

Statistics and Visualization Techniques for Verification and Validation

Dr. Hamdi Kavak Computational and Data Sciences Department



http://www.hamdikavak.com hkavak@gmu.edu



Soon





Original photo: https://knowyourmeme.com/photos/1326438-ben-swolo



Soon



MASON UNIVERSITY

Original photo: https://knowyourmeme.com/photos/1326438-ben-swolo



Next week (SWA)

- Short writing assignment (SWA 1) due at noon (12 pm), Monday (Sep 18).
 - http://hamdikavak.com/course-v-and-v/docs/content/05runtime-verification/
 - Choose one of the papers and submit your document on Blackboard.

Assigned Reading (choose one for your SWA 1):

- Ahrens, J., Heitmann, K., Petersen, M., Woodring, J., Williams, S., Fasel, P., ... & Geveci, B. (2010).
 Verifying scientific simulations via comparative and quantitative visualization. IEEE Computer
 Graphics and Applications, 30(6), 16-28.
- Feldkamp, N., Bergmann, S., & Strassburger, S. (2020, January). Visualization and Interaction for Knowledge Discovery in Simulation Data. In Proceedings of the 53rd Hawaii International Conference on System Sciences.





Next week (final project proposal)

- Final project proposal due at 11:59 pm, Monday (Sep 18).
- Check what is of expected
 - <u>http://hamdikavak.com/course-v-and-v/docs/assignments/03final-project/</u>
- Feel free to ask questions after class or schedule a virtual meeting.







Paper presentation assignment

- See: Blackboard
 - -> Assignments
 - -> Paper Presentations (2 per student)
 - -> Paper presentation assignment







Short recap of V&V terminology

Verification: have we built the model right?

Validation: have we built the right model?

Cook, D. A., & Skinner, J. M. (2005). How to perform credible verification, validation, and accreditation for modeling and simulation. *The Journal of Defense Software Engineering*, *18*(5), 20-24.





Objective of this lecture

- Getting familiar with some basic statistical and visualization techniques for verification and validation.
- Topics include
 - Uncertainty and random number generators
 - Confidence intervals
 - Visual inspection of model data







Python packages used in this lecture

- NumPy
- Matplotlib
- Scipy







• uncertain: "not known or definite" (Oxford Dictionary)





- uncertain: "not known or definite" (Oxford Dictionary)
- Many real-world systems contain uncertainty
 - E.g.: traffic, queues, weather, voting, package delivery, disease spread,...







- uncertain: "not known or definite" (Oxford Dictionary)
- Many real-world systems contain uncertainty
 - E.g.: traffic, queues, weather, voting, package delivery, disease spread,...
- Can be of varying degrees







- uncertain: "not known or definite" (Oxford Dictionary)
- Many real-world systems contain uncertainty
 - E.g.: traffic, queues, weather, voting, package delivery, disease spread,...
- Can be of varying degrees
- If one or more components of a system has uncertainty, we call such systems "**stochastic**."





Social Complexity

- uncertain: "not known or definite" (Oxford Dictionary)
- Many real-world systems contain uncertainty
 - E.g.: traffic, queues, weather, voting, package delivery, disease spread,...
- Can be of varying degrees
- If one or more components of a system has uncertainty, we call such systems "stochastic."
- We use a **probabilistic framework** to model a stochastic system and call the model as a **stochastic model**.





Some sources of uncertainty



Imperfect knowledge

(e.g., small sample, resolution)



Changes in the environment

(e.g., weather, human decisions)



Time dependency

(e.g., different traffic patterns at different times, store visits)



Presence of noise

(e.g., measurement precision or accuracy)



Failure

(e.g., power outage, defect)

Social Complexity



CSI 709/CSS 739 - Verification and Validation of Models — $\ensuremath{\mathbb{C}}$ Dr. Hamdi Kavak

Deterministic vs. stochastic models

Deterministic

- For the same input, each model run gives the same result.
- Assumes no uncertainty.

Stochastic

- For the same input, each model run gives a different result.
- Assumes uncertainty.
- We can still reproduce the same results if we want.





Random variables

- Stochastic systems contain components that are modeled by random variables.
- Random variables (e.g., X) are used to quantify the outcome of stochastic processes.
- We can technically call a stochastic model a "random variable"
- Two main types
 - Discrete (i.e., countable number of possible values)
 - E.g.: 5, 16, 8, 210, ...
 - Continuous (i.e., cannot count the number of possible values)
 - E.g.: 4.5234, 99.12, 3.14, 7.78, ...







Stochastic models

- Most stochastic systems we deal do not have a closed form solution
- We approximate them by developing stochastic models and running multiple times. A.k.a.?



Do you like Solitaire?

What is the chance of winning the solitaire game before seeing the cards?



Source: https://www.flickr.com/photos/bobb/43826727





Do you like Solitaire?

What is the chance of winning the solitaire game before seeing the cards?



Source: https://www.flickr.com/photos/bobb/43826727

After playing 100,000,000 games, it is found that only 8.7% of cases win*.



*According to "Bill's Solitaire Tester" using the deal 3 cards and 3 times around the deck option



Strategy used in the solitaire game

The Game The Logic The Game The Logic Image: Logic Description Occurs Used Occurs Used Image: Logic Description 1 V Deck Acet to Sut Pile 1 1 144654 1 1 1 2 V Table Acet to Sut Pile 0 0 134426 123465 2 2 Image: Logic Description Move selected 3 V Deal Deck fron Deck card value 1 1 1000082 93854 4 4 1 1 1000082 93854 4 4 1 1 1000082 93854 4 4 1 1 1000082 93854 4 4 1 1 1000082 93854 7 8 1 1 1	File	olitaire											
Used? Logic Description Occurs Used Occurs Used Index Priority Break? 1 V Deck Ace to Sut Pile 1 1 144654 144654 14 1 2 V Table Ace to Sut Pile 0 0 194245 124455 2 2 1 3 V Deal Deck for Deck card visble 1 1 31095 31094 3 3 1 4 V Tum over hidden table card 1 1 1000082 93223 76823 12 5 1 6 1 1 1000082 93257 6 7 1 1 175713 2971 6 7 1 1 1 175713 2971 6 7 1	The G	ame The	Logic										
Used? Logic Description Occurs Used Index Priority Break? 1 1 14654 144654 1 1 1 1 14654 1													
Image: Control of the card to Sut Pile 1 1 144654 144654 1 1 1 2 Image: Control of the card to Sut Pile 0 0 136426 124485 2 2 1 3 Image: Control of the card to Sut Pile 0 0 136426 124485 2 2 1 4 Image: Control of the card to suble card 1 1 1000082 983654 4 4 1 1 1000082 983654 4 4 1 1 1 1000082 983654 4 4 1			Used?	Logic Description	Occurs	Used	Occurs	Used	Index	Priority 🔺	Break?	-	
2 V Table Ace to Sut Pile 0 0 136426 122485 2 2 1 3 V Deal Dack f no Dack card visible 1 1 31095 31094 3 3 1 4 V Tum over hidden table card 1 1 1000082 93654 4 4 1 1 5 V Table king to empty space 0 0 93223 78823 12 5 1 6 7 1 1 175713 72971 6 7 1 1 1 175713 72971 6 7 1	•	1	V	Deck Ace to Suit Pile	1	1	144654	144654	1	1		ſ	
3 Image: Deal Deck if no Deck card visible 1 1 31095 31094 3 3 1 4 Image: Transport Nidden table card 1 1 1000082 933654 4 4 1 5 Image: Table king to empty space 0 0 93223 76823 12 5 1 6 Image: Table card column with Nidden card to visible card on another column 2 2 1527112 836068 10 6 1 1 177713 72971 6 7 1 6 7 8 1 1 177713 72971 6 7 8 1 1 177713 72971 6 7 8 1 9 1 7 1376530 833562 7 8 1 9 1 1 177713 72971 6 11 7 1376530 833562 7 8 1 1 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 <td< td=""><td></td><td>2</td><td>~</td><td>Table Ace to Suit Pile</td><td>0</td><td>0</td><td>136426</td><td>123485</td><td>2</td><td>2</td><td></td><td></td><td>Move selected item up</td></td<>		2	~	Table Ace to Suit Pile	0	0	136426	123485	2	2			Move selected item up
4 V Turn over hidden table card 1 1 1000082 983654 4 4 - 5 V Table king to empty space 0 0 93223 76823 12 5 - <td></td> <td>3</td> <td>V</td> <td>Deal Deck if no Deck card visible</td> <td>1</td> <td>1</td> <td>31095</td> <td>31094</td> <td>3</td> <td>3</td> <td></td> <td></td> <td></td>		3	V	Deal Deck if no Deck card visible	1	1	31095	31094	3	3			
5 Image: Table king to empty space 0 0 93223 76823 12 5 6 Image: Table card column with hidden card to visble card on another column 2 2 1527112 836068 10 6 7 Image: Dock King to table 1 1 175713 72971 6 7 Image: Table card to table 7 8 Image: Table card to table 11 1 175713 72971 6 7 Image: Table card to table 7 Image: Table card to table 7 8 Image: Table card to table 11 1 175713 72971 6 7 Image: Table card to table 7 Image: Table card to table 7 Image: Table card to table card to table 11 1 175713 72971 6 7 Image: Table card to table card to table 11 1 175713 72971 6 7 Image: Table card to table card can tum over 0 0 30378 0 18 13 Image: Table card to table card to another column, no king move needed 6 0 1651799 2288 15 18 Image: Table card to table card to another column, no king move needed 6 0 1651799 2288 15 18 Image: Table card to table card to another column, no king move needed 6 0 1651799 2288 15 18 Image: Table card to table card to tanot move		4	V	Tum over hidden table card	1	1	1000082	983654	4	4			
6 Image: card column with hidden card to visible card on another column 2 2 1527112 836068 10 6 Image: card column with hidden card to visible card on another column 1 1 175713 72971 6 7 Image: card column with hidden card to visible card on another column 1 1 175713 72971 6 7 Image: card column with hidden card to table 1 1 175713 72971 6 7 Image: card column with hidden card to table 1 1 175713 72971 6 7 Image: card column with hidden card to table 1 1 175713 72971 6 7 Image: card column with hidden card to table 0 0 1212867 516881 11 9 Image: card column voit bit file 0 0 2124941 63744 17 10 Image: card column voit file 0 0 340920 116707 5 12 Image: card column voit file 0 0 30378 0 18 13 Image: card column voit file 0 0 30378 14 17 Image: card column voit file 0 0 18932 193		5	V	Table king to empty space	0	0	93223	76823	12	5			
7 V Deck King to table 1 1 175713 72971 6 7 8 V Non Ace or King Deck card to table 11 7 1376830 833562 7 8 9 V Non-Ace table card to Suit Pile: no king move considerations 0 0 1212867 516881 11 9 1 10 V Split columns to allow a move to Suit Pile 0 0 214941 63744 17 10 1 11 V Non-Ace table card, no hidden card, to Suit Pile, king move not needed 0 0 306327 75078 16 11 <td< td=""><td></td><td>6</td><td>V</td><td>Table card column with hidden card to visible card on another column</td><td>2</td><td>2</td><td>1527112</td><td>836068</td><td>10</td><td>6</td><td></td><td></td></td<>		6	V	Table card column with hidden card to visible card on another column	2	2	1527112	836068	10	6			
8 Image: Non Ace or King Deck card to table 11 7 1376830 833562 7 8 Image: Reload Defilies 9 Non-Ace table card to Suit Pile: no king move considerations 0 0 1212867 516881 11 9 Image: Reload Defilies 10 Image: Split columns to allow a move to Suit Pile 0 0 214941 63744 17 10 Image: Reload Defilies 11 Image: Non-Ace table card, no hidden card, to Suit Pile 0 0 306327 75078 16 11 Image: Reload Defilies 12 Non-Ace Deck card to Suit Pile 0 0 306327 75078 16 11 Image: Reload Defilies 13 Image: Reload Defilies 0 0 306327 75078 16 11 Image: Reload Defilies 14 V Non-Ace Deck card to Suit Pile 0 0 30378 0 18 13 Image: Reload Defilies Image: Reload Defilies		7	V	Deck King to table	1	1	175713	72971	6	7			
9 V Non-Ace table card to Suit Pile: no king move considerations 0 0 1212867 516881 11 9 1 10 V Split columns to allow a move to Suit Pile 0 0 214941 63744 17 10 1 11 V Non-Ace single table card, no hidden card, to Suit Pile, king move not needed 0 0 306327 75078 16 11 1 12 V Non-Ace Deck card to Suit Pile 0 0 3040920 116707 5 12 1 13 V Table card to Suit Pile to allow Deck card to go to Suit Pile 0 0 30378 0 18 13 1 14 V Suit Pile card to table so table card can move 0 0 18992 1993 9 15 1 16 V Deal Deck if Deck card already visible 28 16 4814869 1633564 13 16 1 17 V Rip Deck 2 2 529369 196383 14 17 1 18 V Table card with no hidden card to another column, no king move needed 6 0 1651799 2288 15 18 V 18 V Table card with no hidden card to another column, no king move needed 6 0 1651799 2288 15 18 V 18 V Table card with no hidden card to another column, no king move needed 6 0 1651799 2288 15 18 V 18 V </td <td></td> <td>8</td> <td>V</td> <td>Non Ace or King Deck card to table</td> <td>11</td> <td>7</td> <td>1376830</td> <td>833562</td> <td>7</td> <td>8</td> <td></td> <td>=</td> <td></td>		8	V	Non Ace or King Deck card to table	11	7	1376830	833562	7	8		=	
10 V Split columns to allow a move to Suit Pile 0 0 214941 63744 17 10 11 11 V Non-Ace single table card, no hidden card, to Suit Pile, king move not needed 0 0 306327 75078 16 11 1		9	V	Non-Ace table card to Suit Pile: no king move considerations	0	0	1212867	516881	11	9			Logic
11 Image: Non-Ace single table card, no hidden card, to Suit Pile, king move not needed 0 0 306327 75078 16 11 12 Image: Non-Ace Deck card to Suit Pile 0 0 340920 116707 5 12 13 Image: Table card to Suit Pile 0 0 30378 0 18 13 14 Image: Suit Pile card to table so Deck card can move 0 0 58283 7609 8 14 15 Image: Suit Pile card to table so table card can tum over 0 0 18992 1993 9 15 16 Image: Deck card already visible 28 16 4814869 1633564 13 16 17 Image: Pile card with no hidden card to another column, no king move needed 6 0 1651799 2228 15 18 18 Image: Pile card with no hidden card to another column, no king move needed 6 0 1651799 2228 15 18 Move from pile with Sight mouse click to change cell color Image: Pile card so the same time Image: Pile card so the same time Save this logic Image: Reset Image: Pile card so the same time Image: Pile card so the same time Image: Pile card so the same time Save this logic		10	V	Split columns to allow a move to Suit Pile	0	0	214941	63744	17	10			
12 Image: Non-Ace Deck card to Suit Pile 0 0 340920 116707 5 12 13 Image: Table card to Suit Pile to allow Deck card to go to Suit Pile 0 0 30378 0 18 13 14 Image: Suit Pile card to table so Deck card can move 0 0 58283 7609 8 14 14 15 Image: Suit Pile card to table so table card can tum over 0 0 18992 1993 9 15 16 16 Image: Deck if Deck card already visible 28 16 4814869 1633564 13 16 16 17 Image: Pilp Deck 2 2 529369 196383 14 17 Image: Pilp Deck 18 17 Image: Pilp Deck 2 2 529369 196383 14 17 Image: Pilp Deck		11	V	Non-Ace single table card, no hidden card, to Suit Pile, king move not needed	0	0	306327	75078	16	11			
13 Image: Table card to Suit Pile to allow Deck card to go to Suit Pile 0 0 30378 0 18 13 14 Image: Suit Pile card to table so Deck card can move 0 0 58283 7609 8 14 14 14 15 14 15 14 15 15 16 15 16 15 16 15 16 17 18 Table card with no hidden card to another column, no king move needed 6 0 1651799 22.88 15 18 16 Move selected it 18 Table card with no hidden card to another column, no king move needed 6 0 1651799 22.88 15 18 17 Move selected it 18 17 18 17 18 17 18 17 18 16 18 17 18 17 18 18 17 18 18 17 18		12	V	Non-Ace Deck card to Suit Pile	0	0	340920	116707	5	12			
14 Image: Solid Pile card to table so Deck card can move 0 0 58283 7609 8 14 15 Image: Solid Pile card to table so table card can tum over 0 0 18992 1993 9 15 16 Image: Deck card already visible 28 16 4814869 1633564 13 16 17 Image: File Deck 2 2 529369 196383 14 17 18 Image: Table card with no hidden card to another column, no king move needed 6 0 1651799 22288 15 18 Image: File Deck Image: Move from pile with file card with no hidden card to another column, no king move needed 6 0 1651799 22288 15 18 Image: File Deck Image: Move from pile with file card with no hidden card to another column, no king move needed 6 0 1651799 2288 15 18 Image: File Deck Image: Move from pile with file card sold color Image: File Deck Image: File Dec		13	V	Table card to Suit Pile to allow Deck card to go to Suit Pile	0	0	30378	0	18	13			
15 Image: Solid Pile card to table so table card can turn over 0 0 18992 1993 9 15 16 16 Image: Deck if Deck card already visible 28 16 4814869 1633564 13 16 16 16 17 16 17 16 17 16 17 18 17 18 17 18 14 17 17 18 18 18 16 18 16 17 18 18 18 18 16 17 18 17 18 17 18 17 18 17 18 17 18 17 18 17 18 17 18 17 18 16 17 18 16 17 18 17 18 17 18 17 18 16 17 18 18 16 17 18 17 18 17 18 18 17 18 18 17 18 17 18 17 17 18 18 18 17 18 18 <td< td=""><td></td><td>14</td><td>V</td><td>Suit Pile card to table so Deck card can move</td><td>0</td><td>0</td><td>58283</td><td>7609</td><td>8</td><td>14</td><td></td><td></td><td></td></td<>		14	V	Suit Pile card to table so Deck card can move	0	0	58283	7609	8	14			
16 Image: Colors 28 16 4814869 1633564 13 16 Image: Colors Move from pile with change cell color 2 2 529369 196383 14 17 Image: Colors Image: Colors Image: Colors Only play games with Aces showing at deal Image: Colors Only preak if 2 or more of the same time Save this logic Image: Ima		15	V	Suit Pile card to table so table card can turn over	0	0	18992	1993	9	15			
17 Image: Second se		16	V	Deal Deck if Deck card already visible	28	16	4814869	1633564	13	16		1	
18 Image: Table card with no hidden card to another column, no king move needed 6 0 1651799 22288 15 18 Image: Table card with no hidden card to another column, no king move needed 6 0 1651799 22288 15 18 Image: Table card with no hidden card to another column, no king move needed 6 0 1651799 22288 15 18 Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another column, no king move needed Image: Table card with no hidden card to another card to another card to another card with no hidden card to another card to an		17	V	Flip Deck	2	2	529369	196383	14	17			Move selected item down
 Colors Move from pile with Most or Least cards Colors Colors		18	V	Table card with no hidden card to another column, no king move needed	6	0	1651799	22288	15	18		-	
Colors Only play games with Aces showing at deal Only break if 2 or more of the same time Only break if 2 or more of the same time Colors Only break if 2 or more of the same	•			III							•		
Move from pile with Change cell color Image cell color Image real color Image cell color Image real color Image cell color Image real color Image cell color Image cell color			4 41	Colors Only play games with Aces showin	ng at deal								
Colors	Mo	Move from pile with change cell color Yes 0						Only brea	Only break if 2 or more of the same			Save this logic to file	
CODIS	6	Reset							Filonity of	cour at the san	ie une	[Load Logic from file
				Colors									

Source: http://www.billturnbull.com/solitaire/Logic.html





Strategy used in the solitaire game

Game The	Logic									
	Used?	Logic Description	Occurs	Used	Occurs	Used	Index	Priority 🔺	Break?]
• 1		Deck Ace to Suit Pile	1	1	144654	144654	1	1		
2		Table Ace to Suit Pile	0	0	136426	123485	2	2		Move selected item up
3	V	Deal Deck if no Deck card visible	1	1	31095	31094	3	3		
4	V	Tum over hidden table card	1	1	1000082	983654	4	4		
5	V	Table king to empty space	0	0	93223	76823	12	5		
6	V	Table card column with hidden card to visible card on another column	2	2	1527112	836068	10	6		
7	V	Deck King to table	1	1	175713	72971	6	7		
8	V	Non Ace or King Deck card to table	11	7	1376830	833562	7	8		
9	V	Non-Ace table card to Suit Pile: no king move considerations	0	0	1212867	516881	11	9		Reload Default Logic
10	V	Split columns to allow a move to Suit Pile	0	0	214941	63744	17	10		
11	V	Non-Ace single table card, no hidden card, to Suit Pile, king move not needed	0	0	306327	75078	16	11		
12	V	Non-Ace Deck card to Suit Pile	0	0	340920	116707	5	12		
13	V	Table card to Suit Pile to allow Deck card to go to Suit Pile	0	0	30378	0	18	13		
14	V	Suit Pile card to table so Deck card can move	0	0	58283	7609	8	14		
15	V	Suit Pile card to table so table card can turn over	0	0	18992	1993	9	15		
16	V	Deal Deck if Deck card already visible	28	16	4814869	1633564	13	16		
17	V	Flip Deck	2	2	529369	196383	14	17		Move selected item down
18	V	Table card with no hidden card to another column, no king move needed	6	0	1651799	22288	15	18		
18 Move from pi	ile with	Table card with no hidden card to another column, no king move needed	6 ng at deal	0	1651799	22288	15 Only bre Priority o	18 tak if 2 or more	of the same ne time	Save this logic to fil

Credits to Bill Turnbull www.billturnbull.com

who reminds the young generation how websites looked like in the 90s.

True legend!



Source: http://www.billturnbull.com/solitaire/Logic.html

22 M Social Complexity

Quantifying model uncertainty





Confidence intervals (CIs)

• Cls are used to quantify the uncertainty of a model based a sample of observations (or model runs).







Confidence intervals (CIs)

- Cls are used to quantify the uncertainty of a model based a sample of observations (or model runs).
- We need three quantities
 - Confidence level (e.g., 90%)
 - Point estimate (e.g., sample mean)
 - Margin of error to be calculated
 - Use t-score when sample size < 30; use z-score otherwise.





Confidence intervals (CIs)

- Cls are used to quantify the uncertainty of a model based a sample of observations (or model runs).
- We need three quantities
 - Confidence level (e.g., 90%)
 - Point estimate (e.g., sample mean)
 - Margin of error to be calculated
 - Use t-score when sample size < 30; use z-score otherwise.
- Confidence interval = *Sample statistic* ± *Margin of error*





Precision vs. reliability in Cls

- For instance, confidence interval for height of adults
 - 5'8" \pm 0.01" (precise but not very reliable)
 - 5'8" \pm 1.00' (not very precise but reliable)

margin of error

mean







Confidence intervals example







Confidence intervals example



```
# for 95% confidence interval
m = np.mean(model_results)
std = np.std(model_results)
lb = m - 1.96*std
ub = m + 1.96*std
```

plt.hist(model_results)
plt.axvline(x=m,color="black",label="mean")
plt.axvline(x=lb,color="red",label="Lower bound (95% CI)")
plt.axvline(x=ub,color="purple",label="Upper bound (95% CI)")
plt.legend()
plt.show()







Confidence intervals example

for 90% confidence interval
m = np.mean(model_results)
std = np.std(model_results)
lb = m - 1.645*std
ub = m + 1.645*std

plt.hist(model_results)
plt.axvline(x=m,color="black",label="mean")
plt.axvline(x=lb,color="red",label="Lower bound (90% CI)")
plt.axvline(x=ub,color="purple",label="Upper bound (90% CI)")
plt.legend()
plt.show()



for 80% confidence interval
m = np.mean(model_results)
std = np.std(model_results)
lb = m - 1.282*std
ub = m + 1.282*std

plt.hist(model_results)
plt.axvline(x=m,color="black",label="mean")
plt.axvline(x=lb,color="red",label="Lower bound (80% CI)")
plt.axvline(x=ub,color="purple",label="Upper bound (80% CI)")
plt.legend()
plt.show()





30 M Social Complexity

Where to use confidence intervals?

- Inspect the output of the quantity of interest.
 - E.g., a **component of a model** or the **entire model output**.
- Inspect observations collected from the real world.
 - Can be related to **model input parameters** (e.g., age distribution of the population) or **expected output values** (e.g., number of people died of COVID-19).





Where to use confidence intervals?

- Inspect the output of the quantity of interest.
 - E.g., a **component of a model** or the **entire model output**.
- Inspect observations collected from the real world.
 - Can be related to **model input parameters** (e.g., age distribution of the population) or **expected output values** (e.g., number of people died of COVID-19).
- Things to check
 - How precise or reliable?
 - Difference from the real world measurements?
 - Is it chaotic?





Introducing uncertainty in models





Pseudo-random number generators (RNGs)

- A family of algorithms that can generate "random-looking" numbers.
- Different from physical random number generators (e.g., hardware that use thermal noise to generate random number).
- Given initial parameters, one can reproduce the entire sequence of numbers to be generated.
- All major programming languages provide RNGs.







Important features of RNGs

Randomness	 not biased towards certain values or distributions
Controllability	 "ability to reproduce its output, if desired"
Portability	 "ability to produce the same output on a variety of computer systems"
Efficiency	 "speed, with minimal computer-resource requirement"
Documentation	 "theoretical analysis and extensive testing"
LE N	Leemis and Park (2006)



RNGs



Source: https://www.explainxkcd.com/wiki/index.php/1277:_Ayn_Random





RNGs

- Given a *seed* value, we can regenerate the exact sequence of numbers.
- A simple technique: Linear Congruential Generator (LCG) Lehmer (1951).
 - $x_i = (a x_{i-1} + c) \mod m$
 - *a*: multiplier
 - *m*: modulus
 - *c*: increment
 - x_0 : the seed
 - One of the standard RNG in programming languages
 - Java still uses it w/ parameters a = 25214903917, $m = 2^{48}$, c = 11
- Another popular one is Mersenne Twister (Matsumoto 1998)





• Python's NumPy package and built-in random package implement the Mersenne Twister

Import the NumPy package

import numpy as np







• Python's NumPy package and built-in random package implement the Mersenne Twister

Import the NumPy package

Generate a value between 0 and 1 = U(0,1)

import numpy as np

np.random.random()

0.5488135039273248





Social Complexity

• Python's NumPy package and built-in random package implement the Mersenne Twister

Import the NumPy package

Generate a value between 0 and 1 = U(0,1)

Generate an array of five values U(0,1)

import numpy as np
np.random.random()
0.5488135039273248
np.random.random(5)

array([0.71518937, 0.60276338, 0.54488318, 0.4236548 , 0.64589411])





• Python's NumPy package and built-in random package implement the Mersenne Twister

Import the NumPy package

Generate a value between 0 and 1 = U(0,1)

Generate an array of five values U(0,1)

import numpy as np

np.random.random()

0.5488135039273248

np.random.random(5)

array([0.71518937, 0.60276338, 0.54488318, 0.4236548 , 0.64589411])

Generate a matrix U(0,1)

np.random.random((2,3))

array([[0.43758721, 0.891773 , 0.96366276], [0.38344152, 0.79172504, 0.52889492]])





How critical RNGs in model reproducibility

A personal story





• Can be *continuous* or *discrete*



(NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/)



• Can be *continuous* or *discrete*

- Usually, we fit a statistical distribution to data to understand the underlying mechanism that generates the data.
 - Often need to try many different distributions and check the goodness of fit.



(NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/)



• Can be *continuous* or *discrete*

- Usually, we fit a statistical distribution to data to understand the underlying mechanism that generates the data.
 - Often need to try many different distributions and check the **goodness of fit**.
- Probability density function (PDF) is used for continuous statistical distributions to describe the likelihood of sampling values from the value space.
 - **Probability mass function (PMF)** is used for discrete distributions



(NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/)



• Can be *continuous* or *discrete*

- Usually, we fit a statistical distribution to data to understand the underlying mechanism that generates the data.
 - Often need to try many different distributions and check the goodness of fit.
- Probability density function (PDF) is used for continuous statistical distributions to describe the likelihood of sampling values from the value space.
 - **Probability mass function (PMF)** is used for discrete distributions
- *Cumulative distribution function (CDF)* is to describe "the probability that the variable takes a value less than or equal to" given value. In the continuous case, CDF can be calculated by integrating its PDF.

MASON

(NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/)



PDF (PMF) and CDF examples

For more: <u>https://statdist.com/</u>







Sampling from statistical distributions example

Uniform distribution within 0-10 interval

np.random.uniform(0,10,size=10)

array([3.78653645, 9.68804991, 8.22695094, 2.36785104, 8.32063758, 0.15156402, 7.01776218, 6.63535641, 0.73149233, 7.88035412])







Sampling from statistical distributions example

Uniform distribution within 0-10 interval

np.random.uniform(0,10,size=10)

array([3.78653645, 9.68804991, 8.22695094, 2.36785104, 8.32063758, 0.15156402, 7.01776218, 6.63535641, 0.73149233, 7.88035412])

Normal distribution with mean=0 and standard deviation=5.

np.random.normal(0,5,size=10)

array([-0.34837998, 1.85523413, 4.78192638, 0.00827601, -0.42549727, -2.91218348, -5.89590159, 1.03210198, 2.46722228, -3.45734937])





Sampling from statistical distributions example

Uniform distribution within 0-10 interval

np.random.uniform(0,10,size=10)

array([3.78653645, 9.68804991, 8.22695094, 2.36785104, 8.32063758, 0.15156402, 7.01776218, 6.63535641, 0.73149233, 7.88035412])

Normal distribution with mean=0 and standard deviation=5.

np.random.normal(0,5,size=10)

array([-0.34837998, 1.85523413, 4.78192638, 0.00827601, -0.42549727, -2.91218348, -5.89590159, 1.03210198, 2.46722228, -3.45734937])

Poisson distribution with lambda=10.

np.random.poisson(10,size=10)

array([6, 20, 6, 11, 8, 4, 12, 12, 6, 10])



50 M Social Complexity

Other distributions you can sample from

- beta
- binomial
- chisquare
- dirichlet
- exponential
- f
- gamma

- geometric
- gumbel
- hypergeometric
- laplace
- logistic
- lognormal
- ...

More comprehensive index: <u>https://numpy.org/doc/stable/reference/random/index.html</u>

Older version: <u>https://docs.scipy.org/doc/numpy-1.14.0/reference</u>

```
s = np.random.lognormal(3.0, 1.0, size=1000)
```

plt.hist(s,50)
plt.show()





CSI 709/CSS 739 - Verification and Validation of Models — $\ensuremath{\mathbb{C}}$ Dr. Hamdi Kavak



Where do we use statistical distributions

- Input data modeling
- Output data modeling
- Compare model's output distribution against the real system's distribution.
 - Z-test
 - Kolmogorov–Smirnov (KS) test
 - ...







Multiple competing models

• How would you pick the best one?







Ways to compare model performance

- Kullback–Leibler (K L) divergence if we have a distribution of quantities (e.g., firm size distribution).
 - Describes the difference between two probability distributions.
 - Works for both continuous and discrete data.
- Wasserstein metric
- Pearson correlation
- Rank correlation (e.g., Kendall's tau)
- Various measures
 - MAE
 - MSE
 - RMSE
 - ...



Social Complexity

Visual techniques to inspect model output





Basic definitions

Let $X = \{x_1, x_2, ..., x_n\}$ be a set of **ordered** values gathered from *n* model runs.

• Mean: $\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$

• Variance:
$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{X})^2}{n-1}$$

• Standard deviation: $s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{X})^2}{n-1}}$

• First quartile*: median of the numbers from start to general median.

• Median**:
$$\begin{cases} x_{(n+1)/2} & \text{if } n \text{ is odd} \\ \frac{(x_{n/2}+x_{n/2+1})}{2} & \text{if } n \text{ is even} \end{cases}$$

- Third quartile***: median of the numbers from general median to end.
 - * First quartile(Q₁) or 25th percentile
 - ** Second quartile(Q₂) or 50th percentile
 - *** Third quartile(Q₃) or 75th percentile





Box and Whiskers plots

• Used to visually display the spread of data





M Social Complexity



Probability-Probability (P-P) Plots

- Used to compare a sample ($X = \{x_1, x_2, ..., x_n\}$) to a theoretical distribution (T).
- Steps
 - 1. Order your sample ascending.
 - 2. $\forall x_i \in X$, determine $p_i = P(X < x_i)$ using the CDF of T.
 - 3. Scatter plot p_i against (i 0.5)/n
 - 4. If the constructed plot looks like y = x, we can say they are similar.





Probability-Probability (P-P) Plots





CSI 709/CSS 739 - Verification and Validation of Models — © Dr. Hamdi Kavak

Social Complexity

Probability-Probability (P-P) Plots





CSI 709/CSS 739 - Verification and Validation of Models — © Dr. Hamdi Kavak



Probability-Probability (P-P) Plots

Uniform (0,1) vs. Standard Normal

Beta (1,5) vs. Standard Normal









Sources

- Collins, A. MSIM 651 Lecture Slides. Old Dominion University.
- Leemis, L. M., & Park, S. K. (2006). *Discrete-event simulation: A first course*. Upper Saddle River, NJ: Pearson Prentice Hall.
- Lehmer, D. H. (1951). Mathematical methods in large-scale computing units. *Annu. Comput. Lab. Harvard Univ., 26,* 141-146.
- Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: a 623dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1), 3-30.
- NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/



