Data Visualization

Estimated time needed: **30** minutes

In this lab, you will learn how to visualize and interpret data

Objectives

- Import Libraries
- Lab Exercises
  - Identifying duplicates
  - Plotting Scatterplots
  - Plotting Boxplots

Import Libraries

All Libraries required for this lab are listed below. The libraries pre-installed on Skills Network Labs are commented. If you run this notebook in a different environment, e.g. your desktop, you may need to uncomment and install certain libraries.

In [ ]:

```
#install specific version of libraries used in lab
#! mamba install pandas==1.3.3
#! mamba install numpy=1.21.2
#! mamba install scipy=1.7.1-y
#!  mamba install seaborn=0.9.0-y
#!  mamba install matplotlib=3.4.3-y
```

Import the libraries we need for the lab

In [1]:

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

Read in the csv file from the url using the request library

In [2]:

```
ratings_url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-ST0151EN-SkillsNetwork/labs/teachingratings.csv'
ratings_df = pd.read_csv(ratings_url)
```

Lab Exercises

Identify all duplicate cases using prof. Using all observations, find the average and standard deviation for age. Repeat the analysis by first filtering the data set to include one observation for each instructor with a total number of observations restricted to 94.

Identify all duplicate cases using prof variable - find the unique values of the prof variables

In [3]:

```
ratings_df.prof.unique()
```

Out[3]:

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36,
       37, 38, 39, 41, 42, 43, 44, 45, 46, 48, 49, 50, 51, 52, 53, 54, 55,
       56, 57, 58, 59, 60, 63, 64, 65, 66, 67, 68, 70, 71, 72, 73, 74, 75,
       76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
       93, 94, 22, 30, 40, 47, 61, 62, 69])
```

Print out the number of unique values in the prof variable

In [4]:

```
ratings_df.prof.nunique()
```

Out[4]:

94

Using all observations, Find the average and standard deviation for age

In [5]:

```
ratings_df['age'].mean()
```

Out[5]:

48.365010799136066

In [6]:

```
ratings_df['age'].std()
```

Out[6]:

9.80274203786482

Repeat the analysis by first filtering the data set to include one observation for each instructor with a total number of observations restricted to 94.

> first we drop duplicates using prof as a subset and assign it a new dataframe name called no_duplicates_ratings_df

In [7]:

```
no_duplicates_ratings_df = ratings_df.drop_duplicates(subset =['prof'])
no_duplicates_ratings_df.head()
```

Out[7]:

	minority	age	gender	credits	beauty	eval	division	native	tenure	students	allstu
0	yes	36	female	more	0.289916	4.3	upper	yes	yes	24	
4	no	59	male	more	-0.737732	4.5	upper	yes	yes	17	
7	no	51	male	more	-0.571984	3.7	upper	yes	yes	55	
9	no	40	female	more	-0.677963	4.3	upper	yes	yes	40	
17	no	31	female	more	1.509794	4.4	upper	yes	yes	42	

> Use the new dataset to get the mean of age

In [8]:

```
no_duplicates_ratings_df['age'].mean()
```

Out[8]:

```
47.5531914893617
```

In [9]:

```
no_duplicates_ratings_df['age'].std()
```

Out[9]:

```
10.25651329515495
```

Using a bar chart, demonstrate if instructors teaching lower-division courses receive higher average teaching evaluations.

In [31]:

```
ratings_df.head(25)
```

Out[31]:

	minority	age	gender	credits	beauty	eval	division	native	tenure	students	allstu
0	yes	36	female	more	0.289916	4.3	upper	yes	yes	24	
1	yes	36	female	more	0.289916	3.7	upper	yes	yes	86	
2	yes	36	female	more	0.289916	3.6	upper	yes	yes	76	
3	yes	36	female	more	0.289916	4.4	upper	yes	yes	77	
4	no	59	male	more	-0.737732	4.5	upper	yes	yes	17	
5	no	59	male	more	-0.737732	4.0	upper	yes	yes	35	
6	no	59	male	more	-0.737732	2.1	upper	yes	yes	39	
7	no	51	male	more	-0.571984	3.7	upper	yes	yes	55	
8	no	51	male	more	-0.571984	3.2	upper	yes	yes	111	
9	no	40	female	more	-0.677963	4.3	upper	yes	yes	40	
10	no	40	female	more	-0.677963	3.5	upper	yes	yes	24	
11	no	40	female	more	-0.677963	4.1	upper	yes	yes	24	
12	no	40	female	more	-0.677963	4.6	upper	yes	yes	17	
13	no	40	female	more	-0.677963	3.8	upper	yes	yes	14	
14	no	40	female	more	-0.677963	3.8	upper	yes	yes	37	
15	no	40	female	more	-0.677963	3.8	upper	yes	yes	18	
16	no	40	female	more	-0.677963	4.2	upper	yes	yes	15	
17	no	31	female	more	1.509794	4.4	upper	yes	yes	42	
18	no	31	female	more	1.509794	3.9	upper	yes	yes	40	
19	no	31	female	more	1.509794	4.5	upper	yes	yes	38	
20	no	31	female	more	1.509794	4.5	upper	yes	yes	40	
21	no	31	female	more	1.509794	4.4	upper	yes	yes	52	
22	no	31	female	more	1.509794	4.4	upper	yes	yes	49	
23	no	62	male	more	0.588569	4.2	upper	yes	yes	182	
24	no	62	male	more	0.588569	4.4	upper	yes	yes	160	

Find the average teaching evaluation in both groups of upper and lower-division

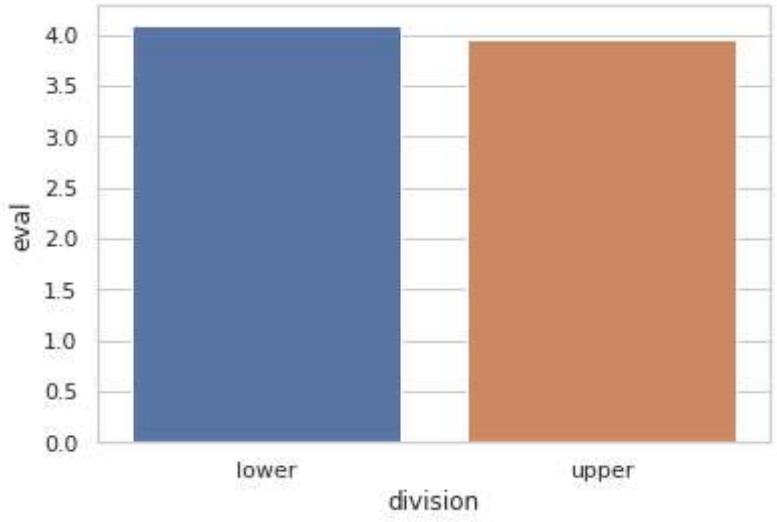
In [11]:

```
division_eval = ratings_df.groupby('division')[['eval']].mean().reset_index()
```

Plot the barplot using the seaborn library

In [16]:

```
sns.set(style="whitegrid")
ax = sns.barplot(x="division", y="eval", data=division_eval)
#ax = sns.barplot(x="division", y="eval", data=ratings_df)
```

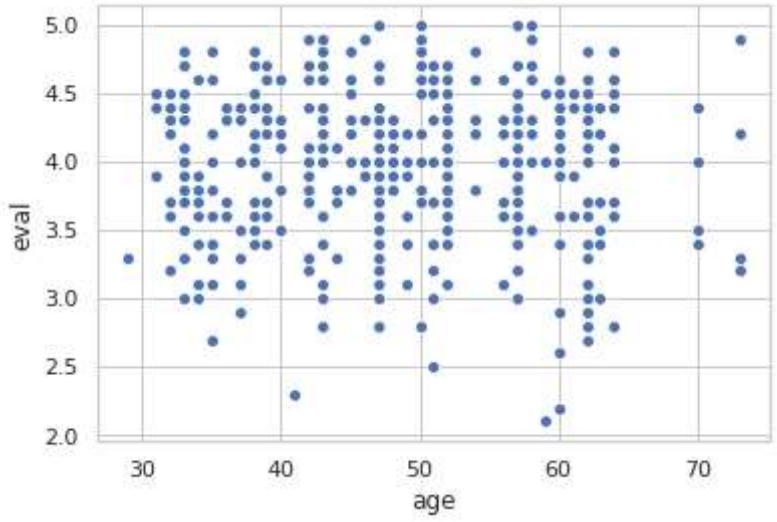


Plot the relationship between age and teaching evaluation scores.

Create a scatterplot with the scatterplot function in the seaborn library

In [17]:

```
ax = sns.scatterplot(x='age', y='eval', data=ratings_df)
```

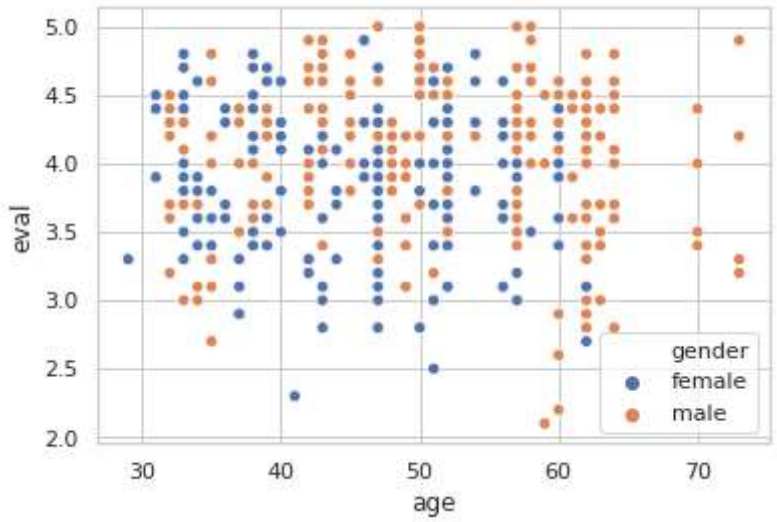


Using gender-differentiated scatter plots, plot the relationship between age and teaching evaluation scores.

Create a scatterplot with the scatterplot function in the seaborn library this time add the hue argument

In [18]:

```
ax = sns.scatterplot(x='age', y='eval', hue='gender',
                     data=ratings_df)
```

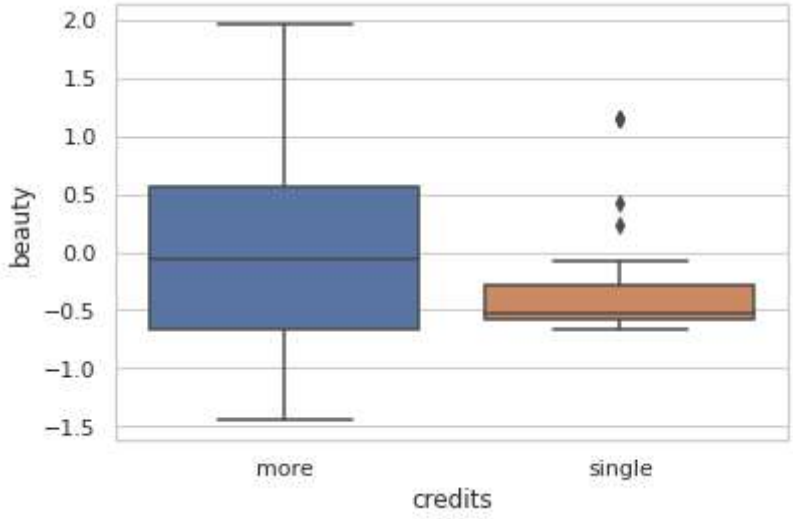


Create a box plot for beauty scores differentiated by credits.

We use the `boxplot()` function from the seaborn library

In [19]:

```
ax = sns.boxplot(x='credits', y='beauty', data=ratings_df)
```



What is the number of courses taught by gender?

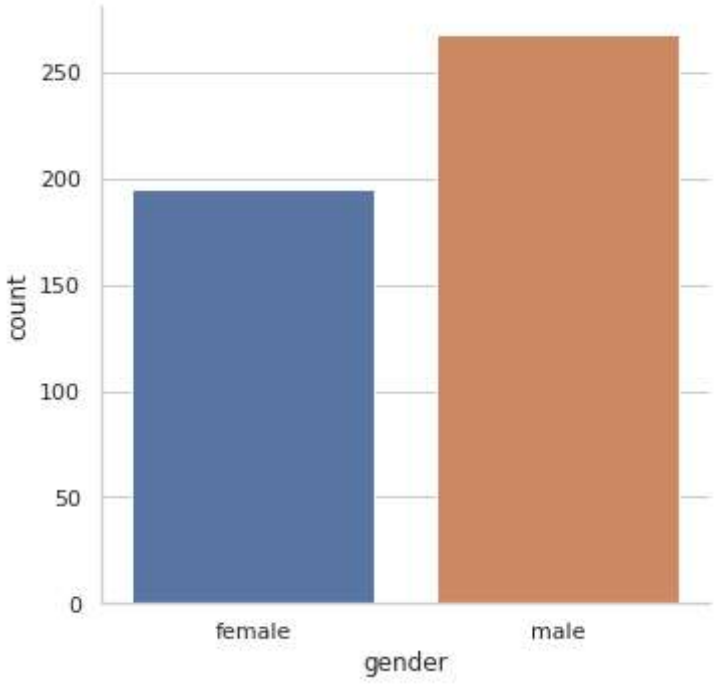
We use the `catplot()` function from the seaborn library

In [20]:

```
sns.catplot(x='gender', kind='count', data=ratings_df)
```

Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x7f27cccc1650>
```



Create a group histogram of taught by gender and tenure

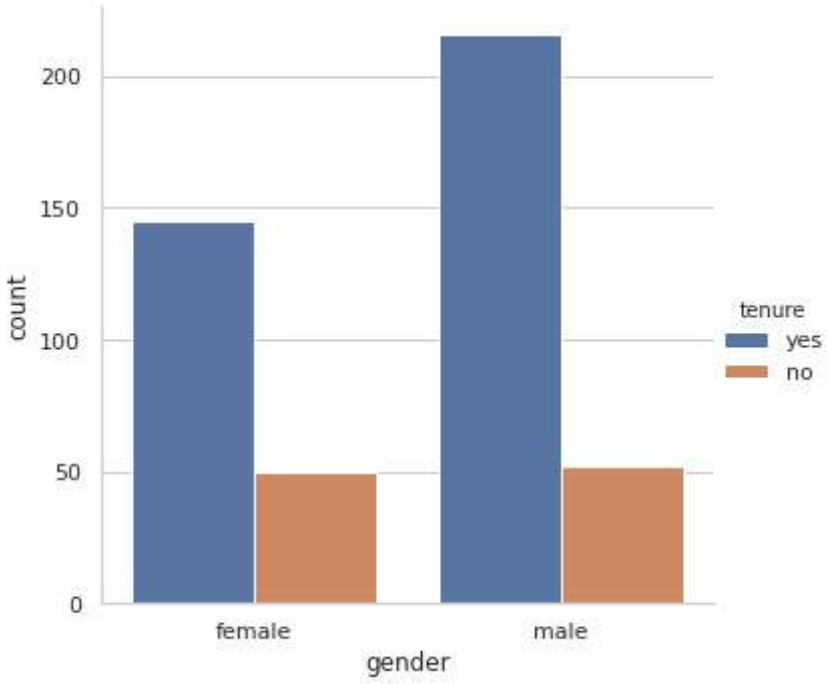
We will add the `hue = Tenure` argument

In [21]:

```
sns.catplot(x='gender', hue = 'tenure', kind='count', data=ratings_df)
```

Out[21]:

```
<seaborn.axisgrid.FacetGrid at 0x7f27cca14b50>
```



Add division as another factor to the above histogram

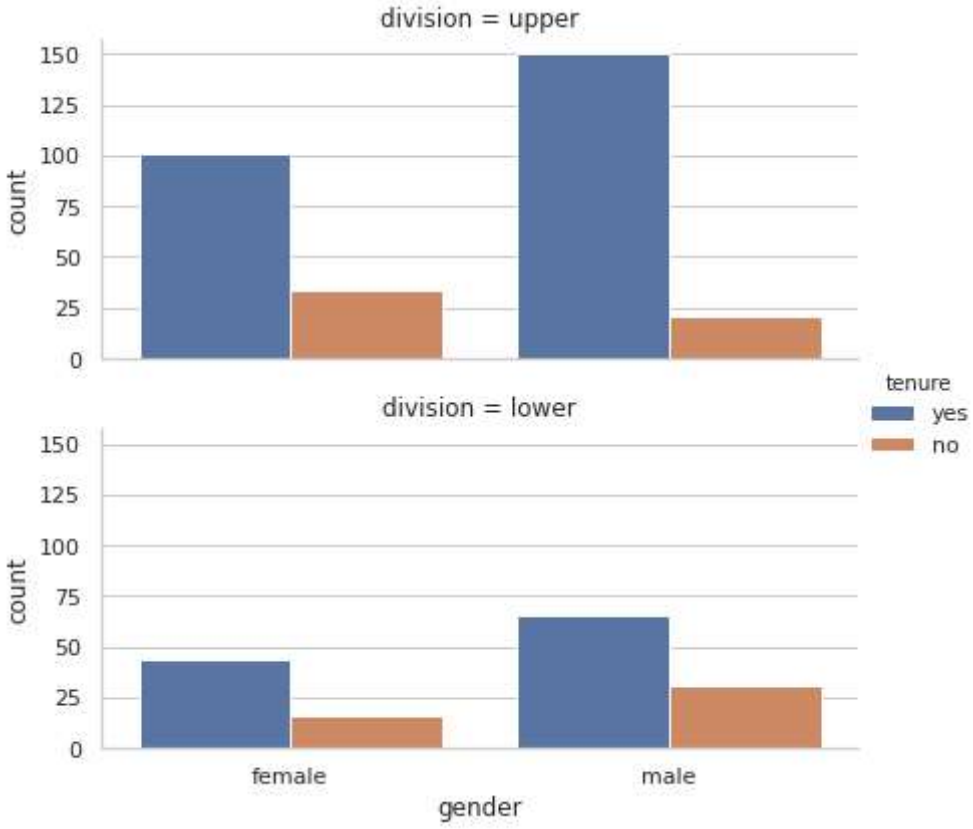
We add another argument named `row` and use the division variable as the row

In [22]:

```
sns.catplot(x='gender', hue = 'tenure', row = 'division',
            kind='count', data=ratings_df,
            height = 3, aspect = 2)
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x7f27cca14b90>
```



Create a scatterplot of age and evaluation scores, differentiated by gender and tenure

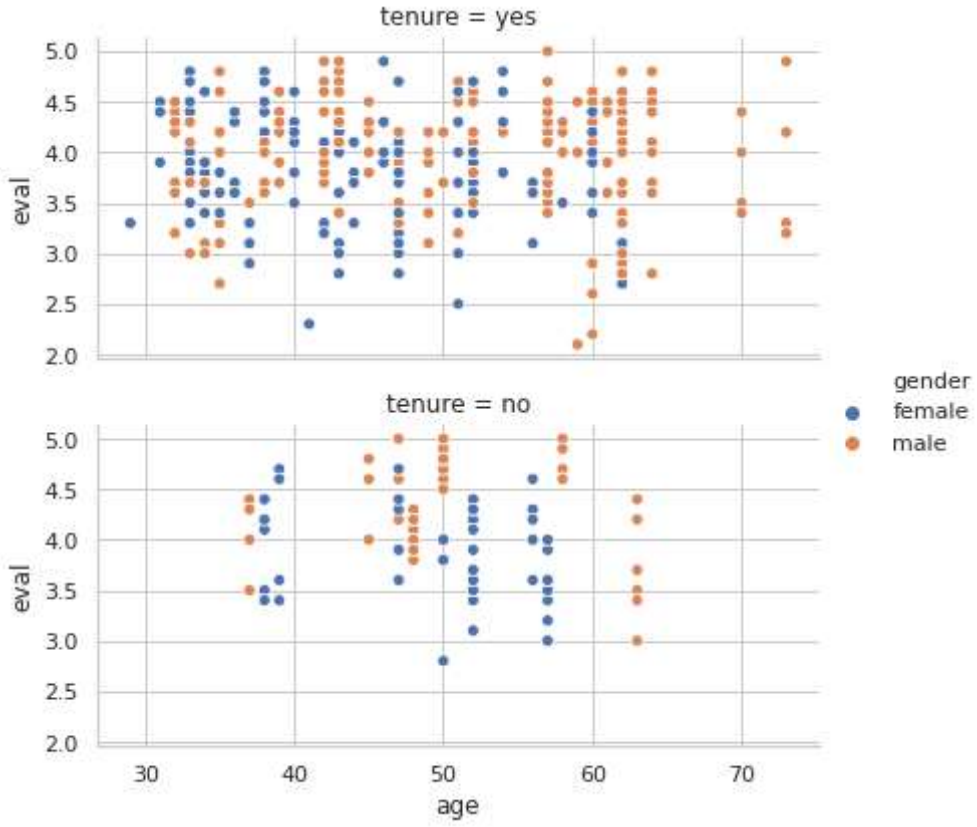
Use the `relplot()` function for complex scatter plots

In [23]:

```
sns.relplot(x="age", y="eval", hue="gender",
            row="tenure",
            data=ratings_df, height = 3, aspect = 2)
```

Out[23]:

```
<seaborn.axisgrid.FacetGrid at 0x7f27ccd3e050>
```

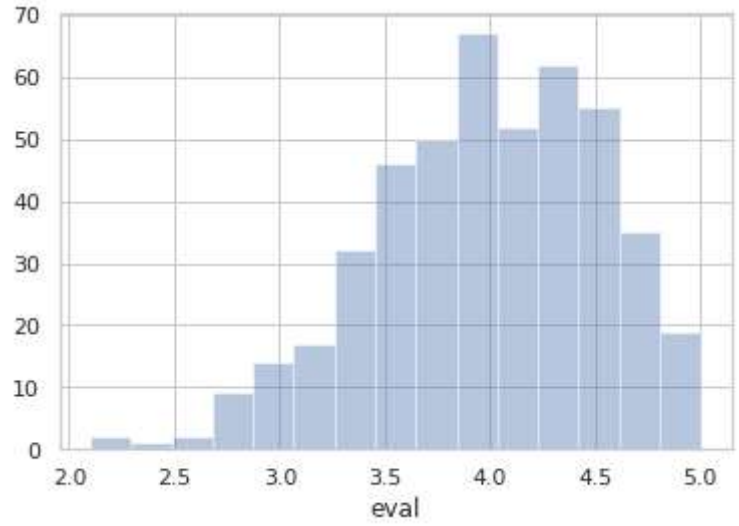


Create a distribution plot of teaching evaluation scores

We use the `distplot()` function from the seaborn library, set `kde = false` because we don'e need the curve

In [26]:

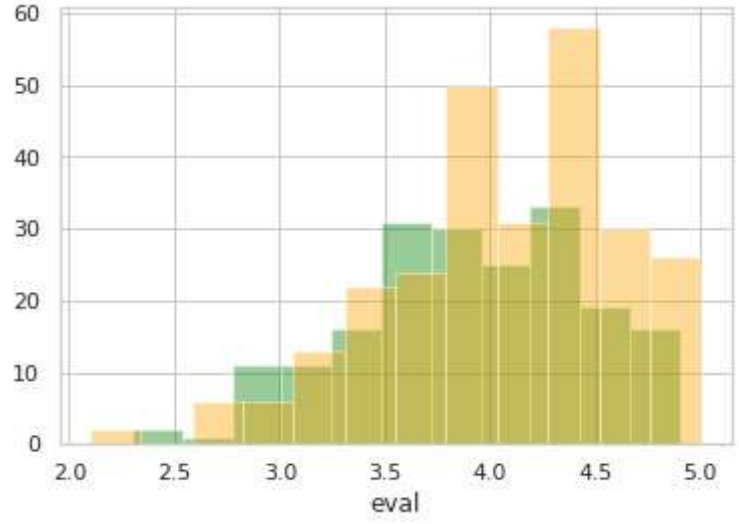
```
ax = sns.distplot(ratings_df['eval'], kde = False)
```



Create a distribution plot of teaching evaluation score with gender as a factor

In [27]:

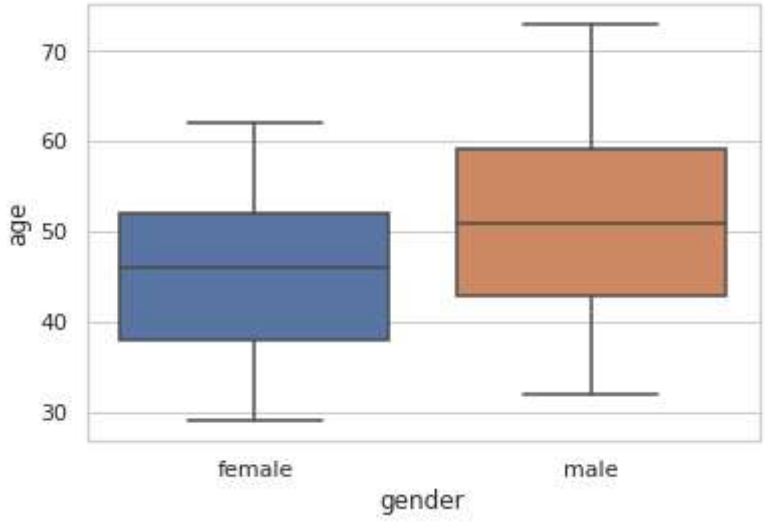
```
## use the distplot function from the seaborn library
sns.distplot(ratings_df[ratings_df['gender'] == 'female']['eval'], color='green', kde=F
alse)
sns.distplot(ratings_df[ratings_df['gender'] == 'male']['eval'], color="orange", kde=Fa
lse)
plt.show()
```



Create a box plot - age of the instructor by gender

In [28]:

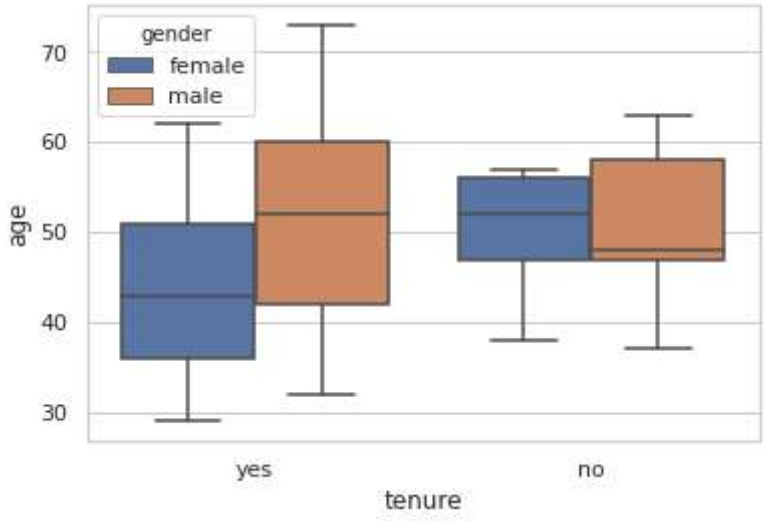
```
ax = sns.boxplot(x="gender", y="age", data=ratings_df)
```



Compare age along with tenure and gender

In [29]:

```
ax = sns.boxplot(x="tenure", y="age", hue="gender",
                 data=ratings_df)
```



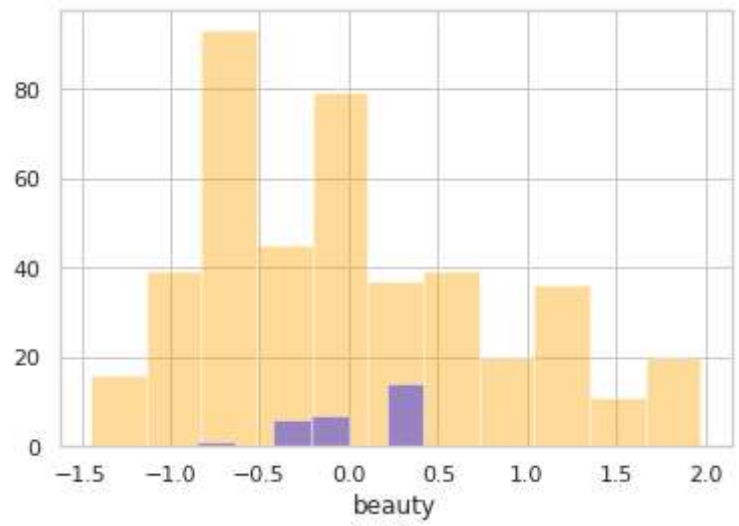
Practice Questions

Question 1: Create a distribution plot of beauty scores with Native English speaker as a factor

- Make the color of the native English speakers plot - orange and non - native English speakers - blue

In [33]:

```
## insert code
sns.distplot(ratings_df[ratings_df['native'] == 'yes']['beauty'], color='orange', kde=False)
sns.distplot(ratings_df[ratings_df['native'] == 'no']['beauty'], color="blue", kde=False)
plt.show()
```

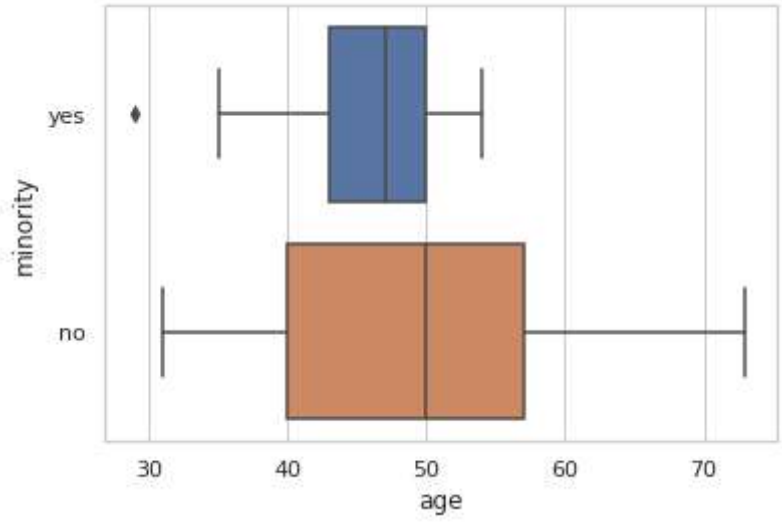


Double-click **here** for the solution.

Question 2: Create a Horizontal box plot of the age of the instructors by visible minority

In [35]:

```
## insert code
ax = sns.boxplot(y="minority", x="age", data=ratings_df)
```



Double-click **here** for a hint.

Double-click **here** for the solution.

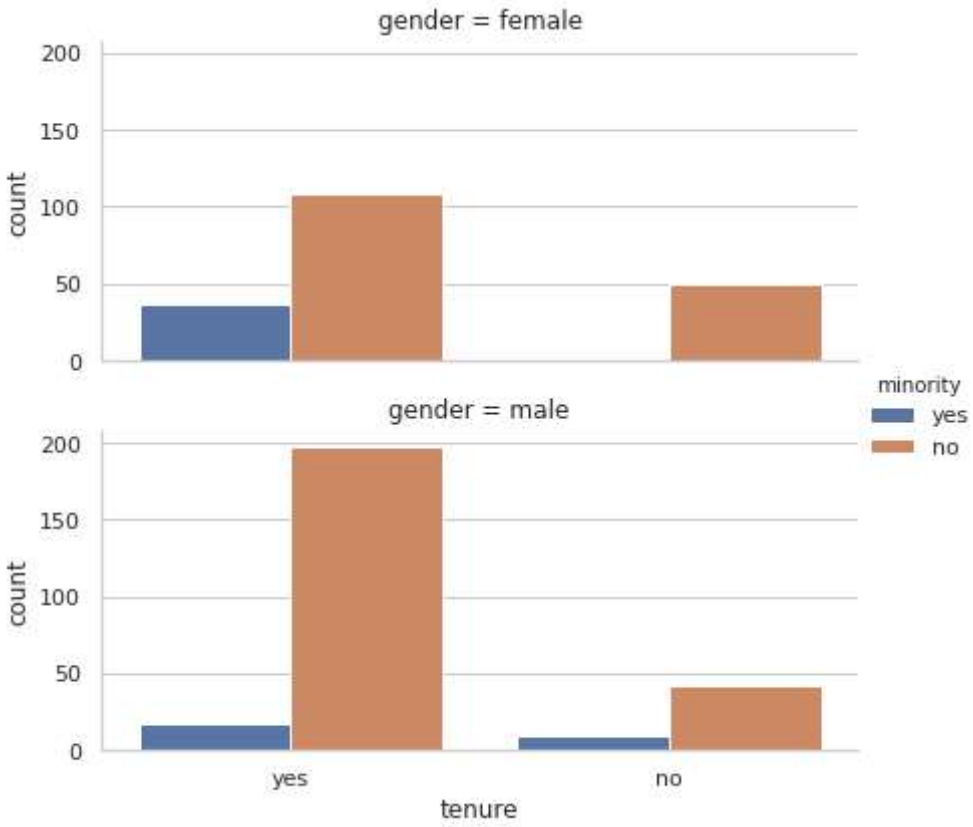
Question 3: Create a group histogram of tenure by minority and add the gender factor

In [36]:

```
## insert code
sns.catplot(x='tenure', hue = 'minority', row = 'gender',
            kind='count', data=ratings_df,
            height = 3, aspect = 2)
```

Out[36]:

```
<seaborn.axisgrid.FacetGrid at 0x7f27c5c88c50>
```

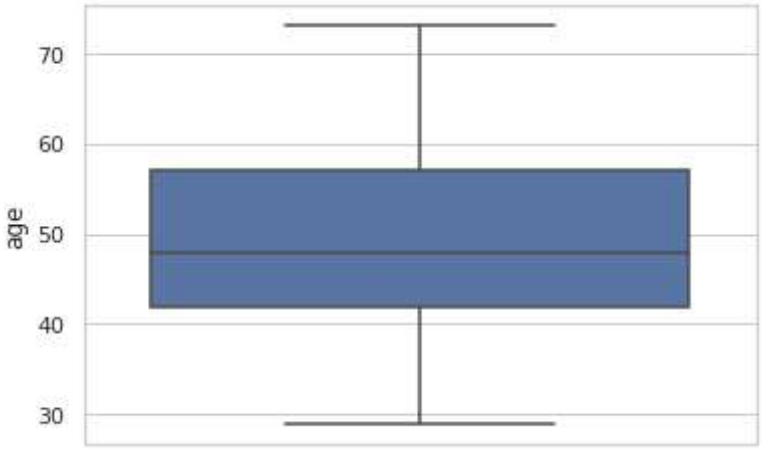


Double-click **here** for the solution.

Question 4: Create a boxplot of the age variable

In [40]:

```
## insert code
ax = sns.boxplot(y='age', data = ratings_df)
```



Double-click **here** for the solution.

Authors

Aije Egwaikhide (https://www.linkedin.com/in/aije-egwaikhide/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkST0151ENSkillsNetwork20531532-2021-01-01) is a Data Scientist at IBM who holds a degree in Economics and Statistics from the University of Manitoba and a Post-grad in Business Analytics from St. Lawrence College, Kingston. She is a current employee of IBM where she started as a Junior Data Scientist at the Global Business Services (GBS) in 2018. Her main role was making meaning out of data for their Oil and Gas clients through basic statistics and advanced Machine Learning algorithms. The highlight of her time in GBS was creating a customized end-to-end Machine learning and Statistics solution on optimizing operations in the Oil and Gas wells. She moved to the Cognitive Systems Group as a Senior Data Scientist where she will be providing the team with actionable insights using Data Science techniques and further improve processes through building machine learning solutions. She recently joined the IBM Developer Skills Network group where she brings her real-world experience to the courses she creates.

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-08-14	0.1	Aije Egwaikhide	Created the initial version of the lab