In [1]:
```
from functions import *
```

Reading the future price file, naming the futures as per month, extracting the price information

In [2]:
```
filename = "new_file.xlsx"
initial_investment = 1000000
bid_ask = 0


file = preprocess_future_data(filename,initial_investment, bid_ask)
```

Finding the expiry date for each future

In [3]:
```
roll_start_day = 5
expiry = create_expiry_schedule(file, roll_start_day)
```

Creating a futures df, assigning price to each future

In [4]:
```
future = create_future_df(file,expiry)
```

In [5]:
```
future = fill_future_current_value(future, file, expiry)
```

In [6]:
```
future.head(5)
```

Out[6]:

	Dates	Future	Current Future	Next Future	Next Future current val	Number of contracts	Contract Value
0	2000-01-03	SPH00 index	1466.8	1466.8	1485.7	NaN	NaN
1	2000-01-04	SPH00 index	1411.8	1411.8	1430.0	NaN	NaN
2	2000-01-05	SPH00 index	1413.5	1413.5	1431.8	NaN	NaN
3	2000-01-06	SPH00 index	1404	1404	1422.1	NaN	NaN
4	2000-01-07	SPH00 index	1460.5	1460.5	1479.3	NaN	NaN

Finding the number of contracts and contract value for each future (observe change in contract value at future expiry)

In [7]:
```
future = fill_future_contract_value(future, initial_investment, bid_ask)
```

In [8]:
```
future.to_csv("Outputs/Futures_pre_logic.csv")
```

Futures Roll Logic

In [4]:
```
future = pd.read_csv("Outputs/Futures_pre_logic.csv").drop(columns = 'Unnam
```

In [9]:
```
weights = [0.2,0.2,0.2,0.2,1]

for i in range(len(expiry)):
    if expiry.loc[i,"Expiry_date"] != expiry.loc[i,"roll_start_date"]:
        roll_start_index = future[future.Dates == expiry.loc[i,"roll_start_
        roll_df = future[roll_start_index:roll_start_index+5]
        roll(roll_df,future, weights)
```

In [10]:
```
future
```

Out[10]:

	Dates	Future	Current Future	Next Future	Next Future current val	Number of contracts	Contract Value
0	2000-01-03	SPH00 index	1466.8	1466.8	1485.7	2.72702	1000000
1	2000-01-04	SPH00 index	1411.8	1411.8	1430.0	2.72702	962503
2	2000-01-05	SPH00 index	1413.5	1413.5	1431.8	2.72702	963662
3	2000-01-06	SPH00 index	1404	1404	1422.1	2.72702	957186
4	2000-01-07	SPH00 index	1460.5	1460.5	1479.3	2.72702	995705
...	...	...	...	...	...	...	...
5257	2020-09-21	SPZ20 index	3275.1	3275.1	3275.1	2.63406	2.15671e+06
5258	2020-09-22	SPZ20 index	3299.3	3299.3	3299.3	2.63406	2.17264e+06
5259	2020-09-23	SPZ20 index	3231.2	3231.2	3231.2	2.63406	2.1278e+06
5260	2020-09-24	SPZ20 index	3238	3238	3238.0	2.63406	2.13227e+06
5261	2020-09-25	SPZ20 index	NaN	NaN	0.0	2.63406	NaN

5262 rows × 7 columns

In [89]:
```
roll_df
```

Out[89]:

	Unnamed: 0	Dates	Future	Current Future	Next Future	Next Future current val	Number of contracts	Contract Value
5250	5250	2020-09-10	SPU20 index	3340.6	3340.6	3330.1	2.626137	2.193218e+06
5251	5251	2020-09-11	SPU20 index	3333.7	3333.7	3323.3	2.626137	2.188698e+06
5252	5252	2020-09-14	SPU20 index	3382.4	3382.4	3372.2	2.626137	2.220755e+06
5253	5253	2020-09-15	SPU20 index	3405.3	3405.3	3395.0	2.626137	2.235781e+06
5254	5254	2020-09-16	SPU20 index	3389.6	3379.4	3379.4	2.626137	2.225493e+06

In [10]:
```
# Before
future[5251:]["Contract Value"]
```

Out[10]:
```
5251    2.18869e+06
5252    2.22066e+06
5253     2.2357e+06
5254    2.22539e+06
5255    2.20669e+06
5256    2.18377e+06
5257    2.15671e+06
5258    2.17264e+06
5259     2.1278e+06
5260    2.13227e+06
5261            NaN
Name: Contract Value, dtype: object
```

In [11]:
```
# After
future[5251:]["Contract Value"]
```

Out[11]:
```
5251     2.1887e+06
5252    2.22076e+06
5253    2.23578e+06
5254    2.22549e+06
5255    2.20669e+06
5256    2.18377e+06
5257    2.15671e+06
5258    2.17264e+06
5259     2.1278e+06
5260    2.13227e+06
5261            NaN
Name: Contract Value, dtype: object
```

Start from here

In [24]:
```
# future = pd.read_excel("future_post_contract_value_roll.xlsx")
# future = future.drop(columns = ["Unnamed: 0"])
```

In [26]:
```
future[420:440]
```

Out[26]:

	Dates	Future	Current Future	Next Future	Next Future current val	Number of contracts	Contract Value
420	2001-08-20	SPU01 index	1175.7	1175.7	1182.5	2.52935	743439
421	2001-08-21	SPU01 index	1156.3	1156.3	1162.8	2.52935	731172
422	2001-08-22	SPU01 index	1167.2	1167.2	1173.9	2.52935	738064
423	2001-08-23	SPU01 index	1163.5	1163.5	1170.1	2.52935	735725
424	2001-08-24	SPU01 index	1187.5	1187.5	1194.4	2.52935	750901
425	2001-08-27	SPU01 index	1180.5	1180.5	1187.4	2.52935	746474
426	2001-08-28	SPU01 index	1163.6	1163.6	1170.3	2.52935	735788
427	2001-08-29	SPU01 index	1152.4	1152.4	1158.9	2.52935	728706
428	2001-08-30	SPU01 index	1129.3	1129.3	1135.6	2.52935	714099
429	2001-08-31	SPU01 index	1135.1	1135.1	1141.5	2.52935	717766
430	2001-09-03	SPU01 index	1134.7	1134.7	1141.5	2.52935	717513
431	2001-09-04	SPU01 index	1131	1131	1137.4	2.52935	715174
432	2001-09-05	SPU01 index	1133.5	1133.5	1140.0	2.52935	716755
433	2001-09-06	SPU01 index	1103.5	1103.5	1109.6	2.52935	697784
434	2001-09-07	SPU01 index	1083.3	1083.3	1088.9	2.52935	685011
435	2001-09-10	SPU01 index	1095.7	1095.7	1101.4	2.52935	692857
436	2001-09-11	SPU01 index	1095.7	1095.7	1101.4	2.52935	692857
437	2001-09-17	SPU01 index	1039	1039	1043.1	2.52935	656602
438	2001-09-18	SPU01 index	1035.5	1039.5	1039.5	2.52935	654358
439	2001-09-19	SPZ01 index	1016.8	1016.8	1019.8	2.51962	640487

Loading SPX and Tbill data and merging it with futures file(Keeping all dates from SPX)

In [14]:
```
spx_loc = "SPX.xlsx"
t_bill_loc = "tbill.xlsx"

spx = pd.read_excel(spx_loc)
spx["Dates"] = pd.to_datetime(spx["Dates"])
future = future.merge(spx,on = "Dates", how = "right")

tbill = pd.read_excel(t_bill_loc)
tbill["Dates"] = pd.to_datetime(tbill["Dates"])
future = future.merge(tbill,on = "Dates", how = "left")

future["SPX_return"] = future["SPX_return"]/100
future["Tbill_returns"] = round(future["T-Bill price"].pct_change(),6)
future["Future_returns"] = round(future["Contract Value"].pct_change(),10)


# Remove this step if nan values are not there at end.
future_full_backup = future
future = future[0:5211]

#future = future.fillna(method='ffill')


future = future.drop(index = future[future.Dates == "2001-09-11"].index)

future = future.reset_index()
future[future.Dates == "2001-09-11"]
```

Out[14]:

index	Dates	Future	Current Future	Next Future	Next Future current val	Number of contracts	Contract Value	SPX	SPX_return	T-Bill price	Tbill

Calculating the cumulative returns difference of SPX vs (Futures+Tbill) portfolio.

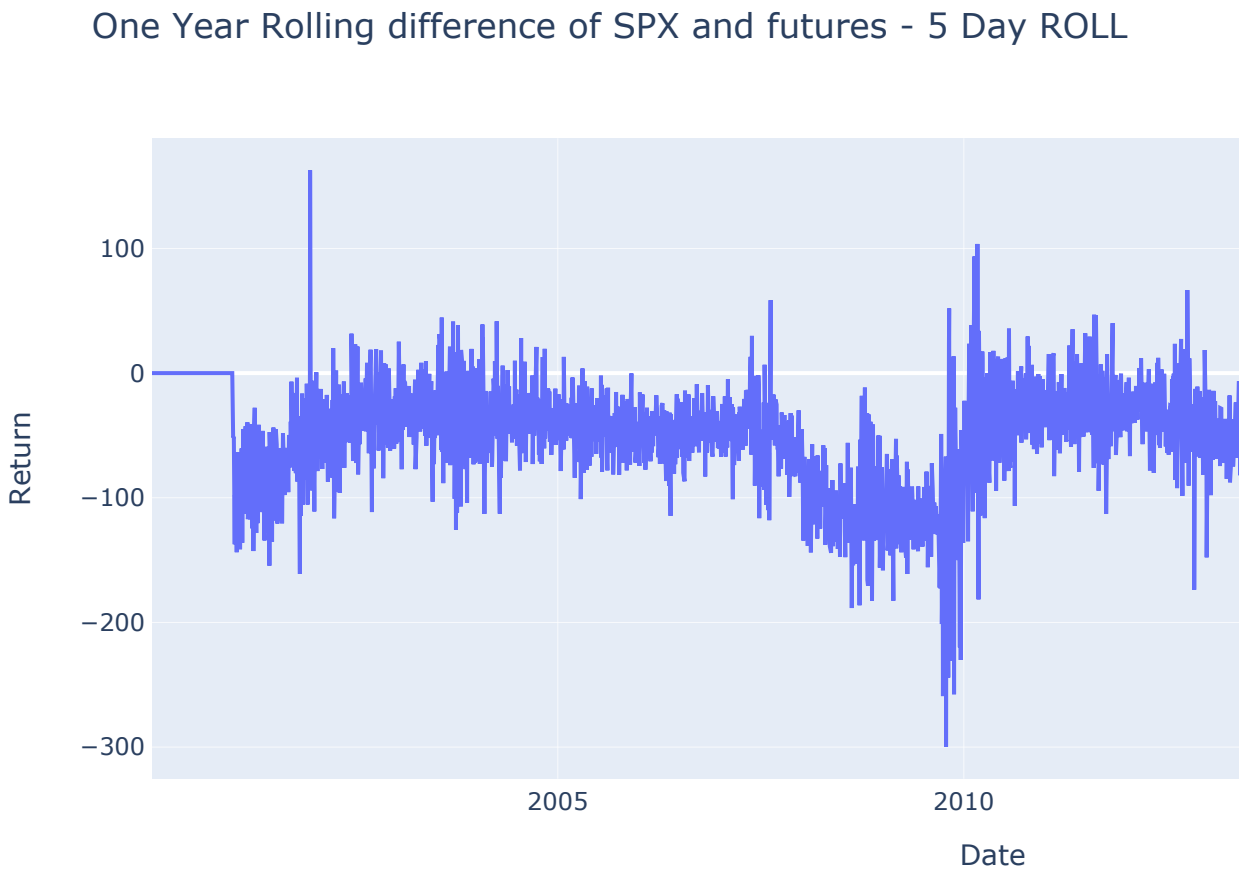
In [17]:
```
future["Cum_spx_cash"] = 1
future["Cum_fut_tbill_cash"] = 1

for i in range(len(future)-1):
    future.loc[i+1,"Cum_spx_cash"] = round(future.loc[i,"Cum_spx_cash"]* (1
    future.loc[i+1,"Cum_fut_tbill_cash"] = round(future["Cum_fut_tbill_cash
                                                 (1+future["Future_returns"


future["Cum_SPX_future_diff"] =  future["Cum_fut_tbill_cash"] - future["Cum


#Cum SPX is exactly same
#Cum tbill wala is little different
```

In [19]:
```
future.to_excel("Outputs/futures_with_returns.xlsx")
```

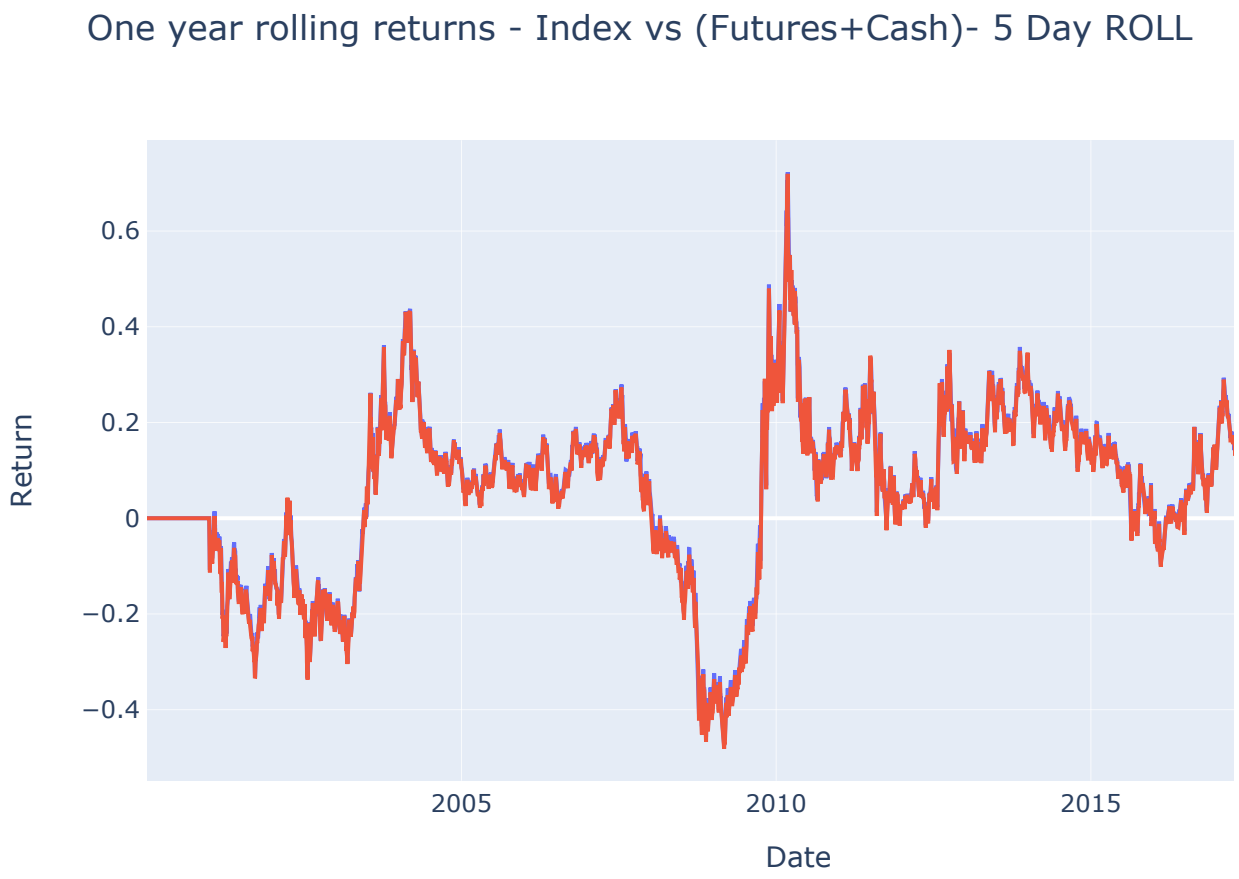
Calculating 1yr rolling returns (Annualized returns)

In [20]:
```
def rolling(new_col, return_col ,ann_freq , rolling_no_days):
    future[new_col] = 0
    for i in range(rolling_no_days,len(future)):
        future.loc[i,new_col] = ((future.loc[i,return_col]/future.loc[i-rol

rolling("1yr_rolling_return_cum_spx","Cum_spx_cash",1,252)
rolling("1yr_rolling_return_Cum_fut_tbill","Cum_fut_tbill_cash",1,252)
future["1yr_rolling_return_Cum_diff"] = 10000* (future["1yr_rolling_return_
```

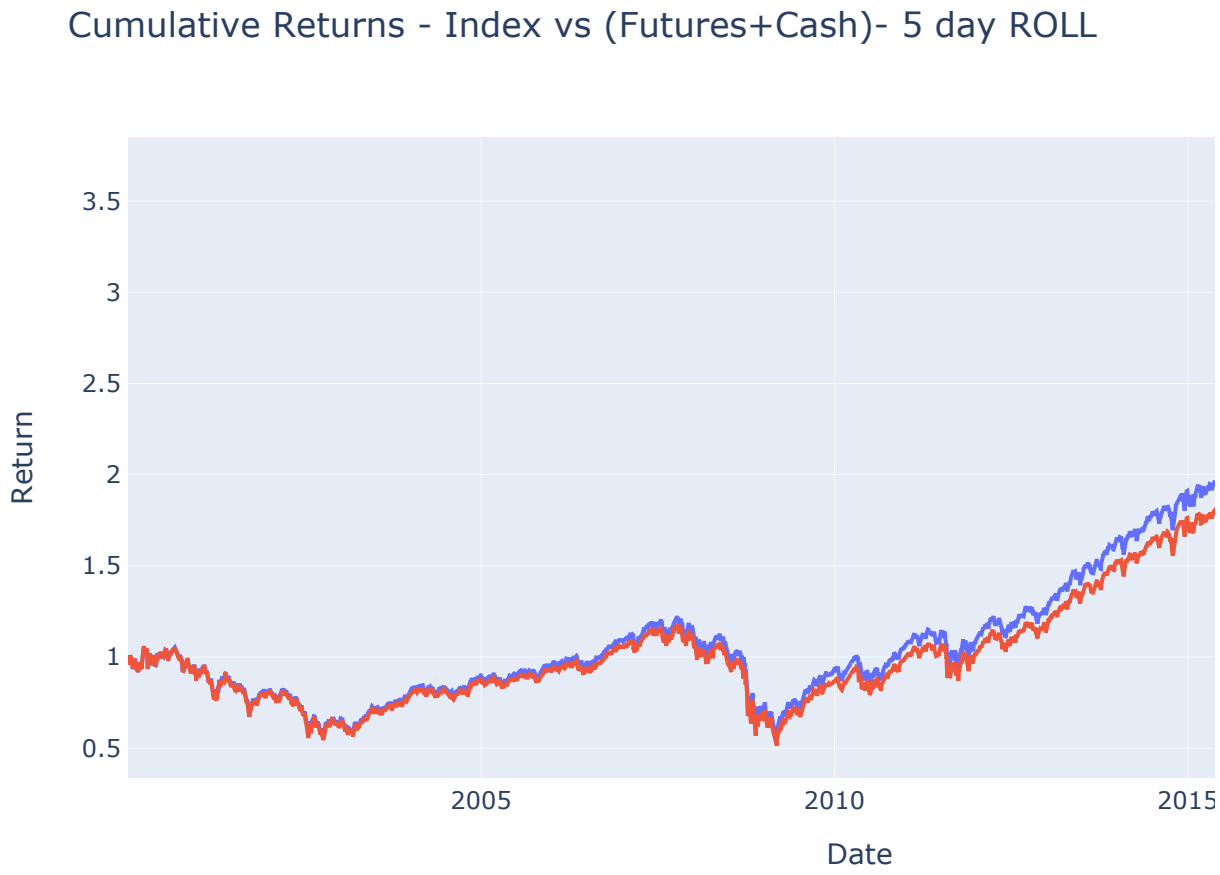
In [21]:
```
title = "One Year Rolling difference of SPX and futures - 5 Day ROLL"
xlabel = 'Date'
ylabel = 'Return'
df = future
col = ["1yr_rolling_return_Cum_diff"]
data = []

#get_pyplot(col,title,xlabel,ylabel,df)
get_go_plotly(col,title,xlabel,ylabel,df)
```



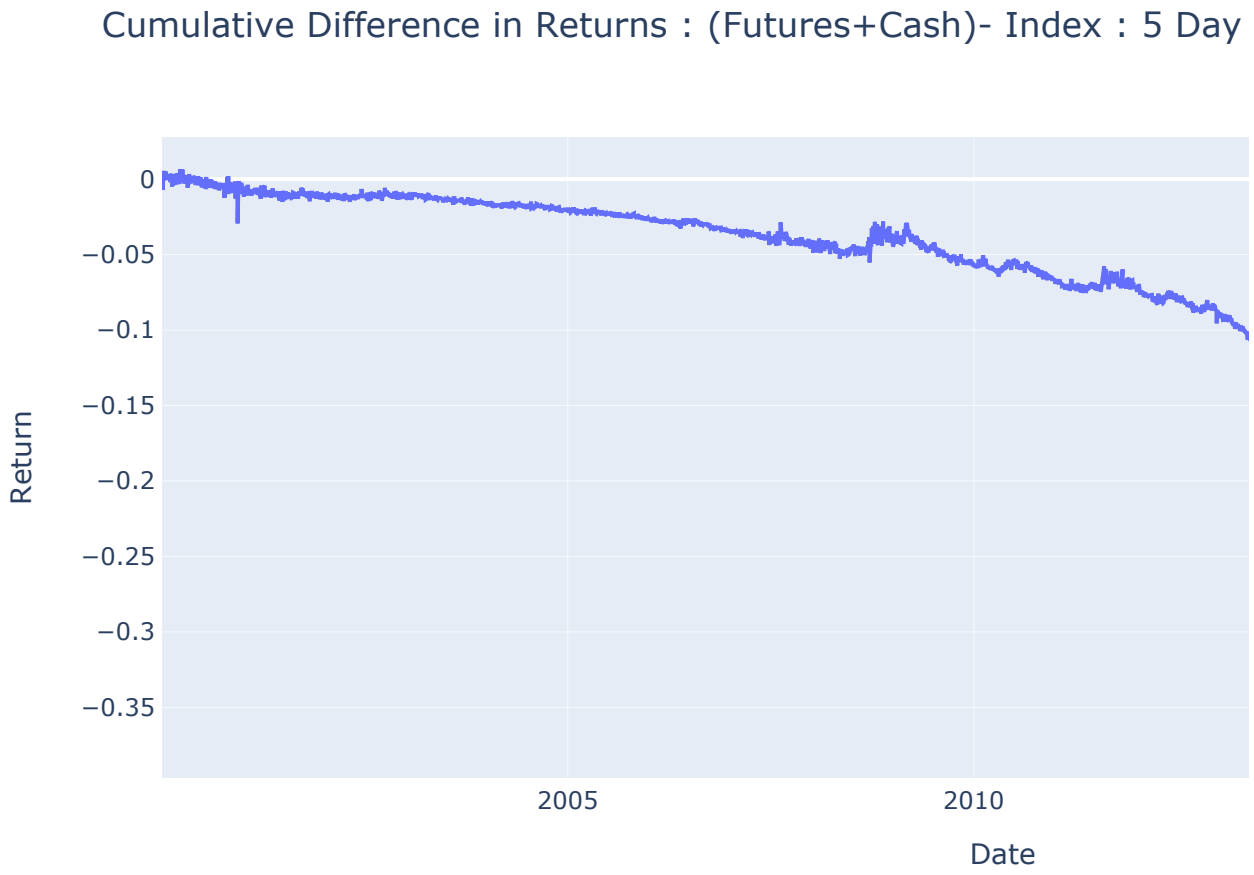
In [22]:
```
import matplotlib.pyplot as plt
title = "One year rolling returns - Index vs (Futures+Cash)- 5 Day ROLL"
xlabel = 'Date'
ylabel = 'Return'
df = future
col = ["1yr_rolling_return_cum_spx", "1yr_rolling_return_Cum_fut_tbill"]


#get_pyplot(col,title,xlabel,ylabel,df)
get_go_plotly(col,title,xlabel,ylabel,df)
```



In [23]:
```
title = "Cumulative Returns - Index vs (Futures+Cash)- 5 day ROLL"
xlabel = 'Date'
ylabel = 'Return'
df = future
col = ["Cum_spx_cash", "Cum_fut_tbill_cash"]

get_go_plotly(col,title,xlabel,ylabel,df)
```



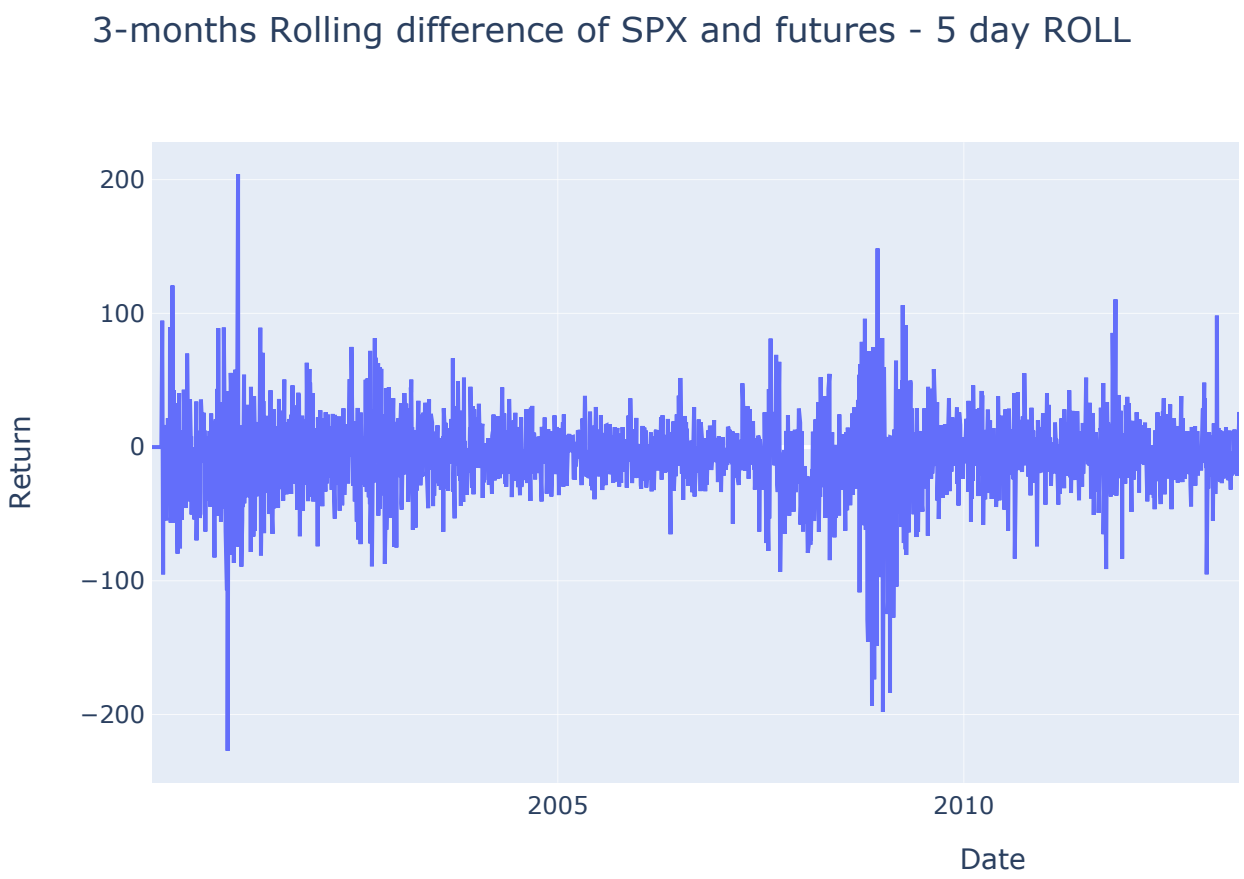
In [25]:
```
title = "Cumulative Difference in Returns : (Futures+Cash)- Index : 5 Day R
xlabel = 'Date'
ylabel = 'Return'
df = future
col = ["Cum_SPX_future_diff"]

get_go_plotly(col,title,xlabel,ylabel,df)
```



3 month rolling difference

In [26]:
```
rolling("3m_rolling_return_cum_spx","Cum_spx_cash",1,30)
rolling("3m_rolling_return_Cum_fut_tbill","Cum_fut_tbill_cash",1,30)
future["3m_rolling_return_Cum_diff"] = 10000*(future["3m_rolling_return_Cum
```

In [27]:
```
title = "3-months Rolling difference of SPX and futures - 5 day ROLL"
xlabel = 'Date'
ylabel = 'Return'
df = future
col = ["3m_rolling_return_Cum_diff"]
data = []

#get_pyplot(col,title,xlabel,ylabel,df)
get_go_plotly(col,title,xlabel,ylabel,df)
```

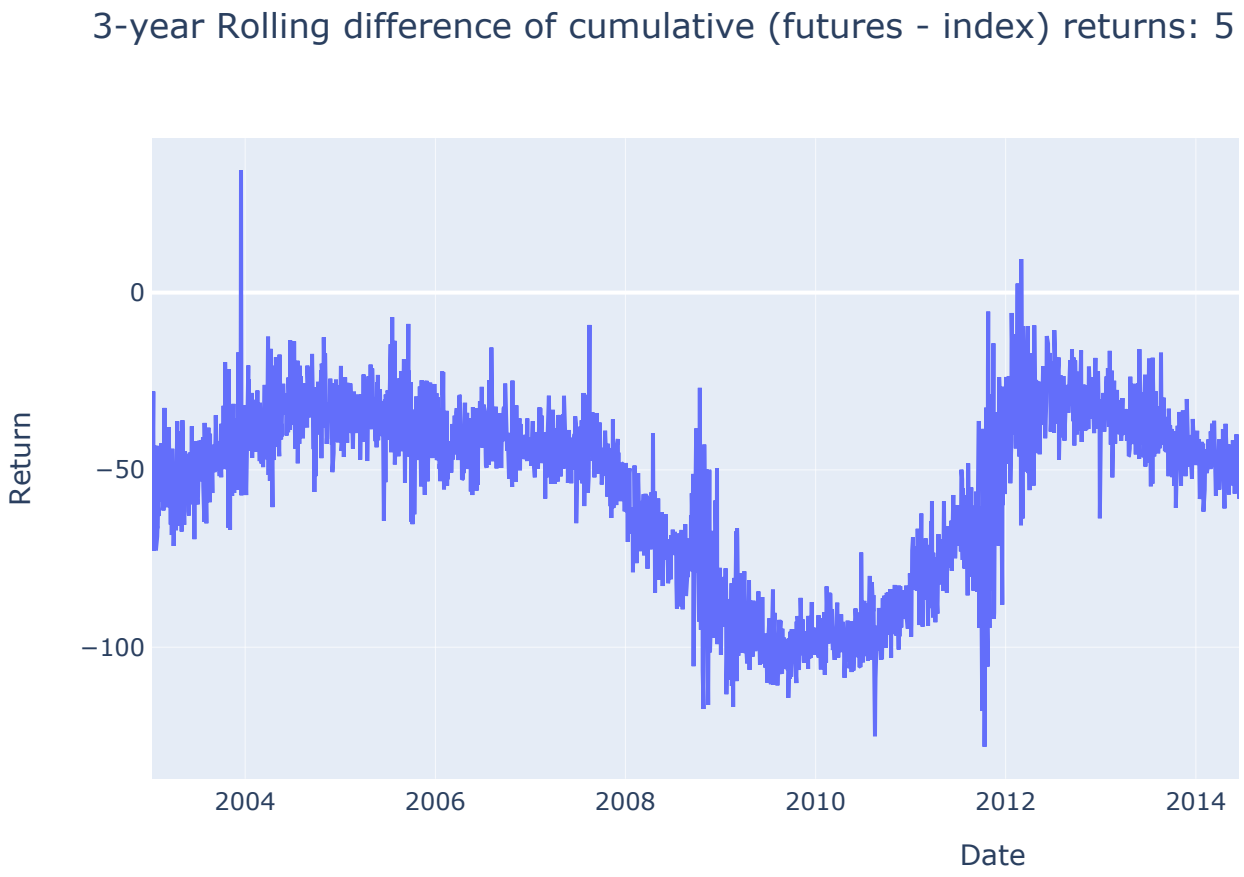


3yr rolling difference

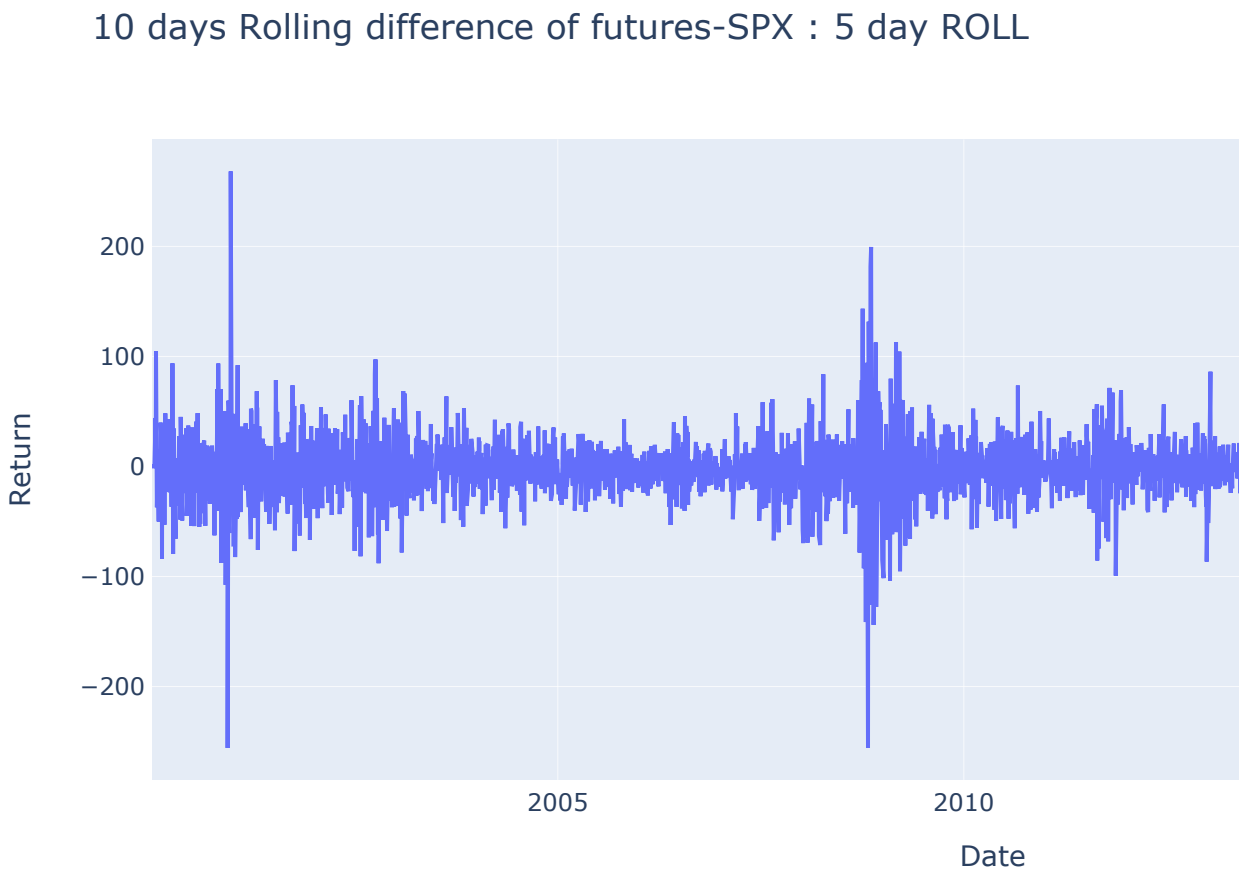
In [28]:
```
rolling("3year_rolling_return_cum_spx","Cum_spx_cash",3,252*3)
rolling("3year_rolling_return_Cum_fut_tbill","Cum_fut_tbill_cash",3,252*3)
future["3year_rolling_return_Cum_diff"] = 10000*(future["3year_rolling_retu
```

In [45]:
```
future[["Dates","3year_rolling_return_cum_spx","3year_rolling_return_Cum_fut
```

In [30]:
```
title = "3-year Rolling difference of cumulative (futures - index) returns:
xlabel = 'Date'
ylabel = 'Return'
df = future[756:]
col = ["3year_rolling_return_Cum_diff"]
data = []

#get_pyplot(col,title,xlabel,ylabel,df)
get_go_plotly(col,title,xlabel,ylabel,df)
```



In [33]:
```
future = pd.read_excel("Output_5Day ROLL /futures_post_rolling_diff.xlsx")
rolling("10d_rolling_return_cum_spx","Cum_spx_cash",1,10)
rolling("10d_rolling_return_Cum_fut_tbill","Cum_fut_tbill_cash",1,10)
future["10d_rolling_return_Cum_diff"] = 10000*(future["10d_rolling_return_C

title = "10 days Rolling difference of futures-SPX : 5 day ROLL"
xlabel = 'Date'
ylabel = 'Return'
df = future
col = ["10d_rolling_return_Cum_diff"]
data = []

#get_pyplot(col,title,xlabel,ylabel,df)
get_go_plotly(col,title,xlabel,ylabel,df)
```



Annualized returns table

In [37]:
```
future = future.set_index(future.Dates)
annual_table = pd.DataFrame( index = ["Start Period", "End Period",
                                      "No.of yrs", "SPX return" ,
                                      "Futures+Cash return","Diff (bps)" ])
print("Annualized 1yr Returns - 5 Day ROLL")
year = 2020
day = 15
month = 9

for i in [1,5,10,20]:
    annualization(year,month,day ,i, annual_table, "Cum_spx_cash","Cum_fut_


annual_table
```

```
Annualized 1yr Returns - 5 Day ROLL
```

Out[37]:

	annual_1_yrs	annual_5_yrs	annual_10_yrs	annual_20_yrs
Start Period	2019-09-16	2015-09-17	2010-09-17	2000-09-20
End Period	2020-09-15	2020-09-15	2020-09-15	2020-09-15
No.of yrs	1	5	10	20
SPX return	0.156349	0.135895	0.140286	0.0642439
Futures+Cash return	0.154945	0.131504	0.135663	0.0589201
Diff (bps)	-14.0376	-43.9066	-46.23	-53.2387

Average Rolling returns

In [39]:
```
rolling_avg_1yr = pd.DataFrame( index = ["Start Period", "End Period",
                                         "No.of yrs", "SPX return" ,
                                         "Futures+Cash return","Diff (bps)" ])

print("Average of Rolling 1yr Returns - 5 Days ROLL")
year = 2020
day = 15
month = 9

for i in [1,5,10,20]:
    rolling_avg_returns(year,month,day ,i, rolling_avg_1yr, "1yr_rolling_re


rolling_avg_1yr
```

```
Average of Rolling 1yr Returns - 5 Days ROLL
```

Out[39]:

	Rolling Avg1_yrs	Rolling Avg5_yrs	Rolling Avg10_yrs	Rolling Avg20_yrs
Start Period	2019-09-16	2015-09-17	2010-09-17	2000-09-20
End Period	2020-09-15	2020-09-15	2020-09-15	2020-09-15
No.of yrs	1	5	10	20
SPX return	0.121377	0.110641	0.134545	0.0740629
Futures+Cash return	0.117616	0.105185	0.129589	0.0687599
Diff (bps)	-37.6052	-54.5604	-49.5597	-53.0304

In [41]:
```
rolling_avg_3yr = pd.DataFrame( index = ["Start Period", "End Period",
                                         "No.of yrs", "SPX return" ,
                                         "Futures+Cash return","Diff (bps)" ])

print("Average of Rolling 3yr Returns - 5 Days ROLL")
year = 2020
day = 15
month = 9

for i in [1,5,10,20]:
    rolling_avg_returns(year,month,day ,i, rolling_avg_3yr, "3year_rolling_


rolling_avg_3yr
```

```
Average of Rolling 3yr Returns - 5 Days ROLL
```

Out[41]:

	Rolling Avg1_yrs	Rolling Avg5_yrs	Rolling Avg10_yrs	Rolling Avg20_yrs
Start Period	2019-09-16	2015-09-17	2010-09-17	2000-09-20
End Period	2020-09-15	2020-09-15	2020-09-15	2020-09-15
No.of yrs	1	5	10	20
SPX return	0.120433	0.119647	0.124131	0.0677305
Futures+Cash return	0.114415	0.113946	0.118725	0.0628157
Diff (bps)	-60.1753	-57.0026	-54.0606	-49.1485

In [ ]:
```
```