Chapter 9: Use Case Notebook for Instructors

Ram Gopal, Dan Philps, and Tillman Weyde

2022

Contents

Load library

```
library(ggplot2)
```

Use case: Fast Food Marketing Campaign - Non-parametric Tests

If we are not sure about the distribution of our population, and we do not want to make many assumptions about it, we can use non-parametric tests. This could be sensible in those cases where we just do not know enough about the population, perhaps because our data comes from a new process, or we have limited data to judge matters.

One such case is a fast-food chain which launches a new product and has three possible ways of promoting it. Over 500 outlet locations are selected to trial the product promotions, and the trial is conducted over several weeks. However, we cannot wait until the end of the trial to judge the outcome, and the senior management need regular updates on progress. If there is a stand out "winner", why wait until the end of the trial to roll it out to other outlets?

We will begin with the null hypothesis that all the Promotions generate equal median sales (SalesinThousands).

```
df = read.csv("../../data/WA_Marketing-Campaign.csv")
```

Which Promotion? A rough take from a Small Dataset

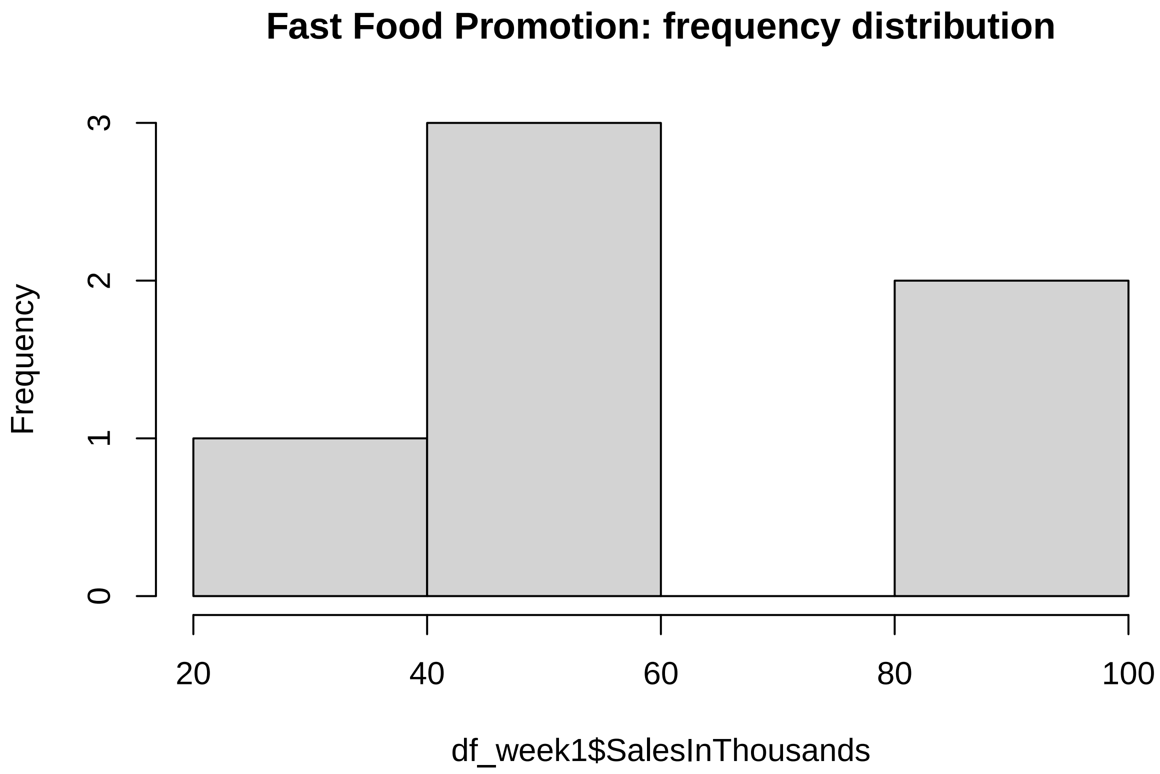
We have a problem though due to the store level management taking time to disaggregate the sales numbers for the different promotions. At the end of week 1, only 6 of our stores have reported:

```
store_locs = c(920, 217, 2, 3, 302, 203)
df_week1 = df[df$week==1 & df$LocationID %in% store_locs,]
```

This is an imbalanced and small sample, with three stores reporting Promotion 3, and only one reporting Promotion 1. We still need to get an idea of Promotion performance. Are SalesinThousands of Promotion 3 significantly better or worse than Promotion 1? We do not have distribution and other knowledge (yet), but we need to answer the question.

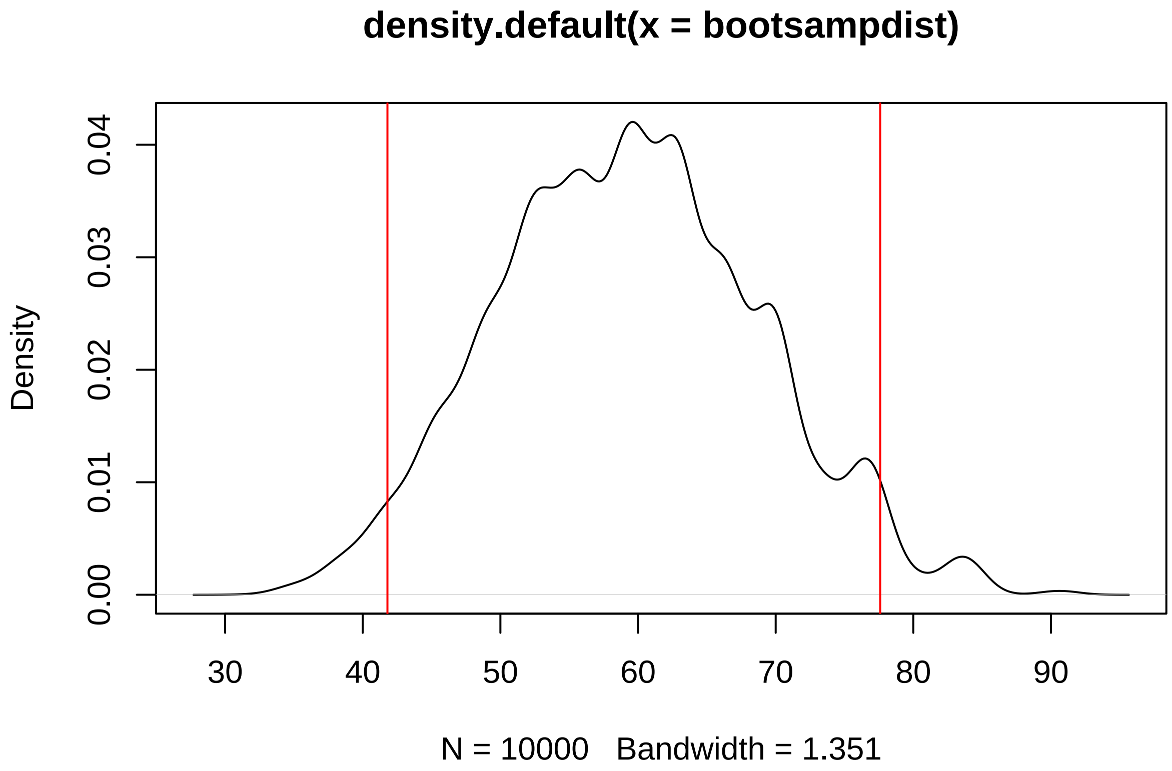
A frequency distribution with so few samples is not that insightful, but it shows the challenge we have to judge the different promotions:

```
hist(df_week1$SalesInThousands,breaks=4,
     main="Fast Food Promotion: frequency distribution")
```



Using bootstrapping we can construct a distribution and then compare the SalesInThousands numbers we have so far, and get a handle on the relative strength of the Promotions.

```
bootsampdist = replicate(10000, mean(sample(df_week1$SalesInThousands, replace = T)))
q2 = quantile(bootsampdist, c(.05/2,1-(.05/2)))
plot(density(bootsampdist))
abline(v = q2, col = "red")
```



```
paste("95% Confidence interval = [", round(q2,2)[1],", ",round(q2,2)[2],"]")
```

```
## [1] "95% Confidence interval = [ 41.79 ,  77.59 ]"
```

We now compare the median SalesInThousands values for each Promotion in turn:

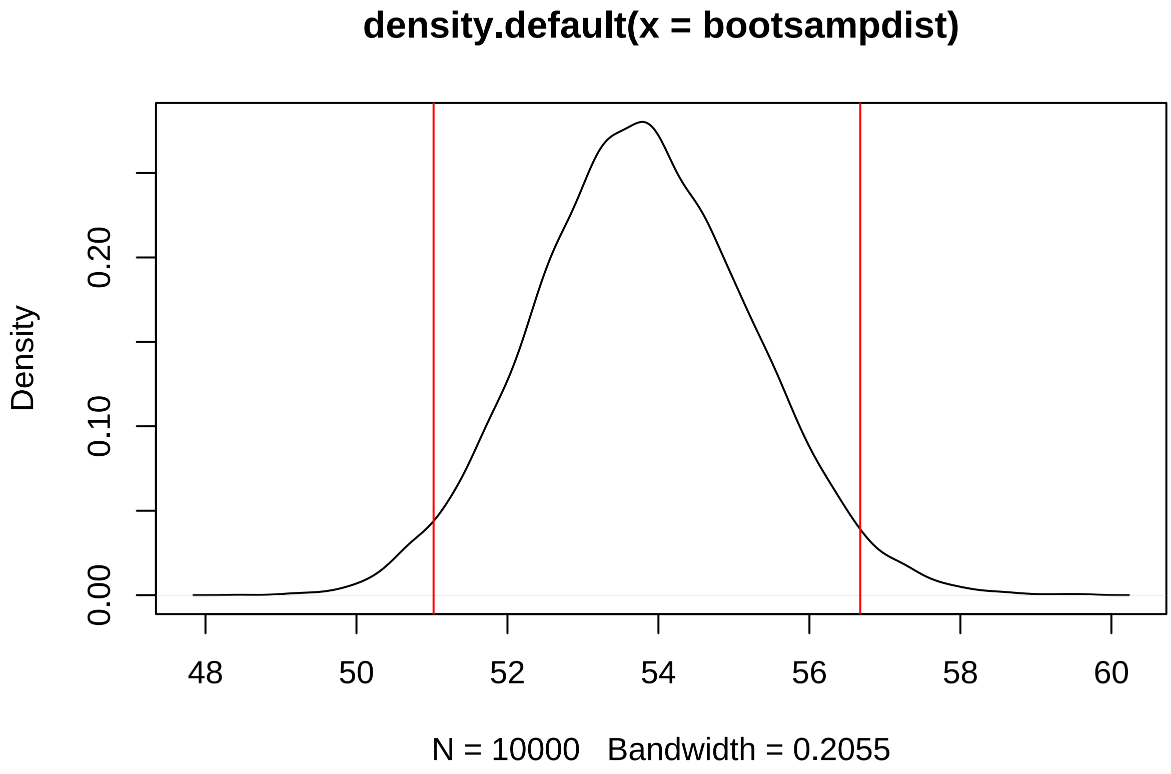
```
aggregate(df_week1$SalesInThousands,list(df_week1$Promotion),median)
```

```
##   Group.1     x
## 1       1 44.54
## 2       2 39.01
## 3       3 89.70
```

Which Promotion? More accurate… all results in for Week 1

We now have all the week 1 data in, significantly more than we did before, with 137 stores reporting. We can run the bootstrapping and testing process again on this larger dataset:

```
df1 = df[df$week==1,]
bootsampdist = replicate(10000, mean(sample(df1$SalesInThousands, replace = T)))
q2 = quantile(bootsampdist, c(.05/2,1-(.05/2)))
plot(density(bootsampdist))
abline(v = q2, col = "red")
```



```
paste("95% Confidence interval = [", round(q2,2)[1],", ",round(q2,2)[2],"]")
```

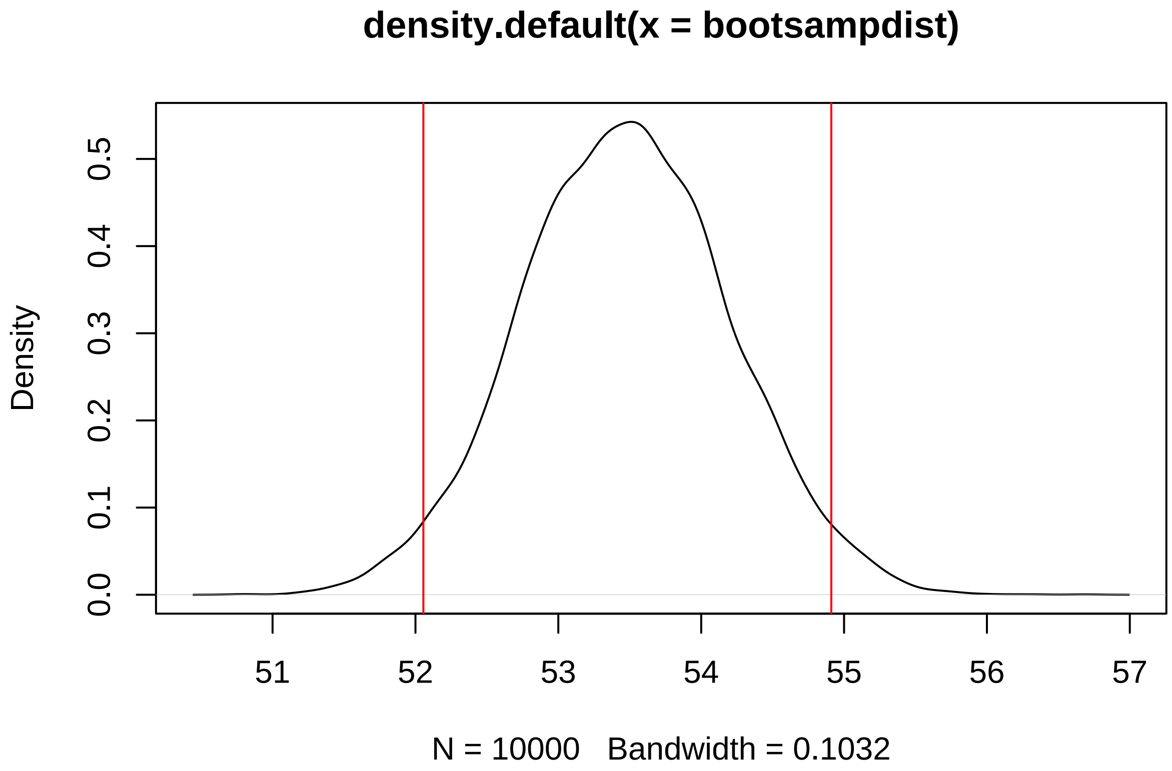
```
## [1] "95% Confidence interval = [ 51.02 ,  56.67 ]"
```

```
df1 = df[df$week==1,]
aggregate(df1$SalesInThousands,list(df1$Promotion),median)
```

```
##   Group.1     x
## 1       1 53.79
## 2       2 46.02
## 3       3 51.01
```

All the data available: judging from the empirical distribution

```
bootsampdist = replicate(10000, mean(sample(df$SalesInThousands, replace = T)))
q2 = quantile(bootsampdist, c(.05/2,1-(.05/2)))
plot(density(bootsampdist))
abline(v = q2, col = "red")
```



```r
paste("95% Confidence interval = [", round(q2,2)[1],", ",round(q2,2)[2],"]")
```

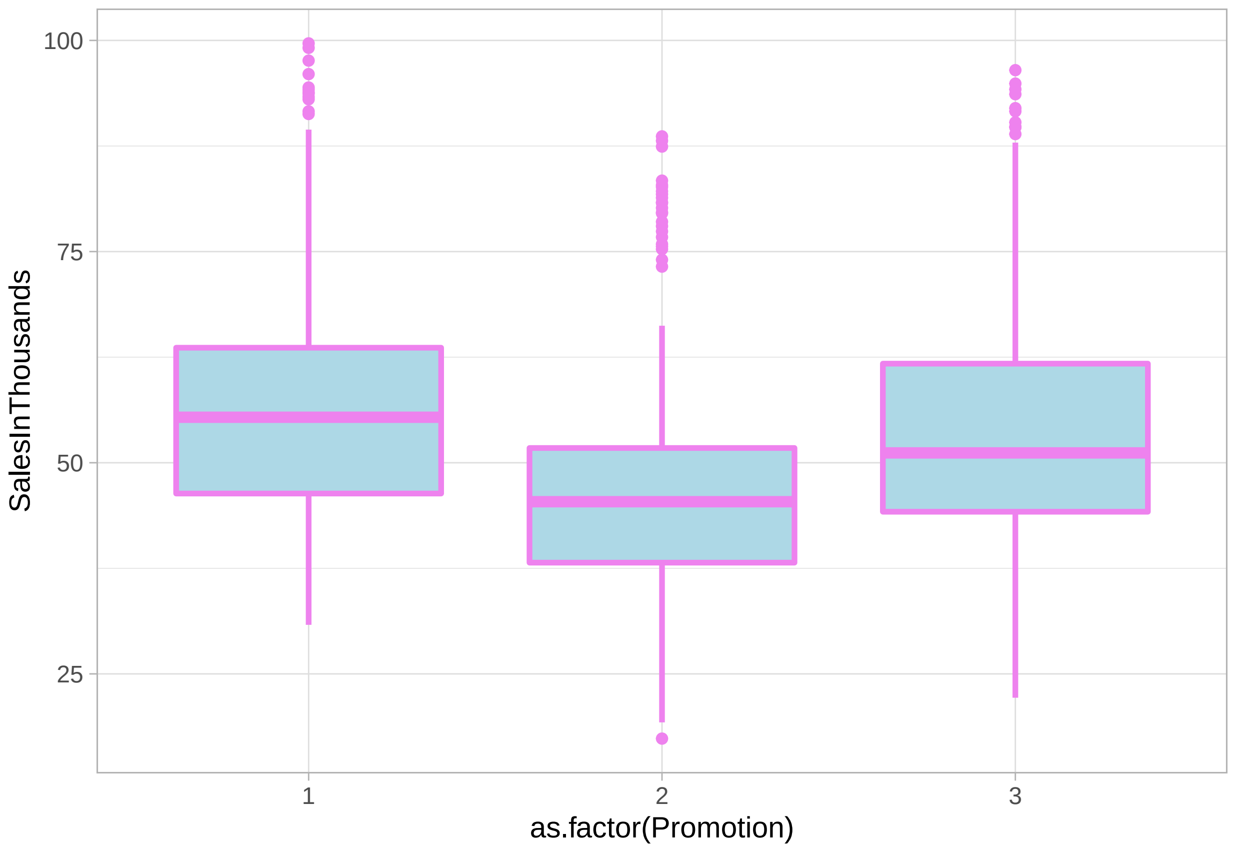
```
## [1] "95% Confidence interval = [ 52.06 ,  54.91 ]"
```

```r
aggregate(df$SalesInThousands,list(df$Promotion),median)
```

```
##   Group.1     x
## 1       1 55.39
## 2       2 45.39
## 3       3 51.16
```

Examining the empirical distributions of the Sales we can see that Promotion 1 does indeed appear to be the best option. However, we would be well advised to keep monitoring the situation going forwards.

```r
ggplot(df,aes(x = as.factor(Promotion) ,y=SalesInThousands)) +
  geom_boxplot(col = "violet", fill = "lightblue", size = 1) +
  theme_light()
```


100
75
50
25
SalesInThousands
1
2
3
as.factor(Promotion)