In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```python
df=pd.read_csv("Desktop/Studies/Datasets/TaxiFare/train.csv")
```

In [3]:

```python
df.head(20)
```

Out[3]:

	trip_distance	rate_code	store_and_fwd_flag	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	total_an
0	9.01	1	N	1	26.0	0.0	0.5	8.14	5.76	0.3	
1	0.20	1	N	1	3.0	0.0	0.5	0.75	0.00	0.3	
2	9.65	1	N	1	41.5	0.0	0.5	9.61	5.76	0.3	
3	9.50	1	N	1	30.0	0.5	0.5	9.25	5.76	0.3	
4	5.80	1	N	1	21.5	0.5	0.5	4.56	0.00	0.3	
5	12.65	1	N	1	41.5	0.0	0.5	0.02	5.76	0.3	
6	8.92	1	N	1	27.0	0.0	0.5	6.71	5.76	0.3	
7	5.98	1	N	1	22.0	0.5	0.5	5.82	0.00	0.3	
8	12.30	1	N	1	38.0	0.5	0.5	9.80	0.00	0.3	
9	11.92	1	N	1	34.5	0.5	0.5	0.00	0.00	0.3	
10	14.12	1	N	1	40.0	0.0	0.5	0.00	0.00	0.3	
11	9.13	1	N	1	29.0	0.0	0.5	4.00	0.00	0.3	
12	3.50	1	N	1	25.0	0.0	0.5	5.15	0.00	0.3	
13	11.90	1	N	1	44.0	0.0	0.5	10.10	5.76	0.3	
14	12.60	1	N	1	38.5	0.5	0.5	4.00	0.00	0.3	
15	8.51	1	N	1	30.5	0.0	0.5	7.41	5.76	0.3	
16	4.25	1	N	1	22.5	1.0	0.5	4.86	0.00	0.3	
17	11.10	1	N	2	35.5	0.0	0.5	0.00	2.64	0.3	
18	7.60	1	N	1	23.5	1.0	0.5	6.30	0.00	0.3	
19	3.40	1	N	1	46.0	0.0	0.5	9.36	0.00	0.3	

In [4]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35000 entries, 0 to 34999
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   trip_distance          35000 non-null  float64
 1   rate_code              35000 non-null  int64  
 2   store_and_fwd_flag     35000 non-null  object 
 3   payment_type           35000 non-null  int64  
 4   fare_amount            35000 non-null  float64
 5   extra                  35000 non-null  float64
 6   mta_tax                35000 non-null  float64
 7   tip_amount             35000 non-null  float64
 8   tolls_amount           35000 non-null  float64
 9   imp_surcharge          35000 non-null  float64
 10  total_amount           35000 non-null  float64
 11  pickup_location_id     35000 non-null  int64  
 12  dropoff_location_id    35000 non-null  int64  
 13  year                   35000 non-null  int64  
 14  month                  35000 non-null  int64  
 15  day                    35000 non-null  int64  
 16  day_of_week            35000 non-null  int64  
 17  hour_of_day            35000 non-null  int64  
 18  trip_duration          35000 non-null  float64
 19  calculated_total_amount  35000 non-null  float64
dtypes: float64(10), int64(9), object(1)
memory usage: 5.3+ MB
```

In [5]:

```python
df.describe()
```

Out[5]:

	trip_distance	rate_code	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	total_

count	35000.000000	35000.000000	35000.000000	35000.000000	35000.000000	35000.000000	35000.000000	35000.000000	35000.000000	3500
mean	9.088815	1.110086	1.123400	31.920911	0.320337	0.486929	6.142190	2.262871	0.299940	4
std	4.496854	0.581456	0.350842	14.689516	0.402590	0.079781	4.397599	3.578315	0.004242	1
min	0.010000	1.000000	1.000000	0.010000	0.000000	0.000000	0.000000	0.000000	0.000000	0
25%	6.470000	1.000000	1.000000	24.000000	0.000000	0.500000	4.460000	0.000000	0.300000	3
50%	8.700000	1.000000	1.000000	29.000000	0.000000	0.500000	6.050000	0.000000	0.300000	3
75%	10.990000	1.000000	1.000000	36.000000	0.500000	0.500000	8.100000	5.760000	0.300000	4
max	79.010000	5.000000	4.000000	400.000000	18.500000	0.500000	215.000000	189.500000	0.300000	46

In [6]:

```
#df.drop(['store_and_fwd_flag'],axis=1)
```

In [7]:

```
df['rate_code'].unique()
```

Out[7]:

```
array([1, 3, 5, 4, 2], dtype=int64)
```

In [8]:

```
df['payment_type'].unique()
```

Out[8]:

```
array([1, 2, 3, 4], dtype=int64)
```

In [9]:

```
df['store_and_fwd_flag'].unique()
```

Out[9]:

```
array(['N', 'Y'], dtype=object)
```

In [10]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
le.fit_transform(df['store_and_fwd_flag'])
```

Out[10]:

```
array([0, 0, 0, ..., 0, 0, 0])
```

In [11]:

```
df1=pd.get_dummies(df['store_and_fwd_flag'])
df1
```

Out[11]:

	N	Y
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
...	...	...
34995	1	0
34996	1	0
34997	1	0
34998	1	0
34999	1	0

35000 rows × 2 columns

In [12]:

```
df=df.drop(['store_and_fwd_flag'],axis=1)
df
```

Out[12]:

	trip_distance	rate_code	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	total_amount	pickup_loc
0	9.01	1	1	26.0	0.0	0.5	8.14	5.76	0.3	40.70	
1	0.20	1	1	3.0	0.0	0.5	0.75	0.00	0.3	4.55	

2	9.65	1	1	41.5	0.0	0.5	9.61	5.76	0.3	57.67
3	9.50	1	1	30.0	0.5	0.5	9.25	5.76	0.3	46.31
4	5.80	1	1	21.5	0.5	0.5	4.56	0.00	0.3	27.36
...	...	...	...	...	...	...	...	...	...	...
34995	22.43	1	1	59.5	0.5	0.5	10.00	5.76	0.3	76.56
34996	9.16	1	1	30.0	0.0	0.5	6.58	5.76	0.3	43.14
34997	6.78	1	1	23.0	0.0	0.5	5.95	0.00	0.3	29.75
34998	0.26	1	2	3.0	0.0	0.5	0.00	0.00	0.3	3.80
34999	18.40	1	1	53.0	1.0	0.5	10.96	0.00	0.3	65.76

35000 rows × 19 columns

In [13]:

```
df=pd.concat([df,df1],axis=1)
df
```

Out[13]:

	trip_distance	rate_code	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	total_amount	...	dropof
0	9.01	1	1	26.0	0.0	0.5	8.14	5.76	0.3	40.70	...	
1	0.20	1	1	3.0	0.0	0.5	0.75	0.00	0.3	4.55	...	
2	9.65	1	1	41.5	0.0	0.5	9.61	5.76	0.3	57.67	...	
3	9.50	1	1	30.0	0.5	0.5	9.25	5.76	0.3	46.31	...	
4	5.80	1	1	21.5	0.5	0.5	4.56	0.00	0.3	27.36	...	
...	...	...	...	...	...	...	...	...	...	...	...	
34995	22.43	1	1	59.5	0.5	0.5	10.00	5.76	0.3	76.56	...	
34996	9.16	1	1	30.0	0.0	0.5	6.58	5.76	0.3	43.14	...	
34997	6.78	1	1	23.0	0.0	0.5	5.95	0.00	0.3	29.75	...	
34998	0.26	1	2	3.0	0.0	0.5	0.00	0.00	0.3	3.80	...	
34999	18.40	1	1	53.0	1.0	0.5	10.96	0.00	0.3	65.76	...	

35000 rows × 21 columns

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
trip_distance            0
rate_code                0
payment_type             0
fare_amount              0
extra                    0
mta_tax                  0
tip_amount               0
tolls_amount             0
imp_surcharge            0
total_amount             0
pickup_location_id       0
dropoff_location_id      0
year                     0
month                    0
day                      0
day_of_week              0
hour_of_day              0
trip_duration            0
calculated_total_amount  0
N                        0
Y                        0
dtype: int64
```

In [15]:

```
df.drop(['year'],axis=1,inplace=True)
```

In [16]:

```
df.corr()
```

Out[16]:

	trip_distance	rate_code	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	tc
trip_distance	1.000000	0.273355	-0.060372	0.829692	-0.016118	-0.270702	0.455714	0.432261	-0.000183	
rate_code	0.273355	1.000000	-0.002726	0.543670	-0.080895	-0.692998	0.268014	0.318800	-0.049447	

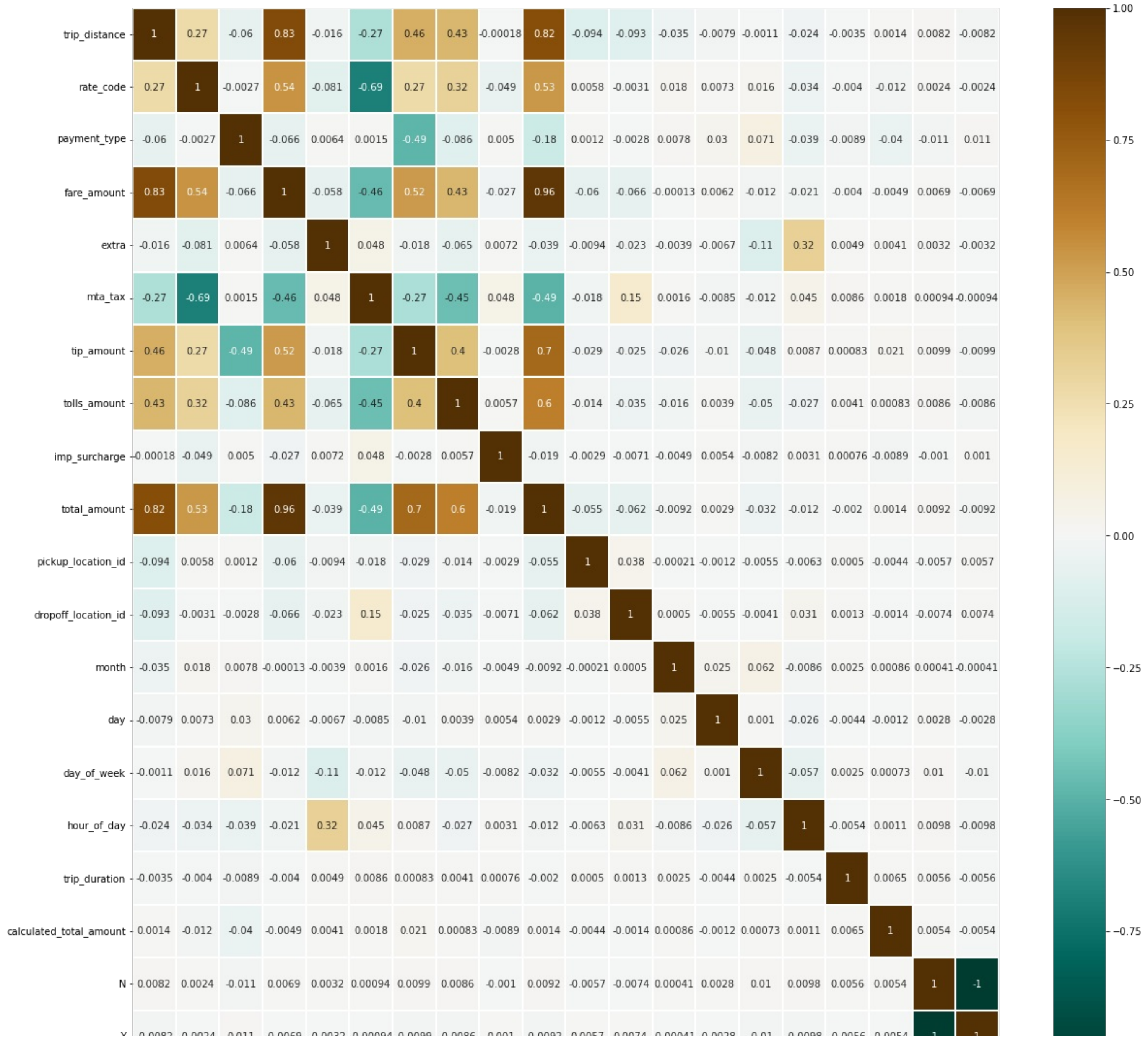
payment_type	-0.060372	-0.002726	1.000000	-0.066491	0.006365	0.001486	-0.491274	-0.086443	0.004975
fare_amount	0.829692	0.543670	-0.066491	1.000000	-0.057712	-0.461236	0.516761	0.430221	-0.026675
extra	-0.016118	-0.080895	0.006365	-0.057712	1.000000	0.047640	-0.018170	-0.065128	0.007239
mta_tax	-0.270702	-0.692998	0.001486	-0.461236	0.047640	1.000000	-0.273454	-0.448595	0.048335
tip_amount	0.455714	0.268014	-0.491274	0.516761	-0.018170	-0.273454	1.000000	0.399708	-0.002809
tolls_amount	0.432261	0.318800	-0.086443	0.430221	-0.065128	-0.448595	0.399708	1.000000	0.005692
imp_surcharge	-0.000183	-0.049447	0.004975	-0.026675	0.007239	0.048335	-0.002809	0.005692	1.000000
total_amount	0.816478	0.531029	-0.179079	0.958428	-0.039287	-0.492964	0.695904	0.602555	-0.019369
pickup_location_id	-0.094002	0.005835	0.001166	-0.059885	-0.009397	-0.017613	-0.029003	-0.014116	-0.002914
dropoff_location_id	-0.092665	-0.003110	-0.002771	-0.065842	-0.022558	0.151476	-0.025302	-0.035023	-0.007065
month	-0.035207	0.017861	0.007847	-0.000127	-0.003911	0.001563	-0.026451	-0.016343	-0.004881
day	-0.007886	0.007309	0.029884	0.006167	-0.006703	-0.008550	-0.010239	0.003878	0.005432
day_of_week	-0.001131	0.015818	0.071458	-0.012430	-0.106173	-0.011898	-0.048319	-0.049999	-0.008182
hour_of_day	-0.023668	-0.033840	-0.039133	-0.020998	0.320355	0.045090	0.008729	-0.027309	0.003125
trip_duration	-0.003486	-0.004011	-0.008933	-0.004027	0.004950	0.008558	0.000827	0.004057	0.000756
calculated_total_amount	0.001392	-0.011716	-0.040477	-0.004883	0.004083	0.001764	0.021426	0.000835	-0.008944
N	0.008222	0.002414	-0.010514	0.006881	0.003201	0.000941	0.009874	0.008639	-0.001008
Y	-0.008222	-0.002414	0.010514	-0.006881	-0.003201	-0.000941	-0.009874	-0.008639	0.001008

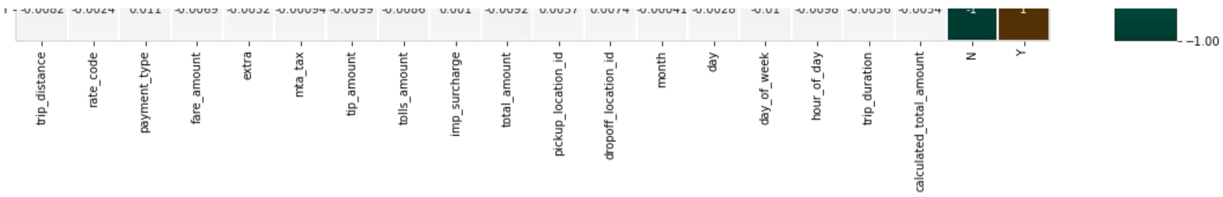
In [17]:

```
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(),linewidth=2,annot=True,cmap='BrBG_r')
```

Out[17]:

```
<AxesSubplot:>
```





In [18]:

```
X=df.drop(['calculated_total_amount'],axis=1)
X
```

Out[18]:

	trip_distance	rate_code	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	total_amount	pickup_loc
0	9.01	1	1	26.0	0.0	0.5	8.14	5.76	0.3	40.70	
1	0.20	1	1	3.0	0.0	0.5	0.75	0.00	0.3	4.55	
2	9.65	1	1	41.5	0.0	0.5	9.61	5.76	0.3	57.67	
3	9.50	1	1	30.0	0.5	0.5	9.25	5.76	0.3	46.31	
4	5.80	1	1	21.5	0.5	0.5	4.56	0.00	0.3	27.36	
...	...	...	...	...	...	...	...	...	...	...	
34995	22.43	1	1	59.5	0.5	0.5	10.00	5.76	0.3	76.56	
34996	9.16	1	1	30.0	0.0	0.5	6.58	5.76	0.3	43.14	
34997	6.78	1	1	23.0	0.0	0.5	5.95	0.00	0.3	29.75	
34998	0.26	1	2	3.0	0.0	0.5	0.00	0.00	0.3	3.80	
34999	18.40	1	1	53.0	1.0	0.5	10.96	0.00	0.3	65.76	

35000 rows × 19 columns

In [19]:

```
Y=df['calculated_total_amount']
Y
```

Out[19]:

```
0        24.30
1        37.40
2        30.36
3         4.30
4        23.80
         ...  
34995    29.76
34996    29.15
34997    42.67
34998    26.73
34999    62.80
Name: calculated_total_amount, Length: 35000, dtype: float64
```

In [20]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,train_size=0.8,random_state=25)
```

In [21]:

```
X_train
```

Out[21]:

	trip_distance	rate_code	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	imp_surcharge	total_amount	pickup_loc
14561	7.49	1	1	31.0	0.0	0.5	10.00	5.76	0.3	47.56	
4611	11.20	1	1	37.0	0.0	0.5	9.45	0.00	0.3	47.25	
34007	16.90	1	1	45.5	0.5	0.5	5.20	0.00	0.3	52.00	
217	11.55	1	1	34.0	0.0	0.5	5.22	0.00	0.3	40.02	
29867	4.11	1	1	25.0	1.0	0.5	5.36	0.00	0.3	32.16	
...	...	...	...	...	...	...	...	...	...	...	
33943	10.52	1	1	33.5	0.0	0.5	8.01	5.76	0.3	48.07	
26767	15.50	1	1	46.0	0.0	0.5	7.88	5.76	0.3	60.44	
6618	13.85	1	1	39.0	0.0	0.5	8.00	5.76	0.3	53.56	
24894	15.30	1	1	44.0	0.0	0.5	8.95	0.00	0.3	53.75	
29828	9.05	1	1	26.0	0.5	0.5	6.82	0.00	0.3	34.12	

28000 rows × 19 columns

In [22]:

```
Y_train
```

Out[22]:

```
14561    36.05
4611     27.35
34007    36.07
217      66.06
29867    32.15
         ...  
33943    28.56
26767    38.16
6618     57.30
24894    23.30
29828    39.35
Name: calculated_total_amount, Length: 28000, dtype: float64
```

In [23]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train,Y_train)
```

Out[23]:

```
LinearRegression()
```

In [28]:

```
Y_predicted=lr.predict(X_test)
```

Out[28]:

```
array([41.54929049, 39.22023101, 41.28781638, 41.68241497, 41.98994828,
       41.90215273, 41.92338192, 41.81247894, 41.47568355, 41.76313024,
       41.71418845, 41.60524191, 42.09190182, 41.44693365, 41.66324663,
       41.64794927, 41.50088286, 41.73668494, 39.02541886, 41.67335489,
       41.6295581 , 41.49414305, 41.7014165 , 41.89884492, 41.74026014,
       41.59897328, 41.69076732, 41.78529764, 41.70784954, 41.69655074])
```

In [30]:

```
from sklearn.metrics import r2_score
r2=r2_score(Y_test,Y_predicted)
r2
```

Out[30]:

```
-0.0009839465394116953
```

In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js