Challenging the Link Between Early Childhood Television Exposure and Later Attention Problems: A Multiverse Approach

Matthew T. McBee, Rebecca Brand, and Wallace Dixon

Supplementary Materials

We used R version 3.6.3 (R Core Team, 2020) for data manipulation, analysis, figure generation, reporting, and automation. The *documentation* component of our project's Github page[1] presents a spreadsheet mapping our analysis variables to the variable codes and labels from the NLSY dataset. Our raw and processed analysis datasets as well as our analysis code are disclosed on this site, allowing interested readers to replicate or extend our analysis. We recommend downloading ("cloning") the repository and viewing the files locally, as html tables will not render in the browser when examining the results online. Our code and computational environment is also preserved as a Docker compute container[2], which archives the entire software toolchain (operating system + R + packages + analysis script) in a virtual machine, hedging against the possibility that future updates to any of the software components could break the computational reproducibility of our analysis.

Why Were Some Models Significant? We performed some 'post-mortem' analyses to better understand why some of these models detected a relationship between variables. Our explanation is that the nonlinear "wiggle" in the scatterplots displayed in Figure 1 can trigger significance if it is brought into sharp relief by the analysis.

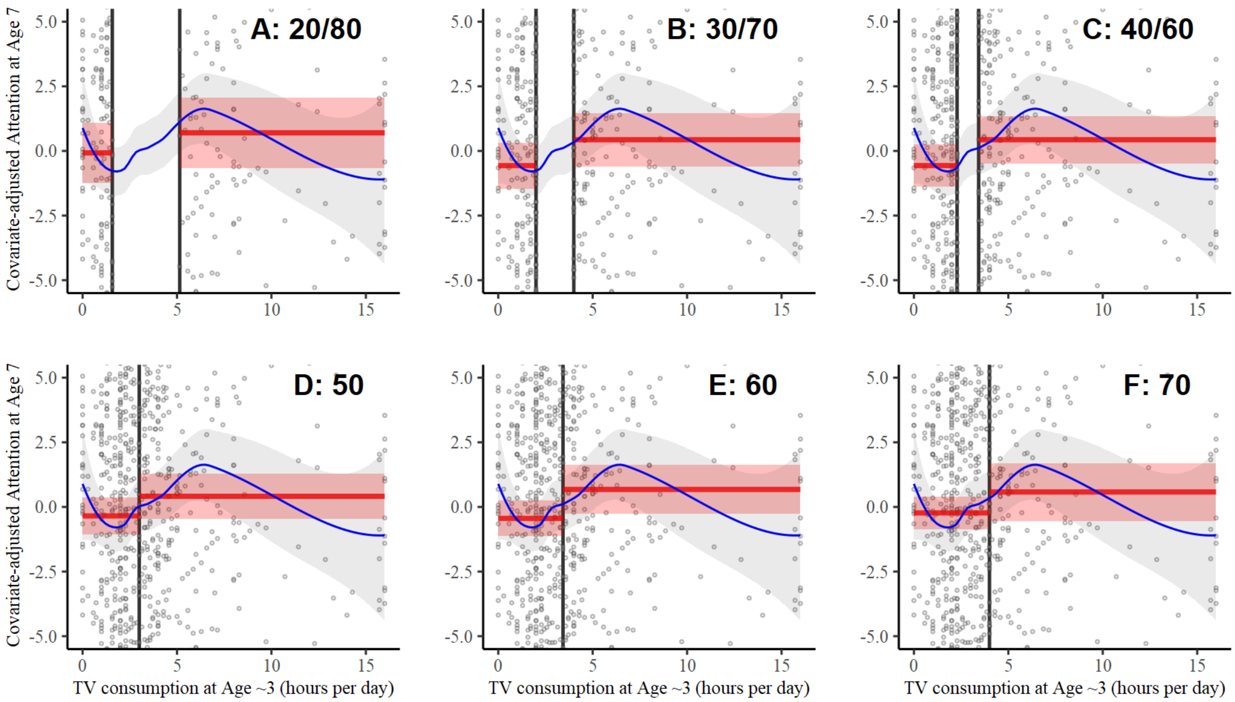
Our argument rests on a few observations. As Figure 1 from the manuscript indicates (comparing left panels to right panels), this feature is visually more pronounced when TV is measured at age ~3 than at age ~1.5. Over both multiverse analyses, only 19/424 (4.5%) of the age ~1.5 models were significant. compared to 147/424 (34.7%) of the age ~3 models. Furthermore, panels C and D of the figure illustrates how this "wiggle" is visually diminished by imputation of missing data; an unconfounded comparison reveals that 19/72 (26.4%) of models using listwise deletion were significant compared to 12/80 (15.0%) of those using multiple imputation.

The pattern of results in the IPTW propensity score models revealed that the likelihood of significance varied across TV cutpoints, as shown by Table 4 in the manuscript. The highest significance rates occurred at 50th and 60th percentile cutpoints (2.86 and 3.43 hours of TV per day, respectively) for the models in which TV was measured at age ~3. This is consistent with our hypothesis. Supplementary Figure 1 displays a magnified view of the nonlinear 'wiggle'

[1] https://github.com/mcbeem/TVAttention
[2] https://hub.docker.com/repository/docker/mmcbee/rstudio_tvattention

and indicates how the various TV percentile cutpoints for these models aligned with this feature. The 50th and 60th percentile cutpoints, which had the highest significance rates, placed the dividing line between low and high TV groups almost precisely in the center of this nonlinear feature of the data, and resulted in the largest precision-weighted difference in the means between these groups.

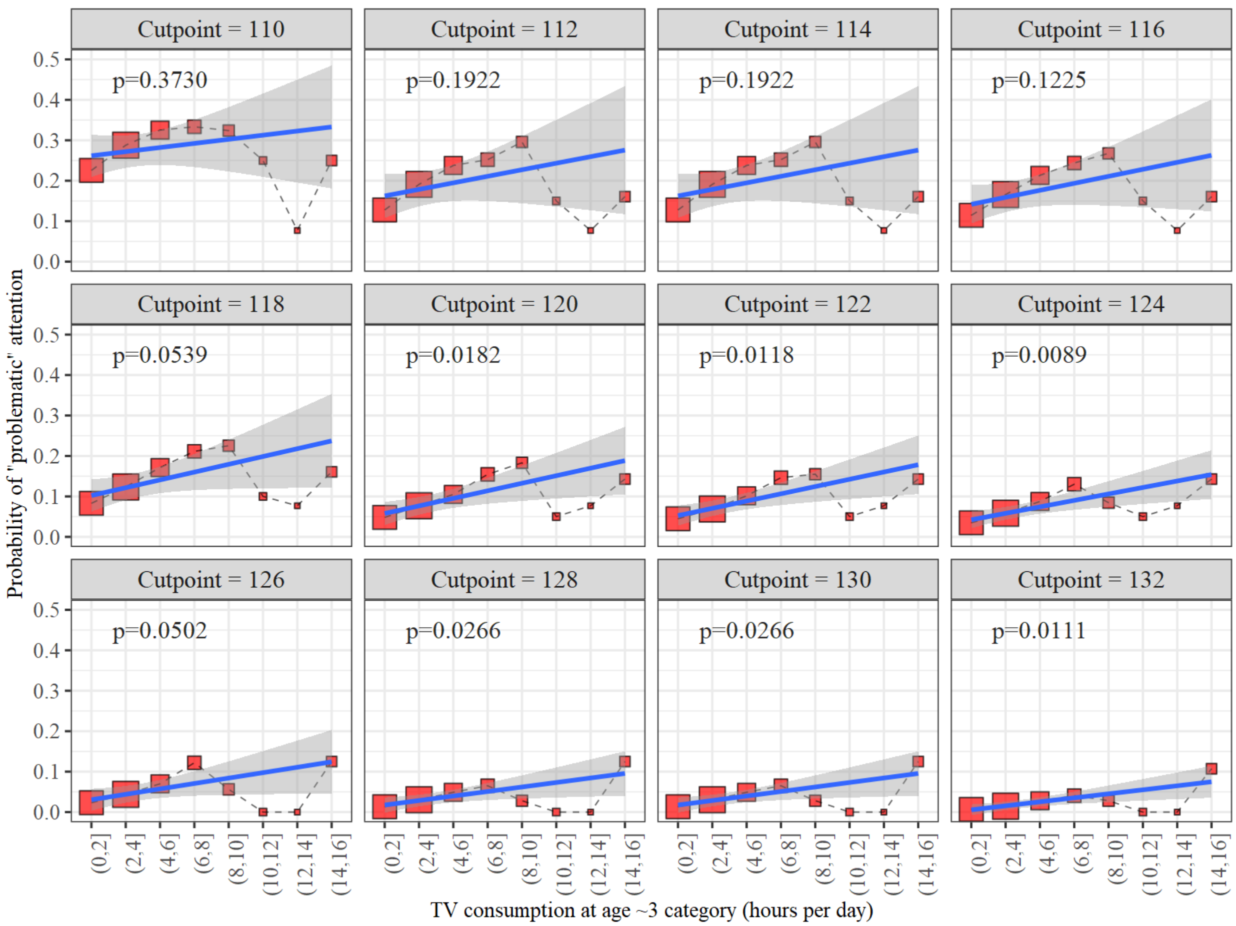
Supplementary Figure 1
IPTW propensity score model post-mortem



Note: Panels display the zoomed-in residualized attention outcome (e.g., controlling for covariates) versus TV measured at age ~3. Each panel depicts a different set of TV percentile cutpoints for defining the low- and high-TV groups; which are displayed as dark vertical lines and in the label for each panel. The blue curve on each plot is the loess smoother, illustrating the location and scale of the nonlinear feature. The red horizontal lines and their shaded uncertainty regions display the conditional mean attention in the low- and high-TV categories.

The results of the logistic regression models in multiverse I also support our hypothesis. As shown in the manuscript's Figures 2 and 3, the significance of these models was strongly related to the attention cutpoint defining the 'normal' and 'problematic' attention groups. We believe that higher cutpoints in these models magnify the nonlinear feature of the data to make it more consistent with a TV-attention relationship. Supplementary Figure 2 plots the proportion of cases in the 'problematic' attention category by TV use at age 3 (which has been categorized into bins to permit rate calculations) for twelve different attention cutpoints. The nonlinearity can be easily observed in the pattern of dots in the upper left panel. Each panel of the figure displays a fitted linear regression line and shaded confidence interval, which roughly represents the performance of a logistic regression model using that cutpoint. The *p*-value for the slope coefficient of those regression lines is displayed in each panel. At low cutoffs for defining problematic attention, the nonlinear configuration of points reduces the slope of the fitted line and, more importantly, adds uncertainty regarding the slope, increasing the *p*-value for the test. As the cutoff defining problematic attention increases, the base rate of attention 'problems' is reduced, and the points migrate downward. This alters the pattern of points, making them more consistent with a linear trend. As the cutpoint surpasses 120, the *p*-value for the slope becomes significant and remains so through the highest cutpoint examined. This pattern is consistent with the results of the logistic models.

Supplementary Figure 2
Logistic regression post-mortem



Note: Panels display the probability of impaired attention (*y*-axis), as defined by the attention cutpoint displayed on each panel, versus TV measured at age ~3 (*x*-axis). The size of each point is proportional to the number of cases in that level of TV consumption. A fitted regression line (weighted by the sample size within each TV grouping) and its slope *p*-value are depicted on each plot; these represent the performance of the logistic regression model in each situation. As the cutoff rises, the association between TV and the probability of impaired attention seems to increase.

Appendix
Guide for reproducing this analysis using the Docker image

The Docker container is preloaded with the versions of R, RStudio, and all the R packages that were used to perform the analyses reported in this paper.

1) Make a Docker account at http://www.docker.com
2) Log in and download Docker Desktop. If prompted, make sure Docker is set to use Linux containers.
3) Start Docker Desktop, logging in to your account. Under Docker Desktop – Settings – Advanced, make sure that the Docker Engine can use at least 6 GB of memory. Insufficient resources can cause the code to hang. Under Settings – Shared Drives, grant access to one of your local drives so you can copy the generated files out of the container to your local machine.
4) Open a Terminal (Mac / Linux) or Command Prompt (Windows)
5) Type `docker run --rm -e PASSWORD=TV -p 8787:8787 mmcbee/rstudio_tvattention:psychscience`

   (If this image isn't found on the local machine, it will be downloaded automatically from Docker Hub)
6) Open a browser tab and navigate to this url: localhost:8787

   RStudio will begin running in your browser.
7) Log in to RStudio with username rstudio, password TV
8) In RStudio, open the file /Code/analysis.r in the Files pane
9) Run the code by highlighting it all (Ctrl-A or Cmd-A) and then Ctrl-Enter or Cmd-Enter (Note: it will take several hours to run).
10) Inspect the results in the /Results and /Manuscript/Tables and /Manuscript/Figures folders.
11) End the Docker session by pressing Ctrl-C or in the terminal or command prompt window. On a Mac, this will end the Docker session. On Windows machines you'll need to determine either the Container ID or the Name of the session by typing `docker ps`

The Container Id is 12-character string such as b1971e3eea21. It will be different each time you run the container. Next, type `docker stop CONTAINERID`, for example, `docker stop b1971e3eea21`. Alternatively, you can refer to the container by its name, which will be a combination of random words such as priceless_galois or competent_ellis. The container's name and id are shown by `docker ps`

Copying files from the Docker container to your local machine

After the analysis script is finished running, you will likely want to copy the files to your local computer for examination. If you stop the Docker container, you will have to run the analysis again to recreate all the files.

1) Determine the Container Id for the Docker session by typing `docker ps`. On a Mac you will need to open a new Terminal window, as pressing Ctrl-C in the active Terminal window will end the Docker session. On a Windows machine, pressing Ctrl-C will allow you to enter additional commands in the active Command Prompt window without disturbing the Docker session.
2) Change the directory in your Command Prompt / Terminal session to the local directory to which you want to copy the files with the `cd` command. For example, `cd "c:\Users\Matt\Documents\TVAttention"` (Windows) or `cd "/home/Users/Matt/Documents"` (Mac)
3) Copy the files by typing `docker cp CONTAINERID:home/rstudio .` For example, `docker cp b1971e3eea21:home/rstudio .` (You can substitute the container name for the Container id).