# Reproducibility Report for: The Forecasting Collaborative. Insights into the accuracy of social scientists' forecasts of societal change

Ana Martinovici, Ekaterina Pronizius, Erin M. Buchanan (order in report)

Last compiled on 2024-12-11 20:28:12.443407

For the original version of our pre-re-analysis plan, see Link to the original re-analysis plan. In this report, original wording of our re-analysis plan is included in highlighted **orange** infoboxes, and the original wording of the pre-registration from the authors is present in **blue** infoboxes. Note that infoboxes are displayed only in the html output (not in pdf).

The article aims to answer "How well can social scientists predict societal change, and what processes underlie their predictions?" (abstract, page 484). Our robustness reproduction will focus on a subset of the analyses reported in the paper. Specifically, we will focus on results and claims reported in the section "How accurate were behavioural and social scientists at forecasting?" (pages 485-486). For this section, we plan to:

## Computational Reproducibility

1. check computational reproducibility using the data and code provided by the authors. We expect that we have to make changes in order to execute the analysis code, and we aim to many as few changes as possible.

We plan to spend at most 12 hours on this step (3 sessions of max 4 hours, spread over several days). Within the team we have extensive experience with R, statistics, and good practices for computational reproducibility. We think that 12 hours of work is a very high amount of effort, and ideally readers should be able to reproduce results in a much shorter amount of time (<2 hours). We will report the amount of time that we spend and the changes we have to make to the existing code.

### Summary

The results were not computationally reproducible within the planned time. Despite spending more time than initially planned (aprox. 20 hours for Ana), we were unable to reproduce the results. This is due to how data, code, and instructions are provided by the authors in the reproducibility package. We list below specific issues and then recommendations for how to address them:

### Issues

- the README file in the repository provides incomplete information about the content of the repo.

The repository contains 124 files: 44 in the root directory and the rest in one of 5 subdirectories. Only a subset are mentioned in the README file. Only some of the files that seem to contain data (based on filename) are described in the README file (e.g., `Wave1+2data.csv`, `Wave1+2demographics.csv`, `contrast1.csv` are not described)

- some files are duplicates (i.e., have the same name and same content). It is unclear why and which one should be used. For example: `BenchmarkSimulations.R` is both in the root directory and the `sim` directory.
- some files have the same name yet different content

`dat_long.csv` is both in the root directory (V1) and the `Data Cleaning` directory (V2). The README file states: *"`dat_long.csv` contains a file with predictions of forecasting teams in a long format, used for plotting estimates of each team."* I have checked, and the two files do not have the same content. V1 has 26960 observations and 144 variables, while V2 has 27032 observations and 134 variables. The two datasets share 131 variables. I checked if there are differences in the observations for these shared 131 variables: of the 26960 observations in V1, 477 observations match V2. Given the size (observations and variables) of these datasets, and the limited information available in the repo, it is not feasible to conduct further checks. Thus, we cannot state if the mismatch is due to a minor issue (e.g., numerical precision) or major issues (e.g., different values used in the analysis than the ones provided by participants).

`Wave 1+2 Descriptives.Rmd` saves data to `dat_long.csv`:

```
file_name <- here::here("original_files", "Wave 1+2 Descriptives.Rmd")
start_line <- 239
end_line <- 240
knitr::read_chunk(path = file_name, labels = "my-query-params", from = start_line, to = end_line)

file_in_repo <- stringr::str_sub(file_name, start = stringr::str_locate(file_name, "/repro")[1, "start"]
```

Specifically, lines 239:240 from /repro-Forecasting_Tournament/original_files/Wave 1+2 Descriptives.Rmd:

```
write.csv(dat_long,"dat_long.csv")
```

It is not immediately clear which of the two files, if any, is generated by the Rmd - this depends on the knit directory settings of the user who last knitted this file. The Rmd file contains code that sets a working directory to `setwd("~/GitHub/Forecasting-Tournament") #igor's working directory`. Based on this, it is more likely that the csv file is saved in the root directory. After commenting out the `setwd` command, I tried to knit the file. The file could then be knit without error and two files were generated `dat_for_analyses.csv` and `dat_long.csv`. I will refer to the `dat_long.csv` generated after I knitted the Rmd as V3. V3 has 26960 observations (same as V1) and 138 variables (different from either V1 or V2). Thus, neither of the two files was generated by `Wave 1+2 Descriptives.Rmd`. It is thus unclear how the dataset that contains predictions of forecasting teams was generated.

This is a major issue that severely reduces usefulness of further checks of computational reproducibility. Even if we were to assume that the results in the paper can be reproduced using one of the two data files but not using the other. Without knowing which of the datasets is the correct one to use, we wouldn't be able to make any meaningful statements about reproducing the results.

We also noticed that the shared datasets contain information (e.g., email addresses of participants, IPAddresses, and geolocation data) that probably should not be made publicly available.

- it is unclear which file needs to be used to produce results reported in the paper. There seem to be 3 options: 2 based on the names of the files and a third based on the information in the README file:
  - `Wave 1+2 Analyses FINAL FOR MANUSCRIPT.Rmd`

The file cannot be knit without error, even after installing all packages and commenting out the `setwd` line of code. "Quitting from lines 1149-1372 [Phases 1 and 2 along with sims] (Wave-1+2-Analyses-FINAL-FOR-MANUSCRIPT.Rmd)". I have checked these lines of code, and there is a mistake in how arguments to the `stan_lmer` function are provided. A parenthesis is missing, which means that this Rmd file was not knitted by the authors, so probably this is not the file that was used to generate the results. Trying to knit the file, overwrote 28 of the 29 files in `Wave-1+2-Analyses-FINAL-FOR-MANUSCRIPT_files\figure-html`. These are all image files for plots. Some of these plots are substantively different (i.e., the original version shared by authors is different from what is generated when knitting the Rmd file). These differences are important (e.g., different number of variables being plotted) and cannot be explained by differences in resolution or other display settings between the device of the authors and the device of the user trying to reproduce results. In conclusion, this version of the file could not have been used to generate the results in the paper.

- `Wave 1+2 Analyses update.Rmd`

Next, I tried to knit this file. After commenting out the `setwd` line, the file could not be knit without error. I summarise below the steps I took to fix these errors.

"Quitting from lines 747-907 [Phases 1 and 2 along with sims] (Wave-1+2-Analyses-update.Rmd) Error in `initializePtr()`: ! function 'cholmod_factor_ldetA' not provided by package 'Matrix' "

This seems to be due to an issue related to the Matrix version (https://stackoverflow.com/questions/77481539/error-in-initializeptr-function-cholmod-factor-ldeta-not-provided-by-pack). I have followed the suggestion and removed the Matrix package and then install the binary version. "Warning in install.packages : package 'Matrix' is not available for this version of R" My version of R is 4.3.2 Unclear which version of R and R packages were used by the authors, so it is not feasible to try and guess it by trial and error (the analyses files load more than 20 packages - unclear how many of these are in fact used). I have followed the second suggestion in the Stackoverflow post and uninstalled `lme4` and then installed it again from source. "Quitting from lines 1119-1156 [unnamed-chunk-3] (Wave-1+2-Analyses-update.Rmd) Error: ! object 'match_fonts' is not exported by 'namespace:systemfonts' "

This is most likely related to using `newggslopegraph` from the package `CGPfunctions`. I commented out the call of this function and tried to knit again.

"Quitting from lines 1258-1303 [inaccuracy on odd and even month - stability of inaccuracy] (Wave-1+2-Analyses-update.Rmd) Error in `parse()`: ! :5:71: unexpected '::' " related to `dplyr::dplyr::select`. This version of the file could not have been knitted without error on the device of the authors. I fixed the mistake (two instances in the same code chunk) and tried to knit again.

The file makes uses of R packages that are not listed at the start of the file. This means that knitting stops multiple times due to missing packages. Ideally, the list of necessary packages and their version would be shared. The file contains 162 instances where `lmer` was used. Unclear which of these 162 estimated models are reported in the paper.

"Quitting from lines 1511-1593 [unnamed-chunk-6] (Wave-1+2-Analyses-update.Rmd) Error in `assert_package()`: ! `quick_docx` requires the"flextable" package. To install, type: install.packages("flextable")"

Despite following the instruction in the error message and installing the package, the file cannot be knit without error. I have then executed all code before this chunk and then tried to execute the code within the chunk line by line to figure out which line of code generates the error. After executing the code above, the environment has 124 objects. This is only half way through the code. The very large number of objects makes it error prone - easy to make one spelling error and get other results than intended. I commented out the lines of code that produced the previous error and tried to knit again.

"Quitting from lines 1692-1856 [SUPPLEMENTARY PHASE 1 analyses] (Wave-1+2-Analyses-update.Rmd) Error in `lme4::lFormula()`: ! 0 (non-NA) cases" This error seems to suggest there is a problem with the data used for analysis. After one hour of trying to knit the file, it is unfortunately not feasible to go through the code line by line and try to figure out what the problem is.

Up to line 1692 where the knitting stopped, this Rmd generates a few csv files that are saved in the repo:

- wave1.scores.csv
- wave2.scores.csv
- final.results.csv
- top.t1.csv
- medianMASE.t1.csv
- top.t2.csv
- medianMASE.t2.csv
- contrast1.csv
- contrast2.csv

I checked and the first 7 are reproduced. The last 2 are not.

- The third potential file to produce results is based on the README file

This points to a previous version of `Wave 1+2 Analyses FINAL FOR MANUSCRIPT.Rmd` as the one used to generate the results. The code in this version is different from the most recent one in the repo.

In summary, the Rmd files were probably not knited by the authors. This is based on the fact that multiple Rmd files contain code that generates errors that would not allow knitting on any device. For example, `Wave1+2_Merge_2021-07-14.Rmd` has duplicate chunk labels ("Duplicate chunk label 'add historic data to data frame' "). Going through all the files, finding and then fixing all errors is not feasible. I estimate that it could take several weeks of full time work, and it is uncertain if in the end the results will be reproduced. There is a very low probability of computational reproducibility, based on how the code and files are structured. There doesn't seem to be a consistent coding style used, which increases the probability of making mistakes and lowers the probability of detecting these mistakes. long lines of code that span multiple lines on screen or that require horizontal scrolling, inconsistent spacing between operators and objects, mix of tidyverse (dplyr) and R base functions to process data, partial matching, accessing observations by using manually input row indices rather than filtering on explicit criteria (`# baselines for participant phase 1 submissions baseline_1 <- dat_hist[nrow(dat_hist) - 12, 1:ncol(dat_hist)]`), code repetition and manual changes rather than using functions, no testing. All these issues increase the probability of mistakes in the code.

## Recommendations

The current structure and content of the repo show a variety of error prone practices. To clarify, error prone doesn't mean that errors are guaranteed to exist in the code. It simply means that the user needs to put in a lot more effort to avoid mistakes. We know that people have limited time, attention, and working memory. To prevent mistakes, it is best to use practices that take these limits into account. The principle is called defensive programing and is similar to the idea of defensive driving. The following recommendations aim to lower the probability of mistakes in the code, and make it easier to notice mistakes when they appear.

1. Use a consistent code style

   spa cinga ndpun ctuationareimp ortantbo thwh enwritingtex tan dwhenwritingcode

   Spacing and punctuation are important both when writing text and when writing code.

The two lines of text above contain the same letters. Yet, one can be read much easier, while the other could easily be mistaken for random characters. Writing code in a way that makes it easier for another person to read it lowers the probability of mistakes and increases the chances that someone else can verify the results. The current version of the code in the repo would benefit from use of spacing and indentation to improve readability. For specific suggestions and additional examples, see the `tidyverse` style guide (https://style.tidyverse.org/) or any other similar resource on style guides for programming.

2. Use R projects and the `here` package

See: https://www.tidyverse.org/blog/2017/12/workflow-vs-script/

3. The repo should contain ALL the files and NOTHING BUT the files needed to reproduce the results

It is easy to see why including ALL the files is needed. Including NOTHING BUT the files that are needed is an error reduction measure. The more unnecessary files there are, the more difficult it is to keep track of which ones should be used, and the more likely it is to make mistakes.

4. R scripts and Rmd files should contain ALL the code and NOTHING BUT the code needed for a specific goal

For example, `Wave 1+2 descriptives.Rmd` contains lines of code that are commented out. If something is not needed, then it is best removed. This way, fewer lines need to be checked for correctness and those lines can more easily be read and understood compared to when presented in a cluttered file.

5. If you use Rmd, then `knit` the file to ensure that it can be knitted without error.

While developing the code, if you want to test a minor change (e.g., changing labels in a plot), then you might execute the code and check the output in the console (i.e., without knitting the file). If you do this,

then first restart the R session and then execute all the code before the section you are working on before executing the lines of code that you are trying to change. When you are done with all changes, knit the file and make sure it knits without error.

6. Share a version of the data that is as close as possible to the raw data, and scripts that process it.

The raw dataset might contain information that cannot be publicly shared due to ethical and legal considerations. Depending on the specifics of each situation, this type of information should be pseudonimized, removed, or otherwise transformed before publicly sharing a dataset.

For additional recommendations see:

Bryan, J. (2018). Excuse Me, Do You Have a Moment to Talk About Version Control? The American Statistician, 72(1), 20–27. https://doi.org/10.1080/00031305.2017.1399928

Ellis, S. E., & Leek, J. T. (2018). How to Share Data for Collaboration. The American Statistician, 72(1), 53–57. https://doi.org/10.1080/00031305.2017.1375987

Thomas, D., & Hunt, A. (2019). The Pragmatic Programmer: your journey to mastery. Addison-Wesley Professional.

Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, Teal TK (2017) Good enough practices in scientific computing. PLoS Comput Biol 13(6): e1005510. https://doi.org/10.1371/journal.pcbi.1005510

Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. (2014) Best Practices for Scientific Computing. PLoS Biol 12(1): e1001745. https://doi.org/10.1371/journal.pbio.1001745

# Preregistration

2. check the match between the preregistration plan of the paper (submitted by the authors) and the final paper

## Summary

When comparing the manuscript against the preregistration, we used this version of the preregistration: https://osf.io/u9x4m Note that there are more preregistration documents, as explained in the infobox below.

Overall, the main manuscript follows the plan outlined in the preregistration. In the main manuscript, the authors transparently describe the deviations and clearly distinguish between explanatory and confirmatory analyses in their manuscript. Yet, it is unclear what was the rationale for extending the recruitment of teams and the general sample, which resulted in much larger targeted sample sizes? The other deviations are rather minor, as detailed below.

From the authors' response to reviewers:

The pre-registered plan was previously submitted to NHB (NATHUMBEHAV-200410305PI) as an inquiry in April 2020; you can find this submitted plan here https://osf.io/7ekfm. Note, it is not a conventional pre-registration. We initially aimed to present the manuscript as a pre-registered report which we submitted to the journal. We subsequently officially pre-registered our methods on September 11, 2020 https://osf.io/u9x4m. The exact copy of this report is uploaded on OSF, but the time stamp suggests a later date (in the chaos of the first COVID lockdown, we failed to upload it to the OSF and subsequently shifted focus to the actual tournament, with reports on GitHub). We have now harmonized the GitHub and OSF parts of the project, so all aspects of the project can be found together. We also added the direct link to the pre-registered plan document, and moved the section about deviations from the pre-registration to the front of the revised methods section.

In the revision, we have restructured the manuscript (initial submission was directly forwarded from Nature, and methods were in the supplementary materials), highlighting the pre-registration section at the beginning of the revised Method section. Our pre-registration included:

- key research questions;
- data processing;
- locking-in forecasts of participating teams;
- use of key metric (MASE) for evaluating performance across domains,
- naive benchmarks (e.g., simple interpolation algorithms);
- comparison of forecasting approaches;
- examination of opportunity to update forecasts; and
- types of covariates we consider in analysis of exogenous variables that may enhance accuracy (e.g., confidence, conditional factors and counterfactuals, number of team members, disciplinary diversity).
- We also pre-registered data analytic procedures, including how we categorized forecasts in terms of method, categorization of additional parameters in the model, teams, and update justifications (https://osf.io/u9x4m).
- In addition, we pre-registered comparisons against naïve benchmarks (naïve model in forecasting literature is used synonymously with a random walk;
- we also included historical mean as another frequently mentioned naïve method).
- Further, we pre-registered a two-tailed comparison of MASE scores across forecasting types (purely theoretical, purely data-driven and hybrid models) in linear mixed models (MASE scores nested in teams), and a contrast of theory-free models to theory-inclusive models and use of post-hoc pairwise tests for evaluating accuracy.

We did not pre-register use of a lay crowd sample prior to collecting their forecasts in June 2020 (but we did pre-register this sample in early September, 2020, prior to cleaning or evaluating their data) and we deviated from the pre-registration in testing all individual predictors (e.g., team characteristics, model simplicity, number of parameters in the data model) simultaneously, instead of performing separate analyses. We explain the above in the relevant section of the revised methods section, and refer readers to the pre-registration plan we initially submitted to Nature Human Behavior for review in May 2020, and which is posted on the Open Science Framework: "Pre-registration and deviations. Forecasts of all participating teams along with their rationales were pre-registered on Open Science Framework (https://osf.io/u9x4m).

Additionally, in an a priori specific document shared with the journal in May 2020, we outlined the operationalization of the key dependent variable (MASE), operationalization of covariates and benchmarks (i.e., use of naive forecasting methods), along with the key analytic procedures (linear mixed model and contrasts being different forecasting approaches).

## Study Information

### Hypotheses

This study asks 3 main questions:

1. How good are behavioral and social scientists at forecasting the social consequences of a COVID-19 pandemic? Following established procedures, we will examine the absolute percentage deviation for each forecast, and mean absolute scaled error (MASE) within and across forecasted time-points and social issues. MASE compares forecasted values against those obtained via a one-step "naïve forecast method." It is independent of the scale of the data and can be used to compare forecasts across datasets with different scale, is asymptotically normal and easy to interpret, with lowest MASE scores indicating greatest forecasting accuracy. Critically, we will compare forecasting accuracy of scientists' predictions against basic interpolation algorithms (e.g., moving average models with different lags). We will also compare the stability of model accuracy measured for different subsets of time, to assess the extent to which models might be accurate simply by chance.

This part of the preregistration aligns with the main manuscript. Note, as transparently mentioned in the manuscript, *due to scale differences between domains, [the authors] chose not to feature analyses concerning absolute percentage errors of each time point in the main paper.*

2. Are some societal shifts in response to the COVID-19 pandemic easier to accurately forecast than others (e.g., is it easier to accurately forecast changes in prejudice toward outgroups vs. well-being vs. shifts

in political preferences)? We will examine overall forecasting accuracy, and stability of forecasting accuracy, across domains above and beyond the naïve forecasting method using MASE.

This part of the preregistration aligns with the main manuscript.

3. Are there characteristics (discipline, methodological approach to forecasts) of some teams that lead to more accurate forecasts in social science domains? Here, we focus on comparisons of forecasting approaches relying on a. pure expertise (but no data modeling); b. pure modeling (but no consideration of expert theories); c. hybrid approaches. We will explicitly examine the reasoning process, evaluating the role of confidence, and quality of forecast rationales (e.g., consideration of conditional factors and counterfactuals) for forecasting accuracy.

This part of the preregistration aligns with the main manuscript.

## Design Plan

### Study type

Observational Study - Data is collected from study subjects that are not randomly assigned to a treatment. This includes surveys, "natural experiments," and regression discontinuity designs.

### Blinding

No blinding is involved in this study.

### Study design

The behavioral science forecasting collaborative is based on a common task framework in which 39 months of data were collected for 10 different domains: life satisfaction, affect (positive & negative), ideology (Democrat & Republican), Polarization, Asian-American implicit bias, Asian-American explicit bias, African-American implicit bias, African-American explicit bias, gender-career implicit bias, gender-career explicit bias.

Participants were given access to this data prior to participation and were asked to submit predictions for one or more domains (see https://predictions.uwaterloo.ca/datasets/).

In the main manuscript, there are 12 domains, consistent with the preregistration, though the numbering differs slightly. The exact months of historical data for Tournament 1 (January 2017 to March 2020) are provided below, where the domains are operationalized. However, we couldn't find specific information regarding the exact months of data for Tournament 2 in the preregistration.

Data for each domain was generated through the following processes:

*Affective well-being and life satisfaction.*

We used monthly Twitter data to estimate markers of affective well-being (positive and negative affect) and life satisfaction over time. We rely on Twitter because no polling data for monthly well-being over the required time period exists, and because prior work suggests that national estimates obtained via social media language can reliably track subjective well-being (Luhmann, 2017). For each month, we used previously validated predictive models of well-being, as measured by affective well-being and life satisfaction scales (Schwartz et al., 2016). Affective well-being was calculated by applying a custom lexicon (Kiritchenko, Zhu, & Mohammad, 2014) to message unigrams; life satisfaction was estimated using a ridge regression model trained on latent Dirichlet allocation topics, selected using univariate feature selection and dimensionally reduced using randomized principal component analysis, to predict Cantril ladder life satisfaction scores. Such twitter-based estimates closely follow nationally representative polls (Witters, & Harter, 2020). We applied the respective models to Twitter data from January 2017 to March 2020 to obtain estimates of affective well-being and life satisfaction via language on social media.

The operationalization of this variable is identical to the one provided in the main manuscript.

*Ideological Preferences.*

We approximated monthly ideological preferences via aggregated weighted data from the Congressional Generic Ballot polls conducted between January 2017 and March 2020 (projects.fivethirtyeight.com/congress-generic-ballot-polls), which ask representative samples of Americans to indicate which party they would support in an election. We weighted polls based on FiveThirtyEight pollster ratings, poll sample size, and poll frequency. FiveThirtyEight pollster ratings are determined by their historical accuracy in forecasting elections since 1998, participation in professional initiatives that seek to increase disclosure and enforce industry best practices and inclusion of live-caller surveys to cellphones and landlines. Based on this data, we then estimated monthly averages for support of Democrat and Republican parties across pollsters (e.g., Marist College, NBC/Wall Street Journal, CNN, YouGov/Economist).

The operationalization of this variable is identical to the one provided in the main manuscript.

*Political Polarization.*

We assessed political polarization by examining differences in presidential approval ratings by party identification from Gallup polls (https://news.gallup.com/poll/203198/presidential-approval-ratings-donald-trump.aspx). We obtained a difference score in % of Republican versus Democrat approval ratings and estimated monthly averages for the time period of interest. The absolute value of the difference score will ensure possible change after 2020 Presidential election will not change the direction of the estimate.

The operationalization of this variable is identical to the one provided in the main manuscript.

*Explicit and Implicit Bias.*

1. Given the natural history of the COVID-19 pandemic, we sought to examine forecasted bias in attitudes towards Asian-American (vs. European-Americans).
2. To further probe racial bias, we sought to examine forecasted racial bias in preferences for African-American (versus European-American) people.
3. Finally, we sought to examine gender bias in associations of female (vs. male) gender with family versus career.

For each domain we sought to obtain both reliable estimates of explicit attitudes (Axt, 2018) and estimates of implicit attitudes (Nosek, 2007). To this end, we obtained data from the Project Implicit website (http://implicit.harvard.edu) which has collected continuous data concerning explicit stereotypes and implicit associations from a heterogeneous pool of volunteers (50,000 - 6,000 unique tests on each of these categories per month). Further details about the website and test materials, are publicly available at https://osf.io/t4bnj. Recent work suggests that Project Implicit data can provide reliable societal estimates of consequential outcomes (Hehman, Flake, & Calanchini, 2018; Ofosu, Chambers, Chen, & Hehman, 2019) and when studying cross-temporal societal shifts in U.S. attitudes (Charlesworth & Banaji, 2019). Despite the non-representative nature of the Project Implicit data, recent analyses suggest that bias scores captured by Project Implicit are highly correlated with nationally representative estimates of explicit bias, r = .75, indicating that group aggregates of the bias data from Project Implicit can reliably approximate group-level estimates (Ofosu, Chambers, Chen, & Hehman, 2019). To further correct possible non-representativeness, we applied stratified weighting to the estimates, as described below.

For explicit attitude scores, participants provided ratings on feeling thermometers towards Asian-Americans and European Americans (to assess Asian-American bias), and White and Black Americans (to assess racial bias). For explicit bias in the Gender – Career task, participants rated the extent to which they associated career with male or female (from Strongly Female to Strongly Male) and then used the same scale to rate the extent to which they associated family with male or female. Relative explicit bias was then calculated as the difference in responses to minority and majority groups on feeling thermometers (for Asian-American and racial bias) and the family and career items (for gender bias).

Implicit attitude scores were computed using the revised scoring algorithm of the implicit association test (IAT) (Greenwald, Nosek, & Banaji, 2003). The IAT is a computerized task comparing reaction times to categorize paired concepts (in this case, social groups, e.g., Asian American vs. European American) and attributes (in this case, valence categories, e.g., good vs. bad). Average response latencies in correct

categorizations were compared across two paired blocks in which participants categorized concepts and attributes with the same response keys. Faster responses in the paired blocks are assumed to reflect a stronger association between those paired concepts and attributes. In all tests, positive IAT D scores indicate a relative preference for the typically preferred group. Respondents whose scores fell outside of the conditions specified in the scoring algorithm did not have a complete IAT D score and were therefore excluded from analyses. Restricting the analyses to only complete IAT D scores resulted in an average retention of 92% of the complete sessions across tests. The sample was further restricted to include only respondents from the United States to increase shared cultural understanding of attitude categories. The sample was restricted to include only respondents with complete demographic information on age, gender, race/ethnicity, and political ideology.

We used explicit and implicit bias data for January 2017 – March 2020 and created monthly estimates for each of the explicit and implicit bias domains. Because of possible selection bias among the Project Implicit participants, we adjusted population estimates by weighting the monthly scores based on their representativeness of the demographic frequencies in the U.S. population (age, race, gender, education; estimated biannually by the U.S. Census Bureau; https://www.census.gov/data/tables/time-series/demo/popest/2010s-national-detail.html). Further, we adjusted weights based on political orientation (1 = "strongly conservative;" 2 = "moderately conservative;" 3 = "slightly conservative;" 4 = "neutral;" 5 = "slightly liberal;" 6 = "moderately liberal;" 7 = "strongly liberal"), using corresponding annual estimates from the General Social Survey. With the weighting values for each participant, we computed weighted monthly means for each attitude test. These procedures ensured that weighted monthly averages approximated the demographics in the U.S. population. We cross-validated this procedure by comparing weighted annual scores to nationally representative estimates for feeling thermometer for African-American and Asian-American estimates from the American National Election studies in 2017 and 2018.

Participant_Responses.csv https://osf.io/nwj35

Operationalization mostly aligns with that in the manuscript. Few minor remarks:

Implicit bias: the operationalization of the implicit gender–career bias is missing (probably just an oversight).

Explicit bias: The preregistration has overlapping components with the manuscript but the operationalizations are not exactly the same. The operationalization in the manuscript is more detailed, particularly with how explicit bias and relative scores are computed. For example, that explicit bias is measured on a seven-point scale, and it includes a thorough explanation of the calculation for relative bias scores (e.g., subtracting incongruent from congruent associations).

Participant_Responses.csv contains forecasts. Some values appear inconsistent; for example, for some participants, the odd and even values are the same, respectively.

- "R_2UgadrXE2OiyQIB 4/27/2020 3:59 TheMets 1,2,3,4,5,6,7,8,9,10 lifesat 6.362 6.387 6.362 6.387 6.362 6.387 6.362 6.387 6.362 6.387 6.362 6.387"
- "R_2UgadrXE2OiyQIB 4/27/2020 3:59 TheMets 1,2,3,4,5,6,7,8,9,10 iasian 0.407 0.38 0.407 0.38 0.407 0.38 0.407 0.38 0.407 0.38 0.407 0.38"

## Sampling Plan

### Existing Data

Registration prior to creation of data

### Explanation of existing data

Teams submitted their forecasts for the first portion of the study between late May and June 2020 and the data was reviewed by a research assistant to remove blank and duplicate entries. Accuracy data does not exist yet (we don't have a time machine!), and consequently none of the research questions for the present pre-registration can be analyzed yet.

*Registration prior to creation of data* is misleading, as the data exists, as noted below. We understand that the hypotheses could not have been tested since the predictions had not yet been submitted. However, it would have been more accurate to refer to it as *Registration prior to analysis of the data."*

### Data collection procedures

Participants were recruited via large scale advertising on social media, mailing lists in the behavioral and social sciences, decision sciences, and data science, advertisement on academic social networks including Researchgate, and through word of mouth.

To ensure broad representation across the academic spectrum of relevant disciplines, we targeted groups of scientists working on computational modeling, social psychology, judgment and decision-making, and data science.

### Sample size

Our targeted sample size was 40 teams/participants. A total of 86 different teams submitted predictions.

**Sample size rationale** Using GPower 3.1, we estimated that for a typical effect size for the forecasted social issues 37, f = .14, with 12 measurement points provided by participants (reflecting forecasts for 12 months), with 80% power, and an expected correlation among repeated times series data points of .7, we would need 33 scientist teams for the tournament to statistically compare accuracy among three groups (experts vs. data-based forecasts vs. hybrid-based forecasts).

**Stopping rule** Participants were recruited over a 1 month period, with the intent of extending the deadline until the 40 team minimum was reached.

This part of the preregistration aligns with the main manuscript; however, it is unclear how many teams had been recruited by May, when the first predictions started. Was the rationale for extending recruitment due to not having enough teams by the end of April, leading to 46 teams joining between May and mid-June? Teams that joined in mid-June 2020 likely had to submit predictions for May 2020 retrospectively and had more information compared to teams that submitted by the end of April. This raises further confusion.

Further, the final sample size is double the preregistered sample size. It would be of interest to know whether the planned sample size would have yield the same results.

## Variables

### Measured variables

*Forecasting justifications.*

For each forecasting model submitted to the tournament, participants provide detailed descriptions. They describe the type of model they computed (e.g., time series, game theoretic models, other algorithms), model parameters, additional variables they included in their predictions (e.g.., COVID-19 trajectory, presidential election outcome), and underlying assumptions. Additional parameters can be continuous variables (e.g., COVID-19 deaths; unemployment rate) or based on a single discrete event (e.g., political leadership change; implementation of a policy measure). Participants also provide a theoretical justification for these decisions.

The description of the measured variable is almost identical to that provided in the manuscript.

*Confidence.*

Participants will rate their confidence in their forecasted points for each forecast model they submit. Confidence will be rated on a 7-point scale from 1 (not at all) to 7 (extremely).

The description of the measured variable is almost identical to that provided in the manuscript.

*COVID-19 Conditional.*

Next, we zero-in on the COVID-19 pandemic as a conditional of interest given links between infectious disease and the target social issues we selected for this tournament. Continuous real-time data for this variable is being currently being gathered and continue to be available over the course of the forecasting tournament. Participants will report if they used the past or predicted trajectory of the COVID-19 pandemic (as measured by number of deaths or prevalence of cases or new infections) as a conditional in their model, and if so will provide their forecasted estimates for the COVID-19 variable included in their model.

The description of the measured variable is almost identical to that provided in the manuscript.

*Counterfactuals.*

Counterfactuals are hypothetical alternative historic events that would be thought to affect the forecast outcomes, if they were to occur. Participants will describe the key counterfactual events between December 2019 and April 2020 that they theorize would have led to different forecasts (e.g., U.S.-wide implementation of social distancing practices in February). Two independent coders will evaluate the distinctiveness of counterfactuals. If discrepancies arise, they will discuss individual cases with other members of the forecasting collaborative to make the final evaluation.

The description of the measured variable is almost identical to that provided in the manuscript, except that the binary coding for the presence of counterfactuals (yes/no) is missing.

*Team characteristics.*

To assess objective expertise, teams report if any of their members have previously researched or published on the topic of their forecasted variable. They also report each member's areas of expertise and amount of education. To assess subjective expertise, teams will report their agreement with the statement: "My team has strong expertise on the research topic of Life Satisfaction."

This part is termed as the *confidence in expertise* in the manuscript.

### Categorization of Forecasts

We will categorize forecasts based on modeling approaches. Specifically, two independent research associates will categorize forecasts for each domain based on provided justifications: 1. purely based on (a) theoretical model(s); 2. purely based on data-driven model(s); 3. a combination of theoretical and data-driven models – i.e., computational model relies on specific theoretical assumptions.

We will further identify modelling approaches that solely rely on extrapolation of time series from the data we provided (e.g., ARIMA, moving average with lags; yes/no). Disagreements between coders here and below will be resolved through joint discussion with the leading three authors of the project.

The description of the categorization is almost identical to that provided in the manuscript.

### Categorization of Additional variables

We will test how the presence and number of additional variables as parameters in the model impact forecasting accuracy. To this end, we will ensure that additional variables are distinct from one another. Two independent coders will evaluate the distinctiveness of each reported parameter. When there are discrepancies arise, the coders will discuss the case with lead members of the forecasting collaborative to arrive at a consensus.

The description of the categorization is almost identical to that provided in the manuscript.

## Categorization of Teams

We will next categorize teams based on compositions. First, we will sort contributors into three categories: 1. singular forecaster; 2. small group ($n < 6$); 3. large group ($n \geq 6$).

Next, we will sort teams based on disciplinary orientation: 1. behavioral sciences; 2. social sciences; 3. computer sciences; 4. interdisciplinary/other.

Finally, we will use information teams provided concerning their objective and subjective expertise level for a given subject domain. We will use each covariate in separate multi-level analyses with domains and time points as predictors and absolute percentage error scores for a given forecast as a dependent variable.

The preregistration entails more details regarding categorization of teams based on the group size. In the main manuscript, the authors explicitly state that they were *"comparing behavioural and social scientists with teams from computer and data science."*

It is also unclear to which analysis in the main manuscript, the last paragraph refers. Our assumption is, the analysis presented in Fig. 6 and Supplementary Table 5. If correct, then the authors have deviated from the preregistered analysis. This deviation has been transparently declared: *"Deviating from the pre-registration, we performed a single analysis with all covariates in the same model rather than performing separate analyses for each set of covariates, to protect against inflating P values. Furthermore, due to scale differences between domains, we chose not to feature analyses concerning absolute percentage errors of each time point in the main paper."*

### Forecasting Update Justifications

Given that participants will receive both new data and a summary of diverse theoretical positions they can use as a basis for their updates, two independent research associates will score participants' justifications for forecasting updates on three dummy-categories: 1. new six months of data we provide; 2. theoretical insights from the summary of teams' rationales we provide; 3. consideration of other external events.

This part of the preregistration aligns with the main manuscript.

### Indices

We will calculate accuracy scores for each domain, using MASE (mean absolute scale error).

MASE is operationalized in the "Hypotheses" of the present preregistration, which is slightly confusing. Moreover, the authors write in their manuscript that *Additionally, in an a priori specific document shared with the journal in May 2020, we outlined the operationalization of the key dependent variable (MASE),….*

## Analysis Plan

### Confirmatory Analyses: Comparison of Forecasting Models

We will first investigate overall forecasting accuracy in behavioral and social sciences by examining MASE for each of the forecasting domains. Using MASE scores will allow us to compare forecasted models against the naïve baseline model. We will also compare forecasting accuracy agaist accuracy of classic naive forecasting estimators (i. average and ii. seting all forecasts to be the value of the last observation).

It is unclear what exact specification is used for the naïve baseline model. Based on the text in the infobox above, we assume that this is something different than the average and setting all forecasts to be the value of the last observations - as these are referred to as other comparisons that will be performed.

To our understanding, the conducted analysis slightly deviates from the preregistered analysis. Such as:

- **Main Text**: *(1): The historical mean, calculated by randomly resampling the historical time series data;*
- **Preregistration**: *Average classic naive forecasting estimator.*

Comment: This is largely equivalent, but the preregistration lacks the additional detail about random resampling in the main text.

- **Main Text**: *(2): A naive random walk, calculated by randomly resampling historical changes in the time series data with an autoregressive component;*
- **Preregistration**: *All forecasts set to the value of the last observation (less detail, and some deviation).*

Comment: The preregistered version is a simpler naive random walk (just setting forecasts to the last observation), while the main text adds complexity with the autoregressive component and random resampling.

- **Main Text**: *(3): Extrapolation from linear regression, based on a randomly selected interval of the historical time series data;*
- **Preregistration: Not reported.**

Comment: This is an additional method introduced in the main text that was not part of the preregistration.

### Inference criteria

alpha of 5 %, two-tailed tests.

### Data exclusion

Teams will be contacted to confirm their predictions in cases where the forecasted values differ dramatically (sign, direction) from the historical data provided or is 3 SD above the mean of the sample for a given time point. Upon confirming and correcting entries based on participants' feedback, we will keep all remaining outliers confirmed by each forecasting team.

### Missing data

For each team, any domain that does not include 12 monthly predictions will will be excluded from analysis. In the event that a partial prediction has been submitted, the team will be contacted to confirm their intended predictions.

The preregistered inference criteria make no reference to the Bayes factor, which is reported in the main manuscript.

The data exclusion criteria in the main manuscript do not specify if and under what circumstances teams were contacted to confirm their predictions. However, according to the manuscript, the teams had the opportunity to confirm their entries upon submission.

As to the "missing data", we fail to find any mention regarding handling of missing data in the main manuscript. The handling of missing data in the general sample is reported.

### Exploratory analysis

**Exploratory Analyses: Comparison of Different Approaches/Teams** The main exploratory (two-tailed) analyses will compare MASE scores for the whole forecasted time series as well as percent of absolute error for each individual forecasted time point when using different forecasting approaches. To this end, we will fit a series of linear mixed effect models. For models evaluating overall accuracy of the forecasted model, we will use forecasting type (purely theoretical, purely data-driven and hybrid models), forecasting domain as predictors, with MASE scores nested within teams. Next, we will examine how the theory-free "extrapolation of time series" models compare in forecasting accuracy to models that rely on other model parameters and/or theoretical assumptions, by including this contrast between models and forecasting domain as predictors, with MASE scores nested within teams. For models evaluating accuracy of individual time points, we will use forecasting type (purely theoretical, purely data-driven and hybrid models), forecasting domain and time points as predictors, with absolute percent deviation scores nested within teams.

This part of the preregistration aligns with the main manuscript.

We will use equivalent analyses with team type and confidence (instead of forecasting type) as predictors. Further, we will examine whether presence of additional parameters (beyond time series data we provide) and counterfactuals significantly alters forecasting accuracy. First, in a series of linear mixed models similar to the one outlined above we will examine whether presence (dummy-coded yes/no) or number of considered additional parameters and counterfactuals moderate the forecasting accuracy (MASE scores for total accuracy / percent of absolute error for accuracy at specific time points).

The results of this analysis are reported in the supplementary Table 5.

Next, we will zero-in forecasts including COVID-19 virus trajectory as a conditional. For these forecasts, we will first estimate the forecasting accuracy of the COVID-19 trajectory by evaluating MASE scores for COVID-19 death against the actual number of deaths. We will use these conditional forecasting accuracy scores as a moderator in linear models evaluating accuracy of each of the targeted domain. We will further conduct simple slope analyses, evaluating the role of conditional forecasting accuracy for the accuracy of the forecast in targeted domains. Such analyses can reveal whether participants' forecasting errors in targeted domains may be qualified by their accuracy in expectations for the virus trajectory.

The results of this analysis are reported in the supplementary "Were forecasting teams wrong for the right reasons?"

Additionally, expert forecasts will be compared to the "wisdom of the crowds" standard by comparing the accuracy of expert forecasts to those of lay forecasts (aiming N = 200 American residents for each of the five domains - racism, gender bias, Asian-American bias, politcs, well-being; gathered at the same time via Prolific, who followed a very similar procedure completed by forecasting teams) and contrasting expertise and forecasting domain as predictors, with MASE scores nested within teams. Lay forecasts will additionally be fit to a series of linear mixed models, with forecasting domain as predictors and MASE scores for each participant.

In the main manuscript, the authors transparently state that the comparison to the 'wisdom of the crowds' was preregistered prior to data pre-processing and analysis. However, in their preregistration, they aimed for a sample size of N = 200. The final sample comprised N = 802, significantly exceeding the targeted N. The authors neither provided a justification for the increased sample size nor addressed whether the results in the main manuscript would have held with N = 200.

Further, the main manuscript entails more details regarding recruitment and data cleaning.

## Other

### References


- Axt, J. R. (2018). The best way to measure explicit racial attitudes is to ask about them. Social Psychological and Personality Science, 9(8), 896-906.
- Charlesworth, T. E., & Banaji, M. R. (2019). Patterns of implicit and explicit attitudes: I. Long-term change and stability from 2007 to 2016. Psychological science, 30(2), 174-192.
- Greenwald, A. G., Nosek, B. A., & Banaji, M. R. (2003). Understanding and using the implicit association test: I. An improved scoring algorithm. Journal of personality and social psychology, 85(2), 197.
- Hehman, E., Flake, J. K., & Calanchini, J. (2018). Disproportionate use of lethal force in policing is associated with regional racial biases of residents. Social psychological and personality science, 9(4), 393-401.
- Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research, 50, 723-762.
- Luhmann, M. (2017). Using big data to study subjective well-being. Current Opinion in Behavioral Sciences, 18, 28-33.
- Nosek, B. A., Smyth, F. L., Hansen, J. J., Devos, T., Lindner, N. M., Ranganath, K. A., … & Banaji, M. R. (2007). Pervasiveness and correlates of implicit attitudes and stereotypes. European Review of Social Psychology, 18(1), 36-88.
- Ofosu, E. K., Chambers, M. K., Chen, J. M., & Hehman, E. (2019). Same-sex marriage legalization associated with reduced implicit and explicit antigay bias. Proceedings of the National Academy of Sciences, 116(18), 8846-8851.
- Schwartz, H. A., Sap, M., Kern, M. L., Eichstaedt, J. C., Kapelner, A., Agrawal, M., … & Kosinski, M. (2016). Predicting individual well-being through the language of social media. In Biocomputing 2016: Proceedings of the Pacific Symposium (pp. 516-527).

# Robustness 1: MAPE instead of MASE

3. check robustness reproducibility by using a different operationalization of forecasting accuracy (i.e., the DV). Instead of MASE, we will use MAPE (mean absolute percent error) and use random effects for domain.

Given the results from the reproducibility check and the results for Robustness 3 test, we did not complete this test.

# Robustness 2: sensitivity to forecast size

4. check robustness reproducibility by examining how sensitive the results are when accounting for forecast size (i.e., the number of claims that participating teams self-selected into submitting).

Given the results from the reproducibility check and the results for Robustness 3 test, we did not complete this test.

# Robustness 3: Comparison to judgments of learning literature

5. perform a robustness comparison to judgements of learning literature. We will calculate bias and sensitivity of the forecasting for each team by domain combination, by using a regression equation of `actual answer ~ real answer` and then extracting estimated intercept (bias) and slope (sensitivity) for each team and domain. Then, we can estimate an MLM (multilevel model) of `bias ~ 1 + (1|domain)` to determine if bias is different from zero and an MLM of `sensitivity ~ 1 + (1|domain)` to determine if sensitivity is different from zero. If these are different from zero, the forecasts are "biased" and "sensitive". Bias is traditionally .4-.6 on a standardized scale range - any values outside this range would be considered different from traditional results. Sensitivity is traditionally .2 to .4 on a standardized scale range - any values outside this range would be considered different from traditional results.

Perfect predictions would predict actual data with a slope of 1 and intercept of 0. Bias is considered the upward or downward difference from an intercept - generally, we have an overestimation bias found in the traditional Dunning-Kruger effect and other types of judgments of learning and numerical judgments of relatedness. Sensitivity is our ability judge the differences in effect size across time in this study. Sensitivity close to zero means no ability to predict effect size, while largest sensitivities relate to better ability to judge the change across time.

## Summary

We analyzed the data as intended but found results that were improbable given the range/scale of the data and deviated from our expected values. We then investigated the data to determine if these values were due to large outliers, out of range data, misalignment of values, or simply results not in line with expectations. As shown below, we found potential issues with the data used in analyses, which draw into question our ability to interpret any results from this analysis. The report of the analyses and the exploration of the data are shown below.

### Libraries

```
library(rio)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
```

```
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(tidyr)
library(easystats)
```

```
## # Attaching packages: easystats 0.7.3 (red = needs update)
## ✓ bayestestR  0.15.0   ✓ correlation 0.8.6
## ✓ datawizard  0.13.0   x effectsize  0.8.9
## ✓ insight     1.0.0    x modelbased  0.8.8
## ✓ performance 0.12.4   ✓ parameters  0.24.0
## ✓ report      0.5.9    ✓ see         0.9.0
##
## Restart the R-Session and update packages with `easystats::easystats_update()`.
```

```
library(ggplot2)
```

### Data

To complete this analysis, we first looked for the appropriate author provided data files to get the *actual* answer and the team *predicted* answer. Given the file `Wave 1+2 descriptives.Rmd`:

- the actual answer is in the `historical_data.csv` and is the column `Actual_Score`.
- the predicted answer is in the `Wave1+2data.csv` and is in the `Month` columns. This dataset represents the rawest form of data we could find - as noted later, investigating the study methods showed that teams uploaded excel files as their answers for each of the topic areas. This data could not be found. We also investigated various different forms of the wave 1 and 2 data (as there are multiple files labeled with wave 1 and wave 2), and the problems mentioned below persist.

```
# actual / real answer ... historical data by month
DF_actual <- import(here::here("original_files", "historical_data.csv")) %>%
    select(-V1) %>%
    pivot_longer(cols = -Month,
             names_to = "domain",
             values_to = "Actual_Score") %>%
    na.omit()

# predicted answer ... month variables
DF_predicted_1 <- import(here::here("original_files", "Wave1+2data.csv")) %>%
    filter(phase == 1) %>%
    select(team_name, domain, phase, Month.1:Month.12)

# kind of unclear what scores wave 2 were compared to but guess is 1-12 since no historical data is the
DF_predicted_2 <- import(here::here("original_files", "Wave1+2data.csv")) %>%
    filter(phase == 2) %>%
    select(team_name, domain, phase, Month.7:Month.18) %>%
```

```
    rename(Month.12 = Month.18,
           Month.11 = Month.17,
           Month.10 = Month.16,
           Month.9 = Month.15,
           Month.8 = Month.14,
           Month.7 = Month.13,
           Month.6 = Month.12,
           Month.5 = Month.11,
           Month.4 = Month.10,
           Month.3 = Month.9,
           Month.2 = Month.8,
           Month.1 = Month.7)

DF_predicted <- bind_rows(DF_predicted_1, DF_predicted_2) %>%
    pivot_longer(cols = c(Month.1:Month.12),
             names_to = "Month",
             values_to = "Predicted_Score") %>%
    mutate(Month = gsub("Month.", "", Month),
         Month = as.integer(Month))

# merge together
DF_long <- DF_predicted %>%
    left_join(
        DF_actual,
        by = c("domain", "Month")
    ) %>%
    na.omit()

head(DF_long)
```

```
## # A tibble: 6 x 6
##   team_name domain  phase Month Predicted_Score Actual_Score
##   <chr>     <chr>   <int> <int>           <dbl>        <dbl>
## 1 ms607     lifesat     1     1            6.24         6.33
## 2 ms607     lifesat     1     2            6.26         6.22
## 3 ms607     lifesat     1     3            6.22         6.30
## 4 ms607     lifesat     1     4            6.24         6.33
## 5 ms607     lifesat     1     5            6.27         6.34
## 6 ms607     lifesat     1     6            6.26         6.34
```

To calculate bias and sensitivity in the ranges we predicted, we needed to scale the data so that they were always in the same range from 0 to 1. After examining the min/max and the methods, we rescaled the data dividing by the max of the range or reversing the scale when necessary.

```
# rescale
DF_long %>%
    group_by(domain) %>%
    summarize(min = min(Actual_Score),
            max = max(Actual_Score),
            min_p = min(Predicted_Score),
            max_p = max(Predicted_Score))
```

```
## # A tibble: 12 x 5
##    domain       min    max  min_p   max_p
##    <chr>      <dbl>  <dbl>  <dbl>   <dbl>
```

```
##  1 eafric    -0.226 -0.177 -0.31    0.1
##  2 easian    -0.247 -0.138 -0.284   0.4
##  3 egend      0.736  0.852  0.670   1.12
##  4 iafric     0.256  0.286  0.25    0.385
##  5 iasian     0.327  0.366  0.299   0.7
##  6 ideoldem  44.2   50.1   35      65
##  7 ideolrep  40.5   48     25      50
##  8 igend      0.339  0.359  0.3     0.487
##  9 lifesat    6.22   6.35   3       7
## 10 negaffect  1.16   2.23   0.7     3.38
## 11 polar     78     91     65     100.
## 12 posaffect -1.66  -1.14  -2.5    -0.4
```

```
DF_long <- DF_long %>%
    mutate(Actual_Score =
            ifelse(
                domain == "ideoldem" | domain == "ideolrep" | domain == "polar",
                Actual_Score / 100,
                Actual_Score),
        Predicted_Score =
            ifelse(
                domain == "ideoldem" | domain == "ideolrep" | domain == "polar",
                Predicted_Score / 100,
                Predicted_Score),
        Actual_Score =
            ifelse(
                domain == "negaffect" | domain == "posaffect" | domain == "lifesat",
                Actual_Score / 7,
                Actual_Score),
        Predicted_Score =
            ifelse(
                domain == "negaffect" | domain == "posaffect" | domain == "lifesat",
                Predicted_Score / 7,
                Predicted_Score),
        Actual_Score = ifelse(
            domain == "eafric" | domain == "easian" | domain == "posaffect",
            -Actual_Score,
            Actual_Score
        ),
        Predicted_Score = ifelse(
            domain == "eafric" | domain == "easian" | domain == "posaffect",
            -Predicted_Score,
            Predicted_Score
        )
    )

head(DF_long)
```

```
## # A tibble: 6 x 6
##   team_name domain  phase Month Predicted_Score Actual_Score
##   <chr>     <chr>   <int> <int>           <dbl>        <dbl>
## 1 ms607     lifesat     1     1           0.891        0.905
## 2 ms607     lifesat     1     2           0.894        0.888
## 3 ms607     lifesat     1     3           0.889        0.901
## 4 ms607     lifesat     1     4           0.891        0.904
```

```
## 5 ms607     lifesat     1     5           0.896        0.905
## 6 ms607     lifesat     1     6           0.894        0.905
```

### Analysis: Calculate Bias and Sensitivity

In this section, we calculated the bias and sensitivity as follows:

- Create a separate calculation for each team and domain and phase
- Use the predicted score to predict the actual score
- Save the team, domain, bias (intercept), and sensitivity (slope) in a dataframe.

```
store_results <- list()
iter <- 1
for (i in unique(DF_long$team_name)) {
    for (q in unique(DF_long$domain)){
        for (r in 1:2){
            # cat(i)
            temp <- DF_long %>%
                filter(team_name == i) %>%
                filter(domain == q) %>%
                filter(phase == r)

            if (nrow(temp) > 0){
            temp.model <- lm(
                Actual_Score ~ Predicted_Score,
                data = temp
            )
            store_results[[iter]] <- data.frame(
                team = i,
                domain = q,
                bias = coef(temp.model)[1],
                sensitivity = coef(temp.model)[2],
                n_est = nrow(temp),
                phase = r
            )
            iter <- iter + 1
            }
        }
    }
}
```

### Analysis: Analyze

Next, we calculated the overall bias and sensitivity using a multilevel model controlling for domain as a random intercept. At first glance the results for the average bias intercept appeared within the normal range (~ .40). The average sensitivity slope appeared unusual (~ -.50), as generally these are not negative.

We also investigated the values - there are multiple sensitivity values that were calculated as *NA*, which occurs when there is no variance in estimates (and therefore, slope is technically infinity). We excluded those values for this analysis.

```
DF_results <- bind_rows(store_results)

nrow(DF_results)
```

```
## [1] 726
```

```r
DF_results <- DF_results %>%
    na.omit()

nrow(DF_results)
```

```
## [1] 625
```

```r
model.bias <- lme(
    fixed = bias ~ 1,
    data = DF_results,
    random = list(~1|domain)
    )

summary(model.bias)
```

```
## Linear mixed-effects model fit by REML
##   Data: DF_results
##        AIC      BIC    logLik
##   3294.047 3307.355 -1644.023
##
## Random effects:
##  Formula: ~1 | domain
##         (Intercept) Residual
## StdDev:   0.1597194 3.352079
##
## Fixed effects:  bias ~ 1
##                 Value Std.Error  DF t-value p-value
## (Intercept) 0.4090989 0.1422943 613 2.87502  0.0042
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -11.28009131  -0.07700560  -0.02722771   0.07289299  10.51692785
##
## Number of Observations: 625
## Number of Groups: 12
```

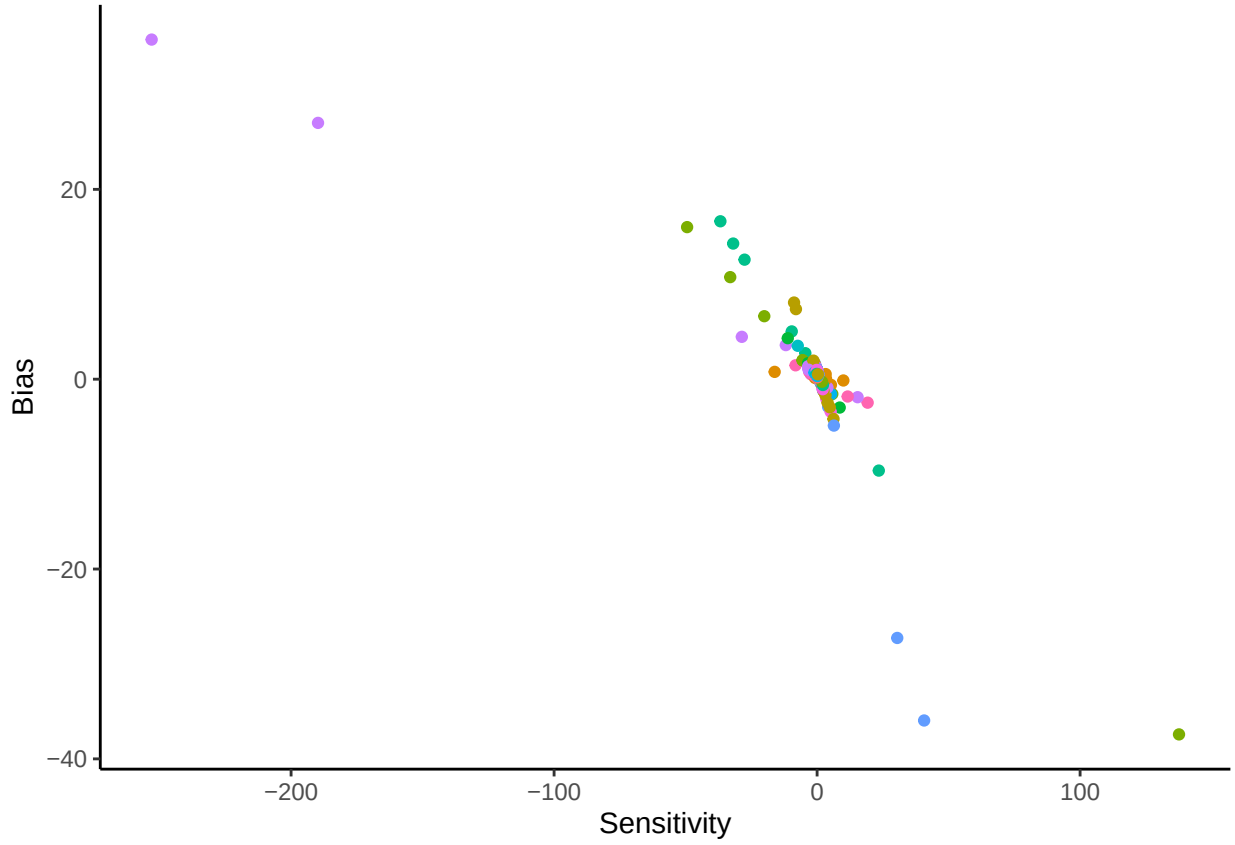
```r
model.sensitivity <- lme(
    fixed = sensitivity ~ 1,
    data = DF_results,
    random = list(~1|domain),
    na.action = "na.omit"
    )

summary(model.sensitivity)
```

```
## Linear mixed-effects model fit by REML
##   Data: DF_results
##        AIC      BIC    logLik
##   5124.514 5137.822 -2559.257
##
## Random effects:
##  Formula: ~1 | domain
##         (Intercept) Residual
## StdDev:    1.668922 14.47919
##
```

```
## Fixed effects:  sensitivity ~ 1
##                   Value Std.Error  DF    t-value p-value
## (Intercept) -0.4843511 0.7597456 613 -0.6375174   0.524
##
## Standardized Within-Group Residuals:
##           Min            Q1           Med            Q3           Max
## -17.216322356  -0.006973405   0.024029124   0.066989363   9.507152934
##
## Number of Observations: 625
## Number of Groups: 12
```

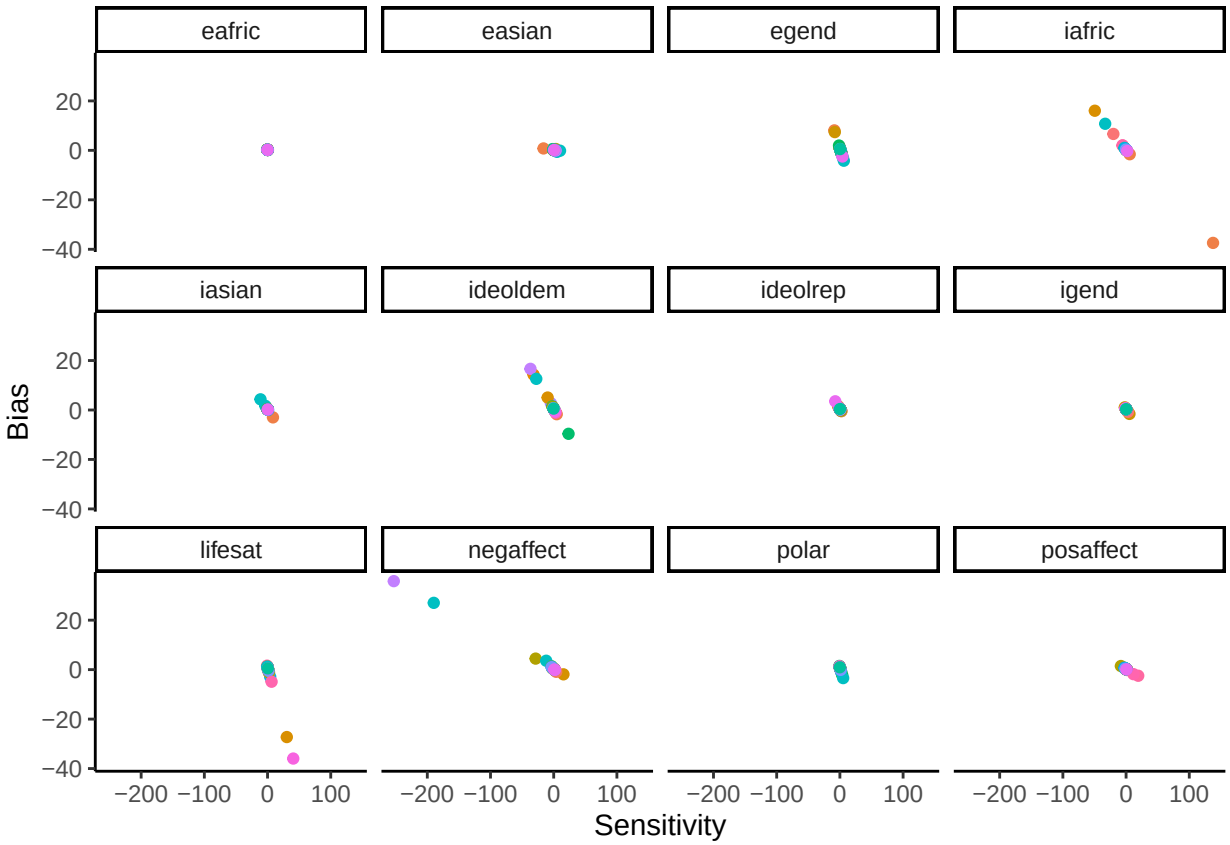
To investigate further, we examined the values for bias and sensitivity using `ggplot2`. Both graphs show extremely unusual scores for bias and sensitivity - remember that the data is scaled to 0-1 for each of the scales to ensure the data is in the same range as normal judgment of learning results. Therefore, finding bias/sensitivity results that are over 1 to 2 points is extremely odd - and many of these scores well over 50 to 100 times that amount.

```
ggplot(DF_results, aes(sensitivity, bias, color = domain)) +
    geom_point() +
    theme_classic() +
    xlab("Sensitivity") +
    ylab("Bias") +
    theme(legend.position = "none")
```




```
ggplot(DF_results, aes(sensitivity, bias, color = team)) +
    geom_point() +
    theme_classic() +
```

```
    facet_wrap(~domain) +
    xlab("Sensitivity") +
    ylab("Bias") +
    theme(legend.position = "none")
```




Therefore, we decided to examine the data to determine if this result is due to the task or some other issue in the data.

### Review Raw Scores

One thing we noticed by reviewing the data for the very high bias/sensitivity teams was that the data repeated the exact same numbers for odd months and even months with a high level of precision. For example:

```
# note we went back to DF_predicted because it's the original raw data before edits in DF_long
DF_predicted %>%
    filter(team_name == "TheMets")
```

```
## # A tibble: 144 x 5
##    team_name domain  phase Month Predicted_Score
##    <chr>     <chr>   <int> <int>           <dbl>
##  1 TheMets   lifesat     1     1            6.36
##  2 TheMets   lifesat     1     2            6.39
##  3 TheMets   lifesat     1     3            6.36
##  4 TheMets   lifesat     1     4            6.39
##  5 TheMets   lifesat     1     5            6.36
##  6 TheMets   lifesat     1     6            6.39
```

```
##  7 TheMets   lifesat     1     7            6.36
##  8 TheMets   lifesat     1     8            6.39
##  9 TheMets   lifesat     1     9            6.36
## 10 TheMets   lifesat     1    10            6.39
## # i 134 more rows
```

The odds of teams entering the same scores, for all their estimates, for odd months and then separately for even months seems very low. To investigate this problem on a larger scale, we calculated the correlation matrix of each team's scores. We then grabbed the lower triangle of the correlations (i.e., all unique pairwise correlations that do not include the month correlated with itself). Any team with a perfect correlation between monthly estimates, we marked as suspicious.

```
teams <- unique(DF_predicted$team_name)

correl_matrix <- list()
correl_num <- list()
for (team in teams){
    for (phase_num in 1:2){
        temp <- DF_predicted %>%
            filter(team_name == team) %>%
            filter(phase == phase_num) %>%
            pivot_wider(id_cols = c(team_name, domain),
                    names_from = Month,
                    values_from = Predicted_Score) %>%
            select(-team_name, -domain)

          if (nrow(temp) > 2){

          correl_matrix[[paste0(team,"_", phase_num)]] <- temp %>%
            reframe(correl = cor(., use = "pairwise.complete.obs")) %>%
            as.matrix()

          lower.triangle <- correl_matrix[[paste0(team,"_", phase_num)]][lower.tri(as.matrix(correl_mat

         correl_num[[paste0(team,"_", phase_num)]] <- sum(lower.triangle == 1, na.rm = TRUE)

    }

  }

}
```

Should not have perfectly correlated monthly data:

```
correl_total <- bind_rows(correl_num) %>%
  pivot_longer(cols = everything()) %>%
  filter(value > 0)

nrow(correl_total) # number of teams/phases that have odd correlations
```

```
## [1] 29
```

```
length(correl_num) # total number of possible team/phase combos
```

```
## [1] 98
```

Therefore, 29.59% of team and phase combinations are potentially odd data.

No variability in scores:

Another issue is that some teams' scores are simply the same score repeated for each month - we don't know if that's what they entered, but that data should be at least examined because it does not appropriately complete the required task.

```
teams <- unique(DF_predicted$team_name)
no_variable <- list()

for (team in teams){
    for (phase_num in 1:2){
        temp <- DF_predicted %>%
            filter(team_name == team) %>%
            filter(phase == phase_num) %>%
            pivot_wider(id_cols = c(team_name, domain),
                    names_from = Month,
                    values_from = Predicted_Score) %>%
            select(-team_name)

            if (nrow(temp) > 0){
                no_variable[[paste0(team, "_", phase_num)]] <- data.frame(
                sd = apply(temp %>% select(-domain), 1, sd, na.rm = T),
                domain = temp$domain,
                team = team,
                phase = phase_num
                )
                }
  }
}

sd_issues <- bind_rows(no_variable)

nrow(sd_issues) # total number of teams by domains by phase
```

```
## [1] 726
```

```
nrow(sd_issues %>% filter(sd == 0)) # number of problematic estimates
```

```
## [1] 101
```

Therefore, 13.91% of team and phase combinations are potentially odd data.

Given the potential issues with the data, we did not continue with the judgment of learning analysis. It was unclear how to interpret the results given the questions about the data.

# Session Info

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 20.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3;  LAPACK version 3.9.0
```

```
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Vienna
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggplot2_3.5.1      see_0.9.0          report_0.5.9       parameters_0.24.0
##  [5] performance_0.12.4 modelbased_0.8.8   insight_1.0.0      effectsize_0.8.9
##  [9] datawizard_0.13.0  correlation_0.8.6  bayestestR_0.15.0  easystats_0.7.3
## [13] tidyr_1.3.1        nlme_3.1-166       dplyr_1.1.4        rio_1.2.3
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.6       xfun_0.49          lattice_0.22-6     vctrs_0.6.5
##  [5] tools_4.4.0        generics_0.1.3     sandwich_3.1-1     tibble_3.2.1
##  [9] fansi_1.0.6        pkgconfig_2.0.3    R.oo_1.26.0        Matrix_1.7-0
## [13] data.table_1.16.2  lifecycle_1.0.4    compiler_4.4.0     farver_2.1.2
## [17] stringr_1.5.1      munsell_0.5.1      codetools_0.2-20   htmltools_0.5.8.1
## [21] yaml_2.3.10        pillar_1.9.0       MASS_7.3-61        R.utils_2.12.3
## [25] multcomp_1.4-26    tidyselect_1.2.1   digest_0.6.37      mvtnorm_1.3-1
## [29] stringi_1.8.4      purrr_1.0.2        labeling_0.4.3     splines_4.4.0
## [33] rprojroot_2.0.4    fastmap_1.2.0      grid_4.4.0         here_1.0.1
## [37] colorspace_2.1-1   cli_3.6.3          magrittr_2.0.3     survival_3.7-0
## [41] utf8_1.2.4         TH.data_1.1-2      withr_3.0.2        scales_1.3.0
## [45] estimability_1.5.1 rmarkdown_2.29     emmeans_1.10.4     zoo_1.8-12
## [49] R.methodsS3_1.8.2  coda_0.19-4.1      evaluate_1.0.1     knitr_1.49
## [53] rlang_1.1.4        xtable_1.8-4       glue_1.8.0         rstudioapi_0.16.0
## [57] R6_2.5.1
```