Credit_card_EDA.R

rocka

2024-03-04

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

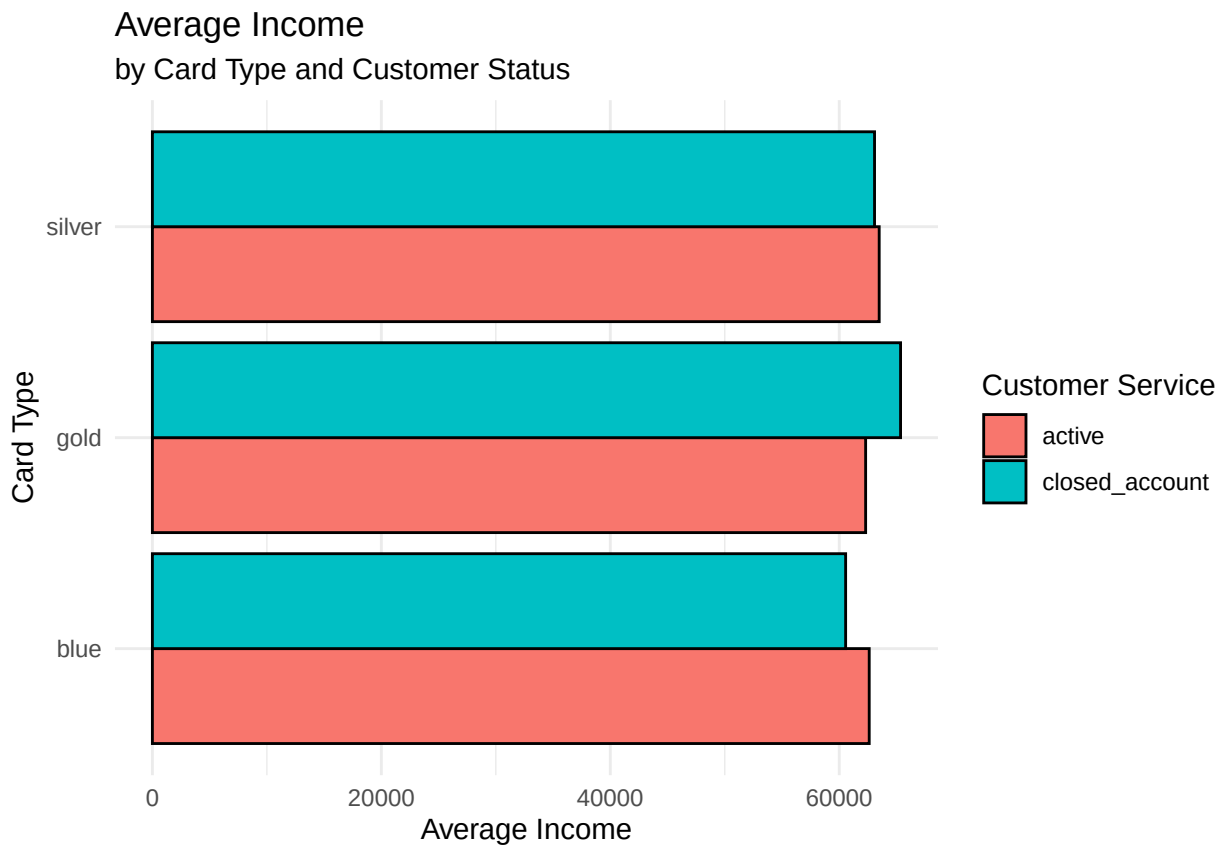
```
credit_card_df<-read.csv("customer.csv")

#Exploratory Data Analysis

(df1 <- credit_card_df %>%
    group_by(card_type,customer_status) %>%
    summarise(
      n_customers = n(),
      avg_climit=mean(credit_limit),
      min_income=min(income),
      avg_income=mean(income),
      max_income=max(income)
    ))
```

```
## `summarise()` has grouped output by 'card_type'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 7
## # Groups:   card_type [3]
##   card_type customer_status n_customers avg_climit min_income avg_income
##   <chr>     <chr>                 <int>      <dbl>      <int>      <dbl>
## 1 blue      active                 1054      8992.      30333     62619.
## 2 blue      closed_account         1497      8012.      30198     60558.
## 3 gold      active                  609      8643.      30132     62305.
## 4 gold      closed_account          299      8410.      30259     65357.
```

```
## 5 silver    active                 872      8661.      30094     63489.
## 6 silver    closed_account         296      8558.      31297     63083.
## # i 1 more variable: max_income <int>
```

```
ggplot(df1, aes(x = card_type, y = avg_income, fill = customer_status)) +
  geom_bar(stat = "identity", position = position_dodge(),color="black") +
  labs(title = "Average Income",subtitle="by Card Type and Customer Status",
       x = "Card Type",
       y = "Average Income",
       fill="Customer Service")+
  theme_minimal() +coord_flip()
```



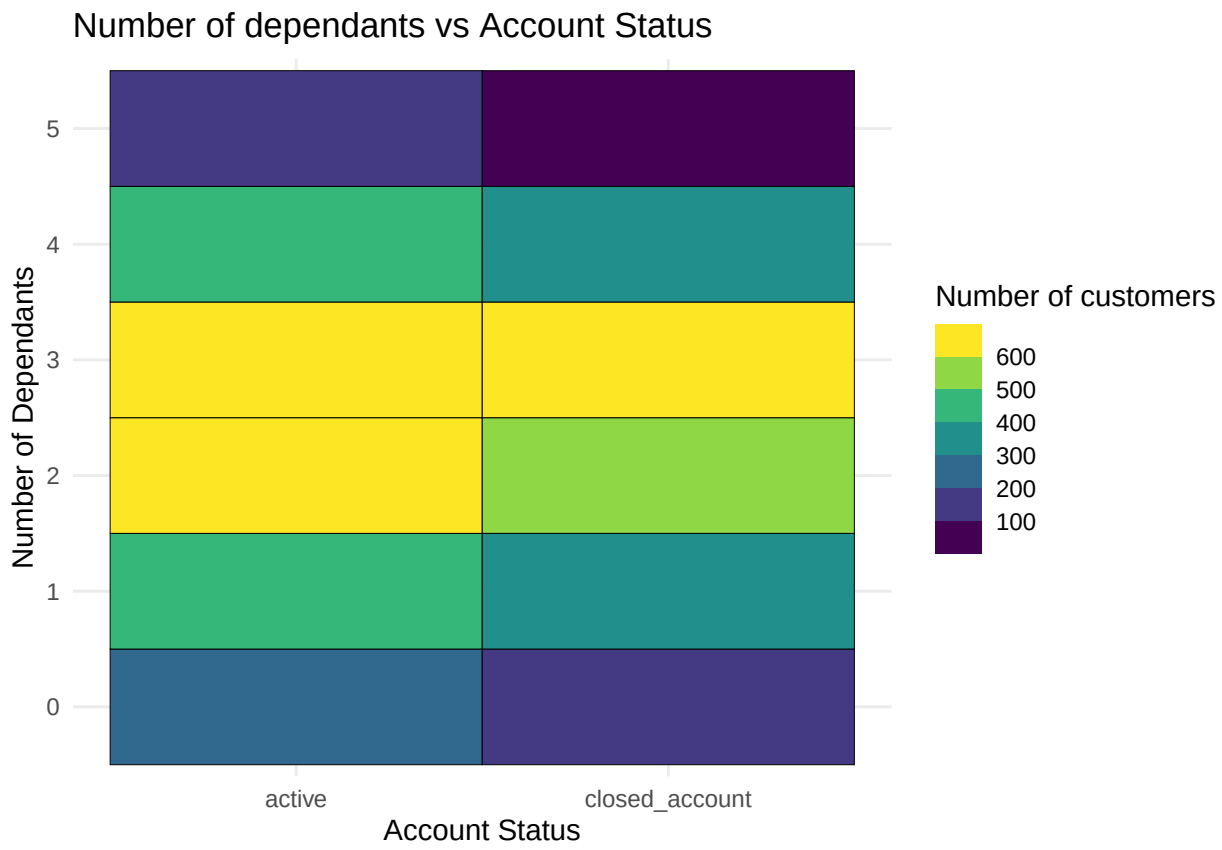

```
(df2 <- credit_card_df %>%
    group_by(dependents,customer_status) %>%
    summarise(
      n_customers = n()
    ))
```

```
## `summarise()` has grouped output by 'dependents'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 12 x 3
## # Groups:   dependents [6]
##    dependents customer_status n_customers
##         <int> <chr>                 <int>
```

```
##  1          0 active                233
##  2          0 closed_account        178
##  3          1 active                475
##  4          1 closed_account        345
##  5          2 active                612
##  6          2 closed_account        521
##  7          3 active                693
##  8          3 closed_account        630
##  9          4 active                413
## 10          4 closed_account        336
## 11          5 active                109
## 12          5 closed_account         82
```

```
ggplot(df2, aes(x = customer_status, y = factor(dependents), fill = n_customers)) +
  geom_tile(color="black") +
  scale_fill_viridis_b() +
  labs(title = "Number of dependants vs Account Status",
       x = "Account Status",
       y = "Number of Dependants",
       fill = "Number of customers") +
  theme_minimal()
```

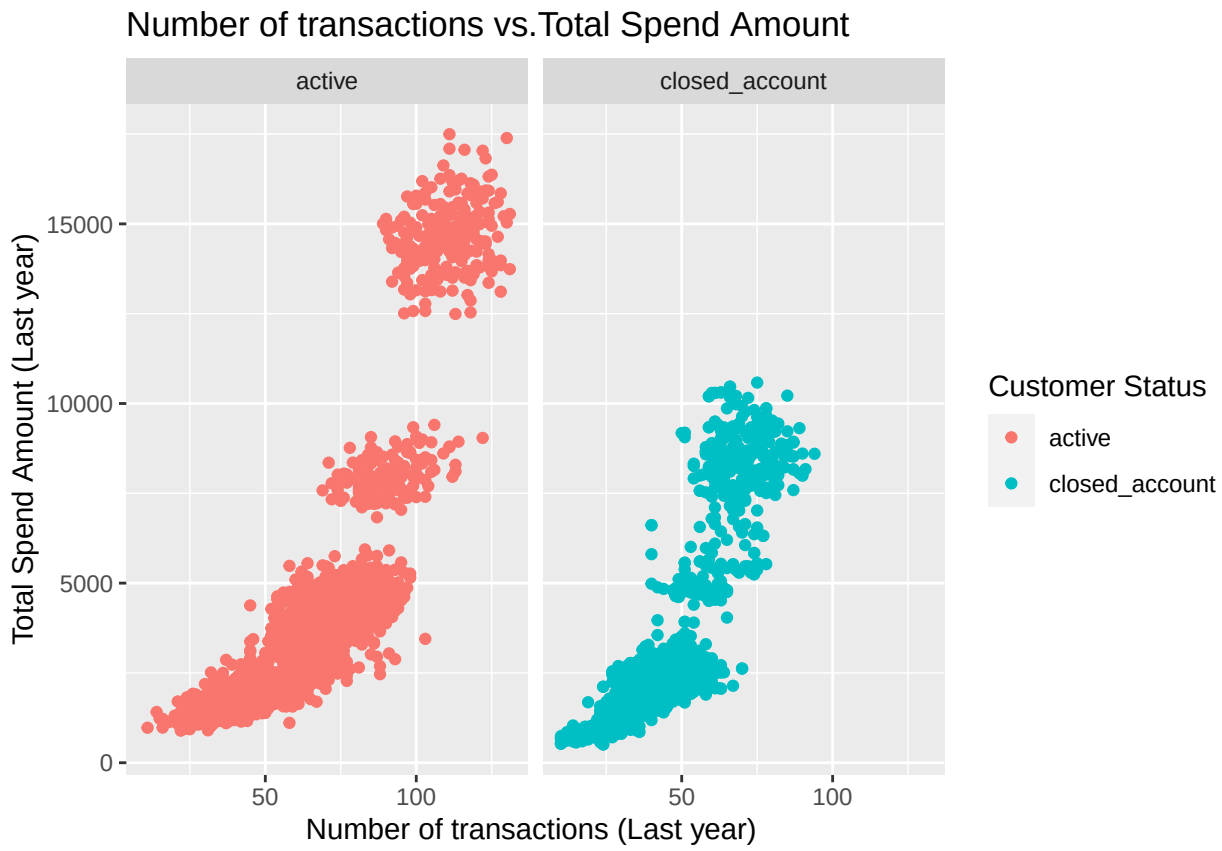



```
(df3 <- credit_card_df %>%
    group_by(customer_status) %>%
    summarise(
```

```
      n_customers = n(),
      avg_num_trans=mean(transactions_last_year),
      avg_total_spend=mean(total_spend_last_year)
    ))
```

```
## # A tibble: 2 x 4
##   customer_status n_customers avg_num_trans avg_total_spend
##   <chr>                <int>         <dbl>           <dbl>
## 1 active                2535          68.5           4597.
## 2 closed_account        2092          45.0           3121.
```

```
ggplot(credit_card_df, aes(x = transactions_last_year, y = total_spend_last_year, color=customer_status)
  geom_point() +
  facet_wrap(~ customer_status)+
  labs(title = "Number of transactions vs.Total Spend Amount",
       x="Number of transactions (Last year) ",
       y="Total Spend Amount (Last year)",
       color="Customer Status")
```

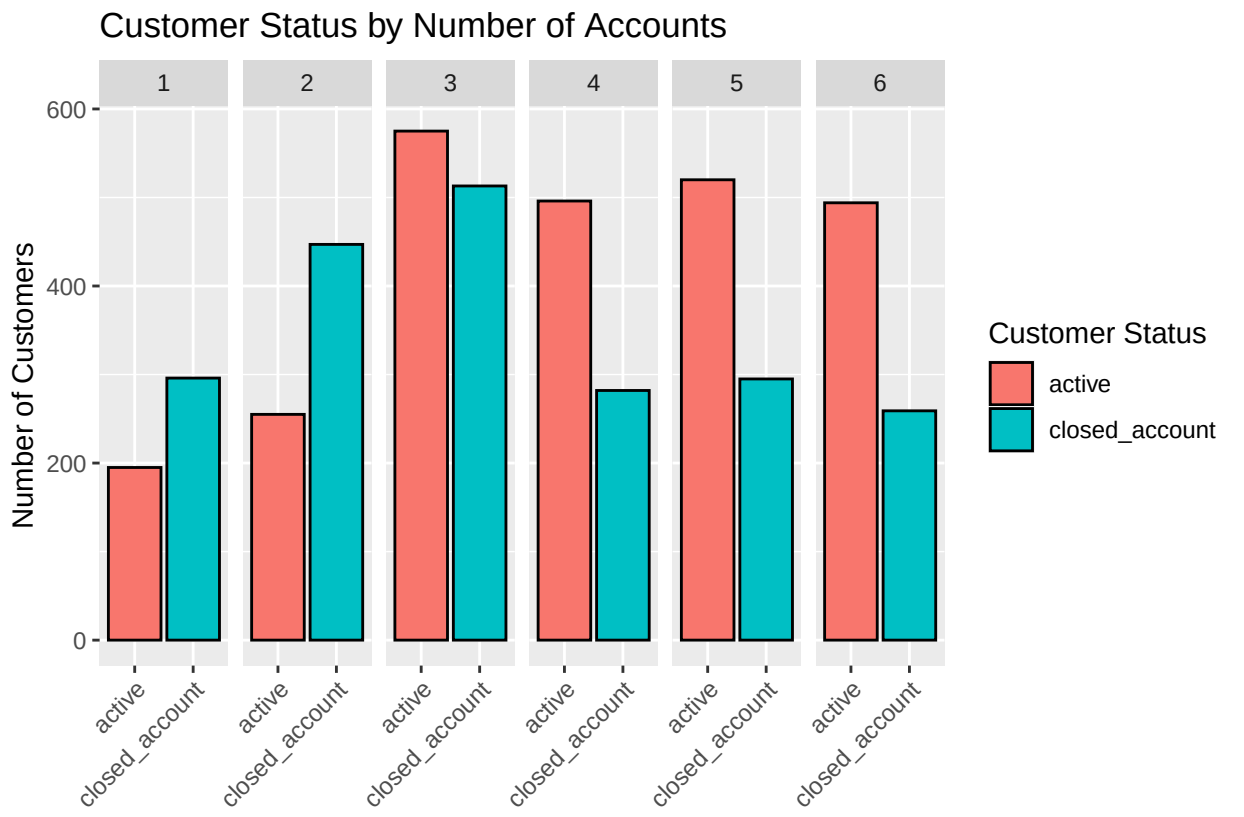



```
(df4 <- credit_card_df %>%
    group_by(total_accounts,customer_status) %>%
    summarise(
      n_customers = n(),
    ))
```

```
## `summarise()` has grouped output by 'total_accounts'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 12 x 3
## # Groups:   total_accounts [6]
##    total_accounts customer_status n_customers
##             <int> <chr>                 <int>
##  1              1 active                  195
##  2              1 closed_account          296
##  3              2 active                  255
##  4              2 closed_account          447
##  5              3 active                  575
##  6              3 closed_account          513
##  7              4 active                  496
##  8              4 closed_account          282
##  9              5 active                  520
## 10              5 closed_account          295
## 11              6 active                  494
## 12              6 closed_account          259
```

```
ggplot(data = credit_card_df, mapping = aes(x = customer_status, fill = customer_status)) +
  geom_bar(stat = "count",color="black") +
  facet_wrap(~ total_accounts, nrow = 1) +
  labs(title = "Customer Status by Number of Accounts", x = " ",
       y = "Number of Customers",fill="Customer Status") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

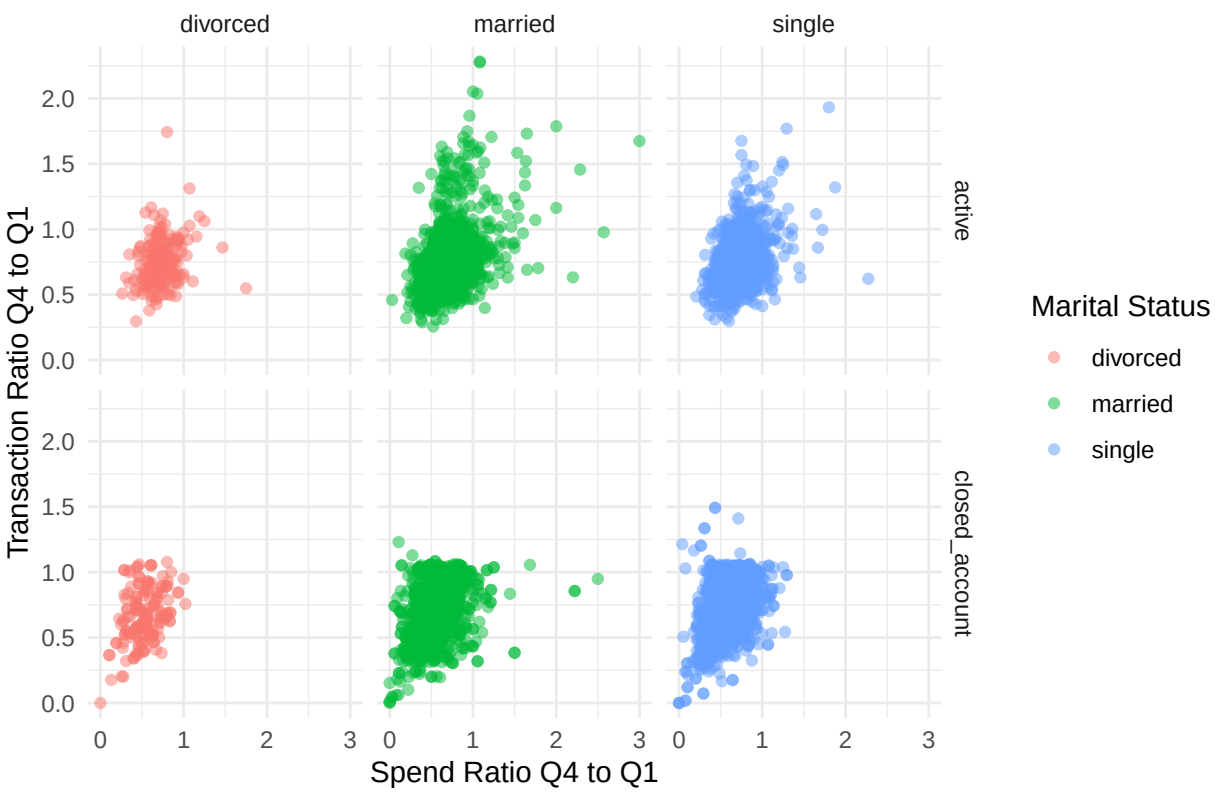


```r
(df5 <- credit_card_df %>%
    group_by(marital_status,customer_status) %>%
    summarise(
      n_cust = n(),
      avg_TR=mean(transaction_ratio_q4_q1),
      min_TR=min(transaction_ratio_q4_q1),
      max_TR=max(transaction_ratio_q4_q1),
      avg_SR=mean(spend_ratio_q4_q1),
      min_SR=min(spend_ratio_q4_q1),
      max_TR=max(spend_ratio_q4_q1),
    ))
```

```
## `summarise()` has grouped output by 'marital_status'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 6 x 8
## # Groups:   marital_status [3]
##   marital_status customer_status n_cust avg_TR min_TR max_TR avg_SR min_SR
##   <chr>          <chr>            <int>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 divorced       active             195  0.727  0.263   1.74  0.752  0.296
## 2 divorced       closed_account     159  0.532  0       1.08  0.671  0
## 3 married        active            1277  0.740  0.028   2.28  0.786  0.256
## 4 married        closed_account     989  0.548  0       1.23  0.686  0
## 5 single         active            1063  0.735  0.207   1.93  0.757  0.298
## 6 single         closed_account     944  0.567  0       1.49  0.708  0
```

```r
# TR & SR stands for transaction ratio and spend ratio respectively

ggplot(credit_card_df, aes(x = transaction_ratio_q4_q1, y = spend_ratio_q4_q1, color = marital_status)) +
  geom_point(alpha = 0.5) +
  facet_grid(customer_status ~ marital_status) +  # Changed from facet_wrap to facet_grid
  labs(title = "Spend Ratio vs Transaction Ratio by Customer and Marital Status",
       x = "Spend Ratio Q4 to Q1",
       y = "Transaction Ratio Q4 to Q1",
       color = "Marital Status") +
  theme_minimal()
```


Spend Ratio vs Transaction Ratio by Customer and Marital Status
divorced
married
single
active
closed_account
Transaction Ratio Q4 to Q1
Spend Ratio Q4 to Q1
0.0
0.5
1.0
1.5
2.0
0
1
2
3
Marital Status
divorced
married
single