Appendices

Supplemental Information: Three Privacy Protection Demos

Statistical Disclosure Control Tools: `sdcMicro`

Various tools exist for measuring and implementing k-anonymity and other statistical disclosure control principles in microdata. One highly-developed suite of tools in `R` is the `sdcMicro` package (Templ et al., 2020). This application shows how to use `sdcMicro` to measure de-anonymization risk in nominally anonymous data, and how to implement *non-perturbative* changes in microdata to decrease the risk of de-anonymzation.[19]

I use data from a massive household survey of Indian citizens, the India Human Development Survey II (IHDS-II) (Desai and Vanneman, 2015), to demonstrate how `sdcMicro` can be used to decrease the risk of de-anonymization in sensitive data. IHDS-II surveys over 200,000 individuals in more than 40,000 households across all 35 states and union territories (prior to the creation of Telangana and the dissolution of Jammu and Kashmir), covering standard demographic information, household finance, education, health, and a wide range of other topics. I use a small subset of the data to construct a statistical disclosure "problem": a range of quasi-identifying variables for which risk must be gauged and disclosure-mitigation steps taken, and a range of sensitive variables for which the values should not be matchable to specific individuals.

Though the SDC literature—which grows out of technical research at statistical agencies and the International Household Survey Network—primarily focuses on quasi-identifying variables that are part of public record, I use a broader set of quasi-identifiers that are relevant to protecting research participants from de-anonymization by knowledgeable local partners. When local knowledge is at play, it is worth including variables like religion, caste, marital status, etc. for which a research partner with substantial local knowledge would know the values *without consulting public record*.[20]

For this example, I use the quasi identifiers of age, state, district, village name, marital status, and caste or religion, along with the potentially sensitive information of how much income a respondent receives annually from government schemes and benefits. IHDS-II wisely replaces village names with a unique code in their data to prevent re-identification. For the purpose of this example, I treat the codes as identifiable, even though they are not.

Ingredients:

1. Dataset with quasi-identifiers

[19] Perturbative methods like value swapping, post-Randomization, and simple additive noise are also implemented in `sdcMicro`, but it seems they have been overtaken in popularity by more sophisticated differential privacy algorithms with more elegant statistical properties.

[20] In many parts of India, it is actually conceivable that religion and caste are part of public record (with some uncertainty) given naming conventions. Many observant Sikh women, for example, take the name Kaur as either a middle name, or in place of their family name. The male equivalent Singh is a weaker signal of religious identity.

2. A computer running R 2.10 or newer

3. An installation of the `sdcMicro` package from CRAN—this demo uses version 5.5.1[21]

Using `sdcMicro`:

1. Set up your "SDC Problem" by creating an SDC Object:

   (a) Load necessary packages and import data as a `data.frame` object

   ```
   library(readr); library(sdcMicro)
   ihds <- read_tsv('/your/file/path.tsv')
   ```

   (b) Create an SDC object using your data. SDC objects take a number of arguments. See comments in the code chunk below for a brief description of each

   ```
   sdc <- createSdcObj(# Your microdata, as a data.frame object
                       dat = ihds,
                       # Column names: categorical quasi-identifiers
                       keyVars = c("district", "male", "mar_stat",
                                   "rel_caste", "state", "vill_code",
                                   "age"),
                       # Column names: numeric quasi-identifiers
                       numVars = NULL,
                       # Cluster ID
                       hhId = IDPSU,
                       # Vector of sample weights
                       weightVar = WT,
                       seed= 02139)
   ```

   (c) Print the SDC object for an initial read-out of the unicity of records in the dataset. Pay special attention to two features: the proportion of records that violate k-anonymity for $k \in \{2,3,4\}$, and the size of the smallest categories for your key variables.

   ```
     print(sdc)
    The input dataset consists of 204376 rows and 13 variables.
    --> Categorical key variables: district, male, mar_stat, rel_caste,
    state, vill_code, age
    --> Weight variable: WT
    --> Cluster/Household-Id variable: IDPSU
   ----------------------------------------------------------------------
   Information on categorical key variables:

   Reported is the number, mean size and size of the smallest category
   >0 for recoded variables.
   In parenthesis, the same statistics are shown for the
   ```

[21] Use this code in a .R script to install and load. `if (!require('sdcMicro')) install.packages('sdcMicro'); library('sdcMicro')`.

```
unmodified data.
Note: NA (missings) are counted as seperate categories!

 Key Variable       Number of categories      Mean size
   district              372(372)     549.398(549.398)
   male                 2(2)        102188.000(102188.000)
   mar_stat              6(6)        34062.667(34062.667)
   rel_caste             7(7)        29196.571(29196.571)
   state                33(33)       6193.212(6193.212)
   vill_code             39(39)       5240.410(5240.410)
   age                  100(100)     2043.760(2043.760)
 Size of smallest (>0)
          29    (29)
       101964 (101964)
         341   (341)
        5388   (5388)
         272   (272)
          58    (58)
           5     (5)
---------------------------------------------------------------------
Infos on 2/3-Anonymity:

Number of observations violating
  - 2-anonymity: 147648 (72.243%)
  - 3-anonymity: 187432 (91.709%)
  - 5-anonymity: 202378 (99.022%)
---------------------------------------------------------------------
```

2. Now, begin modifying the data to reduce identifiability. Start by recoding variables that have a large number of small "bins," like age, to be less granular. The function globalRecode, applied to your SDC object, will recode specified variables to be less granular.[22]. Simply specify the SDC object, the column you want to recode, and what you want the new categories to be. Then print the SDC object to evaluate the effect of recoding on k-anonymity. When we recode "age" from specific ages to decade bins, the number of observations that are unique across our large number of quasi-identifiers drops from 72% of the dataset to 21% of the dataset. More gains are possible from this single operation by creating even wider bins for age, but wider bins are less useful for analysis. Consider also using the functions topBottomCoding() and groupAndRename() to provide similar functions for numerical and categorical variables, respectively.

```
sdc <- globalRecode(sdc, column = "age",
                    breaks = seq(from=min(sdc@manipKeyVars$age),
                    to=max(sdc@manipKeyVars$age), length.out = 10))
print(sdc)
```

[22] Counterintuitively, the function microaggregation() does something else

```
Infos on 2/3-Anonymity:

Number of observations violating
  - 2-anonymity: 44088 (21.572%) | in original data: 147648 (72.243%)
  - 3-anonymity: 77362 (37.853%) | in original data: 187432 (91.709%)
  - 5-anonymity: 123333 (60.346%) | in original data: 202378 (99.022%)
----------------------------------------------------------------------
```

3. Once satisfied with recoding, try value suppression. The function `localSuppression()` implements an algorithm to prune the dataset into k-anonymity (where k is an argument supplied by the user) by suppressing *individual values* of quasi-identifier variables. The algorithm used by the package suppresses quasi-identifier values for particular observations that have the highest risk of de-anonymization in the existing format of the data. Users can (and should) use the "importance" argument in the function, in order to constrain the algorithm's choice about which variables to suppress in a given observation. Variables ranked as most important are used as last-resort suppression. Users should also note that `localSuppression()` runs slowly, especially for large datasets and datasets that have a high number of key variables. It continues pruning until k anonymity is achieved for 100% of observations. Note that in order to achieve 3-anonymity across 7 key variables (an unusually high number), 89,865 values are suppressed—roughly 44 cells for every 100 observations in the dataset. When suppression functions this aggressively, users should consider deleting certain quasi-identifier variables entirely, or using perturbative techniques like post-randomization or one of the variety of available differential privacy algorithms. Note, also, that the variables specified as high-importance in the function are suppressed very sparingly. Specifying theoretically important variables as "high importance" during local suppression minimizes the rate at which observations in SDC-treated data will drop out of key regressions due to missingness.

```
sdc <- localSuppression(sdc, k=3,  importance = c(6,1, 2, 3, 7, 5, 4))
# which vars (rank in order of sdc@keyvars) should be maintained?
# Varibles with higher "rank" (1-n) will be last for suppression

sdc      # To confirm k-anon and see what was suppressed

The input dataset consists of 204376 rows and 13 variables.
  --> Categorical key variables: district, male,
  mar_stat, rel_caste, state, vill_code, age
  --> Weight variable: WT
  --> Cluster/Household-Id variable: IDPSU
----------------------------------------------------------------------



Information on categorical key variables:

Reported is the number, mean size and size of the smallest
category >0 for recoded variables.
```

```
In parenthesis, the same statistics are shown for the unmodified data.
Note: NA (missings) are counted as seperate categories!


 Key Variable Number of categories      Mean size
 district               373 (372)   496.634    (549.398)
    male                  3   (2) 102177.000 (102188.000)
 mar_stat                 7   (6)  33936.833  (34062.667)
rel_caste                 8   (7)  29154.143  (29196.571)
   state                34  (33)   4254.879   (6193.212)
vill_code                40  (39)   5137.103   (5240.410)
     age                10 (100)  22251.556   (2043.760)
 Size of smallest (>0)
             20     (29)
          101951 (101964)
            308    (341)
           5345   (5388)
            308    (272)
             12     (58)
            351      (5)
-------------------------------------------------------------------------


Infos on 2/3-Anonymity:

Number of observations violating
  - 2-anonymity: 0 (0.000%) | in original data: 147648 (72.243%)
  - 3-anonymity: 0 (0.000%) | in original data: 187432 (91.709%)
  - 5-anonymity: 42928 (21.004%) | in original data: 202378 (99.022%)

-------------------------------------------------------------------------


Local suppression:

  KeyVar | Suppressions (#) | Suppressions (%)
district |           19628 |           9.604
male     |              22 |          0.011
mar_stat |             755 |           0.369
rel_caste|             297 |           0.145
state    |          63965 |          31.298
vill_code|            4029 |           1.971
age      |           1169 |          0.572
-------------------------------------------------------------------------
```

4. After recoding and suppressing, users should re-measure disclosure risk before exporting mod-

ified datasets. `sdcMicro` provides various metrics for disclosure risk, nicely summarized in a print function. There does not seem to be a universally accepted threshold for how much risk is tolerable, but researchers should decide on thresholds they feel they can defend. Risk measures, plus a full summary of changes can also be output as a report.

```
measure_risk(sdc) # This runs slowly
   report(sdc, internal = T, verbose = T) # generates HTML report
print(sdc, "risk")

Risk measures:

Number of observations with higher risk than the main part of the data:
  in modified data: 0
  in original data: 0
Expected number of re-identifications:
  in modified data: 9.99 (0.00 %)
  in original data: 357.17 (0.17 %)

Information on hierarchical risk:
Expected number of re-identifications:
  in modified data: 1023.10 (0.50 %)
  in original data: 31574.67 (15.45 %)
------------------------------------------------------------------------
```

5. Users should also consider measuring l-diversity, a measure of disclosure risk related to k-anonymity. l-diversity measures, for a group of k observations that have identical values across a set of quasi-identifiers, the number l of well-represented values for some sensitive attribute. A dataset is l diverse if every group of k observations is represented by l different values for a sensitive attribute. In practical terms, if a 3-anonymous dataset is only 1-diverse for some sensitive attribute, an adversary looking for a person known to be represented in the dataset and having known quasi-identifiers might be able to learn sensitive information about the person simply because all people who share a set of quasi-identifiers also share a value for sensitive information. Within reason, higher l-diversity is better for privacy. Given the unusually high number of quasi-identifiers in this example, achieving high l-diversity would require very drastic modifications to the data.

```
print(ldiversity(d_sub_new, # New dataset
   keyVars = c("male", "age", "rel_caste",
   "district", "state", "mar_stat"),
   ldiv_index = "ben_income")) # Sensitive Variable
--------------------------

L-Diversity Measures

--------------------------
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   3.882   5.000  37.000
```

6. Once satisfied (perhaps after multiple iterations through the above steps) users can export their data, though the process is made slightly cumbersome by `sdcMicro`. All variables other than the quasi-identifier/key variables must be re-assembled separately from the original dataset as they are unchanged during the process.

```
ihds_new <- cbind.data.frame(ben_income = ihds$ben_income,
disab_pension = ihds$disab_pension,
hhid = ihds$hhid,
resp = ihds$resp,
WT = ihds$WT,
sdc@manipKeyVars)
```

7. The "information" costs of the SDC modifications are measured in terms of distance between the old and new values of continuous values (and differences in eignevalues) the report described in step 4, but users looking to measure the "information" costs in more practical terms or for mostly categorical variables should consider comparing the performance of pre- and post-modification data in substantively meaningful regressions. Unlike other privacy tools that rely on simulation or noise to obscure sensitive information, SDC tools *will* change the central tendencies and dispersion of key variables. The relevant question, then, is whether the change is tolerable for the purposes of the research. Table 1 shows the difference in regression coefficients for the same OLS model fit to pre- and post-modification data. The differences, depending on perspective, are substantial, and tens of thousands of observations are dropped because NAs have been induced in the course of local suppression. Whether these differences are acceptable, either for primary analysis or for sharing data, is up to the researcher.

Privacy Protection with Qualitative Data: Topic Modeling on Small Corpora

Unlike `sdcMicro` and the PGP lockbox, this final demonstration focuses on a tool for privacy-preserving presentation of text data—especially text data in small corpora that are primarily collected for qualitative analysis.

I use structural topic modeling to accomplish this task. Topic modeling helps identify patterns in the contents of documents under a set of assumptions about the relationship between semantic choice and meaning: topic models (starting with Latent Dirichlet Allocation in Blei et al. (2003)) model the appearance of a given word in a document as a function of some latent or unobserved category, a "topic" that the word is used to describe. A fitted topic model produces summaries for each document: a vector (summing to 1) of topic proportions which describes the prevalence of each latent category in a document. Per Grimmer and Stewart (2013), identification of the substantive meaning of a topic/cluster returned by the model is the responsibility of the researcher, not the model. The topic prevalence can be compared across documents to identify patterns in the ways that topics relate to each other—when a document discusses topic 1, it is also likely to discuss topic 8—and with structural

Table 1: Comparison between regressions on pre-modification and post-modification data

	Dependent variable:	
	ben_income	
	Pre-modification	Post-modification
disab_pension	5,073.788*** (237.984)	4,872.312*** (327.661)
rel_caste - Forward caste	232.416*** (67.173)	408.418*** (105.006)
rel_caste- OBC	258.449*** (63.640)	381.109*** (100.315)
rel_caste - Dalit	467.991*** (65.320)	617.885*** (101.783)
rel_caste - Adivasi	347.042*** (79.689)	518.968*** (119.222)
rel_caste - Muslim	109.470 (71.851)	233.291** (108.347)
rel_caste - Christian, Sikh, Jain	252.876** (106.496)	594.696*** (163.238)
mar_stat - Married	−354.006*** (96.193)	−315.579* (187.374)
mar_stat - Unmarried	−21.081 (99.345)	−275.114 (195.852)
mar_stat - Widowed	127.066 (111.180)	−25.319 (238.022)
mar_stat - Separated/Divorced	−207.232 (208.632)	−95.489 (607.482)
mar_stat -Married no gauna	476.453 (325.016)	−942.231 (576.728)
age (numeric)	11.078*** (1.083)	
age(11,22]		196.695*** (42.749)
age(22,33]		52.223 (77.683)
age(33,44]		80.673 (91.009)
age(44,55]		−45.490 (95.144)
age(55,66]		299.156*** (113.114)
age(66,77]		1,127.492*** (171.932)
age(77,88]		1,095.003*** (352.118)
age(88,99]		461.485 (960.741)
male	−95.988*** (26.038)	−38.081 (31.644)
Observations	204,376	127,955
R^2	0.050	0.049
Adjusted R^2	0.048	0.046
Residual Std. Error	5,702.491 (df = 203990)	5,508.105 (df = 127562)
F Statistic	27.772*** (df = 385; 203990)	16.787*** (df = 392; 127562)
Note:		*p<0.1; **p<0.05; ***p<0.01

topic models, but not all other varieties of topic model, topic prevalence can be related to document metadata to identify further patterns—respondents over the age of 35 have higher topic prevalence for topic 1 than respondents under the age of 35.

The benefit of STM for privacy preservation is that the main data format that must be shared in order to reproduce analyses, the Document-Term Matrix, naturally makes de-anonymization difficult in its standard pre-processing steps. Table 2 shows the DTM realization of a document analyzed using STM for privacy preservation in Milliff (2020). Though a motivated reader could learn *something* about the themes discussed in the document by reading the DTM alone, it would be extremely difficult (likely not possible with any degree of certainty) to reconstruct the document to the extent that contextual knowledge could be used to re-identify the respondent. Turning a DTM back into a document would require an adversary to: 1) reverse the process of stemming—turning stems back into words with proper conjugation and declension; 2) re-arrange the words into the order they appeared in the document and re-insert meaning-critical punctuation (especially full stops); and 3) re-conjure the missing stop words like articles, personal pronouns, direct object and indirect object pronouns, etc.

Stem	Count
anywher	1
carri	4
church	4
day	2
doesnt	1
even	1
everi	1
everywher	1
garbag	1
gun	1
happen	1
kill	1
laundromat	1
littl	1
mean	1
much	1
news	1
nothing	1
one	1
period	1
realli	1
see	1
shot	1
sometim	1
start	3
street	1
take	1
time	1
took	1
wife	1
without	1
work	1
wouldnt	1

Table 2: DTM vector corresponding to a single document in the corpus.

Another of the major benefits to STM as a deanonymization-prevention tool is user-friendliness. The optimization algorithm that fits structural topic models is complex, but using the `stm` package in

R is straightforward, especially with thorough instructions in the package vignette by Roberts et al. (2018).

Using STM to prevent de-anonymization follows largely the same steps as normal use for digesting large, public corpora. The important modifications come in pre-processing and presentation of the model findings.

First, researchers fitting topic models to sensitive data should do an additional set of pre-processing in order to eliminate personal identifiers before using STM's built-in tools to stem the text, remove stopwords, and create a DTM. The process of creating a DTM is likely to do a fairly good job of removing identifiers in its standard function. Identifiers, by definition, occur in one or very few records, so STM pre-processing may automatically drop them as sparse terms. Because identifiers are particularly risky, though, additional steps should be taken to ensure they are cut out of the data. Two possibilities exist: larger corpora could be stripped of identifiers using a Named Entity Recognition (NER) model like the pre-trained models in the python library `spaCy`. The NER model uses statistical (as opposed to rules-based) entity recognition to identify spans of text indicating people's names, particular locations, etc. A researcher could use the pre-trained tool to find and delete information like names and locations that is unique enough to aid de-anonymization and too unique to provide much value in the topic model fitting. NER models are likely to remove identifying information, but not certain. Instead of NER models, researchers could also use brute force: for corpora that are small enough to read, researchers could go through and manually delete identifiers like addresses, cross streets, names of people and locations, in order to ensure they do not end up in the model fit. This process is more labor intensive, but provides better assurances.

Second, for STM specifically because it allows users to estimate topic contents and prevalence as a function of document-level covariates, researchers must take steps—perhaps including the statistical disclosure control tools shown above—to ensure that the prevalence and content covariates they include (and which would be necessary to reproduce the model) are not easy to de-anonymize. The same cautions about disclosure control apply to document-level covariates which are used after model fitting to estimate the association between topic prevalence and respondent characteristics.

Third, researchers should be aware of the importance of un-processed documents in interpreting STM and other topic models. The topics that are generated by a topic model are not guaranteed to be substantively meaningful, and they require substantial interpretation by the user to figure out what, if anything, they mean. One accepted way to label the topics is reading the documents that have the highest proportions for each topic, and then deciding what thematically links those documents (Grimmer and Stewart, 2013). Verifying the interpretation of the model, therefore, is easiest if some documents are shared. Researchers have two choices for dealing with this. First, they might take their chances with refusing to share full documents given privacy concerns. It is uncommon to share interview notes for qualitative interviews as part of "replication files," so researchers might be able to avoid sharing STM documents as well. Second, researchers can split the longer interviews into shorter documents (even paragraph length works) and preserve the order and respondent information by specifying them as prevalence/content covariates in the STM. Under this system, the documents that might be shared to verify model interpretation would be sufficiently short to lessen the risk of de-anonymization. Of course, the most transparent path still poses some de-anonymization risk, and is potentially a weak point in the attemopt to use STM for privacy preservation.

The remainder of this demo shows the topic model fit from Milliff (2020), which uses STM to present trends in the contents of interviews about emotional and political responses to violent trauma. The sensitive data used in the topic models are the transcripts of 31 in-depth interviews (semi-structured) conducted in January 2018 with the surviving relatives of homicide victims who were killed between 2015 and 2017 in Chicago, IL. In the interviews, which lasted between 90 and 180 minutes, respondents share their experiences of trauma, their interactions with the state, and their thoughts on the causes of violence with surprising candor. Respondents were recruited with help from a non-academic partner: a social service organization that provides free case management and services to families of homicide victims.

A tool like STM is useful for sharing the results gleaned from these interviews because the views and experiences shared in the interviews are potentially sensitive—perhaps the most sensitive are assignations of blame for the death of a family member—and because the narrative format of the interviews would make re-identification possible even if identifiers like name, place, dates, etc. were deleted. Staff from the partner organization would be able to easily re-identify respondents given full interview transcripts. Some respondents would be identifiable by a broader audience as well: a number of the homicides discussed in the interviews were covered in local press or memorialized in music.

The goal of this topic model is to show, in a transparent and reproducible way, how the author reached conclusions about the correlates of anger at the perpetrator of homicide vs. anger at other targets based on primarily qualitative analysis of the interviews. An STM fit at the paragraph level with ten topics shows that discussion of anger (topic 5) is positively correlated with conversations about the motive behind the homicide (topic 3) and that when respondents are talking about confusion with respect to what happened (topic 6) they are not using words from the anger topic.

The same model can also be used to estimate associations between respondent-level metadata and topic prevalence. Since respondent transcripts are broken into many paragraphs, these estimates group documents by respondent. This presentation supports qualitative analysis about *who* and *what circumstances* were most likely to be associated with high levels of anger directed at the perpetrator.

STM results in this application are not a stand-alone presentation of the rich interview evidence in this application. In Milliff (2020), STM results support traditional qualitative interpretation of evidence and single case vignettes—themselves carefully written to avoid including information that could be cross-referenced against public sources—by showing that key patterns obtain across the whole sample, and are not cherry picked from particularly evocative interviews or dramatic stories. The paper further negotiates between privacy protection and transparency by including the "top

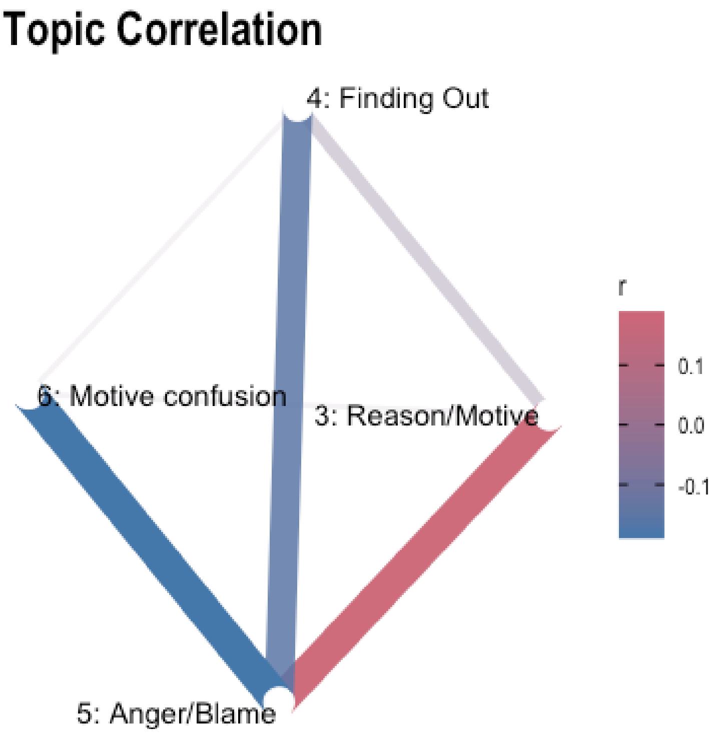


Figure 1: Inter-topic correlation for topics where $r > 0.1$ with Topic 5 (anger, blame).

document” paragraphs for each topic. The author read the 25 top documents in order to label each topic—three of the top 25 are included in an appendix of the paper.

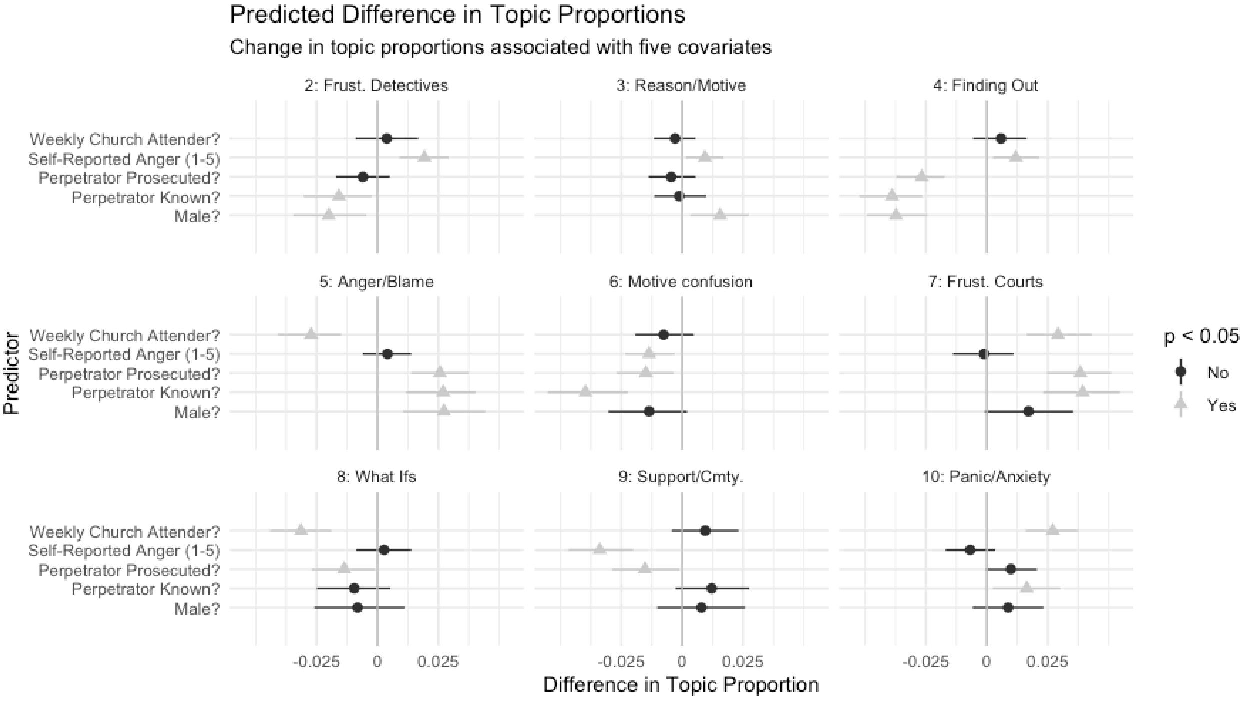


Figure 2: Bivariate associations between respondent characteristics and topic proportions

PGP "LockBox"

PGP encryption uses a pair of keys (strings of alphanumeric characters) called the "public key" and the "private key" to encrypt and decrypt information. The intuition behind the encryption is that any files encrypted using a particular public key can *only* be decrypted using the corresponding private key (See Foundation, 2014, for a good introduction). While PGP is most commonly used to encrypt email traffic (B sends a message to A that is encrypted with A's public key; A uses her private key to decrypt B's message), social scientists can use it to create a "vault" that they can deposit into easily, but cannot access in the field.

To create a vault, a researcher must first generate a key pair,[23] and then stores the private key on a local drive, at their home institution or somewhere else that is not accessible during data collection. It is crucial that the private key *not* be accessible to the researcher once she is in the field; this means it cannot be stored on the cloud, available in an email, or carried with the researcher on local media like a USB drive. Once in the field, the researcher can use software like GNU Privacy Guard to encrypt data using the public key, and then either send that encrypted data in an email, upload the encrypted data to the cloud, or simply keep it on her hard drive. No matter where the encrypted data are stored, they cannot be decrypted without the private key. After encryption, the researcher destroys the unencrypted data. The encrypted data are then inaccessible until the researcher returns to the physical machine that

[23] A number of different software packages can be used to generate key pairs and manage encryption. Two popular, and well-regarded implementations of the PGP framework are GNU Privacy Guard and OpenPGP.

has the private key.

Using PGP (pretty good privacy) encryption to make research data temporarily inaccessible is a good way combat threats of theft and expropriation of sensitive data. PGP works across all major operating systems, and can successfully encrypt many types of files, including .csv tabular data, many types of text files, and .mp3 audio files. PGP is useful for generating temporary inaccessability because it uses one key (the "public key") to encrypt data, and a separate key (the "private key") to decrypt. Data encrypted with a particular public key can *only* be decrypted using the particular private key that matches it. The two keys together are called a "key pair." As far as the maintainers of PGP's open-source implementation know, the encryption standard has not been broken, though some commercial tools that use PGP have had flaws (Brandom, 2014).

When a user has access to a public key, but not the corresponding private key, they can encrypt data and then transport or copy the encrypted data as they please, but they cannot reverse the encryption process. For this reason, PGP is often used by journalists who want sources to be able to share private information via otherwise unsecure channels like email.

A PGP lockbox for social science research serves a slightly different purpose with the same basic tools. Whereas PGP encryption is normally used to transfer data such that it is inaccessible to anyone but the target recipient, social scientists can use the same standards to transport and store data such that it is temporarily inaccessible to *everyone*, by storing the private key somewhere that it cannot be applied to the data while data collection is ongoing (i.e. local storage on a computer at a researcher's home institution). The rest of this section shows step-by-step instructions for setting up and using a PGP lockbox to encrypt sensitive data and make it temporarily inaccessible to all parties, including the researcher.[24]

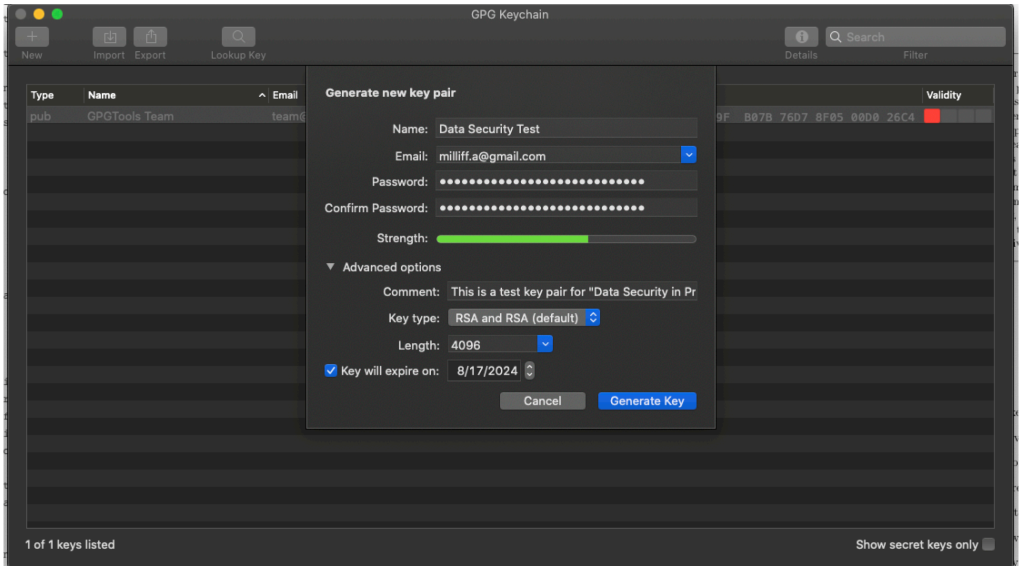
Ingredients:

- Two computers (any OS, any variety)
  - One computer remains at home institution
  - Second computer used during data collection
- Open PGP Software like Gnu Privacy Guard/GPG Tools

Setup:

1. Download and install GPG Tools or other software that implements the OpenPGP framework onto both computers
2. Using the computer that will not be carried during data collection, open the GPG Keychain application, click "new" in the top left corner, and follow instructions to generate a new key pair with the default options.

[24] Instructions and screenshots are specifically for GPG Tools on Mac OSX, but the process is similarly simple using Windows and Linux. Good resources for both exist online.



pub GPGTools Team team@ … 9F B07B 76D7 8F05 00D0 26C4

Generate new key pair

Name: Data Security Test

Email: milliff.a@gmail.com

Password: ••••••••••••••••••••••••••

Confirm Password: ••••••••••••••••••••••••••

Strength:

▼ Advanced options

Comment: This is a test key pair for "Data Security in Pr

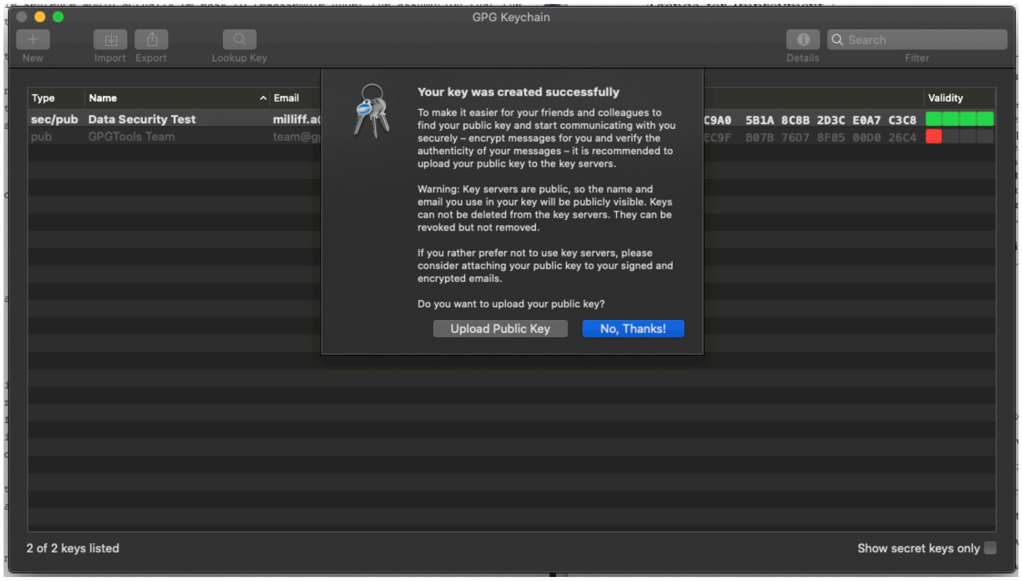
Key type: RSA and RSA (default)

Length: 4096

☑ Key will expire on: 8/17/2024

Figure 3: Creating a new keypair in GPG Keychain for Mac OS X

3. Once the key is successfully created, a prompt will ask you to upload the key pair to a key server, where other users can find your *public* key, and encrypt files with it so that only you (using your new private key) can decrypt. *Unless you are planning to have other users encrypt files with your new key pair, you can select "No, thanks!"*



sec/pub Data Security Test milliff.a … C9A0 5B1A 8C8B 2D3C E0A7 C3C8

pub GPGTools Team team@g … EC9F B07B 76D7 8F05 00D0 26C4

Your key was created successfully

To make it easier for your friends and colleagues to find your public key and start communicating with you securely – encrypt messages for you and verify the authenticity of your messages – it is recommended to upload your public key to the key servers.

Warning: Key servers are public, so the name and email you use in your key will be publicly visible. Keys can not be deleted from the key servers. They can be revoked but not removed.

If you rather prefer not to use key servers, please consider attaching your public key to your signed and encrypted emails.

Do you want to upload your public key?

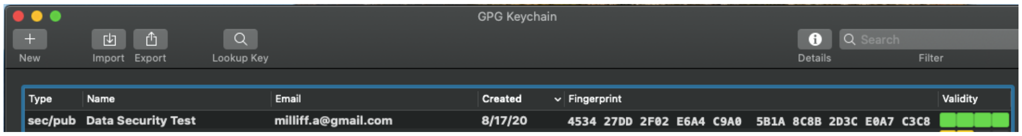
Figure 4: Confirmation of new key pair, GPG Keychain for Mac OS X.

4. Now, it's time to export your public key in order to transfer it. Right click (CTRL + click) on your new key in GPG Keychain, and select "Export" (you can also email the public key to yourself). Make sure the box labeled "include secret key in exported file" is *not checked*, and save the key.

5. Transfer the file with the public key to the data collection computer however you like.

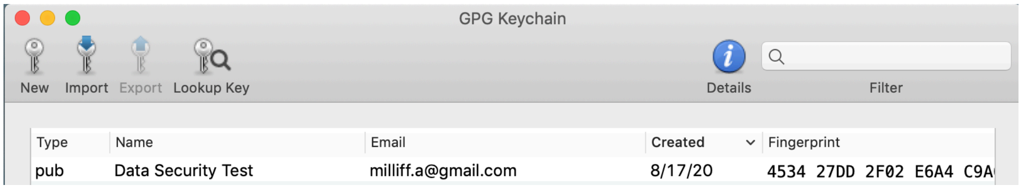
6. Double click on the key file transferred to the data collection computer. Clicking should automatically open GPG Keychain and import the new public key.

7. Verify that your stay-home computer has both public and private keys, and that your data collection computer has only the public key. The leftmost column in GPG Keychain (see figure) shows the "type."



Type	Name	Email	Created	Fingerprint	Validity
sec/pub	Data Security Test	milliff.a@gmail.com	8/17/20	4534 27DD 2F02 E6A4 C9A0 5B1A 8C8B 2D3C E0A7 C3C8	🟩🟩🟩🟩

Figure 5: Verifying key pair type on the stay home computer. The top line shows that both secret/private and public keys for "Data Security Test" fingerprint 4534... are stored on this machine.[26]



Type	Name	Email	Created	Fingerprint
pub	Data Security Test	milliff.a@gmail.com	8/17/20	4534 27DD 2F02 E6A4 C9A

Figure 6: Verifying key pair type on the data collection computer. The top line shows that only the public key for "Data Security Test" fingerprint 4534... is stored on this machine.

8. Nice work! You could theoretically use this key pair for all your PGP needs, but it is probably more cautious to create a separate keypair if you plan to use PGP in emails, etc.

Now that the lockbox is set up, how do you use it? Once again, the objective is that it functions like a timed safe at a convenience store: once you deposit something into it, getting it back is not possible at a moment's notice, no matter how much you may want it. Data encrypted with your new public key will become accessible when you have access to the private key, stored *only* on the stay-home computer.

Using the Lockbox:

0. Before you leave your home institution to collect data, make sure your stay-home computer is password protected. Put it in your desk, lock it if you can. Be sure the private key is only in GPG Keychain, and not in some directory that you can access remotely. Turn off remote access/ssh.

1. On the data collection computer, collect your data. At regular intervals during data collection, encrypt your data and destroy the unencrypted copies:

   (a) In Finder, navigate to the file you wish to encrypt, for example a photo of Chance the Dog.

   (b) Right click (Ctrl + click) on the file, navigate to "services" and then select "OpenPGP: Encrypt File."

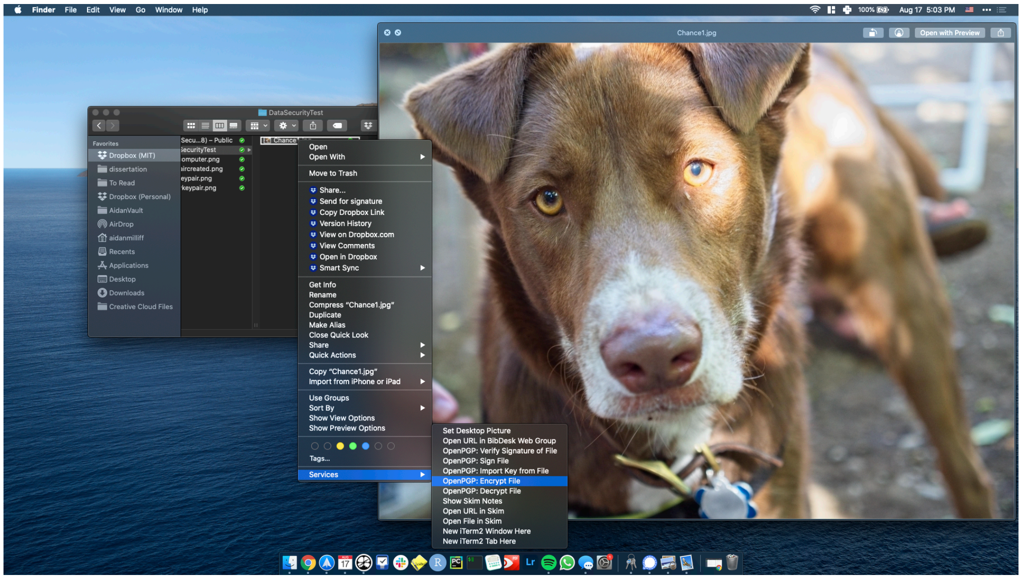
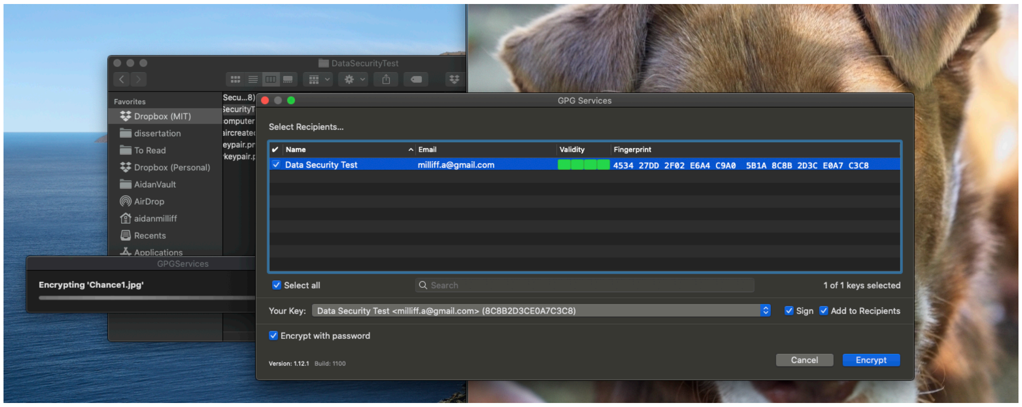


Figure 7: Menus that appear after right-clicking a file in Finder.

(c) A GPG Tools window appears, allowing you to select a key with which to encrypt the file. Use the key created above, and add a file-specific passphrase that is different from the key-specific passphrase created above.



Select Recipients...

✔	Name	Email	Validity	Fingerprint
☑	Data Security Test	milliff.a@gmail.com	🟩🟩🟩🟩	4534 27DD 2F02 E6A4 C9A0 5B1A 8C8B 2D3C E0A7 C3C8

☑ Select all 1 of 1 keys selected

Your Key: Data Security Test <milliff.a@gmail.com> (8C8B2D3CE0A7C3C8) ☑ Sign ☑ Add to Recipients

☑ Encrypt with password

Version: 1.12.1 Build: 1100

Figure 8: Selecting a key for encryption with a passphrase.

(d) Follow the prompts from GPG Tools.

(e) Now you have an encrypted file! Verify the file name now ends in .gpg. *Delete the unencrypted file and empty the trash.*[27]

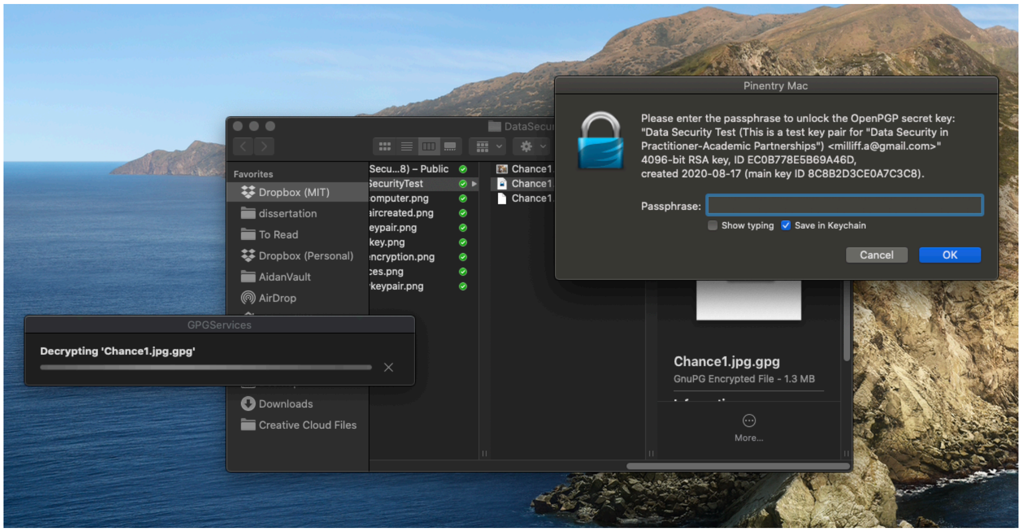
2. Once files are encrypted, treat them as you would normally. Backing them up is not a bad idea, so long as none of them are stored where the private key is.

[27] As many people know, deleted files are often still recoverable. Unfortunately, the solid state hard drives (SSDs) in many new computers make it harder to "overwrite" deleted files than old HDDs did. On a Mac, you can and should still overwrite deleted files when feasible. Open terminal and enter the following prompt to "overwrite" free space on your internal SSD, but the process is slow! `diskutil secureErase freespace 4 /Volumes/Macintosh\ HD`. The numeral 4 is the option for 3-pass secure erasing with the U.S. Department of Energy algorithm. Other options include: US Department of Defense algo. 7-pass erasing (2), Gutmann algorithm 35-pass secure erase (3), overwriting with zeros (0), or a single-pass random overwrite (1).

3. Upon return to your home institution (or when you need to analyze the data), transfer the encrypted files to your stay-home computer and reverse the encryption process.

   (a) Navigate to the encrypted file on your stay-home computer, right click, and select "Services » OpenPGP: Decrypt File"

   (b) A window will prompt you for a password—enter the passphrase you set earlier.



Please enter the passphrase to unlock the OpenPGP secret key:
"Data Security Test (This is a test key pair for "Data Security in Practitioner-Academic Partnerships") <milliff.a@gmail.com>"
4096-bit RSA key, ID EC0B778E5B69A46D,
created 2020-08-17 (main key ID 8C8B2D3CE0A7C3C8).

Passphrase:

□ Show typing ☑ Save in Keychain

Decrypting 'Chance1.jpg.gpg'

Figure 9: Prompts for decryption.

   (c) Success! A decrypted copy of your file should have appeared in the same directory! Open it up and go to work

The PGP Lockbox keeps everyone's hands off your data, including yours. This means the system only works if you can wait to analyze your data until you have returned to your home institution. Keeping the private key on your data collection computer to decrypt and encrypt the data at your convenience offers only as much protection as password-protecting a file.

References in Supplementary Information

Blei, D., A. Ng, and M. Jordan (2003). Latent dirichlet allocation. *Journal of Machine Learning Research 3*(993-1022).

Brandom, R. (2014). New documents reveal which encryption tools the nsa couldn't crack. *TheVerge.com*.

Desai, S. and R. Vanneman (2015). India human development survey ii (IHDS-II).

Grimmer, J. and B. M. Stewart (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis 21*(3), 267–297.

Milliff, A. (2020). Facts shape feelings: An information-based framework for emotional responses to trauma. Working Paper, 2019-06, Massachusetts Institute of Technology, Cambridge, MA.

Roberts, M. E., B. M. Stewart, and D. Tingley (2018). stm: R package for structural topic models. *Journal of Statistical Software*.

Templ, M., B. Meindl, and A. Kowarik (2020). Introduction to statistical disclosure control SDC. Technical report, Zurich University of Applied Sciences, Zurich, Switzerland.