Statistical and software tools for quantifying uncertainty in event attribution

Christopher Paciorek   UC Berkeley Statistics

Dáithí Stone   Lawrence Berkeley National Lab
Michael Wehner   Lawrence Berkeley National Lab
Hari Krishnan   Lawrence Berkeley National Lab

IDAG Workshop
March 2018

Work funded under the CASCADE project by DOE DE-AC0205CH11231

Outline

- Statistical methods for event attribution
  - Sources of uncertainty
  - Risk ratio estimation methods
  - Methods for quantifying uncertainty
  - Recommendations
- Illustration with Texas heatwave of 2011
  - Sensitivity analysis with respect to event definition
  - Use of climextremes package for risk ratio estimation
- climextremes package
  - Illustration on station precipitation data
  - Features/capabilities

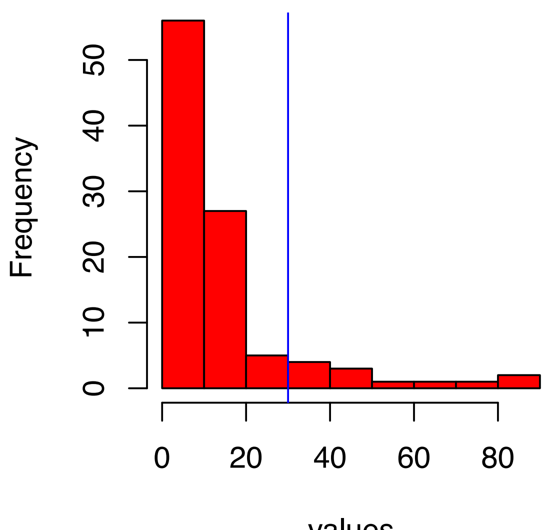
Sources of uncertainty in model-based event attribution

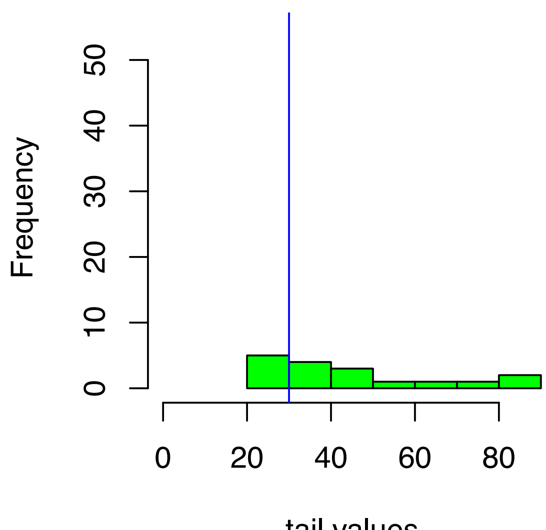
- Sampling uncertainty
  - Due to variability in the earth system
  - Amenable to frequentist or Bayesian statistical treatment
  - Uncertainty decreases with larger ensembles

- Non-sampling uncertainty
  - Sources:
    - Boundary condition uncertainty
    - Model parametric uncertainty
    - Model structural uncertainty
  - Not amenable to frequentist treatment
  - Does not decrease with larger ensembles
  - Possibly characterized based on sensitivity analysis or drawing from prior distribution over boundary conditions, parameter values, models, etc.

Methods to estimate probabilities for risk ratio (RR)

- Fit a parametric statistical model to climate variable values
  - E.g., lognormal or gamma distribution
  - Estimate probability of exceeding the threshold from fitted distribution
  - Strong assumption about appropriateness of model (all data used)
- Count exceedances of threshold (binomial sample) amongst climate variable values
  - "Nonparametric" – no distributional assumption
  - More involved to account for dependence (e.g., daily data)
  - High uncertainty when there are very few events, but can be effective for RR
- Fit extreme value distribution (e.g., GEV or peaks-over-threshold (POT))
  - Theoretically justified when threshold is far in tail
  - Informed only by extreme values
  - Hard to use with seasonal events because of small sample sizes





Methods to estimate uncertainty in RR

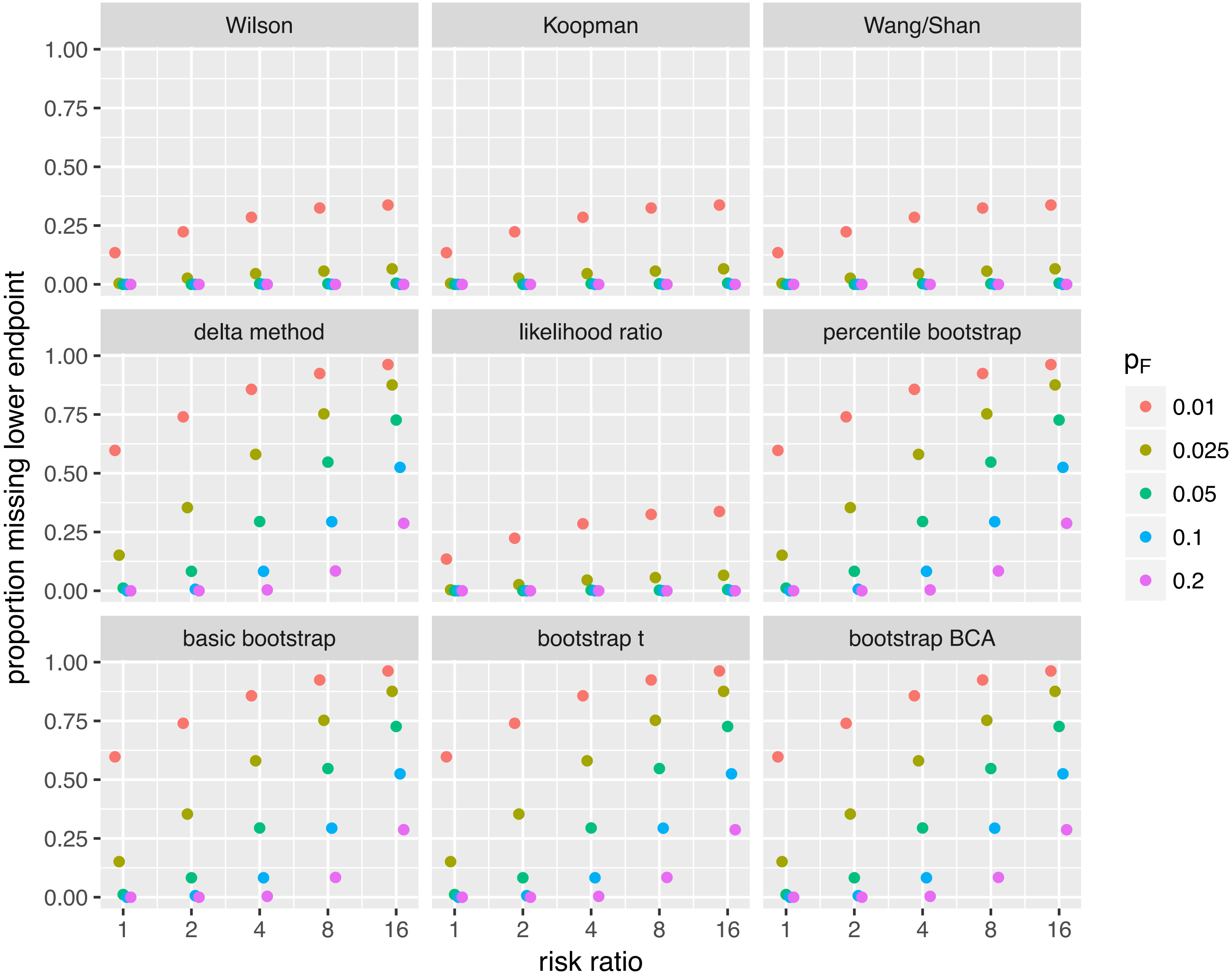
- Methods for **extreme value methods and binomial counting**
  - Asymptotic statistical calculations (delta method / propagation of error)
    - Assumption of normality
  - Bootstrap
    - Standard statistical bootstrap gives a confidence interval, not a Bayesian probability interval
    - Estimation of RR in bootstrap samples often fails (lack of EVA convergence, zeros in binomial counting approach)
  - Likelihood ratio-based interval

- Methods for **binomial counting** (from epidemiology/biostatistics)
  - Wilson's method
  - Koopman's method
  - Wang/Shan method

- All but asymptotic and bootstrap can give interval when RR estimate is 0 or infinity, e.g., (12.8, Infinity)

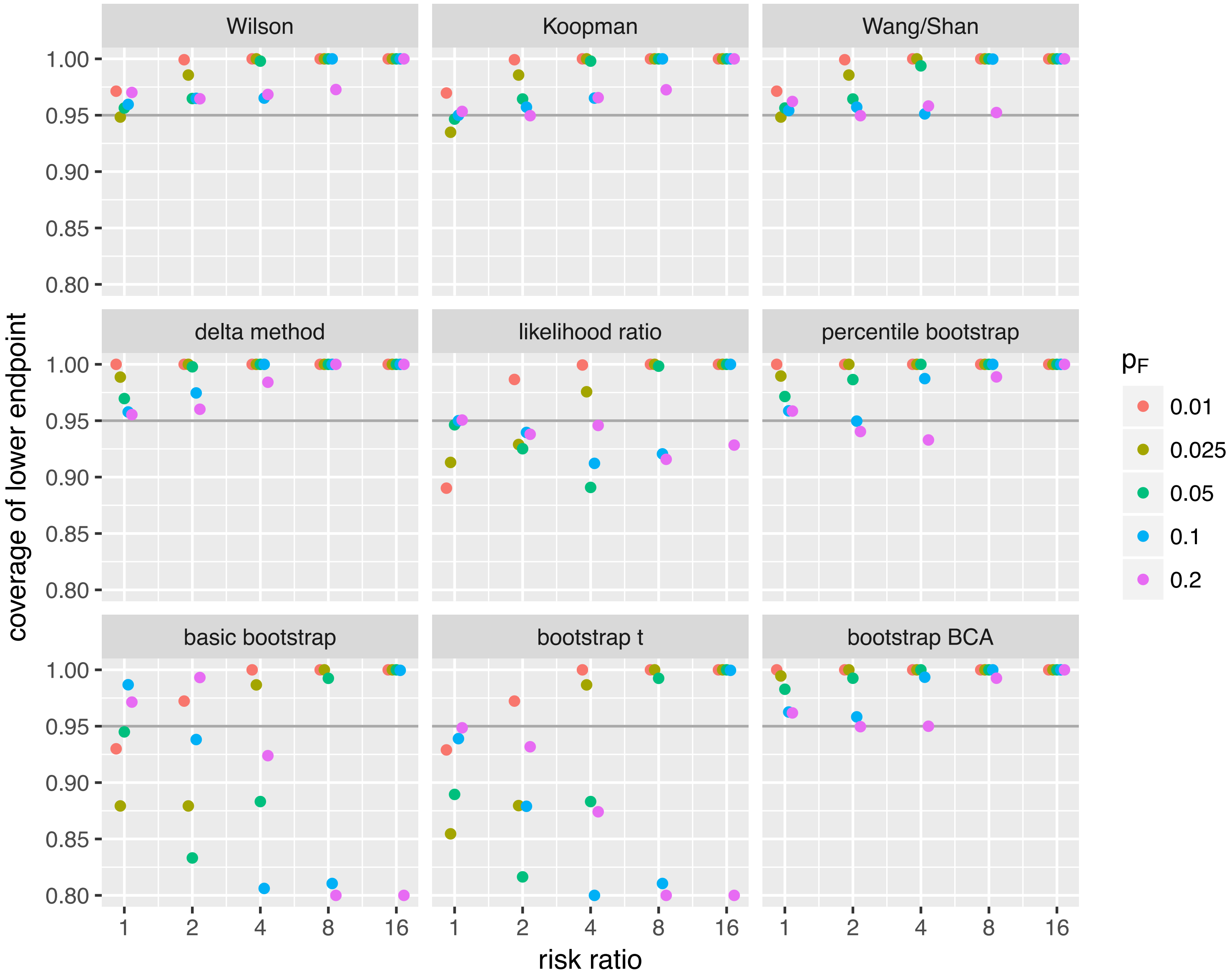
RR confidence intervals (binomial counting) simulation results

Proportion of times interval cannot be calculated (e.g., 0 events / 0 events)



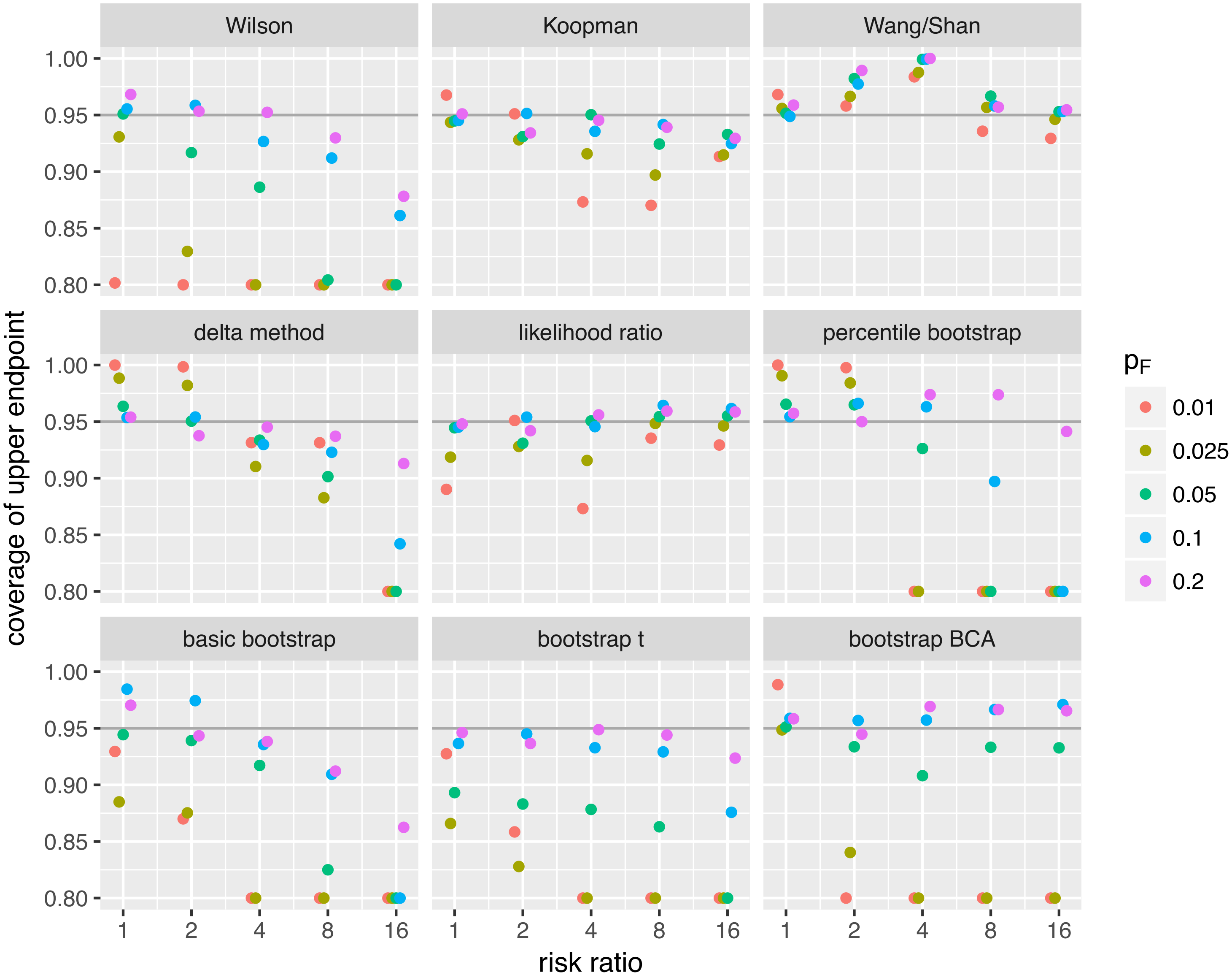
RR confidence intervals (binomial counting) simulation results

Proportion of times lower interval endpoint includes true RR (95% is best)



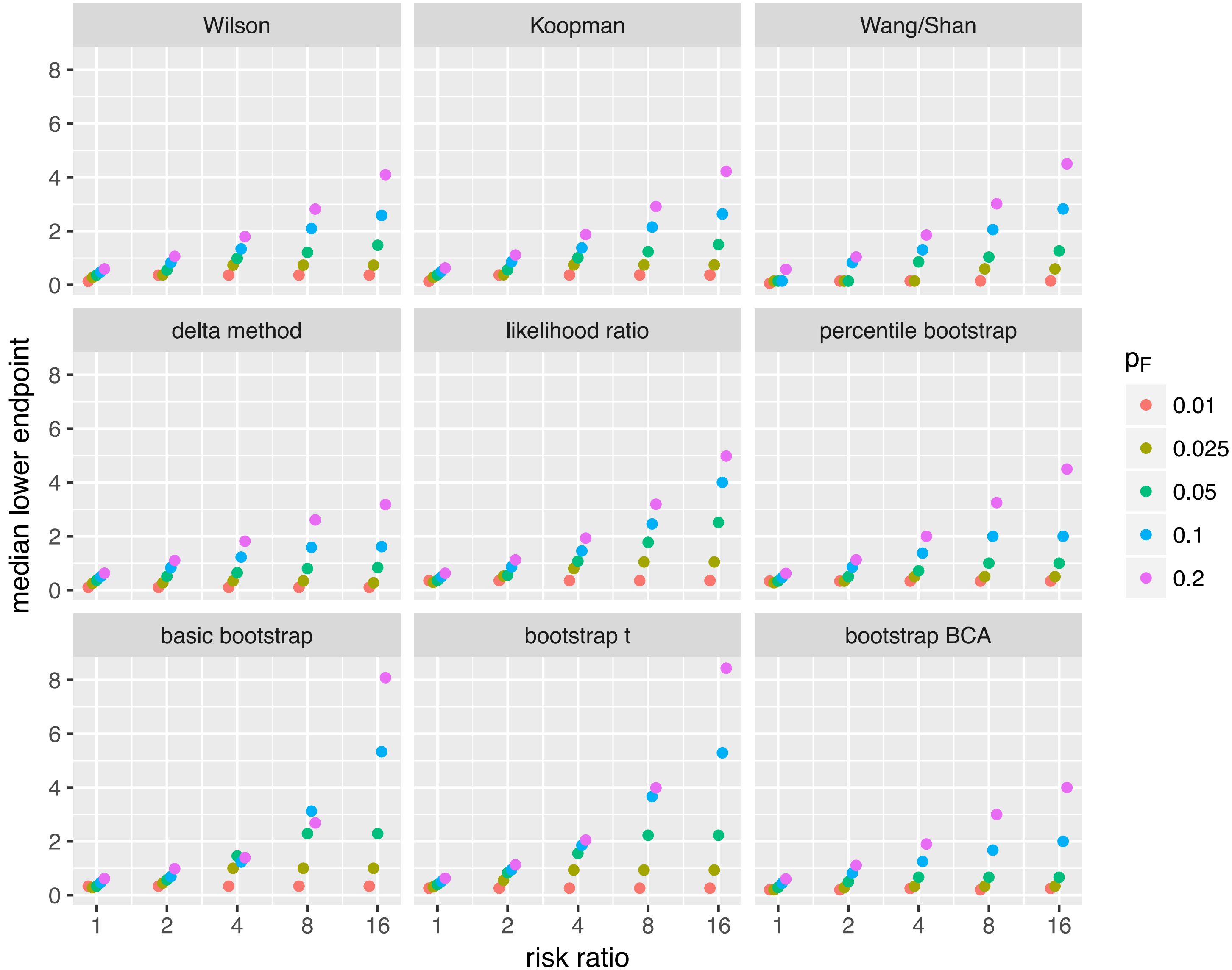
RR confidence intervals (binomial counting) simulation results

Proportion of times upper interval endpoint includes true RR (95% is best)



RR confidence intervals (binomial counting) simulation results

Average value of lower endpoint (higher is better)

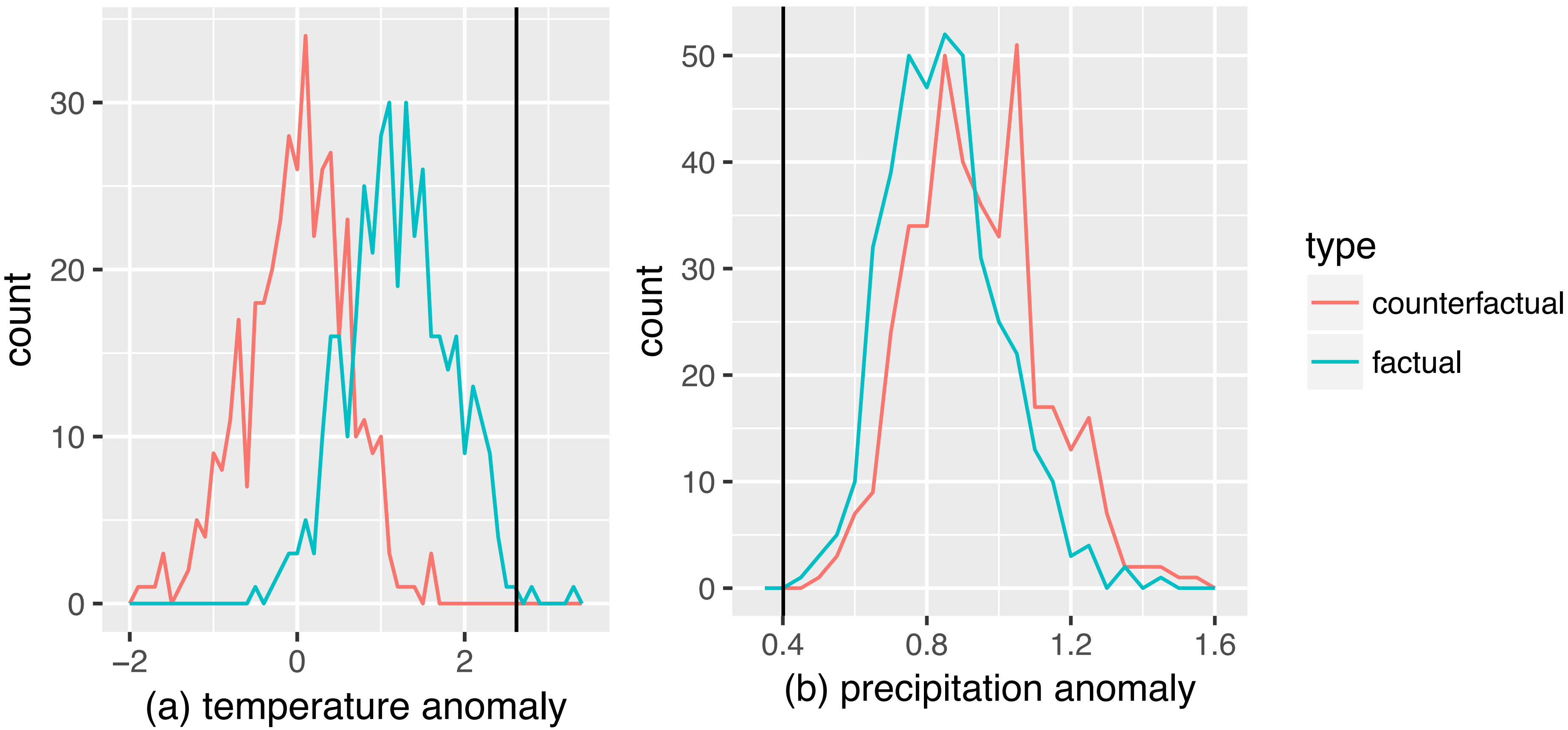


Methods to estimate uncertainty in RR: conclusions

- Bootstrap methods
  - Often fail to provide an interval
  - Poor statistical performance
- Likelihood ratio method
  - Reasonably good statistical performance, but coverage sometimes too low
  - Will work with both extreme value analysis and binomial counting
- Epidemiology/biostatistical methods
  - Work only for binomial counting
  - Koopman and Wang-Shan methods generally perform well
- All methods except Wang-Shan available in climextremes software package

RR analysis example: Texas 2011 heatwave / drought

- CAM5.1 400-member ensembles for factual and counterfactual
- March-August temperature and rainfall over Texas
- Estimation done using climextremes software
- Temperature: 2/0 is RR count-based estimate
- Precipitation: 0/0 is RR count-based estimate



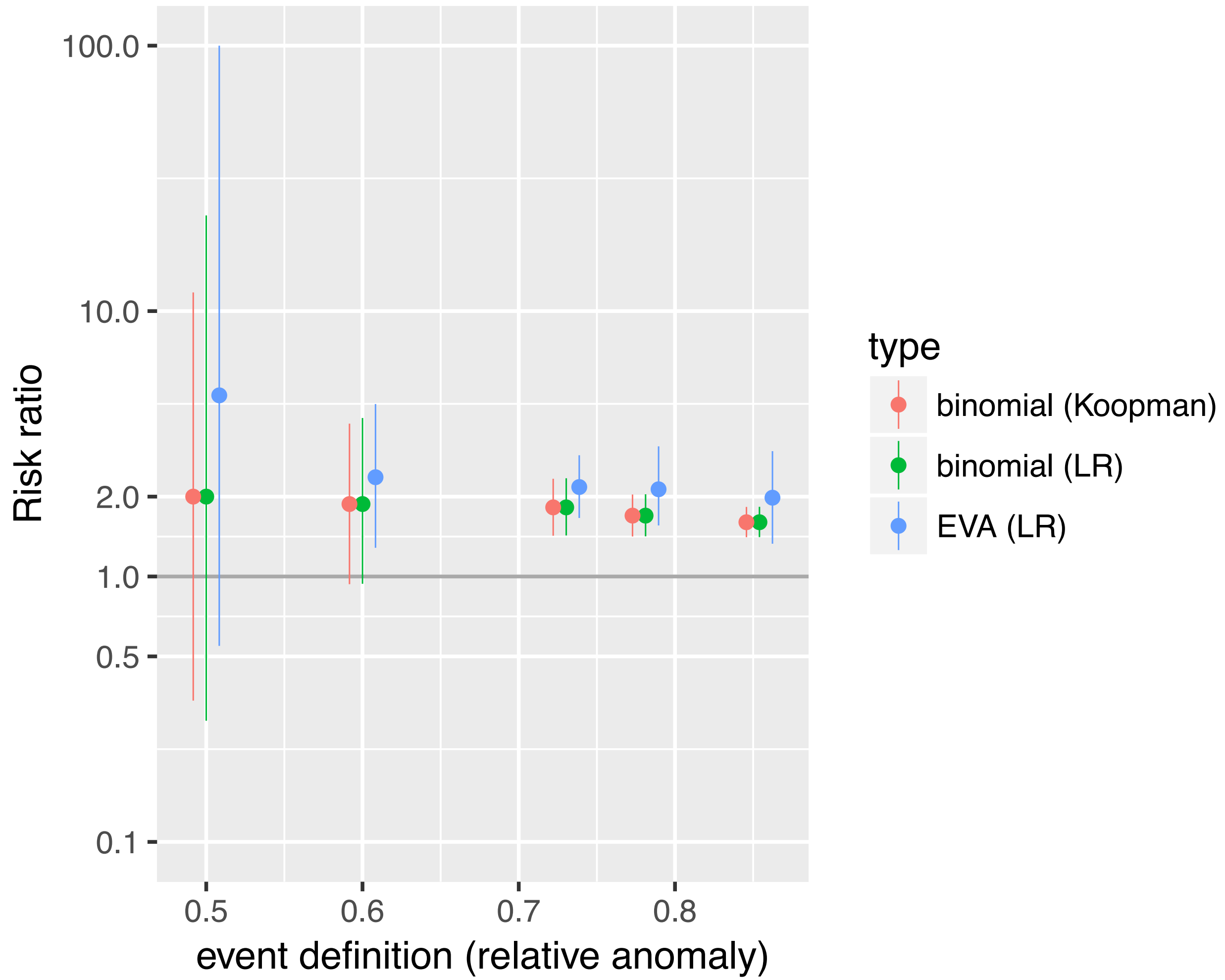
Texas 2011 temperature analysis

Actual event is 2.62 degree anomaly

Event	Number exceedances	RR esti-mate	EVA lik ratio CI	Binomial lik ratio CI	Binomial Koopman CI
2.62 (actual)	2/0	Inf	(12.8, Inf)	(1.0, Inf)	(0.7, Inf)
2.0	43/0	Inf	NA	(31, Inf)	(16, Inf)
1.5	129/3	43	NA	(19, 133)	(17, 108)
1.03 (20-year event)	245/11	22	NA	(14, 38)	(14, 36)
0.73 (10-year event)	314/40	7.9	NA	(6.2, 10.2)	(6.1, 10.1)
0.43 (5-year event)	357/90	4.0	NA	(3.4, 4.7)	(3.4, 4.6)

Texas 2011 precipitation analysis

- Recall that actual event (40% of historical average precipitation) has no events in factual or counterfactual ensembles
- Extreme value analysis (EVA) gives (0.01, Inf) as interval
- EVA not really appropriate for less extreme events but shown anyway



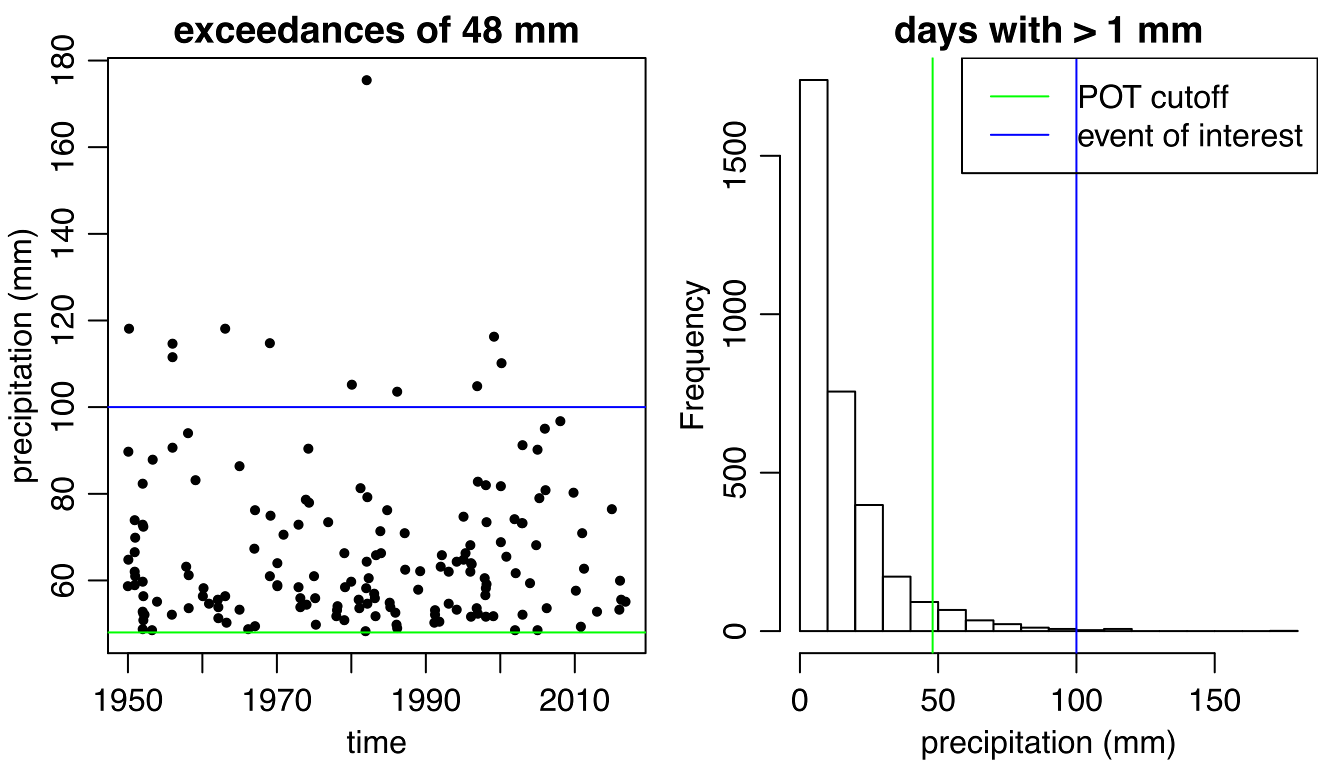
climextremes software

- High-level goals
  - Operate from Python or R
  - Provide risk ratio calculations and extreme value analysis fitting (GEV and POT)
  - Handle common situations with climate data
  - Designed for both observations and model output

- Technical features
  - Use of covariates for any extreme value distribution parameter (nonstationary fitting)
  - Estimation with uncertainty for risk ratios, return values, return periods, differences in return values
  - Various techniques for estimating uncertainty
  - Statistically rigorous estimation with model ensembles
  - Statistically rigorous treatment of missing values (for POT)
  - Allows weighting (e.g., weighting nearby stations)

climextremes example

- US GHCN Santa Cruz precipitation
  - 1950-2016
  - November-May rainy season
  - 270 missing days



climextremes example

Stationary peaks-over-threshold fit

result = climextremes.fit_pot(numpy.array(exc.y), nBlocks = nyears, threshold = threshold, firstBlock = missing.seasonyear[0], blockIndex = numpy.array(exc.seasonyear), index = numpy.array(exc.day), proportionMissing = numpy.array(missing.propMiss), declustering = 'noruns', returnPeriod = 20, returnValue = 100, bootSE = False)

Nonstationary peaks-over-threshold fit

linear location trend in time -
contrast 2015 returnValue and return probability with that for 1950

resultNS = climextremes.fit_pot(numpy.array(exc.y), x = numpy.array(missing.seasonyear), locationFun = 1, nBlocks = nyears, threshold = threshold, firstBlock = missing.seasonyear[0], blockIndex = numpy.array(exc.seasonyear), index = numpy.array(exc.day), proportionMissing = numpy.array(missing.propMiss), declustering = 'noruns', xNew = 2015, xContrast = 1950, returnPeriod = 20, returnValue = 100, bootSE = False)

climextremes example

Stationary peaks-over-threshold fit

```
# 20-year return value and standard error
result['returnValue']
# 120.3 mm
result['se_returnValue']      # return value standard error (asymptotic)
#   7.9 mm
result['logReturnProb']       # log of probability of exceeding 'returnValue=100'
# -1.98
# confidence interval on return probability for 100 mm event
np.exp(result['logReturnProb'] + np.array((-2, 2))*result['se_logReturnProb'])
# (0.0872262, 0.2200104)
```

Nonstationary fit with location linear in year

```
# change in return value (2015 - 1950) and standard error of the change
resultNS['returnValueDiff']
# -2.68  mm
resultNS['se_returnValueDiff']
#  5.37 mm
# risk ratio 2015 / 1950 for 100 mm event with confidence interval
np.exp(resultNS['logReturnProbDiff'])
#  0.88
np.exp(resultNS['logReturnProbDiff'] + np.array((-2, 2))*resultNS['se_logReturnProbDiff'])
# (0.51, 1.49)
```

References / Links

- Statistical methods:
  - Paciorek C.J., D.A. Stone, and M.F. Wehner. 2018. Quantifying statistical uncertainty in the attribution of human influence on severe weather. Weather and Climate Extremes, accepted.
  - https://arxiv.org/abs/1706.03388

- climextremes software (version 0.2.0):
  - Available via conda for python
  - Available via CRAN for R
  - Repository: https://bitbucket.org/lbl-cascade/climextremes-dev