# The Statistical Sleuth in R: Chapter 4

Ruobing Zhang      Kate Aloisio      Nicholas J. Horton[*]

August 24, 2024

## Contents

## 1 Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Second Edition of the *Statistical Sleuth* (2002) by Fred Ramsey and Dan Schafer. More information about the book can be found at `http://www.proaxis.com/~panorama/home.htm`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.amherst.edu/~nhorton/sleuth`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')                # note the quotation marks
```

Once this is installed, it can be loaded by running the command:

---

[*]Department of Mathematics, Amherst College, nhorton@amherst.edu

```
> require(mosaic)
```

This needs to be done once per session.

In addition the data files for the *Sleuth* case studies can be accessed by installing the `Sleuth2` package.

```
> install.packages('Sleuth2')                    # note the quotation marks
```

```
> require(Sleuth2)
```

We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3, show.signif.stars=FALSE)
```

The specific goal of this document is to demonstrate how to calculate the quantities described in Chapter 4: The Rank-Sum Test using R.

## 2   Space Shuttle O-Ring Failures

Does launch temperature tend to cause O-ring incidents? This is the question being addressed by case study 4.1 in the *Sleuth*.

### 2.1   Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0401)

   Incidents        Launch
 Min.   :0.000    Cool: 4
 1st Qu.:0.000    Warm:20
 Median :0.000
 Mean   :0.417
 3rd Qu.:1.000
 Max.   :3.000

> favstats(Incidents ~ Launch, data=case0401)

  Launch min Q1 median  Q3 max mean    sd  n missing
1   Cool   1  1      1 1.5   3  1.5 1.000  4       0
2   Warm   0  0      0 0.0   2  0.2 0.523 20       0
```

A total of 24 subjects are included in the data: 4 O-ring incidents when the temperature was cold and 20 incidents when the temperature was warm (Display 4.1, page 86).

```
> histogram(~ Incidents | Launch, data=case0401)
```
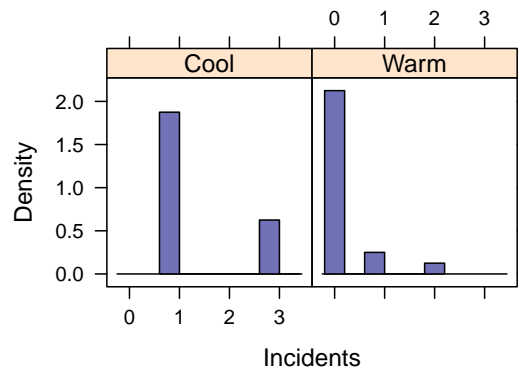


## 2.2 Permutation test on t-statistics

To replicate the permutation test performed on page 96 we use the following code, which first calculates the $t$-statistic of the observed outcome.

```
> t.test(Incidents ~ Launch, var.equal=TRUE, data=case0401)


 Two Sample t-test

data:  Incidents by Launch
t = 4, df = 22, p-value = 8e-04
alternative hypothesis: true difference in means between group Cool and group Warm is not equal
95 percent confidence interval:
 0.607 1.993
sample estimates:
mean in group Cool mean in group Warm
              1.5                0.2
```

We observe a test statistic of 3.888.

We want to get the total number of regroupings by calculating $C_{24,4}$.

```
> C244=factorial(24)/(factorial(4)*factorial(24-4)); C244

[1] 10626
```

There are a total of $1.063 \times 10^4$ regroupings with 8 possible (non-equiprobable) outcomes: (0, 0, 0, 0), (0, 0, 0, 1), (0, 0, 0, 2), (0, 0, 0, 3), (0, 0, 1, 1), (0, 0, 1, 2), (0, 0, 1, 3), (0, 0, 2, 3), (0, 1, 1, 1), (0, 1, 1, 2), (0, 1, 1, 3), (0, 1, 2, 3), (1, 1, 1, 1), (1, 1, 1, 2), (1, 1, 1, 3), (1, 1, 2, 3). Because the observed cold temperature outcomes was (1, 1, 1, 3), we will only examine the same or more extreme groupings, which are (1, 1, 2, 3) and (0, 1, 2, 3).

```
> # t.test for (1, 1, 2, 3)   # observations 1, 2, 4 and 24
> case0401$Incidents[c(1,2,4,24)]

[1] 1 1 3 2

> with(case0401, t.test(Incidents[c(1,2,4,24)], Incidents[-c(1,2,4,24)], var.equal=TRUE))


 Two Sample t-test

data:  Incidents[c(1, 2, 4, 24)] and Incidents[-c(1, 2, 4, 24)]
t = 6, df = 22, p-value = 5e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.04 2.16
sample estimates:
mean of x mean of y
     1.75      0.15

> # t.test for (0, 1, 2, 3)   # observation 1, 4, 5 and 24
> case0401$Incidents[c(1,4,5,24)]

[1] 1 3 0 2

> with(case0401, t.test(Incidents[c(1,4,5,24)], Incidents[-c(1,4,5,24)], var.equal=TRUE))


 Two Sample t-test

data:  Incidents[c(1, 4, 5, 24)] and Incidents[-c(1, 4, 5, 24)]
t = 4, df = 22, p-value = 8e-04
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.607 1.993
sample estimates:
mean of x mean of y
      1.5       0.2
```

The test statistic for $(1, 1, 2, 3)$ is 5.952 and the test statistic for $(0, 1, 2, 3)$ is 3.888.

We already know that the total number of regroupings is $C_{24,4}=1.063\times10^4$. In order to calculate the $p$-value, we need to know the combinations of $(1, 1, 1, 3)$, $(2, 1, 2, 3)$ and $(0, 1, 2, 3)$. There are 17 zeros, 5 ones, 1 two and 1 three.

For outcome $(1, 1, 1, 3)$, we calculate C1113$=C_{5,3}$*$C_{1,1}$:

```
> C1113 = factorial(5)/(factorial(3)*factorial(5-3))*1; C1113

[1] 10
```

For outcome (1, 1, 2, 3), we calculate C1123=$C_{5,2}$*$C_{1,1}$*$C_{1,1}$:

```
> C1123 = factorial(5)/(factorial(2)*factorial(5-2))*1*1; C1123

[1] 10
```

For outcome (0, 1, 2, 3), we calculate C0123=$C_{17,1}$*$C_{5,1}$*$C_{1,1}$*$C_{1,1}$

```
> C0123 = 17*5*1*1; C0123

[1] 85
```

Now we can calculate the $p$-value as the proportion of the number of rearrangements that are as extreme or more extreme over the total number of rearrangements:

```
> onep = (C1113+C1123+C0123)/C244; onep

[1] 0.00988
```

The one-sided $p$-value from the permutation test on the $t$-statistic is 0.01.
Alternatively, we can approximate the $p$-value using the difference of means and simulating repeatedly from the null distribution (note that the book enumerates all of the possible outcomes to get an exact result).

```
> result = t.test(Incidents ~ Launch, var.equal=TRUE, data=case0401)$statistic; result

     t
3.89

> nulldist = do(10000)*t.test(Incidents ~ shuffle(Launch), var.equal=TRUE, data=case0401)$stat
> histogram(~ t, groups=t >= result, v=result, data=nulldist)
> tally(~ t >= result, format="proportion", data=nulldist)

t >= result
  TRUE  FALSE
0.0109 0.9891
```
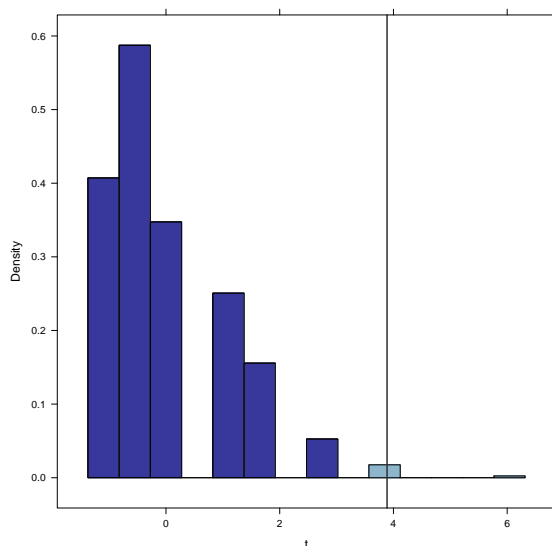
This simulation resulted in a *p*-value of 0.011.

# 3   Cognitive Load Theory in Teaching

Does use of modified instructional materials lead to quicker problem solving? That's the question being addressed by case study 4.2 in the *Sleuth*.

## 3.1   Summary statistics and graphical display

We begin by reading the data and summarizing the variables.

```
> summary(case0402)

      Time              Treatmt          Censor
 Min.   : 68    Modified    :14   Min.    :0.000
 1st Qu.:118    Conventional:14   1st Qu.:0.000
 Median :158                      Median :0.000
 Mean   :175                      Mean    :0.179
 3rd Qu.:232                      3rd Qu.:0.000
 Max.   :300                      Max.    :1.000

> favstats(Time ~ Treatmt, data=case0402)

        Treatmt min    Q1 median  Q3 max mean   sd  n missing
1      Modified  68  75.5    106 176 210  125 56.6 14       0
2  Conventional 130 152.8    235 300 300  224 70.5 14       0
```
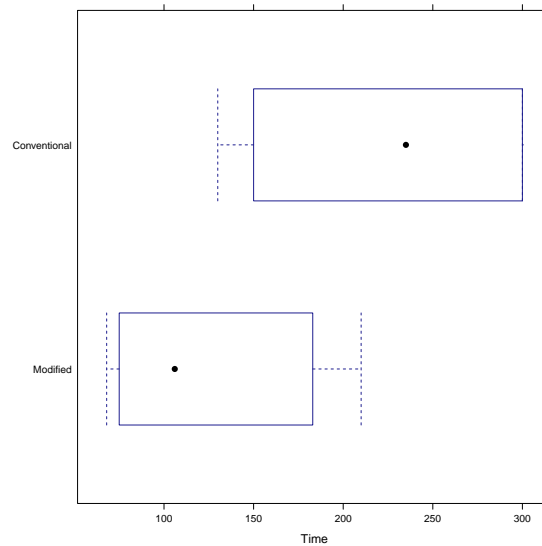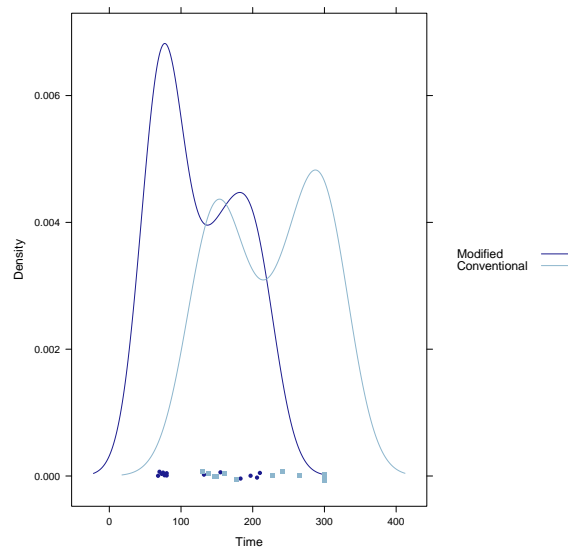
A total of 28 subjects are included in the data: 14 students were assigned to modified instructional materials and 14 students were assigned to conventional materials.

```
> bwplot(Treatmt ~ Time, data=case0402)
```



```
> densityplot(~ Time, groups=Treatmt, auto.key=TRUE, data=case0402)
```



## 3.2 Rank-sum test

We can calculate the one-sided $p$-value by following rank-sum procedure. First, we try to find the statistic T (display 4.5, page 91):

```
> obsrank = rank(case0402$Time, ties.method="average"); obsrank

 [1]  1.0  2.0  3.0  4.0  5.0  6.5  6.5  9.0 12.0 14.0 17.0 18.0 19.0 20.0  8.0
[16] 10.0 11.0 13.0 15.0 16.0 21.0 22.0 23.0 26.0 26.0 26.0 26.0 26.0

> mt = sum(obsrank[1:14]); mt

[1] 137
```

Next we calculate the $p$-value using a normal approximation (Display 4.7, page 93).

```
> average = mean(obsrank); average

[1] 14.5

> sd = sd(obsrank); sd

[1] 8.2

> n = nrow(subset(case0402, Treatmt=="Modified")); n

[1] 14

> MEANT = n * average; MEANT

[1] 203

> SDT = sd * sqrt((n^2)/(2*n)); SDT

[1] 21.7

> z = (mt-MEANT)/SDT; z

[1] -3.04

> p = pnorm(-abs(z)); p

[1] 0.00118
```

The one-sided $p$-value is 0.001.
Alternatively, we can use following code to calculate the Wilcoxon rank-sum test:

```
> wilcox.test(Time ~ Treatmt, conf.int=TRUE, exact=TRUE, data=case0402)

Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...):  cannot compute
exact p-value with ties
Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...):  cannot compute
exact confidence intervals with ties
```

```
 Wilcoxon rank sum test with continuity correction

data:  Time by Treatmt
W = 32, p-value = 0.003
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -160  -57
sample estimates:
difference in location
                  -94
```

So the one-sided $p$-value is 0.001. The 95% confidence interval is (-160, -57). The book suggests that the 95% confidence interval should be (58, 159), which is slightly narrower than these results. Their procedure is on page 94.