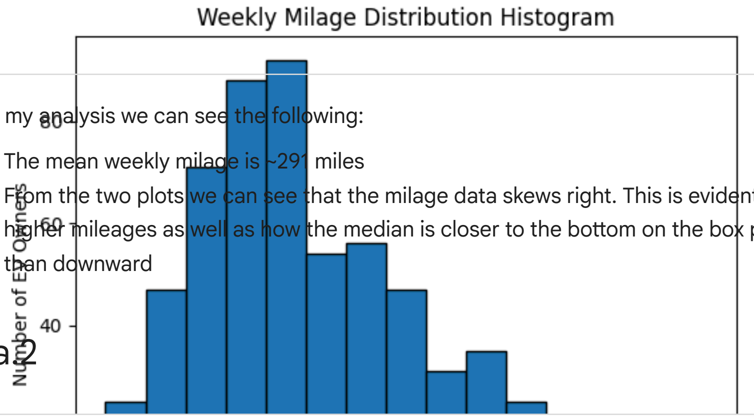
Question 1

Q1a.1

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

EV_df = pd.read_excel("/content/drive/MyDrive/CEE 3040/HW3/EVmileageSample.xlsx")

# Compute and report descriptive statistics for weekly mileage
EV_data = EV_df.describe()
print(EV_data)

# Histogram
fig, ax = plt.subplots()
plt.hist(EV_df['WeekMiles'], bins=15, edgecolor='black')
ax.set_xlabel('Miles per Week')
ax.set_ylabel('Number of EV Owners')
plt.title("Weekly Milage Distribution Histogram")
plt.show()

# Box Plot
sns.boxplot(data=EV_df.WeekMiles, width=0.35, whis=1.5);
plt.ylabel("Weekly Milage")
plt.title("Weekly Mileage Distribution Box Plot")
plt.show()
```

```
         WeekMiles
count   624.000000
mean    291.027244
std     166.615466
min      10.000000
25%     168.750000
50%     248.000000
75%     386.000000
max     750.000000
```

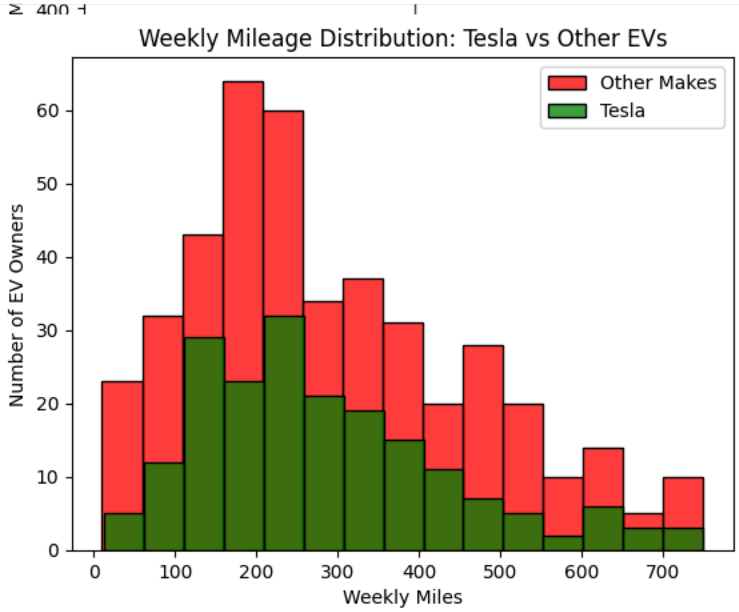


From my analysis we can see the following:

- The mean weekly milage is ~291 miles
- From the two plots we can see that the milage data skews right. This is evident from the histogram's long tail stretching toward higher mileages as well as how the median is closer to the bottom on the box plot and the whisker extends much further upward than downward

Q1a2

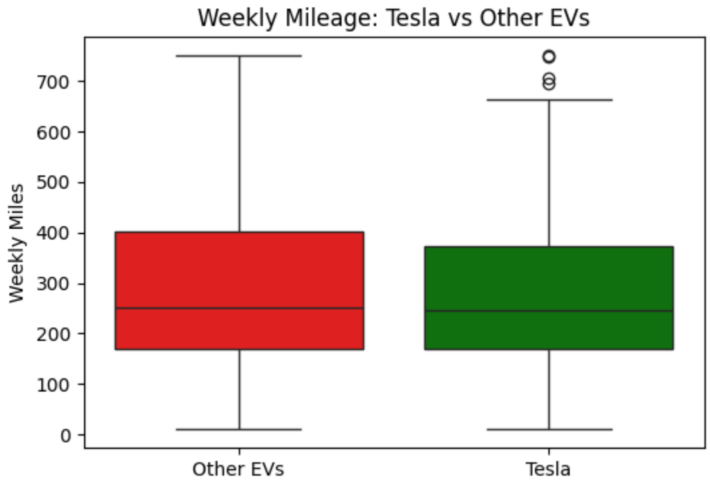
```
tesla_df = EV_df[EV_df["Make"] == "Tesla"]   # all Tesla rows
other_df = EV_df[EV_df["Make"] != "Tesla"]   # all non-Tesla rows

f, ax = plt.subplots()

# Other EV owners (red) — draw first
sns.histplot(other_df["WeekMiles"], bins=15, color="r", label="Other Makes")

# Tesla owners (blue) — draw on top
sns.histplot(tesla_df["WeekMiles"], bins=15, color="g", label="Tesla")

plt.title("Weekly Mileage Distribution: Tesla vs Other EVs")
ax.set_xlabel("Weekly Miles")
ax.set_ylabel("Number of EV Owners")
ax.legend()
plt.show()
```




```
other_df["Category"] = "Other EVs"
tesla_df["Category"] = "Tesla"

# Combine via Panda concat function
plot_df = pd.concat([other_df[["WeekMiles", "Category"]], tesla_df[["WeekMiles", "Category"]]])

# Plot with hue so colors work cleanly
plt.figure(figsize=(6,4))
```

```
sns.boxplot(x="Category", y="WeekMiles", data=plot_df, order=["Other EVs", "Tesla"], hue="Category", palette={"Other E\

plt.title("Weekly Mileage: Tesla vs Other EVs")
plt.ylabel("Weekly Miles")
plt.xlabel("")
plt.show()
```

```
/tmp/ipython-input-1481069656.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
  other_df["Category"] = "Other EVs"
/tmp/ipython-input-1481069656.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
  tesla_df["Category"] = "Tesla"
```



The following differences can be observed between Teslas and other EVs

- From the stacked histogram we can see that Tesla miles are more concentrated around the 200 mi to 300 mi range, while other EV owners are more spread out. This is also evident from the whiskers in the box plots.
- From the box plot we can see that Teslas have a slightly lower weekly milage median than other EVs along with being more tightly clustered as visible from the shorter whiskers and a smaller box.

Q1b

```
# Total number of EV owners
N = len(EV_df)

# Proability Owner Drives a Tesla Model S
T = EV_df['Model'] == 'Model S'
P_T = T.sum()/N

# Probability of Exceeding 300 Miles
M = EV_df['WeekMiles'] > 300
P_M = M.sum()/N

# Intersection of Events
T_AND_M = T & M
P_T_AND_M = T_AND_M.sum() / N

# Union via frequencies
T_OR_M = T | M
P_T_OR_M = T_OR_M.sum() / N

# Addition Rule
P_T_U_M = P_T + P_M - P_T_AND_M

print(f"P(T) = {P_T:.4f}")
print(f"P(M) = {P_M:.4f}")
print(f"P(T ∪ M) via frequencies = {P_T_OR_M:.4f}")
```

```
print(f"P(T ∪ M) via addition rule = {P_T_U_M:.4f}")
```

```
P(T) = 0.0705
P(M) = 0.4263
P(T ∪ M) via frequencies = 0.4760
P(T ∪ M) via addition rule = 0.4760
```

Discussion: From the above printed values we can see that 47.6% of EV owners have a weekly milage that exceed 300 miles, they own a Tesla Model S, or both of those two events. Both the frequency method and the addition rule method resulted in the same answer.

Q1c

```
# Proability EV Owner Drives a Nissan Leaf
L = EV_df['Model'] == 'LEAF'
P_L = L.sum()/N

# Probability Owner Weekly Milage Exceeds 200 Miles
M = EV_df['WeekMiles'] > 200
P_M = M.sum()/N

# Probability of L ∩ M
M_AND_L = M & L
P_M_AND_L = M_AND_L.sum() / N

# Probability of L * M
P_M_AND_L_2 = P_L * P_M

print(f"P(T ∪ M) = {P_M_AND_L:.4f}")
print(f"P(M) x P(L) = {P_M_AND_L_2:.4f}")
```

```
P(T ∪ M) = 0.0064
P(M) x P(L) = 0.0137
```

Discussion: From the above print statments we can see that P(T ∪ M) is less than P(M) x P(L); since the two are not equal we know event A (owner drives a Nissan LEAF) and B (weekly milage exceeds 200 miles) are not independent. Since P(T ∪ M) is less than P(M) x P(L), we know that driving a Nissan LEAF has a negative assosiation with driving more than 200 miles. In other words, they are related because driving a LEAF makes you less likely to exceed 200 miles per week than if the two events were independent.

Q1d

```
# Proability EV Owner Drives a Tesla
T = EV_df['Make'] == 'Tesla'
P_T = T.sum()/N

# Probability Owner Weekly Milage Exceeds 300 Miles
M = EV_df['WeekMiles'] > 300
P_M = M.sum()/N

# Probability of T ∩ M
M_AND_T = M & T
P_M_AND_T = M_AND_T.sum() / N

# Probability P(M|T)
P_M_GIV_T = P_M_AND_T / P_T

print(f"P(M|T) = {P_M_GIV_T:.4f}")
```

```
P(M|T) = 0.4093
```

Discussion: P(M|T) = 0.4093 is interpreted as the probability that an EV driver's weekly milage is greater than 300 miles given that they drive a tesla. Earlier we calculated the probability that any EV owner has a weekly milages of greater than 300 miles which had a value of 0.4263. Thus, since 0.4263 > 0.409, we know that EV owners drive 300 miles weekly more often than the average tesla owner.

Q1e

```
# Proability EV Owner has Tesla
T = EV_df['Make'] == 'Tesla'
P_T = T.sum()/N

# Probability Owner Weekly Milage Exceeds 400 Miles
M = EV_df['WeekMiles'] > 400
P_M = M.sum()/N

# Probability P(M|T)
M_AND_T = M & T
P_M_AND_T = M_GIV_T.sum() / N
P_M_GIV_T = P_M_AND_T / P_T


# Probability P(T|M) - Bayes' Theorem
P_T_GIV_M = (P_M_GIV_T * P_T)/P_M

print(f"P(T|M) = {P_T_GIV_M:.4f}")
```

```
P(T|M) = 0.2568
```

Discussion: From the above printed P(T|M) value we can see that there is only a 25.68% probability that an EV is a Tesla given that its weekly milage is greater than 400 miles. From this we can infer than Teslas are not as dominant among the high-mileage EVs.

Question 2

Q2a

Let:

- S = material passes perfect strength test (is actually strong)
- T = imperfect test says strong
- $P(S) = 0.8$
- $P(T \mid S) = 1 - 0.1 = 0.9$

We want $P(S \cap T)$:

$$P(S \cap T) = P(S) \cdot P(T \mid S) = (0.8)(0.9) = \mathbf{0.72}$$

Q2b

Per the textbook:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} = \frac{P(B \mid A)P(A)}{P(B \mid A)P(A) + P(B \mid \bar{A})P(\bar{A})}$$

And tailored for our problem:

$$P(S \mid T) = \frac{P(T \mid S)P(S)}{P(T \mid S)P(S) + P(T \mid \bar{S})P(\bar{S})}$$

$$P(S \mid T) = \frac{(0.9)(0.8)}{(0.9)(0.8) + (0.08)(.2)} = \mathbf{0.978}$$

Question 3

Q3a

First we will start by determining the probability that the **crack occurs exactly on the 5th cycle**

Let:

- $P(\text{crack development in a cycle}) = C = 0.40$
- $P(\text{no crack development in a cycle}) = Q = 1 - 0.40 = 0.60$

- Inspection A: $P(A \text{ Detects} \mid \text{Crack Exists}) = 0.70$
- Inspection B: $P(B \text{ Detects} \mid \text{Crack Exists}) = 0.80$

Assumption: There is slight ambiguity in the problem statement. I will assume that we are not considering any false positives before the fifth cycle; we are only conditioning on cracks that occur exactly on the fifth cycle.

No cracks in the first 4 cycles AND a crack on the fifth:

$$P(\text{Cracks on 5th Cycle}) = (0.60)^4(0.40) = 0.05184$$

Now we want probability that at least one inspection detects the crack

Assumption: Per the problem statement, if A does not detect a crack, then inspection B is not run. Additionally, if A does detect a crack then per the problem statment, "**at least one** of the inspections" already detected a crack and the event is satisfied. *I will not be calculating the probability that either A detects the crack (and thus inspection B will not be performed per problem statement)* ***or*** *A does not detect and then Inspection B is performed as the problem statment clearly states than inspection B is not run if A does not detect a crack.*

The probability that A detects a crack is:

$$P(A \text{ Detects} \mid \text{Crack Exists}) = 0.70$$

Now if we factor in the probability calculated earlier for the crack in the fifth cycle, we get the final probability where the crack at the fifth cycle is detected by at least one of the inspections:

$$P(\text{Cracks on 5th Cycle AND A Detects}) = P(\text{Cracks on 5th Cycle}) \times P(A \text{ Detects} \mid \text{Crack Exists})$$
$$= (0.05184)(0.70) = \boxed{\mathbf{0.0363}}$$

To summarize: There is a 3.63% joint probability that the crack is exactly on the fifth cycle **and** that the crack is detected by at least one of the two inspections.

Q3b

Now we focus on the probability of a crack at the fifth cycle and it being detected by BOTH inspection

The same assumptions as above apply. The probability that both inspections detect a crack is follows:

$$P(\text{Both Detect} \mid \text{Crack Exists}) = (0.70)(0.80) = 0.56$$

Thus, the probability thatthe crack is exactly on the fifth cycle **AND** that both inspections detect it is:

$$P(\text{Cracks on 5th Cycle AND Both Detect}) = P(\text{Cracks on 5th Cycle}) \times P(\text{Both Detect} \mid \text{Crack Exists})$$
$$= (0.05184)(0.56) = \boxed{\mathbf{0.0290}}$$

To summarize: There is a 2.90% joint probability that the crack is exactly on the fifth cycle **and** that the crack is detected by both of the two inspections.

```
Start coding or generate with AI.
```