Online Supplement
Racing the Clock: Using Response Time as a Proxy for Attentiveness on Self-Administered Surveys

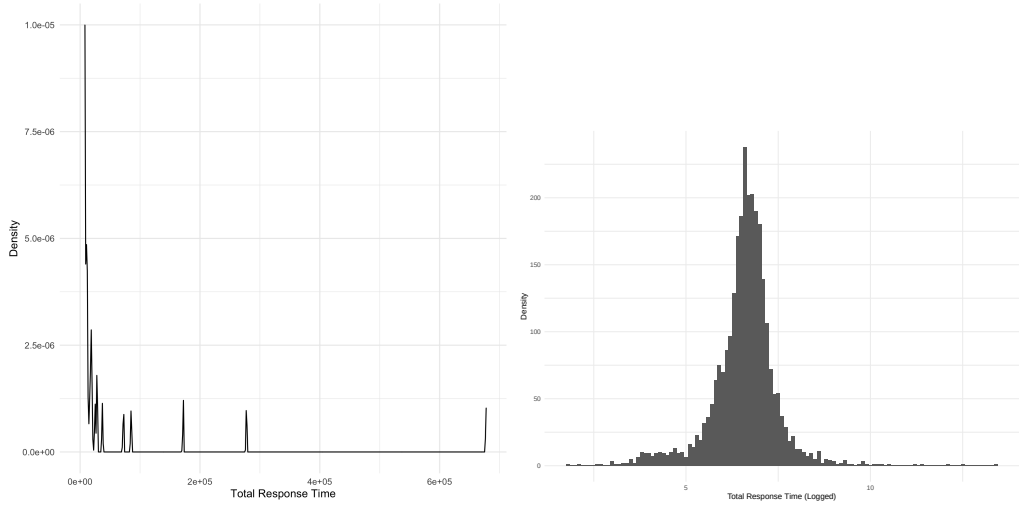
Blair Read, Lukas Wolters, and Adam J. Berinsky

Summary

This is the Online Supplement for "Racing the Clock: Using Response Time as a Proxy for Attentiveness on Self-Administered Surveys" (Read et al. 2021). Please direct questions and comments to the corresponding author at bmread@mit.edu.

A Conceptualizing Slow Respondents

In the text, we discussed the potential ambiguities surrounding the interpretation of slow respondents. Here, we highlight the extremely skewed nature of the response time data, and discuss how to conceptualize the data. Figure A.1 shows the distribution of global response times with the raw (left panel) and normalized (right panel) data. This shows that although the logged data remain skewed, the raw data are skewed such that there is a sole data point on the slowest end of the distribution.



(a) Raw Data (b) Transformed Data

Figure A.1. Distributions of Global Response Time with Raw and Transformed Data.

Table A.1 outlines how response time conceptually maps onto respondent category under our approach. Of note here are the two diverging potential interpretations for slow respondents. Figure A.2 shows that indeed, the variance of slow respondents is higher than that of fast respondents. This also serves as a useful visualization of the importance of both global mean and variance, which we are able to account for by using our PCA dimension-reduction step.

The word count analysis in Figure 6 shows that slow respondents, *on average* provide more succinct and lower-quality answers to the open-ended question. This suggests that there are a large number of slow respondents who are satisficers. While the variance in Figure A.2 indicates wide variability in the response time behavior of slow respondent, our use of the PCA algorithm, which extracts multidimensional data, allows us to consider both mean and variance in our clusters.

Finally, Figure A.3 shows the frequency of slow respondents using a different metric: how often did respondents seemingly leave and then return? We calculated the proportion of respondents who left the survey during at least one

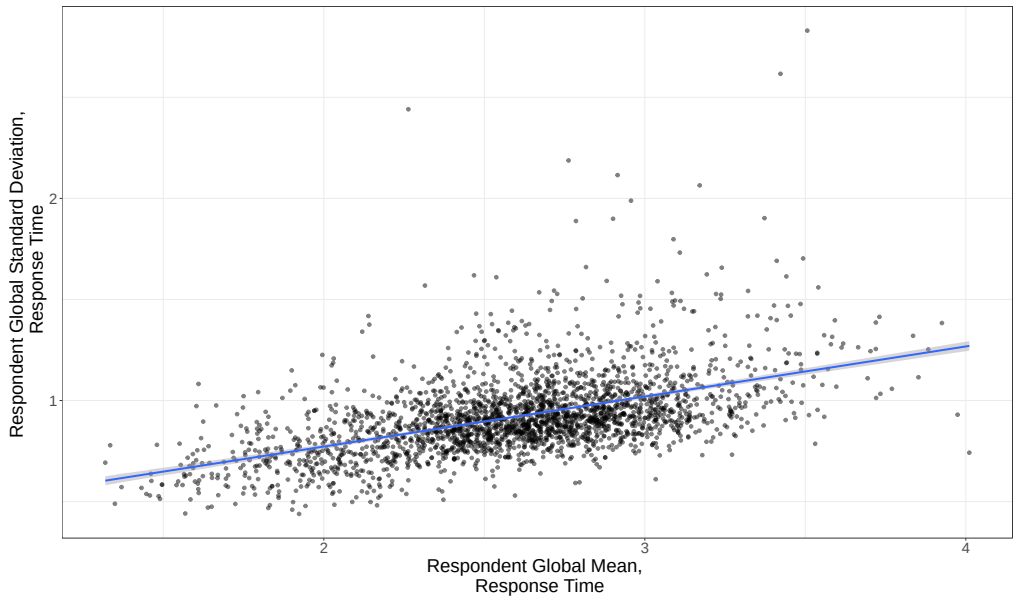
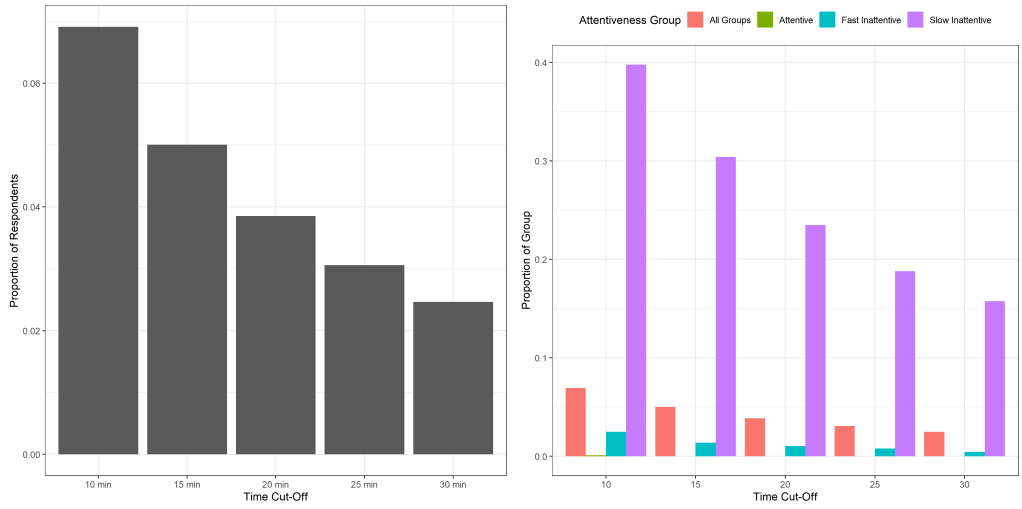


Figure A.2. Comparison of Respondent-Wise Global Mean and Variance: This plot shows the mean and variance for each respondent across all question timers. As respondents take longer on the survey across all questions, they also show higher variance. Overall, slower respondents exhibit more erratic survey-taking behavior.

question for a meaningful amount of time. Since we are unable to definitively say when respondents left the survey to do something else just by the amount of time they spent, we identify five different thresholds (between 10 and 30 minutes) that we count as “leaving" the survey. Figure A.3a shows the proportion of respondents who left the survey for longer than the threshold time for at least one question. Figure A.3b shows the proportion of respondents in each attentiveness group who left for longer than the threshold time. Importantly, we can see that *none of the attentive* respondents left the survey at any question for longer than ten minutes, while over 40% of the slow inattentive group left the survey for longer than the threshold time at some point.

Respondent Category	Observable Response Time Behavior	Assumptions about Respondent
Fast	Primarily rushes, particularly on long questions	Inattentive and satisficing
Baseline Duration	Takes longer on more complex questions	Attentive
Slow	Slow on some question, rushing on others *and/or* Slow on all questions	Inattentive and satisficing *and/or* Attentive and less cognitively advanced

Table A.1. Typology of Respondent Attentiveness Categories: This table presents a typology of the different types of survey respondents we hypothesize, and indicates their observable response time behavior as well as the associated assumption about their survey attentiveness. Contrary to previous work, we add a respondent category for slow survey-takers, and hypothesize that these respondents are either distracted and inattentive or cognitively restrained.



(a) (b)

Figure A.3. Proportion of Respondents Who Left: This figure shows the proportion of respondents who left the survey for substantial amounts of time, using different time markers.

B Algorithm Performance and Modifications

The number of PCA dimensions used in final analyses is often selected by theory. However, because our interest is in the latent behavior of attentiveness, it is difficult to decide on a theoretically-driven number of components. Instead, we opt to include 80% of variation.

In the paper, we show the first component's loadings, i.e., the weights that indicate the relationship between each variable and principal component, in Figure 4(a). As expected, longer and more complicated questions are responsible for a lot of the variation captured in the first component.

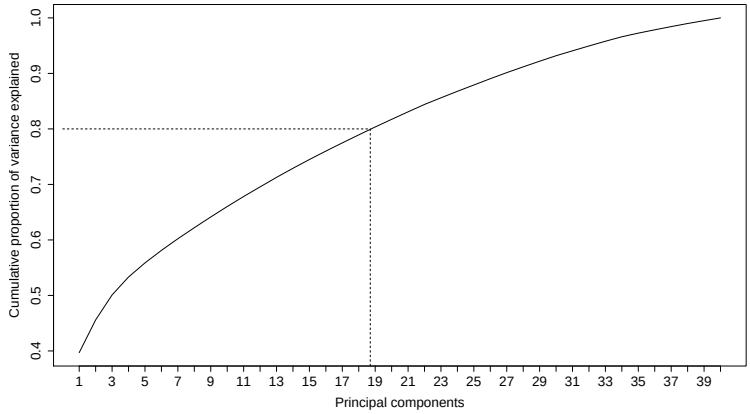


Figure B.1. Cumulative amount of total variance explained by each principal component: This plot shows the cumulative share of total variance that is explained by the principal components we obtained from our data. We retain the components that account for 80% of the total variation found in the data, which is the first 19 components.

We then select the first 19 components, which together account for 80% of the variance explained, as depicted in Figure B.1. Once we had selected the first 19 PCA components to include in our sample, we ran the EM algorithm. Figure B.2 shows that cluster assignments begins to stabilize around 80% of the variation.

Figure B.3 shows the distribution of responsibilities for all observations across all clusters. Each panel in this figure shows distribution of the the posterior probability that an observation is a member of this cluster. If observations were equally likely to be in each cluster (e.g. 33%, 33%, 33% posterior probabilities), then our assignment of observations to clusters based on the maximum responsibility would be more suspect. Instead, we can see that the vast majority of points are very likely to either be or to *not* be in each cluster, with the densities clustering around zero and one. Finally, Table B.1 shows the mean global response time and variance of each cluster. Although we are interested in capture response time across a survey rather than the global response time, manually inspecting the global response time, and variance of that quantity, can help us understand which types of respondents were in which cluster. As we can see, fastest respondents were assigned to cluster 1, and slowest respondents to cluster 3.

Cluster	Group Count	Average Time	Time Variance
Cluster 1	1168	814.92	323725.96
Cluster 2	987	879.24	91520.39
Cluster 3	362	5911.41	1567733695.56

Table B.1. Mean Global Response Time by Cluster: This table shows that respondents assigned to cluster 1 are the fastest, and those assigned to cluster 3 the slowest, to complete the survey. Combined with our qualitative inspection of the data in Figure 5, we therefore label cluster one as *fast and inattentive*, cluster 2 as *attentive*, and cluster 3 as *slow and inattentive*.

Finally, in Figure B.4, we show that the Kahneman and Tversky replication validation exercise is largely consistent with our main findings, even when we drop variables that are measured post-treatment. Figure B.4a shows the experimental treatment effects when we drop only the timers associated with the survey experiment, while Figure B.4b shows the treatment effects when we drop all timers following the treatment. As we can see, the results do not

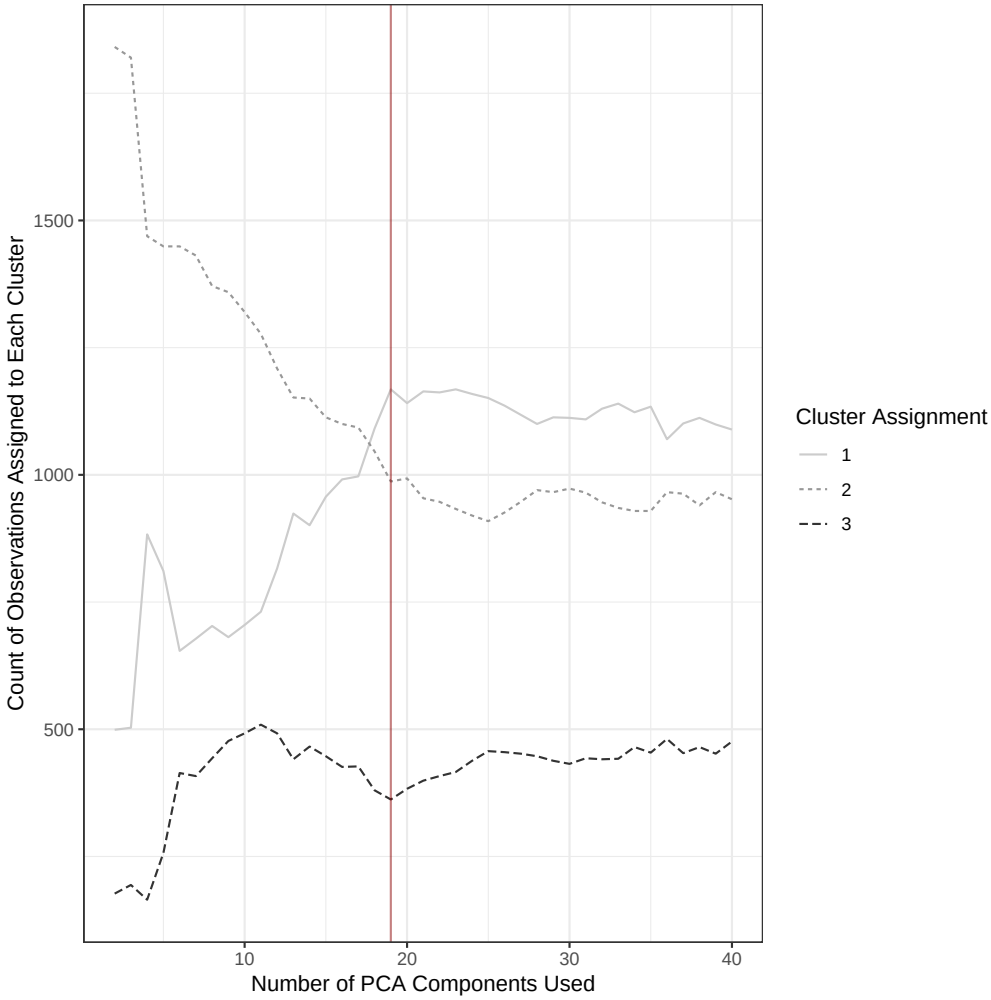


Figure B.2. PCA Components and Cluster Assignment: This figure shows the count of observations assigned to each cluster, using different numbers of PCA components. The x-axis indicates the number of PCA components used in the EM estimation, and the y-axis indicates the number of observations assigned to each cluster. The red vertical line indicates the number of components that we used in the analysis.

change substantially when post-treatment variables are dropped. In the case of Figure B.4b, the slow inattentive group is noisier and has a slightly higher ATE. However, in this particular survey, demographic questions were measured post-treatment, and these questions were substantively important for discerning different types of respondents, as inattentive respondents would likely breeze through them quite quickly, or maybe become distracted and take substantially longer. Therefore, these are likely to be high-variance questions. If researchers are concerned about introducing post-treatment bias into the analysis by stratifying on RTAC, then they should use incorporate this consideration into survey sequencing to ensure that they are not dropping meaningful question timers by excluding post-treatment questions.

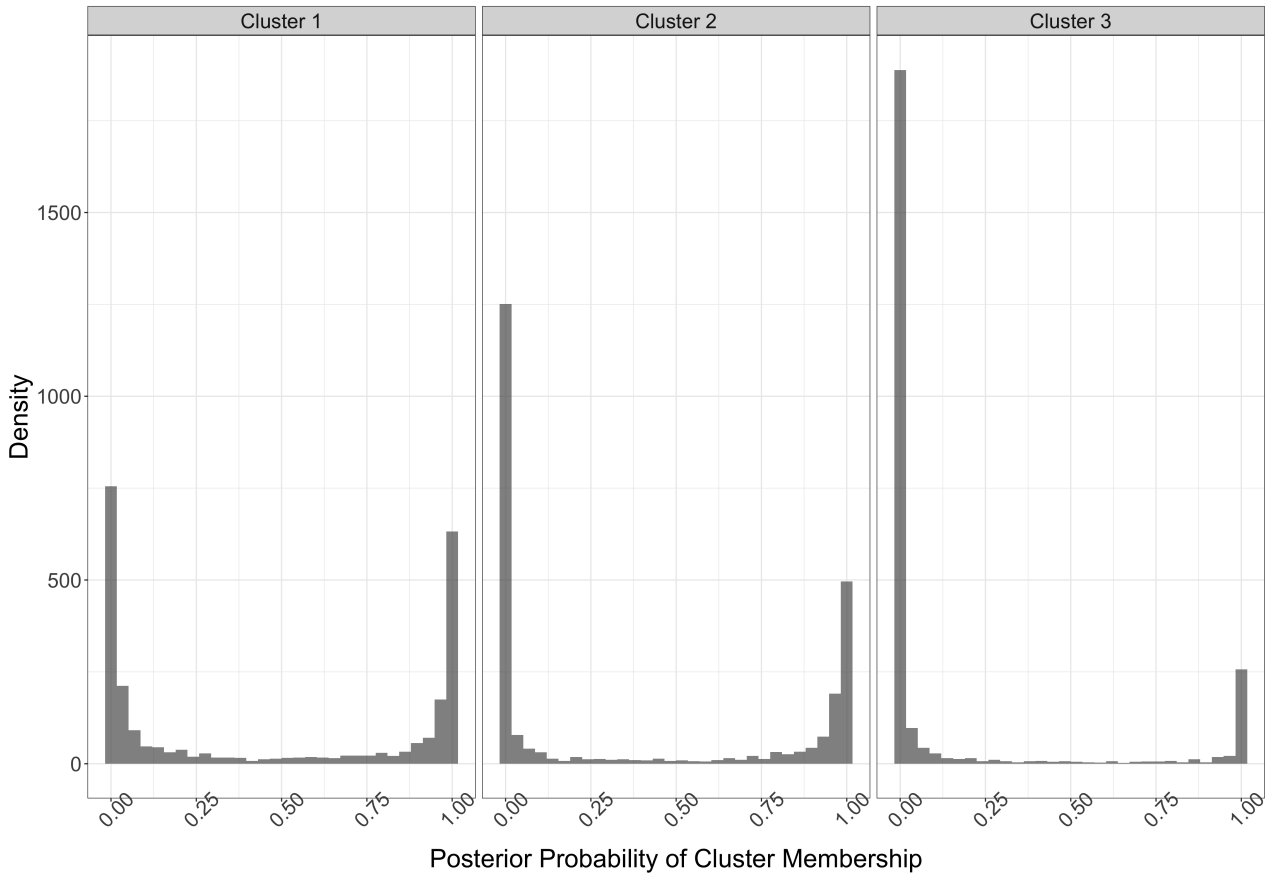
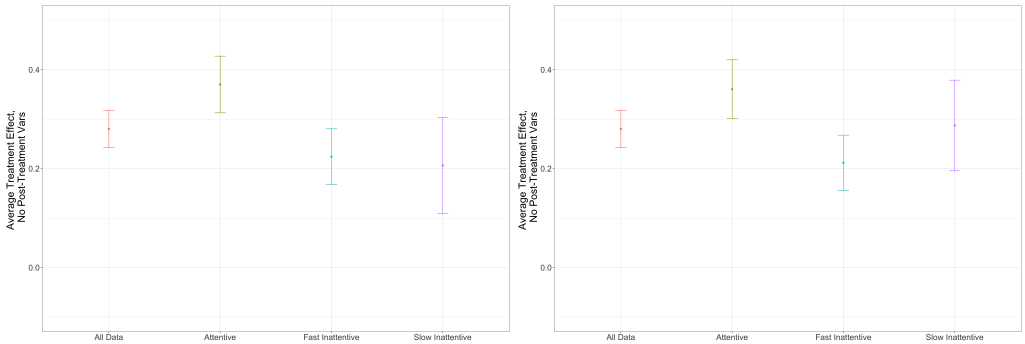


Figure B.3. Clear Assignment of Cluster Membership: This plot shows the responsibility of each respondent for each cluster, i.e., the estimated posterior probability that an individual belongs to a given cluster. Most respondents are assigned values close to 1 or 0, meaning that their probability of belonging to a cluster is either very high or very low. In other words, the algorithm is assigning clear cluster membership in most cases.



(a) (b)

Figure B.4. KT Distribution, with Post-Treatment Variables Dropped: Panel (a) replicates the Kahneman and Tversky validation exercise, after dropping response timers associated with the experiment. Panel (b) replicates the Kahneman and Tversky validation exercise after dropping *all* post-treatment timers.

C Data Sources and Replication

The analysis contained in this paper comes from three distinct surveys conducted between 2016 and 2019. The first sample was on SSI, while the second two samples were on Lucid. All were diverse national samples. The data in the body of the paper comes from the 2016 SSI sample, while the replication discussed below comes from two 2019 Lucid samples. The final sample is also used to measure attentiveness through word count in an attempt to understand whether slow respondents are satisficing or find taking the survey cognitively taxing. For each of these surveys, we conducted the RTAC analysis.

Year	Survey Firm	Purpose	Sample	Sample Size	Sample Size Without Missingness
2016	SSI	Main Analysis	Diverse National	3000	2517
2019	Lucid	Replication Analysis	Diverse National	2297	1914
2019	Lucid	Replication, Word - Count Analysis	Diverse National	1836	1498

Table C.1. Overview of data sources for main findings and replication.

We replicate our main results using a survey of 3000 respondents on Lucid, a survey research firm, conducted in the spring of 2019, and a second survey – also on Lucid – of just under 2000 respondents conducted in the summer of 2019. Question timers were embedded throughout the survey, which we analyze here to show that our method replicates outside of the original dataset. The first Lucid sample only contains one screener question and does not contain the Kahneman and Tversky replication, nor the reversed ideological scale measure, but we can still look at the distribution of respondent classification and correspondence with the screener question. The second Lucid sample contains the reversed ideological scale and the Kahneman and Tversky framing experiment.

Tables C.2 and C.3, and Figure C.1 show that the classification algorithm replicates the main results in the paper for both surveys. For both surveys, the tables shows that the three clusters have different distributions of global times, wherein the global time of the three different clusters corresponds to an interpretation of response time with slow, baseline, and fast respondents, and slow respondents who vary widely in their global response time behavior. In Figure C.1, we can see that the algorithm is, once again, able to assign each observation to a cluster with relative certainty in both the replication surveys. Most observations are either very likely or very unlikely to be sorted into a given cluster, as shown by the peak in posterior probability around 0 and 1 for each cluster. This lends confidence to our approach by showing that in two other surveys – conducted at different times, and with different online survey firms – the RTAC process behaves similarly.

We now turn to a comparison between the screener question and RTAC behavior in the first replication survey (Spring 2019). In Figure C.2, we show that, consistent with the main results of the paper, respondents classified as attentive are more likely to have passed the screener question. The results are noisier because we have to rely on only one screener question, this figure shows that respondents who were classified as attentive using RTAC were vastly

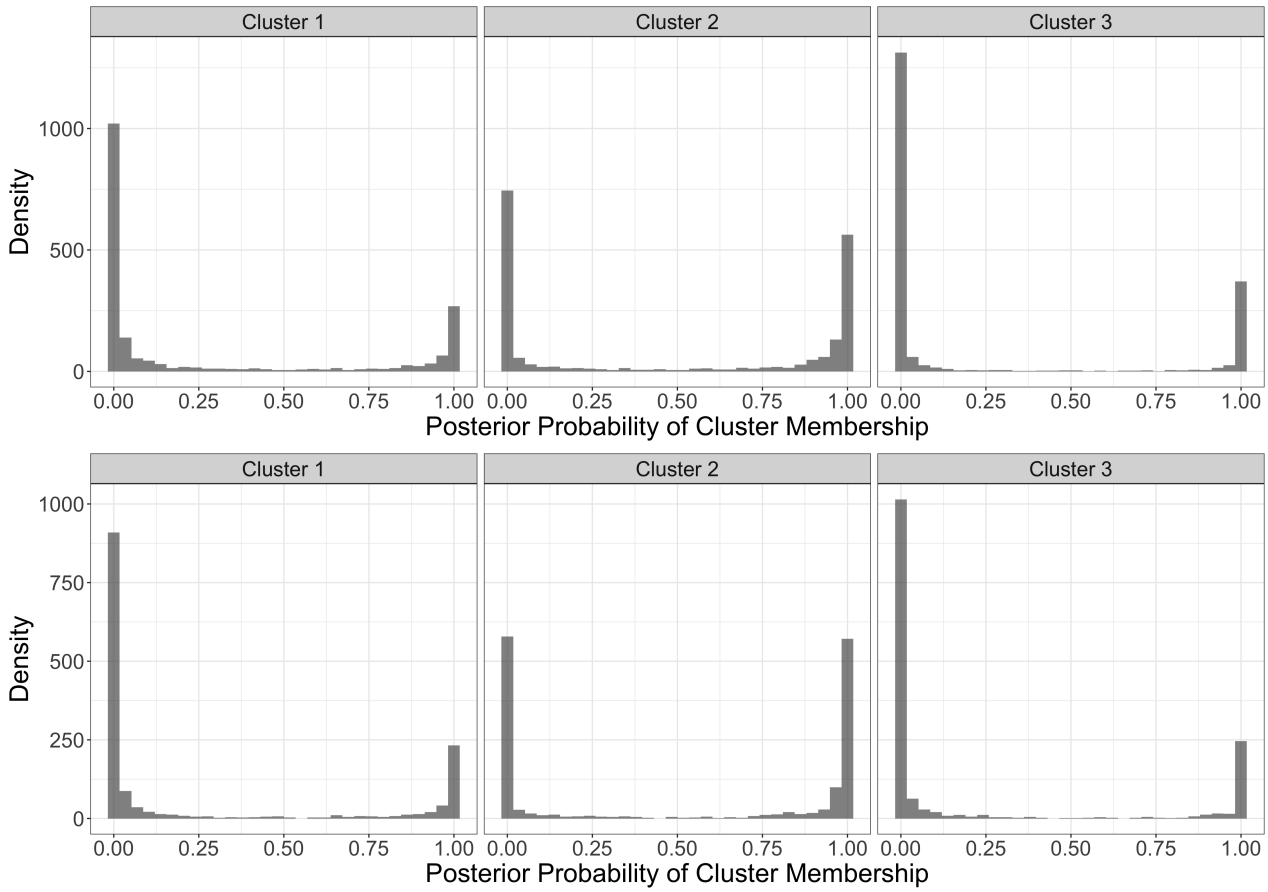


Figure C.1. These plots shows the responsibility of each respondent for each cluster, i.e., the estimated posterior probability that an individual belongs to a given cluster. The upper plot shows the responsibility distributions for the Spring 2019 survey, while the lower plot shows the responsibility distributions for the Summer 2019 survey. Most respondents are assigned values close to 1 or 0, meaning that their probability of belonging to a cluster is either very high or very low. In other words, the algorithm is assigning clear cluster membership in most cases

Cluster	Group Count	Average Time	Time Variance
Cluster 1	508	551.22	148196.12
Cluster 2	954	657.01	60895.01
Cluster 3	452	1352.53	6809446.32

Table C.2. Summary Statistics by Cluster Grouping (Spring Survey): The table shows cluster classification for the Spring 2019 Omnibus study, along with the frequency of cluster classification, the average global time within that cluster, and the variance within that cluster.

Cluster	Group Count	Average Time	Time Variance
Cluster 1	379	669.42	752208.07
Cluster 2	802	1029.90	317368.03
Cluster 3	317	1528.57	1647064.92

Table C.3. Summary Statistics by Cluster Grouping (Summer Survey): The table shows cluster classification for the Summer 2019 Omnibus study, along with the frequency of cluster classification, the average global time within that cluster, and the variance within that cluster.

more likely to have also passed screener questions, successfully replicating the main findings in the paper.

Finally, we replicate the Kahneman and Tversky, and the reverse ideological scale analyses from the main text of the paper. Although the experimental results are very noisy, both validation exercises replicate in the new Lucid dataset. Note that in the Kahneman and Tversky replication exercise, the noise in estimating the ATE point estimates means that there is no statistical difference between the three groups – particularly, the slow and the attentive groups. However, as we have shown, the algorithm does not perform when only two states are used, given the nature of behavior during online self-administered surveys, making the inclusion of a third group necessary to derive an estimate for attentive respondents.

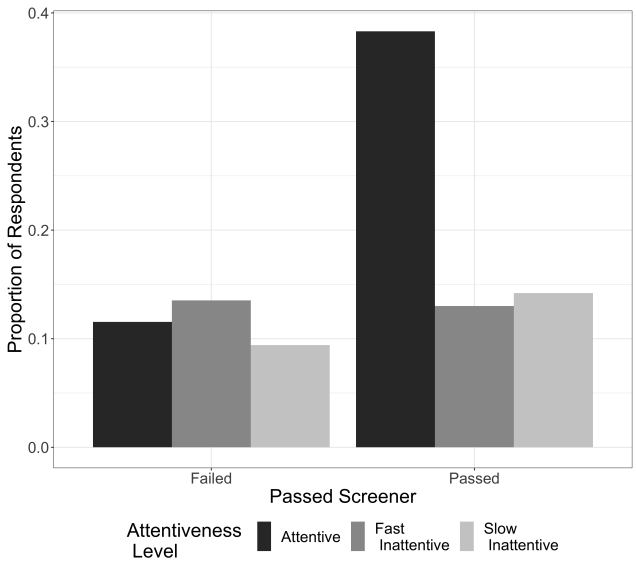
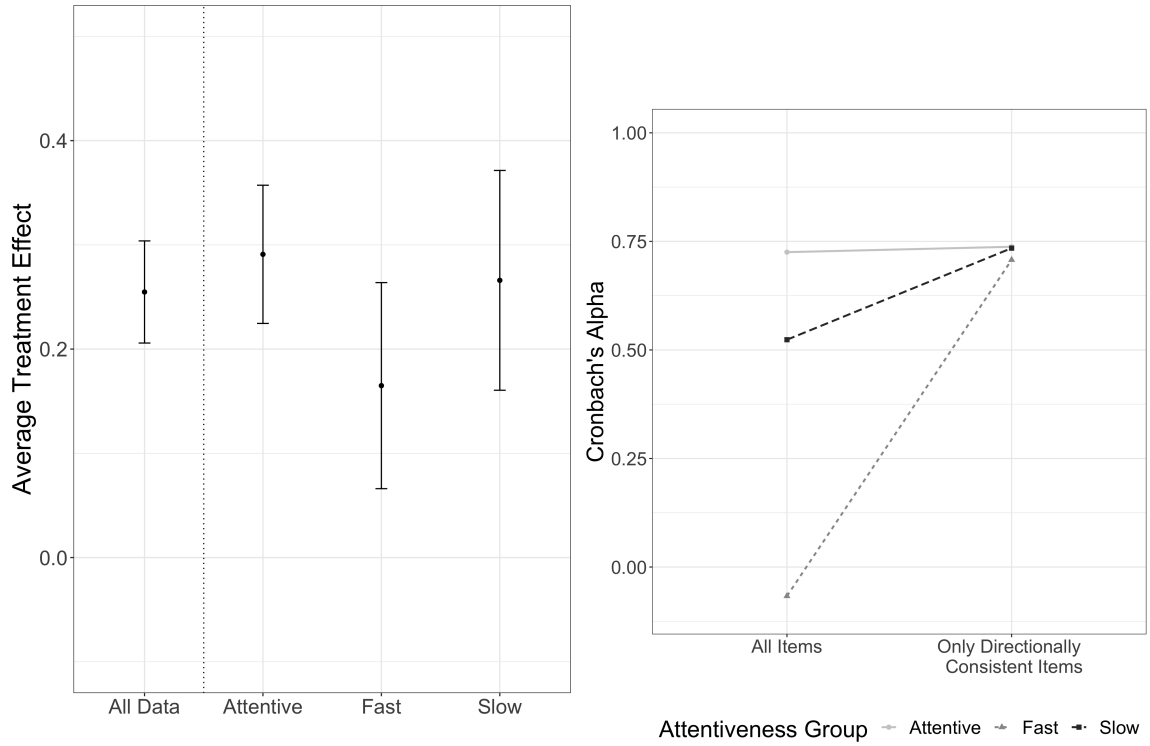


Figure C.2. Attentive Respondents are More Likely to Pass Screeners. Respondents who passed screener questions were more likely to also be classified as attentive.



(a) **Framing Experiment Results by Cluster, Summer 2019**

(b) **Cronbach's Alpha, Summer 2019**

Figure C.3. Replication of Figure 7 This left-hand figure shows the average treatment effect of a well-known and replicated framing experiment first introduced by Kahneman and Tversky (1981). Survey respondents in the attentive group display a high and significant treatment effect, whereas those assigned to one of the inattentive clusters display a much weaker and close to null effect. The dot-dash line indicates the naive average treatment effect using all the data. The right-hand figure shows Cronbach's alpha for three related ideology questions in which respondents had to position themselves on a scale ranging from liberal to conservative. Crucially, the scale was reversed for one of the questions, meaning that the correlation across the three questions will be lower for those respondents who simply click through the questions without carefully reading the instructions and thus noticing the change in the scale. While Cronbach's alpha is similar for all clusters when computed just over the two questions with the same scale, the coefficient dramatically decreases for all but the baseline cluster when computed over all three ideology questions, including the question with the reversed scale.

D RTAC Vignette

D.1 Formatting and Inspecting the Data

Survey data usually come in the form of a large data frame, in which each row corresponds to one survey respondent. In the data frame's columns, we usually find information ranging from individual respondent ID numbers, to their IP addresses, the start and finish time of the survey, response times for each question, and other data automatically saved by most survey programs, in addition to the actual responses given by each subject.[1]

	Respondent_ID	IP_address	start_date	state	Q1_answer	Q1_time	Q2_answer	Q2_time	Q3_answer	Q3_time
1	R_abcd	123.45.678.9	2020-01-01 06:08:33	MA	Yes	3.231	Somewhat oppose	8.729	Right amount	11.239
2	R_efgh	456.78.901.2	2020-01-01 12:34:16	TX	Yes	5.221	-	-	-	-
3	R_ijkl	789.01.234.5	2020-01-02 09:12:59	CA	No	3.331	Somewhat support	10.467	Too much	9.772
4	R_mnop	012.34.567.8	2020-01-03 04:46:27	NJ	Yes	2.432	Don't know	16.428	Too little	12.332
5	R_qrst	345.67.890.1	2020-01-03 07:50:11	FL	No	4.856	Don't know	12.478	Right amount	14.657

Figure D.1. Raw Survey Data .

We start by cleaning and formatting these data. In particular, we want to retain a data frame that contains just an identifier for each survey respondent who completed the survey, and the response time for each survey question they answered, usually measured in seconds to two or three decimal places. In other words, we want a data frame of size N x Q+1, where N is the number of survey participants who answered all questions and Q is the number of survey questions.

	Respondent_ID	Q1_time	Q2_time	Q3_time	Q4_time	Q5_time	Q6_time
1	R_abcd	3.231	8.729	11.239	4.345	16.342	7.234
2	R_ijkl	3.331	10.467	9.772	3.858	14.274	7.944
3	R_mnop	2.432	16.428	12.332	3.256	14.285	8.123
4	R_qrst	4.856	12.478	14.657	4.263	17.234	6.134
5	R_uvwx	2.639	11.673	10.101	2.373	16.844	7.765

Figure D.2. Formatted and cleaned data .

Before starting any serious analysis, we recommend inspecting the response time data and looking at descriptive statistics. Since the data are likely to be highly skewed - think of a few respondents who start the survey but then leave the computer to focus on some other task and don't return until hours later - we recommend logging the response times. It may then be helpful to look at the response time distributions for each questions. We do so by using the `R` packages `ggplot2` and `reshape2`. This exercise will often reveal that there is not just variation in response times within single questions with some respondents taking more time than others to answer the question but also across questions: For some questions, response times follow a tighter distribution with less variance, while for others there is a much wider distribution of times.

```
# Load necessary libraries
library(ggplot2)
library(reshape2)

# Melt data to work with ggplot
melted_d <- melt(formatted_d, id.vars = "Respondent_ID")

# Plot boxplots for each survey question
ggplot(melted_d, aes(x = reorder(variable, value, FUN = mean), y = value))
    + geom_boxplot()
```

1. This vignette features simplified toy data to provide the code commands for analysis. It does not directly run. To view the original code, readers should visit https://doi.org/10.1017/pan.xxxx.xx. to examine the paper's replication code.

```
  + coord_flip()
  + labs(main = "Distribution of Response Times across Questions",
         x = "Survey Question Label",
         y = "Response Time (in seconds, logged)")
```

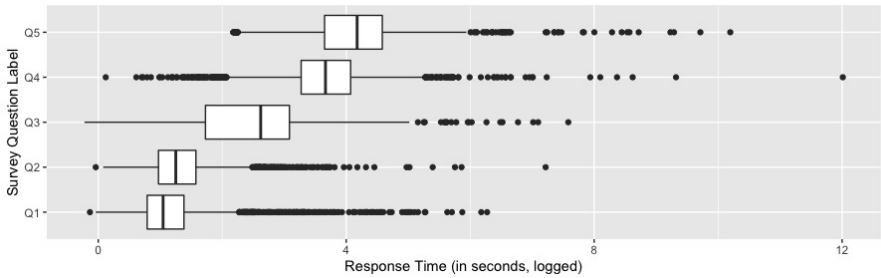


Figure D.3. Response Time Distribution for each Question .

D.2 Dimension Reduction

We can see from Figure D.3 that some questions are more likely to provide us with valuable information about the respondents' attentiveness, namely those where we can discern a good amount of variation in response time behavior, whereas others are less informative (think of a simple question asking the respondents' gender). To avoid having to come up with an ad-hoc measure of which questions are informative and which are not, we prefer to let the data speak. We therefore begin the analysis by taking high dimensional data – the response time for each question and survey taker – and condensing them such that the data are parsimonious while still capturing sufficient variation to characterize respondents. This focuses our analysis on those parts of the data where here we can find the most information about respondents' survey behavior. To do so, we perform a principal component analysis (PCA), using the `princomp` command in `R`.

```
pca <- princomp(formatted_d[,-"Respondent_ID"])
```

How many components of the PCA should we retain? To understand this part better intuitively, imagine first that there are just two questions, and that response times response times across these two questions are nicely correlated for each respondent. In other words, respondents either answer both question very quickly or very slowly. We plot such respondents in black on the left-hand side in Figure D.4. In this case, a single principal component, indicated by the black dotted line, is likely sufficient to capture most the the variation and indicate whether respondents are fast or slow.

Now imagine that there are actually some respondents who either take longer to answer question 1 but are then quick to answer question 2, or inversely give a quick answer to the first question but are then very slow to answer the second question. These respondents are represented by the red dots on the left-hand side in Figure D.4. Clearly, their survey taking behavior is very different from that of the black dots. Yet if we were to use just the one principal component represented by the dotted black line, we would likely overlook this behavior and assume these respondents take an average amount of time to answer questions. This is where including a second principal components, represented by the dotted red line, is helpful. It allows us to capture a completely different kind of variation (see right-hand side of Figure D.4).

Of course, our data are likely to contain many more than just two questions. We therefore recommend retaining enough principal components to capture at least 80% of the total variation contained in the data. We can simply compute the cumulative proportion of the total variation explained with each additional principal component. In our case, the first 19 components explain just over 80% of the overall variation.

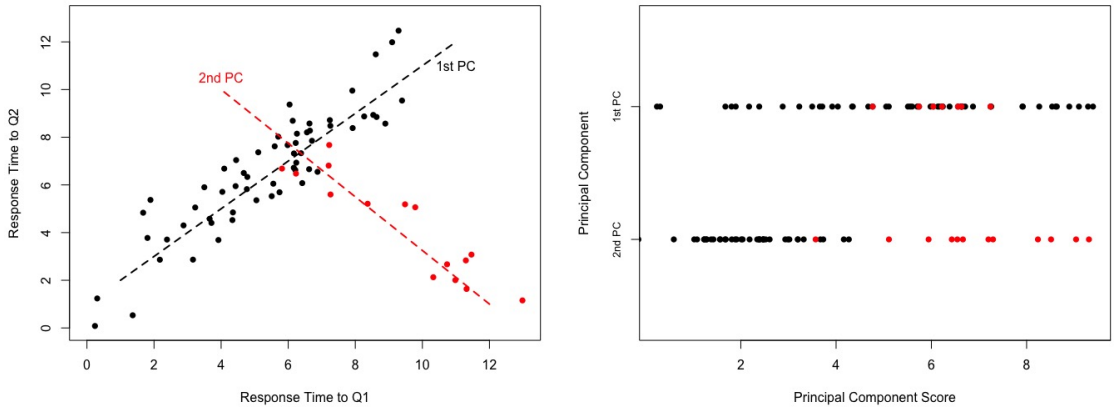


Figure D.4. Illustrative Principal Components: The left-hand side graph shows illustrative response times for two survey questions. The black dots depict respondents who behaved similarly across the two questions, meaning they either answered both questions quickly or slowly. The red dots depict respondents who behaved differently across the two questions, answering one question quickly and the other more slowly. The dotted lines represent principal components. The right-hand side figure illustrates what the first and second principal component scores might look like. It show that the second component is necessary to identify the particular behavior of the red dots.

```
cumsum(pca$sdev^2/sum(pca$sdev^2))
```

```
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
0.3969580 0.4559120 0.5009140 0.5332151 0.5585667 0.5811824 0.6021353
   Comp.8    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14
0.6220387 0.6413415 0.6601577 0.6782277 0.6955452 0.7127215 0.7290998
  Comp.15   Comp.16   Comp.17   Comp.18   Comp.19   Comp.20   Comp.21
0.7448594 0.7600798 0.7748366 0.7892449 0.8034999 0.8172378 0.8307881
  Comp.22   Comp.23   Comp.24   Comp.25   Comp.26   Comp.27   Comp.28
0.8441255 0.8560263 0.8677832 0.8791367 0.8904590 0.9014031 0.9117780
  Comp.29   Comp.30   Comp.31   Comp.32   Comp.33   Comp.34   Comp.35
0.9219609 0.9318473 0.9407003 0.9494580 0.9579505 0.9660084 0.9725624
  Comp.36   Comp.37   Comp.38   Comp.39   Comp.40
0.9785273 0.9843232 0.9899062 0.9950227 1.0000000
```

D.3 Clustering Algorithm

In the second step, we use a clustering algorithm, namely an expectation maximization algorithm, to estimate the latent attentiveness of each survey respondent. We use the `mvnormalmixEM` command, contained in the `mixtools` package, which provides a reliable EM algorithm for mixtures of multivatriate normal distributions. The EM algorithm will iteratively determine the shape of each distribution in terms of their mean and variance, as well as the probability of belonging to each distribution for each individual respondent, captured in each distribution's relative density or responsibility.

We first need to determine the number of underlying distributions we expect in the data. Following our theory on fast-inattentive, slow-inattentive, and attentive respondents, we build a model consisting of three distinct normal distributions, i.e., attentiveness clusters. We then need to tell the command the initial parameters of each distribution, which the algorithm will use during its first iteration. The actual input values are of little importance since the algorithm will adjust them according to the data with each iteration.

- We specify the initial mixing proportions λ, i.e., the share of respondents that might belong to each of the three

attentiveness groups.

- We then specify the initial values for the mean parameters μ. To place the starting position of each cluster not too close to one another, we use the first, second, and third quartile of the PCA scores as the initial means for the first, second, and third attentiveness cluster. Another popular method to set initial mean values μ is to run a k-means analysis on the data.
- We next specify the initial values for the variance-covariance matrices σ by simply using the identity matrix.
- Then, we specify the convergence criterion ϵ. An algorithm is considered to have converged when the estimated parameters become stable and do not change from iteration to iteration. However, to save time and computational power, it is common practice to stop the algorithm when the estimated parameters change by less than a small constant, namely ϵ. Here, we put ϵ at 0.01.
- Finally, we specify the maximum number of iterations. In case the difference in estimated parameter values remains larger than the specified constant ϵ, the algorithm will stop once the maximum number of iterations is reached. We set this maximum to 10,000.

```
library(mixtools)

em <- mvnormalmixEM(x = pca,

        lambda = c(1/3, 1/3, 1/3),

        mu = list(
             as.numeric(apply(pca, 2, function(x) quantile(x, 0.25, na.rm=T))),
             as.numeric(apply(pca, 2, function(x) quantile(x, 0.50, na.rm=T))),
             as.numeric(apply(pca, 2, function(x) quantile(x, 0.75, na.rm=T)))
                  ),

        sigma = list(
                    diag(1, ncol(pca)),
                    diag(1, ncol(pca)),
                    diag(1, ncol(pca))
                    ),

        epsilon = 1e-02,

        maxit = 10000)
```

Once the algorithm has run and converged, it will have calculated each attentiveness cluster's responsibility, i.e., the posterior probability that survey respondents belong to it. Since an EM algorithm produces probabilities rather than hard cluster assignments, it is useful to inspect the distribution of that probability (see Figure D.5). Very high and low probabilities indicate that the algorithm is fairly certain about attentiveness cluster membership.

```
responsibilities <- em$posterior

responsibilities_melted <- melt(responsibilities)

ggplot(responsibilities_melted, aes(x = value)) +
  geom_histogram(alpha = 0.7) +
  labs(x = "Posterior Probability of Cluster Membership",
       y = "Density") +
```

```
  facet_wrap(~ Var2)
```

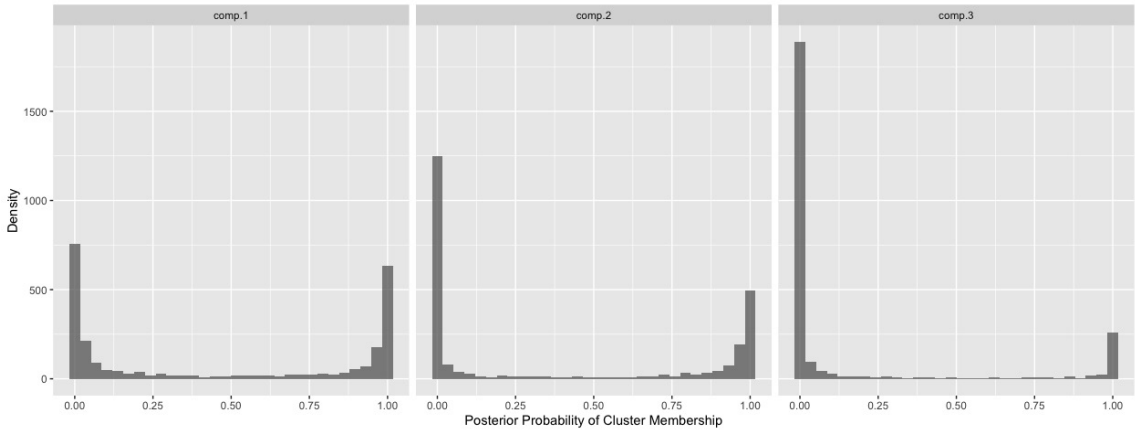


Figure D.5. Inspecting the estimated probabilities of attentiveness cluster membership .

We can then use the posterior probabilities as continuous measures of attentiveness cluster membership or simply hard assign the most likely cluster to each respondent. To assign respondents to attentiveness groups, we simply look at the mean response time for each attentiveness group and assign groups accordingly.

```
max_assign <- apply(responsibilities, 1, function(x) which.max(x))

classify <- cbind.data.frame(formatted_d, max_assign

assign_groups <- classify %>%
  group_by(max_assign) %>%
  mutate(group_mean = mean(total_time),
         group_var = var(total_time))
```