Tim Duffy
CS 558 – Network Security
Differentially Private Data Analysis Lab
3/12/2012

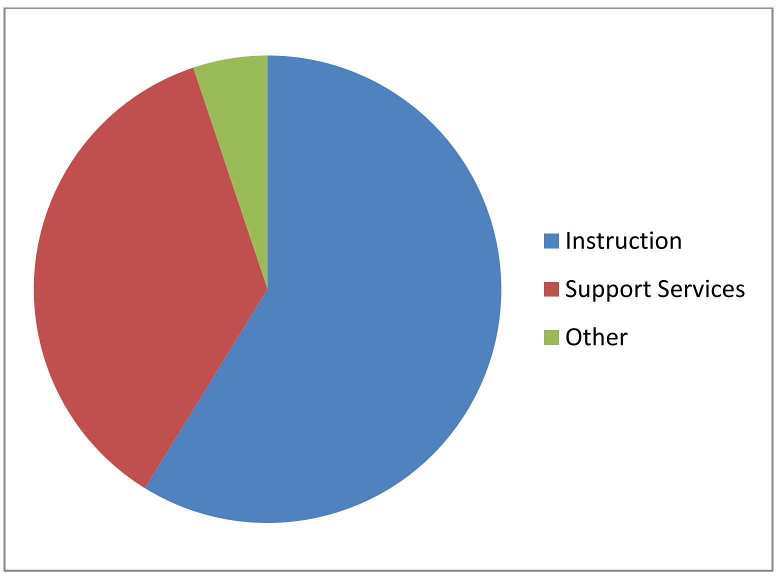
Part 0: Meta Information

For this lab, I used the dataset " Public Elementary-Secondary Education Finance Data". It can be found here: http://www.census.gov/govs/school/

This data set has 14634 records, which contain various attributes of revenue and spending for all school districts in the US. Each record has 142 attributes, but for my analysis, I only consider about 50 of these. For the data analysis, I used PINQ.

Part 1: Differentially Private Report

Figure 1: Breakdown of Revenue Sources



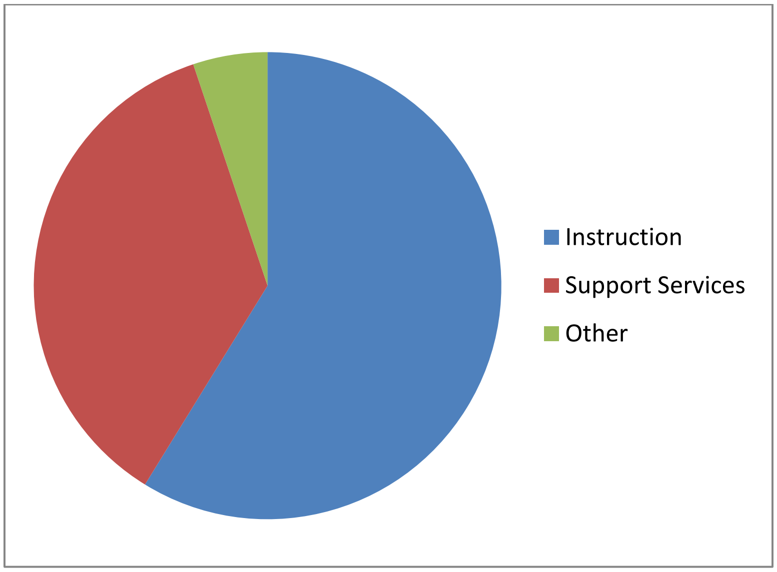
Percentages (based on DP analysis)

Federal Sources = 10.29%
State Sources = 47.76%
Local Sources = 50.84%

This figure shows the distribution of revenue Sources (Federal, State, or Local). Based on the differentially private analysis, we see that the majority of revenue comes from State and Local Sources, while only 10% comes from Federal sources.

Looking deeper into these revenue sources, the following chart shows the breakdown of revenue from Local Sources.

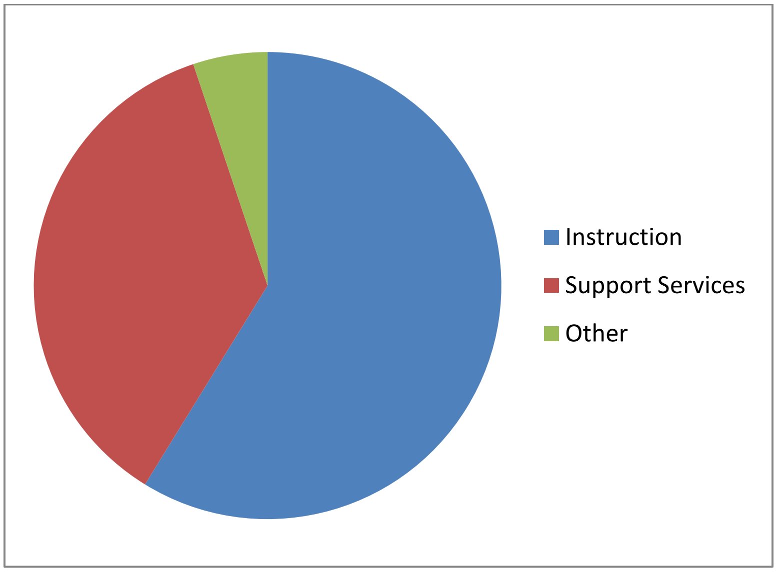
Figure 2: Breakdown of Local Revenue Sources



Percentages (based on DP analysis) Taxes = 62.39%
Charges = 3.94%
Other = 13.31%

Here, we see that the majority of local revenue comes from taxes. It is interesting to see that 13% of local income comes from other sources, which consist only of private contributions and "other miscellaneous local income". This seems to suggest that a sizable portion of Local income is in the form of private contributions.

Figure 3: Breakdown of Expenses by Type



Percentages (based on DP analysis) Instruction = 55.7%
Support Services = 34.17%
Other = 4.88%

This figure shows the breakdown of expenses by type. We see that an average of 55.7% of all expenses is spent on Instruction. This includes the salary and bonuses of teachers, and the cost of teaching supplies. Nearly 35% of expenses are used on support services, which include general administration, transportation, and maintenance. I find it very interesting that so much money is spent just on running and maintaining the schools.

The first aspect of this data that I chose to analyze is the debt that the districts have. This figure is calculated by subtracting the total revenue of a district from its total expenses. The amount left over is how much the school district overspent, and how much debt it is in at the end of the fiscal year. In total, there are approximately **555(figure 4)** schools in debt in this year. Of the schools in debt, there was an average of **$2721679.64(figure 5)** in overspending. Although the majority of schools are not in debt, these 555 schools have accumulated a total of approximately **$6,010,050,960,000(figure 6) in debt!**

Next, I looked at the amount spent on each student, based on the number of students enrolled in the school. I found that schools with more students spent less per student than schools with fewer students. Of the **279 (figure 7)** school districts with over 25 thousand students, there was an average of **$12750.05 (figure 8)** spent per pupil. However, of the **6338 (figure 9)** districts with fewer than 3 thousand pupils, an average of **$17993.27 (figure 10)** was spent per pupil. Does this mean a lesser learning experience at schools with higher enrollment? Or perhaps this number differs so much because bigger schools cost more to operate, thus needing more of the school's expenses to be spent on maintenance? This is a question that I would like to know, but one that this data can not answer.

Finally, I looked at the average pay a teacher gets per student. This was calculated by dividing the amount spent on teacher salaries by the number of pupils enrolled. This does not give us any information about average teacher salaries, but allows us to infer a ratio of teacher salaries to students. Through this analysis, I found that although there is no great difference, there is a clear trend. Teachers in school districts with more students get paid a small amount less per student than teachers in districts with lower enrollment. In districts with more than 25 thousand students, an average of **$0.952 (figure 11)** is paid toward teacher salaries per student. In districts with less than 25 thousand, but more than 3 thousand, an average of **$0.9905 (figure 12)** is paid. In districts with less than 3 thousand students, an average of **$0.999 (figure 13)** is paid per pupil. Because there is no significant difference, it seems to be that a teacher in a school with large enrollment will be paid better than a teacher in a district with fewer students.

Part 2: Privacy and Utility Analysis

1) Distribution of Total Revenue

a. Average Total Revenue

(a) Pseudo code:

```
// For each record x, return x.totalRevenue/ 1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x.totalRevenue / 1,000,000;
// For each record in database, apply f () and take the Noisy average
Average Total Revenue = NoisyAverage (database, f, 0.25);
```

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $41049.98
SNR = 41049.98 / (sqrt(2) / 0.25) = 7256.68

(e) Root-Mean-Square Error

Computed Value = $37117.455
RMSE = sqrt((1 – (41049.98/37117.455))^2) = 0.11

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.25 = 11.313**

b. Average Revenue From Federal Sources

(a) Pseudo code:

```
// For each record x, return x.federalRevenue/ 1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x. federalRevenue / 1,000,000;
// For each record in database, apply f () and take the Noisy Average
Average Federal Revenue = NoisyAverage (database, f, 0.25) * 1,000,000; // Rescale our data
```

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $3820.14
SNR = 3820.14 / (sqrt(2) / 0.25) = 675.311

(e) Root-Mean-Square Error

Computed Value = $5989.29
RMSE = sqrt((1 – (5989.29/3820.14))^2) = 0.568

(f)Total Noise Added

Total Noise = Std. Dev = 2* sqrt(2) / 0.25 = 11.313

c. Average Revenue From State Sources

(a) Pseudo code:
// For each record x, return x.stateRevenue/ 1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x. stateRevenue / 1,000,000;
// For each record in database, apply f () and take the Noisy Average
Average State Revenue = NoisyAverage (database, f, 0.25) * 1,000,000; // Rescale our data

(b) *This mechanism takes epsilon = 0.25*

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $17728.95
SNR = 17728.95/ (sqrt(2) / 0.25) = 3134.06

(e) Root-Mean-Square Error

Computed Value = $18871.99
RMSE = sqrt((1 – (18871.99/17728.95))^2) = 0.064

(f)Total Noise Added

Total Noise = Std. Dev = 2* sqrt(2) / 0.24 = 11.131

d. Average Revenue From Local Sources

(a) Pseudo code:
// For each record x, return x.localRevenue/ 1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x. localRevenue / 1,000,000;
// For each record in database, apply f () and take the Noisy Average
Average Local Revenue = NoisyAverage (database, f, 0.25) * 1,000,000; // Rescale our data

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $18357.845
SNR = 18357.845/ (sqrt(2) / 0.25) = 3245.24

(e) Root-Mean-Square Error

Computed Value = $18871.99

RMSE = sqrt((1 – (18871.99/18357.845))^2) = 9.92

(f)Total Noise Added

Total Noise = Std. Dev = 2* **sqrt(2) / 0.25 = 11.131**

2) Distribution of Local Revenue

a. Average Local Revenue

(*(a) Pseudo code:*

// For each record x, return x.localRevenue/ 1,000,000 so it fits in the [-1, +1] interval

Function<Record => double> f = x. localRevenue / 1,000,000;

// For each record in database, apply f () and take the Noisy Average

Average Local Revenue = NoisyAverage (database, f, 0.25) * 1,000,000; // Rescale our data

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.

Noise Free Value = $18357.845

SNR = 18357.845/ (sqrt(2) / 0.25) = 3245.24

(e) Root-Mean-Square Error

Computed Value = $18871.99

RMSE = sqrt((1 – (18871.99/18357.845))^2) = 9.92

(f)Total Noise Added

Total Noise = Std. Dev = 2* **sqrt(2) / 0.25 = 11.131**

b. Average Revenue From Taxes

(a) Pseudo code:

// For each Record x, return x.RevenueFromTax/1,000,000 so it fits in the [-1, +1] interval

Function<Record => double> f = x. RevenueFromTax / 1,000,000;

// For each record in database, apply f () and take the Noisy Average

/// Rescale data

Average Local Revenue From Taxes = NoisyAverage (database, f, 0.25) * 1,000,000;

(b) This mechanism takes epsilon = 0.25

(a) Pseudo code:

// For each record x, return x.localRevenue/ 1,000,000 so it fits in the [-1, +1] interval

Function<Record => double> f = x. localRevenue / 1,000,000;
// For each record in database, apply f () and take the Noisy Average
Average Local Revenue = NoisyAverage (database, f, 0.25) * 1,000,000; // Rescale our data

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $15152.608
SNR = 15152.608 / (sqrt(2) / 0.25) = 2678.628

(e) Root-Mean-Square Error

Computed Value = $11454.341
RMSE = sqrt((1 – (11454.341/15152.608))^2) = 0.244

(f)Total Noise Added

Total Noise = Std. Dev = 2* sqrt(2) / 0.25 = 11.131

c. Average Revenue From Charges (Fines, Tuition, other)

(a) Pseudo code:
// For each Record x, return x.revenueFromCharges/ 1000000 so it fits in the [-1, +1] interval
Function<Record => double> f = x.revenueFromCharges / 1,000,000;
// For each record in database, apply f () and take the Noisy Average, then rescale our data
Average Revenue From Charges = NoisyAverage (database, f, 0.25) * 1,000,000;

(b) *This mechanism takes epsilon = 0.25*

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $633.68
SNR = 633.68/ (sqrt(2) / 0.25) = 112.02

(e) Root-Mean-Square Error

Computed Value = $723.52
RMSE = sqrt((1 – (723.52/633.68))^2) = 0.3228

(f)Total Noise Added

Total Noise = Std. Dev = 2* sqrt(2) / 0.25 = 11.131

d. Average Revenue From Other Sources (Private Contributions)

(a) Pseudo code:
// For each record x, return x.RevenueFromOther/ 1,000,000 so it fits in the [-1,+1] interval
Function<Record => double> f = x.RevenueFromOther / 1,000,000;
// For each record in database, apply f() and take the Noisy Average , then rescale our data
Average Revenue From Other = NoisyAverage(database, f, 0.25) * 1,000,000;

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $2571.56
SNR = 2571.56/ (sqrt(2) / 0.25) = 454.591

(e) Root-Mean-Square Error

Computed Value = $2443.34
RMSE = sqrt((1 – (2443.34/2571.56))^2) = 0.0498

(f)Total Noise Added

Total Noise = Std. Dev = 2* **sqrt(2) / 0.25 = 11.131**

3) Distribution of Expenses

a. Average Total Expenses

(a) Pseudo code:
// For each Record x, return x.totalExpenses/ 1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x.totalExpenses / 1,000,000;
// For each record in database, apply f () and take the Noisy Average
Average Total Expenses = NoisyAverage (database, f, 0.25) * 1,000,000; // Rescale our data

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $42149.27
SNR = 42149.27/ (sqrt(2) / 0.25) = 7451

(e) Root-Mean-Square Error

Computed Value = $38265.48
RMSE = sqrt((1 – (38265.48/42149.27))^2) = 0.0921

(f)Total Noise Added

Total Noise = Std. Dev = 2* sqrt(2) / 0.25 = 11.131

b. Average Expenses on Instruction (Teacher Salaries + Instructional Equipment)

(a) Pseudo code:
// For each Record x, return x.InstructionExpenses/1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x.InstructionExpenses / 1,000,000;
// For each record in database, apply f () and take the Noisy Average, then rescale our data
Average Expenses on Instruction = NoisyAverage (database, f, 0.25) * 1,000,000;

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $19480.94
SNR = **19480.94/ (sqrt(2) / 0.25) = 3443.77**

(e) Root-Mean-Square Error

Computed Value = $21314.228
RMSE = sqrt((1 – (21314.228/19480.94))^2) = 0.094

(f)Total Noise Added

Total Noise = Std. Dev = 2* sqrt(2) / 0.25 = 11.131

c. Total Expenses on Support Services (Administration, Transportation, Maintenance)

(a) Pseudo code:
// For each Record x, return x.supportExpenses/ 1000000 so it fits in the [-1, +1] interval
Function<Record => double> f = x.supportExpenses / 1,000,000;
// For each record in database, apply f () and take the Noisy Average, then rescale our data
Average Expenses on Support Services = NoisyAverage (database, f, 0.25) * 1,000,000;

(b)*This mechanism takes epsilon = 0.25*

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $12211.69
SNR = 12211.69/ (sqrt(2) / 0.25) = 2158.74

(e) Root-Mean-Square Error

Computed Value = $13075.96

RMSE = sqrt((1 – (13075.96/12211.69))^2) = 0.071

(f)Total Noise Added

***Total Noise = Std. Dev = 2** sqrt(2) / 0.25 = 11.131*

d. Total Expenses in Other (Food, Construction)

(a) Pseudo code:

// For each record x, return x.otherExpenses/ 1,000,000 so it fits in the [-1, +1] interval
Function<Record => double> f = x.otherExpenses / 1,000,000;
// For each record in database, apply f () and take the Noisy Average then rescale our data
Average Expenses on Other = NoisyAverage (database, f, 0.25) * 1,000,000;

(b) This mechanism takes epsilon = 0.25

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage = 2*sqrt(2) / epsilon.
Noise Free Value = $1402.89
SNR = 1402.89/ (sqrt(2) / 0.25) = 247.998

(e) Root-Mean-Square Error

Computed Value = $1867.68

RMSE = sqrt((1 – (1867.68/1402.89))^2) = 0.331

(f)Total Noise Added

***Total Noise = Std. Dev = 2** sqrt(2) / 0.25 = 11.131*

Figures:

1) Number of Schools in Debt

(a) Pseudo code:

// Record x.debt holds the amount of debt (Total Expenses – Total Revenue).
// If a district is not in debt, this value is 0.

// First, select from our database, records with debt > 0;
var query = db.Where (Record x.debt > 0);

// Count the number of records in 'query' and add noise = Laplace (1/e) where e=0.25
Number of Schools in Debt = query.NoisyCount (0.1)

(b) This mechanism takes epsilon = 0.1

(c) Signal to Noise Ratio

Standard Deviation for NoisyCount = sqrt(2) / epsilon.

Noise Free Value = 5553

SNR = 5553 / sqrt(2) / 0.1 = 392.66

(e) Root-Mean-Square Error

Computed Value = 554.705

RMSE = sqrt((1 – (554.705/5553))^2) = 0.9

(f)Total Noise Added

***Total Noise = Std. Dev =* sqrt(2) / 0.1 = 14.14**

2) Average Debt

(a) Pseudo code:

```
// First, select from our database, records with debt > 0;
var query1 = db.Where (Record x.debt > 0);

// Fit the debt of each record into the [-1, +1] interval
Function<Record => double> f = x.otherExpenses / 10,000;
Average Debt = query1.NoisyAverage (0.2) * 10,000;// Rescale our data
// This number is in thousands of dollars
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.

Noise Free Value = $6010050.96

SNR = 6010050.96 / (2*sqrt(2) / 0.2) = 424974.78

(e) Root-Mean-Square Error

Computed Value = $2721679.64

RMSE = sqrt((1 – (2721679.64/6010050.96))^2) = 0.55

(f)Total Noise Added

***Total Noise = Std. Dev = 2**sqrt(2) / 0.2 = 14.14**

3) Total Debt of the US School System

(a) Pseudo code:

```
// First, select from our database, records with debt > 0;
var query1 = db.Where (Record x.debt > 0);

// Fit the debt of each record into the [-1, +1] interval
Function<Record => double> f = x.otherExpenses / 10,000;

// Add all values for debt, and add noise = Laplace (1/e) where e=0.2
Average Debt = query1.NoisySum (0.2) * 10,000;      // Rescale our data
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisySum= sqrt(2) / epsilon.
Noise Free Value = $6010050.96
SNR = 6010050.96 / (2*sqrt(2) / 0.2) = 424974.78

(e) Root-Mean-Square Error

Computed Value = $2721679.64
RMSE = sqrt((1 – (2721679.64/6010050.96))^2) = 0.547

(f)Total Noise Added

***Total Noise = Std. Dev =* sqrt(2) / 0.2 = 7.07**

4) Average Amount Spent Per Student

(a) Pseudo code:

```
// Fit the amount spent per student (Total Expenses / Enrollment) of each record into [-1, +1]
Function<Record => double> f = x.spentPerStudent / 1,000;

// Apply f to each record, and take the Noisy Average of the result
Average Amount Spent Per Student = NoisyAverage (db, f, 0.2) * 1,000;        // Rescale our data
// The result is in thousands of dollars
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.
Noise Free Value = $12927.36
SNR = 12927.36 / (2*sqrt(2) / 0.2) = 914.10

(e) Root-Mean-Square Error

Computed Value =$ 12806.91

RMSE = sqrt((1 – (2721679.64/12927.36**))^2) = 0.55**

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14***

5) Number of districts with 25K+ students

(a) Pseudo code:

```
// First, select from our database, records with enrollment > 25,000
var query = db.Where (Record x.enrollment>25,000);

// Count the number of records in 'query' and add noise = Laplace (1/e) where e=0.2
Number of Schools in Debt = query.NoisyCount (0.2)
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisyCount = sqrt(2) / epsilon.

Noise Free Value = 281

SNR = 281 / (sqrt(2) / 0.2) = 39.74

(e) Root-Mean-Square Error

Computed Value = 279.11

RMSE = sqrt((1 – (279.11/281))^2) = 0.0067

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14***

6) Average Amount Spent on a Student in Districts with 25K+ students

(a) Pseudo code:

```
// First, select from our database, records with enrollment > 25,000
var query = db.Where (Record x.enrollment>25,000);

// Fit the amount spent per student (Total Expenses / Enrollment) of each record into [-1, +1]
Function<Record => double> f = x.spentPerStudent / 1,000;

// Apply f to each record in 'query', and take the Noisy Average of the result
Average Amount Spent Per Student = NoisyAverage (query, f, 0.2) * 1,000;    // Rescale our data
// The result is in thousands of dollars
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.
Noise Free Value = $11567.39
SNR = 11567.39 / (2*sqrt(2) / 0.2) = 817.94

(e) Root-Mean-Square Error

Computed Value = $12750.05
RMSE = sqrt((1 – (12750.05/11567.39))^2) = 0.102

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14**

7) Number of districts with Less Than 3K students

(a) Pseudo code:

```
// First, select from our database, records with enrollment < 3,000
var query = db.Where (Record x.enrollment < 3,000);

// Count the number of records in 'query' and add noise = Laplace (1/e) where e=0.2
Number of Schools in Debt = query.NoisyCount (0.2)
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisyCount = sqrt(2) / epsilon.
Noise Free Value = 6337
SNR = 6337 / (sqrt(2) / 0.2) = 896.19

(e) Root-Mean-Square Error

Computed Value = 6338.69
RMSE = sqrt((1 – (6338.69/6337))^2) = 0.00027

(f)Total Noise Added

***Total Noise = Std. Dev =* sqrt(2) / 0.2 = 7.07**

8) Average Amount Spent on a Student in Districts with Less Than 3K students

(a) Pseudo code:

```
// First, select from our database, records with enrollment < 3,000
var query = db.Where (Record x.enrollment < 3,000);

// Fit the amount spent per student (Total Expenses / Enrollment) of each record into [-1, +1]
Function<Record => double> f = x.spentPerStudent / 1,000;

// Apply f to each record in 'query', and take the Noisy Average of the result
Average Amount Spent Per Student = NoisyAverage (query, f, 0.2) * 1,000;    // Rescale our data
// The result is in thousands of dollars
```

(b) This mechanism takes epsilon = 0.2

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.
Noise Free Value = $15420.26
SNR = 15420.26 / (2*sqrt(2) / 0.2) = 1090.38

(e) Root-Mean-Square Error

Computed Value =$ 17993.27
RMSE = sqrt((1 – (17993.27/15420.26))^2) = 0.167

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14

9) Average Teacher Pay Per Student in Districts with 25K+ students

(a) Pseudo code:

```
// First, select from our database, records with enrollment > 25,000
var query = db.Where (Record x.enrollment > 25,000);

// Fit the amount paid per student (Teacher Salary / Enrollment) of each record into [-1, +1]
Function<Record => double> f = x.payPerStudent/ 1,000;

// Apply f to each record in 'query', and take the Noisy Average of the result
Average Amount Spent Per Student = NoisyAverage (query, f, 0.17) * 1,000;  // Rescale our data
```

(b) This mechanism takes epsilon = 0.17

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.
Noise Free Value = $4.01

SNR = 4.01 / (2*sqrt(2) / 0.17) = 0.24

(e) Root-Mean-Square Error

Computed Value =$ 0.952
RMSE = sqrt((1 – (0.952/4.01))^2) = 0.76

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14**

10) Average Teacher Pay Per Student in Districts with Between 3K and 25K students

(a) Pseudo code:

```
// First, select from our database, records with enrollment < 25,000 and > 3,000
var query = db.Where (Record x.enrollment > 25,000 && Record x.enrollment < 3,000);

// Fit the amount paid per student (Teacher Salary / Enrollment) of each record into [-1, +1]
Function<Record => double> f = x.payPerStudent/ 1,000;

// Apply f to each record in 'query', and take the Noisy Average of the result
Average Amount Spent Per Student = NoisyAverage (query, f, 0.16) * 1,000;  // Rescale our data
```

(b) This mechanism takes epsilon = 0.16

(c) Signal to Noise Ratio
Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.
Noise Free Value = $4.35
SNR = 4.35 / (2*sqrt(2) / 0.16) = 0.246

(e) Root-Mean-Square Error

Computed Value =$ 0.9905
RMSE = sqrt((1 – (0.9905/4.35))^2) = 0.772

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14**

11) Average Teacher Pay Per Student in Districts with Less than 3K students

(a) Pseudo code:

```
// First, select from our database, records with enrollment < 3,000
```

```
var query = db.Where (Record x.enrollment  < 3,000);

// Fit the amount paid per student (Teacher Salary / Enrollment) of each record into [-1, +1]
Function<Record => double> f = x.payPerStudent/ 1,000;

// Apply f to each record in 'query', and take the Noisy Average of the result
Average Amount Spent Per Student = NoisyAverage (query, f, 0.17) * 1,000;  // Rescale our data
```

(b) This mechanism takes epsilon = 0.17

(c) Signal to Noise Ratio

Standard Deviation for NoisyAverage= 2*sqrt(2) / epsilon.

Noise Free Value = $4.62

SNR = 4.2 / (2*sqrt(2) / 0.17) = 0.252

(e) Root-Mean-Square Error

Computed Value =$ 0.999

RMSE = sqrt((1 – (0.999/4.62))^2) = 0.784

(f)Total Noise Added

Total Noise = Std. Dev = 2*sqrt(2) / 0.2 = 14.14

Part 3: Summary

a. For this analysis, a total privacy budget of exactly epsilon = 5 was used.

b. **Total Root Square Mean Error** = sqrt((0.11)^2 + (0.568)^2 + (0.064)^2 + (9.9)^2 + (9.92)^2 + (2.44)^2 + (0.3228)^2 + (0.0498)^2 + (0.0921)^2 + (0.094)^2 + (0.071)^2 + (0.331)^2 + (0.9)^2 + (0.55)^2 + (0.547)^2 + (0.55)^2 + (0.0067)^2 + (0.102)^2 + (0.00027)^2 + (0.167)^2 + (0.76)^2 + (0.772)^2 + (0.784)^2) = **14.369675**

c. Total Noise Added = (11.313 *12) + (14.14 * 9) + (7.07 * 2) **= 277.156**