```
PyObject <class 'sklearn.linear_model._base.LinearRegression'>
```

```
begin
    using RDatasets    ,ScikitLearn    ,DataFrames    ,CSV    ,Latexify    ,StatsPlots
    ,GLM
    @sk_import linear_model: LinearRegression
end
```

```
[[4.0], [4.0], [7.0], [7.0], [8.0], [9.0], [10.0], [10.0], [10.0], [11.0], [11.0], [12.0]
```

```
begin
    path="/Users/lunarcheung/Public/github/StatsWithJuliaBook/data/cars.csv"
    data=CSV.read(path, DataFrame).|>float
    row,col=size(data)
    X,y=data[!,1]|>Array,data[!,2]|>Array
    X=[[x] for x in X]
end
```

DataFrameRow (2 columns)

	speed	dist
	Float64	Float64
1	4.0	2.0

```
data|>first
```

1. 使用 scikitlearn 拟合结果

```
md"""
### 1. 使用 scikitlearn 拟合结果
"""
```

```
model =
▼ LinearRegression
LinearRegression()
```

```
model = LinearRegression()
```

```
reg=ScikitLearn.fit!(model, X, y);
```

```
0.6510793807582509
```

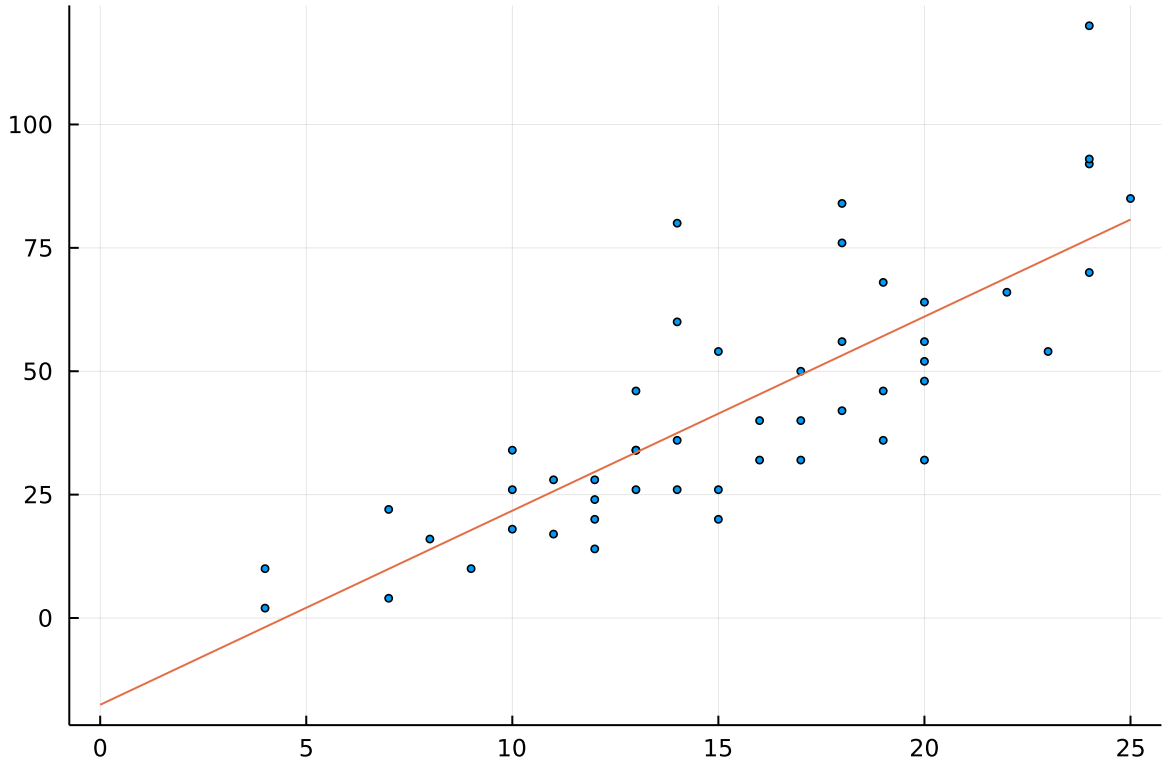
```
reg.score(X,y)
```

−1.8494598540146168 2.0	−1.8494598540146168 10.0	9.947766423357649 4.0	9.9477664233576 2

```
latexify([reg.predict(X)';y'],env=:mdtable)
```

```
line1 (generic function with 1 method)
```

```
line1(x)=reg.intercept_[1] + reg.coef_[1]*x
```




```
begin
    scatter(data[!,1],data[!,:2],label=false, ms=2)
    plot!(0:25,line1, label=false )
end
```

2. 使用 GLM 拟合的结果

```
md"""
### 2. 使用 GLM 拟合的结果
"""
```

```
ols =
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredCh

dist ~ 1 + speed

Coefficients:
```

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	-17.5791	6.75844	-2.60	0.0123	-31.1678	-3.99034
speed	3.93241	0.415513	9.46	<1e-11	3.09696	4.76785

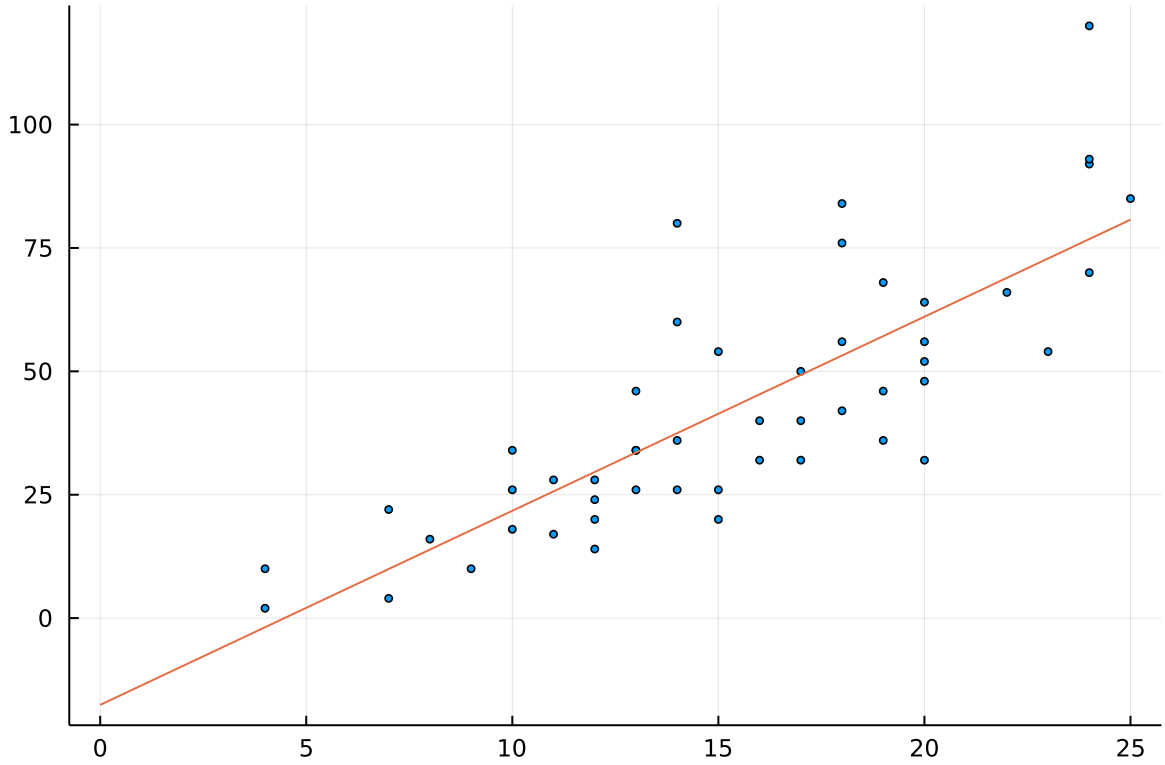
```
ols = lm(@formula(dist~speed), data)
```

```
line2 (generic function with 1 method)
```

```
line2(x)= coef(ols)'*[1, x]
```

.744992700729973	25.677401459854053	25.677401459854053	29.609810218978133	2
34.0	17.0	28.0	14.0	

```
latexify([GLM.predict(ols)';data[!,:2]'],env=:mdtable)
```




```
begin
    scatter(data[!,1],data[!,:2],label=false, ms=2)
    plot!(0:25,line2, label=false )
end
```

▼ LinearRegression

LinearRegression()

```
begin
    res2=ScikitLearn.fit!(model,[[0, 0], [1, 1], [2, 2]], [0, 1, 2])
end
```