```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

url = "https://raw.githubusercontent.com/plotly/datasets/master/diabetes.csv"
df = pd.read_csv(url)
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPed
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

Next steps: Generate code with df | New interactive sheet

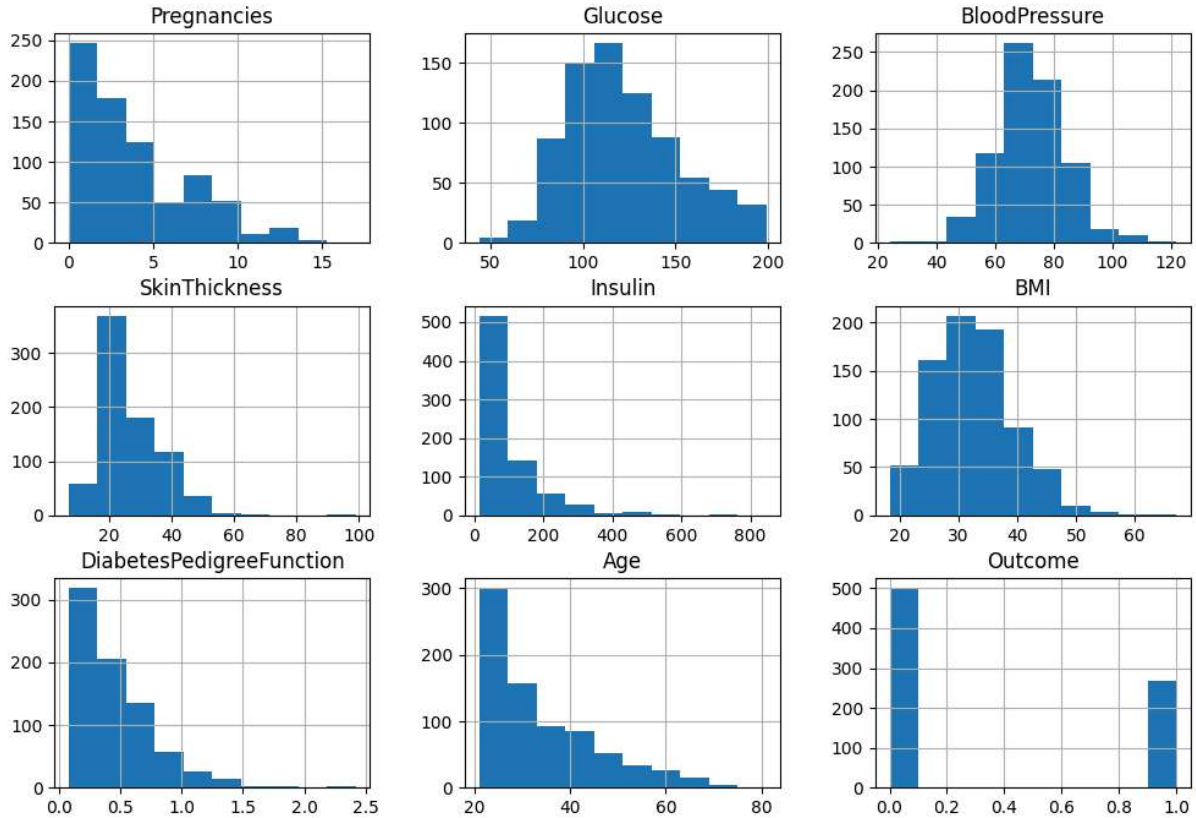
```
df.describe()
```

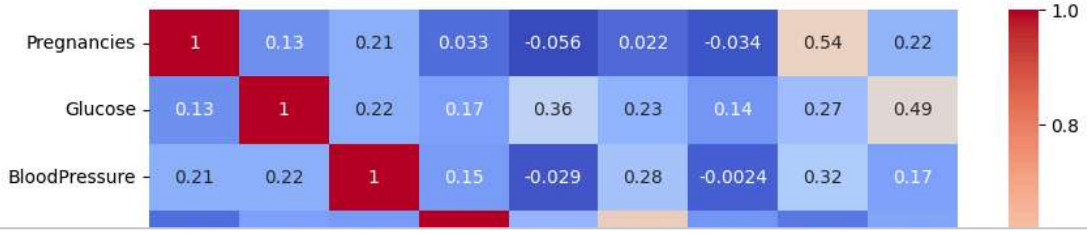
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100

```
df.isnull().sum()
df.duplicated().sum()
cols = ["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]
```

```
for col in cols:
    df[col] = df[col].replace(0, df[col].median())
```

```
df.hist(figsize=(12,8))
plt.show()
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.show()
sns.countplot(x="Outcome", data=df)
plt.show()
```






```
X = df.drop("Outcome", axis=1)
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
from sklearn.cluster import KMeans

inertia = []
K_range = range(2, 10)

for k in K_range:
    km = KMeans(n_clusters=k, random_state=42)
    km.fit(X_scaled)
    inertia.append(km.inertia_)

plt.plot(K_range, inertia, marker="o")
plt.xlabel("Number of Clusters (K)")
plt.ylabel("Inertia")
```

```
plt.title("Elbow Method")
plt.show()
from sklearn.metrics import silhouette_score

for k in range(2,6):
    km = KMeans(n_clusters=k, random_state=42)
    labels = km.fit_predict(X_scaled)
    print(f"K={k}, Silhouette Score={silhouette_score(X_scaled, labels):.3f}")
    from sklearn.decomposition import PCA

kmeans = KMeans(n_clusters=2, random_state=42)
clusters = kmeans.fit_predict(X_scaled)

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.scatter(X_pca[:,0], X_pca[:,1], c=clusters, cmap="viridis")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.title("K-Means Clustering")
plt.show()
```