015 - Inference about a Population Mean (μ)

EPIB 607

Sahir Rai Bhatnagar
Department of Epidemiology, Biostatistics, and Occupational Health
McGill University

sahir.bhatnagar@mcgill.ca

slides compiled on September 27, 2021

The t distribution

Examples

Inference for μ when σ is not known

Up until now, all of our calculations have relied on us knowing the value of the population standard deviation (σ). It is rare that this is the case.

We now consider methods of inference for when σ is unknown.

When σ is unknown, we must estimate it from the data using s, the sample standard deviation.

Inference for μ when σ is unknown

- When the true variance was known, we performed our calculations using the standardization

$$Z = \frac{\overline{y} - \mu}{\sigma / \sqrt{n}} \sim \mathcal{N}(0, 1).$$

Inference for μ when σ is unknown

- When the true variance was known, we performed our calculations using the standardization

$$Z = \frac{\overline{y} - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1).$$

- We no longer can use this, so instead we use

$$t = \frac{\overline{y} - \mu}{s/\sqrt{n}} \sim t_{(n-1)}$$

  which follows a ***t*-distribution** with $n - 1$ degrees of freedom based on the n values, $y_1, \ldots, y_n$ in an SRS

Inference for μ when σ is unknown

- When the true variance was known, we performed our calculations using the standardization

$$Z = \frac{\overline{y} - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1).$$

- We no longer can use this, so instead we use

$$t = \frac{\overline{y} - \mu}{s/\sqrt{n}} \sim t_{(n-1)}$$

  which follows a ***t*-distribution** with $n - 1$ degrees of freedom based on the n values, $y_1, \ldots, y_n$ in an SRS

- There is a different t distribution for each sample size. The degrees of freedom specify which distribution we use, and are determined by the denominator used in estimating s which is $(n - 1)$.

σ known vs. unknown

σ	known	unknown
Data	$\{y_1, y_2, \ldots, y_n\}$	$\{y_1, y_2, \ldots, y_n\}$
Pop'n param	μ	μ
Estimator	$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$	$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$
SD	σ	$s = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}}$
SEM	$\sigma/\sqrt{n}$	$s/\sqrt{n}$
$(1-\alpha)100\%$ CI	$\bar{y} \pm z^{\star}_{1-\alpha/2}(\text{SEM})$	$\bar{y} \pm t^{\star}_{1-\alpha/2,(n-1)}(\text{SEM})$
test statistic	$\frac{\bar{y}-\mu_0}{\text{SEM}} \sim \mathcal{N}(0,1)$	$\frac{\bar{y}-\mu_0}{\text{SEM}} \sim t_{(n-1)}$

t distribution vs. Normal distribution

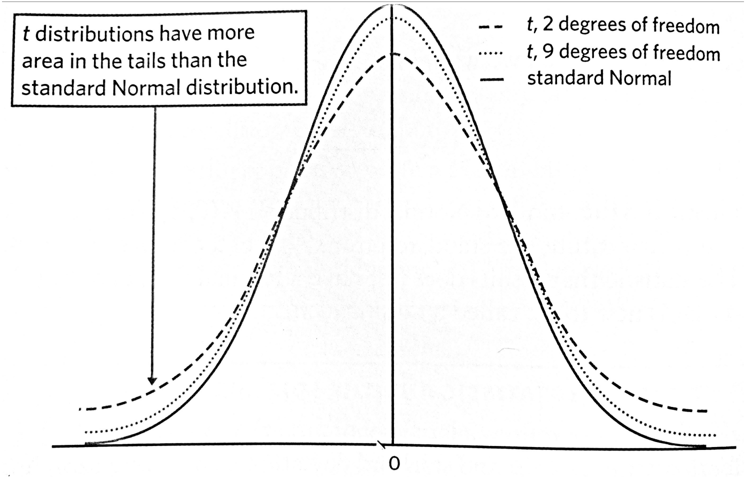
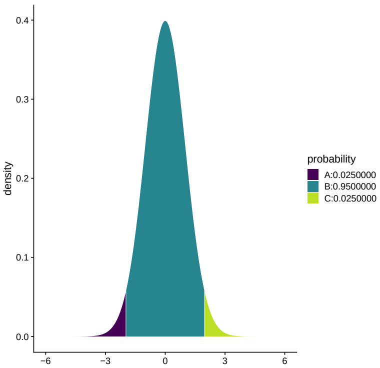


Figure: Density curves for the t distribution with 2 and 9 degrees of freedom and for the standard Normal distribution. All are symmetric with center 0. The t distributions are somewhat more spread out.

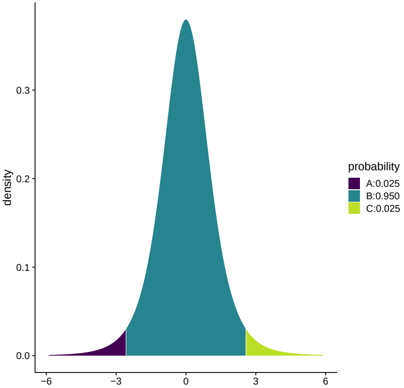
$t_{(5)}$ distribution vs. Standard Normal distribution

```r
library(mosaic)
xqnorm(p = c(0.025, 0.975))
```




```
## [1] -1.959964  1.959964
```

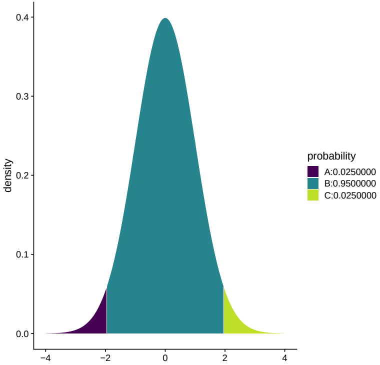
```r
library(mosaic)
xqt(p = c(0.025, 0.975), df = 5)
```




```
## [1] -2.570582  2.570582
```

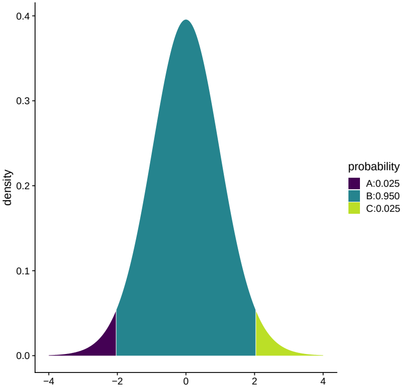
$t_{(30)}$ distribution vs. Standard Normal distribution

```r
library(mosaic)
xqnorm(p = c(0.025, 0.975))
```




```
## [1] -1.959964  1.959964
```

```r
library(mosaic)
xqt(p = c(0.025, 0.975), df = 30)
```




```
## [1] -2.042272  2.042272
```

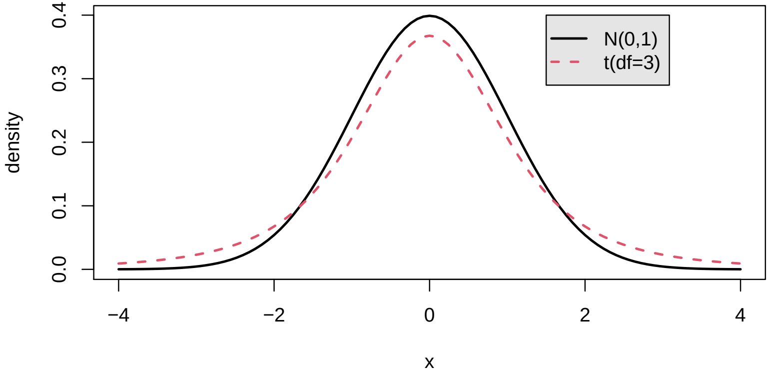
t distributions

- Is symmetric around 0 (just like the $\mathcal{N}(0, 1)$)
- Has a shape like that of the Z distribution, but with a SD slightly larger than unity i.e. slightly flatter and heavier-tailed
- Shape becomes indistinguishable from Z distribution as $n \to \infty$ (in fact as n goes much beyond 30)
- Instead of $\pm 1.96\times$ SEM for 95% confidence (or to use as the critical value in a null-hypothesis test), we need these multiples (or critical values):

n	'degrees of freedom'	Multiple	from `R`
2	1	12.71	`qt(0.975, 1)`
3	2	4.30	`qt(0.975, 2)`
4	3	3.18	`qt(0.975, 3)`
11	10	2.23	`qt(0.975, 10)`
21	20	2.09	`qt(0.975, 20)`
31	30	2.04	`qt(0.975, 30)`
121	120	1.98	`qt(0.975,120)`
∞	∞	1.96	`qt(0.975,Inf)`

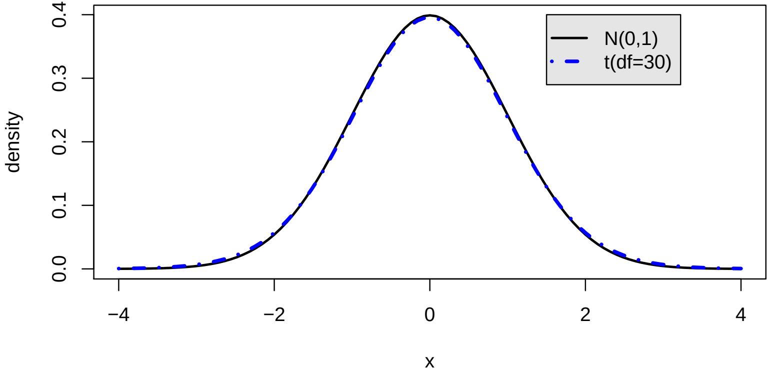
t distributions

Sample size increases → degrees of freedom increase → *t* starts to look like $\mathcal{N}(0,1)$



t distributions

Sample size increases $\rightarrow$ degrees of freedom increase $\rightarrow$ t starts to look like $\mathcal{N}(0, 1)$



This is where the infamous $n = 30$ comes from !!

t procedures

We can calculate CIs and perform significance tests much as before (example coming up soon).

A significance test of a single sample mean using the t-statistic is called a one-sample t-test.

Collectively, the significance tests and confidence-interval based tests using the t distribution are called t procedures.

The one-sample t test

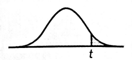
THE ONE-SAMPLE t TEST

Draw an SRS of size n from a large population having unknown mean μ. To **test the hypothesis H_0: $\mu = \mu_0$**, compute the **one-sample t statistic**

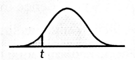
$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

In terms of a variable T having the $t(n-1)$ distribution, the P-value for a test of H_0 against

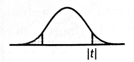
H_a: $\mu > \mu_0$ is $P(T \geq t)$



H_a: $\mu < \mu_0$ is $P(T \leq t)$



H_a: $\mu \neq \mu_0$ is $2P(T \geq |t|)$



These P-values are exact if the population distribution is Normal; they are approximately correct for large n in other cases.

A note about the conditions for *t* procedures

- B&M stress that the **first** of their conditions as *very important*: *we can regard* our data as a simple random sample (SRS) from the population

A note about the conditions for t procedures

- B&M stress that the **first** of their conditions as *very important*: *we can regard* our data as a simple random sample (SRS) from the population
- The **second**, observations from the population have a *Normal* distribution with unknown mean parameter μ and unknown standard deviation parameter σ less so
- *In practice*, inference procedures *can accommodate some deviations from the Normality condition* when the sample is large enough.

Robustness of the t procedures

A statistical procedure is said to be **robust** if it is insensitive to violations of the assumptions made.

- t procedures are not robust against *extreme* skewness, in small samples, since the procedures are based on using $\bar{y}$ and s (which are sensitive to outliers).
- Recall: Unless there is a very compelling reason (e.g. known/confirmed error in the recorded data), outliers should not be discarded.

Robustness of the *t* procedures

- *t* procedures **are** robust against other forms of non-normality and, even with considerable skew, perform well when n is large. Why?

Robustness of the t procedures

- t procedures **are** robust against other forms of non-normality and, even with considerable skew, perform well when n is large. Why?

- When n is large, s is a good estimate of σ (recall that s is unbiased and, like most estimates, precision improves with increasing sample size)

Robustness of the t procedures

- t procedures **are** robust against other forms of non-normality and, even with considerable skew, perform well when n is large. Why?

- When n is large, s is a good estimate of σ (recall that s is unbiased and, like most estimates, precision improves with increasing sample size)

- CLT: $\overline{y}$ will be Normal when n is large, even if the population data are not

When and why we use the t-distribution

- When σ is unknown use t distribution. but why?

When and why we use the t-distribution

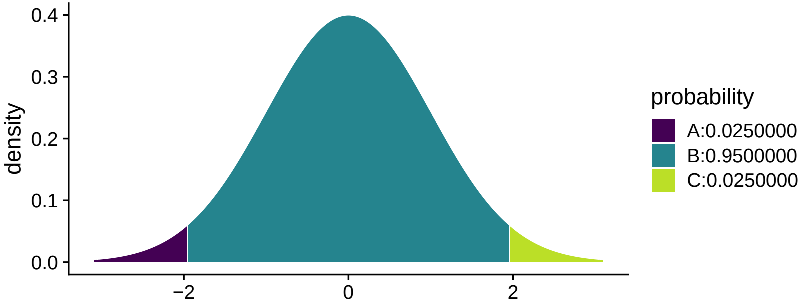
- When σ is unknown use t distribution. but why?
- the spread of the t distribution is greater than $\mathcal{N}(0, 1)$

Rejecting the Null ($H_0 : \mu = \mu_0$) when σ is known

$$\underbrace{z_{0.975}}_{\text{critical value}} = 1.96 = \frac{\bar{y} - \mu_0}{\sigma/\sqrt{n}} \rightarrow \frac{1.96}{\sqrt{n}}\sigma = \bar{y} - \mu_0$$

which means that to reject H_0 the difference between your sample mean and μ_0 needs to be greater than $\frac{1.96}{\sqrt{n}}$ standard deviations

```
mosaic::xqnorm(p = c(0.025, 0.975))
```



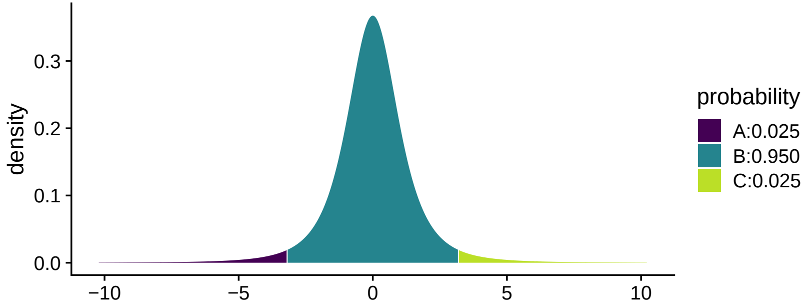

```
## [1] -1.959964  1.959964
```

Rejecting the Null ($H_0 : \mu = \mu_0$) when σ is unknown

$$\underbrace{t^{\star}_{0.975, df=3}}_{\text{critical value}} = 3.18 = \frac{\bar{y} - \mu_0}{s/\sqrt{n}} \rightarrow \frac{3.18}{\sqrt{n}} s = \bar{y} - \mu_0$$

which means that to reject H_0 the difference between your sample mean and μ_0 needs to be greater than $\frac{3.18}{\sqrt{n}}$ standard deviations

```
mosaic::xqt(p = c(0.025, 0.975), df = 3)
```




```
## [1] -3.182446  3.182446
```

Summary of t distribution

- Its harder to reject the null when using the t distribution

Summary of t distribution

- Its harder to reject the null when using the t distribution
- Confidence intervals are also wider

Summary of *t* distribution

- Its harder to reject the null when using the *t* distribution
- Confidence intervals are also wider
- This is due to our uncertainty about the estimated variance

Summary of t distribution

- Its harder to reject the null when using the t distribution
- Confidence intervals are also wider
- This is due to our uncertainty about the estimated variance
- Larger samples lead to more accurate estimates of σ

Summary of t distribution

- Its harder to reject the null when using the t distribution
- Confidence intervals are also wider
- This is due to our uncertainty about the estimated variance
- Larger samples lead to more accurate estimates of σ
- This is reflected in the fact that there is a different t distribution for each sample size

Summary of t distribution

- Its harder to reject the null when using the t distribution
- Confidence intervals are also wider
- This is due to our uncertainty about the estimated variance
- Larger samples lead to more accurate estimates of σ
- This is reflected in the fact that there is a different t distribution for each sample size
- As $n \to \infty$, sample standard deviation s gets closer to σ

Summary of t distribution

- Its harder to reject the null when using the t distribution
- Confidence intervals are also wider
- This is due to our uncertainty about the estimated variance
- Larger samples lead to more accurate estimates of σ
- This is reflected in the fact that there is a different t distribution for each sample size
- As $n \to \infty$, sample standard deviation s gets closer to σ
- As degrees of freedom increase, t distribution gets closer to Normal distribution

The t distribution

Examples

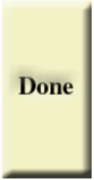
Application: How fast is your reaction time?

https://faculty.washington.edu/chudler/java/redgreen.html

RED LIGHT - GREEN LIGHT Reaction Time Test

Instructions:

1. Click the large button on the right to begin.
2. Wait for the stoplight to turn green.
3. When the stoplight turns green, click the large button quickly!
4. Click the large button again to continue to the next test.

Test Number	Reaction Time	The stoplight to watch.	The button to click.
1	0.325		
2	0.327		
3	0.357		
4	0.299		
5	0.378		
AVG.	0.3372		

Start Over

Application: How fast is your reaction time?

```r
reaction.times <- c(325,327,357,299,378)/1000
summary(reaction.times)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2990  0.3250  0.3270  0.3372  0.3570  0.3780
```

```r
round(sd(reaction.times),3)
```

```
## [1] 0.031
```

```r
length(reaction.times)
```

```
## [1] 5
```

5 ways of calculating a confidence interval

We are interested in calculating a 95% confidence interval for the mean reaction time based on the sample of 5 reaction times.

5 ways of calculating a confidence interval

We are interested in calculating a 95% confidence interval for the mean reaction time based on the sample of 5 reaction times.

Five ways of doing this:

1. By hand (using the $\pm$ formula and `R` as a calculator)
2. Using the quantile function for the t distribution `stats::qt`
3. Fitting an intercept-only regression model ($y = \beta_0 + \varepsilon$)
4. Using a canned function (`mosaic::t.test`, `stats::t.test`)
5. Bootstrap

1. By hand using the ± formula

```
n <- length(reaction.times)
SEM <- sd(reaction.times)/sqrt(n)

## [1] 0.01372734

ybar <- mean(reaction.times)

## [1] 0.3372

multiple.for.95pct <- stats::qt(p = c(0.025, 0.975), df = n-1)

## [1] -2.776445  2.776445

by_hand_CI <- ybar + multiple.for.95pct * SEM

## [1] 0.29909 0.37531
```

2. Using `stats::qt`

Note: R only provides the standard t distribution. In order to get a scaled version we must define our own function.

```
n <- length(reaction.times)
SEM <- sd(reaction.times)/sqrt(n)
ybar <- mean(reaction.times)

# scaled version of the standard t distribution
qt_ls <- function(p, df, mean, sd) qt(p = p, df = df) * sd + mean

qt_ls(p = c(0.025, 0.975), df = n - 1, mean = ybar, sd = SEM)

## [1] 0.2990868 0.3753132
```

3. Fitting an intercept-only regression model

```r
fit <- stats::lm(reaction.times ~ 1)
summary(fit)

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33720    0.01373   24.56 1.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0307 on 4 degrees of freedom

stats::confint(fit)

##                 2.5 %    97.5 %
## (Intercept) 0.2990868 0.3753132
```

3. Fitting an intercept-only regression model

In the regression output:

- `Estimate`: the mean reaction time (an estimate of the intercept β_0)
- `t value`: the test statistic
- `Std. Error`: the standard error of the mean (SEM)
- `Pr(>|t|)`: is the *p*-value

3. Fitting an intercept-only regression model

These are based on the (useless) null hypothesis $H_0 : \mu_0 = 0$

- `t value` = $\frac{\bar{y}-\mu_0}{s/\sqrt{n}} = \frac{0.33720-0}{0.01373} = 24.56$
- `Pr(>|t|)`

  = $P(\text{t value} > t_{(n-1)}) + P(-\text{t value} < t_{(n-1)})$

  = `pt(q = 24.56, df = n-1, lower.tail = FALSE)` + `pt(q = -24.56, df = n-1)`

  = $8.155 \times 10^{-6} + 8.155 \times 10^{-6} = 1.631 \times 10^{-5}$

4. Canned function

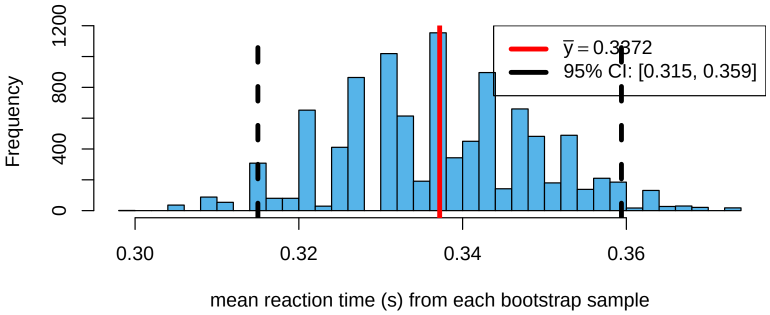
```
stats::t.test(reaction.times)

## One Sample t-test with reaction.times
## t = 24.6, df = 4, p-value = 1.63e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.299 0.375
## sample estimates:
## mean of x
##     0.337
```

5. Bootstrap

```
df_react <- data.frame(reaction.times) # need data.frame to bootstrap
B <- 10000 ; N <- nrow(df_react)
R <- replicate(B, {
  dplyr::sample_n(df_react, size = N, replace = TRUE) %>%
  dplyr::summarize(r = mean(reaction.times)) %>%
  dplyr::pull(r)
})

##  2.5% 97.5%
## 0.315 0.363
```



Summary

- We use t procedures instead of Z when we have very small samples ($n \leq 30$)
- This is because our estimate of σ is probably not accurate with such a small sample
- We account for this extra uncertainty by widening the interval $\rightarrow$ larger multiplicative factor ($t_{(n-1)} > z^{\star}$)

Summary

- We use t procedures instead of Z when we have very small samples ($n \leq 30$)
- This is because our estimate of σ is probably not accurate with such a small sample
- We account for this extra uncertainty by widening the interval $\rightarrow$ larger multiplicative factor ($t_{(n-1)} > z^{\star}$)
- Reality check: It is unlikely you will have such a small sample unless you're working with rats
- In practice you don't need to worry about t vs. Z. The software does it for you.
- However, you should still understand where the numbers are coming from and how it is being calculated. Computers aren't intelligent, they're just well trained.

Session Info

```
R version 4.0.4 (2021-02-15)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Pop!_OS 21.04

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.13.so

attached base packages:
[1] tools     stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] latex2exp_0.4.0   DT_0.16           mosaic_1.7.0      Matrix_1.3-2
 [5] mosaicData_0.20.1 ggformula_0.9.4   ggstance_0.3.4    lattice_0.20-41
 [9] kableExtra_1.2.1  socviz_1.2        gapminder_0.3.0   here_0.1
[13] NCStats_0.4.7     FSA_0.8.30        forcats_0.5.1     stringr_1.4.0
[17] dplyr_1.0.7       purrr_0.3.4       readr_1.4.0       tidyr_1.1.3
[21] tibble_3.1.3      ggplot2_3.3.5     tidyverse_1.3.0   knitr_1.33

loaded via a namespace (and not attached):
 [1] fs_1.5.0           lubridate_1.7.9    webshot_0.5.2      httr_1.4.2
 [5] rprojroot_2.0.2    backports_1.2.1    utf8_1.2.2         R6_2.5.1
 [9] DBI_1.1.1          colorspace_2.0-2   withr_2.4.2        tidyselect_1.1.1
[13] gridExtra_2.3      leaflet_2.0.3      curl_4.3.2         compiler_4.0.4
[17] cli_3.0.1          rvest_1.0.0        pacman_0.5.1       xml2_1.3.2
[21] ggdendro_0.1.22    labeling_0.4.2     mosaicCore_0.8.0   scales_1.1.1
[25] digest_0.6.27      foreign_0.8-81     rmarkdown_2.9.7    rio_0.5.16
[29] pkgconfig_2.0.3    htmltools_0.5.2    highr_0.9          dbplyr_1.4.4
[33] fastmap_1.1.0      htmlwidgets_1.5.3  rlang_0.4.11       readxl_1.3.1
[37] rstudioapi_0.13    farver_2.1.0       generics_0.1.0     jsonlite_1.7.2
[41] crosstalk_1.1.1    zip_2.2.0          car_3.0-9          magrittr_2.0.1
[45] Rcpp_1.0.7         munsell_0.5.0      fansi_0.5.0        abind_1.4-5
[49] lifecycle_1.0.0    stringi_1.7.3      carData_3.0-4      MASS_7.3-53.1
[53] plyr_1.8.6         grid_4.0.4         blob_1.2.1         ggrepel_0.8.2
[57] crayon_1.4.1       cowplot_1.1.0      haven_2.3.1        splines_4.0.4
[61] hms_1.0.0          pillar_1.6.2       reprex_0.3.0       glue_1.4.2
[65] evaluate_0.14      data.table_1.14.0  modelr_0.1.8       vctrs_0.3.8
[69] tweenr_1.0.1       cellranger_1.1.0   gtable_0.3.0       polyclip_1.10-0
[73] assertthat_0.2.1   TeachingDemos_2.12 xfun_0.25          ggforce_0.3.2
[77] openxlsx_4.1.5     broom_0.7.2        viridisLite_0.4.0  ellipsis_0.3.2
```