

BigData: Anwendungsergebnisse der verschiedenen Anwendungsmöglichkeiten von Hadoop



Vorgelegt von:

Sun Huayi Matrikelnummer: 33338

e-Mail: huayi.juergens@studenten.hs-bremerhaven.de

Daniel Paul Hoffmann Matrikelnummer: 26355

e-Mail: daniel.hoffmann@studenten.hs-bremerhaven.de

Michael Günster Matrikelnummer: 32465

e-mail: michael.guenster@studenten.hs-bremerhaven.de

Bearbeitungszeitraum: 22.03.2018 - 05.07.2018

Modulverantwortliche: Prof. Dr. Nadija Syrjakow

Inhaltsverzeichnis

1	Umfrage: Mahout / Recommender (22.03.2018)	2
2	PIG(12.04.2018)	6
3	Hbase (03.05.2018)	11
4	R Turotium (17.05.2018, 24.05.2018)	17
5	R RJDBC / Access (31.05.2018)	25
6	R / Hive (07.06.2018)	32
7	ARIS (14.06.2018)	38

Abbildungsverzeichnis

1	Tabelle: Umfrage	2
2	Tabelle: Aufarbeitung der Umfrage	3
3	Tabelle: Ergebnis für User1	4
4	Tabelle: Ergebnis für User2	4
5	Tabelle: Ergebnis für User3	5
6	Tabellen	7
7	Tabellen im Datastore	7
8	Tabelle: companies2	13
9	Tabelle: products2	14
10	Ausgabe aller Tabellen	15
11	Standardpfad festlegen	17
12	Anzeige der Zeilen	18
13	Ausgabe Variable	18
14	Bevölkerung 1966	19
15	Variablenausgabe	20
16	Bevölkerung 2016	21
17	Zeilenausgabe	21
18	Diagramm: Frankreich	22
19	Variablenverwendung	23
20	Bevölkerungsentwicklung DE	23
21	Import MS-Access	26
22	Anzeigen der Zeilen	26
23	Analyse Histogramm	27
24	Weiteres Histogramm	27
25	Abfrage Preis	27
26	Boxplot SQFT per year of construction	28
27	Scatterplot	29
28	Variablenverwendung	30
29	Ausgabe Berechnungsmodell	31
30	Ausgabe Tabellen	33
31	Ausgabe: model	35
32	Ausgabe: model	35
33	Ausgabe: autos	35

34	Ausgabe: hotels	36
35	Ausgabe: hotels	36
36	Ausgabe: hotels	36

1 Umfrage: Mahout / Recommender (22.03.2018)

- Fragen Sie Ihre Kommilitonen nach Filmen, Getränken, Büchern, Musikvorlieben, etc.
- Lassen Sie obiges bewerten (1 bis 5 Sterne).
- Analysieren Sie die Umfrage mit Mahout Recommender

Kurzdarstellung der Aufgabenstellung

Es soll eine Bewertungsumfrage bzgl. Büchern, Film o.Ä. getätigt werden. Hierbei sollen die Teilnehmer die jeweiligen Titel mit 1 bis 5 Sternen bewerten. Es ist hierbei von kritischer Bedeutung das kein Teilnehmer alle Titel/Objekt der Umfrage bewertet.

Die gesammelten Daten müssen entsprechend in Form einer .csv Datei aufbereitet werden für das anschließende Importieren in die Software. Nach dem Einlesen werden für einige Umfrageteilnehmer x Empfehlungen mit höchster Gefallenswahrscheinlichkeit abgefragt (abgeleitet aus den Bewertungen anderer Teilnehmer).

Lösung

- Visualisierung der getätigten Umfrage von 10 Teilnehmern zu 17 verschiedenen Filmen (Abbildung 1):

PERSON/FILM	1-JUCELENE	2-DANIEL	3-TANTE	4-SONIA	5-HEIKE	6-DAVID	7-UBERDAN	8-CHRISTOPH	9-VATTERN	10-RAFAEL
1-RAMBO	5	3	5	2	1	4		1	3	5
2-TERMINATOR	4	3		2	1		5		2	
3-TITANIC	3		2	5		2				3
4-FOREST GUMP		4	2		5	5	2	4	5	1
5-BLUE LAGOON	2		1	3	2		1	2		1
6-MISSION IMPOSSIBLE	5	4			1	3		1	2	4
7-MUMMIE	4	1	5	5		2	5		1	
8-BATMAN		3	4	5	2		3	1	1	
9-SPIDERMAN	2		5		2	3		1	1	
10-FAST&FURIOUS	3	4		4			5		2	5
11-ME BEFORE YOU	5		1	5	4	4	1	3		1
12-DER PATE		5	3		3	5	4			2
13-ET	1	1		4		2		2	2	
14-LORD OF THE RINGS	2	5	1	5	4		4	4	4	4
15-JURASSIC PARK		2	5		1	3	4	1	1	5
16-RUSHHOUR	3	5	5	1		5			3	
17-KEVIN ALLEIN ZU HAUSE	2		1		2	2	1	2		3

Abbildung 1: Tabelle: Umfrage

- Aufarbeitung Umfrage mit Excel für vorgegebenes Format: userID, itemId und prefValue (Abbildung 2)

	A	B	C	D
1	FILM	PERSON	WERTUNG	EXPORT
2	1	1	5	1,1,5.0
3	2	1	4	1,2,4.0
4	3	1	3	1,3,3.0
5	5	1	2	1,5,2.0
6	6	1	5	1,6,5.0
7	7	1	4	1,7,4.0
8	9	1	2	1,9,2.0
9	10	1	3	1,10,3.0

Abbildung 2: Tabelle: Aufarbeitung der Umfrage

- Starten Oracle 4.9 VM
 - Kopieren der Daten aus Spalte D in eine Datei mit dem Namen „dataset.csv“ im Ordner /usr/temp/
 - Oracle Jdeveloper starten
 - Neues Java project angelegt: „myproject“ (gleicher Pfad Projekt)
 - Bibliotheken aus den folgenden Ordnern hinzugefügt: usr/lib/hadoop (all Libraries), usr/lib/hadoop/lib (all Libraries), usr/lib/Mahout/ (all Libraries), usr/lib/Mahout/lib (all Libraries)
 - Inhalt der recommender.txt Datei vom ELLI in das Java Projekt kopieren und den Filedatamodel und für die Empfehlungsabfrage entsprechend die gewünschte Userid und Anzahl von Empfehlungen aufführen:
1. User 1 sollen nun 2 Filme empfohlen werden was zum Ergebnis in Abbildung 3 führt:

Es werden mit einer „Wahrscheinlichkeit des Gefallens“ von 3,87/5 bzw. 3,99/5 Userid 1 die Filme 12 („Der Pate“) und 15 („Jurassic Park“) empfohlen. Was augenscheinlich plausibel ist.
 2. User 2 sollen nun 2 Filme empfohlen werden was zum Ergebnis in Abbildung 4 führt:

```

List<RecommendedItem> recommendations = recommender.recommend(1, 2);
for (RecommendedItem recommendation : recommendations) {
    System.out.println(recommendation);
}

} catch (Exception e) {}
}
}

```

Running: myProject.jpr - Log x Build - Issues

```

/u01/jdev/oracle_common/jdk/bin/java -server -classpath /home/oracle/jdeveloper/mywo
log4j:WARN No appenders could be found for logger (org.apache.mahout.cf.taste.impl.m
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
RecommendedItem[item:15, value:3.9944458]
RecommendedItem[item:12, value:3.871978]
Process exited with exit code 0.

```

Abbildung 3: Tabelle: Ergebnis für User1

```

List<RecommendedItem> recommendations = recommender.recommend(2, 2);
for (RecommendedItem recommendation : recommendations) {
    System.out.println(recommendation);
}

} catch (Exception e) {}
}
}

```

Running: myProject.jpr - Log x Build - Issues

```

/u01/jdev/oracle_common/jdk/bin/java -server -classpath /home/oracle/jdeveloper/mywor
log4j:WARN No appenders could be found for logger (org.apache.mahout.cf.taste.impl.mc
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
RecommendedItem[item:11, value:3.7471833]
RecommendedItem[item:5, value:2.0]
Process exited with exit code 0.

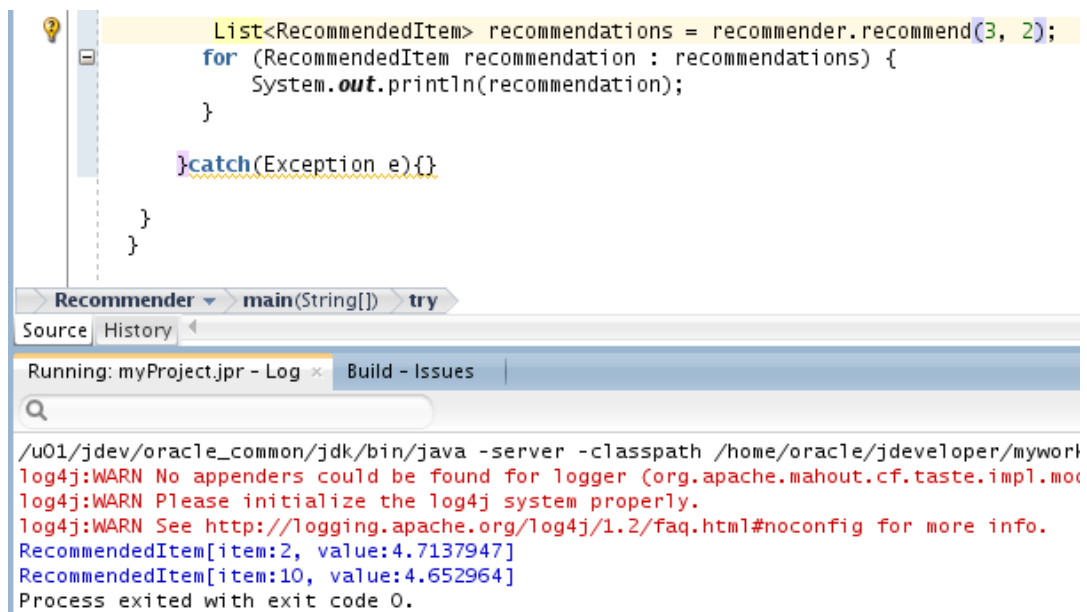
```

Abbildung 4: Tabelle: Ergebnis für User2

Es werden mit einer „Wahrscheinlichkeit des Gefallens“ von 3,75/5 bzw. 2/5 Userid 2 die Filme 11 („Me before You“) und 5 („Blue Lagoon“) empfohlen. Was augenscheinlich plausibel ist.

3. User 3 sollen nun 2 Filme empfohlen werden was zum Ergebnis in Abbildung 5

führt:



The screenshot shows an IDE window with a Java source file. The code defines a `main` method in a class named `Recommender`. It calls `recommender.recommend(3, 2)` to get a list of recommendations, iterates over them, and prints each item. A try-catch block handles potential exceptions. Below the code editor, the 'Run' button is highlighted. The output console shows the execution of the program, including log4j warnings and the printed recommendations: `RecommendedItem[item:2, value:4.7137947]` and `RecommendedItem[item:10, value:4.652964]`. The process exited with code 0.

```
List<RecommendedItem> recommendations = recommender.recommend(3, 2);
for (RecommendedItem recommendation : recommendations) {
    System.out.println(recommendation);
}
} catch (Exception e) {}
}
```

Running: myProject.jpr - Log x Build - Issues

/u01/jdev/oracle_common/jdk/bin/java -server -classpath /home/oracle/jdeveloper/mywork
log4j:WARN No appenders could be found for logger (org.apache.mahout.cf.taste.impl.mo
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
RecommendedItem[item:2, value:4.7137947]
RecommendedItem[item:10, value:4.652964]
Process exited with exit code 0.

Abbildung 5: Tabelle: Ergebnis für User3

Es werden mit einer "Wahrscheinlichkeit des Gefallens" von 4,71/5 bzw. 4,65/5 Userid 2 die Filme 10 ("Me before You") und 5 ("Fast & Furious") empfohlen. Was augenscheinlich plausibel ist.

Aufteilung der Aufgaben im Team

Darstellung der benutzten Werkzeuge und Systeme

Entwurfswerkzeug

Microsoft Excel

Entwicklungsumgebung

JDeveloper

2 PIG(12.04.2018)

Führen Sie folgende Pig Operationen mit Ihren Testdaten (Bags) aus: LOAD, Dump, Join, Cross, Union, Split

Kurzdarstellung der Aufgabenstellung

Datensätze in Form von .csv Dateien in HDFS Ordner Laden und Mittels PIG die im HDFS bereitgestellten Datensätze auswerten. Der grundlegende Umgang mit Daten in PIG als Bestandteil von HADOOP soll durch bereitstellen von Daten in HDFS und das abfragen/analysieren in PIG selbiger geprobt werden.

Lösung

- Starten der Oracle 4.9 VM
- Ändern von Tastaturlayout auf DE mittels Terminaleingabe:

```
setxkbmap -layout de
```

- Anlegen eines eigenen Ordners für die Aufgabenstellung auf der VM:

```
mkdir /usr/tmp/gruppe
```

- Drei Textdateien (artikel, ek_pos_1, ek_pos_2) in /usr/tmp/gruppe/ mit Testdaten anlegen (Abbildung 6):
- Über den Termin einen HDFS Ordner für die Aufgabenstellung anlegen:

```
hdfs dfs -mkdir hdfs://bigdatalite.localdomain:8020/tmp/gruppe
```

- Zuvor angelegte Textdatei in usr/tmp/gruppe in HDFS Ordner (.../tmp/gruppe) kopieren:

```
hdfs dfs -put /usr/tmp/gruppe/ek_pos_1 hdfs://bigdatalite.localdomain:8020/tmp/gruppe
hdfs dfs -put /usr/tmp/gruppe/ek_pos_2 hdfs://bigdatalite.localdomain:8020/tmp/gruppe
hdfs dfs -put /usr/tmp/gruppe/artikel hdfs://bigdatalite.localdomain:8020/tmp/gruppe
```

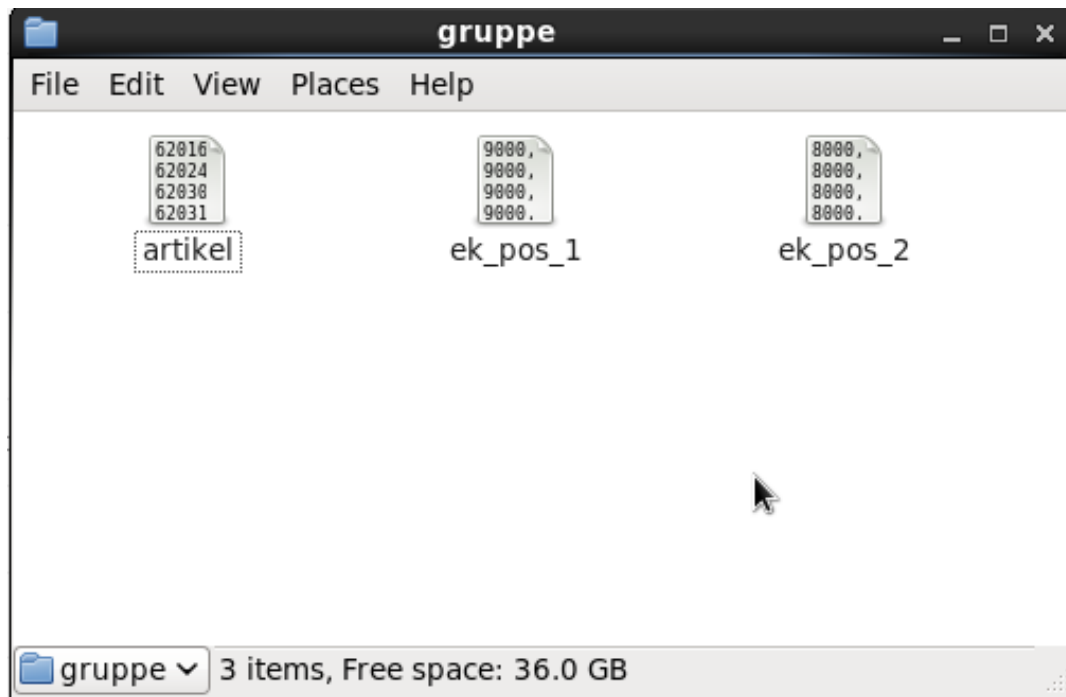


Abbildung 6: Tabellen

- Show files on datastore (printscreen Ergebnis)(Abbildung 7):

```
hdfs dfs -ls /tmp/gruppe
```

```
[oracle@bigdatalite ~]$ hdfs dfs -ls /tmp/gruppe
Found 4 items
-rw-r--r-- 1 oracle supergroup 180 2018-07-02 04:48 /tmp/gruppe/artikel
-rw-r--r-- 1 oracle supergroup 92 2018-07-02 04:48 /tmp/gruppe/ek_pos_1
-rw-r--r-- 1 oracle supergroup 138 2018-07-02 04:48 /tmp/gruppe/ek_pos_2
-rw-r--r-- 1 oracle supergroup 30 2018-07-02 04:27 /tmp/gruppe/test
[oracle@bigdatalite ~]$
```

Abbildung 7: Tabellen im Datastore

- Start Pig

```
pig -x mapreduce
```

- pig schließen:

```
exit the Grunt shell using 'ctrl + d'
```

- Daten aus HDFS Ordner in PIG laden, unter angabe der Spaltennamen und Datentypen:

```

artikel = LOAD 'hdfs://bigdatalite.localdomain:8020/tmp/gruppe/
    artikel' USING
PigStorage(',') as (art_id:int, art_name:chararray);

ek_pos_1 = LOAD 'hdfs://bigdatalite.localdomain:8020/tmp/gruppe/
    ek_pos_1' USING PigStorage(',') as (kd_id:int, pos:int,
    art_id:int, menge:int);

ek_pos_2 = LOAD 'hdfs://bigdatalite.localdomain:8020/tmp/gruppe/
    ek_pos_2' USING PigStorage(',') as (kd_id:int, pos:int,
    art_id:int, menge:int);

```

- Ausführen von Abfragen auf den und deren Ausgaben:

UNION

```

ek_positionen = UNION ek_pos_1, ek_pos_2;
dump ek_positionen;
--(8000,1,6202443,2)
--(8000,2,6203175,1)
--(8000,3,6203174,4)
--(8000,4,6203173,2)
--(8000,5,6205923,2)
--(8000,6,6201680,2)
--(9000,1,6203338,2)
--(9000,2,6203207,1)
--(9000,3,6204711,2)
--(9000,4,6203040,2)

```

JOIN

```

ek_art_join = JOIN ek_positionen by art_id, artikel by art_id;
dump ek_art_join;
--(8000,6,6201680,2,6201680,ARTIKEL_1)
--(8000,1,6202443,2,6202443,ARTIKEL_2)
--(9000,4,6203040,2,6203040,ARTIKEL_3)
--(8000,4,6203173,2,6203173,ARTIKEL_4)
--(8000,3,6203174,4,6203174,ARTIKEL_5)
--(8000,2,6203175,1,6203175,ARTIKEL_6)

```

```
--(9000,2,6203207,1,6203207,ARTIKEL_7)
--(9000,1,6203338,2,6203338,ARTIKEL_8)
--(9000,3,6204711,2,6204711,ARTIKEL_9)
--(8000,5,6205923,2,6205923,ARTIKEL_0)
```

SPLIT

```
SPLIT ek_pos_1 into einkauf_1_1 if pos<3, einkauf_1_2 if pos>2;
dump einkauf_1_1;
--(9000,1,6203338,2)
--(9000,2,6203207,1)

dump einkauf_1_2; (Ausgabe fehlt)
```

CROSS

```
ek_pos_cross = CROSS ek_pos_1, ek_pos_2;
dump ek_pos_cross;
--(9000,4,6203040,2,8000,6,6201680,2)
--(9000,4,6203040,2,8000,5,6205923,2)
--(9000,4,6203040,2,8000,4,6203173,2)
--(9000,4,6203040,2,8000,3,6203174,4)
--(9000,4,6203040,2,8000,2,6203175,1)
--(9000,4,6203040,2,8000,1,6202443,2)
--(9000,3,6204711,2,8000,6,6201680,2)
--(9000,3,6204711,2,8000,5,6205923,2)
--(9000,3,6204711,2,8000,4,6203173,2)
--(9000,3,6204711,2,8000,3,6203174,4)
--(9000,3,6204711,2,8000,2,6203175,1)
--(9000,3,6204711,2,8000,1,6202443,2)
--(9000,2,6203207,1,8000,6,6201680,2)
--(9000,2,6203207,1,8000,5,6205923,2)
--(9000,2,6203207,1,8000,4,6203173,2)
--(9000,2,6203207,1,8000,3,6203174,4)
--(9000,2,6203207,1,8000,2,6203175,1)
--(9000,2,6203207,1,8000,1,6202443,2)
--(9000,1,6203338,2,8000,6,6201680,2)
--(9000,1,6203338,2,8000,5,6205923,2)
--(9000,1,6203338,2,8000,4,6203173,2)
--(9000,1,6203338,2,8000,3,6203174,4)
```

<ul style="list-style-type: none">– – (9000,1,6203338,2,8000,2,6203175,1)– – (9000,1,6203338,2,8000,1,6202443,2)

Aufteilung der Aufgaben im Team

Alle Punkte wurden gemeinsam bearbeitet.

Darstellung der benutzten Werkzeuge und Systeme

Entwurfswerkzeug

Entwicklungsumgebung

3 Hbase (03.05.2018)

- Erzeugen Sie für Ihr Beispiel 2 Hbase (Shell) Tabellen.
- Füllen Sie die Tabellen mit einigen Testdaten.
- (Java API) Erzeugen Sie für Ihr Beispiel zwei Hbase Tabellen.
- Füllen Sie die Tabellen mit einigen Testdaten.
- Listen Sie die Tabellen auf.

Kurzdarstellung der Aufgabenstellung

Zum Einstieg in NoSQL Datenbanken sollen in Apache Hbase je zwei Tabellen mittels Shell Interface (a) und java SCRIPT/CODE? (b) erstellt und mit Daten befüllt werden. Weiterhin sollen die angelegten Tabellen in Hbase ausgegeben werden mittels vorgegebenen java SCRIPT/CODE (c)?

Lösung

- Oracle VM 4.9 starten
- Tastatur auf Deutsch stellen mittels Eingabe Befehl in Terminal:

```
setxkbmap -layout de
```

HBASE

- HBASE über folgenden Befehl im Terminal der VM starten:

```
hbase shell
```

- Anlegen der Tabelle „companies“ mit den attributen „id“ und „property“

```
create 'companies1', 'id', 'property'
```

- Daten in die zuvor angelegte Tabelle „companies1“ schreiben:

```
put 'companies1', 'row1', 'id:HRNR', '162'
put 'companies1', 'row1', 'property:Name', 'Musterfirma'
put 'companies1', 'row1', 'property:Gruendung', '1979'
```

- Tabelle „companies1“ mit geschriebenen Daten ausgeben lassen:

```
get 'companies1', 'row1'

COLUMN                                CELL
id:HRNR                               timestamp=1529566648268, value=162
property:Gruendung                    timestamp=1529566667304, value=1979
property:Name                          timestamp=1529566659389, value=Musterfirma
3 row(s) in 0.0400 seconds
```

- Anlegen der Tabelle „products1“ mit den attributen „id“ und „property“

```
create 'products1', 'id', 'property'
```

- Daten in die zuvor angelegte Tabelle „products1“ schreiben:

```
put 'products1', 'row1', 'id:ISBN10', '3866471769'
put 'products1', 'row1', 'id:ISBN13', '978-3866471764'
put 'products1', 'row1', 'property:Typ', 'Buch'
put 'products1', 'row1', 'property:Titel', 'Krieg und Frieden'
put 'products1', 'row1', 'property:Seitenanzahl', '1536'
```

- Tabelle „products1“ mit geschriebenen Daten ausgeben lassen:

```
get 'products1', 'row1'

COLUMN                                CELL
id:ISBN10                             timestamp=1529566771386, value=3866471769
id:ISBN13                             timestamp=1529566777916, value=978-3866471764
property:Seitenanzahl                 timestamp=1529566805924, value=1536
property:Titel                       timestamp=1529566797799, value=Krieg und Frieden
property:Typ                         timestamp=1529566790781, value=Buch
5 row(s) in 0.0220 seconds
```

Mittels Java-API

In der Oracle VM verifizieren ob HBASE als Service gestartet ist und ggf. starten.

- Oracle JDeveloper 12c starten und Java Developer auswählen
- Java Projekt „A0305“ mit gewünschtem Zielpfad anlegen „/usr/tmp/dph/A0305/“
- Alle Java Klassen zu A0305 durch Rechtsklick auf Projektnamen hinzufügen aus:
Location : usr/lib/hadoop (all Library) Location : usr/lib/hadoop/lib (all Library)
Location : usr/lib/hbase (all Library) Location : usr/lib/hbase/lib (all Library)
- Übernahme und Ausführung der SCRIPTE aus den Materialien (Korrektur: Table-Descriptor in HtableDescriptor)
- Anlegen Tabelle „companies2“ mit Spalten „id“ und „property“ (Abbildung 8)

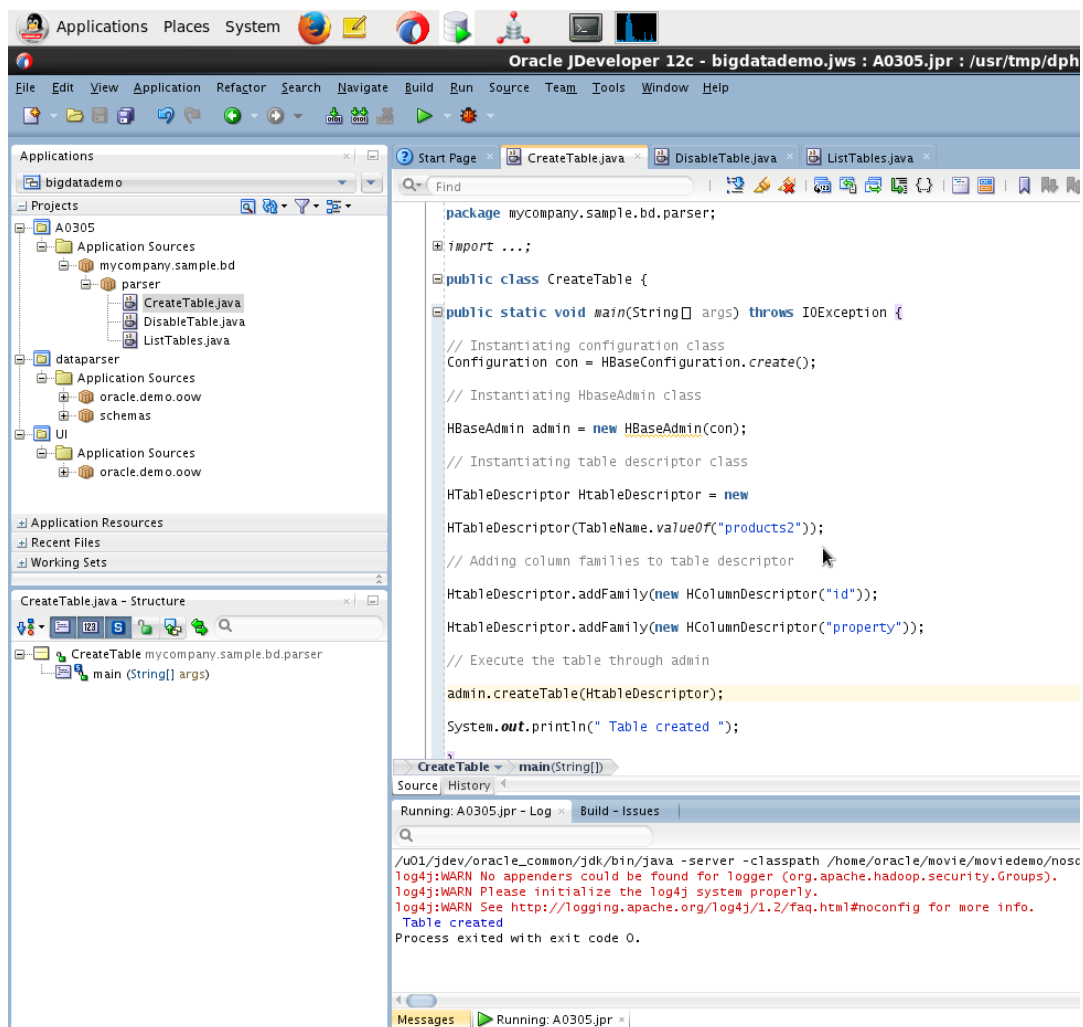


Abbildung 8: Tabelle: companies2

- Anlegen Tabelle „products2“ mit Spalten „id“ und „property“ (Abbildung 9)

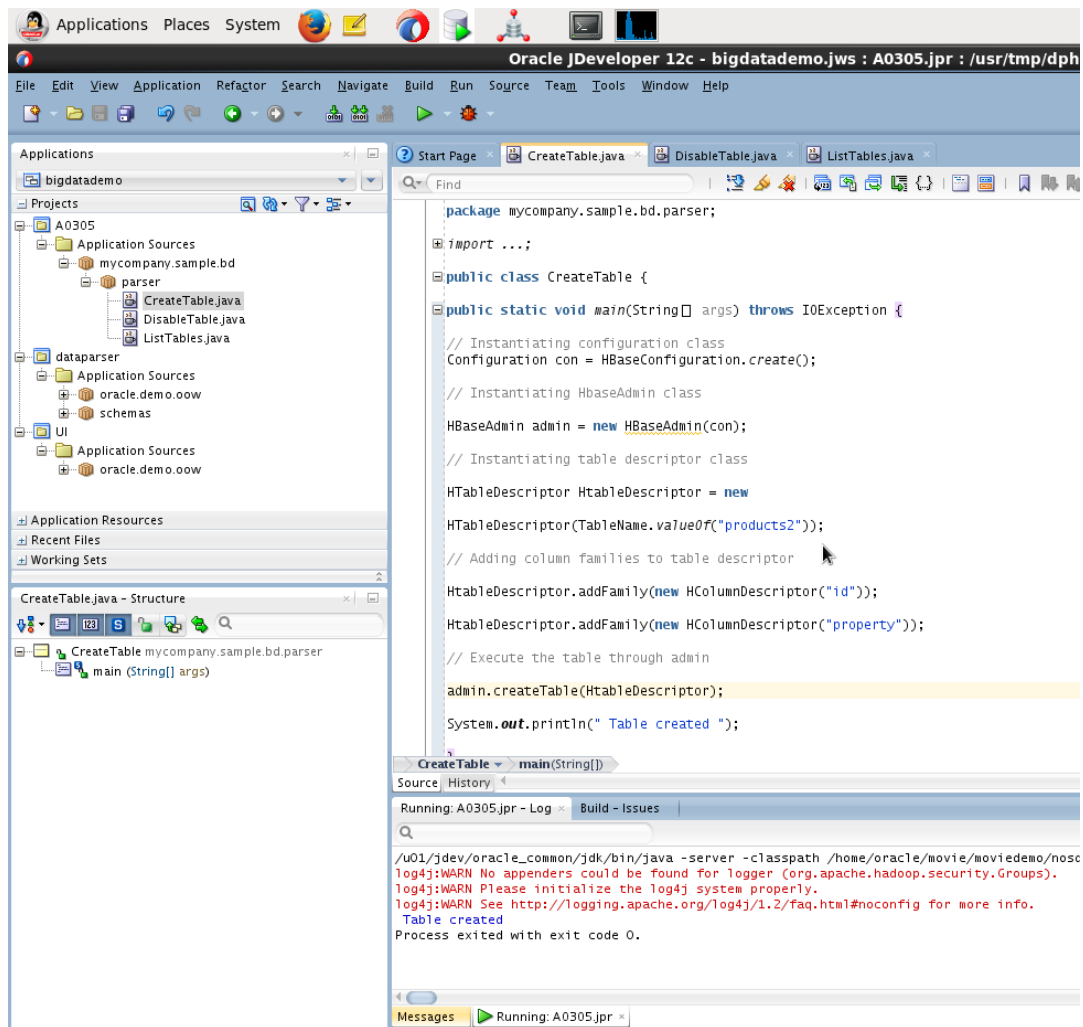


Abbildung 9: Tabelle: products2

- Ausgabe aller angelegten Tabellen in HBASE mittels vorgebenem JAVA code(Abbildung 10):

- Hbase ueber den Terminal der Oracle VM starten:

```
hbase shell
```

- Daten in die zuvor angelegte Tabelle „companies2“ schreiben:

```
put 'companies2', 'row1', 'id:HRNR', '976'
put 'companies2', 'row1', 'property:Name', 'TheRealDeal'
put 'companies2', 'row1', 'property:Gruender', 'Mr. X'
put 'companies2', 'row1', 'property:Employees', '35'
```

- Tabelle „companies2“ mit geschriebenen Daten ausgeben lassen:

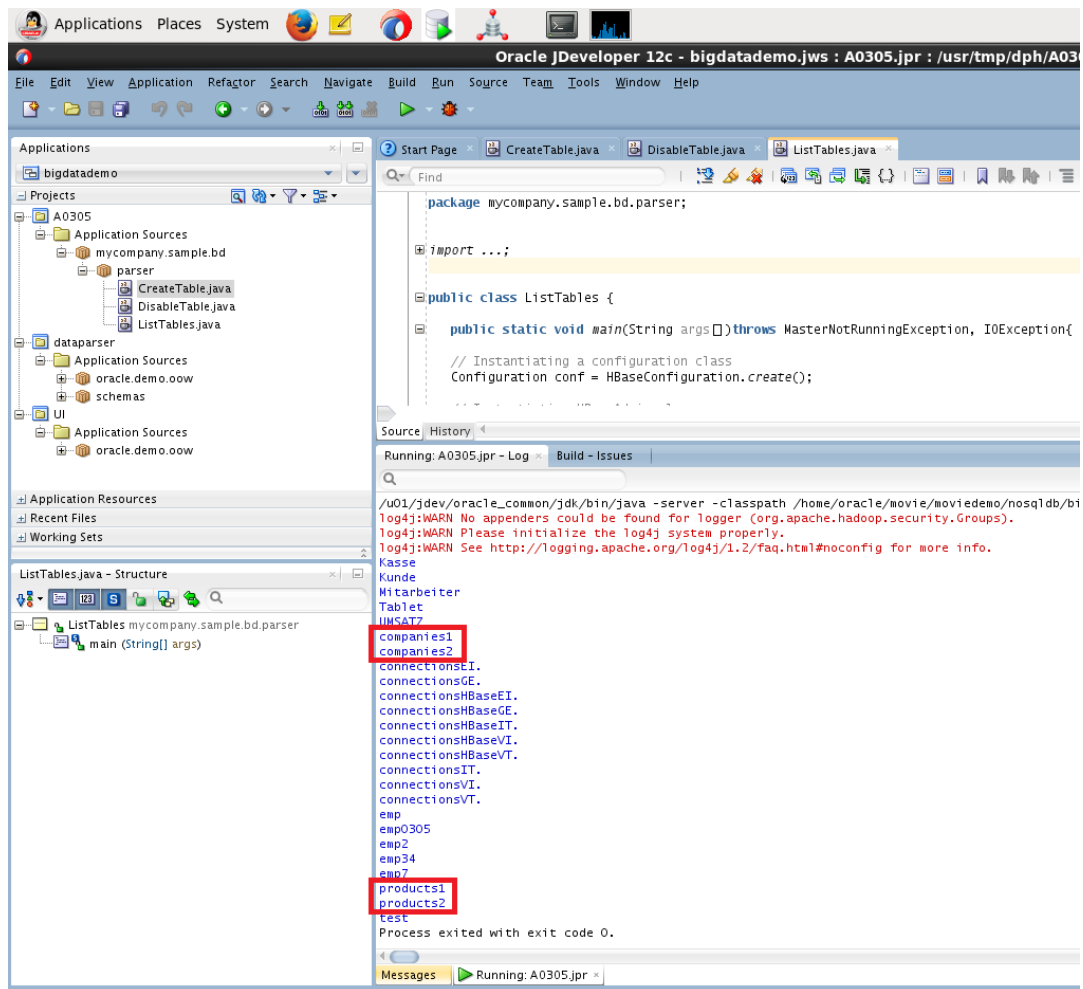


Abbildung 10: Ausgabe aller Tabellen

```
get 'companies2', 'row1'
COLUMN          CELL
id:HRNR         timestamp=1529567187758, value=976
property:Employees timestamp=1529567210676, value=35
property:Gruender timestamp=1529567203411, value=Mr. X
property:Name    timestamp=1529567196336, value=TheRealDeal
4 row(s) in 0.0290 seconds
```

- Daten in die zuvor angelegte Tabelle „products2“ schreiben:

```
put 'products2', 'row1', 'id:ASIN', 'B015XI6BWE'
put 'products2', 'row1', 'id:Herstellernr', '20B7S1C600'
put 'products2', 'row1', 'id:EAN', '4260444483708'
put 'products2', 'row1', 'property:Typ', 'Laptop'
put 'products2', 'row1', 'property:Model', 'T440'
```

```
put 'products2', 'row1', 'property:Marke', 'Think Pad'
0 row(s) in 0.0140 seconds
```

- Tabelle „companies1“ mit geschriebenen Daten ausgeben lassen:

```
get 'products2', 'row1'
COLUMN          CELL
id:ASIN          timestamp=1529567272010, value=B015XI6BWE
id:EAN           timestamp=1529567286841, value=4260444483708
id:Herstellernr  timestamp=1529567279612, value=20B7S1C600
property:Marke   timestamp=1529567306209, value=Think Pad
property:Model   timestamp=1529567299462, value=T440
property:Typ      timestamp=1529567293694, value=Laptop
6 row(s) in 0.0380 seconds
```

Aufteilung der Aufgaben im Team

Alle Punkte wurden gemeinsam bearbeitet.

Darstellung der benutzten Werkzeuge und Systeme

Entwurfswerkzeug

Jdeveloper/HBASE

Entwicklungsumgebung

HBASE

4 R Tutorium (17.05.2018, 24.05.2018)

- Führen Sie die im Tutorium: https://www.tutorialspoint.com/r/r_web_data.htm angegebenen Unterpunkte zu R Data Interfaces, R Charts & Graphs und R Statistics Examples aus.
- Wenden Sie anschließend die Beispiele aus den Tutorien auf Ihre eigenen Daten an

Kurzdarstellung der Aufgabenstellung

Es soll die grundlegende Funktionsweise von R verstanden werden durch die Anwendung öffentlich zugänglicher Tutorials.

Lösung

Als Dataset wurde die Bevölkerungshistorie der einzelnen EU Länder nach Jahren zwischen 1960 und 2016 gewählt.

- Zuerst wird R-Studio gestartet
- Mittels folgendem Befehl ermittelt wir das working directory auf die R-Installation eingestellt ist (Abbildung 11):

```
> getwd()  
[1] "C:/Users/admin/Documents"
```

Abbildung 11: Standardpfad festlegen

- Die .csv Datei („BEVOELKERUNG.csv“) wird in den zuvor ermittelten Ordner gelegt
- Über den folgenden Befehl werden die Daten aus „C:/Users/admin/Documents/BEVOELKERUNGSDATEN.csv“ in eine R variable geladen:

```
data <- read.csv("BEVOELKERUNG.csv")
```

- Über diesen Befehl werden die ersten 6 Zeilen einer Variable ausgegeben werden (Abbildung 12):

```
> head(data)
  Land Jahr Bevoelkerung
1   AT 1960      7047539
2   AT 1961      7086299
3   AT 1962      7129864
4   AT 1963      7175811
5   AT 1964      7223801
6   AT 1965      7270889
```

Abbildung 12: Anzeige der Zeilen

Tortendiagramm

Zum Vergleich der Bevoelkerungsverteilung ueber die Laender der heutigen EU soll je ein Torten-Diagramm für 1996 und 2016 erstellt werden:

1966

- Daten für 1966 muessen zuerst ein eine Variable separiert werden:

```
J1966 <- data[ which(data$Jahr==1966), ]
```

- Ausgabe der ersten 6 Zeilen von J1966(Abbildung 13):

```
> head(J1966)
  Land Jahr Bevoelkerung
7   AT 1966      7322066
64  BE 1966      9527807
121 BG 1966      8258057
178 CY 1966       585308
235 CZ 1966      9821040
292 DE 1966     76600311
```

Abbildung 13: Ausgabe Variable

- Zum Erstellen des Diagramm werden zuerst die betreffenden Spalten in Variablen verwiesen mittels der folgenden beiden Befehle:

```
x <- J1966$Bevoelkerung
labels <- J1966$Land
```

- Dateiname und Typ für die Diagramm-Ausgabe festlegen:

```
png(file = "Bevoelkerung_1966.jpg")
```

- Plotten des Diagramms

```
pie(x, labels, main = "Bevoelkerung 1966")
```

- Diagramm in zuvor angegebene Datei "Bevoelkerung_1966.jpg" speichern:

```
dev.off()
```

- Fertiges Diagramm (Abbildung 14):

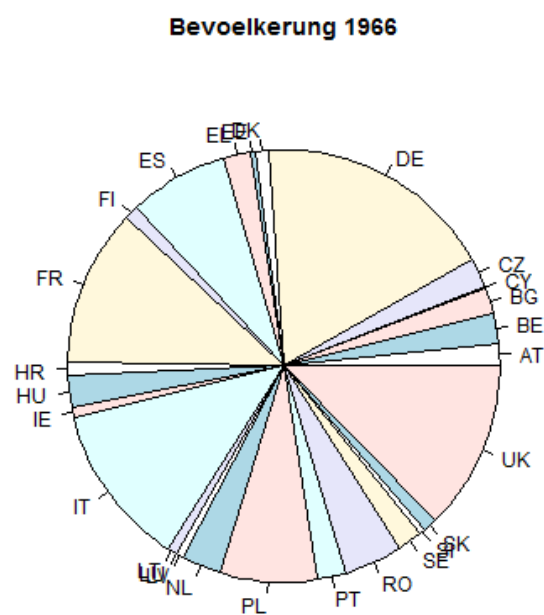


Abbildung 14: Bevölkerung 1966

2016

- Daten für 2016 müssen zuerst in eine Variable separiert werden:

```
J2016 <- data[ which(data$Jahr==2016), ]
```

- Ausgabe der ersten 6 Zeilen von J2016(??):

```
> head(J2016)
      Land Jahr Bevoelkerung
57      AT 2016      8731471
114     BE 2016     11338476
171     BG 2016      7127822
228     CY 2016      1170125
285     CZ 2016     10566332
342     DE 2016     82487842
```

Abbildung 15: Variablenausgabe

- Zum Erstellen des Diagramm werden zuerst die betreffenden Spalten in Variablen verwiesen mittels der folgenden beiden Befehle:

```
x <- J2016$Bevoelkerung
labels <- J2016$Land
```

- Dateiname und Typ für die Diagramm-Ausgabe festlegen:

```
png( file = "Bevoelkerung_2016.jpg ")
```

- Plotten des Diagramms

```
pie(x, labels , main = "Bevoelkerung 2016")
```

- Diagramm in zuvor angebene Datei "Bevoelkerung_2016.jpg" speichern:

```
dev.off()
```

- fertiges Diagramm(Abbildung 16):

Bar-Charts

Frankreich

- Daten für Frankreich muessen zuerst ein eine Variable separiert werden:

```
FR <- data[ which(data$Land=='FR') , ]
```

- Ausgabe der ersten 6 Zeilen von FR(Abbildung 17):

Bevoelkerung 2016

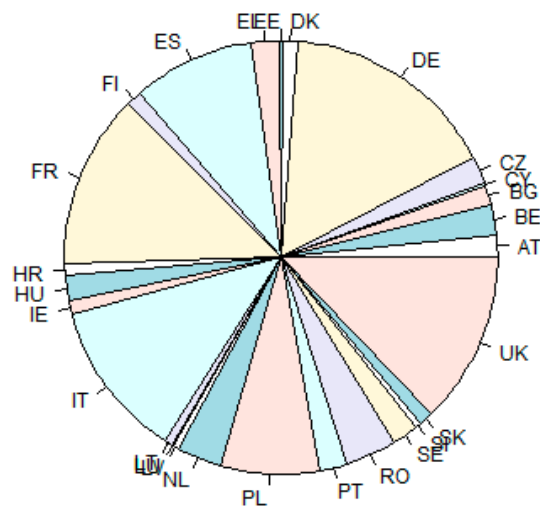


Abbildung 16: Bevölkerung 2016

```
> head(FR)
      Land Jahr Bevoelkerung
628    FR 1960    46814237
629    FR 1961    47444751
630    FR 1962    48119649
631    FR 1963    48803680
632    FR 1964    49449403
633    FR 1965    50023774
```

Abbildung 17: Zeilenausgabe

- Zum Erstellen des Diagramm werden zuerst die betreffenden Spalten in Variablen verwiesen mittels der folgenden beiden Befehle:

```
Y <- FR$Bevoelkerung
X <- FR$Jahr
```

- Dateiname und Typ für die Diagramm-Ausgabe festlegen:

```
png(file = "Bevoelkerunghistorie_Frankreich.png")
```

- Plotten des Diagramms

```
barplot(Y, names.arg=X, xlab="Jahr", ylab="Bevoelkerung", col="blue", main="Bevoelkerungsentwicklung FR")
```


- Diagramm in zuvor angegebene Datei "Bevoelkerung_2016.jpg" speichern:

```
dev.off()
```

- fertiges Diagramm(Abbildung 18):

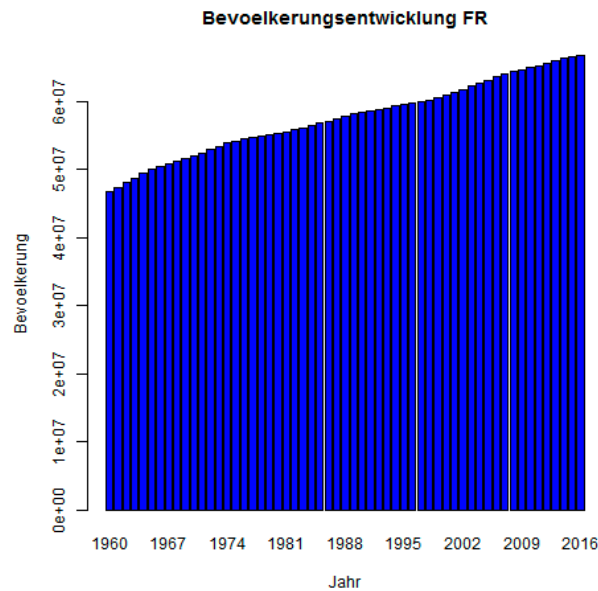


Abbildung 18: Diagramm: Frankreich

Deutschland

- Daten für Deutschland muessen zuerst in eine Variable separiert werden:

```
DE <- data[ which(data$Land=='DE') , ]
```

- Ausgabe der ersten 6 Zeilen von de(Abbildung 19):
- Zum Erstellen des Diagramm werden zuerst die betreffenden Spalten in Variablen verwiesen mittels der folgenden beiden Befehle:

```
Y <- DE$Bevoelkerung  
X <- DE$Jahr
```

- Dateiname und Typ für die Diagramm-Ausgabe festlegen:

```
png(file = "Bevoelkerungshistorie_Deutschland.png")
```

```
> head(DE)
      Land Jahr Bevoelkerung
286    DE 1960    72814900
287    DE 1961    73377632
288    DE 1962    74025784
289    DE 1963    74714353
290    DE 1964    75318337
291    DE 1965    75963695
```

Abbildung 19: Variablenverwendung

- Plotten des Diagramms

```
barplot(Y, names.arg=X, xlab="Jahr", ylab="Bevoelkerung", col="blue", main=
      ="Bevoelkerungsentwicklung DE")
```

- Diagramm in zuvor angebene Datei "Bevoelkerung_2016.jpg" speichern:

```
dev.off()
```

- fertiges Diagramm (Abbildung 20):

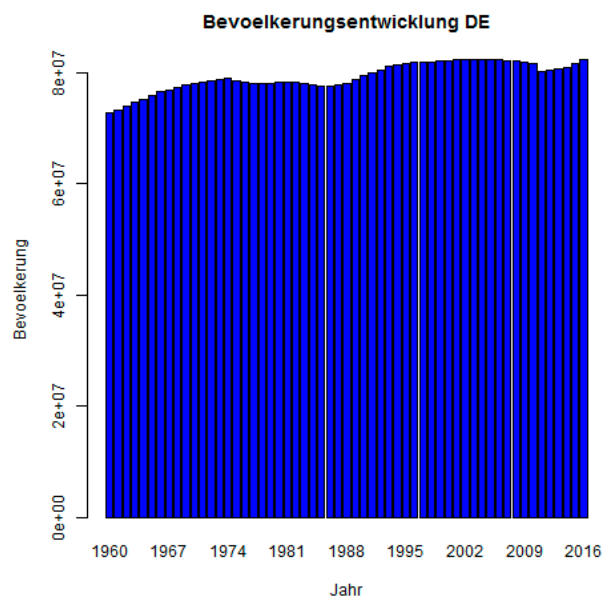


Abbildung 20: Bevölkerungsentwicklung DE

Aufteilung der Aufgaben im Team

Alle Aufgabenpunkte wurden gemeinsam bearbeitet

Darstellung der benutzten Werkzeuge und Systeme

Entwurfswerkzeug

R-Studio

Entwicklungsumgebung

R-Studio

5 R RJDBC / Access (31.05.2018)

- Suchen Sie sich einen für Sie interessanten Bereich aus bspw. Marktforschung, soziale Netzwerke, Wetterstationen, Gesundheitswesen, Natur und Sozialwissenschaften, etc.
- Führen Sie einen Tabellen Entwurf durch.
- Erzeugen Sie die Tabellen (Oracle Datenbank).
- Füllen Sie die Tabellen mit einigen Testdaten
- Bauen Sie eine RJDBC Verbindung (R) zur Oracle Datenbank auf.
- Analysieren Sie die Daten
- Suchen Sie in deren Datenbeständen nach unbekannten Zusammenhängen.

Kurzdarstellung der Aufgabenstellung

Es sollen Daten aus einem beliebigen Bereich gewählt werden und in einer Datenbank abgelegt werden. Auf die Datenbank soll mit R zugegriffen werden um die Daten analysieren zu können.

Lösung

- Gewählter Datensatz: <https://www.kaggle.com/harlfoxem/housesalesprediction> beinhaltet Hausverkaufsdaten von Häusern in King County (USA) im Jahre 2014
- Daten wurden in MS-Access (C:/users/admin/documents/DATA_DB.mdb) mit der CSV-Importfunktion eingelesen (Abbildung 21)

Für die Analyse wurde nur R gestartet und folgende Aktionen/Befehle ausgeführt:

- Installieren RODBC Package:

```
install.packages("RODBC")
```

- Aufrufen Package RODBC:



Abbildung 21: Import MS-Access

```
library(RODBC)
```

- Zuweisen Genutzten Datenbank auf (abgelegt in R source Ordner)

```
channel <- odbcConnectAccess("DATA_DB.mdb")
```

- Zeigen der Datentabelle auf Objekt:

```
data = sqlQuery(channel, paste('SELECT * FROM HOUSE_PRICE_SQRT'))
```

- Anzeigen der ersten sechs Zeilen der Tabelle/Objekt (Abbildung 22)

```
> head(data)
  id          date    price bedrooms bathrooms sqft_living sqft_lot floors waterfront view condition
1 "7129300520" "20141013T000000" 221900      3      1.00      1180      5650    "1"         0      0      3
2 "6414100192" "20141209T000000" 538000      3      2.25      2570      7242    "2"         0      0      3
3 "5631500400" "20150225T000000" 180000      2      1.00      770      10000    "1"         0      0      3
4 "2487200875" "20141209T000000" 604000      4      3.00      1960      5000    "1"         0      0      5
5 "1954400510" "20150218T000000" 510000      3      2.00      1680      8080    "1"         0      0      3
6 "7237550310" "20140512T000000" 1225000     4      4.50      5420     101930    "1"         0      0      3
  grade sqft_above sqft_basement yr_built yr_renovated zipcode    lat    long sqft_living15 sqft_lot15
1      7      1180           0    1955      0 "98178" 47.5112 -122.257      1340      5650
2      7      2170          400    1951     1991 "98125" 47.7210 -122.319      1690      7639
3      6       770           0    1933      0 "98028" 47.7379 -122.233      2720      8062
4      7      1050          910    1965      0 "98136" 47.5208 -122.393      1360      5000
5      8      1680           0    1987      0 "98074" 47.6168 -122.045      1800      7503
6     11      3890         1530    2001      0 "98053" 47.6561 -122.005      4760     101930
```

Abbildung 22: Anzeigen der Zeilen

- Analyse Histogramm Preis (Abbildung 23):

```
hist(data$price, xlab = "price", col = "green", border = "red")
```

- Datensatz auf Preise bis maximal 1mio reduzieren:

```
data = sqlQuery(channel, paste('SELECT * FROM HOUSE_PRICE_SQRT WHERE PRICE < 1000001'))
```

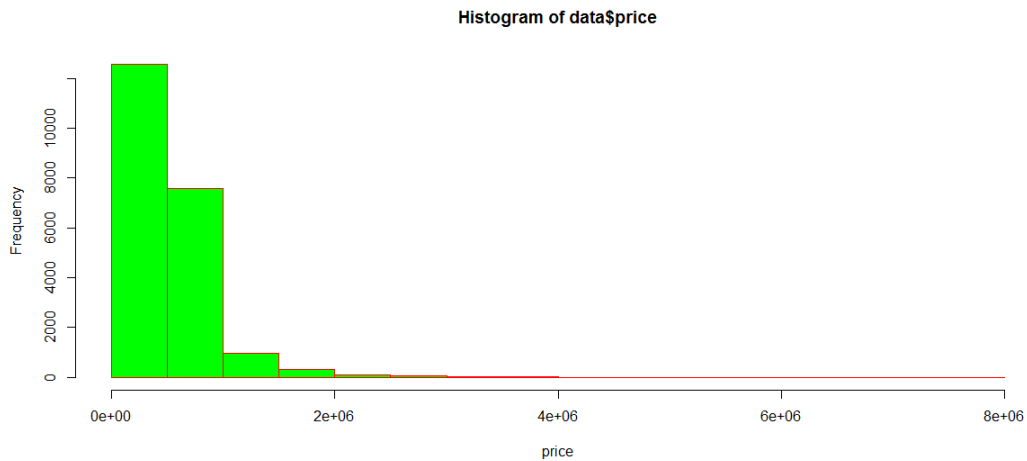


Abbildung 23: Analyse Histogramm

- Neues Histogramm ausführen(Abbildung 24):

```
hist(data$price, xlab = "price", col = "green", border = "red")
```

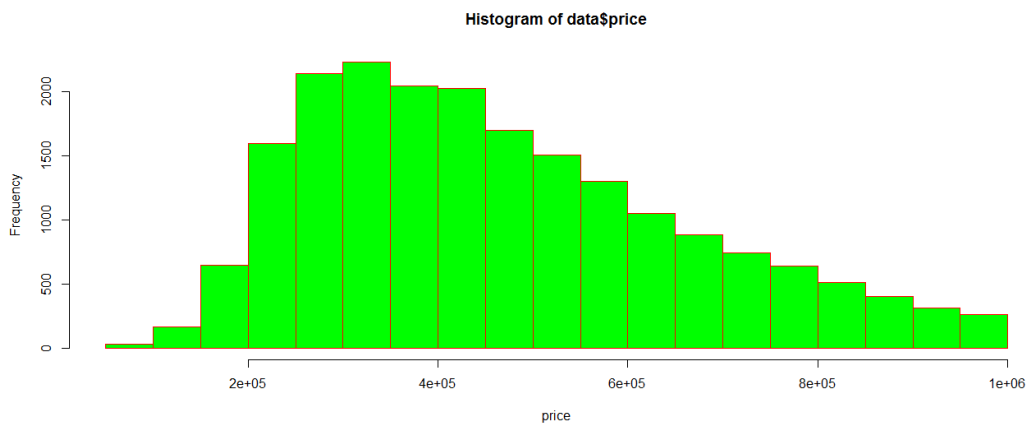


Abbildung 24: Weiteres Histogramm

- Abfrage minimum Preis in der Datei(Abbildung 25):

```
> print(min(data$price))
[1] 75000
```

Abbildung 25: Abfrage Preis

- Boxplot (Wohnraufgrößen Variation pro Baujahr)(Abbildung 26):

```
boxplot(sqft_living ~ yr_built, data = data, xlab = "YEAR BUILT",+
        ylab = "SQFT_LIVINV", main = "SQFT PER YEAR OF CONSTRUCTION")
```

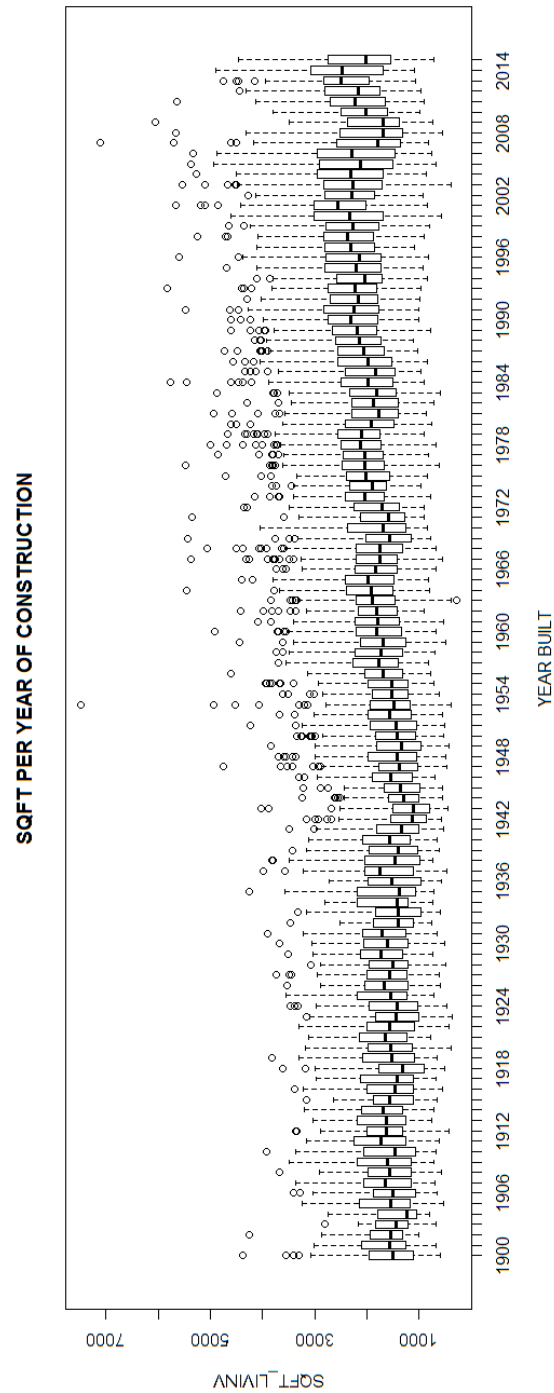


Abbildung 26: Boxplot SQFT per year of construction

- Erstellen von SCATTER PLOT MATRIX(Abbildung 27):

```
pairs(~price+sqft_living+sqft_lot ,data = data , + main = "
Scatterplot Matrix")
```

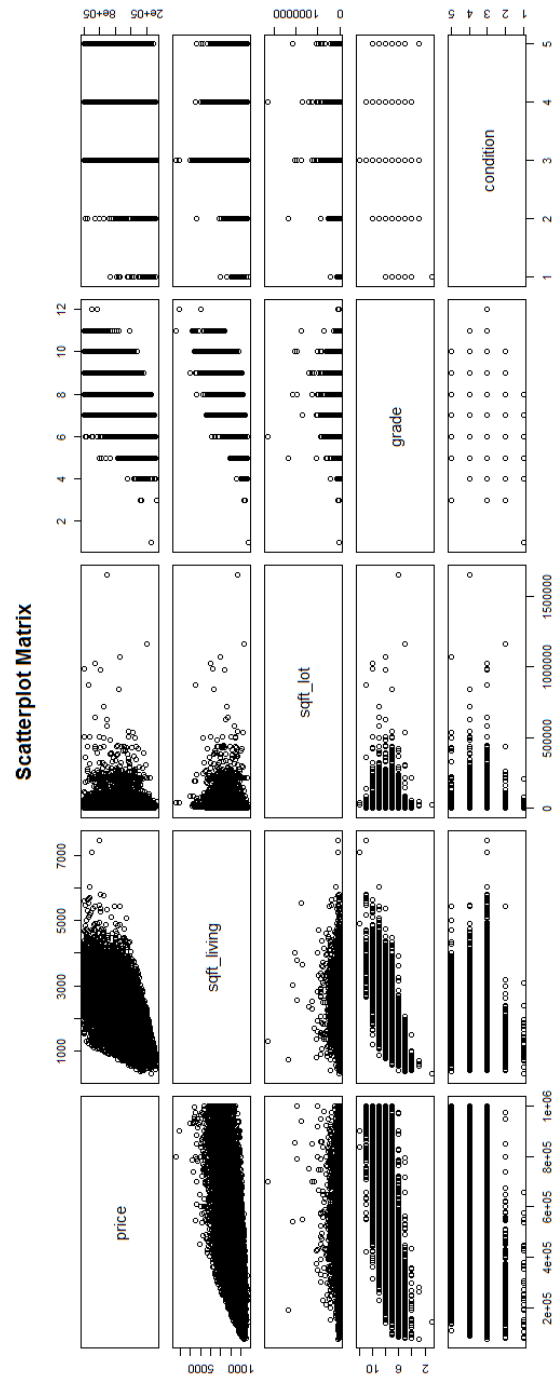


Abbildung 27: Scatterplot

- multiple lineare Regression (Price in Abhängigkeit Sqft_Living und Sqft_Loft)
- Installation und Öffnung der relevanten Daten

```
install.packages("caret")
install.packages("ggplot2")
install.packages("lattice")
library(caret)
```

- Verwendung der Daten in der variable „data“ (Abbildung 28)

```
> head(data)
  id            date  price bedrooms bathrooms sqft_living sqft_lot floors waterfront view condition
1 "7129300520" "20141013T000000" 221900      3         1.00      1180     5650    "1"          0      0          3
2 "6414100192" "20141209T000000" 538000      3         2.25      2570     7242    "2"          0      0          3
3 "5631500400" "20150225T000000" 180000      2         1.00       770    10000    "1"          0      0          3
4 "2487200875" "20141209T000000" 604000      4         3.00      1960     5000    "1"          0      0          5
5 "1954400510" "20150218T000000" 510000      3         2.00      1680     8080    "1"          0      0          3
6 "1321400060" "20140627T000000" 257500      3         2.25      1715     6819    "2"          0      0          3
  grade sqft_above sqft_basement yr_built yr_renovated zipcode    lat    long sqft_living15 sqft_lot15
1     7         1180           0    1955           0 "98178" 47.5112 -122.257      1340      5650
2     7         2170          400    1951          1991 "98125" 47.7210 -122.319      1690      7639
3     6          770           0    1933           0 "98028" 47.7379 -122.233      2720      8062
4     7         1050          910    1965           0 "98136" 47.5208 -122.393      1360      5000
5     8         1680           0    1987           0 "98074" 47.6168 -122.045      1800      7503
6     7         1715           0    1995           0 "98003" 47.3097 -122.327      2238      6819
```

Abbildung 28: Variablenverwendung

- Verwendung Modell für multiple lineare Regression mit allen mathematisch vorstellbar relevanten spalten

```
model <-lm( price~bedrooms+bathrooms+sqft_living+sqft_lot+waterfront+
  view+condition+grade+sqft_above+sqft_basement+yr_built+
  yr_renovated ,      data = data)
```

- Ausgabe des berechneten Modells (Abbildung 29):

Aufteilung der Aufgaben im Team

Alle Aufgabenpunkte wurden gemeinsam bearbeitet

Darstellung der benutzten Werkzeuge und Systeme

Entwurfswerkzeug

MS-ACCESS

```

> print(summary(model))

Call:
lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
    waterfront + view + condition + grade + sqft_above + sqft_basement +
    yr_built + yr_renovated, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-652840  -93491   -5368    83282   672474

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.184e+06  8.995e+04  46.516 < 2e-16 ***
bedrooms     -1.417e+04  1.321e+03 -10.727 < 2e-16 ***
bathrooms     3.721e+04  2.205e+03  16.876 < 2e-16 ***
sqft_living   7.542e+01  3.002e+00  25.121 < 2e-16 ***
sqft_lot      3.698e-03  2.397e-02   0.154  0.8774
waterfront    8.152e+04  1.826e+04   4.463 8.11e-06 ***
view          2.128e+04  1.644e+03  12.940 < 2e-16 ***
condition     1.264e+04  1.586e+03   7.967 1.71e-15 ***
grade         9.984e+04  1.437e+03  69.497 < 2e-16 ***
sqft_above    5.935e+00  2.759e+00   2.151  0.0315 *
sqft_basement      NA         NA      NA      NA
yr_built      -2.384e+03  4.629e+01 -51.509 < 2e-16 ***
yr_renovated   1.039e+00  2.647e+00   0.392  0.6948
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 133000 on 20136 degrees of freedom
Multiple R-squared:  0.5412,    Adjusted R-squared:  0.541
F-statistic: 2160 on 11 and 20136 DF,  p-value: < 2.2e-16

```

Abbildung 29: Ausgabe Berechnungsmodell

Entwicklungsumgebung

R (32bit Version da MS-ACCESS 32bit Version ansonsten Verbindung in der Form nicht möglich)

6 R / Hive (07.06.2018)

- Suchen Sie sich einen für Sie interessanten Bereich aus bspw. Marktforschung, soziale Netzwerke, Wetterstationen, Gesundheitswesen, Natur und Sozialwissenschaften, etc.
- Erzeugen Sie für Ihr Beispiel zwei Hive Tabellen
- Füllen Sie die Tabellen mit einigen Testdaten (LOAD DATA, Insert).
- Bauen Sie eine R Verbindung zur HIVE auf.
- Analysieren Sie die Daten

Kurzdarstellung der Aufgabenstellung

Mittels dem Anlegen von zwei Tabellen in HIVE und dem anschließenden Bearbeiten/-Analysieren mit R soll die das Zusammenspiel von relevanter Software für BigData geübt/demonstriert werden.

Lösung

HIVE

- Ordner in /usr/tmp/hive/ anlegen
- Datensets „hotels.csv“ und „autos.csv“ in /usr/tmp/hive/ legen
- HIVE über den Terminal in Oracle VM4.9:

```
HIVE
```

- Separate Datenbank im HIVE anlegen, falls keine namentlich identische existiert:

```
CREATE DATABASE IF NOT EXISTS diim18;
```

- Tabelle „hotels“ anlegen mit folgendem Befehl:

```
Create TABLE hotels(gewinn FLOAT, preisInMio FLOAT,qm FLOAT,stadt
String) ROW Format DELIMITED FIELDS Terminated by ',' Lines
terminated by '\n';
```

- Tabelle „autos“ anlegen mit folgendem Befehl:

```
Create TABLE autos(preis FLOAT, registrierungJahr INT,ps FLOAT,km INT
, modell String, kraftstoff STRING, name STRING) ROW Format
DELIMITED FIELDS Terminated by ',' Lines terminated by '\n';
```

- Daten aus hotels.csv in HIVE Tabelle hotels laden:

```
LOAD DATA LOCAL INPATH '/usr/tmp/hive/hotels/hotels.csv' Overwrite
INTO Table hotels;
```

- Daten aus autos.csv in HIVE Tabelle autos laden:

```
LOAD DATA LOCAL INPATH '/usr/tmp/hive/autos/autos.csv' Overwrite INTO
Table autos;
```

- Ausgabe der angelegten Tabellen.(Abbildung 30) Die Daten können nun mit SQL

```
hive> Select * from autos LIMIT 5;
OK
NULL    NULL    NULL    NULL    model    fuelType    name
1450.0  1997    75.0    90000   andere   benzin      Toyota_Toyota_Starlet_1_Hand__TueV_nei
13100.0 2005    280.0    5000    golf     benzin      R32_tauschen_oder_kaufen
4500.0   2008    87.0     90000   yaris    benzin      Toyota_Yaris_1.3_VVT_i
6000.0   2009    177.0    125000  3er      diesel      320_Alpinweiss_Kohlenstoff
Time taken: 0.619 seconds, Fetched: 5 row(s)
hive> Select * from hotels LIMIT 5;
OK
NULL    NULL    NULL    Stadt
119000.0    21.88  3938.0  Berlin
250000.0    27.95  3986.0  Muenchen
250000.0    16.09  2574.0  Koeln
145000.0    27.58  4155.0  Muenchen
Time taken: 0.097 seconds, Fetched: 5 row(s)
```

Abbildung 30: Ausgabe Tabellen

Statements betrachtet und manipuliert werden

R

- Starten von R durch Eingabe in den terminal auf der Oracle 4.9 VM:

R

- Installieren des RJDBC packages:

```
install.packages("RJDBC", dep=TRUE)
```

- Aufrufen des RJDBC packages:

```
library("RJDBC")
```

- Driver auf variable in R zur Verwendung zuweisen:

```
drv <- JDBC("org.apache.hive.jdbc.HiveDriver", "/usr/lib/hive/lib  
/hive-jdbc.jar")
```

- Connection Type hinzufügen:

```
library("ORCH")  
ore.connect(type="HIVE")  
ore.sync()
```

- Verbindung zu R aufbauen

```
conn <- dbConnect(drv, "jdbc:hive2://localhost:10000/diim18", "", "")
```

- Inhalt der Tabelle autos im Hive auf die Variable autos in R zuweisen:

```
autos <- dbGetQuery(conn, "Select * from autos")
```

- Lineares Regressionsmodell (lRm) erstellen: $\text{autos.preis} = a * \text{autos.km} + b$

```
model <- lm(autos.preis~autos.km, data = autos)
```

- Modell Zusammenfassung ausgeben lassen (Abbildung 31):

- Erstellen einer neuen Spalte in Autos die den Preis auf Basis des lRm:

```
autos$predicted_1 <- predict(model, autos)
```

- multiples Rm (mRm) erstellen: $\text{autos.preis} = a * \text{autos.km} + b * \text{autos.ps} + b$

```
model <- lm(autos.preis~autos.km+autos.ps, data = autos)
```

```
> print(model)

Call:
lm(formula = autos.preis ~ autos.km, data = autos)

Coefficients:
(Intercept)      autos.km
 1.599e+04    -8.797e-02
```

Abbildung 31: Ausgabe: model

```
> print(model)

Call:
lm(formula = autos.preis ~ autos.km + autos.ps, data = autos)

Coefficients:
(Intercept)      autos.km      autos.ps
 4469.3522    -0.0802    92.0031
```

Abbildung 32: Ausgabe: model

- Modell Zusammenfassung ausgeben lassen (Abbildung 32):
- Erstellen einer neuen Spalte in Autos die den Preis auf Basis des mRm (Abbildung 33)

```
autos$predicted_2 <- predict(model, autos)
print(model)
head(autos)
```

```
> head(autos)
  autos.preis autos.registrierungsjahr autos.ps autos.km autos.modell predicted_1 predicted_2
1      1450          1997         75    90000    andere      8071.301      4151.486
2     13100          2005        280     5000     golf     15548.870     29829.254
3      4500          2008         87    90000     yaris      8071.301      5255.525
4      6000          2009        177   125000        3er      4992.302     10728.774
5      3990          1999        118    90000        3er      8071.301      8107.625
6      4400          2008         71    40000    fortwo     12469.871      7793.534
```

Abbildung 33: Ausgabe: autos

- Inhalt der Tabelle autos im Hive auf die Variable autos in R zuweisen:

```
hotels <- dbGetQuery(conn, "Select * from hotels")
```

- Lineares Regressionsmodell (lRm) erstellen: $\text{hotels.preisinmio} = a * \text{hotels.gewinn} + b * \text{hotels.qm}$

```
model <- lm(hotels.preisinmio ~ hotels.gewinn + hotels.qm, data = hotels)
```

- Erstellen einer neuen Spalte in Autos die den Preis auf Basis des lRm:

```
hotels$predicted_1 <- predict(model, hotels)
```

- Modell Zusammenfassung ausgeben lassen (Abbildung 34):

```
> print(model)

Call:
lm(formula = hotels.preisinmio ~ hotels.gewinn + hotels.qm, data = hotels)

Coefficients:
(Intercept)  hotels.gewinn  hotels.qm
  6.213e+00    4.422e-06    4.016e-03
```

Abbildung 34: Ausgabe: hotels

- multiples Rm (mRm) erstellen: $\text{hotels.preisinmio} = a * \text{hotels.gewinn} + b * \text{hotels.qm} + c * \text{hotels.stadt}$

```
model <-lm(hotels.preisinmio~hotels.gewinn+hotels.qm+hotels.stadt ,
  data = hotels)
```

- Modell Zusammenfassung ausgeben lassen (Abbildung 35)

```
> print(model)

Call:
lm(formula = hotels.preisinmio ~ hotels.gewinn + hotels.qm +
  hotels.stadt, data = hotels)

Coefficients:
(Intercept)          hotels.gewinn          hotels.qm
  6.108e+00          9.021e-06          3.935e-03
hotels.stadtKoeln hotels.stadtMuenchen
 -2.658e+00          3.324e+00
```

Abbildung 35: Ausgabe: hotels

- Erstellen einer neuen Spalte in Autos die den Preis auf Basis des mRm (Abbildung 36):

```
hotels$predicted\_2 <- predict(model, hotels)
```

```
> head(hotels)
  hotels.gewinn hotels.preisinmio hotels.qm hotels.stadt predicted_1 predicted_2
1      119000         21.88      3938      Berlin      22.55325      22.67613
2      250000         27.95      3986      Muenchen      23.32535      27.37071
3      250000         16.09      2574      Koeln      17.65516      15.83256
4      145000         27.58      4155      Muenchen      23.53964      27.08853
5      110000         23.76      3795      Berlin      21.93920      22.03228
6      246000         22.88      2773      Muenchen      18.43660      22.56180
```

Abbildung 36: Ausgabe: hotels

Aufteilung der Aufgaben im Team

Darstellung der benutzten Werkzeuge und Systeme

Entwurfswerkzeug

- HIVE

Entwicklungsumgebung

- R

7 ARIS (14.06.2018)

Definieren Sie für Ihr fiktives/reales Unternehmen folgende Diagrammtypen:

Datenmodell, Geschäftsprozess, IT-Infrastruktur, Organigramm, Prozesslandschaft, BPMN, Whiteboard Beispiel, Systemlandschaft. Die Beispiele finden Sie im Elli.

Kurzdarstellung der Aufgabenstellung

Die Darstellung von Prozessen soll mit ARIS anhand eines fiktiven Unternehmens erprobt werden. Hierfür sollen ein Datenmodell, Geschäftsprozessmodell, IT-Infrastruktur-Modell Organigramm, Prozesslandschafts-Modell, BPMN, Whiteboard-Beispiel und Systemlandschafts-Modell ausgearbeitet in ARIS ausgearbeitet werden.

Lösung

- Installation ARIS
- Starten ARIS
- Diagramme für den entsprechenden Aufgabenbereich Auswählen

1. Organigramm

Es wurde auf die Darstellung der einzelnen Personen mit Namen weitestgehend verzichtet, stellvertretend sind die Rollen („Leiter“, „MAS“ und „Azubi“) zu verstehen. Wobei es in jeder Organisationseinheit („Einkauf“, „Marketing“, etc.) einen Leiter und mehrere Mitarbeiter/Azubis geben kann. Da das Unternehmen auf Grund seiner Größe und des Aktionsfeldes nur im Kaufmännischen Bereich ausbilden kann, gibt es in der IT Abteilung keinen Azubi.

2. Prozesslandschaft Die an der Wertschöpfung beteiligten Prozesse im folgenden Diagramm dargestellt:

3. Geschäftsprozess Der Geschäftsprozess des PSA Handels (keine eigene Produktion) für das fiktive Unternehmen ist im Folgenden dargestellt.

-
4. Datenmodell In diesem Modell werden die Entitäten mit Ihren Attributen dargestellt für den vorherigen Geschäftsprozess bzw. die Prozesslandschaft.
 5. BPMN Die Arbeitsprozesse bezogen auf Angebot und Verkauf ist im Folgenden dargestellt.
 6. Whiteboard Ziel des Brainstorming ist die Effizienzsteigerung mit folgenden Maßnahmen:
 - Lieferantenangebotsübernahme in ERP mittels OCR
 - Webshop-Konzept mit Chat-Fkt.
 - Predictive-Analytice für Bestands-/Bedarfsplanung
 7. IT-Infrastrukturmodell
 - -
 - -
 - -
 8. IT-Systemlandschaftsmodell
 - -
 - -
 - -

Aufteilung der Aufgaben im Team

Darstellung der benutzen Werkzeuge und Systeme

Entwurfswerkzeug

Entwicklungsumgebung