1 Praktikum 3

Muhamad Rizal Arfiyan - 22.11.5227 - IF11

https://github.com/rizalarfiyan/big-data

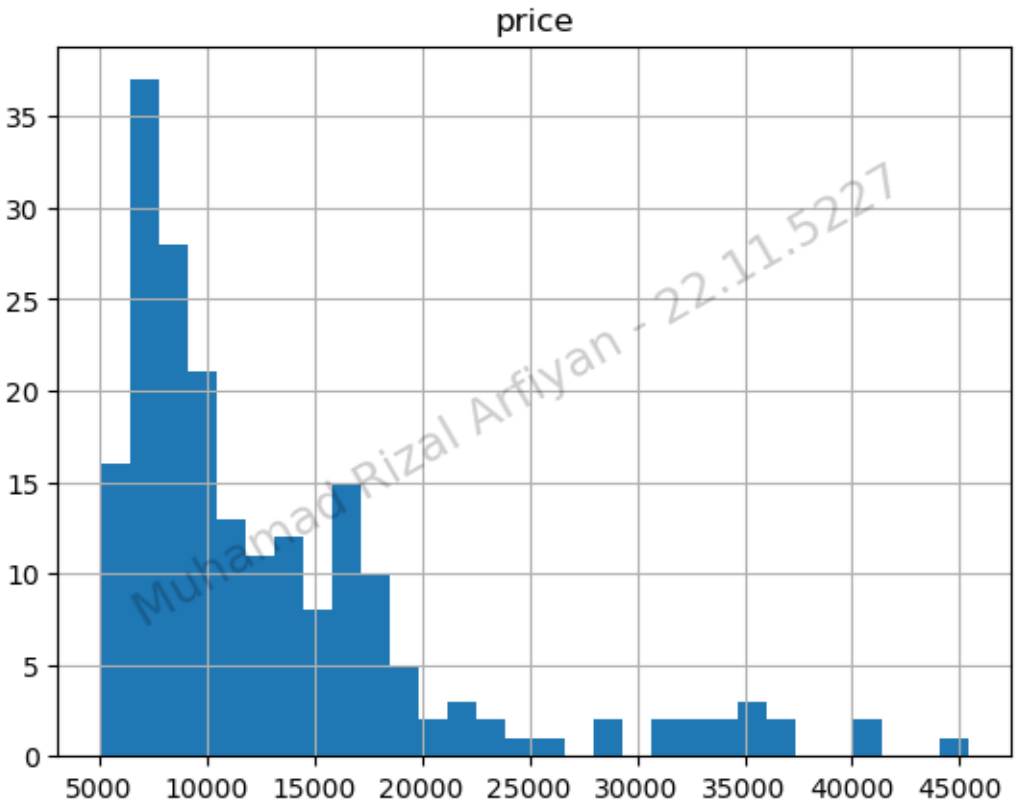
[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

path = "./automobileEDA.csv"
df = pd.read_csv(path)
df.head()
```

[1]:
```
   symboling  normalized-losses         make aspiration num-of-doors  \
0          3                122  alfa-romero        std          two
1          3                122  alfa-romero        std          two
2          1                122  alfa-romero        std          two
3          2                164         audi        std         four
4          2                164         audi        std         four

    body-style drive-wheels engine-location  wheel-base    length  …  \
0  convertible          rwd           front        88.6  0.811148  …
1  convertible          rwd           front        88.6  0.811148  …
2    hatchback          rwd           front        94.5  0.822681  …
3        sedan          fwd           front        99.8  0.848630  …
4        sedan          4wd           front        99.4  0.848630  …

   compression-ratio  horsepower  peak-rpm city-mpg highway-mpg    price  \
0                9.0       111.0    5000.0       21          27  13495.0
1                9.0       111.0    5000.0       21          27  16500.0
2                9.0       154.0    5000.0       19          26  16500.0
3               10.0       102.0    5500.0       24          30  13950.0
4                8.0       115.0    5500.0       18          22  17450.0

  city-L/100km  horsepower-binned  diesel  gas
0    11.190476             Medium       0    1
1    11.190476             Medium       0    1
2    12.368421             Medium       0    1
3     9.791667             Medium       0    1
4    13.055556             Medium       0    1

[5 rows x 29 columns]
```

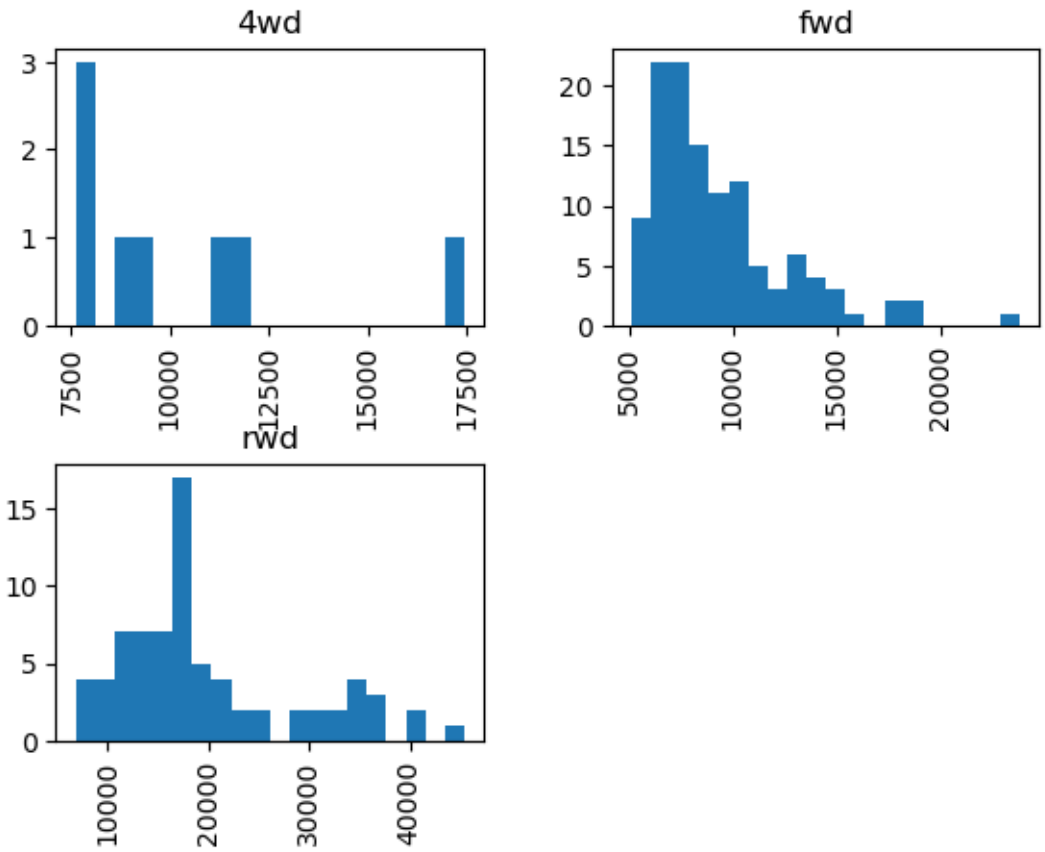
[2]:
```
df.hist(column="price", bins=30)
plt.text(
    0.5,
```

```
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=18,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
```

[2]: Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')



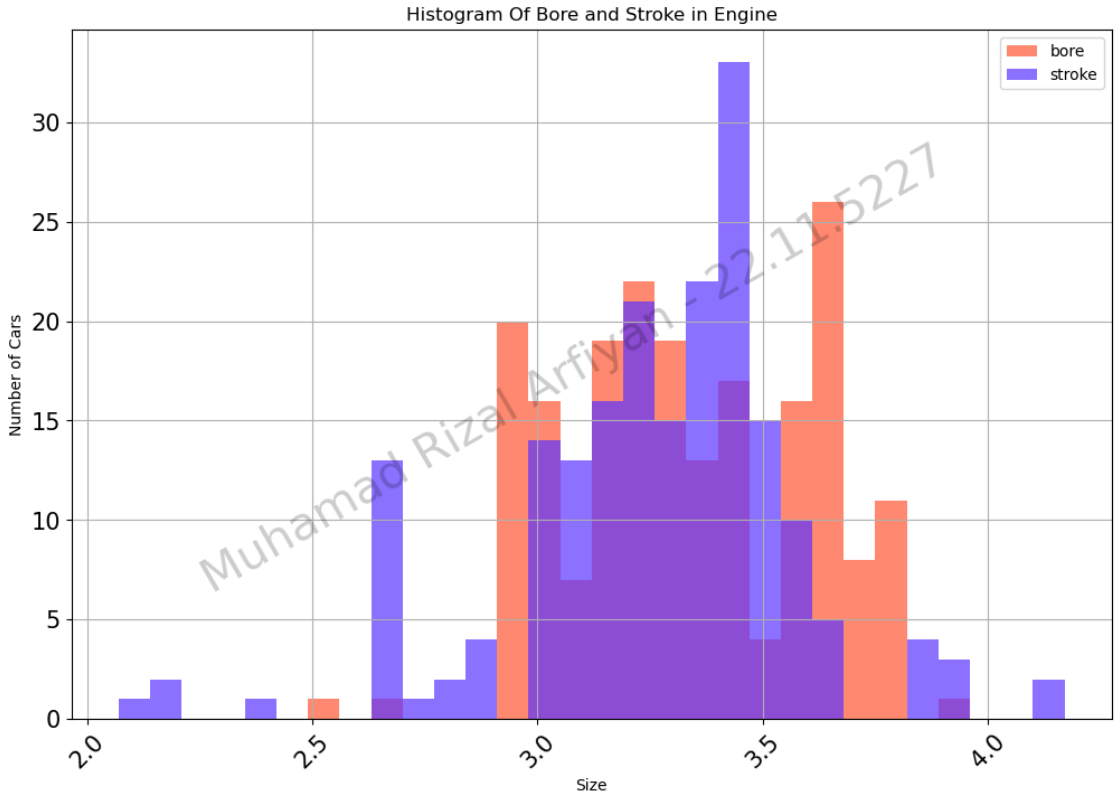
[3]:
```
df.hist(column="price", by="drive-wheels", bins=20);
```



[4]:
```
df[["bore", "stroke"]].plot(
    kind="hist",
    alpha=0.7,
    bins=30,
    title="Histogram Of Bore and Stroke in Engine",
    rot=45,
    grid=True,
    figsize=(12, 8),
    fontsize=15,
    color=["#FF5733", "#5C33FF"],
)
plt.xlabel("Size")
plt.ylabel("Number of Cars")
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=30,
    color="black",
    ha="center",
```

```
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
```

[4]: Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')



[5]:
```
from scipy import stats

pearson_coef, p_value = stats.pearsonr(df["wheel-base"], df["price"])
print(
    "The Pearson Correlation Coefficient is",
    pearson_coef,
    " with a P-value of P =",
    p_value,
)
```

```
The Pearson Correlation Coefficient is 0.5846418222655083  with a P-value of P =
8.076488270732552e-20
```

[6]:
```
pearson_coef, p_value = stats.pearsonr(df["horsepower"], df["price"])
print(
    "The Pearson Correlation Coefficient is",
    pearson_coef,
    " with a P-value of P = ",
    p_value,
)
```

```
The Pearson Correlation Coefficient is 0.8095745670036555  with a P-value of P =
6.369057428260919e-48
```

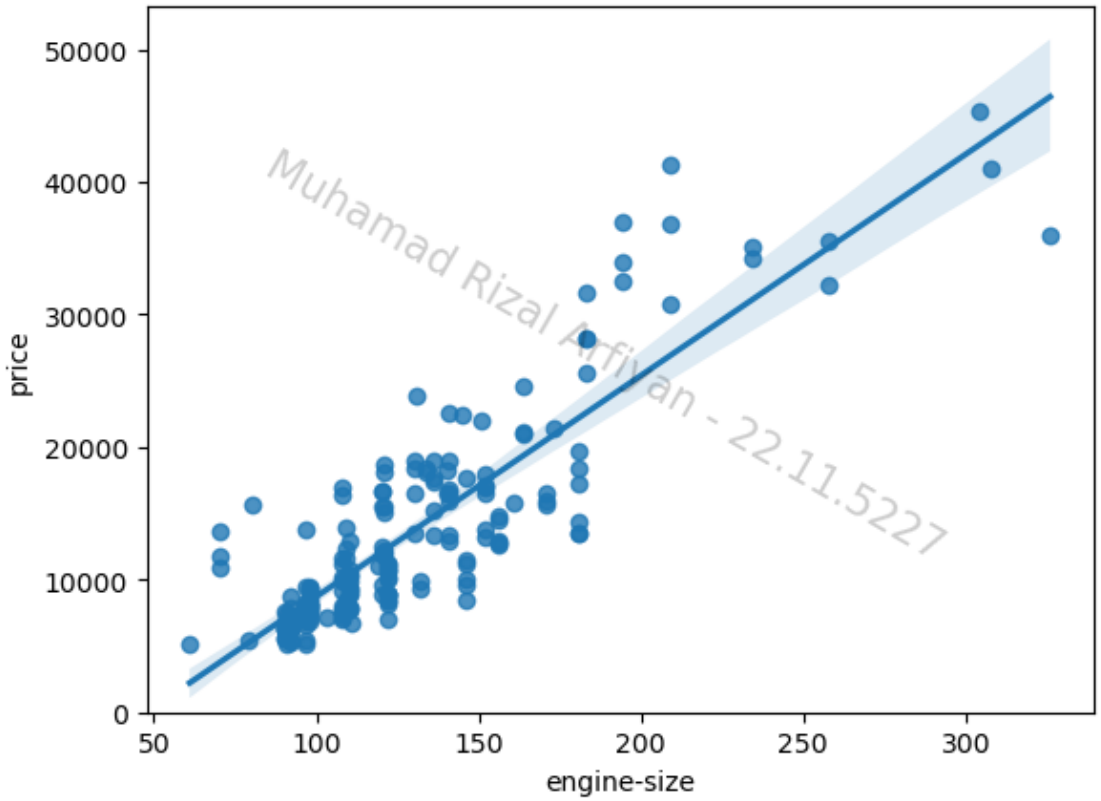
[7]:
```
print(df.dtypes)
```

```
symboling              int64
normalized-losses      int64
make                  object
aspiration            object
num-of-doors          object
body-style            object
drive-wheels          object
engine-location       object
wheel-base           float64
length               float64
width                float64
height               float64
curb-weight            int64
engine-type           object
num-of-cylinders      object
engine-size            int64
fuel-system           object
bore                 float64
stroke               float64
compression-ratio    float64
horsepower           float64
peak-rpm             float64
city-mpg               int64
highway-mpg            int64
price                float64
city-L/100km         float64
horsepower-binned     object
diesel                 int64
gas                    int64
dtype: object
```

[8]:
```
sns.regplot(x="engine-size", y="price", data=df)
plt.ylim(
    0,
```

```
)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=16,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=-30,
)
```

[8]: Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')



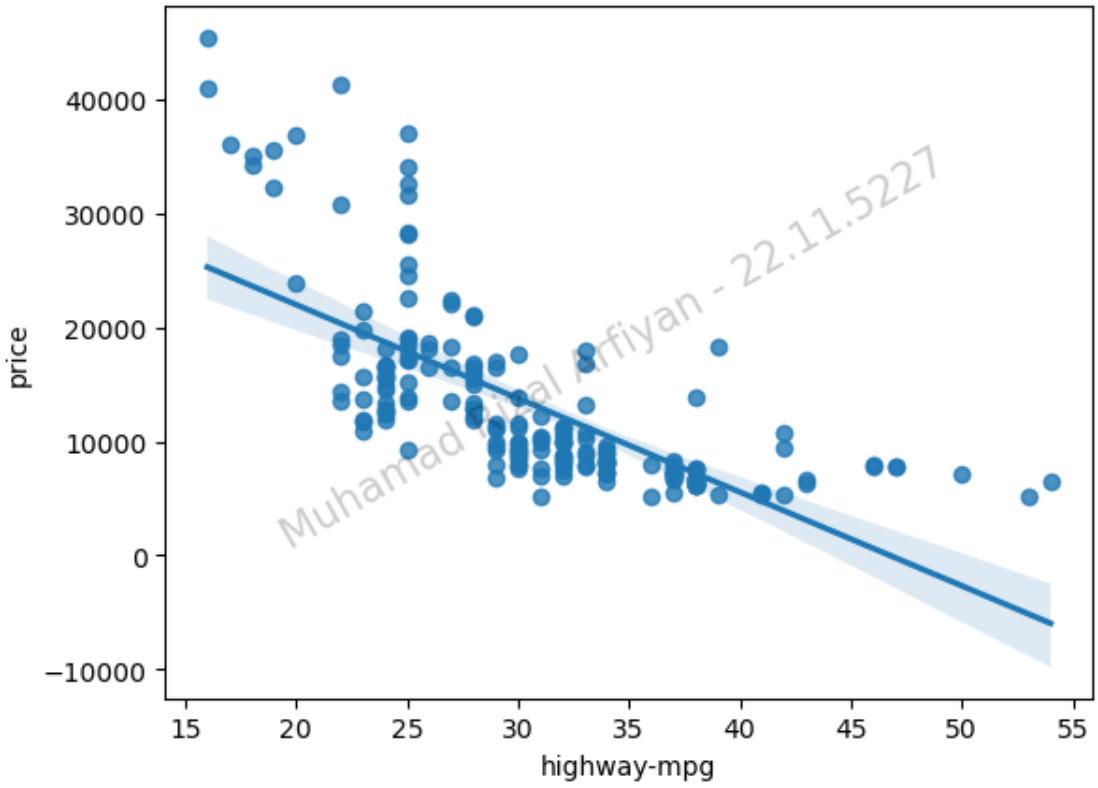
[9]:
```
df[["engine-size", "price"]].corr()
```

[9]:

	engine-size	price
engine-size	1.000000	0.872335
price	0.872335	1.000000

[10]:
```
sns.regplot(x="highway-mpg", y="price", data=df)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=16,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
```

[10]:
```
Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')
```



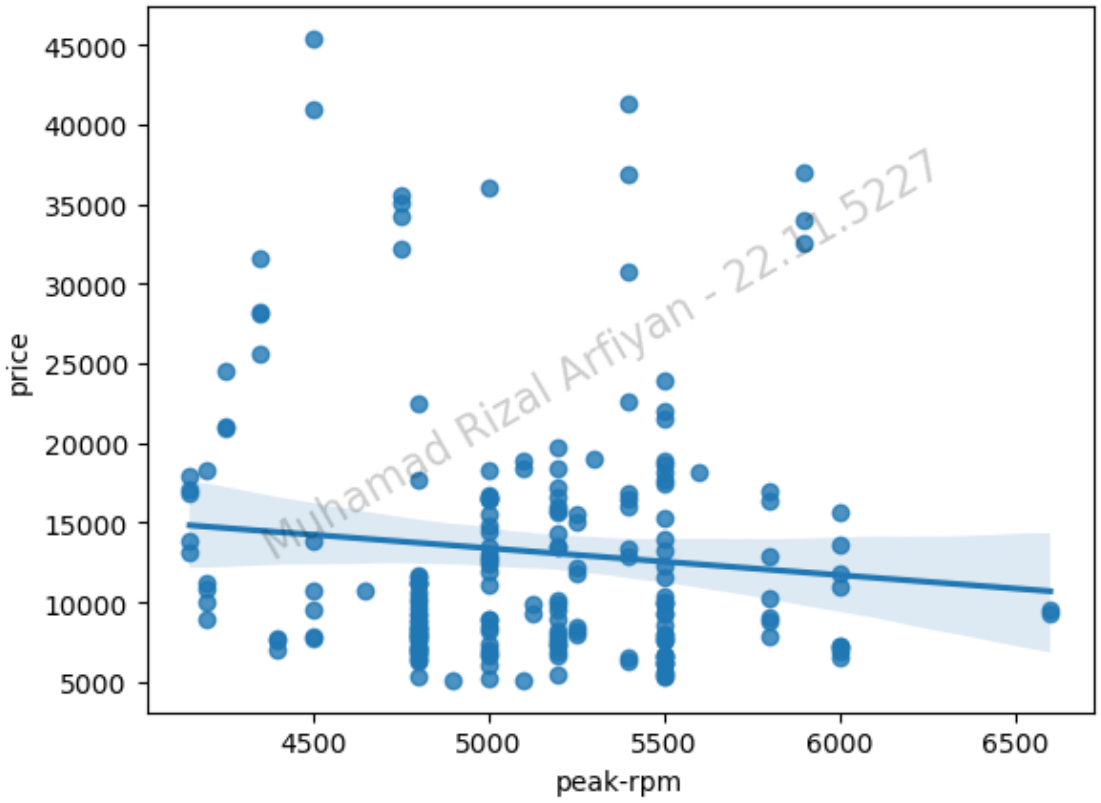
[11]:
```
df[["highway-mpg", "price"]].corr()
```

[11]:

	highway-mpg	price
highway-mpg	1.000000	-0.704692
price	-0.704692	1.000000

[12]:
```
sns.regplot(x="peak-rpm", y="price", data=df)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=16,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
```

[12]:
```
Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')
```

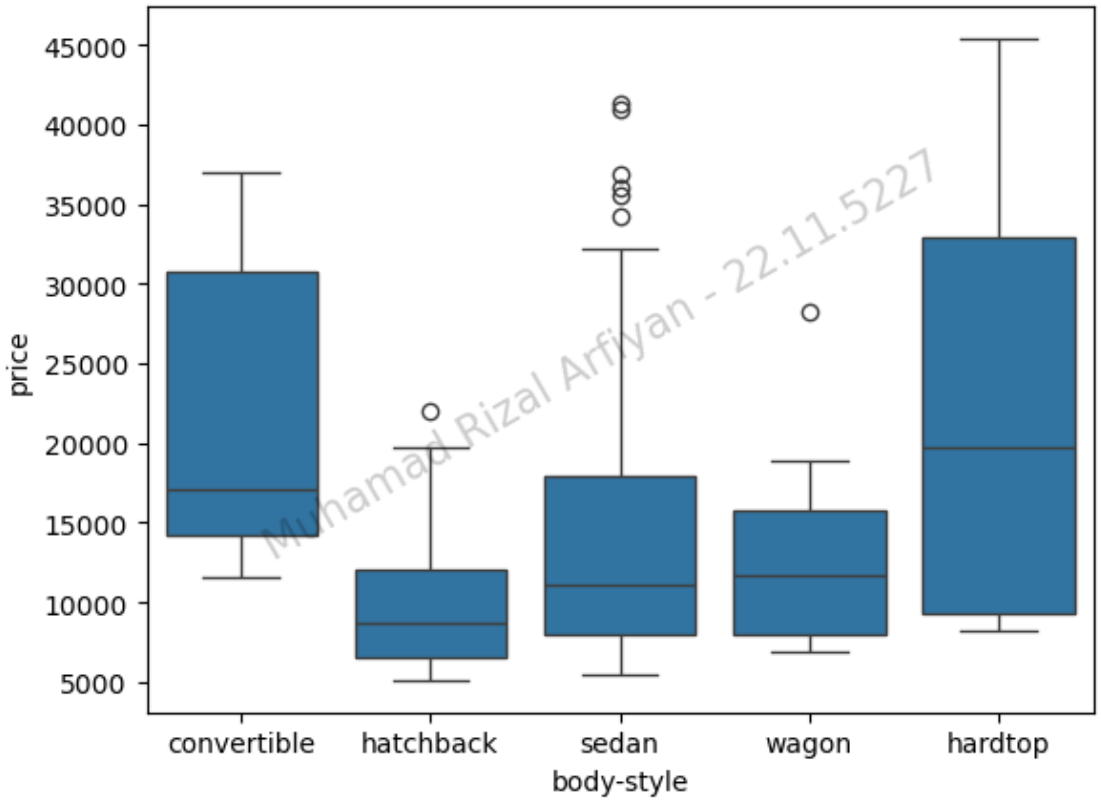


[13]:
```
df[["peak-rpm", "price"]].corr()
```

[13]:
```
           peak-rpm     price
peak-rpm   1.000000 -0.101616
price     -0.101616  1.000000
```

[14]:
```
sns.boxplot(x="body-style", y="price", data=df)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=16,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
```

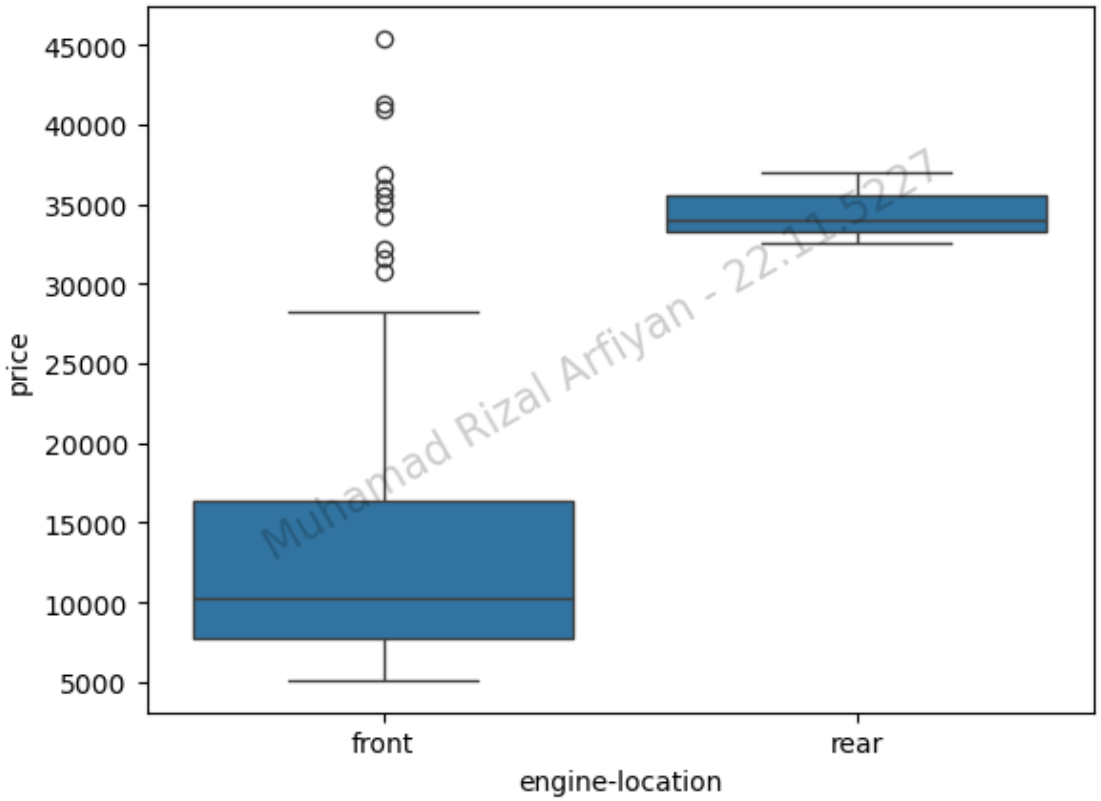
[14]:
```
Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')
```



[15]:
```
sns.boxplot(x="engine-location", y="price", data=df)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
```

```
    fontsize=16,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
```

[15]: Text(0.5, 0.5, 'Muhamad Rizal Arfiyan - 22.11.5227')



[16]:
```
df.describe()
```

[16]:

	symboling	normalized-losses	wheel-base	length	width	\
count	201.000000	201.00000	201.000000	201.000000	201.000000	
mean	0.840796	122.00000	98.797015	0.837102	0.915126	
std	1.254802	31.99625	6.066366	0.059213	0.029187	
min	-2.000000	65.00000	86.600000	0.678039	0.837500	
25%	0.000000	101.00000	94.500000	0.801538	0.890278	
50%	1.000000	122.00000	97.000000	0.832292	0.909722	
75%	2.000000	137.00000	102.400000	0.881788	0.925000	

```
max      3.000000          256.00000  120.900000    1.000000    1.000000

           height  curb-weight  engine-size        bore      stroke  \
count  201.000000   201.000000   201.000000  201.000000  197.000000
mean    53.766667  2555.666667   126.875622    3.330692    3.256904
std      2.447822   517.296727    41.546834    0.268072    0.319256
min     47.800000  1488.000000    61.000000    2.540000    2.070000
25%     52.000000  2169.000000    98.000000    3.150000    3.110000
50%     54.100000  2414.000000   120.000000    3.310000    3.290000
75%     55.500000  2926.000000   141.000000    3.580000    3.410000
max     59.800000  4066.000000   326.000000    3.940000    4.170000

       compression-ratio  horsepower     peak-rpm    city-mpg  highway-mpg  \
count         201.000000  201.000000   201.000000  201.000000   201.000000
mean           10.164279  103.405534  5117.665368   25.179104    30.686567
std             4.004965   37.365700   478.113805    6.423220     6.815150
min             7.000000   48.000000  4150.000000   13.000000    16.000000
25%             8.600000   70.000000  4800.000000   19.000000    25.000000
50%             9.000000   95.000000  5125.369458   24.000000    30.000000
75%             9.400000  116.000000  5500.000000   30.000000    34.000000
max            23.000000  262.000000  6600.000000   49.000000    54.000000

              price  city-L/100km      diesel         gas
count    201.000000    201.000000  201.000000  201.000000
mean   13207.129353      9.944145    0.099502    0.900498
std     7947.066342      2.534599    0.300083    0.300083
min     5118.000000      4.795918    0.000000    0.000000
25%     7775.000000      7.833333    0.000000    1.000000
50%    10295.000000      9.791667    0.000000    1.000000
75%    16500.000000     12.368421    0.000000    1.000000
max    45400.000000     18.076923    1.000000    1.000000
```

[17]:
```
df.describe(include=["object"])
```

[17]:
```
          make aspiration num-of-doors body-style drive-wheels  \
count      201        201          201        201          201
unique      22          2            2          5            3
top     toyota        std         four      sedan          fwd
freq        32        165          115         94          118

       engine-location engine-type num-of-cylinders fuel-system  \
count              201         201              201         201
unique               2           6                7           8
top              front         ohc             four        mpfi
freq               198         145              157          92

       horsepower-binned
```

```
count                 200
unique                  3
top                   Low
freq                  115
```

[18]:
```
df["drive-wheels"].value_counts()
```

[18]:
```
drive-wheels
fwd    118
rwd     75
4wd      8
Name: count, dtype: int64
```

[19]:
```
df["drive-wheels"].value_counts().to_frame()
```

[19]:
```
              count
drive-wheels
fwd             118
rwd              75
4wd               8
```

[20]:
```
drive_wheels_counts = df["drive-wheels"].value_counts().to_frame()
drive_wheels_counts.rename(columns={"drivewheels": "value_counts"},␣
 ↪inplace=True)
drive_wheels_counts
```

[20]:
```
              count
drive-wheels
fwd             118
rwd              75
4wd               8
```

[21]:
```
drive_wheels_counts.index.name = "drive-wheels"
drive_wheels_counts
```

[21]:
```
              count
drive-wheels
fwd             118
rwd              75
4wd               8
```

[22]:
```
# engine-location as variable
engine_loc_counts = df["engine-location"].value_counts().to_frame()
engine_loc_counts.rename(columns={"enginelocation": "value_counts"},␣
 ↪inplace=True)
engine_loc_counts.index.name = "engine-location"
engine_loc_counts.head(10)
```

[22]:
```
                  count
engine-location
front               198
rear                  3
```

[23]:
```
df["drive-wheels"].unique()
```

[23]:
```
array(['rwd', 'fwd', '4wd'], dtype=object)
```

[24]:
```
df_group_one = df[["drive-wheels", "price"]]
df_group_one
```

[24]:
```
    drive-wheels    price
0            rwd  13495.0
1            rwd  16500.0
2            rwd  16500.0
3            fwd  13950.0
4            4wd  17450.0
..           …      …
196          rwd  16845.0
197          rwd  19045.0
198          rwd  21485.0
199          rwd  22470.0
200          rwd  22625.0

[201 rows x 2 columns]
```

[25]:
```
df_group_one = df_group_one.groupby(["drive-wheels"], as_index=False).mean()
df_group_one
```

[25]:
```
  drive-wheels         price
0          4wd  10241.000000
1          fwd   9244.779661
2          rwd  19757.613333
```

[26]:
```
# grouping results
df_gptest = df[["drive-wheels", "body-style", "price"]]
grouped_test1 = df_gptest.groupby(["drive-wheels", "body-style"],␣
 ↪as_index=False).mean()
grouped_test1
```

[26]:
```
   drive-wheels   body-style         price
0           4wd    hatchback   7603.000000
1           4wd        sedan  12647.333333
2           4wd        wagon   9095.750000
3           fwd  convertible  11595.000000
4           fwd      hardtop   8249.000000
```

```
5           fwd    hatchback   8396.387755
6           fwd        sedan   9811.800000
7           fwd        wagon   9997.333333
8           rwd  convertible  23949.600000
9           rwd      hardtop  24202.714286
10          rwd    hatchback  14337.777778
11          rwd        sedan  21711.833333
12          rwd        wagon  16994.222222
```

[27]:
```
grouped_pivot = grouped_test1.pivot(index="drive-wheels", columns="body-style")
grouped_pivot
```

[27]:
```
                  price                                            \
body-style   convertible       hardtop     hatchback         sedan
drive-wheels
4wd                  NaN           NaN   7603.000000  12647.333333
fwd              11595.0   8249.000000   8396.387755   9811.800000
rwd              23949.6  24202.714286  14337.777778  21711.833333


body-style           wagon
drive-wheels
4wd            9095.750000
fwd            9997.333333
rwd           16994.222222
```

[28]:
```
grouped_pivot = grouped_pivot.fillna(0)
grouped_pivot
```

[28]:
```
                  price                                            \
body-style   convertible       hardtop     hatchback         sedan
drive-wheels
4wd                  0.0      0.000000   7603.000000  12647.333333
fwd              11595.0   8249.000000   8396.387755   9811.800000
rwd              23949.6  24202.714286  14337.777778  21711.833333


body-style           wagon
drive-wheels
4wd            9095.750000
fwd            9997.333333
rwd           16994.222222
```

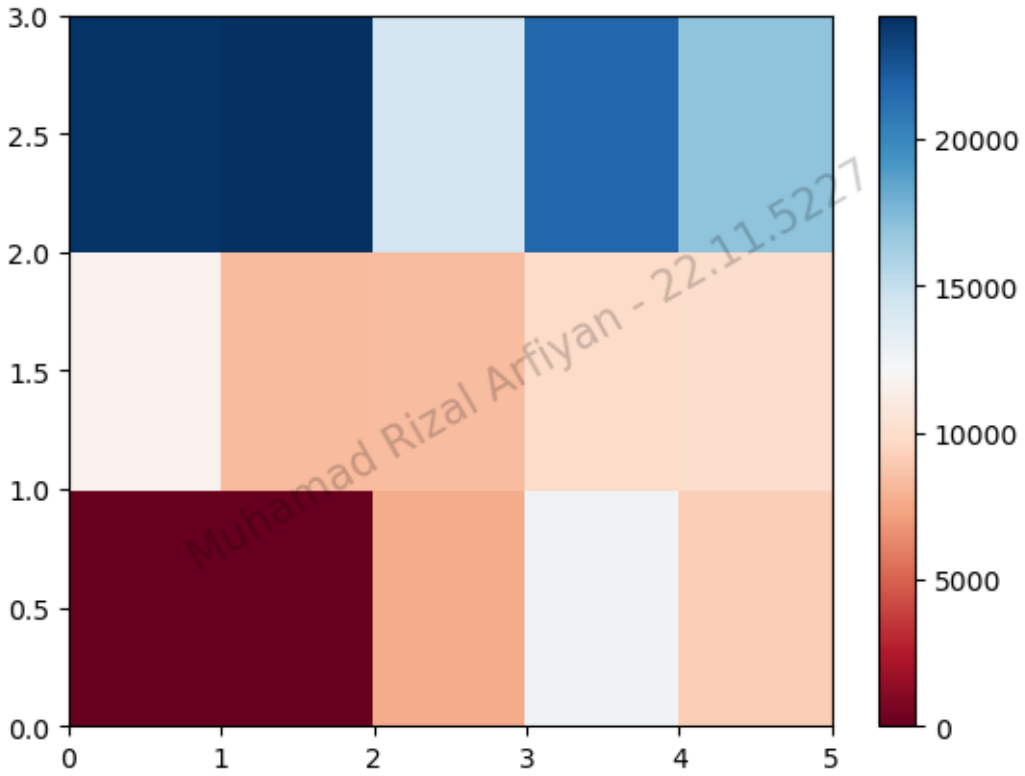
[29]:
```
# Write your code below and press Shift+Enter to execute
df_gptest2 = df[["body-style", "price"]]
grouped_test_bodystyle = df_gptest2.groupby(["body-style"], as_index=False).
 ↪mean()
```

```
grouped_test_bodystyle
```

[29]:

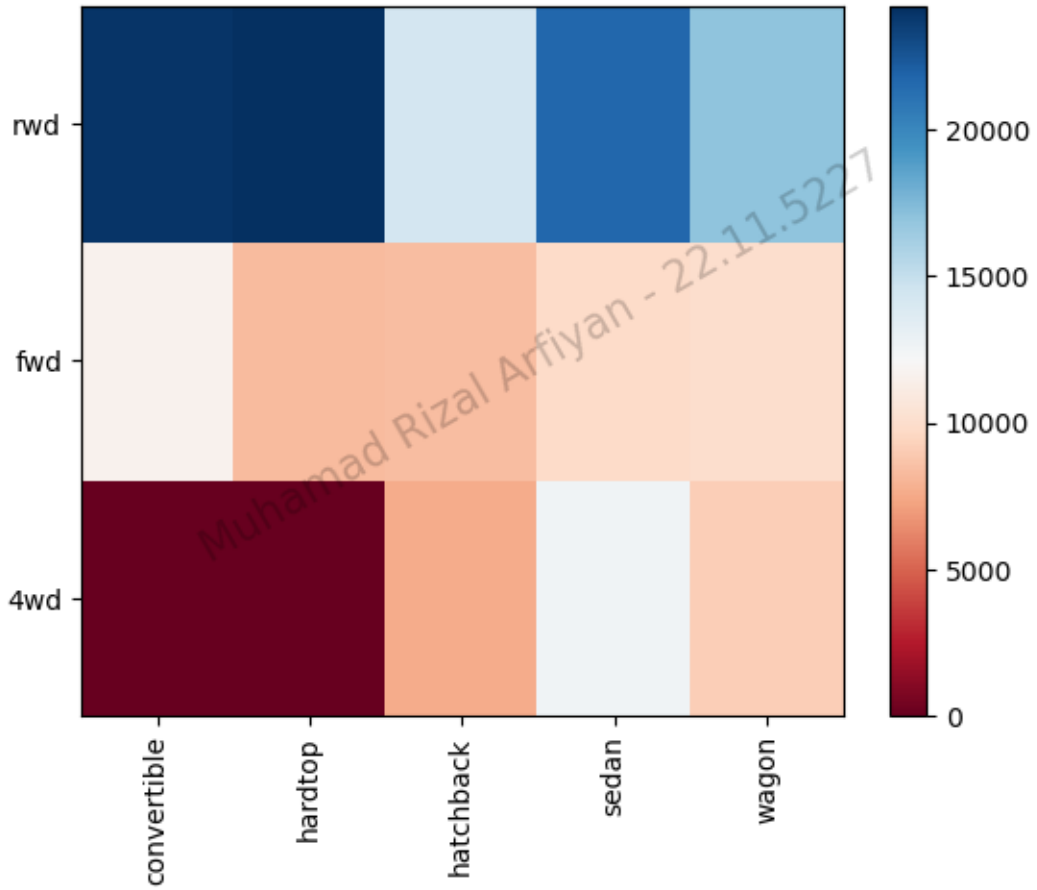
	body-style	price
0	convertible	21890.500000
1	hardtop	22208.500000
2	hatchback	9957.441176
3	sedan	14459.755319
4	wagon	12371.960000

[30]:
```
import matplotlib.pyplot as plt

# use the grouped results
plt.pcolor(grouped_pivot, cmap="RdBu")
plt.colorbar()
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=16,
    color="black",
    ha="center",
    va="center",
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
plt.show()
```



[31]:
```
fig, ax = plt.subplots()
im = ax.pcolor(grouped_pivot, cmap="RdBu")
# label names
row_labels = grouped_pivot.columns.levels[1]
col_labels = grouped_pivot.index
# move ticks and labels to the center
ax.set_xticks(np.arange(grouped_pivot.shape[1]) + 0.5, minor=False)
ax.set_yticks(np.arange(grouped_pivot.shape[0]) + 0.5, minor=False)
# insert labels
ax.set_xticklabels(row_labels, minor=False)
ax.set_yticklabels(col_labels, minor=False)
# rotate label if too long
plt.xticks(rotation=90)
fig.colorbar(im)
plt.text(
    0.5,
    0.5,
    "Muhamad Rizal Arfiyan - 22.11.5227",
    fontsize=16,
    color="black",
    ha="center",
    va="center",
```

```
    alpha=0.2,
    transform=plt.gcf().transFigure,
    rotation=30,
)
plt.show()
```



[32]:
```
grouped_test2 = df_gptest[["drive-wheels", "price"]].groupby(["drive-wheels"])
grouped_test2.head(2)
```

[32]:

	drive-wheels	price
0	rwd	13495.0
1	rwd	16500.0
3	fwd	13950.0
4	4wd	17450.0
5	fwd	15250.0
136	4wd	7603.0

[33]:
```
df_gptest
```

[33]:
```
    drive-wheels   body-style    price
0            rwd  convertible  13495.0
1            rwd  convertible  16500.0
2            rwd    hatchback  16500.0
3            fwd        sedan  13950.0
4            4wd        sedan  17450.0
..           …          …      …
196          rwd        sedan  16845.0
197          rwd        sedan  19045.0
198          rwd        sedan  21485.0
199          rwd        sedan  22470.0
200          rwd        sedan  22625.0

[201 rows x 3 columns]
```

[34]:
```
grouped_test2.get_group("4wd")["price"]
```

[34]:
```
4      17450.0
136     7603.0
140     9233.0
141    11259.0
144     8013.0
145    11694.0
150     7898.0
151     8778.0
Name: price, dtype: float64
```

[35]:
```
f_val, p_val = stats.f_oneway(
    grouped_test2.get_group("fwd")["price"],
    grouped_test2.get_group("rwd")["price"],
    grouped_test2.get_group("4wd")["price"],
)
print("ANOVA results: F=", f_val, ", P =", p_val)
```

```
ANOVA results: F= 67.95406500780399 , P = 3.3945443577151245e-23
```

[36]:
```
f_val, p_val = stats.f_oneway(
    grouped_test2.get_group("fwd")["price"], grouped_test2.
  ↪get_group("rwd")["price"]
)
print("ANOVA results: F=", f_val, ", P =", p_val)
```

```
ANOVA results: F= 130.5533160959111 , P = 2.2355306355677845e-23
```

[39]:
```
f_val, p_val = stats.f_oneway(
    grouped_test2.get_group("4wd")["price"], grouped_test2.
  ↪get_group("fwd")["price"]
```

```
)
print("ANOVA results: F=", f_val, ", P =", p_val)
```

```
ANOVA results: F= 0.665465750252303 , P = 0.41620116697845666
```