Answering Business Questions using SQL

Eda AYDIN

19 05 2021

Contents

Creating Helper Functions

```
library(RSQLite)
```

```
## Warning: package 'RSQLite' was built under R version 4.0.5
```

```
library(DBI)
```

```
## Warning: package 'DBI' was built under R version 4.0.5
```

```
db <- "chinook.db"

run_query <- function(q) {
  conn <- dbConnect(SQLite(), db)
  result <- dbGetQuery(conn, q)
  dbDisconnect(conn)
  return(result)
}

show_tables <- function() {
  q = "SELECT name, type FROM sqlite_master WHERE type IN ('table', 'view')"
  return(run_query(q))
}

show_tables()
```

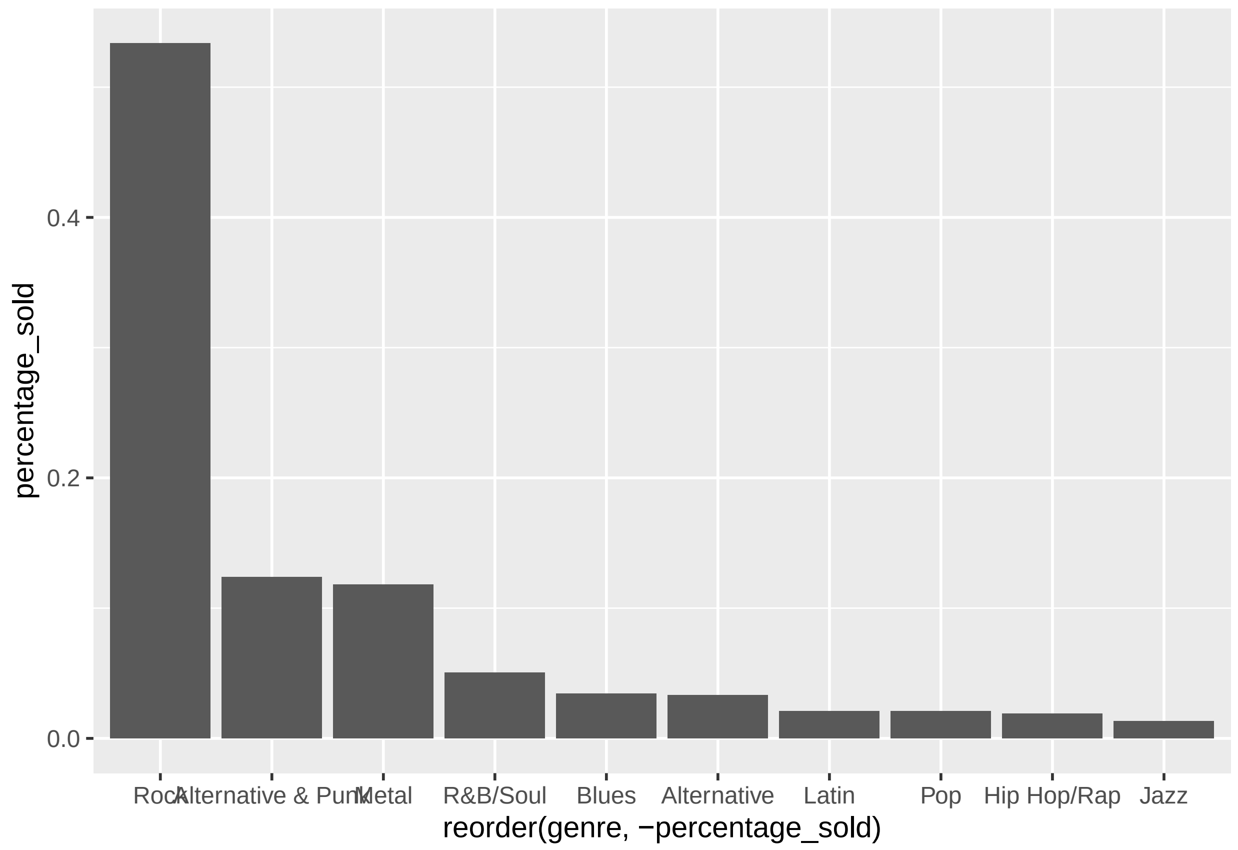
```
##               name  type
## 1            album table
## 2           artist table
## 3         customer table
## 4         employee table
## 5            genre table
## 6          invoice table
## 7     invoice_line table
## 8       media_type table
## 9         playlist table
## 10 playlist_track table
## 11           track table
```

Selecting New Albums to Purchase

```
albums_to_purchase = '
WITH usa_tracks_sold AS
   (
    SELECT il.* FROM invoice_line il
    INNER JOIN invoice i on il.invoice_id = i.invoice_id
    INNER JOIN customer c on i.customer_id = c.customer_id
    WHERE c.country = "USA"
   )
SELECT
    g.name genre,
    count(uts.invoice_line_id) tracks_sold,
    cast(count(uts.invoice_line_id) AS FLOAT) / (
        SELECT COUNT(*) from usa_tracks_sold
    ) percentage_sold
FROM usa_tracks_sold uts
INNER JOIN track t on t.track_id = uts.track_id
INNER JOIN genre g on g.genre_id = t.genre_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
'
run_query(albums_to_purchase)
```

```
##                 genre tracks_sold percentage_sold
## 1                Rock         561      0.53377735
## 2  Alternative & Punk         130      0.12369172
## 3               Metal         124      0.11798287
## 4            R&B/Soul          53      0.05042816
## 5               Blues          36      0.03425309
## 6         Alternative          35      0.03330162
## 7                 Pop          22      0.02093245
## 8               Latin          22      0.02093245
## 9         Hip Hop/Rap          20      0.01902950
## 10               Jazz          14      0.01332065
```

```
library(ggplot2)
genre_sales = run_query(albums_to_purchase)
ggplot(data = genre_sales, aes(x = reorder(genre, -percentage_sold),
                               y = percentage_sold)) +
  geom_bar(stat = "identity")
```

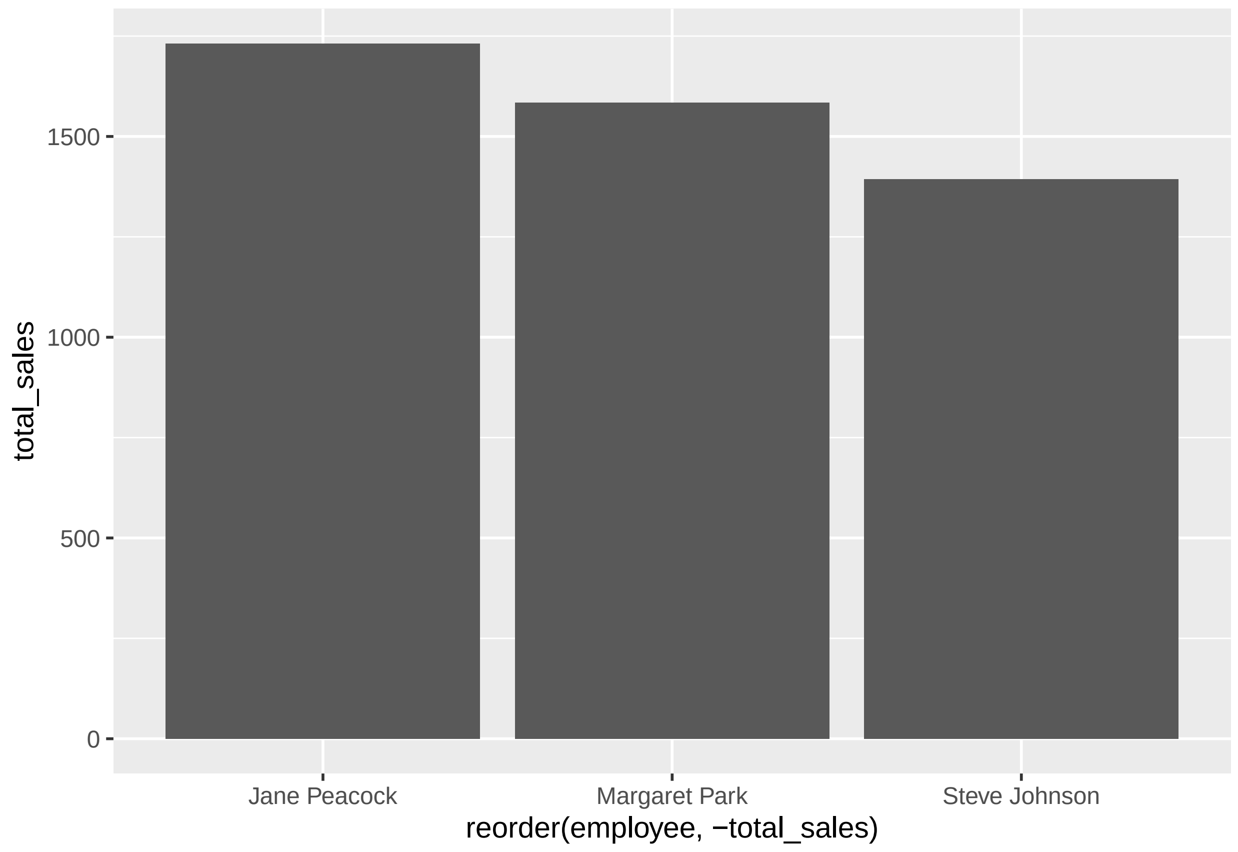


Analyzing Employee Sales Performance

```
employee_sales_performance = '
WITH customer_support_rep_sales AS
    (
     SELECT
         i.customer_id,
         c.support_rep_id,
         SUM(i.total) total
     FROM invoice i
     INNER JOIN customer c ON i.customer_id = c.customer_id
     GROUP BY 1,2
    )
SELECT
    e.first_name || " " || e.last_name employee,
    e.hire_date,
    SUM(csrs.total) total_sales
FROM customer_support_rep_sales csrs
INNER JOIN employee e ON e.employee_id = csrs.support_rep_id
GROUP BY 1;
'
run_query(employee_sales_performance)
```

```
##        employee           hire_date total_sales
## 1  Jane Peacock 2017-04-01 00:00:00     1731.51
## 2 Margaret Park 2017-05-03 00:00:00     1584.00
## 3 Steve Johnson 2017-10-17 00:00:00     1393.92
```

```
employee_sales = run_query(employee_sales_performance)
ggplot(data = employee_sales, aes(x = reorder(employee, -total_sales),
                                 y = total_sales)) +
  geom_bar(stat = "identity")
```

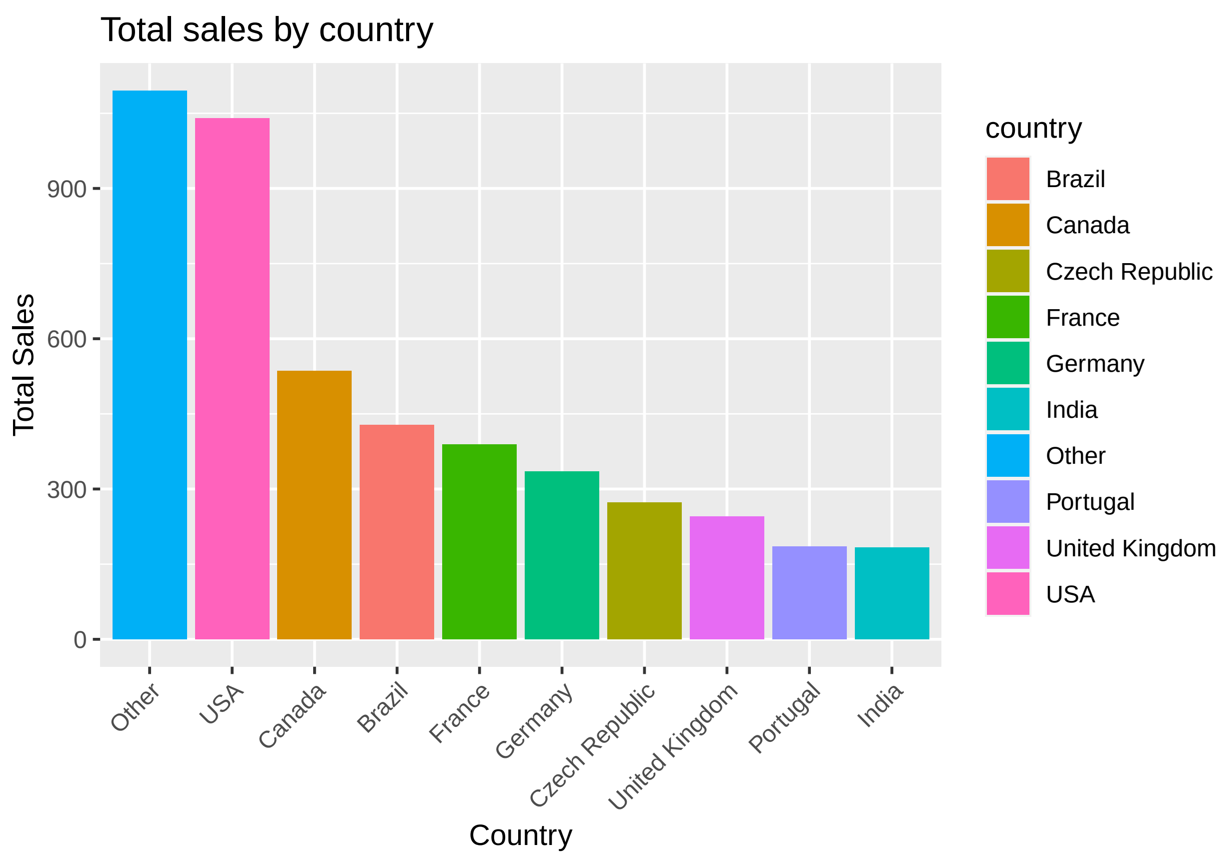


```
sales_by_country = '
WITH country_or_other AS
    (
     SELECT
       CASE
           WHEN (
                 SELECT count(*)
                 FROM customer
                 where country = c.country
                ) = 1 THEN "Other"
           ELSE c.country
       END AS country,
       c.customer_id,
       il.*
     FROM invoice_line il
     INNER JOIN invoice i ON i.invoice_id = il.invoice_id
     INNER JOIN customer c ON c.customer_id = i.customer_id
    )
SELECT
    country,
    customers,
    total_sales,
    average_order,
    customer_lifetime_value
FROM
    (
    SELECT
        country,
        count(distinct customer_id) customers,
        SUM(unit_price) total_sales,
        SUM(unit_price) / count(distinct customer_id) customer_lifetime_value,
        SUM(unit_price) / count(distinct invoice_id) average_order,
        CASE
            WHEN country = "Other" THEN 1
            ELSE 0
        END AS sort
    FROM country_or_other
    GROUP BY country
    ORDER BY sort ASC, total_sales DESC
    );
'
run_query(sales_by_country)
```

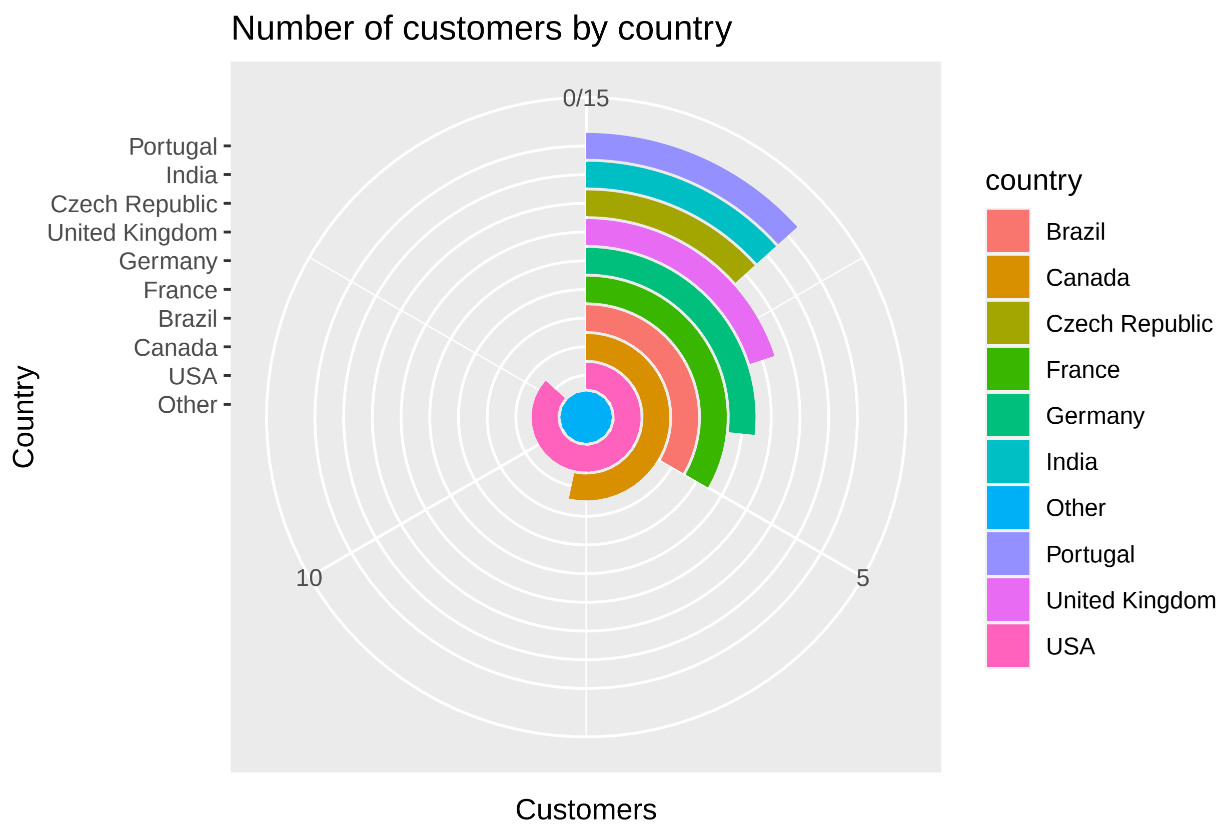
```
##           country customers total_sales average_order customer_lifetime_value
## 1             USA        13     1040.49      7.942672                80.03769
## 2          Canada         8      535.59      7.047237                66.94875
## 3          Brazil         5      427.68      7.011148                85.53600
## 4          France         5      389.07      7.781400                77.81400
## 5         Germany         4      334.62      8.161463                83.65500
## 6  Czech Republic         2      273.24      9.108000               136.62000
## 7  United Kingdom         3      245.52      8.768571                81.84000
## 8        Portugal         2      185.13      6.383793                92.56500
## 9           India         2      183.15      8.721429                91.57500
## 10          Other        15     1094.94      7.448571                72.99600
```

Visualizing Sales by Country

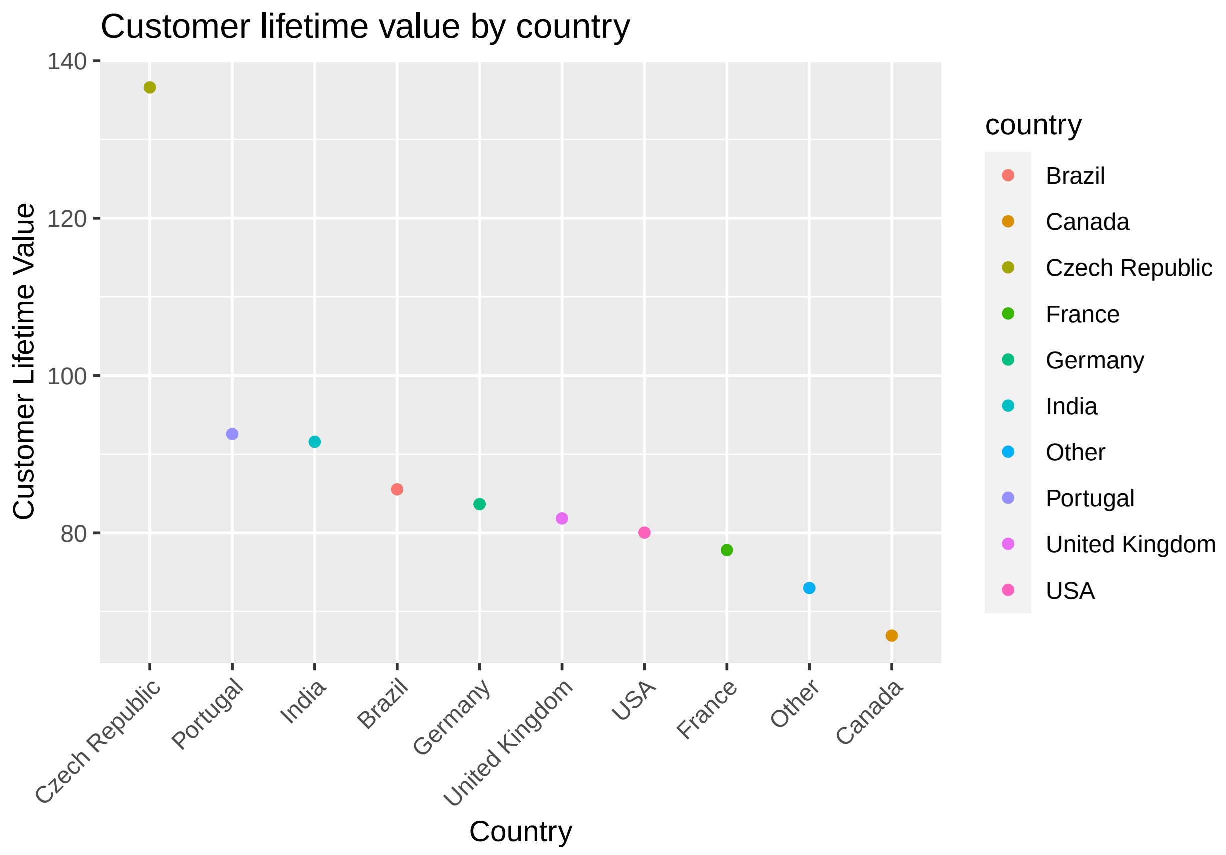
```r
country_metrics = run_query(sales_by_country)
ggplot(data = country_metrics, aes(x = reorder(country, -total_sales),
                                   y = total_sales,
                                   fill = country)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Total sales by country",
    x = "Country",
    y = "Total Sales"
  ) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
ggplot(data = country_metrics, aes(x = reorder(country, -customers),
                                   y = customers,
                                   fill = country)) +
  geom_bar(stat = "identity") +
  coord_polar("y") +
  labs(
    title = "Number of customers by country",
    x = "Country",
    y = "Customers"
  )
```



```
ggplot(data = country_metrics, aes(x = reorder(country, -customer_lifetime_value),
                                   y = customer_lifetime_value,
                                   color = country)) +
  geom_point(stat = "identity") +
  labs(
    title = "Customer lifetime value by country",
    x = "Country",
    y = "Customer Lifetime Value"
  ) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Albums vs Individual Tracks

```
albums_vs_tracks = '
WITH invoice_first_track AS
    (
     SELECT
         il.invoice_id invoice_id,
         MIN(il.track_id) first_track_id
     FROM invoice_line il
     GROUP BY 1
    )
SELECT
    album_purchase,
    COUNT(invoice_id) number_of_invoices,
    CAST(count(invoice_id) AS FLOAT) / (
                                         SELECT COUNT(*) FROM invoice
                                      ) percent
FROM
    (
    SELECT
        ifs.*,
        CASE
            WHEN
```

```
                 (
                  SELECT t.track_id FROM track t
                  WHERE t.album_id = (
                                      SELECT t2.album_id FROM track t2
                                      WHERE t2.track_id = ifs.first_track_id
                                     )
                  EXCEPT
                  SELECT il2.track_id FROM invoice_line il2
                  WHERE il2.invoice_id = ifs.invoice_id
                 ) IS NULL
             AND
                 (
                  SELECT il2.track_id FROM invoice_line il2
                  WHERE il2.invoice_id = ifs.invoice_id
                  EXCEPT
                  SELECT t.track_id FROM track t
                  WHERE t.album_id = (
                                      SELECT t2.album_id FROM track t2
                                      WHERE t2.track_id = ifs.first_track_id
                                     )
                 ) IS NULL
             THEN "yes"
             ELSE "no"
         END AS "album_purchase"
     FROM invoice_first_track ifs
    )
GROUP BY album_purchase;
'
run_query(albums_vs_tracks)
```

```
##   album_purchase number_of_invoices   percent
## 1             no                500 0.8143322
## 2            yes                114 0.1856678
```