local bakery data analysis

Yufenyuy

2025-07-17

setwd("C:/gitrepos/ranalytics/r\_data\_analysis")

**Hier werden R-Packete für die Daten Manipulation bzw. Auswertung beladen**

library(conflicted)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2

library(DBI)

## Warning: Paket 'DBI' wurde unter R Version 4.4.3 erstellt

library(RPostgres)

## Warning: Paket 'RPostgres' wurde unter R Version 4.4.3 erstellt

library(ggplot2)  
library(lubridate)  
library(dplyr)  
library(tidyr)  
library(stringr)

**Erstellte Umgebung Variablen werden für die Datenbank Verbindung eingelesen**

con <- dbConnect(  
 Postgres(),  
 dbname = Sys.getenv("DBNAME"),  
 host = Sys.getenv("HOST"),   
 user = Sys.getenv("USER"),  
 password = Sys.getenv("PASSWORD"),  
 port = Sys.getenv("PORT")  
)

**Die gezielte Daten Tabelle liegt in einer spezifisches Schema in PostgreSQL. Die Tabellen enhalten in diesem Schema werden aufgelisted und geprüft, ob die gewünschte Tabelle drin liegt**

schema\_name <- "baker\_yelp\_dbt\_prod"  
  
# SQL query to list tables in the given schema  
tables <- dbGetQuery(con, paste0("  
 SELECT table\_name  
 FROM information\_schema.tables  
 WHERE table\_schema = '", schema\_name, "'  
 AND table\_type = 'BASE TABLE';  
"))  
  
print(tables)

## table\_name  
## 1 pdtn\_fuel  
## 2 monthly\_ts\_issales  
## 3 weekly\_product\_pdtn\_ts  
## 4 sales\_t1  
## 5 monthly\_ts\_should\_sales  
## 6 products\_weekly\_ts  
## 7 pdtn\_t1  
## 8 feeding  
## 9 sales\_t2  
## 10 business\_reviews\_t1  
## 11 weekly\_timeseries\_issales  
## 12 pdtn\_oldstuck  
## 13 business\_reviews\_t2  
## 14 weekly\_ts\_is\_should\_sales  
## 15 businesses\_t1  
## 16 weekly\_ts\_should\_sales  
## 17 businesses\_t2  
## 18 daily\_expected\_production\_dates  
## 19 daily\_product\_pdtn\_amt  
## 20 businesses\_t3  
## 21 monthly\_ts\_is\_should\_sales  
## 22 items  
## 23 datetable

**Hier werden die Daten mittels SQL selektiert**

product\_ts <- dbGetQuery(con, "SELECT \* FROM baker\_yelp\_dbt\_prod.products\_weekly\_ts")

**Datentypen und Werte einsehen**

str(product\_ts)

## 'data.frame': 425 obs. of 15 variables:  
## $ endofweek : Date, format: "2023-07-09" "2023-07-02" ...  
## $ banana50\_amt : num 9 3 0 0 0 0 0 0 0 0 ...  
## $ square50\_amt : num 148 142 124 136 148 ...  
## $ local50\_amt : num 7.5 12.5 17.5 17.5 22.5 22.5 25 15 30 30 ...  
## $ banana100\_amt : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ local100\_amt : num 190 185 177 201 190 ...  
## $ special100\_amt: num 675 1732 1674 1752 1657 ...  
## $ special150\_amt: num 1319 128 149 158 163 ...  
## $ local200\_amt : num 253 244 250 234 269 ...  
## $ special200\_amt: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ local250\_amt : num 50.4 50.4 42 50.4 49 42 40.6 33.6 42 50.4 ...  
## $ local300\_amt : num 569 519 524 580 543 ...  
## $ special400\_amt: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ special500\_amt: num 706 624 623 697 630 ...  
## $ special800\_amt: num 60.5 21.6 32.7 68.7 44.1 ...

**Statistik der Daten einsehen, um die Zentrale Tendenz und die Streuung der Daten zu verstehen.**

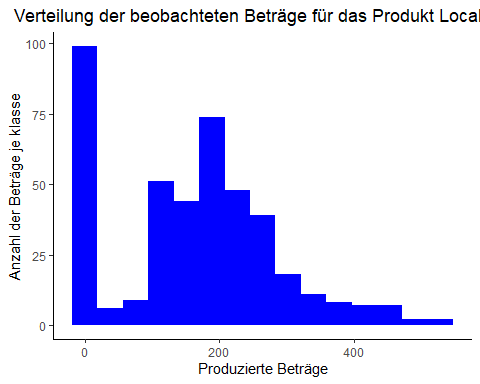
*Beobachtungen:*

* Die Daten wurden von mitte 2015 bis mitte 2023 gesammelt.
* Bei viele Produkte(spezial150\_amt) liegt der mittleren Wert bei 0. D.H Bis zu 50% ihre Beträge liegen bei 0 Geld Einheiten(GE).
* Der minimale produzierte Betrag bei alle Produkte liegt bei 0 GE.
* Der höchste durchschnittliche produzierte Betrag über den ganzen Zeitraum ist bei dem Produkt special200\_amt beobachtet und dasselbe Produkt hat der höchste produzierte Betrag unter alle Produkten.Allerding liegt sein Mittelwert bei 0 GE.
* Bei der Produkte local100\_amt, local300\_amt und special100\_amt liegen der Mittelwert und der Durchschnitt nah beieinander.Diese weißt auf einer quasi Normal Verteilung hin spricht die Daten sind quasi gleichverteilt.

summary(product\_ts)

## endofweek banana50\_amt square50\_amt local50\_amt   
## Min. :2015-05-24 Min. : 0.0 Min. : 0.00 Min. : 0.000   
## 1st Qu.:2017-06-04 1st Qu.: 0.0 1st Qu.: 0.00 1st Qu.: 0.000   
## Median :2019-06-16 Median : 48.0 Median : 0.00 Median : 0.000   
## Mean :2019-06-16 Mean :135.8 Mean : 66.88 Mean : 1.688   
## 3rd Qu.:2021-06-27 3rd Qu.:276.0 3rd Qu.:135.00 3rd Qu.: 0.000   
## Max. :2023-07-09 Max. :542.4 Max. :450.00 Max. :40.000   
## banana100\_amt local100\_amt special100\_amt special150\_amt   
## Min. : 0.000 Min. : 0.0 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 0.000 1st Qu.: 66.3 1st Qu.: 196.3 1st Qu.: 0.0   
## Median : 0.000 Median :170.0 Median : 973.5 Median : 0.0   
## Mean : 7.916 Mean :161.9 Mean : 889.5 Mean : 164.8   
## 3rd Qu.: 0.000 3rd Qu.:238.0 3rd Qu.:1440.5 3rd Qu.: 301.0   
## Max. :107.500 Max. :528.7 Max. :2352.9 Max. :1318.6   
## local200\_amt special200\_amt local250\_amt local300\_amt   
## Min. : 0.0 Min. : 0.000 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 0.0 1st Qu.: 0.000 1st Qu.: 15.4 1st Qu.: 72.1   
## Median : 93.2 Median : 0.000 Median : 56.0 Median :349.3   
## Mean :113.5 Mean : 8.179 Mean :143.2 Mean :319.5   
## 3rd Qu.:231.6 3rd Qu.: 0.000 3rd Qu.:237.3 3rd Qu.:518.7   
## Max. :351.6 Max. :134.940 Max. :607.6 Max. :777.7   
## special400\_amt special500\_amt special800\_amt   
## Min. : 0.00 Min. : 0.0 Min. : 0.00   
## 1st Qu.: 0.00 1st Qu.: 100.4 1st Qu.: 2.40   
## Median : 0.00 Median : 435.2 Median : 19.20   
## Mean : 13.17 Mean : 384.1 Mean : 25.08   
## 3rd Qu.: 0.00 3rd Qu.: 581.0 3rd Qu.: 38.40   
## Max. :534.48 Max. :1004.7 Max. :110.40

g1 <- product\_ts %>%  
 ggplot(aes(x = local100\_amt)) +  
 geom\_histogram(fill = "blue", bins = 15) +  
 xlab("Produzierte Beträge") +  
 ylab("Anzahl der Beträge je klasse") +  
 theme\_classic() +  
 ggtitle("Verteilung der beobachteten Beträge für das Produkt Local100") +  
 theme(  
 plot.title = element\_text(hjust = 0.5)  
 )  
g1



product\_ts <- product\_ts %>% pivot\_longer(cols = ends\_with("amt"), names\_to = "products", values\_to = "weekly\_amount", names\_repair = "check\_unique")

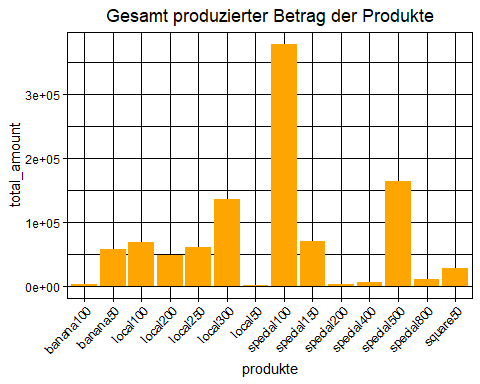
str(product\_ts)

## tibble [5,950 × 3] (S3: tbl\_df/tbl/data.frame)  
## $ endofweek : Date[1:5950], format: "2023-07-09" "2023-07-09" ...  
## $ products : chr [1:5950] "banana50\_amt" "square50\_amt" "local50\_amt" "banana100\_amt" ...  
## $ weekly\_amount: num [1:5950] 9 148.5 7.5 0 190.4 ...

summary(product\_ts)

## endofweek products weekly\_amount   
## Min. :2015-05-24 Length:5950 Min. : 0.0   
## 1st Qu.:2017-06-04 Class :character 1st Qu.: 0.0   
## Median :2019-06-16 Mode :character Median : 13.8   
## Mean :2019-06-16 Mean : 174.0   
## 3rd Qu.:2021-06-27 3rd Qu.: 229.9   
## Max. :2023-07-09 Max. :2352.9

gplot\_1 <- product\_ts %>%   
 mutate(  
 produkte = str\_sub(products, 1, str\_length(products) - 4)  
 ) %>%  
 group\_by(produkte) %>%  
 summarise(total\_amount = sum(weekly\_amount)) %>%  
 ggplot(aes(x = produkte, y = total\_amount)) +   
 geom\_col(fill = "orange") +  
 ggtitle("Gesamt produzierter Betrag der Produkte") +  
 theme\_linedraw() +  
 theme(  
 plot.title = element\_text(hjust = 0.5),  
 axis.text.x = element\_text(angle = 45, hjust = 1)  
 )  
gplot\_1



gplot\_2 <- dplyr::filter(product\_ts, products == "special100\_amt") %>%  
 ggplot(aes(x = endofweek, y = weekly\_amount)) +   
 geom\_line(color = "blue") +  
 xlab("Ende der Woche") +  
 ylab("Produzierter Betrag") +  
 ggtitle("Zeitliche Entwicklung des Meist produzierten Produkt") +  
 theme\_classic() +  
 theme(plot.title = element\_text(hjust = 0.5))  
gplot\_2

