```
# Try to install package tidyverse

install.packages("tidyverse")

# Call functions in "tivyverse" every day

library(tidyverse)

#Add library: stringr

library(stringr)

#Add lubridate
library(lubridate)

# Add geosphere to calculate the Geographic distance:


library(sp)
library(geosphere)

#Create the RoRotrem Analysis by arrange the Data
#for
RoRoTrem.Analysis <-
  as_tibble(Sample.T1.1.Delivery) %>%

  # Choose the record which is below 21 Tons.
  filter(!is.na(Current.trans.cost),
           !actual.Gross.weight.Kg <=0)


# Insert the distance per Customer.
# by create the lookup table
# mutate(Cust.Look.up=df.customer$`Cidade Recebedor Mercadoria (SAP)`) %>%
# Lookup with customer data frame
RoRoTrem.Analysis <-RoRoTrem.Analysis %>%
  rowwise() %>%
  mutate(Distance.GPS = distm (c(df.GPS.site$lng[match(Fiscal.Unit,df.GPS.site$`Planta Origem`)],
                                 df.GPS.site$lat[match(Fiscal.Unit,df.GPS.site$`Planta Origem`)]),
                               c(df.GPS.customer$lng[match(Dest.Code,df.GPS.customer$`SAP Recebedor Mercadoria`)],
                                 df.GPS.customer$lat[match(Dest.Code,df.GPS.customer$`SAP Recebedor Mercadoria`)]
                               ), fun = distHaversine)/1000
  ) %>%

  mutate(First.geography=df.customer$Microrregiao[match(str_to_upper(Dest.City..SAP.),df.customer$`Cidade Recebedor Mercadoria`)]) %>%

  mutate(Second.geography=df.customer$Messorregiao[match(str_to_upper(Dest.City..SAP.),df.customer$`Cidade Recebedor Mercadoria`)]) %>%


  # Start to arrange the table by Date,
  arrange(transaction.Date,Distance.GPS) %>%

  # Consolidate the quantity per customer

  group_by(transport.Date,Fiscal.Unit,Dest.City..SAP.,Dest.State,First.geography,Second.geography,Dest.Code) %>%
  summarise(current.frei.cost = sum(Current.trans.cost),
            Distance.gps = mean(Distance.GPS),
            Total.pallet = sum(actual.pallet),
            Total.weight= sum(actual.Gross.weight.Kg))




  # arrage by date, distance, and frirst geography
  RoRoTrem.Analysis <-RoRoTrem.Analysis %>%
  arrange(transport.Date,Distance.gps,First.geography)


  # Check number of Rodotrem truck daily:
    RoRoTrem.Analysis <-RoRoTrem.Analysis %>%
  mutate(Nu.RoRo.cust = floor(Total.pallet / 48),
         Re.pal.cust = Total.pallet - Nu.RoRo.cust *48,
         Lookup.cust = paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),"BR012",sep="-"),
         Cost.cust = (df.cost.transportation$TARIFA.FRETE[match(Lookup.cust,df.cost.transportation$Look.Up)])*Nu.RoRo.cust
         ) %>%


    # create indication at the Microrregiao level:
      arrange(transport.Date,First.geography,desc(Re.pal.cust),Distance.gps) %>%
    group_by(transport.Date,First.geography) %>%
    mutate(Rank.Date=row_number()) %>%
    mutate(Max.Date=max(Rank.Date)) %>%
      mutate(Cumpallet=cumsum(Re.pal.cust)) %>%
    ungroup()

    # create indication at the Microrregiao level:

    RoRoTrem.Analysis <- RoRoTrem.Analysis %>%
      group_by(transport.Date) %>%
      mutate(Consol.micro =ifelse(Max.Date==1 | Rank.Date==2, Cumpallet,
                                   ifelse((Rank.Date %% 2 )==0,
                                          Cumpallet - lag(Cumpallet, n=2L, default = 0),
                                          ifelse(
                                            Rank.Date==Max.Date,
                                            Cumpallet - lag(Cumpallet, n=1L, default = 0),
                                            0
                                          )
                                   )
      )
      )


    # Check number of Rodotrem for case of consolidation in Microrregiao level:


    RoRoTrem.Analysis <-RoRoTrem.Analysis %>%
      mutate(Nu.RoRo.micro = ifelse(Consol.micro> 48*0.85,1,floor(Consol.micro / 48)),
             Re.pal.micro = ifelse((Consol.micro - Nu.RoRo.micro *48) <0,0,Consol.micro - Nu.RoRo.micro *48),
             Lookup.micro = paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),"BR012",sep="-"),
             Cost.micro = (df.cost.transportation$TARIFA.FRETE[match(Lookup.micro,df.cost.transportation$Look.Up)])*Nu.RoRo.micro
             )


      # Create the new data frame for consolidation in State

      RoRoTrem.Analysis_state <-RoRoTrem.Analysis %>%
      group_by(transport.Date,Fiscal.Unit,Dest.City..SAP.,Dest.State) %>%
      summarise(current.frei.cost = sum(current.frei.cost),
                Distance.gps = mean(Distance.gps),
                Total.pallet = sum(Re.pal.micro),
                Cost.cust = sum (Cost.cust),
                Cost.micro = sum(Cost.micro))%>%
      # Add all the cost at the state level
        group_by(Dest.State)%>%
      mutate(Total.cost.cust = sum (Cost.cust),
             Total.cost.micro = sum (Cost.micro),
             Total.current = sum (current.frei.cost)
         )
  #create the indicators
    RoRoTrem.Analysis_state <-RoRoTrem.Analysis_state %>%
    arrange(transport.Date,Dest.State,Distance.gps) %>%
  group_by(transport.Date,Dest.State) %>%
      filter(Total.pallet>0) %>% # remove the record with 0 value
    mutate(Rank.Date=row_number(Distance.gps)) %>%
    mutate(Max.Date=max(Rank.Date)) %>%
      mutate(Cumpallet=cumsum(Total.pallet)) %>% # add cum pallet
    ungroup()


   #consolidation at the state level

    RoRoTrem.Analysis_state <-RoRoTrem.Analysis_state %>%
    group_by(transport.Date) %>%
      mutate(Consol.state =ifelse(Max.Date==1 | Rank.Date==2, Cumpallet,
                                  ifelse((Rank.Date %% 2 )==0,
                                         Cumpallet - lag(Cumpallet, n=2L, default = 0),
                                         ifelse(
                                           Rank.Date==Max.Date,
                                           Cumpallet - lag(Cumpallet, n=1L, default = 0),
                                           0
                                         )
                                  )
      )
      )


    # number of Rodotrem for case of consolidation in state level:


    RoRoTrem.Analysis_state <-RoRoTrem.Analysis_state %>%
      mutate(Nu.RoRo.state = ifelse(Consol.state > 48*0.85,1,floor(Consol.state / 48)),
             Consolidation.case = ifelse((Rank.Date %% 2 )==0 ,1,0))

      RoRoTrem.Analysis_state <-RoRoTrem.Analysis_state %>%
    mutate( Re.pal.state = ifelse((Consol.state - Nu.RoRo.state *48) <0,0,Consol.state - Nu.RoRo.state *48),
             Final.remaing.qty = ifelse( Consolidation.case == 1 & Consol.state <48 ,0, Re.pal.state)
      )

# Final consolidation

      RoRoTrem.Analysis_state <-RoRoTrem.Analysis_state %>%
        group_by(transport.Date,Dest.State) %>%
      mutate (Consolidation = ifelse(Rank.Date == Max.Date, sum(Final.remaing.qty),0)) %>%

      # Calculate the transportation for consolidation at the state
      mutate(S.Truck.size_State= ifelse( Consolidation.case == 1 & Consol.state <48 ,
               (Re.pal.state>0)*(Re.pal.state<=1.5)*1
             +(Re.pal.state>1.5)*(Re.pal.state<=4.5)*4
             +(Re.pal.state>4.5)*(Re.pal.state<=8.5)*8
             +(Re.pal.state>8.5)*(Re.pal.state<=12.5)*12
             +(Re.pal.state>12.5)*(Re.pal.state<=28.5)*28
             +(Re.pal.state>28.5)*48, # Assign the suitatble truck for remaining
             0)) %>%

        mutate(S.Truck.code=ifelse(S.Truck.size_State==0,"0", df.trucksize$SAP.Replace.Code[match(S.Truck.size_State,df.trucksize$Pallets)])) %>%
        mutate( Lookup.state.rodo = paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),"BR012",sep="-"),
                Lookup.state.s = paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),S.Truck.code,sep="-"),
                Cost.state = ifelse(S.Truck.code=="0",0,(df.cost.transportation$TARIFA.FRETE[match(Lookup.state.rodo,df.cost.transportation$Look.Up)]*Nu.RoRo.state) +
                              df.cost.transportation$TARIFA.FRETE[match(Lookup.state.s,df.cost.transportation$Look.Up)]
                               )
                               )

        # Calculate the transportation for final consolidation

        RoRoTrem.Analysis_state <-RoRoTrem.Analysis_state %>%

        mutate(Nu.RoRo.final = floor(Consolidation / 48)) %>%
        mutate(Remaining.Qty.final = Consolidation - Nu.RoRo.final *48) %>%
        mutate(S.Truck.size.final=
                 (Remaining.Qty.final>0)*(Remaining.Qty.final<=1.5)*1
               +(Remaining.Qty.final>1.5)*(Remaining.Qty.final<=4.5)*4
               +(Remaining.Qty.final>4.5)*(Remaining.Qty.final<=8.5)*8
               +(Remaining.Qty.final>8.5)*(Remaining.Qty.final<=12.5)*12
               +(Remaining.Qty.final>12.5)*(Remaining.Qty.final<=28.5)*28
               +(Remaining.Qty.final>28.5)*48) %>% # Assign the suitatble truck for remaining

        mutate(S.Truck.code.f=ifelse(Consolidation==0,"0", df.trucksize$SAP.Replace.Code[match(S.Truck.size.final,df.trucksize$Pallets)])) %>%
        mutate( Lookup.final.rodo = paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),"BR012",sep="-"),
                Lookup.final.s = paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),S.Truck.code.f,sep="-"),
                Cost.final = ifelse(S.Truck.code.f=="0",0,(df.cost.transportation$TARIFA.FRETE[match(Lookup.final.rodo,df.cost.transportation$Look.Up)]*Nu.RoRo.final) +
                                      df.cost.transportation$TARIFA.FRETE[match(Lookup.final.s,df.cost.transportation$Look.Up)]
                )
        )
              # Create the summary table for state
```

```
   # create indication at the state level:
  arrange(transport.Date,Dest.State,desc(Re.pal.Cust),Distance.gps) %>%
  group_by(transport.Date,First.Dest.State) %>%
  mutate(Rank.Date=row_number()) %>%
  mutate(Max.Date=max(Rank.Date)) %>%
  mutate(Cumpallet=cumsum(Re.pal.Cust)) %>%
  ungroup()
```

```r
  # arrage by date, distance, and frirst geography






  #  Create the Acccumulative weight by date.
  group_by(transaction.Date,Second.geography) %>%
  mutate(Cumweight=cumsum(actual.Gross.weight.Kg)) %>%
  ungroup() %>%



  # Add the Ranks for the date
  group_by(transaction.Date) %>%
  mutate(Rank.Date=row_number()) %>%
  mutate(Max.Date=max(Rank.Date)) %>%
  ungroup()



# Link with SKus to have CBM, Qty, MT and Pallet.

#RoRoTrem.Analysis <- RoRoTrem.Analysis %>%
# left_join(Sum.Ship.Order2)

#  Create the Acccumulative pallet by date.
RoRoTrem.Analysis <- RoRoTrem.Analysis %>%
  group_by(transaction.Date,Second.geography) %>%
  mutate(Cumpallet=cumsum(actual.pallet))


#



#Consolidation Case to group the volume per customer by,
# 2 shipments into 1

RoRoTrem.Analysis <- RoRoTrem.Analysis %>%
  group_by(transaction.Date) %>%
  mutate(Consolidation =ifelse(Max.Date==1 | Rank.Date==2, Cumpallet,
                               ifelse((Rank.Date %% 2 )==0,
                                      Cumpallet - lag(Cumpallet, n=2L, default = 0),
                                      ifelse(
                                        Rank.Date==Max.Date,
                                        Cumpallet - lag(Cumpallet, n=1L, default = 0),
                                        0
                                      )
                               )
  )
  )


# of Roro Truck
RoRoTrem.Analysis <-RoRoTrem.Analysis %>%
  mutate(Nu.RoRo = floor(Consolidation / 48)) %>%
  mutate(Remaining.Qty = Consolidation - Nu.RoRo *48) %>%
  mutate(S.Truck.Vol=
           (Remaining.Qty>0)*(Remaining.Qty<=1.5)*1
         +(Remaining.Qty>1.5)*(Remaining.Qty<=4.5)*4
         +(Remaining.Qty>4.5)*(Remaining.Qty<=8.5)*8
         +(Remaining.Qty>8.5)*(Remaining.Qty<=12.5)*12
         +(Remaining.Qty>12.5)*(Remaining.Qty<=28.5)*28
         +(Remaining.Qty>28.5)*48) %>% # Assign the suitatble truck for remaining

  mutate(S.Truck.code=df.trucksize$SAP.Replace.Code[match(S.Truck.Vol,df.trucksize$Pallets)]) %>%

  # Create the lookup columns in transport data following the format
  #1208-ARARAS-TRANSFERENCIA-BR275-100223303
  # Origin Code (1208)
  #-> Desination City (ARARAS in capital)
  #-> Type of Transport (TRANSFERENCIA)
  #-> Vehicle Type SAP (BR275)
  #-> Carrer Name (100223303)

  mutate(S.Lookup.re= paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),S.Truck.code,sep="-")) %>%

  mutate(S.Lookup.RO= paste(Fiscal.Unit,str_to_upper(Dest.City..SAP.),"BR012",sep="-")) %>%

  # Calculate the  transportation Cost.
  mutate(S_trans.cost=ifelse(Consolidation ==0,0,
                             (df.cost.transportation$TARIFA.FRETE[match(S.Lookup.re,df.cost.transportation$Look.Up)])
                             +(df.cost.transportation$TARIFA.FRETE[match(S.Lookup.RO,df.cost.transportation$Look.Up)])*Nu.RoRo)) %>%
  mutate (Cost.different = S_trans.cost - Current.trans.cost)

# Check error
Check.erro.Rodo = RoRoTrem.Analysis %>%
  filter(is.na(S_trans.cost))
write.csv(Check.erro.Rodo, file = "C:/Users/Bao/Desktop/errorRodo.csv", row.names = FALSE)

sum(RoRoTrem.Analysis$S_trans.cost)
sum(RoRoTrem.Analysis$Current.trans.cost)
# export to Excel File:

write.csv(RoRoTrem.Analysis, file = "C:/Users/Bao/Desktop/Rodo.Analysis5.csv", row.names = FALSE)
```