JUS
JOURNAL OF USABILITY STUDIES

Vol. 9, Issue 4, August 2014 pp. 152-172

Probability Plotting: A Tool for Analyzing Task Completion Times

Bernard Rummel
User Research Expert
SAP AG
Dietmar-Hopp-Allee 16
D-69190 Walldorf
Germany
bernard.rummel@sap.com

Abstract

Task completion time is a valuable metric for assessing or comparing the usability of a product. In online, unmoderated usability tests and other automated user behavior tracking methods, the large amount of time data that such tests yield must be carefully examined to exclude invalid data before further analysis can be meaningful. Other methodological challenges arise from the typically skewed statistical distributions in time data and varying task completion rates.

This paper describes how probability plotting, a technique developed in reliability analysis, can be applied to task completion times in usability tests. The method can be used to quickly identify outliers (such as speeders or cheaters), to verify distribution types, and to generate or verify hypotheses about the mechanism that may have generated the observed data distribution. In addition, the method affords using data from unsuccessful tasks and comparing completion times for tasks with different failure rates. Distribution parameters, in particular from the exponential distribution, can serve as additional usability metrics for comparative analysis. In this paper, probability plotting is applied to examples of task completion time data to illustrate its use in a usability test, and a link to a spreadsheet that automates the calculations is provided.

Keywords

usability metrics, efficiency, task completion time, statistical distribution, probability plotting

Introduction

Task completion time (TCT) is one of the most important metrics collected in usability tests, being gathered in the vast majority of studies (Coursaris & Kim, 2011; Sauro & Lewis, 2009). Typically it is used to assess an application's efficiency, which constitutes a core component of usability according to the well-known ISO definition (ISO 9241-11, 1998).

While it is fairly easy to record and tabulate completion times, upon close inspection, treating those times in a methodologically sound way can be surprisingly tricky. There are several statistical, conceptual, and practical questions that need attention. Usability practitioners need tools and techniques to efficiently tell valid data points from invalid ones, to aggregate information into meaningful parameters, and to identify and separate the various factors that contributed to the observed data. Probability plotting, a statistical technique developed in reliability analysis for examining failure time data of technical system components, is a useful tool to address those questions. The present paper will first discuss the methodological problems in TCT analysis, then describe the probability plotting method, and then provide an example and practical guidance on applying it. Because this is the first paper the author is aware of about probability plotting in the context of usability engineering, I could not avoid some theory, but I also have provided practical guidance and hints for application.

Methodological Questions in Task Completion Time Analysis

There are five questions to consider when analyzing TCT data.

Question 1: Which data should we include in the analysis?

Before beginning the analysis, we need to decide which data to include and which to discard as outliers. Some test participants may take an unusually short or long time to solve a task. In particular in unmoderated tests, how can we be sure that a participant worked on the task in the first place? They may have been distracted or been cheating in some way. A method is needed to distinguish invalid task completion times from legitimate ones.

Albert, Tullis, and Tedesco (2010) described a number of techniques to identify outliers in unmoderated online tests, such as "speed traps" to catch cheaters with impossibly short TCT, and threshold times to identify instances where test participants took unusually long. If a user takes more than 3 standard deviations longer than average, Albert et al. (2010) argued, they are likely not to have worked continuously on the task. Another technique involves sorting TCT and looking for "natural breaks" (p.112). We shall see below how probability plotting can help in spotting such breaks and in identifying rational threshold TCT values.

Question 2: How do we handle failed tasks?

Test participants sometimes do not complete a task. In that case, it is considered best practice to disregard time measurements from those participants (e.g., Sauro, 2011; Sauro & Lewis, 2012). However, the more competent participants who solve more tasks typically can be expected to also solve them faster—using their data may lead to an underestimation of the population's completion time ("survival of the fittest"). Albert et al. (2010) reported such a case and demonstrated that the difference can be substantial in the case of low completion rates.

In addition, when discarding failed participants' data we lose information: If a participant gives up, we know that up to that time he or she was not able to solve the task. However, to make use of this information, specific statistical methods are needed. Below we will discuss censoring, a concept developed in reliability analysis, and how it can be applied to usability test data.

Question 3: How do we account for system performance times?

System response time consumes a part of the overall task completion time. Subtracting system response time from user times requires some detailed event tracking, which is not always technically feasible.

If system response time is not negligible, this constitutes a major problem in comparative studies. When evaluating mobile applications running on different platforms, we want to separate the UI design and usability factor from any technical performance factor from the different computing environments. In a retest scenario, often the later test will run on a faster machine. Over a longer time period it may not even be feasible to run a retest on "historical equipment."

We shall see below how probability plotting can, by separating stochastic variations in TCT from systematic ones, help to separately assess system performance and UI efficiency.

Question 4: What parameters should we report?

The Common Industry Format for usability test reports explicitly requires reporting the "mean time taken to complete each task, together with the range and standard deviation of times across participants" as an efficiency metric (ISO/IEC 25062, 2006, section 5.4.4.2). Because task completion times typically do not follow a normal distribution, several authors more recently proposed a different approach.

No user can complete a task in less than 0 seconds, and typically, there are test participants who take a comparatively long time to do so. The average time therefore will be inflated by the lack of very fast times and by the long times spent by slower participants. In general, due to this asymmetry, skewed distributions are not appropriately described by the arithmetic mean alone (Cordes, 1993).

Albert, Tullis, and Tedesco (2010) as well as Sauro and Lewis (2012; see also Sauro, 2011) therefore recommend reporting median times or geometric means, depending on the sample size. Both parameters effectively correct the distribution skewedness found in most practical cases. Nevertheless, a more thorough investigation of the distribution type can be beneficial for two reasons.

The first reason is the accuracy of the parameter estimate. The median, being a "distribution-free" parameter, comes with rather large confidence intervals because the distribution information is not used in its estimation. The geometric mean can be estimated more exactly; however, that only holds if the data follow a lognormal distribution—we will see below how this assumption can be tested. For other distribution types that we will discuss, more specific parameters can be even more informative.

The second reason involves the "long tail" of the distribution. In lognormal, exponential, and Weibull distributions, it is rather common for the slowest 25% of test participants to take several times the average time to solve a task. Only if the distribution type is known, can the usability engineer correctly model completion times in order to eventually make valid predictions about usage times.

While several sources point out that time data are rarely normally distributed (e.g., Sauro, 2011; Sauro & Lewis, 2009; for an in-depth treatment see Luce, 1986), it is less clear what the distribution, or distribution family, might actually be. Cordes (1993) suggested using the gamma distribution for modeling TCT data, for its ability to cover a large range of "skewedness." However, the generating mechanism of the gamma distribution, as the sum of exponential-distributed times from basically the same distribution, is plausible only in a very limited range of experimental settings. Because in each usability test task, the number and structure of valid or invalid solution paths is vastly different, there may be several other candidate distributions that might apply. Bradley (1977) provided impressive simulation data illustrating how adding up random times from different, plausible distributions can drastically influence the overall time distribution even in apparently simple cases. The resulting distribution shapes may even be "bizarre" (p.147).

To summarize, we need a method to efficiently determine the distribution type of empirical task completion times. Once the distribution type is known, we can decide which parameters are the most informative and appropriate to report. In addition, those parameters, together with the distribution information, are essential for modeling and predicting usage times, in particular with regard to slow participants.

Question 5: How can we compare completion times?

The classical statistical approach for comparing task completion times is to run a *t*-test or analysis of variance. Both methods basically compare sample means in relation to the data's variance, assuming that the sample means follow a normal distribution and variances are homogeneous (i.e., roughly equal). Violation of these assumptions, although rather common in task completion times from usability tests, can typically be ignored due to the robustness of the methods. In addition, Sauro and Lewis (2012, found on p.66) provided a number of corrections to apply once certain violations are observed. If data are approximately lognormal distributed, logarithmizing times effectively normalizes the distribution; however, the log transformation complicates the interpretation of results. The lack of homogeneity of variances can be dealt with by adjusting the degrees of freedom in a *t*-test (Sauro & Lewis, 2012, found on p.70).

While these corrections are valid and effective, better knowledge of the distributions involved would certainly be helpful, for three reasons. First, Bradley (1975) mentioned "some spectacular degrees of nonrobustness" when sampling data from "L-shaped distributions," which are common in the time domain (p. 326). Understanding the distributions at hand may help control the risk of misapplying statistical procedures. Second, once a specific distribution type is verified, this can validate any corrective methods being used and, third, inform the usability practitioner which distribution parameters are the most informative and appropriate to compare.

A Solution Approach: Probability Plotting

The issues described above have led textbook authors (Dumas & Redish, 1999) and bloggers (Nielsen, 2004) to discourage the use of quantitative data by researchers who are not trained experts in experimental methodology. Clearly, simpler tools are needed to efficiently assess the overall quality and statistical distribution of TCT data, to derive meaningful parameters to describe this distribution, to deal with the fact that not all users may successfully complete a given task, and to efficiently identify and eliminate irrelevant information.

Reliability analysis, as an engineering discipline, offers a wide range of tools (Nelson, 1982; NIST/SEMATECH, 2012a) to address such questions. Typically, the reliability analyst deals with failures of technical parts. Such failures happen at random, but they often follow typical patterns and distributions over time. While individual failures are impossible to predict, the statistical distribution of failures allows fairly exact predictions about which percentage of parts will have failed at any given time. In addition, analyzing the distribution of failure times may provide insights about the mechanisms that generated those failures and the likelihood of failures to occur in a given time frame.

In usability tests, the reliability analysis perspective is turned upside down: The event under consideration is not the failure of a part, but the participant's successful task completion. However, a great deal of the mathematics involved in reliability analysis can be applied in a straightforward manner to TCT data collected in usability tests—especially probability plotting, a methodology that provides efficient visual tools to analyze TCT distributions.

Plotting methods have a long tradition in reliability analysis (Nelson, 1982), where they are used to quickly inspect failure times of parts. Plots are used to identify distribution types, weed out outliers, generate hypotheses on failure mechanisms, and estimate distribution parameters that allow quantitative predictions. As we will show, usability practitioners can take advantage of plotting methods that they can implement using standard office spreadsheet and charting tools.

The basic procedure begins with sorting the observed times in a task from all test participants, smallest (fastest) to largest (slowest). Table 1 shows an example: The Task Time column contains TCT in seconds for one specific task from 19 test participants. The fastest is ranked at position 1, the slowest on position 19. Those rank numbers roughly correspond to the respective user's percentile in the overall user population, with participant number 10 being located somewhere near the median. Because the slowest and fastest users obviously don't represent the 0th and 100th percentile, respectively, we need to apply specific corrections to estimate percentiles from ranks more precisely. These corrections are described below in detail; for now let's assume we already have determined the percentiles.

Next, the observed times are plotted against the corresponding estimated percentiles in a scatterplot. Typically, the data points are arranged along a curve because the slower users can take disproportionally longer to solve the task than the fast ones. The shape of that curve is specific to the statistical distribution type of the TCT.

The basic principle of a probability plot is to apply specific transformations to the *x* and *y* axes of the scatterplot, so that the curve is turned into a straight line. Because the shape of the original curve is specific to the distribution, so are the transformations that make it straight. In practice, plots are set up for each distribution type separately, with the *x* and *y* axes already transformed. When data are entered into such a plot, data points will align along a straight line, if and only if the data follow the respective distribution type (see Figure 1).

Once the plot is set up for the respective distribution type, all the usability practitioner needs to do is to paste in the data, sort them, and check whether or not they line up straight. If they do, the data follow the distribution the plot is set up for.

The extent to which the arrangement of scatterplot points is linear provides a measure for the goodness-of-fit of the distribution model. This can be quantified via a linear regression analysis of the plotting positions; the regression equation's R^2 parameter indicates the proportion of variance the regression model explains. R^2 then can be used for testing whether or not the observed data significantly deviate from the distribution model. Once the distribution type is identified, distribution parameters can be read from the plot or calculated from the regression model.

There are methods to analyze data with regard to various distribution families; the appropriateness of the methods depends on theoretical assumptions about the failure process (NIST/SEMATECH, 2012a). The present paper focuses on three distribution types that I have commonly found in usability testing: the exponential, Weibull, and lognormal distributions.

Dealing with Task Failure: Censoring

In many usability tests, not all participants complete the task they are asked to perform. Some give up; others may reach a predetermined time limit. Still others think they are finished but in fact did not complete the task. In reliability analysis, data from such a situation are called *censored*: At a given time, the part under consideration was observed but did not yet fail, so an exact failure time cannot be determined. However, because the part did not yet fail, failure time must be greater than the time of the observation. In the reliability analysis literature this situation is called Type I or *right-censoring*. If observations are made at different, individual points in time (i.e., each after a different time interval) this is called *multi-censoring*.

Similarly, in a usability test, if we observe task failure, the task completion time for that participant can certainly not be smaller than the time at which failure was observed. If the testing procedure involves a time limit, observed times are right-censored. If participants fail at individually different times, we have multi-censoring.

The procedure described below uses, in order to determine percentiles corresponding to observed ranks, the modified Kaplan-Meier (K-M) Product Limit recommended by the National Institute of Standards and Technology for multi-censored, small samples (NIST/SEMATECH, 2012b), as small samples are typical in usability testing. The K-M calculation procedure makes use of the rank information of times at which censoring occurred and estimates percentiles corrected for censoring based on this rank information. In case of a 100% task completion rate the K-M estimate yields the same value as other methods specifically designed for the uncensored case. In addition, even though it is designed for the multi-censoring case, it also converges to the simple right-censoring case if all failed participants fail at the same time, which may happen when a time limit is used.

However, to make legitimate use of this procedure, we need to make another important consideration about the reason why each individual failure occurred. The K-M correction for censored data assumes *non-informative* censoring. This means that the reason for censoring is independent from the effects under investigation, that is, the set of censored data is an *unbiased subsample* of the studied population (Cox & Oakes, 1984). Unfortunately, this is typically not the case in usability studies because it is rather unlikely to assume that participants who fail a task have the same proficiency level as participants who complete it. Treating our censoring case as non-informative would assume that we merely stopped observation at the time failure was observed, that is, the participant in principle might have solved the task immediately afterwards. There may be such cases, for instance, when a task trial was discontinued because of an unprovoked equipment failure. However, in most cases it will be safe to assume that failed participants would have needed much more time to solve the task because they gave up or made mistakes they were not aware of. Unfortunately, procedures for informative censoring are much more complicated than the ones for non-informative censoring.

The K-M procedure has convenient computational properties that mitigate this problem in our case. The procedure is based solely on ranking information, and the subsequent probability plotting procedure does not make use of the exact time assigned to failed participants—they are not included in the plot. This means that we can assign failed participants any time, as long as their *rank* in the overall sample is "correct."

Let's assume the failed participants, had they continued working on the task, would have needed at least as much time to solve it as the slowest successful participant. If they gave up or made a critical error in solving the task that they did not discover themselves, this assumption may be quite reasonable. Then their "correct" ranks would be at the slow end of the sample—irrespective of the actual time they might have taken, so we can actually assign them *any* time greater than the slowest successful user's TCT. Computationally, this leads to the same result as if they had continued working on the task but then run into an arbitrary time limit—which would be a legitimate case of non-informative right-censoring, where the K-M procedure can be validly applied.

It is up to the tester to decide whether or not the above assumption holds, that non-successful participants would have taken longer than any successful user. On an individual basis, this may or may not be plausible; in case of doubt, practitioners may want to calculate separate plots using different assumptions and then decide which variant informs their decisions best. Note that cases of "true" non-informative censoring (e.g., unprovoked hardware failure) still can be treated as such, that is, the time of failure can be directly used in the computation of plotting positions. For the other cases, where the assumption is made, it will make no computational difference whether the censoring was informative or non-informative.

To summarize, simply discarding the TCT of failed participants will over-estimate the UI's efficiency in the respective task, because we are discarding potentially slow participants. With the K-M procedure, we can correct this. If participants fail for reasons unrelated to the test (e.g., unprovoked equipment failure), we can assign them the time when the observation was discontinued (non-informative multiple censoring). If they give up or failure is declared because of critical flaws in the solution, we can assume those participants would have taken longer to solve the task than the slowest successful participant, if they had continued. We pretend they had run into a time limit and assign them a time greater than the slowest successful user's TCT, before calculating plotting positions.

Creating a Probability Plot

The procedure to create a probability plot is as follows (after NIST/SEMATECH, 2012b; A description of how to perform the calculations in the workbook associated with this article is described in Appendix A):

For the task under consideration, arrange the *N* participants' times or assigned times, in order from smallest to largest, and indicate which participants succeeded and which failed.

1. For each time, calculate the K-M estimate for the reliability function $R_i(t)$, as follows:
   a. Calculate a coefficient *C* with

      $$C = (N + 0.7) / (N + 0.4)$$

   b. For each observation *i*, calculate a multiplier M_i with

      $$M_i = (N - \text{Rank}_i + 0.7) / (N - \text{Rank}_i + 1.7)$$

   c. For each successful participant *i*, calculate the product P_i of all multipliers M_j where the participant was also successful and the time rank (*j*) is smaller than the present participant's (*i*). In other words, multiply the *M* of all successful participants who are listed above the current one.
   d. For each observed time *i*, the reliability function $R_i(t)$ is

      $$R_i(t) = C \cdot P_i$$

2. Create a scatterplot of the t_i and R_i, using only data from successful participants (the information from unsuccessful ones is factored in via the rank information used to calculate the R_i). Alternatively to R_i, its complement $F_i = 1 - R_i$ or, for log scales, $1/R_i$ can be used to invert axes for more convenient reading. The axes' orientations and scale settings depend on the distribution under investigation:
   - For exponential distributions, plot time on the horizontal axis using a linear scale and *R* on the vertical axis using a log scale. Alternatively, you can plot *ln*(1/*R*) on a linear scale; then distribution parameters can be read directly from the trendline's regression equation (see The Exponential Distribution, Mathematical model section).
   - For Weibull distributions, set both axes to log scale; plot time on the horizontal and ln(1/*R*) on the vertical axis. Alternatively, you can plot *ln*[*ln*(1/*R*)] on a linear scale; then distribution parameters can be read directly from the trendline's regression equation (see below).
   - For normal and lognormal distributions, on the horizontal axis, instead of $R(t_i)$, plot the inverse normal distribution value $z[F(t_i)] = z[1- R(t_i)]$.
     Set the vertical (time) axis to linear or log scale, respectively, to check for normal and lognormal distributions.
3. Add a trendline to the scatterplot (depending on the scale settings: exponential, linear, or logarithmic); it should appear as a straight line. If the data follow the respective distribution type, data points align more or less straight along the regression line.

Table 1 shows data used to generate Figure 1 as a calculation example. Note that unsuccessful participants are treated here as cases of non-informative censoring, so readers can better verify the calculation procedure.

Table 1. Data Used in Figure 1

Time Rank	**Task Time (seconds)**	**Task Success**	**Coefficient *C***	**Multiplier M_i**	**Kaplan-Meier Estimate for $R_i(t)$**
1	92	TRUE	1.015464	0.949239	0.963918
2	102	TRUE	1.015464	0.946524	0.912371
3	108	TRUE	1.015464	0.943503	0.860825
4	115	FALSE	1.015464		
5	118	TRUE	1.015464	0.936306	0.805995
6	133	TRUE	1.015464	0.931973	0.751166
7	146	TRUE	1.015464	0.927007	0.696336
8	156	TRUE	1.015464	0.92126	0.641506
9	170	TRUE	1.015464	0.91453	0.586677
10	172	TRUE	1.015464	0.906542	0.531847
11	193	TRUE	1.015464	0.896907	0.477018
12	196	TRUE	1.015464	0.885057	0.422188
13	198	TRUE	1.015464	0.87013	0.367358
14	254	TRUE	1.015464	0.850746	0.312529
15	276	FALSE	1.015464		
16	320	TRUE	1.015464	0.787234	0.246033
17	396	TRUE	1.015464	0.72973	0.179538
18	612	FALSE	1.015464		
19	877	TRUE	1.015464	0.411765	0.073927

Determining the Distribution Type and Outliers

The focus of the present paper is on distributions I have found in usability tests the exponential, Weibull, lognormal, and—for contrast—the normal distribution. Figure 1 gives a schematic illustration of those distributions. The vertical axis denotes the frequency at which one would expect to observe participants with the corresponding time on the horizontal axis. The curves are idealized from the distributions' density functions; in real tests drawing histograms of observed times would roughly produce the same shapes; however, to reproduce the curves exactly, we would have to run tests with very large numbers of participants.

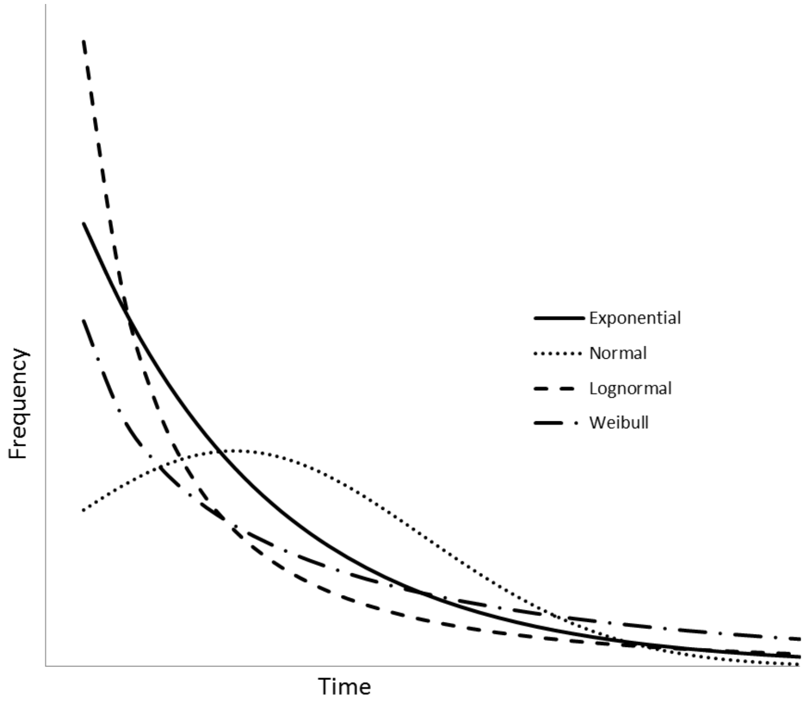


Figure 1. Schematic envelopes of hypothetic histograms for four distribution types.

Note that in Figure 1 all but the normal distribution start at high frequencies that decline over time. This means that the majority of participants will be faster than the average time, where the normal distribution has its maximum. However, the non-normal distributions have marked "long tails," that is, a number of participants take a relatively long time to solve the task. Those long TCTs inflate the average. In comparison to the exponential, the lognormal distribution contains a relatively larger number of fast participants, and the typical Weibull distribution contains a larger number of slow participants.

With the typically low number of participants in usability tests, it is not possible to create histograms like those in Figure 1, which are diagnostically meaningful—for a simple curve with 5 category bins and 5 data points per bin, you would need 25 tests participants. However, useful probability plots, like those in Figure 2, can be drawn from as little as 5 data points; their accuracy and diagnostic value does increases with the number of test participants.

Let's consider the dataset in Table 1. Figure 2 shows for this dataset a synopsis of probability plots for exponential, Weibull, lognormal, and normal distributions, respectively. Note the different scale settings; in order to achieve a consistent plot orientation 1/*R* or *F* is used in the plots, respectively. In addition, an offset time of t_0 = 90 s was subtracted from the times used in the Weibull and lognormal plots; reasons for this will be discussed below. All plots were generated using the K-M estimation method to account for the three observed task failures. Note that the plots show only data points for the 16 successful participants, but the plotting positions were calculated using data from all 19 participants.

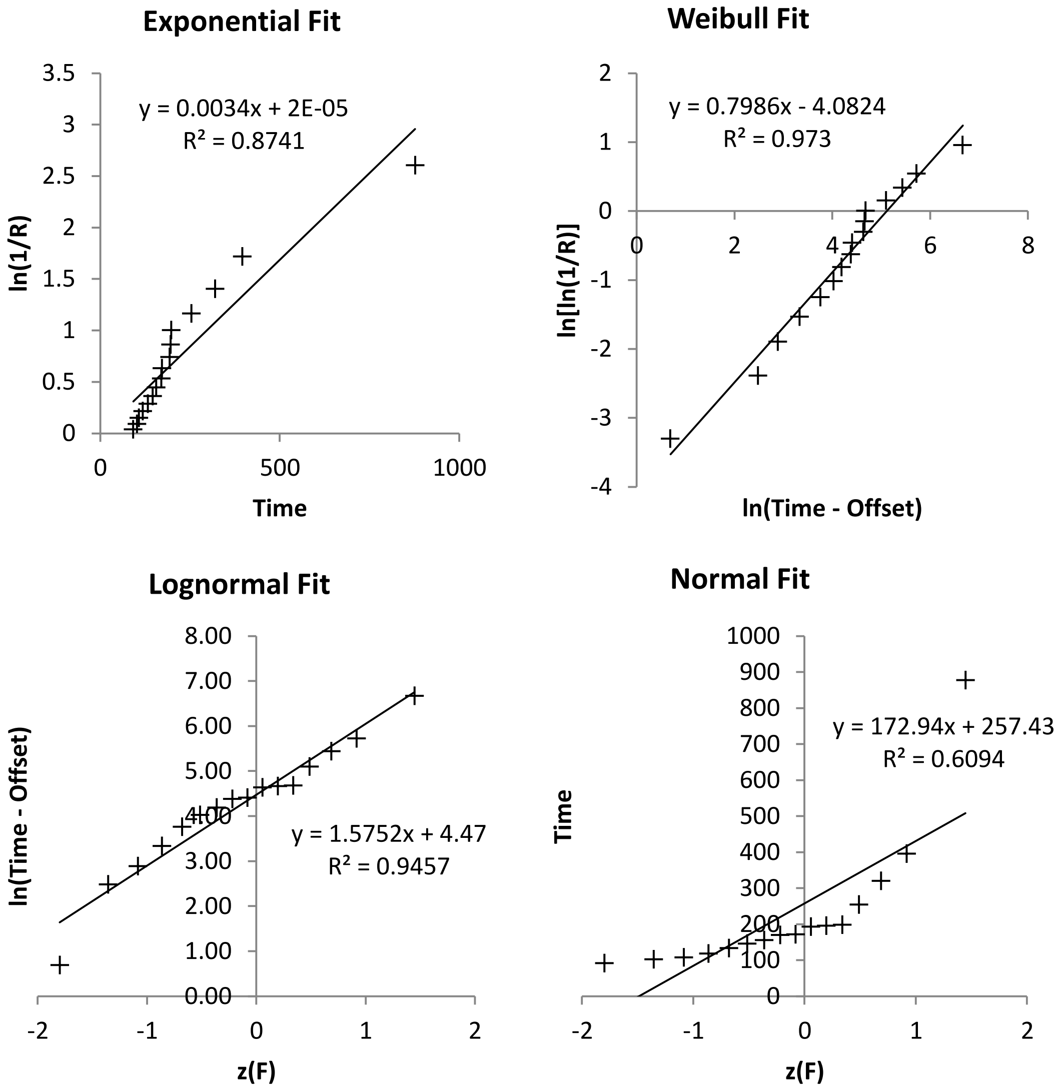


Figure 2. Probability plots of successful task completion times for four distribution types, with the three task failures used in the calculation of the plotting positions. Offset time t_0 = 90 s. All time is in seconds.

The exponential and normal plots show a more or less systematic curvature in the arrangement of data points. In the case of the Weibull and lognormal plots, data points are apparently randomly dispersed around the regression line, which indicates that both distributions are likely candidates for the distribution type at hand.

The model fit can be easily quantified using the R^2 (the multiple correlation coefficient) values of the regression equations given in the respective plots; $R^2 = 1$ would indicate a perfect fit. NIST/SEMATECH (2012c) provides a table of critical R values for significance testing of normal probability plots. The table is based on simulations carried out by Filliben (1975) who pointed out that it can be extended to other distribution families that afford probability plotting. If the observed R value is lower than the critical one, the p value given in the table indicates the likelihood of error when rejecting the hypothesis that data follow the respective distribution. Here, we can reject the exponential and normal distributions at the p = .05 level. However, note that the *R²* value does not distinguish between systematic and random deviations from the regression line, so the visual inspection of the data points' linear versus curved arrangement should have precedence for determining the distribution type.

An inspection of the data points shows that the curvature in the exponential and normal plots is mostly due to the three slowest participants, in particular the slowest one who spent about twice as long on the task as the next slowest successful participant. Could the slow participants be outliers? If we discarded them, the exponential and normal distribution models would also fit and could be used to describe the data. However, the close-to-perfect fit of the Weibull and lognormal model for *all* participants suggests that they all participated in the same process to generate those completion times.

The final decision whether or not a data point is an outlier obviously should be made considering all available information—the experimental situation, participant demographics, any incidents in the testing session, etc. However, the plots give a quick indication where to look and what to look for.

Detailed Distribution Analysis

Now that we're able to identify the distribution type underlying a set of task completion data and to weed out outliers based on this information, let's consider what it actually means to find a specific distribution type and what other useful information the respective probability plots contain.

While much of this information depends on the respective distributions, which we will discuss below separately, there is one aspect common to all probability plots. Once a distribution model with a good fit has been found, it can be used to predict success rates at any given point in time (and vice versa, times required to reach a certain success rate). *F*(*t*) can be interpreted as the percentage of participants who are expected to succeed in the task, which is 0 at *t* = 0 and increases over time. In any reasonably fitting probability plot, the regression equation therefore provides a comprehensive quantitative model of task completion rates over time. The time corresponding to *F* = *R* = 0.5 (*R* is the reliability function) actually corresponds to the median, that is, 50% of participants are slower than this time, and 50% faster.

This quantitative analysis of probability plots has three advantages over more traditional metrics, such as the mean, median, or geometric mean. First, making use of the censoring concept, we can include non-successful participants in the quantitative analysis, and thereby avoid efficiency overestimation—that is, estimating that the average time is faster than it actually is. Applying the regression equation in the Figure 1 Weibull plot to *F* = 0.5 (and adding the offset time) yields a time t_{50} = 195 s for the 50th percentile, which is larger than the median time (171 s) and geometric mean (187 s) from successful participants only. Note that the plot in Figure 1 was calculated assuming non-informative censoring; if we assign failed participants a TCT greater than the slowest successful participant's and recalculate, we come to a time t_{50} = 212 s. Second, in this calculation, we're actually making use of the distribution information, that is, we can select the best-fitting (and theoretically most plausible, see the Distribution sections below) distribution type in order to calculate the estimate. Third, estimates can be determined for *any* task completion rate or time, which is especially important in asymmetrical distributions—as to why this is important, we will discuss below in the context of the lognormal distribution.

Exponential Distribution

An exponential distribution of time intervals is characteristic for processes where events occur randomly, but at a constant rate over time and independently from each other. Examples for such processes are radioactive decay or the time intervals between telephone calls (Nelson, 1982).

Mathematical model

The exponential distribution is convenient for analysis because it is fully described by one parameter λ, which is also called the *failure rate* because it describes the (constant) proportion of parts failing in a given time interval. The cumulative density function of the exponential distribution for any positive x is given by

Equation 1: $F(x\ ;\ \lambda) = 1 - e^{-\lambda x}$

For applying this model to usability test data, a small modification is needed. Typically the observed distribution is *translated* by a constant t_0; however, in the probability plot, the regression line does not intersect the time axis at the origin, but at a positive time close to the completion time of the fastest participant or an expert. This can be modeled by the equation

Equation 2: $F(t\ ;\ t_0\ ;\ \lambda) = 1 - e^{-\lambda (t - t_0)}$

where F describes the proportion of participants completing the task at a given time t, and t_0 denotes the offset time. The failure rate λ in usability tests constitutes the *solution rate*, that is, the proportion of participants who solve a task per time interval after t_0 has passed.

To determine λ, an easy way is to calculate $\ln[F(t)]$ and create a linear scatterplot; λ then is the slope of the regression line (if R instead of F is plotted, the negative slope). Using the central limit theorem, 95% confidence interval boundaries can be calculated as

Equation 3: $\lambda_{CI} = \lambda_{est}\ [1 \pm 1.96/\mathrm{SQRT}(N)]$

Interpretation

Equation 2 mathematically describes a random process that sets in after a constant time t_0, and apart from this is fully described by the solution rate λ. Note that because the exponential distribution is translated by t_0, one can usually not observe it directly without previously knowing the value of this offset. Here, we infer t_0 and the solution rate λ from the plot. If TCT data follow an exponential distribution, we can attempt to relate its parameters to concepts in the real world.

Equation 2 segregates task completion times into a constant (t_0) and a random component. Now the process of a participant solving a usability test task can also be treated as two separate process parts being overlaid on each other, each contributing time and time variance to the overall process. Obviously there is a minimum time needed to perform a task that is determined by the technical response time of the user interface and the sensorimotor actions required to operate the user interface’s controls in the correct sequence. This time can be determined by cognitive modeling using software like CogTool (http://cogtool.com/), or empirically by having an expert click through the task repeatedly on the ideal solution path until completion times stabilize asymptotically.

On top of this time, test participants need additional time to understand the user interface, to discover functionalities and information, to make and correct mistakes, and to overcome the numerous usability hurdles present in most software. The time needed for this second part of the process is highly variable, being determined by the number and impact of latent usability impediments a participant actually experiences. This part of the process is essentially random and typically contributes most variance to the overall task completion time.

Both processes combined constitute the observed task completion times. A mathematically exact model would treat this as a *convolution* (Luce, 1986) of two different distributions. From a pragmatic perspective, however, we can simplify the model substantially. In many cases, the first part of the process contributes comparatively little variance to task completion times. In most lab usability studies, technical performance can be treated as a constant, and the variance contributed by participants’ different sensorimotor capabilities to click through the UI is rather small. In comparison to the much larger TCT variance contributed by the time needed by participants to understand the UI and find functions, it can be neglected. Under this assumption, we can treat the first process part as basically deterministic, and the second process part as basically stochastic. With this simplification, we can interpret the exponential distribution parameters as follows:

- The t_0 parameter describes the time needed to mechanically click through the user interface on the ideal usage path and for the user interface to react; therefore, it is a measure of *mechanical efficiency.*
- The λ parameter describes the task solution rate over time after mechanical click-through time has passed. It therefore is a measure of *cognitive efficiency*.

Note that this interpretation is idealized because the random part of Equation 2 includes *all* variance contributed by stochastic sub-processes, that is, including varying system performance and sensorimotor speed of participants; we're merely choosing here to neglect this contribution, assuming it is sufficiently small to justify doing so.

Weibull Distribution

The Weibull distribution is a generalization of the exponential distribution that allows modeling solution rates that are not constant but increase or decrease over time. Weibull distributions are widely used in reliability analysis to model fatigue processes. Weibull distributions have also been found in the analysis of users' dwell time on a website (Liu, White, & Dumais, 2010), where "failure" can be conceived of a user's leaving the site.

Mathematical model

The cumulative density function of the Weibull distribution for $t > 0$ is described by

Equation 4: $F(t\ ;\ k\ ;\ \lambda) = 1 - e^{-\xi}$
with $\xi = [(t - t_0)/\gamma]^{\alpha}$

For $\gamma = 1$, the Weibull distribution is identical to the exponential distribution. The parameter γ models a systematic variation in the solution rate, which in the Weibull case is no longer constant but increases ($\gamma > 1$) or decreases ($\gamma < 1$) over time. The α parameter is also called the characteristic life parameter, it denotes the time at which 63.2% of users will solve the task. The t_0 parameter denotes an offset time, as discussed with the exponential distribution model.

Interpretation

Verifying data against a Weibull distribution is useful when the exponential probability plot is not straight but curved, typically with slow users taking even longer to solve a task than expected under the exponential model (see Figures 1 and 2).

Parameters γ and α can be read from the plot, γ being the regression line's slope (note $\gamma < 1$ in Figure 2) and α the time at which it crosses the *x* axis. The model Equation 4 (above) also includes an offset time t_0 that cannot be read directly from the plot, as the logarithmic time axis is no longer invariant to translations in time. Nevertheless, following the rationale given above for the exponential distribution, we may want to uncouple the uninteresting "mechanical" process part from the more interesting cognitive part. Hence, we need to subtract t_0 from the raw data before calculating plotting positions. Conceptually, if t_0 describes the mechanical process time, it cannot be negative and must be smaller than any observed actual task completion time. We can derive an initial estimate for t_0 from an exponential probability plot, which typically is fairly linear on the left-hand side also in Weibull distributions. The estimated t_0 corresponds to the point on the time axis where a line drawn through the leftmost data points would intersect. Alternatively, the minimum observed time, or the time needed by an expert to click through the task on the ideal solution path, can be used. Because all those are mere estimates, another approach is to select t_0 so that the regression equation's R^2 assumes a maximum. This approach essentially assumes the model with the best-fitting parameters to be true. In the Figure 2 dataset, the latter approach was used: Using the minimum observed time (92 s) as the starting point, the plot was iteratively redrawn with decreasing t_0. R^2 eventually peaked at $t_0 = 90$ s.

Estimating t_0 requires some care because its selection has substantial influence on parameter estimates and the fit of the model. For practical purposes in usability testing, however, it is usually sufficient to verify that the decrease in solution rate over time is systematic, such as caused by fatigue, as opposed to a random fluctuation.

If TCT data follow a Weibull distribution, the tester should investigate for any systematic influence that may have shaped the TCT distribution. In technical reliability analysis, Weibull distributions are a typical indicator of fatigue. While in the technical domain fatigue leads to more failures, in usability tests we would expect participant fatigue to delay solutions, resulting in a decrease of the solution rate over time. While actual fatigue would take a while to take effect, loss of concentration and/or motivation is rather common in tasks that last over several minutes, in particular if the long duration is because of usability issues.

Lognormal Distribution

The lognormal distribution is also commonly found in the time domain. While its generation mechanism is not as clear as with the exponential or Weibull distributions, finding and confirming a lognormal distribution has important practical implications for statistical analysis.

Mathematical model

If logarithmized data follow a normal distribution, the untransformed data follow a lognormal distribution.

If the lognormal model fit for TCT data is at least good enough not to reject the hypothesis of lognormal distribution, this is important for statistical data analysis for it means that the analyst can confidently apply common parametric significance tests, which typically assume normal-distributed data, simply by logarithmizing data before calculating the test.

As with the distributions discussed above, an offset time t_0 can be included in the model by subtracting this time from the original times before logarithmizing.

Interpretation

The Central Limit Theorem states that a normal distribution can be generated by adding up evenly distributed random numbers. In an analogous way, one can assume lognormal distributions being generated by multiplying such random numbers because then their logarithms would behave in an additive way. Consequently, Kolmogorov argued in 1941 (in NIST/SEMATECH, 2012a) that lognormal-distributed failure times can be due to *multiplicative degradation*—when causes of failure are not independent (this would generate an exponential distribution), but add up in a multiplicative way, with one failure cause leading to subsequent ones. In usability testing, this would correspond to a task consisting of a number of mutually dependent subtasks, one's solution substantially facilitating the others' (an example is given below).

The lognormal model provides a justification for the frequently made recommendation to report geometrical means instead of averages (e.g., Sauro, 2011). The geometrical mean is the average of logarithmized values, transformed back to the linear scale. In the probability plot, (natural) log times are plotted over $z(R)$, which is symmetrical around $R = 0.5$, the median percentile. The intersection point of the regression line with the vertical axis ($z = 0$) therefore corresponds to the log median time, which in case of a good model fit equals the average log time. Inverting the log transformation yields the geometric mean. The slope of the regression line equals the standard deviation of (natural) log times (note: this is not the log of the standard deviation!). Averages and standard deviations of log times read from the plot can be used directly in *t*-tests, with the advantage that they are corrected for the censoring effect in cases in which failed participants are included.

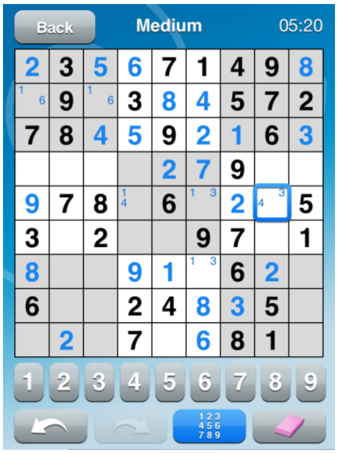
Note that these arguments, including the multiplicative degradation argument, hold only for the stochastic part of the model. If an offset time is included in the model ($t_0 > 0$), t_0 needs to be subtracted from original data before the log transformation and added again after retransforming calculation results. As in the Weibull distribution, t_0 cannot be read from the logarithmized time scale and needs to be plausibly estimated.

In case of a lognormal distribution, confidence intervals (CI) can be determined directly from the (normally distributed) log times. In case of censored data, the slope of the regression line gives an estimate for the standard deviation of logarithmized times, which can be used in the calculation. The end points of the confidence interval are calculated on the log scale, and then re-transformed to the original time scale. Alternatively, the prediction confidence band of the regression equation can be used for approximating CIs conveniently also for other percentiles.

Note that in lognormal-distributed TCT data, interpreting and communicating geometric means and confidence intervals involves a significant risk of misunderstandings. Symmetric intervals around a value in log data (such as the confidence interval or the standard deviation) are no longer symmetrical when transformed back to the linear scale: the right-hand side of the distribution will appear "stretched." While a person performing a task one standard deviation faster than average might not appear overly quick, a person performing one standard deviation slower may actually take several times as long as the average. In other words, a statistically rather common task completion time may look to the naïve observer as stark underperforming. Because the geometric mean in lognormal distributions is smaller than the arithmetic mean, this appearance will even be exaggerated. Consequently, great care must be taken when communicating such data.

Distribution Analysis Example

Sudoku :) is an Apple iOS app that affords playing the popular game Sudoku on the iPhone (Figure 3; app is available at http://ticbits.com/games/sudoku). The app provides four levels of difficulty for the game, two of which were used in the study. It also lists the 10 most recent completion times for each user and difficulty level separately. Data for two participants and two difficulty levels each (Easy and Medium), played on an iPhone4 (iOS5), are provided in Table 2.



2	3	5	6	7	1	4	9	8
1 6	9	1 6	3	8	4	5	7	2
7	8	4	5	9	2	1	6	3
				2	7	9		
9	7	8	1 4	6	1 3	2	4 3	5
3		2			9	7		1
8			9	1	1 3	6	2	
6			2	4	8	3	5	
	2		7		6	8	1	

Figure 3. Sudoku :), an iOS Sudoku game. Blue numbers entered by the user; the goal of the game is that each row, column, and 3 x 3 box contains the numbers 1 to 9. Small numbers are "candidates" (see text).

Figure 4 shows exponential probability plots of the data in Table 2, with all conditions combined in one plot for easier comparison. The exponential distribution fits well for three out of four conditions; let's consider those conditions first. The different difficulty levels are clearly reflected by the slopes of the corresponding trend lines (λ= 0.0328 and 0.0061 for participant U1, 0.0173 for participant U2). Slopes of the Easy condition are clearly steeper than in the Medium condition, with the 95% confidence intervals according to Equation 3 (in the Detailed Distribution Analysis section) clearly separated (U1 Medium upper λ CI boundary 0.0098, U2 Easy lower λ CI boundary 0.0142). Regression lines intersect the time axis at roughly the same t_0 in three of the four conditions (210 and 212 s for participants U1 and U2 in the Easy condition). The offset time for the U1 Medium condition is slightly larger (231 s) because in the Medium difficulty level the user has to enter more numbers than in the Easy level. The time difference is proportional to the difference in numbers of empty cells (43 vs. 49).

Table 2. Task Completion Times (in Seconds) in the Sudoku :) Example

Rank	U1, Easy	U2, Easy	U1, Medium	U2, Medium
1	202	183	261	491
2	216	225	272	757
3	220	251	276	776
4	227	252	339	859
5	229	254	339	905
6	237	267	344	909
7	250	284	352	960
8	257	294	421	974
9	260	312	572	987
10	284	338	660	1556

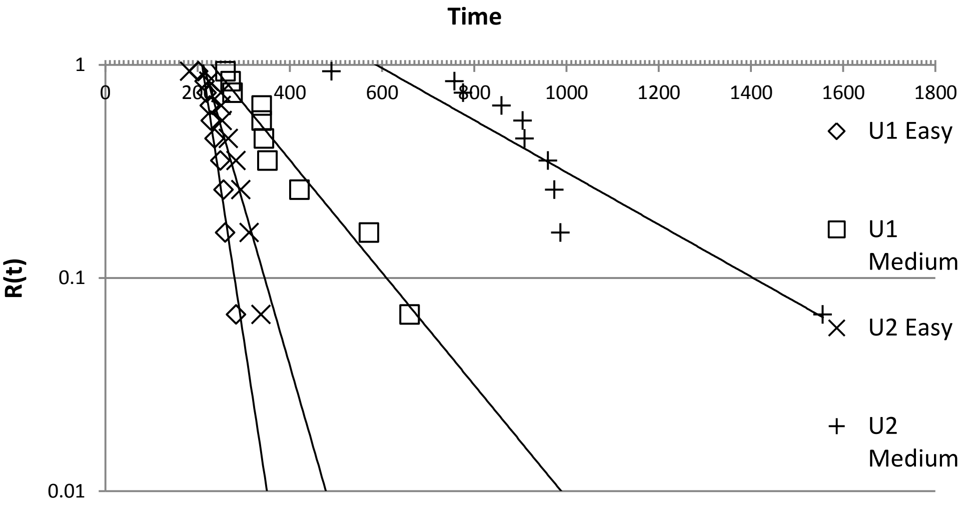


Figure 4. Exponential probability plots of task completion times for two participants at two difficulty levels each of an iOS Sudoku game. All time is in seconds.

The exponential distribution does not fit for participant U2 in the Medium difficulty condition. Upon questioning, U2 owned up to having played the game once at 2 a.m. in bed in a period of insomnia, resulting in a slow outlier time of 1556 s. There is also a second, fast outlier time of 491 s to be explained. Yet, when removing those two outliers the data points still are not linearly arranged.

Figure 5 shows a lognormal probability plot for U2, Medium difficulty after removal of the two outliers. An offset time $t_0 = 231$ s was included in the model based on the t_0 estimation from U1's exponential probability plot in the Medium condition. With R^2=.91, we do not have to reject the lognormal model (p = .05).

But how is it that the data in this condition follow a different distribution than in the others?

Sudoku :) lets the user enter multiple "candidate" numbers in each Sudoku cell, that is, numbers that might fit but where the user cannot yet decide. This feature is quite useful in the higher difficulty levels of the game, but in the easier levels, the game can be solved faster without using it. It turned out that U1 did not use the feature at all, and U2 only in the Medium difficulty condition—in all games but the "fast outlier" one. Entering the additional numbers obviously consumed time in the mechanical part of the process, which explains the greater t_0 and the outlier, but not yet the lognormal distribution.

When a user of *Sudoku :)* has identified the correct number for a cell and enters it as "final," the app automatically removes this number from the set of "candidate" numbers in the corresponding rows, columns, and 3 x 3 subgrids. Solving one cell thereby simplifies the solution of others: a clearly multiplicative mechanism that may very well have generated the lognormal distribution. Because it is only present when the candidate number feature is used, it takes effect only with participant U2 in the Medium difficulty condition.

What does this finding mean for the design of the *Sudoku :)* game? First, we can see that the candidate number feature comes with an efficiency cost—this might also be demonstrated with a *t*-test. Second, the plot shows that the feature actually is useful, because completion times are indeed accelerated. Figure 4 shows that U2's slower completion times are actually faster than we would expect in a mere stochastic process, where data points would follow a straight regression line. Its benefit however will only show when task difficulty is so high that the acceleration overrules the initial cost of typing more numbers. In the next section we will see how probability plots can help balancing interaction costs with gains in cognitive efficiency.

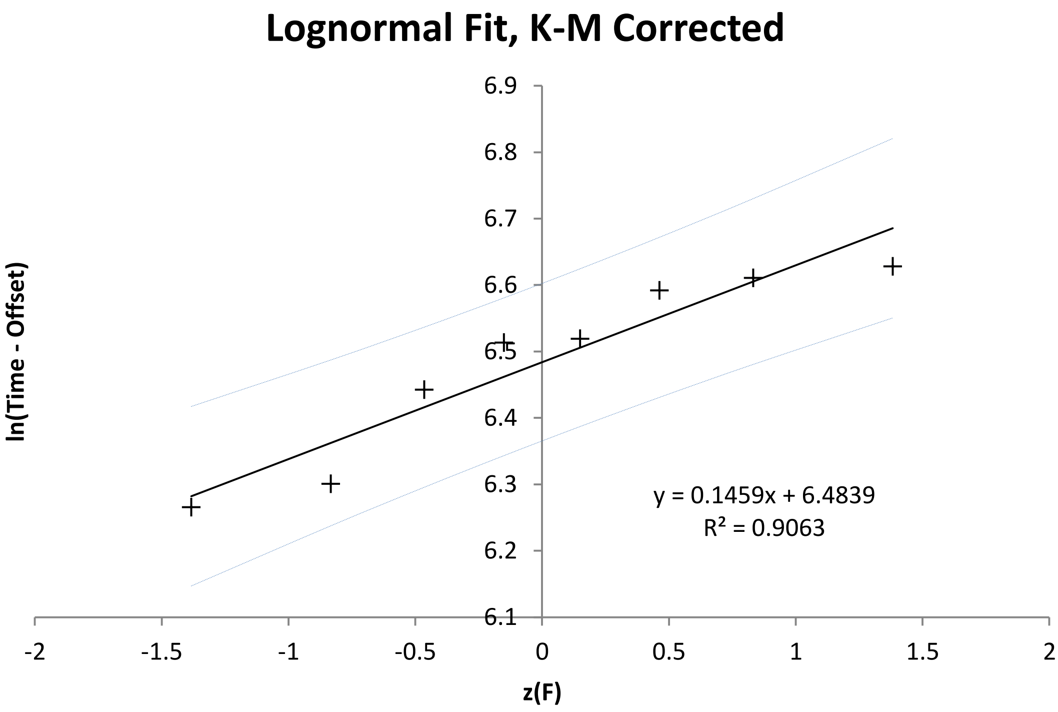


Figure 5. Lognormal probability plot of task completion times for participant U2, difficulty Medium of an iOS Sudoku game, with 90% prediction band. Offset time t_0 = 231 s. All time is in seconds.

Comparative Reasoning With Probability Plots

Probability plots visualize the entire distribution along with its basic parameters, such as the offset t_0 and the slope λ in an exponential plot. Figure 4 is an example how overlaid probability plots can be used to effectively compare task completion time data. Let's use a hypothetical example in Figure 6 to further discuss information that can be drawn from probability plots, in addition to more traditional approaches.

Figure 6 shows schematic probability plots for three hypothetical apps. Apps A and B have the same offset time t_0, but very different solution rates λ - A's cognitive efficiency is much better than B's, whose lower solution rate λ generates a standard deviation much larger than A's. Because statistical tests basically assess systematic differences *relative* to standard deviations, a large standard deviation can effectively conceal those differences. Further, if standard deviations—that is, λ—are substantially different, a basic assumption of most statistical tests based on variances can be violated, namely, the homogeneity of variance in the samples to be compared. In other words: The very fact that B is less cognitively efficient than A makes it hard to demonstrate this statistically.

Next, consider apps B and C. The differences are substantial: C has a better solution rate but worse system performance than A. However, because both have the same median value, we would not be able to spot any difference using this parameter alone.

The apps B and C illustrate a typical dilemma in UI design: Are clear, simple screens worth a longer click path? Should you rather invest in the system performance of C or in the interaction design of B? Because both factors are visualized separately in the plot, the design team can make informed decisions. While B and C have the same median time, with app C the risk of overly long task completion times is clearly lower. Investing in the system performance of B—not an unlikely suggestion in technology-fixated teams—does not make sense; B already has the better technical performance, as can be seen by its smaller t_0. Further, it would be ineffective: We would still see very long task completion times, as can be seen in the lower half of the plot.

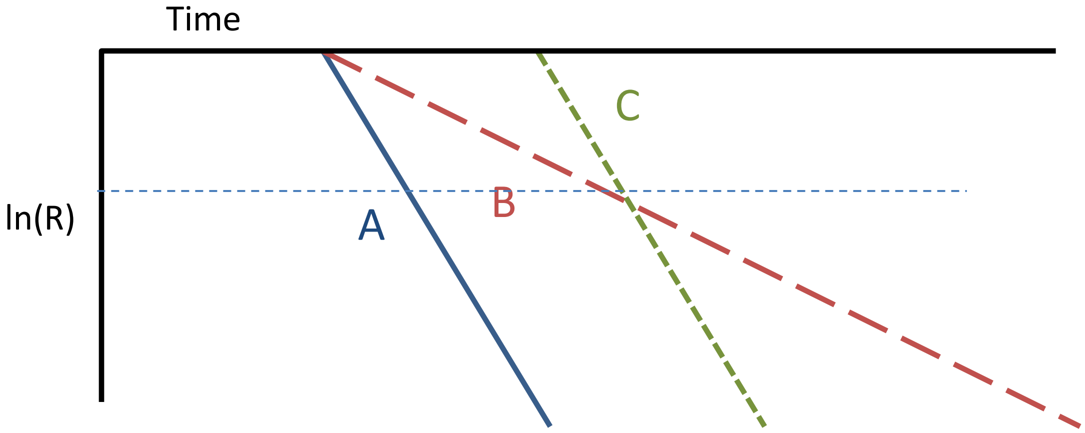


Figure 6. Schematic exponential probability plots for three hypothetical apps. The horizontal dotted line denotes the median (R = 0.5).

Discussion

Probability plotting effectively addresses the methodological problems outlined in the introduction. As a statistical instrument, it adds valuable information to established methods. It is an efficient, highly visual technique to quickly assess a data set's overall quality and distribution type. Because the process has been automated in a spreadsheet, distributions can be identified quickly (see Appendix A). Note that the approach is not limited to task completion time analysis; it can be used to inspect any data set for the underlying distribution type.

Albert, Tullis, and Tedesco (2010) define outliers as "extreme data points in data [that] are the result of atypical behavior" (p. 112). In probability plots, outliers are immediately visible in the plot as data points not appearing along the regression line. From a usability practitioner's point of view, the visual character of the technique is a great advantage. The overall arrangement pattern of data points supports quick checks of data quality as well as the generation of hypotheses and verification questions. In larger data sets, such as in data from unmoderated usability tests, plots can point the tester at speeding, cheating, or suspiciously slow participants as well as critical tasks that warrant closer inspection.

Note that the level of analysis conceptually goes beyond merely discarding extreme values. Identification of suspicious values is not limited to the ends of the distribution, because any larger deviation from the regression line will stick out. On the other hand, a very long completion time that still meets the regression line in an otherwise well-fitting Weibull probability plot, may very well be valid. If the suspicious data point belongs to the same distribution model as the rest of the data, it is likely to have been generated by the same process. Being able to demonstrate how an observation fits into the common picture can be important for a usability practitioner because outlier values always have the connotation of something not being quite right with the experimental setup or user recruitment.

Probability plots can also support detailed quantitative analyses that make efficient use of the data collected and provide information that is not available in other methods. Because a well-fitting plot provides a simple linear model of the data, usability practitioners can determine task completion rates for predefined times, and times needed to reach predefined completion rates, such as the median rate of 50%. Instead of discarding unsuccessful participants' data, and consequently over-estimating the UI's efficiency, the censoring concept lets practitioners make use of task failure information. This information factors in via adjusting the ranks used in the $R(t)$ calculation and the Kaplan-Meier estimation method of $R(t)$ itself. Making use of the exact time information that was observed and ranking information of those participants where no direct observation could be made, the plot extrapolates to model the *entire sample's* distribution and its parameters.

It should be noted here that the plots' regression lines do not stop at the observed task completion rate. For times longer than the longest observed completion time, task completion rates higher than the observed one can be read from the plot. Consequently, the UI's efficiency may still be overestimated, but to a much lesser extent than if unsuccessful users' data were simply discarded.

Insights gained from probability plots also inform the interpretation of other statistical techniques. Probability plots visualize the entire data set with regard to an expected distribution. Therefore, apart from the distribution fit, they offer a visual indication of which parameters are appropriate to describe the data at hand, as well as methods to determine such parameters. In addition to completion rates for predefined times, and times needed to reach predefined completion rates, other parameters pertaining to the distribution found can be evaluated. In exponential distributions, the solution rate λ and offset time t_0 are particularly interesting, as they allow usability practitioners to separately consider an application's technical and cognitive efficiency.

We have seen in the discussion of the exponential distribution that a large standard deviation is actually an indicator of a low solution rate—in fact, in a perfect exponential distribution, $\sigma=1/\lambda$. Thus, the standard distribution here is not a mere error term but actually an efficiency metric. It is not a lack of experimental rigor that causes large standard deviations in task completion times, but an inefficient user interface. Nevertheless, large standard deviations can conceal important differences in task completion times in statistical tests based on the analysis of those deviations, such as *t*-tests and analysis of variance. A more thorough analysis of the distributions involved helps understanding the magnitude and source of such effects.

Analyzing distribution types in task completion times also offers new perspectives in understanding task structures. First, identifying the distribution type allows generating hypotheses which generating mechanisms produced the observed distribution. As of today, our understanding of those mechanisms is basic at best; definitely more research is needed here. Nevertheless, and this is a second new perspective, being able to model task completion time distributions for individual tasks allows usability practitioners to model completion times for more complex, composite tasks. There is sophisticated methodology to do this (beyond the scope of the present paper) available in the reliability analysis literature; the basis is always an understanding of the individual subtasks' distribution type and of the interdependencies between subsystems, or here, subtasks.

Tips for Usability Practitioners

Probability plotting routines are available in various statistical software packages, but the plotting process can also be easily automated in a spreadsheet. Appendix A contains further information about the workbook available from this article.

Probability plots describe the sample, not the population; its parameter estimates become more accurate with increasing sample sizes. Therefore, it will prove most useful in high-data volume settings like unmoderated online usability tests and automated data collection schemes like Google Analytics. In those settings, it can provide valuable guidance for further detail analyses of the data and their underlying processes.

In traditional small-sample lab tests with 5–20 participants, probability plots are still useful for verifying data quality, for instance, when testing is outsourced or times are not automatically measured. In addition, the improved accuracy in parameter estimation, in particular if task failures occurred, pays off in all comparative settings. However, identifying distributions is rarely decisive in small samples—in particular, the Weibull and lognormal distributions often are both equally applicable.

I typically begin the analysis with the exponential probability plot—if there are no outliers and the model fit is good, the parametrical analysis can start right there. In addition, the exponential plot is useful for determining an initial estimate for the offset time t_0. Next, I check for the best fitting distribution type (i.e., the plot with the most linear stretch of data), and any outliers, on a synoptical plot like Figure 2. If there are outliers, I remove them and recalculate the plots. The next step, if the Weibull or lognormal plots fit well, is to fine-tune the offset time t_0 until R^2 peaks. As a starting value I use the t_0 estimate from the exponential plot, or if this is not realistic (e.g., because the distribution is not exponential), the minimum observed time.

If the lognormal model fit is acceptable, this is an indication that I may (after subtracting the offset time t_0) logarithmize times and use the log data in standard statistical procedures, like *t*-tests and ANOVA.

Other typical patterns are

- the exponential plot is flattened out and the Weibull plot is straight with $\gamma < 1$.
  Slow participants are even slower than expected by chance; inspect your records for signs of boredom or loss of concentration in test participants. As such effects are inherently problematic, analyze those records most carefully.
- the exponential plot is curved downward and the lognormal plot is straight.
  Slow participants tend to solve the task faster than expected by chance. There may be learning effects, or the task contains subtasks that are dependent from each other: solving one makes solving others easier. For summative, quantitative usability assessment, consider splitting your task into independent units.

References

Albert, W., Tullis, T,. & Tedesco, D. (2010). *Beyond the usability lab: Conducting large-scale online user experience studies.* Burlington, MA: Morgan Kaufmann Publishers.

Bradley, J. V. (1975). The optimal-pessimal paradox. *Human Factors, 17*(4), 321–327.

Bradley, J. V. (1977). A common situation conducive to bizarre distribution shapes. *The American Statistician, 31*(4), 147–150.

Cordes, R. E. (1993). The effects of running fewer subjects on time-on-task measures. *International Journal of Human-Computer Interaction, 5*(4), 393–403.

Coursaris, C., & Kim, D. (2011). A meta-analytical review of empirical mobile usability. *Journal of Usability Studies, 6*(3), 117–171.

Cox, D. R., & Oakes, D. (1984). *Analysis of survival data*. London: Chapman & Hall.

Dumas, J. S., & Redish, J. C. (1999). *A practical guide to usability testing.* Exeter, UK: Intellect Books.

Filliben, J. J. (1975). The probability plot correlation coefficient test for normality, *Technometrics*, *17*(1), 111–117.

ISO (1998). *Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability*. ISO 9241-11:1998(E).

ISO (2006). *Software engineering–Software product quality requirements and evaluation (SQuaRE)–Common Industry Format (CIF) for usability test reports.* ISO/IEC 25062:2006(E).

Liu, C., White, R. W., & Dumais, S. (2010). Understanding web browsing behaviors through Weibull analysis of dwell time. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 379-386). New York: ACM.

Luce, R. D. (1986). *Response times: Their role in inferring elementary mental organization*. Oxford psychology series No.8.

Nelson. W. (1982). *Applied life data analysis*. Hoboken, NJ: John Wiley & Sons.

Nielsen, J. (2004). *Risks of quantitative studies.* Retrieved December 2013, from http://www.nngroup.com/articles/risks-of-quantitative-studies/

NIST/SEMATECH (2012a). E-handbook of statistical methods. *National Institute of Standards and Technology.* Retrieved December 2013, from http://www.itl.nist.gov/div898/handbook/.

NIST/SEMATECH (2012b). Empirical model fitting—Distribution free (Kaplan-Meier) approach. In E-handbook of statistical methods. *National Institute of Standards and Technology*. Retrieved December 2013, from http://www.itl.nist.gov/div898/handbook/apr/section2/apr215.htm#Modified K – M.

NIST/SEMATECH (2012c). Critical values of the normal PPCC distribution. In E-handbook of statistical methods. *National Institute of Standards and Technology*. Retrieved December 2013, from http://www.itl.nist.gov/div898/handbook/eda/section3/eda3676.htm.

Sauro, J. (2011). 10 things to know about task times*. Measuring Usability*. Retrieved December 2013, from http://www.measuringusability.com/blog/task-times.php

Sauro, J., & Lewis, J. R. (2012). *Quantifying the user experience.* Waltham, MA: Morgan Kaufmann.

Sauro, J., & Lewis, J. R. (2009). Correlations among prototypical usability metrics: Evidence for the construct of usability. *Proc. CHI 2009*, ACM Press.

Appendix A

A Microsoft® Excel^{TM} workbook is available at this link for up to 49 task times; it can be extended for larger datasets. The formulas in the workbook require Excel 2010 or a later version.

The workbook implements K-M estimates for task completion rates < 1 and also contains critical R^2 values as provided by NIST/SEMATECH (2012c). The workbook contains four separate spreadsheets:

1. The first sheet, Distribution Check, contains an overview sheet with four small charts (see Figure 2) visualizing probability plots for the exponential, Weibull, lognormal, and normal distributions, together with the R^2 values of the respective regression equations. This sheet affords an at-a-glance inspection of the overall distribution type and identification of outliers.
2. The second sheet, Exponential, contains a detailed sheet with one large chart for the exponential distribution probability plot (see Figure 4).
3. The third sheet, Lognormal, contains a detailed sheet with one large chart for the lognormal distribution probability plot (see Figure 5).
4. The fourth sheet, Significance Levels Table, contains a significance table listing critical R^2 values for different sample sizes.

The latter two sheets are designed for detailed inspection and parameter estimation. Because those charts are larger, they afford visualizing additional information such as

- regression confidence bands for easier outlier spotting (removed from Figure 4), and
- horizontal percentile lines for the 50th (median; $R(t)$ = .5) or any other interesting percentile. The time coordinate of the line's intersection point indicates when the corresponding percentage of participants can be expected to solve the task. The median line also shows whether the numeric median and geometric mean differ from the modeled time.

Because the model fit for lognormal and Weibull distributions strongly depends on the offset time t_0, it can be manipulated right on the overview (first) sheet to consider the various plots' model fit with different offset times. Initial estimates for t_0 can be determined from the exponential plot on the second sheet. Outliers are data points way off the regression line. Note that in some cases, outliers in one plot fit the regression line well in another plot. In that case, consider the better-fitting distribution type than discarding any data. Outliers, if any, must be completely removed from the data and plots redrawn before continuing.

If on the first sheet the exponential plot's R^2 indicates that the exponential distribution model fits, you can interpret the exponential plot on the second sheet with all its parameters, in particular t_0 and λ.

If on the first sheet, the lognormal plot's R^2 indicates that the lognormal distribution model fits with offset time t_0 = 0, you can validly report the geometric mean as an efficiency metric; the third sheet is set up to support this. You can also run parametric statistical tests, provided you first logarithmize your data, which turns them into normal-distributed data, and are aware of the implications the transformation has for data interpretation. For $t_0 > 0$, note that the parameters derived from the plot describe the distribution of $(t_0 - t)$, not t; conclusions and calculations need to be made accordingly.

About the Author

Bernard Rummel
Trained in experimental psychology, Mr. Rummel has been working in the Human Factors, Usability and UI Design field for >20 years. After nine years at the German Naval Medical Institute, he joined SAP in 2000, where he is currently responsible for usability testing methodology.