Personal Science Cookbook

Practical techniques to apply scientific principles to your own life.

Friends of Personal Science

9/9/2022

Table of contents

1 Getting Started

This book is intended to offer practical, step-by-step instructions to solve common problems when doing data analysis for Personal Science.

1.1 Prerequisites

We'll assume some basic tools.

- A spreadsheet like Microsoft Excel
- The programming language R and the associated development environment RStudio

A good introduction to R is Hands On Programming with R

1.1.1 R Packages

Nearly all of the examples in this book will use the following packages, each of which is probably already familiar if you've done some basic R programming already.

```
library(tidyverse)
library(lubridate)
library(kableExtra)
```

In addition, specific solutions may require other packages that will be identified in the instructions.

Part I

General Overview

2 Introduction

There's something you want to understand, probably about yourself, maybe something health- or wellness-related, but it might be something about the world around you. The point is that it's a question of deep interest to you, though unlikely in its current form to be of enough interest to involve professionals. You'd like to apply the principles of science — hypothesis, experiment, analysis — but you don't know enough of the mechanics to get started.

In other words, like a hungry person in a kitchen full of ingredients, you need a cookbook of recipes that can explain in a step-by-step, repeatable manner, how to go from the raw data around you to some fully-baked insights. That's the purpose of the Personal Science Cookbook. Each "recipe" is short and self-contained. Some are more complex than others, but none require any tools or knowledge beyond what is explained in the book.

3 The Principles of Personal Science

Personal Science is about empowering normal people to use the tools of science to help themselves in their daily lives.

When the first microchips enabled desktop computers in the 1970s, people were unsure what to call them.

The word "mini-computer" was already taken, referring to a generation of computers that didn't require entire rooms, so the techie engineers who confronted these new machines called them "microcomputers", a moniker that lives on in the name for one of the first software companies of that generation, Microsoft.

Some people called them "hobby computers", because that seemed to be all they were good for. The most influential early gathering of people using them was called the "Homebrew Computer Club". The term "desktop" was gaining traction, and inspired later generations that called them "laptops", but then the most traditional of all computer companies introduced its first "IBM PC", and suddenly the industry had a new term.

It was a "personal computer" because, for the first time, it was cheap enough and easy enough for a single individual to use it by him (or her) self. In contrast to all previous generations of computing, everything about the device was intended to be used by a single individual. Even if the computer was shared, only one person would use it at a time, and all design decisions reflected that: a single keyboard, monitor, one power switch. You didn't need a team of people to set up and care for the device — it was out-of-the-box something that a single person could set up and use.

It's easy to forget how transformative this was at the time. Computers until then were very expensive — many times more than the cost of a car or even a house. You had to be a large organization — a university, a business — to afford one, and even if somebody magically just gave one to you, you'd need a special place to keep it, with highly-trained technicians just to keep it running, and of course even more well-trained engineers and scientists to get it to do anything useful.

A similar situation exists today in science. New discoveries are made in large institutions, by teams of high-trained people with access to large, expensive equipment. The discoveries are discussed and shared by specialists who are followed by a cadre of specialized interpreters —

journalists, educators, clinicians — who decipher the new scientific results into lay language and ultimately into face-to-face interaction with the public. Committees meet to discuss takeaways from the expensive and time-consuming research, reaching conclusions that are considered generally acceptable enough to result in new actionable treatments and suggestions for "normal people".

This gap between the specialists and the general public, like the gap between mainframe computers and PCs, is eroding thanks to technology.

Actually that's not quite true: the *potential* gap between specialists and the general public is eroding. But reality is still different. It's as if PCs had been invented but no software.

The personal computer revolution was about more than simply cheaper devices. The hardware became useful after it spawned an entire industry of dedicated software makers, educational experts, consultants and systems integrators,

Professional science

We all think science is great…

but what do people mean when they say "science"? 1. Wonder (photos of stars, micrographs, etc.) 2. Technology (photos of roman arch, integrated circuit, moon landing) 3. A way of thinking (photos of "amateur" scientists)

It's tempting to assume that the scientific way of thinking is obvious, and maybe even obviously the only way to think rigorously but that's not really true.

Alternatives to the scientific way of thinking: recipes

My definition of science: a predisposition to the assumption that you're wrong, a nasty mischievous inclination to disbelieve things you can't prove.

A core scientific skill is *curiosity*. Always ask "what if…" thinking in hypotheticals

Religion seems like a classic example of unscientific thinking, but even that I'll challenge. What if you're wrong? Is there a way to experiment, test it?

Science is:

- Curiosity
- Skepticism : an unending belief that you are wrong
  - Low interest in credentials … just because you are "certified" doesn't mean you know any more than I do.
- Bias toward experiments

See Roberts (2004) for examples.

Part II

Techniques

4 Methods

4.1 What is a dataframe?

Self-collected data is almost always best represented by a table of the variables you want to study and the values that you collected for each of those variables. The most common type of table is a spreadsheet, a specific form of which in Personal Science we refer to as a data table or a *data frame*. Abbreviated "dataframe" or often just "df", it's a table of values and variables that always has the same form:

- columns are variables: the parameters you want to study
- rows are observations: each incident of data you collected.

It's important to get in the habit of this row/column approach to data collection because, as you'll see, all of our tools assume that data will come in a dataframe format.

4.2 Is it chance? T-Test

Problem

You tried an intervention and want to see if it worked. How likely is it that the results were chance?

Solution

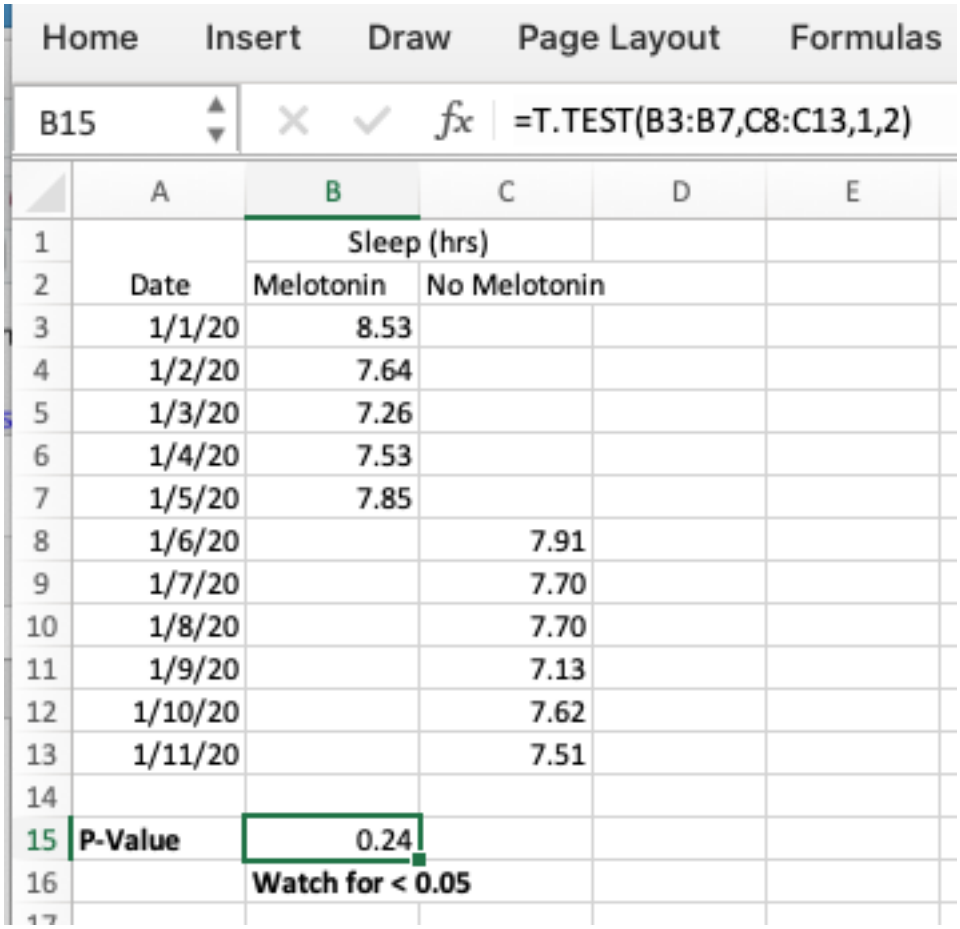
One of the simplest tests is a "T-Test", sometimes called a "Student T Test".

Statisticians use the concept of *P Value* to discuss the how often a result might appear to be significant even when it's not. While this crude measure doesn't describe all the ways something might happen due to chance, generally the lower the P Value, the better. Professional scientists, especially those who understand statistics, will get touchy if you claim a result based purely on P Values, but for Personal Science purposes, it's a good start. There is no "correct" cutoff value that can determine the likelihood that something is due to chance alone, but traditionally people assume that anything under 0.05 deserves a closer look.

Here's an example for how to do this in Excel.

Suppose you'd like to know if taking a melotonin supplement will help you sleep longer. You've measured your daily sleep, taking the supplements on some days (the "intervention") and not on others ("control").

A simple spreadsheet might look like this:



	A	B	C	D	E
1		Sleep (hrs)			
2	Date	Melotonin	No Melotonin		
3	1/1/20	8.53			
4	1/2/20	7.64			
5	1/3/20	7.26			
6	1/4/20	7.53			
7	1/5/20	7.85			
8	1/6/20		7.91		
9	1/7/20		7.70		
10	1/8/20		7.70		
11	1/9/20		7.13		
12	1/10/20		7.62		
13	1/11/20		7.51		
14					
15	P-Value	0.24			
16		Watch for < 0.05			

Track your sleep under two columns: one for nights when you took the supplement, and the other for nights you didn't.

The built-in Excel statistical function `T.TEST` will calculate the P-Value when you give it two ranges, the "intervention" (nights we took melotonin) and the "control" (nights without).

See the screenshot for the exact formula in this case:

```
=T.TEST(array1,array2,tails,type)
```

Enter a `1` for `tails` (because we're only interested in one measurement, sleep) and a `2` for type (because in this case our samples are not of the same length).

The P Value in this example, `0.24`, is above `0.05` and therefore we will assume that any difference in sleep between the nights is due to pure chance.

4.3 Rolling average

Problem You want to take the rolling 7-day average of a series of numbers.

```
library(tidyverse)
headache_df <- readr::read_csv("headache-variables.csv")
headache_df %>% head() %>% knitr::kable()
```

date	headache	icecream	z	wine
2022-06-06	FALSE	TRUE	7.422476	0
2022-06-07	FALSE	FALSE	6.844830	0
2022-06-08	TRUE	FALSE	5.960463	0
2022-06-09	FALSE	FALSE	7.597812	0
2022-06-10	FALSE	FALSE	7.644585	0
2022-06-11	FALSE	FALSE	7.103670	0

Solution use the `rolling()` functions in package `zoo`:

```
library(zoo)

headache_df %>%
    mutate(sleep7A = rollapply(z,
                               7,
                               function(x) {x = mean(x,na.rm = TRUE)},
                               align = 'right',
                               fill = NA)) %>%
  tail() %>% knitr::kable()
```

date	headache	icecream	z	wine	sleep7A
2022-09-06	FALSE	FALSE	7.106073	0	7.417971
2022-09-07	FALSE	TRUE	7.491277	0	7.323586
2022-09-08	FALSE	FALSE	8.449556	0	7.531646
2022-09-09	FALSE	FALSE	7.028875	0	7.672542
2022-09-10	FALSE	FALSE	8.197885	0	7.656885
2022-09-11	FALSE	FALSE	8.398064	0	7.679493

5 Applications

Some *significant* applications are demonstrated in this chapter.

Let's say you are suffering from unexplained headaches that appear somewhat randomly. You suspect they may be associated with something you eat, but you're not sure, so you've been tracking 14 weeks (98 days) worth of your own data in a spreadsheet that looks like this:

```
library(tidyverse, quietly=TRUE)
library(lubridate, quietly=TRUE)
```

```
x <- tibble(date=seq(from = today()-weeks(14),
                     by = "1 day", length.out = 7*14),
            headache = sample(c(TRUE,FALSE), 7*14,
                              prob = c(.05,.95),
                              replace = TRUE))

knitr::kable( head(x) ) %>% kableExtra::kable_styling()
```

```
write_csv(x,"headache-days.csv")
```

You can download a copy of this file here

With my 14 weeks of data, we can do a few basic calculations:

How frequent are my headaches? Simply total the number of headaches and divide by number of days:

date	headache
2022-06-06	FALSE
2022-06-07	FALSE
2022-06-08	FALSE
2022-06-09	FALSE
2022-06-10	FALSE
2022-06-11	FALSE

```
# headaches per day
sum(x$headache) / length(x$headache)
```

```
[1] 0.04081633
```

5.1 Hypothesis

With the data collected and in a nice dataframe format, we can start to ask what might be driving the headaches. One of the first suspected culprits might be something that I eat. It's easy to add a few more variables (columns) to the dataframe: (download)

```
z <- function(x){
m = NULL
for(i in 1:14){
  m = c(c(rep(0,6),
              floor(runif(1,min=0,max=3))),
        m)
}

m
}
x <- tibble(date=seq(from = today()-weeks(14),
                     by = "1 day", length.out = 7*14),
            headache = sample(c(TRUE,FALSE), 7*14,
                              prob = c(.05,.95),
                              replace = TRUE),
            icecream = sample(c(TRUE,FALSE), 7*14,
                              prob = c(.10,.90),
                              replace = TRUE),
            z = runif(7*14, min = -2.5, max = .5) + 8,
            wine = z(0))

knitr::kable( head(x,10), digits = 2) %>%
  kableExtra::kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

```
write_csv(x,"headache-variables.csv")
```

- `headache`: a day when I have a headache
- `icecream`: did I eat ice cream that day?

date	headache	icecream	z	wine
2022-06-06	FALSE	FALSE	7.07	0
2022-06-07	FALSE	FALSE	5.70	0
2022-06-08	FALSE	FALSE	7.26	0
2022-06-09	FALSE	FALSE	5.76	0
2022-06-10	FALSE	FALSE	8.31	0
2022-06-11	FALSE	FALSE	7.30	0
2022-06-12	FALSE	FALSE	5.67	0
2022-06-13	TRUE	FALSE	6.03	0
2022-06-14	FALSE	TRUE	7.12	0
2022-06-15	FALSE	FALSE	8.03	0

date	headache	icecream	z	wine
2022-06-13	TRUE	FALSE	6.029775	0
2022-07-20	TRUE	FALSE	7.027660	0
2022-08-08	TRUE	FALSE	8.089417	0
2022-08-10	TRUE	FALSE	8.271079	0
2022-08-15	TRUE	FALSE	7.462703	0

- `wine`: Number of glasses of wine I drank.
- `z`: Number of hours I slept that day.

Based on the data collected so far, can I make any guesses about what might be driving my headaches?

The most obvious place to check is whether I see any patterns on the days when I have headaches. Let's filter for headache days only:

```
x %>% filter(headache)  %>% kableExtra::kable() %>%
  kableExtra::kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

But maybe the headache takes a day or two to kick in. We can divide the data by week and see if we can spot any patterns in headache frequency:

```
x %>% group_by(week = ntile(date,7)) %>%
  summarise(headaches = sum(headache),
            alcohol = sum(wine),
            icecream = sum(icecream))  %>% kableExtra::kable() %>%
  kableExtra::kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

week	headaches	alcohol	icecream
1	1	0	2
2	0	2	2
3	0	2	2
4	1	2	1
5	2	3	2
6	1	3	1
7	0	0	0

Buy simply eye-balling the data this way, you might see a pattern. For example, you might spot a week or two with an unusually large number of headaches and notice those weeks are accompanied by an unusually large consumption of some particular food.

But how do you know you're not just guessing? What *looks* like a pattern might be a coincidence. To find out with more certainty, we will apply some statistics.

5.2 T-Testing

The simplest test is called a "T Test". This is a formula that can compare two equal-sized lists of numbers and return the probability that any differences between the two are the result of chance.

What are the chances that the number of headaches per week is related to the amount of ice cream I eat per week?

If there were a relationship between ice cream and headaches each week, I'd expect that over the weeks in this period, the total number of headaches and the total number of ice cream days should be roughly equal.

```
x_week <- x %>% group_by(week = ntile(date,7)) %>%
  summarise(headaches = sum(headache),
            alcohol = sum(wine),
            icecream = sum(icecream))
x_week %>% kableExtra::kable() %>%
  kableExtra::kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

```
with(x_week, t.test(headaches,icecream))[["p.value"]]
```

```
[1] 0.108905
```

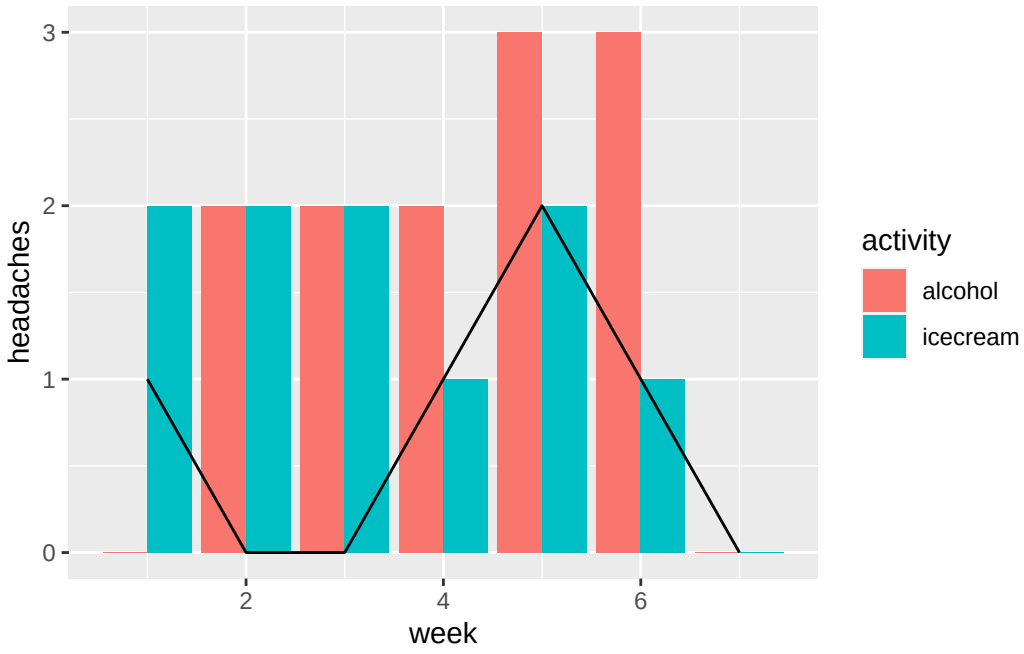
week	headaches	alcohol	icecream
1	1	0	2
2	0	2	2
3	0	2	2
4	1	2	1
5	2	3	2
6	1	3	1
7	0	0	0

By convention, a p-value less than 0.05 (that is, less than 5\%) is considered statistically significant. While this is not a hard and fast rule, it's often a good place to start. A p-value greater than this is almost certainly due to chance.

5.3 Data visualization

The first step in a more sophisticated analysis is to plot the data to see if we can spot any particular patterns.

```
x_week %>% pivot_longer(names_to = "activity",
                        values_to = "amount",
                        cols = alcohol:icecream ) %>%
  ggplot(aes(x=week,y=headaches)) +
  geom_bar(aes(x=week,y=amount, fill = activity),
           position = "dodge",
           stat = "identity") +
  geom_line(aes(x=week,y=headaches))
```



5.4 Example two

6 Final Words

Download a copy in PDF or ePub.

7 References

Roberts, Seth. 2004. "Self-Experimentation as a Source of New Ideas: Ten Examples about Sleep, Mood, Health, and Weight." http://www.escholarship.org/uc/item/2xc2h866.