Assignment: Linear regression on the Advertising data

TODO: Edit this cell to fill in your NYU Net ID and your name:

- **Net ID**:
- **Name**:

To illustrate principles of linear regression, we are going to use some data from the textbook "An Introduction to Statistical Learning withApplications in R" (Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani) (available via NYU Library).

The dataset is described as follows:

> Suppose that we are statistical consultants hired by a client to provide advice on how to improve sales of a particular product. The `Advertising` data set consists of the sales of that product in 200 different markets, along with advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper.
>
> ...
>
> It is not possible for our client to directly increase sales of the product. On the other hand, they can control the advertising expenditure in each of the three media. Therefore, if we determine that there is an association between advertising and sales, then we can instruct our client to adjust advertising budgets, thereby indirectly increasing sales. In other words, our goal is to develop an accurate model that can be used to predict sales on the basis of the three media budgets.

Sales are reported in thousands of units, and TV, radio, and newspaper budgets, are reported in thousands of dollars.

For this assignment, you will fit a linear regression model to a small dataset. You will iteratively improve your linear regression model by examining the residuals at each stage, in order to identify problems with the model.

Make sure to include your name and net ID in a text cell at the top of the notebook.

```
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set()

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

1. Read in and pre-process data

In this section, you will read in the "Advertising" data, and make sure it is loaded correctly. Visually inspect the data using a pairplot, and note any meaningful observations. In particular, comment on which features appear to be correlated with product sales, and which features appear to be correlated with one another. Then, split the data into training data (70%) and test data (30%).

The code in this section is provided for you. However, you should add a text cell at the end of this section, in which you write your comments and observations.

Read in data

```
url = 'https://www.statlearning.com/s/Advertising.csv'
df  = pd.read_csv(url, index_col=0)
df.head()
```

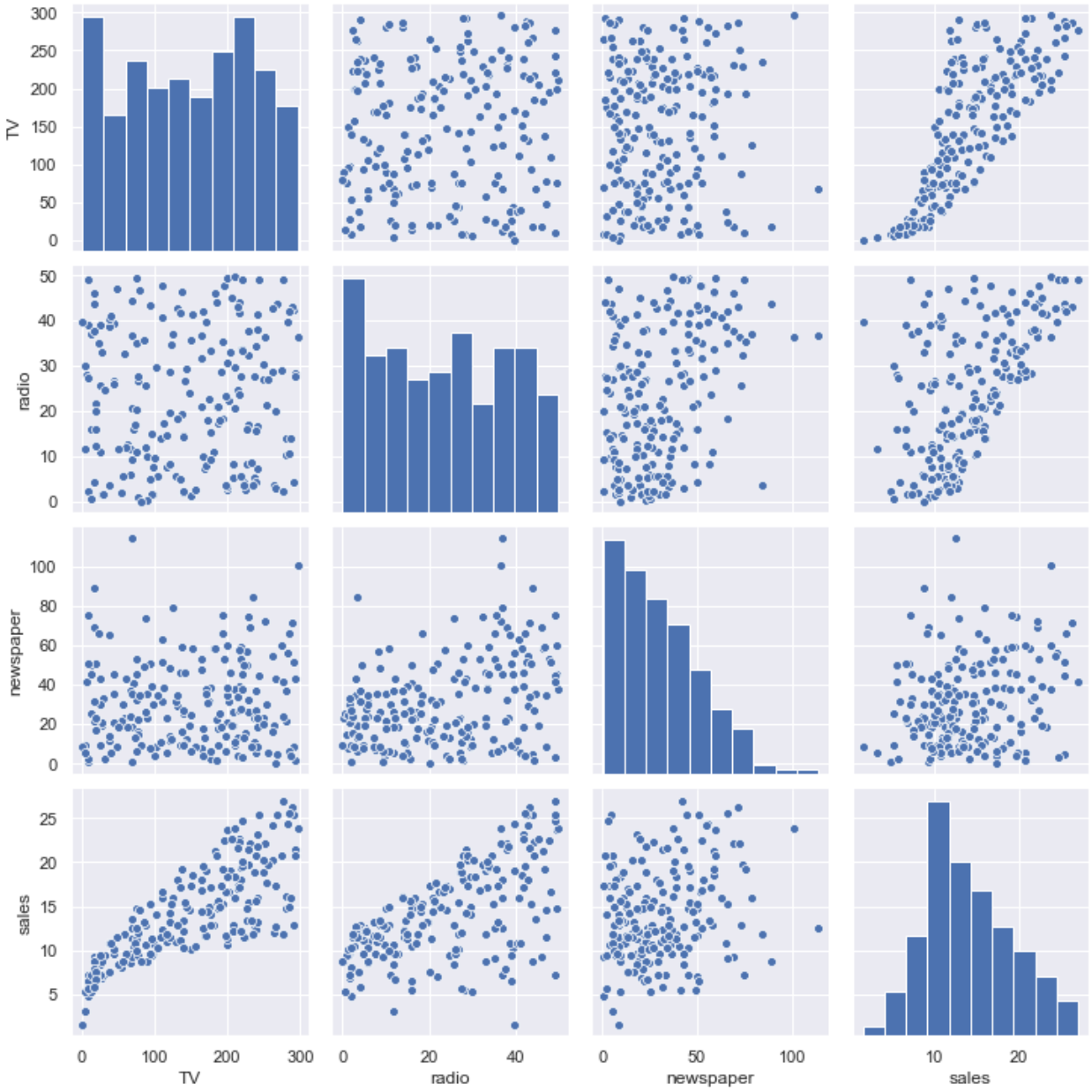
```
      TV  radio  newspaper  sales
1  230.1   37.8       69.2   22.1
2   44.5   39.3       45.1   10.4
3   17.2   45.9       69.3    9.3
4  151.5   41.3       58.5   18.5
5  180.8   10.8       58.4   12.9
```

Note that in this dataset, the first column in the data file is the row label; that's why we use `index_col=0` in the `read_csv` command. If we would omit that argument, then we would have an additional (unnamed) column in the dataset, containing the row number.

(You can try removing the `index_col` argument and re-running the cell above, to see the effect and to understand why we used this argument.)

Visually inspect the data

```
sns.pairplot(df);
```



The most important panels here are on the bottom row, where `sales` is on the vertical axis and the advertising budgets are on the horizontal axes.

Split up data We will use 70% of the data for training and the remaining 30% to test the regression model.

```
train, test = train_test_split(df, test_size=0.3)
```

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 140 entries, 200 to 8
Data columns (total 4 columns):
TV           140 non-null float64
```

```
radio        140 non-null float64
newspaper    140 non-null float64
sales        140 non-null float64
dtypes: float64(4)
memory usage: 5.5 KB
```

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60 entries, 44 to 147
Data columns (total 4 columns):
TV           60 non-null float64
radio        60 non-null float64
newspaper    60 non-null float64
sales        60 non-null float64
dtypes: float64(4)
memory usage: 2.3 KB
```

2. Fit simple linear regression models

Use the training data to fit a simple linear regression to predict product sales, for each of three features: TV ad budget, radio ad budget, and newspaper ad budget. In other words, you will fit *three* regression models, with each model being trained on one feature. For each of the three regression models, create a plot of the training data and the regression line, with product sales (y) on the vertical axis and the feature on which the model was trained (x) on the horizontal axis.

Also, for each regression model, print the intercept and coefficients, and compute the MSE and R2 on the training data, and MSE and R2 on the test data.

Comment on the results. Which type of ads seems to have the greatest association with increased product sales? Which regression model is most effective at predicting product sales?

The code in this section is provided for you. However, you should add text cells in which you write your comments, observations, and answers to the questions.

Fit a simple linear regression

```
reg_tv    = LinearRegression().fit(train[['TV']], train['sales'])
reg_radio = LinearRegression().fit(train[['radio']], train['sales'])
reg_news  = LinearRegression().fit(train[['newspaper']], train['sales'])
```

Look at coefficients

```
print("TV       : ", reg_tv.coef_[0], reg_tv.intercept_)
print("Radio    : ", reg_radio.coef_[0], reg_radio.intercept_)
print("Newspaper: ", reg_news.coef_[0], reg_news.intercept_)
```

```
TV       :  0.04432019964379259 7.458077887016144
Radio    :  0.16814578973441247 9.873543261975547
Newspaper:  0.04458318893858172 12.48551850369126
```

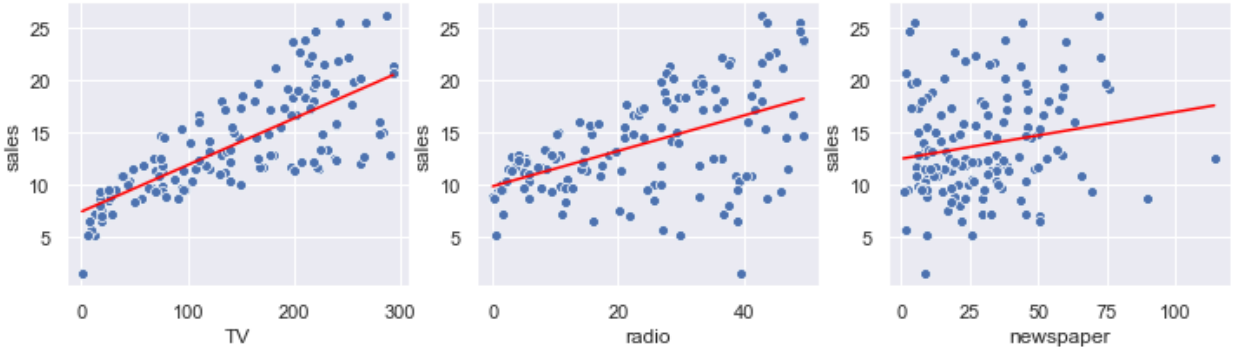
Plot data and regression line

```
fig = plt.figure(figsize=(12,3))

plt.subplot(1,3,1)
sns.scatterplot(data=train, x="TV", y="sales");
sns.lineplot(data=train, x="TV", y=reg_tv.predict(train[['TV']]), color='red');

plt.subplot(1,3,2)
sns.scatterplot(data=train, x="radio", y="sales");
sns.lineplot(data=train, x="radio", y=reg_radio.predict(train[['radio']]), color='red');

plt.subplot(1,3,3)
sns.scatterplot(data=train, x="newspaper", y="sales");
sns.lineplot(data=train, x="newspaper", y=reg_news.predict(train[['newspaper']]),
    color='red');
```



Compute R2, MSE for simple regression

```
y_pred_tr_tv    = reg_tv.predict(train[['TV']])
y_pred_tr_radio = reg_radio.predict(train[['radio']])
y_pred_tr_news  = reg_news.predict(train[['newspaper']])
```

```
r2_tr_tv    = metrics.r2_score(train['sales'], y_pred_tr_tv)
r2_tr_radio = metrics.r2_score(train['sales'], y_pred_tr_radio)
r2_tr_news  = metrics.r2_score(train['sales'], y_pred_tr_news)
print("TV       : ", r2_tr_tv)
print("Radio    : ", r2_tr_radio)
print("Newspaper: ", r2_tr_news)
```

```
TV       :  0.5702187599826805
Radio    :  0.2643649168894361
Newspaper:  0.035972731005180725
```

```
mse_tr_tv    = metrics.mean_squared_error(train['sales'], y_pred_tr_tv)
mse_tr_radio = metrics.mean_squared_error(train['sales'], y_pred_tr_radio)
mse_tr_news  = metrics.mean_squared_error(train['sales'], y_pred_tr_news)
print("TV       : ", mse_tr_tv)
print("Radio    : ", mse_tr_radio)
print("Newspaper: ", mse_tr_news)
```

```
TV       :  10.012028300658184
Radio    :  17.1370887914121
Newspaper:  22.45763053639427
```

```
y_pred_ts_tv    = reg_tv.predict(test[['TV']])
y_pred_ts_radio = reg_radio.predict(test[['radio']])
y_pred_ts_news  = reg_news.predict(test[['newspaper']])
```

```
r2_ts_tv    = metrics.r2_score(test['sales'], y_pred_ts_tv)
r2_ts_radio = metrics.r2_score(test['sales'], y_pred_ts_radio)
r2_ts_news  = metrics.r2_score(test['sales'], y_pred_ts_news)
print("TV       : ", r2_ts_tv)
print("Radio    : ", r2_ts_radio)
print("Newspaper: ", r2_ts_news)
```

```
TV       :  0.6636384746835695
Radio    :  0.39800055410825474
Newspaper:  0.05801336328882978
```

```
mse_ts_tv    = metrics.mean_squared_error(test['sales'], y_pred_ts_tv)
mse_ts_radio = metrics.mean_squared_error(test['sales'], y_pred_ts_radio)
mse_ts_news  = metrics.mean_squared_error(test['sales'], y_pred_ts_news)
print("TV       : ", mse_ts_tv)
print("Radio    : ", mse_ts_radio)
print("Newspaper: ", mse_ts_news)
```

```
TV       :  11.941204146411135
Radio    :  21.37164252854744
Newspaper:  33.44156178854693
```

3. Explore the residuals for the single linear regression models

We know that computing MSE or R2 is not sufficient to diagnose a problem with a linear regression.

Create some additional plots as described below to help you identify any problems with the regression. Use training data for all of the items below.

For each of the three regression models,

- Plot predicted sales ($\hat{y}$) on the vertical axis, and actual sales (y) on the horizontal axis. Make sure both axes use the same scale. Comment on your observations. What would you expect this plot to look like for a model that explains the data well?
- Compute the residuals ($y - \hat{y}$). Note that some of these will be negative, and some will be positive. What is the mean residual for each of the regression models? What *should* be the mean residual for a fitted linear regression model? Explain your answer.
- Plot the residuals ($y - \hat{y}$) on the vertical axis, and actual sales (y) on the horizontal axis. Use the same scale for all three subplots. Comment on your observations. Is there a pattern in the residuals (and if so, what might it indicate), or do they appear to have no pattern with respect to actual sales?
- For each of the three regression models AND each of the three features, plot the residuals ($y - \hat{y}$) on the vertical axis, and the feature (x) on the horizontal axis. This plot will include nine subplots in total. Make sure to clearly label each axis, and also label each subplot with a title that indicates which regression model it uses. Is there a pattern in the residuals (and if so, what might it indicate), or do they appear to have no pattern with respect to each of the three features?

The code in this section is not provided for you. You will need to write code, in addition to the text cells in which you write your comments, observations, and answers to the questions.

4. Try a multiple linear regression

Next, fit a multiple linear regression to predict product sales, using all three features to train a single model: TV ad budget, radio ad budget, and newspaper ad budget.

Print the intercept and coefficients, and compute the MSE and R2 on the training data, and MSE and R2 on the test data. Comment on the results. Make sure to explain any differences between the coefficients of the multiple regression model, and the coefficients of the three simple linear regression models. If they are different, why?

The code in the first part of this section is provided for you. However, you should add text cells in which you write your comments, observations, and answers to the questions.

Also repeat the analysis of part (3) for this regression model. Use training data for all of these items:

- Plot predicted sales ($\widehat{y}$) on the vertical axis, and actual sales (y) on the horizontal axis. Make sure both axes use the same scale. Comment on your observations. What would you expect this plot to look like for a model that explains the data well?
- Compute the residuals ($y - \widehat{y}$). What is the mean of the residuals? What *should* be the mean of the residuals for a fitted linear regression model? Explain your answer.
- Plot the residuals ($y - \widehat{y}$) on the vertical axis, and actual sales (y) on the horizontal axis. Comment on your observations. Is there a pattern in the residuals (and if so, what might it indicate), or do they appear to have no pattern with respect to actual sales?
- For each of the three features, plot the residuals ($y - \widehat{y}$) on the vertical axis, and the feature (x) on the horizontal axis. Make sure to clearly label each axis. Is there a pattern in the residuals (and if so, what might it indicate), or do they appear to have no pattern with respect to each of the three features?

The code in the last part of this section is not provided for you. You will need to write code, in addition to the text cells in which you write your comments, observations, and answers to the questions.

Fit a multiple linear regression

```
reg_multi = LinearRegression().fit(train[['TV', 'radio', 'newspaper']], train['sales'])
```

Look at coefficients

```
print("Coefficients (TV, radio, newspaper):", reg_multi.coef_)
print("Intercept: ", reg_multi.intercept_)
```

```
Coefficients (TV, radio, newspaper): [ 0.04606318  0.18245635 -0.0017739 ]
Intercept:  3.013511914838114
```

Compute R2, MSE for multiple regression

```
y_pred_tr_multi = reg_multi.predict(train[['TV', 'radio', 'newspaper']])

r2_tr_multi  = metrics.r2_score(train['sales'], y_pred_tr_multi)
mse_tr_multi = metrics.mean_squared_error(train['sales'], y_pred_tr_multi)

print("Multiple regression R2:  ", r2_tr_multi)
print("Multiple regression MSE: ", mse_tr_multi)
```

```
Multiple regression R2:   0.8780768933991069
Multiple regression MSE:  2.840276587556763
```

```
y_pred_ts_multi = reg_multi.predict(test[['TV', 'radio', 'newspaper']])

r2_ts_multi  = metrics.r2_score(test['sales'], y_pred_ts_multi)
mse_ts_multi = metrics.mean_squared_error(test['sales'], y_pred_ts_multi)

print("Multiple regression R2:  ", r2_ts_multi)
print("Multiple regression MSE: ", mse_ts_multi)
```

```
Multiple regression R2:   0.9241616089749901
Multiple regression MSE:  2.692346303617978
```

5. Linear regression with interaction terms

Our multiple linear regression includes additive effects of all three types of advertising media. However, it does not include *interaction* effects, in which combining different types of advertising media together results in a bigger boost in sales than just the additive effect of the individual media. The pattern in the residuals plots from parts (1) through (4) suggest that a model including an interaction effect may explain sales data better than a model including additive effects. Add four columns to your data frame:

- `newspaper` × `radio`
- `TV` × `radio`
- `newspaper` × `TV`
- `newspaper` × `radio` × `TV`

Then, train a linear regression model on all seven features: the three types of ad budgets, and the four interaction effects. Repeat the analysis of part (4) for the model including interaction effects. Comment on the results. Are the interaction effects helpful for explaining the effect of ads on product sales? Are there any patterns evident in the residual plots that suggest further opportunities for improving the model?

(If you think the results suggest further opportunities for improving the model, you are welcome to try and to comment on the results!)

The code in this section is not provided for you. You will need to write code, in addition to the text cells in which you write your comments, observations, and answers to the questions.