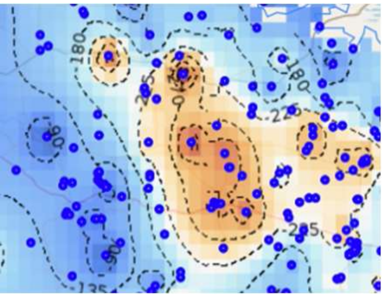
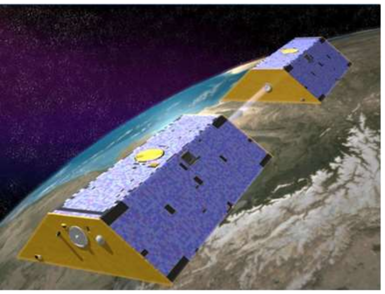
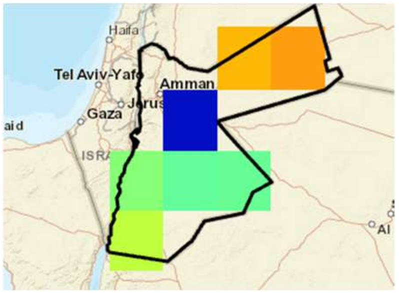
Imputing gaps in the GRACE GWSa time series

Use of the Gravity Recovery and Climate Experiment (GRACE) mission to monitor groundwater storage change: National workshop for Jordan and State of Palestine

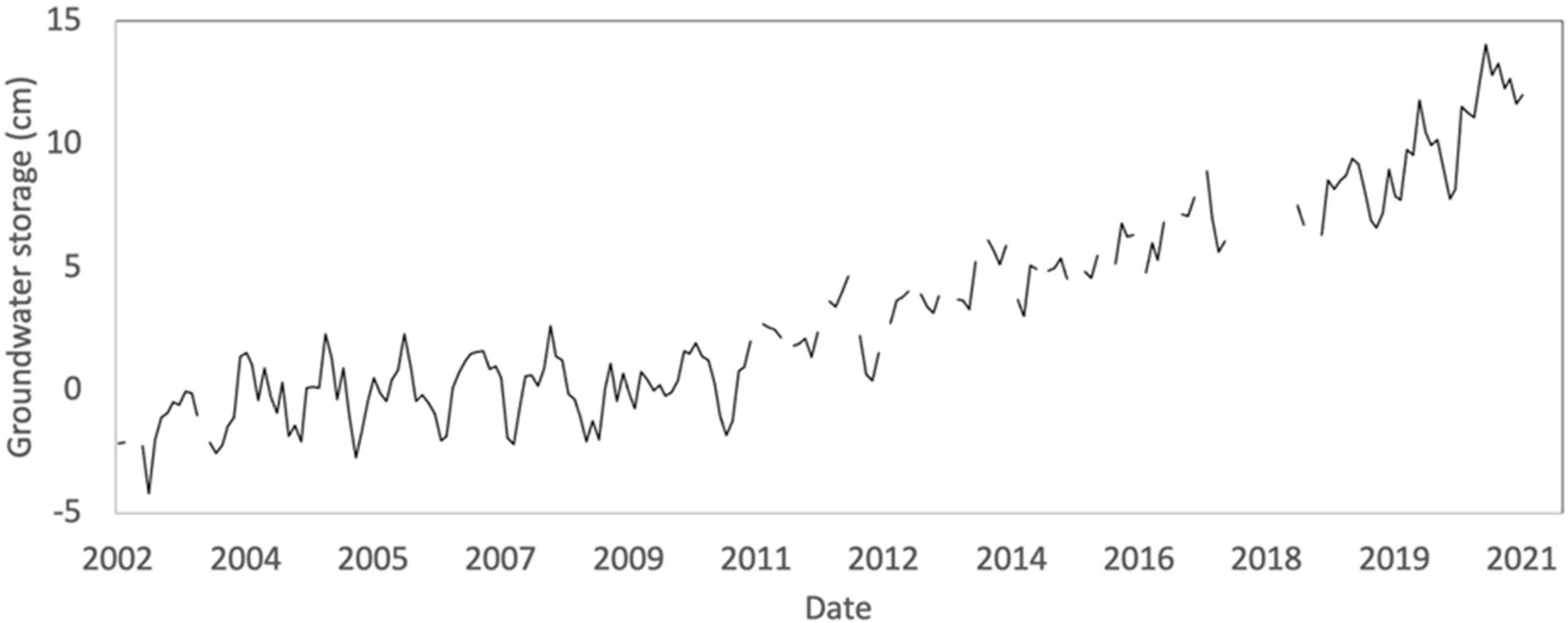
Amman Jordan, February 25-26



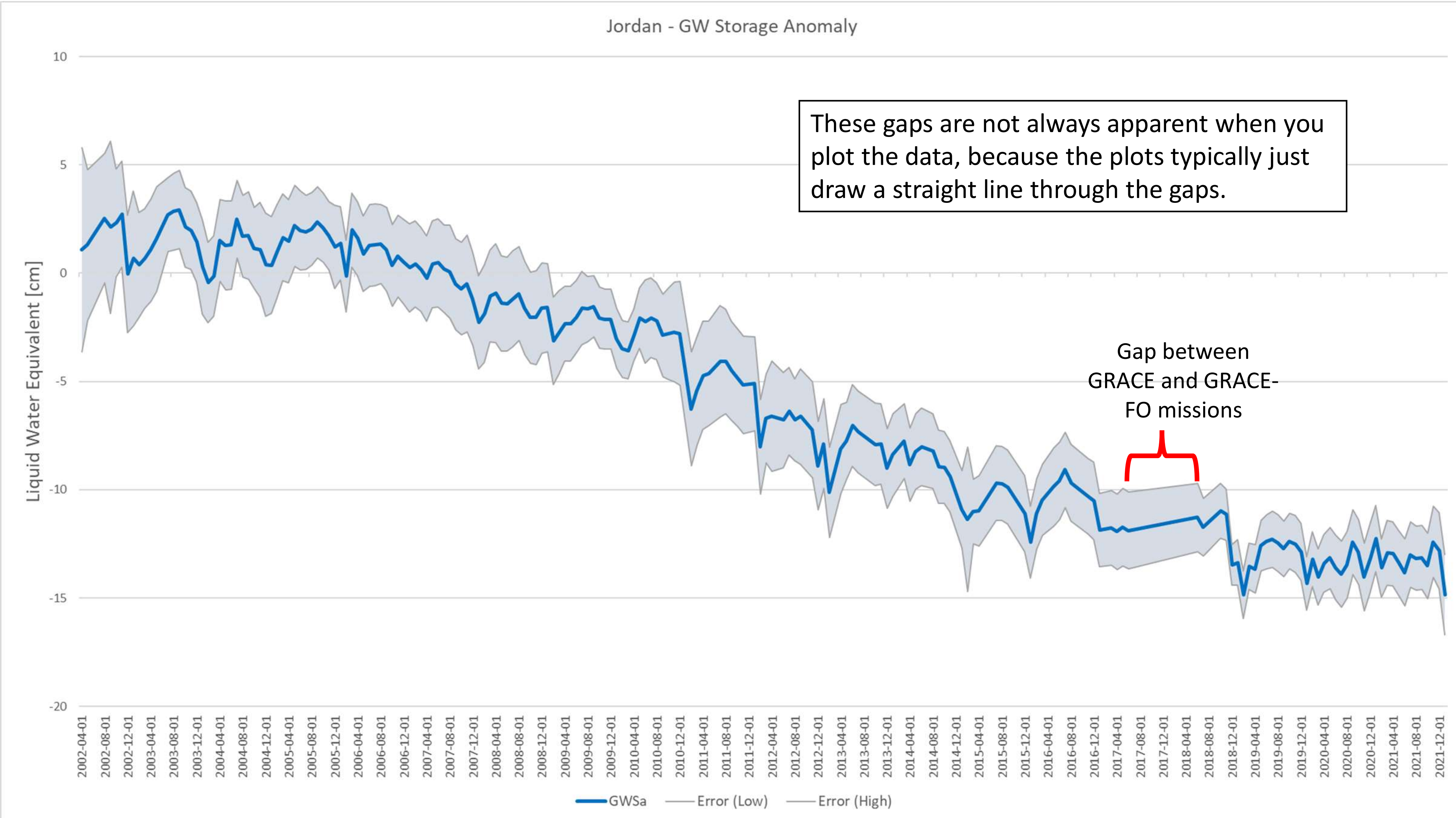




The Problem



Due to technical problems with the GRACE satellites and a one-year gap from 2017-2018, the GRACE data includes several gaps ranging from one month to several months. Before estimating recharge using the Water Table Fluctuation method, it helps to impute these gaps to get a more complete record.

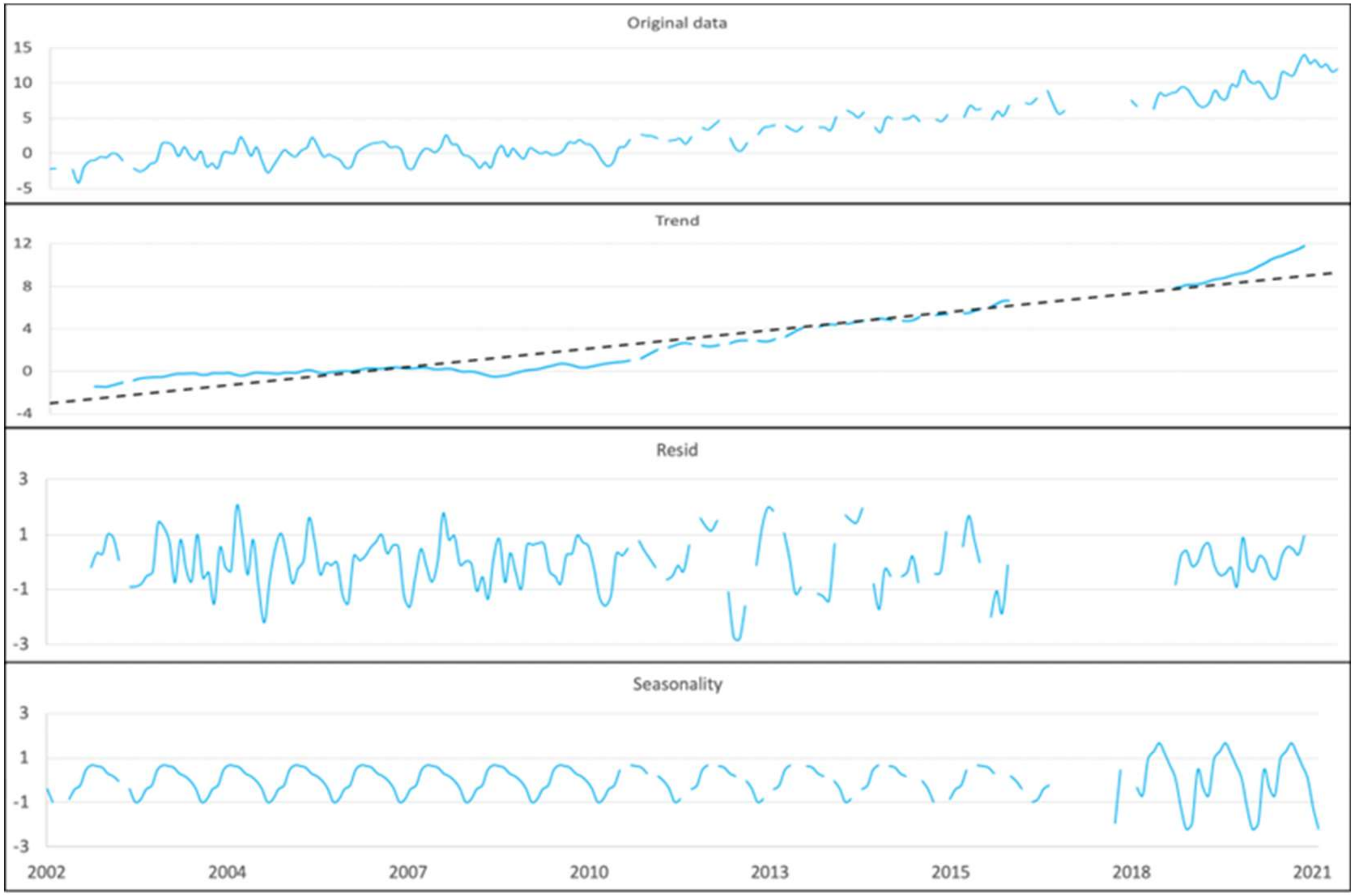


Jordan - GW Storage Anomaly
Liquid Water Equivalent [cm]
10
5
0
-5
-10
-15
-20
These gaps are not always apparent when you plot the data, because the plots typically just draw a straight line through the gaps.
Gap between GRACE and GRACE-FO missions
GWSa
Error (Low)
Error (High)

To impute the gaps, we use a simple seasonal decomposition model implemented in the **statsmodels** Python package. With this approach, we decompose the GWSa time series into three components: the trend, the seasonal, and the random components:

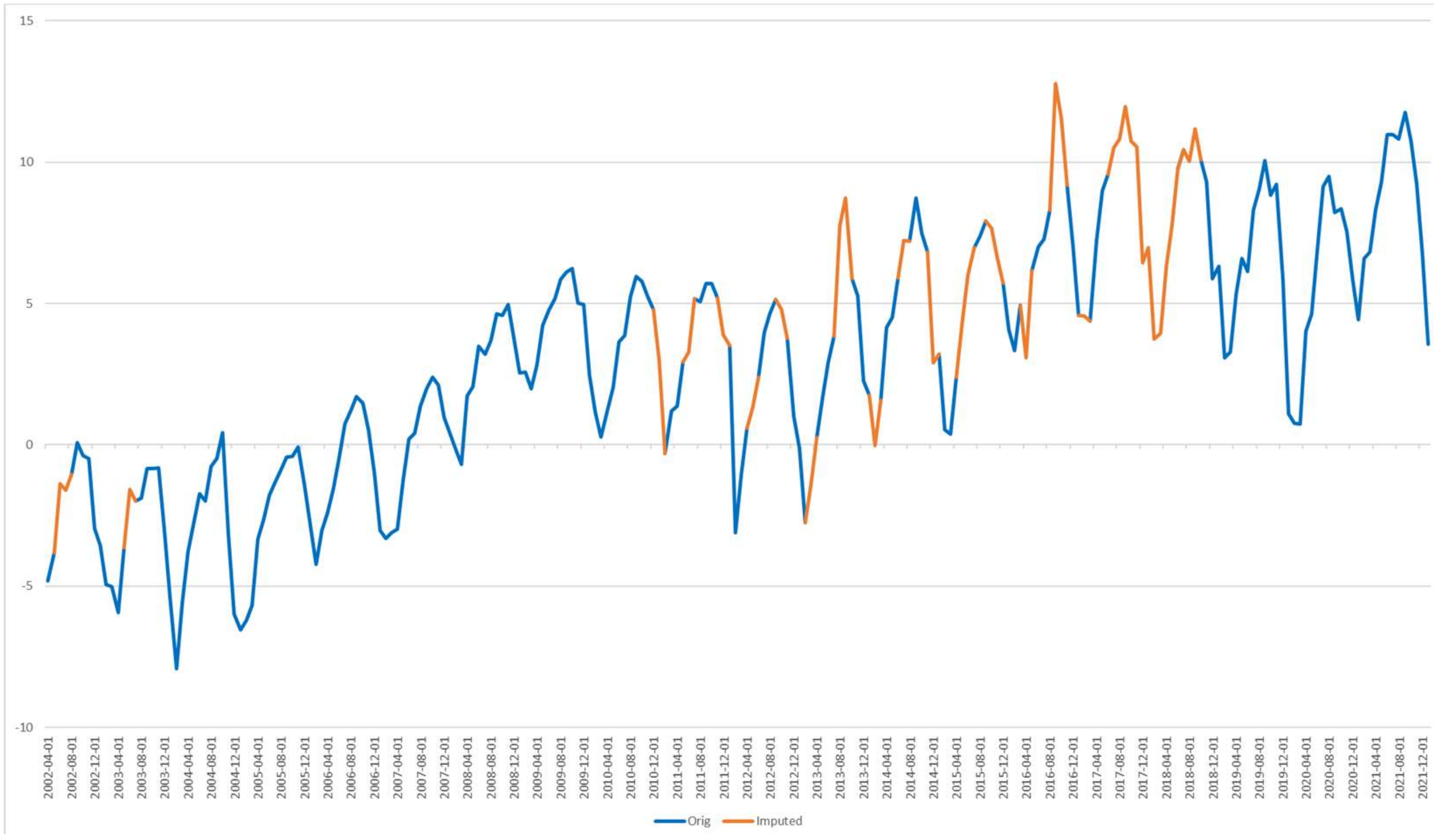
$$Y[t] = T[t] + S[t] + e[t]$$

Where Y[t] is the GWSa, T[t] is the GWSa trend, S[t] is the seasonal GWSa component, and e[t] is the residual GWSa component.



To impute the missing data, we use the trend from the data decomposition, then add the average of the monthly and residual values for that month to estimate the missing value. This model can be written as:

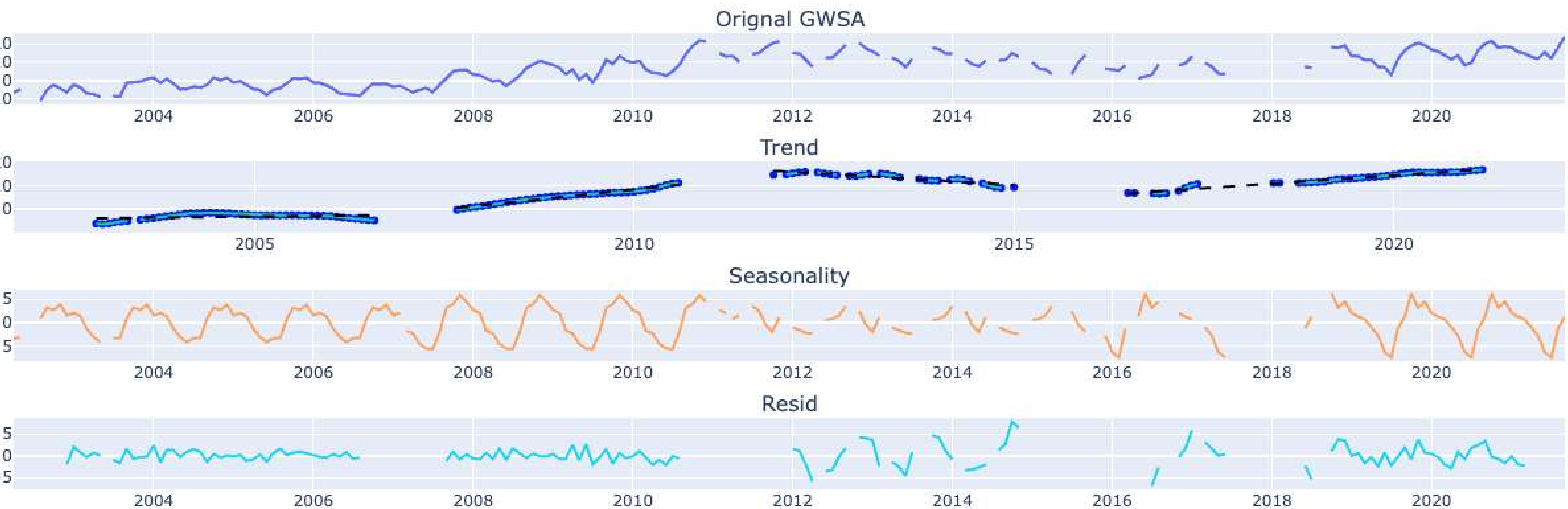
$$Y[t] = y(T[t]) + \overline{S[t] + e[t]}$$



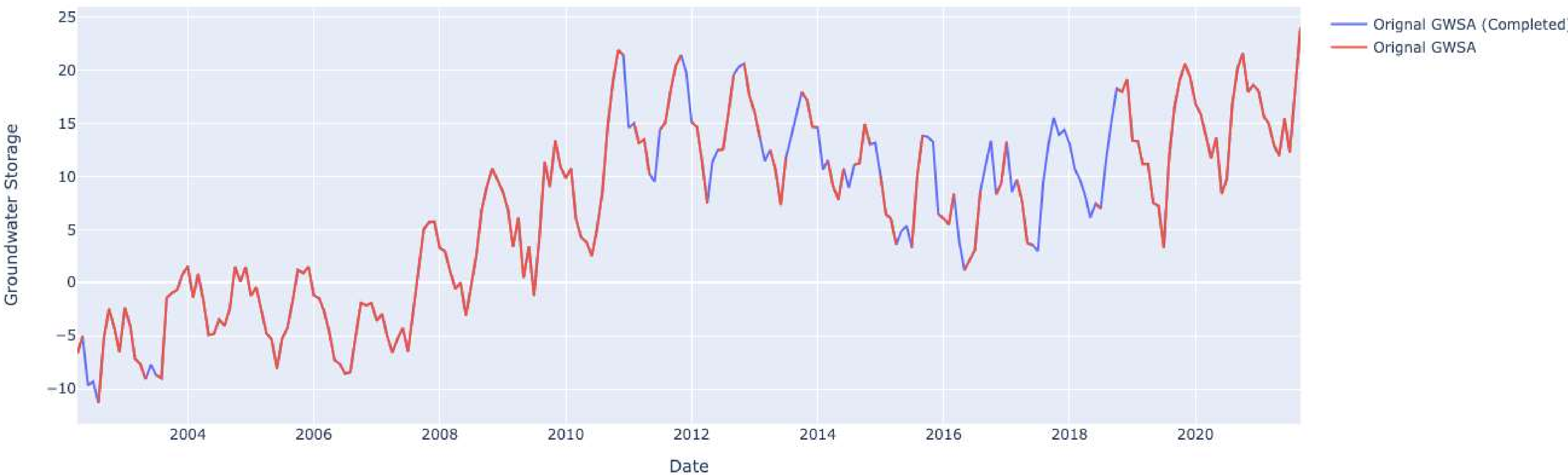
Multi-Linear Trend Analysis

In the seasonal decomposition method described above for gap imputation, a single linear trend was described. Here is the trend resulting from a sample dataset fitted with a single trend line:

For the sample dataset, there are four distinct trends. The Python script has an option to perform a multi-linear regression analysis. For this dataset, we set the **number_breakpoints** variable to 3, and run a multi-linear regression algorithm that fits the data as follows:



Resulting in the following imputation:



Prior to running the gap imputation script, we must first create a two-column CSV file.

	A	B	C	D	E
1		date	ts	error_min	error_max
2	0	4/1/2002	-6.66	-18.978	5.658
3	1	5/1/2002	-5.033	-11.749	1.683
4	2	8/1/2002	-11.323	-15.81	-6.835
5	3	9/1/2002	-5.273	-11.971	1.426
6	4	10/1/2002	-2.464	-5.961	1.034
7	5	11/1/2002	-4.24	-8.214	-0.267
8	6	12/1/2002	-6.551	-10.684	-2.419
9	7	1/1/2003	-2.345	-7.653	2.962
10	8	2/1/2003	-4.075	-8.033	-0.118
11	9	3/1/2003	-7.156	-10.076	-4.237
12	10	4/1/2003	-7.638	-10.015	-5.261
13	11	5/1/2003	-9.063	-11.364	-6.762
14	12	7/1/2003	-8.686	-10.879	-6.494
15	13	8/1/2003	-9.002	-10.775	-7.229
16	14	9/1/2003	-1.429	-4.881	2.022
17	15	10/1/2003	-0.977	-7.424	5.47
18	16	11/1/2003	-0.676	-7.373	6.021
19	17	12/1/2003	0.771	-4.438	5.979
20	18	1/1/2004	1.561	-1.312	4.435
21	19	2/1/2004	-1.426	-3.165	0.313
22	20	3/1/2004	0.829	-0.689	2.348
23	21	4/1/2004	-1.735	-3.668	0.199

	A	B	C
1	date	ts	
2	4/1/2002	-6.66	
3	5/1/2002	-5.033	
4	8/1/2002	-11.323	
5	9/1/2002	-5.273	
6	10/1/2002	-2.464	
7	11/1/2002	-4.24	
8	12/1/2002	-6.551	
9	1/1/2003	-2.345	
10	2/1/2003	-4.075	
11	3/1/2003	-7.156	
12	4/1/2003	-7.638	
13	5/1/2003	-9.063	
14	7/1/2003	-8.686	
15	8/1/2003	-9.002	
16	9/1/2003	-1.429	
17	10/1/2003	-0.977	
18	11/1/2003	-0.676	
19	12/1/2003	0.771	
20	1/1/2004	1.561	
21	2/1/2004	-1.426	

1) Copy the date and gw storage columns to a new workbook.

2) Export the workbook to a CSV file ("dsb-gwsa-raw-clean.csv" for example).

impute_gaps_GRACE.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

RAM Disk Colab AI Share

[18]

```
    if ((c == a[0]).all()):
      d = globals()['seasonal_mean_{}'.format(i+1)].index == mes
      if ((d == b[0]).all()):
        globals()['grace_df_subset_completed_{}'.format(i+1)].loc[globals()['grace_df_subset_completed_{}'.format(i+1)].index == date] = (seasonal_mean_df[seasonal_mean_df.index == mes].values[0][0]
      else:
        globals()['grace_df_subset_completed_{}'.format(i+1)].loc[globals()['grace_df_subset_completed_{}'.format(i+1)].index == date] = (globals()['seasonal_mean_{}'.format(i+1)][globals()['seasona
    else:
      globals()['grace_df_subset_completed_{}'.format(i+1)].loc[globals()['grace_df_subset_completed_{}'.format(i+1)].index == date] = (globals()['seasonal_mean_df_{}'.format(i+1)][globals()['seasona

grace_df_total_v2 = grace_df_total_v2.fillna(globals()['grace_df_subset_completed_{}'.format(i+1)])

```

```
# Plot final results

total_series = go.Scatter(name='Orignal GWSA', x=grace_df_total.index, y=grace_df_total.iloc[:, 0].values)
completed_series = go.Scatter(name='Orignal GWSA (Completed)', x=grace_df_total_v2.index, y=grace_df_total_v2.iloc[:, 0].values)

plot_series = [completed_series, total_series]
layout = go.Layout(title="Orignal GWSA (Calculated)", xaxis=dict(title='Date',), yaxis=dict(title='Groundwater Storage',),)
chart_obj = go.Figure(data=plot_series, layout=layout)
chart_obj.show()

```

Scripts uploads CSV file and performs the gap imputation. The results can be downloaded as a new CSV file.

Questions?