Linear Regression in StatsModels - Lab

Introduction

It's time to apply the StatsModels skills from the previous lesson! In this lab , you'll explore a slightly more complex example to study the impact of spending on different advertising channels on total sales.

Objectives

You will be able to:

- Perform a linear regression using StatsModels
- Evaluate a linear regression model using StatsModels
- Interpret linear regression coefficients using StatsModels

Let's Get Started

In this lab, you'll work with the "Advertising Dataset", which is a very popular dataset for studying simple regression. The dataset is available on Kaggle, but we have downloaded it for you. It is available in this repository as `advertising.csv`. You'll use this dataset to answer this question:

> Which advertising channel has the strongest relationship with sales volume, and can be used to model and predict the sales?

The columns in this dataset are:

1. `sales` : the number of widgets sold (in thousands)
2. `tv` : the amount of money (in thousands of dollars) spent on TV ads
3. `radio` : the amount of money (in thousands of dollars) spent on radio ads
4. `newspaper` : the amount of money (in thousands of dollars) spent on newspaper ads

Step 1: Exploratory Data Analysis

```
# Load necessary libraries and import the data
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('ggplot')
data = pd.read_csv('advertising.csv', index_col=0)
```

```
# Check the columns and first few rows
data.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

```
# Generate summary statistics for data with .describe()
data.describe()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

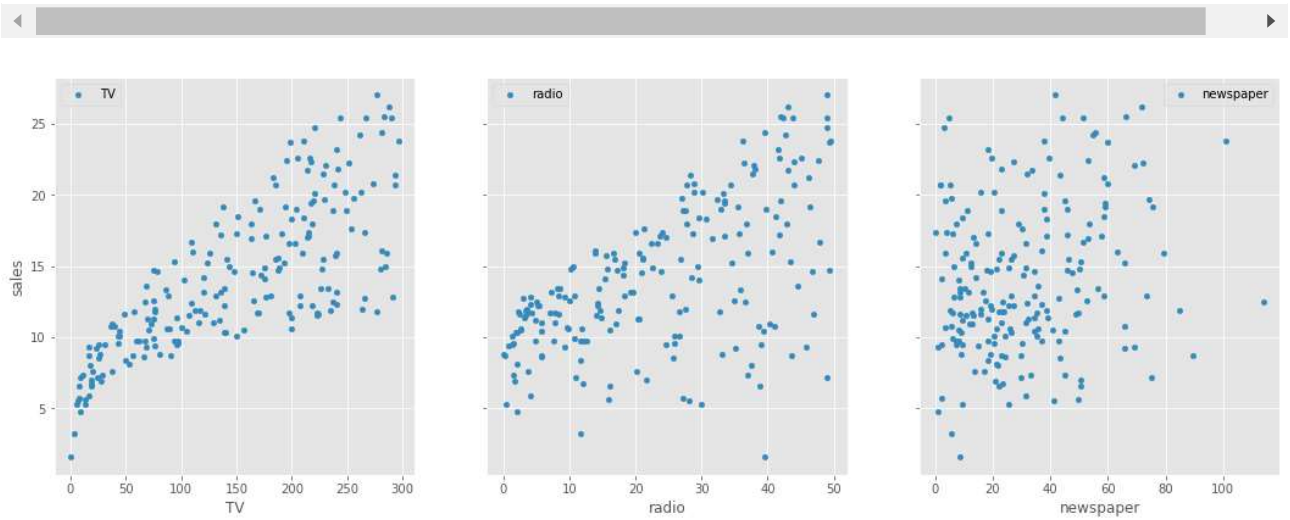
	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

Based on what you have seen so far, describe the contents of this dataset. Remember that our business problem is asking us to build a model that predicts sales.

▶ **Answer (click to reveal)**

Now, use scatter plots to plot each predictor (TV, radio, newspaper) against the target variable.

```
# Visualize the relationship between the preditors and the target using scatter plot
fig, axs = plt.subplots(1, 3, sharey=True, figsize=(18, 6))
for idx, channel in enumerate(['TV', 'radio', 'newspaper']):
    data.plot(kind='scatter', x=channel, y='sales', ax=axs[idx], label=channel)
    axs[idx].legend()
```



Does there appear to be a linear relationship between these predictors and the target?

▶ **Answer (click to reveal)**

Step 2: Run a Simple Linear Regression with `TV` as the Predictor

As the analysis above indicates, `TV` looks like it has the strongest relationship with `sales`. Let's attempt to quantify that using linear regression.

```
# Import libraries
import statsmodels.api as sm

# Determine X and y values
X = data[["TV"]]
y = data["sales"]

# Create an OLS model
model = sm.OLS(endog=y, exog=sm.add_constant(X))
```

```
# Get model results
results = model.fit()

# Display results summary
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.612
Model:                            OLS   Adj. R-squared:                  0.610
Method:                 Least Squares   F-statistic:                     312.1
Date:                Fri, 06 May 2022   Prob (F-statistic):           1.47e-42
Time:                        18:09:18   Log-Likelihood:                -519.05
No. Observations:                 200   AIC:                             1042.
Df Residuals:                     198   BIC:                             1049.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          7.0326      0.458     15.360      0.000       6.130       7.935
TV             0.0475      0.003     17.668      0.000       0.042       0.053
==============================================================================
Omnibus:                        0.531   Durbin-Watson:                   1.935
Prob(Omnibus):                  0.767   Jarque-Bera (JB):                0.669
Skew:                          -0.089   Prob(JB):                        0.716
Kurtosis:                       2.779   Cond. No.                         338.
==============================================================================

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

Step 3: Evaluate and Interpret Results from Step 2

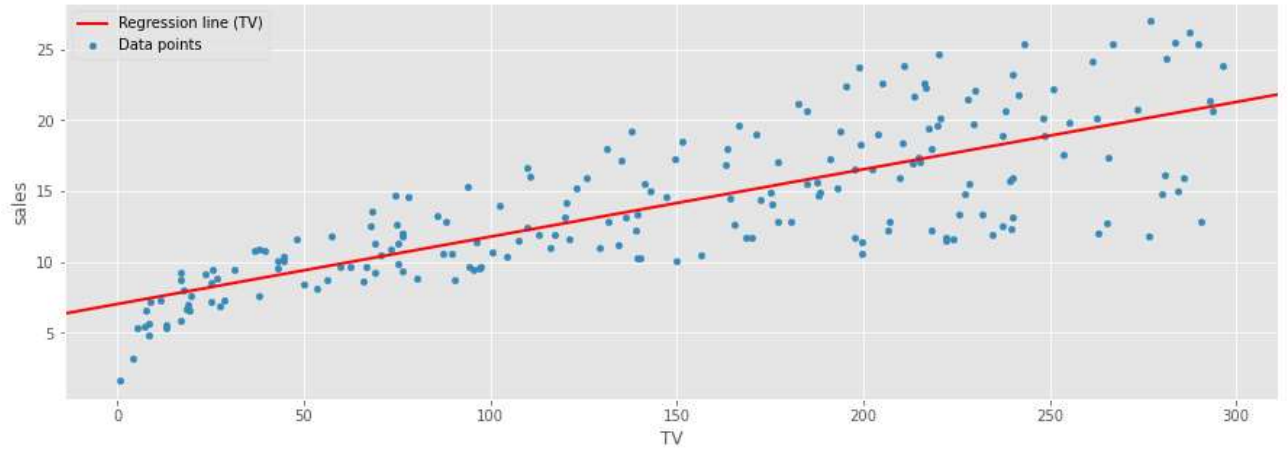
How does this model perform overall? What do the coefficients say about the relationship between the variables?

▶ **Answer (click to reveal)**

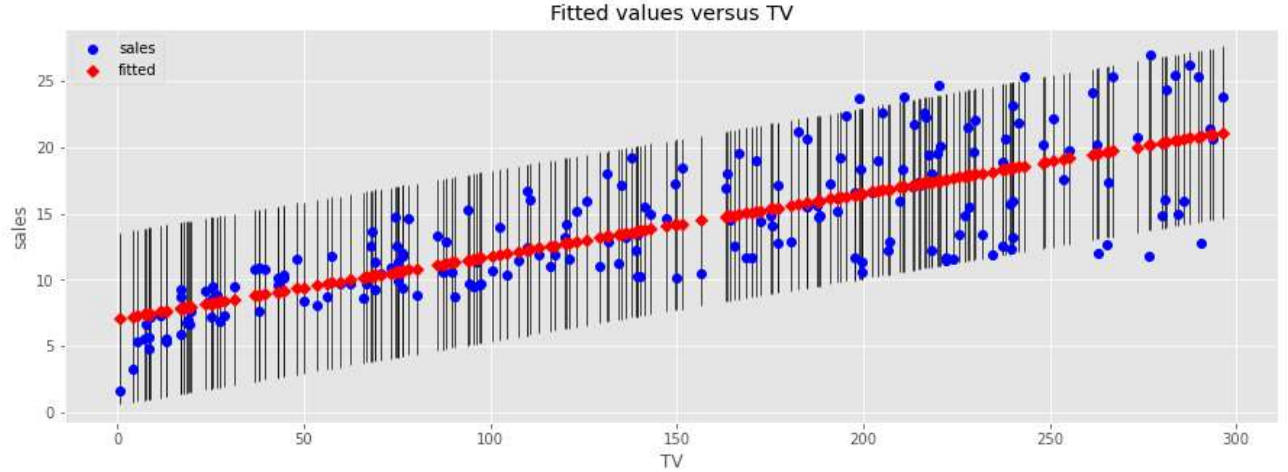
Step 4: Visualize Model with `TV` as Predictor

Create at least one visualization that shows the prediction line against a scatter plot of `TV` vs. sales, as well as at least one visualization that shows the residuals.

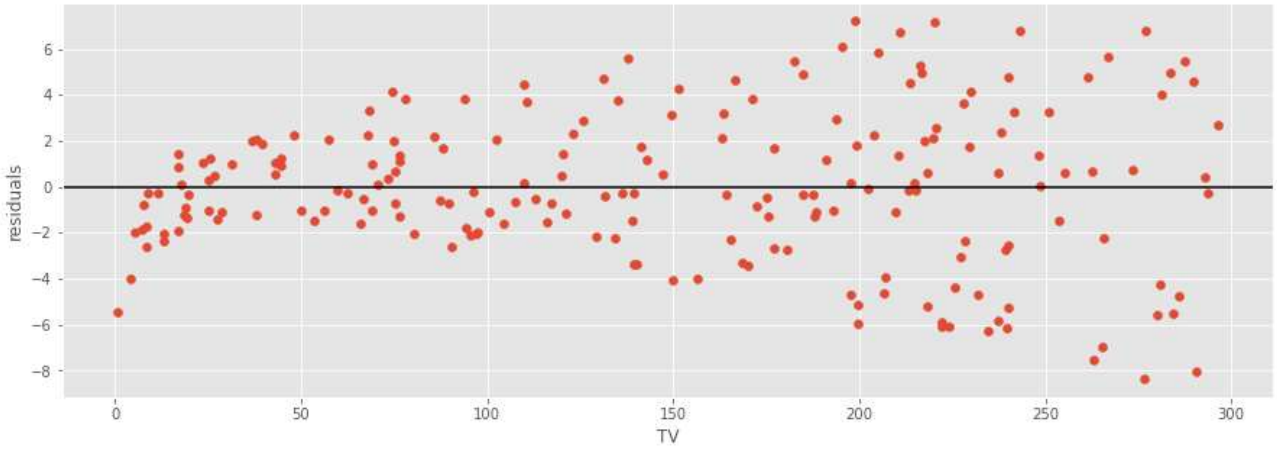
```
# abline_plot version of model fit
fig, ax = plt.subplots(figsize=(15,5))
data.plot(x="TV", y="sales", kind="scatter", label="Data points", ax=ax)
sm.graphics.abline_plot(model_results=results, label="Regression line (TV)", c="red"
ax.legend()
plt.show()
```



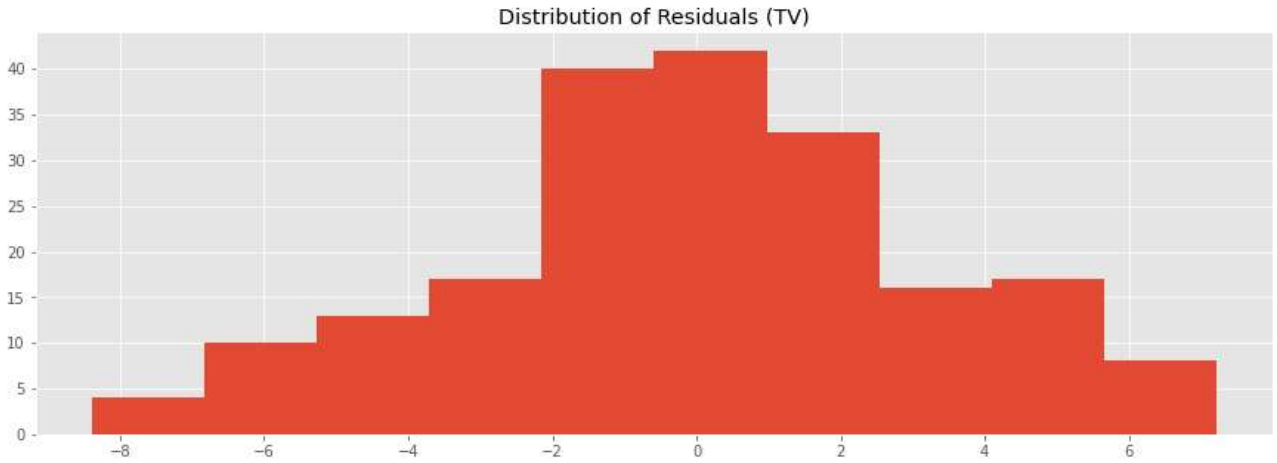

```
# plot_fit version of model fit
fig, ax = plt.subplots(figsize=(15,5))
sm.graphics.plot_fit(results, "TV", ax=ax)
plt.show()
```



```
# Plotting residuals vs. TV

fig, ax = plt.subplots(figsize=(15,5))

ax.scatter(data["TV"], results.resid)
ax.axhline(y=0, color="black")

ax.set_xlabel("TV")
ax.set_ylabel("residuals");
```

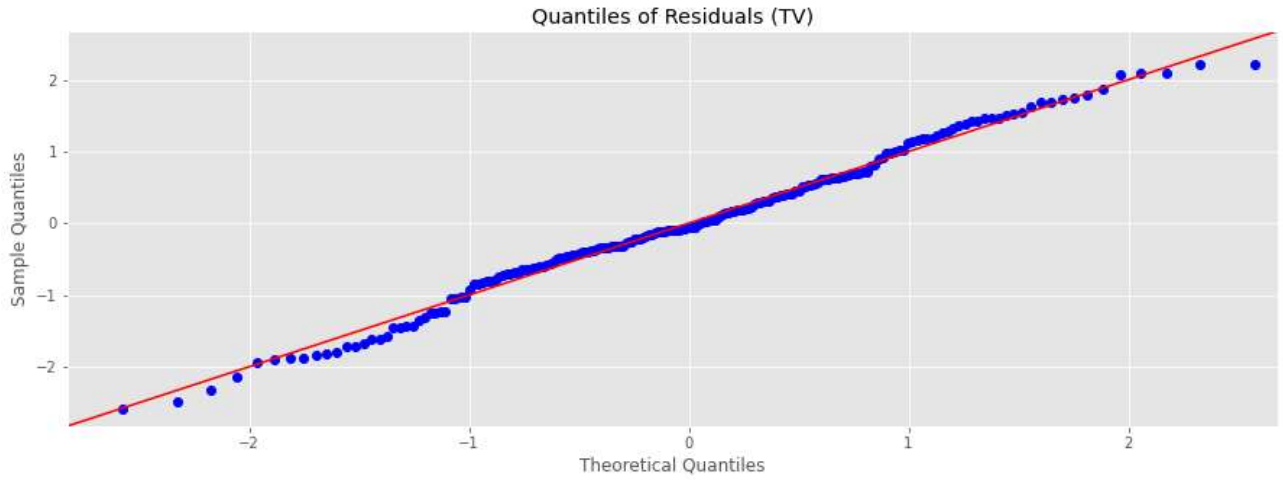



```
# Plotting residual histogram
fig, ax = plt.subplots(figsize=(15,5))

ax.hist(results.resid)
ax.set_title("Distribution of Residuals (TV)");
```




```
# Plotting residual Q-Q plot
from scipy.stats import norm
fig, ax = plt.subplots(figsize=(15,5))
sm.graphics.qqplot(results.resid, dist=norm, line="45", fit=True, ax=ax)
```

```
ax.set_title("Quantiles of Residuals (TV)")
plt.show()
```



Step 5: Repeat Steps 2-4 with `radio` as Predictor

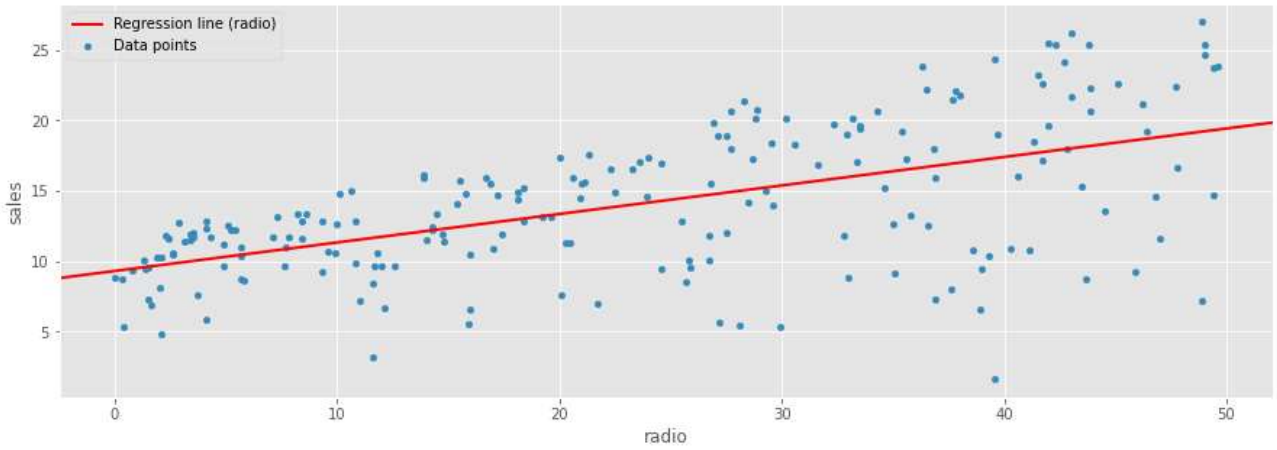
Compare and contrast the model performance, coefficient value, etc. The goal is to answer the business question described above.

```
# Run model
X_radio = data[["radio"]]
model_radio = sm.OLS(endog=y, exog=sm.add_constant(X_radio))

# Display results
results_radio = model_radio.fit()
print(results_radio.summary())
```

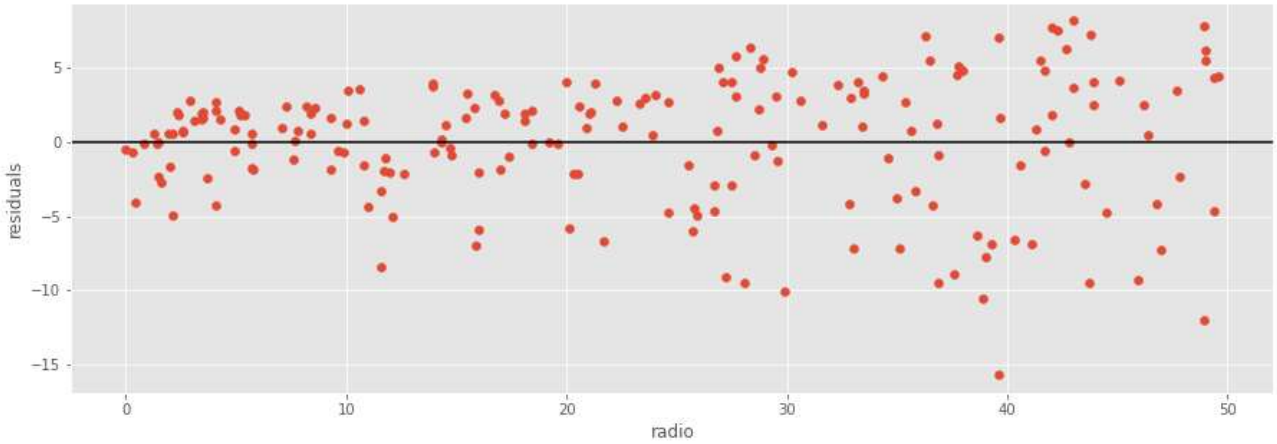
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.332
Model:                            OLS   Adj. R-squared:                  0.329
Method:                 Least Squares   F-statistic:                     98.42
Date:                Fri, 06 May 2022   Prob (F-statistic):           4.35e-19
Time:                        18:09:37   Log-Likelihood:                -573.34
No. Observations:                 200   AIC:                             1151.
Df Residuals:                     198   BIC:                             1157.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          9.3116      0.563     16.542      0.000       8.202      10.422
```

```
radio          0.2025      0.020      9.921      0.000       0.162       0.243
==============================================================================
Omnibus:                       19.358   Durbin-Watson:                   1.946
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               21.910
Skew:                          -0.764   Prob(JB):                     1.75e-05
Kurtosis:                       3.544   Cond. No.                         51.4
==============================================================================

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

```
# Visualize model fit
fig, ax = plt.subplots(figsize=(15,5))
data.plot(x="radio", y="sales", kind="scatter", label="Data points", ax=ax)
sm.graphics.abline_plot(model_results=results_radio, label="Regression line (radio)"
ax.legend()
plt.show()
```




```
# Visualize residuals
fig, ax = plt.subplots(figsize=(15,5))

ax.scatter(data["radio"], results_radio.resid)
ax.axhline(y=0, color="black")

ax.set_xlabel("radio")
ax.set_ylabel("residuals");
```



▶ Answer (click to reveal)

Step 6: Repeat Steps 2-4 with `newspaper` as Predictor

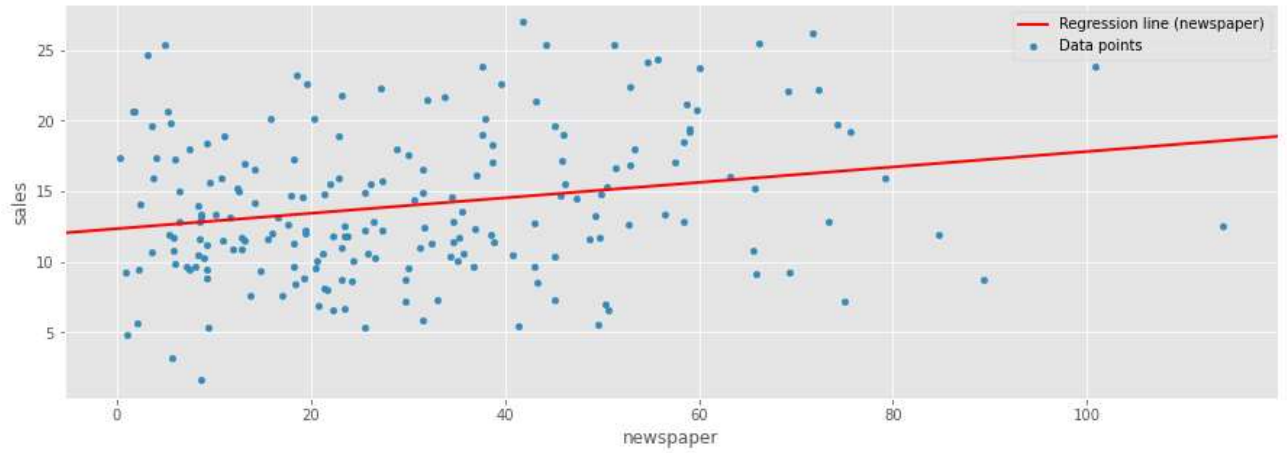
Once again, use this information to compare and contrast.

```
# Run model
X_newspaper = data[["newspaper"]]
model_newspaper = sm.OLS(endog=y, exog=sm.add_constant(X_newspaper))

# Display results
results_newspaper = model_newspaper.fit()
print(results_newspaper.summary())
```

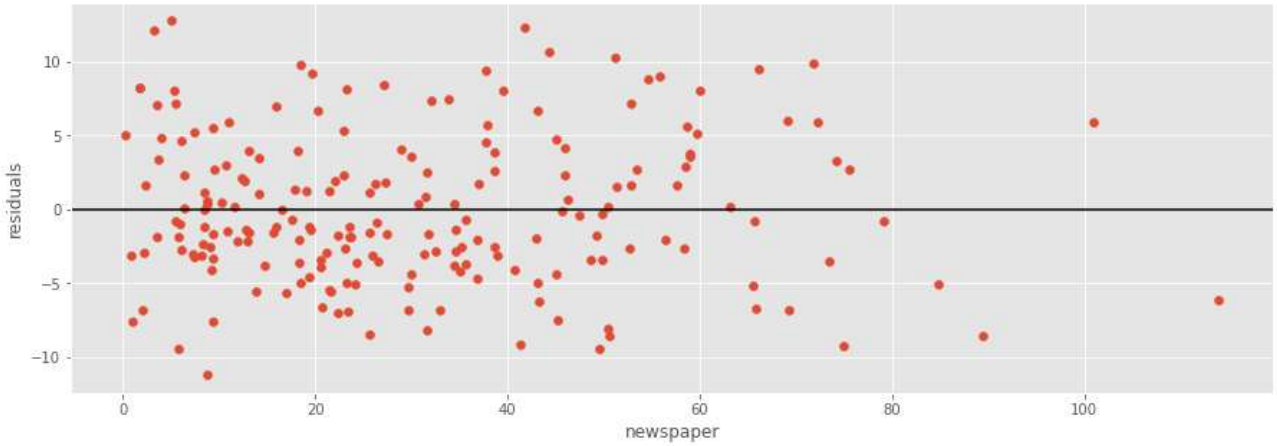
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.052
Model:                            OLS   Adj. R-squared:                  0.047
Method:                 Least Squares   F-statistic:                     10.89
Date:                Fri, 06 May 2022   Prob (F-statistic):            0.00115
Time:                        18:09:43   Log-Likelihood:                -608.34
No. Observations:                 200   AIC:                             1221.
Df Residuals:                     198   BIC:                             1227.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         12.3514      0.621     19.876      0.000      11.126      13.577
newspaper      0.0547      0.017      3.300      0.001       0.022       0.087
==============================================================================
Omnibus:                        6.231   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.044   Jarque-Bera (JB):                5.483
```

```
Skew:                          0.330   Prob(JB):                       0.0645
Kurtosis:                      2.527   Cond. No.                         64.7
==============================================================================

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

```
# Visualize model fit
fig, ax = plt.subplots(figsize=(15,5))
data.plot(x="newspaper", y="sales", kind="scatter", label="Data points", ax=ax)
sm.graphics.abline_plot(model_results=results_newspaper, label="Regression line (new
ax.legend()
plt.show()
```




```
# Visualize residuals
fig, ax = plt.subplots(figsize=(15,5))

ax.scatter(data["newspaper"], results_newspaper.resid)
ax.axhline(y=0, color="black")

ax.set_xlabel("newspaper")
ax.set_ylabel("residuals");
```



▶ Answer (click to reveal)

Summary

In this lab, you ran a complete regression analysis with a simple dataset. You used StatsModels to perform linear regression and evaluated your models using statistical metrics as well as visualizations. You also reached a conclusion about how you would answer a business question using linear regression.

Releases

No releases published

Packages

No packages published

Contributors 6

Languages

- **Jupyter Notebook** 100.0%