

Package ‘uniswapper’

May 27, 2022

Type Package

Title Interact with the Uniswap Platform

Version 0.6.0

Description

Routines to interact with the Uniswap trading platform and its API <<https://uniswap.org>>. The package contains codebase to interact with the uniswap platform directly from R console, Ability to pull and export data related to the platform and analyse some aspects.

Depends R (>= 3.1)

License GPL-3

Encoding UTF-8

URL <https://github.com/Omni-Analytics-Group/uniswapper>

BugReports <https://github.com/Omni-Analytics-Group/uniswapper/issues>

RoxygenNote 7.2.0

Imports httr, lubridate, dplyr, rlang, tibble, tidyr, ggplot2, scales, ghql, jsonlite, patchwork, purrr, reticulate, utils

NeedsCompilation no

Author Omni Analytics Group [aut],
Eric Hare [cre]

Maintainer Eric Hare <eric@omnianalytics.io>

Repository CRAN

Date/Publication 2022-05-27 08:00:05 UTC

R topics documented:

check_eth.to.tok_eth.fix	3
check_eth.to.tok_tok.fix	4
check_eth_balance	4
check_tok.to.eth_eth.fix	5
check_tok.to.eth_tok.fix	5

check_tok1.to.tok2_tok1.fix	6
check_tok1.to.tok2_tok2.fix	6
check_tok_balance	7
export_data	8
export_plot	8
factory_stats_v2	9
factory_stats_v3	9
get_infura_node	10
liquidity_range_all_v3	10
liquidity_range_v3	11
liquidity_range_visualization	12
pairs_all_v2	12
pairs_all_v3	13
pair_burn_txs_v2	13
pair_burn_txs_v3	14
pair_liq_positions_hist_v2	14
pair_liq_positions_hist_v3	15
pair_liq_positions_v2	15
pair_liq_positions_v3	16
pair_mint_txs_v2	17
pair_mint_txs_v3	17
pair_stats_hist_daily_v2	18
pair_stats_hist_daily_v3	18
pair_stats_hist_hourly_v2	19
pair_stats_hist_hourly_v3	20
pair_stats_v2	20
pair_stats_v3	21
pair_swap_txs_v2	21
pair_swap_txs_v3	22
py_int	23
r_num	23
set_infura_node	24
swaps	24
swap_performance	25
swap_statistics	25
swap_visualizations	26
token_pair_map_v2	27
token_pair_map_v3	27
token_stats_hist_v2	28
token_stats_hist_v3	28
token_stats_v2	29
token_stats_v3	30
trade_eth.to.tok_eth.fix	30
trade_eth.to.tok_tok.fix	31
trade_tok.to.eth_eth.fix	31
trade_tok.to.eth_tok.fix	32
trade_tok1.to.tok2_tok1.fix	32
trade_tok1.to.tok2_tok2.fix	33

uniswap_session	33
uniswap_stats_hist_v2	34
uniswap_stats_hist_v3	34
user_burns_v2	35
user_burns_v3	36
user_hist_lps_v2	36
user_hist_lps_v3	37
user_lps_v2	37
user_lps_v3	38
user_mints_v2	39
user_mints_v3	39
user_swaps_v2	40
user_swaps_v3	40
vis_pair_liq_positions_v2	41
vis_pair_liq_positions_v3	42
vis_pair_stats_hist_daily_v2	42
vis_pair_stats_hist_daily_v3	43
vis_token_pair_map_v2	43
vis_token_pair_map_v3	44
vis_token_stats_hist_v2	45
vis_token_stats_hist_v3	45
vis_uniswap_stats_hist_v2	46
vis_uniswap_stats_hist_v3	46

Index	47
--------------	-----------

check_eth.to.tok_eth.fix

Swap ETH for a Token, Given ETH Qty check how much token you would get

Description

Swap ETH for a Token, Given ETH Qty check how much token you would get

Usage

check_eth.to.tok_eth.fix(t_a, t_d, e_q, u_w)

Arguments

t_a	Token Address
t_d	Token Decimals
e_q	Ethereum Qty.
u_w	Uniswap Session

Value

Numeric vector representing the Token Amount you get

```
check_eth.to.tok_tok.fix
```

Swap ETH for a Token, Given Token Qty check how much ETH you need

Description

Swap ETH for a Token, Given Token Qty check how much ETH you need

Usage

```
check_eth.to.tok_tok.fix(t_a, t_d, t_q, u_w)
```

Arguments

t_a	Token Address
t_d	Token Decimals
t_q	Token Qty.
u_w	Uniswap Session

Value

Numeric vector representing the ETH needed

```
check_eth_balance      Get your ETH Balance
```

Description

Get your ETH Balance

Usage

```
check_eth_balance(u_w)
```

Arguments

u_w	Uniswap Session
-----	-----------------

Value

Numeric vector representing the User Ethereum balance

check_tok.to.eth_eth.fix

Swap Token for ETH, Given ETH Qty check how much Token you need

Description

Swap Token for ETH, Given ETH Qty check how much Token you need

Usage

check_tok.to.eth_eth.fix(t_a, t_d, e_q, u_w)

Arguments

t_a	Token Address
t_d	Token Decimals
e_q	Ethereum Qty.
u_w	Uniswap Session

Value

Numeric vector representing the Token Amount Needed

check_tok.to.eth_tok.fix

Swap Token for ETH, Given Token Qty check how much ETH you would get

Description

Swap Token for ETH, Given Token Qty check how much ETH you would get

Usage

check_tok.to.eth_tok.fix(t_a, t_d, t_q, u_w)

Arguments

t_a	Token Address
t_d	Token Decimals
t_q	Token Qty.
u_w	Uniswap Session

Value

Numeric vector representing the ETH Amount you get

check_tok1.to.tok2_tok1.fix

Swap Token1 for Token2, Given Token1 Qty check how much Token2 you would get (Use Token1 -> ETH -> Token2 Route)

Description

Swap Token1 for Token2, Given Token1 Qty check how much Token2 you would get (Use Token1 -> ETH -> Token2 Route)

Usage

check_tok1.to.tok2_tok1.fix(t1_a, t1_d, t2_a, t2_d, t1_q, u_w)

Arguments

t1_a	Token 1 Address
t1_d	Token 1 Decimals
t2_a	Token 2 Address
t2_d	Token 2 Decimals
t1_q	Token 1 Qty.
u_w	Uniswap Session

Value

Numeric vector representing the Token 2 Amount you get

check_tok1.to.tok2_tok2.fix

Swap Token1 for Token2, Given Token2 Qty check how much Token1 you would need (Use Token1 -> ETH -> Token2 Route)

Description

Swap Token1 for Token2, Given Token2 Qty check how much Token1 you would need (Use Token1 -> ETH -> Token2 Route)

Usage

check_tok1.to.tok2_tok2.fix(t1_a, t1_d, t2_a, t2_d, t2_q, u_w)

Arguments

t1_a	Token 1 Address
t1_d	Token 1 Decimals
t2_a	Token 2 Address
t2_d	Token 2 Decimals
t2_q	Token 2 Qty.
u_w	Uniswap Session

Value

Numeric vector representing the Token 1 Amount Needed

check_tok_balance *Check any Token Balance*

Description

Check any Token Balance

Usage

check_tok_balance(t_a, t_d, u_w)

Arguments

t_a	Token Address
t_d	Token Decimals
u_w	Uniswap Session

Value

Numeric vector representing the User Token balance

export_data	<i>Write out the data object given file name</i>
-------------	--

Description

Write out the data object given file name

Usage

```
export_data(data_to_export, path_to_export)
```

Arguments

data_to_export Object containing data we want to export
path_to_export Path of the CSV file, we want to export to

Value

Character vector indicating status of the write

Examples

```
## Not run:  
data_to_export <- token_stats_hist_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")  
path_to_export <- "~/Desktop/uniswappeR_export.csv"  
export_data(data_to_export, path_to_export)  
  
## End(Not run)
```

export_plot	<i>Write out the analysis plots</i>
-------------	-------------------------------------

Description

Write out the analysis plots

Usage

```
export_plot(plot_to_export, path_to_export, width = 7, height = 7)
```

Arguments

plot_to_export Object containing plot we want to export
path_to_export Path of the .png file, we want to export to
width Width of plot in inches
height Height of plot in inches

Value

Character vector of the status of the write

Examples

```
## Not run:
plot_to_export <- vis_uniswap_stats_hist_v2()
path_to_export <- "~/Desktop/uniswapR_plot_export.png"
export_plot(plot_to_export, path_to_export)

## End(Not run)
```

factory_stats_v2 *Get UniswapV2 Factory Stats*

Description

Get UniswapV2 Factory Stats

Usage

```
factory_stats_v2()
```

Value

List representing data on the UniswapV2 Factory contract

Examples

```
## Not run:
factory_stats_v2()

## End(Not run)
```

factory_stats_v3 *Get UniswapV3 Factory Stats*

Description

Get UniswapV3 Factory Stats

Usage

```
factory_stats_v3()
```

Value

List representing data on the UniswapV3 Factory contract

Examples

```
## Not run:
factory_stats_v3()

## End(Not run)
```

get_infura_node	<i>Gets the Infura Node</i>
-----------------	-----------------------------

Description

Gets the Infura Node

Usage

```
get_infura_node()
```

Value

Character vector, representing your Infura Node, if set

Examples

```
q
## Not run:
get_infura_node()

## End(Not run)
```

liquidity_range_all_v3	<i>Forecast Price for liquidity range for a pair's tokens UniswapV3</i>
------------------------	---

Description

Forecast Price for liquidity range for a pair's tokens UniswapV3

Usage

```
liquidity_range_all_v3(
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801",
  days = 30,
  cap = 10,
  sims = 1000
)
```

Arguments

pair_address	Pair's Address
days	How long in future to forecast
cap	Max Percentage Increase that can occur in a day, default capped to 10%
sims	Number of simulations

Value

Data frame representing Forecast Price for a pair's tokens UniswapV3

Examples

```
liquidity_range_all_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801",
  days = 30, cap = 10, sims = 1000)
```

liquidity_range_v3	<i>Get a suggested range for liquidity</i>
--------------------	--

Description

Get a suggested range for liquidity

Usage

```
liquidity_range_v3(pair_address, ...)
```

Arguments

pair_address	The address of the pair to analyze
...	Additional arguments passed to the liquidity_range_all_v3 function

Value

Character vector representing a Suggestion for liquidity Range

Examples

```
liquidity_range_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

liquidity_range_visualization
Get a visualization liquidity range estimates

Description

Get a visualization liquidity range estimates

Usage

```
liquidity_range_visualization(pair_address, ...)
```

Arguments

pair_address The address of the pair to analyze
... Additional arguments passed to the liquidity_range_all_v3 function

Value

Visualization on the liquidity range for the given pair

Examples

```
liquidity_range_visualization("0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

pairs_all_v2 *Get UniswapV2 All Pairs*

Description

Get UniswapV2 All Pairs

Usage

```
pairs_all_v2()
```

Value

Data frame representing All Pair Data

Examples

```
## Not run:  
pairs_all_v2()  
  
## End(Not run)
```

pairs_all_v3	<i>Get UniswapV3 All Pairs</i>
--------------	--------------------------------

Description

Get UniswapV3 All Pairs

Usage

```
pairs_all_v3()
```

Value

Data frame representing All Pair Data

Examples

```
## Not run:  
pairs_all_v3()  
  
## End(Not run)
```

pair_burn_txs_v2	<i>Get UniswapV2 Burn Transactions in a pair</i>
------------------	--

Description

Get UniswapV2 Burn Transactions in a pair

Usage

```
pair_burn_txs_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")
```

Arguments

pair_address Pair's Address

Value

Data frame representing Burn Transactions in a pair

Examples

```
## Not run:  
pair_burn_txs_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

pair_burn_txs_v3 *Get UniswapV3 Burn Transactions in a pair*

Description

Get UniswapV3 Burn Transactions in a pair

Usage

```
pair_burn_txs_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

Arguments

pair_address Pair's Address

Value

Data frame representing Burn Transactions in a pair

Examples

```
## Not run:  
pair_burn_txs_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

pair_liq_positions_hist_v2
Get Uniswap2 Historical Liquidity Positions in a pair

Description

Get Uniswap2 Historical Liquidity Positions in a pair

Usage

```
pair_liq_positions_hist_v2(  
  pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Historical Liquidity Positions in a pair

Examples

```
## Not run:  
pair_liq_positions_hist_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

pair_liq_positions_hist_v3

Get Uniswap3 Historical Liquidity Positions in a pair

Description

Get Uniswap3 Historical Liquidity Positions in a pair

Usage

```
pair_liq_positions_hist_v3(  
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Historical Liquidity Positions in a pair

Examples

```
## Not run:  
pair_liq_positions_hist_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

pair_liq_positions_v2 *Get UniswapV2 Current Liquidity Positions in a pair*

Description

Get UniswapV2 Current Liquidity Positions in a pair

Usage

```
pair_liq_positions_v2(  
  pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Current Liquidity Positions in a pair

Examples

```
## Not run:  
pair_liq_positions_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

pair_liq_positions_v3 *Get UniswapV3 Current Liquidity Positions in a pair*

Description

Get UniswapV3 Current Liquidity Positions in a pair

Usage

```
pair_liq_positions_v3(  
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Current Liquidity Positions in a pair

Examples

```
## Not run:  
pair_liq_positions_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

pair_mint_txs_v2 *Get UniswapV2 Mint Transactions in a pair*

Description

Get UniswapV2 Mint Transactions in a pair

Usage

```
pair_mint_txs_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")
```

Arguments

pair_address Pair's Address

Value

Data frame representing Mint Transactions in a pair

Examples

```
## Not run:  
pair_mint_txs_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

pair_mint_txs_v3 *Get UniswapV3 Mint Transactions in a pair*

Description

Get UniswapV3 Mint Transactions in a pair

Usage

```
pair_mint_txs_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

Arguments

pair_address Pair's Address

Value

Data frame representing Mint Transactions in a pair

Examples

```
## Not run:  
pair_mint_txs_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

```
pair_stats_hist_daily_v2
```

Get UniswapV2 Daily Pair Historical Stats

Description

Get UniswapV2 Daily Pair Historical Stats

Usage

```
pair_stats_hist_daily_v2(  
  pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Daily Historical Data on a particular Pair

Examples

```
## Not run:  
pair_stats_hist_daily_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

```
pair_stats_hist_daily_v3
```

Get UniswapV3 Daily Pair Historical Stats

Description

Get UniswapV3 Daily Pair Historical Stats

Usage

```
pair_stats_hist_daily_v3(  
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Daily Historical Data on a particular Pair

Examples

```
## Not run:  
pair_stats_hist_daily_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

pair_stats_hist_hourly_v2
Get UniswapV2 Hourly Pair Historical Stats

Description

Get UniswapV2 Hourly Pair Historical Stats

Usage

```
pair_stats_hist_hourly_v2(  
  pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Hourly Historical Data on a particular Pair

Examples

```
## Not run:  
pair_stats_hist_hourly_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

pair_stats_hist_hourly_v3

Get UniswapV3 Hourly Pair Historical Stats

Description

Get UniswapV3 Hourly Pair Historical Stats

Usage

```
pair_stats_hist_hourly_v3(  
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801"  
)
```

Arguments

pair_address Pair's Address

Value

Data frame representing Hourly Historical Data on a particular Pair

Examples

```
## Not run:  
pair_stats_hist_hourly_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

pair_stats_v2

Get UniswapV2 Pair Stats

Description

Get UniswapV2 Pair Stats

Usage

```
pair_stats_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")
```

Arguments

pair_address Pair's Address

Value

List data on a particular Pair

Examples

```
## Not run:
pair_stats_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")

## End(Not run)
```

pair_stats_v3 *Get UniswapV3 Pair Stats*

Description

Get UniswapV3 Pair Stats

Usage

```
pair_stats_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

Arguments

pair_address Pair's Address

Value

List data on a particular Pair

Examples

```
## Not run:
pair_stats_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")

## End(Not run)
```

pair_swap_txs_v2 *Get UniswapV2 Swap Transactions in a pair*

Description

Get UniswapV2 Swap Transactions in a pair

Usage

```
pair_swap_txs_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")
```

Arguments

pair_address Pair's Address

Value

Data frame representing Swap Transactions in a pair

Examples

```
## Not run:  
pair_swap_txs_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")  
  
## End(Not run)
```

pair_swap_txs_v3 *Get UniswapV3 Swap Transactions in a pair*

Description

Get UniswapV3 Swap Transactions in a pair

Usage

```
pair_swap_txs_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

Arguments

pair_address Pair's Address

Value

Data frame representing Swap Transactions in a pair

Examples

```
## Not run:  
pair_swap_txs_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

py_int

Python builtin functions

Description

Python builtin functions

Usage

py_int(x)

Arguments

x The value to pass through the python int function

Value

An R integer converted from a Python integer

r_num

python object to R

Description

python object to R

Usage

r_num(val)

Arguments

val Numeric converter to export

Value

An R numeric vector converted from a Python numeric vector

set_infura_node	<i>Sets the Infura Node</i>
-----------------	-----------------------------

Description

Sets the Infura Node

Usage

```
set_infura_node(infura_node)
```

Arguments

infura_node	The Infura Node
-------------	-----------------

Value

A boolean TRUE if the INFURA NODE was successfully set

Examples

```
## Not run:
set_infura_node("https://mainnet.infura.io/v3/XXXXXXXXXXXXXXXXXXXX")

## End(Not run)
```

swaps	<i>Get all the swaps data for a given address or addresses</i>
-------	--

Description

Get all the swaps data for a given address or addresses

Usage

```
swaps(address)
```

Arguments

address	A wallet address (or vector of addresses) for the account owner's account
---------	---

Value

dataframe on the swaps for the given account owner

Examples

```
addresses <- c("0xb1b117a45aD71d408eb55475FC3A65454edCc94A",  
              "0x41D2a18E1DdACdAbFDdADB62e9AEE67c63070b76",  
              "0x0De20c4bDBE0d0EEFFd2956Be4c148CA86C6cC45")
```

```
swaps(addresses)
```

swap_performance *Get a visualization on the swap performance*

Description

Get a visualization on the swap performance

Usage

```
swap_performance(swap_data)
```

Arguments

swap_data The data on swaps as generated by the swaps() function

Value

ggplot2 object of visualizations on the swap performance for the given account owner

Examples

```
addresses <- c("0xb1b117a45aD71d408eb55475FC3A65454edCc94A",  
              "0x41D2a18E1DdACdAbFDdADB62e9AEE67c63070b76",  
              "0x0De20c4bDBE0d0EEFFd2956Be4c148CA86C6cC45")
```

```
swap_data <- swaps(addresses)
```

swap_statistics *Get statistics on the swaps data for a given address or addresses*

Description

Get statistics on the swaps data for a given address or addresses

Usage

```
swap_statistics(swap_data, aggregate_addresses = TRUE)
```

Arguments

swap_data The data on swaps as generated by the swaps() function
aggregate_addresses If TRUE, aggregate the addresses passed in

Value

dataframe of statistics on the swaps for the given account owner

Examples

```
addresses <- c("0xb1b117a45aD71d408eb55475FC3A65454edCc94A",  
"0x41D2a18E1DdACdAbFDdADB62e9AEE67c63070b76",  
"0x0De20c4bDBE0d0EEFFd2956Be4c148CA86C6cC45")  
  
swap_data <- swaps(addresses)  
swap_statistics(swap_data)
```

swap_visualizations *Get a visualization on the swap data*

Description

Get a visualization on the swap data

Usage

```
swap_visualizations(swap_data)
```

Arguments

swap_data The data on swaps as generated by the swaps() function

Value

ggplot2 object of visualizations on the swaps for the given account owner

Examples

```
addresses <- c("0xb1b117a45aD71d408eb55475FC3A65454edCc94A",  
"0x41D2a18E1DdACdAbFDdADB62e9AEE67c63070b76",  
"0x0De20c4bDBE0d0EEFFd2956Be4c148CA86C6cC45")  
  
swap_data <- swaps(addresses)  
swap_visualizations(swap_data)
```

token_pair_map_v2 *Get UniswapV2 Token's associated pairs*

Description

Get UniswapV2 Token's associated pairs

Usage

```
token_pair_map_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

Arguments

token_address Token's Address

Value

Data frame representing associated Pairs of a particular Token

Examples

```
## Not run:  
token_pair_map_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")  
  
## End(Not run)
```

token_pair_map_v3 *Get UniswapV3 Token's associated pairs*

Description

Get UniswapV3 Token's associated pairs

Usage

```
token_pair_map_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

Arguments

token_address Token's Address

Value

Data frame representing associated Pairs of a particular Token

Examples

```
## Not run:  
token_pair_map_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")  
  
## End(Not run)
```

token_stats_hist_v2 *Get UniswapV2 Token Historical Stats*

Description

Get UniswapV2 Token Historical Stats

Usage

```
token_stats_hist_v2(  
  token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984"  
)
```

Arguments

token_address Token's Address

Value

Data frame representing historical Data on a particular Token

Examples

```
## Not run:  
token_stats_hist_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")  
  
## End(Not run)
```

token_stats_hist_v3 *Get UniswapV3 Token Historical Stats*

Description

Get UniswapV3 Token Historical Stats

Usage

```
token_stats_hist_v3(  
  token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984"  
)
```

Arguments

token_address Token's Address

Value

Data frame representing historical Data on a particular Token

Examples

```
## Not run:  
token_stats_hist_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")  
  
## End(Not run)
```

token_stats_v2	<i>Get UniswapV2 Token Stats</i>
----------------	----------------------------------

Description

Get UniswapV2 Token Stats

Usage

```
token_stats_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

Arguments

token_address Token's Address

Value

List representing data on a particular Token

Examples

```
## Not run:  
token_stats_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")  
  
## End(Not run)
```

token_stats_v3	<i>Get UniswapV3 Token Stats</i>
----------------	----------------------------------

Description

Get UniswapV3 Token Stats

Usage

```
token_stats_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

Arguments

token_address Token's Address

Value

List representing data on a particular Token

Examples

```
## Not run:
token_stats_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")

## End(Not run)
```

trade_eth.to.tok_eth.fix	<i>Swap ETH for a Token, Receive Tokens for specified ETH amount</i>
--------------------------	--

Description

Swap ETH for a Token, Receive Tokens for specified ETH amount

Usage

```
trade_eth.to.tok_eth.fix(t_a, t_d, e_q, u_w)
```

Arguments

t_a	Token Address
t_d	Token Decimals
e_q	Ethereum Qty.
u_w	Uniswap Session

Value

Character vector representing the Transaction Hash

`trade_eth.to.tok_tok.fix`*Swap ETH for a Token, Buy specified fixed Token Amount*

Description

Swap ETH for a Token, Buy specified fixed Token Amount

Usage`trade_eth.to.tok_tok.fix(t_a, t_d, t_q, u_w)`**Arguments**

<code>t_a</code>	Token Address
<code>t_d</code>	Token Decimals
<code>t_q</code>	Token Qty.
<code>u_w</code>	Uniswap Session

Value

Character vector representing the Transaction Hash

`trade_tok.to.eth_eth.fix`*Swap Token for ETH, Swap tokens to receive specified ETH amount*

Description

Swap Token for ETH, Swap tokens to receive specified ETH amount

Usage`trade_tok.to.eth_eth.fix(t_a, t_d, e_q, u_w)`**Arguments**

<code>t_a</code>	Token Address
<code>t_d</code>	Token Decimals
<code>e_q</code>	ETH Qty.
<code>u_w</code>	Uniswap Session

Value

Character vector representing the Transaction Hash

```
trade_tok.to.eth_tok.fix
```

Swap Token for ETH, Receive ETH after swapping specified token amount

Description

Swap Token for ETH, Receive ETH after swapping specified token amount

Usage

```
trade_tok.to.eth_tok.fix(t_a, t_d, t_q, u_w)
```

Arguments

t_a	Token Address
t_d	Token Decimals
t_q	Token Qty.
u_w	Uniswap Session

Value

Character vector representing the Transaction Hash

```
trade_tok1.to.tok2_tok1.fix
```

Swap Token1 for Token2, Receive Token2 for specified Token1 Amount

Description

Swap Token1 for Token2, Receive Token2 for specified Token1 Amount

Usage

```
trade_tok1.to.tok2_tok1.fix(t1_a, t1_d, t2_a, t2_d, t1_q, u_w)
```

Arguments

t1_a	Token 1 Address
t1_d	Token 1 Decimals
t2_a	Token 2 Address
t2_d	Token 2 Decimals
t1_q	Token 1 Qty.
u_w	Uniswap Session

Value

Character vector representing the Transaction Hash

```
trade_tok1.to.tok2_tok2.fix
      Swap Token1 for Token2, Receive specified Token2 Amount
```

Description

Swap Token1 for Token2, Receive specified Token2 Amount

Usage

```
trade_tok1.to.tok2_tok2.fix(t1_a, t1_d, t2_a, t2_d, t2_q, u_w)
```

Arguments

t1_a	Token 1 Address
t1_d	Token 1 Decimals
t2_a	Token 2 Address
t2_d	Token 2 Decimals
t2_q	Token 2 Qty.
u_w	Uniswap Session

Value

Character vector representing the Transaction Hash

```
uniswap_session      Start the python session
```

Description

Start the python session

Usage

```
uniswap_session(
  node = get_infura_node(),
  user_add = NULL,
  pvt_key = NULL,
  version = 2
)
```

Arguments

node	The Infura Node
user_add	User Address
pvt_key	Private Key of the user Address
version	Choose UniswapV2 or UniswapV3 by supplying the version as 2 or 3 respectively.

Value

The python session uniswap endpoint

uniswap_stats_hist_v2 *Get UniswapV2 Historical Stats*

Description

Get UniswapV2 Historical Stats

Usage

```
uniswap_stats_hist_v2()
```

Value

data frame representing historical Data on the Uniswap Platform

Examples

```
## Not run:  
uniswap_stats_hist_v2()  
  
## End(Not run)
```

uniswap_stats_hist_v3 *Get UniswapV3 Historical Stats*

Description

Get UniswapV3 Historical Stats

Usage

```
uniswap_stats_hist_v3()
```

Value

data frame representing historical Data on the Uniswap Platform

Examples

```
## Not run:  
uniswap_stats_hist_v3()  
  
## End(Not run)
```

user_burns_v2	<i>Get UniswapV2 User Burn Txs</i>
---------------	------------------------------------

Description

Get UniswapV2 User Burn Txs

Usage

```
user_burns_v2(user_address = "0xcd8aa390e6eabbd2169b3580c1f7ce854675fd03")
```

Arguments

user_address User's Address

Value

Data frame representing User Burn Txs

Examples

```
## Not run:  
user_burns_v2(user_address = "0xcd8aa390e6eabbd2169b3580c1f7ce854675fd03")  
  
## End(Not run)
```

user_burns_v3 *Get UniswapV3 User Burn Txs*

Description

Get UniswapV3 User Burn Txs

Usage

```
user_burns_v3(user_address = "0x431B5A84aCC1297Eda88259f300262F1bc3A74f3")
```

Arguments

user_address User's Address

Value

Data frame representing User Burn Txs

Examples

```
## Not run:  
user_burns_v3(user_address = "0x431B5A84aCC1297Eda88259f300262F1bc3A74f3")  
  
## End(Not run)
```

user_hist_lps_v2 *Get UniswapV2 Historical User Liquidity Positions*

Description

Get UniswapV2 Historical User Liquidity Positions

Usage

```
user_hist_lps_v2(user_address = "0x2502f65d77ca13f183850b5f9272270454094a08")
```

Arguments

user_address User's Address

Value

Data frame representing Historical User Liquidity Positions

Examples

```
## Not run:
user_hist_lps_v2(user_address = "0x2502f65d77ca13f183850b5f9272270454094a08")

## End(Not run)
```

user_hist_lps_v3 *Get UniswapV3 Historical User Liquidity Positions*

Description

Get UniswapV3 Historical User Liquidity Positions

Usage

```
user_hist_lps_v3(user_address = "0xF1c206dd83ee2b8E6Ea675Cf827C93c58486B972")
```

Arguments

user_address User's Address

Value

Data frame representing Historical User Liquidity Positions

Examples

```
## Not run:
user_hist_lps_v3(user_address = "0xF1c206dd83ee2b8E6Ea675Cf827C93c58486B972")

## End(Not run)
```

user_lps_v2 *Get UniswapV2 User Liquidity Positions*

Description

Get UniswapV2 User Liquidity Positions

Usage

```
user_lps_v2(user_address = "0x2502f65d77ca13f183850b5f9272270454094a08")
```

Arguments

user_address User's Address

Value

Data frame representing User Liquidity Positions

Examples

```
## Not run:  
user_lps_v2(user_address = "0x2502f65d77ca13f183850b5f9272270454094a08")  
  
## End(Not run)
```

user_lps_v3

Get UniswapV3 User Liquidity Positions

Description

Get UniswapV3 User Liquidity Positions

Usage

```
user_lps_v3(user_address = "0xF1c206dd83ee2b8E6Ea675Cf827C93c58486B972")
```

Arguments

user_address User's Address

Value

Data frame representing User Liquidity Positions

Examples

```
## Not run:  
user_lps_v3(user_address = "0xF1c206dd83ee2b8E6Ea675Cf827C93c58486B972")  
  
## End(Not run)
```

user_mints_v2	<i>Get UniswapV2 User Mint Tx</i>
---------------	-----------------------------------

Description

Get UniswapV2 User Mint Tx

Usage

```
user_mints_v2(user_address = "0xcd8aa390e6eabbd2169b3580c1f7ce854675fd03")
```

Arguments

user_address User's Address

Value

Data frame representing User Mint Tx

Examples

```
## Not run:  
user_mints_v2(user_address = "0xcd8aa390e6eabbd2169b3580c1f7ce854675fd03")  
  
## End(Not run)
```

user_mints_v3	<i>Get UniswapV3 User Mint Tx</i>
---------------	-----------------------------------

Description

Get UniswapV3 User Mint Tx

Usage

```
user_mints_v3(user_address = "0x431B5A84aCC1297Eda88259f300262F1bc3A74f3")
```

Arguments

user_address User's Address

Value

Data frame representing User Mint Tx

Examples

```
## Not run:
user_mints_v3(user_address = "0x431B5A84aCC1297Eda88259f300262F1bc3A74f3")

## End(Not run)
```

user_swaps_v2	<i>Get UniswapV2 User Swap Tx</i>
---------------	-----------------------------------

Description

Get UniswapV2 User Swap Tx

Usage

```
user_swaps_v2(user_address = "0xcd8aa390e6eabbd2169b3580c1f7ce854675fd03")
```

Arguments

user_address User's Address

Value

Data frame representing User Swap Tx

Examples

```
## Not run:
user_swaps_v2(user_address = "0xcd8aa390e6eabbd2169b3580c1f7ce854675fd03")

## End(Not run)
```

user_swaps_v3	<i>Get UniswapV3 User Swap Tx</i>
---------------	-----------------------------------

Description

Get UniswapV3 User Swap Tx

Usage

```
user_swaps_v3(user_address = "0x431B5A84aCC1297Eda88259f300262F1bc3A74f3")
```

Arguments

user_address User's Address

Value

Data frame representing User Swap Tx

Examples

```
## Not run:  
user_swaps_v3(user_address = "0x431B5A84aCC1297Eda88259f300262F1bc3A74f3")  
  
## End(Not run)
```

vis_pair_liq_positions_v2

Visualise Liquidity Positions spread in a given pair

Description

Visualise Liquidity Positions spread in a given pair

Usage

```
vis_pair_liq_positions_v2(  
  pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee"  
)
```

Arguments

pair_address Pair's Address

Value

ggplot2 Plot of Liquidity Positions spread in a given pair

Examples

```
vis_pair_liq_positions_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")
```

```
vis_pair_liq_positions_v3
```

Visualise Liquidity Positions spread in a given pair

Description

Visualise Liquidity Positions spread in a given pair

Usage

```
vis_pair_liq_positions_v3(  
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801"  
)
```

Arguments

pair_address Pair's Address

Value

ggplot2 Plot of Liquidity Positions spread in a given pair

Examples

```
## Not run:  
vis_pair_liq_positions_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")  
  
## End(Not run)
```

```
vis_pair_stats_hist_daily_v2
```

Visualise various growth metrics of a given pair UniswapV2

Description

Visualise various growth metrics of a given pair UniswapV2

Usage

```
vis_pair_stats_hist_daily_v2(  
  pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee"  
)
```

Arguments

pair_address Pair's Address

Value

ggplot2 Plot of growth metrics of a given pair

Examples

```
vis_pair_stats_hist_daily_v2(pair_address = "0xf00e80f0de9aea0b33aa229a4014572777e422ee")
```

```
vis_pair_stats_hist_daily_v3
```

Visualise various growth metrics of a given pair UniswapV3

Description

Visualise various growth metrics of a given pair UniswapV3

Usage

```
vis_pair_stats_hist_daily_v3(  
  pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801"  
)
```

Arguments

pair_address Pair's Address

Value

ggplot2 Plot of growth metrics of a given pair

Examples

```
vis_pair_stats_hist_daily_v3(pair_address = "0x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801")
```

```
vis_token_pair_map_v2
```

Visualise Number of pairs the token is present UniswapV2

Description

Visualise Number of pairs the token is present UniswapV2

Usage

```
vis_token_pair_map_v2(  
  token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984"  
)
```

Arguments

token_address Token's Address

Value

ggplot2 Plot of Number of pairs the token is present

Examples

```
vis_token_pair_map_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

vis_token_pair_map_v3 *Visualise Number of pairs the token is present UniswapV3*

Description

Visualise Number of pairs the token is present UniswapV3

Usage

```
vis_token_pair_map_v3(  
  token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984"  
)
```

Arguments

token_address Token's Address

Value

ggplot2 Plot of Number of pairs the token is present

Examples

```
vis_token_pair_map_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

`vis_token_stats_hist_v2`*Visualise various growth metrics of a given token in UniswapV2*

Description

Visualise various growth metrics of a given token in UniswapV2

Usage

```
vis_token_stats_hist_v2(  
  token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984"  
)
```

Arguments

token_address Token's Address

Value

ggplot2 Plot of growth metrics of a given token

Examples

```
vis_token_stats_hist_v2(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

`vis_token_stats_hist_v3`*Visualise various growth metrics of a given token in UniswapV3*

Description

Visualise various growth metrics of a given token in UniswapV3

Usage

```
vis_token_stats_hist_v3(  
  token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984"  
)
```

Arguments

token_address Token's Address

Value

ggplot2 Plot of growth metrics of a given token

Examples

```
vis_token_stats_hist_v3(token_address = "0x1f9840a85d5af5bf1d1762f925bdaddc4201f984")
```

```
vis_uniswap_stats_hist_v2
```

Visualise various growth metrics of the UniswapV2 Platform

Description

Visualise various growth metrics of the UniswapV2 Platform

Usage

```
vis_uniswap_stats_hist_v2()
```

Value

ggplot2 plot of growth metrics of the UniswapV2 Platform

Examples

```
## Not run: vis_uniswap_stats_hist_v2()
```

```
vis_uniswap_stats_hist_v3
```

Visualise various growth metrics of the UniswapV3 Platform

Description

Visualise various growth metrics of the UniswapV3 Platform

Usage

```
vis_uniswap_stats_hist_v3()
```

Value

ggplot plot of growth metrics of the UniswapV3 Platform

Examples

```
## Not run: vis_uniswap_stats_hist_v3()
```

Index

check_eth.to.tok_eth.fix, 3
check_eth.to.tok_tok.fix, 4
check_eth_balance, 4
check_tok.to.eth_eth.fix, 5
check_tok.to.eth_tok.fix, 5
check_tok1.to.tok2_tok1.fix, 6
check_tok1.to.tok2_tok2.fix, 6
check_tok_balance, 7

export_data, 8
export_plot, 8

factory_stats_v2, 9
factory_stats_v3, 9

get_infura_node, 10

liquidity_range_all_v3, 10
liquidity_range_v3, 11
liquidity_range_visualization, 12

pair_burn_txs_v2, 13
pair_burn_txs_v3, 14
pair_liq_positions_hist_v2, 14
pair_liq_positions_hist_v3, 15
pair_liq_positions_v2, 15
pair_liq_positions_v3, 16
pair_mint_txs_v2, 17
pair_mint_txs_v3, 17
pair_stats_hist_daily_v2, 18
pair_stats_hist_daily_v3, 18
pair_stats_hist_hourly_v2, 19
pair_stats_hist_hourly_v3, 20
pair_stats_v2, 20
pair_stats_v3, 21
pair_swap_txs_v2, 21
pair_swap_txs_v3, 22
pairs_all_v2, 12
pairs_all_v3, 13
py_int, 23

r_num, 23

set_infura_node, 24
swap_performance, 25
swap_statistics, 25
swap_visualizations, 26
swaps, 24

token_pair_map_v2, 27
token_pair_map_v3, 27
token_stats_hist_v2, 28
token_stats_hist_v3, 28
token_stats_v2, 29
token_stats_v3, 30
trade_eth.to.tok_eth.fix, 30
trade_eth.to.tok_tok.fix, 31
trade_tok.to.eth_eth.fix, 31
trade_tok.to.eth_tok.fix, 32
trade_tok1.to.tok2_tok1.fix, 32
trade_tok1.to.tok2_tok2.fix, 33

uniswap_session, 33
uniswap_stats_hist_v2, 34
uniswap_stats_hist_v3, 34
user_burns_v2, 35
user_burns_v3, 36
user_hist_lps_v2, 36
user_hist_lps_v3, 37
user_lps_v2, 37
user_lps_v3, 38
user_mints_v2, 39
user_mints_v3, 39
user_swaps_v2, 40
user_swaps_v3, 40

vis_pair_liq_positions_v2, 41
vis_pair_liq_positions_v3, 42
vis_pair_stats_hist_daily_v2, 42
vis_pair_stats_hist_daily_v3, 43
vis_token_pair_map_v2, 43

[vis_token_pair_map_v3](#), [44](#)
[vis_token_stats_hist_v2](#), [45](#)
[vis_token_stats_hist_v3](#), [45](#)
[vis_uniswap_stats_hist_v2](#), [46](#)
[vis_uniswap_stats_hist_v3](#), [46](#)