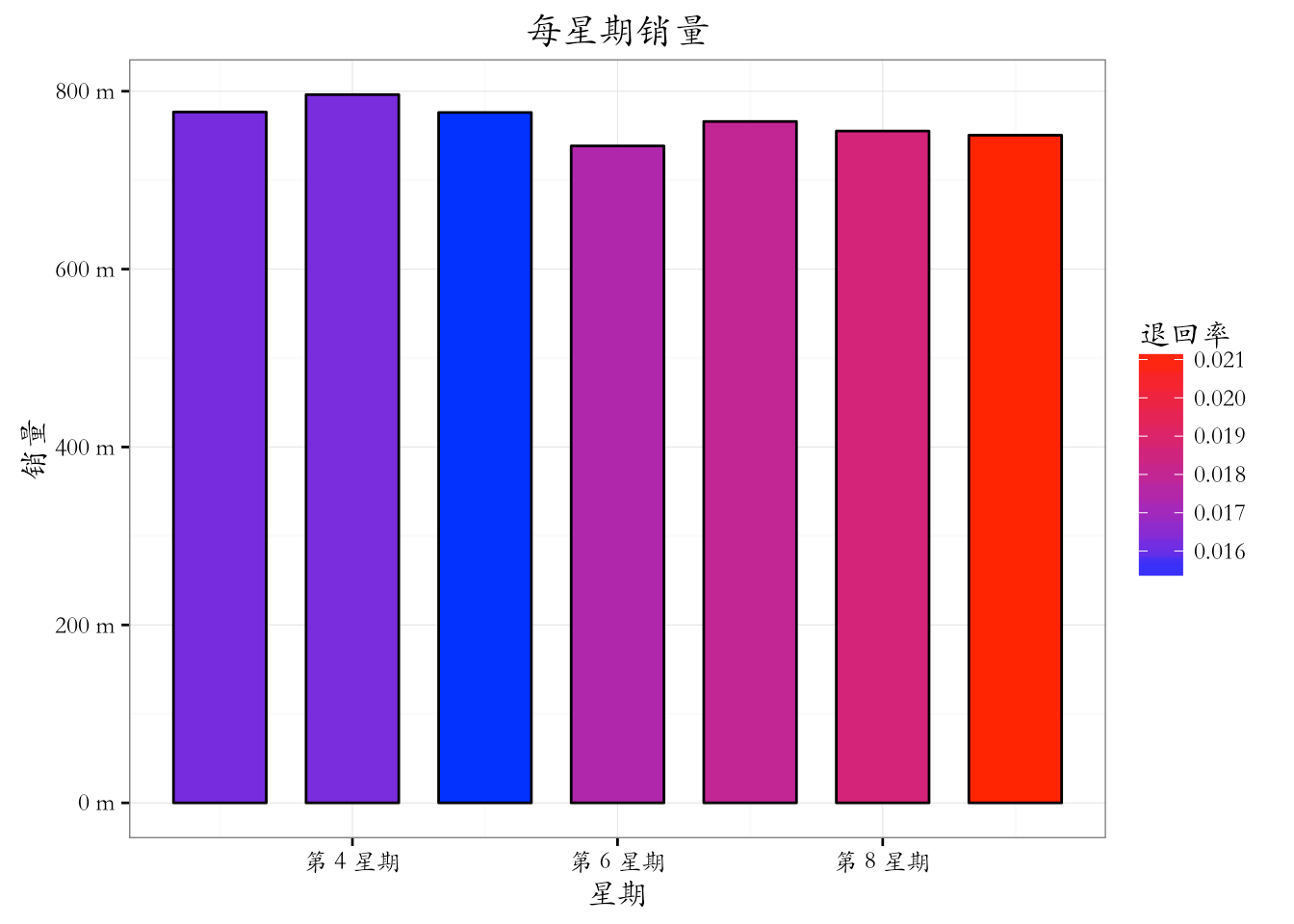
**数据分析：**

**描述性统计**

1、分析数据的周数：Semana  
## 条形图

可视化每周的情况

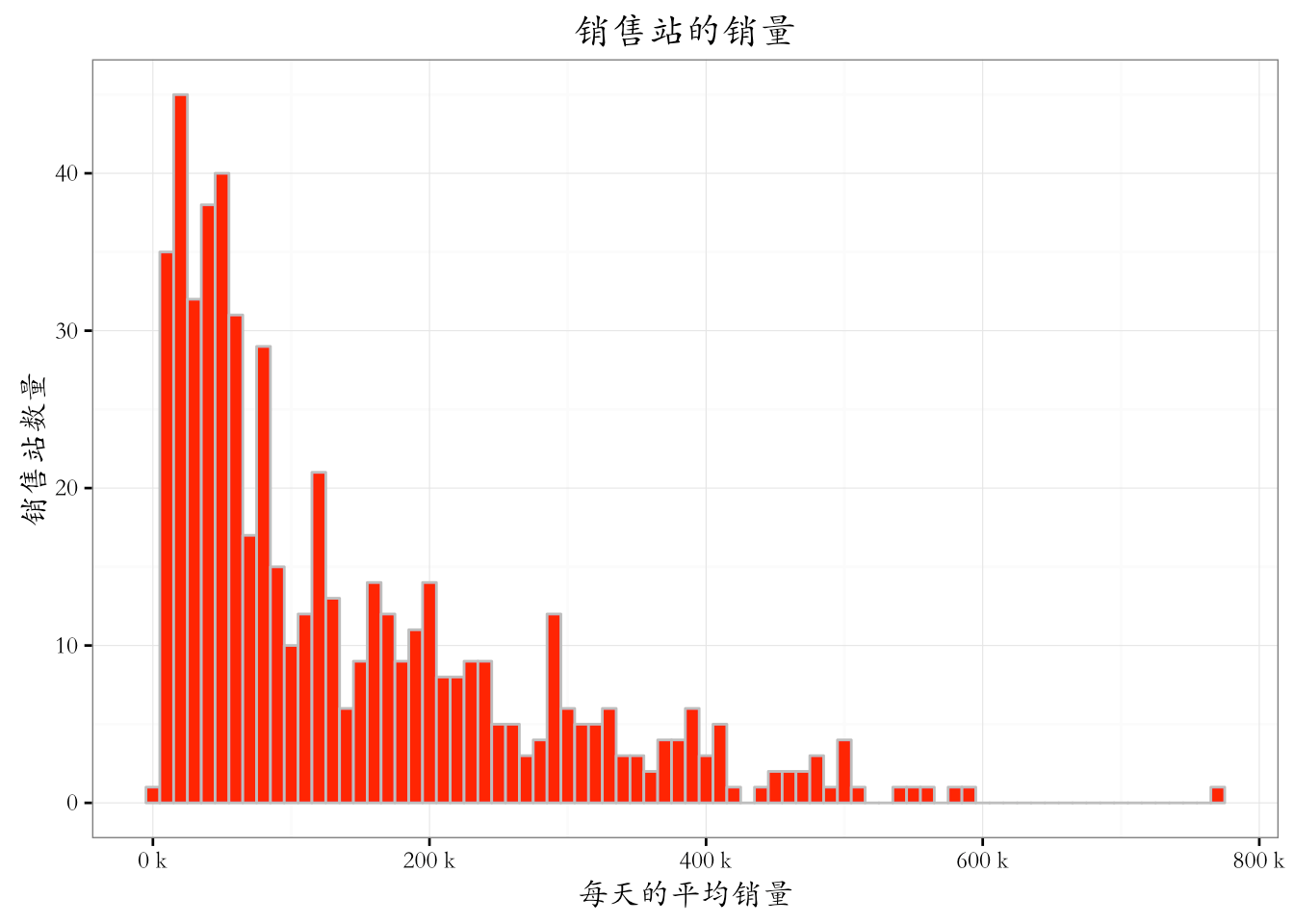


分析：该图表示每个星期的销售量，整体来看都相对较平均，相差不是很大。第五周的退货率最小，但是整天来看随着周数的增长，产品退货率越来越大，说明产品退货率有逐渐增加的趋势。

2、销售站的数据分析  
销售站Agencia 和 州State

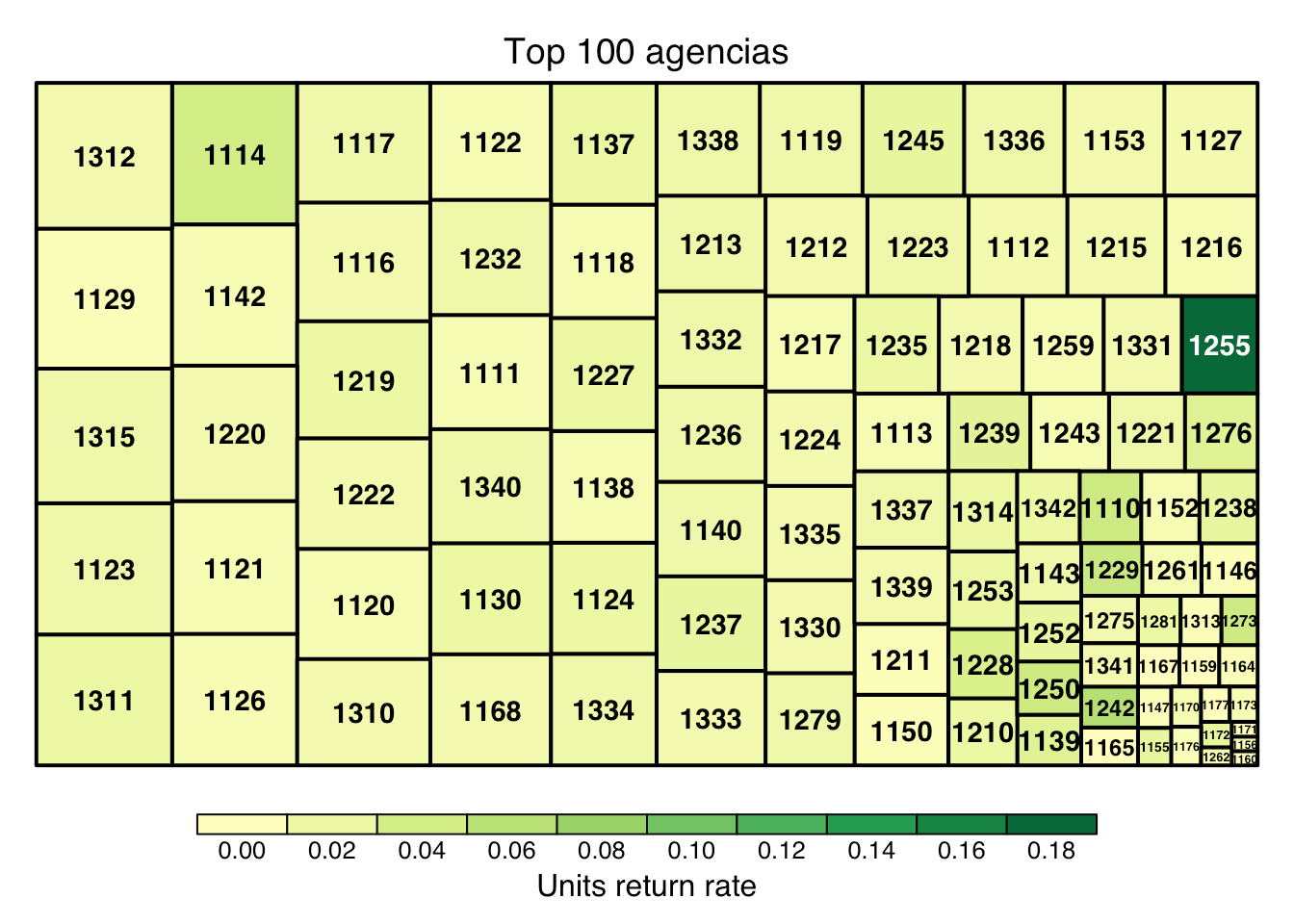
分析：一共有552个销售站

可视化x：每天销量，y：销售站的数量



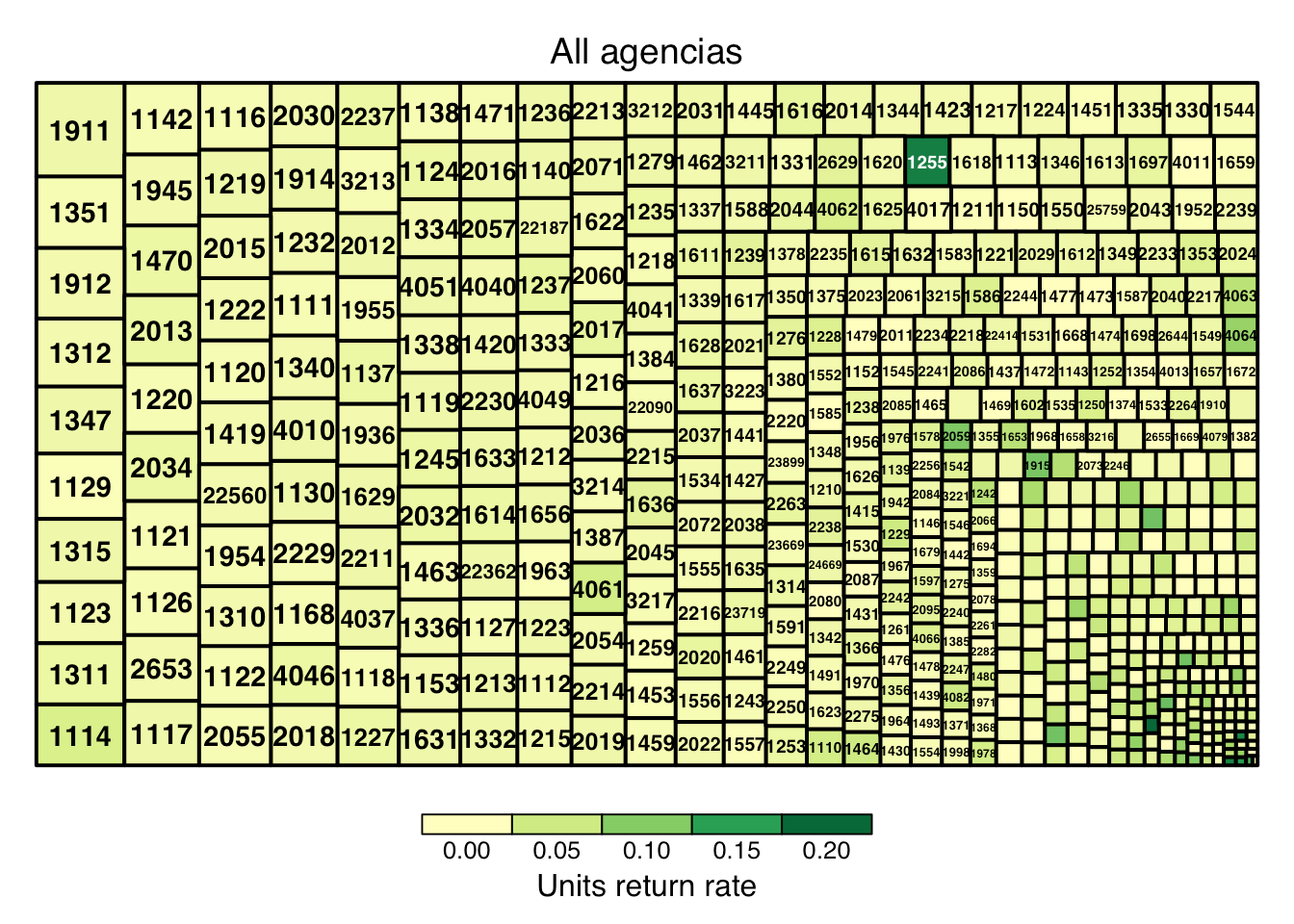
分析：由图可知，每个销售站每天的平均销量分布相对较集中，大多数都分布在0~200k期间，说明多数的销售站销量还是一般，只有少数的销量较高，达到400k的销售站数量更少。

前100的销售站的销量树图



分析：在前100的销售站中，由其图形面积代表销量，不同颜色确定其退回率。由图可知ID为1312、1129的销售站销量最大，且退货率比较低；ID为1250、1139的销售站则销量较小，退货率稍高；ID为1255的销售站退回率最高，近达18%。

## 所有销售站的销量树图



分析：图表显示了552个销售站的销售量情况及退货率。图形面积大小代表了销量的多少，不同颜色呈现了产品退货率的高低。由图形面积可知ID为1911、1351、1912、1312、1347的销售站销售量比较高，退货率也相对较低；而ID为1255的销售站退货率最高，约为20%。整体来看，销售量小的销售站的退货率也相对较高。

## 销售站的历史数据分析

## Agencia\_ID Semana Units Pesos Return\_Units Return\_Pesos Net

## 1 1110 3 120285 1296739 3577 29712.03 119951

## 2 1110 4 135788 1385112 5260 29932.82 135327

## 3 1110 5 127420 1345680 3342 26249.25 127077

## 4 1110 6 115255 1239051 5721 26595.87 114865

## 5 1110 7 122955 1297072 6950 34845.26 122513

## 6 1110 8 127277 1345696 8030 35157.98 126735

## Net\_Pesos Avg\_Pesos Return\_Rate Town State

## 1 1267027 10.78055 0.02887891 2008 AG. LAGO FILT MÉXICO, D.F.

## 2 1355179 10.20055 0.03729227 2008 AG. LAGO FILT MÉXICO, D.F.

## 3 1319431 10.56098 0.02555788 2008 AG. LAGO FILT MÉXICO, D.F.

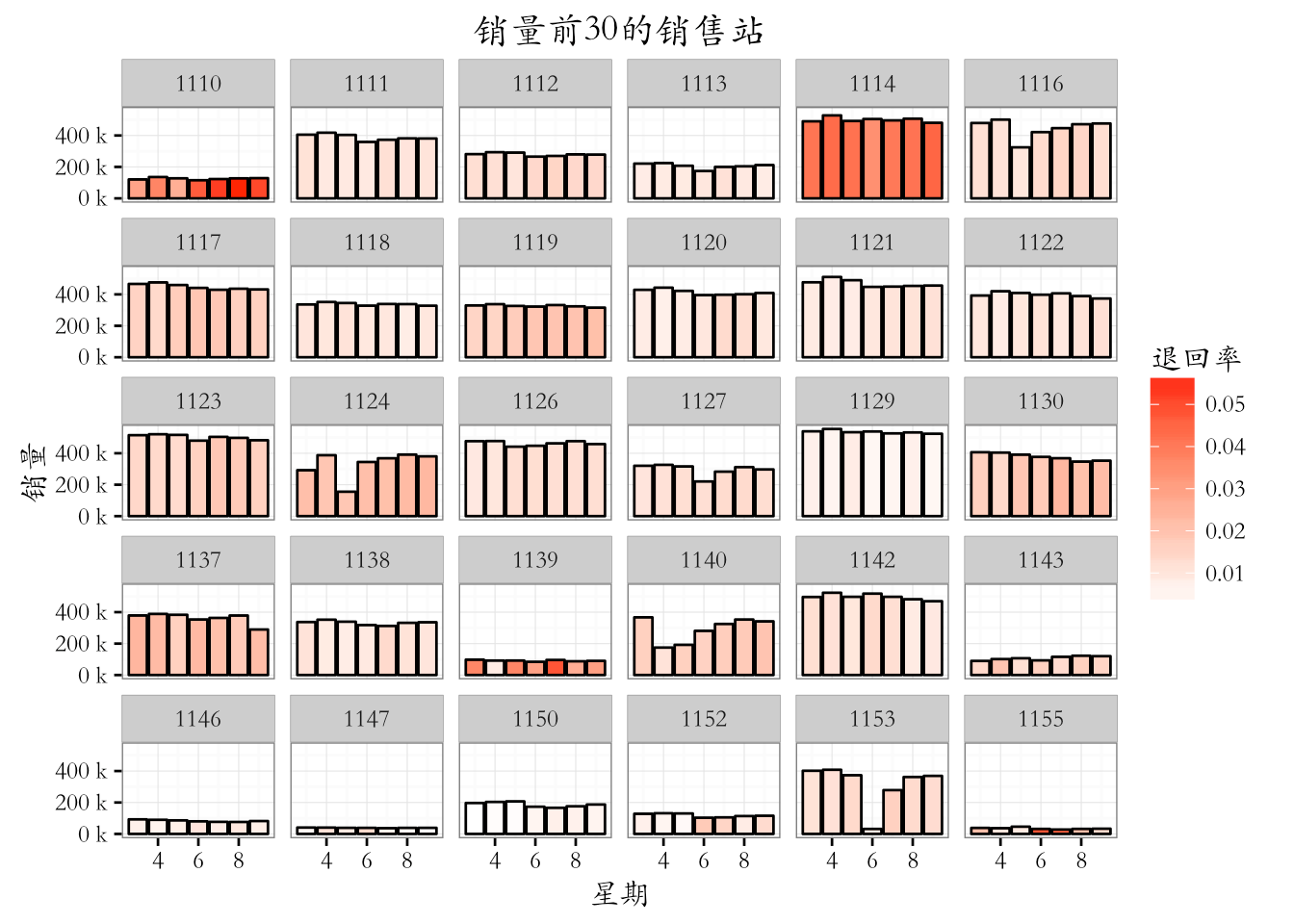
## 4 1212456 10.75052 0.04729037 2008 AG. LAGO FILT MÉXICO, D.F.

## 5 1262227 10.54916 0.05350064 2008 AG. LAGO FILT MÉXICO, D.F.

## 6 1310538 10.57297 0.05934652 2008 AG. LAGO FILT MÉXICO, D.F.

dim(agencias\_history)

## [1] 3863 12

## 取出销售量前30的销售站ID

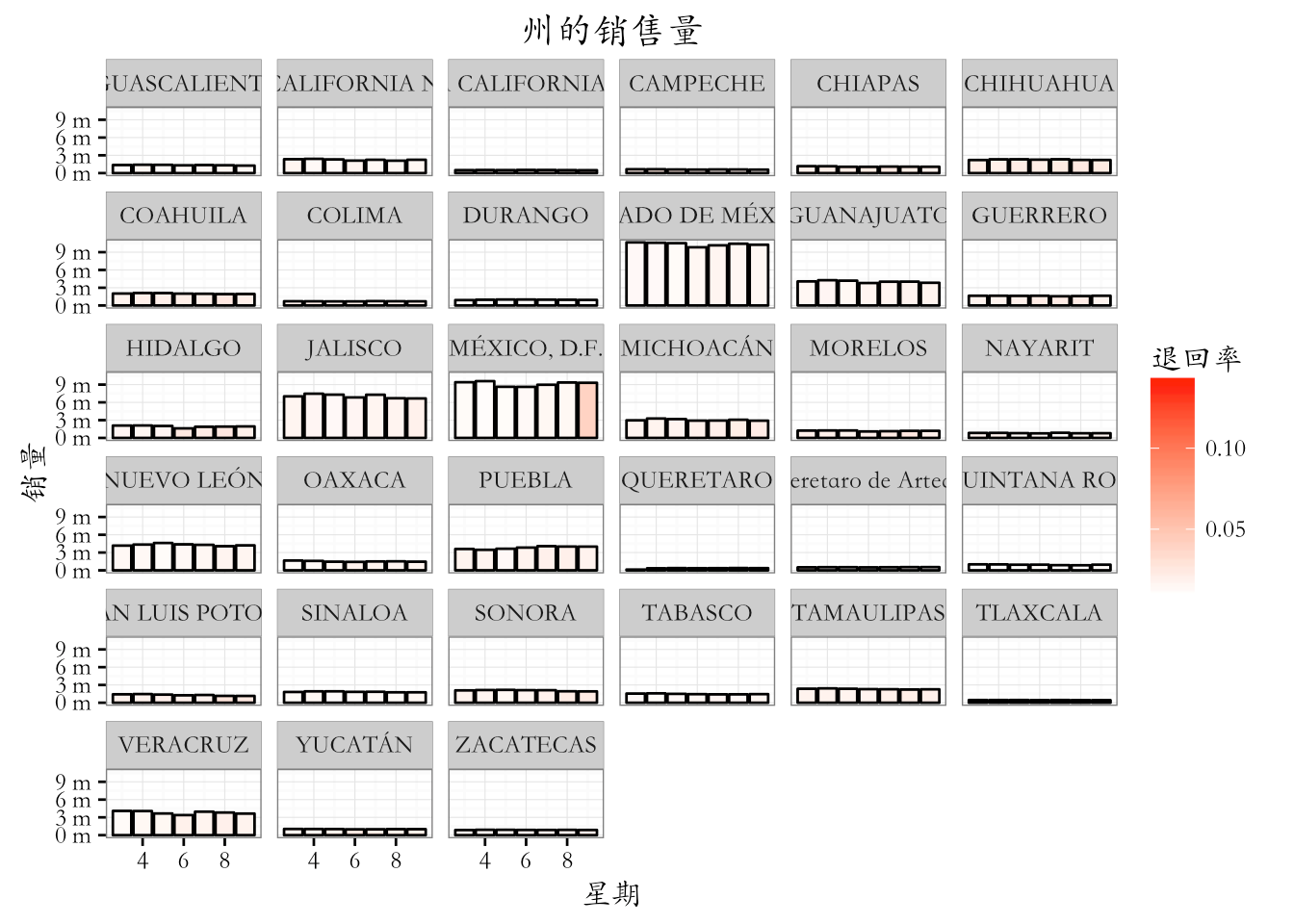
分析：在销量为前30的销售站中，在一星期中，其各自销量都较平均，退回率也相差不多。例如，销售站1114每天销量都较靠前，但是每天的退回率也都较高；销售站1129销量很高，退回率很低，还有1110销量低，退回率高，1146销量低，退回率低。

分析：在销量为前30的销售站中，有21个销售站每星期的销售量都在200k之上，有8个销售站每周的销售量都超过400k；大部分销售站周四到周三的销量均衡，只有1124,1140,1153会有大的波动；大部分的销售站退货率都在30%以下，只有1110，1114,1139销售站的退货率将近5%，1139销售站在七周之内的退货率有较大的波动。1114的销量与退货率在销量前30的销售站中都是最高的，可能该店的宣传力度够好，然而产品质量及售后不到位。

3、每个州的销售数据的分析

一共有33个州。

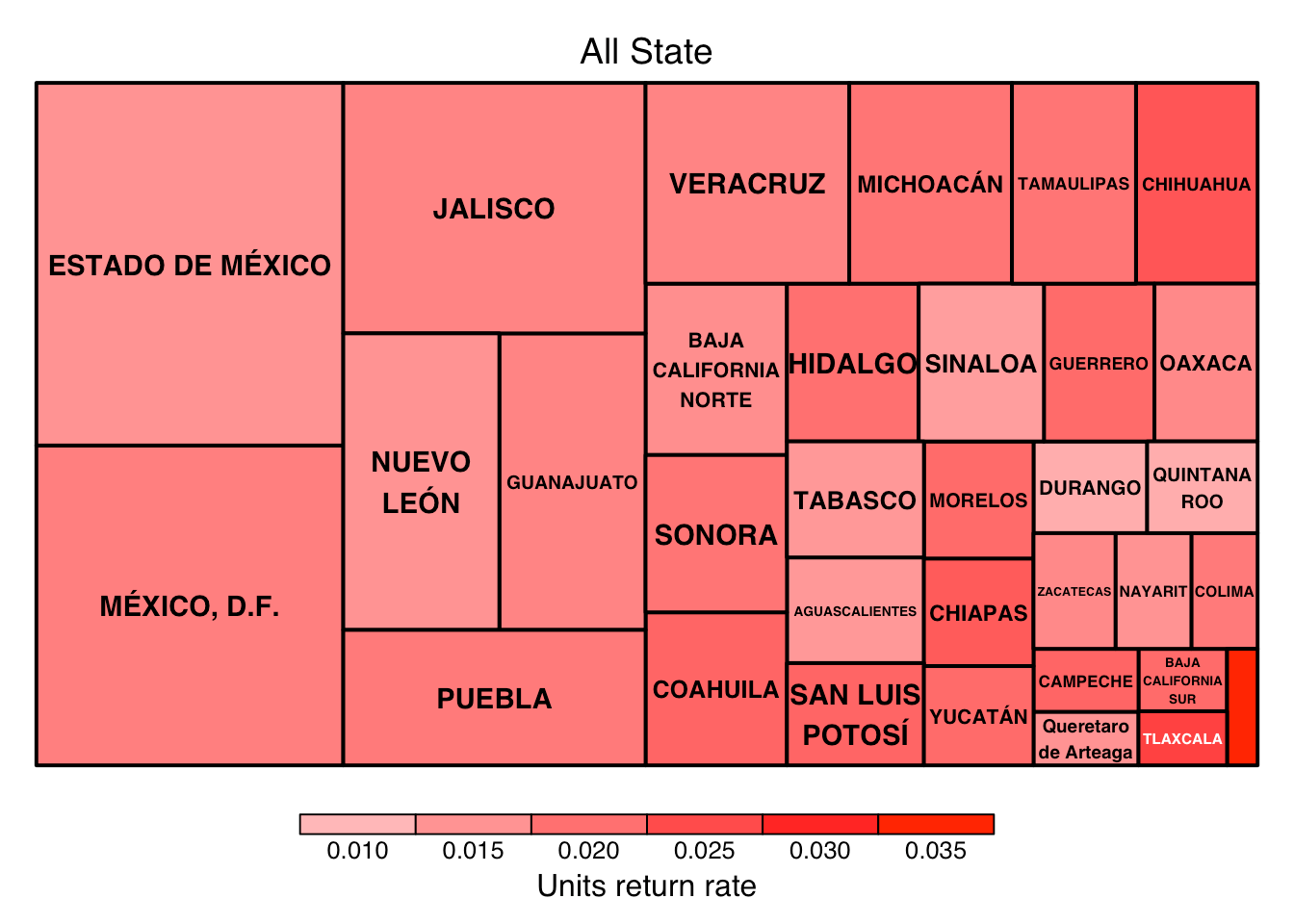
## 地点－－星期 －－退回百分比 图像可视化



分析：该图按照州的分布进行一星期的销售量统计，并对每个州的退回率也进行统计。只有几个州的销量较高，例如，ADO DE MEX和MEXICO.D.F.,大体销量销量都还可以。每个州的每天销量分布相对较平均，退货率也都相对较低，不到0.05。

分析：该图按照州的分布对七周的销售量进行统计，并对退货率也做相应的统计。销售量前三的州分别是是ADO DE MEX，JALISCO，MEXICO.D.F，较高的有GUANAJUATC,MICHOACAN,NUEVO LEON,PUEBLA,VERACRUZ，剩下其它州的销量都比较低。在七周内每个州的销量分布及退货率基本都维持在同一水平上，且退货率都比较低，只有MEXICO，在第九周的退货率明显偏高。

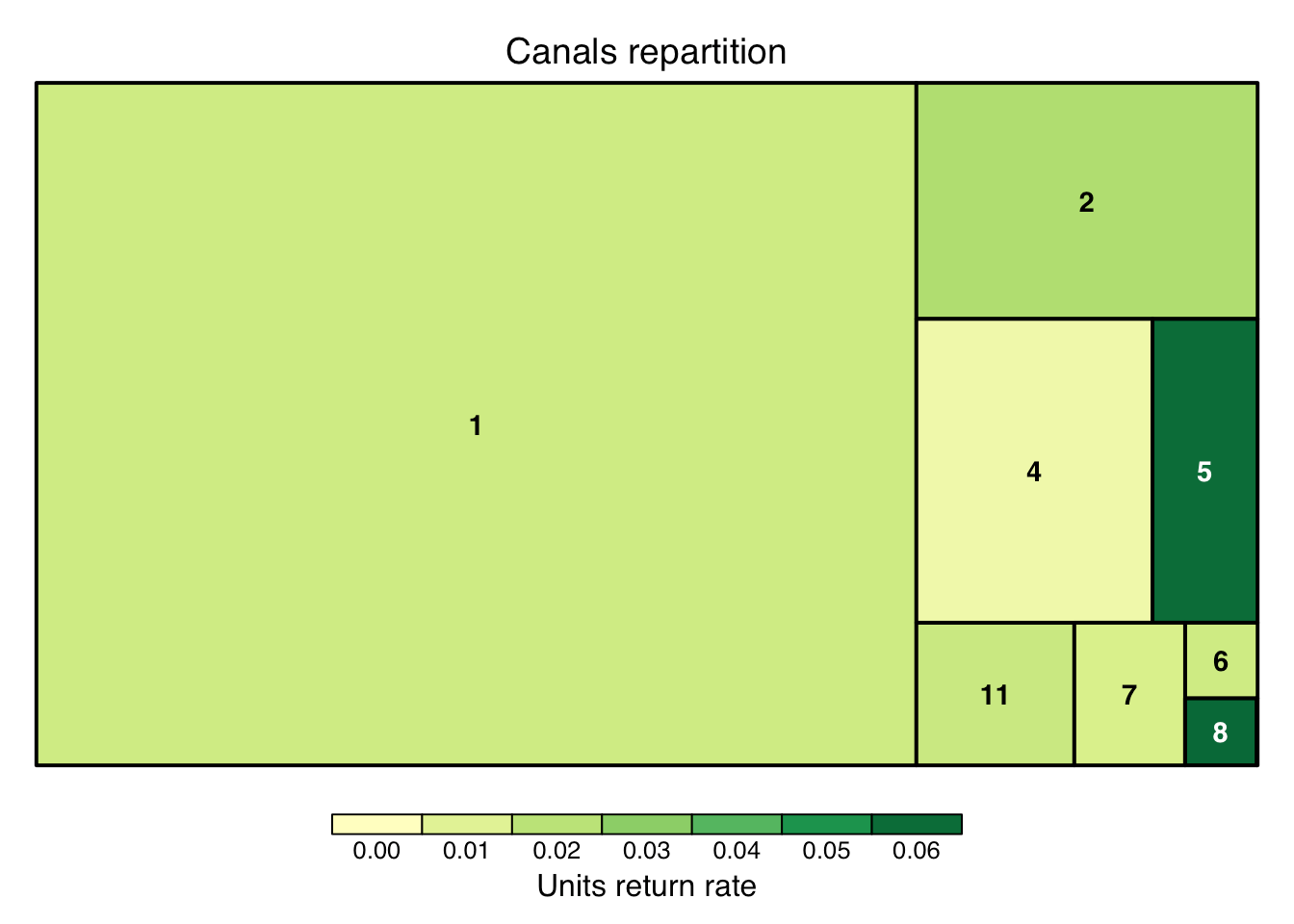
## 州的销量树图



分析：该图反映了所有州的销售量及其退货率，从图中可以看出，销售量前三的州分别为ESTADD DE MEXICO、MEXICO.D.F与JALISCO，所有州的退货率基本都维持在[0.015,0.025]之内。ESTADD DE MEXICO的销量最高，退货率相对较低，其他州都可以将其销售方法进行借鉴与参考。而销售量最小的州退货率也最高，约可达3.5%。

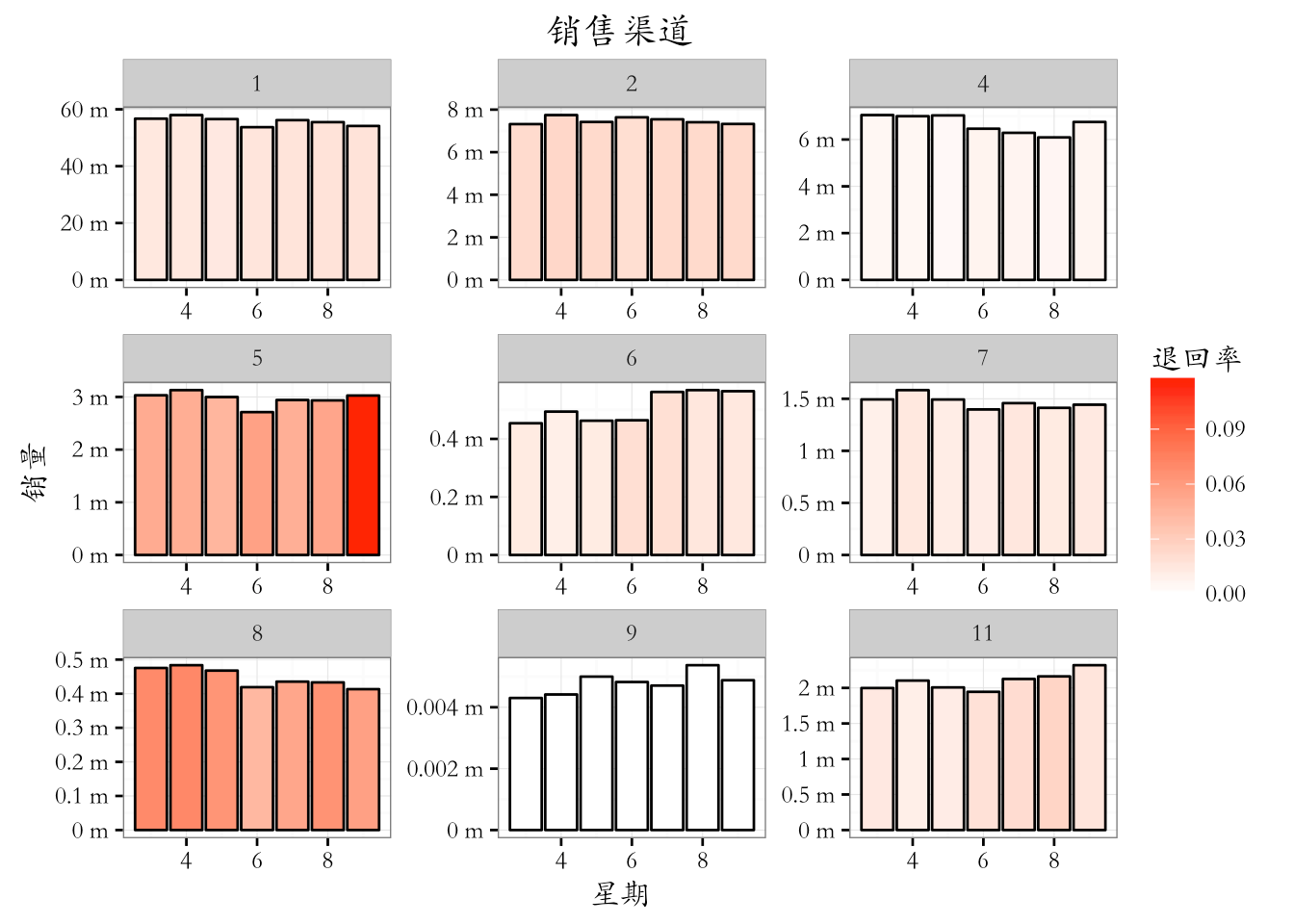
4、销售渠道的分析

一共有9个销售渠道。



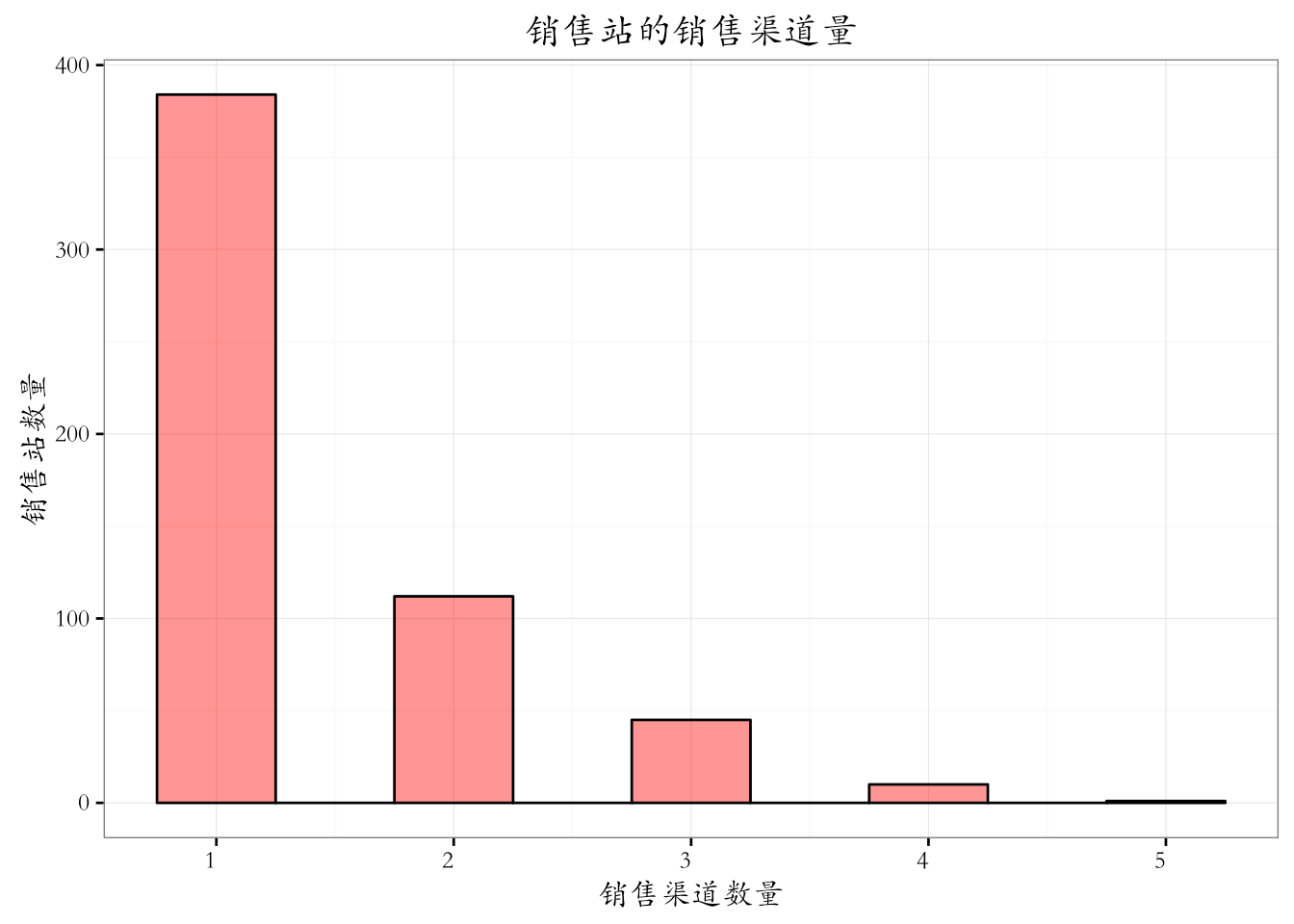
分析：主要有8种销售渠道，其中销售渠道1占销售量的绝大部分，退货率约为2%，销售渠道4的退货率最低，几乎为0，而销售渠道5和8的退货率最高，约为6%。由此可知，大部分顾客都更愿意选择销售渠道1，并且从退货率来看，销售渠道1的产品销售方式更好。

5、销售渠道的销量和星期和退货率



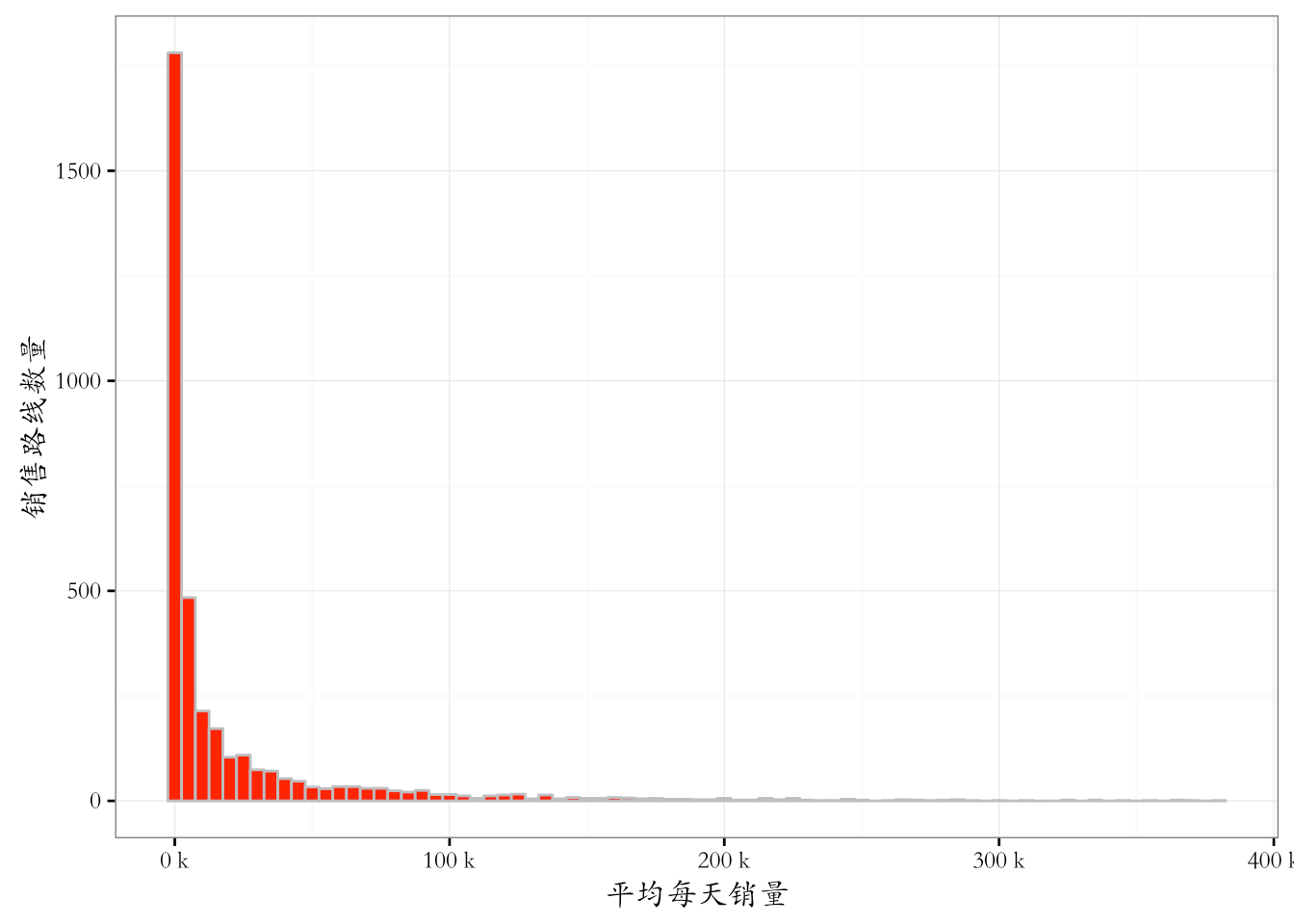
分析：该图呈现了不同的销售渠道每个星期所对应的销量，可见每种销售渠道每个星期的销量都维持在一个相对稳定的水平上；然而不同渠道的销量差距较大，渠道1销量最高，基本在5千万以上，渠道9销量最低，基本在4万左右。但是销售渠道5、8退回率都较高，特别是销售渠道5在第九周的退货率最高。

6、销售渠道和销售站分析  
销售渠道有多少销售站可视化



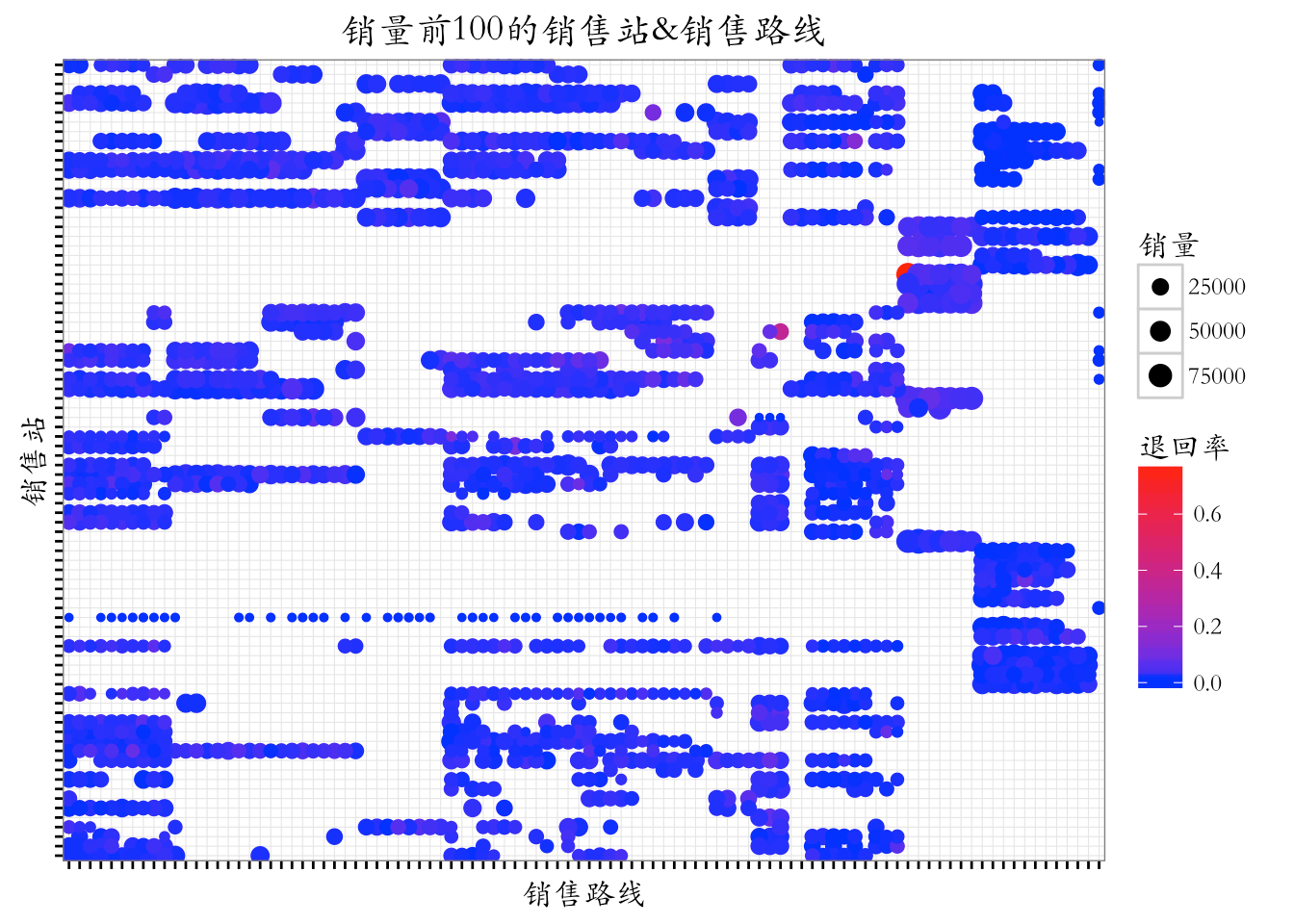
分析：由图可知，将近400个销售站只有1条销售渠道，只有不到100个销售站有超过三条的销售渠道。

7、销售路线的分析



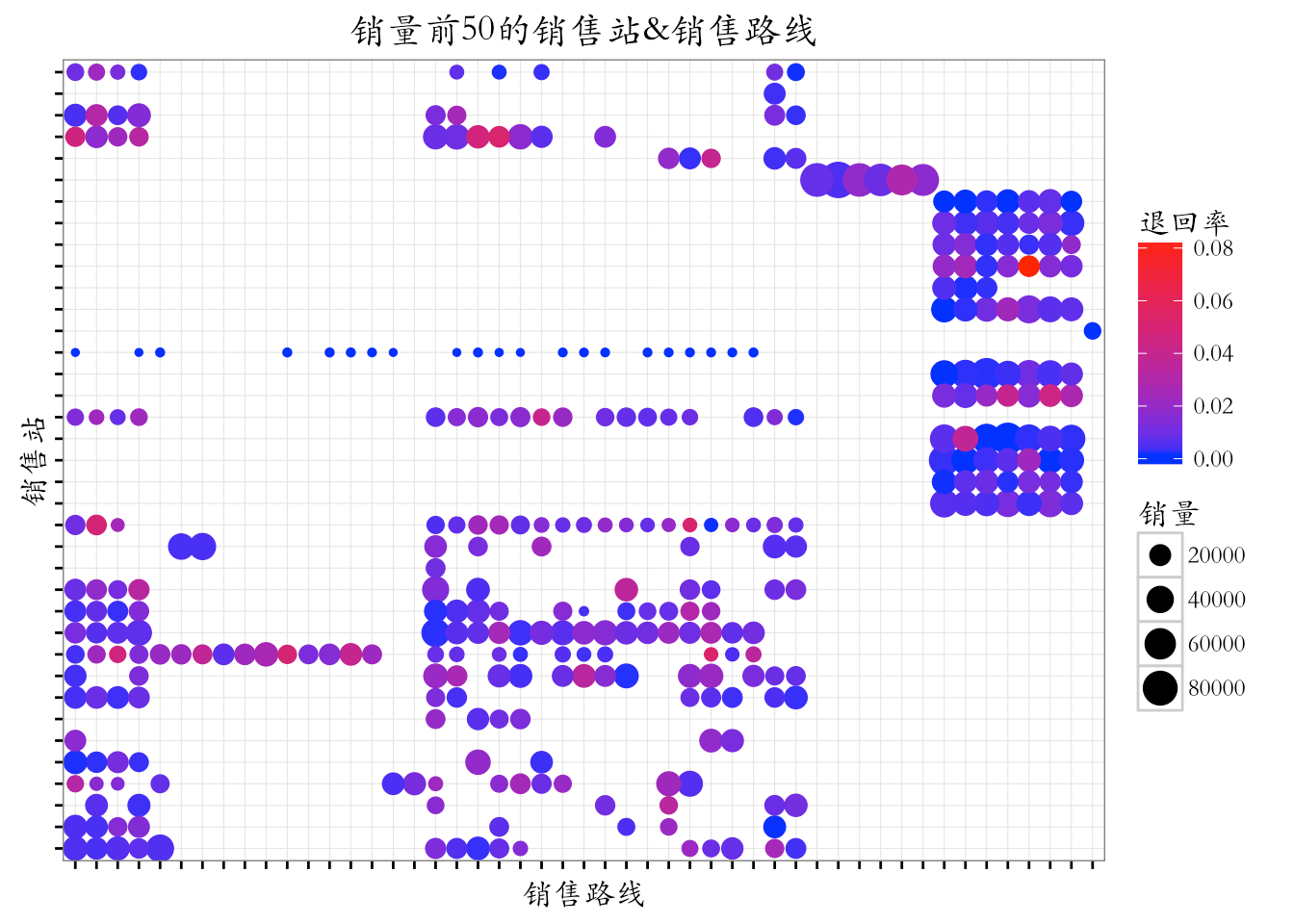
分析：从平均每天销量和销售路线数量的条形图可以看出，大部分销售路线的平均销售量并不多，超过2/3的销售路线每天的销售量不超过10000，且随着销量的增多销售路线的数量有递减的趋势。  
8、销售路线和销售站

销量前100的销售站&销售路线：



分析：图中每一个点都是对应的销售路线与销售站的销量，其中点的大小表示销量的多少，点分布越多的地方销量越高；点的颜色表示退回的百分比，由红到蓝退回百分比递减。销量前100的点的分布较为均匀，无法得出体现其中内在规律的结论。

销量前50的销售站&销售路线：

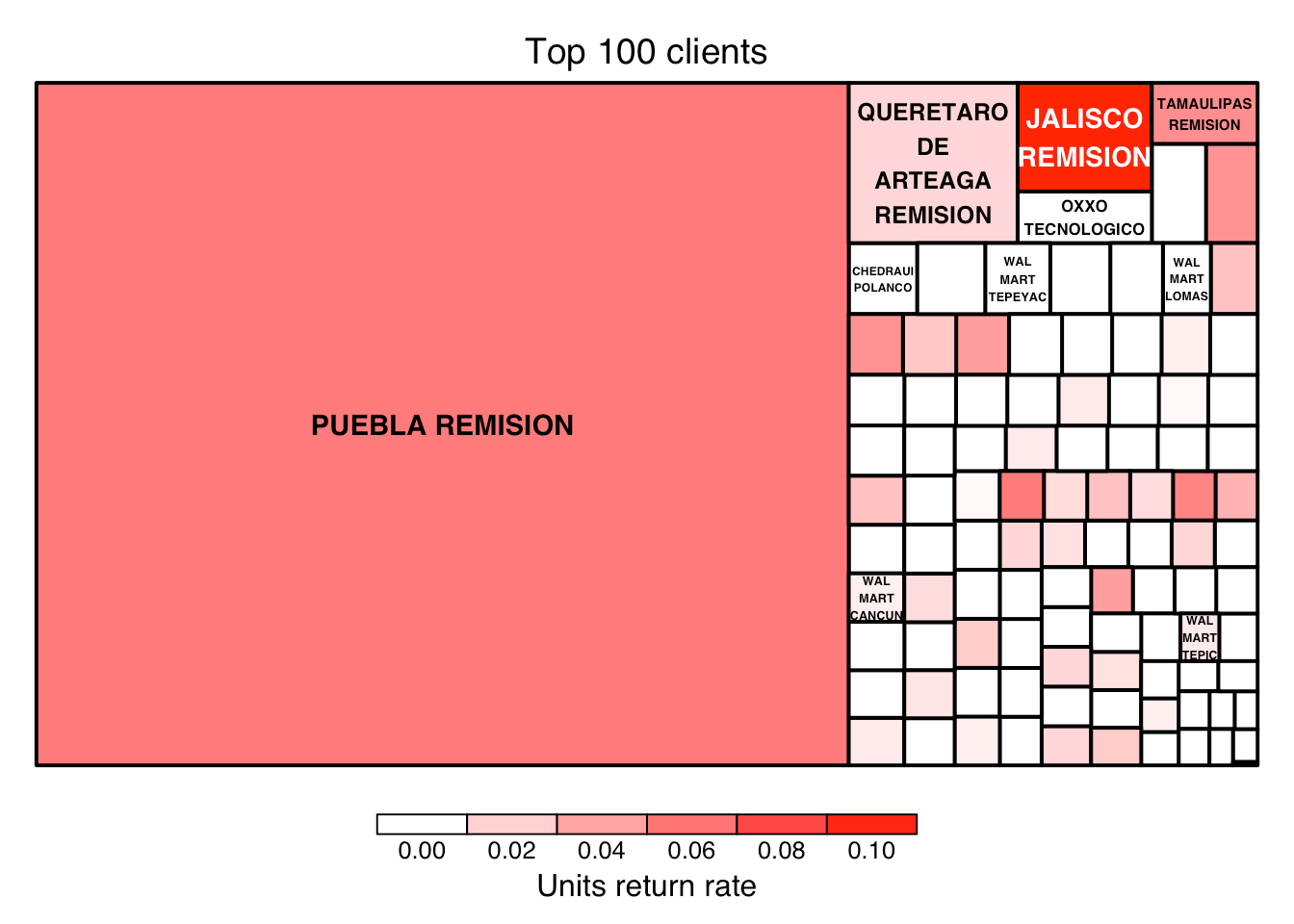


分析：从图中可以看出，在图的右上侧点的 分布较为密集，点较大，颜色偏蓝，说明销售量大相应的退回百分比较小。

9、对客户数据进行分析

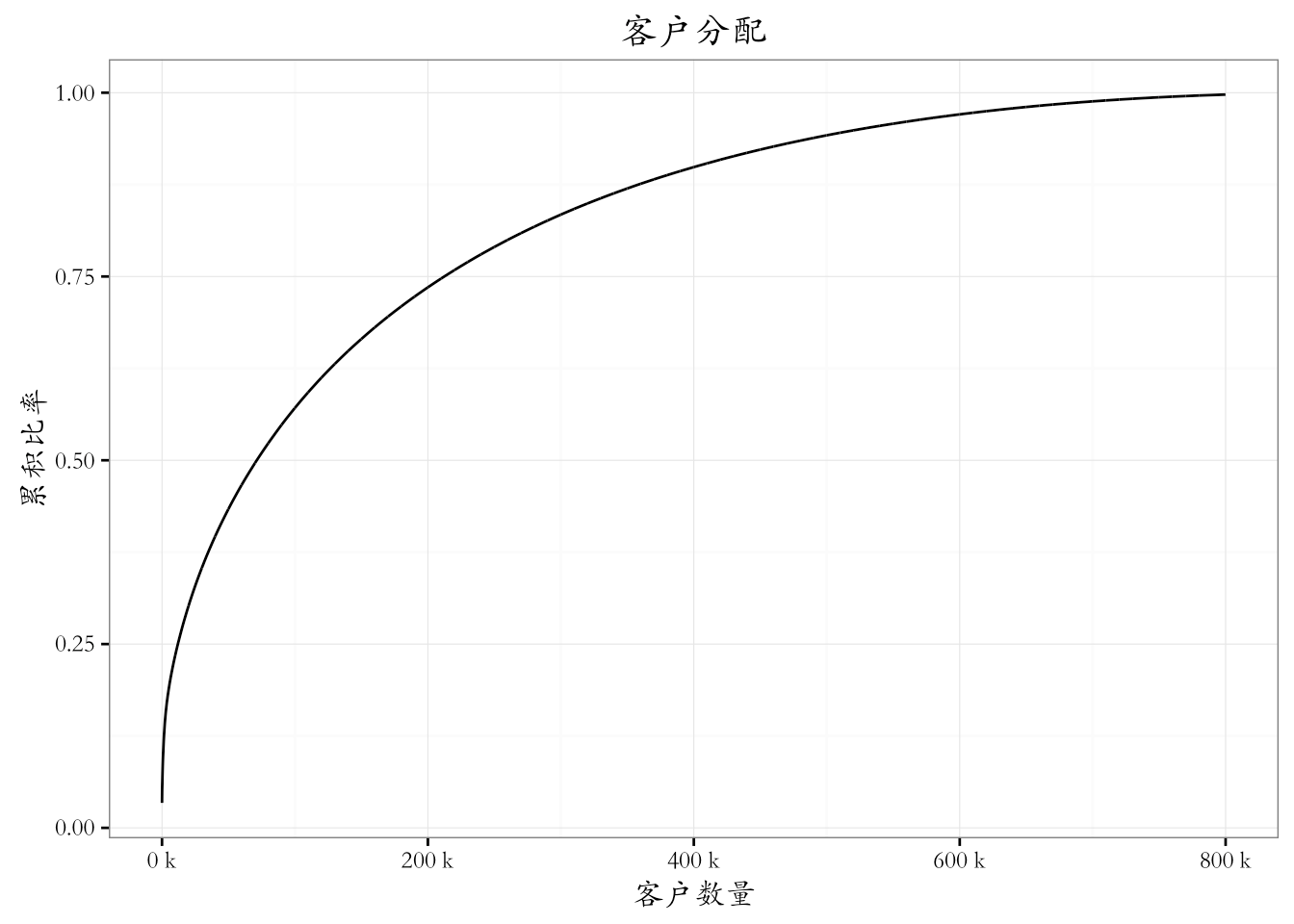
一共有885416个客户。

# 花费量前100个客户的树形图



分析：在树形图中面积最大的是Puebla Remision，可见他是一个大客户，但是退货率也偏高。但其他客户与Puebla Remision花费相差较大，说明该店客户源较单一，应当多挖掘一些新的客户。

10、客户的累积消耗量



分析：曲线随着客户数量的增多逐渐趋于平缓，销量的累积百分比逐渐趋于1，可以看出前20万客户贡献了约75％的销售量。  
11、客户和销售站分析

## Cliente\_ID n\_agencias NombreCliente

## 1 26 2 BODEGA COMERCIAL MEXICANA TOLUCA

## 2 60 2 SAMS CLUB TOLUCA

## 3 65 2 WAL MART METEPEC

## 4 101 1 WAL MART TOLUCA

## 5 105 1 SUPER KOMPRAS SAN BUENAVENTURA

## 6 106 1 ISSSTE 21

分析：客户ID信息其对应的销售站数据集的前6条有3个客户通过2个销售站购买产品，另外3个客户则通过1个销售站购买产品。

dim(agencias\_by\_client)

## [1] 885416 3

分析：一共有885416个销售站信息。

单个客户使用销售站的数量

table(agencias\_by\_client$n\_agencias)

##

## 1 2 3 4 5 9 62

## 844113 37510 3771 19 1 1 1

分析：大部分的客户只从一个销售站购买产品，只有几个客户通过5个以上的销售站购买产品。

使用销售站多的客户信息

agencias\_by\_client %>% filter(n\_agencias %in% c(5, 9, 62)) #返回符合条件的行

## Source: local data table [3 x 3]

##

## Cliente\_ID n\_agencias NombreCliente

## (int) (int) (chr)

## 1 188391 9 DESAYUNOS ESCOLARES

## 2 653378 62 PUEBLA REMISION

## 3 1274327 5 COMERCIALIZADORA LA PUERTA DEL SOL

分析：客户DESAYUNOS ESCOLARES、PUEBLA REMISION、COMERCIALIZADORA LA PUERTA DEL SOL分别通过9、62、5个销售站购买产品。

12、客户和购买渠道分析  
（1）table(clients\_canals$n\_canals)

# 1 2 3 4   
# 874022 6516 65 1   
分析：大多数客户只有一个购买渠道。不同的销售渠道可以为一个客户提供服务。  
（2）table(clients\_agencies\_canals$n\_canals)  
## 1 2 3   
## 922108 3271 3

分析：很少有销售站有同一个客户通过多个渠道。

13、客户和路线分析

## Cliente\_ID n\_routes

## 1 15766 1

## 2 22926 2

## 3 24080 1

## 4 24695 1

## 5 50379 1

## 6 50395 1

分析：客户ID信息其对应的销售路线数据集的前6条有5个客户通过1个路线购买产品，只有1个客户通过2条路线购买产品。

dim(clients\_routes)

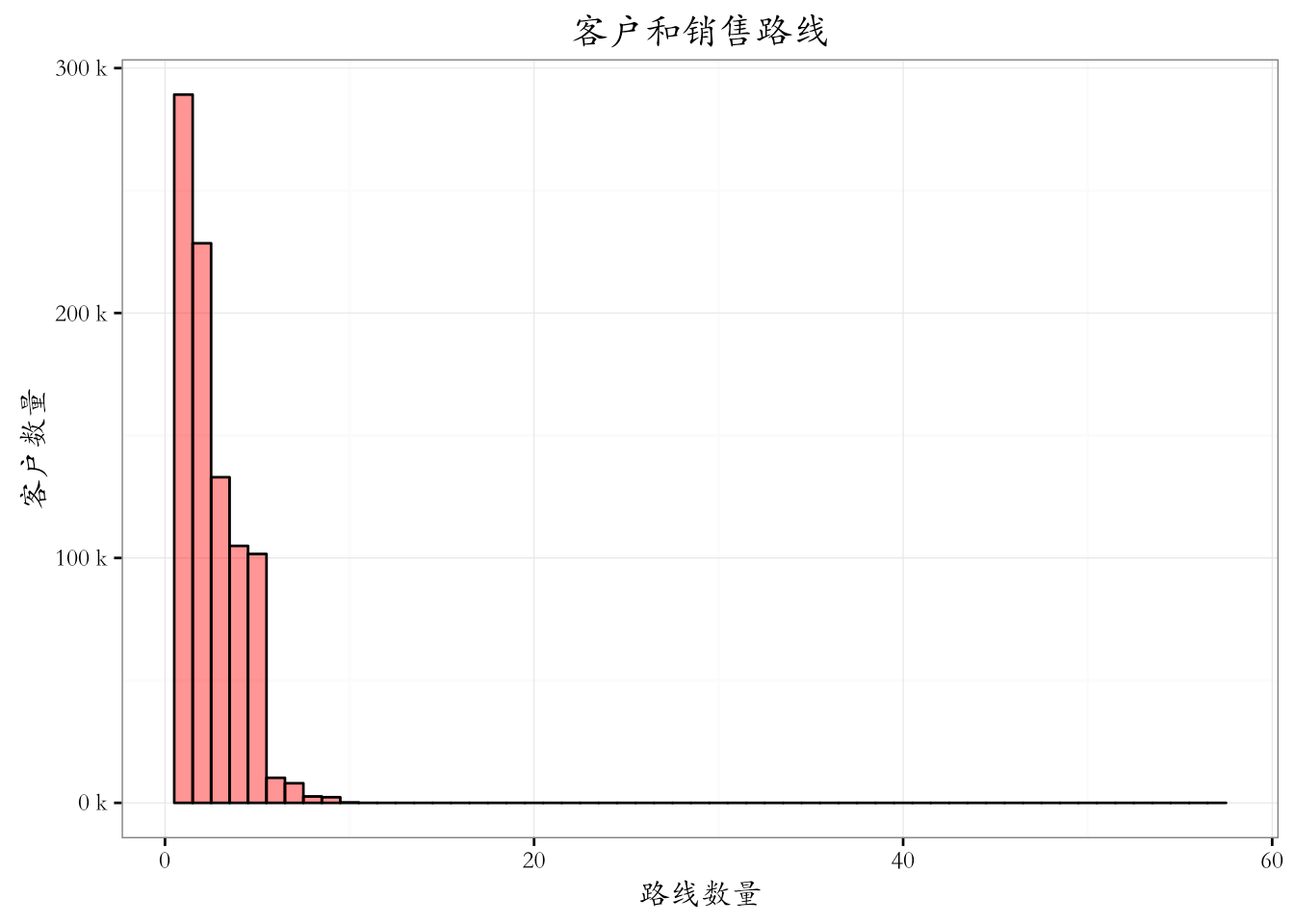
## [1] 880604 2

分析：一共有880604条购买产品的路线。

sum(clients\_routes$n\_routes >= 10)

## [1] 242

## 大多数客户只有不到5个仓库的交货，但超过240个客户的工作与10个仓库或更多。



分析：大部分客户选择的销售路线数量不超过10条，5条路线以下的客户数量较多，随着路线的增多客户数量递减。

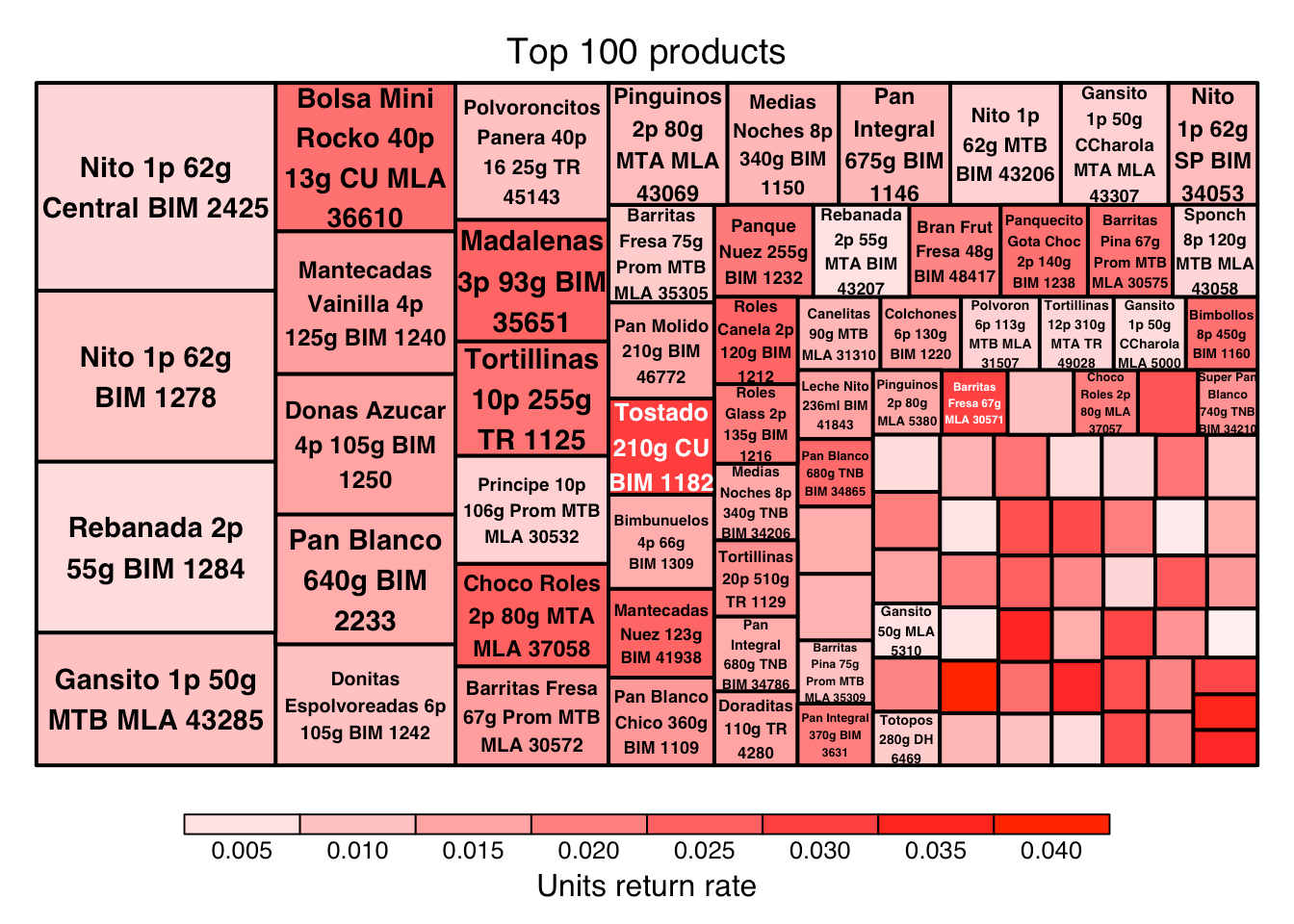
1. 对集团销售的产品进行分析

dim(products) # 产品数量

## [1] 1719 9

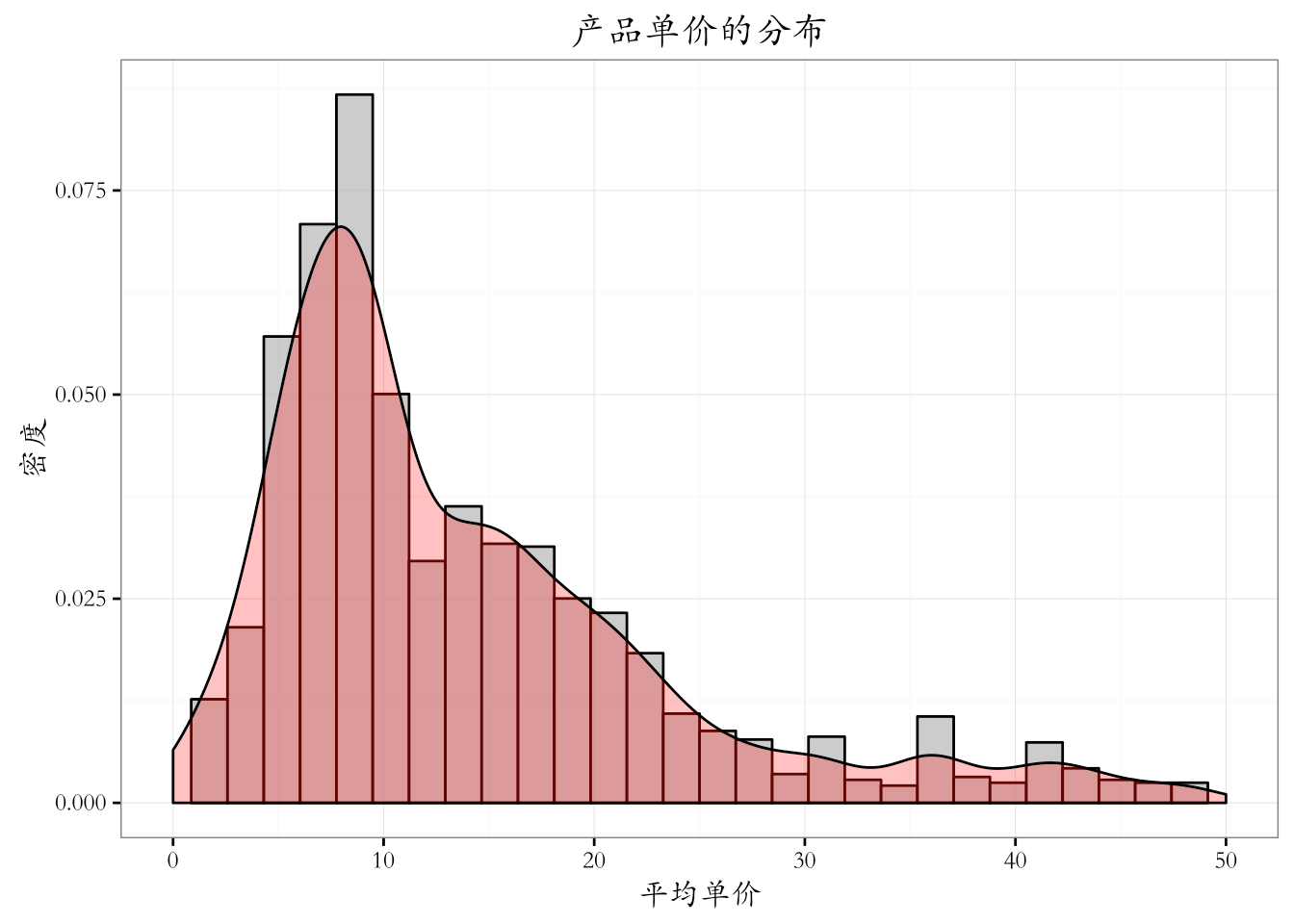
一共销售1719种产品。

销量前100的产品树形图



分析：销量较多的产品前三名是Nito 1p 62g Central BIM 2425，Rebanada 2p 55g BIM 1284，Nito 1p 62g BIM 1278，且退回率都基本在1.5%左右。产品Bolsa Minl Rocko 40p 13g CU MLA 36610的销量高，然而退回率也较高，说明该产品应做相应的改进，减少退货率 ，有更好的提升空间。

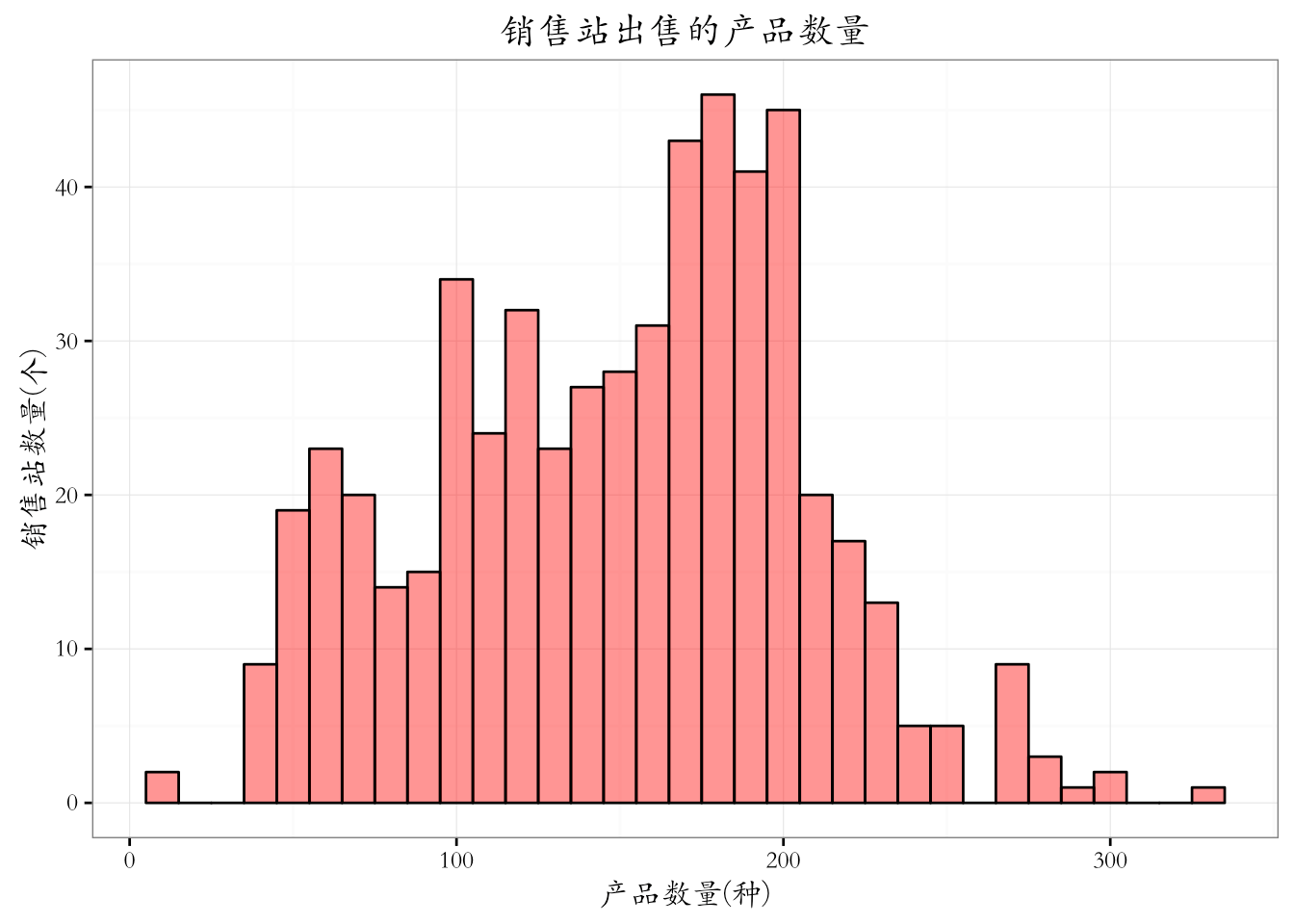
15、产品单价的密度分布



分析：每种产品的单价=该产品的营业额/该产品的销量，条形图和和密度曲线图可以看出平均单价主要集中在[0,20]之内，其中单价为8的产品销量最多。

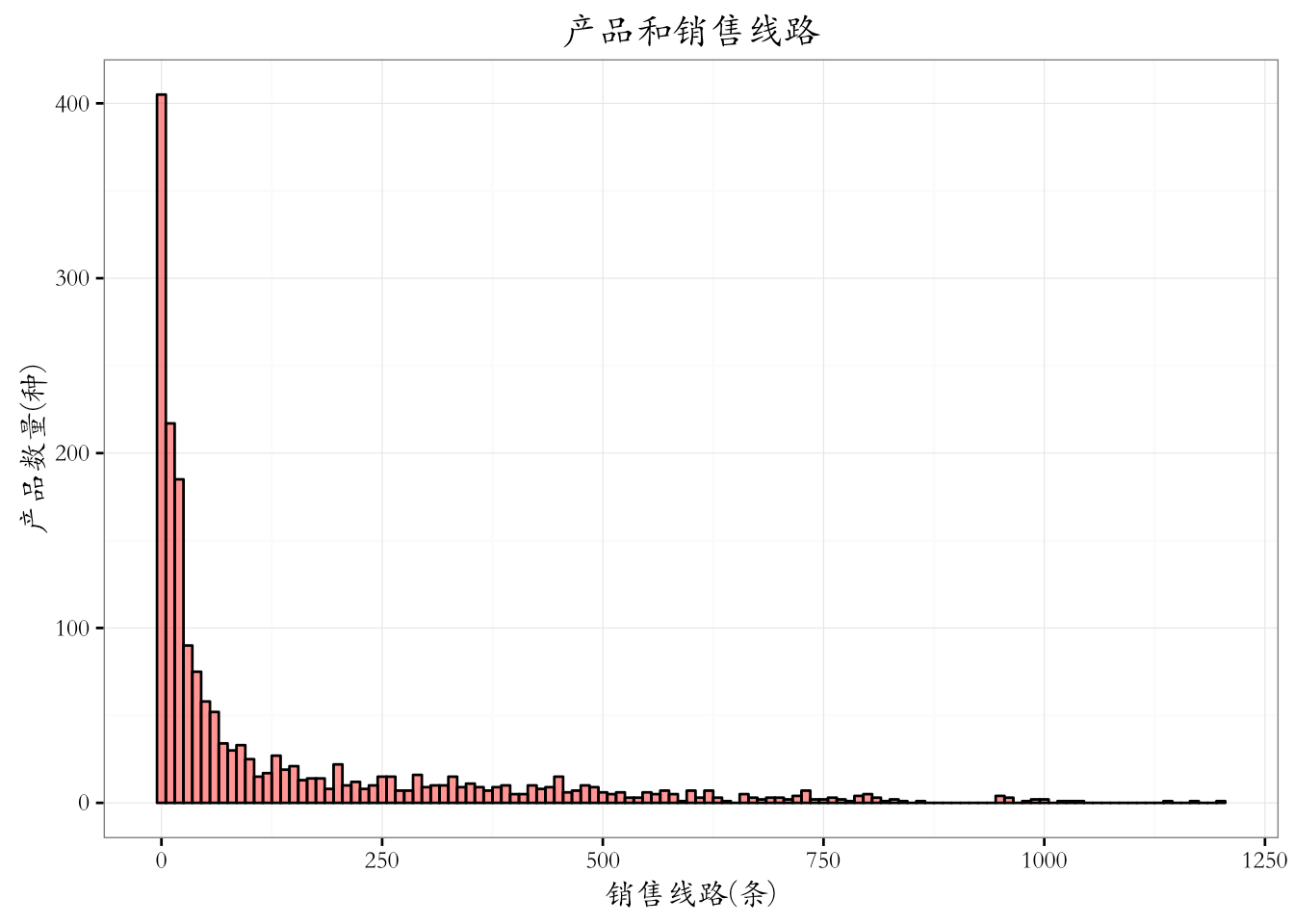
16、产品和销售站

dim(products\_agencies)

## [1] 552 2  


分析：大部分的销售站产品数量都在100~200之内。

17、产品和销售路线

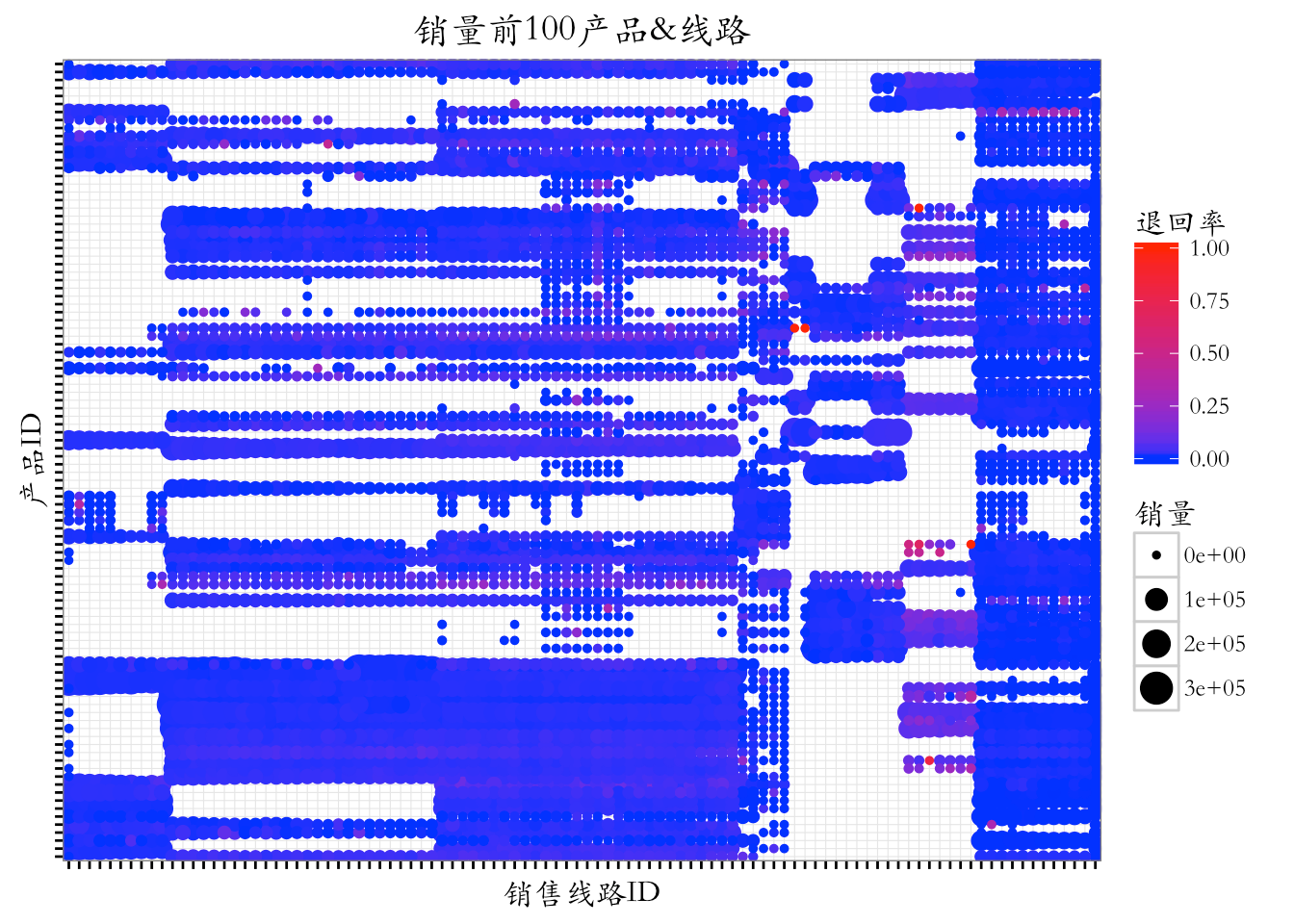


分析：大部分产品只有几条销售路线，只有几种产品的销售路线很多

产品和销售路线 top100

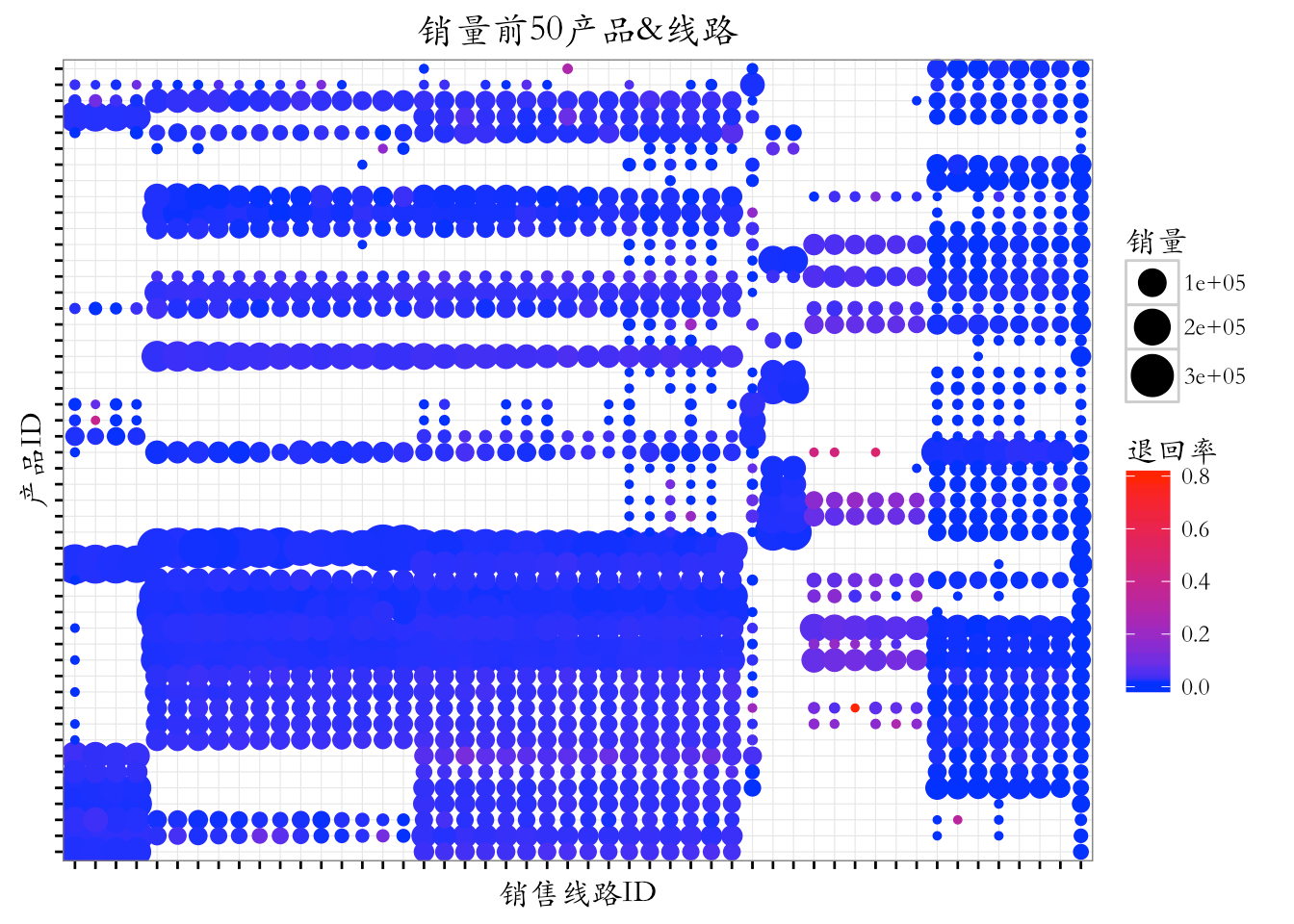
dim(routes.products)

## [1] 250265 8



分析:图中每一个点都是对应的销售路线与产品ID的销量，其中点的大小表示销量的多少，点分布越多的地方销量越高；点的颜色表示退回的百分比，由红到蓝退回百分比递减。销量前100的点的分布较为均匀，无法得出体现其中内在规律的结论。继续分析产品和销售路线前50的销量及返回率。

产品和销售路线 top50

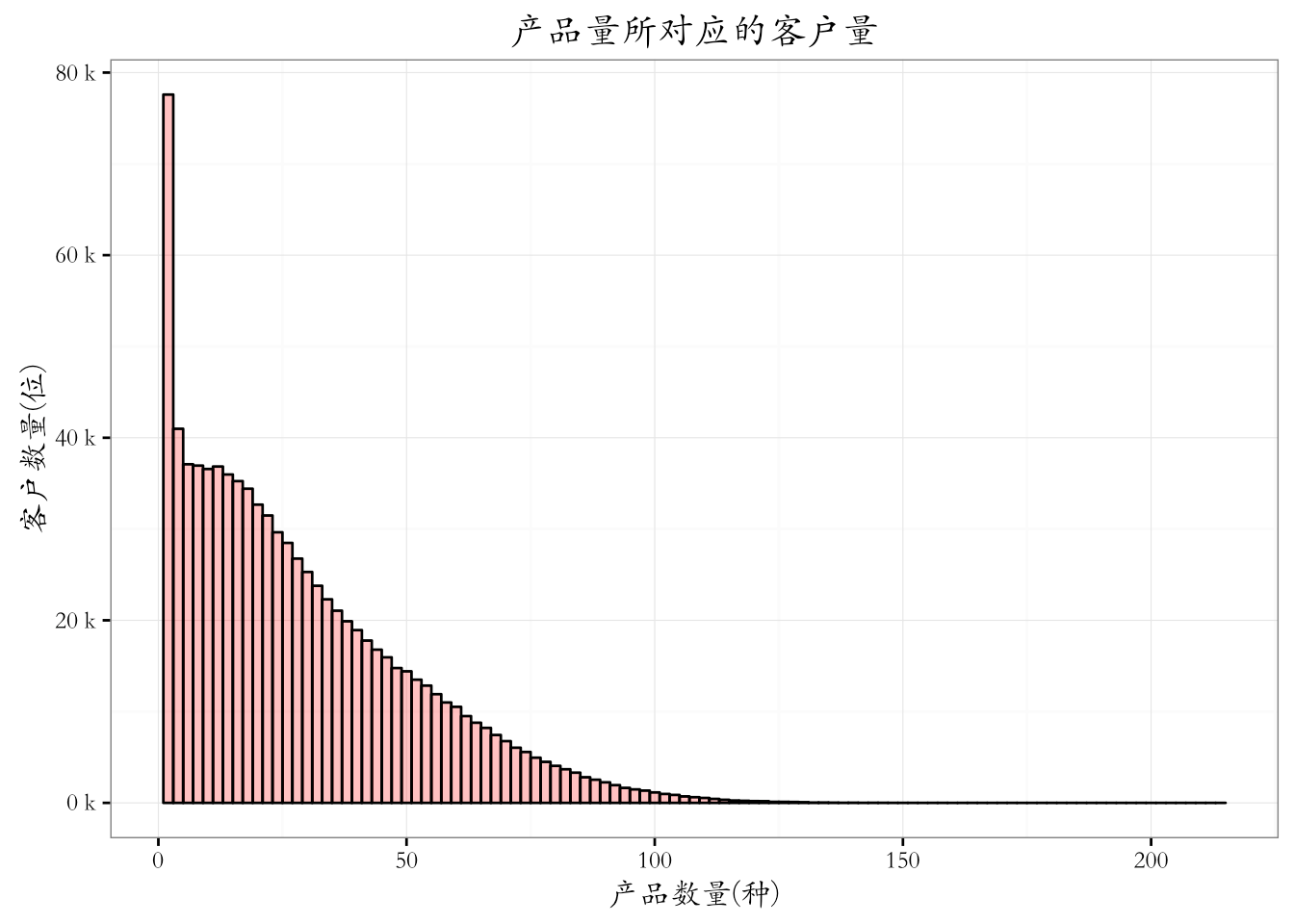


分析：从图中可以看出，在图的左下侧点的分布较为密集，点较大，颜色偏蓝，说明销售量大相应的退回百分比较小。与销量前50的销售站&销售路线分析的结果一致。

18、产品和客户

dim(products\_by\_client)

## [1] 885416 3



分析：从图中可以看出，随着产品数量的增多，客户量逐渐减少。产品数量小于10时客户数量最多。汪嫂得出结论，物以稀为贵。

## 需求量的数据分布####

## Producto\_ID Semana Agencia\_ID Canal\_ID Ruta\_SAK Demanda logDemanda

## 1 1212 3 1110 7 3301 55 4.290459

## 2 1216 3 1110 7 3301 27 3.610918

## 3 1238 3 1110 7 3301 49 4.127134

## 4 1240 3 1110 7 3301 109 4.867534

## 5 1242 3 1110 7 3301 37 3.912023

## 6 1250 3 1110 7 3301 165 5.225747

dim(Demanda\_uni\_equil)

## [1] 5363216 7

summary(Demanda\_uni\_equil$Demanda)

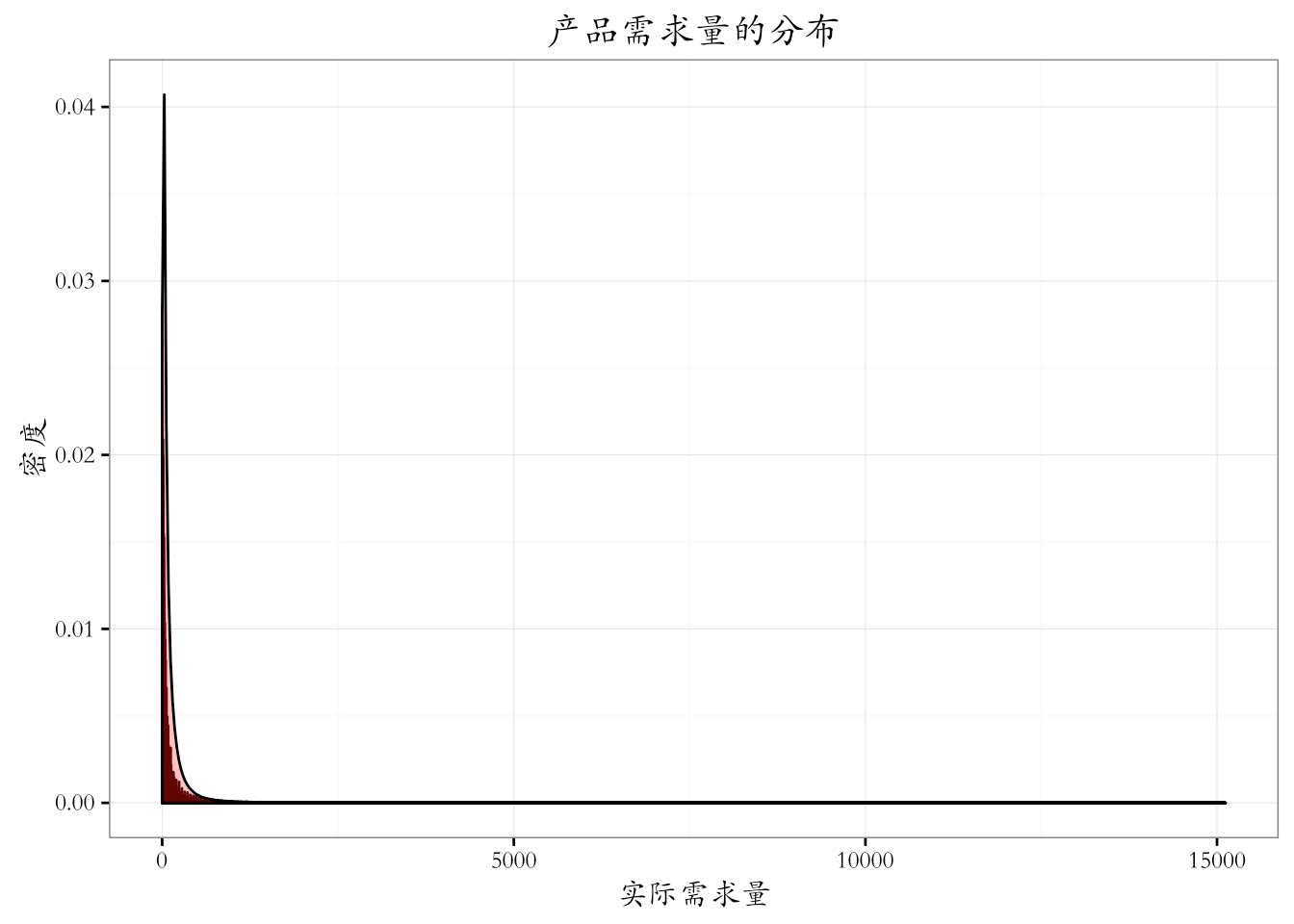
## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 0.00 14.00 40.00 99.93 104.00 15120.00

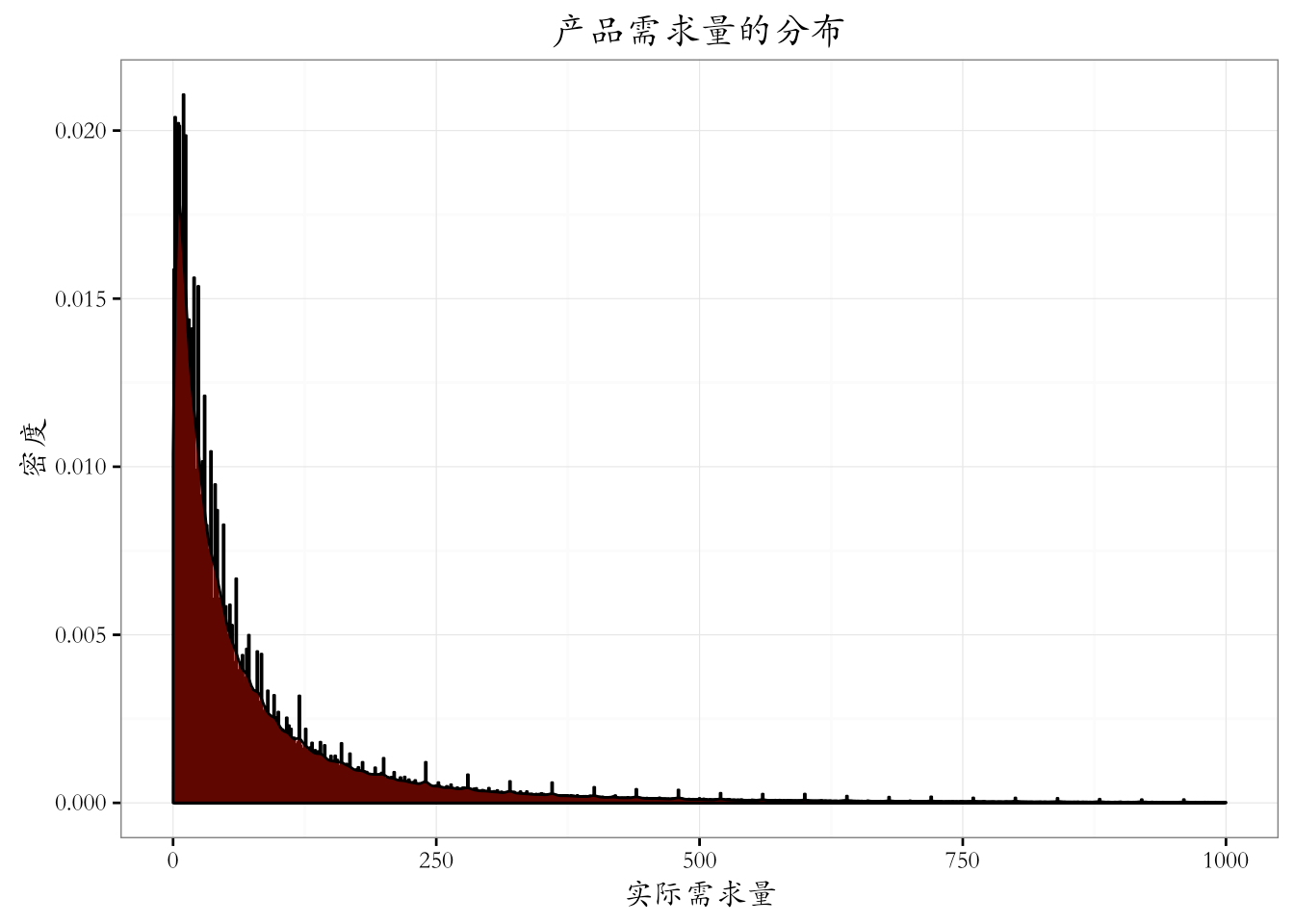
summary(Demanda\_uni\_equil$logDemanda)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

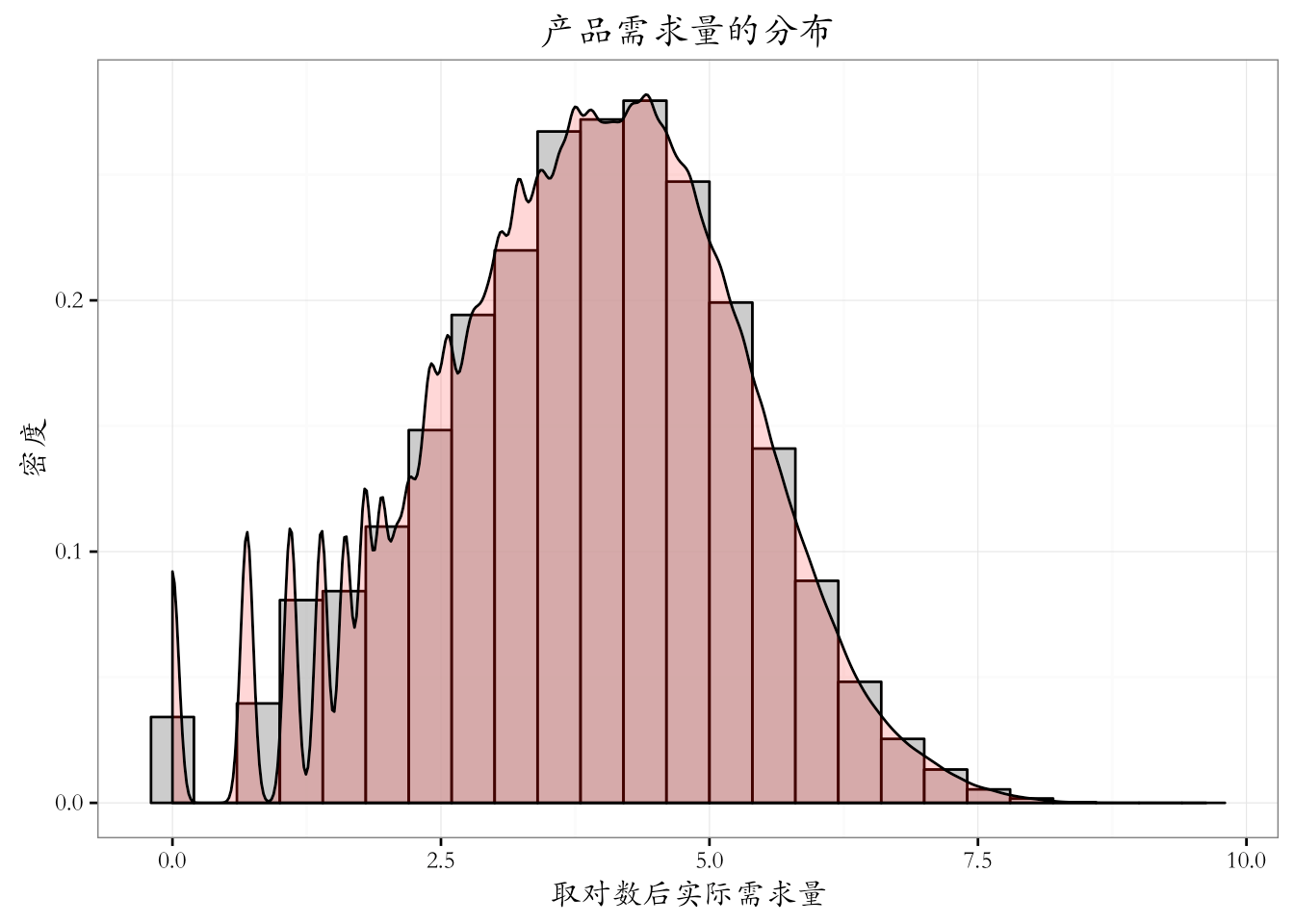
## 0.000 2.890 3.892 3.824 4.828 9.624



分析：上图为实际需求量与其密度的条形图，可以看出大部分都集中在[0,2500]区间内，趋势不明显，因此考虑将需求量的区间扩大。



分析：将区间扩大后，可以发现产品实际需求量随着数量的增多，密度有逐渐下降的趋势，在需求量为大于500时密度趋于0。说明市场对产品的需求量大部分在[0,250]之间，然而对于分析其内在规律仍不太合适，因此考虑将其取对数。



分析：在取对数后实际需求量的分布近似服从正态分布，可以更直观的反映出需求量的分布趋势，可以看出在取对数后的实际需求量为4.5时密度达到最大。

## 2016年暑期课程设计####

## 问题：Grupo Bimbo Inventory Demand

## 宾堡集团的库存需求

## 最大限度地提高销售和最大限度地减少烘焙食品的退回

## Daitu

## start:2016.06.22

## 参考借鉴kaggle上的公开程序

## **使用梯度提升机进行预测**

##设置工作文件夹

setwd("/Users/Daitu/数据分析/kaggle/Grupo Bimbo")

getwd()

## [1] "/Users/daitu/数据分析/kaggle/Grupo Bimbo"

## 设置集群 ####

print(paste("Set up Cluster",Sys.time()))

## [1] "Set up Cluster 2016-06-23 10:51:40"

library(h2o) # R API is just a library

## Warning: package 'h2o' was built under R version 3.2.5

## Loading required package: statmod

##

## ----------------------------------------------------------------------

##

## Your next step is to start H2O:

## > h2o.init()

##

## For H2O package documentation, ask for help:

## > ??h2o

##

## After starting H2O, you can use the Web UI at http://localhost:54321

## For more information visit http://docs.h2o.ai

##

## ----------------------------------------------------------------------

##

## Attaching package: 'h2o'

## The following objects are masked from 'package:stats':

##

## sd, var

## The following objects are masked from 'package:base':

##

## &&, %\*%, %in%, ||, apply, as.factor, as.numeric, colnames,

## colnames<-, ifelse, is.character, is.factor, is.numeric, log,

## log10, log1p, log2, round, signif, trunc

## 启动一个集群; 定一位4核同时计算；

h2o.init(nthreads=4,max\_mem\_size='12G')

## Connection successful!

##

## R is connected to the H2O cluster:

## H2O cluster uptime: 15 minutes 7 seconds

## H2O cluster version: 3.8.2.6

## H2O cluster name: H2O\_started\_from\_R\_daitu\_agg862

## H2O cluster total nodes: 1

## H2O cluster total memory: 8.26 GB

## H2O cluster total cores: 8

## H2O cluster allowed cores: 4

## H2O cluster healthy: TRUE

## H2O Connection ip: localhost

## H2O Connection port: 54321

## H2O Connection proxy: NA

## R Version: R version 3.2.3 (2015-12-10)

## 加载数据####

print(paste("加载数据",Sys.time()))

## [1] "加载数据 2016-06-23 10:51:40"

## 读取整个训练数据，使用所有的核

system.time({

train<-h2o.uploadFile("train.csv",destination\_frame = "train.hex")

})

##

|

| | 0%

|

|=================================================================| 100%

## user system elapsed

## 0.507 3.752 34.486

train[1:5,] ## 查看训练集的前几行

## Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID Venta\_uni\_hoy

## 1 3 1110 7 3301 15766 1212 3

## 2 3 1110 7 3301 15766 1216 4

## 3 3 1110 7 3301 15766 1238 4

## 4 3 1110 7 3301 15766 1240 4

## 5 3 1110 7 3301 15766 1242 3

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil

## 1 25.14 0 0 3

## 2 33.52 0 0 4

## 3 39.32 0 0 4

## 4 33.52 0 0 4

## 5 22.92 0 0 3

##

## [5 rows x 11 columns]

## 将训练集的(预测目标+1)取对数

train$target<-log(train$Demanda\_uni\_equil+1)

train[1:5,]

## Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID Venta\_uni\_hoy

## 1 3 1110 7 3301 15766 1212 3

## 2 3 1110 7 3301 15766 1216 4

## 3 3 1110 7 3301 15766 1238 4

## 4 3 1110 7 3301 15766 1240 4

## 5 3 1110 7 3301 15766 1242 3

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 25.14 0 0 3 1.386294

## 2 33.52 0 0 4 1.609438

## 3 39.32 0 0 4 1.609438

## 4 33.52 0 0 4 1.609438

## 5 22.92 0 0 3 1.386294

##

## [5 rows x 12 columns]

h2o.median(train$target)

## [1] 1.386294

## 数据分区

print(paste("数据分区",Sys.time()))

## [1] "数据分区 2016-06-23 10:52:21"

## 这个模型将会把数据分为3个部分，根据星期数据进行分区：

## one to generate product averages, a second to fit a model, and a third to evaluate the model

## 第一个数据用来生成产品均值，第二部分数据用来拟合一个模型，第三部分数据用来计算模型

dev<-train[train$Semana <= 5,] ## gets Semana 3,4,5

val<-train[train$Semana > 5 & train$Semana <= 8,] ## gets Semana 6, 7,8

val[1:5,]

## Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID Venta\_uni\_hoy

## 1 6 1110 7 3301 15766 1216 1

## 2 6 1110 7 3301 15766 1238 2

## 3 6 1110 7 3301 15766 1242 3

## 4 6 1110 7 3301 15766 1250 1

## 5 6 1110 7 3301 15766 1309 6

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 8.38 0 0 1 0.6931472

## 2 19.66 0 0 2 1.0986123

## 3 22.92 0 0 3 1.3862944

## 4 7.64 0 0 1 0.6931472

## 5 40.56 0 0 6 1.9459101

##

## [5 rows x 12 columns]

final<-train[train$Semana == 9,] ## gets Semana 9

final[1:5,]

## Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID Venta\_uni\_hoy

## 1 9 1110 7 3301 15766 1212 1

## 2 9 1110 7 3301 15766 1238 2

## 3 9 1110 7 3301 15766 1240 2

## 4 9 1110 7 3301 15766 1242 1

## 5 9 1110 7 3301 15766 1250 10

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 8.38 0 0 1 0.6931472

## 2 19.66 0 0 2 1.0986123

## 3 16.76 0 0 2 1.0986123

## 4 7.64 0 0 1 0.6931472

## 5 76.40 0 0 10 2.3978953

##

## [5 rows x 12 columns]

## 模型：产品分组&GBM####

print(paste("Model: Product Groups & GBM",Sys.time()))

## [1] "Model: Product Groups & GBM 2016-06-23 10:52:31"

## 使用测试集中用来预测的字段变量进行预测，剔除ID和星期，

predictors<-c("Agencia\_ID","Canal\_ID","Ruta\_SAK","Cliente\_ID","Producto\_ID")

## first part of model: use product averages, created on the dev set

## this is the only time we will use the dev set

## 模型的第一部分：使用产品的均值，在dev数据集上创建

## 这是dev数据集的唯一的一次使用

groups<-h2o.group\_by(data=dev,by="Producto\_ID",mean("target"))

groups[1:5,]

## Producto\_ID mean\_target

## 1 41 4.357809

## 2 53 5.852552

## 3 72 1.651355

## 4 73 1.102258

## 5 100 1.427448

##

## [5 rows x 2 columns]

h2o.median(groups$mean\_target)

## [1] 1.865737

## apply groups back into dev and validation data sets as "mean\_target"

## if there are NAs for this (new products), use a constant; used median of entire train target

## 使用分组后的数据集dev，生成新的确认数据（val）

## 如果数据集中有NAS（代表新的产品），使用中位数进行代替。

newVal<-h2o.merge(x=val,y=groups,all.x = T)

newVal[1:5,]

## Producto\_ID Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Semana Venta\_uni\_hoy

## 1 1216 1110 7 3301 15766 6 1

## 2 1238 1110 7 3301 15766 6 2

## 3 1242 1110 7 3301 15766 6 3

## 4 1250 1110 7 3301 15766 6 1

## 5 1309 1110 7 3301 15766 6 6

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 8.38 0 0 1 0.6931472

## 2 19.66 0 0 2 1.0986123

## 3 22.92 0 0 3 1.3862944

## 4 7.64 0 0 1 0.6931472

## 5 40.56 0 0 6 1.9459101

## mean\_target

## 1 1.207841

## 2 1.257216

## 3 1.586583

## 4 1.636534

## 5 1.370717

##

## [5 rows x 13 columns]

newVal$mean\_target[is.na(newVal$mean\_target)]<-h2o.median(groups$mean\_target)

newVal[1:5,]

## Producto\_ID Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Semana Venta\_uni\_hoy

## 1 1216 1110 7 3301 15766 6 1

## 2 1238 1110 7 3301 15766 6 2

## 3 1242 1110 7 3301 15766 6 3

## 4 1250 1110 7 3301 15766 6 1

## 5 1309 1110 7 3301 15766 6 6

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 8.38 0 0 1 0.6931472

## 2 19.66 0 0 2 1.0986123

## 3 22.92 0 0 3 1.3862944

## 4 7.64 0 0 1 0.6931472

## 5 40.56 0 0 6 1.9459101

## mean\_target

## 1 1.207841

## 2 1.257216

## 3 1.586583

## 4 1.636534

## 5 1.370717

##

## [5 rows x 13 columns]

newFinal<-h2o.merge(x=final,y=groups,all.x = T)

newFinal[1:5,]

## Producto\_ID Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Semana Venta\_uni\_hoy

## 1 1212 1110 7 3301 15766 9 1

## 2 1238 1110 7 3301 15766 9 2

## 3 1240 1110 7 3301 15766 9 2

## 4 1242 1110 7 3301 15766 9 1

## 5 1250 1110 7 3301 15766 9 10

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 8.38 0 0 1 0.6931472

## 2 19.66 0 0 2 1.0986123

## 3 16.76 0 0 2 1.0986123

## 4 7.64 0 0 1 0.6931472

## 5 76.40 0 0 10 2.3978953

## mean\_target

## 1 1.179147

## 2 1.257216

## 3 1.623425

## 4 1.586583

## 5 1.636534

##

## [5 rows x 13 columns]

newFinal$mean\_target[is.na(newFinal$mean\_target)]<-h2o.median(groups$mean\_target)

newFinal[1:5,]

## Producto\_ID Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Semana Venta\_uni\_hoy

## 1 1212 1110 7 3301 15766 9 1

## 2 1238 1110 7 3301 15766 9 2

## 3 1240 1110 7 3301 15766 9 2

## 4 1242 1110 7 3301 15766 9 1

## 5 1250 1110 7 3301 15766 9 10

## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil target

## 1 8.38 0 0 1 0.6931472

## 2 19.66 0 0 2 1.0986123

## 3 16.76 0 0 2 1.0986123

## 4 7.64 0 0 1 0.6931472

## 5 76.40 0 0 10 2.3978953

## mean\_target

## 1 1.179147

## 2 1.257216

## 3 1.623425

## 4 1.586583

## 5 1.636534

##

## [5 rows x 13 columns]

## 训练 GBM; 使用参数以保持整体运行时间在20分钟内

## this model is fit on Semana 6 & 7 & 8, and evaluated on Semana 9.

g<-h2o.gbm(

training\_frame = newVal, ## H2O frame holding the training data

validation\_frame = newFinal, ## extra holdout piece for three layer modeling

x=predictors, ## this can be names or column numbers

y="target", ## target: using the logged variable created earlier

model\_id="gbm1", ## internal H2O name for model

ntrees = 25, ## use fewer trees than default (50) to speed up training

learn\_rate = 0.3, ## lower learn\_rate is better, but use high rate to offset few trees

score\_tree\_interval = 3, ## score every 3 trees

sample\_rate = 0.5, ## use half the rows each scoring round

col\_sample\_rate = 0.8, ## use 4/5 the columns to decide each split decision

offset\_column = "mean\_target"

)

##

|

| | 0%

|

|=== | 4%

|

|===== | 8%

|

|======== | 12%

|

|========== | 16%

|

|============= | 20%

|

|================ | 24%

|

|================== | 28%

|

|===================== | 32%

|

|======================= | 36%

|

|========================== | 40%

|

|============================= | 44%

|

|=============================== | 48%

|

|================================== | 52%

|

|==================================== | 56%

|

|======================================= | 60%

|

|========================================== | 64%

|

|============================================ | 68%

|

|=============================================== | 72%

|

|================================================= | 76%

|

|==================================================== | 80%

|

|======================================================= | 84%

|

|========================================================= | 88%

|

|============================================================ | 92%

|

|============================================================== | 96%

|

|=================================================================| 100%

## 查看模型

summary(g)

## Model Details:

## ==============

##

## H2ORegressionModel: gbm

## Model Key: gbm1

## Model Summary:

## number\_of\_trees model\_size\_in\_bytes min\_depth max\_depth mean\_depth

## 1 25 10710 5 5 5.00000

## min\_leaves max\_leaves mean\_leaves

## 1 26 32 31.56000

##

## H2ORegressionMetrics: gbm

## \*\* Reported on training data. \*\*

##

## MSE: 0.4036441

## R2 : 0.4312556

## Mean Residual Deviance : 0.4036441

##

##

## H2ORegressionMetrics: gbm

## \*\* Reported on validation data. \*\*

##

## MSE: 0.408705

## R2 : 0.419607

## Mean Residual Deviance : 0.408705

##

##

##

##

## Scoring History:

## timestamp duration number\_of\_trees training\_MSE

## 1 2016-06-23 10:53:11 0.000 sec 0 0.46541

## 2 2016-06-23 10:53:28 17.580 sec 3 0.42546

## 3 2016-06-23 10:53:47 35.789 sec 6 0.41739

## 4 2016-06-23 10:54:07 56.048 sec 9 0.41320

## 5 2016-06-23 10:54:29 1 min 18.640 sec 12 0.41126

## 6 2016-06-23 10:54:54 1 min 42.858 sec 15 0.40897

## 7 2016-06-23 10:55:19 2 min 8.434 sec 18 0.40759

## 8 2016-06-23 10:55:46 2 min 35.250 sec 21 0.40696

## 9 2016-06-23 10:56:17 3 min 6.513 sec 24 0.40411

## 10 2016-06-23 10:56:40 3 min 29.646 sec 25 0.40364

## training\_deviance validation\_MSE validation\_deviance

## 1 0.46541 0.47392 0.47392

## 2 0.42546 0.43293 0.43293

## 3 0.41739 0.42416 0.42416

## 4 0.41320 0.41954 0.41954

## 5 0.41126 0.41757 0.41757

## 6 0.40897 0.41543 0.41543

## 7 0.40759 0.41407 0.41407

## 8 0.40696 0.41342 0.41342

## 9 0.40411 0.40907 0.40907

## 10 0.40364 0.40871 0.40871

##

## Variable Importances: (Extract with `h2o.varimp`)

## =================================================

##

## Variable Importances:

## variable relative\_importance scaled\_importance percentage

## 1 Canal\_ID 1285062.000000 1.000000 0.519037

## 2 Ruta\_SAK 639263.500000 0.497457 0.258199

## 3 Producto\_ID 330203.656250 0.256955 0.133369

## 4 Cliente\_ID 126865.828125 0.098724 0.051241

## 5 Agencia\_ID 94461.750000 0.073508 0.038153

# 删除不再需要的较大的数据集

h2o.rm(train)

h2o.rm(dev)

h2o.rm(val)

h2o.rm(newVal)

## 进行预测#####

print(paste("Create Predictions",Sys.time()))

## [1] "Create Predictions 2016-06-23 10:57:00"

## 加载测试集

test<-h2o.uploadFile("test.csv",destination\_frame = "test.hex")

##

|

| | 0%

|

|=================================================================| 100%

test[1:5,] ## 查看测试集的前几行数据

## id Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID

## 1 0 11 4037 1 2209 4639078 35305

## 2 1 11 2237 1 1226 4705135 1238

## 3 2 10 2045 1 2831 4549769 32940

## 4 3 11 1227 1 4448 4717855 43066

## 5 4 11 1219 1 1130 966351 1277

##

## [5 rows x 7 columns]

## merge in the offset column, just as with val and final

newTest<-h2o.merge(x=test,y=groups,all.x = T)

newTest[1:5,]

## Producto\_ID Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID id

## 1 35305 11 4037 1 2209 4639078 0

## 2 1238 11 2237 1 1226 4705135 1

## 3 32940 10 2045 1 2831 4549769 2

## 4 43066 11 1227 1 4448 4717855 3

## 5 1277 11 1219 1 1130 966351 4

## mean\_target

## 1 NaN

## 2 1.257216

## 3 1.450861

## 4 1.094048

## 5 NaN

##

## [5 rows x 8 columns]

newTest$mean\_target[is.na(newTest$mean\_target)]<-h2o.median(groups$mean\_target)

newTest[1:5,]

## Producto\_ID Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID id

## 1 35305 11 4037 1 2209 4639078 0

## 2 1238 11 2237 1 1226 4705135 1

## 3 32940 10 2045 1 2831 4549769 2

## 4 43066 11 1227 1 4448 4717855 3

## 5 1277 11 1219 1 1130 966351 4

## mean\_target

## 1 1.865737

## 2 1.257216

## 3 1.450861

## 4 1.094048

## 5 1.865737

##

## [5 rows x 8 columns]

p<-h2o.predict(g,newTest)

##

|

| | 0%

|

|==== | 7%

|

|========= | 13%

|

|============= | 20%

|

|====================== | 33%

|

|========================== | 40%

|

|============================== | 47%

|

|====================================== | 58%

|

|=========================================== | 67%

|

|================================================ | 73%

|

|==================================================== | 80%

|

|========================================================= | 88%

|

|=================================================================| 100%

p<-exp(p)-1

summary(p)

## Warning in summary.H2OFrame(p): Approximated quantiles computed! If you

## are interested in exact quantiles, please pass the `exact\_quantiles=TRUE`

## parameter.

## C1

## Min. : -0.4573

## 1st Qu.: -0.4573

## Median : -0.4573

## Mean : 5.4255

## 3rd Qu.: 3.8110

## Max. :4267.8600

# ## 创建提交文件#####

#

# print(paste("Create Submission",Sys.time()))

# submissionFrame<-h2o.cbind(test$id,p)

# colnames(submissionFrame)<-c("id","Demanda\_uni\_equil")

# h2o.exportFile(submissionFrame,path="h2o\_gbmother.csv") ## 输出文件

**## 对各种方法预测得到的结果进行分析**

setwd("/Users/Daitu/数据分析/kaggle/Grupo Bimbo")

getwd()

## [1] "/Users/daitu/数据分析/kaggle/Grupo Bimbo"

## 加载包

library(data.table)

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.4

library(dplyr)

##

## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':

##

## between, last

## The following objects are masked from 'package:stats':

##

## filter, lag

## The following objects are masked from 'package:base':

##

## intersect, setdiff, setequal, union

library(treemap)

## Warning: package 'treemap' was built under R version 3.2.4

## 读取数据####

## 1:读取训练集

system.time({

traindata <- fread("train.csv",sep=",",header = TRUE)

})

##

Read 0.0% of 74180464 rows

Read 4.0% of 74180464 rows

Read 8.0% of 74180464 rows

Read 12.1% of 74180464 rows

Read 16.4% of 74180464 rows

Read 20.6% of 74180464 rows

Read 24.6% of 74180464 rows

Read 28.7% of 74180464 rows

Read 32.7% of 74180464 rows

Read 36.8% of 74180464 rows

Read 40.7% of 74180464 rows

Read 44.7% of 74180464 rows

Read 48.7% of 74180464 rows

Read 52.5% of 74180464 rows

Read 56.5% of 74180464 rows

Read 60.2% of 74180464 rows

Read 64.4% of 74180464 rows

Read 68.4% of 74180464 rows

Read 72.5% of 74180464 rows

Read 76.3% of 74180464 rows

Read 80.1% of 74180464 rows

Read 83.8% of 74180464 rows

Read 87.6% of 74180464 rows

Read 91.4% of 74180464 rows

Read 95.2% of 74180464 rows

Read 99.1% of 74180464 rows

Read 74180464 rows and 11 (of 11) columns from 2.980 GB file in 00:00:34

## user system elapsed

## 28.899 4.335 37.284

head(traindata)

## Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID

## 1: 3 1110 7 3301 15766 1212

## 2: 3 1110 7 3301 15766 1216

## 3: 3 1110 7 3301 15766 1238

## 4: 3 1110 7 3301 15766 1240

## 5: 3 1110 7 3301 15766 1242

## 6: 3 1110 7 3301 15766 1250

## Venta\_uni\_hoy Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil

## 1: 3 25.14 0 0 3

## 2: 4 33.52 0 0 4

## 3: 4 39.32 0 0 4

## 4: 4 33.52 0 0 4

## 5: 3 22.92 0 0 3

## 6: 5 38.20 0 0 5

system.time({

testdata <- fread("test.csv",sep=",",header = TRUE)

})

## user system elapsed

## 1.681 0.257 2.255

head(testdata)

## id Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID

## 1: 0 11 4037 1 2209 4639078 35305

## 2: 1 11 2237 1 1226 4705135 1238

## 3: 2 10 2045 1 2831 4549769 32940

## 4: 3 11 1227 1 4448 4717855 43066

## 5: 4 11 1219 1 1130 966351 1277

## 6: 5 11 1146 4 6601 1741414 972

## 1:对通过GBM预测得到的结果进行分析####

## 读取预测数据，ntree ＝ 25

h2o\_gbm <- fread("h2o\_gbmother.csv",sep = ",",header = TRUE)

##

Read 53.0% of 6999251 rows

Read 87.4% of 6999251 rows

Read 6999251 rows and 2 (of 2) columns from 0.171 GB file in 00:00:04

# 对 需求量数据向上取整

h2o\_gbm$Demanda\_uni\_equil <- ceiling(h2o\_gbm$Demanda\_uni\_equil)

head(h2o\_gbm)

## id Demanda\_uni\_equil

## 1: 0 5

## 2: 1 2

## 3: 2 3

## 4: 3 2

## 5: 4 8

## 6: 5 3

## 将预测数据与测试数据组合

testdata$Demanda\_uni\_equil <- h2o\_gbm$Demanda\_uni\_equil

## 将数据根据星期分组

trainsemana <- traindata %>%

group\_by(Semana,Canal\_ID) %>%

summarise(Demanda\_uni\_equil = sum(Demanda\_uni\_equil))

testsemana <- testdata %>%

group\_by(Semana,Canal\_ID) %>%

summarise(Demanda\_uni\_equil = sum(Demanda\_uni\_equil))

semana\_canal <- rbind(trainsemana,testsemana)

semana\_canal$Semana <- as.factor(semana\_canal$Semana)

## 可视化查每周的需求量

ggplot(data = semana\_canal,aes(x = Semana,y = Demanda\_uni\_equil)) +

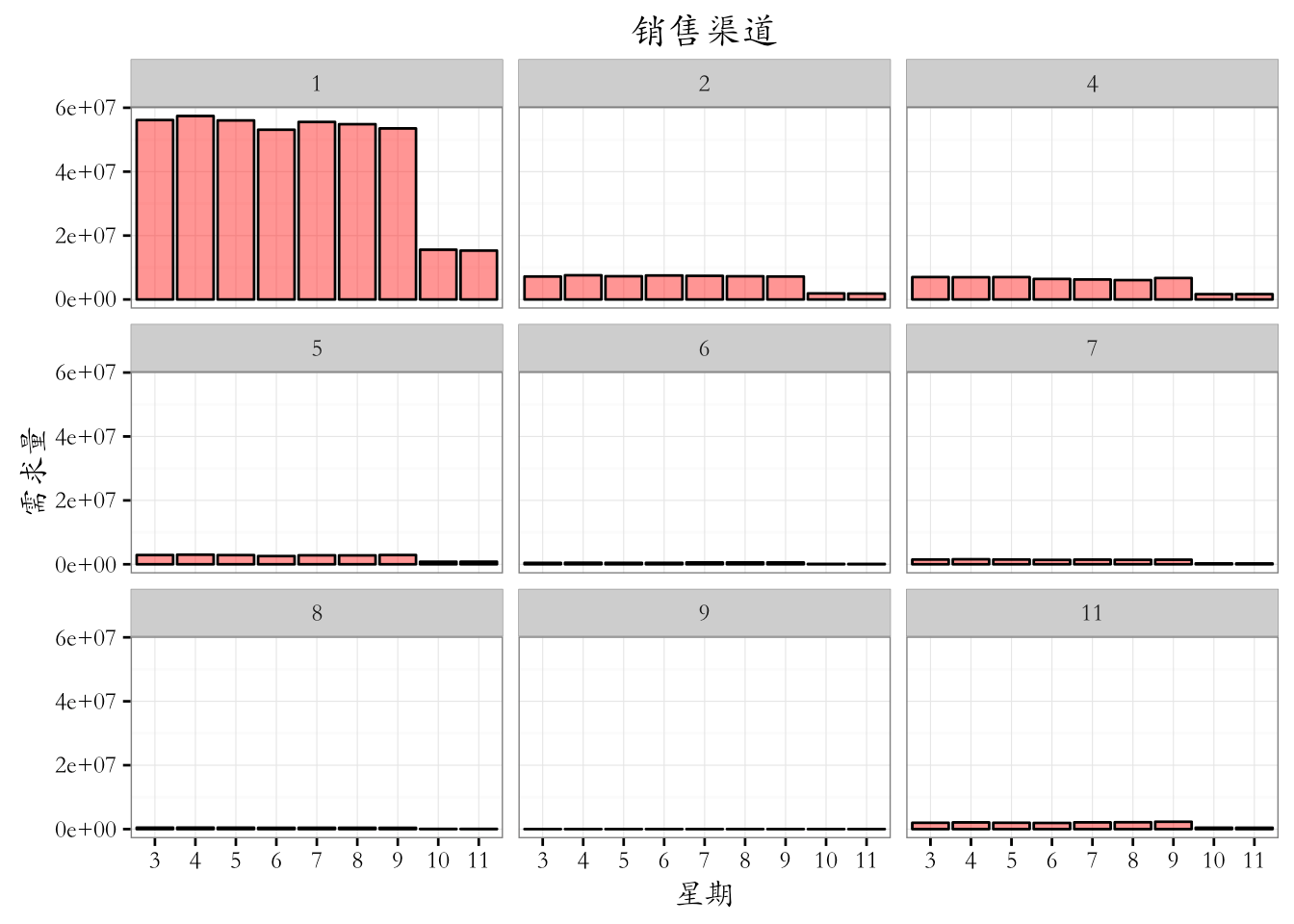
geom\_bar(stat = "identity", fill = "red",color = "black",alpha = 0.5) +

theme\_bw(base\_family = "STKaiti") +

facet\_wrap(~Canal\_ID) +

scale\_x\_discrete() +

labs(x = "星期",y = "需求量",title = "销售渠道")



## 没有办法放在一块对比，因为测试集中的预测数据可能不是所有的数据