Assignment 0: Solution

Alex Hoagland

2024-09-20

## Problem 1:

Here, we are just using R as a calculator to compute the following values. Any discussion you have about the problem goes here (or below the code chunk).

27\*(38-17)

## [1] 567

log(14^7) # note: log in computer language is base e, so we would write this as ln in math

## [1] 18.4734

sqrt(436/12)

## [1] 6.027714

So you can see our answers are 567, 18.5, and 6, respectively. If you have any other takeaways, they go here.

## Problem 2:

This problem asked us to create and manipulate some vectors:

a <- c(5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100,105,110,115,120,125,130,135,140,145,150,155,160) # one way to do it  
a <- seq(5,160,5) # another way to do it  
b <- seq(87,56,-1) # this is a decreasing sequence  
 # note: did you try to get Copilot to create these? How can you? (show by copying and pasting)  
  
d <- a \* b # what does this do?   
d2 <- a %\*% b # note that this is how we do matrix multiplication (why would we want this?)!  
  
d[19:21] # question (a)

## [1] 6555 6800 7035

d[which(d < 2000)] # question (b)

## [1] 435 860 1275 1680

length(d[which(d > 6000)]) # question (c)

## [1] 16

The answers to the specific questions here are: (a) 6555,6800, and 7035 (b) 435, 860, 1275, and 1680 (c) 16 elements (50%).

## Problem 3:

Now we summarize the vector d:

# these are easy!  
sum(d)/length(d)

## [1] 5472.5

mean(d)

## [1] 5472.5

median(d)

## [1] 5897.5

sd(d)

## [1] 2608.563

We find that the average value in d is 5,472.5, with a median of 5897.5 (what does that tell us about the distribution?). The standard deviation is quite large at 2608.57.

## Problem 4:

This problem asks us to perform matrix multiplication.

A <- rbind(c(7,9,12),c(2,4,13))  
B <- rbind(c(1,7,12,19),c(2,8,13,20), c(3,9,14,21))  
A %\*% B

## [,1] [,2] [,3] [,4]  
## [1,] 61 229 369 565  
## [2,] 49 163 258 391

The resulting matrix is a 2x4 matrix, which we could represent in R Markdown using LaTeX (if we wanted to):

## Problem 5:

Now we get to work with some data!

# install.packages("NHSRdatasets") # note: these installation commands need to be commented out when knitting a document!  
library(NHSRdatasets)

## Warning: package 'NHSRdatasets' was built under R version 4.2.3

mydata <- ons\_mortality  
  
# Now we answer the questions  
# We have two categorical variables (category\_1 and category\_2)  
# We have two numeric variables (counts and week\_no)  
# We also have a date variable   
summary(mydata$counts)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0 220 912 1448 1412 16387 51

summary(mydata)

## category\_1 category\_2 counts date   
## Length:18803 Length:18803 Min. : 0 Min. :2010-01-08   
## Class :character Class :character 1st Qu.: 220 1st Qu.:2012-08-31   
## Mode :character Mode :character Median : 912 Median :2015-04-10   
## Mean : 1448 Mean :2015-04-14   
## 3rd Qu.: 1412 3rd Qu.:2017-12-08   
## Max. :16387 Max. :2020-04-03   
## NA's :51   
## week\_no   
## Min. : 1.0   
## 1st Qu.:12.0   
## Median :25.0   
## Mean :25.5   
## 3rd Qu.:39.0   
## Max. :53.0   
##

sd(mydata$counts, na.rm=T)

## [1] 2247.576

summary(mydata$week\_no)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 12.0 25.0 25.5 39.0 53.0

sd(mydata$week\_no, na.rm=T)

## [1] 15.28142

levels(factor(mydata$category\_1))

## [1] "All respiratory diseases (ICD-10 J00-J99) ICD-10"   
## [2] "average of same week over 5 years"   
## [3] "Deaths where COVID-19 was mentioned on the death certificate (ICD-10 U07.1 and U07.2)"  
## [4] "Females"   
## [5] "Males"   
## [6] "Persons"   
## [7] "Region"   
## [8] "Total deaths"

levels(factor(mydata$category\_2))

## [1] "01-14" "1-4"   
## [3] "10-14" "15-19"   
## [5] "15-44" "20-24"   
## [7] "25-29" "30-34"   
## [9] "35-39" "40-44"   
## [11] "45-49" "45-64"   
## [13] "5-9" "50-54"   
## [15] "55-59" "60-64"   
## [17] "65-69" "65-74"   
## [19] "70-74" "75-79"   
## [21] "75-84" "80-84"   
## [23] "85-89" "85+"   
## [25] "90+" "all ages"   
## [27] "average of same week over 5 years" "East"   
## [29] "East Midlands" "London"   
## [31] "North East" "North West"   
## [33] "South East" "South West"   
## [35] "Under 1 year" "v 2001"   
## [37] "v 2010" "v 2013 (IRIS)"   
## [39] "Wales" "West Midlands"   
## [41] "Yorkshire and The Humber"

# give me all of the unique values of mydata$category2  
# unique(mydata$category\_2)

These commands give some hopefully useful additional commands you can play around with to get a sense of what your data is telling you.

## Problem 6:

Now we will summarize that data and create a nice table.

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.2

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.2

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ 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

mydata %>% filter(category\_1 == "Total deaths") %>% group\_by(week\_no) %>% summarise(mean\_deaths = mean(counts)) %>%  
knitr::kable()

| week\_no | mean\_deaths |
| --- | --- |
| 1 | 12071.381 |
| 2 | 13009.952 |
| 3 | 12070.410 |
| 4 | 11543.248 |
| 5 | 11170.914 |
| 6 | 10909.067 |
| 7 | 10824.219 |
| 8 | 10834.038 |
| 9 | 10667.171 |
| 10 | 10670.105 |
| 11 | 10539.248 |
| 12 | 10136.724 |
| 13 | 9724.705 |
| 14 | 10249.600 |
| 15 | 10186.800 |
| 16 | 10081.170 |
| 17 | 9943.770 |
| 18 | 9750.590 |
| 19 | 9280.630 |
| 20 | 9715.530 |
| 21 | 9573.170 |
| 22 | 8320.680 |
| 23 | 9315.560 |
| 24 | 9241.130 |
| 25 | 9033.460 |
| 26 | 8956.070 |
| 27 | 8925.780 |
| 28 | 8836.660 |
| 29 | 8810.330 |
| 30 | 8778.940 |
| 31 | 8765.710 |
| 32 | 8739.110 |
| 33 | 8754.130 |
| 34 | 8846.020 |
| 35 | 7987.470 |
| 36 | 8957.430 |
| 37 | 8911.810 |
| 38 | 8929.330 |
| 39 | 9083.570 |
| 40 | 9211.640 |
| 41 | 9343.620 |
| 42 | 9415.290 |
| 43 | 9493.030 |
| 44 | 9605.800 |
| 45 | 9779.560 |
| 46 | 9888.840 |
| 47 | 9834.700 |
| 48 | 9819.030 |
| 49 | 10366.250 |
| 50 | 10532.860 |
| 51 | 11270.010 |
| 52 | 8200.080 |
| 53 | 7831.500 |

Note that kable isn’t the only option for making tables! You’ll explore this more in Assignment 1, but there are many packages and you should explore the ones that make sense to you and have output you like.

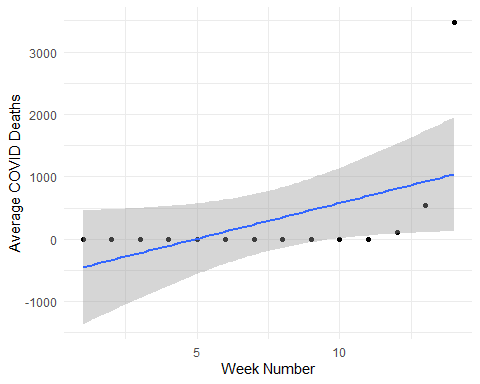
## Problem 7:

Now we will create a plot of the data.

Using your loaded data, make a scatterplot showing the relationship between week\_no and (the average number of) deaths where COVID-19 was mentioned on the death certificate

# make a scatterplot showing the relationship in mydata between week\_no counts when category1 includes COVID-19  
  
# note: at least for me, copilot was useful but insufficient here!  
mydata %>% filter(category\_1 == "Deaths where COVID-19 was mentioned on the death certificate (ICD-10 U07.1 and U07.2)") %>% group\_by(week\_no) %>% summarise(mean\_deaths = mean(counts)) %>% ggplot(aes(x=week\_no, y=mean\_deaths)) + geom\_point() + geom\_smooth(method="lm") + labs(x="Week Number", y = "Average COVID Deaths") + theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



# What's happening here? Did we do something wrong?

Notice that the data is incomplete for this! We don’t have any COVID death information after week 14 (mid-March 2020?), so we wouldn’t be able to use this data to think about COVID deaths over the course of 2020. So plotting the data is really helpful before we just jump off into an analysis.

## Problem 8:

Finally, we compile the document. Note: if you have any install.packages commands in your document, you should comment them out before knitting the document.