End-to-End Machine Learning: Exploring and Plotting Data

Glenn Bruns
CSUMB

Learning outcomes

After this lecture you should be able to:

1. Preprocess, explore and plot data using Pandas, matplotlib, and seaborn
   - detect and process missing data
   - compute basic statistics
   - create plots with matplotlib and seaborn

How many NaN values in the data?

```
In [232]: dat.count()/len(dat)
Out[232]:
PassengerId    1.000000
Survived       1.000000
Pclass         1.000000
Name           1.000000
Sex            1.000000
Age            0.801347
SibSp          1.000000
Parch          1.000000
Ticket         1.000000
Fare           1.000000
Cabin          0.228956
Embarked       0.997755
dtype: float64
```

In Pandas, NaN is used to mean 'missing'.

dat.count() gives number of non-NaN (and non-None)values by column

len(dat) gives number of rows in data frame

dat.count(axis=1) gives number of non-NaN values by row

Only 23% of Cabin variable is non-NaN

What fraction of passengers survived?

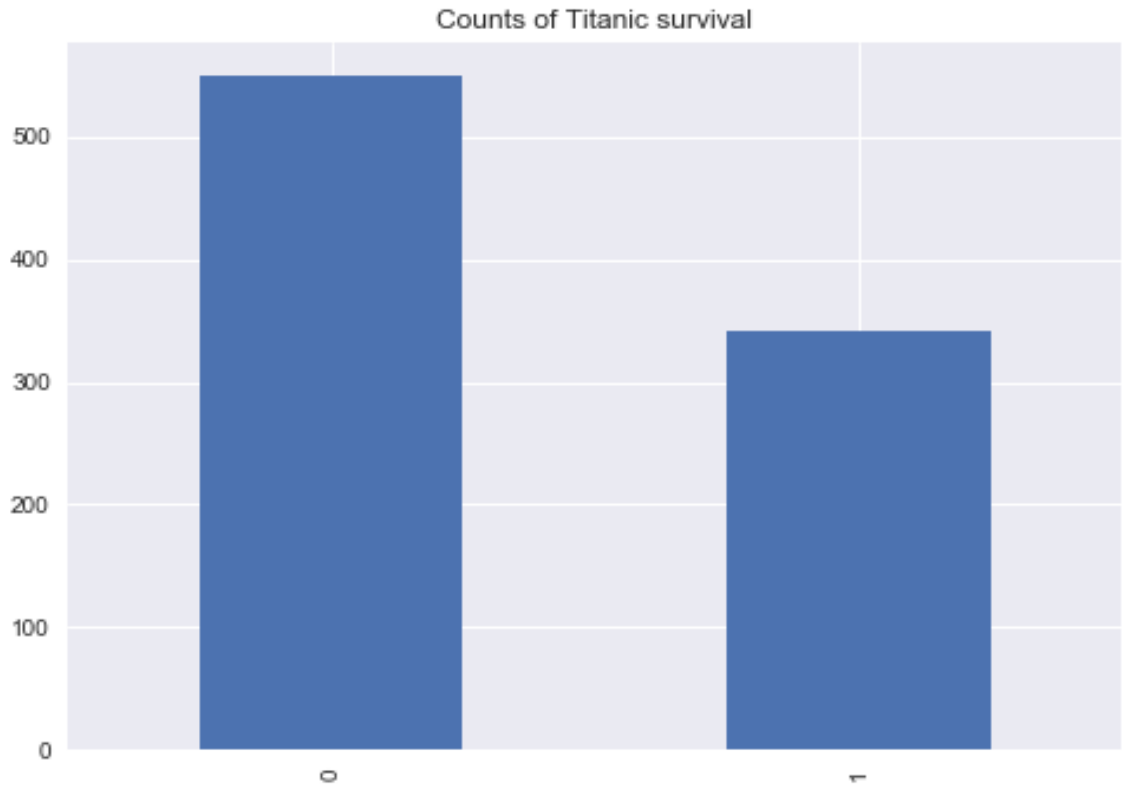
```
In [235]: dat["Survived"].value_counts()
Out[235]:
0    549
1    342
Name: Survived, dtype: int64

In [236]: dat["Survived"].value_counts()/len(dat)
Out[236]:
0    0.616162
1    0.383838
Name: Survived, dtype: float64
```

value_counts() is similar to R's 'table' function

We see that about 38% of Titanic passenger's survived.

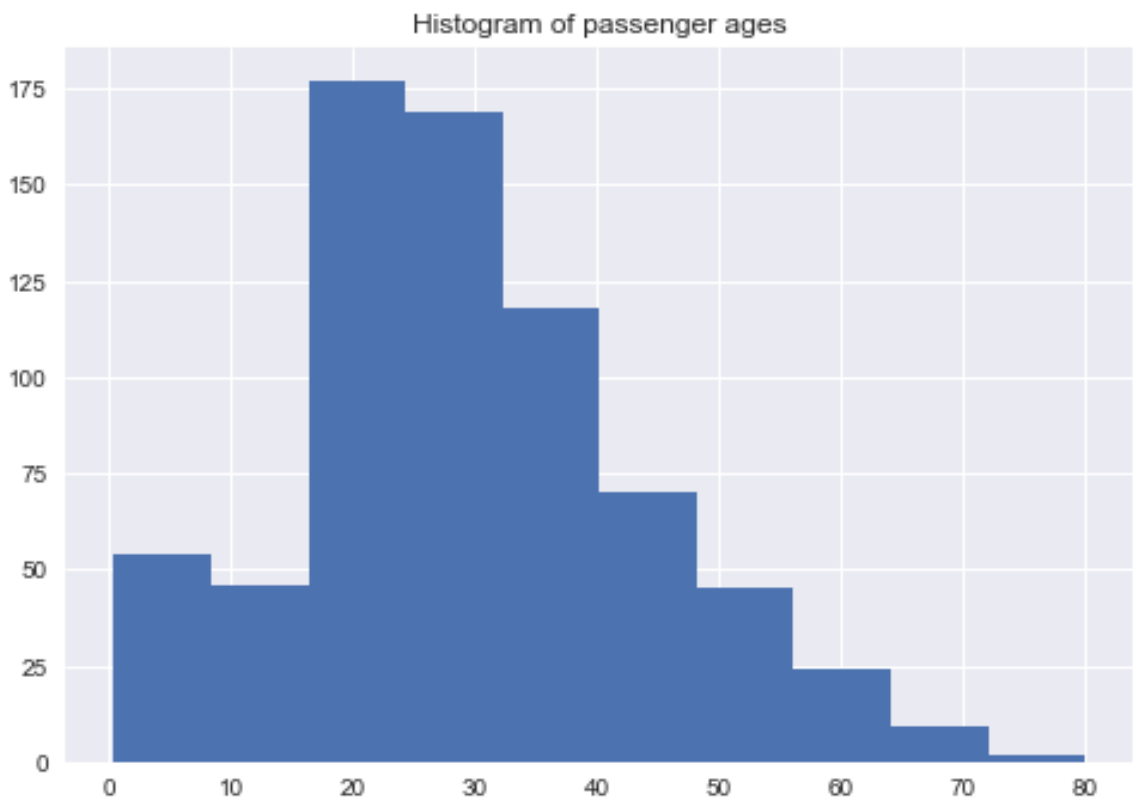
Plotting survival counts

```
import matplotlib.pyplot as plt
dat["Survived"].value_counts().plot(kind="bar")
plt.title("Counts of Titanic survival")
```



Histogram of passenger ages

```
plt.hist(dat['Age'].dropna())
plt.title("Histogram of passenger ages")
```



Seaborn

matplotlib is a low-level and can be painful, even compared to R's base graphics

Seaborn is a package for "statistical data visualization"

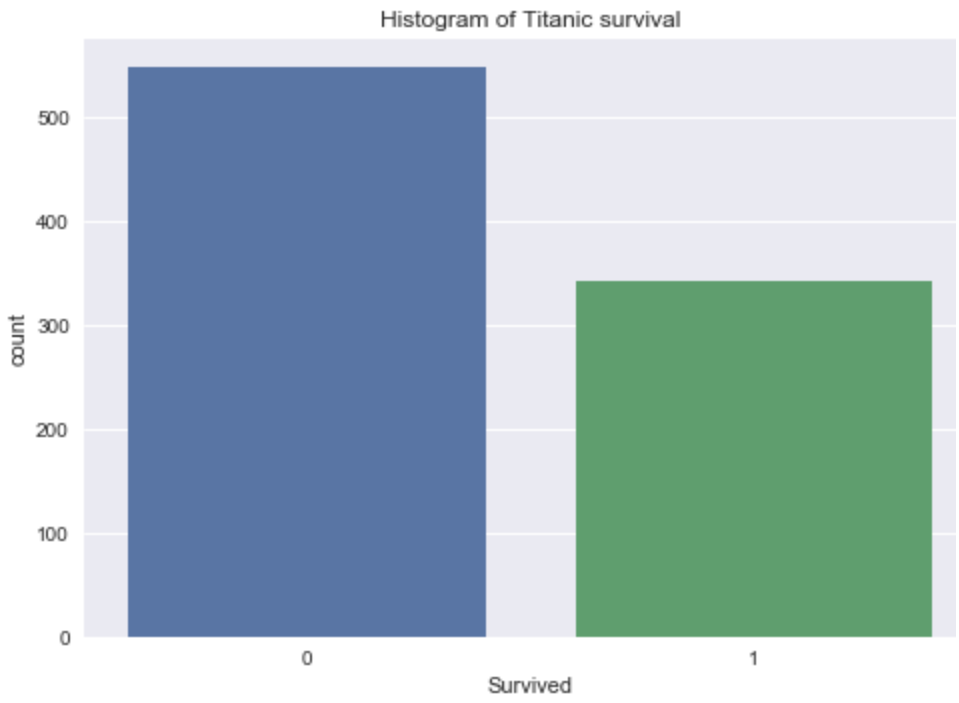
It "understands" statistics, so is much easier to use for common plots

Built on top of matplotlib; understands Pandas

seaborn.pydata.org/tutorial.html

Barplot with Seaborn

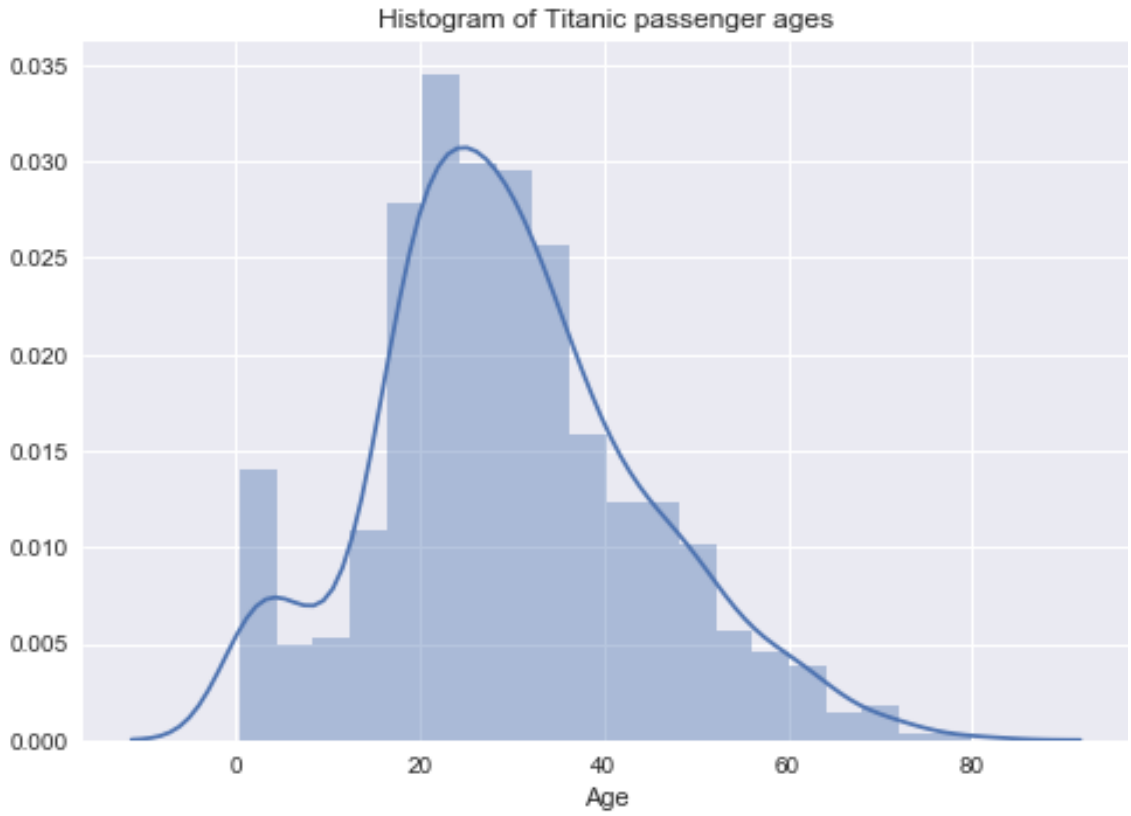
```
import seaborn as sns
sns.countplot(x="Survived", data=dat)
plt.title("Histogram of Titanic survival")
```



Sadly, the two bars get different colors by default.

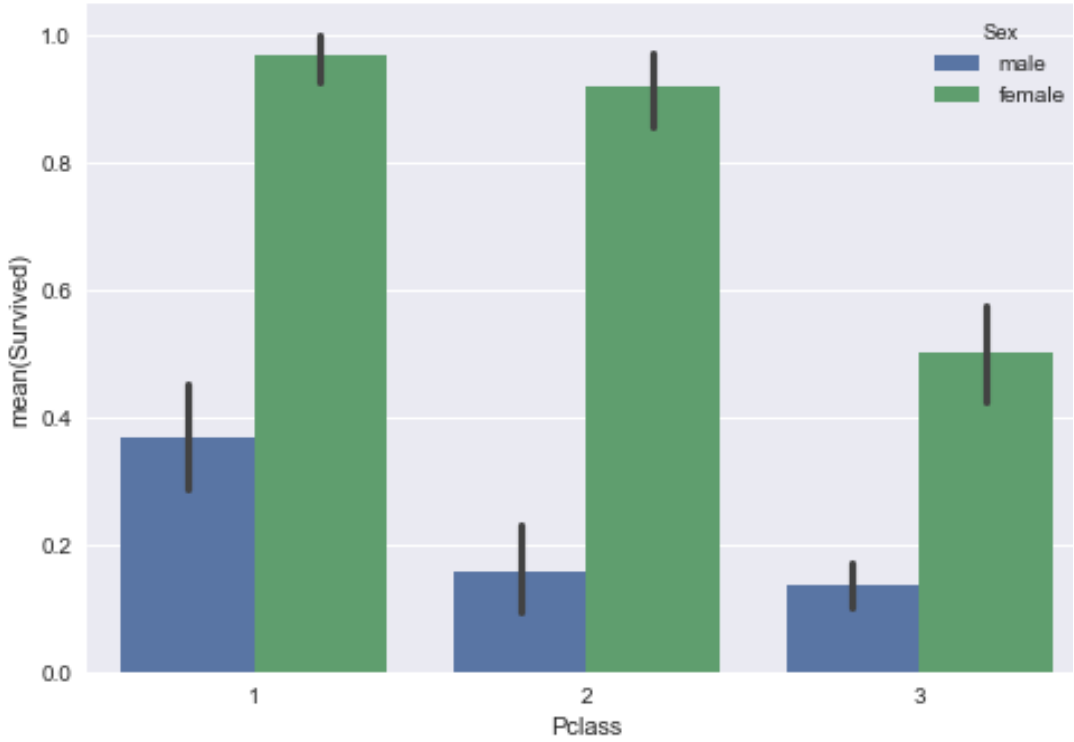
Histogram with Seaborn

```
sns.distplot(dat['Age'].dropna())
plt.title("Histogram of Titanic passenger ages")
```



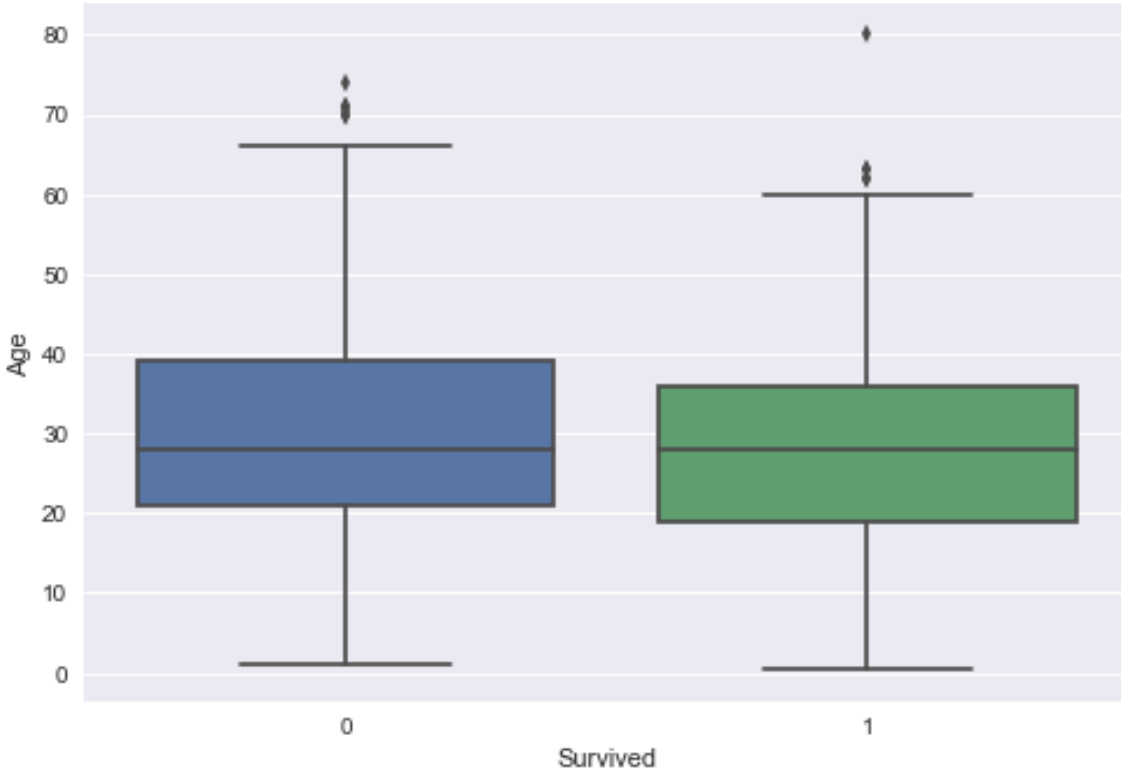
Grouped barplot with Seaborn

```
sns.barplot(x="Pclass", y="Survived", hue="Sex", data=dat)
```



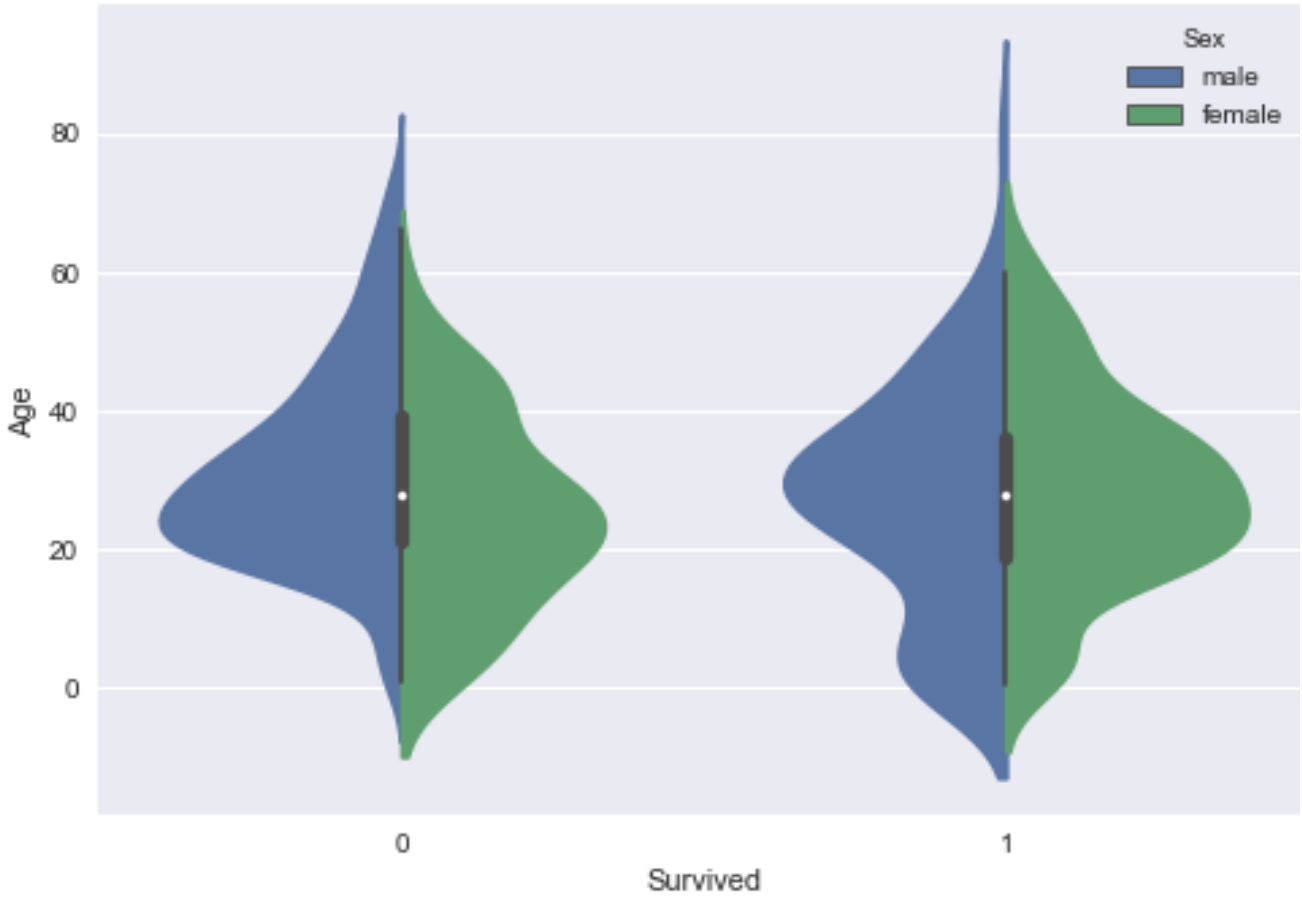
Is Age a good predictor of survival?

```
sns.boxplot(x="Survived", y="Age", data=dat)
```



Showing distribution with violin plot

```
sns.violinplot(x="Survived", y="Age", hue="Sex", data=dat, split=True)
```



Summary

- Count of NaN values
- Tabulation, like R's 'table' function, using value_counts()
- Plotting with matplotlib
- Higher-level plotting with Seaborn