# PRE-Workshop Data Download file

## Table of contents



Welcome to the "*R You out of Memory Short Course*" at the 2026 useR!! confernce. We are so glad you will be joining us. In preperation for a productive learning session we have some pre work for you to do.

As a REMINDER: When you open this Quarto document (.qmd file):

1. ****Make sure it's in your project folder**** alongside your data folder and .Rproj file
2. ****If it's not in the right place:****

- Use ****File → Save As**** and navigate to your 'useR2026_bigdata_shortcourse' folder
- Or drag/move the file into your project folder using your computer's file manager

3. ****Open your .Rproj file**** to ensure RStudio is working in the correct directory
4. ****Verify everything is ready:****

Our dataset is to large to be housed in github or as a file. We can add it as a zip file but after all we are laearning how to havndle large data in R So…we will do so.

The dataset we are using will take anywhere for 8 minutes -15 minutes to download.

We want you to set up a Project folder and download it to their so lets get doing that.

Data Scientist Thinking: When working with big data, we need to consider both computational efficiency and memory management. How might these features impact our daily workflow?

Resource: Apache cookbook

```r
#Load Library
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.3.3

Warning: package 'tibble' was built under R version 4.3.3

Warning: package 'tidyr' was built under R version 4.3.3

Warning: package 'readr' was built under R version 4.3.3

Warning: package 'purrr' was built under R version 4.3.3

Warning: package 'dplyr' was built under R version 4.3.3

Warning: package 'stringr' was built under R version 4.3.3

Warning: package 'forcats' was built under R version 4.3.3

Warning: package 'lubridate' was built under R version 4.3.3

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

## Downloading the Data

A dataset of item checkouts from Seattle public libraries, available online at data.seattle.gov/Community/Checko
by-Title/tmmm-ytt6.

### Step 1: Create a Directory

First, let's create a special folder to store our data:

```
# Create a "data" directory if it doesn't exist already
# Using showWarnings = FALSE to suppress warning if directory already exists

dir.create("data", showWarnings = FALSE)
```

### Step 2: Download the Dataset

Now for the fun part! We'll download the Seattle Library dataset (9GB).

**Important:** This is a 9GB file, so:

- Make sure you have enough disk space

```
# Download Seattle library checkout dataset:
# 1. Fetch data from AWS S3 bucket URL
# 2. Save to local data directory
# 3. Use resume = TRUE to allow continuing interrupted downloads

curl::multi_download(
  "https://r4ds.s3.us-west-2.amazonaws.com/seattle-library-checkouts.csv",
  "data/seattle-library-checkouts.csv",
  resume = TRUE
)
```

Why USE: `curl::multi_download()`

- Shows a progress bar (great for tracking large downloads)
- Can resume if interrupted (super helpful for big files!)
- More reliable than base R download methods

Resource: CURL

### While the file downloads, let's think about:

1. Why do we need special tools for such large datasets?
2. What challenges might we face with traditional R methods?
3. How might a library use this kind of data?

We will discuss these questions and more when we meet at the short-course.

### Step 3: Verify the Download

After the download completes, let's make sure everything worked:

```
# Check if the Seattle library dataset file exists and print its size:
# 1. Verify file exists at specified path
# 2. Calculate file size in gigabytes by dividing bytes by 1024^3

file.exists("data/seattle-library-checkouts.csv")
```

```
[1] FALSE
```

```
file.size("data/seattle-library-checkouts.csv") / 1024^3  # Size in GB
```

```
[1] NA
```