Homework 5

Due: Monday, Dec 2, at 11:59pm via Blackboard

All statistical tests are significant if the p-values are less than aplha of 0.05

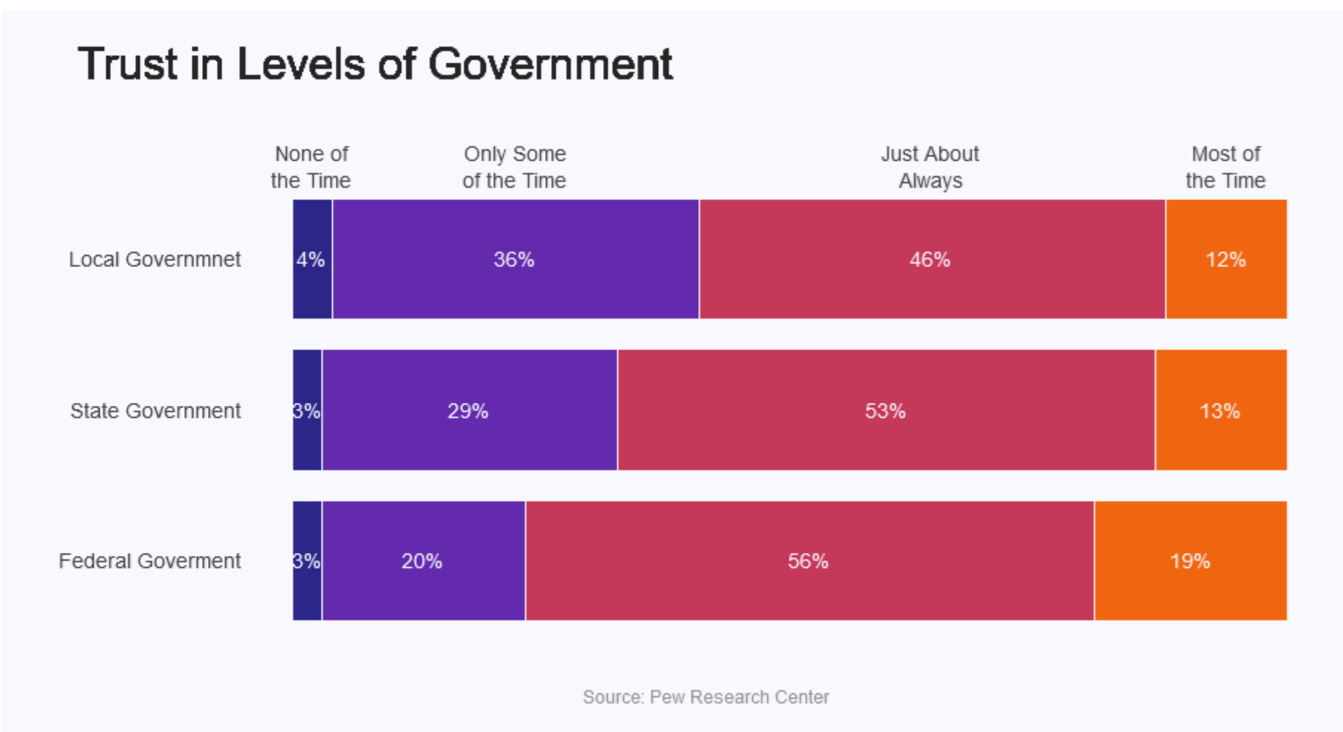
In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime  # to access datetime
import scipy.stats as stats

import plotly.express as px # for interactive plotting
import plotly.graph_objects as go # for interactive plotting

# set the graphics style initially to defaul
plt.style.use('default')

```

In [ ]:

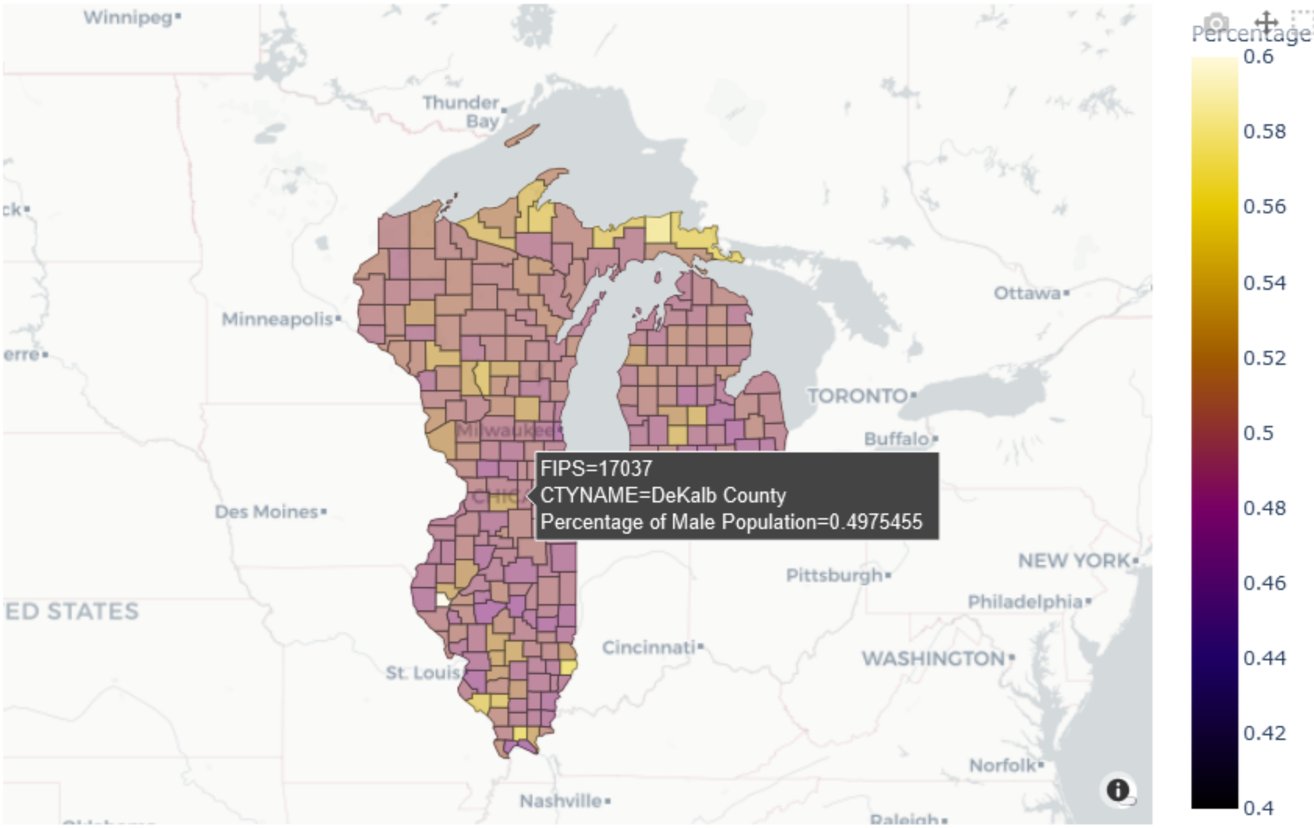
```

```

Q1. The stacked bar graph below shows the results of Pew Research Center's study on Trust in different levels of Government by the American public. Using plotly graph objects, re-create the bar graph below, but using the Seaborn palette 'CMRmap'. (3 points)



Q2.The plot below shows the percentage of male population by counties, only for the states of Illinois, Michigan and Wisconsin. Import the csv file 'population' and using plotly's choropleth mapbox function, re-create the plot below. Adjust the hower data to show the name of the counties and label to show "Percentage of Male

Population." Use the "Inferno" color scale and adjust the color range from 0.4 to 0.6. Hint: you need to create a new variable that divides male population by the total population by county. (3 points)



In [62]:
```
1
```

Out[62]:

	Unnamed: 0	FIPS	STNAME	CTYNAME	TOT_POP	TOT_MALE	TOT_FEMALE	WA_MALE	WA_FEMALE	NHWA_
0	0	18049	Indiana	Fulton County	20737	10369	10368	9985	10020	
1	1	18051	Indiana	Gibson County	33458	16642	16816	15873	16117	
2	2	18053	Indiana	Grant County	69330	33282	36048	29587	32460	
3	3	18055	Indiana	Greene County	32940	16479	16461	16179	16167	
4	4	18057	Indiana	Hamilton County	289495	141103	148392	125675	131785	

In [76]:

Out[76]:

	Unnamed: 0	FIPS	STNAME	CTYNAME	TOT_POP	TOT_MALE	TOT_FEMALE	WA_MALE	WA_FEMALE	NHWA_
0	0	18049	Indiana	Fulton County	20737	10369	10368	9985	10020	
1	1	18051	Indiana	Gibson County	33458	16642	16816	15873	16117	
2	2	18053	Indiana	Grant County	69330	33282	36048	29587	32460	
3	3	18055	Indiana	Greene County	32940	16479	16461	16179	16167	
4	4	18057	Indiana	Hamilton County	289495	141103	148392	125675	131785	

In [ ]:

Q3: The Excel file "ConSpendA" shows the consumer spending patterns (Sales) by several variables including gender, when the purchase was made (day), payment method type. Import the file.

In [7]:

```
ConSpend = pd.read_excel('ConSpendA.xlsx',parse_dates=['Date'],index_col='Date')
ConSpend.head()
```

Out[7]:

Date	Day	Time	Region	Paid With	Gender	Items Ordered	Sales
2013-03-10	Sunday	Morning	West	VISA	Female	4	136.97
2013-03-10	Sunday	Morning	West	Mastercard	Female	1	25.55
2013-03-10	Sunday	Afternoon	West	VISA	Female	5	113.95
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	1	6.82
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	5	142.15

In [8]:

```
ConSpend.index = pd.to_datetime(ConSpend.index)
ConSpend.head()
```

Out[8]:

Date	Day	Time	Region	Paid With	Gender	Items Ordered	Sales
2013-03-10	Sunday	Morning	West	VISA	Female	4	136.97
2013-03-10	Sunday	Morning	West	Mastercard	Female	1	25.55
2013-03-10	Sunday	Afternoon	West	VISA	Female	5	113.95
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	1	6.82
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	5	142.15

Q3a. Create new variables that show the Year and Month of Purchases and add it to the dataframe (2 points)

In [9]:

Q3b: Identify any missing values in the dataframe (1 point)

In [107]:

Out[107]:

```
Day              0
Time             0
Region           0
Paid With        6
Gender           3
Items Ordered    0
Sales            6
Year             0
Month            0
dtype: int64
```

Q3c. Drop all missing values and create a new dataframe Spend3 (1 point)

In [13]:

Out[13]:

	Day	Time	Region	Paid With	Gender	Items Ordered	Sales	Year	Month
Date									
2013-03-10	Sunday	Morning	West	VISA	Female	4	136.97	2013	3
2013-03-10	Sunday	Morning	West	Mastercard	Female	1	25.55	2013	3
2013-03-10	Sunday	Afternoon	West	VISA	Female	5	113.95	2013	3
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	1	6.82	2013	3
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	5	142.15	2013	3

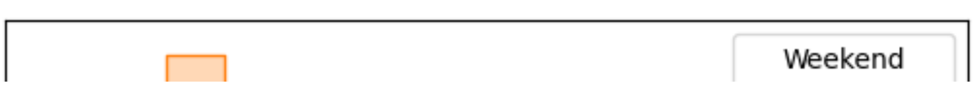
Q3d. We are interested to see if there is a difference between weekend and weekday spenings. Using a List Comprehension, create a new categorial variable "Weekend" that classifies the "day" into weekend if its Friday, Saturday or Sunday, and weekday otherwise. (3 points)

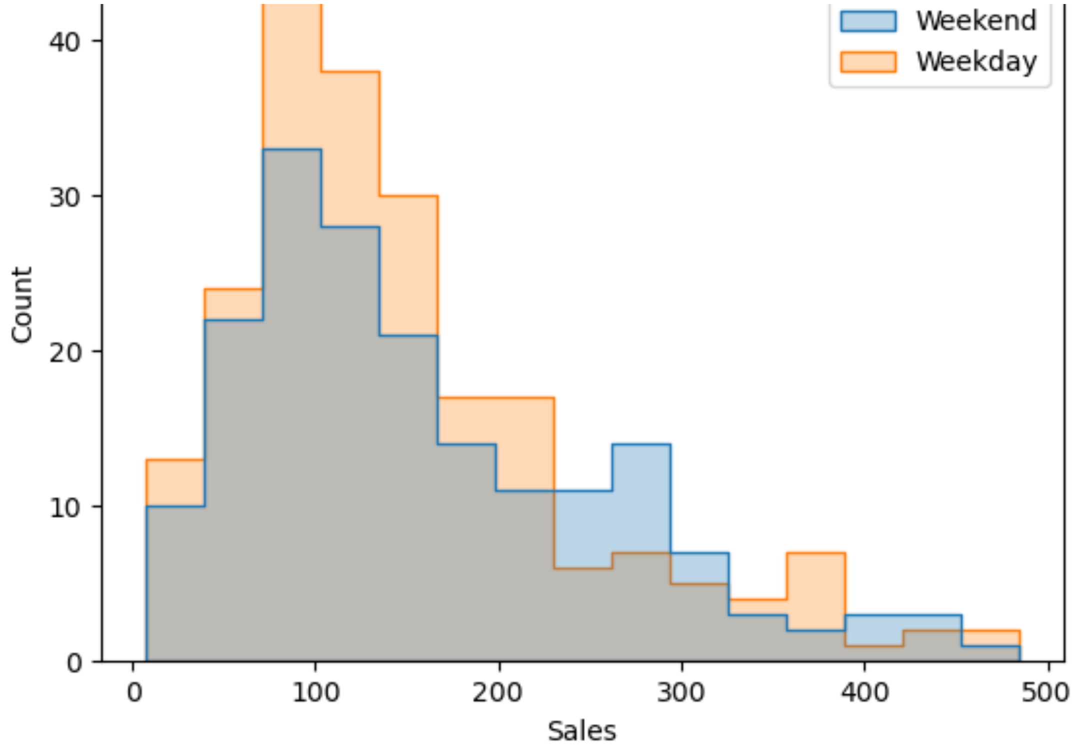
In [15]:

Out[15]:

	Day	Time	Region	Paid With	Gender	Items Ordered	Sales	Year	Month	Weekend
Date										
2013-03-10	Sunday	Morning	West	VISA	Female	4	136.97	2013	3	Weekend
2013-03-10	Sunday	Morning	West	Mastercard	Female	1	25.55	2013	3	Weekend
2013-03-10	Sunday	Afternoon	West	VISA	Female	5	113.95	2013	3	Weekend
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	1	6.82	2013	3	Weekend
2013-03-10	Sunday	Afternoon	NorthEast	VISA	Female	5	142.15	2013	3	Weekend

Q3e. Using Seaborn, create a histogram showing Sales for weekend and weekdays, with a transparanecy of 0.3 (2 points)

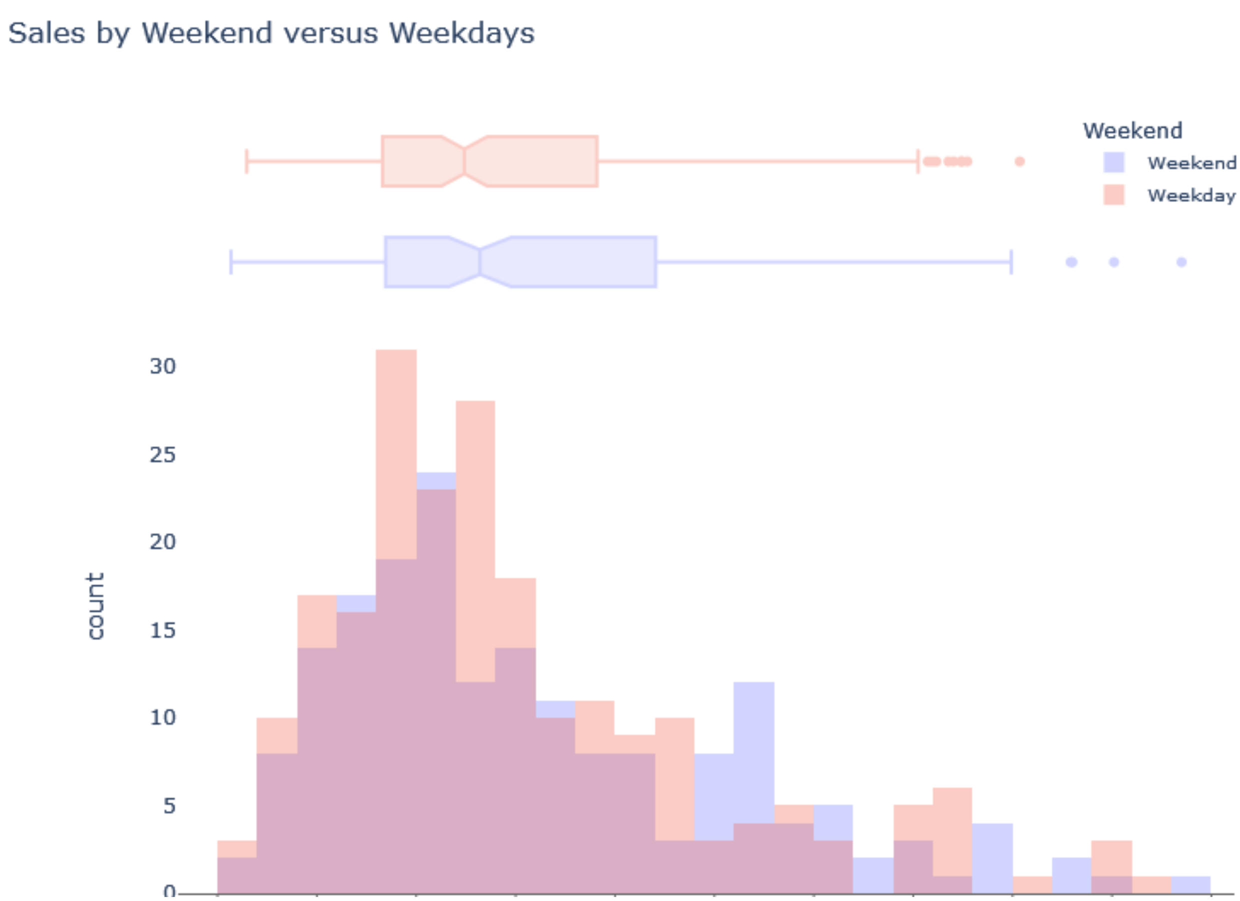


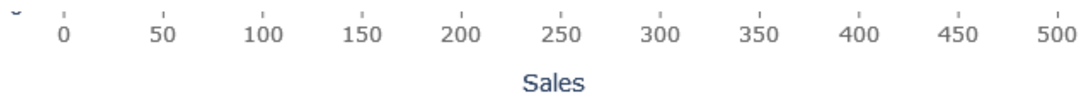



```
In [ ]:  1
```

Q3f. Using plotly, create overlapping histograms that show the sales by weekdays and weekends, with an opacity of 0,3. Pass the 'marginal' argument into the function to also show a "box" plot." Also, set x-axis ticks to '50' and the plot background color to white (3 points).

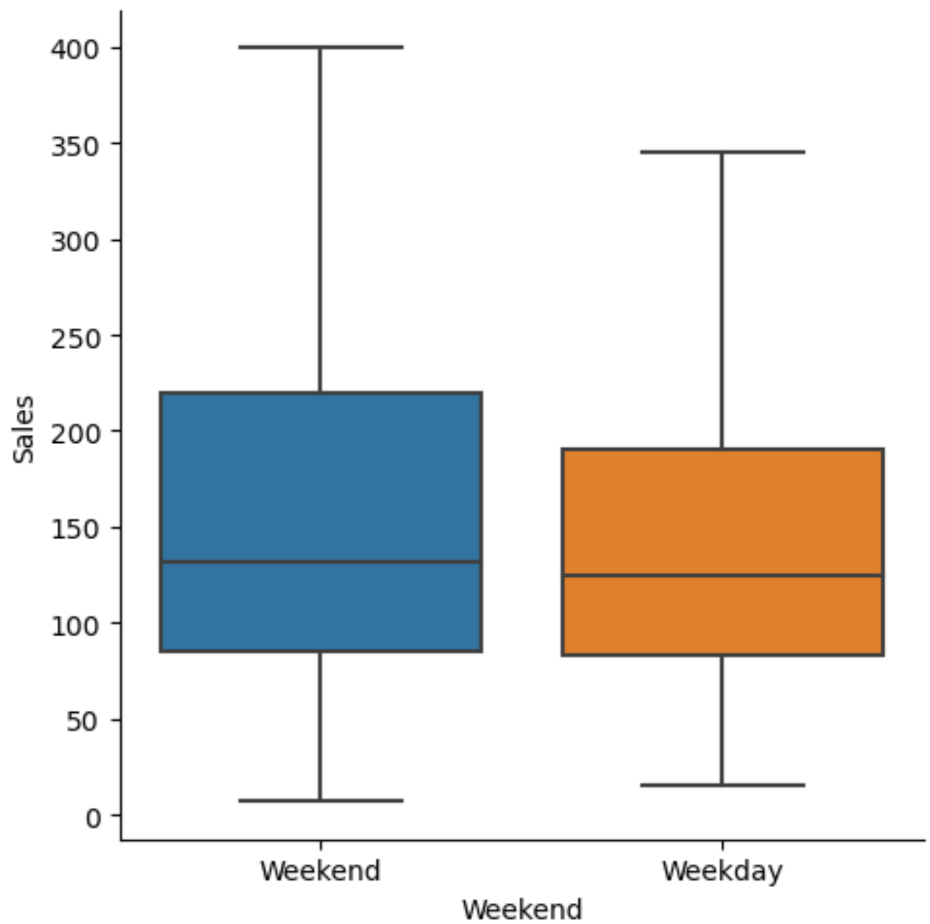
```
In [ ]:  1
```





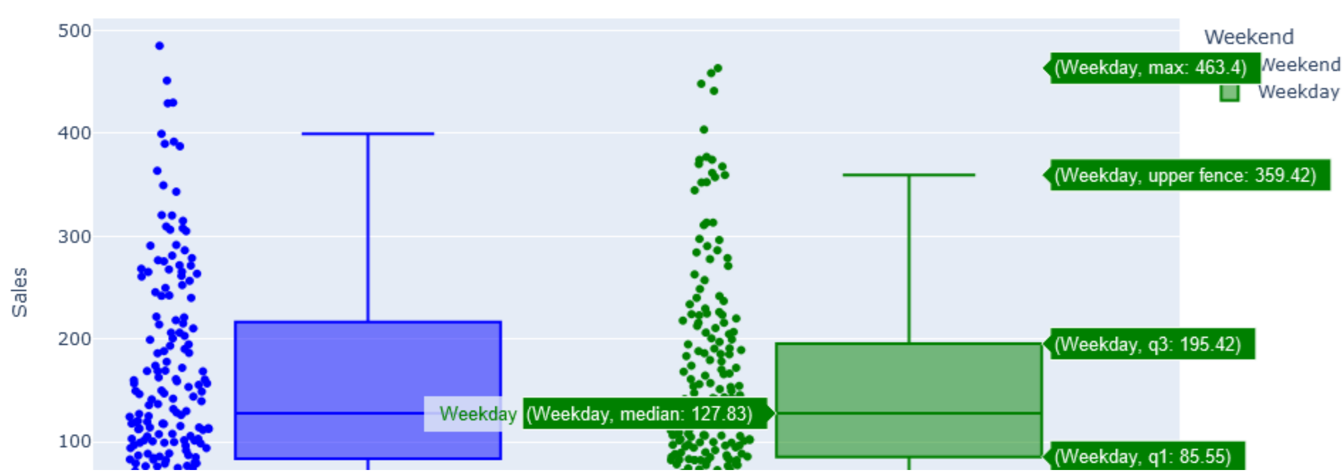
Q3g. Using Seaborn, create boxplots showing Sales for weekend and weekdays and eliminated the outliers (2 points)

In [ ]:
```
1
```



Q3h. Using plotly, create box plots to show weekend versu weekday sales, differentiated by color (green and blue). Also show the distribution of data points on the plot (2 points).

In [ ]:
```
1
```



Q3i. Create a cross-tabulation table that shows total sales by Month and Gender (1 point)

In [124]:

```
```

Out[124]:

Month	3	4	5	6
Gender				
Female	42	63	59	62
Male	31	39	41	48

Q3j. # Is Gender and Spending across different months independent? Run a Chi-Sq test and explain your results (3 points)

In [125]:

```
```

```
Chi-square Statistic: 0.6874917057402256
p-value: 0.8761420083436133
Degrees of Freedom: 3
Expected Frequencies:
[[42.85194805 59.87532468 58.7012987  64.57142857]
 [30.14805195 42.12467532 41.2987013  45.42857143]]
```

Q3k. Generate the average sales by months. (1 point)

In [18]:

```
```

Out[18]:

```
Month
3     97.750959
4    147.820098
5    170.633800
6    185.024000
Name: Sales, dtype: float64
```

In [ ]:

```
```

Q3l. Is there a statitsical difference between the average sales in March (3) versus June(6)? Run a two-sample test for difference in population means. Explain your statistical results. (2 points)

In [33]:

```
```

In [127]:

Out[127]:

```
Ttest_indResult(statistic=-6.526635003619645, pvalue=6.54981628491037e-10)
```

Q4a. Import the datafile audi.csv and bmw.csv, then create the Dataframes audiSales and bmwSales. Add a 'make' column to the bmwSales and audiSales DataFrames to show the make of the car, either "BMW' or 'Audi." Then concatenate both Dataframes, naming the new dataframe CBSales2 (2 points)

In [18]:

Out[18]:

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	2.0
1	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2.0
2	5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	3.0
3	1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	1.5
4	7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	3.0

In [19]:

Out[19]:

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
1	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2.0
2	A1	2016	11000	Manual	29946	Petrol	30	55.4	1.4
3	A4	2017	16800	Automatic	25952	Diesel	145	67.3	2.0
4	A3	2019	17300	Manual	1998	Petrol	145	49.6	1.0

In [20]:

Out[20]:

		model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	make
BMW	**0**	5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	2.0	BMW
	1	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2.0	BMW
	2	5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	3.0	BMW
	3	1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	1.5	BMW
	4	7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	3.0	BMW

Q4b. Create a function (Eff) to define a new categorical variable (Efficiency) with two levels: Efficient if the mpg is greater than 35 (mpg > 35), and Inefficient otherwise (mpg <=35) and apply it to the concatenated dataframe CBSales2 (2 points).

In [21]:

```

```

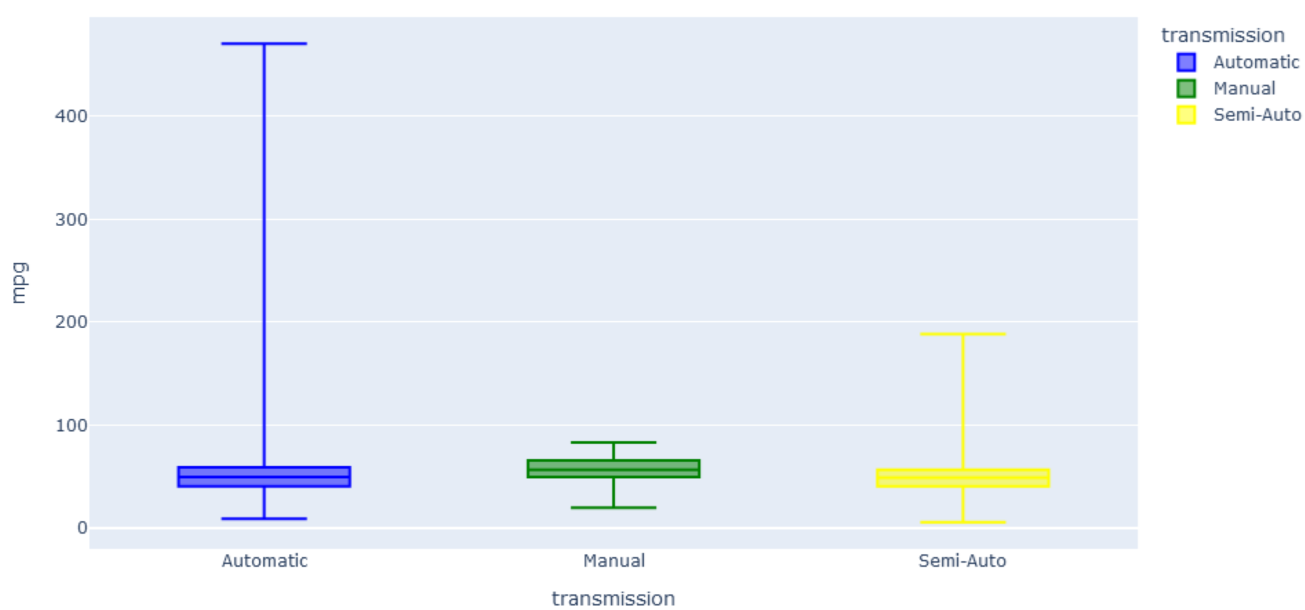
Out[21]:

		model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	make	Efficiency
BMW	**0**	5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	2.0	BMW	Efficient
	1	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2.0	BMW	Efficient
	2	5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	3.0	BMW	Efficient
	3	1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	1.5	BMW	Efficient
	4	7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	3.0	BMW	Efficient

Q4c. Using Plotly, create a box plot to show transmission (x-axis) and mpg(y-axis) and differentiated by transmission, and exclude the outliers (2 points)

In [ ]:

```

```



Q4d. create a cross-tabulation table of Transmission by Efficiency (1 point)

In [22]:

Out[22]:

Efficiency	Efficient	Inefficient
transmission		
Automatic	5596	700
Manual	6839	57
Semi-Auto	7211	1046

Q4e.Is transmission type and Efficiency independent? Run a Chi-Sq Test and explain your statistical results (2 points)

In [23]:

```
Chi-square Statistic: 769.4917033889279
p-value: 8.07234424169003e-168
Degrees of Freedom: 2
Expected Frequencies:
[[5766.75910299  529.24089701]
 [6316.32318523  579.67681477]
 [7562.91771178  694.08228822]]
```

Q4f. create a cross-tabulation table of Efficiency by Make of vehicle. (1 point)

In [12]:

Out[12]:

make	AUDI	BMW
Efficiency		
Efficient	9605	10041
Inefficient	1063	740

Q4g.Is Vehicle make and Efficiency independent? Run a Chi-Sq Test and explain your statistical results (2 points)

In [14]:

```
Chi-square Statistic: 66.54463773261105
p-value: 3.420593086046886e-16
Degrees of Freedom: 1
Expected Frequencies:
[[9771.24938226 9874.75061774]
 [ 896.75061774  906.24938226]]
```

Q4h. Extract the data for BMW or Audi cars and with model years of 2018 or 2019 and store to a new dataframe SalesTTL2 (2 points)

In [6]:

Out[6]:

		model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	make	Efficiency
BMW	**1**	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2.0	BMW	Efficient
	7	2 Series	2018	16250	Manual	10401	Petrol	145	52.3	1.5	BMW	Efficient
	26	3 Series	2019	17800	Automatic	22310	Diesel	145	64.2	2.0	BMW	Efficient
	39	1 Series	2018	14600	Automatic	6522	Petrol	145	37.2	1.5	BMW	Efficient
	43	1 Series	2018	17500	Automatic	14037	Petrol	145	54.3	1.5	BMW	Efficient

Q4i. Create a cross-tabulation table of Efficiency by Make of vehicle for the 2018 and 2019 models. (1 point)

In [7]:

Out[7]:

make	AUDI	BMW
Efficiency		
Efficient	3361	3872
Inefficient	700	461

Q4j.Is Vehicle male and Efficiency independent? Run a Chi-Sq Test and explain your statistical results (2 points)

In [8]:

```
Chi-square Statistic: 76.01506057535607
p-value: 2.815094806205939e-18
Degrees of Freedom: 1
Expected Frequencies:
[[3499.31057898 3733.68942102]
 [ 561.68942102  599.31057898]]
```

Q4k. Generate the mean mpg for the 2018 and 2019 BMW and Audi cars (1 point)

In [9]:

Out[9]:

```
make
AUDI     43.816031
BMW      51.140642
Name: mpg, dtype: float64
```

Q4l. Is there a statistical difference in the fuel efficiency of BMW cars different than those of Audi? Run a two sample test for difference in population means and explain results (2 points)

In [11]:

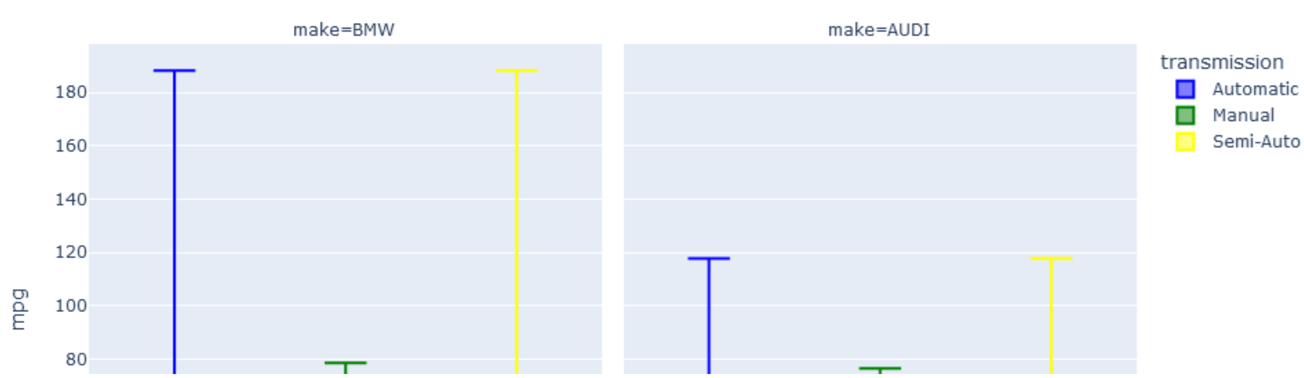
In [12]:

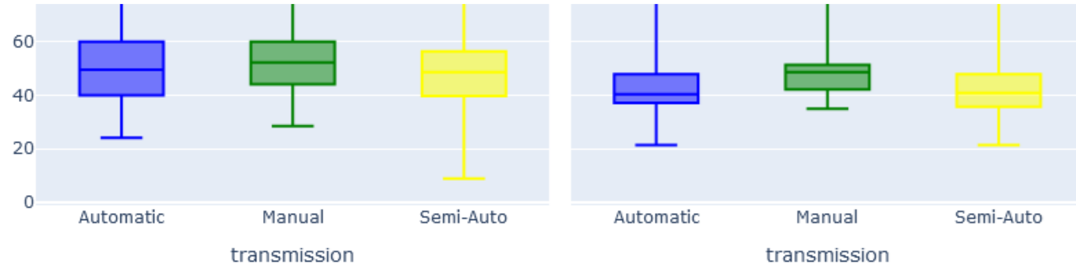
Out[12]:

```
Ttest_indResult(statistic=-24.048657193522846, pvalue=1.225620237555171e-123)
```

Q4m. Using Plotly, create a box plots to show transmission (x-axis) and mpg(y-axis) and differentiated by transmission, and exclude the outliers. Show the plots separately for Audi and BMW (3 points)

In [ ]:





Q4n. Generate the average mpg for Audi and BMW cars by transmission (1 point)

In [13]:

Out[13]:

```
transmission  make
Automatic     AUDI    42.231995
              BMW     53.051702
Manual        AUDI    48.120435
              BMW     53.004545
Semi-Auto     AUDI    42.367948
              BMW     49.689201
Name: mpg, dtype: float64
```

Q4p. Are the fuel efficiency (mpg) of BMW semi-automatic cars different than that of Audi cars? Is there a statistical difference in the fuel efficiency (mpg) of BMW semi-automatic cars and than those of Audi? Run a two sample test for difference in population means and explain results (4 points)

In [15]:

In [42]:

Out[42]:

```
Ttest_indResult(statistic=-18.20654467820251, pvalue=2.4769702471595843e-71)
```

Q4q. For the dataframe CBSales2, fit a regression line with mileage and engineSize as the independent variables and mpg as the dependent variable. (3 points) Report your regression equation.

In [93]:

```
1
2
3
4
5
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.140
Model:                            OLS   Adj. R-squared:                  0.140
Method:                 Least Squares   F-statistic:                     1748.
Date:                Fri, 08 Nov 2024   Prob (F-statistic):               0.00
Time:                        22:30:34   Log-Likelihood:                -97145.
No. Observations:               21449   AIC:                         1.943e+05
Df Residuals:                   21446   BIC:                         1.943e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         76.1964      0.571    133.471      0.000      75.077      77.315
mileage        0.0002   6.29e-06     30.748      0.000       0.000       0.000
engineSize   -13.3996      0.260    -51.588      0.000     -13.909     -12.890
==============================================================================
Omnibus:                    35668.111   Durbin-Watson:                   1.882
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         30908979.117
Skew:                          11.286   Prob(JB):                         0.00
Kurtosis:                     187.596   Cond. No.                     1.42e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.42e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Regression Equation:

In [ ]:

```
1
```