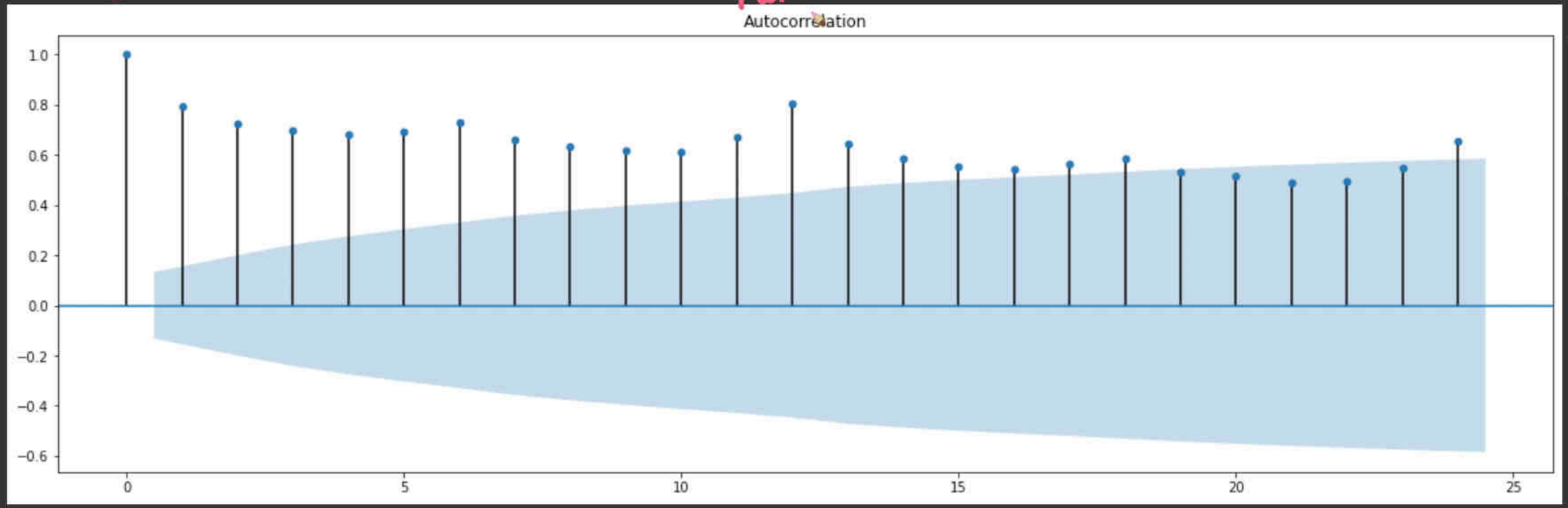
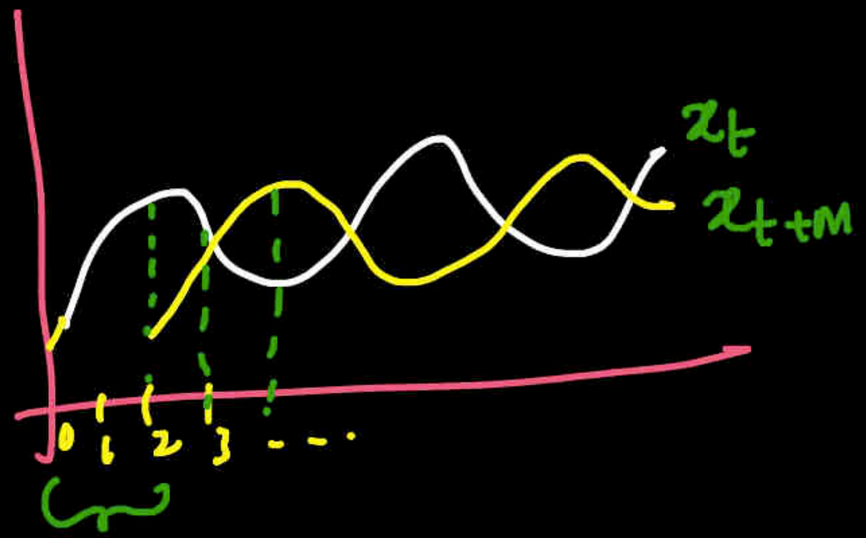
```
[ ] /usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functio
      import pandas.util.testing as tm
```

```
[ ] from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
    plot_acf(mobile_sales.Sales);
```




```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```

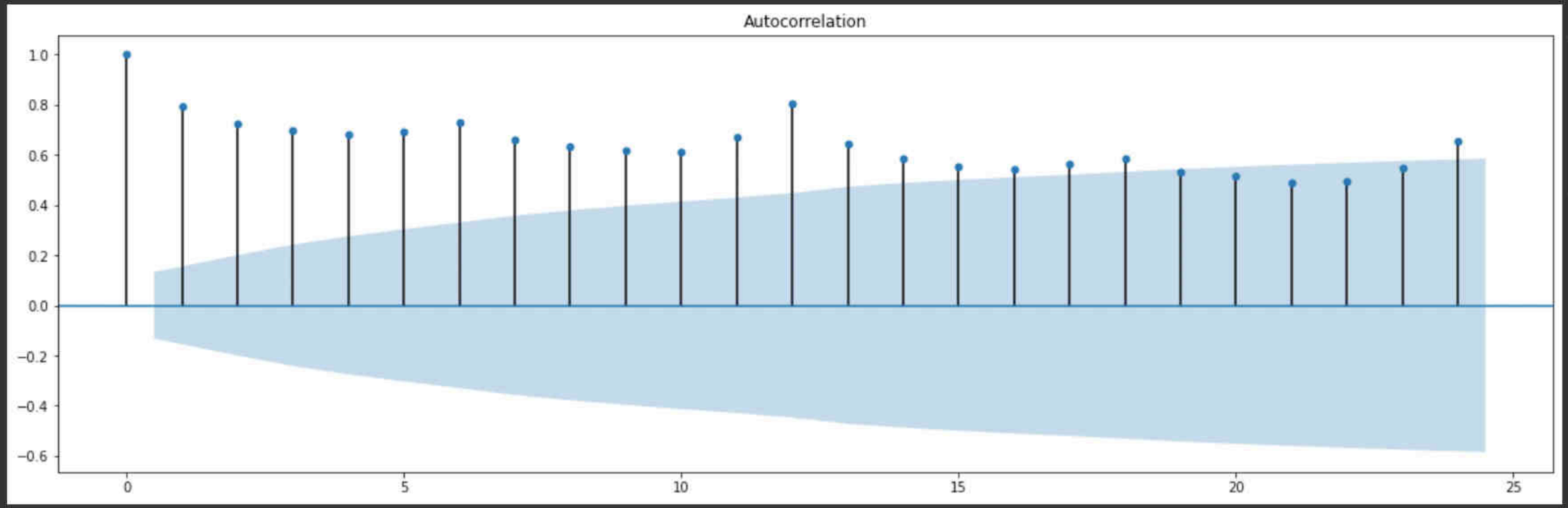
ACF



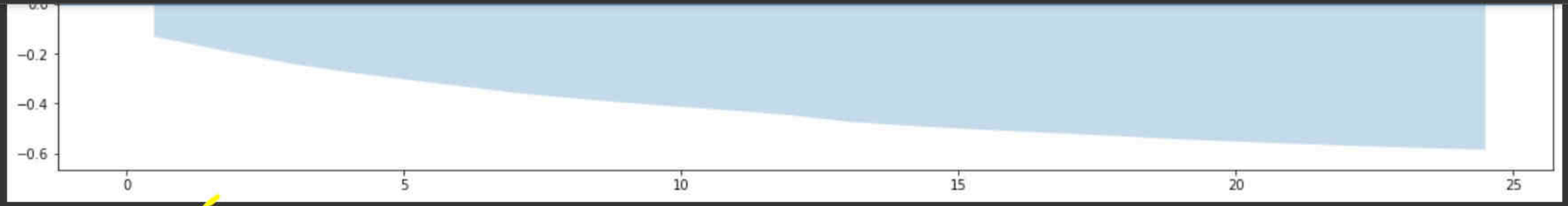
$$\text{Corr}(x_t, x_{t+M})$$

```
[ ] /usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functio
      import pandas.util.testing as tm
```

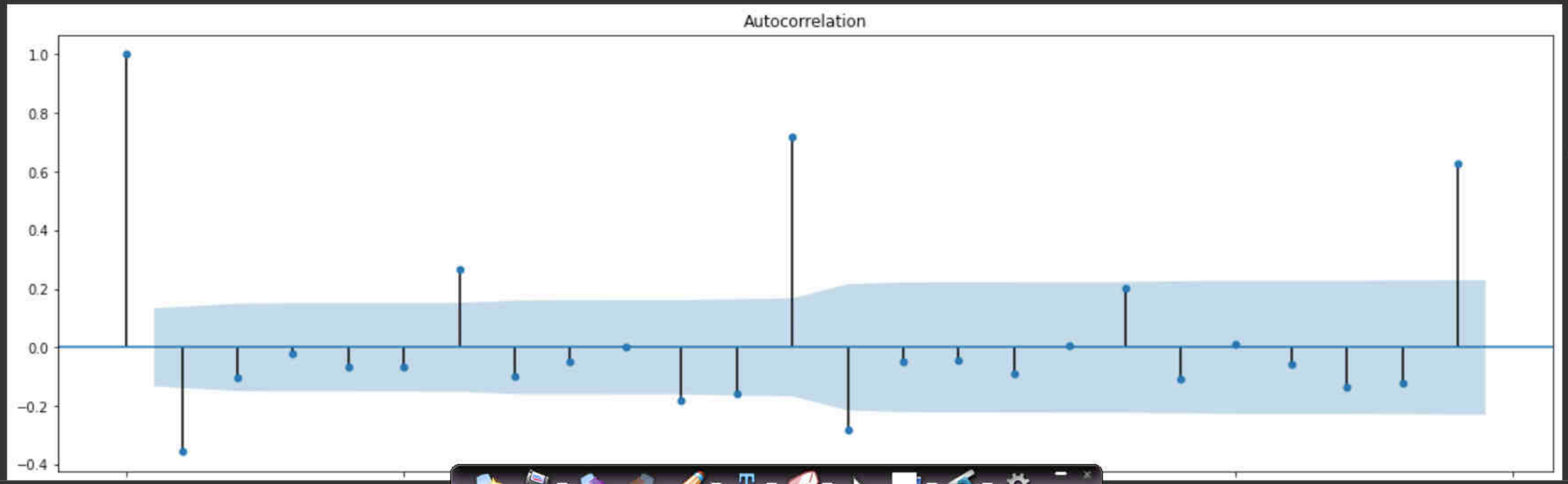
```
[ ] from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
    plot_acf(mobile_sales.Sales);
```




```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```




```
# for detrended time-series
plot_acf(mobile_sales.Sales.diff().dropna());
```



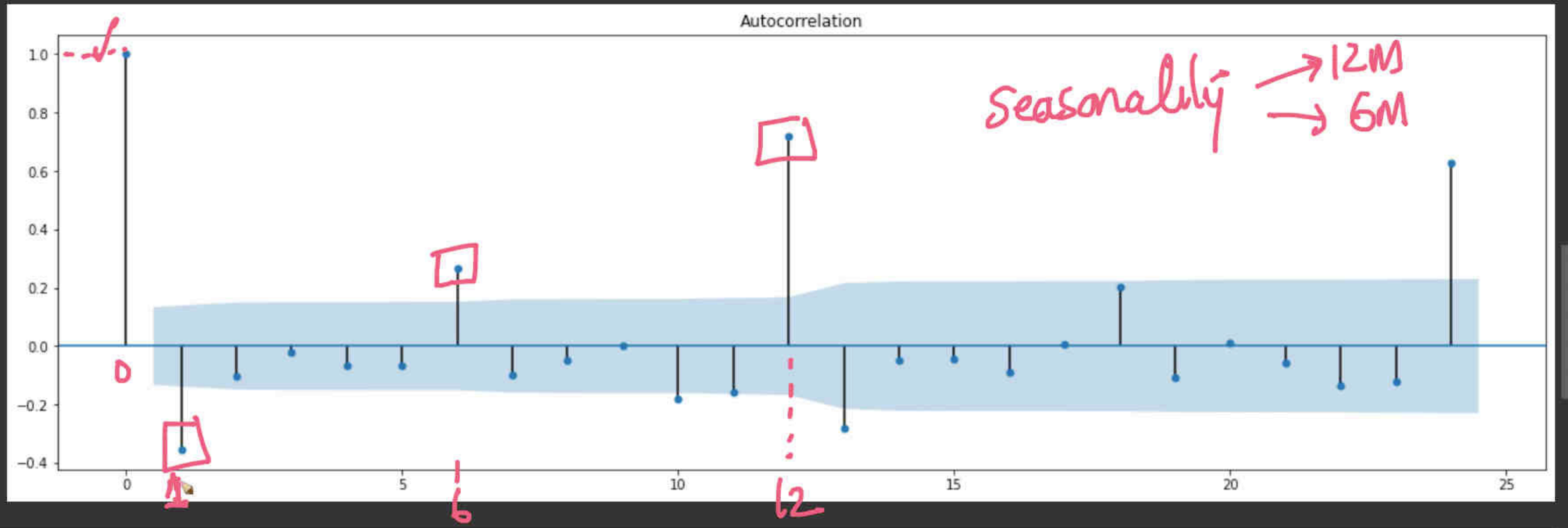
Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:09

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

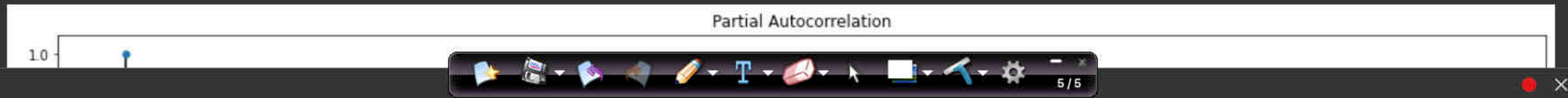
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=KBJlucs4QAP8

+ Code + Text Reconnect

```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```




```
# PACF
plot_pacf(mobile_sales.Sales);
```



5 / 5

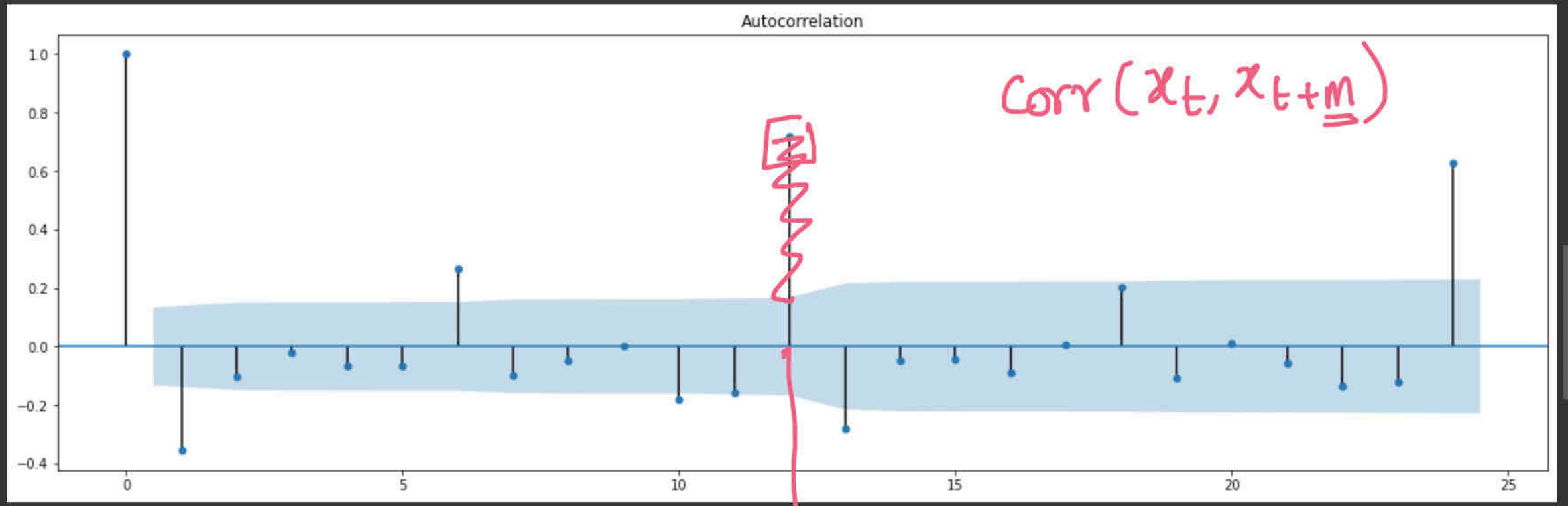
Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

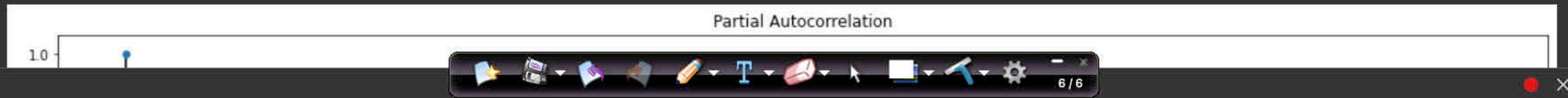
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=KBJlucs4QAP8

+ Code + Text Reconnect

```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```




```
# PACF
plot_pacf(mobile_sales.Sales);
```



6 / 6

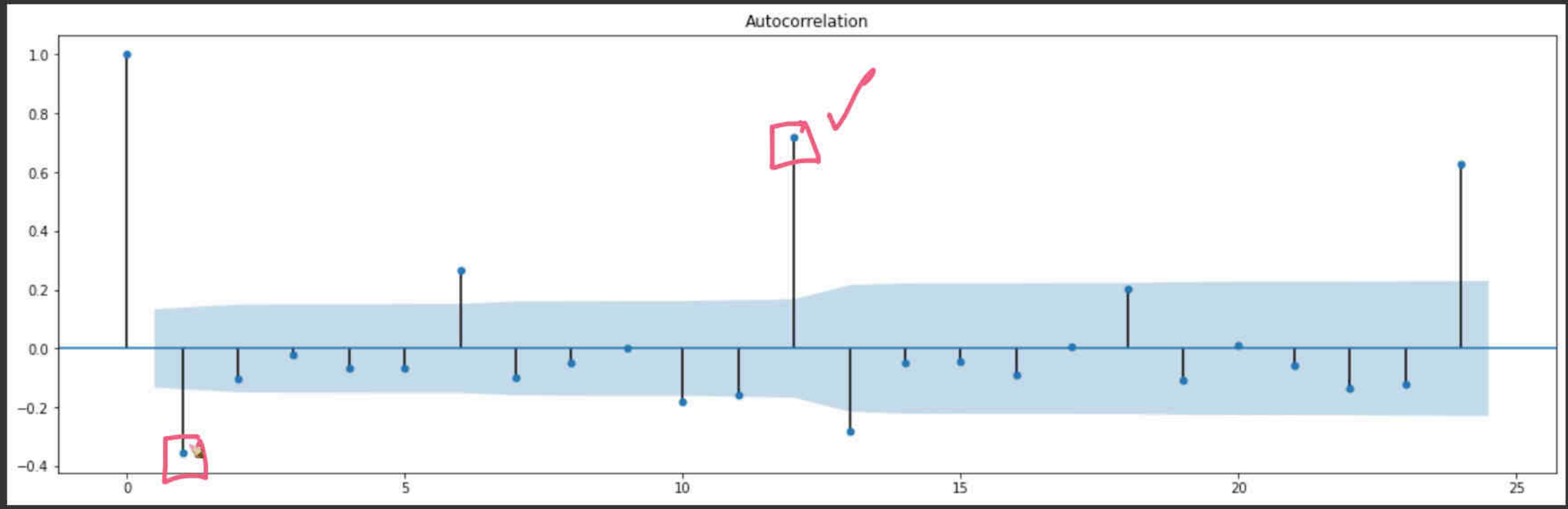
Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

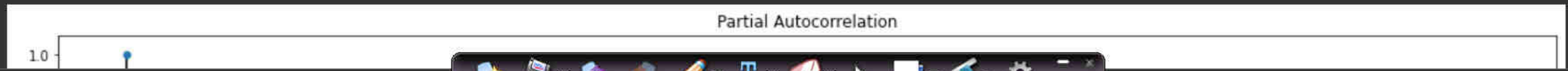
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=KBJlucs4QAP8

+ Code + Text Reconnect

```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```




```
# PACF
plot_pacf(mobile_sales.Sales);
```



7 / 7

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

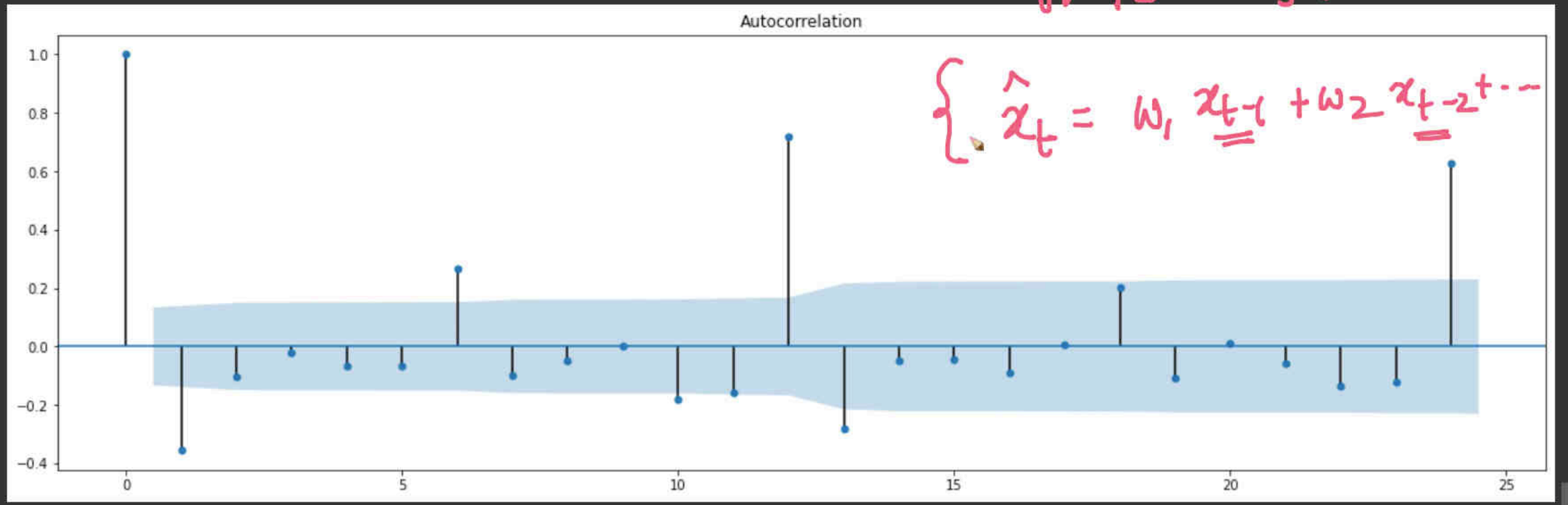
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=0wRN_QsSPBND

+ Code + Text Reconnect

```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```

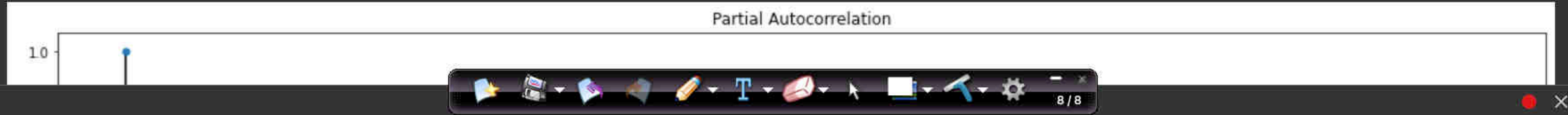
VIF

$f_1\ f_2 \cdots f_d$



$\hat{x}_t = w_1 x_{t-1} + w_2 x_{t-2} + \cdots$

```
# PACF
plot_pacf(mobile_sales.Sales);
```



8 / 8

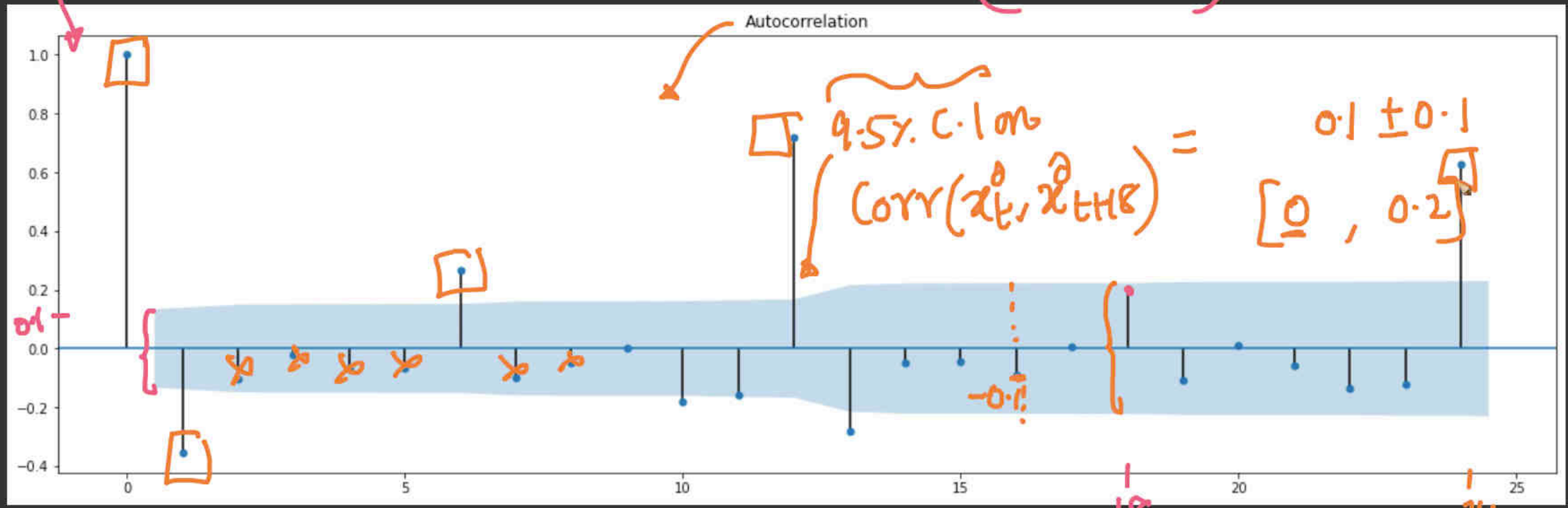
Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=0wRN_QsSPBND

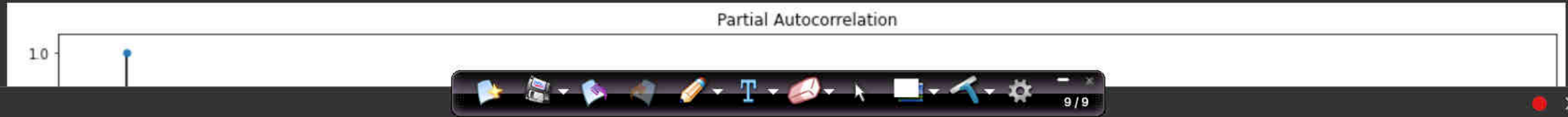
+ Code + Text Reconnect

```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```

Corr $(x_t^d, x_{t+18}^d) = 0.1$




```
[ ] # PACF
    plot_pacf(mobile_sales.Sales);
```



9 / 9

ACF vs PACF

ACF

$Corr(x_t, x_{t+m})$

PACF

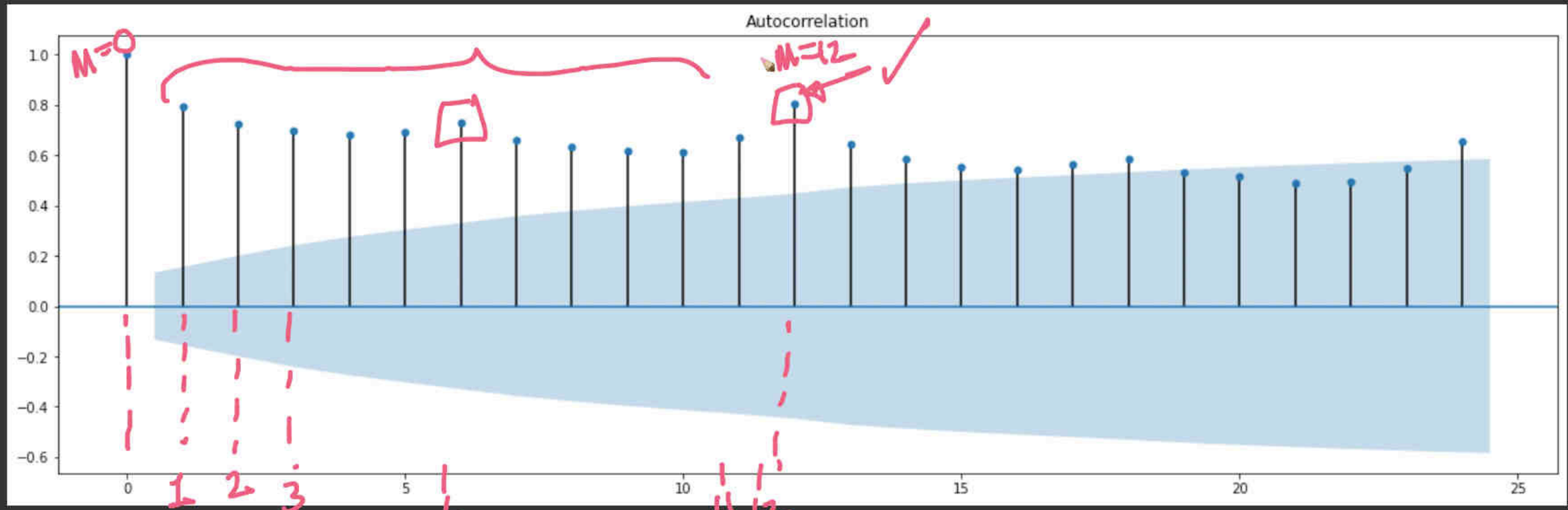
correlation b/w x_t & x_{t+m} (12) after all intermediate correlations have been removed

$m = 1, 2, 3, \ldots 11$

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functio
  import pandas.util.testing as tm
```

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plot_acf(mobile_sales.Sales);
```




```
# for detrended time-series
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=0wRN_QsSPBND

+ Code + Text Reconnect

```
[ ] from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

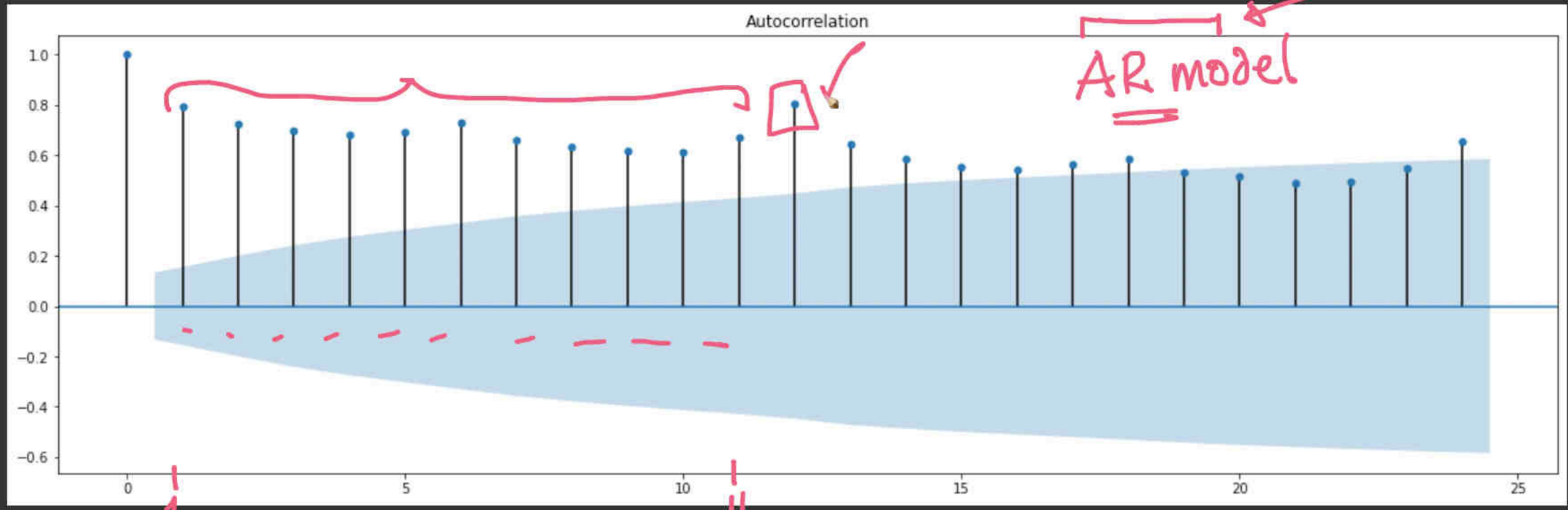
```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functio
  import pandas.util.testing as tm
```

```
[ ] from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
    plot_acf(mobile_sales.Sales);
```

x_t

(later)

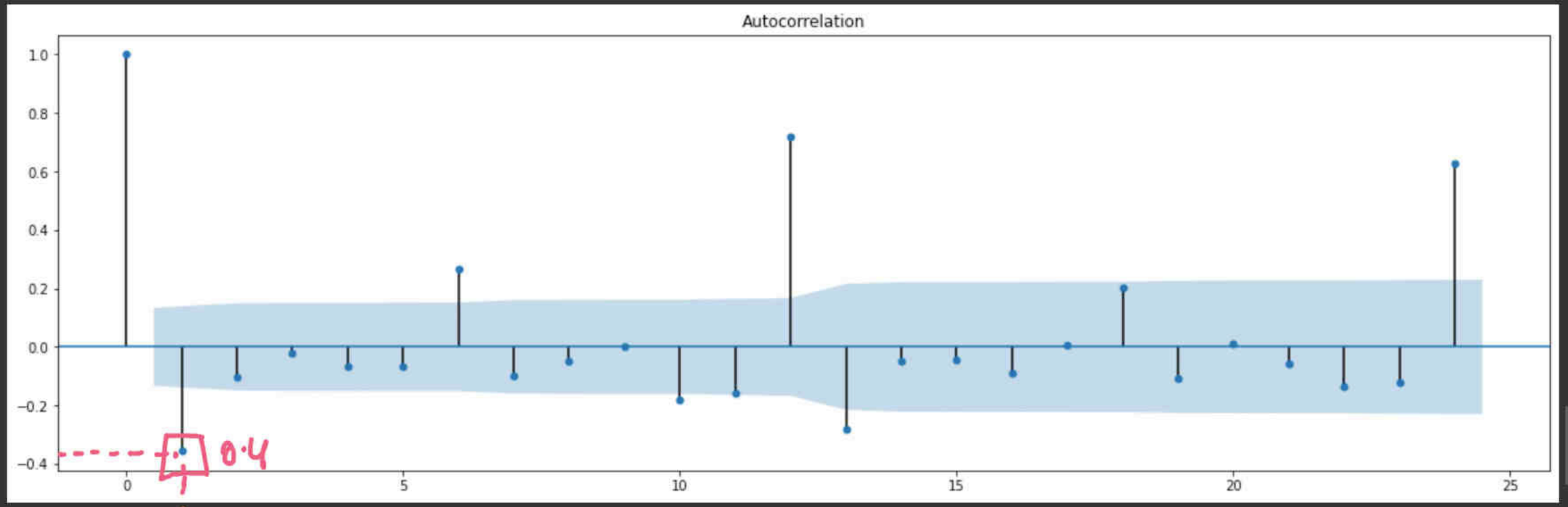
AR model




```
[ ] # for detrended time-series
```

12 / 12


```
[ ] # for detrended time-series
    plot_acf(mobile_sales.Sales.diff().dropna());
```

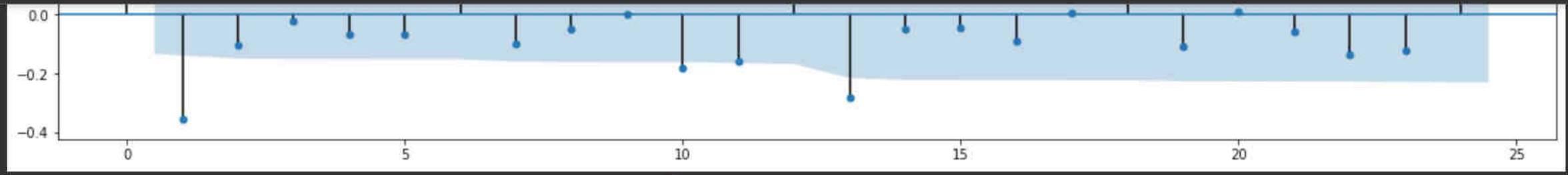



```
# PACF
plot_pacf(mobile_sales.Sales);
```

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

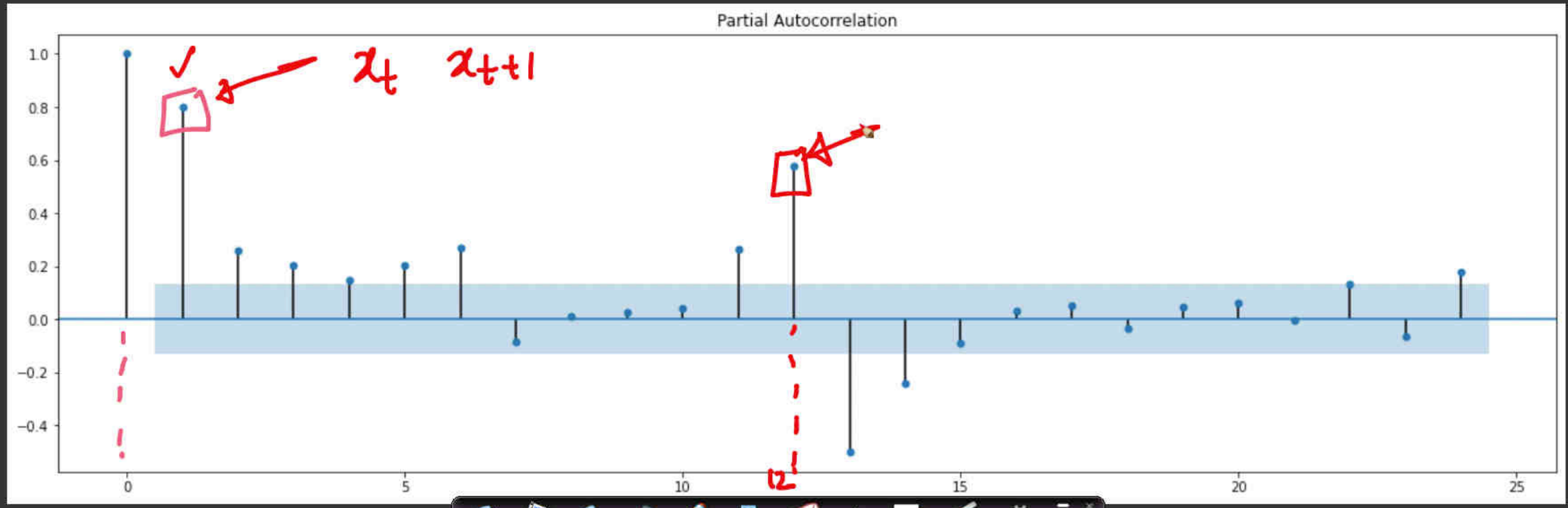
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=0wRN_QsSPBND

+ Code + Text Reconnect




```
[ ] # PACF
    plot_pacf(mobile_sales.Sales);
```

→not detrended



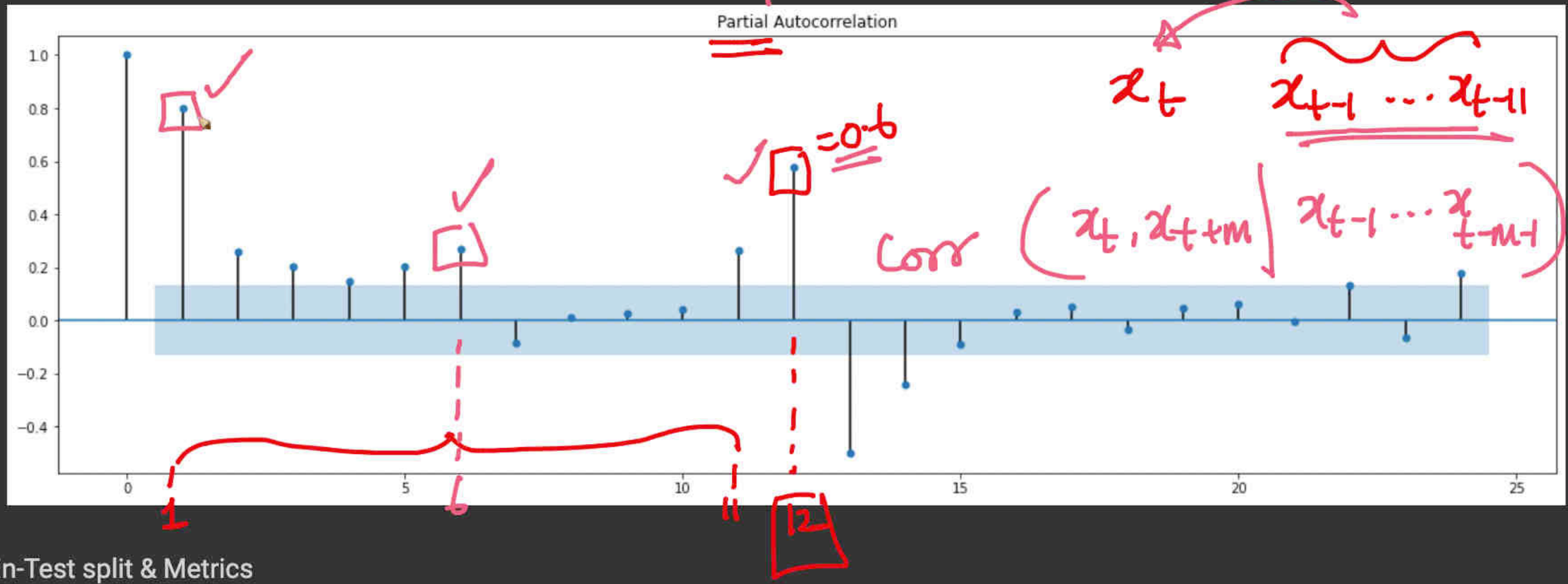
14 / 14

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=0wRN_QsSPBND

+ Code + Text | Reconnect

```
# PACF
plot_pacf(mobile_sales.Sales);
```



Train-Test split & Metrics

[ ] ↳ 2 cells hidden

Models-Foundations

15 / 15

PACF

$$\hat{x}_t = \text{model}(x_{t-1}, \ldots, x_{t-11})$$

$$x_t - \hat{x}_t = e_t$$

$e_t \longrightarrow x_{t-12}$

$\text{Corr}(e_t, x_{t-12})$ = corr. b/w actual TS & lagged TS - 12 after accounting for all intermediate Corr

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

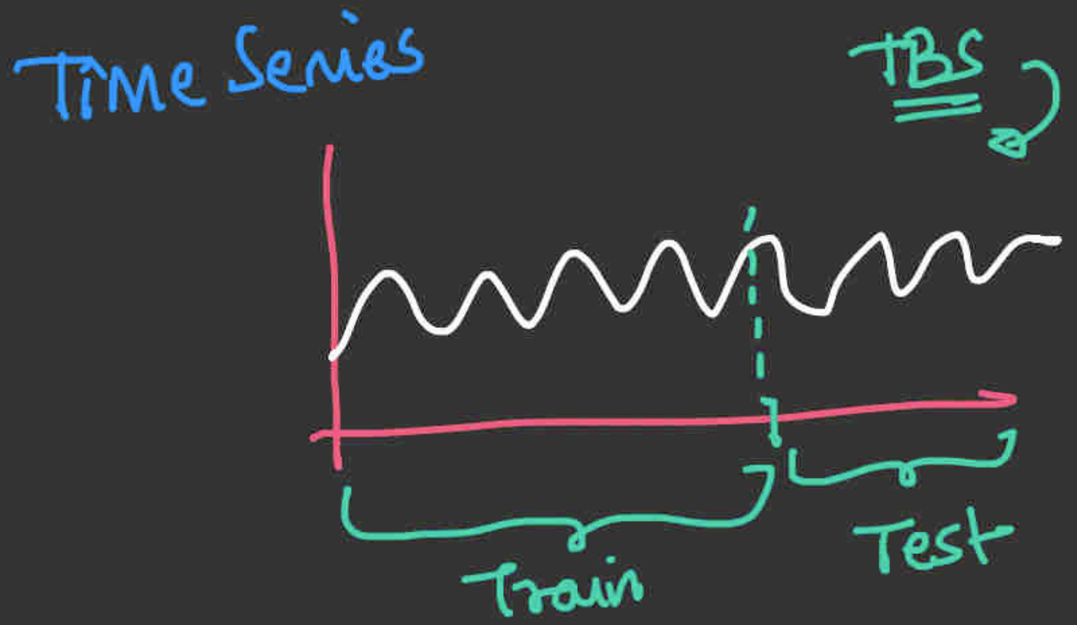
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=eoz59p-GSgCl

+ Code + Text Reconnect

Train-Test split & Metrics

```
#last 12 months as test
train_max_date = mobile_sales.index[-12]
train_x = mobile_sales.loc[mobile_sales.index < mobile_sales.index[-12]].copy()
test_x = mobile_sales.loc[mobile_sales.index >= mobile_sales.index[-12]].copy()

test_x
```

DATE	Sales
2018-02-01	11852.0
2018-03-01	14123.0
2018-04-01	13360.0
2018-05-01	15576.0
2018-06-01	15809.4
2018-07-01	14080.0
2018-08-01	15697.0
2018-09-01	13838.0
2018-10-01	15351.0



17 / 17

Train-Test split & Metrics

Test

run

18yrs 1 month

```
#last 12 months as test
train_max_date = mobile_sales.index[-12]
train_x = mobile_sales.loc[mobile_sales.index < mobile_sales.index[-12]].copy()
test_x = mobile_sales.loc[mobile_sales.index >= mobile_sales.index[-12]].copy()

test_x
```

DATE	Sales
2018-02-01	11852.0
2018-03-01	14123.0
2018-04-01	13360.0
2018-05-01	15576.0
2018-06-01	15809.4
2018-07-01	14080.0
2018-08-01	15697.0
2018-09-01	13838.0
2018-10-01	15351.0

Train-Test split & Metrics

```
#last 12 months as test
train_max_date = mobile_sales.index[-12]
train_x = mobile_sales.loc[mobile_sales.index < mobile_sales.index[-12]].copy()
test_x = mobile_sales.loc[mobile_sales.index >= mobile_sales.index[-12]].copy()

test_x
```

DATE	Sales
2018-02-01	11852.0
2018-03-01	14123.0
2018-04-01	13360.0
2018-05-01	15576.0
2018-06-01	15809.4
2018-07-01	14080.0
2018-08-01	15697.0
2018-09-01	13838.0
2018-10-01	15351.0

```
[ ] from sklearn.metrics import (
        mean_squared_error as mse,
        mean_absolute_error as mae,
        mean_absolute_percentage_error as mape
    )

    # Creating a function to print values of all these metrics.
    def performance(actual, predicted):
        print('MAE :', round(mae(actual, predicted), 3))
        print('RMSE :', round(mse(actual, predicted)**0.5, 3))
        print('MAPE:', round(mape(actual, predicted), 3))
```

500-700 → 20-40

20-40 → 20-40

(Q) Why MAPE ? → 20%

→ robust to outliers

→ relative measure

$$\frac{|x_t - \hat{x}_t|}{x_t}$$

0% → ideal

Models-Foundations

[ ] ↳ 26 cells hidden

ARIMA & SARIMA Models

[ ] ↳ 11 cells hidden

```
[ ] from sklearn.metrics import (
        mean_squared_error as mse,
        mean_absolute_error as mae,
        mean_absolute_percentage_error as mape
    )

    # Creating a function to print values of all these metrics.
    def performance(actual, predicted):
        print('MAE :', round(mae(actual, predicted), 3))
        print('RMSE :', round(mse(actual, predicted)**0.5, 3))
        print('MAPE:', round(mape(actual, predicted), 3))
```

Models-Foundations

[ ] ↳ 26 cells hidden

ARIMA & SARIMA Models

[ ] ↳ 11 cells hidden

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

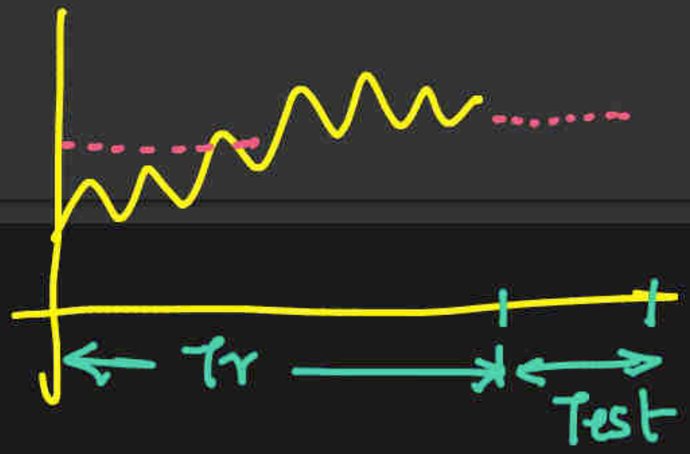
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=XOKdO6C4TCmp

+ Code + Text

Reconnect

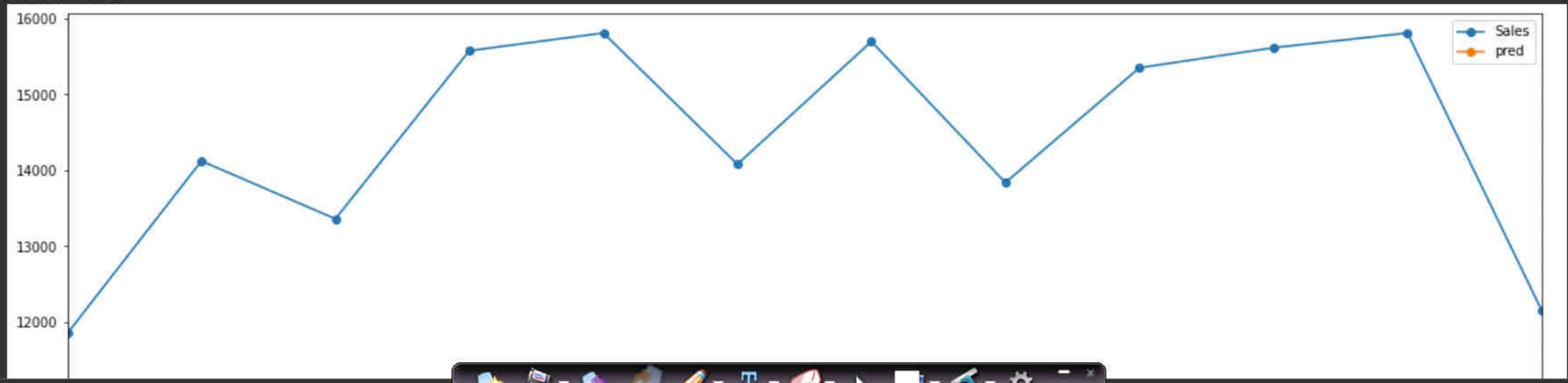
Models-Foundations

Mean model




```
#train mean as test predicted
test_x['pred'] = train_x['Sales'].mean()

test_x.plot(style='-o')

performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 3786.031
RMSE : 4025.906
MAPE: 0.255
```



22 / 22

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=XOKdO6C4TCmp

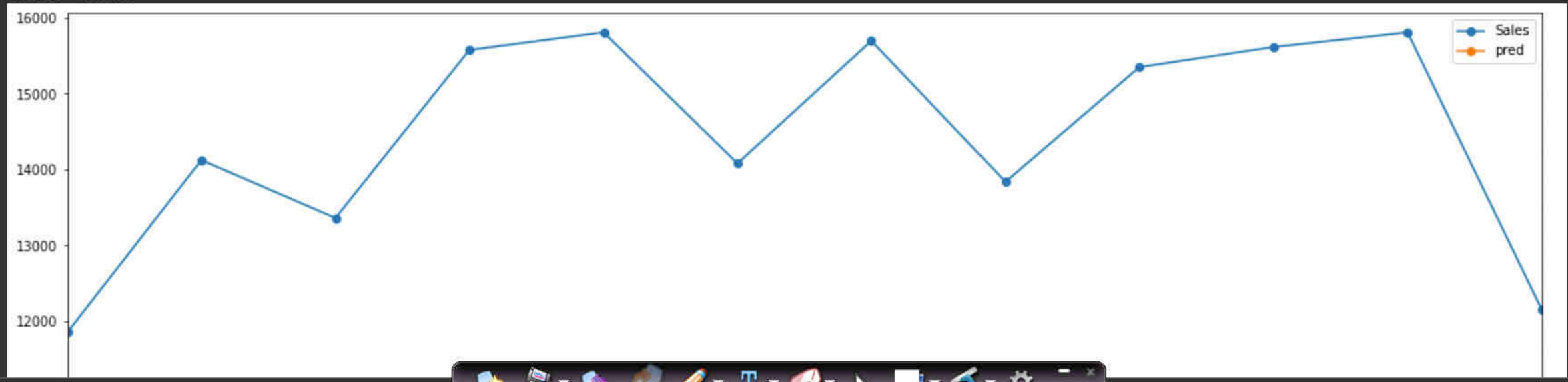
+ Code + Text Reconnect

Models-Foundations

Mean model

```
#train mean as test predicted
test_x['pred'] = train_x['Sales'].mean()

test_x.plot(style='-o')

performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 3786.031
RMSE : 4025.906
MAPE: 0.255
```



23 / 23

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=XOKdO6C4TCmp

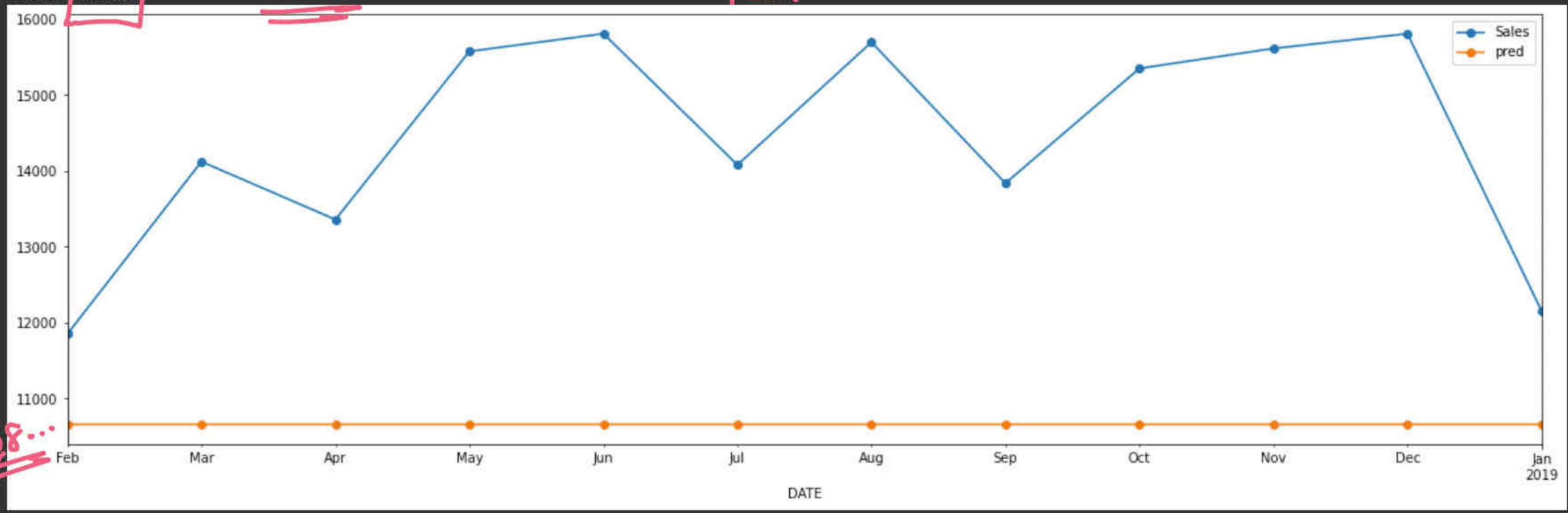
+ Code + Text Reconnect

```
[ ] performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 3786.031
RMSE : 4025.906
MAPE: 0.255
```

25.5%

← Test TS →



108...

Naive forecasts

[ ] ↳ 2 cells hidden

24 / 24

Chrome File Edit View History Bookmarks Profiles Tab Window Help

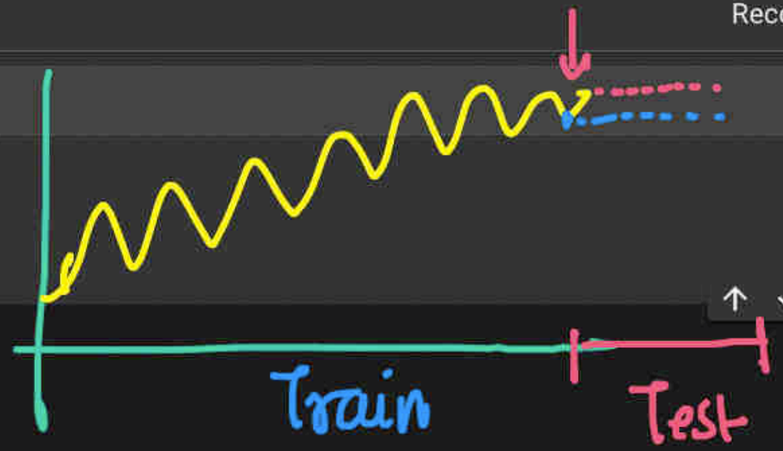
Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=dzJScCFFTYpM

+ Code + Text Reconnect

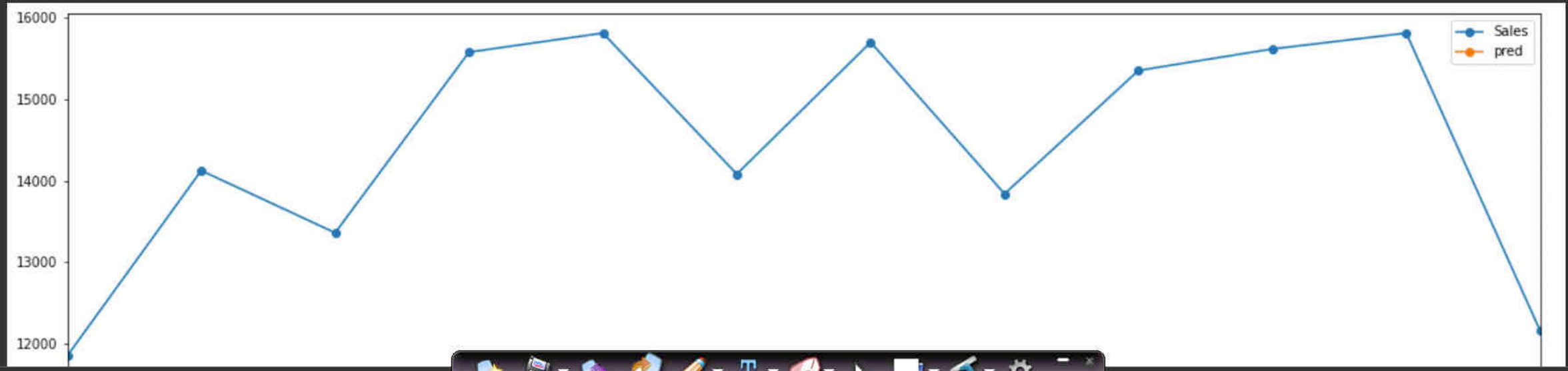
[ ] ↳ 1 cell hidden

Naive forecasts




```
## Last value as the value at all future values: high variance model
test_x['pred'] = train_x['Sales'][-1]

test_x.plot(style='-o')

performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 3434.233
RMSE : 3697.005
MAPE: 0.23
```



25 / 26

Seasonal Naive Forecast

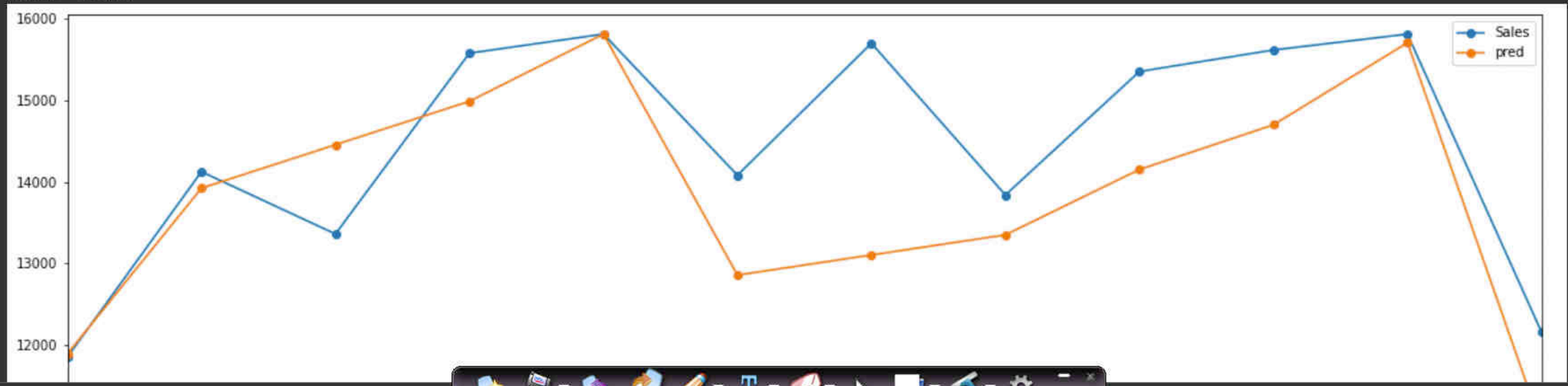
Seasonality
↳ 12 month (PACF)

$$\hat{y}_t = y_{t-12}$$

(using y_t instead of x_t for convinience)


```
[ ] ## Seasonal Naive forecast: 1 year based on ACF and PACF plots
    for i in test_x.index:
      test_x.loc[i]['pred'] = train_x.loc[i - pd.DateOffset(years=1)]['Sales']

    test_x.plot(style='-o')

    performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 800.867
RMSE : 1067.837
MAPE: 0.055
```



Chrome File Edit View History Bookmarks Profiles Tab Window Help

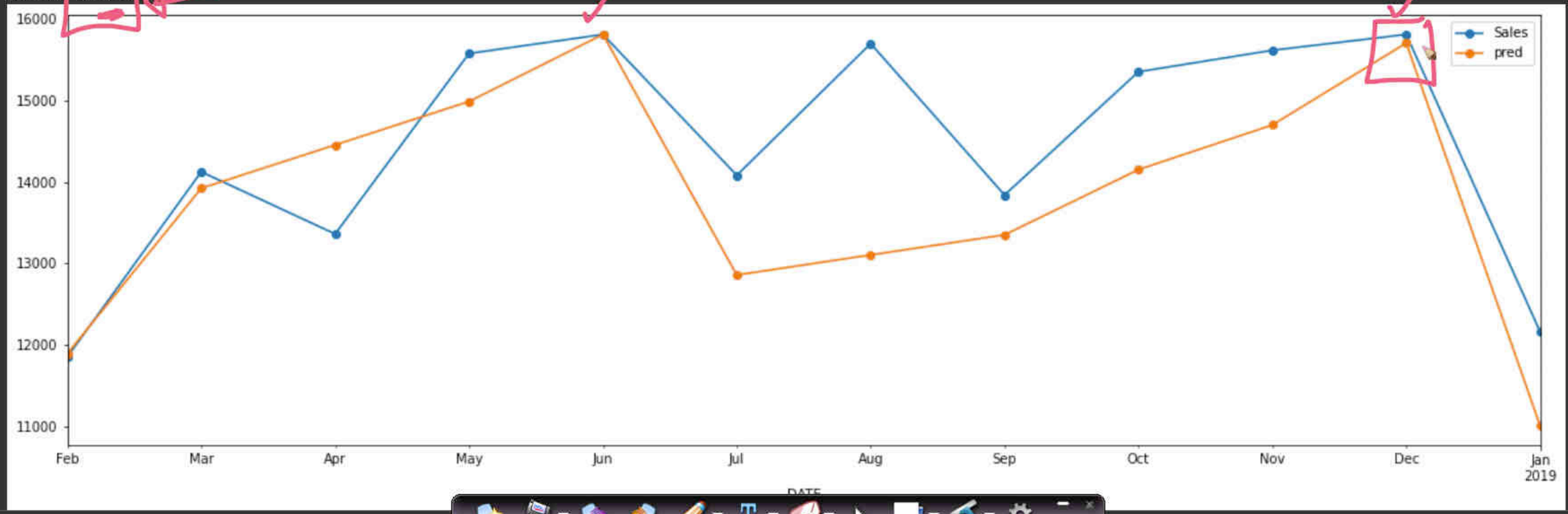
Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=dzJScCFFTYpM

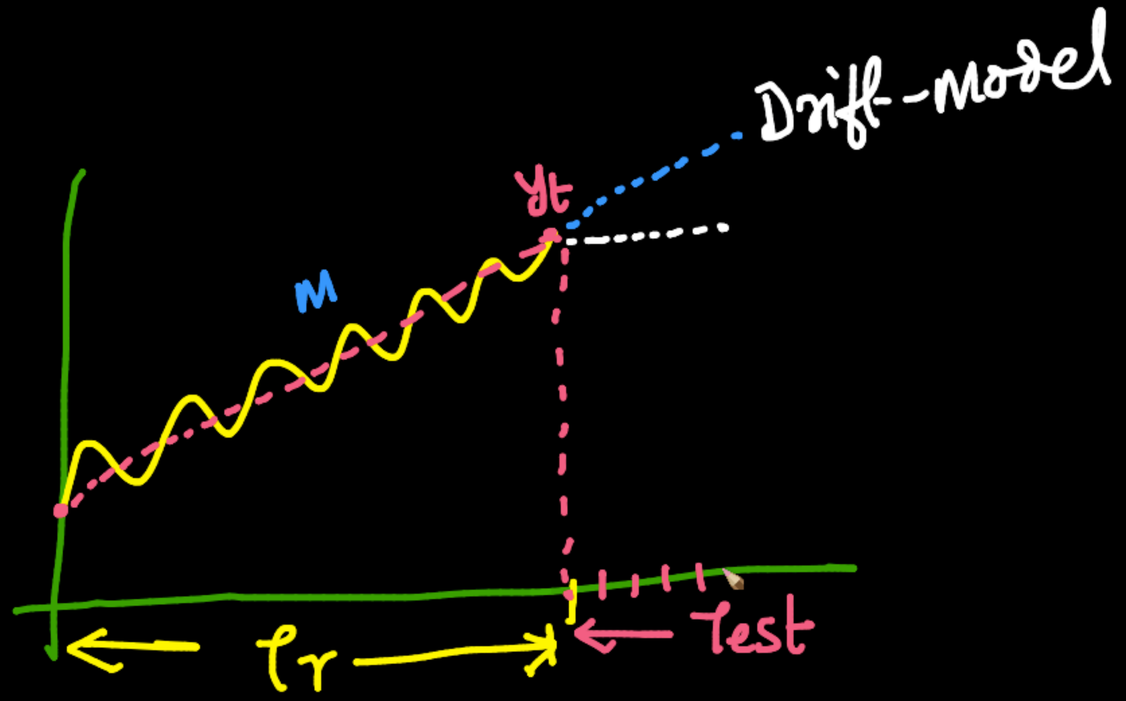
+ Code + Text Reconnect

```
[ ]  ## Seasonal Naive forecast: 1 year based on ACF and PACF plots
     for i in test_x.index:
       test_x.loc[i]['pred'] = train_x.loc[i - pd.DateOffset(years=1)]['Sales']

     test_x.plot(style='-o')

     performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 800.867
RMSE : 1067.837
MAPE: 0.055
```



28 / 29



Drift-model
(Simple)
→ Trend-model

but NOT Seasonality

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=TEu6ZF1HUL1w

+ Code + Text Reconnect

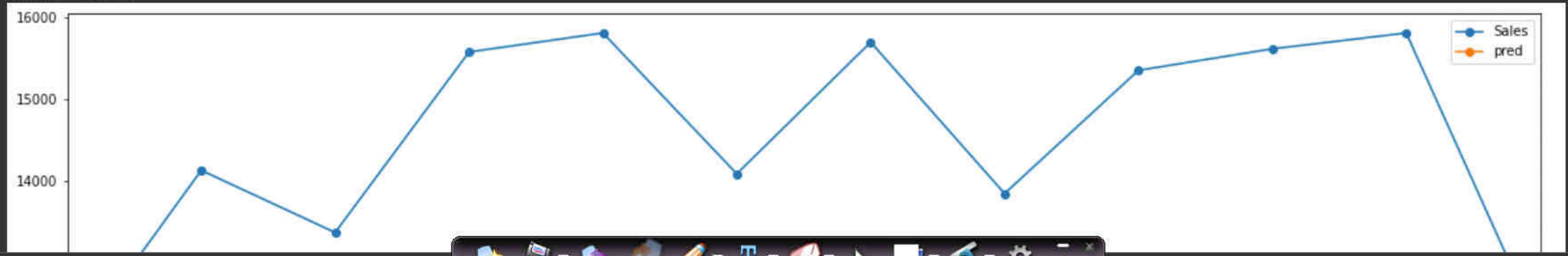
[ ] ↳ 2 cells hidden

Simple Drift Model

```
# Get the slope
y_t = train_x['Sales'][-1]
m = (y_t - train_x['Sales'][0]) / len(train_x)
h = np.linspace(0,len(test_x)-1, len(test_x))

test_x['pred'] = y_t + m * h

test_x.plot(style='-o')

performance(test_x['Sales'], test_x['pred'])
```

1, 2, 3, ~~.

```
MAE : 3321.482
RMSE : 3586.323
MAPE: 0.223
```



30 / 30

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=TEu6ZF1HUL1w

+ Code + Text Reconnect

[ ] ↳ 2 cells hidden

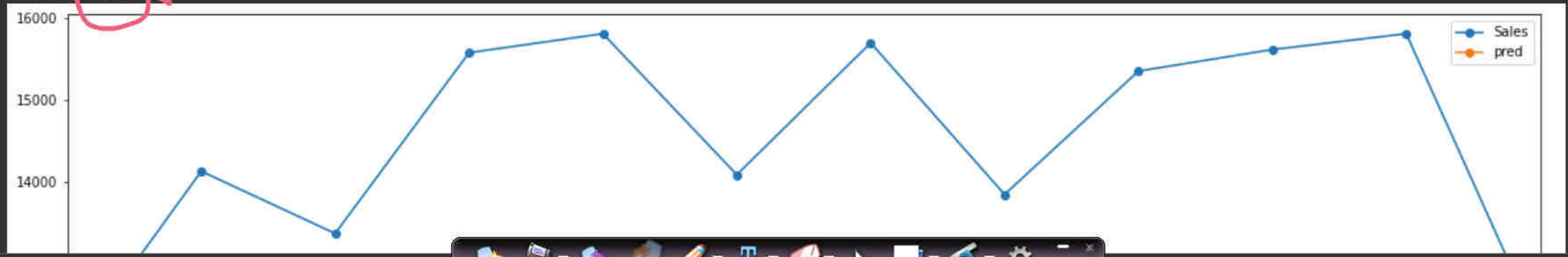
Simple Drift Model

Lesson:

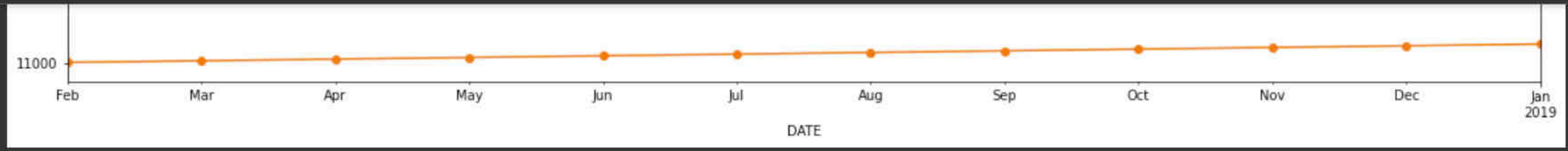
① Trend is less important than Seasonality

```
[ ] # Get the slope
    y_t = train_x['Sales'][-1]
    m = (y_t - train_x['Sales'][0]) / len(train_x)
    h = np.linspace(0,len(test_x)-1, len(test_x))

    test_x['pred'] = y_t + m * h

    test_x.plot(style='-o')

    performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 3321.482
RMSE : 3586.323
MAPE: 0.223
```

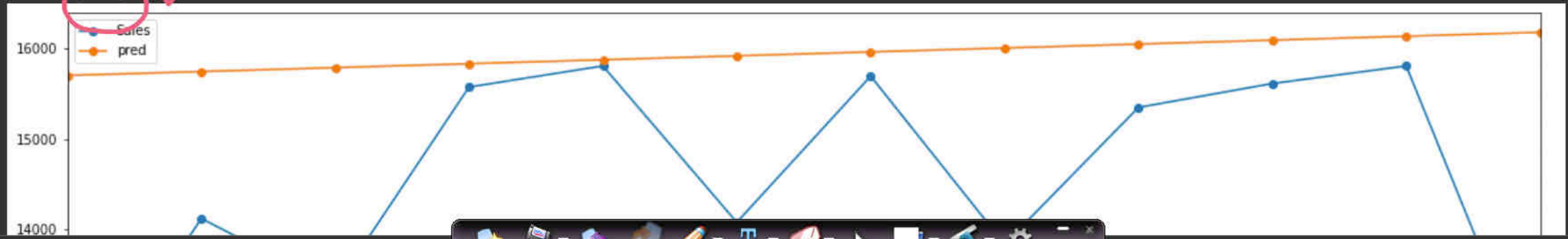


31 / 31




```
# sensitive to last value available
# Get the slope
y_t = train_x['Sales'][-2]
m = (y_t - train_x['Sales'][0]) / len(train_x)
h = np.linspace(0,len(test_x)-1, len(test_x))

test_x['pred'] = y_t + m * h

test_x.plot(style='-o')

performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 1503.589
RMSE : 2013.071
MAPE: 0.115
```



Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:49

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=lGS92kCfcdiS

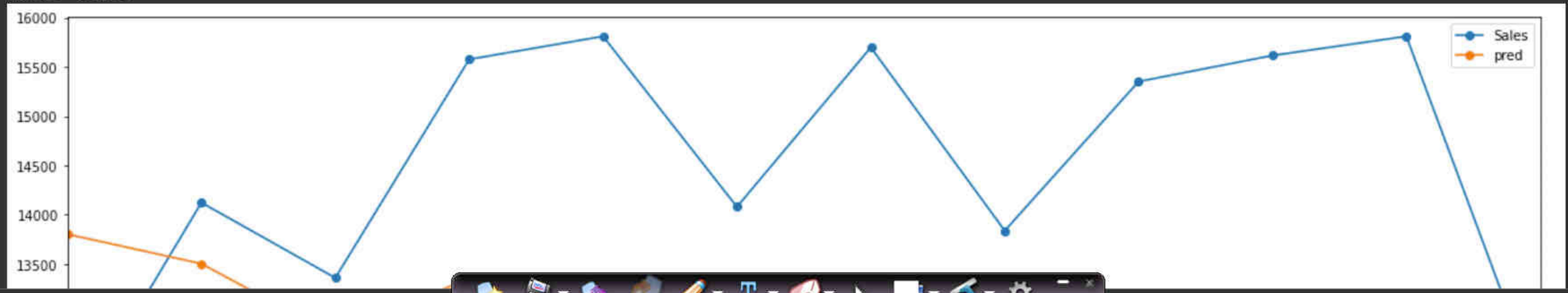
+ Code + Text Reconnect

Moving average forecast

```
df = train_x.copy()

df = df.append(pd.DataFrame(index = pd.date_range(start=df.index[-1], periods=12,freq='MS')[1:]))

pred = df.Sales.dropna().values

for i in range(12):
  pred = np.append(pred, pred[-3:].mean()) ##MA of window-length=3

test_x['pred'] = pred[-12:]
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

$$x_t = \frac{x_{t-1} + x_{t-2} + x_{t-3}}{3}$$

```
MAE : 1692.467
RMSE : 1875.03
MAPE: 0.115
```



33 / 33

Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:50

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

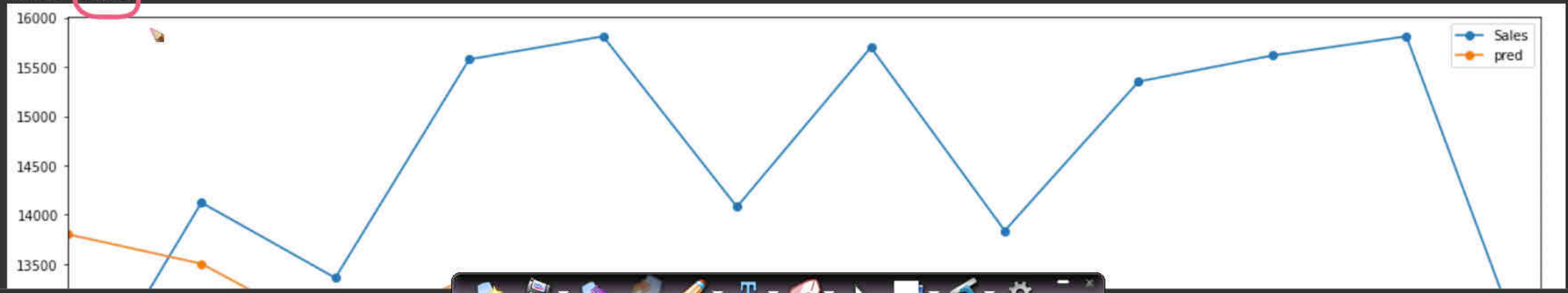
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=lGS92kCfcdiS

+ Code + Text Reconnect

Moving average forecast

```
df = train_x.copy()

df = df.append(pd.DataFrame(index = pd.date_range(start=df.index[-1], periods=12,freq='MS')[1:]))

pred = df.Sales.dropna().values

for i in range(12):
  pred = np.append(pred, pred[-3:].mean()) ##MA of window-length=3

test_x['pred'] = pred[-12:]
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 1692.467
RMSE : 1875.03
MAPE: 0.115
```



34 / 34

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

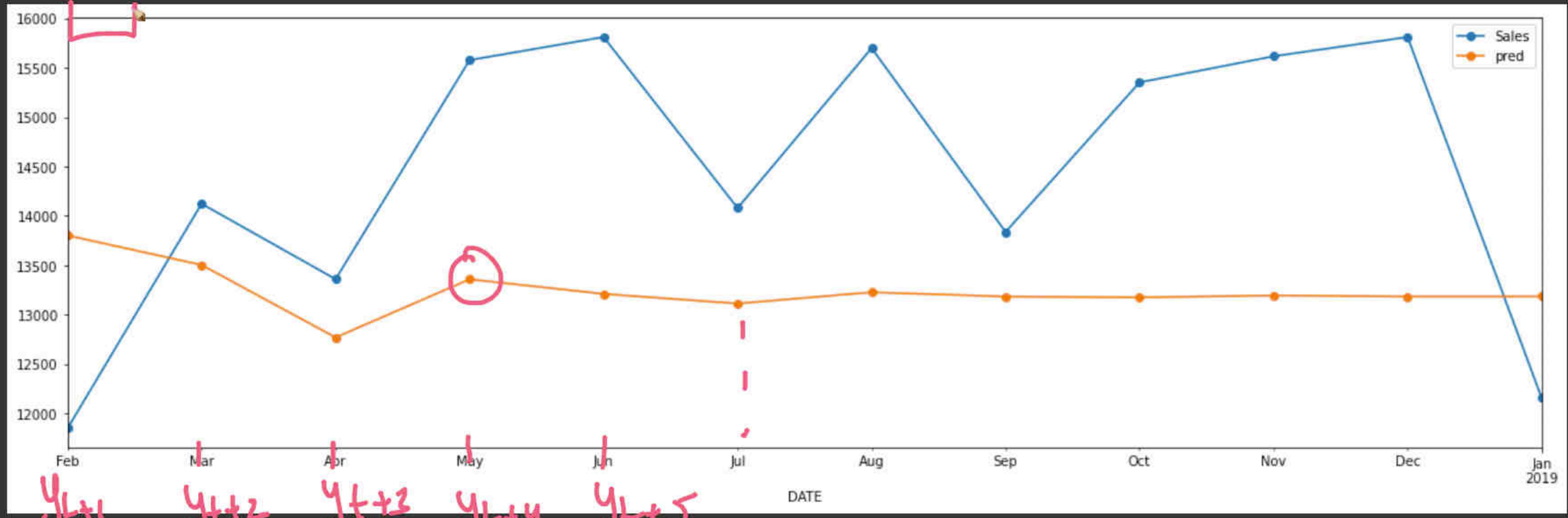
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=lGS92kCfcdiS

+ Code + Text

Reconnect

```
pred = np.append(pred, pred[-3:].mean()) ##MA of window-length=3

test_x['pred'] = pred[-12:]
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
MAE : 1692.467
RMSE : 1875.03
MAPE: 0.115
```



Simple exponential smoothing

35 / 35

Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:51

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=75hQpmq6VrV_

+ Code + Text Reconnect

Simple Drift Model

[ ] ↳ 2 cells hidden

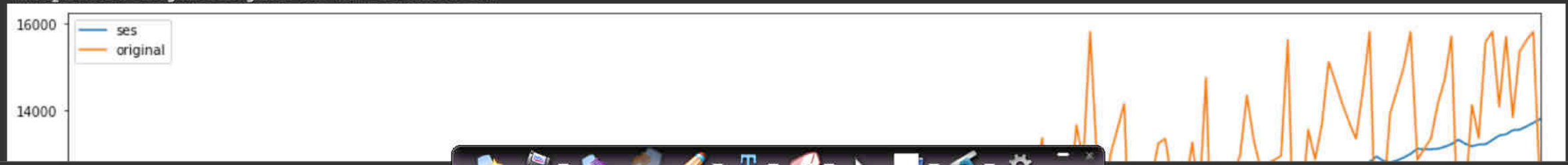
Moving average forecast

[ ] ↳ 1 cell hidden

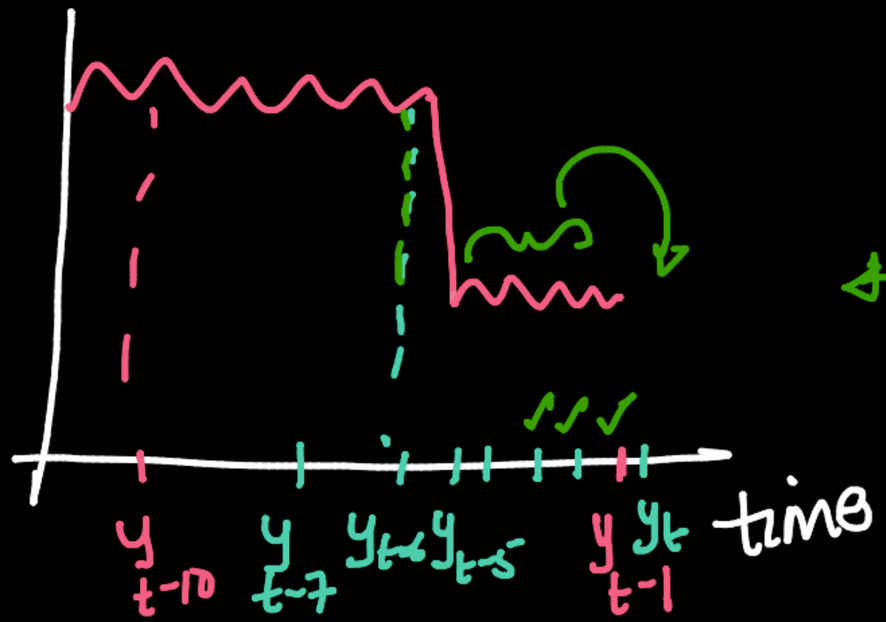
Simple exponential smoothing

```
import statsmodels.api as sm
model = pd.Series(sm.tsa.SimpleExpSmoothing(mobile_sales.Sales).fit(smoothing_level=1/(2*12)).fittedvalues)

model.plot(label='ses')
mobile_sales.Sales.plot(label='original')
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fc0bb4910>
```



36 / 36



[Simple Expo. Smoothing (SES)]

$X\ y_t = SMA(\text{window-10})$

$y_t = WMA$

↓

exponentially

$y_0, y_1 \cdots y_t$

$\hat{y}_{t+1}$

$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t$ ← pred, actual

recursive eqns ⟶ SES

$= \alpha \cdot y_t + (1-\alpha)\left[\alpha\, y_{t-1} + (1-\alpha)\hat{y}_{t-1}\right]$

$= \alpha \cdot y_t + \alpha(1-\alpha)\, y_{t-1} + (1-\alpha)^2 \hat{y}_{t-1}$

$0 < \alpha < 1$

$\left[\alpha y_{t-2} + (1-\alpha)\hat{y}_{t-2}\right]$

= Code =

$y_0, y_1 \cdots y_t \quad \hat{y}_{t+1}$

$y_{t+1} \quad \hat{y}_{t+2}$

y_{t+2}

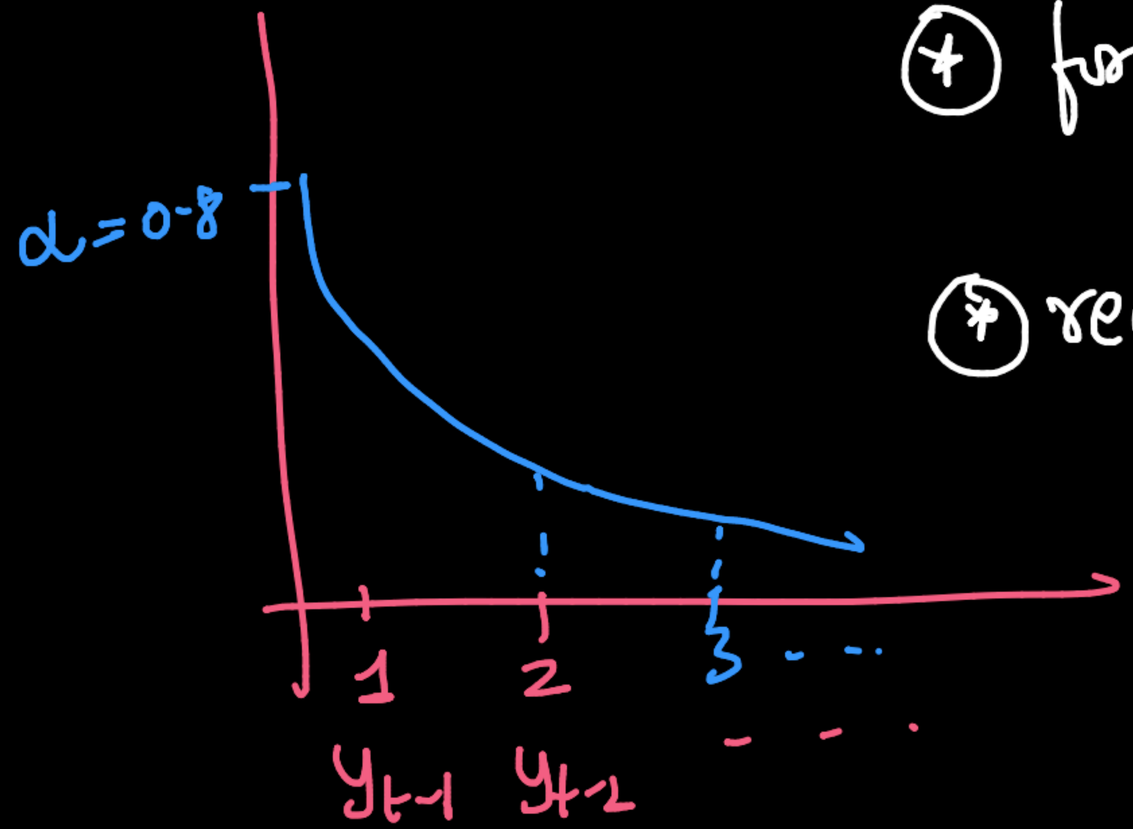
Code

$0 < \alpha < 1$

$\alpha = 0.8$ (let)

$$\hat{y}_{t+1} = \alpha y_t + \alpha(1-\alpha) y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots$$

$$\hat{y}_{t+1} = \underbrace{0.8\, y_t}_{\checkmark} + \underbrace{0.8\,(0.2)}_{0.16}\, y_{t-1} + \underbrace{0.8\,(0.2)^2}_{0.032}\, y_{t-2} + \underbrace{0.8\,(0.2)^3}_{0.0064}\, y_{t-3} + \cdots$$



① forget older values exponentially fast...

② recency-bias heavily...

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:10

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=N8veu82GVqzJ

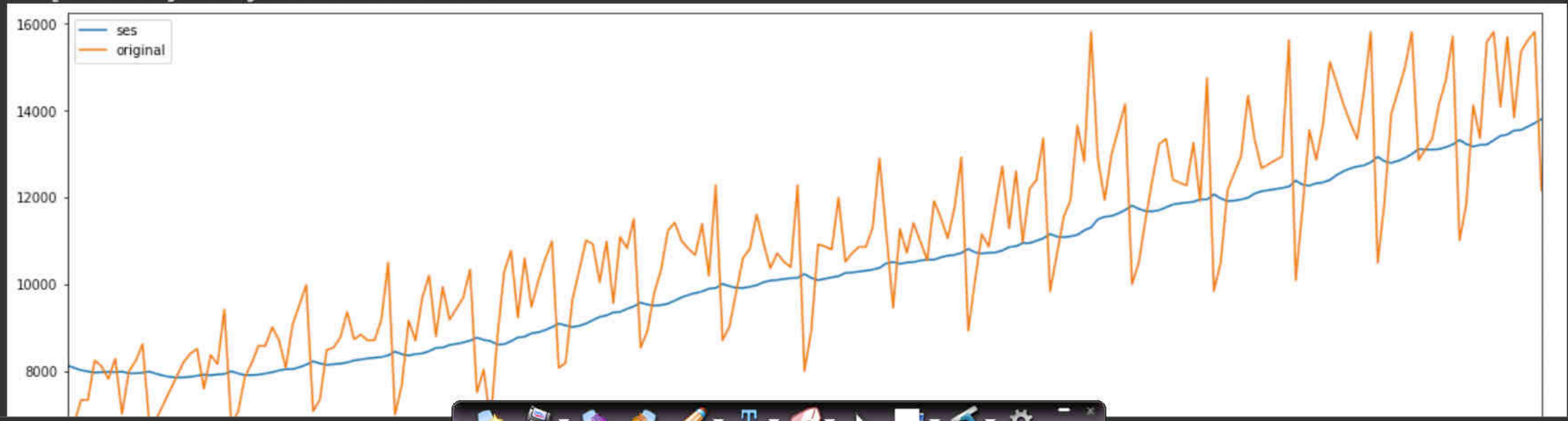
+ Code + Text Reconnect

Simple exponential smoothing

$\alpha_{suggested} = \frac{1}{2 \times Seasonality}$

```
import statsmodels.api as sm
model = pd.Series(sm.tsa.SimpleExpSmoothing(mobile_sales.Sales).fit(smoothing_level=1/(2*12)).fittedvalues)

model.plot(label='ses')
mobile_sales.Sales.plot(label='original')
plt.legend()
```

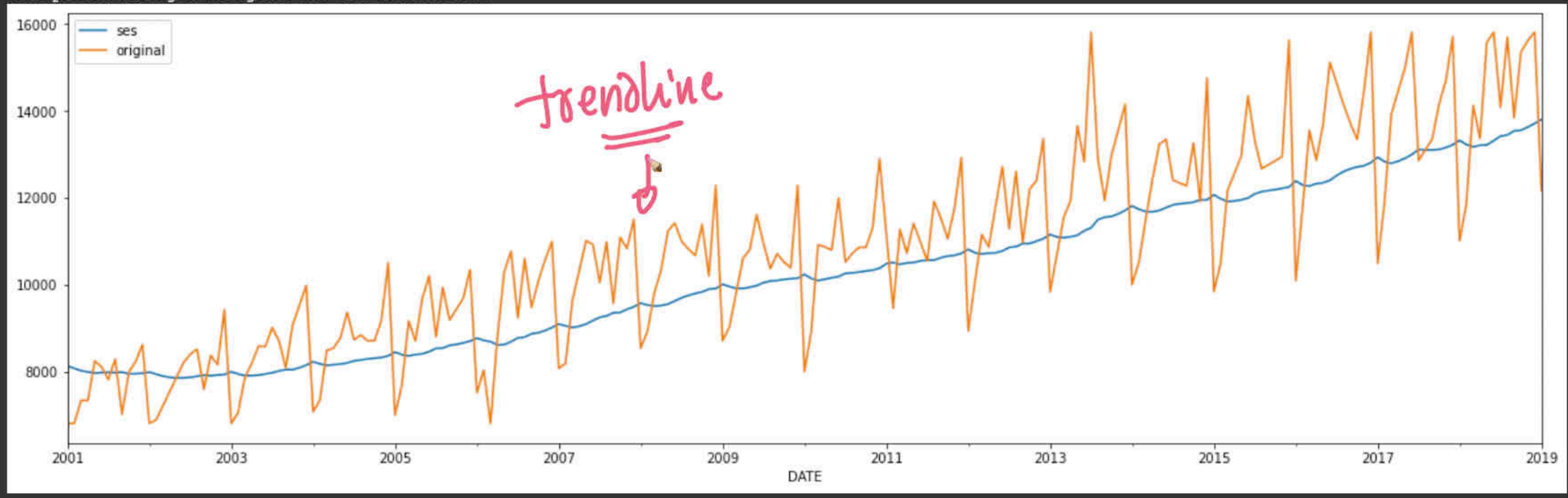
```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fc0bb4910>
```



42 / 43

```
model.plot(label='ses')
mobile_sales.Sales.plot(label='original')
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fc0bb4910>
```




```
[ ] model = sm.tsa.SimpleExpSmoothing(mobile_sales.Sales).fit(smoothing_level=1/(2*12))
    pred = model.forecast(steps = 12)
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:12

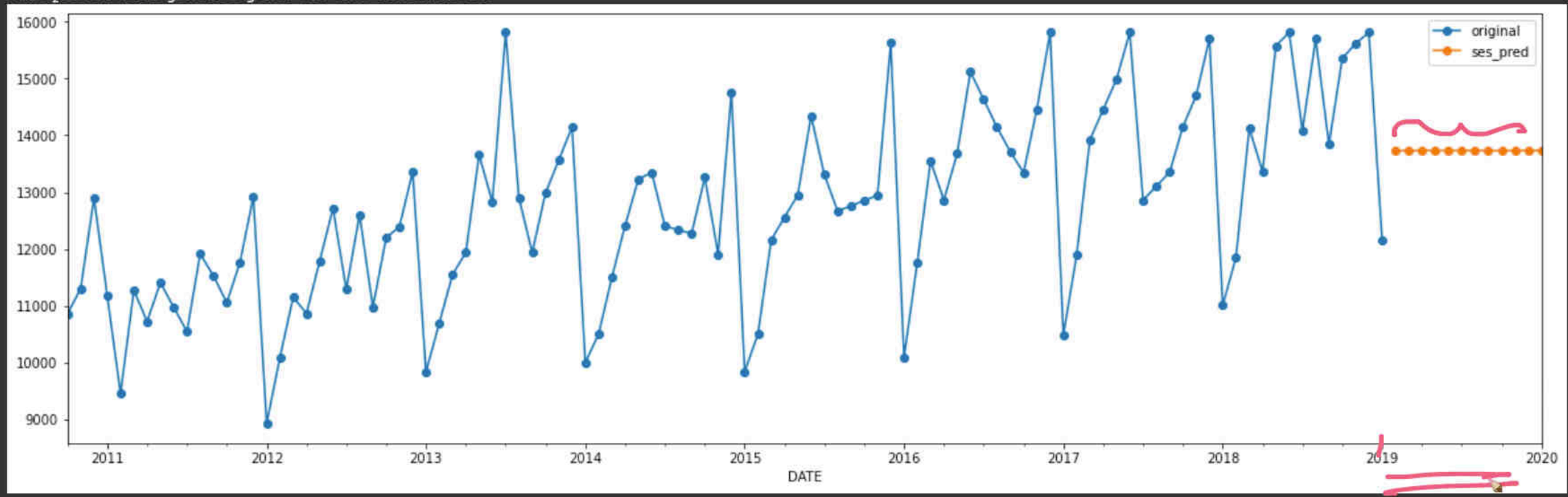
Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.se | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=N8veu82GVqzJ

+ Code + Text Reconnect

```
[ ] mobile_sales.Sales.tail(100).plot(label='original', style='-o')

    pred.plot(label='ses_pred', style='-o')
    plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fb87acad0>
```




```
[ ] model = sm.tsa.SimpleExpSmoothing(train_y_Sales).fit(smoothing_level=1/(2*12))
```

44 / 45

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

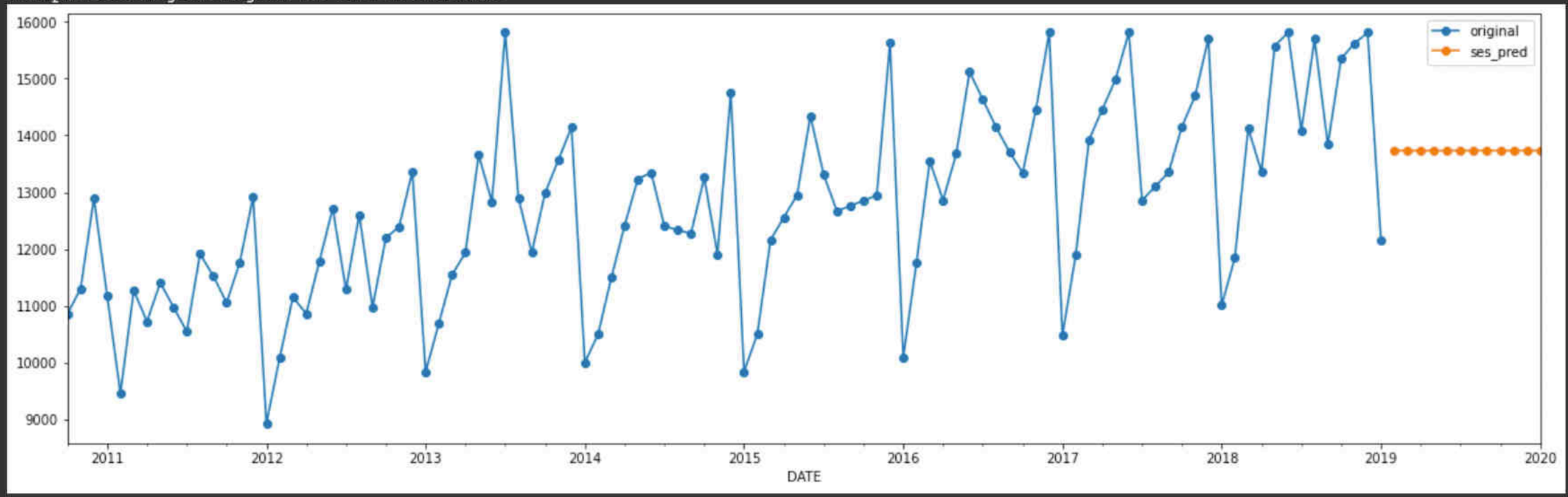
colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=N8veu82GVqzJ

+ Code + Text Reconnect

```
mobile_sales.Sales.tail(100).plot(label='original', style='-o')
[ ]
pred.plot(label='ses_pred', style='-o')
plt.legend()
```

$\hat{y}_{t+1}$ y_{t+1} y_{t+2}

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fb87acad0>
```




```
[ ] model = sm.tsa.SimpleExpSmoothing(train_x.Sales).fit(smoothing_level=1/(2*12))
    test_x['pred'] = model.forecast(steps =
```

45 / 46

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:13

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=N8veu82GVqzJ

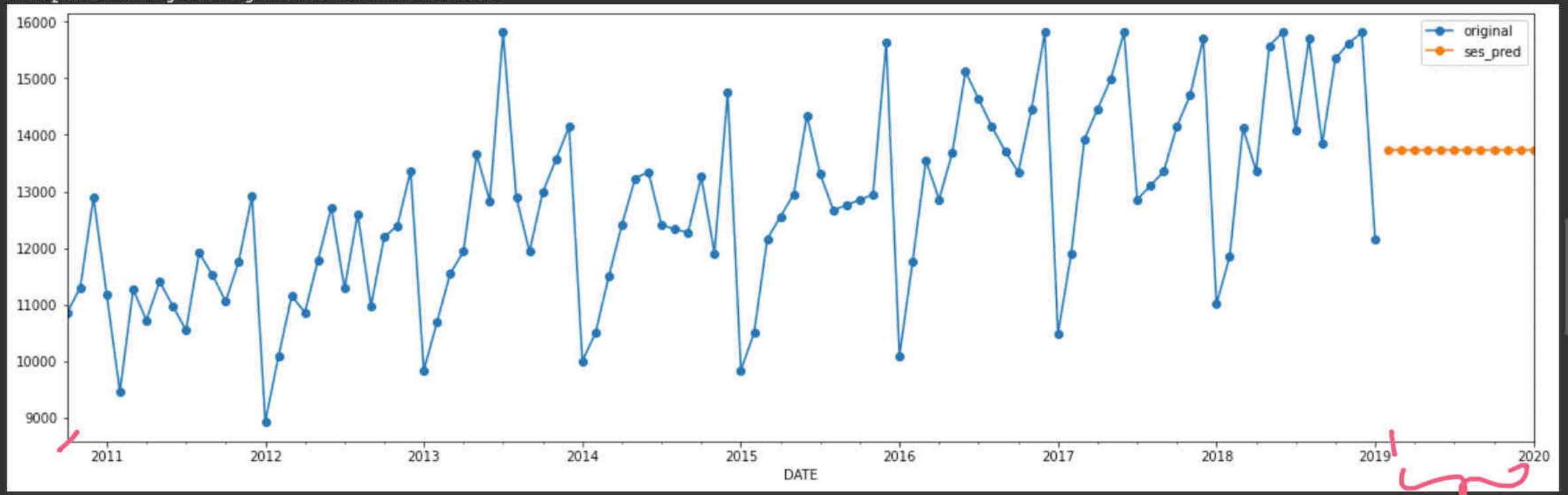
+ Code + Text Reconnect

```
[ ]
    mobile_sales.Sales.tail(100).plot(label='original', style='-o')
    pred.plot(label='ses_pred', style='-o')
    plt.legend()
```

y_{t+1} 88

y_{t+1} $\hat{y}_{t+2}$ ----

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fb87acad0>
```



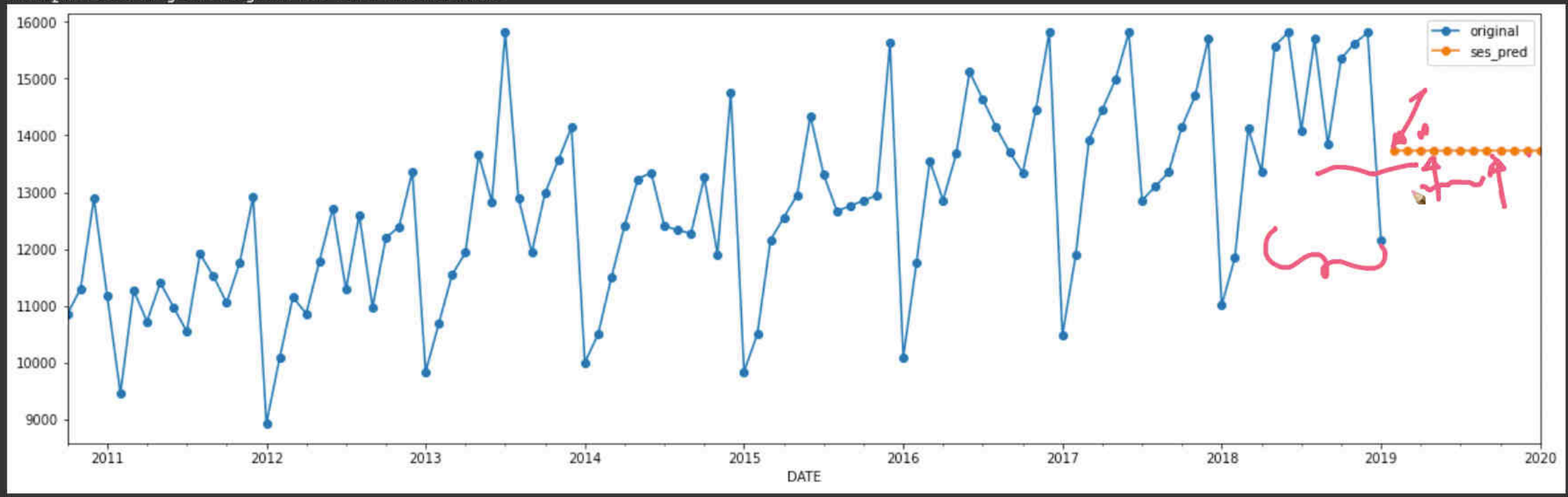
12M

```
[ ] model = sm.tsa.SimpleExpSmoothing(train_x.Sales).fit(smoothing_level=1/(2*12))
    test_x['pred'] = model.forecast(steps =
```

46 / 47

```
[ ]
    pred.plot(label='ses_pred', style='-o')
    plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fb87acad0>
```



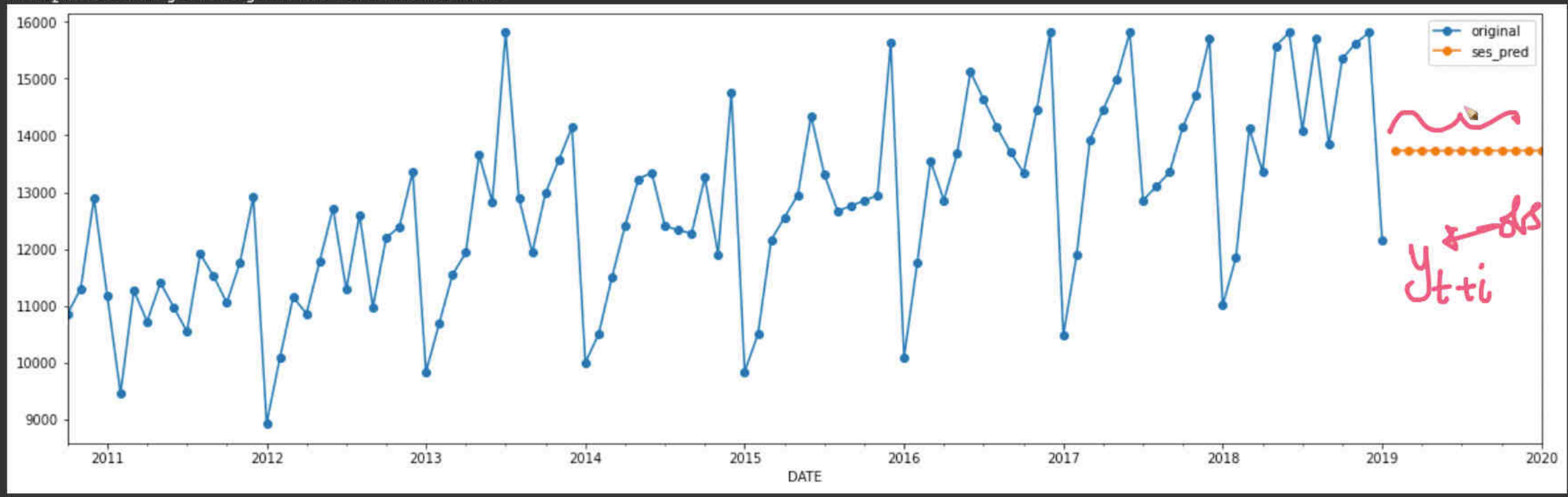

```
[ ] model = sm.tsa.SimpleExpSmoothing(train_x.Sales).fit(smoothing_level=1/(2*12))
    test_x['pred'] = model.forecast(steps =
```

```
mobile_sales.Sales.tail(100).plot(label='original', style='-o')
```

```
pred.plot(label='ses_pred', style='-o')
plt.legend()
```

SES

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fb87acad0>
```




```
model = sm.tsa.SimpleExpSmoothing(train_x.Sales).fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps =
```

Two-ways

SES can detect trend

$\hat{y}_{t+1}$ y_{t+1}

$\hat{y}_{t+2}$ y_{t+2}

$\hat{y}_{t+3}$ y_{t+3}

Simpler

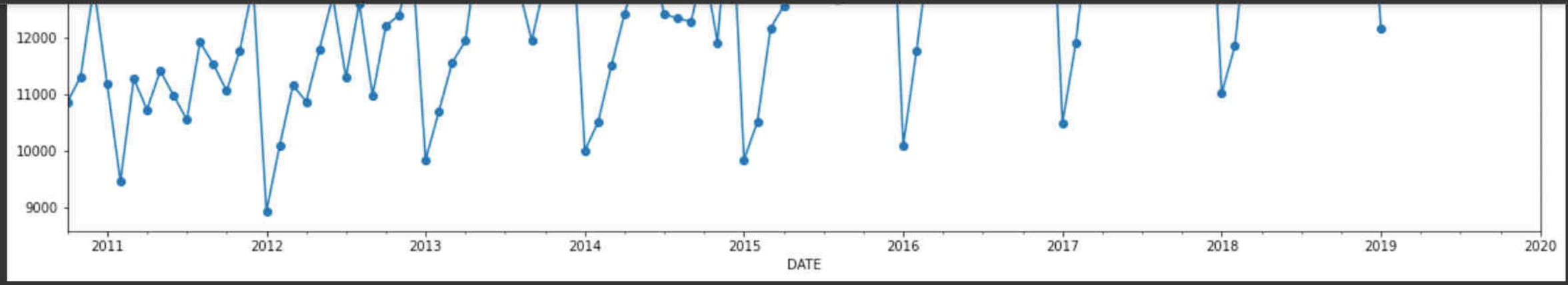
=0=

our problem

SES may not detect trend

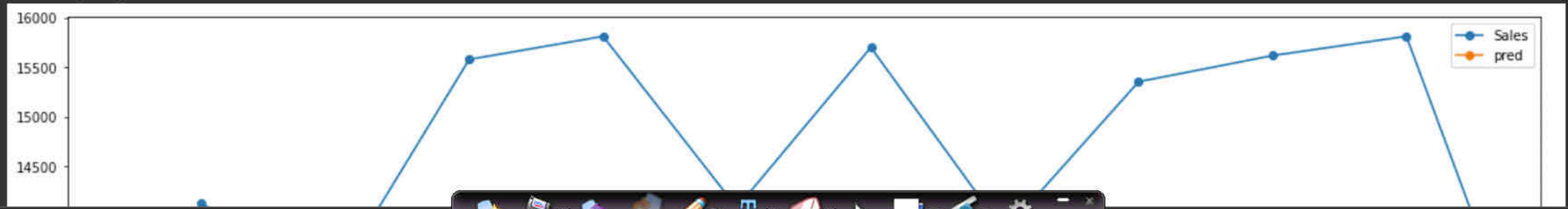
$y_0 \dots y_t$

$\hat{y}_{t+1} \dots \hat{y}_{t+10}$ ← harder



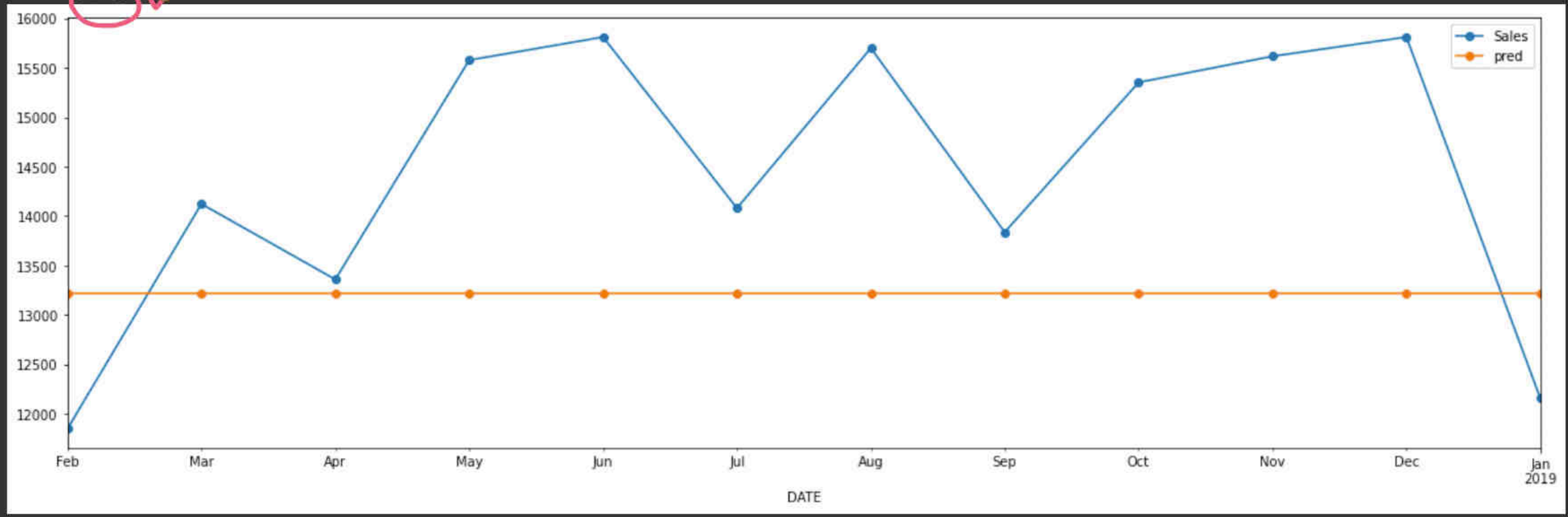

```
model = sm.tsa.SimpleExpSmoothing(train_x.Sales).fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps = 12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
MAE : 1621.347
RMSE : 1830.599
MAPE: 0.109
```



```
model = sm.tsa.SimpleExpSmoothing(train_x.Sales).fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps = 12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
MAE : 1621.347
RMSE : 1830.599
MAPE: 0.109
```



Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:18

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=HThI0P6GUqUG

+ Code + Text Reconnect

Simple Drift Model

[ ] ↳ 2 cells hidden

Moving average forecast

↳ 1 cell hidden

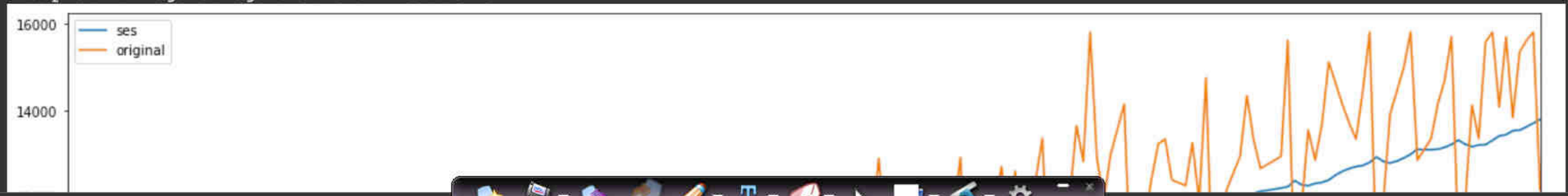
Missing trend & Seasonality

~ 10.5% MAPE

Simple exponential smoothing

```
import statsmodels.api as sm
model = pd.Series(sm.tsa.SimpleExpSmoothing(mobile_sales.Sales).fit(smoothing_level=1/(2*12)).fittedvalues)

model.plot(label='ses')
mobile_sales.Sales.plot(label='original')
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.legend.Legend at 0x7f2fc0bb4910>
```

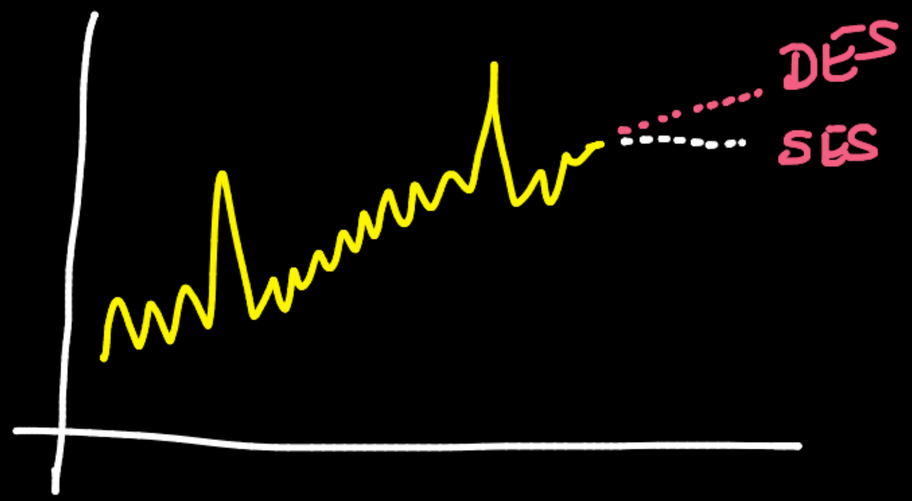


52 / 52

Double Expo Smoothing (DES)

(Intuition)

DES = SES + Trend



[Holt's linear model]

Single Expo Smoothing

SES math:

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t$$

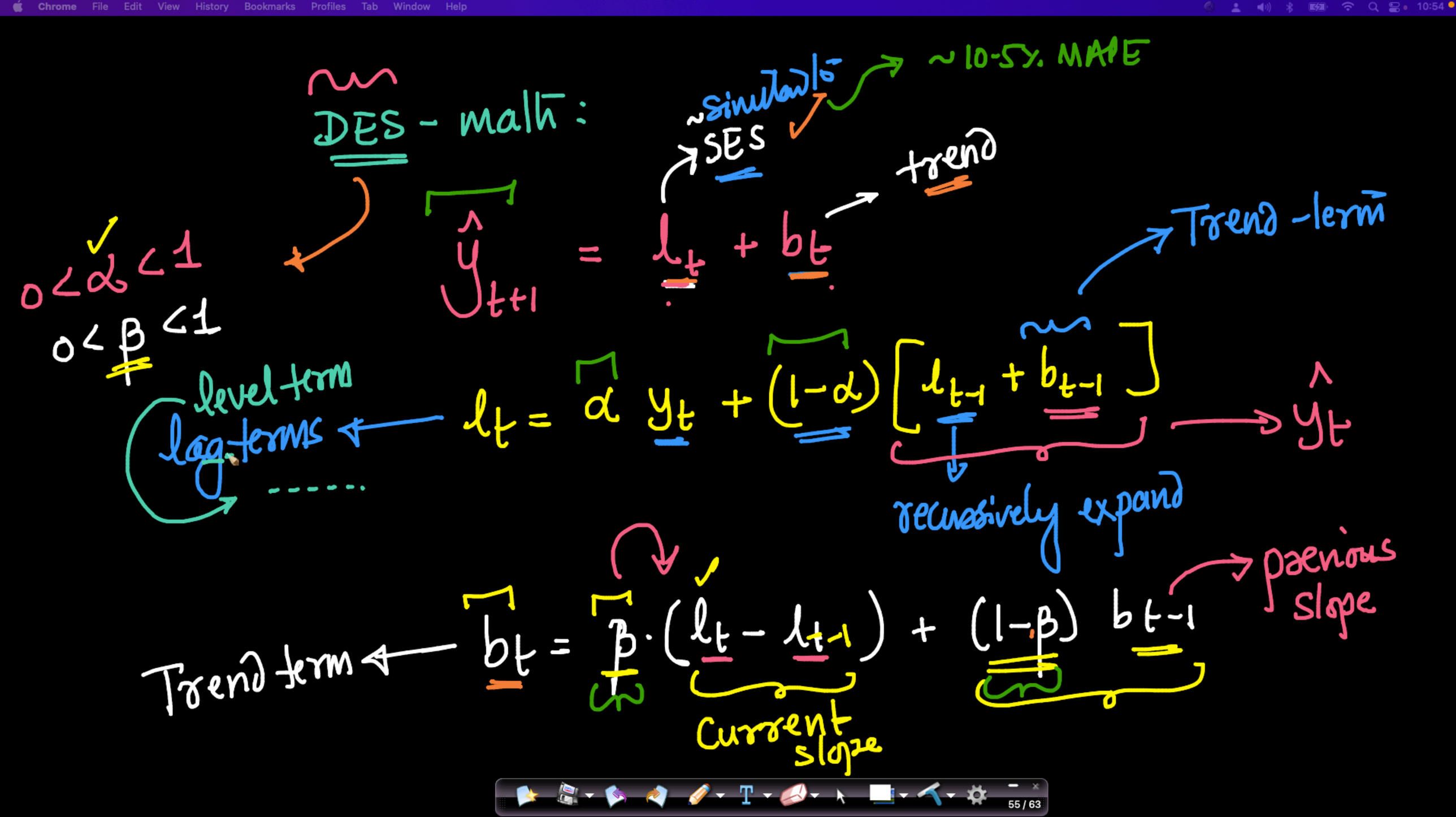
$\cong$ expo. weighting

$$0 < \alpha < 1$$

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\alpha \cdot y_{t-1}$$

$$+ \alpha (1-\alpha)^2 y_{t-2} + \cdots$$

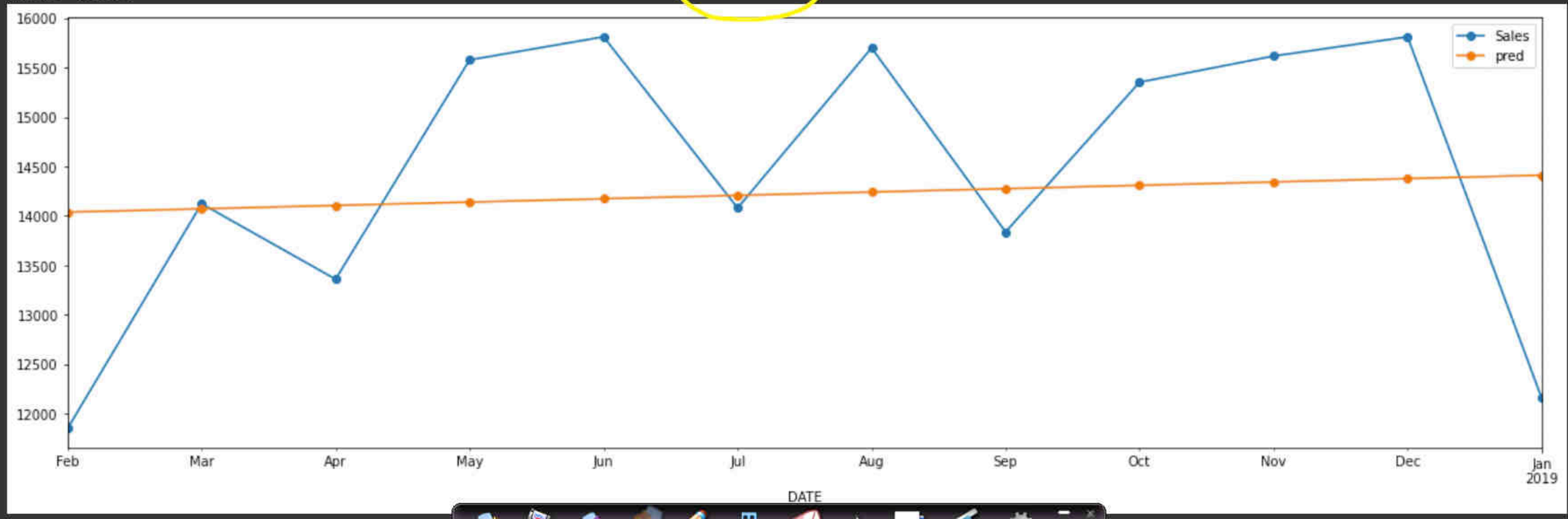
was missing trend & Seasonality


DES - math :
~ simulate SES
~ 10-5% MAPE
trend
$\hat{y}_{t+1} = l_t + b_t$
$0 < \alpha < 1$
$0 < \beta < 1$
Trend - term
level term
lag terms
$l_t = \alpha\, y_t + (1-\alpha)\left[l_{t-1} + b_{t-1}\right] \rightarrow \hat{y}_t$
recursively expand
Trend term
$b_t = \beta \cdot (l_t - l_{t-1}) + (1-\beta)\, b_{t-1}$
previous slope
Current slope

```
# trend="add" or "multiplicative"
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add').fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps = 12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
MAE : 1172.756
RMSE : 1360.928
MAPE: 0.083
```

DES

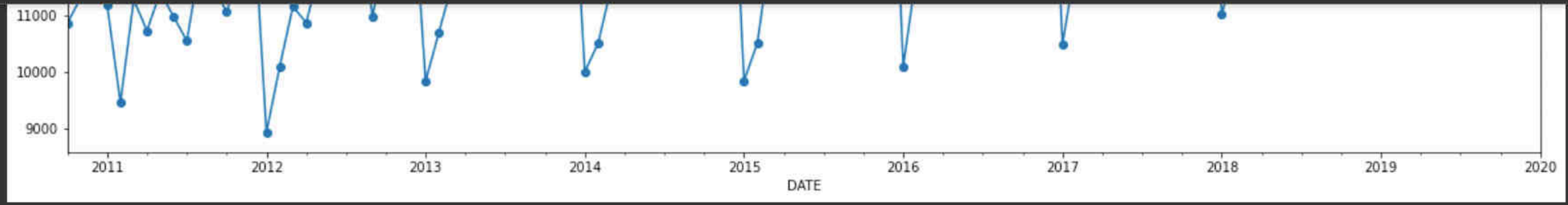


56 / 56

SES: ~10.5% MAPE

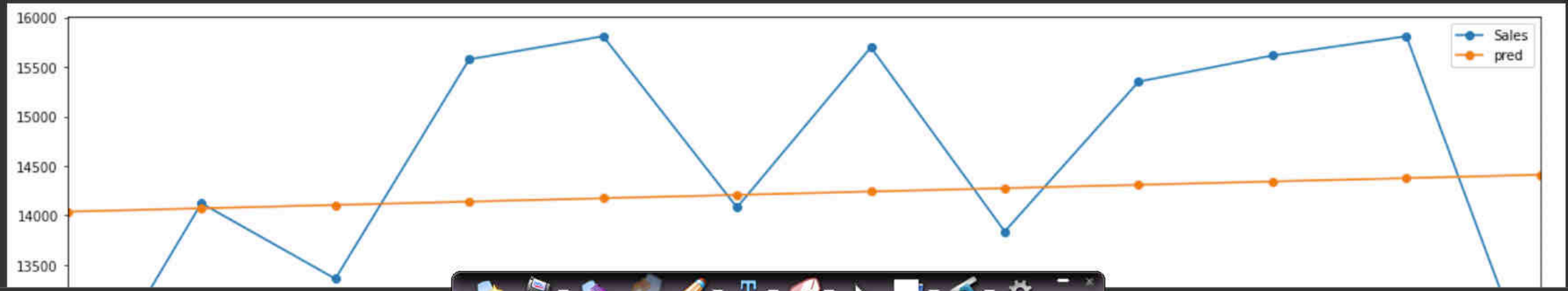
DES: ~8.3% MAPE
(Trend)

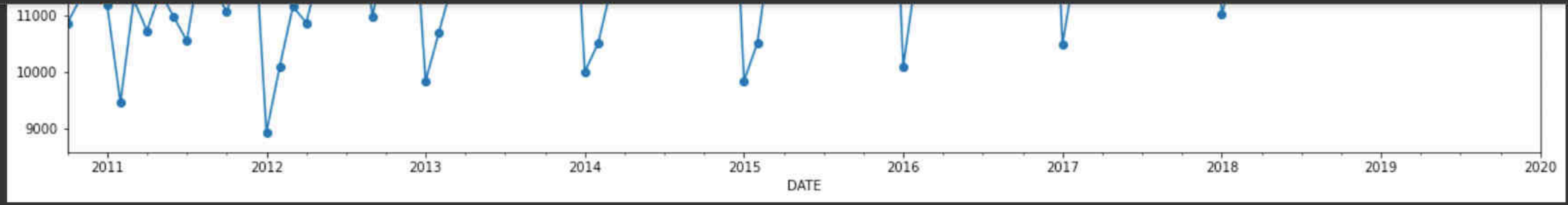
TES
(Trend & Seasonality)




```
# trend="add" or "multiplicative"
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add').fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps = 12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

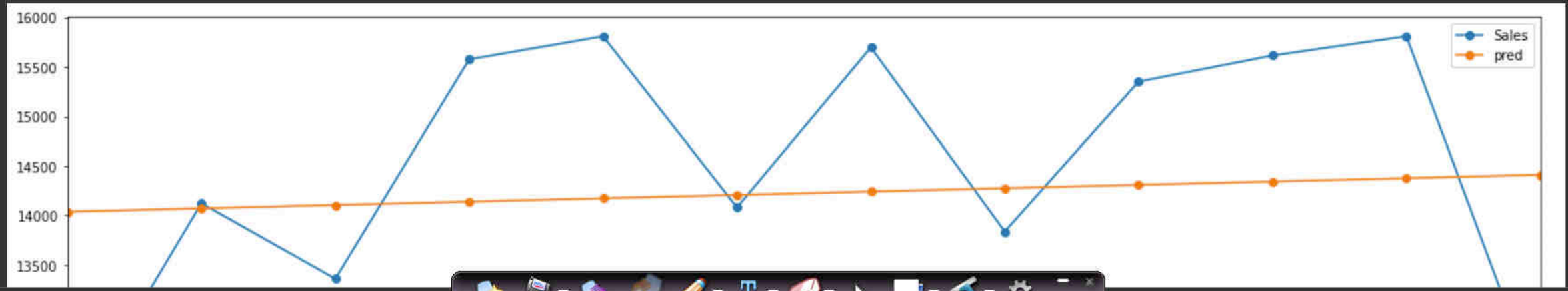
```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
MAE : 1172.756
RMSE : 1360.928
MAPE: 0.083
```

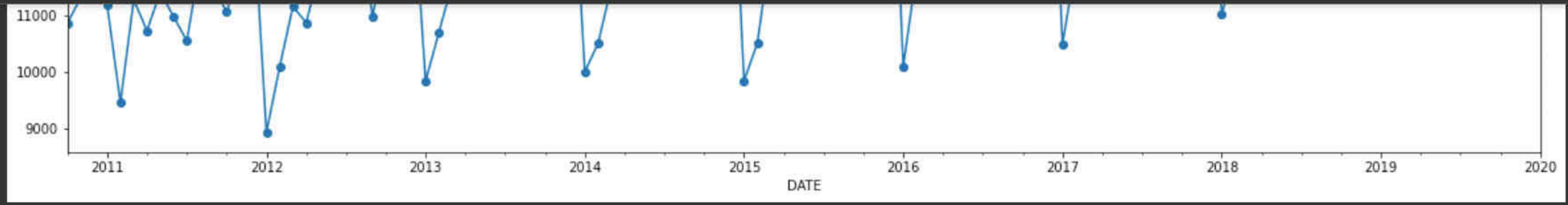





```
# trend="add" or "multiplicative"
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add').fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps = 12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
MAE : 1172.756
RMSE : 1360.928
MAPE: 0.083
```

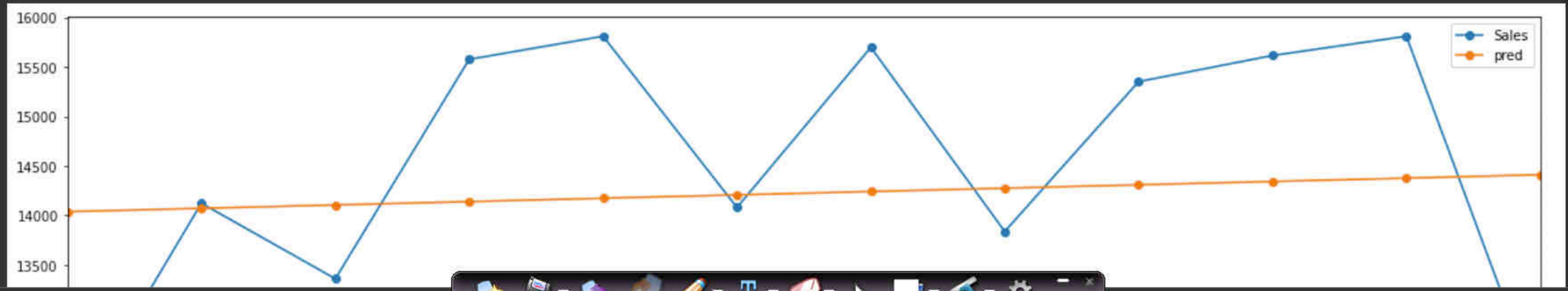




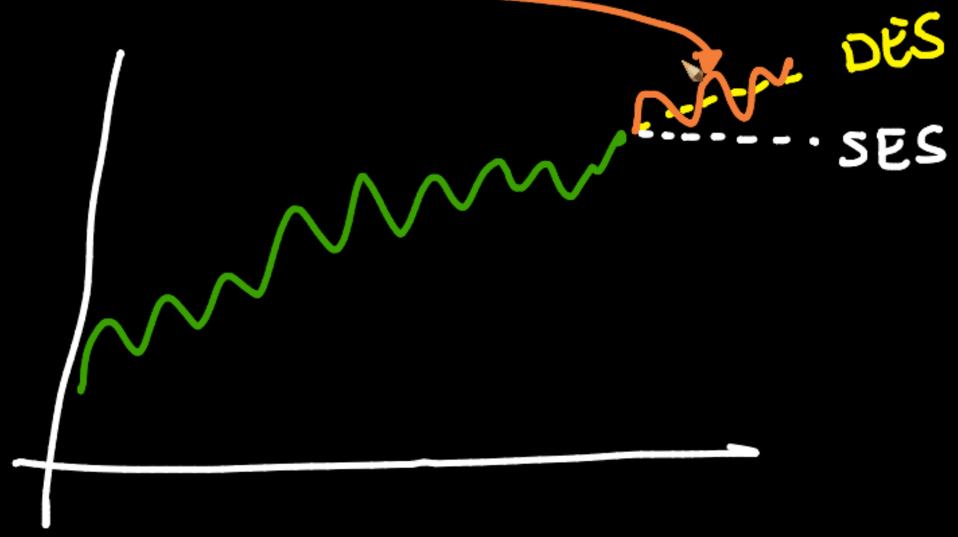

```
# trend="add" or "multiplicative"
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add').fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps = 12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

α, β: hyper params

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
MAE : 1172.756
RMSE : 1360.928
MAPE: 0.083
```



TES = DES + Seasonality



TES

(Holt-winters model)

↓

lag-terms (SES)

+ trend (DES)

+ Seasonality

TES:

$(0 < \alpha, \beta, \gamma < 1)$

$$\hat{y}_{t+1} = \underbrace{l_t}_{\text{lag terms}} + \underbrace{b_t}_{\text{trend}} + S_{t+1-m}$$

{ (data) → $m = 12$ months

$$l_t = \alpha \left[y_t - S_{t-m} \right] + (1-\alpha) \left[l_{t-1} + b_{t-1} \right]$$

$\hat{y}_t - S_{t-m}$

Subtract the seasonality effect

$$b_t = \beta (l_t - l_{t-1}) + (1-\beta)\, b_{t-1}$$ → Same as earlier

$$s_t = \gamma \left[y_t - l_{t-1} - b_{t-1} \right] + (1-\gamma) \left[s_{t-m} \right]$$

$y_t - l_{t-1} - b_{t-1}$ → Current Seasonality after removing level term & trend term

s_{t-m} → past Seasonality

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:56

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=sxw1fndME6hE

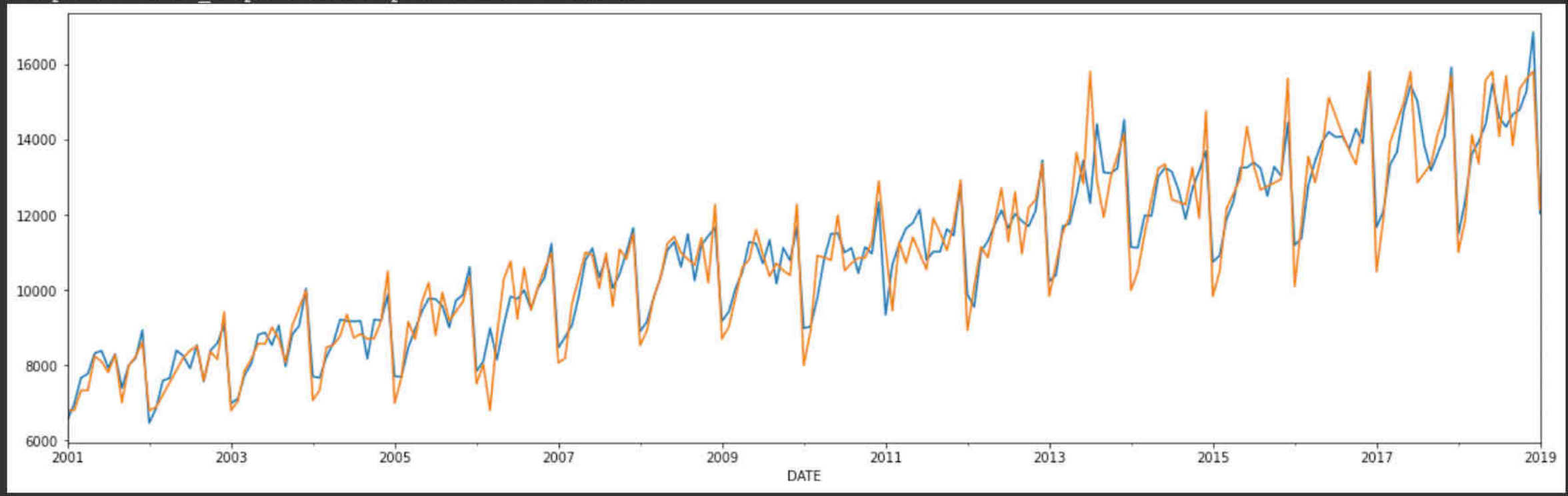
+ Code + Text Reconnect

```
## seasonality="add" or "multiplicative"
model = sm.tsa.ExponentialSmoothing(mobile_sales.Sales, trend='add', seasonal='add').fit(smoothing_level=0.4)
model.fittedvalues.plot()
mobile_sales.Sales.plot()
```

α

1 / 2×12=M

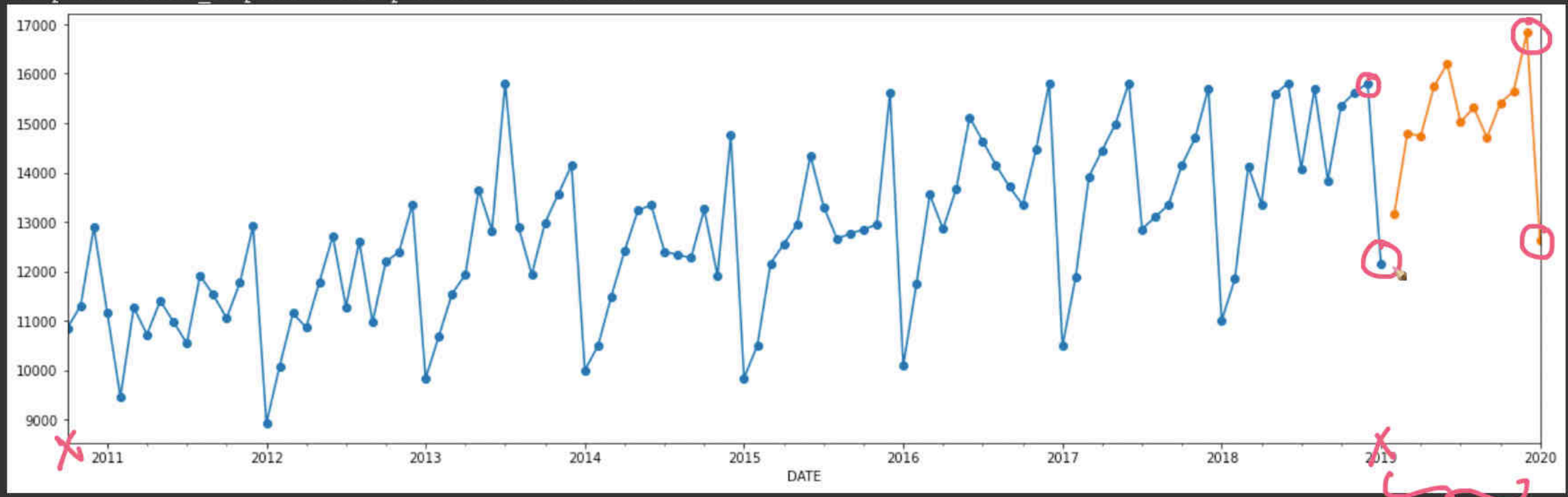
```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f2fb85e0d90>
```



64 / 64

```
[ ] pred = model.forecast(steps = 12)
    mobile_sales.Sales.tail(100).plot(style='-o')
    pred.plot(style='-o')
```

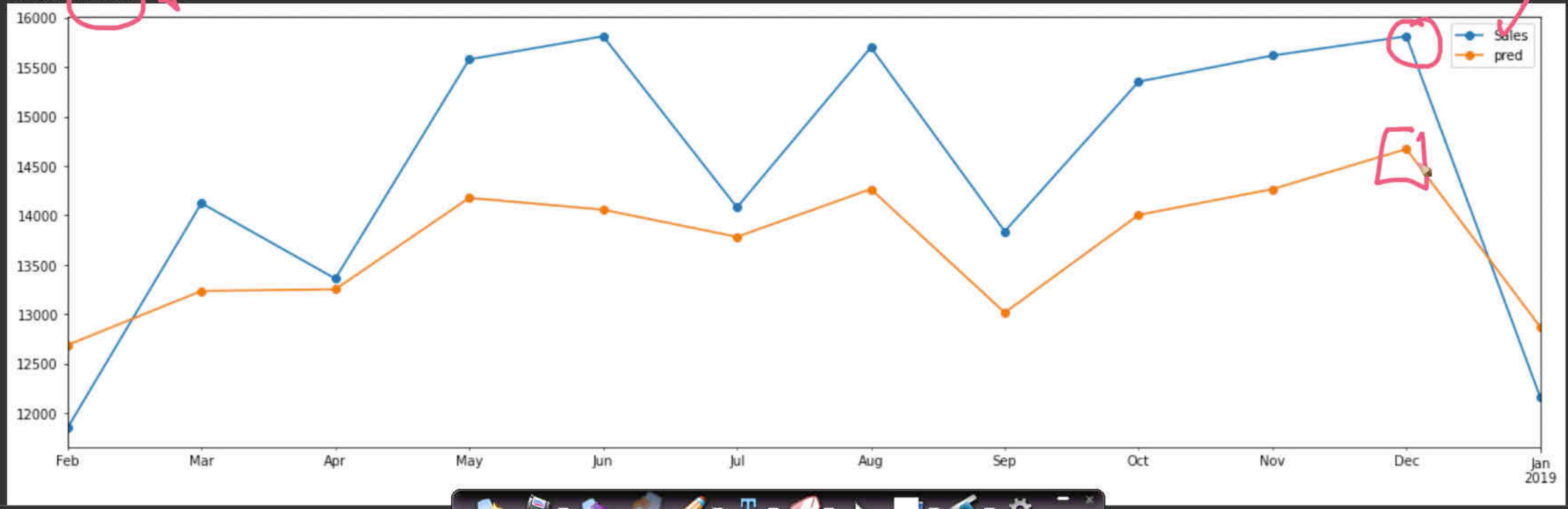
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2fb876dfd0>
```




```
[ ] model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add', seasonal='add').fit(smoothing_level=1/(2*12))
    test_x['pred'] = model.forecast(steps=12)
    test_x.plot(style='-o')
```

```
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
/usr/local/lib/python3.7/dist-packages/scipy/optimize/_numdiff.py:579: RuntimeWarning: overflow encountered in true_divide
  J_transposed[i] = df / dx
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/holtwinters.py:712: ConvergenceWarning: Optimization failed to converge. Check mle_
  ConvergenceWarning)
MAE : 1006.813
RMSE : 1109.849
MAPE: 0.068
```



Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:59

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=cZmZEBsAFX0w

+ Code + Text Reconnect

[ ] ↳ 3 cells hidden

Exp: *Mixture* of Additive and Multiplicative models

level terms

```
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add', seasonal='mul').fit(smoothing_level=1/(2*12))
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/holtwinters.py:712: ConvergenceWarning: Optimization failed to converge. Check mle_
  ConvergenceWarning)
MAE : 1006.813
RMSE : 1109.849
MAPE: 0.068
```

67 / 67

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:59

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.sa | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=cZmZEBsAFX0w

+ Code + Text Reconnect

[ ] ↳ 3 cells hidden

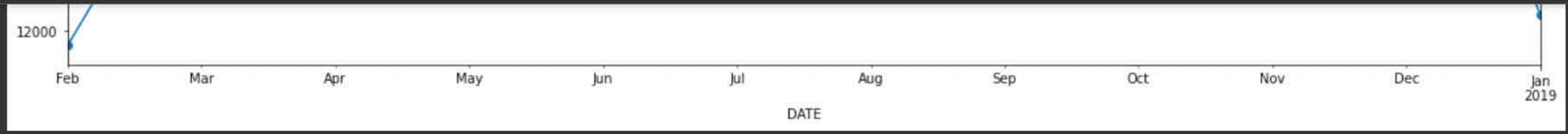
$$\hat{y}_{t+1} = (l_t + b_t)\, s_t$$

Exp: *Mixture* of Additive and Multiplicative models

```
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='add', seasonal='mul').fit(smoothing_level=1/(2*12))
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/holtwinters.py:712: ConvergenceWarning: Optimization failed to converge. Check mle_
  ConvergenceWarning)
MAE : 1006.813
RMSE : 1109.849
MAPE: 0.068
```

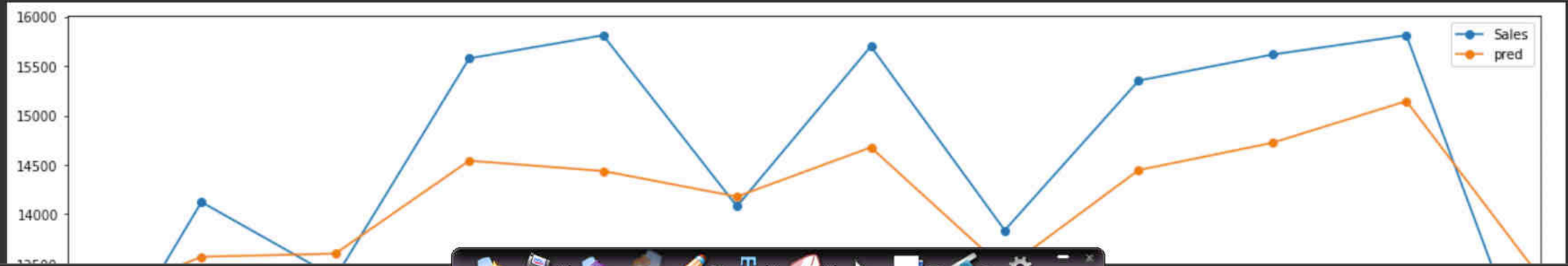
68 / 68




```
model = sm.tsa.ExponentialSmoothing(train_x.Sales, trend='mul', seasonal='add').fit(smoothing_level=1/(2*12))
test_x['pred'] = model.forecast(steps=12)
test_x.plot(style='-o')
performance(test_x['Sales'], test_x['pred'])
```

$$\hat{y}_{t+1} = (l_t + s_{t+1-m}) \cdot b_t$$

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/base/tsa_model.py:165: ValueWarning: No frequency information was provided, so infe
  % freq, ValueWarning)
/usr/local/lib/python3.7/dist-packages/scipy/optimize/_numdiff.py:579: RuntimeWarning: overflow encountered in true_divide
  J_transposed[i] = df / dx
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/holtwinters.py:712: ConvergenceWarning: Optimization failed to converge. Check mle_
  ConvergenceWarning)
MAE : 794.587
RMSE : 883.252
MAPE: 0.055
```



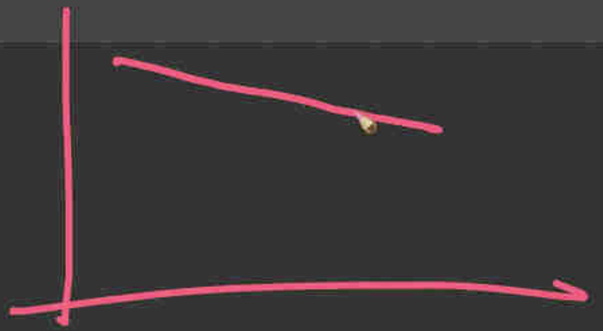
QUIZ: What is the best model to predict monthly total population of a country? [Pre-covid]

1. SES
2. DES
3. TES
4. Moving Avergae

level ✓
Trend ✓
Seasonality

ARIMA & SARIMA Models

[ ] ↳ 11 cells hidden



Recap:

- TS: Data Analysis → ACF / PACF
  - → outlier
  - → Imputation
  - ⋮
- TSF:
  - → Naive models
  - → SES, DES, TES

AR , MA, ARIMA, SARIMA

modified version of

Gr. 2eg

Math now

Code next

Math + code

ARIMA (models)

SARIMA

SARIMAX

exogenous Variables

Notation:

$y_0, y_1, y_2 \dots y_t, y_{t+1} \dots$ ← observations

$\dots \hat{y}_{t-1}, \hat{y}_t, \hat{y}_{t+1} \dots$ ← Predictions

$\epsilon_{t-1}, \epsilon_t, \dots$ ← Error terms

$\epsilon_t = y_t - \hat{y}_t$

μ ← mean value of all y_i's

AR(p) : Auto regressive model

Ideally

$$y_t = \mu + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots - \alpha_p y_{t-p} + \epsilon_t$$

intercept → μ

weights ← $\alpha_1, \alpha_2, \ldots, \alpha_p$

$$\hat{y}_t = \mu + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots - \alpha_p y_{t-p}$$

Params: $[\alpha_1, \cdots \alpha_p, \mu]$

hyper-param: p

SES

weighted-avg

↳ exponential

α : hyper-param

vs

AR(p)

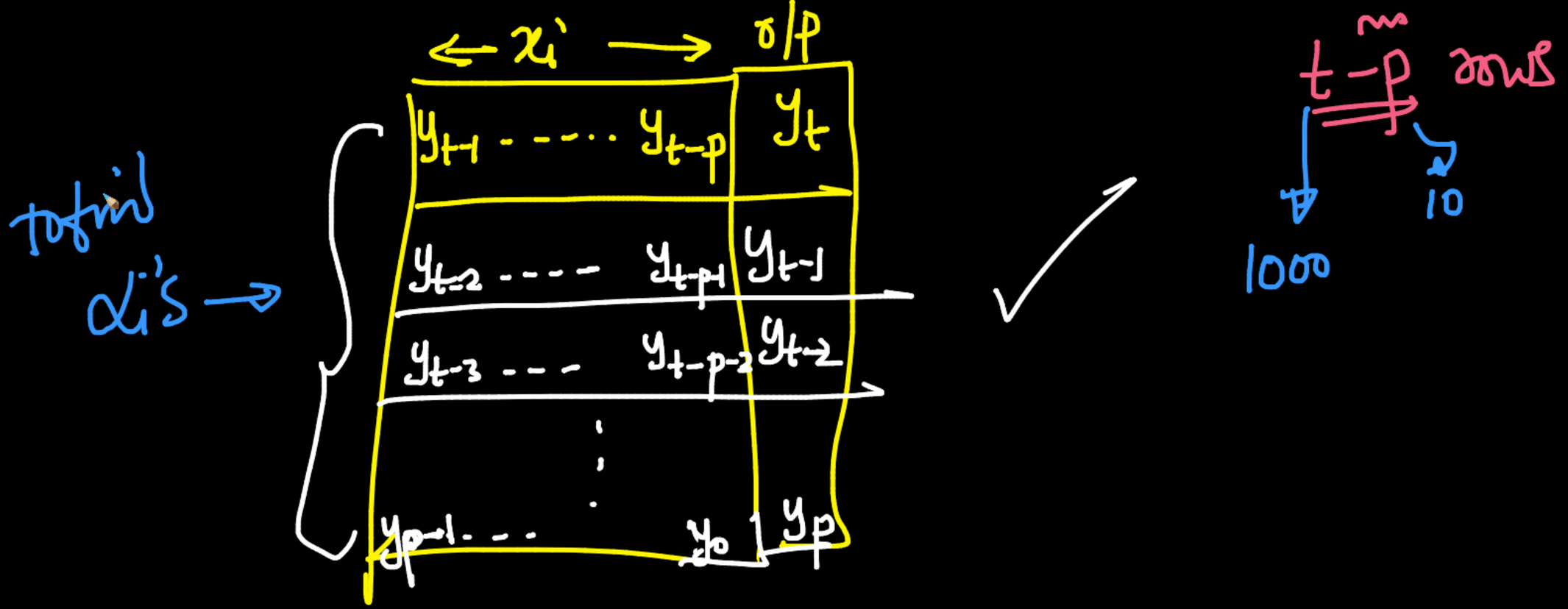
weighted-avg

↳ learnt → α_i's

p : hyperparam

D_{train} for Lr-reg

$y_0 \dots y_t$



avg of all y_i's in train-TS

$$y_t = \epsilon_t + \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots \theta_q \epsilon_{t-q}$$

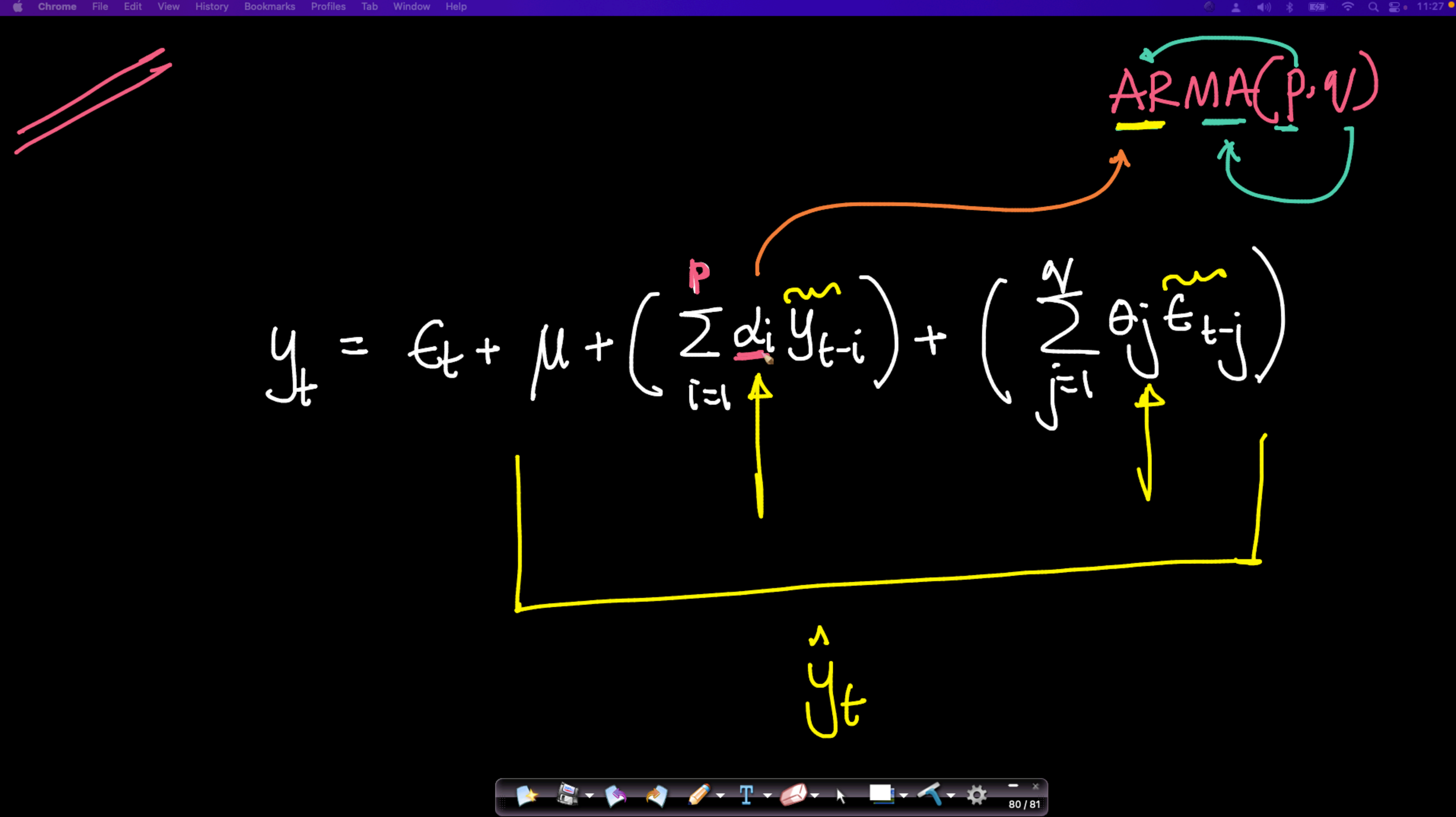
MA(q)

moving average

Linear regr model on last q-errors

$y_{t-1} - \hat{y}_{t-1}$

$y_{t-2} - \hat{y}_{t-2}$


ARMA(p,q)
$y_t = \epsilon_t + \mu + \left(\sum_{i=1}^{p} \alpha_i \tilde{y}_{t-i} \right) + \left(\sum_{j=1}^{q} \theta_j \tilde{\epsilon}_{t-j} \right)$
$\hat{y}_t$
80 / 81

Big Picture

AR(P)

MA(Q)

ARMA(P,Q)

ARIMA(P,Q,d)

SARIMA(P,Q,d,s,P,Q,D)

Seasonality

SARIMAX

exogenous/external variables

differencing → removed trend

Integration ← get back trend

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Time-Series.ipynb - Colaborat | Exponential smoothing - Wiki | statsmodels.tsa.statespace.s | Forecasting at Uber: An Introd

colab.research.google.com/drive/1vLADuayVVAWtY-fo3XXbi608UUnVQPdB#scrollTo=YI_UqzgEL-q4

Time-Series.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text Reconnect Editing

ARIMA & SARIMA Models

AR(p)

↳ 2 cells hidden

MA(q) models

[ ] ↳ 1 cell hidden

ARIMA(p,d,q)

[ ] ↳ 2 cells hidden

SARIMA (p,d,q,P,D,Q,s)

↳ 1 cell hidden

82 / 82

✓ 8s completed at 18:41

Time_Series (1).ipynb

Show all