Introduction to Time Series

Nikoleta Glynatsi

Mathematics & CUBRIC workshop

Software
Sustainability
Institute

Superstore Sales

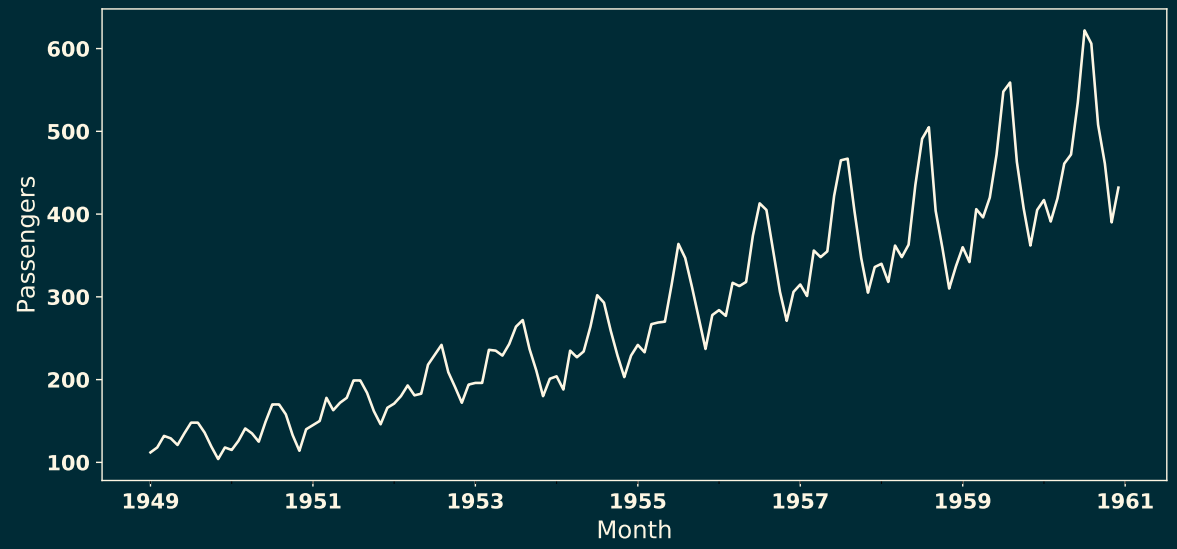
Noise or randomness in the data points

Predictable fluctuations correlated with the calendar

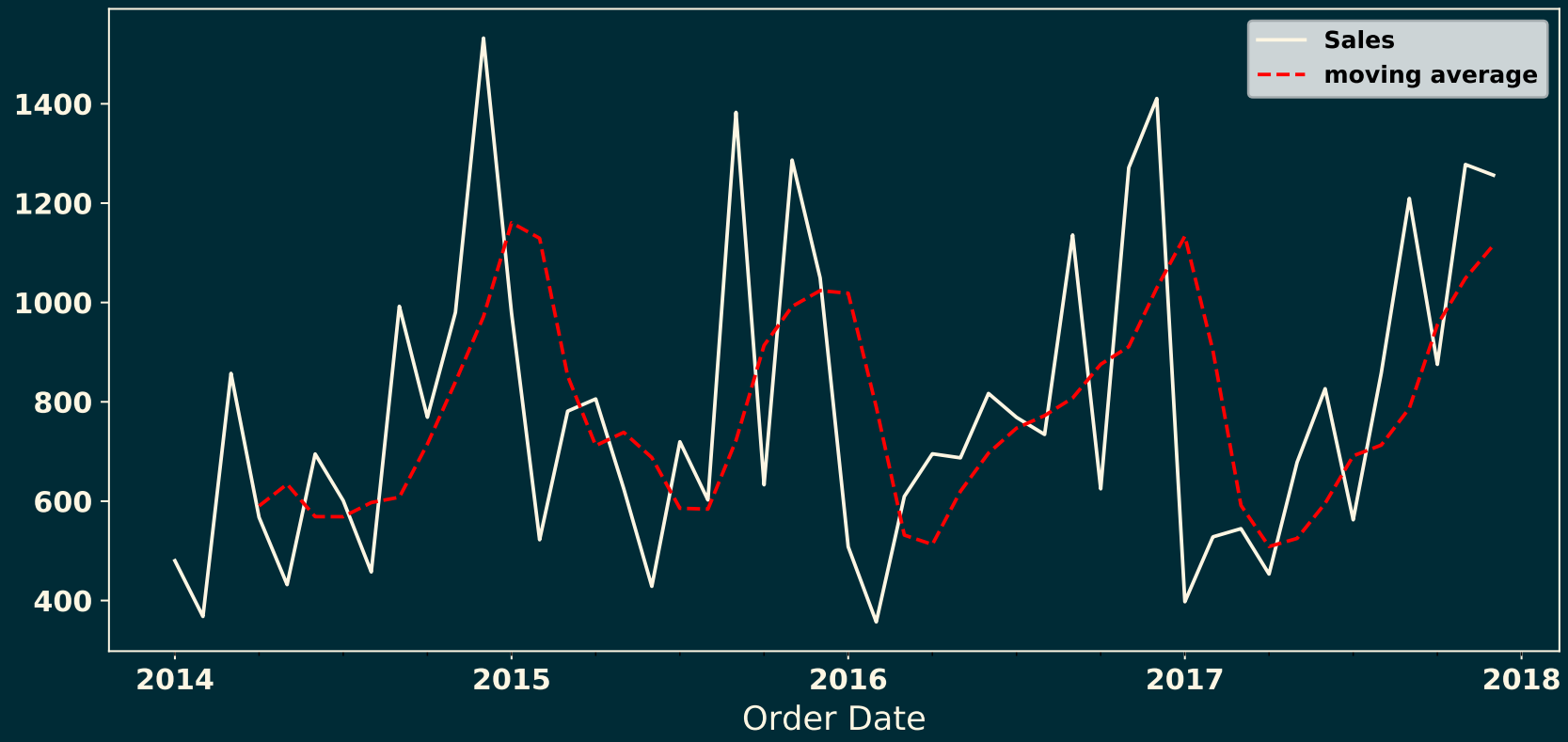
Long term trajectory, positive or negative

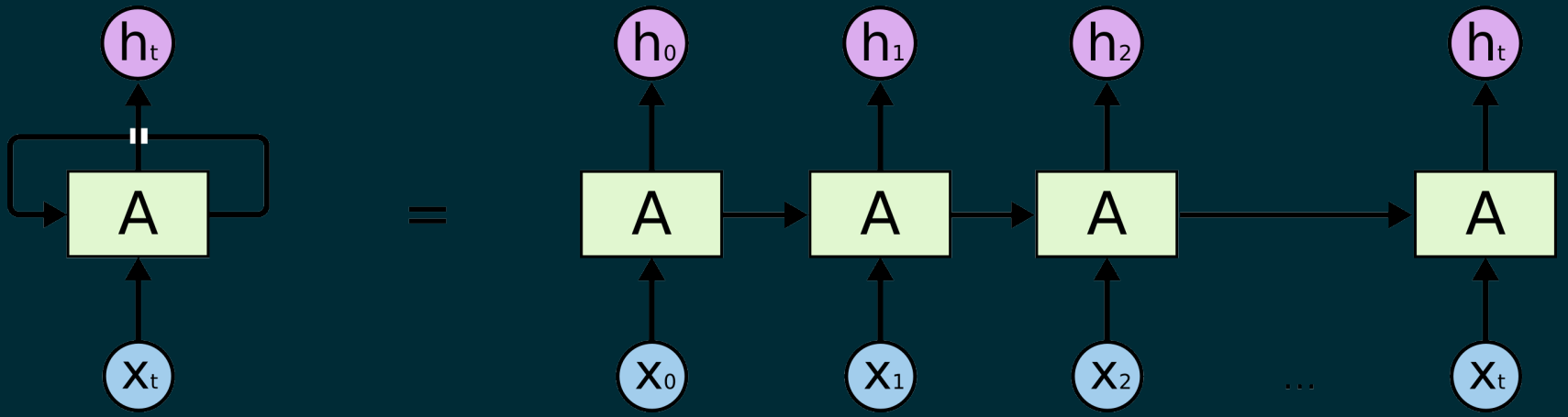
Repeating periods that are not related to the calendar


Sales
1400
1200
1000
800
600
400
2014
2015
2016
2017
2018
Order Date


Passengers
600
500
400
300
200
100
1949
1951
1953
1955
1957
1959
1961
Month


Observed
1500
1000
500
Trend
850
800
750
Seasonal
500
0
Residual
250
0
−250
2014
2015
2016
2017
2018
Order Date


Sales
moving average
1400
1200
1000
800
600
400
2014
2015
2016
2017
2018
Order Date



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Python and R

@pydatacardiff

@PyDiff

@CardiffRUG

@NikoletaGlyn

```
>>> import pandas
>>> nba = pandas.read_csv("nba_2013.csv")
```

```
> library(readr)
> nba <- read_csv("nba_2013.csv")
```

```
>>> import pandas
>>> nba = pandas.read_csv("nba_2013.csv")
```

```
> library(readr)
> nba <- read_csv("nba_2013.csv")
```

```
>>> train = nba.sample(frac=0.8, random_state=1)
>>> test = nba.loc[~nba.index.isin(train.index)]
```

```
> trainRowCount <- floor(0.8 * nrow(nba))
> set.seed(1)
> trainIndex <- sample(1:nrow(nba),
                       trainRowCount)
> train <- nba[trainIndex,]
> test <- nba[-trainIndex,]
```

```
>>> import pandas
>>> nba = pandas.read_csv("nba_2013.csv")
```

```
> library(readr)
> nba <- read_csv("nba_2013.csv")
```

```
>>> train = nba.sample(frac=0.8, random_state=1)
>>> test = nba.loc[~nba.index.isin(train.index)]
```

```
> trainRowCount <- floor(0.8 * nrow(nba))
> set.seed(1)
> trainIndex <- sample(1:nrow(nba),
                       trainRowCount)
> train <- nba[trainIndex,]
> test <- nba[-trainIndex,]
```

```
>>> from sklearn.linear_model import LinearRegression
>>> lr = LinearRegression()
>>> lr.fit(train[["fg"]], train["ast"])
>>> predictions = lr.predict(test[["fg"]])
```

```
> fit <- lm(ast ~ fg, data=train)
> predictions <- predict(fit, test)
```