ASCOT 3: NONLINEAR PRINCIPAL COMPONENTS ANALYSIS AND UNCERTAINTY QUANTIFICATION IN EARLY LIFECYCLE SPACECRAFT FLIGHT SOFTWARE COST ESTIMATION

NASA COST AND SCHEDULE SYMPOSIUM

MAY 2-4, 2023, PASADENA, CA

Melissa Hooke, Melissa.A.Hooke@jpl.nasa.gov;
Patrick Bjornstad; Jairus Hihn, PhD; Sam Fleischer, PhD
NASA Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109



James Johnson, James.K.Johnson@nasa.gov
National Aeronautics and Space Administration
Washington, DC 20546


OVERVIEW

- Challenges in spacecraft flight software cost estimation
- Why does ASCoT exist?
- Bayesian regression and improving our understanding of uncertainty
- Non-numerical data and Nonlinear Principal Components Analysis
- *k*-Nearest-Neighbors and Clustering algorithms

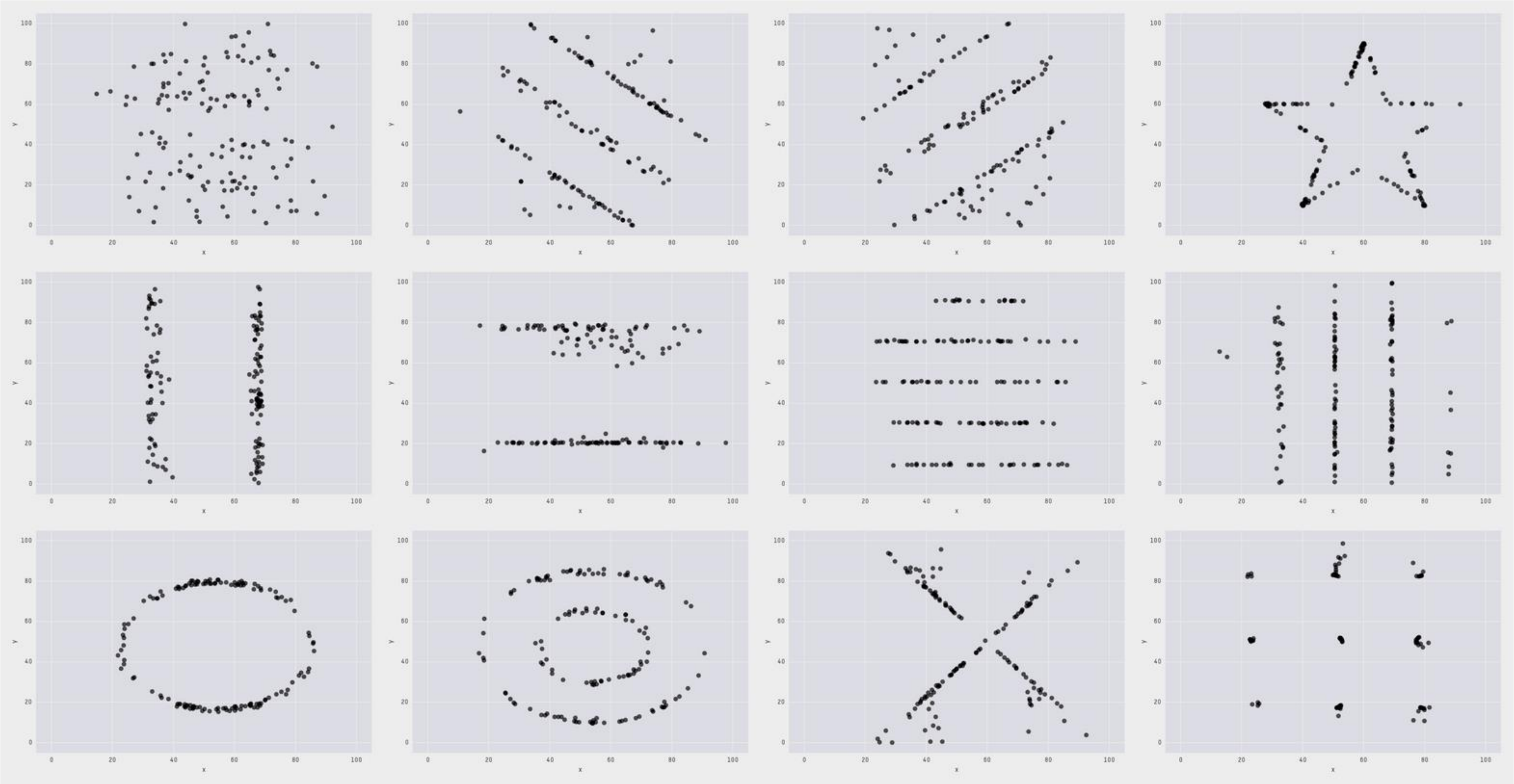
CHALLENGES IN SPACECRAFT FLIGHT SOFTWARE COST ESTIMATION

- Requirements are not known at early phases of the mission, and architecture trade studies are routine.
- Software estimation is, to some degree, fundamentally uncertain under the best conditions.
- It is difficult to budget with a large amount of uncertainty.
- Budget 'bogies' get set very early in the lifecycle… sometimes based on casual conversation… and project managers will want to hold you to that number.
- Current proposal and planning processes encourages (demands) under-estimating.

WHY DOES ASCOT EXIST? (1/2)

- ASCoT was created to enable estimators to better embrace the uncertainty
- ASCoT expands the range of cost estimation models to include formal analogic cost estimation, which can be better suited to early project formulation
  - ASCoT includes both parametric & analogic cost models
  - Analogic models can perform much better than parametric models with sparse, noisy data
  - Analogic models represent what is known in the very early lifecycle more accurately than parametric models
- ASCoT provides models that only require basic system-level inputs that are known in the early lifecycle

WHY DOES ASCOT EXIST? (2/2) – THE DATASAURUS DOZEN



```
X Mean: 54.26
Y Mean: 47.83
X SD  : 16.76
Y SD  : 26.93
Corr. : -0.06
```

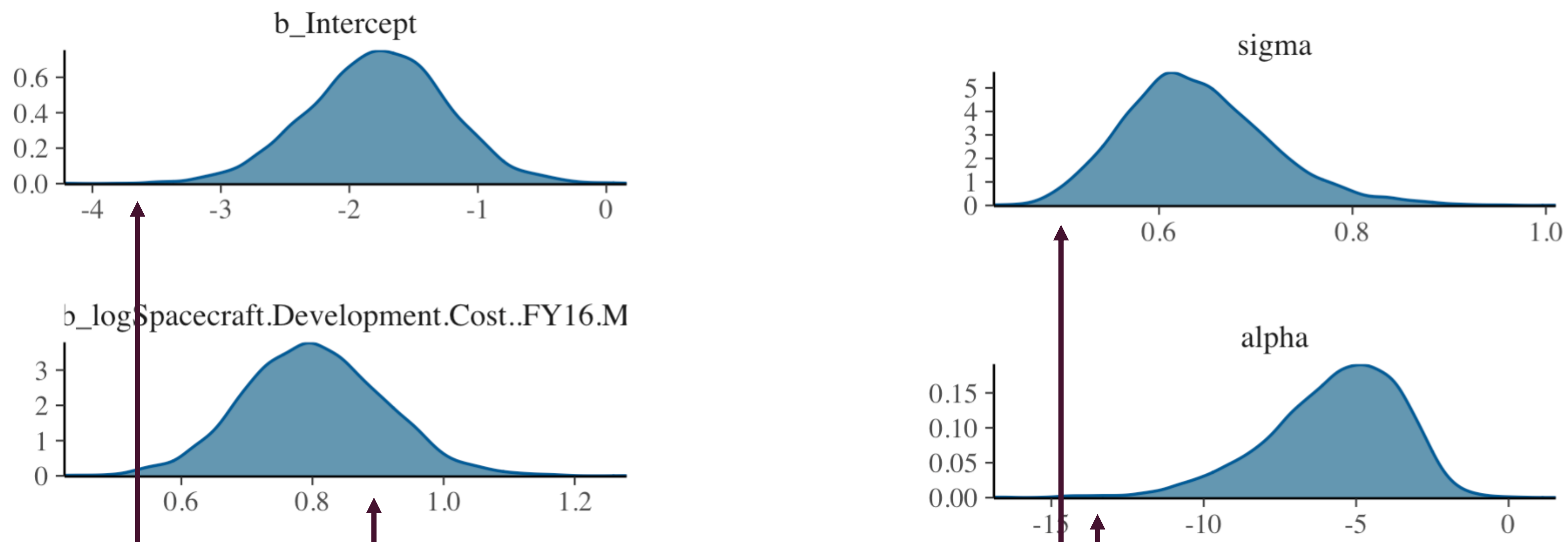
All of these datasets have identical statistics when rounded to the nearest 100th.

J. Matejka and G. Fitzmaurice, 2017

BAYESIAN REGRESSION AND IMPROVING OUR UNDERSTANDING OF UNCERTAINTY

- When regression *is* appropriate, ASCoT improves parametric models by providing as much uncertainty as is appropriate, in the regression.
  - Epistemic uncertainty is uncertainty in model parameters or model form
  - Aleatoric uncertainty is uncertainty inherent to the data generation process (i.e. distribution around the mean line)
- Bayesian statistics allows us to set smart priors based on expert opinion prior to ingesting data.
  - "In the absence of data, what is appropriate to assume?"

BAYESIAN CER – POSTERIOR DISTRIBUTION



$$\log(\text{Software Cost}) \sim \text{SkewNormal}(\mu, \sigma, \alpha)$$
$$\mu = \beta_0 + \beta_1 \log(\text{Spacecraft Cost}) + \epsilon$$

Priors

$\alpha \sim N(0,4)$

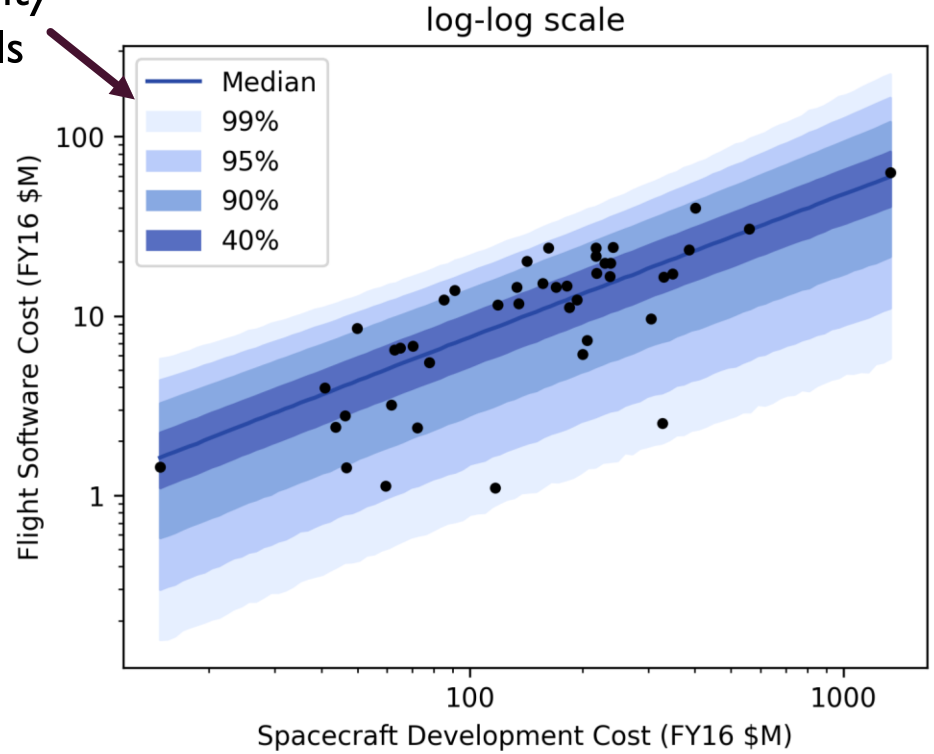
$\sigma \sim t(3,0,2.5)$

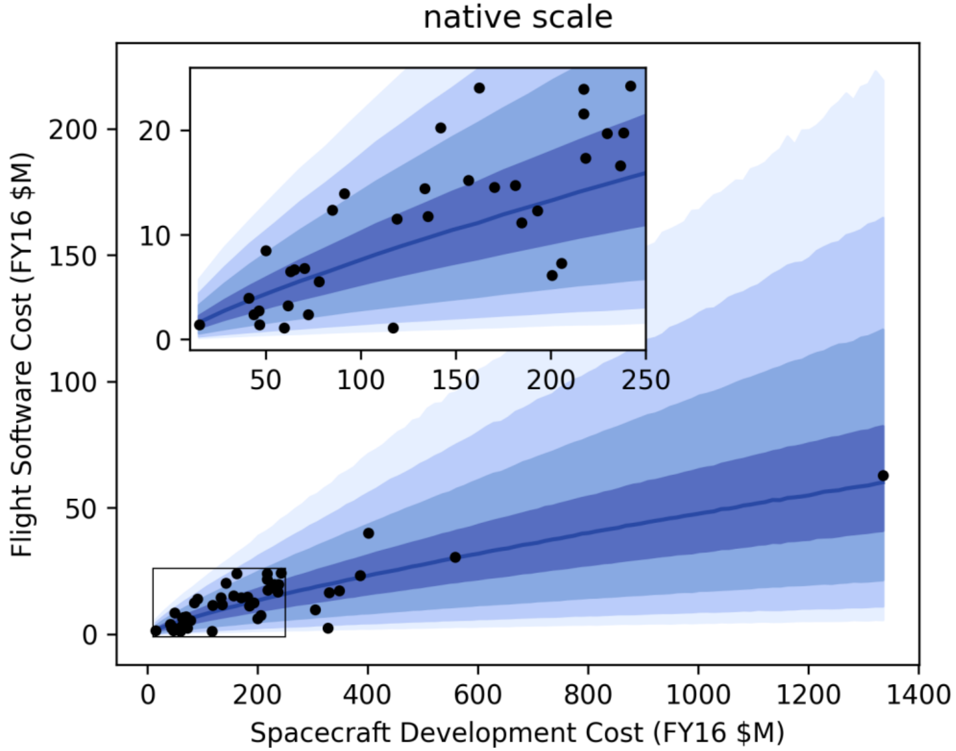
$\beta_0 \sim t(3,2.5,2.5)$

$\beta_1 \sim U(-\infty, \infty)$

BAYESIAN CER – POSTERIOR PREDICTIVE DISTRIBUTION

credibility intervals





$$\log(\text{Software Cost}) = \beta_0 + \beta_1 \log(\text{Spacecraft Cost}) + \epsilon$$
$$\epsilon \sim \text{SkewNormal}(\sigma, \alpha)$$

Priors

$\alpha \sim N(0,4)$

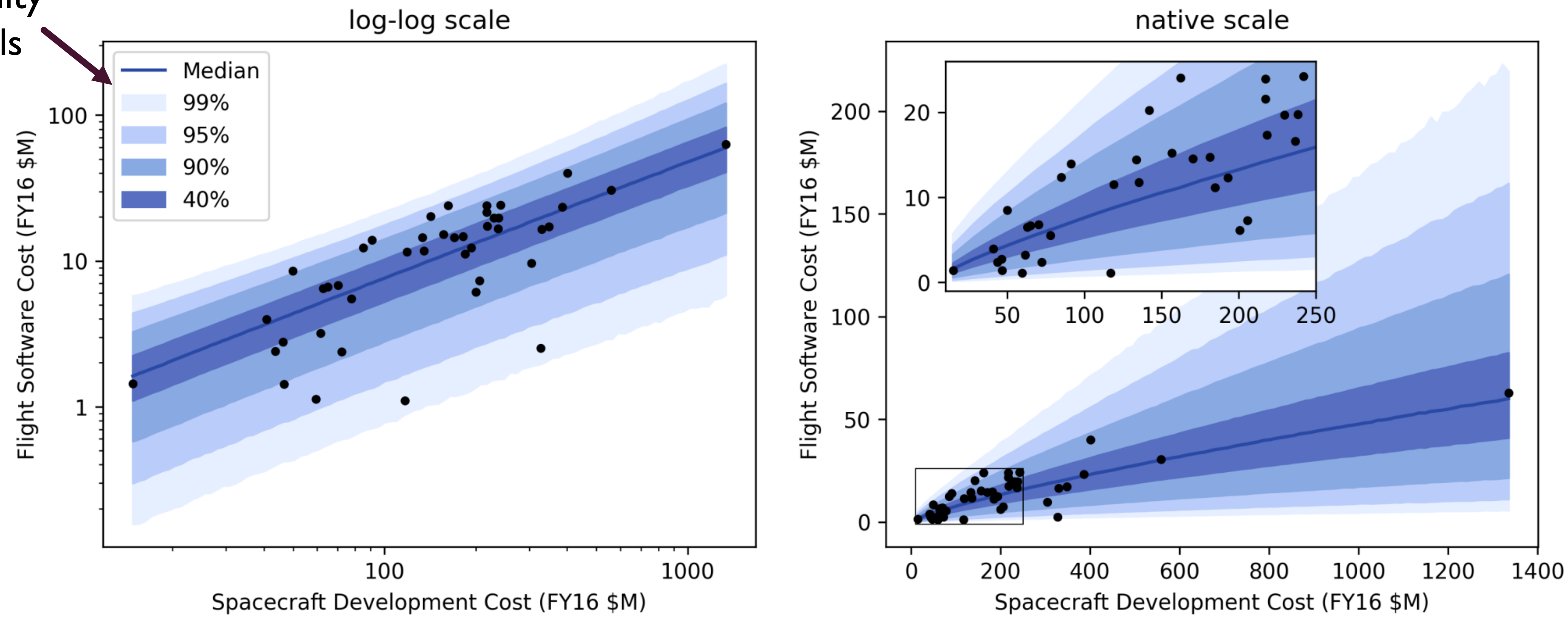
$\sigma \sim t(3,0,2.5)$

$\beta_0 \sim t(3,2.5,2.5)$

$\beta_1 \sim U(-\infty, \infty)$

BAYESIAN CER – POSTERIOR PREDICTIVE DISTRIBUTION

credibility intervals



- Model with skew normal error term performs better *predictively* than model with normal error term
  - Captures low outliers without pulling median prediction down
- Simple regression performs better *predictively* than regression models with other perceived software cost drivers such as number of instruments, destination, or redundancy **(short version: avoids overfitting)**

K-NEAREST-NEIGHBORS AND CLUSTERING ALGORITHMS – INPUT VARIABLES (1/3)

- Inheritance (as-is or modified code from a previous mission)
  - Theoretically, this is a number between 0% and 100%.
  - In practice, Project Software Systems Engineers (PSSEs) have only rough estimates. We categorize them into five bins:
    - "Very Low to None" "Low" "Medium" "High" "Very High"
- Mission Size (total mission cost, including operations)
  - Theoretically, this is a precise positive number.
  - In practice, we have only rough estimates of what will be the total cost
  - *However,* we have a very good idea of the cost target or mission class. The categories are:
    - "Small" "Medium" "Large" "Very Large"

K-NEAREST-NEIGHBORS AND CLUSTERING ALGORITHMS – INPUT VARIABLES (2/3)

- Mission Type
  - "Orbiter/flyby" "Observatory" "Lander" "Rover"
- Redundancy
  - "Single String" (no backup computer on board) "Dual String – Cold" (backup on board but nominally off)
    "Dual String – Warm" (backup maintaining continuous operations)
- Destination
  - "Earth" "Inner Planetary" "Asteroid / Comet" "Outer Planetary"
- Number of Instruments (particle detectors, magnetometers, spectometers, and other scientific instruments)
- Number of Deployables (solar arrays, booms, arms, etc.)

K-NEAREST-NEIGHBORS AND CLUSTERING ALGORITHMS – INPUT VARIABLES (3/3)

Nominal and Categorical Variables

Inheritance
Mission Size
Mission Type
Redundancy
Destination

How do you calculate the "distance" between missions with non-numerical data?

Numerical Variables

Number of Instruments
Number of Deployables

Is the "distance" between 2 instruments and 3 instruments equal to that of between 3 instruments and 4?

HOW DO YOU NUMERICIZE CATEGORICAL DATA?

- *k*NN and Clustering algorithms need numbers, so we need to quantify the non-numerical data.

$$y(x) = \frac{\sum_{i=1}^{k} \frac{y_i}{d(x_i, x)}}{\sum_{i=1}^{k} \frac{1}{d(x_i, x)}}$$

Use the k data points x_i closest to the input x. The prediction is ***an average of those k points***, weighted inversely by their distance to x.

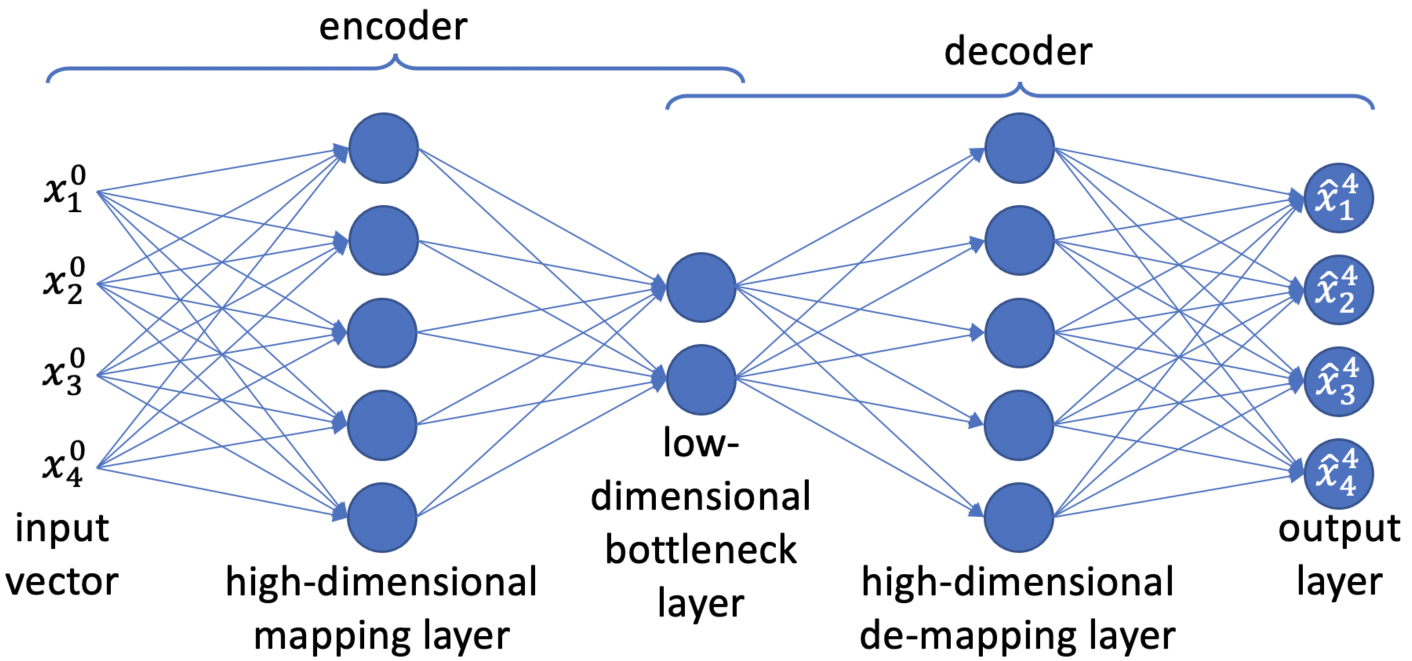
Use all points in the closest cluster to the input x. The prediction is ***an average of the points in the cluster***, weighted inversely by their distance to x.

$$y(x) = \frac{\sum_{(x_i, y_i) \in C_j} \frac{y_i}{d(x_i, x)}}{\sum_{(x_i, y_i) \in C_j} \frac{1}{d(x_i, x)}}$$

- We let the data tell us what really is the best way to quantify the data.
  - We rely on a Nonlinear Principal Components Analysis (NLPCA) algorithm to teach us the optimal weights.

NONLINEAR PRINCIPAL COMPONENTS ANALYSIS – AUTO-ASSOCIATIVE NEURAL NETWORKS (ANN)

Auto-associative neural network



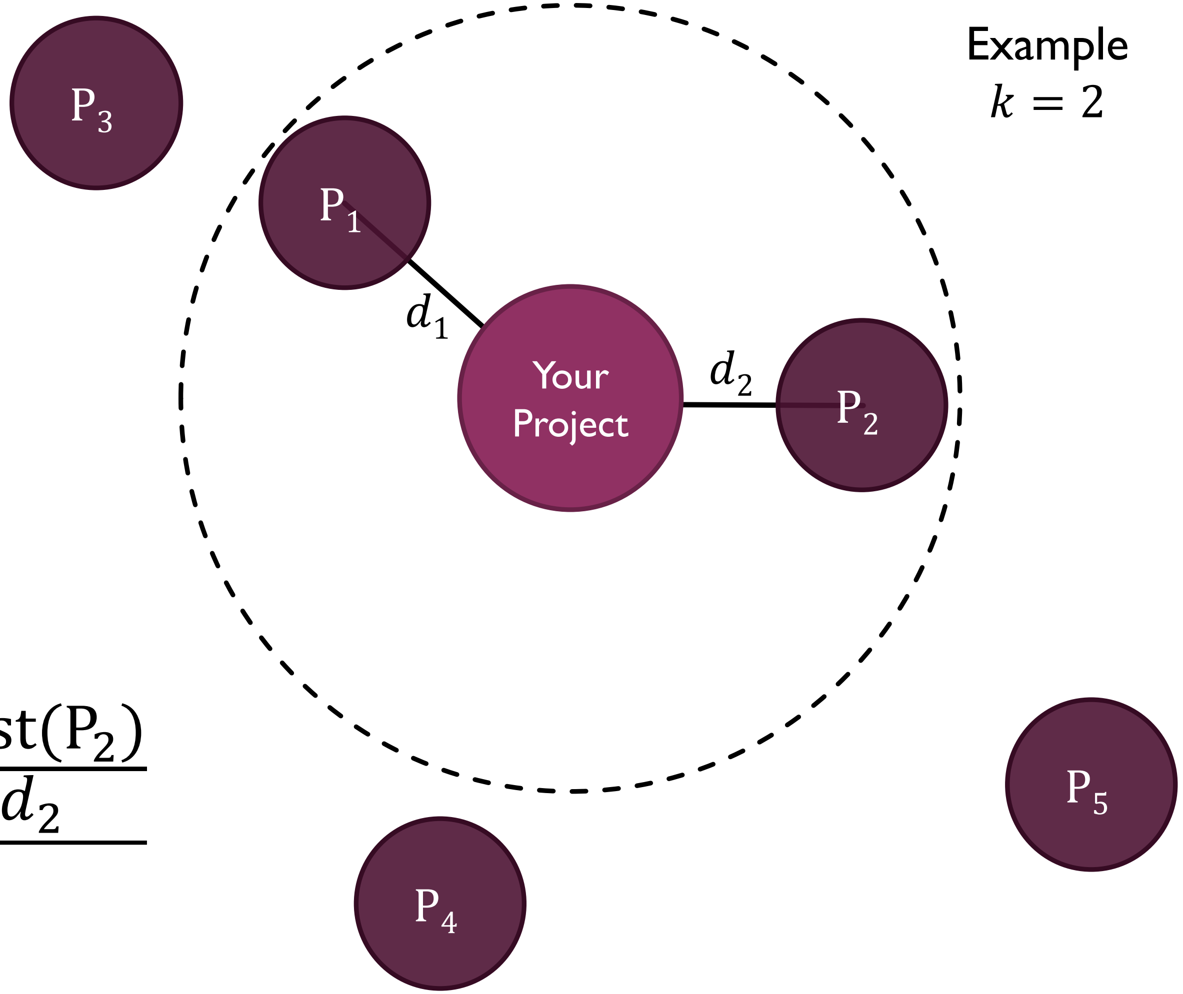
ANN parameters are optimized such that the difference between the output layer and the input layer is minimized.

Goal: the *low-dimensional bottleneck layer* must adequately retain the information contained in the input layer.

Result: A non-numeric input layer can be projected onto a numeric, low-dimensional space.

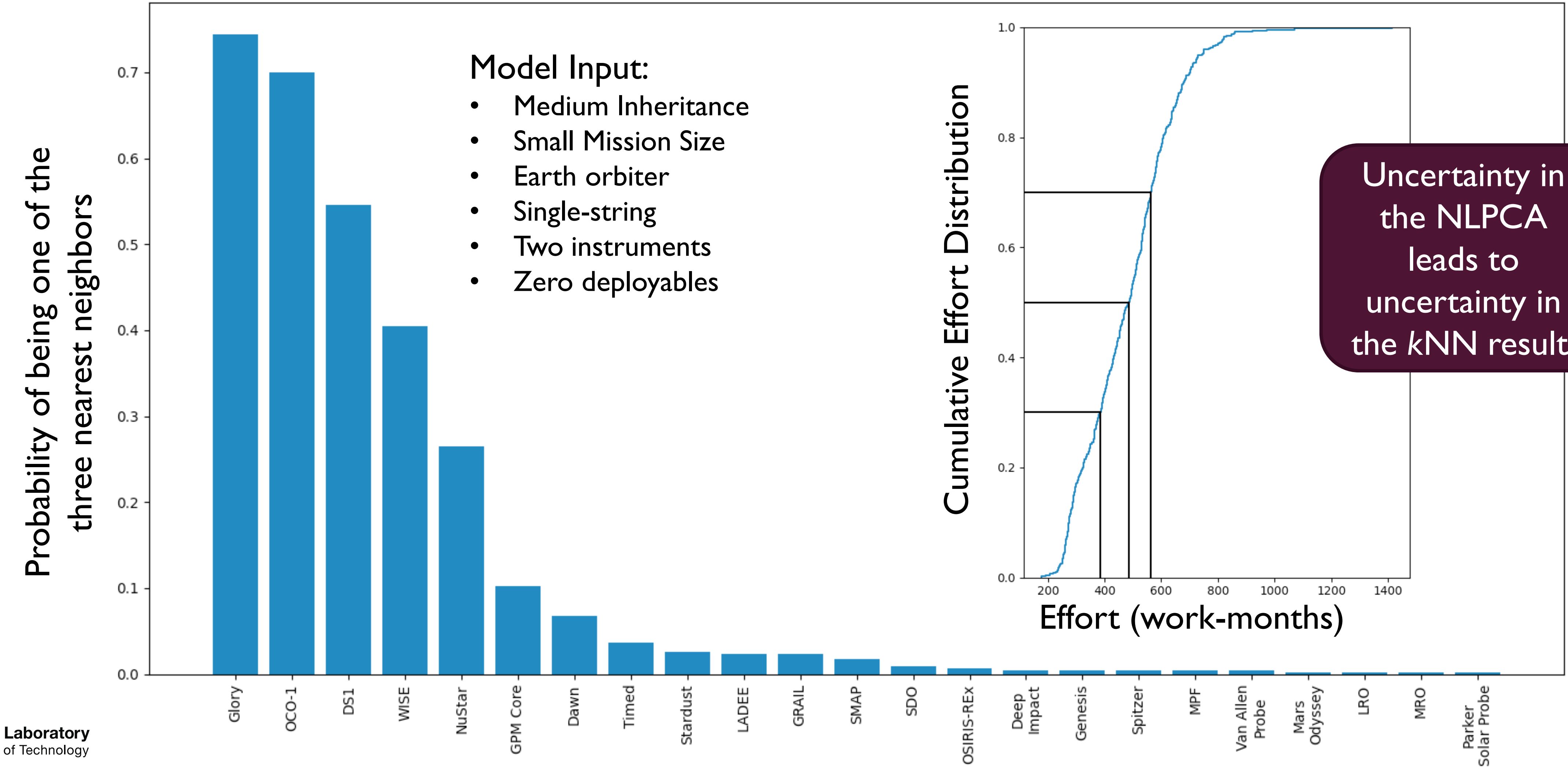
KNN ALGORITHM OVERVIEW

- Once we have our missions in a low-dimensional numeric space, we can calculate the distance from each mission to any model input easily (in a well-defined manner)
- If we choose $k = 2$, we only use the closest two missions to generate an estimate.



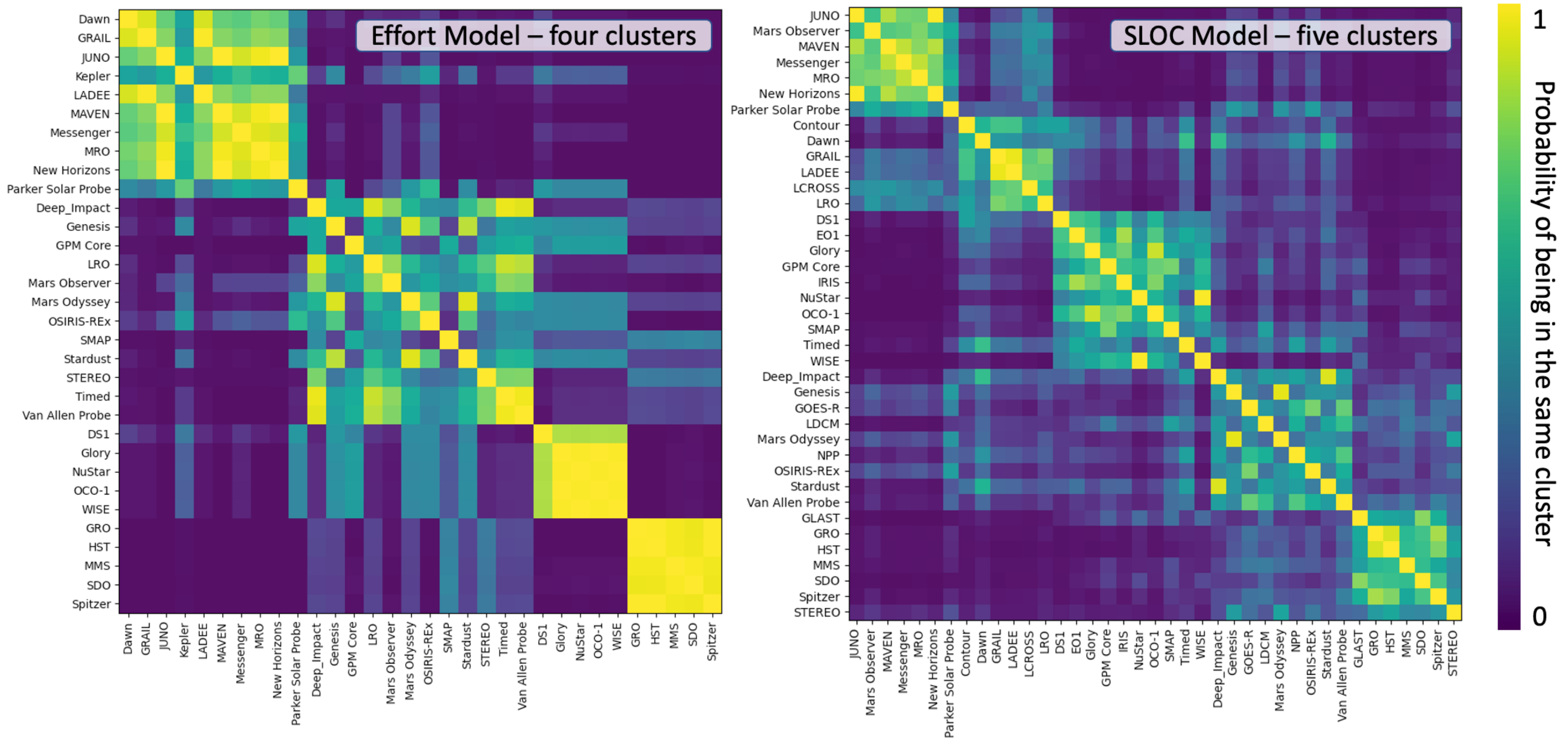
$$\text{Cost(Your Project)} = \frac{\frac{\text{Cost}(P_1)}{d_1} + \frac{\text{Cost}(P_2)}{d_2}}{\frac{1}{d_1} + \frac{1}{d_2}}$$

*K*NN MODEL EXAMPLE OUTPUT



Uncertainty in the NLPCA leads to uncertainty in the *k*NN result.

CLUSTERING ALGORITHM OVERVIEW



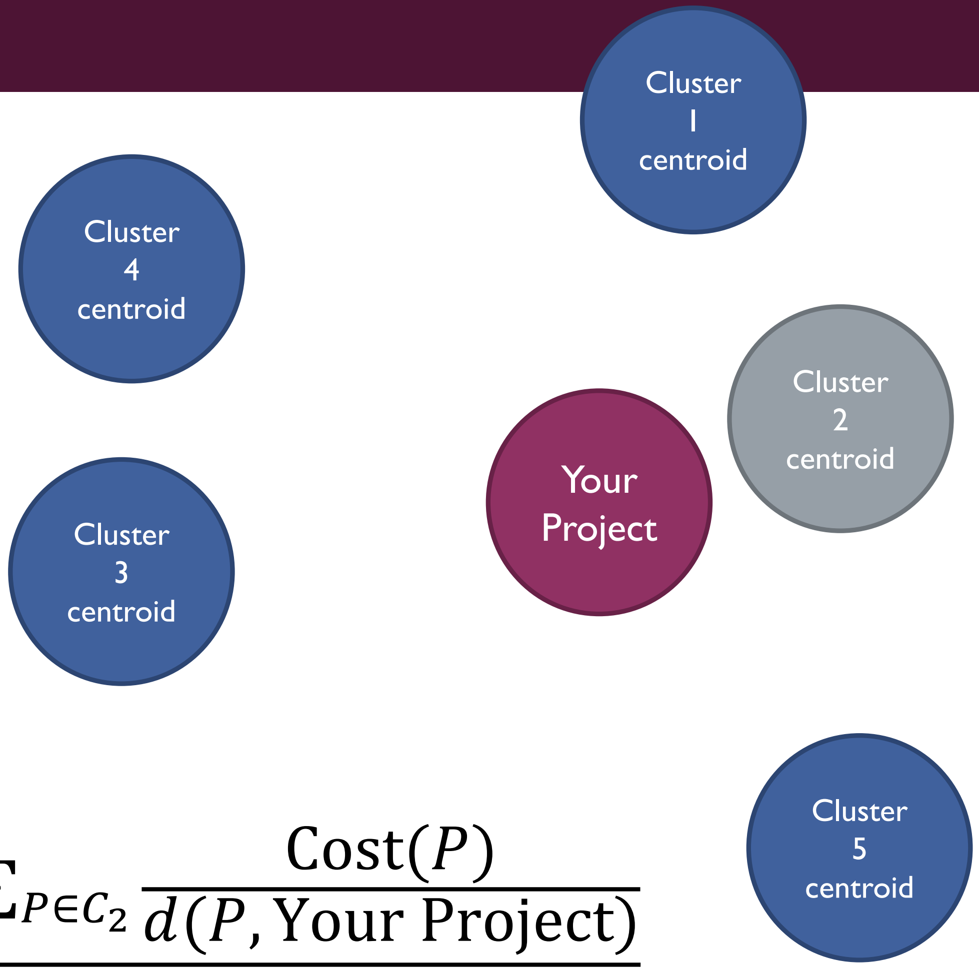
Probabilistic Linkage Matrices
Calculated using the *k*-Means algorithm in NLPCA space
(Cassini, Galileo, and Rovers and Landers are removed).

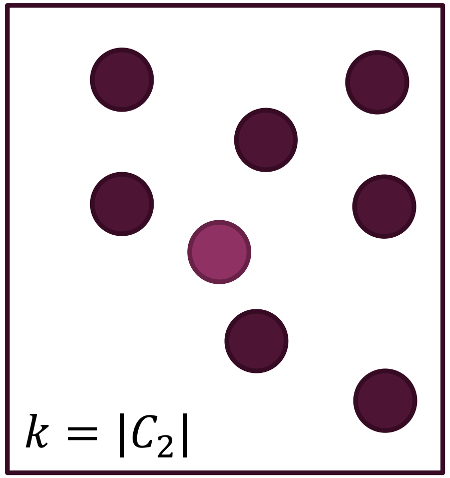
Effort Model Clusters						
1. Very Large, Old, Outer Planetary	2. Rovers	3. Landers	4. Large, Complex, Inner-Outer Planetary	5. Large, Complex, Earth-Inner Planetary	6. Smaller, Higher Inheritance	7. Large, Earth Observatories and Constellations
Cassini	MER	Insight	Dawn	Deep Impact	DS1	GRO
Galileo	MPF	Phoenix	GRAIL	Genesis	GLORY	HST
	MSL		JUNO	GPM Core	NuStar	MMS
			Kepler	LRO	OCO-1	SDO
			LADEE	Mars Observer	WISE	Spitzer
			MAVEN	Mars Odyssey		
			Messenger	OSIRIS-REx		
			MRO	SMAP		
			New Horizons	Stardust		
			Parker Solar Probe	STEREO		
				TIMED		
				Van Allen Probe		

SLOC Model Clusters							
1. Very Large, Old, Outer Planetary	2. Rovers	3. Landers	4. Large, Complex, Inner-Outer Planetary	5. Large, Moderately Complex, Dual String (Cold)	6. Smaller or Simple, Earth – Asteroid/ Comet	7. Small-Medium, Single-String Inner-Planetary or Dual String (Cold) Asteroid/Comet	8. Large, Earth Observatories and Constellations
Cassini	MER	Insight	JUNO	Deep Impact	DS1	Contour	GLAST
Galileo	MPF	Phoenix	Mars Observer	Genesis	EO1	Dawn	GRO
	MSL		MAVEN	GOES-R	GLORY	GRAIL	HST
			Messenger	LDCM	GPM Core	LADEE	MMS
			MRO	Mars Odyssey	IRIS	LCROSS	SDO
			New Horizons	NPP	NuStar	LRO	Spitzer
			Parker Solar Probe	OSIRIS-REx	OCO-1		STEREO
				Stardust	SMAP		
				Van Allen Probe	TIMED		
					WISE		

CLUSTERING ALGORITHM OVERVIEW

- Once we have our missions in a low-dimensional numeric space, we can calculate the distance from each mission to the "center" of any cluster
- Once in a cluster with *k* missions, use the *k*NN weighted average formula for the estimate.





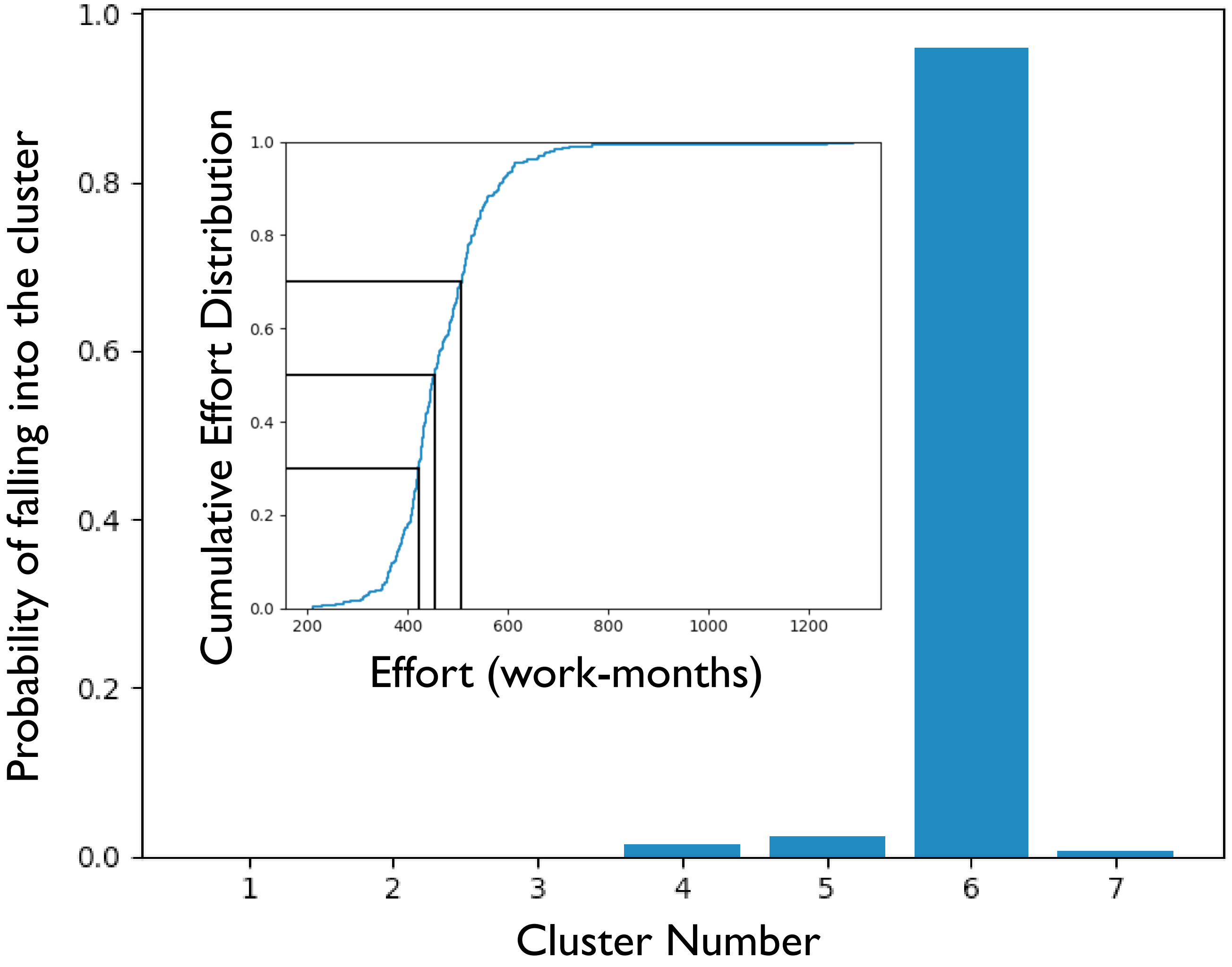
$$\text{Cost(Your Project)} = \frac{\sum_{P \in C_2} \frac{\text{Cost}(P)}{d(P, \text{Your Project})}}{\sum_{P \in C_2} \frac{1}{d(P, \text{Your Project})}}$$

CLUSTERING MODEL EXAMPLE OUTPUT

Model Input:

- Medium Inheritance
- Small Mission Size
- Earth orbiter
- Single-string
- Two instruments
- Zero deployables

Cluster 6 (Smaller, Higher Inheritance)
DS1
GLORY
NuStar
OCO-1
WISE



Uncertainty in the NLPCA leads to uncertainty in the cluster result.

Uncertainty in the Effort distribution is caused by uncertainty in the NLPCA as well as uncertainty in the cluster.

KEY TAKEAWAYS

- ASCoT was created to enable estimators to embrace the uncertainty of software cost estimation
- ASCoT's new Bayesian regressions and new KNN/Clustering algorithms better capture the uncertainty of estimating during early project formulation
- Go check out ASCoT on ONCE!!!

THANKS!

- Please contact
  - Melissa Hooke (Melissa.A.Hooke@jpl.nasa.gov)
  - James Johnson (James.K.Johnson@nasa.gov),
  - Patrick Bjornstad (Patrick.T.Bjornstad@jpl.nasa.gov), and
  - Jairus Hihn (Jairus.M.Hihn@jpl.nasa.gov)

  We love to chat about collecting and cleaning data, statistics and machine learning, and software costing.

- Thank you! Any questions?

 NASA HQ OCFO Strategic Investments Division provides the funding and management for the development of the ASCoT model and the ONSET framework.

BACKUP

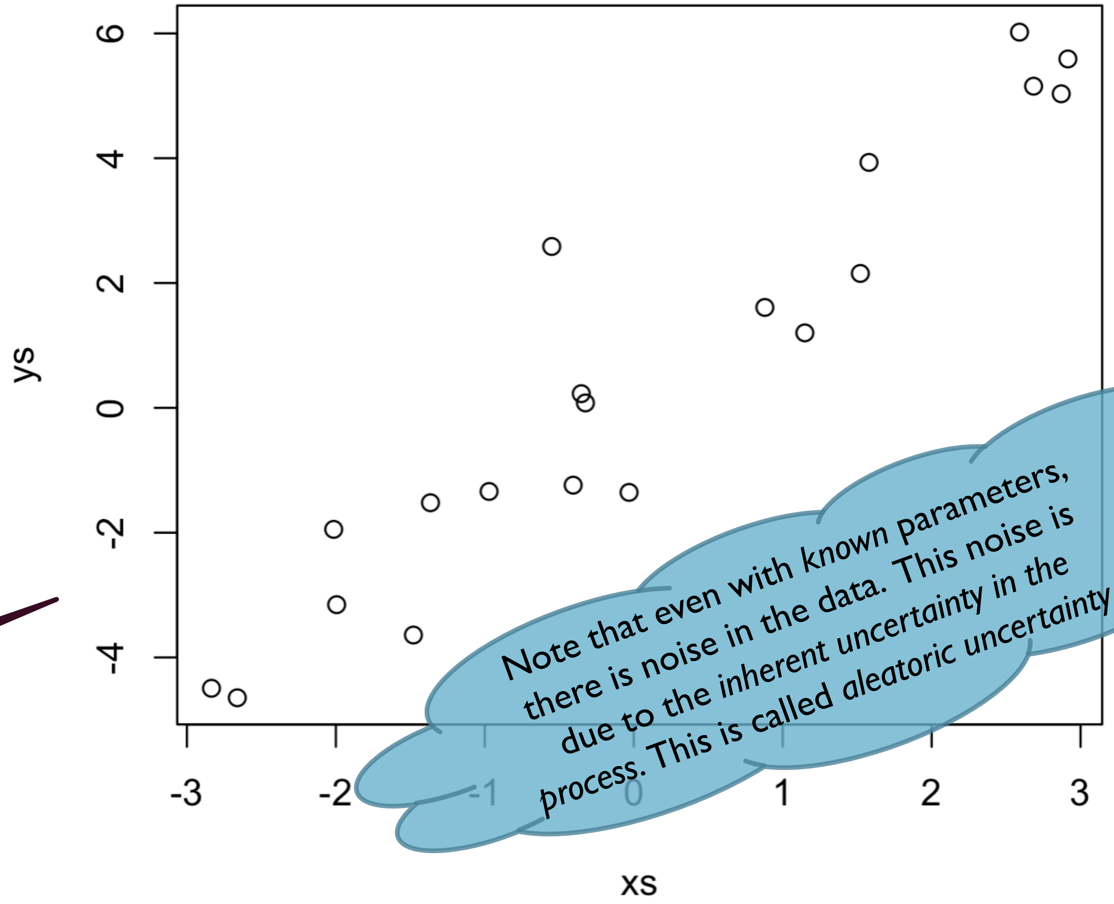
BAYESIAN SIMPLE LINEAR REGRESSION USING THE R PACKAGE BRMS (BAYESIAN REGRESSION MODELS USING STAN) (1/2)

```
slope <- 1.9
intercept <- 0.4
sigma <- 1.3

N <- 20
xs <- runif(N, min=-3, max=3)
signal <- slope*xs + intercept
noise <- rnorm(N, mean=0, sd=sigma)
ys <- signal + noise

plot(xs, ys)
```

Set the parameters of the model.

Simulate the process.

Plot the data.



Note that even with known parameters, there is noise in the data. This noise is due to the inherent uncertainty in the process. This is called *aleatoric uncertainty*.

BAYESIAN SIMPLE LINEAR REGRESSION USING THE R PACKAGE BRMS (BAYESIAN REGRESSION MODELS USING STAN) (2/2)

```
library(brms)

d <- data.frame(x=xs, y=ys)
model <- brm(y~x, data=d)

plot(model)
plot(conditional_effects(
      model, method='predict'),
      points=TRUE)

post <- as_draws_df(model)
head(post)
```

Load the BRMS library.

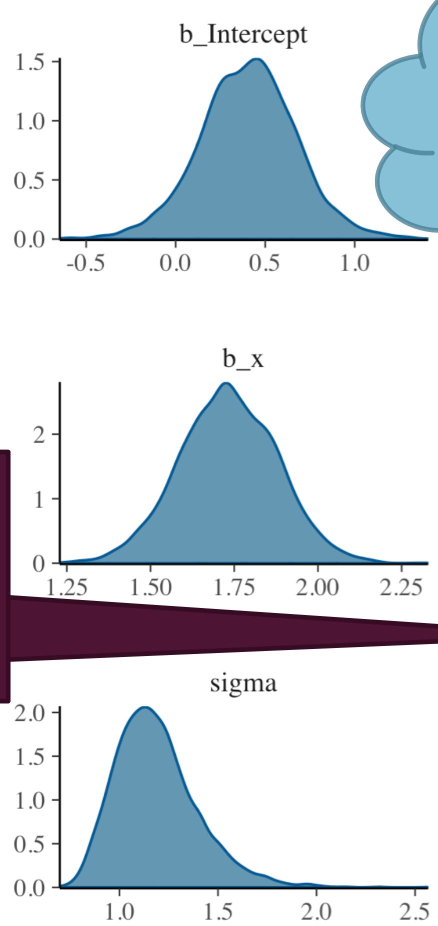
Define and fit the model.

See the fitted parameters

See how the model looks over the data.

Sample from the posterior.

	b_Intercept	b_x	sigma	lprior	lp__
1	-0.1447024	1.711225	1.3742691	-3.865604	-35.77348
2	0.8070629	1.549148	1.1941961	-3.864858	-35.20632
3	0.6598251	1.584357	1.1770281	-3.852215	-34.26019
4	0.1113087	1.498301	1.2273181	-3.841217	-35.30354
5	0.3810465	1.884662	0.8651947	-3.803884	-35.22898
6	0.5099172	1.653497	1.4078682	-3.876983	-34.48557



There is uncertainty in the fitted parameters. This is called *epistemic uncertainty* and represents a lack of knowledge.