

```
In [1]: from datetime import datetime
start_time = datetime.now()
print("===== START =====")
print(start_time.strftime("%Y-%m-%d %H:%M:%S"))

===== START =====
2026-02-02 15:10:36

In [2]: import pandas as pd
import numpy as np
import datetime as dt
import wrds
from dateutil.relativedelta import *

In [3]: conn = wrds.Connection()

WRDS recommends setting up a .pgpass file.
You can create this file yourself at any time with the create_pgpass_file() function.
Loading library list...
Done

In [4]: msf = conn.raw_sql("""select a.permno, a.date,
                                a.ret, a.shrout, a.prc
                                from crsp.dsf as a
                                where a.date >= '01/01/2007'
                                """, date_cols = ["date"])

msf.to_pickle("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/control_variables/wrds/raw/crspm_dsf_2007_to_latest_20260202.pkl")
print(msf.shape)
msf.head()

(34362936, 5)

Out[4]:      permno      date      ret  shrout  prc
0    10001  2007-01-03  0.000000  2959.0  11.10
1    10002  2007-01-03 -0.007445  11166.0  25.33
2    10025  2007-01-03 -0.079722   7886.0  49.06
3    10026  2007-01-03 -0.009179  18515.0  41.02
4    10028  2007-01-03   0.051181   4913.0   2.67

In [5]: msf = pd.read_pickle("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/control_variables/wrds/raw/crspm_dsf_2007_to_latest_20260202.pkl")
print(msf.shape)
msf.head()

(34362936, 5)

Out[5]:      permno      date      ret  shrout  prc
0    10001  2007-01-03  0.000000  2959.0  11.10
1    10002  2007-01-03 -0.007445  11166.0  25.33
2    10025  2007-01-03 -0.079722   7886.0  49.06
3    10026  2007-01-03 -0.009179  18515.0  41.02
4    10028  2007-01-03   0.051181   4913.0   2.67

In [6]: msf["permno"] = msf["permno"].astype(int)
msf = msf.sort_values(["permno", "date"]).reset_index(drop = True)
msf["size"] = msf["shrout"] * msf["prc"].abs()
print(msf.shape)
msf.head()

(34362936, 6)

Out[6]:      permno      date      ret  shrout  prc      size
0    10001  2007-01-03  0.000000  2959.0  11.100  32844.900
1    10001  2007-01-04   0.023423  2959.0  11.360  33614.240
2    10001  2007-01-05 -0.009683  2959.0  11.250  33288.750
3    10001  2007-01-08   0.008444  2959.0 -11.345  33569.855
4    10001  2007-01-09 -0.009255  2959.0  11.240  33259.160

In [7]: msf = msf[msf["size"].notna()]
print(msf.shape)
msf.head()

(34065304, 6)

Out[7]:      permno      date      ret  shrout  prc      size
0    10001  2007-01-03  0.000000  2959.0  11.100  32844.900
1    10001  2007-01-04   0.023423  2959.0  11.360  33614.240
2    10001  2007-01-05 -0.009683  2959.0  11.250  33288.750
3    10001  2007-01-08   0.008444  2959.0 -11.345  33569.855
4    10001  2007-01-09 -0.009255  2959.0  11.240  33259.160

In [8]: # compute decile
msf = msf.sort_values(["date"])
msf["decile"] = 1 + msf.groupby("date")["size"].transform(lambda x: pd.qcut(x, 10, labels = False))
msf = msf.reset_index(drop = True)

print(msf.shape)
msf.head()

(34065304, 7)

Out[8]:      permno      date      ret  shrout  prc      size  decile
0    10001  2007-01-03  0.000000  2959.0  11.1000  32844.9000    1
1    84837  2007-01-03  0.000004   3300.0  24.0400  79332.0000    3
2    84834  2007-01-03   0.019608  39182.0   2.0800  81498.5600    3
3    84833  2007-01-03   0.105954  10268.0   5.7399  58937.2932    2
4    90375  2007-01-03   0.003268  17755.0   9.2100  163523.5500    4

In [9]: # compute size weighted returns

msf_groups = msf.sort_values(["decile", "date"])

# function to calculate value weighted return
def wavg(group, avg_name, weight_name):
    d = group[avg_name]
    w = group[weight_name]
    try:
        return (d * w).sum() / w.sum()
    except ZeroDivisionError:
        return np.nan

# value-weighted return
vwrets = msf_groups.groupby(["decile", "date"]).apply(wavg, "ret", "size").to_frame().reset_index().rename(columns = {0: "vwret"})

print(vwrets.shape)
vwrets.head()

(45300, 3)
/var/folders/nz/_7x_b77n3rxzf42sssqc5s3m0000gn/T/ipykernel_50771/2254846598.py:15: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.
  vwrets = msf_groups.groupby(["decile", "date"]).apply(wavg, "ret", "size").to_frame().reset_index().rename(columns = {0: "vwret"})

Out[9]:      decile      date      vwret
0      1  2007-01-03   0.003967
1      1  2007-01-04   0.003701
2      1  2007-01-05  -0.004095
3      1  2007-01-08   0.001353
4      1  2007-01-09   0.001032

In [10]: ret = msf[["permno", "date", "decile", "ret"]]

In [11]: ret.to_csv("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/control_variables/wrds/raw/ret_decile.csv")

In [12]: vwrets.to_csv("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/control_variables/wrds/raw/vwret_decile.csv")

In [13]: end_time = datetime.now()
print("===== END =====")
print(end_time.strftime("%Y-%m-%d %H:%M:%S"))
print("Elapsed seconds:", (end_time - start_time).total_seconds())

===== END =====
2026-02-02 15:24:58
Elapsed seconds: 862.236618
```