COR1305 - Modelling & Data Analytics

Week 1: Course overview & Business modelling 4
Autofill 4
Referencing 4
Trend() 4
Modelling 4
- Influence Diagrams - Formulate relationships between variables 4
- Black Box View - Classify and Summarise all variables 5
- Construct the model (Excel) 5
- Perform analysis and validate the model 5

Solver 6
Goal Seek(simplified version of solver) 6
Week 2: Financial Modeling 7
- Use a linear formula to get the prediction 7
- Financial Functions 8
  - PMT -> periodic payment 8
  - PV -> present value 8
  - FV -> future value 9
  - Rate → interest rate, investment rate, discount rate 9
  - NPER -> number of periods 9
  - IRR -> internal rate of return for a series of cash flow 10
  - NPV -> net present value 10
- Match() 11
- Data Tables 11
  - One-Variable Data table 11
  - Two-Variable Data table 12
- Circular Reference 12

Week 3: Data Analysis & Manipulation 13
1. Data Tables 13
2. NPV 13
3. IRR → internal rate of return 14
4. Basic Statistical Information 15
   a. Univariate Analysis → analysis of a single variable x 15
5. COUNTIF() 16
6. COUNTIFS() 16
7. SUMIF() 16
8. SUMIFS() 16
9. AVERAGEIF() 16
10. AVERAGEIFS() 16
11. CONCATENATE() or & 16
12. Frequency() 16

Week 4: Descriptive Analytics 17
Lookup Functions 17
- LOOKUP(lookup_value,lookup_vector,result_vector) 17
- VLOOKUP(lookup_value,table_array,col_index_num,range_lookup) 17
- MATCH(lookup_value,lookup_array,[match_type]) 17
- INDEX(array,row_num,column_num) 17
- XLOOKUP(lookup_value,lookup_array,return_array,[if_not_found],[match_mode],[search_mode]) 17
- RANK(number,ref,[order]) 17
- RANK(array,[sort_index],[sort_index],[by_col]) 17
- SMALL(array,k) 17
- LARGE(array,k) 17
- FILTER(array,include,[if_empty]) 17

Descriptive Analytics 17
Week 5: Predictive Analytics 18
In class notes 18
Lookup recap 18
Dynamic Trends 18
LINEST() 19
Quantitative Forecasting 19
- 2 main types of quantitative forecasting 19
- Linear regression 20
- Multiple Linear Regression 20
- Significant value that affects regression 20
- Time Series Forecasting 21
- FORECAST.ETS 21
- FORECAST.ETS.CONFINT 22
- FORECAST.ETS.SEASONALITY 22
- FORECAST.ETS.STAT 22

Predictive Analytics Models 23
Supervised Learning 23
Unsupervised Learning 23
Classification → Decision Tree 23
- Decision Tree Algorithm 23
- Pruning Decision Tree 25

Functions 25
- UNIQUE(array,[by_col],[exactly_once]) 25
- DCOUNT(database, [field],criteria) 25
- LOG(number,[base]) 26
- SUMPRODUCT(array1,array2…) 26

Week 6: Prescriptive Analytics 1 26
In class note 26
Pruning Decision Tree 26
Prescriptive Analytics 26
Optimization Theory 26
- Linear Programming 27
- Solver-Solving Methods 27

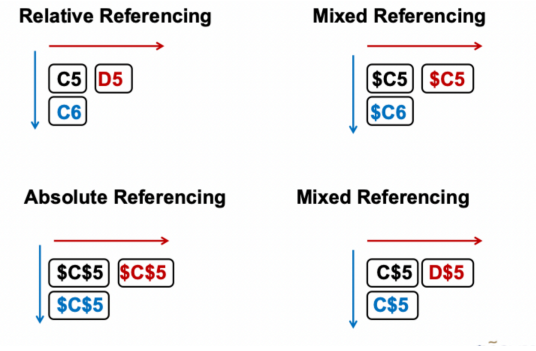
Week 7: Prescriptive Analytics 2 28
In class 28
Rand() 28
RANDBETWEEN(bottom,top) 29
Reveal/Change Door Options Matrix 29
Simulate Random Data 30
Simulating Data from Frequency Bins 30
Resampling: Discrete Distribution 31
Resampling: Continuous Distribution 31
Probability Functions 32
Inverse Method: Discrete Distribution 33
Inverse Method: Continous Distribution 33
Exponential Distribution 33
Binomial Distribution 34
Normal Distribution 35
Date & Time Management 36
Week 8: Queueing System 37
In class notes 37
Exponential Distribution 37
Observing Queues 38
Week 9 38
Week 10 - No Class 39
Week 11 - Tableau 39
Week 12 39
In class notes 39
Week 13 39
Week 14 39

Week 1: Course overview & Business modelling

Autofill

- It can be used for linear extrapolation
- Use to compute productivity and extrapolation

Referencing



Trend()

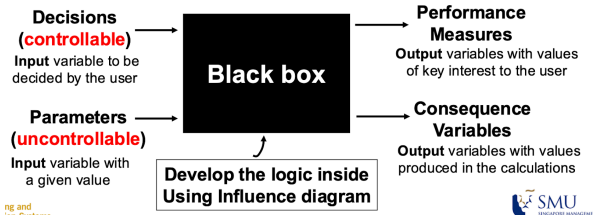
- Trend(known_y,known_x,new_x,const)
  - CONST: TRUE
    - Intercept not at zero
  - CONST: FALSE
    - The intercept is at zero
- Return a new_y value

Modelling

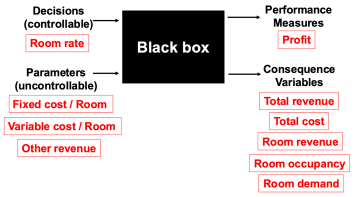
- Influence Diagrams - Formulate relationships between variables
  - Picture the connection between the model's exogenous variables
  - (known and given input variables) with the performance measure
  - (output variables)
  - Steps in building an influence diagrams

- Start with the **performance measure**
- Decompose the performance measure into 2 or more **intermediate variables** that combine mathematically to define the performance measure
- Further, decompose each intermediate variable into more intermediate variables until the i**nput parameters or decision variables** are defined

- Black Box View - Classify and Summarise all variables
  - A simple model to summarize the input variables, output variables and logic within the black box
    - Decisions (**Controllable** input)
    - Parameters (**Uncontrollable** input)
    - Performance Measures (Interested output)
    - Consequence Variables (Calculated output)



F1 Night City Race – Blackbox Model

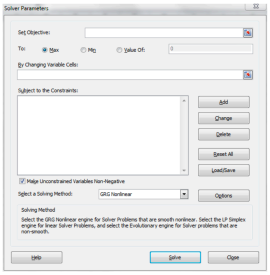


- Construct the model (Excel)
- Perform analysis and validate the model

Solver

Solver to find break-even rate

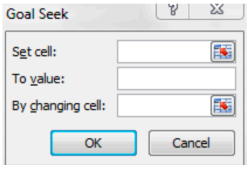
- Set Objective = Profit
- To Value of = 0
- By Changing Variable Cell = Price



Goal Seek(simplified version of solver)

Goal Seek to find break-even rate

- Set cell = Profit
- To value = 0
- By changing cell = Price



Goal Seek cannot "maximize" or "minimize". It can only change and input cell value to ensure that the target cell reaches a target value.

Week 2: Financial Modeling

- Use a linear formula to get the prediction

Scenario 1 - F1 w/o Wine & Cheese

Intercept	Slope
1.70	-0.00175

Room Rate	Occupancy
400	100%
600	65%

Scenario 2

Intercept	Slope
1.54	-0.00135

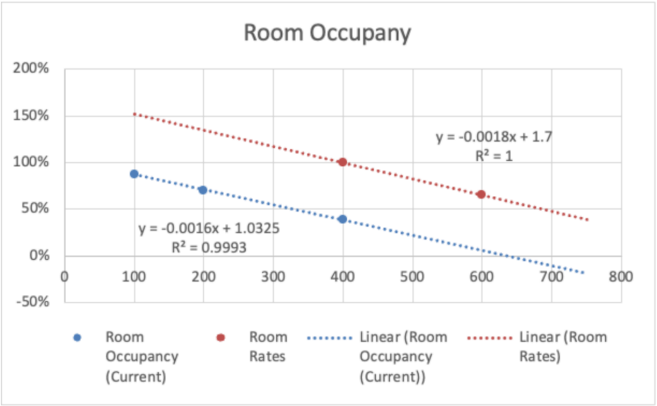
Room Rate	Occupancy
400	100%
600	73%

Scenario 3 - F1 with Wine & Cheese

Intercept	Slope
1.40	-0.00100

Room Rate	Occupancy
400	100%
600	80%

#from assum



- Financial Functions

The flow of money depends on the point of reference of the bank or your own wallet!!
They need to be in the same time frame months or years or weeks

- **Positive**
  - **Cash coming to us, coming into our pockets**
    - Example: Lump Sum $$$ borrowed for housing loan, car loan, withdrawals for expenses, etc.
- **Negative**
  - **Cash we pay out, leaving our pockets**
    - Example: Payment of monthly installments, monthly savings, balloon payments, etc.

- PMT -> periodic payment

Returns the periodic payment amount (-)

- Type: sets the payment type
  - 0 = payment due end of period (default)
  - 1 = payment due beginning of period
- Last 2 arguments are optional when their values are = 0

I want to borrow $140,000 to buy a car. The bank is willing to lend me the money for a period of 5 years at 5% per annum (monthly rest). The payment schedule is a fixed amount every month for 60 months and a "balloon" payment of $10,000 with the last payment. How much do I have to pay each month?

• rate (loan interest)	= 5% p.a. => monthly interest = 5%/12 (= i)
• nper (loan period)	= 5 years/60 months
• pv (loan amount)	= $140,000
• fv (last balloon payment)	= - $10,000

- PV -> present value

Returns the present value of an investment based on periodic, constant payments or a single future value

- When pmt is entered, fv is optional; when pmt is omitted, then fv must be entered

I want to spend 9 months away in an overseas exchange program. I would need $3,500 each month for expenses and in the last month, an additional $600 for buying gifts. At 3% per year interest, how much money (after paying for airfare etc.) should I have initially in my bank account to cover for this adventure?

• rate (interest rate)	= 3% p.a. => monthly interest = 3%/12 (= i)
• nper (time period)	= 9 months
• pmt (monthly expenses)	= $3,500
• fv (last balloon payment)	= $600
• type	= 1

- FV -> future value

❑ **Returns the future value of an investment based on periodic, constant payments or a single investment today**

- When pmt is entered, pv is optional; when pmt is omitted, then pv must be entered

I am investing $50,000 right now and will add $8,000 each year for the next 10 years. The expected return of this investment is 6.5% per yr. At the end of 10 yrs, how much money would I be expected to have?

• rate	= 6.5% p.a.
• nper (time period)	= 10 years
• pmt	= - $8,000
• pv	= - $50,000

- Rate → interest rate, investment rate, discount rate

❑ **Returns the interest rate per period of an annuity**

- RATE is calculated by iteration and can have zero or more solutions. If the successive results of RATE do not converge to within 0.0000001 after 20 iterations, RATE returns #NUM! error
- When pmt is ignored, then fv must be entered, else fv is optional
- GUESS is the initial guess for rate. If omitted, then Excel will use 10% as initial guess.

I am investing $50,000 right now and will add $8,000 each year for the next 10 years. If I want the money to grow to $250,000, what kind of expected return rate must I get for my money?

• nper	= 10 years
• pmt	= - $8,000
• pv	= - $50,000
• fv	= $250,000

- NPER -> number of periods

❑ **Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate**

❑ *Excel function allows you to enter any value using any sign (+ or -) and returns accordingly. You have to know if the results returned made sense or not:*

- NPER(12%,-100,-1000)= - 6.95 years?
- NPER(-12%,-100,1000)=NPER(-12%,100,-1000)= 6.17 years

I am investing $50,000 right now and will add $8,000 each year. If the return rate is 8% per year, how many years must I invest before the investment grows to $250,000?

• rate	= 8%
• pmt	= - $8,000
• pv	= - $50,000
• fv	= $250,000

- IRR -> internal rate of return for a series of cash flow

❑ **IRR assumes the first value in *values* is at today!**

❑ Example

- I am investing in a project with an initial $200,000 (today) and a sequence of cash flows generated is as shown below. What would the return rate need to be for the sum of the return cash flows to be equivalent to the initial investment?

Present Value = Cash Flow Value / ((1+rate)^n)
Where n = nth year

- Using the trial-and-error method, we find the present value of all values (except initial investment of $200,000) and find the rate such that NPV = 0
- When using Excel Function IRR(), include all values. IRR will assume the first value to be evaluated as net present value

- NPV -> net present value

❑ Example

- I am investing in a project with an initial $200,000 (today) and a sequence of cash flows thus generated is as shown below. If the return rate is 8% per year, what in today's value is the project worth?
  - Initial investment today = $200,000
  - Rate = 8%

Present Value = Cash Flow Value / ((1+rate)^n)
Where n = nth year

- By manual computation, we find the present value of all values (except initial investment of $200,000) and find the net present value by summing
- When using Excel Function NPV(), include all values except initial investment of $200,000, which is to be added to the result of NPV() function

- Match()
  - Returns the relative position of an item in an array that matches a specified value in a specified order

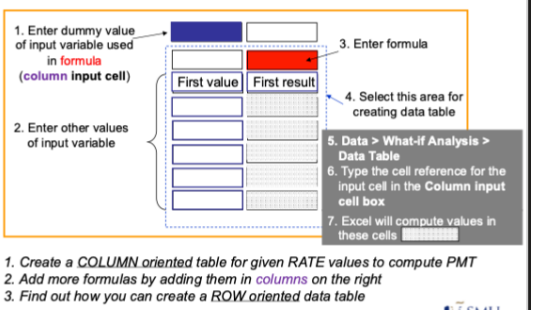
Use Match instead of lookup functions when you need position of an item in a range instead of the item itself

- If match_type is 1, MATCH finds the largest value that is less than or equal to lookup_value. Lookup_array must be placed in ascending order
- If match_type is 0, MATCH finds the first value that is exactly equal to lookup_value. Lookup_array can be in any order
- If match_type is -1, MATCH finds the smallest value that is greater than or equal to lookup_value. Lookup_array must be placed in descending order

Note: match_type 1 and -1 are approximated matches

- Data Tables
  - Useful for what-if analysis
  - A shortcut for calculating multiple versions instantaneously
  - **One-Variable Data table**

If the data table is column-oriented:

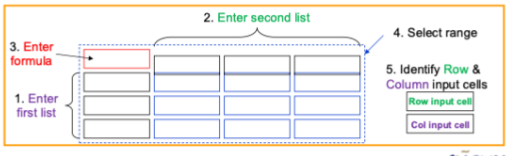


1. Create a COLUMN oriented table for given RATE values to compute PMT
2. Add more formulas by adding them in columns on the right
3. Find out how you can create a ROW oriented data table

  - **Two-Variable Data table**

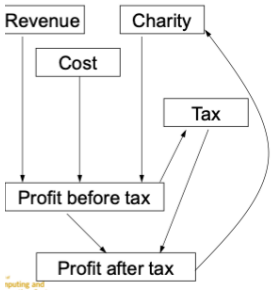
0. Prepare **dummy cells** for the 2 variables
1. Type one list of input values in the same column, below the formula
2. Type the second list in the same row, to the right of the formula.
3. In the intersection cell, enter the formula that refers to the two input cells
4. Select the range of cells that contains the formula and both the row and column of values.
5. **Data > What-if Analysis > Data Table**

5.1 In the **Row input cell** box, enter the reference to the input cell for the input values in the row
5.2 In the **Column input cell** box, enter the reference to the input cell for the input values in the column



- Circular Reference

❑ **When a formula refers back to its own cell, either directly or indirectly, it is called a circular reference**



Profit before tax =
Revenue – Cost – Charity

Profit after tax =
Profit before tax – tax

Charity = % * Profit after tax

Therefore,

Charity = % *
[(Revenue – Cost – Charity) – tax]

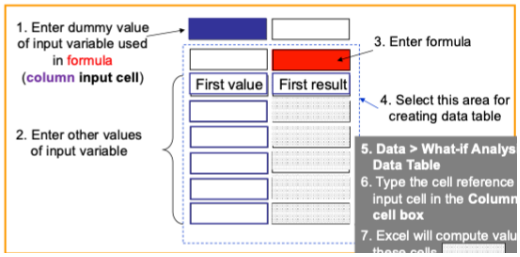
Week 3: Data Analysis & Manipulation

- Black box: Input parameters → decision →
- Inference diagram → performance measure → decompose into intermediate variables and more → arrive in input variable/dei=vision
- Sensitivity analysis -> different parameters (some parameters are from assumptions)

1. Data Tables

a. Useful for what-if analysis → shows how changing the values of input variables in your formulas affects the results of the formulas
b. Calculates multiple results in one operation without the need to bother with relative & absolute referencing
c. One-Variable Data table

If the data table is column-oriented:



d. Two-Variable Data table

2. NPV

a. Calculates the net present value of an investment by using a discount rate and a series of **future payments (negative values) and income (positive values)**
b. Rate is the discount rate applied to future payments (-) or incomes (+) to compute present worth, and must be expressed in the same terms as the values (all in months/years)
c. Each value must be equally spaced in time and occur at the end of each period
d. **If your first investment occurs at the beginning of the first period, the first value must be added to the NPV result, and not included in the values arguments.**
e. Different from **PV() function**
   i. PV allows cash flows to begin either at the end or at the beginning of the period
   ii. PV cash flow must be constant throughout the investment

❑ Example

- I am investing in a project with an initial $200,000 (today) and a sequence of cash flows thus generated is as shown below. If the return rate is 8% per year, what in today's value is the project worth?
  - Initial investment today = $200,000
  - Rate = 8%

Present Value = Cash Flow Value / ((1+rate)^n)
Where n = nth year

- By manual computation, we find the present value of all values (except initial investment of $200,000) and find the net present value by summing
- When using Excel Function NPV(), include all values except initial investment of $200,000, which is to be added to the result of NPV() function

3. IRR → internal rate of return

a. Return the internal rate of return for a series of cash flows represented by the number in values
b. IRR is the interest rate corresponding to a **zero net present value**
c. **IRR IS the Interest rate received for an investment** consisting of **payments (negative values) and income (positive values)** that occur at regular periods.
d. Cash flow need not be the same but must be equally spaced in time and occur at the end of each period
e. Must contain at least one **positive value(income)** and one **negative value(payment)**
f. IRR calculation is iterative until the result is accurate within 0.00001 percent. If IRR can't find a result that works after 20 tries, **the #NUM! error value is returned.** Add a guess value to enhance the search

❑ **IRR assumes the first value in *values* is at today!**

❑ Example

- I am investing in a project with an initial $200,000 (today) and a sequence of cash flows generated is as shown below. What would the return rate need to be for the sum of the return cash flows to be equivalent to the initial investment?

Present Value = Cash Flow Value / ((1+rate)^n)
Where n = nth year

- Using the trial-and-error method, we find the present value of all values (except initial investment of $200,000) and find the rate such that NPV = 0
- When using Excel Function IRR(), include all values. IRR will assume the first value to be evaluated as net present value

4. Basic Statistical Information

a. Univariate Analysis → analysis of a single variable x
   i. max()
   ii. min()
   iii. Central tendency
      1. average()
      2. median()
      3. mode()
   iv. Dispersion
      1. stdev() or stdev.p()
      2. var() or var.p() → **square of the standard deviation**
      3. Covariance = stdvev()/mean()
   v. Distribution
      1. Uniform distribution
      2. Normal distribution
      3. Exponential distribution
      4. Binomial distribution
      5. Poisson distribution
b. Coefficient of Variation (CV)
   i. If standard deviation measures the absolute dispersion of data, then the CV measures the relative dispersion of data
   ii. CV = stdev()/mean()
   iii. **CV is dimensionless**, it is useful for comparing distributions with different units or magnitudes
c. Bivariate Analysis
   i. For bivariate data, we are often interested in the **relationship between the two variables**
   ii. **Statistical Information** between the 2 variables ->
      1. Covariance → covar()
      2. Correlation → correl()
   iii. Potential Casualty relationships between the 2 variables
      1. Intercept → intercept()
      2. Slope → slope()
      3. R^2 → RSQ()
      4. Trend. → trend() / forecast()
   iv. CORRELATION IS NOT = CAUSALITY

5. COUNTIF()
6. COUNTIFS()
7. SUMIF()
8. SUMIFS()
9. AVERAGEIF()
10. AVERAGEIFS()
11. CONCATENATE() or &
12. Frequency()
    a. A special function that returns different results depending on how the function is executed
    b. @frequency
       i. To **return cumulative count** which is the number of items in the data_array with values <= bin value
    c. frequency() → with spill/array
       i. Return **non-comulative** count results which is the number of items in the data_array between the upper and lower bin sizes.
       ii. SPILL
          1. Select 1 cell to begin
          2. Spill to multiple cells
       iii. ARRAY
          1. Select the number of cells = number of bins to begin
          2. Work on multiple cells
13. Data Manipulation
    a. ranking()
    b. sort(array,[sort_index],[sort_order],[by_col])
       i. Sort_index → a number indicating row or column to sort by
       ii. Sort_order is 1 for ascending order , -1 for descending order
       iii. By_col is false for sort by row, true for sort by column
    c. small(array,k) → sort data in ascending order according to kth position where k means smallest
    d. large(array,k) → sort data in descending order according to kth position
    e. filter() -> data cleaning
    f. rank(number, ref) → number is the rank to determine, and ref is the selection of cells which includes number (non-numeric values in ref are ignored)

Week 4: Descriptive Analytics

Lookup Functions

- LOOKUP(lookup_value,lookup_vector,result_vector)
- VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)
  - Range_lookup = approximate or exact match
- MATCH(lookup_value,lookup_array,[match_type])
  - 1 → largest value less than or equal to value [array must be in ascending order]
  - 0 → exact match [array any order]
  - -1 → smallest value that is greater than or equal to value [array must be in descending order]
- INDEX(array,row_num,column_num)
- XLOOKUP(lookup_value,lookup_array,return_array,[if_not_found],[match_mode],[search_mode])
- RANK(number,ref,[order])
  - The number is the number whose rank you wish to determine
  - Ref is the selection of the cells which includes the number
  - Order is 0 will sort ref in descending order while any non-zero will sort ref in ascending order
- RANK(array,[sort_index],[sort_index],[by_col])
  - array is the selection of cells that you want to sort
  - sort_index is a number indicating the row or column to sort by
  - sort_order is 1 for ascending order (default), -1 for descending
  - order
  - by_col is FALSE for sort by row (default), TRUE for sort by column
- SMALL(array,k)
  - K is the index/position of the k-th smallest value in a dataset
- LARGE(array,k)
  - K is the index/position of the k-th argest value in a dataset
- FILTER(array,include,[if_empty])
  - Include is a Boolean result (True/False)
  - Is a spill function

Descriptive Analytics

- Pivot Table & Chart
- LINEST(known_y,known_x,[const],[stats])
  - returns the statistical information of a linear line fitted using least squared method for (n+1) variables, where Y is the dependent variable and there are n independent variables X
  - Y = m1X1 + m2X2 + m3X3 + ... + mnXn + b
  - known_y = range of y values
  - known_x = range of x values
  - const = TRUE/FALSE
  - TRUE or Omitted = to compute b normally
  - FALSE = to set b = 0
  - stats = TRUE/FALSE
    - TRUE = to output all regression statistics
    - FALSE or Omitted = to output only m coefficients and b
  - **provides the full regression analysis, including slope, intercept, and statistical significance, and is ideal for in-depth analysis of the relationship between two or more variables.**

Week 5: Predictive Analytics

In class notes

- Lookup → must be sorted
- Vlookup → must be the left-most to search then can specify exact or not
- Xlookup → no criteria
- Match → return index and not data
- Rmb to use data validation, index, concatenate, and filter

Lookup recap

- LOOKUP
  - Approximate search
  - Search in ascending order
- VLOOKUP
  - Approximate search
  - Look up in ascending order
  - False → exact search, lookup array in any order
- MATCH
  - 1: approximate search → ascending order
  - 0: exact search → any order
  - -1: approximate search → descending order
- XLOOKUP
  - ['if_not_found'],[match_mode],[search_mode]

Dynamic Trends

1. SUMIFS()
2. FILTER()
3. LINEST()
4. Insert > map
5. Insert > scatter
6. Form controls > check box

LINEST()

- (known_y,known_x,[const,[stats])
  - Returns the statistical information of a linear line fitted using the least squared method for (n+1) variables, where Y is the dependent variable and there n independent variables x
- Const = TRUE/FALSE
  - true/omitted = compute b normally
  - False = set b = 0
- Stats = TRUE/FALSE
  - True = to output all regression statistics
  - False or omitted = to output only m coefficients and b

	A	B	C	D	E	F
1	m_n	m_{n-1}	...	m_2	m_1	b
2	se_n	se_{n-1}	...	se_2	se_1	se_b
3	r_2	se_y				
4	F	d_f				
5	ss_{reg}	ss_{resid}				

Statistical results output for (n+1) variables

- m = coefficient of x
- b = intercept
- se = standard error for m and b
- r_2 = R-squared value
- se_y = standard error for y
- F = F statistic
- d_f = degrees of freedom
- ss_{reg} = regression sum of squares
- ss_{resid} = residual sum of squares

To execute LINEST function as SPILL
1. Select first cell
2. Enter LINEST function and make the necessary known_y, known_x selections, and set TRUE, TRUE
3. Hit Enter

- Se1,se2 → IMPORTANT
  - Standard error values for the coefficients
- R^2 → IMPORTANT
  - Coefficient of determination
  - Compares estimated and actual y-values
  - Ranges from 0 to 1

Quantitative Forecasting

- 2 main types of quantitative forecasting
  - **Casual Models**
    - Predict a future parameter as a function of others
    - Linear regression, multiple regression
    - Rely on casualty of input factors on the outcome
  - **Time series Models**
    - Predict a future parameter as a function of past values of the parameter
    - Assume past values have an impact on future values across time
- Linear regression
  - Model relationship between a dependent variable Y with an independent X
  - When there are >2 variables, we term it as multiple regression
    - relationship between the dependent variable Y and n independent variables Xn
    - The value of coefficients m1,m2,m3, and the constant b are estimated using the concepts of a best-fit line
- Multiple Linear Regression
  - As a general rule of thumb, N should at least be 30 to be able to establish a reliable relationship
  - With the established expression or relationship, onecan use it to predict the new value of Y for given values of X1,X2,X3
- Significant value that affects regression
  - In order to correctly assess which independent variable is indeed significant, we will look at the adjusted R^2 given as

$$R^2_{adjusted} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where
- N = total sample size
- p = number of predictors

  - Adjusted R-square increases only when the independent variable is significant and affects the dependent variable
  - **R2 vs Adjusted R^2**

- Sales revenue is affected by money spent on
  - X1 = YouTube, X2 = Facebook, X3 = Newspaper, X4 = Roadshow
- Based on 171 records, the R-squared and Adjusted R-squared are

Small change

Dependent Variables	R-squared	Adjusted R-squared
X1	0.61152	0.60923
X1, X2	0.90022	0.89903
X1, X2, X3	0.90048	0.89869
X1, X2, X3, X4	0.90123	0.89885

When we use X_1, X_2, and X_4, adjusted R-square is 0.89908

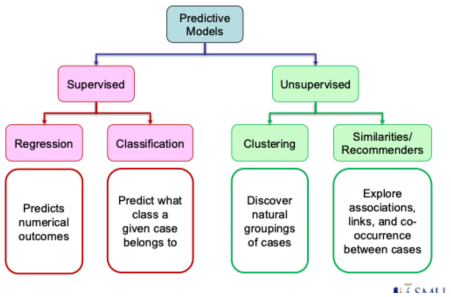
R-squared increases, even if the independent variable is insignificant

Adjusted R-squared increases, then decreases when X_3 is added

  - P-value
    - A high p-value shows that they are not significant

- Time Series Forecasting
  - Collection of observations of well-defined data items collected through REPEATED MEASUREMENTS over EQUALLY SPACED TIME INTERVALS (hourly, daily, weekly, monthly, yearly)
  - Can have one or a combination of the components
    - Trend
      - Gradual upwards/downwards movement of data over time
      - Long-term movement in a time series not related to calendar & irregular effects
    - Seasonality
      - The pattern of demand fluctuation above or below the trendline that occurs regularly
      - Short-term regular and repetitive variations which can be as long as a year or as short as a few seconds
    - Cyclical
      - Pattern in the data that occurs every several years (business cycle)
      - Has a duration of at least one year
      - Longer term thus require many years of data to determine its repetitiveness or unusual circumstances
    - error/irregularity/residual
      - Short term fluctuations in the series which are neither systematic nor predictable
      - Appear as small random ups and down and can dominate movements, which will mask the trend and seasonality
  - So usually, BEFORE FORECASTING, we will remove seasonality and cycle!!. We will add them back after forecasting to reflect seasonality or cycle
- FORECAST.ETS
  - Perform time series forecasting into the future
  - (target_data,values,timeline,[seasonality],[data_completion],[aggregation])
  - Target_data = future date which you want to predict value for
    - Can be date/time or numeric
  - Values = historical values
  - Timeline = historical dates which correspond to historical values
  - Seasonality
    - 0 = no seasonality
    - 1 = detect seasonality → default
  - Data_completion
    - When there are missing data (<30%)
    - 0 = missing data set as 0
    - 1 = missing dataset as linear interpolation of adjacent data points -> default
  - Aggregation
    - Useful for multiple data points of the same date
      - 1 = average
      - 2 = count
      - 3 = counta
      - 4 = max
      - 5 = median
      - 6 = min
      - 7 = sum
- FORECAST.ETS.CONFINT
  - Returns the confidence interval for the forecast value at the specified target date
  - (target_data,values,timeline,[**confidence_level**],[data_completion],[aggregation])
  - Confidence level
    - Numerical value between 0 and 1
    - Upper confidence bound = forecast value + confidence interval
    - Lower confidence bound = forecast value - confidence interval
    - Smaller interval would imply more confidence in the prediction for this specific point
- FORECAST.ETS.SEASONALITY
  - Return the length of the repetitive pattern that excel detects
  - (values, timeline,[data_completion],[aggregation])
  - If result = 12, the data provided is in months, then the seasonality repeat pattern is 12 months
- FORECAST.ETS.STAT
  - Return the statical values selected by the user by setting the statistic_type
  - (values,timeline,statistic_type,[seasonality],[data_completion],[aggregation])
  - Statistic_type
    - 1 = alpha
    - 2 = veta
    - 3 = gamma
      - 1 = alpha = base value parameter, a higher value gives more weight to recent data points
      - 2 = beta = value parameter, a higher value gives more weight to the recent trend
      - 3 = gamma = seasonality value parameter, a higher value gives more weight to
    - 4 = MASE -> mean absolute scaled error
    - 5 = SMAPE -> symmetric mean absolute percentage error
    - 6= MAE -> mean absolute percentage error
    - 7 = RMSE -> root mean squared error
    - 8 = step size detected

Predictive Analytics Models



Supervised Learning

- Learning where a training set of actual outcome is available

Unsupervised Learning

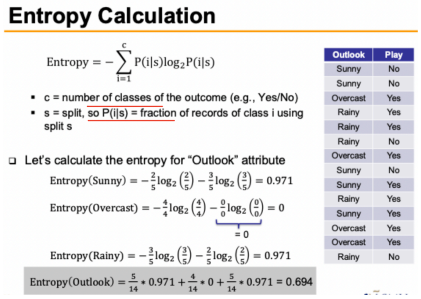
- Understand pattern without specifying purpose or target → outcome is not available

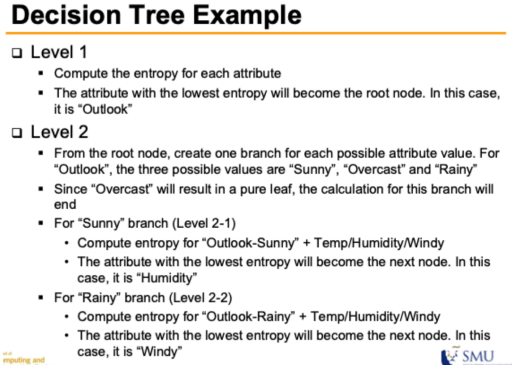
Classification → Decision Tree

- An inverted decision tree that originates with a root node at the top of the tree to interior nodes to leaves representing class values
- Each interior node corresponds to one of the input attributes
- Each leaf represents a class given the values of the input represented by the path from the root to the lead
- Decision Tree Algorithm
  - Measure with impurity value
    - The impurity value of a split is the **weighted sum of the impurity of the child nodes and the impurity of the child node is computed as the entropy**

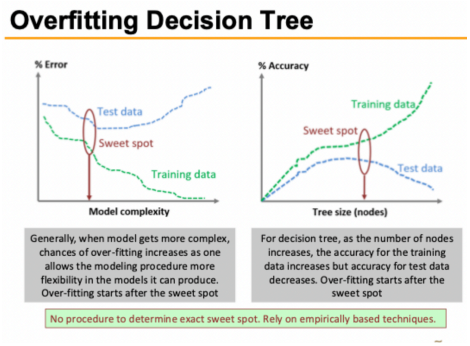
$$\text{Impurity Value} = \sum_{j=1}^{b} P(j|s) \cdot \text{Entropy}_j$$

    - B = number of branches (child_nodes)
    - S = split → P(j|s) = fraction of records of child node j using split s
    - Entropy = entropy of the child node j
      - Measure of how much information
      - Measure of chaos or disorder
      - **Lower entropy means higher information!!!!**
      - We compute the entropy of each attribute and we sum up to get the entropy





- Pruning Decision Tree



  - One way to prevent over-fitting is to use pruning
    - We need to trade off of model complexity and model performance
    - Forward pruning
      - Do not split further when the number of instances in the node is too small
    - Backward pruning
      - Build a full tree and work backwards from the leaves and apply a stats test at each stage to reduce nodes that are not statistically significant
    - Interior node pruning
      - Raise a sub-tree up one level?

Functions

- UNIQUE(array,[by_col],[exactly_once])
  - Return a list of unique values in array
  - By_col is a True/Fase
    - True → return distinct column
    - False → return distinc row
  - Exactly_once
    - True → only those occur once! will be returned
    - False → return all unique values
- DCOUNT(database, [field],criteria)
  - Counts the records that contain numbers in a field or records in a list or database that matches conditions specified in criteria
  - **Database** refers to a selection of cells including all columns and rows, and headers
  - **Field** indicated the column used in the function
    - Such as 'Age' and 'Yield' or an integer number to represent position of column
    - If omitted, counts all records in the database that matches the criteria
  - **Criteria**
    - Selection of criteria headers and criteria values
    - Headers must be spelled exactly as the headers in the database
- LOG(number,[base])
  - Return the log of a number to the based specified
  - Base default is 10
- SUMPRODUCT(array1,array2…)
  - Returns the sum of products of corresponding ranges or arrays

Week 6: Prescriptive Analytics 1

In class note

- If past data matters → put old data as a variable for multi regression equation
  - Put it as an additional variable for forecast?
- COUNTIF is a more decomposition version of DCOUNT?
- Rearrange the DCOUNT according to the row or column of the database!!!
- Simplex LP → for linear LP optimisation
- GRG Nonlinear → sometimes will be stuck in the local optimal
- Evolutionary → slower then GRG Nonlinear
- Can set max iterations and max time
- Solver will return different solution but objective value will be the same!

Pruning Decision Tree

- One way to prevent over-fitting is to use pruning
  - Forward pruning
    - Do not split further when the number of instances in the node is too small
  - Backward pruning
    - Build a full tree and then work backward from the leaves and apply a statistical test at each stage to reduce nodes that are not statistically significant
  - Interior node pruning
    - Raise a sub-tree up one level

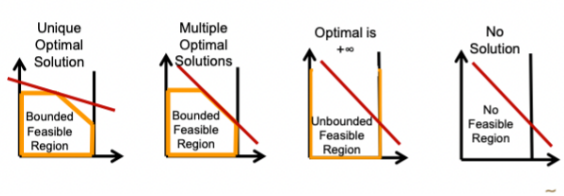
Prescriptive Analytics

- Look at multiple options and strategies and then decide on the BEST decision/course of action
- Includes methods such as experimental design, optimisation and simulation

Optimization Theory

- Determine the optimised solution for a problem which maximizes or minimizes an objectives

- The objective is usually a cost or profit calculation that represents the performance measure of the problem
- **Linear Programming**
  - Problem of minimising/maxisming a linear objective function subject to linear constraints (equality and inequality)
  - LP is made up of
    - Decision variables (real, binary, integer)
    - Objective function (minimize or maximise)
    - Constraints (<= , >= , =)
    - Non-negativity constraints
  - There are three types of Integer programming
    - Pure Integer Programming(IP)
      - The mathematical model for pure integer programming (IP) is simply the linear programming model with one additional constraint that ALL the decision variables must be integers
    - Mixed integer programming (MIP)
      - If only SOME of the decision variables must be integers, then we have Mixed integer programming (MIP)
    - Binary Integer Programming (BIP)
      - If the decision variable can only take 2 values (0 , 1) then we have Binary Integer Programming
  - 4 possible results for optimisation of LP
    - Unique optimal solution
      - This solution must be at a corner point
    - There exist multiple optimal solutions
      - Objective function line lies exactly on a constraint
    - The optimal cost is infinity
      - for minimization or maximisation
      - Feasible region is unbounded
    - There is no feasible solution
      - No feasible region



- **Solver-Solving Methods**
  - Solving methods
    - simplex
      - Most efficient method to solve linear problem, but unable to solve nonlinear problem
    - Evolutionary
      - Slower, non-smooth nonlinear, more robust since it is more likely to find a global optimum solution
      - It NEED to set lower and upper bound values for decision variables
    - GRG Nonlinear
      - Faster, smooth nonlinear, trapped at the local optimum
      - If we do not know if the problem is linear or nonlinear, just use GRG which may give different results

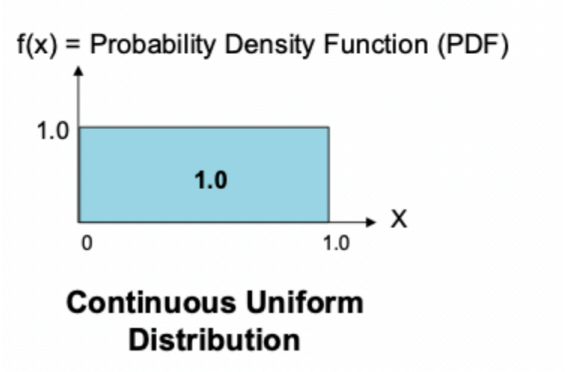
Week 7: Prescriptive Analytics 2

In class

- Solver Constraints convention
  - The left-hand side is a calculation
  - The right-hand side is the parameters
- Figure out the constraints of the business context!!
  - What constraint do we need to fulfill
- Class part → paste special -> value only
- Randbetween → discrete distribution function
- Rand → continuous distribution
- Inverse method
  - Given a distribution then we simulate a random position with the rand function what is the corresponding random number
- The larger the lambda the more often the event happens

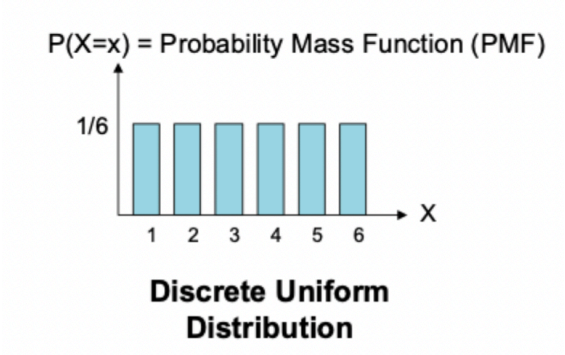
Rand()

- Returns a uniformly distributed random real (continuous) number greater than or equal to 0 or less than 1
- A new random real number is returned every time the worksheet is calculated
- Represents a continuous uniform distribution where each number is equal likely to occur
- The number generated is usually used as the cumulative probability
- To generate a real (continuous) number between numbers A AND B
  - RAND()*(B-A) + A
  - E.g 1 & 9
- To generate a random number among 2 non-contiguous numbers A and C
  - if(rand()<0.5,A, C)
  - E.g 1 & 3



RANDBETWEEN(bottom,top)

- Returns a random integer number defined between bottom (inclusive) and top (inclusive)
- Represents a discrete uniform distribution where each number is equally likely to occur
- A new random integer number is returned every time the worksheet is calculated
- The number generated is usually used as a position number



Reveal/Change Door Options Matrix

- The table allows us to determine which door to open and reveal given prior knowledge of the prize door and selected door

		Selected Door		
		1	2	3
Prize Door	1	RB(2,3)	3	2
	2	3	If(RAND()>0.5,1,3)	1
	3	2	1	RB(1,2)

- Revealed door = index(array, prize door , selected door)

Simulate Random Data

Simulating Data from Frequency Bins

- Build CRF table and use RAND() to match CRF to get new X
  - Frequency count
  - Percentage of occurrence
  - Distribution functions

❑ Cumulative Relative Frequency Table

	B	D
2	# Children	CumRF
3		0.000
4	0	0.093
5	1	0.283
6	2	0.704
7	3	0.927
8	4	1.000

Down-shifted Lookup() table

=LOOKUP(RAND(), D3:D7, B4:B8)

RAND() = 0.500 will lie between 0.283 and 0.704.
LOOKUP() will look for the largest value in the lookup_vector (D3:D7) that is smaller than 0.500, which is still the value 0.283. It will return me the corresponding value 2, which is the correct result.

Building CRF Table

❑ **(i) In the census example, we build the CRF table using sample data**

❑ **(ii) Build CRF table using percentage of occurrence**

- E.g., Arrivals of kids to a toy store. 10% of the minutes no child arrive, in 20% of the minutes 1 child arrive, in 30% of the minutes 2 children arrive, in 30% of the minutes 3 children arrive, and in 10% of the minutes 4 children arrive

# of Arrivals	Probability	CumProb
0	0.1	0.0
1	0.2	0.1
2	0.3	0.3
3	0.3	0.6
4	0.1	0.9
		1.0

❑ **(iii) Build CRF table using known distribution**

- Normal, Binomial, Exponential, Poisson, etc.

Resampling: Discrete Distribution

- RANDBETWEEN(1,N) to generate a random integer number from 1 to the number of raw data points N, say K.
- This number is used to return the Kth value in the raw data collection as if the raw data is already sorted in ascending order

X' = **SMALL**(array, k)
X' = **SMALL**(raw data, **RANDBETWEEN**(1,N))

Sorted Data	Order
100	1
200	2
200	3
500	4

For RANDBETWEEN(1,4)=2, the resampled X' will be 200

Resampling: Continuous Distribution

- RAND() to generate a random number between 0 and 1.0, to represent the percentile value K. This percentile value is used to return the corresponding percentile number in the raw data collection

X' = **PERCENTILE**(array, k)
X' = **PERCENTILE**(raw data, **RAND**())

- PERCENTILE() sorts and interpolates among the raw data using the number returned by RAND()
- The new data X generated may be different from the raw data due to interpolation

For a percentile value (k) provided by the RAND(), it is rather unlikely that the RAND() value will coincide with the actual percentile value of the raw data

Sorted Data	Actual Percentile
1.0	0.00
3.0	0.33
5.0	0.66
7.0	1.00

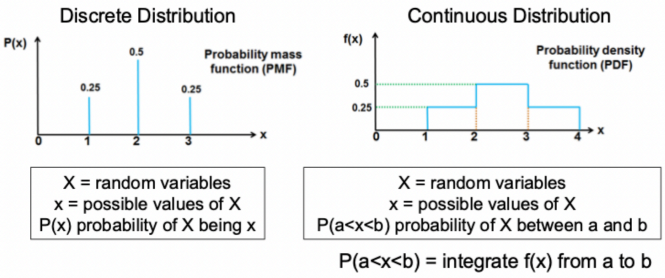
For RAND()=0.5, the resampled X' will be between 3.0 and 5.0

Computing X' by interpolation

(0.5 – 0.33)/(0.66 – 0.33) = (X' – 3.0) / (5.0 – 3.0)

Probability Functions

- Random variables (r.v) are either discrete or continuous
  - Discrete: Uniform & Binomial
  - Continuous: Uniform, Exponential, Normal

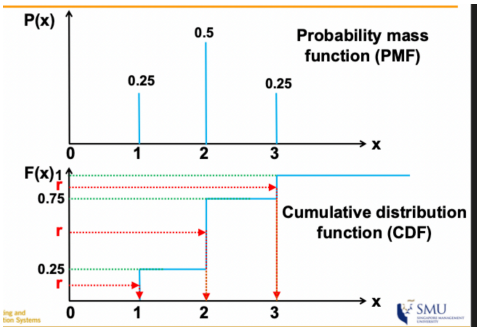


X = random variables
x = possible values of X
P(x) probability of X being x

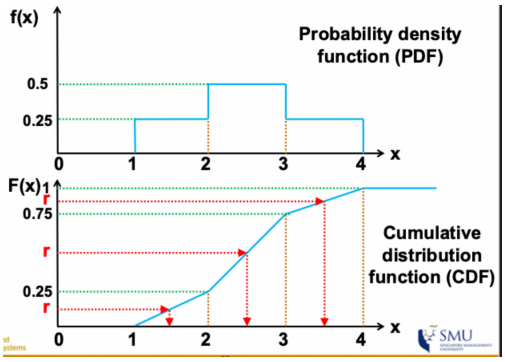
X = random variables
x = possible values of X
P(a<x<b) probability of X between a and b

P(a<x<b) = integrate f(x) from a to b

Inverse Method: Discrete Distribution



Inverse Method: Continous Distribution

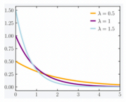


Exponential Distribution

- λ > 0 is a parameter of the distribution (arrival rate)

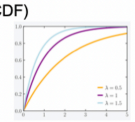
- The probability density function (PDF)

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



- The cumulative distribution function (CDF)

$$F_X(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



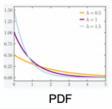
- Expectation and variance

$$E(X) = \frac{1}{\lambda} \qquad Var(X) = \frac{1}{\lambda^2}$$

SMU

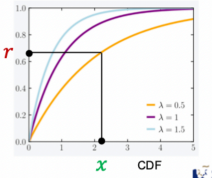
Inverse Method for Exponential Dist.

- A random position r from 0 to 1
- Find the corresponding x on the CDF
- CDF $F_X(x) = 1 - e^{-\lambda x}$
- Solve the equation $r = F_X(x) = 1 - e^{-\lambda x}$



$$e^{-\lambda x} = 1 - r$$
$$-\lambda x = \ln(1 - r)$$
$$x = -\frac{\ln(1 - r)}{\lambda}$$

- Ln(1-Rand()) / Lambda
- Average * LN (1-RAND())
- Average * LN (RAND())



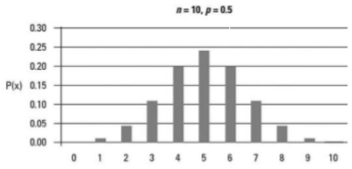
SMU

Binomial Distribution

- Binomial r.x X is best described as counting the number of success in n scenario.
- If the probability of head is p and the probability of tail is 1-p,

$$P(X = k) = C(n,k)\, p^k (1-p)^{n-k} \quad k=0,1,2,\ldots,n$$

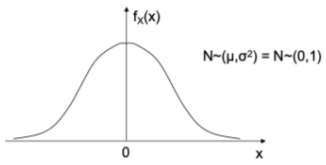
Binomial is represented as Bin(n,p)



- BINOM.DIST(no. Of trials,probability_s,cumulative)
  - Cumulative
    - True returns CDF
    - False return PMF
- BINOM.INV(trials,probability_s,RAND())
  - Generates a random percentile position from 0 to 1
  - Return a random value following binomial distribution with specific trials and proability_s

Normal Distribution

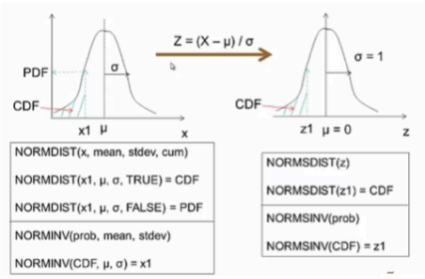
- Normal r.v X is best described as the natural, random occurrence of errors
- The normal distribution can be standardized to the standard form with mean 0 and variance 1



Standard Normal: Z = (x – μ)/σ

- NORM.DIST(x,mean,SD,cumulative)
  - True return CDF
  - False return PDF
- X' = NORM.INV(RAND(),mean,SD)
  - Return a random value following normal distribution with the specific mean and standard deviation
- NORM.DIST(z) = standard normal
  - Return only CDF
- Z' = NORM.INV(RAND())
  - Return a random value following normal distribution with mean 0 and standard deviation 1

Normal Distribution



Date & Time Management

- 2 Date systems in Excel
  - 1900 date system (Default)
  - 1904 date system
- Today()
- Year(serial_number)
- Month()
- DAY()
- DATE(year,month,day)

Subtracting

- WRONG: 14-Jan-05 – 23-Sep-04
- OK: "14-Jan-05" – "23-Sep-04" = 113
- OK: DATE(2005,1,14) – DATE(2004,9,23) = 113

- NOW()
- HOUR()
- MINUTE()
- SECOND()

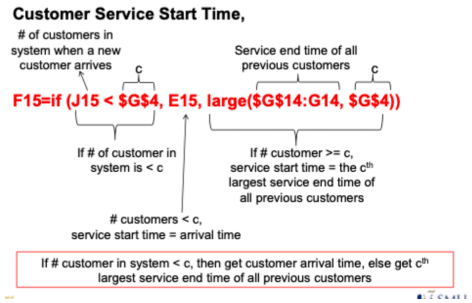
Code	Format description
d	Day as 1, 2, …, 31
dd	Day as 01, 02, …, 31
ddd	Day of week as Sun, Mon, …, Sat
dddd	Day of week as Sunday, Monday, …, Saturday

Note the inconsistency

Code	Format description
m	Months as 1, 2, …,12
mm	Months as 01, 02, …, 12
mmm	Months as Jan, Feb, …, Dec
mmmm	Months as January, …, December

Code	Format description
yy	Year in 2 digits (e.g., 08)
yyyy	Year in 4 digits (e.g., 2008)

Code	Format description
[h]	Hours as 0, 1, …, 23, 24, 25, …
h	Hours as 0, 1, …, 23
hh	Hours as 00, 01, …, 23
m	Minutes as 0, 1, …, 59
mm	Minutes as 00, 01, …, 59
s	Seconds as a 0, 1, …, 59
ss	Seconds as 00, 01, …, 59
AM/PM am/pm	Time as in a 12-hour clock



Week 8: Queueing System

In class notes

- Higher lambda then lower average → interval decreases

Exponential Distribution

- Exponential r.v X is best described as the time interval between random events, given the expected value 1/λ

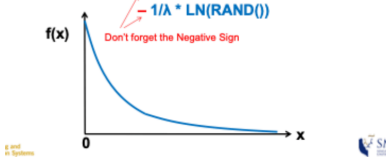
  Probability density function (PDF) $f_x(x) = \lambda e^{-\lambda x}$

  Cumulative distribution function (CDF) $F_x(x) = 1- e^{-\lambda x}$

  1/λ = average (mean) inter-arrival time; λ = arrival rate

  A random value = – 1/λ * LN(1-RAND()) or

  – 1/λ * LN(RAND())



- Inter-arrival time = -average * LN(1-RAND())
- Service Time (simulateed) = PERCENTILE(service time array, RAND())
- When there is more than 1 server!!!

Observing Queues

- Recording Arrivals
  - Timer
    - Designed for Queue system observation and analysis
    - Tabulates intermediate variables (like inter-arrival time, service time, wait time, and system time (service + wait)
    - Records arrival, service-start, and service-end times
    - Limitations of Timer
      - First come, first serve
      - Single server
  - Clicker
    - Adaptation of Timer
    - Counts 'arrivals'
      - Record arrival times of up to 3 types of customer
      - Tabulates their cumulative frequency counts for given time intervals
    - Example
      - No. of vehicles using a stretch of road
  - Others include
    - Balking
    - Reneging
    - Retrail
    - Priority

Week 9

Week 10 - No Class

Week 11 - Tableau

Week 12

In class notes

- it is not exact!!
- RMB to give an upper bound!!!
  - For inverse norm
- High IRR and High NPV
- Sort is spill function
-

Week 13

Week 14