ANOVA

Analysis of variance

In [ ]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
# import dataset
kashti = sns.load_dataset('titanic')
kashti.head()
```

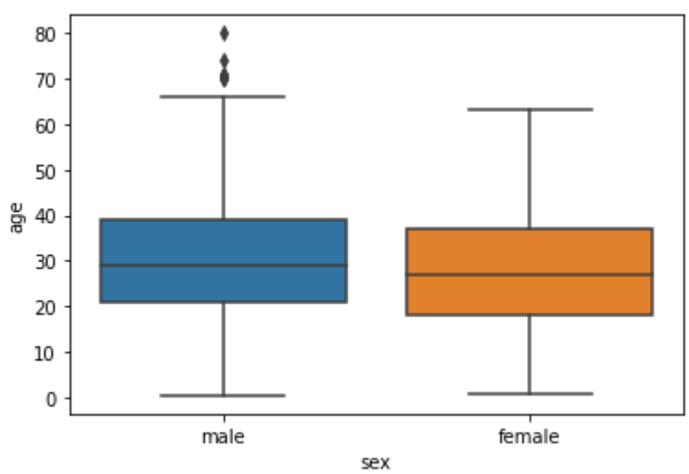
Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN

Comparison between 2 categorical and 1 continuous variable (will apply t-test)

In [ ]:
```
# make box plot
sns.boxplot(x='sex', y='age', data=kashti)
```

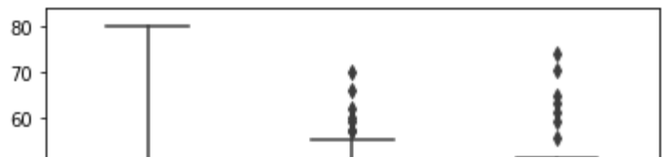
Out[ ]:
```
<AxesSubplot:xlabel='sex', ylabel='age'>
```



Comparison between 3 categorical and 1 continuous variable (will apply ANNOVA-test)

In [ ]:
```
sns.boxplot(x='class', y='age', data=kashti)
```

Out[ ]:
```
<AxesSubplot:xlabel='class', ylabel='age'>
```



example of ANNOVA with iris dataset

In [ ]:
```
phool = sns.load_dataset('iris')
phool.head()
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [ ]:
```
# to check random sample
phool.sample(10)
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width	species
100	6.3	3.3	6.0	2.5	virginica
15	5.7	4.4	1.5	0.4	setosa
38	4.4	3.0	1.3	0.2	setosa
149	5.9	3.0	5.1	1.8	virginica
102	7.1	3.0	5.9	2.1	virginica
1	4.9	3.0	1.4	0.2	setosa
49	5.0	3.3	1.4	0.2	setosa
143	6.8	3.2	5.9	2.3	virginica
67	5.8	2.7	4.1	1.0	versicolor
45	4.8	3.0	1.4	0.3	setosa

In [ ]:
```
# to check columns name
phool.columns
```

Out[ ]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [ ]:
```
# to check mean, IQR etc.
phool.describe()
```

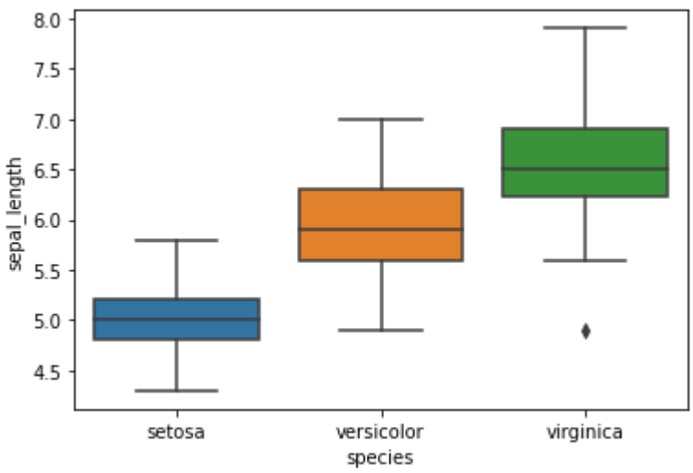
Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [ ]:
```
# draw the boxplot
sns.boxplot('species','sepal_length', data=phool)
```

```
c:\Users\kalee\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[ ]: <AxesSubplot:xlabel='species', ylabel='sepal_length'>



In [ ]:
```
# import models of stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [ ]:
```
# One Way ANNOVA
# first give numerical value and then give categorical(~ is called tilda sign)
mod = ols('sepal_length ~ species', data=phool).fit()
aov_table = sm.stats.anova_lm(mod, type=2) # assignment (why the type is 2)
print(aov_table)
```

```
              df     sum_sq    mean_sq           F        PR(>F)
species      2.0  63.212133  31.606067  119.264502  1.669669e-31
Residual   147.0  38.956200   0.265008         NaN           NaN
```

Solution to assignment

- as we get 2 categorial variable so we set type 2 here

In [ ]:
```
# Pairwise Comparison
pair_t = mod.t_test_pairwise('species', method='bonferroni') # method use for pairwis
# also (sidak) can be used in place of boneferroni
pair_t.result_frame
# True result in reject_boneferroni tells that HO
# hypothesis(Compared classes are Same ) rejects.
# So we will accept H1 which tells us that
# Compared classes are Significantly diff )
```

Out[ ]:

	coef	std err	t	P>|t|	Conf. Int. Low	Conf. Int. Upp.	pvalue-bonferroni	reject-bonferroni
versicolor-setosa	0.930	0.102958	9.032819	8.770194e-16	0.726531	1.133469	2.631058e-15	True
virginica-setosa	1.582	0.102958	15.365506	2.214821e-32	1.378531	1.785469	6.644464e-32	True
virginica-versicolor	0.652	0.102958	6.332686	2.765638e-09	0.448531	0.855469	8.296915e-09	True

In [ ]:
```
# tukey hsd test
import pingouin as pg

# first calculate ANOVA Table
aov = pg.anova(data=phool, dv = 'sepal_length', between = 'species', detailed = True)
# in dv should give continuous and in between categorical variable
print(aov)
```

```
    Source         SS   DF         MS           F         p-unc       np2
0  species  63.212133    2  31.606067  119.264502  1.669669e-31  0.618706
1   Within  38.956200  147   0.265008         NaN           NaN       NaN
c:\Users\kalee\anaconda3\lib\site-packages\pingouin\parametric.py:992: FutureWarning:
Not prepending group keys to the result index of transform-like apply. In the future,
the group keys will be included in the index, regardless of whether the applied funct
ion returns a like-indexed object.
To preserve the previous behavior, use

        >>> .groupby(..., group_keys=False)

To adopt the future behavior and silence this warning, use

        >>> .groupby(..., group_keys=True)
  sserror = grp.apply(lambda x: (x - x.mean()) ** 2).sum()
```

In [ ]:
```
# Apply tukey hsd

pt = pg.pairwise_tukey(data=phool, dv = 'sepal_length', between = 'species')
print(pt)
```

```
c:\Users\kalee\anaconda3\lib\site-packages\pingouin\parametric.py:992: FutureWarning:
Not prepending group keys to the result index of transform-like apply. In the future,
the group keys will be included in the index, regardless of whether the applied funct
ion returns a like-indexed object.
To preserve the previous behavior, use

        >>> .groupby(..., group_keys=False)

To adopt the future behavior and silence this warning, use

        >>> .groupby(..., group_keys=True)
  sserror = grp.apply(lambda x: (x - x.mean()) ** 2).sum()
           A           B  mean(A)  mean(B)   diff        se          T  \
0      setosa  versicolor    5.006    5.936 -0.930  0.102958  -9.032819
1      setosa   virginica    5.006    6.588 -1.582  0.102958 -15.365506
2  versicolor   virginica    5.936    6.588 -0.652  0.102958  -6.332686

        p-tukey    hedges
0  2.420286e-14 -1.792703
1  2.153833e-14 -3.049522
2  8.287554e-09 -1.256820
```

Assignments

1. How to read the ANOVA table
2. Check to apply sidak in place of bonferroni
3. How to see significant of p-tukey in tukey test
4. How to show significant diff on boxplot
5. Also see hedges in tukey test

1. How to read the ANOVA table

1. In this table SS stands for sum square for treatment/between.
2. DF represent the degree of freedom
3. MS stands for Mean square.
   - MS is calculated by dividing of SS values of treatment(between) and of error(within) by DF to get mean square for treatment called **MST** and mean square for error called **MSE**.
   - When the null hypothesis of equal means is true, the two mean squares estimate the same quantity (error variance), and should be of approximately equal magnitude. In other words, their ratio should be close to 1. If the null hypothesis is false, MST should be larger than MSE. SO in above case rejects the null hypothesis (H0) as they are

significantly different(H1) and variance exists.

4. F is the test statistic, used in testing the equality of treatment means is: F=MST/MSE.
5. p-unc means uncorrected p-values
6. np^2 means Partial eta-square effect sizes. np^2=SS(treatment)/SS(treatment)+SS(error)

2. Check to apply sidak in place of bonferroni

•The Bonferroni and Šídák methods can determine statistical significance, compute adjusted P value. •The Šídák method has a bit more power than the Bonferroni method.\ **•The Šídák method assumes that each comparison is independent of the others. If this assumption is independence cannot be supported, choose the Bonferroni method, which does not assume independence**.

•The Bonferroni method is used more frequently, because it is easier to calculate (which doesn't matter when a computer does the work), easier to understand, and much easier to remember.\ But the main difference in two test is that:\ **The Bonferroni test is offered because it is easy to understand. If we enter data into two columns, and wish to compare the two values at each row, then we recommend the Bonferroni method, because it can compute confidence intervals for each comparison. The alternative is the Holm-Šídák method, which has more power, but doesn't compute confidence intervals**.

In [ ]:
```
pair_t = mod.t_test_pairwise('species', method='sidak') # method use for pairwise t_t
# also (sidak) can be used in place of boneferroni
pair_t.result_frame
```

Out[ ]:

	coef	std err	t	P>|t|	Conf. Int. Low	Conf. Int. Upp.	pvalue-sidak	reject-sidak
versicolor-setosa	0.930	0.102958	9.032819	8.770194e-16	0.726531	1.133469	2.631058e-15	True
virginica-setosa	1.582	0.102958	15.365506	2.214821e-32	1.378531	1.785469	6.644464e-32	True
virginica-versicolor	0.652	0.102958	6.332686	2.765638e-09	0.448531	0.855469	8.296915e-09	True

3. How to see significant of p-tukey in tukey test

- p-tukey is the **Tukey-HSD** corrected p-values.

4. How to show significant diff on boxplot

In [ ]:
```

```

5. Also see hedges in tukey test

- Hedges are the effect size in tukey test