Statistics for Astronomers: Lecture 14, 2020.12.04

Prof. Sundar Srinivasan

IRyA/UNAM

Review

Hypothesis testing.

Two-sample Z- and t-tests.

Jupyter demos

1. ▸ Download this Jupyter notebook.
2. ▸ Navigate to Colaboratory.
3. Sign in
4. Click on "Upload" and upload the notebook you downloaded in step 1.

F-test

Data: two independent samples of sizes N_1, N_2 drawn from $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$.
Question: Is $\sigma_1 \neq \sigma_2$? Null hypothesis H_0: $\sigma_1 = \sigma_2$.

F-test

Data: two independent samples of sizes N_1, N_2 drawn from $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$.
Question: Is $\sigma_1 \neq \sigma_2$? Null hypothesis H_0: $\sigma_1 = \sigma_2$.

Recall: If $X \sim \mathcal{N}(\mu, \sigma^2)$, the sample variance $S^2 \sim \sigma^2 \dfrac{\chi^2(N-1)}{N-1} = \sigma^2 \chi^2_{\mathrm{red}}(N-1)$.

F-test

Data: two independent samples of sizes N_1, N_2 drawn from $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$.

Question: Is $\sigma_1 \neq \sigma_2$? Null hypothesis H_0: $\sigma_1 = \sigma_2$.

Recall: If $X \sim \mathcal{N}(\mu, \sigma^2)$, the sample variance $S^2 \sim \sigma^2 \dfrac{\chi^2(N-1)}{N-1} = \sigma^2 \chi^2_{\text{red}}(N-1)$.

$$\text{Let } S_2^2 > S_1^2. \text{ Define the ratio } F = \frac{S_2^2}{S_1^2} = \frac{\sigma_2^2}{\sigma_1^2} \frac{\chi^2_{\text{red}}(N_2-1)}{\chi^2_{\text{red}}(N_1-1)}.$$

$$\text{Under } H_0,\ F = \frac{\chi^2_{\text{red}}(N_2-1)}{\chi^2_{\text{red}}(N_1-1)} \qquad \text{by design, } F > 1.$$

F-test

Data: two independent samples of sizes N_1, N_2 drawn from $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$.
Question: Is $\sigma_1 \neq \sigma_2$? Null hypothesis H_0: $\sigma_1 = \sigma_2$.

Recall: If $X \sim \mathcal{N}(\mu, \sigma^2)$, the sample variance $S^2 \sim \sigma^2 \dfrac{\chi^2(N-1)}{N-1} = \sigma^2 \chi^2_{\rm red}(N-1)$.

Let $S_2^2 > S_1^2$. Define the ratio $F = \dfrac{S_2^2}{S_1^2} = \dfrac{\sigma_2^2}{\sigma_1^2} \dfrac{\chi^2_{\rm red}(N_2-1)}{\chi^2_{\rm red}(N_1-1)}$.

Under H_0, $F = \dfrac{\chi^2_{\rm red}(N_2-1)}{\chi^2_{\rm red}(N_1-1)}$ by design, $F > 1$.

This quantity, the ratio of two reduced χ^2 variables, has an F distribution with parameters $(\nu_{\rm numerator}, \nu_{\rm denominator}) = (N_2 - 1, N_1 - 1)$.

`Python` implementation: `scipy.stats.f`. Also see Sec. 4.7.6 in the AstroML book.

F-test

Data: two independent samples of sizes N_1, N_2 drawn from $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$.
Question: Is $\sigma_1 \neq \sigma_2$? Null hypothesis H_0: $\sigma_1 = \sigma_2$.

Recall: If $X \sim \mathcal{N}(\mu, \sigma^2)$, the sample variance $S^2 \sim \sigma^2 \dfrac{\chi^2(N-1)}{N-1} = \sigma^2 \chi^2_{\rm red}(N-1)$.

Let $S_2^2 > S_1^2$. Define the ratio $F = \dfrac{S_2^2}{S_1^2} = \dfrac{\sigma_2^2}{\sigma_1^2} \dfrac{\chi^2_{\rm red}(N_2-1)}{\chi^2_{\rm red}(N_1-1)}$.

Under H_0, $F = \dfrac{\chi^2_{\rm red}(N_2-1)}{\chi^2_{\rm red}(N_1-1)}$ by design, $F > 1$.

This quantity, the ratio of two reduced χ^2 variables, has an F distribution with parameters $(\nu_{\rm numerator}, \nu_{\rm denominator}) = (N_2 - 1, N_1 - 1)$.

`Python` implementation: `scipy.stats.f`. Also see Sec. 4.7.6 in the AstroML book.

Example (from Barlow): see Jupyter notebook.

F-test

Data: two independent samples of sizes N_1, N_2 drawn from $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$.
Question: Is $\sigma_1 \neq \sigma_2$? Null hypothesis H_0: $\sigma_1 = \sigma_2$.

Recall: If $X \sim \mathcal{N}(\mu, \sigma^2)$, the sample variance $S^2 \sim \sigma^2 \dfrac{\chi^2(N-1)}{N-1} = \sigma^2 \chi^2_{\rm red}(N-1)$.

Let $S_2^2 > S_1^2$. Define the ratio $F = \dfrac{S_2^2}{S_1^2} = \dfrac{\sigma_2^2}{\sigma_1^2} \dfrac{\chi^2_{\rm red}(N_2 - 1)}{\chi^2_{\rm red}(N_1 - 1)}$.

Under H_0, $F = \dfrac{\chi^2_{\rm red}(N_2 - 1)}{\chi^2_{\rm red}(N_1 - 1)}$ by design, $F > 1$.

This quantity, the ratio of two reduced χ^2 variables, has an F distribution with parameters $(\nu_{\rm numerator}, \nu_{\rm denominator}) = (N_2 - 1, N_1 - 1)$.

`Python` implementation: `scipy.stats.f`. Also see Sec. 4.7.6 in the AstroML book.

Example (from Barlow): see Jupyter notebook.

For you to solve:

Given $N_1 = 10$, $N_2 = 7$, and $S_2^2 = (1+\lambda)S_1^2$, with $\lambda > 0$.
What is λ if the F-test rejects the null hypothesis with 99.5% confidence?

Application of F-test: detecting time-variability

Data: fluxes f_i ($i = 1, \cdots, N$) for a source observed at times t_i.
Question: Are the data consistent with a variable source?

Application of F-test: detecting time-variability

Data: fluxes f_i ($i = 1, \cdots, N$) for a source observed at times t_i.

Question: Are the data consistent with a variable source?

- H_0: No. Variability solely due to measurement error.
- H_A: Observed time dependence is combination of intrinsic variability and measurement error.

Application of F-test: detecting time-variability

Data: fluxes f_i ($i = 1, \cdots, N$) for a source observed at times t_i.

Question: Are the data consistent with a variable source?

H_0: No. Variability solely due to measurement error.

H_A: Observed time dependence is combination of intrinsic variability and measurement error.

Model: $f_i = f_{\mathrm{true},i} + f_{\mathrm{meas},i}$, with $f_{\mathrm{meas},i} \sim \mathcal{N}(0, \sigma^2)$.

$\mathrm{Var}[f_i] = \mathrm{Var}[f_{\mathrm{true},i}] + \mathrm{Var}[f_{\mathrm{meas},i}] = \mathrm{Var}[f_{\mathrm{true},i}] + \sigma^2$.

Under H_0, $\mathrm{Var}[f_i] = \sigma^2$. Under H_A, $\mathrm{Var}[f_i] > \sigma^2$.

Application of F-test: detecting time-variability

Data: fluxes f_i ($i = 1, \cdots, N$) for a source observed at times t_i.

Question: Are the data consistent with a variable source?

H_0: No. Variability solely due to measurement error.

H_A: Observed time dependence is combination of intrinsic variability and measurement error.

Model: $f_i = f_{\mathrm{true},i} + f_{\mathrm{meas},i}$, with $f_{\mathrm{meas},i} \sim \mathcal{N}(0, \sigma^2)$.

$$\mathrm{Var}[f_i] = \mathrm{Var}[f_{\mathrm{true},i}] + \mathrm{Var}[f_{\mathrm{meas},i}] = \mathrm{Var}[f_{\mathrm{true},i}] + \sigma^2.$$

Under H_0, $\mathrm{Var}[f_i] = \sigma^2$. Under H_A, $\mathrm{Var}[f_i] > \sigma^2$.

Perform F-test on the data! Define $F = \dfrac{\text{Sample variance for data } f_i}{\text{Sample variance of measurement error}}$

See Jupyter notebook for illustration.

Nonparametric tests

References

- This series of videos.
- Barlow
- Wall & Jenkins
- AstroML

Nonparametric tests

How they work: typically by sorting/ordering (ranking) data values.

Nonparametric tests

How they work: typically by sorting/ordering (ranking) data values.

Some advantages:

1. Very little assumed about the data and underlying distributions (typically: continuity).
   Great for when we don't know!
2. Ideal for small sample sizes, poorly-defined distributions, data with outliers.
   difficult to determine parametric model for such data.
3. Also applicable when data is non-numeric (*e.g.*, classifications).
4. Can work with samples drawn from several different populations.

Major drawback: some tests require binning of data.

Nonparametric tests

How they work: typically by sorting/ordering (ranking) data values.

Some advantages:

1. Very little assumed about the data and underlying distributions (typically: continuity).
   Great for when we don't know!
2. Ideal for small sample sizes, poorly-defined distributions, data with outliers.
   difficult to determine parametric model for such data.
3. Also applicable when data is non-numeric (*e.g.*, classifications).
4. Can work with samples drawn from several different populations.

Major drawback: some tests require binning of data.

We've already used nonparametric inference!

Location estimate: sort data, pick central value – median.
Scale estimate: sort data, compute interquartile range.
Distribution estimate: EDF. Used this for interval estimates (CI).
The bootstrap procedure.

Kolmogorov-Smirnov test

One-sample test (implementation: `scipy.stats.kstest`):

Given: sample of size N. Question: Is the sample drawn from a particular distribution?

H_0: yes. H_A: No (two-sided) or *greater*/*less* than CDF of distribution (one-sided).

Two-sample test (implementation: `scipy.stats.ks_2samp`):

Given: samples of sizes N_1, N_2. Question: Are samples drawn from the same distribution?

H_0: yes. H_A: No (two-sided).

Kolmogorov-Smirnov test

One-sample test (implementation: `scipy.stats.kstest`):

Given: sample of size N. Question: Is the sample drawn from a particular distribution?

H_0: yes. H_A: No (two-sided) or *greater*/*less* than CDF of distribution (one-sided).

Two-sample test (implementation: `scipy.stats.ks_2samp`):

Given: samples of sizes N_1, N_2. Question: Are samples drawn from the same distribution?

H_0: yes. H_A: No (two-sided).

KS statistic: maximum distance between CDFs that are being compared. One-sample case:

$D_{\rm KS} = \max\left|CDF_{\rm model}(x) - ECDF(x)\right|$ (two-sided). $D_{\rm KS} = \max\left(CDF_{\rm model}(x) - ECDF(x)\right)$ (one-sided).

Kolmogorov-Smirnov test

One-sample test (implementation: `scipy.stats.kstest`):

Given: sample of size N. Question: Is the sample drawn from a particular distribution?

H_0: yes. H_A: No (two-sided) or *greater*/*less* than CDF of distribution (one-sided).

Two-sample test (implementation: `scipy.stats.ks_2samp`):

Given: samples of sizes N_1, N_2. Question: Are samples drawn from the same distribution?

H_0: yes. H_A: No (two-sided).

KS statistic: maximum distance between CDFs that are being compared. One-sample case:

$D_{\mathrm{KS}} = \max\left|CDF_{\mathrm{model}}(x) - ECDF(x)\right|$ (two-sided). $D_{\mathrm{KS}} = \max\left(CDF_{\mathrm{model}}(x) - ECDF(x)\right)$ (one-sided).

Let's go to the Jupyter notebook again...

Kolmogorov-Smirnov test

One-sample test (implementation: `scipy.stats.kstest`):

Given: sample of size N. Question: Is the sample drawn from a particular distribution?

H_0: yes. H_A: No (two-sided) or *greater*/*less* than CDF of distribution (one-sided).

Two-sample test (implementation: `scipy.stats.ks_2samp`):

Given: samples of sizes N_1, N_2. Question: Are samples drawn from the same distribution?

H_0: yes. H_A: No (two-sided).

KS statistic: maximum distance between CDFs that are being compared. One-sample case:

$D_{\mathrm{KS}} = \max\left|CDF_{\mathrm{model}}(x) - ECDF(x)\right|$ (two-sided). $D_{\mathrm{KS}} = \max\left(CDF_{\mathrm{model}}(x) - ECDF(x)\right)$ (one-sided).

Let's go to the Jupyter notebook again...

Advantages: No binning! Small samples! More powerful for intermediate-size samples! Can also work as a one-tailed test (see `sscipy.stats.kstest`).

Disadvantages: Not sensitive to differences in the tails. Doesn't care about $\#dof$.

Kolmogorov-Smirnov test

One-sample test (implementation: `scipy.stats.kstest`):

Given: sample of size N. Question: Is the sample drawn from a particular distribution?

H_0: yes. H_A: No (two-sided) or *greater*/*less* than CDF of distribution (one-sided).

Two-sample test (implementation: `scipy.stats.ks_2samp`):

Given: samples of sizes N_1, N_2. Question: Are samples drawn from the same distribution?

H_0: yes. H_A: No (two-sided).

KS statistic: maximum distance between CDFs that are being compared. One-sample case:

$D_{\mathrm{KS}} = \max\left|CDF_{\mathrm{model}}(x) - ECDF(x)\right|$ (two-sided). $D_{\mathrm{KS}} = \max\left(CDF_{\mathrm{model}}(x) - ECDF(x)\right)$ (one-sided).

Let's go to the Jupyter notebook again...

Advantages: No binning! Small samples! More powerful for intermediate-size samples! Can also work as a one-tailed test (see `sscipy.stats.kstest`).

Disadvantages: Not sensitive to differences in the tails. Doesn't care about $\#dof$.

Beware the KS test!