Introduction to Tidy Finance

Book Reading

John Karuitha

2022-08-15

Download prices using the `tidyquant` package

I download the data for APPLE from 2010 to 2022.

```
prices <- tq_get("AAPL", get = "stock.prices",

       from = "2010-01-01",

       to = "2022-07-16")

head(prices)
```

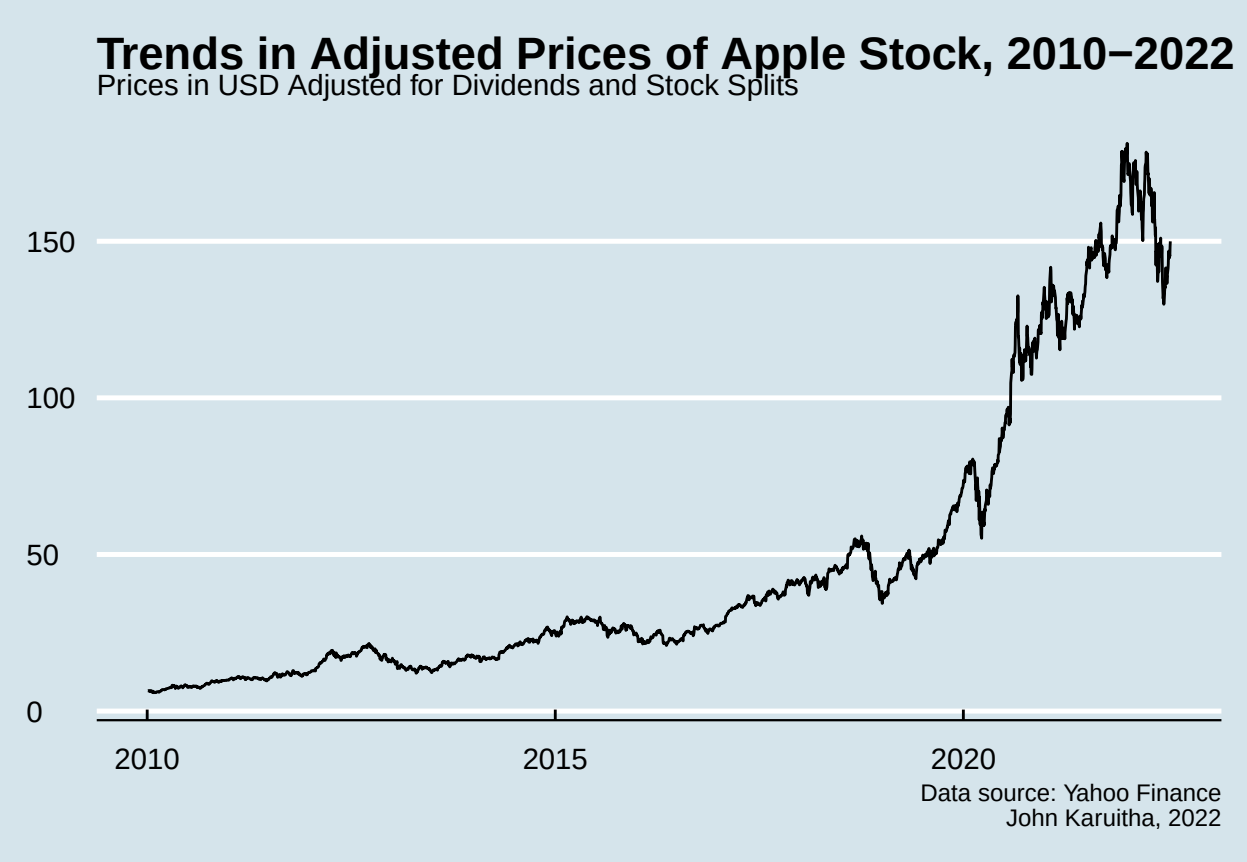
```
## # A tibble: 6 x 8
##   symbol date        open  high   low close    volume adjusted
##   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>    <dbl>
## 1 AAPL   2010-01-04  7.62  7.66  7.58  7.64 493729600     6.53
## 2 AAPL   2010-01-05  7.66  7.70  7.62  7.66 601904800     6.54
## 3 AAPL   2010-01-06  7.66  7.69  7.53  7.53 552160000     6.43
## 4 AAPL   2010-01-07  7.56  7.57  7.47  7.52 477131200     6.42
## 5 AAPL   2010-01-08  7.51  7.57  7.47  7.57 447610800     6.46
## 6 AAPL   2010-01-11  7.6   7.61  7.44  7.50 462229600     6.41
```

```
tail(prices)
```

```
## # A tibble: 6 x 8
##   symbol date        open  high   low close   volume adjusted
##   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 AAPL   2022-07-08  145.  148.  145   147. 64547800     147.
## 2 AAPL   2022-07-11  146.  147.  144.  145. 63141600     145.
## 3 AAPL   2022-07-12  146.  148.  145.  146. 77588800     146.
## 4 AAPL   2022-07-13  143.  146.  142.  145. 71185600     145.
## 5 AAPL   2022-07-14  144.  149.  143.  148. 78140700     148.
## 6 AAPL   2022-07-15  150.  151.  148.  150. 76259900     150.
```

Next I plot adjusted prices over time.

```
prices %>%

  ggplot(mapping = aes(x = date, y = adjusted)) +

  geom_line() +

  labs(title = "Trends in Adjusted Prices of Apple Stock, 2010-2022",

       subtitle = "Prices in USD Adjusted for Dividends and Stock Splits",

       x = NULL, y = NULL,

       caption = "Data source: Yahoo Finance
       John Karuitha, 2022")
```

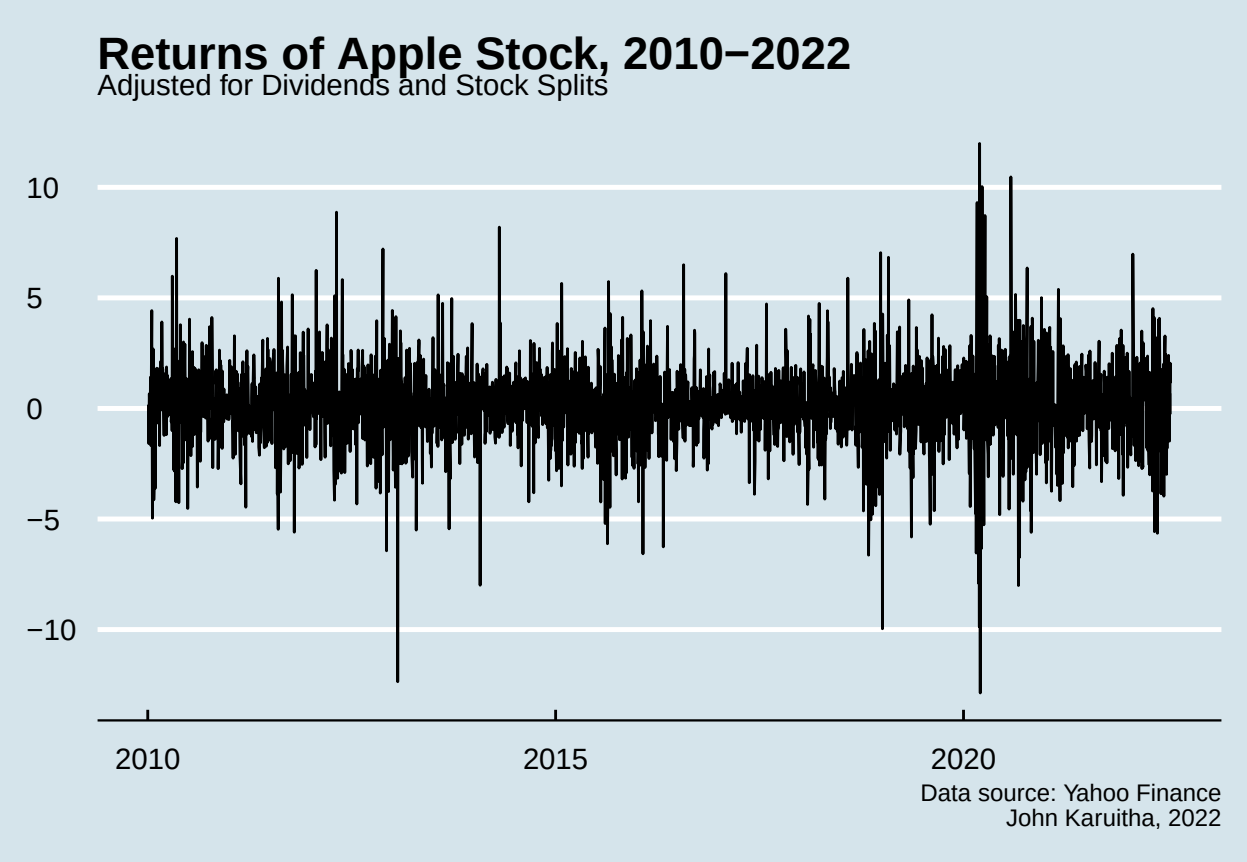


Next, I compute returns and then plot them over time.

```
returns <- prices %>%

  arrange(date) %>%

  mutate(returns = adjusted / lag(adjusted) - 1) %>%

  select(symbol, date, returns) %>%

  drop_na(returns)
```

Next I plot these returns

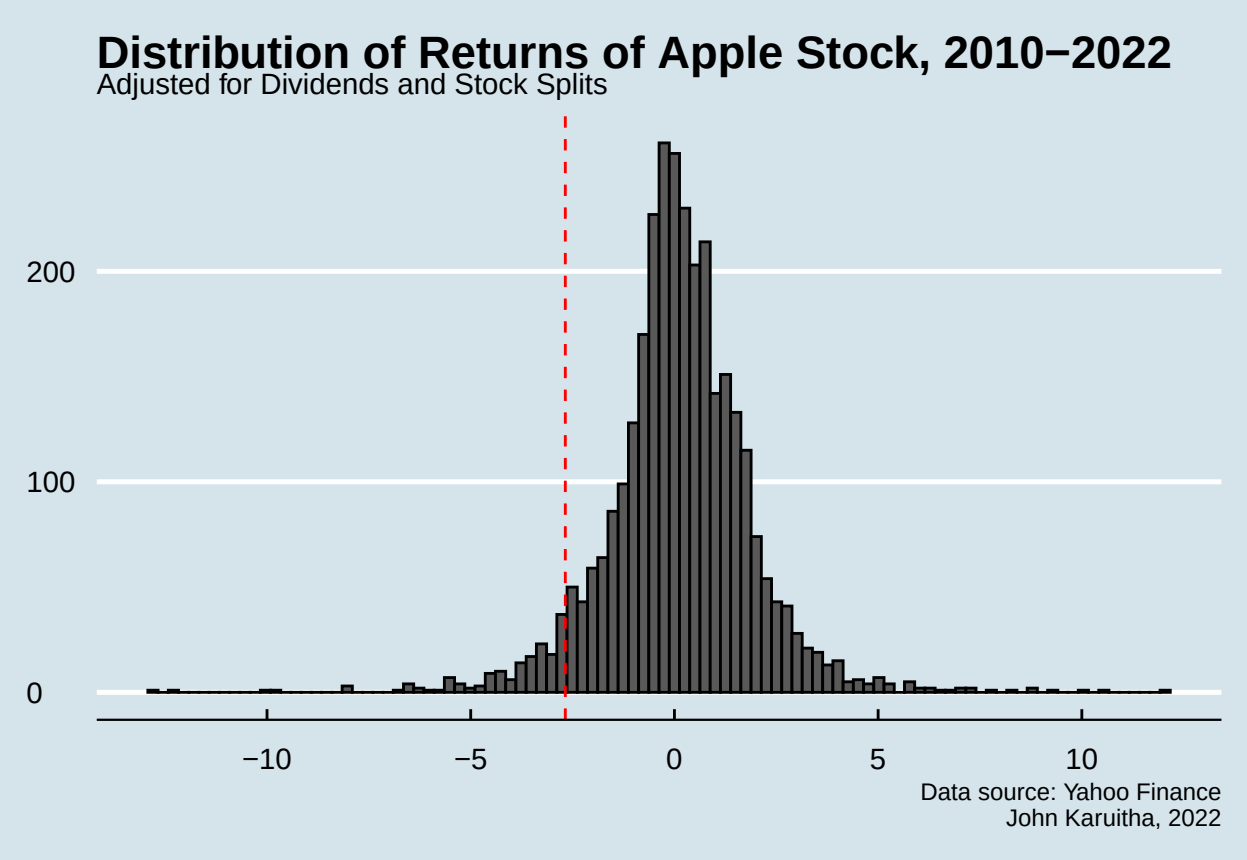
```
returns %>%

  ggplot(mapping = aes(x = date, y = returns * 100)) +

  geom_line() +

  labs(title = "Returns of Apple Stock, 2010-2022",

       subtitle = "Adjusted for Dividends and Stock Splits",

       x = NULL, y = NULL,

       caption = "Data source: Yahoo Finance
       John Karuitha, 2022")
```



A histogram will also do.

```
returns %>%

  ggplot(mapping = aes(x = returns * 100)) +

  geom_histogram(bins = 100, col = "black") +

  geom_vline(xintercept = quantile(returns$returns * 100, probs = 0.05),
```

```
            color = "red", linetype = "dashed") +
  labs(title = "Distribution of Returns of Apple Stock, 2010-2022",
       subtitle = "Adjusted for Dividends and Stock Splits",
       x = NULL, y = NULL,
       caption = "Data source: Yahoo Finance
       John Karuitha, 2022")
```



I summarise the data, getting the minimum, maximum, mean and median.

```
returns %>%
  mutate(ret = returns * 100) %>%
  summarise(across(ret, .fns = list(
    daily_mean = mean,
    daily_median = median,
    daily_min = min,
```

```
    daily_max = max


  )))
```

```
## # A tibble: 1 x 4
##   ret_daily_mean ret_daily_median ret_daily_min ret_daily_max
##            <dbl>            <dbl>         <dbl>         <dbl>
## 1          0.115           0.0979         -12.9          12.0
```

We also summarise the data by year.

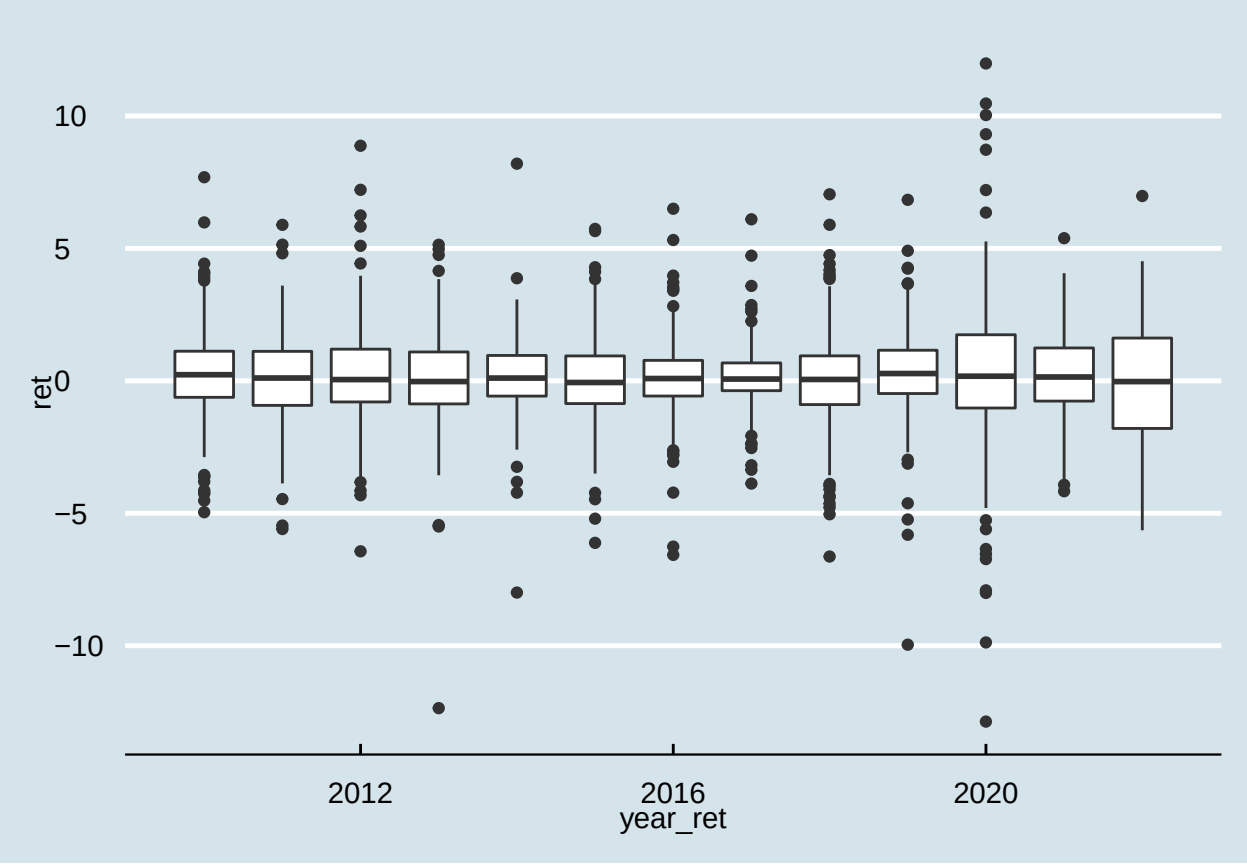
```
returns %>%

  mutate(ret = returns * 100) %>%

  group_by(year_ret = year(date)) %>%

  summarise(across(ret, .fns = list(

    daily_mean = mean,

    daily_median = median,

    daily_min = min,

    daily_max = max
  )))
```

```
## # A tibble: 13 x 5
##    year_ret ret_daily_mean ret_daily_median ret_daily_min ret_daily_max
##       <dbl>          <dbl>            <dbl>         <dbl>         <dbl>
##  1     2010        0.178             0.230         -4.96          7.69
##  2     2011        0.104             0.107         -5.59          5.89
##  3     2012        0.130             0.0473        -6.44          8.87
##  4     2013        0.0472           -0.0278       -12.4           5.14
##  5     2014        0.145             0.104         -7.99          8.20
##  6     2015        0.00199          -0.0630        -6.12          5.74
##  7     2016        0.0575            0.0873        -6.57          6.50
##  8     2017        0.164             0.0667        -3.88          6.10
##  9     2018       -0.00573           0.0525        -6.63          7.04
## 10     2019        0.266             0.275         -9.96          6.83
## 11     2020        0.281             0.175        -12.9          12.0
## 12     2021        0.131             0.144         -4.17          5.39
## 13     2022       -0.0986           -0.0282        -5.64          6.98
```

We could capture this yearly returns summary in a boxplot.

```
returns %>%

  mutate(ret = returns * 100,
```

```
        year_ret = year(date)) %>%

ggplot(mapping = aes(x = year_ret, y = ret,

                     group = year_ret)) +

geom_boxplot()
```



Taking it to the Next Level

In this section, we download the data for firms that make up the DOW JONES INDUSTRIAL INDEX (DJIA). The following code does the trick.

```
ticker <- tq_index("DOW")
head(ticker)
```

```
## # A tibble: 6 x 8
##   symbol company                    ident~1 sedol weight sector share~2 local~3
##   <chr>  <chr>                      <chr>   <chr>  <dbl> <chr>    <dbl> <chr>
## 1 UNH    UnitedHealth Group Incorpo~ 91324P~ 2917~ 0.106  Healt~ 5735859 USD
## 2 GS     Goldman Sachs Group Inc.    38141G~ 2407~ 0.0691 Finan~ 5735859 USD
## 3 HD     Home Depot Inc.             437076~ 2434~ 0.0615 Consu~ 5735859 USD
## 4 MSFT   Microsoft Corporation       594918~ 2588~ 0.0570 Infor~ 5735859 USD
## 5 MCD    McDonald's Corporation      580135~ 2550~ 0.0512 Consu~ 5735859 USD
```

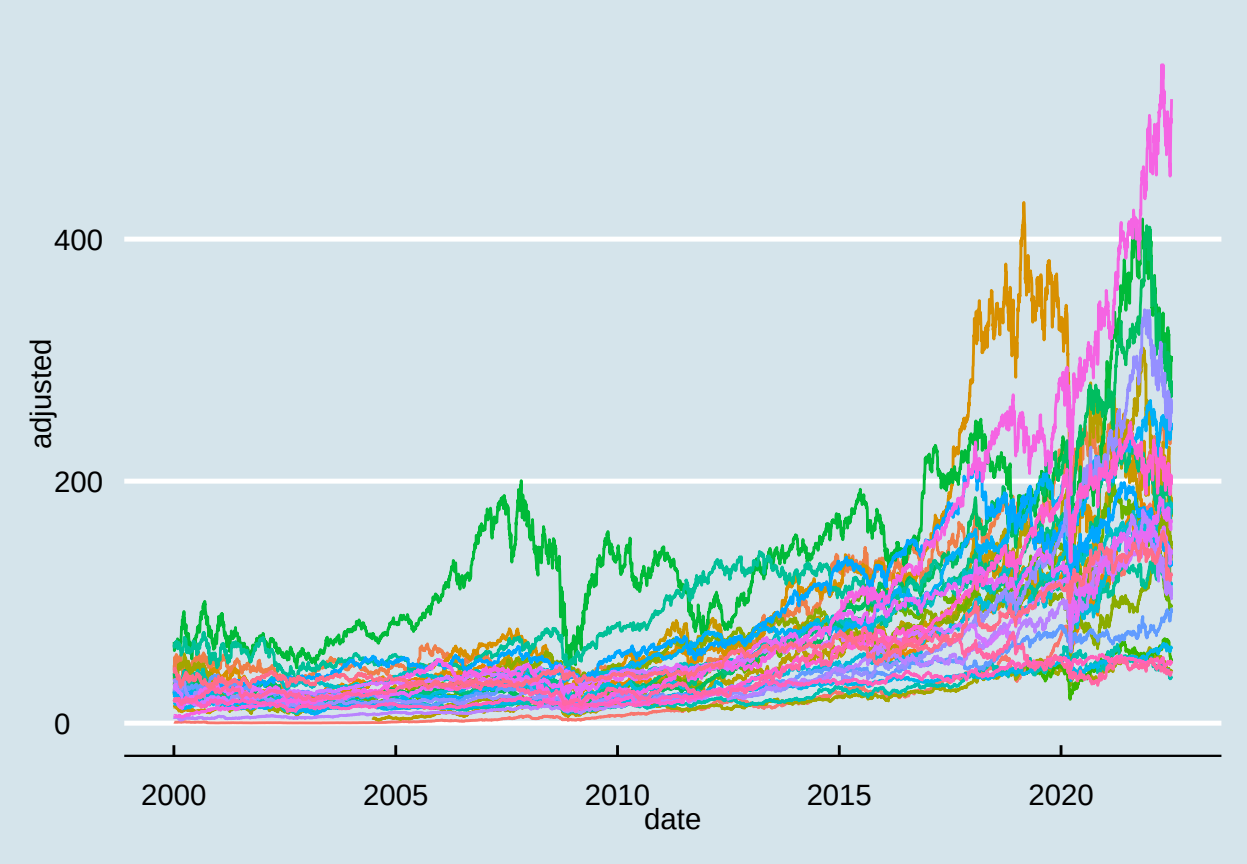
```
## 6 AMGN   Amgen Inc.                  031162~ 2023~ 0.0485 Healt~ 5735859 USD
## # ... with abbreviated variable names 1: identifier, 2: shares_held,
## #   3: local_currency
```

```
index_prices <- tq_get(ticker,
  get = "stock.prices",
  from = "2000-01-01",
  to = "2022-06-30"
)

head(index_prices)
```

```
## # A tibble: 6 x 15
##   symbol company    ident~1 sedol weight sector share~2 local~3 date        open
##   <chr>  <chr>      <chr>   <chr>  <dbl> <chr>    <dbl> <chr>   <date>     <dbl>
## 1 UNH    UnitedHea~ 91324P~ 2917~  0.106 Healt~ 5735859 USD     2000-01-03  6.64
## 2 UNH    UnitedHea~ 91324P~ 2917~  0.106 Healt~ 5735859 USD     2000-01-04  6.67
## 3 UNH    UnitedHea~ 91324P~ 2917~  0.106 Healt~ 5735859 USD     2000-01-05  6.64
## 4 UNH    UnitedHea~ 91324P~ 2917~  0.106 Healt~ 5735859 USD     2000-01-06  6.62
## 5 UNH    UnitedHea~ 91324P~ 2917~  0.106 Healt~ 5735859 USD     2000-01-07  7.19
## 6 UNH    UnitedHea~ 91324P~ 2917~  0.106 Healt~ 5735859 USD     2000-01-10  7.73
## # ... with 5 more variables: high <dbl>, low <dbl>, close <dbl>, volume <dbl>,
## #   adjusted <dbl>, and abbreviated variable names 1: identifier,
## #   2: shares_held, 3: local_currency
## # i Use `colnames()` to see all variable names
```

Now, let us plot the trends in the prices over time.

```
index_prices %>%

  ggplot(mapping = aes(y = adjusted, x = date,

                       color = symbol)) +

  geom_line(show.legend = FALSE)
```


400
200
0
adjusted
2000
2005
2010
2015
2020
date