HERA Memo #130 V3
Received: 31 July 2024

Excess Variance of Visibilities and the Z-squared Statistic

Steven G. Murray

July 31, 2024

1 Excess Variance of Visibilities and the $|Z|^2$ statistic

Steven Murray and the H6C Analysis Team. 2nd of May, 2024.

1.1 Introduction and Motivation

In this memo, we describe the statistics of *ideal* visibilities that can be inferred when multiple redundant visibilities are measured. These redundancies could be due to baseline redundancy, or LST-redundancy. The key point is that we have in hand some visibilities, $V \in \{V_0, V_1, \cdots, V_n\}$ which we *expect* to be drawn from the same underlying distribution. A key question in this case is whether they *are* indeed drawn from the same distribution, and if not, which visibilities are outliers, and at what level?

While many measures of "outlierness" could be imagined, and these measures could be exanded if you simply want to determine "badness" (i.e. not outliers w.r.t. the data in hand, but rather incompatibility with the underyling priors we have for the data, e.g. spectral smoothness), we will restrict ourselves here to those measures that are uniquely available when you have many visibilities stacked together (e.g. at the LST-stacking stage). Thus, we will not appeal to measures that account for spectral non-smoothness, only those that show that a given visibility on a given baseline and channel is an outlier with respect to the other supposedly redundant measurements for the same baseline and channel.

Since visibilities—at a given baseline, channel, and time—are expected to be drawn from Gaussian distributions, it turns out that that the full information about the distribution is contained in the first two moments – the mean and variance. Thus, if we have a handle on what those two values are for the data, we can rescale the data such that all values *should* arise from the same distribution. This re-scaled statistic is the Z-score, and since it is a mean-zero Gaussian, all the information we care about can be expressed through the second moment, via functions of $|Z|^2$.

To be clear, we are saying that all useful measures of the outlierness of some bit of data should be representable by a *function* of $|Z|^2$ for that particular bit of data. Each (baseline, channel, time) will have an associated value for $|Z|^2$. The function $f(|Z|^2) = |Z|^2$ is a simple function that specifies how much of an outlier a particular (baseline, channel, time) is with respect to its peers. But one could also imagine taking other functions that might highlight certain aspects of the data. Other simple such functions might be to take the mean of $|Z|^2$ over different subsets of data, to find an overall outlierness for that subset (e.g. over all channels for a particular LST, or a subset of baselines).

This memo will thus describe the statistics of the $|Z|^2$ metric – its expected distribution, and the distribution of averages over multiple measurements of $|Z|^2$, as well as relating this metric to the "excess variance" metric described in HERA Memo 123.

1.2 Notation and Identities

Throughout we will use an overbar (e.g. $\bar{x}$) to represent a sample mean (whether a weighted mean or not), and angle brackets (e.g. $\langle X \rangle$) to denote an *ensemble average*. We will use $\mathcal{R}(x)$ and $\mathcal{I}(x)$ to denote the real and imaginary components of x respectively, and $\mathcal{K}(x)$ to represent an arbitrary (real *or* imaginary) component (when $\mathcal{K}$ is used in an equation, it is understood to represent the *same* component everywhere in the equation).

Throughout, we will make much use of the Gamma distribution, $\Gamma(\alpha, \beta)$. A few key properties of this distribution will be useful:

1. Γ is closely related to the central chi-square distribution: $\Gamma(k/2, k/2) \equiv \chi^2_k$.
2. The mean of $\Gamma(\alpha, \beta)$ is $\mu = \alpha/\beta$ and its variance is α/β^2.
3. The sum of N gamma-variates G_i, where $G_i \sim \Gamma(\alpha_i, \beta)$ has a sampling distribution $\sum_i G_i \sim \Gamma(\sum_i \alpha_i, \beta)$.
4. The distribution of a gamma-variate $G \sim \Gamma(\alpha, \beta)$ scaled by a constant, a is $aG \sim \Gamma(\alpha, \beta/a)$.
5. Combining points 3 and 4, the mean of N gamma-variates is $\bar{G} \sim \Gamma(\sum \alpha_i, N\beta)$.

1.3 Basic Definitions

Let V_i be a complex-valued visibility, measured for some baseline, channel and time. Throughout, we will assume that measurements for different baselines, channels and times are uncorrelated, and we will thus neglect them.

The complex visibility $V_i = \mathcal{R}(V)_i + i\mathcal{I}(V)_i$ is drawn from a complex gaussian distribution, for which

$$\mathcal{K}(V)_i \sim \mathcal{N}(\mathcal{K}(\mu), \sigma^2/2), \tag{1}$$

$$\tag{2}$$

That is, each of the components (real and imaginary) may have different means, but are expected to have the same variance, which depends on the sky temperature, the channel width and the integration time. Note that the variance of each component is *half* the variance of the full complex visibility.

We would like to form an estimate of a "Z-score" — i.e. a rescaled random variable that is drawn from a standard normal distribution. To do this, we need an estimate of the mean and the variance, σ^2. The latter is easy to come by: the autocorrelations measured by the instrument give a good estimate of the variance, and these estimates are essentially uncorrelated with the visibilities themselves. The mean is not quite so simple. If we *did* have a theoretical prediction of the mean – $\mu = \mathcal{R}(\mu) + i\mathcal{I}(\mu)$ – we could form:

$$Z'_i = \frac{V_i - \mu}{\sigma/\sqrt{2}}$$

for which each component (real and imaginary) has a standard normal distribution. In this case, we would have $\mathcal{K}(Z'_i)^2 \sim \chi^2_1$ and $|Z'_i|^2 \sim \chi^2_2$.

However, we have no reliable way of theoretically predicting μ. We can instead calculate an estimate by taking the average of other data we have in hand that is supposed to be drawn from the same distribution as V_i (i.e. from redundant baselines at the same LST and channel). Let us assume that we have N such visibilities, and also that each may be the average of n_i redundant visibilities themselves (for example, perhaps we do a two-stage averaging, where first we average over redundant baselines, then we average over redundant LSTs—in this case, at the LST-stacking stage, we have redundant visibililities that have already been averaged with potentially different nsamples).

We define the weighted mean visibility, which is an estimate of μ, as

$$\hat{\mu} = \bar{V} \equiv \left(\sum_i^N n_i\right)^{-1} \sum_i^N n_i V_i = \frac{\sum_i^N n_i V_i}{M},$$

with $M = \sum_i^N n_i$ the total number of samples in the mean.

We now define a Z-score as

$$Z_i \equiv \sqrt{\frac{2n_i}{\sigma^2} \frac{M}{M - n_i}} (V_i - \bar{V}),$$

where we again stress that $\bar{V}$ is itself a random variable correlated with V_i, but that σ is uncorrelated with V_i as it is estimated from the autos (for our purposes here, we will treat it as non-random).

The reason for the scaling factor involving n_i and M in our *definition* of Z_i is that with this factor the variance of a component $\mathcal{K}(Z)_i$ becomes unity (recall that $\mathcal{K}(V)_i$ itself is a Gaussian variable with variance σ^2/n_i). More importantly, the distribution of Z_i is then found to be independent of i. We will discuss this further below.

We also recall the definition for *excess variance* as outlined in HERA Memo 123:

$$\mathcal{K}(\gamma) \equiv S^2/\langle S^2\rangle = \frac{\sum_i^N 2n_i \mathcal{K}(V_i - \bar{V})^2}{\sigma^2(N-1)} \tag{3}$$

where we note that this definition specifically concerns only the real/imag part of the visibility, and is defined in such a way that the mean of the distribution of γ is unity. We note that γ is a single metric describing the combined properties of a set of visibilities – the same set that form the mean. In contrast, Z_i is defined for a single visibility in the set (but requires defining the full set in order to calculate it).

The square of this quantity – whether $\mathcal{K}(Z)_i^2$ or $|Z|_i^2$ – forms the basis of all other metrics considered here.

We can write the excess variance as a function of Z_i:

$$\mathcal{K}(\gamma) = \sum_i^N \frac{M - n_i}{M(N-1)} \mathcal{K}(Z)_i^2.$$

Here, we further define the *excess absolute variance*:

$$|\gamma| \equiv \frac{\sum_i^N n_i |V_i - \bar{V}|^2}{\sigma^2 (N-1)}, \tag{4}$$

where we note that we have removed the factor of two from the numerator (w.r.t. the component-excess variance), as this will yield a distributional mean of unity (which is more in line with our label of 'excess variance'). See below.

We define three more metrics – all of them (weighted) means of Z_i^2 over different combinations of data. We will write each of them below as the mean over $|Z|_i^2$, keeping in mind that each of them can be defined also for a single real/imaginary component.

We define the mean $|Z|^2$ *over independent data* as

$$|\bar{Z}|^2 \equiv \frac{1}{Q} \sum_j^Q |Z|_j^2.$$

where Q is the total number of *independent* measurements of Z_i (this would generally be the excess variance measured for different non-redundant baselines or channels).

We also define the mean over the $|Z|^2$ that *form the redundant set*:

$$\zeta^2 \equiv \frac{1}{N} \sum_i^N \frac{M - n_i}{M} |Z|_i^2,$$

where we note the pre-factor.

Finally, we define the mean over ζ^2 for different non-redundant sets:

$$\bar{\zeta^2} \equiv \frac{1}{\sum_j N_j} \sum_j^Q N_j \zeta_j^2.$$

Note that we don't define a statistic that involves a *partial* sum over the redundant set – there is no analytic form of the distribution for this case, to our knowledge.

1.4 Statistics

1.4.1 Of the Z-score

$\mathcal{K}(Z_i)$ is simply a Gaussian random variable with mean zero and variance $(M - n_i)/M$. To show this, we first note that it is a sum of Gaussian variables (V_i and $\bar{V}$), which thus must be Gaussian itself. Note also that the expectation value is zero since $\langle V_i - \bar{V} = \langle \sum V_{j \neq i} \rangle + \langle V_i - V_i \rangle = 0$. Finally, the variance may be calculated by considering a single component (e.g. real) and seeing:

$$
\begin{aligned}
\mathrm{Var}\left[\mathcal{K}(V_i - \bar{V})\right] &= \mathrm{Var}(M^{-1}\sum_j^M \mathcal{K}(V)_j + \mathrm{Var}\left[\mathcal{K}(V)_i\right] - 2\mathrm{Cov}\left[\mathcal{K}(V)_i, \frac{n_i \mathcal{K}(V)_i}{M}\right] && (5)\\
&= \frac{\sigma^2}{2M} + \frac{\sigma^2}{2n_i} - 2\frac{n_i}{M}\frac{\sigma^2}{2n_i} && (6)\\
&= \frac{\sigma^2}{2n_i}\left(1 - \frac{n_i}{M}\right) && (7)\\
&= \frac{\sigma^2}{2n_i}\frac{M - n_i}{M}. && (8)
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
\mathrm{Var}(\mathcal{K}(Z_i)) &= \frac{2n_i}{\sigma^2}\frac{M}{M - n_i}\frac{\sigma^2}{2n_i}\frac{M - n_i}{M} && (9)\\
&= 1 && (10)
\end{aligned}
$$

1.4.2 Of Z_i^2

We can express $\mathcal{K}(Z_i)^2$ as

$$\mathcal{K}(Z_i)^2 = X^2,$$

where X is a standard normal variate. Thus $\mathcal{K}(Z_i)^2$ has the distribution

$$\mathcal{K}(Z_i)^2 \sim \chi_1^2 = \Gamma\left(\frac{1}{2}, \frac{1}{2}\right).$$

This has mean 1 and variance 2 (see property (2) above).

Since $|Z|^2 = \mathcal{R}(Z)^2 + \mathcal{I}(Z)^2$, we have

$$|Z|^2 \sim \Gamma\left(1, \frac{1}{2}\right)$$

1.4.3 Of $\bar{Z}^2$

Here we have a sum of gamma variables:

$$\overline{\mathcal{K}(Z)^2} \equiv \frac{1}{Q}\sum_j^Q \Psi_j \quad \Psi_j \sim \Gamma\left(\frac{1}{2}, \frac{1}{2}\right) \tag{11}$$

Using property (5) above we have

$$\overline{\mathcal{K}(Z)^2} \sim \Gamma\left(\frac{Q}{2}, \frac{Q}{2}\right).$$

In the same way as previous statistics, the absolute visibility squared is

$$\overline{|Z|^2} \sim \Gamma\left(Q, \frac{Q}{2}\right).$$

1.4.4 Of the Excess Variance

In HERA Memo 123 we showed that the distribution of $\mathcal{K}(\gamma)$ is a Gamma distribution:

$$\mathcal{K}(\gamma) \sim \Gamma\left(\frac{N-1}{2}, \frac{N-1}{2}\right).$$

Since

$$|\gamma| \equiv \frac{\mathcal{R}(\gamma) + \mathcal{I}(\gamma)}{2},$$

we can use property (5) to obtain

$$|\gamma| \sim \Gamma\left(N-1, N-1\right),$$

which also has unity mean.

1.4.5 Of ζ^2

Here we note that for $\mathcal{K}(\zeta)^2$ we have

$$\mathcal{K}(\zeta)^2 \equiv \frac{1}{N}\sum_i^N \mathcal{K}(Z)_i^2 = \frac{1}{N}\sum_i^N \frac{2n_i}{\sigma^2}\mathcal{K}(V_i - \bar{V})^2 \tag{12}$$

$$= \frac{N-1}{N}\mathcal{K}(\gamma) \tag{13}$$

Thus, the distribution of $\mathcal{K}(\zeta)^2$ is

$$\mathcal{K}(\zeta)^2 \sim \Gamma\left(\frac{N-1}{2}, \frac{N}{2}\right).$$

As in previous statistics, the absolute value is simple:

$$|\zeta|^2 \sim \Gamma\left(N-1, \frac{N}{2}\right)$$

1.4.6 Of $\overline{\zeta^2}$

This is simply the mean of Q variates of ζ^2. Setting $P = \sum_j^Q N_j$, we have

$$\overline{\mathcal{K}(\zeta)^2} \sim \Gamma\left(\frac{P-Q}{2}, \frac{P}{2}\right),$$

and

$$\overline{|\zeta|^2} \sim \Gamma\left(P-Q, \frac{P}{2}\right).$$

1.4.7 Conditional Distribution

In the process of trying to determine if a particular datum V_i is "bad" or not, a more powerful statistic than its raw Z^2 score (which depends on itself) is its *conditional* Z, i.e. the expected distribution of Z_i given that all the other data are known.

This is simply a Gaussian distribution with

$$\text{mean} = -\sqrt{\frac{2n_i}{\sigma^2}} \frac{\sum_{j\neq i} n_j V_j}{M - n_i}, \tag{14}$$

$$\text{Var} = 2. \tag{15}$$

Thus, forming a new per-visibility statistic:

$$Z_{C,i} \equiv \frac{Z_i + \sqrt{\frac{2n_i}{\sigma^2}} \frac{\sum_{j\neq i} n_j V_j}{M - n_i}}{\sqrt{2}},$$

we can determine the probability that the visibility V_i is part of the underlying Gaussian distribution as determined by the other variables.

Of course, what we have presented here are merely *distributions*, and they do not say anything specific about the data without a *prior model* for how the data could be driven away from the underlying distribution. That is, to *interpret* the statistics requires using a Bayesian framework. Nevertheless, for simplicity, placing a threshold on the retained data *in the space of the distributions here outlined* should be sufficient to retain decent data quality.

1.5 Demonstration Plots

Here, we simply demonstrate the veracity of each of the above distributions by drawing monte carlo samples and comparing.

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gamma
from hera_cal.lst_stack.stats import MixtureModel
```

The theoretical distributions

[3]:
```python
def zsquare(absolute: bool = True):
    return gamma(a=1 / (1 if absolute else 2), scale=2)

def mean_zsquare_over_redundant_set(n: int, absolute: bool = True):
    return gamma(a=(n - 1) / (1 if absolute else 2), scale=2 / n)

def mean_zsquare_over_independent_set(q: int, absolute: bool = True):
    return gamma(a=q / (1 if absolute else 2), scale=2 / q)

def mean_zsquare_over_redundant_and_independent_sets(nsets: int, ntot: int,␣
↪absolute: bool = True):
    return gamma(a=(ntot - nsets)/(1 if absolute else 2), scale=2 / ntot)

def excess_variance(n: int, absolute: bool = True):
    return gamma(a=(n - 1) / (1 if absolute else 2), scale=(1 if absolute else␣
↪2) / (n - 1))
```

MC Sample Generation

[112]:
```python
def get_vis(
    ndays: int = 10,
    ninds: int = 1,
    nvars: int = 200000,
    weighted: bool = False,
    allow_zeros: bool = True,
):
    rng = np.random.default_rng()

    if weighted:
        weights = rng.integers(low=0 if allow_zeros else 1, high=10,␣
↪size=(ndays, ninds, nvars))
        weights[:2] = 1  # Ensure at least one day is included
        #weights = weights * np.ones((ndays, ninds, nvars))
    else:
        weights = np.ones((ndays, ninds, nvars))

    scale = np.ones_like(weights).astype(float)
    scale[weights>0] = 1/np.sqrt(weights[weights>0])

    x = rng.normal(scale=scale) + 1j*rng.normal(scale=scale)

    return x, weights

def get_samples(
    ndays: int = 10,
    ninds: int = 1,
    nvars: int = 200000,
    absolute: bool = True,
```

```python
    mean_over_days: bool = False,
    mean_over_ind: bool = False,
    weighted: bool = True,
    allow_zeros: bool = True
):
    x, weights = get_vis(ndays, ninds, nvars, weighted, allow_zeros)

    avg = np.average(x, axis=0, weights=weights)
    m = np.sum(weights, axis=0)
    prefac = weights * m/(m - weights)
    if absolute:
        zsq = prefac * np.abs(x - avg)**2
    else:
        zsq = prefac * np.abs(x.real - avg.real)**2

    n_averaged = np.sum(weights > 0, axis=0)
    if mean_over_days:
        zsq = np.sum(zsq * (m-weights)/m, axis=0) / n_averaged

    if mean_over_ind:
        if not mean_over_days:
            zsq = np.mean(zsq, axis=1)[0]
        else:
            zsq = np.mean(zsq, axis=0)
            n_averaged = np.sum(n_averaged, axis=0)
    else:
        if ninds > 1:
            raise ValueError("only use ninds>1 if averaging over ind")

        n_averaged=n_averaged[0]

    return zsq.flatten(), n_averaged


def get_excess_variance(
    ndays: int = 10,
    nvars: int = 200000,
    absolute: bool = True,
    weighted: bool = True,
    allow_zeros: bool = True
):
    x, weights = get_vis(ndays, ninds, nvars, weighted, allow_zeros)
    avg = np.average(x, axis=0, weights=weights)

    var = np.average((x.real - avg.real)**2, axis=0, weights=weights)

    if absolute:
```

```
        yvar = np.average((x.imag - avg.imag)**2, axis=0, weights=weights)
        var += yvar

    excess_var = var * np.sum(weights, axis=0) / (ndays - 1)  # Bessel's␣
↪correction, true var is 1

    if absolute:
        excess_var /= 2

    n_averaged = np.sum(weights > 0, axis=0)

    return excess_var, n_averaged
```

[76]:
```
def make_comparison_plot(
    ndays: int = 10,
    nvars: int = 200000,
    ninds: int = 1,
    absolute: bool = True,
    weighted: bool = False,
    allow_zeros: bool = False,
    excess_variance: bool = False,
    mean_over_redset: bool = False,
    mean_over_indset: bool = False,
    lbl=None,
    color='C0'
):
    if allow_zeros and (mean_over_indset or not mean_over_redset):
        raise ValueError("allow_zeros = True only supported for mean over␣
↪redset only")

    if excess_variance:
        zsq, n = get_excess_variance(absolute=absolute, ndays=ndays,␣
↪ninds=ninds, weighted=weighted, nvar=nvars)
        dist = excess_variance(absolute=absolute, n=ndays)
    else:
        zsq, n = get_samples(
            ndays = ndays,
            ninds = ninds,
            nvars = nvars,
            absolute = absolute,
            mean_over_days=mean_over_redset,
            mean_over_ind = mean_over_indset,
            weighted = weighted,
            allow_zeros = allow_zeros
        )

        if mean_over_redset:
```

```
            if mean_over_indset:
                dist =␣
↪mean_zsquare_over_redundant_and_independent_sets(nsets=ninds, ntot=n[0],␣
↪absolute=absolute)
                label=r"$\overline{|\zeta|^2}$" if absolute else␣
↪r"$\overline{\mathcal{K}(\zeta)^2}$"
            else:
                if allow_zeros:
                    unique_n, counts = np.unique(n, return_counts=True)
                    indx = np.argwhere(unique_n >= 2)[:, 0]
                    unique_n = unique_n[indx]
                    counts = counts[indx]
                    dist = MixtureModel([mean_zsquare_over_redundant_set(n=nn,␣
↪absolute=absolute) for nn in unique_n], weights=counts)
                    label=r"$|\zeta|^2$ (non-uniform)" if absolute else␣
↪r"$\mathcal{K}(\zeta)^2$ (non-uniform)"
                else:
                    dist = mean_zsquare_over_redundant_set(n=n[0],␣
↪absolute=absolute)
                    label=r"$|\zeta|^2$" if absolute else␣
↪r"$\mathcal{K}(\zeta)^2$"
        elif mean_over_indset:
            dist = mean_zsquare_over_independent_set(q=ninds, absolute=absolute)
            label=r"$\bar{|Z|^2}$" if absolute else r"$\bar{\mathcal{K}(Z)^2}$"
        else:
            dist = zsquare(absolute=absolute)
            label=r"$|Z|^2}$" if absolute else r"$\mathcal{K}(Z)^2$"

    xx = np.logspace(np.log10(zsq.min()), np.log10(zsq.max()), 100)

    plt.hist(zsq, density=True, bins=50, histtype='step', label=lbl,␣
↪color=color)

    plt.plot(xx, dist.pdf(xx), color=color, ls='--')
    plt.yscale('log')

    plt.title(f"PDF of {label}")
```

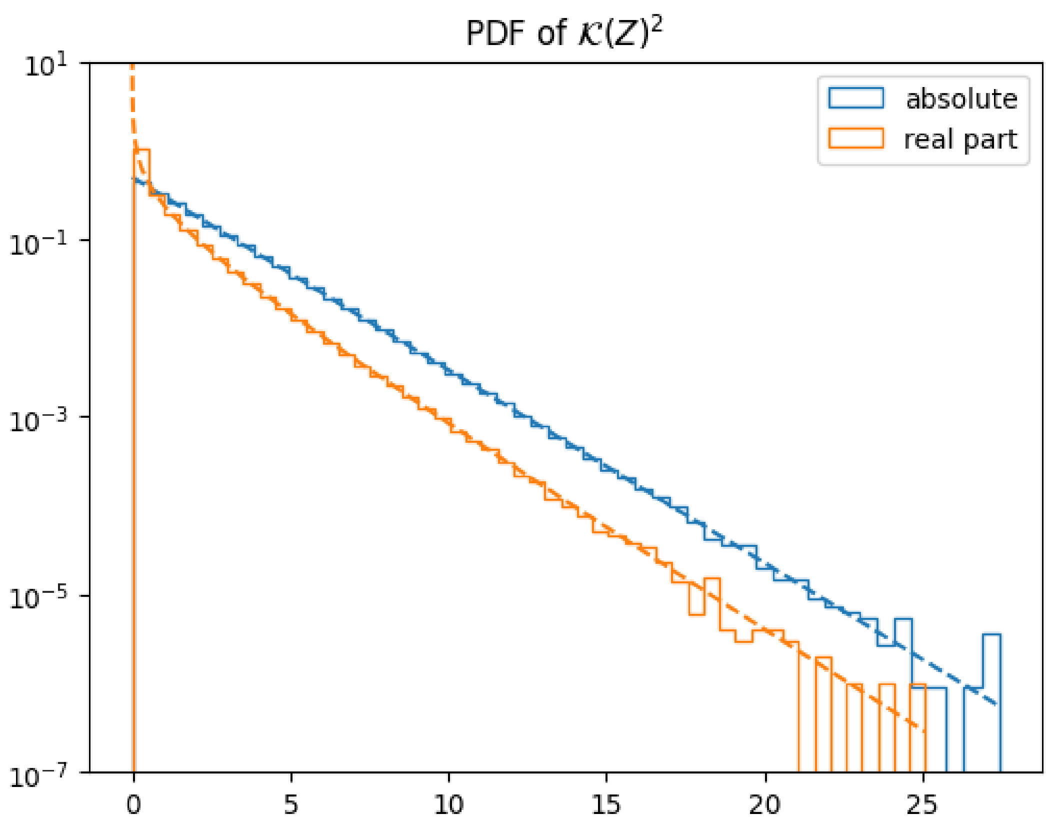
[6]:
```
fig, ax = plt.subplots(1, 1)

make_comparison_plot(
    ndays=10,
    absolute=True,
    weighted=True,
    lbl='absolute',
    color='C0'
```

```
)

make_comparison_plot(
    ndays=10,
    absolute=False,
    weighted=True,
    lbl='real part',
    color='C1'
)

plt.legend()
plt.ylim(1e-7, 1e1);
```

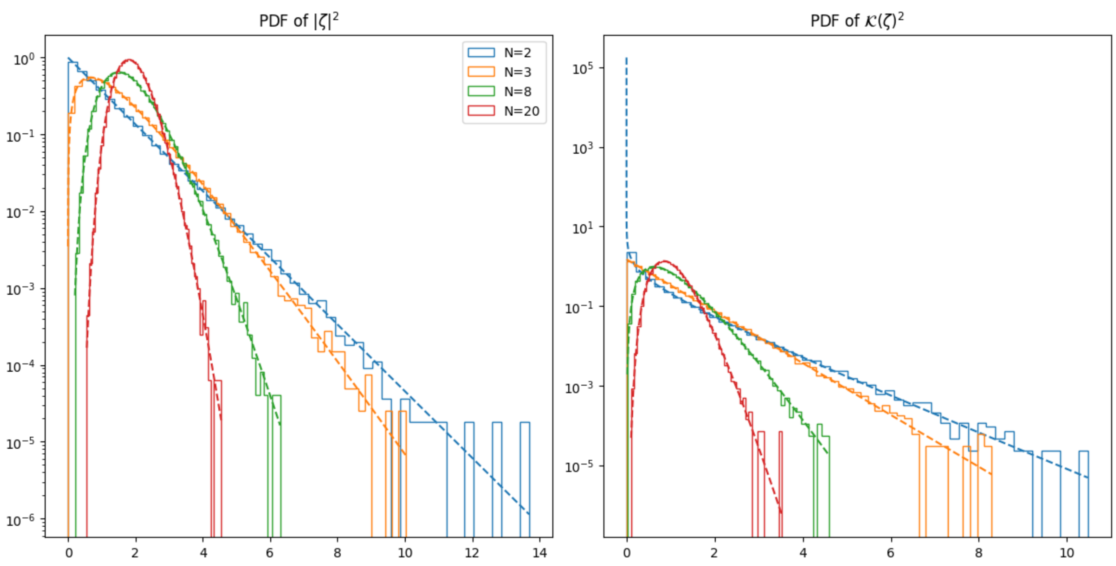


[74]:
```
fig, ax = plt.subplots(1, 2, constrained_layout=True, figsize=(12, 6))

plt.sca(ax[0])
for i, N in enumerate((2, 3, 8, 20)):
    make_comparison_plot(
        ndays=N,
        absolute=True,
```

```
        weighted=True,
        mean_over_redset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
ax[0].legend()

plt.sca(ax[1])
for i, N in enumerate((2, 3, 8, 20)):
    make_comparison_plot(
        ndays=N,
        absolute=False,
        weighted=True,
        mean_over_redset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
```

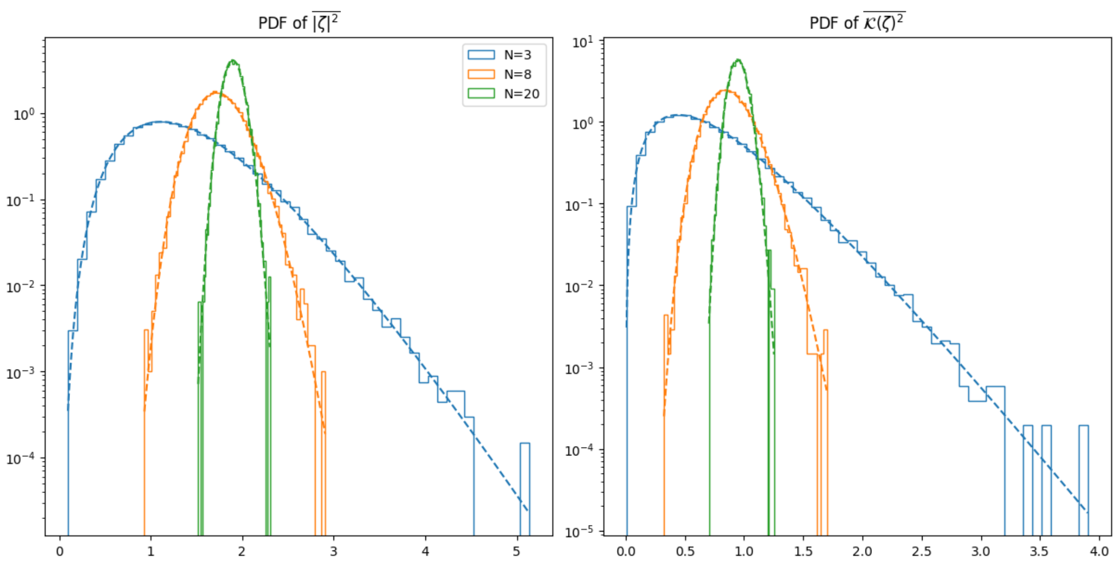


[77]:
```
fig, ax = plt.subplots(1, 2, constrained_layout=True, figsize=(12, 6))

plt.sca(ax[0])
for i, N in enumerate((3, 8, 20)):
    make_comparison_plot(
        ndays=N,
        ninds=N,
        nvars=200000//N,
        absolute=True,
        weighted=True,
```

```python
        mean_over_redset=True,
        mean_over_indset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
ax[0].legend()

plt.sca(ax[1])
for i, N in enumerate((3, 8, 20)):
    make_comparison_plot(
        ndays=N,
        nvars=200000//N,
        ninds=N,
        absolute=False,
        weighted=True,
        mean_over_redset=True,
        mean_over_indset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
```

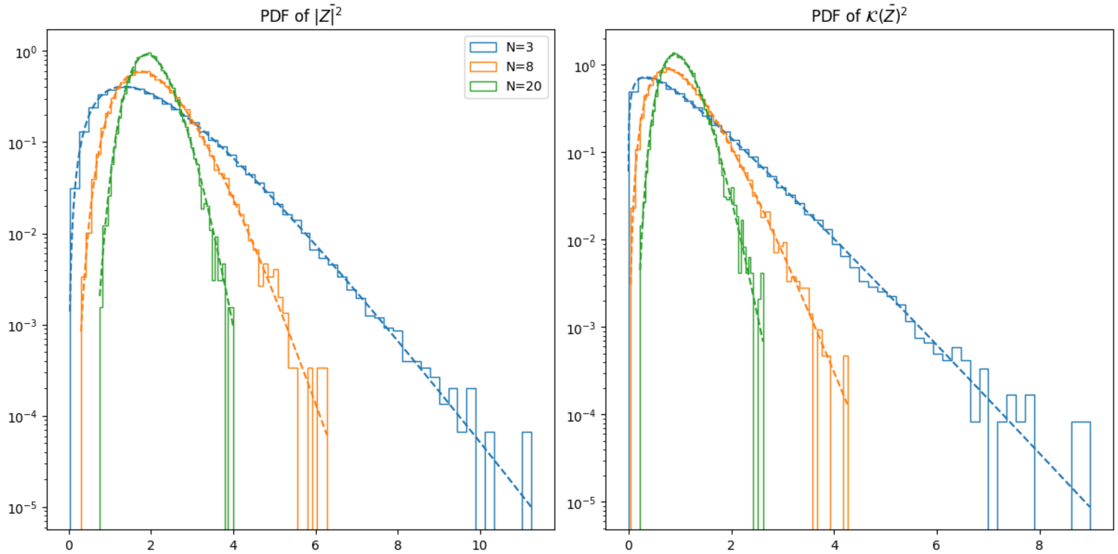


[78]:
```python
fig, ax = plt.subplots(1, 2, constrained_layout=True, figsize=(12, 6))

plt.sca(ax[0])
for i, N in enumerate((3, 8, 20)):
    make_comparison_plot(
        ndays=N,
        ninds=N,
```

```
        nvars=200000//N,
        absolute=True,
        weighted=True,
        mean_over_redset=False,
        mean_over_indset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
ax[0].legend()

plt.sca(ax[1])
for i, N in enumerate((3, 8, 20)):
    make_comparison_plot(
        ndays=N,
        nvars=200000//N,
        ninds=N,
        absolute=False,
        weighted=True,
        mean_over_redset=False,
        mean_over_indset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
```



[118]:
```
fig, ax = plt.subplots(1, 2, constrained_layout=True, figsize=(12, 6))

plt.sca(ax[0])
for i, N in enumerate((3, 4, 8)):
```

```
    make_comparison_plot(
        ndays=N,
        absolute=True,
        weighted=True,
        mean_over_redset=True,
        allow_zeros=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
ax[0].legend()

plt.sca(ax[1])
for i, N in enumerate((3, 4, 8)):
    make_comparison_plot(
        ndays=N,
        absolute=False,
        weighted=True,
        allow_zeros=True,
        mean_over_redset=True,
        lbl=f'N={N}',
        color=f'C{i}'
    )
```