In [1]:
```
# Dependencies and Setup
import json
import requests
import pandas as pd
import gmaps
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
#Read in csv files: original file is too big to read in as a whole.
location = "locations.csv"
location_df = pd.read_csv(location)

date_time = "date_time.csv"
date_time_df = pd.read_csv(date_time)

weather = "weather.csv"
weather_df = pd.read_csv(weather)

severity = "severity.csv"
severity_df = pd.read_csv(severity)
```

In [3]:
```
#Merge files
df = pd.concat([date_time_df, location_df, severity_df, weather_df], axis=1, join='inner')
#df.set_index('ID', inplace=True)
accident_df = pd.DataFrame(df)
```

In [4]:
```
#Ensure data is read in and merged correctly
accident_df.head()
```

Out[4]:

	ID	Date	Time	Month	Day	Year	ID	Street	City	State	...	Severity	Side	ID	Temperature(F)	Humidity(%)	Visibility(mi)	W
0	A-10000	1/6/2017	4:22 PM	January	Friday	2017	A-10000	I-80 W	West Sacramento	CA	...	3	R	A-10000	46.0	71.0	10.0	
1	A-10001	1/6/2017	4:18 PM	January	Friday	2017	A-10001	CA-99 N	Salida	CA	...	2	R	A-10001	48.9	52.0	10.0	
2	A-10002	1/6/2017	4:17 PM	January	Friday	2017	A-10002	CA-99 N	Salida	CA	...	2	R	A-10002	48.9	52.0	10.0	
3	A-10003	1/6/2017	4:27 PM	January	Friday	2017	A-10003	CA-99 N	Salida	CA	...	2	R	A-10003	48.0	56.0	10.0	
4	A-10004	1/6/2017	4:40 PM	January	Friday	2017	A-10004	County Road 95	Davis	CA	...	2	L	A-10004	44.6	61.0	NaN	

5 rows × 21 columns

In [5]:
```
#Basic statistics of the data
accident_df.describe()
```

Out[5]:

	Year	Severity	Temperature(F)	Humidity(%)	Visibility(mi)	Wind_Speed(mph)	Precipitation(in)
count	438997.000000	438997.000000	429128.000000	428541.000000	426058.000000	357112.000000	45349.000000
mean	2016.466422	2.367148	63.708555	65.338901	9.184726	8.982098	0.075767
std	0.498872	0.484799	17.232990	21.334785	2.193992	4.633510	0.600242
min	2016.000000	0.000000	-77.800000	4.000000	0.000000	1.200000	0.000000
25%	2016.000000	2.000000	53.100000	50.000000	10.000000	5.800000	0.000000
50%	2016.000000	2.000000	65.500000	67.000000	10.000000	8.100000	0.010000
75%	2017.000000	3.000000	77.000000	83.000000	10.000000	11.500000	0.040000
max	2017.000000	4.000000	123.800000	100.000000	105.000000	241.700000	10.140000

In [6]:
```
#Look for null objects
accident_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 438997 entries, 0 to 438996
Data columns (total 21 columns):
ID                  438997 non-null object
Date                438997 non-null object
Time                438997 non-null object
Month               438997 non-null object
Day                 438997 non-null object
Year                438997 non-null int64
ID                  438997 non-null object
Street              438997 non-null object
City                438979 non-null object
State               438997 non-null object
ID                  438997 non-null object
Severity            438997 non-null int64
Side                438997 non-null object
ID                  438997 non-null object
Temperature(F)      429128 non-null float64
Humidity(%)         428541 non-null float64
Visibility(mi)      426058 non-null float64
Wind_Direction      431998 non-null object
Wind_Speed(mph)     357112 non-null float64
Precipitation(in)   45349 non-null float64
Weather_Condition   426398 non-null object
dtypes: float64(5), int64(2), object(14)
memory usage: 70.3+ MB
```

In [7]:

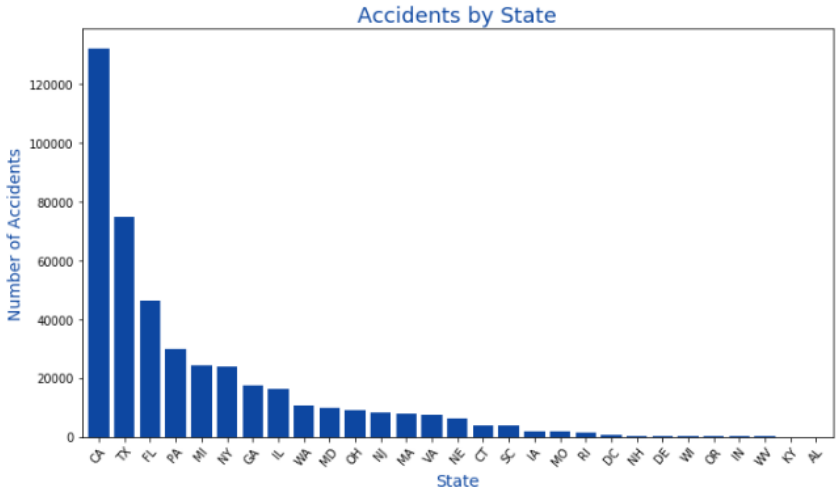
```
#Find out number of states
print("State names in Dataset")
print(accident_df.State.unique())

#Find dates in dataset
print("\nDates in Dataset")
print(np.sort(accident_df.Date.unique()))

#Find different weather conditions
print("\nWeather conditions in Dataset")
print(accident_df.Weather_Condition.unique())
```

```
State names in Dataset
['CA' 'FL' 'GA' 'SC' 'NE' 'IA' 'IL' 'MO' 'WI' 'IN' 'MI' 'OH' 'NJ' 'NY'
 'CT' 'MA' 'RI' 'NH' 'PA' 'MD' 'VA' 'DC' 'DE' 'WV' 'TX' 'WA' 'OR' 'KY'
 'AL']

Dates in Dataset
['1/1/2017' '1/10/2017' '1/11/2017' '1/12/2017' '1/13/2017' '1/14/2017'
 '1/15/2017' '1/16/2017' '1/17/2017' '1/18/2017' '1/19/2017' '1/2/2017'
 '1/20/2017' '1/21/2017' '1/22/2017' '1/23/2017' '1/24/2017' '1/25/2017'
 '1/26/2017' '1/27/2017' '1/28/2017' '1/29/2017' '1/3/2017' '1/30/2017'
 '1/31/2017' '1/4/2017' '1/5/2017' '1/6/2017' '1/7/2017' '1/8/2017'
 '1/9/2017' '10/1/2016' '10/10/2016' '10/11/2016' '10/12/2016'
 '10/13/2016' '10/14/2016' '10/15/2016' '10/16/2016' '10/17/2016'
 '10/18/2016' '10/19/2016' '10/2/2016' '10/20/2016' '10/21/2016'
 '10/22/2016' '10/23/2016' '10/24/2016' '10/25/2016' '10/26/2016'
 '10/27/2016' '10/29/2016' '10/3/2016' '10/30/2016' '10/31/2016'
 '10/4/2016' '10/5/2016' '10/6/2016' '10/7/2016' '10/8/2016' '10/9/2016'
 '11/1/2016' '11/10/2016' '11/11/2016' '11/12/2016' '11/13/2016'
 '11/14/2016' '11/15/2016' '11/16/2016' '11/17/2016' '11/18/2016'
 '11/19/2016' '11/2/2016' '11/20/2016' '11/21/2016' '11/22/2016'
 '11/23/2016' '11/24/2016' '11/25/2016' '11/26/2016' '11/27/2016'
 '11/28/2016' '11/29/2016' '11/3/2016' '11/30/2016' '11/4/2016'
 '11/5/2016' '11/6/2016' '11/7/2016' '11/8/2016' '11/9/2016' '12/1/2016'
 '12/10/2016' '12/11/2016' '12/12/2016' '12/13/2016' '12/14/2016'
 '12/15/2016' '12/16/2016' '12/17/2016' '12/18/2016' '12/19/2016'
 '12/2/2016' '12/20/2016' '12/21/2016' '12/22/2016' '12/23/2016'
 '12/24/2016' '12/25/2016' '12/26/2016' '12/27/2016' '12/28/2016'
 '12/29/2016' '12/3/2016' '12/30/2016' '12/31/2016' '12/4/2016'
 '12/5/2016' '12/6/2016' '12/7/2016' '12/8/2016' '12/9/2016' '2/1/2017'
 '2/10/2017' '2/11/2017' '2/12/2017' '2/13/2017' '2/14/2017' '2/15/2017'
 '2/16/2017' '2/17/2017' '2/18/2017' '2/19/2017' '2/2/2017' '2/20/2017'
 '2/21/2017' '2/22/2017' '2/23/2017' '2/24/2017' '2/25/2017' '2/26/2017'
 '2/27/2017' '2/28/2017' '2/3/2017' '2/4/2017' '2/5/2017' '2/6/2017'
 '2/7/2017' '2/8/2017' '2/9/2017' '3/1/2017' '3/10/2017' '3/11/2017'
 '3/12/2017' '3/13/2017' '3/14/2017' '3/15/2017' '3/16/2017' '3/17/2017'
 '3/18/2017' '3/19/2017' '3/2/2017' '3/20/2017' '3/21/2017' '3/22/2017'
 '3/23/2017' '3/24/2017' '3/25/2017' '3/26/2017' '3/27/2017' '3/28/2017'
 '3/29/2017' '3/3/2017' '3/30/2017' '3/31/2017' '3/4/2017' '3/5/2017'
 '3/6/2017' '3/7/2017' '3/8/2017' '3/9/2017' '4/1/2017' '4/10/2017'
 '4/11/2017' '4/12/2017' '4/13/2017' '4/14/2017' '4/15/2017' '4/16/2017'


 '4/17/2017' '4/18/2017' '4/19/2017' '4/2/2017' '4/20/2017' '4/21/2017'
 '4/22/2017' '4/23/2017' '4/24/2017' '4/25/2017' '4/26/2017' '4/27/2017'
 '4/28/2017' '4/29/2017' '4/3/2017' '4/30/2017' '4/4/2017' '4/5/2017'
 '4/6/2017' '4/7/2017' '4/8/2017' '4/9/2017' '5/1/2017' '5/10/2017'
 '5/11/2017' '5/12/2017' '5/13/2017' '5/14/2017' '5/15/2017' '5/16/2017'
 '5/17/2017' '5/18/2017' '5/19/2017' '5/2/2017' '5/20/2017' '5/21/2017'
 '5/22/2017' '5/23/2017' '5/24/2017' '5/25/2017' '5/3/2017' '5/4/2017'
 '5/5/2017' '5/6/2017' '5/7/2017' '5/8/2017' '5/9/2017' '6/1/2017'
 '6/10/2017' '6/11/2017' '6/12/2017' '6/13/2017' '6/14/2017' '6/15/2017'
 '6/16/2017' '6/17/2017' '6/18/2017' '6/19/2017' '6/2/2017' '6/20/2017'
 '6/21/2017' '6/22/2017' '6/23/2017' '6/24/2017' '6/25/2017' '6/26/2017'
 '6/27/2017' '6/28/2017' '6/29/2017' '6/3/2017' '6/30/2017' '6/4/2017'
 '6/5/2017' '6/6/2017' '6/7/2017' '6/8/2017' '6/9/2017' '7/1/2016'
 '7/10/2016' '7/11/2016' '7/12/2016' '7/13/2016' '7/14/2016' '7/15/2016'
 '7/16/2016' '7/17/2016' '7/18/2016' '7/19/2016' '7/2/2016' '7/20/2016'
 '7/21/2016' '7/22/2016' '7/23/2016' '7/24/2016' '7/25/2016' '7/26/2016'
 '7/27/2016' '7/28/2016' '7/29/2016' '7/3/2016' '7/30/2016' '7/31/2016'
 '7/4/2016' '7/5/2016' '7/6/2016' '7/7/2016' '7/8/2016' '7/9/2016'
 '8/1/2016' '8/10/2016' '8/11/2016' '8/12/2016' '8/13/2016' '8/14/2016'
 '8/15/2016' '8/16/2016' '8/17/2016' '8/18/2016' '8/19/2016' '8/2/2016'
 '8/20/2016' '8/21/2016' '8/22/2016' '8/23/2016' '8/24/2016' '8/25/2016'
 '8/26/2016' '8/27/2016' '8/28/2016' '8/29/2016' '8/3/2016' '8/30/2016'
 '8/31/2016' '8/4/2016' '8/5/2016' '8/6/2016' '8/7/2016' '8/8/2016'
 '8/9/2016' '9/1/2016' '9/10/2016' '9/11/2016' '9/12/2016' '9/13/2016'
 '9/14/2016' '9/15/2016' '9/16/2016' '9/17/2016' '9/18/2016' '9/19/2016'
 '9/2/2016' '9/20/2016' '9/21/2016' '9/22/2016' '9/23/2016' '9/24/2016'
 '9/25/2016' '9/26/2016' '9/27/2016' '9/28/2016' '9/29/2016' '9/3/2016'
 '9/30/2016' '9/4/2016' '9/5/2016' '9/6/2016' '9/7/2016' '9/8/2016'
 '9/9/2016']

Weather conditions in Dataset
['Clear' nan 'Mostly Cloudy' 'Overcast' 'Partly Cloudy' 'Scattered Clouds'
 'Light Rain' 'Rain' 'Haze' 'Heavy Rain' 'Light Thunderstorms and Rain'
 'Light Drizzle' 'Fog' 'Patches of Fog' 'Mist'
 'Heavy Thunderstorms and Rain' 'Thunderstorm' 'Shallow Fog' 'Light Snow'
 'Light Rain Showers' 'Drizzle' 'Smoke' 'Thunderstorms and Rain'
 'Funnel Cloud' 'Light Freezing Fog' 'Light Freezing Rain' 'Snow'
 'Light Freezing Drizzle' 'Blowing Snow' 'Light Thunderstorms and Snow'
 'Heavy Snow' 'Low Drifting Snow' 'Light Ice Pellets' 'Ice Pellets'
 'Rain Showers' 'Heavy Drizzle' 'Squalls' 'Light Fog' 'Sand' 'Snow Grains'
 'Widespread Dust' 'Snow Showers' 'Heavy Thunderstorms and Snow'
 'Volcanic Ash' 'Heavy Ice Pellets' 'Heavy Freezing Rain' 'Small Hail'
 'Heavy Rain Showers' 'Blowing Sand' 'Hail' 'Light Haze']
```

In [8]:
```
#Look at accidents by state
accidents_by_state = accident_df["State"].value_counts()
accidents_by_state
```

Out[8]:
```
CA    132185
TX     74796
FL     46349
PA     29872
MI     24276
NY     24059
GA     17299
IL     16453
WA     10710
MD      9836
OH      9045
NJ      8267
MA      7724
VA      7627
NE      6426
CT      3923
SC      3825
IA      1932
MO      1849
RI      1509
DC       574
NH       145
DE        99
WI        68
OR        62
IN        41
WV        40
KY         3
AL         3
Name: State, dtype: int64
```

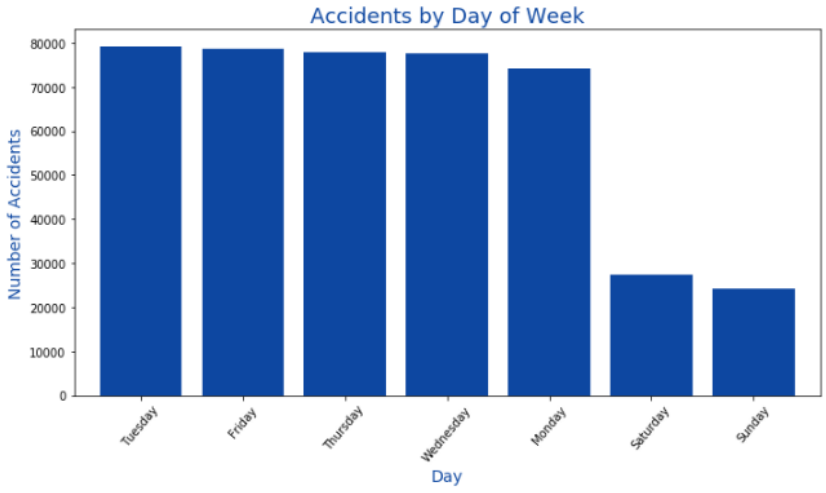
In [9]:
```
#Chart accidents by state
chart_accidents_by_state = accidents_by_state.plot(kind='bar', rot=50, figsize = (10, 6), width = .8, color = "#0D47A1")
plt.title("Accidents by State", size =18, color="#0D47A1")
plt.xlabel("State",size =14, color="#0D47A1")
plt.ylabel("Number of Accidents",size =14, color="#0D47A1")
plt.tight_layout()
plt.savefig("Images/state.png")
plt.show()
```



In [10]:
```
#Look at accidents by day of week
accident_day = accident_df["Day"].value_counts()
accident_day
```

Out[10]:
```
Tuesday      79108
Friday       78590
Thursday     77853
Wednesday    77675
Monday       74167
Saturday     27494
Sunday       24110
Name: Day, dtype: int64
```

In [11]:
```
#Chart accidents by day of week
chart_accident_day = accident_day.plot(kind='bar', rot=50, figsize = (10, 6), width = .8, color = "#0D47A1")
plt.title("Accidents by Day of Week", size =18, color="#0D47A1")
plt.xlabel("Day",size =14, color="#0D47A1")
plt.ylabel("Number of Accidents",size =14, color="#0D47A1")
plt.tight_layout()
plt.savefig("Images/day.png")
plt.show()
```



In [12]:
```
#Look at accidents by date
accident_date = accident_df["Date"].value_counts()
accident_date.head(10)
```

Out[12]:
```
11/30/2016    2138
12/16/2016    2110
11/29/2016    2081
11/15/2016    2078
11/23/2016    2064
11/21/2016    2054
11/22/2016    2040
11/14/2016    2026
11/10/2016    2024
11/16/2016    2020
Name: Date, dtype: int64
```

In [13]:
```
#Look at accidents by month
accident_month = accident_df["Month"].value_counts()
accident_month.head(10)
```

Out[13]:
```
November     46540
December     40655
September    39565
March        39465
January      38719
August       38541
October      38178
February     35374
April        33645
June         32128
Name: Month, dtype: int64
```

In [14]:

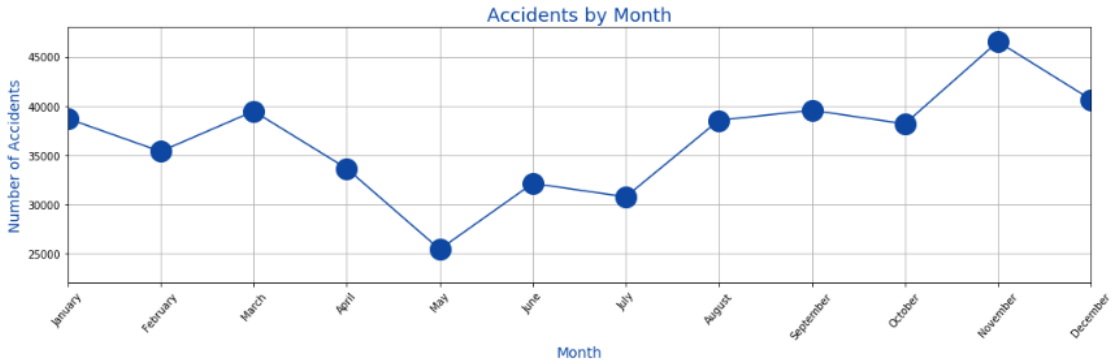
```
#Set up for axis in scatter
month = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October",
         "November", "December"]
accidents = [38719, 35374, 39465, 33645, 25427, 32128, 30760, 38541, 39565, 38178, 46540, 40655]
accidents_by_month_df = pd.DataFrame({"Month": month, "Number of Accidents": accidents})
accidents_by_month_df

```

Out[14]:

	Month	Number of Accidents
0	January	38719
1	February	35374
2	March	39465
3	April	33645
4	May	25427
5	June	32128
6	July	30760
7	August	38541
8	September	39565
9	October	38178
10	November	46540
11	December	40655

In [15]:

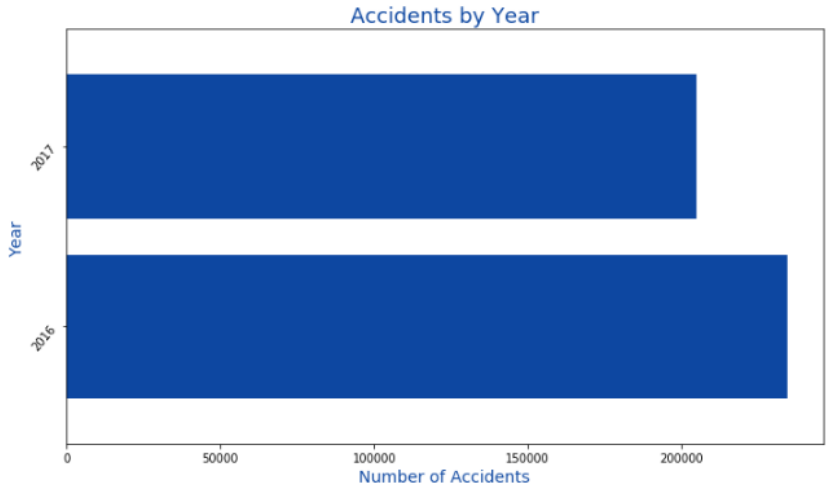
```
#Create scatter plot for accidents by month
accidents_by_month_df.plot(rot=50, figsize=(15,5), marker="o", colors="#0D47A1",
                           markersize=20, legend = False)
plt.ylim(22000, 48000)
labels = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October",
          "November", "December"]
ticks = np.arange(0,len(labels),1)
plt.xticks(ticks, labels)
plt.title("Accidents by Month", size =18, color="#0D47A1")
plt.xlabel("Month", size =14, color="#0D47A1")
plt.ylabel("Number of Accidents", size =14, color="#0D47A1")
plt.grid(True)
plt.tight_layout()

plt.savefig("Images/month.png")
plt.show()

```



In [16]:
```
#Look at accidents by year
accident_year = accident_df["Year"].value_counts()
accident_year
```

Out[16]:
```
2016    234239
2017    204758
Name: Year, dtype: int64
```

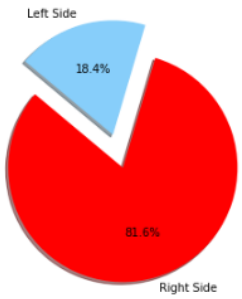
In [17]:
```
#Create horizontal bar chart for accidents by year
accident_year = accident_year.plot(kind='barh', rot=50, figsize = (10, 6), width = .8, color = "#0D47A1")
plt.title("Accidents by Year", size =18, color="#0D47A1")
plt.xlabel("Number of Accidents",size =14, color="#0D47A1")
plt.ylabel("Year",size =14, color="#0D47A1")
plt.tight_layout()
plt.savefig("Images/year.png")
plt.show()
```



In [18]:
```
#Distribution plot of accidents occuring on the left/right side of the road
road_side_count = accident_df["Side"].value_counts()
road_side_count
```

Out[18]:
```
R    358010
L     80986
         1
Name: Side, dtype: int64
```

In [19]:
```
#Create pie chart for accidents by side of road
# Labels for the sections of our pie chart
labels = ["Right Side", "Left Side"]

# The values of each section of the pie chart
sizes = [358017, 80987]

# The colors of each section of the pie chart
colors = ["red", "lightskyblue"]

# plot to show left and right side accidents
explode = (0.3, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct="%1.1f%%", shadow=True, startangle=140)
plt.savefig("Images/road_side.png")
plt.tight_layout()
```



In [20]:

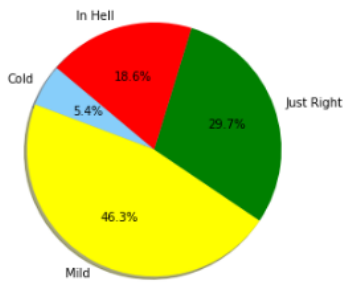
```
#Get number of accidents at temperatur points
temperature = accident_df["Temperature(F)"].value_counts()
temperature_count_df = pd.DataFrame({"Number of Accidents":temperature})
temperature_count_df = temperature_count_df.sort_values("Number of Accidents", ascending = False)
temperature_count_df.head(20)
```

Out[20]:

	Number of Accidents
77.0	11158
68.0	10963
59.0	10119
73.0	7452
73.9	7406
63.0	7328
70.0	7129
72.0	7114
66.9	7106
62.1	7093
64.9	7073
75.0	7007
75.9	6996
50.0	6950
64.0	6917
66.0	6897
60.1	6881
71.1	6856
69.1	6838
61.0	6834

In [21]:

```
#Binned in Excel to capture datapoints
#Data for points below came from an excel pivot chart
# Labels for the sections of our pie chart
labels = ["Cold", "Mild", "Just Right", "In Hell"]

# The values of each section of the pie chart
sizes = [22962, 198693, 127533, 79934]

# The colors of each section of the pie chart
colors = ["lightskyblue", "yellow", "green", "red"]

# plot to show accidents and temp

plt.pie(sizes, labels=labels, colors=colors,
        autopct="%1.1f%%", shadow=True, startangle=140)
plt.savefig("Images/temp_pie.png")
plt.tight_layout()
```



In [22]:

```
#Get number of accidents at humidity points
humidity = accident_df["Humidity(%)"].value_counts()
humidity_count_df = pd.DataFrame({"Number of Accidents":humidity})
humidity_count_df = humidity_count_df.sort_values("Number of Accidents", ascending = False)
humidity_count_df.head(20)
```

Out[22]:

	Number of Accidents
100.0	15999
93.0	14516
87.0	10071
90.0	9362
78.0	8326
84.0	7610
81.0	7448
89.0	7360
94.0	7316
83.0	7290
65.0	7231
73.0	7174
61.0	6907
52.0	6859
72.0	6781
96.0	6777
63.0	6738
68.0	6732
59.0	6691
70.0	6674

In [23]:

```
#Distribution plot of accidents occuring on the left/right side of the road
street_count = accident_df["Street"].value_counts()
street_count
```

Out[23]:

```
I-5 N               5179
I-95 N              4275
I-405 N             3336
I-10 E              3116
US-101 N            3105
                    ...
S Ridgeway Ave         1
Fairacres Rd           1
S 104th Ave            1
Laguna Beach Cir       1
Aintree Rd             1
Name: Street, Length: 42824, dtype: int64
```