Übung 06
`MySQL-Zugriff mit R`
INFI-IS
5xHWII

Albert Greinöcker
Thomas Kefer

March 30, 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:

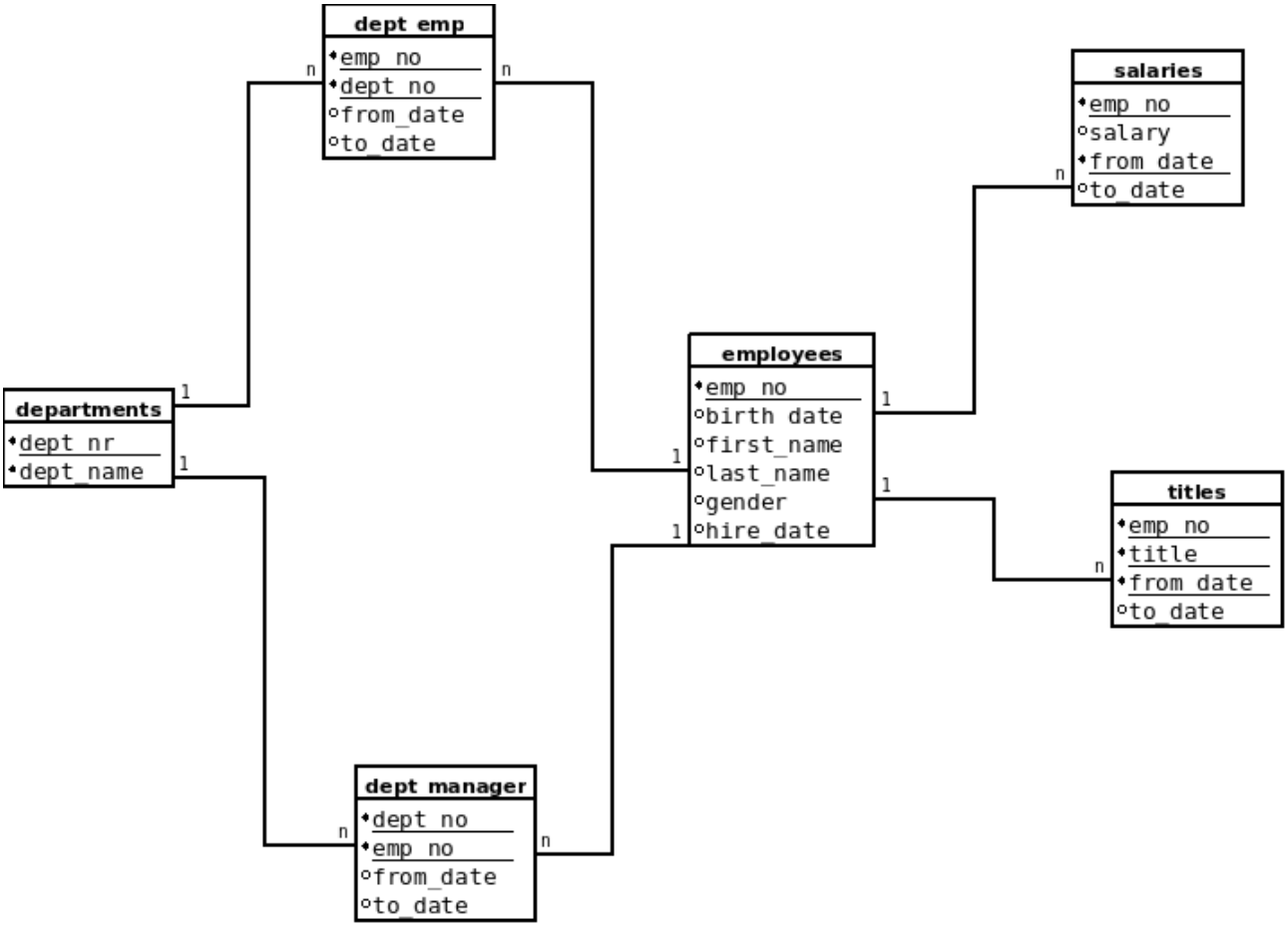
a. Herunterladen und Entpacken (nicht nur ins Archiv wechseln) der Datei unter `Employees DB on GitHub`

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:



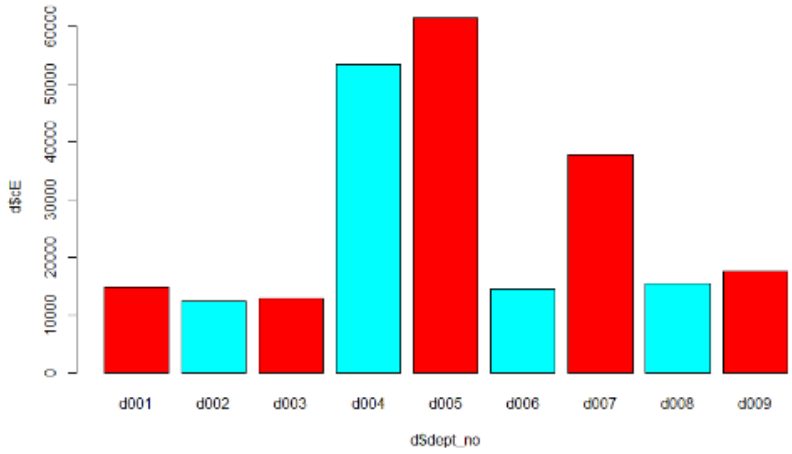
2 Ein paar Auswertungen dazu...

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

2.1 Bitte als Boxplots veranschaulichen und interpretieren:

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

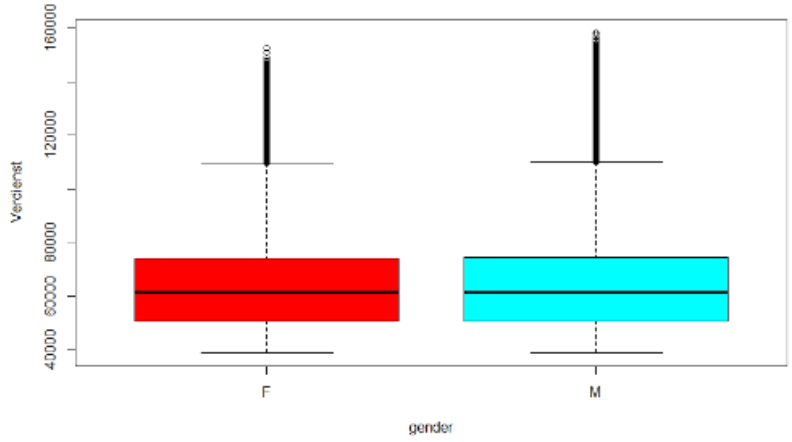
```
q <- dbSendQuery(c, "select count(*)cE, dept_no from dept_emp where year(to_date) = 9999
    group by dept_no")
d <- fetch(q, -1)
barplot(d$cE~d$dept_no, col=rainbow(2))
dbClearResult(q)
```



Anzahl der Personen, die bis zum Jahr 9999 angestellt sind.

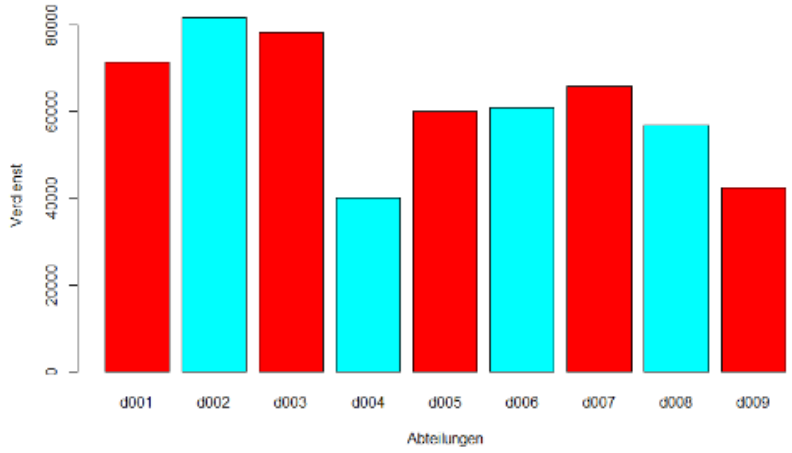
2.1.2 Den aktuellen Verdienst von Frauen und Männern gegenübergestellt

```
q <- dbSendQuery(c,"select (employees.gender)gnd, (salary) sal  from salaries inner join
    employees on salaries.emp_no = employees.emp_no")
d <- fetch(q, -1)
boxplot(d$sal~d$gnd, xlab= "gender", ylab = "Verdienst", col=rainbow(2))
dbClearResult(q)
```



2.1.3 Den aktuellen Verdienst in den einzelnen Abteilungen gegenübergestellt

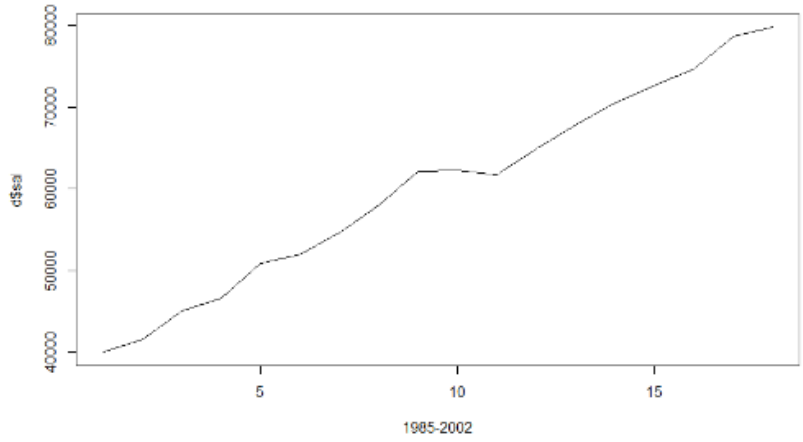
```
q <- dbSendQuery(c,"select (employees.gender)gnd, (salary) sal  from salaries inner join
    employees on salaries.emp_no = employees.emp_no")
d <- fetch(q, -1)
boxplot(d$sal~d$gnd, xlab= "gender", ylab = "Verdienst", col=rainbow(2))
dbClearResult(q)
```



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.

```
q <- dbSendQuery(c,"select (salaries.salary) sal, (salaries.from_date) date from employees
    inner join salaries on employees.emp_no = salaries.emp_no where employees.emp_no =
    492917;")
d <- fetch(q,-1)
plot(d$sal,xlab = "1985-2002",type="l")
dbClearResult(q)
```



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.

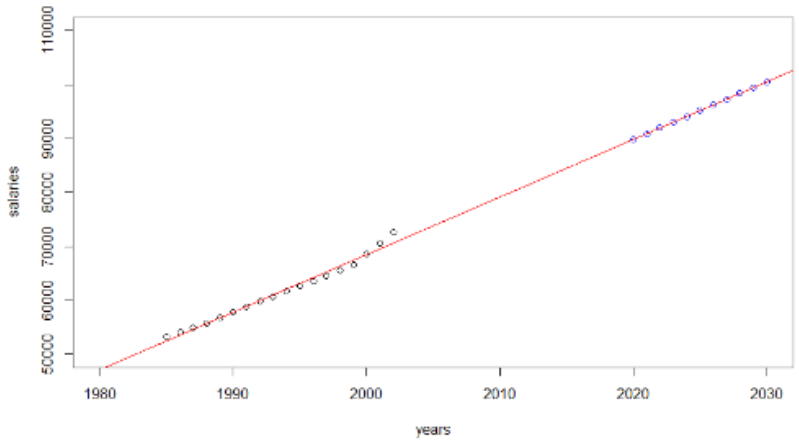
2.3.1 Bitte grafisch darstellen

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

2.3.2 Den Vorhersagewert für die Jahre 2020-2030 berechnen

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

```
gem <- function(year, sal)
{
  l <- lm(year ~ sal)
  plot(sal,year, xlab = "years" , ylab="salaries", xlim= c(1980,2030), ylim=c(50000,110000))
  abline(l, col="red")
}
q <- dbSendQuery(c,"select (avg(salaries.salary)) avgsal, (year(salaries.from_date)) ysal
    from employees inner join salaries on employees.emp_no = salaries.emp_no group by year (
    salaries.from_date) order by year(salaries.from_date);")
d <- fetch(q,-1)
lm_sal <- lm(avgsal ~ ysal, data = d)
sal_predict <- predict(lm_sal, data.frame(ysal=c(2020:2030)))
gem(d$avgsal,d$ysal)
points(c(2020:2030),sal_predict,col="blue")
dbClearResult(q)
```



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

```
q <- dbSendQuery(c,"select (year(curdate()) - year(birth_date)) age , (salaries.salary)
    salar from employees inner join salaries on employees.emp_no = salaries.emp_no where
    year(salaries.to_date) = 9999;")
d <- fetch(q,-1)
cor.test(d$age,d$salar , method="kendall")
```

```
# Kendalls rank correlation tau
#
# data: d$age and d$salar
# z = -0.86685, p-value = 0.386
# alternative hypothesis: true tau is not equal to 0
# sample estimates:
# tau = -0.001223331
```

Man erkennt, dass der Zusammenhang relativ groß ist, da der P-Wert über null, bei etwa 0,38 liegt. Das Maximum wäre bei P-Wert = 1.
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:

```
res <- dbSendQuery(con, '<my_query>')
data1 <- fetch(res, n = -1)
```