```
* SAS 3: Calculate the monthly value-weighted portfolio returns ******************;

data portfolio1;
     set My_lib.assignment2_data;
     eretadj_ME = (eretadj*100)*ME_lag1;
run;
proc sort data = portfolio1;
     by date p1 p2;
run;
proc means data = portfolio1 sum noprint;
     by date p1 p2;
     var eretadj_ME ME_lag1;
     output out = portfolio2 (drop = _FREQ_ _TYPE_) sum = / autoname;
run;
data portfolio2;
     set portfolio2;
     vw_pret = eretadj_ME_Sum/ME_lag1_Sum;
     keep date p1 p2 vw_pret;
run;
```

```
* Calculate the return difference
between the fifth (p2 = 5) and first (p2 = 1) BM sorted portfolios
within each Size sorted portfolio;
data portfolio3;
     set portfolio2(where = (p2 in (1,5)));
run;
proc sort data = portfolio3;
     by date p1 p2;
run;
proc transpose data = portfolio3 out = portfolio4;
     by date p1;
     id p2;
     var vw_pret;
run;
data portfolio4;
     set portfolio4;
     p2 = 51;
     vw_pret = _5 - _1;
     keep date p1 p2 vw_pret;
run;

* Append the two datasets;
data portfolio5;
     set portfolio2 portfolio4;
     year = year(date);
     month = month(date);
run;
proc sort data = portfolio5;
     by year month date p1 p2;
run;
```

```
* SAS4: Add FF-3 factors to the portfolio return data set **********************************;

* Load monthly factors data;
proc import out = factors
     datafile = "&my_directory\factors_monthly.csv"
     dbms = csv replace;
run;

* Be careful. The "date" column in the montly factor data set is NOT in a date format.
We will merge the factor data set with the portfolio return data set
using "year" and "month" variables as the key variables;
data factors;
     set factors;

     * convert factors from decimal to percent;
     mktrf = mktrf*100;
     smb = smb*100;
     hml = hml*100;
     keep year month mktrf smb hml;
run;

* Merge;
data portfolio6;
     merge portfolio5 (in = a) factors (in = b);
     by year month;
     if a;
run;
```

```
* SAS 5: Test if the BM5 portfolio has a higher expected return than BM1 portfolio
within each size group using time-series regressions ********************************;

proc sort data = portfolio6;
     by p1 p2 date;
run;

* To perform Newey-West standard error correction,
PROC MODEL is run specifying the GMM estimation method in the FIT statement.
KERNEL=(BART, L+1, 0) is also specified which requests the Bartlett kernel with a lag length of L.
The VARDEF(specify the denominator for computing variances and covariances)=n option is specified
to be consistent with the original Newey-West formula;

* Calculate the FF3 alpha of the long-short portfolio (p2=51);
proc model data = portfolio6 (where = (p2 = 51));
     by p1;
     exog mktrf hml smb;
     instruments _exog_;
     vw_pret = a + b1*mktrf + b2*hml + b3*smb;
     fit vw_pret / gmm kernel = (bart, 7, 0) vardef = n;
     ods output parameterestimates = table3;
     quit;
ods exclude none;
```

In python: OLS(...).fit(cov_type='HAC')

https://www.statsmodels.org/stable/generated/statsmodels.regression.linear_model.OLSResults.get_robustcov_results.html

```
cov_type = HAC
maxlag = 6
```