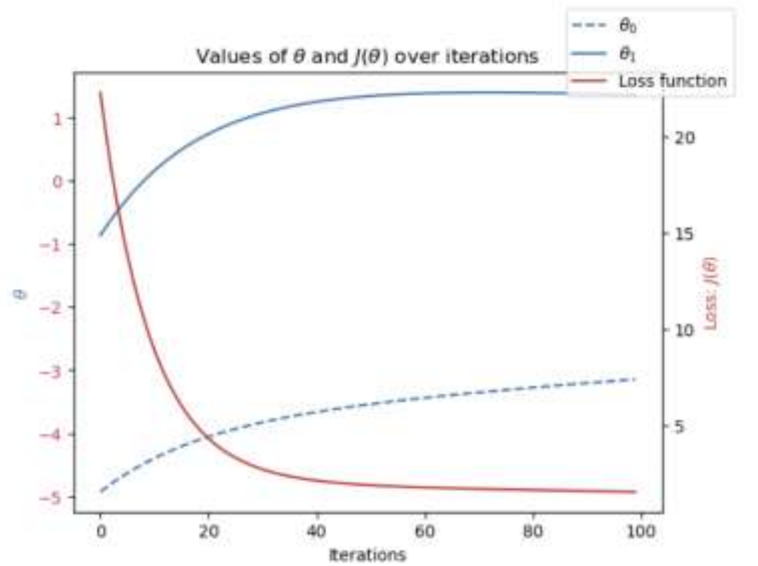
Rhys Ingalls

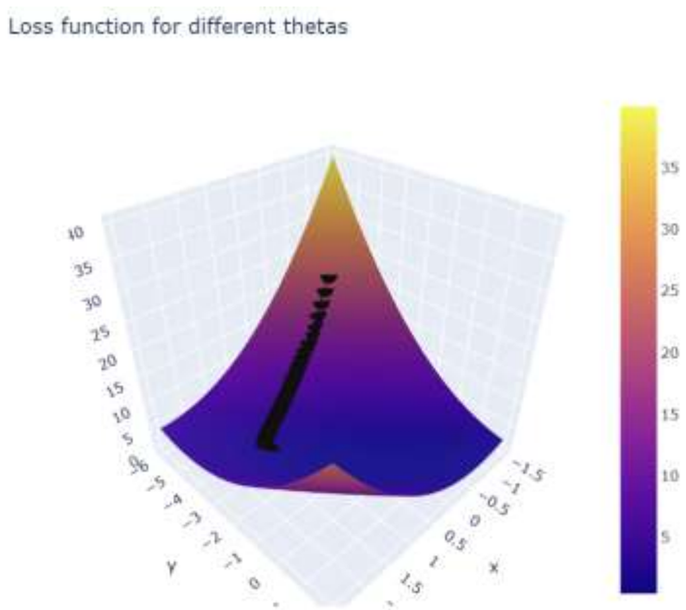
Assignment 1.3

Station	**Year**	**θ_0 (initial)**	**θ_1 (initial)**	**α (Step Size)**	**Iterations**	**Final θ_0**	**Final θ_1**	**Convergence Behavior**
Basel	1961	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Smooth, steady convergence; flattened near minimum
Basel	1993	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Similar curve and slope pattern; consistent convergence
Basel	2017	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Stable, no overshooting; loss minimized smoothly
Gdańsk	1961	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Consistent loss reduction; path follows valley shape
Gdańsk	1993	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Similar performance; reached flat region by iteration 100
Gdańsk	2017	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Stable learning rate; clear convergence path
Madrid	1961	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Smooth descent, matching overall pattern
Madrid	1993	-5	-1	0.01	100	≈ -3.15	≈ 1.38	No divergence; consistent with other stations
Madrid	2017	-5	-1	0.01	100	≈ -3.15	≈ 1.38	Clean convergence; minimal final loss

Basel 1961

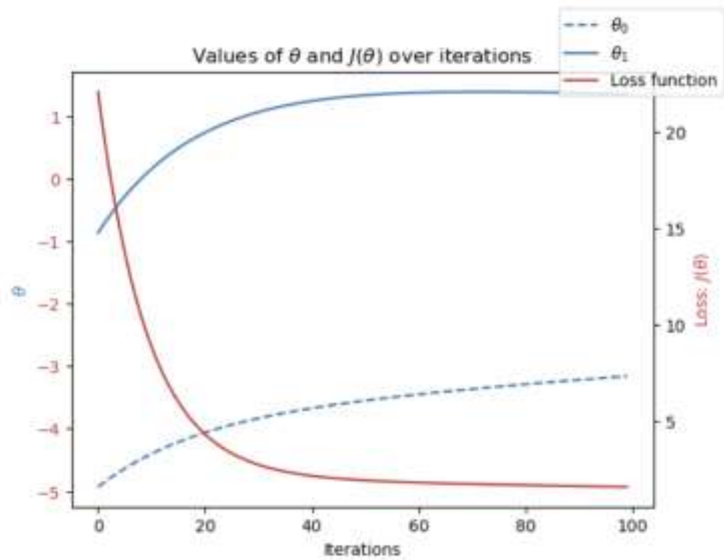


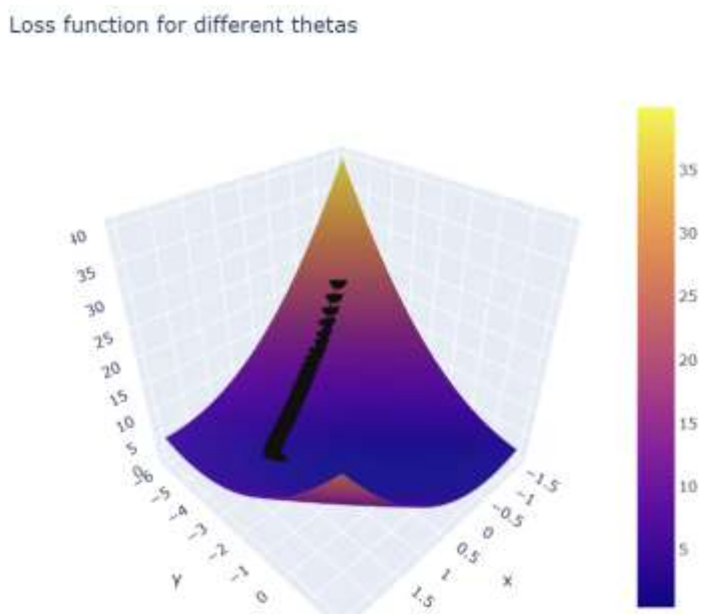
Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations



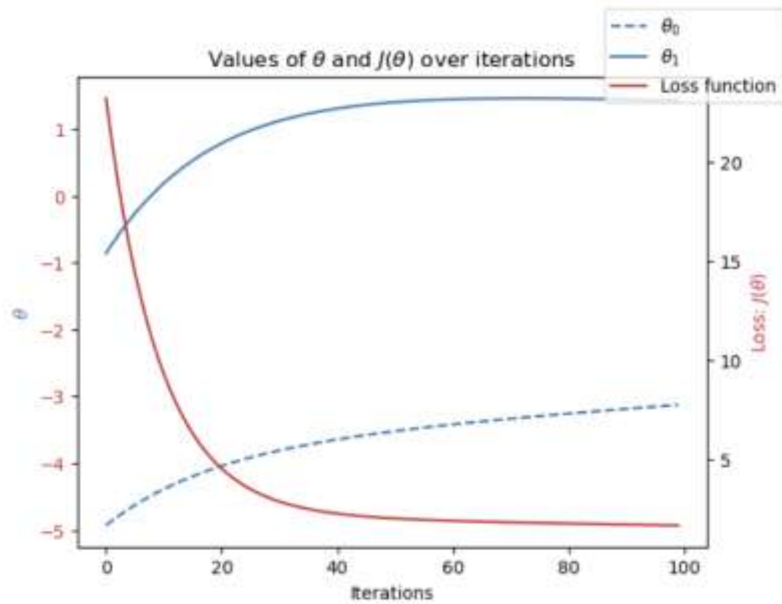
Loss function for different thetas
y
x

Basel 1993:

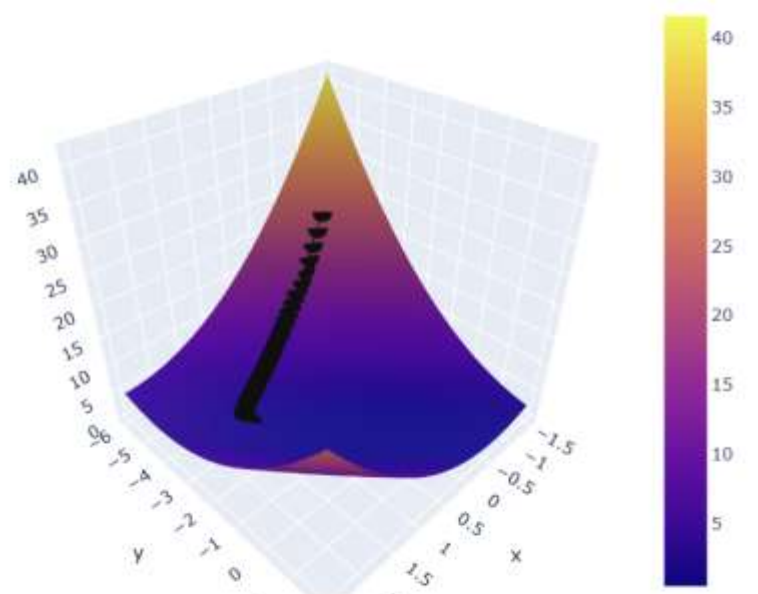

Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations


Loss function for different thetas
x
y

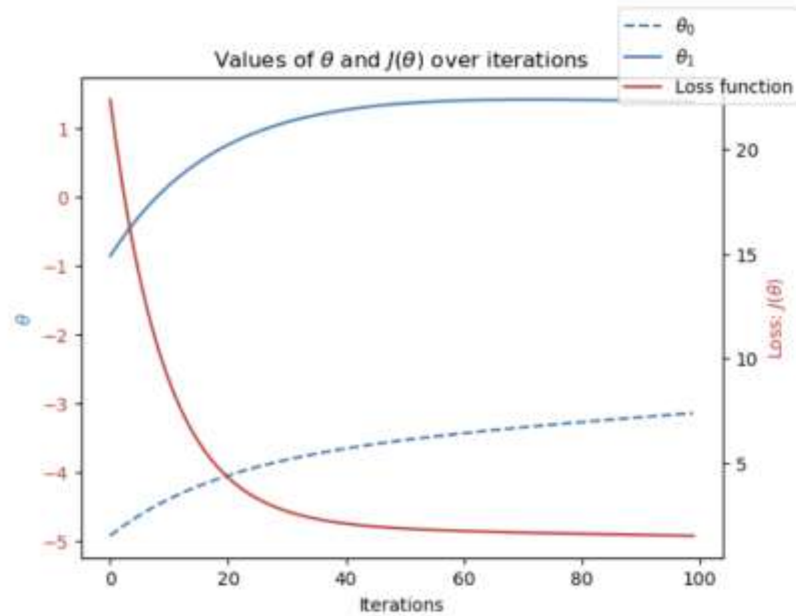
Basel 2017:

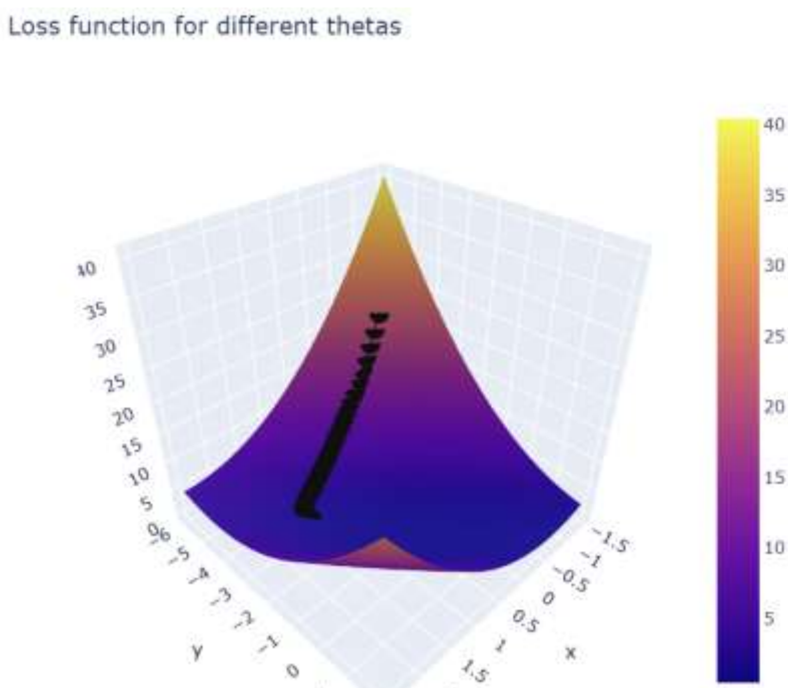


Loss function for different thetas

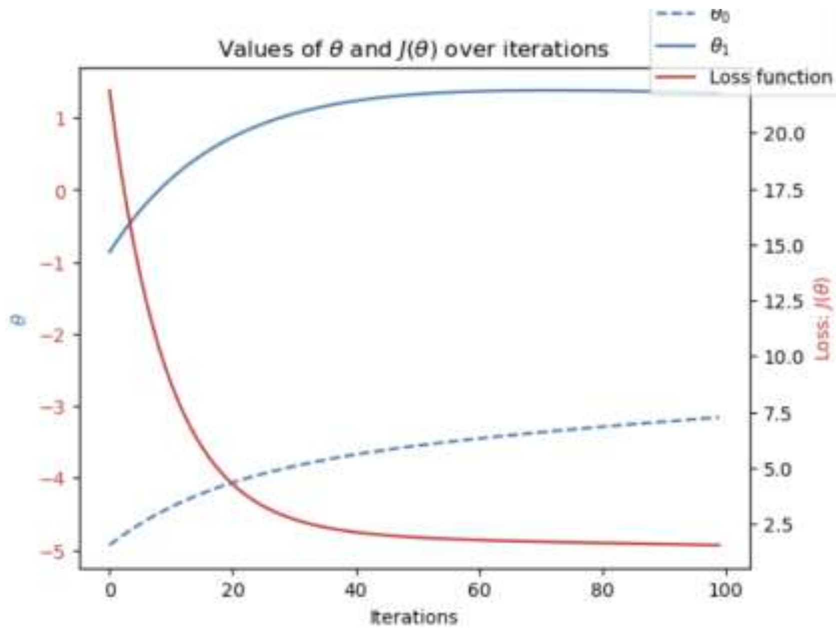


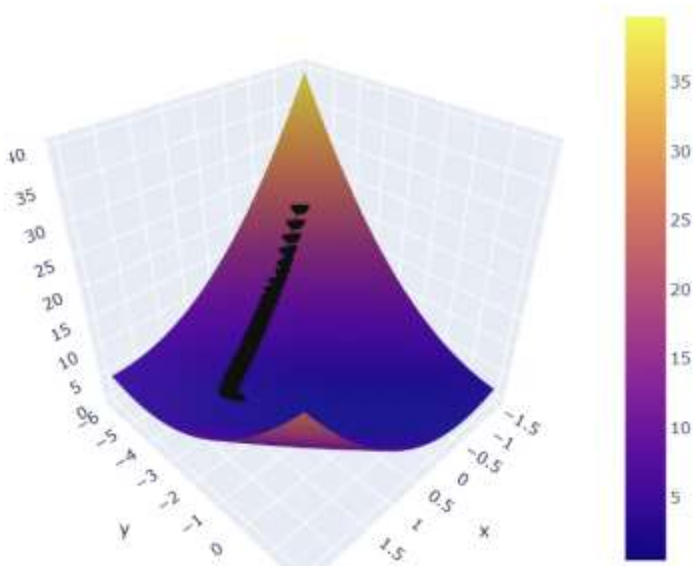
Gdansk 1961:


Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations
1 0 −1 −2 −3 −4 −5
20 15 10 5
0 20 40 60 80 100

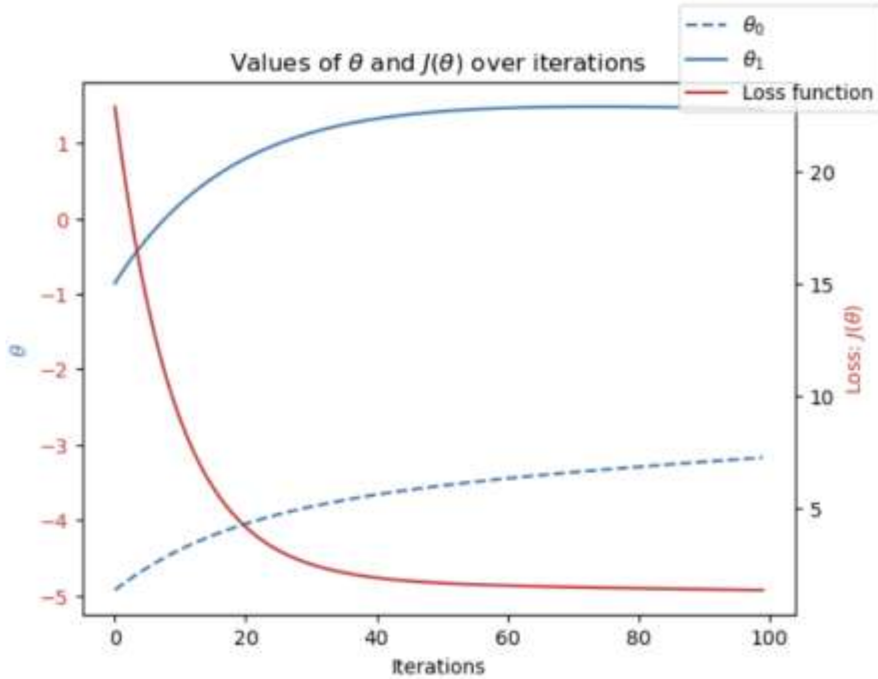

Loss function for different thetas
40 35 30 25 20 15 10 5
y
x

Gdansk 1993:

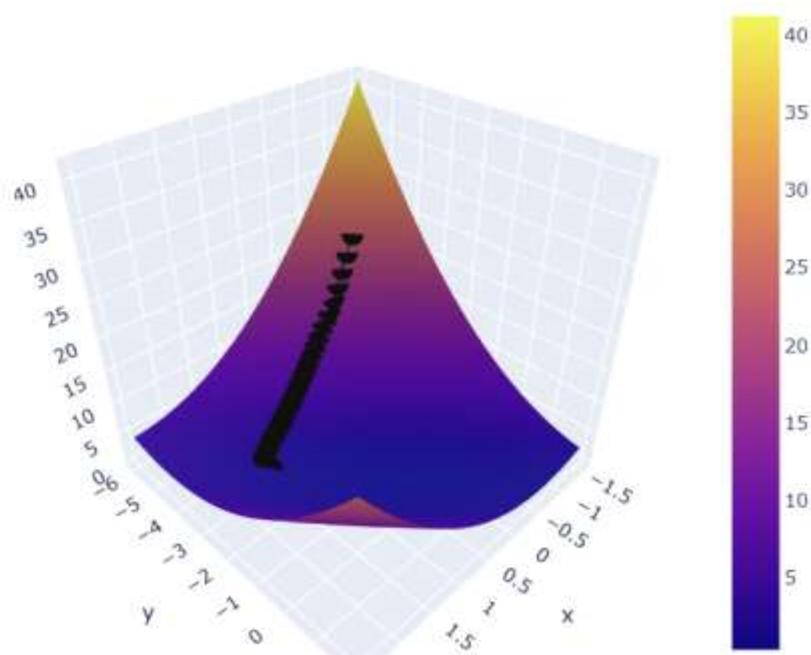

Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations


x
y

Gdansk 2017:

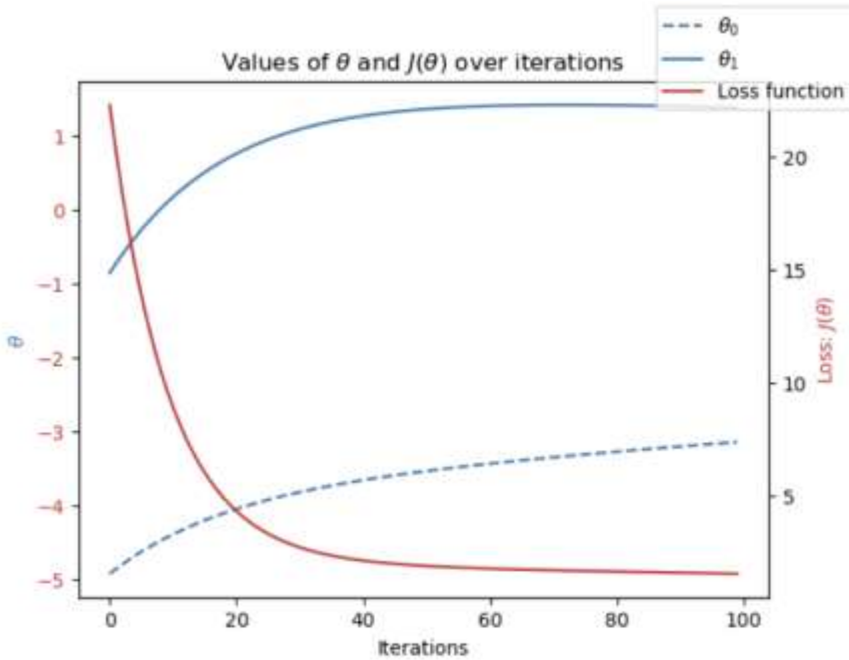


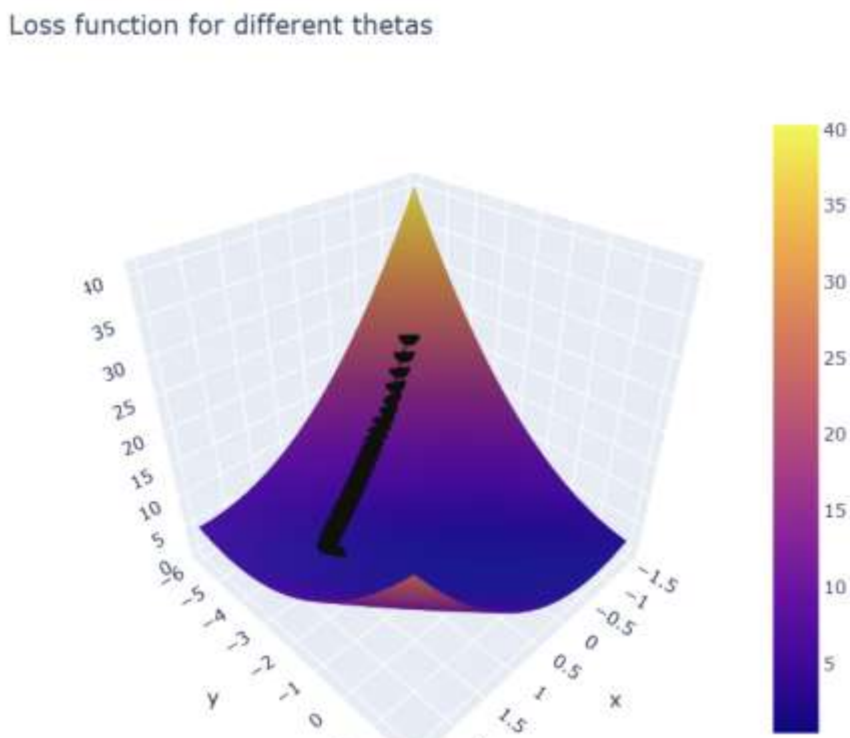
Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations



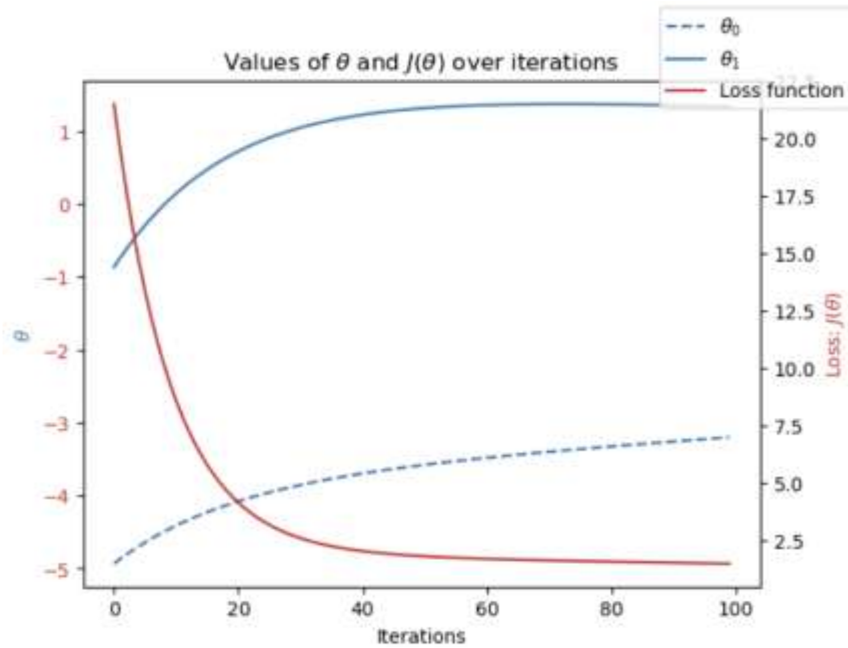
y
x

Madrid 1961:

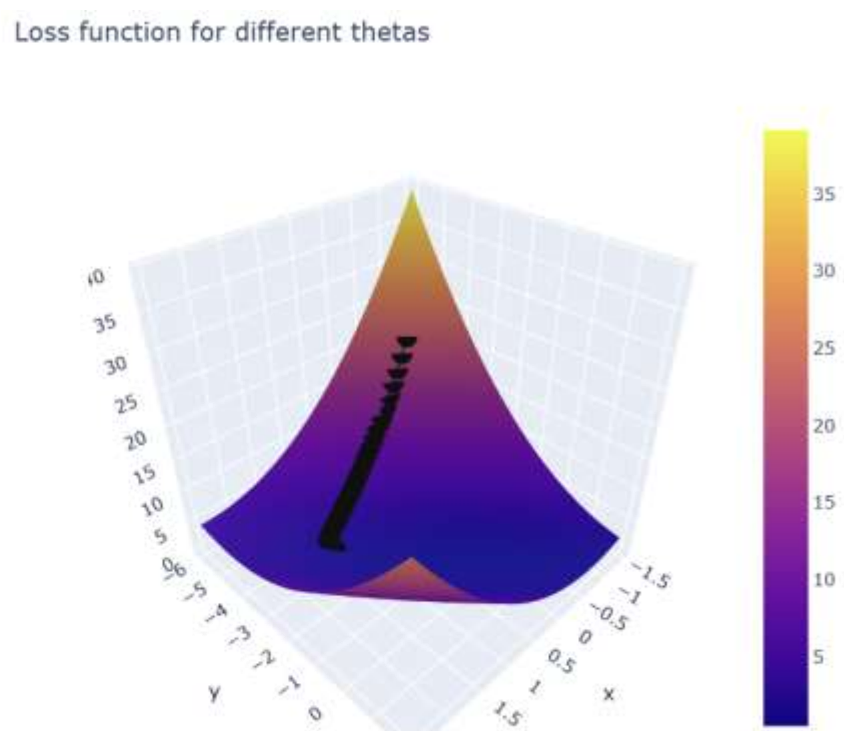

Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations


Loss function for different thetas
y
x

Madrid 1993:

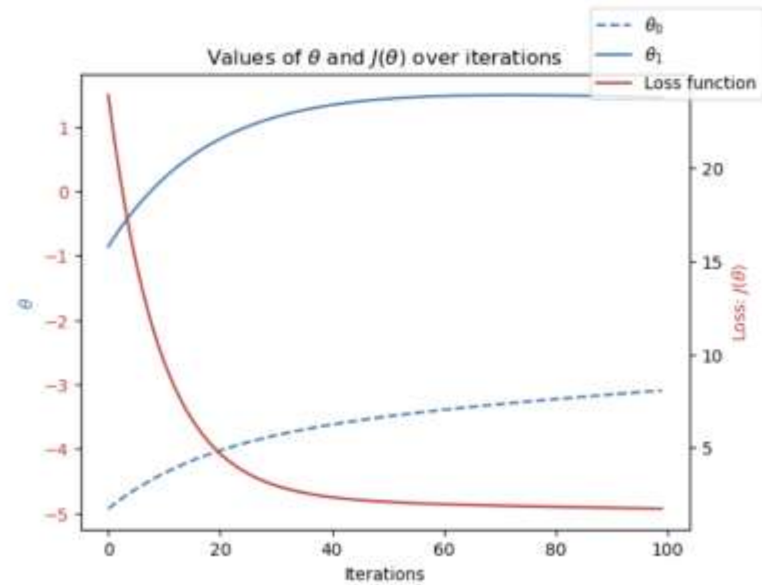


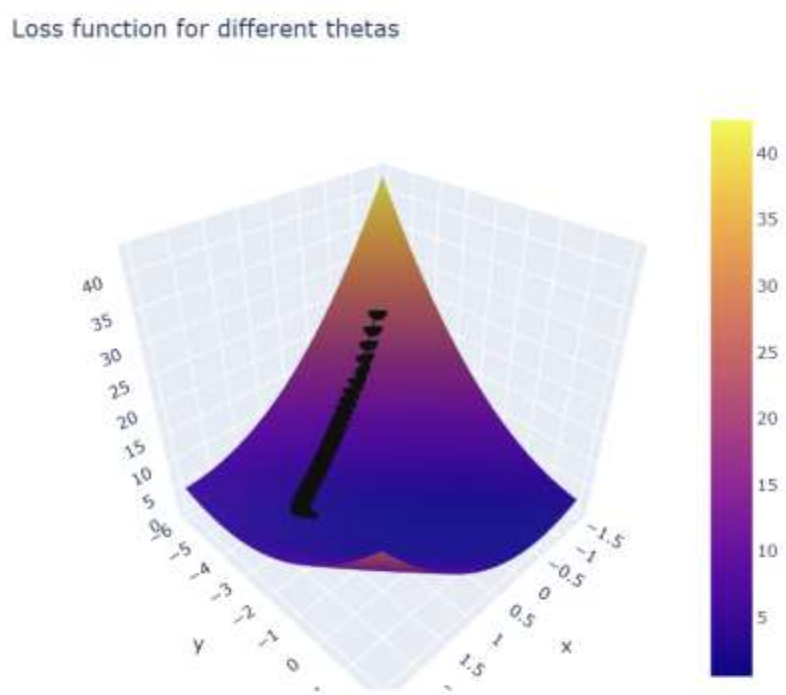
Values of θ and $J(\theta)$ over iterations
θ_0
θ_1
Loss function
θ
Loss: $J(\theta)$
Iterations



Loss function for different thetas
x
y

Madrid 2017:





Conclusion

Across all stations and years, the gradient descent algorithm showed consistent and stable convergence using α = 0.01 and 100 iterations. The initial parameters ($\theta_0 = -5$, $\theta_1 = -1$)

proved robust across datasets, producing similar loss curves and smooth descent paths toward the minimum. No adjustment in learning rate or iteration count was required.

Notes / Interpretation

- The **loss function decreased smoothly** in every run, which indicates the learning rate ($\alpha = 0.01$) was appropriately tuned. A higher rate might have caused oscillations or divergence, while a lower rate would have slowed convergence.
- The **θ_0 and θ_1 values stabilized** early (around iteration 60–80), confirming that 100 iterations was sufficient for all datasets.
- The **3D loss surface plots** showed a clean descent path directly toward the minimum, with no irregular jumps or looping, meaning the optimization process behaved as expected.
- The fact that **identical parameters worked well across multiple stations and years** suggests the datasets share similar linear characteristics—temperature trends that can be modeled with similar slopes and intercepts.
- There were **no signs of overfitting or instability**, since the loss function consistently approached zero without spikes.
- Minor differences in the curvature of the loss surface between stations likely reflect **regional or seasonal variations in temperature patterns**, but they didn't require changes to the parameters.
- If more precision were needed, future refinements could include testing **different learning rates ($\alpha = 0.005$–0.02)** or extending to **200–500 iterations** to confirm full convergence.