Simple operations

```
A =   Int64[12,  3]
```

```
A=[12,3]
```

```
print(A)
```

```
y = 3×1 LinearAlgebra.Adjoint{Int64,Array{Int64,2}}:
     3
     4
     5
```

```
y = [3 4 5]'
```

```
X = 3×2 LinearAlgebra.Adjoint{Int64,Array{Int64,2}}:
     1  1
     1  6
     1  7
```

```
X = [1 1 1; 1 6 7]'
```

```
2×1 Array{Float64,2}:
 2.64516129032258
 0.29032258064516103
```

```
(X'X)^(-1)*X'y
```

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

```
f(x) = 2x^3 + 3x^2 + 6
```

```
34
```

```
f(2)
```

Packages

```
begin
    import Pkg
    Pkg.add("RDatasets")
    Pkg.add("Econometrics")
    Pkg.add("GLM")
    Pkg.add("CSV")
    Pkg.add("Gadfly")
    Pkg.add("LinearAlgebra")
    Pkg.add("Plots")
    Pkg.add("PyPlot")
end
```

```
begin
    Pkg.add("DataFrames")
end
```

```
begin
    Pkg.add("CSV")
end
```

Multiple definitions for CSV.

Combine all definitions into a single reactive cell using a `begin ... end` block.

```
using CSV
```

Multiple definitions for CSV.

Combine all definitions into a single reactive cell using a `begin ... end` block.

```
using DataFrames, GLM, Statistics, CSV, Econometrics
```

```
using RDatasets
```

```
let
    Pkg.activate("Plots")
    Pkg.add("Plots")
    using Plots
end
```

First seminar

psid =

	IntNum	PersNum	Age	Educatn	Earnings	Hours	Kids	Marrie
1	4	4	39	12	77250	2940	2	CategoricalValue{Stri
2	4	6	35	12	12000	2040	2	CategoricalValue{Stri
3	4	7	33	12	8000	693	1	CategoricalValue{Stri
4	4	173	39	10	15000	1904	2	CategoricalValue{Stri
5	5	2	47	9	6500	1683	5	CategoricalValue{Stri
6	6	4	44	12	6500	2024	2	CategoricalValue{Stri
7	6	172	38	16	7000	1144	3	CategoricalValue{Stri
8	7	4	38	9	5000	2080	4	CategoricalValue{Stri
9	7	170	39	12	21000	2575	3	CategoricalValue{Stri
10	7	171	37	11	0	0	5	CategoricalValue{Stri

more

```
psid = dataset("Ecdat", "PSID")
```

Int32[39, 35, 33, 39, 47, 44, 38, 38, 39, 37, 48, 47, 40, 38, 41, 43, 41, 36,

```
psid.Age
```

	variable	mean	min	median	
1	:IntNum	4598.1	4	5464.0	9306
2	:PersNum	59.2136	1	4.0	205
3	:Age	38.4629	30	38.0	50
4	:Educatn	16.3771	0	12.0	99
5	:Earnings	14244.5	0	11000.0	240000
6	:Hours	1235.33	0	1517.0	5160
7	:Kids	4.48126	0	2.0	99
8	:Married	nothing	CategoricalValue{String,UInt8} "marrie	nothing	CategoricalValue{

```
describe(psid)
```

	IntNum	PersNum	Age	Educatn	Earnings	Hours	Kids	Married
1	4	4	39	12	77250	2940	2	CategoricalValue{String,
2	4	6	35	12	12000	2040	2	CategoricalValue{String,
3	4	7	33	12	8000	693	1	CategoricalValue{String,
4	4	173	39	10	15000	1904	2	CategoricalValue{String,
5	5	2	47	9	6500	1683	5	CategoricalValue{String,
6	6	4	44	12	6500	2024	2	CategoricalValue{String,
7	6	172	38	16	7000	1144	3	CategoricalValue{String,

```
first(psid, 7)
```

Model first seminar

```
model_a =
Continuous Response Model
Number of observations: 4855
Null Loglikelihood: -13840.79
Loglikelihood: -13838.29
R-squared: 0.0010
LR Test: 5.01 ~ χ²(2) ⇒  Pr > χ² = 0.0818
Formula: log(1 + Earnings) ~ 1 + Age + Educatn
Variance Covariance Estimator: OIM
```

	PE	SE	t-value	Pr > |t|	2.50%	97.50%
(Intercept)	6.15726	0.421631	14.6034	<1e-46	5.33068	6.98385
Age	0.024032	0.0107385	2.23793	0.0253	0.00297964	0.0450844
Educatn	0.000165885	0.00325684	0.0509343	0.9594	-0.006219	0.00655077

```
model_a = fit(EconometricModel, @formula(log(1+Earnings) ~ Age + Educatn), psid)
```

Second seminar

diamonds =

	Carat	Cut	Color
1	0.23	CategoricalValue{String,UInt8} "Ideal"	CategoricalValue{String,UInt8} "E
2	0.21	CategoricalValue{String,UInt8} "Premiu	CategoricalValue{String,UInt8} "E
3	0.23	CategoricalValue{String,UInt8} "Good"	CategoricalValue{String,UInt8} "E
4	0.29	CategoricalValue{String,UInt8} "Premiu	CategoricalValue{String,UInt8} "I
5	0.31	CategoricalValue{String,UInt8} "Good"	CategoricalValue{String,UInt8} "J
6	0.24	CategoricalValue{String,UInt8} "Very G	CategoricalValue{String,UInt8} "J
7	0.24	CategoricalValue{String,UInt8} "Very G	CategoricalValue{String,UInt8} "I
8	0.26	CategoricalValue{String,UInt8} "Very G	CategoricalValue{String,UInt8} "H
9	0.22	CategoricalValue{String,UInt8} "Fair"	CategoricalValue{String,UInt8} "E
10	0.23	CategoricalValue{String,UInt8} "Very G	CategoricalValue{String,UInt8} "H

more

```
diamonds = dataset("ggplot2", "diamonds")
```

"diam.csv"

```
CSV.write("diam.csv", diamonds)
```

	Cut	Table
1	CategoricalValue{String,UInt8} "Ideal"	55.0
2	CategoricalValue{String,UInt8} "Premiu	61.0
3	CategoricalValue{String,UInt8} "Good"	65.0
4	CategoricalValue{String,UInt8} "Premiu	58.0
5	CategoricalValue{String,UInt8} "Good"	58.0
6	CategoricalValue{String,UInt8} "Very G	57.0
7	CategoricalValue{String,UInt8} "Very G	57.0
8	CategoricalValue{String,UInt8} "Very G	55.0
9	CategoricalValue{String,UInt8} "Fair"	61.0
10	CategoricalValue{String,UInt8} "Very G	61.0

```
diamonds[!, [:Cut,:Table]]
```

	Cut	Table
1	CategoricalValue{String,UInt8} "Premiu	61.0
2	CategoricalValue{String,UInt8} "Good"	65.0
3	CategoricalValue{String,UInt8} "Premiu	58.0
4	CategoricalValue{String,UInt8} "Good"	58.0

```
diamonds[2:5, [:Cut,:Table]]
```

```
CategoricalArrays.CategoricalArray{String,1,UInt8,String,CategoricalArrays.CategoricalVal
```

```
diamonds.Cut
```

```
CategoricalArrays.CategoricalArray{String,1,UInt8,String,CategoricalArrays.CategoricalVal
```

```
diamonds[!, :Cut]
```

	variable	mean	min	median	I
1	:Carat	0.79794	0.2	0.7	5.01
2	:Cut	nothing	CategoricalValue{String,UInt8} "Fair"	nothing	CategoricalValue{
3	:Color	nothing	CategoricalValue{String,UInt8} "D" (1/	nothing	CategoricalValue{
4	:Clarity	nothing	CategoricalValue{String,UInt8} "I1" (1	nothing	CategoricalValue{
5	:Depth	61.7494	43.0	61.8	79.0
6	:Table	57.4572	43.0	57.0	95.0
7	:Price	3932.8	326	2401.0	18823
8	:X	5.73116	0.0	5.7	10.74
9	:Y	5.73453	0.0	5.71	58.9
10	:Z	3.53873	0.0	3.53	31.8

```
describe(diamonds)
```

Cut

	Cut
1	CategoricalValue{String,UInt8} "Fair"
2	CategoricalValue{String,UInt8} "Fair"
3	CategoricalValue{String,UInt8} "Ideal"
4	CategoricalValue{String,UInt8} "Premiu
5	CategoricalValue{String,UInt8} "Ideal"
6	CategoricalValue{String,UInt8} "Premiu
7	CategoricalValue{String,UInt8} "Fair"
8	CategoricalValue{String,UInt8} "Fair"
9	CategoricalValue{String,UInt8} "Very G
10	CategoricalValue{String,UInt8} "Very G

```
diamonds[diamonds.X .>6, ["Cut"]] #отбор строк
```

Model

```
model = Continuous Response Model
        Number of observations: 53940
        Null Loglikelihood: -523775.49
        Loglikelihood: -471866.68
        R-squared: 0.8541
        LR Test: 103817.61 ~ χ²(4) ⇒  Pr > χ² = 0.0000
        Formula: Price ~ 1 + Carat + X + Y + Z
        Variance Covariance Estimator: OIM
```

	PE	SE	t-value	Pr > |t|	2.50%	97.50%
(Intercept)	1921.17	104.372	18.4069	<1e-74	1716.6	2125.74
Carat	10233.9	62.9361	162.608	<1e-99	10110.6	10357.3
X	-884.209	40.4701	-21.8485	<1e-99	-963.531	-804.887
Y	166.038	25.8582	6.42112	<1e-9	115.356	216.721
Z	-576.203	39.2819	-14.6684	<1e-47	-653.196	-499.211

```
model = fit(EconometricModel, @formula(Price~1+Carat+X+Y+Z), diamonds)
```

```
5×5 LinearAlgebra.Hermitian{Float64,Array{Float64,2}}:
 10893.6      6205.91    -2364.7      -59.1365  -540.0
  6205.91     3960.95    -1422.2      -20.4162  -310.451
 -2364.7     -1422.2      1637.83    -589.677   -708.05
   -59.1365    -20.4162   -589.677    668.645   -107.215
  -540.0      -310.451    -708.05    -107.215   1543.07
```

```
vcov(model)
```

```
5×5 LinearAlgebra.Hermitian{Float64,Array{Float64,2}}:
 100962.0    67384.6     -4462.5     1093.29   -38171.3
  67384.6    45850.7     -5241.13     982.147  -22377.8
  -4462.5    -5241.13    41915.8   -10874.8    -47866.5
   1093.29     982.147  -10874.8    15301.4     -7711.91
 -38171.3   -22377.8    -47866.5    -7711.91   105877.0
```

```
vcov(model, HC1)
```

```
5×5 LinearAlgebra.Hermitian{Float64,Array{Float64,2}}:
      1.00954e5   67379.6      -4462.17     1093.21    -38168.5
  67379.6         45847.3      -5240.74      982.074   -22376.1
  -4462.17        -5240.74     41912.7    -10874.0     -47863.0
   1093.21          982.074   -10874.0     15300.3      -7711.33
 -38168.5        -22376.1     -47863.0     -7711.33      1.05869e5
```

```
vcov(model, HC0) #Уайта
```

```
Float64[1921.17, 10233.9, -884.209, 166.038, -576.203]
```

```
coef(model)
```

```
Float64[282.993, 388.72, 288.327, -28.2579, -58.9271, 213.943, 213.702, 129.117, 39
```

```
residuals(model)
```

```
Float64[43.0065, -62.7202, 38.6735, 362.258, 393.927, 122.057, 122.298, 207.883, -5
```

```
fitted(model)
```

	PE	SE	t-value	Pr > |t|	2.50%	97.50%
(Intercept)	1921.17	104.372	18.4069	<1e-74	1716.6	2125.74
Carat	10233.9	62.9361	162.608	<1e-99	10110.6	10357.3
X	-884.209	40.4701	-21.8485	<1e-99	-963.531	-804.887
Y	166.038	25.8582	6.42112	<1e-9	115.356	216.721
Z	-576.203	39.2819	-14.6684	<1e-47	-653.196	-499.211

```
coeftable(model)
```

UndefVarError: diag not defined

1. top-level scope @ *Local: 1*

```
confint(model, level=0.9, se=sqrt.(diag(vcov(model, HC1))))
```

Tests

```
943745.3697994003
```

```
aic(model)
```

```
943798.7435649805
```

```
bic(model)
```

```
Int32[106276, 106276, 106929, 111556, 112225, 112896, 112896, 113569, 113569, 11424
```

```
diamonds[!, "plot2"] = diamonds[!, :Price] .^2
```

```
Float64[80085.3, 1.51103e5, 83132.2, 798.512, 3472.41, 45771.7, 45668.8, 16671.1, 1
```

```
diamonds[!, "resid2"] = residuals(model) .^2
```

```
aux_model =
Continuous Response Model
Number of observations: 53940
Null Loglikelihood: -938649.49
Loglikelihood: -930453.77
R-squared: 0.2621
LR Test: 16391.44 ~ χ²(5) ⟹  Pr > χ² = 0.0000
Formula: resid2 ~ 1 + Carat + X + (Carat ^ 2) + (X ^ 2) + X & Carat
Variance Covariance Estimator: OIM
```

	PE	SE	t-value	Pr > |t|	2.50%	97.50%
(Intercept)	-2.6743e7	1.50694e6	-17.7466	<1e-69	-2.96966e7	-2.37894e7
Carat	2.76898e7	1.22161e6	22.6667	<1e-99	2.52954e7	3.00842e7
X	3.50164e6	655055.0	5.34557	<1e-7	2.21773e6	4.78556e6
Carat ^ 2	3.18021e7	4.45418e5	71.3984	<1e-99	3.09291e7	3.26752e7
X ^ 2	9.90712e5	86860.2	11.4058	<1e-29	820465.0	1.16096e6
X & Carat	-1.45887e7	2.84973e5	-51.1933	<1e-99	-1.51472e7	-1.40301e7

```
aux_model = fit(EconometricModel,
@formula(resid2~Carat+X+Carat^2+X^2+X*Carat),diamonds)
```

Graphics

```
iris =
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	CategoricalValue{String,UInt8} "
2	4.9	3.0	1.4	0.2	CategoricalValue{String,UInt8} "
3	4.7	3.2	1.3	0.2	CategoricalValue{String,UInt8} "
4	4.6	3.1	1.5	0.2	CategoricalValue{String,UInt8} "
5	5.0	3.6	1.4	0.2	CategoricalValue{String,UInt8} "
6	5.4	3.9	1.7	0.4	CategoricalValue{String,UInt8} "
7	4.6	3.4	1.4	0.3	CategoricalValue{String,UInt8} "
8	5.0	3.4	1.5	0.2	CategoricalValue{String,UInt8} "
9	4.4	2.9	1.4	0.2	CategoricalValue{String,UInt8} "
10	4.9	3.1	1.5	0.1	CategoricalValue{String,UInt8} "

more

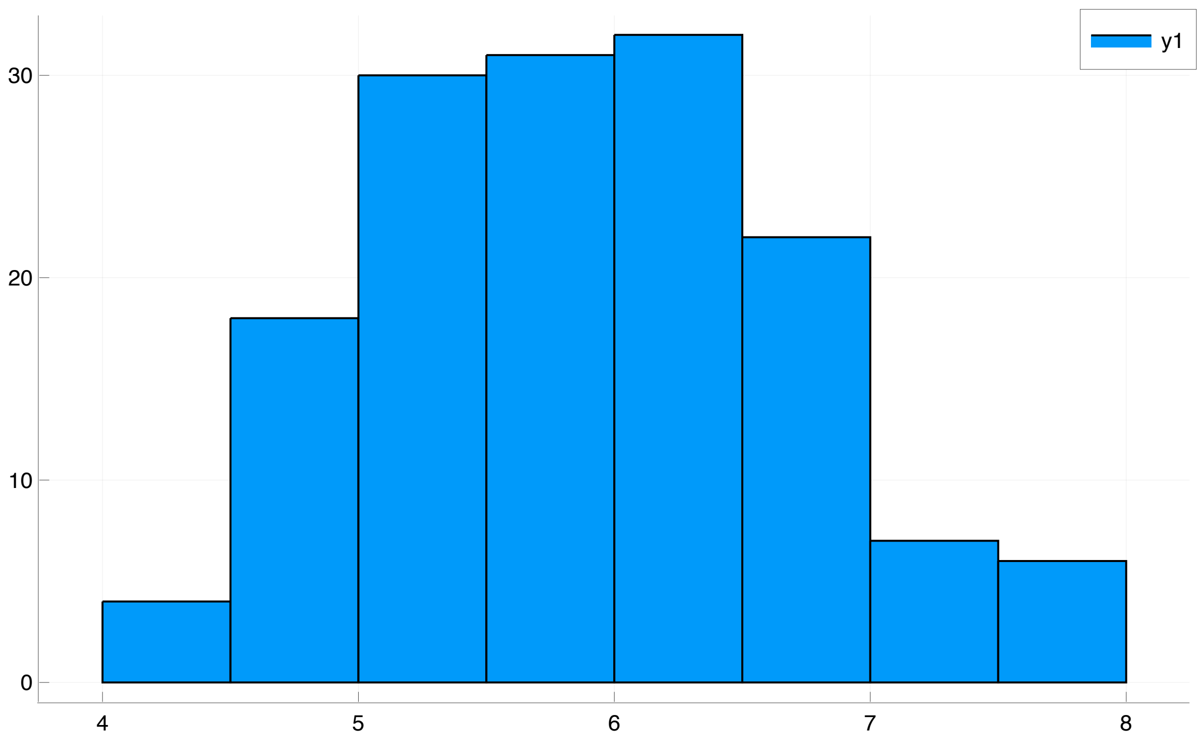
```
iris = dataset("datasets", "iris")
```

```
let
    Pkg.activate("Plots")
    Pkg.add("Plots")
    using Plots
end
```

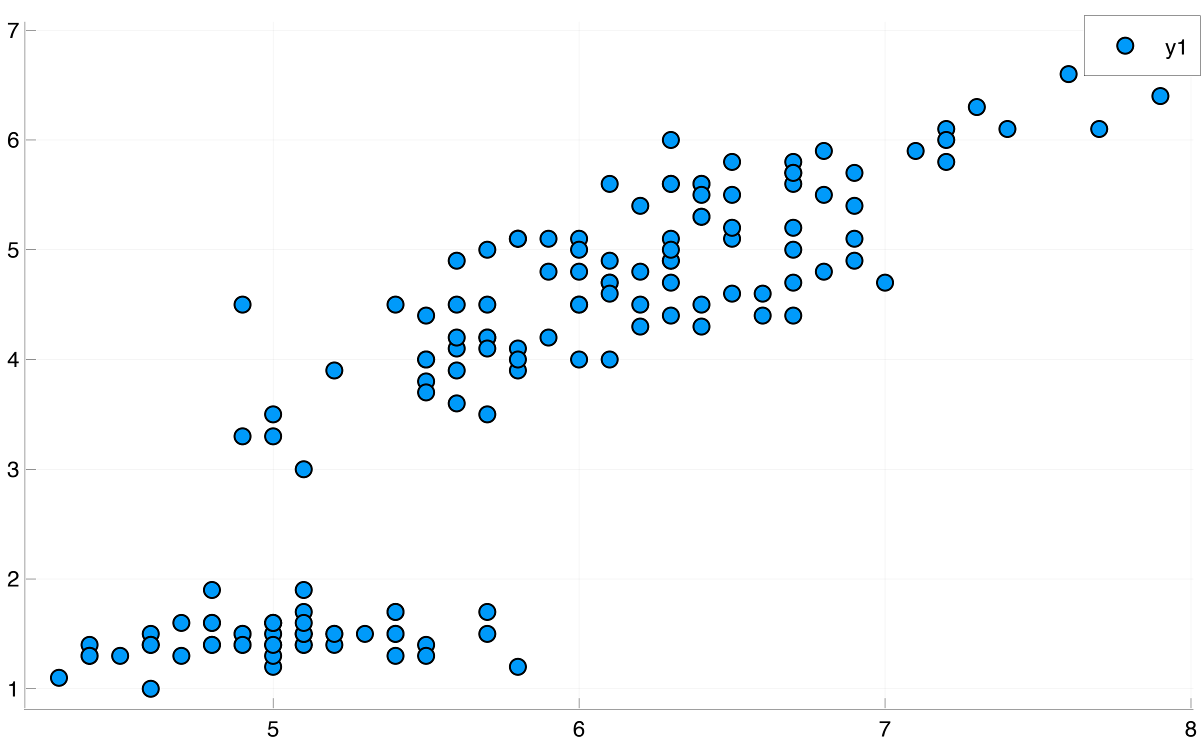
```
using Plots
```

```
Plots.PlotlyBackend()
```

```
plotly()
```




```
histogram(iris.SepalLength)
```




```
scatter(iris.SepalLength, iris.PetalLength)
```

```
Enter cell code...
```