Härdle, Wolfgang; Klinke, Sigbert; Marron, J. S.

Working Paper

Connected teaching of statistics

SFB 373 Discussion Paper, No. 1999,24

Provided in Cooperation with:
Collaborative Research Center 373: Quantification and Simulation of Economic Processes, Humboldt University Berlin

Suggested Citation: Härdle, Wolfgang; Klinke, Sigbert; Marron, J. S. (1999) : Connected teaching of statistics, SFB 373 Discussion Paper, No. 1999,24, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin, https://nbn-resolving.de/urn:nbn:de:kobv:11-10056250

This Version is available at:
https://hdl.handle.net/10419/61749

Connected Teaching of Statistics

Version : WH: Friday, 26.02.1999

W. Härdle
Institut für Statistik und Ökonometrie
Wirtschaftswissenschaftliche Fakultät
Spandauer Straße 1
Humboldt-Universität zu Berlin
10178 Berlin
Germany

Sigbert Klinke
Institut für Statistik und Ökonometrie
Wirtschaftswissenschaftliche Fakultät
Spandauer Straße 1
Humboldt-Universität zu Berlin
10178 Berlin
Germany

J. S. Marron
Department of Statistics
University of North Carolina
Chapel Hill, NC 27599-3260
USA

Abstract

Statistics is considered to be a difficult science since it requires a variety of skills including handling of quantitative data, graphical insights as well as mathematical ability. Yet ever increasing special knowledge of statistics is demanded since data of increasing complexity and size need to be understood and analyzed. Although this changing demand on educated statisticians is visible, our methods of teaching statistics follow essentially the ideas developed by our grandfathers in the fifties. An attractive and powerful new way of incorporating today's and future demands is via tools based on an intra- or the internet. In this article we suggest a set of criteria for effective web based teaching and propose the first net based approach to meet these criteria.

1. Introduction

The study of statistics is commonly considered difficult by students, since it requires a variety of skills including quantitative and graphical insights as well as mathematical ability. Yet an increasing number of people need facility with quantitative methods and students need to acquire statistical capabilities because they are confronted with more and more data sets to be understood. In addition these data sets grow rapidly in size and structural complexity. An example for such

data are the files that are collected on mobile phone applications, transaction records, etc. Despite these changing needs the teaching methods used have been surprisingly constant in recent years. An attractive and potentially powerful new way of updating current teaching methodology is via tools based on an intra- or the internet. In this article we suggest a set of criteria for effective web based teaching and propose the first net based approach to meet these criteria.

New technology is accepted more widely if its use is immediately understandable and easy for everybody. The same is true for teaching statistics in face of the new challenges in structure and size of data. It can only be effective if statistical methods are explained in a way that gives the student easy access to them. Two viewpoints are important for understanding this effective teaching, that of the student and that of the teacher. The new additional component of web based computing in teaching has an impact on both viewpoints. For example, large data sets, interactive graphics and on-line information were unusable for undergraduate statistics teaching a few years ago. Now easy availability of these features requires an update of the criteria behind "what is good teaching?" from both viewpoints.

The student will benefit from
Quick and easy access to methodology and data via browsers
Interactive examples since doing is one of the fastest methods of learning
Smooth transition from class room to home work to full scale statistical tools

The teacher will benefit from
Quick and easy broadcast of methodology and data via browsers
A user friendly environment
A powerful and flexible environment for dissemination of research

Many current teachers of statistics have a lot of inertia and reservation against changing teaching practice. Hence a stepwise plan towards smooth integration of web based teaching elements will gain the largest following. A series of steps through levels which allows gradual involvement and time commitment is:
Level 1: Display off the shelf class examples via a web browser. This requires only standard display equipment. It involves minimal effort by the instructor assuming ready made examples are available.
Level 2: Do examples as in Level 1, live on the web and give *interested* students the link coordinates of exercises, data or further programs and *suggest* some enriching examples they try on their own. This requires web access for most students and in the classroom. Again the ready made examples can be used and student questions will be minimal because no requirement is made of less capable students.
Level 3: Do examples as in Level 2, but assign homework using web based examples and methodology. This requires web access for all students and much more instructional support to address the inevitable large number of questions and problems.
Level 4: Become a developer of examples. This involves more time and energy on the part of the teacher (the amount depends on the friendliness of the environment and on the integrability of other web based documents), but also yields the most rewards in terms of the customizability that more creative teachers will want. Our goal for teachers who reach this stage is to provide tools which will maximize their individual creativity.

2. The solution

Our approach to meet the above criteria for teaching statistics in elementary courses is based on macros written in XploRe (**http://www.XploRe-stat.de**). Some examples are discussed in detail in Section 3. Here we develop the general framework and present the various outlets of XploRe for different platforms. XploRe is the interactive statistical computing environment which works equally well on single user machines, intra-networks and web connected clients. This is technically available via XploRe's server-client concept. The server (professor's machine) makes the course documents (data and statistical methods) available. The client (class room machine or students PC) connects to the server via the web without additional software downloads. The Java technology (**http://www.javasoft.com**) and standard web browsers enable this universal access despite the well known heterogeneity across hardware platforms. The overhead of earlier methods of handing out software and data sets is thus dramatically reduced.

The *quick and easy access to methodology and data via browsers* comes from the good integrability of XploRe data, macros and tutorials into web documents. Since most students are familiar with using browsers on the Internet, there will be no overhead of learning the environment, which would otherwise distract from their learning the desired statistical lessons.

A set of *interactive examples* is discussed in Section 3 below. These are intended to illustrate the point that interactive learning is very effective and all based on a standard browser front end (with a java swing class from SUN). For example, when a student is involved in choosing numerical parameters for a particular case study, the level of thought needed, followed by anticipation of the answer, which is then immediately displayed, results in a deep type of learning. In particular, *doing is the best method of learning.*

Some of the earlier approaches (e.g. **http://www.stat.sc.edu/~west/webstat/**, **http://www.stat.berkeley.edu/users/stark/Java/index.htm** and **http://www.ruf.rice.edu/~lane/stat_sim/index.html**) to web based teaching of statistics have included a *smooth transition from class room to home work* by allowing the student to use the Java applet shown in class also for homework. A natural next step to producing truly quantitatively equipped students is to also provide a *smooth transition to full scale statistical tools* that will continue to be useful long after the class is over. Because our examples are based on the statistical computing environment XploRe it is simple to move from class room examples to more elaborate data analysis.

Traditional methods of conveying data to students, such as writing on a chalkboard or piece of paper, have severe limitations, due to the effort involved at both ends of the process. Exchange of floppy disks allows software, and also larger data sets to be conveyed, but this involves a lot of overhead in terms of effort (e.g. control of homogeneity of hardware platforms) on the part of the teacher. The internet clearly allows *quick and easy broadcast of methodology and data via browsers*.

Many teachers of statistics have not learned web development skills, and perhaps may not have even learned other types of computational skills. For such potential users a *user friendly environment* means class examples must be already completely developed and ready to use. We offer class room ready examples on (**http://ise.wiwi.hu-berlin.de/statistik/lehrmaterial/statmat.html**).

Other teachers of statistics will be higher end users, who have their own ideas for class examples, or else would like to customize those that are provided. For them a *user friendly environment* means the existing examples are coded in a very high level language, which is easy to modify, and provides a convenient basis for other types of development. XploRe is matrix (array) based and thus development occurs at a higher level than is available from Java programming. An important advantage of XploRe over other high level statistical languages such as SPLUS (**http://www.mathsoft.com/Splus**), GAUSS (**http://www.aptech.com**) or STATA (**http://www.stata.com**) is that XploRe macros may be automatically converted to web transparent methodology via an HTML translator.

Teachers who wish to modify the given examples, or develop their own, will need more than just a user friendly environment. They will also need a *powerful and flexible environment*, which contains a wide range of quickly usable tools. XploRe has a wide range of statistical tools with the possibility of specialization for different fields like finance, econometrics, etc. Java based approaches to specializing software for teaching cannot provide this full scale since they are based on combinations of the limited set of fixed applets available in the toolbox provided by the applets' constructor.

Statistical Technology on the net

Three hardware platforms are in widespread use for statistical computing and graphical data interaction: Macintosh, UNIX, and Windows. The first has a simple graphically oriented user interface and allows highly interactive dialogues with data. UNIX is used for high-speed and distributed computing but is often less satisfactory in graphical interaction. Windows aims at facilitating both high-speed computing and graphics but is weaker at present than UNIX for Internet access. Distributed computing is simply not possible under Windows unless one uses certain add-ons. An overview of current internet technology and statistics is given by Symanzik (1998) (http://www.galaxy.gmu.edu/~symanzik)

Many software platforms for statistical computing exist but are unfortunately not easily interchangeable. The reasons for this include the history of software development, the targeted user groups, and the optimization of certain software for specific hardware configurations. The original version of GAUSS (**http://www.aptech.com**), for example, was optimized for INTEL chips and, therefore, could not be transferred to the Mac or UNIX platforms. Now GAUSS is available on UNIX, but the UNIX version does not have a graphical device that allows, for e.g. interactive changes in the layout of graphs. SPLUS (**http://www.mathsoft.com/Splus**) was developed for UNIX systems and was only later transferred to PCs. Consequently, the PC version is different from the UNIX version. EVIEWS was developed for DOS and is now available for Windows but not

for UNIX or for the Mac. TSP is a DOS program and is not easily transferred to a Windows/NT platform. SPSS exists for Windows but has still a batch structure that makes many mouse clicks necessary in order to generate implicitly the batch commands. STATA (**http://www.stata.com**) , SAS (**http://www.sas.com/**) and SHAZAM (**http://shazam.econ.ubc.ca**) are unusual in that mutually compatible versions exist for all platforms. Besides the software that we mentioned here as examples, there are many other platforms which also share the property of heterogeneity.

Heterogeneity of software platforms creates relatively few problems if there is no need to exchange programs. Exchange of graphs, document files, and ASCII-based data sets can be carried out by FTP, provided that the user has the appropriate graphics plug-in and document reader (e.g., Ghostscript or Acrobat). However, there is also a need for exchangeable computer programs for implementing advanced statistical methods, as these are becoming increasingly complex mathematically, and writing the necessary programs can be a difficult and time-consuming task, which puts it effectively out of reach in many cases.

Graduate-level instruction in statistics provides one example of the usefulness of exchangeability. It is not unusual for a faculty member at one university to give a short course at another. In some cases, a faculty member at one university may use electronic communication to present a course at several geographically dispersed locations. Calculation of an estimator may require heavy computing that is available on the researcher's home machine. During the course, modifications of this estimator and different applications may be discussed, and these may require access to the software at multiple locations. Exchangeability of software is necessary to enable students at all locations to carry out computational and empirical exercises that the instructor has prepared at his own university.

Collaboration among researchers at different locations provides another example of the desirability of exchangeability. In this case, the goal is to enable each collaborator to carry out computations using the same software. Ideally such cooperation should be based on a pool of easily accessible software and computing power for all parties. For effective progress on a project that involves heterogeneous hardware and software, it is desirable for partners to have the ability to contribute methods despite being at different locations and working with different computing environments. In addition it may simply be a problem for a researcher who is a visitor in another establishment to be able to continue using his own programs.

On the other hand, heterogeneity of software does have the important advantage of enabling a developer of new methods to choose the software system that is best suited to the problem under consideration. Therefore the problem of exchangeability should not be solved by standardizing statistical software but by making software from different sources accessible to diversely equipped users.

3. Teachware Quantlet Examples

Example 1

This example illustrates that gathering a *random sample* is different from "just choosing some", i.e. proper random sampling requires some mechanism to ensure that "all samples are equally likely", which is quite different from "arbitrary human choice".

This is intended for a classroom setting, where students are asked to *write down* (to avoid changing during the course of the exercise) a "randomly chosen" number among 1, 2, 3, 4. Since most people choose 3, and most of the rest choose 2, the resulting distribution is quite far from the random uniform distribution.

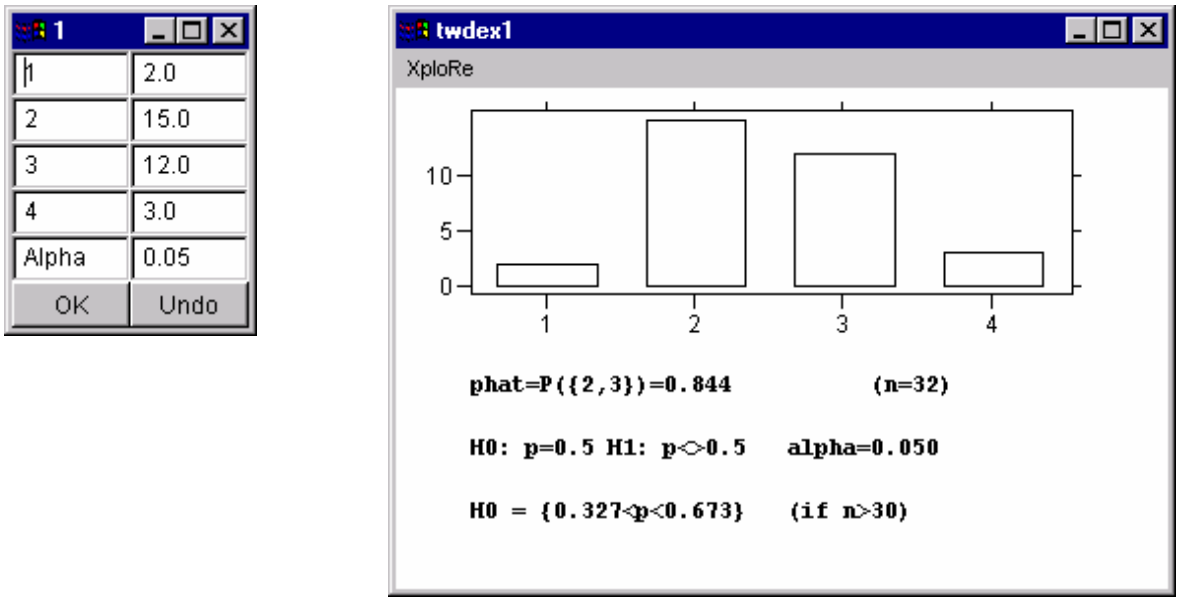


Figure 1: Demonstrates Example 1, "random" is different from "just some". Left: Menu to enter the number of people who have chosen 1, 2, 3 or 4 and α for the confidence level. Right: Resulting window with graphical and text output, which assesses the amount of "randomness" of the entered numbers.

Nonrandomness of the chosen numbers is demonstrated "on line" by entering numbers (the actual counts for each of 1,2,3,4) into the textbox on the left. The numbers currently shown come from an actual class. Clicking "OK" generates the result on the right, which is a bar graph showing the counts (for easy visual interpretation), together with a text window summarizing the results of some simple statistical analysis, including a confidence interval for the proportion of 2's and 3's. While confidence intervals have likely not been explained at this point in the course, it can simply be said "this range gives a feeling for the variability in the data, and it will contain the given proportion if these numbers are actually random". This provides motivation and interest for the time when confidence intervals and hypothesis tests are developed (when this example should be revisited).

For level 2 or level 3 teaching, students should be encouraged to experiment with changing the input values, and watching the change in the interval bracketing 0.5. For example, what happens for (25, 25, 25, 25)? For (0, 0, 100, 0)? What is the difference between (10, 70, 0, 20) and (10, 0, 70, 20)? Students could be challenged to "explore the boundary between random and not" by finding data vectors which are near each other, but give opposite test results. This example may be repeated as many times as desired and may be run directly from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface (swing classes have to be in the corresponding JAVA directory), enters **library("tware")** and then enters the quantlet name **twrandomsample().**

Example 2

This example is intended to illustrate the concept of a p-value for hypothesis testing. For simplicity, it is done in the context of the Binomial(n,p) distribution. The hypothesis tested is: H0: p < c, for some choice of c.

The example starts with a menu of input boxes, which allows input of:
- The binomial parameter, n (number of Bernoulli trials),
- The binomial parameter, p (probability of success in the Bernoulli trials),
- The observed binomial value, x.

The intention is to motivate the p-value, i.e. the "observed significance level" for the observed value x, through graphical display of the region represented by P{X>=x}.

The main graphic is a bar chart, where bar heights show the Binomial(n,p) distribution. The bars corresponding to the event {X>=x} are shown with a black outline, which gives a visual impression for this probability. There is text added to the graph, which gives the numerical value of this probability.

There are two check boxes, which allow choice of the displayed probability as either P{X>=x} (the usual "p-value") or as P{X=x} (another candidate for "observed significance"). See discussion below about this.

In class it is recommended to demonstrate:

i. When the observed value x becomes larger, the p-value decreases, i.e. the evidence against H0 becomes stronger.

ii. When p becomes larger, the p-value increases, i.e. the evidence against H0 becomes weaker. This makes sense, since then the null hypothesis has a better chance of explaining the observed value.

iii. To see why the p-value is P{X>=x}, and not P{X=x}, use the checkboxes mentioned above. The parameters p = .5, and x = n / 2 are recommended, and then take several values of n, such as n = 10, 40, 160. These show that P{X=x} has the problem that it depends strongly on n, and worse gets small even when there is clearly no strong evidence. against H0. On the other hand, P{X>=x} is stable for increasing n, and stays large when there is no strong evidence.

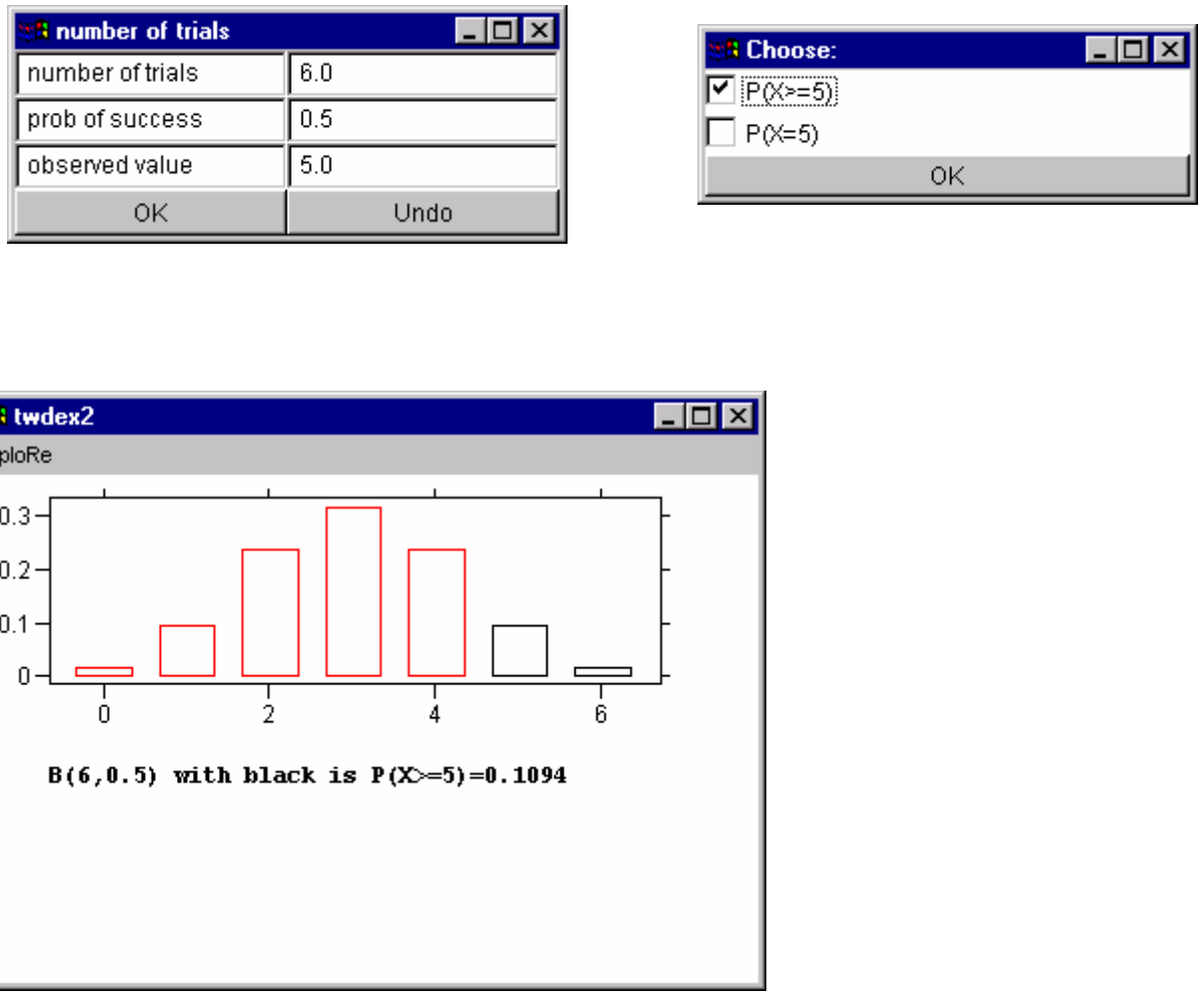


Figure 2: Demonstrates Example 2, how p-values work. Upper left: Menu to choose the Binomial distribution parameters: the number of trials, n , the probability of success, p, and the observed value, x. Upper Right: menu allows choice between P(X>=x) or P(X=x). Lower resulting plot under JAVA with area representing the p-value, P(X>=5), shown as black outline boxes.

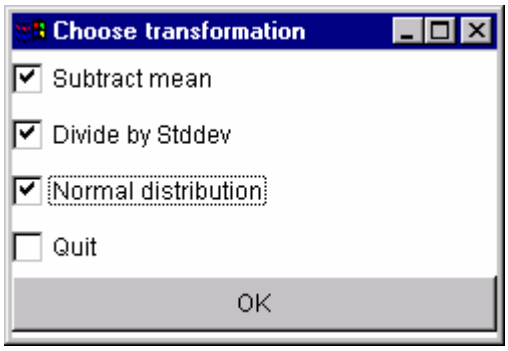
This example may be run from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface, enters **library("tware")** and then enters **twpvalue()**.

Example 3

This example illustrates two points. First how the normal distribution provides a good approximation to the binomial for large n. Second why it is both important

and natural to subtract the mean, and divide by the standard deviation, when doing a normal approximation.

The example starts with an overlay of three theoretical probability histograms (bar graphs where heights are probabilities), representing the Binomial distribution, with a fixed value of p, say p = 0.6, and with three choices of n, say n = 10, 20, 40, as shown in the left panel of Figure 3. The instructor points out that there is a common "mound shape" to the three graphs, but that they are not close to any fixed distribution, and will not get closer to anything as the sample size n grows, since the probability mass moves to the right. However the effect can be understood, and perhaps adjusted for, by the development of the concept of centerpoint of a probability distribution, e.g. the mean.



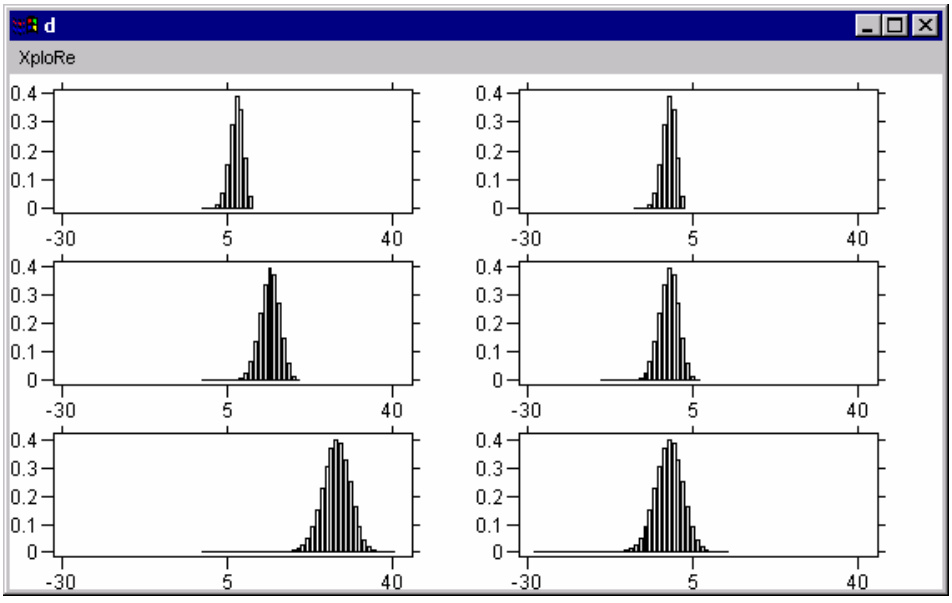


Figure 3a: Demonstrates Example 3, Standardization and Normal approximation of the Binomial. Upper: menus for controlling the transformation of the Binomial

distributions. Lower: main graphic window, showing three Binomial distributions in the left column, and the corresponding transformed versions in the right column.

When the centerpoint is understood, its effect in the present example is illustrated in the right column of the main graphic. This shows the three theoretical probability histograms of the random variables minus their means. Now it is apparent that mean adjustment overcomes the problem of probability mass moving off to the right, but there is a second problem with the distribution becoming more spread as the sample size grows. Again the effect can be quantitatively understood, and adjusted for, by the development of a concept of spread of a distribution, i.e. the standard deviation.

When the spread is understood, its effect in this example is illustrated by checking the ‘’divide by Stddev” box in the control menu. This changes the right column to plots of the probability histograms of the random variables minus their means, divided by their standard deviations as shown in Figure 3b. This shows that the distribution is clearly converging to a common shape. Then the instructor states that with more mathematics, it can be shown that this common distribution is the Gaussian, i.e. normal distribution, which is then overlaid using the “Normal Distribution” checkbox.

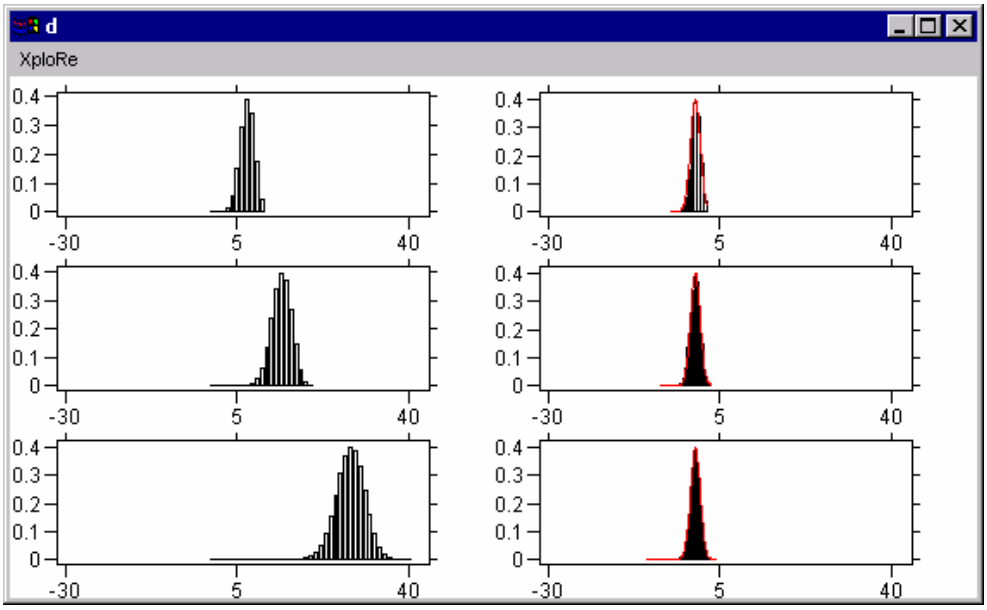


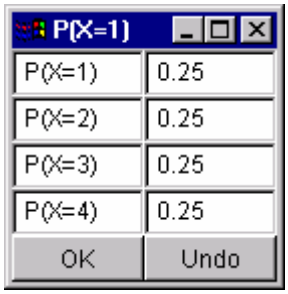
Figure 3b: Shows the effect of adjusting for the scale, on the histograms in the left part of the main output window in Figure 3a. Also shows the effect of overlaying the approximating Normal probability density.

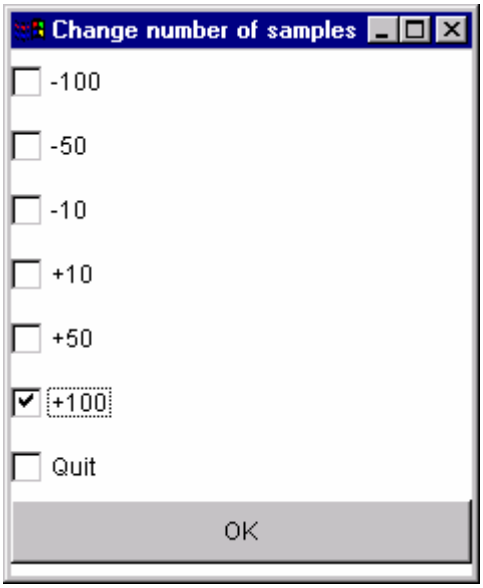
This example may be run from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface, enters **library(“tware”)** and then enters **twnormalize()**.

Example 4

This example shows the main point of the Central Limit Theorem: that averages tend to have a Normal probability distribution, even when the individual underlying distribution is far from Normal. The example starts with a menu containing text boxes (shown in the upper left of Figure 4) for entering an initial discrete probability distribution. This distribution is supported on the integers 1,2,3,4, and after the probabilities are entered a bar graph is displayed (the lower main window shown in Figure 4), showing the probability histogram of the entered distribution. The upper right window controls the number of realizations to average, n.

Clicking OK in the upper right window shows the probability histogram (in the main window) of the average of X1,...,Xn (computed by simple discrete convolution). This demonstrates how the shape tends towards that of the Normal distribution. Another push button will overlay the approximating normal distribution onto the current probability histogram. For level 2 and level 3 teaching, students could be encouraged to try this with other choices of the underlying probability distribution. They could be challenged to find shapes which give rapid convergence to the Normal, and shapes which give very slow convergence. The student has the possibility to increase and decrease the number of the repetitions of the random drawing. This is designed for discovery of “how” and “when” the Normal limit distribution is a valid approximation as a function of sample size.





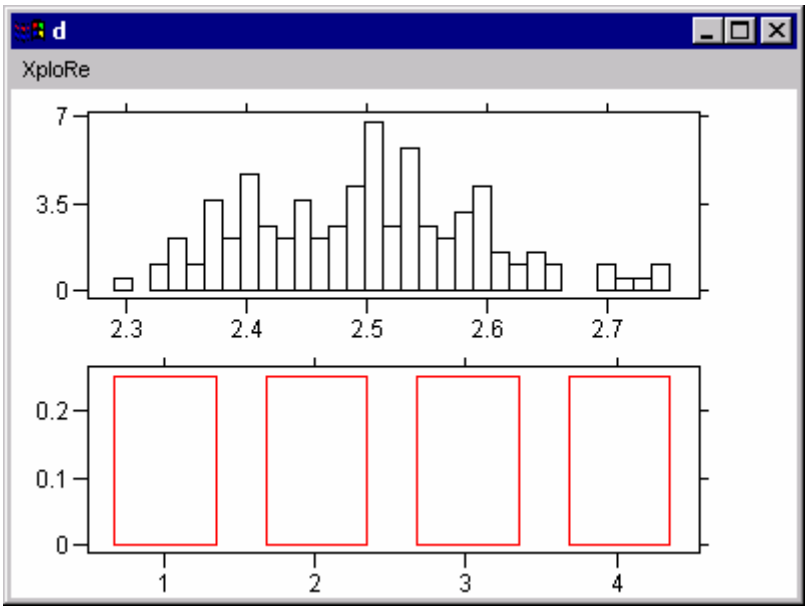


Figure 4: Demonstrates Example 4, Central Limit Theorem. Upper left: menu controlling probabilities of a 4 point distribution. Upper right: menu controlling number of realizations to average. Lower: main graphic window, showing result of repeated convolution, which demonstrates that the distribution of averages converges to the Gaussian.

This example may be run from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface, enters **library("tware")** and then enters **twclt()**.

Example 5

This example illustrates the use of visual display devices for one dimensional data. It shows the relationship between the mean and median, shows how transformation can be used to make data have a distribution closer to Normal, and shows the resulting impact of transformations on the mean and the median. Display devices include histograms, jitter plots, and Q-Q (normal probability) plots. Currently considered transformations are the square root, and the logarithm.

The control menu, shown on the upper left of Figure 5, allows choice of which statstical graphics to include. Check boxes will allow the exploration of various notions of "center" and "spread" via overlaid boxplots. The "mean boxplot" is centered at the mean, with the box endpoints showing the mean plus and minus one standard deviation, and with the whiskers showing the mean plus and minus two standard deviations. The "median box plot" is centered at the median, with the box

endpoints showing the quartiles and the whiskers showing the 2.5 and 97.5 percentiles.

The chosen example has substantial skewness which shows that these two boxplots can be quite different, and furthermore that the percentile methods are giving a better notion of "center" and "spread". The square root and the logarithmic (base 10) transformations, show how this situation changes dramatically when the data are transformed. Closeness to normality, in each case, can also be studied via a Q-Q, i.e. normal probability, plot using that checkbox.

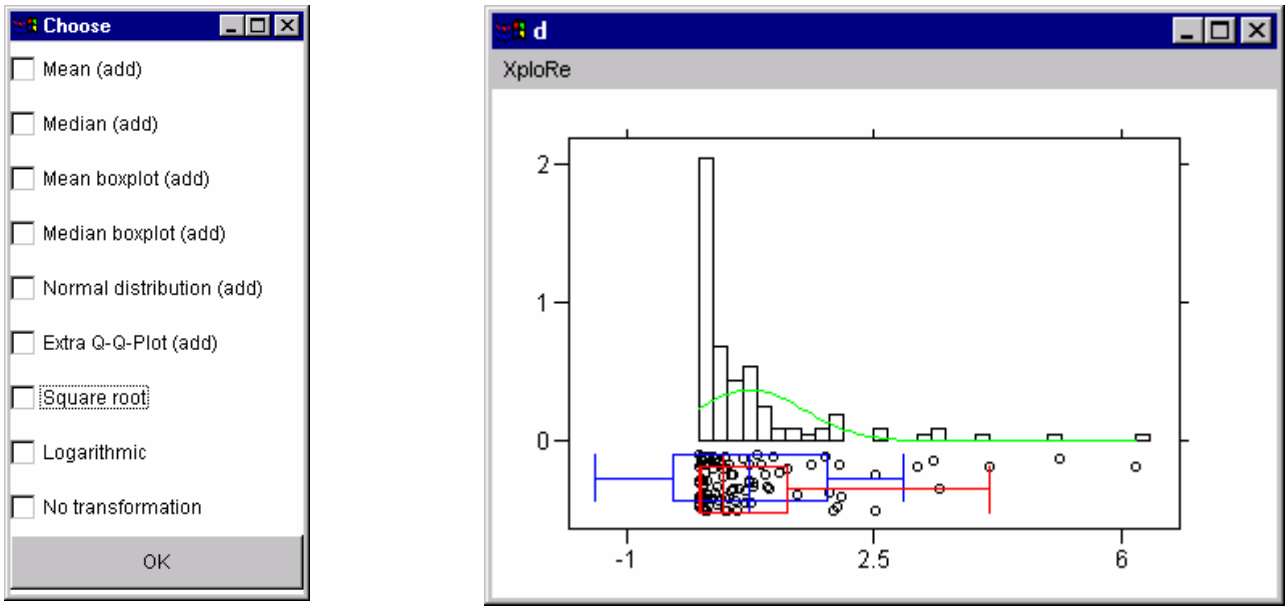


Figure 5: Demonstrates Example 5, Display of 1d data. Left: Control menu, with checkboxes allowing different displays. Right: Main graphics window, currently showing histogram with overlaid Normal distribution, and jitter plot, with both types of boxplot (mean – standard deviation boxplot in blue, median- quartile boxplot in red).

This example may be run from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface, enters **library("tware")** and then enters **tw1d()**.

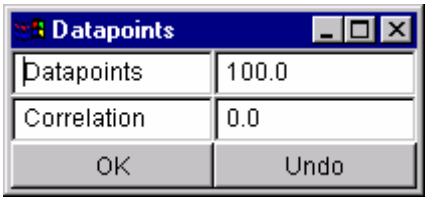
Example 6

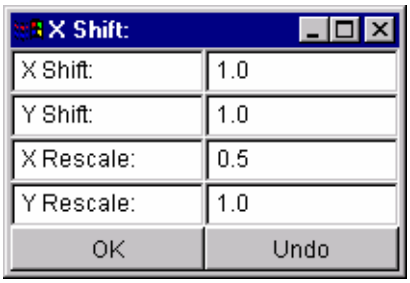
This example gives a visual demonstration of the form of the Pearson correlation coefficient. In particular, it shows why the product moment gives a measure of "dependence", and why it is essential to "normalize", i.e. to subtract means, and divide by standard deviations, to preserve that property.

It uses simulated bivariate Gaussian data, with the number of data points, and the correlation entered through checkboxes as shown in the upper left of Figure 6. The data are shown with a scatterplot in the main graphics window appearing in the bottom of Figure 6. Text below shows the numerical value of the product moment, sum_i (x_i y_i), the recentered product moment, sum_i ((x_i –xbar) (y_i - ybar)), and the rescaled, recentered product moment, sum_i ((x_i –xbar) (y_i - ybar)) / (s_x s_y), which is the Pearson correlation coefficient.

Starting with N(0,1) marginals shows how the ordinary product moment quantifies "dependence", since most values in the first and third quadrants make the product moment positive, and most values in the second and fourth quadrants make the product moment negative, while independence gives cancellation of these effects, so the product moment is essentially zero.

To understand the need for recentering and rescaling, the other menu (shown in the upper right of Figure 6) allows changing the center point of the point cloud. When the centerpoint is changed, the point cloud moves accordingly (with the original position shown in gray) and the various moments also updated. The teacher can comment that the original product moment changes dramatically, while the recentered product moment stays the same. Another menu allows changing the scales, and again this change is apparent both visually, and in the product moments, which shows why normalizing by the product of the standard deviations is essential.





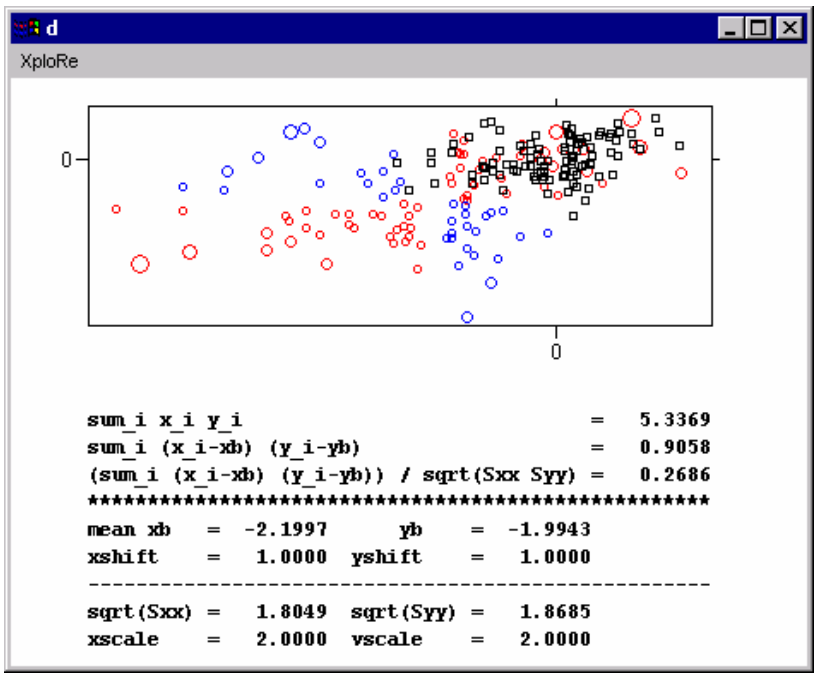


Figure 6: Demonstrates Example 6, Correlation Coefficient. Upper right: menu for controlling number and correlation of underlying normal data. Upper right: menu for demonstrating how shifts and scales affect the product moment, but not the Pearson correlation coefficient.

This example may be run from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface, enters **library("tware")** and then enters **twpearson()**.

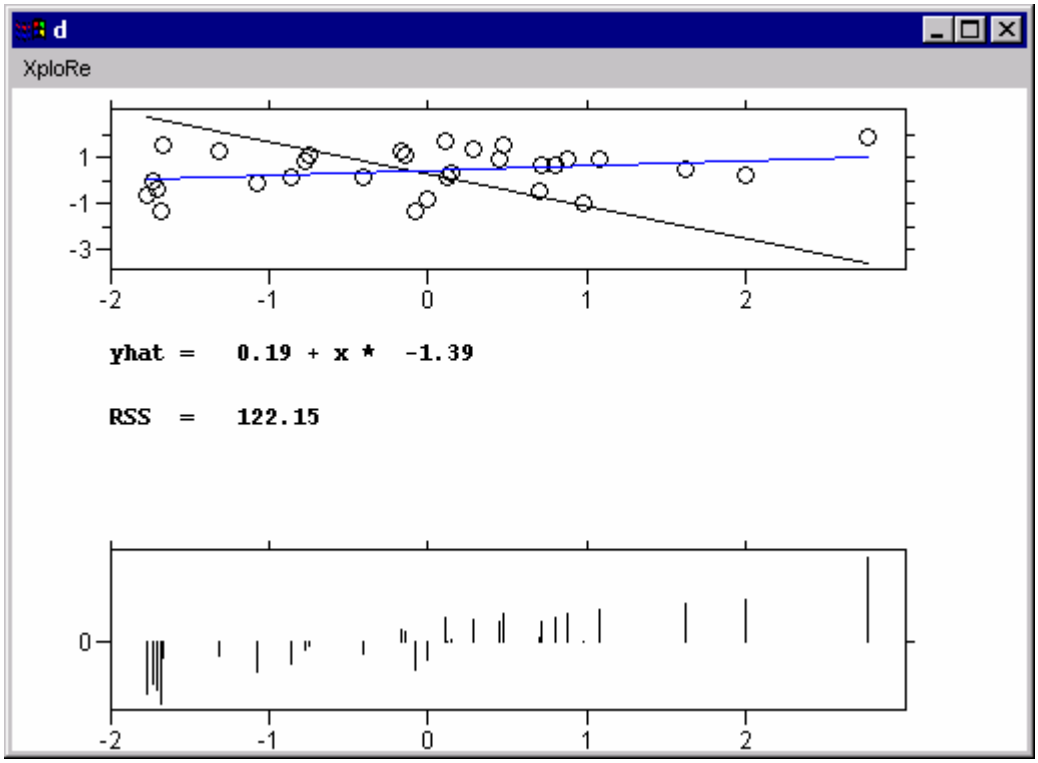
Example 7

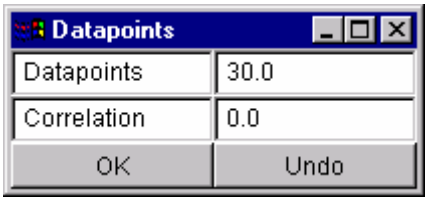
This example gives visual insight into how least squares simple linear regression works, and the relationship between the regression of Y on X, X on Y, and total regression.

As for example 6 the data are bivariate Gaussian, and a menu (shown upper left in Figure 7) allows control of the number of data points, and the correlation. Intuitive understanding of least squares fitting is conveyed through interactive manipulation of a candidate fit line. The upped right menu in Figure 7 gives control over this process, through incremental adjustments that are selected by check boxes, followed by a push of the "OK" button. The main graphics window shows the data scatterplot,

together with the least squares fit line. A text component shows the equation of the current line (which changes as the line is manipulated), together with the Residual Sum of Squares which gives a numerical summary of the goodness of fit. Very effective visual indication of what RSS means comes from the lower graphics part of this window, which represents the residuals as vertical lines. When the fit is poor (and hence the RSS is large), the residual plot shows why, and give a clear indication of how the line should be moved to improve the quality of the fit to the data.

Additional check boxes allow understanding the variations of regression of X on Y, and total variation, and result in appropriate shifts of the graphics. This example could be modified to allow other types of fitting, such as least L1, or other types of robust fits.





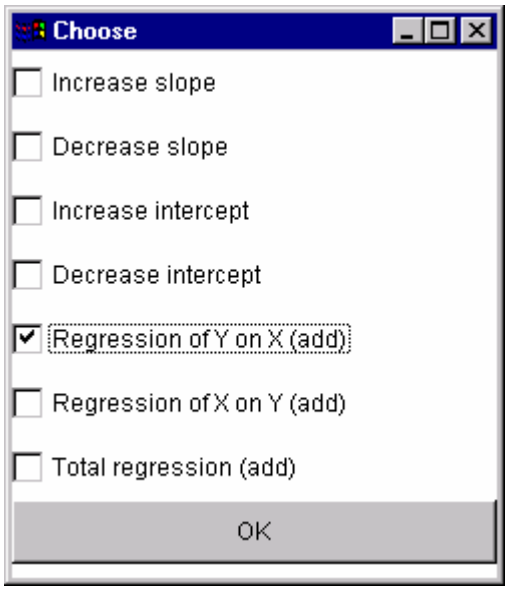


Figure 7: Demonstrates Example 7, Simple Linear Regression. Upper: menu for the changes of the regression line. Lower: main graphic window, showing result of repeated application of changing slope and intercept in comparison with LS line.

This example may be run from the (**http://www.XploRe-stat.de**) directly. One opens the Java 1.1 interface, enters **library("tware")** and then enters **twlinreg()**.

Acknowledgements

We would like to thank Nathan Derby, Marlene Müller and Bernd Rönz for helpful suggestions and corrections. The paper was financially supported by the Sonderforschungsbereich 373 "Quantifikation und Simulation Ökonomischer Prozesse", Deutsche Forschungsgemeinschaft. The research of J. S. Marron was supported in part by NSF grant DMS-9504414.

References

Class room ready examples in XploRe
(**http://ise.wiwi.hu-berlin.de/statistik/lehrmaterial/statmat.html**)

GLM tutorial
(http://www.xplore-stat.de/tutorials/glmstart.html)

GAUSS software
(http://www.aptech.com)

GAUSS programming for Econometricians
(http://eclab.econ.pdx.edu/gpe/)

Help system pages
(http://www.xplore-stat.de/help/_Xpl_Start.html)

Image processing with Java
(http://www.utdallas.edu/~degroat/javadip/JavaDIP.html)

MD*Tech – Method and Data Technologies
(http://www.mdtech.de)

Non- and Semiparametric Modelling course text (passwd protected)
(http://www.quantlet.de/~scripts/scripts/spm/spm.html)

SAS software
(http://www.sas.com/)

SHAZAM software
(http://shazam.econ.ubc.ca)

Splus software
(http://www.mathsoft.com/Splus)

Stata software
(http://www.stata.com)

STATLIB server of SPLUS
(http://lib.stat.cmu.edu/S/)

SticiGui© Java Tools
(http://www.stat.berkeley.edu/users/stark/Java/index.htm)

SUN's Java Development Kit (JDK)
(http://www.javasoft.com/).

Support Vector Machine
(http://svm.dcs.rhbnc.ac.uk/pagesnew/1D-Reg.shtml)

Symanzik (1998)
(http://www.galaxy.gmu.edu/~symanzik)

Virtual Stat Lab
(http://www.ruf.rice.edu/~lane/stat_sim/index.html)

wavelet book in PDF format (password protected)
(http://www.quantlet.de/~scripts/scripts/wav/wavpdf.pdf)

Webstat Project
(http://www.stat.sc.edu/~west/webstat/)

XLISP-STAT
(http://www.cern.ch/WebMaker/examples/xlisp/www/cldoc_1.html)

XploRe – the internet interactive statistical computing environment
(http://www.XploRe-stat.de)