

Double-click (or enter) to edit

بسم الله الرحمن الرحيم

اشتغلت على داتا سيت لمرض السكر ولاكن لاتوجد بها اخطاء لكن تم تنفيذ أكواد فحص ومعالجة الأخطاء وشكرا

This project is a study of diabetic patients Dataset.

Dr:Hager Saleh

Student:Jihad Abdullah Alharibi,253002789

Dataset description

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df = pd.read_csv('diabetes.csv')
```

df.dtypes

	0
Pregnancies	int64
Glucose	int64
BloodPressure	int64
SkinThickness	int64
Insulin	int64
BMI	float64
DiabetesPedigreeFunction	float64
Age	int64
Outcome	int64

dtype: object

df.head()

	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

df.dtypes

	0
Pregnancies	int64
Glucose	int64
BloodPressure	int64
SkinThickness	int64
Insulin	int64
BMI	float64
DiabetesPedigreeFunction	float64
Age	int64
Outcome	int64

dtype: object

df.columns

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
      'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

df.columns

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
      'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

df.isnull().sum()

	0
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

dtype: int64

```
missing_percentage = df.isnull().mean() * 100
threshold = 30
```

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in df.select_dtypes(include=['object']).columns:

    df[column] = label_encoder.fit_transform(df[column])
df.head()
```

	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

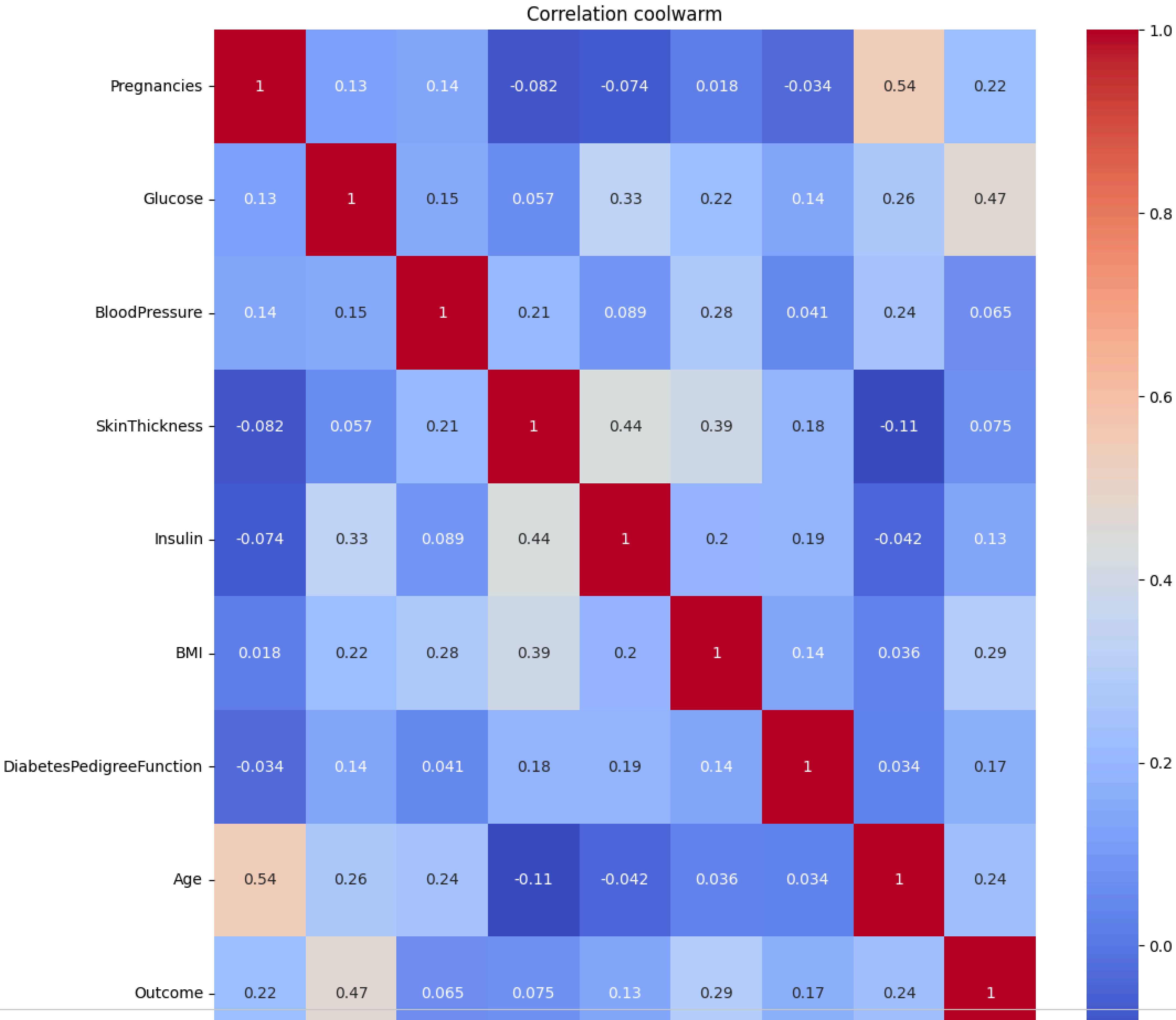
```
for column in df.select_dtypes(include=['float64']).columns:
    mean_value = df[column].mean() # Get the mode (most frequent

    df[column] = df[column].fillna(mean_value)
# Fill missing values in categorical columns with mode
for column in df.select_dtypes(include=['object']).columns:
    mode_value = df[column].mode()[0] # Get the mode (most frequent

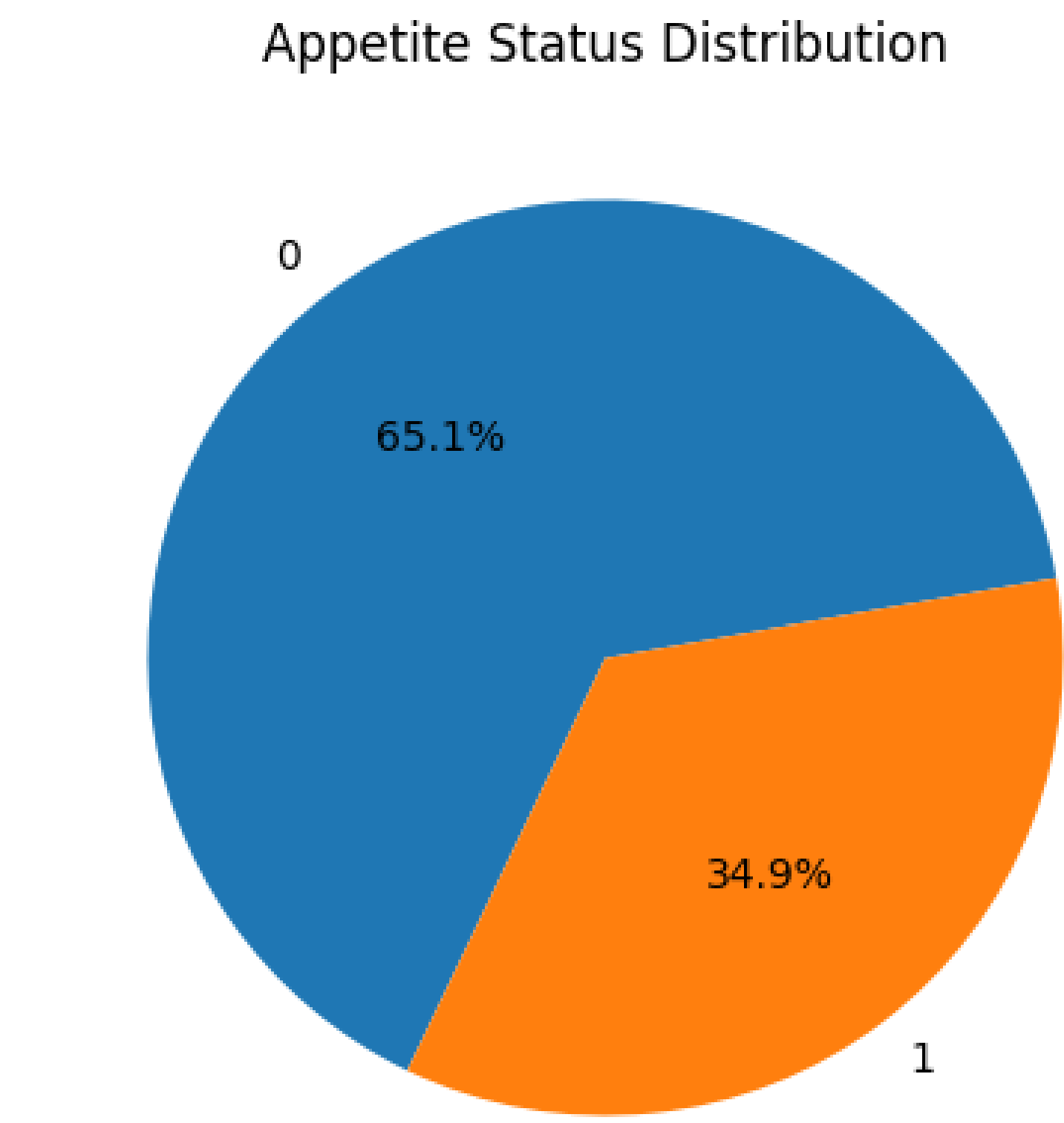
    df[column] = df[column].fillna(mode_value)
df.isnull().sum()
```

```
plt.figure(figsize=(12, 12))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation coolwarm')
```

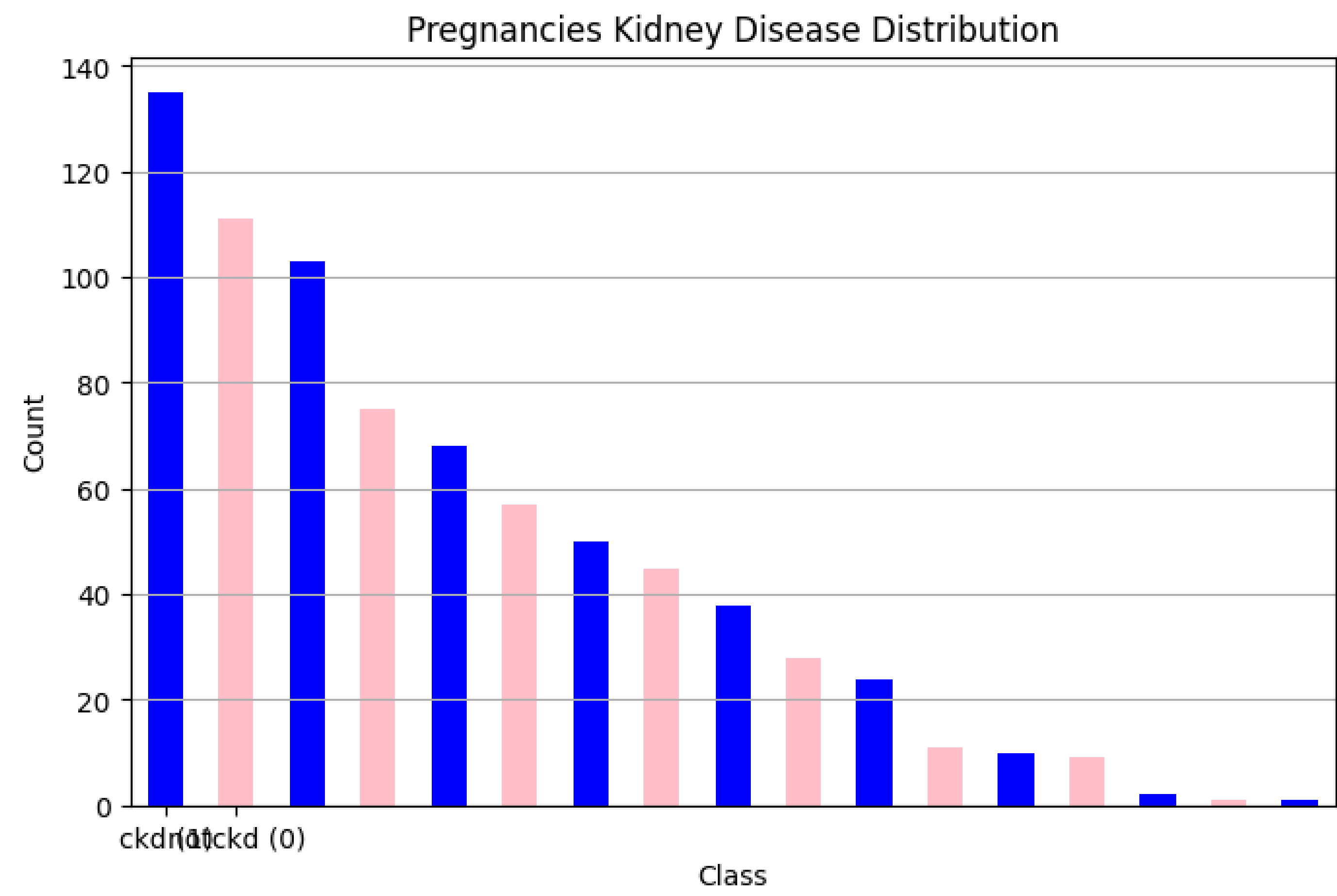
Text(0.5, 1.0, 'Correlation coolwarm')



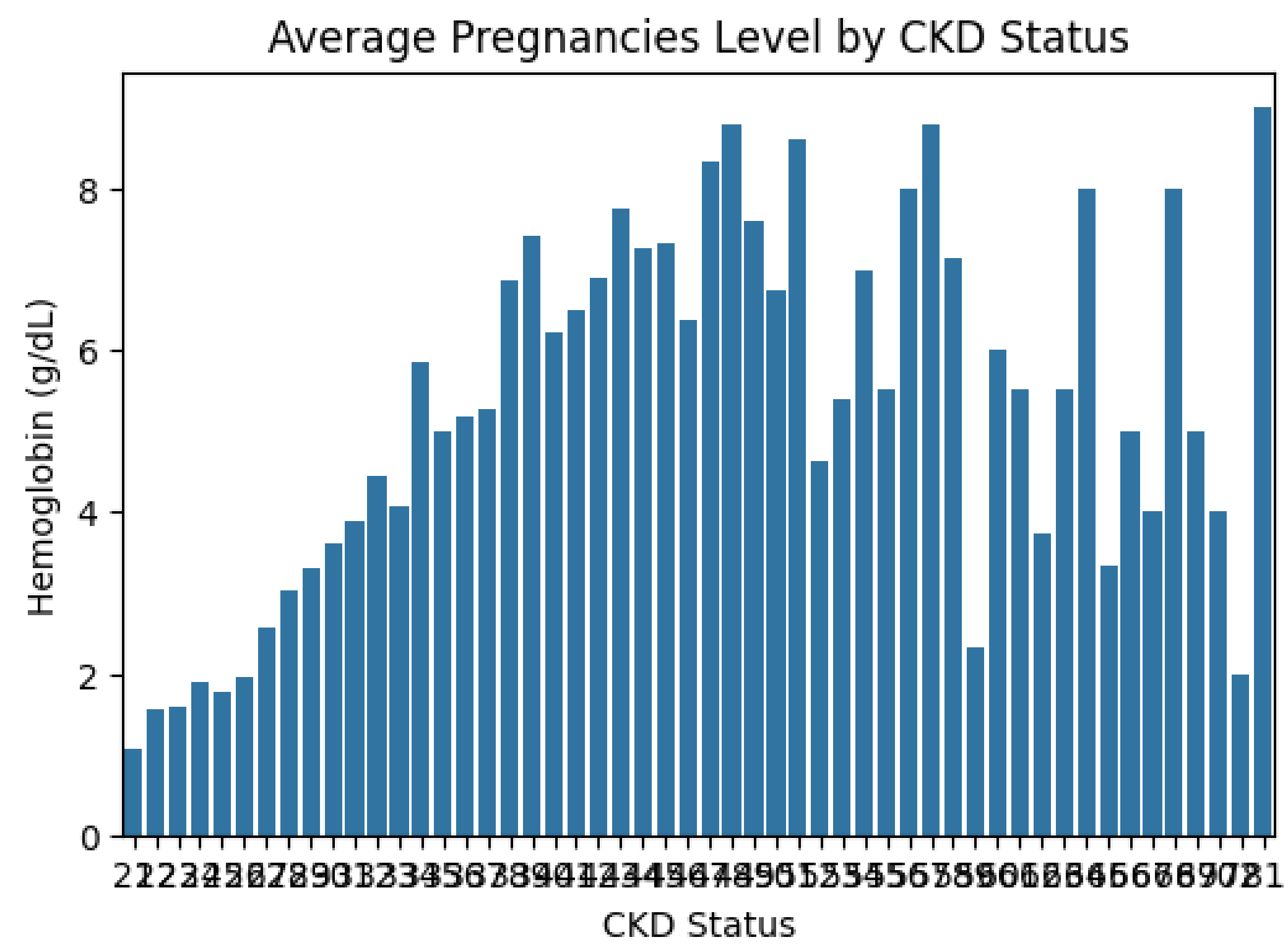
appet_counts = df['Outcome'].value_counts()
plt.figure(figsize=(5,5))
plt.pie(appet_counts, labels=appet_counts.index, autopct='%1.1f%%',
startangle=10)
plt.title("Appetite Status Distribution")
plt.show()



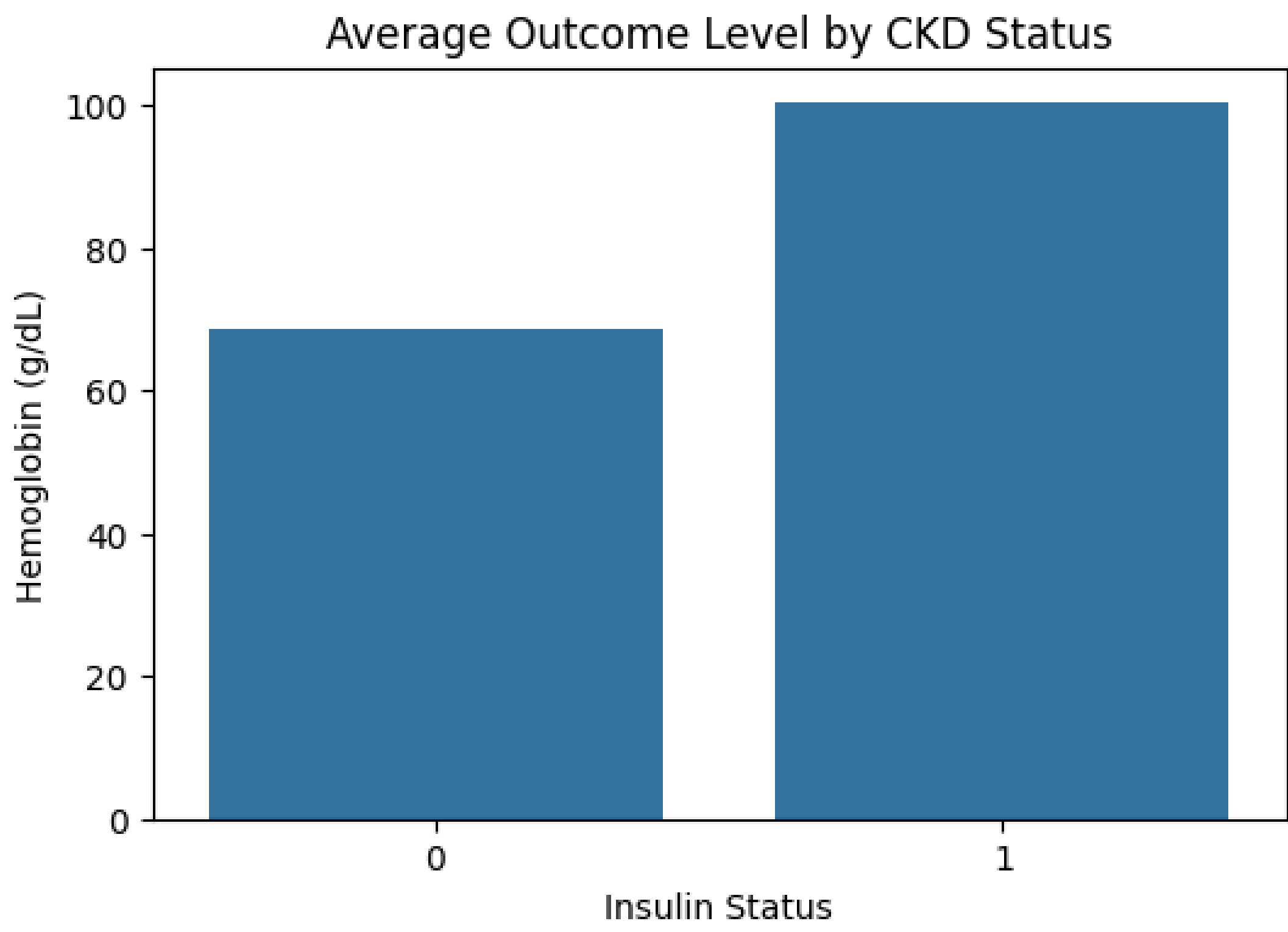
plt.figure(figsize=(8, 5))
df['Pregnancies'].value_counts().plot(kind='bar', color=['blue',
'pink'])
plt.xticks([0, 1], ['ckd (1)' , 'notckd (0)'], rotation=0)
plt.title('Pregnancies Kidney Disease Distribution')
plt.xlabel('Class')
plt.ylabel('Count')
plt.grid(axis='y')
plt.show()



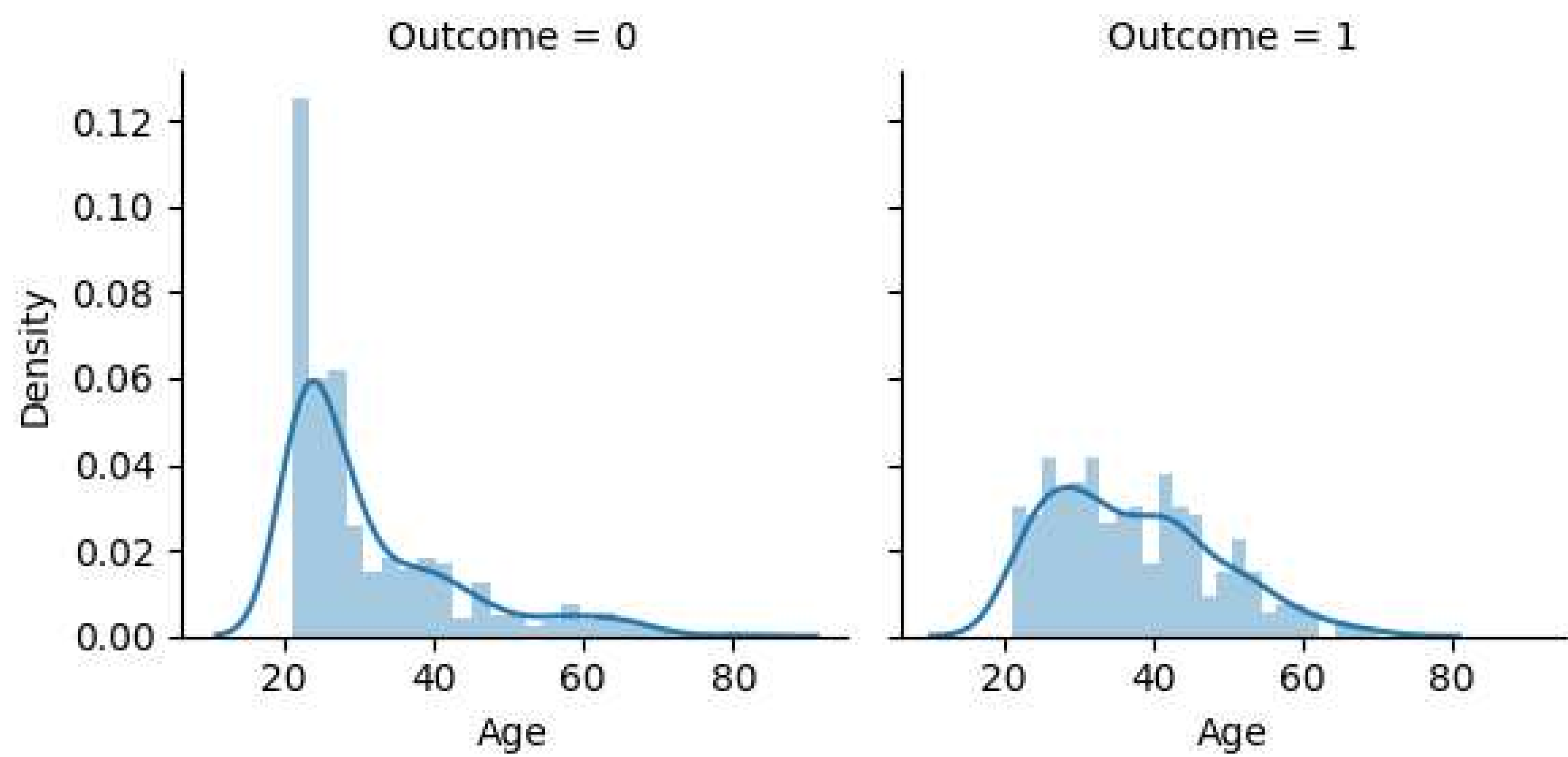
import seaborn as sns
plt.figure(figsize=(6,4))
sns.barplot(x='Age', y='Pregnancies', data=df, estimator='mean',
ci=None)
plt.title("Average Pregnancies Level by CKD Status")
plt.xlabel("CKD Status")
plt.ylabel("Hemoglobin (g/dL)")
plt.show()



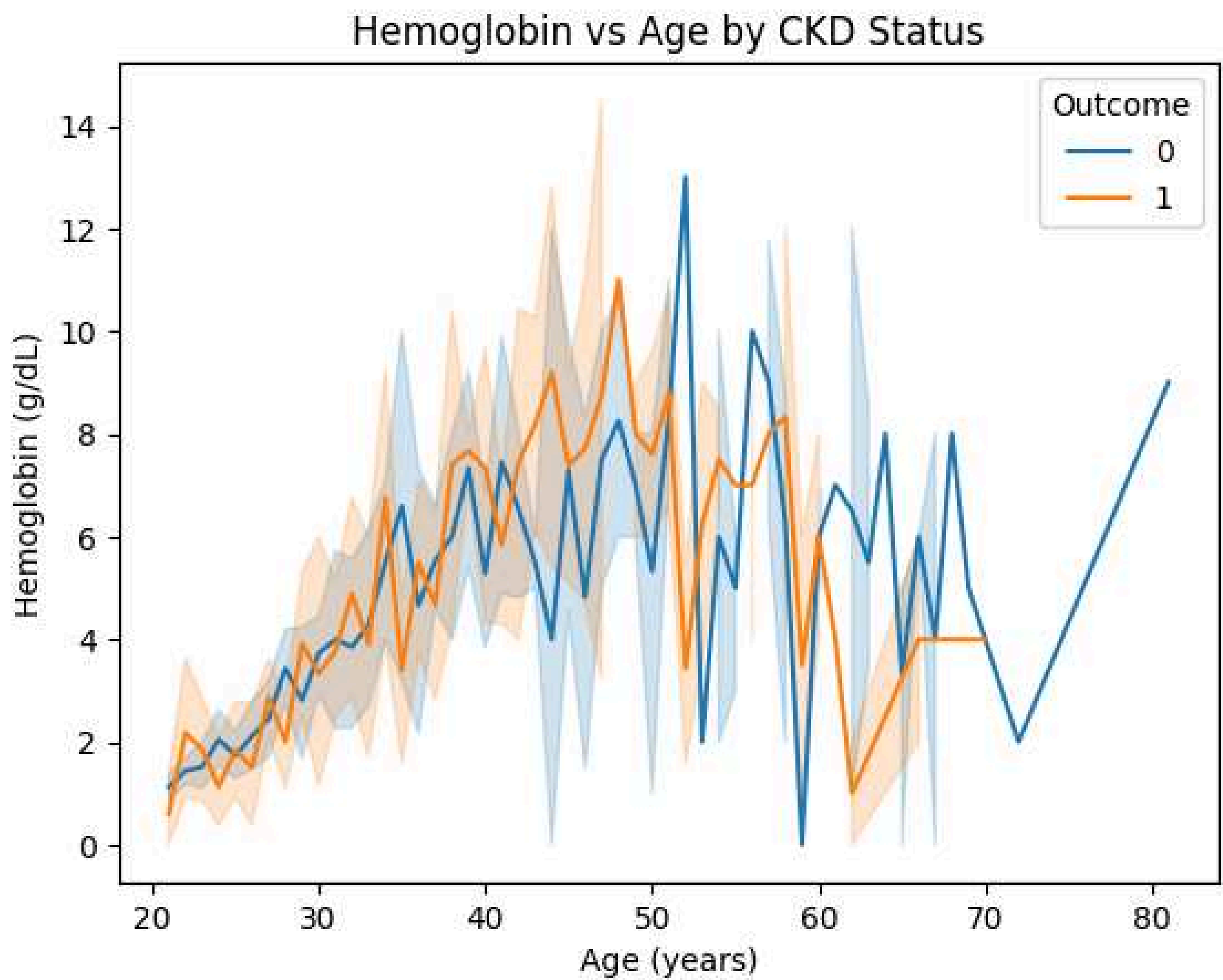
import seaborn as sns
plt.figure(figsize=(6,4))
sns.barplot(x='Outcome', y='Insulin', data=df, estimator='mean',
ci=None)
plt.title("Average Outcome Level by CKD Status")
plt.xlabel("Insulin Status")
plt.ylabel("Hemoglobin (g/dL)")
plt.show()



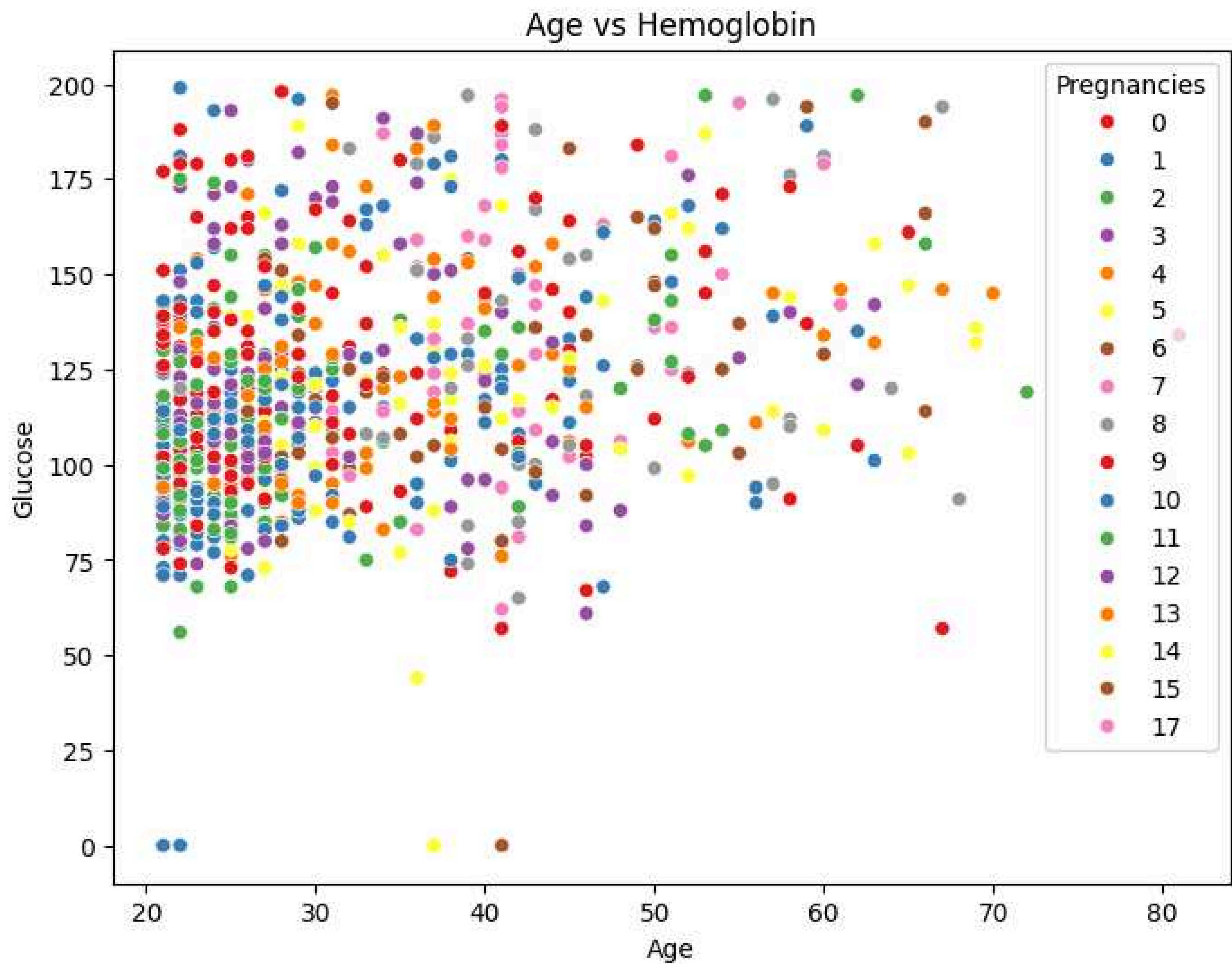
```
g = sns.FacetGrid(df,col="Outcome")
g.map(sns.distplot,"Age", bins=25)
plt.show()
```



```
sns.lineplot(x='Age', y='Pregnancies', hue='Outcome', data=df)
plt.title("Hemoglobin vs Age by CKD Status")
plt.xlabel("Age (years)")
plt.ylabel("Hemoglobin (g/dL)")
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='Glucose', data=df, hue='Pregnancies',
palette='Set1')
plt.title('Age vs Hemoglobin')
plt.xlabel('Age')
plt.ylabel('Glucose')
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.boxplot(x='Outcome', y='Age', data=df, palette='Set1')
plt.title('Hemoglobin Levels by Classification')
```

```
plt.figure(figsize=(10,6))
plt.xlabel('Outcome')
plt.ylabel('Outcome')
plt.show()
```

