```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
from seaborn import PairGrid

# Load dataset
df = pd.read_csv("insurance.csv")
#print(df.head())

# CHECK SHAPE OF DATA AND TYPE OF COLUMNS
print("\nThe shape of the data is: \n", df.shape)
print("\nThe data type of columns are: \n", df.dtypes)

print("\nStatistics on the dataset: \n")
print(df.describe())

# CHECK MISSING VALUES IN THE DATA FRAME
print("\nMissing values and their count: \n", df.isnull().sum())

# COUNTER FOR FIGURE NUMBER
fig_num = 1

# VARIABLE DEFINITIONS
numerical_columns = ["age", "bmi", "charges"]
categorical_columns = ["sex", "children", "smoker", "region"]
numerical_features = ["age", "bmi"]

# COUNT PLOTS FOR CATEGORICAL COLUMNS
for column in categorical_columns:
    plt.figure(num=fig_num, figsize=(8, 5))
    ax = sb.countplot(x=column, data=df)
    plt.ylabel("Count")
    plt.xlabel(column.capitalize())

    # CUSTOM TITLES
    if column == "sex":
        plt.title("Count of Customers by Sex")
        observation = f"Observation: Dataset has
{df[column].value_counts().to_dict()} distribution by sex."
    elif column == "children":
        plt.title("Count of Customers by Number of Children")
        observation = f"Observation: Most customers have
{df[column].mode()[0]} children, distribution:
{df[column].value_counts().to_dict()}."
    elif column == "smoker":
        plt.title("Count of Smokers vs Non-Smokers")
        observation = f"Observation: Smokers:
{df[column].value_counts().to_dict().get('yes',0)}, Non-smokers:
{df[column].value_counts().to_dict().get('no',0)}."
    elif column == "region":
        plt.title("Count of Customers by Region")
        observation = f"Observation: Region distribution is
{df[column].value_counts().to_dict()}."

    # Add bar labels
    for container in ax.containers:
        ax.bar_label(container)
```

```
    plt.tight_layout()
    plt.show()
    #print(observation)
    print(f"Figure {fig_num}: {observation}")
    fig_num += 1

# SCATTER PLOTS FOR NUMERICAL FEATURES VS CHARGES

for feature in numerical_features:
    plt.figure(num=fig_num, figsize=(8, 5))
    sb.scatterplot(x=feature, y="charges", data=df, hue="smoker",
alpha=0.6)
    plt.title(f"Insurance Charges vs {feature.capitalize()}")
    plt.ylabel("Charges")
    plt.xlabel(feature.capitalize())
    plt.tight_layout()
    plt.show()

    corr_value = df[[feature, "charges"]].corr().iloc[0, 1]
    observation = f"Observation: {feature.capitalize()} has a correlation
of {corr_value:.3f} with charges. Charges tend to {'increase' if
corr_value > 0 else 'decrease'} with {feature}."
    #print(observation)
    print(f"Figure {fig_num}: {observation}")
    fig_num += 1

# PAIR PLOTS FOR NUMERICAL COLUMNS

g = PairGrid(df[numerical_columns], diag_sharey=False, corner=True)
g.map_lower(sb.scatterplot, alpha=0.6)
g.map_diag(sb.kdeplot, fill=True)
g.figure.set_size_inches(8, 6)
g.figure.suptitle("Pairwise Relationships between Numerical Variables",
fontsize=16)
g.figure.subplots_adjust(top=0.95)
plt.show()
observation = "Observation: Pairplot shows mild positive correlation of
age and bmi with charges, with smokers standing out in higher charge
ranges."
print(f"Figure {fig_num}: {observation}")
fig_num += 1  # Increment figure counter

# MULTIVARIATE ANALYSIS USING SCATTER PLOTS BETWEEN NUMERICAL VARIABLES

#numerical_columns = ["age", "bmi", "charges"]

for i in range(len(numerical_columns)):
    for j in range(i + 1, len(numerical_columns)):
        plt.figure(num=fig_num, figsize=(8, 5))
        sb.scatterplot(x=numerical_columns[i], y=numerical_columns[j],
data=df, hue="smoker", alpha=0.6)
        plt.title(f"Scatter Plot of {numerical_columns[i]} vs
{numerical_columns[j]}")
        plt.tight_layout()
        plt.show()

        corr_val = df[[numerical_columns[i],
numerical_columns[j]]].corr().iloc[0, 1]
```

```
        observation = f"Observation: Correlation between
{numerical_columns[i]} and {numerical_columns[j]} is {corr_val:.3f}."
        print(f"Figure {fig_num}: {observation}")
        #print(observation)
        fig_num += 1

# CHARGES VS AGE WITH SMOKER STATUS

plt.figure(num=fig_num, figsize=(10, 6))
sb.regplot(x="age", y="charges", data=df[df["smoker"] == "yes"],
scatter_kws={"alpha":0.5}, label="Smoker", color="red")
sb.regplot(x="age", y="charges", data=df[df["smoker"] == "no"],
scatter_kws={"alpha":0.5}, label="Non-Smoker", color="blue")
plt.title("Insurance Charges vs Age by Smoker Status")
plt.xlabel("Age")
plt.ylabel("Insurance Charges")
plt.legend()
plt.tight_layout()
plt.show()
observation = "Observation: Charges increase with age for both smokers
and non-smokers, but the slope is much steeper for smokers, indicating a
compounding risk effect."
print(f"Figure {fig_num}: {observation}")
fig_num += 1
```