

# Linear regression using R

June 13, 2021 - Ing. Duván Mejia Cortes

Hi! I'm [Duván Mejia](#) . I'm a Software Engineer and today I will write a little code using R and write my first **Linear Regression**. If you want to learn about Linear regression using R, you can read me. If you want to play with Linear regression using R, you can do some more research.

## 1. Linear regression: libraries

- `library(tidyverse)`
- `library(boot)`
- `library(car)`
- `library(QuantPsyc)`
- `library(ggplot2)`

build passing

## 2. Dataset

- Import csv file: You can download the file in the next url.
- <https://github.com/DuvanSGF/R/blob/main/2%20-%20Regresion%20Lineal/csv/sales.xlsx>

## 3. Run

Dillinger requires [R-4.1.0](#) or higher to run.

Install the libraries and execute the next commands.

```
> attach(sales)
> names(sales)
[1] "Publicidad" "ventas"
> class(ventas)
[1] "numeric"
> class(Publicidad)
[1] "numeric"
> ventas
[1] 330 120 360 270 220 170 70 210 200 300 290 70 150 190 240 100 250 210 280 230
210 230 320 210 230 250 60 330 150 150 180
[32] 80 180 130 320 280 200 130 190 150 230 310 340 240 180 220 40 190 290 340 250
190 120 230 190 210 170 310 90 140 300 340
[63] 170 100 200 80 100 70 50 240 160 290 140 210 300 230 280 160 200 110 110 70
100 190 70 360 360 300 120 150 220 280 300
[94] 140 290 180 140 210 250 250 120 290 60 140 290 160 100 160 150 140 230 230 30
80 190 90 120 150 230 150 210 180 140 360
[125] 10 240 270 290 220 230 220 240 260 170 130 270 140 60 210 210 240 210 200 140
90 120 100 360 180 150 110 90 160 230 40
[156] 60 230 230 120 150 120 60 280 120 230 230 40 140 360 210 260 250 200 150 250
100 260 210 290 220 70 110 250 320 300 180
[187] 180 200 320 140 100 120 230 150 250 190 240 250 230 110
```

```
> sales
```

	Publicidad	ventas
1	10.256	330
2	985.685	120
3	1445.563	360
4	1188.193	270
5	574.513	220
6	568.954	170
7	471.814	70
8	537.352	210
9	514.068	200
10	174.093	300
11	1720.806	290
12	611.479	70
13	251.192	150
14	97.972	190
15	406.814	240
16	265.398	100
17	1323.287	250
18	196.650	210
19	1326.598	280
20	1380.689	230
21	792.345	210
22	957.167	230
23	1789.659	320
24	656.137	210
25	613.697	230
26	313.362	250
27	336.510	60
28	1544.899	330
29	68.954	150
30	785.692	150
31	125.628	180
32	377.925	80
33	217.994	180
34	759.862	130
35	1163.444	320
36	842.957	280
37	125.179	200
38	236.598	130
39	669.811	190
40	612.234	150
41	922.019	230
42	50.000	310
43	2000.000	340
44	1054.027	240
45	385.045	180
46	1507.972	220
47	102.568	40
48	204.568	190
49	1170.918	290
50	689.547	340

51	784.220	250
52	405.913	190
53	179.778	120
54	607.258	230
55	1542.329	190
56	1112.470	210
57	856.985	170
58	836.331	310
59	236.908	90
60	1077.855	140
61	579.321	300
62	1500.000	340
63	731.364	170
64	25.689	100
65	391.749	200
66	233.999	80
67	275.700	100
68	56.895	70
69	255.117	50
70	566.501	240
71	102.568	160
72	250.568	290
73	68.594	140
74	642.786	210
75	1500.000	300
76	102.563	230
77	756.984	280
78	51.229	160
79	644.151	200
80	15.313	110
81	243.237	110
82	256.894	70
83	22.464	100
84	45.689	190
85	724.938	70
86	1126.461	360
87	1985.119	360
88	1837.516	300
89	135.986	120
90	237.703	150
91	976.641	220
92	1452.689	280
93	1600.000	300
94	268.598	140
95	900.889	290
96	982.063	180
97	201.356	140
98	746.024	210
99	1132.877	250
100	1000.000	250
101	75.896	120
102	1351.254	290

103	202.705	60
104	365.985	140
105	305.268	290
106	263.268	160
107	513.694	100
108	152.609	160
109	35.987	150
110	102.568	140
111	215.368	230
112	426.784	230
113	507.772	30
114	233.291	80
115	1035.433	190
116	102.642	90
117	526.142	120
118	624.538	150
119	912.349	230
120	215.994	150
121	561.963	210
122	474.760	180
123	231.523	140
124	678.596	360
125	70.922	10
126	1567.548	240
127	263.598	270
128	1423.568	290
129	715.678	220
130	777.237	230
131	509.430	220
132	964.110	240
133	583.627	260
134	923.373	170
135	344.392	130
136	1095.578	270
137	100.025	140
138	30.425	60
139	1080.342	210
140	799.899	210
141	1071.752	240
142	893.355	210
143	283.161	200
144	917.017	140
145	234.568	90
146	456.897	120
147	206.973	100
148	1294.099	360
149	826.859	180
150	564.158	150
151	192.607	110
152	10.652	90
153	45.689	160
154	42.568	230

155	20.456	40
156	635.192	60
157	1002.273	230
158	1177.047	230
159	507.638	120
160	215.689	150
161	526.480	120
162	26.895	60
163	883.877	280
164	9.104	120
165	103.568	230
166	169.583	230
167	429.504	40
168	223.639	140
169	145.585	360
170	985.968	210
171	500.922	260
172	226.652	250
173	1051.168	200
174	68.093	150
175	1547.159	250
176	393.774	100
177	804.282	260
178	801.577	210
179	450.562	290
180	26.598	220
181	179.061	70
182	345.687	110
183	295.840	250
184	2271.860	320
185	1134.575	300
186	601.434	180
187	45.298	180
188	759.518	200
189	832.869	320
190	56.894	140
191	709.399	100
192	56.895	120
193	767.134	230
194	503.172	150
195	700.929	250
196	910.851	190
197	888.569	240
198	800.615	250
199	1500.000	230
200	785.694	110

```
> modelo1 = lm(ventas ~ Publicidad, data = sales, na.action=na.exclude)
> summary(modelo1)
```

Call:

```
lm(formula = ventas ~ Publicidad, data = sales, na.action = na.exclude)
```

```

Residuals:
    Min       1Q   Median       3Q      Max
-152.949  -43.796   -0.393   37.040  211.866

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.341e+02  7.537e+00  17.799  <2e-16 ***
Publicidad   9.612e-02  9.632e-03   9.979  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 65.99 on 198 degrees of freedom
Multiple R-squared:  0.3346,    Adjusted R-squared:  0.3313
F-statistic: 99.59 on 1 and 198 DF,  p-value: < 2.2e-16

> sqrt(.3346)
[1] 0.5784462

> library(ggplot2)
> grafica1 = ggplot(sales, aes(Publicidad, ventas))
> grafica1 + geom_point()
> grafica1 + geom_point()+ geom_smooth(method = "lm", colours="Red")

```

## Development

Want to contribute? Great!

```

docker run -d -p 8000:8080 --restart=always --cap-add=SYS_ADMIN --name=dillinger
## License

MIT

**Free Software, Hell Yeah!**

```