Spring 2021 Final Exam
Yield Curve Interpolation and Government Bond Future Contract

FRE6411 - Fixed Income Securities
Tandon School of Engineering
Ken Chen (kc3037), Yipei Zhang (yz4369)

April 11, 2021

For June 2021 Maturity, US Ultra-Long Bond(USM1), US Ten Year (TYM1) and UK Long Gilt (G_M1), Futures Contracts, the deliverable bonds against each of these contracts are given.

A) For each of the above bonds, using Cubic Spline interpolation and CMY From the US Fed and nominal zero-coupon yield from Bank of England find the time series of daily prices, YTM, Duration and Convexity for each bond from 1/1/2020-2/28/2021.

First, we collect CMY from US Fed (https://www.treasury.gov/resource-center/data-chart-center/interest-rates/pages/TextView.aspx?data=yieldYear&year=2021) and nominal zero-coupon yield from Bank of England (https://www.bankofengland.co.uk/statistics/yield-curves), shown in table below.

Date	0.083333	0.166667	0.250000	0.500000	1.000000	2.000000	3.000000	5.000000	7.000000	10.000000	20.000000	30.000000
2020-01-02	1.53	1.55	1.54	1.57	1.56	1.58	1.59	1.67	1.79	1.88	2.19	2.33
2020-01-03	1.52	1.55	1.52	1.55	1.55	1.53	1.54	1.59	1.71	1.80	2.11	2.26
2020-01-06	1.54	1.54	1.56	1.56	1.54	1.54	1.56	1.61	1.72	1.81	2.13	2.28
2020-01-07	1.52	1.53	1.54	1.56	1.53	1.54	1.55	1.62	1.74	1.83	2.16	2.31
2020-01-08	1.50	1.53	1.54	1.56	1.55	1.58	1.61	1.67	1.78	1.87	2.21	2.35

Table: CMY from US Fed

Date	0.083333	0.166667	0.250000	0.333333	0.416667	0.500000	0.583333	0.666667	0.750000	0.833333	...	4.250000	4.333333	4.416666	4.500000	4
2020-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	
2020-01-02	NaN	NaN	NaN	NaN	NaN	NaN	0.572733	0.554150	0.537397	0.522640	...	0.626752	0.632677	0.638734	0.644932	0
2020-01-03	NaN	NaN	NaN	NaN	NaN	NaN	0.569617	0.550312	0.532435	0.516221	...	0.570232	0.575911	0.581745	0.587742	0
2020-01-06	NaN	NaN	NaN	NaN	NaN	NaN	0.606692	0.586997	0.568867	0.552466	...	0.592980	0.598362	0.603910	0.609632	0
2020-01-07	NaN	NaN	NaN	NaN	NaN	NaN	0.616413	0.598556	0.582039	0.567112	...	0.633611	0.638771	0.644068	0.649511	0

Table: zero-coupon yield from Bank of England

The columns are years and the Date index is the specific date of the yield curve.

Then, we use Cubic Spline interpolation to get yield curve for those dates.

```
def get_yield_today(today_yield):
    yield_curve=interp1d(today_yield.index.values,
                         today_yield.values,
                         kind='cubic',
                         fill_value='extrapolate')
    return yield_curve
```

The month-end yield curve for US and UK are shown below:

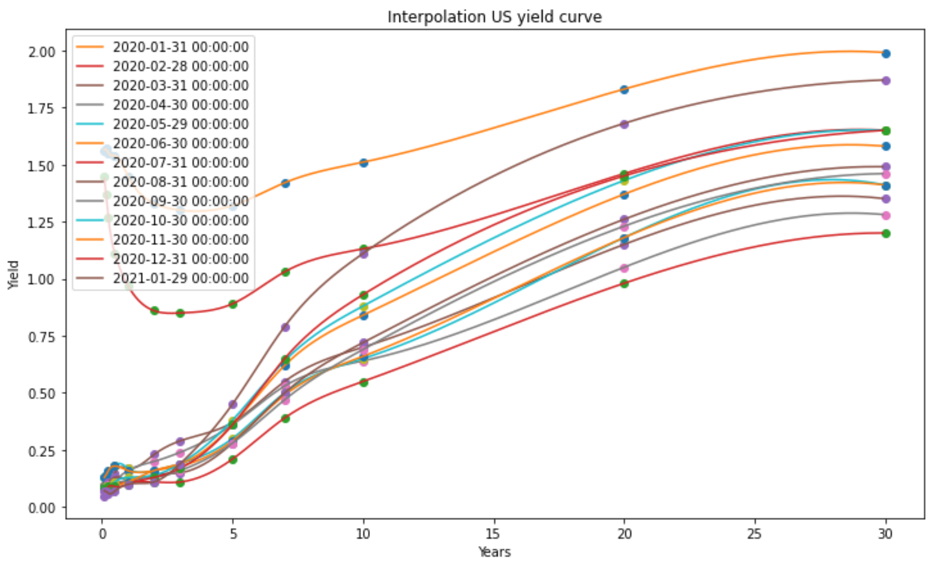

Figure: Continuous yield curve for US

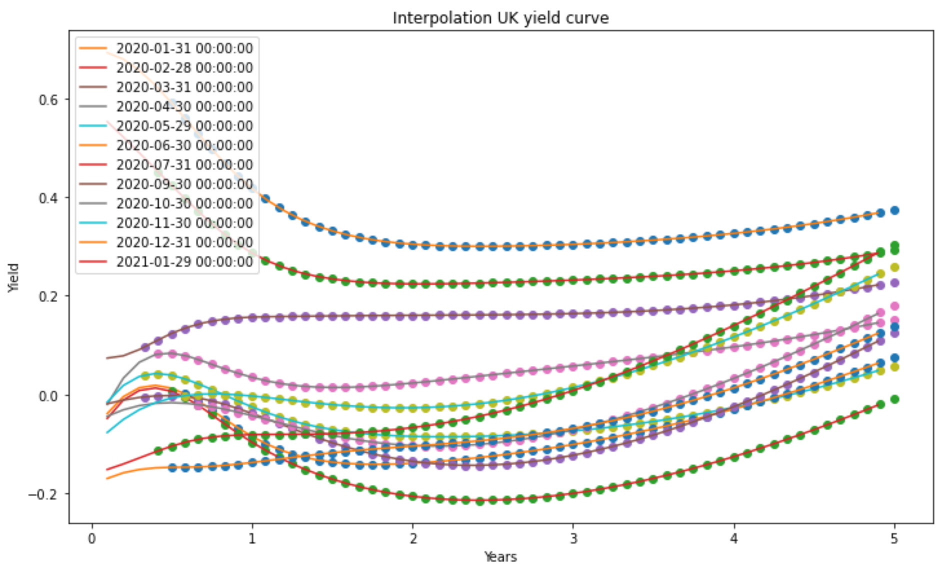

Figure: Continuous yield curve for UK

Based on the yield curve, we create a **BondPricer** class to calculate the prices, YTM, Duration and Convexity for each bond.

```
class BondPricer(FixedIncomePricer):
    def __init__(self,
                 settlement_date,
                 maturity_date,
                 yield_curve,
                 price,
                 coupon_rate,
                 freq,
                 par):
        super().__init__()
        ...
    def SolvePrice(self):
        ...
    def SolveYTM(self):
        ...
    def SolveDuration(self):
        ...
    def SolveConvexity(self):
        ...
    def GetCouponDates(self, start_date, end_date):
        ...
```

The following pricing formula is used to calculated the bonds' price:

$$P_t(T_J, C_J) = \sum_{i=1}^{N-1} \frac{C_J}{[1+y(t,T_i)]^{\frac{(T_i-t)}{365}}} + \frac{(100+C_J)}{[1+y(t,T_J)]^{\frac{(T_J-t)}{365}}}$$

Where T_J is bond maturity and C_J is the coupon payment, N is the number of semi-annual coupon payment $y(t,T_i)$is the interpolated yield T_i at t.

Result:

Some results for USM1 are displayed. More results are shown in Jupyter Notebook file.

Bond Date	T 2 ¾ 02/15/28	T 2 ⅞ 05/15/28	T 1 ⅛ 02/29/28	T 2 ⅞ 08/15/28	T 0 ¾ 01/31/28	T 0 ⅝ 12/31/27	T 3 ⅛ 11/15/28	T 2 ⅝ 02/15/29	T 2 ⅜ 05/15/29	T 1 ⅝ 08/15/29
2020-01-02	108.084402	108.536354	95.119405	109.423321	92.297064	91.151519	110.954768	107.565019	104.852705	98.605575
2020-01-03	108.688157	109.160560	95.688501	110.066389	92.853018	91.699965	111.624393	108.241216	105.541523	99.291222
2020-01-06	108.636259	109.106050	95.641674	110.009575	92.807779	91.655422	111.562989	108.178106	105.474875	99.223199
2020-01-07	108.494966	108.960044	95.504341	109.859057	92.672730	91.521977	111.407110	108.019739	105.313419	99.061098
2020-01-08	108.192398	108.647031	95.222978	109.536872	92.398858	91.252162	111.071386	107.682328	104.970671	98.722056

Table: Daily Price of USM1

Bond Date	T 2 ¾ 02/15/28	T 2 ⅞ 05/15/28	T 1 ⅛ 02/29/28	T 2 ⅞ 08/15/28	T 0 ¾ 01/31/28	T 0 ⅝ 12/31/27	T 3 ⅛ 11/15/28	T 2 ⅝ 02/15/29	T 2 ⅜ 05/15/29	T 1 ⅝ 08/15/29
2020-01-02	0.018123	0.018185	0.018210	0.018249	0.018205	0.018185	0.018297	0.018377	0.018441	0.018533
2020-01-03	0.017360	0.017422	0.017443	0.017485	0.017437	0.017416	0.017531	0.017606	0.017666	0.017752
2020-01-06	0.017445	0.017508	0.017525	0.017572	0.017518	0.017497	0.017620	0.017696	0.017758	0.017846
2020-01-07	0.017632	0.017695	0.017718	0.017758	0.017712	0.017692	0.017806	0.017884	0.017947	0.018038
2020-01-08	0.018026	0.018088	0.018107	0.018152	0.018101	0.018081	0.018201	0.018278	0.018343	0.018433

Table: YTM of USM1

Bond Date	T 2 ¾ 02/15/28	T 2 ⅞ 05/15/28	T 1 ⅛ 02/29/28	T 2 ⅞ 08/15/28	T 0 ¾ 01/31/28	T 0 ⅝ 12/31/27	T 3 ⅛ 11/15/28	T 2 ⅝ 02/15/29	T 2 ⅜ 05/15/29	T 1 ⅝ 08/15/29
2020-01-02	7.299699	7.516639	7.780777	7.668187	7.820880	7.804286	7.855320	8.138360	8.458539	8.884758
2020-01-03	7.299920	7.516947	7.779547	7.669032	7.819205	7.802296	7.856371	8.139598	8.459543	8.885367
2020-01-06	7.291369	7.508383	7.771167	7.660407	7.810874	7.793998	7.847711	8.130914	8.450877	8.876744
2020-01-07	7.287905	7.504900	7.768048	7.656792	7.807865	7.791068	7.844052	8.127205	8.447227	8.873185
2020-01-08	7.283639	7.500589	7.764542	7.652206	7.804586	7.787950	7.839360	8.122428	8.442577	8.868748

Table: Duration of USM1

Bond Date	T 2 ¾ 02/15/28	T 2 ⅞ 05/15/28	T 1 ⅛ 02/29/28	T 2 ⅞ 08/15/28	T 0 ¾ 01/31/28	T 0 ⅝ 12/31/27	T 3 ⅛ 11/15/28	T 2 ⅝ 02/15/29	T 2 ⅜ 05/15/29	T 1 ⅝ 08/15/29
2020-01-02	63.352681	66.830897	69.047864	69.927515	69.124303	68.446066	73.214294	78.145835	83.343876	90.474296
2020-01-03	63.390342	66.872417	69.072193	69.977059	69.143586	68.461755	73.269611	78.206421	83.406126	90.535887
2020-01-06	63.255461	66.733412	68.931148	69.834558	69.002482	68.321322	73.123044	78.054420	83.248302	90.370854
2020-01-07	63.194074	66.669665	68.869344	69.767844	68.941578	68.261412	73.053829	77.981928	83.173288	90.292987
2020-01-08	63.111221	66.583066	68.789966	69.675948	68.864557	68.186378	72.957435	77.880832	83.068950	90.185941

Table: Convexity of USM1

B) Find the daily rate of return of each bond and conversion factor adjusted return (conversion factor adjusted return is simply the daily return multiply by the conversion factor)

Methodology:
The bonds' conversion factors have been derivatized from CME method.
(site:https://www.cmegroup.com/trading/interest-rates/calculating-us-treasury-futures-conversion-factors.html)

Using results from A, we can simply find the daily returns and adjusted daily returns.

```
price_df = pd.pivot_table(result_df,
                          values='DailyPrice',
                          index=['Date'],
                          columns=['Bond'],
                          aggfunc=np.sum)[bonds_name]
dailyret_df = price_df.pct_change(periods=1).fillna(0)
factor_df = pd.pivot_table(result_df,
                           values='ConverFactor',
                           index=['Date'],
                           columns=['Bond'],
                           aggfunc=np.sum)[bonds_name]
adjret_df = dailyret_df * factor_df
```

Reuslt:

The time series of daily rate of return of the each bonds are calculated and the results of USM1 are displayed below:

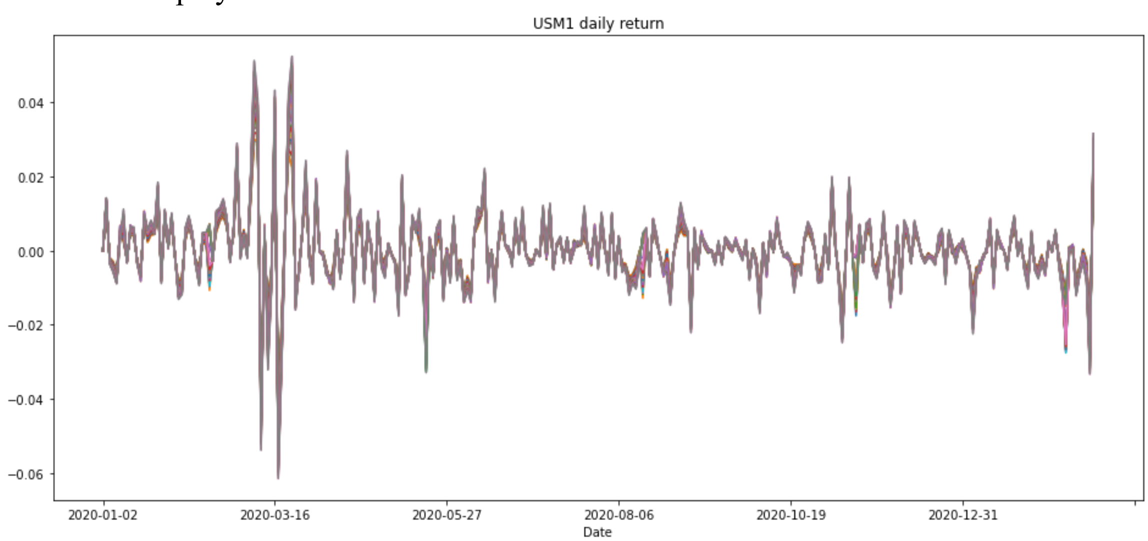

Figure: USM1 daily return

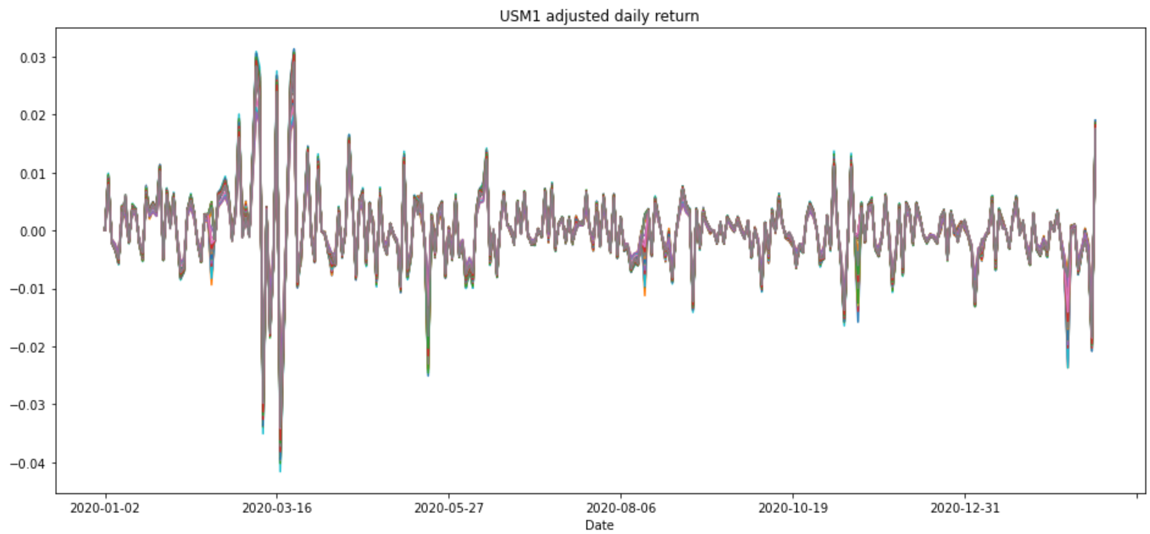

Figure: USM1 adjusted daily return

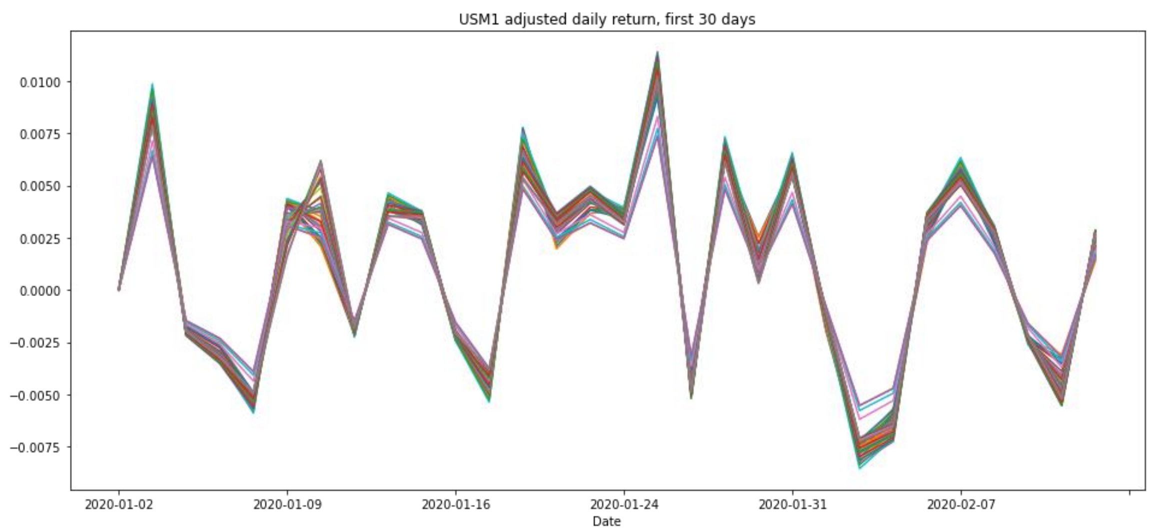

Figure: USM1 adjusted daily return, first 30 days

Comments:
Conversion factor adjusted price matters because it determines the Cheapest-to-deliver bond which can make an impact on the quote of the US Treasury bond Futures. So the conversion factor adjusted return should be monitored carefully in case that current Cheapest-to-deliver bond changes. In general, conversion factor adjusted return of all bonds move at almost the same direction
and degree.

C) For each Day, identify the Cheapest to Delivery Bond for each of the above contract (DCTD) .

Methodology:
The cheapest-to-deliver (CTD) bond is the one which it is most cost-effective for the

futures seller to deliver to the buyer if required to do so.
(site:https://www.oreilly.com/library/view/key-financial-market/9780273750123/html/chapter-024.html)
Knowing which security is the CTD is important because the futures contract tends to trade like the CTD security. Cheapest to delivery bond on specific date is the bond with the lowest price. Using the equation below, we will be able to compute each bond's prices from day one till the end.

```
dctd_df = (price_df.iloc[0, :] * (adjret_df.fillna(0).cumsum()+1)).idxmin(axis=1)
```

Then for each date, the bond with minimum price will be the Cheapest to deliver bond (CTD) on that date. Gathering CTDs for all date, we create a series of DCTD as the instruction required.

Result:
The time series of daily cheapest to delivery bond are calculated and some results are displayed below, more results are shown in Jupyter Notebook file.:

```
DCTD

Date
2020-01-02    T 1 ⅛ 05/15/40
2020-01-03    T 1 ⅛ 05/15/40
2020-01-06    T 1 ⅛ 05/15/40
2020-01-07    T 1 ⅛ 05/15/40
2020-01-08    T 1 ⅛ 05/15/40
2020-01-09    T 1 ⅛ 05/15/40
2020-01-10    T 1 ⅛ 05/15/40
2020-01-13    T 1 ⅛ 05/15/40
2020-01-14    T 1 ⅛ 05/15/40
2020-01-15    T 1 ⅛ 05/15/40
```

Table: DCTD of USM1

```
DCTD

Date
2020-01-02    T 0 ⅝ 08/15/30
2020-01-03    T 0 ⅝ 08/15/30
2020-01-06    T 0 ⅝ 08/15/30
2020-01-07    T 0 ⅝ 08/15/30
2020-01-08    T 0 ⅝ 08/15/30
2020-01-09    T 0 ⅝ 08/15/30
2020-01-10    T 0 ⅝ 08/15/30
2020-01-13    T 0 ⅝ 08/15/30
2020-01-14    T 0 ⅝ 08/15/30
2020-01-15    T 0 ⅝ 08/15/30
```

Table: DCTD of TYM1

```
DCTD

Date
2020-01-02    UKT 0 ¼ 07/31/31
2020-01-03    UKT 0 ¼ 07/31/31
2020-01-06    UKT 0 ¼ 07/31/31
2020-01-07    UKT 0 ¼ 07/31/31
2020-01-08    UKT 0 ¼ 07/31/31
2020-01-09    UKT 0 ¼ 07/31/31
2020-01-10    UKT 0 ¼ 07/31/31
2020-01-13    UKT 0 ¼ 07/31/31
2020-01-14    UKT 0 ¼ 07/31/31
2020-01-15    UKT 0 ¼ 07/31/31
```

Table: DCTD of G_M1

Comments:
After computing the DCTD series for all three future contracts, we find that CTD does not change frequently among the dates. For USM1, T 1 ⅛ 05/15/40 plays the role in DCTD for the majority of the time, and then T 1 ⅛ 08/15/40 becomes DCTD later. For TYM1, T 0 ⅝ 08/15/30 is the DCTD of all time. For G_M1, UKT 0 ¼ 07/31/31 is the DCTD of all time. It means that for the majority of time, even though for each of future contracts, it contains multiple of bonds, the future contract still tends to act like few bonds contained in its own basket.

D) Find the conversion factor adjusted daily rate of return of the above DCTD bonds for each Futures contracts.

Having DCTD series in section C, we will be able to compute the conversion factor adjusted daily rate of return, which is daily return multiply by the conversion factor.
Using the equation below, we can find the return for specific DCTD on specific date.

```
adjret_dctd_df = pd.Series([adjret_df.loc[i, dctd_df[i]] for i in dates.unique()],
                           index=dates.unique())
```

Result:
The time series of daily cheapest to delivery bond rate of return are calculated and some results are displayed below, more results are shown in Jupyter Notebook file.:

```
adjusted return for DCTD

2020-01-02    0.000000
2020-01-03    0.006447
2020-01-06   -0.001470
2020-01-07   -0.002309
2020-01-08   -0.003911
2020-01-09    0.003099
2020-01-10    0.002461
2020-01-13   -0.001490
2020-01-14    0.003158
2020-01-15    0.002440
```

Table: DCTD return of USM1

```
adjusted return for DCTD

2020-01-02    0.000000
2020-01-03    0.005175
2020-01-06   -0.000587
2020-01-07   -0.001225
2020-01-08   -0.002530
2020-01-09    0.001473
2020-01-10    0.001189
2020-01-13   -0.001184
2020-01-14    0.002049
2020-01-15    0.001927
```

Table: DCTD return of TYM1

```
adjusted return for DCTD

2020-01-02    0.000000
2020-01-03    0.009577
2020-01-06   -0.000211
2020-01-07    0.000068
2020-01-08   -0.000470
2020-01-09   -0.003421
2020-01-10    0.004345
2020-01-13    0.000669
2020-01-14    0.005946
2020-01-15    0.005464
```

Table: DCTD return of G_M1

E) Calculate the actual daily rate of return of the three Futures contract in the given time period.

Methodology:
We compute synthetic price to be actual daily rate of return of futures contract.
synthetic price=futures contract price=min(bond1 price*cf1, bond2 price*cf2,...)

Using the below equation, we will be able to compute the price for all future contracts.

```
synthetic_price = pd.Series([price_df.loc[i, dctd_df[i]] * factor_df.loc[i, dctd_df[i]] for i
                             in dates.unique()],
                             index=dates.unique())
```

Then, using the following equation to get the rate of return:

```
dailyret_future_df = synthetic_price.pct_change(periods=1).fillna(0)
```

Result:
The time series of actual daily rate of return of future contracts and some results are displayed below, more results are shown in Jupyter Notebook file.:

2020-01-02	0.000000
2020-01-03	0.014147
2020-01-06	-0.003226
2020-01-07	-0.005066
2020-01-08	-0.008583
	...
2021-02-22	-0.006480
2021-02-23	-0.001111
2021-02-24	-0.005985
2021-02-25	-0.028700
2021-02-26	0.027663

Table: USM1 return

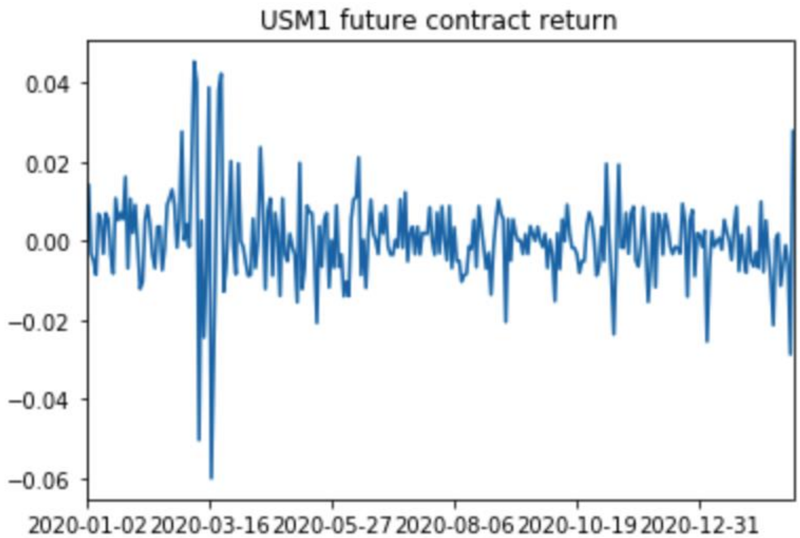

Figure: USM1 future contract return

2020-01-02	0.000000
2020-01-03	0.008209
2020-01-06	-0.000932
2020-01-07	-0.001944
2020-01-08	-0.004014
	...
2021-02-22	-0.002293
2021-02-23	-0.000137
2021-02-24	-0.000925
2021-02-25	-0.015215
2021-02-26	0.008790

Table: TYM1 return

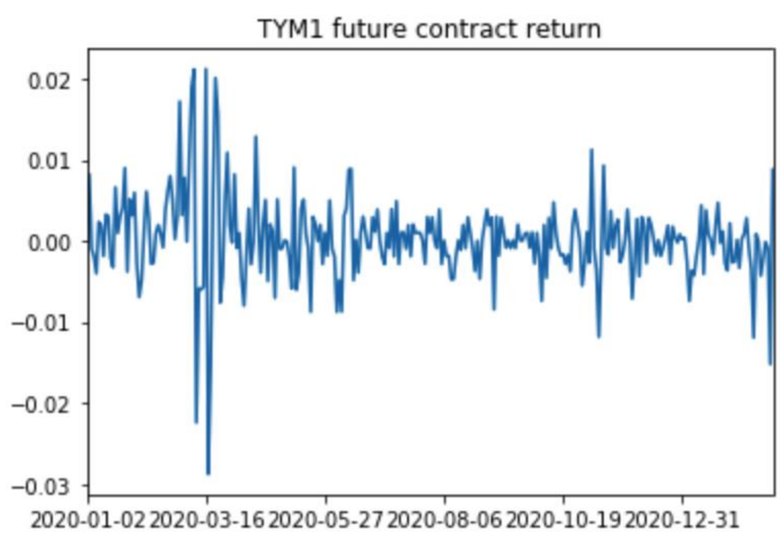

Figure: G_M1 future contract return

2020-01-02	0.000000
2020-01-03	0.013894
2020-01-06	-0.000306
2020-01-07	0.000099
2020-01-08	-0.000683
	...
2021-02-22	0.003135
2021-02-23	0.000604
2021-02-24	-0.003762
2021-02-25	-0.003768
2021-02-26	0.000299

Table: G_M1 return

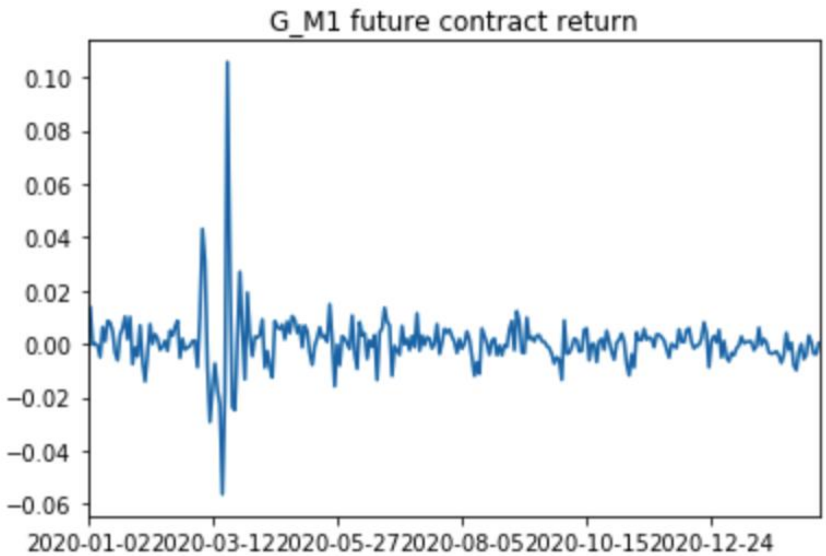

Figure: G_M1 future contract return

F) For all three contracts, compare and comment on the daily rate of return in D and E.

By making the overlap graph, we can see the trends of both DCTD adjusted return computed from D, and future contract return from E. The graphs are shown below.

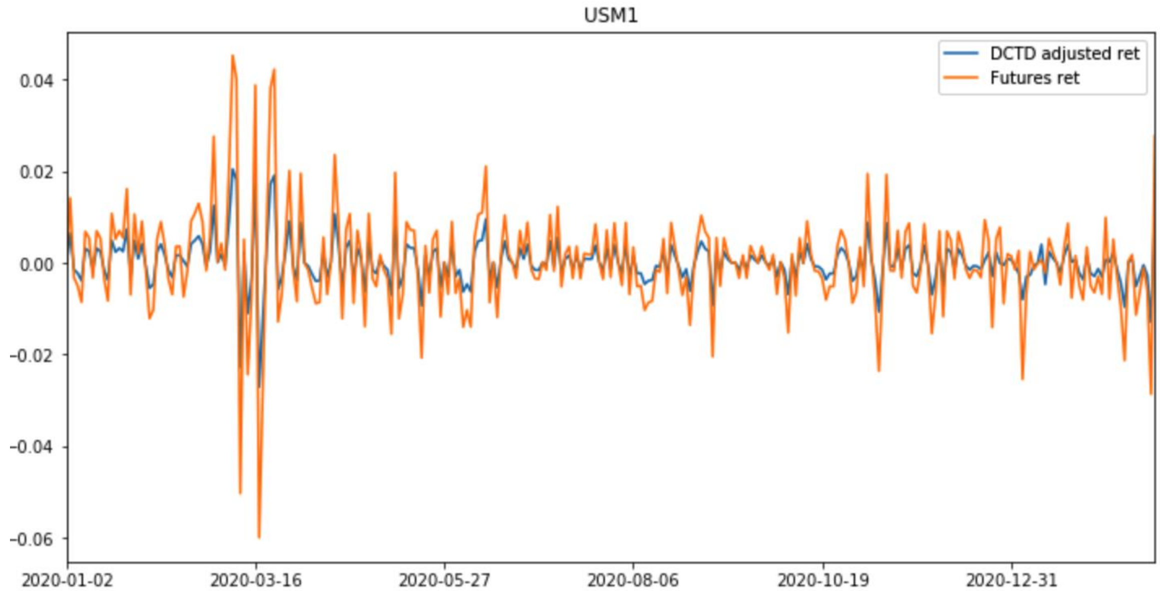

Figure: USM1 DCTD adjusted return vs futures contract return

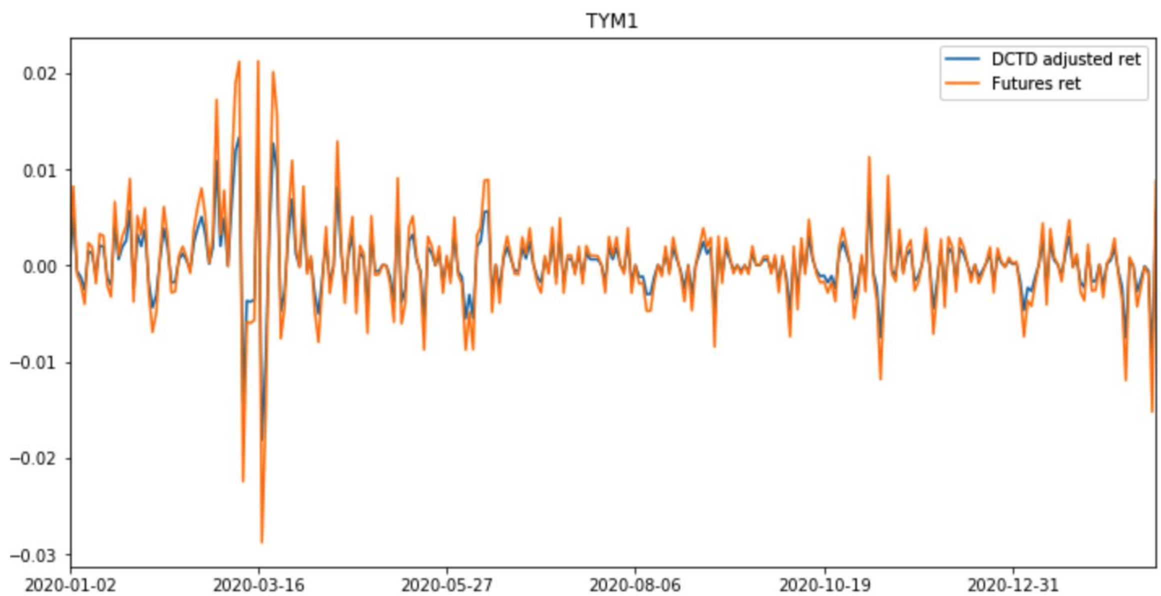

Figure: TYM1 DCTD adjusted return vs futures contract return

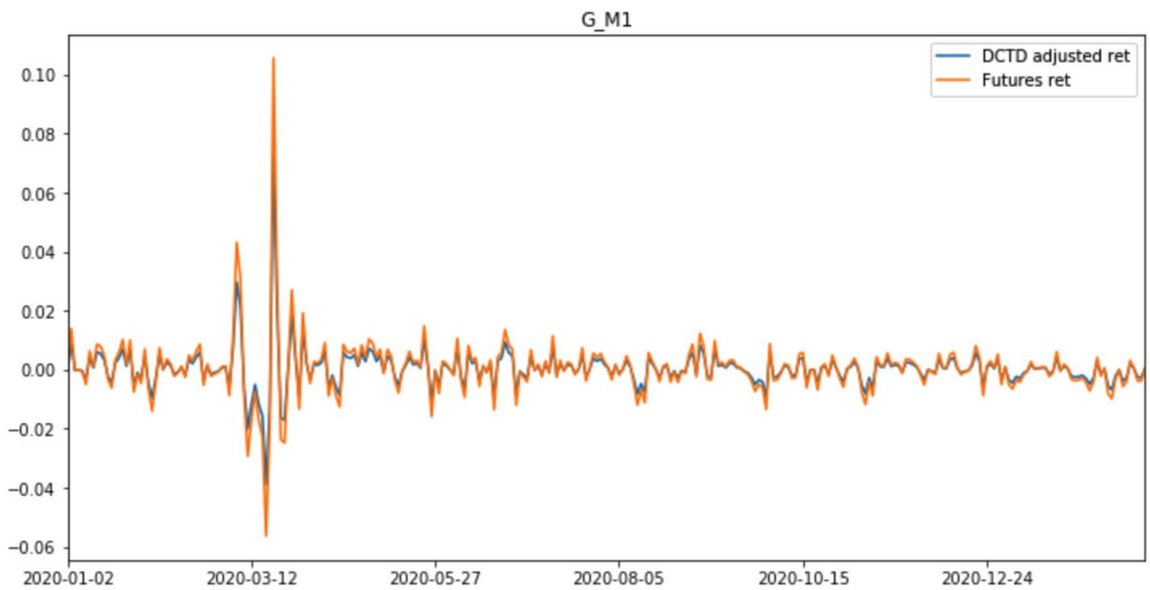

Figure: G_M1 DCTD adjusted return vs futures contract return

Comments:

1) From the graph above, although, the range of DCTD adjusted return is a little narrower than actual future price return, we can see that the actual and computed daily rate of return has the same trend, and they seem to be highly positive correlated. Computing their correlation, we find out that the adjusted return is highly correlated with future contract prices.

```
USM1_rets.cov()
```

	DCTD adjusted ret	Futures ret
DCTD adjusted ret	0.000025	0.000054
Futures ret	0.000054	0.000123

```
TYM1_rets.corr()
```

	DCTD adjusted ret	Futures ret
DCTD adjusted ret	1.0	1.0
Futures ret	1.0	1.0

```
G_M1_rets.corr()
```

	DCTD adjusted ret	Futures ret
DCTD adjusted ret	1.0	1.0
Futures ret	1.0	1.0

2) From the above study, the DCTD adjusted return can be viewed as a linear relationship with actual future price return. Therefore, we can use DCTD adjusted return to predict the trend of future price movement.