```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns
import statsmodels.api as smf
```

```
df = pd.read_excel("Airlines+Data.xlsx")
df
```

1 to 25 of 96 entries Filter

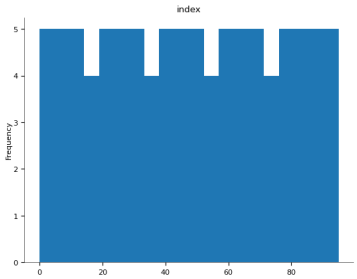
index	Month	Passengers
0	1995-01-01 00:00:00	112
1	1995-02-01 00:00:00	118
2	1995-03-01 00:00:00	132
3	1995-04-01 00:00:00	129
4	1995-05-01 00:00:00	121
5	1995-06-01 00:00:00	135
6	1995-07-01 00:00:00	148
7	1995-08-01 00:00:00	148
8	1995-09-01 00:00:00	136
9	1995-10-01 00:00:00	119
10	1995-11-01 00:00:00	104
11	1995-12-01 00:00:00	118
12	1996-01-01 00:00:00	115
13	1996-02-01 00:00:00	126
14	1996-03-01 00:00:00	141
15	1996-04-01 00:00:00	135
16	1996-05-01 00:00:00	125
17	1996-06-01 00:00:00	149
18	1996-07-01 00:00:00	170
19	1996-08-01 00:00:00	170
20	1996-09-01 00:00:00	158
21	1996-10-01 00:00:00	133
22	1996-11-01 00:00:00	114
23	1996-12-01 00:00:00	140
24	1997-01-01 00:00:00	145

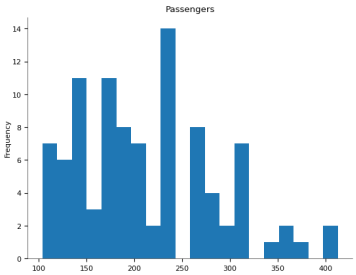
Show 25 per page

1 2 3 4

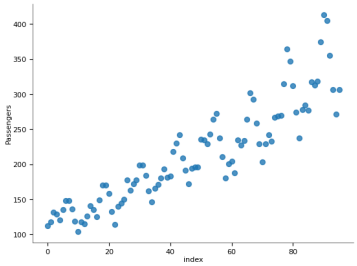
Like what you see? Visit the data table notebook to learn more about interactive tables.

Distributions

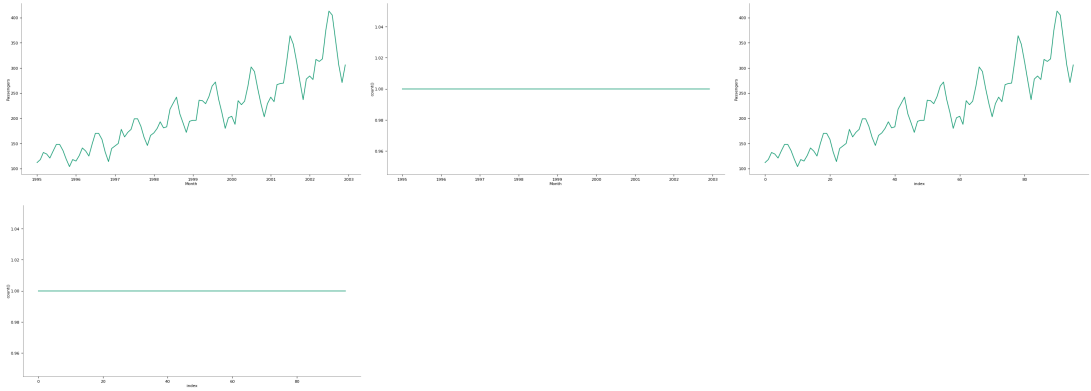




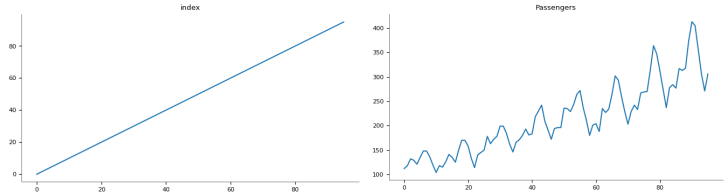
2-d distributions



Time series



Values



Next steps: View recommended plots

Passengers

Month vs Passengers

Month vs count()

Passengers

```
df = pd.read_excel("Airlines+Data.xlsx")
df
```

1 to 25 of 96 entries Filter

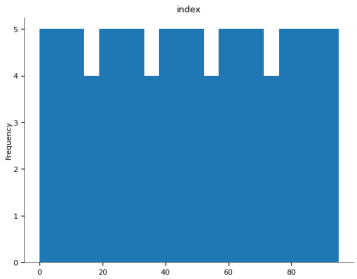
index	Month	Passengers	Months
0	1995-01-01 00:00:00	112	Jan
1	1995-02-01 00:00:00	118	Feb
2	1995-03-01 00:00:00	132	Mar
3	1995-04-01 00:00:00	129	Apr
4	1995-05-01 00:00:00	121	May
5	1995-06-01 00:00:00	135	Jun
6	1995-07-01 00:00:00	148	Jul
7	1995-08-01 00:00:00	148	Aug
8	1995-09-01 00:00:00	136	Sep
9	1995-10-01 00:00:00	119	Oct
10	1995-11-01 00:00:00	104	Nov
11	1995-12-01 00:00:00	118	Dec
12	1996-01-01 00:00:00	115	Jan
13	1996-02-01 00:00:00	126	Feb
14	1996-03-01 00:00:00	141	Mar
15	1996-04-01 00:00:00	135	Apr
16	1996-05-01 00:00:00	125	May
17	1996-06-01 00:00:00	149	Jun
18	1996-07-01 00:00:00	170	Jul
19	1996-08-01 00:00:00	170	Aug
20	1996-09-01 00:00:00	158	Sep
21	1996-10-01 00:00:00	133	Oct
22	1996-11-01 00:00:00	114	Nov
23	1996-12-01 00:00:00	140	Dec
24	1997-01-01 00:00:00	145	Jan

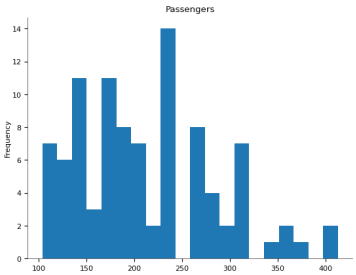
Show 25 per page

1 2 3 4

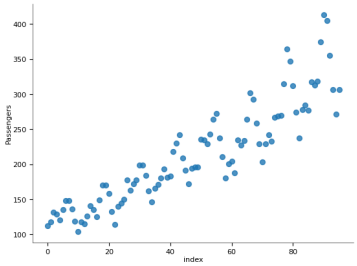
Like what you see? Visit the data table notebook to learn more about interactive tables.

Distributions

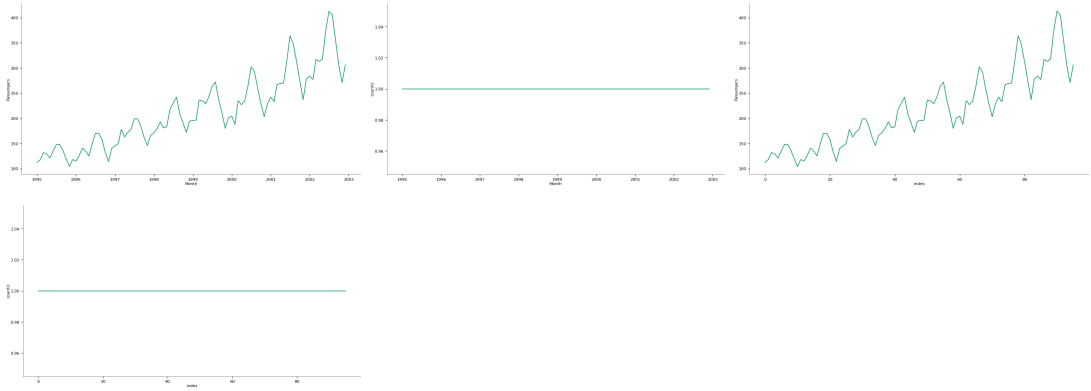




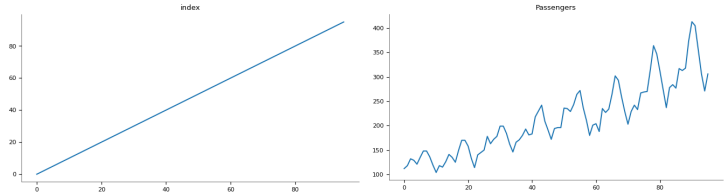
2-d distributions



Time series



Values



Next steps: View recommended plots

Passengers

Month vs Passengers

Month vs count()

Passengers

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96 entries, 0 to 95
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Month       96 non-null     datetime64[ns]
 1   Passengers  96 non-null     int64
dtypes: datetime64[ns](1), int64(1)
memory usage: 1.6 KB
```

```
df.describe()
```

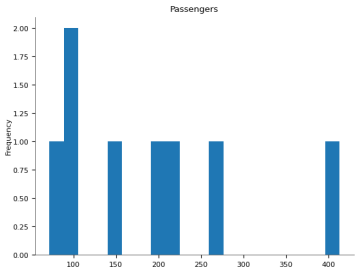
1 to 8 of 8 entries Filter

index	Passengers
count	96.0
mean	213.70833333333334
std	71.91821622204908
min	104.0
25%	156.0
50%	200.0
75%	264.75
max	413.0

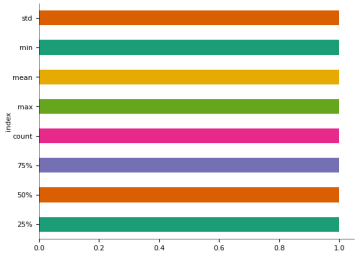
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

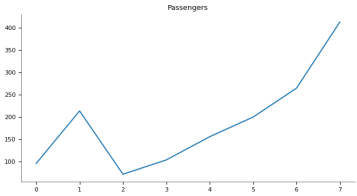
Distributions



Categorical distributions

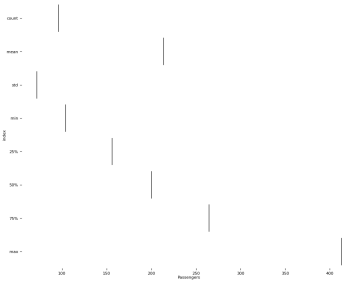


Values



Faceted distributions

```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```

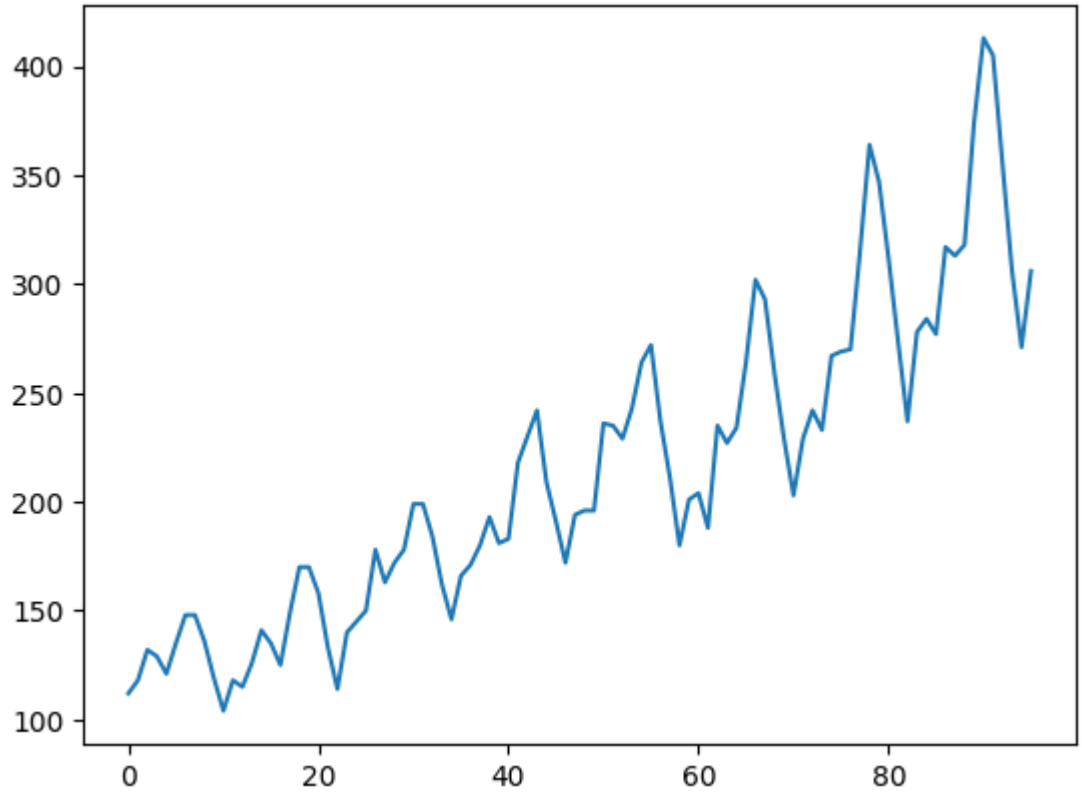


```
df.shape
```

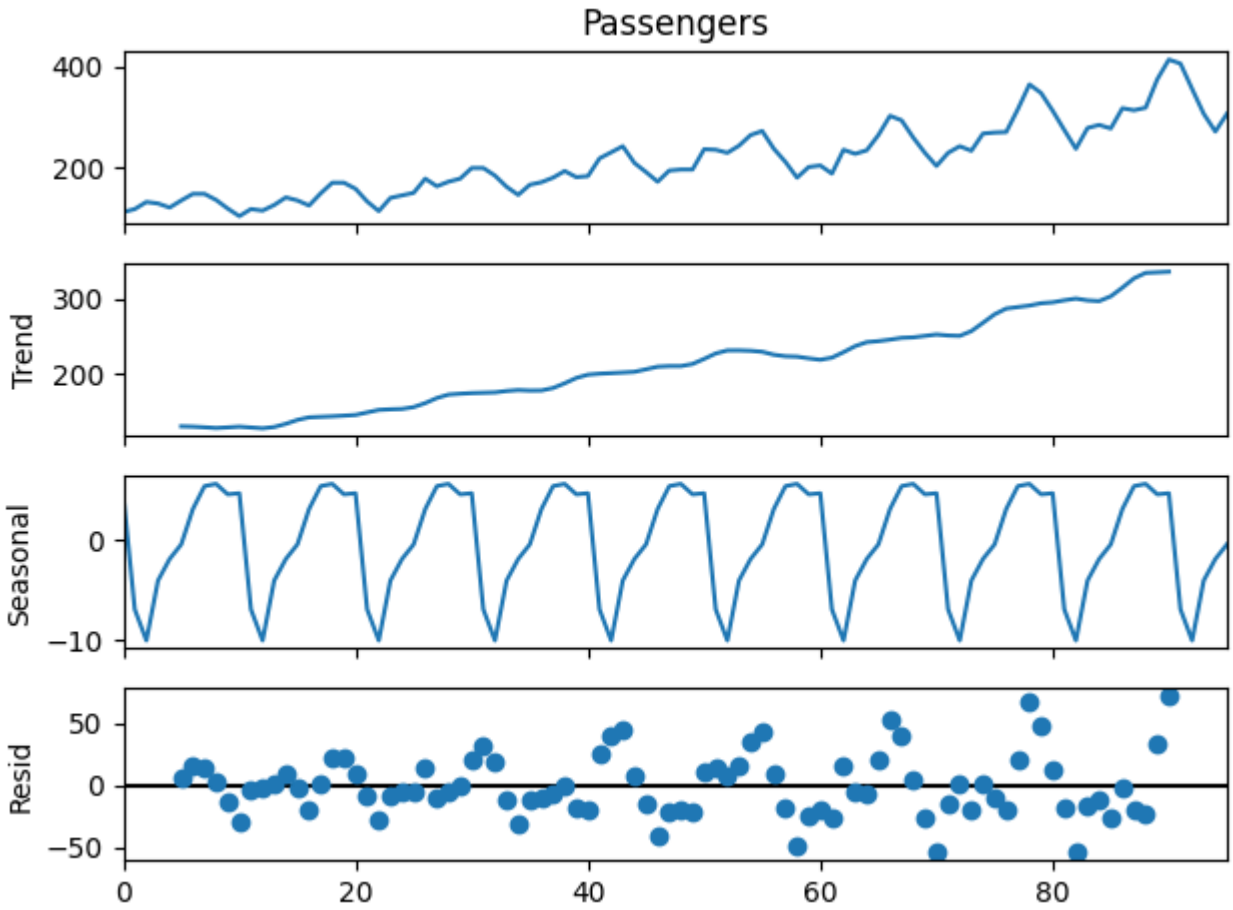
```
(96, 2)
```

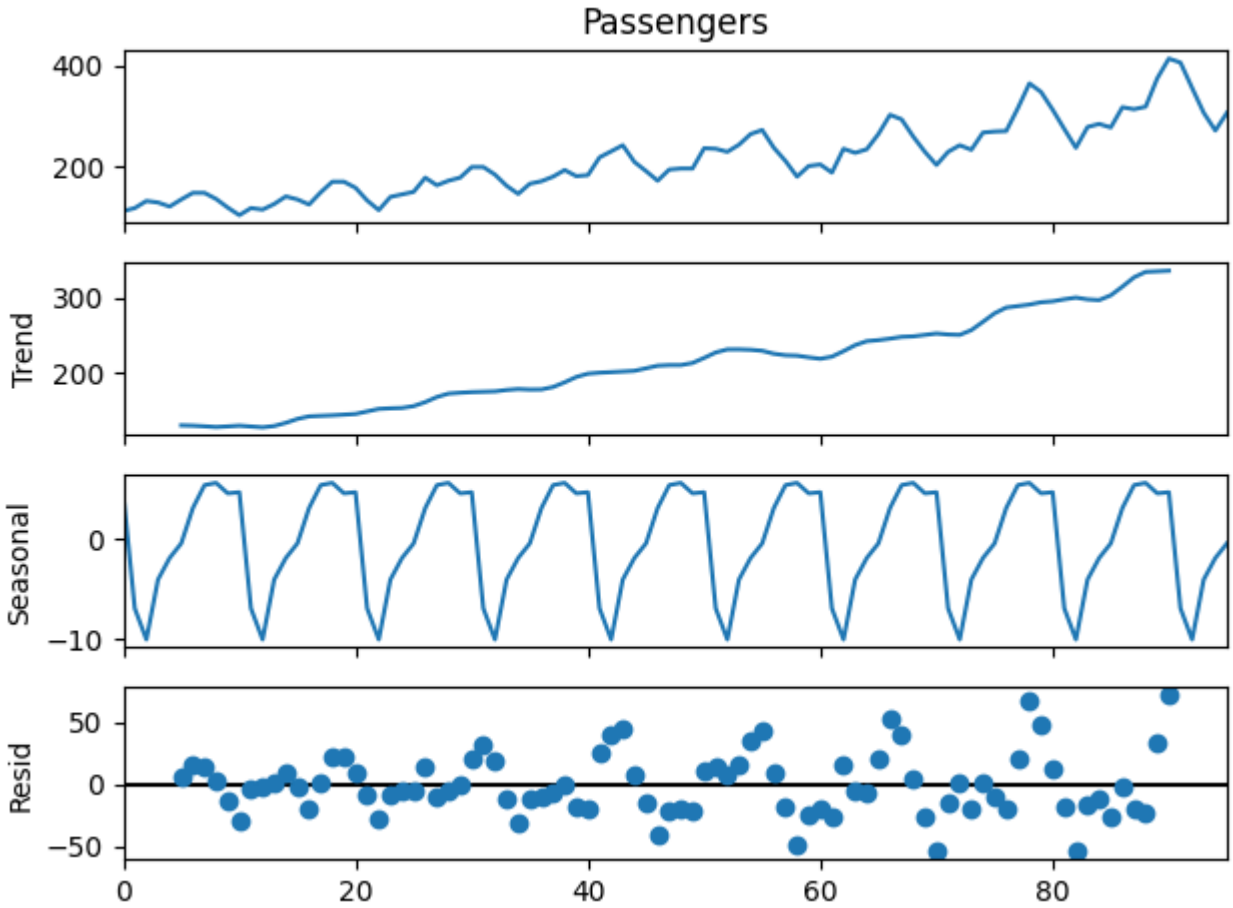
```
df['Passengers'].plot()
```

```
<Axes: >
```




```
import statsmodels.api as smf
from statsmodels.tsa.seasonal import seasonal_decompose
seasonal_ts_add=smf.tsa.seasonal_decompose(df["Passengers"],period=10)
seasonal_ts_add.plot()
```






```
df['Month'] = pd.to_datetime(df['Month'])
df['Months'] = df['Month'].dt.strftime('%b')
df.head()
```

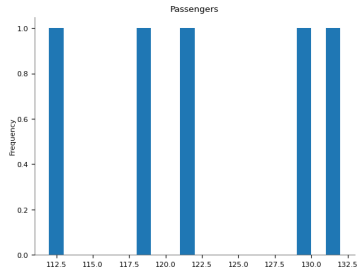
1 to 5 of 5 entries Filter

index	Month	Passengers	Months
0	1995-01-01 00:00:00	112	Jan
1	1995-02-01 00:00:00	118	Feb
2	1995-03-01 00:00:00	132	Mar
3	1995-04-01 00:00:00	129	Apr
4	1995-05-01 00:00:00	121	May

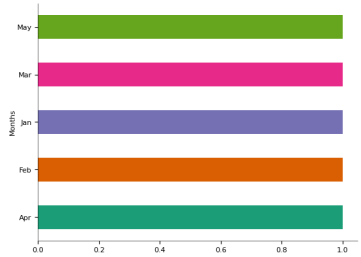
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

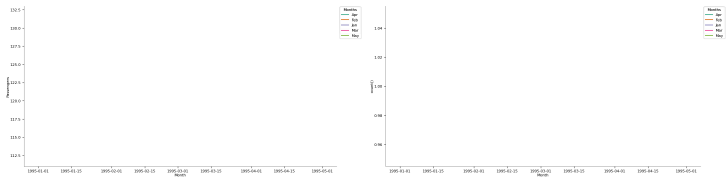
Distributions



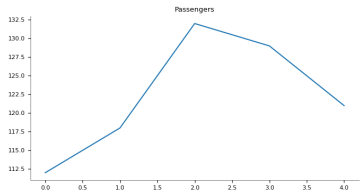
Categorical distributions



Time series

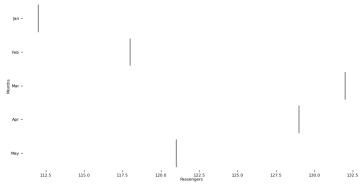


Values

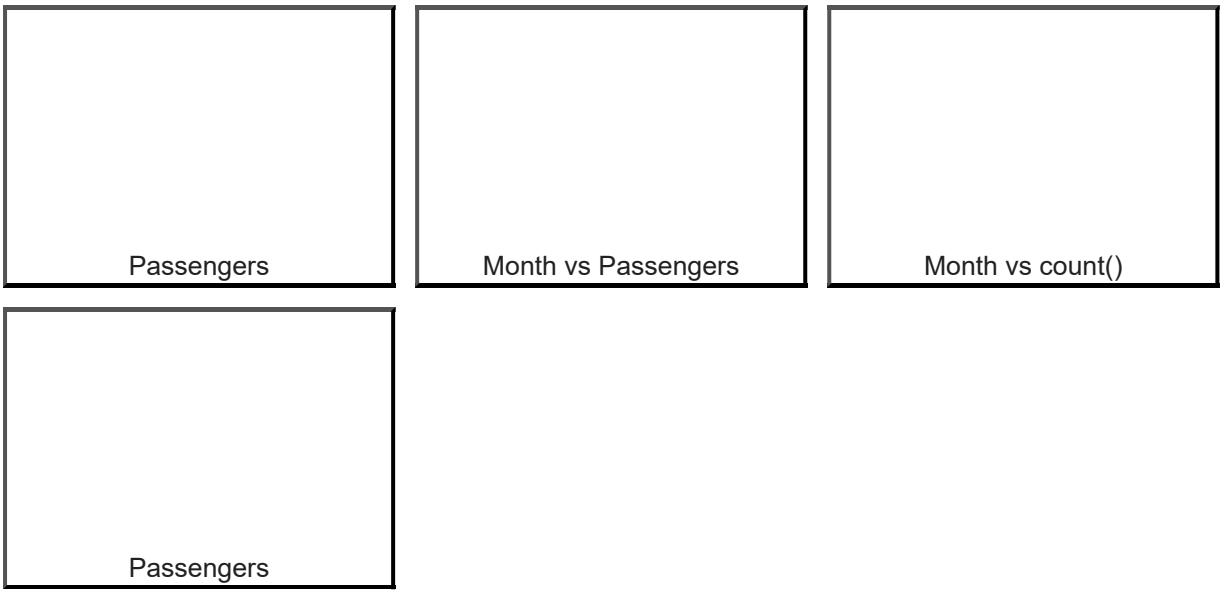


Faceted distributions

```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```

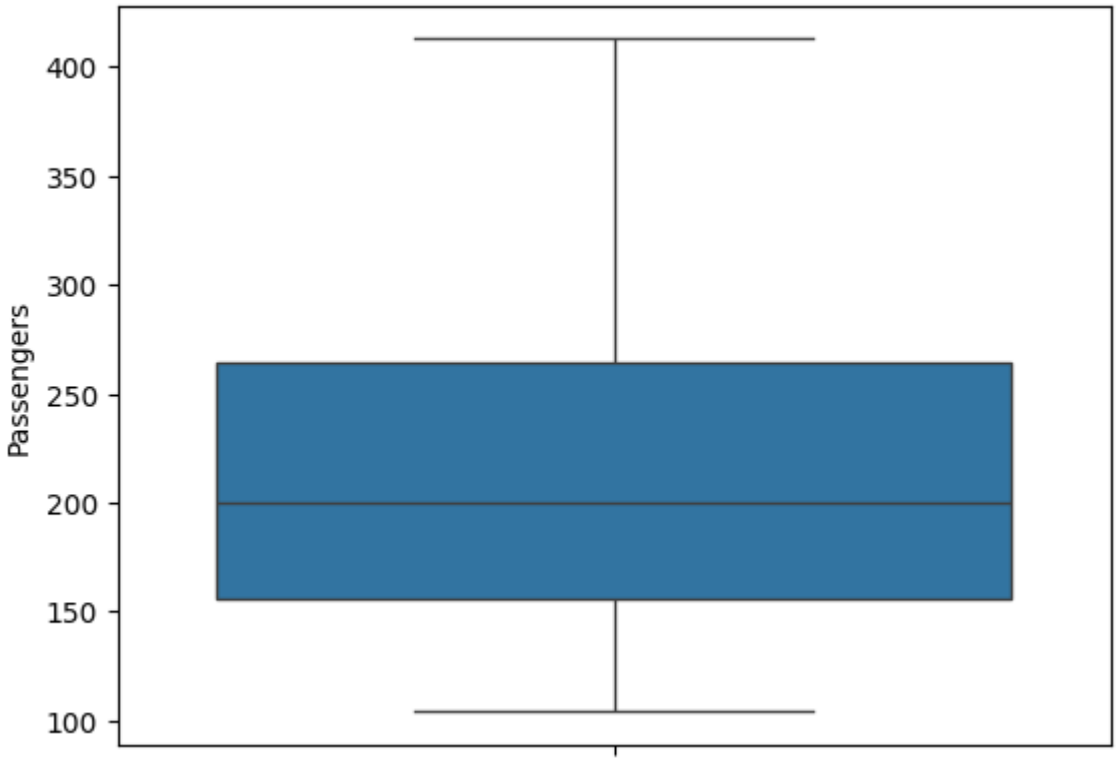


Next steps: View recommended plots




```
sns.boxplot(data=df['Passengers'])
```

```
<Axes: ylabel='Passengers'>
```




```
month_dummies = pd.DataFrame(pd.get_dummies(df['Months']))
```

```
df1 = pd.concat([df,month_dummies],axis = 1)
df1.head()
```

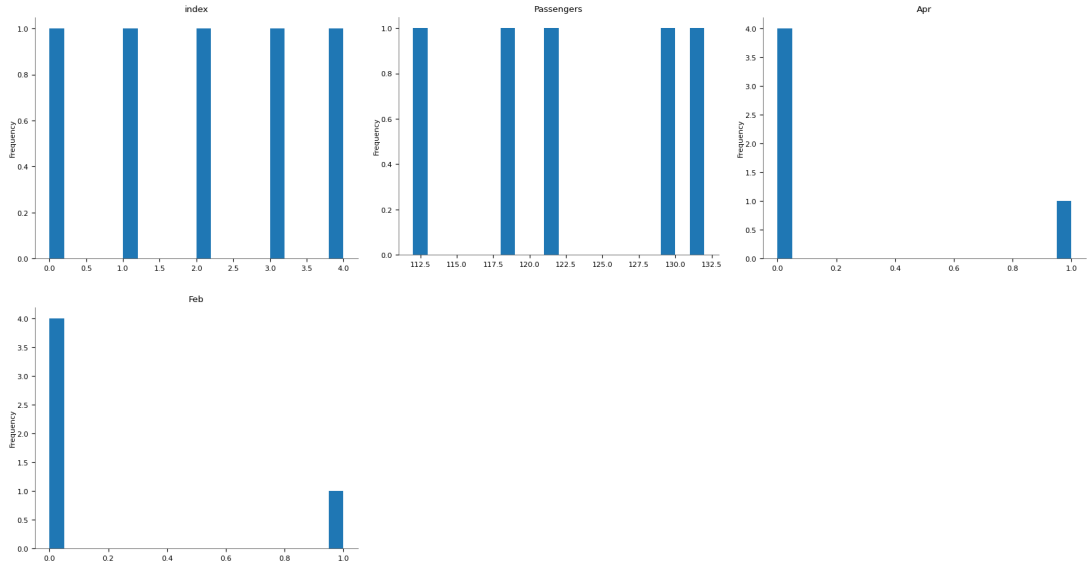
1 to 5 of 5 entries Filter

index	Month	Passengers	Months	Apr	Aug	Dec	Feb	Jan	Jul	Jun	Mar	May	Nov	Oct	Sep
0	1995-01-01 00:00:00	112	Jan	0	0	0	0	1	0	0	0	0	0	0	0
1	1995-02-01 00:00:00	118	Feb	0	0	0	1	0	0	0	0	0	0	0	0
2	1995-03-01 00:00:00	132	Mar	0	0	0	0	0	0	0	1	0	0	0	0
3	1995-04-01 00:00:00	129	Apr	1	0	0	0	0	0	0	0	0	0	0	0
4	1995-05-01 00:00:00	121	May	0	0	0	0	0	0	0	0	1	0	0	0

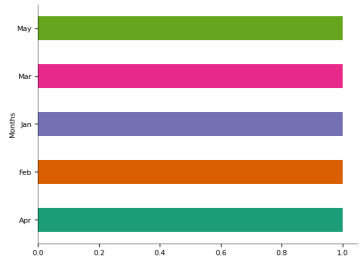
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

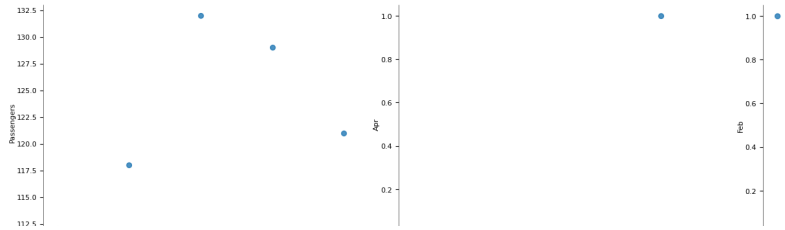
Distributions

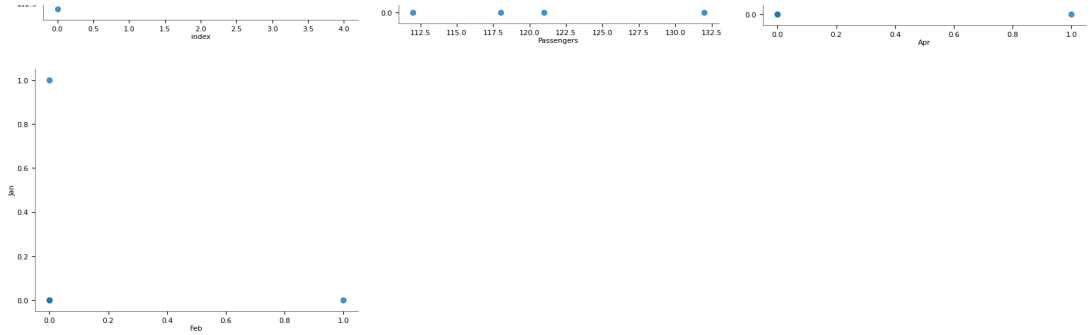


Categorical distributions

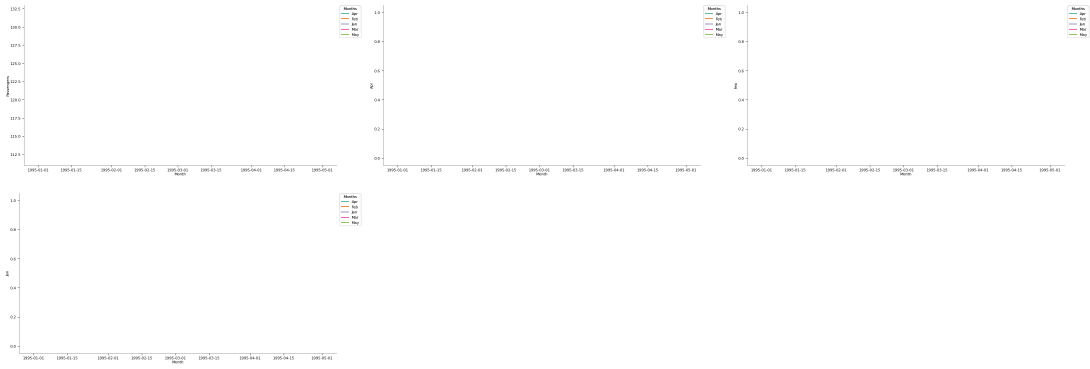


2-d distributions

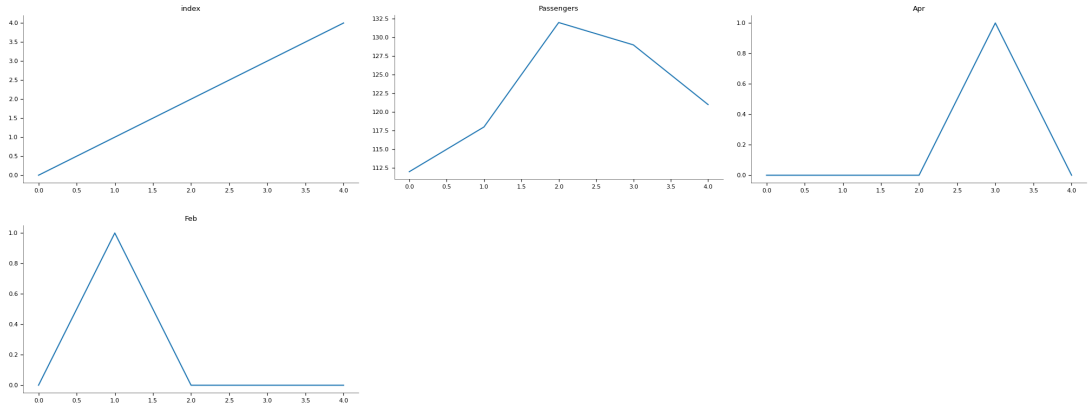




Time series

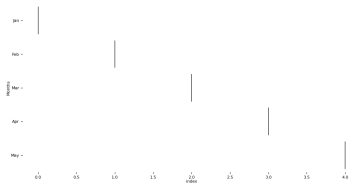


Values

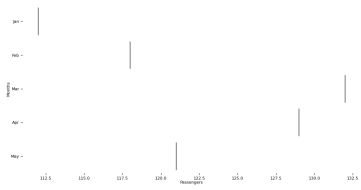


Faceted distributions

```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```

```
<string>:5: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```

Next

View recommended plots

Passengers

Apr

Aug

Dec

Passengers vs Apr

Apr vs Aug

Aug vs Dec

Dec vs Feb

Month vs Passengers

Month vs Apr

Month vs Aug

Month vs Dec

Passengers

Apr

Aug

Dec

```
df1["t"] = np.arange(1,97)
df1["t_squared"] = df1["t"]*df1["t"]
df1["log_Passengers"] = np.log(df1["Passengers"])
df1.columns
df1.head()
```

1 to 5 of 5 entries Filter

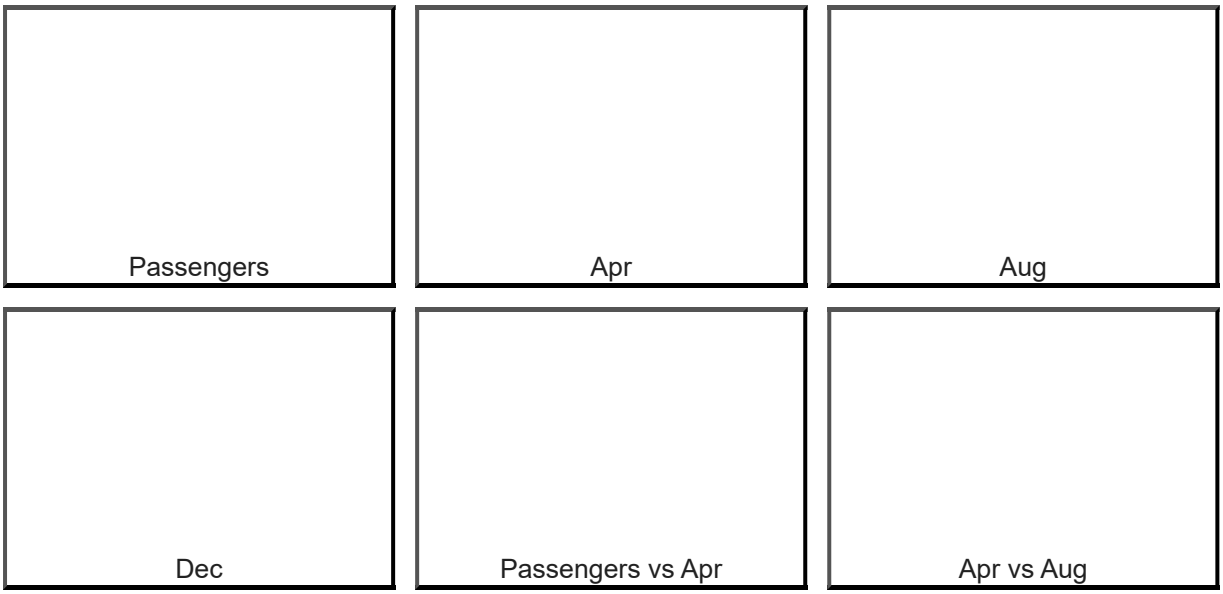
index	Month	Passengers	Months	Apr	Aug	Dec	Feb	Jan	Jul	Jun	Mar	May	Nov	Oct	Sep
0	1995-01-01 00:00:00	112	Jan	0	0	0	0	1	0	0	0	0	0	0	0
1	1995-02-01 00:00:00	118	Feb	0	0	0	1	0	0	0	0	0	0	0	0
2	1995-03-01 00:00:00	132	Mar	0	0	0	0	0	0	0	1	0	0	0	0
3	1995-04-01 00:00:00	129	Apr	1	0	0	0	0	0	0	0	0	0	0	0
4	1995-05-01 00:00:00	121	May	0	0	0	0	0	0	0	0	1	0	0	0

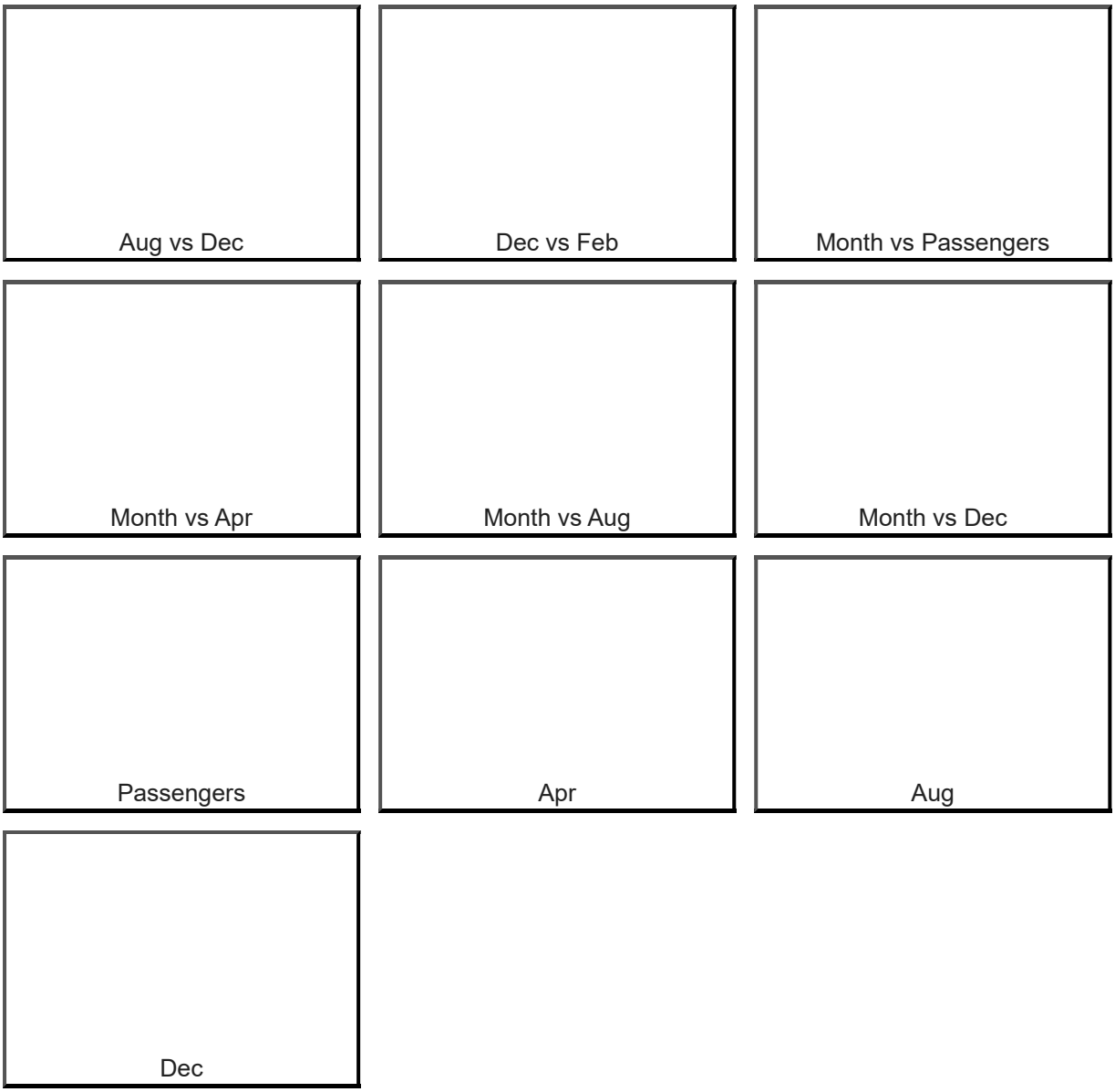
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

```
WARNING: Runtime no longer has a reference to this dataframe, please re-run this cell
```

Next steps: View recommended plots






```
Train = df1.head(75)
Test = df1.tail(25)
```

```
import statsmodels.formula.api as smf
```

```
linear_model = smf.ols('Passengers~t',data=Train).fit()
pred_linear = pd.Series(linear_model.predict(pd.DataFrame(Test['t'])))
rmse_linear = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(pred_linear))**2))
print("RMSE Linear: ",rmse_linear)
```

```
RMSE Linear:  51.83809749584509
```

```
Exp = smf.ols('log_Passengers~t',data=Train).fit()
pred_Exp = pd.Series(Exp.predict(pd.DataFrame(Test['t'])))
rmse_Exp = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(np.exp(pred_Exp)))**2))
print("RMSE Exponential: ",rmse_Exp)
```

```
RMSE Exponential:  42.77525975019796
```

```
Quad = smf.ols('Passengers~t+t_squared',data=Train).fit()
pred_Quad = pd.Series(Quad.predict(Test[["t","t_squared"]]))
rmse_Quad = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(pred_Quad))**2))
print("RMSE Quadratic: ",rmse_Quad)
```

```
RMSE Quadratic:  54.031406456252654
```

```
add_sea = smf.ols('Passengers~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec',data=Train
pred_add_sea = pd.Series(add_sea.predict(Test))
rmse_add_sea = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(pred_add_sea))**2))
print("RMSE Additive seasonality: ",rmse_add_sea)
```

```
RMSE Additive seasonality:  123.02763788084235
```

```
add_sea_Quad = smf.ols('Passengers~t+t_squared+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+No
pred_add_sea_quad = pd.Series(add_sea_Quad.predict(Test))
rmse_add_sea_quad = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(pred_add_sea_q
print("RMSE Additive Seasonality Quadratic:",rmse_add_sea_quad )
```

```
RMSE Additive Seasonality Quadratic: 36.536274445472245
```

```
Mul_sea = smf.ols('log_Passengers~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec',data =
pred_Mult_sea = pd.Series(Mul_sea.predict(Test))
rmse_Mult_sea = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(np.exp(pred_Mult_s
print("RMSE Multiplicative Seasonality:",rmse_Mult_sea)
```

```
RMSE Multiplicative Seasonality: 128.1662281759615
```

```
Mul_Add_sea = smf.ols('log_Passengers~t+Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec',
pred_Mult_add_sea = pd.Series(Mul_Add_sea.predict(Test))
rmse_Mult_add_sea = np.sqrt(np.mean((np.array(Test['Passengers'])-np.array(np.exp(pred_Mu
print("RMSE Multiplicative Additive Seasonality:",rmse_Mult_add_sea )
```

```
RMSE Multiplicative Additive Seasonality: 11.565825437686215
```

```
data1 = {"MODEL":pd.Series(["rmse_linear","rmse_Exp","rmse_Quad","rmse_add_sea","rm
'RMSE Value':pd.Series([rmse_linear,rmse_Exp,rmse_Quad,rmse_add_sea,rmse_add_sea_qu
table_rmse=pd.DataFrame(data1)
table_rmse
```

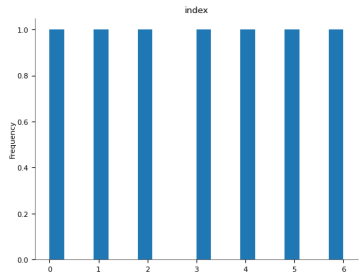
1 to 7 of 7 entries Filter

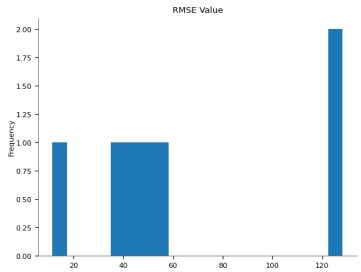
index	MODEL	RMSE Value
0	rmse_linear	51.83809749584509
1	rmse_Exp	42.77525975019796
2	rmse_Quad	54.031406456252654
3	rmse_add_sea	123.02763788084235
4	rmse_add_sea_quad	36.536274445472245
5	rmse_Mult_sea	128.1662281759615
6	rmse_Mult_add_sea	11.565825437686215

Show 25 per page

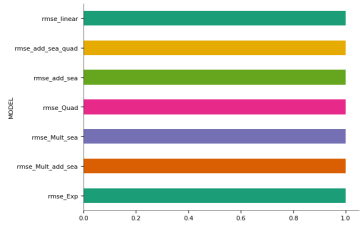
Like what you see? Visit the data table notebook to learn more about interactive tables.

Distributions

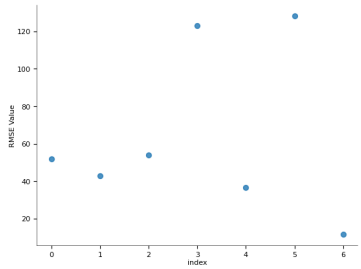




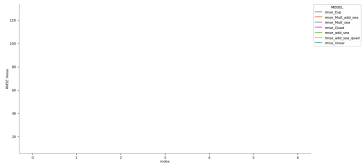
Categorical distributions

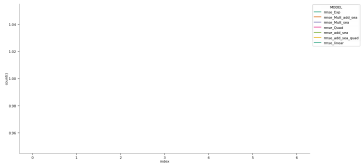


2-d distributions

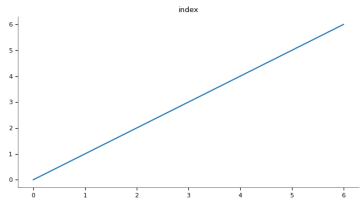


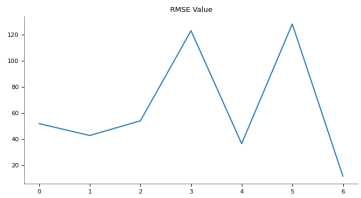
Time series





Values

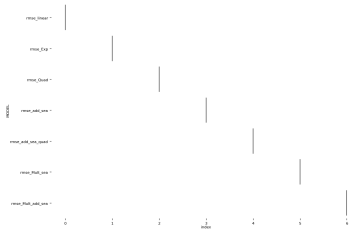




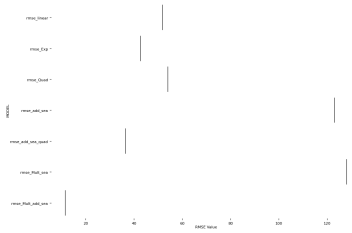
Faceted distributions

```
<string>:5: FutureWarning:
```

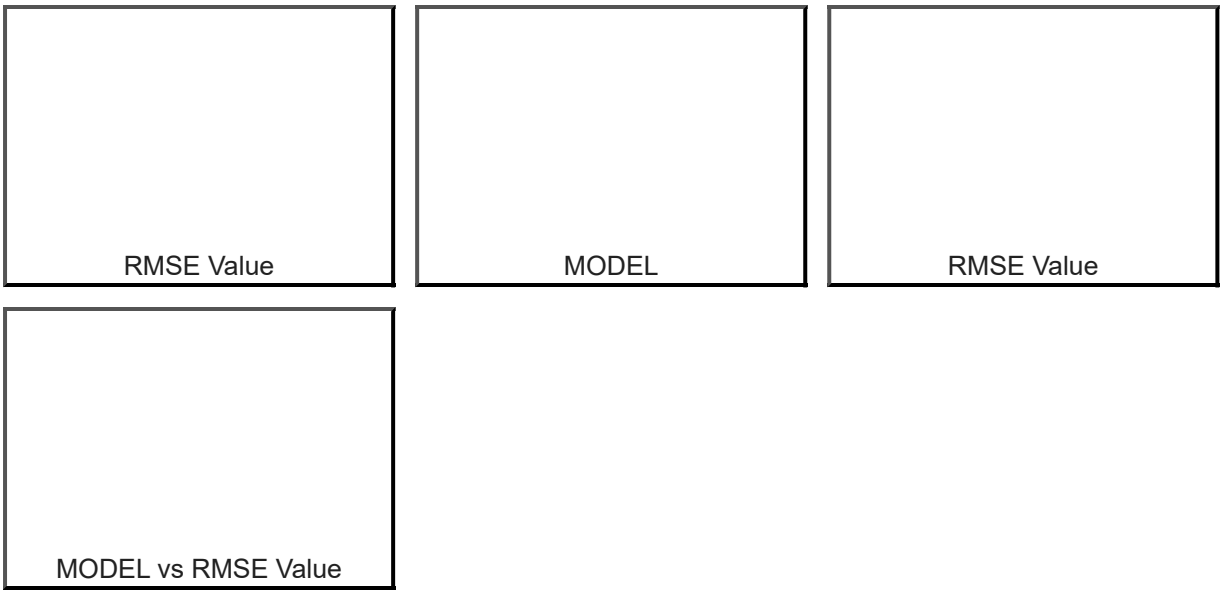
```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

                                    <string>:5: FutureWarning:
```



```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
```



Next steps: View recommended plots




```
data = [['2003-01-01', 'Jan'], ['2003-02-01', 'Feb'], ['2003-03-01', 'Mar'],['2003-04-01'
        ['2003-07-01', 'Jul'],['2003-08-01', 'Aug'],['2003-09-01', 'Sep'],['2003-10-01',
forecast = pd.DataFrame(data, columns = ['Date', 'Months'])
forecast
```

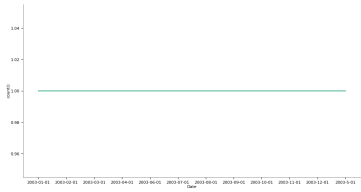
1 to 12 of 12 entries Filter

index	Date	Months
0	2003-01-01	Jan
1	2003-02-01	Feb
2	2003-03-01	Mar
3	2003-04-01	Apr
4	2003-5-01	May
5	2003-06-01	Jun
6	2003-07-01	Jul
7	2003-08-01	Aug
8	2003-09-01	Sep
9	2003-10-01	Oct
10	2003-11-01	Nov
11	2003-12-01	Dec

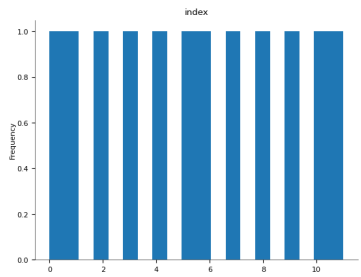
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Time series



Distributions



Time series

```
dummies = pd.DataFrame(pd.get_dummies(forecast['Months']))                    # Create
```