Freie Universität Berlin

Baysian estimation with MCMC and ABC

by Lydia Buntrock

Course: Applied Numerics

Tasks

Freie Universität Berlin

Paper: Preconditioned Metropolis sampling as a strategy to improve efficiency in Posterior exploration.

1) Implement a simple pre-MCMC method.

2) Investigate the relationship between the acceptance rate of the MCMC and the choice of different ‘preconditioners’

MCMC method

Given we start at σ_1 ,the acceptance probability moving to σ_2 is given by

$$\min\left(1.0, \frac{Q(\sigma_1|\sigma_2)\mathcal{L}(\sigma_2)}{Q(\sigma_2|\sigma_1)\mathcal{L}(\sigma_1)}\right)$$

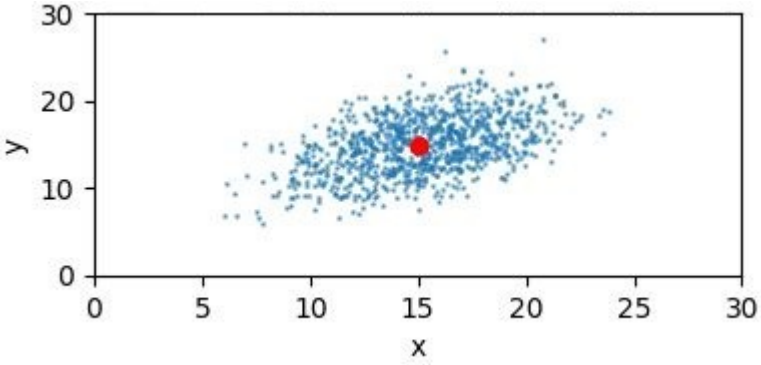
Q(·|·): proposal distribution

L(·): likelihood function

Motivation for a preconditioner: The computation of L(·) can be based on difficult equations and hard to approximate.

MCMC method

```
def walker():
    """Metropolis Hastings"""

    n = 1000
    samples = []
     # start point
    x = np.array([15.0, 15.0])
    samples.append(x)
    sigma = np.array([[16.0, 1.8], [1.8, 16.0]]) # stepsize
    A = 0 # accept
    R = 0 # reject

    while len(samples) <= n:
        # make proposal
        y = x + np.random.multivariate_normal(np.zeros(2), sigma)
        tau = np.random.uniform(0, 1)

        if tau < min(1.0, (pi(y)*q(x, y))/(pi(x)*q(y, x))):
            A +=1
            x = y.copy()
            samples.append(x)
        else:
            R += 1
    return samples
```

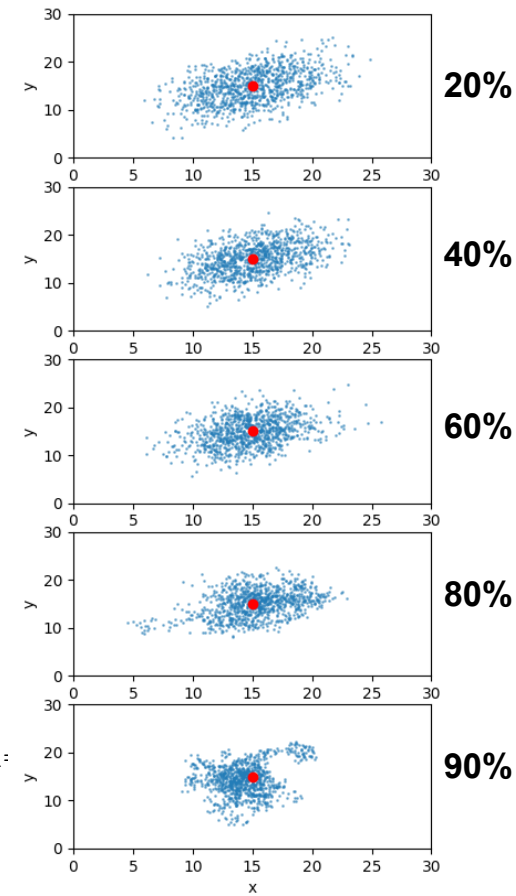



```
def q(x, y):
    # Transition Probability
    sigma = np.array([[16.0, 1.8], [1.8, 16.0]]) # stepsize
    return multivariate_normal.pdf(x, mean=y, cov=sigma)

def pi(x):
    # Target distribution
    mu = np.array([15.0, 15.0])
    std = np.array([[9, 4], [4, 9]])
    return multivariate_normal.pdf(x, mean=mu, cov=std)
```

try different proposals

```
def walker(sigma):
    """Metropolis Hastings"""

    n = 1000
    samples = []
    # start point
    x = np.array([15.0, 15.0])
    samples.append(x)

    A = 0 # accept
    R = 0 # reject

    while len(samples) <= n:
        # make proposal
        y = x + np.random.multivariate_normal(np.zeros(2), sigma)
        tau = np.random.uniform(0, 1)

        if tau < min(1.0, (pi(y)*q(x, y, sigma))/(pi(x)*q(y, x, sigma))):
            A +=1
            x = y.copy()
            samples.append(x)
        else:
            R += 1
    return samples
```

```
def main(sigma):
    sigma = np.array([[16.0, 1.8], [1.8, 16.0]])
    sigma_1 = 3.5   * sigma
    sigma_2 =         sigma
    sigma_3 = 0.4   * sigma
    sigma_4 = 0.065 * sigma
    sigma_5 = 0.02  * sigma

    walker(sigma_i)
```

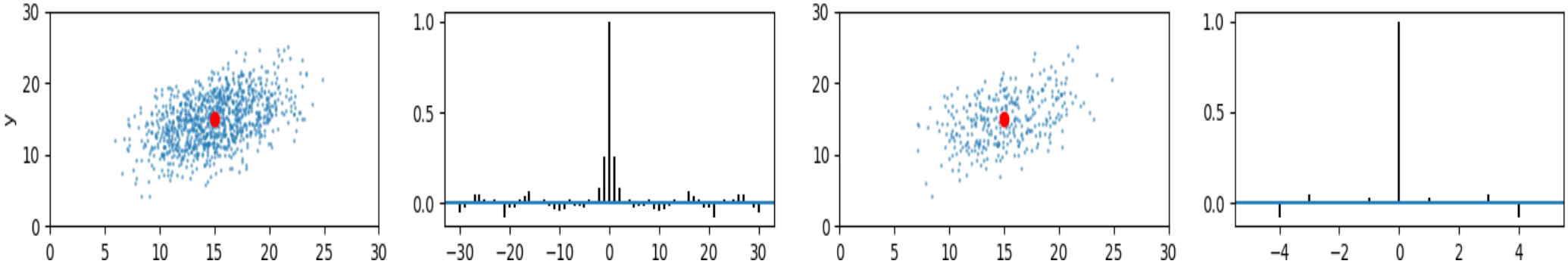



```
def q(x, y, sigma):
    # Transition Probability
    return multivariate_normal.pdf(x, mean=

def pi(x):
    # Target distribution
    mu = np.array([15.0, 15.0])
    std = np.array([[9, 4], [4, 9]])
    return multivariate_normal.pdf(x, mean=mu, cov=std)
```

Correlation

20% acceptance rate:
Tried 5000
accepted = 1000

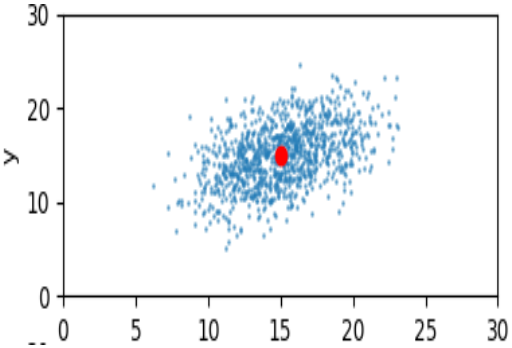
Correlation = 3
Keep 333 points

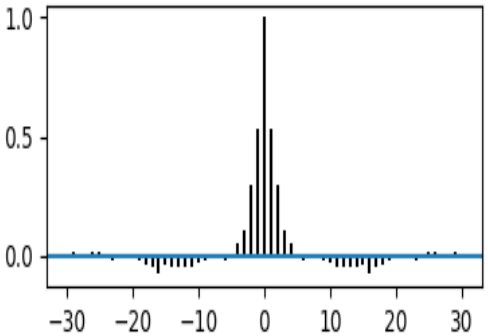


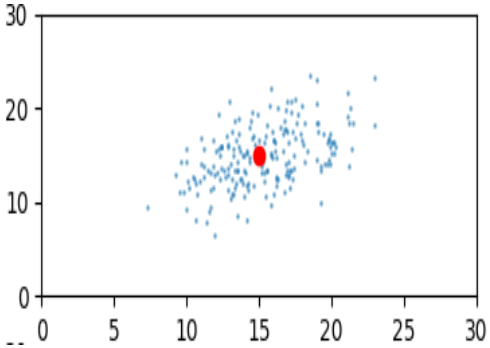
Correlation

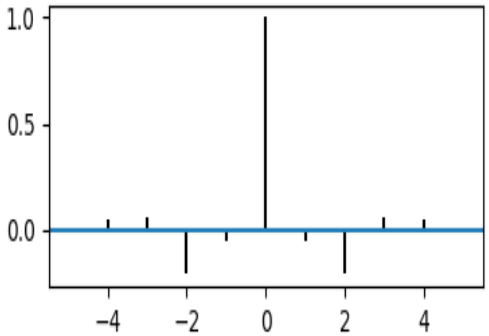
40% acceptance rate:
Tried 2500
accepted = 1000

Correlation = 5
Keep 200 points





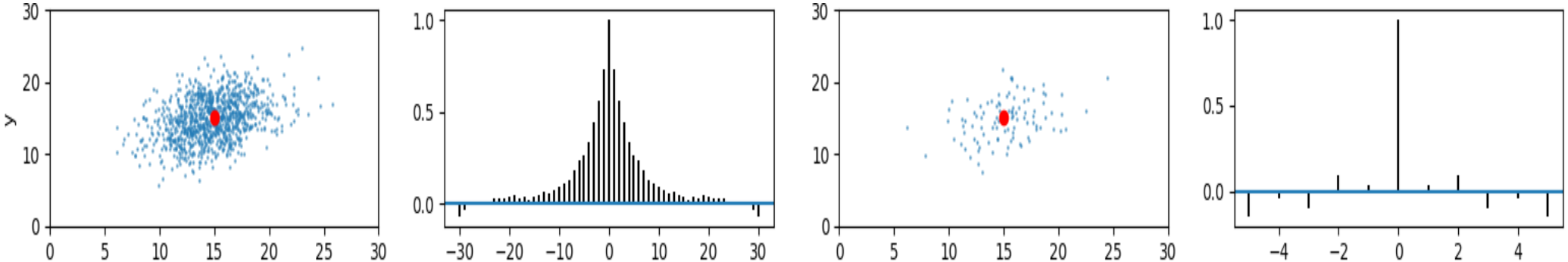




Correlation

60% acceptance rate:
Tried 1666
accepted = 1000

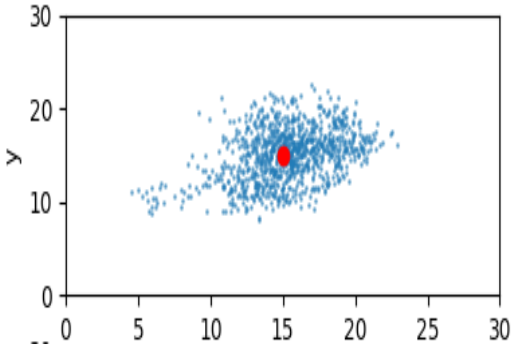
Correlation = 10
Keep 100 points

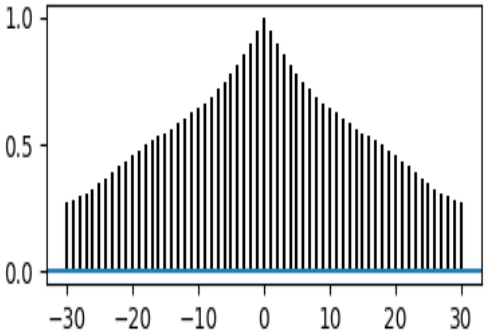


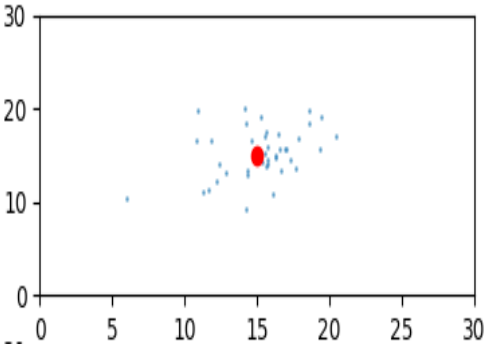
Correlation

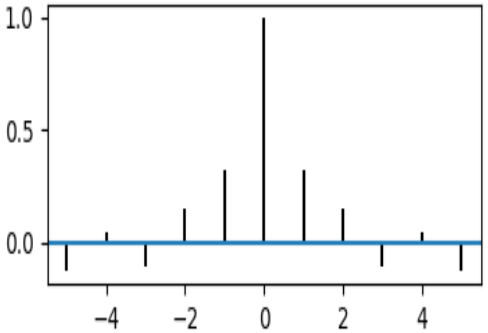
80% acceptance rate:
Tried 1250
accepted = 1000

Correlation = 25
Keep 40 points





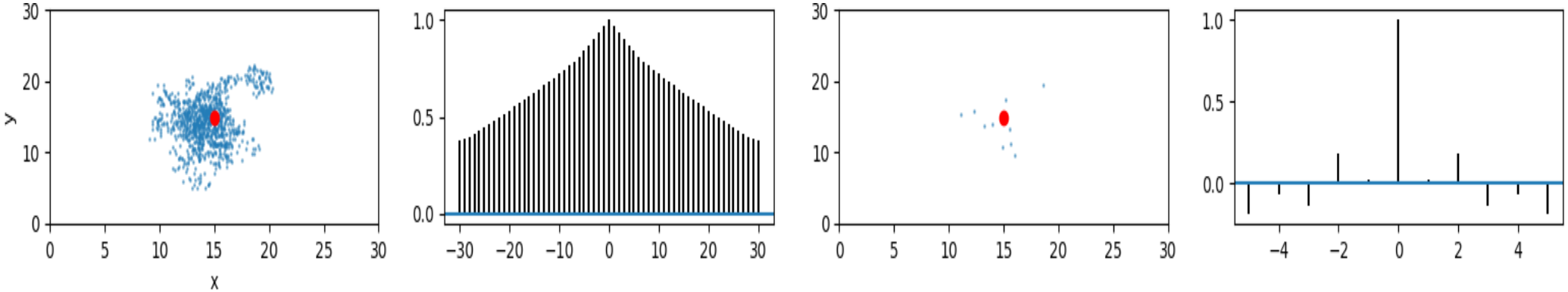




Correlation

90% acceptance rate:
Tried 1111
accepted = 1000

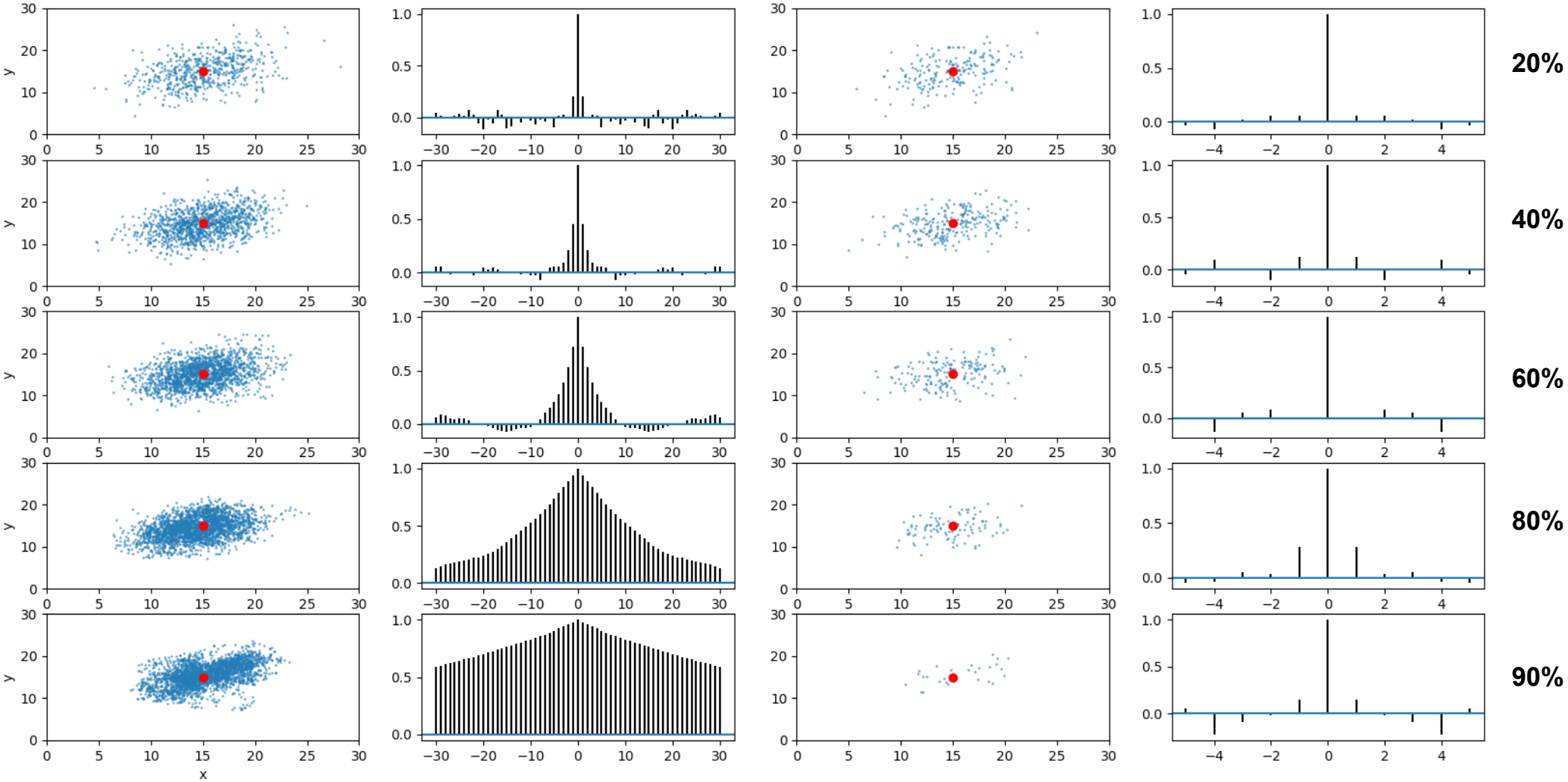
Correlation = 100
Keep 10 points



Correlation

Fix tried = 3000

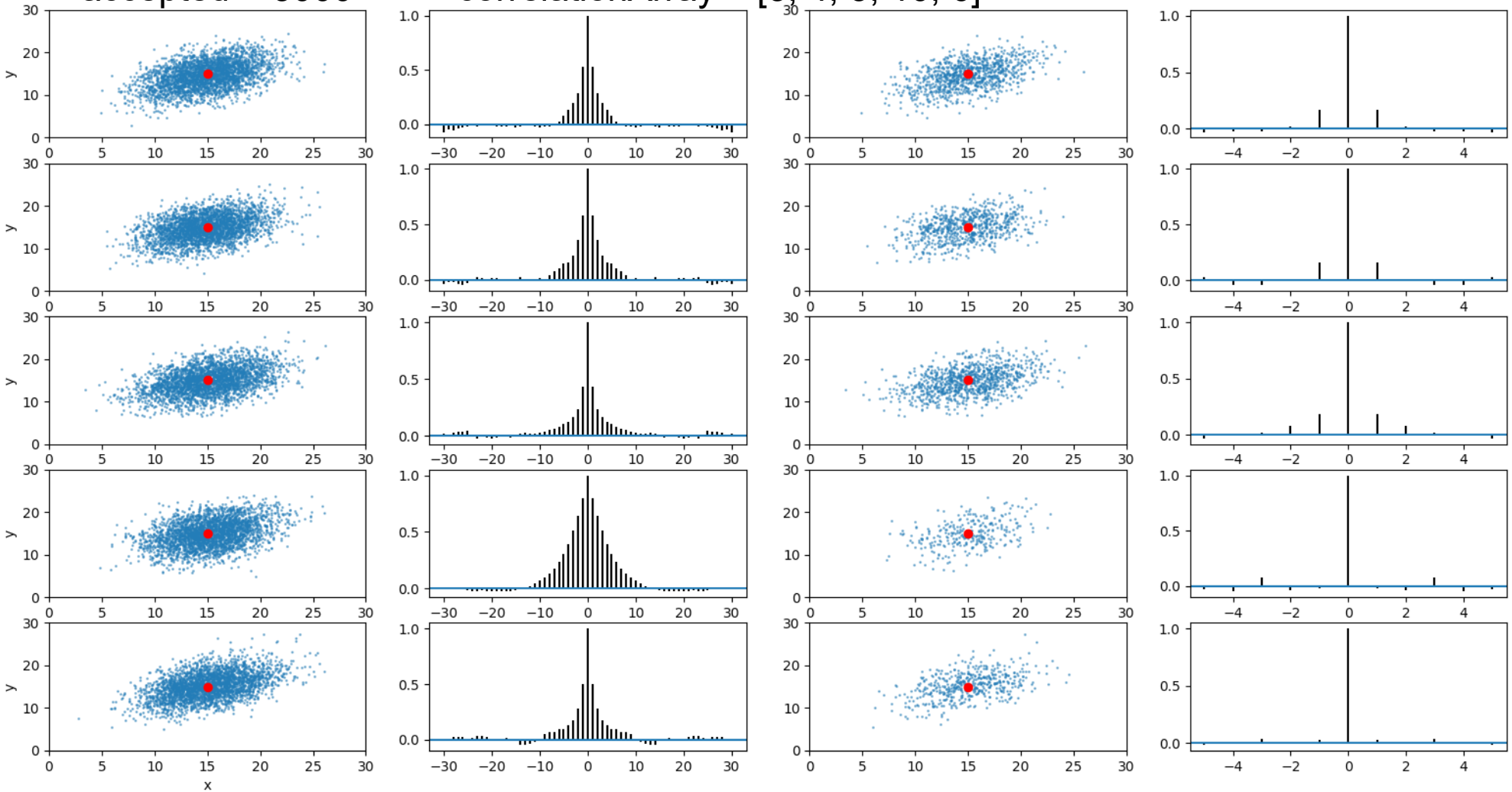
Correlation = [3, 5, 10, 25, 100]



trying different sigma's:

```
sigma_1 = [[16.0, 1.8], [1.8, 16.0]]
sigma_2 = [[16.0, -5.0], [-5.0, 16.0]]
sigma_3 = [[45.0, 0.0], [0.0, 7.5]]
sigma_4 = [[5.0, 0.0], [0.0, 30.0]]
sigma_5 = [[18, 8], [8, 18]]
```

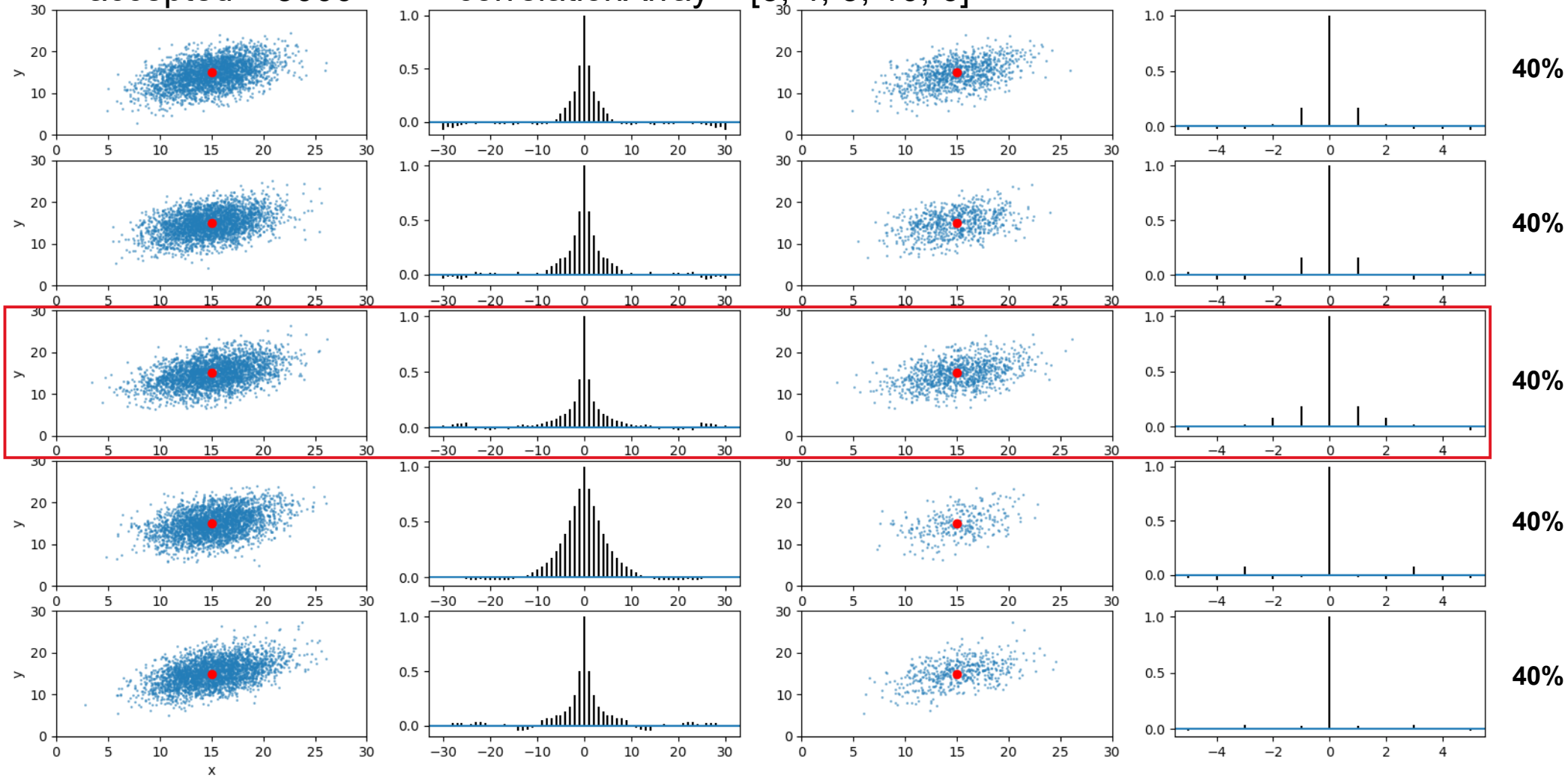
accepted = 3000 correlationArray = [3, 4, 3, 10, 6]



trying different sigma's:

```
sigma_1 = [[16.0, 1.8], [1.8, 16.0]]
sigma_2 = [[16.0, -5.0], [-5.0, 16.0]]
sigma_3 = [[45.0, 0.0], [0.0, 7.5]]
sigma_4 = [[5.0, 0.0], [0.0, 30.0]]
sigma_5 = [[18, 8], [8, 18]]
```

accepted = 3000 correlationArray = [3, 4, 3, 10, 6]



Two proposal steps

Extend our algorithm with a pre-proposal step.

(P1) *Accept* σ_2 in the first proposal step if

$$\tau_1 < \min\left(1.0, \frac{Q(\sigma_1|\sigma_2)\mathcal{L}_\star(\cdot|\sigma_2)}{Q(\sigma_2|\sigma_1)\mathcal{L}_\star(\cdot|\sigma_1)}\right).$$

(P2) *Accept* σ_2 in the second proposal step if

$$\tau_2 < \min\left(1.0, \frac{\mathcal{L}(\cdot|\sigma_2)\mathcal{L}_\star(\cdot|\sigma_1)}{\mathcal{L}(\cdot|\sigma_1)\mathcal{L}_\star(\cdot|\sigma_2)}\right).$$

$Q(\cdot|\cdot)$: proposal distribution

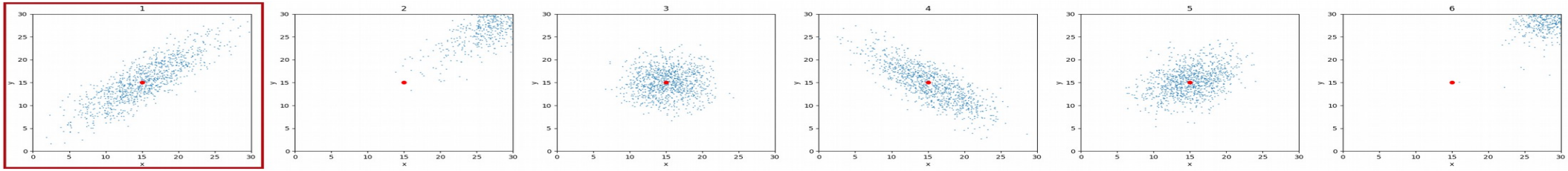
$L_*(\cdot)$: approximate likelihood function

$L(\cdot)$: likelihood function

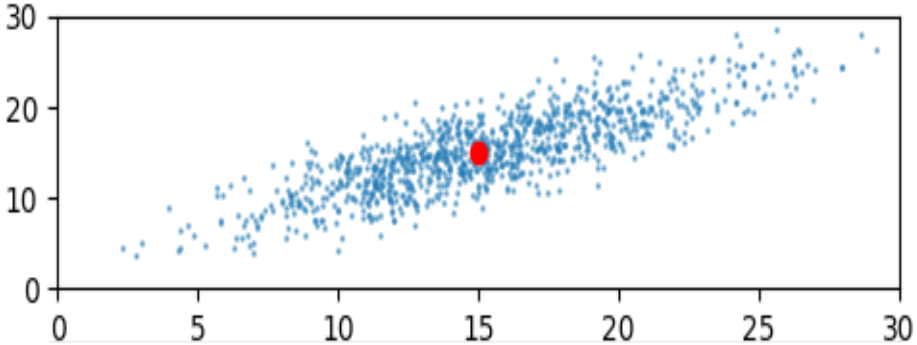
Two proposal steps

```
def walker(pre_mu, pre_std):
    """Metropolis Hastings"""
    …

    while len(samples) <= n:
        # make proposal
        y = x + np.random.multivariate_normal(...)
        pre_tau = np.random.uniform(0, 1)

        pre_pi_x = pre.pi(x, pre_mu, pre_std)
        pre_pi_y = pre.pi(y, pre_mu, pre_std)
        pre_q_x_y = pre.q(x, y)
        pre_q_y_x = pre.q(y, x)

        if pre_tau < min(1.0, (pre_pi_y*pre_q_x_y)/(pre_pi_x*pre_q_y_x)):
            A += 1 # accept
            tau = np.random.uniform(0, 1)
            preSamples.append(y)

            pi_x = mcmc.pi(x)
            pi_y = mcmc.pi(y)

            if tau < min(1.0, (pi_y*pre_pi_x)/(pi_x*pre_pi_y)):
                AA +=1 # accept, accept
                x = y.copy()
                samples.append(x)
            else:
                AR += 1 # accept, reject
        else:
            R += 1 # reject

    return [preSamples, samples]
```

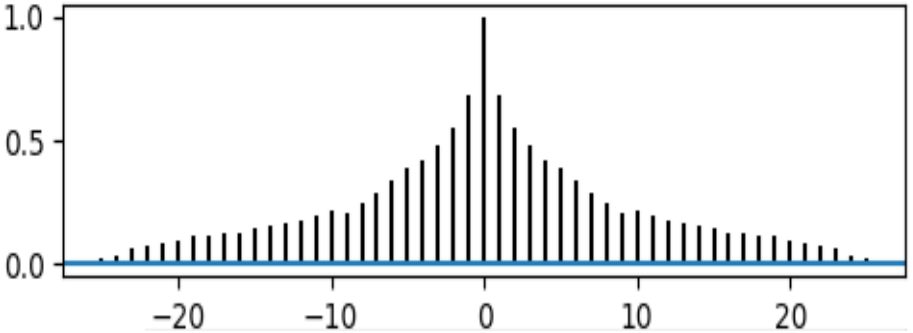
Selected target distribution:



samples

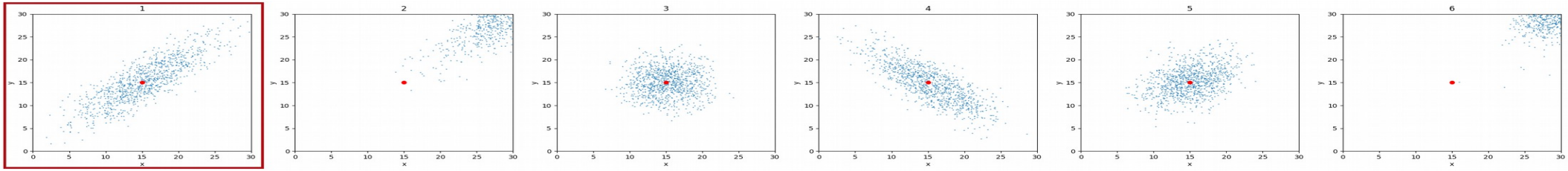


correlation

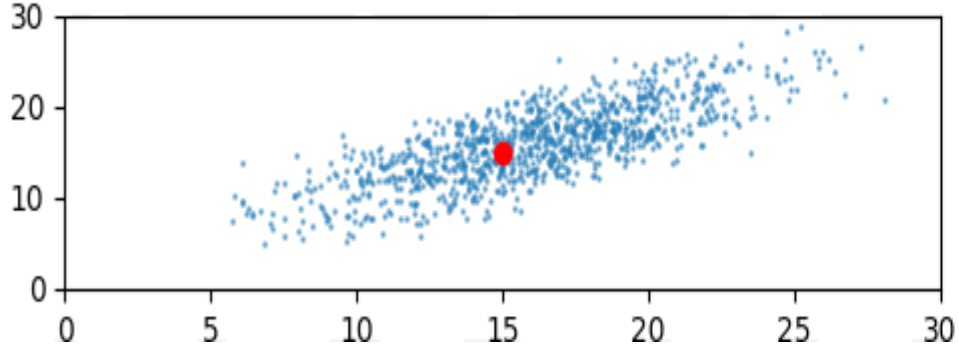


same distributions: 1 with 1
pre acceptance rate: 40%
acceptance rate: **100%**

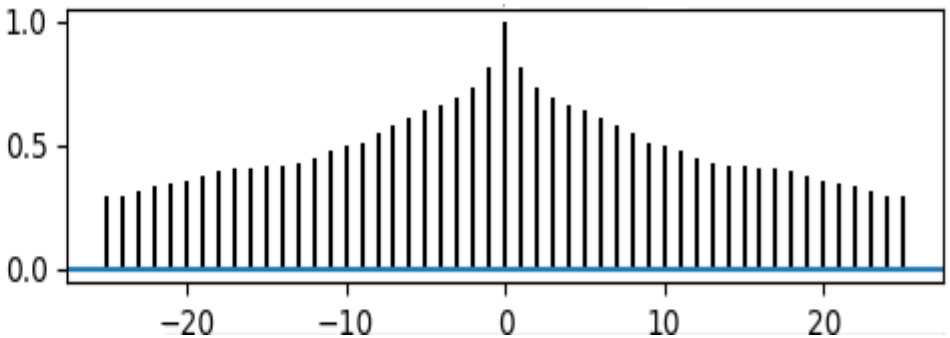
Selected target distribution:



samples

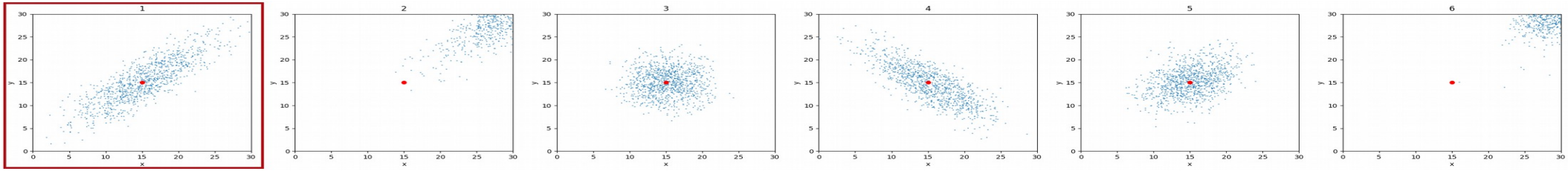


correlation

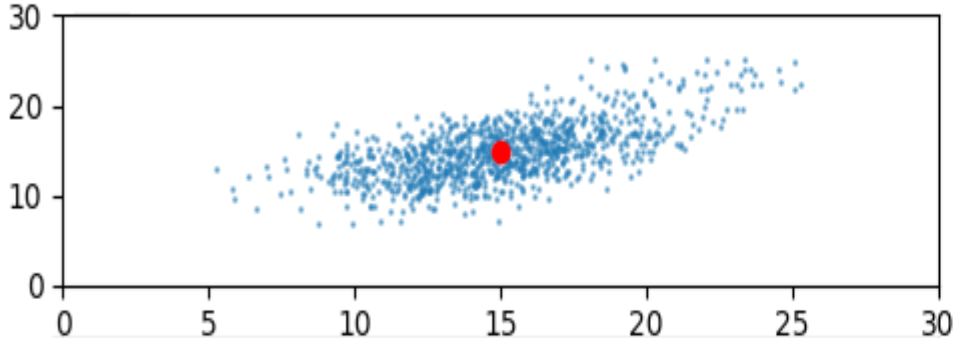


shifted distribution: 2 with 1
pre acceptance rate: 40%
acceptance rate: **40%**

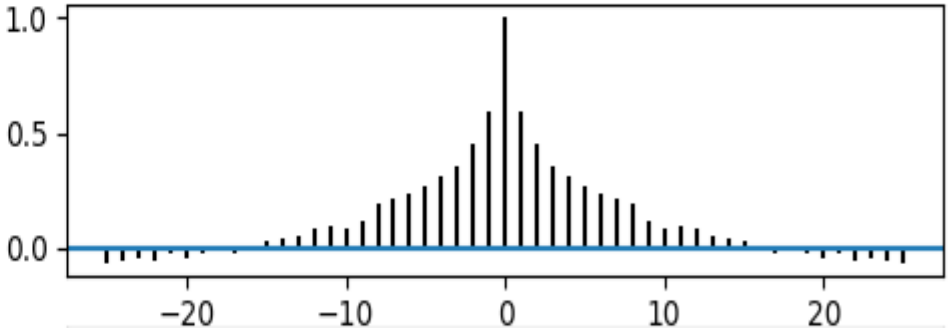
Selected target distribution:



samples

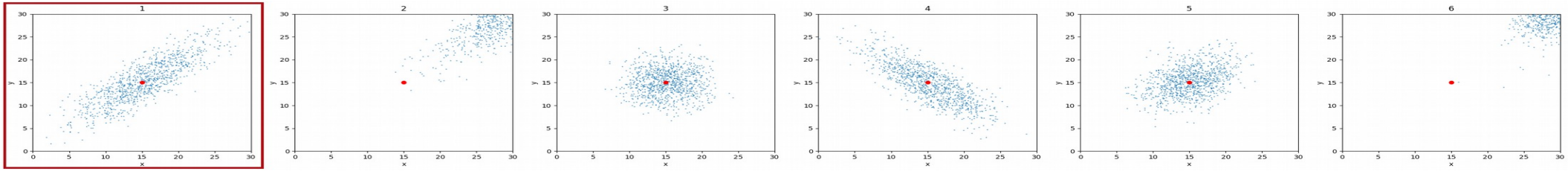


correlation

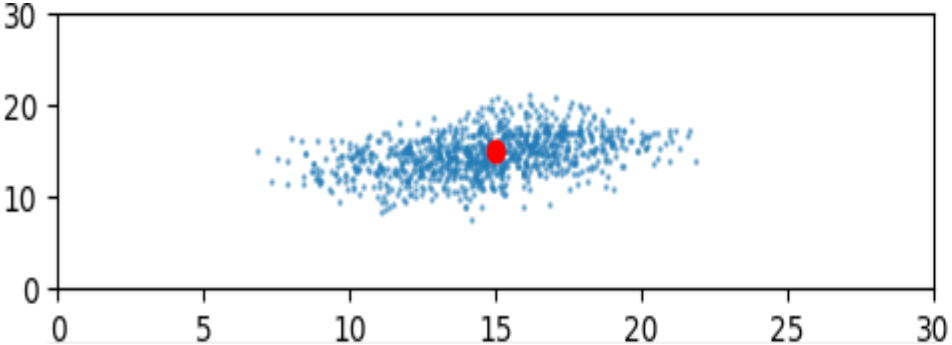


circular distribution: 3 with 1
pre acceptance rate: 40%
acceptance rate: **50%**

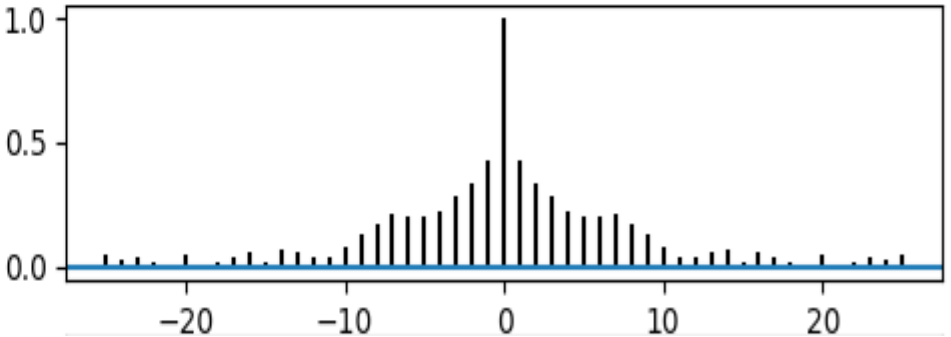
Selected target distribution:



samples

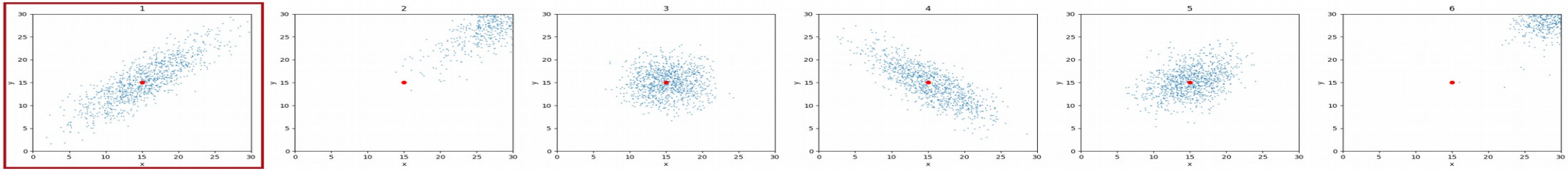


correlation

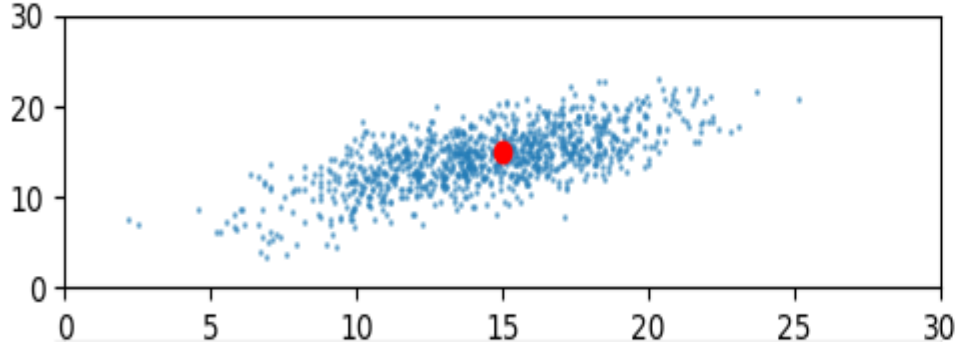


5 with 1
pre acceptance rate: 40%
acceptance rate: **30%**

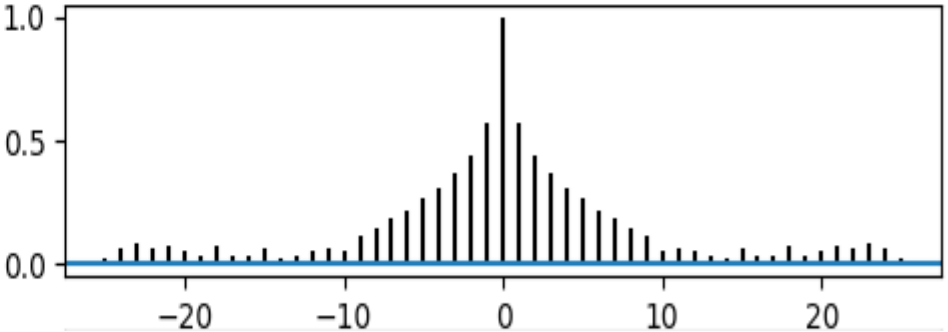
Selected target distribution:



samples

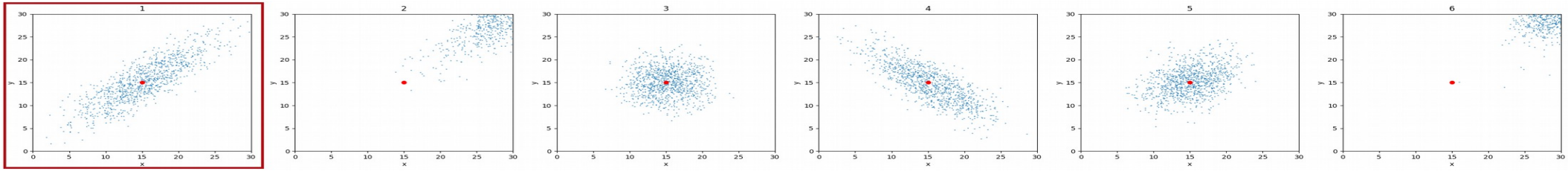


correlation

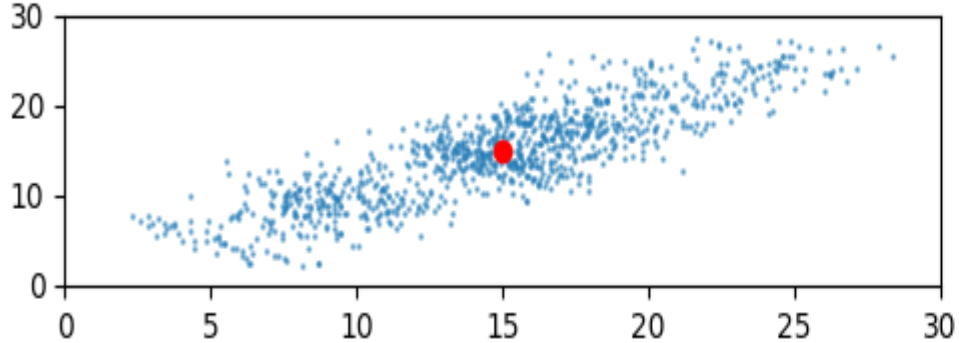


same distributions: 1 with 1
pre acceptance rate: 40%
acceptance rate: **70%**

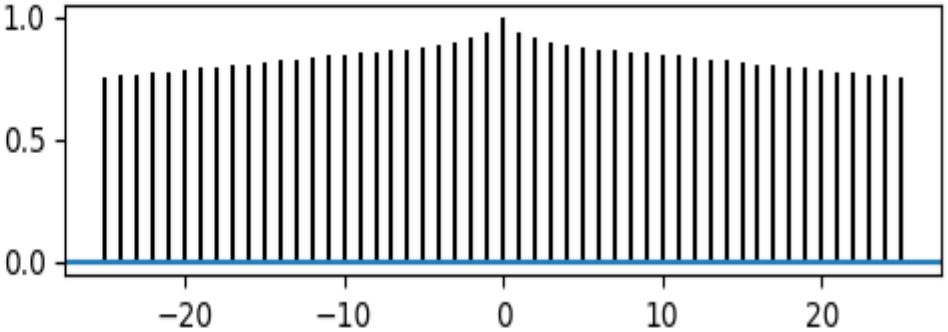
Selected target distribution:



samples

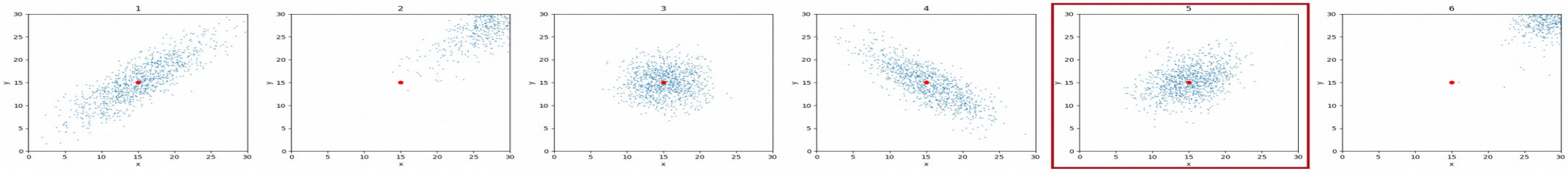


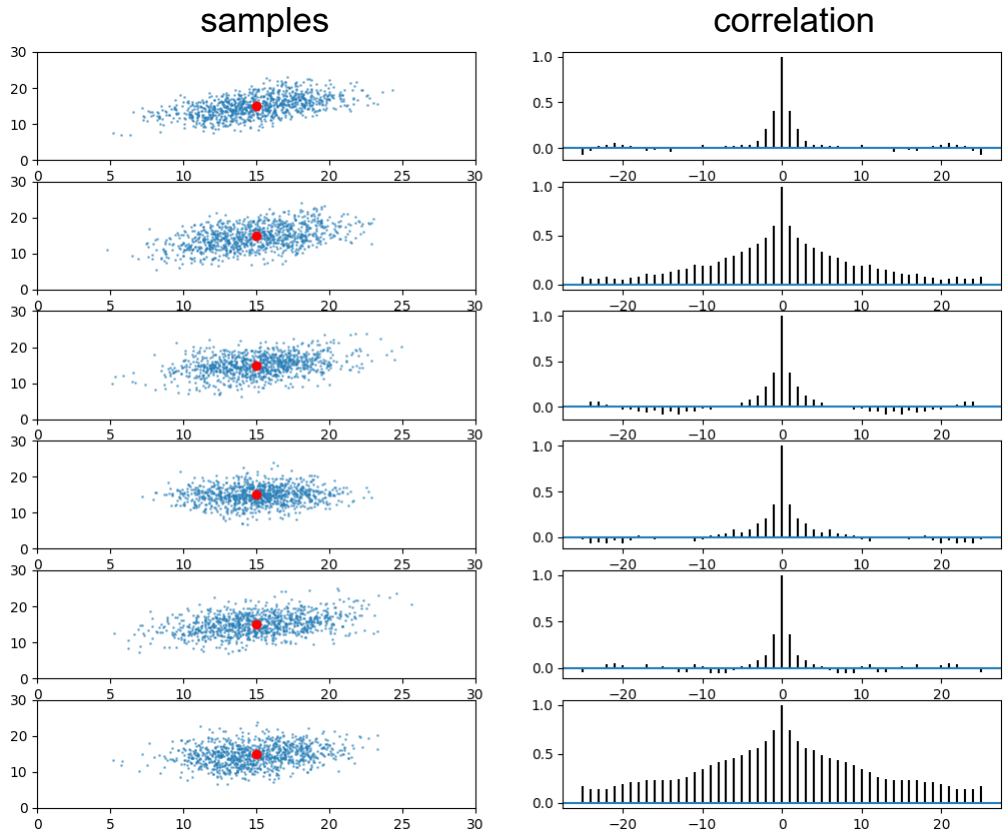
correlation



6 with 1
pre acceptance rate: 40%
acceptance rate: **13%**

Selected target distribution:





1 with 5
pre acceptance rate: 40%
acceptance rate: **65%**

2 with 5
pre acceptance rate: 40%
acceptance rate: **40%**

circular distribution: 3 with 5
pre acceptance rate: 40%
acceptance rate: **80%**

4 with 5
pre acceptance rate: 40%
acceptance rate: 45**%**

same distributions: 5 with 5
pre acceptance rate: 40%
acceptance rate: 10**0%**

shifted distribution: 6 with 5
pre acceptance rate: 40%
acceptance rate: **15%**

Results / Discussion / Conclusion

- distributions must look identical
  - with same orientation
  - and unshiftet

References

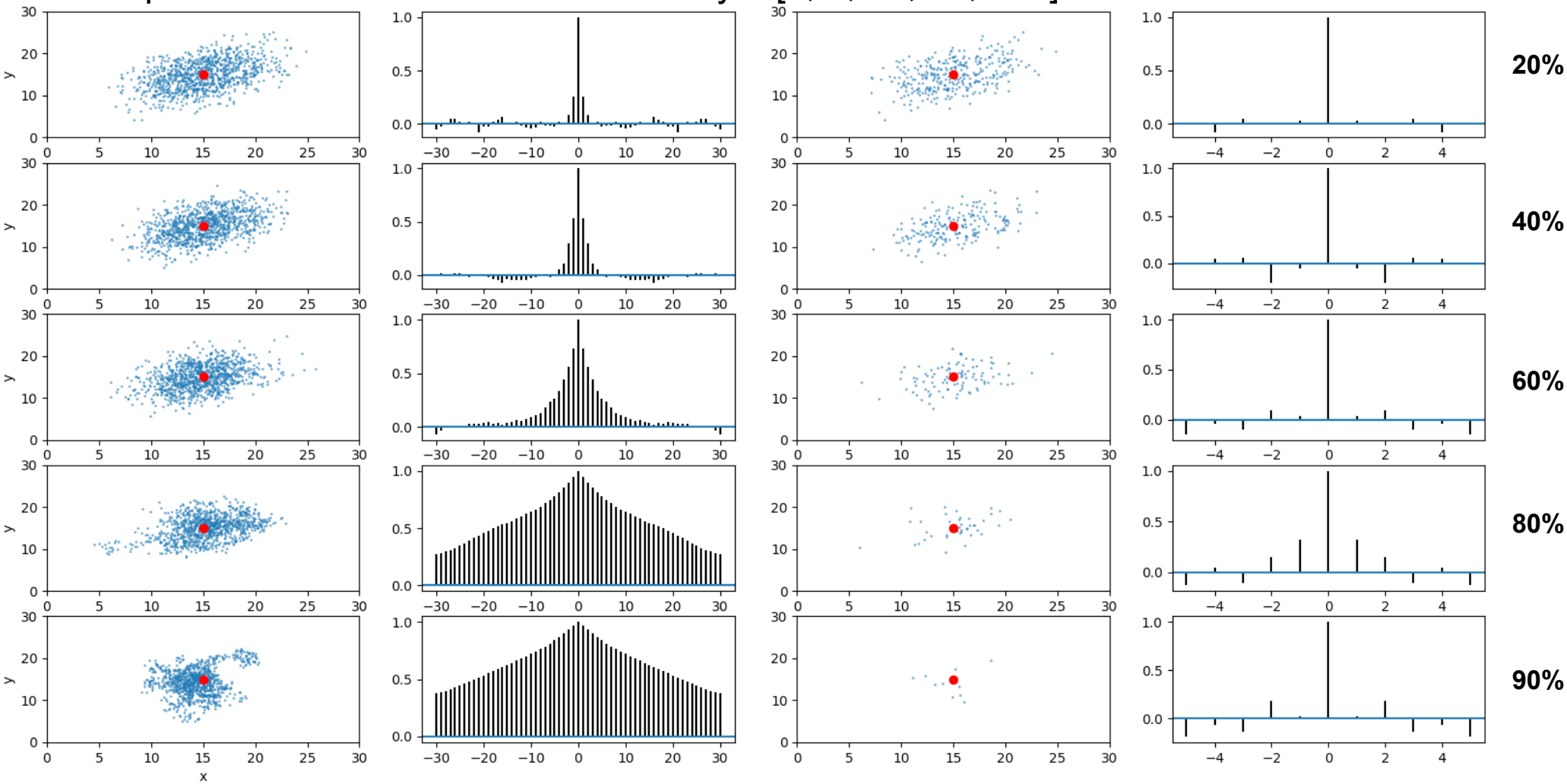
[1] Stefan Engblom, Vikram Sunkara:
Preconditioned Metropolis sampling as a strategy to improve efficiency in Posterior exploration.
IFAC-PapersOnLine 49-26 (2016) 089-094

[2] Vikram Sunkara, Max von Kleist:
Numerics for Bioinformaticians, Semester 1
http://systems-pharmacology.de/?page_id=724

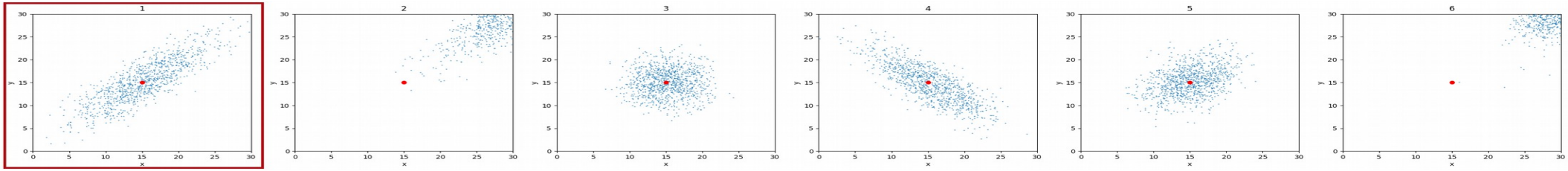
Thank you for your attention!

Summary - Correlation

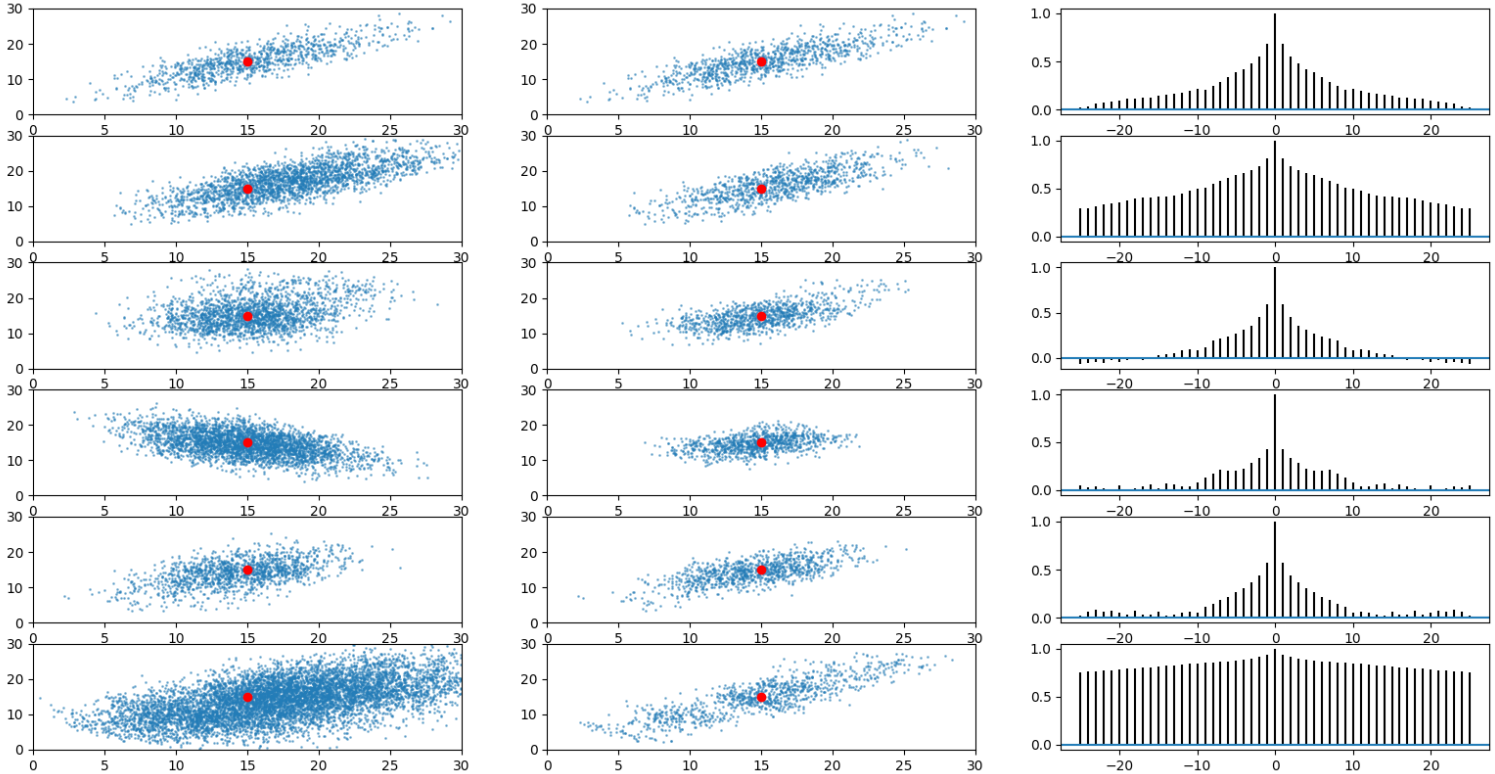
accepted = 1000 correlationArray = [3, 5, 10, 25, 100]



Summary - Selected target distribution:



pre-samples | samples | correlation



same distributions: 1 with 1
pre acceptance rate: 40%
acceptance rate: **100%**

shifted distribution: 2 with 1
pre acceptance rate: 40%
acceptance rate: **40%**

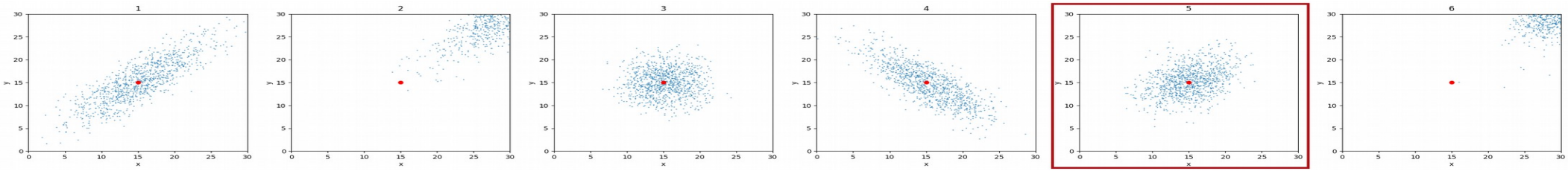
circular distribution: 3 with 1
pre acceptance rate: 40%
acceptance rate: **50%**

twisted distribution: 4 with 1
pre acceptance rate: 40%
acceptance rate: **30%**

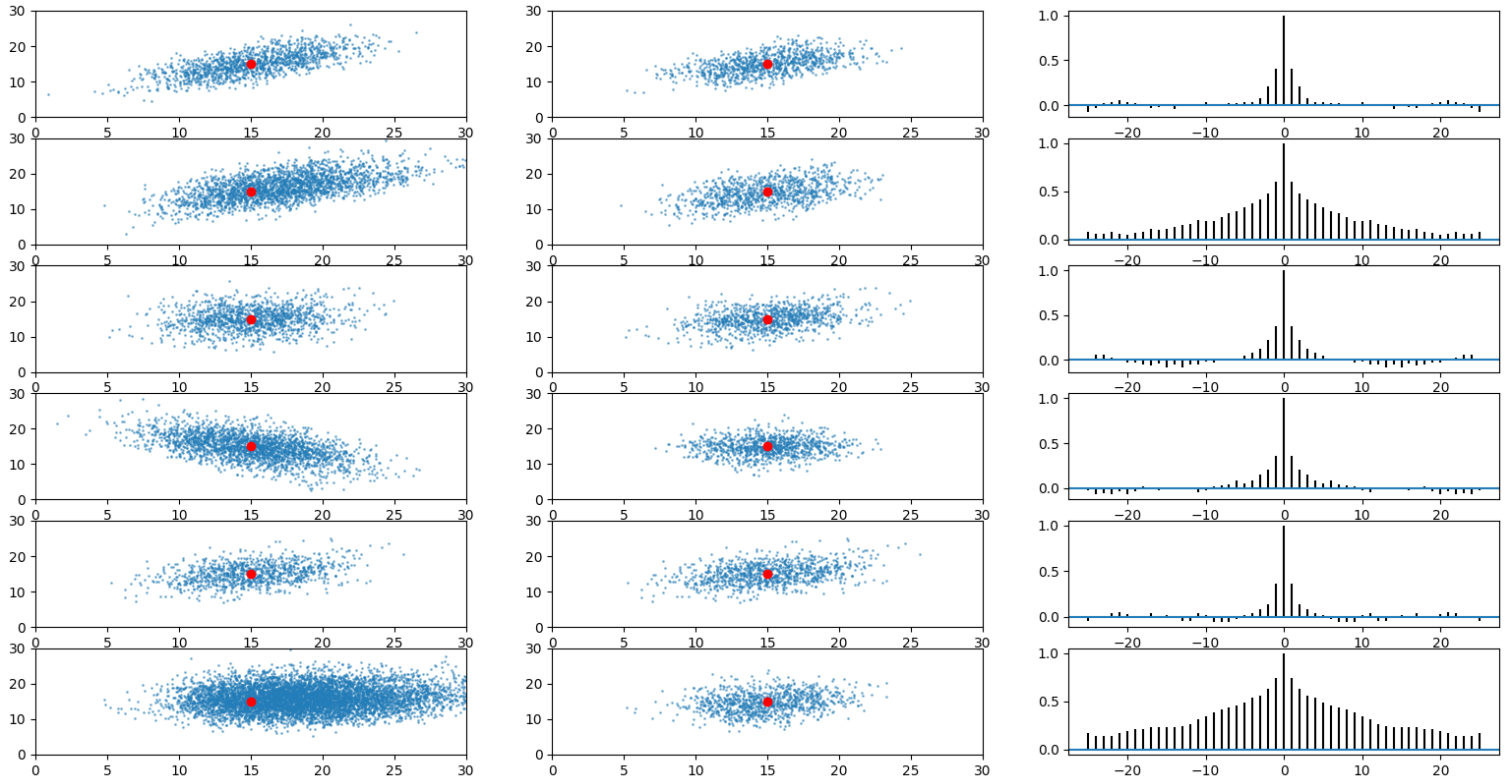
5 with 1
pre acceptance rate: 40%
acceptance rate: **70%**

6 with 1
pre acceptance rate: 40%
acceptance rate: **13%**

Summary - Selected target distribution:



pre-samples | samples | correlation



1 with 5
pre acceptance rate: 40%
acceptance rate: **65%**

2 with 5
pre acceptance rate: 40%
acceptance rate: **40%**

circular distribution: 3 with 5
pre acceptance rate: 40%
acceptance rate: **80%**

4 with 5
pre acceptance rate: 40%
acceptance rate: **45%**

same distributions: 5 with 5
pre acceptance rate: 40%
acceptance rate: **100%**

shifted distribution: 6 with 5
pre acceptance rate: 40%
acceptance rate: **15%**