Data Analyst for Finance

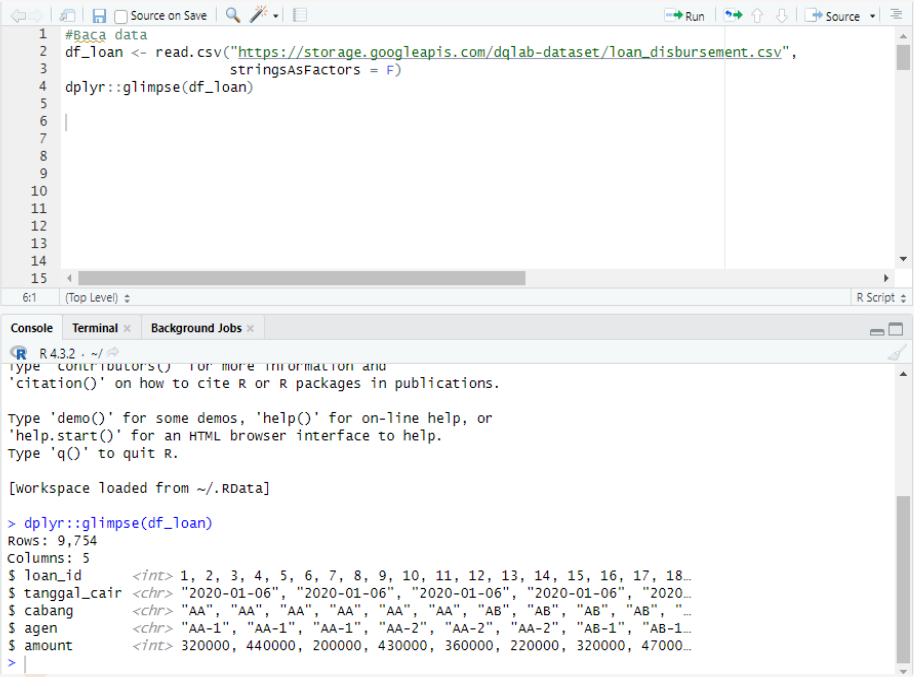
Dinda Febriani

Branch Performance

Background

- DQLab Finance has established branches in various locations since its founding in January 2020. Despite being less than a year old, DQLab Finance has consistently provided financial assistance to the community and has been expanding each month by opening new branches.
- With numerous branches already in operation, it becomes essential to monitor the performance of each branch.
- Within each branch, there are agents responsible for identifying and documenting prospective partners who intend to apply for loans with DQLab Finance. Once approved, these agents are also responsible for disbursing funds to the approved partners.

The Utilized Data



- Because the data is available in the RDS format, it can be directly read in R using the read.csv code

Tables in Data

loan_id	The table contains unique IDs
tanggal_cair	Disbursement dates
cabang	Agent work locations
Agen	the field officer responsible for disbursement
amount	the disbursed amount

01

Explore The Data

Total Amount in Mei 2020 per Branch

Project Data Analysis for Finance Perfor... *

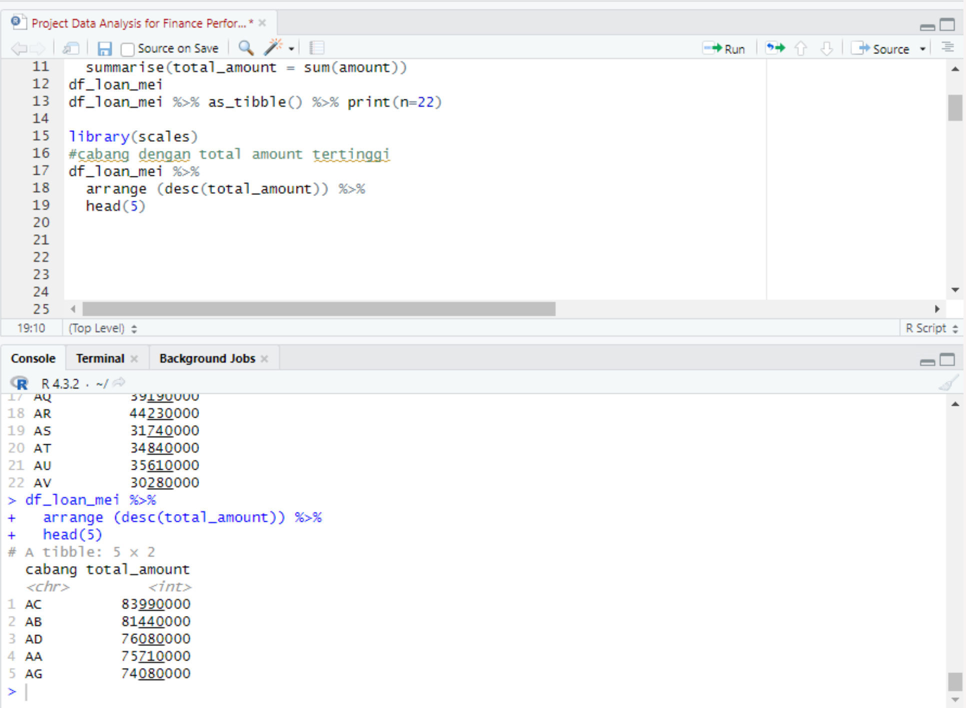
```r
df_loan <- read.csv("https://storage.googleapis.com/dqlab-dataset/loan_disbursement.csv",
                    stringsAsFactors = F)
dplyr::glimpse(df_loan)

#filter data bulan mei, dan total data per cabang
library(dplyr)
df_loan_mei <- df_loan %>%
  filter(tanggal_cair >= '2020-05-01', tanggal_cair <= '2020-05-31') %>%
  group_by(cabang) %>%
  summarise(total_amount = sum(amount))
df_loan_mei
df_loan_mei %>% as_tibble() %>% print(n=22)
```

Console | Terminal | Background Jobs

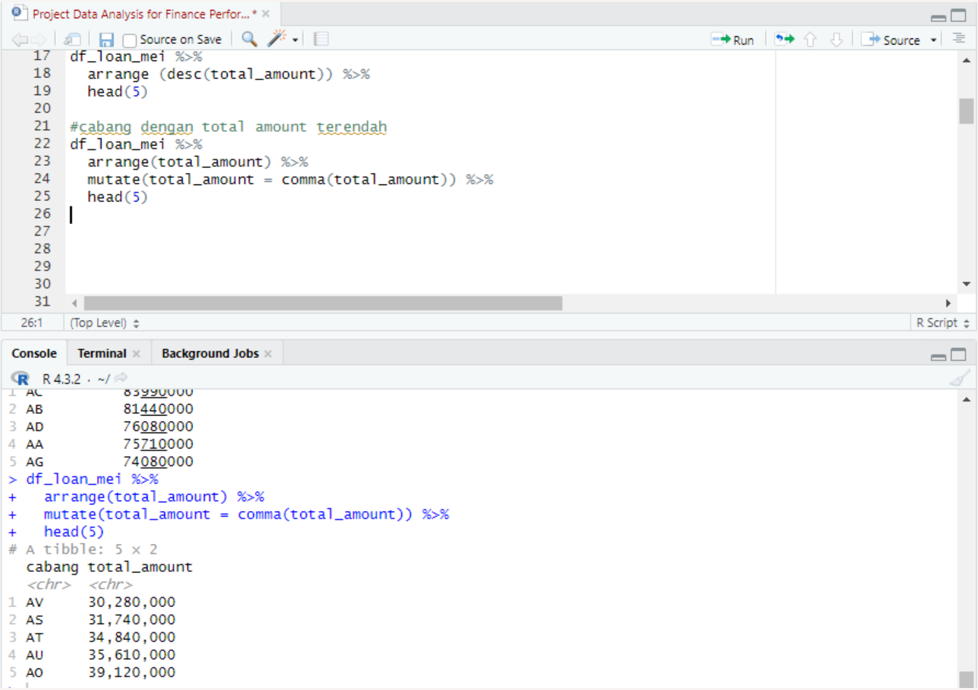
R 4.3.2 · ~/

```
# A tibble: 22 × 2
   cabang total_amount
   <chr>         <int>
 1 AA         75710000
 2 AB         81440000
 3 AC         83990000
 4 AD         76080000
 5 AE         54200000
 6 AF         68040000
 7 AG         74080000
 8 AH         73840000
 9 AI         46640000
10 AJ         43580000
11 AK         44590000
12 AL         40650000
13 AM         44010000
14 AN         39700000
15 AO         39120000
```

Top 5 Highest Total Amount by Branch



Top 5 Lowest Total Amount by Branch



Insight

There is a significant difference between the highest and lowest amounts. Let's explore whether there is a correlation between the age of branches and the total amount

The Age of Each Branch

```r
  mutate(total_amount = comma(total_amount)) %>%
  head(5)

#hitung umur cabang
df_cabang_umur <- df_loan %>%
  group_by(cabang) %>%
  summarise(pertama_cair = min(tanggal_cair)) %>%
  mutate(umur = as.numeric(as.Date("2020-05-15") - as.Date(pertama_cair)) %/% 30)
df_cabang_umur
df_cabang_umur %>% as_tibble() %>% print(n=22)
```

R 4.3.2 · ~/

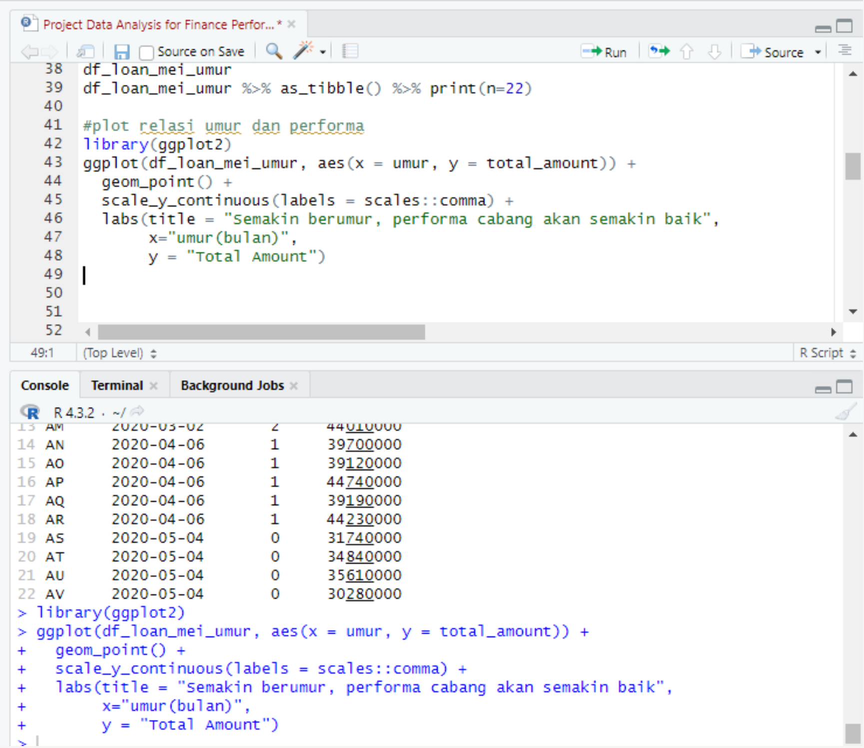
	cabang	pertama_cair	umur
	<chr>	<chr>	<dbl>
1	AA	2020-01-06	4
2	AB	2020-01-06	4
3	AC	2020-01-06	4
4	AD	2020-01-06	4
5	AE	2020-02-03	3
6	AF	2020-02-03	3
7	AG	2020-02-03	3
8	AH	2020-02-03	3
9	AI	2020-03-02	2
10	AJ	2020-03-02	2
11	AK	2020-03-02	2
12	AL	2020-03-02	2
13	AM	2020-03-02	2
14	AN	2020-04-06	1
15	AO	2020-04-06	1
16	AP	2020-04-06	1

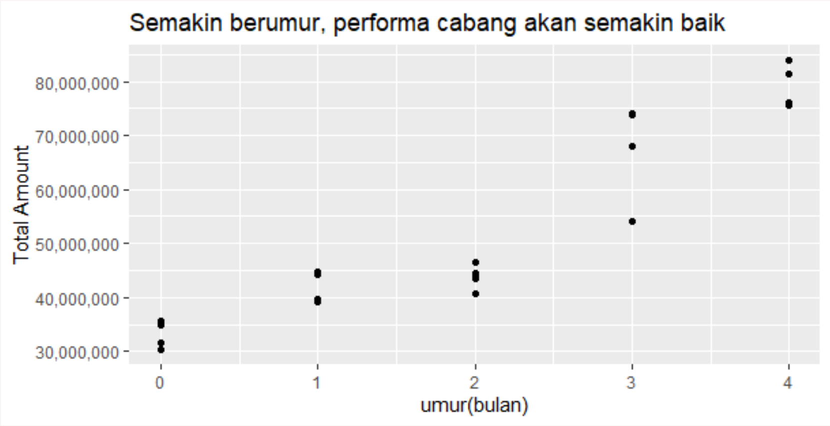
Combine May Performance with Branch Age

```r
  summarise(pertama_cair = min(tanggal_cair)) %>%
  mutate(umur = as.numeric(as.Date("2020-05-15") - as.Date(pertama_cair)) %/% 30)
df_cabang_umur
df_cabang_umur %>% as_tibble() %>% print(n=22)

#gabung data umur dan performa mei
df_loan_mei_umur <- df_cabang_umur %>%
  inner_join(df_loan_mei, by='cabang')
df_loan_mei_umur
df_loan_mei_umur %>% as_tibble() %>% print(n=22)
```

```
   cabang pertama_cair  umur total_amount
   <chr>  <chr>        <dbl>        <int>
 1 AA     2020-01-06       4     75710000
 2 AB     2020-01-06       4     81440000
 3 AC     2020-01-06       4     83990000
 4 AD     2020-01-06       4     76080000
 5 AE     2020-02-03       3     54200000
 6 AF     2020-02-03       3     68040000
 7 AG     2020-02-03       3     74080000
 8 AH     2020-02-03       3     73840000
 9 AI     2020-03-02       2     46640000
10 AJ     2020-03-02       2     43580000
11 AK     2020-03-02       2     44590000
12 AL     2020-03-02       2     40650000
13 AM     2020-03-02       2     44010000
14 AN     2020-04-06       1     39700000
15 AO     2020-04-06       1     39120000
16 AP     2020-04-06       1     44740000
```

Age-Performance Relationship in May with ScatterPlot





Insight

It seems that with the increasing age of branches, performance generally improves. However, we can observe that some branches exhibit lower performance during the mid-age phase. Let's further explore

Low Performance in Each Branch

```r
       x="umur(bulan)",
       y = "Total Amount")

#cabang rendah disetiap umur
df_loan_mei_flag <- df_loan_mei_umur %>%
  group_by(umur) %>%
  mutate(Q1 = quantile(total_amount, 0.25),
         Q3 = quantile(total_amount, 0.75),
         IQR = (Q3-Q1)) %>%
  mutate(flag = ifelse(total_amount < (Q1 - IQR), 'rendah', 'baik'))
df_loan_mei_flag %>%
  filter(flag == 'rendah') %>%
  mutate(umur = as.numeric(umur),
         total_amount = as.numeric(total_amount),
         Q1 = as.numeric(Q1),
         Q3 = as.numeric(Q3),
         IQR = as.numeric(IQR))

```

```
+   filter(flag == 'rendah') %>%
+   mutate(umur = as.numeric(umur),
+          total_amount = as.numeric(total_amount),
+          Q1 = as.numeric(Q1),
+          Q3 = as.numeric(Q3),
+          IQR = as.numeric(IQR))
# A tibble: 2 x 8
# Groups:   umur [2]
  cabang pertama_cair  umur total_amount       Q1       Q3      IQR flag
  <chr>  <chr>        <dbl>        <dbl>    <dbl>    <dbl>    <dbl> <chr>
1 AE     2020-02-03       3     54200000 64580000 73900000  9320000 rendah
2 AL     2020-03-02       2     40650000 43580000 44590000  1010000 rendah
>
```

Insight

"By using quartiles and interquartile range, we observe the smallest total amounts for each age. It can be seen that branches AE and AL exhibit low performance at the age of 3 and 2 months.

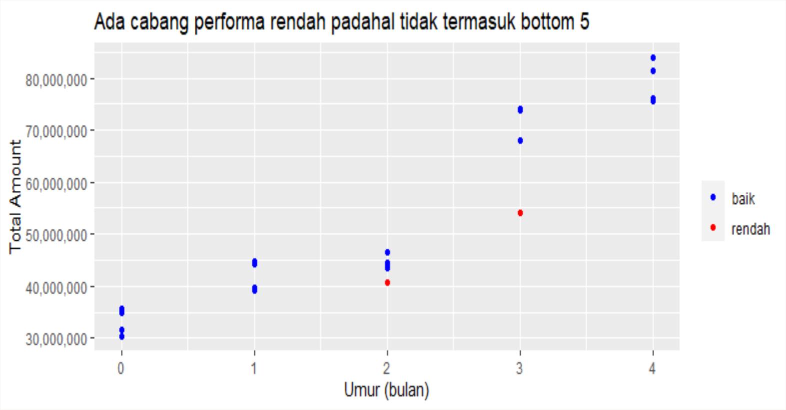
Exploring with a Scatter Plot but in Different Color

Project Data Analysis for Finance Perfor... *

```r
       x="umur(bulan)",
       y = "Total Amount")

#cabang rendah disetiap umur
df_loan_mei_flag <- df_loan_mei_umur %>%
  group_by(umur) %>%
  mutate(Q1 = quantile(total_amount, 0.25),
         Q3 = quantile(total_amount, 0.75),
         IQR = (Q3-Q1)) %>%
  mutate(flag = ifelse(total_amount < (Q1 - IQR), 'rendah', 'baik'))
df_loan_mei_flag %>%
  filter(flag == 'rendah') %>%
  mutate(umur = as.numeric(umur),
         total_amount = as.numeric(total_amount),
         Q1 = as.numeric(Q1),
         Q3 = as.numeric(Q3),
         IQR = as.numeric(IQR))
```

Console

```
+   mutate(umur = as.numeric(umur),
+          total_amount = as.numeric(total_amount),
+          Q1 = as.numeric(Q1),
+          Q3 = as.numeric(Q3),
+          IQR = as.numeric(IQR))
# A tibble: 2 × 8
# Groups:   umur [2]
  cabang pertama_cair  umur total_amount       Q1       Q3     IQR flag
  <chr>  <chr>        <dbl>        <dbl>    <dbl>    <dbl>   <dbl> <chr>
1 AE     2020-02-03       3     54200000 64580000 73900000 9320000 rendah
2 AL     2020-03-02       2     40650000 43580000 44590000 1010000 rendah
```



Compare Performance at The Same Age

Project Data Analysis for Finance Perfor...*

```r
  geom_point() +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Semakin berumur, performa cabang akan semakin baik",
       x="umur(bulan)",
       y = "Total Amount")
#cabang rendah disetiap umur
df_loan_mei_flag <- df_loan_mei_umur %>%
  group_by(umur) %>%
  mutate(Q1 = quantile(total_amount, 0.25),
         Q3 = quantile(total_amount, 0.75),
         IQR = (Q3-Q1)) %>%
  mutate(flag = ifelse(total_amount < (Q1 - IQR), 'rendah', 'baik'))
df_loan_mei_flag %>%
  filter(flag == 'rendah')
```

57:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · ~/

```
+   mutate(Q1 = quantile(total_amount, 0.25),
+          Q3 = quantile(total_amount, 0.75),
+          IQR = (Q3-Q1)) %>%
+   mutate(flag = ifelse(total_amount < (Q1 - IQR), 'rendah', 'baik'))
> df_loan_mei_flag %>%
+   filter(flag == 'rendah')
# A tibble: 2 × 8
# Groups:   umur [2]
  cabang pertama_cair  umur total_amount       Q1       Q3     IQR flag
  <chr>  <chr>        <dbl>        <int>    <dbl>    <dbl>   <dbl> <chr>
1 AE     2020-02-03       3     54200000 64580000 73900000 9320000 rendah
2 AL     2020-03-02       2     40650000 43580000 44590000 1010000 rendah
>
```

Insight

Let's examine the reasons behind the low performance at Branch with an age of 3 months. From the exploration results, we can see that the number of days and average loan are almost the same, but not the total disbursed loan. Let's take a closer look at the total disbursed loan within 1 month

Low Branch Performance in May

```r
        total_amount = as.numeric(total_amount))




#performa agen bulan mei
df_loan_mei_flag %>%
  filter(umur == 3, flag == "rendah") %>%
  inner_join(df_loan, by = "cabang") %>%
  filter(tanggal_cair >= '2020-05-01', tanggal_cair <= '2020-05-31') %>%
  group_by(cabang, agen) %>%
  summarise(jumlah_hari = n_distinct(tanggal_cair),
            total_loan_cair = n_distinct(loan_id),
            avg_amount = mean(amount), total_amount = sum(amount)) %>%
  arrange(total_amount)
```

```
+             avg_amount = mean(amount), total_amount = sum(amount)) %>%
+   arrange(total_amount)
`summarise()` has grouped output by 'cabang'. You can override using the `.groups`
argument.
# A tibble: 3 x 6
# Groups:   cabang [1]
  cabang agen  jumlah_hari total_loan_cair avg_amount total_amount
  <chr>  <chr>       <int>           <int>      <dbl>        <int>
1 AE     AE-3            4              16    310625       4970000
2 AE     AE-2           18              73    320274.     23380000
3 AE     AE-1           21              86    300581.     25850000
```

Good Branch Performance in May

Project Data Analysis for Finance Perfor...*

```r
            total_loan_cair = n_distinct(loan_id),
            avg_amount = mean(amount), total_amount = sum(amount)) %>%
  arrange(total_amount)


#perbandingan performa agen cabang paling baik umur 3 bulan AH
df_loan %>%
  filter(cabang == "AH") %>%
  filter(tanggal_cair >= '2020-05-01', tanggal_cair <= '2020-05-31') %>%
  group_by(cabang, agen) %>%
  summarise(jumlah_hari = n_distinct(tanggal_cair),
            total_loan_cair = n_distinct(loan_id),
            avg_amount = mean(amount),
            total_amount = sum(amount),
            .groups = "drop") %>%
  arrange(total_amount)
```

115:24 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · ~/

```
+   summarise(jumlah_hari = n_distinct(tanggal_cair),
+             total_loan_cair = n_distinct(loan_id),
+             avg_amount = mean(amount),
+             total_amount = sum(amount),
+             .groups = "drop") %>%
+   arrange(total_amount)
# A tibble: 3 × 6
  cabang agen  jumlah_hari total_loan_cair avg_amount total_amount
  <chr>  <chr>       <int>           <int>      <dbl>        <int>
1 AH     AH-3           19              74    303649.     22470000
2 AH     AH-1           21              81    301358.     24410000
3 AH     AH-2           21              86    313488.     26960000
>
```

Insight

Looking at the high performance of Branch AH, the lowest total disbursed loan is 74 with 19 days. Compare this with the low performance of Branch AE, where agent AE-3 has a total disbursed loan of only 16 and number of days only 4.

02

Summary & Recommendation

Summary & Recommendation

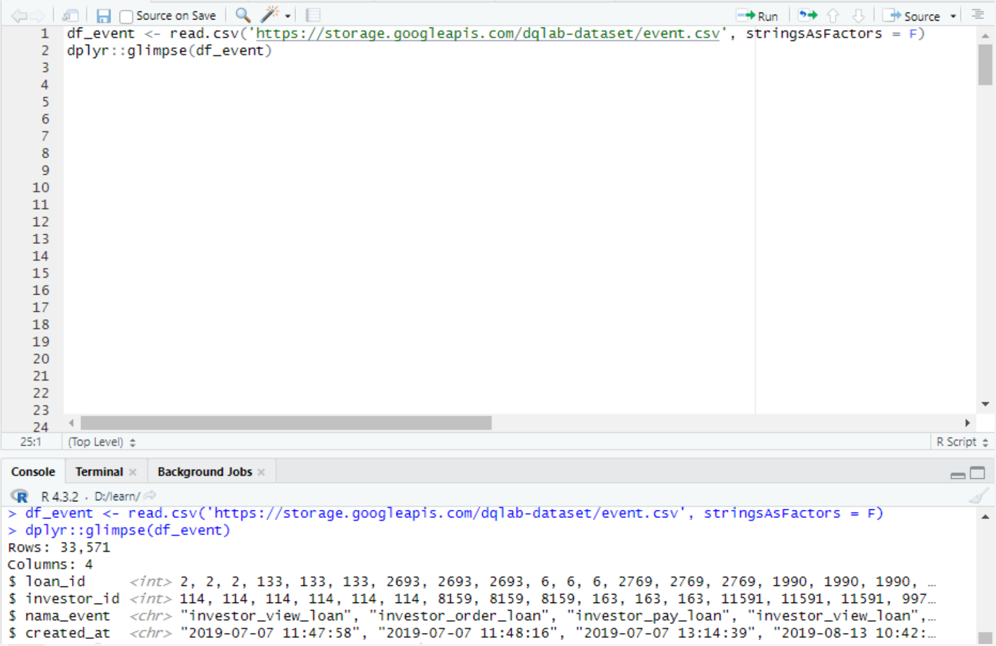
- When evaluating branch performance, we should not only focus on the total_amount but also consider how long the branch has been established, allowing for a more comprehensive analysis.
- From the previous exploration, it can be observed that the performance of each branch is influenced by the number of working days and loan disbursement. Therefore, to enhance performance, we can improve the agents' efficiency to increase monthly loan disbursements.

The Investor Investment Process

Background

- DQLab Finance is a peer-to-peer lending company, relying on investors for potential borrowers.
- Each prospective borrower applying for a loan will have their loan uploaded to the marketplace. Registered investors can then choose loans that align with their preferences

The Utilized Data



- Because the data is available in the RDS format, it can be directly read in R using the read.csv code

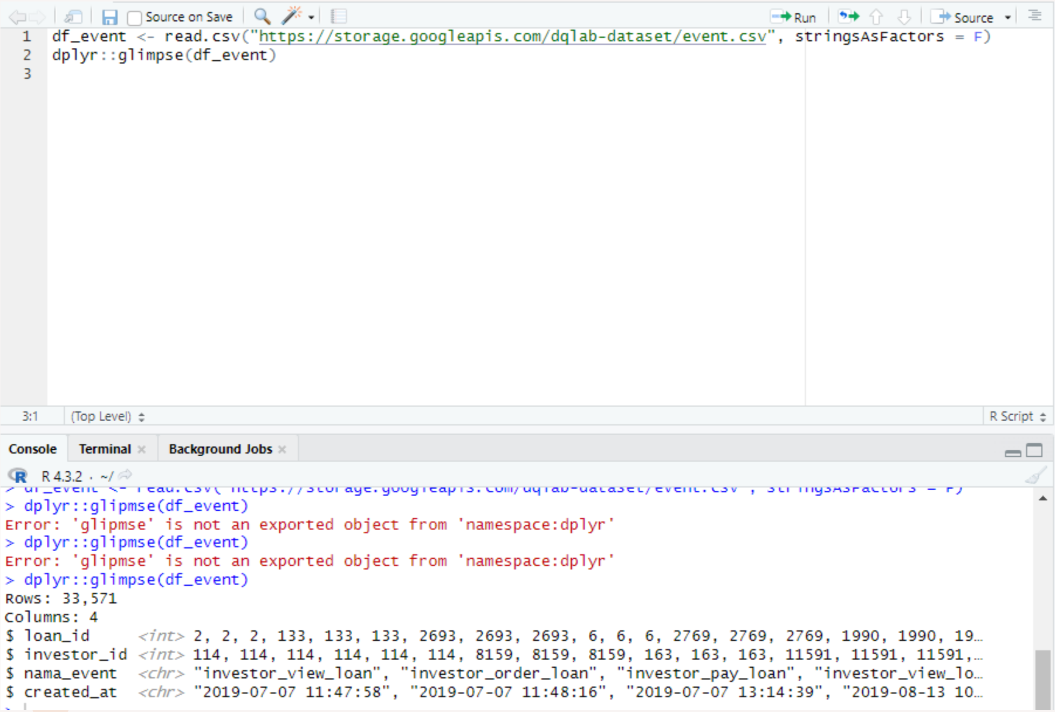
Tables in Data

loan_id	Unique ID of the loan uploaded to the marketplace, unique ID of the registered investor, activities performed by the investor, and changes in the loan status
Investor_id	Unique ID of the registered investor
Nama_event	activities performed by the investor and loan status changes
Created_at	Time the event occurred

01

Explore The Data

Change Data Type



- From the data, we can see that the data type of the created_at column is chr (character)
- To simplify the data exploration, we'll proceed with converting it to the timestamp data type

Summarised nama_event

```r
df_event <- read.csv('https://storage.googleapis.com/dqlab-dataset/event.csv', stringsAsFactors = F)
dplyr::glimpse(df_event)

#mengubah created_at menjadi timestamp
library(lubridate)
df_event$created_at <- ymd_hms(df_event$created_at)
dplyr::glimpse(df_event)

#summarise nama_event
library(dplyr)
df_event %>%
  group_by(nama_event) %>%
  summarise(jumlah_event = n(),
            loan = n_distinct(loan_id),
            investor = n_distinct(investor_id))
```

```
# A tibble: 5 x 4
  nama_event           jumlah_event  loan investor
  <chr>                       <int> <int>    <int>
1 investor_order_loan          3714  3641      804
2 investor_pay_loan            3632  3632      771
3 investor_register           17931     1    17931
4 investor_view_loan           4616  3678     1095
5 loan_to_marketplace          3678  3678        1
```

Since the data in the 'nama_event' consists of several events, let's summarise these events

Summarised Event Explanation

1. Investor_order_loan: event when an investor places an order for a loan, awaiting payment. The number of events does not match unique loans or unique investors because one loan can be ordered by more than one investor (if the previous order has not been paid).

2. Investor_pay_loan: event when an investor pays for a loan from a previous order. The number of events is the same as unique loans, indicating that one loan can only be paid by one investor. The number of investors is greater than the number of loans, suggesting that one investor can purchase multiple loans.

3. Investor_register: event when an investor registers. The number of events is the same as unique investors, indicating that each investor get registered once. There is one loan with NA because registration does not need a loan.

4. Investor_view_loan: event when an investor views loan details in the marketplace. The number of events does not match unique loans or unique investors because one loan can be viewed by multiple investors, and an investor can view the same loan multiple times

5. Loan_to_marketplace: event when a loan is uploaded to the marketplace. The number of events is the same as the number of loans because one loan can only be uploaded once. The number of investors is only 1, with NA as its content because uploading is not associated with an investor

Upload Loan to Marketplace Table

```r
df_event$created_at <- ymd_hms(df_event$created_at)
dplyr::glimpse(df_event)

#summarise nama_event
library(dplyr)
df_event %>%
  group_by(nama_event) %>%
  summarise(jumlah_event = n(),
            loan = n_distinct(loan_id),
            investor = n_distinct(investor_id))

#even loan di upload di marketplace
df_market_place <- df_event %>%
  filter(nama_event == 'loan_to_marketplace') %>%
  select(loan_id, marketplace=created_at)
df_market_place
```

21:16 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · D:/learn/

```
4 investor_view_loan          4616  3678       1095
5 loan_to_marketplace         3678  3678          1
> #even loan di upload di marketplace
> df_market_place <- df_event %>%
+   filter(nama_event == 'loan_to_marketplace') %>%
+   select(loan_id, marketplace=created_at)
> df_market_place
   loan_id         marketplace
1        1 2019-07-06 09:03:04
2        2 2019-07-06 09:00:00
3        3 2019-07-06 09:03:04
4        4 2019-07-06 09:03:04
5        5 2019-07-05 11:45:07
6        6 2019-07-08 16:35:28
```

Investor View Loan Table

```r
df_market_place <- df_event %>%
  filter(nama_event == 'loan_to_marketplace') %>%
  select(loan_id, marketplace=created_at)
df_market_place

#even investor melihat detail loan
df_view_loan <- df_event %>%
  filter(nama_event == "investor_view_loan") %>%
  group_by(loan_id, investor_id) %>%
  summarise(jumlah_view = n(),
            pertama_view = min(created_at),
            terakhir_view = max(created_at),
            .groups = 'drop')
df_view_loan
```

```
# A tibble: 4,309 × 5
   loan_id investor_id jumlah_view pertama_view        terakhir_view
     <int>       <int>       <int> <dttm>              <dttm>
 1       1         107           1 2019-07-07 11:48:11 2019-07-07 11:48:11
 2       2         114           1 2019-07-07 11:47:58 2019-07-07 11:47:58
 3       3          97           1 2019-07-06 09:50:00 2019-07-06 09:50:00
 4       4          97           1 2019-07-06 09:49:20 2019-07-06 09:49:20
 5       5         107           1 2019-07-05 12:54:25 2019-07-05 12:54:25
 6       6         163           1 2019-07-08 16:40:31 2019-07-08 16:40:31
 7       7         133           2 2019-07-14 11:04:46 2019-07-14 11:16:18
 8       8          71           1 2019-07-05 11:47:10 2019-07-05 11:47:10
 9       8          79           1 2019-07-05 12:05:14 2019-07-05 12:05:14
10       9          79           1 2019-07-05 12:09:43 2019-07-05 12:09:43
# ℹ 4,299 more rows
```

Event Investor Loan Orders and Payments

```r
              terakhir_view = max(created_at),
              .groups = 'drop')
df_view_loan


#even investor pesan dan bayar loan
library(dplyr)
library(tidyr)
df_order_pay <- df_event %>%
  filter(nama_event %in% c("investor_order_loan", "investor_pay_loan")) %>%
  group_by(loan_id, investor_id, nama_event) %>%
  spread(nama_event, created_at) %>%
  select(loan_id, investor_id, order=investor_order_loan, pay=investor_pay_loan)
df_order_pay
```

42:13 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · D:/learn/

```
# Groups:   loan_id, investor_id [3,714]
   loan_id investor_id order               pay                
     <int>       <int> <dttm>              <dttm>             
 1       1         107 2019-07-07 11:48:57 2019-07-07 12:02:18
 2       2         114 2019-07-07 11:48:16 2019-07-07 13:14:39
 3       3          97 2019-07-06 09:50:02 2019-07-06 10:14:44
 4       4          97 2019-07-06 09:49:23 2019-07-06 09:59:51
 5       5         107 2019-07-05 12:55:15 2019-07-05 13:55:54
 6       6         163 2019-07-08 16:42:03 2019-07-08 16:45:56
 7       7         133 2019-07-14 11:16:54 2019-07-14 11:22:00
 8       8          79 2019-07-05 12:06:21 2019-07-05 17:04:56
 9       9          79 2019-07-05 12:11:43 2019-07-05 17:04:52
10      10         107 2019-07-10 15:57:07 2019-07-10 16:19:07
```

Combine The Table

```r
  filter(nama_event %in% c("investor_order_loan", "investor_pay_loan")) %>%
  group_by(loan_id, investor_id, nama_event) %>%
  spread(nama_event, created_at) %>%
  select(loan_id, investor_id, order=investor_order_loan, pay=investor_pay_loan)
df_order_pay

#gabungan data loan investasi
df_loan_invest <- df_market_place %>%
  left_join(df_view_loan, by = 'loan_id') %>%
  left_join(df_order_pay, by = c("loan_id", "investor_id"))
df_loan_invest
```

```
      <int>       <int> <dttm>              <dttm>
 1        1         107 2019-07-07 11:48:57 2019-07-07 12:02:18
 2        2         114 2019-07-07 11:48:16 2019-07-07 13:14:39
 3        3          97 2019-07-06 09:50:02 2019-07-06 10:14:44
 4        4          97 2019-07-06 09:49:23 2019-07-06 09:59:51
 5        5         107 2019-07-05 12:55:15 2019-07-05 13:55:54
 6        6         163 2019-07-08 16:42:03 2019-07-08 16:45:56
 7        7         133 2019-07-14 11:16:54 2019-07-14 11:22:00
 8        8          79 2019-07-05 12:06:21 2019-07-05 17:04:56
 9        9          79 2019-07-05 12:11:43 2019-07-05 17:04:52
10       10         107 2019-07-10 15:57:07 2019-07-10 16:19:07
# ℹ 3,704 more rows
# ℹ Use `print(n = ...)` to see more rows
```

- Combine view_loan, loan_upload_to_marketplace, and order & pay loan tables into one, as each created_at in these tables differs.

Relationship Between Total Views and Orders

```r
#gabungan data loan investasi
df_loan_invest <- df_market_place %>%
  left_join(df_view_loan, by = 'loan_id') %>%
  left_join(df_order_pay, by = c("loan_id", "investor_id"))
df_loan_invest

#hubungan jumlah view dengan order
df_loan_invest %>%
  mutate(status_order = ifelse(is.na(order), "not_order", "order")) %>%
  count(jumlah_view, status_order) %>%
  spread(status_order, n, fill = 0) %>%
  mutate(persen_order = scales::percent(order/(order + not_order)))
```

55:68 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · D:/learn/

```
> df_loan_invest %>%
+   mutate(status_order = ifelse(is.na(order), "not_order", "order")) %>%
+   count(jumlah_view, status_order) %>%
+   spread(status_order, n, fill = 0) %>%
+   mutate(persen_order = scales::percent(order/(order + not_order)))
  jumlah_view not_order order persen_order
1           1       570  3513        86.0%
2           2        20   173        89.6%
3           3         3    23        88.5%
4           4         0     3       100.0%
5           5         1     1        50.0%
6           7         0     1       100.0%
7          40         1     0         0.0%
```

Insight

From the table, we can see that there is no specific correlation between the number of times a loan is viewed and the number of orders. According to the data, investors who have viewed loan details once are likely to place orders up to 86%.

The Time Taken to Order a Loan After Viewing Loan Details

```r
#waktu yang diperlukan investor untuk memesan sejak pertama kali melihat detail loan
library(dplyr)
library(tidyr)
df_loan_invest %>%
  filter(!is.na(order)) %>%
  mutate(lama_order_view = as.numeric(difftime(order, pertama_view, units = "mins"))) %>%
  group_by(jumlah_view) %>%
  summarise(total = n(), min = min(lama_order_view),
            median = median(lama_order_view),
            mean = mean(lama_order_view),
            max = max(lama_order_view)) %>%
  mutate_if(is.numeric, ~round(.,2))
```

```
+             median = median(lama_order_view),
+             mean = mean(lama_order_view),
+             max = max(lama_order_view)) %>%
+   mutate_if(is.numeric, ~round(.,2))
# A tibble: 6 x 6
  jumlah_view total     min  median    mean     max
        <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1           1  3513    0.03    1.35    2.97    79.6
2           2   173    0.43   22.1    61.1   2446.
3           3    23    7.25   32.0    66.4    495.
4           4     3   17.1    33.9    34.1     51.2
5           5     1 1113.   1113.   1113.   1113.
6           7     1  549.    549.    549.    549.
>
```

Insight

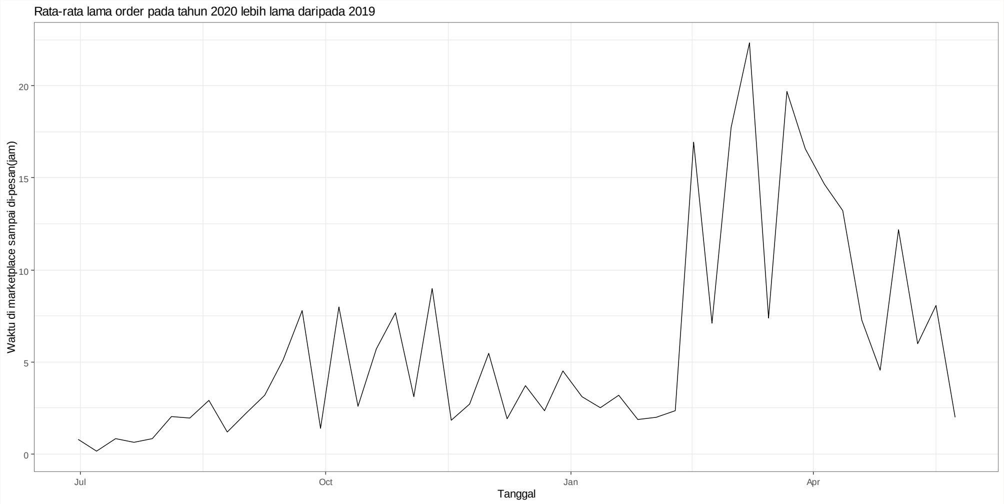
Interestingly, many investors who view loan details once place orders in less than 3 minutes. However, for those who view 2 or 3 times, there is an outlier, causing a significant gap between the mean and median, reaching up to 1 hour.

Average Time of Loan Ordering Since Uploaded Every Week

```r
#Rata-rata Loan dipesan sejak di upload setiap minggunya
library(dplyr)
library(lubridate)
library(ggplot2)
df_lama_order_per_minggu <- df_loan_invest %>%
  filter(!is.na(order)) %>%
  mutate(tanggal = floor_date(marketplace, "week"),
         lama_order = as.numeric(difftime(order, marketplace, units = "hour"))) %>%
  group_by(tanggal)%>%
  summarise(lama_order = median(lama_order))

ggplot(df_lama_order_per_minggu) +
  geom_line(aes(x=tanggal, y=lama_order)) +
  theme_bw()+
  labs(title = "Rata-rata lama order pada tahun 2020 lebih lama daripada 2019",
       x="Tanggal", y="Waktu di marketplace sampai di-pesan(jam)")
```

```r
> library(ggplot2)
> df_lama_order_per_minggu <- df_loan_invest %>%
+   filter(!is.na(order)) %>%
+   mutate(tanggal = floor_date(marketplace, "week"),
+          lama_order = as.numeric(difftime(order, marketplace, units = "hour"))) %>%
+   group_by(tanggal)%>%
+   summarise(lama_order = median(lama_order))
>
> ggplot(df_lama_order_per_minggu) +
+   geom_line(aes(x=tanggal, y=lama_order)) +
+   theme_bw()+
+   labs(title = "Rata-rata lama order pada tahun 2020 lebih lama daripada 2019",
+        x="Tanggal", y="Waktu di marketplace sampai di-pesan(jam)")
>
```

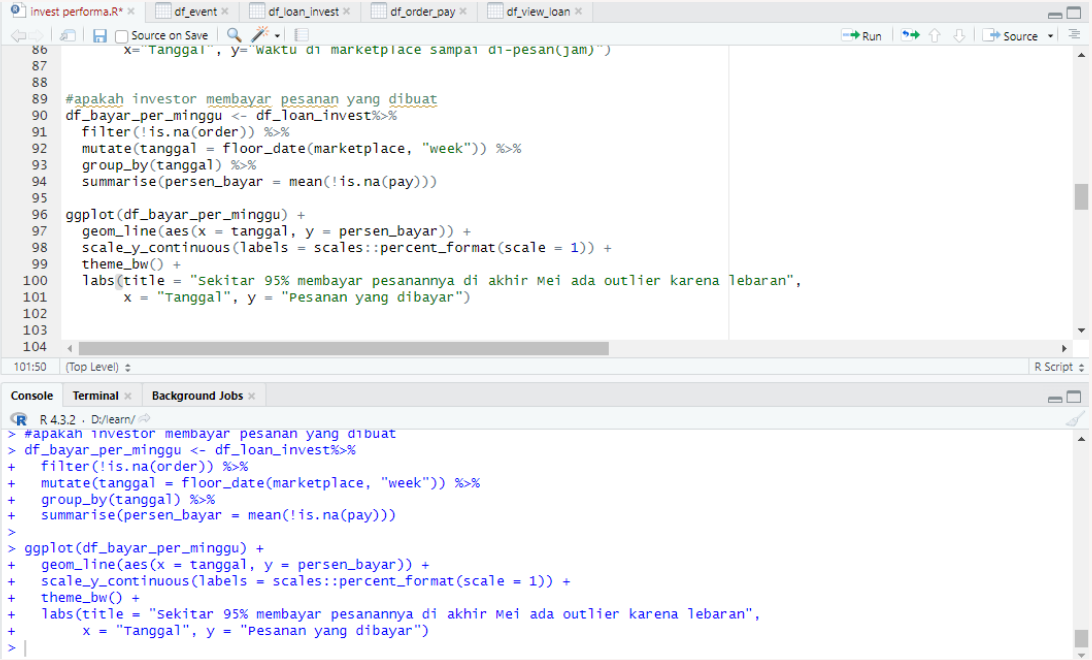
The Graph



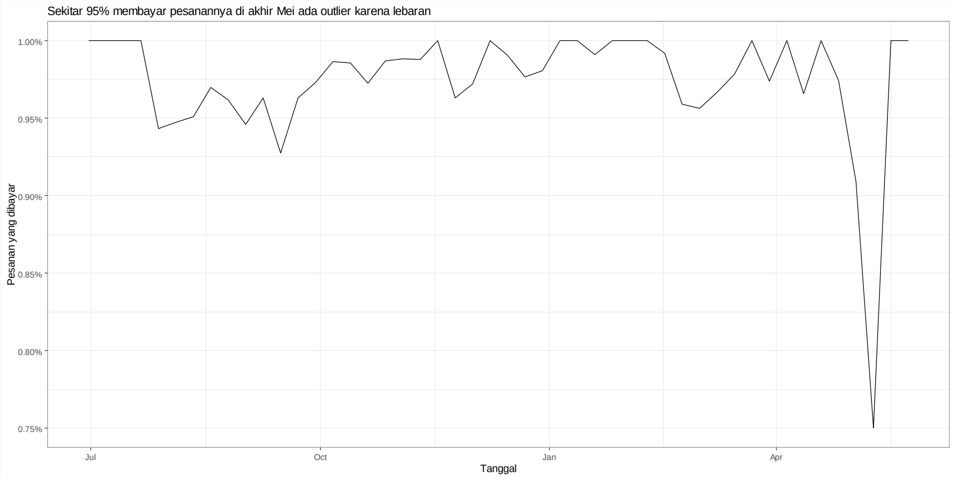
Insight

In the graph, a difference in the time taken for loan orders is noticeable between the years 2019 and 2020. The time required for loan orders in 2020 appears to be longer

Did Investors Pay for the Placed Orders



The Graph



Insight

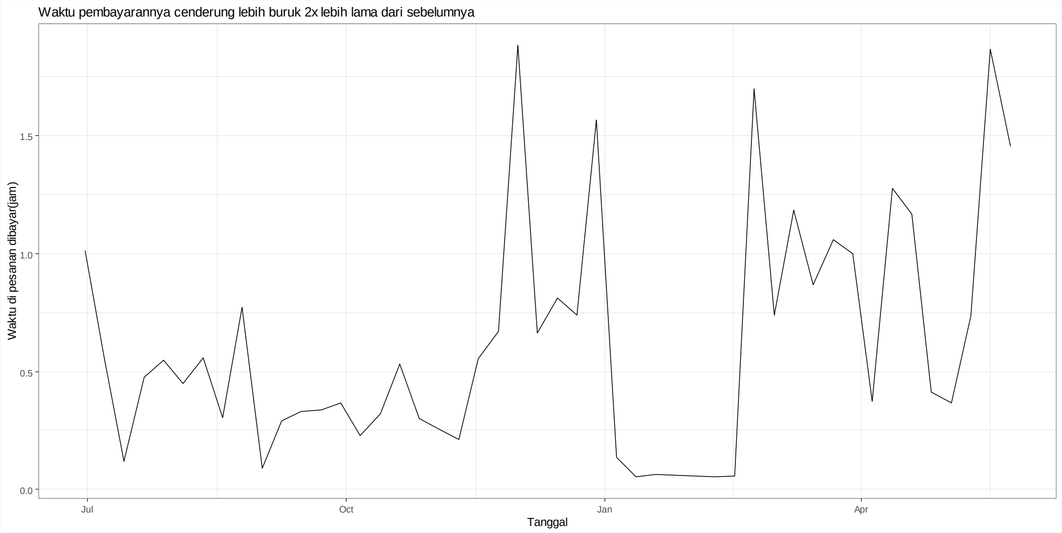
In the graph, it is evident that over 95% paid for their orders by the end of May. However, there is an outlier due to the Idul Fitri, resulting in investors rarely paying for their orders

The Time Taken by Investors to Pay for Their Orders

```r
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +
  theme_bw() +
  labs(title = "Sekitar 95% membayar pesanannya di akhir Mei ada outlier karena lebaran",
       x = "Tanggal", y = "Pesanan yang dibayar")


#waktu yang dibutuhkan investor dalam membayar
df_lama_bayar_per_minggu <- df_loan_invest %>%
  filter(!is.na(pay)) %>%
  mutate(tanggal = floor_date(marketplace, "week"),
         lama_bayar = as.numeric(difftime(pay, order, units = "hour"))) %>%
  group_by(tanggal)%>%
  summarise(lama_bayar = median(lama_bayar))

ggplot(df_lama_bayar_per_minggu) +
  geom_line(aes(x=tanggal, y=lama_bayar)) +
  theme_bw()+ labs(title = "Waktu pembayarannya cenderung lebih buruk 2x lebih lama dari sebelumnya",
                   x="Tanggal", y="Waktu di pesanan dibayar(jam)")

```

```
> #apakah investor membayar pesanan yang dibuat
> df_bayar_per_minggu <- df_loan_invest%>%
+   filter(!is.na(order)) %>%
+   mutate(tanggal = floor_date(marketplace, "week")) %>%
+   group_by(tanggal) %>%
+   summarise(persen_bayar = mean(!is.na(pay)))
>
> ggplot(df_bayar_per_minggu) +
+   geom_line(aes(x = tanggal, y = persen_bayar)) +
+   scale_y_continuous(labels = scales::percent_format(scale = 1)) +
+   theme_bw() +
+   labs(title = "Sekitar 95% membayar pesanannya di akhir Mei ada outlier karena lebaran",
+        x = "Tanggal", y = "Pesanan yang dibayar")
>
```

The Graph



Insight

The payment time taken by investors is worse in 2020 compared to 2019. This might be due to the pandemic, leading investors to reconsider their investment decisions and whether they want to pay for the placed orders or not

Trend Investor Register

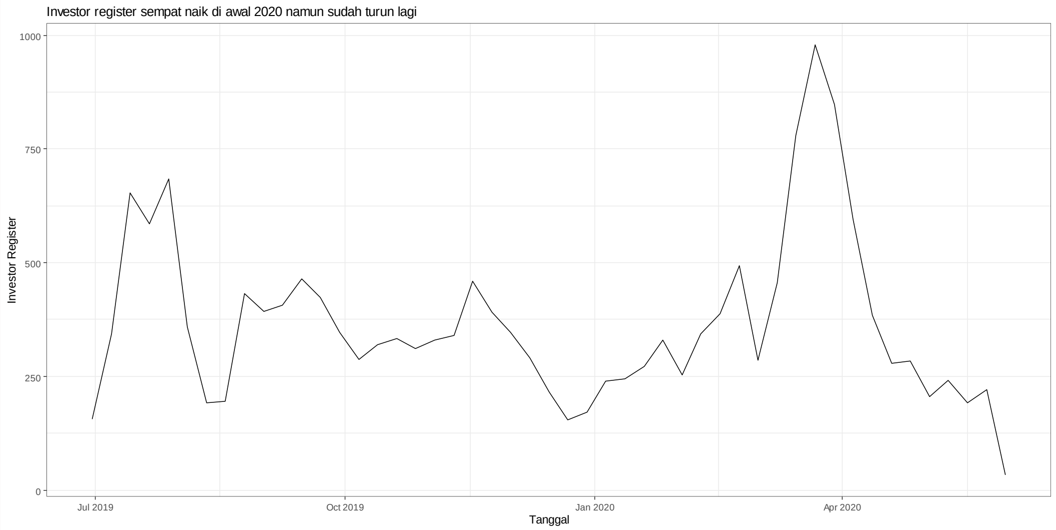
```r
  theme_bw()+ labs(title = "Waktu pembayarannya cenderung lebih buruk 2x lebih lama dari sebelumnya",
                  x="Tanggal", y="Waktu di pesanan dibayar(jam)")

#trend investor register
library(dplyr)
library(lubridate)
library(ggplot2)
df_investor_register <- df_event %>%
  filter(nama_event == "investor_register") %>%
  mutate(tanggal = floor_date(created_at, "week")) %>%
  group_by(tanggal) %>%
  summarise(investor = n_distinct(investor_id))

ggplot(df_investor_register) +
  geom_line(aes(x=tanggal, y=investor)) +
  theme_bw() +
  labs(title = "Investor register sempat naik di awal 2020 namun sudah turun lagi",
       x="Tanggal", y="Investor Register")
```

Console | Terminal | Background Jobs

R 4.3.2 · D:/learn/

```r
> library(lubridate)
> library(ggplot2)
> df_investor_register <- df_event %>%
+   filter(nama_event == "investor_register") %>%
+   mutate(tanggal = floor_date(created_at, "week")) %>%
+   group_by(tanggal) %>%
+   summarise(investor = n_distinct(investor_id))
>
> ggplot(df_investor_register) +
+   geom_line(aes(x=tanggal, y=investor)) +
+   theme_bw() +
+   labs(title = "Investor register sempat naik di awal 2020 namun sudah turun lagi",
+        x="Tanggal", y="Investor Register")
>
```

The Graph



Insight

Investor registrations showed a continuous increase in early 2020, but a significant and drastic decline occurred in late April 2020.

Trend in First-time Investor Investments

```r
  labs(title = "Investor register sempat naik di awal 2020 namun sudah turun lagi",
       x="Tanggal", y="Investor Register")

#trend investor investasi pertama
df_investor_pertama_invest <- df_event %>%
  filter(nama_event == "investor_pay_loan") %>%
  group_by(investor_id) %>%
  summarise(pertama_invest = min(created_at)) %>%
  mutate(tanggal = floor_date(pertama_invest, "week")) %>%
  group_by(tanggal) %>%
  summarise(investor = n_distinct(investor_id))

ggplot(df_investor_pertama_invest) +
  geom_line(aes(x=tanggal, y=investor)) +
  theme_bw() +
  labs(title = "Ada kenaikan trend investor pertama invest, namun turun drastis mulai Maret 2020",
       y="Investor pertama invest")
```

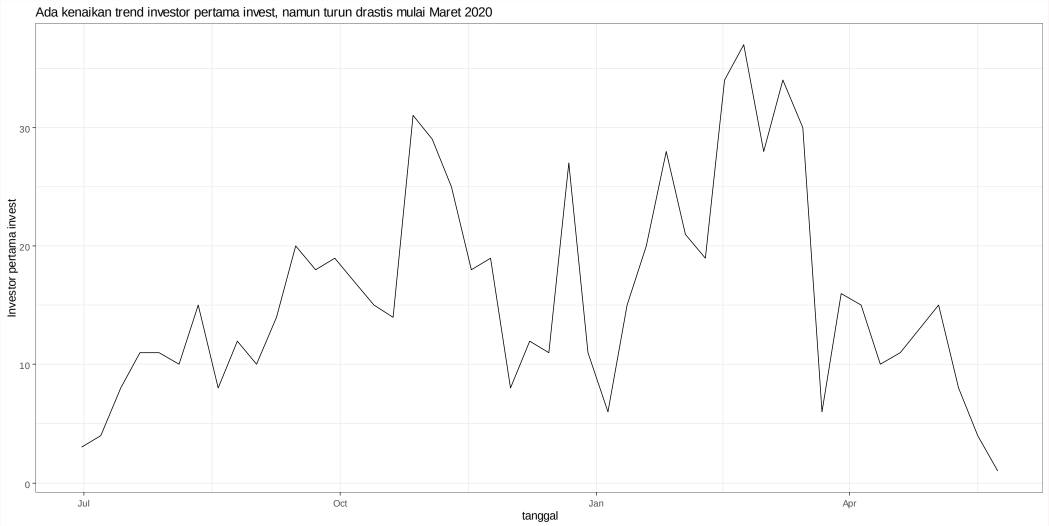
150:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · D:/learn/

```
> #trend investor investasi pertama
> df_investor_pertama_invest <- df_event %>%
+   filter(nama_event == "investor_pay_loan") %>%
+   group_by(investor_id) %>%
+   summarise(pertama_invest = min(created_at)) %>%
+   mutate(tanggal = floor_date(pertama_invest, "week")) %>%
+   group_by(tanggal) %>%
+   summarise(investor = n_distinct(investor_id))
> ggplot(df_investor_pertama_invest) +
+   geom_line(aes(x=tanggal, y=investor)) +
+   theme_bw() +
+   labs(title = "Ada kenaikan trend investor pertama invest, namun turun drastis mulai Maret 2020",
+        y="Investor pertama invest")
>
```

The Graph



Insight

Similarly with the trend of first-time investor investments increased at the beginning of 2020 but showed a drastic decline starting from March 2020

First Investment Cohort Based on Registration

```r
  theme_bw() +
  labs(title = "Ada kenaikan trend investor pertama invest, namun turun drastis mulai Maret 2020",
       y="Investor pertama invest")

#cohort pertama invest berdasarkan bulan register
library(dplyr)
library(lubridate)
library(tidyr)

df_register_per_investor <- df_event %>%
  filter(nama_event == "investor_register") %>%
  rename(tanggal_register = "created_at") %>%
  mutate(bulan_register = floor_date(tanggal_register, 'month')) %>%
  select(investor_id, tanggal_register, bulan_register)

df_pertama_invest_per_investor <- df_event %>%
  filter(nama_event == "investor_pay_loan") %>%
  group_by(investor_id) %>%
  summarise(pertama_invest = min(created_at))

df_register_per_investor %>%
  left_join(df_pertama_invest_per_investor, by = "investor_id") %>%
  mutate(lama_invest = as.numeric(difftime(pertama_invest, tanggal_register, units = "day")) %/% 30) %>%
  group_by(bulan_register, lama_invest) %>%
  summarise(investor_per_bulan = n_distinct(investor_id), .groups = "drop") %>%
  group_by(bulan_register) %>%
  mutate(register = sum(investor_per_bulan)) %>%
  filter(!is.na(lama_invest)) %>%
  mutate(invest = sum(investor_per_bulan)) %>%
  mutate(percent_invest = scales::percent(invest/register)) %>%
  mutate(breakdown_persen_invest = scales::percent(investor_per_bulan/invest)) %>%
  select(-investor_per_bulan) %>%
  spread(lama_invest, breakdown_persen_invest)
```

The Output

```
# A tibble: 11 x 14
# Groups:   bulan_register [11]
   bulan_register      register invest percent_invest `0`    `1`    `2`    `3`    `4`    `5`    `6`    `7`    `8`    `9`
   <dttm>                 <int>  <int> <chr>          <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>
 1 2019-07-01 00:00:00     2142     73 3%             61.6%  8.2%   6.8%   5.5%   1.4%   4.1%   5.5%   4.1%   1.4%   1.4%
 2 2019-08-01 00:00:00     1458     74 5%             55.4%  8.1%   14.9%  10.8%  4.1%   2.7%   4.1%   NA     NA     NA
 3 2019-09-01 00:00:00     1763     94 5%             67.0%  21.3%  4.3%   2.1%   3.2%   1.1%   1.1%   NA     NA     NA
 4 2019-10-01 00:00:00     1437     83 6%             77.1%  8.4%   4.8%   7.2%   1.2%   1.2%   NA     NA     NA     NA
 5 2019-11-01 00:00:00     1607     87 5%             75.9%  11.5%  9.2%   1.1%   1.1%   1.1%   NA     NA     NA     NA
 6 2019-12-01 00:00:00     1085     55 5%             69.1%  16.4%  7.3%   5.5%   1.8%   NA     NA     NA     NA     NA
 7 2020-01-01 00:00:00     1138     78 7%             78.2%  15.4%  3.8%   2.6%   NA     NA     NA     NA     NA     NA
 8 2020-02-01 00:00:00     1520    115 8%             86.09% 6.96%  6.09%  0.87%  NA     NA     NA     NA     NA     NA
 9 2020-03-01 00:00:00     2776     53 2%             94%    6%     NA     NA     NA     NA     NA     NA     NA     NA
10 2020-04-01 00:00:00     2034     51 3%             86%    14%    NA     NA     NA     NA     NA     NA     NA     NA
11 2020-05-01 00:00:00      971      8 1%             100%   NA     NA     NA     NA     NA     NA     NA     NA     NA
>
```

Insight

1. Cohort Analysis is a technique used to analyze and understand the behavior of a group of individuals over time, grouped based on their characteristics.
2. We can observe that the highest total registrations occurred in March 2020, but out of that, only 2% invested, a significantly lower percentage compared to the previous month, which had a conversion rate of over 7%, the highest conversion rate.
3. Generally, only 5% of investors out of all registrations will convert. And when they convert, the majority do so within the first month (less than 30 days) since registration.

Cohort Retention for Investments

```r
  group_by(bulan_register, lama_invest) %>%
  summarise(investor_per_bulan = n_distinct(investor_id), .groups = "drop") %>%
  group_by(bulan_register) %>%
  mutate(register = sum(investor_per_bulan)) %>%
  filter(!is.na(lama_invest)) %>%
  mutate(invest = sum(investor_per_bulan)) %>%
  mutate(percent_invest = scales::percent(invest/register)) %>%
  mutate(breakdown_persen_invest = scales::percent(investor_per_bulan/invest)) %>%
  select(-investor_per_bulan) %>%
  spread(lama_invest, breakdown_persen_invest)


#Cohort Retention Invest
df_investor_per_investor <- df_event %>%
  filter(nama_event == "investor_pay_loan") %>%
  rename(tanggal_invest = created_at) %>%
  select(investor_id, tanggal_invest)

df_pertama_invest_per_investor %>%
  mutate(bulan_pertama_invest = floor_date(pertama_invest, 'month')) %>%
  inner_join(df_investor_per_investor, by = "investor_id") %>%
  mutate(jarak_invest = as.numeric(difftime(tanggal_invest, pertama_invest, units = "day")) %/% 30) %>%
  group_by(bulan_pertama_invest, jarak_invest) %>%
  summarise(investor_per_bulan = n_distinct(investor_id)) %>%
  group_by(bulan_pertama_invest) %>%
  mutate(investor = max(investor_per_bulan)) %>%
  mutate(breakdown_persen_invest = scales::percent(investor_per_bulan/investor)) %>%
  select(-investor_per_bulan) %>%
  spread(jarak_invest, breakdown_persen_invest) %>%
  select(-`0`)
```

The Output

```
# A tibble: 11 x 11
# Groups:   bulan_pertama_invest [11]
   bulan_pertama_invest investor `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `10`
   <dttm>                  <int> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
 1 2019-07-01 00:00:00        31 25.8% 25.8% 19.4% 6.5%  16.1% 16.1% 19.4% 12.9% 3.2%
 2 2019-08-01 00:00:00        51 35.3% 19.6% 25.5% 19.6% 19.6% 21.6% 9.8%  2.0%  NA
 3 2019-09-01 00:00:00        70 25.7% 18.6% 18.6% 15.7% 18.6% 10.0% 1.4%  NA    NA
 4 2019-10-01 00:00:00        80 32.5% 28.8% 17.5% 23.7% 8.7%  6.2%  NA    NA    NA
 5 2019-11-01 00:00:00        99 30.3% 24.2% 24.2% 8.1%  7.1%  1.0%  NA    NA    NA
 6 2019-12-01 00:00:00        63 38.1% 30.2% 3.2%  4.8%  1.6%  NA    NA    NA    NA
 7 2020-01-01 00:00:00        71 32.4% 12.7% 4.2%  1.4%  NA    NA    NA    NA    NA
 8 2020-02-01 00:00:00       115 16.5% 3.5%  0.9%  NA    NA    NA    NA    NA    NA
 9 2020-03-01 00:00:00       102 10.8% 1.0%  NA    NA    NA    NA    NA    NA    NA
10 2020-04-01 00:00:00        58 5%    NA    NA    NA    NA    NA    NA    NA    NA
11 2020-05-01 00:00:00        31 NA    NA    NA    NA    NA    NA    NA    NA    NA
```

Insight

1. Cohort retention invest ialah memprediksi apakah investor akan kembali invest pada bulan selanjutnya setelah invest pertama.
2. Pada cohort yang kita dapat sebelumnya dapat dilihat pada bulan Februari investor paling banyak melakukan invest pertama dibandingkan bulan lainnya, namun pada bulan tersebut retention nya jelek dibanding bulan-bulan sebelumnya. Dimana pada bulan sebelumnya investor akan invest sekitar 30% setelah invest pertama mereka
3. Cohort yang paling stabil adalah pada bulan agustus 2019. Di sekitar angka 20% setiap bulannya, walaupun pada bulan ke-7 persentasenya ikut turun juga.

The Output

```
# A tibble: 11 x 11
# Groups:   bulan_pertama_invest [11]
   bulan_pertama_invest investor `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `10`
   <dttm>                  <int> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
 1 2019-07-01 00:00:00        31 25.8% 25.8% 19.4% 6.5%  16.1% 16.1% 19.4% 12.9% 3.2%
 2 2019-08-01 00:00:00        51 35.3% 19.6% 25.5% 19.6% 19.6% 21.6% 9.8%  2.0%  NA
 3 2019-09-01 00:00:00        70 25.7% 18.6% 18.6% 15.7% 18.6% 10.0% 1.4%  NA    NA
 4 2019-10-01 00:00:00        80 32.5% 28.8% 17.5% 23.7% 8.7%  6.2%  NA    NA    NA
 5 2019-11-01 00:00:00        99 30.3% 24.2% 24.2% 8.1%  7.1%  1.0%  NA    NA    NA
 6 2019-12-01 00:00:00        63 38.1% 30.2% 3.2%  4.8%  1.6%  NA    NA    NA    NA
 7 2020-01-01 00:00:00        71 32.4% 12.7% 4.2%  1.4%  NA    NA    NA    NA    NA
 8 2020-02-01 00:00:00       115 16.5% 3.5%  0.9%  NA    NA    NA    NA    NA    NA
 9 2020-03-01 00:00:00       102 10.8% 1.0%  NA    NA    NA    NA    NA    NA    NA
10 2020-04-01 00:00:00        58 5%    NA    NA    NA    NA    NA    NA    NA    NA
11 2020-05-01 00:00:00        31 NA    NA    NA    NA    NA    NA    NA    NA    NA
```

Insight

1. Cohort retention for investments involves predicting whether investors will reinvest in the following month after their first investment.
2. In the cohort we observed earlier, it can be seen that in February, the highest number of investors made their first investment compared to other months. However, the retention for that month is poor compared to the previous months, where investors would reinvest around 30% after their first investment.
3. The most stable cohort is in August 2019, maintaining around a 20% reinvestment rate each month, although the percentage also drops in the 7th month

02

Summary & Recommendation

Summary & Recommendation

- In general, DQLab Finance shows positive development with fluctuations occurring due to specific dates, influenced by external factors such as salary disbursement. Overall, 5% of the total monthly registered investors will make investments, with the majority occurring within the first 30 days after registration, and only a small portion in the second month. The chances of conversion in the following months are very slim. Therefore, it is crucial to ensure a smooth investor journey in the first month to encourage further conversions in DQLab Finance.

- Next, it is essential to examine the continuity of investments in the subsequent months. Generally, 30% of investors will reinvest in the following month. In February, the conversion rate was good at 7.57%, with a high number of conversions. However, the retention was only 16% of those who invested, and in the next month, it was only half of the previous months. It's possible that the observed trends are linked to the pandemic, so need a deeper analysis.

Thanks!

Do you have questions?

dindafebriani222@gmail.com
+628 5281878152

Rpubs Branch Performance
Rpubs Investor Investment Process