Übung 06
MySQL-Zugriff mit R
INFI-IS
5AHWII

Tobias Laser

January 22, 2022

CCA – COMPETENCE CENTRE
HTL Anichstraße

Bei dieser Übung wird eine Auswertung gemacht die direkt auf Daten aus einer Datenbank basiert. Nebenbei sollen auch SQL-Abfragen über mehrere Tabellen wiederholt werden.

1 Installation und Kennenlernen der employees-Datenbank

In Moodle befindet sich oben bei den Datensätzen die Testdatenbank employees. Diese bitte installieren. Folgende Vorgangsweise ist empfohlen:

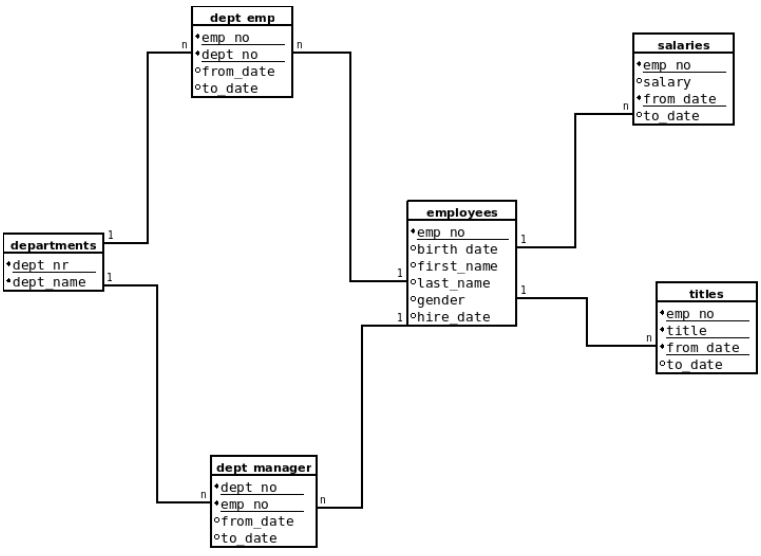
a. Herunterladen und Entpacken (nicht nur ins Archiv wechseln) der Datei unter `Employees DB on GitHub`

b. Umgebungsvariablen für MySQL setzen, damit der mysql-client im Pfad zu finden ist. Eine Anleitung befindet sich hier: `https://michster.de/wie-setze-ich-die-path-umgebungsvariablen-unter-windo`

c. Auf der `CMD` in das Verzeichnis wechseln, wo die heruntergeladenen Dateien liegen

d. Befehl eingeben: `mysql -u root -p < employees.sql`

Ein Überblick über das Datenmodell befindet sich hier:



2 Ein paar Auswertungen dazu...

Bitte mit `RMySQL` auf die **employees**-Datenbank verbinden und folgende Darstellungen/Analysen erstellen. Die Vorgangsweise ist immer gleich wie beim gemeinsamen Beispiel zu den Abfragen:

```
require(RMySQL)
dbConnection = dbConnect(MySQL(), user="infi-greinoecker", password="infi-greinoecker",
    dbname="employees", host="localhost")
```

2.1 Bitte als Boxplots veranschaulichen und interpretieren:

Hinweis zur Auswertung dieser Aufgaben: Die Daten nicht zusammengefasst aus der DB holen, der Boxplot soll ja die Verteilung zeigen.

2.1.1 Wie viele Personen arbeiten aktuell (YEAR(to_date =9999)) in welchen Abteilungen?

```
query <- "SELECT  departments.dept_name as abteilung, dept_emp.emp_no as anzahl_
    mitarbeiter FROM dept_emp RIGHT JOIN departments ON dept_emp.dept_no = departments.
    dept_no WHERE dept_emp.to_date = \"9999-01-01\" ORDER BY dept_emp.dept_no;"
resultsWorkingPeople <- dbSendQuery(dbConnection, query)
dataWorkingPeople <- fetch(resultsWorkingPeople, -1)
boxplot(dataWorkingPeople$anzahl_mitarbeiter~dataWorkingPeople$abteilung, las=2, xlab = ""
    , ylab = "")
```

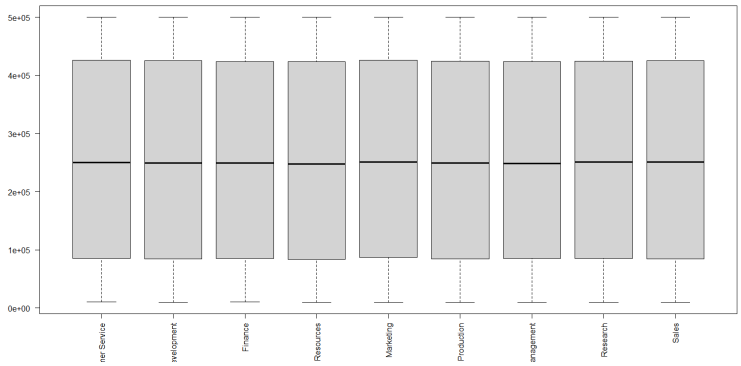


Figure 1: Anzahl der Personen in jeder Abteilung, welche bis Jahr 9999 arbeiten

2.1.2 Den aktuellen Verdienst von Frauen und Männern gegenübergestellt

```
query <- "SELECT employees.emp_no, sum(salaries.salary), employees.gender FROM salaries
    LEFT JOIN employees ON salaries.emp_no = employees.emp_no GROUP BY employees.emp_no;
    "
resultSalaryOverGender <- dbSendQuery(dbConnection, query)
dataSalaryOverGender<- fetch(resultSalaryOverGender, -1)
boxplot(dataSalaryOverGender$`sum(salaries.salary)`~dataSalaryOverGender$gender, xlab =
    "Gender", ylab = "Verdienst")
```

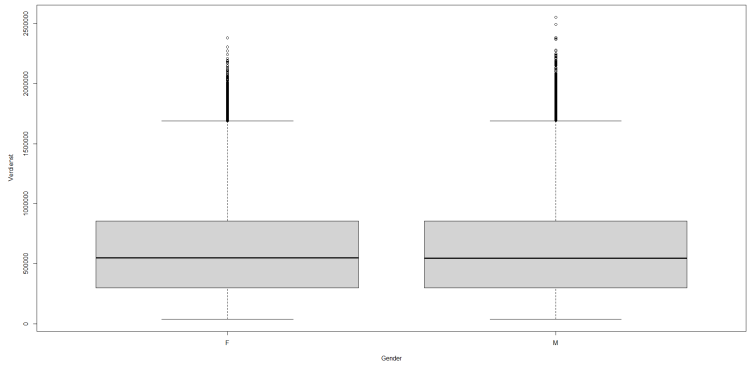


Figure 2: Gegenüberstellung Verdienst von Männer zu Frauen

2.1.3 Den aktuellen Verdienst in den einzelnen Abteilungen gegenübergestellt

```
query <- "SELECT sum(salary) as salaries, d.dept_name as abteilung FROM salaries s INNER
    JOIN dept_emp de ON s.emp_no = de.emp_no INNER JOIN departments d ON de.dept_no = d.
    dept_no GROUP BY de.emp_no ORDER BY d.dept_no;"
resultSalaryOverDepartment <- dbSendQuery(dbConnection, query)
dataSalaryOverDepartment <- fetch(resultSalaryOverDepartment, -1)
boxplot(dataSalaryOverDepartment$salaries~dataSalaryOverDepartment$abteilung, las=2, xlab
    = "", ylab = "")
```

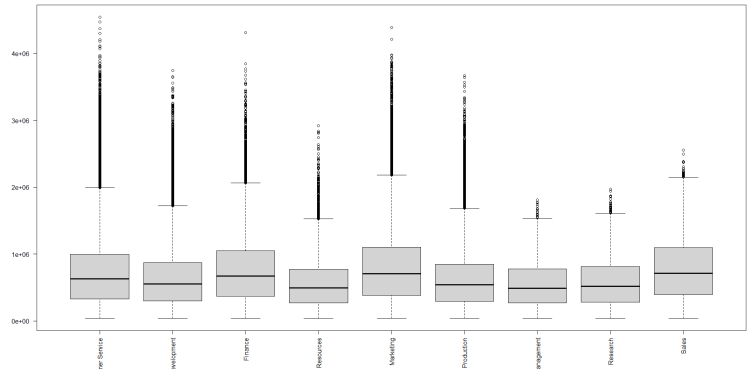


Figure 3: Gegenüberstellung Verdienst pro Abteilung

2.2 Die Gehaltsentwicklung (also die einzelnen Gehälter über die Zeit - ein Mitarbeiter kann mehrere Gehaltssprünge hinter sich haben) des Mitarbeiters mit der `emp_no 492917` als Liniendiagramm dargestellt

Auf die Zeitsprünge, wann die einzelnen Gehaltserhöhungen stattfinden, muss keine Rücksicht genommen werden.

```
query <- "SELECT salary, from_date FROM salaries WHERE emp_no = 492917 ORDER BY from_
    date;"
resultSalaryHistoryFROM492917 <- dbSendQuery(dbConnection, query)
dataSalaryHistoryFrom492917 <- fetch(resultSalaryHistoryFROM492917, -1)
plot(1985:2002,dataSalaryHistoryFrom492917$salary, xlab="year", ylab="salary", type="l")
```

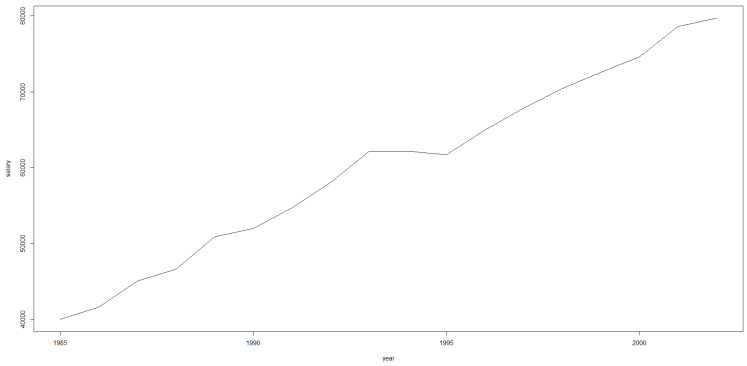


Figure 4: Vom Angestellten 492917 die Gehaltssprünge anzeigen

2.3 Angenommen, die Gehälter wachsen linear, wie schaut dann das Durchschnittsgehalt der aktuellen Gehälter im Jahr 2020 aus?

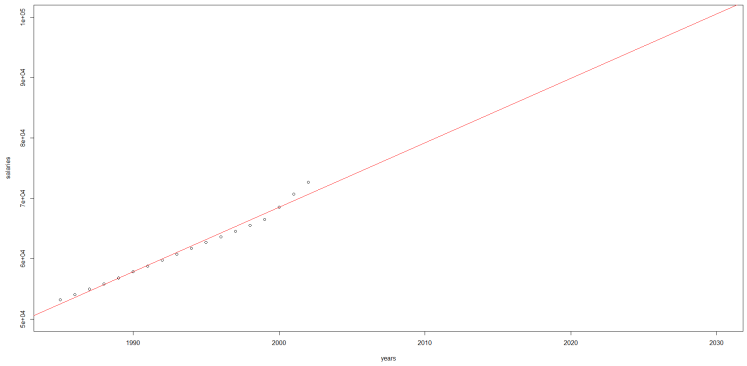
Hier ist eine Abfrage notwendig, die das aktuelle Gehalt nach Jahren gruppiert.

```
query <- "SELECT avg(salary) as average_salary, YEAR(from_date) as year FROM salaries group
    by YEAR(from_date) order by YEAR(from_date);"
resultAverageSalaryPerYear <- dbSendQuery(dbConnection, query)
dataAverageSalaryPerYear <- fetch(resultAverageSalaryPerYear, -1)
```

2.3.1 Bitte grafisch darstellen

Also die einzelnen Gehälter pro Jahr gemeinsam mit der Regressionsgeraden zeichnen lassen.

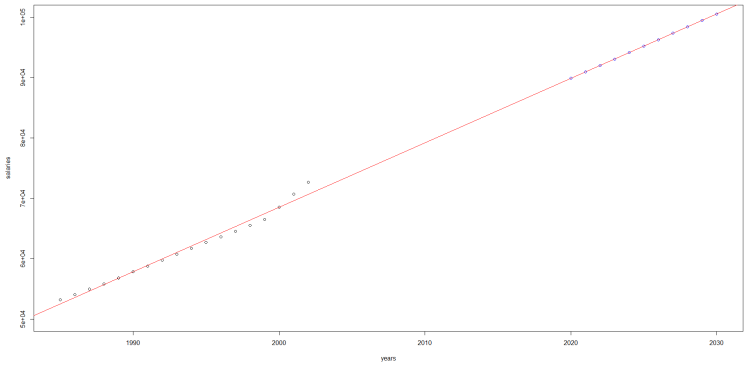
```
  years= dataAverageSalaryPerYear$year
  salaries = as.numeric(dataAverageSalaryPerYear$average_salary)
  lmAverageSalaryPerYear <- lm(salaries~years)
  plot(years, salaries, xlim= c(1985, 2030), ylim=c(50000,100000));
  abline(lmAverageSalaryPerYear, col="red")
```



2.3.2 Den Vorhersagewert für die Jahre 2020-2030 berechnen

Dafür bitte den Befehl `predict` (siehe Beispiel zur Regression) verwenden.

```
predictAverageSalaryPerYear <- predict(lmAverageSalaryPerYear, data.frame(years
    =(2020:2030)))
points(2020:2030, predictAverageSalaryPerYear, col="green")
```



2.4 Gibt es einen Zusammenhang (Korrelation) zwischen Alter der Mitarbeiter und deren Gehalt? Bitte die entsprechenden statistischen Parameter angeben.

Die Korrelation wird mit dem Befehl `cor.test` berechnet und ergibt einen Wert zwischen -1 und 1:

- `-1`: Negative Korrelation: Wenn bei einer Variable die Werte höher sind, dann sind sie bei der zweiten Variable niedriger
- `0`: Kein Zusammenhang zwischen den beiden Variablen
- `1`: Positive Korrelation: : Wenn bei einer Variable die Werte höher sind, dann sind sie bei der zweiten Variable auch höher

```r
query <- "SELECT (year(now())-year(e.birth_date)) as age, avg(s.salary) FROM employees e
    INNER JOIN salaries s ON e.emp_no = s.emp_no GROUP BY age ORDER BY age;"
resultCorrelation <- dbSendQuery(dbConnection, query)
dataCorrelation <- fetch(resultCorrelation, -1)
cor = cor.test(dataCorrelation$age,dataCorrelation$`avg(s.salary)`)
cor # 0.3420241
```

```
> cor # 0.3420241

        Pearson's product-moment correlation

data:  dataCorrelation$age and dataCorrelation$`avg(s.salary)`
t = 1.2608, df = 12, p-value = 0.2313
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2303589  0.7385739
sample estimates:
      cor
0.3420241
```

Hinweise:

- Ein Beispiel-Skript zur Datenbankverbindung und Abfrage befindet sich im Moodle
- Idealerweise sollten die Abfragen so gemacht werden, dass die Daten schon optimal für die Weiterverwendung in R sind
- Es gibt ein Datenmodell in Moodle, um einen Überblick über die `employees`-DB zu bekommen
- Die aktuellen Werte für Gehalt, Zugehörigkeit zu einer Abteilung, ... bekommt man immer mit dem MySQL-Befehl `YEAR(to_date) = 9999`
- RMySQL holt per default nicht alle Werte, das kann man mit der Einstellung `n=-1` abstellen: