Technical Appendices for Post-Secondary Employment Outcomes Data

A Appendix about Protection System

This appendix details how we protect the released data. First, we discuss the Laplace noise infusion, which is used to protect the count queries for Graduate Earnings and Pipeline Flows. Second, we detail how we use Laplace noise in the histogram approach to calculate percentiles of earnings within a cell. This appendix was the basis for the paper from (1). In that paper, we outline a number of ways the earnings percentiles could be protected, while this appendix only outlines the method used for PSEO.

A.1 Protecting Count Data: Laplacian Noise Infusion

Consider a dataset d, and a neighboring dataset d' which differs by one observation. Furthermore, consider a count query $q_c(.)$ on a dataset, which returns the number of observations with certain attributes, which we will refer to as X. Now consider the cases below:

$$|q_c(d) - q_c(d')| = \begin{cases} 1, \text{if the differing observation has the attributes } X \\ 0, otherwise. \end{cases} \tag{1}$$

The sensitivity $S(q_c)$ is then the smallest number such that for any neighboring datasets,

$$|q_c(d) - q_c(d')| \leq S(q_c)$$

In the case of the count query, $S(q_c) = 1$. Therefore, for any count query $q_c(d)$, $q_c(d) + \zeta$ is ϵ-differentially private, where $\zeta \sim X - Y$, where $X, Y \sim Geometric(p)$ where $p = 1 - \frac{1}{e^\epsilon}$. For

these tables, all the counts are integers, and therefore we will draw noise using the geometric noise mechanism.

A.2 Protecting Percentiles: Histogram Approach

To protect percentiles, we use an approach called the histogram approach.

Consider a dataset d, which has sorted values $e_1, e_2, ..., e_N$ and a function $H(.)$ that assigns each value e_i into a bin, grouping them together. Formally, this function is defined as:

$$H(e_i) = \begin{cases} 1, & \text{if } e_i \in [b_1, b_2) \\ & ... \\ j, & \text{if } e_i \in [b_j, b_{j+1}) \\ & ... \\ M-1, & \text{if } e_i \in [b_{M-1}, b_M) \\ M, & \text{if } e_i \geq b_M \end{cases} \tag{2}$$

Where the borders of the histogram bins, b_i, are public information. The key decision in implementing this protection method is determining how to set the bin definitions, which we describe in the next subsection.

Choosing Bin Definitions

We use the following bin cutoffs for the PSEO protection system. The bottom cutoff is \$10,000 (in 2016 dollars), which is very close to the minimum value in the data by construction, given that we restrict the sample based on earnings. For the next 19 b_is, we choose every 5th percentile of the log normal distribution with mean 11.003 and standard deviation 0.753.[1] Additionally, for b_M, we use the 97.5th percentile value of the distribution, which is about \$260,000. Finally, for any earnings greater than that value, we count it in the final bin, M. Together, we have 21 bins.

For reference, these histogram values are in the appendix.

[1]The log normal distribution is a good approximation of the overall earnings distribution. The mean and standard deviation were calculated using the 5-year ACS Public-Use Microsample. We calculated the mean and standard deviation of wage and salary income for employed individuals with a BA or above.

Queries to Protect

From the definition of the histogram function above, the set of queries we protect are of the form q_j^c, which returns the count of the observations in a given bin j. Additionally, these queries imply the corresponding empirical CDF:

$$F(j) = \frac{\sum_{i=1}^{j} q_i^c}{\sum_{i=1}^{M} q_i^c} \tag{3}$$

The sensitivity of each of these queries is 1, and therefore we can protect each of these queries with privacy loss ϵ by adding geometric noise as described above in Section A.1. Therefore, our protected counts are:

$$\tilde{q}_j^c = q_j^c + \zeta$$

Where ζ is drawn from a geometric noise distribution. The resulting histogram list of counts is ϵ-differentially private (Proposition 1 in (2)), and any function of these counts is also ϵ-differentially private because of the composition properties of differential privacy.[2]

Calculating Protected Percentiles

We use these fuzzed values to create a fuzzed CDF,

$$\tilde{F}(j) = \frac{\sum_{i=1}^{j} \tilde{q}_i^c}{\sum_{i=1}^{M} \tilde{q}_i^c}$$

If we assume that earnings are distributed uniformly within a bin, we can use $\tilde{F}(j)$ to extract protected percentiles.[3]

To calculate a percentile Y, suppose that it is in bin J such that

$$\frac{\sum_{i=1}^{J-1} \tilde{q}_i^c}{\sum_{i=1}^{M} \tilde{q}_i^c} < Y/100 \leq \frac{\sum_{i=1}^{J} \tilde{q}_i^c}{\sum_{i=1}^{M} \tilde{q}_i^c} \tag{4}$$

Then, the Yth percentile is $b_J + (b_{J+1} - b_J) \times \frac{(Y/100 \times \sum^{J} \tilde{q}_i^c) - \sum^{J-1} \tilde{q}_i^c}{\tilde{q}_J^c}$.[4] In the case where a

[2]This result from Hay et al. (2009) allows the list of values $(\tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_M)$ to also be considered releasable.

[3]Note that $\tilde{F}(j)$ will not necessarily be a true CDF, because there may be cases when $\tilde{q}_j < 0$.

[4]In words, if a bin J includes the Yth percentile, and the Yth percentile is W of the way through the interval

percentile is in the largest bin, we define b_{M+1} to be the 99.9th percentile of earnings from the log normal distribution, which is 614597.[5]

We use this technique to calculate the 25th, 50th, and 75th percentile values.

Calculating Protected Counts

We use the fuzzed counts from the histogram approach to calculate the total cell count. In this application, this is just a sum of all the histogram counts: $cellcount_c = \sum_{i=1}^{M} \tilde{q}_i^c$.

We determine whether or not to release values for a cell based on the total protected cell count, because values are noisier for smaller cells. Specifically, we do not release any data for cells with protected counts below 30. When we release tables, we will simply indicate that the cell count is below 30 and publish missing values.

A.3 Protecting Counts in a Sparse Matrix: Employment Flows

The EF data product is a set of count queries, and we will protect these queries using the geometric mechanism for noise.

Consider a flow ($Flow_{(clfg,ks),T}$) from institution c, degree level l, degree field f, and graduation cohort g, into industry k and state s, T years after graduation.[6] To protect the flow count, we use the geometric noise mechanism to add noise to the count such that the protected count is:

$$\widetilde{Flow}_{(clfg,ks),T} = Flow_{(clfg,ks),T} + \eta$$

where $\eta \sim X - Y$, where $X, Y \sim Geometric(p)$ where $p = 1 - \frac{1}{e^{\epsilon}}$.[7] Therefore, $\widetilde{Flow}_{(clfg,ks),T}$ is ϵ-differentially private.

defined by bin J, then the Yth percentile is the lower-bound value of bin J, b_J, plus $W \times width$.

[5] We calculate the percentile for the smallest bin such that Equation 4 is satisfied. This addresses the issue of negative counts in a subsequent bin, since it is possible for Equation 4 to be fulfilled in two distinct bins if the intervening bins have negative counts.

[6] One possible destination in the matrix is non-employed or marginally attached to the labor force, which we denote as industry ZZ and state Z

[7] In our setting, $\epsilon = 1.5$

A.3.1 Post-Processing Protected Counts

Because the flows matrix is sparse, there will be a number of cases where $\widetilde{Flow}_{(clfg,ks),T} < 0$. While this flow is still ϵ-differentially private, it does not make sense logically from the perspective of a user since flows are strictly non-negative.

To that end, we post-process the differentially private counts in order to preserve the logic of the flows data. We post-process the flows in three steps, which we describe in detail. First, any flow that is negative is set to zero. Second, we calculate the difference between the new total count within a (c, g, f, d, T) cell before and after step 1.[8] Finally, we adjust the counts of other non-zero cells so that the new total count within a (c, g, f, d, T) cell is the same. We describe this formally in the next sub-section.

A.3.2 Correcting Cell Counts

Because of the correction of negative flows above, as long as any flows are negative, $\sum_{k,s} \hat{Flow}_{(clfg,ks),T} > \sum_{k,s} \tilde{Flow}_{(clfg,ks),T}$, and therefore we need to correct the other counts in the matrix such that the overall total is unchanged. We do this correction by randomly selecting cells with non-zero counts in them, and subtracting, thereby adjusting to total count.

However, we do not want each cell's probability of being selected to be equal, because we know (from external sources) that some flows are less likely than others. We use three sources of data to weight a cell's probability of being selected for correction.[9]

- State-to-State Job-to-Job (J2J) Flows: We use the job-to-job hires to destination states from the state of the institution. We call this value $J_{h,d}$, for the flows from home state h to destination state d. This addresses the fact that some states are more connected than others; it is less likely for a graduate in Colorado to move to Maine than Arizona. J2J data are drawn from the 2011-2015 period, where all states are available.

- Quarterly Census of Employment and Wages state-by-industry employment data: We use QCEW state-by-industry employment ($Emp_{s,k}$) to address the fact that within a state, some industries are smaller and therefore less likely to employ a graduate. For example, a flow into

[8] If the total is negative, the entire cell is set to zero.

[9] The flow to "non-observed/insufficient" employment is given an arbitrary probability of selection for correction.

Information is more likely in California than a flow into mining. These data are drawn from the full range of PSEO earnings periods.

- Institution by Field (2-digit CIP) to Industry Sector Flows: Some institutions and majors have large flows to specific industry sectors, which may have small overall employment (e.g. Engineering to Mining sector). We call these flows $INDFLOW_{c,f,k} = \sum_g \sum_d \sum_T \sum_s \widetilde{Flow}_{(c,g,f,d,T),k,s}$, and use the protected counts from above.

To weight the cells for selection in the below procedure, we multiply all of these values together and use the inverse, such that the $weight = \frac{1}{J_{j,d} Emp_{s,k} INDFLOW_{c,f,k}}$. In the case of any of these components being zero (or negative) we assign a value of 1. Using these weights, the algorithm for the correction is below.

1. Let $F = \sum_{k,s} \widehat{Flow}_{(clfg,ks),T} - \widetilde{Flow}_{(clfg,ks),T}$ which is number of jobs that need to be subtracted from the overall cell for the counts to be equal.
2. Randomly draw across the cells (using the weights above) with non-zero counts and each time a cell is drawn, subtract one from the flow count.
3. Repeat step 2 F times.
4. Recalculate the new flows $\overline{Flow}_{(clfg,ks),T}$ after these corrections.

By construction, $\sum_{k,s} \overline{Flow}_{(clfg,ks),T} = \sum_{k,s} \widetilde{Flow}_{(clfg,ks),T}$. These are the counts that are released to the public, as they satisfy the logical constraints on the data, while also being a consistent measure of the number of employed individuals.

A.3.3 Suppression of Division Flows Data

Using the protected data at the state level, we will determine if missing data from a particular state causes an appreciable impact on the division-level flows that we report publicly. First, aggregate institution to division flows are calculated for post-graduation year observations for which all states are available. Then, we identify years for which earnings data are only available for a subset of states. We recalculate the division flows from the complete period, using only the subset of states,

and then make an estimate of the share of flows that are unobserved. If the unobserved share is above a metric (TBD), the flows will be suppressed.

B Appendix about Reallocation to 2020 CIP Codes

All the data releases prior to 2020 were released in the 2010 CIP coding edition (formally, CIP-2010).[10] With the release of the most recent data, we needed to update the data outputs into 2020 CIP codes.

There are two main constraints on this process. First, the crosswalks from CIP-2010 to CIP-2020 are at the 6 digit level, while we report outcomes at the 4- and 2-digit level. Second, because we have already released the outcomes and incurred a specific privacy loss, we cannot simply re-release the new data after applying the crosswalks to the microdata.

This appendix presents the methodology we use to recast the earnings histograms and employment flows data into CIP-2020.

B.1 Step 1: Calculate the posterior for 2010-2020 transitions

First, we identify all possible transitions at the 2- and 4-digit CIP level, based on transitions at the 6-digit level, using the crosswalk provided by National Center for Education Statistics. We then take the historical IPEDS data, crosswalked to CIP-2010, and cast it into CIP-2020 at the six-digit level.

If IPEDS does not report any individuals making a transition, assign an arbitrarily small weight. Aggregate the counts for 2 and 4 digit before and after codes, at the following levels:

- Degree level x OPEID x graduation cohort
- Degree level x OPEID (across cohorts)
- Degree level (across OPEIDs)
- All degree levels

[10] This includes data from Colorado, University of Texas System, University of Michigan-Ann Arbor, and University of Wisconsin-Madison.

For all 2010 2/4 digit codes at all aggregation levels, we calculate the share of the counts in each 2020 2/4 digit code, which we use as probability weights in the next step.

B.2 Step 2: Allocate counts to 2020 codes

Our data tables which are protected by differential privacy (described in the previous appendix section) are counts. These steps describe how we reallocate those counts from CIP-2010 to CIP-2020. The below steps are in order of priority.

1. If a 2010 code (at 2/4 digit level) always has same destination code, assign code directly.
2. Select codes where IPEDS counts are inconsistent or unavailable have hardcoded transitions (some codes are not reported to IPEDS, but in our data)
3. Codes that have multiple possible destinations in IPEDS are assigned probabilistically for each histogram bin count, using the tables created in Step 1.
4. Total counts will match at the OPEID x degree level x grad cohort x year postgrad level, between the 2010 CIP vintage and 2020 CIP vintage.

The above algorithm is used to recast the histogram bin counts and the flows data *before* the corrections are applied (when flows/bin counts may be negative). After the protected flow counts are reallocated, they are run through the same correction algorithm to correct for negative flows.

References

[1] Andrew Foote, Ashwin Machanavajjhala, and Kevin McKinney. Releasing Earnings Distributions using Differential Privacy: Disclosure Avoidance System for Post-Secondary Employment Outcomes (PSEO). *Journal of Privacy and Confidentiality*, 9(2), October 2019.

[2] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially-private queries through consistency. *CoRR*, abs/0904.0942, 2009.

C Tables Appendix

Table 1: Histogram bin values

Bin	Lower Bound	Upper Bound
1	10000	17403
2	17403	22876
3	22876	27512
4	27512	31857
5	31857	36128
6	36128	40449
7	40449	44914
8	44914	49605
9	49605	54609
10	54609	60027
11	60027	65982
12	65982	72639
13	72639	80226
14	80226	89080
15	89080	99735
16	99735	113106
17	113106	130970
18	130970	157509
19	157509	207050
20	207050	262475
21	262475	614597

Notes: Except for the lowest value, these are all percentiles from a log normal distribution with mean 11.003 and standard deviation 0.753. Any observation will be classified into the final bin (21) if it has a value above 262475. For purposes of calculating the percentiles, we use the upper bound value for bin 21 of 614597, which is the 99.9th percentile of the log normal distribution.