

```
In [21]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [4]: data = pd.read_csv('diabetes.csv')
```

```
In [5]: data.head(5)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
In [6]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Pregnancies         768 non-null    int64
1   Glucose             768 non-null    int64
2   BloodPressure       768 non-null    int64
3   SkinThickness       768 non-null    int64
4   Insulin             768 non-null    int64
5   BMI                 768 non-null    float64
6   DiabetesPedigreeFunction 768 non-null    float64
7   Age                 768 non-null    int64
8   Outcome             768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
In [7]: data.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.349958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

```
In [8]: pd.notnull(data).sum()
```

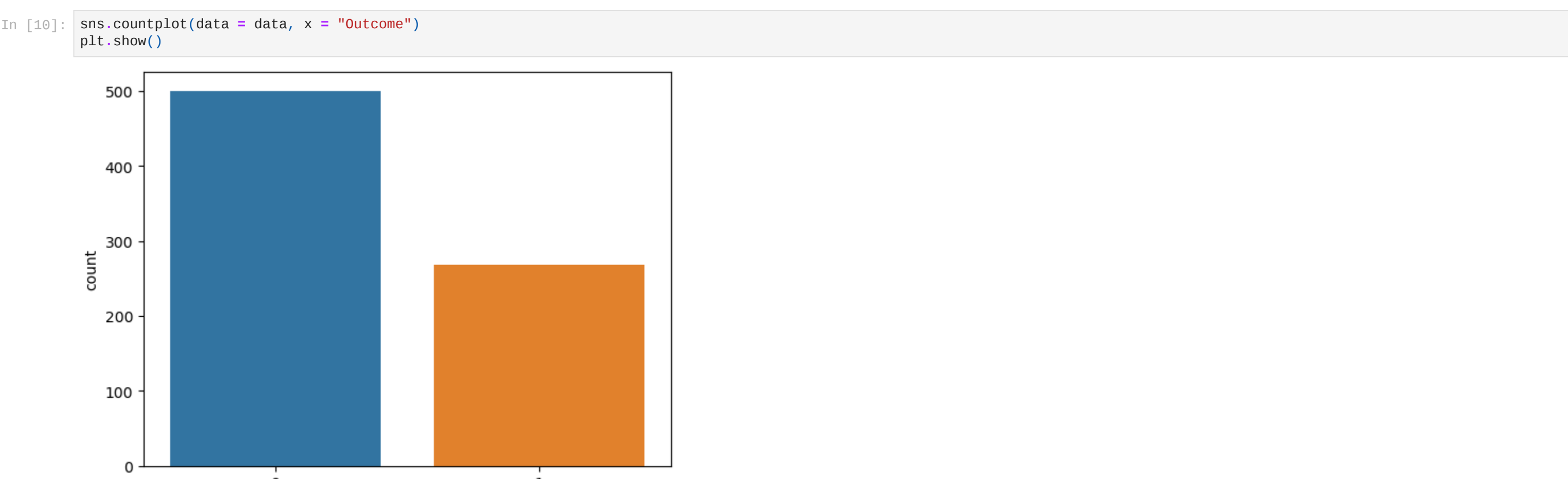
Pregnancies	768
Glucose	768
BloodPressure	768
SkinThickness	768
Insulin	768
BMI	768
DiabetesPedigreeFunction	768
Age	768
Outcome	768
dtype:	int64

```
In [9]: data.rename(columns = {'DiabetesPedigreeFunction':'DPF'})
```

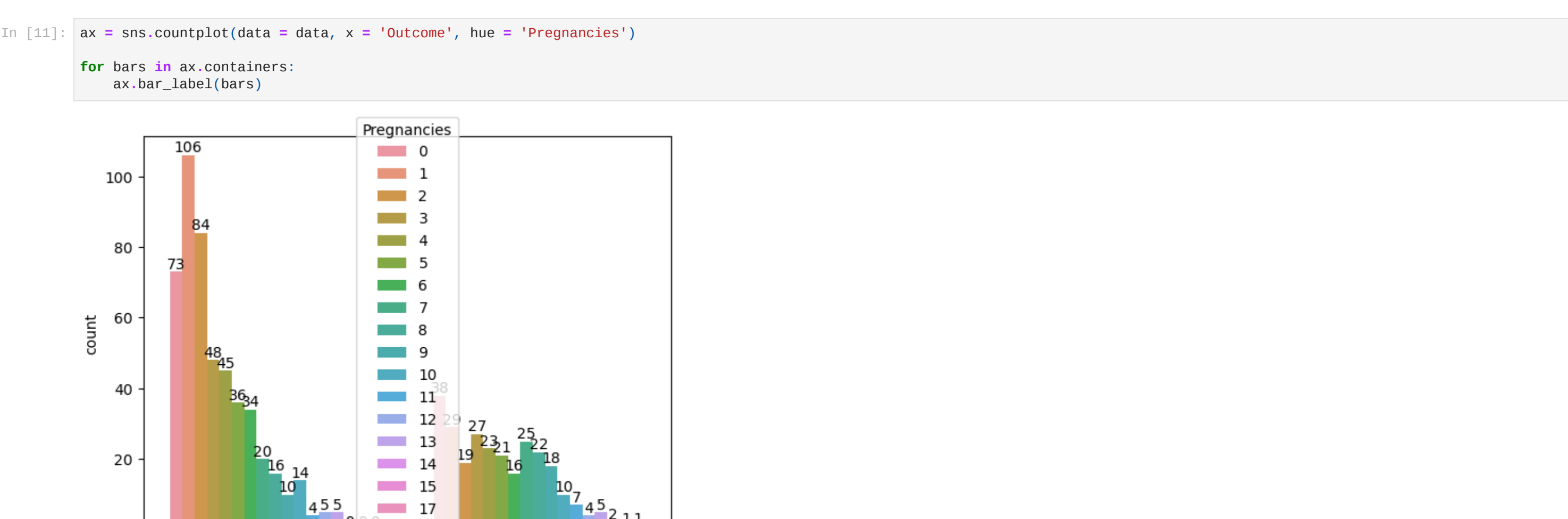
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows x 9 columns

```
In [10]: sns.countplot(data = data, x = "Outcome")
plt.show()
```

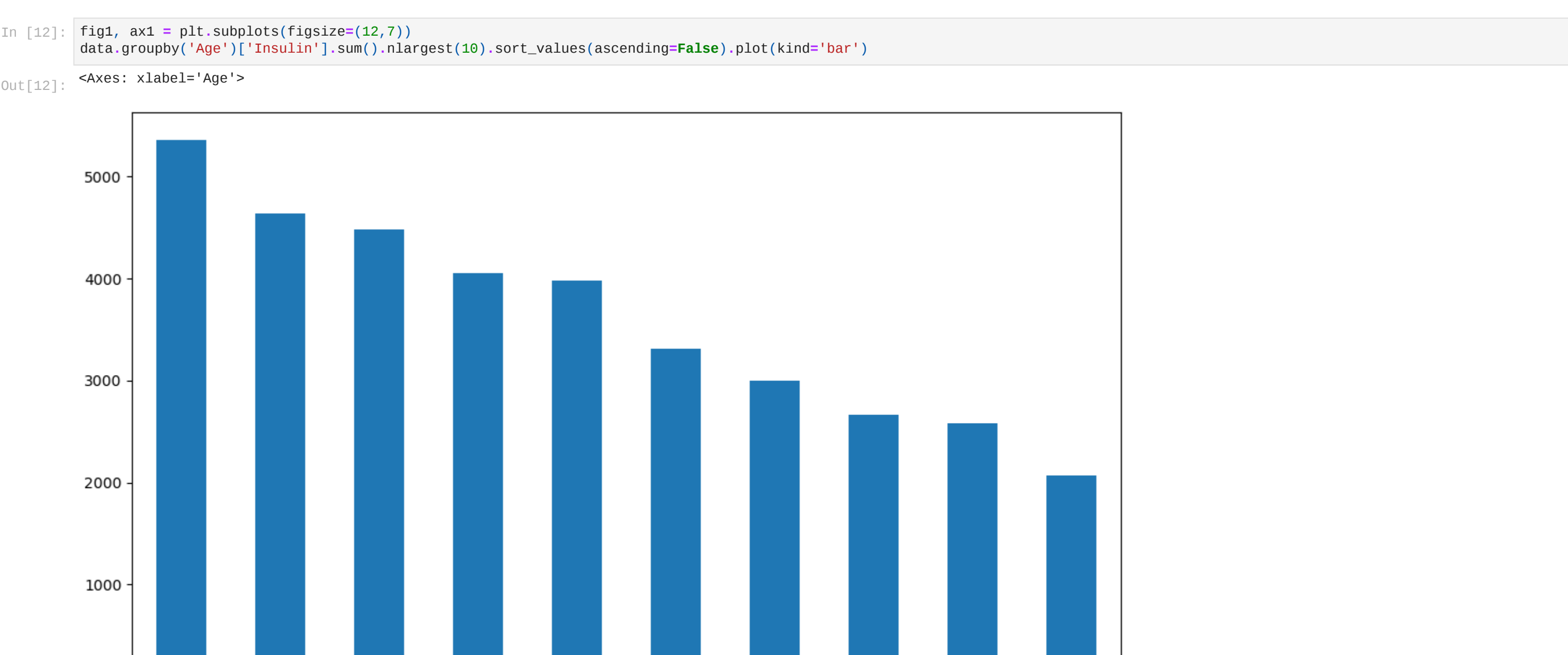


```
In [11]: ax = sns.countplot(data = data, x = 'Outcome', hue = 'Pregnancies')
for bars in ax.containers:
    ax.bar_label(bars)
```



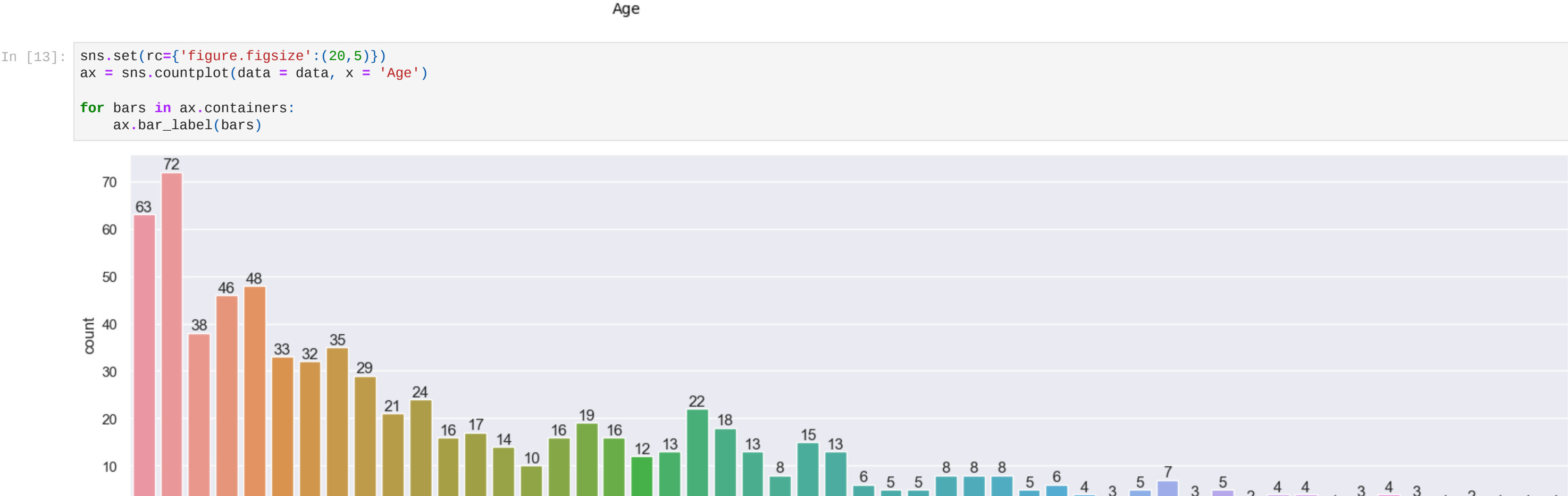
```
In [12]: fig1, ax1 = plt.subplots(figsize=(12,7))
data.groupby('Age')['Insulin'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')
```

```
<Axes: xlabel='Age'>
```



```
In [13]: sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = data, x = 'Age')
```

```
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [14]: sns.set_style('ticks')
sns.set_palette('rainbow')
```

```
g = sns.pairplot(data, hue = 'Outcome', height = 6, aspect = 2)
```

```
g.fig.set_size_inches(13,13)
```

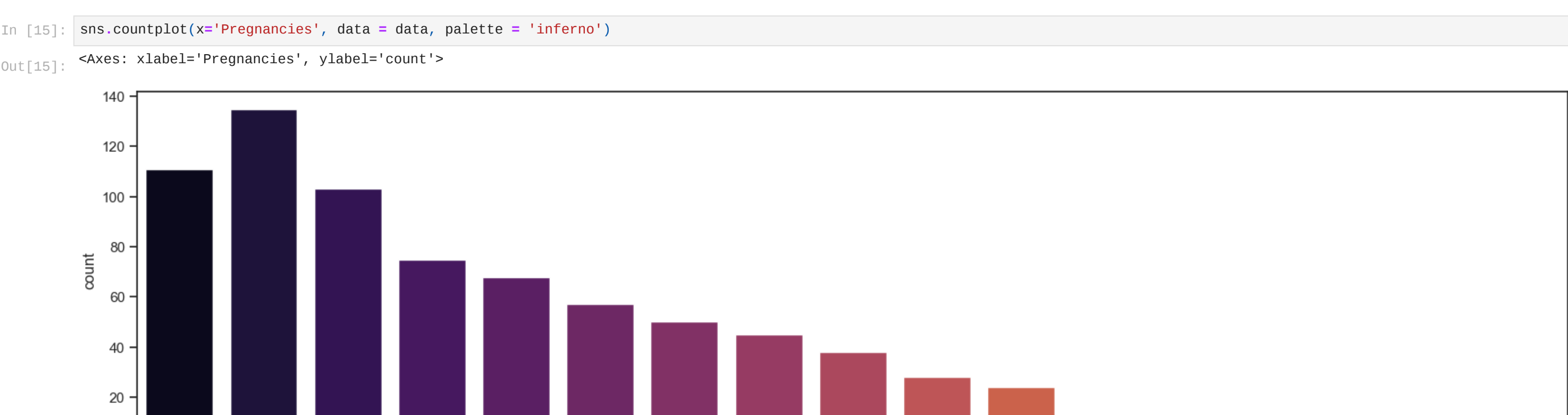
```
C:\ProgramData\Anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
```

```
self.figure.tight_layout(*args, **kwargs)
```



```
In [15]: sns.countplot(x='Pregnancies', data = data, palette = 'inferno')
```

```
<Axes: xlabel='Pregnancies', ylabel='count'>
```



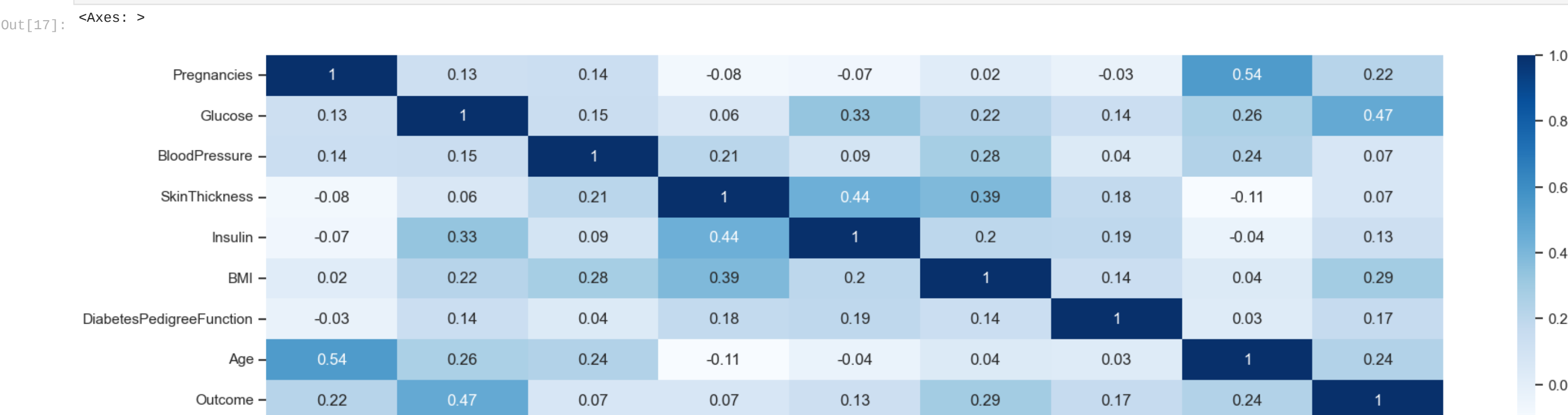
```
In [16]: data.corr()
```

```
Out[16]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

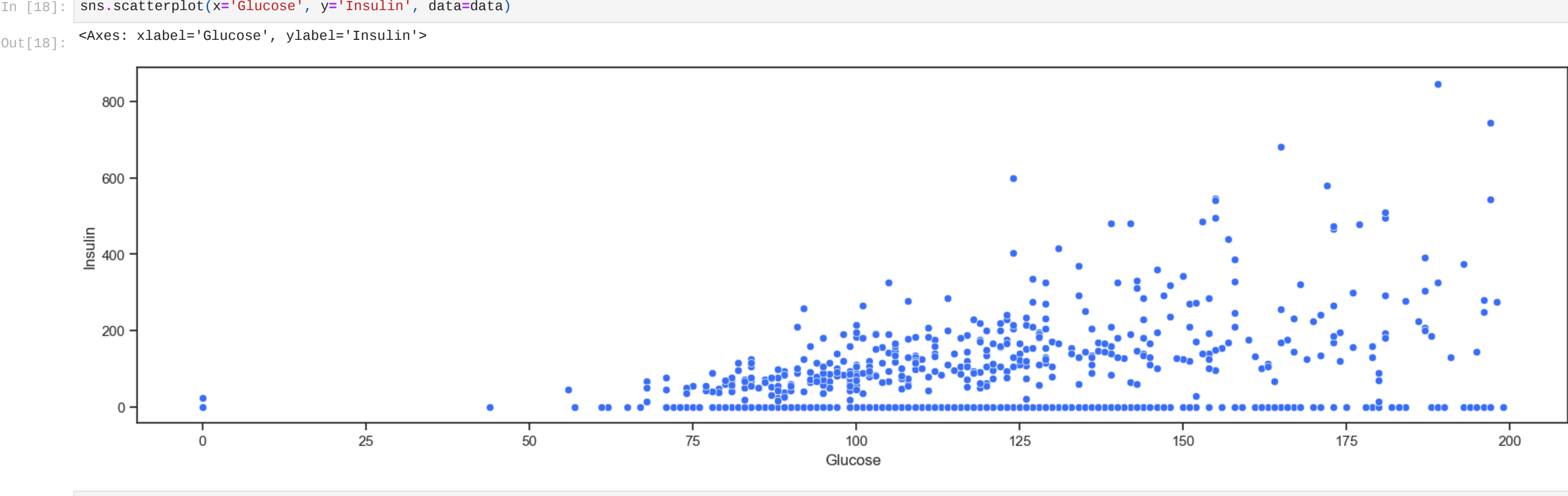
```
In [17]: sns.heatmap(round(data.corr(), 2), annot = True, cmap = 'Blues')
```

```
Out[17]: <Axes: >
```

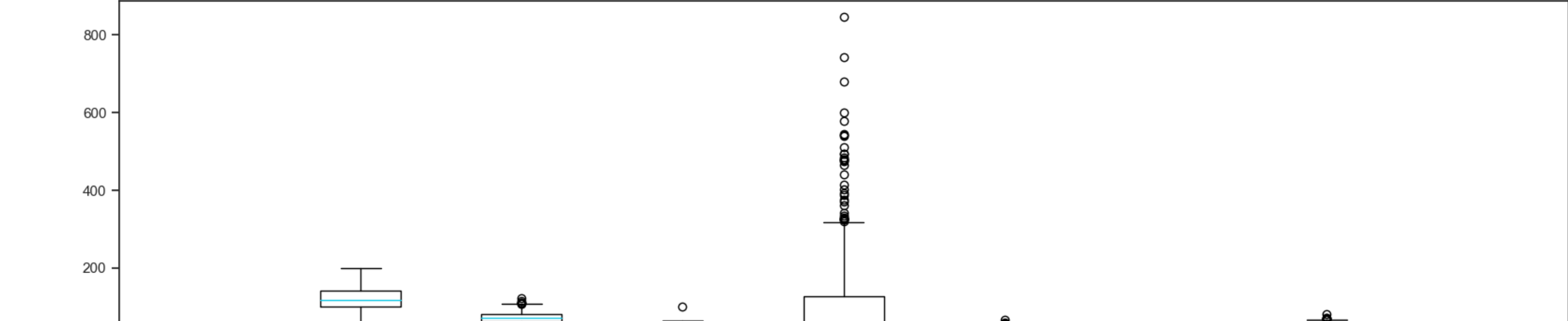


```
In [18]: sns.scatterplot(x='Glucose', y='Insulin', data=data)
```

```
<Axes: xlabel='Glucose', ylabel='Insulin'>
```



```
In [19]: plt.boxplot(data);
```



```
In [ ]:
```