Senior Data Scientist Exercise

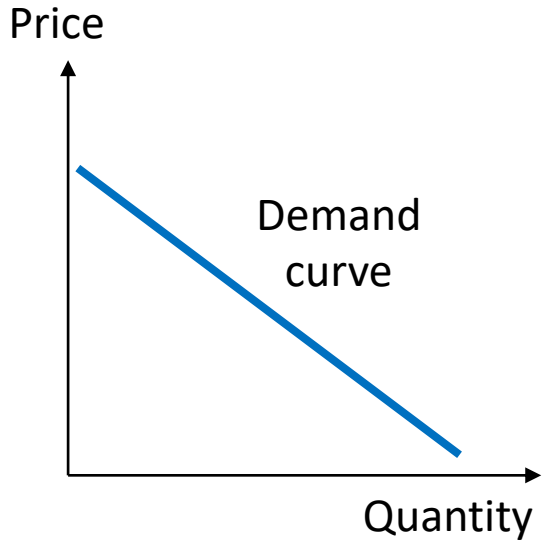
Case study results presented by

Lucia Leardini

1 – Pricing

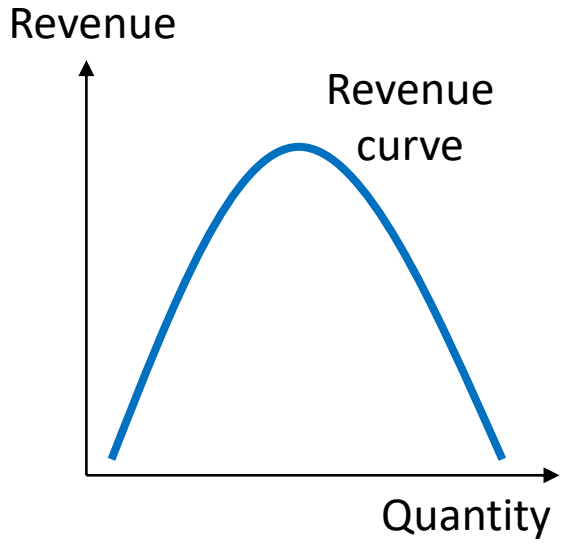
Definitions

The **demand** describes the relationship between the price of a product and the quantity of this product that is sold. Higher prices correspond to lower demand, and vice versa. A simplification of this relationship is shown in the plot on the upper right.



The **revenue** obtained from the sale of *n* units of a certain product is calculated considering the selling price, the demand and is influenced by other companies selling the same product on the same market.
Simplifying this model to a scenario where the company selling this product has the monopoly of the market (*i.e.,* there are no competitors):

revenue = (price per unit)·(number of units of product sold).

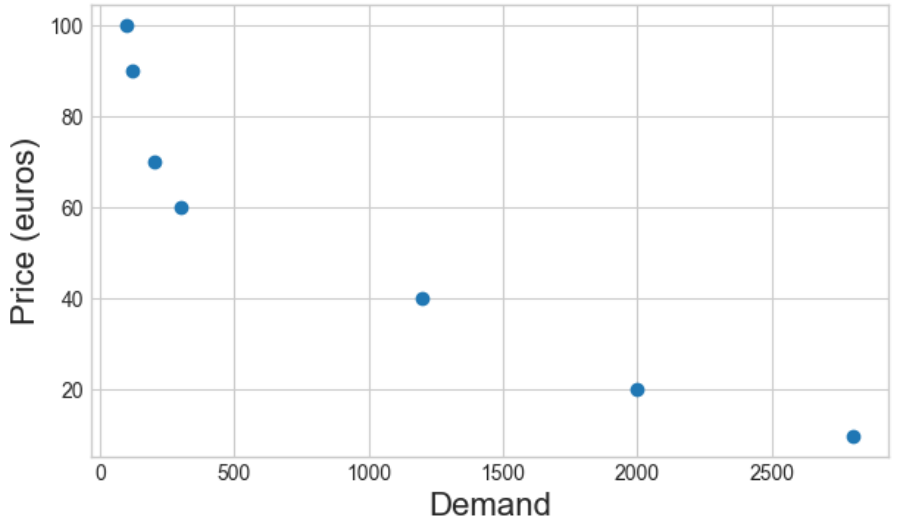


Pricing: problem statement 1/2

In our problem statement we have knowledge of

- the prices at which certain fashion items are sold
- the demand quantities, i.e. the number of people willing to buy an item, at various price levels

Their relationship (demand curve) for the observations gathered is shown in the figure on the right.



As we have no additional information, **we will consider the simplest case, in which the revenue** from the sale of an item **is calculated as the product of selling price and demand quantities**.

Our goal here is to **provide the optimal selling price at which we maximize revenue**.
This can be simply done by identifying **which price correspond to the maximum revenue among our data points**.

Pricing: maximize revenue

```python
def calculate_revenue(prices, demand):

    if prices.isnull().values.any() or (prices == 0).any():
        return ValueError('The prices argument contains nulls or zeros.')
    if demand.isnull().values.any():
        return ValueError('The demand argument contains nulls.')
    if (demand == 0).any():
        print('>>> WARNING: The demand argument contains zeros. Is this correct? <<<')

    return prices * demand
```

Null or zero prices raise an error

Null demand quantities raise an error (but zero demand is acceptable if, for example, the price is unreasonably high)

```python
def revenue_maximizing_price(prices, demand):
    """
    This function calculates the optimal price to maximize the revenue,
    given two pandas Series as input arguments.
    Arguments:
    :prices: list of the prices of the product of interest.
    :demand: list of the quantity of product bought at a certain price.
    """

    revenue = calculate_revenue(prices, demand)
    optimal_price = prices.loc[revenue.idxmax()]

    return print(f"\nThe optimal price that maximize revenue is {optimal_price} euros.\n")
```

Pricing: maximize revenue

With the values of price and demand we observed:

```
pricing = pd.DataFrame({
'prices': [100, 90, 70, 60, 40, 20, 10],
'demand': [100, 120, 200, 300, 1200, 2000, 2800] })
```

the price that maximize the revenue is 40 euros (index 4).

In a real-case scenario, price and demand have a non-trivial relationship.
The revenue function would be expressed using a single variable (usually the demand) to simplify the calculation.
The maximum revenue is then calculated from the first derivative of the revenue function with respect to the chosen variable.

Pricing: problem statement 2/2

Now that we know which price would maximize the revenue, we can move our focus on the other end of the sales like of a product: the supplier side.
Mass quantity is preferable in terms of resources consumed and production cost. The supplier applies a **progressive discount on the production cost of an item**, *i.e.* quantities of an item above a certain value cost progressively less than the initial quantities.
For a business to be successful, maximizing revenue is not enough if the production costs of the item sold are high.

Our goal here is to **provide the optimal selling price at which we maximize profit**.
Assuming the only costs we incur are production costs of the goods sold, the profit will be given by the difference between the revenue and the supplier cost.

We are again assuming a simple world here. The supplier cost is not the only one to be considered. Other costs a business might incur are logistics, rent of warehouses and shops, worker wages, utilities, marketing. All these need to be considered if the aim is to maximize profits.

Pricing: maximize profit

```python
def profit_maximizing_price(selling_price, demand):
    """
    This function calculates the optimal price to maximize profit,
    given, as input arguments, two pandas Series.
    Arguments:
    :selling_price: list of the prices of the product of interest.
    :demand: list of the quantity of product bought at a certain price.
    """
    supplier_price = []
    for q in demand:
        if q <= 50:
            supplier_price.append(15)
        elif q > 50 and q <=100:
            supplier_price.append(12)
        elif q > 100 and q <=200:
            supplier_price.append(8)
        else:
            supplier_price.append(4)

    # calculate the revenue, cost and profit
    revenue = calculate_revenue(selling_price, demand)
    cost    = demand * supplier_price
    profit  = revenue - cost

    # retrieve the optimal price using the index of the max profit
    optimal_price = selling_price.loc[profit.idxmax()]

    return  print(f"\nThe optimal price that maximize profit is {optimal_price} euros.\n")
```

Supplier price is given as a discrete function of the demand

Previously calculated revenue. Any exceptions are caught at this point

With the given supplier price and demand quantities, **the optimum price to maximize profit is 40 euros**.

2 – Regression

Ex-cursus: time-series analysis

Time-series are a collection of points gathered over an interval of time and ordered according to time/date. Each data point is usually collected at regular intervals. The more data points collected, the more reliable the conclusions drawn.

Some examples of where time-series can be useful:

- descriptive analysis: **illustrating the changes an observable undergoes over time**
- exploratory analysis: highlighting in a visual format insights over the data
- forecasting: **predicting the future using historical data**
- intervention analysis: describing the behaviour of an observable before and after a certain event takes place

Ex-cursus: time-series decomposition

A time-series can be separated into distinct components:

- **level**: the average value in the series
- **trend**: the increasing or decreasing value in the series
- **seasonality**: the repeating short-term cycle in the series
- **residual/noise**: the random variation in the series

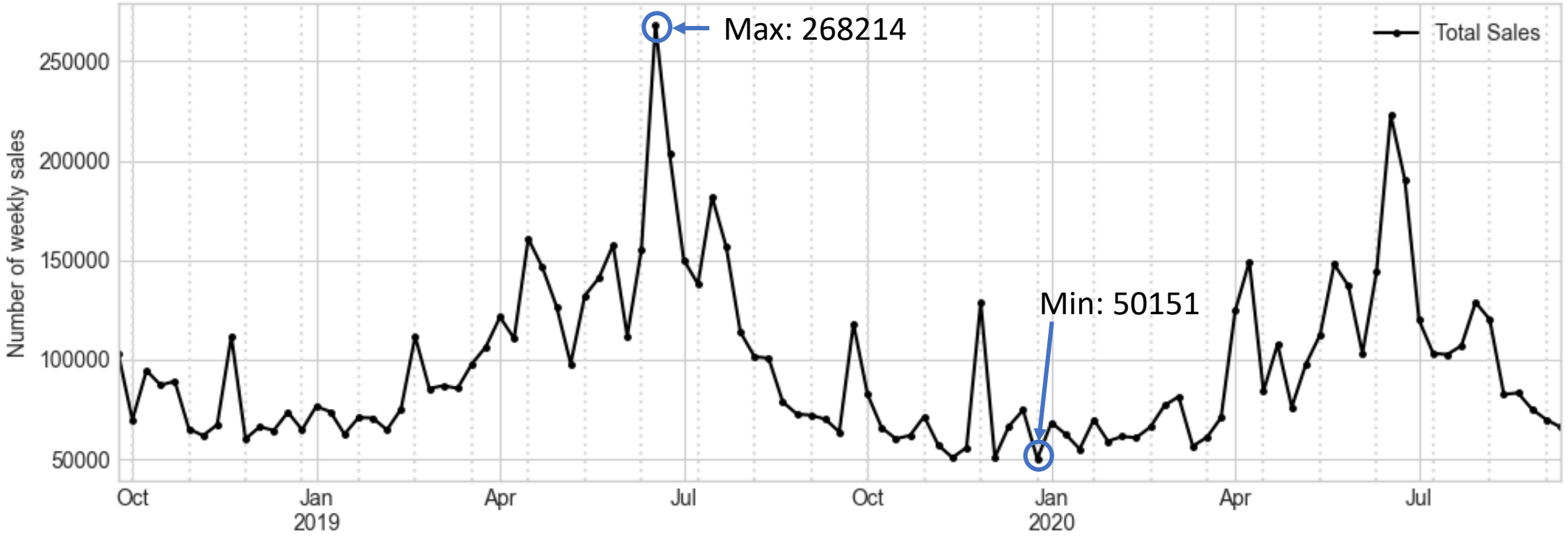
While *level* and *noise* are always present, *trend* and *seasonality* are not.
These components can be combined with an **additive** (describing linear changes over time) **or multiplicative** (non-linear) **model**.

The seasonal-trend decomposition (STL) method from `statsmodels`, a python module for statistical analysis and modelling, will be used with an additive model for the results presented here.

R is a great alternative to python `statsmodels`.

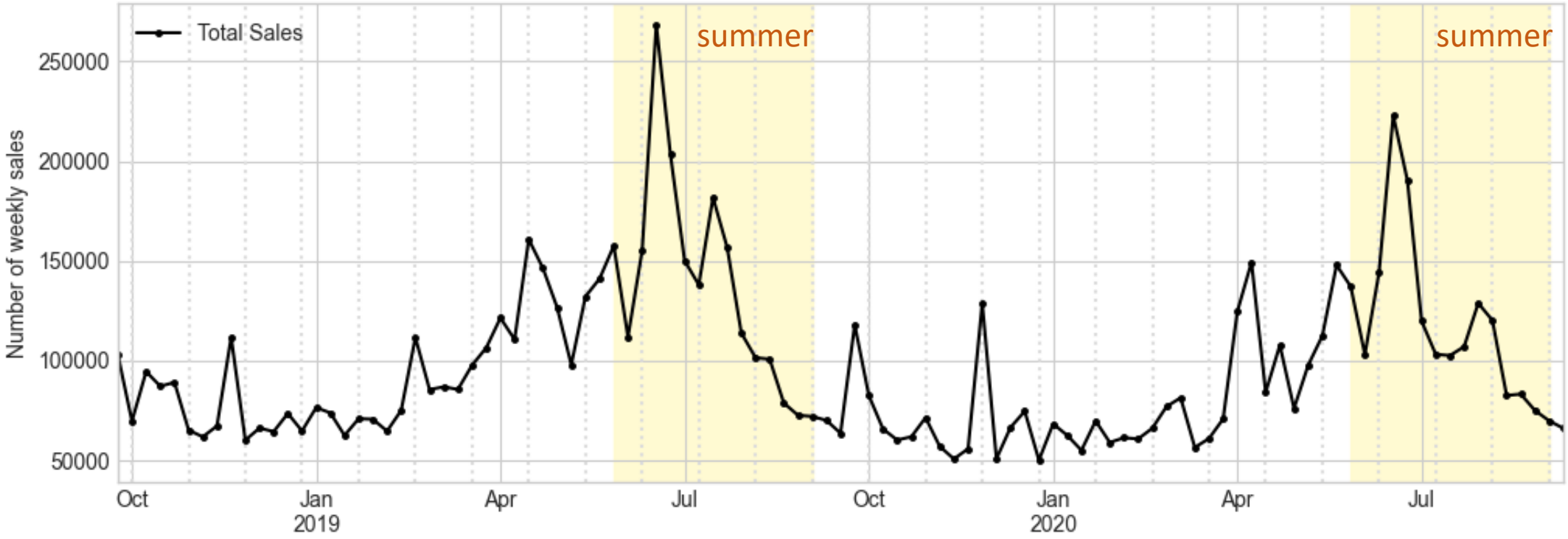
Weekly overall sales

We look into the overall sales for a given set of products over about two years of data.
The data is aggregated on a weekly level, for a total of 103 (data points) weeks*, from 2018-09-27 to 2020-09-10. The time-series distibution of the total sales is shown below. Vertical dotted line help in separating between month periods.



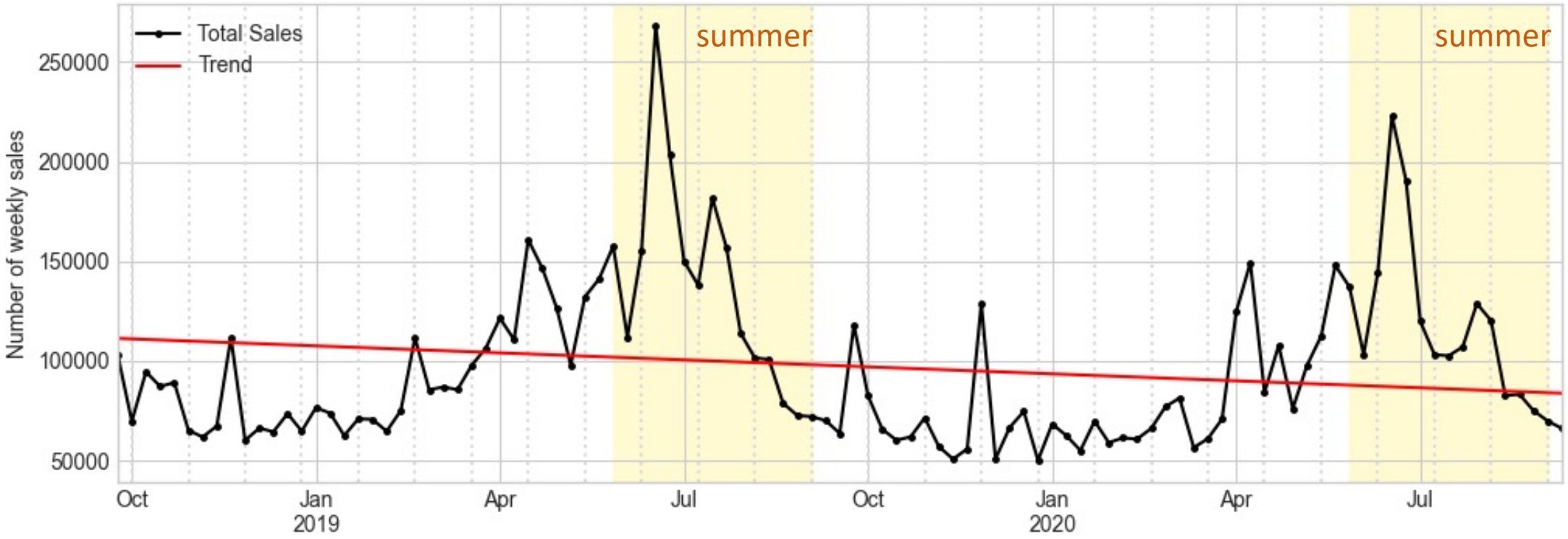
* It is not a complete cycle for a time-series. It would have to be 104 weeks over 2 years.

Weekly overall sales



The total sales time-series shows a clear **yearly seasonality** (*i.e.*, periodic nature of the data), **with higher sales in the summer months** (highlighted by the pale-yellow areas).

Weekly overall sales



The total sales time-series shows a clear **yearly seasonality** (*i.e.*, periodic nature of the data), **with higher sales in the summer months** (highlighted by the pale-yellow areas).

The **trend** (*i.e.,* defined as the long-term increase/decrease in the data of a time-series) is linear and decreasing over time.

Weekly sales – individual products

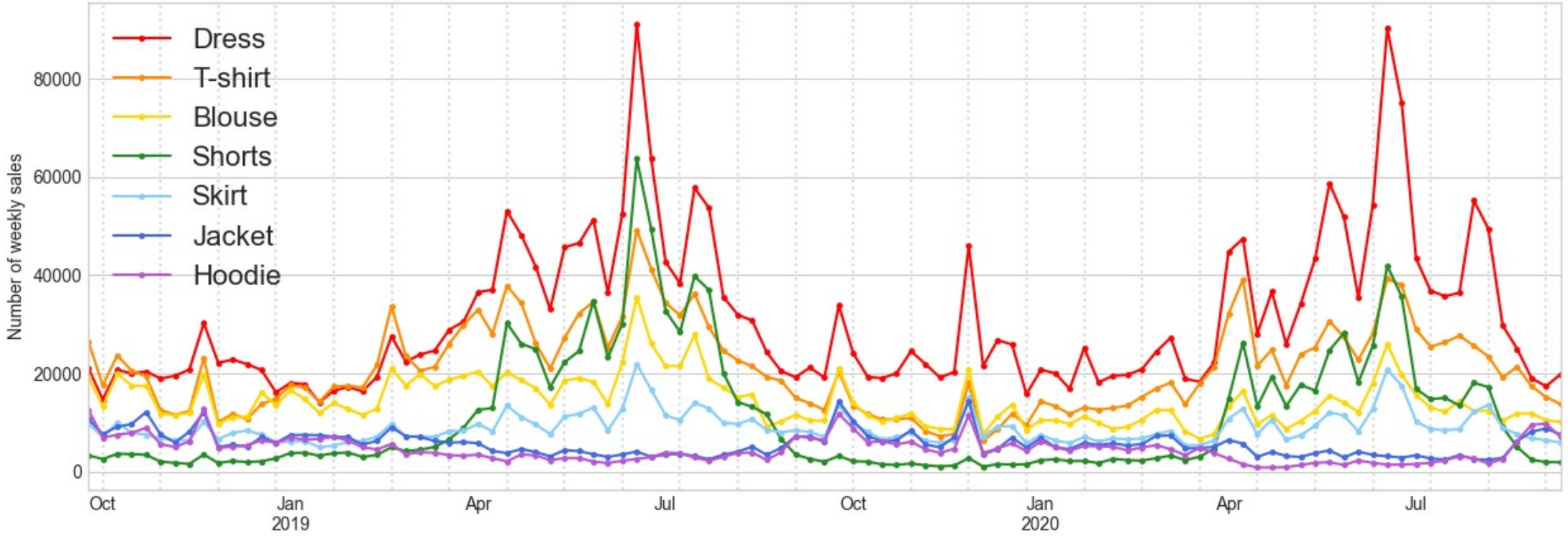
The time-series for the individual product are studies to highlight different sales patterns depending on the different item of clothing.

Below is a table with an overview of the values of each item.
The items are ordered according to the number of total sales, in descending order.

	Dress	T-shirt	Blouse	Shorts	Skirt	Jacket	Hoodie
mean	31116	21098	14373	11147	8952	5675	4556
min	14134	6277	6718	1001	5010	2405	825
max	91052	49272	35598	63788	21842	14411	12696

Weekly sales – individual products

The time-series for the individual product are studies to highlight different sales patterns depending on the different item of clothing.
The distributions are first visualised together for comparison.

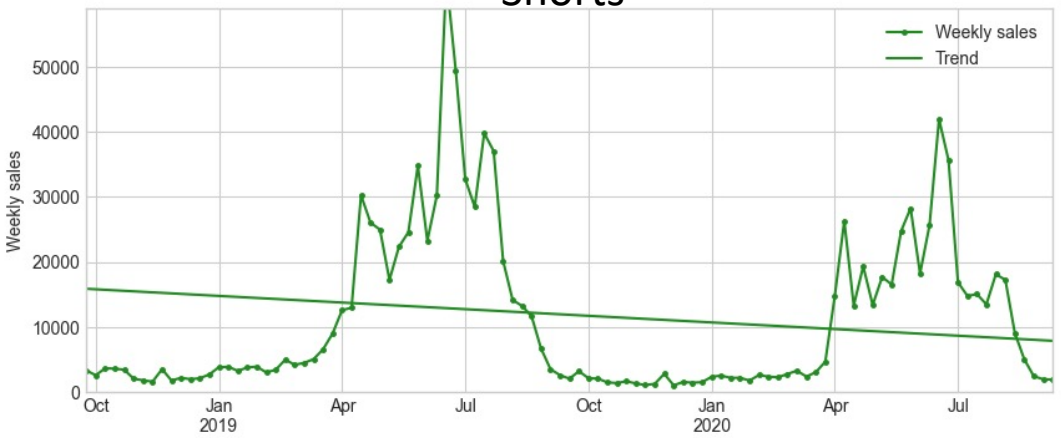


Weekly sales – individual products

Focusing on the individual products time-series and on their trends allows us to observe how the sales of the different items change depending on the period of the year considered.

Dress

Shorts



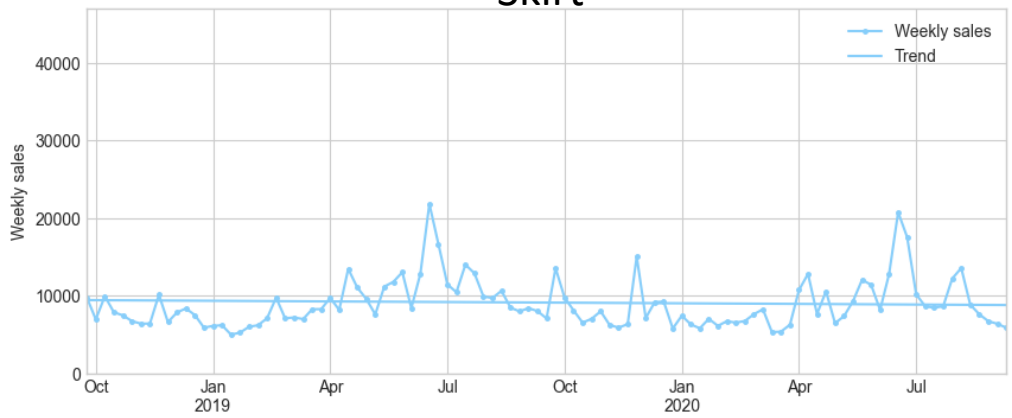
T-shirt

Dresses, t-shirts and shorts are the largest contributors to the sales summer peaks.

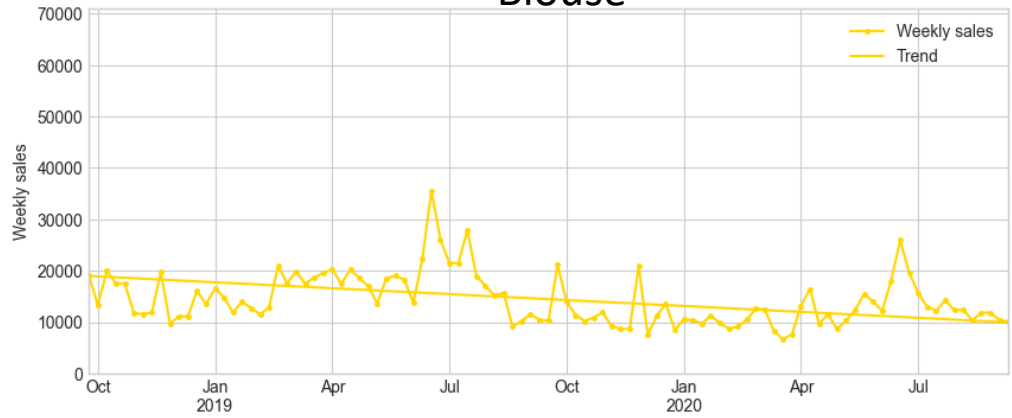
While dresses and t-shirts have similar baseline (off-peak) values, **shorts have almost no sales outside of the summer months**.

Weekly sales – individual products

Skirt



Blouse

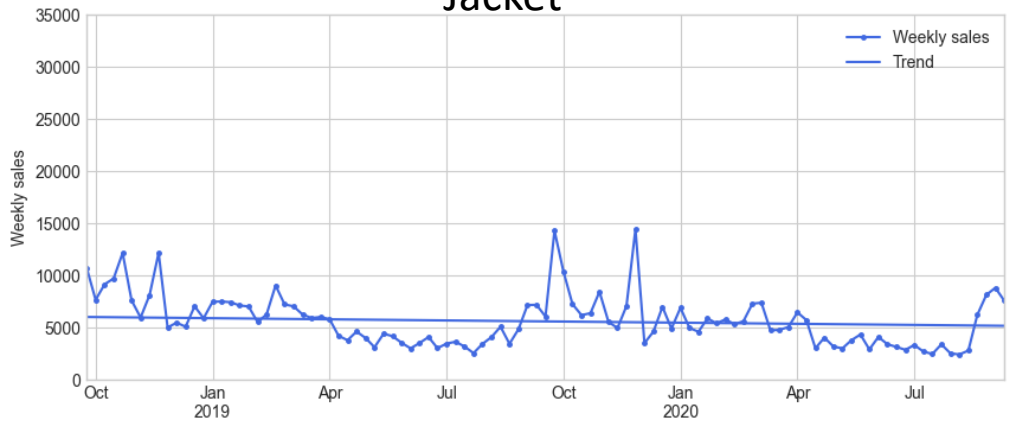


Skirts and blouses also contribute to the sales summer peak, although in a minor way.

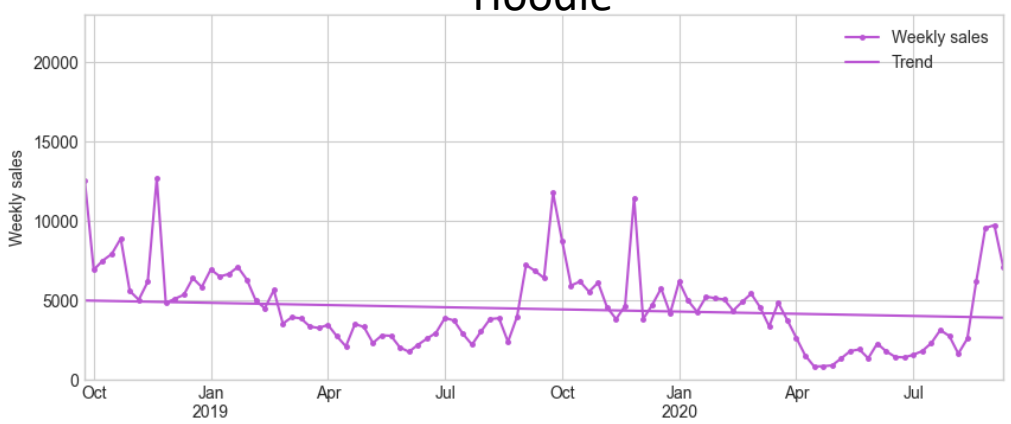
What is more interesting to notice here, is that **the average values of these products is relatively higher and constant throughout the year** (dresses also exhibit the same). We could understand why by reasoning that these items of clothing sometimes come with a formal or business cut. Therefore, an explanation of the constant component could be that these purchases are motivated by formal occasions or work.

Weekly sales – individual products

Jacket



Hoodie

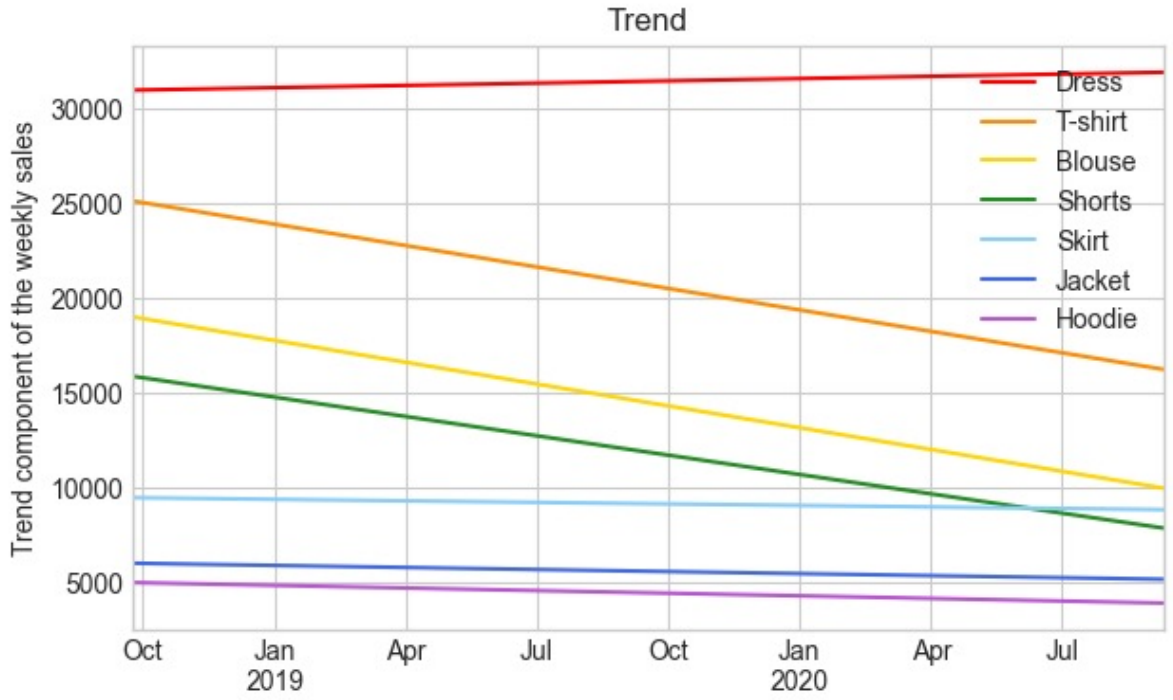


Jackets and hoodies are items typical of winter, hence we see **peaks in the sales of these products before and during the winter months**.

Weekly sales – individual products trends

The individual trends give a simpler view, removing from the comparison the seasonal peaks and random fluctuations.

One would assume that a linear and roughly constant trend is an acceptable and expected result for clothing items: people tend to buy new items regardless of their actual need.



We observe roughly constant trends for all items except for the ones of **t-shirts, blouses and shorts, that are decreasing with time**.

A more in-depth analysis would be required here to explain such behaviour, potentially also considering external factors or fashion trends.

Correlation between products

We mentioned earlier that the sales of items such as dresses, skirts and blouses might be influenced by their cut. Other factors that might influence sales of a certain typology of item are age and sex of the buyer, global and local fashion trends.

Systematically studying the correlation between the sales of each product, is one way to get additional insights on these aspects.
The **correlation coefficient matrix** is given by

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}}$$

where *C* is the covariance matrix. Covariance is an indication of how closely two variables vary together.

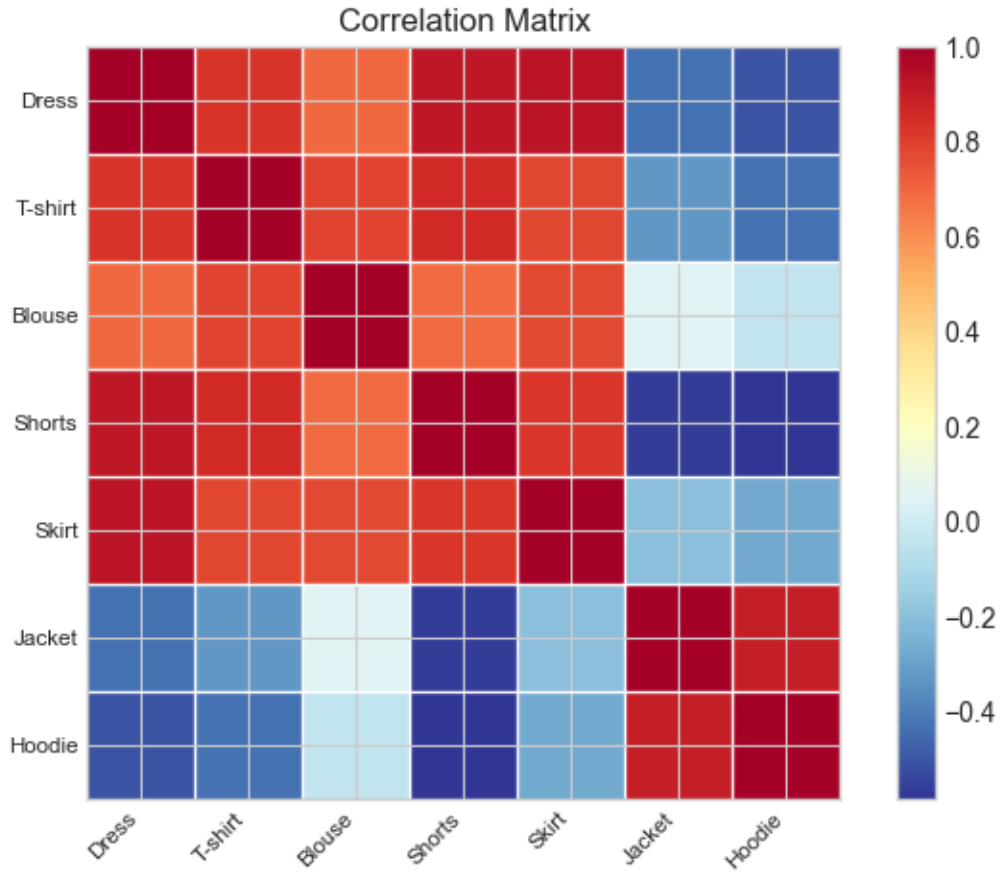
Correlation between products

The largest correlations observed are linked to the seasonal aspect of the sales.

Fro example:

- sales of dresses/shorts/skirts and jackets/hoodies are positively correlated
- sales of shorts/jackets are negatively correlated

	Dress	T-shirt	Blouse	Shorts	Skirt	Jacket	Hoodie
Dress	1.00	0.83	0.69	0.91	0.93	-0.43	-0.51
T-shirt	0.83	1.00	0.79	0.86	0.78	-0.33	-0.43
Blouse	0.69	0.79	1.00	0.69	0.77	0.05	-0.04
Shorts	0.91	0.86	0.69	1.00	0.83	-0.56	-0.58
Skirt	0.93	0.78	0.77	0.83	1.00	-0.20	-0.27
Jacket	-0.43	-0.33	0.05	-0.56	-0.20	1.00	0.90
Hoodie	-0.51	-0.43	-0.04	-0.58	-0.27	0.90	1.00



Sales forecasting

Forecasting is a powerful tool for businesses because it allows them to prepare for future sales (or lack of thereof) and set in place new strategies based on these results that might increase their profit or reduce losses.

Logistics and provisioning would also benefit from forecasting, to avoid surpluses and optimize the movement of merchandize between warehouse and shops.

For this exercise, we focus on **dresses to forecast the sales for the 5 weeks (periods) after the last observed data point (2020-09-10)**.
For the forecast we use SARIMAX, (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors) model from `statsmodels`.

Sales forecasting

```
# resampling to have complete cycles, improves estimation
sales_dress = sales['Dress'].resample(sales['Dress'].index.inferred_freq).last()
model_dress = sm.tsa.SARIMAX(sales_dress,
                      order=(1,0,0),
                      seasonal_order=(1,1,1,52),
                      trend='t',
                      freq=sales_dress.index.inferred_freq).fit(disp=False)
forecast_dress = model_dress.get_forecast(steps=steps).summary_frame()
```

specifies the orders of the non-seasonal, denoted as *(p, d, q)*:

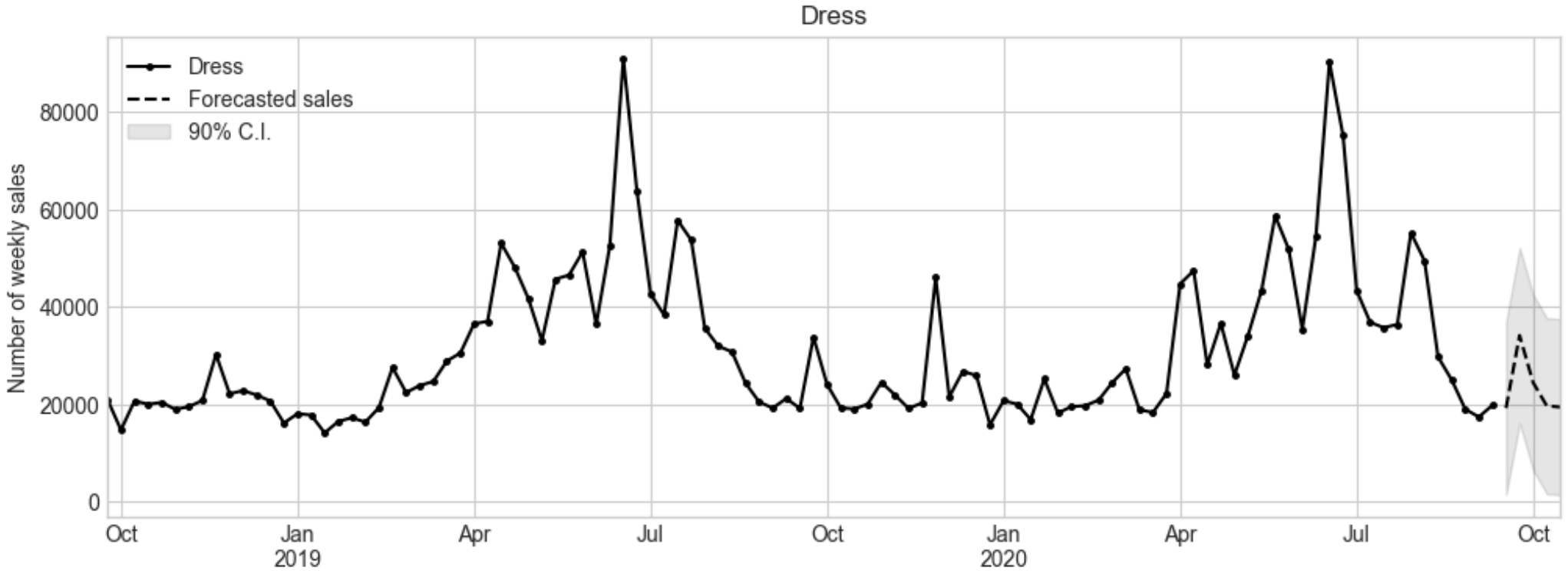
- *p* represents the order of the AutoRegressive (AR) component, which captures the linear relationship between the current observation and its past values
- d represents the order of differencing required to make the time series stationary
- q represents the order of the Moving Average (MA) component, which captures the linear relationship between the current observation and past noise (residuals) terms

specifies the orders of the seasonal components, denoted as *(P, D, Q, s)* :

- *P* is the seasonal order of the Seasonal AutoRegressive (SAR) component, capturing the seasonal linear relationship between current observation and its past values, separated by a seasonal period *s*
- *D* is the seasonal order of differencing for seasonal stationarity
- *Q* is the seasonal order of the Seasonal Moving Average (SMA) component, capturing the seasonal linear relationship between current observations and residuals, separated by a seasonal period *s*.

Sales forecasting

The forecasted point seems to be in line with the previous seasonal data points behaviour, although the confidence interval (chosen at 90%) is quite large.



Using a longer time series and exogenous (*i.e.,* external) inputs might reduce the intervals to more reliable values.

Outlook

Concerning the second part of this case study (Regression), there are various ways to improve on the results obtained here.

- Given more information about the time-series, the trends can be given a more detailed interpretation.
- Differentiations in the demography of the buyers could also help in gathering additional insights on the observed distribution and plan targeted marketing campaigns
- Relying on different tooling as method of comparison can improve on the limitations of using just one. The use of R, that is specific for statistical heavy problems might improve on the conclusions drawn.
- Exploring different forecasting models and library, improving on the parameter tune could likewise bring much better results.