The Stata Journal (2022)
22, Number 3, pp. 496–520 DOI: 10.1177/1536867X221124466

A Stata implementation of second-generation p-values

Sven-Kristjan Bormann
School of Economics and Business Administration
University of Tartu
Tartu, Estonia
sven-kristjan@gmx.de

Abstract. In this article, I introduce new commands to calculate second-generation *p*-values (SGPVs) for common estimation commands in Stata. The `sgpv` command and its companions allow the easy calculation of SGPVs and their associated diagnostics, as well as the plotting of SGPVs against the standard *p*-values.

Keywords: st0681, sgpv, fdrisk, plotsgpv, sgpower, sgpvalue, second-generation p-values

1 Introduction

Second-generation *p*-values (SGPVs) were introduced by Blume et al. (2018, 2019) as an alternative to traditional *p*-values.[1] There is a need for such an alternative because traditional *p*-values have several well-known problems.[2]

One problem is that they are often misinterpreted as the probability of the null hypothesis being true given some observed data $\{P(H_0\,|\,\text{data})\}$, while *p*-values are defined as $P(\text{data}\,|\,H_0)$, the probability of observing the data given that the null hypothesis is true. Typically, data in the context of null-hypothesis testing are the values of some calculated test statistic, for example, a t test. Another well-known problem with classical *p*-values is that they can be small yet also associated with confidence intervals (CIs) that include hypotheses or parameter values that are essentially null.[3]

As stated by Blume et al. (2018, 15), an SGPV is a descriptive summary of the evidence in favor of the null hypothesis given the observed data instead of a probability statement like the traditional *p*-value. Therefore, Blume et al. (2018) argue that they should be easier to understand than "traditional" *p*-values. SGPVs can be used like traditional *p*-values when you keep in mind that an SGPV is a summary statistic of the data. An SGPV is the proportion of null hypotheses supported by the data. The emphasis here is on the plural "hypotheses" instead of the singular "hypothesis". Usually, a point-zero null hypothesis is used when calculating the traditional *p*-value or whatever equals no observed effect, for example, an odds ratio of 1. SGPVs can be calculated for interval and point null hypotheses. An interval null hypothesis can reflect knowledge

1. See https://www.statisticalevidence.com/second-generation-p-values for a quick summary about SGPVs by Jeffrey D. Blume, one of the authors of Blume et al. (2018, 2019).
2. For a list of problems, see, for example, Blume and Peipert (2003).
3. See Blume et al. (2018, 15).

about measurement errors or effect sizes that are different from zero but still do not matter scientifically.

One potential problem of SGPVs is precisely that they require an interval null hypothesis to display their full power. Setting up interval hypotheses is not commonly taught, and they are uncommon in statistical software because *p*-values work only for a point null hypothesis. Yet it is relatively easy to come up with an interval null hypothesis even if you have not used them before in your analysis.

For example, assume that you regressed the log wage on some variables of interest. Now you might find that some coefficient is significantly different at the 5% level from 0 according to its *p*-value, but the coefficient is equal to 0.01. This equals a wage increase of one percentage point. If coefficients of size 0.01 and less are the same as being 0 for your given problem, then you have an interval null hypothesis with an interval going from -0.01 to 0.01. To be more precise, you have null hypotheses because each point within this interval represents an individual point null hypothesis. The SGPV in this case could be larger than 0, indicating that some null hypotheses are compatible with the data. The exact calculation and interpretation of SGPVs will be discussed in the next section.

In this article, I first introduce the formulas required to calculate SGPVs and their associated diagnostic statistics. Guidance is also provided about how to interpret SGPVs. Then I introduce the commands of the `sgpv` package and provide examples of how to use these commands. The focus of this article is to provide only a general overview about what SGPVs are and how they can be calculated in Stata. For a thorough discussion of the formulas and concepts, see the articles by Blume et al. (2018, 2019).

There is already the `sgpv` package[4] for the R programming language[5] to reproduce the results of Blume et al. (2018, 2019). Version 1.0 of the official `sgpv` package was translated from R into Stata, and several features were added that are not part of the original R library to make using SGPVs easier. The current version of the R package as of February 20, 2022, is version 1.1.0, which contains two more plotting commands than the previous version on which this article is based. These new commands have not been ported to Stata for this article.

4. Available from https://github.com/weltybiostat/sgpv (last accessed February 20, 2022).
5. Available from https://www.r-project.org/.

2 SGPVs

2.1 How to calculate SGPVs

The SGPV is defined as[6]

$$p_\delta = \frac{|\,I \cap H_0\,|}{|\,I\,|} \times \max\left(\frac{|\,I\,|}{2\,|\,H_0\,|}, 1\right)$$
$$= \begin{cases} \frac{|\,I \cap H_0\,|}{|\,I\,|} & \text{when } |\,I\,| \leq 2\,|\,H_0\,| \\ \frac{1}{2}\,\frac{|\,I \cap H_0\,|}{|\,H_0\,|} & \text{when } |\,I\,| > 2\,|\,H_0\,| \end{cases}$$

with $I = [\theta_l, \theta_u]$ being the interval estimate of θ and $|\,I\,| = \theta_u - \theta_l$ being the width of the interval. θ_u and θ_l are typically the upper and lower bounds of a $100(1-\alpha)\%$ CI, but any other interval estimate is also possible. H_0 denotes an interval null hypothesis. $|\,I \cap H_0\,|$ denotes the width of the intersection or overlap of the two intervals. $\max(\,|\,I\,|\,/\,2\,|\,H_0\,|\,, 1)$ is a correction term, in which $|\,H_0\,|$ is the width of the null hypothesis. The correction term is needed if the interval estimate is very wide. Then the SGPV is bounded by 1/2.

The SGPV is denoted as p_δ, where δ is the half-width of the interval null hypothesis $(\,|\,H_0\,|\,/\,2)$. The width of the null hypothesis $(2 \times \delta)$ should be driven by scientific context and should be specified prior to conducting the experiment or analyzing the data. While an SGPV can be calculated for a point null hypothesis that is equal to an interval with the width 0, the SGPV needs an interval null hypothesis to display its full power. Therefore, using a point null hypothesis is discouraged.[7]

The null hypothesis should be an interval that contains all effects that are not scientifically relevant. The p-values reported by most of Stata's estimation commands are based on the null hypothesis of a parameter being exactly 0. The SGPV for such a point-zero null hypothesis can take only the values 0.5 or 0 because of the correction term. A point is either contained within the interval estimate or not. You could set a small null-hypothesis interval that includes effects of less than 1% or 0.1%, as in the example in section 1. The exact numbers depend on what you deem a priori as not scientifically relevant.

6. See (1) in Blume et al. (2019).
7. See answer 11 in the supplementary file 2 in Blume et al. (2018).

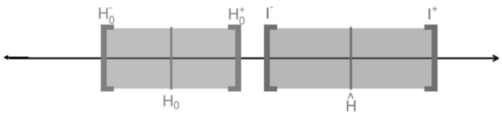
2.2 How to interpret SGPVs

Compared with traditional p-values, SGPVs have a fixed interpretation.[8] If you have a general interval null hypothesis like $H_0\colon \mu_{0a} \leq \mu \leq \mu_{0b}$, then

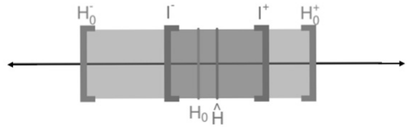
1. p_δ always lies between 0 and 1 (inclusive),
2. a p_δ of 0 indicates that 0% of the null hypotheses are compatible with the data,
3. a p_δ of 1 indicates that 100% of the null hypotheses are compatible with the data,
4. a p_δ between 0 and 1 indicates inconclusive evidence, and
5. a p_δ of 1/2 indicates strictly inconclusive evidence. The degree of inconclusiveness is represented by p_δ itself.

To illustrate the different regimes of p_δ, I show them in figure 1. This figure is similar to figure 1 in Blume et al. (2018).[9] The widths of the interval estimate and the interval null hypotheses are chosen purely for illustrative purposes. You can construct an interval estimate and an interval null hypothesis that have a traditional p-value of 0.05 but an SGPV of 0.5, meaning that a result could be considered significantly different from the null hypothesis at the 5% level using the traditional p-value but providing inconclusive evidence under the SGPV. More generally, you can have results that are significant in the traditional sense at the 1% level but still inconclusive with a p_δ between 0 and 1.

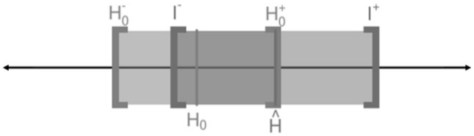
8. See section 2.3 in Blume et al. (2018) for more information.
9. The subfigures were created with the help of the website https://lucy.shinyapps.io/sgpvalue/, which illustrates the differences between SGPVs and classical p-values.



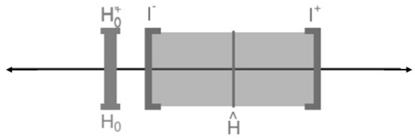
(a) $p_\delta = 0$



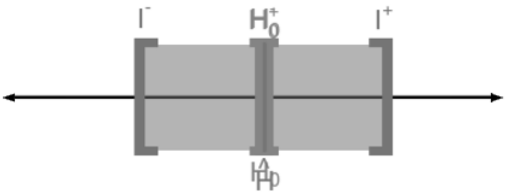
(b) $p_\delta = 1$



(c) $p_\delta = 0.5$



(d) Point null hypothesis outside of CI



(e) Point null hypothesis within CI

Figure 1. Illustration of interval and point null hypothesis, H_0; the estimated effect that is the best-supported hypothesis, $\widehat{H} = \widehat{\theta}$; the 95% CI for the estimated effect (I^-, I^+); and the interval null hypothesis (H_0^-, H_0^+)

Figure 1(a) shows that there is no overlap between the estimated interval and the interval null hypothesis when $p_\delta = 0$. Therefore, 0% of the null hypotheses are supported by the data. The opposite case is shown in figure 1(b), where the interval estimate is completely contained within the interval null hypothesis. Figure 1(c) showcases inconclusive evidence, where 50% of the interval null hypothesis is contained within the interval estimate. Figure 1(d) illustrates the usual case of a point null hypothesis that is not contained within an interval estimate. Because the point is not contained in the interval, p_δ is equal to 0. The last case is the opposite case of the previous one. Now a point null hypothesis is contained within the interval estimate, which leads to an SGPV of 0.5 as shown in figure 1(e). The SGPV is 0.5 because of the correction term, as discussed in the previous section.

For more information about how to interpret the SGPVs and other common questions, see the frequently asked questions by Blume et al. (2018).[10]

10. Available from https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0188299.s002&type=supplementary.

2.3 The delta gap

The delta gap is a way of ranking two studies that both have SGPVs of 0 (p_δ= 0). It is defined as the distance between the intervals in δ units, with δ being the half-width of the interval null hypothesis. The formula presented here for the delta gap is based on the actual code for the calculation of the delta gap.

The delta gap is calculated as[11]

$$\begin{aligned}\text{gap} &= \max(\theta_l, H_{0l}) - \min(H_{0u}, \theta_u)\\ \text{delta} &= \mid H_0 \mid /2\\ \text{delta.gap} &= \text{gap/delta}\end{aligned}$$

For the standard case of a point-zero null hypothesis and a 95% CI, the delta is set to be equal to 1. Then the delta gap is just the distance between either the upper or the lower bound of the CI and 0. If both θ_u and θ_l are negative, then the delta gap is just θ_u, the upper bound of the CI. If both bounds of the CI are positive, then the delta gap is equal to the lower bound of the CI.

2.4 False discovery or confirmation risk

The relevant uncertainty measures that the calculated SGPV, say, $p_\delta = 0$ or 1, is mistaken are given by the false discovery risk (FDR) and the false confirmation risk (FCR).

The FDR is defined as

$$P(H_0 \mid p_\delta = 0) = \left(1 + \frac{P(p_\delta = 0 \mid H_1)}{P(p_\delta = 0 \mid H_0)} r\right)^{-1}$$

The FCR is defined as

$$P(H_1 \mid p_\delta = 1) = \left(1 + \frac{P(p_\delta = 1 \mid H_0)}{P(p_\delta = 1 \mid H_1)} \frac{1}{r}\right)^{-1}$$

with $r = P(H_1)/P(H_0)$ being the ratio of the prior probabilities for the null and the alternative hypotheses.[12] This ratio will be later referred to as `pi0()` in the `fdrisk` command. This ratio can be rewritten as $r = 1 - P(H_0) \,/\, P(H_0)$ with $1 - P(H_0) = P(H_1)$. This reformulation clarifies the connection between the prior probabilities $P(H_1)$ and $P(H_0)$. The hypothesis notation, H_i for $i = 0, 1$, can represent a point or an interval hypothesis.[13]

11. See section 2.4 in Blume et al. (2018) for a less formal description of the delta gap.
12. See (3) in Blume et al. (2019).
13. See section 5, "Frequency properties", in Blume et al. (2019) and section 4.2, "Reliability of an observed second-generation p-value", in Blume et al. (2018) for further discussion.

2.5 Power functions

Last but not least, the power functions are shown based on (S4), (S6), (S8), and (S9) from the supplementary material in Blume et al. (2018). With regard to the formulas in the supplementary material, I replaced $Z_{\alpha/2}$ with $Z_{1-\alpha/2}$, which is more in line with the actual code. The power functions are needed, among other things, to calculate the FDR and FCR.

The probability of observing data compatible with the alternative hypothesis for a normally distributed test statistic θ is given by

$$P_\theta(p_\delta = 0) = \Phi\left(\frac{(\theta_0 - \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} + Z_{1-\alpha/2}\right) + \Phi\left(-\frac{(\theta_0 + \delta)}{\text{SE}} + \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right)$$

The probability of observing data compatible with the null hypothesis is given by

$$P_\theta(p_\delta = 1) = \Phi\left(\frac{(\theta_0 + \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right) - \Phi\left(\frac{(\theta_0 - \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} + Z_{1-\alpha/2}\right)$$

The probability of observing data that are inconclusive is given by

$$\begin{aligned} P_\theta(0 < p_\delta < 1) = 1 &- \Phi\left(\frac{(\theta_0 - \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right) + \Phi\left(-\frac{(\theta_0 + \delta)}{\text{SE}} + \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right) \\ &- \Phi\left(\frac{(\theta_0 + \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right) - \Phi\left(\frac{(\theta_0 - \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} + Z_{1-\alpha/2}\right) \\ &\text{when } \delta > Z_{1-\alpha/2} \times \text{SE} \\ P_\theta(0 < p_\delta < 1) = 1 &- \Phi\left(\frac{(\theta_0 - \delta)}{\text{SE}} - \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right) + \Phi\left(-\frac{(\theta_0 + \delta)}{\text{SE}} + \frac{\theta}{\text{SE}} - Z_{1-\alpha/2}\right) \\ &\text{when } \delta \le Z_{1-\alpha/2} \times \text{SE} \end{aligned}$$

where SE denotes the standard error, $(\theta_0 - \delta)$ and $(\theta_0 + \delta)$ denote, respectively, the lower and upper bounds of the null interval, $\Phi(\ldots)$ is the cumulative distribution function of the standard normal distribution, and $Z_{1-\alpha/2}$ is the value for the inverse cumulative distribution function of the standard normal distribution at the point $1 - \alpha/2$ with α being the interval or significance level, which is typically 5%.

3 The sgpv package

The `sgpv` package consists of the following commands:

- `sgpv`: the main program to calculate SGPVs; to be used after estimation commands
- `sgpvalue`: calculate the SGPVs
- `sgpower`: power functions for the SGPVs
- `fdrisk`: false discovery or confirmation risks for the SGPVs
- `plotsgpv`: plot interval estimates according to SGPV rankings

I briefly discuss the most important options for each command. The commands try to emulate the possibilities of the original R code by Valerie F. Welty and Jeffrey D. Blume. Because of the (internal) differences between R and Stata, this is not 100% possible. Furthermore, some options have been removed or renamed and work slightly differently from the R version to be more in line with Stata standards. In addition to the already existing commands, `sgpv` was written to simplify calculating SGPVs after common estimation commands.

You can call the other commands of the `sgpv` package with the `sgpv` command. This is mostly a convenience feature, so the help files for the individual commands should be consulted for the options of these commands. Supported subcommands are `value` to call `sgpvalue`, `power` to call `sgpower`, `risk` to call `fdrisk`, `plot` to call `plotsgpv`, and `menu`, the only "true" subcommand belonging to `sgpv`. Using the individual commands directly is comparable with the immediate form of other Stata commands, like `ttesti`. This can be useful in different scenarios, for example, when you want to get the SGPV (and its diagnostics) for a few existing estimation results, similar to how you would use the other immediate commands in Stata.

3.1 The sgpv command

The syntax for the `sgpv` command is

sgpv [*subcommand*] [, estimate(*name*) matrix(*name*) coefficient(*coeflist*)
noconstant nulllo(*boundlist*) nullhi(*boundlist*) level(#) quietly
nonullwarnings matlistopt(*options*) deltagap fdrisk all format(%*fmt*)
truncnormal likelihood(#) pi0(#) permdialog remove]
[: *estimation_command*]

The default behavior is to calculate the SGPVs for the last estimation command unless the options `estimate()` and `matrix()` are used or `sgpv` is used to prefix an estimation command. `sgpv` also allows the installation of the dialog boxes for each command into the **User** menu bar.

3.1.1 Options for sgpv

Main

`replay` is the default behavior if no estimation command, matrix, or stored estimate is set. The `replay` option is available only in the dialog box. The `sgpv` command behaves like any other estimation command (for example, `regress`) that replays the previous results when run without a list of variables. The results from previous runs of `sgpv` are not used to display the results. Instead, the results are freshly calculated on every run of `sgpv`. You can display the results from a previous run of `sgpv` without recalculation with the command `matlist r(comparison)` if no other commands were run after `sgpv`. Only one of the following can be used to calculate the SGPVs: an estimation command, the results from the previous estimation command, a stored estimation result, or a matrix with the necessary information.

`estimate(`*name*`)` takes the name of a previously stored estimation.

`matrix(`*name*`)` takes the name of a matrix as input for the calculation. The matrix must follow the structure of the `r(table)` matrix returned after commonly used estimation commands because of the hardcoded row numbers used for identifying the necessary numbers. This means that the parameter estimate has to be in the first row, the standard errors in the second row, the test statistics in the third row, the *p*-values in the fourth row, the lower bound in the fifth row, and the upper bound in the sixth row. As an additional check, the row names of the supplied matrix need to match the row names of the `r(table)` matrix. The row names are `b`, `se`, `t`, `pvalue`, `ll`, and `ul`. To set the row names, run `matrix rownames` *your_matrix* `= b se t pvalue ll ul`. Example code is located in the ancillary file `sgpv-leukemia-example.do`.

`coefficient(`*coeflist*`)` allows the selection of the coefficients for which the SGPVs and other statistics are calculated. The selected coefficients need to have the same names as displayed in the estimation output. If you did not use factor-variable notation, then the names are identical to the variable names. Otherwise, you have to use factor-variable notation, for example, `1.foreign` if you estimated `reg price mpg i.foreign`. Multiple coefficients must be separated with a space. You can also select only an equation by using `eq:` or select a specific equation and variable `eq:var`. See "Setting multiple null hypotheses" in the next section for an example.

A *coeflist* is
coef [*coef* ...]
`eq:`*coef* [`eq:`*coef* ...]
`eq:` [`eq:` ...]

`noconstant` does not calculate SGPVs, delta gaps, and FDRs for the constant term. The constant term is also removed from the list of coefficients if the `coefficient()` option is used and only equations are specified.

`nulllo(`*boundlist*`)` changes the lower limit of the null-hypothesis interval. The default is `nulllo(0)` (the same limit as for the usually reported *p*-values). Missing values, strings, and variable names are not allowed. Expressions or formulas[14] are also allowed as input. More than one null hypothesis is also supported. Each lower bound must be separated with a space. The number of lower bounds must match the number of arguments set in the `coefficient()` option. The number of lower and upper bounds must also match.

`nullhi(`*boundlist*`)` changes the upper limit of the null-hypothesis interval. The default is `nullhi(0)` (the same limit as for the usually reported *p*-values). Missing values, strings, and variable names are not allowed. Expressions or formulas are also allowed as input. More than one null hypothesis is also supported. Each upper bound must be separated with a space. The number of upper bounds must match the number of arguments set in the `coefficient()` option. The number of lower and upper bounds must also match.

The default value, 0, is just meant to be used for an easier beginning when starting to use SGPVs. Please change this value to something more reasonable. The definition of a reasonable upper bound with no scientifically interesting effects depends on your dataset and your research question. Using this default value will always result in having SGPVs of value 0 or 0.5!

A *boundlist* is

[*#* ...]

Reporting

`level(`*#*`)` sets the level of the CI. The default is `level(95)` or as set by `set level`. See [R] **Estimation options**. This option overrides the same-named option of an estimation command. A warning is displayed in the beginning if this happens.

`quietly` suppresses the output of the estimation command.

`nonullwarnings` disables warning messages when the default point-zero null hypothesis is used for calculating the SGPVs. You should disable these warning messages only if using the default point-zero null hypothesis is what you want to do and you understand the consequences of doing so. The warning messages are displayed after the displayed results and are meant to act as a reminder that Blume et al. (2018) are very much in favor of using an interval null hypothesis over a point null hypothesis.

`matlistopt(`*options*`)` changes the options of the displayed matrix. The same options as for Stata's `matlist` command[15] can be used.

`deltagap` calculates and displays the delta gap if the SGPV is 0.

`fdrisk` calculates and displays the FDR if the SGPV is 0.

14. See [U] **13 Functions and expressions**.
15. See [P] **matlist**.

`all` calculates and displays both the delta gap and the FDR if the SGPV is 0. This option takes precedence over the options `deltagap` and `fdrisk`. Using both options together has the same effect as using this option alone.

`format(%`*fmt*`)`[16] specifies the format for displaying the individual elements of the result matrix. The default is `format(%5.4f)`. This option is not identical to the same-named option of the `matlist` command but works independently of it. Setting the format option via `matlistopt()` overrides the setting here and also changes the format of the column names.

`truncnormal` uses the truncated normal distribution as the probability distribution for the null and alternative parameter spaces. The default is to use the uniform distribution as the probability distribution for the null and alternative parameter spaces. The mean and standard deviation of the distribution are automatically set based on the estimated coefficient.

`likelihood(#)` is the likelihood support interval. The likelihood support level with level $1/k$ was used to calculate the SGPV. The level is $1/k$ (not k). For technical reasons, a fraction like 1/8 as # must be entered as a real number like 0.125 first when used for this option.

`pi0(#)` is the prior probability of the null hypothesis. The default is `pi0(0.5)`. This value can be between only 0 and 1 (exclusive). A prior probability outside this interval is not sensible. The default value assumes that both hypotheses are equally likely.

Menu

`permdialog` permanently installs the dialog boxes into the **User** menu bar. The necessary commands are added to the user's `profile.do`. If no `profile.do` exists or can be found, then a new `profile.do` is created in the current directory. A `profile.do` will be found only in the current directory and in the Stata installation base folder. Other possible places, like the user's home folder, are not accessed yet. Without this option, the dialog boxes will be available only from the menu bar until the next restart of Stata. The dialog boxes can be accessed as usual by, for example, `db sgpv`.

`remove` removes the entries created by the option `permdialog` from the `profile.do` file. A backup of the original file is kept with the name `profile.do.bak`. This option has been tested only under Windows; other operating systems should work but could not be tested. The `profile.do` file should not contain a line like "`global F4`" or "`global F5`". If it does, the option returns an error and will not delete the menu entries. Also, make sure that `profile.do` is not open in another program and that the file `profile.do.bak` or `profile.do.new` does not already exist. If they exist, you can delete them.

16. See [D] **format** for more information about how to define a format.

As an alternative to this subcommand, open the `profile.do` file by running the following two commands, and manually delete the lines added by the `permdialog` option:

```
. findfile profile.do, path(STATA;.)
. doedit ```r(fn)'''
```

3.1.2 Stored results for sgpv

`sgpv` stores the following in `r()`:

Scalar

`r(level)`	CI

Matrices

`r(comparison)`	a matrix containing the displayed results
`r(table)`	coefficient statistics

3.2 The sgpvalue command

The syntax for the `sgpvalue` command is

`sgpvalue, estlo(`*boundlist*`) esthi(`*boundlist*`) nulllo(`*boundlist*`)`
`nullhi(`*boundlist*`)` [`nowarn`ings `nodeltagap noshow nomata replace`
`inf`correction(#)]

The `sgpvalue` command is the main workhorse of this package. While the calculations of the SGPVs are simple, this command allows for many ways to provide lower and upper bounds of the estimated and null intervals.

3.2.1 Options for sgpvalue

Main

`estlo(`*boundlist*`)` specifies the lower bound of the interval estimate. Values may be finite or infinite. To specify that the lower limit is minus infinity, just specify the missing value "`.`" in this option. Multiple lower bounds can be entered. The number of lower bounds must match the number of upper bounds specified in option `esthi()`. They must be separated by spaces. Expressions or formulas are also allowed as input. Typically, the lower bound of a CI can be used. A variable or matrix containing the lower bound can also be used, but then a variable or matrix containing the upper bound must also be used for option `esthi()`. `estlo()` is required.

`esthi(`*boundlist*`)` specifies the upper bound of the interval estimate. Values may be finite or infinite. To specify that the upper limit is plus infinity, just specify the missing value "`.`" in this option. The number of upper bounds must match the number of lower bounds specified in option `estlo()`. Multiple upper bounds can be entered. They must be separated by spaces. Expressions or formulas are also allowed as input. Typically, the upper bound of a CI is used. A variable or matrix containing the upper bound can also be used, but then a variable or matrix containing the lower bound must also be used for option `estlo()`. `esthi()` is required.

`nulllo(`*boundlist*`)` specifies the lower bound of the null interval. Values may be finite or infinite. To specify that the lower limit is minus infinity, just specify the missing value "`.`" in this option. Multiple lower bounds can be entered. In this case, the number of lower bounds must match the number of lower bounds specified in option `estlo()`. The number of lower bounds must always match the number of upper bounds in option `nullhi()`. They must be separated by spaces. Expressions or formulas are also allowed as input. `nulllo()` is required.

`nullhi(`*boundlist*`)` specifies the upper bound of the null interval. Values may be finite or infinite. To specify that the upper limit is plus infinity, just specify the missing value "`.`" in this option. In this case, the number of upper bounds must match the number of upper bounds specified in option `esthi()`. The number of upper bounds must always match the number of lower bounds in option `nulllo()`. Multiple upper bounds can be entered. They must be separated by spaces. Expressions or formulas[17] are also allowed as input. `nullhi()` is required.

A *boundlist* is
[# ...]
exp [*exp* ...]

Further options

`nowarnings` disables showing the warnings about potentially problematic intervals.

`nodeltagap` disables the display of the delta gap. The `nodeltagap` option is mainly used inside of `sgpv` because delta gaps are less useful to most users of *p*-values.

`noshow` specifies not to show the outcome of the calculations. This option is useful for larger calculations.

`nomata` deactivates the usage of Mata for calculating the SGPVs with large matrices or variables. If this option is set, a variable-based approach is used. Using variables instead of Mata will be faster, but new variables containing the results are created. If you do not want to create new variables and time is not an issue, then do not set this option. Stata might become unresponsive when using Mata because it takes time to return a large matrix.

`replace` replaces existing variables when the `nomata` option was used.

`infcorrection(#)` specifies a small number to denote a positive but infinitesimally small SGPV. The default is `infcorrection(1e-5)`. SGPVs that are infinitesimally close to 1 are assigned 1 − `infcorrection()`. This option can be only invoked when one of the intervals has infinite width.

17. See [U] **13 Functions and expressions**.

3.2.2 Stored results for sgpvalue

`sgpvalue` stores the following in `r()`:

Matrices
`r(results)` matrix with the results

3.3 The sgpower command

The syntax for the `sgpower` command is

`sgpower, true(#) nulllo(#) nullhi(#)` [`stderr(#) level(#)`
`likelihood(#) bonus`]

3.3.1 Options for sgpower

Main

`true(#)` specifies the true value for the parameter of interest at which to calculate power. Note that this is on the absolute scale of the parameter and not the standard deviation or standard error scale. `true()` is required.

`nulllo(#)` specifies the lower bound of the indifference zone (null interval) upon which the SGPV is based. `nulllo()` is required.

`nullhi(#)` specifies the upper bound for the indifference zone (null interval) upon which the SGPV is based. `nullhi()` is required.

`stderr(#)` specifies the standard error for the distribution of the estimator for the parameter of interest. Note that this is the standard deviation for the estimator, not the standard deviation parameter for the data themselves. This will be a function of the sample sizes.

`level(#)` uses a CI with level $(1-\alpha)100\%$. The default is `level(95)` or as set by `set level` if option `likelihood()` or another CI is not set.

`likelihood(#)` uses a likelihood support interval with level $1/k$. The level is $1/k$ (not k). For technical reasons, a fraction like 1/8 as # must be entered as a real number like 0.125 first when used for this option.

`bonus` displays the additional diagnostics for the type I error.

3.3.2 Stored results for sgpower

`sgpower` stores the following in `r()`:

Scalars

`r(poweralt)`	probability of SGPV = 0 calculated assuming the parameter is equal to `true()`; that is, `r(poweralt)` $= P(\text{SGPV} = 0 \,\vert\, \theta =$ `true()`$)$
`r(powernull)`	probability of SGPV = 1 calculated assuming the parameter is equal to `true()`; that is, `r(powernull)` $= P(\text{SGPV} = 1 \,\vert\, \theta =$ `true()`$)$
`r(powerinc)`	probability of $0 <$ SGPV < 1 calculated assuming the parameter is equal to `true()`; that is, `r(powerinc)` $= P(0 < \text{SGPV} < 1 \,\vert\, \theta =$ `true()`$)$

The next three scalars are returned only if the `bonus` option was used.

Scalars

`r(minI)`	minimum type I error over the range (`nulllo()`, `nullhi()`), which occurs at the midpoint of (`nulllo()`, `nullhi()`)
`r(maxI)`	maximum type I error over the range (`nulllo()`, `nullhi()`), which occurs at the boundaries of the null hypothesis, `nulllo()` and `nullhi()`
`r(avgI)`	average type I error (unweighted) over the range (`nulllo()`, `nullhi()`)

If 0 is included in the null-hypothesis region, then "type I error summaries" also contains at 0:

Scalars

`r(pow0)`	type I error calculated assuming true parameter value θ is equal to 0

3.4 The fdrisk command

The syntax for the `fdrisk` command is

```
fdrisk, nulllo(#) nullhi(#) stderr(#) nullspace(# [#])
   altspace(# [#]) [fcr nulltruncnormal alttruncnormal level(#)
   likelihood(#) pi0(#)]
```

The default is to calculate the FDR when the observed SGPV is 0.

3.4.1 Options for fdrisk

Main

`nulllo(#)` specifies the lower bound of the indifference zone (null interval) upon which the SGPV was based. `nulllo()` is required.

`nullhi(#)` specifies the upper bound of the indifference zone (null interval) upon which the SGPV was based. `nullhi()` is required.

`stderr(#)` specifies the standard error of the point estimate. `stderr()` is required.

`nullspace(#` [`#`]`)` specifies the support of the null probability distribution. The `nullspace()` option can contain either one number or two numbers. These numbers can also be formulas that must be enclosed in `" "`. If `nullspace()` contains one number, then no distribution for the null parameter space is used. If `nullspace()` contains two numbers separated by a space, then the distribution for the alternative parameter space is either the uniform distribution or the truncated normal distribution (option `nulltruncnormal`). The uniform distribution is used as the default. `nullspace()` is required.

`altspace(#` [`#`]`)` specifies the support for the alternative probability distribution. The `altspace()` option can contain either one number or two numbers. These numbers can also be formulas that must be enclosed in `" "`. If `altspace()` contains one number, then no distribution for the null parameter space is used. If `altspace()` contains two numbers separated by a space, then the distribution for the alternative parameter space is either the uniform distribution or the truncated normal distribution (option `alttruncnormal`). The uniform distribution is used as the default. `altspace()` is required.

`fcr` calculates the FCR when the observed SGPV is 1 instead of the default FDR.

`nulltruncnormal` uses the truncated normal distribution as the probability distribution for the alternative parameter space with the mean being the middle point of the `nullspace()` and standard deviation given by option `stderr()`.

`alttruncnormal` uses the truncated normal distribution as the probability distribution for the alternative parameter space with the mean being the middle point of the `altspace()` and the standard deviation given by option `stderr()`.

`level(#)` specifies the CI. The CI with level $(1 - \alpha)100\%$ was used to calculate the SGPV. The default is `level(95)` if option `likelihood()` or another CI is not set.

`likelihood(#)` uses a likelihood support interval with level $1/k$. The level is $1/k$ (not k). For technical reasons, a fraction like 1/8 as # must be entered as a real number like 0.125 first when used for this option.

`pi0(#)` specifies the prior probability of the null hypothesis. The default is `pi0(0.5)`. This value can be between only 0 and 1 (exclusive). A prior probability outside this interval is not sensible. The default value assumes that both hypotheses are equally likely.

3.4.2 Stored results for fdrisk

`fdrisk` stores the following in `r()`:

Scalars

`r(fdr)`	FDR
`r(fcr)`	FCR

3.5 The plotsgpv command

The syntax for the `plotsgpv` command is

`plotsgpv` [*if*] [*in*] `, estlo(`*name*`) esthi(`*name*`) nulllo(`*boundlist*`)`
`nullhi(`*boundlist*`)` [`nullcol(`*color*`) intcol1(`*color*`) intcol2(`*color*`)`
`intcol3(`*color*`) title(`*string*`) xtitle(`*string*`) ytitle(`*string*`) noploty_axis`
`noplotx_axis nooutlinezone nolegend nomata replace setorder(`*string*`)`
`xshow(`*#*`) nullpt(`*#*`) twoway_opt(`*gropt*`)`]

3.5.1 Options for plotsgpv

Main

`estlo(`*name*`)` specifies the lower bound of the interval estimate. Typically, the lower bound of a CI can be used. A variable or matrix containing the lower bound can also be used, but then a variable or matrix containing the upper bound must also be used for option `esthi()`. `estlo()` is required.

`esthi(`*name*`)` specifies the upper bound of the interval estimate. Typically, the upper bound of a CI can be used. A variable or matrix containing the upper bound can also be used, but then a variable or matrix containing the lower bound must also be used for option `estlo()`. `esthi()` is required.

`nulllo(`*boundlist*`)` specifies the lower bound of the null interval. `nulllo()` is required.

`nullhi(`*boundlist*`)` specifies the upper bound of the null interval. `nullhi()` is required.

A *boundlist* is
[*#* ...]
exp [*exp* ...]

Color options

`nullcol(`*color*`)` sets the coloring of the null interval (indifference zone). The default is `nullcol(r 208 216 232)` for the R color Hawkes Blue. You can see the color before plotting via `palette color 208 216 232`. You can set the color to any other available color in Stata. See [G-4] ***colorstyle*** for more information.

The coloring of the intervals is according to SGPV ranking. The defaults are the R colors ("firebrick3", "cornflowerblue", and "darkslateblue") for SGPVs of 0, in (0, 1), and 1, respectively. You can see the default colors before plotting via the following commands:

`palette color 205 38 38`	firebrick3	for SGPV $= 0$
`palette color 100 149 237`	cornflowerblue	for $0 <$ SGPV < 1
`palette color 72 61 139`	darkslateblue	for SGPV $= 1$

`intcol1(`*color*`)` sets the color of the interval with the R color firebrick3 as the default for SGPV = 0.

`intcol2(`*color*`)` sets the color of the interval with the R color cornflowerblue as the default for $0 <$ SGPV < 1.

`intcol3(`*color*`)` sets the color of the interval with the R color darkslateblue as the default for SGPV = 1.

Title options

`title(`*string*`)` specifies the title of the plot.

`xtitle(`*string*`)` specifies the label of the x axis. The default is `xtitle("Ranking according to` *order*`")`, where *order* can refer to the value of the option `setorder()` or the original sorting of the input.

`ytitle(`*string*`)` specifies the label of the y axis.

Further options

`noploty_axis` deactivates showing the y axis.

`noplotx_axis` deactivates showing the x axis.

`nooutlinezone` deactivates drawing a slim white outline around the null zone. This is a helpful visual aid when plotting many intervals. The default is on.

`nolegend` deactivates plotting the legend.

`nomata` deactivates the usage of Mata for calculating the SGPVs with large matrices or variables. If this option is set, a variable-based approach is used. Using variables instead of Mata will be faster, but new variables containing the results are created. If you do not want to create new variables and time is not an issue, then do not set this option. Stata might become unresponsive when using Mata because it takes time to return a large matrix.

`replace` replaces existing variables when the `nomata` option was used.

`setorder(`*string*`)` specifies a variable giving the desired order along the x axis. If `setorder()` is set to `sgpv`, the SGPV ranking is used. If `setorder()` is empty, the original input ordering is used.

`xshow(`*#*`)` specifies a number representing the maximum ranking on the x axis that is displayed. The default is to display all intervals.

`nullpt(`*#*`)` specifies a number representing a point null hypothesis. If set, the command will draw a horizontal dashed red line at this location.

`twoway_opt(`*gropt*`)` specifies any additional options for the plotting. See [G-3] **twoway_options** for more information about the possible options. Options set here may override the values set in other options before.

4 Examples

A few examples show how to use each individual command. There are more examples available in the help file for each command.

The standard `auto.dta` is used for all calculations in the examples of the `sgpv` command. The values of the null hypotheses in these examples are chosen for illustrative purposes and have no deeper meaning.

4.1 sgpv command as a prefix command

The first example shows how to use the `sgpv` command as a prefix command with all bonus statistics calculated. By default, only the SGPVs are calculated. First, the usual regression output is shown; second, the output of the `sgpv` command is shown. The title of the output indicates which null hypothesis has been used to calculate the SGPVs. A point-zero null hypothesis is the default used by the `sgpv` command as in many Stata commands.

The SGPVs are displayed to the right of the p-values from the estimation command to allow an easier comparison of these two statistics. The distance between the lower bound of the estimated 95% CI and 0 is given in the column `Delta Gap`. The last column contains the FDR. For a point null hypothesis, this risk is close to the chosen α value. The standard value for α is 5%. Below the table is a warning to inform the user that using the default point-zero null hypothesis results in having SGPVs of either 0 or 0.5. This example is meant only to illustrate how the output of the `sgpv` command looks. In general, an interval null hypothesis is needed to display the true power of SGPVs.

To better illustrate the power of SGPVs, I would have liked to show here an example where the conclusions from a traditional p-value and the SGPVs are shown in the opposite direction. However, constructing such an example with the example datasets of Stata is not possible unless the interval null hypothesis is specified to get the desired results. Specifying a somewhat realistic interval hypothesis, which means something like a 1% deviation from 0 as being equal to 0, leads in most cases to the same conclusion as when the traditional p-value is used to indicate significant deviation. You can show what conditions must be fulfilled so that the SGPV is different from 0 but the p-value is less than 0.05.

Assuming the traditional point null hypothesis of 0 and an α of 0.05, then as long as the estimated (confidence) interval does not contain 0, the p-value will be less than 0.05. But any interval null hypothesis that overlaps with the estimated interval will lead to an SGPV greater than 0. By specifying a sufficiently large interval null hypothesis, you can reverse the conclusions of the traditional p-value and the SGPV. But usually, the interval null hypothesis should be relatively small. While the conclusions of the two p-values can seemingly contradict each other, the contradiction can be removed by explicitly stating that the estimated interval contains effects that are equal to 0 when reporting the traditional p-value.

The SGPV informs us about the same result, albeit it forces us to explicitly state what we consider to be an effect equal to 0. So the contradiction becomes a matter of interpreting the statistical reality provided by the estimated interval.

```
. sysuse auto
(1978 automobile data)
. sgpv, all: regress price mpg weight foreign
      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(3, 70)        =     23.29
       Model |   317252881         3   105750960   Prob > F        =    0.0000
    Residual |   317812515        70  4540178.78   R-squared       =    0.4996
-------------+----------------------------------   Adj R-squared   =    0.4781
       Total |   635065396        73  8699525.97   Root MSE        =    2130.8

------------------------------------------------------------------------------
       price | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         mpg |    21.8536   74.22114     0.29   0.769    -126.1758     169.883
      weight |   3.464706    .630749     5.49   0.000     2.206717    4.722695
     foreign |    3673.06   683.9783     5.37   0.000     2308.909    5037.212
       _cons |  -5853.696   3376.987    -1.73   0.087    -12588.88    881.4934
------------------------------------------------------------------------------

Comparison of ordinary P-Values and Second Generation P-Values for a point
Null-Hypothesis of 0 based on a 95% confidence interval
   Variables |   P-Value        SGPV  Delta-Gap          Fdr
-------------+--------------------------------------------
         mpg |     .7693          .5          .            .
      weight |         0           0          .        .0479
     foreign |         0           0          .         .048
       _cons |     .0874          .5          .            .
Warning:
You used the default point 0 null-hypothesis for calculating the SGPVs.
This is allowed but you are strongly encouraged to set a more reasonable
> interval null-hypothesis.
The default point 0 null-hypothesis will result in having SGPVs of either 0
> or 0.5.
```

4.2 Setting an alternative null hypothesis

The second example shows the setting of an alternative null hypothesis and suppresses the calculations of the SGPVs and bonus statistics. The constant in the estimation is also ignored in the calculations by setting the `noconstant` option. The null hypothesis is that any coefficient less than 1% of the mean value of the price variable is not scientifically relevant. Because 1% of the mean equals roughly 62, the lower bound of the interval null hypothesis is -62, and the upper bound of the interval null hypothesis is 62.

```
. sgpv, all nulllo(-62) nullhi(62) quietly noconstant: regress price mpg weight
> foreign
```

Comparison of ordinary P-Values and Second Generation P-Values for an interval Null-Hypothesis of [-62,62] based on a 95% confidence interval

Variables	P-Value	SGPV	Delta-Gap	Fdr
mpg	.7693	.5	.	.
weight	0	1	.	.
foreign	0	0	36.2405	.0394

Compared with the first example, the weight coefficient would be interpreted as being not significantly different from the null hypothesis, while it would be significant under the default point-zero null hypothesis. The delta gap and the FDR for the `foreign` coefficient have also changed.

4.3 Setting multiple null hypotheses

The third example for the `sgpv` command calculates the SGPVs after a simultaneous quantile regression where each variable has its own null hypothesis. The lower and upper bounds of the null hypotheses are displayed in the columns `Null-LB` and `Null-UB`.

```
. set seed 123456
. sgpv, coefficient(mpg weight foreign) nulllo(20 2 3000) nullhi(40 4 6000)
> quietly: sqreg price mpg rep78 foreign weight, quantiles(10 25 50 75 90)
```

Comparison of ordinary P-Values and Second Generation P-Values with an individual null-hypothesis for each variable

Variables	P-Value	SGPV	Null-LB	Null-UB
q10				
mpg	.2606	.5	20	40
weight	.5148	.0995	2	4
foreign	.3055	0	3000	6000
q25				
mpg	.9657	.5	20	40
weight	.1051	.4086	2	4
foreign	.0894	.1281	3000	6000
q50				
mpg	.9376	.5	20	40
weight	.0077	.5121	2	4
foreign	.0105	.5225	3000	6000
q75				
mpg	.6775	.5	20	40
weight	.1635	.5	2	4
foreign	.0506	.4419	3000	6000
q90				
mpg	.8697	.5	20	40
weight	.0157	.5	2	4
foreign	.033	.327	3000	6000

4.4 Using subcommands

The fourth and last example for the sgpv command shows how to use subcommands. The first subcommand, menu, permdialog, creates menu entries for the dialog boxes included in this package in the **User** menu bar and copies the necessary commands to a profile.do file so that the menu items will be added on every invocation of Stata. The second subcommand, power, is a call to the command sgpower.

```
. sgpv menu, permdialog
Append your existing profile.do
Menu entries successfully created.
Go to User->Statistics->SGPV to access the dialog boxes for this package.
. sgpv power, true(2) nulllo(-1) nullhi(1) stderr(1)
power.alt: .1685         power.inc: .8315                  power.null: 0
```

4.5 sgpvalue command example

The next example shows how to use the sgpvalue command to calculate the SGPVs for three different estimated intervals and one common null hypothesis. At first, the lower and upper bounds for estimated intervals are set in the two local macros lb and ub. Then the macros are used as input for the options estlo() and esthi(). The options for estimated intervals and for the null hypotheses accept any valid Stata expression and variable names. By default, the delta gap is also calculated and displayed.

```
. local lb log(1.05) log(1.3) log(0.97)
. local ub log(1.8) log(1.8) log(1.02)
. sgpvalue, estlo(`lb') esthi(`ub') nulllo(log(1/1.1)) nullhi(log(1.1))
Second Generation P-Values
     SGPV  Delta-Gap
---------------------
 .1220227          .
        0   1.752741
        1          .
```

4.6 sgpower command example

The example for the sgpower command calculates the power function for a calculated SGPV. The true value is 2, the standard error of the estimator is equal to 1, and the interval null hypothesis has a lower bound of -1 and upper bound of $+1$. Additional bonus statistics are also displayed.

```
. sgpower, true(2) nulllo(-1) nullhi(1) stderr(1) bonus
power.alt: .1685         power.inc: .8315                  power.null: 0
type I error summaries
at 0: .003077         Min: .003077         Max: .025037         Mean :.009437
```

power.alt denotes the power of the alternative hypothesis, which is the probability of the SGPV being equal to 0 if the true value of the estimated parameter is 2.

`power.inc` denotes the power of an inconclusive SGPV, which is the probability of the SGPV being between 0 and 1 (exclusive) if the true value of the estimated parameter is 2.

`power.null` denotes the power of the null hypothesis, which is the probability of the SGPV being equal to 1 if the true value is 2. The bonus statistics show the minimum, maximum, and mean of the type I error.

4.7 fdrisk command example

By default, the `fdrisk` command calculates the FDR or the FCR by setting the option `fcr`. In this example, the support of the null probability distribution in the option `nullspace()` is set to be equal to the lower and upper bounds of the null hypothesis, for which the SGPV was calculated, set in the options `nulllo()` and `nullhi()`. This is not a requirement but rather a coincidence.

```
. fdrisk, nulllo(log(1/1.1)) nullhi(log(1.1)) stderr(0.8)
> nullspace(log(1/1.1) log(1.1))
> altspace("2-1*invnorm(1-0.05/2)*0.8" "2+1*invnorm(1-0.05/2)*0.8")
The false discovery risk (fdr) is:  .0594986
```

4.8 plotsgpv command example

This example uses `leukstats.dta`, which belongs to the ancillary files of the `sgpv` package. The data in the dataset are from 7,218 gene-specific *t* tests for a difference in mean expression. The `plotsgpv` command is used to create a figure that is similar to figure 3 in Blume et al. (2018). Plotting many CIs with their associated SGPVs can give a quick overview about the results and indicate, for example, which genes deserve further attention. This is a typical scenario in medical research, but this scenario might also be encountered during variable or feature selection when you have a large set of variables to choose from when creating a machine learning model.[18]

In this example, variables are used as inputs for the `esthi()` and `estlo()` options to provide the upper and lower bounds of the estimated CIs. Only the first 7,000 CIs are shown and ordered by the classical *p*-value. By setting the `nomata` option, we create two variables containing the calculated SGPVs (`pdelta`) and delta gaps (`dg`).

```
. use leukstats, clear
. plotsgpv, esthi(ci_hi) estlo(ci_lo) nulllo(-0.3) nullhi(0.3) setorder(p_value)
> xshow(7000) title("Leukemia genes expression")
> xtitle("Genes ordered by classical {it:p}-value")
> ytitle("Fold change (AML versus ALL (base 10)") nullpt(0) nomata
> replace twoway_opt(ylabel(`=log10(1/1000)' "1/1000" `=log10(1/100)'
> "1/100" `=log10(1/10)' "1/10" `=log10(1/2)' "1/2" `=log10(2)'
> "2" =log10(10)' "10" `=log10(100)' "100" `=log10(1000)' "1000"))
```

18. See also Stewart and Blume (2019) and Zuo, Stewart, and Blume (2022) for how to use SGPVs for variable selection.

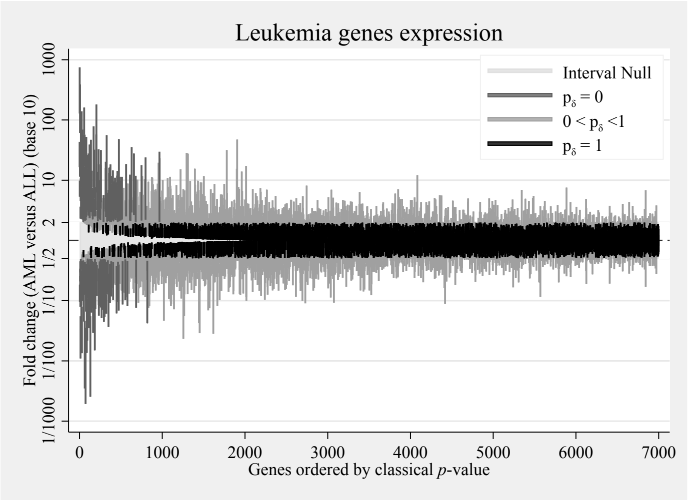


Figure 2. Display of 95% CIs (shown here in grayscale) for gene-specific fold changes (acute myeloid leukemia versus acute lymphoblastic leukemia) in the gene expression levels of patients from the Leukemia Microarray study (Golub et al. 1999) (based on figure 3 in Blume et al. [2018])

5 Conclusion

The `sgpv` package offers an easy way to introduce SGPVs into a user's routine statistical analysis. As stated in the conclusions of Blume et al. (2018), the SGPV is not intended to be the final statistical decision criteria but rather a fast and easy-to-understand summary measure about the statistical evidence. SGPVs offer three regions of values (conclusive evidence for the H_1 or H_0 and inconclusive evidence) for a decision and interpretation instead of two (significant or insignificant) like the traditional p-values.

6 Acknowledgments

This package started out as a response to a thread on Statalist.[19] I am thankful to everyone who has provided feedback regarding this package. The comments received from an anonymous reviewer for the *Stata Journal* were very helpful to improve the quality of this article and make using the package easier. I thank Christopher F. Baum for uploading the package to the Statistical Software Components Archive.

19. See https://www.statalist.org/forums/forum/general-stata-discussion/general/1529372-t-test-p-values-philosophical-question.

7 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 22-3
. net install st0681      (to install program files, if available)
. net get st0681          (to install ancillary files, if available)
```

8 References

Blume, J., and J. F. Peipert. 2003. What your statistician never told you about *p*-values. *Journal of the American Association of Gynecologic Laparoscopists* 10: 439–444. https://doi.org/10.1016/S1074-3804(05)60143-0.

Blume, J. D., L. D'Agostino McGowan, W. D. Dupont, and R. A. Greevy, Jr. 2018. Second-generation *p*-values: Improved rigor, reproducibility, and transparency in statistical analyses. *PLOS ONE* 13: e0188299. https://doi.org/10.1371/journal.pone.0188299.

Blume, J. D., R. A. Greevy, V. F. Welty, J. R. Smith, and W. D. Dupont. 2019. An introduction to second-generation *p*-values. *American Statistician* 73: 157–167. https://doi.org/10.1080/00031305.2018.1537893.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286: 531–537. https://doi.org/10.1126/science.286.5439.531.

Stewart, T. G., and J. D. Blume. 2019. Second-generation *p*-values, shrinkage, and regularized models. *Frontiers in Ecology and Evolution* 7: 486. https://doi.org/10.3389/fevo.2019.00486.

Zuo, Y., T. G. Stewart, and J. D. Blume. 2022. Variable selection with second-generation *p*-values. *American Statistician* 76: 91–101. https://doi.org/10.1080/00031305.2021.1946150.

About the author

Sven-Kristjan Bormann is a PhD student in economics at the University of Tartu, Tartu, Estonia.