Have you ever cheated on a test?
Have you ever cheated on a test?

Don’t answer, yet. This talk is about how I can encourage (or incentivise) you to tell the truth.
Randomised Response

Find a secret spot to flip your penny
(you don't want anyone to see the result of your penny flip)
Randomised Response

Find a secret spot to flip your penny
(you don't want anyone to see the result of your penny flip)

Flip the penny once

- HEADS: Tell the truth
- TAILS: Flip the penny again

Flip the penny again

- TAILS: Answer YES
- HEADS: Answer NO
Features of Randomised Response

Promotes Truthfulness and the Data is still Useful!
Randomised Response

Features of Randomised Response

Promotes Truthfulness and the Data is still Useful!

Truthfulness: The algorithm gives you plausible deniability. That is, you might as well tell the truth because you can always just deny that the answer you gave was the truth.
Features of Randomised Response

Promotes Truthfulness and the Data is still Useful!

**Truthfulness:** The algorithm gives you plausible deniability. That is, you might as well tell the truth because you can always just deny that the answer you gave was the truth.

We removed the negative consequences of telling the truth by promising you privacy.
Features of Randomised Response

Promotes Truthfulness and the Data is still Useful!

Truthfulness: The algorithm gives you plausible deniability. That is, you might as well tell the truth because you can always just deny that the answer you gave was the truth.

We removed the negative consequences of telling the truth by promising you privacy.

Caveat: I didn’t actually incentivise truth-telling. So you might as well tell the truth, but you also might as well lie.
Features of Randomised Response

Promotes Truthfulness and the Data is still Useful!

**Truthfulness:** The algorithm gives you plausible deniability. That is, you might as well tell the truth because you can always just deny that the answer you gave was the truth.

We removed the negative consequences of telling the truth by promising you privacy.

**Caveat:** I didn’t actually incentivise truth-telling. So you might as well tell the truth, but you also might as well lie.

We’ll talk about removing this caveat at the end.
Utility: The error from randomised response is of the same order as the sample error!
Utility: The error from randomised response is of the same order as the sample error!

Utility aim: We want to estimate the probability, $\alpha$, that a person in the general population has cheated.
Utility: The error from randomised response is of the same order as the sample error!

Utility aim: We want to estimate the probability, $\alpha$, that a person in the general population has cheated.

Sample Error: Assume $n$ people are sampled and they all tell the truth. The best estimate I can get to $\alpha$ is the fraction of my sample that has cheated, $\hat{\alpha}$. I have the guarantee that

$$|\alpha - \hat{\alpha}| \leq \frac{1}{\sqrt{n}}$$

with probability $e^{-2} \approx 0.14$
Randomised Response Error: The probability that a randomly sampled person answers YES is

\[
\frac{1}{2} \alpha + \frac{1}{2} \frac{1}{2}
\]

\(P(\text{first coin HEADS})\)

\(P(X=1)\)

\(P(\text{first coin TAILS})\)

\(P(\text{second coin TAILS})\)

Thus, the estimate, \(\hat{\alpha}\), I get from Randomised Response satisfies

\[|\hat{\alpha} - \alpha| \leq \frac{2}{\sqrt{n}}\]

with probability \(e^{-2}\)
Randomised Response Error: The probability that a randomly sampled person answers YES is

\[ \frac{1}{2} \alpha + \frac{1}{2} \frac{1}{2} \]

Thus, the estimate, \( \hat{\alpha} \), I get from Randomised Response satisfies

\[ |\hat{\alpha} - \alpha| \leq \frac{2}{\sqrt{n}} \quad \text{with probability } e^{-2} \]

I only need 4 times as many people to get the same error in randomized response than if I assumed everyone was telling the truth.
Randomised Response Error: The probability that a randomly sampled person answers YES is

\[
\frac{1}{2} \alpha + \frac{1}{2} \frac{1}{2}
\]

\[\mathbb{P}(\text{first coin HEADS}) \quad \mathbb{P}(\text{first coin TAILS})\]

\[\mathbb{P}(X=1) \quad \mathbb{P}(\text{second coin TAILS})\]

Thus, the estimate, \(\hat{\alpha}\), I get from Randomised Response satisfies

\[|\hat{\alpha} - \alpha| \leq \frac{2}{\sqrt{n}}\quad \text{with probability } e^{-2}\]

I only need 4 times as many people to get the same error in randomized response than if I assumed everyone was telling the truth AND I removed the additional error obtained from people lying.
Let’s consider what privacy guarantee I gave you in randomised response:

Whatever response you gave me, I can’t with any confidence work out whether that was your true answer or not.
Let’s consider what privacy guarantee I gave you in randomised response:

Whatever response you gave me, I can’t with any confidence work out whether that was your true answer or not.

Note: I didn’t promise you that I wouldn’t learn anything about you. If I learn 99% of people have cheated at least once, then I have learnt that you have probably cheated at least once.
Let’s consider what privacy guarantee I gave you in randomised response:

Whatever response you gave me, I can’t with any confidence work out whether that was your true answer or not.

Note: I didn’t promise you that I wouldn’t learn anything about you. If I learn 99% of people have cheated at least once, then I have learnt that you have probably cheated at least once.
I DO promise that I’ll learn approximately the same amount about you whether or not you agree to answer the question truthfully.
**Talk math to me.**

**Differentially private algorithms** are a class of algorithms that satisfy the same privacy guarantee as randomised response.
Talk math to me.

Differentially private algorithms are a class of algorithms that satisfy the same privacy guarantee as randomised response.

Let $D$ and $D'$ be two databases that differ on the data of a single person. As algorithm $\mathcal{A}$ is $\epsilon$-differentially private if for all events $E$,

$$\Pr(\mathcal{A}(D) \in E) \leq e^\epsilon \Pr(\mathcal{A}(D') \in E) \approx (1 + \epsilon)\Pr(\mathcal{A}(D') \in E)$$
Differentially private algorithms are a class of algorithms that satisfy the same privacy guarantee as randomised response.

Let $D$ and $D'$ be two databases that differ on the data of a single person. As algorithm $\mathcal{A}$ is $\epsilon$-differentially private if for all events $E$,

$$
P(\mathcal{A}(D) \in E) \leq e^{\epsilon} P(\mathcal{A}(D') \in E) \approx (1 + \epsilon) P(\mathcal{A}(D') \in E)
$$

$D = \text{everyone else's data + your true data}$,
$D' = \text{everyone else's data + your fake data}$.
Differentially private algorithms are a class of algorithms that satisfy the same privacy guarantee as randomised response. Let $D$ and $D'$ be two databases that differ on the data of a single person. As algorithm $A$ is $\epsilon$-differentially private if for all events $E$,

$$\mathbb{P}(A(D) \in E) \leq e^\epsilon \mathbb{P}(A(D') \in E) \approx (1 + \epsilon) \mathbb{P}(A(D') \in E)$$

$\epsilon$ is a parameter. The smaller it is, the more “private” the algorithm $A$ is.
Talk math to me.

**Differentially private algorithms** are a class of algorithms that satisfy the same privacy guarantee as randomised response.

Let $D$ and $D'$ be two databases that differ on the data of a single person. As algorithm $A$ is $\epsilon$-differentially private if for all events $E$,

$$P(A(D) \in E) \leq e^{\epsilon}P(A(D') \in E) \approx (1 + \epsilon)P(A(D') \in E)$$

$\epsilon$ is a parameter. The smaller it is, the more “private” the algorithm.

The probability of any response is approximately the same, regardless of what data I provide.

$D =$ everyone else’s data + your true data,

$D' =$ everyone else’s data + your fake data.

$A$ may take as input the true datapoint and output something private.
Imagine two parallel universes, one where you lied and one where you told the truth. Processing the data in a differentially private way ensures that these two universes are almost identical in every way.
Post-processing

Imagine two parallel universes, one where you lied and one where you told the truth. Processing the data in a differentially private way ensures that these two universes are almost identical in every way.

In particular,

- I can’t determine which universe we are in.
- You can’t force significant changes in our universe by altering your answer.
- Any positive or negative consequences you suffer as a result of the analysis are almost equally as likely to occur in both universes.
Would you lie to me?

I am going to take the estimate of how many students cheat and decide whether or to give oral exams in your class.
Would you lie to me?

I am going to take the estimate of how many students cheat and decide whether or to give oral exams in your class. You now have a motivation to lie, outside of just being embarrassed, because don't want an oral exam.
Would you lie to me?

I am going to take the estimate of how many students cheat and decide whether or to give oral exams in your class. You now have a motivation to lie, outside of just being embarrassed, because don’t want an oral exam. Since the randomised response algorithm was $1.1$-differentially private:

$$\mathbb{E}[u(f(x_1, \cdots, x_{n-1}, A(x_n)))] \leq e^{1.1} \mathbb{E}[u(f(x_1, \cdots, x_{n-1}, A(x_n)))]$$

$x_1, \cdots, x_{n-1}$ is everyone else’s data

$A$ is randomised response.

The expectation is over the coin flips in $A$

$u$ is a quantification of your happiness, which depends on my decision.

$f$ is the function I use to make my decision.
Would you lie to me?

I am going to take the estimate of how many students cheat and decide whether or to give oral exams in your class. You now have a motivation to lie, outside of just being embarrassed, because don't want an oral exam. Since the randomised response algorithm was 1.1-differentially private:

\[ E[u(f(x_1, \ldots, x_{n-1}, A(x_n)))] \leq e^{1.1} E[u(f(x_1, \ldots, x_{n-1}, A(x_n)))] \]

\( A \) is randomised response.
\( x_1, \ldots, x_{n-1} \) is everyone else's data
The expectation is over the coin flips in \( A \)
\( u \) is a quantification of your happiness, which depends on my decision.
\( f \) is the function I use to make my decision.

That is, you can't increase your expected happiness very much lying.
For the remainder of this talk, you are all rational and self-interested.
For the remainder of this talk, you are all rational and self-interested.

**Game Theory** is the study of mathematical models of conflict and cooperation between intelligent rational decision-makers. Usually, the rules are given to us. For example, finding an optimal strategy in chess is a game theory problem.
For the remainder of this talk, you are all rational and self-interested.

Game Theory is the study of mathematical models of conflict and cooperation between intelligent rational decision-makers. Usually, the rules are given to us. For example, finding an optimal strategy in chess is a game theory problem.

In Mechanism Design or “reverse game theory”, I get to decide the rules. I get to design the game in such a way that the player’s behave how I would like them to behave. For example, auction design is a mechanism design problem.
There is a set of outcomes $\mathcal{O}$ and $n$ people. Suppose each person has a type, $t_i \in \mathcal{T}$, and each type has a utility function, $u_i : \mathcal{O} \to \mathbb{R}^+$, which is how much a person of that type values each outcome.
Formalising Mechanism Design

There is a set of outcomes $\mathcal{O}$ and $n$ people. Suppose each person has a type, $t_i \in \mathcal{T}$, and each type has a utility function, $u_i : \mathcal{O} \to \mathbb{R}^+$, which is how much a person of that type values each outcome.

Each person reports their type and I use a mechanism $M : \mathcal{T}^n \to \mathcal{O}$ to decide an outcome.
There is a set of outcomes $\mathcal{O}$ and $n$ people. Suppose each person has a type, $t_i \in \mathcal{T}$, and each type has a utility function, $u_i : \mathcal{O} \to \mathbb{R}^+$, which is how much a person of that type values each outcome.

Each person reports their type and I use a mechanism $\mathcal{M} : \mathcal{T}^n \to \mathcal{O}$ to decide an outcome.

**Your aim** is to maximise your utility, i.e. maximise $\mathbb{E}_{o \sim \mathcal{M}(D)}[u_i(o)]$. 
Formalising Mechanism Design

There is a set of outcomes $\mathcal{O}$ and $n$ people. Suppose each person has a type, $t_i \in \mathcal{T}$, and each type has a utility function, $u_i : \mathcal{O} \to \mathbb{R}^+$, which is how much a person of that type values each outcome.

Each person reports their type and I use a mechanism $\mathcal{M} : \mathcal{T}^n \to \mathcal{O}$ to decide an outcome.

**Your aim** is to maximise your utility, i.e. maximise $\mathbb{E}_{o \sim \mathcal{M}(D)}[u_i(o)]$.

**My aim** is maximise some other function, $q : \mathcal{T}^n \times \mathcal{O} \to \mathbb{R}^+$. 
Formalising Mechanism Design

There is a set of outcomes $\mathcal{O}$ and $n$ people. Suppose each person has a type, $t_i \in \mathcal{T}$, and each type has a utility function, $u_i : \mathcal{O} \rightarrow \mathbb{R}^+$, which is how much a person of that type values each outcome.

Each person reports their type and I use a mechanism $M : \mathcal{T}^n \rightarrow \mathcal{O}$ to decide an outcome.

**Your aim** is to maximise your utility, i.e. maximise $\mathbb{E}_{o \sim M(D)}[u_i(o)]$.

**My aim** is maximise some other function, $q : \mathcal{T}^n \times \mathcal{O} \rightarrow \mathbb{R}^+$.

Our aims might at odds! You are rational and self-interested so you might lie to me in order to increase your utility! Your lie may decrease the benefit I see.
Example: Digital Goods Auction

I have a groovy CD that I would like to sell and $n$ possible buyers.
I have a groovy CD that I would like to sell and $n$ possible buyers.

- The outcome space $\mathcal{O}$ is the set of possible prices for the CD.
- Your type is the maximum amount you would pay for the CD, $v_i$.
- Your utility is $u(p) = v_i - p$ if you purchase and 0 otherwise. So you will always purchase if $p < v_i$.
- I am trying to maximise my profit, $q(D, p) = p \cdot |\{i \mid p < v_i\}|$. 

You are motivated with underestimate your value for the CD. If you lie, then I might undervalue the CD and lose profit. In order to make the most informed decision, I need you to tell the truth.
Example: Digital Goods Auction

I have a groovy CD that I would like to sell and $n$ possible buyers.

- The outcome space $\mathcal{O}$ is the set of possible prices for the CD.
- Your type is the maximum amount you would pay for the CD, $v_i$.
- Your utility is $u(p) = v_i - p$ if you purchase and 0 otherwise. So you will always purchase if $p < v_i$.
- I am trying to maximise my profit, $q(D, p) = p \cdot |\{i \mid p < v_i\}|$.

You are motivated with underestimate your value for the CD. If you lie, then I might to undervalue the CD and lose profit. In order to make the most informed decision, I need you to tell the truth.
A **dominant strategy** is a strategy such that, regardless of what the other’s player do, the player can’t improve their utility by playing a different strategy.

A mechanism $\mathcal{M} : \mathcal{T}^n \to \mathcal{O}$ is $\epsilon$-approximately **dominant strategy truthful** if no player can substantially improve their expected utility by lying.
A dominant strategy is a strategy such that, regardless of what the other’s player do, the player can’t improve their utility by playing a different strategy.

A mechanism $\mathcal{M} : \mathcal{T}^n \rightarrow \mathcal{O}$ is $\epsilon$-approximately dominant strategy truthful if no player can substantially improve their expected utility by lying. That is, if for every player $i$, and pair of types $t, t' \in \mathcal{T}$:

$$\mathbb{E}_{o \sim \mathcal{M}([D,t])}[u_i(o)] \leq \mathbb{E}_{o \sim \mathcal{M}([D,t'])}[u_i(o)] + \epsilon$$
Approximately dominant strategy truthfulness

A dominant strategy is a strategy such that, regardless of what the other’s player do, the player can’t improve their utility by playing a different strategy.

A mechanism $\mathcal{M} : \mathcal{T}^n \to \mathcal{O}$ is $\epsilon$-approximately dominant strategy truthful if no player can substantially improve their expected utility by lying. That is, if for every player $i$, and pair of types $t, t' \in \mathcal{T}$:

$$\mathbb{E}_{o \sim \mathcal{M}([D, t])}[u_i(o)] \leq \mathbb{E}_{o \sim \mathcal{M}([D, t'])}[u_i(o)] + \epsilon$$

**Lemma:** If $u_i \in [0, 1]$ and $\epsilon \leq 1$ then any $\epsilon$-differentially private mechanism is also $2\epsilon$-approximately dominant strategy truthful.
We’ve established that if I make my decision using a differentially private 
mechanism then you have little incentive to lie. But we still have two 
wrinkles.
We’ve established that if I make my decision using a differentially private mechanism then you have little incentive to lie. But we still have two wrinkles.

- **Degraded utility.** Recall that the data from randomised response was noisy. We want to make sure that the data was are getting is still useful. Remember our utility is probably still better than it would be if everyone lied.
Wrinkles

We’ve established that if I make my decision using a differentially private mechanism then you have little incentive to lie. But we still have two wrinkles.

- **Degraded utility.** Recall that the data from randomised response was noisy. We want to make sure that the data we are getting is still useful. Remember our utility is probably still better than it would be if everyone lied.

- **Truthfulness is not the only approximately dominant strategy.** Actually, any answer is an approximately dominant strategy!
Actually differentially private algorithms often have good utility!

Assume $u_i \in [0, 1]$ and $\epsilon < 1$. Let

$$\Delta = \max_{D, D'} \max_{o \in \mathcal{O}} |q(D, o) - q(D', o)|$$

be the maximum amount that you can change my quality function by lying.

**Theorem** There exists a $\epsilon$-differentially private algorithm $M : \mathcal{T}^n \rightarrow \mathcal{O}$, that with probability $1 - \beta$ outputs $o \in \mathcal{O}$, such that

$$q(D, o) \geq \max_{o^* \in \mathcal{O}} q(D, o^*) - \frac{2\Delta}{\epsilon} \ln \frac{1}{\beta}$$
Actually differentially private algorithms often have good utility!

Assume $u_i \in [0, 1]$ and $\epsilon < 1$. Let

$$\triangle = \max_{D, D', o} \max_{o \in \mathcal{O}} |q(D, o) - q(D', o)|$$

be the maximum amount that you can change my quality function by lying.

**Theorem** There exists a $\epsilon$-differentially private algorithm $M : T^n \rightarrow \mathcal{O}$, that with probability $1 - \beta$ outputs $o \in \mathcal{O}$, such that

$$q(D, o) \geq \max_{o^* \in \mathcal{O}} q(D, o^*) - \frac{2\triangle}{\epsilon} \ln \frac{\mathcal{O}}{\beta}$$

That is, the mechanism **simultaneously** satisfies

- Lying doesn’t buy you very much (so you might as well tell the truth)
- The benefit I see is almost as good as if everyone had just directly told me their true value!
How do we get rid of the caveat that differentially private mechanisms are only approximately truthful?
How do we get rid of the caveat that differentially private mechanisms are only approximately truthful?

We can pair our differentially private mechanism (which had good utility for me) with a truth-forcing mechanism (which might have bad utility for me). We can randomise between the two algorithms. If we distribute the weight correctly, then we can inherit both truth-forcing and decent utility.
How can we design a truth-forcing mechanism? (BRIEFLY)

Remember that I, the mechanism designer, get to make up the rules.
How can we design a truth-forcing mechanism? (BRIEFLY)

Remember that I, the mechanism designer, get to make up the rules.

One way to force truth-telling is to enforce that the type you report is binding. That is, after I made a decision, you must act as if that is your type.

Recall our auction example: As the auctioneer, I can decree that the price you report is binding. So, if you underestimate your price then you might miss the opportunity to purchase my groovy CD!
Results of the Survey! The Important Stuff

- What fraction of people (who are on the applied math listserv and likely to take such a quiz) like Justin Beiber? Recall this question was answered using randomised response.

\[
\frac{1}{2} \pm \frac{2}{\sqrt{n}} = \frac{1}{2} \pm \frac{2}{\sqrt{12}} = \frac{1}{2} \pm 0.6
\]

with probability \( e^{-2} \approx 0.1 \).
What fraction of people (who are on the applied math listserv and likely to take such a quiz) like Justin Beiber? Recall this question was answered using randomised response.

\[
\frac{1}{2} \pm \frac{2}{\sqrt{n}} = \frac{1}{2} \pm \frac{2}{\sqrt{12}} = \frac{1}{2} \pm 0.6
\]

with probability \( e^{-2} \approx 0.1 \).

How many days last week did you brush your teeth? This question was answered by adding Gaussian noise with variance 13.2.

\[
13.2 \pm \frac{2\sigma}{\sqrt{n}} = 13.2 \pm 7.6
\]

with probability \( e^{-2} \).
Results of the Survey! The Important Stuff

- What fraction of people (who are on the applied math listserv and likely to take such a quiz) like Justin Beiber? Recall this question was answered using randomised response.

\[
\frac{1}{2} \pm \frac{2}{\sqrt{n}} = \frac{1}{2} \pm \frac{2}{\sqrt{12}} = \frac{1}{2} \pm 0.6
\]

with probability \( e^{-2} \approx 0.1 \).

- How many days last week did you brush your teeth? This question was answered by adding Gaussian noise with variance 13.2.

\[
13.2 \pm \frac{2\sigma}{\sqrt{n}} = 13.2 \pm 7.6
\]

with probability \( e^{-2} \).

The error bounds are pretty bad. Differential privacy, unsurprisingly does not perform very well for small sample sizes. But neither do traditional statistics. However, the error rates decay fairly quickly as the sample size increases!
Thank you!

If you’re interested in differential privacy, it’s applications to mechanism design or machine learning, come and talk to me!