

Instructions to specify milestones for analyzing traces from RCmdrTR

Specification of observational milestones

Observation milestones are specified with the following elements:

- Name (must be unique, without spaces) [milNames]
- A regular expression (which may capture some elements) [regExps]
- An R expression which evaluates to a logical value (may be empty "") [logTests]

Each milestone specified in this way will be included in a tab-separated file, with headers, such as the one copied below these lines.

milNames	regExps	logTests
MEx01	name='Command' value='runif%282%29' />	
MEx02	name='Result' value='numeric~~%20([^~]+)'	as.numeric(m[[1]])>=100

Rules to specify a milestone

The suggestion for creating the milestone is to run the action to be detected with RCmdrTR and open the log file that has been created.

Test whether the student has run a specific command

The regular expression for the test will be the fragment of the trace that goes from `name='Command' value='` to the next mark of tag end `' />`. Between these two string we find the URL-encoded command run by user.

In case a period (.) appears in the URL-encoded command, it should be enclosed in square brackets.

The logical evaluation part of the milestone will remain empty in this case.

Examples

If the command being sought is `runif(2)`...

milName	regExps	logTests
MEx01	name='Command' value='runif%282%29' />	

If the command being sought is `with(NormalSamples, (t.test(obs, alternative='two.sided', mu=0.0, conf.level=.95)))...`

milName	regExps	logTests
MEx02	name='Command' value='with%28NormalSamples%2C%20%28t[.] test%28obs%2C%20alternative%3D%27two[.]s ided%27%2C%20mu%3D0[.]0%2C%20conf[.]leve l%3D[.]95%29%29%29' />	

If the command being sought is `with(NormalSamples, Hist(obs, scale="frequency", breaks="Sturges", col="darkgray"))...`

milName	regExps	logTests
MEx03	name='Command' value='with%28NormalSamples%2C%20Hist%28 obs%2C%20scale%3D%22frequency%22%2C%20br eaks%3D%22Sturges%22%2C%20col%3D%22darkg ray%22%29%29' />	

If the command being sought is `9*89...`

milName	regExps	logTests
MEx04	name='Command' value='9%2A89' />	

Test whether the student has obtained a specific result

The regular expression for the test will be the fragment of the trace that goes from `name='Result' value='` to the next mark of tag end `' />`. Between these two string we find an URL-encoded that includes the class of the result and its value. In case it is a vector or a table, it may also include the column names.

In case a period (.) appears in the URL-encoded result, it should be enclosed in square brackets.

The logical evaluation part of the milestone will remain empty in this case.

Examples

If the result being sought is 0.27624900522332...

milName	regExps	logTests
MEx05	name='Result' value='numeric~~%200[.]27624900522332' </>	

If the result being sought is

mean	sd	IQR	0%	25%	50%	75%	100%	n
0.276	0.718	1.024	-0.70	-0.34	0.327	0.681	1.468	10
24900	96436	84911	46350	35912	46302	25787	64864	
52233	68369	60495	48197	44308	60033	17415	86114	
2	66	6	227	031	75	31	3	

milName	regExps	logTests
MEx06	name='Result' value='numSummary~~%20type%2Ctable%2Cstatistics%2Cn~~3~~c%280[.]27624900522332%2C%200[.]718964366836966%2C%201[.]02484911604956%2C%20-0[.]704635048197227%2C%20-0[.]343591244308031%2C%200[.]327463026003375%2C%200[.]681257871741531%2C%201[.]46864864861143%29~~c%28%22mean%22%2C%20%22sd%22%2C%20%22IQR%22%2C%20%22quantiles%22%29~~10' />	

If the results being sought are 0.618130590766668 and 0.182565469993278...

milName	regExps	logTests
MEx07	name='Result' value='numeric~~%200[.]618130590766668~~0[.]182565469993278' />	

Test whether the student has used a specific function or statistical procedure

The regular expression for the test will be `name='Command' value='.*?` followed by the function name and `%28`. In case a period (.) appears in the name of the function, it should be enclosed in square brackets.

The logical evaluation part of the milestone will also remain empty in this case.

Examples

If the function being sought is `rnorm`...

milName	regExps	logTests
MEx08	<code>name='Command' value='.*?rnorm%28</code>	

If we look whether the student has done a histogram...

milName	regExps	logTests
MEx09	<code>name='Command' value='.*?Hist%28</code>	

Test whether the student has run a function or statistical procedure among different options

The regular expression for the test will start with `name='Command' value='.*?`. Next, a group with the different options of functions follows. The group and each of the function are enclosed between `(?:` and `)`. The groups for each function are separated by a pipe character `|`. Each function should be followed by `%28`. In case a period (.) appears in the name of the functions, it should be enclosed in square brackets.

The logical evaluation part of the milestone will also remain empty in this case.

Examples

If the function being sought is `t.test` or `wilcox.test`...

milName	regExps	logTests
MEx10	<code>name='Command' value='.*?(?: (?:t[.]test%28) (?:wilcox[.] test%28))</code>	

Test whether the student has obtained a result within an interval

In order to check a numerical value from a part of a regular expression, we will need to capture it and evaluate a logical test from the captured data. To capture some data from the result, the regular expression will start with `name='Result' value='`. It will then include whatever characters needed to reach the number to be collected. To collect the number, we will use a captured group that includes as many characters as needed such they are different than the character that comes after the number, e.g. `( [^%] )` if a `%` follows the number.

This captured string will then be evaluated in an R expression. The captured strings are stored in a list named `m` that is available for the R logical calculation.

Examples

To check if the students obtains a result larger than or equal to 100...

milName	regExps	logTests
MEx11	<code>name='Result'</code> <code>value='numeric~~%20([^'~]+) '</code>	<code>as.numeric(m[1])>=100</code>

To check whether the student has made a summary of a dataset which mean is between 0 and 1. The summary can be detected by the class of the result and the mean is collected knowing that this is the first number that appears after `c%28`.

milName	regExps	logTests
MEx12	<code>name='Result'</code> <code>value='numSummary~~.*?c%28([^%]+) %2C</code>	<code>as.numeric(m[1])>=0 &</code> <code>as.numeric(m[1])<=1</code>

Test whether the student has run a command with an argument within an interval

To test whether a student has run a command that includes an argument expected to be in a specific interval or to comply any other condition, the milestones can be constructed to capture the argument and then the argument can be logically tested. In this case, the regular expression part of the milestone will start with `name='Command' value='` and will follow with the function and whatever characters are needed to reach the position of the argument

to be collected. To pick the argument, we will use a captured group that includes as many characters as needed to reach the character that comes after it, i.e a comma (`%2C`) or a closing bracket (`%29`).

This captured string will then be evaluated in an R expression. The captured strings are stored in a list named `m` that is available for the R logical calculation.

Example

To check if the students has created a sample with at least 50 uniformly distributed random numbers...

milName	regExps	logTests
MEx13	name='Command' value='UniformSamples%20%3C-%20as.data.frame%28matrix%28runif%28(.*)%2C	as.numeric(m[1])>=50

References

- <https://www.datacamp.com/community/blog/submission-correctness-testing-in-datacamp-datacamp-part-i#gs.Q2uf1CU>
- <http://www.regular-expressions.info/tutorial.html>
- <https://regexone.com/>
- <http://web.mit.edu/hackl/www/lab/turkshop/slides/regex-cheatsheet.pdf>
- <https://www.princeton.edu/~mlovett/reference/Regular-Expressions.pdf>

