620
2	1.886	2.920	4.303	6.965	9.925	22.326	31.598
3	1.638	2.353	3.182	4.541	5.841	10.213	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	1.296	1.671	2.000	2.390	2.660	3.232	3.460
120	1.289	1.658	1.980	2.358	2.617	3.160	3.373
∞	1.282	1.645	1.960	2.326	2.576	3.090	3.291

References

- Sauro, Lewis – Quantifying the User Experience: Practical Statistics for User Research, 2nd edition – Elsevier
- Brooke et al., SUS: A Retrospective, Journal of Usability Studies, vol. 8, no. 2, 2013, pp. 29-40.