

Package: MetaculR (via r-universe)

August 31, 2024

Title Analyze Metaculus Predictions and Questions

Version 0.5.0

URL <https://ntrlshrp.gitlab.io/metaculr>,
<https://gitlab.com/ntrlshrp/metaculr>

BugReports <https://gitlab.com/ntrlshrp/metaculr/-/issues>

Description Login, download, and analyze questions predicted by you and/or the Metaculus community by interacting with the Metaculus API, currently located at <https://www.metaculus.com/api2/>.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

RdMacros mathjaxr

Suggests httpptest, knitr, rmarkdown, testthat

VignetteBuilder knitr

Imports magrittr, dplyr, ggplot2, httr, jsonlite, progress, tidyr, verification, stats, clipr, spatstat.geom, ggrepel, assertthat, cowplot, lubridate, Hmisc, mathjaxr

Repository <https://josephd000.r-universe.dev>

RemoteUrl <https://gitlab.com/ntrlshrp/metaculr>

RemoteRef HEAD

RemoteSha 3a923e0909bd9ceb326454a9fe4581010c141d9d

Contents

MetaculR_aggregated_forecasts	2
MetaculR_analysis_binary_resolved	3
MetaculR_brier	6

MetaculR_categories	8
MetaculR_excitement	9
MetaculR_login	10
MetaculR_markdown_table	10
MetaculR_myCategories	11
MetaculR_myChallenges	12
MetaculR_myDiff	13
MetaculR_myPredictions	14
MetaculR_myPredictions_Resolved	15
MetaculR_plot	16
MetaculR_predict	17
MetaculR_probabilistic_consensus	18
MetaculR_questions	19
MetaculR_review	21

Index	22
--------------	-----------

MetaculR_aggregated_forecasts

Aggregate Community Forecasts for MetaculR

Description

Provides different results of aggregating current community forecasts to help you make your next forecast.

Usage

```
MetaculR_aggregated_forecasts(MetaculR_questions, Metaculus_id, baseline = 0.5)
```

Arguments

MetaculR_questions	A MetaculR_questions object
Metaculus_id	The ID of the question to plot
baseline	Climatological baseline for binary questions

Details

Sevilla (2021) found a Metaculus baseline of 0.36 looking at ~900 questions. While Sevilla has at times referred to the geometric mean of odds, this function uses the equivalent mean of logodds. Also note that $\mu + (d - 1)(\mu + b)$ (Neyman & Roughgarden) is equivalent to $b + d(\mu + b)$, this function uses the former.

Value

A dataframe of forecast aggregations.

id	Question ID.
community_q2	Community median.
community_ave	Community mean.
community_q2_unweighted	Community median, unweighted by recency.
community_ave_unweighted	Community mean, unweighted by recency.
community_mean_logodds	Community mean of logodds.
community_mean_logodds_extremized_baseline	Community mean of logodds, extremized with reference to a baseline. If the baseline is 0.5, this is "classical extremizing."

References

Neyman, E., & Roughgarden, T. (2022). Are You Smarter Than a Random Expert? The Robust Aggregation of Substitutable Signals. ArXiv:2111.03153 [Cs]. <https://arxiv.org/abs/2111.03153>

Sevilla, J. (2021, December 29). Principled extremizing of aggregated forecasts. <https://forum.effectivealtruism.org/posts/biL94PKfeHmgHY6qe/principled-extremizing-of-aggregated-forecasts>

Examples

```
## Not run:
MetaculR_aggregate_forecasts(
  MetaculR_questions = questions_myPredictions,
  Metaculus_id = 10004)

## End(Not run)
```

MetaculR_analysis_binary_resolved

Make dataframe of resolved questions for analysis

Description

Make dataframe of resolved questions for analysis

Usage

```
MetaculR_analysis_binary_resolved(MetaculR_questions)
```

Arguments

MetaculR_questions
A MetaculR_questions object

Value

A large dataframe of resolved questions by tick.

id	The Metaculus question ID.
Date	Seconds since 1970-01-01 00:00.00 UTC.
obs	Observed resolution.
np	Number of predictions.
nu	Number of predictors.
c_q1	Community 25th centile.
c_q2	Community median.
c_q3	Community 75th centile.
c_ave	Community mean.
c_var	Community variance.
m_q2	Metaculus prediction.
x	Self prediction.
title	Question title.
Date_open	Date opened.
Date_close	Date scheduled to close.
Date_resolve	Date actually resolved.
c_q2_rnd	Community median, rounded to 0.01 - 0.99, 2 digits.
m_q2_rnd	Metaculus prediction, rounded to 0.01 - 0.99, 2 digits.
Count_pred	Count of Self predictions.
Tick	Tick by question.
Countdown_tick	Ticks remaining.
Countdown_weeks_Close	Weeks until Date_close.
Countdown_weeks_Resolve	Weeks until Date_resolve.
Close_Pct	Percentage of open to close time.
Resolve_Pct	Percentage of open to resolve time.
Cum_Close_Pct	Cumulative percentage of open to close time.
Weight_Resolve	Weights for each question to have equal weighted ticks to resolve.
Weight_Close	Weights for each question to have equal weighted ticks to close.
Brier_me	Self Brier score of tick.
Brier_comm	Community Brier score of tick.

Brier_met Metaculus Brier score of tick.
Brier_comm_rnd Community-rounded Brier score of tick.
Brier_met_rnd Metaculus-rounded Brier score of tick.
Log_me Self Log score of tick.
Log_comm Community Log score of tick.
Log_met Metaculus Log score of tick.
Log_comm_rnd Community-rounded Log score of tick.
Log_met_rnd Metaculus-rounded Log score of tick.
Overconfidence_me Self Overconfidence score of tick.
Overconfidence_comm Community Overconfidence score of tick.
Overconfidence_met Metaculus Overconfidence score of tick.
Overconfidence_comm_rnd Community-rounded Overconfidence score of tick.
Overconfidence_met_rnd Metaculus-rounded Overconfidence score of tick.
RelLogScore_me Self Relative Log score of tick, compared to Community median.
RelLogScore_met Metaculus Relative Log score of tick, compared to Community median.
RelLogScore_met_rnd Metaculus-rounded Relative Log score of tick, compared to Community median.
Duration Number of seconds tick in effect.
Cumulative versions of the above

Cum_Brier_me
Cum_Brier_comm
Cum_Brier_met
Cum_Brier_comm_rnd

Cum_Brier_met_rnd

Cum_Log_me
Cum_Log_comm
Cum_Log_met
Cum_Log_comm_rnd

Cum_Log_met_rnd

Cum_RelLogScore_me
Cum_RelLogScore_met
Cum_RelLogScore_met_rnd

Examples

```
## Not run:
questions_resolved_analysis_binary <-
  MetaculR_analysis_binary_resolved(
    questions_resolved)

## End(Not run)
```

MetaculR_brier

*Calculate Brier statistics on MetaculR_analysis_binary object***Description**

Calculate Brier statistics on MetaculR_analysis_binary object

Usage

```
MetaculR_brier(
  MetaculR_analysis_binary,
  me = TRUE,
  time = c("resolve", "close", "all"),
  unit = c("moment", "question", "second"),
  thresholds = seq(0, 1, 0.1)
)
```

Arguments

MetaculR_analysis_binary	A MetaculR_analysis_binary object
me	Use scores only during periods with my predictions
time	When to use scores: c("resolve", "close", "all") (See details.)
unit	Scoring unit for weights: c("moment", "question", "second") (See details.)
thresholds	Thresholds to bin questions

Details

$$B_{T,U} = REL_{T,U} - RES_{T,U} + UNC_{T,U}$$

$$REL_{T,U} = \frac{1}{w_{it}} \sum \left(\frac{\sum p_{itb} \times w_{itb}}{\sum w_{itb}} - \frac{\sum o_{itb} \times w_{itb}}{\sum w_{itb}} \right)^2$$

$$RES_{T,U} = \frac{1}{w_{it}} \sum \left(\frac{\sum o_{itb} \times w_{itb}}{\sum w_{itb}} - \frac{\sum o_{it} \times w_{it}}{\sum w_{it}} \right)^2$$

$$UNC_{T,U} = \frac{\sum o_{it} \times w_{it}}{\sum w_{it}} \left(1 - \frac{\sum o_{it} \times w_{it}}{\sum w_{it}} \right)$$

where $B_{T,U}$ is the Brier score, $REL_{T,U}$ is the Reliability component, $RES_{T,U}$ is the Resolution component, $UNC_{T,U}$ is the Uncertainty component, p_{itb} is the prediction for question i at time t in bin b , o_i is the observed resolution for question i , and w_{it} is the weight assigned to the prediction for question i at time t . The weight assigned depends on the parameters used,

$$w_{it} = \begin{cases} 1, & T = \text{resolve}, [U = \text{moment}], t = t_{i,R}, \\ \frac{t_{i,k+1} - t_{i,k}}{t_{i,C} - t_{i,O}}, & T = \text{close}, U = \text{question}, t \leq t_{i,C}, \\ t_{i,k+1} - t_{i,k}, & T = \text{close}, U = \text{second}, t \leq t_{i,C}, \\ \frac{t_{i,k+1} - t_{i,k}}{t_{i,R} - t_{i,O}}, & T = \text{resolve}, U = \text{question}, t \leq t_{i,R}, \\ t_{i,k+1} - t_{i,k}, & T = \text{resolve}, U = \text{second}, t \leq t_{i,R}. \end{cases}$$

where $t_{i,k}$ is the time of the tick k for question i , $t_{i,R}$, $t_{i,C}$, and $t_{i,O}$ are, respectively, the resolve, close, and open time of question i . As this function is concerned with comparisons among Self, Community, and Metaculus, time t is only used if all parties have registered a prediction. That is, if you made a prediction 20% into a 10-day question and another prediction 80% into a 10-month question, the Community and Metaculus Brier scores will not account for any of their predictions prior to your first prediction in either question. Lastly, if `unit = "question"`, the last 80% of the 10-day question will receive 4x the weight of the last 20% of the 10-month question.

Value

A list of Brier statistics for you and Metaculus.

`brier_me`, `brier_Metaculus`, `brier_community`

<code>baseline.tf</code>	Logical indicator of whether climatology was provided.
<code>bs</code>	Brier score
<code>bs.baseline</code>	Brier Score for climatology
<code>ss</code>	Skill score
<code>bs.reliability</code>	Reliability portion of Brier score.
<code>bs.resolution</code>	Resolution component of Brier score.
<code>bs.uncert</code>	Uncertainty component of Brier score.
<code>y.i</code>	Forecast bins – described as the center value of the bins.
<code>obar.i</code>	Observation bins – described as the center value of the bins.
<code>prob.y</code>	Proportion of time using each forecast.
<code>obar</code>	Forecast based on climatology or average sample observations.
<code>thresholds</code>	The thresholds for the forecast bins.
<code>check</code>	Reliability - resolution + uncertainty should equal brier score.
Other	
<code>ss_me_Metaculus</code> , <code>ss_me_community</code> , <code>ss_Metaculus_community</code>	Skill score, me vs. Metaculus, etc.
<code>questions</code>	Dataframe of questions included.
<code>id</code>	Question ID.

title	Question title.
obs	Observed resolution.
brier_df	Used for plotting Brier score statistics
ID	Predictor.
name	Name of value, see above.
value	Value.
brier_bins_df	Used for plotting histogram and calibration plots.
ID	Predictor.
centers	y.i, see above.
freqs	prob.y, see above.
obars	obar.i, see above.
ideal	Ideal calibration where centers equals obars.
ci_low	Low end of 95% confidence interval for obar.i.
ci_high	High end of 95% confidence interval for obar.i.

Examples

```
## Not run:
brier_me <-
  MetaculR_brier(
    questions_resolved_analysis_binary)

## End(Not run)
```

MetaculR_categories *One hot encode categories for questions from Metaculus API*

Description

One hot encode categories for questions from Metaculus API

Usage

```
MetaculR_categories(api_domain = "www", ids = NULL)
```

Arguments

api_domain	Use "www" unless you have a custom Metaculus domain
ids	A vector of Metaculus question IDs

Value

A dataframe of questions, with one hot encoded categories.

See Also

Other Question Retrieval functions: [MetaculR_myPredictions_Resolved\(\)](#), [MetaculR_myPredictions\(\)](#), [MetaculR_questions\(\)](#)

Examples

```
## Not run:
questions_categories <-
  MetaculR_categories(
    ids = questions_resolved_analysis_binary %>%
      dplyr::distinct(id) %>%
      dplyr::pull()

## End(Not run)
```

MetaculR_excitement *Find exciting questions*

Description

Find exciting questions

Usage

```
MetaculR_excitement(MetaculR_questions, days = 30)
```

Arguments

MetaculR_questions	A MetaculR_questions object
days	The time period used for the excitement calculations starts this number of days ago, prior to today. E.g., if your clock says it is day 12 and your days argument is 10, the time period is day 2 until the present.

Value

A dataframe of questions with excitement measures.

id	Question ID.
title	Question title.
Total_Change	Cumulative delta in time period, by probability.
Total_logodds_Change	Cumulative delta in time period, by logodds.
Total_Change_Even	Cumulative delta toward even odds in time period, by probability.
Total_logodds_Change_Even	Cumulative delta toward even odds in time period, by logodds.

Examples

```
## Not run:
questions_myPredictions_byExcitement <-
  MetaculR_excitement(
    questions_myPredictions)

## End(Not run)
```

MetaculR_login	<i>Login to Metaculus</i>
----------------	---------------------------

Description

Login to Metaculus

Usage

```
MetaculR_login(api_domain = "www")
```

Arguments

api_domain Use "www" unless you have a custom Metaculus domain

Value

Your Metaculus_user_ID.

Examples

```
## Not run:
Metaculus_user_id <-
  MetaculR_login()

## End(Not run)
```

MetaculR_markdown_table	<i>Easily translate R dataframes to Metaculus Markdown</i>
-------------------------	--

Description

Easily translate R dataframes to Metaculus Markdown

Usage

```
MetaculR_markdown_table(df)
```

Arguments

df A dataframe.

Value

A Markdown table.

Examples

```
## Not run:
my_data <- data.frame(Year = c(2020,2021), Value = c(6, 7.2))

MetaculR_markdown_table(my_data)

## End(Not run)
```

MetaculR_myCategories *Plot categories sorted by Brier score*

Description

Plot categories sorted by Brier score

Usage

```
MetaculR_myCategories(
  MetaculR_analysis_binary = NULL,
  MetaculR_categories = NULL,
  me = TRUE
)
```

Arguments

MetaculR_analysis_binary
 A MetaculR analysis binary object

MetaculR_categories
 A MetaculR categories object

me Focus only on categories in which I've made a prediction and only on my Brier scores

Value

A ggplot

Examples

```
## Not run:
questions_categories <-
  MetaculR_categories(
    ids = questions_resolved_analysis_binary %>%
      dplyr::distinct(id) %>%
      dplyr::pull()

  MetaculR_myCategories(
    MetaculR_analysis_binary = questions_resolved_analysis_binary,
    MetaculR_categories = questions_categories)

## End(Not run)
```

MetaculR_myChallenges *Plot Brier scores by question, sorted by comparison to Community median*

Description

Plot Brier scores by question, sorted by comparison to Community median

Usage

```
MetaculR_myChallenges(MetaculR_analysis_binary = NULL, me = TRUE)
```

Arguments

MetaculR_analysis_binary
A MetaculR analysis binary object

me
Focus only on questions in which I've made a prediction

Value

A plot

Examples

```
## Not run:
MetaculR_myChallenges(
  MetaculR_analysis_binary = questions_resolved_analysis_binary)

## End(Not run)
```

MetaculR_myDiff	<i>Find important changes within MetaculR_questions object</i>
-----------------	--

Description

Find important changes within MetaculR_questions object

Usage

```
MetaculR_myDiff(MetaculR_questions)
```

Arguments

MetaculR_questions
A MetaculR_questions object

Value

A dataframe of questions with difference measures (your most recent prediction vs. community's most recent prediction, etc.).

id	Question ID.
title	Question title.
my_prediction	My most recent prediction.
community_q2	Community median.
community_ave	Community average.
community_q2_pre_me	Community median immediately prior to my_prediction.
community_ave_pre_me	Community average immediately prior to my_prediction.
diff_me_q2	Difference between me and the community median, by logodds.
diff_me_ave	Difference between me and the community average, by logodds.
diff_comm_q2_pre_me	Difference between community_q2_pre_me and the community average, by logodds.
diff_comm_ave_pre_me	Difference between community_ave_pre_me and the community average, by logodds.
diff_me_q2_abs	Absolute difference between me and the community median, by logodds.
diff_me_ave_abs	Absolute difference between me and the community average, by logodds.
diff_comm_q2_pre_me_abs	Absolute difference between community_q2_pre_me and the community average, by logodds.

```

diff_comm_ave_pre_me_abs
    Absolute difference between community_ave_pre_me and the community average, by logodds.
diff_me_q2_abs_odds
    Absolute difference between me and the community median, by odds.
diff_me_ave_abs_odds
    Absolute difference between me and the community average, by odds.
diff_comm_q2_pre_me_abs_odds
    Absolute difference between community_q2_pre_me and the community average, by odds.
diff_comm_ave_pre_me_abs_odds
    Absolute difference between community_ave_pre_me and the community average, by odds.

```

Examples

```

## Not run:
questions_myPredictions_byDiff <-
  MetaculR_myDiff(
    questions_myPredictions)

## End(Not run)

```

MetaculR_myPredictions

Retrieve questions from Metaculus API (A wrapper for MetaculR_questions())

Description

Retrieve questions from Metaculus API (A wrapper for MetaculR_questions())

Usage

```

MetaculR_myPredictions(
  api_domain = "www",
  order_by = "last_prediction_time",
  status = "all",
  search = "",
  guessed_by = "",
  offset = 0,
  pages = 10
)

```

Arguments

api_domain	Use "www" unless you have a custom Metaculus domain
order_by	Default is "last_prediction_time"
status	Choose "all", "upcoming", "open", "closed", "resolved"
search	Search term(s)
guessed_by	Generally your Metaculus_user_id
offset	Question offset
pages	Number of pages to request

Value

A list of questions that I've predicted, ordered by last prediction time.

See Also

Other Question Retrieval functions: [MetaculR_categories\(\)](#), [MetaculR_myPredictions_Resolved\(\)](#), [MetaculR_questions\(\)](#)

Examples

```
## Not run:
questions_myPredictions <-
  MetaculR_myPredictions(
    guessed_by = Metaculus_user_id)

## End(Not run)
```

MetaculR_myPredictions_Resolved

Retrieve questions from Metaculus API (A wrapper for MetaculR_questions())

Description

Retrieve questions from Metaculus API (A wrapper for MetaculR_questions())

Usage

```
MetaculR_myPredictions_Resolved(
  api_domain = "www",
  order_by = "-resolve_time",
  status = "resolved",
  search = "",
  guessed_by = "",
  offset = 0,
  pages = 10
)
```

Arguments

api_domain	Use "www" unless you have a custom Metaculus domain
order_by	Default is "-resolve_time"
status	Default is "resolved"
search	Search term(s)
guessed_by	Generally your Metaculus_user_id
offset	Question offset
pages	Number of pages to request

Value

A list of questions that I've predicted, ordered by last prediction time, and resolved.

See Also

Other Question Retrieval functions: [MetaculR_categories\(\)](#), [MetaculR_myPredictions\(\)](#), [MetaculR_questions\(\)](#)

Examples

```
## Not run:
questions_myPredictions_resolved <-
  MetaculR_myPredictions_Resolved(
    guessed_by = Metaculus_user_id)

## End(Not run)
```

MetaculR_plot

Plot the history of a single question

Description

Plot the history of a single question

Usage

```
MetaculR_plot(
  MetaculR_questions,
  Metaculus_id,
  scale_binary = "prob",
  tournament = FALSE
)
```


Arguments

MetaculR_questions	A MetaculR_questions object
Metaculus_id	The ID of the question to plot
scale_binary	Choose "prob", "odds", or "logodds"
tournament	Plot relative log score below main plot

Value

A ggplot.

Examples

```
## Not run:
MetaculR_plot(
  MetaculR_questions = questions_myPredictions,
  Metaculus_id = 10004)

## End(Not run)
```

MetaculR_predict

Make predictions via Metaculus API

Description

Make predictions via Metaculus API

Usage

```
MetaculR_predict(
  api_domain = "www",
  Metaculus_id = NULL,
  prediction = NULL,
  csrftoken = NULL
)
```

Arguments

api_domain	Use "www" unless you have a custom Metaculus domain
Metaculus_id	The ID of the question to predict
prediction	Your new prediction for the question, e.g., .25 or 1:3
csrftoken	The csrftoken returned by MetaculR_login()

Value

API response

Examples

```
## Not run:
Metaculus_response_login <- MetaculR_login()

MetaculR_predict(
  Metaculus_id = 10004,
  prediction = 0.42, # prediction = "42:58"
  csrftoken = Metaculus_response_login$csrftoken)

## End(Not run)
```

MetaculR_probabilistic_consensus

Generate probabilistic consensus from multiple parameterized forecasts

Description

Generate probabilistic consensus from multiple parameterized forecasts

Usage

```
MetaculR_probabilistic_consensus(f)
```

Arguments

f A list of forecasts (see example for necessary structure).

Value

A list of forecasts.

pdf A dataframe of probability density functions corresponding to original forecasts and consensus forecast.

cdf A dataframe of cumulative distribution functions corresponding to original forecasts and consensus forecast.

summary A dataframe of summary statistics corresponding to original forecasts and consensus forecast, i.e., 10th, 25th, 50th, 75th, 90th centiles and mean.

References

McAndrew, T., & Reich, N. G. (2020). An expert judgment model to predict early stages of the COVID-19 outbreak in the United States [Preprint]. *Infectious Diseases (except HIV/AIDS)*. <https://doi.org/10.1101/2020.09.21.20196725>

Examples

```

## Not run:
forecasts <- list(list(range = c(0, 250), resolution = 1),
  list(source = "Pishkalo",
    dist = "Norm",
    params = c("mu", "sd"),
    values = c(116, 12),
    weight = 0.2),
  list(source = "Miao",
    dist = "Norm",
    params = c("mu", "sd"),
    values = c(121.5, 32.9)),
  list(source = "Labonville",
    dist = "TPD",
    params = c("min", "mode", "max"),
    values = c(89-14, 89, 89+29)),
  list(source = "NOAA",
    dist = "PCT",
    params = c(0.2, 0.8),
    values = c(95, 130)),
  list(source = "Han",
    dist = "Norm",
    params = c("mu", "sd"),
    values = c(228, 40.5)),
  list(source = "Dani",
    dist = "Norm",
    params = c("mu", "sd"),
    values = c(159, 22.3)),
  list(source = "Li",
    dist = "Norm",
    params = c("mu", "sd"),
    values = c(168, 6.3)),
  list(source = "Singh",
    dist = "Norm",
    params = c("mu", "sd"),
    values = c(89, 9)))

MetaculR_probabilistic_consensus(
  f = forecasts)

## End(Not run)

```

MetaculR_questions *Retrieve questions from Metaculus API*

Description

Retrieve questions from Metaculus API

Usage

```
MetaculR_questions(  
  api_domain = "www",  
  order_by = "last_prediction_time",  
  status = "all",  
  search = "",  
  guessed_by = "",  
  offset = 0,  
  pages = 10  
)
```

Arguments

api_domain	Use "www" unless you have a custom Metaculus domain
order_by	Choose "last_prediction_time", "-activity", "-votes", "-publish_time", "close_time", "resolve_time", "last_prediction_time"
status	Choose "all", "upcoming", "open", "closed", "resolved"
search	Search term(s)
guessed_by	Generally your Metaculus_user_id
offset	Question offset
pages	Number of pages to request

Value

A list of questions, ordered by last prediction time.

See Also

Other Question Retrieval functions: [MetaculR_categories\(\)](#), [MetaculR_myPredictions_Resolved\(\)](#), [MetaculR_myPredictions\(\)](#)

Examples

```
## Not run:  
questions_recent_open <-  
  MetaculR_questions(  
    order_by = "close_time",  
    status = "open",  
    guessed_by = "")  
  
## End(Not run)
```

MetaculR_review	<i>Systematically review your predictions</i>
-----------------	---

Description

This currently only works for binary questions.

Usage

```
MetaculR_review(MetaculR_questions_open = NULL, csrftoken = NULL, offset = 0)
```

Arguments

MetaculR_questions_open	MetaculR_questions object of your open questions.
csrftoken	The csrftoken returned by MetaculR_login()
offset	An offset to start at question 8/47 if you've already reviewed questions 1 - 7.

Value

Plots

Examples

```
## Not run:
questions_recent_open <-
  MetaculR_questions(order_by = "close_time",
                    status = "open",
                    guessed_by = MetaculR_response_login$Metaculus_user_id)

MetaculR_review(questions_recent_open,
               MetaculR_response_login$csrftoken)

## End(Not run)
```

Index

* Question Retrieval functions

- MetaculR_categories, [8](#)
- MetaculR_myPredictions, [14](#)
- MetaculR_myPredictions_Resolved,
[15](#)
- MetaculR_questions, [19](#)

- MetaculR_aggregated_forecasts, [2](#)
- MetaculR_analysis_binary_resolved, [3](#)
- MetaculR_brier, [6](#)
- MetaculR_categories, [8](#), [15](#), [16](#), [20](#)
- MetaculR_excitement, [9](#)
- MetaculR_login, [10](#)
- MetaculR_markdown_table, [10](#)
- MetaculR_myCategories, [11](#)
- MetaculR_myChallenges, [12](#)
- MetaculR_myDiff, [13](#)
- MetaculR_myPredictions, [9](#), [14](#), [16](#), [20](#)
- MetaculR_myPredictions_Resolved, [9](#), [15](#),
[15](#), [20](#)
- MetaculR_plot, [16](#)
- MetaculR_predict, [17](#)
- MetaculR_probabilistic_consensus, [18](#)
- MetaculR_questions, [9](#), [15](#), [16](#), [19](#)
- MetaculR_review, [21](#)