“How-to”-guide for Pricing codes

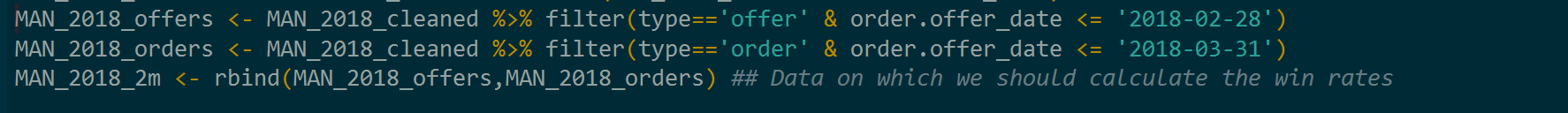
# Pricing

## Input files / Output files

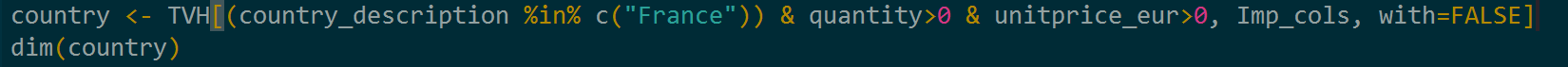
|  |  |  |  |
| --- | --- | --- | --- |
| **Pricing Code** | **File Name in code** | **Name of file** | **Location/Description** |
| Win Rate Code | Country | MAN 2017.csv | Input file(Offers/Orders data) for which Win Rate code to be calculated  (Input from Celine) |
| Win Rate Code | Qty\_discounts | Quantity\_discounts\_simplified.csv | Quantity discounts file (Input from Celine) – restructured to used in model  Location: TVH McKinsey Pricing project\Pricing handover\01\_Win\_Rate\03\_Data |
| Win Rate Code | All\_FM\_SKUs | All\_FM\_SKUs.csv | List of All Fast moving SKUs - Calculated from All Countries data consolidated  Location: TVH McKinsey Pricing project\Pricing handover\01\_Win\_Rate\03\_Data |
| Win Rate Code | country\_new1 | France\_FinalOffers.csv | Output file with Final Offers tagged |
| Customer Clustering | MAN\_2018 | MAN 2018.csv | Input file for which Clusters to be created |
| Customer Clustering | cust\_char1 | customer\_characteristics\_part1\_v2.csv | Customer Characteristics file with customer segment description  Location: TVH McKinsey Pricing project\Pricing handover\02\_Clustering\03\_Data |
| Customer Clustering | Cust\_UARC1 | Clusters.csv | Output file with mapping of Clusters and Related Customers |
| Creating Price Ranges | Final\_Offers\_data | France\_3yrs\_Final\_Offers.csv | Final Offers tagged – Output from Win Rate Code  Location: TVH McKinsey Pricing project\Pricing handover\01\_Win\_Rate\02\_Output\ France\_WR\_Output\_InStock\_Jan15\_Nov17.csv |
| Creating Price Ranges | Clusters | Final\_Clusters\_data.csv | Clusters output from Customer Clustering code  Location: TVH McKinsey Pricing project\Pricing handover\02\_Clustering\02\_Output |
| Creating Price Ranges | Cluster\_SKU\_PRanges\_Final | Cluster\_SKU\_PRanges\_Final.csv | Price Ranges created for SKU-Cluster combination  Location: TVH McKinsey Pricing project\Pricing handover\03\_Price\_Ranges\02\_Output\ Cluster\_SKU\_Price\_Ranges\_France.csv |
| Normalization | Final\_Offers\_data | Appended Final Offers from France/Italy and UK | Final Offers tagged for 3 countries – Output from Win Rate Code  Location: TVH McKinsey Pricing project\Pricing handover\01\_Win\_Rate\02\_Output\ |
| Normalization | Clusters | Final\_Clusters\_data.csv | Clusters output from Customer Clustering code  Location: TVH McKinsey Pricing project\Pricing handover\02\_Clustering\02\_Output |
| Normalization | Country\_SKU\_avg\_price2 | FR\_IT\_UK\_Normalization\_table.csv | A sku-country level table for normalization of sku level prices for each Country  Location: TVH McKinsey Pricing project\Pricing handover\03\_Price\_Ranges\02\_Output\ Normalization\_table\_FR\_IT\_UK.csv |
| Normalization | Cluster\_SKU\_PRanges\_Final | Cluster\_SKU\_PRanges\_Final.csv | Price Ranges defined for SKU-Cluster combination  Location: TVH McKinsey Pricing project\Pricing handover\04\_Normalization\02\_Output\Cluster\_SKU\_Price\_Ranges\_norm\_FR\_IT\_UK.csv |
| Pricing Model | Cluster\_SKU\_Pranges | Cluster\_SKU\_PRanges\_Final.csv | Input : Either from Individual Countries or Normalized countries data |
| Pricing Model | Price\_recomm\_table | Price\_recomm\_table.csv | Price Recommendation table for France :  Location: TVH McKinsey Pricing project\Pricing handover\05\_Pricing\_Model\02\_Output\Price\_Recommendation\_table\_France.csv |
| Self-Learning Pricing Model | Cluster\_SKU\_Pranges\_Prev | Cluster\_SKU\_PRanges\_Final.csv | Output of Code - Creating Price ranges from Previous refresh |
| Self-Learning Pricing Model | Cluster\_SKU\_Pranges\_New | Cluster\_SKU\_PRanges\_Final.csv | Output of Code - Creating Price ranges from Latest refresh |
| Self-Learning Pricing Model | SL\_model | SL\_model | Self\_Learning\_model.csv |

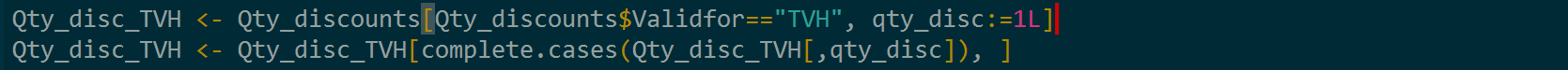
Following guide explains what needs to be changed/modified in Pricing codes to smoothly run over time:

1. **Win Rate(WR) Code:**
2. **Add New data**: The WR code is currently run for 3 years (Jan 2015-Dec 2017). To add new timeline, run latest months of data and append in existing file.
3. **Offers/orders selection**: If you wish to run latest n months of data, use n-1 months of offers and n months of orders since we look out next 30 days to check if offer is won/lost. e.g. If WR code is run for Jan18 – Mar18 data, take offers from Jan18-Feb18 and orders from Jan18-Mar18. (for reporting purposes, the WR is available for 5 months, i.e.

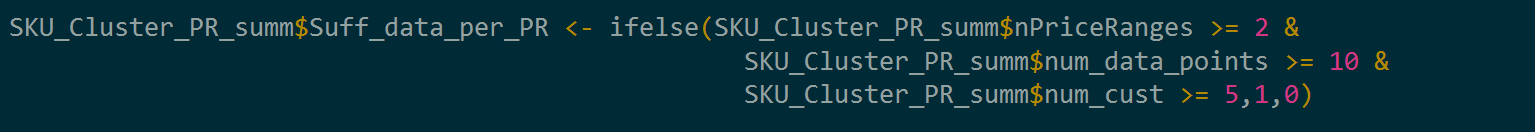


1. **Run for different Country**: Filter selected country in R code. Filter corresponding company from quantity discounts file (**Validfor column**). (**TVHSA** for South Africa, **QFS** for Australia, **TVPNZ** for NewZealand)





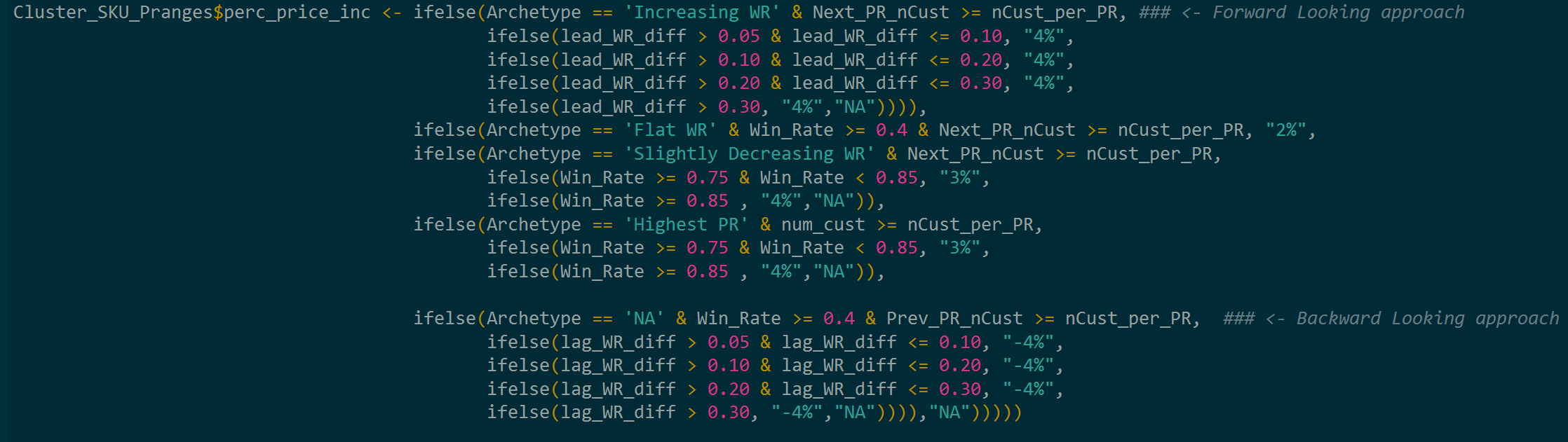
1. **Item level analysis**: To run WR code at item\_no level, change sku to item\_no for tagging final offers and orders w/o offers in for-loops.
2. **Customer Clustering Code:** Customer Clusters are refreshed based on two main variables :
3. **Customer annual size:** Please ensure to annualize the customer revenue accordingly. i.e. If using 3 months’ of data, use x4 to annualize, if 6 months’ of data, use x2 etc.
4. **Customer main segment:** Include latest customer information(Repairer/Trader etc.) as soon as it’s available.
5. **Price Ranges:** Thiscodeuses output from (1) and (2) and creates WR Graphs at sku-cluster level.Pricerangesare defined always in quintiles. One can change definition of how to define sufficient Win Rate Graphs in code. Current definition is :
6. >10 data points per WR Graph
7. At least 2 price ranges
8. >5 customers per price range



1. **Normalization:** This is an optional step, if one wish to normalize few countries’ together to increase more data points per WR Graph. Include all countries’ WR output data that you wish to normalize, specify priority of normalization(i.e. reference country order, If wish to normalize Russia, Poland, Turkey -> depends on where most SKUs are sold, Russia could be the first reference country, if few SKUs are not sold in Russia, Poland could be next reference country for remaining SKUs and so on so forth). **The output is for each sku of the base country – what is the reference country and normalization factor.**

After normalization, we build the price ranges with more sufficient data.

1. **Pricing Model:** This code uses price ranges(3)/(4) as input and assign archetype shape for next price range and recommends price increase factor.
2. **Exclude Backward looking :** If you wish to use only forward looking approach(increase prices), exclude backward looking part(*Comment the backward looking part*).
3. **Price Increase factor:** Price increases are currently mentioned in data, one can change conditions and change the price increase factors for each archetype.



1. **Self-learning pricing model:** This code uses output from last 2 refreshes and compares the WR differences in 2 price ranges over time. If Win Rate difference increase widely(**>30%** - adjustable), price increase isn’t recommended. **Please test the code exhaustively with actual data over 2 refreshes before implementing.**