COMP1013_22176385_Assigment

2025-05-22

By including this statement, I, the author of this work, verify that:

- I hold a copy of this assignment that I can produce if the original is lost or damaged.
- I hereby certify that no part of this assignment/product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.
- No part of this assignment/product has been written/produced for me by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned.
- I am aware that this work may be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism (which may retain a copy on its database for future plagiarism checking).
- I hereby certify that I have read and understand what the School of Computing, Data Science and Mathematics defines as minor and substantial breaches of misconduct as outlined in the learning guide for this unit.

Identity

Student Surname: Prema Santhi

Student Firstname: I Gusti Ayu Agung

Student ID: 22176385

Unit Name: Analytics Programming

Unit Number: COMP1013

Dataset information

In this assigment, we use several provided datasets such as below:

1. customers.csv
2. geolocation.csv
3. order_items.csv
4. orders.csv
5. payments.csv
6. products.csv
7. sellers.csv

Load all the dataset

Before begin to the analysis, load all the data first

```
customer.data=read.csv("customers.csv")
geo.data=read.csv("geolocation.csv")
item.data=read.csv("order_items.csv")
orders=read.csv("orders.csv")
payment.data=read.csv("payments.csv")
product.data=read.csv("products.csv")
sellers.data=read.csv("sellers.csv")
```

Call the necessery libraries

```
library(tidyverse)
library(ggplot2)
```

Task 1

We are required to analyze the distribution of customers across different states. So in that case we will use customers data.

Use table to count the states from customer_state variable inside customer.data

```
state.count=table(customer.data$customer_state)
```

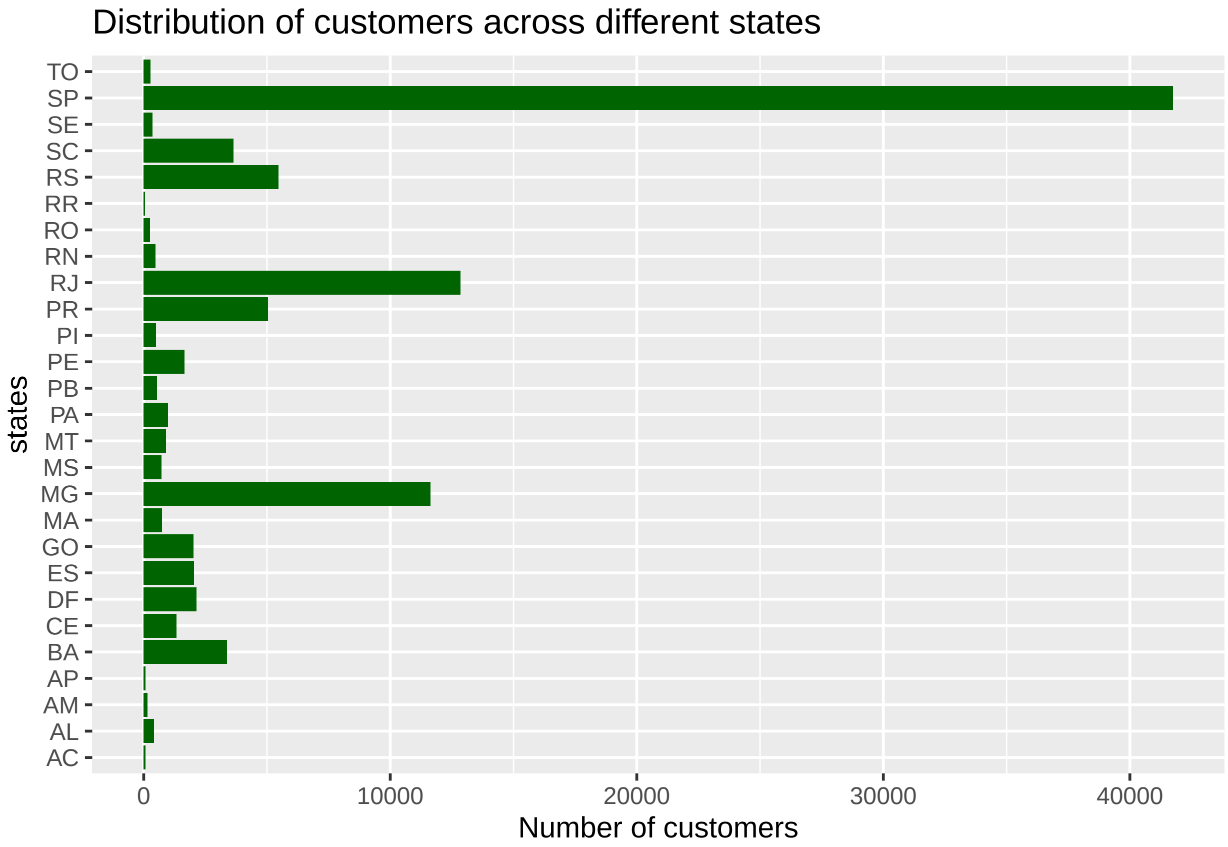
Convert it into data frame. There are 2 reasons why to do so:

1. We can easily read or interpret data in the data frame format
2. To be able to use ggplot, the data should be in data frame

```
state.count.df=as.data.frame(state.count)
colnames(state.count.df)=c("state_name", "count")
```

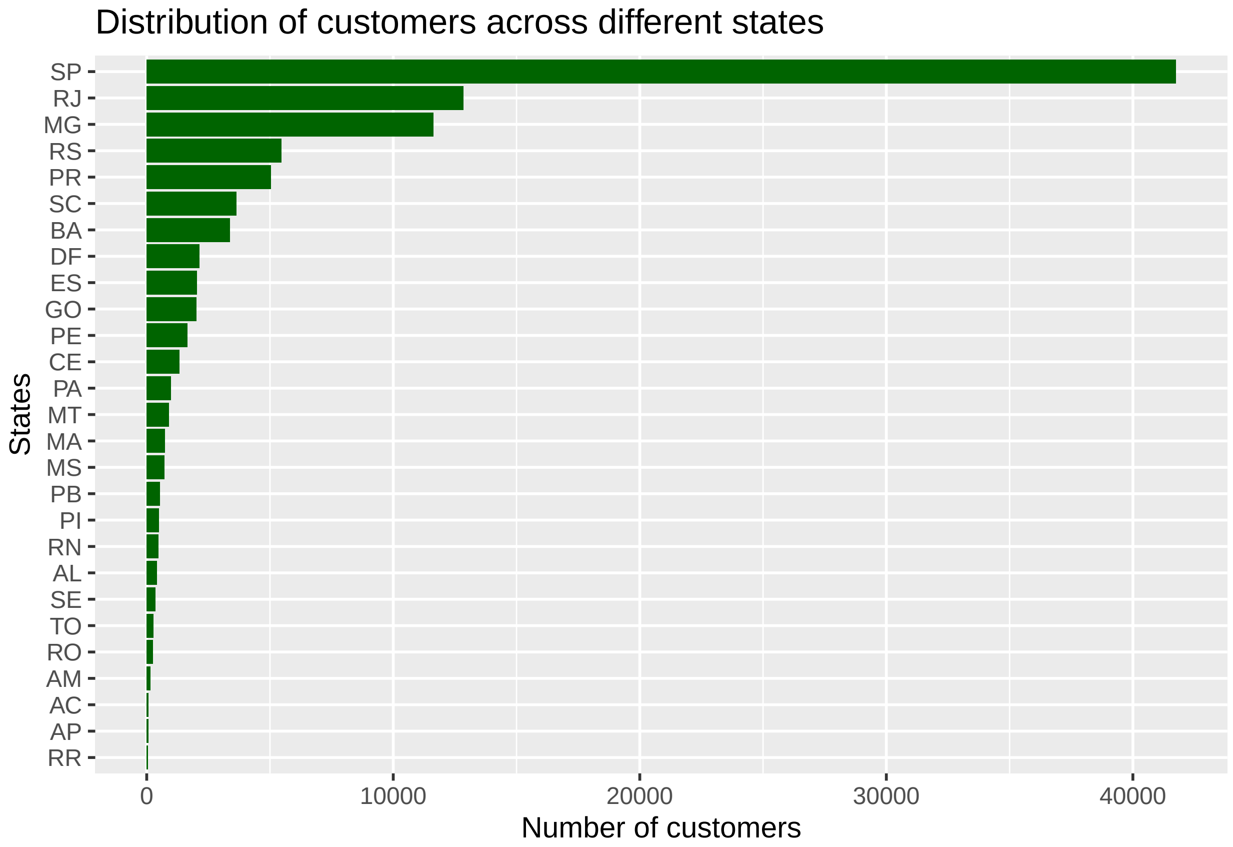
Now visualize the data

```
ggplot(data=state.count.df)+
  geom_col(mapping=aes(x=state_name, y=count), fill="dark green")+
  labs(title="Distribution of customers across different states",
       y="Number of customers", x="states")+
  coord_flip()
```



The graph is messy and hard to understand. Even though we can immediately know the highest is SP, but it is better to read the graph by the order. So:

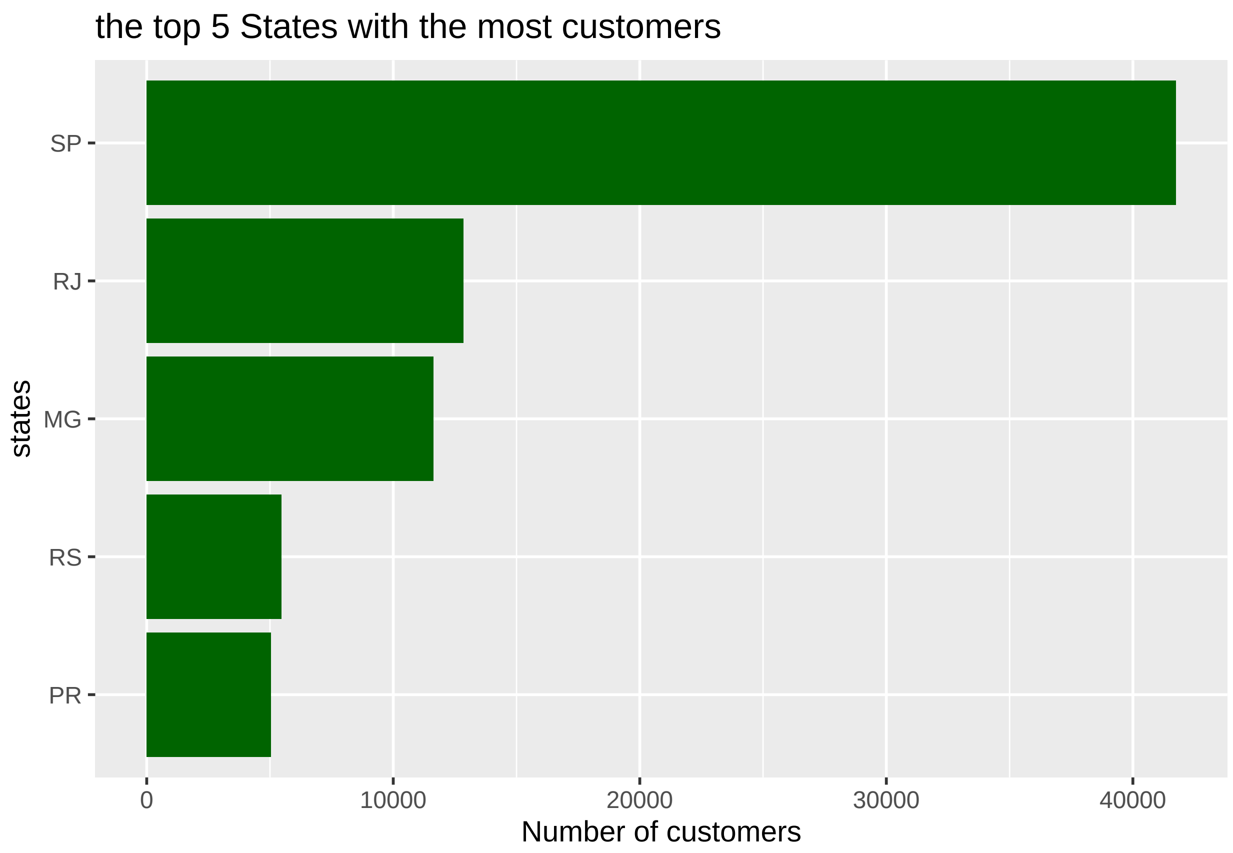
```
## put (reorder) in the x axis
ggplot(data=state.count.df)+
  geom_col(mapping=aes(x=reorder(state_name, count), y=count), fill="dark green")+
  labs(title="Distribution of customers across different states",
       y="Number of customers", x="States")+
  coord_flip()
```



Since there are so many states name in that graph, we filter it into top 5 states with the most customers

```r
state.count.sorted=sort(state.count, decreasing=TRUE) #sort from the highest to the smallest
top5.states=state.count.sorted[1:5]
top5.states.df=as.data.frame(top5.states)
colnames(top5.states.df)=c("states","count")

ggplot(data=top5.states.df)+
  geom_col(mapping=aes(x=reorder(states, count), y=count), fill="dark green")+
  labs(title="the top 5 States with the most customers",
       y="Number of customers", x="states")+
  coord_flip()
```



From the graph above, we could see that SP state has the highest customers. It also create a big gap between the top 1 with top 2 (RJ state) which is around $30,000$

Task 2

We assigned to identify the top 3 most frequent product categories based on the number of items sold

First, we need to join product.data and item.data and stored it in a variable

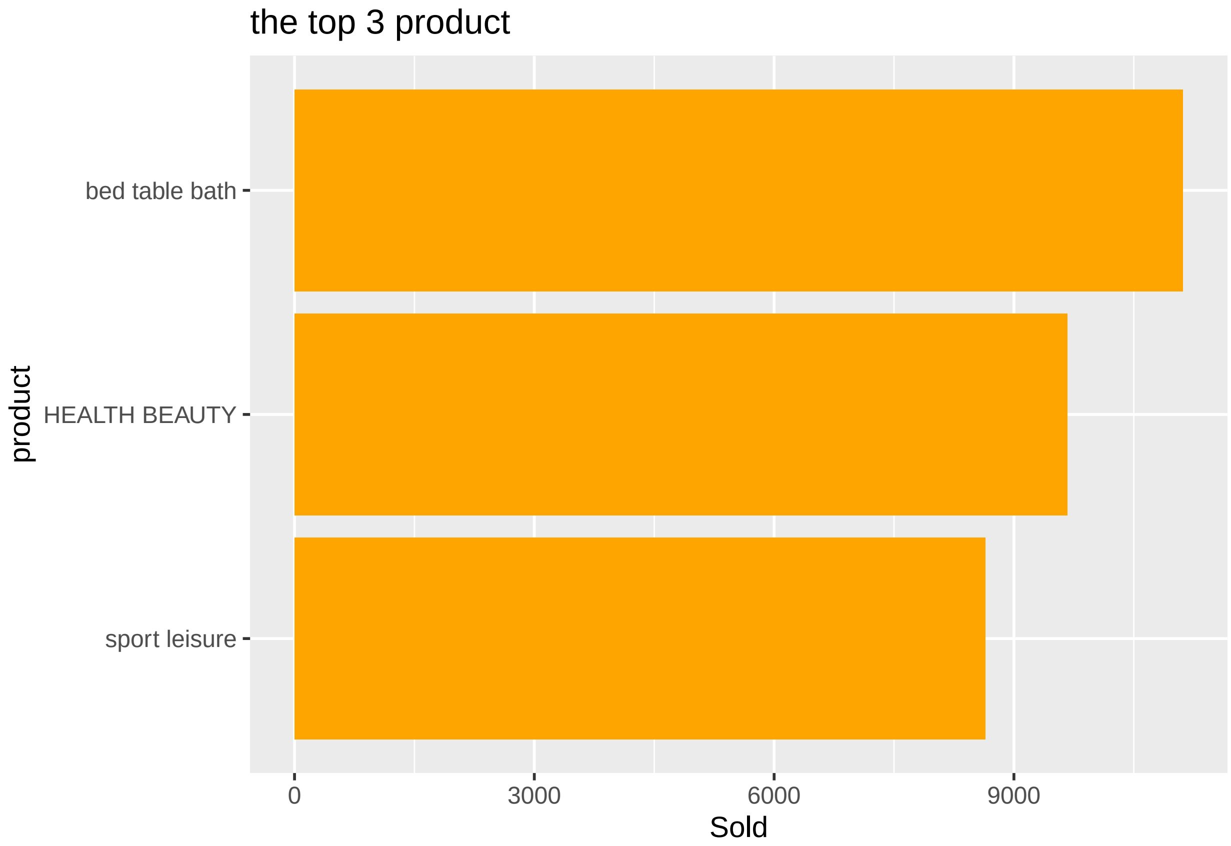
```r
colnames(item.data)
colnames(product.data)
product.item.join=product.data%>%
  inner_join(item.data, by=c("product_id"))
```

Count and convert it into a data frame

```r
category.count=table(product.item.join$product.category)
category.sorted=sort(category.count, decreasing = TRUE)
top3.category=category.sorted[1:3]
top3.category.df=as.data.frame(top3.category)
colnames(top3.category.df)=c("product", "count")
```

VIsualize using bar chart (geom_col)

```
ggplot(data=top3.category.df)+
  geom_col(mapping=aes(x=reorder(product, count), y=count), fill="orange")+
  labs(title="the top 3 product",
       y="Sold", x="product")+
  coord_flip()
```



Bassed on the finding, we could see that the customers in Brazil like to buy bed table bath, health beauty, and sport leisure.

Now we will analyze customer behaviour when buying the top 3 category.

First, join the order data, product data, and item data

```
colnames(product.data)
colnames(item.data)
colnames(orders)

# bed table bath
ordertime.btb=item.data%>%
  inner_join(orders, by="order_id")%>%
  inner_join(product.data, by="product_id")%>%
  filter(product.category==c("bed table bath"))%>%
  select(order_purchase_timestamp)

# health beauty
ordertime.HB=item.data%>%
  inner_join(orders, by="order_id")%>%
  inner_join(product.data, by="product_id")%>%
  filter(product.category==c( "HEALTH BEAUTY"))%>%
  select(order_purchase_timestamp)

# sport leisure
ordertime.sport.l=item.data%>%
  inner_join(orders, by="order_id")%>%
  inner_join(product.data, by="product_id")%>%
  filter(product.category==c("sport leisure"))%>%
  select(order_purchase_timestamp)

# top 3 product
ordertime.item=item.data%>%
  inner_join(orders, by="order_id")%>%
  inner_join(product.data, by="product_id")%>%
  filter(product.category==c("sport leisure", "HEALTH BEAUTY", "bed table bath"))%>%
  select(product.category, order_purchase_timestamp)
```

Then we extract the day from the purchase time

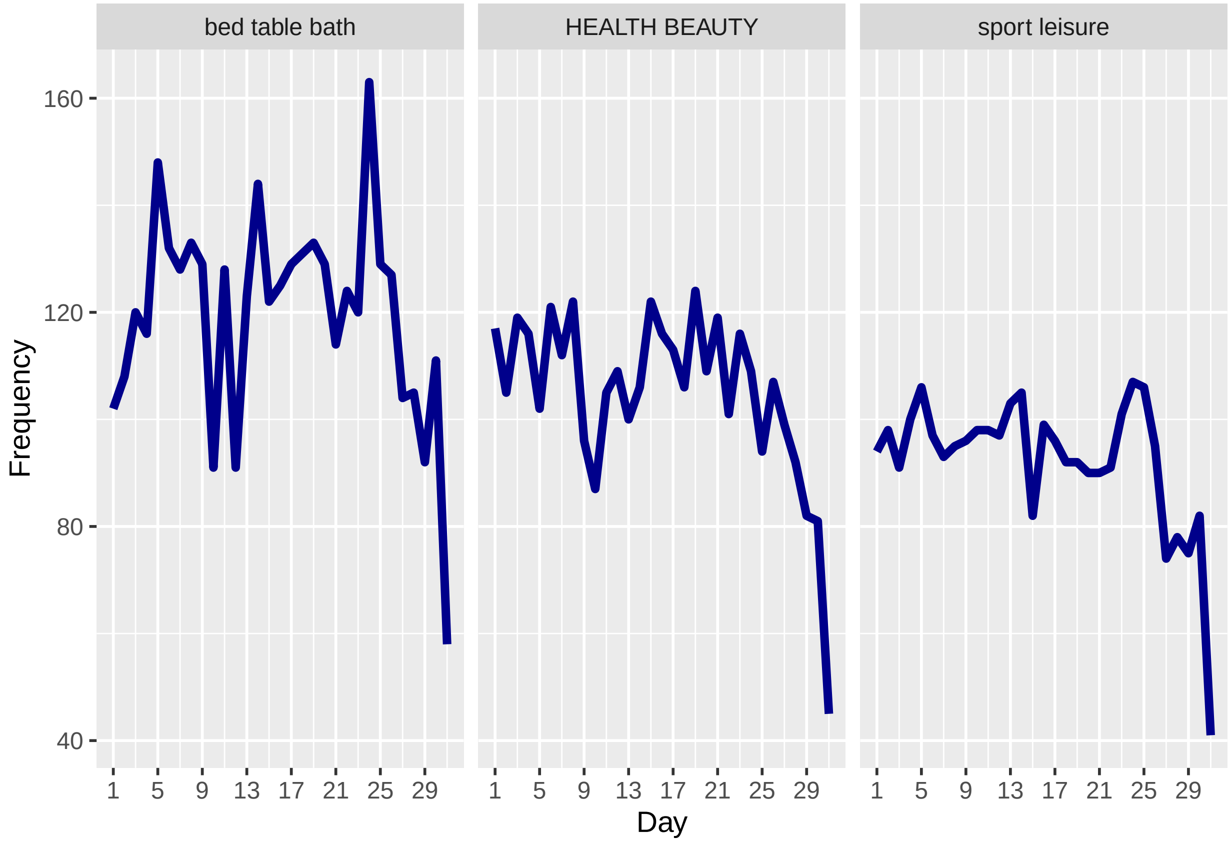
```
ordertime.btb$order_purchase_timestamp=as.numeric(format(as.POSIXct(ordertime.btb$order_purchase_timesta
ordertime.btb.df=as.data.frame(table(ordertime.btb))

ordertime.HB$order_purchase_timestamp=as.numeric(format(as.POSIXct(ordertime.HB$order_purchase_timestamp
ordertime.HB.df=as.data.frame(table(ordertime.HB))

ordertime.sport.l$order_purchase_timestamp=as.numeric(format(as.POSIXct(ordertime.sport.l$order_purchase
ordertime.sport.l.df=as.data.frame(table(ordertime.sport.l))

ordertime.item$order_purchase_timestamp=as.numeric(format(as.POSIXct(ordertime.item$order_purchase_times
ordertime.item.df=as.data.frame(table(ordertime.item))
```

```
ggplot(data=ordertime.item.df)+
  geom_line(mapping=aes(x=as.numeric(order_purchase_timestamp), y=Freq),
            col="dark blue",
            size=1.5)+
  scale_x_continuous(breaks = seq(1,31, by=4))+
  facet_wrap(~ product.category, ncol = 3)+
  labs(x = "Day", y = "Frequency")
```



According to the data, most people are more likely to stop buying things from the top 3 product (bed table bath, health beauty, sport leisure) at the end of the month. Let us break down for each product:

1. Bed table bath

Customers often buy this product at 5th, around 12th, and the peak is at 24th. Company can make sure they stock a lot of bed table bath so people do not run out of the product.

2. Health beauty

The purchse for health beauty are fluctuative. And it went down at the end of the end of the month.

3. sport leisure

Customer behaviour for this product are quite stable.So make sure to promote it well throught the month

Task 3

Calculate the actual delivery time in days for each delivered order

First we need to check the class for the purchase timestamp and the delivery date

```
class(orders$order_purchase_timestamp) # it's character
```

```
## [1] "character"
```

```
orders$order_purchase_timestamp=as.POSIXct(orders$order_purchase_timestamp, format= "%Y-%m-%d %H:%M:%S"
class(orders$order_purchase_timestamp)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(orders$order_delivered_customer_date) # it's character
```

```
## [1] "character"
```

```
orders$order_delivered_customer_date=as.POSIXct(orders$order_delivered_customer_date, format= "%Y-%m-%d
class(orders$order_purchase_timestamp)
```

```
## [1] "POSIXct" "POSIXt"
```

Now we filter

```
table(orders$order_status) # check status name that exist in the category
```

```
##
##    approved    canceled     created   delivered    invoiced  processing
##           2         625           5       96478         314         301
##     shipped unavailable
##        1107         609
```

```
# Now we just want the row for "delivered" only
delivered.order=orders %>%
  filter(order_status=="delivered")
```

Next, see the difference between the purchase time and customer receipt time

```
delivered.order$delivery.time.days=as.numeric(difftime(delivered.order$order_delivered_customer_date,
                                                       delivered.order$order_purchase_timestamp),
                                              units="days")
```

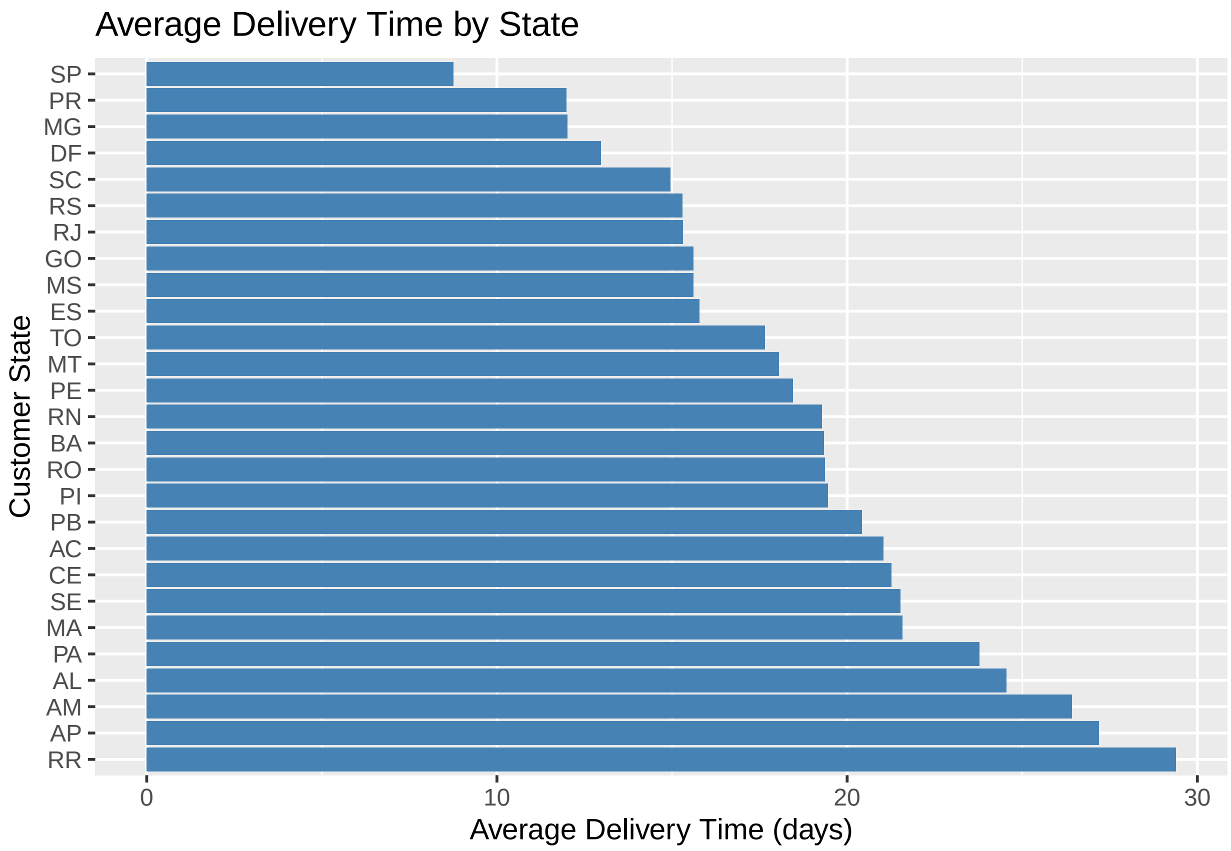
Second thing, we analyze how delivery time varies across different customer states. That means we need to join between customer data and the new delivered.order variabe

```
colnames(delivered.order)
colnames(customer.data)
cust.delivered=delivered.order%>%
  inner_join(customer.data, by="customer_id")
```

Calculate avarage delivery time by state

```
avg.delivery.states=aggregate(delivery.time.days~customer_state, data=cust.delivered, mean)

ggplot(data=avg.delivery.states)+
  geom_col(mapping=aes(x=reorder(customer_state, -delivery.time.days), y=delivery.time.days), fill="stee
  coord_flip()+
  labs(
    title = "Average Delivery Time by State",
    x = "Customer State",
    y = "Average Delivery Time (days)")
```



From the graph, we found out that SP state got the shortest delivery time. Refers back to the first task, most customers are from SP so we an state a hypothesis that they improve the delivery time in SP because many customers are from there

Task 4

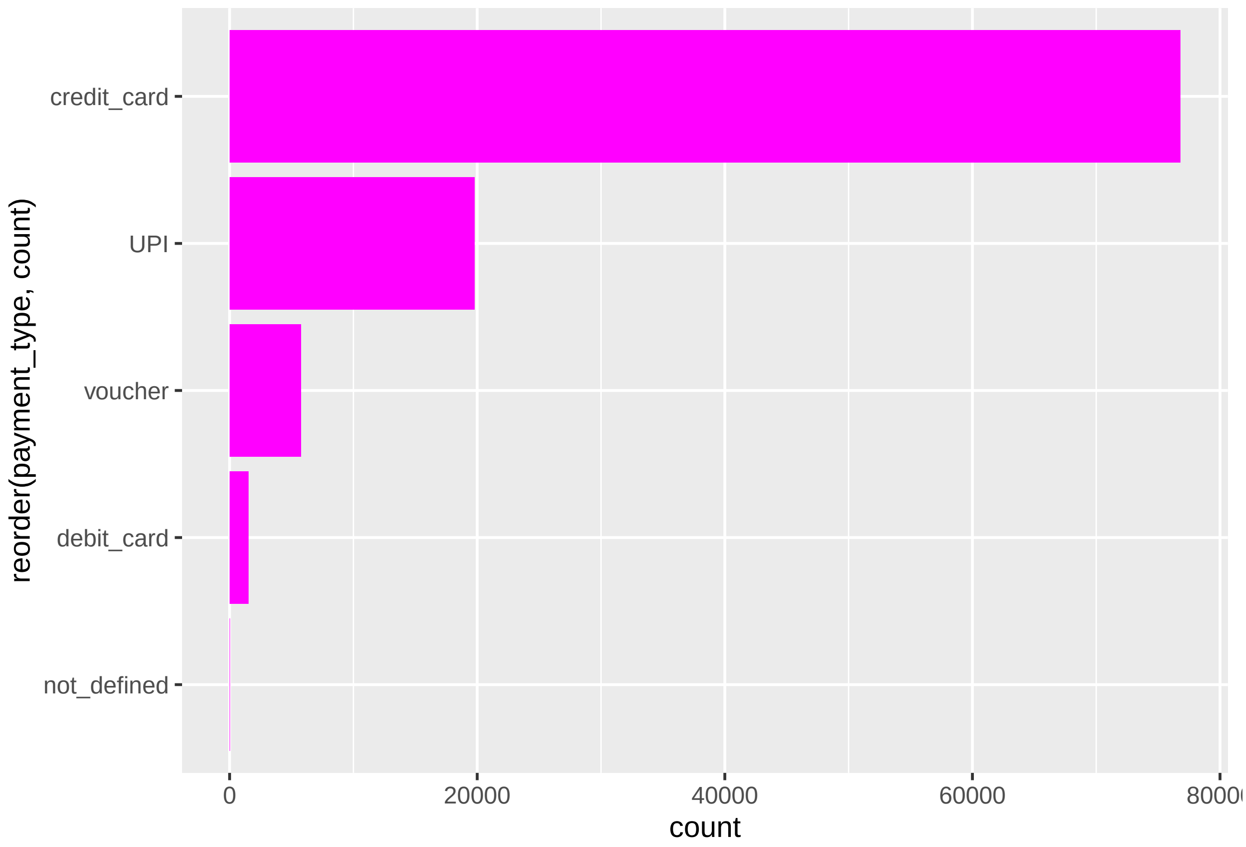
Analyze the usage frequency of different payment types

First, count the payment type and stored it in a data frame

```
payment.count=table(payment.data$payment_type)
payment.sorted=sort(payment.count, decreasing = TRUE)
payment.df=as.data.frame(payment.sorted)
colnames(payment.df)=c("payment_type", "count")
```

Next, visualize the data into a bar chart

```r
ggplot(data=payment.df)+
  geom_col(mapping=aes(x=reorder(payment_type, count), y=count), fill="magenta")+
  coord_flip()
```



The most frequent used payment method is credit card with around 70,000 transactions. This suggest that the customers strongly prefer using credit cards.It is more convenient because people do not need to worry the amount of cash they had at that moment(Kelton and Little 2025).

According to EBANX("UPI (Unified Payment Interface)" n.d), UPI is a payment system that was used in India. However, since this dataset is based in Brazil, we are afraid that there's a mislabeled in the dataset.

References

Kelton, Katie, and Kendall Little. 2025. "Credit Card Pros and Cons."

"UPI (Unified Payment Interface)." n.d. EBANX.