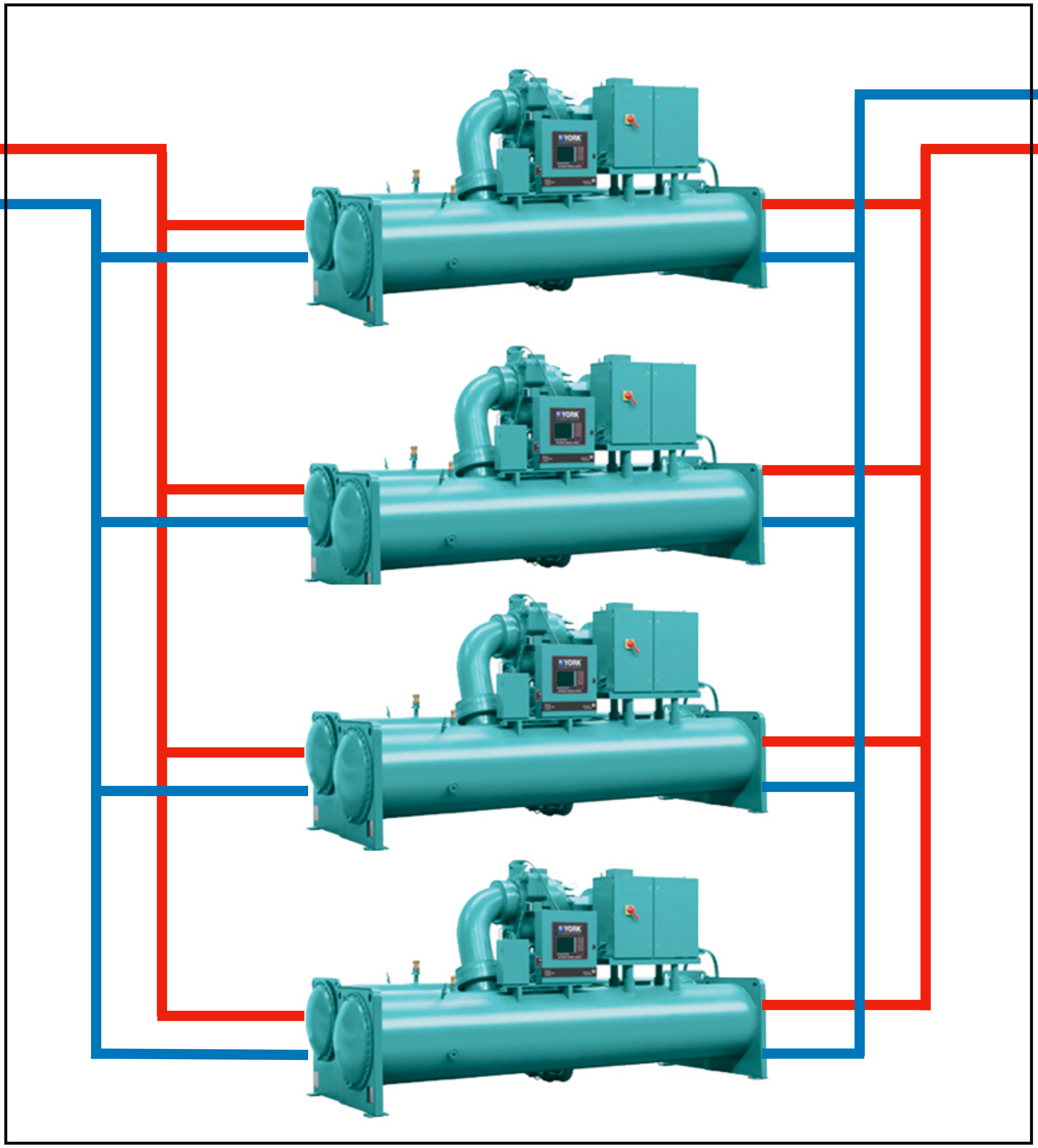
Predicting Industrial Chiller Performance

Dana Lindquist, PhD

Chiller = Heat Exchanger

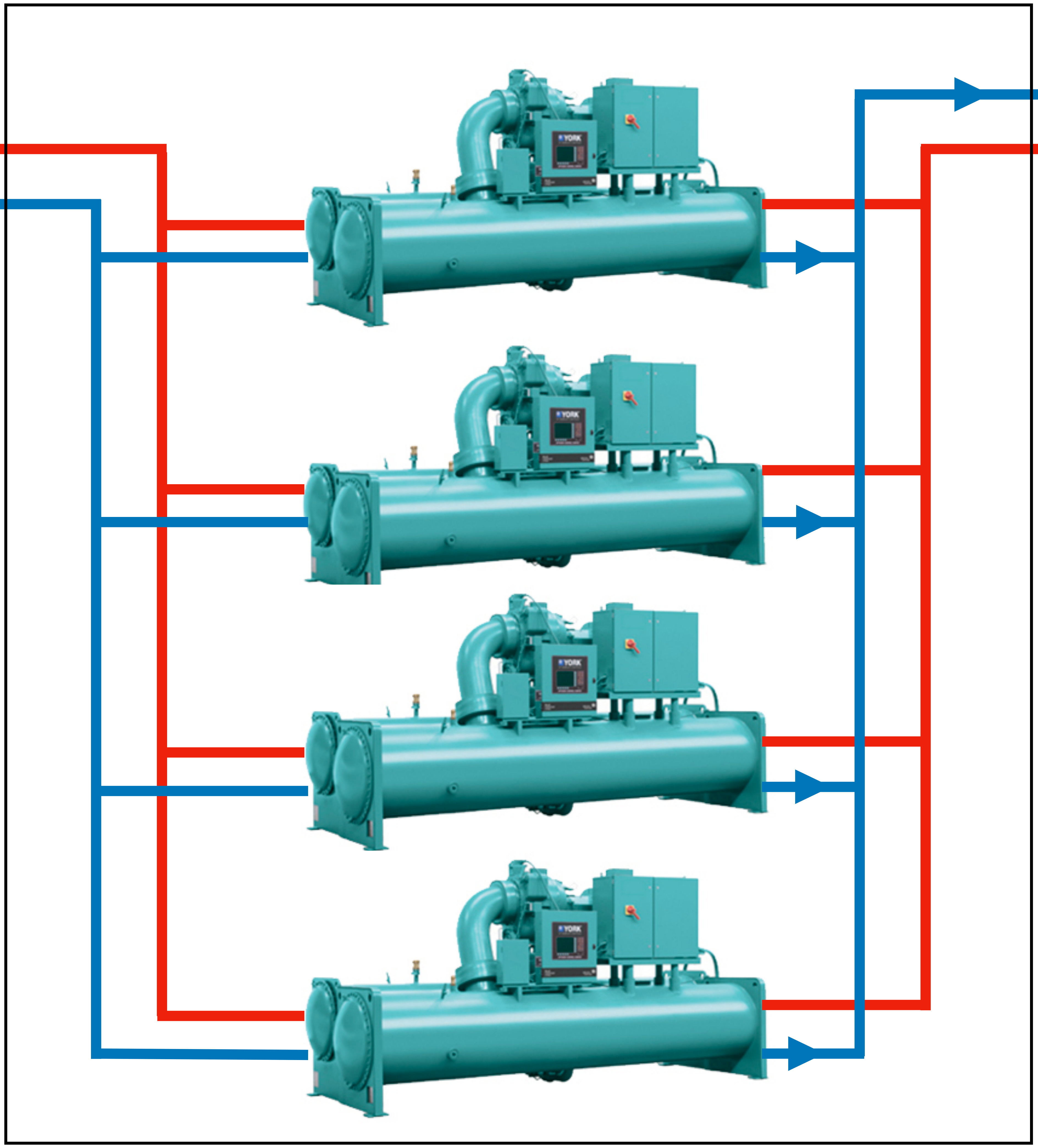
Cooling towers used to remove heat



Supply cold water to buildings

Chiller = Heat Exchanger

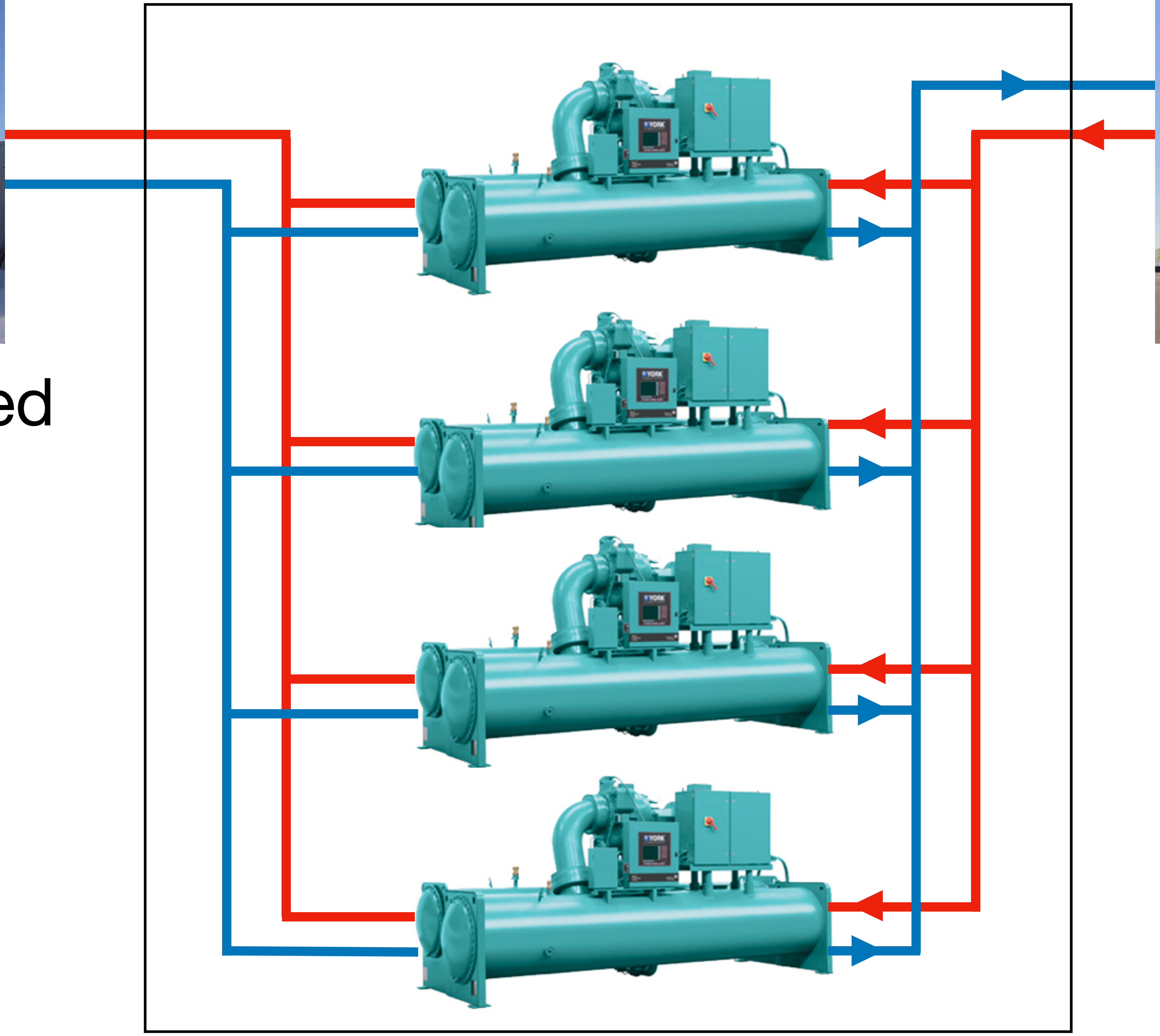
Cooling towers used
to remove heat



Supply cold water
to buildings

Chiller = Heat Exchanger

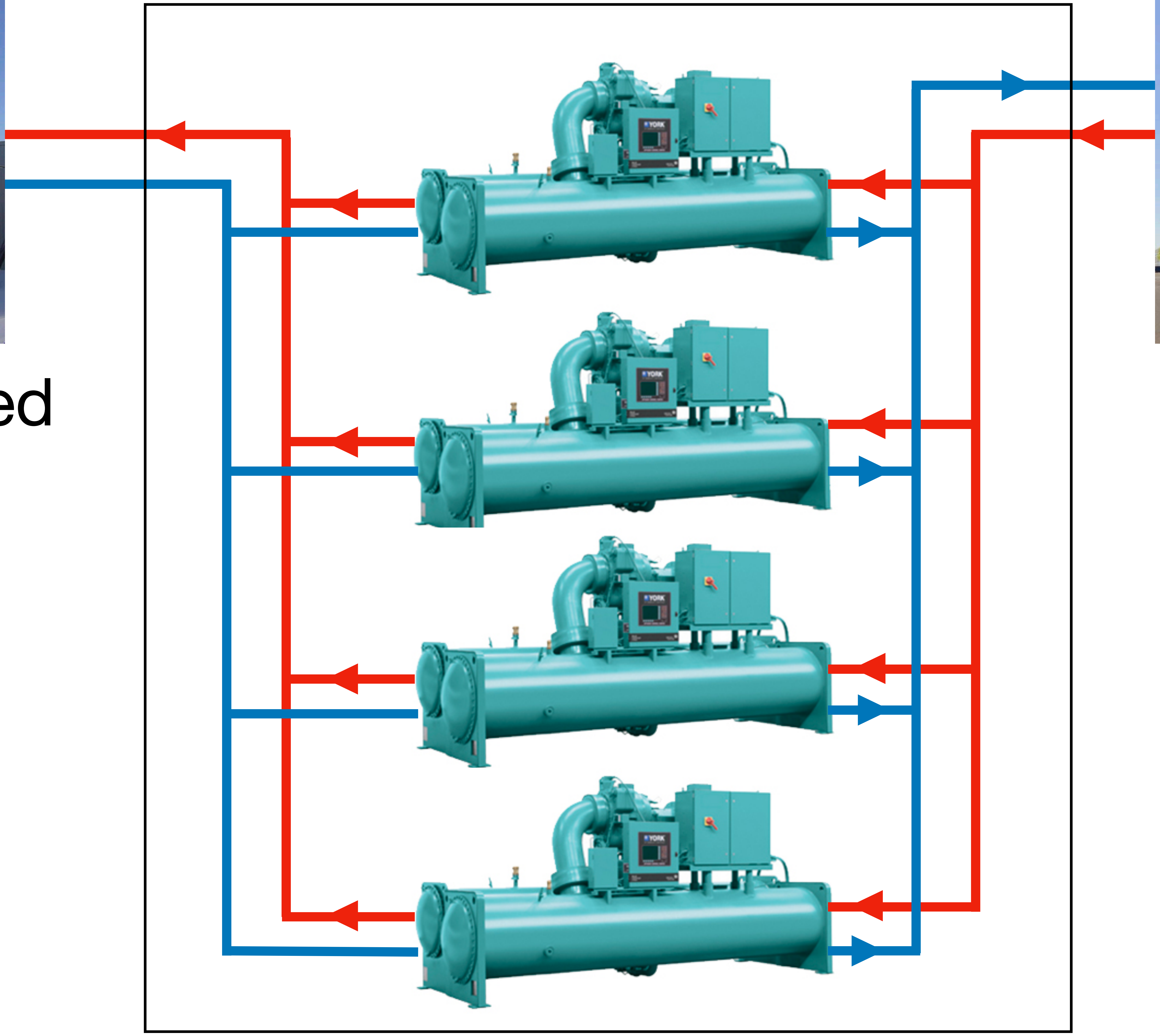
Cooling towers used to remove heat



Supply cold water to buildings

Chiller = Heat Exchanger

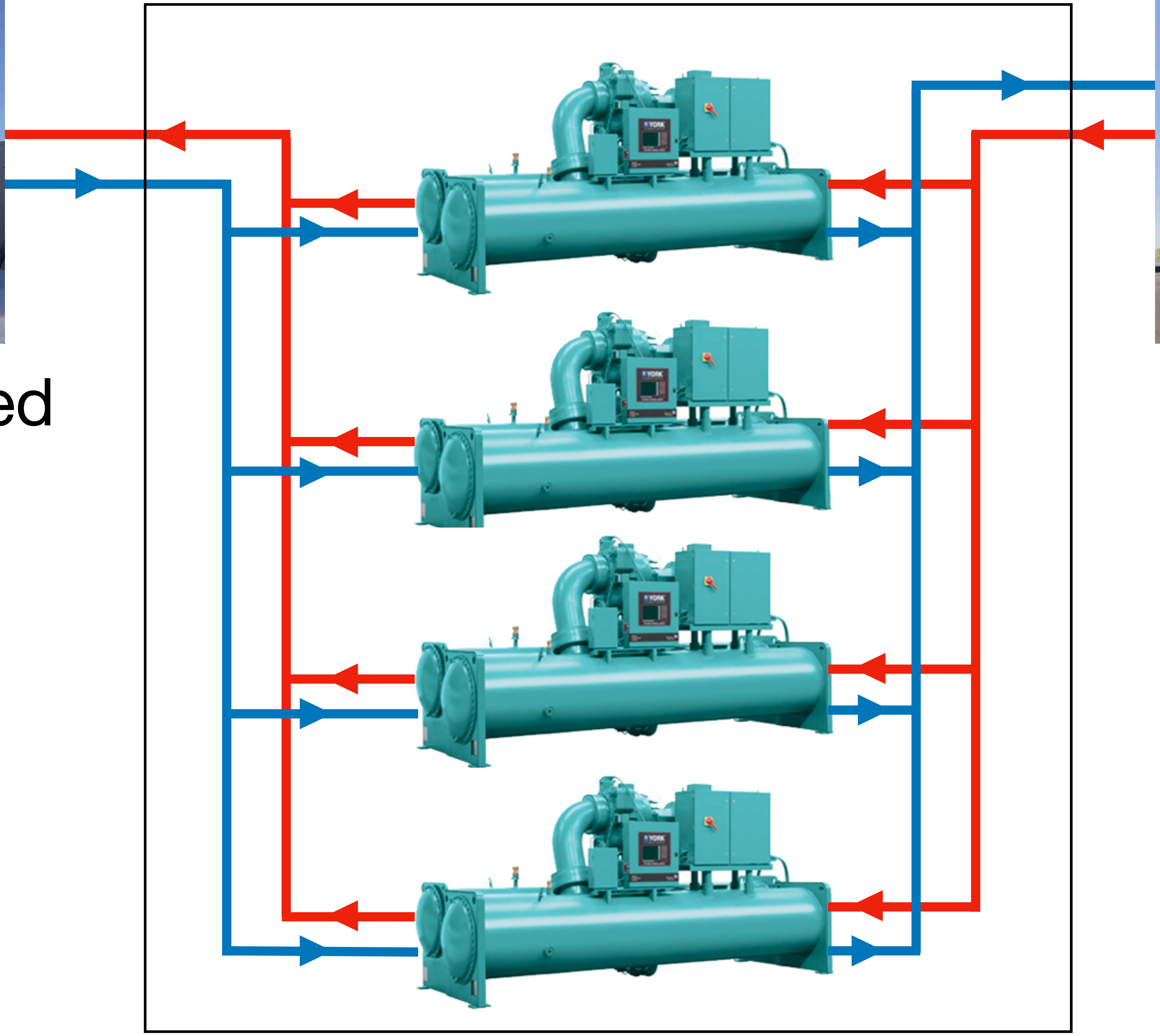
Cooling towers used
to remove heat



Supply cold water
to buildings

Chiller = Heat Exchanger

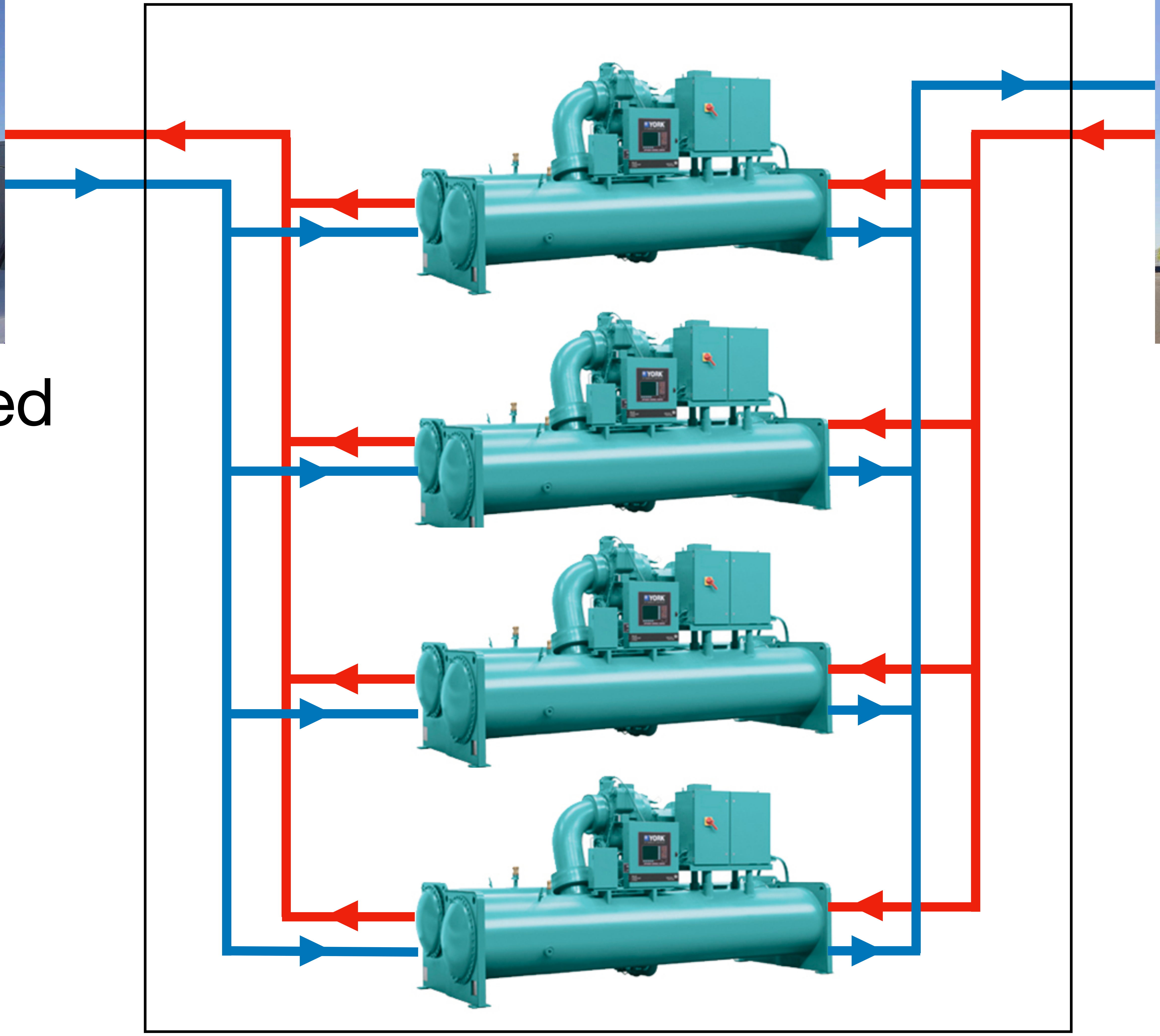
Cooling towers used to remove heat



Supply cold water to buildings

Chiller = Heat Exchanger

Cooling towers used to remove heat



Supply cold water to buildings

Thank you to Optimum Energy For providing chiller data

Chiller Plants

- Chiller manufacturers provide some limited design specifications when they sell a Chiller (usually end up collecting dust in a filing cabinet).

Chiller Plants

- Chiller manufacturers provide some limited design specifications when they sell a Chiller (usually end up collecting dust in a filing cabinet).
- Most Chiller Plant operators do not think about efficiency.

Chiller Plants

- Chiller manufacturers provide some limited design specifications when they sell a Chiller (usually end up collecting dust in a filing cabinet).
- Most Chiller Plant operators do not think about efficiency.
- Optimum Energy collects data on Chiller Plants.

Gather/Clean Data

Remove data when Chiller in ALARM or just starting or stopping

↓

Separate Into Individual Chillers

30 distinct but similar Chillers
8 different sizes & configurations

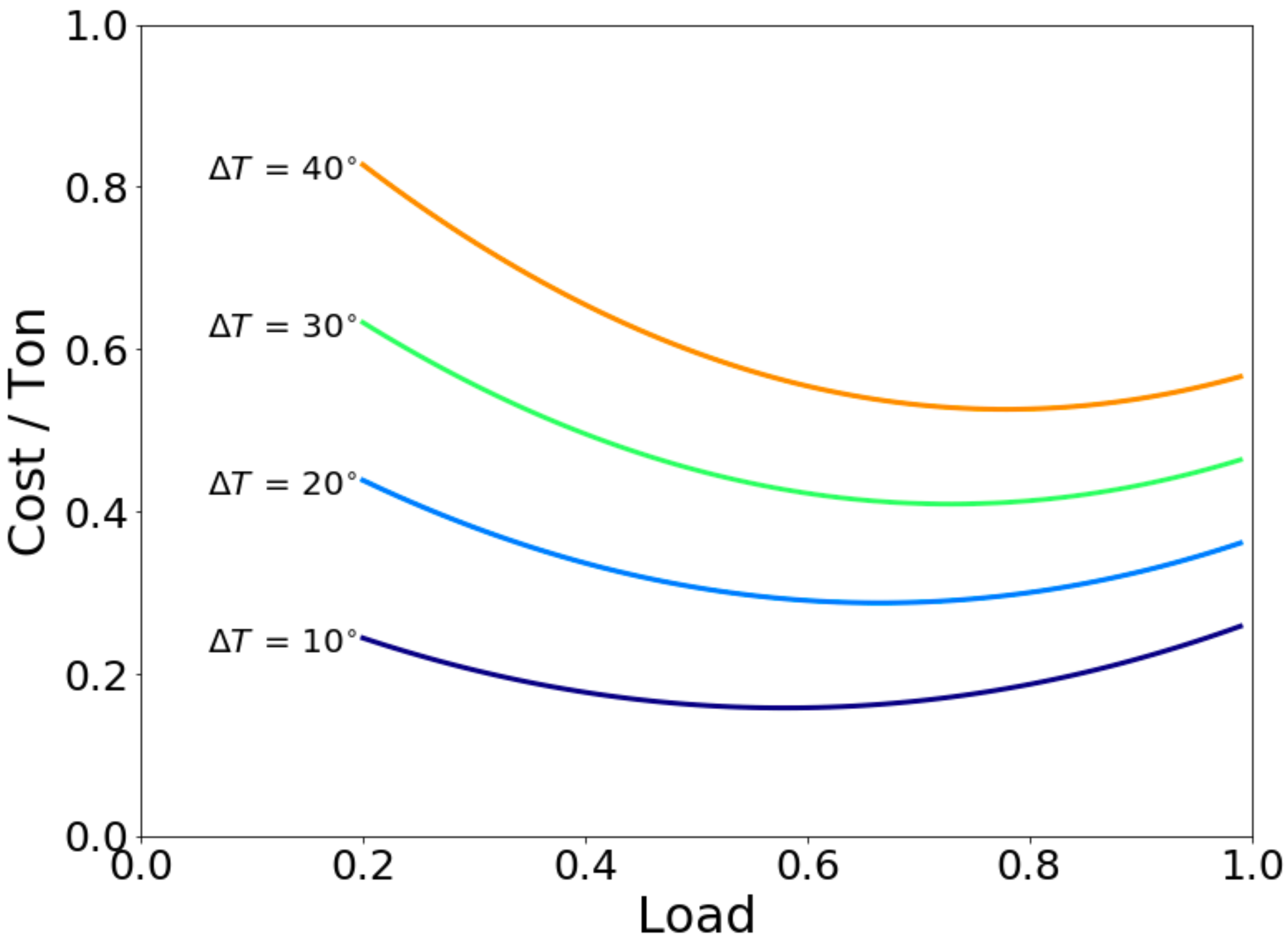
↓

Linear Regression for ONE Chller

Scikit-learn Linear Regression with LASSO Regularization

↓

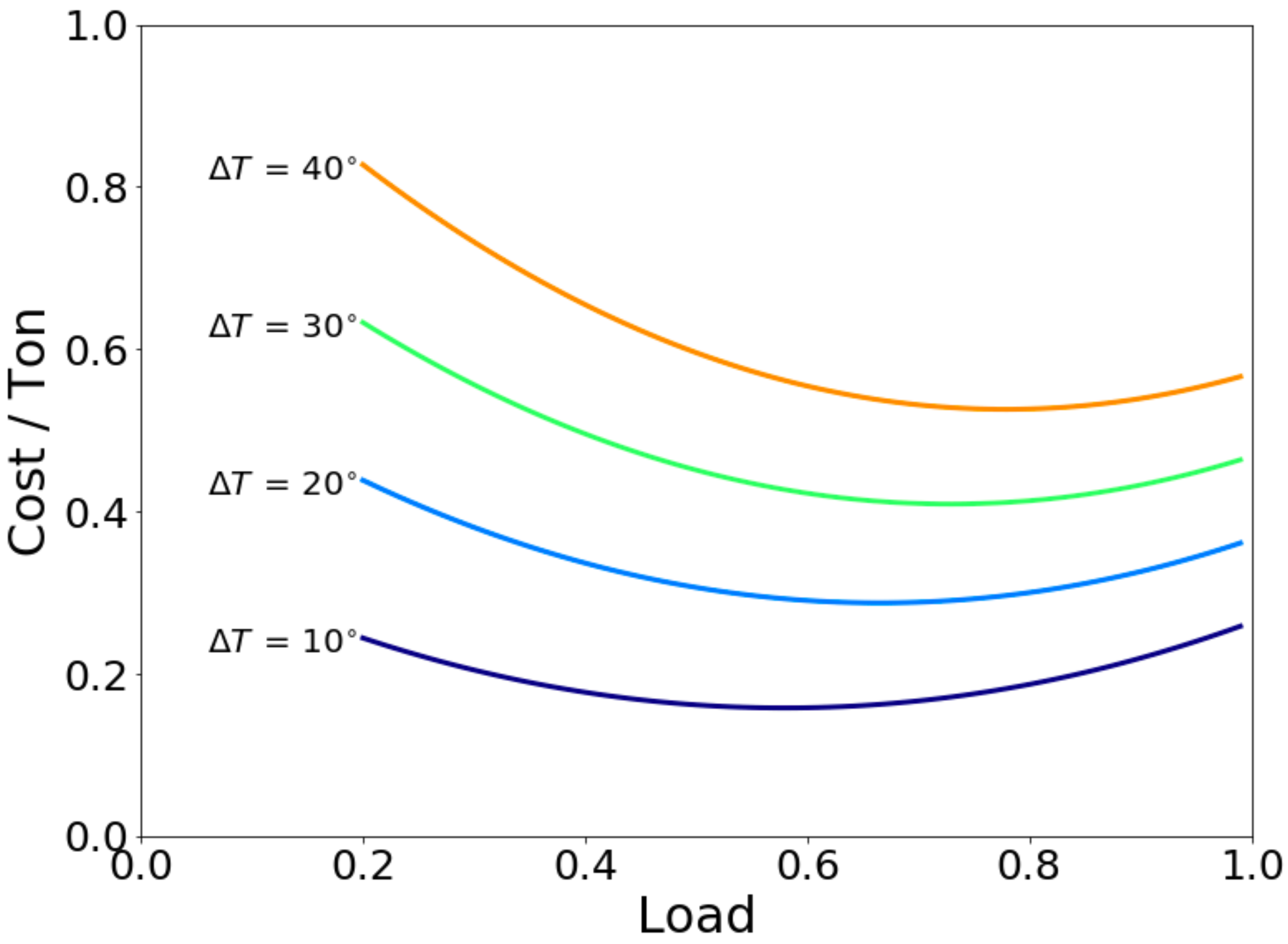
Just One Chiller



Ton = unit of refrigeration

Load = how hard the Chiller is working (fraction of max power)

Just One Chiller



Ton = unit of refrigeration

Load = how hard the Chiller is working (fraction of max power)

Quadratic in Load

$$\text{Cost/Ton} = (A + B\,\Delta\text{T}) + (C + D\,\Delta\text{T})\,\text{Load} + (E + D\,\Delta\text{T})\,\text{Load}^2$$

Want to predict *A, B, C, D, E & F* based on chiller characteristics

Combine ALL Chillers in to One Dataset

1 million data points

Add Chiller Characteristics to Dataset

Rated Tons, Variable Speed or not, . . .

Determine constants in Quadratic based on Chiller Characteristics

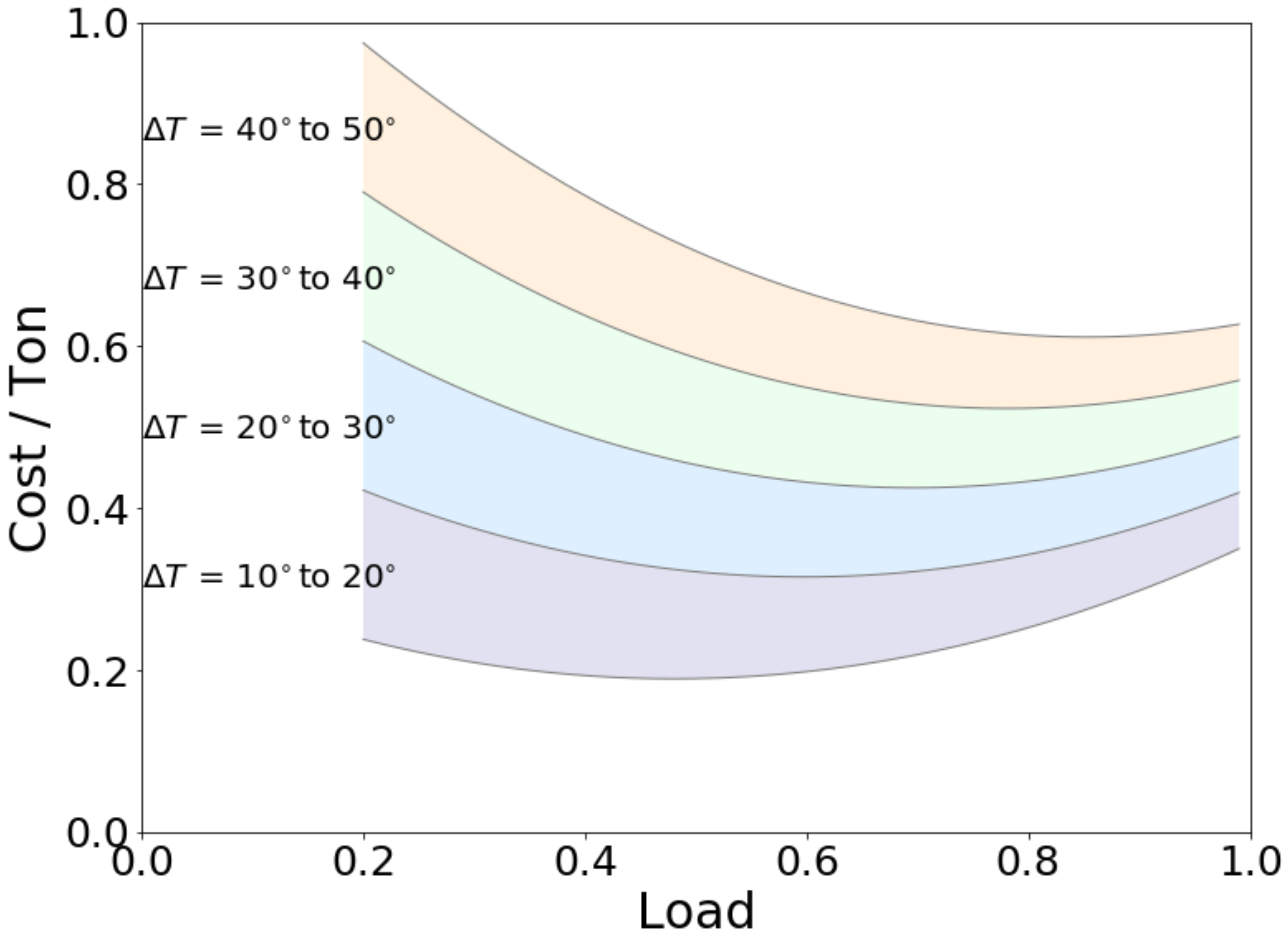
1. **Linear Regression**
2. **Bayesian Model using PYMC3**

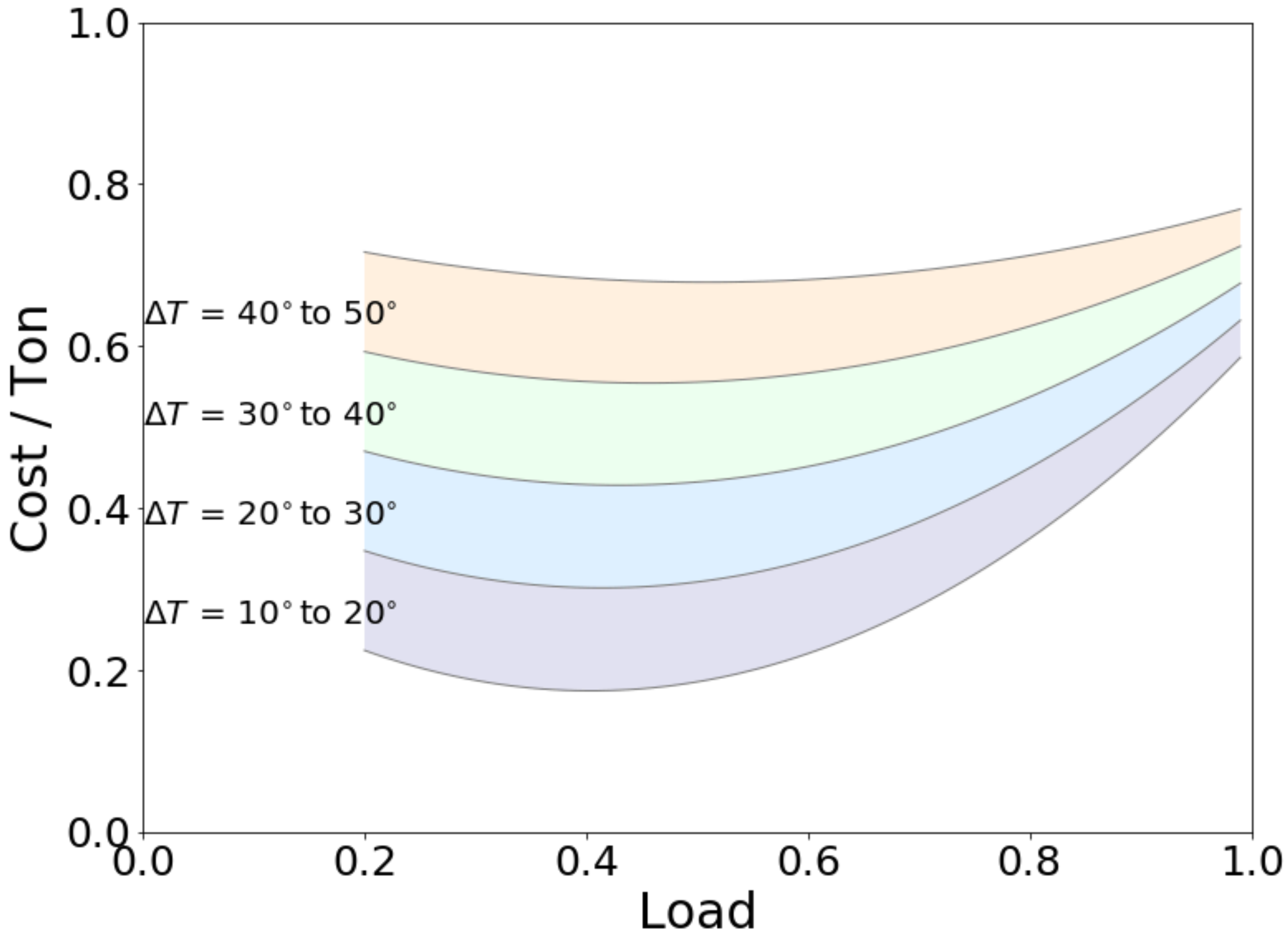
Predict quadratic constants

Has the ability to extend this analysis to more chiller characteristics

Results

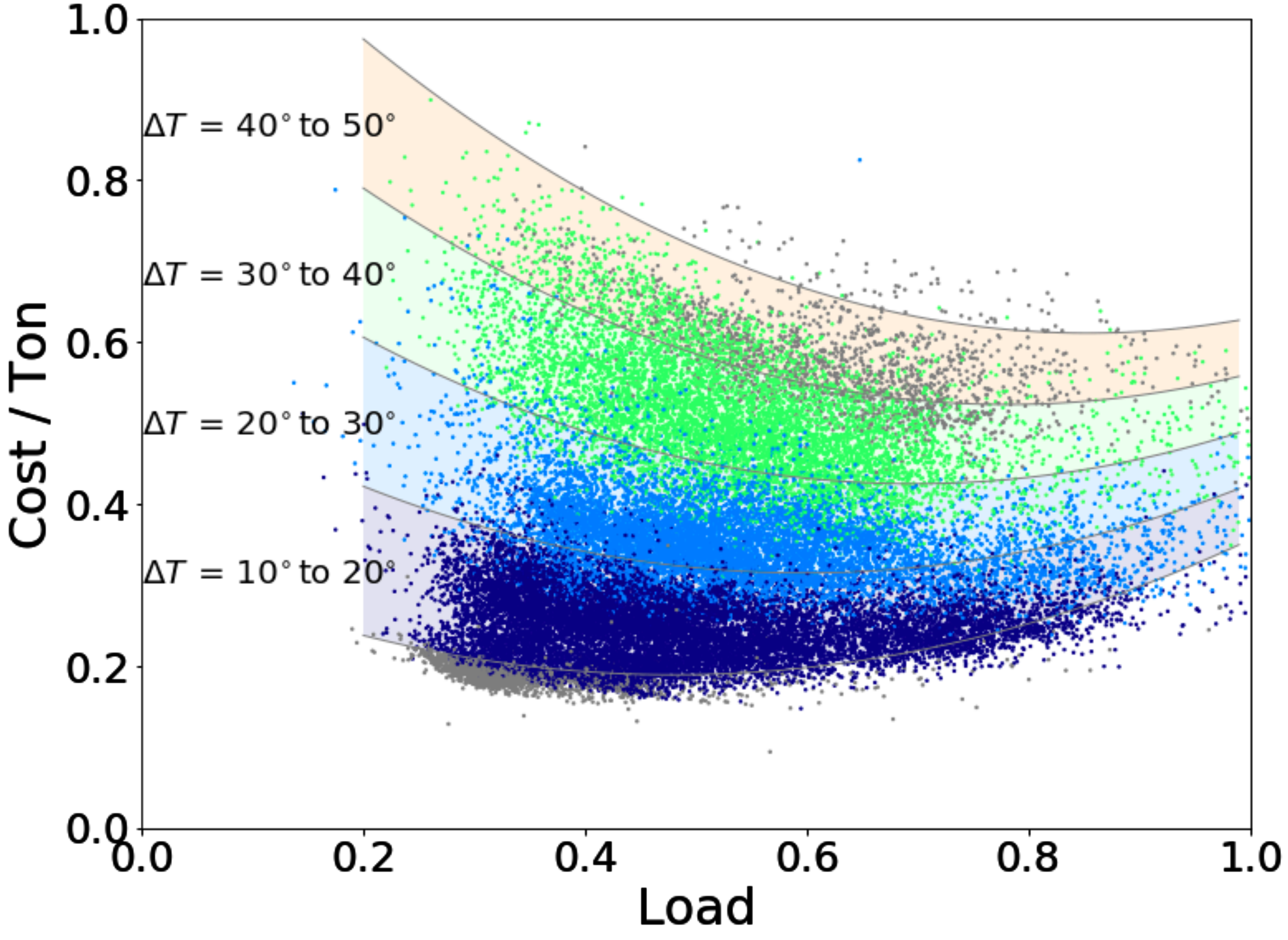
Two DIFFERENT 900 Ton Chillers

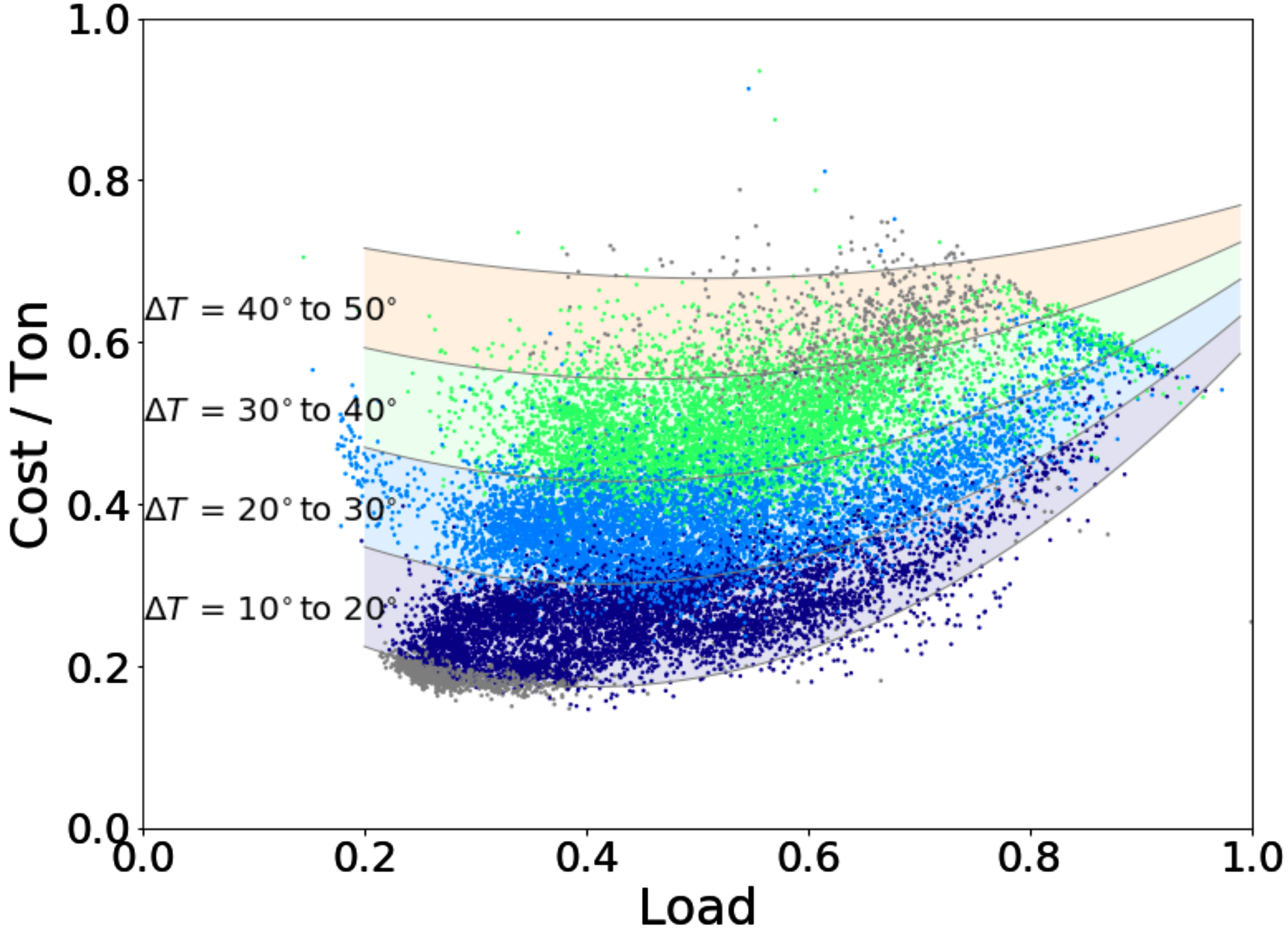




Results

Two DIFFERENT 900 Ton Chillers

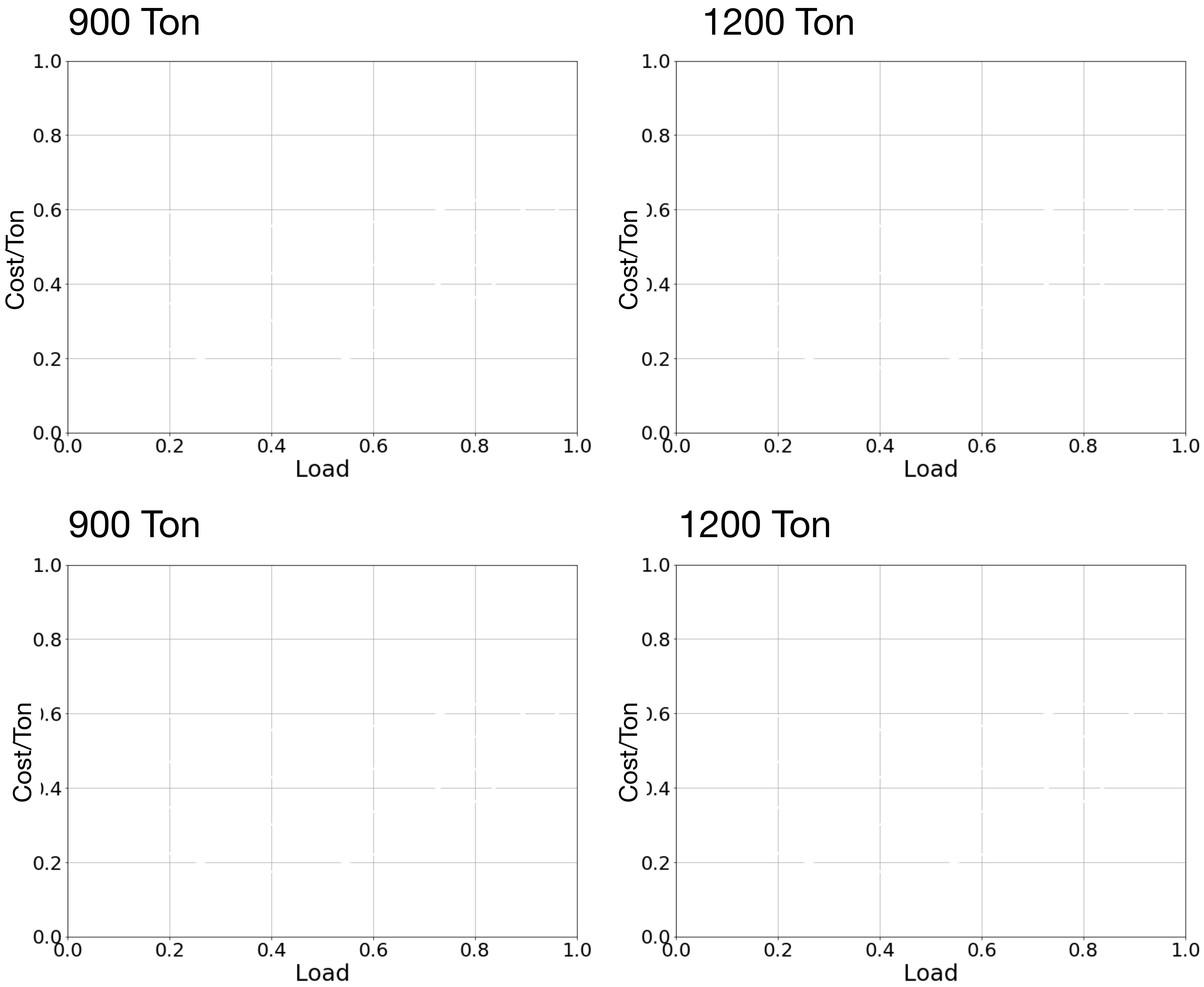




How to Run the Plant Efficiently?

Requirement:
  2400 Tons of cooling
  30° F ΔT

Goal: Lowest Cost

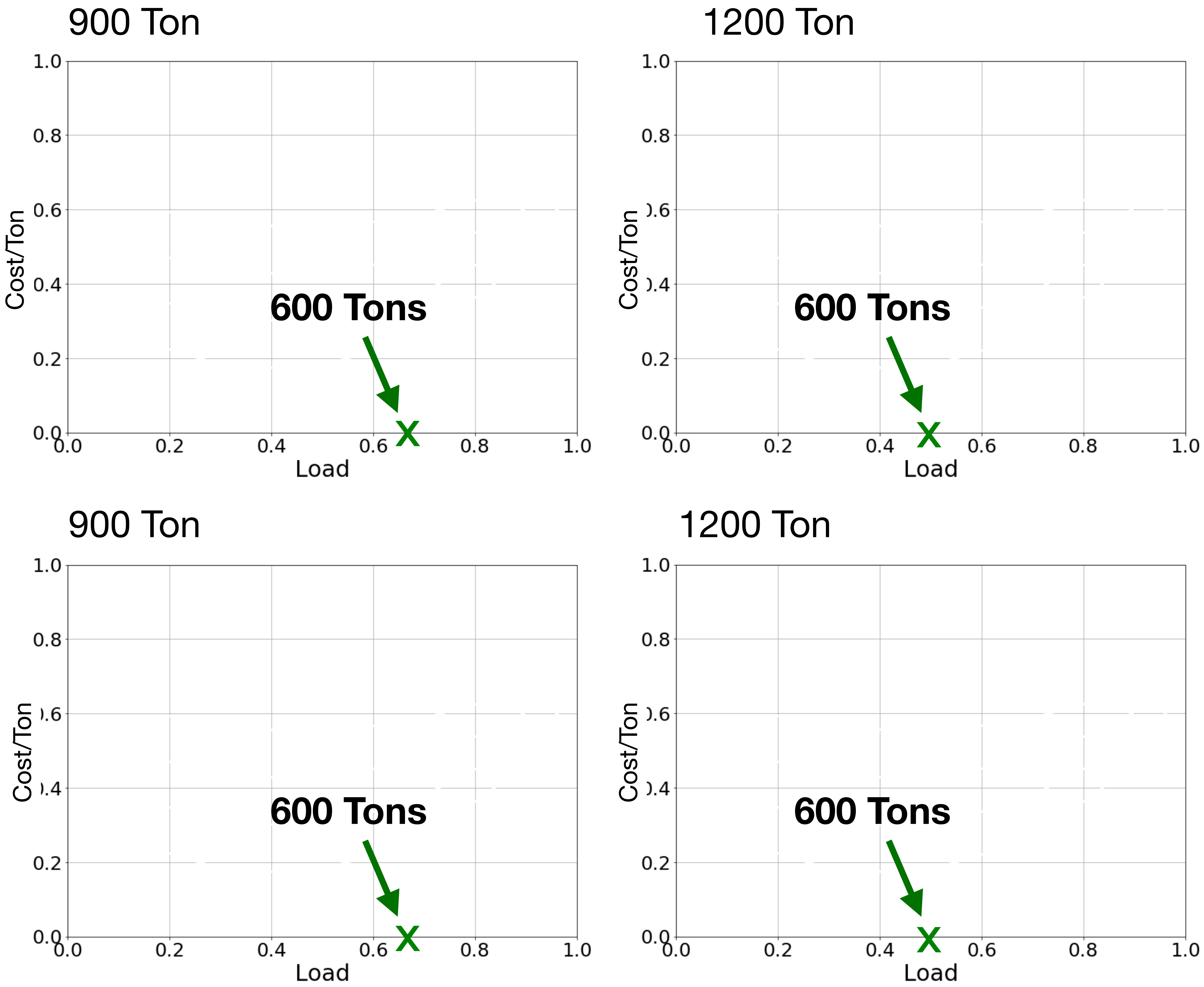


How to Run the Plant Efficiently?

Requirement:
  2400 Tons of cooling
  30° F ΔT

Goal: Lowest Cost

Run all 4 chillers at 600 Tons each

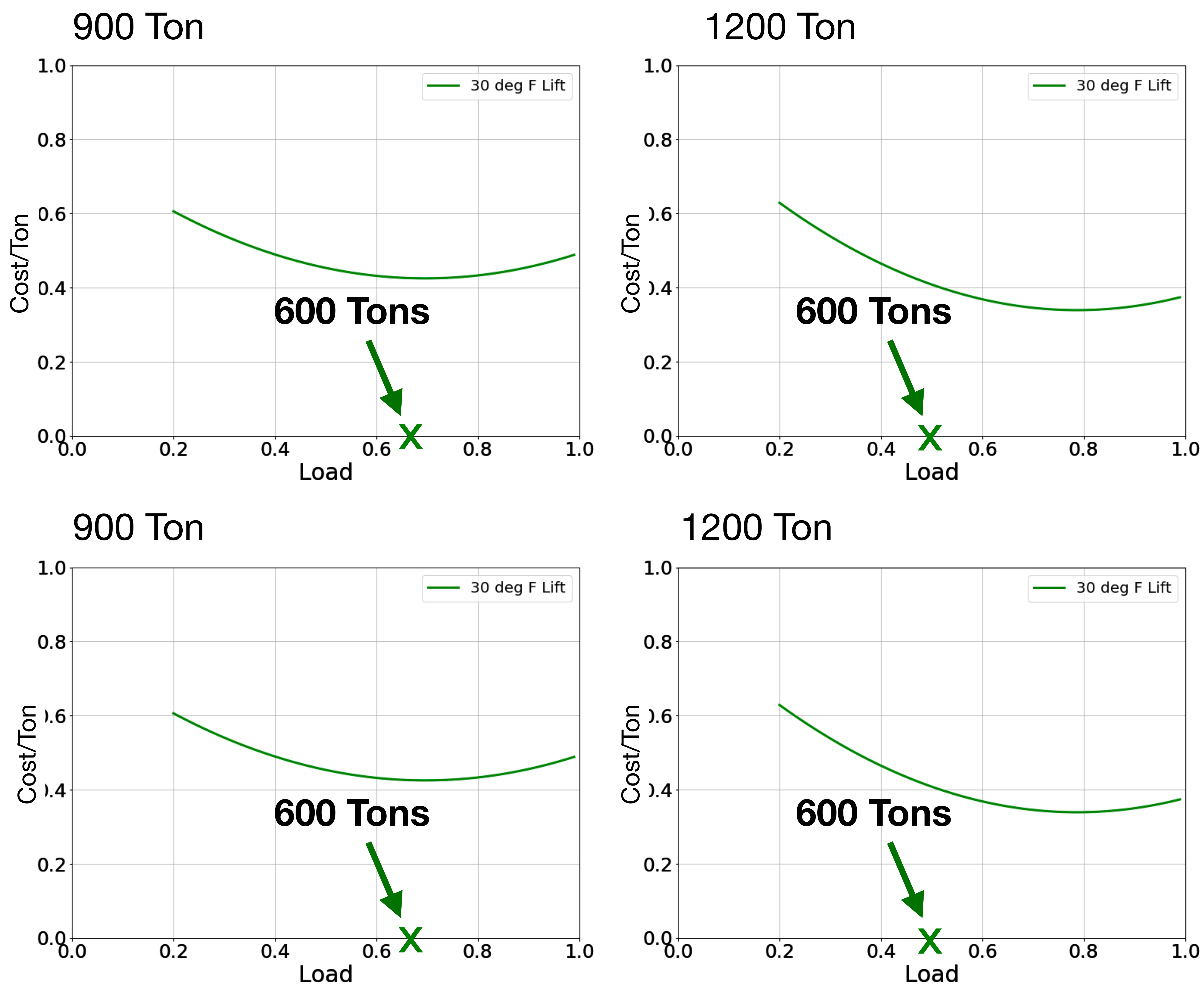


How to Run the Plant Efficiently?

Requirement:
2400 Tons of cooling
30° F ΔT

Goal: Lowest Cost

Run all 4 chillers at 600 Tons each



How to Run the Plant Efficiently?

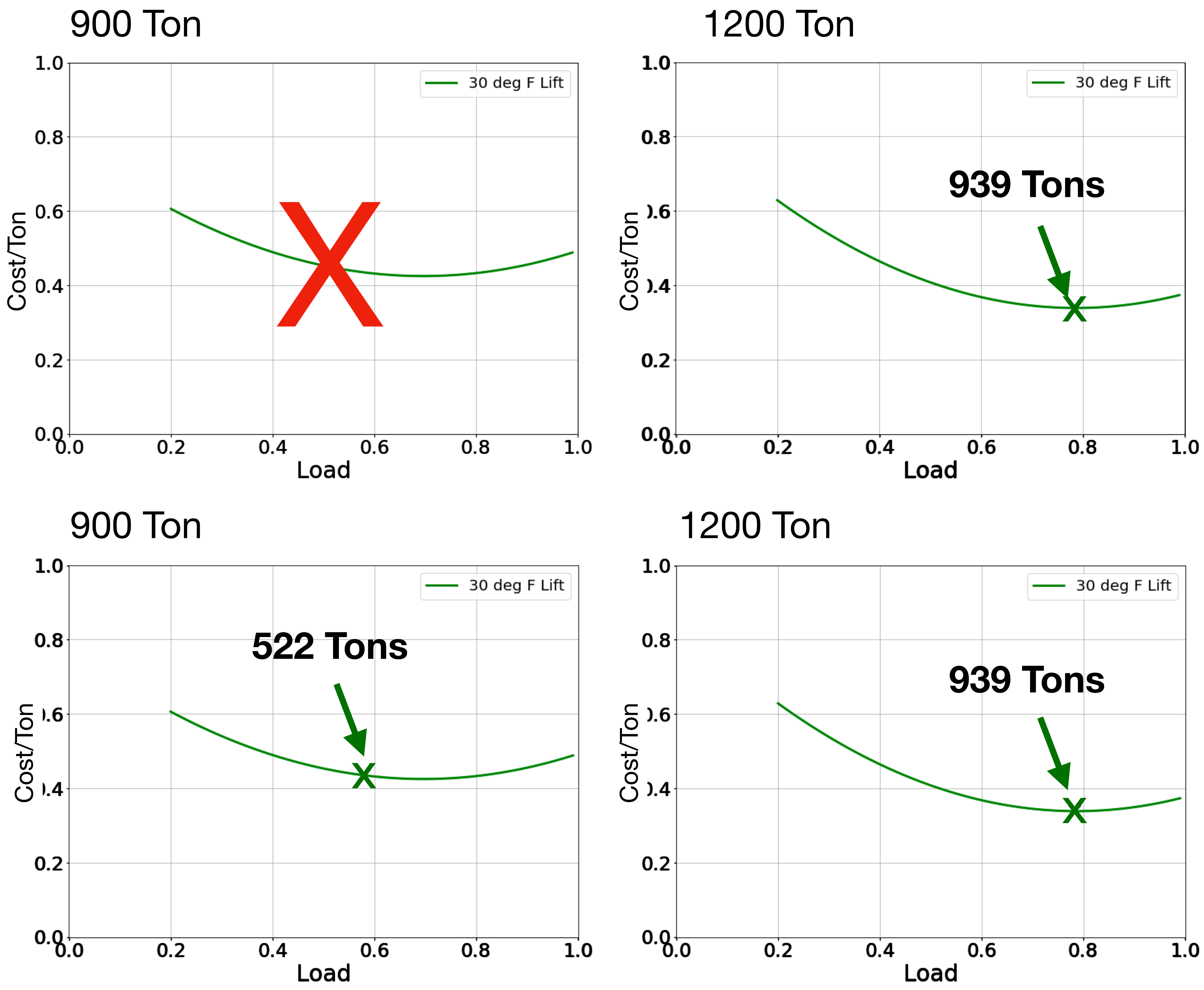
Requirement:

2400 Tons of cooling

30° F ΔT

Goal: Lowest Cost

Saving $12,000 (16%) per year



Conclusion

Given the data, we can predict Chiller performance

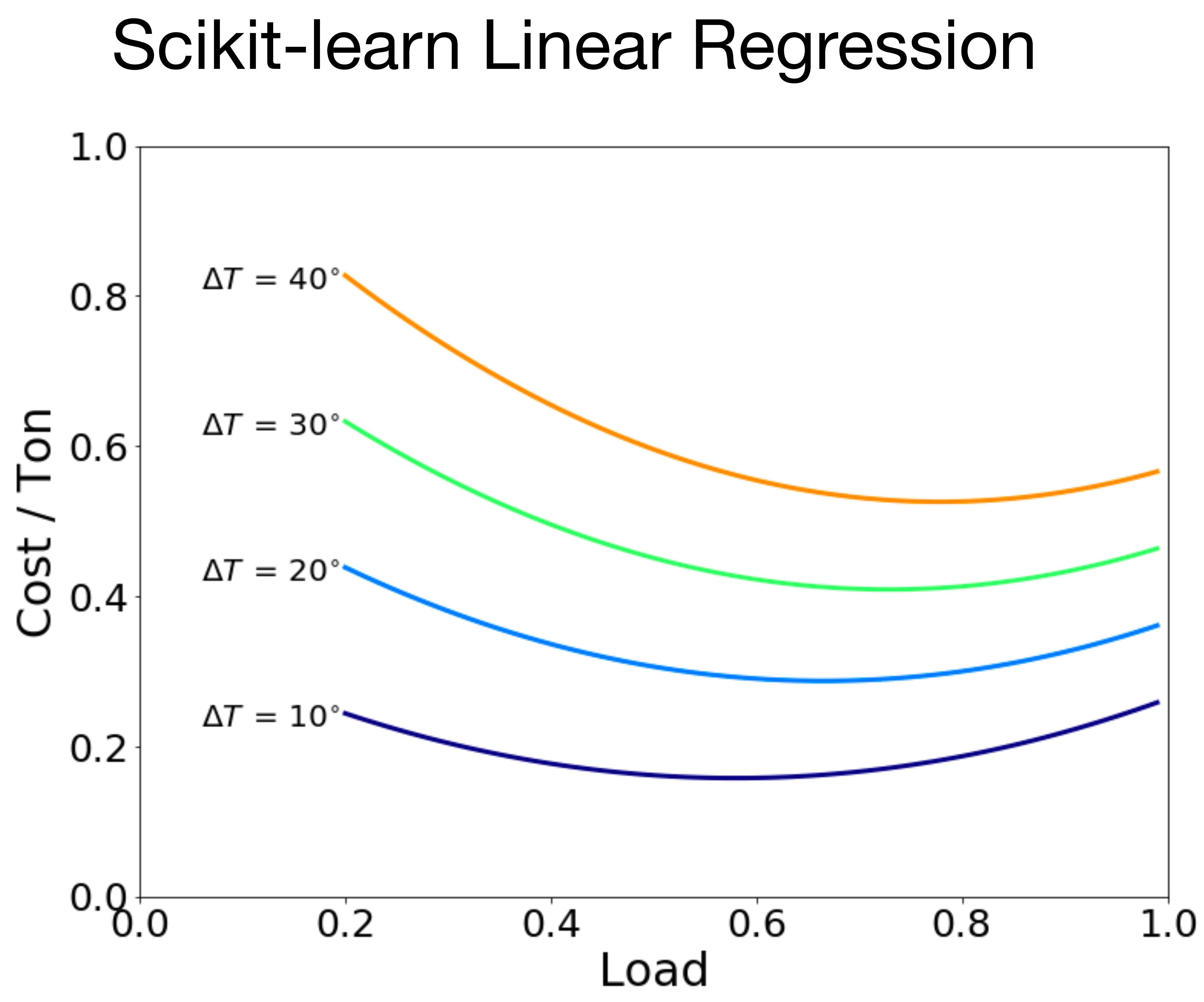
With a model for the system, we can make intelligent decisions on how to operate the system.

Thank You

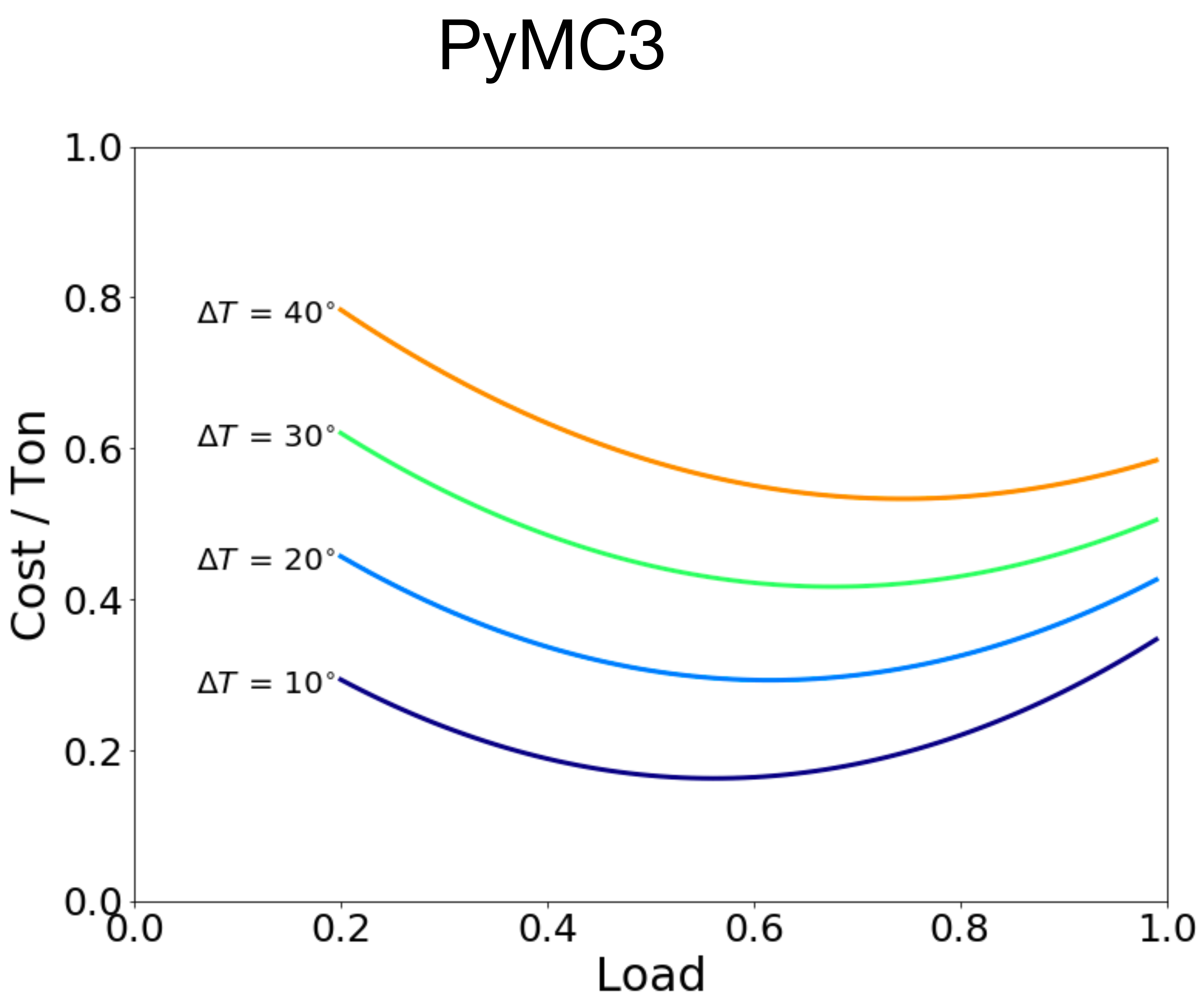
Dana Lindquist, PhD

danalindquist@silverbeach.com
www.linkedin.com/in/danalindquist
github.com/Stitchmaker

PyMC3



Scikit-learn Linear Regression
1.0
0.8
0.6
0.4
0.2
0.0
Cost / Ton
ΔT = 40°
ΔT = 30°
ΔT = 20°
ΔT = 10°
0.0
0.2
0.4
0.6
0.8
1.0
Load



PyMC3
1.0
0.8
0.6
0.4
0.2
0.0
Cost / Ton
ΔT = 40°
ΔT = 30°
ΔT = 20°
ΔT = 10°
0.0
0.2
0.4
0.6
0.8
1.0
Load

PyMC3

$$Cost/Ton_{obs} \;=\; (A \;+\; B\,\Delta T) \;+\; (C \;+\; D\,\Delta T)\,Load \;+\; (E \;+\; F\,\Delta T)\,Load^2$$

$$Cost/Ton \;=\; \mathcal{N}(Cost/Ton_{obs}, \mu, \sigma)$$

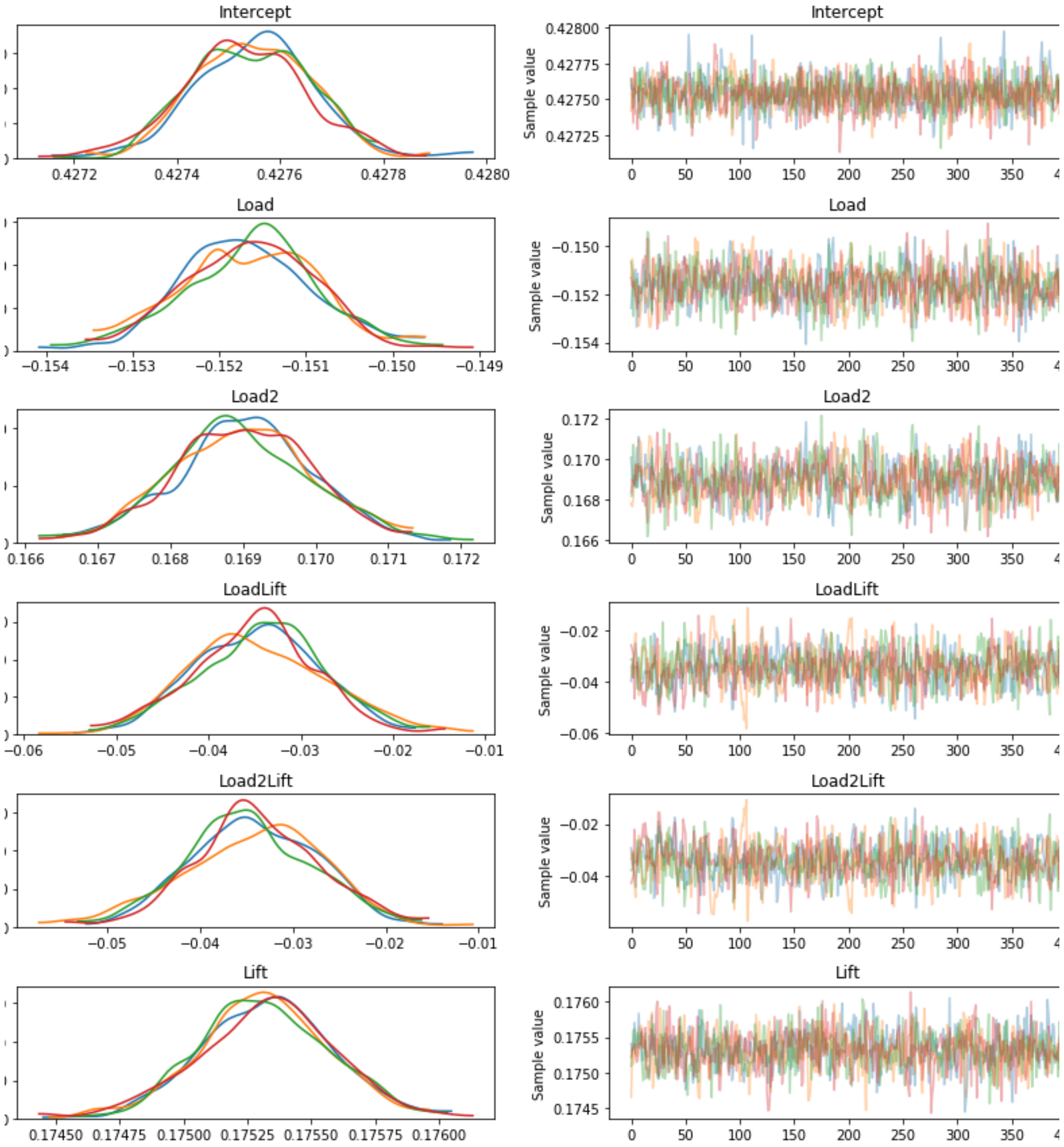
$$A, B, C, D, E\,\&\,F \;=\; \mathcal{N}(\mu, \sigma)$$

PyMC3

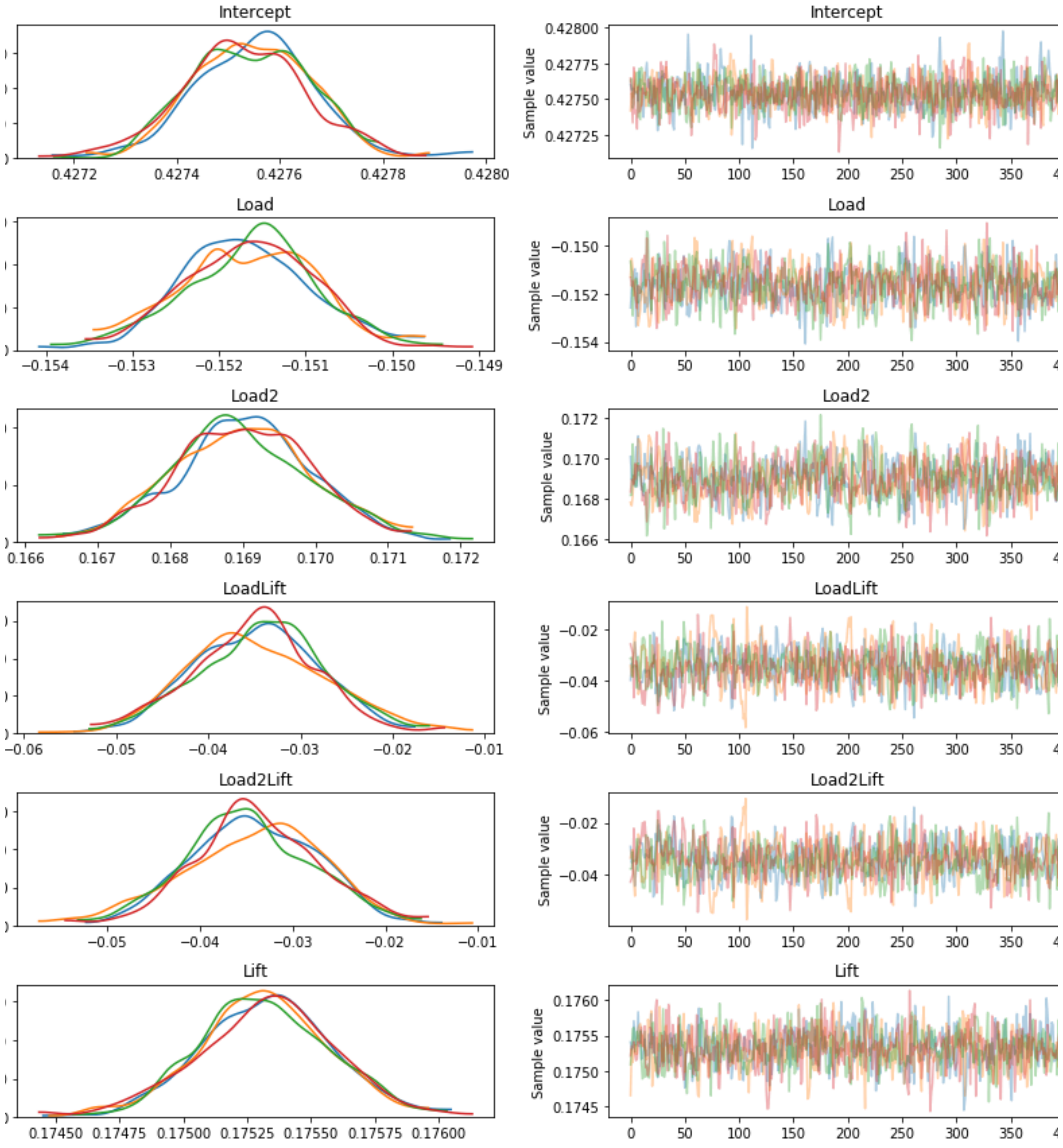
```
with Model() as model:
    sigma = HalfCauchy('sigma', beta=10)
    inter = Normal('Intercept', 0, sd=.01)
    D_coef = Normal('Load', 0, sd=.01)
    D2_coef = Normal('Load2', 0, sd=.01)
    DT_coef = Normal('LoadLift', 0, sd=.01)
    D2T_coef = Normal('Load2Lift', 0, sd=.01)
    T_coef = Normal('Lift', 0, sd=.01)

    # Define likelihood
    likelihood = Normal('kWTon', mu=inter +
                 D_coef * X['Load'] + D2_coef*X['Load^2'] + T_coef*X['DTLift']
                 + DT_coef*X['Load^2*DTLift'] +
                 D2T_coef*X[‘Load^2*DTLift'],
                 sd=sigma, observed=y)
```

PyMC3


Intercept
0.4272
0.4274
0.4276
0.4278
0.4280
Intercept
Sample value
0.42800
0.42775
0.42750
0.42725
Load
−0.154
−0.153
−0.152
−0.151
−0.150
−0.149
Load
−0.150
−0.152
−0.154
Load2
0.166
0.167
0.168
0.169
0.170
0.171
0.172
Load2
0.172
0.170
0.168
0.166
LoadLift
−0.06
−0.05
−0.04
−0.03
−0.02
−0.01
LoadLift
−0.02
−0.04
−0.06
Load2Lift
−0.05
−0.04
−0.03
−0.02
−0.01
Load2Lift
−0.02
−0.04
Lift
0.17450
0.17475
0.17500
0.17525
0.17550
0.17575
0.17600
Lift
0.1760
0.1755
0.1750
0.1745
0
50
100
150
200
250
300
350

PyMC3



Intercept
Intercept
Sample value
Load
Load
Sample value
Load2
Load2
Sample value
LoadLift
LoadLift
Sample value
Load2Lift
Load2Lift
Sample value
Lift
Lift
Sample value