Statistics Final Project Report

Team 7: 110612117 張仲瑜 112550159 鄭雯儀

- **Introduction**
  a. Topic: Has the penalty modification of drunk driving been effective in reducing its incident rate in Taiwan?
  b. Motivation: Known for a living hell for pedestrian, traffic accidents has long been a fatal issue in Taiwan. Among various types of them, drunk driving is one of the most hotly discussed topic. Whenever a severe drunk driving accident occurs, people tend to blame for the government, arguing that the punishment is too lenient to effectively deter offenders. Though the government did modify and legislate related laws, horrifying cases seem to be proliferated on news. As a result, we aim to conduct multiple hypothesis tests to examine whether those punishment are effective by comparing the incident rate caused by drunk driving.

- **Data Collection**
  a. The proposed statistical project will analyze trends in drunk driving over a specific period, focusing on the effectiveness of government penalties and regulations. The primary objective is to determine whether the number of drunk driving incidents has increased or decreased over time. Additionally, the project will evaluate the impact of legal reforms on drunk driving behaviors, and explore correlations between drunk driving and injury or death rates.
  b. Data collection will involve gathering incident data (including the annual number of drunk driving cases, which will be sourced from the National Police Agency's statistics. Death and injury data will be collected from the Ministry of Health and Welfare, which will be providing insights in to the human cost of drunk driving. Demographic data on teenage drunk driving will be obtained from the Health Promotion Administration, while information on legal frameworks and penalty updates will be reviewed through government legal database and timelines. These datasets will cover a period of at least 10 years to facilitate longitudinal analysis. Raw data will be downloaded from government portals and categorized by year for trend analysis. However, potential biases may be considered since such underreporting incidents and demographic-specific behaviors. Teenage drunk driving may be underrepresented due to legal and social stigmas, and the immediate impact of policy changes may not be evident due to lag effect.
  c. To ensure comprehensive analysis, the data scope will include nationwide statistics spanning at least a decade. This approach allow robust examination of long-term trends and policy impacts.

- To streamline the data collection process, we have organized the information into five tables. These tables represent a combination of data from all periods and separate tables for each period. The periods are defined as follows:
  - Period 1 (96~102/2): This period focused on increased penalties, including fines ranging from NT$15,000 to NT$60,000, and jail sentences of up to two years for offenders. Penalties for fatal accidents ranged from one to seven years of imprisonment, while severe injuries led to sentences of six months to five years.
  - Period 2 (102/3 ~ 108/6): This period involved the removal of some penalties, with the introduction of new fines (NT$15,000 to NT$90,000) and a clearer alcohol limit. Drivers with a BAC over 0.15 were penalized, and those with a BAC over 0.25 faced criminal charges. Penalties for fatal accidents were increased to three to ten years of imprisonment for fatal accidents, and one to seven years for severe injuries.
  - Period 3 (108/7 ~ 111/02): This period saw higher fines for repeat offenders, including a maximum fine of NT$120,000 for car drivers and NT$90,000 for motorcycle drivers. Additionally, new laws were introduced to enforce alcohol locks, mandatory treatment, and collective liability for drivers. A repeat offense leading to death could result in life imprisonment or a sentence of over five years.
  - Period 4 (111/3 ~ 113/9): The period extended the repeat offender penalty timeframe from five years to ten years, with higher fines for those exceeding certain BAC levels. New rules also included the public release of names and photos of repeat offenders.
- For each of these periods, the data is organized into 12 columns, including statistics on the number of enforcement cases, incidents, fatalities, injuries, and related rates per 100,000 people. These columns allow us to analyze trends and the impact of legal reforms on drunk driving behavior over time. Combining this data into tables allows for a comprehensive and clear examination of the trends across different periods.
- the data is presented by per 100,000 people to account for fluctuations in population size over time. This normalization allows more accurate comparison trends across different periods, as it eliminates the potential bias caused by population growth or decline. By standardizing the data in this way, we ensure that the trends reflect the true changes in drunk driving incidents, penalties, and outcomes, rather than being influenced by demographic shifts. This approach makes it easier to assess the effectiveness of legal reforms and their impact on society, regardless of population changes.

a. Data Types and Definitions
    i. Incident Data: Number of drunk driving cases annually.
        1. Source: NPA statistics.
    ii. Death and Injury Data: Annual deaths and injuries caused by drunk driving.
        1. Source: Ministry of transportation statistic
    iii. Legal Data: History and updates of penalties for drunk driving (e.g., amendments in 2022).
        1. Sources: Government laws summary, Timeline.
b. Planned Data Collection Process
    i. Download raw data from open data portals and government websites.
    ii. Review legislation updates and categorize changes in penalty severity.
    iii. Aggregate data by year for longitudinal analysis.
c. Potential Bias Considerations
    i. Underreporting: Some incidents may not be recorded or categorized accurately.
    ii. Policy Lag Effect: Policy changes may show delayed impact, creating misleading short-term trends.
d. Reasonable Data Scope
    i. Time Period: Collect data from at least 10 years (e.g., 2012–2022) to observe long-term trends.
    ii. Geographic Scope: Focus on nationwide data to ensure comprehensive insights.
    iii. Data Size: Gather incident counts, demographic breakdowns, and legal changes over time.
e. Data Sources Overview
    i. Incident Reports and Accident Data:
        1. NPA Drunk Driving Statistics
        2. Data on Accidents
    ii. Legal and Penalty Records:
        1. Drunk Driving Penalty Information
        2. Timeline of Penalty Revisions
    iii. Demographic and Mortality Data:
        1. Death Reason Statistics

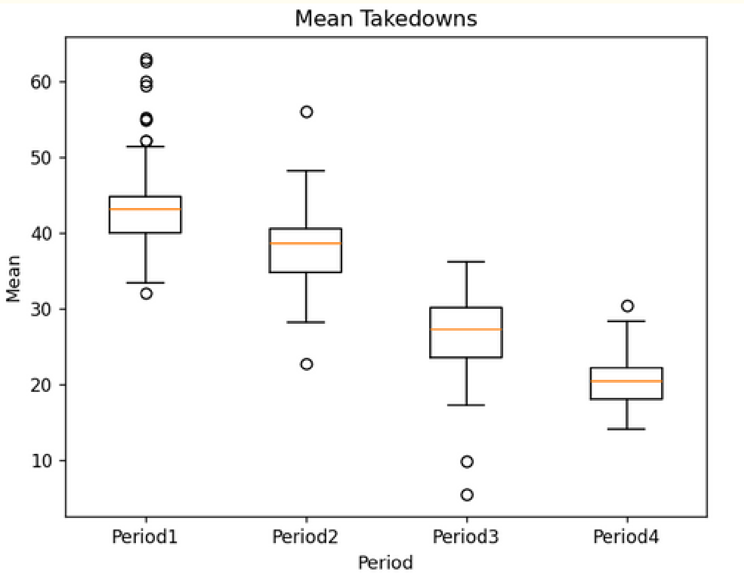
- Analysis Methods
  - Test
    - T-Test/ANOVA: To compare the means of incidents before and after policy changes.
    - Regression Analysis: To analyze the impact of stricter punishments on the number of incidents, controlling for other variables like public awareness campaigns.
    - Correlation Analysis: To evaluate the relationship between the number of incidents and injuries or fatalities.
  - Hypothesis:
    - H1: Stricter punishments lead to a significant reduction in drunk driving incidents.
      - Important event: 1999 (criminal penalty is added), 2013 (convert imprisonment into fine or detention is removed), 2019 (progressive punishment), 2022 (name and picture revealed if is a recidivist)
      - Throughout our analysis, we found that drunk driving incidents can be categorized into various types of offenses and circumstances. This distinction is important because different levels of offenses (e.g., alcohol concentration, repeat offenders, accidents causing injuries or fatalities) may be influenced by legal penalties in different ways. Therefore, it is necessary to conduct a more detailed investigation to determine whether each specific type of drunk driving incident is impacted differently by the changes in penalties over time
      - H_0: new regulations or penalties have not significantly reduced drunk driving related incidents
        - 2013≥1999. 2019≥2013, 2022≥2019
      - H_1: new regulations or penalties have significantly reduced drunk driving related incidents
        - 2013<1999, 2019<2013, 2022<2019
    - ~~H2: Teenagers are disproportionately represented in drunk driving statistics~~
      - ~~during the course of our analysis, however, due to limitations in available data, particularly the lack of sufficient demographic breakdowns for age group, we found it challenging to effectively assess this hypothesis. As a result, we were unable to perform analysis~~
    - ~~H3: There is a correlation between the severity of punishments and the reduction in drunk driving-related injuries and fatalities.~~
      - We change H2 and H3 since we found that there's lot's of classification in drunk driving behaviors that are worth discussing separately instead of only "incident", see the statistic analysis part.

- Potential bias
    - The awareness of severity of drunk driving due to incident on the news.
    - The awareness of severity of drunk driving due to education.
    - The popularity of chauffeur service and public transportation.

- Expected Results
    - Reduction in Drunk Driving Incidents Over Time
        - It is expected that the data will show a decrease in the number of drunk driving incidents over the years, especially following the implementation of stricter punishments or public awareness campaigns.
    - Impact of Stricter Punishments
        - Based on the hypothesis that harsher penalties deter drunk driving, the statistical tests (e.g., regression analysis) may indicate a significant negative correlation between the severity of punishments and the frequency of drunk
    - Correlation Between Drunk Driving and Fatalities/Injuries
        - A strong positive correlation is anticipated between the number of drunk driving incidents and related fatalities or injuries, reinforcing the importance of preventive measures.
    - Effectiveness of Punitive Measures Over Time
        - Over time, the data might suggest that punitive measures are effective in reducing both the overall number of incidents and the severity of their consequences (e.g., injuries and deaths).

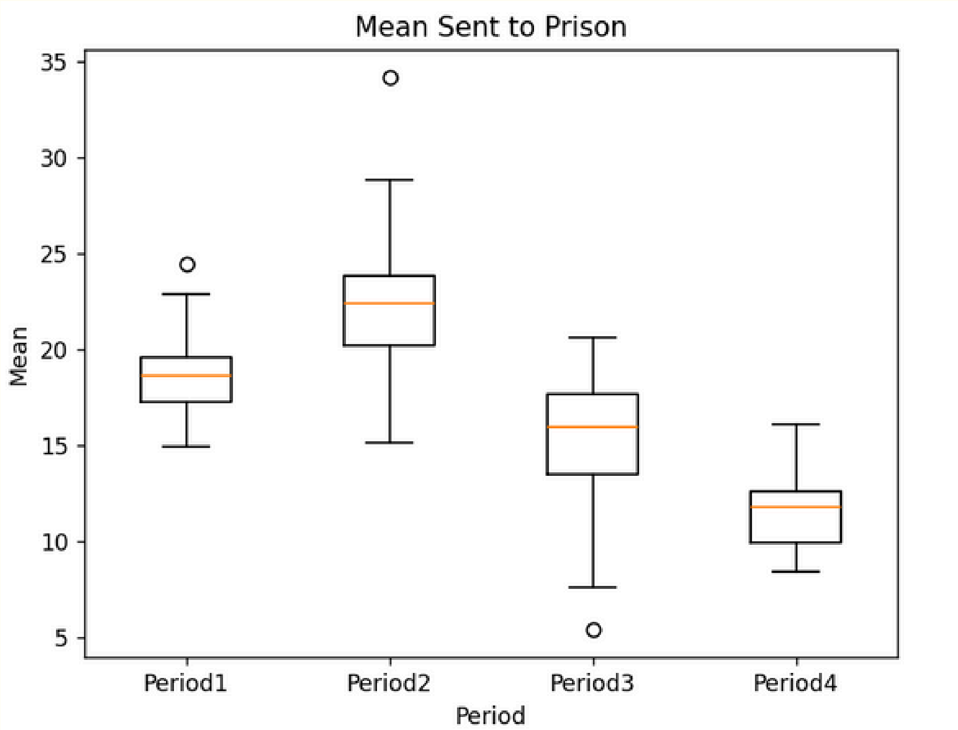
- Descriptive Data Analysis
    - explores patterns and distributions of law enforcement activities, accidents, deaths, and injuries over different periods
    - the following metrics are considered:
    - Takedowns per 100,000 people
    - Sent to prison per 100,000 people
    - Accidents per 100,000 people
    - Deaths per 100,000 people
    - Injuries per 100,000 people
    - Mean takedowns per 100,000 people:
        - boxplot is used to illustrate distribution of takedown counts across four periods since it summarizes data, showing the median, interquartile range and any outliers
        - this graph helps identify central tendency and variation of law enforcement activities over time

```
#--------------------Takedowns--------------------
plt.clf()
plt.boxplot([period1["取締件數每十萬人"], period2["取締件數每十萬人"], period3["取締件數每十萬人"], period4["取締件數每十萬人"]], tick_labels=["Period1", "Period2", "Period3", "Period4"])
plt.title("Mean Takedowns")
plt.xlabel("Period")
plt.ylabel("Mean")
plt.savefig("Mean Takedowns.png")
# plt.show()
```



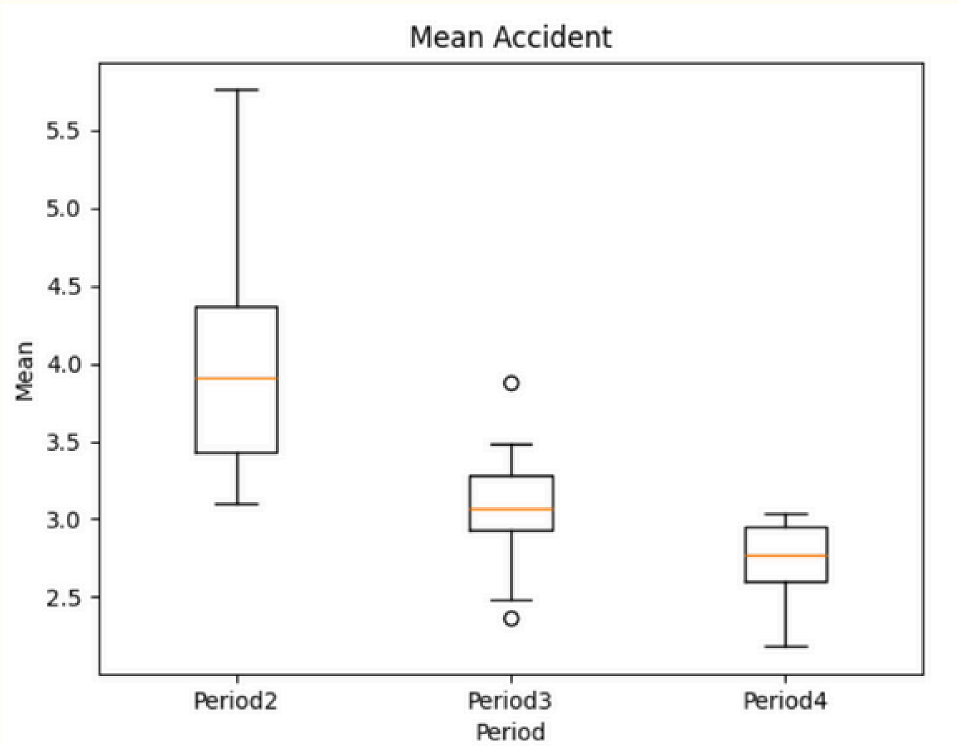
- Mean sent to prison per 100,000 people
    - boxplot describe the distribution of prison sentences per 100,00 people across the four periods
    - helps identify trends in legal enforcement and potential shifts in law enforcement practices

```python
#--------------------Sent to prison--------------------
plt.clf()
plt.boxplot([period1["移送法辦每十萬人"], period2["移送法辦每十萬人"], period3["移送法辦每十萬人"], period4["移送法辦每十萬人"]], tick_labels=["Period1", "Period2", "Period3",
plt.title("Mean Sent to Prison")
plt.xlabel("Period")
plt.ylabel("Mean")
plt.savefig("Mean Sent to Prison.png")
plt.show()
```



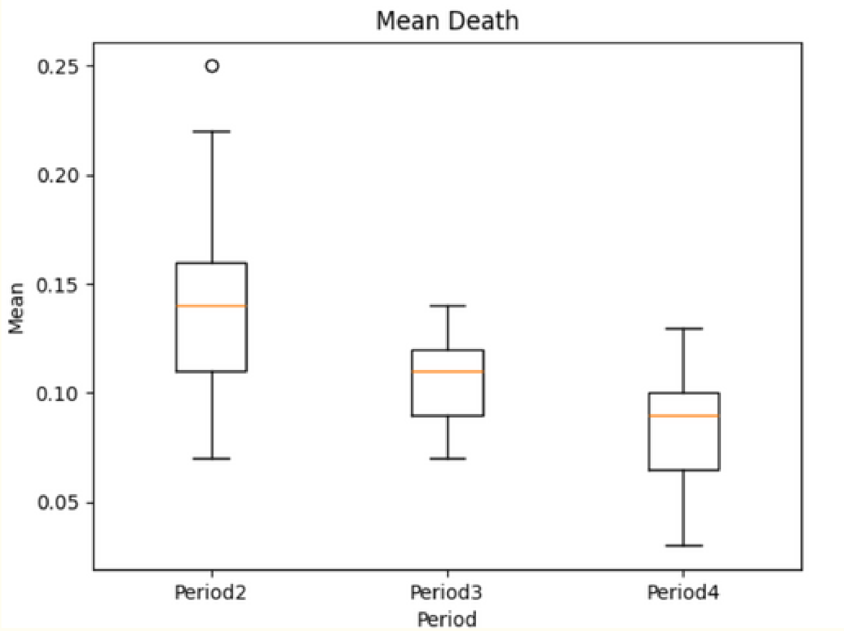
- Mean accidents per 100,00 people
  - boxplot shows distribution of accidents per 100,00 people for period 2,3,4
  - period 1 is omitted due to missing or incomplete values
  - highlight central tendency and spread of accident counts over the periods

```python
#--------------------Accident--------------------
plt.clf()
plt.boxplot([period2["事故件數每十萬人"].dropna(), period3["事故件數每十萬人"], period4["事故件數每十萬人"]], tick_labels=["Period2", "Period3", "Period4"])
plt.title("Mean Accident")
plt.xlabel("Period")
plt.ylabel("Mean")
plt.savefig("Mean Accident.png")
# plt.show()
```



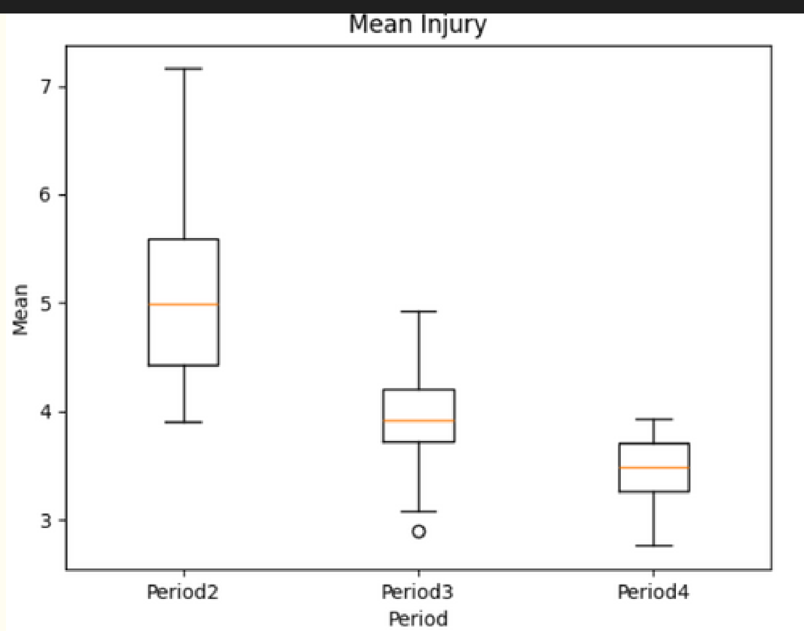
- Mean death per 100,000 people
  - same as mean accidents, missing values are excluded for clearer analysis
  - provides insights into potential impacts of accidents and enforcement on public health

```
#-------------------Death--------------------
plt.clf()
plt.boxplot([period2["死亡人數每十萬人"].dropna(), period3["死亡人數每十萬人"], period4["死亡人數每十萬人"]], tick_labels=["Period2", "Period3", "Period4
plt.title("Mean Death")
plt.xlabel("Period")
plt.ylabel("Mean")
plt.savefig("Mean Death.png")
# plt.show()
```



- Mean injuries per 100,000 people
  - same as mean death, missing values are excluded
  - provides insight on the variations in injuries over time, essential for accessing public health impact of accidents and law enforcement practices

```
#--------------------Injury--------------------
plt.clf()
plt.boxplot([period2["受傷人數每十萬人"].dropna(), period3["受傷人數每十萬人"], period4["受傷人數每十萬人"]], tick_labels=["Period2", "Period3", "Period
plt.title("Mean Injury")
plt.xlabel("Period")
plt.ylabel("Mean")
plt.savefig("Mean Injury.png")
# plt.show()
```

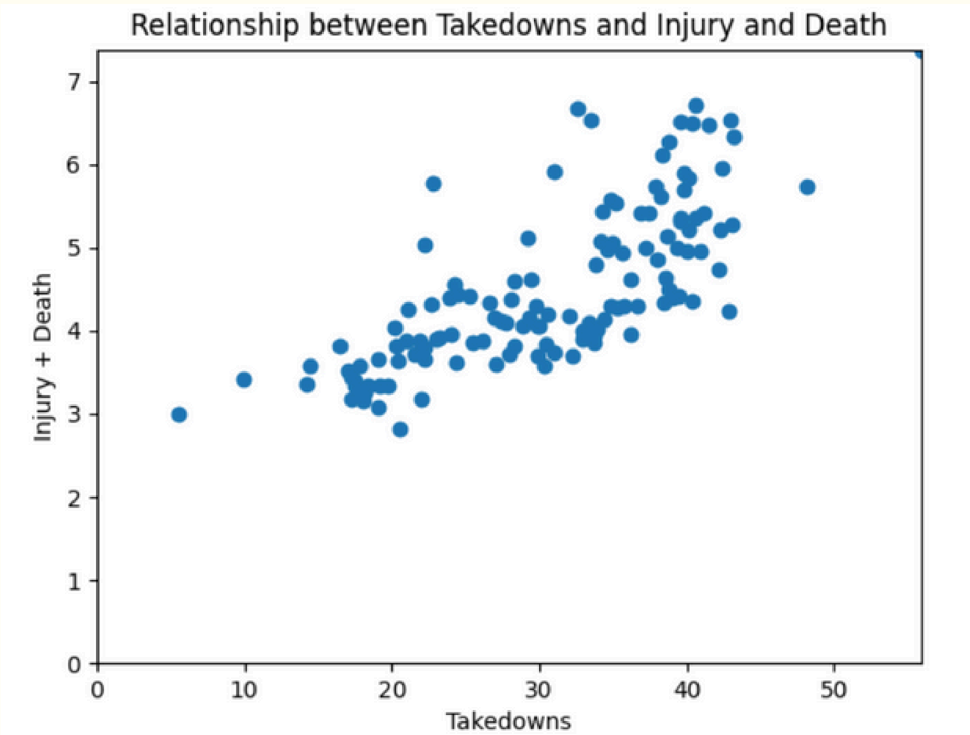


- Relationship between takedowns and injury + death rates
  - using a scatter plot to visualize this relationship and calculate the Pearson correlation coefficient to access strength and direction of the association (helps detect linear relationship between takedowns and combined injuries/deaths

```
#--------------------Relationship between takedowns and injury and death--------------------
x_axis = period_all["取締件數每十萬人"]
y_axis = period_all["受傷人數每十萬人"] + period_all["死亡人數每十萬人"]
data = pd.DataFrame({"Takedowns": x_axis, "Injury + Death": y_axis}).dropna()
# print(data)
x_axis = data["Takedowns"]
y_axis = data["Injury + Death"]
print(len(x_axis))
print(len(y_axis))

r, p_value = stats.pearsonr(x_axis, y_axis)
print("r: ", r)
print("p-value: ", p_value)
plt.clf()

plt.xlim(0, max(x_axis))
plt.ylim(0, max(y_axis))
plt.scatter(x_axis, y_axis)
plt.title("Relationship between Takedowns and Injury and Death")
plt.xlabel("Takedowns")
plt.ylabel("Injury + Death")
plt.savefig("Relationship between Takedowns and Injury and Death.png")
# plt.show()
```



- **Statistical Tests:**
  - **H1: The amendment of laws are effective in reducing takedowns (column 2 of data)**
  - **H2: The amendment of laws are effective in reducing imprisonment (column 4 of data)**
  - **H3: The amendment of laws are effective in reducing accidents (column 6 of data)**
  - **H4: The amendment of laws are effective in reducing injury (column 8 of data)**
  - **H5: The amendment of laws are effective in reducing death (column 10 of data)**
  - **H6: There is a negative relationship between number of takedowns and the sum of injuries and death.**
  - **ANOVA** test is used to compare the means across multiple groups (in this case, the four periods) to see if there is a statistically significant difference. The procedure involves calculating the sum of squares between treatments (SST) and error (SSE), then deriving the mean squares (MST and MSE) and comparing the F-statistic against the critical value to assess the null hypothesis.
    - F-statistic: Compares the variance between the groups to the variance within groups.
    - Critical F-value: Derived from the F-distribution to determine the rejection region for the null hypothesis.
    - alpha = 0.05
    - p-value: Provides the probability of obtaining the observed F-statistic (or one more extreme) if the null hypothesis is true.
    - formula used:
      - SST=i∑ni(Xi−Xall)^2
      - SSE=i∑(ni−1)⋅Vari
        - ni: sample size of each group
        - Xi: mean for each period
        - Xall: overall mean
    - F-statistic and F-critical values are compared.
    - If F-statistic > F-critical, reject the null hypothesis (indicating that there is a significant difference between periods).
    - If p-value < 0.05, the null hypothesis is rejected, suggesting a significant difference.
  - **Pairwise t-test** used to explore differences between individual periods. Each pair of periods was tested for significant differences using the following approach:
    - t-statistic: Compares the means of two groups.
    - p-value: Tests the hypothesis that the means of two periods are equal.
    - formula used:
      - t=(∣ X1−X2 ∣)/ sqrt(pooled variance ⋅ (1/n1+1/n2))
        - n1 n2 : sample sizes of two groups being compared
        - pooled variances = sqrt(S1^2 / n1 + S2^2 / n2)
    - alpha = 0.05 / 6 or 0.05 / 3
    - p-value is compared against the adjusted alpha level (alpha/6) due to multiple comparisons (Bonferroni correction).

- Confidence intervals are calculated to provide a range of values for the difference in means, providing further insights into the size of the difference between periods.

- Takedown ANOVA
    - compare means of multiple period to determine if at least one period mean is statistically different from the others
    - formula
        - SST: variation due to difference in means between groups
        - SSE: variation within each group
        - MST: average variability due to treatment
        - MSE: average variability within groups
        - F-statistic: ratio of MST and MSE
        - F- critical: critical value of F for a give alpha level and degrees of freedom
        - p-value: probability of observing a test statistic as extreme as the one calculated if the null hypothesis is true

```
#--------------------Takedowns anova--------------------
print("takedowns anova")
alpha = 0.05
treatments = 4
len_period1 = len(period1)
len_period2 = len(period2)
len_period3 = len(period3)
len_period4 = len(period4)
                                                                  (variable) mean_takedowns_all: float
SST = len_period1 * (mean_takedowns_period1 - mean_takedowns_all) ** 2 + len_period2 * (mean_takedowns_period2 - mean_takedowns_all) ** 2 + len_period3 * (mean_takedowns_period3 - mean_t
SSE = (len_period1 - 1) * var_takedowns_period1 + (len_period2 - 1) * var_takedowns_period2 + (len_period3 - 1) * var_takedowns_period3 + (len_period4 - 1) * var_takedowns_period4
MST = SST / (treatments - 1)
MSE = SSE / (len_period1 + len_period2 + len_period3 + len_period4 - treatments)
F = MST / MSE
F_critical = stats.f.ppf(1 - alpha, treatments - 1, len_period1 + len_period2 + len_period3 + len_period4 - treatments
p_value = 1 - stats.f.cdf(F, treatments - 1, len_period1 + len_period2 + len_period3 + len_period4 - treatments)
print("F: ", F)
print("F critical: ", F_critical)
print("p-value: ", p_value)
```

```
takedowns anova
F:  158.0403667105012
F critical:  2.6478014150542575
p-value:  1.1102230246251565e-16
```

- We have enough evidence to show that the mean numebr of takedowns of four periods are not the same since the p value is small enough.

- Takedown Pairwise T-tests
    - compare means of each pair of periods
    - whether the difference between the means of two periods is significantly different from zero
    - formula:
        - p-value: using t-distribution.
            - Tests null hypothesis that the means of the two periods being compared are equal.
        - Confidence Interval (Upper and Lower Bounds): For each pair of periods, the code also calculates the 95% confidence interval for the difference in means.

p-value (right-below)	period1	period2	period3
period2	4.736e-09	x	x
period3	0	0	x
period4	0	0	0.00038

```python
print("takedowns t-test")
alpha = 0.05
alpha = alpha / 6
dof_MSE = len_period1 + len_period2 + len_period3 + len_period4 - treatments
pool_estimator = MSE ** 0.5

dof_pair1 = len_period1 + len_period2 - 2
dof_pair2 = len_period1 + len_period3 - 2
dof_pair3 = len_period1 + len_period4 - 2
dof_pair4 = len_period2 + len_period3 - 2
dof_pair5 = len_period2 + len_period4 - 2
dof_pair6 = len_period3 + len_period4 - 2

p_value_of_pair1 = 2 * (1 - stats.t.cdf((mean_takedowns_period1 - mean_takedowns_period2) / (pool_estimator * (1 / len_period1 + 1 / len_period2) ** 0.5), dof_pair1))
p_value_of_pair2 = 2 * (1 - stats.t.cdf((mean_takedowns_period1 - mean_takedowns_period3) / (pool_estimator * (1 / len_period1 + 1 / len_period3) ** 0.5), dof_pair2))
p_value_of_pair3 = 2 * (1 - stats.t.cdf((mean_takedowns_period1 - mean_takedowns_period4) / (pool_estimator * (1 / len_period1 + 1 / len_period4) ** 0.5), dof_pair3))
p_value_of_pair4 = 2 * (1 - stats.t.cdf((mean_takedowns_period2 - mean_takedowns_period3) / (pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5), dof_pair4))
p_value_of_pair5 = 2 * (1 - stats.t.cdf((mean_takedowns_period2 - mean_takedowns_period4) / (pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5), dof_pair5))
p_value_of_pair6 = 2 * (1 - stats.t.cdf((mean_takedowns_period3 - mean_takedowns_period4) / (pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5), dof_pair6))

upper_bound_pair1 = (mean_takedowns_period1 - mean_takedowns_period2) + stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period1 + 1 / len_period2) ** 0.5
lower_bound_pair1 = (mean_takedowns_period1 - mean_takedowns_period2) - stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period1 + 1 / len_period2) ** 0.5
upper_bound_pair2 = (mean_takedowns_period1 - mean_takedowns_period3) + stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period1 + 1 / len_period3) ** 0.5
lower_bound_pair2 = (mean_takedowns_period1 - mean_takedowns_period3) - stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period1 + 1 / len_period3) ** 0.5
upper_bound_pair3 = (mean_takedowns_period1 - mean_takedowns_period4) + stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period1 + 1 / len_period4) ** 0.5
lower_bound_pair3 = (mean_takedowns_period1 - mean_takedowns_period4) - stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period1 + 1 / len_period4) ** 0.5
upper_bound_pair4 = (mean_takedowns_period2 - mean_takedowns_period3) + stats.t.ppf(1 - alpha/2, dof_pair4) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
lower_bound_pair4 = (mean_takedowns_period2 - mean_takedowns_period3) - stats.t.ppf(1 - alpha/2, dof_pair4) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
upper_bound_pair5 = (mean_takedowns_period2 - mean_takedowns_period4) + stats.t.ppf(1 - alpha/2, dof_pair5) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
lower_bound_pair5 = (mean_takedowns_period2 - mean_takedowns_period4) - stats.t.ppf(1 - alpha/2, dof_pair5) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
upper_bound_pair6 = (mean_takedowns_period3 - mean_takedowns_period4) + stats.t.ppf(1 - alpha/2, dof_pair6) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
lower_bound_pair6 = (mean_takedowns_period3 - mean_takedowns_period4) - stats.t.ppf(1 - alpha/2, dof_pair6) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
```

```python
print("1-2 p-value: ", p_value_of_pair1)
print("1-2 Upper bound pair1: ", upper_bound_pair1)
print("1-2 Lower bound pair1: ", lower_bound_pair1)

print("1-3 p-value: ", p_value_of_pair2)
print("1-3 Upper bound pair2: ", upper_bound_pair2)
print("1-3 Lower bound pair2: ", lower_bound_pair2)

print("1-4 p-value: ", p_value_of_pair3)
print("1-4 Upper bound pair3: ", upper_bound_pair3)
print("1-4 Lower bound pair3: ", lower_bound_pair3)

print("2-3 p-value: ", p_value_of_pair4)
print("2-3 Upper bound pair4: ", upper_bound_pair4)
print("2-3 Lower bound pair4: ", lower_bound_pair4)

print("2-4 p-value: ", p_value_of_pair5)
print("2-4 Upper bound pair5: ", upper_bound_pair5)
print("2-4 Lower bound pair5: ", lower_bound_pair5)

print("3-4 p-value: ", p_value_of_pair6)
print("3-4 Upper bound pair6: ", upper_bound_pair6)
print("3-4 Lower bound pair6: ", lower_bound_pair6)
```

```
takedowns t-test
1-2 p-value:  4.375682705060058e-09
1-2 Upper bound pair1:  8.21376194010675
1-2 Lower bound pair1:  3.2854130243313753
1-3 p-value:  0.0
1-3 Upper bound pair2:  20.87014926599172
1-3 Lower bound pair2:  14.448499382656943
1-4 p-value:  0.0
1-4 Upper bound pair3:  26.251392394328274
1-4 Lower bound pair3:  19.75661109303007
2-3 p-value:  0.0
2-3 Upper bound pair4:  15.106608644674898
2-3 Lower bound pair4:  8.712865039535638
2-4 p-value:  0.0
2-4 Upper bound pair5:  20.487954850905304
2-4 Lower bound pair5:  14.020873672014913
3-4 p-value:  0.0003844961283427928
3-4 Upper bound pair6:  9.222257338908198
3-4 Lower bound pair6:  1.467097499801484
```

- We have enough evidence to show that the mean number of takedowns had been decreased while entering new period (new laws are applied) due to the CI.

- Sent-to-prison ANOVA
  - compare means of number of people sent to prison across the four periods

```
#--------------------Sent to prison anova--------------------
print("sent to prison anova")
alpha = 0.05
treatments = 4
len_period1 = len(period1)
len_period2 = len(period2)
len_period3 = len(period3)
len_period4 = len(period4)

SST = len_period1 * (mean_sent_to_prison_period1 - mean_sent_to_prison_all) ** 2 + len_period2 * (mean_sent_to_prison_period2 - mean_sent_to_prison_all) ** 2 + len_period3 * (mean_sent_to_prison_period3 -
SSE = (len_period1 - 1) * var_sent_to_prison_period1 + (len_period2 - 1) * var_sent_to_prison_period2 + (len_period3 - 1) * var_sent_to_prison_period3 + (len_period4 - 1) * var_sent_to_prison_period4
MST = SST / (treatments - 1)
MSE = SSE / (len_period1 + len_period2 + len_period3 + len_period4 - treatments)
F = MST / MSE
F_critical = stats.f.ppf(1 - alpha, treatments - 1, len_period1 + len_period2 + len_period3 + len_period4 - treatments)
p_value = 1 - stats.f.cdf(F, treatments - 1, len_period1 + len_period2 + len_period3 + len_period4 - treatments)
print("F: ", F)
print("F critical: ", F_critical)
print("p-value: ", p_value)
```

```
sent to prison anova
F:  153.5555070632156
F critical:  2.6478014150542575
p-value:  1.1102230246251565e-16
```

  - We have enough evidence to show that the mean Sent-to-prison of four periods are not the same since the p value is small enough.
- Sent to prison Pairwise T-tests

p-value (right-below)	period1	period2	period3
period2	2.000	x	x
period3	5.253e-07	0	x
period4	0	0	4.849e-08

```python
print("sent to prison t-test")
alpha = 0.05
alpha = alpha / 6
dof_MSE = len_period1 + len_period2 + len_period3 + len_period4 - treatments
pool_estimator = MSE ** 0.5

dof_pair1 = len_period1 + len_period2 - 2
dof_pair2 = len_period1 + len_period3 - 2
dof_pair3 = len_period1 + len_period4 - 2
dof_pair4 = len_period2 + len_period3 - 2
dof_pair5 = len_period2 + len_period4 - 2
dof_pair6 = len_period3 + len_period4 - 2

p_value_of_pair1 = 2 * (1 - stats.t.cdf((mean_sent_to_prison_period1 - mean_sent_to_prison_period2) / (pool_estimator * (1 / len_period1 + 1 / len_period2) ** 0.5), dof_pair1))
p_value_of_pair2 = 2 * (1 - stats.t.cdf((mean_sent_to_prison_period1 - mean_sent_to_prison_period3) / (pool_estimator * (1 / len_period1 + 1 / len_period3) ** 0.5), dof_pair2))
p_value_of_pair3 = 2 * (1 - stats.t.cdf((mean_sent_to_prison_period1 - mean_sent_to_prison_period4) / (pool_estimator * (1 / len_period1 + 1 / len_period4) ** 0.5), dof_pair3))
p_value_of_pair4 = 2 * (1 - stats.t.cdf((mean_sent_to_prison_period2 - mean_sent_to_prison_period3) / (pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5), dof_pair4))
p_value_of_pair5 = 2 * (1 - stats.t.cdf((mean_sent_to_prison_period2 - mean_sent_to_prison_period4) / (pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5), dof_pair5))
p_value_of_pair6 = 2 * (1 - stats.t.cdf((mean_sent_to_prison_period3 - mean_sent_to_prison_period4) / (pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5), dof_pair6))


upper_bound_pair1 = (mean_sent_to_prison_period1 - mean_sent_to_prison_period2) + stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period1 + 1 / len_period2) ** 0.5
lower_bound_pair1 = (mean_sent_to_prison_period1 - mean_sent_to_prison_period2) - stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period1 + 1 / len_period2) ** 0.5
upper_bound_pair2 = (mean_sent_to_prison_period1 - mean_sent_to_prison_period3) + stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period1 + 1 / len_period3) ** 0.5
lower_bound_pair2 = (mean_sent_to_prison_period1 - mean_sent_to_prison_period3) - stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period1 + 1 / len_period3) ** 0.5
upper_bound_pair3 = (mean_sent_to_prison_period1 - mean_sent_to_prison_period4) + stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period1 + 1 / len_period4) ** 0.5
lower_bound_pair3 = (mean_sent_to_prison_period1 - mean_sent_to_prison_period4) - stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period1 + 1 / len_period4) ** 0.5
upper_bound_pair4 = (mean_sent_to_prison_period2 - mean_sent_to_prison_period3) + stats.t.ppf(1 - alpha/2, dof_pair4) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
lower_bound_pair4 = (mean_sent_to_prison_period2 - mean_sent_to_prison_period3) - stats.t.ppf(1 - alpha/2, dof_pair4) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
upper_bound_pair5 = (mean_sent_to_prison_period2 - mean_sent_to_prison_period4) + stats.t.ppf(1 - alpha/2, dof_pair5) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
lower_bound_pair5 = (mean_sent_to_prison_period2 - mean_sent_to_prison_period4) - stats.t.ppf(1 - alpha/2, dof_pair5) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
upper_bound_pair6 = (mean_sent_to_prison_period3 - mean_sent_to_prison_period4) + stats.t.ppf(1 - alpha/2, dof_pair6) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
lower_bound_pair6 = (mean_sent_to_prison_period3 - mean_sent_to_prison_period4) - stats.t.ppf(1 - alpha/2, dof_pair6) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
```

```python
print("1-2 p-value: ", p_value_of_pair1)
print("1-2 Upper bound pair1: ", upper_bound_pair1)
print("1-2 Lower bound pair1: ", lower_bound_pair1)

print("1-3 p-value: ", p_value_of_pair2)
print("1-3 Upper bound pair2: ", upper_bound_pair2)
print("1-3 Lower bound pair2: ", lower_bound_pair2)

print("1-4 p-value: ", p_value_of_pair3)
print("1-4 Upper bound pair3: ", upper_bound_pair3)
print("1-4 Lower bound pair3: ", lower_bound_pair3)

print("2-3 p-value: ", p_value_of_pair4)
print("2-3 Upper bound pair4: ", upper_bound_pair4)
print("2-3 Lower bound pair4: ", lower_bound_pair4)

print("2-4 p-value: ", p_value_of_pair5)
print("2-4 Upper bound pair5: ", upper_bound_pair5)
print("2-4 Lower bound pair5: ", lower_bound_pair5)

print("3-4 p-value: ", p_value_of_pair6)
print("3-4 Upper bound pair6: ", upper_bound_pair6)
print("3-4 Lower bound pair6: ", lower_bound_pair6)
```

```
sent to prison t-test
1-2 p-value:  1.9999999999999505
1-2 Upper bound pair1:  -2.4426739436823937
1-2 Lower bound pair1:  -4.751316098991865
1-3 p-value:  5.252745416406412e-07
1-3 Upper bound pair2:  4.496084687437333
1-3 Lower bound pair2:  1.4879186909410496
1-4 p-value:  0.0
1-4 Upper bound pair3:  8.660562365879777
1-4 Lower bound pair3:  5.618138593143769
2-3 p-value:  0.0
2-3 Upper bound pair4:  8.086543482206398
2-3 Lower bound pair4:  5.091449938846242
2-4 p-value:  0.0
2-4 Upper bound pair5:  12.251069446591071
2-4 Lower bound pair5:  9.221621555106733
3-4 p-value:  4.848944579194381e-08
3-4 Upper bound pair6:  5.963767354805243
3-4 Lower bound pair6:  2.3309302258399205
```

- Excluded the period 1-2 pair, we have enough evidence to show that the mean sent-to-prison had been decreased while entering new period (new laws are applied) due to the CI.

- We infer that the reason why the number of sent-to-prison had increased from period 1 to period 2 is that people can no longer covert imprisonment into fine or detention due to the new law, so there existed a short burst in that period.

- Accident ANOVA
  - test whether there are any statistically significant differences between means of three of more independent groups (comparing accident counts in period 2, 3, 4)
  - Data below were only recorded after period 2, so period 1 is not included.

```
#--------------------Accident anova--------------------
print("accident anova")
alpha = 0.05
treatments = 3
len_period2 = len(period2)
len_period3 = len(period3)
len_period4 = len(period4)

SST = len_period2 * (mean_accident_period2 - mean_accident_all) ** 2 (variable) var_accident_period3: Scalar mean_accident_all) ** 2 + len_period4 * (mean_accident_period4 - mean_accident_all) ** 2
SSE = (len_period2 - 1) * var_accident_period2 + (len_period3 - 1) * var_accident_period3 + (len_period4 - 1) * var_accident_period4
MST = SST / (treatments - 1)
MSE = SSE / (len_period2 + len_period3 + len_period4 - treatments)
F = MST / MSE
F_critical = stats.f.ppf(1 - alpha, treatments - 1, len_period2 + len_period3 + len_period4 - treatments)
p_value = 1 - stats.f.cdf(F, treatments - 1, len_period2 + len_period3 + len_period4 - treatments)
print("F: ", F)
print("F critical: ", F_critical)
print("p-value: ", p_value)
```

```
accident anova
F:  83.50770471757689
F critical:  3.0627003994564683
p-value:  1.1102230246251565e-16
```

  - We have enough evidence to show that the mean number of accident of three periods are not the same.

- Accident Pairwise T-test

p-value (right-below)	**period2**	**period3**
period3	**1.255e-13**	**x**
period4	**0**	**0.00635**

```
print("accident t-test")
alpha = 0.05
alpha = alpha / 3
dof_MSE = len_period2 + len_period3 + len_period4 - treatments
pool_estimator = MSE ** 0.5

dof_pair1 = len_period2 + len_period3 - 2
dof_pair2 = len_period2 + len_period4 - 2
dof_pair3 = len_period3 + len_period4 - 2

p_value_of_pair1 = 2 * (1 - stats.t.cdf((mean_accident_period2 - mean_accident_period3) / (pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5), dof_pair1))
p_value_of_pair2 = 2 * (1 - stats.t.cdf((mean_accident_period2 - mean_accident_period4) / (pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5), dof_pair2))
p_value_of_pair3 = 2 * (1 - stats.t.cdf((mean_accident_period3 - mean_accident_period4) / (pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5), dof_pair3))


upper_bound_pair1 = (mean_accident_period2 - mean_accident_period3) + stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
lower_bound_pair1 = (mean_accident_period2 - mean_accident_period3) - stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
upper_bound_pair2 = (mean_accident_period2 - mean_accident_period4) + stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
lower_bound_pair2 = (mean_accident_period2 - mean_accident_period4) - stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
upper_bound_pair3 = (mean_accident_period3 - mean_accident_period4) + stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
lower_bound_pair3 = (mean_accident_period3 - mean_accident_period4) - stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
print("2-3 p-value: ", p_value_of_pair1)
print("2-3 Upper bound pair1: ", upper_bound_pair1)
print("2-3 Lower bound pair1: ", lower_bound_pair1)

print("2-4 p-value: ", p_value_of_pair2)
print("2-4 Upper bound pair2: ", upper_bound_pair2)
print("2-4 Lower bound pair2: ", lower_bound_pair2)

print("3-4 p-value: ", p_value_of_pair3)
print("3-4 Upper bound pair3: ", upper_bound_pair3)
print("3-4 Lower bound pair3: ", lower_bound_pair3)
```

```
accident t-test
2-3 p-value:  1.254552017826427e-13
2-3 Upper bound pair1:  1.1356021756862766
2-3 Lower bound pair1:  0.6305910061319029
2-4 p-value:  0.0
2-4 Upper bound pair2:  1.4893682335211011
2-4 Lower bound pair2:  0.9785789805551415
3-4 p-value:  0.006352823526291074
3-4 Upper bound pair3:  0.656469064627567
3-4 Lower bound pair3:  0.045284967630496
```

   - We have enough evidence to show that the mean number of accident had been decreased while entering new period (new laws are applied) due to the CI.

- Death ANOVA
   - test whether there are any statistically significant differences between means of three of more independent groups (comparing accident counts in period 2, 3, 4)

```python
#--------------------Death anova--------------------
print("death anova")
alpha = 0.05
treatments = 3
len_period2 = len(period2)
len_period3 = len(period3)
len_period4 = len(period4)

SST = len_period2 * (mean_death_period2 - mean_death_all) ** 2 + len_period3 * (mean_death_period3 - mean_death_all) ** 2 + len_period4 * (mean_death_period4 - mean_death_all) ** 2
SSE = (len_period2 - 1) * var_death_period2 + (len_period3 - 1) * var_death_period3 + (len_period4 - 1) * var_death_period4
MST = SST / (treatments - 1)
MSE = SSE / (len_period2 + len_period3 + len_period4 - treatments)
F = MST / MSE
F_critical = stats.f.ppf(1 - alpha, treatments - 1, len_period2 + len_period3 + len_period4 - treatments)
p_value = 1 - stats.f.cdf(F, treatments - 1, len_period2 + len_period3 + len_period4 - treatments)
print("F: ", F)
print("F critical: ", F_critical)
print("p-value: ", p_value)
```

```
death anova
F:  41.28827263165483
F critical:  3.0627003994564683
p-value:  9.769962616701378e-15
```

   - We have enough evidence to show that the mean number of death of three periods are not the same since the p value is small enough.

- Death Pairwise T-test

p-value (right-below)	**period2**	**period3**
period3	**7.387e-07**	**x**
period4	**8.282e-14**	**0.00563**

```python
#--------------------Death t-test--------------------
print("death t-test")
alpha = 0.05
alpha = alpha / 3
dof_MSE = len_period2 + len_period3 + len_period4 - treatments
pool_estimator = MSE ** 0.5

dof_pair1 = len_period2 + len_period3 - 2
dof_pair2 = len_period2 + len_period4 - 2
dof_pair3 = len_period3 + len_period4 - 2

p_value_of_pair1 = 2 * (1 - stats.t.cdf(abs(mean_death_period2 - mean_death_period3) / (pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5), dof_pair1))
p_value_of_pair2 = 2 * (1 - stats.t.cdf(abs(mean_death_period2 - mean_death_period4) / (pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5), dof_pair2))
p_value_of_pair3 = 2 * (1 - stats.t.cdf(abs(mean_death_period3 - mean_death_period4) / (pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5), dof_pair3))


upper_bound_pair1 = (mean_death_period2 - mean_death_period3) + stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
lower_bound_pair1 = (mean_death_period2 - mean_death_period3) - stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
upper_bound_pair2 = (mean_death_period2 - mean_death_period4) + stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
lower_bound_pair2 = (mean_death_period2 - mean_death_period4) - stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
upper_bound_pair3 = (mean_death_period3 - mean_death_period4) + stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
lower_bound_pair3 = (mean_death_period3 - mean_death_period4) - stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5


print("2-3 p-value: ", p_value_of_pair1)
print("2-3 Upper bound pair1: ", upper_bound_pair1)
print("2-3 Lower bound pair1: ", lower_bound_pair1)

print("2-4 p-value: ", p_value_of_pair2)
print("2-4 Upper bound pair2: ", upper_bound_pair2)
print("2-4 Lower bound pair2: ", lower_bound_pair2)

print("3-4 p-value: ", p_value_of_pair3)
print("3-4 Upper bound pair3: ", upper_bound_pair3)
print("3-4 Lower bound pair3: ", lower_bound_pair3)
```


```python
print("2-3 p-value: ", p_value_of_pair1)
print("2-3 Upper bound pair1: ", upper_bound_pair1)
print("2-3 Lower bound pair1: ", lower_bound_pair1)

print("2-4 p-value: ", p_value_of_pair2)
print("2-4 Upper bound pair2: ", upper_bound_pair2)
print("2-4 Lower bound pair2: ", lower_bound_pair2)

print("3-4 p-value: ", p_value_of_pair3)
print("3-4 Upper bound pair3: ", upper_bound_pair3)
print("3-4 Lower bound pair3: ", lower_bound_pair3)
```


```
death t-test
2-3 p-value:  7.387416200987929e-07
2-3 Upper bound pair1:  0.053150720498090456
2-3 Lower bound pair1:  0.019557612835242292
2-4 p-value:  8.282263763703668e-14
2-4 Upper bound pair2:  0.07704249498025559
2-4 Lower bound pair2:  0.04306503190146446
3-4 p-value:  0.005631776956772194
3-4 Upper bound pair3:  0.044027437433995294
3-4 Lower bound pair3:  0.003371756114391826
```

  - We have enough evidence to show that the mean number of death had been decreased while entering new period (new laws are applied) due to the CI.

- Injury ANOVA
  - test whether there are any statistically significant differences between means of three of more independent groups (comparing accident counts in period 2, 3, 4)

```python
print("injury anova")
alpha = 0.05
treatments = 3
len_period2 = len(period2)
len_period3 = len(period3)
len_period4 = len(period4)

SST = len_period2 * (mean_injury_period2 - mean_injury_all) ** 2 + len_period3 * (mean_injury_period3 - mean_injury_all) ** 2 + len_period4 * (mean_injury_period4 - mean_injury_all) ** 2
SSE = (len_period2 - 1) * var_injury_period2 + (len_period3 - 1) * var_injury_period3 + (len_period4 - 1) * var_injury_period4
MST = SST / (treatments - 1)
MSE = SSE / (len_period2 + len_period3 + len_period4 - treatments)
F = MST / MSE
F_critical = stats.f.ppf(1 - alpha, treatments - 1, len_period2 + len_period3 + len_period4 - treatments)
p_value = 1 - stats.f.cdf(F, treatments - 1, len_period2 + len_period3 + len_period4 - treatments)
print("F: ", F)
print("F critical: ", F_critical)
print("p-value: ", p_value)
```

```
injury anova
F:  88.05586119625725
F critical:  3.0627003994564683
p-value:  1.1102230246251565e-16
```

  - We have enough evidence to show that the mean number of injuty of three periods are not the same since the p value is small enough.

- Injury Pairwise T-test

p-value (right-below)	period2	period3
period3	**2.065e-14**	**x**
period4	**0**	**0.00801**

```python
#--------------------Injury t-test--------------------
print("injury t-test")
alpha = 0.05
alpha = alpha / 3
dof_MSE = len_period2 + len_period3 + len_period4 - treatments
pool_estimator = MSE ** 0.5

dof_pair1 = len_period2 + len_period3 - 2
dof_pair2 = len_period2 + len_period4 - 2
dof_pair3 = len_period3 + len_period4 - 2

p_value_of_pair1 = 2 * (1 - stats.t.cdf(abs(mean_injury_period2 - mean_injury_period3) / (pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5), dof_pair1))
p_value_of_pair2 = 2 * (1 - stats.t.cdf(abs(mean_injury_period2 - mean_injury_period4) / (pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5), dof_pair2))
p_value_of_pair3 = 2 * (1 - stats.t.cdf(abs(mean_injury_period3 - mean_injury_period4) / (pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5), dof_pair3))

upper_bound_pair1 = (mean_injury_period2 - mean_injury_period3) + stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
lower_bound_pair1 = (mean_injury_period2 - mean_injury_period3) - stats.t.ppf(1 - alpha/2, dof_pair1) * pool_estimator * (1 / len_period2 + 1 / len_period3) ** 0.5
upper_bound_pair2 = (mean_injury_period2 - mean_injury_period4) + stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
lower_bound_pair2 = (mean_injury_period2 - mean_injury_period4) - stats.t.ppf(1 - alpha/2, dof_pair2) * pool_estimator * (1 / len_period2 + 1 / len_period4) ** 0.5
upper_bound_pair3 = (mean_injury_period3 - mean_injury_period4) + stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
lower_bound_pair3 = (mean_injury_period3 - mean_injury_period4) - stats.t.ppf(1 - alpha/2, dof_pair3) * pool_estimator * (1 / len_period3 + 1 / len_period4) ** 0.5
```

```python
print("2-3 p-value: ", p_value_of_pair1)
print("2-3 Upper bound pair1: ", upper_bound_pair1)
print("2-3 Lower bound pair1: ", lower_bound_pair1)

print("2-4 p-value: ", p_value_of_pair2)
print("2-4 Upper bound pair2: ", upper_bound_pair2)
print("2-4 Lower bound pair2: ", lower_bound_pair2)

print("3-4 p-value: ", p_value_of_pair3)
print("3-4 Upper bound pair3: ", upper_bound_pair3)
print("3-4 Lower bound pair3: ", lower_bound_pair3)
```

```
injury t-test
2-3 p-value:  2.0650148258027912e-14
2-3 Upper bound pair1:  1.4875564505287147
2-3 Lower bound pair1:  0.8466291555318939
2-4 p-value:  0.0
2-4 Upper bound pair2:  1.9231282760909734
2-4 Lower bound pair2:  1.2748678138406015
3-4 p-value:  0.008013730576046596
3-4 Upper bound pair3:  0.8197427727163056
3-4 Lower bound pair3:  0.04406771115466074
```

  - We have enough evidence to show that the mean number of injury had been decreased while entering new period (new laws are applied) due to the CI.

- Relationship between number of takedowns and the sum of injuries and death.
    - We use Pearson correlation to check the relationship between takedowns and Injury/Death.

```
#--------------------Relationship between takedowns and injury and death--------------------
x_axis = period_all["取締件數每十萬人"]
y_axis = period_all["受傷人數每十萬人"] + period_all["死亡人數每十萬人"]
data = pd.DataFrame({"Takedowns": x_axis, "Injury + Death": y_axis}).dropna()
# print(data)
x_axis = data["Takedowns"]
y_axis = data["Injury + Death"]
print(len(x_axis))
print(len(y_axis))

r, p_value = stats.pearsonr(x_axis, y_axis)
print("r: ", r)
print("p-value: ", p_value)
plt.clf()

plt.xlim(0, max(x_axis))
plt.ylim(0, max(y_axis))
plt.scatter(x_axis, y_axis)
plt.title("Relationship between Takedowns and Injury and Death")
plt.xlabel("Takedowns")
plt.ylabel("Injury + Death")
plt.savefig("Relationship between Takedowns and Injury and Death.png")
# plt.show()
plt.close()
```

```
r:  0.7522143004359515
p-value:  9.160617098174 68e-25
```

    - The result shows that there exists a positive correlation between takedowns and injury/death with correlation coefficient = 0.75, thus H6 is completely rejected.

- Conclusions
    - The analyses show that the number of takedowns, accidents, injury and death did reduce as relative laws amended, noticing that other factors like awareness of drunk driving may also affect the result, too.

    - For imprisonment, there exist a slight bounce when entering period 2, but in long term, the number did reduce in period 3 and 4. This may show that people need time to raise the awareness that drunk driving is a behavior that should be completely prevented.

    - We can found that the amendment of new fines (NT$15,000 to NT$90,000) and a clearer alcohol limit exerts a huge influence on most of the indicator.

    - When the number of takedowns increases, it doesn’t mean that the injury/death are reduced, it simply means that the occurrence of drunk driving increases.

- Response to peer review question
    - Q: 是否新法規對酒駕的發生是有時效性的
    - A: 根據實驗結果，整體數據呈下滑趨勢，即便是在新法實施後期，各項指標仍持續下降。

    - Q: 以車禍平均值為統計量的話，可能要考慮歷年人民交通工具擁有數量的變化也可能是一個偏差 (如: 近年經濟狀況比較好，人民普遍擁有自己的汽機車，發生車禍的機會也就提高)
    - A: 在調閱相關資料後，整體載具的變化量相較於人口數的變化量小很多，因此我們選擇處理掉人口數的變因為優先，以每十萬人的數據來檢驗假設

- Creativity
    - The hot-map of drunk driving occurrence (see the video).
- Demo link
    - https://youtu.be/ASqbb4I5k1I