Exercise 2

```
using CSV, DataFrames, Statistics, GLM, StatsPlots, StatsModels, StatsBase
```

```
file_path = "/Users/michelletorres/Desktop/Homeworks AI/archive/bottle.csv"
data = CSV.read(file_path, DataFrame)
```

864863×74 DataFrame *864838 rows omitted*

Row	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat
	Int64	Int64	String15	String	Int64	Float64?	Float64?	Float64?	Float64?	Float64?
1	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.5	33.44	*missing*	25.649	*missing*
2	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.46	33.44	*missing*	25.656	*missing*
3	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.46	33.437	*missing*	25.654	*missing*
4	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.45	33.42	*missing*	25.643	*missing*
5	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.45	33.421	*missing*	25.643	*missing*
				19-						

6	1	6	054.0 056.0	4903CR-HY-060-0930-05400560-0030A-7	30	10.45	33.431	*missing*	25.651	*missing*
7	1	7	054.0 056.0	19-4903CR-HY-060-0930-05400560-0039A-3	39	10.45	33.44	*missing*	25.658	*missing*
8	1	8	054.0 056.0	19-4903CR-HY-060-0930-05400560-0050A-7	50	10.24	33.424	*missing*	25.682	*missing*
9	1	9	054.0 056.0	19-4903CR-HY-060-0930-05400560-0058A-3	58	10.06	33.42	*missing*	25.71	*missing*
10	1	10	054.0 056.0	19-4903CR-HY-060-0930-05400560-0075A-7	75	9.86	33.494	*missing*	25.801	*missing*
11	1	11	054.0 056.0	19-4903CR-HY-060-0930-05400560-0078A-3	78	9.83	33.51	*missing*	25.819	*missing*
12	1	12	054.0 056.0	19-4903CR-HY-060-0930-05400560-0100A-7	100	9.67	33.58	*missing*	25.9	*missing*
13	1	13	054.0 056.0	19-4903CR-HY-060-0930-05400560-0117A-3	117	9.5	33.64	*missing*	25.975	*missing*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
				20-						

864852	34403	864852	093.3 120.0	20-1611SR-MX-313-2053-09331200-0300A-7	300	7.831	34.0234	2.218	26.5407	33.2
864853	34403	864853	093.3 120.0	20-1611SR-MX-313-2053-09331200-0321A-3	321	7.538	34.042	1.984	26.5979	29.5
864854	34403	864854	093.3 120.0	20-1611SR-MX-313-2053-09331200-0381A-3	381	6.943	34.1104	1.108	26.7357	16.26
864855	34403	864855	093.3 120.0	20-1611SR-MX-313-2053-09331200-0400A-7	400	6.694	34.1101	1.096	26.7693	15.99
864856	34403	864856	093.3 120.0	20-1611SR-MX-313-2053-09331200-0440A-3	440	6.312	34.1563	0.718	26.8564	10.38
864857	34403	864857	093.3 120.0	20-1611SR-MX-313-2053-09331200-0500A-7	500	5.993	34.216	0.456	26.9452	6.55
864858	34403	864858	093.3 120.0	20-1611SR-MX-313-2053-09331200-0521A-3	521	5.818	34.2382	0.366	26.9848	5.23
864859	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.8706	108.74
				20-						

864860	34404	864860	093.4 026.4	1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805	23.8707	108.7
864861	34404	864861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.415	5.796	23.8891	108.4
864862	34404	864862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816	24.0143	107.7
864863	34404	864863	093.4 026.4	20-1611SR-MX-310-2239-09340264-0015A-3	15	17.533	33.388	5.774	24.153	105.6

```
names(data) .= strip.(names(data)) # Remove extra spaces from column names
println("Columnas disponibles:")
for col in names(data)
    println("`$col`")
end
```

```
Columnas disponibles:
`Cst_Cnt`
`Btl_Cnt`
`Sta_ID`
`Depth_ID`
`Depthm`
`T_degC`
`Salnty`
`O2ml_L`
`STheta`
`O2Sat`
`Oxy_µmol/Kg`
`BtlNum`
`RecInd`
`T_prec`
`T_qual`
`S_prec`
```

```
    `S_qual`
    `P_qual`
    `O_qual`
    `SThtaq`
    `O2Satq`
    `ChlorA`
    `Chlqua`
    `Phaeop`
    `Phaqua`
    `PO4uM`
    `PO4q`
    `SiO3uM`
    `SiO3qu`
    `NO2uM`
    `NO2q`
    `NO3uM`
    `NO3q`
    `NH3uM`
    `NH3q`
    `C14As1`
    `C14A1p`
    `C14A1q`
    `C14As2`
    `C14A2p`
    `C14A2q`
    `DarkAs`
    `DarkAp`
    `DarkAq`
    `MeanAs`
    `MeanAp`
    `MeanAq`
    `IncTim`
    `LightP`
    `R_Depth`
    `R_TEMP`
    `R_POTEMP`
    `R_SALINITY`
    `R_SIGMA`
    `R_SVA`
    `R_DYNHT`
    `R_O2`
    `R_O2Sat`
```

```
columns_of_interest = [:T_degC, :Salnty, :Depthm, :O2ml_L] # Necessary columns as
missing_columns = setdiff(columns_of_interest, Symbol.(names(data))) # Check if r
if !isempty(missing_columns)
    println("Faltan las siguientes columnas en DataFrame: $missing_columns")
    error("Faltan columnas necesarias")
end
```

```
filtered_data = data[:, columns_of_interest] # Filter columns
filtered_data = dropmissing(filtered_data) # Ensure there are no missing values i
println("Datos después de filtrado:") # Verify that the columns have been loaded
```

```
Datos después de filtrado:
```

```
data_model = @formula(T_degC ~ Salnty + Depthm + O2ml_L) # Linear regression with
lm_model = lm(data_model, filtered_data)
println("Resumen del modelo:")
println(coef(lm_model))
println(summary(lm_model))
```

```
Resumen del modelo:
[-168.2751028141903, 5.115126418319546, -0.005011889278487319, 2.117981438058
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM
```

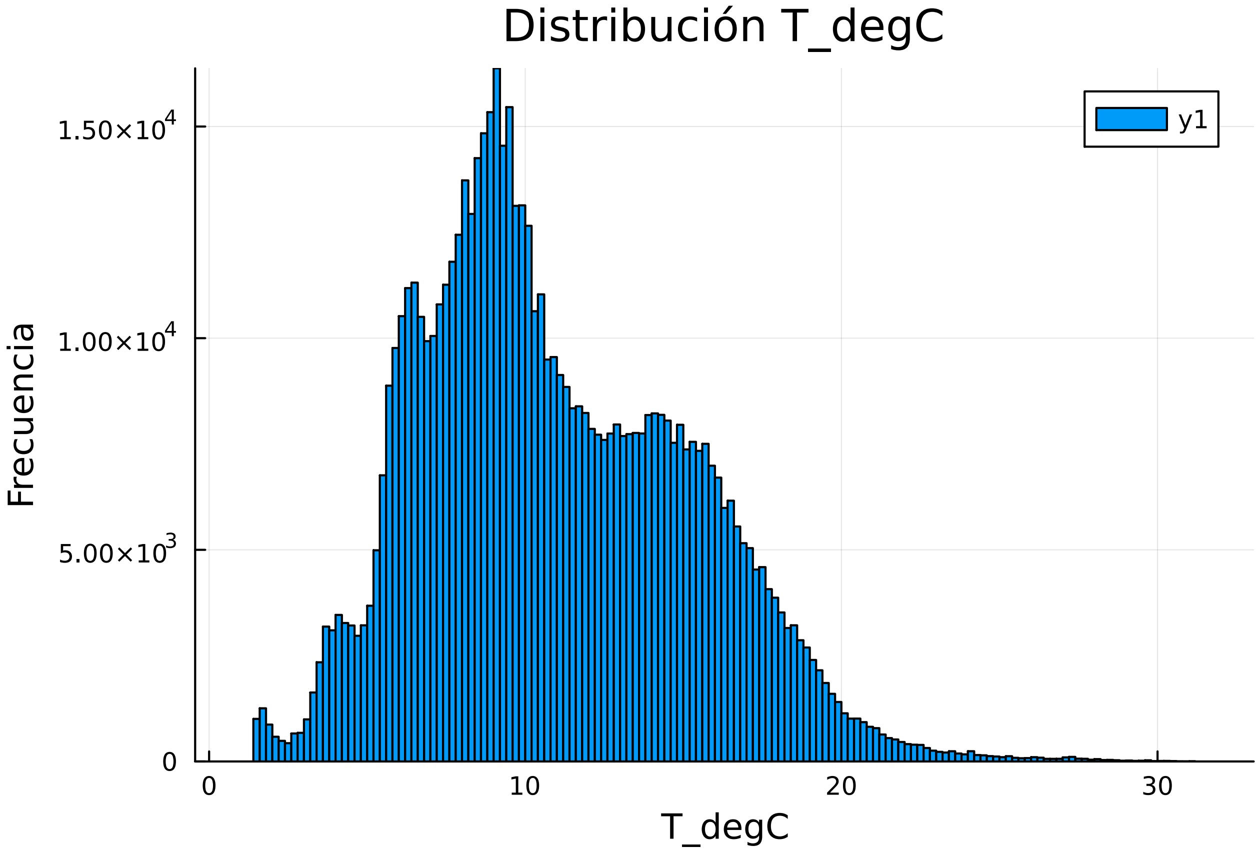
```
function calculate_rmse(model, data)
    predictions = StatsBase.predict(model, data)  # Utiliza StatsBase.predict
    residuals = data[:, :T_degC] .- predictions
    return sqrt(mean(residuals .^ 2))
end
```

```
calculate_rmse (generic function with 1 method)
```

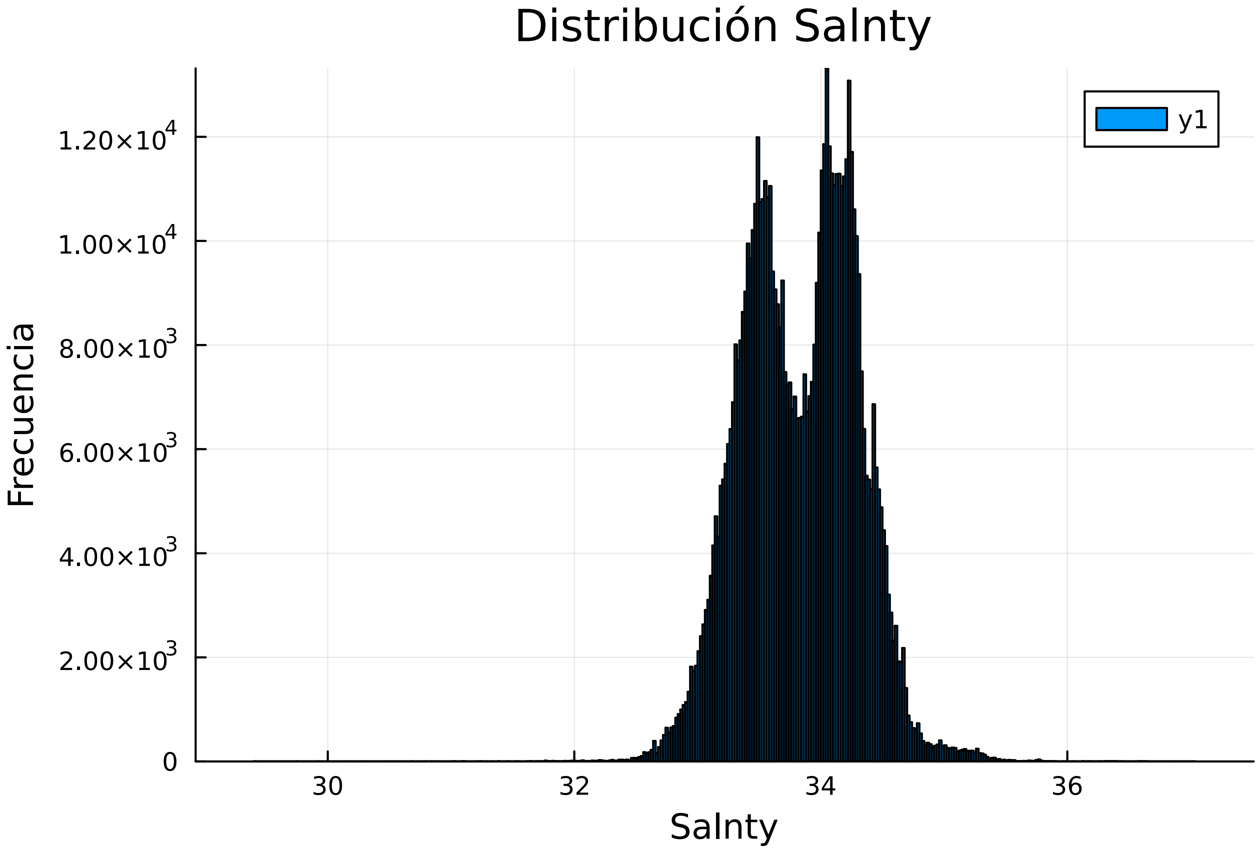
```
rmse = calculate_rmse(lm_model, filtered_data)
println("RMSE del modelo: $rmse")
```

```
RMSE del modelo: 1.9436411276934393
```

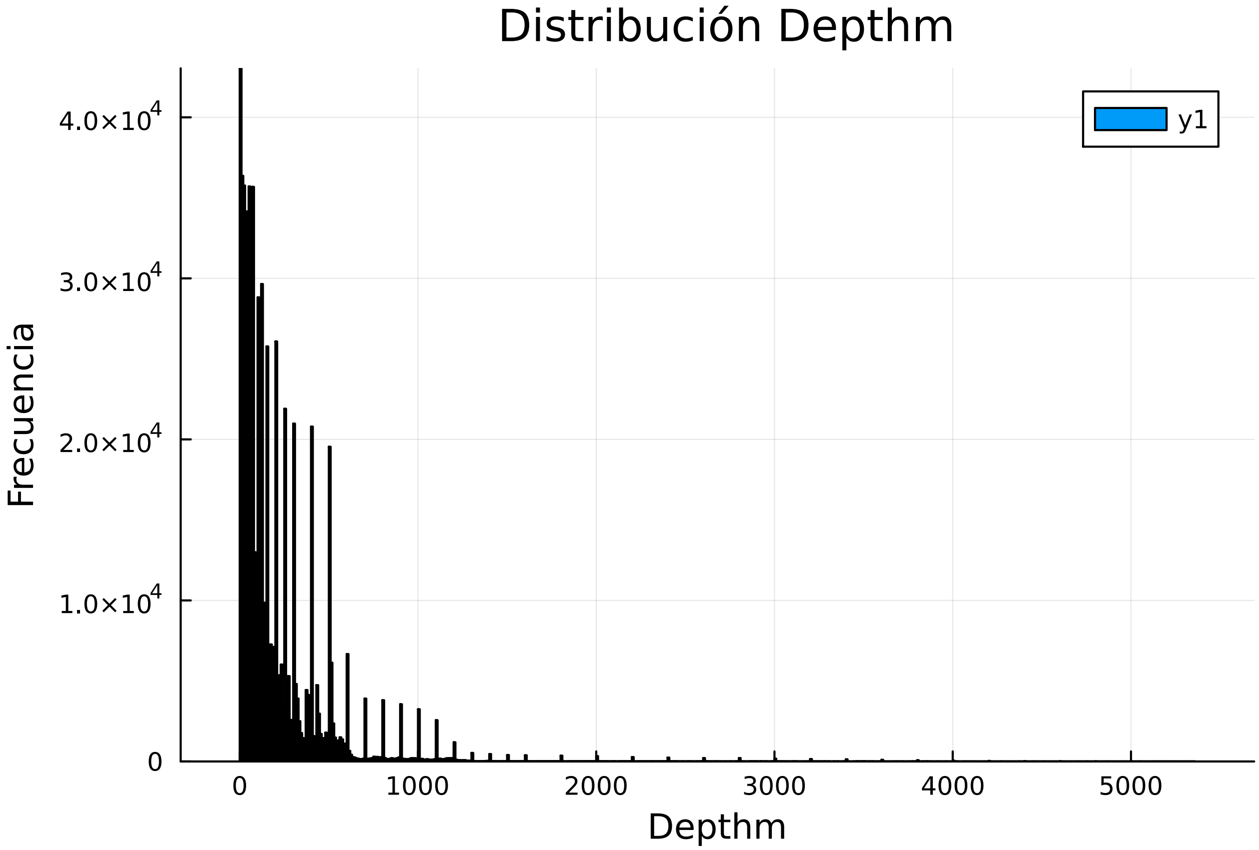
```
histogram(filtered_data[!, :T_degC], title="Distribución T_degC", xlabel="T_degC"
```



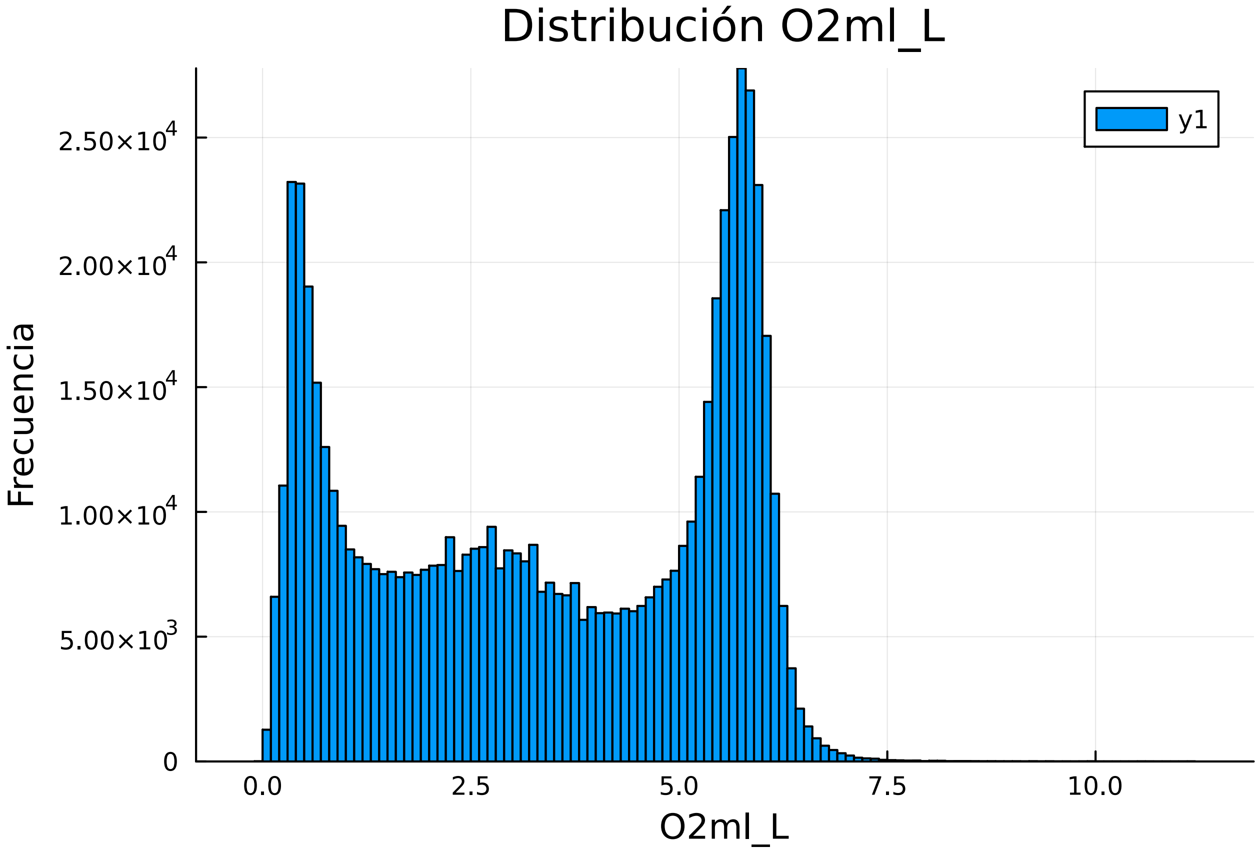
```
histogram(filtered_data[!, :Salnty], title="Distribución Salnty", xlabel="Salnty"
```



```
histogram(filtered_data[!, :Depthm], title=" Distribución Depthm", xlabel="Depthm
```



```
histogram(filtered_data[!, :O2ml_L], title="Distribución O2ml_L", xlabel="O2ml_L"
```



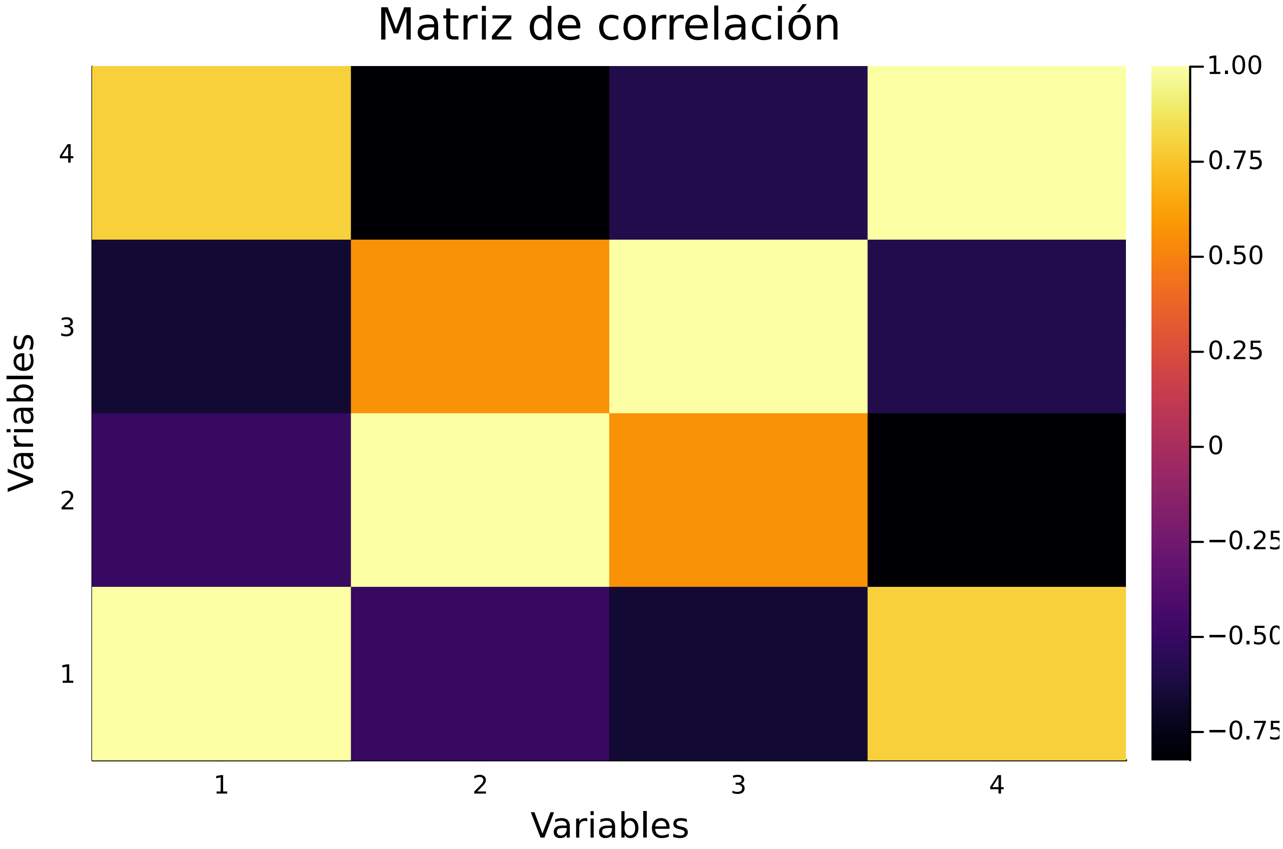
```
combinations = [ # List of independent v
    [:Salnty, :Depthm, :O2ml_L],
    [:Salnty, :Depthm],
    [:Salnty, :O2ml_L],
    [:Depthm, :O2ml_L],
    [:Salnty],
    [:Depthm],
    [:O2ml_L]
]

best_rmse = Inf
best_model = nothing
best_combination = nothing
##################################
names(filtered_data)
println(names(filtered_data))  # Verify
#################################
for combination in combinations
    formula = @eval @formula(T_degC ~ $(
    lm_model = lm(formula, filtered_data
    rmse = calculate_rmse(lm_model, filt
    println("RMSE para combinación $comb
    if rmse < best_rmse     # If RMSE is
        best_rmse = rmse
        best_model = lm_model
        best_combination = combination
    end
end
```

"@eval" no es una anotación válida. Se permiten los siguientes valores: [@param, @title, @markdown].

```
["T_degC", "Salnty", "Depthm", "O2ml_L"]
RMSE para combinación [:Salnty, :Depthm, :O2ml_L]: 1.9436411276934393
RMSE para combinación [:Salnty, :Depthm]: 3.0811490028449944
RMSE para combinación [:Salnty, :O2ml_L]: 2.3063058428905685
RMSE para combinación [:Depthm, :O2ml_L]: 2.345609992014096
RMSE para combinación [:Salnty]: 3.64457684656509
RMSE para combinación [:Depthm]: 3.152476963183015
RMSE para combinación [:O2ml_L]: 2.5625749572805048
```

```
println("Mejor variable para combianción: $best_combination with RMSE: $best_rmse
correlation_matrix = cor(Matrix(filtered_data[:, [:T_degC, :Salnty, :Depthm, :O2m
heatmap(correlation_matrix, xlabel="Variables", ylabel="Variables", title="Matriz
```

```
Mejor variable para combianción: [:Salnty, :Depthm, :O2ml_L] with RMSE: 1.9436
```


Matriz de correlación
Variables
Variables