

43260_LR.r

tanmay

2021-10-05

```
library("Metrics")
```

```
getwd()
```

```
## [1] "/home/tanmay/Downloads"
```

```
setwd('/home/tanmay/Downloads')  
getwd()
```

```
## [1] "/home/tanmay/Downloads"
```

```
df <- read.csv("IceCreamData.csv", header=TRUE, sep=',')  
head(df)
```

```
##   Temperature  Revenue  
## 1    24.56688  534.7990  
## 2    26.00519  625.1901  
## 3    27.79055  660.6323  
## 4    20.59534  487.7070  
## 5    11.50350  316.2402  
## 6    14.35251  367.9407
```

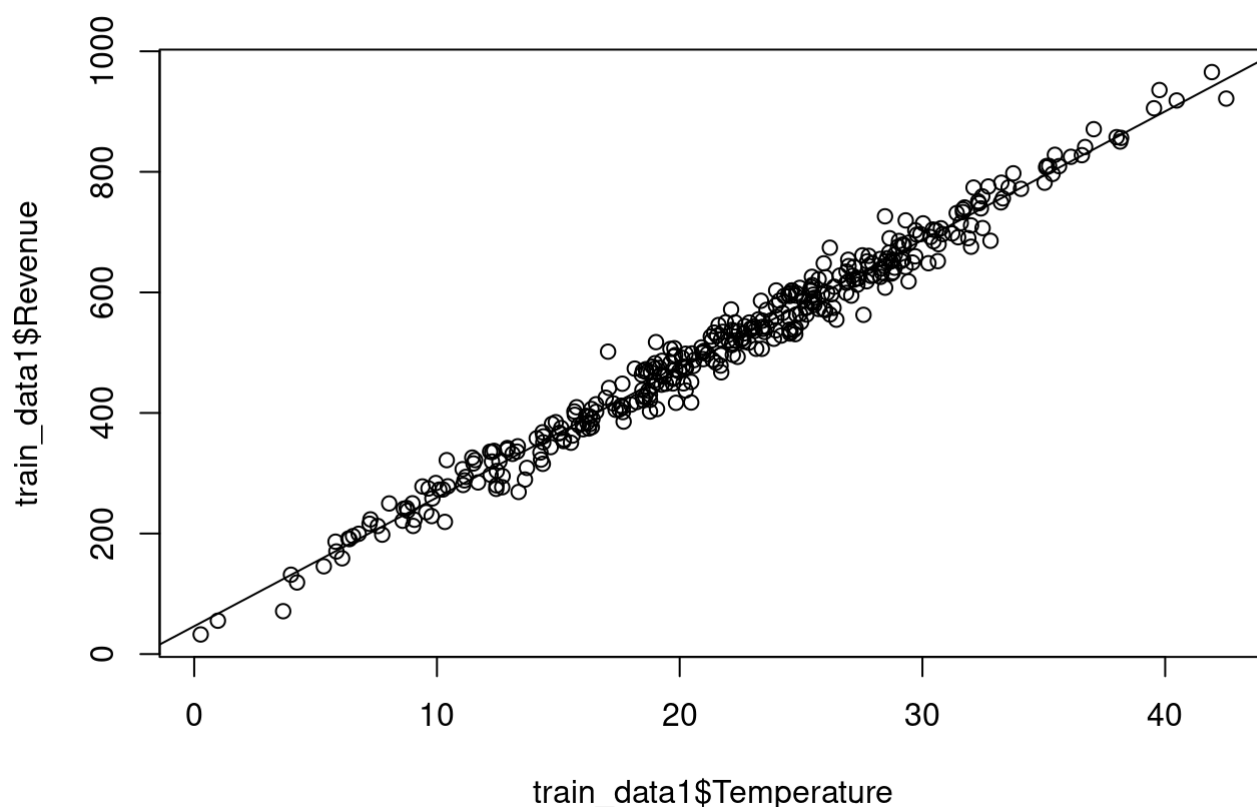
```
# Pass 1
```

```
train_data1 <- df[1:375,]  
test_data1 <- df[376:500,]
```

```
model1 <- lm(Revenue ~ Temperature, data=train_data1)  
summary(model1)
```

```
##
## Call:
## lm(formula = Revenue ~ Temperature, data = train_data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -72.240 -15.319   0.675  16.067  91.404
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.2635     3.7106   12.47  <2e-16 ***
## Temperature  21.3531     0.1593  134.03  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.24 on 373 degrees of freedom
## Multiple R-squared:  0.9797, Adjusted R-squared:  0.9796
## F-statistic: 1.796e+04 on 1 and 373 DF,  p-value: < 2.2e-16
```

```
plot(train_data1$Temperature, train_data1$Revenue)
abline(model1)
```



```
res1 <- predict(model1, test_data1)
print(res1)
```

```
##      376      377      378      379      380      381      382
## 670.86679 790.64889 284.93076 604.14076 897.38675 482.10308 427.86969
##      383      384      385      386      387      388      389
## 438.02568 257.66582 581.29375 304.30766 836.26739 360.83418 198.03072
##      390      391      392      393      394      395      396
## 859.74570 621.16699 311.75004 262.35163 421.52857 303.83520 695.56154
##      397      398      399      400      401      402      403
## 202.89826 529.54860 645.14228 761.90414 563.87154 871.11098 46.26347
##      404      405      406      407      408      409      410
## 566.19330 605.05079 159.59542 430.55625 707.46530 680.84711 743.44213
##      411      412      413      414      415      416      417
## 497.43665 442.39923 356.97754 938.11130 625.31236 473.67613 485.34841
##      418      419      420      421      422      423      424
## 633.83026 691.72840 509.19735 474.39790 415.65579 413.52531 803.11583
##      425      426      427      428      429      430      431
## 454.39998 450.95930 604.05314 472.96045 638.20326 495.24310 588.21551
##      432      433      434      435      436      437      438
## 561.16448 635.59020 233.23165 634.37082 386.13452 413.30148 661.08782
##      439      440      441      442      443      444      445
## 412.10245 234.04945 708.85852 456.30718 625.56160 688.58919 1007.15478
##      446      447      448      449      450      451      452
## 450.01908 368.75828 513.84242 676.35608 457.92365 684.29472 507.71858
##      453      454      455      456      457      458      459
## 579.84682 711.62796 786.74866 492.55637 839.04301 609.14336 800.69571
##      460      461      462      463      464      465      466
## 569.09316 871.95102 663.37331 305.12789 293.85532 418.98598 492.47191
##      467      468      469      470      471      472      473
## 906.87547 612.76695 889.99340 524.53062 245.04723 462.53745 520.83962
##      474      475      476      477      478      479      480
## 449.56368 510.73894 434.01977 363.70146 661.86936 587.21154 441.47508
##      481      482      483      484      485      486      487
## 526.34181 688.67674 409.22178 628.79985 150.16515 546.08191 308.94158
##      488      489      490      491      492      493      494
## 743.07746 402.94316 622.03419 555.00038 782.35253 538.58603 365.09801
##      495      496      497      498      499      500
## 582.48493 521.90252 748.63428 315.06015 523.77098 664.60207
```

```
mse1 <- mse(res1, test_data1$Revenue)
print(mse1)
```

```
## [1] 742.982
```

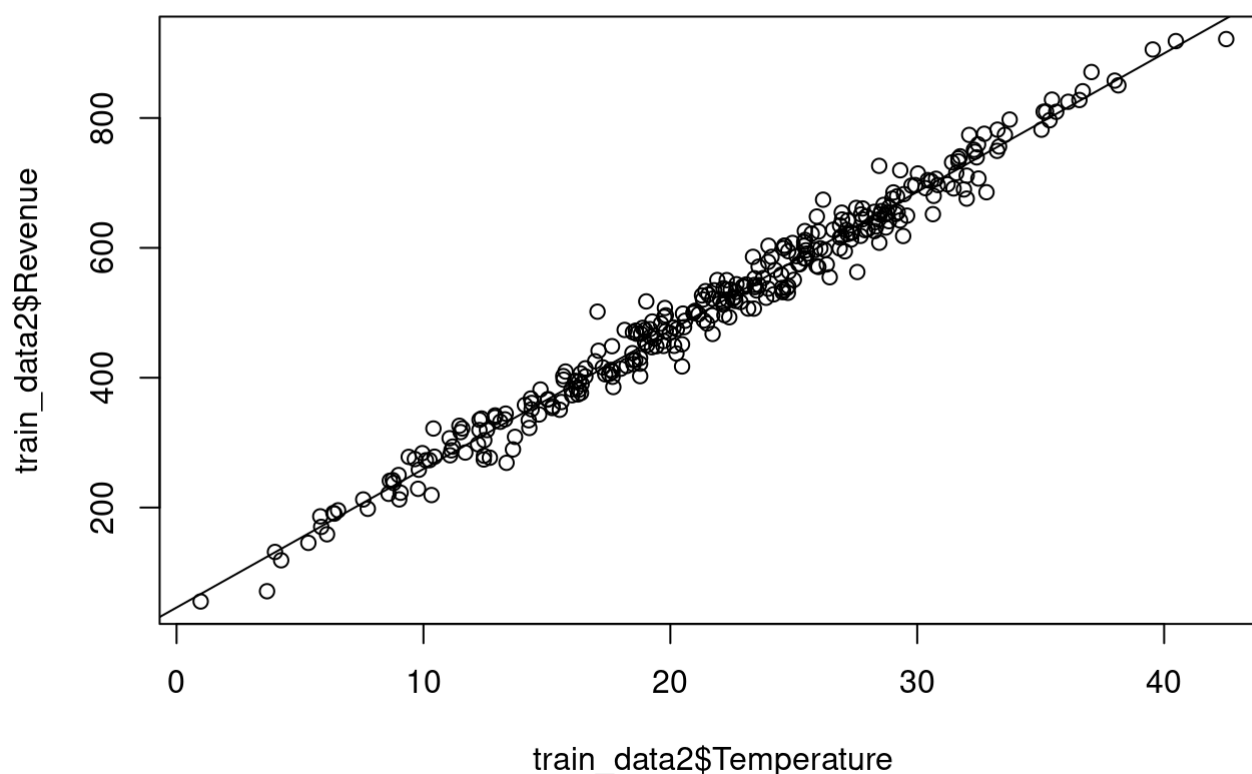
```
# Pass 2
```

```
train_data2 <- df[1:320,]
test_data2 <- df[321:500,]
```

```
model2 <- lm(Revenue ~ Temperature, data=train_data2)
summary(model2)
```

```
##
## Call:
## lm(formula = Revenue ~ Temperature, data = train_data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -71.61 -14.99   0.92  15.65  91.86
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.0778     4.0284   11.44  <2e-16 ***
## Temperature  21.3371     0.1733  123.14  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.12 on 318 degrees of freedom
## Multiple R-squared:  0.9795, Adjusted R-squared:  0.9794
## F-statistic: 1.516e+04 on 1 and 318 DF,  p-value: < 2.2e-16
```

```
plot(train_data2$Temperature, train_data2$Revenue)
abline(model2)
```



```
res2 <- predict(model2, test_data2)
print(res2)
```

##	321	322	323	324	325	326	327
##	368.65591	587.48711	439.53309	534.06873	352.50860	201.01373	584.36474
##	328	329	330	331	332	333	334
##	493.54089	468.02166	940.62293	657.36780	670.01130	384.12157	491.99530
##	335	336	337	338	339	340	341
##	695.93558	575.63744	469.60257	518.02631	772.85434	250.00212	591.26571
##	342	343	344	345	346	347	348
##	452.90311	538.70105	217.48179	679.93884	306.16433	794.89814	578.66074
##	349	350	351	352	353	354	355
##	860.83833	451.16806	445.26198	200.20354	317.15933	569.45181	894.52806
##	356	357	358	359	360	361	362
##	691.46565	568.24796	478.02877	477.65048	363.93563	660.31752	679.87823
##	363	364	365	366	367	368	369
##	608.73111	190.64088	542.09478	564.74576	594.82393	508.76005	604.93143
##	370	371	372	373	374	375	376
##	506.99901	449.00115	51.77537	464.66591	475.03626	557.83992	670.21124
##	377	378	379	380	381	382	383
##	789.90324	284.56553	603.53541	896.56080	481.58953	427.39694	437.54528
##	384	385	386	387	388	389	390
##	257.32110	580.70558	303.92785	835.48742	360.41185	197.73087	858.94806
##	391	392	393	394	395	396	397
##	620.54883	311.36464	262.00339	421.06058	303.45575	694.88741	202.59474
##	398	399	400	401	402	403	404
##	528.99936	644.50608	761.18010	563.29648	870.30480	46.07778	565.61649
##	405	406	407	408	409	410	411
##	604.44475	159.32448	430.08148	706.78222	680.18406	742.73199	496.91157
##	412	413	414	415	416	417	418
##	441.91555	356.55812	937.25472	624.69108	473.16892	484.83242	633.20257
##	419	420	421	422	423	424	425
##	691.05716	508.66342	473.89015	415.19223	413.06335	802.36080	453.90727
##	426	427	428	429	430	431	432
##	450.46918	603.44785	472.45378	637.57229	494.71966	587.62213	560.59145
##	433	434	435	436	437	438	439
##	634.96119	232.90532	633.74273	385.69317	412.83969	660.43963	411.64156
##	440	441	442	443	444	445	446
##	233.72250	708.17439	455.81304	624.94014	687.92031	1006.24625	449.52966
##	447	448	449	450	451	452	453
##	368.32999	513.30499	675.69641	457.42829	683.62907	507.18576	579.25974
##	454	455	456	457	458	459	460
##	710.94175	786.00594	492.03496	838.26095	608.53425	799.94250	568.51417
##	461	462	463	464	465	466	467
##	871.14420	662.72340	304.74747	293.48338	418.51991	491.95056	906.04239
##	468	469	470	471	472	473	474
##	612.15511	889.17302	523.98515	244.71201	462.03862	520.29693	449.07461
##	475	476	477	478	479	480	481
##	510.20385	433.54239	363.27698	661.22058	586.61892	440.99210	525.79498
##	482	483	484	485	486	487	488
##	688.00780	408.76305	628.17595	149.90130	545.52023	308.55829	742.36760
##	489	490	491	492	493	494	495
##	402.48916	621.41537	554.43199	781.61312	538.02999	364.67247	581.89587
##	496	497	498	499	500		
##	521.35903	747.92023	314.67226	523.22609	663.95124		

```
mse2 <- mse(res2, test_data2$Revenue)
print(mse2)
```

```
## [1] 708.4151
```

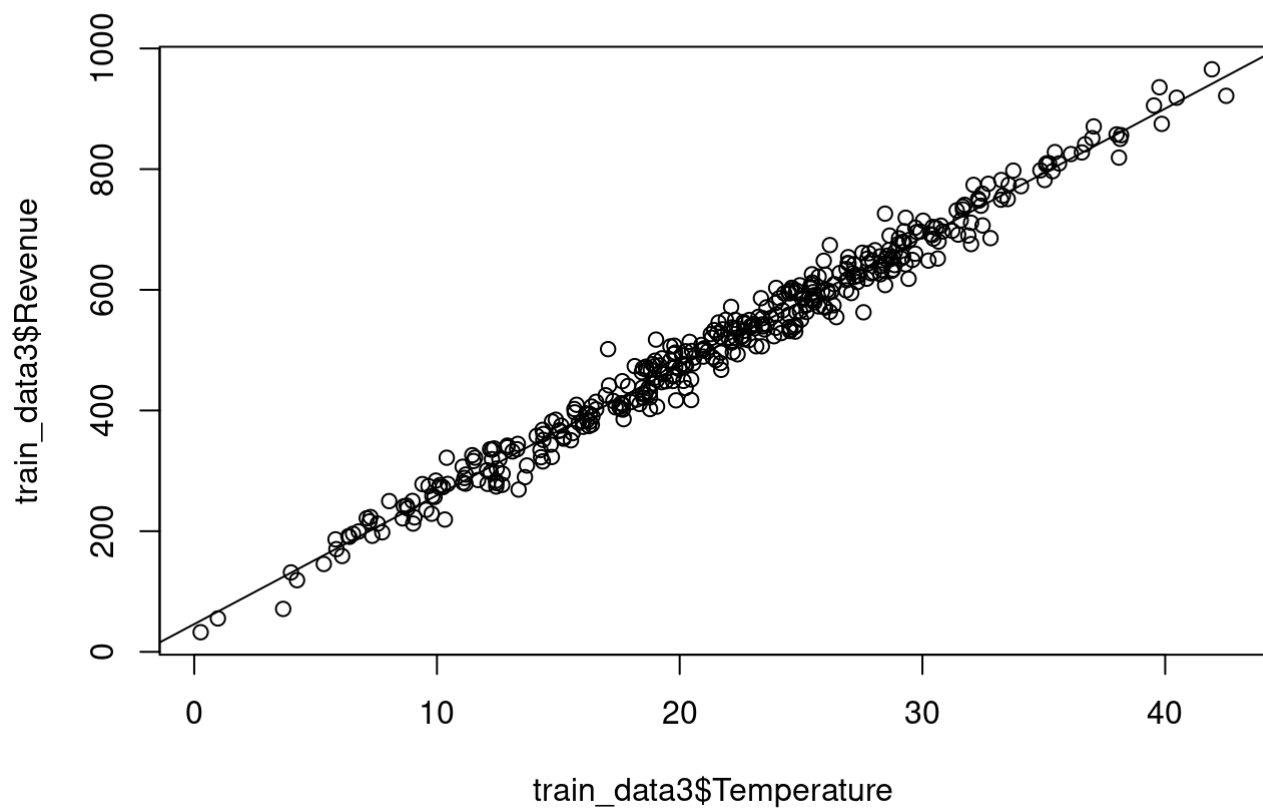
```
# Pass 3
```

```
train_data3 <- df[1:400,]  
test_data3 <- df[401:500,]
```

```
model3 <- lm(Revenue ~ Temperature, data=train_data3)  
summary(model3)
```

```
##  
## Call:  
## lm(formula = Revenue ~ Temperature, data = train_data3)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -72.137 -15.217   0.552  16.159  91.560   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  46.0234     3.4978   13.16  <2e-16 ***  
## Temperature  21.3581     0.1499  142.47  <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 24.03 on 398 degrees of freedom  
## Multiple R-squared:  0.9808, Adjusted R-squared:  0.9807   
## F-statistic: 2.03e+04 on 1 and 398 DF,  p-value: < 2.2e-16
```

```
plot(train_data3$Temperature, train_data3$Revenue)  
abline(model3)
```



```
res3 <- predict(model3, test_data3)
print(res3)
```

##	401	402	403	404	405	406	407
##	563.75176	871.06259	46.02342	566.07406	604.94058	159.38170	430.40550
##	408	409	410	411	412	413	414
##	707.37889	680.75451	743.36408	497.30144	442.25123	356.80969	938.07848
##	415	416	417	418	419	420	421
##	625.20686	473.53540	485.21038	633.72674	691.63833	509.06487	474.25733
##	422	423	424	425	426	427	428
##	415.50158	413.37060	803.05165	454.25476	450.81329	603.94270	472.81955
##	429	430	431	432	433	434	435
##	638.10076	495.10737	588.10138	561.04407	635.48709	233.03505	634.26743
##	436	437	438	439	440	441	442
##	385.97345	413.14672	660.99063	411.94741	233.85304	708.77243	456.16241
##	443	444	445	446	447	448	449
##	625.45616	688.49839	1007.13800	449.87285	368.59316	513.71102	676.26244
##	450	451	452	453	454	455	456
##	457.77925	684.20292	507.58575	579.73075	711.54252	786.68067	492.42002
##	457	458	459	460	461	462	463
##	838.98717	609.03410	800.63096	568.97459	871.90282	663.27666	304.94799
##	464	465	466	467	468	469	470
##	293.67281	418.83254	492.33554	906.83539	612.65853	889.94940	524.40170
##	471	472	473	474	475	476	477
##	244.85337	462.39413	520.70984	449.41734	510.60682	433.86982	363.53517
##	478	479	480	481	482	483	484
##	661.77235	587.09719	441.32687	526.21331	688.58596	409.06606	628.69516
##	485	486	487	488	489	490	491
##	149.94925	545.95800	308.76256	742.99933	402.78599	621.92792	554.87854
##	492	493	494	495	496	497	498
##	782.28352	538.46038	364.93204	582.36948	521.77299	748.55743	314.88256
##	499	500					
##	523.64189	664.50570					

```
mse3 <- mse(res3, test_data3$Revenue)
print(mse3)
```

```
## [1] 822.5496
```

```
# Pass 4
```

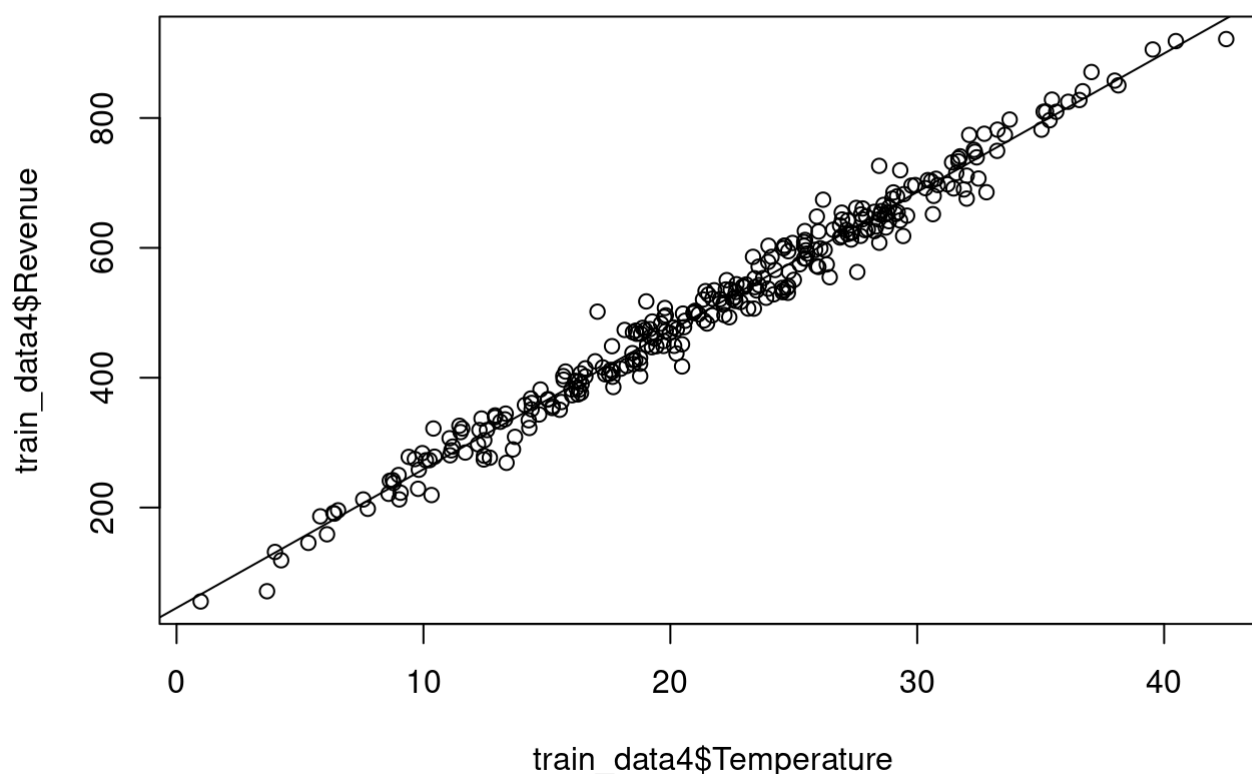
```
train_data4 <- df[1:300,]
test_data4 <- df[301:500,]
```

```
model4 <- lm(Revenue ~ Temperature, data=train_data4)
summary(model4)
```



```
##
## Call:
## lm(formula = Revenue ~ Temperature, data = train_data4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -71.35 -15.07   1.27  15.41  92.25
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  45.4884     4.1333   11.01  <2e-16 ***
## Temperature  21.3491     0.1782  119.78  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.28 on 298 degrees of freedom
## Multiple R-squared:  0.9797, Adjusted R-squared:  0.9796
## F-statistic: 1.435e+04 on 1 and 298 DF, p-value: < 2.2e-16
```

```
plot(train_data4$Temperature, train_data4$Revenue)
abline(model4)
```



```
res4 <- predict(model4, test_data4)
print(res4)
```

##	301	302	303	304	305	306	307
##	170.56102	618.91874	568.40158	513.08702	686.56296	499.83771	738.58561
##	308	309	310	311	312	313	314
##	410.35793	756.73290	545.32473	450.65104	307.46225	583.30211	623.37823
##	315	316	317	318	319	320	321
##	594.70349	521.80938	581.69336	518.42577	649.63936	509.01972	368.24798
##	322	323	324	325	326	327	328
##	587.20226	439.16503	533.75384	352.09159	200.51152	584.07813	493.20320
##	329	330	331	332	333	334	335
##	467.66962	940.53670	657.12226	669.77287	383.72234	491.65675	695.71173
##	336	337	338	339	340	341	342
##	575.34593	469.25141	517.70240	772.67374	249.52745	590.98299	452.54257
##	343	344	345	346	347	348	349
##	538.38876	216.98884	679.70599	305.72126	794.72995	578.37092	860.70722
##	350	351	352	353	354	355	356
##	450.80655	444.89714	199.70087	316.72244	569.15681	894.41590	691.23928
##	357	358	359	360	361	362	363
##	567.95229	477.68236	477.30385	363.52505	660.07363	679.64534	608.45821
##	364	365	366	367	368	369	370
##	190.13284	541.78440	564.44813	594.54321	508.43092	604.65639	506.66889
##	371	372	373	374	375	376	377
##	448.63841	51.18922	464.31198	474.68816	557.53839	669.97292	789.73224
##	378	379	380	381	382	383	384
##	284.11031	603.25959	896.44978	481.24512	427.02205	437.17610	256.85056
##	385	386	387	388	389	390	391
##	580.41692	303.48352	835.34205	359.99929	197.22681	858.81589	620.28258
##	392	393	394	395	396	397	398
##	310.92449	261.53548	420.68213	303.01115	694.66297	202.09341	528.68162
##	399	400	401	402	403	404	405
##	644.25330	760.99295	562.99803	870.17902	45.48843	565.31934	604.16944
##	406	407	408	409	410	411	412
##	158.79882	429.70810	706.56447	679.95134	742.53446	496.57577	441.54882
##	413	414	415	416	417	418	419
##	356.14339	937.16659	624.42716	472.81978	484.48983	632.94344	690.83056
##	420	421	422	423	424	425	426
##	508.33424	473.54141	414.81048	412.68040	802.19680	453.54729	450.10727
##	427	428	429	430	431	432	433
##	603.17198	472.10423	637.31561	494.38264	587.33736	560.29148	634.70304
##	434	435	436	437	438	439	440
##	232.42104	633.48390	385.29482	412.45661	660.19581	411.25781	233.23869
##	441	442	443	444	445	446	447
##	707.95742	455.45413	624.67635	687.69194	1006.19693	449.16722	367.92188
##	448	449	450	451	452	453	454
##	512.97842	675.46117	457.07029	683.39829	506.85574	578.97026	710.72634
##	455	456	457	458	459	460	461
##	785.83274	491.69643	838.11714	608.26123	799.77714	568.21865	871.01889
##	462	463	464	465	466	467	468
##	662.48087	304.30360	293.03318	418.14003	491.61198	905.93670	611.88413
##	469	470	471	472	473	474	475
##	889.05784	523.66459	244.23437	461.68322	519.97429	448.71191	509.87554
##	476	477	478	479	480	481	482
##	433.17096	362.86603	660.97720	586.33359	440.62485	525.47544	687.77948
##	483	484	485	486	487	488	489
##	408.37768	627.91398	149.37034	545.21178	308.11656	742.16986	402.10026
##	490	491	492	493	494	495	496
##	621.14961	554.12855	781.43745	537.71733	364.26231	581.60787	521.03699

```
##           497           498           499           500
## 747.72561 314.23397 522.90510 663.70939
```

```
mse4 <- mse(res4, test_data4$Revenue)
print(mse4)
```

```
## [1] 686.3274
```