

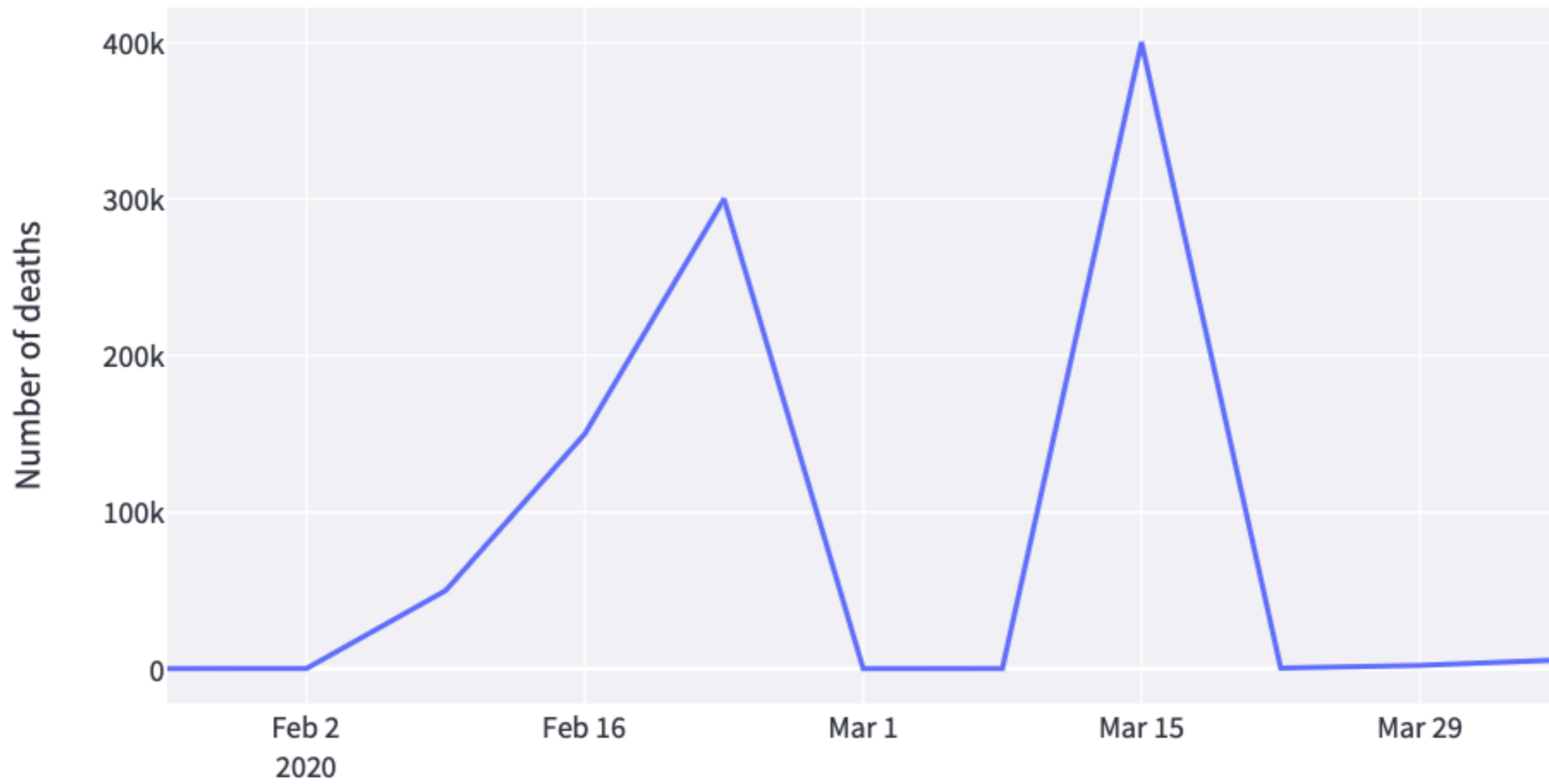
CS 649 Big Data: Tools and Methods
Spring Semester, 2022
Doc 21 Assignment 2 Dates
Mar 22, 2022

Copyright ©, All rights reserved. 2022 SDSU & Roger Whitney,
5500 Campanile Drive, San Diego, CA 92182-7700 USA.
OpenContent (<http://www.opencontent.org/opl.shtml>) license
defines the copyright on this document.

```
positive_line = positive.iloc[:, 8:750]
positive_line.columns = pd.to_datetime(positive_line.columns)
lst = []
week_num = 0
for i in range(0, 742, 7):
    x = positive_line.iloc[:, i:i + 8].sum().sum()
    lst.append(x)
    week_num += 1
st.title("Assignment 2 - Covid Analysis")
st.header("Covid-19 New Cases")
st.line_chart(lst)
```

```
# dropping days to ensure we have full weeks
drop_days = ['2020-01-22', '2020-01-23', '2020-01-24',
             '2020-01-25', '2022-02-06', '2022-02-07']
```

Weekly US Covid Deaths



Week

2020 - 02 - 02



2020 - 01 - 26

2022 - 02 - 06

Map of Covid-19 deaths per 100k people for each county in the USA. Spans between Sunday, 2020-01-26 and Saturday, 2022-02-05.

```
KeyError: "['1/22/2020', '1/23/2020', '1/24/2020', '1/25/2020', '2/6/2022', '2/7/2022'] not found in axis"
```

```
Traceback:
```

```
def build_confirmed_weekly_dataframe():
    question_1 = covid_confirmed.drop(['countyFIPS', 'County Name', 'State', 'StateFIPS'], axis =
1).diff(axis=1)
    question_1 = question_1.fillna(0)
    question_1 = question_1.sum().to_frame("Cases").reset_index()
    question_1 = question_1.rename(columns = {'index': 'Date'})
    question_1['Date'] = pd.to_datetime(question_1['Date'])
    question_1 = question_1.drop([0, 1, 2, 3, 746, 747])
    question_1 = question_1.groupby( pd.Grouper(key='Date', freq='W-SUN'))['Cases'].sum().to_frame()
    return question_1
```

```
date_range = pd.period_range(start = '2020-01-22', end = '2022-02-07', freq = 'W-SAT')  
date_range = date_range.map(str)  
date_range = date_range.str.split('/').str[0]
```



```
def create_slider() :
    death_df = create_df_by_county(ImportFile.get_deaths_file())
    cases_df = create_df_by_county(ImportFile.get_cases_file())

    minn = cases_df.columns[1]
    maxx = cases_df.columns[-2]

    start_time = st.slider(
        "Select Week",
        min_value=minn.to_pydatetime(),
        max_value=maxx.to_pydatetime(),
        value=death_df.columns[100].to_pydatetime(),
        format="YYYY-MM-DD",
        step=timedelta(days=7)),

    st.write("Start date:", start_time[0])

    col1, col2= st.columns(2)
```

New Grading Scale

April 8

	Points
Problem 1	10
Problem 2	10
Problem 3	15
Problem 4	15
Problem 5	5
Problem 6	5
Clean, orderly & Informative display	10
Only need to edit paths to test data in one location	5
No hard-coded dates in source	5
Runs on test data with only modifying path to data sets	20