Over the past week, I attended two Boca Juniors Feminine football matches in Buenos Aires to study a phenomenon that initially seemed quite random: the time intervals between instances when the ball left the field of play. Throughout each 90-minute match, the ball exited the boundaries multiple times for reasons that ranged from throw-ins, corner kicks, and goal kicks to unpredictable deflections and player decisions. I chose this phenomenon because it appeared to be driven by a wide range of chance factors: player tactics evolving during the game, random bounces of the ball, changes in field conditions, and varying intensities of play as players tire over the course of the match. In other words, it wasn't possible to predict exactly when the next out-of-play event would occur.

To observe this, I recorded each time the ball went out of play and then calculated the interval in minutes between consecutive events. Over the course of two matches (totaling 180 minutes of play), I compiled a dataset of 20 observed time intervals. Initially, I expected that the waiting times might align with a simple memoryless pattern - if that were true, then an exponential distribution would be a natural fit. However, as I began analyzing the data, it became evident that something more complex was going on.

To get a better sense of the distributional properties, I started by plotting a histogram of the observed intervals and then fitting both an exponential and a gamma PDF to the data.

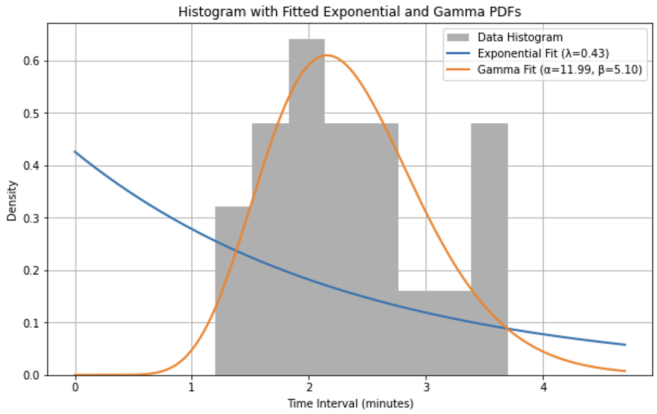



**Figure 1:** This figure shows the observed time intervals as a histogram overlaid with two fitted curves: the exponential fit (in blue) and the gamma fit (in orange). The histogram provides a direct visual representation of how frequently intervals of different lengths occurred. We can see that the gamma curve follows the shape of the histogram more closely, capturing the peak around two to three minutes better than the exponential curve. This visual evidence suggests that a gamma distribution, rather than an exponential one, is the more appropriate model for these data.

To further confirm these findings, I compared the CDFs of the empirical data with the theoretical CDFs of both candidate distributions.

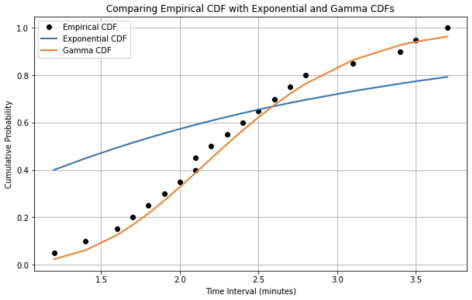



**Figure 2:** In this plot, the black dots represent the empirical CDF of the observed data, while the blue and orange lines represent the CDFs of the exponential and gamma distributions, respectively. Here again, the gamma CDF hugs the empirical points more tightly than the exponential one. Where the exponential model underestimates the probability for certain intervals, the gamma model tracks the observed data's growth more accurately, reinforcing the conclusion that the gamma distribution is the better fit.

I also constructed a Q-Q plot to see how well the quantiles of the observed data aligned with those of a fitted gamma distribution.

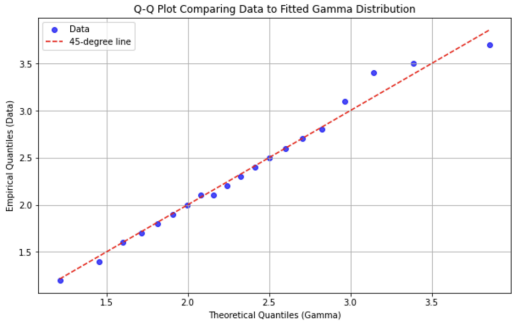
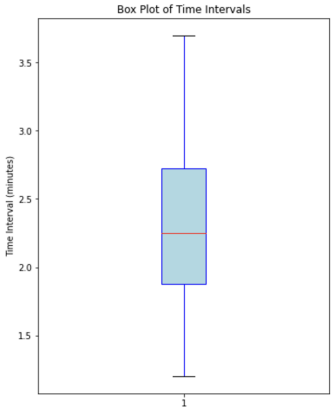



**Figure 3:** The Q-Q plot maps the empirical quantiles of the observed intervals against the theoretical gamma quantiles. Points that lie close to the red 45-degree reference line indicate a good match. We observe that most points fall neatly along this line, especially in the middle quantile range, indicating that the gamma distribution provides a very strong fit to the data. Finally, I created a simple box plot to summarize the distribution of the observed intervals.




```
Summary Statistics:
Mean: 2.35 minutes
Median: 2.25 minutes
Std Dev: 0.68 minutes
Min: 1.20 minutes
Q1: 1.88 minutes
Q3: 2.73 minutes
Max: 3.70 minutes
```

**Figure 4:** The box plot shows the median line slightly above two minutes, with the interquartile range spanning roughly from about 1.9 to 2.7 minutes. The whiskers extend towards the shortest and longest intervals observed without indicating extreme outliers. This summary suggests a fairly tight cluster of times around two to three minutes, consistent with the shape suggested by the gamma distribution fit.

**Summary**:
The time intervals at which the ball leaves the field during the two observed Boca Juniors Feminine matches are not governed by a simple, unchanging pattern. Instead, evolving factors - such as shifting tactics, increasing fatigue, and more aggressive plays down the wings - shape the likelihood of the ball going out at different times. This complexity means that a straightforward, constant-rate model (like the exponential) cannot fully capture the data, whereas the more flexible gamma model provides a closer approximation. Recognizing this interplay of conditions helps us appreciate the dynamic nature of in-play events in the beautiful game of football.

**Figure 5:** This image is meaningful to me as it connects to my love for football. As part of the Sports SI (made by M26) we recently (7th December) played the final football tournament, and my team won! I am proud to share this moment with the medal as a reflection of my passion for the sport.

AI Statement: I used a bit of AI to help me with the Q-Q plot mainly.

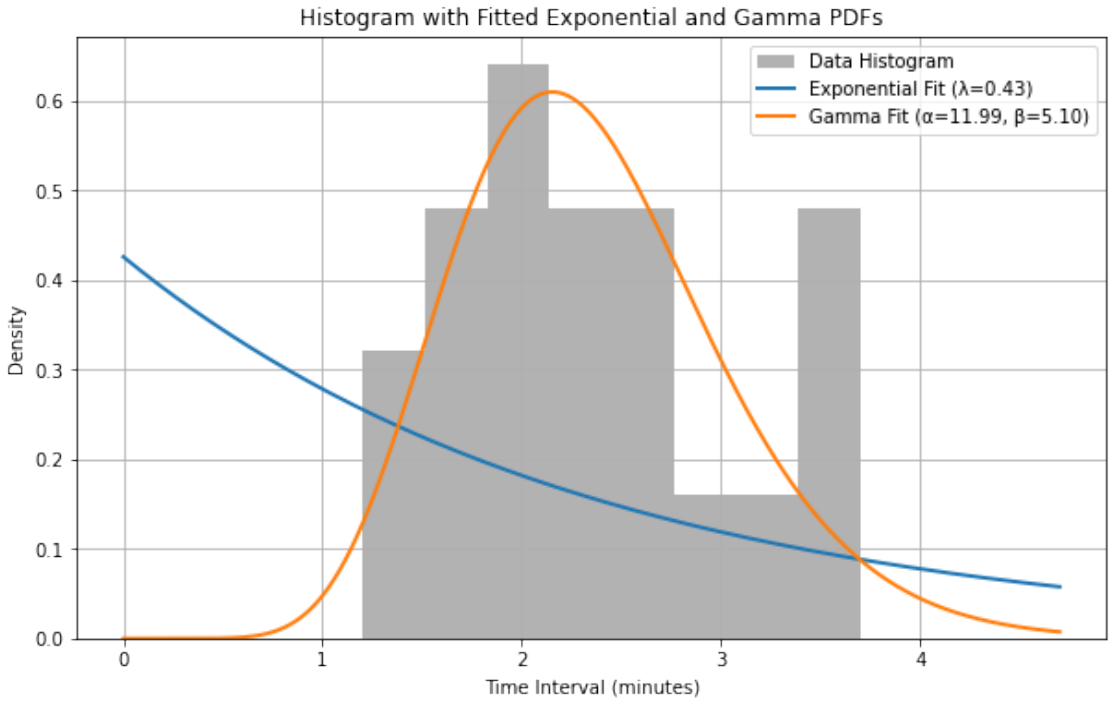
# Appendix - CS114 LBA Code

December 8, 2024

### 0.0.1 Histogram with Fitted PDFs

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import expon, gamma, kstest

# Collected time intervals (in minutes)
#Match 1 Observations: 1.2, 2.4, 1.8, 3.5, 2.1, 1.6, 3.7, 2.8, 1.9, 2.5
#Match 2 Observations: 2.0, 2.3, 1.7, 3.1, 2.2, 1.4, 3.4, 2.6, 2.1, 2.7
# I will combine them
data = np.array([1.2, 2.4, 1.8, 3.5, 2.1, 1.6, 3.7, 2.8, 1.9, 2.5, 2.0, 2.3, 1.
 ↪7, 3.1, 2.2, 1.4, 3.4, 2.6, 2.1, 2.7])

# Step 1: Calculate statistics
mean = np.mean(data)
variance = np.var(data)
std_dev = np.std(data)

# Step 2: Parameters for distributions
# Exponential distribution parameter
lambda_exp = 1 / mean

# Gamma distribution parameters
alpha_gamma = mean**2 / variance
beta_gamma = mean / variance


# Assuming data, lambda_exp, alpha_gamma, and beta_gamma are already defined
# data = np.array([...]) # Already defined previously

plt.figure(figsize=(10, 6))
# Plot histogram
count, bins, _ = plt.hist(data, bins=8, density=True, alpha=0.6, color='gray',␣
 ↪label='Data Histogram')

# Generate x values for PDF
x = np.linspace(0, max(data) + 1, 1000)
```

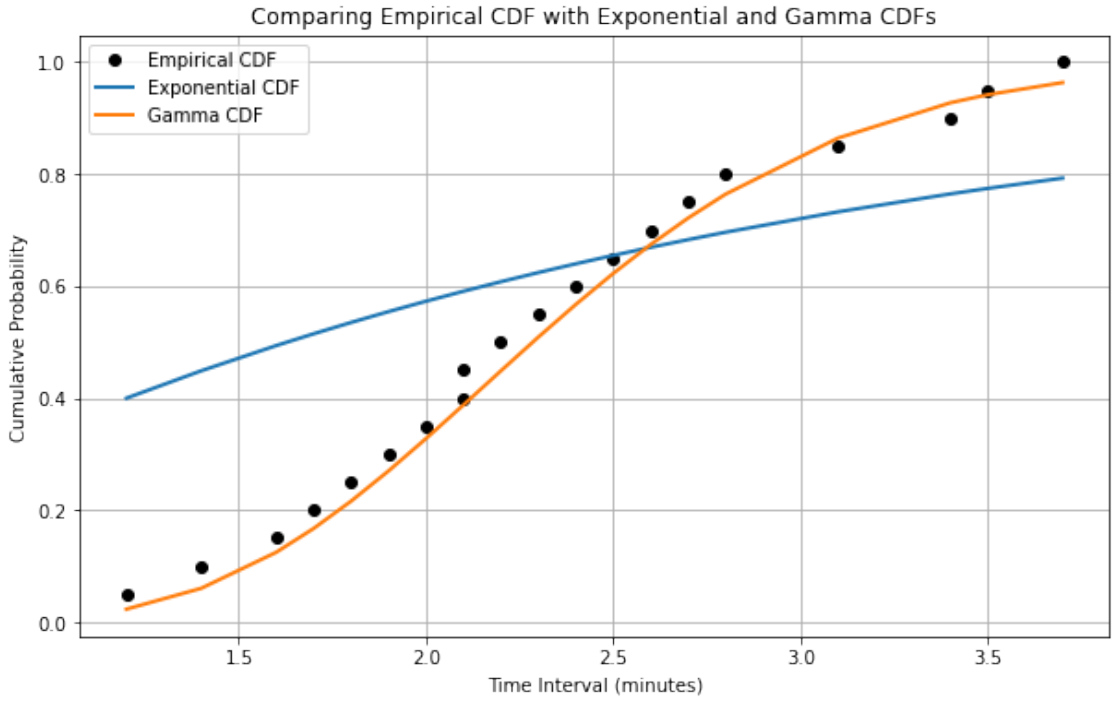
```
# PDFs
exp_pdf = expon.pdf(x, scale=1/lambda_exp)
gamma_pdf = gamma.pdf(x, a=alpha_gamma, scale=1/beta_gamma)

# Plot PDF curves
plt.plot(x, exp_pdf, label=f'Exponential Fit ( ={lambda_exp:.2f})', linewidth=2)
plt.plot(x, gamma_pdf, label=f'Gamma Fit ( ={alpha_gamma:.2f},  ={beta_gamma:.
 ↪2f})', linewidth=2)

plt.xlabel('Time Interval (minutes)')
plt.ylabel('Density')
plt.title('Histogram with Fitted Exponential and Gamma PDFs')
plt.legend()
plt.grid(True)
plt.show()
```




### 0.0.2 Comparing CDFs

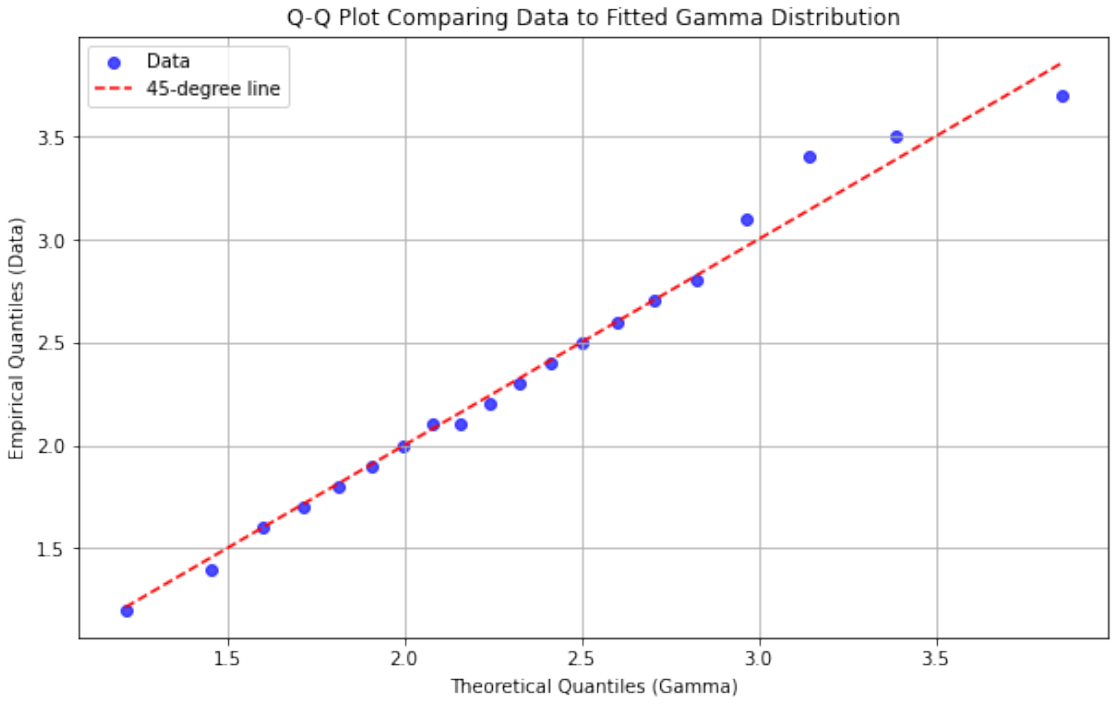
[3]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import expon, gamma

# Empirical CDF
data_sorted = np.sort(data)
```

```
n = len(data)
empirical_cdf = np.arange(1, n+1) / n

# Theoretical CDFs
exp_cdf = expon.cdf(data_sorted, scale=1/lambda_exp)
gamma_cdf = gamma.cdf(data_sorted, a=alpha_gamma, scale=1/beta_gamma)

plt.figure(figsize=(10, 6))
plt.plot(data_sorted, empirical_cdf, marker='o', linestyle='none',␣
 ↪label='Empirical CDF', color='black')
plt.plot(data_sorted, exp_cdf, label='Exponential CDF', linewidth=2)
plt.plot(data_sorted, gamma_cdf, label='Gamma CDF', linewidth=2)

plt.xlabel('Time Interval (minutes)')
plt.ylabel('Cumulative Probability')
plt.title('Comparing Empirical CDF with Exponential and Gamma CDFs')
plt.grid(True)
plt.legend()
plt.show()
```



### 0.0.3 Q-Q Plot for Gamma Distribution

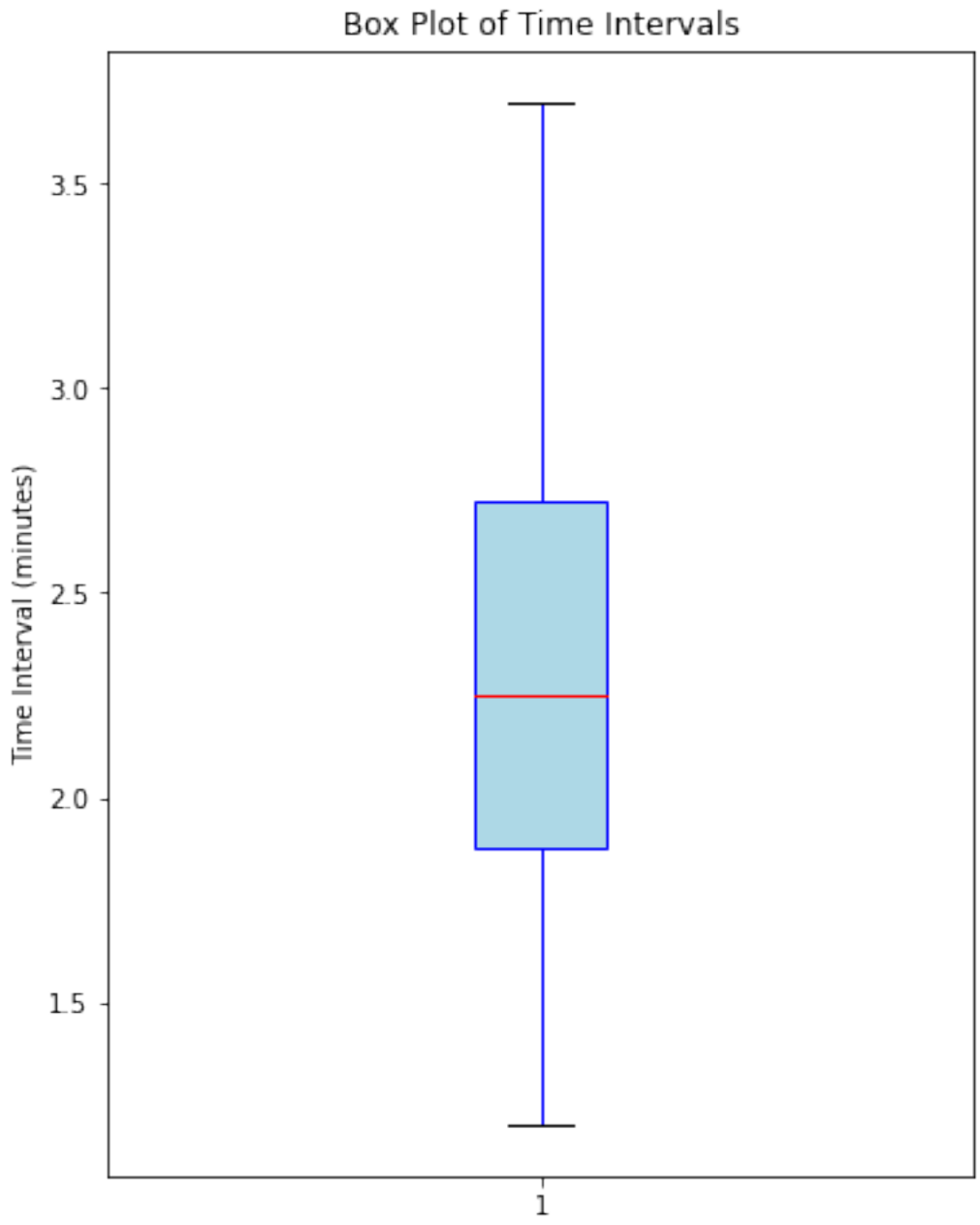
[4]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gamma

# Sort the data
data_sorted = np.sort(data)
n = len(data_sorted)

# Compute theoretical gamma quantiles
probabilities = (np.arange(1, n+1) - 0.5) / n
theoretical_quantiles = gamma.ppf(probabilities, a=alpha_gamma, scale=1/
 ↪beta_gamma)

plt.figure(figsize=(10, 6))
plt.scatter(theoretical_quantiles, data_sorted, color='blue', alpha=0.7,␣
 ↪label='Data')
plt.plot(theoretical_quantiles, theoretical_quantiles, 'r--', label='45-degree␣
 ↪line')

plt.xlabel('Theoretical Quantiles (Gamma)')
plt.ylabel('Empirical Quantiles (Data)')
plt.title('Q-Q Plot Comparing Data to Fitted Gamma Distribution')
plt.grid(True)
plt.legend()
plt.show()
```



### 0.0.4 Box Plot and Summary Statistics

[5]:
```
import numpy as np
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 8))
plt.boxplot(data, vert=True, patch_artist=True,␣
 ↪boxprops=dict(facecolor='lightblue', color='blue'),
            medianprops=dict(color='red'), whiskerprops=dict(color='blue'))
plt.ylabel('Time Interval (minutes)')
plt.title('Box Plot of Time Intervals')

# Calculate summary statistics
mean = np.mean(data)
median = np.median(data)
std_dev = np.std(data)
min_val = np.min(data)
max_val = np.max(data)
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)

# Print summary statistics
print("Summary Statistics:")
print(f"Mean: {mean:.2f} minutes")
print(f"Median: {median:.2f} minutes")
print(f"Std Dev: {std_dev:.2f} minutes")
print(f"Min: {min_val:.2f} minutes")
print(f"Q1: {q1:.2f} minutes")
print(f"Q3: {q3:.2f} minutes")
print(f"Max: {max_val:.2f} minutes")

plt.show()
```

```
Summary Statistics:
Mean: 2.35 minutes
Median: 2.25 minutes
Std Dev: 0.68 minutes
Min: 1.20 minutes
Q1: 1.88 minutes
Q3: 2.73 minutes
Max: 3.70 minutes
```


Box Plot of Time Intervals
Time Interval (minutes)
3.5
3.0
2.5
2.0
1.5
1

[ ]: