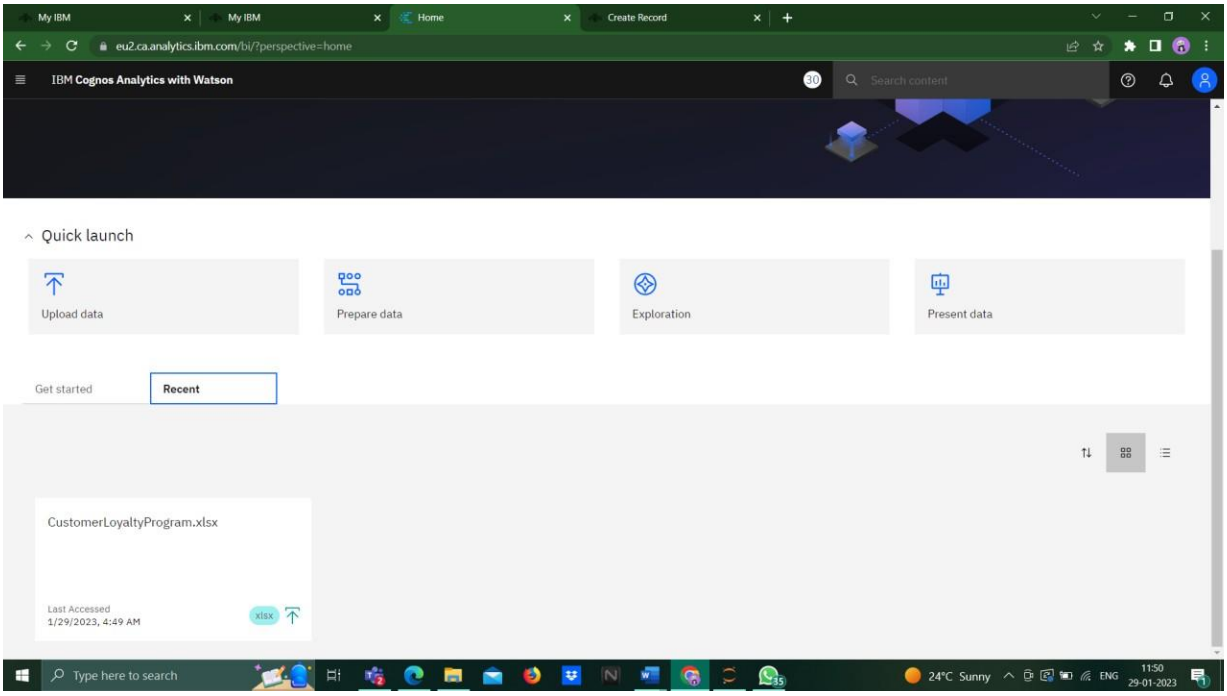
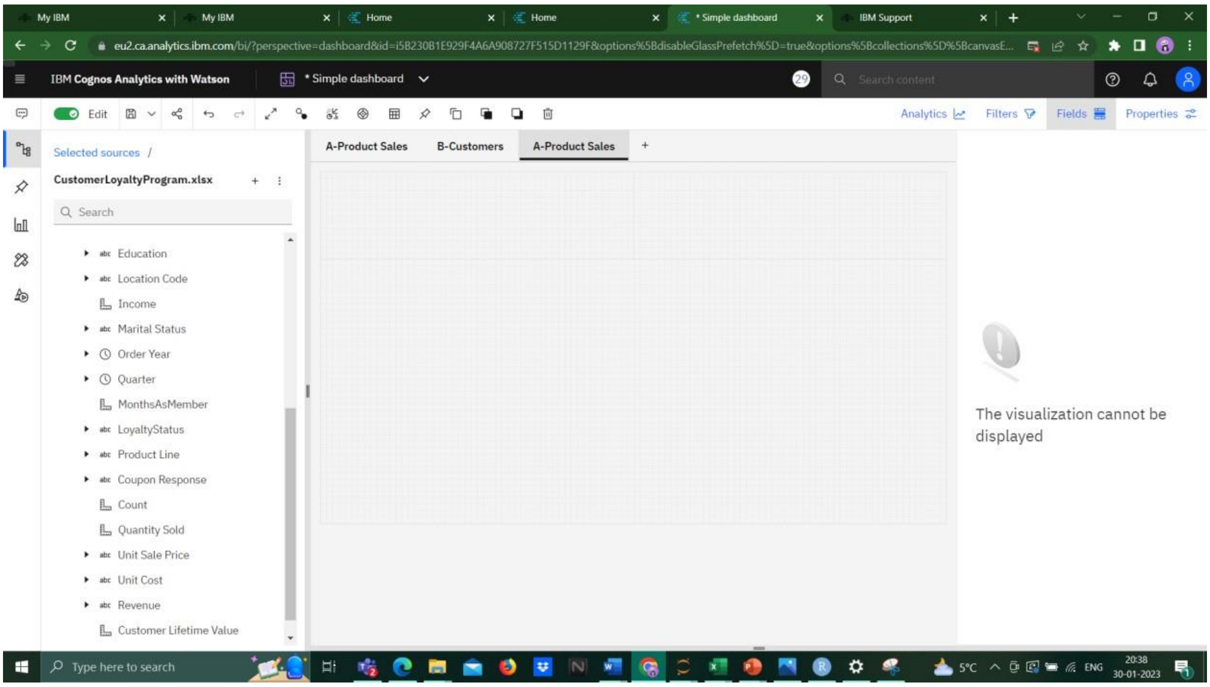
IBM Cognos Tutorial

Once the Dataset is uploaded, we can start creating a dashboard.



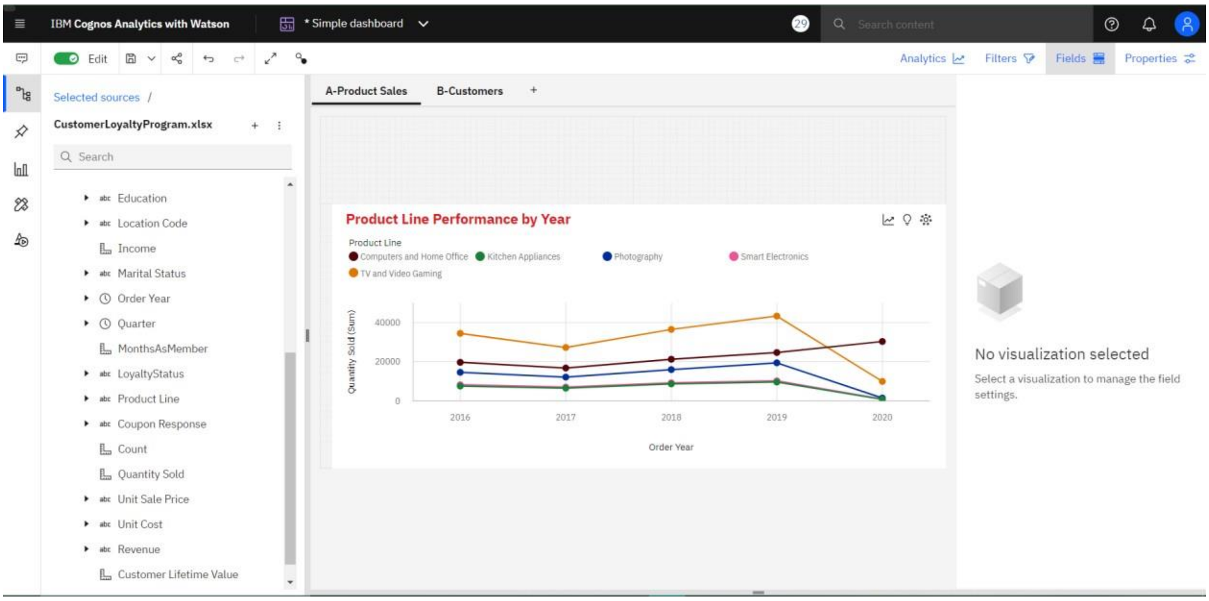
Dashboard A-Product Sales

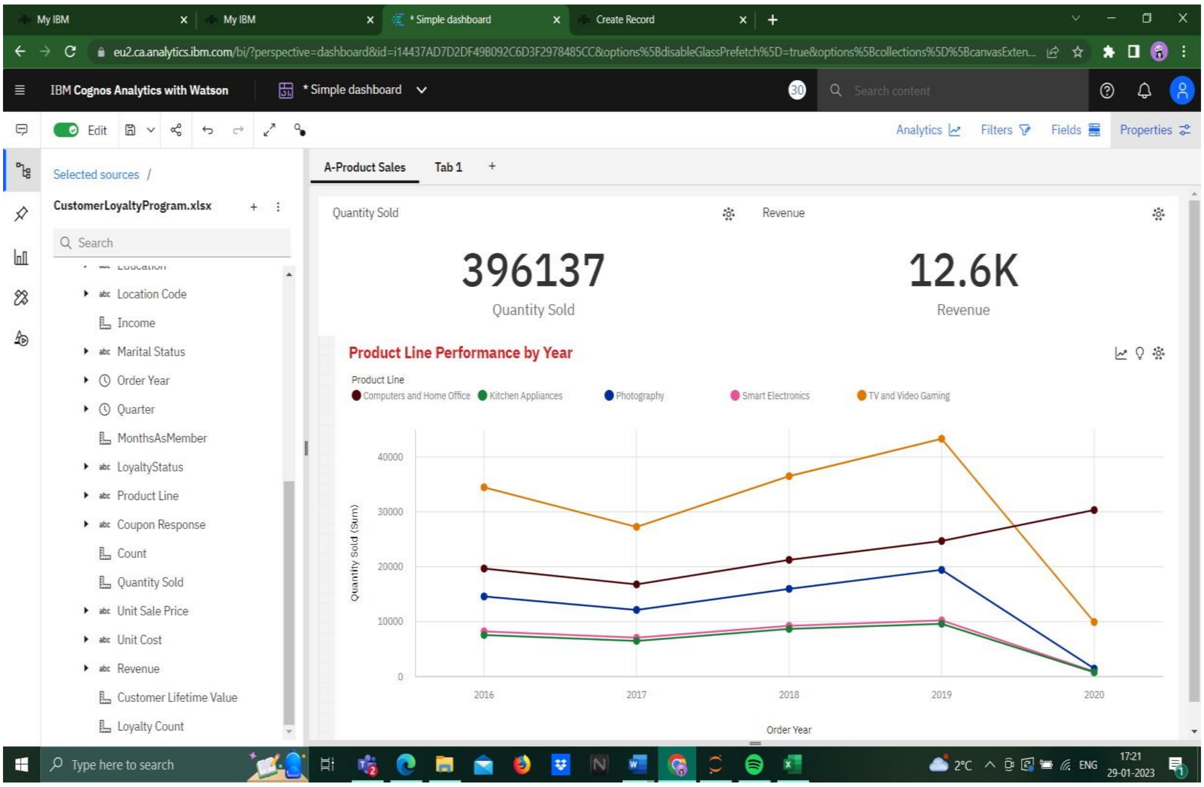
The dashboard template must be chosen.



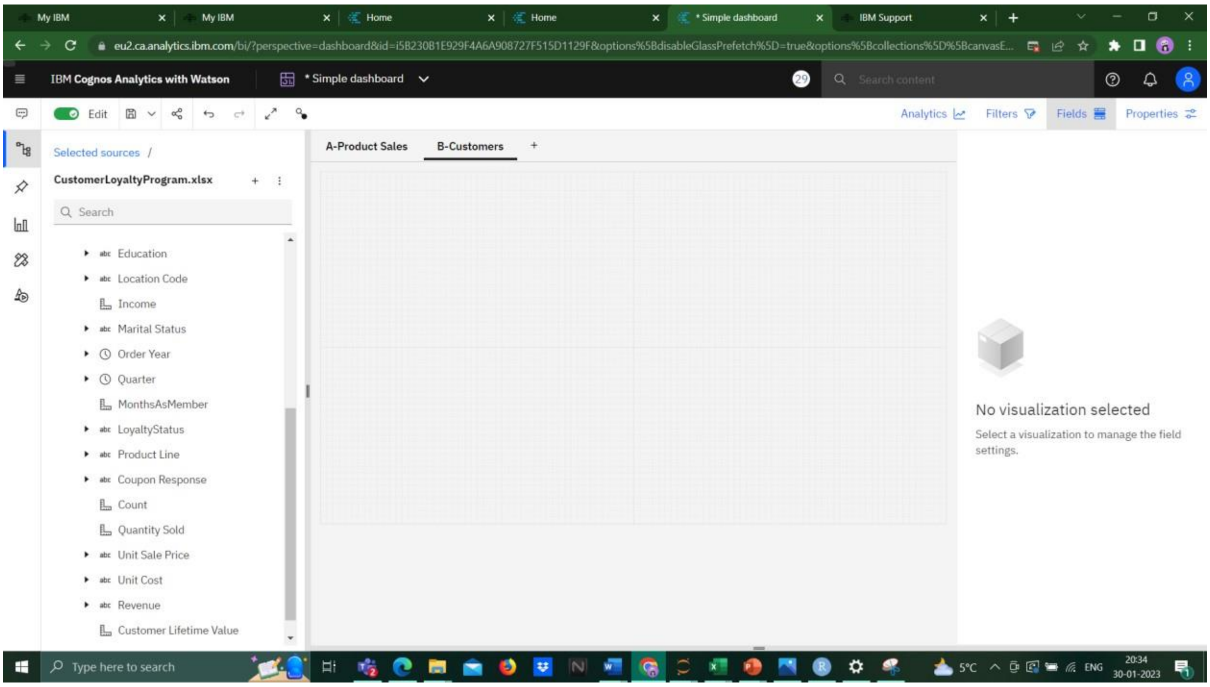
The visualization cannot be displayed

Visualization of data can be done by dragging the required fields into the panels of dashboard once it turns blue.



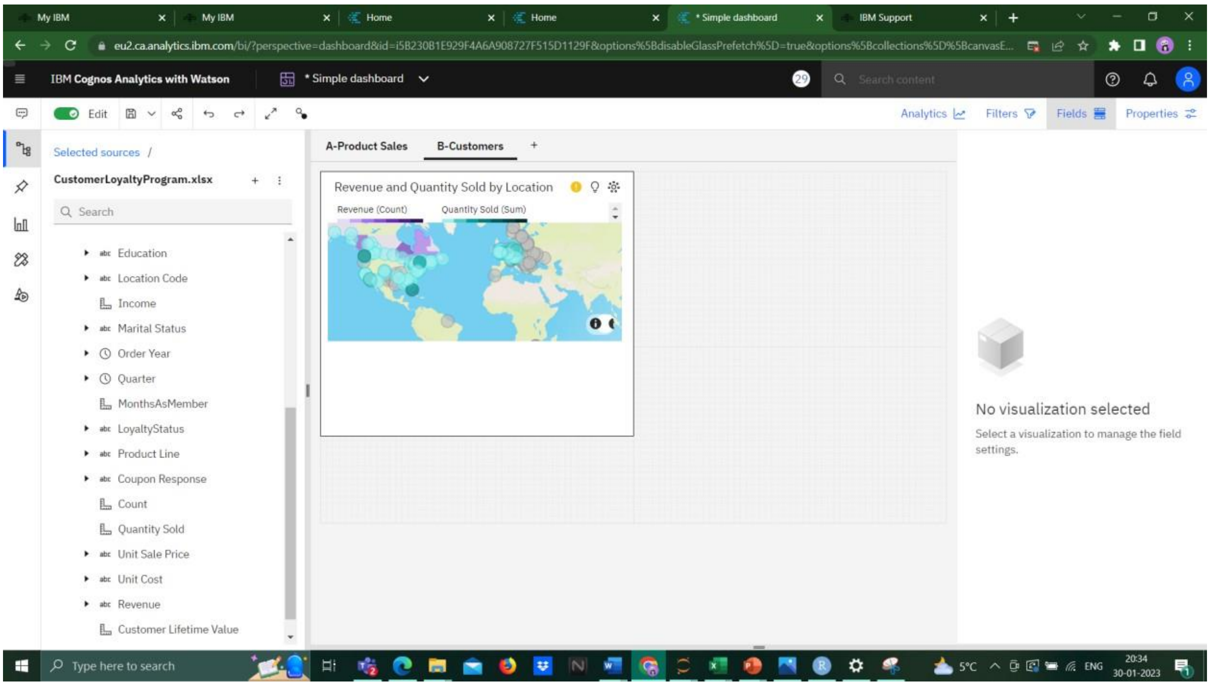


Dashboard B-Customers



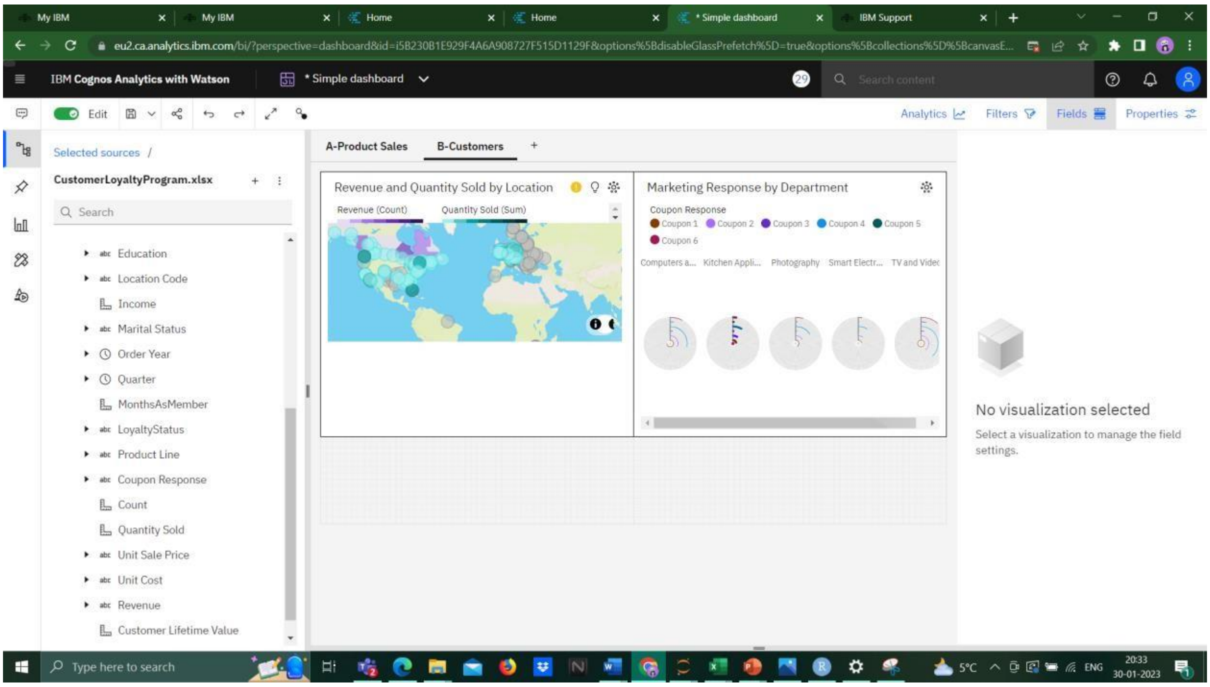
No visualization selected

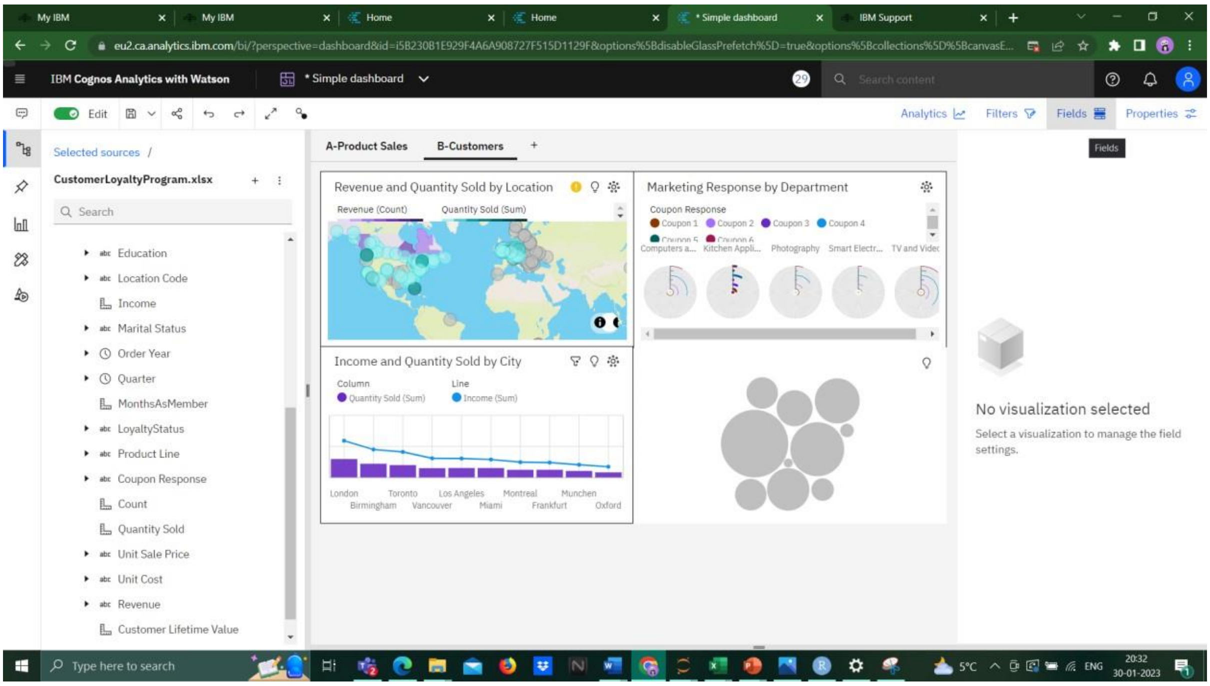
Select a visualization to manage the field settings.

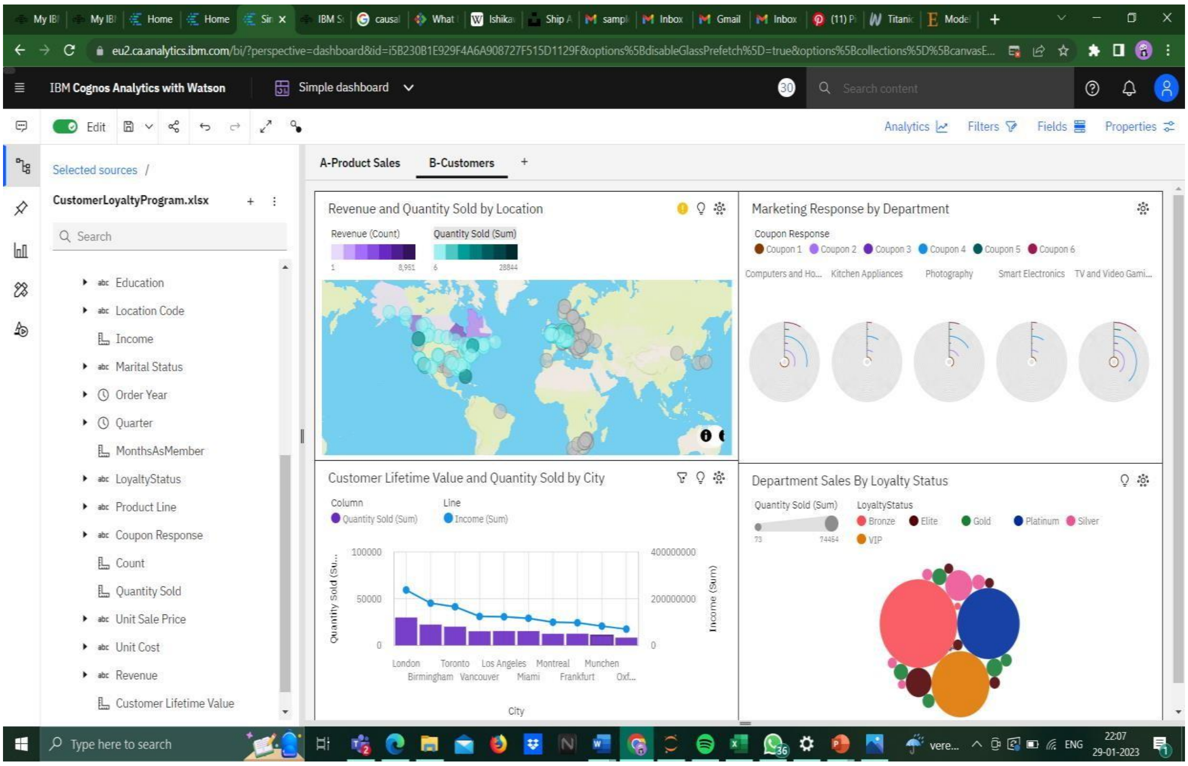


No visualization selected

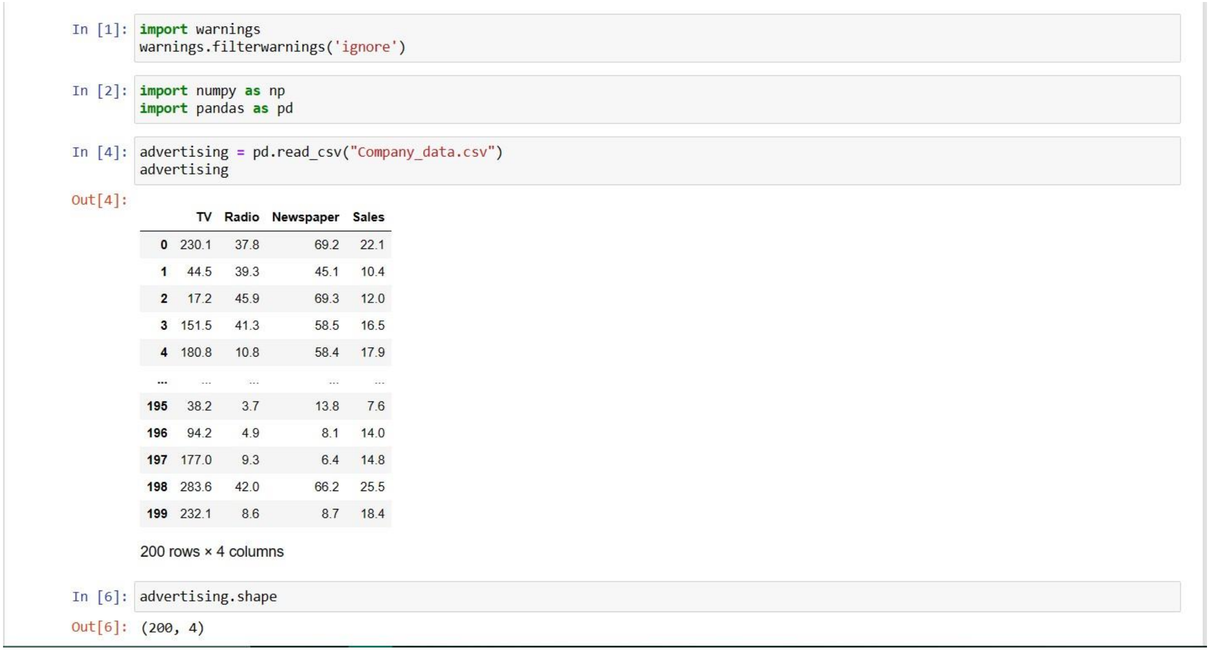
Select a visualization to manage the field settings.


IBM Cognos Analytics with Watson
* Simple dashboard
Analytics
Filters
Fields
Properties
Selected sources /
CustomerLoyaltyProgram.xlsx
Search
Education
Location Code
Income
Marital Status
Order Year
Quarter
MonthsAsMember
LoyaltyStatus
Product Line
Coupon Response
Count
Quantity Sold
Unit Sale Price
Unit Cost
Revenue
Customer Lifetime Value
A-Product Sales
B-Customers
Revenue and Quantity Sold by Location
Revenue (Count)
Quantity Sold (Sum)
Marketing Response by Department
Coupon Response
Coupon 1
Coupon 2
Coupon 3
Coupon 4
Coupon 5
Coupon 6
Computers a...
Kitchen Appli...
Photography
Smart Electr...
TV and Video
No visualization selected
Select a visualization to manage the field settings.
Type here to search
5°C
ENG
20:33
30-01-2023


IBM Cognos Analytics with Watson
* Simple dashboard
Analytics
Filters
Fields
Properties
Fields
Selected sources /
CustomerLoyaltyProgram.xlsx
Search
Education
Location Code
Income
Marital Status
Order Year
Quarter
MonthsAsMember
LoyaltyStatus
Product Line
Coupon Response
Count
Quantity Sold
Unit Sale Price
Unit Cost
Revenue
Customer Lifetime Value
A-Product Sales
B-Customers
Revenue and Quantity Sold by Location
Revenue (Count)
Quantity Sold (Sum)
Marketing Response by Department
Coupon Response
Coupon 1
Coupon 2
Coupon 3
Coupon 4
Coupon 5
Coupon 6
Computers a...
Kitchen Appli...
Photography
Smart Electr...
TV and Video
Income and Quantity Sold by City
Column
Quantity Sold (Sum)
Line
Income (Sum)
London
Birmingham
Toronto
Vancouver
Los Angeles
Miami
Montreal
Frankfurt
Munchen
Oxford
No visualization selected
Select a visualization to manage the field settings.
Type here to search
5°C
ENG
20:32
30-01-2023


IBM Cognos Analytics with Watson
Simple dashboard
Search content
Edit
Analytics
Filters
Fields
Properties
Selected sources /
CustomerLoyaltyProgram.xlsx
Search
Education
Location Code
Income
Marital Status
Order Year
Quarter
MonthsAsMember
LoyaltyStatus
Product Line
Coupon Response
Count
Quantity Sold
Unit Sale Price
Unit Cost
Revenue
Customer Lifetime Value
A-Product Sales
B-Customers
Revenue and Quantity Sold by Location
Revenue (Count)
Quantity Sold (Sum)
1
8,961
6
28844
Marketing Response by Department
Coupon Response
Coupon 1
Coupon 2
Coupon 3
Coupon 4
Coupon 5
Coupon 6
Computers and Ho...
Kitchen Appliances
Photography
Smart Electronics
TV and Video Gami...
Customer Lifetime Value and Quantity Sold by City
Column
Quantity Sold (Sum)
Line
Income (Sum)
100000
50000
0
400000000
200000000
0
Quantity Sold (Su...
Income (Sum)
London
Birmingham
Toronto
Vancouver
Los Angeles
Miami
Montreal
Frankfurt
Munchen
Oxf...
City
Department Sales By Loyalty Status
Quantity Sold (Sum)
73
74464
LoyaltyStatus
Bronze
Elite
Gold
Platinum
Silver
VIP
Type here to search
ENG
22:07
29-01-2023

Python Analysis



In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
advertising = pd.read_csv("Company_data.csv")
advertising
```

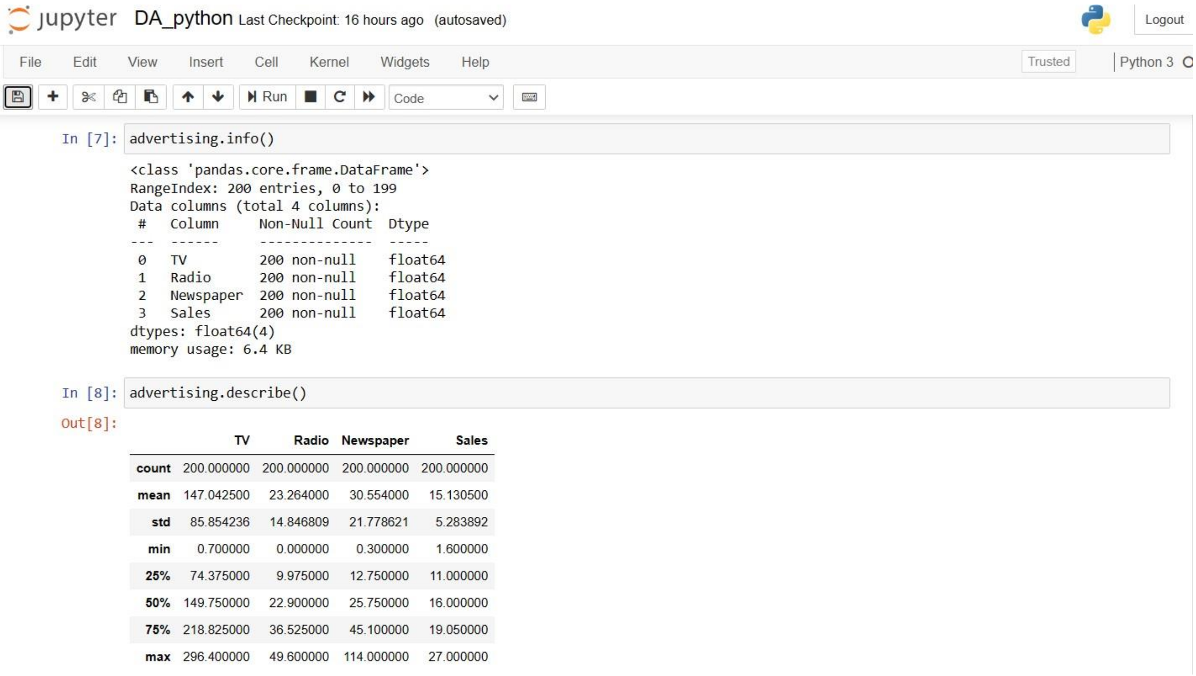
Out[4]:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

In [6]:
```
advertising.shape
```

Out[6]:
```
(200, 4)
```



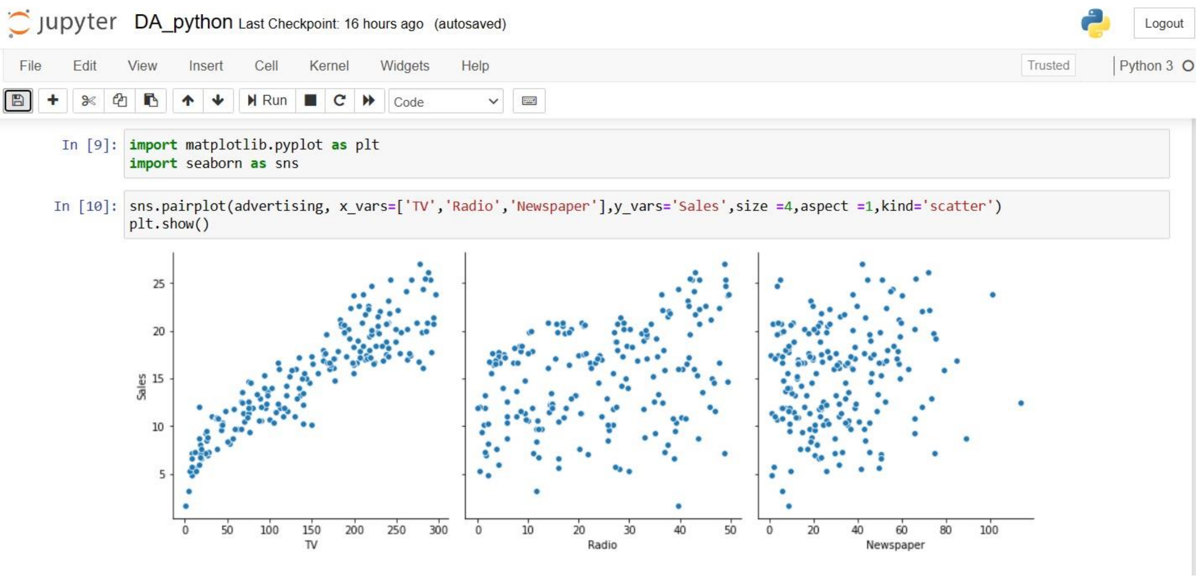
In [7]:
```
advertising.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

In [8]:
```
advertising.describe()
```

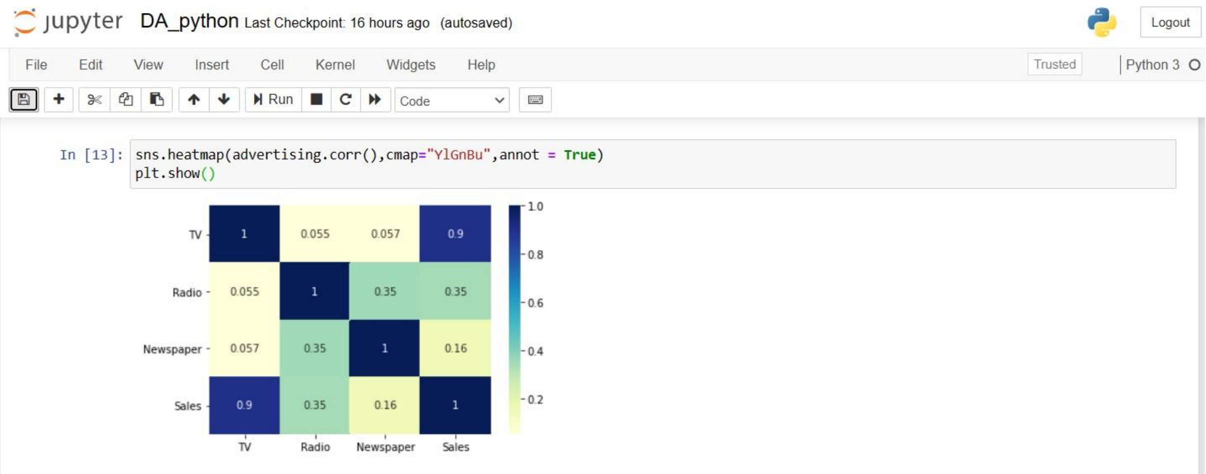
Out[8]:

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000



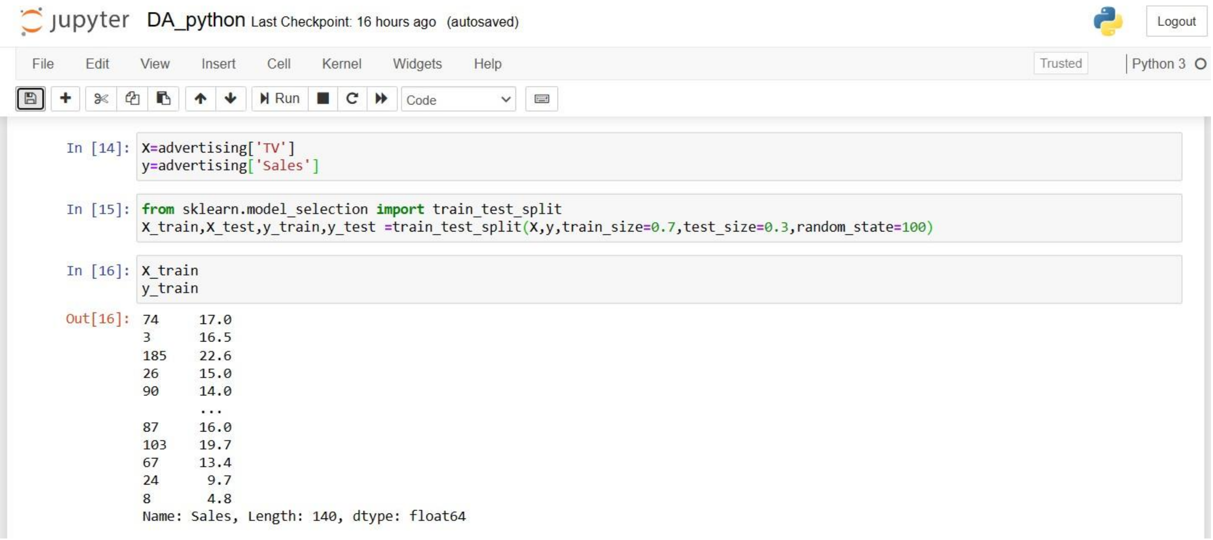
```
In [9]: import matplotlib.pyplot as plt
        import seaborn as sns
```

```
In [10]: sns.pairplot(advertising, x_vars=['TV','Radio','Newspaper'],y_vars='Sales',size =4,aspect =1,kind='scatter')
         plt.show()
```


jupyter DA_python Last Checkpoint: 16 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
1.0
0.8
0.6
0.4
0.2

```
In [13]: sns.heatmap(advertising.corr(),cmap="YlGnBu",annot = True)
         plt.show()
```

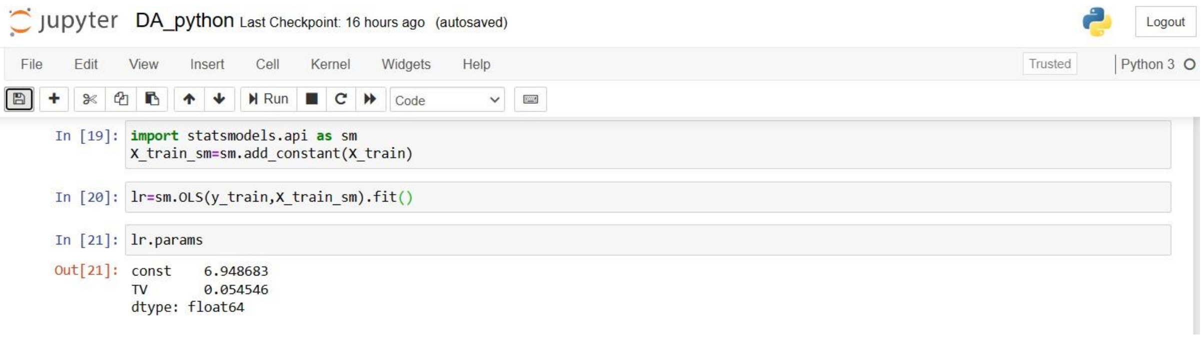
	TV	Radio	Newspaper	Sales
TV	1	0.055	0.057	0.9
Radio	0.055	1	0.35	0.35
Newspaper	0.057	0.35	1	0.16
Sales	0.9	0.35	0.16	1


jupyter DA_python Last Checkpoint: 16 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

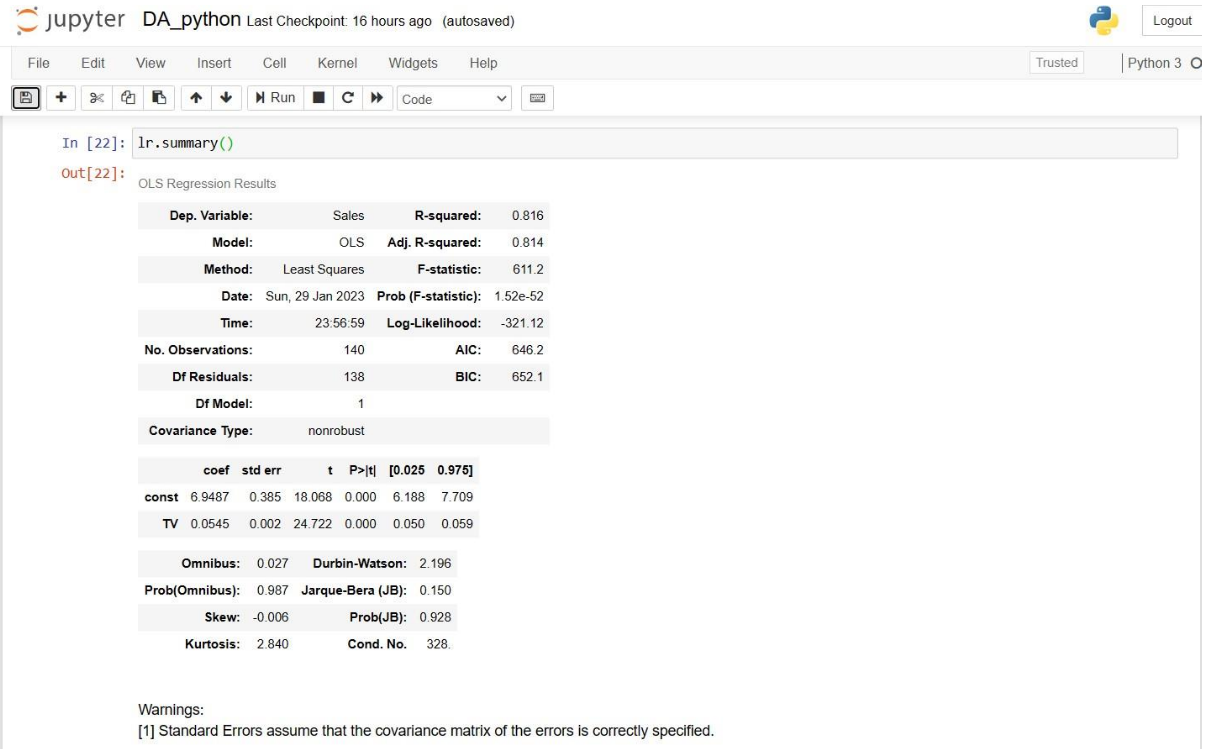
```
In [14]: X=advertising['TV']
         y=advertising['Sales']
```

```
In [15]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test =train_test_split(X,y,train_size=0.7,test_size=0.3,random_state=100)
```

```
In [16]: X_train
         y_train

Out[16]: 74     17.0
         3      16.5
         185    22.6
         26     15.0
         90     14.0
                ...
         87     16.0
         103    19.7
         67     13.4
         24      9.7
         8       4.8
         Name: Sales, Length: 140, dtype: float64
```



```
In [19]: import statsmodels.api as sm
         X_train_sm=sm.add_constant(X_train)

In [20]: lr=sm.OLS(y_train,X_train_sm).fit()

In [21]: lr.params

Out[21]: const    6.948683
         TV       0.054546
         dtype: float64
```



```
In [22]: lr.summary()

Out[22]:
```

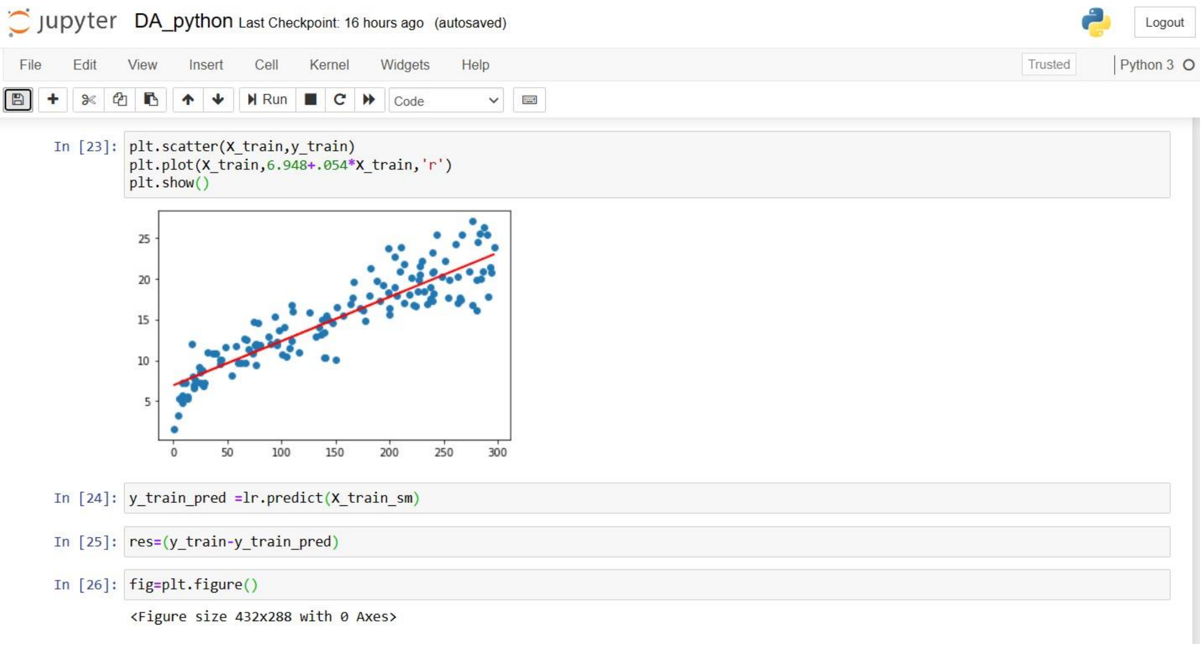
OLS Regression Results

Dep. Variable:	Sales	**R-squared:**	0.816
Model:	OLS	**Adj. R-squared:**	0.814
Method:	Least Squares	**F-statistic:**	611.2
Date:	Sun, 29 Jan 2023	**Prob (F-statistic):**	1.52e-52
Time:	23:56:59	**Log-Likelihood:**	-321.12
No. Observations:	140	**AIC:**	646.2
Df Residuals:	138	**BIC:**	652.1
Df Model:	1		
Covariance Type:	nonrobust		

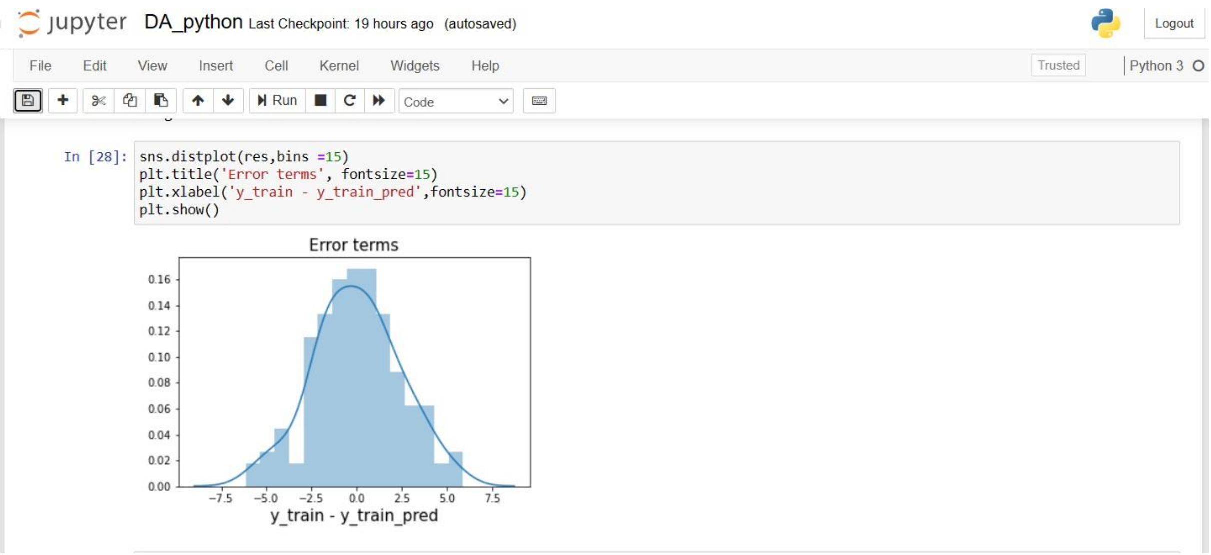
	coef	std err	t	P>|t|	[0.025	0.975]
const	6.9487	0.385	18.068	0.000	6.188	7.709
TV	0.0545	0.002	24.722	0.000	0.050	0.059

Omnibus:	0.027	**Durbin-Watson:**	2.196
Prob(Omnibus):	0.987	**Jarque-Bera (JB):**	0.150
Skew:	-0.006	**Prob(JB):**	0.928
Kurtosis:	2.840	**Cond. No.**	328.

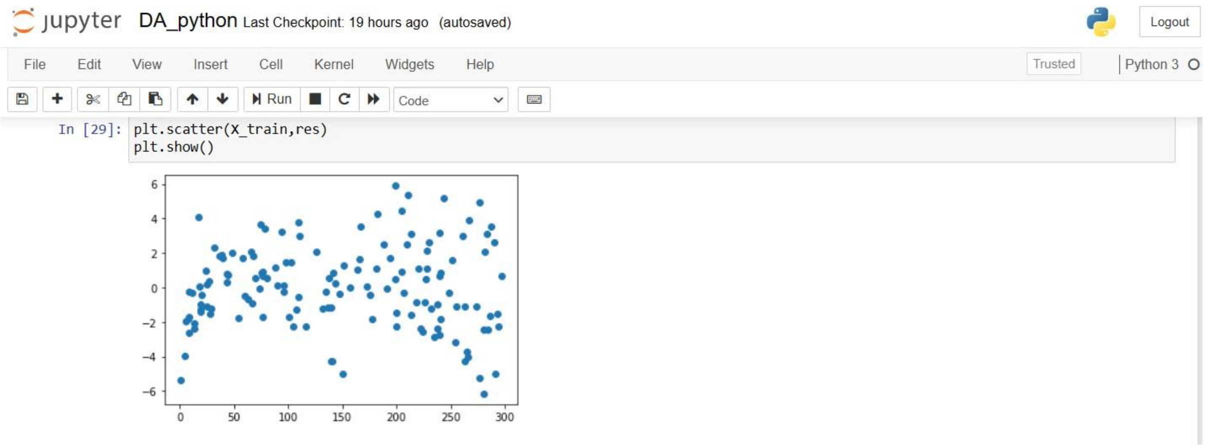
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



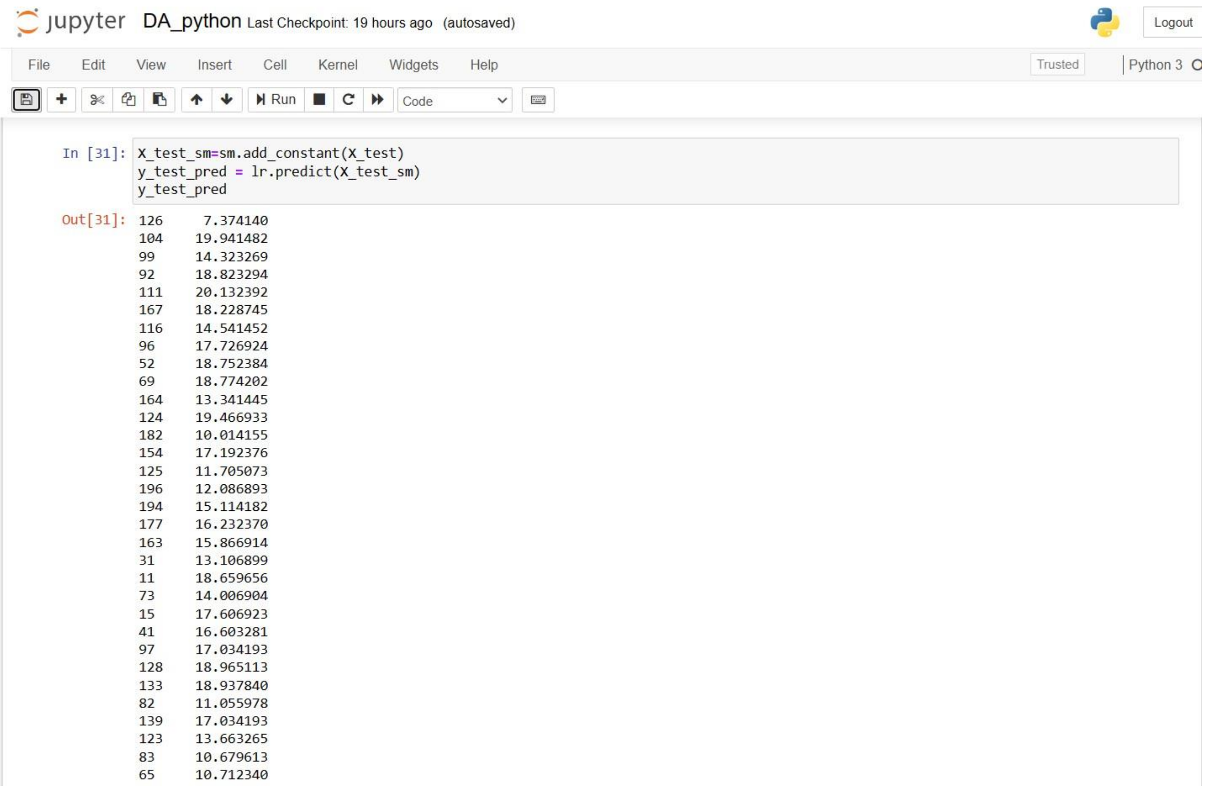
```
In [23]: plt.scatter(X_train,y_train)
         plt.plot(X_train,6.948+.054*X_train,'r')
         plt.show()

In [24]: y_train_pred =lr.predict(X_train_sm)

In [25]: res=(y_train-y_train_pred)

In [26]: fig=plt.figure()

         <Figure size 432x288 with 0 Axes>
```



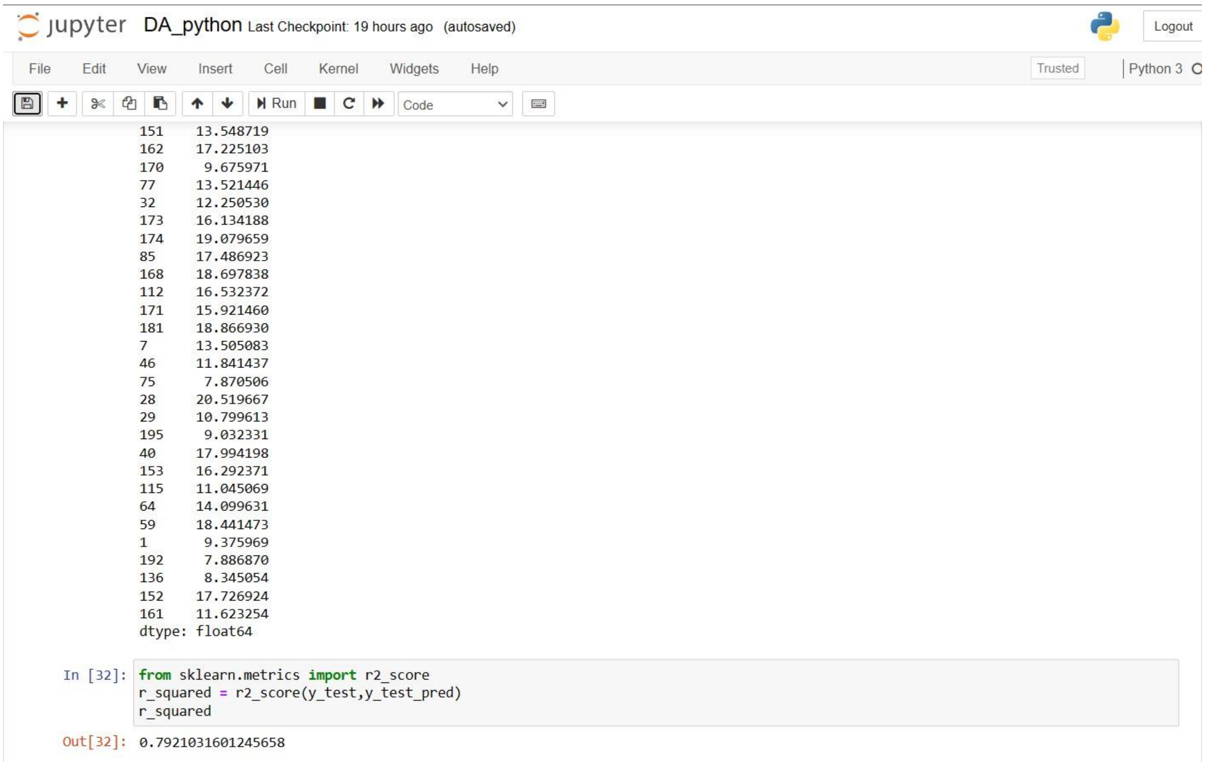
```
In [28]: sns.distplot(res,bins =15)
         plt.title('Error terms', fontsize=15)
         plt.xlabel('y_train - y_train_pred',fontsize=15)
         plt.show()
```


jupyter DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
6
4
2
0
−2
−4
−6
0
50
100
150
200
250
300

```
In [29]: plt.scatter(X_train,res)
         plt.show()
```

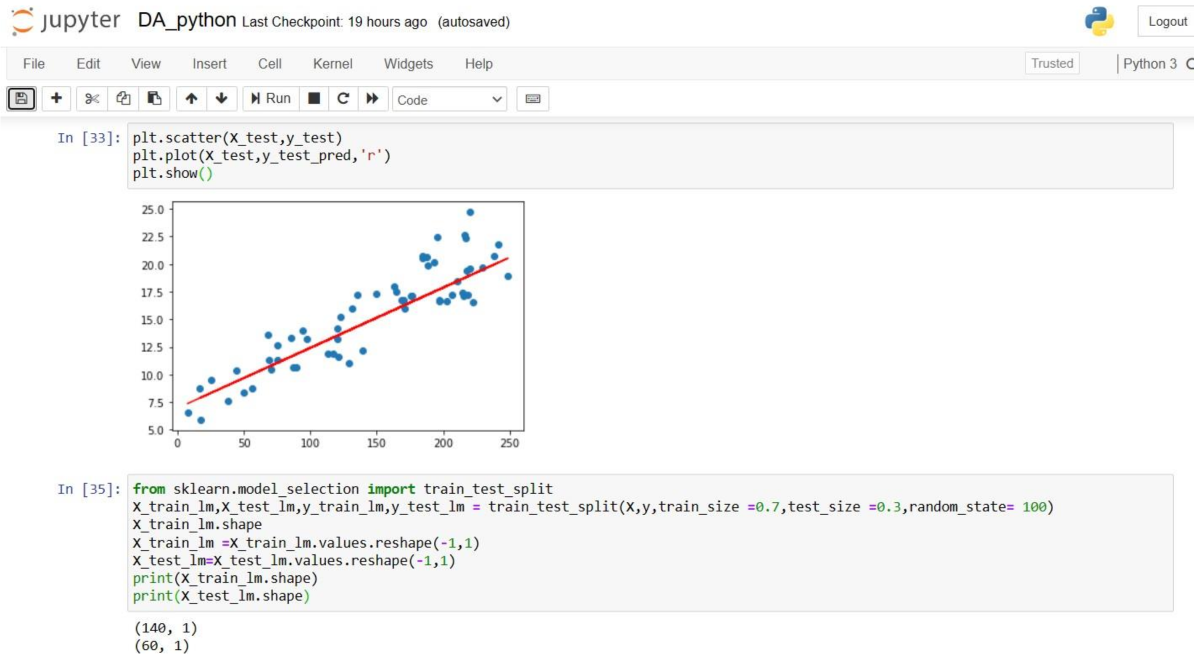

jupyter DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [31]: X_test_sm=sm.add_constant(X_test)
         y_test_pred = lr.predict(X_test_sm)
         y_test_pred

Out[31]: 126     7.374140
         104    19.941482
         99     14.323269
         92     18.823294
         111    20.132392
         167    18.228745
         116    14.541452
         96     17.726924
         52     18.752384
         69     18.774202
         164    13.341445
         124    19.466933
         182    10.014155
         154    17.192376
         125    11.705073
         196    12.086893
         194    15.114182
         177    16.232370
         163    15.866914
         31     13.106899
         11     18.659656
         73     14.006904
         15     17.606923
         41     16.603281
         97     17.034193
         128    18.965113
         133    18.937840
         82     11.055978
         139    17.034193
         123    13.663265
         83     10.679613
         65     10.712340
```


jupyter
DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

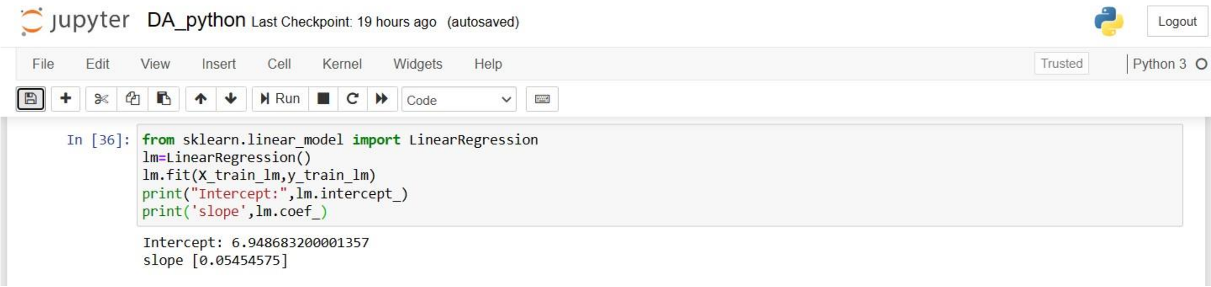
```
151     13.548719
162     17.225103
170      9.675971
77      13.521446
32      12.250530
173     16.134188
174     19.079659
85      17.486923
168     18.697838
112     16.532372
171     15.921460
181     18.866930
7       13.505083
46      11.841437
75       7.870506
28      20.519667
29      10.799613
195      9.032331
40      17.994198
153     16.292371
115     11.045069
64      14.099631
59      18.441473
1        9.375969
192      7.886870
136      8.345054
152     17.726924
161     11.623254
dtype: float64
```

```
In [32]: from sklearn.metrics import r2_score
         r_squared = r2_score(y_test,y_test_pred)
         r_squared

Out[32]: 0.7921031601245658
```

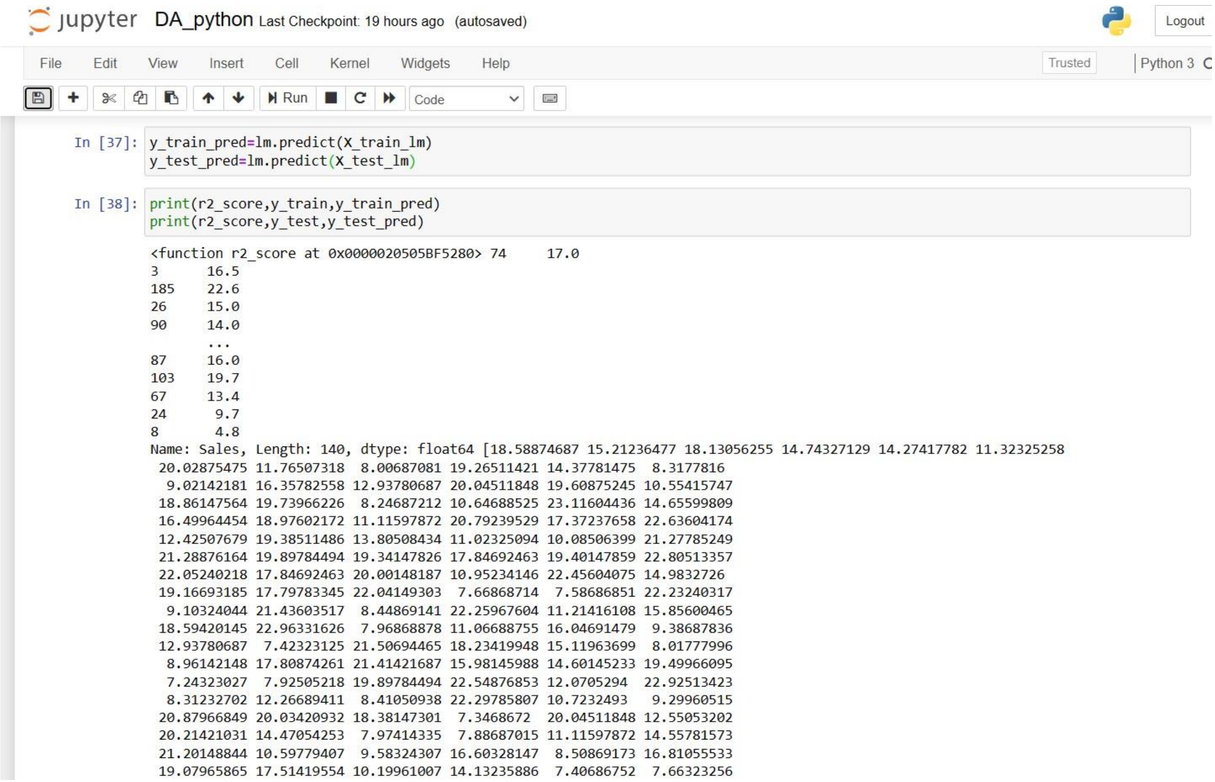

jupyter
DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
0
50
100
150
200
250

```
In [33]: plt.scatter(X_test,y_test)
         plt.plot(X_test,y_test_pred,'r')
         plt.show()
```

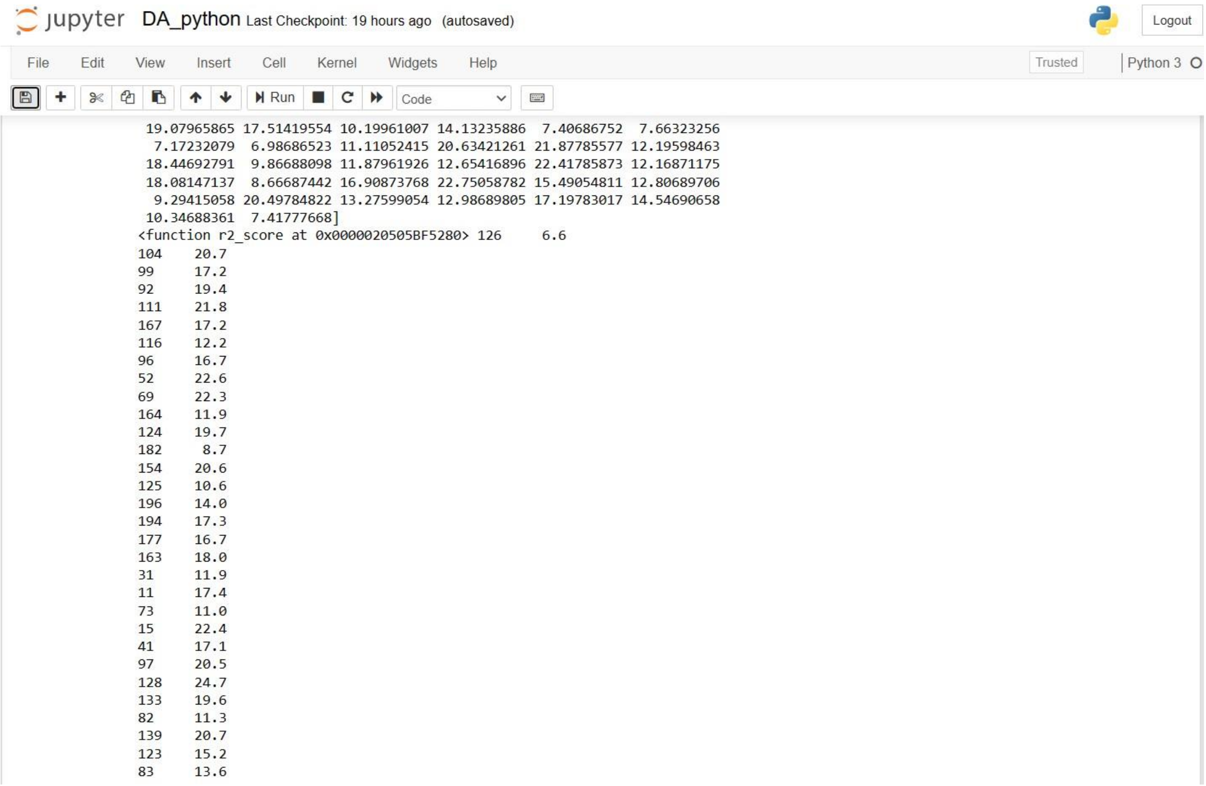
```
In [35]: from sklearn.model_selection import train_test_split
         X_train_lm,X_test_lm,y_train_lm,y_test_lm = train_test_split(X,y,train_size =0.7,test_size =0.3,random_state= 100)
         X_train_lm.shape
         X_train_lm =X_train_lm.values.reshape(-1,1)
         X_test_lm=X_test_lm.values.reshape(-1,1)
         print(X_train_lm.shape)
         print(X_test_lm.shape)

         (140, 1)
         (60, 1)
```


jupyter
DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

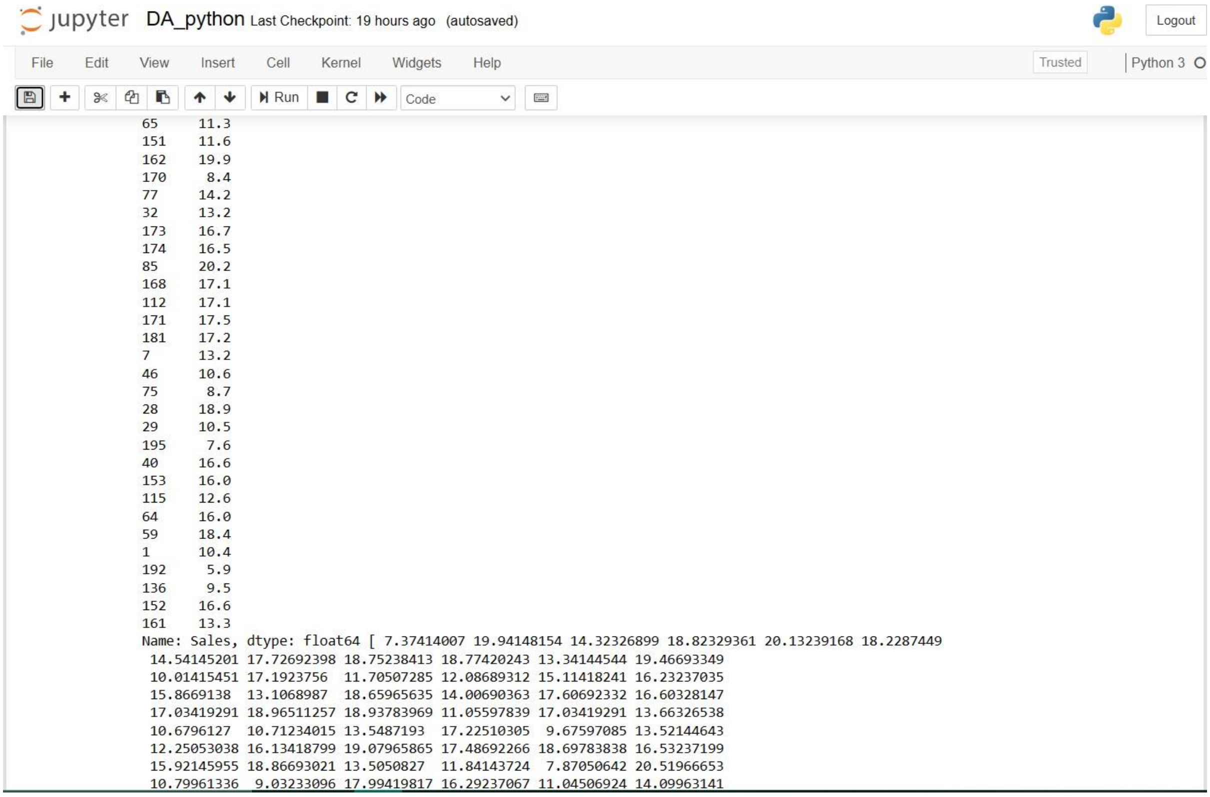
```
In [36]: from sklearn.linear_model import LinearRegression
         lm=LinearRegression()
         lm.fit(X_train_lm,y_train_lm)
         print("Intercept:",lm.intercept_)
         print('slope',lm.coef_)

         Intercept: 6.948683200001357
         slope [0.05454575]
```


jupyter DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

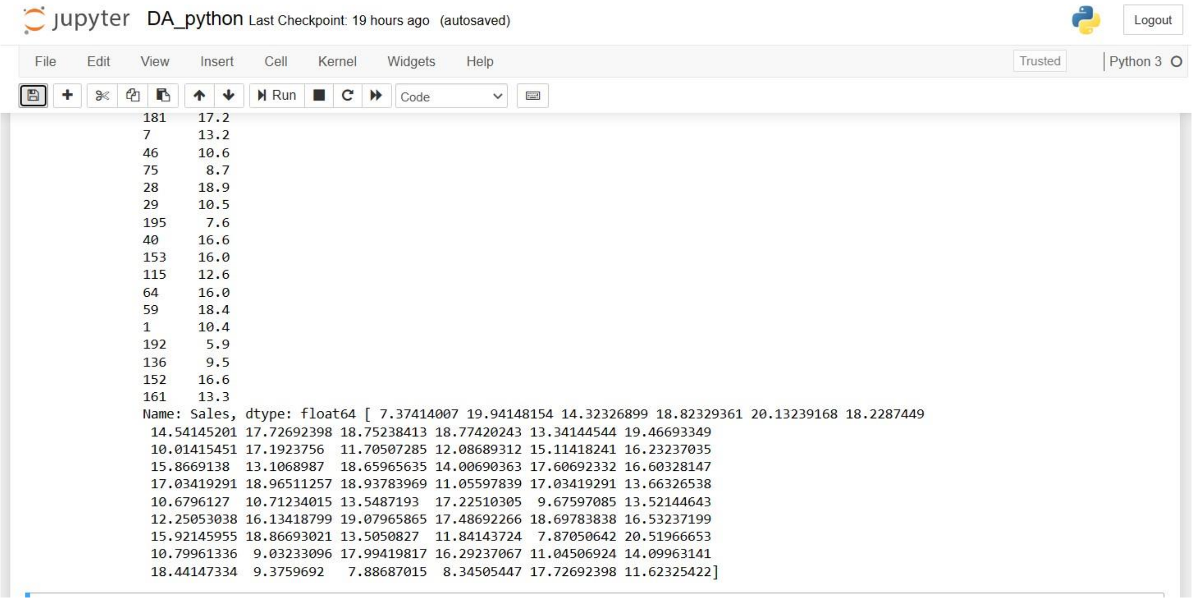
```
In [37]: y_train_pred=lm.predict(X_train_lm)
         y_test_pred=lm.predict(X_test_lm)

In [38]: print(r2_score,y_train,y_train_pred)
         print(r2_score,y_test,y_test_pred)

<function r2_score at 0x0000020505BF5280> 74     17.0
3      16.5
185    22.6
26     15.0
90     14.0
       ...
87     16.0
103    19.7
67     13.4
24      9.7
8       4.8
Name: Sales, Length: 140, dtype: float64 [18.58874687 15.21236477 18.13056255 14.74327129 14.27417782 11.32325258
 20.02875475 11.76507318  8.00687081 19.26511421 14.37781475  8.3177816
  9.02142181 16.35782558 12.93780687 20.04511848 19.60875245 10.55415747
 18.86147564 19.73966226  8.24687212 10.64688525 23.11604436 14.65599809
 16.49964454 18.97602172 11.11597872 20.79239529 17.37237658 22.63604174
 12.42507679 19.38511486 13.80508434 11.02325094 10.08506399 21.27785249
 21.28876164 19.89784494 19.34147826 17.84692463 19.40147859 22.80513357
 22.05240218 17.84692463 20.00148187 10.95234146 22.45604075 14.9832726
 19.16693185 17.79783345 22.04149303  7.66868714  7.58686851 22.23240317
  9.10324044 21.43603517  8.44869141 22.25967604 11.21416108 15.85600465
 18.59420145 22.96331626  7.96868878 11.06688755 16.04691479  9.38687836
 12.93780687  7.42323125 21.50694465 18.23419948 15.11963699  8.01777996
  8.96142148 17.80874261 21.41421687 15.98145988 14.60145233 19.49966095
  7.24323027  7.92505218 19.89784494 22.54876853 12.0705294  22.92513423
  8.31232702 12.26689411  8.41050938 22.29785807 10.7232493   9.29960515
 20.87966849 20.03420932 18.38147301  7.3468672  20.04511848 12.55053202
 20.21421031 14.47054253  7.97414335  7.88687015 11.11597872 14.55781573
 21.20148844 10.59779407  9.58324307 16.60328147  8.50869173 16.81055533
 19.07965865 17.51419554 10.19961007 14.13235886  7.40686752  7.66323256
```


jupyter DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
 19.07965865 17.51419554 10.19961007 14.13235886  7.40686752  7.66323256
  7.17232079  6.98686523 11.11052415 20.63421261 21.87785577 12.19598463
 18.44692791  9.86688098 11.87961926 12.65416896 22.41785873 12.16871175
 18.08147137  8.66687442 16.90873768 22.75058782 15.49054811 12.80689706
  9.29415058 20.49784822 13.27599054 12.98689805 17.19783017 14.54690658
 10.34688361  7.41777668]
<function r2_score at 0x0000020505BF5280> 126     6.6
104    20.7
99     17.2
92     19.4
111    21.8
167    17.2
116    12.2
96     16.7
52     22.6
69     22.3
164    11.9
124    19.7
182     8.7
154    20.6
125    10.6
196    14.0
194    17.3
177    16.7
163    18.0
31     11.9
11     17.4
73     11.0
15     22.4
41     17.1
97     20.5
128    24.7
133    19.6
82     11.3
139    20.7
123    15.2
83     13.6
```


jupyter DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

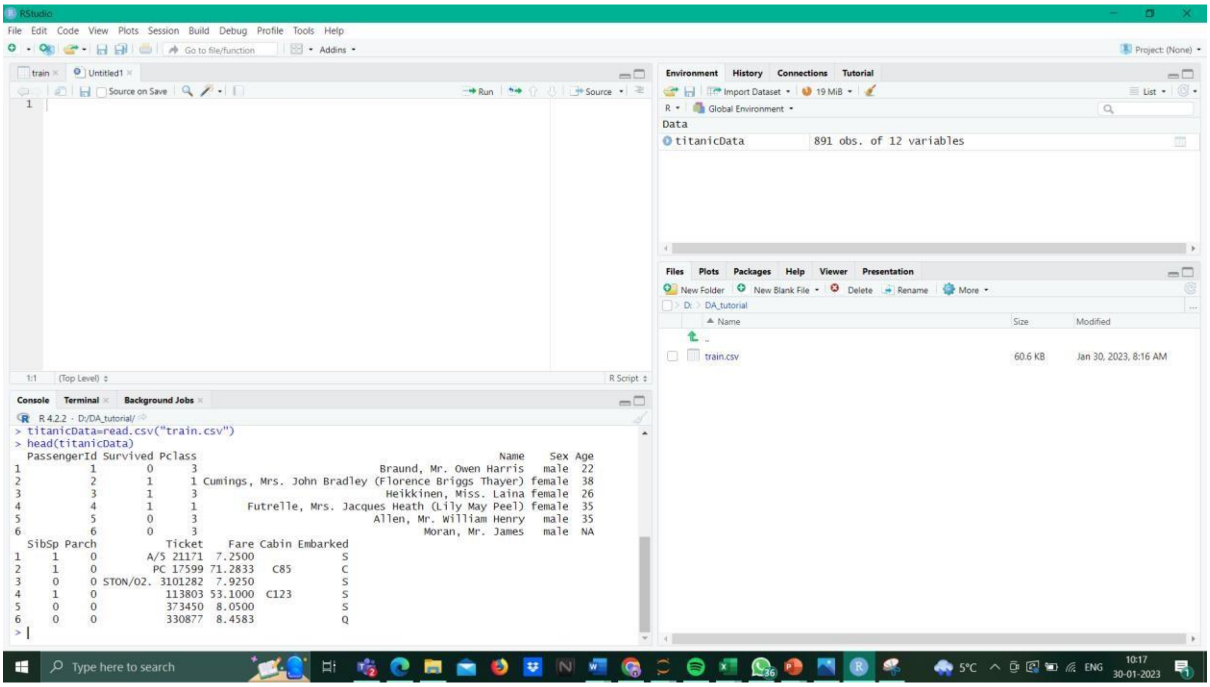
```
65      11.3
151     11.6
162     19.9
170      8.4
77      14.2
32      13.2
173     16.7
174     16.5
85      20.2
168     17.1
112     17.1
171     17.5
181     17.2
7       13.2
46      10.6
75       8.7
28      18.9
29      10.5
195      7.6
40      16.6
153     16.0
115     12.6
64      16.0
59      18.4
1       10.4
192      5.9
136      9.5
152     16.6
161     13.3
Name: Sales, dtype: float64 [ 7.37414007 19.94148154 14.32326899 18.82329361 20.13239168 18.2287449
 14.54145201 17.72692398 18.75238413 18.77420243 13.34144544 19.46693349
 10.01415451 17.1923756  11.70507285 12.08689312 15.11418241 16.23237035
 15.8669138  13.1068987  18.65965635 14.00690363 17.60692332 16.60328147
 17.03419291 18.96511257 18.93783969 11.05597839 17.03419291 13.66326538
 10.6796127  10.71234015 13.5487193  17.22510305  9.67597085 13.52144643
 12.25053038 16.13418799 19.07965865 17.48692266 18.69783838 16.53237199
 15.92145955 18.86693021 13.5050827  11.84143724  7.87050642 20.51966653
 10.79961336  9.03233096 17.99419817 16.29237067 11.04506924 14.09963141
```


jupyter DA_python Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

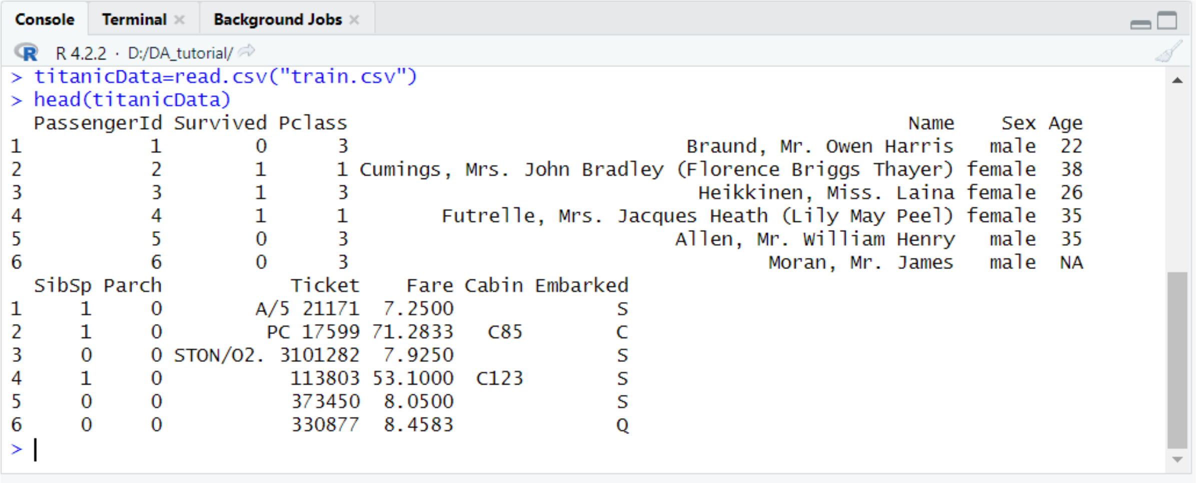
```
181     17.2
7       13.2
46      10.6
75       8.7
28      18.9
29      10.5
195      7.6
40      16.6
153     16.0
115     12.6
64      16.0
59      18.4
1       10.4
192      5.9
136      9.5
152     16.6
161     13.3
Name: Sales, dtype: float64 [ 7.37414007 19.94148154 14.32326899 18.82329361 20.13239168 18.2287449
 14.54145201 17.72692398 18.75238413 18.77420243 13.34144544 19.46693349
 10.01415451 17.1923756  11.70507285 12.08689312 15.11418241 16.23237035
 15.8669138  13.1068987  18.65965635 14.00690363 17.60692332 16.60328147
 17.03419291 18.96511257 18.93783969 11.05597839 17.03419291 13.66326538
 10.6796127  10.71234015 13.5487193  17.22510305  9.67597085 13.52144643
 12.25053038 16.13418799 19.07965865 17.48692266 18.69783838 16.53237199
 15.92145955 18.86693021 13.5050827  11.84143724  7.87050642 20.51966653
 10.79961336  9.03233096 17.99419817 16.29237067 11.04506924 14.09963141
```

```
 18.44147334  9.3759692   7.88687015  8.34505447 17.72692398 11.62325422]
```

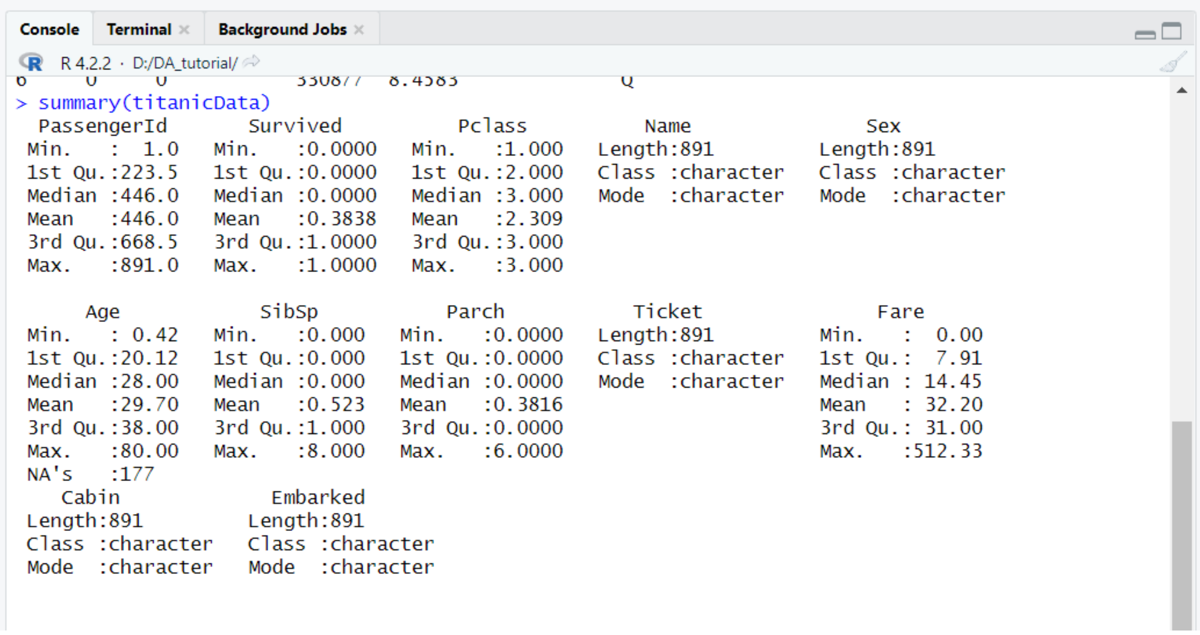
R Analysis



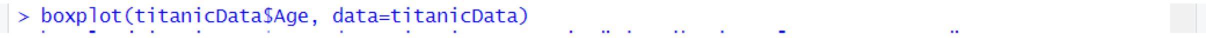
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
train
Untitled1
Source on Save
Run
Source
Environment History Connections Tutorial
Import Dataset
19 MiB
Global Environment
Data
titanicData 891 obs. of 12 variables
Files Plots Packages Help Viewer Presentation
New Folder
New Blank File
Delete
Rename
More
D: DA_tutorial
Name Size Modified
train.csv 60.6 KB Jan 30, 2023, 8:16 AM
Console Terminal Background Jobs
R 4.2.2 · D:/DA_tutorial/
Type here to search
5°C ENG 10:17 30-01-2023



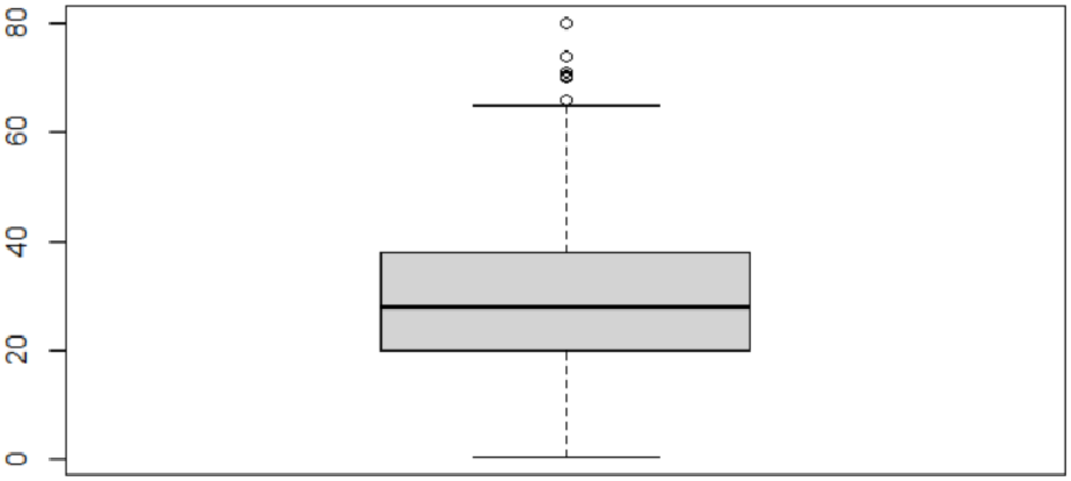
```
> titanicData=read.csv("train.csv")
> head(titanicData)
  PassengerId Survived Pclass                                                Name    Sex Age
1           1        0      3                             Braund, Mr. Owen Harris   male  22
2           2        1      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38
3           3        1      3                              Heikkinen, Miss. Laina female  26
4           4        1      1        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35
5           5        0      3                            Allen, Mr. William Henry   male  35
6           6        0      3                                    Moran, Mr. James   male  NA
  SibSp Parch           Ticket    Fare Cabin Embarked
1     1     0        A/5 21171  7.2500              S
2     1     0         PC 17599 71.2833   C85        C
3     0     0 STON/O2. 3101282  7.9250              S
4     1     0           113803 53.1000  C123        S
5     0     0           373450  8.0500              S
6     0     0           330877  8.4583              Q
> |
```

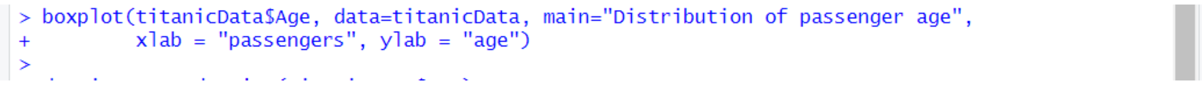


```
6      0      0           330877   8.4583             Q
> summary(titanicData)
  PassengerId       Survived          Pclass          Name               Sex           
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891         Length:891        
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character  
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character   Mode  :character  
 Mean   :446.0   Mean   :0.3838   Mean   :2.309                                        
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000                                        
 Max.   :891.0   Max.   :1.0000   Max.   :3.000                                        
                                                                                       
      Age            SibSp           Parch            Ticket               Fare       
 Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891         Min.   :  0.00  
 1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character   1st Qu.:  7.91  
 Median :28.00   Median :0.000   Median :0.0000   Mode  :character   Median : 14.45  
 Mean   :29.70   Mean   :0.523   Mean   :0.3816                      Mean   : 32.20  
 3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000                      3rd Qu.: 31.00  
 Max.   :80.00   Max.   :8.000   Max.   :6.0000                      Max.   :512.33  
 NA's   :177                                                                          
    Cabin             Embarked        
 Length:891         Length:891        
 Class :character   Class :character  
 Mode  :character   Mode  :character  
```

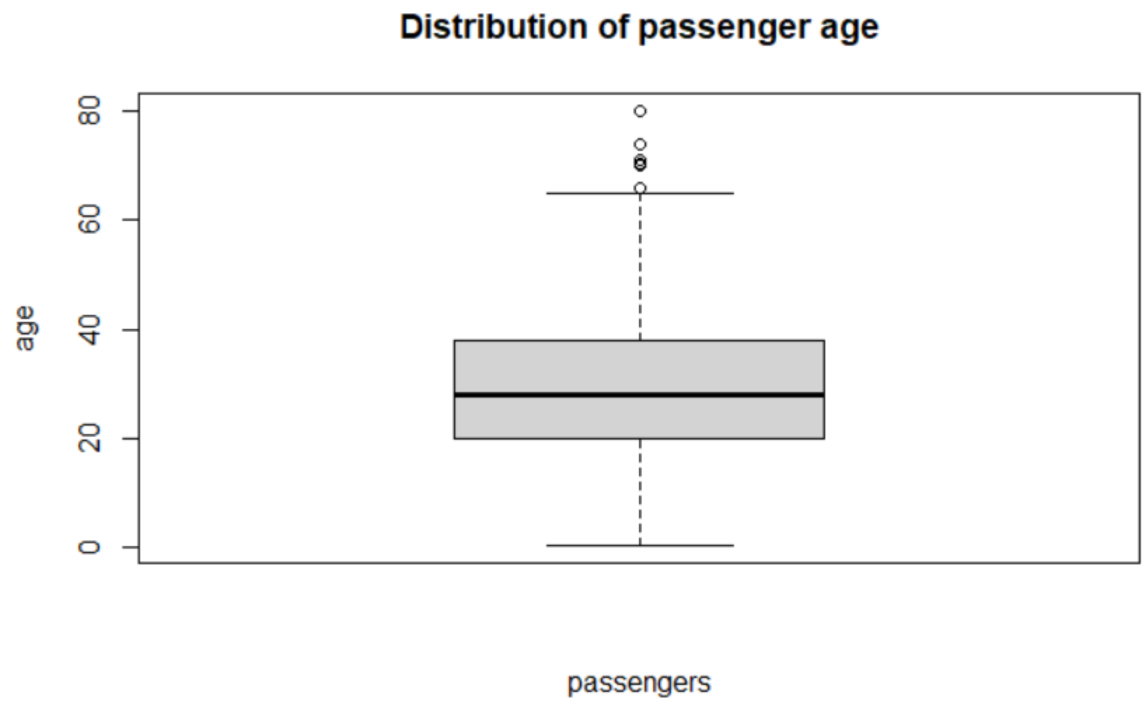


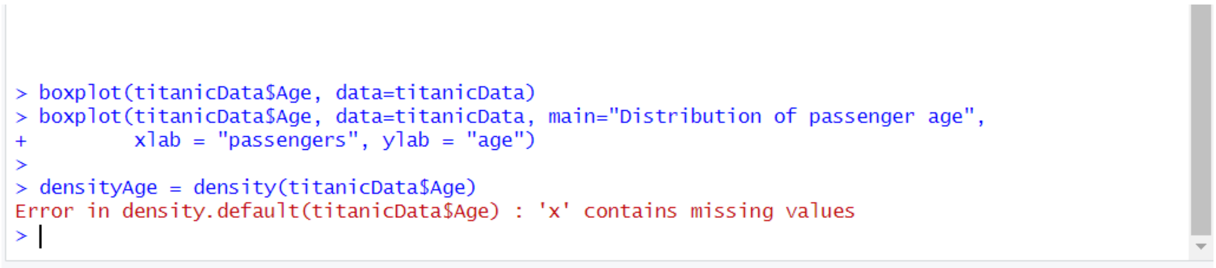
```
> boxplot(titanicData$Age, data=titanicData)
```



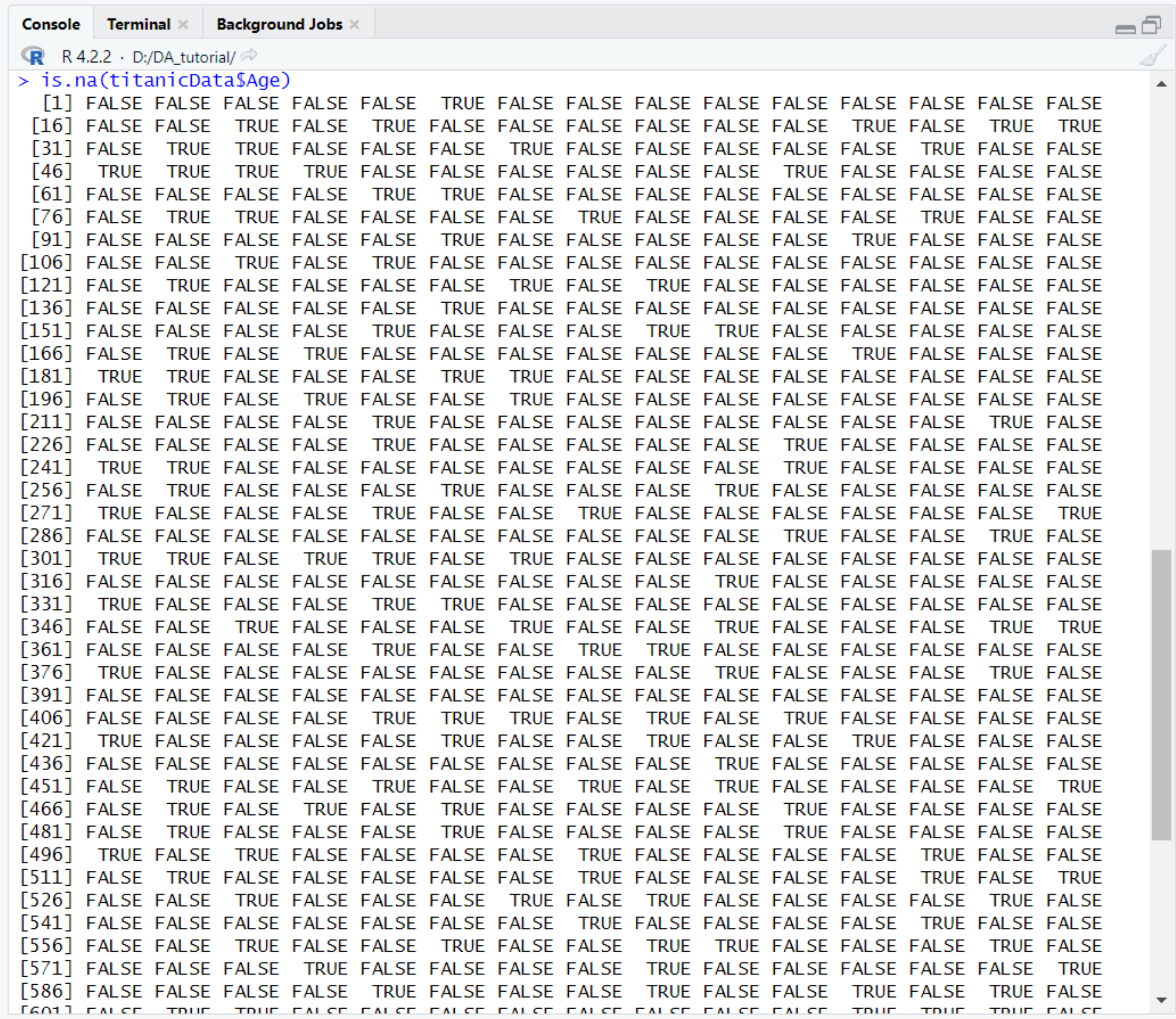


```
> boxplot(titanicData$Age, data=titanicData, main="Distribution of passenger age",
+         xlab = "passengers", ylab = "age")
>
```

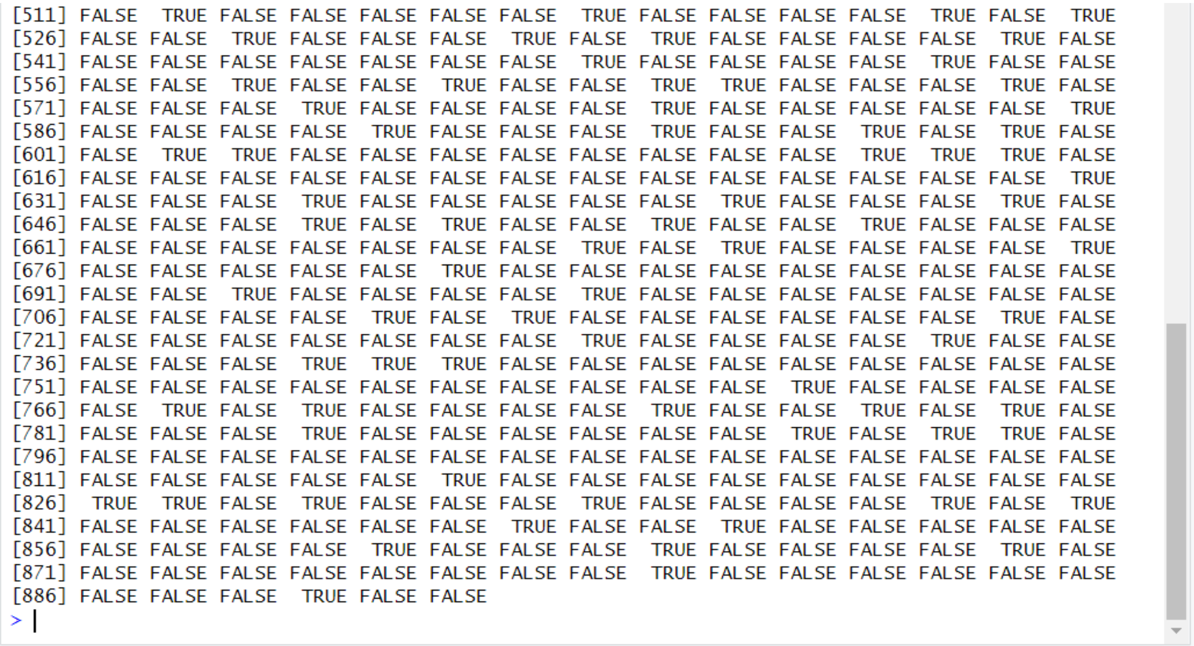




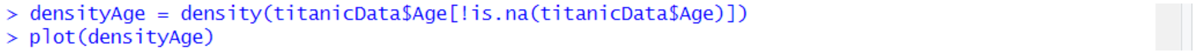
```
> boxplot(titanicData$Age, data=titanicData)
> boxplot(titanicData$Age, data=titanicData, main="Distribution of passenger age",
+         xlab = "passengers", ylab = "age")
>
> densityAge = density(titanicData$Age)
Error in density.default(titanicData$Age) : 'x' contains missing values
> |
```



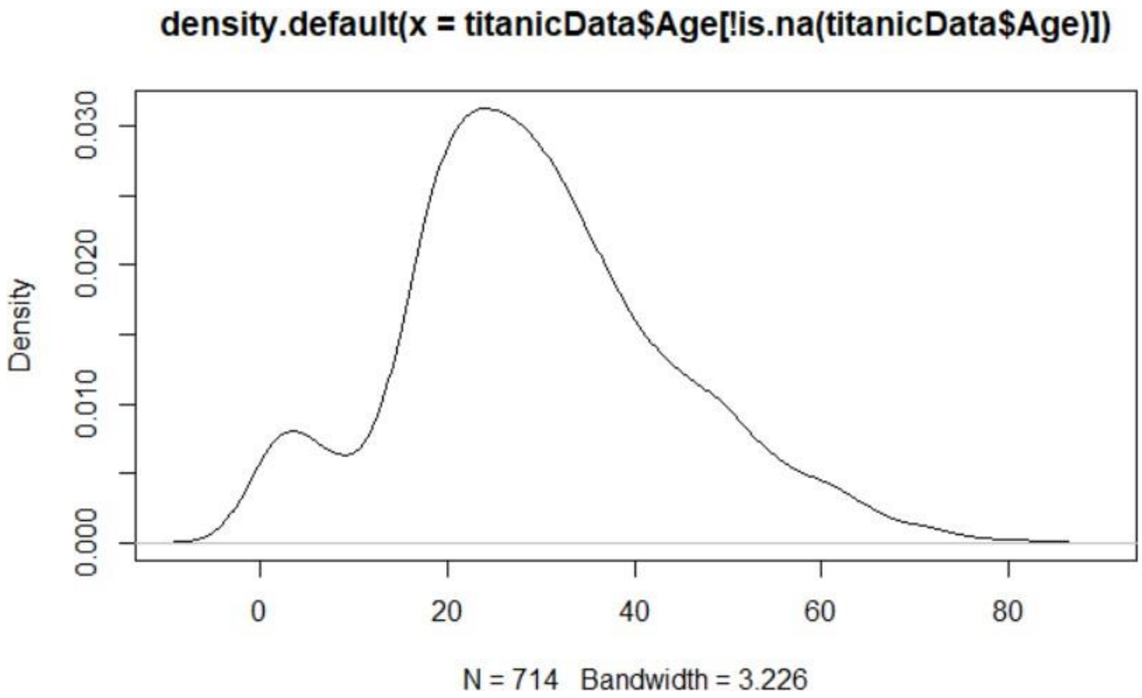
```
> is.na(titanicData$Age)
  [1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [16] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE
 [31] FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
 [46]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
 [61] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [76] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
 [91] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
[106] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[121] FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[136] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[151] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
[166] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
[181]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[196] FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[211] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
[226] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
[241]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
[256] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
[271]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
[286] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
[301]  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[316] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
[331]  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[346] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
[361] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[376]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
[391] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[406] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
[421]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
[436] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
[451] FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
[466] FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
[481] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
[496]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
[511] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
[526] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
[541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
[556] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
[571] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
[586] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
[601] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
```



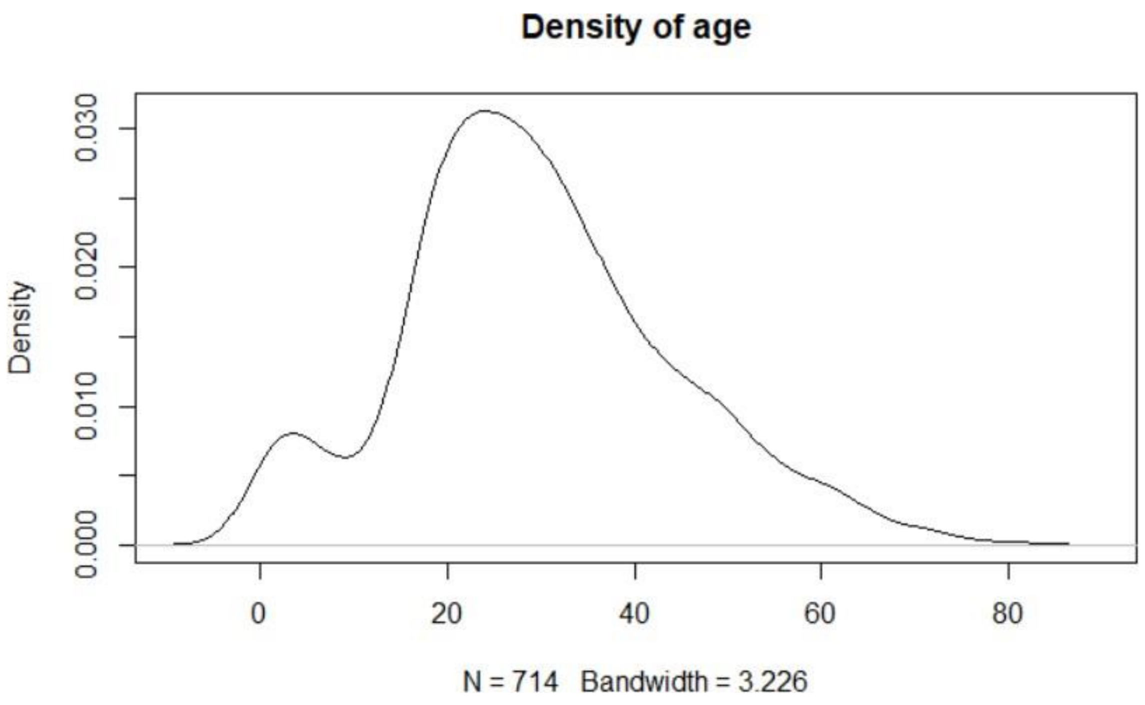
```
[511] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
[526] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
[541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
[556] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
[571] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
[586] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
[601] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
[616] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
[631] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
[646] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
[661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
[676] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[691] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[706] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
[721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
[736] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[751] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
[766] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
[781] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
[796] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[811] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[826]  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
[841] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
[856] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
[871] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[886] FALSE FALSE FALSE  TRUE FALSE FALSE
> 
```

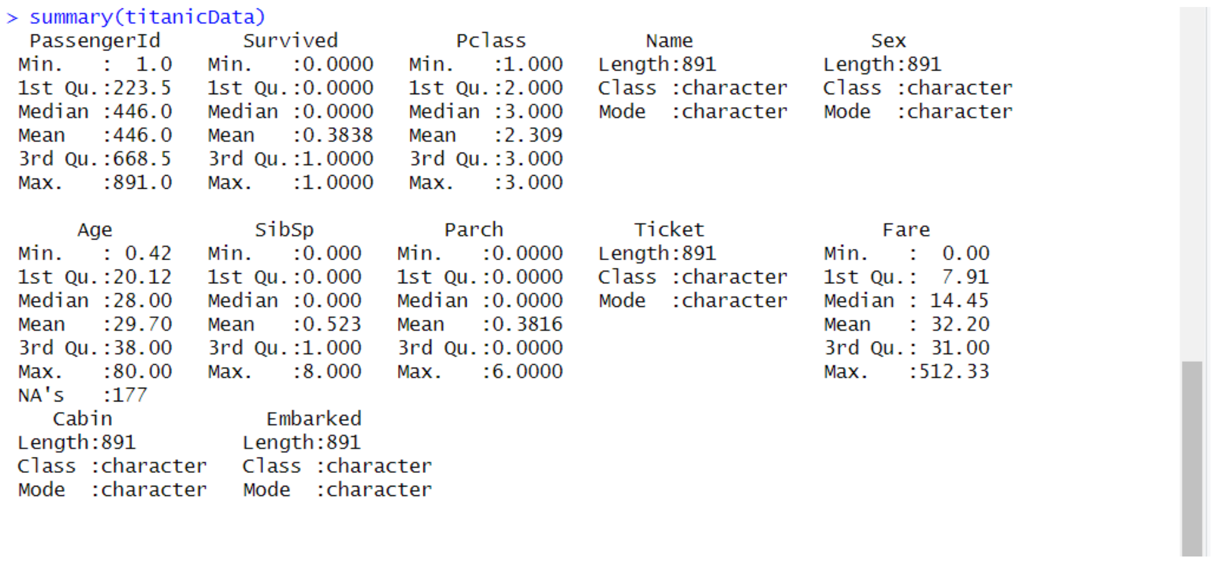


```
> densityAge = density(titanicData$Age[!is.na(titanicData$Age)])
> plot(densityAge)
```

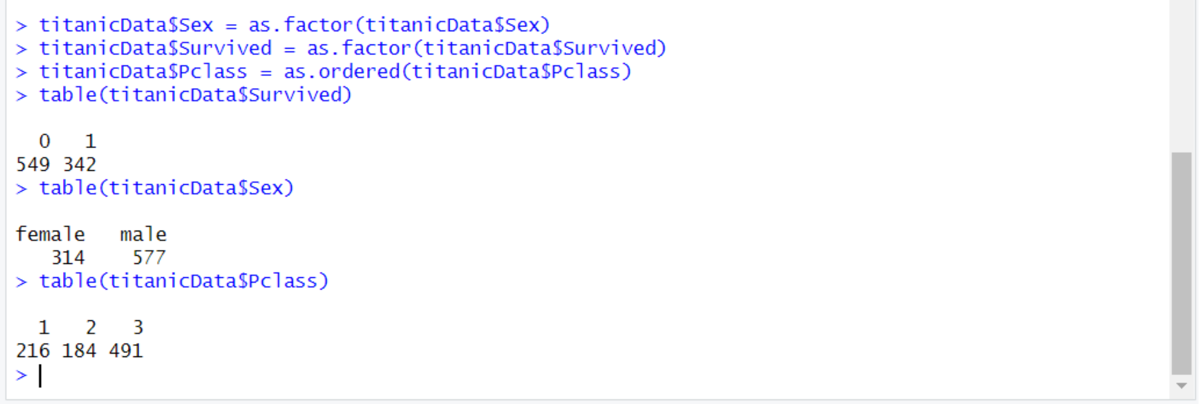


```
> plot(densityAge, main= "Density of age")
```

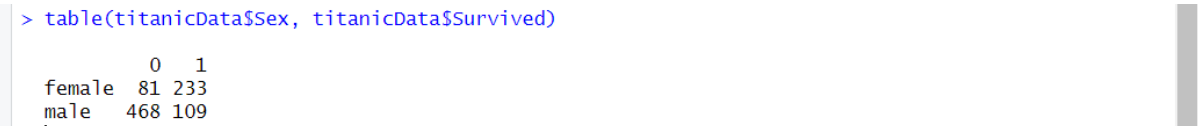




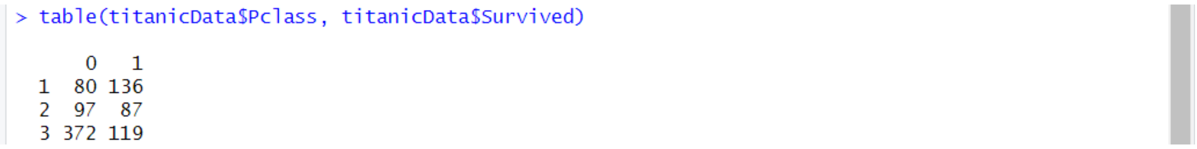
```
> summary(titanicData)
  PassengerId       Survived          Pclass          Name               Sex           
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891         Length:891        
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character  
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character   Mode  :character  
 Mean   :446.0   Mean   :0.3838   Mean   :2.309                                        
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000                                        
 Max.   :891.0   Max.   :1.0000   Max.   :3.000                                        
                                                                                       
      Age            SibSp           Parch           Ticket               Fare       
 Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891         Min.   :  0.00  
 1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character   1st Qu.:  7.91  
 Median :28.00   Median :0.000   Median :0.0000   Mode  :character   Median : 14.45  
 Mean   :29.70   Mean   :0.523   Mean   :0.3816                      Mean   : 32.20  
 3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000                      3rd Qu.: 31.00  
 Max.   :80.00   Max.   :8.000   Max.   :6.0000                      Max.   :512.33  
 NA's   :177                                                                         
    Cabin             Embarked        
 Length:891         Length:891        
 Class :character   Class :character  
 Mode  :character   Mode  :character  
```



```
> titanicData$Sex = as.factor(titanicData$Sex)
> titanicData$Survived = as.factor(titanicData$Survived)
> titanicData$Pclass = as.ordered(titanicData$Pclass)
> table(titanicData$Survived)

  0   1 
549 342 
> table(titanicData$Sex)

female   male 
   314    577 
> table(titanicData$Pclass)

  1   2   3 
216 184 491 
> 
```

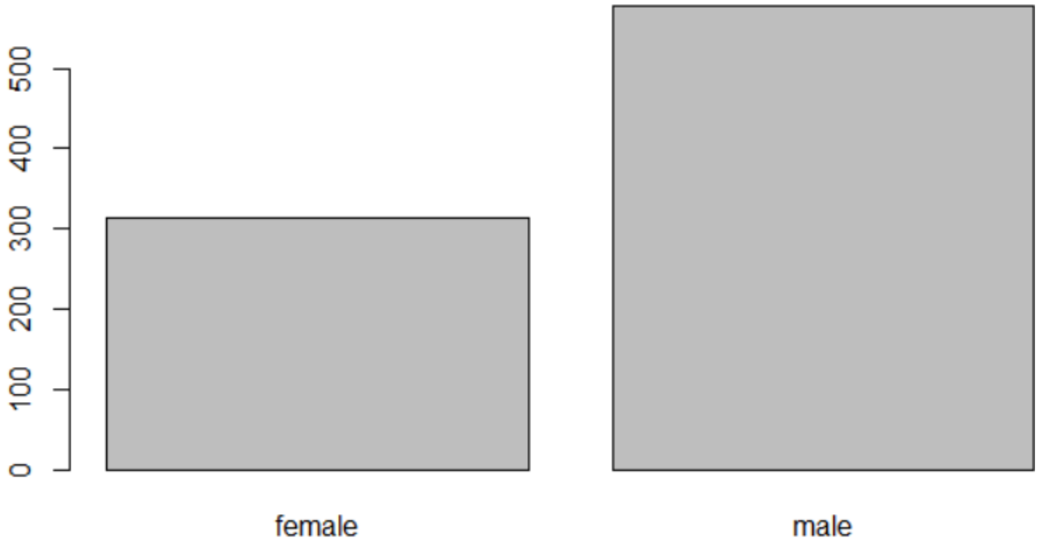


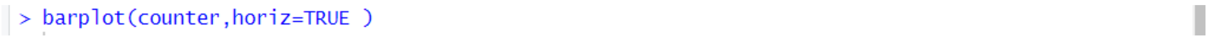
```
> table(titanicData$Sex, titanicData$Survived)
        
           0   1
  female  81 233
  male   468 109
```



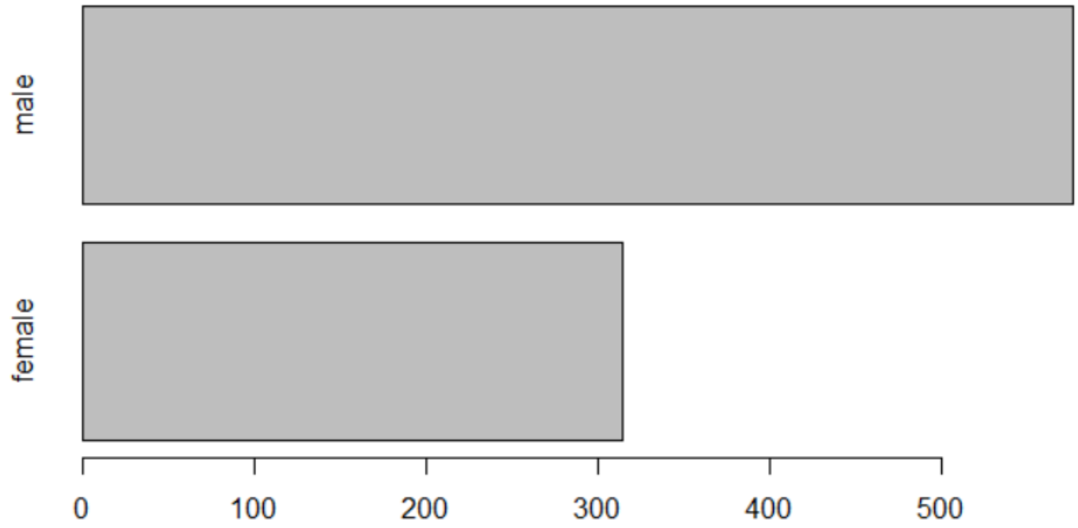
```
> table(titanicData$Pclass, titanicData$Survived)
   
      0   1
  1  80 136
  2  97  87
  3 372 119
```

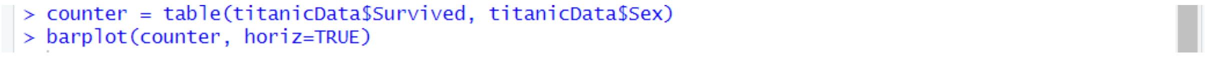
```
> barplot(counter)
```



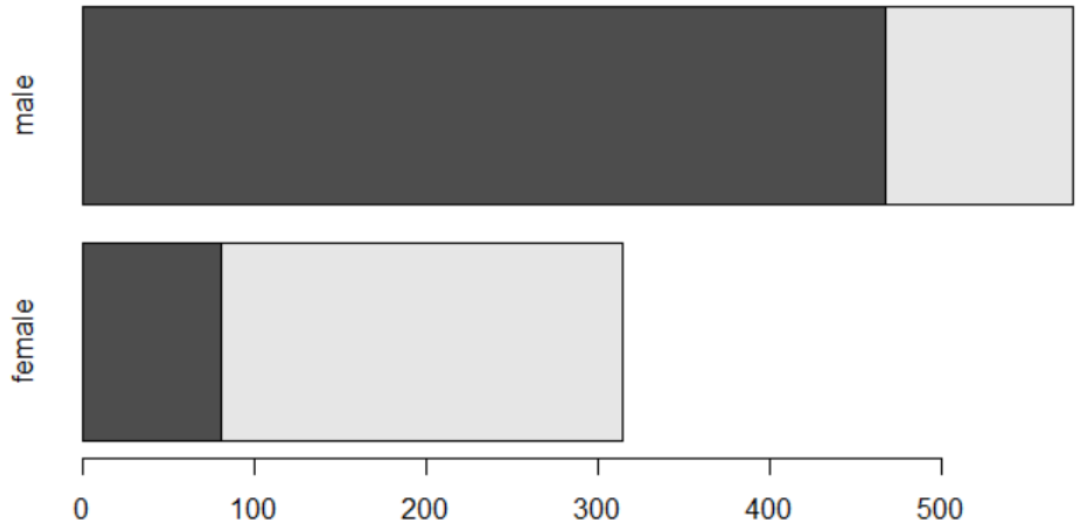


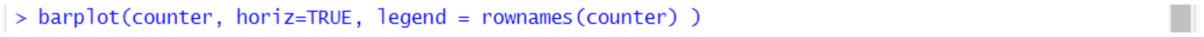
```
> barplot(counter,horiz=TRUE )
```



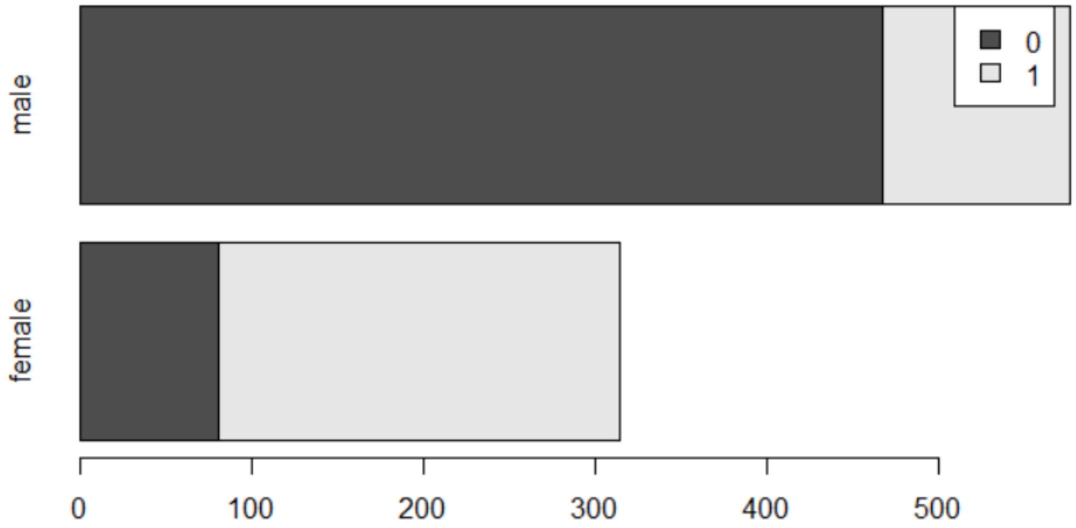


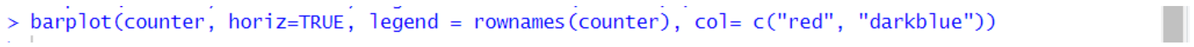
```
> counter = table(titanicData$Survived, titanicData$Sex)
> barplot(counter, horiz=TRUE)
```



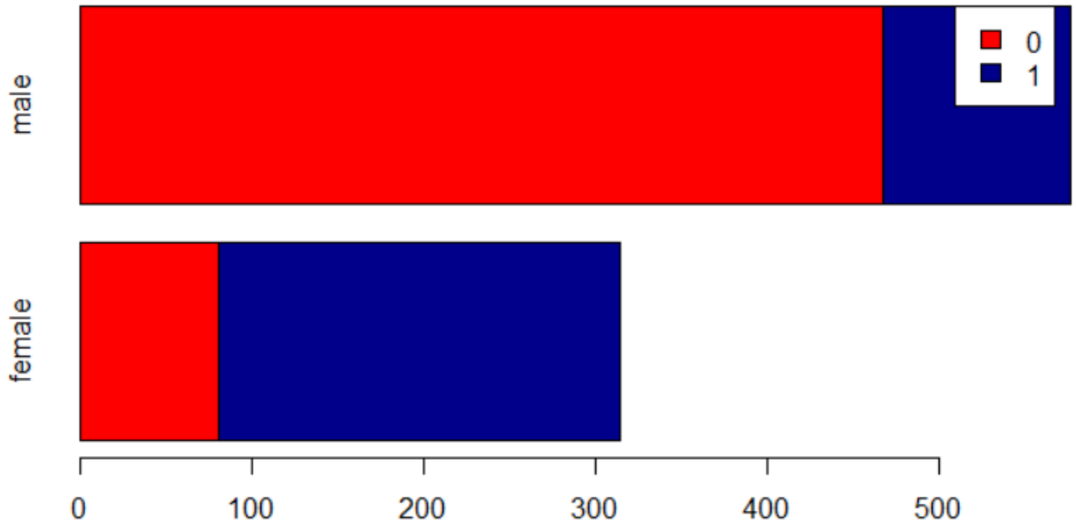


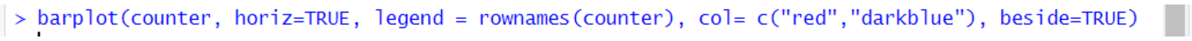
```
> barplot(counter, horiz=TRUE, legend = rownames(counter) )
```



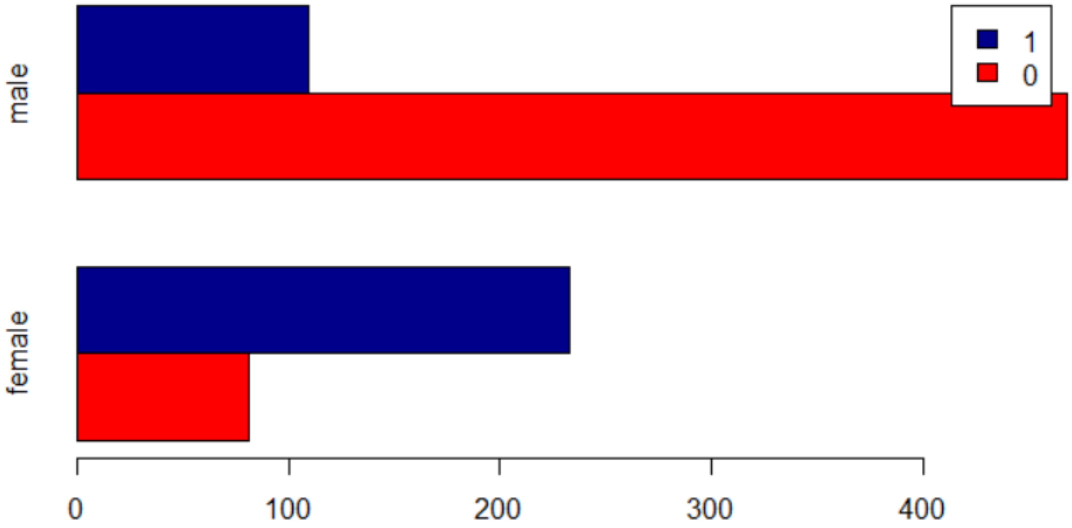


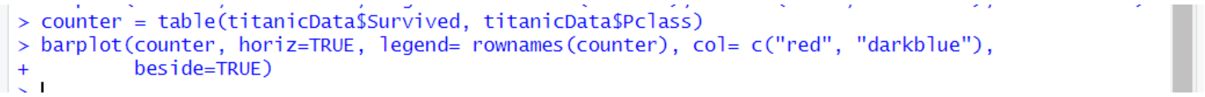
```
> barplot(counter, horiz=TRUE, legend = rownames(counter), col= c("red", "darkblue"))
```



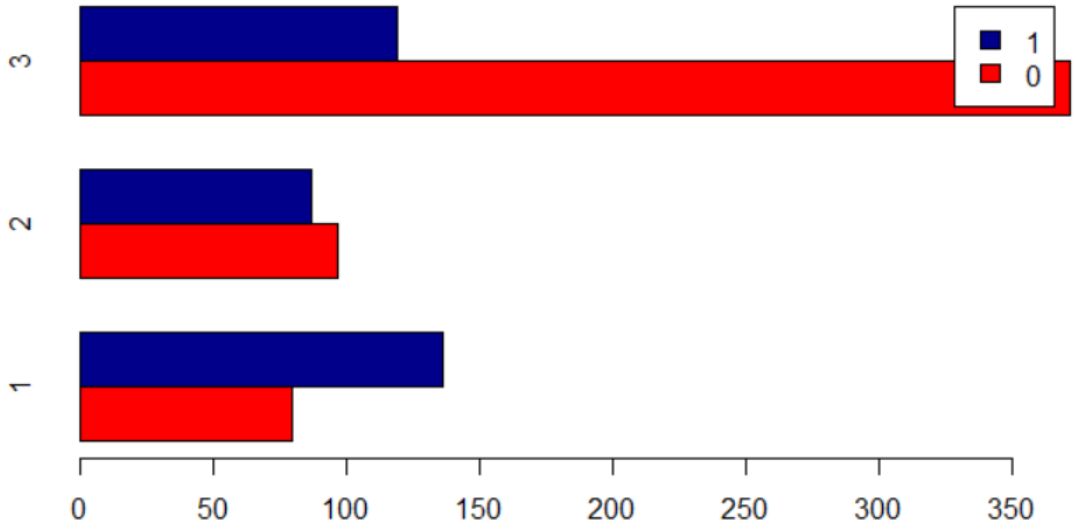


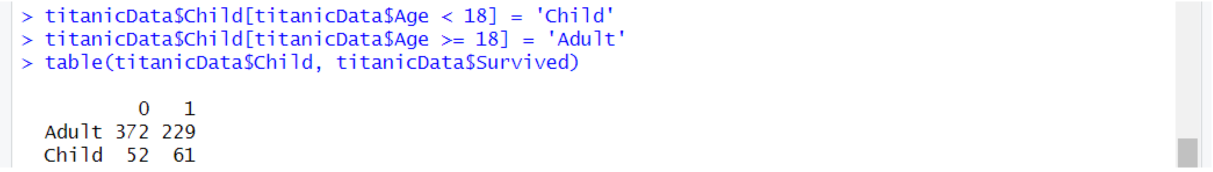
```
> barplot(counter, horiz=TRUE, legend = rownames(counter), col= c("red","darkblue"), beside=TRUE)
```



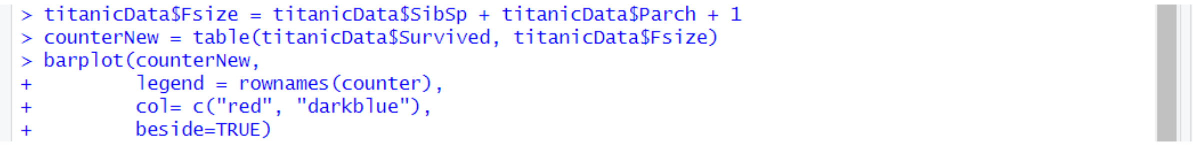


```
> counter = table(titanicData$Survived, titanicData$Pclass)
> barplot(counter, horiz=TRUE, legend= rownames(counter), col= c("red", "darkblue"),
+         beside=TRUE)
> |
```

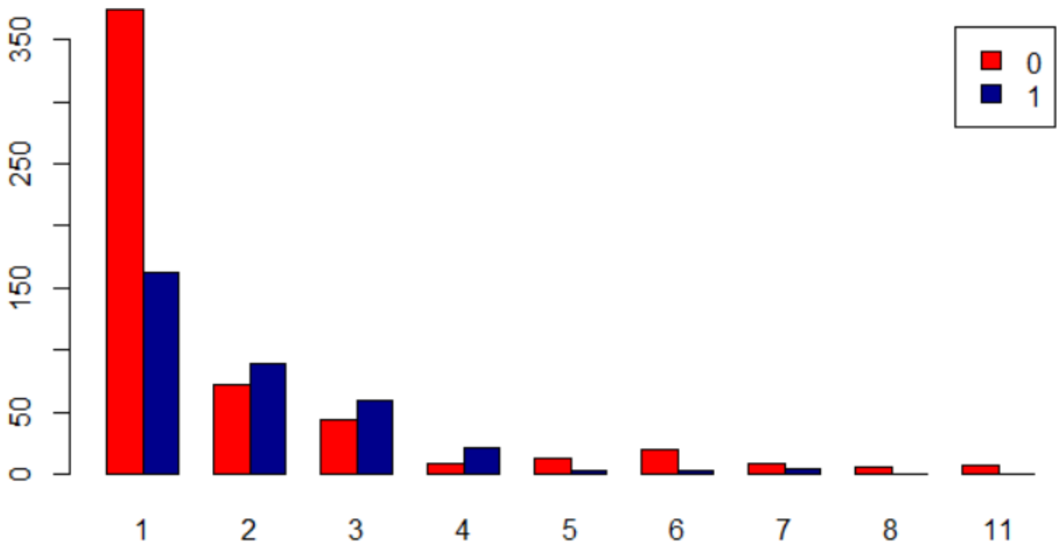




```
> titanicData$Child[titanicData$Age < 18] = 'Child'
> titanicData$Child[titanicData$Age >= 18] = 'Adult'
> table(titanicData$Child, titanicData$Survived)

        0   1
Adult 372 229
Child  52  61
```



```
> titanicData$Fsize = titanicData$SibSp + titanicData$Parch + 1
> counterNew = table(titanicData$Survived, titanicData$Fsize)
> barplot(counterNew,
+         legend = rownames(counter),
+         col= c("red", "darkblue"),
+         beside=TRUE)
```


0
50
150
250
350
1
2
3
4
5
6
7
8
11
0
1