EXPLORING KEY FACTORS BEHIND LIFE INSURANCE POLICY LAPSES

## EXPLORING KEY FACTORS BEHIND LIFE INSURANCE POLICY LAPSES AND WHAT INSURERS CAN DO TO PREVENT IT

The below are the step by step process i used for the process

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

life\_insurance\_data <-read\_csv("life insurance data.csv")

## Rows: 185560 Columns: 20  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (7): SEX, POLICY TYPE 3, PAYMENT MODE, POLICY STATUS, NON LAPSE GUARANT...  
## dbl (13): CHANNEL1, CHANNEL2, CHANNEL3, ENTRY AGE, POLICY TYPE 1, POLICY TYP...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## inspecting the structure of the data

glimpse(life\_insurance\_data) ## View column types and samples

## Rows: 185,560  
## Columns: 20  
## $ CHANNEL1 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ CHANNEL2 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ CHANNEL3 <dbl> 1, 1, 2, 3, 4, 1, 1, 5, 2, 3, 1, 1, 1, 3, …  
## $ `ENTRY AGE` <dbl> 36, 42, 40, 39, 44, 36, 50, 42, 37, 34, 31…  
## $ SEX <chr> "F", "M", "M", "M", "M", "M", "F", "M", "M…  
## $ `POLICY TYPE 1` <dbl> 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, …  
## $ `POLICY TYPE 2` <dbl> 1, 2, 3, 2, 2, 3, 3, 2, 3, 3, 2, 2, 3, 3, …  
## $ `POLICY TYPE 3` <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A…  
## $ `PAYMENT MODE` <chr> "Annually", "Annually", "Annually", "Month…  
## $ `POLICY STATUS` <chr> "Inforce", "Inforce", "Lapse", "Lapse", "L…  
## $ BENEFIT <dbl> 200000, 100000, 80000, 100000, 100000, 400…  
## $ `NON LAPSE GUARANTEED` <chr> "NO NLG", "NO NLG", "NO NLG", "NO NLG", "N…  
## $ `SUBSTANDARD RISK` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ `NUMBER OF ADVANCE PREMIUM` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ `INITIAL BENEFIT` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ `Full Benefit?` <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N…  
## $ `Policy Year (Decimal)` <dbl> 8.000000, 8.000000, 8.000000, 8.000000, 7.…  
## $ `Policy Year` <dbl> 9, 9, 9, 9, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, …  
## $ Premium <dbl> 280, 200, 1289, 216, 140, 336, 537, 200, 8…  
## $ `Issue Date` <chr> "11-Aug", "11-Aug", "11-Aug", "11-Aug", "1…

summary(life\_insurance\_data) ##See min, max, mean, NA counts

## CHANNEL1 CHANNEL2 CHANNEL3 ENTRY AGE   
## Min. :1.0 Min. :1.000 Min. : 0.000 Min. : 0.00   
## 1st Qu.:2.0 1st Qu.:2.000 1st Qu.: 0.000 1st Qu.:25.00   
## Median :2.0 Median :3.000 Median : 0.000 Median :32.00   
## Mean :3.6 Mean :2.572 Mean : 7.966 Mean :32.12   
## 3rd Qu.:6.0 3rd Qu.:3.000 3rd Qu.:11.000 3rd Qu.:41.00   
## Max. :8.0 Max. :3.000 Max. :82.000 Max. :70.00   
##   
## SEX POLICY TYPE 1 POLICY TYPE 2 POLICY TYPE 3   
## Length:185560 Min. : 1.000 Min. : 1.00 Length:185560   
## Class :character 1st Qu.: 3.000 1st Qu.: 6.00 Class :character   
## Mode :character Median : 3.000 Median :13.00 Mode :character   
## Mean : 5.033 Mean :21.89   
## 3rd Qu.: 6.000 3rd Qu.:37.00   
## Max. :20.000 Max. :88.00   
##   
## PAYMENT MODE POLICY STATUS BENEFIT NON LAPSE GUARANTEED  
## Length:185560 Length:185560 Min. : 20 Length:185560   
## Class :character Class :character 1st Qu.: 800 Class :character   
## Mode :character Mode :character Median : 12000 Mode :character   
## Mean : 23447   
## 3rd Qu.: 20000   
## Max. :5000000   
## NA's :23664   
## SUBSTANDARD RISK NUMBER OF ADVANCE PREMIUM INITIAL BENEFIT   
## Min. :-99.0000 Min. :0.000000 Min. : 0.0   
## 1st Qu.: 0.0000 1st Qu.:0.000000 1st Qu.: 0.0   
## Median : 0.0000 Median :0.000000 Median : 0.0   
## Mean : 0.2528 Mean :0.000517 Mean : 170.3   
## 3rd Qu.: 0.0000 3rd Qu.:0.000000 3rd Qu.: 0.0   
## Max. :250.0000 Max. :5.000000 Max. :266017.2   
##   
## Full Benefit? Policy Year (Decimal) Policy Year Premium   
## Length:185560 Min. :0.000 Min. :1.000 Min. : -111   
## Class :character 1st Qu.:1.833 1st Qu.:2.000 1st Qu.: 120   
## Mode :character Median :3.250 Median :4.000 Median : 432   
## Mean :3.284 Mean :3.827 Mean : 1273   
## 3rd Qu.:4.667 3rd Qu.:5.000 3rd Qu.: 960   
## Max. :8.000 Max. :9.000 Max. :444000   
## NA's :77127   
## Issue Date   
## Length:185560   
## Class :character   
## Mode :character   
##   
##   
##   
##

colnames(life\_insurance\_data) ##checking for the column names

## [1] "CHANNEL1" "CHANNEL2"   
## [3] "CHANNEL3" "ENTRY AGE"   
## [5] "SEX" "POLICY TYPE 1"   
## [7] "POLICY TYPE 2" "POLICY TYPE 3"   
## [9] "PAYMENT MODE" "POLICY STATUS"   
## [11] "BENEFIT" "NON LAPSE GUARANTEED"   
## [13] "SUBSTANDARD RISK" "NUMBER OF ADVANCE PREMIUM"  
## [15] "INITIAL BENEFIT" "Full Benefit?"   
## [17] "Policy Year (Decimal)" "Policy Year"   
## [19] "Premium" "Issue Date"

**## Checking for duplicate and removing duplicate**

any(duplicated(life\_insurance\_data)) ## check for duplicate

## [1] TRUE

sum(duplicated(life\_insurance\_data)) ## gives the total number of duplicates in the data

## [1] 11884

life\_insurance\_unique <-life\_insurance\_data[!duplicated(life\_insurance\_data),] ## removing duplicate

## **# checking for missing values in life\_insurance\_unique data**

anyNA(life\_insurance\_unique)

## [1] TRUE

colSums(is.na(life\_insurance\_unique)) ## show column names with count of missing values

## CHANNEL1 CHANNEL2 CHANNEL3   
## 0 0 0   
## ENTRY AGE SEX POLICY TYPE 1   
## 0 0 0   
## POLICY TYPE 2 POLICY TYPE 3 PAYMENT MODE   
## 0 0 0   
## POLICY STATUS BENEFIT NON LAPSE GUARANTEED   
## 0 21548 0   
## SUBSTANDARD RISK NUMBER OF ADVANCE PREMIUM INITIAL BENEFIT   
## 0 0 0   
## Full Benefit? Policy Year (Decimal) Policy Year   
## 0 0 0   
## Premium Issue Date   
## 72390 0

## 

## The entry age to sign up to a life insurance policy is 18 years or more . so i removed ages less than 18 years from “life\_insurance\_unique” data set

## removing ages less than 18 and filtering ages greater than 18

## life\_insurance\_unique <- life\_insurance\_unique %>% filter(`ENTRY AGE` >= 18)

From results of the colsum of the data set earlier , there were missing values in benefit and premium. i will not be needing them in my analysis ,so i took them out

life\_insurance\_reduced <- subset(life\_insurance\_unique, select = -c(BENEFIT, Premium))

## Renaming column names example , entry age =age ,payment mode =payment frequency and payment year=payment term)

library(dplyr)  
life\_insurance\_reduced <- life\_insurance\_reduced %>%  
 rename(  
 age =`ENTRY AGE`,  
 payment\_frequency = `PAYMENT MODE`,  
 policy\_term =`Policy Year`)

## 

## Renaming the new data set created to life\_insurance\_rename

life\_insurance\_rename = life\_insurance\_reduced

The ages in the data set aren’t in range , so placing these ages into range ## grouping ages into range

life\_insurance\_rename <- life\_insurance\_rename %>%  
 mutate(age\_group = cut(age,  
 breaks = c(18, 30, 50, Inf),  
 labels = c("18–30", "31–50", "51+")))

## renaming columnames to lowercase

names(life\_insurance\_rename) <- tolower(names(life\_insurance\_rename))

## checking the summary of new age range to inspect it

summary(life\_insurance\_rename$age\_group)

## 18–30 31–50 51+ NA's   
## 56289 83538 12821 1618

From the summary , i discovered that the are missing values in the age group. ##removing missing values in the age group

life\_insurance\_clean <- life\_insurance\_rename%>%  
   
 filter(!is.na(age) & age >= 18 & age <= 100) %>%  
   
 mutate(age\_group = cut(  
 age,  
 breaks = c(17, 30, 50,Inf),  
 labels = c("18–30", "31–50", "51+")  
 ))

## now checking if there are still missing values in the age\_group

summary(life\_insurance\_clean$age\_group)

## 18–30 31–50 51+   
## 57907 83538 12821

from the results there are no missing values in the age\_group

## to calculate lapse rate , i need to create a lapse variable, creating the lapse variable

life\_insurance\_lapse <- life\_insurance\_clean %>%  
 filter(`policy status` != "Death") %>%   
 mutate(`policy status` = case\_when(  
 `policy status`%in% c("Lapse", "Surrender", "Expired") ~ "Lapsed",  
 `policy status` == "Inforce" ~ "Active",  
 TRUE ~ "Other"  
 ))

## calculating key question 1 (lapse rate by age group)

## Counting and Calculating Lapse Rate per Age Group

Count policies by age group and lapse

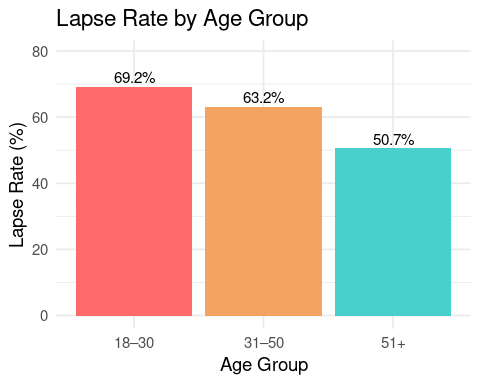
lapse\_by\_age <- life\_insurance\_lapse %>%  
 group\_by(age\_group, `policy status`) %>%  
 summarise(count = n(), .groups = "drop")

## Calculate lapse rate by age group

lapse\_rate\_by\_age <- lapse\_by\_age %>%  
 group\_by(age\_group) %>%  
 mutate(  
 total = sum(count),  
 lapse\_rate = round((count / total) \* 100, 1)  
 ) %>%  
 filter( `policy status` == "Lapsed") # keep only lapse rate rows

## visualizing lapse\_rate by age\_group

ggplot(lapse\_rate\_by\_age, aes(x = age\_group, y = lapse\_rate, fill = age\_group)) +  
 geom\_col() +  
 geom\_text(  
 aes(label = paste0(round(lapse\_rate, 1), "%")),  
 vjust = -0.3,   
 size = 4  
 ) +  
 scale\_fill\_manual(values = c("indianred1", "sandybrown", "mediumturquoise")) +  
 labs(  
 title = "Lapse Rate by Age Group",  
 x = "Age Group",   
 y = "Lapse Rate (%)"  
 ) +  
 ylim(0, max(lapse\_rate\_by\_age$lapse\_rate, na.rm = TRUE) + 10) + # Add 10% headroom  
 theme\_minimal(base\_size = 14) +  
 theme(legend.position = "none")

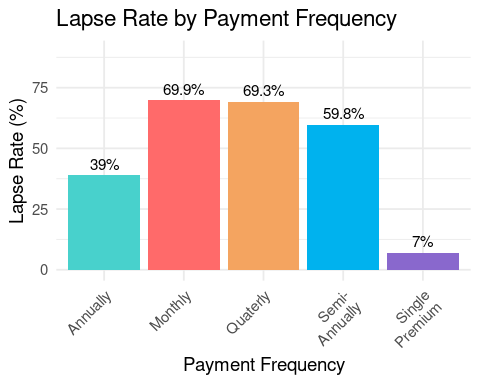


key question 2 : Finding lapse rate by payment frequency

lapse\_rate\_by\_freq <- life\_insurance\_lapse %>%  
 group\_by(payment\_frequency, `policy status` ) %>%  
 summarise(count = n(), .groups = "drop") %>%  
 group\_by(payment\_frequency) %>%  
 mutate(  
 total = sum(count),  
 lapse\_rate = round(count / total \* 100, 1)  
 ) %>%  
 filter(`policy status`== "Lapsed") # only need the Lapsed rows

## visualizing lapse rate by payment frequency

ggplot(lapse\_rate\_by\_freq, aes(x = payment\_frequency, y = lapse\_rate, fill = payment\_frequency)) +  
 geom\_col() +  
 geom\_text(  
 aes(label = paste0(lapse\_rate, "%")), vjust = -0.5, size = 4) + # Labels on top of bars  
 scale\_fill\_manual(values = c(  
 "Monthly" = "indianred1",  
 "Quaterly" = "sandybrown" ,  
 "Annually" = "mediumturquoise",  
 "Semiannually" = "deepskyblue2",  
 "Single Premium" = "mediumpurple3"   
 )) +  
 scale\_x\_discrete(labels = c(  
 "Monthly" = "Monthly",  
 "Quarterly" = "Quarterly",  
 "Annually" = "Annually",  
 "Semiannually" = "Semi-\nAnnually",  
 "Single Premium" = "Single\nPremium"  
 )) +  
 labs(  
 title = "Lapse Rate by Payment Frequency",  
 x = "Payment Frequency",  
 y = "Lapse Rate (%)"  
 ) +  
 ylim(0, max(lapse\_rate\_by\_freq$lapse\_rate) + 20) +  
 theme\_minimal(base\_size = 14) +  
 theme(  
 legend.position = "none",  
 axis.text.x = element\_text(angle = 45, hjust = 1)  
 )



## key question 3

##To Calculating Effect of Original Policy Term on Lapse Rate ,i need to group\_term into term\_group

life\_insurance\_term <- life\_insurance\_lapse %>%  
 mutate(  
 term\_group = if\_else( ### grouping to <=5 & >5year term  
 policy\_term <= 5,  
 "Short (≤5 yrs)",  
 "Long (>5 yrs)"  
 )  
 )

### Count policies by term\_group and lapses

term\_counts <- life\_insurance\_term %>%  
 group\_by(term\_group, `policy status`) %>%  
 summarise(count = n(), .groups = "drop")

##Calculating total policies per term\_group and compute lapse rate:

lapse\_rate\_by\_term <- term\_counts %>%  
 group\_by(term\_group) %>%  
 mutate(  
 total = sum(count), # total policies in each term\_group  
 lapse\_rate = round(count / total \* 100, 1) # % that are lapsed  
 ) %>%  
 filter(`policy status` == "Lapsed") # focus on lapse rates only

## visualizing term\_group by lapse\_group

ggplot(lapse\_rate\_by\_term, aes(x = term\_group, y = lapse\_rate, fill = term\_group)) +  
 geom\_col() +  
 geom\_text(  
 aes(label = paste0(lapse\_rate,"%")),  
 vjust = -0.8,   
 size = 4  
 ) +   
 scale\_fill\_manual(values = c(  
 "Short (≤5 yrs)" = "indianred1",  
 "Long (>5 yrs)" = "sandybrown"  
 )) +  
 labs(  
 title = "Lapse Rate by Policy Term",  
 x = "Policy Term",  
 y = "Lapse Rate (%)"  
 ) +  
 ylim(0, max(lapse\_rate\_by\_term$lapse\_rate) + 8) +   
 theme\_minimal(base\_size = 14) +  
 theme(legend.position = "none")

