In [1]:
```
medical_charges_url = 'https://raw.githubusercontent.com/JovianML/opendatasets
```

In [2]:
```
from urllib.request import urlretrieve
```

In [3]:
```
urlretrieve(medical_charges_url, 'medical.csv')
```

Out[3]:
```
('medical.csv', <http.client.HTTPMessage at 0x17fffddd0a0>)
```

In [4]:
```
import pandas as pd
import plotly.express as px
import matplotlib
import matplotlib.pyplot as plt
```

In [5]:
```
df=pd.read_csv('medical.csv')
df
```

Out[5]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [6]:
```
df.describe()
```

Out[6]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [8]:
```
non_smoker=df[df['smoker']=='no']
smoker=df[df['smoker']=='yes']

non_smoker
```

Out[8]:

	age	sex	bmi	children	smoker	region	charges
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
5	31	female	25.740	0	no	southeast	3756.62160
...	...	...	...	...	...	...	...
1332	52	female	44.700	3	no	southwest	11411.68500
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500

1064 rows × 7 columns

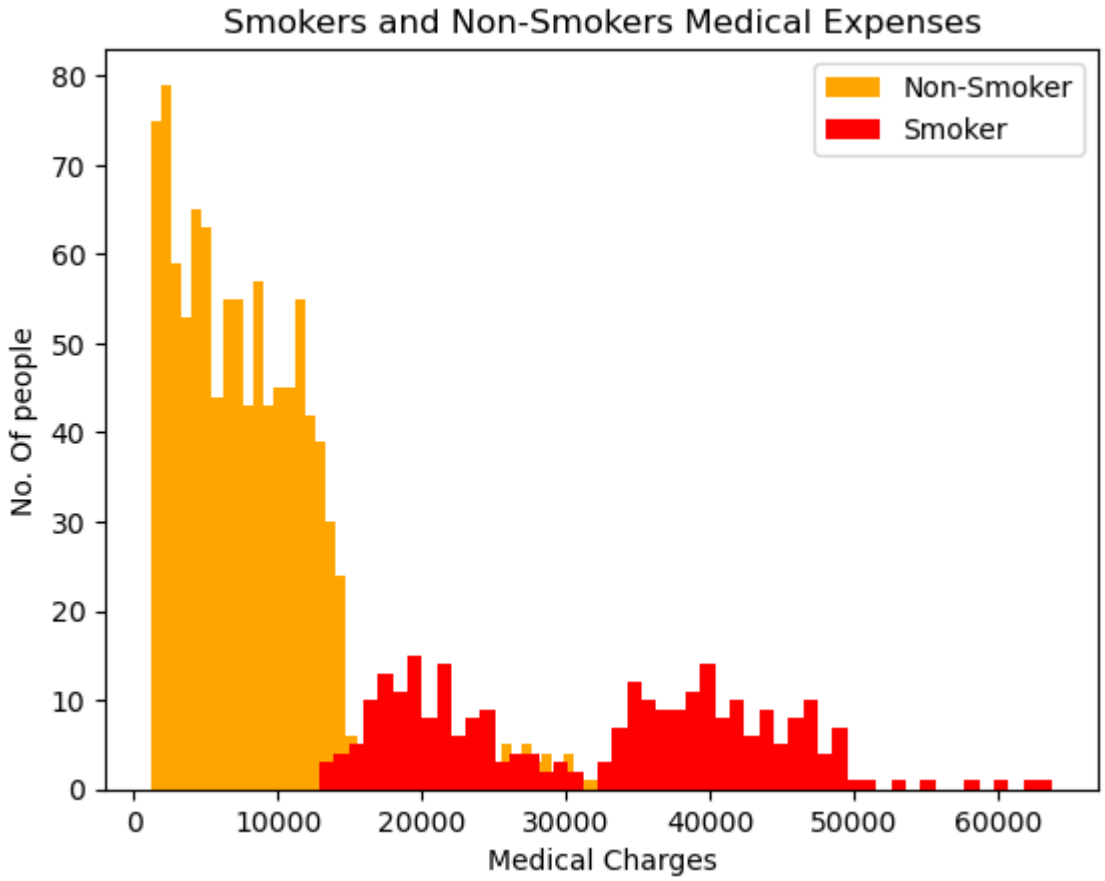
Showing how medical expenses varies for Smokers and Non Smokers

In [9]:
```
plt.hist(non_smoker['charges'],bins=50,color='orange')
plt.hist(smoker['charges'],bins=50,color='red')



plt.title('Smokers and Non-Smokers Medical Expenses')
plt.xlabel('Medical Charges')
plt.ylabel('No. Of people')
plt.legend(labels=['Non-Smoker', 'Smoker'])


plt.show()
```



In [10]:
```
df
```

Out[10]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

No. of Smokers as per age range

In [11]:
```
smoker_age = df[['age','smoker']].copy()
smoker_age
```

Out[11]:

	age	smoker
0	19	yes
1	18	no
2	28	no
3	33	no
4	32	no
...	...	...
1333	50	no
1334	18	no
1335	18	no
1336	21	no
1337	61	yes

1338 rows × 2 columns

In [12]:

```
#create an age range for certain age group
smoker_age_df1=smoker_age[(smoker_age['age']<=20) & (smoker_age['smoker']=='ye
smoker_age_df1['age_group']='young'


non_smoker_age_df1=smoker_age[(smoker_age['age']<=20) & (smoker_age['smoker']=
non_smoker_age_df1['age_group']='young'




smoker_age_df2=smoker_age[((smoker_age['age']>=21) & (smoker_age['age']<=50))
smoker_age_df2['age_group']='adult'


non_smoker_age_df2=smoker_age[((smoker_age['age']>=21) & (smoker_age['age']<=5
non_smoker_age_df2['age_group']='adult'




smoker_age_df3=smoker_age[(smoker_age['age']>=51) & (smoker_age['smoker']=='ye
smoker_age_df3['age_group']='old'



non_smoker_age_df3=smoker_age[(smoker_age['age']>=51) & (smoker_age['smoker']=
non_smoker_age_df3['age_group']='old'



smoker_age_df = pd.concat([smoker_age_df1, smoker_age_df2,smoker_age_df3], ign
non_smoker_df=pd.concat([non_smoker_age_df1, non_smoker_age_df2, non_smoker_ag
```

```
C:\Users\saleh\AppData\Local\Temp\ipykernel_22696\3095028681.py:3: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  smoker_age_df1['age_group']='young'
C:\Users\saleh\AppData\Local\Temp\ipykernel_22696\3095028681.py:7: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  non_smoker_age_df1['age_group']='young'
C:\Users\saleh\AppData\Local\Temp\ipykernel_22696\3095028681.py:14: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  smoker_age_df2['age_group']='adult'
C:\Users\saleh\AppData\Local\Temp\ipykernel_22696\3095028681.py:18: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  non_smoker_age_df2['age_group']='adult'
C:\Users\saleh\AppData\Local\Temp\ipykernel_22696\3095028681.py:24: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  smoker_age_df3['age_group']='old'
C:\Users\saleh\AppData\Local\Temp\ipykernel_22696\3095028681.py:30: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
```

```
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  non_smoker_age_df3['age_group']='old'
```

In [13]:
```
smoker_age_df=smoker_age_df[smoker_age_df['smoker']=='yes'].groupby('age_group
non_smoker_df=non_smoker_df[non_smoker_df['smoker']=='no'].groupby('age_group'
```

In [14]:
```
smoker_age_df
non_smoker_df
```

Out[14]:

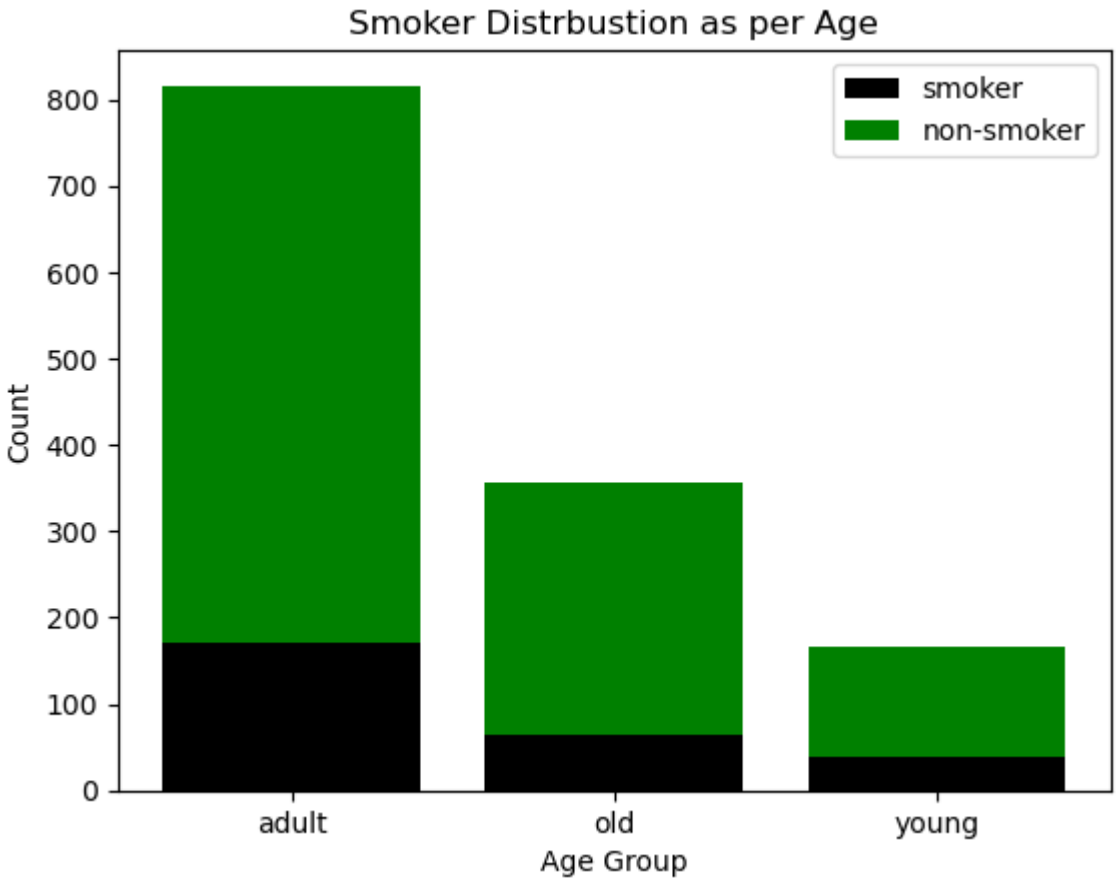
	age_group	count
0	adult	645
1	old	292
2	young	127

In [15]:
```
# Merge the two DataFrames based on the 'AgeGroup' column
merged_df = pd.merge(smoker_age_df, non_smoker_df, on='age_group', suffixes=('

# Plotting a stacked bar chart
plt.bar(merged_df['age_group'], merged_df['count_df1'], label='smoker', color=
plt.bar(merged_df['age_group'], merged_df['count_df2'], label='non-smoker', co

# Add labels and legend
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.title('Smoker Distrbustion as per Age')
plt.legend()

# Show the plot
plt.show()
```



Medical Charges as per Age with Smokers and Non Smokers

In [16]:
```
age_charges = df[['age','smoker','charges']].copy()
age_charges
```

Out[16]:

	age	smoker	charges
0	19	yes	16884.92400
1	18	no	1725.55230
2	28	no	4449.46200
3	33	no	21984.47061
4	32	no	3866.85520
...	...	...	...
1333	50	no	10600.54830
1334	18	no	2205.98080
1335	18	no	1629.83350
1336	21	no	2007.94500
1337	61	yes	29141.36030

1338 rows × 3 columns

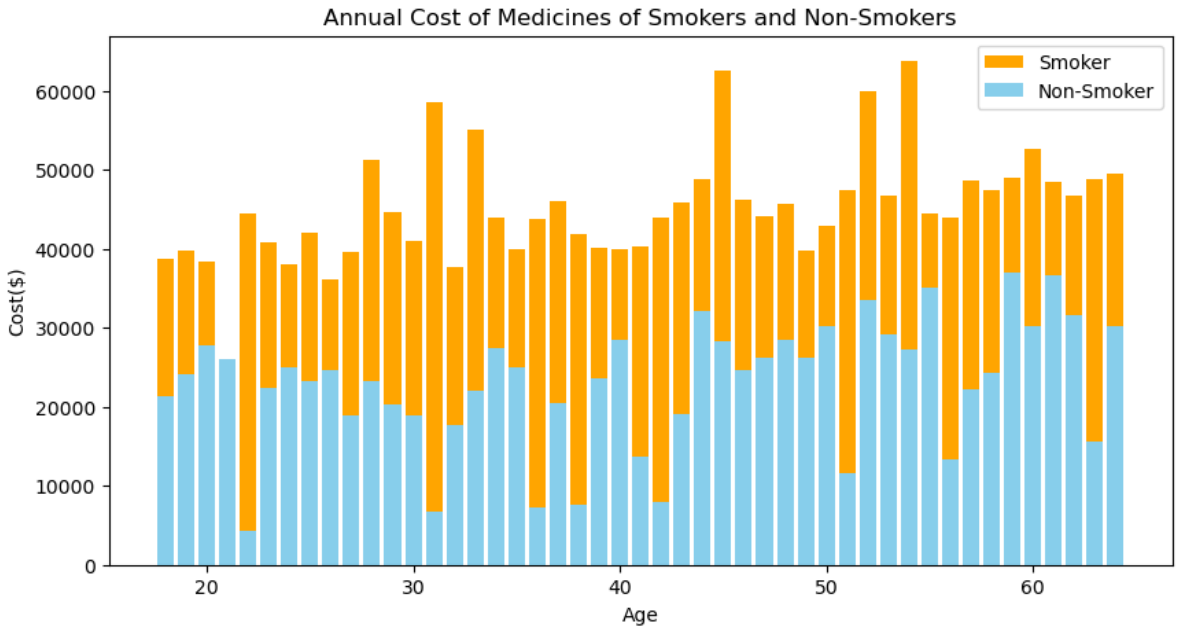
In [17]:
```
# Create a stacked bar chart
fig, ax = plt.subplots(figsize=(10,5))

# Separate data for smokers and non-smokers
smoker_charges = age_charges[age_charges['smoker'] == 'yes']['charges']
non_smoker_charges = age_charges[age_charges['smoker'] == 'no']['charges']

# Plot the bars
bars1 = plt.bar(age_charges[age_charges['smoker'] == 'yes']['age'], smoker_cha
bars2 = plt.bar(age_charges[age_charges['smoker'] == 'no']['age'], non_smoker_

# Add labels and title
plt.xlabel('Age')
plt.ylabel("Cost($)")
plt.title('Annual Cost of Medicines of Smokers and Non-Smokers')
plt.legend()

# Show the plot
plt.show()
```



The total medical costs by location

In [18]:
```
df
```

Out[18]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [19]:
```
area_charge=df[['region','charges']].copy()
area_charge
```

Out[19]:

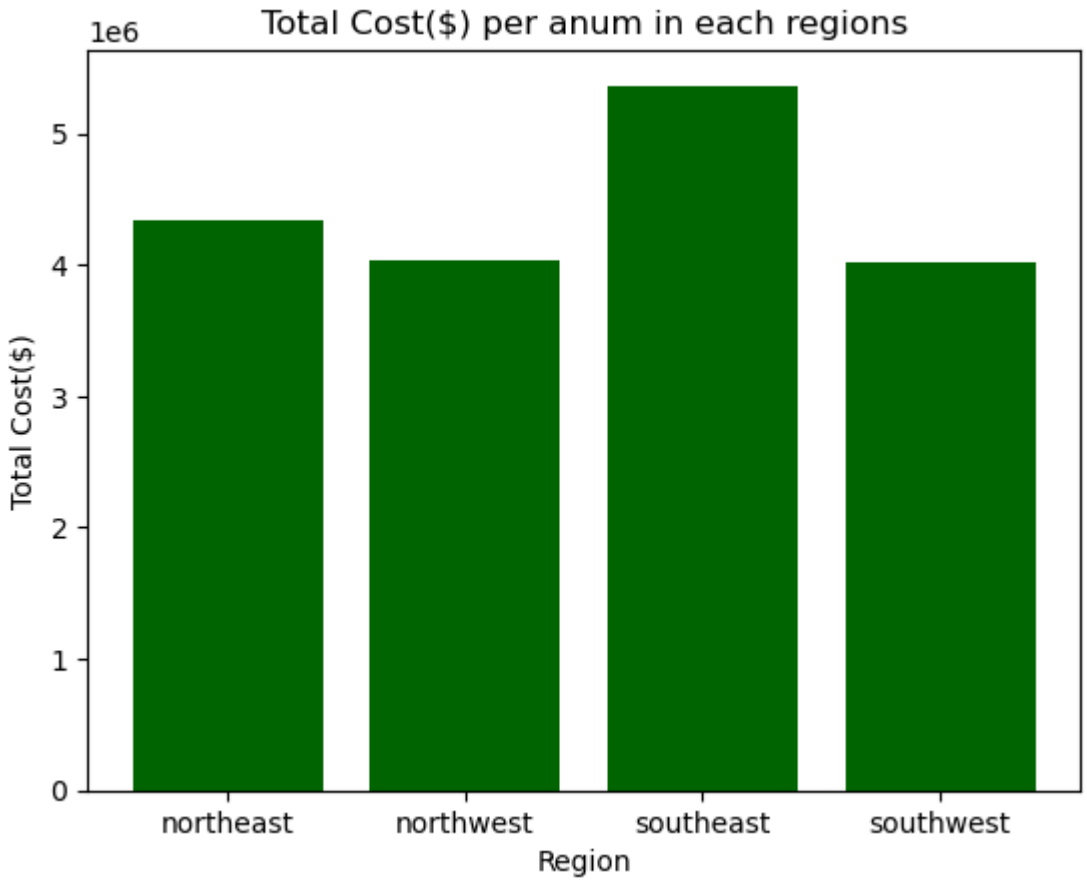
	region	charges
0	southwest	16884.92400
1	southeast	1725.55230
2	southeast	4449.46200
3	northwest	21984.47061
4	northwest	3866.85520
...	...	...
1333	northwest	10600.54830
1334	northeast	2205.98080
1335	southeast	1629.83350
1336	southwest	2007.94500
1337	northwest	29141.36030

1338 rows × 2 columns

In [20]:
```
area_charge=area_charge.groupby('region')['charges'].sum().reset_index()
```

In [21]:
```
plt.bar(area_charge['region'],area_charge.charges,color='darkgreen')
plt.title('Total Cost($) per anum in each regions')
plt.xlabel('Region')
plt.ylabel('Total Cost($)')
```

Out[21]:
```
Text(0, 0.5, 'Total Cost($)')
```



Finding the Smokers per region to see which region has the highest smokers and see if the costs are high because of it?

In [22]:
```
df
```

Out[22]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [23]:
```
asmr=df[['smoker','region','charges']].copy()
asmr
```

Out[23]:

	smoker	region	charges
0	yes	southwest	16884.92400
1	no	southeast	1725.55230
2	no	southeast	4449.46200
3	no	northwest	21984.47061
4	no	northwest	3866.85520
...	...	...	...
1333	no	northwest	10600.54830
1334	no	northeast	2205.98080
1335	no	southeast	1629.83350
1336	no	southwest	2007.94500
1337	yes	northwest	29141.36030

1338 rows × 3 columns

In [24]:
```
asmr_no=asmr[asmr['smoker']=='no'][['charges','region']]
asmr_yes=asmr[asmr['smoker']=='yes'][['charges','region']]
asmr_yes
```

Out[24]:

	charges	region
0	16884.92400	southwest
11	27808.72510	southeast
14	39611.75770	southeast
19	36837.46700	southwest
23	37701.87680	northeast
...	...	...
1313	36397.57600	southwest
1314	18765.87545	northwest
1321	28101.33305	northeast
1323	43896.37630	southeast
1337	29141.36030	northwest

274 rows × 2 columns

In [25]:
```
asmr_yes=asmr_yes.groupby('region')['charges'].sum().reset_index()
asmr_no=asmr_no.groupby('region')['charges'].sum().reset_index()
asmr_no
```

Out[25]:

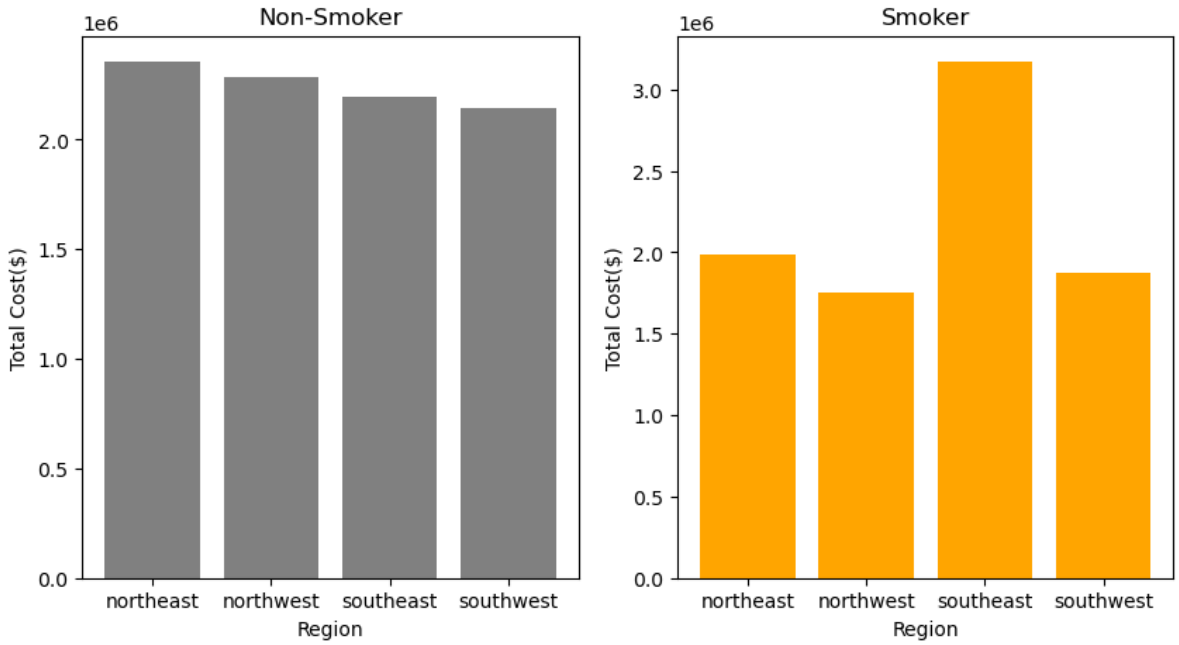
	region	charges
0	northeast	2.355542e+06
1	northwest	2.284576e+06
2	southeast	2.192795e+06
3	southwest	2.141149e+06

In [26]:
```
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(10,5))

ax1.bar(asmr_no['region'],asmr_no['charges'],color='grey',label='non-smoker')
ax2.bar(asmr_yes['region'],asmr_yes['charges'],color='orange',label='smoker')


ax1.set_xlabel('Region')
ax2.set_xlabel('Region')
ax1.set_ylabel('Total Cost($)')
ax2.set_ylabel('Total Cost($)')
ax1.set_title('Non-Smoker')
ax2.set_title('Smoker')


plt.show()
```



In [27]:
```
df
```

Out[27]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

Is smoking related with bmi?

In [28]:
```
#no. of smokers and non-smoker
count_smoker=df[df['smoker']=='yes']
count_non_smoker=df[df['smoker']=='no']
```

In [29]:
```
count_smoker=count_smoker.shape[0]
count_non_smoker=count_non_smoker.shape[0]
```

In [30]:
```
bmis=df[['bmi','smoker']].copy()
bmis
```

Out[30]:

	bmi	smoker
0	27.900	yes
1	33.770	no
2	33.000	no
3	22.705	no
4	28.880	no
...	...	...
1333	30.970	no
1334	31.920	no
1335	36.850	no
1336	25.800	no
1337	29.070	yes

1338 rows × 2 columns

In [31]:
```
# bmis_no=bmis[bmis['smoker']=='no'][['bmi','smoker']]
# bmis_yes=bmis[bmis['smoker']=='yes'][['bmi','smoker']]
```

In [32]:
```
#grouping smoker yes and no in a df
bmis= bmis.groupby('smoker')['bmi'].sum().reset_index()
```

In [33]:
```
bmis.shape
```

Out[33]:
```
(2, 2)
```

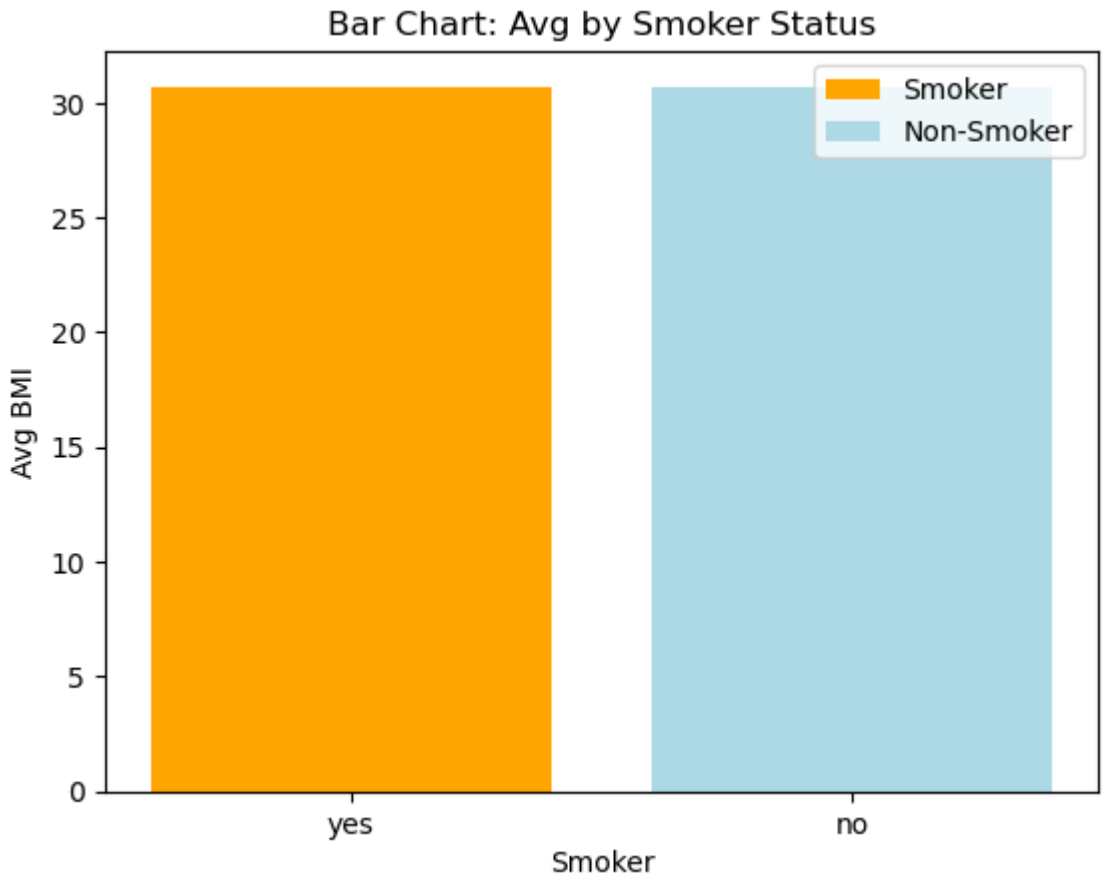
In [34]:
```
# Define colors for each category
colors = {'yes': ['Smoker','orange'], 'no': ['Non-Smoker','lightblue']}

# Create a bar chart with different colors for each category and labels
for category, color in colors.items():
    if category=='yes':
        subset = bmis[bmis['smoker'] == category]
        plt.bar(subset['smoker'], subset['bmi']/count_smoker, color=color[1],
    else:
        subset = bmis[bmis['smoker'] == category]
        plt.bar(subset['smoker'], subset['bmi']/count_non_smoker, color=color[

# Add labels and title
plt.xlabel('Smoker')
plt.ylabel('Avg BMI')
plt.title('Bar Chart: Avg by Smoker Status')

# Add legend
plt.legend()

# Show the plot
plt.show()
```



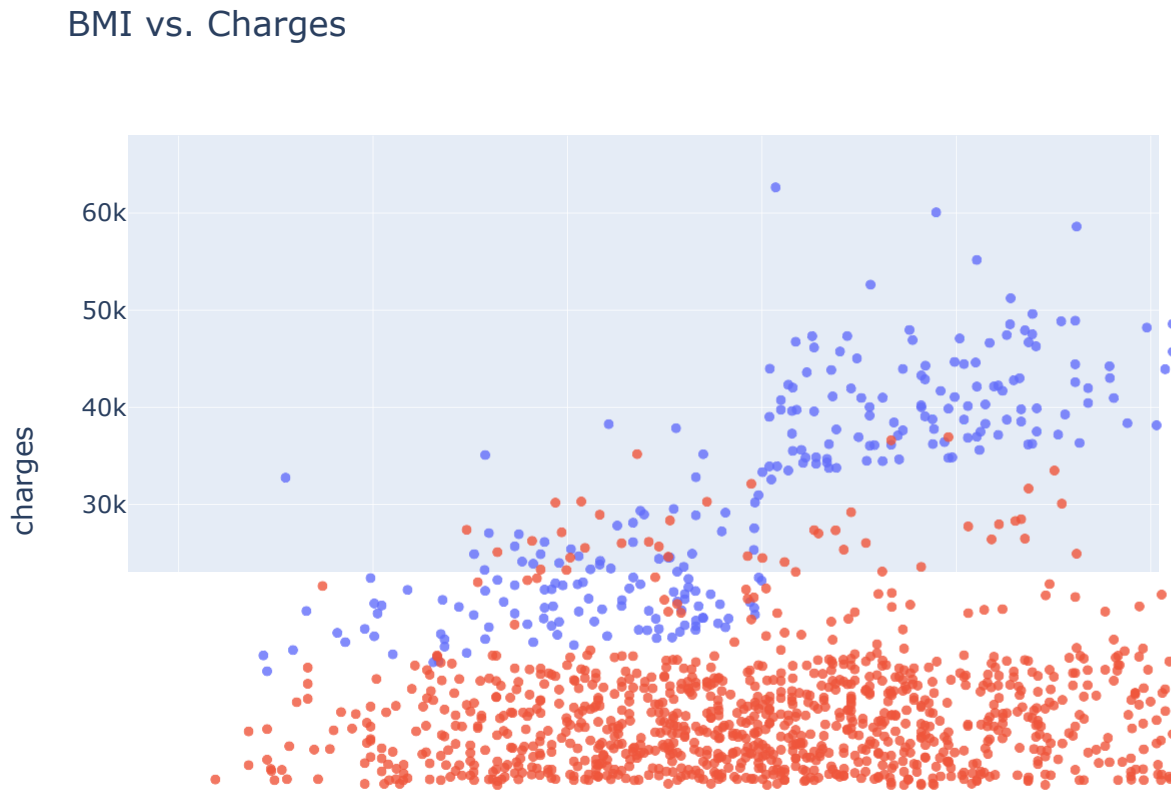
In [35]:
```
df
```

Out[35]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [36]:
```
fig = px.scatter(df,
                 x='bmi',
                 y='charges',
                 color='smoker',
                 opacity=0.8,
                 hover_data=['sex'],
                 title='BMI vs. Charges')
fig.update_traces(marker_size=5)
fig.show()
```



In [ ]:
```
```