Datenbanksysteme - Abgabe DWH

Lukas Forsthuber, Peter Jernej, Christian Brandstätter

Inhalt

Kurzübersicht Workflow....1
Installation Pentaho....1
Auswahl und Kurzbeschreibung Datenset....2
Transformation Datenset in Sternschema (Python)....4
Import in Pentaho und Schreiben der Tabellen in postgres Datenbank....4
Mehrdimensionale Analyse Datenset....8

Kurzübersicht Workflow

1. Installation JDK, Pentaho, postgres
2. Herunterladen Sales-Datensatz
3. Vorbearbeitung Datensatz mit Python -> Sternschema
4. Import in Pentaho
5. Übertragen in postgres-db
6. Re-Import von postgres-db
7. Mehrdimensionale Analyse pentaho
8. Datenexport -> csv
9. Darstellung der Ergebnisse in R

Installation Pentaho

Für den Betrieb von Pentaho wurde das Betriebssystem Windows 10 verwendet.

Die Pentaho–Suite wurde heruntergeladen von Sourceforge[1]. Für den Betrieb wurden auch Java 8 inkl. JRE und JDK[2] installiert. Danach wurde das System neu gestartet, um die Systempfade zu aktualisieren.

Vor der ersten Verwendung von Pentaho wurde die Datei set-pentaho-env.bat (Pentaho-Stammverzeichnis) ausgeführt. Die Pentaho-Suite lässt sich mit der Datei spool.bat im Pentaho-Stammverzeichnis öffnen. Zusätzlich wurden Postgres 12.2 und pgAdmin 4.20 installiert[3].

[1] https://sourceforge.net/projects/pentaho/files/Pentaho%209.0/; April 2020
[2] https://www.oracle.com/java/technologies/javase-jdk8-downloads.html (jdk-8u251-windows-x64), April 2020
[3] http://www.enterprisedb.com/thank-you-downloading-postgresql?cid=48, April 2020

Zum Importieren der CSV-Dateien waren noch weitere Schritte erforderlich:

Installieren eines Plugins im Markplatz: Import from txt. Danach musste ein Pentaho-Neustart durchgeführt werden.

Auswahl und Kurzbeschreibung Datenset

Auswahl:

Link: https://www.kaggle.com/kyanyoga/sample-sales-data#sales_data_sample.csv

Es handelt sich beim Datenset um ein künstliches Verkaufsdatenset. Im Folgenden ein Zitat von Kaggle zum Datenset:

> *Sample Sales Data, Order Info, Sales, Customer, Shipping, etc., Used for Segmentation, Customer Analytics, Clustering and More. Inspired for retail analytics. This was originally used for Pentaho DI Kettle, But I found the set could be useful for Sales Simulation training.*
>
> *Originally Written by María Carina Roldán, Pentaho Community Member, BI consultant (Assert Solutions), Argentina. This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. Modified by Gus Segura June 2014.*

Das Datenset liegt als csv vor. Die Charakter-Enkodierung ist latin-1. Das Datenset enthält 25 Spalten und 2823 Zeilen. Details zu den Variablen werden in der nächsten Tabelle dargestellt (nächste Seite). Offenbar gab es bei der Daten-Erstellung einen Fehler, die Variable Priceeach hatte bei 100 eine Obergrenze. Das wurde über die offenkundige Beziehung Sales = PRICEEACH * QUANTITYORDERED korrigiert. Zur Richtigstellung wurde über die Berechnung Sales / QUANTITYORDERED der korrekte PRICEEACH eingefügt.

Zusätzlich wurde ein Primary Key erzeugt, durch die Zusammenfügung von ORDERNUMBER und ORDERLINENUMBER.

Die Datei wurde in ein Stern-Schema umgewandelt. Dabei wurden die zentralen Variablen, die die Bestellungen (Orders) betreffen, im Zentrum (Dimension center) belassen (siehe nächste Tabelle). Um die Daten aufzuteilen, wurde in Python eine Funktion definiert, die für Unique-Entries aufsteigende Integer-Sequenzen erzeugt, die sich auch mit einem Text kombinieren lassen.

Nach unzähligen Fehlversuchen des csv-Imports in Pentaho wurde der Zwischenweg über Python ausgewählt. Es gab laufend irgendwelche Daten-Typ-Probleme, zu Beginn auch wegen der Datei-Enkodierung. Nachdem alle Datentypen auf String gesetzt wurden, verschwanden diese zwar, das hätte aber andere negative Folgen nach sich gezogen.

```python
def create_key(series, first = ""):
    """Function to create a new key_value, if some entries are not unique

    :param series: pandas series
    :param first: enables the addition of a string before integer
    :returns: pandas series with integer / string keys
    :rtype: series int64 or object if first is given

    """
    step1 = list(series.unique())
    step2 = dict()

    for i in range(0, len(step1)):
        step2[step1[i]] = i+1

    if len(first) > 0:
        for k, v in step2.items():
            step2[k] = first+str(v)

    return series.map(step2)
```

Abbildung: Python-Funktion zum Definieren von Keys

Variable	Anzahl befüllter Zellen	Datentyp	Dimension
ORDERNUMBER	2823	int64	**Center**
QUANTITYORDERED	2823	int64	**Center**
PRICEEACH	2823	float64	**Center**
ORDERLINENUMBER	2823	int64	**Center**
SALES	2823	float64	**Center**
ORDERDATE	2823	object	**Ordertimes**
STATUS	2823	object	**Center**
QTR_ID	2823	int64	**Ordertimes**
MONTH_ID	2823	int64	**Ordertimes**
YEAR_ID	2823	int64	**Ordertimes**
PRODUCTLINE	2823	object	**Products**
MSRP	2823	int64	**Products**
PRODUCTCODE	2823	object	**Products**
CUSTOMERNAME	2823	object	**Customer**
PHONE	2823	object	**Customer**
ADDRESSLINE1	2823	object	**Customer**
ADDRESSLINE2	302	object	**Customer**
CITY	2823	object	**Customer**
STATE	1337	object	**Customer**
POSTALCODE	2747	object	**Customer**
COUNTRY	2823	object	**Customer**
TERRITORY	1749	object	**Customer**
CONTACTLASTNAME	2823	object	**Customer**
CONTACTFIRSTNAME	2823	Object	**Customer**
DEALSIZE	2823	object	**Center**

Sternschema (Python)

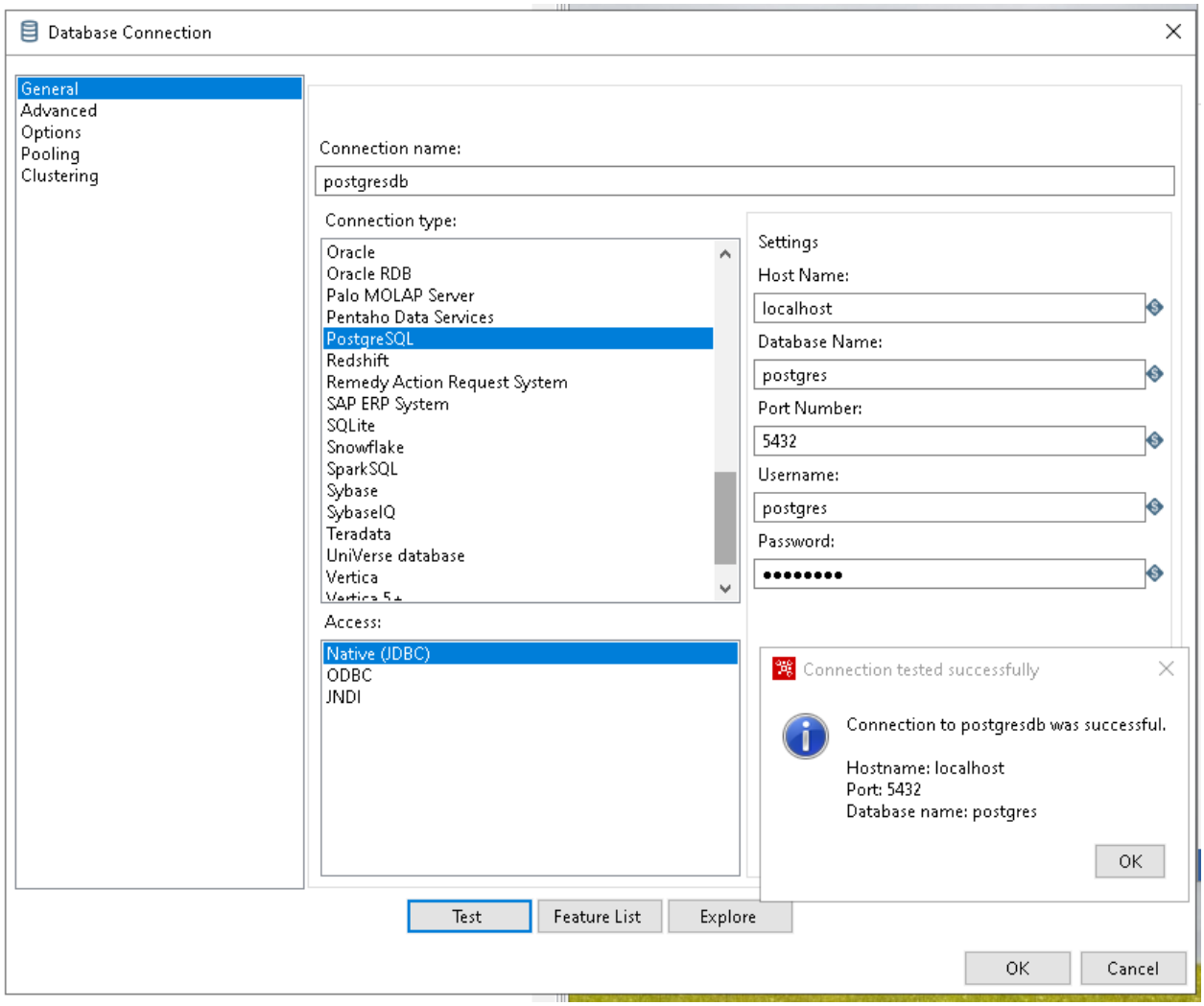
Für die Aufteilung in das Sternschema in die Dimensionen wurden verschiedene Schlüssel erzeugt. In der Zentraltabelle wurden die Schlüssel hinterlegt. PK (Primary Key) enthält die Ordernumber, einen Unterstrich und die Orderlinenumber, ON_ID basiert auf der Ordernumber (notwendig für die Ordertimes, da diese ja mit den Orders zusammenhängen), PR_ID basiert auf dem Productcode und CU_ID auf dem Customername.

Die Dimension Customer enthält auch die räumliche Dimension, da diese in dem verwendeten Datensatz über die Kunden einbezogen wird. In dem gegenständlichen einfachen Datenschema ergibt sich die Verknüpfung mit den Dimensionen über die einzelnen Keys.

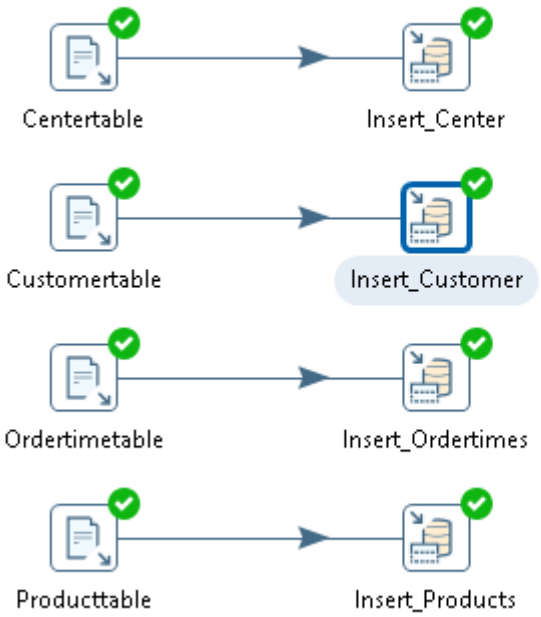
Schlüssel	Dimension
ON_ID	Ordertimes
CU_ID	Customer (auch räumlich)
PR_ID	Product

Import in Pentaho und Schreiben der Tabellen in postgres Datenbank

Durch die Vorbereitung in Python, war der Datenimport und die Übertragung in die Datenbank dann relativ einfach möglich.



Der Übertragungsprozess hat nach der Anpassung der Spaltengrößen funktioniert.



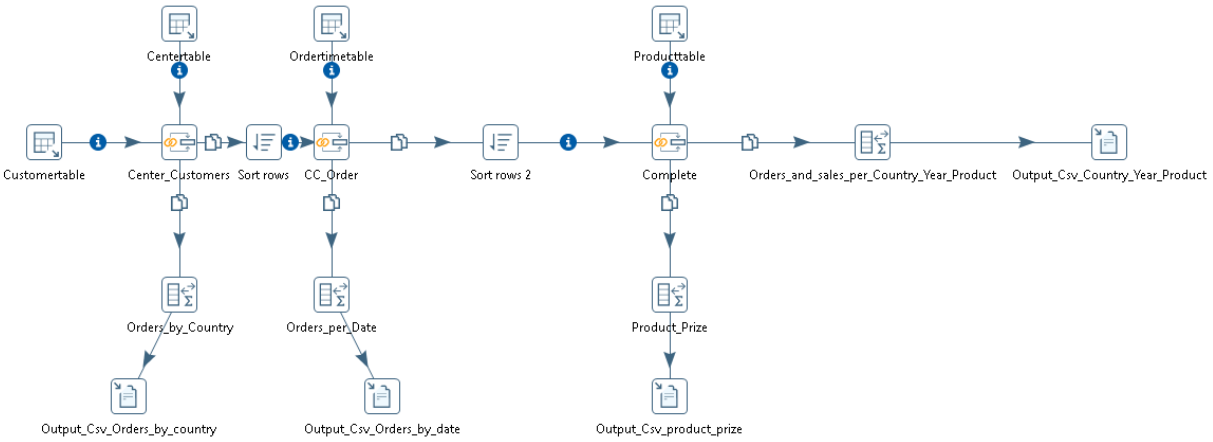
Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

```
2020/05/09 00:33:11 - Customertable.0 - Header row skipped in file 'C:\hitachi\data_import\customers.csv'
2020/05/09 00:33:11 - Customertable.0 - Finished processing (I=2824, O=0, R=0, W=2823, U=0, E=0)
2020/05/09 00:33:11 - Centertable.0 - Finished processing (I=2824, O=0, R=0, W=2823, U=0, E=0)
2020/05/09 00:33:11 - Ordertimetable.0 - Header row skipped in file 'C:\hitachi\data_import\ordertimes.csv'
2020/05/09 00:33:11 - Ordertimetable.0 - Finished processing (I=2824, O=0, R=0, W=2823, U=0, E=0)
2020/05/09 00:33:11 - Producttable.0 - Finished processing (I=2824, O=0, R=0, W=2823, U=0, E=0)
2020/05/09 00:33:12 - Insert_Products.0 - Finished processing (I=2823, O=3, R=2823, W=2823, U=0, E=0)
2020/05/09 00:33:12 - Insert_Ordertimes.0 - Finished processing (I=2823, O=14, R=2823, W=2823, U=0, E=0)
2020/05/09 00:33:13 - Insert_Center.0 - Finished processing (I=2823, O=2048, R=2823, W=2823, U=0, E=0)
2020/05/09 00:33:13 - Insert_Customer.0 - Finished processing (I=2823, O=2677, R=2823, W=2823, U=0, E=0)
2020/05/09 00:33:13 - Spoon - The transformation has finished!!
```

Mehrdimensionale Analyse Datenset

Der Olap-Prozess wurde mehrstufig abgebildet. Zuerst wurden die Kundendaten zur Zentraltabelle gejoined, dann die zeitliche Dimension und abschließend die Produktdetails. Das ermöglicht die mehrdimensionale Betrachtung im Prozessverlauf.



Um die Daten fehlerfrei verknüpfen zu können (inner joins), mussten diese vorher aufsteigend sortiert werden (Sort rows und Sort rows 2). Die eigentliche Analyse passiert mit den Group_by-Memory-Operation, wo verschiedene Variablen unterschiedlicher Join-Stufen zusammen aggregiert werden konnten.

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Customertable	0	0	92	92	0	0	0	0	Finished	0.1s	1,394
2	Centertable	0	0	2823	2823	0	0	0	0	Finished	0.1s	24,336
3	Center_Customers	0	2915	5646	0	0	0	0	0	Finished	0.5s	11,762
4	Sort rows	0	2823	2823	0	0	0	0	0	Finished	0.6s	4,892
5	Producttable	0	0	109	109	0	0	0	0	Finished	0.3s	325
6	Ordertimetable	0	0	307	307	0	0	0	0	Finished	0.3s	1,100
7	Orders_by_Country	0	2823	92	0	0	0	0	0	Finished	0.6s	4,826
8	CC_Order	0	3130	5646	0	0	0	0	0	Finished	0.9s	6,000
9	Sort rows 2	0	2823	2823	0	0	0	0	0	Finished	1.0s	2,754
10	Orders_per_Date	0	2823	252	0	0	0	0	0	Finished	1.0s	2,814
11	Output_Csv_Orders_by_date	0	252	252	0	253	0	0	0	Finished	1.1s	225
12	Output_Csv_Orders_by_country	0	92	92	0	93	0	0	0	Finished	0.6s	154
13	Complete	0	2932	5646	0	0	0	0	0	Finished	1.4s	3,899
14	Product_Prize	0	2823	109	0	0	0	0	0	Finished	1.5s	1,839
15	Output_Csv_product_prize	0	109	109	0	110	0	0	0	Finished	1.6s	70
16	Orders_and_sales_per_Country_Year_Product	0	2823	227	0	0	0	0	0	Finished	1.5s	1,882
17	Output_Csv_Country_Year_Product	0	227	227	0	228	0	0	0	Finished	1.6s	147

Bei der Performance (Abbildung Step Metrics) sieht man, dass vor allem die joins (Center_Customers, CC_Order, Complete), deren Vorsortierung (Sort rows und Sort rows 2) und die CSV-Exporte relativ viel Zeit benötigten. Es wird auch ersichtlich, dass, je weiter rechts ein Prozess in der Prozesskette steht, dieser eher mehr Zeit benötigt.

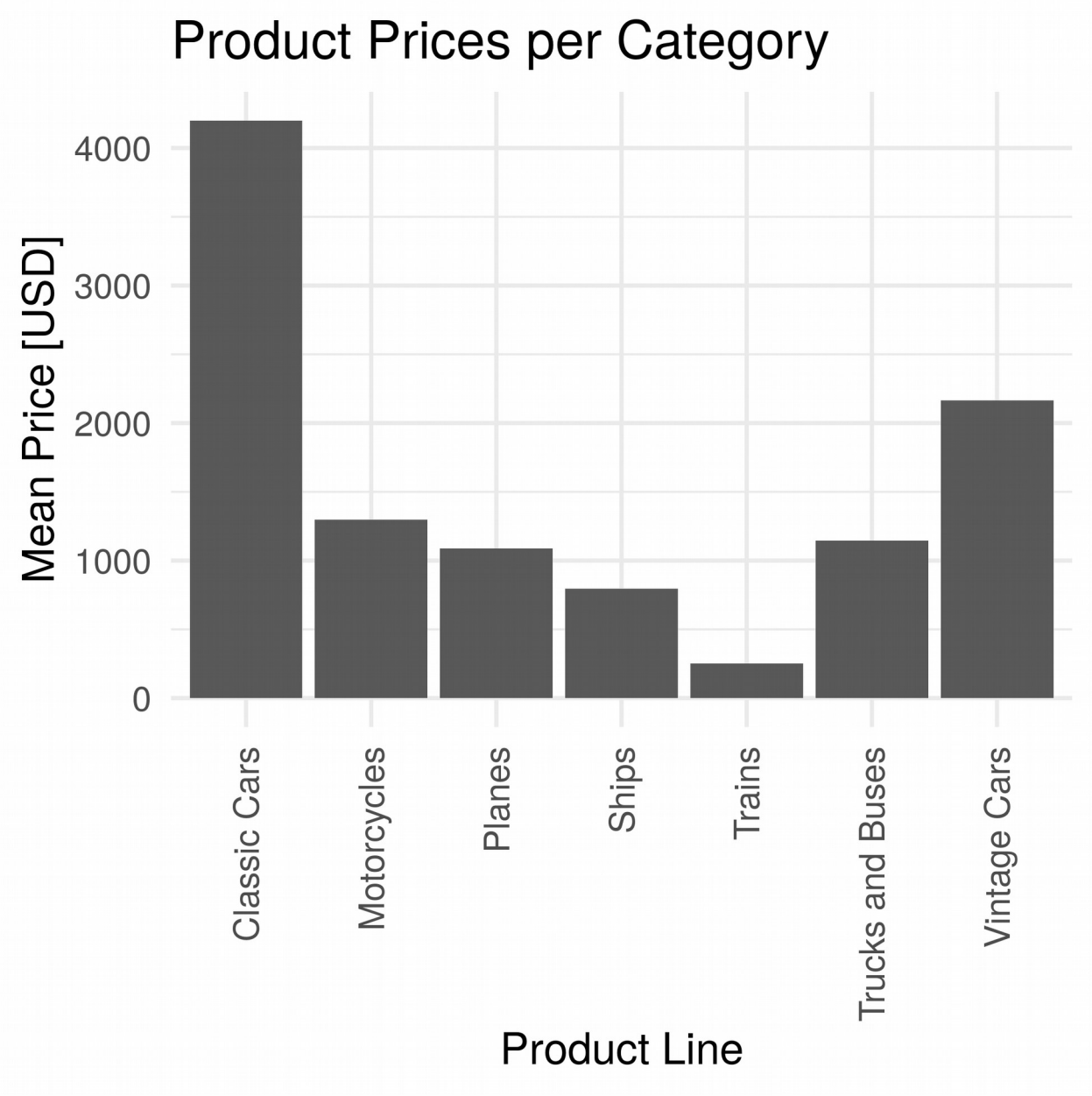
Ergebnisse

Wir haben uns folgende Fragen zum Datenset überlegt:

1. Welche Produkte sind am ertragreichsten?
2. Wie verlaufen die Verkäufe über die Zeit?
3. In welchen Ländern wird am meisten verkauft?
4. Wer sind die wichtigsten Kunden und wo sitzen diese?
5. Welche Produkte werden wann und wo am meisten verkauft?

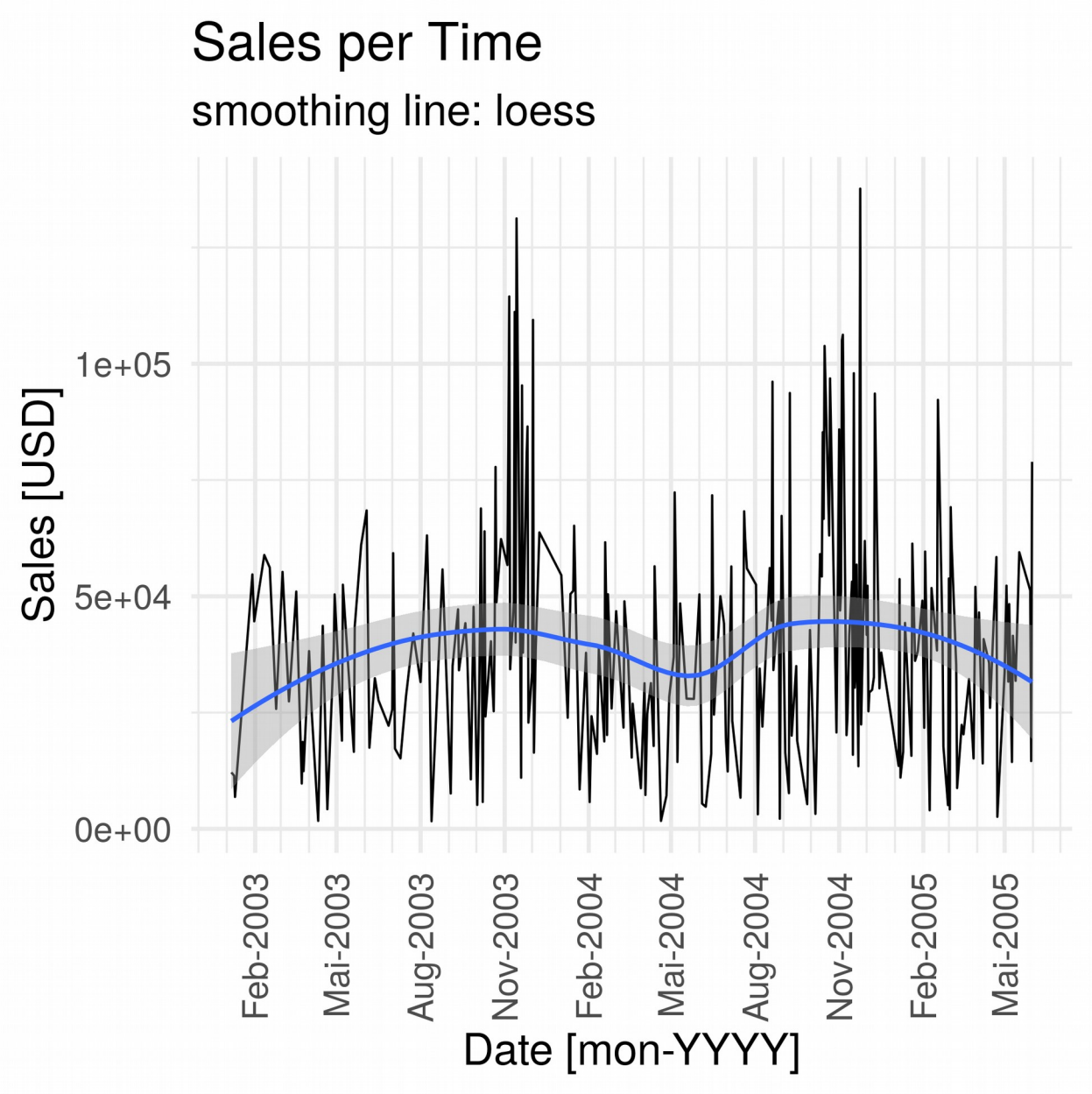
1. Welche Produkte sind am ertragreichsten?

Die höchsten Preise werden für Classic Cars erzielt, gefolgt von Vintage Cars. Natürlich heißt dass nicht, dass diese Produkte auch den höchsten Gewinn erzielen…



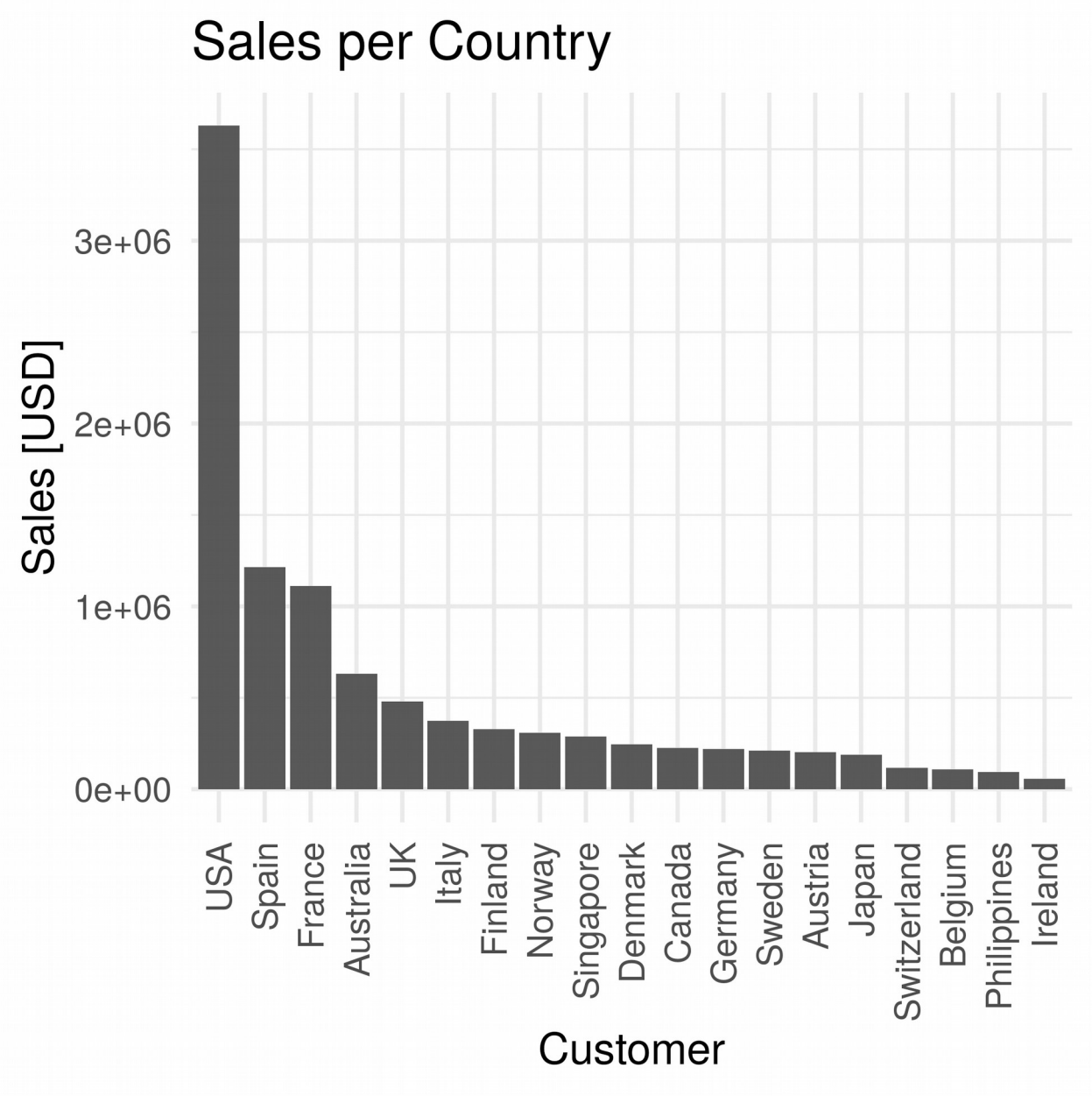
2. Wie verlaufen die Verkäufe über die Zeit?

Offensichtlich gibt es vor Weihnachten große Peaks in den Verkaufszahlen.



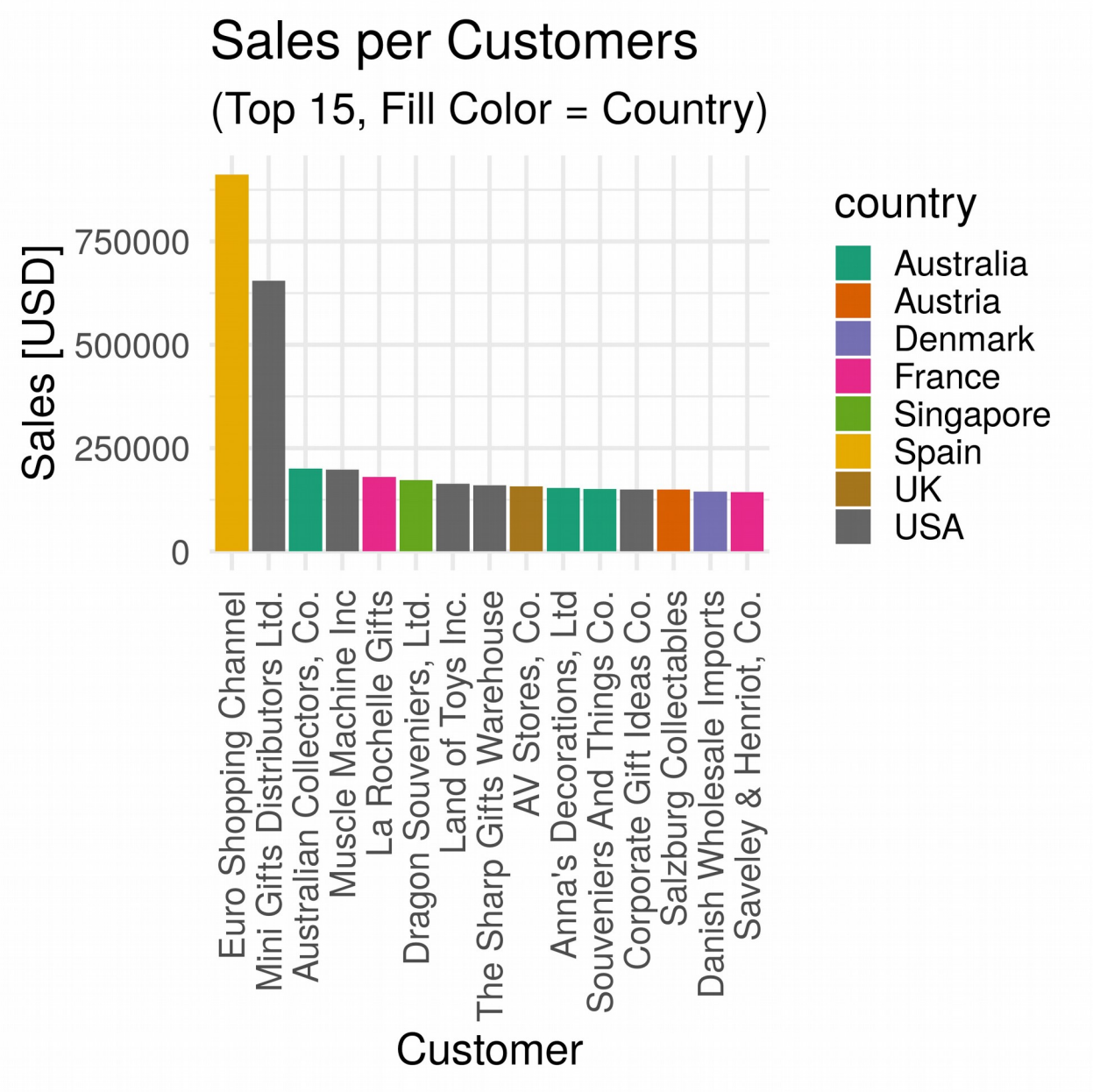
3. In welchen Ländern wird am meisten verkauft?

Offenbar wurde in den USA, gefolgt von Spain und France am meisten abgesetzt.



4. Wer sind die wichtigsten Kunden und wo sitzen diese?

Die Top 3 sind Euro Shopping Channel (Spain), Mini Gifts Distributors Ltd. (USA) und nach einem starken Abfall Australian Collectors, Co. (Australia).



5. Welche Produkte werden wann und wo am meisten verkauft?

Vor allem in den USA wurden viele Classic Cars verkauft, besonders im Jahr 2004. Generell dürften Classic Cars und Vintage Cars am meisten verkauft werden, was vernünftig ist, da diese die höchsten Preise erzielen. 2005 war für alle Länder ein schlechtes Jahr.