2024-1

Multivariate Data Analysis

Youjin Shin
(yj.shinn@catholic.ac.kr)

Lab: Titanic Dataset

```
# import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import warnings



# ignore warnings
warnings.filterwarnings('ignore')
```

Lab: Titanic Dataset

```
### Data Load (titanic) ###
df_titanic = sns.load_dataset('titanic')


### Understanding data ###


print(df_titanic.head())
print(df_titanic.info())
print(df_titanic.describe())
```

```
C:\Users\user\anaconda3\envs\Class_MDA\python.exe
C:\Users\user\PycharmProjects\MDA\MDA_02.py
   survived  pclass    sex   age ... deck  embark_town  alive  alone
0         0       3   male  22.0 ...  NaN  Southampton     no  False
1         1       1 female  38.0 ...    C    Cherbourg    yes  False
2         1       3 female  26.0 ...  NaN  Southampton    yes   True
3         1       1 female  35.0 ...    C  Southampton    yes  False
4   0     3   male  35.0 ...  NaN  Southampton     no   True
[5 rows x 15 columns]
```

Lab: Titanic Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
```

Lab: Titanic Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
```

1. **survived** Survival or not? 0 means death, 1 means survival.
2. **pclass** Room class 1 is grade 1, 2 is grade 2, 3 is grade 3
3. **sex** gender Male is male, female is female.
4. **age** age
5. **sibsp** Number of siblings and spouses traveling together
6. **parch** Number of children and parents traveling together
7. **Fare** fare
8. **embarked** First letter of boarding destination name C is for Cherbourg, Q is Queenstown, S is Southampton
9. **class** Room class First grade 1st grade, Second grade 2nd grade, Third grade 3rd grade
10. **who** man, woman, child
11. **adult_male** Whether you are an adult male If True, it is an adult male, if False, it is anyone else.
12. **deck** Cabin number first alphabet A, B, C, D, E, F, G
13. **embark_town** boarding point name Cherbourg, Queenstown, Southampton
14. **alive** Survival If no, you die, if yes, you survive.
15. **alone** Whether you boarded alone If True, board alone, if False, board with family.

Lab: Titanic Dataset

```
### Data Load (titanic) ###
df_titanic = sns.load_dataset('titanic')


### Understanding data ###


print(df_titanic.head())
print(df_titanic.info())
print(df_titanic.describe())
```

```
       survived      pclass         age       sibsp       parch        fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

Lab: Titanic Dataset

```
# DataFrame[column name].value_counts()
print(df_titanic['survived'].value_counts())     # 0:Dead, 1:survived
print(df_titanic['sex'].value_counts())     # male/female
print(df_titanic['age'].value_counts())
print(df_titanic['fare'].value_counts())
print(df_titanic['class'].value_counts())              # First, second, third
print(df_titanic['who'].value_counts())          # man, woman, child
```

```
survived
0    549
1    342
Name: count, dtype: int64

sex
male      577
female    314
Name: count, dtype: int64
```

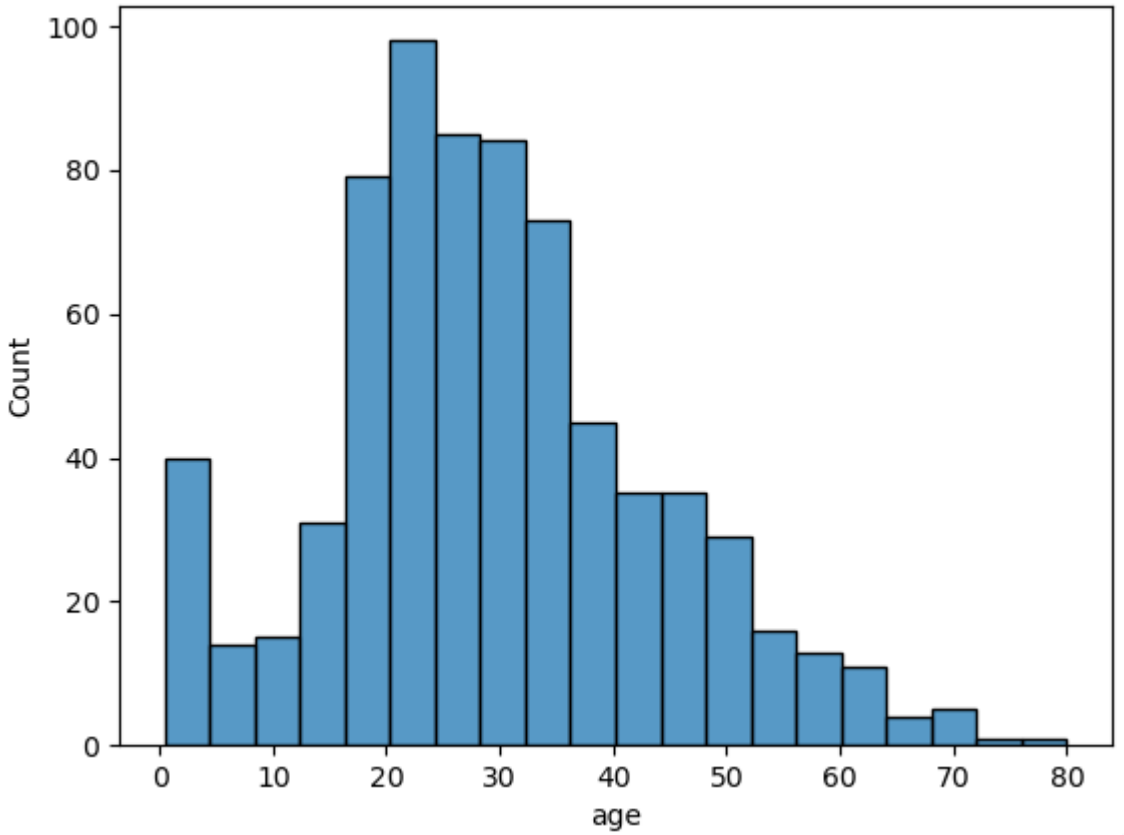
```
age
24.00    30
22.00    27
18.00    26
..
0.92      1
23.50     1
74.00     1
Name: count, Length: 88, dtype: int64
```

```
fare
8.0500     43
13.0000    42
..
14.0000     1
10.5167     1
Name: count, Length: 248, dtype: int64
```

```
class
Third     491
First     216
Second    184
Name: count, dtype: int64

who
man      537
woman    271
child     83
Name: count, dtype: int64
```

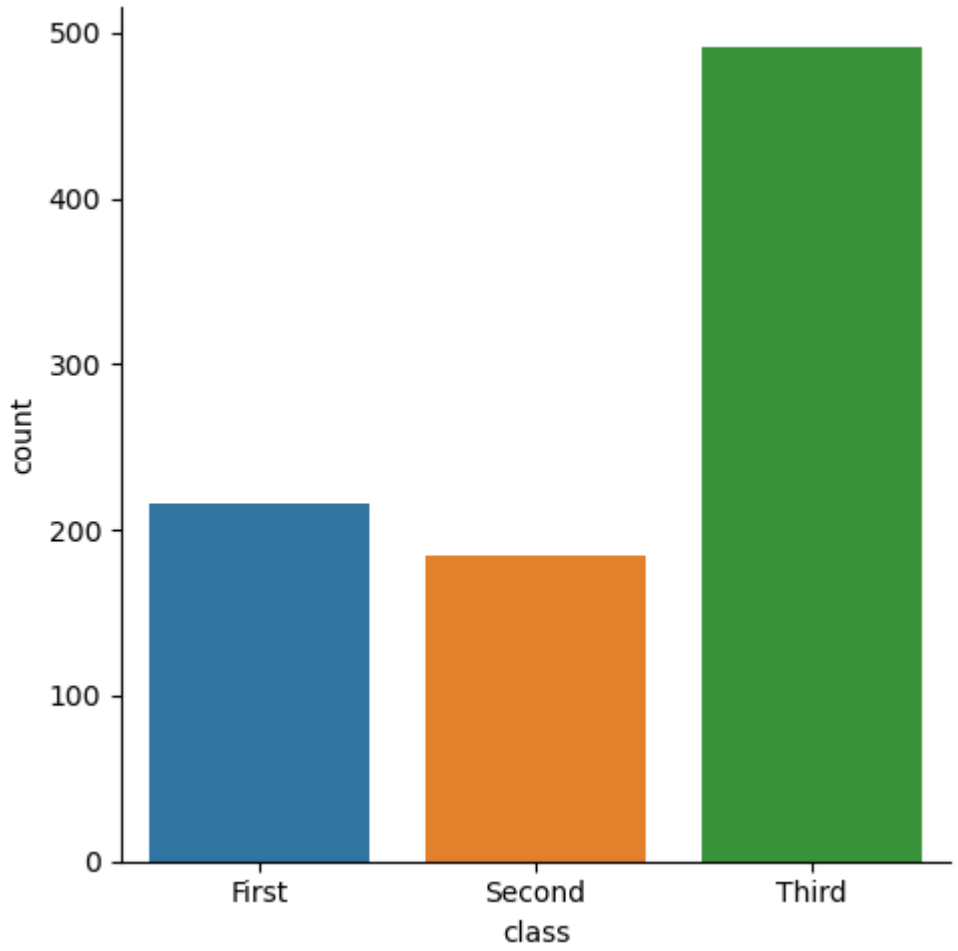
Lab: Titanic Dataset

```
########### Visualization ###########
# (1) Distribution of Age
sns.histplot(data=df_titanic, x='age')
plt.show()
```



Lab: Titanic Dataset

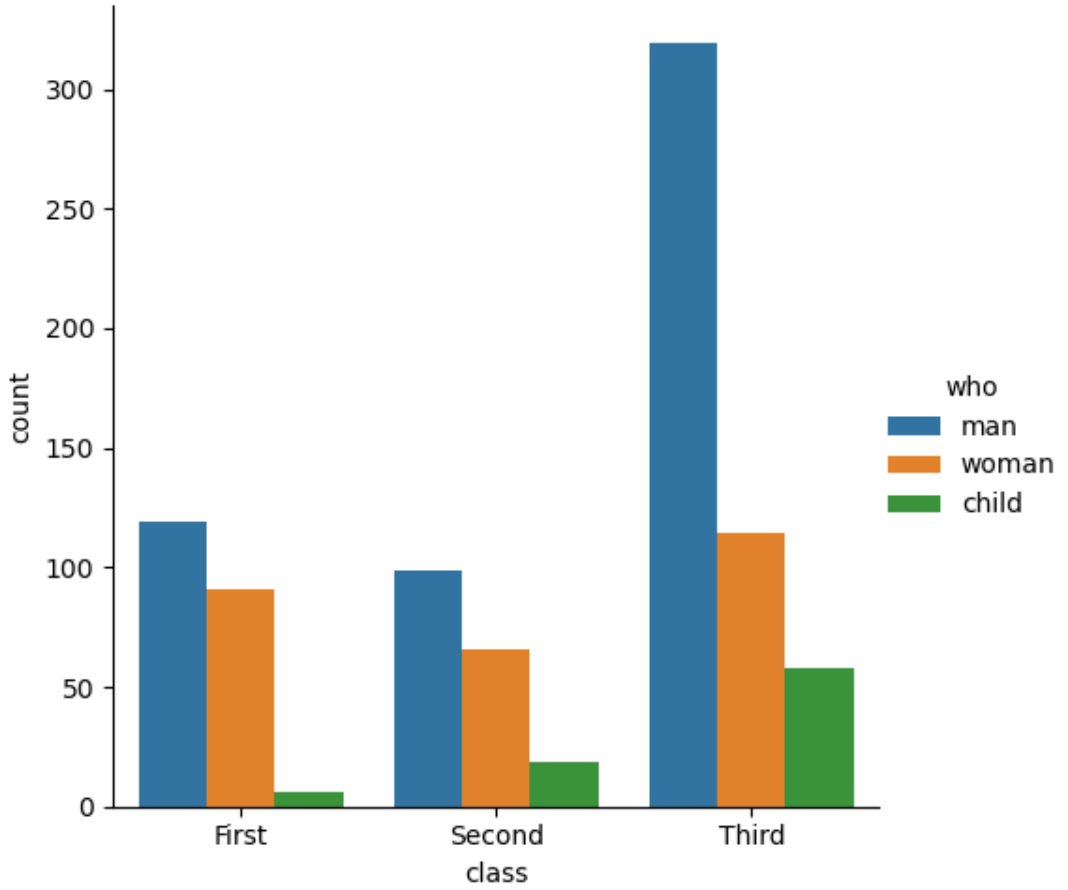
```
########### Visualization ###########
# (2-1) Number of people in each room class
sns.catplot(x='class' ,kind='count', data=df_titanic)
plt.show()
```



Lab: Titanic Dataset

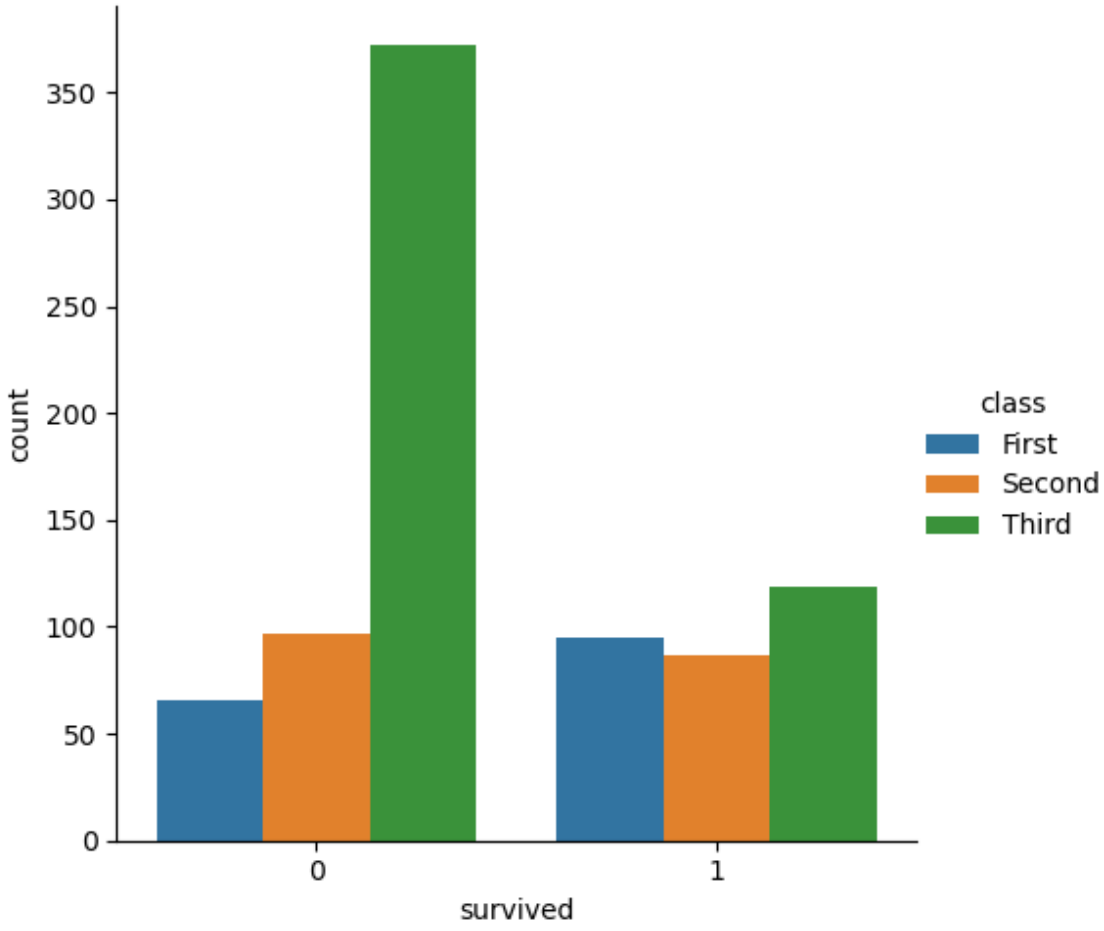
```
########## Visualization ##########
# (2-2) Number of men, women, and children by room class
sns.catplot(x='class', hue='who', kind='count', data=df_titanic)
plt.show()
```

* The 'hue' is a categorical factor that further subdivides the categories.



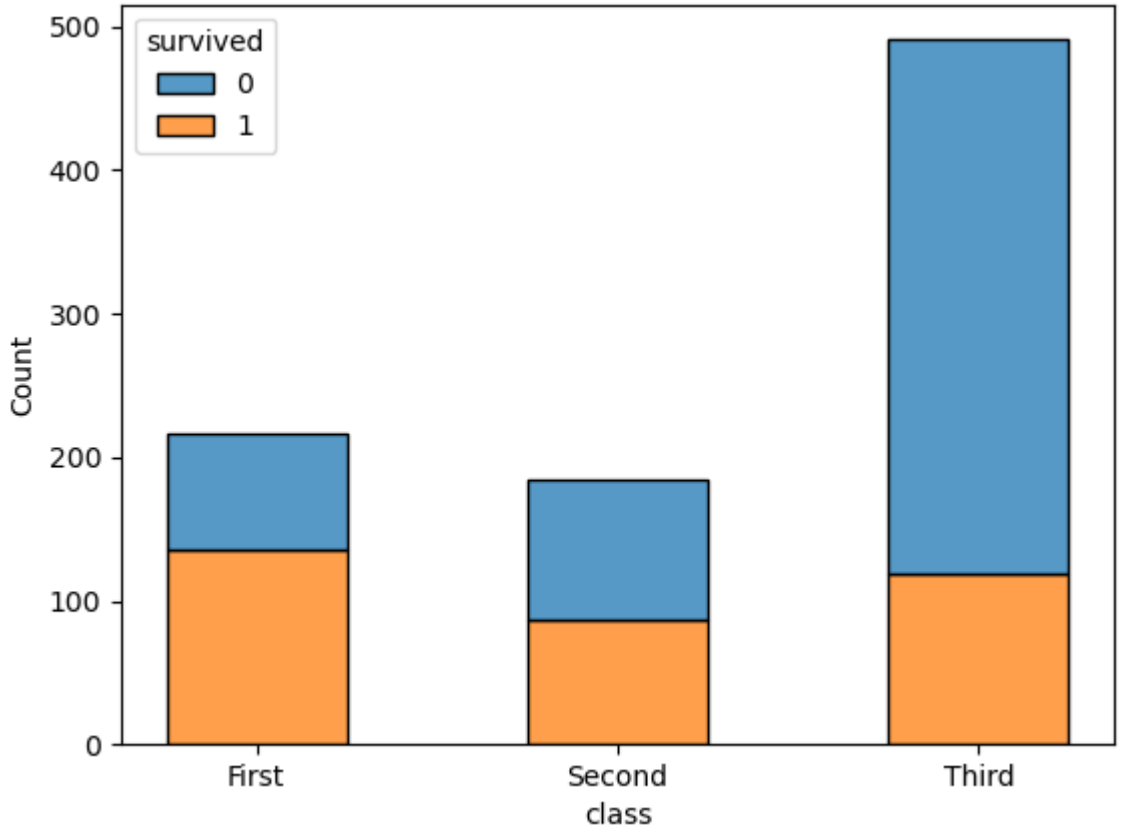
Lab: Titanic Dataset

```
########### Visualization ###########
# (2-3) Number of people in each room class by survival status
sns.catplot(x='survived', hue='class', kind='count', data=df_titanic)
plt.show()
```



Lab: Titanic Dataset

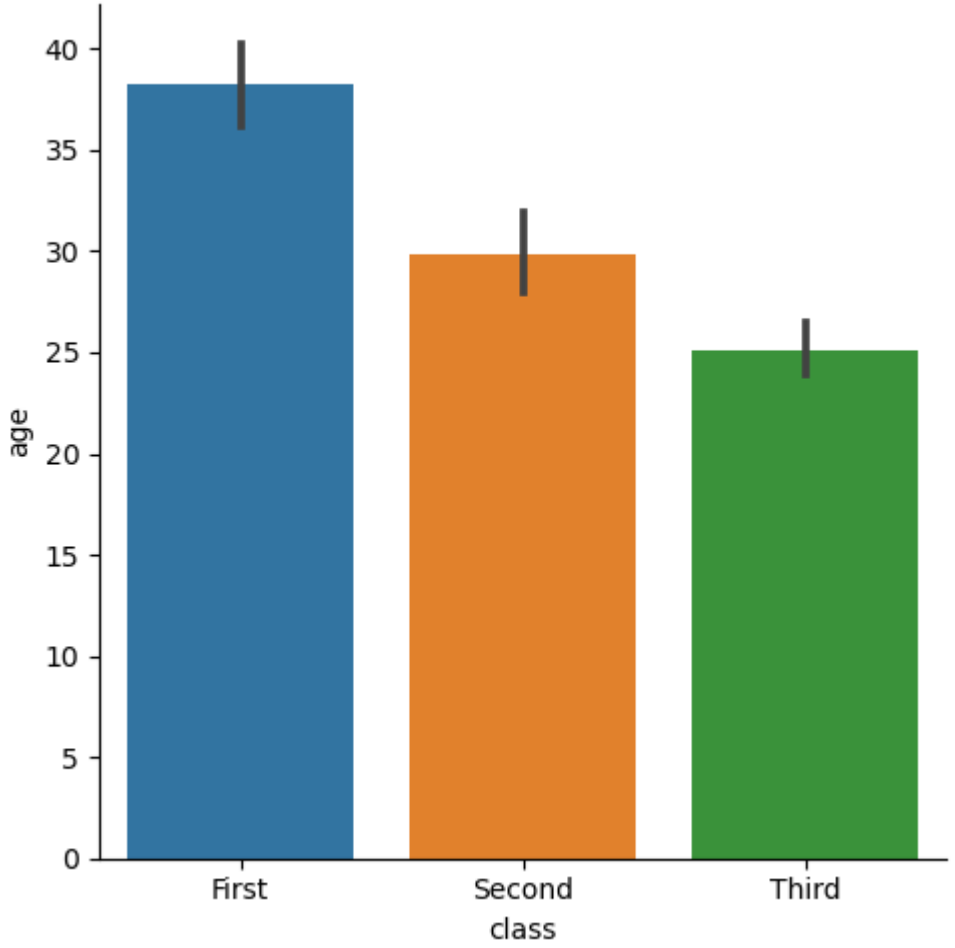
```
########## Visualization ##########
# (2-4) Number of survivors and dead by room class
sns.histplot(x='class', hue='survived', multiple='stack', shrink=.5, data=df_titanic)
plt.show()
```



Lab: Titanic Dataset

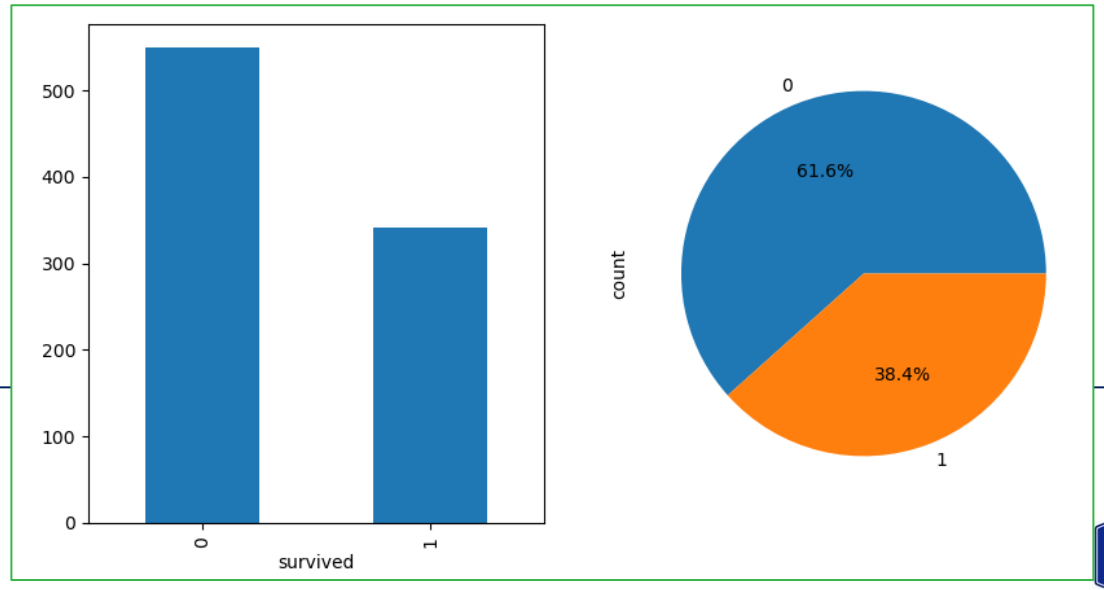
```
########## Visualization ##########
# (3) Average age and deviation by room class
sns.catplot(x='class', y='age', kind='bar', data=df_titanic)
```

* The average is expressed as the height of the bar, and the deviation is expressed as an error bar.



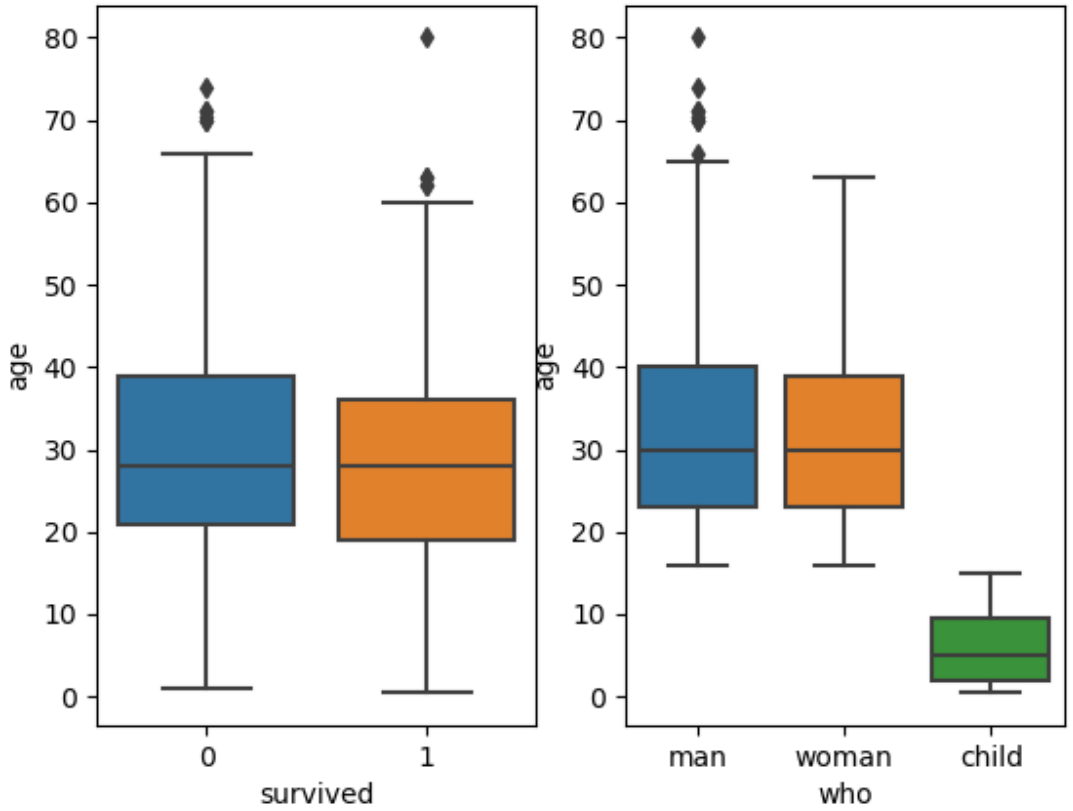
Lab: Titanic Dataset

```
# ########## Visualization ##########
# (4) Proportion of Survivors and Dead  (bar vs. pie)
fig, axes = plt.subplots(ncols=2)
df_titanic["survived"].value_counts().plot(kind = "bar", ax=axes[0])
df_titanic["survived"].value_counts().plot(kind = "pie", autopct='%1.1f%%', ax=axes[1])
plt.show()
```



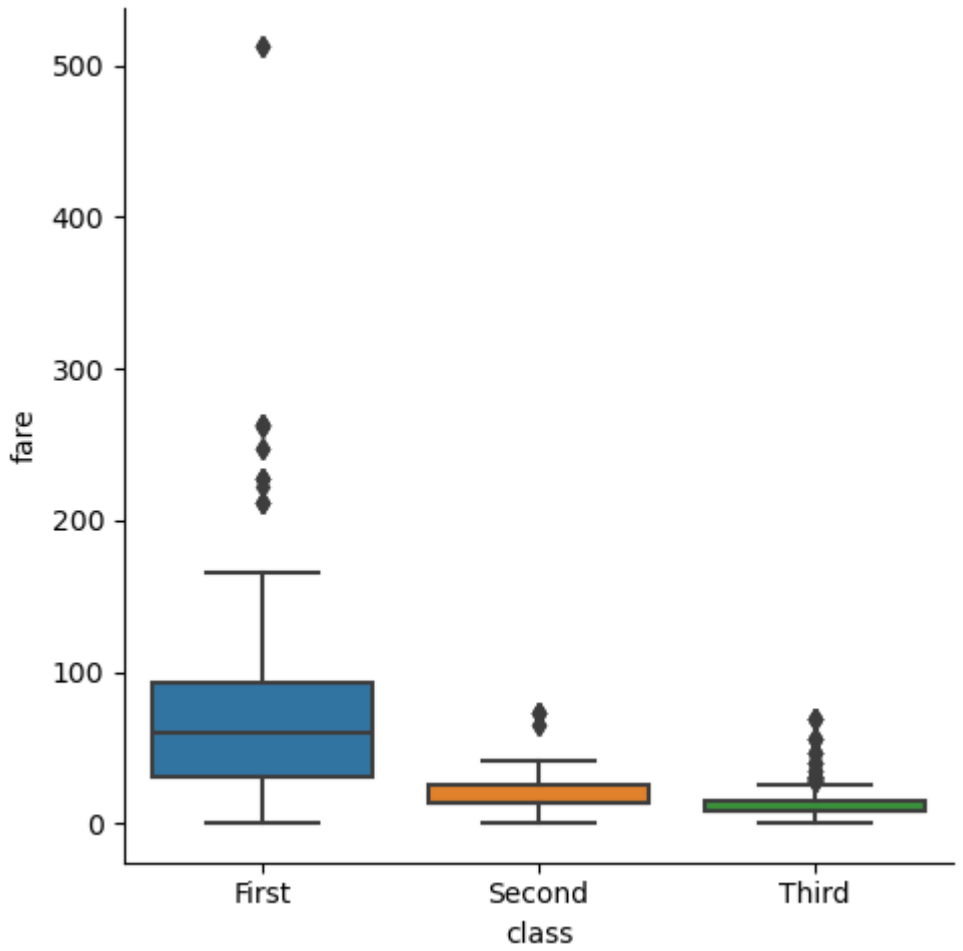
Lab: Titanic Dataset

```
########## Visualization ##########
# (5) boxplot
# (5-1) Age distribution of survivors and dead
fig, axes = plt.subplots(ncols=2)
sns.boxplot(x='survived',y='age',data=df_titanic, ax=axes[0])
# (5-2) Age distribution of men, women and children
sns.boxplot(x='who',y='age',data=df_titanic,ax=axes[1])
plt.show()



# sns.catplot(x='survived',y='age',kind='box',data=df_titanic)
# sns.catplot(x='who',y='age',kind='box',data=df_titanic)
```



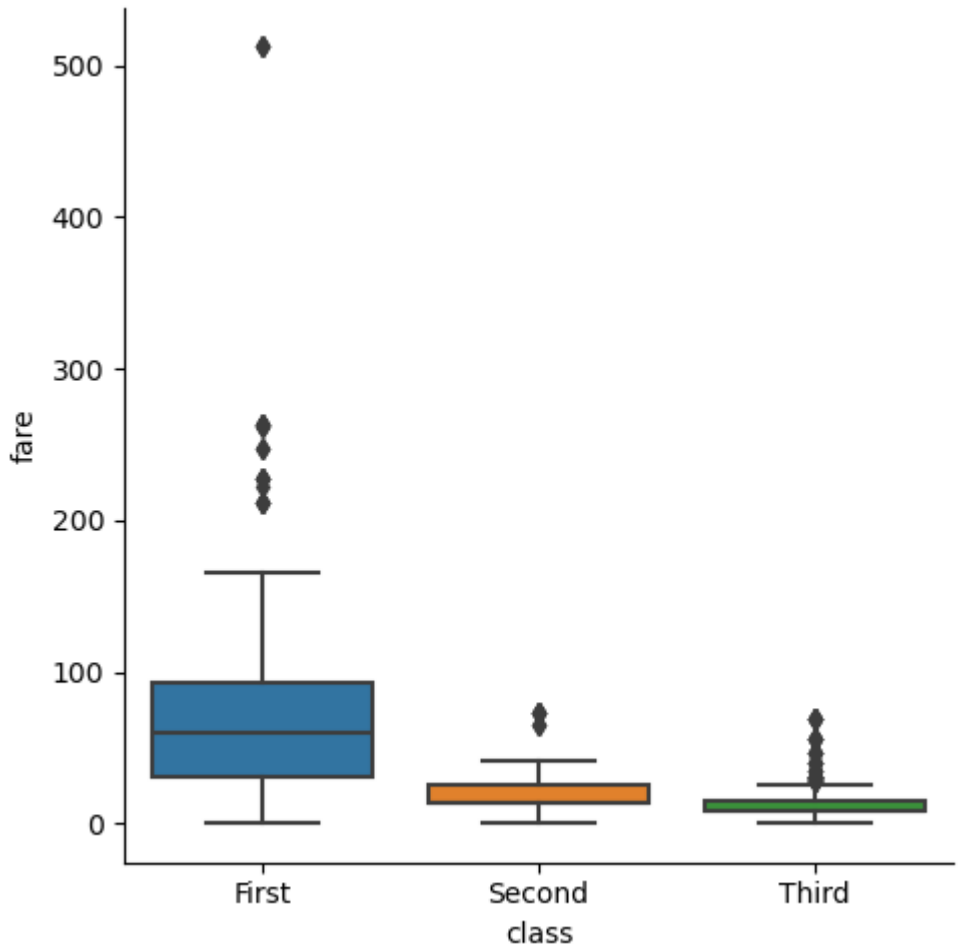
Lab: Titanic Dataset

```
########### Visualization ###########
# (5-3) Lab: Fare distribution according to room class
```



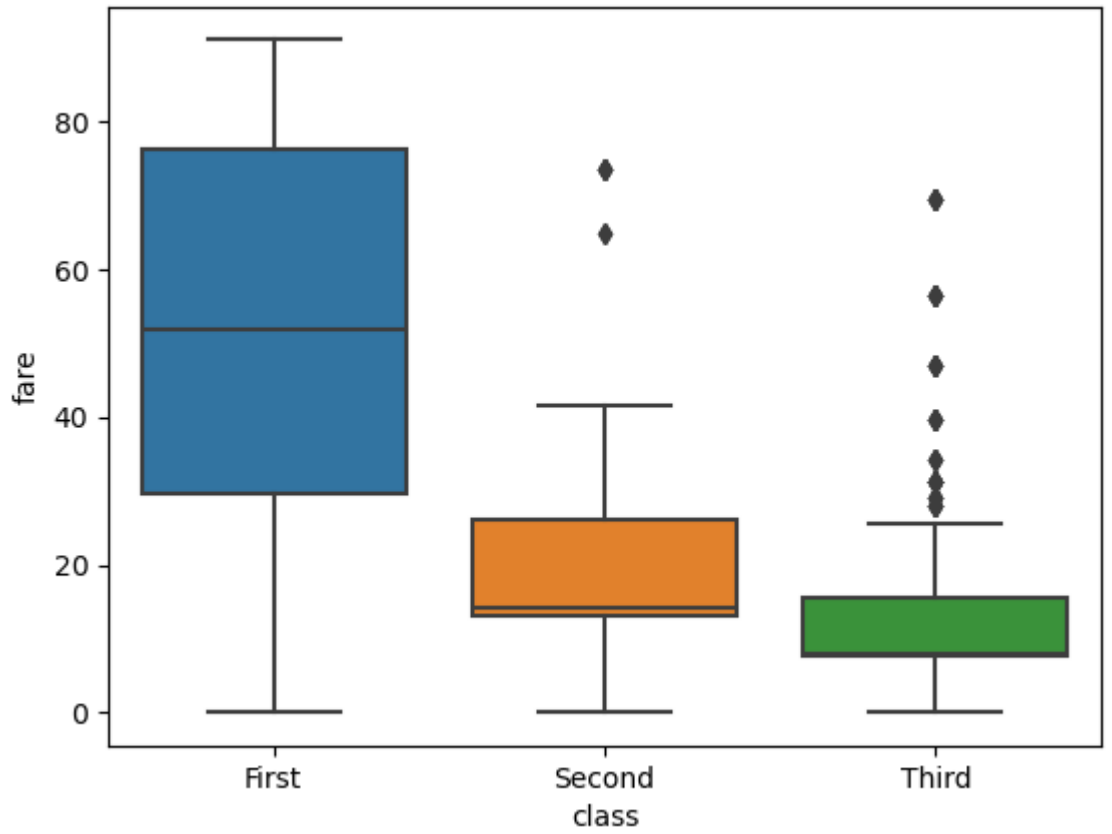
Lab: Titanic Dataset

```
########### Visualization ###########
# (5-3) Lab: Fare distribution according to room class
sns.catplot(x='class',y='fare',kind='box',data=df_titanic)
plt.show()
```



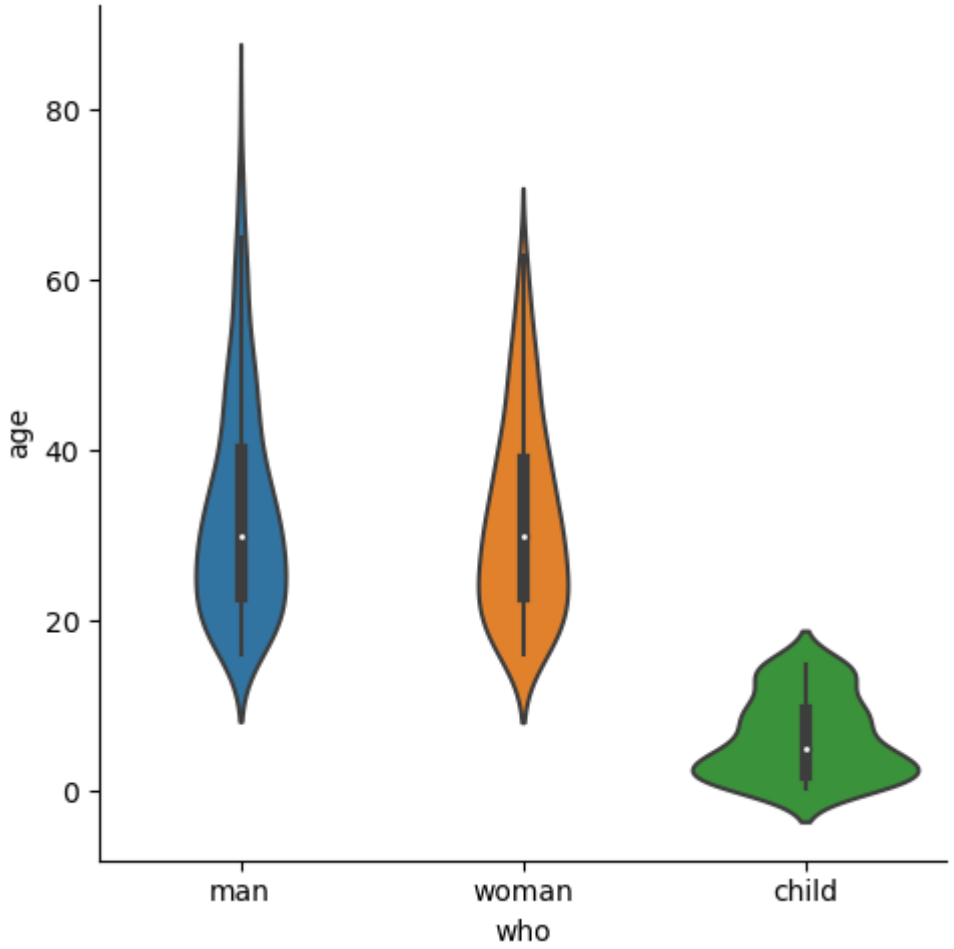
Lab: Titanic Dataset

```
########## Visualization ##########
# (5-4) Remove the outliers
fare_q3 = df_titanic['fare'].quantile(q=0.75)
fare_q1 = df_titanic['fare'].quantile(q=0.25)

fare_iqr = fare_q3 - fare_q1

condition = df_titanic['age'] <= 4*fare_iqr
new_df_titanic = df_titanic[condition]

sns.boxplot(data=new_df_titanic, x='survived', y='age')
plt.show()
```



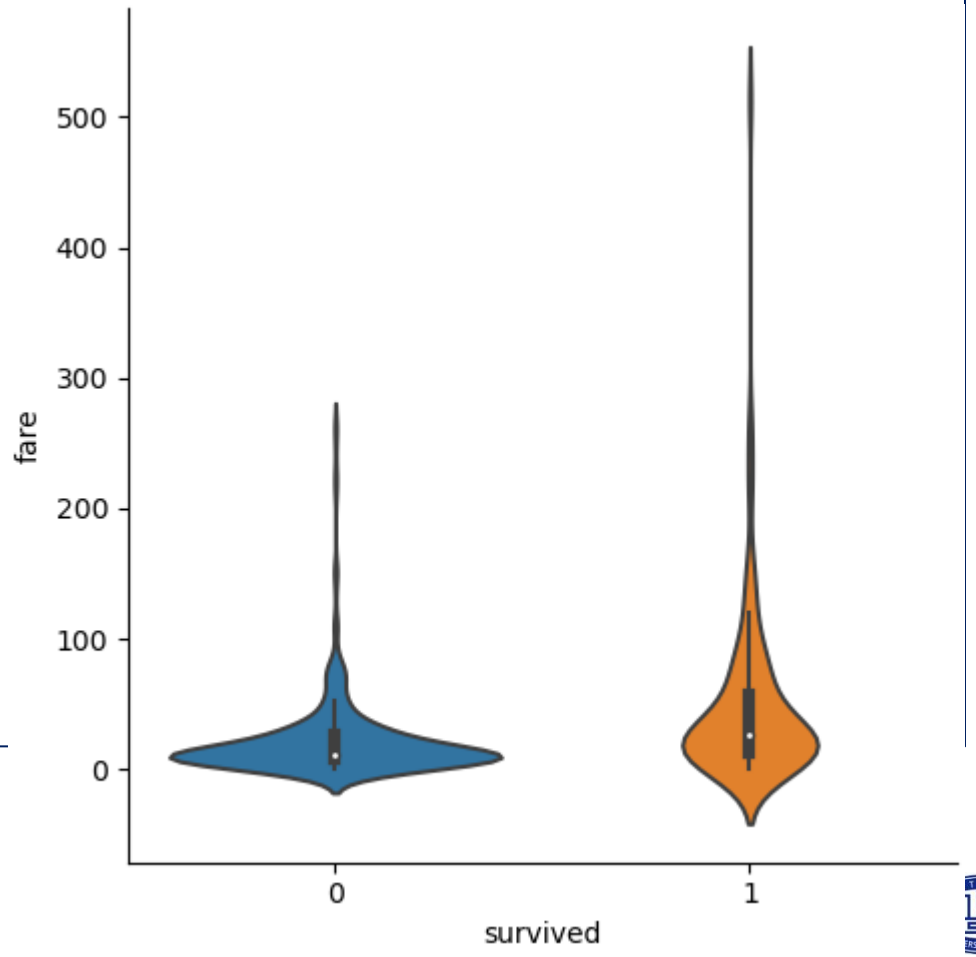
Lab: Titanic Dataset

```
########### Visualization ###########
# (6-1) Age distribution of men, women and children
sns.catplot(x='who',y='age',kind='violin',data=df_titanic)
plt.show()
```



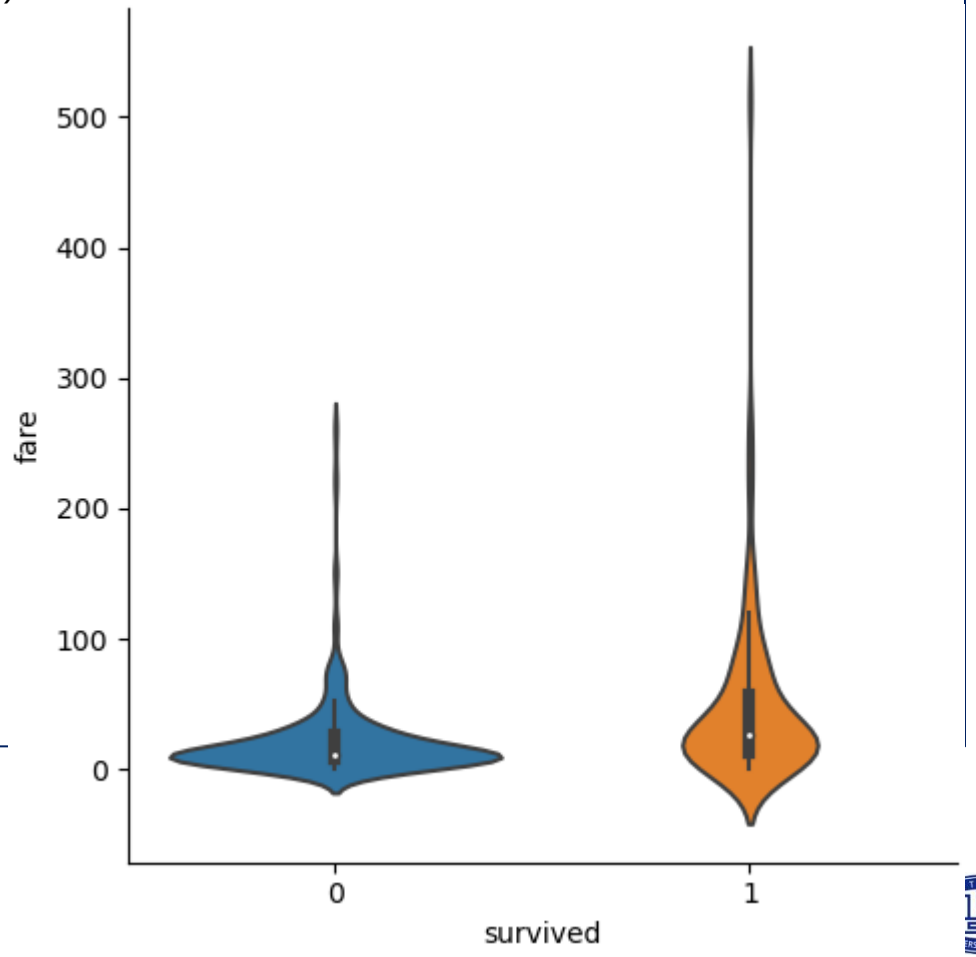
Lab: Titanic Dataset

```
########### Visualization ###########
# (6-2) lab: Fare distribution of survivors and dead
```



Lab: Titanic Dataset

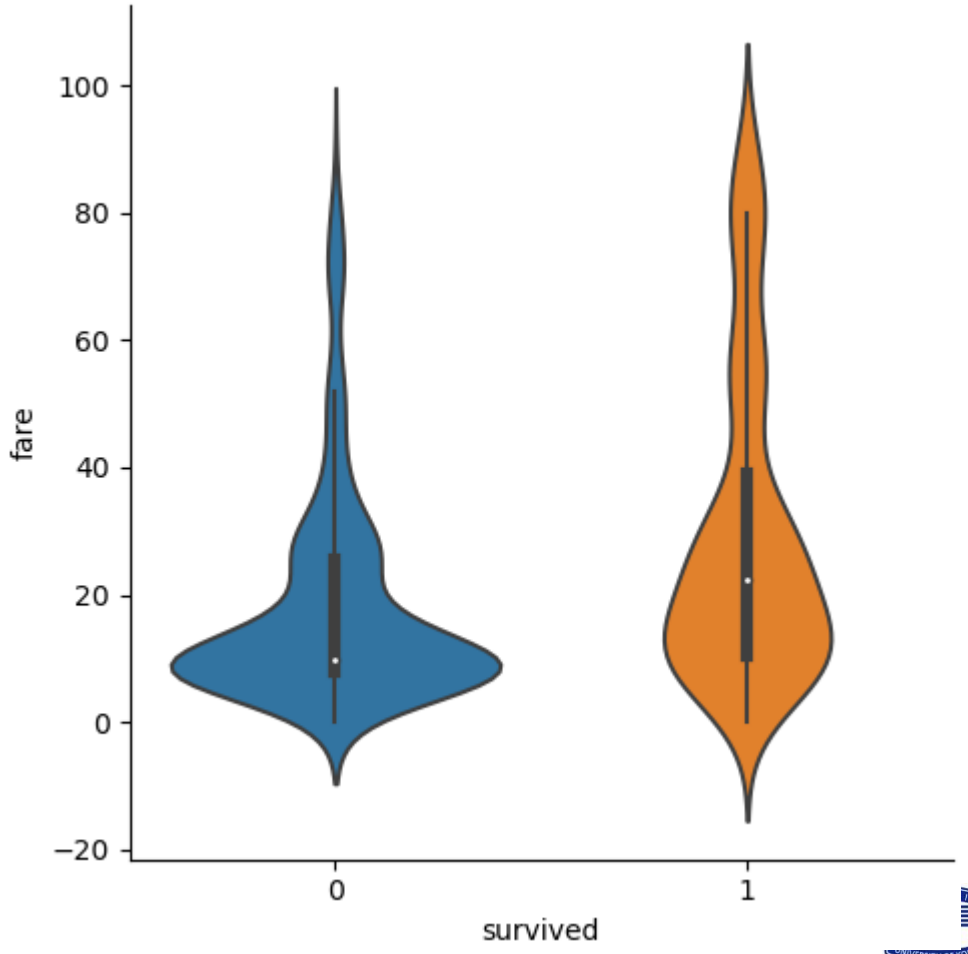
```
########### Visualization ###########
# (6-2) lab: Fare distribution of survivors and dead
sns.catplot(x='survived',y='fare',kind='violin',data=new_df_titanic)
plt.show()
```



Lab: Titanic Dataset

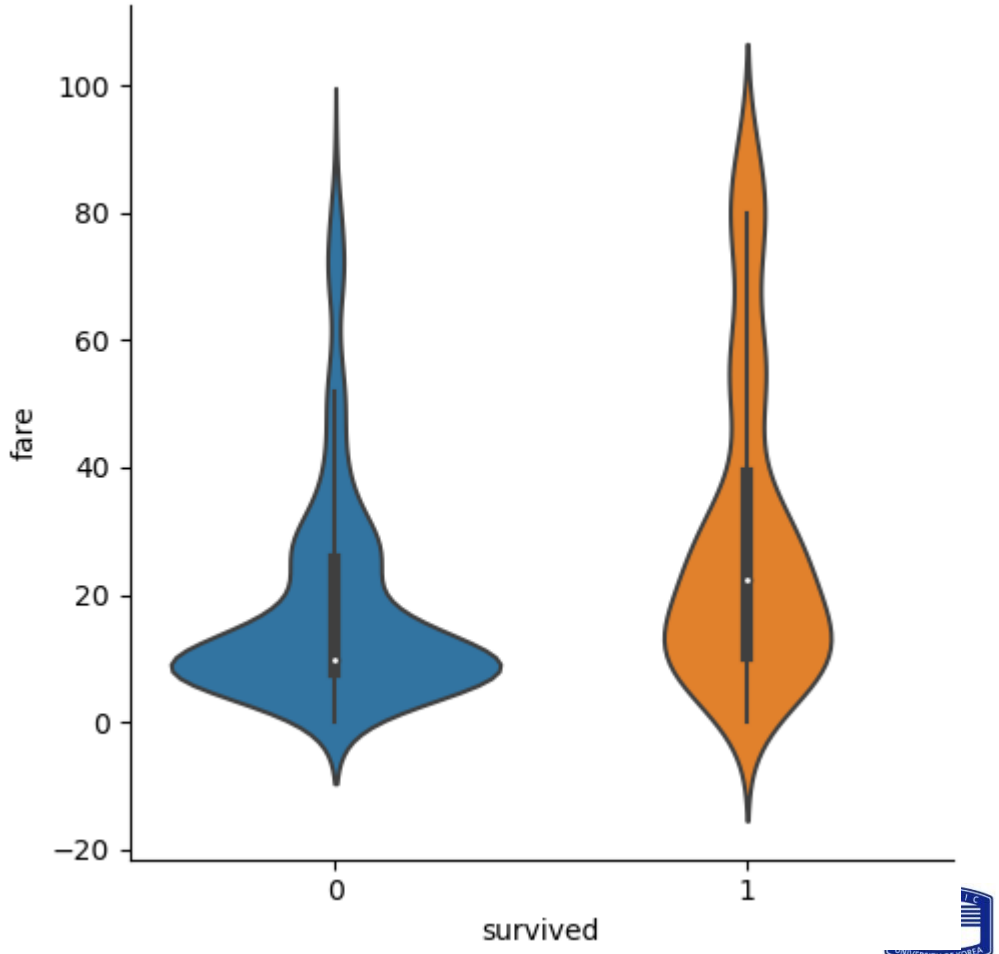
```
########## Visualization ##########
# (6-3) lab: Fare distribution of survivors and dead after removing outliers
```

```
sns.catplot(x='survived',y='fare',kind='violin',data=new_df_titanic)
plt.show()
```



Lab: Titanic Dataset

```
########## Visualization ##########
# (6-3) lab: Fare distribution of survivors and dead after removing outliers
fare_q3 = df_titanic['fare'].quantile(q=0.75)
fare_q1 = df_titanic['fare'].quantile(q=0.25)

fare_iqr = fare_q3 - fare_q1

condition = df_titanic['fare'] <= 4*fare_iqr
new_df_titanic = df_titanic[condition]

sns.catplot(x='survived',y='fare',kind='violin',data=new_df_titanic)
plt.show()
```



Lab: Titanic Dataset

```
########## Visualization ##########
# (7) correlation btw variables (heatmap)
# 'survived','sex','age','fare','class','who'
# remove 'sex' and 'who' (not continuous, just categorical wo the order)


cor_df_titanic=df_titanic[['survived','age','fare','class']]


print(cor_df_titanic.dtypes)
print(cor_df_titanic['class'].unique())
```

```
survived      int64
age         float64
fare        float64
class      category
```

```
['Third', 'First', 'Second']
Categories (3, object): ['First', 'Second', 'Third']
```

Lab: Titanic Dataset

```
########## Visualization ##########
# (7) correlation btw variables (heatmap)
# 'survived','sex','age','fare','class','who'
# remove 'sex' and 'who' (not continuous, just categorical wo the order)

cor_df_titanic=df_titanic[['survived','age','fare','class']]

print(cor_df_titanic.dtypes)
print(cor_df_titanic['class'].unique())

cor_df_titanic['class'] = cor_df_titanic['class'].map({'First': 3, 'Second': 2,'Third': 1,})
print(cor_df_titanic['class'].unique())
```

```
[1, 3, 2]
Categories (3, int64): [3, 2, 1]
```

Lab: Titanic Dataset

```
sns.heatmap(cor_df_titanic.corr(), cmap='coolwarm', vmax=1.0, vmin=-1.0, annot=True)
plt.show()
```

(CF) Only the first row of annotations displayed on seaborn heatmap

⇒ This issue was for Seaborn 0.12.2, Seaborn 0.13.0 solves the issue.

⇒ Or Update your anaconda

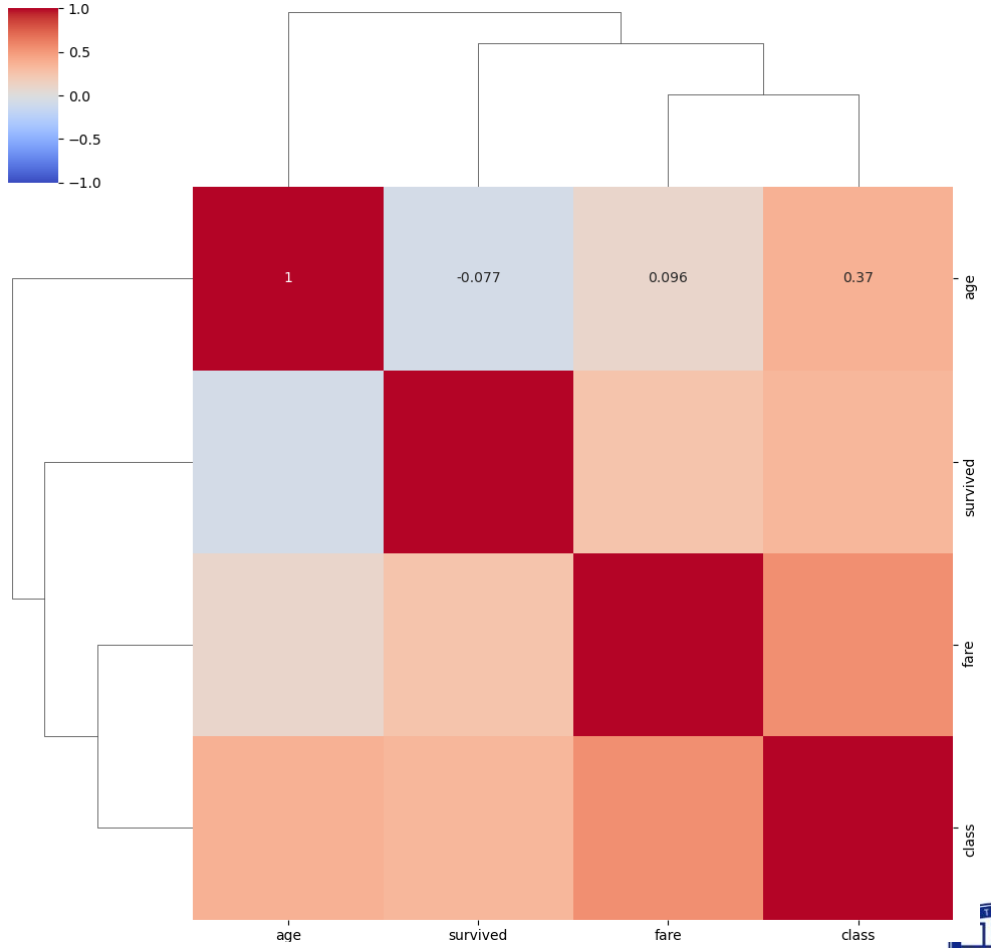
Lab: Titanic Dataset

```
print(cor_df_titanic.corr())
```

	survived	age	fare	class
survived	1.000000	-0.077221	0.257307	0.338481
age	-0.077221	1.000000	0.096067	0.369226
fare	0.257307	0.096067	1.000000	0.549500
class	0.338481	0.369226	0.549500	1.000000

Lab: Titanic Dataset

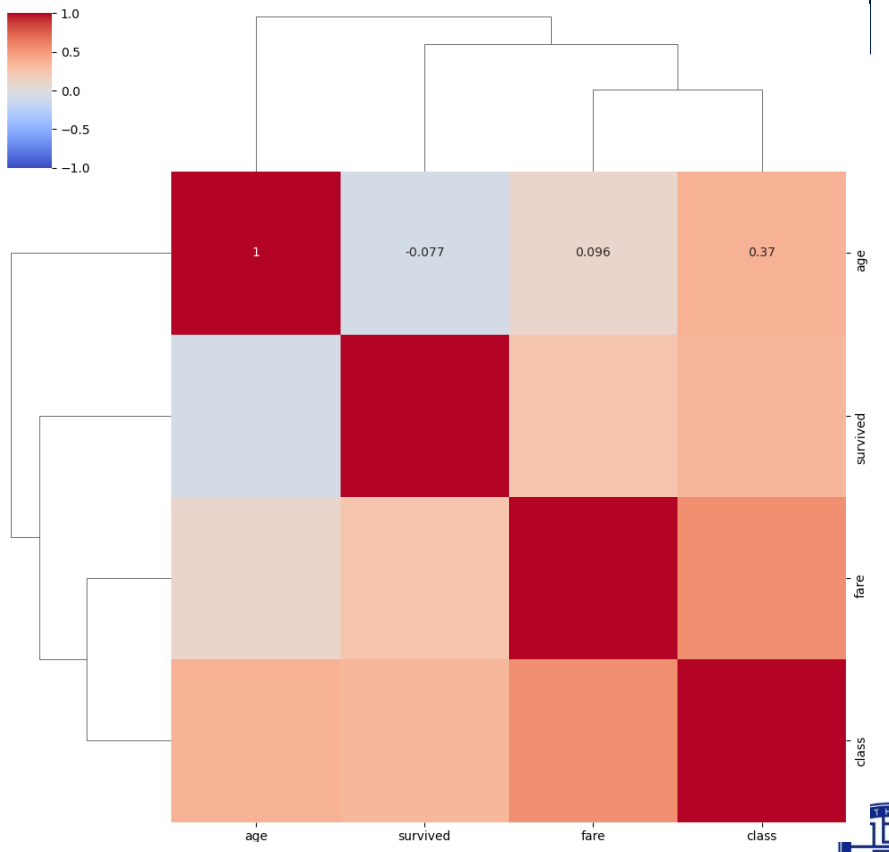
```
sns.clustermap(cor_df_titanic.corr(), cmap='coolwarm', vmax=1.0, vmin=-1.0, annot=True)
plt.show()
```



Lab: Titanic Dataset

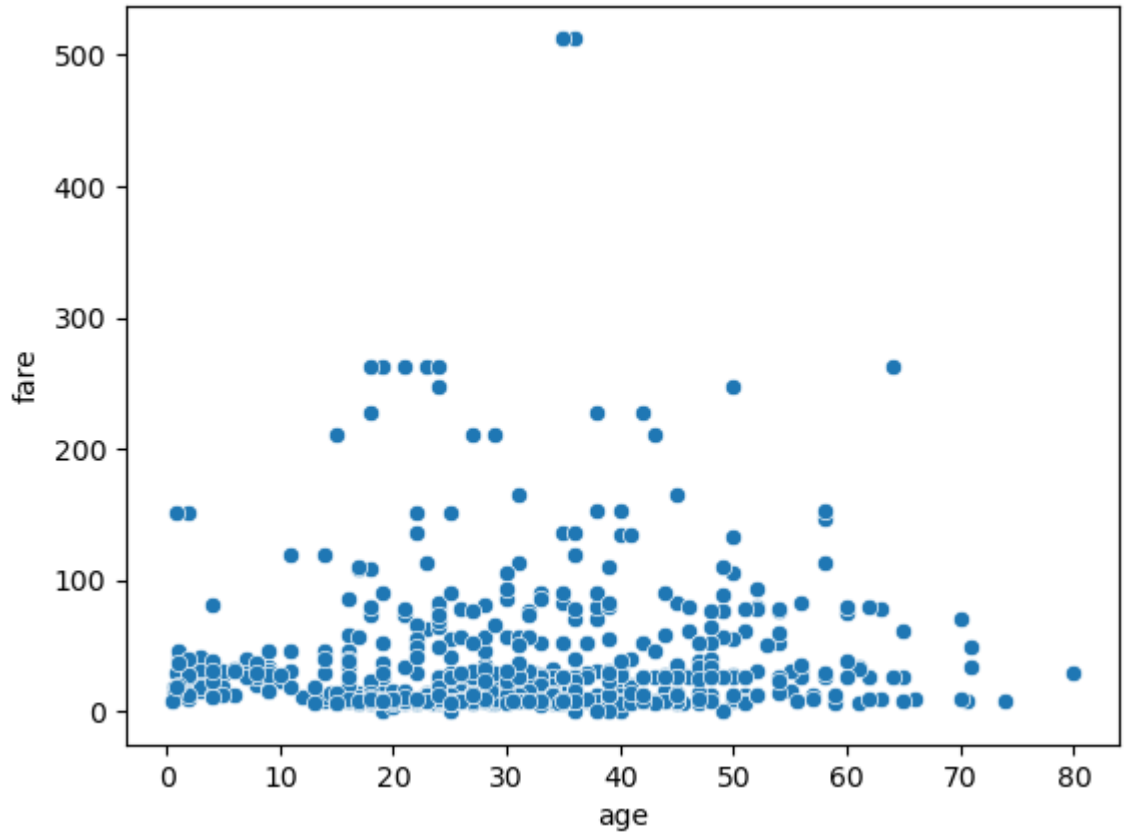
```
# Which variables had the greatest impact on survival rate?
sns.clustermap(cor_df_titanic.corr(), cmap='coolwarm', vmax=1.0, vmin=-1.0, annot=True)
plt.show()
print(cor_df_titanic.corr()['survived'].sort_values(ascending=False))
```

```
survived    1.000000
class       0.338481
fare        0.257307
age        -0.077221
```



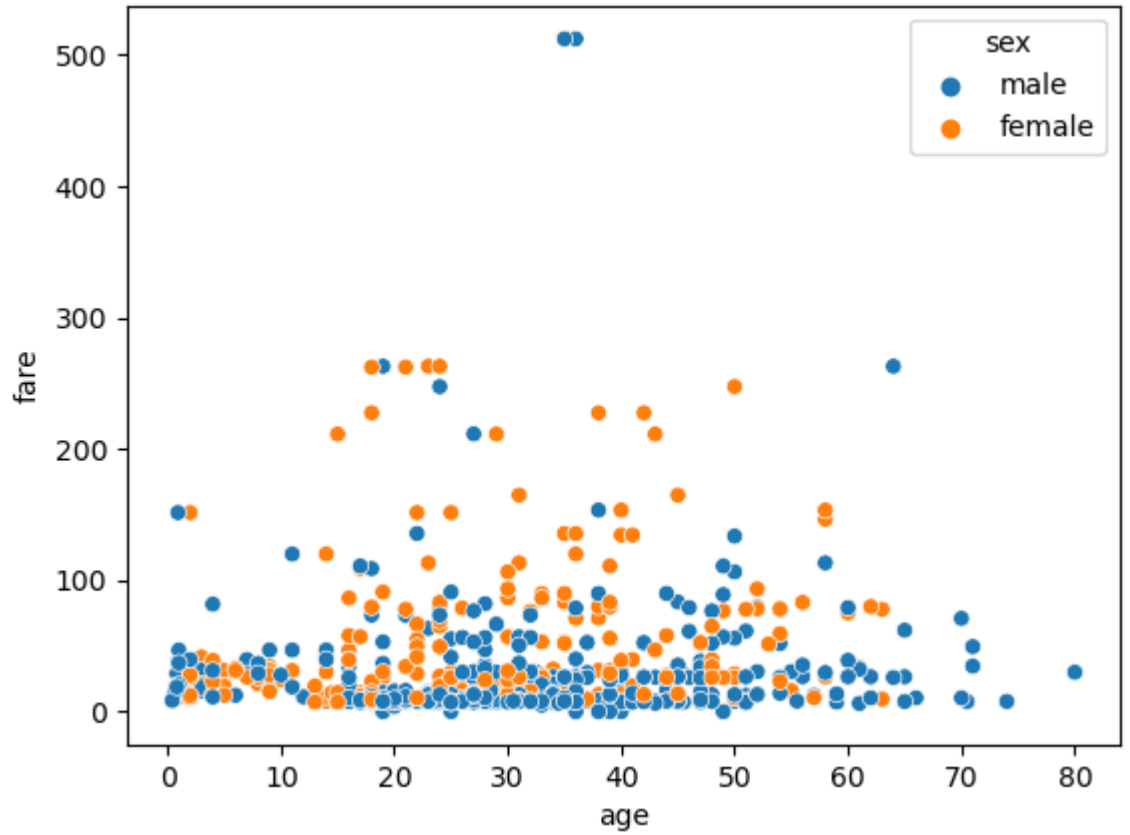
Lab: Titanic Dataset

```
# (8) Fares by age
sns.scatterplot(x='age', y='fare', data=df_titanic)
plt.show()
```



Lab: Titanic Dataset

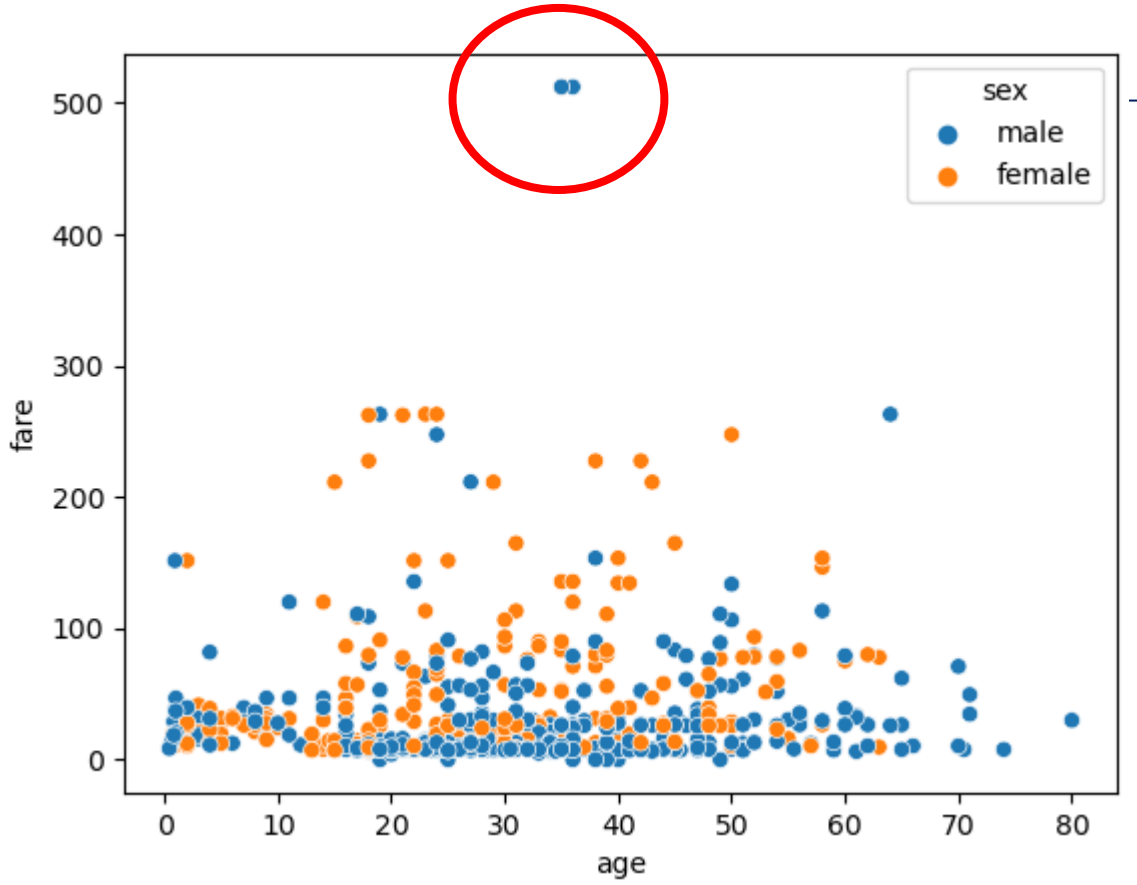
```
# (8) Fares by age (men and women)
sns.scatterplot(x='age', y='fare', hue='sex', data=df_titanic)
plt.show()
```



Lab: Titanic Dataset

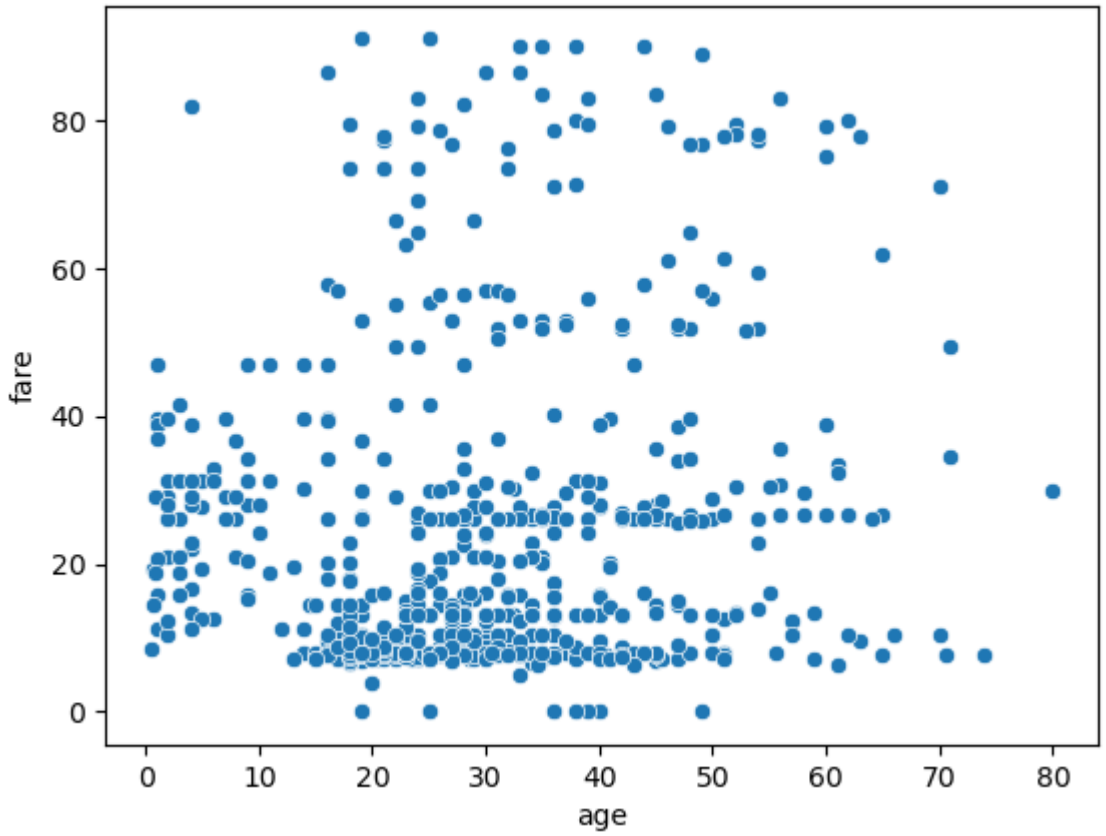
```
# (8) Fares by age (men and women)
sns.scatterplot(x='age', y='fare', hue='sex', data=df_titanic)
plt.show()
```

Remove the outliers



Lab: Titanic Dataset

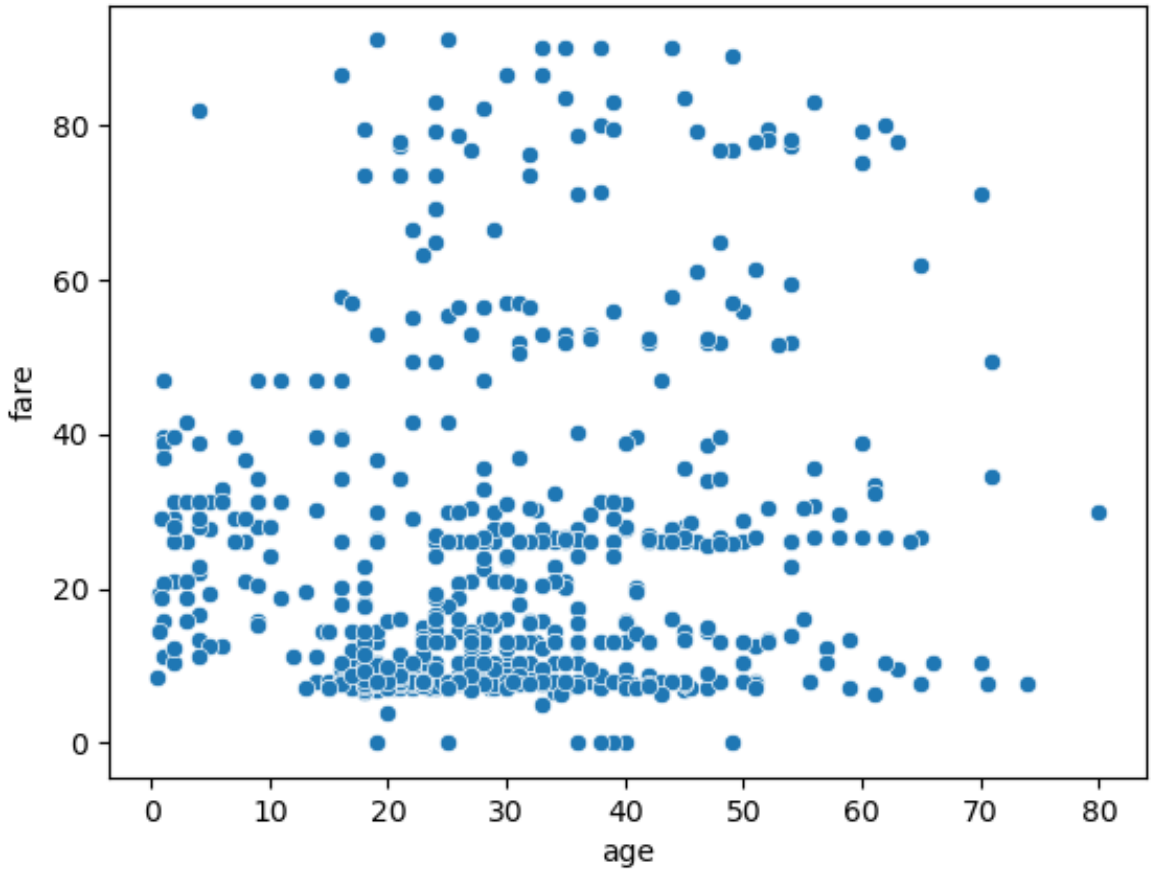
```
# (8) Fares by age
```

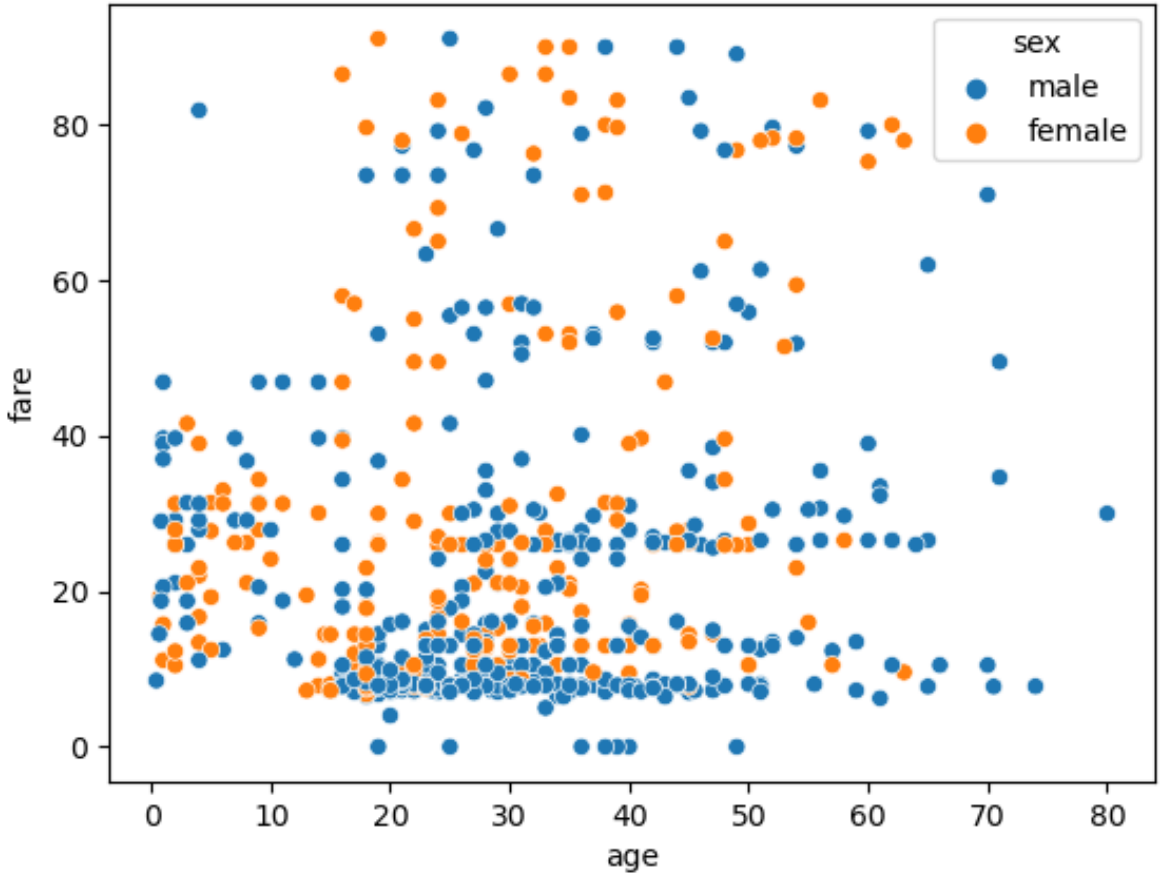


Lab: Titanic Dataset

```
# (8) Fares by age (men and women, -outliers)
fare_q3 = df_titanic['fare'].quantile(q=0.75)
fare_q1 = df_titanic['fare'].quantile(q=0.25)
fare_iqr = fare_q3 - fare_q1
condition = df_titanic['fare'] <= 4*fare_iqr
new_df_titanic = df_titanic[condition]

sns.scatterplot(x='age', y='fare', data=new_df_titanic)
sns.scatterplot(x='age', y='fare', hue='sex', data=df_titanic)

plt.show()
```

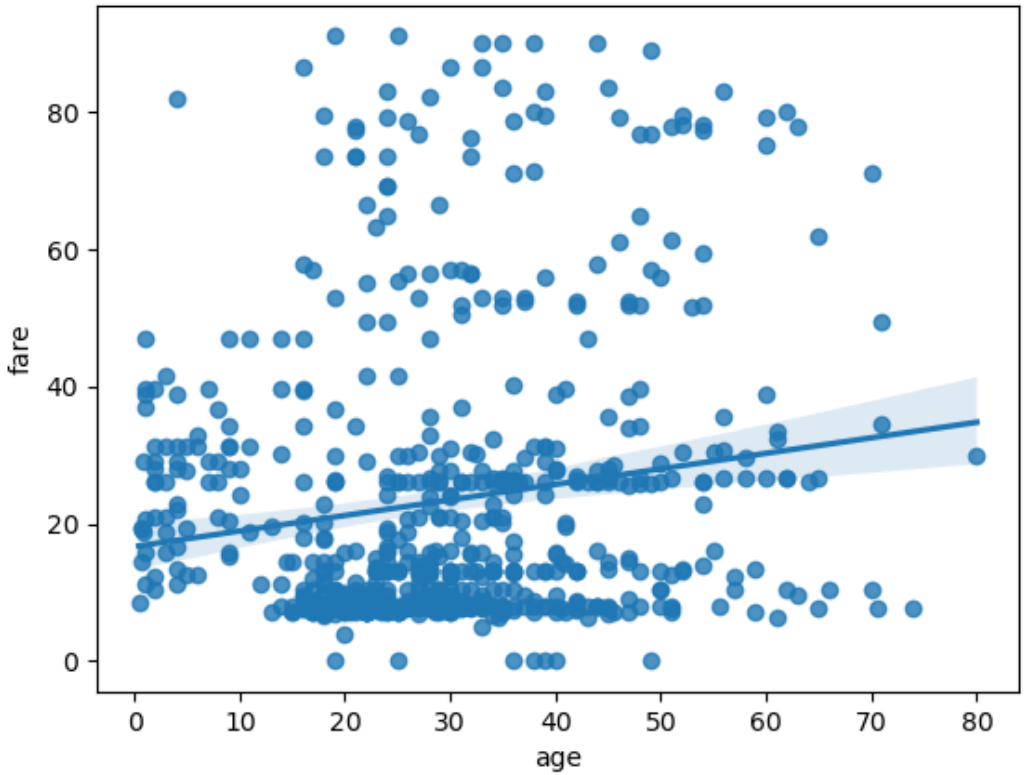
Lab: Titanic Dataset

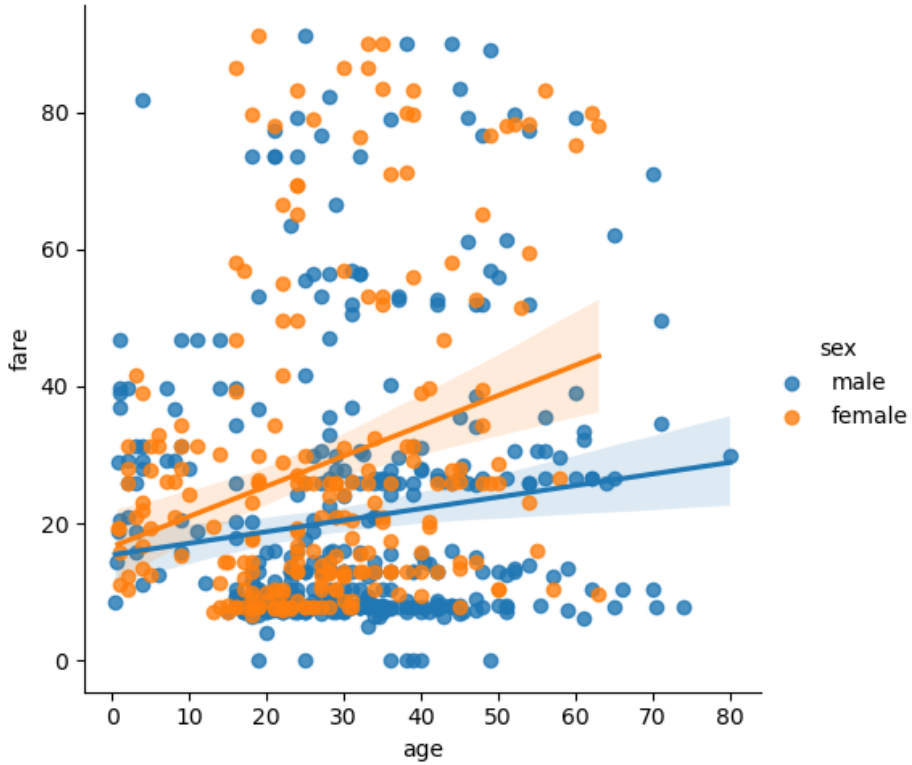




Lab: Titanic Dataset

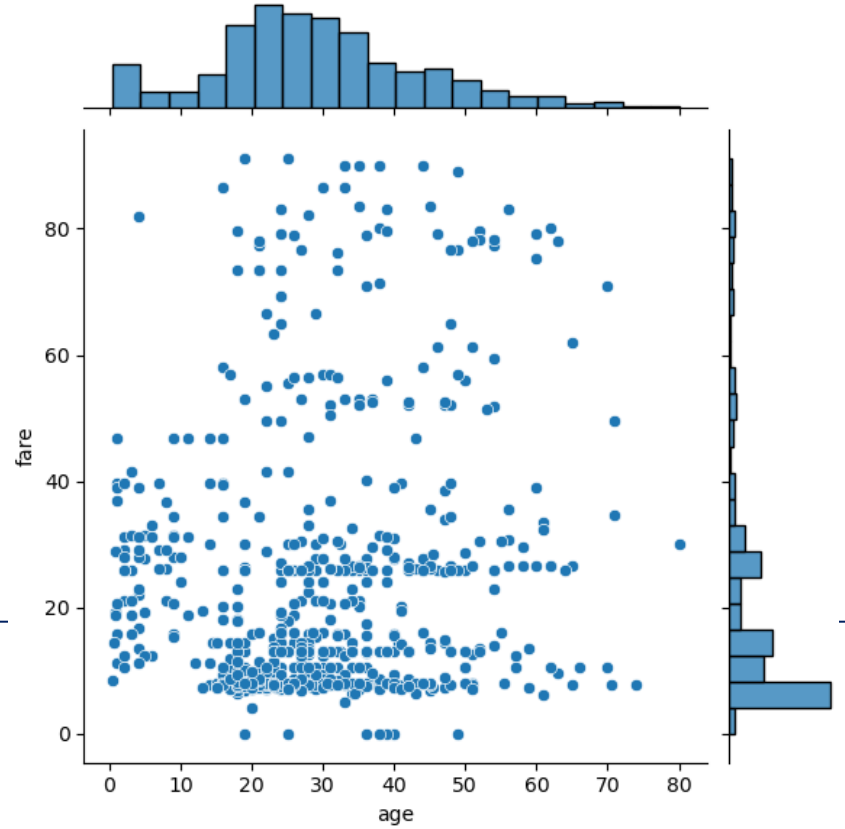
```
# (9) reglot & lmplot
sns.regplot(x='age', y='fare', data=new_df_titanic)
sns.lmplot(x='age', y='fare', hue='sex',data=new_df_titanic)
plt.show()
```

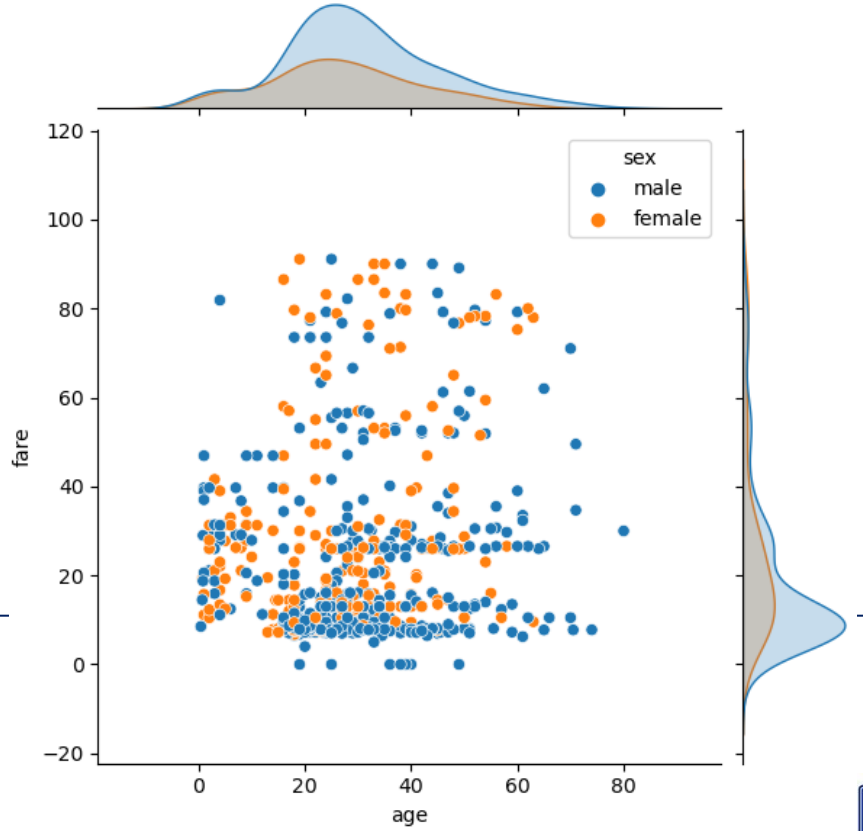




Lab: Titanic Dataset

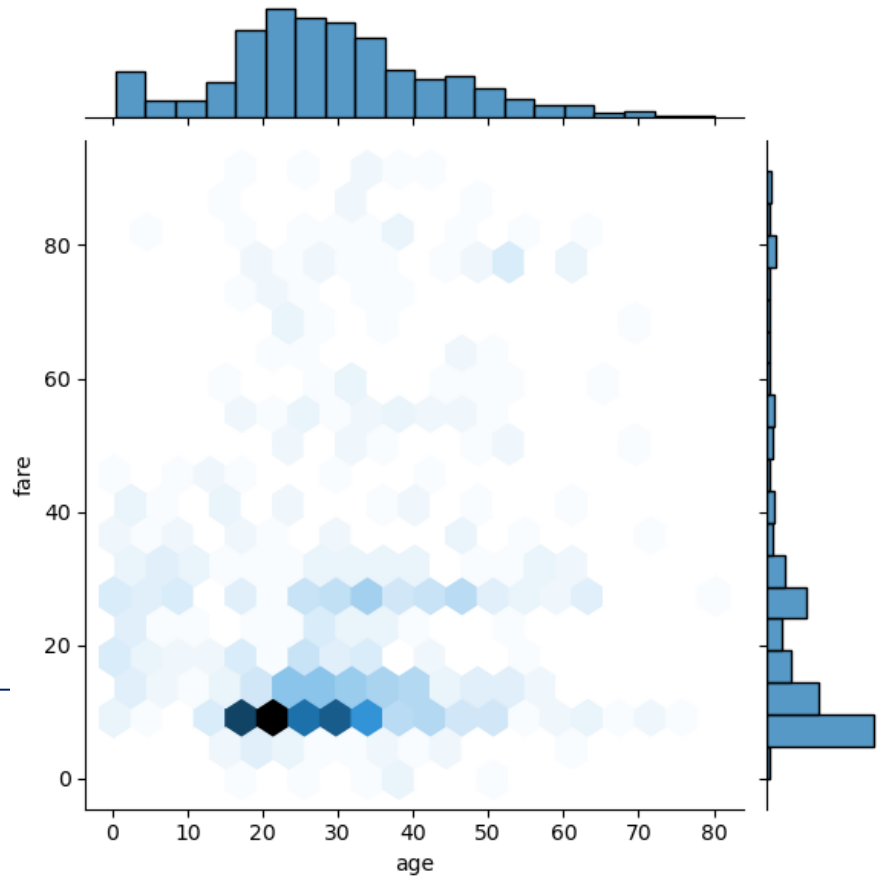
```
# (10) jointplots (fares by ages, -outliers)
sns.jointplot(x='age', y='fare', data=new_df_titanic)
sns.jointplot(x='age', y='fare', hue='sex', data=new_df_titanic)
plt.show()
```

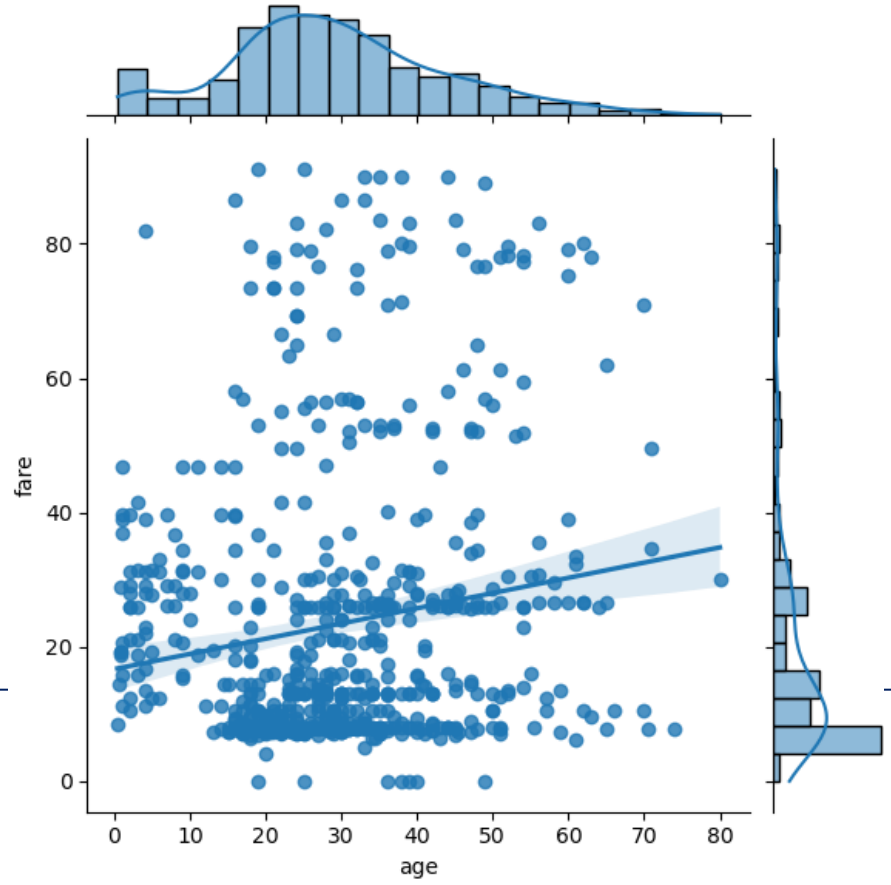




Lab: Titanic Dataset

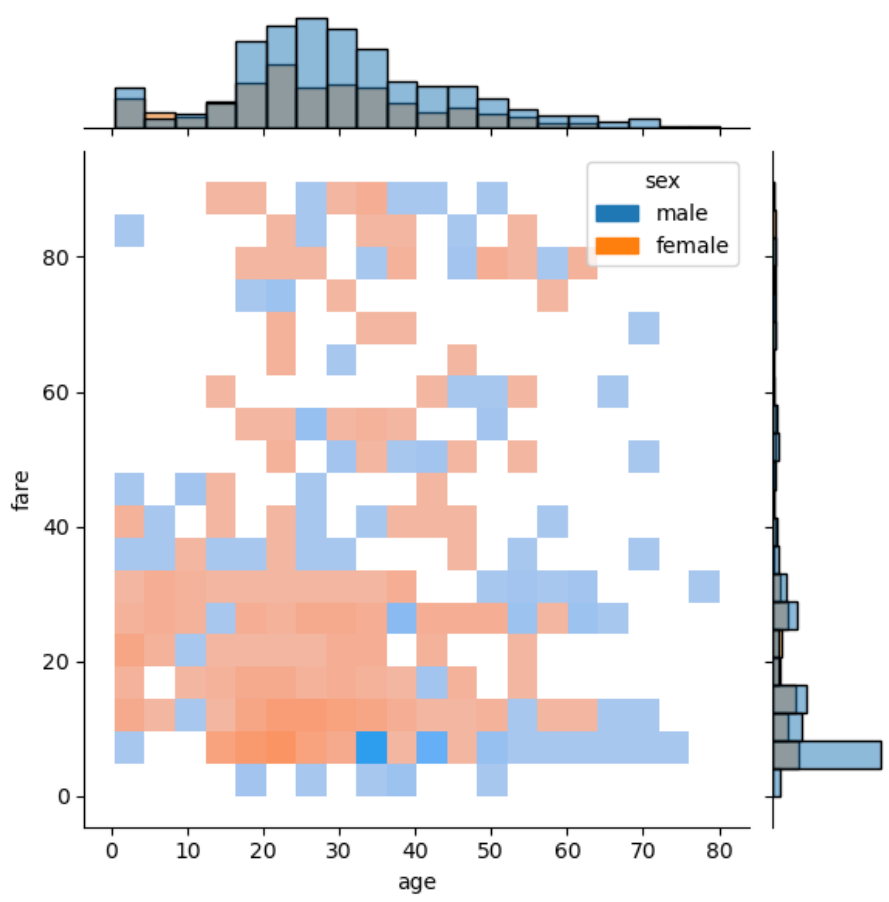
```
# (10) jointplots (fares by ages, -outliers)
sns.jointplot(x='age', y='fare', kind='hex',data=new_df_titanic)
sns.jointplot(x='age', y='fare', kind='reg',data=new_df_titanic)
plt.show()
```

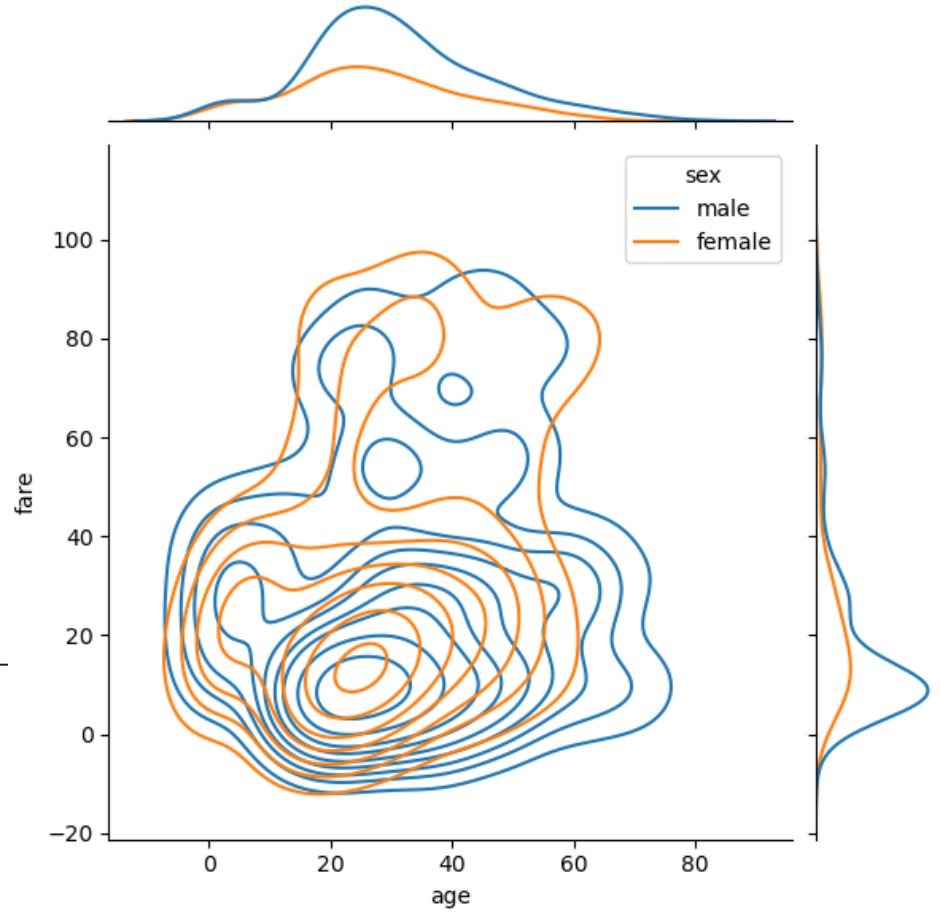




Lab: Titanic Dataset

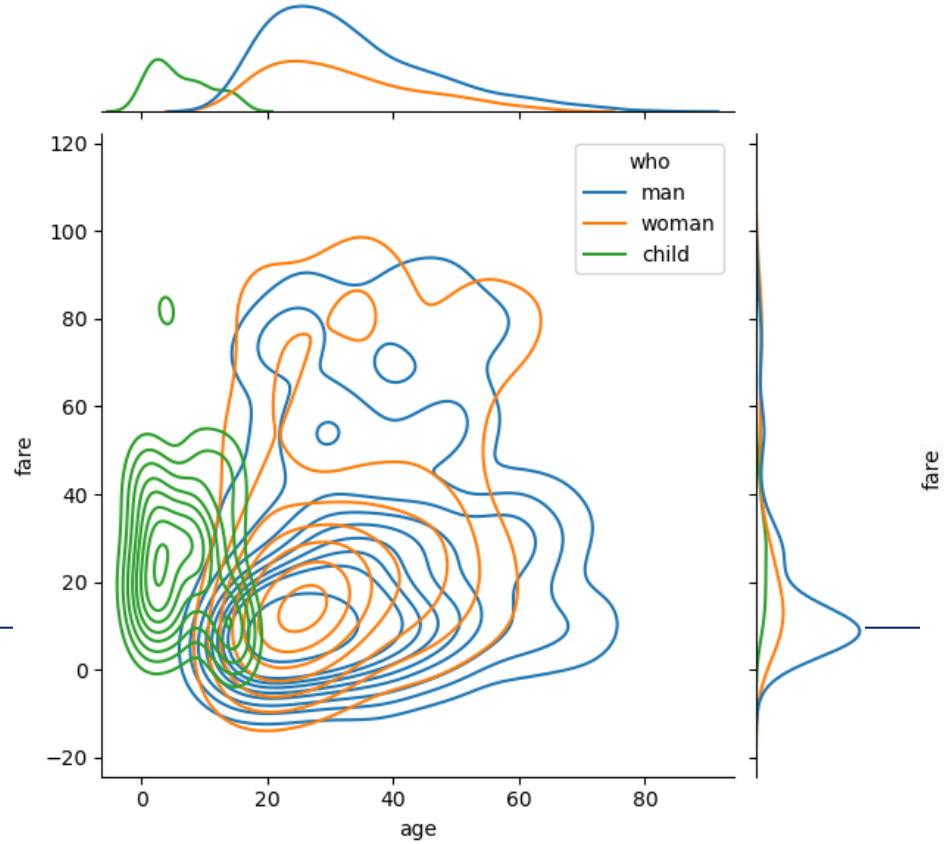
```
# (10) jointplots (fares by ages, -outliers)
sns.jointplot(x='age', y='fare', hue='sex', kind='hist',data=new_df_titanic)
sns.jointplot(x='age', y='fare', hue='sex', kind='kde',data=new_df_titanic)
plt.show()
```

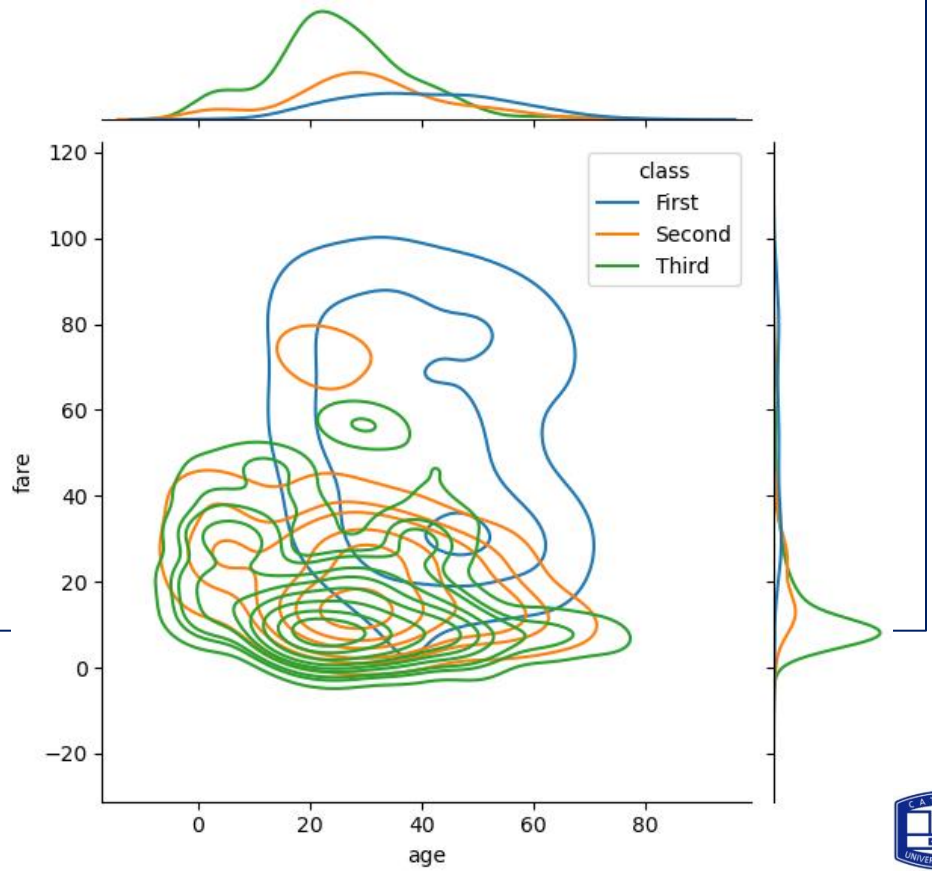




Lab: Titanic Dataset

```
# (10) jointplots (fares by ages, -outliers)
sns.jointplot(x='age', y='fare', hue='who', kind='kde',data=new_df_titanic)
sns.jointplot(x='age', y='fare', hue='class', kind='kde',data=new_df_titanic)
plt.show()
```

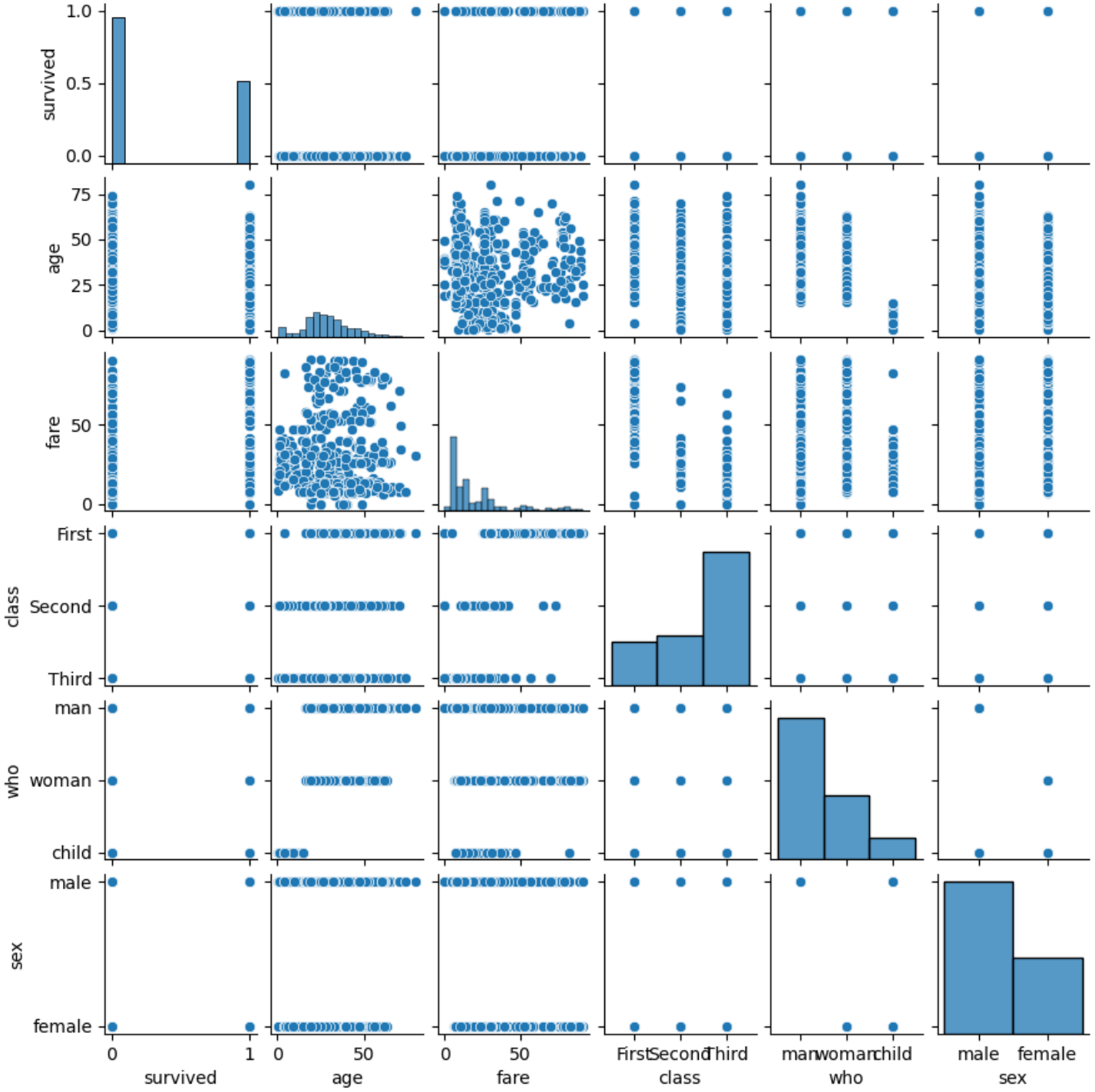




Lab: Titanic Dataset

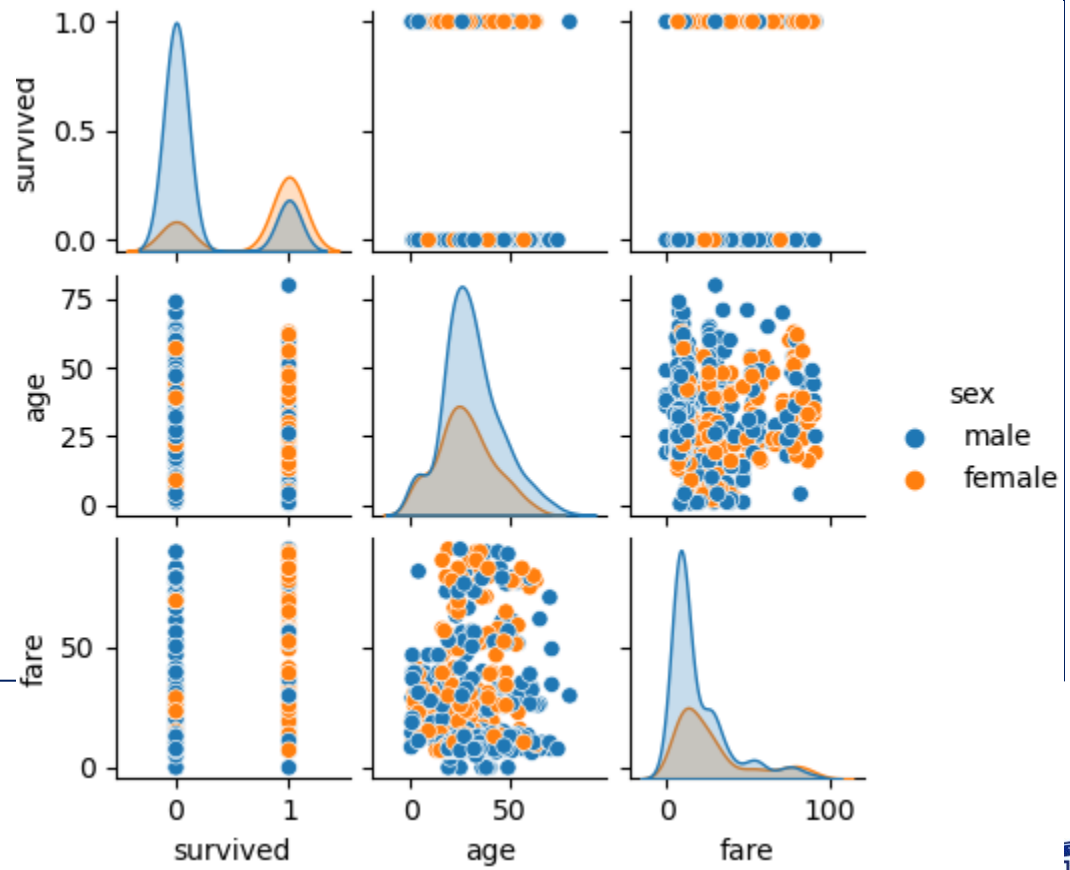
```
# (11) pairplots


sns.pairplot(new_df_titanic,
             x_vars=['survived','age','fare','class','who','sex'],
             y_vars=['survived','age','fare','class','who','sex'],
             kind='scatter',
             height=1.5)
plt.show()
```

Lab: Titanic


survived
age
fare
class
who
sex
1.0
0.5
0.0
75
50
25
0
50
0
First
Second
Third
man
woman
child
male
female
0
1
0
50
0
50
First
Second
Third
man
woman
child
male
female
survived
age
fare
class
who
sex

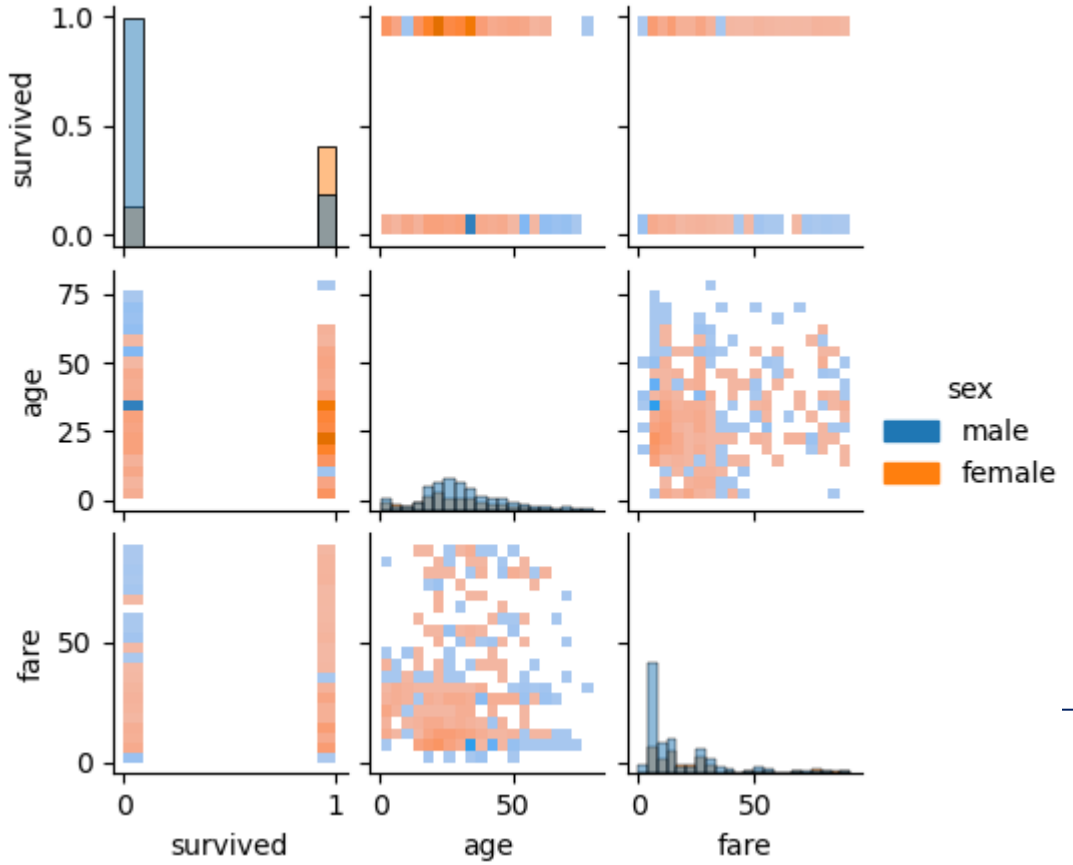
Lab: Titanic Dataset

```
# (11) pairplots


sns.pairplot(new_df_titanic,
             x_vars=['survived','age','fare'],
             y_vars=['survived','age','fare'],
             kind='scatter',
             hue='sex',
             height=1.5)
plt.show()
```



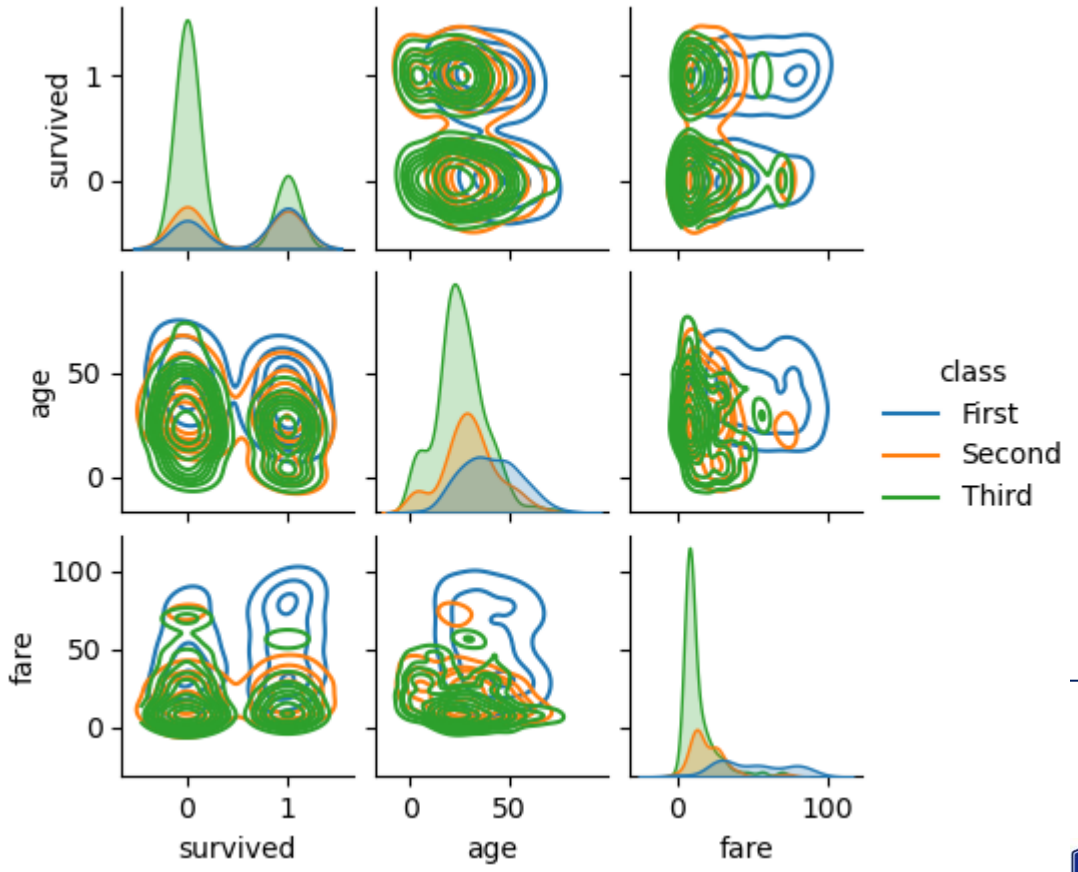
Lab: Titanic Dataset

```
# (11) pairplots


sns.pairplot(new_df_titanic,
             x_vars=['survived','age','fare'],
             y_vars=['survived','age','fare'],
             kind='hist',
             hue='sex',
             height=1.5)
plt.show()
```



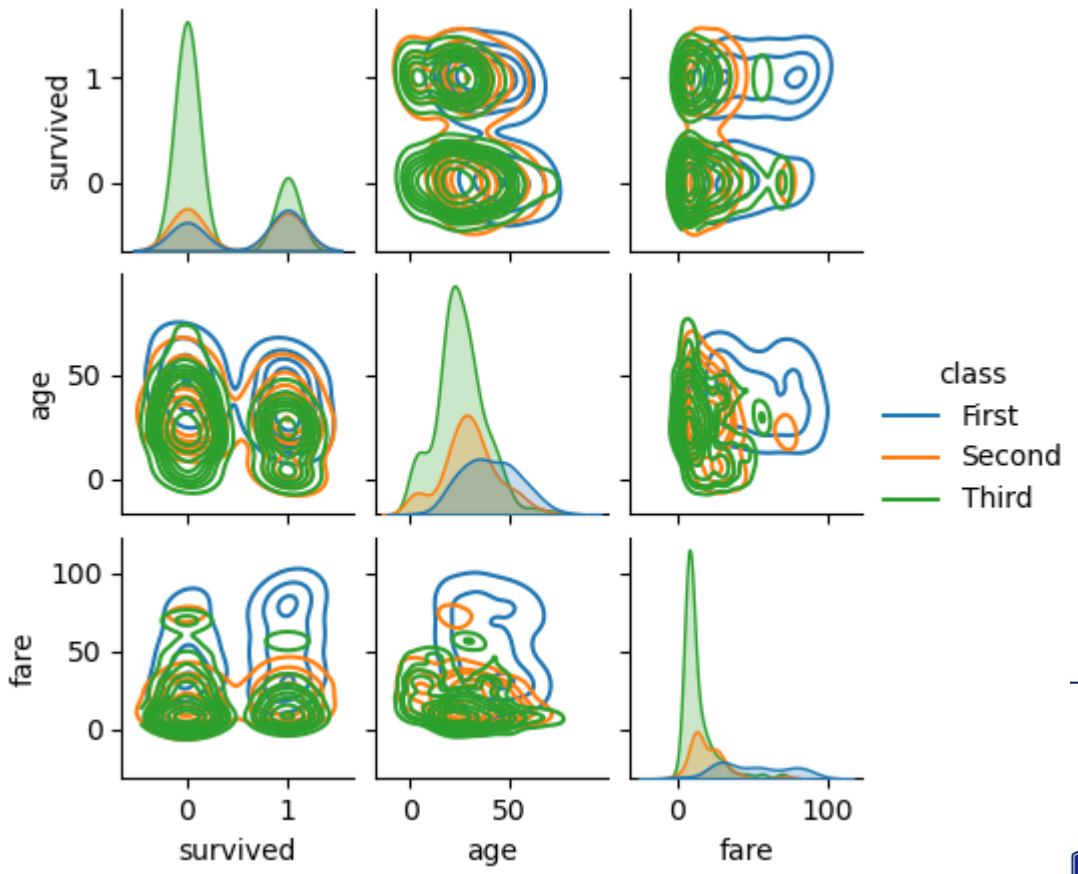
Lab: Titanic Dataset

```
# (11) pairplots
```



Lab: Titanic Dataset

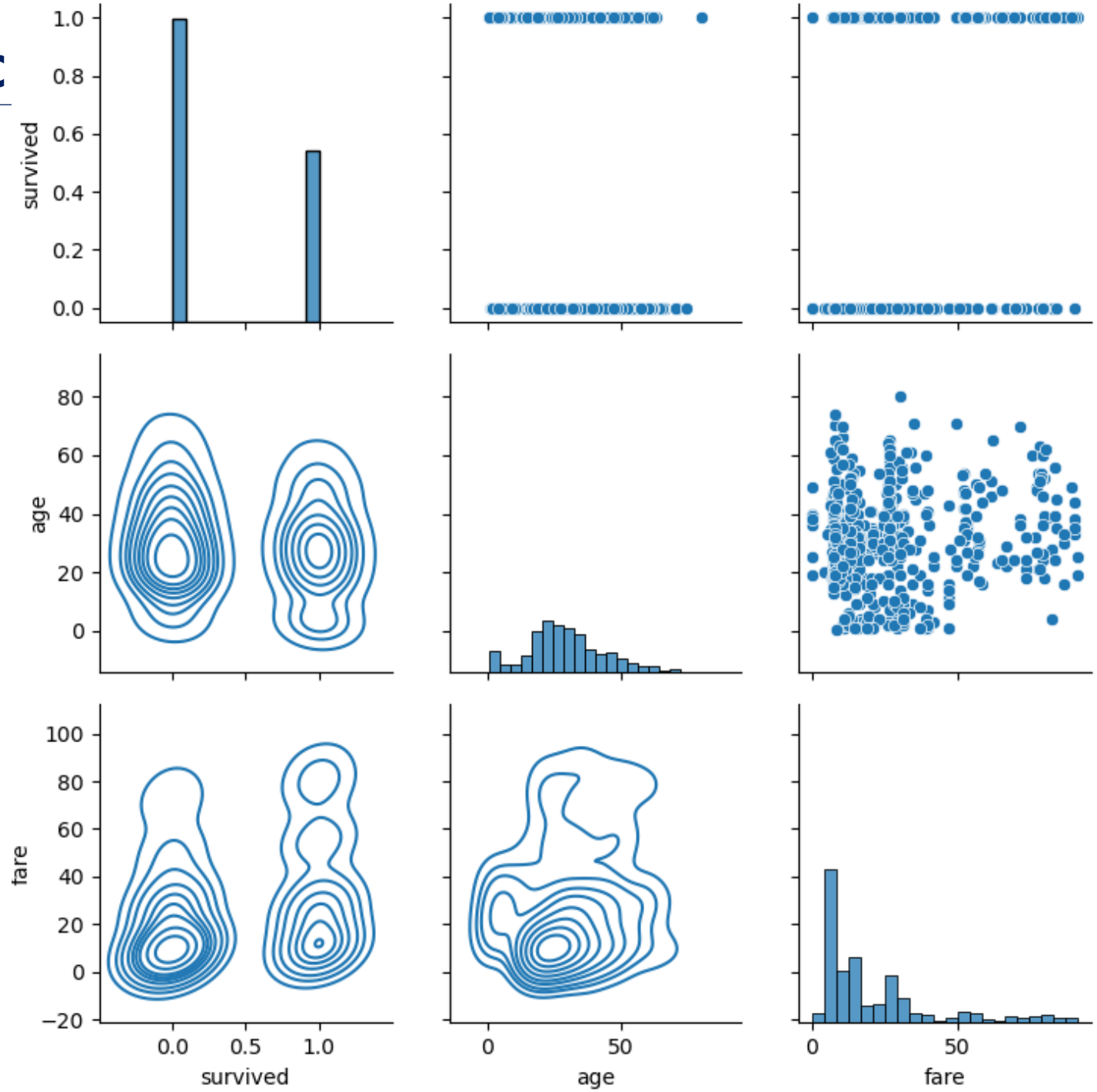
```
# (11) pairplots


sns.pairplot(new_df_titanic,
             x_vars=['survived','age','fare'],
             y_vars=['survived','age','fare'],
             kind='kde',
             hue='class',
             height=1.5)
plt.show()
```



Lab: Titanic Dataset

```
# (12) pairgrid()
grid = sns.PairGrid(new_df_titanic[['survived','age','fare']])
grid.map_diag(sns.histplot)
grid.map_lower(sns.kdeplot)
grid.map_upper(sns.scatterplot)
plt.show()
```

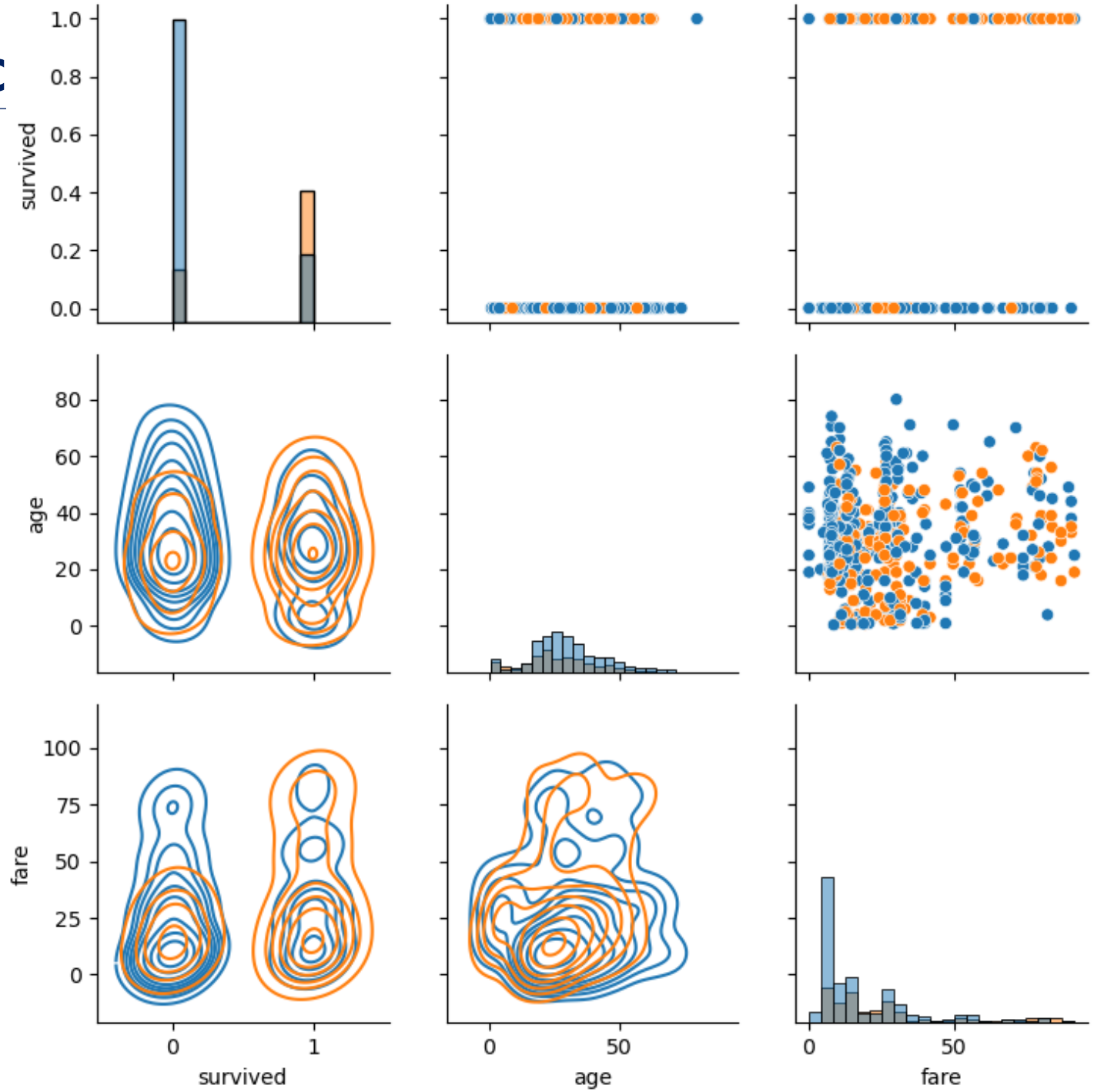
Lab: Titanic


survived
age
fare
1.0
0.8
0.6
0.4
0.2
0.0
80
60
40
20
0
100
80
60
40
20
0
−20
0.0
0.5
1.0
0
50
0
50
survived
age
fare

Lab: Titanic Dataset

```
# (12) pairgrid()
grid = sns.PairGrid(new_df_titanic[['survived','age','fare','sex']],hue='sex')
grid.map_diag(sns.histplot)
grid.map_lower(sns.kdeplot)
grid.map_upper(sns.scatterplot)
plt.show()
```

Lab: Titanic




Bye Bye