When should California Highway Patrol increase staff to prevent car accidents?

Alli Kruger
Sowmya Srinivasan
Tosveena Thomas
Weiqi Liang (Vicky)

Problem: Accidents require more CHP resources

Motivation and Summary

- **Hypothesis:**
  - Weather and Time will affect accident frequency and severity
- **Data:**
  - Kaggle US Accidents dataset
  - Google Maps API
  - United States Census
- **Data Clean Up:**
  - Deleted irrelevant data
  - Filtered on California

- **Questions:**
  - What time of year has the highest accidents?
  - Most common time of day for accidents?
  - How are accidents related to weather?
    - Is there a correlation between weather conditions and accidents?
  - Are there certain counties and cities that CHP should focus on?

What month are more people in accidents?

	Severity	Start_Time	Start_Lat	Start_Lng	Street	City	County	State	Zipcode	Temperature(F)	Weather_Condition
0	3	2016-06-21 10:34:40	38.085300	-122.233017	Magazine St	Vallejo	Solano	CA	94591	75.0	Clear
1	3	2016-06-21 10:30:16	37.631813	-122.084167	I-880 N	Hayward	Alameda	CA	94544	75.0	Clear
2	2	2016-06-21 10:49:14	37.896564	-122.070717	I-680 N	Walnut Creek	Contra Costa	CA	94595	82.9	Clear
3	3	2016-06-21 10:41:42	37.334255	-122.032471	N De Anza Blvd	Cupertino	Santa Clara	CA	95014	75.9	Clear
4	2	2016-06-21 10:16:26	37.250729	-121.910713	Norman Y Mineta Hwy	San Jose	Santa Clara	CA	95118	75.2	Clear
...	...	...	...	...	...	...	...	...	...	...	...
663199	2	2019-08-23 18:03:25	34.002480	-117.379360	Pomona Fwy E	Riverside	Riverside	CA	92501	86.0	Fair
663200	2	2019-08-23 19:11:30	32.766960	-117.148060	I-8 W	San Diego	San Diego	CA	92108	70.0	Fair
663201	2	2019-08-23 19:00:21	33.775450	-117.847790	Garden Grove Fwy	Orange	Orange	CA	92866	73.0	Partly Cloudy
663202	2	2019-08-23 19:00:21	33.992460	-118.403020	San Diego Fwy S	Culver City	Los Angeles	CA	90230	71.0	Fair
663203	2	2019-08-23 18:52:06	34.133930	-117.230920	CA-210 W	Highland	San Bernardino	CA	92346	79.0	Fair

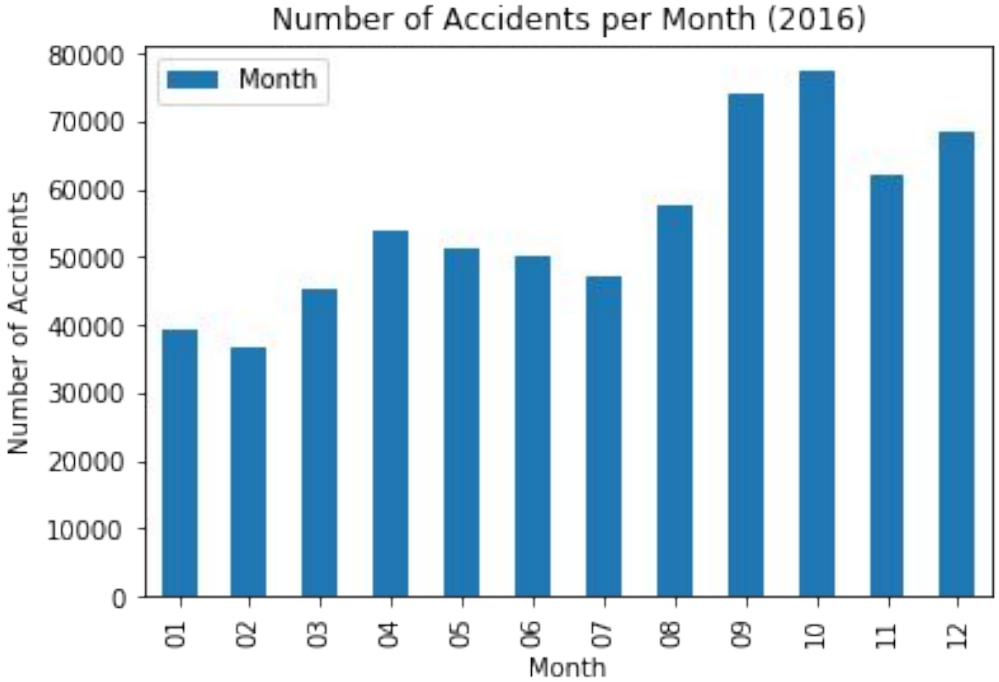
What month are more people in accidents? (cont.)

```
cleaned_accident["Month"]=""
cleaned_accident["Month"]=cleaned_accident["Start_Time"].str.slice(5,7,1)
cleaned_accident
```

	Severity	Start_Time	Start_Lat	Start_Lng	Street	City	County	State	Zipcode	Temperature(F)	Weather_Condition	Month
0	3	2016-06-21 10:34:40	38.085300	-122.233017	Magazine St	Vallejo	Solano	CA	94591	75.0	Clear	06
1	3	2016-06-21 10:30:16	37.631813	-122.084167	I-880 N	Hayward	Alameda	CA	94544	75.0	Clear	06
2	2	2016-06-21 10:49:14	37.896564	-122.070717	I-680 N	Walnut Creek	Contra Costa	CA	94595	82.9	Clear	06
3	3	2016-06-21 10:41:42	37.334255	-122.032471	N De Anza Blvd	Cupertino	Santa Clara	CA	95014	75.9	Clear	06
4	2	2016-06-21 10:16:26	37.250729	-121.910713	Norman Y Mineta Hwy	San Jose	Santa Clara	CA	95118	75.2	Clear	06
...	...	...	...	...	...	...	...	...	...	...	...	...
663199	2	2019-08-23 18:03:25	34.002480	-117.379360	Pomona Fwy E	Riverside	Riverside	CA	92501	86.0	Fair	08

Accidents per Month

- **Greater frequency of accidents in September, October and December.**



What time of day has the most accidents?

```
#pick out the time columns
time_accident_df['Start_Time']
```

```
0         2016-06-21 10:34:40
1         2016-06-21 10:30:16
2         2016-06-21 10:49:14
3         2016-06-21 10:41:42
4         2016-06-21 10:16:26
                 ...
663199    2019-08-23 18:03:25
663200    2019-08-23 19:11:30
663201    2019-08-23 19:00:21
663202    2019-08-23 19:00:21
663203    2019-08-23 18:52:06
Name: Start_Time, Length: 663204, dtype: object
```

```
#add a "hour" column to the DataFrame
time_accident_df['hour']=pd.to_datetime(time_accident_df['Start_Time']).dt.hour
#preview the DF
time_accident_df
```

	ID	Severity	Start_Time	Start_Lat	Start_Lng	Street	City	County	State	Zipcode	Temperature(F)	Weather_Condition	hour
0	A-729	3	2016-06-21 10:34:40	38.085300	-122.233017	Magazine St	Vallejo	Solano	CA	94591	75.0	Clear	10
1	A-730	3	2016-06-21 10:30:16	37.631813	-122.084167	I-880 N	Hayward	Alameda	CA	94544	75.0	Clear	10

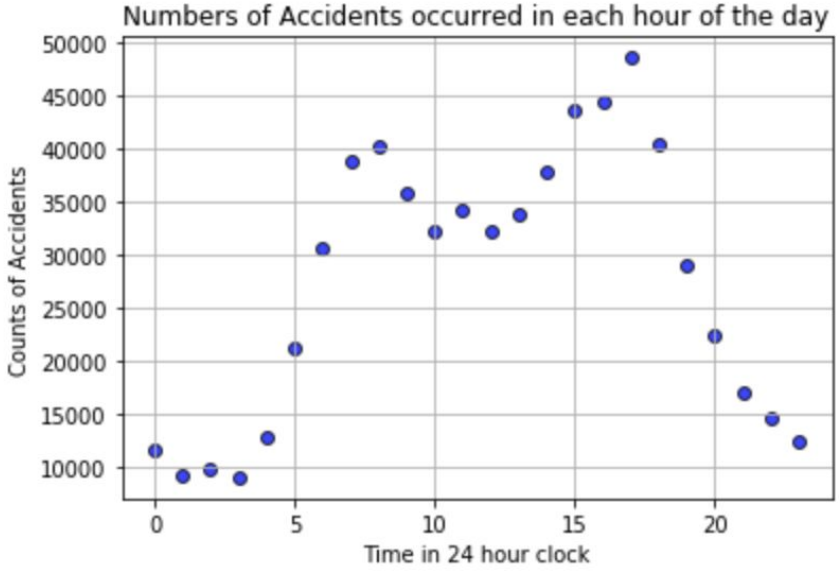
Hour counts & Reindexing

```
#pick out the hour columns
hour=time_accident_df['hour']
#did a value count to check the most frequent hour
hour_count=hour.value_counts()
hour_reindex=hour_count.sort_index(ascending=True)
hour_reindex
```

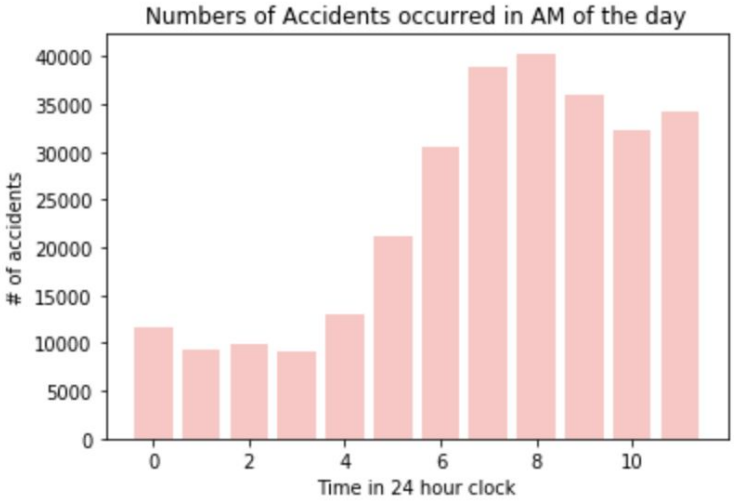
```
0     11721
1      9238
2      9916
3      9088
4     12933
5     21200
```

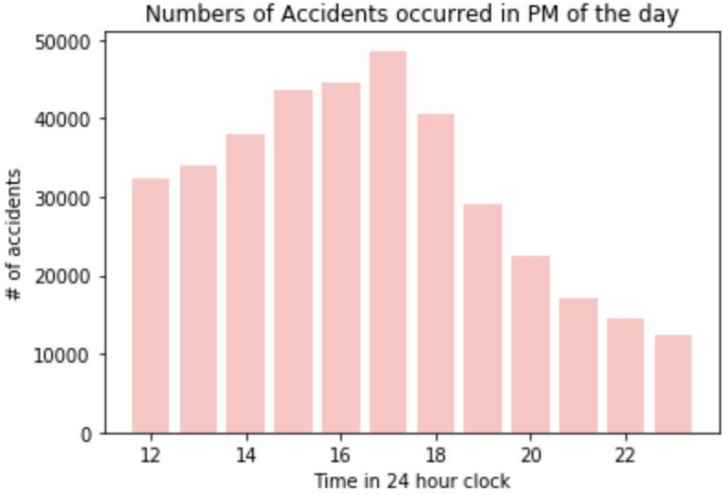
Accidents by hour

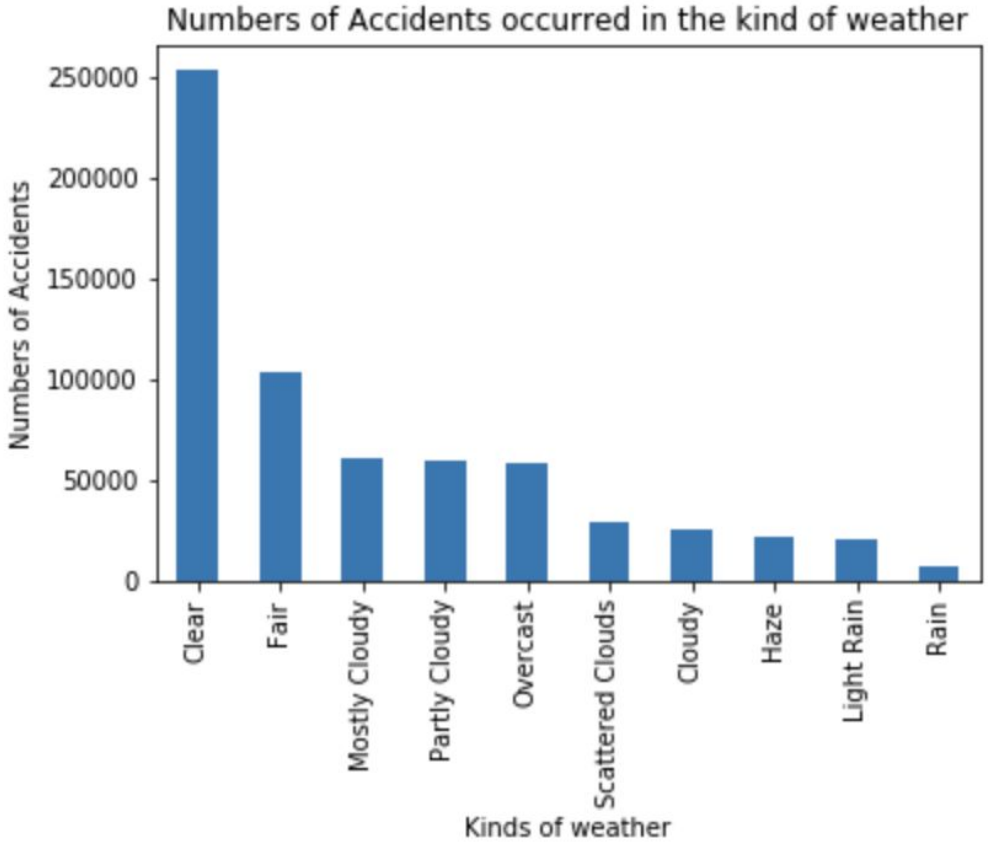
```
hour_reindex.plot(kind="bar")
plt.xlabel("Time in 24 hour clock")
plt.ylabel("# of accidents")
plt.title("Numbers of Accidents occurred in each hour of the day")
```



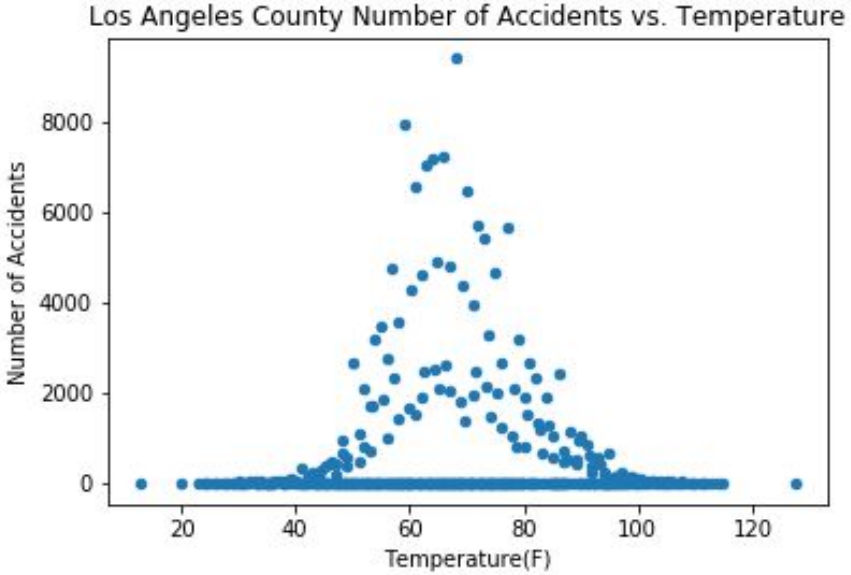
Morning vs Evening

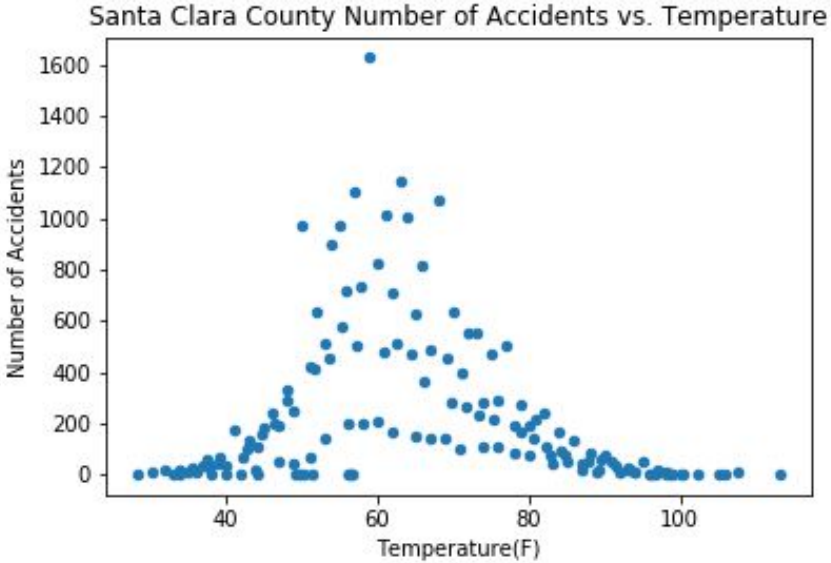




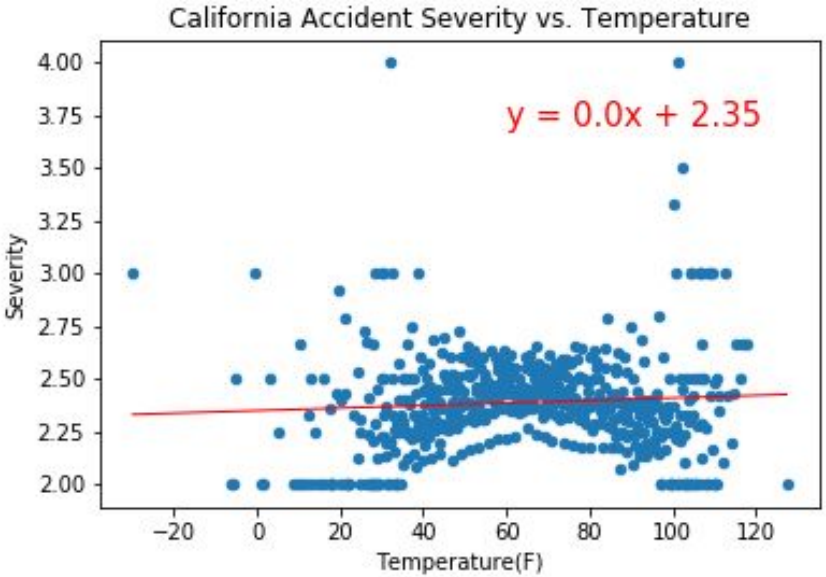

Numbers of Accidents occurred in the kind of weather
Numbers of Accidents
250000
200000
150000
100000
50000
0
Clear
Fair
Mostly Cloudy
Partly Cloudy
Overcast
Scattered Clouds
Cloudy
Haze
Light Rain
Rain
Kinds of weather

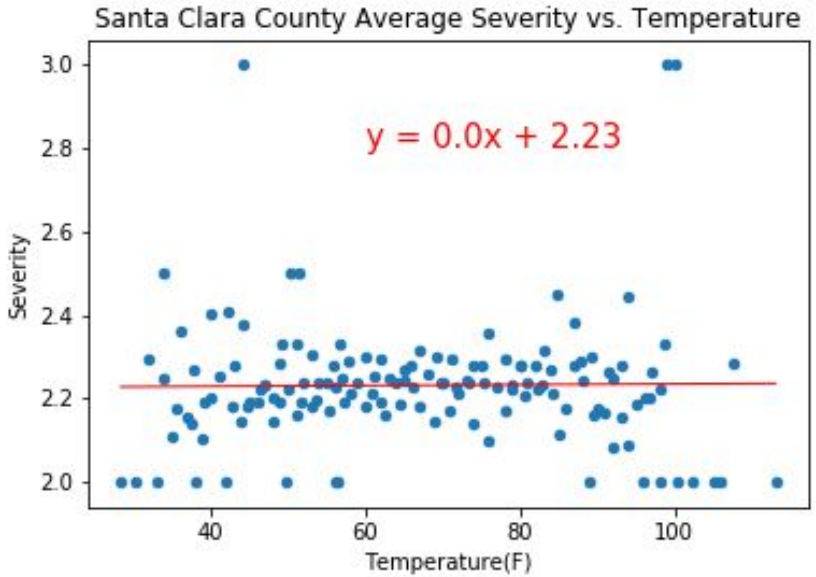
Does weather affect the number of accidents?





Does weather affect the average severity of accidents?





Most "Dangerous" Road

In [4]:
```
street_severity = cal.groupby('Street').mean()['Severity']
```

In [5]:
```
street_count = cal.groupby('Street').count()['Start_Time']
```

In [6]:
```
road = pd.concat([street_severity, street_count], axis=1).reset_index()
road = road.rename(columns={'Start_Time': 'Number of Accidents'})
most_severe_road = road.loc[road['Severity'] == 4].sort_values('Number of Accidents', ascending=False)
most_accidents_road = road.loc[road['Number of Accidents'].idxmax()]
```

In [7]:
```
#Streets with highest average accident severity
most_severe_road.head()
```

Out[7]:

	Street	Severity	Number of Accidents
4223	S Coast Hwy	4.0	4
4929	Sunflower Ave	4.0	4
411	Barstow Rd	4.0	4
4303	S Harbor Blvd	4.0	4
2513	Lake St	4.0	3

In [8]:
```
#Street with highest number of accidents
road_accidents_df = pd.DataFrame(most_accidents_road).T
road_accidents_df
```

Out[8]:

	Street	Severity	Number of Accidents
14534	I-5 N	2.62842	20004

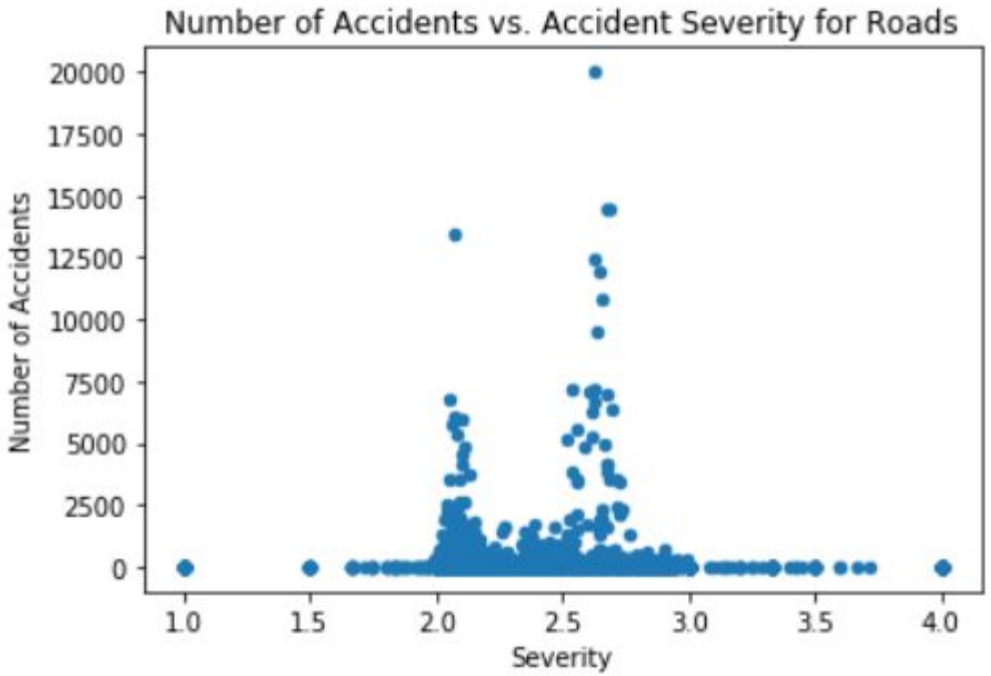
Roads: Accidents vs Severity

Largest number of accidents:

	Street	Severity	Number of Accidents
14534	I-5 N	2.62842	20004

Highest Average Accident Severity:

	Street	Severity	Number of Accidents
4223	S Coast Hwy	4.0	4
4929	Sunflower Ave	4.0	4
411	Barstow Rd	4.0	4
4303	S Harbor Blvd	4.0	4
2513	Lake St	4.0	3



Most "Dangerous" County

In [9]:
```
county_severity = cal.groupby('County').mean()['Severity']
```

In [10]:
```
county_count = cal.groupby('County').count()['Start_Time']
```

In [11]:
```
county = pd.concat([county_severity, county_count], axis=1).reset_index()
county = county.rename(columns={'Start_Time': 'Count'})
most_severe_county = county.loc[county['Severity'].idxmax()]
most_accidents_county = county.loc[county['Count'].idxmax()]
```

In [12]:
```
#County with highest average accident severity
county_severity_df = pd.DataFrame(most_severe_county).T
county_severity_df
```

Out[12]:

	County	Severity	Count
47	Solano	2.55253	10032

In [13]:
```
#County with largest number of accidents
county_frequency_df = pd.DataFrame(most_accidents_county).T
county_frequency_df
```

Out[13]:

	County	Severity	Count
18	Los Angeles	2.40523	227180

In [18]:
```
county
```

Out[18]:

	County	Severity	Count
0	Alameda	2.537417	45367
1	Alpine	2.250000	60
2	Amador	2.060686	758
3	Butte	2.071479	1413
4	Calaveras	2.053937	927
5	Colusa	2.269333	375

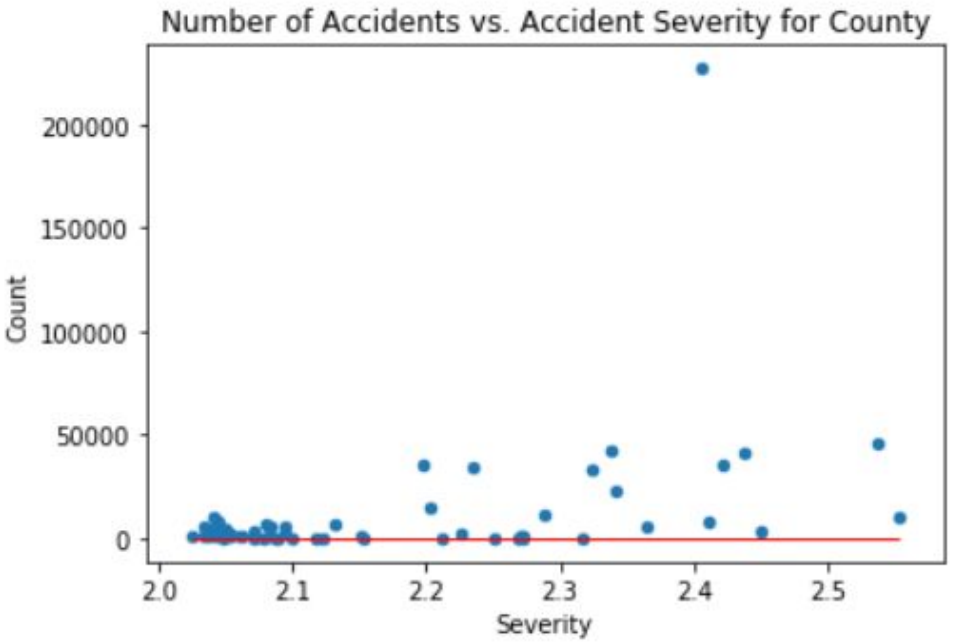
County: Accidents vs Severity

Largest number of accidents:

	County	Severity	Count
18	Los Angeles	2.40523	227180

Highest Average Accident Severity:

	County	Severity	Count
47	Solano	2.55253	10032



Should CHP focus on certain counties or cities?

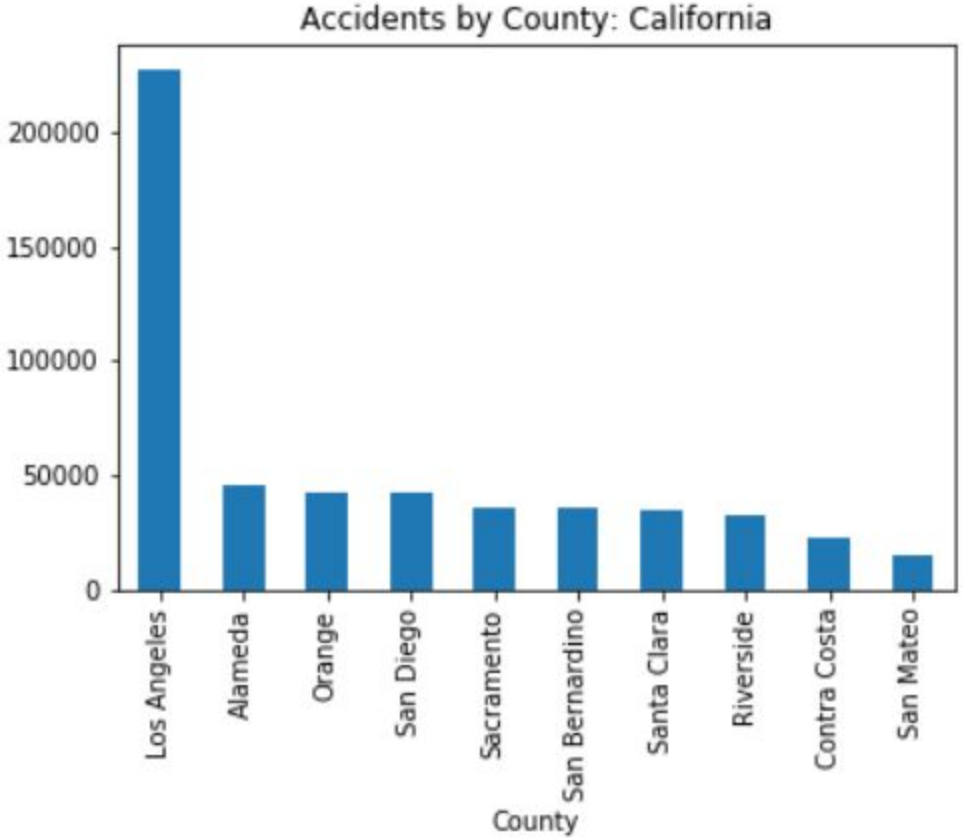
- **Helpful to CHP for allocating staff and resources**
- **By county and city**
- **Adjust City for population**

Top Counties by Number of Accidents

- **Counted the number of accidents by County**
- **Selected only the top 10 of the county_count data frame**
- **Created a bar graph showing Accident by County**

In [47]:
```
county_count = cal.groupby('County').count()
city_count = cal.groupby('City').count()

# city_count
```

In [9]:
```
# California counties by number of accidents (normalized by population)
counties = county_count.Severity.sort_values(ascending = False)
top_counties = counties.nlargest(n=10)

counties_plot = top_counties.plot(kind = 'bar', title = 'Accidents by County: California')
# plt.savefig('Images/AccidentbyCounty.png')
```

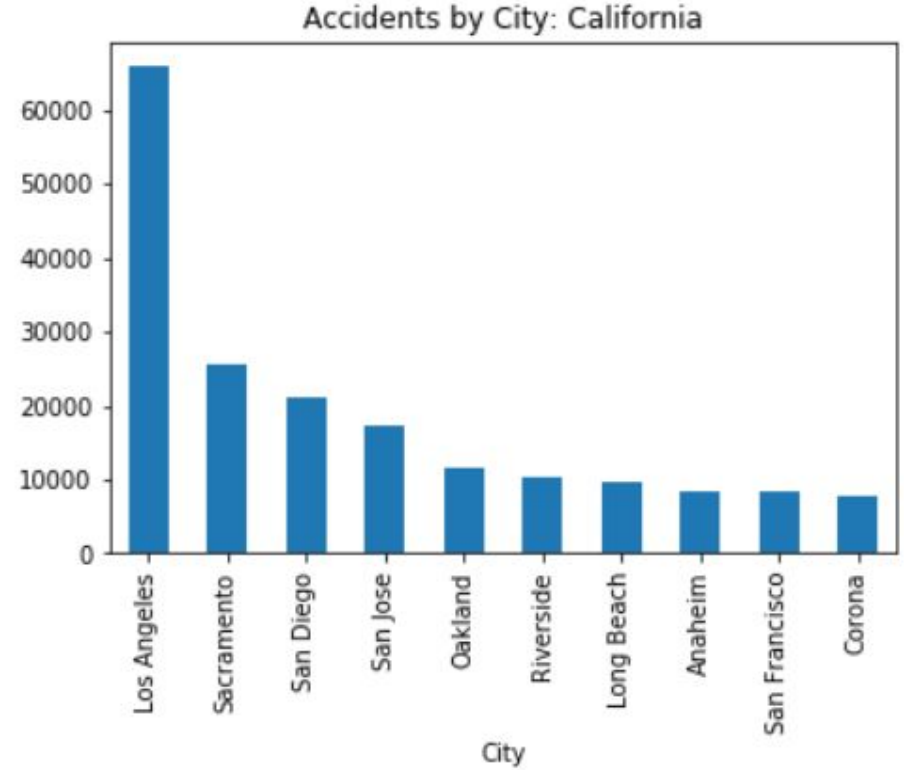

Accidents by County: California
200000
150000
100000
50000
0
Los Angeles
Alameda
Orange
San Diego
Sacramento
San Bernardino
Santa Clara
Riverside
Contra Costa
San Mateo
County

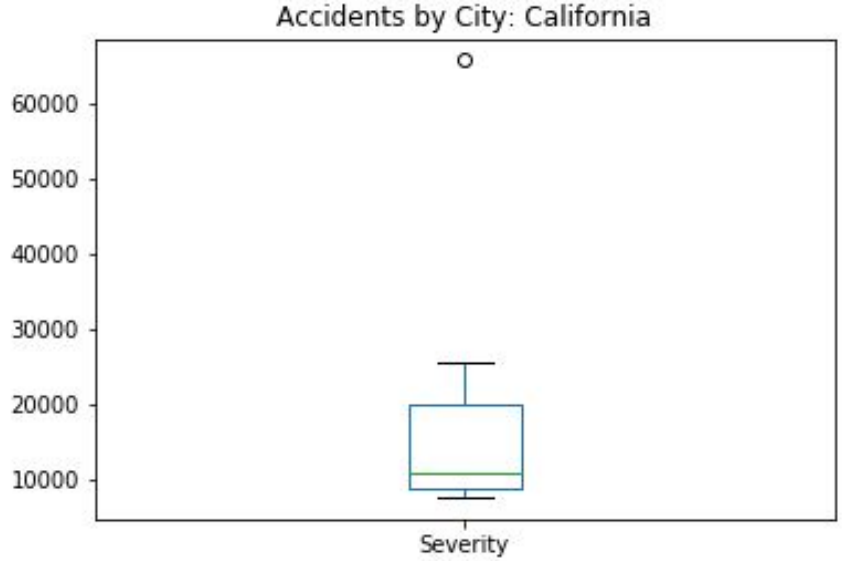
Top 10 Cities by Number of Accidents

- **Counted the number of accidents by City**
- **Selected only the top 10 of the city_count data frame**
- **Created a bar graph showing Accident by City**

In [15]:

```
# Top 5 Cities that have the most accidents
cities_count = city_count.Severity.sort_values(ascending = False)
top_cities = cities_count.nlargest(n =10)

cities_plot = top_cities.plot(kind = 'bar', title = 'Accidents by City: California')

# plt.savefig('Images/AccidentbyCity_notadj.png')
```


Accidents by City: California
60000
50000
40000
30000
20000
10000
0
Los Angeles
Sacramento
San Diego
San Jose
Oakland
Riverside
Long Beach
Anaheim
San Francisco
Corona
City


Accidents by City: California
60000
50000
40000
30000
20000
10000
Severity

Adjusting for Population: Data Wrangling

- **Vintage 2016 Estimated Population Census API provided Data**
- **Created API call to retrieve population estimates for all cities in California**
- **Selected for row 1 to the end**
- **Created empty lists**
- **Appended data to lists**

In [12]:

```
# to adjust for population we will need the data for population from the census
# COUNTY ⟶ County FIPS code https://api.census.gov/data/2016/pep/population/variables.html
# 050 is State > County https://api.census.gov/data/2016/pep/population
# https://api.census.gov/data/2016/pep/population?get=GEONAME,POP&for=county:*&in=state:*

#If you are having trouble with your config file you can instert your api key below
# c_key =
state = "06"
county = "001"

pop_url = "https://api.census.gov/data/2016/pep/population"

query_url = f"{pop_url}?get=GEONAME,POP&for=place:*&in=state:06&key={c_key}"
# query_url = f"{pop_url}?get=GEONAME,POP&for=county:*&in=state:06&key={c_key}"

GEONAME = []
POP = []
state = []
place = []

# https://api.census.gov/data/2016/pep/population?get=GEONAME,POP&for=place:*&in=state:01%20county:003
# https://api.census.gov/data/2016/pep/population?get=GEONAME,POP&for=county:*&in=state:*
response = requests.get(query_url)

if response:
#     Debugging print statements
    print("GET URL: " + response.url)
    print("STATUS CODE: " + str(response.status_code))
    response_json = response.json()
#     print(response_json)

    for data in response_json:

        GEONAME.append(data[0])
        POP.append(data[1])
        state.append(data[2])
        place.append(data[3])


else:
    print("API REQUEST ERROR")
    print("STATUS CODE: " + str(response.status_code))
GEONAME = GEONAME[1:]
POP = POP[1:]
state = state[1:]
place = place[1:]
```

Adjusting for Population: Data Mapping

- **Mapped lists to a dictionary**
- **Created a dataframe from the dictionary**
- **Changed Population to a float rather than a string**

```
# Take the lists and make a dictionary
census_dict = {
    "City": GEONAME,
    "Population": POP,
    "State": state,
    "County": place
}

# Make a data frame from the dictionary
census_data = pd.DataFrame(census_dict)

census_data.head(2)
```

	City	Population	State	County
0	Arvin city, California	21086	06	02924
1	Atascadero city, California	30330	06	03064

```
census_data['Population'] = census_data['Population'].astype(float)

census_data.sort_values('Population', ascending = False)

```

	City	Population	State	County
217	Los Angeles city, California	3976322.0	06	44000
364	San Diego city, California	1406630.0	06	66000
406	San Jose city, California	1025350.0	06	68000
384	San Francisco city, California	870887.0	06	67000
233	Fresno city, California	522053.0	06	27000

Adjusting for Population: Data Cleaning

- **Census Data had City and State as a single string**
- **Splitting the string on the delimiter mapped the state to the correct column**
- **Slicing allowed us to refine the city name.**
- **Selected the top 10 cities from the census data set**

```
census_data[['City','State']] = census_data.City.str.split(",",expand=True)
```

```
census_data["City2"] = census_data.City.str.slice(0,4,1)
census_data["City2"] = census_data.City.str.slice(0,-5,1)

census_data.head(2)
```

	City	Population	State	County	City2
0	Arvin city, California	21086	06	02924	Arvin city, Calif
1	Atascadero city, California	30330	06	03064	Atascadero city, Calif

```
census_data = census_data[['City2', 'Population', 'State','County']]

census_data.rename(columns = {'City2':'City'}, inplace = True)
```

```
census_data

options = ['Los Angeles',
           'Sacramento',
           'San Diego',
           'San Jose',
           'Oakland',
           'Riverside',
           'Long Beach',
           'Anaheim',
           'San Francisco',
           'Corona']

top_cities_pop = census_data[census_data['City'].isin(options)]

print('\nResult dataframe :\n', top_cities_pop)
```

```
Result dataframe :
             City  Population       State County
44         Anaheim    351043.0  California  02000
212     Long Beach    470130.0  California  43000
217    Los Angeles   3976322.0  California  44000
```

Adjusting for Population: Data Cleaning

- **Merged census data frame and top accident cities**
- **Renamed the columns in the new data frame**

```
top_cities = top_cities.to_frame()
```

```
top_cities

top_cites_adj = top_cities.merge(top_cities_pop, on='City')
```

```
# check data frame for errors
top_cites_adj
```

	City	Severity	Population	State	County
0	Los Angeles	65851	3976322.0	California	44000
1	Sacramento	25657	495234.0	California	64000

```
adj_accidents = top_cites_adj.Severity/top_cites_adj.Population

adj_accidents = adj_accidents.to_frame()

top_cites_adj = top_cites_adj.merge(adj_accidents, left_index = True, right_index = True)
```

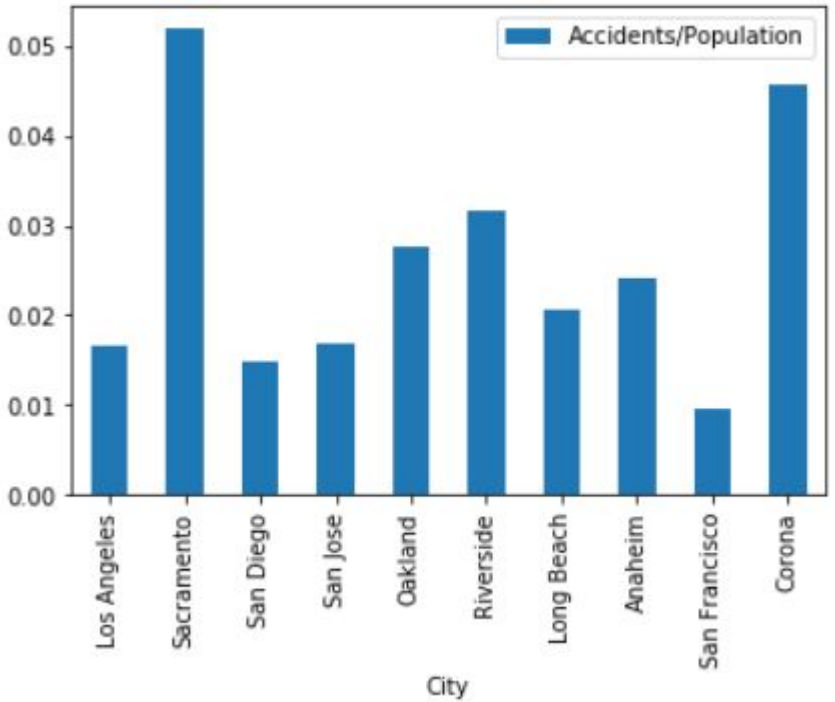
```
top_cites_adj

list(top_cites_adj.columns)
```

```
['City', 'Severity', 'Population', 'State', 'County', 0]
```

```
top_cites_adj = top_cites_adj.rename(columns = {0:'Accidents/Population',
                                                'Severity': 'Accident Count'
                                               })
```

Top Cities Adjusted for population

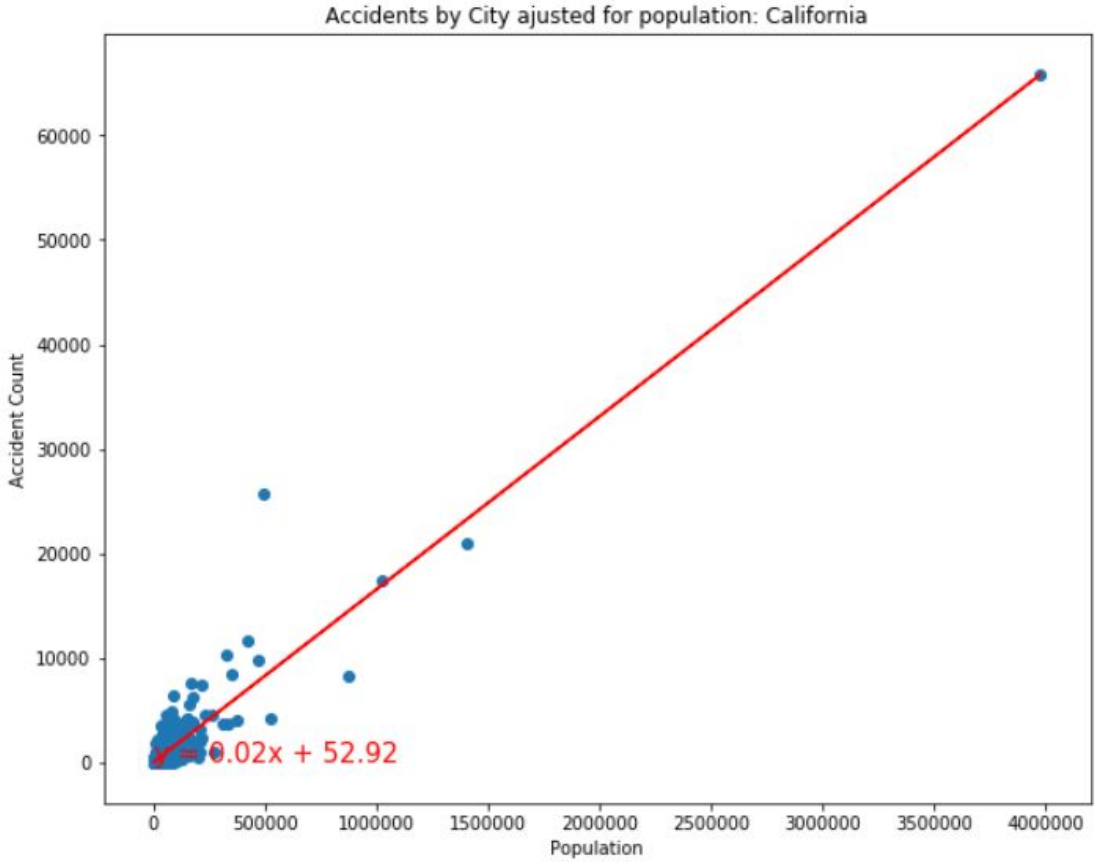
- **Los Angeles was only number one due to population**
- **Sacramento & Corona have more accidents per person**



Accident Count vs Population

- There is a strong relationship between accident count and population.
- R squared = 0.93876
- The more people there are the more accidents happen.

R squared: 0.9387640334862776



California Accidents 2016 - Heat Map

Findings

- Weather does not have any effect on the frequency of accidents
- There is a correlation between Month and the frequencies of accidents
- Population affects the number of accidents

Recommendations

- Increase amount of CHP officers in the late summer/ early winter months
- Increase amount of CHP officers during commute hours
- Staff should be allocated by population for cities

Post mortem

- **Difficulties:**
  - **Not enough information from the data sources.**
  - **Binning similar categories for weather was difficult.**
  - **Not having type of accident and not having a numerical value weather made quantitative analysis difficult.**
  - **Census API documentation is confusing.**
- **Further research:**
  - **Accident data from 2016 to present.**
  - **Get data on type of accident.**
  - **Why do late summer/early winter month have more accidents?**
  - **Relationship between CHP presence and Accidents**
  - **Why does Sacramento have so many accidents per person**
  - **Adjust weather data to be proportional to weather type**

Questions?