Overview

In this project, we aimed to explore the connection between COVID-19 cases (Rt values) and Personal Protective Equipment (PPE) supplies and find a way to better supply vulnerable hospitals. We ran k-means clustering on our datasets related to PPE supply, COVID-19 infection rates, mask usage, and health care workers in hospitals. We found that PPE availability and usage does suppress new COVID-19 cases (TAPs), and that hospitals near the lower Central Valley and Lake Tahoe are in the most need of PPE right now.

Names

- Michael Kusnadi
- Darren Yau
- Tali Edgar
- Sharon Edgar
- Delaney Leach

Note: If you monetize this work, please contact group members.

Research Question

To begin, we considered how the availability and usage of PPE supplies (gloves, surgical masks, N-95 masks) in California help prevent the spread of COVID. As we worked through our data, our question and the scope of our project evolved. Our real research question became this:

"**Did the availability and usage of PPE in California help prevent the spread of COVID? If so, how can we optimize the distribution of medical supplies such that the most vulnerable hospitals are prioritized?"**

Background & Prior Work

The COVID-19 pandemic has changed society as we know it. From lockdowns to interactions at the grocery store, this pandemic has taught us the importance of keeping ourselves and others around us safe. Since March 2020, the World Health Organization has advised the public to wear gloves, surgical masks, and N-95 masks in hopes of preventing the spread of COVID-19. [1] Research conducted by the University of California, San Francisco have helped support these claims.

For example, an experiment using high-speed video found that hundreds of droplets ranging from 20 to 500 micrometers were generated when saying a simple phrase, but that nearly all these droplets were blocked when the mouth was covered by a damp washcloth. [2] However, recent

studies only considered the efficacy of masks on an individual basis, leaving us wondering if there were any larger-scale trends of utilizing PPE. If we can show this point through our own research, perhaps we can reach a segment of people who doubt the benefit of PPE.

Regardless of how effective PPE is in mitigating COVID-19 exposure, none of it matters unless people can get their hands on some. According to the Department of Health and Human Services, the world has found itself in a major PPE shortage such that even frontline workers do not have consistent access to fresh medical equipment. [3] Although we cannot directly solve the problem of PPE shortage, we can attempt to optimize the distribution of remaining PPE, without bias, minimizing inefficienceies, and ultimately limiting the spread of the virus.

Prior work on this topic included a study from the Jama Internal Medicine Journal, which demonstrated that the frequency of SARS-CoV-2 antibody positivity among hospital employees required to use PPE was no higher than that of the general population in that area, suggesting that surgical masks and N95 alternatives continue to keep clinicians and health care workers safe. [4]

We were also inspired by the work of Peter Liu and Victoria Su Yi, who developed a COVID-19 simulator that used **Rt** values to measure the spread of the disease in an area. [5] **Rt** is a “measure of how fast the virus is growing.” If Rt is above the threshold (1.0), the virus spreads. If it is below 1.0, the virus stops spreading. We wanted to expand on their work by by comparing **Rt** to PPE availability and usage. We expect that the more PPE availability there is in a state, the lower its **Rt** value will be. We hope to be able to understand the effect that medical supplies have to the growth of a virus in a certain area.

References:

1. World Health Organization Coronavirus Guidelines (https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19-masks)
2. Science Behind How Face Masks Prevent Coronavirus (https://www.ucsf.edu/news/2020/06/417906/still-confused-about-masks-heres-science-behind-how-face-masks-prevent)
3. New Document Shows Inadequate Distribution of Personal Protective Equipment and Critical Medical Supplies to States (https://oversight.house.gov/news/press-releases/new-document-shows-inadequate-distribution-of-personal-protective-equipment-and)
4. Filtration Efficiency, Effectiveness, and Availability of N95 Face Masks for COVID-19 Prevention (https://jamanetwork.com/journals/jamainternalmedicine/fullarticle/2769441)
5. COVID-19 Simulator (https://towardsdatascience.com/covid-19-simulator-da5be0067193)

Hypothesis

We hypothesize that greater availability and utilization of PPE will lead to lower Rt values. Furthermore, we believe that the best way to distribute PPE is to prioritize based on need and risk, rather than just on location.

Dataset(s)

- **Dataset:** CA COVID-19 PPE Logistics

- **Dataset:** CA COVID-19 PPE Logistics
  - **Link:** https://data.ca.gov/dataset/da1978f2-068c-472f-be2d-04cdec48c3d9/resource/7d2f11a4-cc0f-4189-8ba4-8bee05493af1/download/logistics_ppe.csv (https://data.ca.gov/dataset/da1978f2-068c-472f-be2d-04cdec48c3d9/resource/7d2f11a4-cc0f-4189-8ba4-8bee05493af1/download/logistics_ppe.csv)
  - **Details:** This dataset contains 5.5 million observations of each CA county's cumulative PPE supply (by product) and date of observation. This dataset has been updated daily by the California Open Data Portal since 6-8-2020.
- **Dataset:** Rt Values
  - **Link:** https://d14wlfuexuxgcm.cloudfront.net/covid/rt.csv (https://d14wlfuexuxgcm.cloudfront.net/covid/rt.csv)
  - **Details:** This dataset contains 293 observations of each USA state's name, Rt value, cumulative and new COVID-19 infections, COVID-19 cases, tests, deaths, and date of observation. This dataset has been updated daily by the COVID-19 Tracking Project since 3-2-2020.
- **Dataset:** Definitive Healthcare USA Hospital Beds
  - **Link:** https://raw.githubusercontent.com/COGS108/group046_fa20/master/definitive-healthcare-usa-hospital-beds.csv?token=AG5SAPEWSSLHZIYH2A2NWP273DQ2C (https://raw.githubusercontent.com/COGS108/group046_fa20/master/definitive-healthcare-usa-hospital-beds.csv?token=AG5SAPEWSSLHZIYH2A2NWP273DQ2C)
  - **Details:** This dataset contains 456 observations of each USA hospital's name, location, county FIPS, number of licensed staff, number of ICU beds, and ICU bed utilization. This dataset was last updated in August 2020.
- **Dataset:** Hospitals
  - **Link:** https://opendata.arcgis.com/datasets/6ac5e325468c4cb9b905f1728d6fbf0f_0.zip?outSR=%7B%22latestWkid%22%3A3857%2C%22wkid%22%3A102100%7D (https://opendata.arcgis.com/datasets/6ac5e325468c4cb9b905f1728d6fbf0f_0.zip?outSR=%7B%22latestWkid%22%3A3857%2C%22wkid%22%3A102100%7D)
  - **Details:** This dataset contains 7,611 observations of each USA hospital's name, location, county FIPS, and coordinates. This dataset is updated weekly by the Department of Homeland Security.
- **Dataset:** COVID-19 Reported Patient Impact and Hospital Capacity by Facility
  - **Link:** https://healthdata.gov/sites/default/files/reported_hospital_capacity_admissions_facility-level_weekly_average_timeseries_20201207.csv (https://healthdata.gov/sites/default/files/reported_hospital_capacity_admissions_facility-level_weekly_average_timeseries_20201207.csv)
  - **Details:** This dataset contains 87,369 observations of each USA hospital's name, location, and number of confirmed and suspected COVID-19 cases, among other things. This dataset is updated weekly by the Department of Health and Human Services.
- **Dataset:** Mask-Wearing Survey Data
  - **Link:** https://raw.githubusercontent.com/nytimes/covid-19-data/master/mask-use/mask-use-by-county.csv (https://raw.githubusercontent.com/nytimes/covid-19-data/master/mask-use/mask-use-by-county.csv)
  - **Details:** This dataset contains 3,142 observations of each USA county's name, FIPS, and percentage of respondents who replied ALWAYS, FREQUENTLY, SOMETIMES, RARELY, and NEVER to mask usage. This dataset was last updated by the New York Times on 7-14-2020.
- **Dataset:** USA Counties

- **Link:** https://opendata.arcgis.com/datasets/48f9af87daa241c4b267c5931ad3b226_0.zip (https://opendata.arcgis.com/datasets/48f9af87daa241c4b267c5931ad3b226_0.zip)
- **Details:** This dataset contains 3,220 observations of each USA county's name, FIPS, and geometric shapefile, among other things. This dataset was last updated by ArcGIS in October 2020.

Data Analysis and Results

Setup

In [1]:
```
%config IPCompleter.greedy=True

# install packages
!pip install geopandas &> /dev/null
!pip install descartes &> /dev/null

# dataframe and plotting
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import gridspec
import matplotlib.lines as mlines
import matplotlib.dates as mdates
import datetime

# geolocation
import geopandas as gpd
from geopandas import GeoDataFrame
from shapely.geometry import Point
from mpl_toolkits.axes_grid1 import make_axes_locatable

# zip files
from urllib.request import urlopen
from zipfile import ZipFile
from io import BytesIO

# analysis
import scipy.stats as stats
from sklearn.cluster import KMeans
```

Exploring the PPE Dataset

First of all, we downloaded the 'COVID-19 PPE Logistics' dataset from the California Data Open Portal. This dataset contained information such as the county requesting PPE, the products requested, and the quantity filled.

To begin our analysis, we needed to clean the dataset because it contained a lot of unnecessary information regarding State Agencies, Non-Governmental Entities, etc. For our purposes, we only analyzed the 58 official counties in California.

In [2]:
```
URL = 'https://data.ca.gov/dataset/da1978f2-068c-472f-be2d-04cdec48c3d9/res
ppe_df = pd.read_csv(URL)
```

In [3]:
```
# drop null counties
ppe_df = ppe_df[ppe_df['county'].notna()]

# remove rows without real counties
to_drop = ['Unassigned', 'Non-Governmental Entity', 'State Agency', 'State
ppe_df = ppe_df[~ppe_df['county'].isin(to_drop)]

# remove rows with Nan in quantity_filled (assume 0)
ppe_df = ppe_df[ppe_df['quantity_filled'].notna()]

# format date from str to datetime to avoid complications later
ppe_df['as_of_date'] = pd.to_datetime(ppe_df['as_of_date'])
ppe_df.head()
```

Out[3]:

	county	product_family	quantity_filled	shipping_zip_postal_code	as_of_date
19	Los Angeles	Personnel	0.0	91010	2020-06-25
20	Los Angeles	Personnel	0.0	91010	2020-06-25
21	Orange	Viral Testing Media	2000.0	92705	2020-06-25
22	Orange	Swabs	1000.0	92705	2020-06-25
23	Butte	Surgical or Examination Gowns	1320.0	95965	2020-06-25

After cleaning the PPE dataset, we wanted to see which products were most requested in California. We reasoned that we should only focus on the most requested items since they would be significant enough to work with, and would reduce the possibility of wild variability.

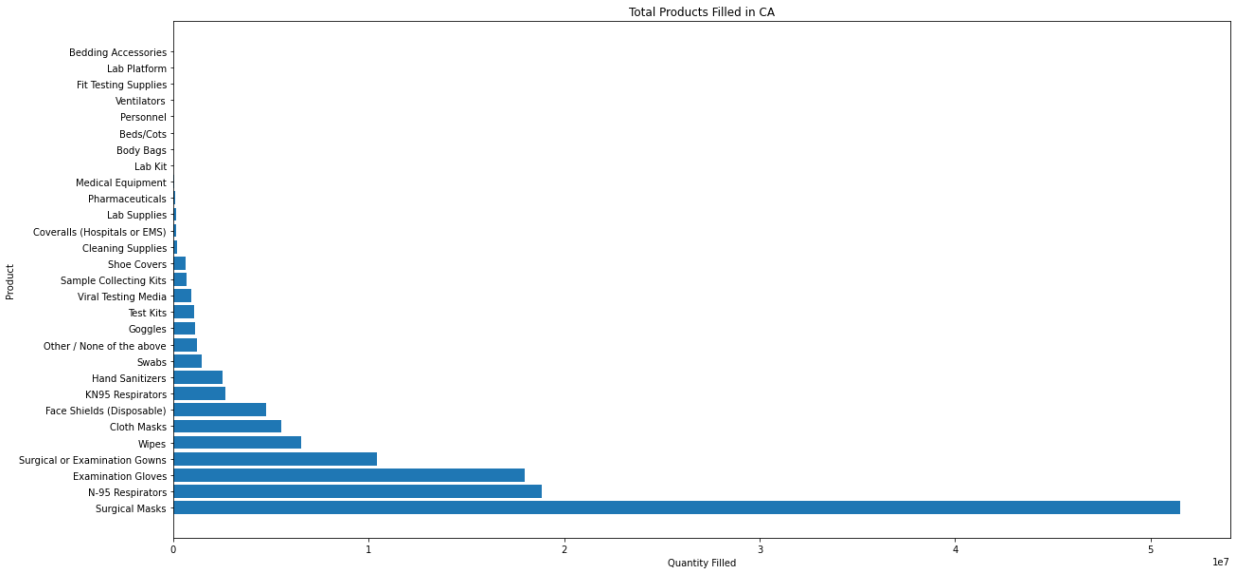
For example, if Yolo County received 0 ventilators while Los Angeles received 10, it would be difficult to model their relative PPE needs solely on that metric.

In [4]:
```
def table_products(county):
    ''' Returns a DataFrame of products and respective quantities given a s

    df_products = ppe_df[ppe_df['county'] == county]
    # remember 'quantity_filled' is cumulative, it represents the TOTAL num
    df_products = df_products.groupby(by=['product_family']).agg({'quantity
    df_products.sort_values(by=[('quantity_filled', 'max')], ascending=Fals
    df_products.reset_index(inplace=True)
    return df_products

def plot_products(county):
    ''' Plots a barplot of products and respective quantities given a speci
    temp_df = table_products(county)

    fig,ax = plt.subplots(1, 1, figsize=(20,10))
    ax.barh(temp_df['product_family'], temp_df['quantity_filled']['max'])
    ax.set_xlabel('Quantity Filled')
    ax.set_ylabel('Product')
    ax.set_title('Total Products Filled in ' + str(county))
```

In [5]:
```
# create dict of products and respective quantities (product: quantity)
product_dict = {}

# initialize dict
for product in ppe_df['product_family'].unique():
    product_dict[product] = 0.0

# create table of products for each county, then add each product to runnin
for county in ppe_df['county'].unique():
    temp_table = table_products(county)
    for product in list(product_dict.keys()):
        if (temp_table['product_family'] == product).any():
            product_dict[product] += float(temp_table.loc[temp_table['produ
                                                          product]['quantity_

cum_products_df = pd.DataFrame.from_dict(product_dict, orient='index').rese
cum_products_df.columns = ['product', 'quantity_filled']
cum_products_df.sort_values(by=['quantity_filled'], ascending=False, inplac
cum_products_df.reset_index(drop=True, inplace=True)
```

In [6]:
```
# plotting all products requested in CA
fig,ax = plt.subplots(1, 1, figsize=(20,10))
ax.barh(cum_products_df['product'], cum_products_df['quantity_filled'])
ax.set_xlabel('Quantity Filled')
ax.set_ylabel('Product')
ax.set_title('Total Products Filled in CA')
```

Out[6]:
```
Text(0.5, 1.0, 'Total Products Filled in CA')
```



We found that California needed Surgical Masks, N-95 Respirators, and Examination Gloves the most. This was consistent with our expectations so we continued our analysis on these 3 medical supplies, dropping the rest.

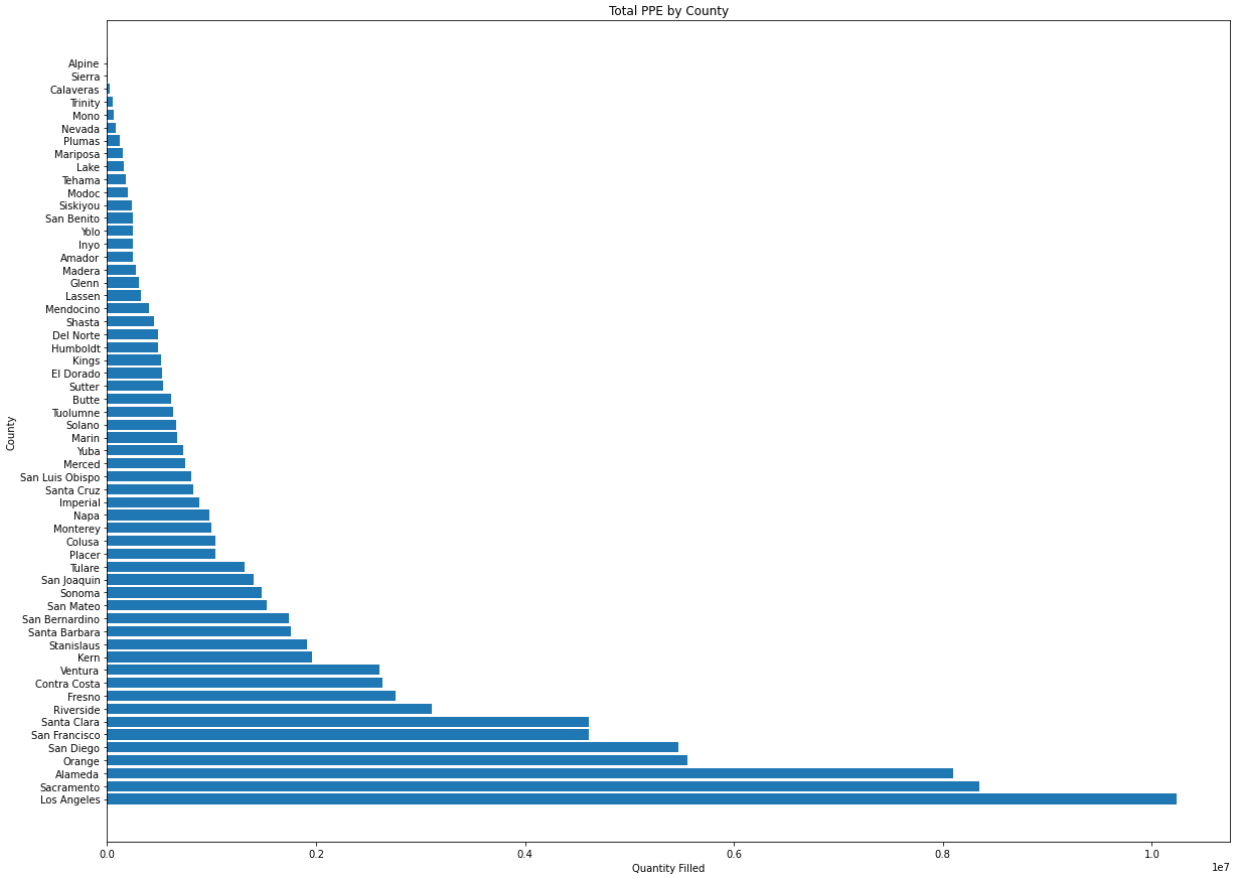
Next, we wanted to visualize the PPE needs of each county. This would be helpful in determining which counties were woefully under-supplied after normalizing for population. Since PPE quantity was reported in the form of cumulative sum over time, we extrapolated only the most recently reported data for each county. This approach was appropriate for our time-series dataset.

In [7]:
```
# only select most common products
ppe_df = ppe_df[(ppe_df['product_family'] == 'Surgical Masks') |
        (ppe_df['product_family'] == 'N-95 Respirators') |
        (ppe_df['product_family'] == 'Examination Gloves')]
```

In [8]:
```
def sum_products(county):
    ''' Returns the cumulative PPE given a specified county '''

    temp_df = table_products(county)
    return temp_df['quantity_filled']['max'].sum()

# create new DataFrame of counties and their total PPE filled
cumulative_list = []

# calculate the cumulative PPE filled for every county
for county in ppe_df['county'].unique():
    tot = sum_products(county)
    cumulative_list.append(tot)

ppe_county_df = pd.DataFrame({'county': ppe_df['county'].unique(), 'quantit
ppe_county_df.sort_values(by=['quantity_filled'], ascending=False, inplace=
ppe_county_df.reset_index(drop=True, inplace=True)
ppe_county_df.columns = ['county', 'cum_ppe']
```

In [9]:
```
fig,ax = plt.subplots(1, 1, figsize=(20,15))
ax.barh(ppe_county_df['county'], ppe_county_df['cum_ppe'])
ax.set_xlabel('Quantity Filled')
ax.set_ylabel('County')
ax.set_title('Total PPE by County')
#ax.set_xscale('log')
```

Out[9]:
```
Text(0.5, 1.0, 'Total PPE by County')
```



Los Angeles, Sacramento, and Alameda counties received the most PPE in California. Again, this was consistent with our expectation of higher-population areas receiving more PPE. Once we had a better feel for our data, we attempted to uncover any correlations between PPE and relevant COVID data.

Correlation to Effective Reproduction Rate (Rt)

We knew that PPE was imperative in preventing the spread of COVID-19. However we wanted to actually quantify this into a correlational study. More specifically, we wanted to ask: **"Does greater utilization and availability of PPE lead to lower COVID-19 infection rates?"** There is no better use of data right now than to convince people to wear masks.

To answer this, we utilized the effective reproduction rate (Rt) of COVID-19. Rt estimates how many secondary infections are likely to occur from a single infection in a specific area. Values over 1.0 mean we should expect more cases in that area, values under 1.0 mean we should expect fewer. It is the perfect tool for quantifying COVID-19 infection rates.

We wrangled our Rt values from rt.live, which in turn calculated its statistics from The COVID-19 Tracking Project. Since PPE data in California was not recorded until 6-8-2020, we removed all Rt data before then and merged the 2 datasets. Then, we plotted California's PPE supply against the Rt to see if there were any obvious correlations.

In [10]:
```
rt_df = pd.read_csv('https://d14wlfuexuxgcm.cloudfront.net/covid/rt.csv')

# convert dates to datetime to avoid complications
rt_df['date'] = pd.to_datetime(rt_df['date'])

# only get data on CA
rt_df = rt_df[rt_df['region'] == 'CA']
rt_df.head()
```

Out[10]:

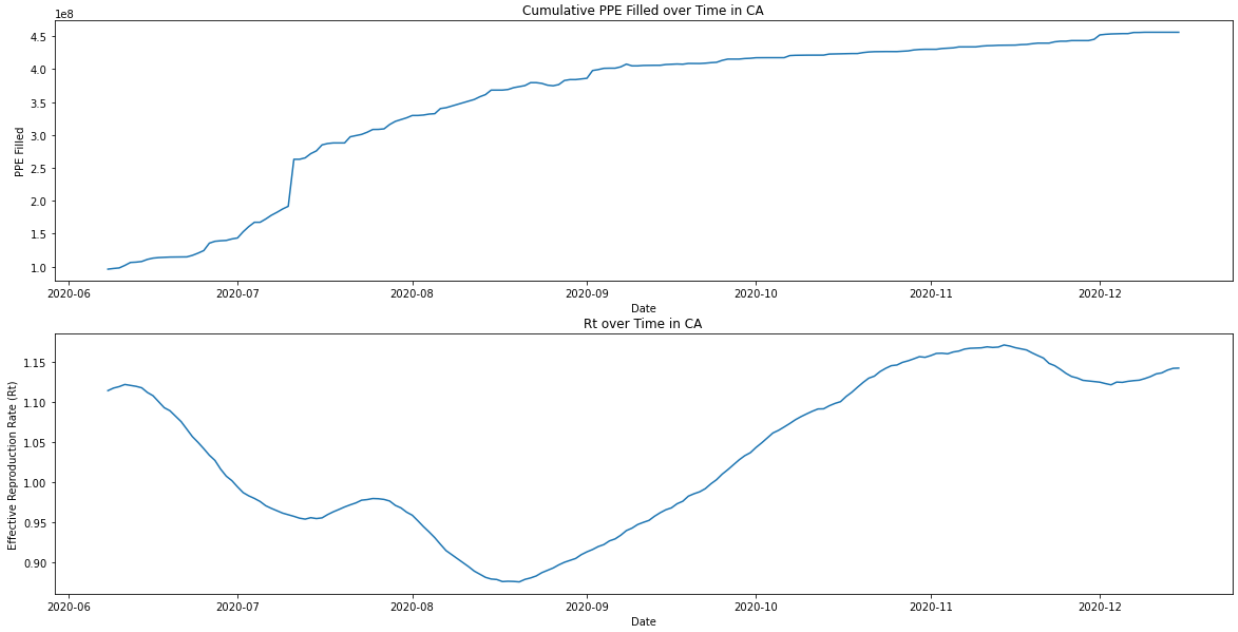
	date	region	index	mean	median	lower_80	upper_80	infections	test_adjusted_pos
14043	2020-02-25	CA	0	2.545476	2.536971	2.184822	2.845932	159.762741	
14044	2020-02-26	CA	1	2.543654	2.534598	2.219595	2.867117	2.933421	
14045	2020-02-27	CA	2	2.540383	2.532476	2.221514	2.819162	41.613516	
14046	2020-02-28	CA	3	2.534329	2.535106	2.230906	2.794974	79.621975	
14047	2020-02-29	CA	4	2.523853	2.522542	2.258555	2.787576	91.372692	

In [11]:
```
# copy PPE
temp_df = ppe_df.copy()
temp_df = temp_df.groupby(by=['as_of_date']).sum()
temp_df.reset_index(inplace=True)
temp_df.columns = ['date', 'quantity_filled']

# discard all Rt data outside range of PPE (warning!)
rt_df = rt_df[rt_df['date'] >= np.datetime64('2020-06-08')]

# substitute zeroes so log operation can work
rt_df.loc[(rt_df['new_cases'] == 0), 'new_cases'] = 1

# merge PPE and Rt on date
rt_ppe_df = rt_df.merge(temp_df, how='inner')
```

In [12]:
```
def time_plot(ax, county='CA', product='PPE'):

    ''' Plot time chart of PPE filled by product '''
    temp_df = ppe_df[ppe_df['county'] == county] if county != 'CA' else ppe
    temp_df = temp_df[ppe_df['product_family'] == product] if product != 'P
    temp_df = temp_df.groupby(by=['as_of_date']).sum()
    temp_df.reset_index(inplace=True)

    ax.plot(temp_df['as_of_date'], temp_df['quantity_filled'])
    ax.set_xlabel('Date')
    ax.set_ylabel(str(product) + ' Filled')
    ax.set_title('Cumulative ' + str(product) + ' Filled over Time in ' + s
```

In [13]:
```
fig,ax = plt.subplots(2, 1, figsize=(20,10))

time_plot(ax[0])

ax[1].plot(rt_ppe_df['date'], rt_ppe_df['mean'])
ax[1].set_xlabel('Date')
ax[1].set_ylabel('Effective Reproduction Rate (Rt)')
ax[1].set_title('Rt over Time in CA')
```

Out[13]:
```
Text(0.5, 1.0, 'Rt over Time in CA')
```



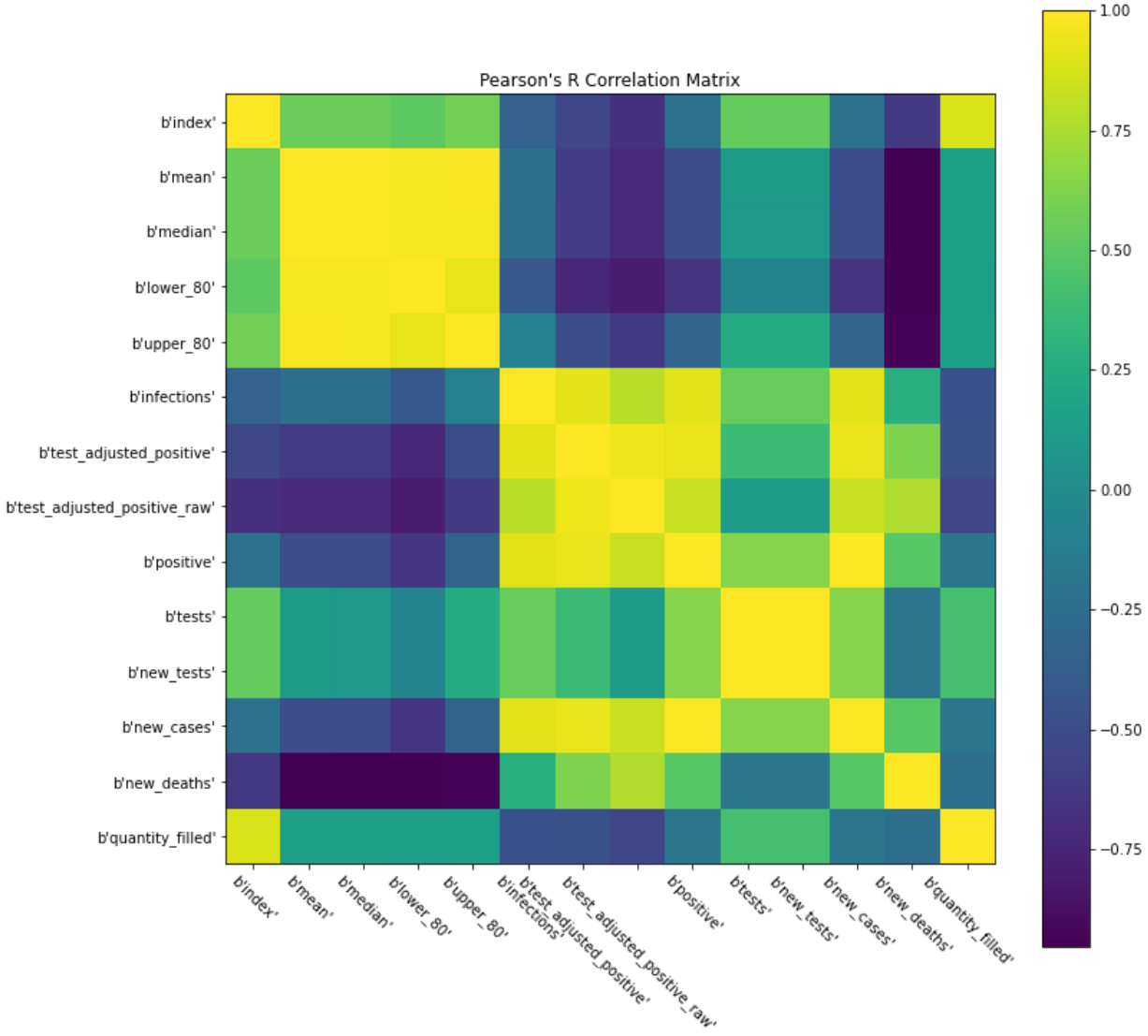
In [14]:
```
rt_ppe_df['quantity_filled'].corr(rt_ppe_df['mean'])
```

Out[14]:
```
0.2087107466230793
```

Although there was a slight Pearson correlation between PPE and Rt, we decided that it wasn't statistically significant enough to consider. However, this correlation was not zero, which prompted us to further explore our data for any potential breakthroughs. Below is the pairwise correlation matrix for our PPE and Rt datasets.

In [15]:
```
# visualize correlation matrix
corr_matrix = rt_ppe_df.corr()

col_labels = [col.encode('utf8') for col in corr_matrix.corr().columns]
plt.figure(figsize=(12,12))
plt.imshow(corr_matrix.corr(),interpolation='none')
plt.xticks(range(len(col_labels)), col_labels, rotation=-45)
plt.yticks(range(len(col_labels)), col_labels)
plt.colorbar()

plt.title("Pearson's R Correlation Matrix")
```

Out[15]:
```
Text(0.5, 1.0, "Pearson's R Correlation Matrix")
```



In [16]:
```
rt_ppe_df['quantity_filled'].corr(np.log(rt_ppe_df['test_adjusted_positive'
```

Out[16]:
```
-0.2370994124210173
```

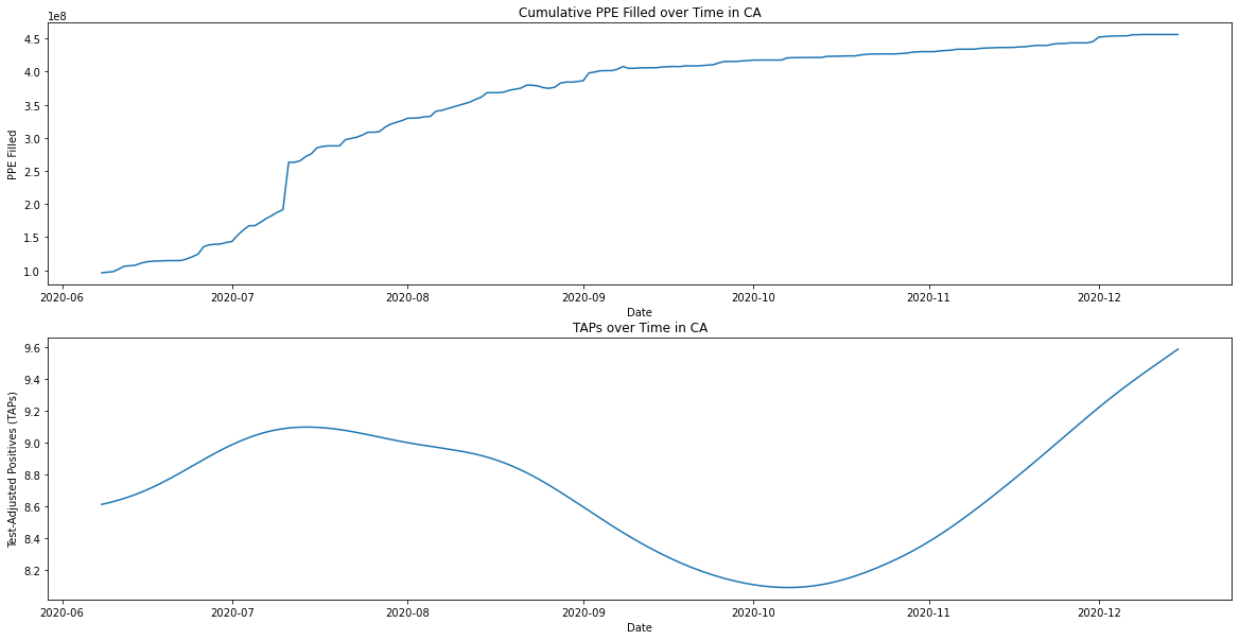
We found that there was a slightly stronger correlation between PPE Filled and Test-adjusted

positives (TAPs). TAPs are a relative measure of how many true positives there are, corrected for false positives and testing rates. We felt that this was a more promising metric for the effectiveness of PPE on new infections since TAPs only reflects new cases, whereas Rt attempts to curve-fit old cases.

Our Pearson coefficient increased further when we took the log of TAPs. It made sense to apply a log transformation because TAPs is essentially raw population data.

In [17]:
```
fig,ax = plt.subplots(2, 1, figsize=(20,10))

time_plot(ax[0])

ax[1].plot(rt_ppe_df['date'], np.log(rt_ppe_df['test_adjusted_positive']))
ax[1].set_xlabel('Date')
ax[1].set_ylabel('Test-Adjusted Positives (TAPs)')
ax[1].set_title('TAPs over Time in CA')
```

Out[17]:
```
Text(0.5, 1.0, 'TAPs over Time in CA')
```

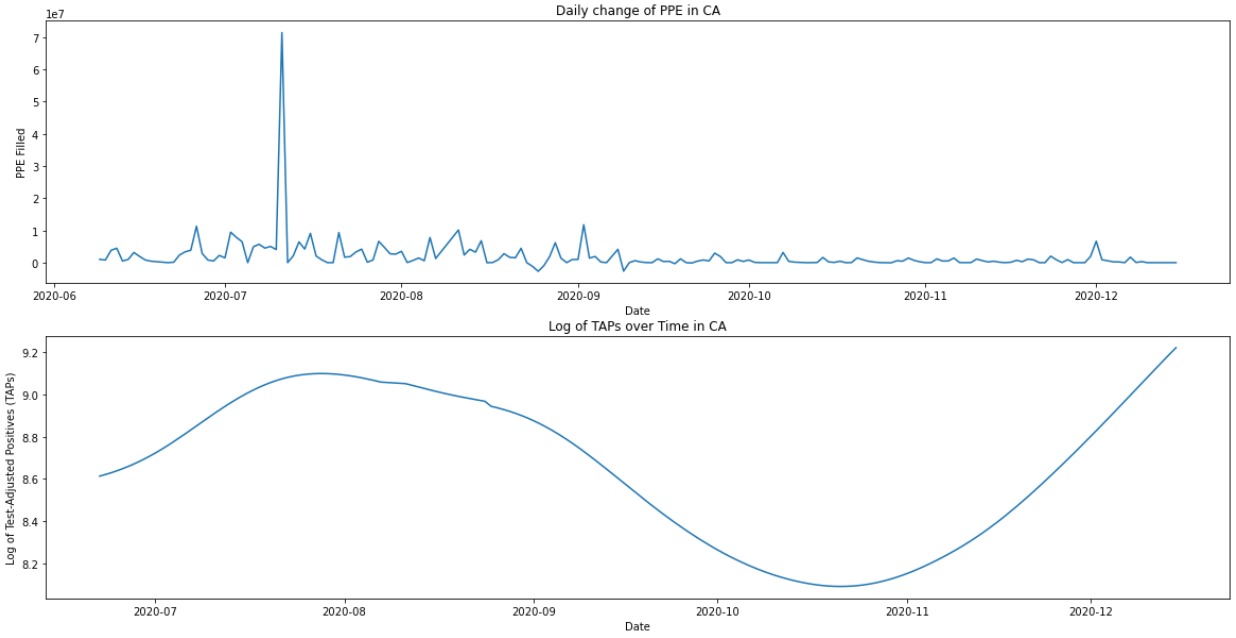


We noticed that around mid-July, a large shipment of PPE seemed to significantly reduce the Test-adjusted positives in California. Instead of plotting the cumulative PPE, we decided to plot the **daily PPE change** to help us see the effect of new PPE shipments on TAPs. We also time-shifted our TAPs data 14-days later to account for any time-delays between the actual contraction of COVID-19 and its subsequent reporting. Time delays include:

1. Incubation period (2-14 days)
2. Delays in booking a doctor's appointment
3. Time for doctor to confirm a positive result

In [18]:
```
# PPE data
#temp_df = temp_df.copy()
temp_df['change'] = temp_df['quantity_filled']
temp_df['change'] = temp_df['change'].shift(1)

# remove first value (NaN)
temp_df = temp_df.iloc[1:]

# calculate daily change in PPE
temp_df['change'] = temp_df['quantity_filled'] - temp_df['change']

# shift TAP by 14-days to account for COVID incubation period
rt_ppe_df['test_adjusted_positive'] = rt_ppe_df['test_adjusted_positive'].s



fig,ax = plt.subplots(2, 1, figsize=(20,10))

ax[0].plot(temp_df['date'], temp_df['change'])
ax[0].set_xlabel('Date')
ax[0].set_ylabel('PPE Filled')
ax[0].set_title('Daily change of PPE in CA')

ax[1].plot(rt_ppe_df['date'], np.log(rt_ppe_df['test_adjusted_positive']))
ax[1].set_xlabel('Date')
ax[1].set_ylabel('Log of Test-Adjusted Positives (TAPs)')
ax[1].set_title('Log of TAPs over Time in CA')
```

Out[18]: Text(0.5, 1.0, 'Log of TAPs over Time in CA')



We discovered that there was indeed a relationship between daily PPE change and TAPs.

We discovered that there was indeed a relationship between daily PPE change and TAPs. **Specifically, the inflection points in TAPs occurred when larger-than-normal shipments of PPE were received.** Conversely, TAPs began to increase when few PPE were received.

Unfortunately, the downside to this approach was that we could no longer do a correlational analysis between the 2 variables. This is because we were originally testing whether total/cumulative PPE supply had an effect on TAPs. If we were to change this to the daily PPE change, we would be implying that all PPE get used up the day they are shipped. This would be a false assumption. The effects of large PPE shipments can extend far into the future as hospitals learn how to ration their new supplies. This is evident in the plot, where the large shipment in mid-July helped suppress TAPs for months.

A New Question

We confirmed a weak relationship between PPE supply and the spread of COVID-19, but we felt that this only lead to more questions. In particular, we wondered: **"How can we optimize the location of distribution centers such that vulnerable hospitals are prioritized?"** If we could get time-critical PPE to the places that mattered, potentially lessen the impact of COVID on our communities.

First, we had to to define "vulnerability" in quantifiable manner. We came up with 3 variables that we believed were most relevant:

1. **PPE Supply:** Hospitals with scarce PPE supplies are at a greater risk of spreading COVID-19 than well-equipped ones.
2. **Operating Capacity:** Hospitals operating at full capacity will burn through their PPE quicker than those with low activities.
3. **Mask Usage:** Hospitals located in areas that report lower mask usage will see an uptick in COVID-19 cases.

Variable 1: PPE Supply

We decided to define a hospital's PPE supply as the **ratio of PPE to staff.** This makes sense because we need to normalize PPE supply against the number of staff working in a hospital. If a hospital has a large PPE supply but an even larger employee base, it may not necessarily be better off than other hospitals with less PPE and a smaller employee base.

We already had data on PPE supply. To address the other half, we downloaded the "ESRI Definitive Healthcare" dataset from Kaggle, which contained the number of licensed staff at each hospital. We cleaned it by removing/replacing all invalid entries and only selecting the hospital names and number of staff.

In [19]:
```
hospitals_esri = pd.read_csv('local_data/definitive-healthcare-usa-hospital
hospitals_esri.head()
```

Out[19]:

	geometry	objectid	hospital_n	hospital_t	hq_address	hq_addre_1	hq_city	hq_state	h
0	POINT (-112.0661569 33.4954978)	1	Phoenix VA Health Care System (AKA Carl T Hayd...	VA Hospital	650 E Indian School Rd	NaN	Phoenix	AZ	
1	POINT (-110.9658852 32.1812634)	2	Southern Arizona VA Health Care System	VA Hospital	3601 S 6th Ave	NaN	Tucson	AZ	
2	POINT (-119.7797421 36.7733235)	3	VA Central California Health Care System	VA Hospital	2615 E Clinton Ave	NaN	Fresno	CA	
3	POINT (-72.9576103 41.2844004)	4	VA Connecticut Healthcare System - West Haven ...	VA Hospital	950 Campbell Ave	NaN	West Haven	CT	
4	POINT (-75.6065325 39.7402063)	5	Wilmington VA Medical Center	VA Hospital	1601 Kirkwood Hwy	NaN	Wilmington	DE	

In [20]:
```
def uppercase(name):
    return name.upper()

# convert hospital names to uppercase
hospitals_esri['hospital_n'] = hospitals_esri['hospital_n'].apply(uppercase

# only get data on CA
hospitals_esri = hospitals_esri[hospitals_esri['hq_state'] == 'CA']

# only get relevant columns
hospitals_esri = hospitals_esri[['hospital_n', 'num_licens']]

# drop hospitals with no staff reported
hospitals_esri.drop(hospitals_esri[hospitals_esri['num_licens'] == "****"].
hospitals_esri['num_licens'] = hospitals_esri['num_licens'].astype('int32')
hospitals_esri.head()
```

Out[20]:

	hospital_n	num_licens
263	ST ROSE HOSPITAL	195
264	ALTA BATES SUMMIT MEDICAL CENTER - SUMMIT CAMPUS	403
265	SAN LEANDRO HOSPITAL	93
266	WASHINGTON HOSPITAL	341
267	ALAMEDA HOSPITAL	100

After cleaning the dataset, we were still missing the absolute locations of each hospital. We decided to merge the dataset with a shapefile so each hospital's coordinates would be embedded into the larger GeoDataFrame.

In [21]:
```
# data on hospital latitude/longitude
URL = 'https://opendata.arcgis.com/datasets/6ac5e325468c4cb9b905f1728d6fbf0
hospitals_shp = gpd.read_file(URL)

# only get CA hospitals
hospitals_shp = hospitals_shp[hospitals_shp['STATE'] == 'CA']

# only get name, num of staff, latitude, and longitude
hospitals_shp = hospitals_shp[['NAME', 'COUNTY', 'COUNTYFIPS', 'LATITUDE',
```

In [22]:
```
# merge hospital data with geolocations
hospitals_staff_df = pd.merge(left=hospitals_shp, right=hospitals_esri, lef
hospitals_staff_df = hospitals_staff_df[['NAME', 'COUNTY', 'COUNTYFIPS', 'n
hospitals_staff_df.columns = ['hospital_n', 'county', 'fips', 'num_licens',
hospitals_staff_df.head()
```

Out[22]:

	hospital_n	county	fips	num_licens	latitude	longitude
0	EAST LOS ANGELES DOCTORS HOSPITAL	LOS ANGELES	06037	102	34.023647	-118.184165
1	LAKEWOOD REGIONAL MEDICAL CENTER	LOS ANGELES	06037	172	33.859707	-118.148403
2	MONROVIA MEMORIAL HOSPITAL	LOS ANGELES	06037	49	34.148759	-117.992668
3	MONTEREY PARK HOSPITAL	LOS ANGELES	06037	101	34.049466	-118.138262
4	MODOC MEDICAL CENTER	MODOC	06049	16	41.480056	-120.545744

Lastly, we merged our previous PPE dataset into this GeoDataFrame, then calculate the PPE-to-staff statistic for each hospital. Before we incorporated this statistic, however, we also wanted to check if was normally distributed. We definitely did not want to deal with annoying outliers later in the analysis.

In [23]:
```
# convert county to uppercase
ppe_county_df['county'] = ppe_county_df['county'].apply(uppercase)

# merge PPE and hospital data, then calculate PPE-to-staff statistic
hospitals_staff_ppe_df = pd.merge(left=hospitals_staff_df, right=ppe_county
hospitals_staff_ppe_df['ppe_per_staff_ratio'] = hospitals_staff_ppe_df['cum
hospitals_staff_ppe_df = hospitals_staff_ppe_df[['hospital_n', 'county', 'f
                                                 'latitude', 'longitude']]

# see which hospitals have the least PPE per staff
hospitals_staff_ppe_df.sort_values(by='ppe_per_staff_ratio', ascending=True
```

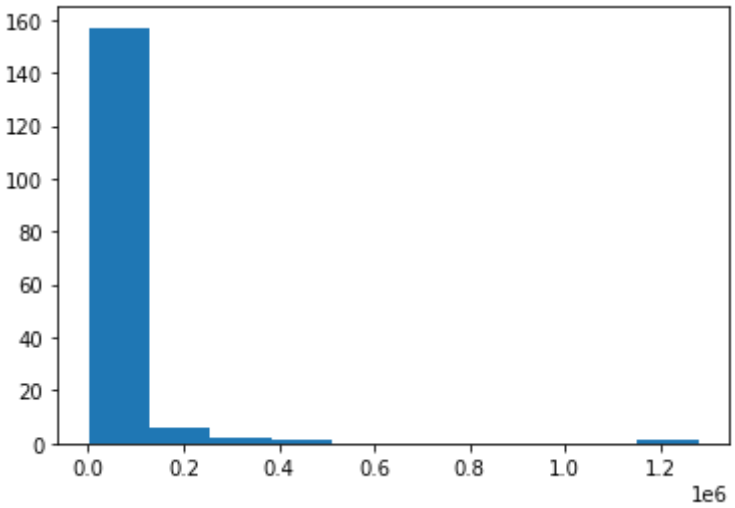
Out[23]:

	hospital_n	county	fips	num_licens	cum_ppe	ppe_per_staff_ratio	latitude	
64	SIERRA NEVADA MEMORIAL HOSPITAL	NEVADA	06057	104	85000.0	8.173077e+02	39.228016	-
112	SHASTA REGIONAL MEDICAL CENTER	SHASTA	06089	226	451600.0	1.998230e+03	40.585632	-
110	TRINITY HOSPITAL	TRINITY	06105	25	52312.0	2.092480e+03	40.738084	-
65	TAHOE FOREST HOSPITAL	NEVADA	06057	35	85000.0	2.428571e+03	39.325107	-
125	MADERA COMMUNITY HOSPITAL	MADERA	06039	106	279800.0	2.639623e+03	36.943812	-
...	...	...	...	...	...	...	...	
2	MONROVIA MEMORIAL HOSPITAL	LOS ANGELES	06037	49	10240000.0	2.089796e+05	34.148759	-
35	PACIFICA HOSPITAL OF THE VALLEY	LOS ANGELES	06037	38	10240000.0	2.694737e+05	34.240098	-
49	SHARP MCDONALD CENTER	SAN DIEGO	06073	16	5464320.0	3.415200e+05	32.806012	-
105	LAGUNA HONDA HOSPITAL AND REHABILITATION CENTER	SAN FRANCISCO	06075	11	4610000.0	4.190909e+05	37.774364	-
25	CATALINA ISLAND MEDICAL CENTER	LOS ANGELES	06037	8	10240000.0	1.280000e+06	33.339204	-

167 rows × 8 columns

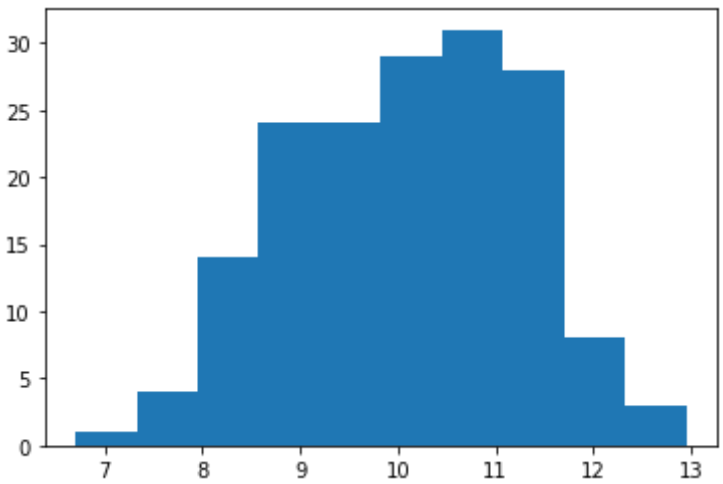
In [24]:
```
# see 'ppe_per_staff_ratio' distribution
plt.hist(hospitals_staff_ppe_df['ppe_per_staff_ratio'])
```

Out[24]:
```
(array([157.,   6.,   2.,   1.,   0.,   0.,   0.,   0.,   0.,   1.]),
 array([8.17307692e+02, 1.28735577e+05, 2.56653846e+05, 3.84572115e+05,
        5.12490385e+05, 6.40408654e+05, 7.68326923e+05, 8.96245192e+05,
        1.02416346e+06, 1.15208173e+06, 1.28000000e+06]),
 <a list of 10 Patch objects>)
```



In [25]:
```
# remove outlier and log
hospitals_staff_ppe_df.drop(hospitals_staff_ppe_df.index[25], inplace=True)
hospitals_staff_ppe_df['log_ppe_per_staff_ratio'] = np.log(hospitals_staff_
plt.hist(hospitals_staff_ppe_df['log_ppe_per_staff_ratio'])
```

Out[25]:
```
(array([ 1.,  4., 14., 24., 24., 29., 31., 28.,  8.,  3.]),
 array([ 6.70601564,  7.32999839,  7.95398114,  8.57796389,  9.20194664,
         9.82592939, 10.44991214, 11.07389489, 11.69787764, 12.32186039,
        12.94584314]),
 <a list of 10 Patch objects>)
```



In [26]:
```
# test for normality
stats.normaltest(hospitals_staff_ppe_df['log_ppe_per_staff_ratio'])
```

Out[26]:
```
NormaltestResult(statistic=3.854306479123892, pvalue=0.14556198933498607)
```

After we removed an obvious outlier and applied a logarithmic transformation, our PPE-to-staff statistic looked normally distributed. Our p-value was sufficiently large to confirm this (> 0.05).

Variable 2: Operating Capacity

We defined Operating Capacity very similarly to PPE Supply. Essentially, it is the **ratio of reported COVID-19 cases to staff.** We figured that more COVID-19 cases meant each employee had more people to attend to, thus increasing their workload and exposure to COVID.

This time, we already had the data on hospital staff. We chose the "Hospital Capacity by Facility" dataset from the Department of Health and Human Services for its reported COVID-19 data. Again, we cleaned the dataset and replaced all extraneous values with more reasonable ones. In the end, we only selected the hospital names and weekly totals for COVID-19 cases.

In [27]:
```
# data on hospital patients, staff, etc.
URL = 'https://healthdata.gov/sites/default/files/reported_hospital_capacit
hospitals_covid_df = pd.read_csv(URL, skipinitialspace=True)

# only get CA hospitals
hospitals_covid_df = hospitals_covid_df[hospitals_covid_df['state'] == 'CA'

# get latest report on hospitals from time series
hospitals_covid_df = hospitals_covid_df.groupby(by='hospital_name', as_inde

# only get name, tot_covid_confirmed
hospitals_covid_df = hospitals_covid_df[['hospital_name', 'total_adult_pati
hospitals_covid_df.columns = ['hospital_n', 'tot_covid_confirmed']

# values less than 4 were replaced with -999999.0, set to 4
hospitals_covid_df['tot_covid_confirmed'].replace({-999999.0: 4}, inplace=T
```

After that, we simply merged it into our larger GeoDataFrame and calculated the cases-to-staff ratio. We had to reclean our dataset because of duplicate entries, but this was easily taken care of. We also checked to make sure our statistic passed the normality test, just as a precaution.

In [28]:
```
# merge with hospitals_ppe_df
hospitals_staff_ppe_covid_df = pd.merge(left=hospitals_staff_ppe_df, right=
                                        left_on='hospital_n', right_on='hosp

# calculate covid-to-staff ratio
hospitals_staff_ppe_covid_df['covid_per_staff_ratio'] = hospitals_staff_ppe
                                                        hospitals_staff_ppe

# select relevant columns
hospitals_staff_ppe_covid_df = hospitals_staff_ppe_covid_df[['hospital_n',
                                                             'cum_ppe', 'ppe_per
                                                             'tot_covid_confirme
                                                             'longitude']]

# see which hospitals have the most covid per staff
hospitals_staff_ppe_covid_df.sort_values(by='covid_per_staff_ratio', ascend
```

Out[28]:

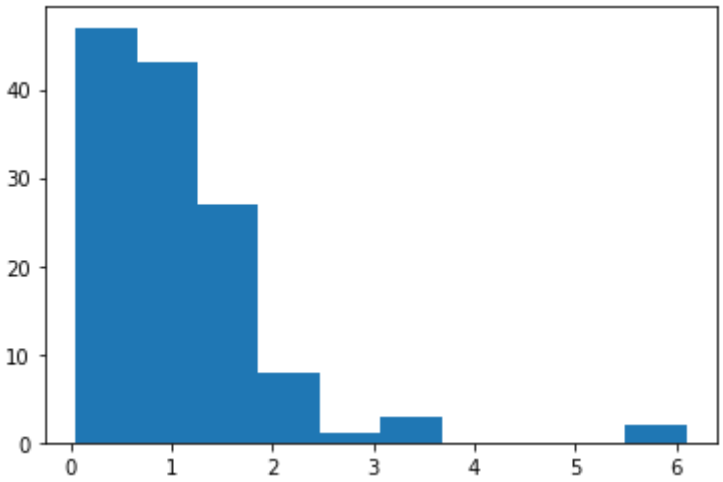
	hospital_n	county	fips	num_licens	cum_ppe	ppe_per_staff_ratio	log_ppe_per_st
74	BARSTOW COMMUNITY HOSPITAL	SAN BERNARDINO	06071	30	1735000.0	57833.333333	1(
77	DESERT VALLEY HOSPITAL	SAN BERNARDINO	06071	148	1735000.0	11722.972973	9
37	SHARP CHULA VISTA MEDICAL CENTER	SAN DIEGO	06073	243	5464320.0	22486.913580	1(
99	MADERA COMMUNITY HOSPITAL	MADERA	06039	106	279800.0	2639.622642	7
81	VICTOR VALLEY GLOBAL MEDICAL CENTER	SAN BERNARDINO	06071	101	1735000.0	17178.217822	9
...	...	...	...	...	...	...	
2	MONROVIA MEMORIAL HOSPITAL	LOS ANGELES	06037	49	10240000.0	208979.591837	12
41	KINDRED HOSPITAL - SAN FRANCISCO BAY AREA	ALAMEDA	06001	99	8100560.0	81823.838384	1(
111	PALO VERDE HOSPITAL	RIVERSIDE	06065	51	3109440.0	60969.411765	1(
89	GOLETA VALLEY COTTAGE HOSPITAL	SANTA BARBARA	06083	52	1757000.0	33788.461538	1(

	hospital_n	county	fips	num_licens	cum_ppe	ppe_per_staff_ratio	log_ppe_per_st
19	KINDRED HOSPITAL - LOS ANGELES	LOS ANGELES	06037	81	10240000.0	126419.753086	1

129 rows × 11 columns

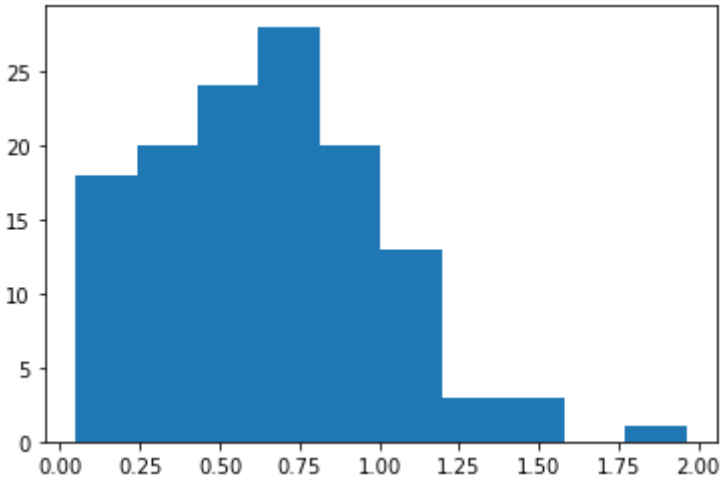
In [29]:
```
# see 'ppe_per_staff_ratio' distribution
plt.hist(hospitals_staff_ppe_covid_df['covid_per_staff_ratio'])
```

Out[29]:
```
(array([47., 43., 27.,  8.,  1.,  3.,  0.,  0.,  0.,  2.]),
 array([0.04938272, 0.65444444, 1.25950617, 1.8645679 , 2.46962963,
        3.07469136, 3.67975309, 4.28481481, 4.88987654, 5.49493827,
        6.1       ]),
 <a list of 10 Patch objects>)
```



In [30]:
```
# remove outlier and log
hospitals_staff_ppe_covid_df.drop(hospitals_staff_ppe_covid_df.index[75], i
# add one here to avoid negative values after logging
hospitals_staff_ppe_covid_df['log_covid_per_staff_ratio'] = np.log(1 + hosp
plt.hist(hospitals_staff_ppe_covid_df['log_covid_per_staff_ratio'])
```

Out[30]:
```
(array([18., 20., 24., 28., 20., 13.,  3.,  3.,  0.,  1.]),
 array([0.0482021 , 0.23939137, 0.43058064, 0.62176991, 0.81295917,
        1.00414844, 1.19533771, 1.38652698, 1.57771625, 1.76890552,
        1.96009478]),
 <a list of 10 Patch objects>)
```



In [31]:
```
# test for normality
stats.normaltest(hospitals_staff_ppe_covid_df['log_covid_per_staff_ratio'])
```

Out[31]:
```
NormaltestResult(statistic=6.618092403547842, pvalue=0.03655101943072068)
```

Our p-value didn't quite meet the standards for a normal distribution. Our statistic had a slight positive skew with a longer tail on the right side. However, we believe this distribution is perfectly acceptable since there are fewer hospitals per person in densely-populated cities. This means there are fewer hospitals in big cities, but they will all be working at full capacity, contributing to the overall skew observed in the graph.

Variable 3: Mask Usage

PPE Supply and Operating Capacity both express "hospital vulnerability" as a function of real-time conditions. We wanted to add an X-factor that also factored in potential future repurcussions.

We decided to include **each county's mask usage (or lack thereof)** as a variable in our model. This is useful because hospitals located in "anti-mask" zones may eventually find themselves treating more COVID-19 patients due to the public's negligence.

To model this, we needed a dataset that captured the public opinion on wearing masks. Luckily we found a survey of mask usage in the appropriately named "Mask Survey" dataset from the New York Times. The dataset contained 250,000 survey responses between July 2 and July 14, 2020. Participants were asked how often they wear a mask in public when expected to be within 6 feet of another person, with choices ranging from "Never" to "Always."

Once again, we cleaned our data and only selected the county FIPS and percentage of people who answered "Always."

In [32]:
```
URL = 'https://raw.githubusercontent.com/nytimes/covid-19-data/master/mask-
mask_df = pd.read_csv(URL)
```

In [33]:
```
mask_df.dtypes
```

Out[33]:
```
COUNTYFP        int64
NEVER         float64
RARELY        float64
SOMETIMES     float64
FREQUENTLY    float64
ALWAYS        float64
dtype: object
```

In [34]:
```
def normalize_countyfp(county):
    '''Append 0 and convert to string'''

    output = ''
    if int(county) < 10000:
        output += '0'
    return output + str(int(county))

# normalize county FIPS
mask_df['COUNTYFP'] = mask_df['COUNTYFP'].apply(normalize_countyfp)
```

We also downloaded the shapefile of California counties so we could merge it with our mask data and obtain a better geographic sense of our data.

In [35]:
```
# get county shapefile
#URL = 'https://opendata.arcgis.com/datasets/48f9af87daa241c4b267c5931ad3b2
#z = urlopen(URL)
#myzip = ZipFile(BytesIO(z.read())).extractall()
county_shp = gpd.read_file('local_data/USA_Counties/USA_Counties.shp')
```

In [36]:
```
county_shp.head()
```

Out[36]:

	FID	OBJECTID	NAME	STATE_NAME	STATE_FIPS	CNTY_FIPS	FIPS	POPULATION	POP_SQ
0	1	1	Kauai	Hawaii	15	007	15007	73169	11
1	2	2	Honolulu	Hawaii	15	003	15003	1014211	168
2	3	3	Hawaii	Hawaii	15	001	15001	204027	5
3	4	4	Kalawao	Hawaii	15	005	15005	91	
4	5	5	Maui	Hawaii	15	009	15009	169713	14

5 rows × 59 columns

In [37]:
```
# only get counties in CA
county_shp = county_shp[county_shp['STATE_NAME'] == 'California']

# remove unnecessary columns
county_shp = county_shp[['NAME', 'FIPS', 'POPULATION', 'geometry']]

# merge mask and county data
mask_county_df = pd.merge(left=county_shp, right=mask_df, left_on='FIPS', r
mask_county_df = mask_county_df[['NAME', 'FIPS', 'POPULATION', 'ALWAYS', 'g
mask_county_df.head()
```

Out[37]:

	NAME	FIPS	POPULATION	ALWAYS	geometry
0	Monterey	06053	431696	0.763	MULTIPOLYGON (((-121.45301 35.87489, -121.4536...
1	Santa Cruz	06087	278575	0.797	POLYGON ((-122.22140 37.21497, -122.22135 37.2...
2	Santa Clara	06085	1958087	0.764	MULTIPOLYGON (((-121.63084 37.48278, -121.6354...
3	San Benito	06069	59354	0.739	POLYGON ((-121.57999 36.89823, -121.57988 36.8...
4	Ventura	06111	861790	0.777	MULTIPOLYGON (((-119.53461 33.28535, -119.5344...

Then we merged our mask dataset, now with county information, with our hospital GeoDataFrame.

In [38]:
```
# merge mask and hospital data
hospitals_df = pd.merge(left=hospitals_staff_ppe_covid_df, right=mask_df, l
                        right_on='COUNTYFP', how='inner')

# select relevant columns
hospitals_df = hospitals_df[['hospital_n', 'county', 'fips', 'num_licens',
                             'log_ppe_per_staff_ratio', 'tot_covid_confirme
                             'log_covid_per_staff_ratio', 'ALWAYS', 'latitu

hospitals_df.head()
```

Out[38]:

	hospital_n	county	fips	num_licens	cum_ppe	ppe_per_staff_ratio	log_ppe_per_staff_ratio
0	EAST LOS ANGELES DOCTORS HOSPITAL	LOS ANGELES	06037	102	10240000.0	100392.156863	11.516839
1	LAKEWOOD REGIONAL MEDICAL CENTER	LOS ANGELES	06037	172	10240000.0	59534.883721	10.994318
2	MONROVIA MEMORIAL HOSPITAL	LOS ANGELES	06037	49	10240000.0	208979.591837	12.249992
3	MONTEREY PARK HOSPITAL	LOS ANGELES	06037	101	10240000.0	101386.138614	11.526692
4	PALMDALE REGIONAL MEDICAL CENTER	LOS ANGELES	06037	184	10240000.0	55652.173913	10.926876

Putting It All Together

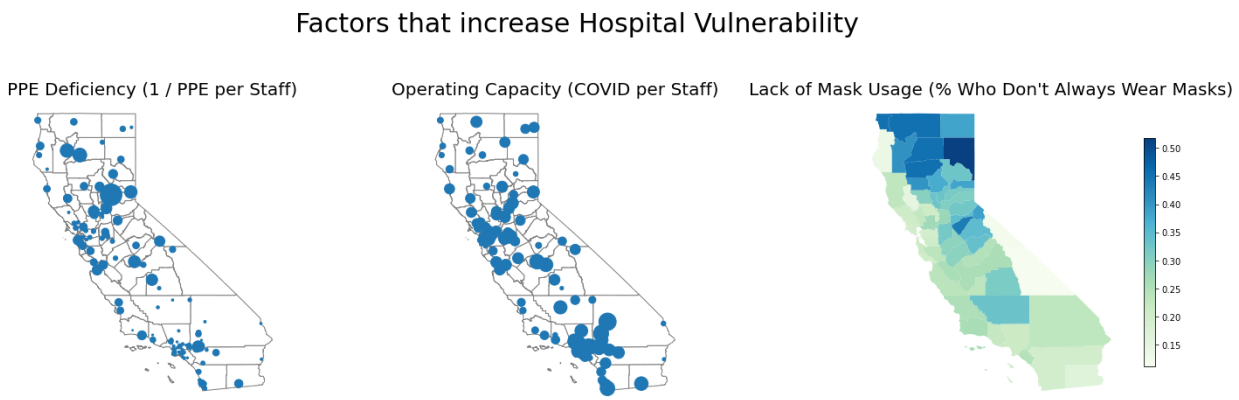
Finally, let's visualize each of our 3 variables. We plotted each variable and increased either 1) the size of the corresponding hospital or 2) the color saturation of the county to reflect the variable's magnitude at that geographic location.

In [39]:
```
# creating a geometry column
geometry = [Point(xy) for xy in zip(hospitals_df['longitude'], hospitals_df
# Coordinate reference system : WGS84
crs = {'init': 'epsg:4326'}
# Creating a Geographic data frame
hospitals_df = GeoDataFrame(hospitals_df, crs=crs, geometry=geometry)
```

```
/usr/local/anaconda3/lib/python3.8/site-packages/pyproj/crs/crs.py:53: Fu
tureWarning: '+init=<authority>:<code>' syntax is deprecated. '<authority
>:<code>' is the preferred initialization method. When making the change,
be mindful of axis order changes: https://pyproj4.github.io/pyproj/stabl
e/gotchas.html#axis-order-changes-in-proj-6 (https://pyproj4.github.io/py
proj/stable/gotchas.html#axis-order-changes-in-proj-6)
  return _prepare_from_string(" ".join(pjargs))
```

In [40]:
```
fig, ax = plt.subplots(1, 3, figsize=(25, 8), gridspec_kw={'width_ratios':

divider = make_axes_locatable(ax[0])
base = mask_county_df.plot(color='white', edgecolor='grey', ax=ax[0], legen
hospitals_df.plot(markersize=(1.0 / hospitals_df['ppe_per_staff_ratio'] * 5
ax[0].set_axis_off()
ax[0].set_title('PPE Deficiency (1 / PPE per Staff)', fontsize=20)

divider = make_axes_locatable(ax[1])
base = mask_county_df.plot(color='white', edgecolor='grey', ax=ax[1], legen
hospitals_df.plot(markersize=hospitals_df['log_covid_per_staff_ratio']*200,
ax[1].set_axis_off()
ax[1].set_title('Operating Capacity (COVID per Staff)', fontsize=20)

divider = make_axes_locatable(ax[2])
mask_county_df['NOT_ALWAYS'] = (1.0 - mask_county_df['ALWAYS'])
base = mask_county_df.plot(column='NOT_ALWAYS', ax=ax[2], cmap='GnBu', lege
ax[2].set_axis_off()
ax[2].set_title('Lack of Mask Usage (% Who Don\'t Always Wear Masks)', font

plt.suptitle('Factors that increase Hospital Vulnerability', fontsize=30)
```

Out[40]:
```
Text(0.5, 0.98, 'Factors that increase Hospital Vulnerability')
```



Looking at our visualization, we noticed some pretty obvious trends here. Not only are there hospitals clustered around big cities, but those same hospitals also have relatively high operating capacities. This makes sense because big cities have very dense populations, which increases the infection rate of COVID and the subsequent burden on hospitals. **Surprisingly, even though operating capacities are high in these big cities, their hospitals are actually relatively well-stocked on PPE.** This may be because metropolitans can afford to buy more PPE from vendors.

On the other hand, **counties in the bend of California (near Lake Tahoe) are experiencing the worst PPE crises in the entire state.** They are also the least likely of all Californians to wear a mask in public, with only 50% of them saying they do, drawing further concern towards their COVID-readiness.

K-Means Analysis

For our final analysis, we ran a k-means clustering algorithm to determine the optimal PPE distribution channels for hospitals in California. We weighed the location of each hospital against the 3 variables we discussed, producing an aggregate "vulnerability" score. **Our goal was to perform a cluster analysis that considered more than just Euclidean distance.** Our model is as follows:

$$Vulnerability = (\frac{1}{PPE\ per\ staff}) * (COVID\ per\ staff) * (\frac{1}{Mask\ Usage})$$

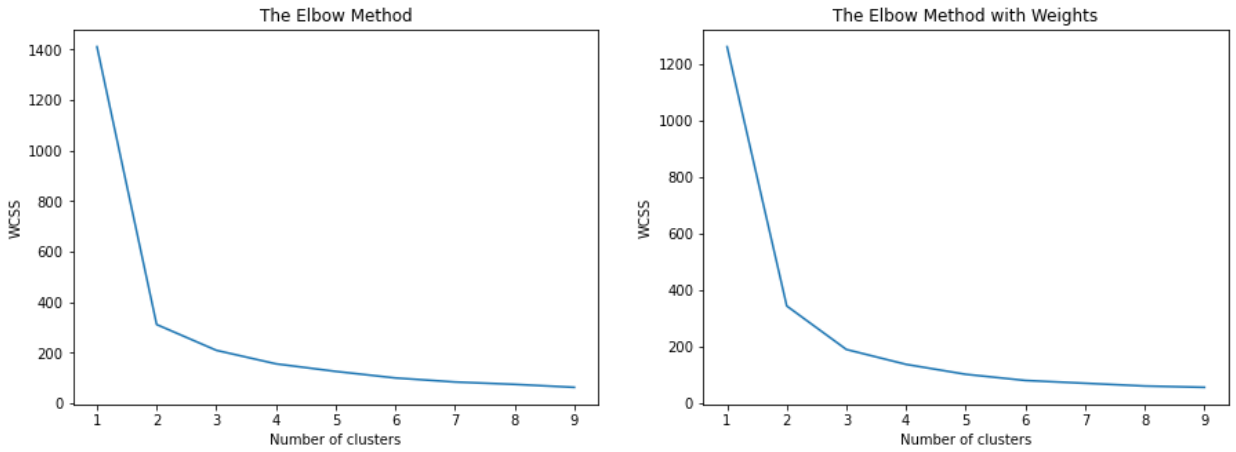
$$= \frac{COVID\ per\ staff}{(PPE\ per\ staff) * (Mask\ Usage)}$$

For each plot, we ran the algorithm twice: once without any "vulnerability" weights, and once with all 3. They are graphed side-by-side to help readers better understand the weights' influence on the resulting clusters.

In [41]:
```
# create ndarray of locations for k-means
a = hospitals_df['longitude']
b = hospitals_df['latitude']
X = np.column_stack((a,b))

# calculate weights
weight_arr = (1.0 / hospitals_df['ppe_per_staff_ratio']) * hospitals_df['co
(1.0 / hospitals_df['ALWAYS'])
```

In [42]:
```
f, ax = plt.subplots(1, 2, figsize=(15, 5))

# elbow method without weights
wcss = []
for i in range(1, 10):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
ax[0].plot(range(1, 10), wcss)
ax[0].set_title('The Elbow Method')
ax[0].set_xlabel('Number of clusters')
ax[0].set_ylabel('WCSS')

# elbow method with weights
wcss = []
for i in range(1, 10):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state=42)
    kmeans.fit(X, sample_weight=weight_arr)
    wcss.append(kmeans.inertia_)
ax[1].plot(range(1, 10), wcss)
ax[1].set_title('The Elbow Method with Weights')
ax[1].set_xlabel('Number of clusters')
ax[1].set_ylabel('WCSS')
```

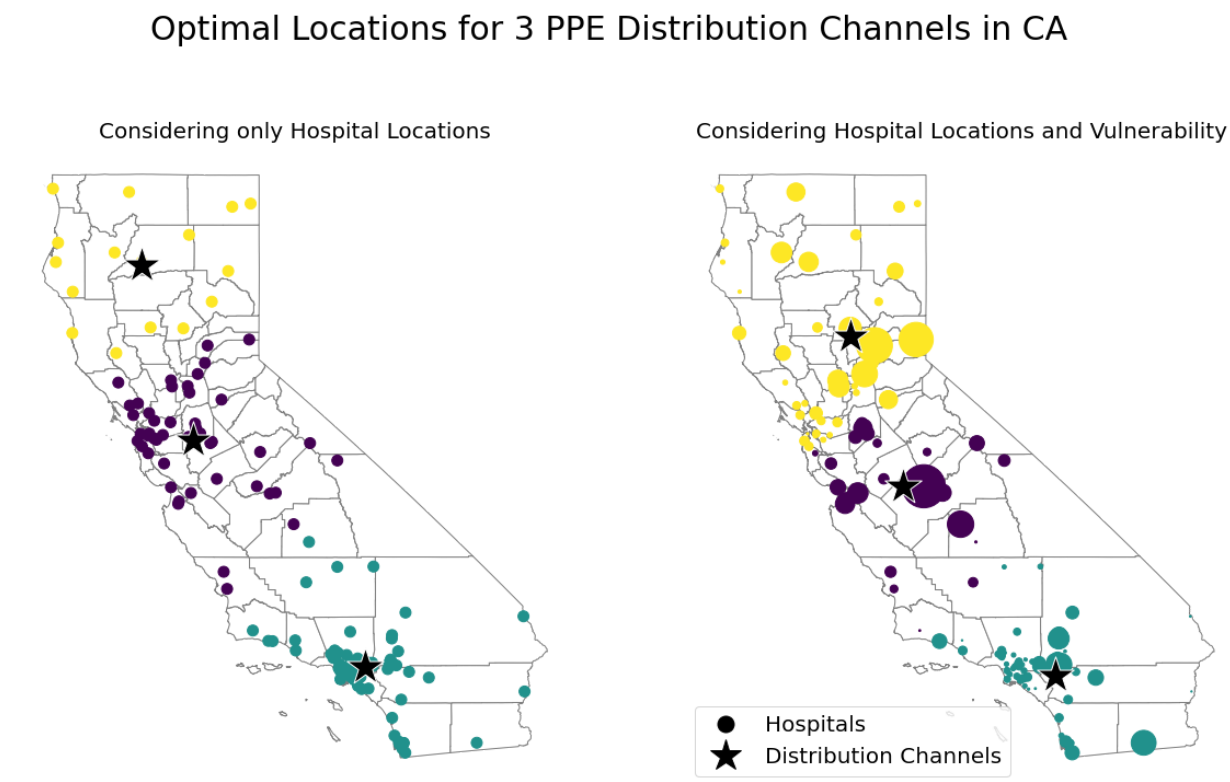
Out[42]:
```
Text(0, 0.5, 'WCSS')
```



We used the Elbow Method to determine the optimal number of clusters to include, reducing within-cluster sum of squares without overfitting. Notice that running the algorithm with weights seemed to increase variance in the data. Where we previously only needed 2 clusters to sufficiently explain the data (probably the LA/SF clusters), we now need at least 3 clusters, **suggesting the creation of a new point-of-interest.**

In [43]:
```
def plot_kmeans(k):

    '''Perform k-means clustering with k clusters, and plot comparison'''
    kmeans = KMeans(n_clusters = k, init = 'k-means++', max_iter=400, rando
    y_kmeans = kmeans.fit_predict(X)
    k_results = pd.DataFrame(y_kmeans, columns=['cluster' + str(k)])
    cluster_plot = hospitals_df.copy()
    cluster_plot = cluster_plot.join(k_results)

    f, ax = plt.subplots(1, 2, figsize=(20, 12))
    ax[0].set_title('Considering only Hospital Locations', fontsize=20)
    ax[0].set_axis_off()
    base = mask_county_df.plot(color='white', edgecolor='grey', ax=ax[0])
    cluster_plot.plot(column='cluster' + str(k), ax=base, cmap='viridis', m
    base.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,
                c='black', edgecolors='white', s=1250);

    kmeans = KMeans(n_clusters = k, init = 'k-means++', max_iter=400, rando
    y_kmeans = kmeans.fit_predict(X, sample_weight=weight_arr)
    k_results = pd.DataFrame(y_kmeans, columns=['cluster' + str(k)])
    cluster_plot = hospitals_df.copy()
    cluster_plot = cluster_plot.join(k_results)

    ax[1].set_title('Considering Hospital Locations and Vulnerability', fon
    ax[1].set_axis_off()
    base = mask_county_df.plot(color='white', edgecolor='grey', ax=ax[1])
    cluster_plot.plot(column='cluster' + str(k), ax=base, cmap='viridis', m
                      (1.0 / hospitals_df['ppe_per_staff_ratio'])*hospitals
                      (1.0 / hospitals_df['ALWAYS'])*1000000)
    base.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,
                c='black', edgecolors='white', s=1250);

    # legend
    hospitals_leg = mlines.Line2D([], [], color='black', marker='.', linest
                                  label='Hospitals')
    distribution_leg = mlines.Line2D([], [], color='black', marker='*', lin
                                     label='Distribution Channels')
    plt.legend(handles=[hospitals_leg, distribution_leg], markerscale=3, pr

    plt.suptitle('Optimal Locations for ' + str(k) + ' PPE Distribution Cha
    plt.show()
```

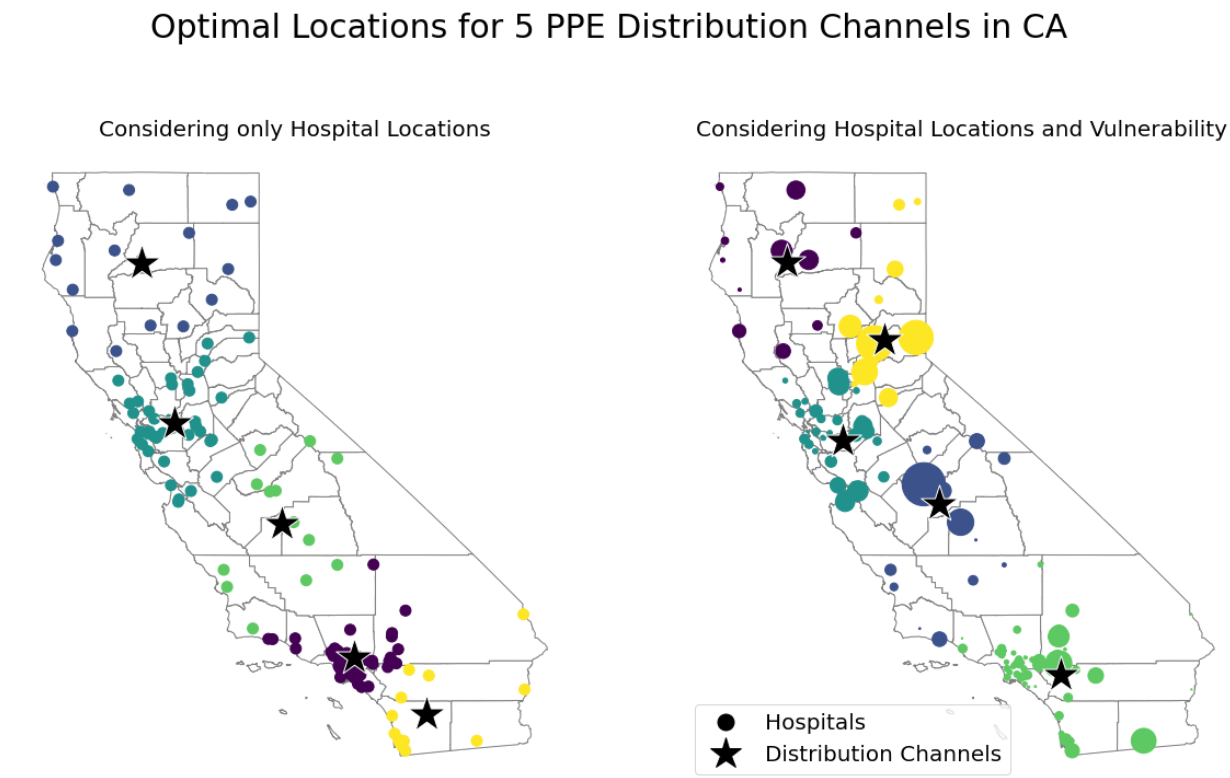
In [44]:
```
# k-mean with 3 clusters
plot_kmeans(3)
```



Running the k-means algorithm with 3 clusters yielded interesting results between no-weights and weights. **With the weights, we noticed that clusters shifted toward the bend of California**, exactly where we pointed it out earlier. There is also apparently a huge vulnerability risk in the lower Central Valley that the algorithm addressed.

We also noticed how Los Angeles posed minimal vulnerability risks despite its short-staffed hospitals. We speculated that this was due to Los Angeles' massive PPE storage and high mask usage.

In [45]:
```
# k-mean with 5 clusters
plot_kmeans(5)
```



Running the k-means algorithm with 5 clusters yielded similar results as before. **A cluster shifted from San Diego (boohoo!) to the bend of California, again addressing the more pressing vulnerability concerns in that area.** The rest of the clusters remained relatively unchanged.

Ethics & Privacy

No data that was collected used any personal information. Neither personal nor corporate privacy was violated in any way.

Overall, this project could really benefit others. Although the proverbial public may not see it, those in our immediate circles will see our project. It is a tool to inform and have a conversation about a confusing, worrisome, and controversial topic. It will also benefit hospitals, healthcare workers, and patients when hospitals receive the equipment they need.

Any research proving that masks are useful can be used as a reason to gouge prices and make substandard equipment. Prices of masks, gloves, and many other items have increased radically since the stay-at-home order began. Sellers could take advantage of others' fear and sell masks with faulty materials that don't offer adequate protection. Hospitals are not immune to corrupt vendors with inadequate supplies.

Furthermore, our data could be insufficient and too ever-changing to make real-life conclusions. Most COVID-19 data is still new and was gathered rapidly to fight the threat. If our data and conclusion are incorrect, we would be sending vital supplies to the wrong areas, leaving other hospitals potentially even worse off. It would also cost the federal and state governments an enormous amount of money, as well as tarnish their reputations.

Conclusion

We first visualized California's PPE needs by the county, then hypothesized that the total supply of PPE helped reduce COVID-19 reproduction rates (Rt). Though we did not find statistically significant evidence to support our claim, we did find some correlation between large PPE shipments and lower cases of new COVID-19 infections.

We postulated that factors such as PPE Supply, Operating Capacity, and Mask Usage could be used to model a hospital's vulnerability to COVID-19. Our data showed us that big cities were operating near full-capacity, yet smaller cities were facing PPE shortages as well. We applied these factors to a k-means clustering algorithm and concluded that PPE distribution could be optimized to better protect vulnerable Californians during the COVID-19 pandemic.

Specifically, hospitals near lower Central Valley and the bend of California are being neglected by the federal government when it comes to PPE distribution. Their staff are overworked, they lack the proper PPE, and the general public does not care enough to wear masks. As fellow Californians, it is our duty to pay attention to these often forgotten cities and lend them a helping hand in times of need.

Discussion

After all the cleaning and wrangling, we only had ~130 data points on hospitals in California. For reference, there are 341 hospitals in California as of 2020. This logistical error was due to the "ESRI Definitive Healthcare" dataset we acquired from Kaggle, which only contained staff information on ~160 hospitals. We had to drop all other rows when merging datasets, and after cleaning for outliers, we were left with less than half of all existing hospitals to work with.

However, most of our datasets were taken from regularly-updated government institutions, so we can guarantee that our data are always as relevant and accurate as possible.

In order to further this research, we believe that by increasing our scope to other states beyond California, we will be able to offer a solution to the PPE shortage in some areas. If we could get a deeper understanding of mask usage and PPE shortages, we will be able to better educate the public and highlight important intervention methods that may prevent many lives in this pandemic.

In light of the recent news that vaccines have now been authorized and are being distributed throughout the US, we would like to expand this study to find the most optimal distribution method of the vaccine as well.

In [ ]: