Visualizing Loans Awarded by Kiva

In this project you'll visualize insights using a dataset from Kaggle. The dataset contains information about loans awarded by the non-profit Kiva.

Using Seaborn, you'll explore the average loan amount by country using aggregated bar charts. You'll also visualize the distribution of loan amounts by project type and gender using box plots and violin plots.

Some of the steps below will have hints that you can access if you need them. Hints will look like this:

▶ Hint (click me)

A Note On `plt.show()` : You may be used to displaying your plots using the code `plt.show()`. This IPython Jupyter notebook removes the necessity of calling `plt.show()` after each plot. You should be able to render your Seaborn plots simply by running the cell with the code for your plot. If you have issues rendering your plot you can try adding `plt.show()` to a cell.

Step 1: Import Necessary Python Modules

Import the modules that you'll be using in this project:

- `from matplotlib import pyplot as plt`
- `import pandas as pd`
- `import seaborn as sns`

In [1]:

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

Step 2: Ingest The Data

Load **kiva_data.csv** into a DataFrame called `df`. Then, quickly inspect the DataFrame using `.head()`.

▶ Hint

In [2]:

```
df= pd.read_csv('kiva_data.csv')
print(df.head())
```

```
   loan_amount               activity   country  gender
0          625  Food Production/Sales  Pakistan  female
1          250  Food Production/Sales  Pakistan  female
2          400  Food Production/Sales  Pakistan  female
3          400  Food Production/Sales  Pakistan  female
4          500  Food Production/Sales  Pakistan  female
```

Step 3: Examine The Data

If you would like, you can examine the raw CSV file on your local machine. You can find **kiva_data.csv** in the project download folder.

Overview of the dataset:

Each entry (row) in the dataset represents a loan that Kiva awarded to a particular project. The `loan_amount` column shows the amount (in U.S. dollars) awarded to the project. The `activity` column has the category type that the project falls under. The `country` column is the country where the project is located. The `gender` column represents the gender of the primary person who applied for the loan.

Print the first 25 rows of `df` using `.head()`

In [3]:

```
print(df.head(25))
```

```
    loan_amount               activity   country  gender
0           625  Food Production/Sales  Pakistan  female
1           250  Food Production/Sales  Pakistan  female
2           400  Food Production/Sales  Pakistan  female
3           400  Food Production/Sales  Pakistan  female
4           500  Food Production/Sales  Pakistan  female
5           500  Food Production/Sales  Pakistan  female
6           400  Food Production/Sales  Pakistan  female
7           500  Food Production/Sales  Pakistan  female
8           400  Food Production/Sales  Pakistan  female
9           450  Food Production/Sales  Pakistan  female
10          250  Food Production/Sales  Pakistan  female
11          300  Food Production/Sales  Pakistan  female
12          275  Food Production/Sales  Pakistan  female
13          425  Food Production/Sales  Pakistan  female
14          425  Food Production/Sales  Pakistan  female
15          475  Food Production/Sales  Pakistan  female
16          225  Food Production/Sales  Pakistan  female
17          475  Food Production/Sales  Pakistan  female
18          525  Food Production/Sales  Pakistan  female
19          425  Food Production/Sales  Pakistan  female
20          475  Food Production/Sales  Pakistan  female
21          550  Food Production/Sales  Pakistan  female
22          450  Food Production/Sales  Pakistan  female
23          250  Food Production/Sales  Pakistan  female
24          600  Food Production/Sales  Pakistan  female
```

Step 4: Bar Charts

Create a bar plot using Seaborn to visualize the average size of Kiva loans given to projects, by country.

We've set up the figure you'll use to plot your bar plot on. The `f` variable gives us access to the figure and `ax` gives us access to the axes.

Use `sns.barplot()` with the following arguments:

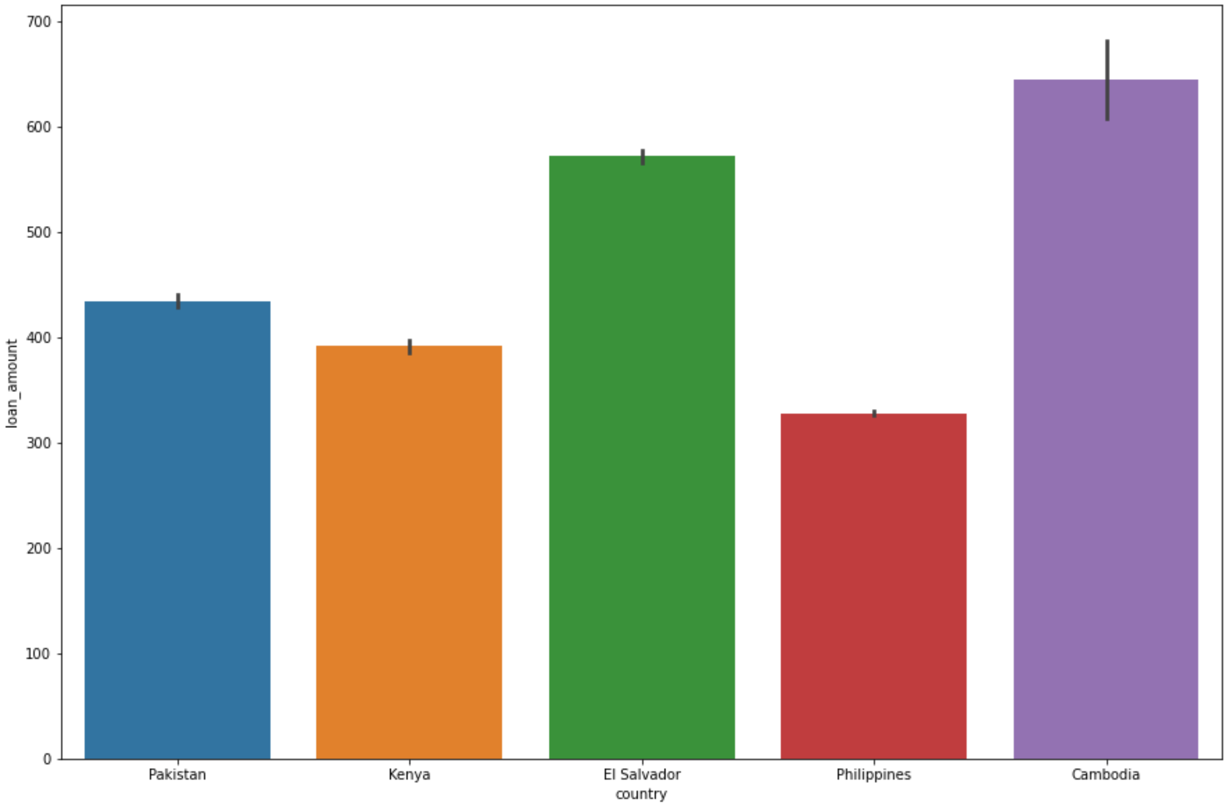
- `data` set to `df`
- `x` set to `country`
- `y` set to `loan_amount`

In [4]:

```python
# Creates the figure, note: you're only using this syntax so that you can modify the y
f, ax = plt.subplots(figsize=(15, 10))

sns.barplot(data=df,
            x= 'country',
            y= 'loan_amount')
```

Out[4]:

```
<AxesSubplot:xlabel='country', ylabel='loan_amount'>
```



Adding `$` units

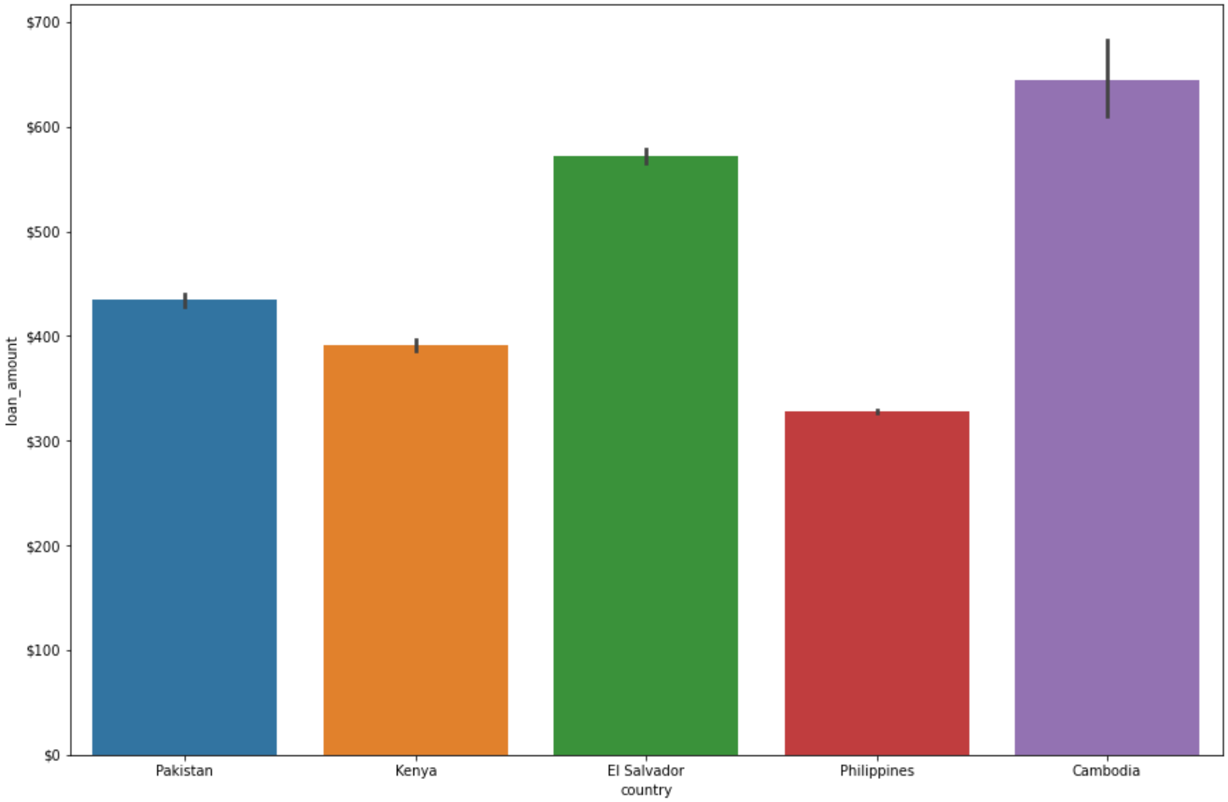
You can use the following code to so that the `loan_amount` ticks on the y-axis begin with a `$` (units of USD).

```python
import matplotlib.ticker as mtick
fmt = '${x:,.0f}'
```

```
tick = mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
```

Run the code in the cell below to see the `$` in action.

In [5]:
```
import matplotlib.ticker as mtick

# Creates the figure
f, ax = plt.subplots(figsize=(15, 10))

# Plot the data
sns.barplot(data=df, x="country", y = "loan_amount")

# Use part of the code above to format the y-axis ticks below this line
fmt= '${x:,.0f}'
tick= mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
```



Step 5: Learn More By Using `hue` In Your Visualization

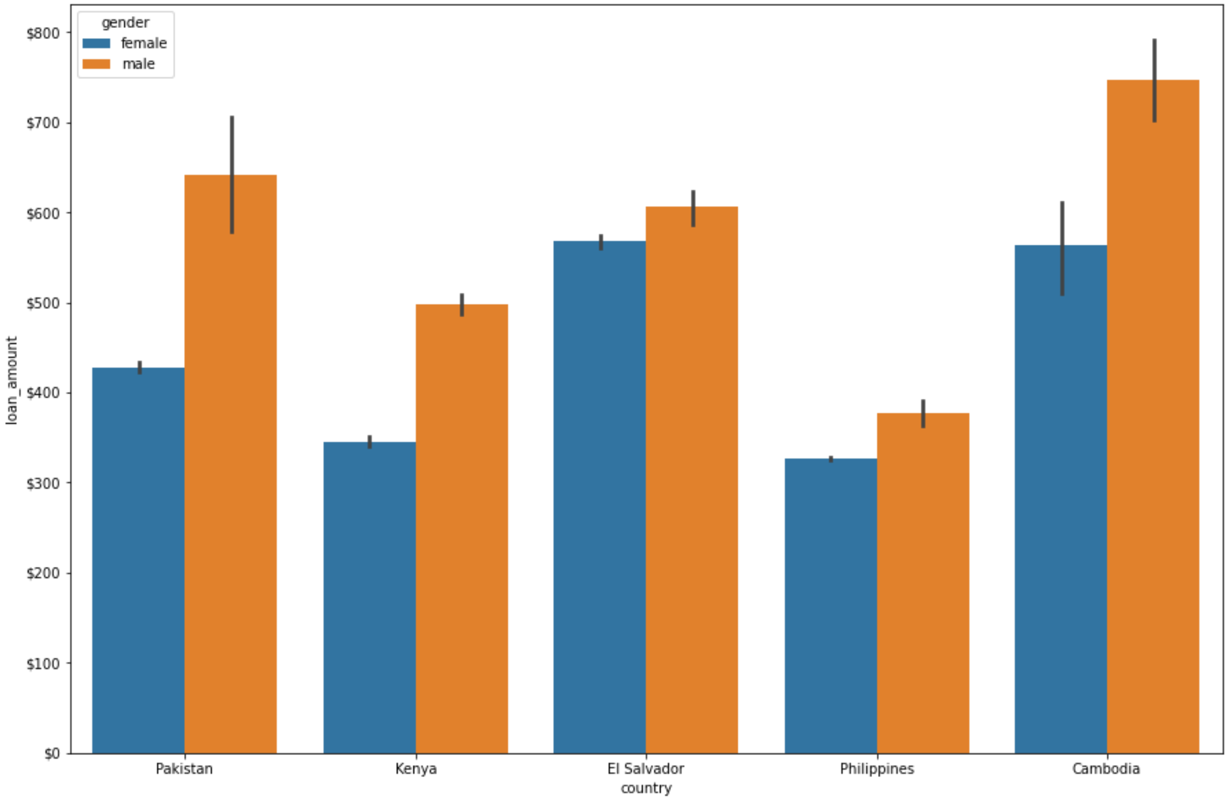
You can visualize even more data on one bar plot by visualizing the loan amount by country, and "nesting" by gender. Add the `hue` parameter to your `sns.barplot()` and set it so that the visualization includes the nested category of gender.

▶ Hint

```
# Creates the figure, you're only using this syntax so you can modify the y-axis ticks
```

In [6]:

```
f, ax = plt.subplots(figsize=(15, 10))

sns.barplot(data=df, x="country", y="loan_amount", hue='gender')

fmt = '${x:,.0f}'
tick = mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
```



Reflection Questions

On average, do female or male recipients receive larger loans from Kiva?

In [ ]:

```
print("On average, male recipients receive larger loans from Kiva.")
```

Which country has the *least* disparity in loan amounts awarded by gender?

In [ ]:

```
print("El Salvador has the least disparity in loan amounts awarded by gender.")
```

Based on the data, what kind of recommendations can you make to Kiva about the loans they give?

In [ ]:

```
print("I would recommend Kiva to investigate reasons for the gender disparity in loan
```

What actions could be taken to implement the recommendations you've made?

In [ ]:

```
print("Identify whether the larger loans issued to men correlate to positive outcomes,
such as higher repayment reliability, higher return on investment, etc. If this is the
case, reassess how women are being qualified for loans; are they being qualified fairl
```

Step 6: Styling

Set a different color palette using `sns.set_palette()`. You can use any of the Color Brewer qualitative color palettes:

- Set1
- Set2
- Set3
- Pastel1
- Pastel2
- Dark2
- Accent

You can read more about qualitative color palettes in the Seaborn documentation.

Set the plot background style using `sns.set_style()`. You can experiment with:

- whitegrid
- darkgrid
- white
- dark

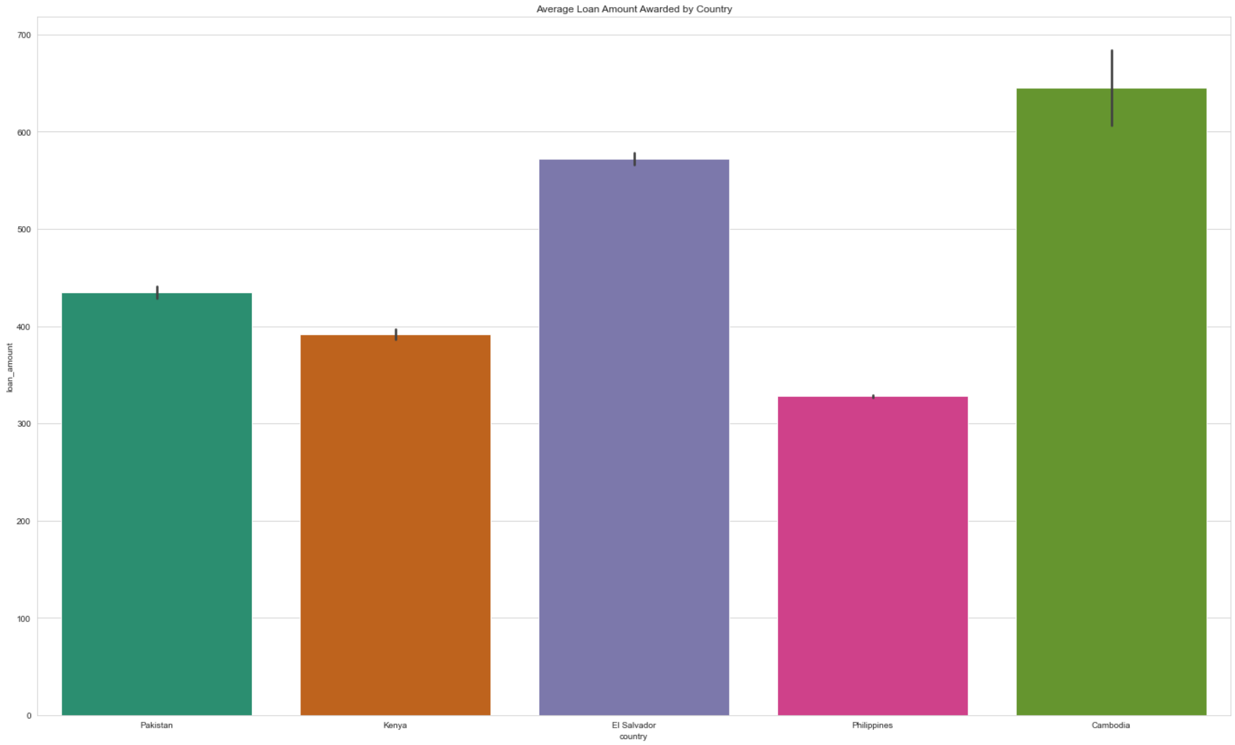
Set the title using `ax.set_title("")`.

In [11]:

```python
# Set color palette
sns.set_palette('Dark2')

# Set style
sns.set_style('whitegrid')

# Create figure and axes (no need to use the previous syntax, as the y-label ticks are
f, ax= plt.subplots(figsize=(25, 15))

# Add a title
ax.set_title("Average Loan Amount Awarded by Country")

# Use Seaborn to create the bar plot
sns.barplot(data=df,
            x='country',
            y='loan_amount')
```

Out[11]:

```
<AxesSubplot:title={'center':'Average Loan Amount Awarded by Country'}, xlabel='count
ry', ylabel='loan_amount'>
```



Step 7: Box Plots With Kiva Data

So far you have visualized the average size of loans by country using bar charts; now you are going to make a box plot to compare the distribution of loans by country.

We have set up a figure for you to plot on. Use `sns.boxplot()` to compare the distribution of loan amounts by country for the Kiva dataset.

`sns.boxplot()` can be passed the same parameters as `sns.barplot()`.

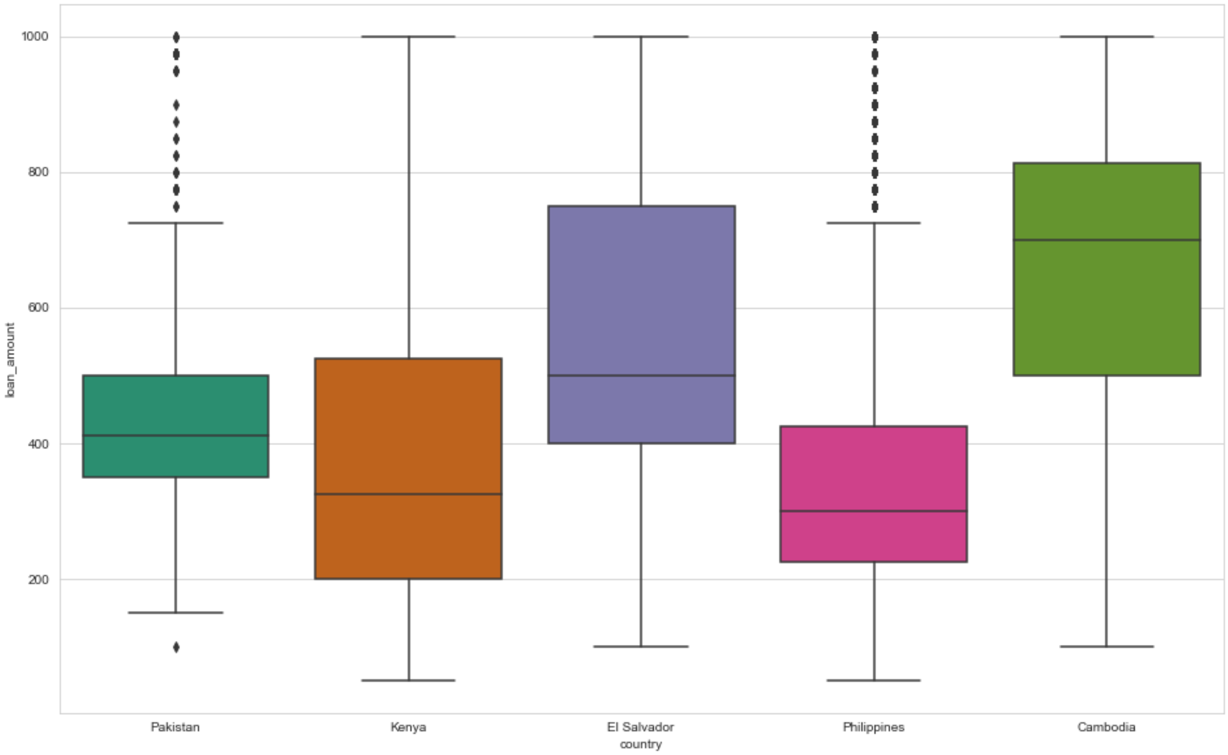
Optional: You may set a new color palette if you would like to continue using `sns.set_palette()`.

In [12]:

```
plt.figure(figsize=(16, 10))
sns.boxplot(data=df,
            x='country',
            y='loan_amount')
```

Out[12]:

```
<AxesSubplot:xlabel='country', ylabel='loan_amount'>
```



Reflection Questions

Which country's box has the widest distribution?

```
In [ ]: print("Kenya's box has the widest distribution.")
```

In which country would you be most likely to receive the largest loan amount?

```
In [ ]: print("You would be most likely to receive the largest loan amount in Cambodia.")
```

Step 8: Box Plot by Activity

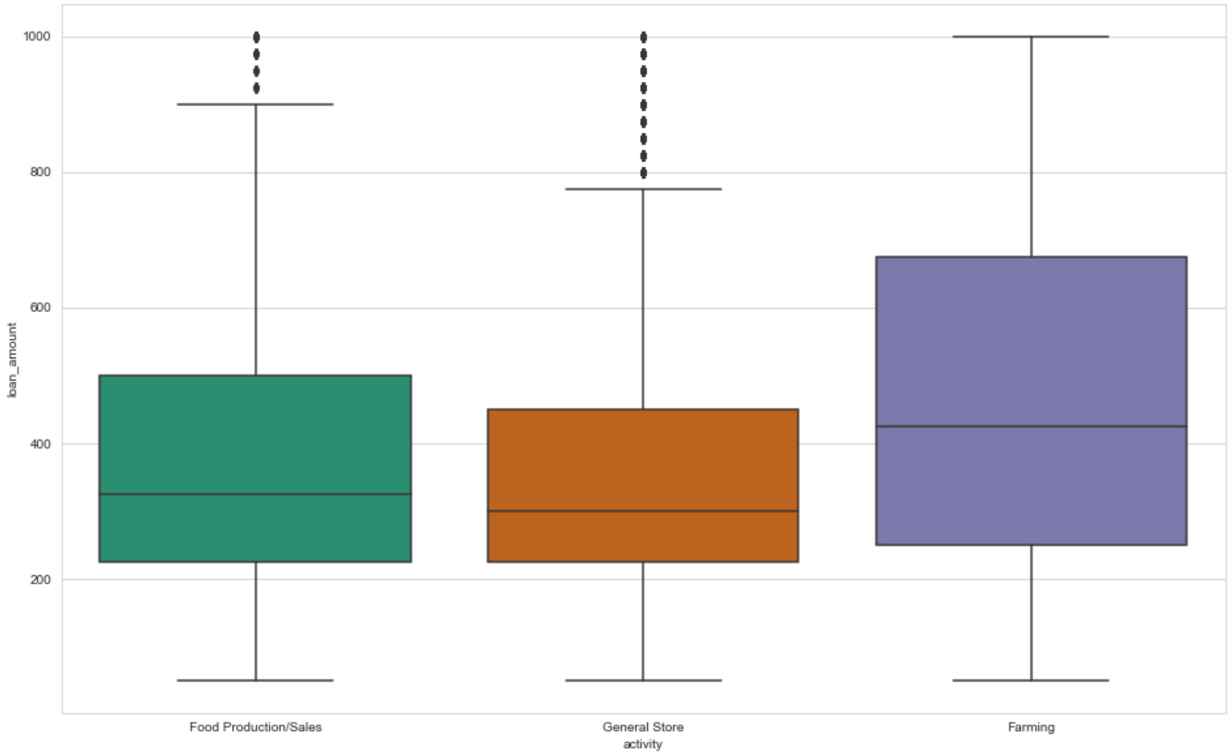
Instead of visualizing the loan amount by *country*, use `sns.boxplot()` to plot the loan amount by *activity*.

▶ Hint

Optional: Set a different plot style and color palette to best visualize this data.

```
In [13]: plt.figure(figsize=(16, 10))
         sns.boxplot(data=df,
                     x='activity',
                     y='loan_amount')
```

```
Out[13]: <AxesSubplot:xlabel='activity', ylabel='loan_amount'>
```



Reflection Questions

What does this visualization reveal that previous ones did not?

In [ ]:

```
print("Farming activities are most likely to receive higher loan amounts.")
```

Step 9: Violin Plots

You can use nearly identical syntax (as you have used for box plots) to create violin plots. Take this line of code from above:

```
sns.boxplot(data=df, x="activity", y="loan_amount")
```

To visualize the distribution of the exact same data as a violin plot you could pass the same parameters to `sns.violinplot()` instead of `sns.boxplot()`.

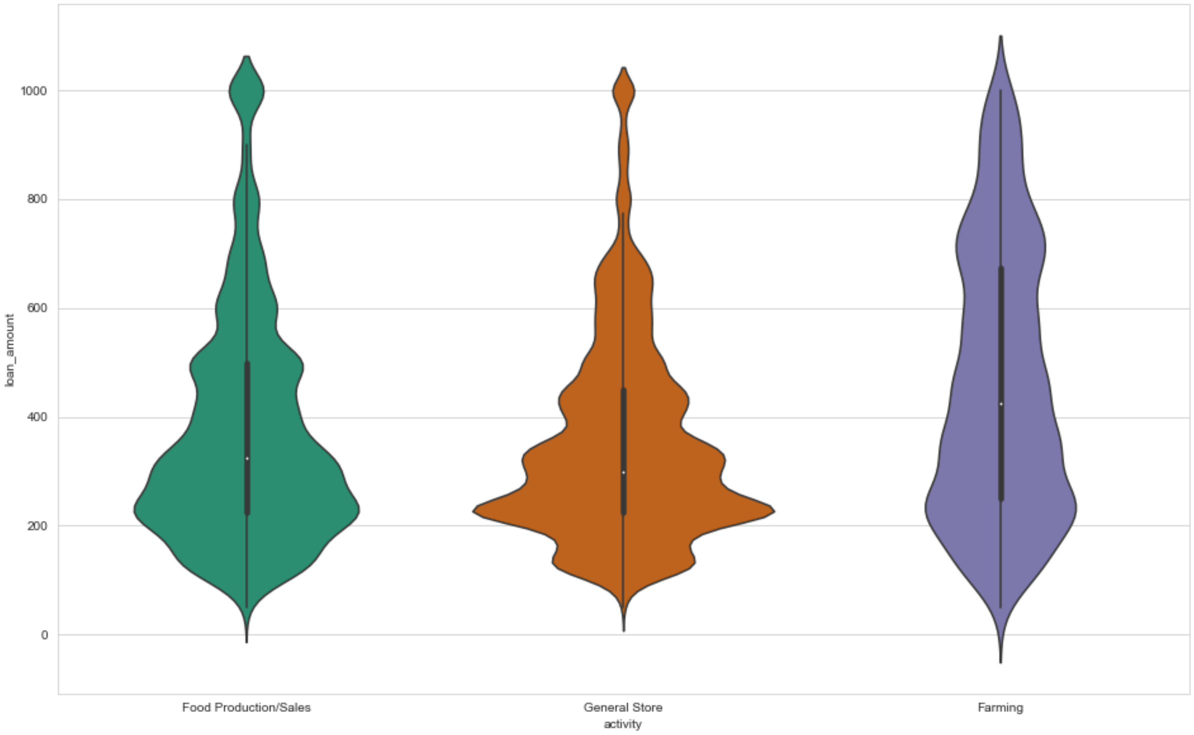
Change the code in the cell below so that the data is plotted as a violin plot instead of a barplot.

In [14]:

```
plt.figure(figsize=(16, 10))

sns.violinplot(data=df, x="activity", y="loan_amount")
```

Out[14]:

```
<AxesSubplot:xlabel='activity', ylabel='loan_amount'>
```



Create a violin plot that visualizes the distribution of loan amount by country.

Previously, you created a violin plot and plotted the data by *activity*. This time, create a violin plot that plots the data by *country*.

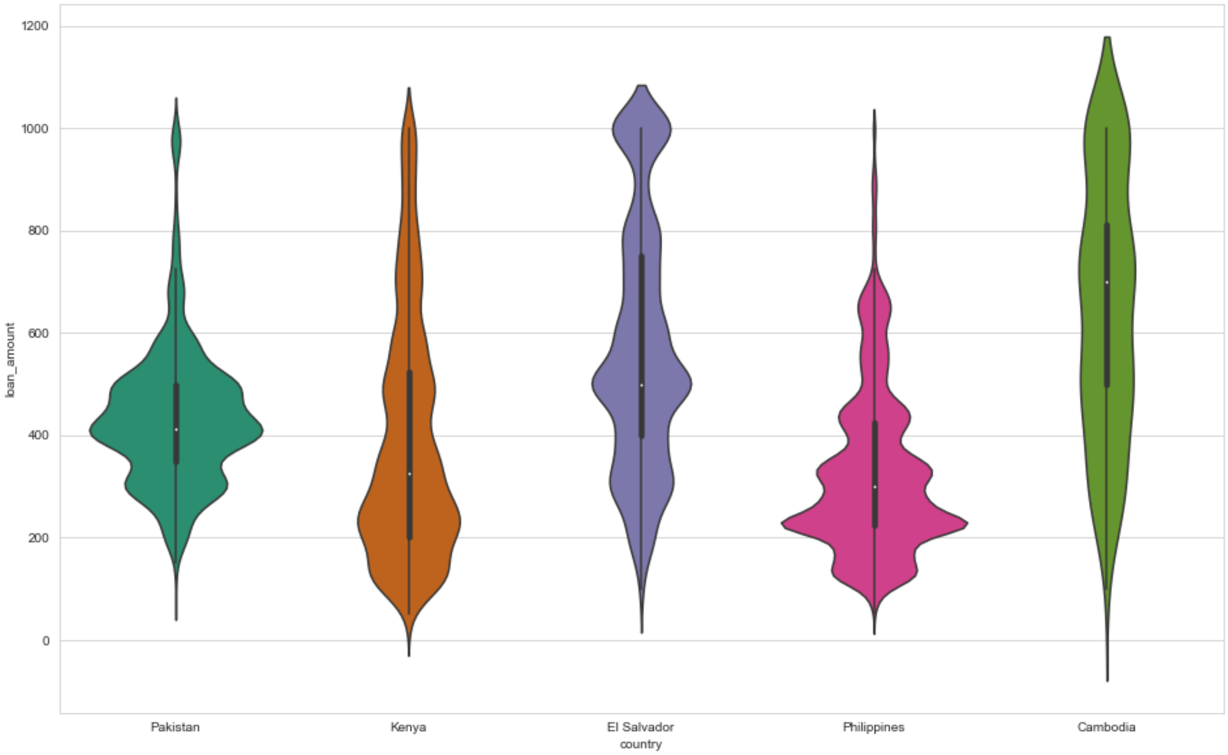
▶ Hint

In [15]:

```
plt.figure(figsize=(16, 10))
sns.violinplot(data=df,
               x='country',
               y='loan_amount')
```

Out[15]:

```
<AxesSubplot:xlabel='country', ylabel='loan_amount'>
```



Step 10: Split Violin Plots

Use the `hue` and `split` parameters with `sns.violinplot()` to visualize the distribution of loan amount by country, split by gender.

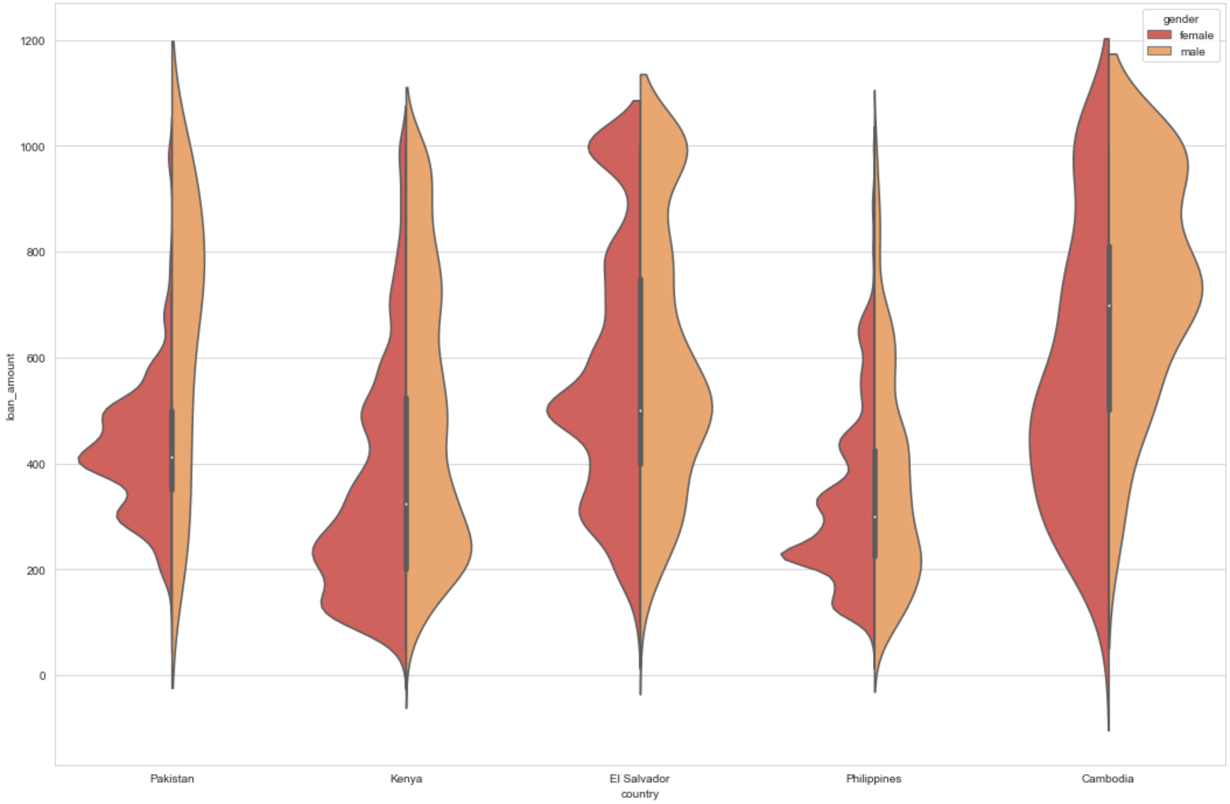
▶ Hint

In [17]:

```python
# Some styling (feel free to modify)
sns.set_palette("Spectral")
plt.figure(figsize=(18, 12))
sns.violinplot(data=df,
               x='country',
               y='loan_amount',
               hue='gender',
               split=True)
```

Out[17]:

```
<AxesSubplot:xlabel='country', ylabel='loan_amount'>
```



Reflection Questions

What does this visualization reveal about the distribution of loan amounts within countries by gender?

```
In [ ]: print("In three of the five countries, Pakistan, Kenya, and Cambodia, men tend to rece
```

You're done! Congratulations!

You used Seaborn to visualize insights using a dataset from Kaggle. You explored the average loan amount by country using aggregated bar charts, box plots, and violin plots. You also nested the data by gender, allowing you to draw additional insights from your charts. Congratulations!

How do you feel?

```
In [ ]: print("How amazing is Data Science?!!")
```