David Baker

EENG 581: Power System Operation and Management

HW3: Load Forecast

Learning Objectives:

- Develop linear models to forecast the demand.
- Use the WLS technique to solve a linear demand forecast model.
- Use the correlation technique to solve a linear demand forecast model.

1. The historical data of the annual peak demand for an electric utility are given below. Model the peak demand using the linear regression technique (i.e., assume $d(t) = a + b \cdot t$) and find a forecast for demand at year 6. Clearly, report the values of parameters a and b.

Year	1	2	3	4	5
Peak Annual Demand (MW)	6,197	6,337	7,411	7,940	8,473

2. Suppose that we have a linear demand model similar to the one considered in the previous problem. We are now going to solve the problem using WLS state estimation. Here, our states (unknowns) are the coefficients of the linear model, i.e., coefficients a and b. Demand values for previous years are then considered as the measurements. This gives us a measure model of the form: $d(t) = a + b \cdot t + e(t)$. Use the WLS technique to find coefficients a and b, and then use the model to find the forecasted demand at year 6. Solve this model twice: first assuming that all previous year measurements have the same weight (accuracy), and a second time, assuming that the weight associated with year t is $w(t) = 0.7^{(\mathrm{T}-t)}$, where T is 5. Notice how the latter assigns a larger weight (importance) to more recent measurements.

3. For problem **1** assume a Markov model of the form $d(t) = a \times d(t-1) + e(t)$. Find coefficient a using the correlation method. Determine the demand forecast for years 6 and 7.

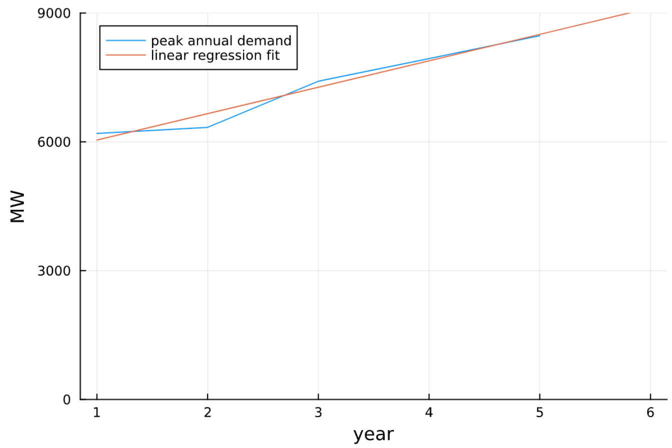
4. The peak load data for a utility company is given in the table below. Assume that only the load data for years 1981–1988 is available. Using this data, derive a time-series linear model for the peak demand based on its four past values, i.e. demand at year t is a linear function of demands of the previous four years. Use the model and its estimated load values to predict the demand for years 1989–1996. Compare with the actual values provided in the table below.

Year	Peak Load (GW)
1981	3,179
1982	3,694
1983	3,981
1984	4,672
1985	5,158
1986	5,361
1987	5,803
1988	6,152
1989	6,279
1990	6,664
1991	7,004
1992	7,215
1993	7,503
1994	7,657
1995	8,149
1996	8,491

1. The historical data of the annual peak demand for an electric utility are given below. Model the peak demand using the linear regression technique (i.e., assume $d(t) = a + b \cdot t$) and find a forecast for demand at year 6. Clearly, report the values of parameters a and b.

Year	1	2	3	4	5
Peak Annual Demand (MW)	6,197	6,337	7,411	7,940	8,473

```
forecasted peak demand in year 6: 9118 MW
```



	Coef.	
(Intercept)	5425.1	= a
year	615.5	= b

in $d(t)=a+bt$

```julia
using DataFrames, GLM, StatsModels
using Statistics
using Plots

demand_column_name = "peak annual demand MW"
demands = [6197, 6337, 7411, 7940, 8473]
df = DataFrame(year=1:5, d=demands)
years = df[!, "year"]

plot(years, df[!, "d"],
    label="peak annual demand",
    ylabel="MW",
    xlabel="year",
    ylims=[0, 9000]
)

ols = lm(@formula(d ~ year), df)

d_predicted = predict(ols, DataFrame(year=1:6))
plot!(1:6, d_predicted, label="linear regression fit")

println(coeftable(ols))
println("forecasted peak demand in year 6: $(round(Int,d_predicted[6])) MW")

# OR, if we were meant to do this more by hand...
x = years
y = demands
x_mean = mean(x)
y_mean = mean(y)
N = length(years)

slope = sum((x[i] - x_mean) * (y[i] - y_mean) for i ∈ 1:N) / sum((x[i] - x_mean)^2 for i ∈ 1:N)
intercept = y_mean - slope*x_mean
```

2. Suppose that we have a linear demand model similar to the one considered in the previous problem. We are now going to solve the problem using WLS state estimation. Here, our states (unknowns) are the coefficients of the linear model, i.e., coefficients a and b. Demand values for previous years are then considered as the measurements. This gives us a measure model of the form: $d(t) = a + b \cdot t + e(t)$. Use the WLS technique to find coefficients a and b, and then use the model to find the forecasted demand at year 6. Solve this model twice: first assuming that all previous year measurements have the same weight (accuracy), and a second time, assuming that the weight associated with year t is $w(t) = 0.7^{(T-t)}$, where T is 5. Notice how the latter assigns a larger weight (importance) to more recent measurements.

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow \text{states}$$

$$z = \begin{bmatrix} \text{demand}(1) \\ \vdots \\ \text{demand}(5) \end{bmatrix} \longrightarrow \text{measurements}$$

Two methods of doing this, both w/ same answer

1. optimization problem

$$\min \sum_i w_i \, r_i^2$$

$$\text{st} \quad r_i = z_i - h_i(x) \qquad \text{where} \quad h(x) = a + bt + e(t)$$

```julia
using JuMP
import GAMS
using DataFrames

function solve_wls(weights)
    model = Model(GAMS.Optimizer)
    set_optimizer_attribute(model, "solver", "CPLEX")
    # ==========================================================
    # SETS
    years = 1:5
    # ==========================================================
    # PARAMETERS
    measurements = [6197, 6337, 7411, 7940, 8473]
    # ==========================================================
    # DECISION VARIABLES
    @variable(model, R[years]) # residual
    @variable(model, A) # intercept
    @variable(model, B) # slope
    # ==========================================================
    # OBJECTIVE
    @objective(model, Min, sum(weights[i] * R[i]^2 for i ∈ years))
    # ==========================================================
    # CONSTRAINTS
    for i ∈ years
        @constraint(model, R[i] == measurements[i] - (A + B * years[i]))
    end

    optimize!(model)

    v = all_variables(model)
    df = DataFrame(
        variable=name.(v),
        val=value.(v),
    )
    show(df, eltypes=false)

    demand_year_6 = value(A) + value(B) * 6
    println("\n\nIntercept (A): $(round(value(A), digits=1)), Slope (B): $(round(value(B),
digits=1))")
    println("forecasted peak demand in year 6: $(round(demand_year_6, digits=1)) MW\n\n")
    return demand_year_6
end

weights = ones(length(years))
equal_weights_wls = solve_wls(weights)

T = 5
weight(t) = 0.7^(T - t)
weights = weight.(years)
weighted_wls = solve_wls(weights)
```

Equal Weights (same answer as problem 1)

```
Intercept (A): 5425.1, Slope (B): 615.5
forecasted peak demand in year 6: 9118.1 MW
```

Weighted

```
Intercept (A): 5419.0, Slope (B): 617.5
forecasted peak demand in year 6: 9124.2 MW
```

2. Using closed form solution

$$X = (H^T W H)^{-1} \cdot H^T W z$$

where W = diagonal matrix with W_i^2 on diagonal

note that $\frac{\partial e(t)}{\partial a} = \frac{\partial e(t)}{\partial b} = 0$

$h = a + bt + e(t)$

$x = \begin{bmatrix} a \\ b \end{bmatrix}$

$$H = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial(a + b \cdot 1)}{\partial a} & \frac{\partial(a + b \cdot 1)}{\partial b} \\ \frac{\partial(a + b \cdot 2)}{\partial a} & \frac{\partial(a + b \cdot 2)}{\partial b} \\ \vdots & \vdots \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix}$$

```julia
using LinearAlgebra

years = 1:5
measurements = [6197, 6337, 7411, 7940, 8473]

## CLOSED FORM SOLUTION
# equal weights scenario
H = hcat(ones(5), years)
z = measurements
W = diagm(ones(5)) # weights

X = (H' * W * H)^-1 * H' * W * z
a = X[1]
b = X[2]
println("\n\nIntercept (A): $(round(a, digits=1)), Slope (B): $(round(b, digits=1))")
println("forecasted peak demand in year 6: $(round(a + b*6, digits=1)) MW\n\n")

# weighted scenario
T = 5
weight(t) = 0.7^(T - t)
weights = weight.(years)
W = diagm(weights) # weights

X = (H' * W * H)^-1 * H' * W * z
a = X[1]
b = X[2]
println("\n\nIntercept (A): $(round(a, digits=1)), Slope (B): $(round(b, digits=1))")
println("forecasted peak demand in year 6: $(round(a + b*6, digits=1)) MW")
```

Equal Weights

```
Intercept (A): 5425.1, Slope (B): 615.5
forecasted peak demand in year 6: 9118.1 MW
```

Weighted

```
Intercept (A): 5419.0, Slope (B): 617.5
forecasted peak demand in year 6: 9124.2 MW
```

3. For problem **1** assume a Markov model of the form $d(t) = a \times d(t-1) + e(t)$. Find coefficient a using the correlation method. Determine the demand forecast for years 6 and 7.

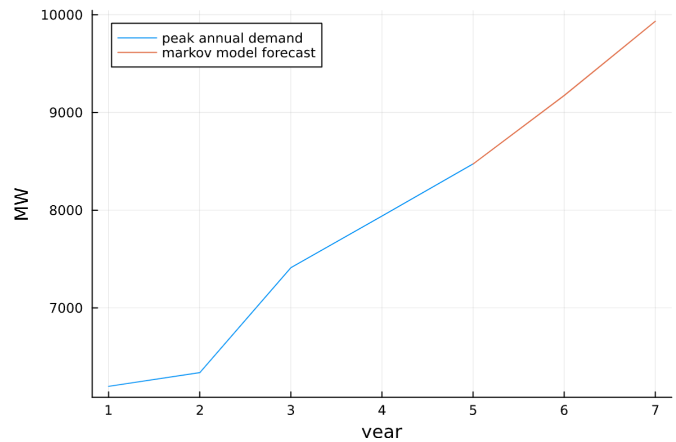
$$d(t) = a\,d(t-1) + e(t)$$

multiply both sides by $d(t)$ and take expectation

$$E[d(t)\cdot d(t)] = a\,E[d(t)\cdot d(t-1)] + \underbrace{E[d(t)\,\cancel{e(t)}]}_{0}$$

$$a = \frac{E[d(t)\cdot d(t)]}{E[d(t)\cdot d(t-1)]}$$

```
a = 1.083
forecasted demand in year 6 = 9173.8 MW
forecasted demand in year 7 = 9932.5 MW
```



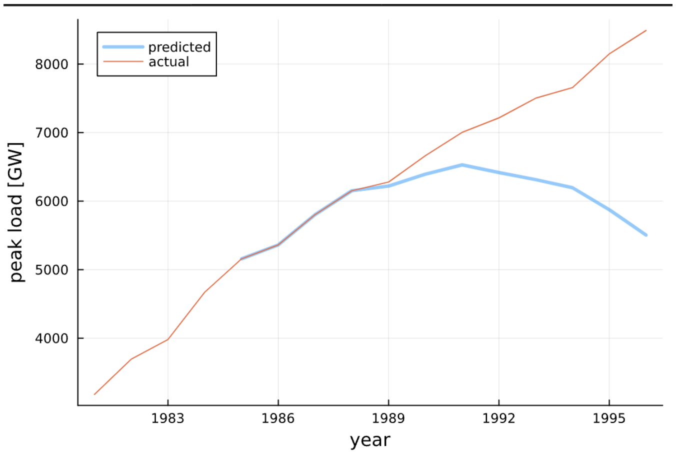

```julia
using Statistics
using Plots

years = 1:5
d = [6197.0, 6337.0, 7411.0, 7940.0, 8473.0]

numerator = mean(d[2:end] .* d[2:end]) # not sure if this should be 1:end or 2:end
denominator = mean(d[2:end] .* d[1:end-1])

a = numerator / denominator
println("a = $(round(a, digits=3))")

d_pred = [d[end]]
for i in 1:2
    d_t = a * d_pred[end]
    push!(d_pred, d_t)
end

d_pred_first = []
for i in 2:5
    push!(d_pred_first, a * d[i-1])
end

println("forecasted demand in year 6 = $(round(d_pred[2], digits=1)) MW")
println("forecasted demand in year 7 = $(round(d_pred[3], digits=1)) MW")

plt = plot(years, d,
    label="peak annual demand",
    ylabel="MW",
    xlabel="year",
)
plot!(5:7, d_pred, label="markov model forecast")
display(plt)
```

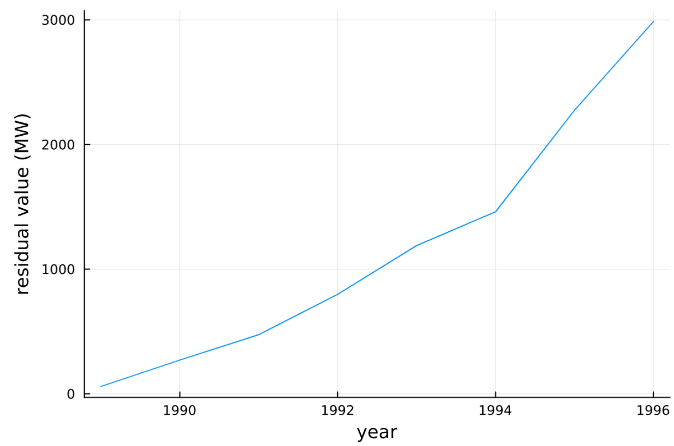
4. The peak load data for a utility company is given in the table below. Assume that only the load data for years 1981–1988 is available. Using this data, derive a time-series linear model for the peak demand based on its four past values, i.e. demand at year *t* is a linear function of demands of the previous four years. Use the model and its estimated load values to predict the demand for years 1989–1996. Compare with the actual values provided in the table below.

Year	Peak Load (GW)
1981	3,179
1982	3,694
1983	3,981
1984	4,672
1985	5,158
1986	5,361
1987	5,803
1988	6,152
1989	6,279
1990	6,664
1991	7,004
1992	7,215
1993	7,503
1994	7,657
1995	8,149
1996	8,491

available

$$d(t) = \sum_{k=1}^{4} a_k \, d(t-k) + e(t)$$





	Coef.
(Intercept)	0.0
d_t_1	1.29289
d_t_2	-0.362
d_t_3	0.853222
d_t_4	-0.815682

Model looks like

$$d(t) = 1.29 \cdot d(t-1) - 0.36 \cdot d(t-2) + 0.85\, d(t-3) - 0.82\, d(t-4)$$

Clearly, the model is not a good fit. The residual plot shows a clear trend of increasing over time.

years	actual MW	predicted MW	residual MW
1981	3179.0	----	----
1982	3694.0	----	----
1983	3981.0	----	----
1984	4672.0	----	----
1985	5158.0	5158.0	6.91216e-11
1986	5361.0	5361.0	5.00222e-11
1987	5803.0	5803.0	2.54659e-11
1988	6152.0	6152.0	6.54836e-11
1989	6279.0	6220.0	59.0036
1990	6664.0	6393.11	270.889
1991	7004.0	6529.56	474.443
1992	7215.0	6416.64	798.359
1993	7503.0	6313.5	1189.5
1994	7657.0	6196.24	1460.76
1995	8149.0	5874.34	2274.66
1996	8491.0	5504.7	2986.3

CODE on following page

```julia
using DataFrames, GLM, StatsModels
using Plots
using PrettyTables

years = 1981:1996
demands = Float64.([3179, 3694, 3981, 4672, 5158, 5361, 5803, 6152, 6279, 6664, 7004, 7215,
7503, 7657, 8149, 8491])

train_years = 1981:1988
train_d = demands[1:length(train_years)]

# have to start at 5 so you have t - 4 data available
df = DataFrame(
    y=train_years[5:end],
    d_t=train_d[5:end],
    d_t_1=train_d[4:end-1],
    d_t_2=train_d[3:end-2],
    d_t_3=train_d[2:end-3],
    d_t_4=train_d[1:end-4],
)

model = lm(@formula(d_t ~ d_t_1 + d_t_2 + d_t_3 + d_t_4), df)

predict(model, df)

pred_demands = train_d[1:4]
for (i, year) in enumerate(1985:1996)
    intercept, a, b, c, d = coef(model)
    d_next = (
        intercept
        + a * pred_demands[end]
        + b * pred_demands[end-1]
        + c * pred_demands[end-2]
        + d * pred_demands[end-3]
    )
    push!(pred_demands, d_next)
end


println("Model is")
println(model)

plt = plot(years, pred_demands, label="predicted")
plot!(years, demands, label="actual", ylabel="peak load [GW]", xlabel="year")
display(plt)

df_answer = DataFrame(
    years=years,
    actual=demands,
    predicted=[missing, missing, missing, missing, pred_demands[5:end]...],
    residuals=[missing, missing, missing, missing, (demands - pred_demands)[5:end]...]
)
plot(1989:1996, df_answer[9:16, "residuals"], xlabel="year", ylabel="residual value (MW)",
legend=:none)
df_final = coalesce.(df_answer, "---")

header = (["years", "actual", "predicted", "residual"], ["", "MW", "MW", "MW"])
pretty_table(df_final, header=header, header_crayon=crayon"yellow bold")
```