

Desafio 8

Maria Eduarda Villéla Silva

25/09/2025 08:27:41

```
library(RSQLite)
```

1. Baixe o arquivo `uwmadison.sqlite3`. Conecte-se a ele usando o pacote `RSQLite`, armazenando a conexão em uma variável `conn`.

```
#conectando o R ao SQLite
```

```
conn <- dbConnect(SQLite(), dbname = "uwmadison.sqlite3")
conn
```

```
## <SQLiteConnection>
```

```
## Path: \\smb\ra212974\Documentos\desafios\uwmadison.sqlite3
```

```
## Extensions: TRUE
```

```
dbListTables(conn) #vendo quais são as tabelas do dataset
```

```
## [1] "course_offerings" "courses" "grade_distributions"
## [4] "instructors" "rooms" "schedules"
## [7] "sections" "subject_memberships" "subjects"
## [10] "teachings"
```

2. Quem são e quantos são os professores que lecionaram disciplinas cujo tópico era estatística (`subjects.abbreviation='STAT'`);

```
#os comentários em sql são indicados por --
```

```
#os comentários ficam da mesma cor q o código
```

```
#consulta SQL que retorna os professores que já lecionaram disciplinas STAT
```

```
professores_STAT <- dbGetQuery(conn, " -- executa a query no banco conectado
```

```
-- seleciona nomes únicos (DISTINCT) dos professores
```

```
SELECT DISTINCT i.name AS professor
```

```
-- tabela com informações de professores (apelidada como i)
```

```
FROM instructors i
```

```
-- liga cada professor à tabela teachings (que indica em qual seção ele lecionou)
```

```
JOIN teachings t ON i.id = t.instructor_id
```

```
-- liga teachings às sections (cada seção de uma oferta de curso)
```

```
JOIN sections se ON t.section_uuid = se.uuid
```

```
-- liga cada seção à oferta específica de um curso
```

```
JOIN course_offerings co ON se.course_offering_uuid = co.uuid
```

```
-- liga a oferta à tabela que mapeia ofertas para assuntos (subjects)
```

```
JOIN subject_memberships sm ON co.uuid = sm.course_offering_uuid
```

```
-- liga o assunto ao seu código para poder filtrar
```

```
JOIN subjects s ON sm.subject_code = s.code
```

```
-- filtra apenas as ofertas cujo assunto (abbreviation) é STAT
```

```
WHERE s.abbreviation = 'STAT'
```

```
-- ordena a lista final em ordem alfabética pelo nome do professor
```

```
ORDER BY i.name
")

#quantidade de professores sendo contada por linha
qtd_professores <- nrow(professores_STAT)
qtd_professores

## [1] 377

#retorna o nome dos professores de estatística
professores_STAT
```

```
##           professor
## 1          ABIGAIL BENZINE
## 2        ADIN-CRISTIAN ANDREI
## 3          AKICHIKA OZEKI
## 4           ALAN HUANG
## 5        ALBERTO DEL PIA
## 6        ALBRECHT KLEMM
## 7    ALEXANDER COVINGTON
## 8        ALEXANDER FISH
## 9        ALEXANDER KISELEV
## 10         ALYSSA DIGILIO
## 11         ANATOLE BECK
## 12        ANDREJ ZLATOS
## 13        ANDREW LESLIE
## 14        ANNE BRUCKNER
## 15           ANQI SHI
## 16           ANRU ZHANG
## 17        ARNOLD MILLER
## 18        AUGUST JENSEN
## 19        BEHZAD AALIPUR
## 20    BEHZAD AALIPUR HAFSHEJANI
## 21        BEN ADAM HAALAND
## 22        BENEDEK VALKO
## 23        BENJAMIN RECHT
## 24           BI CHENG WU
## 25           BIN DAI
## 26           BIN ZHANG
## 27           BIN ZHU
## 28        BINGYING XIE
## 29           BO HUANG
## 30           BO YANG
## 31           BOWEN HU
## 32           BRET HANLON
## 33           BRET LARGET
## 34        BRIAN YANDELL
## 35           BROOK LUERS
## 36           CECILE ANE
## 37           CHAN PARK
## 38        CHANHAN HSU
## 39        CHAOQUN MEI
## 40        CHAOYANG YU
## 41        CHELSEY GREEN
## 42           CHEN CHENG
```

## 43	CHEN JING
## 44	CHENGNING ZHANG
## 45	CHENLIANG XU
## 46	CHENSHENG KUANG
## 47	CHENXI LI
## 48	CHIA-CHIEH LIN
## 49	CHIEN-WEI CHEN
## 50	CHRISTINA M. KENDZIORSKI
## 51	CHRISTINE SORKNESS
## 52	CHRISTOPHER WAGNER
## 53	CLAIRE BOBST
## 54	CLAUDIA SOLIS LEMUS
## 55	COLE COOK
## 56	COLIN LONGHURST
## 57	CRYSTAL CHEN
## 58	CUICUI QI
## 59	CUIZE HAN
## 60	DANIELE CAPPELLETTI
## 61	DAVID ANDERSON
## 62	DAVID DEMETS
## 63	DAVID GRIFFEATH
## 64	DEBRAJ DAS
## 65	DEREK BEAN
## 66	DEREK NORTON
## 67	DEYUAN JIANG
## 68	DIETRICH UHLENBROCK
## 69	DONALD PORTER
## 70	DONG XIA
## 71	DONGGYU KIM
## 72	DONGHYUN LEE
## 73	DOUGLAS M. BATES
## 74	DUY NGUYEN
## 75	DUZHE WANG
## 76	EDWARD ERKER
## 77	ELOISA D CHAVAS
## 78	ERICA LEE DEADMAN
## 79	ERIK NORDHEIM
## 80	FAN GAO
## 81	FAN YANG
## 82	FANG FANG
## 83	FANGFANG WANG
## 84	FLORIAN BERTRAND
## 85	FREDERICK BOEHM
## 86	GARVESH RASKUTTI
## 87	GARY HOWARD SCHROEDER
## 88	GENG LI
## 89	GINA BENNINGER
## 90	GINA OH
## 91	GONZALO CONTADOR
## 92	GRACE WAHBA
## 93	GREGORIO MORENO-FLORES
## 94	GREGORY SHINAULT
## 95	GUANHUA CHEN
## 96	GUANNAN SUN

## 97	GUILHERME ROSA
## 98	GUILHERME VIEIRA NUNES LUDWIG
## 99	GUN WOONG PARK
## 100	HAN CHEN
## 101	HAO CHEN
## 102	HAO TENG
## 103	HAO ZHENG
## 104	HAO ZHOU
## 105	HAODA FU
## 106	HAOYANG FAN
## 107	HEATHER MARIE BRAZEAU
## 108	HUAIBAO FENG
## 109	HUI WANG
## 110	HUIKUN ZHANG
## 111	HUILIN HU
## 112	HYEBIN SONG
## 113	HYUNSEUNG KANG
## 114	IAN BRANSTAD RILEY
## 115	ISMOR FISCHER
## 116	JAMES ANDERSON
## 117	JAMES D KUELBS
## 118	JARED BROWN
## 119	JASON P FINE
## 120	JASON RICHARD SWANSON
## 121	JEE YEON KIM
## 122	JEEA CHOI
## 123	JENNIFER BIRSTLER
## 124	JENNIFER NGUYEN
## 125	JESSE THOMAS HOLZER
## 126	JIAJIE CHEN
## 127	JIALE XU
## 128	JIAN WU
## 129	JIANCHANG HU
## 130	JIE SONG
## 131	JIE WEI
## 132	JIE ZHANG
## 133	JILI WANG
## 134	JINGCI MENG
## 135	JINGJIANG PENG
## 136	JINGLAN LI
## 137	JIWEI ZHAO
## 138	JOHN DAVIS
## 139	JOHN GILLET
## 140	JOHN KANE
## 141	JOHN WILTSHIRE-GORDON
## 142	JONATHON PETERSON
## 143	JOSEP GINEBRA
## 144	JOSEPH DEUTSCH
## 145	JU HEE CHO
## 146	JUN LI
## 147	JUN SHAO
## 148	JUN YIN
## 149	JUN ZHANG
## 150	JUN ZHU

## 151	JUNGWON MUN
## 152	JUNHEE HAN
## 153	JUNHO LEE
## 154	KAILEI CHEN
## 155	KAM-WAH TSUI
## 156	KARL BROMAN
## 157	KARL ROHE
## 158	KATHERINE GOODE
## 159	KAZUHIKO SHINKI
## 160	KEEGAN KORTHAUER
## 161	KEN ONO
## 162	KEVIN HASEGAWA ENG
## 163	KEVIN PACKARD
## 164	KJELL DOKSUM
## 165	KRISHNAKUMAR BALASUBRAMANIAN
## 166	KRISTEN CYFFKA
## 167	KUNLING HUANG
## 168	KYLE HEBERT
## 169	KYLE HERBET
## 170	KYUNGMAHN KIM
## 171	LAM HO
## 172	LAN LUO
## 173	LANCINE KONATE
## 174	LEI XU
## 175	LEV BORISOV
## 176	LIAM JOHNSTON
## 177	LIE XIONG
## 178	LILI LAN
## 179	LILI ZHENG
## 180	LILUN DU
## 181	LIN QI
## 182	LONG PHAN
## 183	LU MAO
## 184	LU YANG
## 185	LUWAN ZHANG
## 186	LUXI CAO
## 187	MANJUSHA KANCHARLA
## 188	MARI PALTA
## 189	MARIA KAMENETSKY
## 190	MARIA ROJO
## 191	MARIAN R FISHER
## 192	MARY LINDSTROM
## 193	MATTHEW BALLARD
## 194	MENG SONG
## 195	MICHAEL FERRIS
## 196	MICHAEL GEORGE ILTIS
## 197	MICHAEL HOGAN
## 198	MICHAEL KUTZLER
## 199	MICHAEL LIOU
## 200	MICHAEL NEWTON
## 201	MICHAEL RENE KOSOROK
## 202	MICHELLE HARRIS
## 203	MIHAELA IFRIM
## 204	MIN JUNG LEE

## 205	MIN NIU
## 206	MING XIE
## 207	MING YUAN
## 208	MINJING TAO
## 209	MITCHELL PAUKNER
## 210	MOO K CHUNG
## 211	MUHONG GAO
## 212	MURRAY CLAYTON
## 213	NATALIA DE LEON GATTI
## 214	NELLIE LAUGHLIN
## 215	NICHOLAS HENDERSON
## 216	NICHOLAS STEPHEN KEULER
## 217	NING FAN
## 218	NORBERT BINKIEWICZ
## 219	NORMAN DRAPER
## 220	PAUL M TERWILLIGER
## 221	PAUL RATHOUZ
## 222	PAUL SAVARIAPPAN
## 223	PEIGEN ZHOU
## 224	PERLA REYES
## 225	PHILIP WOOD
## 226	QI JIANG
## 227	QI TANG
## 228	QIAN ZHIGUANG
## 229	QING LI
## 230	QIONG ZHANG
## 231	QIURONG CUI
## 232	QUEFENG LI
## 233	QUOC TRAN
## 234	REBECCA KOSCIK
## 235	REBECCA POST
## 236	RICHARD A BRUALDI
## 237	RICHARD A. JOHNSON
## 238	RICHARD J. CHAPPELL
## 239	ROBERT HARRON
## 240	ROBERT R MEYER
## 241	ROBERT WARDROP
## 242	ROBERT WAYNE GREEN
## 243	RONALD GANGNON
## 244	RONGJUN ZHU
## 245	RUI CHEN
## 246	RUI TANG
## 247	RUIFANG SONG
## 248	RUIFENG XU
## 249	RUNGANG HAN
## 250	RUOSI GUO
## 251	RYAN ZEA
## 252	SAMUEL STECHMANN
## 253	SANGBUM CHOI
## 254	SCOTT HOTTOVY
## 255	SCOTT JOSEPH HETZEL
## 256	SEAN KENT
## 257	SEBASTIEN ROCH
## 258	SEHO PARK

## 259	SEULKEE YUN
## 260	SEUNGBONG HAN
## 261	SHAN LU
## 262	SHANE HUBLER
## 263	SHANG WU
## 264	SHENG WANG
## 265	SHENG ZHANG
## 266	SHENGJI JIA
## 267	SHI JIN
## 268	SHIXUE LIU
## 269	SHIZHEN WANG
## 270	SHUAI CHEN
## 271	SHUANG HUANG
## 272	SHULEI WANG
## 273	SHUYUN YE
## 274	SIJIAN WANG
## 275	SIJING LI
## 276	SOHEIL SADEGHI
## 277	SOKOL VAKO
## 278	SONA Z SWANSON
## 279	SONG WANG
## 280	SRIKANTHMADHAVAN ARAVAMUTHAN
## 281	STEFFEN LEMPP
## 282	STEPHEN AARON STANHOPE
## 283	STEPHEN BERG
## 284	STEPHEN WAINGER
## 285	STEPHEN WRIGHT
## 286	STEVEN SAM
## 287	SUKHENDU MEHROTRA
## 288	SUNDUZ KELES
## 289	TAERI UHM
## 290	TAO YU
## 291	THEVAASIINEN CHANDERENG
## 292	THOMAS COOK
## 293	THOMAS G. KURTZ
## 294	THU LE
## 295	TIEN VO
## 296	TIMO SEPPALAINEN
## 297	TIMOTHY IDOWU
## 298	TING YE
## 299	TING-LI LIN
## 300	TONG LI
## 301	TONGHAI YANG
## 302	TRAM TA
## 303	TUN LEE NG
## 304	TZU HSIANG HUNG
## 305	VARSHA KULKARNI
## 306	VICTOR LUO
## 307	VICTORIA MANSFIELD
## 308	WAI TONG FAN
## 309	WEI ZHENG
## 310	WEI-YIN LOH
## 311	WENWEN ZHANG
## 312	WENZHI CAO

## 313	WESLEY CHANG
## 314	XIAO GUO
## 315	XIAO NIE
## 316	XIAODAN WEI
## 317	XIAOMAO LI
## 318	XIAOPING FENG
## 319	XIAOWU DAI
## 320	XIN LI
## 321	XIN ZHANG
## 322	XINJIE HE
## 323	XINWEI DENG
## 324	XINXIN YU
## 325	XINYU SONG
## 326	XIPEI YANG
## 327	XIRAN WANG
## 328	XIUFENG SHAO
## 329	XIUYU MA
## 330	XIWEN MA
## 331	XU HE
## 332	XU XU
## 333	XUEYAO CHEN
## 334	XUN ZHAO
## 335	YAJUAN SI
## 336	YALI WANG
## 337	YAN CHEN
## 338	YANBING ZHENG
## 339	YANG ZHAO
## 340	YANNAN QIU
## 341	YAOGUO XIE
## 342	YAOYAO XU
## 343	YAZHEN WANG
## 344	YI CHAI
## 345	YI LI
## 346	YI LIU
## 347	YIFAN MEI
## 348	YILIN ZHANG
## 349	YING ZHANG
## 350	YONGFENG WU
## 351	YONGJOON KIM
## 352	YONGSU LEE
## 353	YOUNG LEE
## 354	YOUNG MIN PARK
## 355	YOUNGDEOK HWANG
## 356	YOURAN QI
## 357	YU MENGANG
## 358	YU QIU LIU
## 359	YUAN JIANG
## 360	YUAN LI
## 361	YUAN WANG
## 362	YUANZHI LI
## 363	YUCHANG WU
## 364	YUCHEN ZHOU
## 365	YUJIN CHUNG
## 366	YUNONG LIN


```
## 367          YUQING XU
## 368          ZHANG CHUNMING
## 369          ZHENGJUN ZHANG
## 370          ZHENGXIAO WU
## 371          ZHIGENG GENG
## 372          ZHIGUO XIAO
## 373          ZHONGJIE YU
## 374          ZHUANG WU
## 375          ZIFENG ZHAO
## 376          ZIJIAN NI
## 377          ZUOFENG SHANG
```

3. O GPA americano é definido numa escala de 0 a 4, em que $A = 4$, $AB = 3.5$, $B = 3$, $BC = 2.5$, $C = 2$, $D = 1$ e $F = 0$. Determinando a nota média de cada oferecimento pela ponderação da quantidade de alunos em cada extrato com os valores numéricos de cada conceito, indique (no que se referente a disciplinas no assunto de estatística):

- Quem é o professor mais difícil?
- Quem é o professor mais fácil?
- Qual é a disciplina mais difícil?
- Qual é a disciplina mais fácil?

```
#pacotes necessários
library(DBI) #fornece funções para conectar e interagir com bancos de dados
library(dplyr) #pacote para manipulação de dados de forma mais legível, tipo pipe
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
#abre uma conexão com o dataset
conn <- dbConnect(SQLite(), "uwmadison.sqlite3")

#seleção de notas em disciplinas de estatística
#o join está juntando tabelas diferentes
gpa_stat <- dbGetQuery(conn, "
  SELECT co.uuid AS course_offering_uuid, -- id único da oferta de curso
         c.name AS disciplina,           -- nome da disciplina
         i.name AS professor,            -- nome do professor
         gd.a_count, gd.ab_count, gd.b_count, gd.bc_count,
         gd.c_count, gd.d_count, gd.f_count -- quantidades de cada nota atribuída
  FROM grade_distributions gd
  JOIN course_offerings co
    ON gd.course_offering_uuid = co.uuid
  JOIN courses c
    ON co.course_uuid = c.uuid
  JOIN subject_memberships sm
    ON co.uuid = sm.course_offering_uuid
  JOIN subjects s
    ON sm.subject_code = s.code
```

```

JOIN sections sec
  ON co.uuid = sec.course_offering_uuid
JOIN teachings t
  ON sec.uuid = t.section_uuid
JOIN instructors i
  ON t.instructor_id = i.id
WHERE s.abbreviation = 'STAT' -- filtra apenas disciplinas de estatística
") #executa a query SQL e retorna em um dataframe chamado gpa_stat

#conversão das colunas de notas
num_cols <- c("a_count", "ab_count", "b_count", "bc_count", "c_count", "d_count", "f_count")
gpa_stat[num_cols] <- lapply(gpa_stat[num_cols], function(x) {
  x <- as.numeric(x) #garante que cada coluna seja numérica
  x[is.na(x)] <- 0    #substitui valores ausentes (NA) por zero
  x
})

#cálculo do GPA por turma
gpa_stat <- gpa_stat %>%
  mutate(
    #soma total de alunos avaliados em cada turma
    total = rowSums(select(., all_of(num_cols))),
    #calcula GPA ponderado (se houver alunos)
    gpa = ifelse(total > 0,
      (4*a_count + 3.5*ab_count + 3*b_count + 2.5*bc_count +
       2*c_count + 1*d_count) / total, #fórmula ponderada de GPA
      NA_real_) #caso não haja alunos, retorna NA

#agrupar por professor e disciplina para calcular o GPA médio
prof_gpa <- gpa_stat %>%
  group_by(professor) %>% #agrupa todas as turmas pelo mesmo professor
  summarise(gpa_medio = mean(gpa, na.rm = TRUE)) #calcula a média de GPA ignorando NAs

disc_gpa <- gpa_stat %>%
  group_by(disciplina) %>% #agrupa todas as turmas da mesma disciplina
  summarise(gpa_medio = mean(gpa, na.rm = TRUE)) #calcula a média de GPA por disciplina

#professores mais difíceis/mais fáceis
gpa_min_prof <- min(prof_gpa$gpa_medio, na.rm = TRUE) #menor GPA médio entre professores
gpa_max_prof <- max(prof_gpa$gpa_medio, na.rm = TRUE) #maior GPA médio entre professores

#filtra todos os professores com menor média (mais difíceis)
professores_mais_dificeis <- prof_gpa %>%
  filter(gpa_medio == gpa_min_prof)

#filtra todos os professores com maior média (mais fáceis)
professores_mais_faceis <- prof_gpa %>%
  filter(gpa_medio == gpa_max_prof)

#disciplinas mais difíceis/mais fáceis
gpa_min_disc <- min(disc_gpa$gpa_medio, na.rm = TRUE) #menor GPA médio entre disciplinas
gpa_max_disc <- max(disc_gpa$gpa_medio, na.rm = TRUE) #maior GPA médio entre disciplinas

```

```

#filtra todas as disciplinas mais difíceis
disciplinas_mais_dificeis <- disc_gpa %>%
  filter(gpa_medio == gpa_min_disc)

# filtra todas as disciplinas mais fáceis
disciplinas_mais_faceis <- disc_gpa %>%
  filter(gpa_medio == gpa_max_disc)

#resultados
resultado <- bind_rows(
  professores_mais_dificeis %>%
    transmute(categoria = "Professor mais difícil", #cria coluna fixa "categoria"
              nome = professor, #coloca o nome do professor na coluna "nome"
              gpa_medio), #mantém a média calculada

#todas as linhas abaixo têm explicação análoga à de cima
  professores_mais_faceis %>%
    transmute(categoria = "Professor mais fácil",
              nome = professor,
              gpa_medio),

  disciplinas_mais_dificeis %>%
    transmute(categoria = "Disciplina mais difícil",
              nome = disciplina,
              gpa_medio),

  disciplinas_mais_faceis %>%
    transmute(categoria = "Disciplina mais fácil",
              nome = disciplina,
              gpa_medio)
)

print(resultado) #mostra o resultado na tela

```

```

## # A tibble: 9 x 3
##   categoria      nome      gpa_medio
##   <chr>          <chr>      <dbl>
## 1 Professor mais difícil JAMES D KUELBS      2.60
## 2 Professor mais fácil  GUANHUA CHEN        4
## 3 Professor mais fácil  SRIKANTHMADHAVAN ARAVAMUTHAN 4
## 4 Professor mais fácil  YAJUAN SI           4
## 5 Disciplina mais difícil Introduction to the Theory of Probability 2.90
## 6 Disciplina mais fácil  Data Science Practicum      4
## 7 Disciplina mais fácil  Empir Proc&Semiparmtrc Infernc 4
## 8 Disciplina mais fácil  Nonparametric Statistics and Machine Learni~ 4
## 9 Disciplina mais fácil  Sample Survey Theory and Method 4

```

4. Desconecte do banco de dados.

```
dbDisconnect(conn)
```