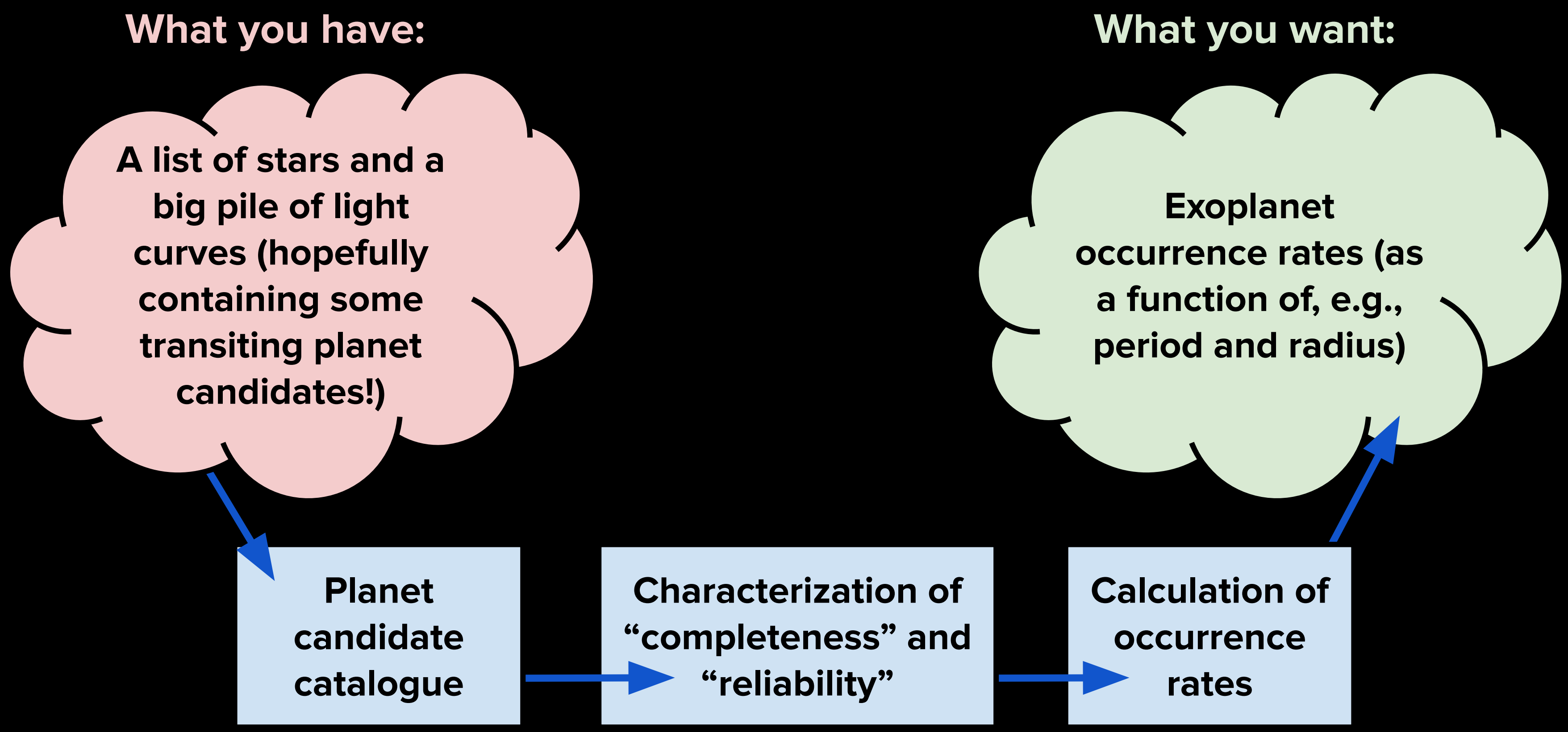
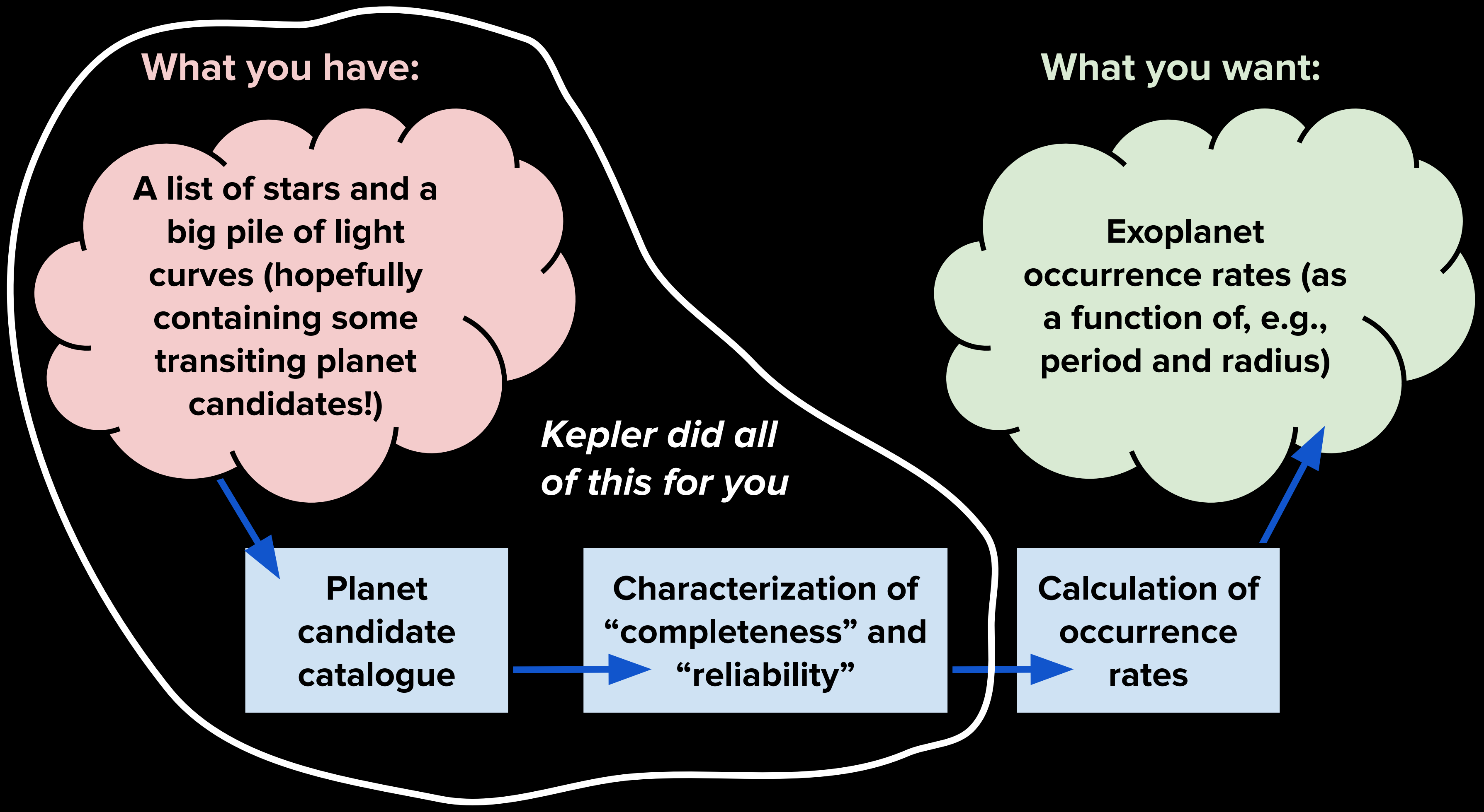
Introduction to Transit Occurrence Rate Methods

Michelle Kunimoto
Assistant Professor, UBC Physics and Astronomy

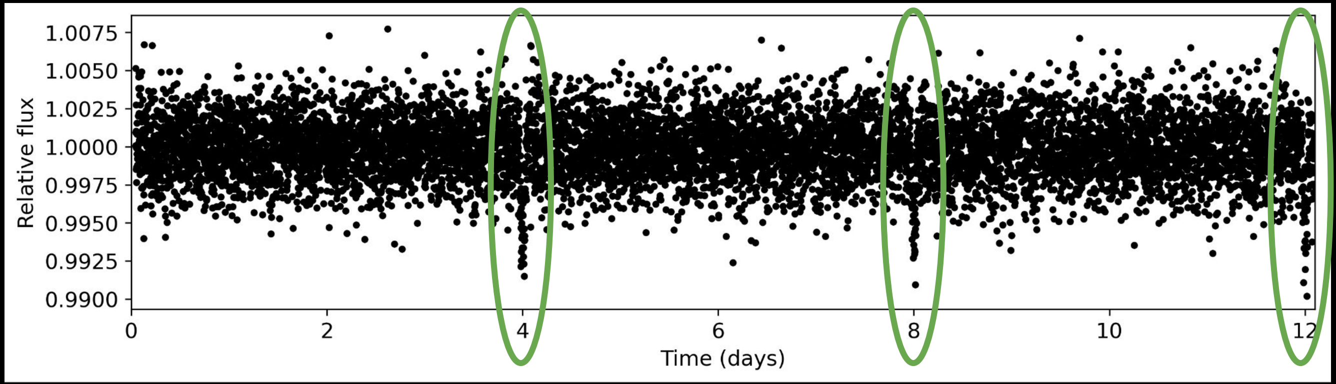
Sagan Summer Workshop, July 22, 2026


What you have:
A list of stars and a big pile of light curves (hopefully containing some transiting planet candidates!)
What you want:
Exoplanet occurrence rates (as a function of, e.g., period and radius)
Planet candidate catalogue
Characterization of "completeness" and "reliability"
Calculation of occurrence rates

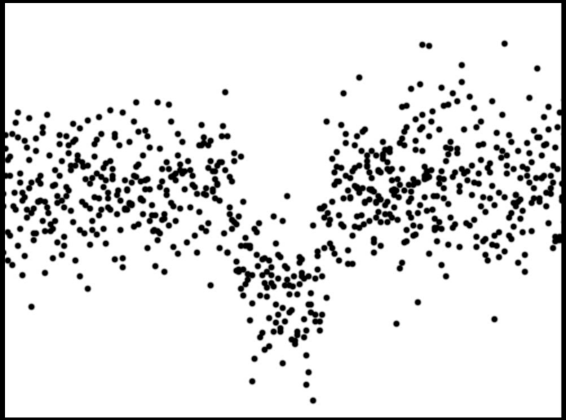

What you have:
A list of stars and a big pile of light curves (hopefully containing some transiting planet candidates!)
Kepler did all of this for you
What you want:
Exoplanet occurrence rates (as a function of, e.g., period and radius)
Planet candidate catalogue
Characterization of "completeness" and "reliability"
Calculation of occurrence rates

Step 1: Making your planet candidate catalogue

Light curves



Search the Data

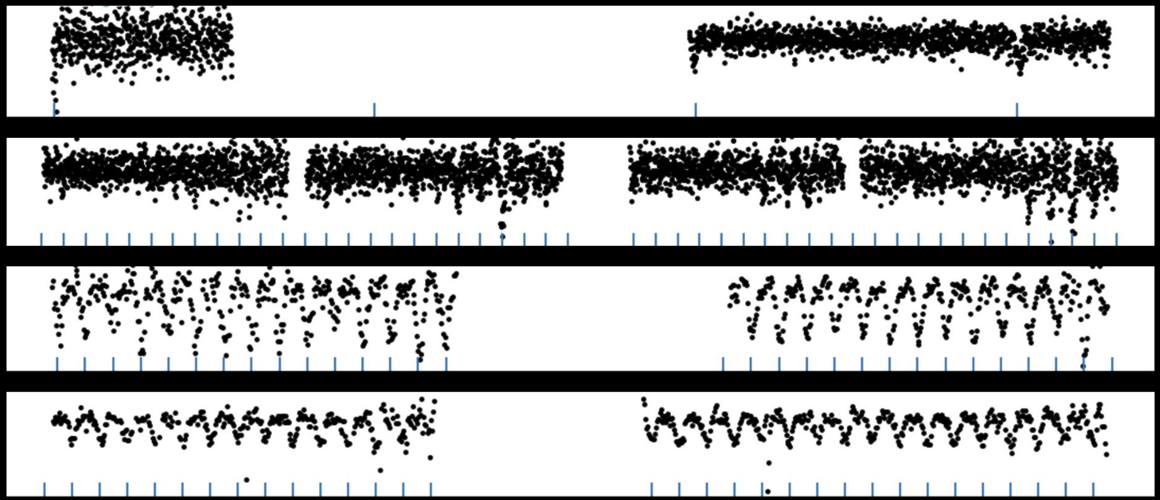


Vetting

"Is this a planet, or not?"

False Alarms (FAs)

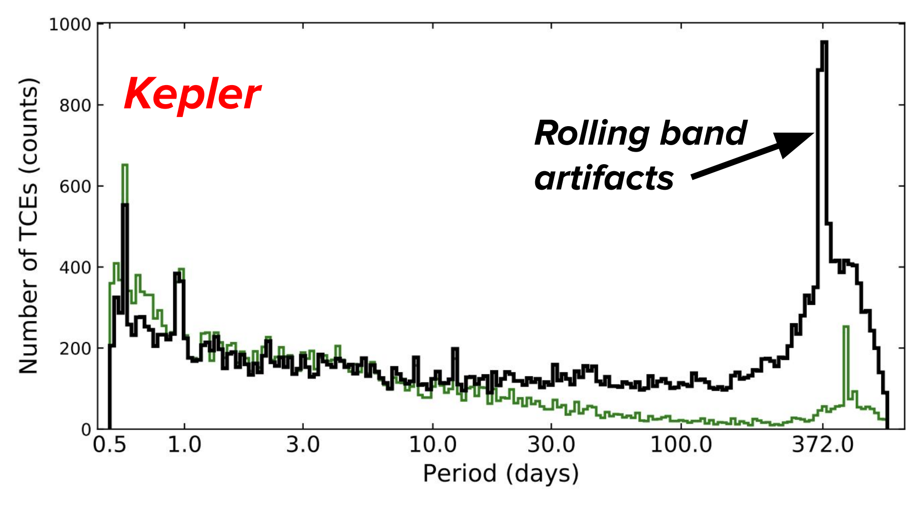
The vast majority of detections from a transit search will be “false alarms” (FAs). These include random fluctuations, instrumental systematics, stellar pulsations, starspots, detrending artifacts...



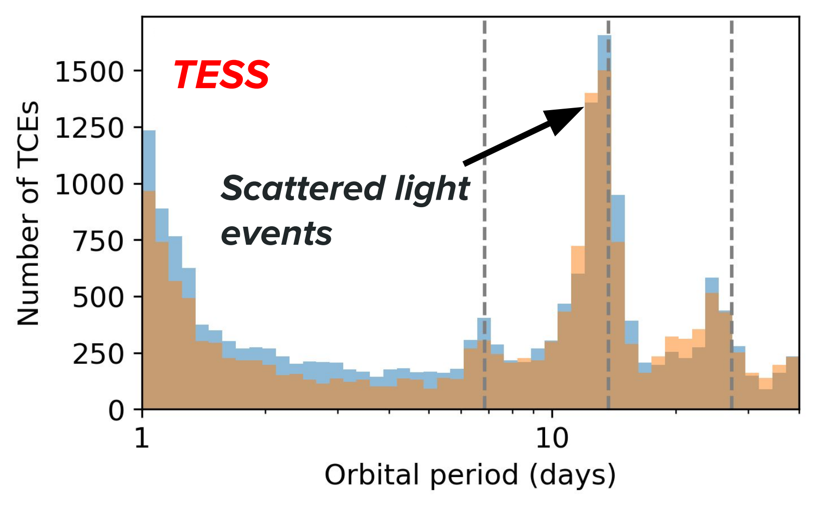
False Alarms (FAs)

Roman will have its own mission-specific false alarms (to be uncovered when we see the data)

Period histograms of detections can show huge pileups dominated by false alarms

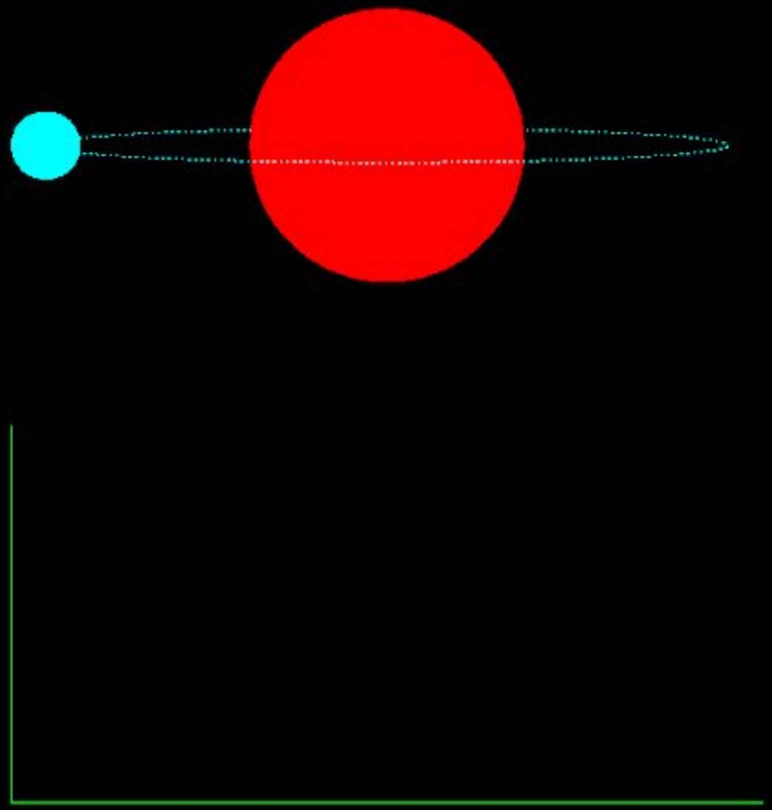


Thompson et al. (2018)



Kunimoto et al. (2023)

False Positives (FPs)



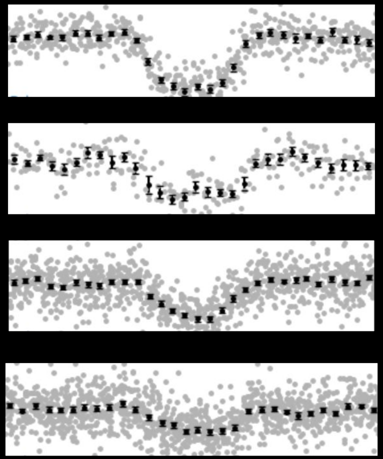
"False positives" include on- and off-target eclipsing binary stars (and even off-target transiting planet systems)

The Lazy-Exoplanet-Operations Vetter (LEO-Vetter)

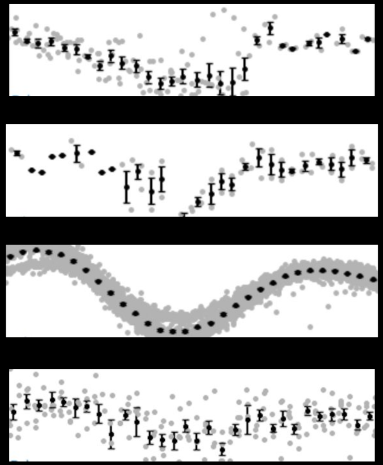
A fully automated vetting tool for identifying TESS planet candidates, inspired by the Kepler Robovetter (will be updated for Roman transit searches!)

github.com/mkunimoto/LEO-vetter

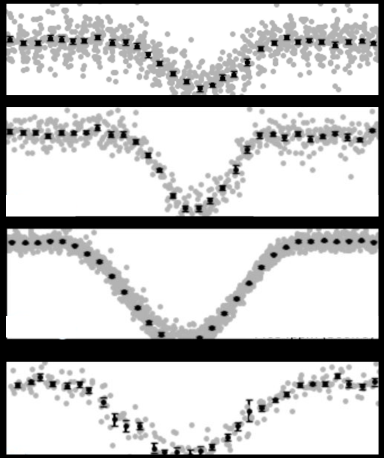
Planets



False Alarms



False Positives

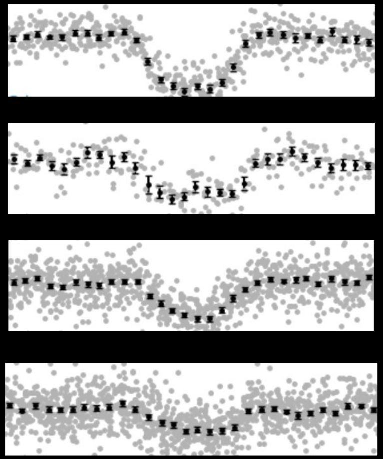


The Lazy-Exoplanet-Operations Vetter (LEO-Vetter)

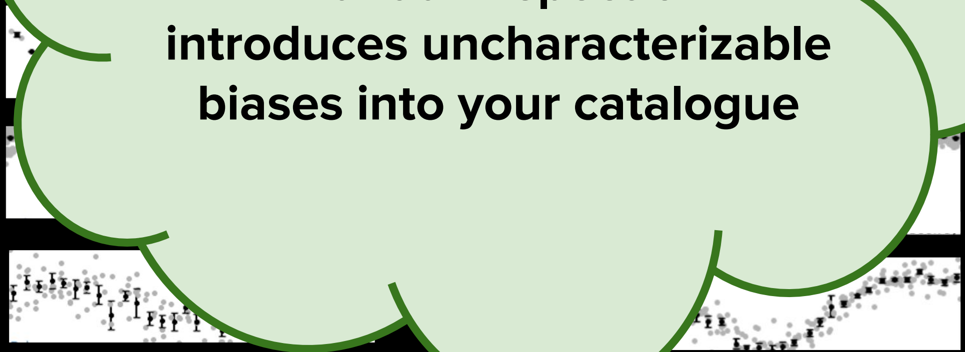
A fully automated vetting tool for identifying TESS planet ca... ...pired by the Kepler Robovetter (will be updated for Roman

github.com/mkunimoto/LEO-vetter

Planets



F



Key word:

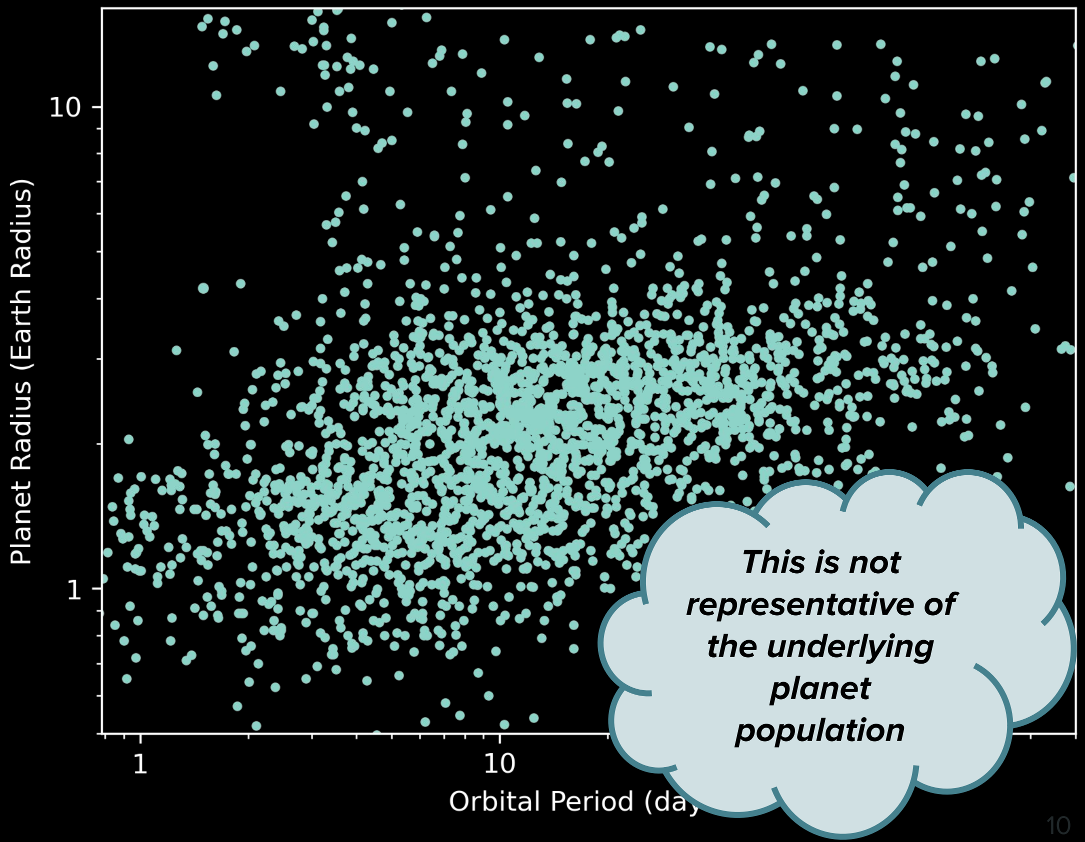
✨***automated***✨

Manual inspection introduces uncharacterizable biases into your catalogue

Step 1: Making your planet candidate catalogue

After running planet search + vetting...

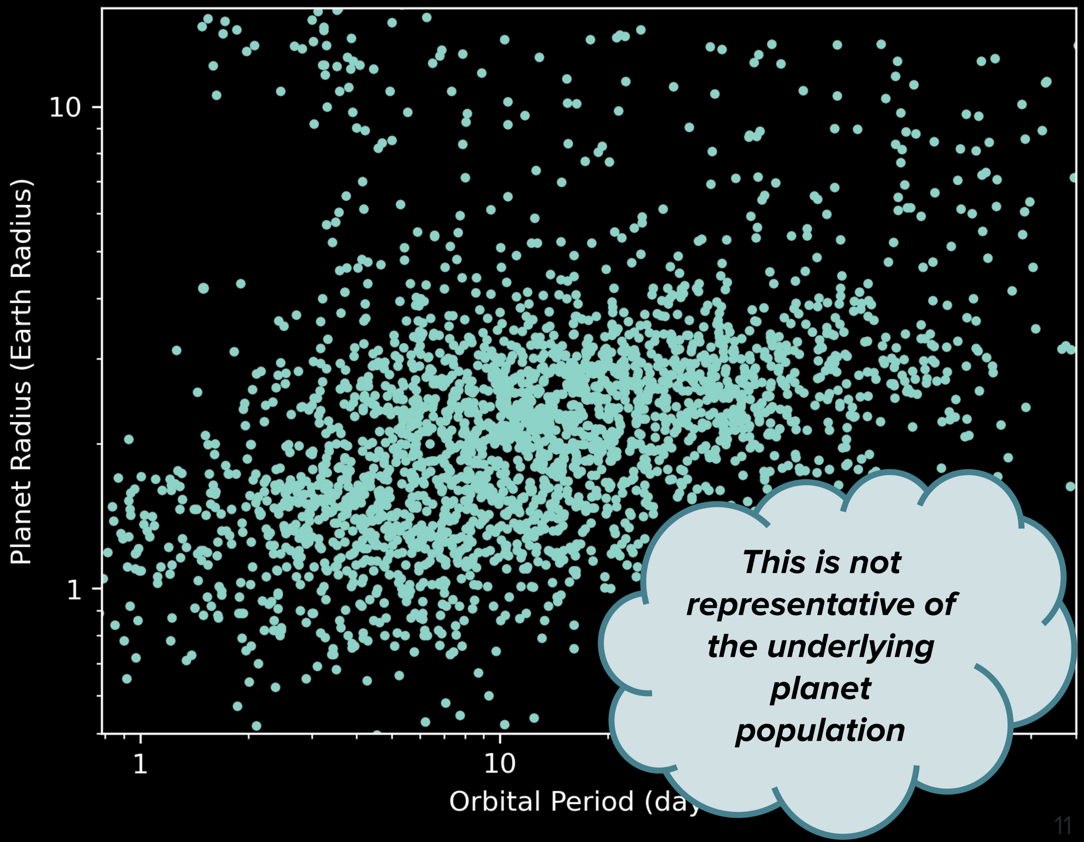
$$\frac{N_p}{N_s}$$ = *η (occurrence rate: number of planets per star)*



Step 1: Making your planet candidate catalogue

After running planet search + vetting... REMEMBER:

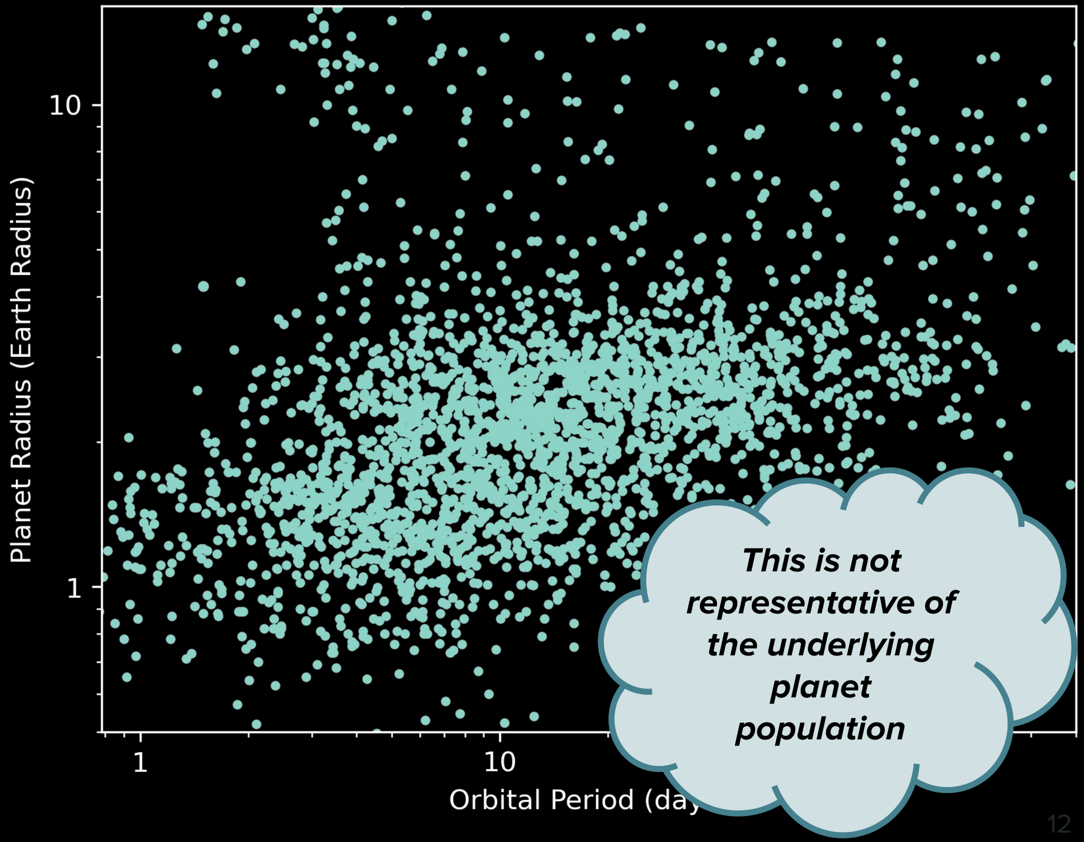
1) **Transit surveys are inherently *incomplete*. Not all planets will transit, and not all transiting planets will be detected.**



Step 1: Making your planet candidate catalogue

After running planet search + vetting... **REMEMBER:**

1) Transit surveys are inherently *incomplete*. Not all planets will transit, and not all transiting planets will be detected.
2) Transit surveys are inherently *unreliable*. Not all “planets” in your catalogue will actually be planets - some will be false alarms/false positives.



Step 2: Characterizing "Completeness"

Completeness quantifies the fraction of planets orbiting stars in our sample that we expect to have been detected by our transit survey.

Example: if we have 10% completeness, our catalogue will represent only 10% of all planets.

Three factors affecting completeness:

1) Transit geometry: *not all planets will actually transit*

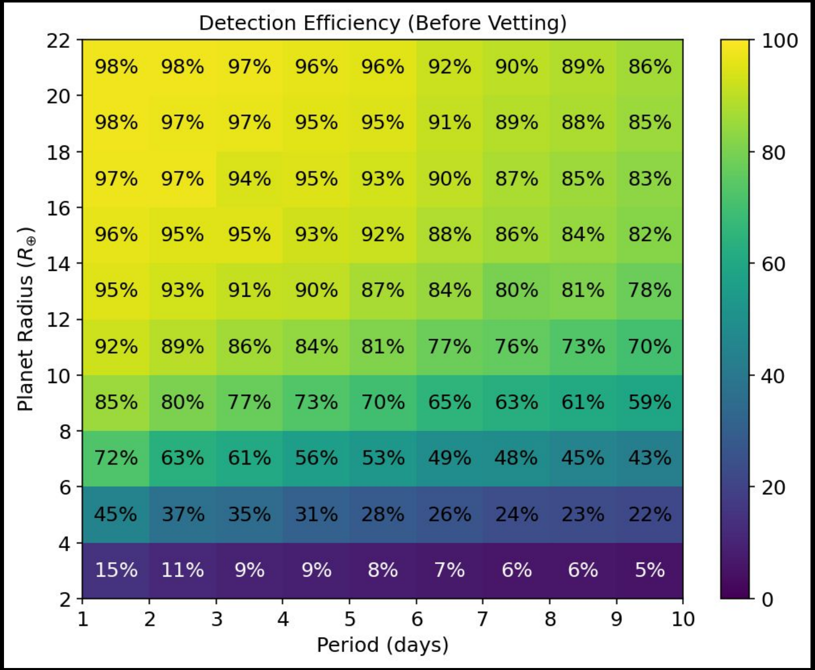
   $$p(\text{transit}) = (R_p + R_s)/a$$

2) Window function: *not all planets will transit while Roman is watching*

3) Detection efficiency: *not all planets with transits will actually be detected*

Step 2: Characterizing "Completeness"

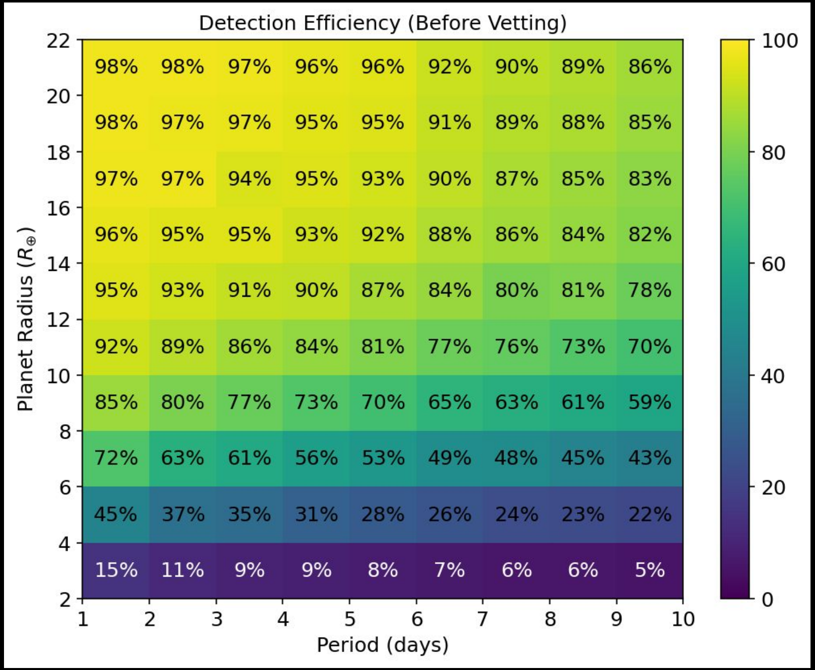
Most common method is to measure the window+detection probability of our planet search with injection/recovery tests

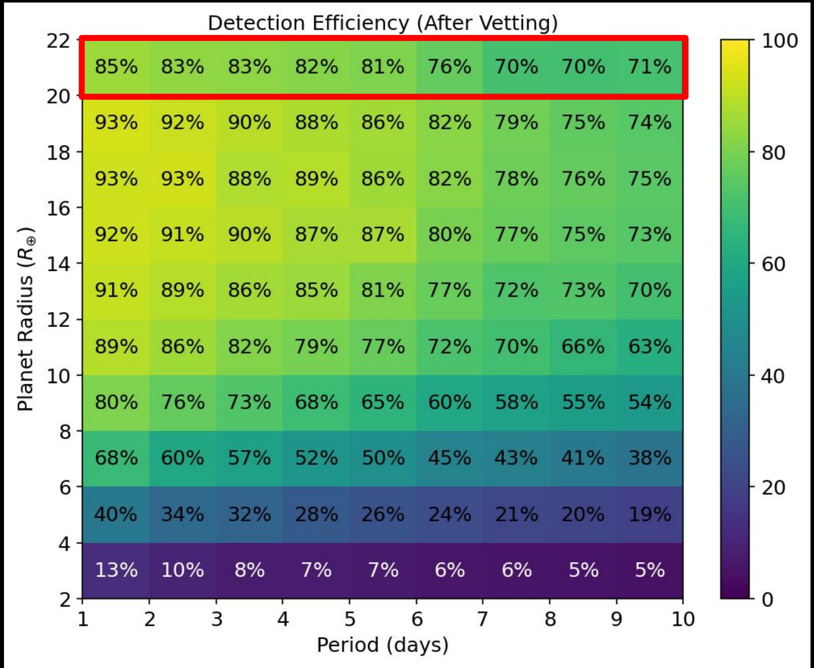


Each grid cell shows the fraction of injected (transiting) planets that were recovered

Step 2: Characterizing "Completeness"

Most common method is to measure the window+detection probability of our planet search with injection/recovery tests





Step 3: Characterizing "Reliability"

Reliability quantifies the fraction of objects in our catalogue that are actually planets

Example: if we have 80% reliability, then only 80% of our catalogue is planets (the other 20% is false alarms or false positives)

Two factors affecting reliability:

1) False alarm contamination: *some catalogue objects may be false alarms*

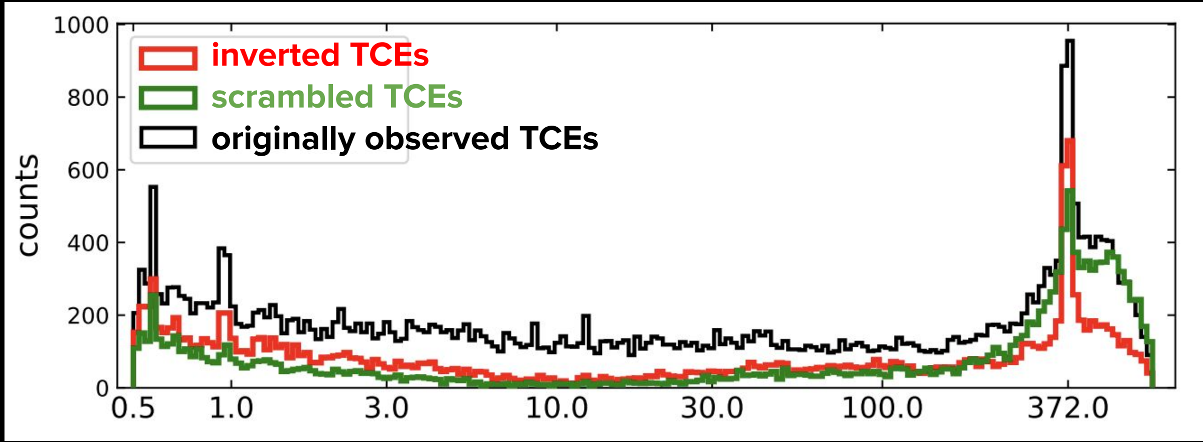
2) False positive contamination: *some catalogue objects may be false positives*

Step 3: Characterizing False Alarm Reliability

Most common method is to **simulate false alarms** and see how often your planet search+vetting pipeline marks them as "planets"

What has worked for Kepler and TESS: looking at events found after...

- inverting light curves
- scrambling light curves (e.g., re-ordering Kepler quarters, or TESS orbits)



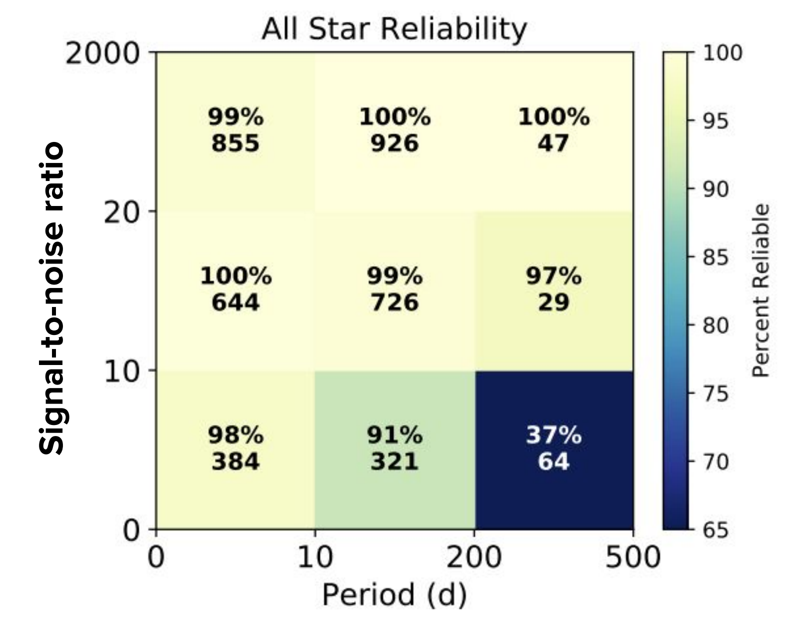
Kepler

Thompson et al. (2018)

Step 3: Characterizing False Alarm Reliability

Most common method is to simulate false alarms and see how often your planet search+vetting pipeline marks them as "planets"

This results in a population-level reliability



Kepler

Thompson et al. (2018)

Step 3: Characterizing False Positive Reliability

Most common method is to use a statistical validation software (e.g., TRICERATOPS; Giacalone et al. 2021) which gives a false positive probability for each candidate in your catalogue

How will this work for Roman? To be figured out :)

Step 4: Actually Calculating Occurrence Rates

To recap, here is what we have so far:

1) A stellar catalogue

2) A uniformly vetted list of planet candidates, where each candidate has:
   a) orbital periods, P
   b) planet radii, R_p
   c) false alarm and false positive reliabilities, R

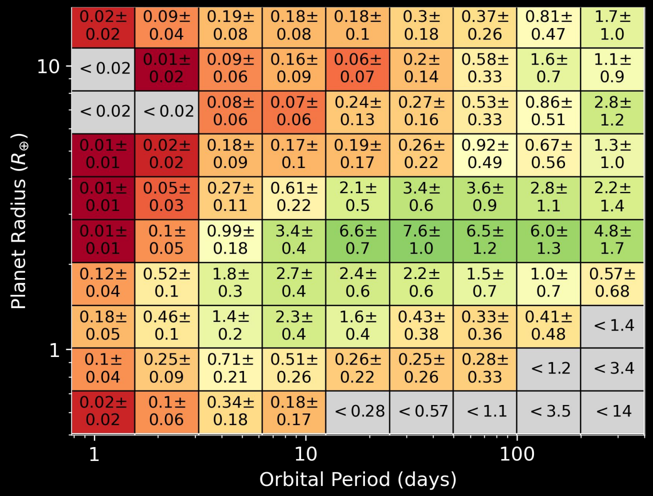
3) A measurement of completeness as a function of period and radius, $C(P, R_p)$

4) Passion for statistically robust occurrence rates :)

Step 4: Actually Calculating Occurrence Rates

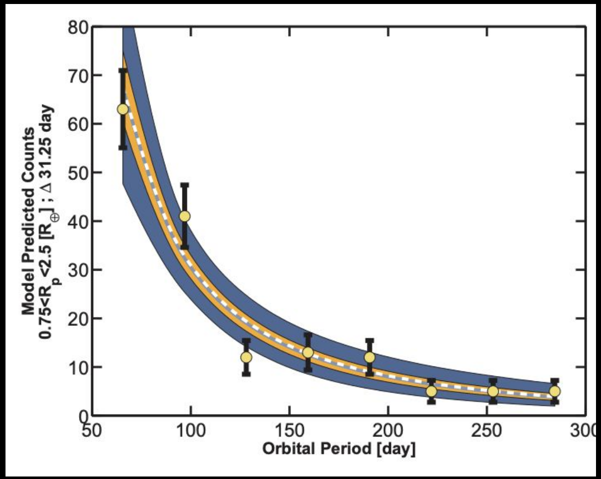
Now you need to decide how to actual model the underlying planet population

Option 1: "Grid-Based" Approach



Kunimoto & Matthews (2020)

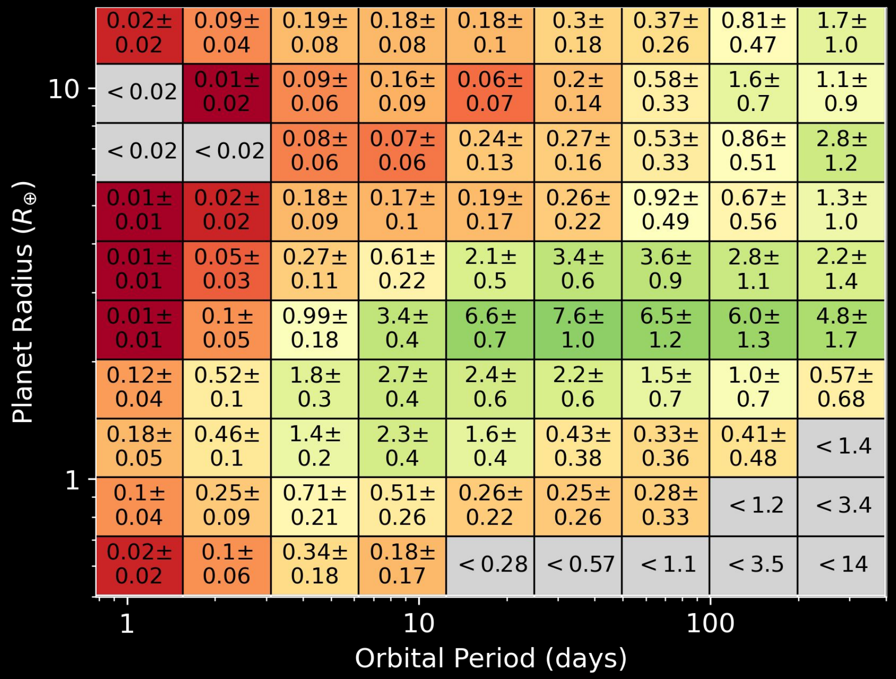
Option 2: "Parametric" Approach



Burke et al. (2015)

Grid-Based Approach

- **planet occurrence rate is averaged over a bin (across period/radius or log-period/log-radius)**
- **need to choose bins that are *small* enough that this is reasonable, but *big* enough that you actually have planets**



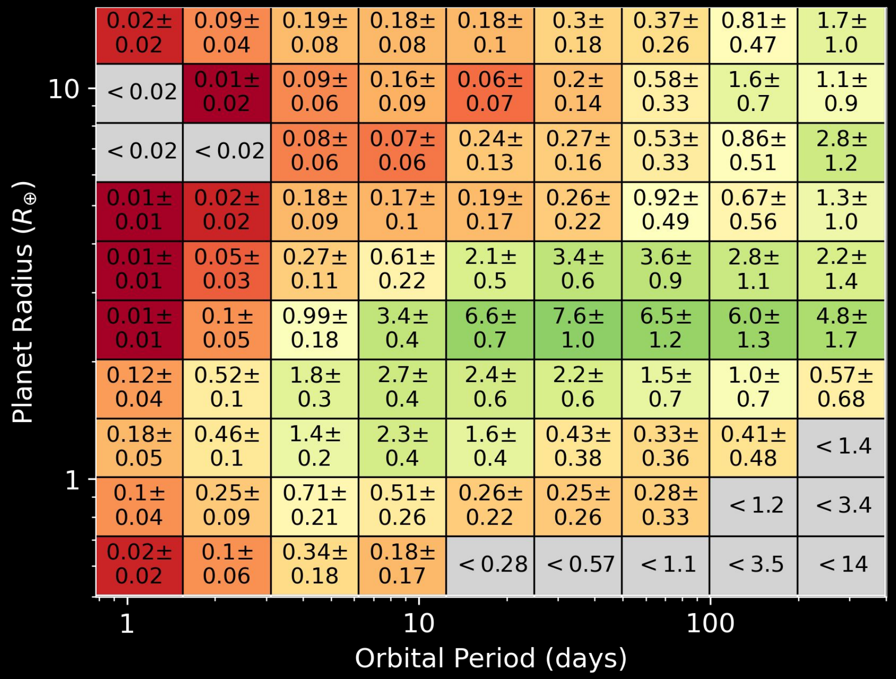
Grid-Based Approach

Example Methods:

- **Inverse Detection Efficiency (IDEM)**
  (Howard et al. 2012, Christiansen et al. 2015...)

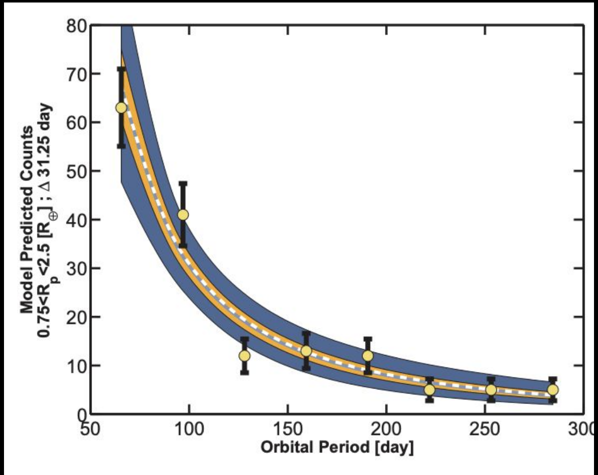
- **Forward modelling**
  (Hsu et al. 2019, Kunimoto & Matthews 2020...)

- **Poisson likelihood**
  (Foreman-Mackey et al. 2014, Cui et al. 2026...)

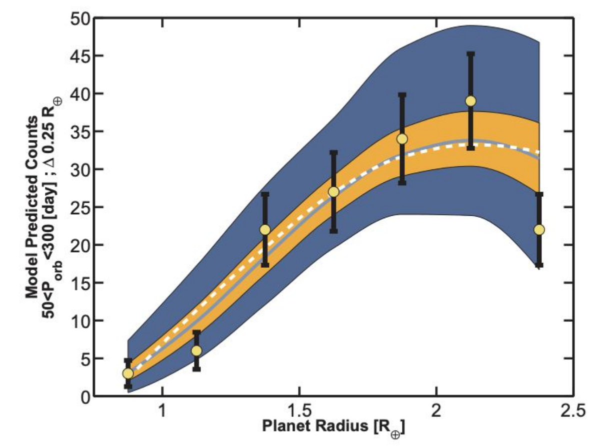


Parametric Approach

- planet occurrence rate is assumed to follow a specific function of period and radius
- common choice assume periods and radii are distributed according to power laws (or broken power laws)



Periods follow a power law



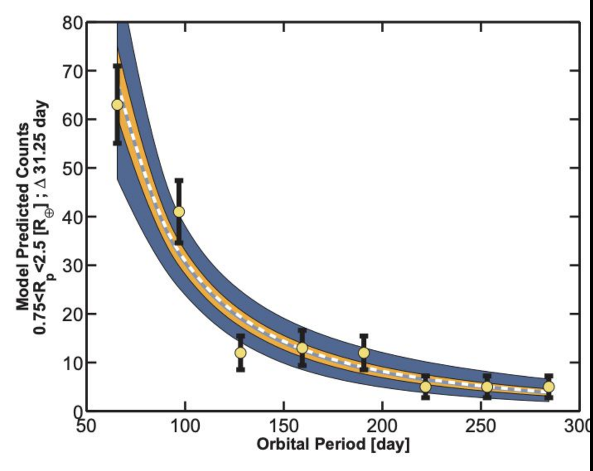
Radii follow a broken power law

Parametric Approach

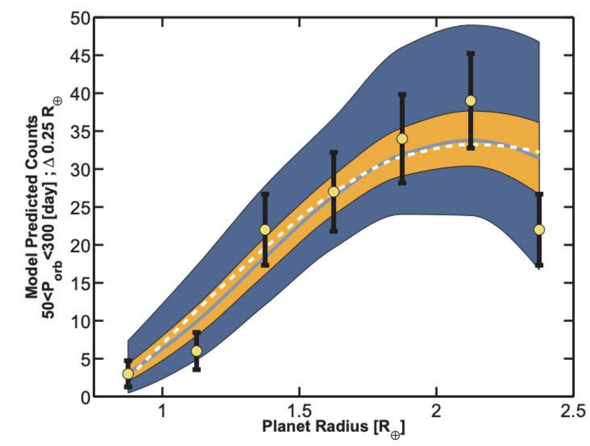
Example Methods:

- **Poisson likelihood method**
  (Youdin et al. 2011, Burke et al. 2015, Bryson et al. 2022...)

- **Forward modelling**
  (Mulders et al. 2018, Bryson et al. 2022...)



Periods follow a power law



Radii follow a broken power law

Strengths/Weaknesses of These Approaches

Grid-Based Approach	Parametric Approach (e.g., power laws)
(+) Captures small-scale variations in occurrence rates	(-) Can impose a model that is wrong; e.g., the radius valley is not a power law!
(-) Very sensitive to bin sizes	(+) Easy to integrate over to compute planet occurrence in any period-radius region

Step 5: Cool Things to Do With Transit Occurrence Rates

You could...

- see how occurrence rates change with orbital period!
- see how occurrence rates change with planet radius!
- see how occurrence rates depend on host star properties!
- extrapolate out to the habitable zone to compute $\eta_\oplus$!
- so much more...

Remember that occurrence rates are the ground-truth datasets needed to test planet formation and evolution theories

Conclusions

- calculating your own exoplanet occurrence rates involves:
  - compiling your own planet candidate catalogue
  - characterizing the completeness of your catalogue, to capture the fact that you will have missed some planets
  - characterizing the reliabilities of all your candidates, to capture the fact that your catalogue will be somewhat contaminated by FAs/FPs
  - computing grid-based or parametric occurrence rates

- huge scientific payoff: occurrence rates are the ground-truth datasets needed to test planet formation and evolution theories

- NASA's Science Interest Group #2 (SIG2: Exoplanet Demographics) is currently writing a "How To" Transit Occurrence Rate Guide - stay tuned!!