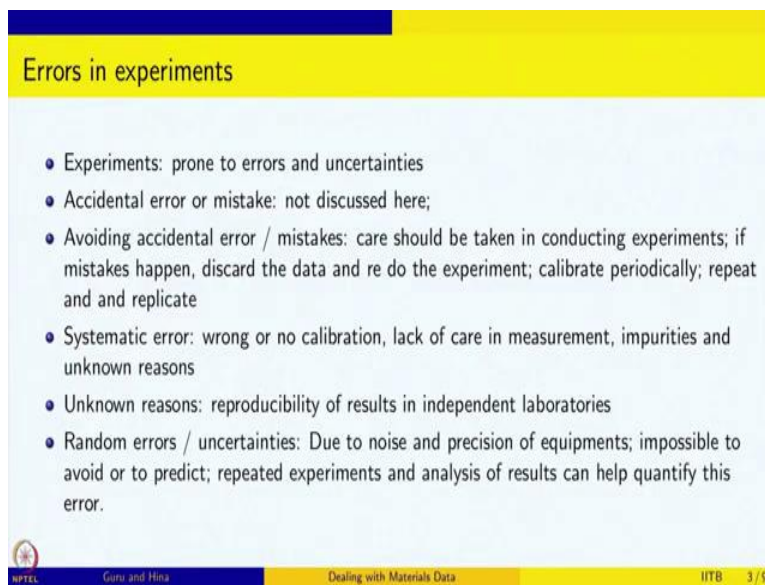


Dealing with Materials Data: Collection, Analysis and Interpretation
Professor. M P Gururajan
Professor. Hina A Gokhale
Department of Metallurgical Engineering and Materials Science
Indian Institute of Technology, Bombay
Lecture 27
Errors and their propagation

Welcome to dealing with materials data. We are looking at the Collection, Analysis and Interpretation of data. Specifically, in this module we are looking at how to do descriptive data analysis using R and one of the aspects that we need to address is errors and we saw how to present data when we know what the errors are, and in this session we are going to learn how to do the analysis and propagation of errors. How do we understand the error propagation?

(Refer Slide Time: 00:53)



Errors in experiments

- Experiments: prone to errors and uncertainties
- Accidental error or mistake: not discussed here;
- Avoiding accidental error / mistakes: care should be taken in conducting experiments; if mistakes happen, discard the data and re do the experiment; calibrate periodically; repeat and replicate
- Systematic error: wrong or no calibration, lack of care in measurement, impurities and unknown reasons
- Unknown reasons: reproducibility of results in independent laboratories
- Random errors / uncertainties: Due to noise and precision of equipments; impossible to avoid or to predict; repeated experiments and analysis of results can help quantify this error.

MPTEL Gokh and Hina Dealing with Materials Data IITB 3 / 9

So, this is about errors and their propagation. The first question is, why are there are errors in experiments? Typically, you will find that any experiment that you do is prone to errors and uncertainties and the errors are of different types. First one is accidental error or a mistake and that is not discussed in detail in this course at all, because they can be avoided.

To avoid accidental error or mistakes, one should do experiments carefully and if in spite of that if mistakes happen one should discard the data and redo the experiment and if you know that there is something that is not correct you should not take the data at all, and you should periodically

calibrate your equipment to make sure that the values that you are reading are actually the correct values.

So, if the calibration does not give you right values, then you should again discard the data and recalibrate and redo the experiment. And it is also very important to repeat and replicate the experiments. Repetition means like we did in the case of copper conductivity for example, same sample you will make the measurement 20 different times. So, this is repetition, but replication means you will go through the entire process once more.

Suppose, you said 2.9 percent deformed sample, so you have deformed a sample and in that sample, if you take 2 or 3 different pieces and did the experiment 20 times on each one of them, this is repetition of the experiment. So, you have a sample which has gone undergone 3 percent deformation and you take several samples and each time you do lots of measurements, so that you are statistics is good and if you did the several such repetitions, then if you are experiment is repeatable, and every time you are getting the same mean and similar standard deviation and so on.

So, you know that your experiment is okay. But replication means you have to repeat the entire process once more. Suppose, you had taken the sample, again you do a 3 percent deformation and then you take several such samples, and then do several experiments. So, this is also very important and repetition and the replication is one way of avoiding accidental errors or mistakes.

Than there could be systematic errors, this happens if your calibration is wrong or if you have not done calibration, so you are measuring some quantity but it is systematically more or less depending on how the calibration went, and it can also happen if you are not careful when you are doing experiments and it can happen, for example, in material science it is very common because there are impurities.

For example, the surface tension measurements have been improving over the period because people are able to do, people are able to remove impurities and get more and more purer materials and so the measurements have been improving. So, systematic errors can come because of the constraints. So, you have some material and it has some impurities and there is no way you can change it. Then there is an error associated with it. And the most dangerous form of systematic

error is some error because of some unknown reason, because it is unknown there is no way you can correct for it.

But one way, that one can learn about such unknown reasons or correct for it is to reproduce the results in a completely different laboratory, in a different group, for example, if they are also able to reproduce the results, then you can assume that there might not be errors of this type even though one cannot completely rule them out, but at least if the experiment is reproducible in different parts and in different conditions in different labs, one can assume that systematic errors are also not there. And the third type of error which is called random error or uncertainty is due to noise and due to precision of equipment's.

So, this error cannot be avoided and it is also not possible to predict it. Repeated experiments and analysis will give some idea about these uncertainties. So, like we did for example, you take the same sample and everything is same, you do 20 measurements you do not get the same number every time. This is because of random noise or there is some problem with the precision of the equipment, the equipment can only measure up to some precision. So, you are getting numbers which are differing beyond that precision.

(Refer Slide Time: 05:45)

Know your uncertainties and errors

- Joke about engineers: Measure with a vernier calliper, mark with chalk and cut with saw!!
- If you are planning to mark with chalk or cut with saw, measuring more accurate than a ruler wastes money and effort!!

IITB 4/9

So, there is a joke about engineers, which says that you measure with a vernier calliper, mark with a chalk and cut with a saw. So, if you are planning to mark with the chalk or cut with a saw, you should not be measuring using a vernier calliper, I know just ruler will do. In other words when

you are doing experiments you should know the different errors that you are going to encounter and so if it is not meaningful for you to measure something very very accurately because you later know that something else is going to mark that accuracy or it is not going to give you to that accuracy, then you can save yourself lots of time effort money by doing things only up to the required accuracy and precision.

(Refer Slide Time: 06:33)

The slide is titled "Propagation of errors" and contains the following content:

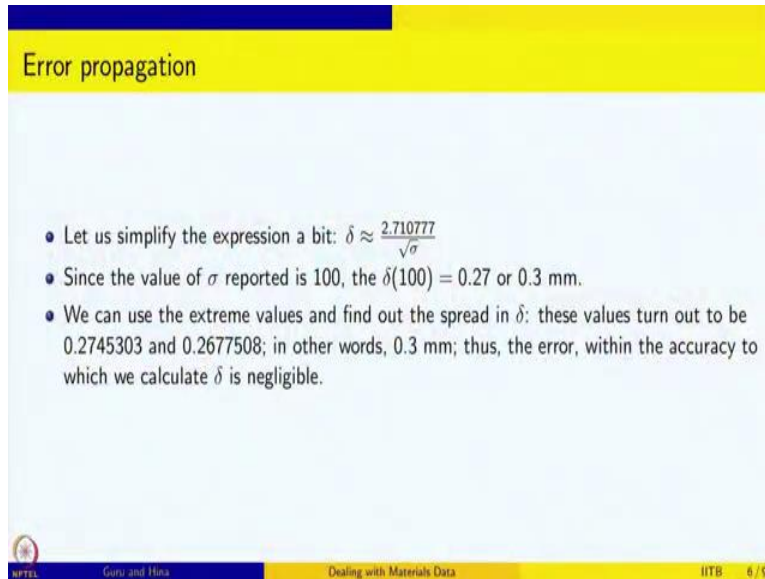
- Errors in one quantity propagate through calculations to other quantities!
- Consider the skin depth formula that we discussed earlier:
$$\delta \approx \frac{664}{\sqrt{f\mu_r\sigma}}$$
- Let us say that the σ in our case has an uncertainty of about 5 units; that is, let us say that we calculated δ assuming $\sigma = 100$ whereas actual σ lies anywhere in the range 97.5 to 102.5. What can you say about the uncertainty in δ ?

At the bottom of the slide, there is a footer with the text "Guru and Hina" and "Dealing with Materials Data".

So, this is very very important and the next thing about errors is that let us say that you have some random errors or uncertainty in one of the quantities, but these errors actually propagate through your calculations to other quantities. The skin depth formula that we discussed earlier I mentioned at the end of that session. That, the formula is delta is 664 by square root of f mu r sigma. We can assume that f and mu are known and they do not have any error and I have just some given number.

Which might not be true, but at least for the moment, let us assume that they are given f is 60 kilo Hertz and mu r is 0.99994. If so, then any error that is there in sigma or uncertainty that is there in sigma measurement will actually also affect the delta measurement. Let us say for argument sake, that sigma is 100, is what we took, but it has an error bar. The error bar is plus or minus 2.5. Let us say that it can lie anywhere between 97.5 and 102.5. What is the uncertainty that you will get in delta because you have uncertainty in the value of sigma? Is a question that one can ask, and with this being a very simple case it can also be answered very easily?

(Refer Slide Time: 07:54)



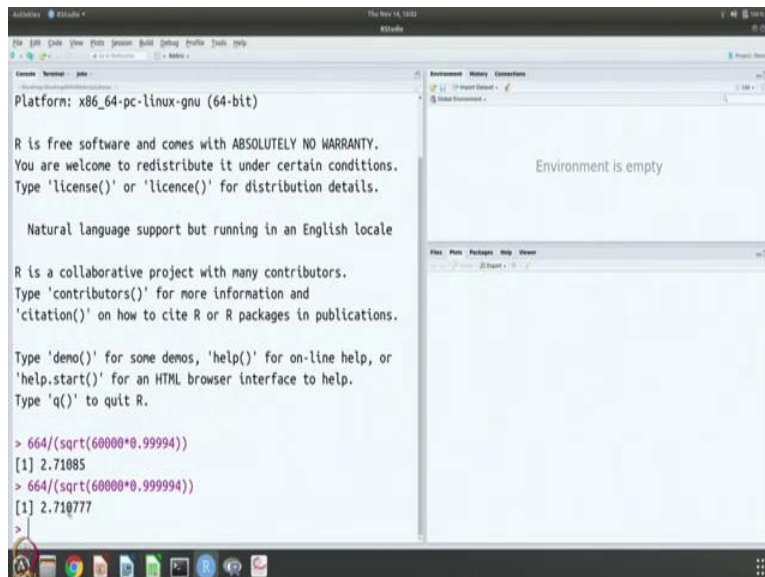
Error propagation

- Let us simplify the expression a bit: $\delta \approx \frac{2.710777}{\sqrt{\sigma}}$
- Since the value of σ reported is 100, the $\delta(100) = 0.27$ or 0.3 mm.
- We can use the extreme values and find out the spread in δ : these values turn out to be 0.2745303 and 0.2677508; in other words, 0.3 mm; thus, the error, within the accuracy to which we calculate δ is negligible.

NPTEL Guru and Hina Dealing with Materials Data IITB 6/9

So, let us simplify the expression a little bit. So, use the values of f and mu r, and you get an expression which is 2.71. So, how do we get that, let us do that.

(Refer Slide Time: 08:07)



```
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 664/(sqrt(60000*0.99994))
[1] 2.71085
> 664/(sqrt(60000*0.999994))
[1] 2.710777
>
```

R can also be used as a calculator. So, we want to get 664 divided by square root of 60000 and 0.99994. So, you get the 2.71085 and that is what is given here. So, 2.710777 and that is the value that is given here.

(Refer Slide Time: 08:42)

The slide is titled "Error propagation" and contains the following text:

- Let us simplify the expression a bit: $\delta \approx \frac{2.710777}{\sqrt{\sigma}}$
- Since the value of σ reported is 100, the $\delta(100) = 0.27$ or 0.3 mm.
- We can use the extreme values and find out the spread in δ : these values turn out to be 0.2745303 and 0.2677508; in other words, 0.3 mm; thus, the error, within the accuracy to which we calculate δ is negligible.

At the bottom of the slide, there is a footer with the following information: IITB, Guru and Hina, Dealing with Materials Data, IITB 6/9.

So that is divided by root sigma, is the expression for delta. Now the value of sigma is reported as 100. So, if you calculate delta for 100 you get 0.27 or it is 0.3 millimetre. Now you can use the other extreme values that you have. Let us say that you have values like 97.5 and 102.5, you can again substitute these values in this expression and you find that value turns out to be 0.2745 and 0.2677. In other words, they also happen to be 0.3 millimetres, we are not going to measure beyond this accuracy let us say then all the 3 values happen to be 0.3. So, the error within the accuracy to which we calculate delta becomes negligible.


However, if we are calculating delta to some fourth or fifth decimal place, let us say we were measuring in microns for example, then these different values will actually have errors and so using this very straight forward method I mean, we have a range of we just substitute for the min and the max the extreme values, and so we know what are the extreme values that delta itself can take. So, we know the error is in this range, the value lies in this range so you can know the what the error is.

(Refer Slide Time: 10:14)

Uncertainty using formula

$$\sigma_{\delta} = \left| \frac{\partial \delta}{\partial \sigma} \right| \Delta \sigma \quad (1)$$
$$\sigma_{\delta} = \frac{2.710777}{2(\sigma)^{1.5}} \Delta \sigma \quad (2)$$

This values is 0.006767 or 0.007. Note that this is consistent with our earlier calculations ($-0.2677508 - 0.2745303 = 0.006780$).

 NIPTEL Guru and Mina Dealing with Materials Data IITB 7/9

There is also another way of calculating the same quantity. Now the sigma of delta that is the standard deviation in delta. So, there are 2 sigma's, so one has to be careful. The left side is the standard deviation, that is obtained by taking the partial derivative of delta with respect to the conductivity sigma and multiplying by the uncertainty in the conductivity. So, if you take the, and it is modulus. So, we are going to take only the positive value. So, if you take this 2.710777 divided by square root of sigma, and you take the derivative then it is sigma to the power minus half on which you are taking derivative.

So, you will get minus by minus half and sigma to the power 3 by 2 in the denominator, and so that is the expression that is written here and you can see that this is the expression, and now you know what is the sigma that you are using which is 100 and you know what is the delta sigma that is 2.5 and the plus or minus 2.5, so you can put this values and you will get the error and that is the same 0.007 is what we calculated even from the range earlier and you are getting the same value here also.

(Refer Slide Time: 11:46)

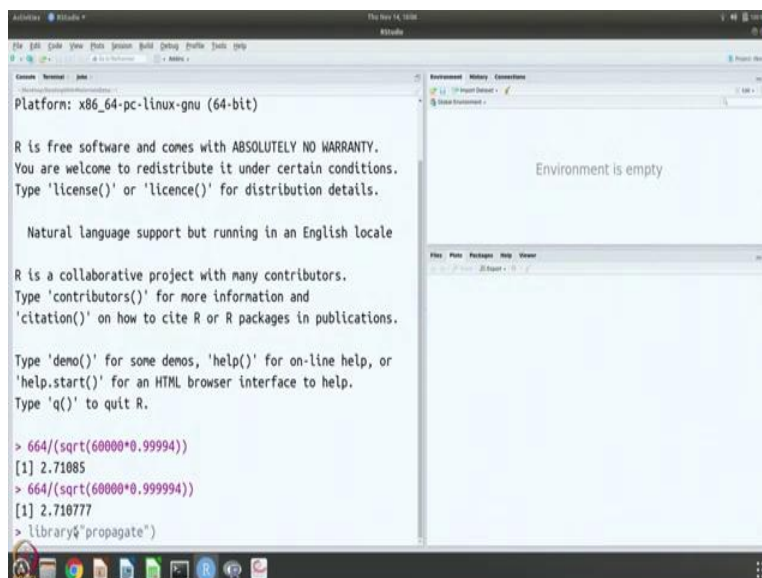
Uncertainty propagation using simulations

- Simulations: a more generic approach
- Can be used even when functional relation between result and several other variables are not known
- propagate library and how it works for our simple case!

NPTL Guru and Mina Dealing with Materials Data IITB 8 / 9

There is one more way of doing uncertainty propagation, that is using simulations and using simulations is a more generic approach. It can be used even when functional relations between the result and several other variables are not known, and there is a library called propagate and so you can use that to solve any of the complex problems of uncertainty propagation and we are going to do a few examples of that later, but for now, I want to take the same simple case, you know, it is a very very simple case.

(Refer Slide Time: 12:30)



```
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 664/(sqrt(60000*0.99994))
[1] 2.71885
> 664/(sqrt(60000*0.99994))
[1] 2.718777
> library("propagate")
```



```
## is.factor, is.ordered
## Loading required package: minpack.lm
x <- rnorm(100,mean=100,sd=2.5)
Z <- data.frame("x"=x)
error <- propagate(expression(2.710777/sqrt(x)),Z)
summary(error)

## Results from error propagation:
##      Mean.1      Mean.2      sd.1      sd.2      2.5%      97.5%
## 0.271339633 0.271409850 0.003563964 0.003665347 0.264421890 0.278397810

## Results from Monte Carlo simulation:
##      Mean      sd      Median      MAD      2.5%      97.5%
## 0.271408312 0.003527163 0.271669933 0.003224807 0.263970476 0.277775755

## Welch-Satterthwaite degrees of freedom:
## [1] 1001553

## Coverage factor (k):
## [1] 1.959966
```

Can we just use propagate and find out how the error propagation happens, and that is what we want to do now. So, let us go to R and let us do the, so first we have to get the library propagate. So, the library is in place now we can get this command.

(Refer Slide Time: 13:01)

```
Attaching package: 'ff'

The following objects are masked from 'package:bit':
  clone, clone.default, clone.list

The following objects are masked from 'package:utils':
  write.csv, write.csv2

The following objects are masked from 'package:base':
  is.factor, is.ordered

Loading required package: minpack.lm
> x <- rnorm(100,mean=100,sd=2.5)
> Z <- data.frame("x"=x)
> error <- propagate(expression(2.710777/sqrt(x)),Z)
> summary(error)
```

So, what is this command, this is the error that propagation we are calculating and that is using the expression 2.71 by square root of x and for doing error propagation, we have to give either the mean and standard deviation or some data generated using a simulation, and this error propagation itself is going to be done using Monte Carlo simulation, and we want to look at the resultant values.

(Refer Slide Time: 13:38)

The first screenshot shows the RStudio console with the following code and output:

```
A <- rnorm(100, mean=100, sd=2.2)
Z <- data.frame("x"=x)
error <- propagate(expression(2.710777/sqrt(x)),Z)
summary(error)
```

Results from error propagation:

Mean.1	Mean.2	sd.1	sd.2	2.5%	97.5%
0.270617641	0.270676899	0.003269676	0.003270750	0.264266340	0.277087458

Results from Monte Carlo simulation:

Mean	sd	Median	MAD	2.5%	97.5%
0.270676667	0.003289076	0.270652713	0.002972072	0.264267969	0.278241548

Welch-Satterthwaite degrees of freedom: [1] 1001314
Coverage Factor (k): [1] 1.959966
Expanded uncertainty: [1] 0.006410559
Symbolic gradient matrix: [1] "(2.710777 * (1/A * x^-0.5)) / (sqrt(x)^2)"

The second screenshot shows the RStudio console with the following code and output:

```
[1] -0.001348499
Symbolic hessian matrix:
[1] "-(2.710777 * (0.5 * (-0.5 * x^-1.5)) / sqrt(x)^2 - 2.710777 * (0.5 * x^-0.5) * (2 * (0.5 * x^-0.5 * sqrt(x)) / (sqrt(x)^2)^2)"
Evaluated hessian matrix:
[1] 2.015889e-05
Covariance matrix:
      x
x 5.87906
Relative contribution:
      x
x 1
Skewness / Excess Kurtosis of MC evaluations:
0.3803128 / 1.094067
Shapiro-Wilk test for normality:
0.07019516 => normal
Kolmogorov-Smirnov test for normality:
0.07832309 => normal
> help("propagate")
```

The right-hand pane shows the Environment window with 'Data' containing 'error' (List of 16) and 'Z' (100 obs. of 1 variable). The Values window shows 'X' with values: nun [1:100] 96.3 101.3 106.3 99.4 10...

The bottom pane shows the help documentation for the 'propagate' function, including a description and arguments.

So, you can see that this analysis also, so it is much more complicated. So, we will come back again and take a relook at this. So, the mean is again 0.27 and so Monte Carlo simulation also gives you 0.27 and the uncertainty happens to be 0.0064 which is similar to 0.007 that we calculated earlier. So, what is this propagate? And so you can use the help. You can see that this is what we use for error propagation, you have to give the expression and it is clear what the expression is and you also have to give some data and that is what we generated here.

(Refer Slide Time: 14:21)

```
The following objects are masked from 'package:utils':  
write.csv, write.csv2  
  
The following objects are masked from 'package:base':  
is.factor, is.ordered  
  
Loading required package: minpack.ln  
> x <- rnorm(100, mean=100, sd=2.5)  
> Z <- data.frame("x"=x)  
> error <- propagate(expression(2.710777/sqrt(x)), Z)  
> summary(error)  
Results from error propagation:  
      Mean.1      Mean.2      sd.1      sd.2      2.5%  
0.270617641 0.270676899 0.003269676 0.003270750 0.264266340  
      97.5%  
0.277087458  
Results from Monte Carlo simulation:  
      Mean      sd      Median      MAD      2.5%  
0.270617641 0.270676899 0.003269676 0.003270750 0.264266340  
      97.5%  
0.277087458
```

We used normal distribution 100 numbers with mean 100 and standard deviation 2.5 because that was the parameters that we assume. Sigma had a standard deviation of 2.5 and mean of 100 is what we assumed, and so you have to generate data of that type. So, this is explained here in the R documentation.

(Refer Slide Time: 14:52)

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
>  
> R is a collaborative project with many contributors.  
> Type 'contributors()' for more information and  
> 'citation()' on how to cite R or R packages in publications.  
  
Re Type 'demo()' for some demos, 'help()' for on-line help, or  
> 'help.start()' for an HTML browser interface to help.  
> Type 'q()' to quit R.  
  
0 > help("propagate")  
Re+
```

```
getOption("ffbatchbytes")==16777216 -- consider a different value for tuning your system
getOption("ffmaxbytes")==536870912 -- consider a different value for tuning your system

Attaching package: 'ff'

The following objects are masked from 'package:bit':
  clone, clone.default, clone.list

The following objects are masked from 'package:utils':
  write.csv, write.csv2

The following objects are masked from 'package:base':
  is.factor, is.ordered

Reloading required package: minpack.lm
```

```
propagate package:propagate R Documentation
Propagation of uncertainty using higher-order Taylor expansion and Monte Carlo simulation

Description:
  A general function for the calculation of uncertainty propagation by first-/second-order Taylor expansion and Monte Carlo simulation including covariances. Input data can be any symbolic/numeric differentiable expression and data based on summaries (mean & s.d.) or sampled from distributions. Uncertainty propagation is based completely on matrix calculus accounting for full covariance structure. Monte Carlo simulation is conducted using a multivariate t-distribution with covariance structure. Propagation confidence intervals are calculated from the expanded uncertainties by means of the degrees of freedom obtained from 'WelchSatter', or from the  $[\frac{\alpha}{2}, 1-\frac{\alpha}{2}]$  quantiles of the MC evaluations.

Usage:
  propagate(expr, data, second.order = TRUE, do.sim = TRUE, cov = TRUE,
```

```

Usage:
  propagate(expr, data, second.order = TRUE, do.sim = TRUE, cov = TRUE,
            df = NULL, nsim = 1000000, alpha = 0.05, ...)

Arguments:
  expr: an expression, such as 'expression(x/y)'.

  data: a dataframe or matrix containing either a) the means mu_i,
        standard deviations sigma_i and degrees of freedom nu_i
        (optionally) in the first, second and third (optionally) row,
        or b) sampled data generated from any of R's 'distributions'
        or those implemented in this package ('rDistr'). If
        'nrow(data)' > 3, sampled data is assumed. The column names
        must match the variable names.

  second.order: logical. If 'TRUE', error propagation will be calculated
                with first- and second-order Taylor expansion. See 'Details'.

  do.sim: logical. Should Monte Carlo simulation be applied?

```

So, probably it is clearer if we use R. So, you can see that propagate requires the expression and it also requires the data. It should be either a data frame or a matrix. It can contain the means and standard deviations and degrees of freedom, and degrees of freedom is optional and, or you can have a sampled data generated from any of R's distributions and that is what we have done.

(Refer Slide Time: 16:08)

```

The following objects are masked from 'package:utils':
  write.csv, write.csv2

The following objects are masked from 'package:base':
  ls.factor, ls.ordered

Loading required package: minpack.ln
> x <- rnorm(100, mean=100, sd=2.5)
> Z <- data.frame("x"=x)
> error <- propagate(expression(2.710777/sqrt(x)), Z)
> summary(error)
Results from error propagation:
      Mean.1 Mean.2 sd.1 sd.2 2.5%
0.270617641 0.270676899 0.003269676 0.003270750 0.264266340
      97.5%
0.277087458
Results from Monte Carlo simulation:
      Mean      sd      Median      MAD      2.5%
0.270617641 0.003269676 0.270617641 0.003269676 0.264266340

```

We have actually generated sample data from the R's distributions and so this, this command that we have here, is basically generates from the distribution rnorm. So, it generates 100 data points, with this mean and with the standard deviation and it turns it into a data frame and the data frame

has to be named appropriately, it should be called the x, you know, this labelling is important that is how it knows that this is the x.

So in other words, instead of doing experiments because we know that this is the mean, this is a standard deviation, on computer we are generating pseudo data, and using that data into this formula we are finding what is the range in which the delta lies, and using those values we are calculating how much is the error that we found in the delta. So, that is what it is done and the result is given and result is given in greater detail. So, some of these things we will come back and take a look at later.

(Refer Slide Time: 16:58)

```

* (0.5 * x^0.5) * (2 * (0.5 * x^0.5 * sqrt(x)))/(sqrt(x)^2)
^2)
Evaluated hessian matrix:
[1] 2.015889e-05
Covariance matrix:
      x
x 5.87906
Relative contribution:
      x
x 1
Skewness / Excess Kurtosis of MC evaluations:
0.3803128 / 1.094067
Shapiro-Wilk test for normality:
0.07019516 => normal
Kolmogorov-Smirnov test for normality:
0.07832309 => normal
> help("propagate")
> x <- rnorm(10, mean=100, sd=2.5)
Z <- data.frame("x"=x)
error <- propagate(expression(2.710777/sqrt(x)), Z)
summary(error)
  
```

```

Evaluated gradient matrix (sensitivity):
[1] -0.001356327
Symbolic hessian matrix:
[1] ^(-(2.710777 + (0.5 * (-0.5 * x^-1.5))/sqrt(x))^2 - 2.710777
* (0.5 * x^0.5) * (2 * (0.5 * x^0.5 * sqrt(x)))/(sqrt(x)^2)
^2)
Evaluated hessian matrix:
[1] 2.035429e-05
Covariance matrix:
      x
x 6.482067
Relative contribution:
      x
x 1
Skewness / Excess Kurtosis of MC evaluations:
0.04046992 / -0.09643077
Shapiro-Wilk test for normality:
0.9537292 => normal
Kolmogorov-Smirnov test for normality:
0.6852314 => normal
  
```

So, of course you can also change. So, let us say that we change the n_{norm} to be 10, what do you see? And you can change it to be 1000 and what do you see. So, as you increase the number of data points, there is no change in these values it still remains 0.27 and the uncertainty still remains at 0.007. So, you can see. So, this is a way of doing the analysis.

(Refer Slide Time: 17:51)

The slide has a yellow header with the title "Uncertainty propagation using simulations". The main content area is light blue and contains three bullet points. At the bottom, there is a footer with the NPTEL logo, the text "Guru and Hina", "Dealing with Materials Data", and "IITB 8 / 9".

- Simulations: a more generic approach
- Can be used even when functional relation between result and several other variables are not known
- propagate library and how it works for our simple case!

So, let us go back. So, simulations is a more generic approach and propagate library allows that to, us to do that.

(Refer Slide Time: 18:01)

The slide has a yellow header with the title "Error propagation: beyond the simple case". The main content area is light blue and contains four bullet points. At the bottom, there is a footer with the NPTEL logo, the text "Guru and Hina", "Dealing with Materials Data", and "IITB 9 / 9".

- We considered a simple expression; calculated error in two ways: direct and using the formula
- What happens if uncertainty is a result of two or more independent variables?
- What happens if uncertainty is a result of several variables which may not be independent, that is, their uncertainties are not independent of each other?
- What happens if we do not have a functional relation between the variables and result?
- Discuss these scenarios in the rest of this module!

Now, beyond the simple case, we looked at a very very simple case. So, there is just an expression that connects delta to sigma, and you can calculate error directly or using the formula which is based on the partial derivative or you can even do some simple generation of data using a Monte Carlo simulation, then you can find out how the error propagates. But what happens if uncertainty is a result of 2 or more independent variables.

Let us say you have a function or you have a variable its value depends on some 3 different variables and those variables have their own uncertainties and how does this calculated parameter, its error depend on the errors on those independent quantities and sometimes let us say a particular quantity f depends on x , y and z , but the uncertainties in x , y and z might not be independent. They might also be related to each other.

So, if there is an interdependence on the uncertainty of these quantities, how does it affect the uncertainty on the quantity that we are trying to calculate. And sometimes you might not even know the functional relationship between the quantity that you are trying to calculate and the quantities on which it depends on even though you know, what is the error on those variables. So, in those cases how do you deal with and understand error propagation or calculate the uncertainty or do the uncertainty propagation analysis.

So, that is the part of the descriptive data analysis using R that we will do and that will complete this module on descriptive data analysis using R. So, we will do that in the following parts. Thank you.