R Notebook

data <- read.csv("C:/Users/SCIENCE/Documents/ndata.csv")  
data

## SALES PRICEEACH ORDERNUMBER QUANTITYORDERED  
## 1 2871.00 95.70 10107 30  
## 2 2765.90 81.35 10121 34  
## 3 3884.34 94.74 10134 41  
## 4 3746.70 83.26 10145 45  
## 5 5205.27 100.00 10159 49  
## 6 3479.76 96.66 10168 36  
## 7 2497.77 86.13 10180 29  
## 8 5512.32 100.00 10188 48  
## 9 2168.54 98.57 10201 22  
## 10 4708.44 100.00 10211 41  
## 11 3965.66 100.00 10223 37  
## 12 2333.12 100.00 10237 23  
## 13 3188.64 100.00 10251 28  
## 14 3676.76 100.00 10263 34  
## 15 4177.35 92.83 10275 45  
## 16 4099.68 100.00 10285 36  
## 17 2597.39 100.00 10299 23  
## 18 4394.38 100.00 10309 41  
## 19 4358.04 94.74 10318 46  
## 20 4396.14 100.00 10329 42  
## 21 7737.93 100.00 10341 41  
## 22 1451.00 72.55 10361 20  
## 23 733.11 34.91 10375 21  
## 24 3207.12 76.36 10388 42  
## 25 2434.56 100.00 10403 24  
## 26 7516.08 100.00 10417 66  
## 27 5404.62 100.00 10103 26  
## 28 7209.11 100.00 10112 29  
## 29 7329.06 100.00 10126 38  
## 30 7374.10 100.00 10140 37  
## 31 10993.50 100.00 10150 45  
## 32 4860.24 100.00 10163 21  
## 33 8014.82 100.00 10174 34  
## 34 5372.57 100.00 10183 23  
## 35 7290.36 100.00 10194 42  
## 36 9064.89 100.00 10206 47  
## 37 6075.30 100.00 10215 35  
## 38 6463.23 100.00 10228 29  
## 39 6120.34 100.00 10245 34  
## 40 7680.64 100.00 10258 32  
## 41 4905.39 100.00 10270 21  
## 42 8014.82 100.00 10280 34  
## 43 7136.19 100.00 10291 37  
## 44 10172.70 100.00 10304 47  
## 45 11623.70 100.00 10312 48  
## 46 6000.40 100.00 10322 40  
## 47 3003.00 100.00 10333 26  
## 48 3944.70 100.00 10347 30  
## 49 5691.84 100.00 10357 32  
## 50 4514.92 100.00 10369 41  
## 51 8254.80 100.00 10381 36  
## 52 2416.56 100.00 10391 24  
## 53 4140.23 100.00 10411 23  
## 54 12001.00 100.00 10424 50  
## 55 3896.49 99.91 10107 39  
## 56 2793.86 96.34 10120 29  
## 57 3307.77 100.00 10134 27  
## 58 5192.95 100.00 10145 37  
## 59 5016.83 100.00 10159 37  
## 60 3660.93 100.00 10168 27  
## 61 4695.60 100.00 10180 42  
## 62 3660.92 96.34 10188 38  
## 63 3025.92 100.00 10201 24  
## 64 3009.09 100.00 10210 23  
## 65 5422.39 100.00 10223 47  
## 66 2852.08 100.00 10236 22  
## 67 5756.52 100.00 10251 44  
## 68 4472.00 100.00 10263 40  
## 69 2904.44 100.00 10275 22

df <- subset(data, select=c("SALES","ORDERNUMBER","QUANTITYORDERED", "PRICEEACH"))

library(caTools)

## Warning: package 'caTools' was built under R version 4.3.3

set.seed(2)  
split <- sample.split(df, SplitRatio=0.7)  
train <- subset(df, split="TRUE")

## Warning: In subset.data.frame(df, split = "TRUE") :  
## extra argument 'split' will be disregarded

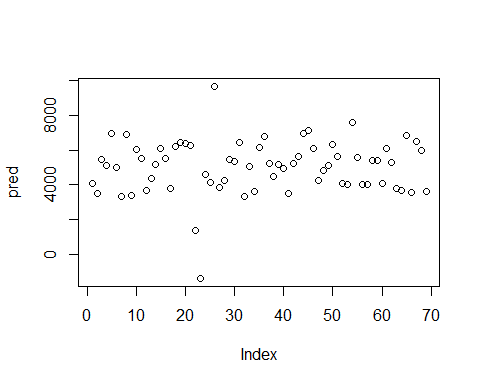
test <- subset(df, split="FALSE")

## Warning: In subset.data.frame(df, split = "FALSE") :  
## extra argument 'split' will be disregarded

train

## SALES ORDERNUMBER QUANTITYORDERED PRICEEACH  
## 1 2871.00 10107 30 95.70  
## 2 2765.90 10121 34 81.35  
## 3 3884.34 10134 41 94.74  
## 4 3746.70 10145 45 83.26  
## 5 5205.27 10159 49 100.00  
## 6 3479.76 10168 36 96.66  
## 7 2497.77 10180 29 86.13  
## 8 5512.32 10188 48 100.00  
## 9 2168.54 10201 22 98.57  
## 10 4708.44 10211 41 100.00  
## 11 3965.66 10223 37 100.00  
## 12 2333.12 10237 23 100.00  
## 13 3188.64 10251 28 100.00  
## 14 3676.76 10263 34 100.00  
## 15 4177.35 10275 45 92.83  
## 16 4099.68 10285 36 100.00  
## 17 2597.39 10299 23 100.00  
## 18 4394.38 10309 41 100.00  
## 19 4358.04 10318 46 94.74  
## 20 4396.14 10329 42 100.00  
## 21 7737.93 10341 41 100.00  
## 22 1451.00 10361 20 72.55  
## 23 733.11 10375 21 34.91  
## 24 3207.12 10388 42 76.36  
## 25 2434.56 10403 24 100.00  
## 26 7516.08 10417 66 100.00  
## 27 5404.62 10103 26 100.00  
## 28 7209.11 10112 29 100.00  
## 29 7329.06 10126 38 100.00  
## 30 7374.10 10140 37 100.00  
## 31 10993.50 10150 45 100.00  
## 32 4860.24 10163 21 100.00  
## 33 8014.82 10174 34 100.00  
## 34 5372.57 10183 23 100.00  
## 35 7290.36 10194 42 100.00  
## 36 9064.89 10206 47 100.00  
## 37 6075.30 10215 35 100.00  
## 38 6463.23 10228 29 100.00  
## 39 6120.34 10245 34 100.00  
## 40 7680.64 10258 32 100.00  
## 41 4905.39 10270 21 100.00  
## 42 8014.82 10280 34 100.00  
## 43 7136.19 10291 37 100.00  
## 44 10172.70 10304 47 100.00  
## 45 11623.70 10312 48 100.00  
## 46 6000.40 10322 40 100.00  
## 47 3003.00 10333 26 100.00  
## 48 3944.70 10347 30 100.00  
## 49 5691.84 10357 32 100.00  
## 50 4514.92 10369 41 100.00  
## 51 8254.80 10381 36 100.00  
## 52 2416.56 10391 24 100.00  
## 53 4140.23 10411 23 100.00  
## 54 12001.00 10424 50 100.00  
## 55 3896.49 10107 39 99.91  
## 56 2793.86 10120 29 96.34  
## 57 3307.77 10134 27 100.00  
## 58 5192.95 10145 37 100.00  
## 59 5016.83 10159 37 100.00  
## 60 3660.93 10168 27 100.00  
## 61 4695.60 10180 42 100.00  
## 62 3660.92 10188 38 96.34  
## 63 3025.92 10201 24 100.00  
## 64 3009.09 10210 23 100.00  
## 65 5422.39 10223 47 100.00  
## 66 2852.08 10236 22 100.00  
## 67 5756.52 10251 44 100.00  
## 68 4472.00 10263 40 100.00  
## 69 2904.44 10275 22 100.00

linear <- glm(df$SALES~., data = train)  
pred <- predict(linear, newdata = test)  
plot(pred)



library(Metrics)

## Warning: package 'Metrics' was built under R version 4.3.3

rmse(pred, test$SALES)

## [1] 1818.152

summary(linear)

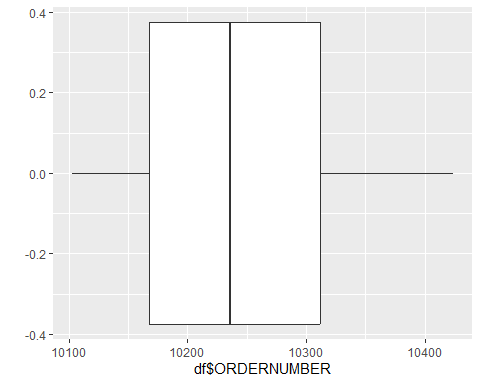
##   
## Call:  
## glm(formula = df$SALES ~ ., data = train)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -24777.787 26553.356 -0.933 0.3542   
## ORDERNUMBER 1.732 2.550 0.679 0.4995   
## QUANTITYORDERED 131.066 24.403 5.371 1.13e-06 \*\*\*  
## PRICEEACH 77.406 24.915 3.107 0.0028 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 3509104)  
##   
## Null deviance: 388713702 on 68 degrees of freedom  
## Residual deviance: 228091730 on 65 degrees of freedom  
## AIC: 1241.6  
##   
## Number of Fisher Scoring iterations: 2

library(tidyverse)

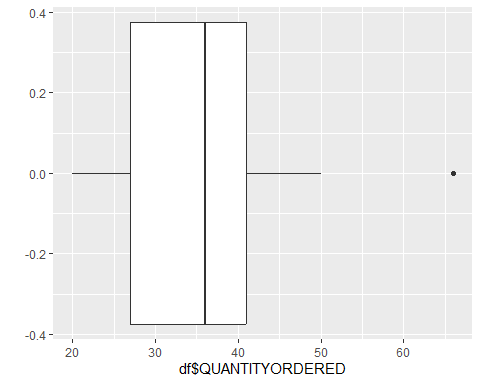
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── 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

qplot(df$ORDERNUMBER, geom="boxplot")

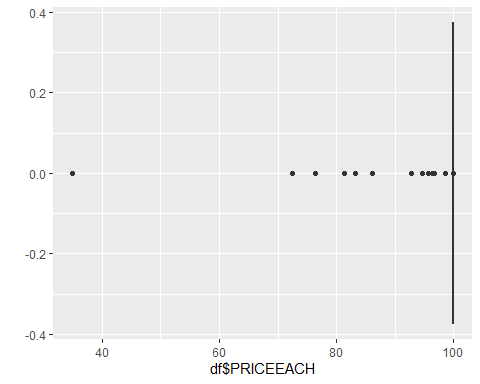
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



qplot(df$QUANTITYORDERED, geom="boxplot")

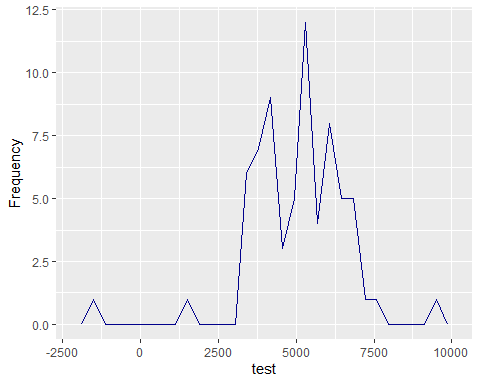


qplot(df$PRICEEACH, geom="boxplot")



qplot(pred, geom="freqpoly", xlab="test", ylab="Frequency", color=I("darkblue"))

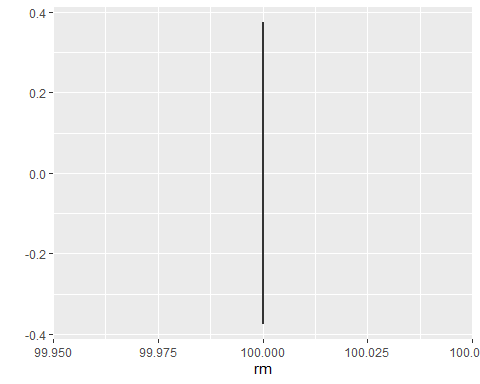
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 `

rm <-df$PRICEEACH[!df$PRICEEACH %in% boxplot.stats(df$PRICEEACH)$out]  
length(df$PRICEEACH)-rm

## [1] -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31  
## [20] -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31  
## [39] -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31 -31

qplot(rm, geom="boxplot")

 `

LN <- glm(df$SALES~df$QUANTITYORDERED)  
pred1<- predict(LN, newdata=test)  
rmse(test$SALES,pred1)

## [1] 1948.849