QAM_hw2

April 21, 2019

1

For Each bond, lag its market value for one period. And then sum all bonds' market value based on date, and this is the total market value the previous month.

Bond equal weighted return is the mean of all bonds by each date.

To get the bond value weighted return, we first calculated the weight of each bond every month. The weight is the marketd value of each bond divided by the total market value. At last, the value weighted return is the sum of the weight based on the market value of previous month multiplied each bond's return of this month.

```
In [1]: import pandas as pd
        import numpy as np
        from scipy import stats
        import math
        crsp_bonds = pd.read_csv('/Users/huanyu/Desktop/'
                                 'QuantitativeAssetManagement/hw2/crsp_bonds.csv')
        crsp_bonds['MCALDT'] = pd.to_datetime(crsp_bonds['MCALDT'])
        crsp_bonds['capital'] = crsp_bonds[['KYCRSPID','TMTOTOUT']]\
            .groupby('KYCRSPID').shift(1)
        crsp_bonds.dropna(subset=['capital'],inplace=True)
        output = crsp_bonds[['MCALDT','capital']].groupby('MCALDT').sum()
        crsp_bonds.replace(-99,np.nan,inplace=True)
        output['Year'] = output.index.year
        output['Month'] = output.index.month
        output['Bond_Ew_Ret'] = crsp_bonds[['MCALDT','TMRETNUA']]\
            .groupby('MCALDT').mean()
        output.rename(columns={'capital':'Bond_lag_MV'},inplace=True)
        output.reset_index(inplace=True)
        crsp_bonds = pd.merge(crsp_bonds,output[['MCALDT','Bond_lag_MV']],
                              how='left',on='MCALDT')
        crsp_bonds['weight'] = crsp_bonds['capital'] / crsp_bonds['Bond_lag_MV']
        output['Bond_VW_Ret'] = pd.DataFrame({'date':crsp_bonds['MCALDT'],
                                'x':crsp_bonds['weight'] * crsp_bonds['TMRETNUA']})\
                                .groupby('date').sum().values
        output.dropna(subset=['Bond_VW_Ret'],inplace=True)
        print(output)

          MCALDT  Bond_lag_MV  Year  Month  Bond_Ew_Ret  Bond_VW_Ret
0    1926-01-30        809.0  1926      1     0.010831     0.013395
```

1	1926-02-27	809.0	1926	2	0.005030	0.006154
2	1926-03-31	809.0	1926	3	0.002123	0.003881
3	1926-04-30	809.0	1926	4	0.006391	0.007441
4	1926-05-28	809.0	1926	5	0.001898	0.001472
5	1926-06-30	809.0	1926	6	0.003308	0.003750
6	1926-07-31	809.0	1926	7	0.002850	0.000734
7	1926-08-31	809.0	1926	8	0.003272	0.000409
8	1926-09-30	809.0	1926	9	0.003808	0.003769
9	1926-10-30	809.0	1926	10	0.005722	0.009623
10	1926-11-30	809.0	1926	11	0.009324	0.015172
11	1926-12-31	809.0	1926	12	0.005243	0.007496
12	1927-01-31	809.0	1927	1	0.005730	0.007256
13	1927-02-28	809.0	1927	2	0.008836	0.008785
14	1927-03-31	809.0	1927	3	0.015889	0.024161
15	1927-04-30	809.0	1927	4	0.002297	-0.000147
16	1927-05-31	809.0	1927	5	0.007364	0.010432
17	1927-06-30	809.0	1927	6	-0.002838	-0.006393
18	1927-07-30	809.0	1927	7	0.005662	0.005055
19	1927-08-31	809.0	1927	8	0.005129	0.007295
20	1927-09-30	809.0	1927	9	0.005332	0.002235
21	1927-10-31	809.0	1927	10	0.011937	0.010115
22	1927-11-30	809.0	1927	11	0.005465	0.009181
23	1927-12-31	809.0	1927	12	0.006103	0.007021
24	1928-01-31	809.0	1928	1	0.003236	-0.002763
25	1928-02-29	809.0	1928	2	0.004246	0.005888
26	1928-03-31	809.0	1928	3	0.003524	0.004419
27	1928-04-30	809.0	1928	4	-0.001451	-0.000494
28	1928-05-31	809.0	1928	5	-0.006338	-0.007544
29	1928-06-30	809.0	1928	6	-0.002973	0.003256
...	...	...	...	...	...	...
1086	2016-07-29	12601706.0	2016	7	0.001952	0.002015
1087	2016-08-31	12463618.0	2016	8	-0.004497	-0.004319
1088	2016-09-30	12606015.0	2016	9	0.000087	0.000015
1089	2016-10-31	12785272.0	2016	10	-0.007679	-0.007448
1090	2016-11-30	12832288.0	2016	11	-0.019958	-0.019044
1091	2016-12-30	12939383.0	2016	12	-0.000509	-0.000435
1092	2017-01-31	12828198.0	2017	1	0.002047	0.001831
1093	2017-02-28	12897181.0	2017	2	0.002527	0.002423
1094	2017-03-31	12827905.0	2017	3	0.001539	0.001442
1095	2017-04-28	12990504.0	2017	4	0.004885	0.004644
1096	2017-05-31	12820131.0	2017	5	0.004681	0.004551
1097	2017-06-30	12962743.0	2017	6	-0.000614	-0.000531
1098	2017-07-31	13084287.0	2017	7	0.001198	0.000997
1099	2017-08-31	13050300.0	2017	8	0.008500	0.008129
1100	2017-09-29	13294367.0	2017	9	-0.006494	-0.006117
1101	2017-10-31	13204087.0	2017	10	-0.000540	-0.000510
1102	2017-11-30	13265256.0	2017	11	-0.000522	-0.000302
1103	2017-12-29	13486542.0	2017	12	0.002148	0.002142

```
1104 2018-01-31   13350861.0  2018      1    -0.009274    -0.008732
1105 2018-02-28   13482123.0  2018      2    -0.005440    -0.005251
1106 2018-03-29   13733443.0  2018      3     0.007033     0.006626
1107 2018-04-30   13629428.0  2018      4    -0.006348    -0.005938
1108 2018-05-31   13712160.0  2018      5     0.006329     0.005883
1109 2018-06-29   14010069.0  2018      6     0.001706     0.001636
1110 2018-07-31   13865396.0  2018      7    -0.002378    -0.002228
1111 2018-08-31   13941011.0  2018      8     0.005910     0.005431
1112 2018-09-28   14248302.0  2018      9    -0.006641    -0.006163
1113 2018-10-31   14123373.0  2018     10    -0.002302    -0.002287
1114 2018-11-30   14173523.0  2018     11     0.007772     0.007177
1115 2018-12-31   14410397.0  2018     12     0.015903     0.014693

[1116 rows x 6 columns]
```

2

Stock excess return is calculated the same way as in problem set 1, with the only difference that it starts from January 1926.

Bond excess value weighted return equals to bond value weighted return minus risk free rate t30ret.

```
In [2]: def PS1_Q1(data):
            data['date'] = pd.to_datetime(data['date'].astype(str))
            data['market_cap'] = np.multiply(abs(data['PRC']), data['SHROUT'])
            data['market_cap'] = data[['PERMNO', 'market_cap']]\
                .groupby('PERMNO').shift(1)
            data.dropna(subset=['market_cap'], inplace=True)
            output = data[['market_cap', 'date']].groupby('date').sum()
            data.loc[data['DLRET'].apply(lambda x: str(x).isalpha()), 'DLRET']\
                = np.nan
            data.loc[data['RET'].apply(lambda x: str(x).isalpha()), 'RET'] = np.nan
            na_return = data[data['RET'].isna()].index
            not_na_delist = data[data['DLRET'].notna()].index
            is_na = np.intersect1d(na_return, not_na_delist)
            not_na = data[data['RET'].notna() & data['DLRET'].notna()].index
            data.loc[is_na, 'RET'] = data.loc[is_na, 'DLRET']
            data.loc[not_na, 'RET'] = (data.loc[not_na, 'RET'].astype(float) + 1)\
                                      * (data.loc[not_na, 'DLRET'].astype(float) + 1) - 1
            data.dropna(subset=['PRC', 'RET', ], inplace=True)
            data['RET'] = data['RET'].astype(float)
            output['Year'] = output.index.year
            output['Month'] = output.index.month
            output['Stock Ew Ret'] = data[['date', 'RET']].groupby('date').mean()
            output.dropna(subset=['market_cap'],inplace=True)
            output.reset_index(inplace=True)
            data = pd.merge(data, output[['date', 'market_cap']],
                            how='left', on=['date'])
```

```
    output['Stock Vw Ret'] = pd.DataFrame({'date': data['date'],
    'x': data['market_cap_x'] / data['market_cap_y'] * data['RET']
                                            }).groupby('date').sum().values
    output = output[['Year', 'Month', 'market_cap', 'Stock Ew Ret',
                     'Stock Vw Ret']]
    return output

stock_data = pd.read_csv('/Users/huanyu/Desktop/'
                         'QuantitativeAssetManagement/hw2/stocks.csv')
stock_data = stock_data.loc[((stock_data['EXCHCD'] == 1) |
            (stock_data['EXCHCD'] == 2) | (stock_data['EXCHCD'] == 3))
            & ((stock_data['SHRCD'] == 10) | (stock_data['SHRCD'] == 11)),:]
stock_data.reset_index(inplace=True,drop=True)
output_stock = PS1_Q1(stock_data)

monthly_crsp_riskless = pd.read_csv('/Users/huanyu/Desktop/'
                        'QuantitativeAssetManagement/hw2/riskfree.csv')
monthly_crsp_riskless['MCALDT'] = \
    pd.to_datetime(monthly_crsp_riskless['caldt'],format='%Y%m%d')
output_q2 = pd.merge(output,output_stock,how = 'left',on=['Year','Month'])
output_q2 = pd.merge(output_q2,monthly_crsp_riskless[['MCALDT','t90ret','t30ret']],
                     how='left',on='MCALDT')
output_q2['Stock_Excess_VW_Ret'] = output_q2['Stock Vw Ret'] - output_q2['t30ret']
output_q2['Bond_Excess_VW_Ret'] = output_q2['Bond_VW_Ret'] - output_q2['t30ret']
output_q2 = output_q2[['Year','Month','market_cap',
            'Stock_Excess_VW_Ret','Bond_lag_MV','Bond_Excess_VW_Ret']]
output_q2.rename(columns={'market_cap':'Stock_lag_MV'},inplace=True)
output_q2['Stock_lag_MV'] = output_q2['Stock_lag_MV'] / 1000
print(output_q2)
```

	Year	Month	Stock_lag_MV	Stock_Excess_VW_Ret	Bond_lag_MV	\
0	1926	1	2.690395e+04	-0.002704	809.0	
1	1926	2	2.703235e+04	-0.036744	809.0	
2	1926	3	2.616208e+04	-0.067570	809.0	
3	1926	4	2.450693e+04	0.033623	809.0	
4	1926	5	2.527439e+04	0.011584	809.0	
5	1926	6	2.560911e+04	0.050952	809.0	
6	1926	7	2.693727e+04	0.028994	809.0	
7	1926	8	2.740211e+04	0.026288	809.0	
8	1926	9	2.842439e+04	0.003560	809.0	
9	1926	10	2.830853e+04	-0.032276	809.0	
10	1926	11	2.750110e+04	0.025092	809.0	
11	1926	12	2.828130e+04	0.026196	809.0	
12	1927	1	2.961535e+04	-0.000564	809.0	
13	1927	2	2.978370e+04	0.041690	809.0	
14	1927	3	3.121784e+04	0.001282	809.0	
15	1927	4	3.119938e+04	0.004570	809.0	
16	1927	5	3.145286e+04	0.054253	809.0	

17	1927	6	3.346963e+04	-0.023369	809.0
18	1927	7	3.278278e+04	0.072182	809.0
19	1927	8	3.501544e+04	0.019508	809.0
20	1927	9	3.579363e+04	0.047524	809.0
21	1927	10	3.905272e+04	-0.042838	809.0
22	1927	11	3.729554e+04	0.065225	809.0
23	1927	12	3.951204e+04	0.020777	809.0
24	1928	1	4.089981e+04	-0.006415	809.0
25	1928	2	4.013160e+04	-0.016856	809.0
26	1928	3	3.940402e+04	0.087581	809.0
27	1928	4	4.292025e+04	0.042069	809.0
28	1928	5	4.471265e+04	0.015479	809.0
29	1928	6	4.565552e+04	-0.048240	809.0
...	...	...	...	...	...
1086	2016	7	2.159993e+07	0.039466	12601706.0
1087	2016	8	2.237217e+07	0.004977	12463618.0
1088	2016	9	2.240320e+07	0.002506	12606015.0
1089	2016	10	2.234065e+07	-0.020207	12785272.0
1090	2016	11	2.184712e+07	0.048547	12832288.0
1091	2016	12	2.281248e+07	0.018129	12939383.0
1092	2017	1	2.314780e+07	0.019347	12828198.0
1093	2017	2	2.356372e+07	0.035595	12897181.0
1094	2017	3	2.429574e+07	0.001667	12827905.0
1095	2017	4	2.427995e+07	0.010852	12990504.0
1096	2017	5	2.453134e+07	0.010570	12820131.0
1097	2017	6	2.474014e+07	0.007788	12962743.0
1098	2017	7	2.483131e+07	0.018624	13084287.0
1099	2017	8	2.515006e+07	0.001720	13050300.0
1100	2017	9	2.515273e+07	0.024936	13294367.0
1101	2017	10	2.576627e+07	0.022477	13204087.0
1102	2017	11	2.629452e+07	0.031170	13265256.0
1103	2017	12	2.704945e+07	0.010544	13486542.0
1104	2018	1	2.730054e+07	0.055742	13350861.0
1105	2018	2	2.883811e+07	-0.036466	13482123.0
1106	2018	3	2.771109e+07	-0.022535	13733443.0
1107	2018	4	2.701560e+07	0.002865	13629428.0
1108	2018	5	2.714952e+07	0.026456	13712160.0
1109	2018	6	2.785599e+07	0.004674	14010069.0
1110	2018	7	2.788886e+07	0.031908	13865396.0
1111	2018	8	2.877993e+07	0.034438	13941011.0
1112	2018	9	2.971025e+07	0.000601	14248302.0
1113	2018	10	2.969744e+07	-0.076512	14123373.0
1114	2018	11	2.736560e+07	0.016811	14173523.0
1115	2018	12	2.767927e+07	-0.095230	14410397.0

	Bond_Excess_VW_Ret
0	0.010444
1	0.003386

2	0.001103
3	0.004369
4	0.001130
5	0.000291
6	-0.001509
7	-0.002127
8	0.001496
9	0.006428
10	0.012079
11	0.004718
12	0.004793
13	0.006215
14	0.021202
15	-0.002693
16	0.007424
17	-0.008976
18	0.002064
19	0.004538
20	0.000139
21	0.007595
22	0.007095
23	0.004791
24	-0.004988
25	0.002594
26	0.001483
27	-0.002718
28	-0.010464
29	0.000134
...	...
1086	0.001844
1087	-0.004484
1088	-0.000204
1089	-0.007616
1090	-0.019180
1091	-0.000674
1092	0.001473
1093	0.002059
1094	0.001089
1095	0.004107
1096	0.003937
1097	-0.001148
1098	0.000275
1099	0.007302
1100	-0.007061
1101	-0.001350
1102	-0.001128
1103	0.001258
1104	-0.009853

```
1105           -0.006324
1106            0.006365
1107           -0.007371
1108            0.004482
1109            0.000228
1110           -0.003786
1111            0.003856
1112           -0.007653
1113           -0.004177
1114            0.005382
1115            0.012763

[1116 rows x 6 columns]
```

3

Excess value return is based on the weight of two different assets, bonds and stocks.

$$\text{Excess value return} = \frac{\text{Market value of stock}}{\text{Total market value of stock and bond}} \times \text{Stock excess return} + \frac{\text{Market value of bon}}{\text{Total market value of stock}}$$

Excess 60/40 return is very similar to excess value return except that the weights of stock and bond are fixed at 60% and 40% correspondingly.

Stock inverse sigma and bond inverse sigma are the inverse of its volatility, estimated using 3-year monthly excess returns up to month t - 1.

Unlevered k: $k_t = \frac{1}{\sigma_{s,t}^{-1} + \sigma_{b,t}^{-1}}$

Excess unlevered risk-parity portfolio return:

$$w_{t,i} = k_t \sigma_{i,t}^{-1}, \text{ i = bond, stock}$$

$$r_t^{RP} = \sum_i w_{t-1,i}(r_{t,i} - rf_t)$$

Levered k: We set k such that the annualized volatility of this portfolio matches the ex post realized volatility of the benchmark (the value-weighted market portfolio).

$$k = \frac{\sigma_{VW}}{\sigma_{RP,k=1}}$$

Excess levered return:

$$w_{t,i} = k \sigma_{i,t}^{-1}, \text{ i = bond, stock}$$

$$r_t^{RP} = \sum_i w_{t-1,i}(r_{t,i} - rf_t)$$

```
In [3]: output_q3 = output_q2[['Year','Month','Stock_Excess_VW_Ret',
                               'Bond_Excess_VW_Ret']].copy()
        output_q3['total_market_cap'] = output_q2['Bond_lag_MV'] + output_q2['Stock_lag_MV']
        output_q3['Excess_Vw_Ret'] = output_q2['Bond_lag_MV'] / \
                output_q3['total_market_cap'] * output_q3['Bond_Excess_VW_Ret'] + \
                output_q2['Stock_lag_MV'] / output_q3['total_market_cap'] * \
                output_q3['Stock_Excess_VW_Ret']
        output_q3['Excess_60_40_Ret'] = 0.6 * output_q3['Stock_Excess_VW_Ret'] +\
                                        0.4 * output_q3['Bond_Excess_VW_Ret']
        output_q3['Stock_inverse_sigma_hat'] = \
            1 / output_q3['Stock_Excess_VW_Ret'].rolling(36).std().shift(1)
        output_q3['Bond_inverse_sigma_hat'] = \
            1 / output_q3['Bond_Excess_VW_Ret'].rolling(36).std().shift(1)
        output_q3['Unlevered_k'] = \
            1 / (output_q3['Stock_inverse_sigma_hat'] + output_q3['Bond_inverse_sigma_hat'])
        output_q3['Excess_Unlevered_RP_Ret'] = \
            (output_q3['Unlevered_k'] * output_q3['Stock_inverse_sigma_hat']).shift(1)\
            * output_q3['Stock_Excess_VW_Ret'] + (output_q3['Unlevered_k'] *
            output_q3['Bond_inverse_sigma_hat']).shift(1) * output_q3['Bond_Excess_VW_Ret']
        constant_k = output_q3.loc[output_q3['Year'] > 1928,'Excess_Vw_Ret'].std() / \
            ((output_q3['Stock_inverse_sigma_hat']).shift(1) *
             output_q3['Stock_Excess_VW_Ret'] +
             (output_q3['Bond_inverse_sigma_hat']).shift(1) *
             output_q3['Bond_Excess_VW_Ret']).std()
        output_q3['Excess_Levered_RP_Ret'] = \
            (constant_k * output_q3['Stock_inverse_sigma_hat']).shift(1) * \
            output_q3['Stock_Excess_VW_Ret'] + \
            (constant_k * output_q3['Bond_inverse_sigma_hat']).shift(1) * \
            output_q3['Bond_Excess_VW_Ret']
        print(output_q3)
```

	Year	Month	Stock_Excess_VW_Ret	Bond_Excess_VW_Ret	total_market_cap	\
0	1926	1	-0.002704	0.010444	2.771295e+04	
1	1926	2	-0.036744	0.003386	2.784135e+04	
2	1926	3	-0.067570	0.001103	2.697108e+04	
3	1926	4	0.033623	0.004369	2.531593e+04	
4	1926	5	0.011584	0.001130	2.608339e+04	
5	1926	6	0.050952	0.000291	2.641811e+04	
6	1926	7	0.028994	-0.001509	2.774627e+04	
7	1926	8	0.026288	-0.002127	2.821111e+04	
8	1926	9	0.003560	0.001496	2.923339e+04	
9	1926	10	-0.032276	0.006428	2.911753e+04	
10	1926	11	0.025092	0.012079	2.831010e+04	
11	1926	12	0.026196	0.004718	2.909030e+04	
12	1927	1	-0.000564	0.004793	3.042435e+04	
13	1927	2	0.041690	0.006215	3.059270e+04	
14	1927	3	0.001282	0.021202	3.202684e+04	
15	1927	4	0.004570	-0.002693	3.200838e+04	

16	1927	5	0.054253	0.007424	3.226186e+04
17	1927	6	-0.023369	-0.008976	3.427863e+04
18	1927	7	0.072182	0.002064	3.359178e+04
19	1927	8	0.019508	0.004538	3.582444e+04
20	1927	9	0.047524	0.000139	3.660263e+04
21	1927	10	-0.042838	0.007595	3.986172e+04
22	1927	11	0.065225	0.007095	3.810454e+04
23	1927	12	0.020777	0.004791	4.032104e+04
24	1928	1	-0.006415	-0.004988	4.170881e+04
25	1928	2	-0.016856	0.002594	4.094060e+04
26	1928	3	0.087581	0.001483	4.021302e+04
27	1928	4	0.042069	-0.002718	4.372925e+04
28	1928	5	0.015479	-0.010464	4.552165e+04
29	1928	6	-0.048240	0.000134	4.646452e+04
...	...	...	...	...	...
1086	2016	7	0.039466	0.001844	3.420163e+07
1087	2016	8	0.004977	-0.004484	3.483578e+07
1088	2016	9	0.002506	-0.000204	3.500921e+07
1089	2016	10	-0.020207	-0.007616	3.512592e+07
1090	2016	11	0.048547	-0.019180	3.467940e+07
1091	2016	12	0.018129	-0.000674	3.575186e+07
1092	2017	1	0.019347	0.001473	3.597600e+07
1093	2017	2	0.035595	0.002059	3.646090e+07
1094	2017	3	0.001667	0.001089	3.712365e+07
1095	2017	4	0.010852	0.004107	3.727046e+07
1096	2017	5	0.010570	0.003937	3.735147e+07
1097	2017	6	0.007788	-0.001148	3.770288e+07
1098	2017	7	0.018624	0.000275	3.791560e+07
1099	2017	8	0.001720	0.007302	3.820036e+07
1100	2017	9	0.024936	-0.007061	3.844710e+07
1101	2017	10	0.022477	-0.001350	3.897036e+07
1102	2017	11	0.031170	-0.001128	3.955977e+07
1103	2017	12	0.010544	0.001258	4.053599e+07
1104	2018	1	0.055742	-0.009853	4.065140e+07
1105	2018	2	-0.036466	-0.006324	4.232023e+07
1106	2018	3	-0.022535	0.006365	4.144453e+07
1107	2018	4	0.002865	-0.007371	4.064502e+07
1108	2018	5	0.026456	0.004482	4.086168e+07
1109	2018	6	0.004674	0.000228	4.186606e+07
1110	2018	7	0.031908	-0.003786	4.175425e+07
1111	2018	8	0.034438	0.003856	4.272094e+07
1112	2018	9	0.000601	-0.007653	4.395856e+07
1113	2018	10	-0.076512	-0.004177	4.382081e+07
1114	2018	11	0.016811	0.005382	4.153912e+07
1115	2018	12	-0.095230	0.012763	4.208967e+07

	Excess_Vw_Ret	Excess_60_40_Ret	Stock_inverse_sigma_hat	\
0	-0.002320	0.002555	NaN	

1	-0.035578	-0.020692	NaN
2	-0.065510	-0.040101	NaN
3	0.032688	0.021921	NaN
4	0.011259	0.007402	NaN
5	0.049400	0.030687	NaN
6	0.028104	0.016792	NaN
7	0.025473	0.014922	NaN
8	0.003503	0.002734	NaN
9	-0.031201	-0.016794	NaN
10	0.024721	0.019887	NaN
11	0.025598	0.017604	NaN
12	-0.000421	0.001579	NaN
13	0.040752	0.027500	NaN
14	0.001785	0.009250	NaN
15	0.004386	0.001665	NaN
16	0.053079	0.035521	NaN
17	-0.023029	-0.017612	NaN
18	0.070494	0.044135	NaN
19	0.019170	0.013520	NaN
20	0.046476	0.028570	NaN
21	-0.041815	-0.022665	NaN
22	0.063991	0.041973	NaN
23	0.020457	0.014383	NaN
24	-0.006388	-0.005844	NaN
25	-0.016472	-0.009076	NaN
26	0.085849	0.053142	NaN
27	0.041241	0.024154	NaN
28	0.015017	0.005102	NaN
29	-0.047398	-0.028890	NaN
...	...	...	...
1086	0.025604	0.024417	29.877716
1087	0.001592	0.001193	30.392767
1088	0.001530	0.001422	30.924122
1089	-0.015624	-0.015170	31.256172
1090	0.023487	0.021457	31.418476
1091	0.011324	0.010608	30.926736
1092	0.012974	0.012198	31.060904
1093	0.023733	0.022181	31.744286
1094	0.001467	0.001436	32.074307
1095	0.008501	0.008154	32.061107
1096	0.008294	0.007917	32.107378
1097	0.004716	0.004213	32.176159
1098	0.012292	0.011284	32.333516
1099	0.003627	0.003953	32.683775
1100	0.013872	0.012137	33.255933
1101	0.014404	0.012946	33.525415
1102	0.020340	0.018251	33.568603
1103	0.007455	0.006830	33.448859

1104	0.034199	0.029504	33.498676
1105	-0.026863	-0.024409	33.333008
1106	-0.012958	-0.010975	33.545558
1107	-0.000567	-0.001229	33.235423
1108	0.019082	0.017666	33.221807
1109	0.003186	0.002896	33.071395
1110	0.020055	0.017630	33.375932
1111	0.024458	0.022205	33.134314
1112	-0.002075	-0.002701	35.807954
1113	-0.053198	-0.047578	37.021040
1114	0.012911	0.012239	34.806149
1115	-0.058256	-0.052033	34.781196

	Bond_inverse_sigma_hat	Unlevered_k	Excess_Unlevered_RP_Ret	\
0	NaN	NaN	NaN	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
5	NaN	NaN	NaN	
6	NaN	NaN	NaN	
7	NaN	NaN	NaN	
8	NaN	NaN	NaN	
9	NaN	NaN	NaN	
10	NaN	NaN	NaN	
11	NaN	NaN	NaN	
12	NaN	NaN	NaN	
13	NaN	NaN	NaN	
14	NaN	NaN	NaN	
15	NaN	NaN	NaN	
16	NaN	NaN	NaN	
17	NaN	NaN	NaN	
18	NaN	NaN	NaN	
19	NaN	NaN	NaN	
20	NaN	NaN	NaN	
21	NaN	NaN	NaN	
22	NaN	NaN	NaN	
23	NaN	NaN	NaN	
24	NaN	NaN	NaN	
25	NaN	NaN	NaN	
26	NaN	NaN	NaN	
27	NaN	NaN	NaN	
28	NaN	NaN	NaN	
29	NaN	NaN	NaN	
...	...	...	...	
1086	139.578257	0.005901	0.008302	
1087	139.896612	0.005872	-0.002815	
1088	140.389025	0.005837	0.000280	

1089	140.992463	0.005806	-0.009889
1090	137.484328	0.005921	-0.006890
1091	124.300110	0.006442	0.002823
1092	126.548899	0.006345	0.005034
1093	130.550146	0.006162	0.008668
1094	130.536116	0.006150	0.001202
1095	130.909774	0.006136	0.005437
1096	130.940687	0.006133	0.005242
1097	131.728783	0.006101	0.000611
1098	131.713311	0.006096	0.003877
1099	131.782272	0.006080	0.006201
1100	132.242550	0.006042	-0.000702
1101	130.774064	0.006086	0.003438
1102	131.628706	0.006053	0.005463
1103	132.849845	0.006013	0.003145
1104	132.840948	0.006012	0.003340
1105	144.927022	0.005610	-0.012394
1106	150.022373	0.005448	0.000961
1107	149.125518	0.005484	-0.005500
1108	147.225633	0.005542	0.008487
1109	146.772768	0.005560	0.001047
1110	148.802278	0.005489	0.002777
1111	151.063959	0.005429	0.009459
1112	150.476999	0.005368	-0.006169
1113	150.855858	0.005323	-0.018081
1114	150.491555	0.005397	0.007634
1115	149.361325	0.005431	-0.007523

	Excess_Levered_RP_Ret
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
5	NaN
6	NaN
7	NaN
8	NaN
9	NaN
10	NaN
11	NaN
12	NaN
13	NaN
14	NaN
15	NaN
16	NaN
17	NaN
18	NaN

```
19                    NaN
20                    NaN
21                    NaN
22                    NaN
23                    NaN
24                    NaN
25                    NaN
26                    NaN
27                    NaN
28                    NaN
29                    NaN
...                   ...
1086             0.036548
1087            -0.012121
1088             0.001210
1089            -0.043039
1090            -0.030151
1091             0.012115
1092             0.019854
1093             0.034708
1094             0.004956
1095             0.022463
1096             0.021704
1097             0.002532
1098             0.016145
1099             0.025846
1100            -0.002934
1101             0.014456
1102             0.022802
1103             0.013201
1104             0.014113
1105            -0.052378
1106             0.004353
1107            -0.025651
1108             0.039319
1109             0.004800
1110             0.012690
1111             0.043780
1112            -0.028867
1113            -0.085573
1114             0.036437
1115            -0.035414

[1116 rows x 12 columns]
```

My result is similar to the table reported in the paper, however, the difference is not 0 between them. The reasons of a nonzero difference may be that the difference methods to deal with NA

values. Second, my table is based on data from 1930 Jan to June 2010 according to the problem set. However, the table in the paper is based on data from 1926 to 2010. Most of the diffence is economically negligible, however, the crsp stock returns are 7.07% and 6.71%, which cannot be neglected. In the paper, crsp stock return maybe use stock index data, and I used all the individual stocks in the market.

```
In [4]: index = ['CRSP stocks','CRSP bonds','Value-weighted portfolio',
                 '60/40 portfolio','unlevered RP','levered RP']
        q4_data = output_q3.loc[48:1014,('Stock_Excess_VW_Ret','Bond_Excess_VW_Ret',
            'Excess_Vw_Ret','Excess_60_40_Ret','Excess_Unlevered_RP_Ret',
            'Excess_Levered_RP_Ret')]
        observations = len(q4_data)
        Annualized_Mean = q4_data.mean() * 12
        t_stats = q4_data.apply(lambda x: x.mean()/x.std() * math.sqrt(observations))
        Annualized_std = q4_data.std() * math.sqrt(12)
        Annualized_SR = Annualized_Mean / Annualized_std
        Skewness = q4_data.skew()
        Excess_Kurtosis = q4_data.kurt()
        output_q4 = pd.DataFrame({
            'Annualized Mean':Annualized_Mean.values,'T stat':t_stats.values,
            'Annualized Standard Deviation':Annualized_std.values,
            'Annualized Sharpe Ratio':Annualized_SR.values,'Skewness':Skewness.values,
            'Excess Kurtosis':Excess_Kurtosis.values},index=index)
        print(output_q4)
```

```
                         Annualized Mean    T stat  \
CRSP stocks                     0.070780  3.359947
CRSP bonds                      0.015594  4.346458
Value-weighted portfolio        0.041400  2.521863
60/40 portfolio                 0.048705  3.762773
unlevered RP                    0.022761  4.821120
levered RP                      0.079824  4.844708

                         Annualized Standard Deviation  \
CRSP stocks                                   0.189103
CRSP bonds                                    0.032206
Value-weighted portfolio                      0.147368
60/40 portfolio                               0.116196
unlevered RP                                  0.042381
levered RP                                    0.147906

                         Annualized Sharpe Ratio  Skewness  Excess Kurtosis
CRSP stocks                             0.374291  0.292549         7.903689
CRSP bonds                              0.484187 -0.052522         4.872829
Value-weighted portfolio                0.280930  0.646639        14.883597
60/40 portfolio                         0.419165  0.286360         7.820896
unlevered RP                            0.537063  0.103982         4.805227
levered RP                              0.539691 -0.334631         2.044417
```