Data Visualization regarding the level of interest in the given topics

In [1]:

```
#Importing the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:

```
#Reading the excel file and converting it to a DataFrame
df= pd.read_csv('Topic_Survey_Assignment (1).csv',index_col= 'Unnamed: 0')
```

In [3]:

```
#Checking the the first 5 rows of the DataFrame
df.head()
```

Out[3]:

	Very interested	Somewhat interested	Not interested
Big Data (Spark / Hadoop)	1332	729	127
Data Analysis / Statistics	1688	444	60
Data Journalism	429	1081	610
Data Visualization	1340	734	102
Deep Learning	1263	770	136

In [4]:

```
#Checking the DataFrame details with the the ".info()" command
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6 entries, Big Data (Spark / Hadoop) to Machine Learning
Data columns (total 3 columns):
Very interested        6 non-null int64
Somewhat interested    6 non-null int64
Not interested         6 non-null int64
dtypes: int64(3)
memory usage: 192.0+ bytes
```

In [5]:

```
#Sorting the DataFrame
df.sort_values(by='Very interested',ascending=False)
```

Out[5]:

	Very interested	Somewhat interested	Not interested
Data Analysis / Statistics	1688	444	60
Machine Learning	1629	477	74
Data Visualization	1340	734	102
Big Data (Spark / Hadoop)	1332	729	127
Deep Learning	1263	770	136
Data Journalism	429	1081	610

In [6]:

```
#Converting the Data into percentage and ronding the figures upto 2 decimal values with ".round()" function
df1=df.apply(lambda x:x*(100/2233) ).round(2)          #total=2233
df1.sort_values(by='Very interested',ascending=False)
```

Out[6]:

	Very interested	Somewhat interested	Not interested
Data Analysis / Statistics	75.59	19.88	2.69
Machine Learning	72.95	21.36	3.31
Data Visualization	60.01	32.87	4.57
Big Data (Spark / Hadoop)	59.65	32.65	5.69
Deep Learning	56.56	34.48	6.09
Data Journalism	19.21	48.41	27.32

In [7]:

```
#df['Very interested']*(100/2233)
```

In [8]:

```
#df['Somewhat interested']*(100/2233)
```

In [9]:

```
#df['Not interested']*(100/2233)
```

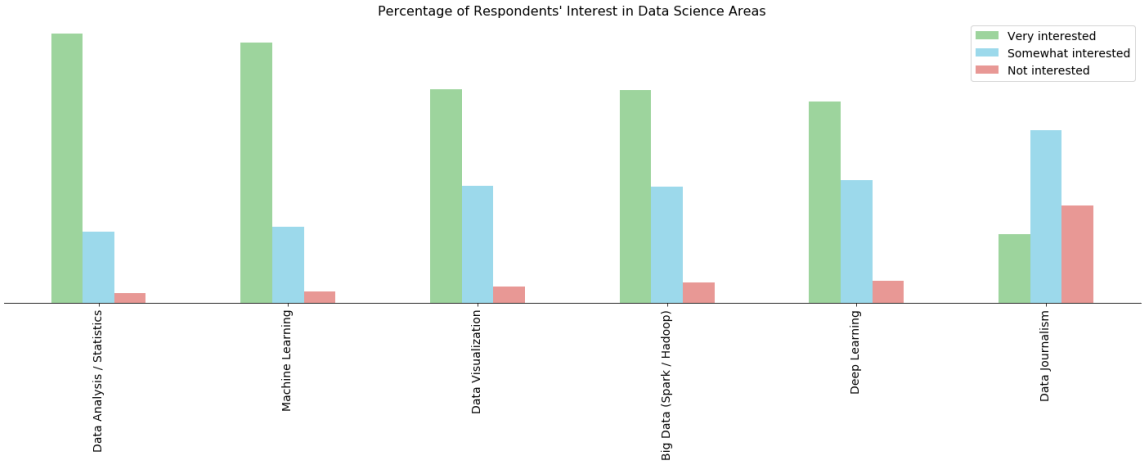
Plotting the Bar Graph using the Artist Layer

In [10]:

```
# Plotted the bar graph keeping the following in mind:
    #1.sort the DataFrame and plot.
    #2.set the figsize=(20,8), barwidth=0.8.
    #3.color list as specified.
    #4.fontsize of the legends and labels. Also, set the legend location.
    #5.Set title and its fontsize as specified.
    #6.Removed the borders and ticks using seaborn library


ax=df1.sort_values(by='Very interested', ascending=False).plot(kind='bar', figsize=(20,
8),subplots=False,
                                                               fontsize=14,color=('#5cb85
c','#5bc0de','#d9534f'),alpha=0.6)
bar_width=0.8;

# Adding the values to the bars

#for p in ax.patches:
#    width, height = p.get_width(), p.get_height()
#    x, y = p.get_xy()
#    ax.annotate('{:.0%}'.format(height), (x, y + height + 0.01))
#plt.annotate(df1['Very interested'])

    #Setting title and legend

ax.set_title("Percentage of Respondents' Interest in Data Science Areas",fontsize=16)
ax.legend(loc='upper right',fontsize=14)
plt.tight_layout()

    #Removing the top, left and right borders and ticks

sns.despine()                   #Removes top and right border
sns.despine(left=True)          #Removes left border
plt.yticks([])                  #Removes yticks
```

Out[10]:

```
([], <a list of 0 Text yticklabel objects>)
```



In [ ]: