

Fall 2018 Data Set

Pritam Dalal

Location and Access

- ▶ The data set that will be used for your final analysis project is now available.
- ▶ It is located on my Google Drive folder.

If you need access, please e-mail me or get a copy of the data from a friend who has access.

- ▶ The file is called `fall_2018_data.zip`.

Guide to The Data

`df_expirations` <- a guide to the 19 expirations included in the data set, and the execution-date associated with each expiration.

`df_underlyings` <- a small file containing data associated with the 50 underlyings that are included in the data set.

`df_trading_universe` <- all the options that ever show up in market-history.

`df_market_history` <- for each trade-date, the set of OTM options for each underlying, for the next regular expiration. Most of you will only interact with the entries with trade-date equals execution-date.

`df_option_history` <- contains all EOD prices for all options in `df_market_history`, until expiration.

Backtesting Parameters

1. For each expiration, you will trade options on execution date (about one month before expiration).
2. Trades will be placed utilizing knowledge that is available on or before execution date.
3. Nearly all position will be held until expiration, but some groups are looking at early unwind strategies.
4. When calculating the PNL of an option, you might as well calculate the naked option PNL as well as the delta hedged PNL, because you have all the data.

Data Wrangling Procedure (1 of 4)

How did I arrive at this particular data set?

1. Finding the Data Source (Delta Neutral)

- ▶ Google
- ▶ Comparing Delta Neutral vs Option Metrics

2. Choosing the set of underlyings.

- ▶ ETFs (non-volatility) - don't have to deal with earnings reports
- ▶ 100 highest volume (201701-201810)
- ▶ *material*: tutorials 3-5, exercise 2, tutorial 11
- ▶ *packages*: readr, dplyr, bizdays
- ▶ *functions*: read_csv, distinct, bizdays, group_by, summarize, arrange, mutate, write_csv

Data Wrangling Procedure (2 of 4)

3. Market History

- ▶ calculate implied-forward for every underlying, expiration, trade-date combination
- ▶ grab all OTM underlying, expiration, trade-date combinations
- ▶ remove low-information options
- ▶ recalculate greeks
- ▶ *material*: tutorials 3-5, tutorial 7, tutorial 9, exercise 3, tutorial 10, tutorial 11
- ▶ *packages*: readr, dplyr, fOptions, bizdays, purrr
- ▶ *functions*: read_csv, select, filter, mutate, arrange, inner_join, rename, top_n, pmap_dbl

Data Wrangling Procedure (3 of 4)

3. Option History

- ▶ for every option that shows up in the market-history, grab all subsequent prices until expiration
- ▶ calculate payoff on expiration date
- ▶ *material*: tutorials 3-5, tutorial 7, tutorial 9, exercise 3, tutorials 10-11
- ▶ *packages*: readr, dplyr, purrr
- ▶ *functions*: read_csv, select, filter, mutate, arrange, left_join, pmap_dbl

Data Wrangling Procedure (4 of 4)

4. Data Cleaning

- ▶ graphing skewness and prices to check for any obvious anomalies
- ▶ checking for missing data
- ▶ removing any underlyings that seem suspect (57 remaining, I rounded down to 50 for simplicity)
- ▶ *material*: tutorials 3-5, tutorial 7-9, exercise 3, tutorial 10, tutorial 11
- ▶ *packages*: readr, tidyr, dplyr, purrr
- ▶ *functions*: distinct, group_by, summarize, inner_join, left_join, pmap_dbl, crossing, ggplot, facet_wrap

5. **Depressing Fact** - very little actual *analysis* has occurred yet.

Data Integrity and Cleaning Exercises (1 of 2)

Note: it will be useful to use the join functions in `dplyr` to complete a lot of these exercise: `inner_join()`, `left_join()`.

1. Check that `df_trading_universe`, `df_market_history`, and `df_option_history` all have the same number of unique options represented.
2. Check that `df_trading_universe` and `df_market_history` have one row per option, per trade-date.
3. Check that every underlying has at least 4 options in `df_market_history` on each execution date.

Data Integrity and Cleaning Exercises (2 of 2)

4. Check that once an option shows up in `df_market_history` that it has entries in `df_option_history` for all subsequent days until expiration.
5. Make sure that in `df_option_history` on expiration day the bid and ask are both equal to the options payoff.
6. In `df_underlyings` write a function that converts the `aum` column from a character into a number.