```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import geopandas as gpd


from google.colab import files
uploaded = files.upload()
```

Choose Files Final.xlsx

- **Final.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 167856810 bytes, last modified: 7/21/2025 - 100% done

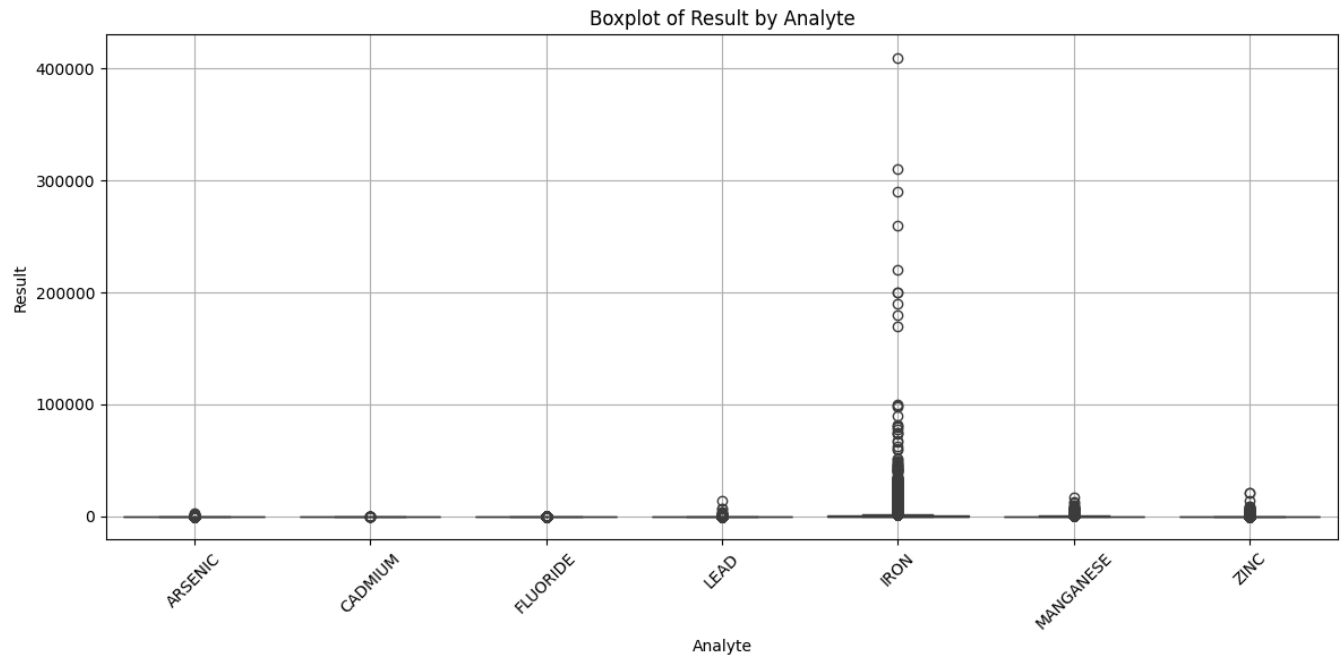
Saving Final.xlsx to Final.xlsx

```
df = pd.read_excel("Final.xlsx")

df.head()
```

	Row #	Regulating Agency	Water System #	System	Status	County	Service Connections	Population TINWSYS
0	1	DISTRICT 04 - SAN FRANCISCO	CA0103040	NORRIS CANYON PROPERTY OWNERS ASSN	A	ALAMEDA	19.0	50
1	2	DISTRICT 04 - SAN FRANCISCO	CA0103040	NORRIS CANYON PROPERTY OWNERS ASSN	A	ALAMEDA	19.0	50
2	3	DISTRICT 04 - SAN FRANCISCO	CA0103040	NORRIS CANYON PROPERTY OWNERS ASSN	A	ALAMEDA	19.0	50
3	4	DISTRICT 04 - SAN	CA0103040	NORRIS CANYON PROPERTY	A	ALAMEDA	19.0	50

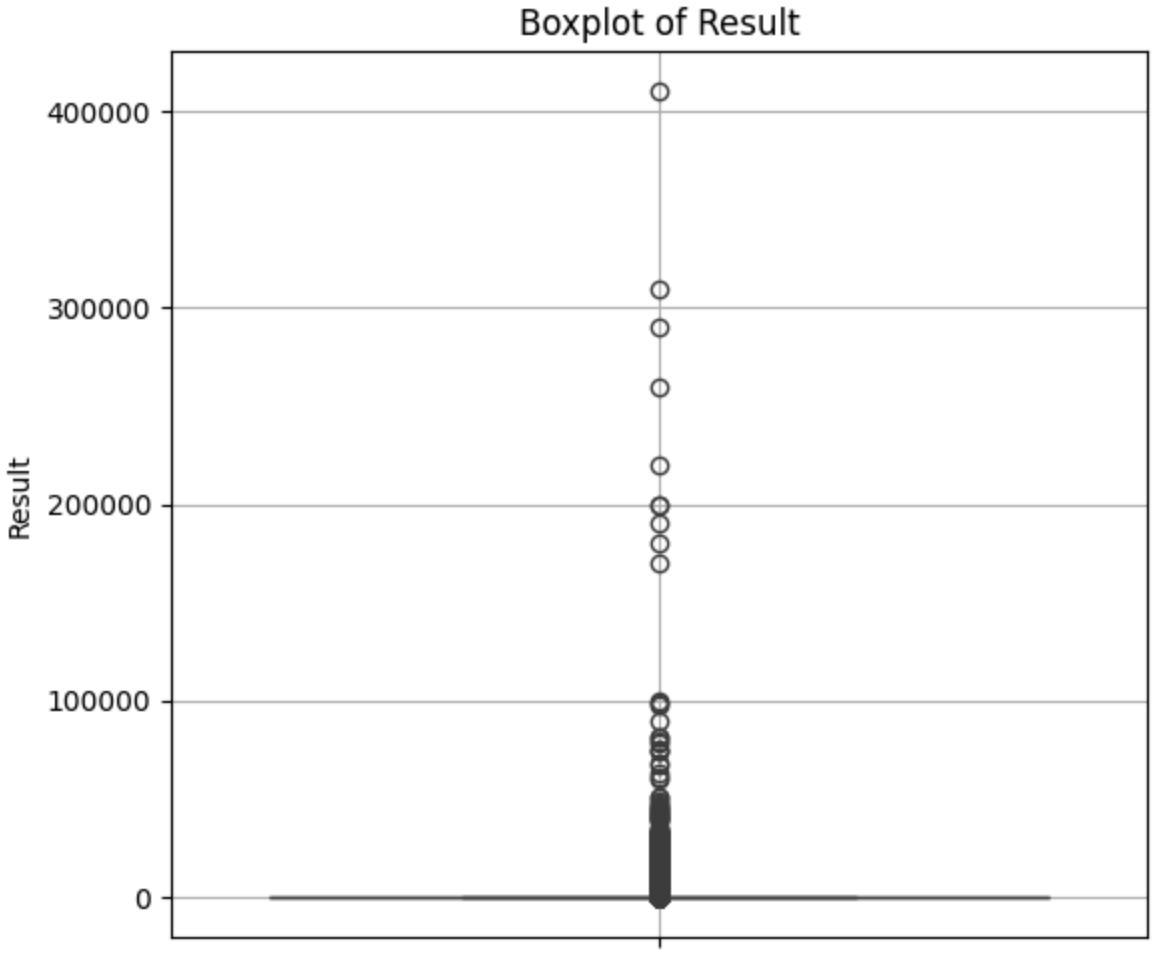
```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Analyte Name', y='Result', data=df)
plt.title("Boxplot of Result by Analyte")
```

```
plt.xlabel("Analyte")
plt.ylabel("Result")
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```

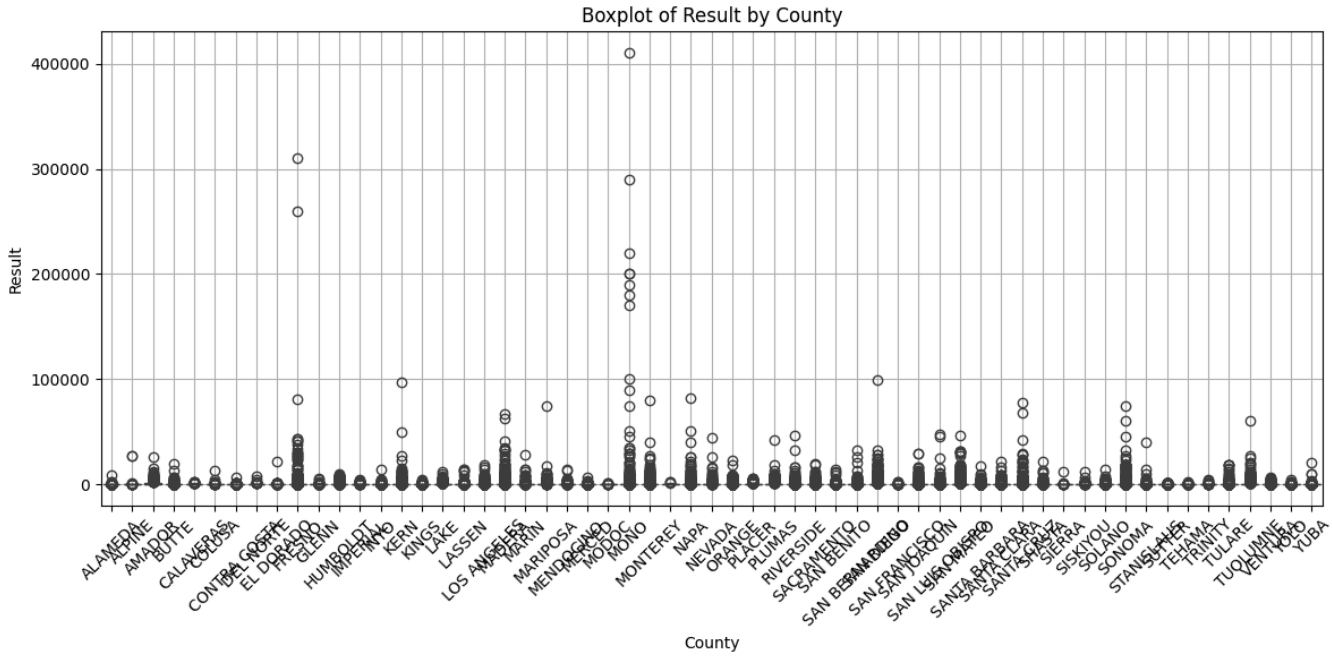



```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 5))
sns.boxplot(y='Result', data=df)
plt.title("Boxplot of Result")
```

```
plt.ylabel("Result")
plt.grid(True)
plt.tight_layout()
plt.show()
```




```
county_col = 'County' if 'County' in df.columns else 'Location'

plt.figure(figsize=(12, 6))
sns.boxplot(x=county_col, y='Result', data=df)
plt.title("Boxplot of Result by County")
plt.xlabel("County")
plt.ylabel("Result")
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```


Boxplot of Result by County
Result
County
400000
300000
200000
100000
0

```
df.describe()
```

	Row #	Service Connections	Population TINWSYS	Population R	Population NT	Population
count	638446.000000	630575.000000	6.384460e+05	5.405230e+05	92203.000000	53328.0000
mean	319223.500000	28072.563145	1.215643e+05	1.431370e+05	874.221446	3054.1322
std	184303.629319	85278.018755	4.371998e+05	4.719257e+05	3298.473769	10776.3036
min	1.000000	0.000000	0.000000e+00	1.000000e+00	1.000000	1.0000
25%	159612.250000	68.000000	2.940000e+02	1.000000e+03	40.000000	50.0000
50%	319223.500000	2683.000000	8.234000e+03	2.050000e+04	105.000000	150.0000
75%	478834.750000	25465.000000	9.462600e+04	1.236790e+05	348.000000	717.0000
max	638446.000000	709623.000000	3.856043e+06	3.856043e+06	41686.000000	141700.0000

```
df.info()
```

```
0   Row #                 638446 non-null  int64
1   Regulating Agency     638446 non-null  object
2   Water System #        638446 non-null  object
3   System                638446 non-null  object
4   Status                638446 non-null  object
5   County                638446 non-null  object
6   Service Connections   630575 non-null  float64
7   Population TINWSYS    638446 non-null  int64
8   Population R          540523 non-null  float64
9   Population NT         92203 non-null   float64
10  Population T          53328 non-null   float64
11  FED Type              638446 non-null  object
12  Facility ID           638446 non-null  object
13  Facility Name         638446 non-null  object
14  Facility Type         638446 non-null  object
15  Treatment             638446 non-null  object
16  Availability          638446 non-null  object
17  Facility Status       638446 non-null  object
18  Water Type Code       638446 non-null  object
19  Filtration Status     638446 non-null  object
20  D Source Flag         638446 non-null  object
21  Latitude              594362 non-null  float64
22  Longitude             594362 non-null  float64
23  Sampling Point ID     638446 non-null  object
24  Description           638446 non-null  object
25  Type                  638446 non-null  object
26  SampPt Status         638446 non-null  object
27  Note 3                638446 non-null  object
28  Sample Input          638446 non-null  object
```

```
 35  Collection Date           638446 non-null  object
 36  Collection Time           633190 non-null  object
 37  Sample Type               638446 non-null  object
 38  Lab Receipt Date          484272 non-null  object
 39  Collector Name            638395 non-null  object
 40  Sample Comments           638446 non-null  object
 41  Result Input              638446 non-null  object
 42  Analyte Name              638446 non-null  object
 43  Analyte Code              638446 non-null  int64
 44  Analysis Start Date       483533 non-null  object
 45  Analysis Start Time       482022 non-null  object
 46  Analysis Complete Date    544057 non-null  object
 47  Analysis Complete Time    390493 non-null  object
 48  Analysis Method Code      485310 non-null  object
 49  Less Than Indicator       638446 non-null  object
 50  Reporting Level           387614 non-null  float64
 51  Result                    254149 non-null  float64
 52   Units                    638446 non-null  object
 53  Radiological Count Error  0 non-null       float64
 54  MCL                       638446 non-null  int64
 55  TRIGGER                   638446 non-null  int64
 56  FLAG                      103221 non-null  object
dtypes: float64(9), int64(6), object(42)
memory usage: 277.6+ MB
```

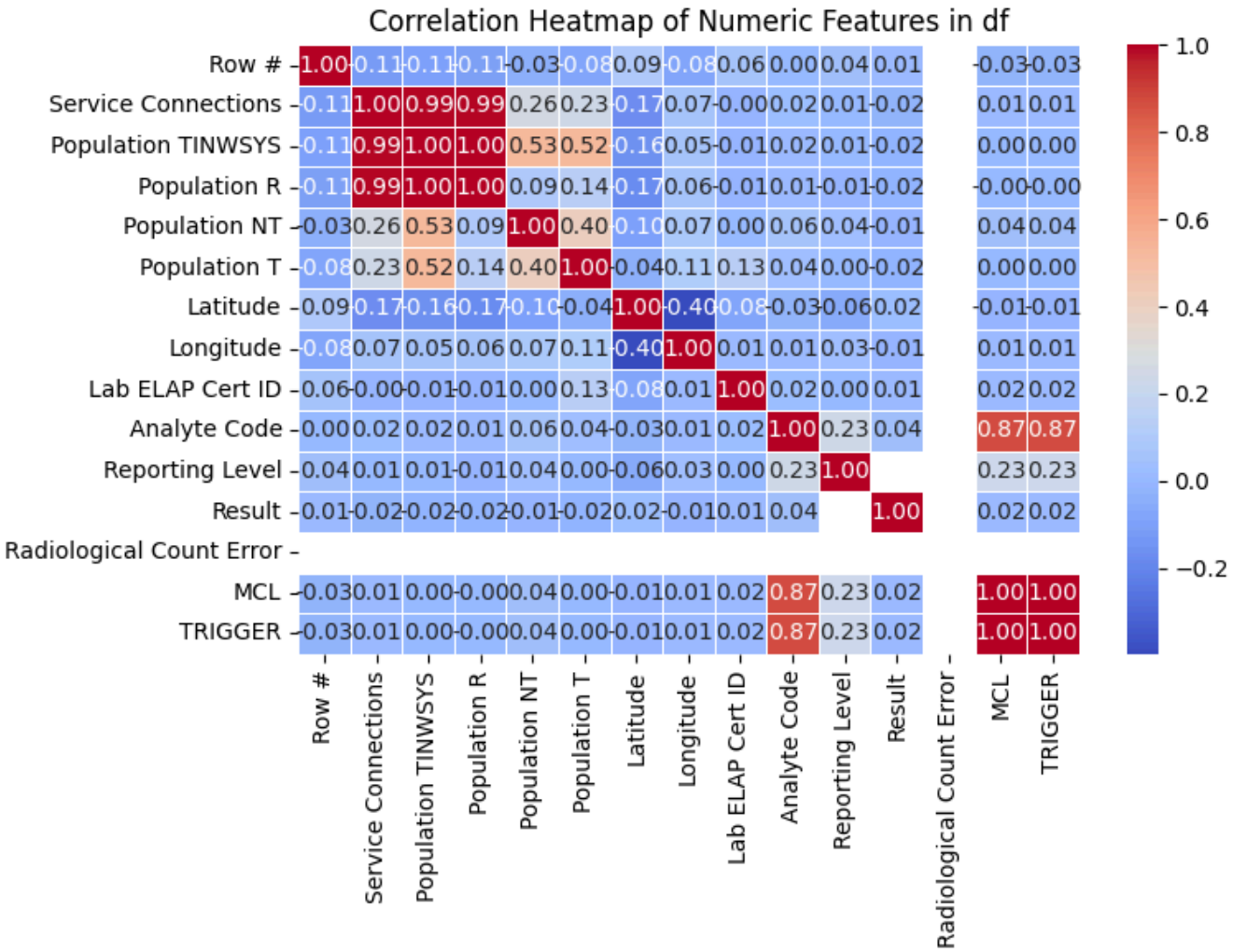
```
print("Data shape:", df.shape)
print(df.head())

print("\nColumn Data Types:")
print(df.dtypes)
```

```
Latitude                  float64
Longitude                 float64
Sampling Point ID          object
Description                object
Type                       object
SampPt Status              object
Note 3                     object
Sample Input               object
Lab ELAP Cert ID            int64
Lab Name                   object
Lab Sample ID              object
Composite YN               object
PS CODE                    object
Collection Address         object
Collection Date            object
Collection Time            object
Sample Type                object
Lab Receipt Date           object
Collector Name             object
Sample Comments            object
Result Input               object
Analyte Name               object
Analyte Code                int64
Analysis Start Date        object
Analysis Start Time        object
Analysis Complete Date     object
Analysis Complete Time     object
Analysis Method Code       object
Less Than Indicator        object
Reporting Level           float64
Result                    float64
 Units                     object
Radiological Count Error  float64
MCL                         int64
TRIGGER                     int64
```

```
print("Summary Statistics:")
print(df.describe(include='all'))

print("\nMissing values per column:")
print(df.isna().sum())

if 'Analyte' in df.columns:
    print("\nNumber of unique analytes:", df['Analyte'].nunique(
    print("Analyte categories:", df['Analyte'].unique()[:10], ".
if 'Location' in df.columns or 'Station' in df.columns:
    loc_col = 'Location' if 'Location' in df.columns else 'Stati
    print("\nNumber of unique locations:", df[loc_col].nunique()
    print("Location examples:", df[loc_col].unique()[:5], "...")
```

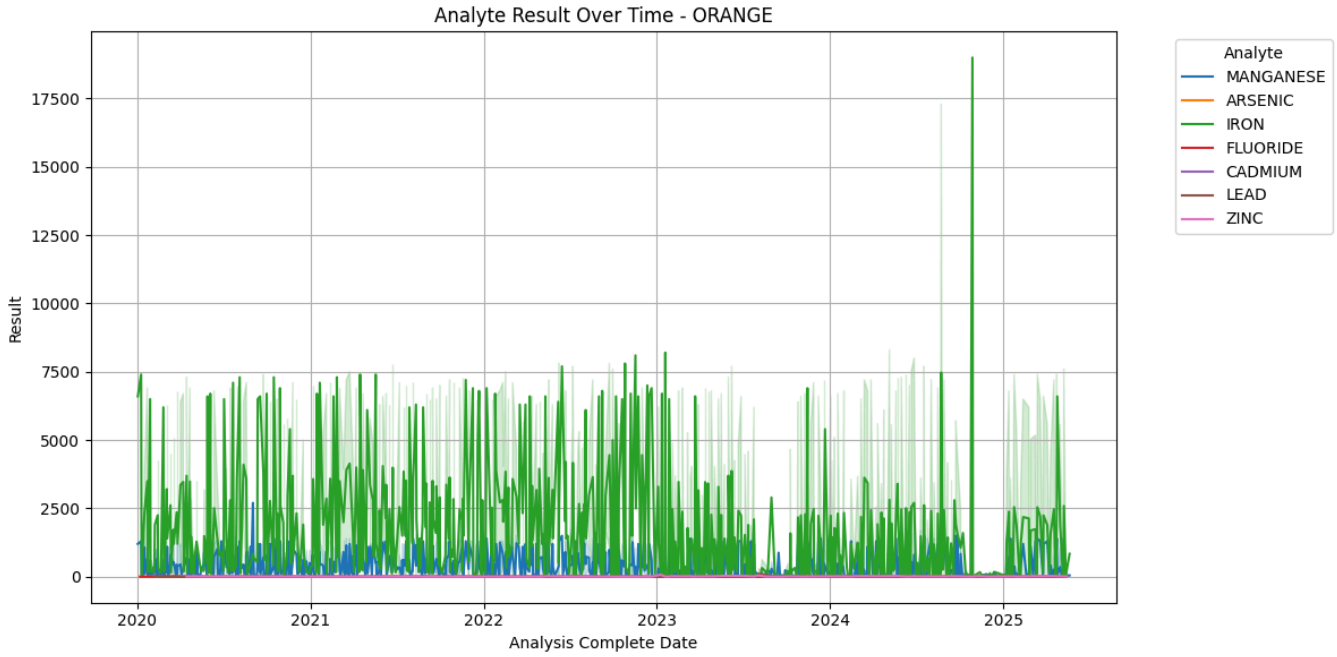
```
Population R                 97925
Population NT               546243
Population T                585118
FED Type                         0
Facility ID                      0
Facility Name                    0
Facility Type                    0
Treatment                        0
Availability                     0
Facility Status                  0
Water Type Code                  0
Filtration Status                0
D Source Flag                    0
Latitude                     44084
Longitude                    44084
Sampling Point ID                0
Description                      0
Type                             0
SampPt Status                    0
Note 3                           0
Sample Input                     0
Lab ELAP Cert ID                 0
Lab Name                         0
Lab Sample ID                    0
```

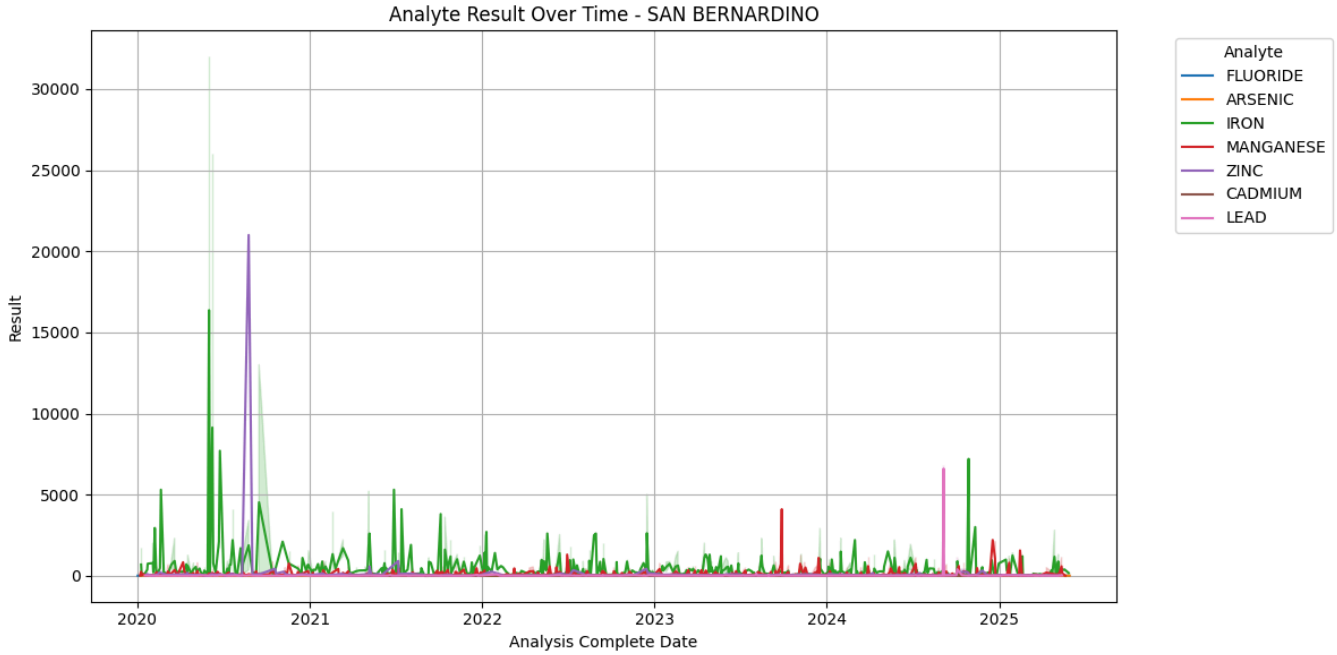
```
import seaborn as sns
import matplotlib.pyplot as plt

numeric_df = df.select_dtypes(include='number')

corr = numeric_df.corr()

plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewi
plt.title("Correlation Heatmap of Numeric Features in df")
plt.tight_layout()
plt.show()
```

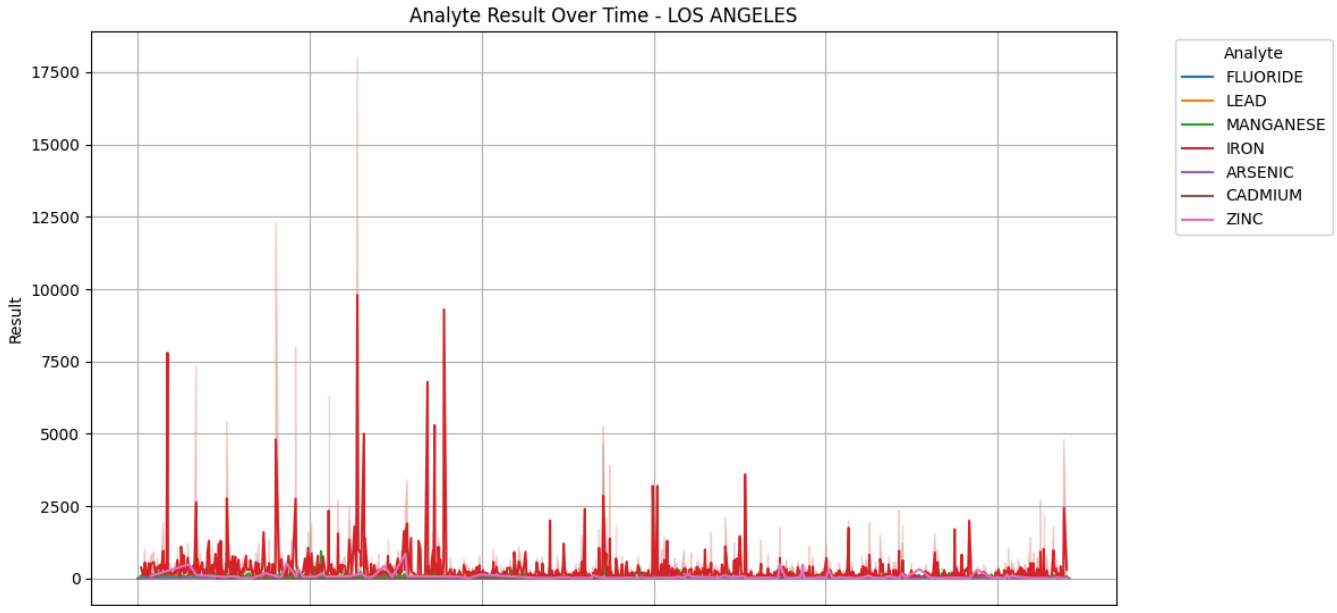



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df['Analysis Complete Date'] = pd.to_datetime(df['Analysis Compl
df = df.dropna(subset=['Analysis Complete Date'])

county_col = 'County' if 'County' in df.columns else 'Location'

df = df.sort_values('Analysis Complete Date')

counties = df[county_col].unique()

for county in counties:
```

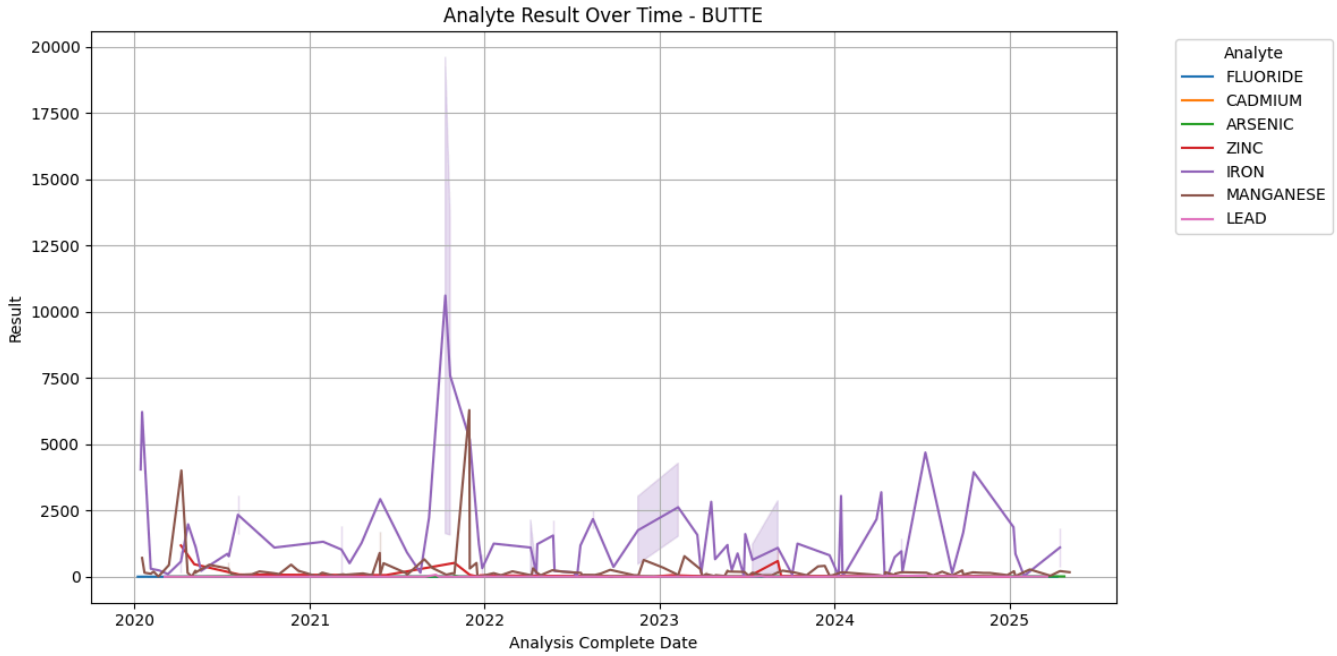
```
    county_df = df[df[county_col] == county]
    if county_df.empty:
        continue

    plt.figure(figsize=(12, 6))

    sns.lineplot(
        data=county_df,
        x='Analysis Complete Date',
        y='Result',
        hue='Analyte Name'
    )

    plt.title(f"Analyte Result Over Time - {county}")
    plt.xlabel("Analysis Complete Date")
    plt.ylabel("Result")
    plt.legend(title='Analyte', bbox_to_anchor=(1.05, 1), loc='u
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

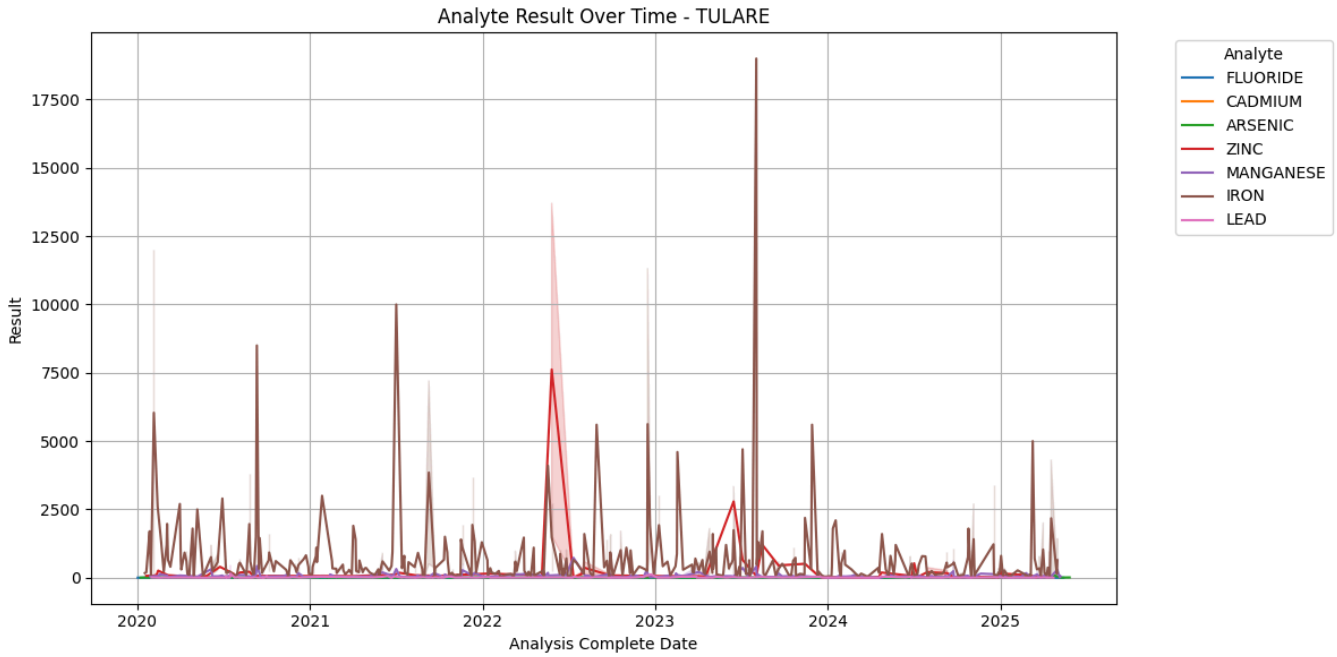

Analyte Result Over Time - ORANGE
Result
Analysis Complete Date
Analyte
MANGANESE
ARSENIC
IRON
FLUORIDE
CADMIUM
LEAD
ZINC

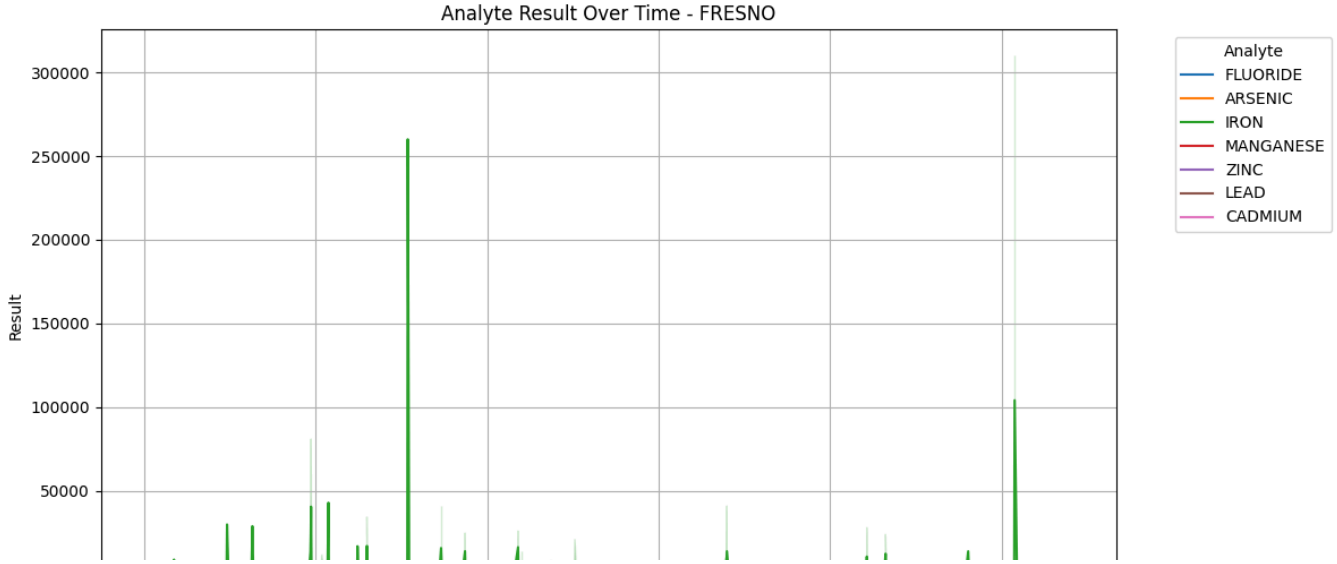

Analyte Result Over Time - SAN BERNARDINO
Result
Analysis Complete Date
Analyte
FLUORIDE
ARSENIC
IRON
MANGANESE
ZINC
CADMIUM
LEAD

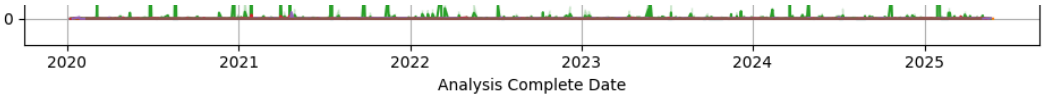

Analyte Result Over Time - LOS ANGELES
Result
Analyte
FLUORIDE
LEAD
MANGANESE
IRON
ARSENIC
CADMIUM
ZINC

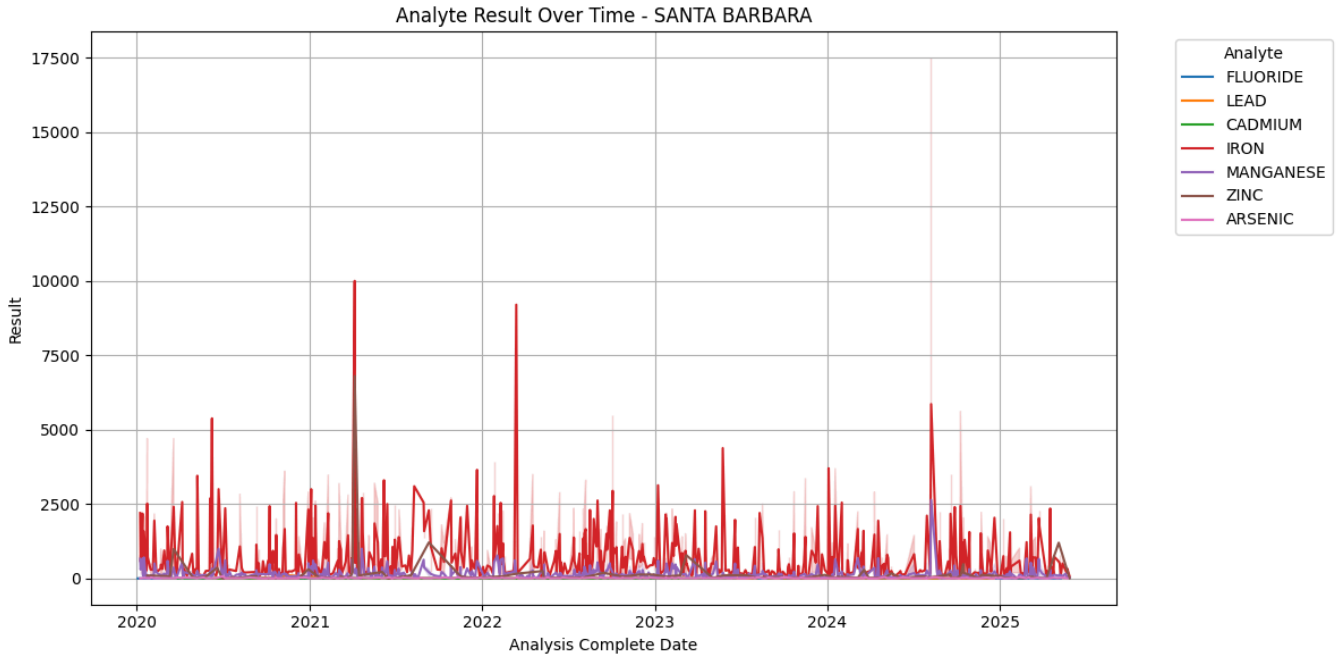

2020
2021
2022
2023
2024
2025
Analysis Complete Date

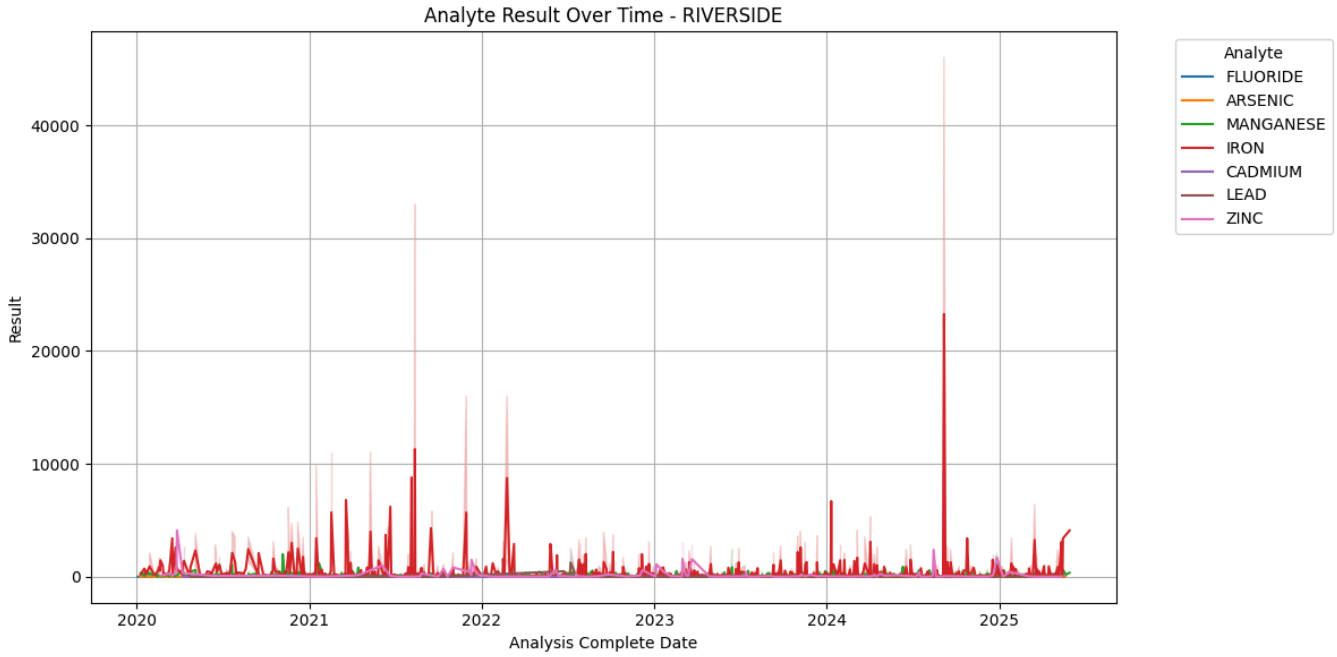

Analyte Result Over Time - BUTTE
Result
Analysis Complete Date
Analyte
FLUORIDE
CADMIUM
ARSENIC
ZINC
IRON
MANGANESE
LEAD

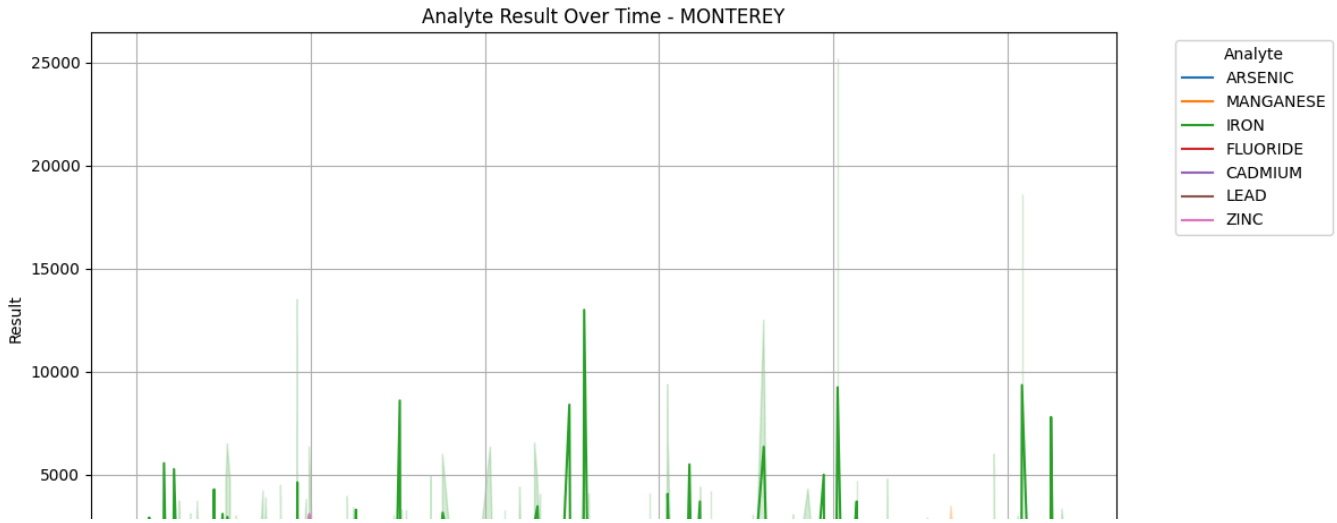

Analyte Result Over Time - TULARE
Result
Analysis Complete Date
Analyte
FLUORIDE
CADMIUM
ARSENIC
ZINC
MANGANESE
IRON
LEAD

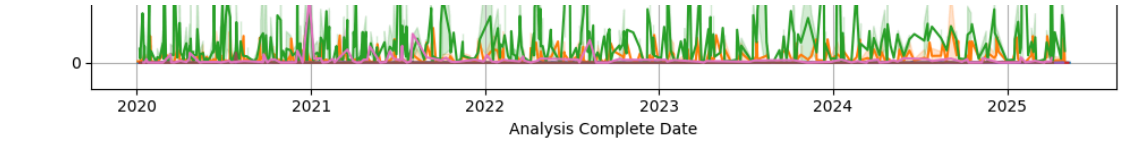

Analyte Result Over Time - FRESNO
Result
Analyte
FLUORIDE
ARSENIC
IRON
MANGANESE
ZINC
LEAD
CADMIUM

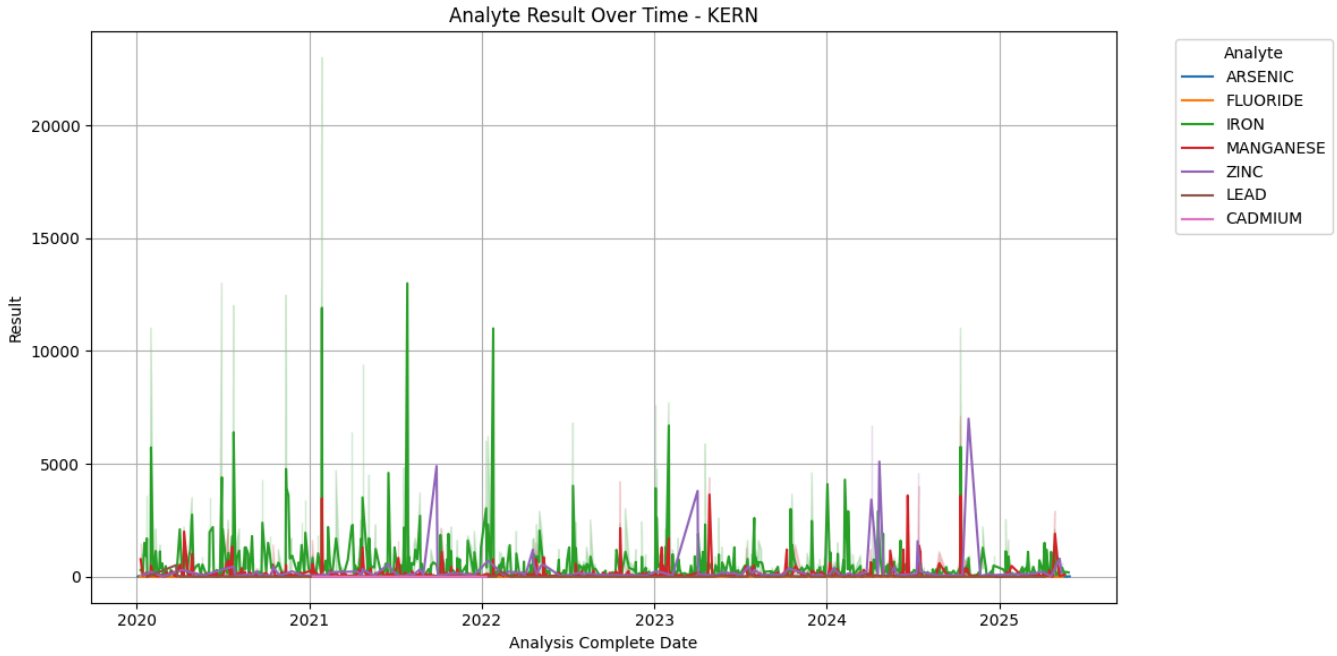

0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

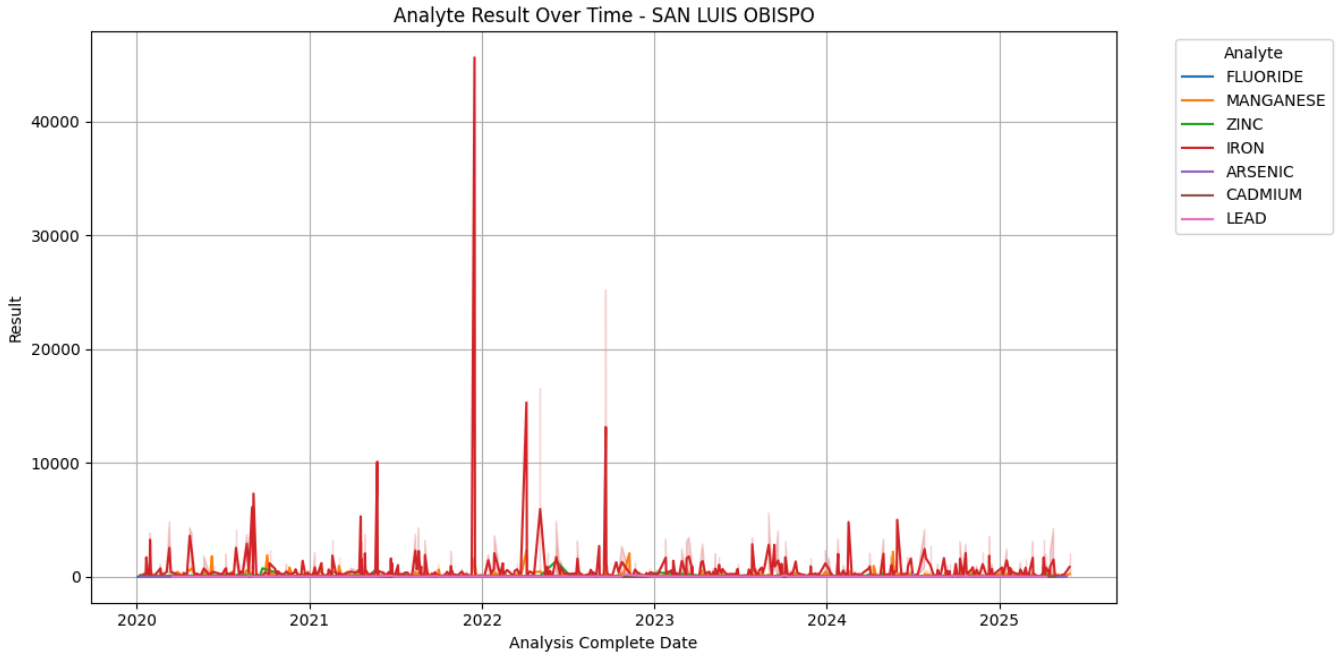

Analyte Result Over Time - SANTA BARBARA
Result
Analysis Complete Date
Analyte
FLUORIDE
LEAD
CADMIUM
IRON
MANGANESE
ZINC
ARSENIC

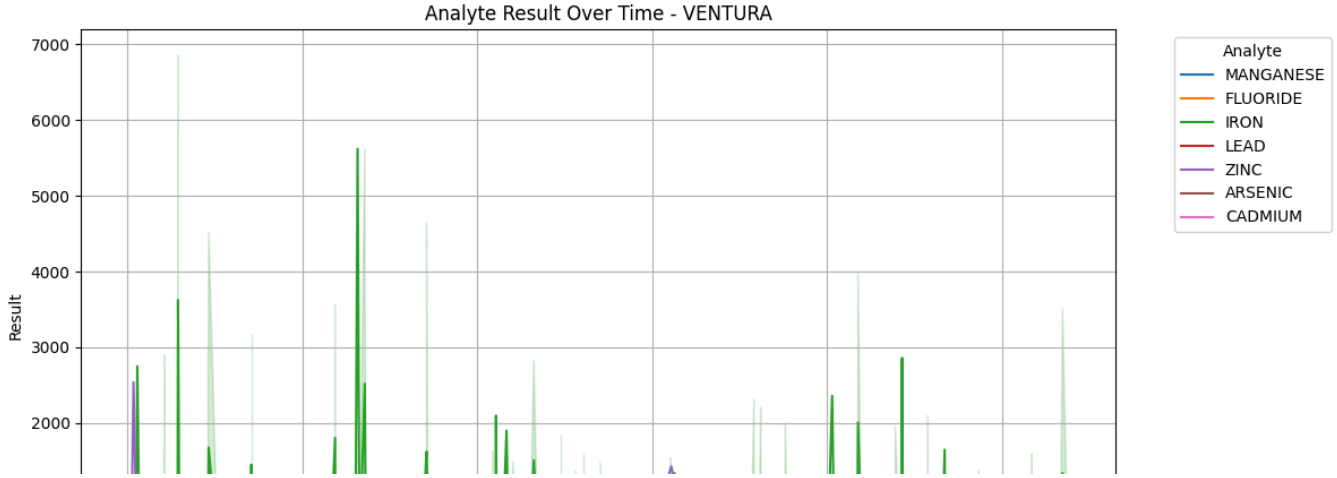

Analyte Result Over Time - RIVERSIDE
Result
Analysis Complete Date
Analyte
FLUORIDE
ARSENIC
MANGANESE
IRON
CADMIUM
LEAD
ZINC

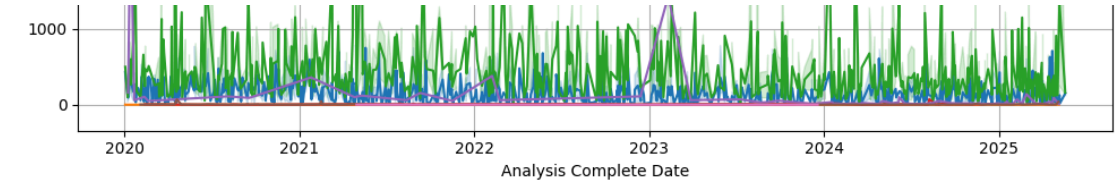

Analyte Result Over Time - MONTEREY
Result
Analysis Complete Date
Analyte
ARSENIC
MANGANESE
IRON
FLUORIDE
CADMIUM
LEAD
ZINC

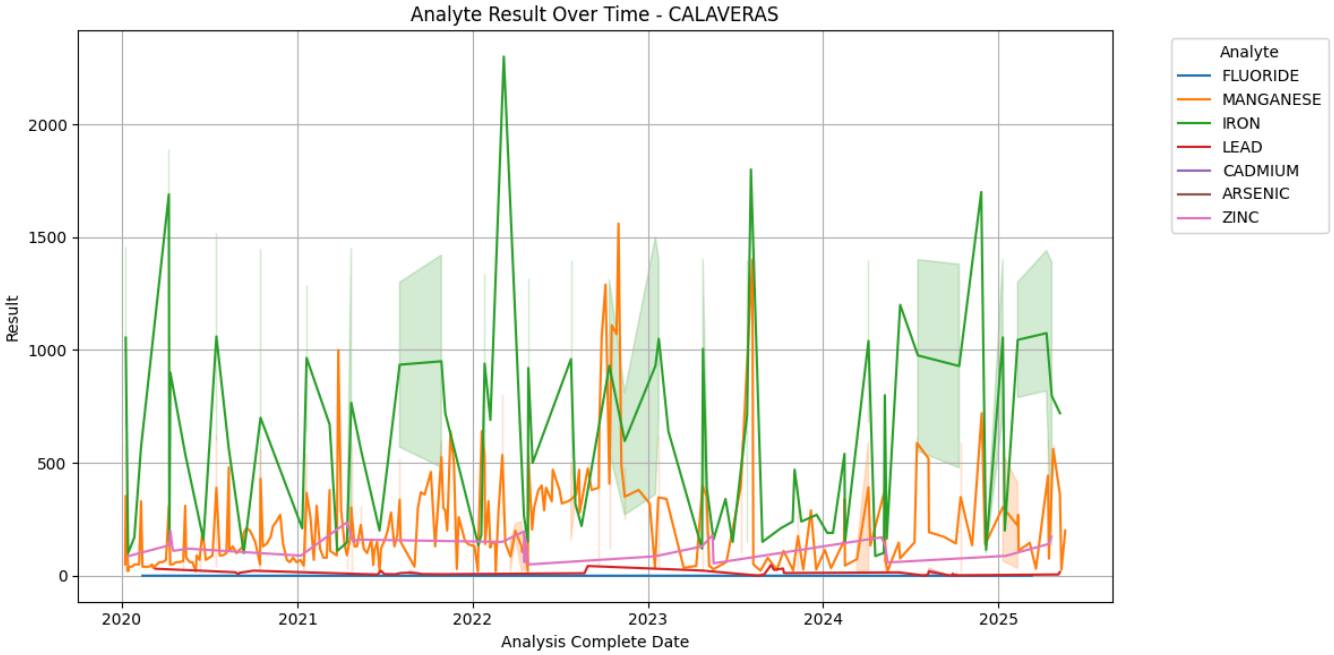

0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

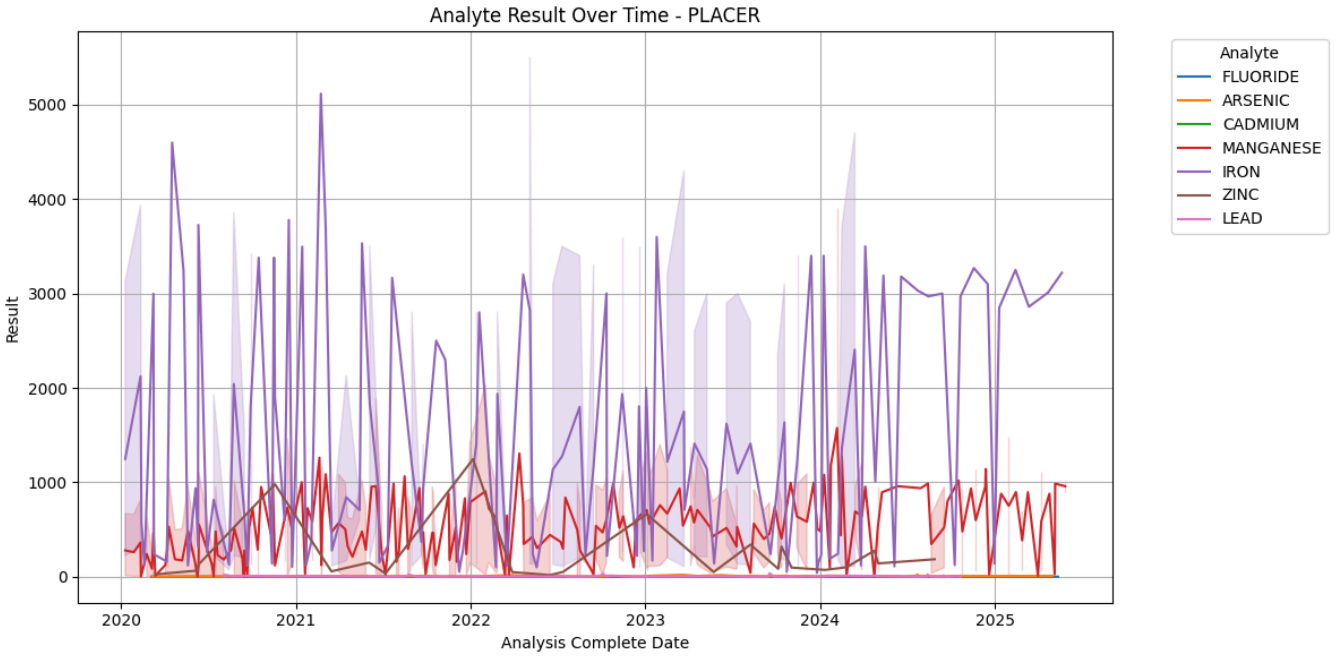

Analyte Result Over Time - KERN
Result
20000
15000
10000
5000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
ARSENIC
FLUORIDE
IRON
MANGANESE
ZINC
LEAD
CADMIUM

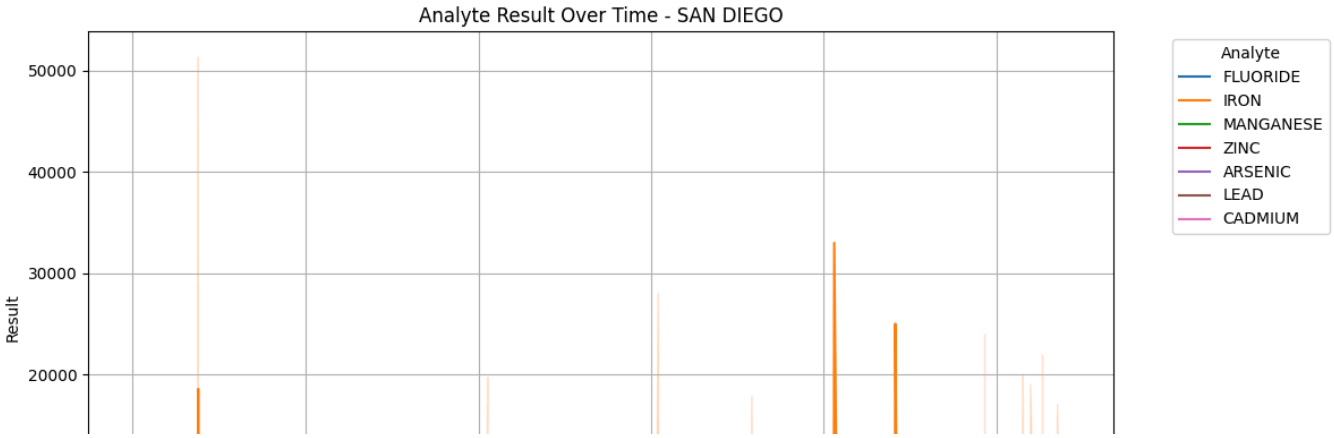

Analyte Result Over Time - SAN LUIS OBISPO
Result
40000
30000
20000
10000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
MANGANESE
ZINC
IRON
ARSENIC
CADMIUM
LEAD

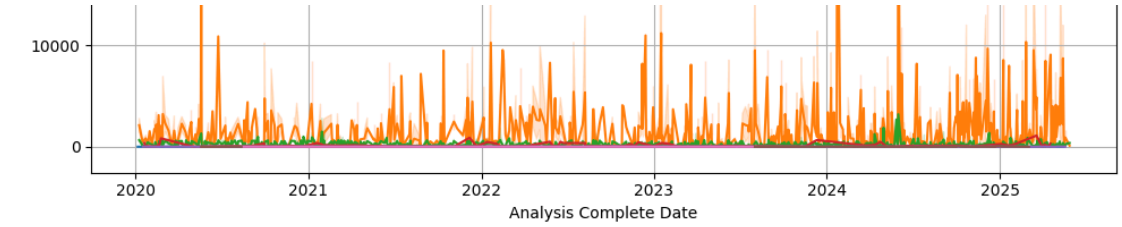

Analyte Result Over Time - VENTURA
Result
7000
6000
5000
4000
3000
2000
Analyte
MANGANESE
FLUORIDE
IRON
LEAD
ZINC
ARSENIC
CADMIUM

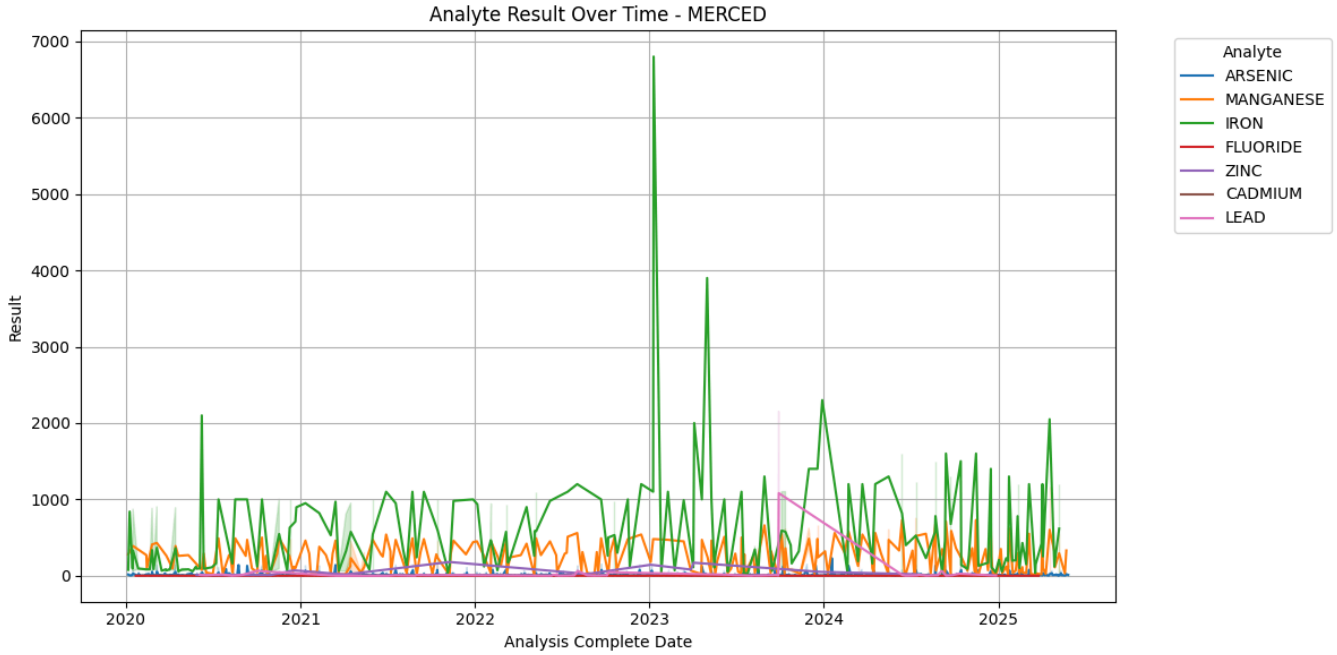

1000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

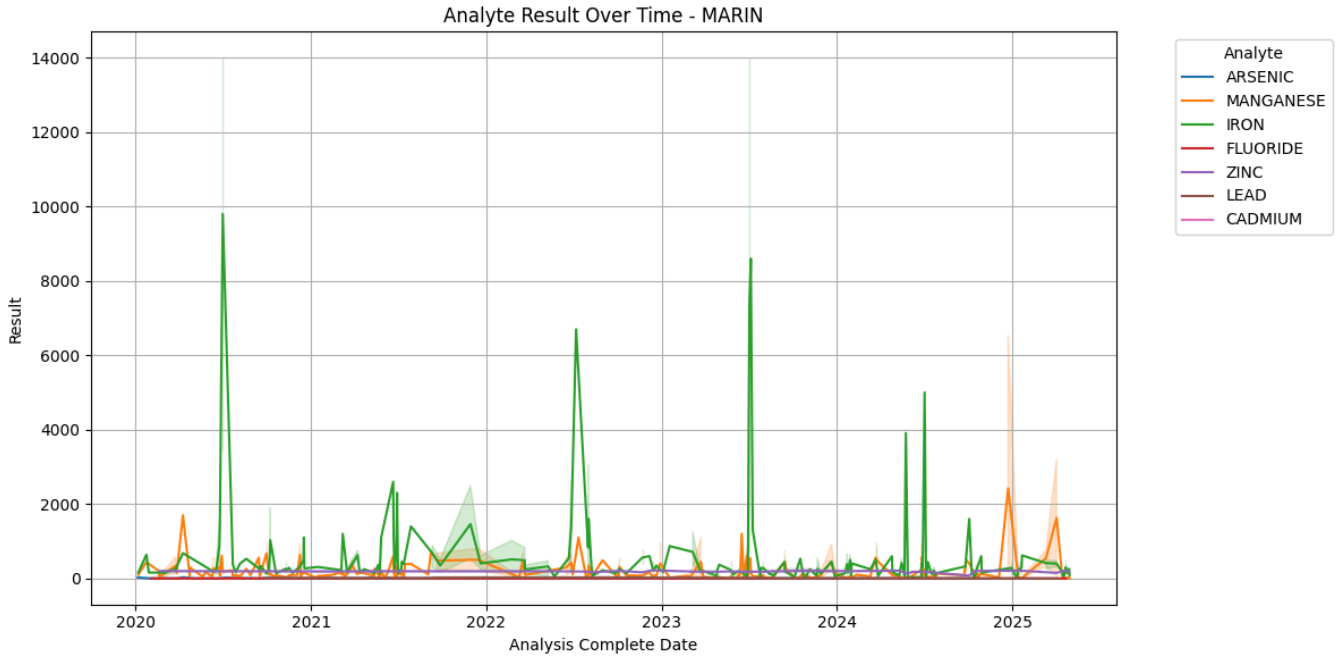

Analyte Result Over Time - CALAVERAS
Result
2000
1500
1000
500
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
MANGANESE
IRON
LEAD
CADMIUM
ARSENIC
ZINC

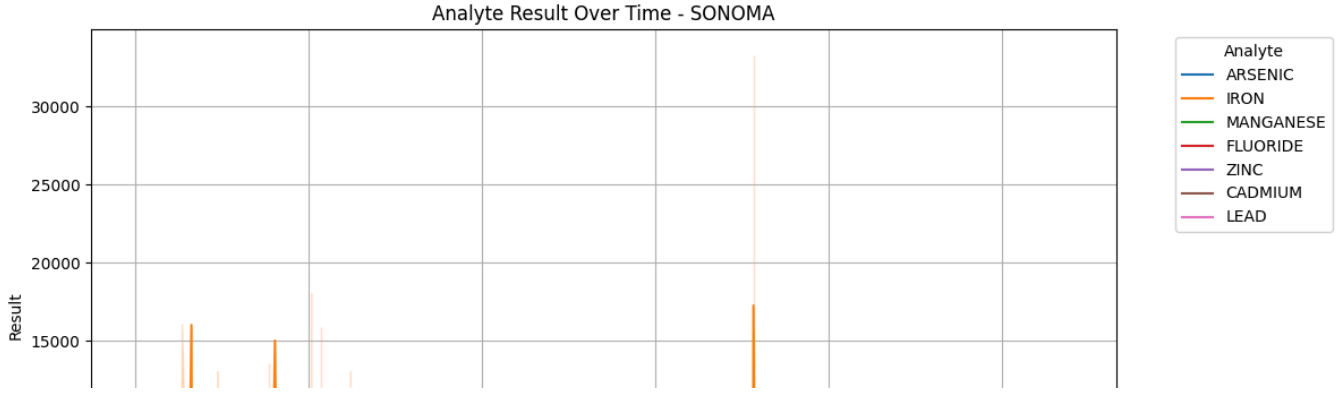

Analyte Result Over Time - PLACER
Result
5000
4000
3000
2000
1000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
ARSENIC
CADMIUM
MANGANESE
IRON
ZINC
LEAD

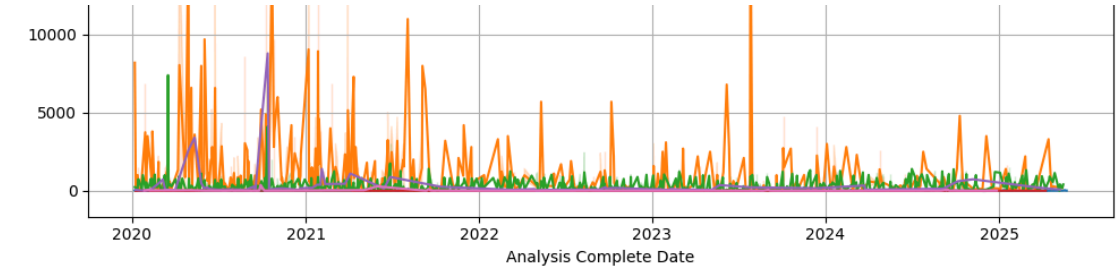

Analyte Result Over Time - SAN DIEGO
Result
50000
40000
30000
20000
Analyte
FLUORIDE
IRON
MANGANESE
ZINC
ARSENIC
LEAD
CADMIUM

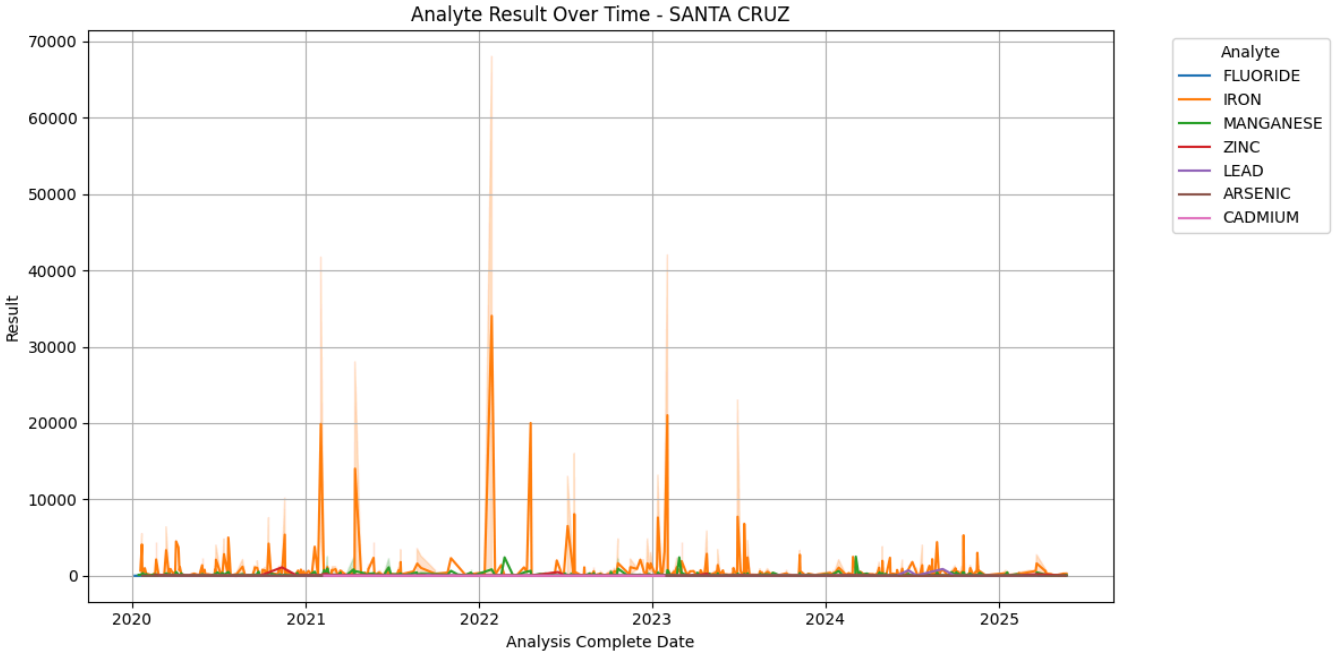

10000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

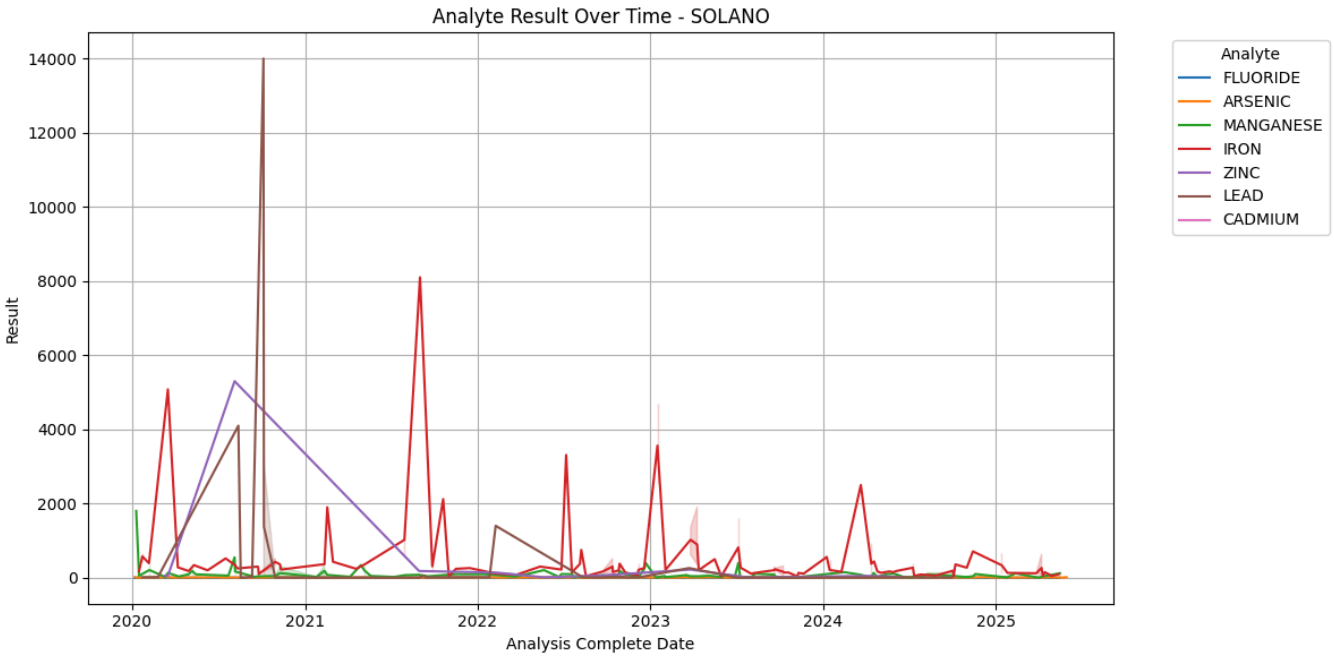

Analyte Result Over Time - MERCED
Result
Analysis Complete Date
Analyte
ARSENIC
MANGANESE
IRON
FLUORIDE
ZINC
CADMIUM
LEAD

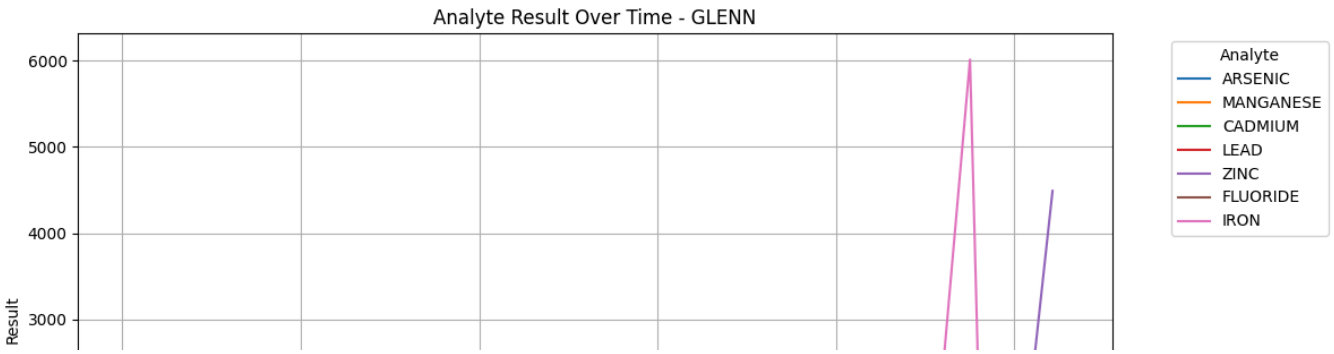

Analyte Result Over Time - MARIN
Result
Analysis Complete Date
Analyte
ARSENIC
MANGANESE
IRON
FLUORIDE
ZINC
LEAD
CADMIUM

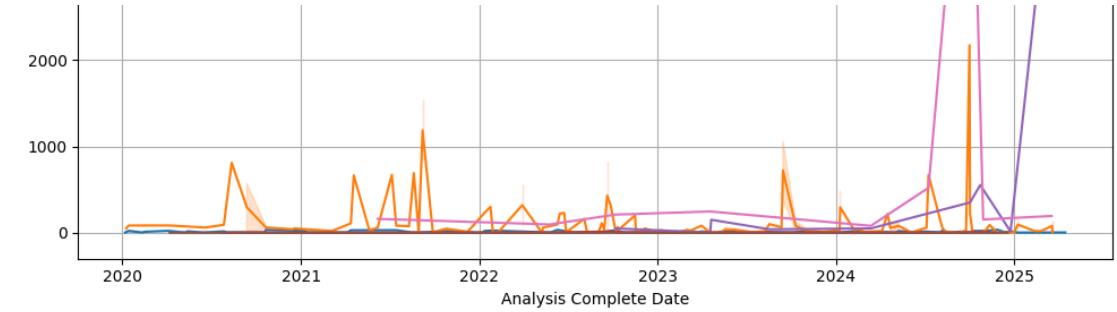

Analyte Result Over Time - SONOMA
Result
Analyte
ARSENIC
IRON
MANGANESE
FLUORIDE
ZINC
CADMIUM
LEAD

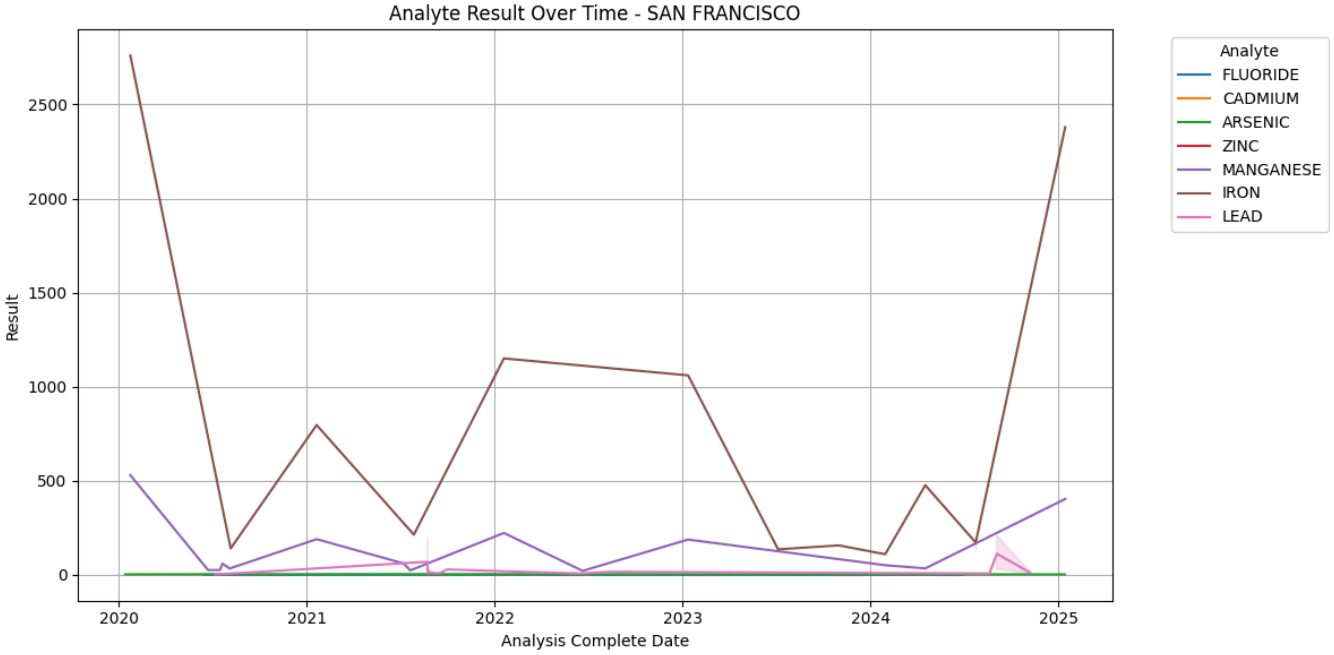

10000
5000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

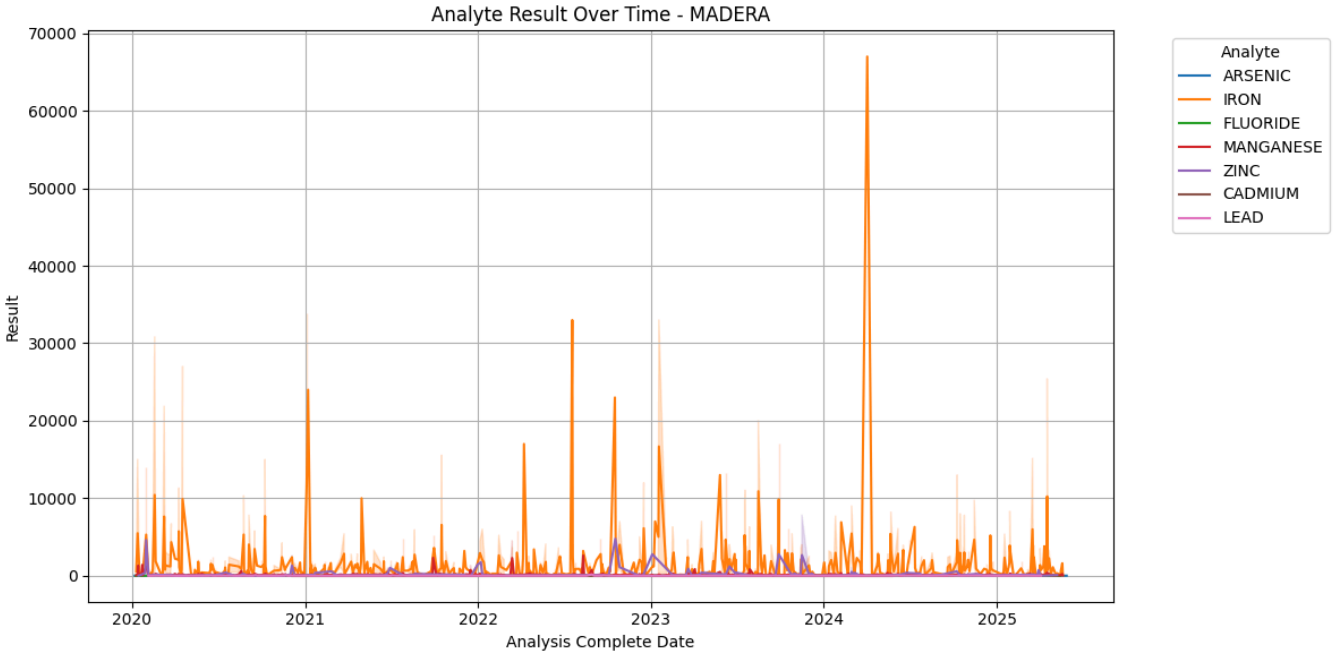

Analyte Result Over Time - SANTA CRUZ
70000
60000
50000
40000
30000
20000
10000
0
Result
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
IRON
MANGANESE
ZINC
LEAD
ARSENIC
CADMIUM

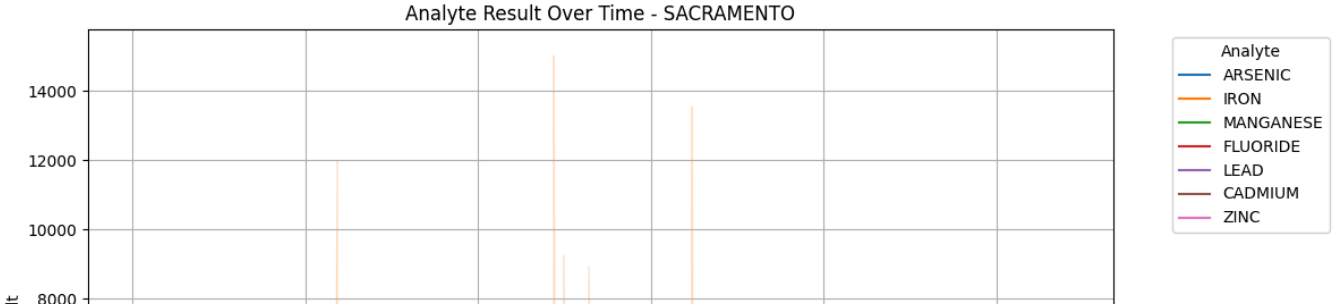

Analyte Result Over Time - SOLANO
14000
12000
10000
8000
6000
4000
2000
0
Result
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
ARSENIC
MANGANESE
IRON
ZINC
LEAD
CADMIUM

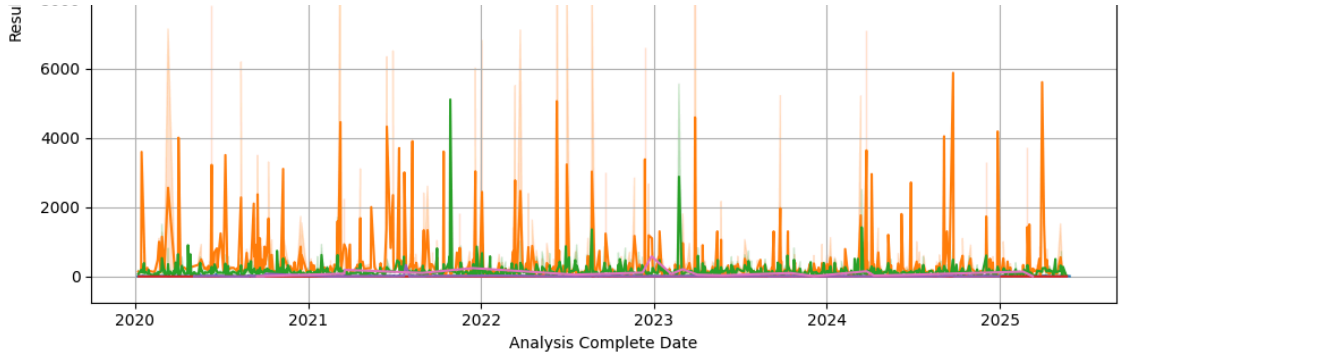

Analyte Result Over Time - GLENN
6000
5000
4000
3000
Result
Analyte
ARSENIC
MANGANESE
CADMIUM
LEAD
ZINC
FLUORIDE
IRON

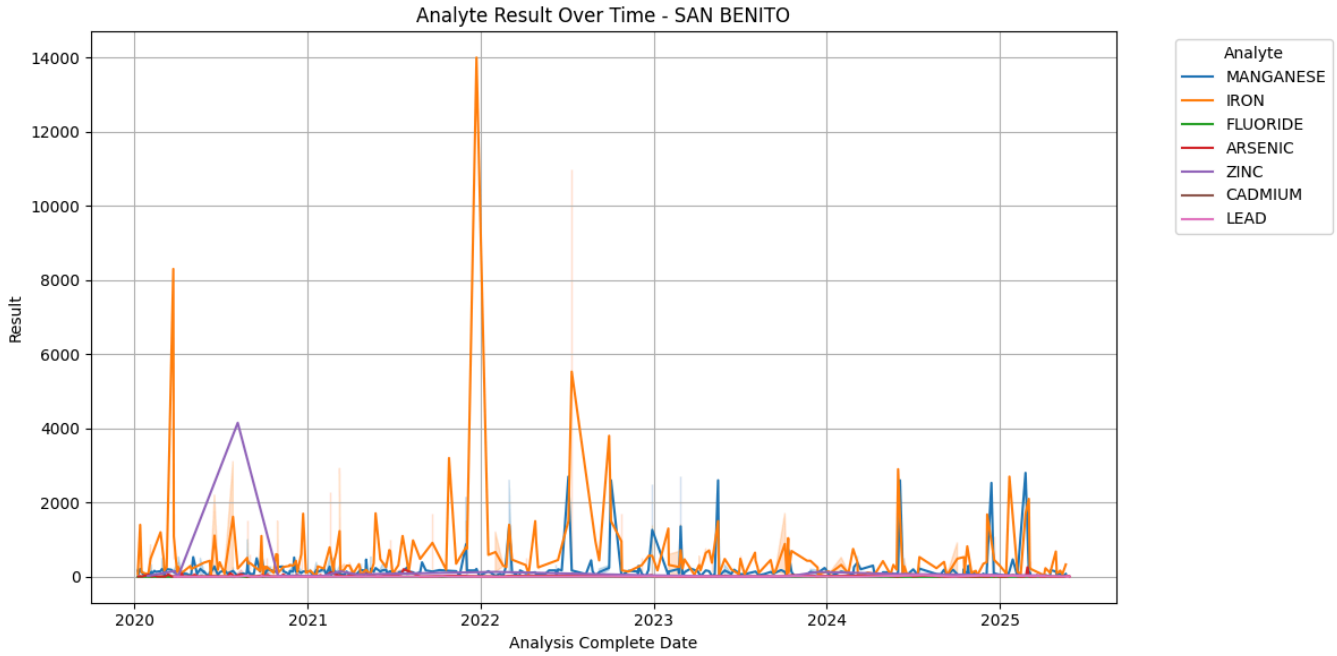

2000
1000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

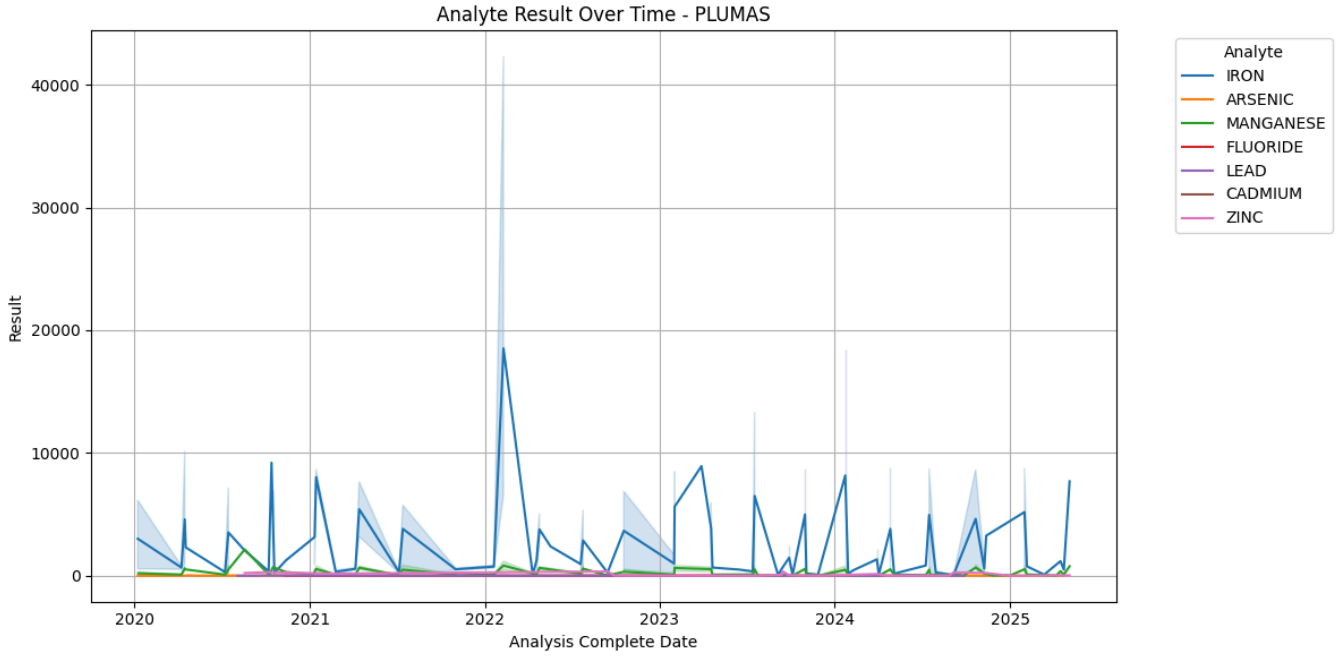

Analyte Result Over Time - SAN FRANCISCO
Result
2500
2000
1500
1000
500
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
CADMIUM
ARSENIC
ZINC
MANGANESE
IRON
LEAD

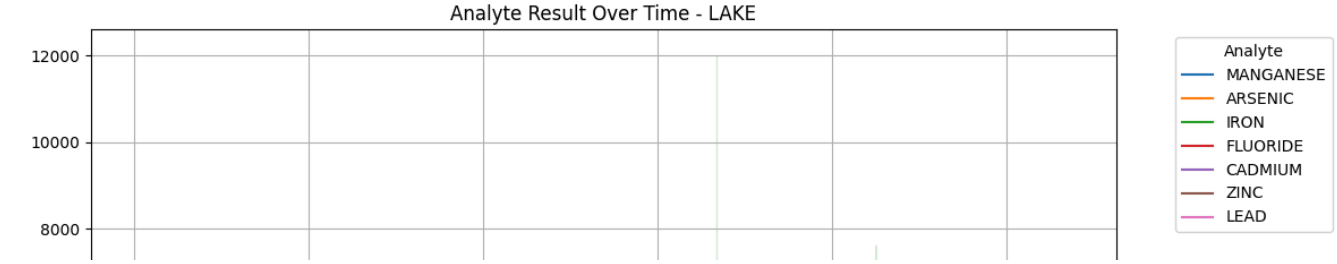

Analyte Result Over Time - MADERA
Result
70000
60000
50000
40000
30000
20000
10000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
ARSENIC
IRON
FLUORIDE
MANGANESE
ZINC
CADMIUM
LEAD

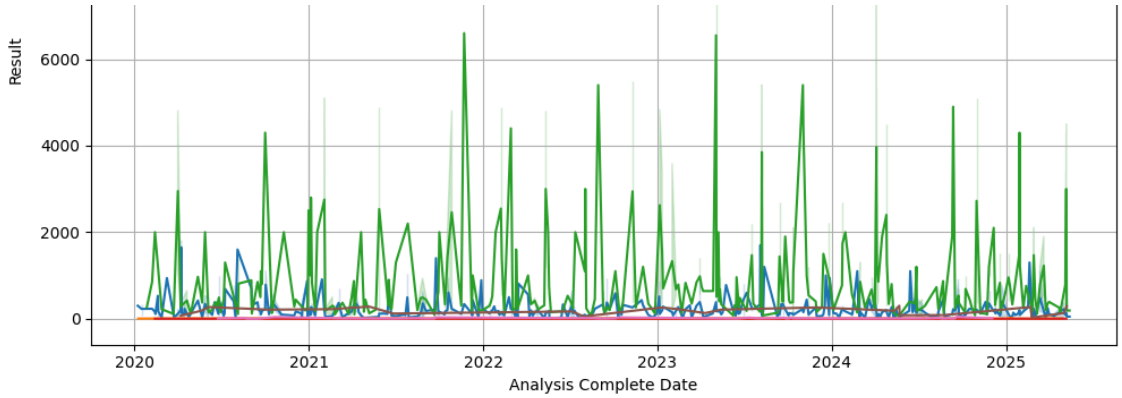

Analyte Result Over Time - SACRAMENTO
14000
12000
10000
8000
Analyte
ARSENIC
IRON
MANGANESE
FLUORIDE
LEAD
CADMIUM
ZINC

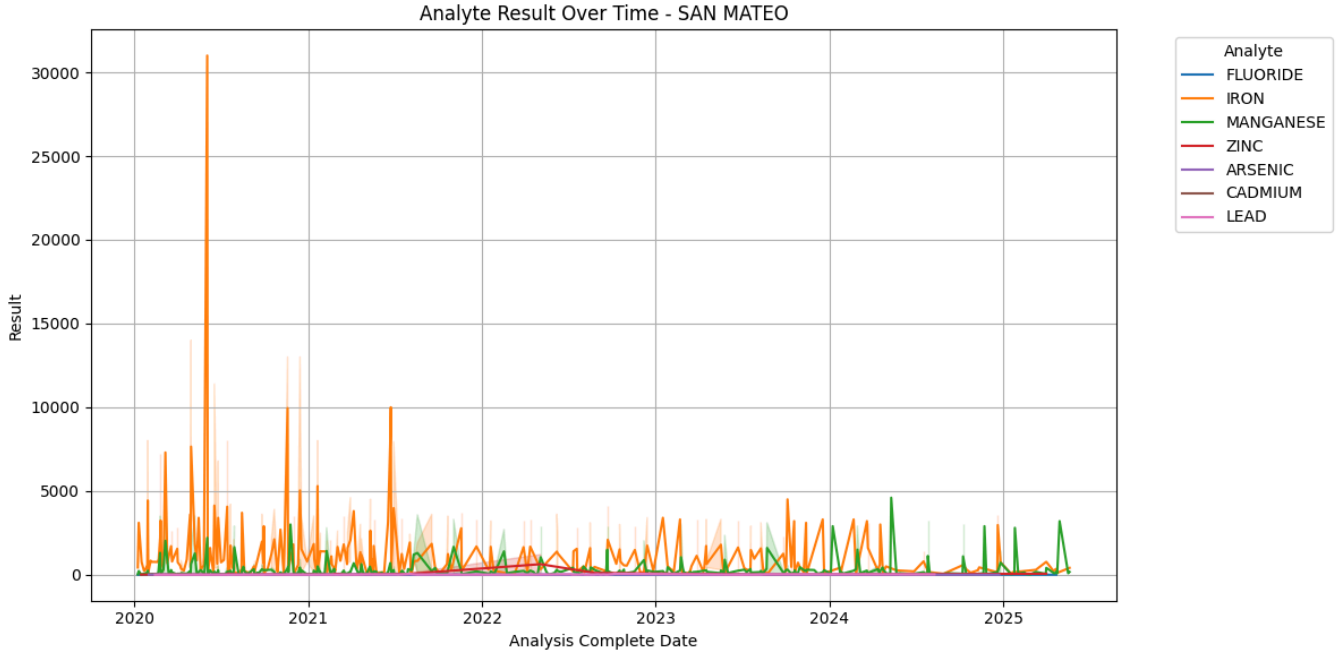

Resu
6000
4000
2000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

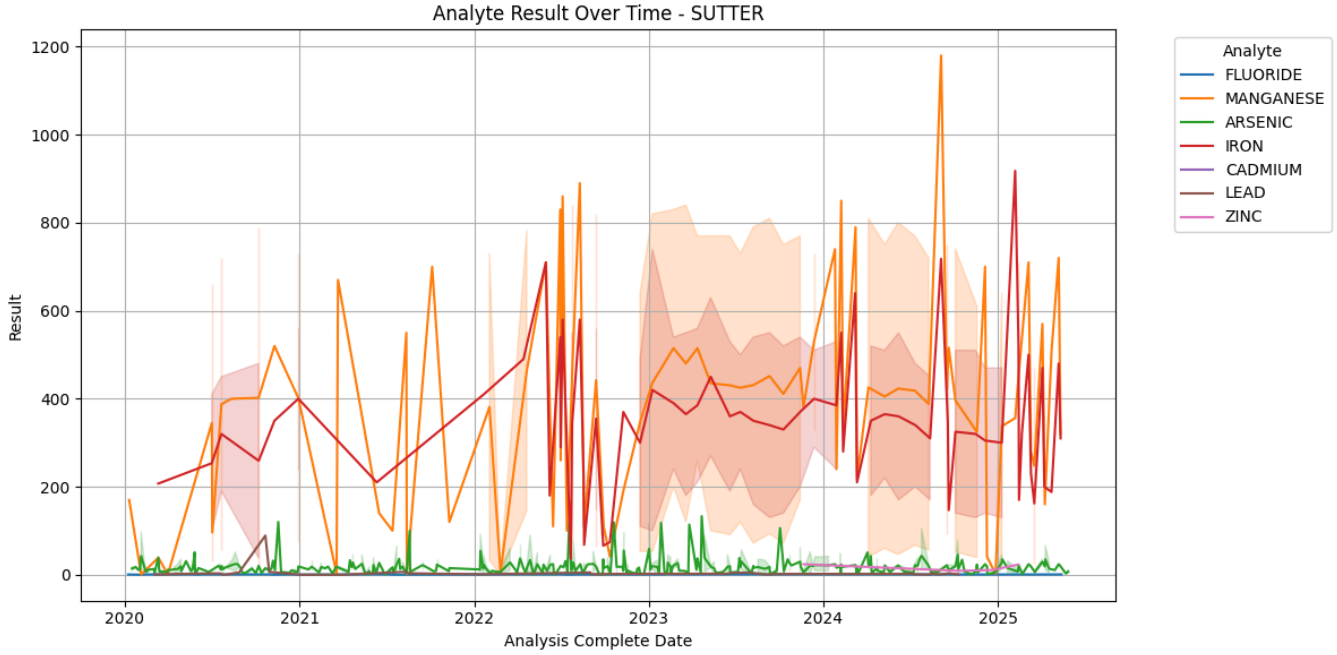

Analyte Result Over Time - SAN BENITO
Result
14000
12000
10000
8000
6000
4000
2000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
MANGANESE
IRON
FLUORIDE
ARSENIC
ZINC
CADMIUM
LEAD

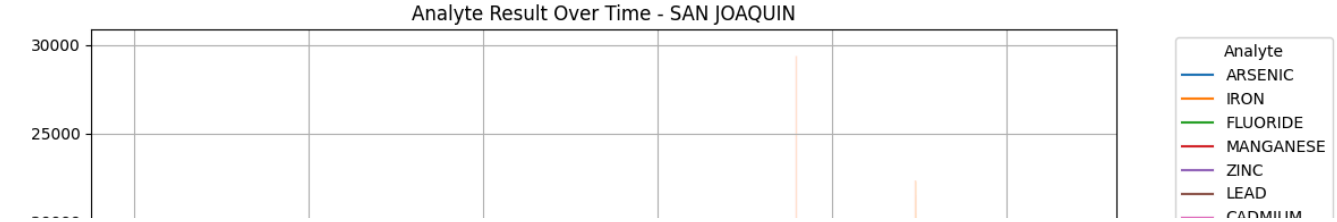

Analyte Result Over Time - PLUMAS
Result
40000
30000
20000
10000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
IRON
ARSENIC
MANGANESE
FLUORIDE
LEAD
CADMIUM
ZINC

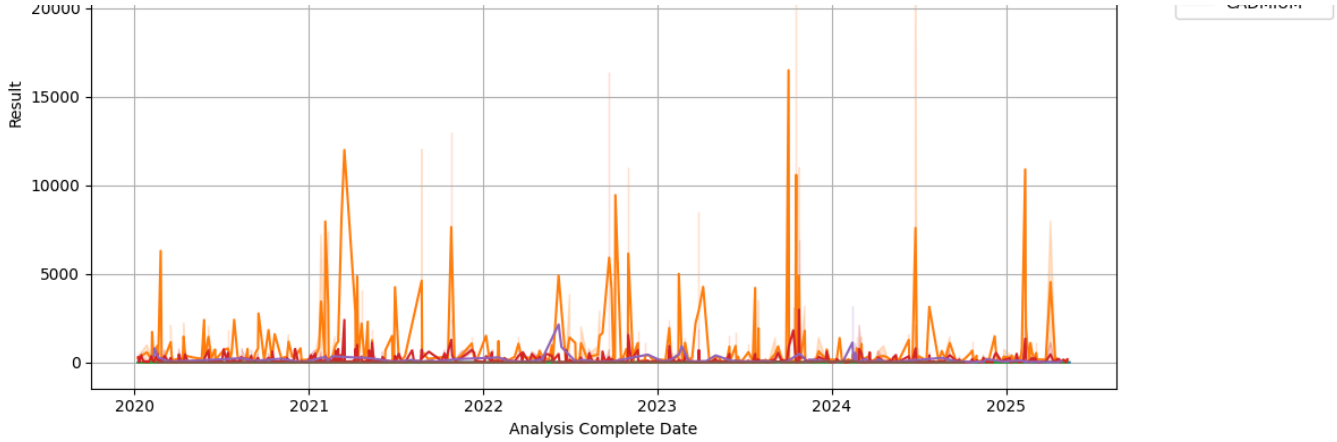

Analyte Result Over Time - LAKE
12000
10000
8000
Analyte
MANGANESE
ARSENIC
IRON
FLUORIDE
CADMIUM
ZINC
LEAD

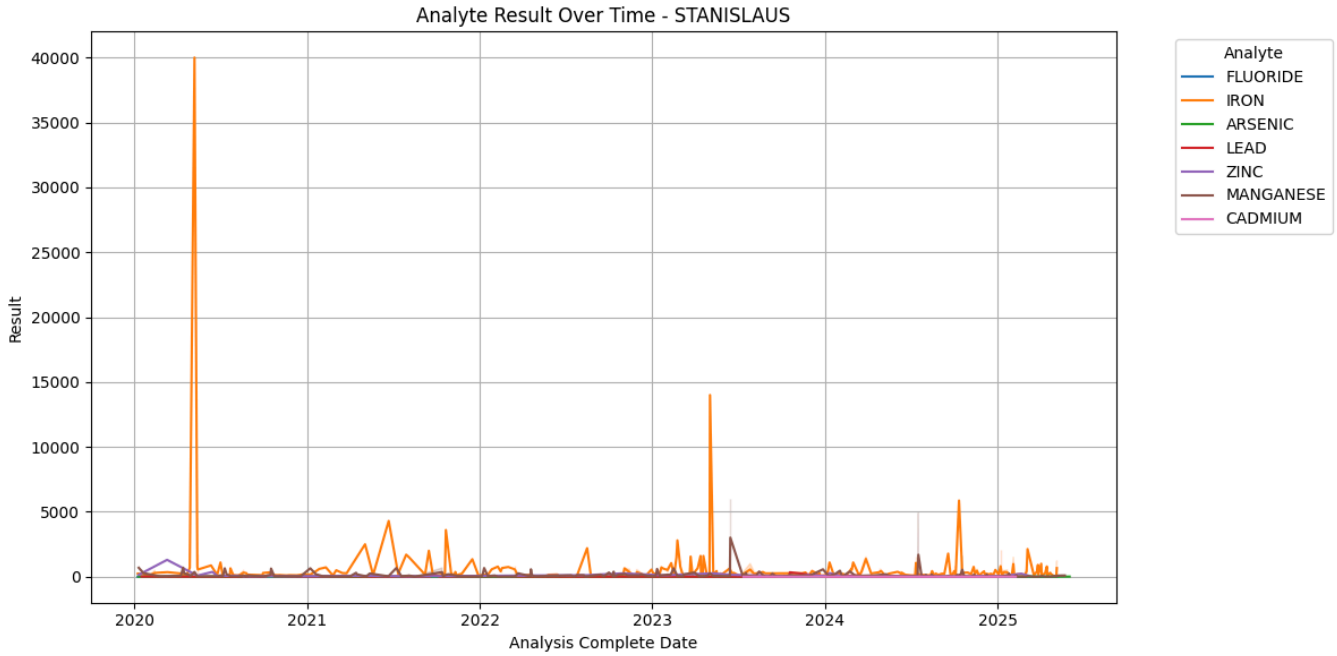

Result
6000
4000
2000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

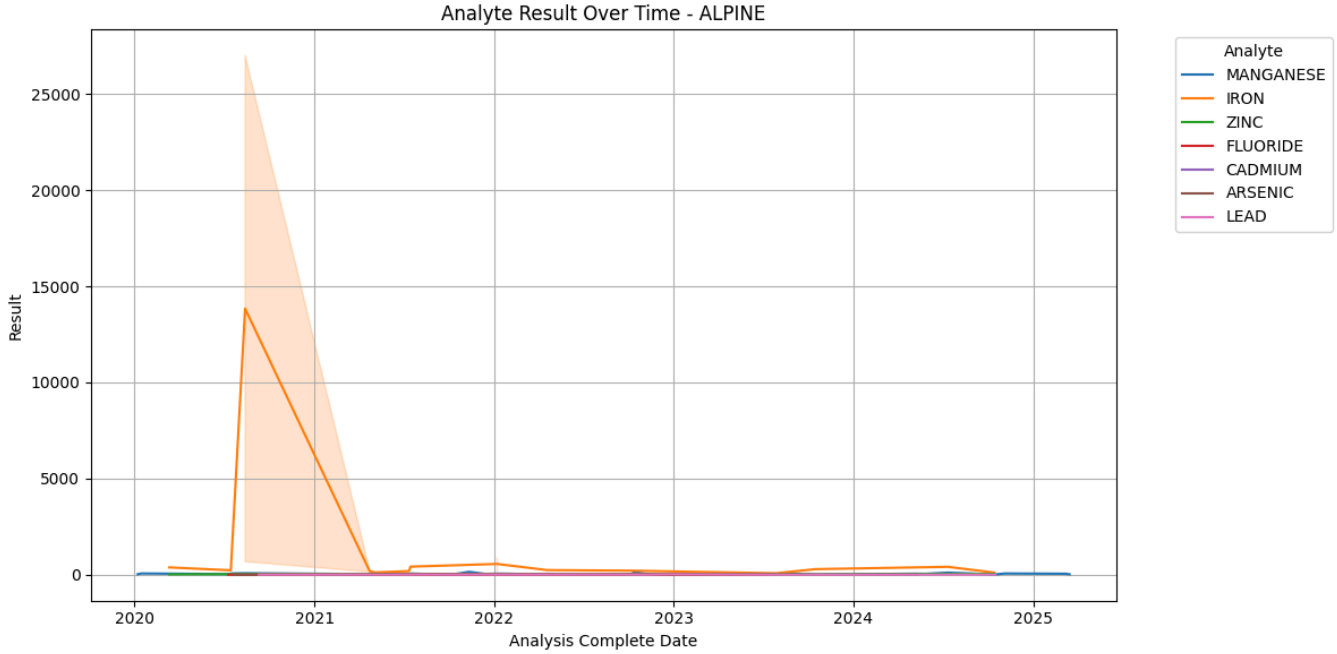

Analyte Result Over Time - SAN MATEO
Result
30000
25000
20000
15000
10000
5000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
IRON
MANGANESE
ZINC
ARSENIC
CADMIUM
LEAD

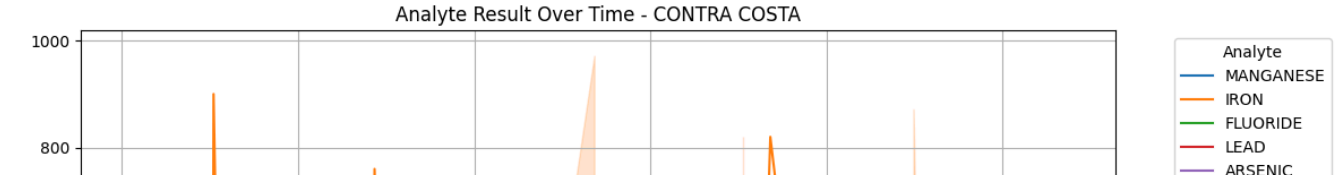

Analyte Result Over Time - SUTTER
Result
1200
1000
800
600
400
200
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
MANGANESE
ARSENIC
IRON
CADMIUM
LEAD
ZINC

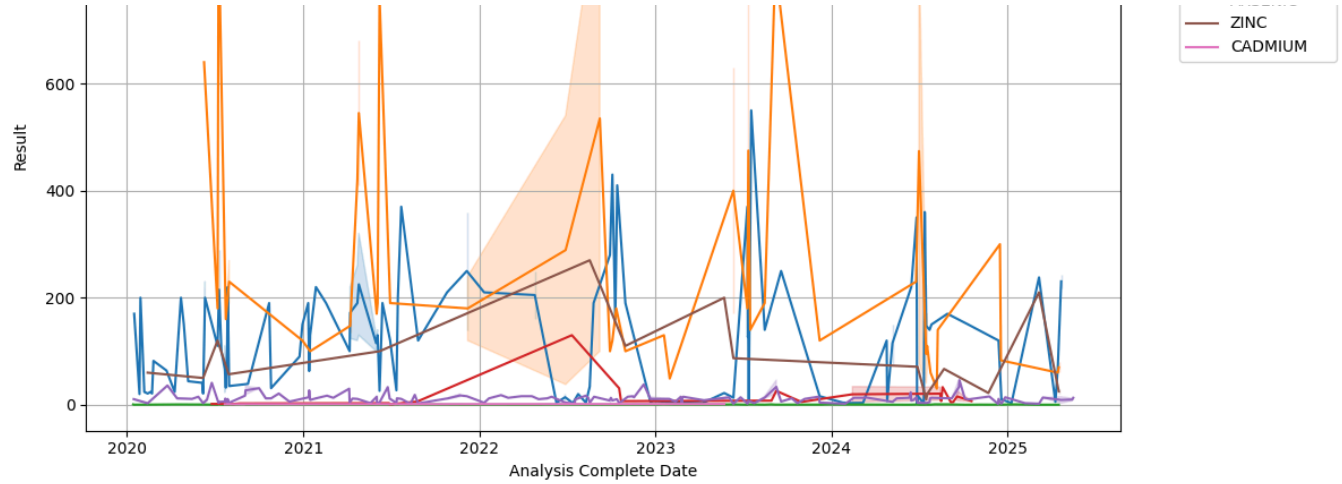

Analyte Result Over Time - SAN JOAQUIN
30000
25000
Analyte
ARSENIC
IRON
FLUORIDE
MANGANESE
ZINC
LEAD

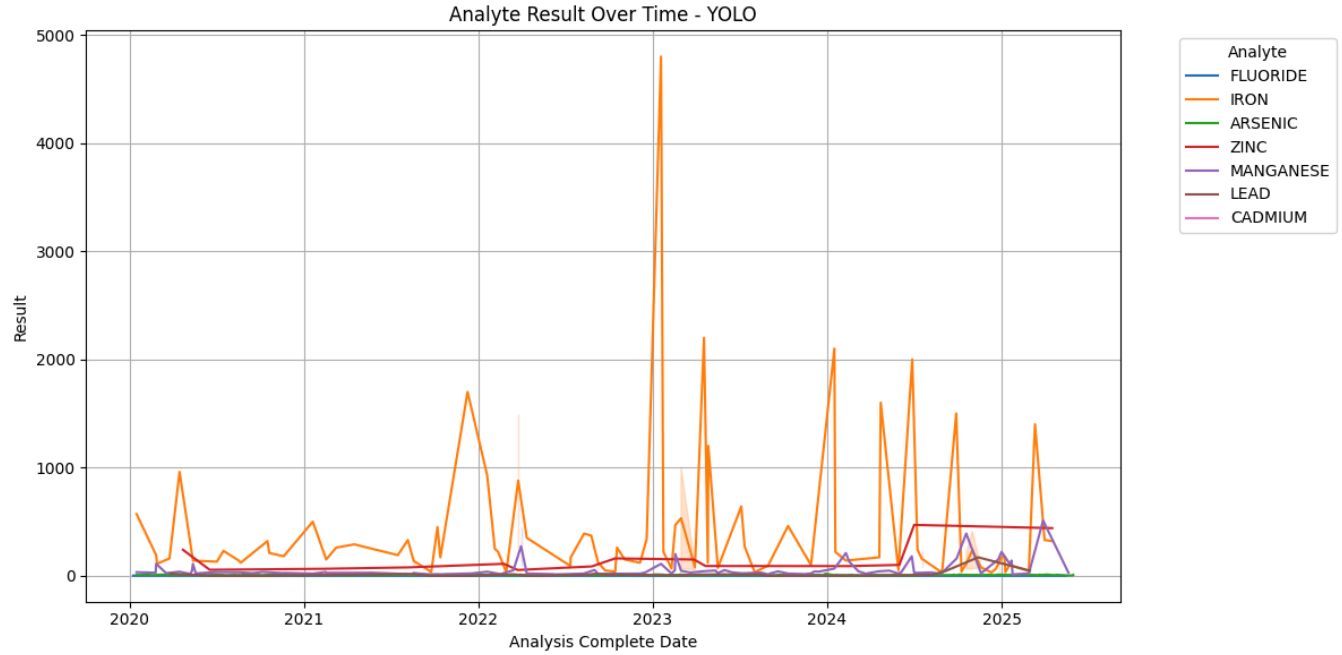

20000
15000
10000
5000
0
Result
2020
2021
2022
2023
2024
2025
Analysis Complete Date

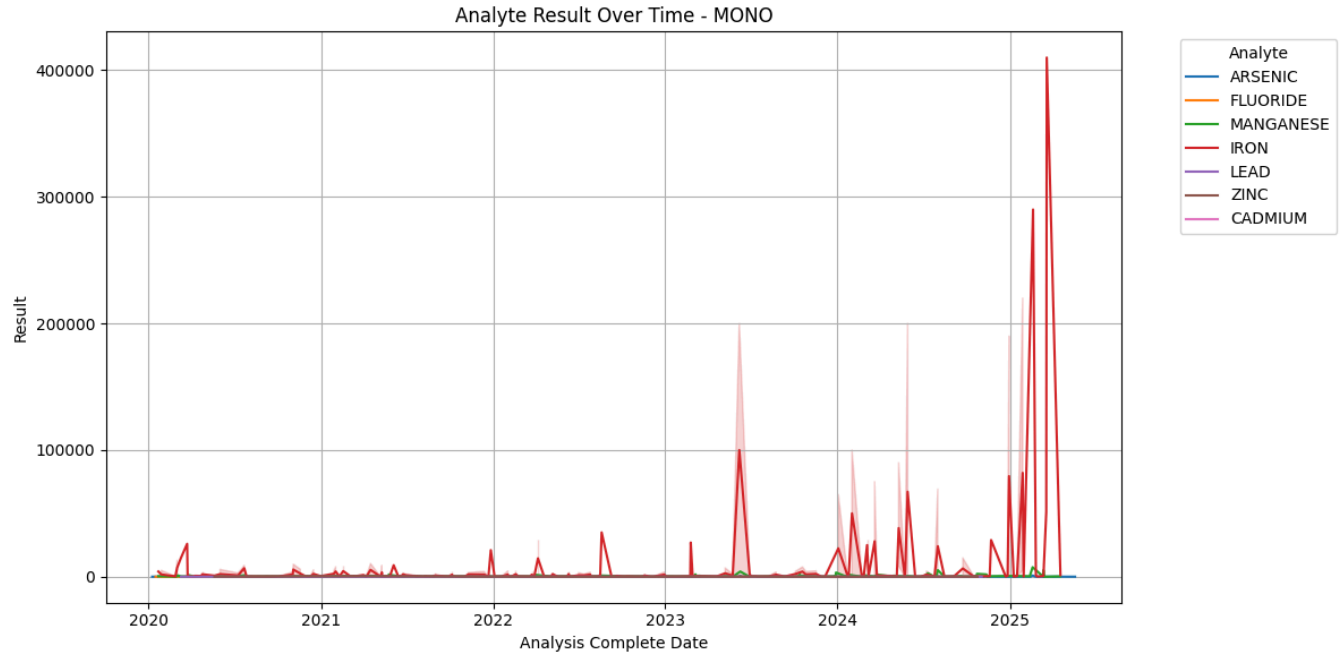

Analyte Result Over Time - STANISLAUS
40000
35000
30000
25000
20000
15000
10000
5000
0
Result
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
FLUORIDE
IRON
ARSENIC
LEAD
ZINC
MANGANESE
CADMIUM

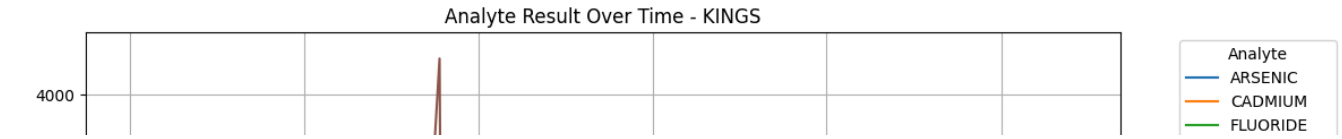

Analyte Result Over Time - ALPINE
25000
20000
15000
10000
5000
0
Result
2020
2021
2022
2023
2024
2025
Analysis Complete Date
Analyte
MANGANESE
IRON
ZINC
FLUORIDE
CADMIUM
ARSENIC
LEAD

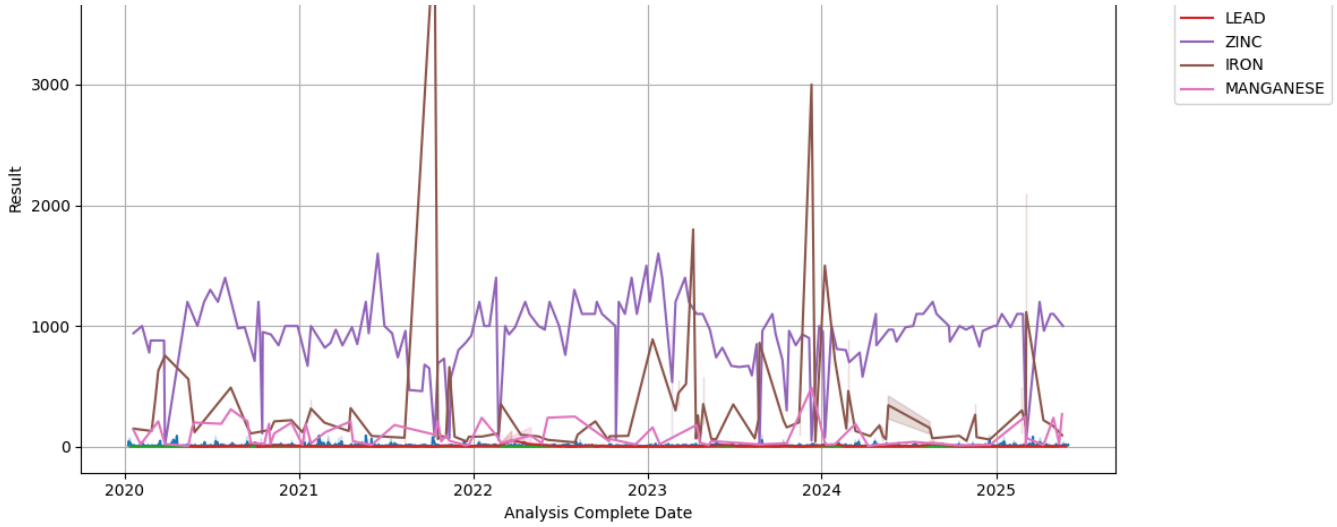

Analyte Result Over Time - CONTRA COSTA
1000
800
Analyte
MANGANESE
IRON
FLUORIDE
LEAD
ARSENIC

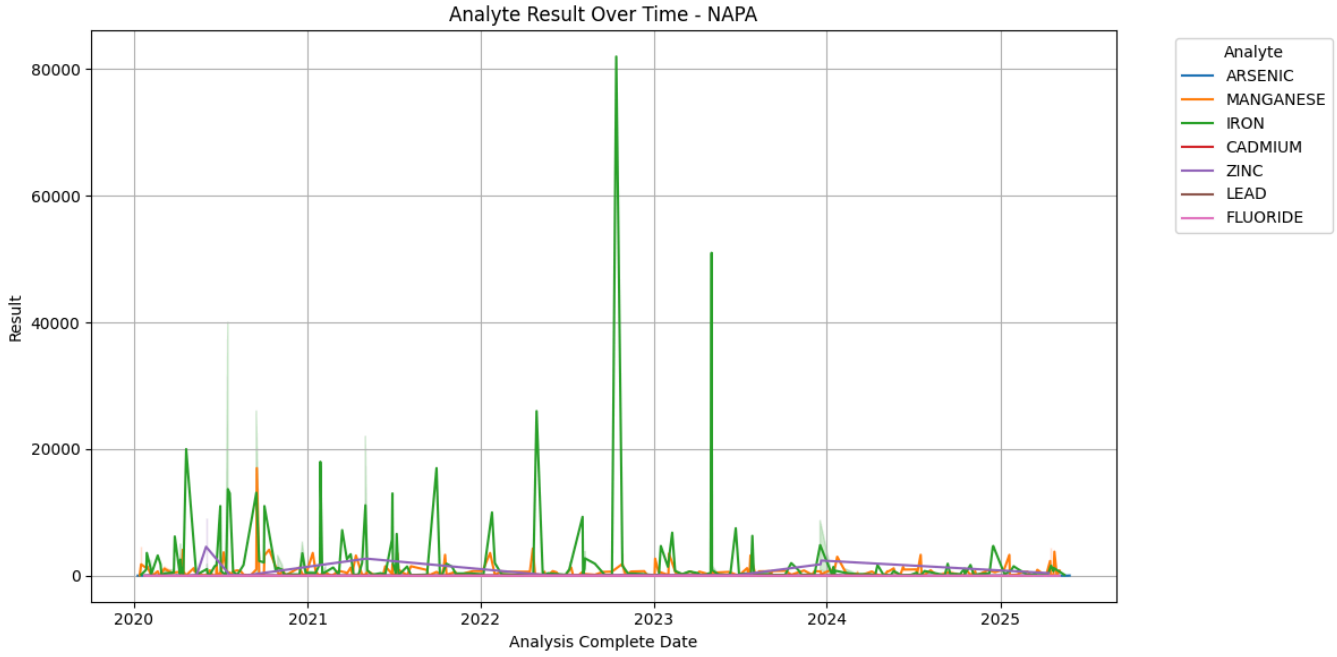

ZINC
CADMIUM
Result
600
400
200
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

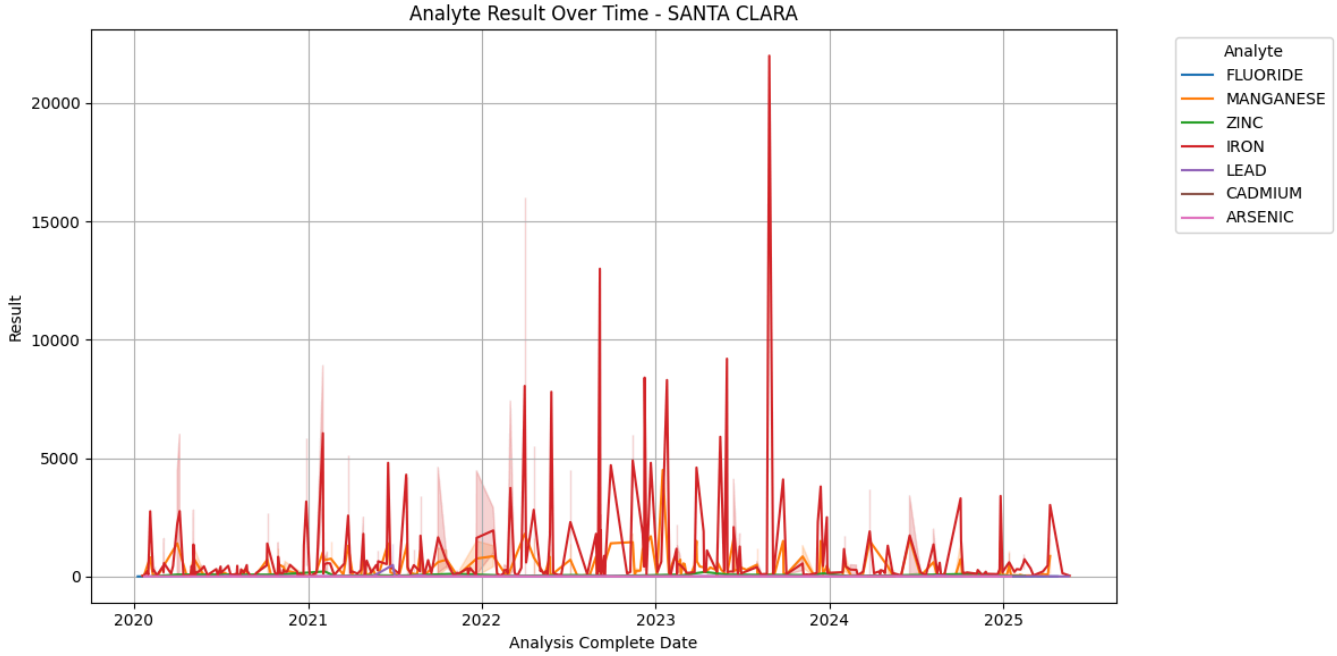

Analyte Result Over Time - YOLO
Analyte
FLUORIDE
IRON
ARSENIC
ZINC
MANGANESE
LEAD
CADMIUM
Result
5000
4000
3000
2000
1000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

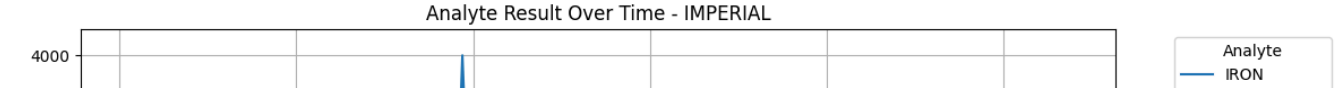

Analyte Result Over Time - MONO
Analyte
ARSENIC
FLUORIDE
MANGANESE
IRON
LEAD
ZINC
CADMIUM
Result
400000
300000
200000
100000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

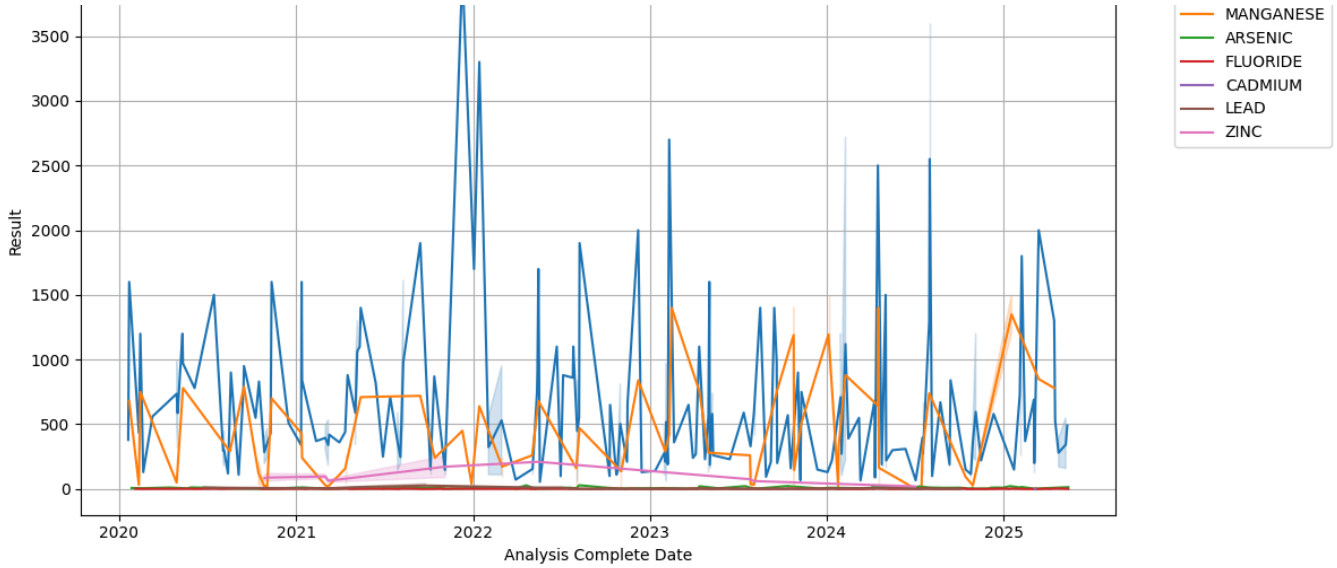

Analyte Result Over Time - KINGS
Analyte
ARSENIC
CADMIUM
FLUORIDE
4000

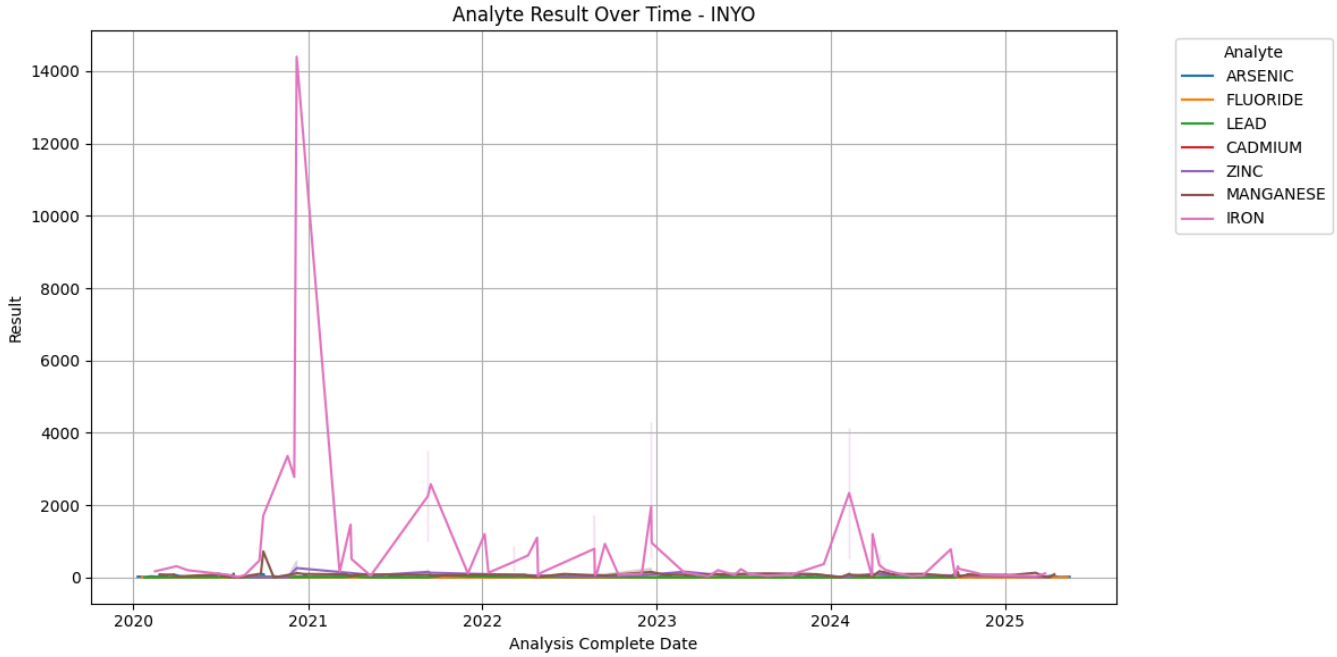

LEAD
ZINC
IRON
MANGANESE
Result
3000
2000
1000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

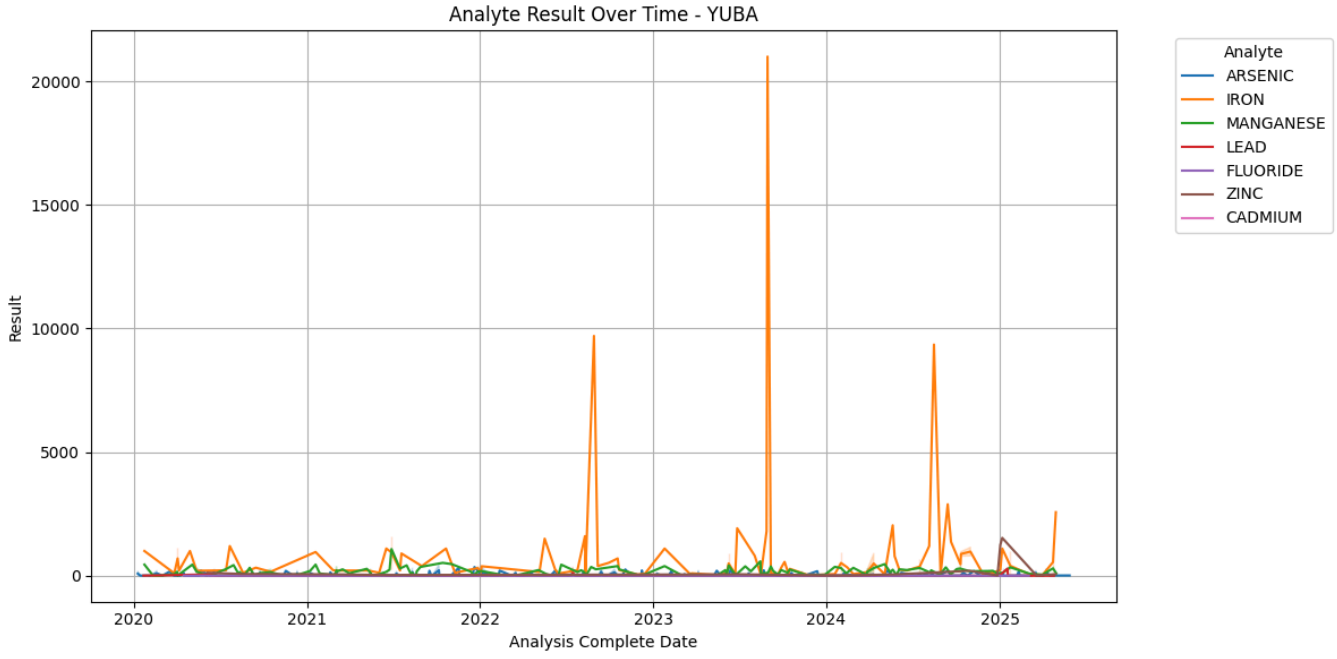

Analyte Result Over Time - NAPA
Analyte
ARSENIC
MANGANESE
IRON
CADMIUM
ZINC
LEAD
FLUORIDE
Result
80000
60000
40000
20000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date


Analyte Result Over Time - SANTA CLARA
Analyte
FLUORIDE
MANGANESE
ZINC
IRON
LEAD
CADMIUM
ARSENIC
Result
20000
15000
10000
5000
0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

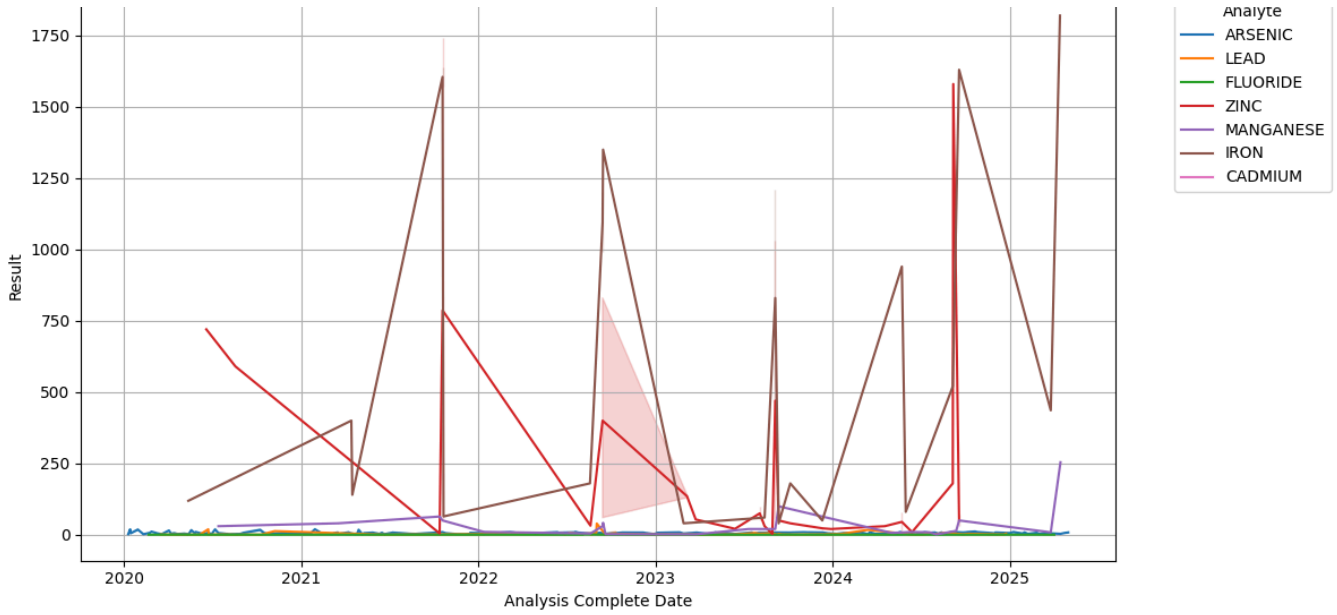

Analyte Result Over Time - IMPERIAL
Analyte
IRON
4000

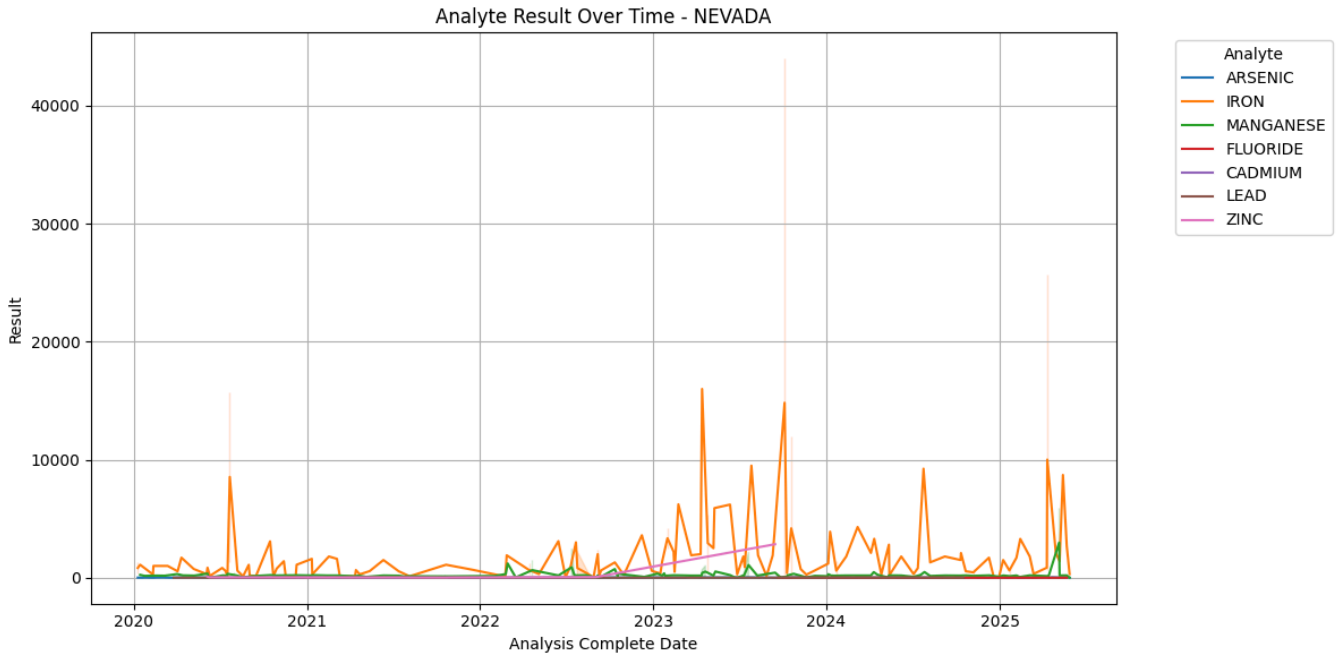

MANGANESE
ARSENIC
FLUORIDE
CADMIUM
LEAD
ZINC
Result
Analysis Complete Date

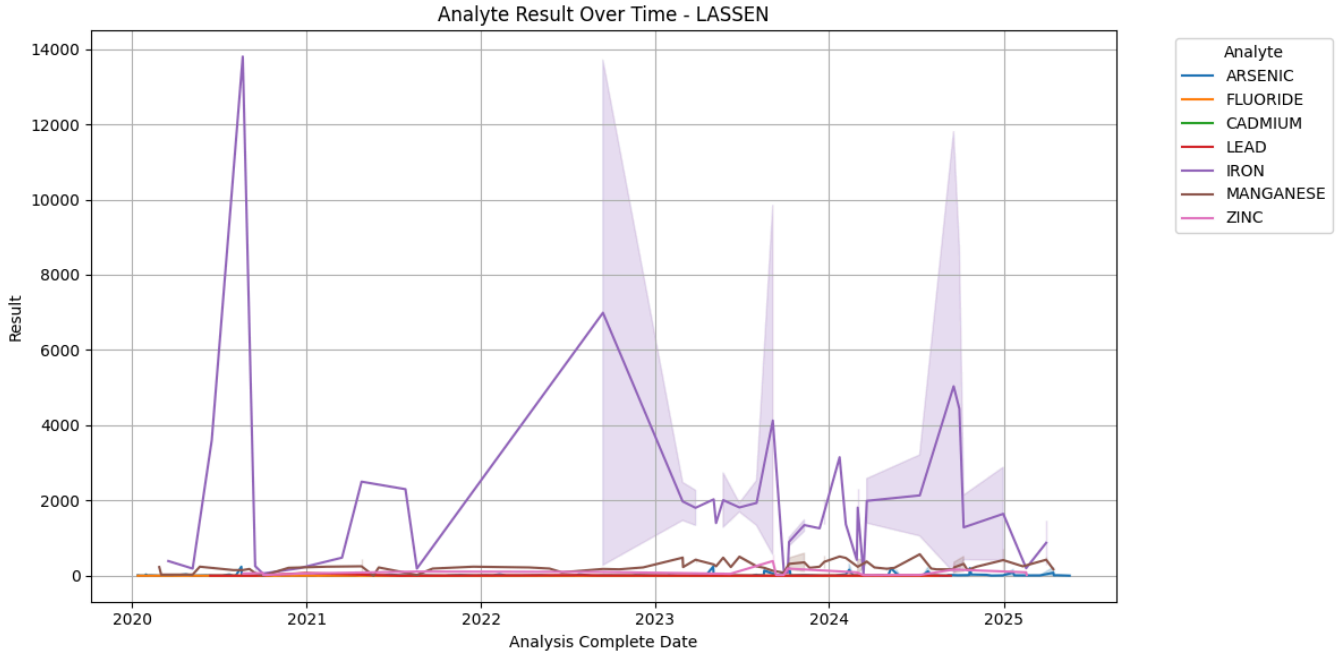

Analyte Result Over Time - INYO
Analyte
ARSENIC
FLUORIDE
LEAD
CADMIUM
ZINC
MANGANESE
IRON
Result
Analysis Complete Date

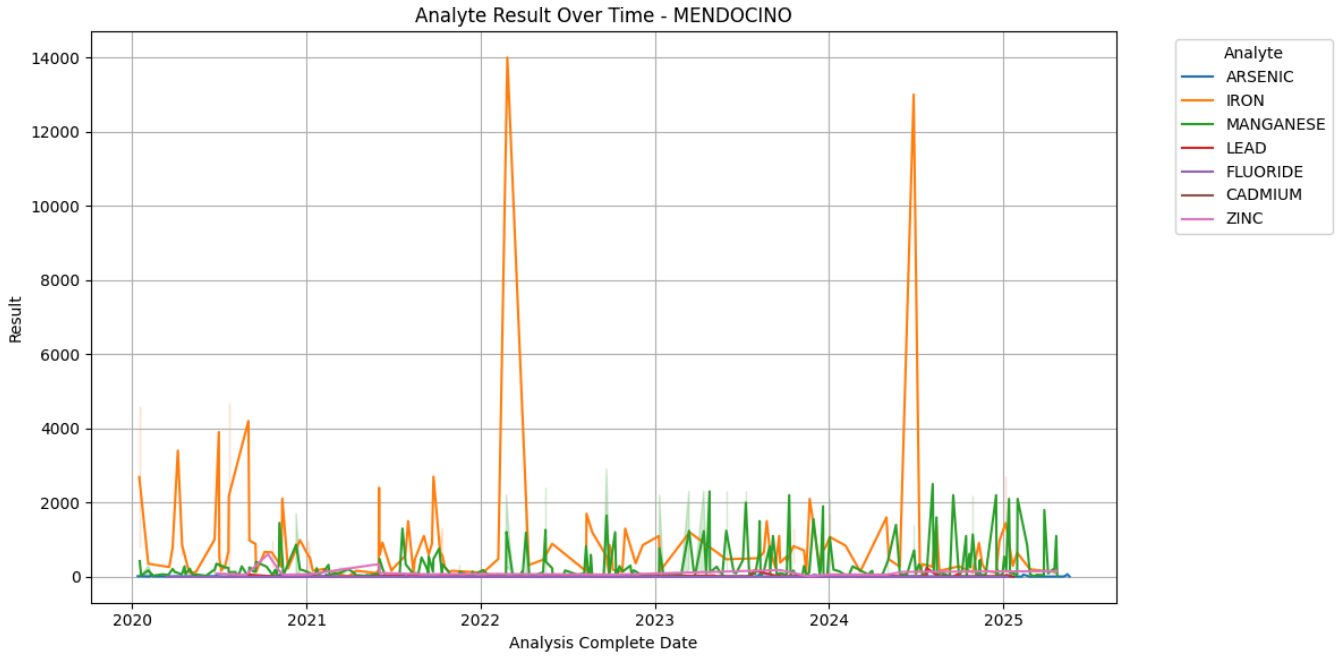

Analyte Result Over Time - YUBA
Analyte
ARSENIC
IRON
MANGANESE
LEAD
FLUORIDE
ZINC
CADMIUM
Result
Analysis Complete Date

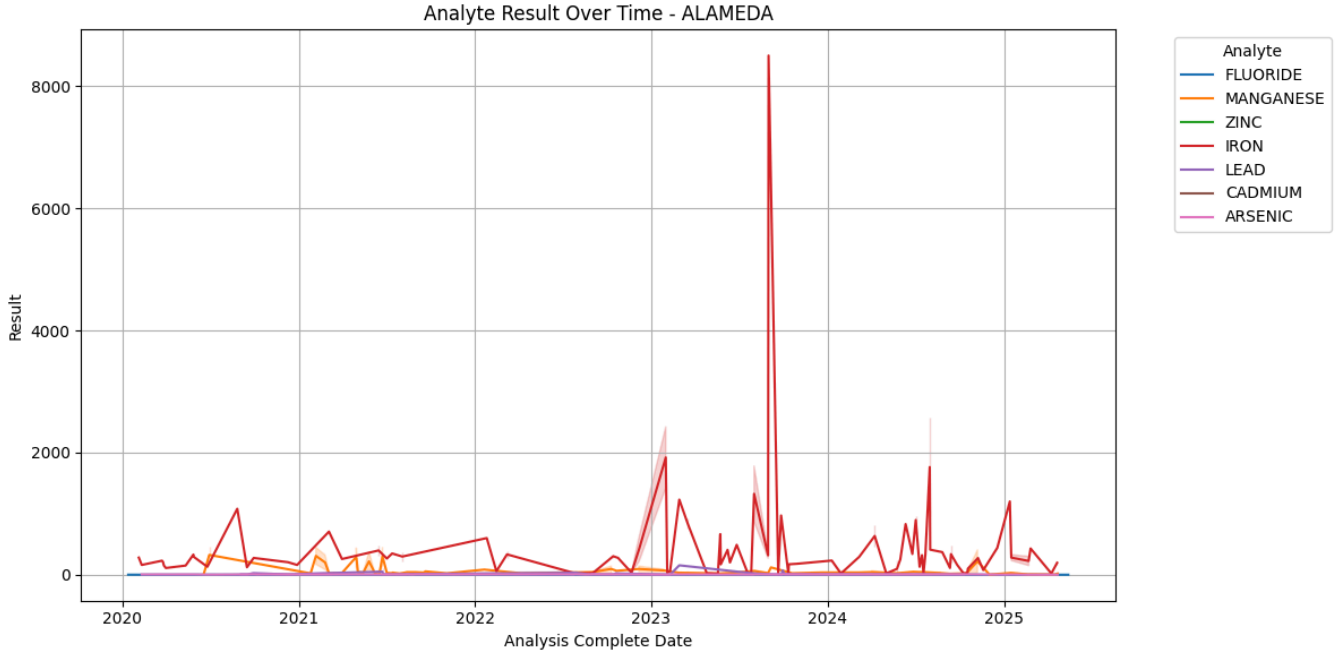

Analyte Result Over Time - TEHAMA

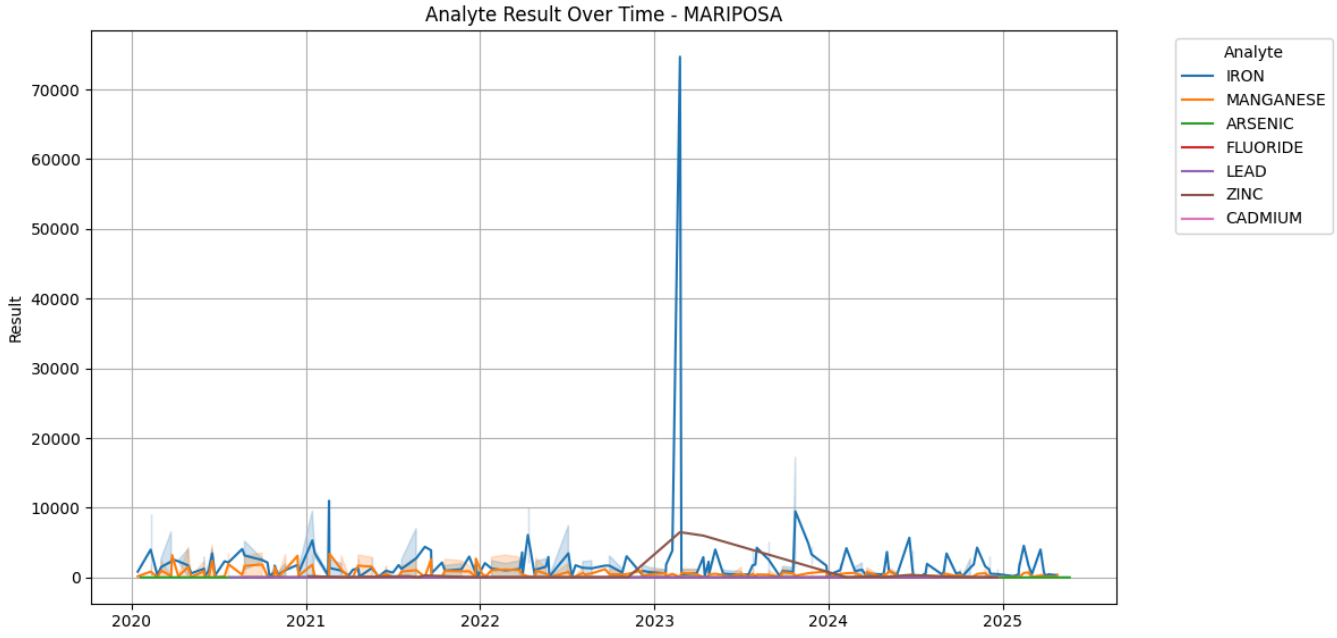

Analyte
ARSENIC
LEAD
FLUORIDE
ZINC
MANGANESE
IRON
CADMIUM
Result
Analysis Complete Date

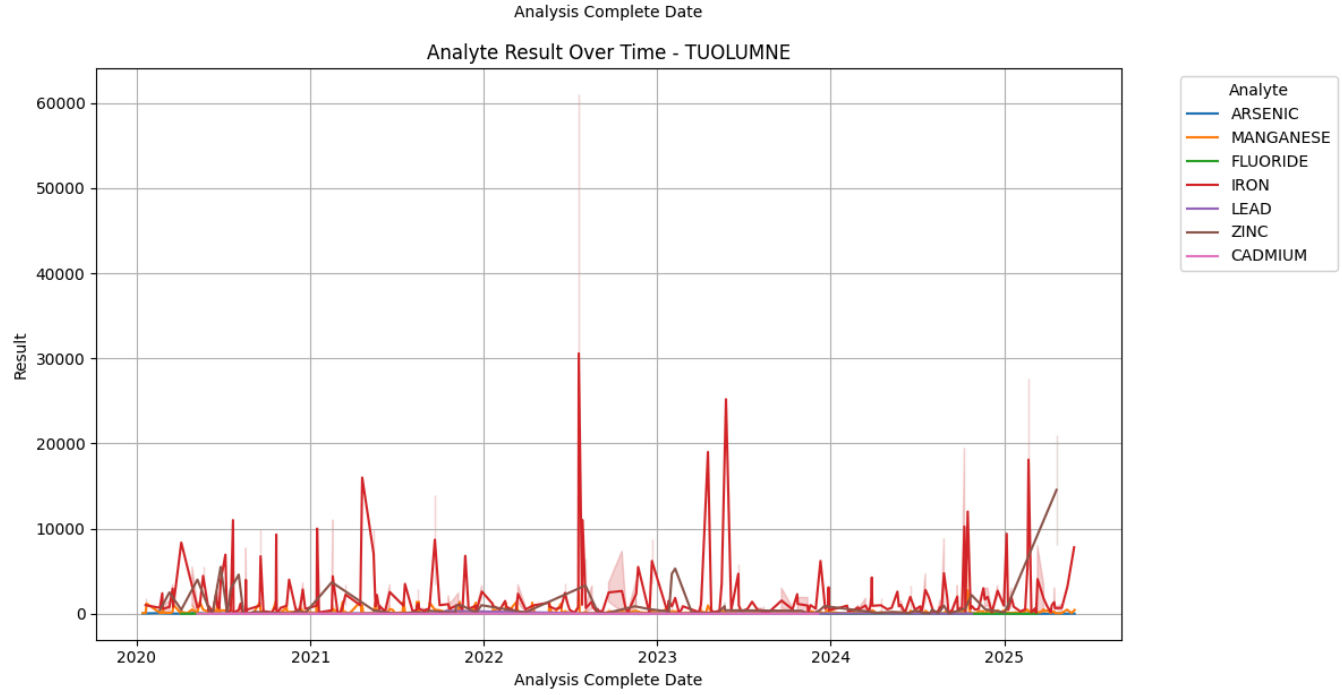

Analyte Result Over Time - NEVADA
Analyte
ARSENIC
IRON
MANGANESE
FLUORIDE
CADMIUM
LEAD
ZINC
Result
Analysis Complete Date

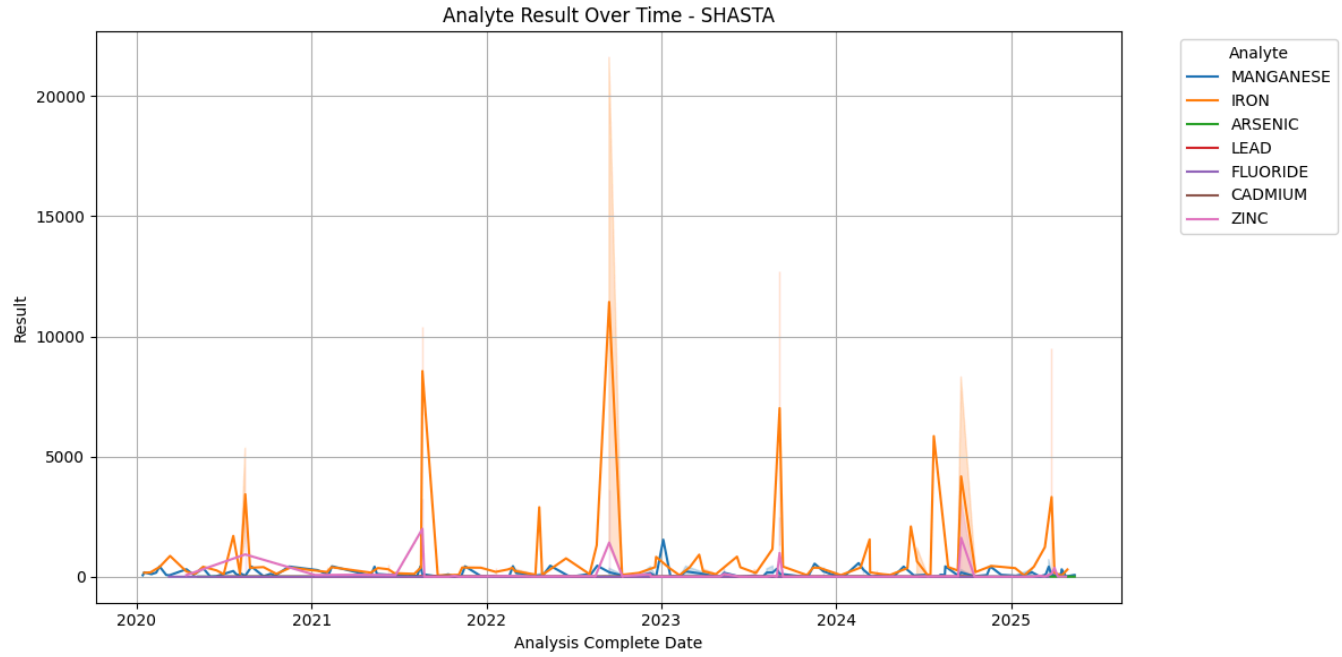

Analyte Result Over Time - LASSEN
Analyte
ARSENIC
FLUORIDE
CADMIUM
LEAD
IRON
MANGANESE
ZINC
Result
Analysis Complete Date

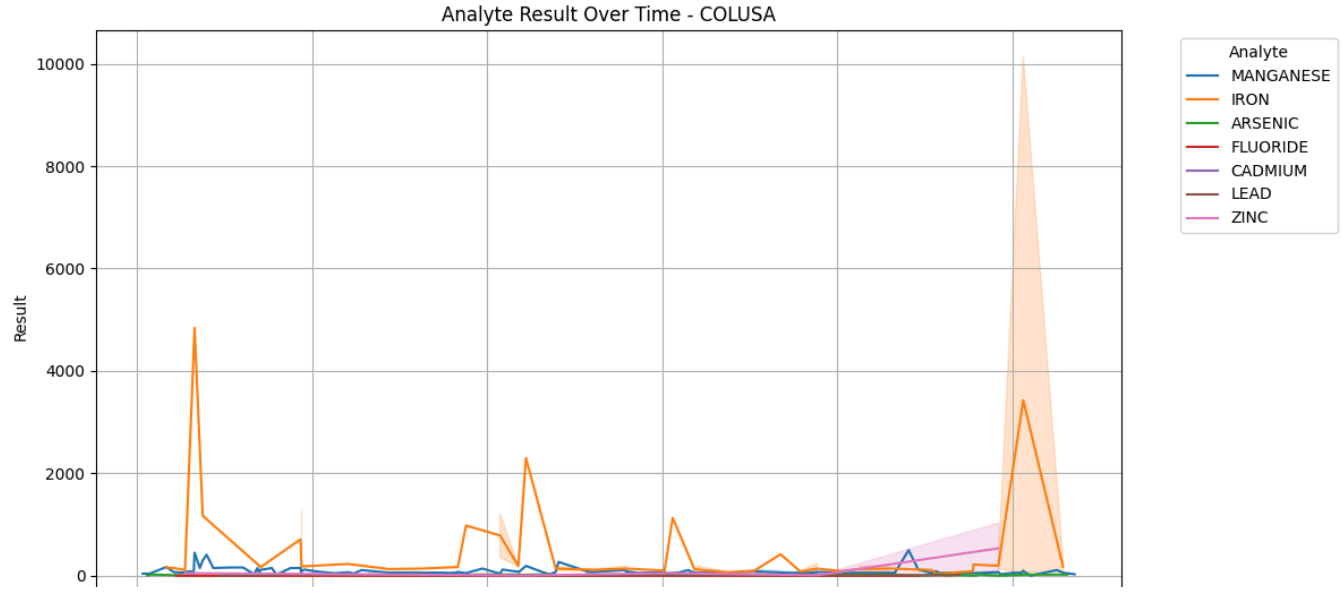

Analyte Result Over Time - MENDOCINO
Result
Analysis Complete Date
Analyte
ARSENIC
IRON
MANGANESE
LEAD
FLUORIDE
CADMIUM
ZINC

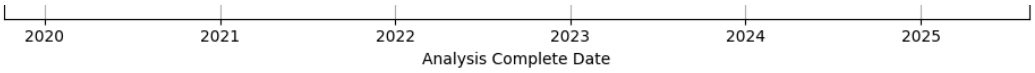

Analyte Result Over Time - ALAMEDA
Result
Analysis Complete Date
Analyte
FLUORIDE
MANGANESE
ZINC
IRON
LEAD
CADMIUM
ARSENIC

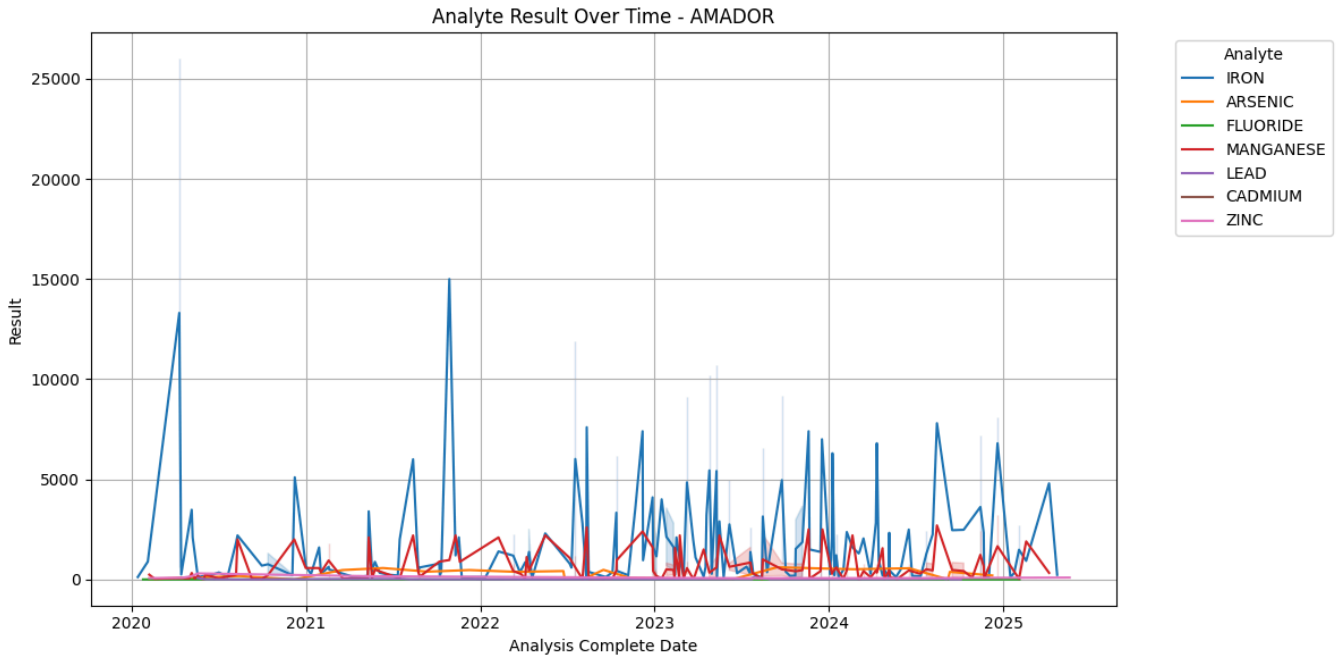

Analyte Result Over Time - MARIPOSA
Result
Analysis Complete Date
Analyte
IRON
MANGANESE
ARSENIC
FLUORIDE
LEAD
ZINC
CADMIUM

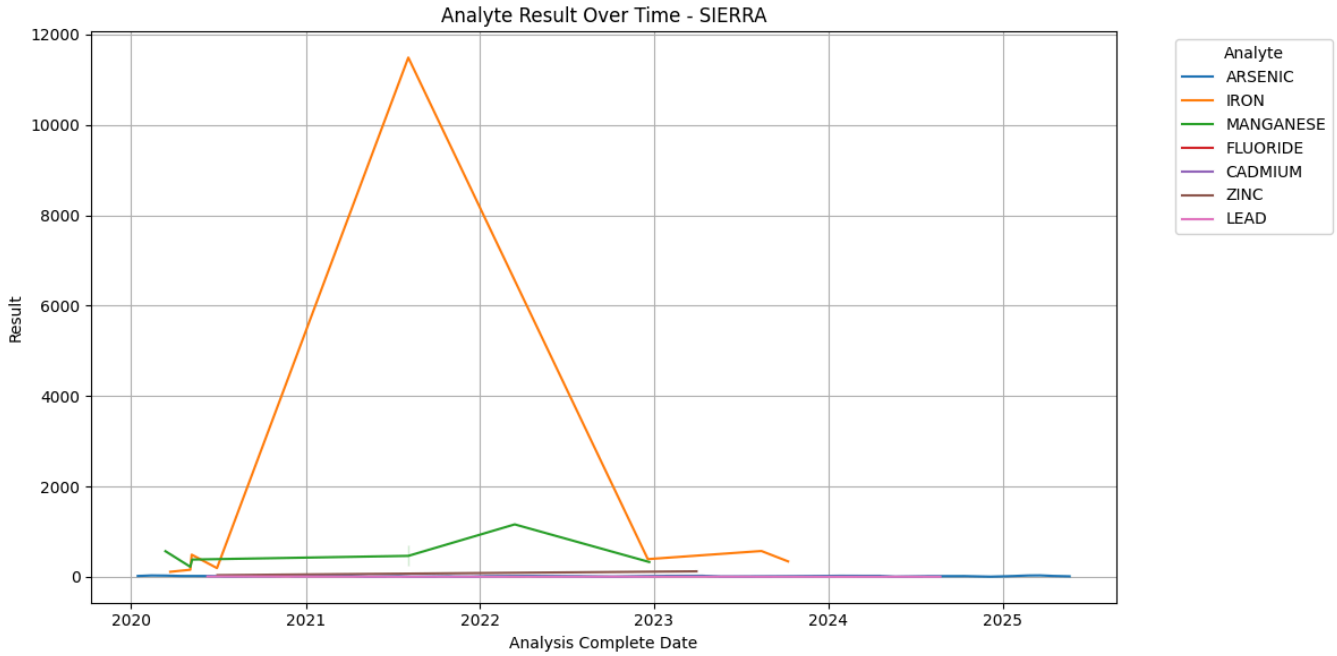

Analysis Complete Date
Analyte Result Over Time - TUOLUMNE
Result
Analysis Complete Date
Analyte
ARSENIC
MANGANESE
FLUORIDE
IRON
LEAD
ZINC
CADMIUM

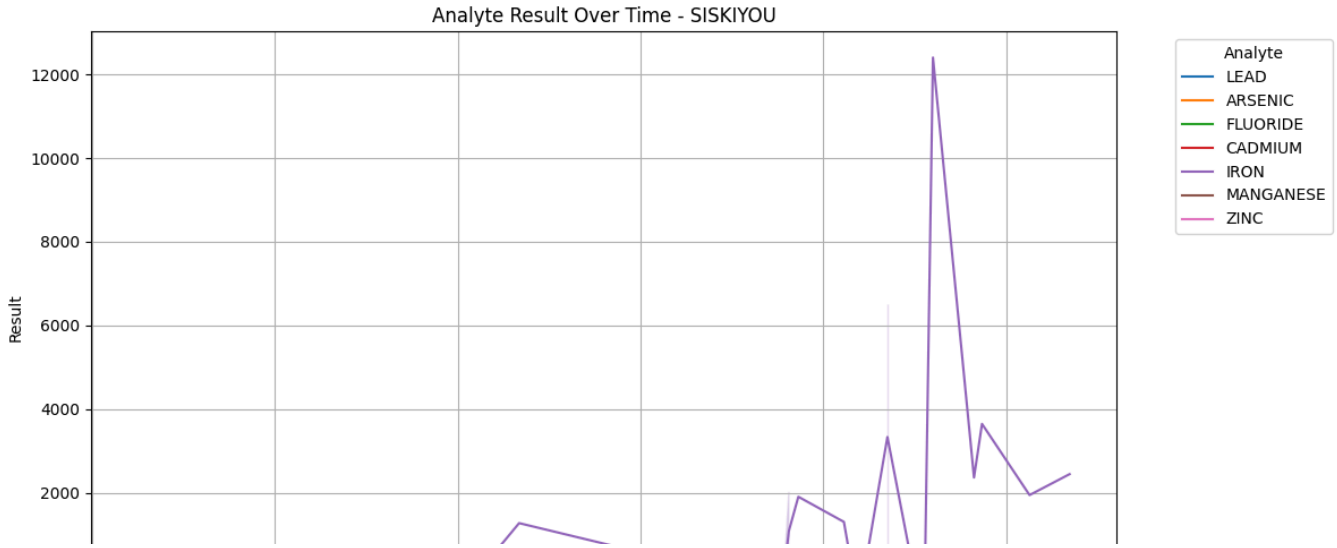

Analyte Result Over Time - SHASTA
Result
Analysis Complete Date
Analyte
MANGANESE
IRON
ARSENIC
LEAD
FLUORIDE
CADMIUM
ZINC

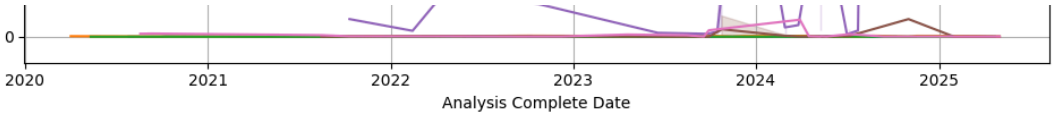

Analyte Result Over Time - COLUSA
Result
Analyte
MANGANESE
IRON
ARSENIC
FLUORIDE
CADMIUM
LEAD
ZINC

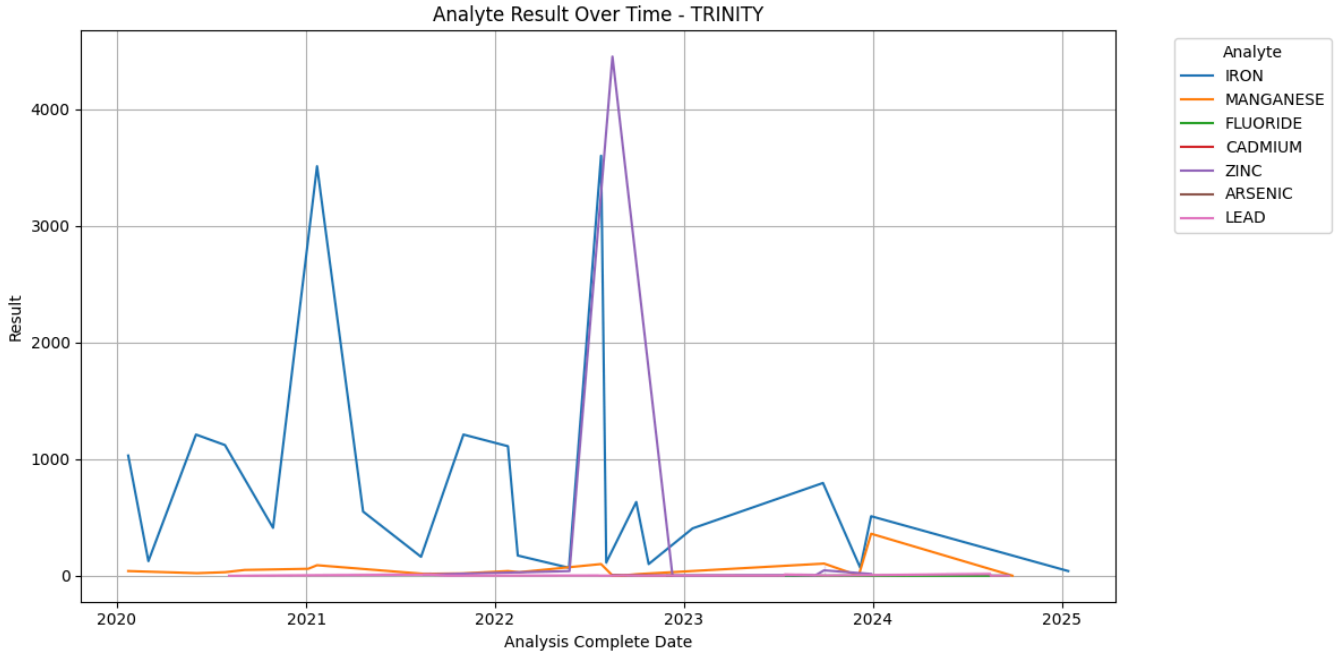

2020
2021
2022
2023
2024
2025
Analysis Complete Date

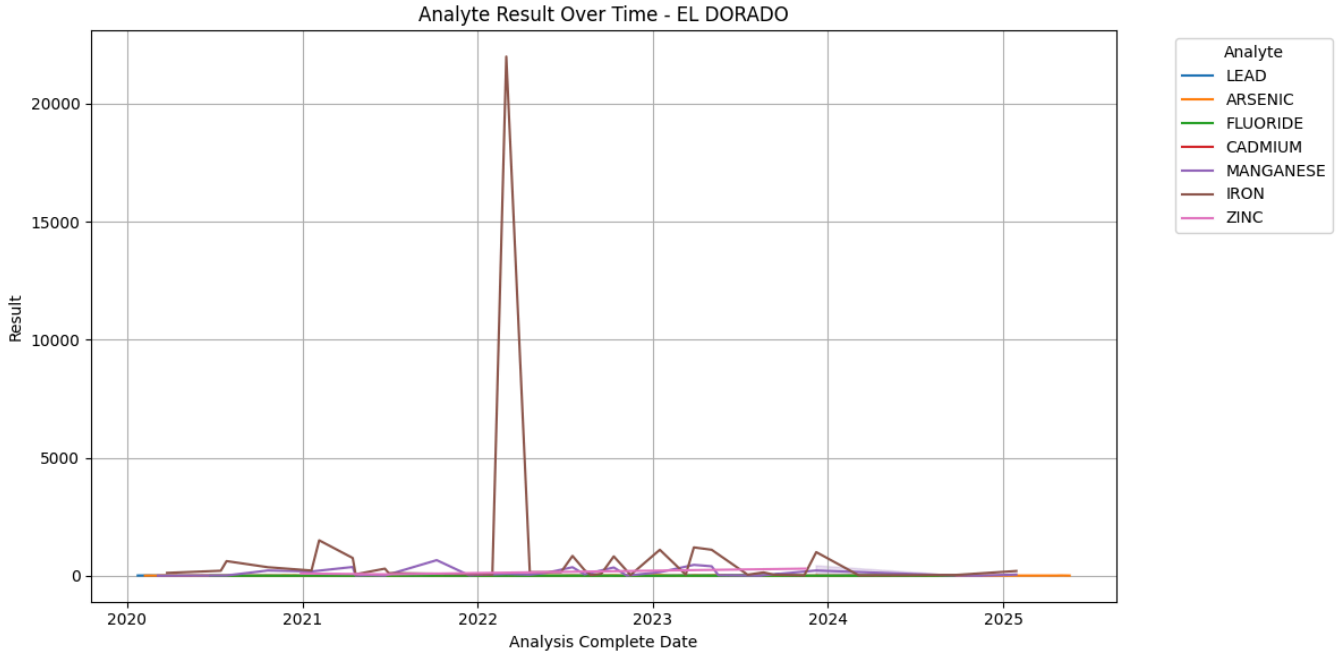

Analyte Result Over Time - AMADOR
Result
Analysis Complete Date
Analyte
IRON
ARSENIC
FLUORIDE
MANGANESE
LEAD
CADMIUM
ZINC

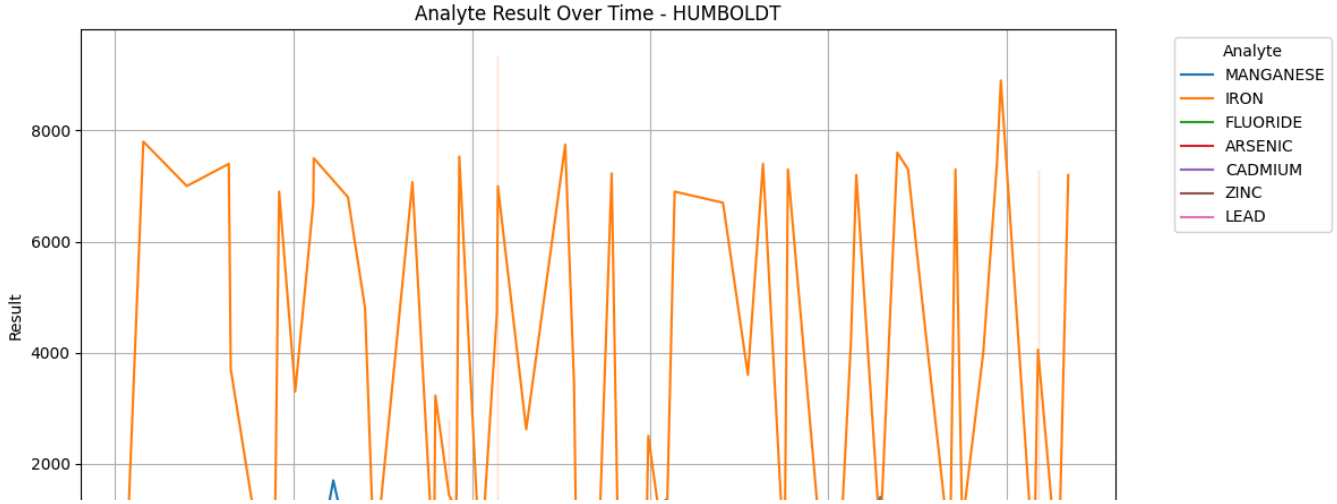

Analyte Result Over Time - SIERRA
Result
Analysis Complete Date
Analyte
ARSENIC
IRON
MANGANESE
FLUORIDE
CADMIUM
ZINC
LEAD

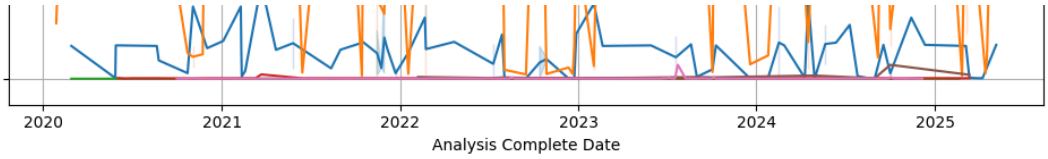

Analyte Result Over Time - SISKIYOU
Result
Analyte
LEAD
ARSENIC
FLUORIDE
CADMIUM
IRON
MANGANESE
ZINC

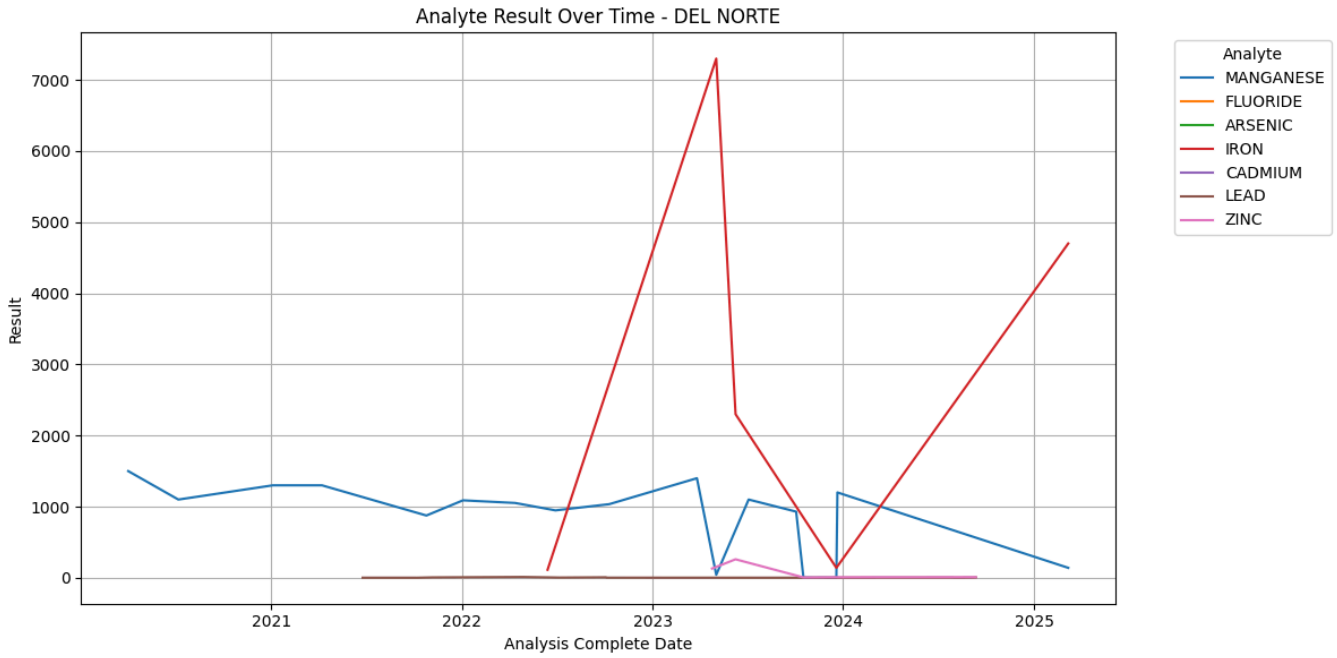

0
2020
2021
2022
2023
2024
2025
Analysis Complete Date

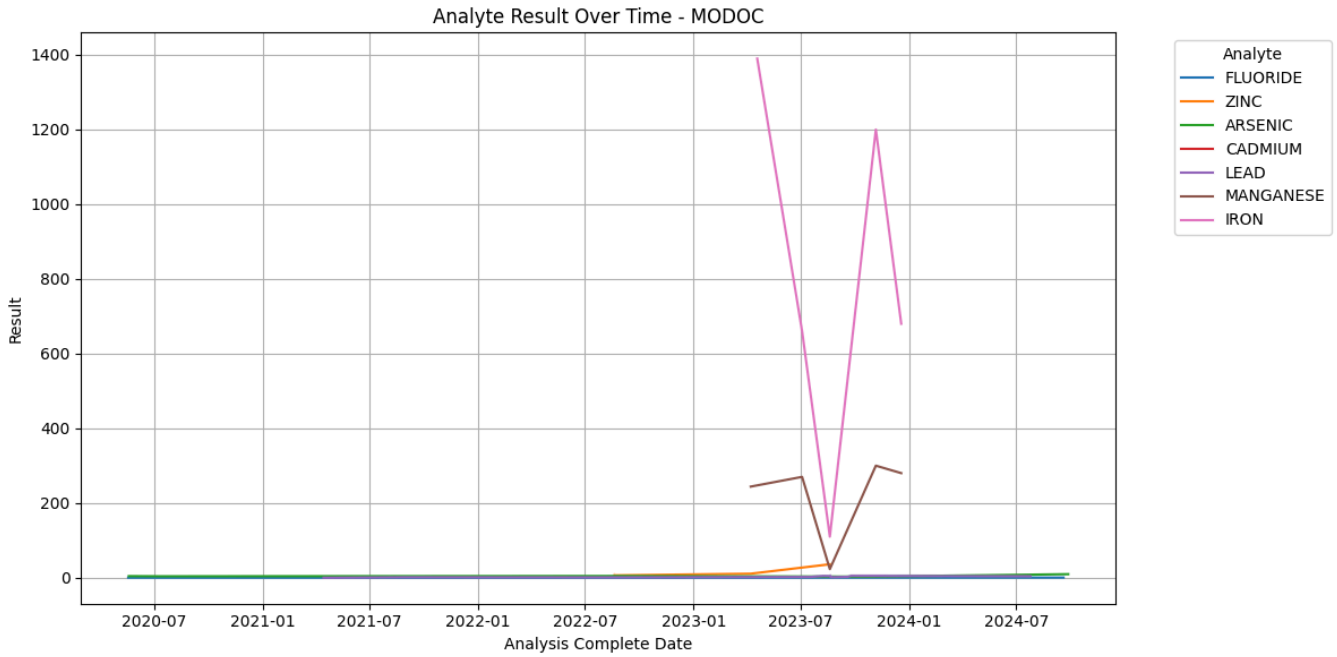

Analyte Result Over Time - TRINITY
Result
Analysis Complete Date
Analyte
IRON
MANGANESE
FLUORIDE
CADMIUM
ZINC
ARSENIC
LEAD

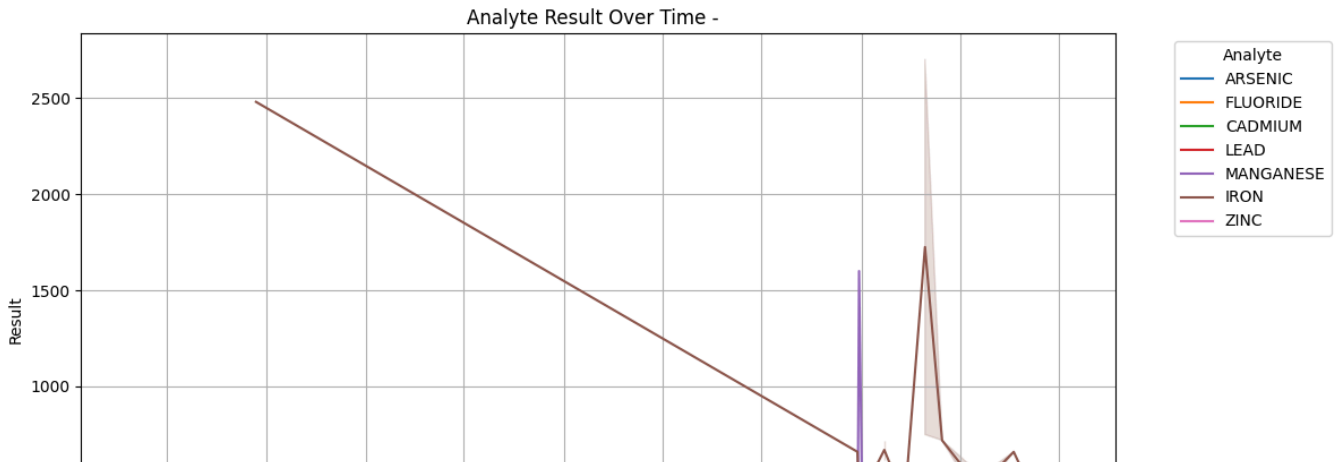

Analyte Result Over Time - EL DORADO
Result
Analysis Complete Date
Analyte
LEAD
ARSENIC
FLUORIDE
CADMIUM
MANGANESE
IRON
ZINC

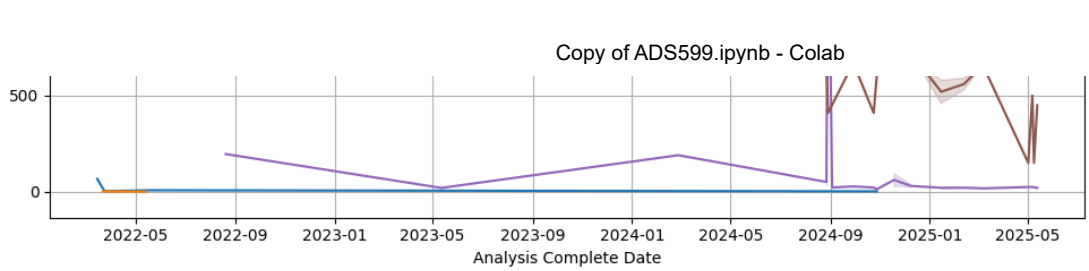

Analyte Result Over Time - HUMBOLDT
Result
Analyte
MANGANESE
IRON
FLUORIDE
ARSENIC
CADMIUM
ZINC
LEAD


0
2020
2021
2022
2023
2024
2025
Analysis Complete Date


Analyte Result Over Time - DEL NORTE
Result
Analysis Complete Date
Analyte
MANGANESE
FLUORIDE
ARSENIC
IRON
CADMIUM
LEAD
ZINC


Analyte Result Over Time - MODOC
Result
Analysis Complete Date
Analyte
FLUORIDE
ZINC
ARSENIC
CADMIUM
LEAD
MANGANESE
IRON


Analyte Result Over Time -
Result
Analyte
ARSENIC
FLUORIDE
CADMIUM
LEAD
MANGANESE
IRON
ZINC

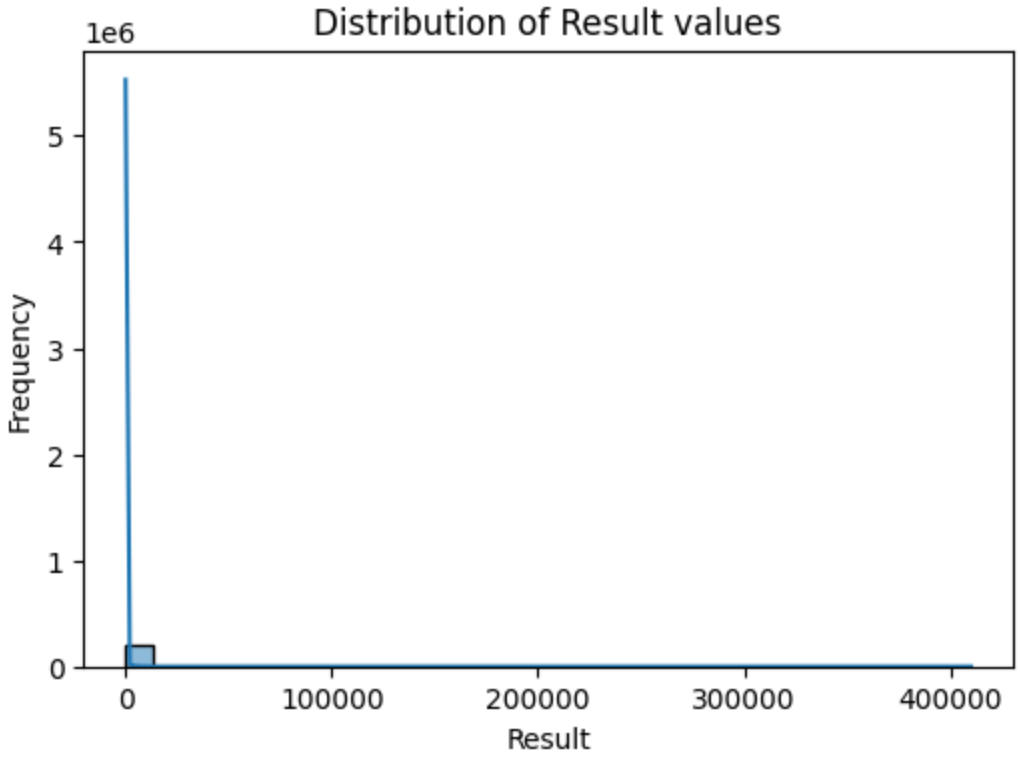

500
0
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09
2025-01
2025-05
Analysis Complete Date

```
df = df.drop_duplicates()

critical_cols = []
for col in ['Analysis Complete Date', 'Analyte', 'Result']:
    if col in df.columns:
        critical_cols.append(col)
if critical_cols:
    df = df.dropna(subset=critical_cols)

if 'Analysis Complete Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Analysis Complete Date'])

if 'Result' in df.columns and df['Result'].dtype != object:
    df = df[df['Result'] >= 0]

print("Data shape after cleaning:", df.shape)
print("Remaining missing values:", df.isna().sum().to_dict())
```

```
Data shape after cleaning: (210670, 58)
Remaining missing values: {'Row #': 0, 'Regulating Agency': 0, 'Water System #': 0, 'Sys
```

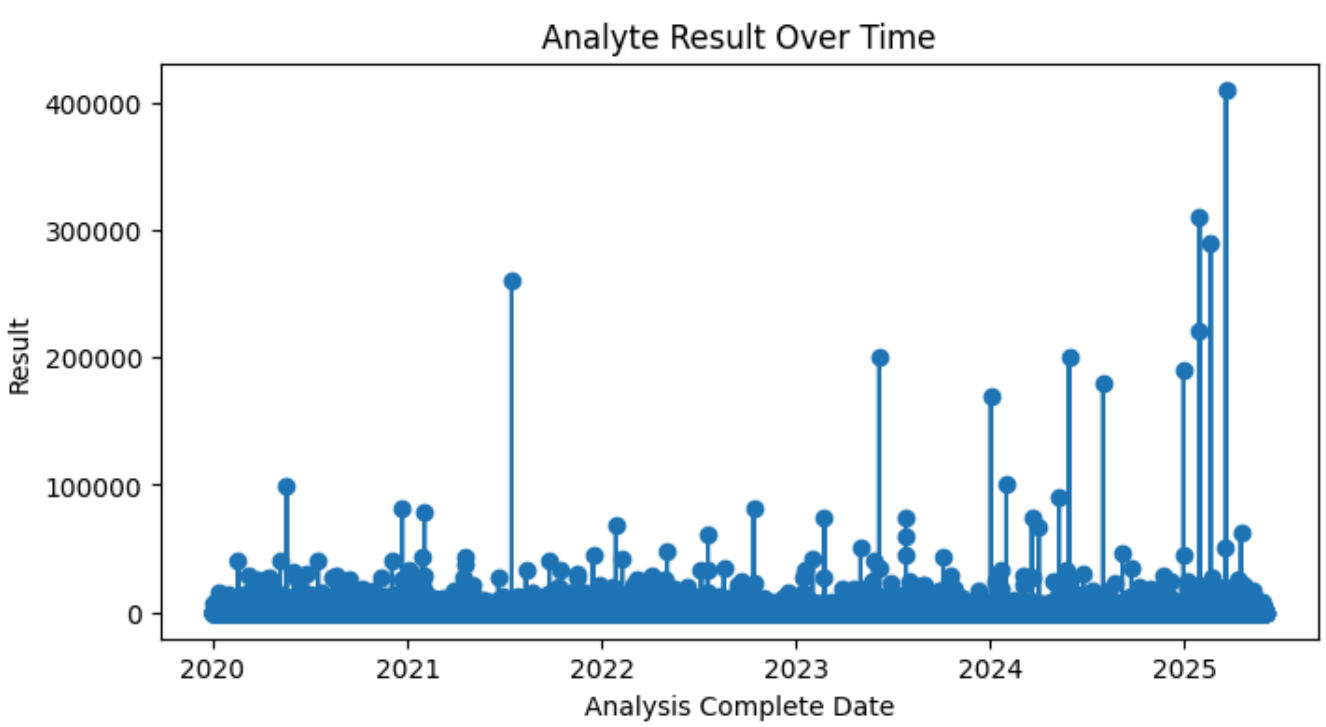
```
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(6,4))
sns.histplot(df['Result'], kde=True, bins=30)
plt.title("Distribution of Result values")
plt.xlabel("Result")
plt.ylabel("Frequency")
plt.show()

if 'Analyte' in df.columns:
    plt.figure(figsize=(8,5))
    sns.boxplot(x='Analyte', y='Result', data=df)
    plt.title("Result distribution by Analyte")
    plt.xticks(rotation=90)
```

```
    plt.ylabel("Result")
    plt.xlabel("Analyte")
    plt.show()
```




```
if 'Analysis Complete Date' in df.columns:
    df = df.sort_values('Analysis Complete Date')
    if 'Analyte' in df.columns:
        analytes = df['Analyte'].unique()
        n_analytes = len(analytes)
        plt.figure(figsize=(8, 3*n_analytes))
        for i, analyte in enumerate(analytes, start=1):
            plt.subplot(n_analytes, 1, i)
            subset = df[df['Analyte'] == analyte]
            plt.plot(subset['Analysis Complete Date'], subset['R
            plt.title(f"{analyte} Trend Over Time")
            plt.xlabel("Analysis Complete Date")
            plt.ylabel("Result")
        plt.tight_layout()
        plt.show()
    else:
```

```
    plt.figure(figsize=(8,4))
    plt.plot(df['Analysis Complete Date'], df['Result'], mar
    plt.title("Analyte Result Over Time")
    plt.xlabel("Analysis Complete Date")
    plt.ylabel("Result")
    plt.show()
```




```
if 'Analyte' in df.columns:
    if 'Location' in df.columns or 'Station' in df.columns:
        loc_col = 'Location' if 'Location' in df.columns else 'S
        pivot_df = df.pivot_table(values='Result', index=['Analy
    else:
        pivot_df = df.pivot_table(values='Result', index='Analys
    corr_matrix = pivot_df.corr()
    print("Correlation matrix between analytes:\n", corr_matrix)
    if corr_matrix.shape[0] <= 20:
        plt.figure(figsize=(6,5))
        sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="co
        plt.title("Correlation between Analytes")
        plt.show()
```

```
if 'Analysis Complete Date' in df.columns:
    date_col = 'Analysis Complete Date'
elif 'Date' in df.columns:
    date_col = 'Date'
else:
    raise ValueError("No valid date column found.")

df[date_col] = pd.to_datetime(df[date_col], errors='coerce')
df = df.dropna(subset=[date_col])  # remove rows with invalid da

df = df.sort_values(date_col)
ts_data = df.set_index(date_col)['Result']

ts_monthly = ts_data.resample('MS').mean()


ts_monthly = ts_monthly.interpolate(method='linear').dropna()

print("Time series start:", ts_monthly.index.min(), " end:", ts_
print("Total time points after resampling:", len(ts_monthly))
```

```
Time series start: 2020-01-01 00:00:00  end: 2025-06-01 00:00:00
Total time points after resampling: 66
```

```
forecast_horizon = 12
if len(ts_monthly) > forecast_horizon:
    train_ts = ts_monthly.iloc[:-forecast_horizon]
    test_ts = ts_monthly.iloc[-forecast_horizon:]
else:
    train_ts = ts_monthly
    test_ts = pd.Series([], dtype=float)

print("Training periods:", len(train_ts), "Testing periods:", le
if not test_ts.empty:
    print("Training data range:", train_ts.index.min(), "to", tr
```

```
print("Testing data range:", test_ts.index.min(), "to", test
```

```
Training periods: 54 Testing periods: 12
Training data range: 2020-01-01 00:00:00 to 2024-06-01 00:00:00
Testing data range: 2024-07-01 00:00:00 to 2025-06-01 00:00:00
```

```
import statsmodels.api as sm
import warnings
warnings.filterwarnings("ignore")

p = d = q = range(0, 2)
pdq = [(x,y,z) for x in p for y in d for z in q]
seasonal_pdq = [(x,y,z, 12) for x in p for y in d for z in q]

best_aic = float("inf")
best_order = None
best_seasonal_order = None

for order in pdq:
    for s_order in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(train_ts, order=orde
                                            enforce_stationarity
            results = mod.fit(disp=False)
            if results.aic < best_aic:
                best_aic = results.aic
                best_order = order
                best_seasonal_order = s_order
        except Exception as ex:
            continue

print(f"Best ARIMA order: {best_order}, Best seasonal order: {be

model = sm.tsa.statespace.SARIMAX(train_ts, order=best_order, se
                                  enforce_stationarity=False, en
results = model.fit()
print("\nModel summary:")
print(results.summary())
```

```
Best ARIMA order: (0, 1, 1), Best seasonal order: (1, 1, 1, 12), AIC: 270.53

Model summary:
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                Result   No. Observations:
Model:            SARIMAX(0, 1, 1)x(1, 1, 1, 12)   Log Likelihood                -131.2
```

```
Date:                            Wed, 06 Aug 2025   AIC                            270.5
Time:                                   04:51:20   BIC                            275.7
Sample:                              01-01-2020   HQIC                           272.6
                                   - 06-01-2024
Covariance Type:                            opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -1.0000   1839.828     -0.001      1.000   -3606.997    3604.997
ar.S.L12      -0.5191      0.386     -1.346      0.178      -1.275       0.237
ma.S.L12      -0.1304      0.416     -0.313      0.754      -0.946       0.685
sigma2       887.1583   1.63e+06      0.001      1.000    -3.2e+06     3.2e+06
===================================================================================
Ljung-Box (L1) (Q):                   1.74   Jarque-Bera (JB):                 2.90
Prob(Q):                              0.19   Prob(JB):                         0.23
Heteroskedasticity (H):               2.37   Skew:                             0.80
Prob(H) (two-sided):                  0.22   Kurtosis:                         3.02
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

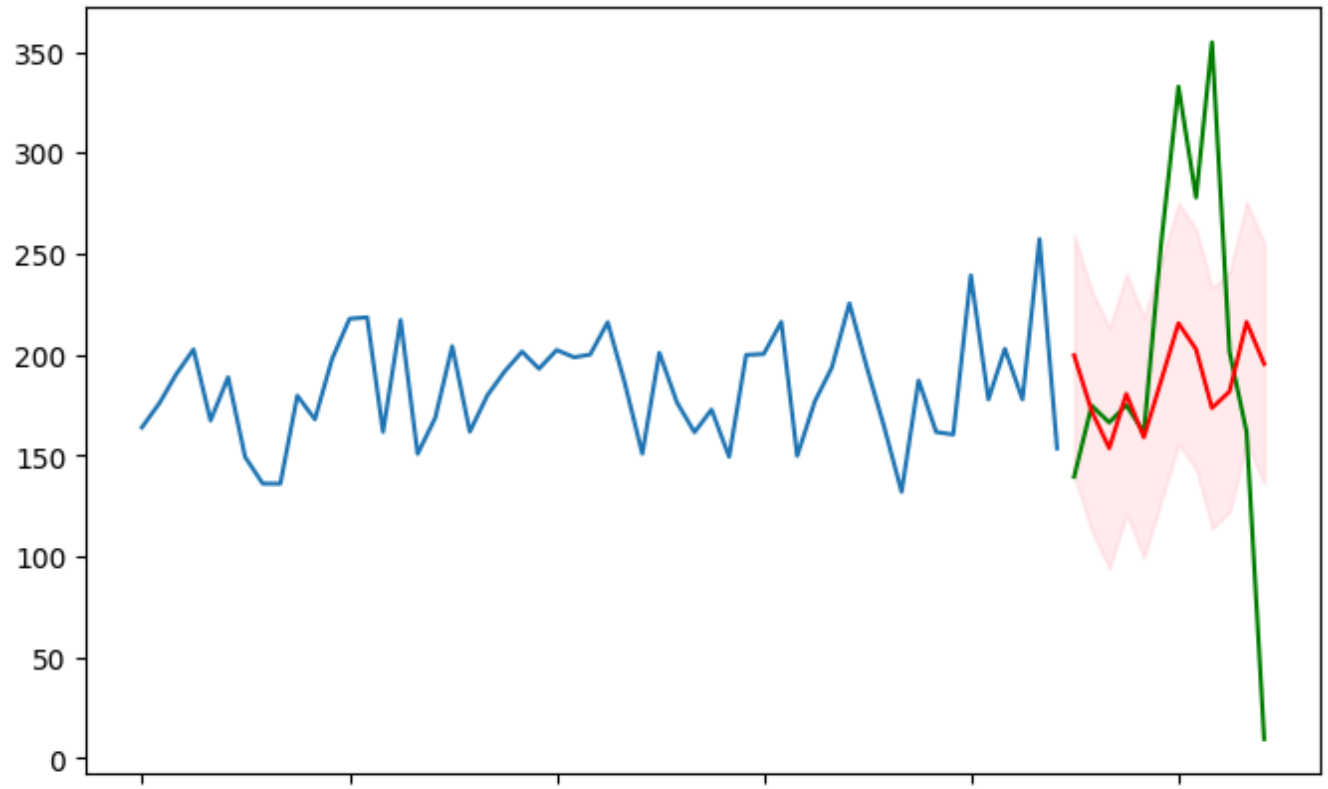
```
if not test_ts.empty:
    forecast_steps = len(test_ts)
else:
    forecast_steps = 12

pred = results.get_forecast(steps=forecast_steps)
pred_mean = pred.predicted_mean
pred_conf = pred.conf_int()
if not test_ts.empty:
    pred_mean.index = test_ts.index
    pred_conf.index = test_ts.index

    from sklearn.metrics import mean_absolute_error, mean_square
    mae = mean_absolute_error(test_ts, pred_mean)
    mse = mean_squared_error(test_ts, pred_mean)
    rmse = np.sqrt(mse)
    print(f"\nForecast evaluation on test data:")
    print(f"Mean Absolute Error: {mae:.3f}")
    print(f"Root Mean Squared Error: {rmse:.3f}")
```

```
Forecast evaluation on test data:
Mean Absolute Error: 65.360
```

```
Root Mean Squared Error: 90.590
```

```
plt.figure(figsize=(8,5))
plt.plot(train_ts.index, train_ts.values, label='Training Data')
if not test_ts.empty:
    plt.plot(test_ts.index, test_ts.values, label='Actual Test D
    plt.plot(pred_mean.index, pred_mean.values, label='Forecast'
    plt.fill_between(pred_conf.index, pred_conf.iloc[:,0], pred_
else:
    forecast_index = pd.date_range(start=train_ts.index.max(), p
    plt.plot(train_ts.index, train_ts.values, label='Training Da
    plt.plot(forecast_index, pred_mean.values, label='Forecast',
    plt.fill_between(forecast_index, pred_conf.iloc[:,0], pred_c

plt.title(f"Forecast for {target_analyte if target_analyte else
plt.xlabel("Date")
plt.ylabel("Result")
plt.legend()
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-2857051117.py in <cell line: 0>()
     14     plt.fill_between(forecast_index, pred_conf.iloc[:,0], pred_conf.iloc[:,1],
color='pink', alpha=0.3, label='Confidence Interval')
     15
---> 16 plt.title(f"Forecast for {target_analyte if target_analyte else 'Analyte'} over
Time")
     17 plt.xlabel("Date")
     18 plt.ylabel("Result")

NameError: name 'target_analyte' is not defined
```



Next steps: Explain error

```
df_heatmap = ts_monthly.to_frame(name='Result')
df_heatmap['Year'] = df_heatmap.index.year
df_heatmap['Month'] = df_heatmap.index.month

heatmap_data = df_heatmap.pivot(index='Year', columns='Month', '

plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data, annot=True, fmt=".1f", cmap="YlGnBu",
plt.title(f"Monthly Average of {target_analyte} Over Years")
plt.xlabel("Month")
```

```
plt.ylabel("Year")
plt.xticks(ticks=np.arange(12) + 0.5, labels=["Jan", "Feb", "Ma
                                              "Jul", "Aug", "Se
plt.yticks(rotation=0)
plt.tight_layout()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-687813996.py in <cell line: 0>()
     10 plt.figure(figsize=(12, 6))
     11 sns.heatmap(heatmap_data, annot=True, fmt=".1f", cmap="YlGnBu", linewidths=0.
linecolor='gray')
---> 12 plt.title(f"Monthly Average of {target_analyte} Over Years")
     13 plt.xlabel("Month")
     14 plt.ylabel("Year")

NameError: name 'target_analyte' is not defined
```