What are the Cowboyest Names in Switzerland?

Data Management and Integration Assignment

Marius Furter and Morley J Weston

2022-11-23

Introduction

For our data integration project we want to determine which Swiss family names are the cowboyest. For this we gathered data that allows us to link familiy name frequency with the number of cattle present in a commune (Gemeinde). We made use of the following data sources:

gemeinde.csv

Provides us with the

- commune identification number
- commune name
- commune population

Gathered from the Swiss federal geodata portal at data.geo.admin.ch, and converted from an ESRI shapefile into a CSV in QGIS.

cattle-map-commune.csv

Provides us with

- commune id
- cattle population

Gathered from identitas animal statistics here.

commune-languageregion.csv

Provides us with

- commune id
- commune language region

Gathered from BFS here.

familiy-names-commune.csv

Lists the 100 most frequent family names in each commune Provides us with

- commune id
- family name
- family name count in commune
- family name rank in commune

Gathered from BFS here.

cattle-NamesFemaleCalves.csv and cattle-NamesMaleCalves

Lists most popular names for named calves by language region. Provides us with

- calf name
- name count
- name rank
- language region

Gathered from identitas here.

Database organization

Using the data above we organize our database according to the following ER diagram.

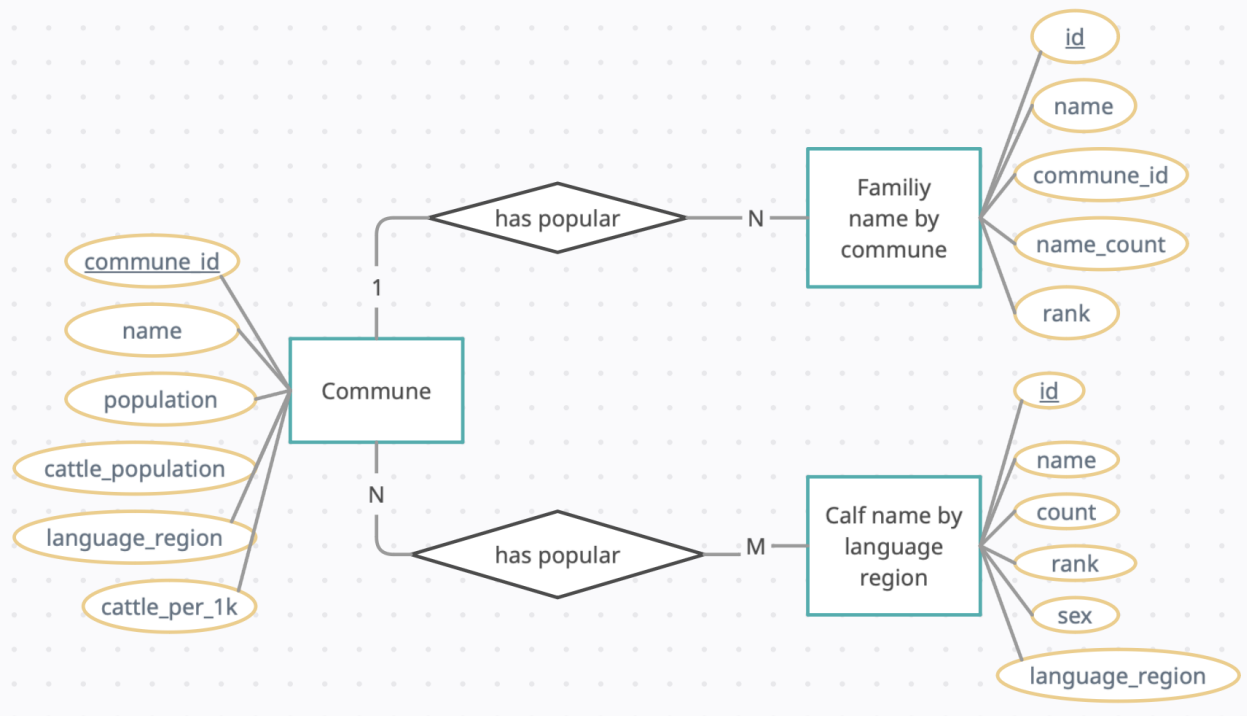

Figure 1: ER diagram of communes and cattle

Each commune is related to the top 100 family names in that commune. Moreover, each commune is related to the top calf names in the language region(s) it belongs to. In reality, no commune is assigned to multiple language regions although this sould be reasonable in principle.

Implementing the database

We start by instantiating a database in SQLite.

```
library(tidyverse)
library(DBI)
library(knitr)
```

```
conn <- dbConnect(RSQLite::SQLite(), "cowboys.db")
```

Commune table

The data in the commune table requires integration from three sources. First we load the commune data from `gemeinde.csv`.

```
communes <- read_csv("input_data/gemeinde.csv")
communes <- communes %>%
  filter(!is.na(EINWOHNERZ)) %>%
  select(BFS_NUMMER, EINWOHNERZ, NAME) %>%
  rename(commune_id = BFS_NUMMER, population = EINWOHNERZ, name = NAME)

communes %>% sample_n(10) %>% kable()
```

commune_id	population	name
3233	3928	Berneck
1123	1959	Altishofen
4005	4400	Erlinsbach (AG)
4104	3157	Lupfig
1093	7174	Neuenkirch
981	5097	Niederbipp
1099	3040	Schenkon
1086	3293	Grosswangen
6192	296	Blatten
2096	5417	Romont (FR)

Next we extract the cattle population per commune from `cattle-map-commune.csv`.

```
cattle_population <- read_delim("input_data/cattle-map-commune.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE)
cattle_population <- cattle_population %>%
  select(MunicipalityNumber, count) %>%
  rename(commune_id = MunicipalityNumber, cattle_population = count)

cattle_population %>% sample_n(10) %>% kable()
```

commune_id	cattle_population
497	271
3701	319
6622	131
6451	156
5061	465
2785	162
567	2714
3408	910
2773	13
6724	170

We then join the cattle population with the commune data using the `commune_id`.

```
communes <- left_join(communes, cattle_population)

communes%>% sample_n(10) %>% kable()
```

commune_id	population	name	cattle_population
5707	1341	Chavannes-de-Bogis	47
4166	1568	Herznach	376
1083	3322	Buttisholz	2302
5634	3127	Echichens	597
2857	608	Rickenbach (BL)	229
580	531	Hofstetten bei Brienz	283
4309	3540	Klingnau	80
4064	1061	Büttikon	164
1055	1421	Gisikon	105
5642	16101	Morges	0

Finally, we extract the language region of each commune from `commune-languageregion.csv`.

```
commune_language <- read_delim("input_data/commune-languageregion.csv",
    delim = ",", escape_double = FALSE, trim_ws = TRUE) %>%
      select(`Regions-ID`, Sprachgebiete) %>%
      rename(commune_id = `Regions-ID`, language_region = Sprachgebiete)

commune_language$language_region <- recode( commune_language$language_region,
                                        `Deutsches Sprachgebiet` = "de",
                                        `Französisches Sprachgebiet` = "fr",
                                        `Italienisches Sprachgebiet` = "it",
                                        `Rätoromanisches Sprachgebiet` = "rt"
                                          )

commune_language %>% sample_n(10) %>% kable()
```

commune_id	language_region
352	de
746	de
3232	de
2275	de
6057	de
1059	de
4075	de
2865	de
670	de
617	de

The result is joined to the commune data via `commune_id`.

```
communes <- left_join(communes, commune_language)

communes%>% sample_n(10) %>% kable()
```

commune_id	population	name	cattle_population	language_region
5398	5163	Gambarogno	17	it
6809	1096	Haute-Ajoie	1853	fr
4881	1332	Amlikon-Bissegg	1341	de
4185	2755	Böztal	1213	NA

commune_id	population	name	cattle_population	language_region
4022	1544	Bellikon	281	de
6215	1923	Massongex	231	fr
5636	2906	Etoy	0	fr
4039	2023	Remetschwil	403	de
6238	2485	Grône	71	fr
1145	927	Ufhusen	1795	de

We calculate the number of cattle per 1000 inhabitants and store this as a new column

```
communes <- communes %>% mutate(cattle_per_1k = cattle_population/population *10^3)
```

We now created a table in the database for the communes with the unique `commune_id` as a primary key.

```
dbExecute(conn, "
  create table if not exists communes (
    commune_id int primary key,
    name text,
    population int,
    language_region text,
    cattle_population int,
    cattle_per_1k real
  )
")
```

```
## [1] 0
```

The joined commune data is then loaded into the database.

```
for (i in 1:nrow(communes)) {
  row = communes[i,]
  dbExecute(conn, "
            insert into communes (commune_id, name, population, language_region,
                                  cattle_population, cattle_per_1k)
            values (?, ?, ?, ?, ?,?)",
          params = c(row$commune_id, row$name,
                     row$population, row$language_region,
                     row$cattle_population, row$cattle_per_1k))
}
```

We check that this has worked properly by querying the `communes` table.

```
dbGetQuery(conn, "select * from communes limit 10") %>% kable()
```

commune_id	name	population	language_region	cattle_population	cattle_per_1k
3762	Scuol	4624	rt	2107	455.66609
1631	Glarus Süd	9480	de	4143	437.02532
3746	Zernez	1506	rt	962	638.77822
3543	Surses	2377	rt	1826	768.19520
6037	Val de Bagnes	10329	NA	2441	236.32491
3851	Davos	10832	de	1746	161.18907
3792	Bregaglia	1556	it	510	327.76350
6252	Anniviers	2742	fr	517	188.54850
6300	Zermatt	5820	de	54	9.27835
784	Innertkirchen	1072	de	765	713.61940

Family names table

We now turn to the family names. First we load the data from `family-names-commune.csv`, retaining only the top names in 2021.

```
family_names <-
  read_delim(
    "input_data/family-names-commune.csv",
    delim = ";",
    escape_double = FALSE,
    trim_ws = TRUE
  )
family_names <- family_names %>%
  filter(TIME_PERIOD == 2021) %>%
  filter(RANG_GDE <= 100) %>%
  select(LASTNAME, GDENR, RANG_GDE, VALUE) %>%
  rename(family_name = LASTNAME, commune_id = GDENR, name_rank = RANG_GDE, name_count = VALUE)

family_names %>% sample_n(10) %>% kable()
```

family_name	commune_id	name_rank	name_count
Rastoder	85	11	10
Gwerder	1213	46	16
Ablondi	1367	23	15
Mildner	4901	70	4
Boschung	2198	7	16
Bernasconi	5162	1	24
Kuhn	97	76	9
Luzi	3673	9	20
Mallet	5746	10	6
Fröhlich	1322	96	15

Next, we create a table `family_names` in the database with an automatically generated `id` as primary key, and `commune_id` as foreign key linking to the `communes` table.

```
dbExecute(conn, "
  create table if not exists family_names (
    id integer primary key,
    name text,
    commune_id int,
    name_count int,
    rank int,
    foreign key(commune_id) references communes(commune_id)
  )
")
```

```
## [1] 0
```

Then we load the data into the created table.

```
pb = txtProgressBar(min = 1, max = nrow(family_names), initial = 1)

for (i in 1:nrow(family_names)) {
  row = family_names[i,]
  dbExecute(conn, "
```

```
            insert into family_names (name, commune_id, name_count, rank)
            values (?, ?, ?, ?)",
          params = c(row$family_name, row$commune_id, row$name_count, row$name_rank))

  if (i %% 100 == 0) {
    setTxtProgressBar(pb,i)}
}
```

```
## ================================================================================
```

```
close(pb)
```

Finally, we check that it worked by querying.

```
dbGetQuery(conn, "select * from family_names limit 10") %>% kable()
```

id	name	commune_id	name_count	rank
1	Frei	1	14	1
2	Meier	1	14	1
3	Weiss	1	14	1
4	Keller	1	13	4
5	Steiner	1	12	5
6	Angst	1	11	6
7	Bachmann	1	11	6
8	Müller	1	11	6
9	Spinner	1	11	6
10	Graf	1	10	10

Calf names table

The last table contains the calf names. We load the data from `cattle-NamesFemaleCalves.csv` and `cattle-NamesMaleCalves.csv`.

```
cattle_NamesFemaleCalves <- read_delim("input_data/cattle-NamesFemaleCalves.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE) %>%
  mutate(sex="F")

cattle_NamesMaleCalves <- read_delim("input_data/cattle-NamesMaleCalves.csv",
    delim = ";", escape_double = FALSE, trim_ws = TRUE) %>%
  mutate(sex="M")

cattle_names <- bind_rows(cattle_NamesFemaleCalves, cattle_NamesMaleCalves)
rm(cattle_NamesFemaleCalves, cattle_NamesMaleCalves)

cattle_names <- cattle_names %>%
  filter(year==2022) %>%
  filter(OwnerLanguage != "__all__") %>%
  select(Name, count, Rank, sex, OwnerLanguage) %>%
  rename(name = Name, rank = Rank, language_region = OwnerLanguage)

cattle_names %>% arrange(rank, sex, language_region) %>% head(10) %>% kable()
```

name	count	rank	sex	language_region
Bella	674	1	F	de
Tulipe	88	1	F	fr
Luna	12	1	F	it
Max	633	1	M	de
Pvv	133	1	M	fr
Rambo	7	1	M	it
Sina	518	2	F	de
Bella	77	2	F	fr
Gina	10	2	F	it
Leo	545	2	M	de

We create the database table `cattle_names` using and automatically generated `id` as primary key, and `language_region` as a foreign key linking to the `communes` table.

```
dbExecute(conn, "
  create table if not exists cattle_names (
    id integer primary key,
    name text,
    count int,
    rank int,
    sex text,
    language_region text,
    foreign key(language_region) references communes(language_region)
  )
")
```

```
## [1] 0
```

We then load the data into the database and query to check if it worked.

```
for (i in 1:nrow(cattle_names)) {
  row = cattle_names[i,]
  dbExecute(conn, "
            insert into cattle_names (name, count, rank, sex, language_region)
            values (?, ?, ?, ?, ?)",
          params = c(row$name, row$count, row$rank, row$sex, row$language_region))
  }
```

```
dbGetQuery(conn, "select * from cattle_names limit 10") %>% kable()
```

id	name	count	rank	sex	language_region
1	Bella	674	1	F	de
2	Sina	518	2	F	de
3	Anna	464	3	F	de
4	Fiona	455	4	F	de
5	Tina	449	5	F	de
6	Olivia	448	6	F	de
7	Nora	446	7	F	de
8	Bianca	430	8	F	de
9	Nina	426	9	F	de
10	Luna	398	10	F	de

Querying the database

We now query the database to see which family names are most popular in communes with the highest cattle density per population.

We can get the communes with the highest cattle density per person as follows.

```
dbGetQuery(conn, "
           select *
           from communes
           order by cattle_per_1k desc
           limit 10
           ") %>% kable()
```

commune_id	name	population	language_region	cattle_population	cattle_per_1k
708	Schelten	39	de	278	7128.205
709	Seehof	56	de	389	6946.429
715	Rebévelier	41	de	279	6804.878
6433	Brot-Plamboz	290	fr	1571	5417.241
6759	Soubey	131	fr	682	5206.107
6432	La Brévine	624	fr	2985	4783.654
437	Mont-Tramelan	111	de	525	4729.730
2549	Kammersrohr	32	de	147	4593.750
2067	Le Châtelard	349	fr	1478	4234.957
6758	Saint-Brais	227	fr	940	4140.969

We see that the Scheltenites better treat their cattle with care, since in case of cow mutiny they are outnumbered 7:1.

Lets now get all communes that have more than 3 cattle per person. These are the cowboy communes.

```
dbGetQuery(conn, "
           select *
           from communes
           where cattle_per_1k > 3000
           order by cattle_per_1k desc
           ") %>% head(n = 10) %>% kable()
```

commune_id	name	population	language_region	cattle_population	cattle_per_1k
708	Schelten	39	de	278	7128.205
709	Seehof	56	de	389	6946.429
715	Rebévelier	41	de	279	6804.878
6433	Brot-Plamboz	290	fr	1571	5417.241
6759	Soubey	131	fr	682	5206.107
6432	La Brévine	624	fr	2985	4783.654
437	Mont-Tramelan	111	de	525	4729.730
2549	Kammersrohr	32	de	147	4593.750
2067	Le Châtelard	349	fr	1478	4234.957
6758	Saint-Brais	227	fr	940	4140.969

We can now plug the resulting `commune_id`s of the above query into `family_names` to get a list of the most frequent family names in these high-cow regions. These are the cowboy names.

```
dbGetQuery(conn, "
           select *
           from family_names
           where commune_id in
             (select commune_id
             from communes
             where cattle_per_1k > 3000)

           ") %>% sample_n(10) %>% kable()
```

id	name	commune_id	name_count	rank
216743	Monnerat	6719	7	15
209736	Givord	6511	4	30
219442	Lehmann	6808	7	27
117727	Nicca	3715	7	12
216741	Schaller	6719	8	12
209729	Eckert	6511	5	24
117737	Baumann	3715	3	21
37583	Cattin	715	4	2
219426	Berthold	6808	13	16
219469	Toro	6808	5	46

We can now aggregate this list by `name_count` to get the most frequent cowboy names. We print the top 30 cowboy names

```
dbGetQuery(conn, "
           select name, sum(name_count) name_count
           from family_names
           where commune_id in
             (select commune_id
             from communes
             where cattle_per_1k > 3000)
           group by name
           order by name_count desc

           ") %>% head(n=30) %>% kable()
```

name	name_count
Amstutz	82
Gerber	67
Frésard	61
Froidevaux	55
Girardin	52
Boillat	51
Marchand	44
Rey	42
Jeannerat	40
Oberson	38
Maire	38
Theurillat	36
Queloz	36
Odiet	36

name	name_count
Roch	35
Piquerez	33
Uldry	32
Rosselet-Christ	31
Pellaton	31
Choulat	31
Cantieni	31
Beuret	30
Willemin	29
Maitre	29
Donzé	29
Grolimund	28
Michael	27
Aubry	27
Gyger	26
Cattin	26

We observe that many of the names are french. Queloz is incidentilly the name of physics nobel laureate Didier Queloz whose research concerns the detection of earth-like exoplanets. Perhaps he is in part motivated to find new pastures for raising cattle.