Seaborn Full Code with Detailed Explanations (US Honey Dataset)

Code 1:

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

Explanation: Importing necessary libraries: Seaborn for visualization, Pandas for data handling, and Matplotlib for displaying plots.

Code 2:

```
honey=pd.read_csv('US_honey_dataset_updated.csv')
```

Explanation: Loading the US Honey dataset CSV file into a pandas DataFrame for analysis.

Code 3:

```
sns.histplot(honey['yield_per_colony'], kde=True)
plt.show()
```

Explanation: Plotting a histogram with KDE for 'yield_per_colony' to analyze the distribution of honey yield per colony across all records.

Code 4:

```
#Observations from your plot:
#Most common yield range:
#Around 45 to 65 pounds per colony (Since tallest bars lie here)

#Distribution Shape:
#This is right-skewed (positively skewed)
#?? Meaning: Most states had lower to mid-range yields, but few had very high yields (over 100
pounds)

#Outliers:
#Some rare years/states had exceptionally high yields (over 120 pounds)
```

Explanation: Interpreting the histogram output: Observing the most frequent yield ranges and identifying any skewness or outliers in production levels.

Code 5:

```
sns.kdeplot(honey['average_price'], shade=True)
plt.show()
```

Explanation: Plotting a KDE plot for 'average_price' to observe the smooth density estimation of average honey prices across all state-year combinations.

Code 6:

```
plt.figure(figsize=(10,12))
sns.countplot(y='state', data=honey)
plt.show()
```

Explanation: Visualizing the count of records per state using a countplot to understand data distribution among states.

Code 7:

Seaborn Full Code with Detailed Explanations (US Honey Dataset)

```
plt.figure(figsize=(10,6))
sns.barplot(x='year', y='production', data=honey)
plt.xticks(rotation=90)
plt.show()
```

Explanation: Creating a barplot to visualize total production per year, helping identify production trends over time.

Code 8:

```
plt.figure(figsize=(10,6))
sns.boxplot(x='state', y='average_price', data=honey)
plt.xticks(rotation=90)
plt.show()
```

Explanation: Using a boxplot to compare the distribution of average prices across different states, helping identify states with high price variation or outliers.

Code 9:

```
plt.figure(figsize=(10,6))
sns.violinplot(x='year', y='yield_per_colony', data=honey)
plt.xticks(rotation=90)
plt.show()
```

Explanation: Generating a violin plot to visualize yield per colony over the years, showing both distribution and median values for each year.

Code 10:

```
plt.figure(figsize=(10,6))
sns.violinplot(x='year', y='yield_per_colony', data=honey, hue='year', palette='viridis',
legend=False)
plt.xticks(rotation=90)
plt.title('Year-wise Honey Yield Distribution')
plt.show()
```

Explanation: Plotting a scatterplot showing the relationship between colonies number and total honey production, with year-wise color coding using hue.

Code 11:

```
plt.figure(figsize=(10,6))
sns.scatterplot(x='colonies_number', y='production', data=honey, hue='year')
plt.show()
```

Explanation: Adding a regression line on the colonies vs production scatterplot to visualize the linear relationship and production trends with colony size.

Code 12:

```
plt.figure(figsize=(10,6))
sns.lineplot(x='year', y='production', data=honey)
plt.show()
```

Explanation: Using pairplot to analyze pairwise relationships between multiple numeric variables like colonies, yield, production, and price, with year-wise hue coloring.

Seaborn Full Code with Detailed Explanations (US Honey Dataset)

Code 13:

```
plt.figure(figsize=(10,6))
sns.regplot(x='colonies_number', y='production', data=honey)
plt.show()
```

Explanation: Creating a heatmap to visualize correlation coefficients between all numeric columns, helping understand which factors influence production most.

Code 14:

```
sns.pairplot(honey, vars=['colonies_number', 'yield_per_colony', 'production', 'average_price'])
plt.xticks(rotation=90)
plt.show()
```

Explanation: Fixing heatmap error by using only numeric columns and visualizing the corrected correlation heatmap with clear color gradients and annotations.

Code 15:

```
plt.figure(figsize=(10,8))
sns.heatmap(honey.select_dtypes(include='number').corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Numerical Columns')
plt.show()
```

Explanation: Generating a FacetGrid with lineplots showing production trends per state over the years, useful for state-wise time-series visualization.

Code 16:

```
g = sns.FacetGrid(honey, col="state", col_wrap=3, height=3)
g.map(sns.lineplot, "year", "production")
plt.show()
```

Explanation: Filtering top 5 states based on total production and plotting their individual production trends using FacetGrid to focus on high-performing states.

Code 17:

```
top_states =
honey.groupby('state')['production'].sum().sort_values(ascending=False).head(5).index.tolist()

top_honey = honey[honey['state'].isin(top_states)]

g = sns.FacetGrid(top_honey, col="state", col_wrap=3, height=3)
g.map(sns.lineplot, "year", "production")
plt.show()
```

Explanation: Explaining how to calculate top 5 states by grouping and sorting the dataset by total production, then filtering for those states.

Code 18:

Explanation: Plotting the final FacetGrid for top 5 states with clear production trends and visual clarity for classroom analysis.