Instacart

19 de outubro de 2023

R-Code

Versão 1.0

https://www.kaggle.com/competitions/instacart-market-basket-analysis

Libraries

Configuração de diretório de arquivos, carregar bibliotecas e desativar *warnings* em geral.

```
### Preparando o ambiente de trabalho ----------------------------------------
# Configurando o diretório de trabalho
setwd("/home/formiga/Desktop/Projetos/10")
getwd()

# Libraries
library(data.table)
library(dplyr)
library(plyr)
library(ggplot2)
library(ggcorrplot)
library(reshape2)
library(tidyr)
library(wordcloud)
library(shiny)
library(fBasics)
library(beepr)
library(DataExplorer)
library(scales)
library(knitr)

options(warn = 0) # 0 é sim, -1 é não
```

Juntando Dataframes

Juntando todos os *dataframes* em um só para facilitar parte do código.

```
### Juntando os dataframes ---------------------------------------------------
# Carregando o dataframe orders
orders <- fread('dados/orders.csv')
dim(orders_data_df)
str(orders_data_df)
summary(orders_data_df)
View(orders_data_df)

# Carregando o dataframe orders_products
```

```
orders_products_prior <- fread('dados/order_products__prior.csv')
dim(orders_products_df)
str(orders_products_df)
summary(orders_products_df)
View(orders_products_df)

# Combinando os últimos 2 dataframes acima
?join
df_geral_1 <- join(orders_data_df, orders_products_df)
dim(df_geral_1)
str(df_geral_1)
View(df_geral_1)

# Carregando o dataframe products
products <- fread('dados/products.csv')
dim(products_df)
str(products_df)
summary(products_df)
View(products_df)

# Combinando os últimos 2 dataframes acima
df_geral_2 <- join(df_geral_1, products_df)
dim(df_geral_2)
str(df_geral_2)
View(df_geral_2)

# Carregando o dataframe departamento
departments <- fread('dados/departments.csv')
dim(departments_df)
str(departments_df)
View(departments_df)

# Combinando os últimos 2 dataframes acima
df_geral_3 <- join(df_geral_2, departments_df)
beep(2)
```

```
### Salvando os dataframes em completo, prior, train e test
# Salvando o dataframe completo
write.csv(df_geral_3,file='/home/formiga/Desktop/Projetos/10/dados/df_completo.csv
')
beep(2)

# Separando os dados em prior e apagando coluna eval_set.
df_prior <- subset(df_geral_3, eval_set == 'prior')
dim(df_prior)
View(df_prior)
```

```
df_prior <- df_prior[,-3]
dim(df_prior)
View(df_prior)
write.csv(df_prior,file='/home/formiga/Desktop/Projetos/10/dados/df_prior.csv')
%>% beep(2)

# Separando os dados em train apagando colunas do tipo factor.
df_train <- subset(df_geral_3, eval_set == 'train')
dim(df_train)
View(df_train)
df_train <- select(df_train, c(-eval_set,-product_name,-department))
dim(df_train)
View(df_train)
write.csv(df_train,file='/home/formiga/Desktop/Projetos/10/dados/df_train.csv')
%>% beep(2)

# Separando os dados em test apagando colunas do tipo factor.
df_test <- subset(df_geral_3, eval_set == 'test')
dim(df_test)
View(df_test)
df_test <- select(df_test, c(-eval_set,-product_name,-department))
dim(df_test)
View(df_test)
write.csv(df_test,file='/home/formiga/Desktop/Projetos/10/dados/df_test.csv')


# Limpando as variáveis de ambiente e abrindo apenas o dataframe completo para
maior desempenho
rm(list=ls())
gc()
```

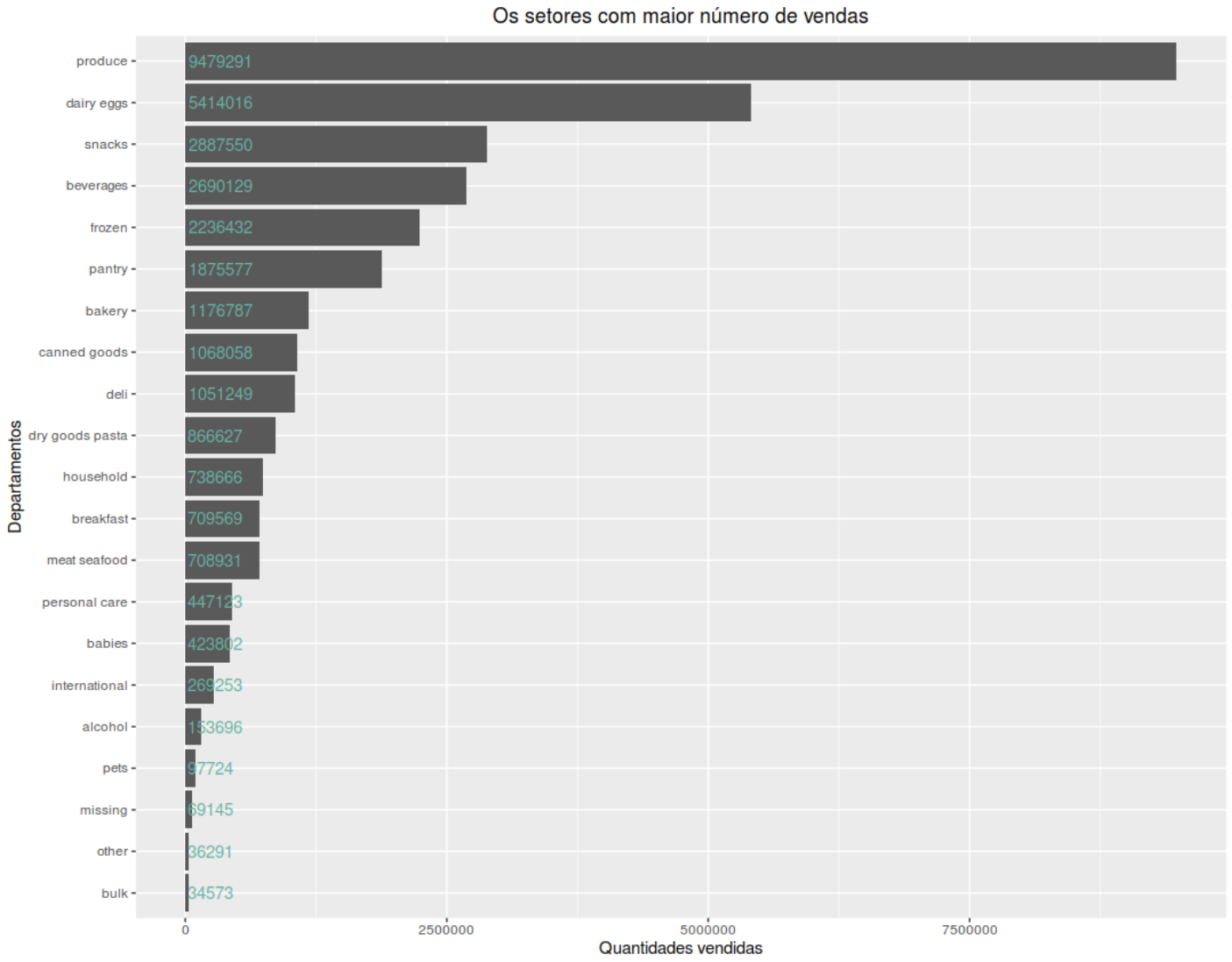
Checkpoint

```
### Checkpoint ------------------------------------------------------------------
df_prior <- fread('dados/df_prior.csv')

# Convertendo todas as colunas char em factor
df_prior <- mutate_if(df_prior, is.character, as.factor)

# Convertendo o data.table em data.frame para melhor manipulação dos dados
df_prior <- data.frame(df_prior)

# Resumo dos dados
dim(df_prior)
str(df_prior)
summary(df_prior)
colSums(is.na(df_prior))
```

```
View(df_prior)
beep(2)
```

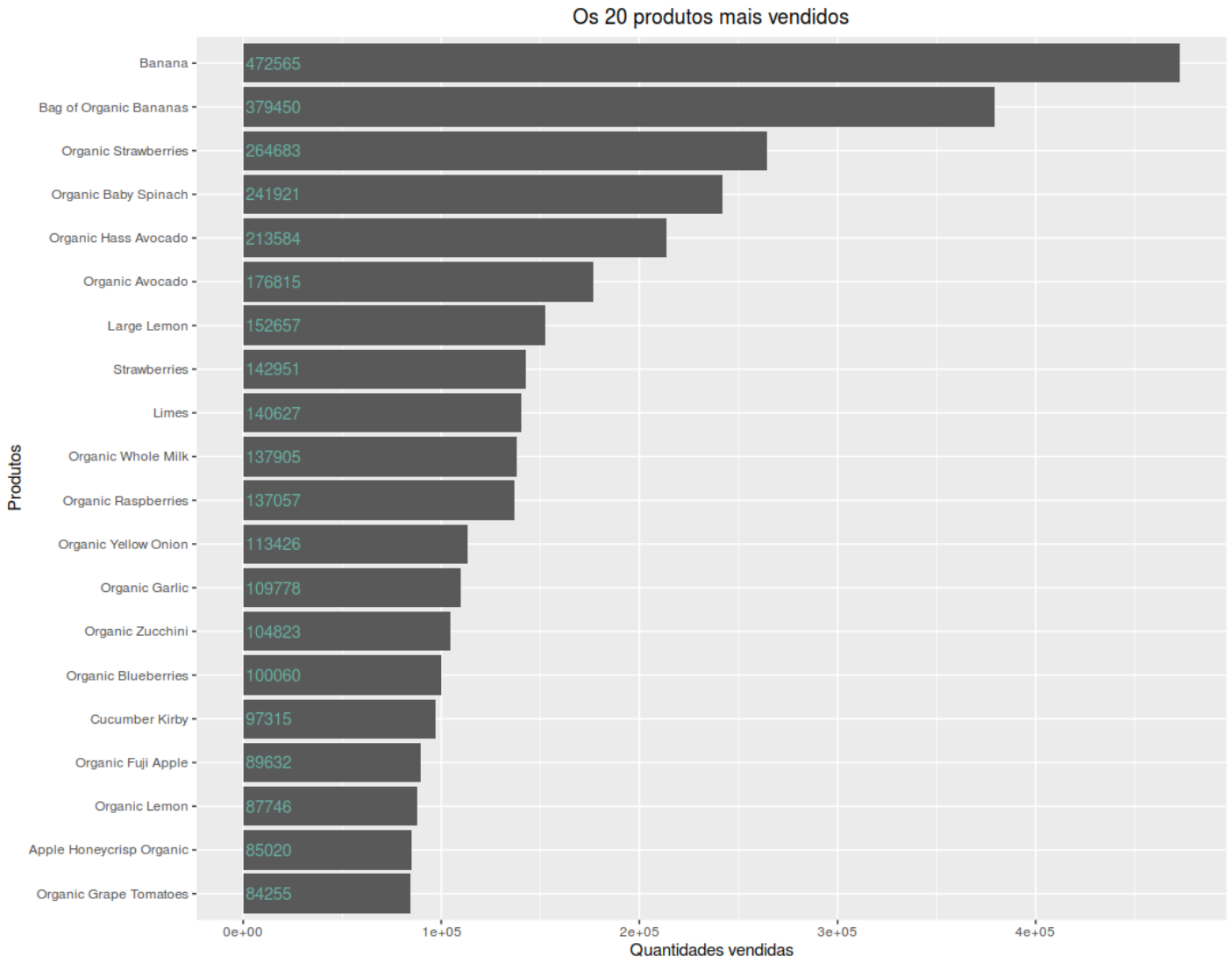
Visualização de gráficos sobre os dados

```
### Visualizando alguns gráficos sobre os dados -------------------------------
options(warn = -1)

# Descobrindo os setores com maior quantidade de produtos vendidos
tab_department <- table(df_prior$department)
sorted_tab_department <- tab_department %>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  top_n(30)

ggplot(sorted_tab_department, aes(x=reorder(Var1, Freq,
                                  function(x) sum(x)),
                                  y = Freq)) +
      geom_col() + coord_flip() + theme(plot.title = element_text(hjust = 0.5)) +
      geom_text(aes(label = Freq), y = 0, hjust = -0.05, vjust = 0.5, colour =
"#69b3a2") +
      xlab("Departamentos") + ylab("Quantidades vendidas") + ggtitle("Os setores
com maior número de vendas")
```



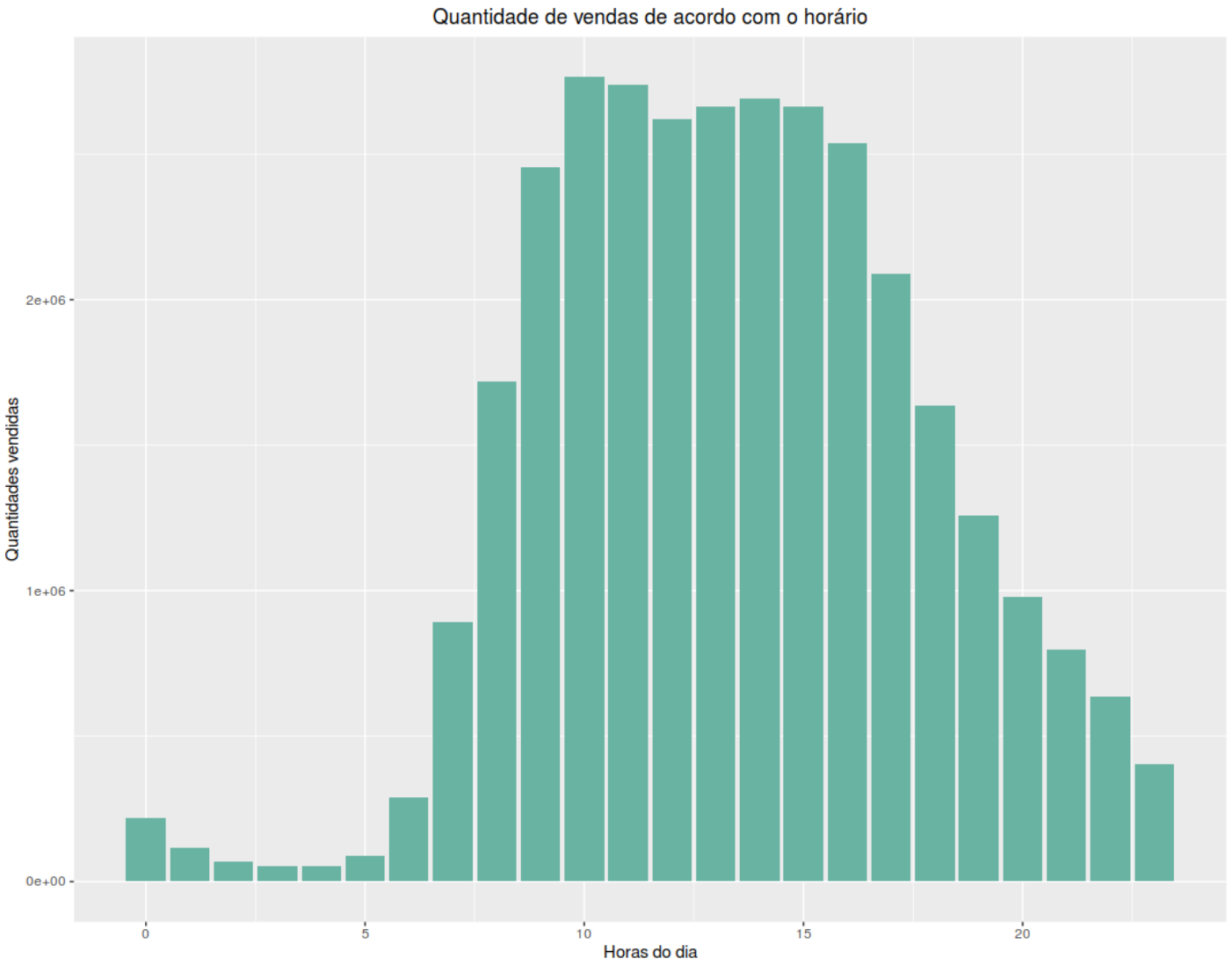
2.

```
# Descobrindo os 20 produtos mais vendidos
tab_product <- table(df_prior$product_name)
sorted_tab_product <- tab_product %>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  top_n(20)

ggplot(sorted_tab_product, aes(x = reorder(Var1, Freq,
                               function(x) sum(x)),
                               y = Freq)) +
       geom_col() + coord_flip() + theme(plot.title = element_text(hjust = 0.5)) +
       geom_text(aes(label = Freq), y = 0, hjust = -0.05, vjust = 0.5, colour =
"#69b3a2") +
       xlab("Produtos") + ylab("Quantidades vendidas") + ggtitle("Os 20 produtos
mais vendidos")
```



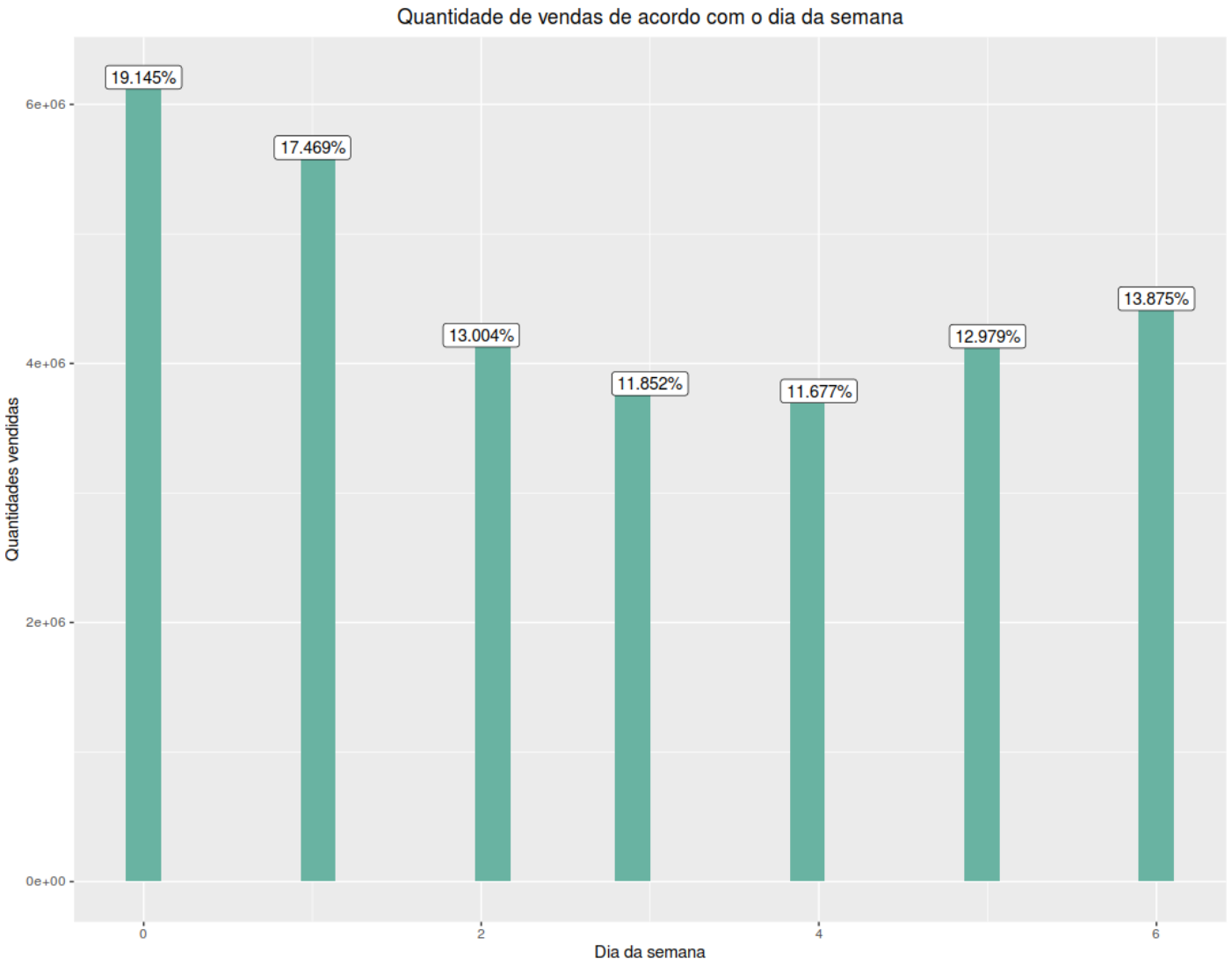
3.

```
# Vendas de acordo com o horário do dia
df_prior %>%
  ggplot (aes (x=order_hour_of_day)) +
  geom_histogram(stat="count",fill="#69b3a2") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Horas do dia") + ylab("Quantidades vendidas") + ggtitle("Quantidade de
vendas de acordo com o horário")
```



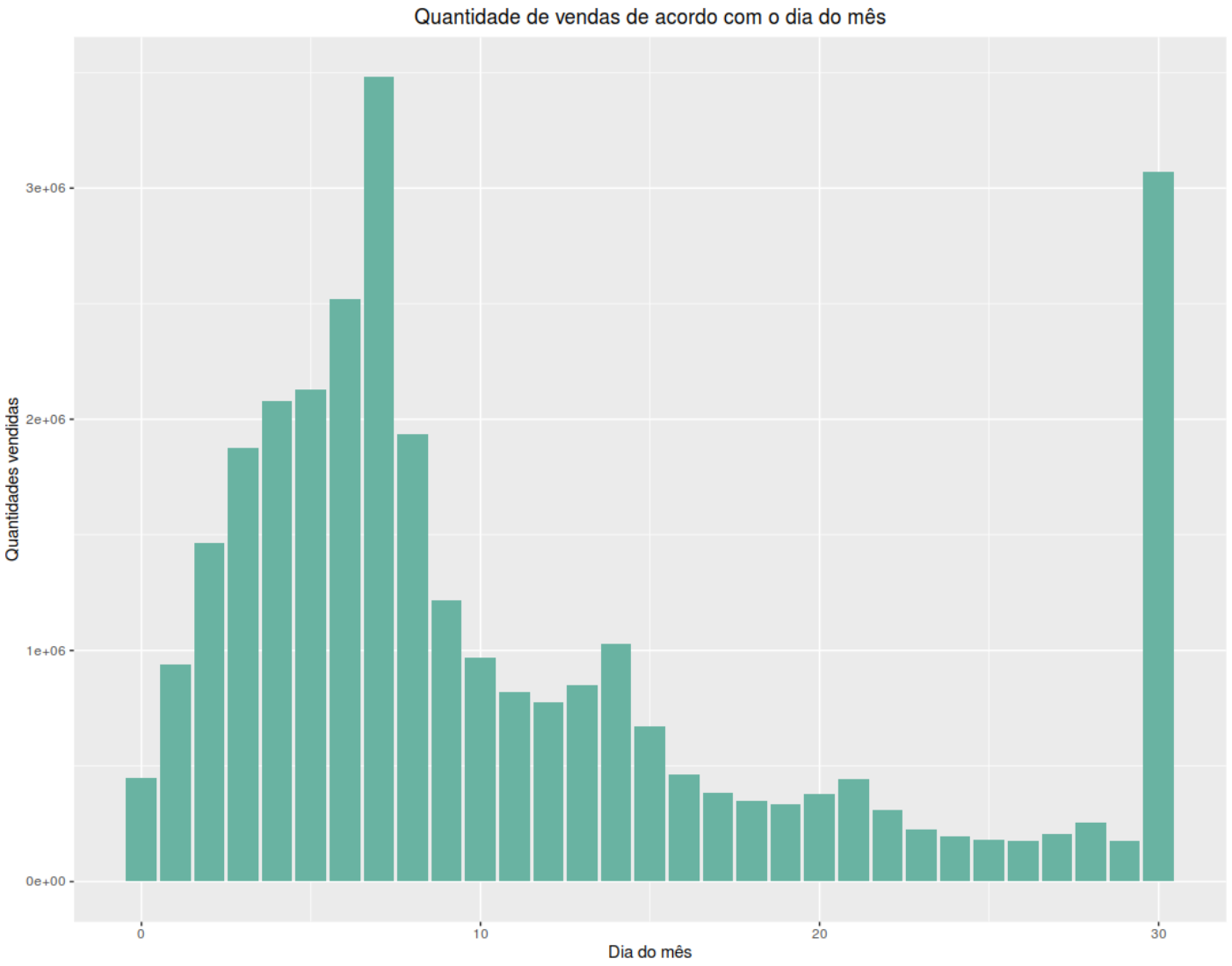
4.

```
# Vendas de acordo com o dia da semana
df_prior %>%
  ggplot (aes(x=order_dow)) +
  geom_histogram(fill="#69b3a2") +
  geom_label(aes(label = scales::percent(..count.. / sum(..count..))),
             stat = "count") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Dia da semana") + ylab("Quantidades vendidas") + ggtitle("Quantidade de
vendas de acordo com o dia da semana")
```



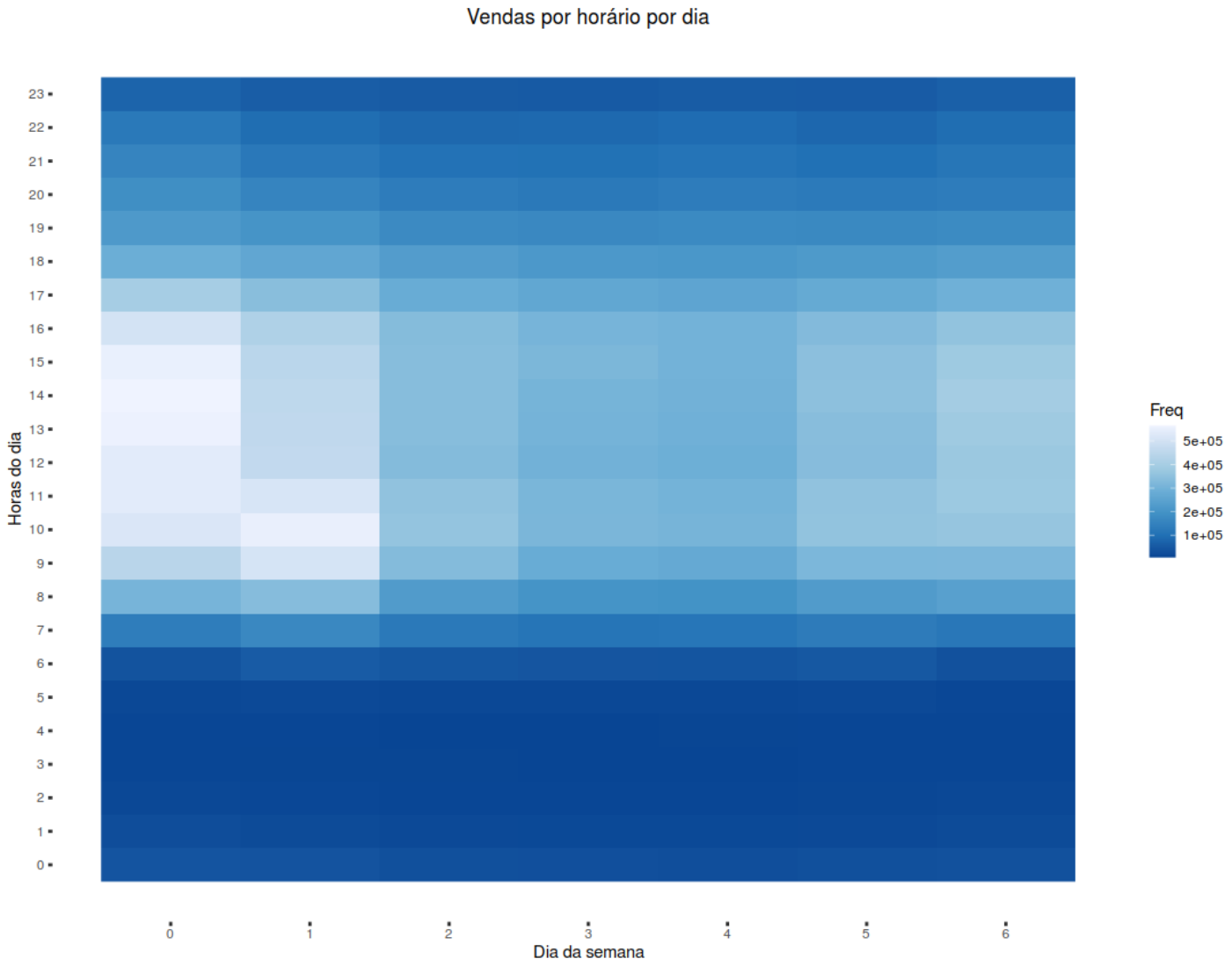
5.

```
# Vendas de acordo com o dia do mês
df_prior %>%
  ggplot(aes(x=days_since_prior_order)) +
  geom_histogram(stat="count",fill="#69b3a2") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Dia do mês") + ylab("Quantidades vendidas") + ggtitle("Quantidade de
vendas de acordo com o dia do mês")
```



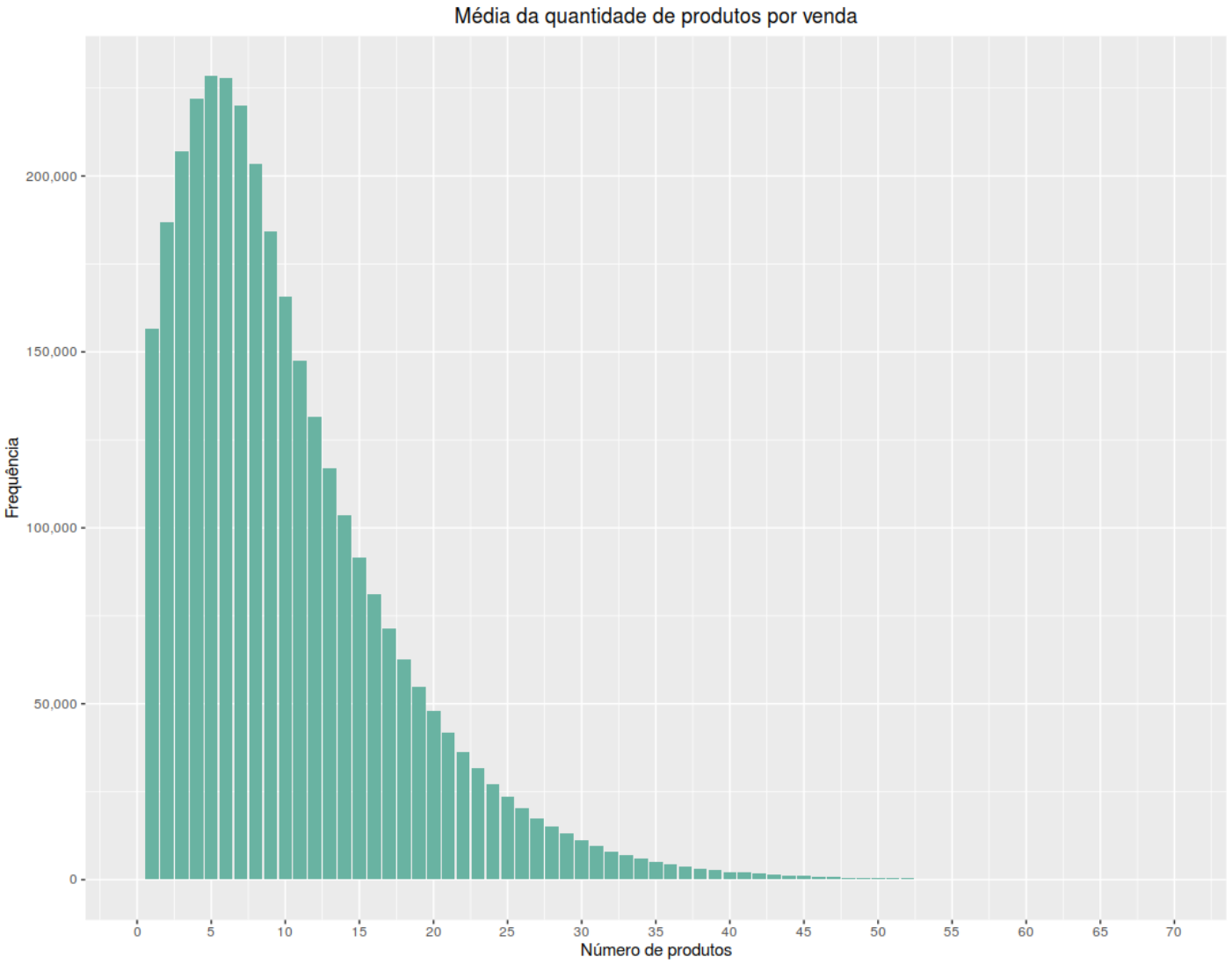
6.

```
# Vendas por horário VS dia
tmp <- df_prior %>% dplyr::group_by(order_dow, order_hour_of_day) %>%
  dplyr::summarise(Freq = n())

ggplot(tmp, aes(x = order_dow, y = order_hour_of_day, fill = Freq))+
  geom_raster()+
  scale_fill_distiller()+
  scale_x_continuous(breaks = c(0:6))+
  scale_y_continuous(breaks = c(0:23))+
  theme(panel.background = element_blank())+
  theme(axis.ticks = element_line(size = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Dia da semana") + ylab("Horas do dia") + ggtitle("Vendas por horário por
dia")
```



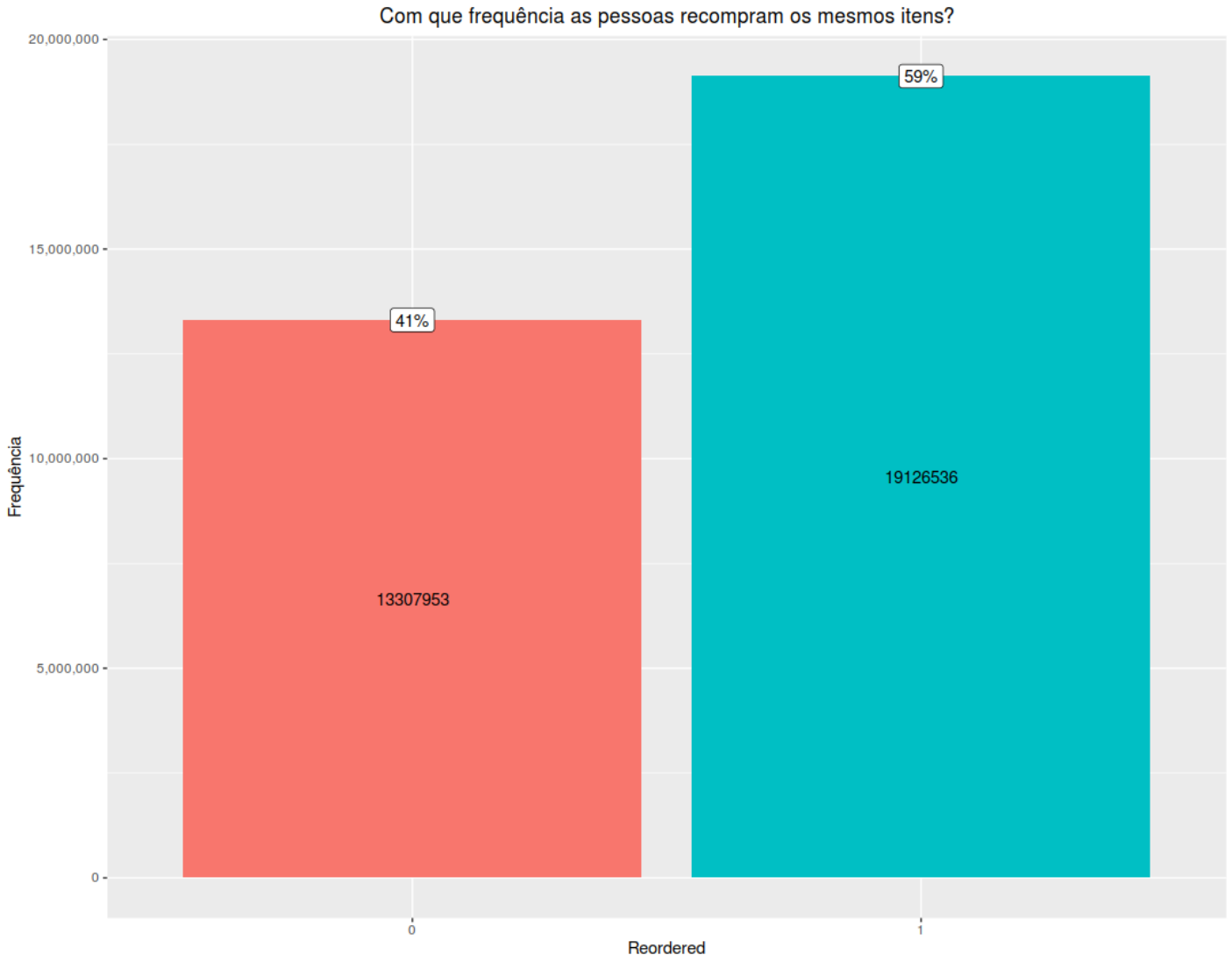
7.

```
# Quantidade de itens no carrinho
tmp <- df_prior %>%
  dplyr::group_by(order_id) %>%
  dplyr::summarise(n_products = last(add_to_cart_order))

tmp %>%
  dplyr::summarise(Min = min(n_products),
                   Max = max(n_products))

ggplot(tmp, aes(x = n_products)) +
  geom_histogram(stat = "count", fill = "#69b3a2") +
  scale_x_continuous(breaks = seq(0, 70, 5)) +
  scale_y_continuous(labels = comma) +
  coord_cartesian(xlim=c(0,70)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Número de produtos") + ylab("Frequência") + ggtitle("Média da quantidade
de produtos por venda")
```



8.

```
# Quantidade de reorders
ggplot(df_prior, aes(x = as.factor(reordered))) +
  geom_bar(aes(fill = as.factor(reordered)),stat = "count") +
  geom_text(aes(label = ..count..), stat = "count",
            position = position_stack(0.5)) +
  geom_label(aes(label = scales::percent(..count.. / sum(..count..))),
             stat = "count") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma) +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Reordered") + ylab("Frequência") + ggtitle("Com que frequência as pessoas
recompram os mesmos itens?")
beep(2)
```



9.

```
# Itens mais recomprados
tmp <- df_prior %>% dplyr::group_by(product_id, reordered) %>%
  dplyr::summarise(reorderedFreq = n())
tmp <- tmp %>% dplyr::group_by(product_id) %>%
  dplyr::mutate(., reorderedProp = round(reorderedFreq / sum(reorderedFreq), 3))
tmp <- tmp %>% dplyr::group_by(product_id) %>%
  dplyr::mutate(., Freq = sum(reorderedFreq))
tmp_1 <- tmp %>% dplyr::filter(., reordered == 1 & Freq > 40) %>%
  arrange(., desc(reorderedProp)) %>%
  dplyr::left_join(x = ., y = df_prior, by = c("product_id" = "product_id"))
tmp_1 <- tmp_1[1:2000, c("product_name", "reorderedFreq", "reorderedProp",
"Freq")]
tmp_1

ggplot(tmp_1, aes(x = as.factor(reorder(product_name, desc(reorderedProp))), y =
reorderedProp)) +
  geom_bar(stat="summary", fun.y = "mean", fill = "#69b3a2") + theme(plot.title =
element_text(hjust = 0.5)) + coord_cartesian(ylim=c(0.8, 1)) +
```

```
  geom_text(aes(label = reorderedProp), position = position_stack(1)) +
  xlab("Reordered Product") + ylab("reorderedProp") + ggtitle("Reordered Product")
+ theme(axis.text.x = element_text(angle = 90, hjust = 1,))

beep(2)
```