

Projeto

Diego,Raul,Thais

18/06/2022

```
options(knitr.duplicate.label = 'allow')
```

#Contextualização

-O conjunto de dados mostra informações sobre os hábitos de compra do cliente. O objetivo da análise é investigar a segmentação de clientes bem como predizer o volume de gasto médio do consumidor, a fim de permitir que a empresa compreenda as diferentes categorias de clientes que possui e predizer o volume médio de receita. O conjunto de dados contém as seguintes informações sobre cada cliente:

-ID: identificador exclusivo do cliente -Year_Birth: Ano de nascimento do cliente -Education: nível de educação do cliente -Marital_Status: estado civil do cliente -Income: Renda familiar anual do cliente -Kidhome: Número de crianças na casa do cliente -Teenhome: Número de adolescentes na casa do cliente -Dt_Customer: Data do cadastro do cliente na empresa -Recency: número de dias desde a última compra do cliente -Complain: 1 se o cliente reclamou nos últimos 2 anos, 0 caso contrário -MntWines: Valor gasto em vinho nos últimos 2 anos -MntFruits: Valor gasto em frutas nos últimos 2 anos -MntMeatProducts: Valor gasto em carne nos últimos 2 anos -MntFishProducts: Valor gasto em pescado nos últimos 2 anos -MntSweetProducts: Valor gasto em doces nos últimos 2 anos -MntGoldProds: Valor gasto em ouro nos últimos 2 anos -NumDealsPurchases: Número de compras feitas com desconto -AcceptedCmp1: 1 se o cliente aceitou a oferta na 1ª campanha, 0 caso contrário -AcceptedCmp2: 1 se o cliente aceitou a oferta na 2ª campanha, 0 caso contrário -AcceptedCmp3: 1 se o cliente aceitou a oferta na 3ª campanha, 0 caso contrário -AcceptedCmp4: 1 se o cliente aceitou a oferta na 4ª campanha, 0 caso contrário -AcceptedCmp5: 1 se o cliente aceitou a oferta na 5ª campanha, 0 caso contrário -Response: 1 se o cliente aceitou a oferta na última campanha, 0 caso contrário -NumWebPurchases: Número de compras feitas através do site da empresa -NumCatalogPurchases: Número de compras feitas usando um catálogo -NumStorePurchases: Número de compras feitas diretamente nas lojas -NumWebVisitsMonth: Número de visitas ao site da empresa no último mês

#Preparação

```
#Pacotes Utilizados
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1.9000 --
```

```
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.6      v dplyr    1.0.8
## v tidyverse 1.2.0     v stringr  1.4.0
## v readr   2.1.2      vforcats  0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(skimr)
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following objects are masked from 'package:base':  
##  
##     date, intersect, setdiff, union
```

```
library(plotly)
```

```
##  
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':  
##  
##     last_plot
```

```
## The following object is masked from 'package:stats':  
##  
##     filter
```

```
## The following object is masked from 'package:graphics':  
##  
##     layout
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggrepel)  
library(cluster)  
library(fpc)  
library(dbSCAN)
```

```
##  
## Attaching package: 'dbSCAN'
```

```
## The following object is masked from 'package:fpc':  
##  
##     dbSCAN
```

```
library(tidyModels)
```

```
## -- Attaching packages ----- tidyModels 0.2.0 --
```

```
## v broom      0.8.0    v rsample     0.1.1  
## v dials      0.1.0    v tune        0.2.0  
## v infer      1.0.0    v workflows   0.2.6  
## v modeldata   0.1.1    v workflowsets 0.2.1  
## v parsnip     0.2.1    v yardstick   0.0.9  
## v recipes     0.2.0
```

```
## -- Conflicts ----- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x plotly::filter()  masks dplyr::filter(), stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Learn how to get started at https://www.tidymodels.org/start/
```

```
library(vip)
```

```
##
## Attaching package: 'vip'
```

```
## The following object is masked from 'package:utils':
##
##     vi
```

```
library(ggcorrplot)
```

#Visualização dos dados

```
df<-read.csv("marketing_campaign.csv",sep = "\t") %>%
  clean_names() # standardização dos nomes das colunas
```

#Alteração dos nomes das colunas

```
#Verificação dos nomes das colunas:
```

```
colnames(df)
```

```
## [1] "id"                  "year_birth"          "education"
## [4] "marital_status"       "income"              "kidhome"
## [7] "teenhome"            "dt_customer"        "recency"
## [10] "mnt_wines"           "mnt_fruits"         "mnt_meat_products"
## [13] "mnt_fish_products"   "mnt_sweet_products" "mnt_gold_prods"
## [16] "num_deals_purchases" "num_web_purchases"  "num_catalog_purchases"
## [19] "num_store_purchases" "num_web_visits_month" "accepted_cmp3"
## [22] "accepted_cmp4"       "accepted_cmp5"       "accepted_cmp1"
## [25] "accepted_cmp2"       "complain"           "z_cost_contact"
## [28] "z_revenue"           "response"
```

#Alteração dos nomes das colunas

```
df<-df %>%
  rename(cust_retention_year=dt_customer,
         days_wo_purchase=recency,
         wine=mnt_wines,
         fruit=mnt_fruits,
         meat=mnt_meat_products,
         fish=mnt_fish_products,
         sweet=mnt_sweet_products,
         gold=mnt_gold_prods,
         deals_purchases=num_deals_purchases,
         web_purchases=num_web_purchases,
         catalog_purchases=num_catalog_purchases,
         store_purchases=num_store_purchases,
         web_visits_month=num_web_visits_month
  )
```

#Verificação dos tipos de dados

```
skim(df)
```

Data summary

Name	df
Number of rows	2240
Number of columns	29

Column type frequency:

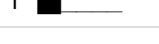
character	3
numeric	26

Group variables	None
-----------------	------

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
education	0	1	3	10	0	5	0
marital_status	0	1	4	8	0	8	0
cust_retention_year	0	1	10	10	0	663	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
id	0	1.00	5592.16	3246.66	0	2828.25	5458.5	8427.75	11191	
year_birth	0	1.00	1968.81	11.98	1893	1959.00	1970.0	1977.00	1996	
income	24	0.99	52247.25	25173.08	1730	35303.00	51381.5	68522.00	666666	
kidhome	0	1.00	0.44	0.54	0	0.00	0.0	1.00	2	
teenhome	0	1.00	0.51	0.54	0	0.00	0.0	1.00	2	
days_wo_purchase	0	1.00	49.11	28.96	0	24.00	49.0	74.00	99	
wine	0	1.00	303.94	336.60	0	23.75	173.5	504.25	1493	
fruit	0	1.00	26.30	39.77	0	1.00	8.0	33.00	199	
meat	0	1.00	166.95	225.72	0	16.00	67.0	232.00	1725	
fish	0	1.00	37.53	54.63	0	3.00	12.0	50.00	259	
sweet	0	1.00	27.06	41.28	0	1.00	8.0	33.00	263	
gold	0	1.00	44.02	52.17	0	9.00	24.0	56.00	362	
deals_purchases	0	1.00	2.33	1.93	0	1.00	2.0	3.00	15	
web_purchases	0	1.00	4.08	2.78	0	2.00	4.0	6.00	27	
catalog_purchases	0	1.00	2.66	2.92	0	0.00	2.0	4.00	28	
store_purchases	0	1.00	5.79	3.25	0	3.00	5.0	8.00	13	
web_visits_month	0	1.00	5.32	2.43	0	3.00	6.0	7.00	20	
accepted_cmp3	0	1.00	0.07	0.26	0	0.00	0.0	0.00	1	
accepted_cmp4	0	1.00	0.07	0.26	0	0.00	0.0	0.00	1	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
accepted_cmp5	0	1.00	0.07	0.26	0	0.00	0.0	0.00	1	
accepted_cmp1	0	1.00	0.06	0.25	0	0.00	0.0	0.00	1	
accepted_cmp2	0	1.00	0.01	0.11	0	0.00	0.0	0.00	1	
complain	0	1.00	0.01	0.10	0	0.00	0.0	0.00	1	
z_cost_contact	0	1.00	3.00	0.00	3	3.00	3.0	3.00	3	
z_revenue	0	1.00	11.00	0.00	11	11.00	11.0	11.00	11	
response	0	1.00	0.15	0.36	0	0.00	0.0	0.00	1	

#Ajuste dos tipos de dados

```
df$accepted_cmp1<- factor(df$accepted_cmp1)
df$accepted_cmp2<- as.factor(df$accepted_cmp2)
df$accepted_cmp3<- as.factor(df$accepted_cmp3)
df$accepted_cmp4<- as.factor(df$accepted_cmp4)
df$accepted_cmp5<- as.factor(df$accepted_cmp5)
df$complain<- as.factor(df$complain)
```

#Como só temos um output (único valor) em algumas colunas, iremos removê-las da base por não agregarem valor a nossa análise

#Output das colunas bem como a quantidade de cada opção

```
table(df['z_revenue'],useNA='always')
```

```
## 
##   11 <NA>
## 2240    0
```

```
table(df['z_cost_contact'],useNA='always')
```

```
## 
##   3 <NA>
## 2240    0
```

```
df['z_cost_contact'] <-NULL
df['z_revenue']<- NULL
```

#Inclusão de novas colunas baseadas nas variáveis que possuímos:

```

# Idade dos consumidores:
df['age']<- 2014 - df['year_birth']
#importante comentar que utilizamos 2014 nesse cálculo, pois a base de dados utilizada foi desse ano

#Número de filhos:
df['children']<- df['kidhome']+df['teenhome']

#Número de anos no qual cliente é consumidor dessa empresa:
df['cust_retention_year']<- 2014 - as.integer(str_sub(df$cust_retention_year,-4))

# Se o consumidor é casado:
df['married'] <-
  ifelse(df['marital_status']=='Alone'|df['marital_status']=='Single'|df['marital_status']=='Divorced'|df['marital_status']=='Widow',0,
  (ifelse(df['marital_status']=='Married'|df['marital_status']=='Together',1,NA)))

# Se consumidor é casado (categórico):
df['marital_status'] <-
  ifelse(df['marital_status']=='Alone'|df['marital_status']=='Single'|df['marital_status']=='Divorced'|df['marital_status']=='Widow','Single',
  (ifelse(df['marital_status']=='Married'|df['marital_status']=='Together','Married',NA)))

#Total de gastos:
df['total_expense'] <- df['wine']+df['fruit']+df['meat']+df['fish']+df['sweet']+df['gold']

#Classe Social:
df['social_class'] <- ifelse(df['income']<=32000,'Class E',
  (ifelse(df['income']>32000 & df['income']<=53000,'Class D',
  (ifelse(df['income']>53000 & df['income']<=107000,'Class C',
  (ifelse(df['income']>107000 & df['income']<=37400,'Class B','Class A'))))))))

#Intervalo de Idade:
df['age_class'] <- ifelse(df['age']<=30,'<=30',
  (ifelse(df['age']>30 & df['age']<=50,'30-50',
  (ifelse(df['age']>50 & df['age']<=70,'50-70','>70')))))

#Número de Compras:
df['total_purchases']<- df['web_purchases']+df['catalog_purchases']+df['store_purchases']

#Porcentagem de Compras Online:
df['online_purchases']<-df['web_purchases']/df['total_purchases']

#Número de anos estudando
df['years_education']<-ifelse(df['education']=='2n Cycle',8,
  (ifelse(df['education']=='Basic',12,
  (ifelse(df['education']=='Graduation',16,
  (ifelse(df['education']=='Master',18,22)))))))

# Formação:
df['education_2'] <-
  ifelse(df['education']=='2n Cycle'|df['education']=='Basic','Not graduated',
  (ifelse(df['education']=='Graduation'|df['education']=='Master'|df['education']=='PhD','Graduated',NA)))

#Volume de compras com desconto
df['purchase_discount']<-df$deals_purchase/df$total_purchases

#O valor da coluna acima, purchase_discount, necessariamente precisa estar num intervalo entre 0 e 1, portanto
#removeremos os demais casos que não se encontram nesse cenário, pois pode ter ocorrido por conta de alguma inconsistência na base

```

```
df<-df %>%
  filter(purchase_discount<=1)

#Volume de compras com desconto (categórico)

df['purchase_discount2'] <- ifelse(df['purchase_discount']<=0.25, '0-25%',
  (ifelse(df['purchase_discount']>0.25 & df['purchase_discount']<=0.5, '25%-50%',
  (ifelse(df['purchase_discount']>0.5 & df['purchase_discount']<=0.75, '50%-75%', '75%-100%'
)))))

#Gasto Médio por compra

df['avg_purchase']<-df$total_expense/df$total_purchases
```

#A adição de novas variáveis nos auxiliará tanto nas análises de segmentação do cliente bem como na predição, pois poderemos utilizar variáveis contínuas adicionais no PCA e K-means, e variáveis categóricas na modelagem

#Remoção de NAs da base

```
#Como o número de NAs é baixo, iremos removê-lo da nossa base

df<-drop_na(df)

#Por conta dessa remoção, removemos cerca de 1,5% de observações da base total
```

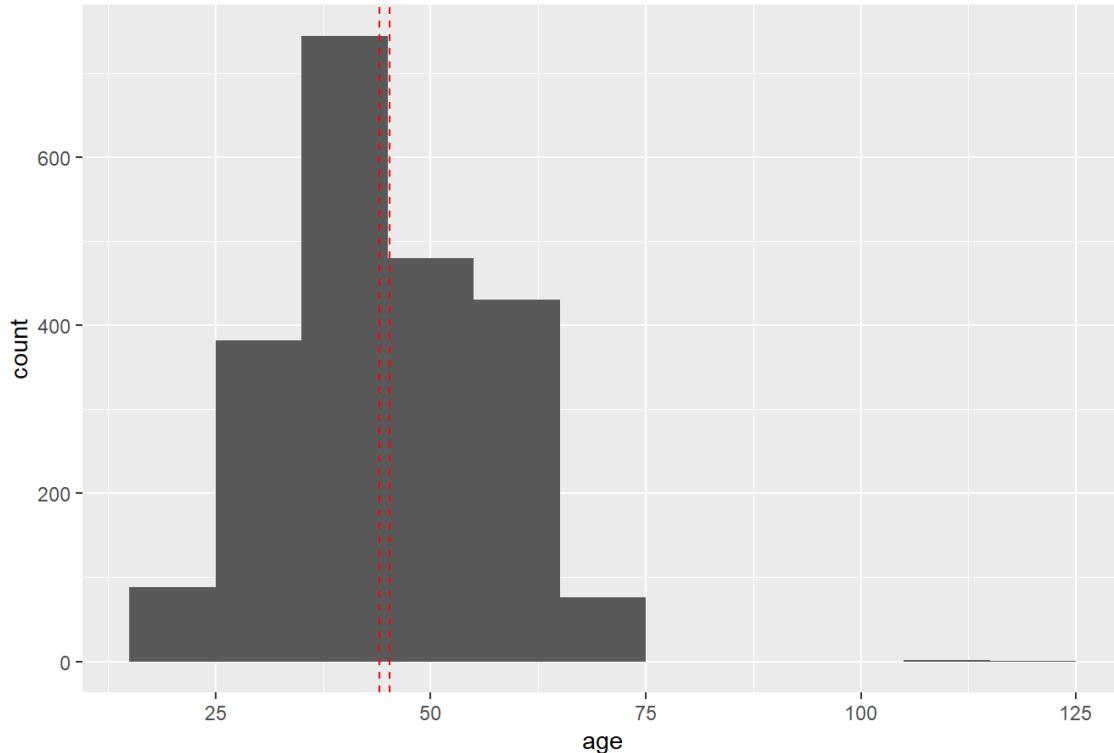
#Análise detalhada da base

#Antes de iniciarmos a análise da base em si através de modelos estatísticos, verificaremos em maiores detalhes as informações disponíveis:

Distribuição de idade

```
# Distribuição de idade
age<-ggplot(data=df) +
  geom_histogram(mapping=aes(x=age), binwidth=10) +
  geom_vline(aes(xintercept=mean(age)), linetype='dashed', color='red', size=0.5) +
  geom_vline(aes(xintercept=median(age)), linetype='dashed', color='red', size=0.5) +
  ggtitle('Histograma - idade dos consumidores')
age
```

Histograma - idade dos consumidores



#A mediana e a média estão bem próximas, parece que a média está tendendo à localização central do intervalo de idades, portanto parece que o dataset tem uma distribuição simétrica.

Porém temos alguns valores extremos acima de 100 anos que podem acabar impactando nas análises posteriores, então como se trata de somente 3 casos, as retiraremos, isso não impactará no volume de dados que teremos para análise

Filtro de idade

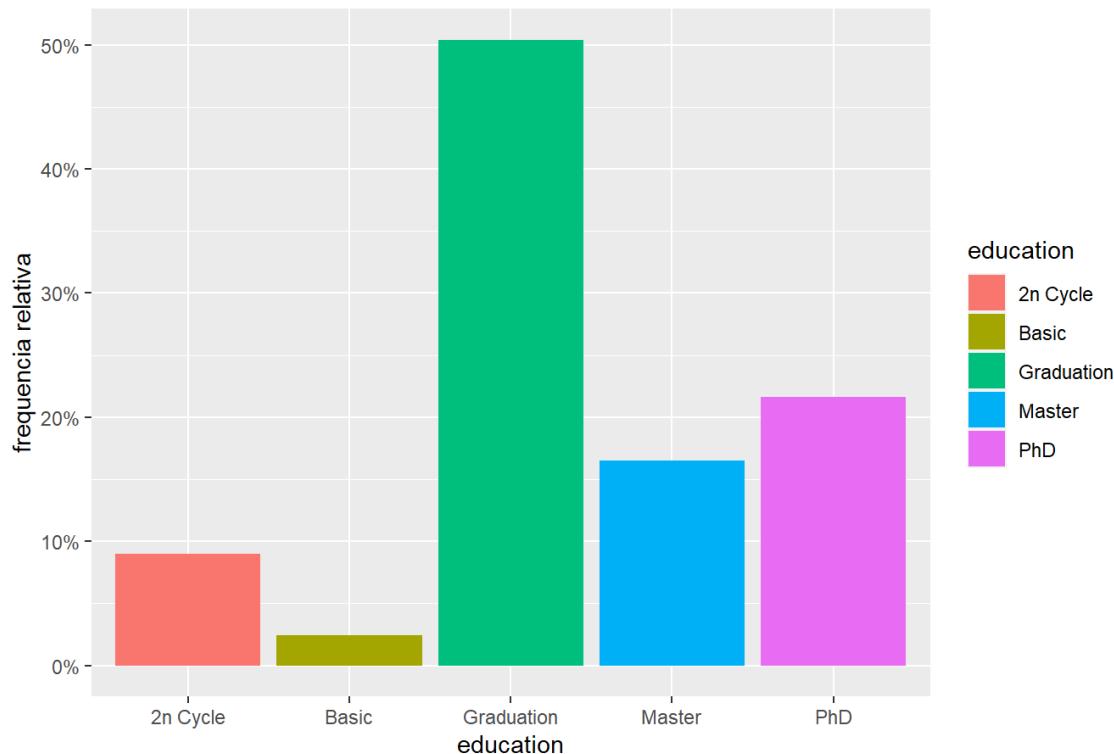
```
df<-df %>%
  filter(age<=100)
```

Distribuição de Escolaridade

```
# Distribuição de Escolaridade
educ<-ggplot(data=df, aes(education,fill=education)) +
  geom_bar(aes(y=(..count..)/sum(..count..))) +
  scale_y_continuous(labels = scales::percent) +
  ylab("frequencia relativa")+
  ggtitle('Nivel de Educacao')

educ
```

Nivel de Educacao



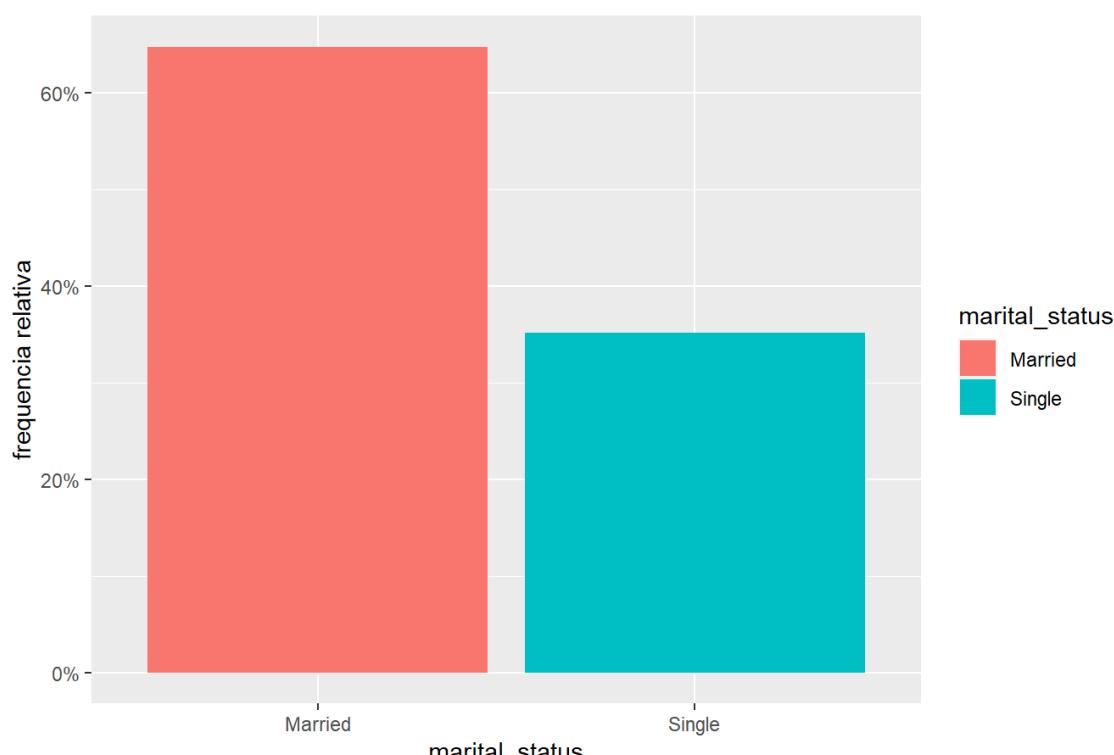
#A maioria dos consumidores parece ter graduação superior, cerca de 90%.

#Distribuição Status Civil

```
# Distribuição Status Civil
marital<-ggplot(data=df, aes(marital_status, fill=marital_status)) +
  geom_bar(aes(y(..count..)/sum(..count..))) +
  scale_y_continuous(labels = scales::percent) +
  ylab("frequencia relativa") +
  ggtitle('Status Civil')
```

marital

Status Civil



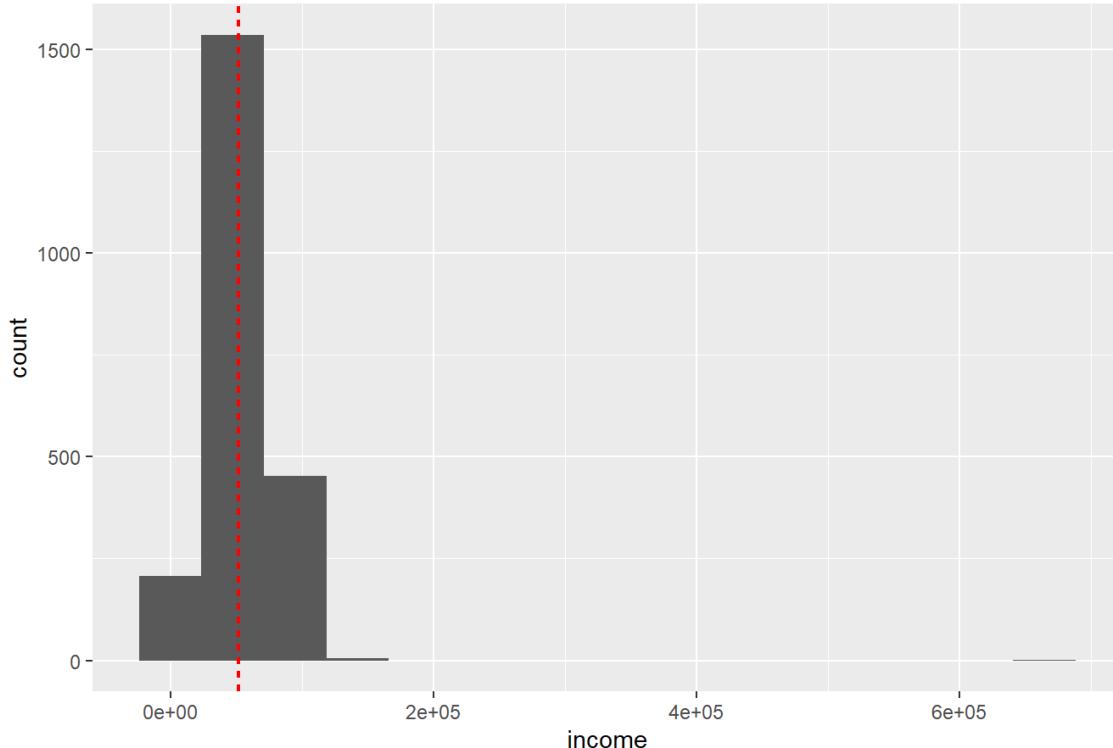
#A distribuição entre as opções de status civil é similar, tendo uma porcentagem um pouco maior de pessoas casadas presentes na base (cerca de 60%)

#Distribuição de Renda

```
#Distribuição de Renda
income<-ggplot(data=df) +
  geom_histogram(mapping=aes(x=income),bins=15) +
  geom_vline(aes(xintercept=mean(income)), linetype='dashed', color='red', size=0.5) +
  geom_vline(aes(xintercept=median(income)), linetype='dashed', color='red', size=0.5) +
  ggtitle('Histograma - Renda dos consumidores')

income
```

Histograma - Renda dos consumidores



#Pode-se observar que temos alguns valores discrepantes na renda (rendas superiores a 200k), e isso pode acabar impactando nas nossas análises posteriores no PCA e K-means, pois são sensíveis à presença de outliers, então para melhorarmos nossa análise, retiraremos esses casos utilizando o Método de intervalo interquartil.

```
df<-df %>%
  filter(income<=500000)
```

#Método de intervalo interquartil

```
df %>%
  count(income>200000)
```

```
##   income > 2e+05      n
## 1          FALSE 2201
```

```
Q1 <- quantile(df$income, .25)
Q3 <- quantile(df$income, .75)
IQR <- IQR(df$income)

no_outliers <- subset(df, df$income > (Q1 - 1.5*IQR) & df$income < (Q3 + 1.5*IQR))
dim(no_outliers)
```

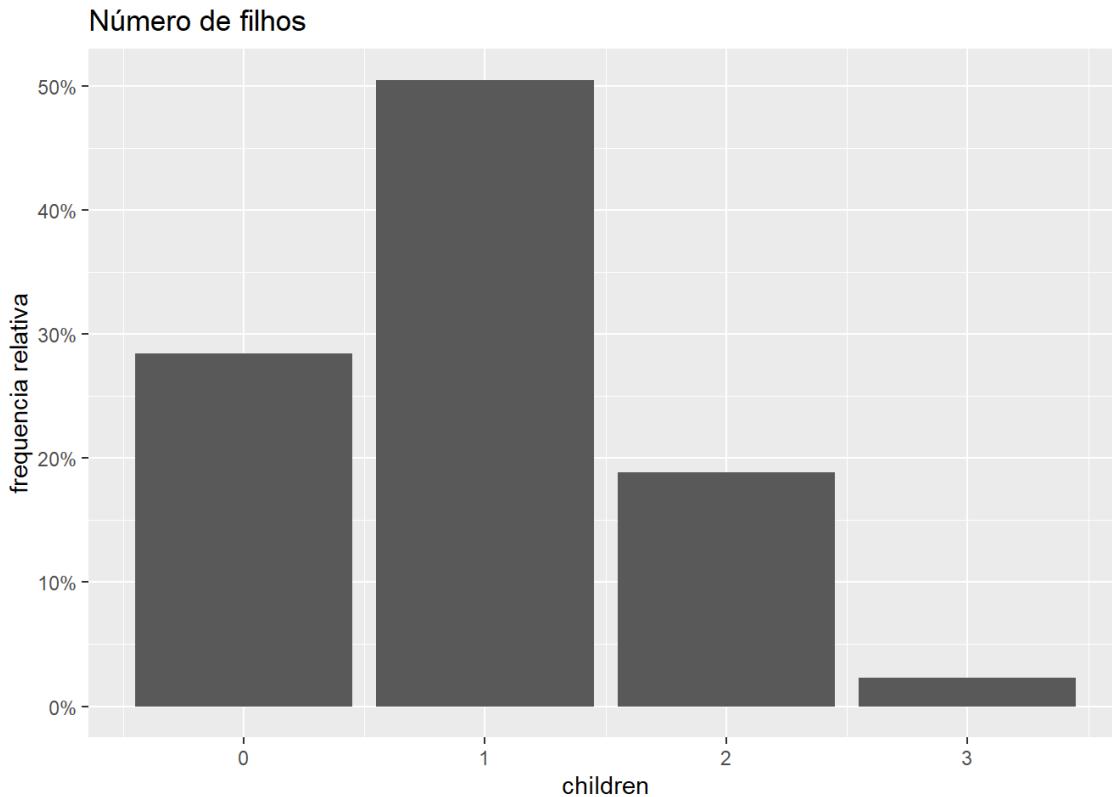
```
## [1] 2196    40
```

#Como temos 40 observações consideradas outliers nesse caso, iremos removê-las por se tratar de um volume baixo

#Distribuição do número de filhos

```
#Distribuição do número de filhos
children<-ggplot(data=df, aes(children,fill=children)) +
  geom_bar(aes(y=..count../sum(..count..))) +
  scale_y_continuous(labels = scales::percent) +
  ylab("frequencia relativa") +
  ggtitle('Número de filhos')
```

children



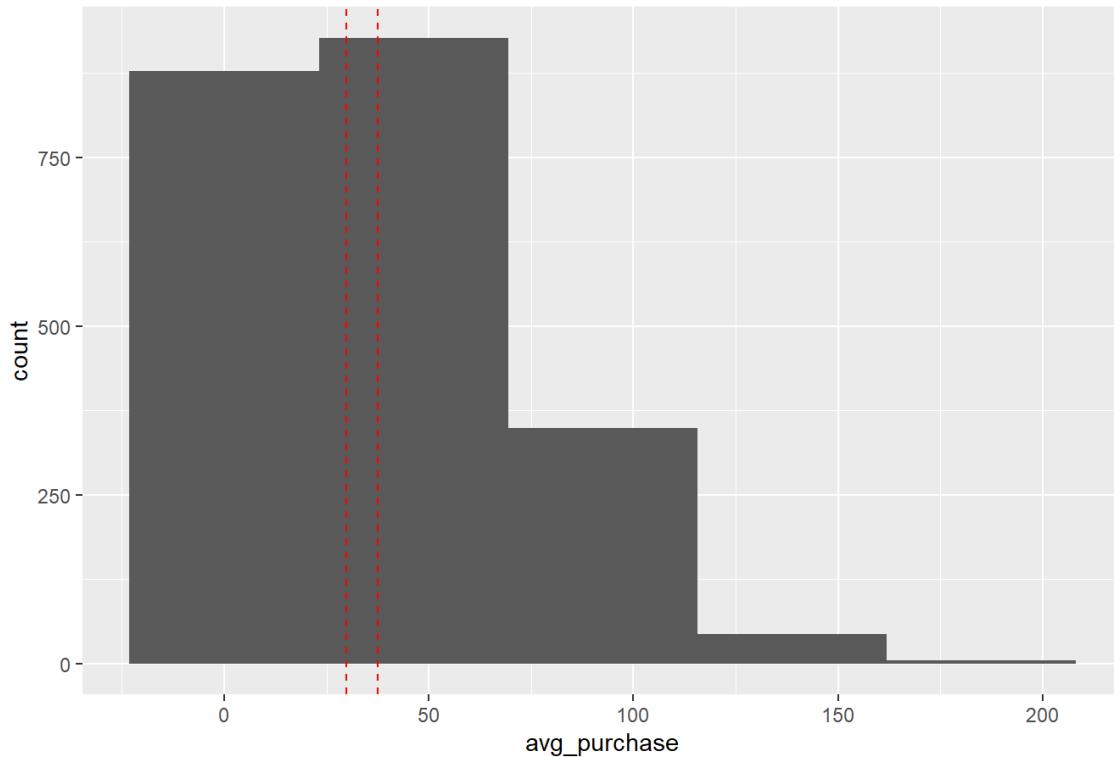
#A maioria dos consumidores possui filhos, cerca de 70% possui ao menos um filho.

#Distribuição do volume de compras médio

```
# Distribuição do volume de compras médio
avg_purchase<-ggplot(data=df) +
  geom_histogram(mapping=aes(x=avg_purchase),bins=5) +
  geom_vline(aes(xintercept=mean(avg_purchase)), linetype='dashed', color='red', size=0.5) +
  geom_vline(aes(xintercept=median(avg_purchase)), linetype='dashed', color='red', size=0.5) +
  ggtitle('Histograma - Volume de compras médio')

avg_purchase
```

Histograma - Volume de compras médio



#Gasto médio de compras é baixo, cerca de 40 dólares, pode-se supor que isso pode estar ocorrendo por dois motivos:

#Perfil de cliente: cliente pode estar realizando compras em valores mais baixos, só que mais recorrentes

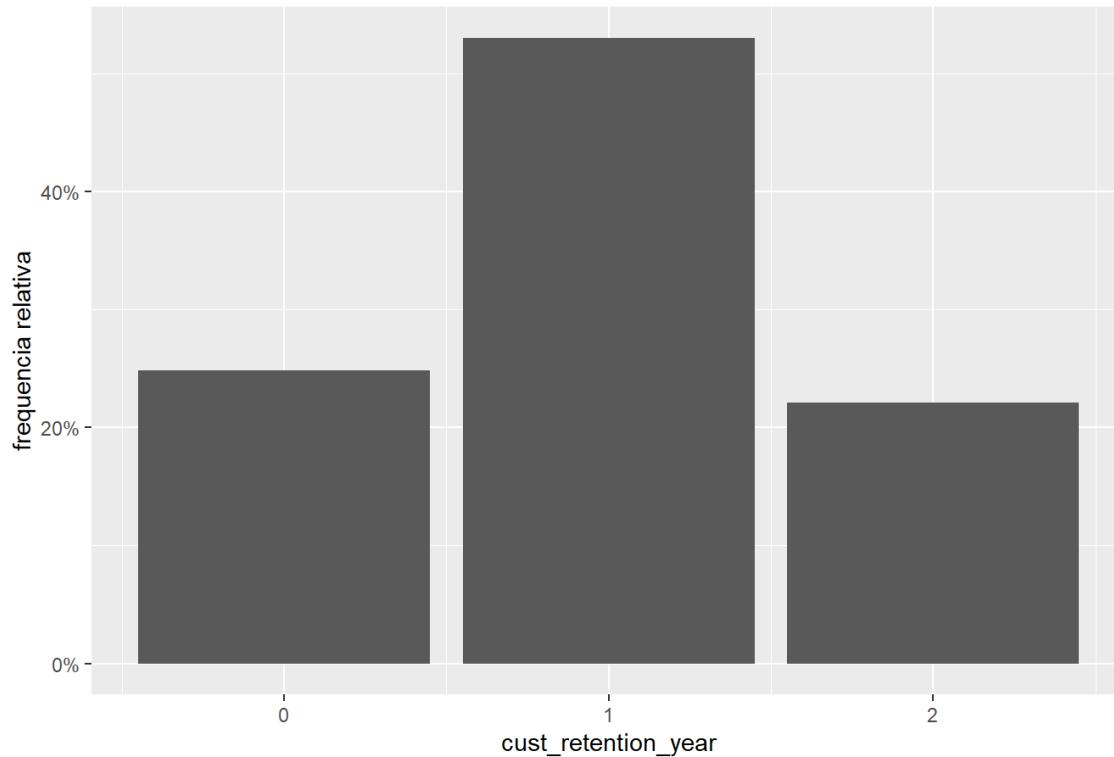
#Perfil do mercado: mercado pode estar sendo utilizado para compras menores

#Distribuição de retenção em anos de clientes

```
#Distribuição de retenção em anos de clientes
retention<-ggplot(data=df, aes(cust_retention_year, fill=cust_retention_year)) +
  geom_bar(aes(y=(..count..)/sum(..count..))) +
  scale_y_continuous(labels = scales::percent)+
  ylab("frequencia relativa")+
  ggtitle('Retenção de cliente (em anos)')

retention
```

Retenção de cliente (em anos)



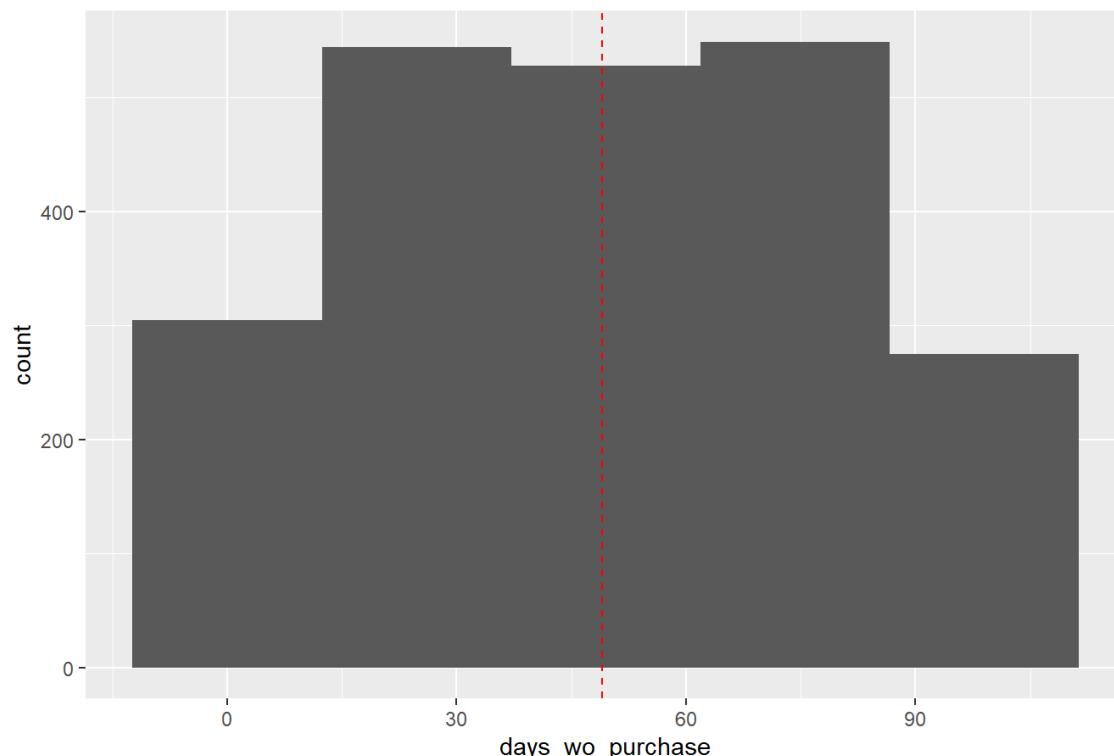
```
#A maioria dos consumidores são clientes há pelo menos um ano.
```

#Distribuição de dias sem compra

```
#Distribuição de dias sem compra
days_wo<-ggplot(data=df) +
  geom_histogram(mapping=aes(x=days_wo_purchase),bins=5) +
  geom_vline(aes(xintercept=mean(days_wo_purchase)), linetype='dashed', color='red', size=0.5) +
  geom_vline(aes(xintercept=median(days_wo_purchase)), linetype='dashed', color='red', size=0.5) +
  ggtitle('Histograma - Dias sem cliente realizar compra')
```

```
days_wo
```

Histograma - Dias sem cliente realizar compra



#Parece ser bem variado, temos um volume similar de clientes sem realizar compras até 45 dias e acima desse intervalo também.

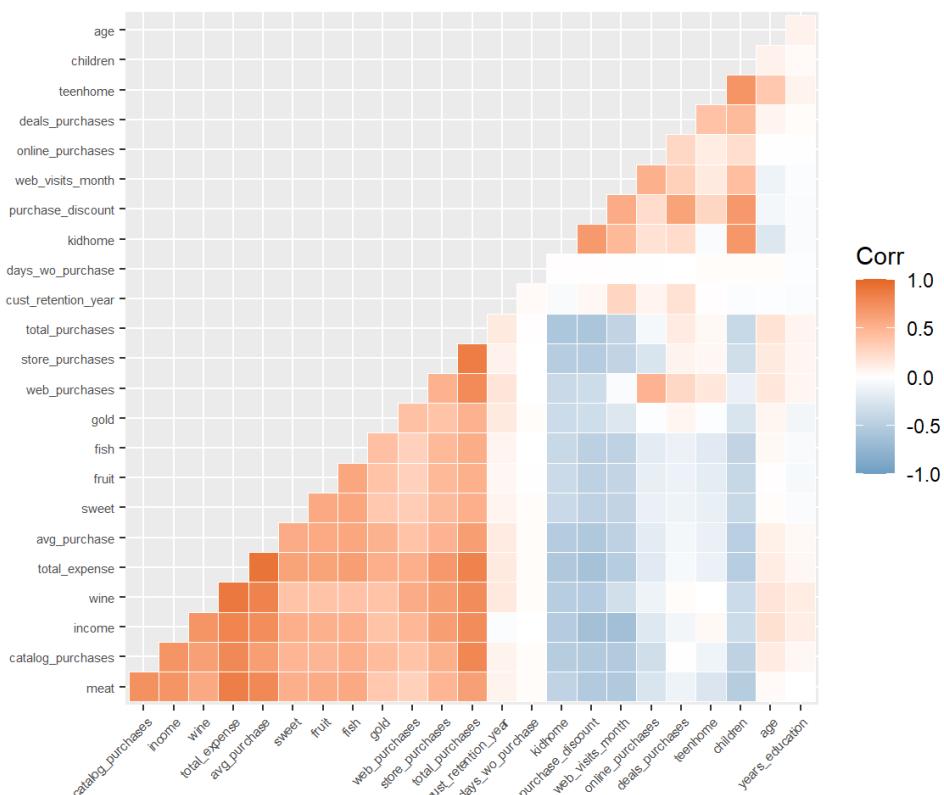
#A partir das informações já conseguimos tirar alguns conclusões superficiais, parece que o perfil do cliente em geral são consumidores de 40 a 50 anos que possuem graduação superior, tem ao menos um filho e um renda média de cerca de 50kUSD/ano.

#Antes de aplicarmos o PCA na base de dados, iremos verificar se as variáveis da nossa base estão correlacionadas, caso não estejam, talvez não seja necessária a utilização do PCA:

```
# Creating a correlation plot
df2<-df %>%
  select(income,kidhome,teenhome,cust_retention_year,days_wo_purchase,wine,fruit,meat,fish,sweet,gold,deals_purchases,web_purchases,catalog_purchases,store_purchases,web_visits_month,age,children,total_expense,total_purchases,years_education,online_purchases,avg_purchase,purchase_discount)

cormat <- round(cor(df2), 2)

ggcorrplot(cormat, hc.order = TRUE, type ='lower',outline.color ='white',tl.cex = 5.5,ggtheme = ggplot2::theme_gray,colors = c("#6D9EC1", "white", "#E46726"))
```



#Pode-se observar que grandes partes dos nossos dados são correlacionados, negativamente (-1) ou positivamente (+1), portanto podemos seguir com o PCA.

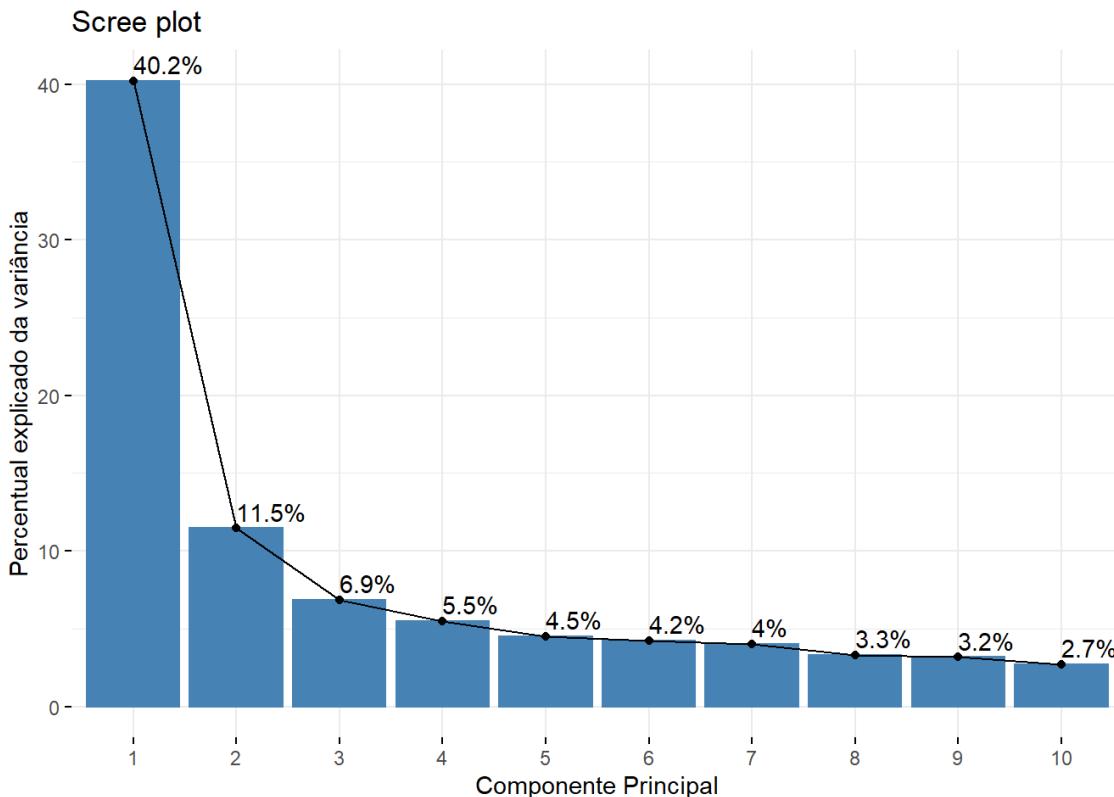
#PCA

-Seleção de variáveis para análise e aplicação do PCA

```
pca<- df %>%
  select(income,kidhome,teenhome,cust_retention_year,days_wo_purchase,wine,fruit,meat,fish,sweet,gold,deals_purchases,web_purchases,catalog_purchases,store_purchases,web_visits_month,age,children,total_expense,total_purchases,years_education,online_purchases,avg_purchase,purchase_discount) %>%
  prcomp(scale = TRUE)
```

-Percentual da variância explicado por cada componente:

```
fviz_eig(pca, addlabels = TRUE,
         ncp = 10) + # ncp - número de componentes mostrados
labs(x='Componente Principal',
     y='Percentual explicado da variância')
```



#Considerando-se o PC1 e PC2, cerca de 50% da variação é explicado por elas.

-Soma acumulado do percentual explicado da variância

```
(cumsum(pca$sdev^2)/sum(pca$sdev^2))[1:10]
```

```
## [1] 0.4022366 0.5167421 0.5854378 0.6403425 0.6852442 0.7275042 0.7677489
## [8] 0.8009588 0.8325951 0.8598555
```

#Matriz de cargas e scores:

```
phi<- -pca$rotation
z<- -pca$x
```

#Alteração do nome das colunas

```
colnames(z) <- paste0('driver:', 1:ncol(z))
```

#Interpretação dos drivers:

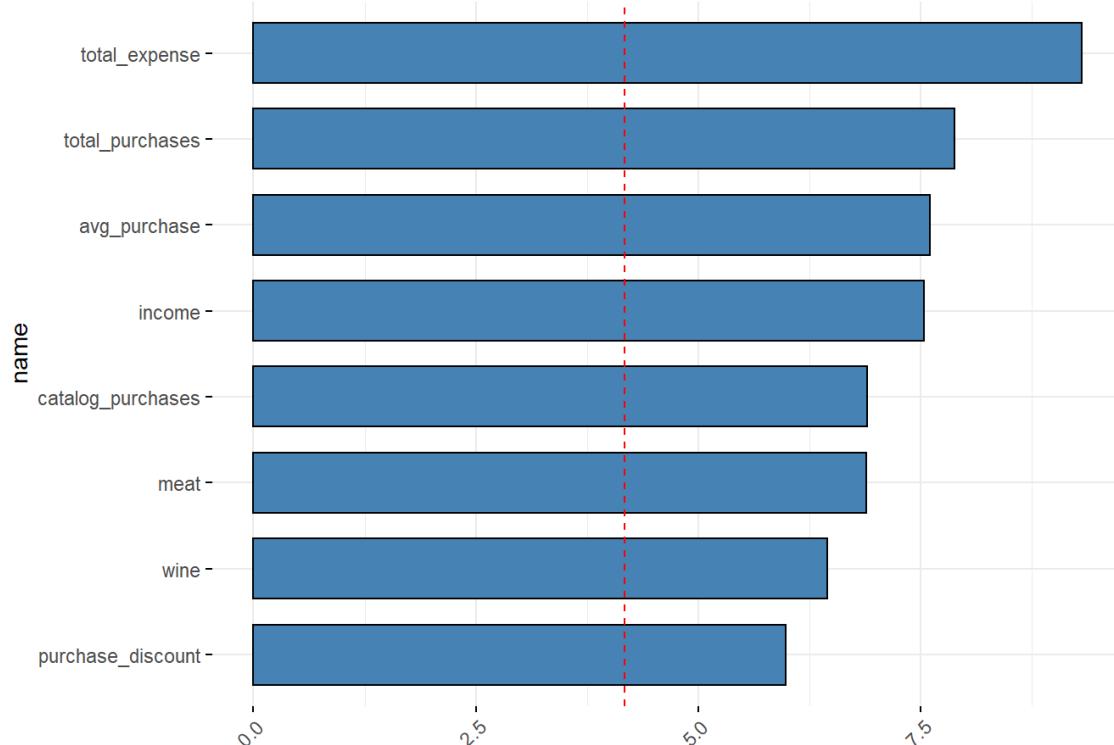
-PC1

#Contribuição das variáveis selecionadas no valor gasto pelos consumidores:

```
pc1 <-pca %>%
  fviz_contrib(choice = "var", axes = 1,top=8,sort.val = "asc",fill = "steelblue", color="black")+
  labs(title="Impacto das variáveis selecionadas no primeiro driver")+
  coord_flip() #inversão das coordenadas x e y

pc1
```

Impacto das variáveis selecionadas no primeiro driver



#Observando-se as variáveis de maior impacto no PC1, pode-se dizer que o maior driver seria o volume de gastos em compras

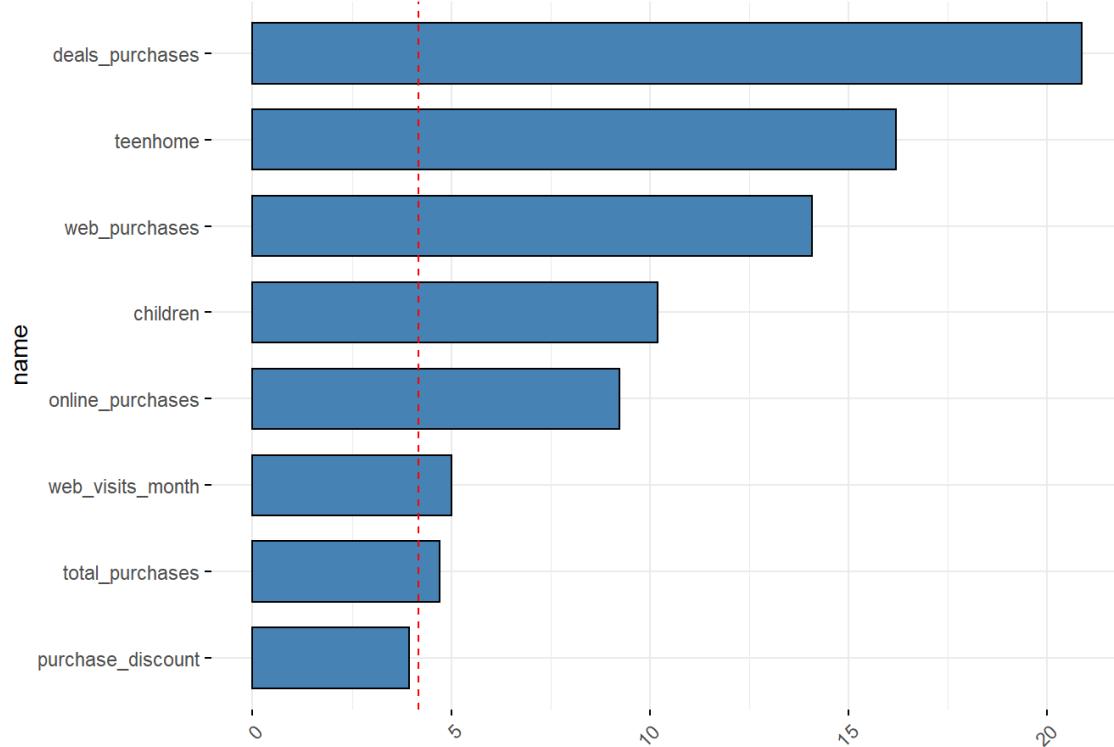
-PC2

#Contribuição das variáveis selecionadas no valor gasto pelos consumidores:

```
pc2<-pca %>%
  fviz_contrib(choice = "var",axes = 2,top=8,sort.val = "asc",fill = "steelblue", color="black")+
  labs(title="Impacto das variáveis selecionadas no segundo driver")+
  coord_flip() #inversão das coordenadas x e y
```

pc2

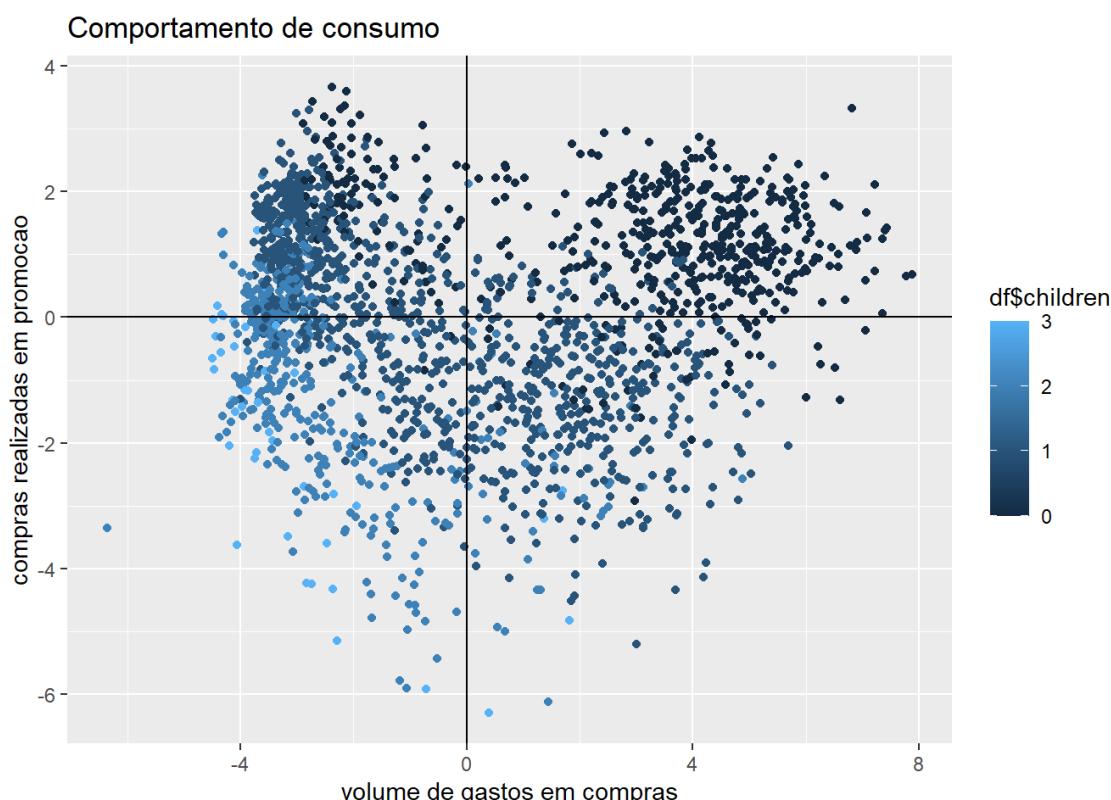
Impacto das variáveis selecionadas no segundo driver



#No PC2, o maior driver seria compras realizadas em promocao

-Gráfico de dispersão em função de filhos

```
#Gráfico de dispersão em função de filhos
tibble(PC1=z[,1], PC2=z[,2]) %>%
  ggplot(aes(PC1,PC2,color=df$children))+geom_point()+
  labs(x = "volume de gastos em compras", y = "compras realizadas em promocao",
       title = "Comportamento de consumo")+
  geom_hline(yintercept =0)+
  geom_vline(xintercept =0)
```

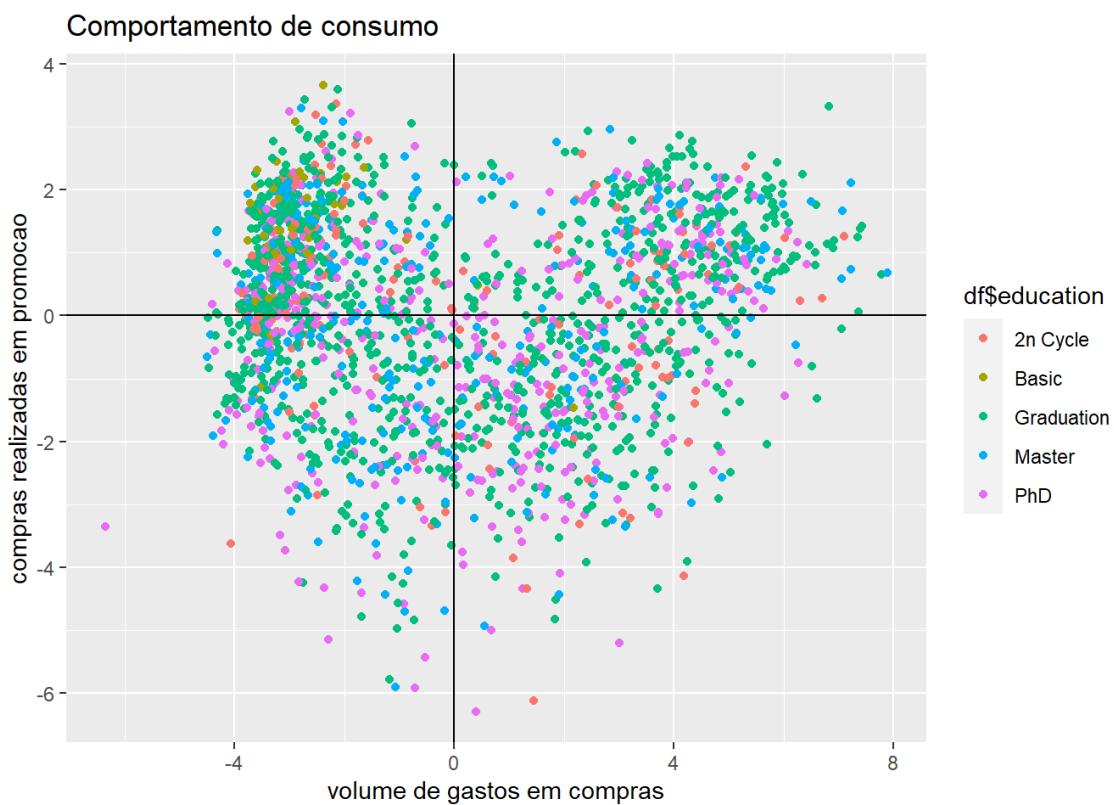


#Parece ter correlação

#Os main drivers definidos através do PCA parecem ter correlação com a questão do consumidor ter filhos. Diferente da percepção inicial do time, de acordo com o gráfico, quanto maior a despesa com compras bem como o número de compras realizadas em promoção, menor é o número de filhos do consumidor. Pode ser que isso esteja ocorrendo por conta do tipo de produto comprado, caso os consumidores comprem em maior quantidade bens de consumo básico, independente do valor, a compra é realizada. Porém, consumidores sem filhos, podem estar comprando produtos premium/dispensáveis em promoção. #Isso pode servir como um noteador para verificar qual deveria ser o cliente alvo nos envios de informações referentes à desconto e o tipo de produto a ser informado.

-Gráfico de dispersão em função do grau de escolaridade

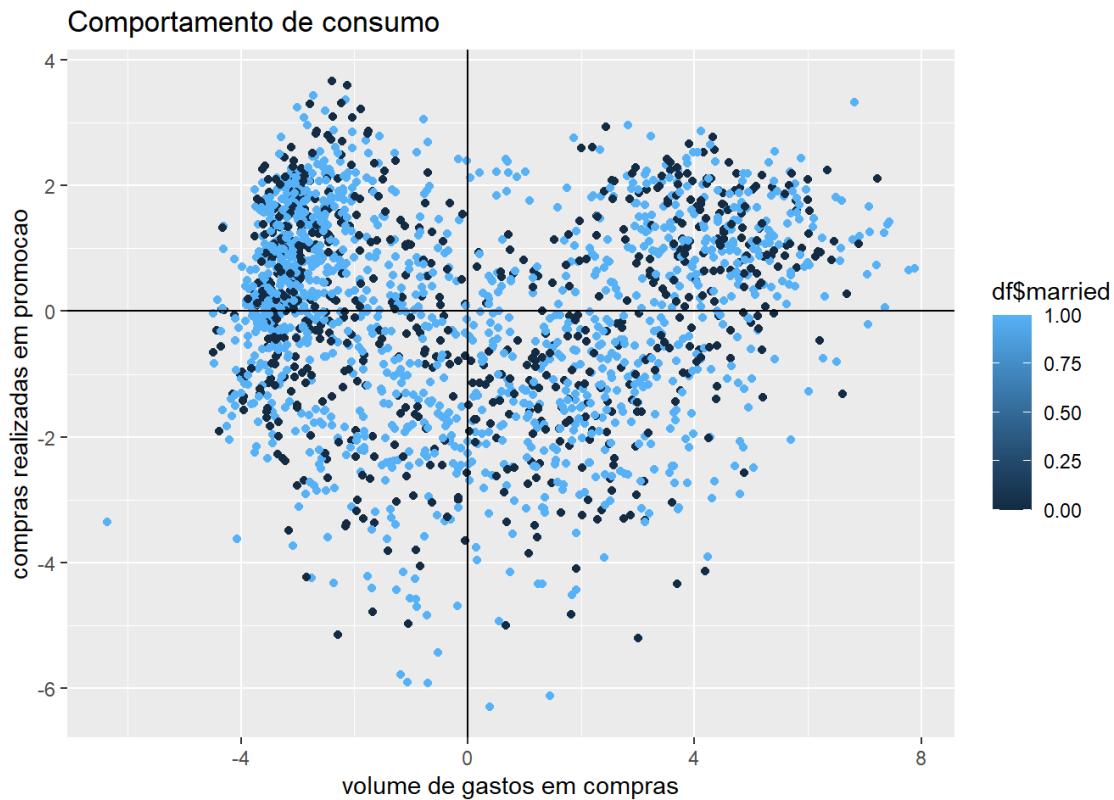
```
#gráfico de dispersão em função do grau de escolaridade
tibble(PC1=z[,1], PC2=z[,2]) %>%
  ggplot(aes(PC1,PC2,color=df$education))+geom_point()+
  labs(x = "volume de gastos em compras", y = "compras realizadas em promocao",
       title = "Comportamento de consumo")+
  geom_hline(yintercept =0)+
  geom_vline(xintercept =0)
```



#Não parece ter correlação

#gráfico de dispersão em função de status civil

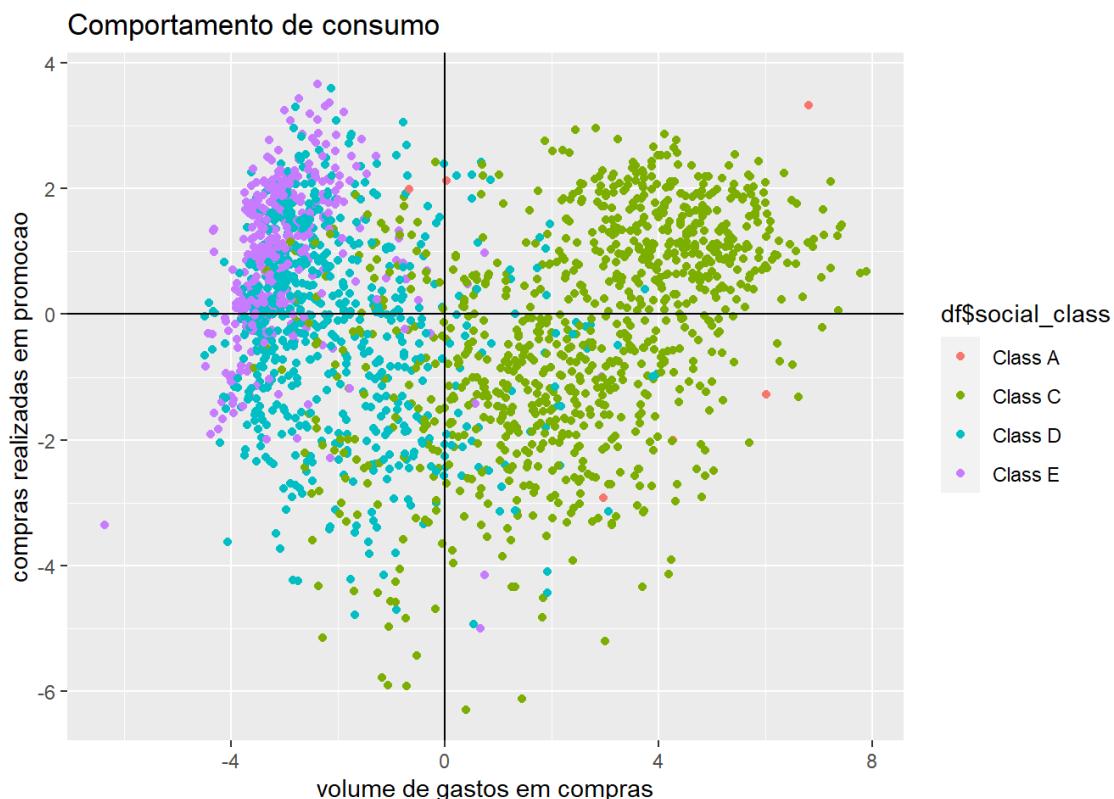
```
#gráfico de dispersão em função de status civil
tibble(PC1=z[,1], PC2=z[,2]) %>%
  ggplot(aes(PC1,PC2,color=df$married))+geom_point()+
  labs(x = "volume de gastos em compras", y = "compras realizadas em promocao",
       title = "Comportamento de consumo")+
  geom_hline(yintercept =0)+
  geom_vline(xintercept =0)
```



#Não parece ter correlação

#Gráfico de dispersão em função de classe social

```
#gráfico de dispersão em função de classe social
tibble(PC1=z[,1], PC2=z[,2]) %>%
  ggplot(aes(PC1, PC2, color=df$social_class))+geom_point()+
  labs(x = "volume de gastos em compras", y = "compras realizadas em promocao",
       title = "Comportamento de consumo")+
  geom_hline(yintercept =0)+
  geom_vline(xintercept =0)
```



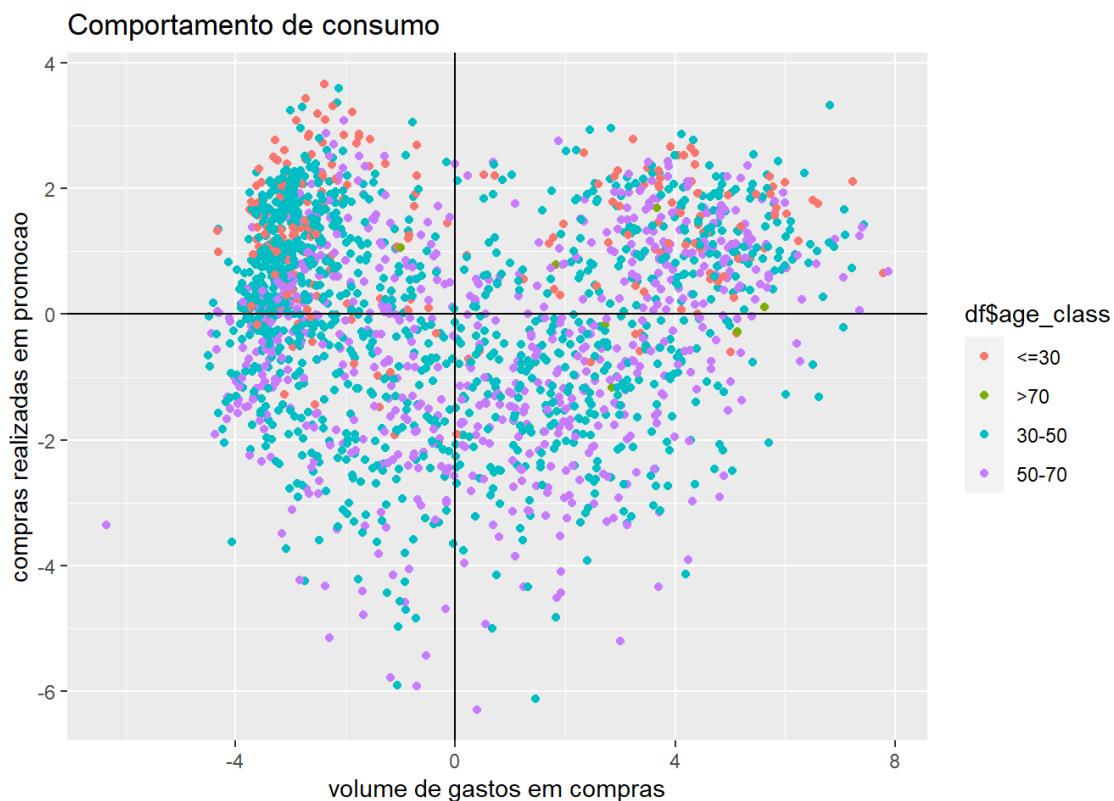
#Parece ter correlação

-Os main drivers definidos através do PCA parecem ter correlação com a classe econômica do cliente. De acordo com o gráfico, quanto maior a despesa com compras bem como o número de compras realizadas em promoção, maior é o poder aquisitivo do cliente. -A classe E, tem um gasto menor de compras que os outros grupos, e realiza em maior frequência compras com desconto. A classe D também possui um gasto menor em compras, porém não é tão suscetível à promoções como o grupo E, isso pode ser um sinal para verificar se as promoções realizadas à esse grupo estão sendo efetivas ou se um ajuste no discurso pode ser necessário. -E como esperado, a classe C possui o maior valor gastos em compra. -Por último, pode-se perceber que os consumidores desse mercado se encontram entre a classe C e E, isso já pode servir como um indicativo de quais são os meios de comunicação mais efetivos para esses grupos.

#Gráfico de dispersão em função da idade

#gráfico de dispersão em função da idade

```
tibble(PC1=z[,1], PC2=z[,2]) %>%
  ggplot(aes(PC1,PC2,color=df$age_class))+geom_point()+
  labs(x = "volume de gastos em compras", y = "compras realizadas em promocao",
       title = "Comportamento de consumo")+
  geom_hline(yintercept =0)+
  geom_vline(xintercept =0)
```



#Parece ter certa relação

#Parece que consumidores com idade igual ou inferior a 30 anos, realizam mais compras com desconto, independente dos gastos em compr. Já o comportamento dos clientes acima de 30 anos é heterogêneo, então é difícil tirar conclusões somente a partir dessa informação.

#K means

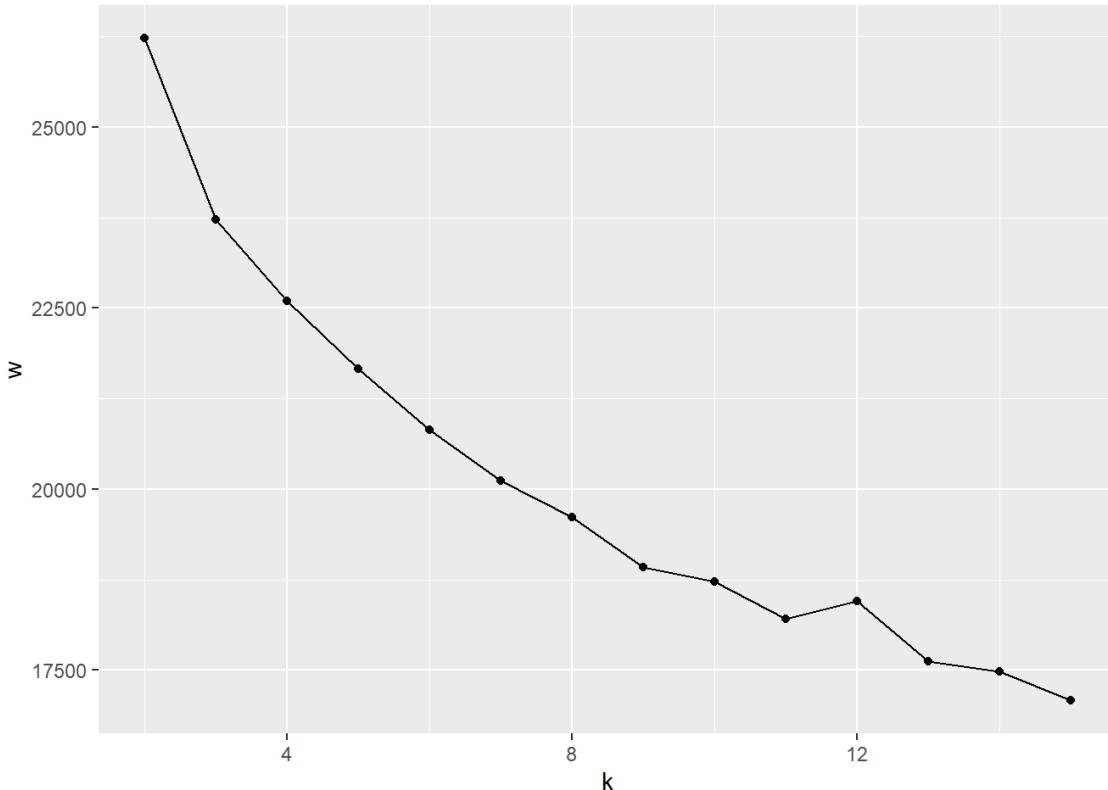
-Seleção das variáveis a serem utilizadas na análise

```
df_k<-df %>%
  select(income,cust_retention_year,days_wo_purchase,wine,fruit,meat,fish,sweet,gold,web_visits_month,age,children,total_expense,total_purchases,avg_purchase,years_education,online_purchases,purchase_discount) %>%
  scale()
```

#Análise do número de cluster

```
set.seed(42)

tibble(k=2:15) %>%
  mutate(w = map_dbl(k, ~kmeans(df_k, centers = .x)$tot.withinss)) %>%
  ggplot(aes(x=k,y=w)) +
  geom_point()+
  geom_line()
```



K=3, observa-se que após esse número, o valor de W começa a diminuir consistentemente.

#Realização da análise quantitativa

```
descricao<-df %>%
  mutate(cluster=factor(kmeans(df_k,centers = 3)$cluster))

tab<- descricao %>%
  select(-id,-year_birth,-marital_status,-kidhome,
         -teenhome,-cust_retention_year,-days_wo_purchase,-web_purchases,-catalog_purchases,-store_purchases,-re
esponse) %>%
  group_by(cluster) %>%
  summarise(across(where(is.numeric),mean))
```

#Baseado na tabela tab que traz a média dos valores de cada variável por cluster, as variáveis nos quais conseguimos verificar diferença seria: #Income,wine,fruit,meat,fish,sweet,gold,deals of purchase,web_visits_month,children,purchase_discount,avg_purchase #Já as seguintes variáveis são similares entre os cluster definidos: #Idade, status civil,online purchases,years_education

#Realização da análise quantitativa - construção de hipóteses

```
tab<- tab %>%
  mutate( p_wine=wine*100/total_expense,
         p_fruit=fruit*100/total_expense,
         p_fish=fish*100/total_expense,
         p_sweet=sweet*100/total_expense,
         p_gold=gold*100/total_expense
    )

tab2 <- tab %>% select(p_wine,p_fruit,p_fish,p_sweet,p_gold)
```

#Realização da análise quantitativa

```
tab<-tab %>%
  select(-wine,-fruit,-fish,meat,-sweet,-gold,-age,-married,-online_purchases,-years_education,-p_wine,-p_fruit,-p_fish,-p_sweet,-p_gold)

tab
```

```
## # A tibble: 3 x 10
##   cluster income meat deals_purchases web_visits_month children total_expense
##   <fct>   <dbl> <dbl>           <dbl>        <dbl>      <dbl>
## 1 1       35503. 26.4            2.32       6.46     1.29      115.
## 2 2       76677. 475.           1.41       2.83     0.235     1428.
## 3 3       59834. 145.           3.14       5.43     0.976     772.
## # ... with 3 more variables: total_purchases <dbl>, purchase_discount <dbl>,
## #   avg_purchase <dbl>
```

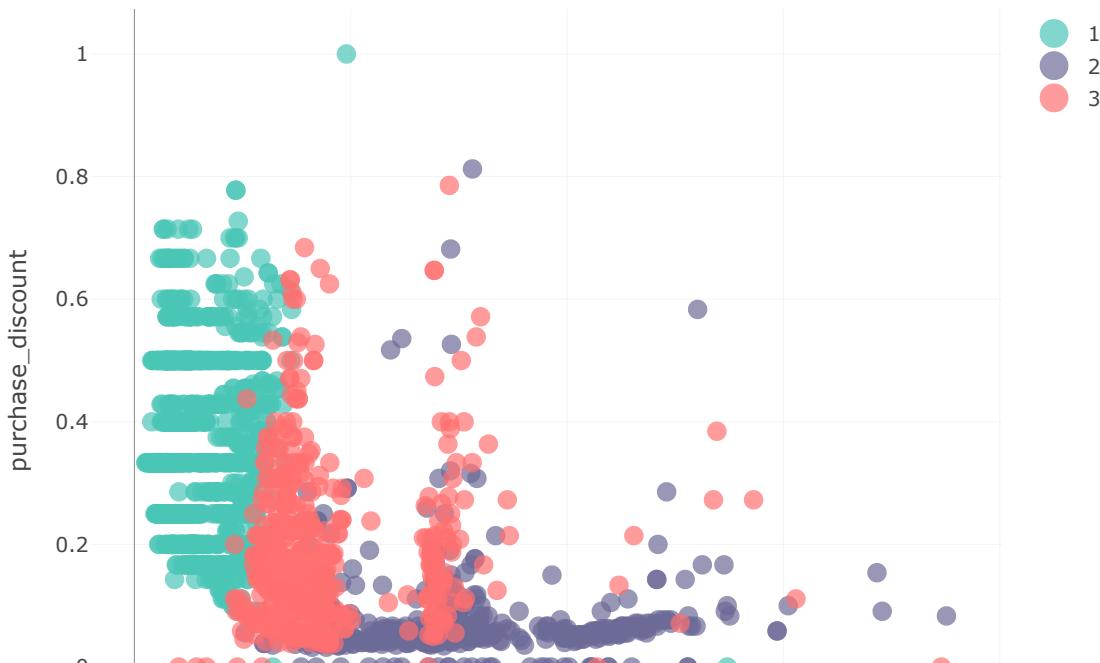
#Análise gráfica 3d - por cluster - discount

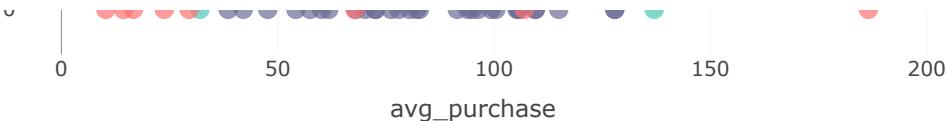
```
colors <- c('#4AC6B7', '#1972A4', '#965F8A', '#FF7070')

plot_ly(descricao,x=~avg_purchase,y=~purchase_discount,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:
## Based on info supplied, a 'scatter' trace seems appropriate.
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:
## Setting the mode to markers
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```





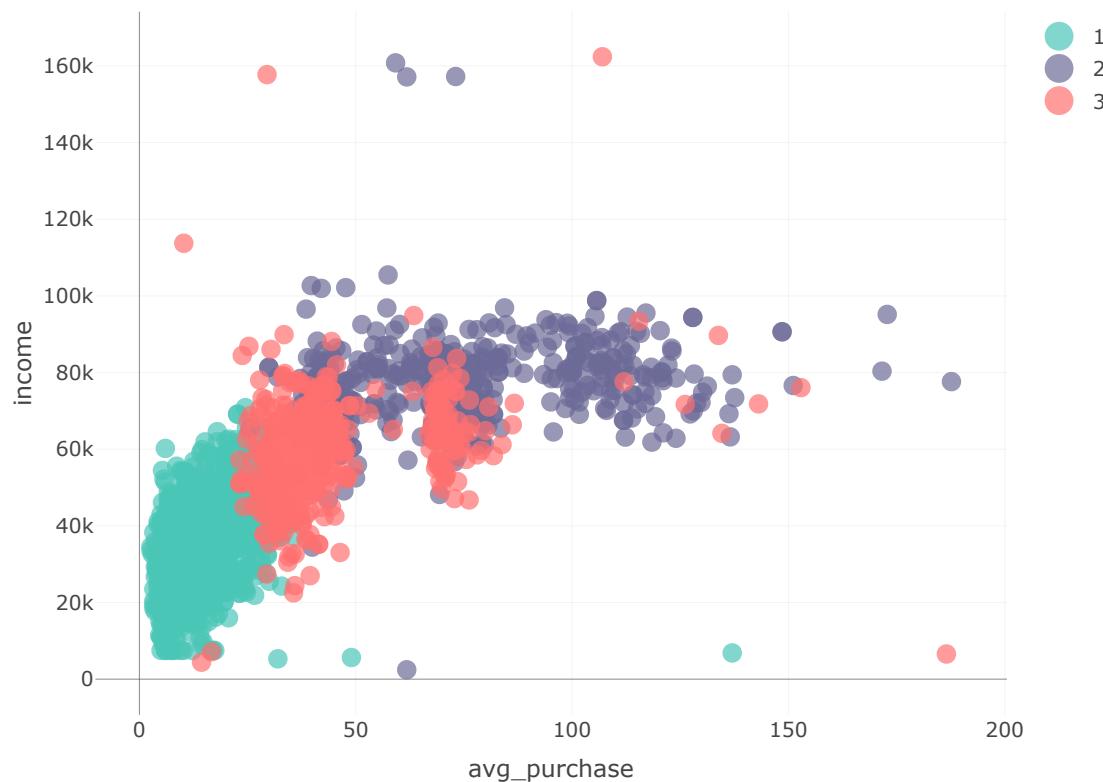
-Observa-se que a forma como o cluster foi dividido está diretamente ligado à renda.O cluster 1 é formado por consumidores de menores renda, de até cerca de 25kUSD, o cluster 2 é formado por consumidores de renda de 25k a 100k, e o último cluster,cluster 3, é composto de consumidores de um intervalo superior ao do cluster 2, entre 50k a 150k. Pode-se observar que a utilização de descontos no grupo tende a zero, diferente do que ocorre nos outros clusters.

#Análise gráfica 2d - por cluster - income

```
plot_ly(descricao,x=~avg_purchase,y=~income,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:  
## Based on info supplied, a 'scatter' trace seems appropriate.  
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:  
## Setting the mode to markers  
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```



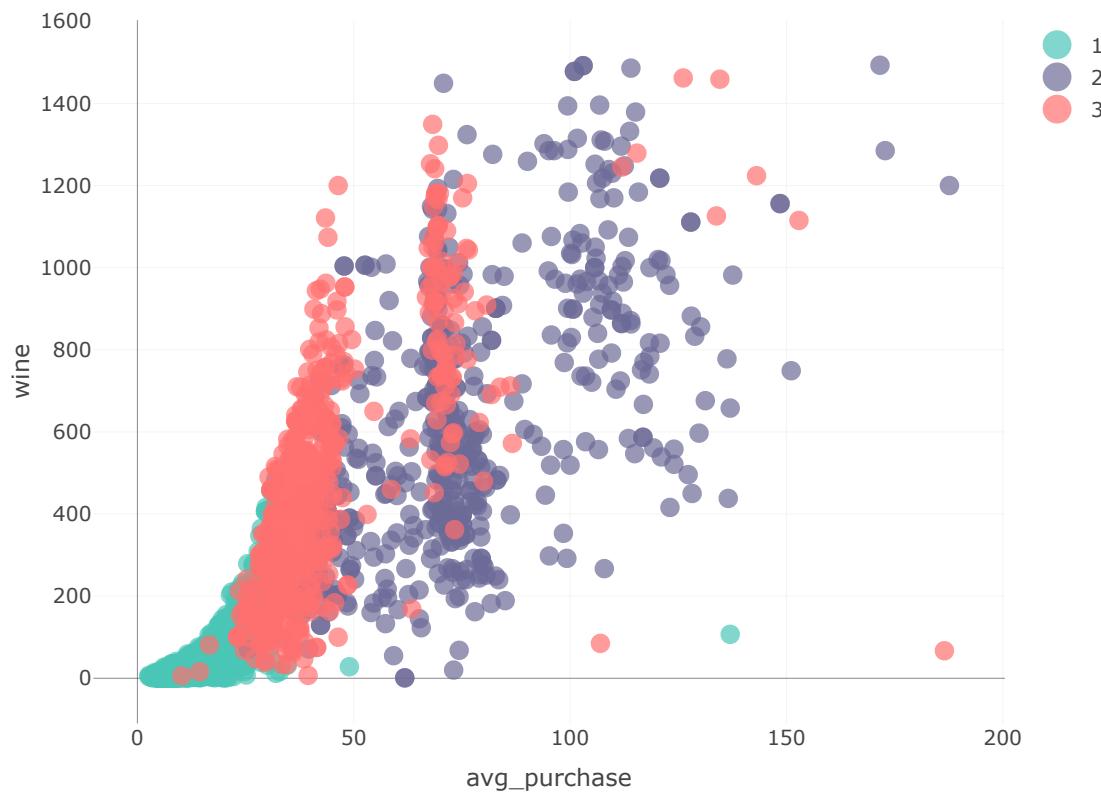
#Conforme comentado no PCA, parece que quanto maior a despesa com compras, maior é a renda do consumidor. Então o cluster 1 parece ser de clientes com menor poder aquisitivo e o grupo 2 com maior poder aquisitivo. Sendo o grupo 3 um intermédio entre eles.

#Análise gráfica 2d - por cluster - wine

```
plot_ly(descricao,x=~avg_purchase,y=~wine,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:  
## Based on info supplied, a 'scatter' trace seems appropriate.  
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:
## Setting the mode to markers
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```



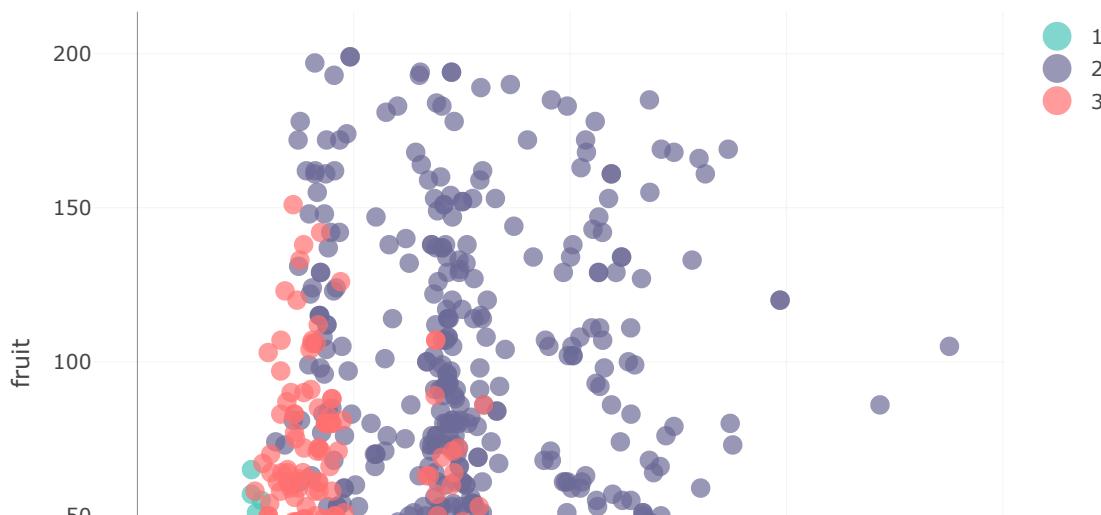
#Seguindo a lógica do gráfico por renda, temos que o grupo 2 seria os clientes com maior gasto em vinhos, isso já poderia auxiliar a empresa em definir o tipo de produto a ser ofertado através dos meios de comunicação para cada grupo, um produto mais premium para o grupo 2, intermediário para o 3 e mais barato ao grupo 1.

#Análise gráfica 3d - por cluster - fruit

```
plot_ly(descricao,x=~avg_purchase,y=~fruit,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:
## Based on info supplied, a 'scatter' trace seems appropriate.
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:
## Setting the mode to markers
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```





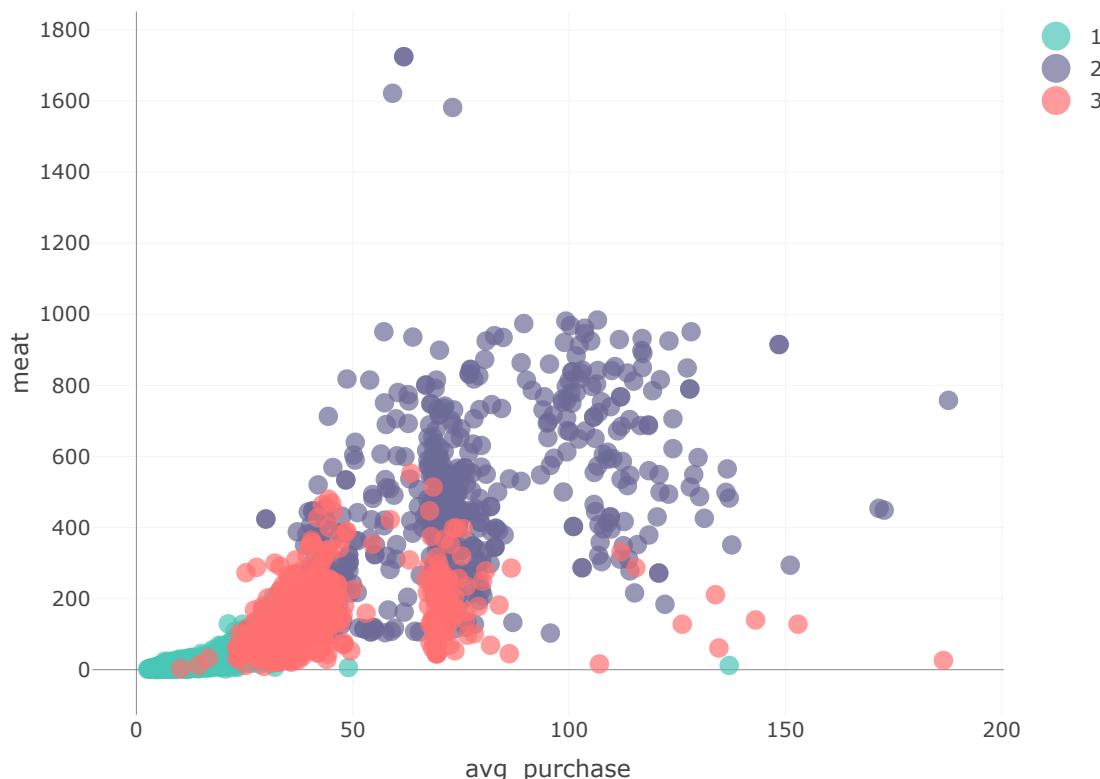
#Seguindo a lógica do gráfico de vinho, temos que o grupo 2 seria os clientes com maior gasto em frutas, isso já poderia auxiliar a empresa em definir o tipo de produto a ser ofertado através dos meios de comunicação para cada grupo, um produto mais premium para o grupo 2, intermediário para o 3 e mais barato ao grupo 1.

#Análise gráfica 3d - por cluster - carne

```
plot_ly(descricao,x=~avg_purchase,y=~meat,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:  
## Based on info supplied, a 'scatter' trace seems appropriate.  
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:  
## Setting the mode to markers  
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```



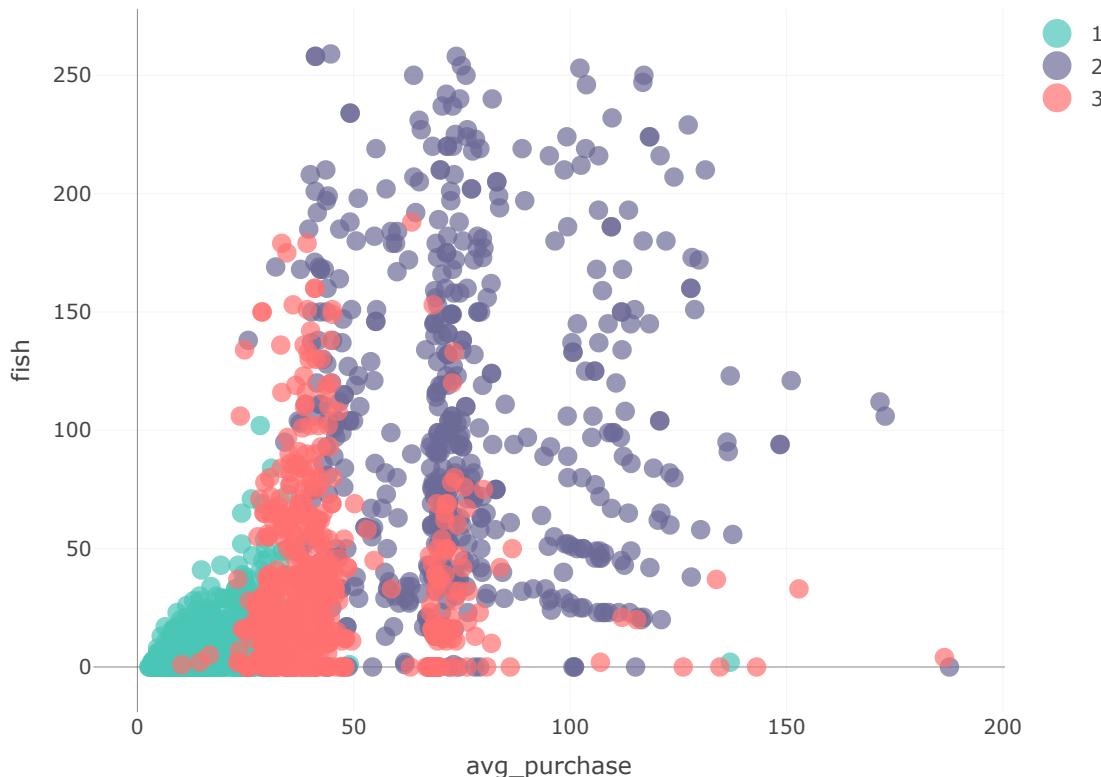
#Seguindo a lógica dos outros gráficos, temos que o grupo 2 seria os clientes com maior gasto em carne, isso já poderia auxiliar a empresa em definir o tipo de produto a ser ofertado através dos meios de comunicação para cada grupo, um produto mais premium para o grupo 2, intermediário para o 3 e mais barato ao grupo 1. Porém, diferentes dos gráficos de vinho e frutas, parece que o intervalo de gastos com carne é mais estreito que em outros casos, então parece não haver uma variação tão alta como nas outras

#Análise gráfica 3d - por cluster - fish

```
plot_ly(descricao,x=~avg_purchase,y=~fish,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:
## Based on info supplied, a 'scatter' trace seems appropriate.
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:
## Setting the mode to markers
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```



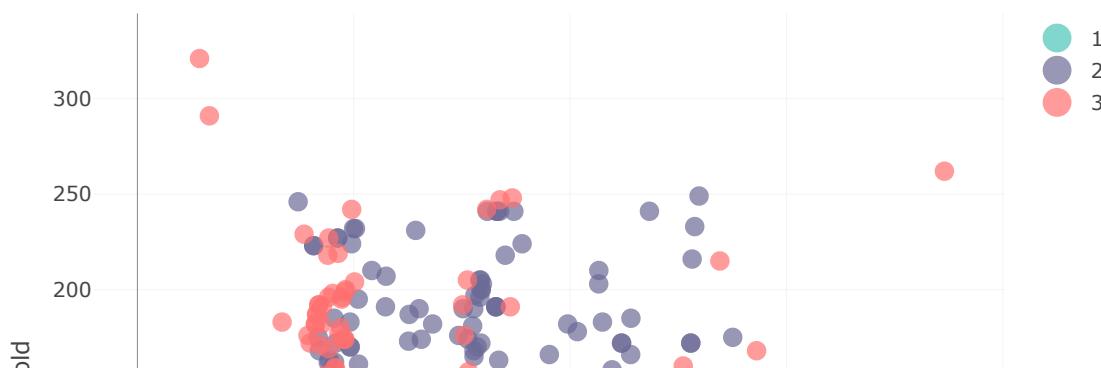
#Seguindo a lógica dos outros gráficos, temos que o grupo 2 seria os clientes com maior gasto em peixes, isso já poderia auxiliar a empresa em definir o tipo de produto a ser ofertado através dos meios de comunicação para cada grupo, um produto mais premium para o grupo 2, intermediário para o 3 e mais barato ao grupo 1.

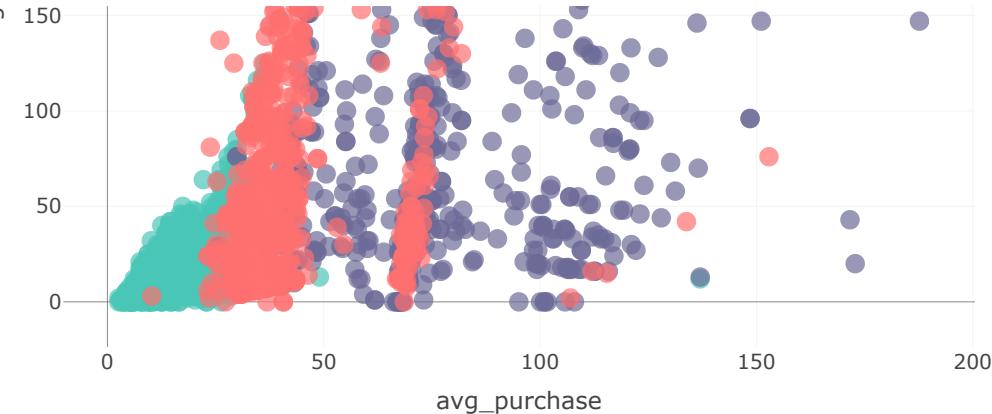
#Análise gráfica 3d - por cluster - gold

```
plot_ly(descricao,x=~avg_purchase,y=~gold,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:
## Based on info supplied, a 'scatter' trace seems appropriate.
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:
## Setting the mode to markers
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```





#Um pouco diferente da lógica dos outros gráficos, temos que o grupo 2 e 3 são bem similares nessa categoris, então nesse caso, pode ser que tenhamos a oportunitade de aumentar o ticket de gasto com o cluster 2 através de um comunicação mais direcionada.

#Análise gráfica 3d - por cluster - sweet

```
plot_ly(descricao,x=~avg_purchase,y=~sweet,color=~cluster,colors=colors,size=3)
```

```
## No trace type specified:  
## Based on info supplied, a 'scatter' trace seems appropriate.  
## Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specified:  
## Setting the mode to markers  
## Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```



#Seguindo a lógica dos outros gráficos, temos que o grupo 2 seria os clientes com maior gasto em doces, isso já poderia auxiliar a empresa em definir o tipo de produto a ser ofertado através dos meios de comunicação para cada grupo, um produto mais premium para o grupo 2, intermediário para o 3 e mais barato ao grupo 1.

#Using DBSCAN

-A presença de ruídos na base pode impactar consideravelmente os resultados obtidos através de análises como PCA e K-means, portanto iremos checar um terceira opção para verificar se conseguimos obter melhores resultados

```
# Compute DBSCAN using fpc package

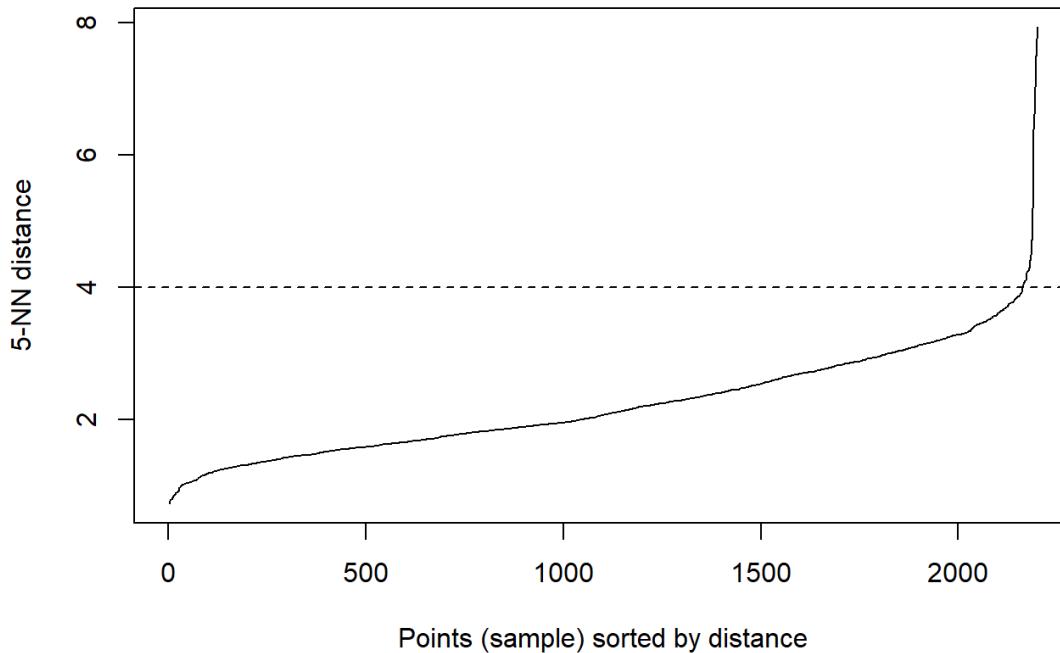
df3<-df %>%
  select(income,cust_retention_year,days_wo_purchase,wine,fruit,meat,fish,sweet,gold,web_visits_month,age,children,total_expense,total_purchases,avg_purchase,years_education,online_purchases,purchase_discount) %>%
  scale()
df3 %>% head()
```

```
##           income cust_retention_year days_wo_purchase      wine      fruit
## [1,]  0.2912158          1.50103013     0.3097519  0.9736661  1.5495429
## [2,] -0.2644933          -1.42007271    -0.3810966 -0.8739345 -0.6379120
## [3,]  0.9261301           0.04047871    -0.7956057  0.3548383  0.5689597
## [4,] -1.1926226          -1.42007271    -0.7956057 -0.8739345 -0.5624825
## [5,]  0.2985191          -1.42007271     1.5532792 -0.3942690  0.4181007
## [6,]  0.4973568           0.04047871    -1.1410300  0.6331628  0.3929576
##           meat       fish      sweet      gold web_visits_month      age
## [1,]  1.6859966  2.4595444  1.478795810  0.85529936     0.7227637  1.0175624
## [2,] -0.7193955 -0.6524375 -0.635297940 -0.73571048    -0.1277056  1.2740408
## [3,] -0.1804095  1.3428921 -0.149299377 -0.03721836    -0.5529403  0.3336200
## [4,] -0.6570334 -0.5059913 -0.586698084 -0.75511304     0.2975290 -1.2907432
## [5,] -0.2204993  0.1530167 -0.003499808 -0.56108745    -0.1277056 -1.0342648
## [6,] -0.3095879 -0.6890490  0.360999115 -0.58049001     0.2975290  0.1626344
##           children total_expense total_purchases avg_purchase years_education
## [1,] -1.26823386   1.6725024    1.3078664  1.20399148    -0.74131386
## [2,]  1.40166832  -0.9649756   -1.1967281 -1.02616306    -0.74131386
## [3,] -1.26823386   0.2774590    1.0295781  0.04464523    -0.74131386
## [4,]  0.06671723  -0.9218470   -0.9184398 -0.95655774    -0.74131386
## [5,]  0.06671723  -0.3097531    0.1947133 -0.24459472     1.40745499
## [6,]  0.06671723   0.1779315    1.0295781 -0.05558644    -0.02505758
##           online_purchases purchase_discount
## [1,]        0.27969199     -0.6070710
## [2,]      -0.66094206      1.5041365
## [3,]        0.58069489     -1.1084828
## [4,]        0.02885624      0.5364997
## [5,]        0.22594147      0.6747335
## [6,]      -0.24706308     -0.8181918
```

#Verificação do melhor eps

```
# to plot the eps values
eps_plot = kNNdistplot(df3, k=5)

# to draw an optimum line
eps_plot %>% abline(h = 4, lty = 2)
```



Plot resultados do DBSCAN

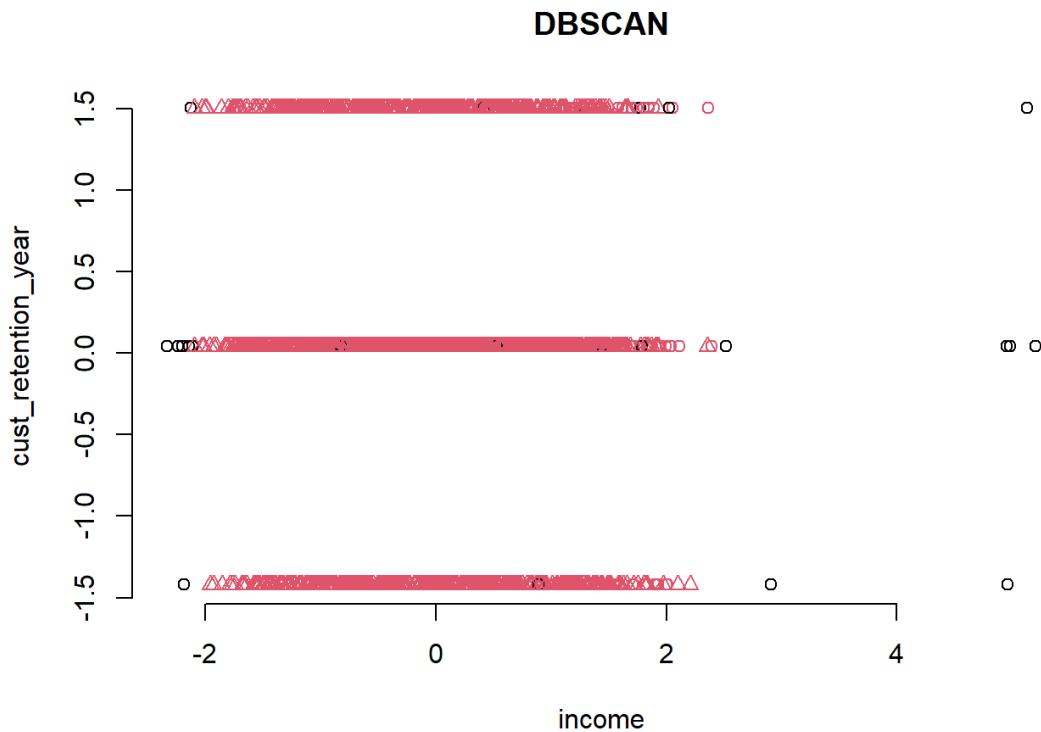
```
set.seed(42)
df_dbs<- df3 %>%
  fpc::: dbscan(MinPts=40,eps=4)

df_dbs
```

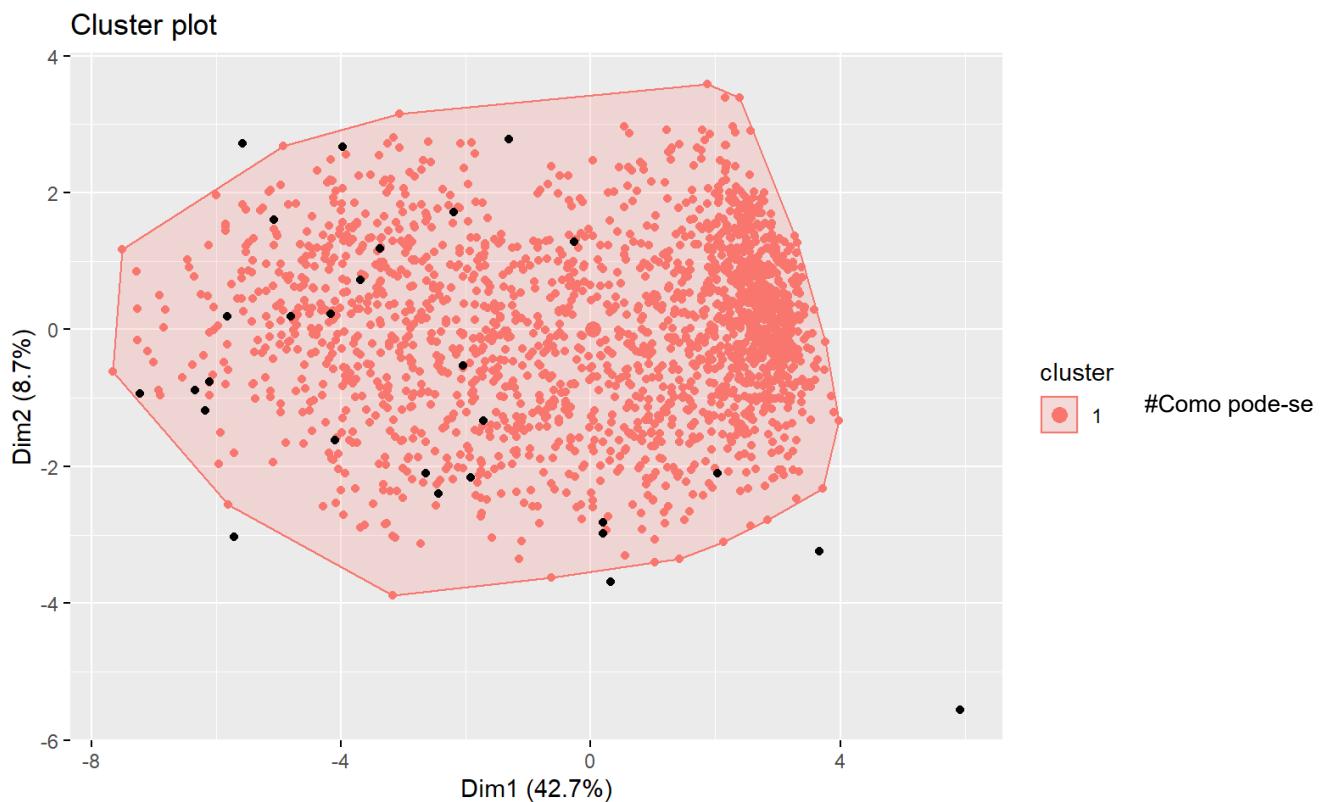
```
## dbscan Pts=2201 MinPts=40 eps=4
##      0   1
## border 28 230
## seed    0 1943
## total   28 2173
```

```
# Plot DBSCAN results

plot(df_dbs,df3,main="DBSCAN",frame=FALSE)
```



```
fviz_cluster(df_dbs, df3, stand = FALSE, frame = FALSE, geom = "point")
```



observar, como nossa base é muito homogênea e está concentrada numa única região, ao utilizamos o DBSCAN, o número de cluster ideal encontrado é 1. Portanto, essa opção não seria a mais viável no nosso caso atual, pois a informação trazida pelo k-means nos permite aprofundar melhor nossas análises.

#CONCLUSÃO:

#A análise realizada através do PCA e Cluster nos permite segmentar os clientes facilitando o planejamento de atuação em cima do público-alvo: Começando pelo público-alvo, já foi possível perceber que o público se encontra na classe C a E, normalmente a compra desse público é norteado principalmente pelo preço, e a qualidade pode acabar ficando em segundo, isso pode ser um dos motivos pelo qual a disponibilização de desconto tem tanto impacto sobre a variação dos valores. #Além disso, o consumidor ter filhos parece ser uma variável de grande impacto, então valeria como um segundo passo analisar mais a fundo a relação

inversamente proporcional de gastos e compras com desconto com o número de filhos do consumidor. #Por último, percebe-se não só pelo PCA, mas como pelo K-means também que os gastos em cada categoria é diretamente proporcional à classe econômica do consumidor, então é importante que uma comunicação direcionada seja realizada para cada grupo nos veículos de comunicação mais apropriados a fim de termos uma receita otimizada futuramente.

#Por fim, como um dos principais drivers da definição dos cluster seria o gastos médio em compras, ao termos um predição de compras por cliente através de modelos preditivos, isso poderia nos ajudar a definir de antemão em qual segmento o cliente se encontraria, e consequentemente definiria o melhor plano de ação (taylor made) junto à esse cliente para otimizarmos a receita da empresa e minimizarmos o custo com promoções, já que vimos que não é efetivo do mesmo modo em todos os clusters.

#Predição

```
df_pred<- df %>%
  select(-id,-year_birth,-marital_status,-kidhome,-teenhome,-wine,-fruit,-meat,-fish,-sweet,-gold,-accepted_cmp
3,-accepted_cmp4,-accepted_cmp5,-accepted_cmp1,-accepted_cmp2 , -social_class,-age_class,-total_purchases,-respo
nse,-education,-web_purchases,-catalog_purchases,-store_purchases,-education_2,-purchase_discount2,-total_expen
se
  )
glimpse(df_pred)
```

```
## Rows: 2,201
## Columns: 13
## $ income      <int> 58138, 46344, 71613, 26646, 58293, 62513, 55635, 3~
## $ cust_retention_year <dbl> 2, 0, 1, 0, 0, 1, 2, 1, 1, 0, 2, 1, 1, 2, 2, 2, ~
## $ days_wo_purchase <int> 58, 38, 26, 26, 94, 16, 34, 32, 19, 68, 59, 82, 53~
## $ deals_purchases <int> 3, 2, 1, 2, 5, 2, 4, 2, 1, 1, 1, 3, 1, 1, 3, 2, ~
## $ web_visits_month <int> 7, 5, 4, 6, 5, 6, 6, 8, 9, 20, 8, 2, 6, 8, 3, 8, 7~
## $ complain        <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ age              <dbl> 57, 60, 49, 30, 33, 47, 43, 29, 40, 64, 38, 55, 62~
## $ children         <int> 0, 2, 0, 1, 1, 1, 1, 1, 2, 0, 0, 2, 0, 0, 2, 0, ~
## $ married          <dbl[,1]> <matrix[26 x 1]>
## $ online_purchases <dbl> 0.3636364, 0.2500000, 0.4000000, 0.3333333, 0.~
## $ years_education   <dbl[,1]> <matrix[26 x 1]>
## $ purchase_discount <dbl> 0.13636364, 0.50000000, 0.05000000, 0.33333333, 0.~
## $ avg_purchase      <dbl> 73.500000, 6.750000, 38.800000, 8.833333, 30.1~
```

#Treinamento e teste

```
set.seed(42)

split<-initial_split(df_pred,prop=0.8)
split
```

```
## <Analysis/Assess/Total>
## <1760/441/2201>
```

```
treinamento<- training(split)
teste<- testing(split)
```

#Criação de uma receita

```
receita <- recipe(avg_purchase~ ., data = treinamento)
receita
```

```
## Recipe
##
## Inputs:
##
##     role #variables
##     outcome      1
##     predictor    12
```

#Normalização

```
receita <- receita %>%
  step_normalize(income,cust_retention_year,days_wo_purchase,
                 deals_purchases,web_visits_month,age,children,online_purchases,years_education,purchase_discount) %>%
  prep() #Preparação da receita

receita
```

```
## Recipe
##
## Inputs:
##
##   role #variables
##   outcome      1
##   predictor    12
##
## Training data contained 1760 data points and no missing data.
##
## Operations:
##
## Centering and scaling for income, cust_retention_year, days_wo_purchase, ... [trained]
```

#receita foi definida, mas ainda não foi aplicada

#Bake:aplicação da receita no conjunto de dados

```
tr_proc<-bake(receita,new_data = NULL)
tst_proc<-bake(receita,new_data= teste)
```

#Modelo Linear

```
lm_fit<-linear_reg(mode="regression",engine = "lm") %>%
  fit(avg_purchase ~ .,tr_proc)

lm_fit<-linear_reg() %>%
  set_engine("lm") %>%
  fit(avg_purchase ~ .,tr_proc)

lm_fit
```

```
## parsnip model object
##
## Call:
## stats::lm(formula = avg_purchase ~ ., data = data)
##
## Coefficients:
## (Intercept)           income  cust_retention_year
##            38.16329        20.92044         2.88898
## days_wo_purchase     deals_purchases  web_visits_month
##             0.58613          2.73503         2.56901
## complain1              age            children
##            -5.29157          0.01854        -8.62495
## married                online_purchases years_education
##             -1.05775          -0.85634        -0.57375
## purchase_discount
##             -0.48782
```

```
tidy(lm_fit)
```

```
## # A tibble: 13 x 5
##   term          estimate std.error statistic p.value
##   <chr>        <dbl>     <dbl>      <dbl>    <dbl>
## 1 (Intercept)  38.2      0.715     53.4     0
## 2 income       20.9      0.764     27.4    1.25e-137
## 3 cust_retention_year 2.89      0.472     6.12    1.17e- 9
## 4 days_wo_purchase 0.586      0.426     1.38    1.69e- 1
## 5 deals_purchases 2.74      0.661     4.14    3.66e- 5
## 6 web_visits_month 2.57      0.716     3.59    3.44e- 4
## 7 complain1     -5.29      4.48     -1.18    2.38e- 1
## 8 age           0.0185    0.442     0.0419  9.67e- 1
## 9 children      -8.62      0.610    -14.1    4.99e-43
## 10 married      -1.06      0.889    -1.19    2.34e- 1
## 11 online_purchases -0.856    0.516    -1.66    9.70e- 2
## 12 years_education -0.574    0.432    -1.33    1.84e- 1
## 13 purchase_discount -0.488    0.943    -0.518   6.05e- 1
```

#Previsão

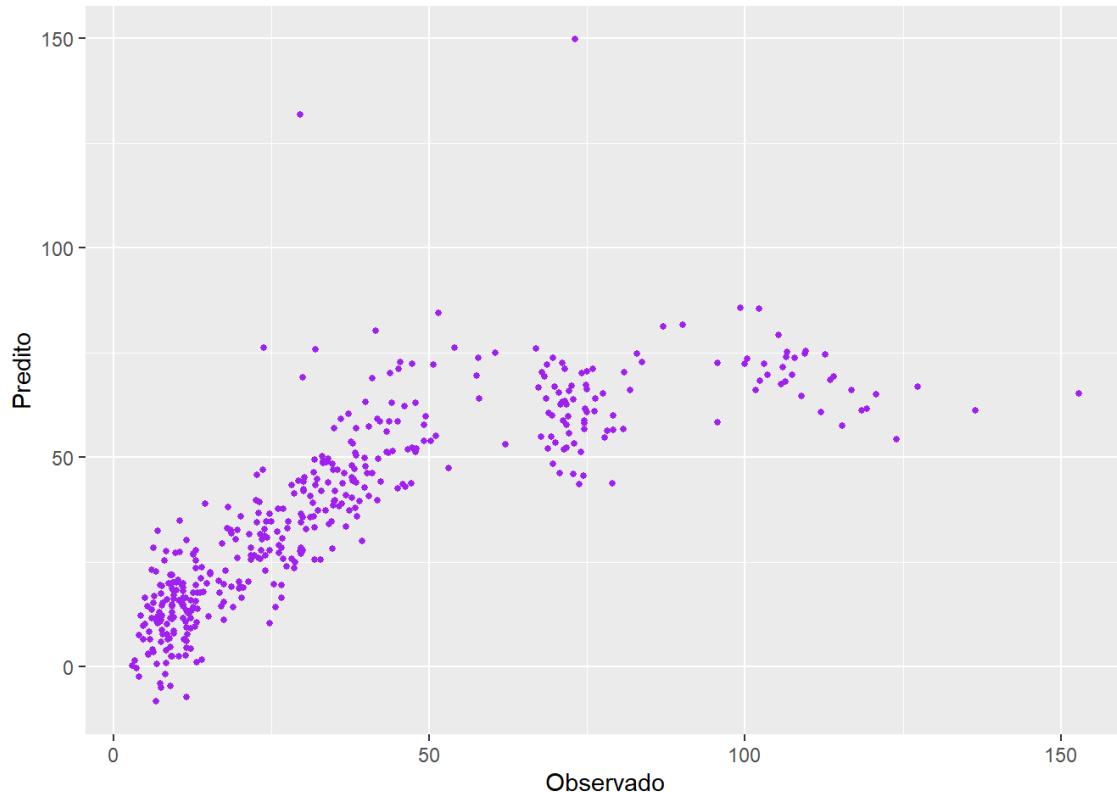
```
fitted_lm<- lm_fit %>%
  predict(new_data= tst_proc) %>%
  mutate(observado= tst_proc$avg_purchase,
    modelo="lm")

head(fitted_lm)
```

```
## # A tibble: 6 x 3
##   .pred observado modelo
##   <dbl>    <dbl> <chr>
## 1 41.9     30.1 lm
## 2 48.4     34.7 lm
## 3 43.5     32.0 lm
## 4 34.0     34.2 lm
## 5 76.0     66.9 lm
## 6 23.6     13.0 lm
```

#Observado Vs Predito

```
fitted_lm %>%
  ggplot(aes(observado,.pred)) +
  geom_point(size=1,col="purple") +
  labs(x="Observado",y="Predito")
```



#Random Forest

```
rf<- rand_forest(mtry=tune(),trees = tune(), min_n = tune(),
                  mode = "regression") %>%
  set_engine("ranger", importance = "permutation")
```

#Ajuste de hiperparâmetros

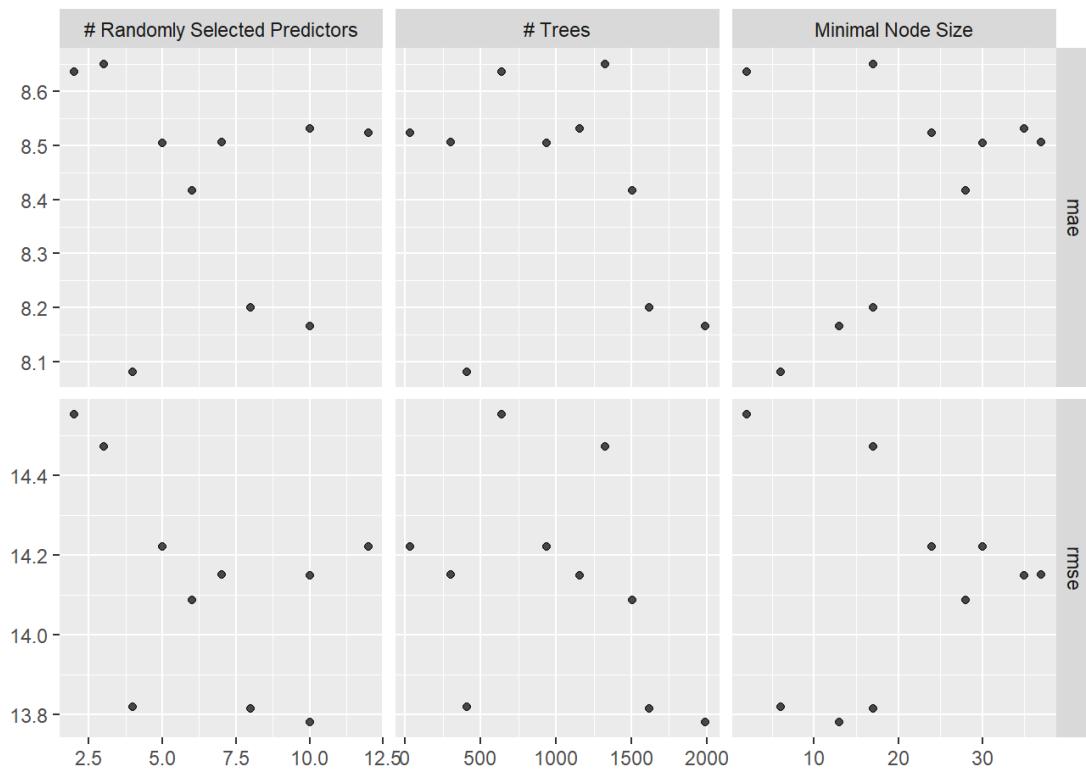
```
cv_split<-vfold_cv(treinamento,v=10)
```

```
doParallel::registerDoParallel()
```

```
rf_grid<-tune_grid(rf,
                     receita,
                     resamples = cv_split,
                     grid = 10,
                     metrics = metric_set(rmse,mae))
```

```
## i Creating pre-processing data to finalize unknown parameter: mtry
```

```
autoplot(rf_grid)
```



```
rf_grid %>%
  collect_metrics() %>%
  head()
```

```
## # A tibble: 6 x 9
##   mtry trees min_n .metric .estimator  mean     n std_err .config
##   <int> <int> <int> <chr>    <chr>    <dbl> <int>   <dbl> <chr>
## 1    10   1156    35 mae     standard  8.53    10  0.192 Preprocessor1_Model01
## 2    10   1156    35 rmse    standard 14.1     10  0.426 Preprocessor1_Model01
## 3     6   1508    28 mae     standard  8.42    10  0.180 Preprocessor1_Model02
## 4     6   1508    28 rmse    standard 14.1     10  0.453 Preprocessor1_Model02
## 5     4    410     6 mae     standard  8.08    10  0.165 Preprocessor1_Model03
## 6     4    410     6 rmse    standard 13.8     10  0.467 Preprocessor1_Model03
```

#Opción com menor RMSE

```
best<-rf_grid %>%
  select_best("rmse")

best
```

```
## # A tibble: 1 x 4
##   mtry trees min_n .config
##   <int> <int> <int> <chr>
## 1    10   1988    13 Preprocessor1_Model05
```

```
#Modelagem

rf_fit<-finalize_model(rf,parameters = best) %>%
  fit(avg_purchase~,tr_proc)

#Predição

fitted_rf<-rf_fit %>%
  predict(new_data=tst_proc) %>%
  mutate(observado=tst_proc$avg_purchase,
        modelo='random forest')

fitted_rf
```

```
## # A tibble: 441 x 3
##   .pred observado modelo
##   <dbl>    <dbl> <chr>
## 1 35.0     30.1 random forest
## 2 40.2     34.7 random forest
## 3 37.9     32.0 random forest
## 4 32.8     34.2 random forest
## 5 66.9     66.9 random forest
## 6 15.5     13.0 random forest
## 7 38.3     41.0 random forest
## 8 15.7     17.4 random forest
## 9 51.6     60.5 random forest
## 10 84.0    80.8 random forest
## # ... with 431 more rows
```

```
fitted<- fitted_lm %>%
  bind_rows(fitted_rf)

fitted %>%
  group_by(modelo) %>%
  metrics(truth=observado,estimate=.pred)
```

```
## # A tibble: 6 x 4
##   modelo      .metric .estimator .estimate
##   <chr>       <chr>   <chr>        <dbl>
## 1 lm         rmse    standard     18.0
## 2 random forest rmse    standard     13.5
## 3 lm         rsq     standard     0.657
## 4 random forest rsq     standard     0.809
## 5 lm         mae     standard     12.2
## 6 random forest mae     standard     8.22
```

#Decision Tree

```
#Modelo

tree <- decision_tree(tree_depth= tune(), min_n = tune(),
                      mode="regression",engine="rpart")

cv_split <- vfold_cv(treinamento, v = 10)

doParallel::registerDoParallel()

tree_grid <- tune_grid(tree,
                       receita,
                       resamples = cv_split,
                       grid = 10,
                       metrics = metric_set(rmse, mae))

best <- tree_grid %>%
  select_best("rmse")

#Modelagem após a escolha dos hiperparâmetros

tree_fit<-finalize_model(tree,parameters = best) %>%
  fit(avg_purchase~.,tr_proc)

#Predição

fitted_tree<- tree_fit %>%
  predict(new_data=tst_proc) %>%
  mutate(observado=tst_proc$avg_purchase,
        modelo="decision tree")

fitted<- bind_rows(fitted,fitted_tree)
```

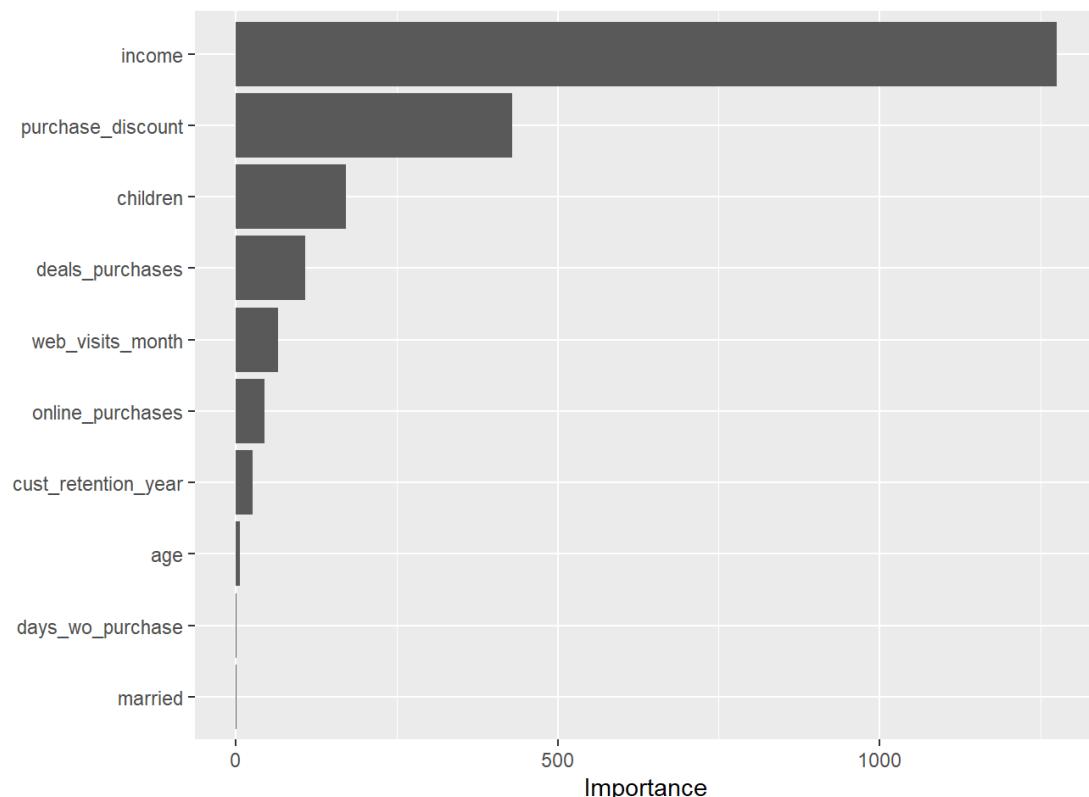
#Comparação entre modelos ajustados

```
fitted %>%
  group_by(modelo) %>%
  metrics(truth=observado,estimate=.pred)
```

```
## # A tibble: 9 x 4
##   modelo      .metric .estimator .estimate
##   <chr>       <chr>    <chr>      <dbl>
## 1 decision tree rmse    standard    17.1 
## 2 lm          rmse    standard    18.0 
## 3 random forest rmse    standard    13.5 
## 4 decision tree rsq     standard    0.689 
## 5 lm          rsq     standard    0.657 
## 6 random forest rsq     standard    0.809 
## 7 decision tree mae    standard    12.4 
## 8 lm          mae    standard    12.2 
## 9 random forest mae    standard    8.22
```

#Levando-se em consideração o RMSE, o melhor modelo seria a floresta aleatória

```
vip(rf_fit)
```



#variáveis que mais contribuíram foram a renda e compras com desconto