**Visualising Distributions and Testing for Statistical Significance**

There are even more powerful arguments we can make to convince our fellow doctors in clinic 1 of the virtues of handwashing. The first are statistics regarding the mean monthly death rate. The second are compelling visualisations to accompany the statistics.

Challenge 1: Calculate the Difference in the Average Monthly Death Rate

* What was the average percentage of monthly deaths before handwashing (i.e., before June 1847)?
* What was the average percentage of monthly deaths after handwashing was made obligatory?
* By how much did handwashing reduce the average chance of dying in childbirth in percentage terms?
* How do these numbers compare to the average for all the 1840s that we calculated earlier?
* How many times lower are the chances of dying after handwashing compared to before?

.

.

..

...

..

.

.

**Solution to Challenge 1**

A lot of statistical tests rely on comparing features of distributions, like the mean. We see that the average death rate before handwashing was 10.5%. After handwashing was made obligatory, the average death rate was 2.11%. The difference is massive. Handwashing decreased the average death rate by 8.4%, a 5x improvement. 😮

1. avg\_prob\_before = before\_washing.pct\_deaths.mean() \* 100
2. print(f'Chance of death during childbirth before handwashing: {avg\_prob\_before:.3}%.')
4. avg\_prob\_after = after\_washing.pct\_deaths.mean() \* 100
5. print(f'Chance of death during childbirth AFTER handwashing: {avg\_prob\_after:.3}%.')
7. mean\_diff = avg\_prob\_before - avg\_prob\_after
8. print(f'Handwashing reduced the monthly proportion of deaths by {mean\_diff:.3}%!')
10. times = avg\_prob\_before / avg\_prob\_after
11. print(f'This is a {times:.2}x improvement!')

Challenge 2: Using Box Plots to Show How the Death Rate Changed Before and After Handwashing

The statistic above is impressive, but how do we show it graphically? With a box plot we can show how the quartiles, minimum, and maximum values changed in addition to the mean.

* Use [NumPy's .where() function](https://numpy.org/doc/stable/reference/generated/numpy.where.html) to add a column to df\_monthly that shows if a particular date was before or after the start of handwashing.
* Then use plotly to create box plot of the data before and after handwashing.
* How did key statistics like the mean, max, min, 1st and 3rd quartile changed as a result of the new policy

.

.

..

...

..

.

.

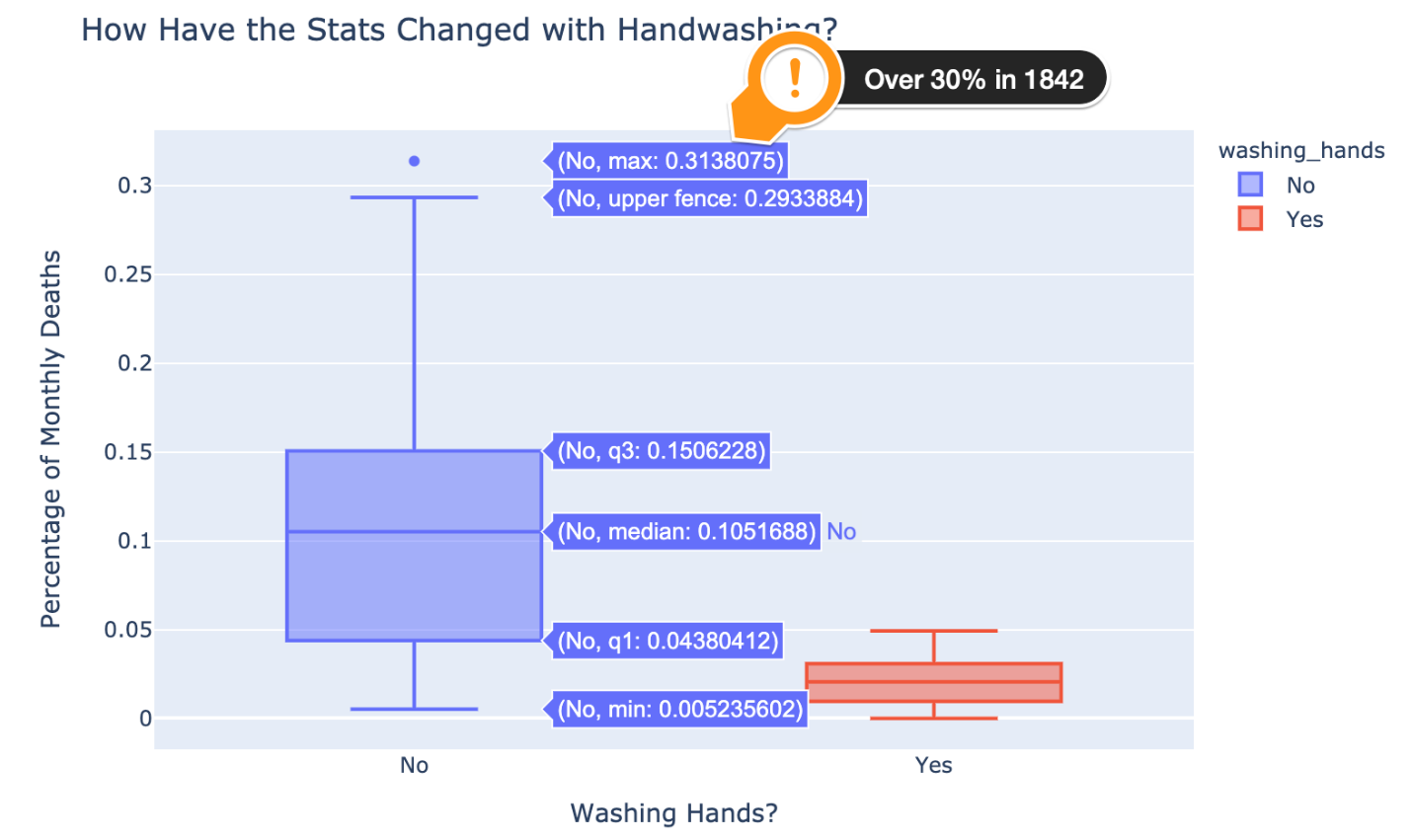
**Solution to Challenge 2**

The easiest way to create a box plot is to have a column in our DataFrame that shows the rows' "category" (i.e., was it before or after obligatory handwashing). NumPy allows us to easily test for a condition and add a column of data.

1. df\_monthly['washing\_hands'] = np.where(df\_monthly.date < handwashing\_start, 'No', 'Yes')

Now we can use plotly:

1. box = px.box(df\_monthly,
2. x='washing\_hands',
3. y='pct\_deaths',
4. color='washing\_hands',
5. title='How Have the Stats Changed with Handwashing?')
7. box.update\_layout(xaxis\_title='Washing Hands?',
8. yaxis\_title='Percentage of Monthly Deaths',)
10. box.show()



The plot shows us the same data as our Matplotlib chart, but from a different perspective. Here we also see the massive spike in deaths in late 1842. Over 30% of women who gave birth that month died in hospital. What we also see in the box plot is how not only did the average death rate come down, but so did the overall range - we have a lower max and 3rd quartile too. Let's take a look at a histogram to get a better sense of the distribution.

Challenge 3: Use Histograms to Visualise the Monthly Distribution of Outcomes

Create a [plotly histogram](https://plotly.com/python/histograms/" \t "_blank) to show the monthly percentage of deaths.

* Use docs to check out the available parameters. Use the [color parameter](https://plotly.github.io/plotly.py-docs/generated/plotly.express.histogram.html) to display two overlapping histograms.
* The time period of handwashing is shorter than not handwashing. Change histnorm to percent to make the time periods comparable.
* Make the histograms slightly transparent
* Experiment with the number of bins on the histogram. Which number works well in communicating the range of outcomes?
* Just for fun, display your box plot on the top of the histogram using the marginal parameter

.

.

..

...

..

.

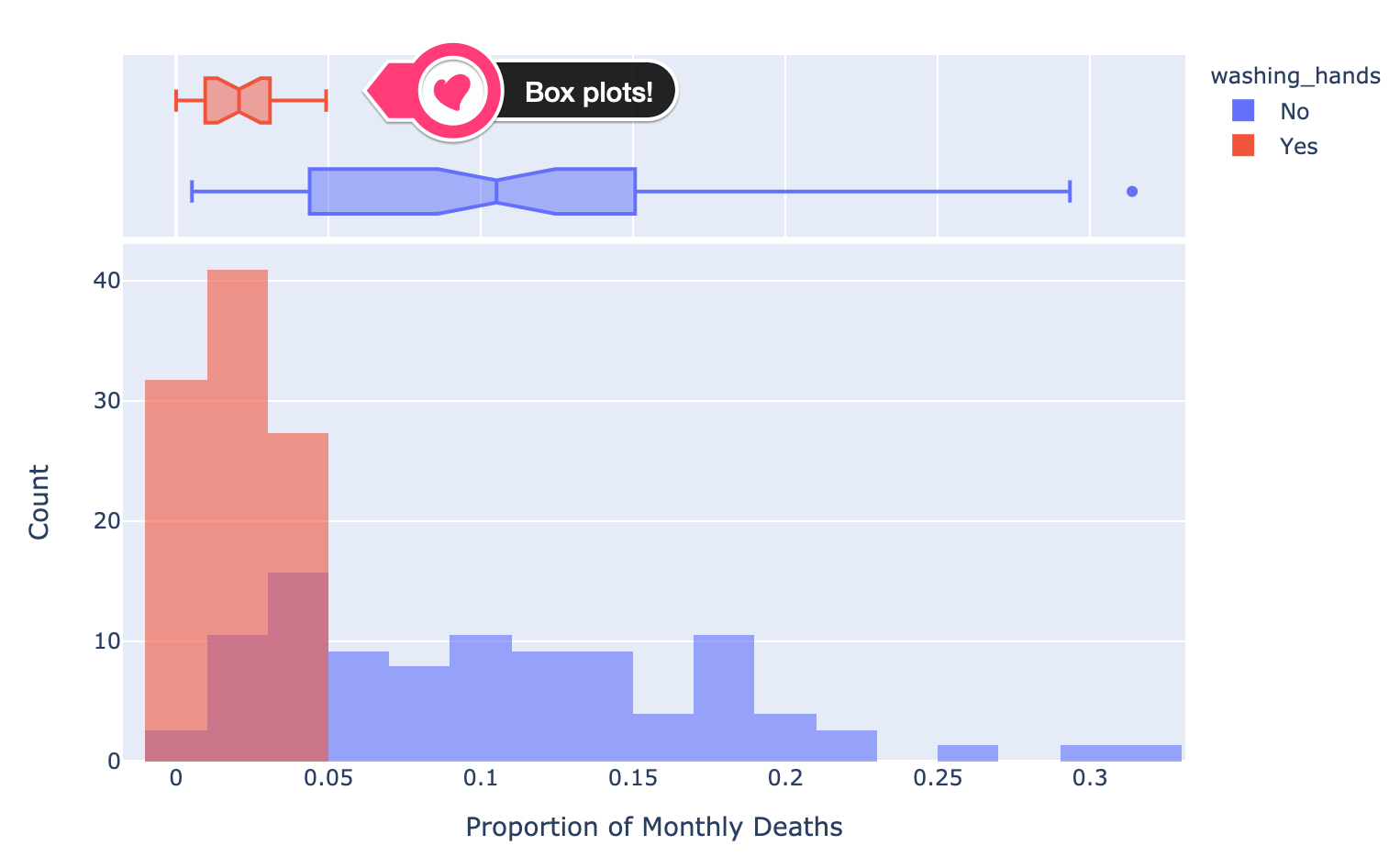
.

**Solution to Challenge 3**

To create our histogram, we once again make use of the color parameter. This creates two separate histograms for us. When we set the opacity to 0.6 or so we can clearly see how the histograms overlap. The trick to getting a sensible-looking histogram when you have a very different number of observations is to set the histnorm to 'percent'. That way the histogram with more observations won't completely overshadow the shorter series.

1. hist = px.histogram(df\_monthly,
2. x='pct\_deaths',
3. color='washing\_hands',
4. nbins=30,
5. opacity=0.6,
6. barmode='overlay',
7. histnorm='percent',
8. marginal='box',)
10. hist.update\_layout(xaxis\_title='Proportion of Monthly Deaths',
11. yaxis\_title='Count',)
13. hist.show()

I quite like how in plotly we can display our box plot from earlier at the top.



Now, we have only about 98 data points or so, so our histogram looks a bit jagged. It's not a smooth bell-shaped curve. However, we can estimate what the distribution would look like with a Kernel Density Estimate (KDE).

Challenge 4: Use a Kernel Density Estimate (KDE) to visualise a smooth distribution

Use [Seaborn's .kdeplot()](https://seaborn.pydata.org/generated/seaborn.kdeplot.html" \t "_blank) to create two kernel density estimates of the pct\_deaths, one for before handwashing and one for after.

* Use the shade parameter to give your two distributions different colours.
* What weakness in the chart do you see when you just use the default parameters?
* Use the clip parameter to address the problem.

.

.

..

...

..

.

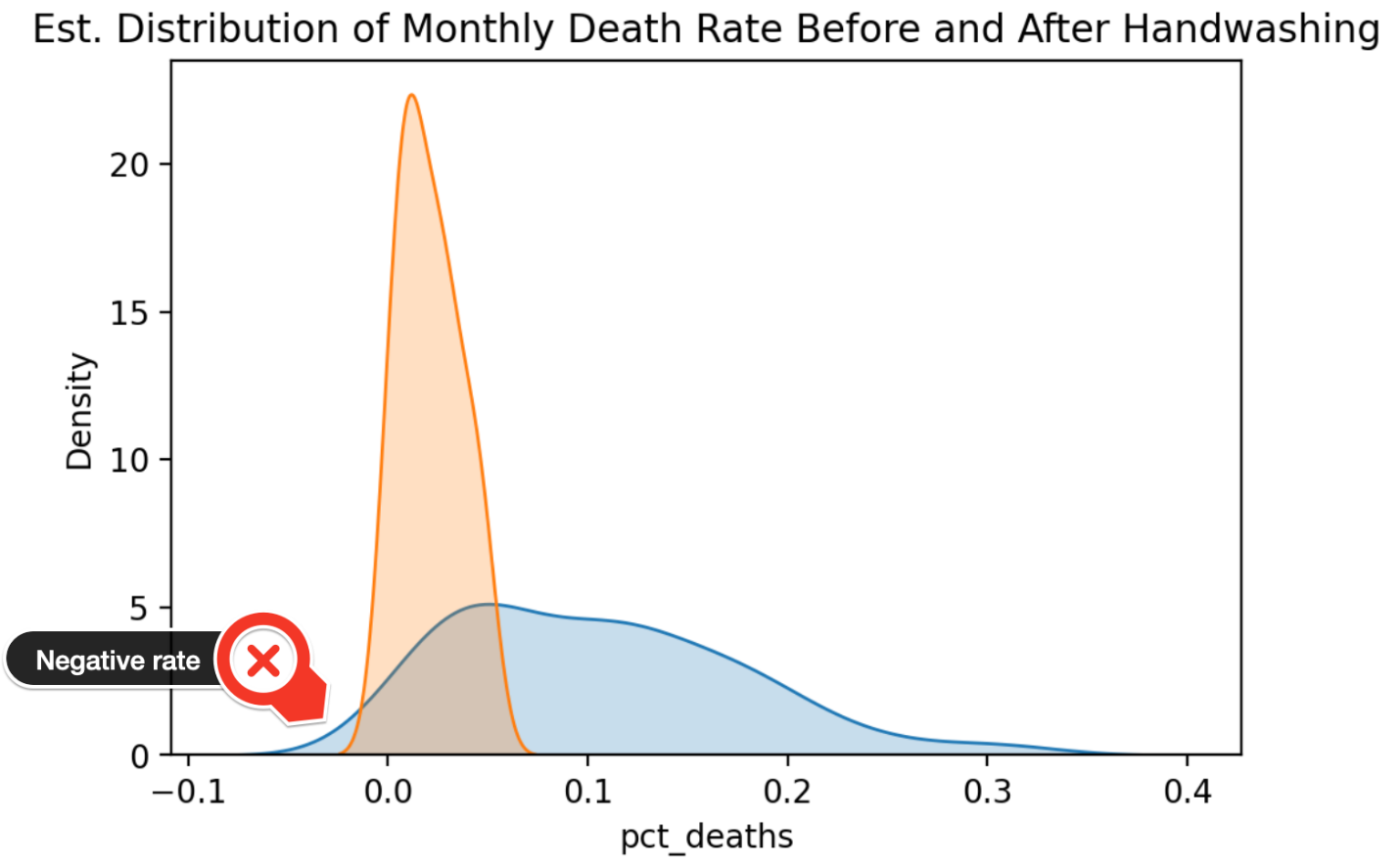
.

**Solution to Challenge 4**

To create two bell-shaped curves of the estimated distributions of the death rates we just call .kdeplot() twice.

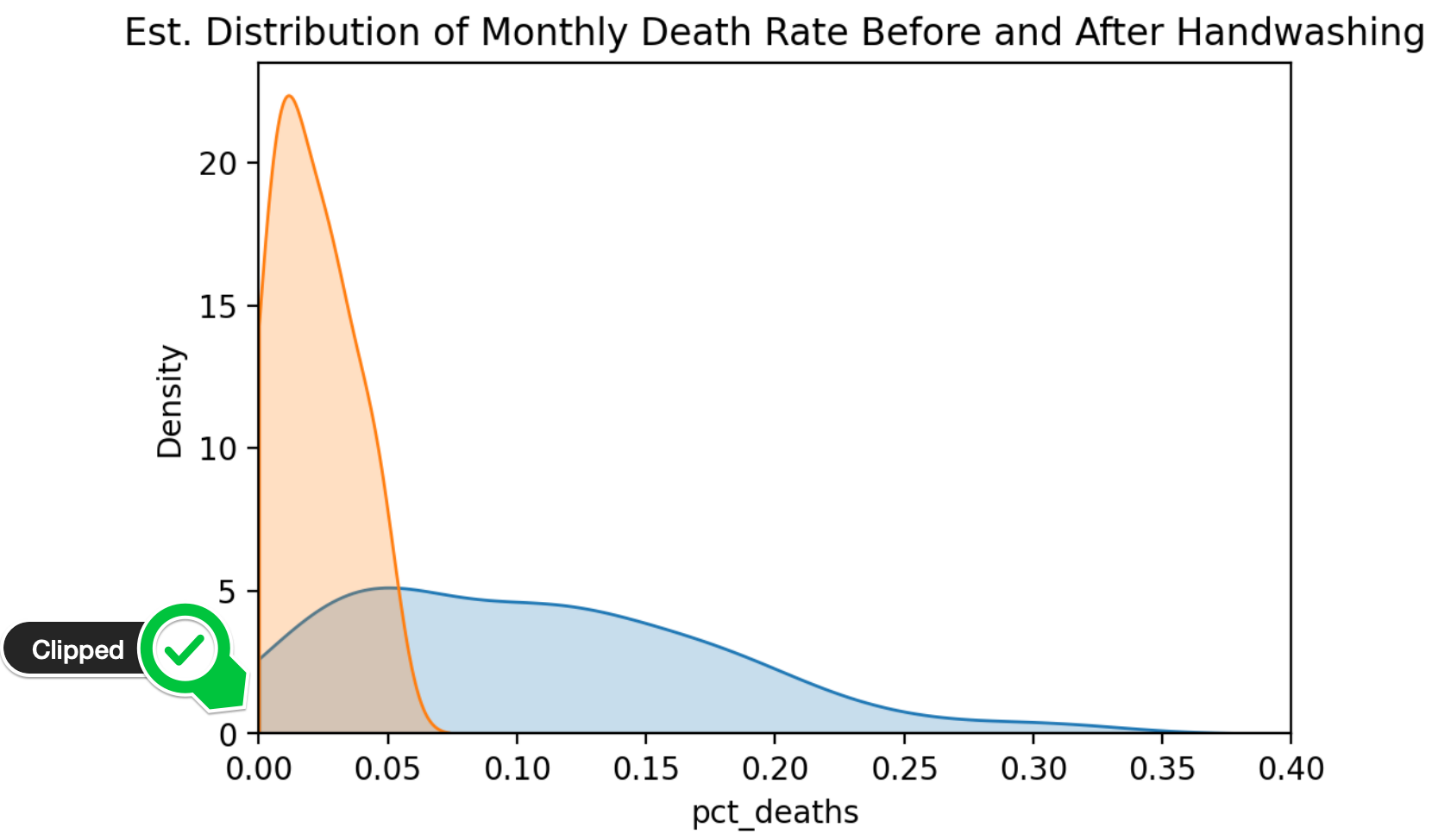
1. plt.figure(dpi=200)
2. # By default the distribution estimate includes a negative death rate!
3. sns.kdeplot(before\_washing.pct\_deaths, shade=True)
4. sns.kdeplot(after\_washing.pct\_deaths, shade=True)
5. plt.title('Est. Distribution of Monthly Death Rate Before and After Handwashing')
6. plt.show()

However, the problem is that we end up with a negative monthly death rate on the left tail. The doctors would be very surprised indeed if a corpse came back to life after an autopsy! 🧟‍♀️



The solution is to specify a lower bound of 0 for the death rate.

1. plt.figure(dpi=200)
2. sns.kdeplot(before\_washing.pct\_deaths,
3. shade=True,
4. clip=(0,1))
5. sns.kdeplot(after\_washing.pct\_deaths,
6. shade=True,
7. clip=(0,1))
8. plt.title('Est. Distribution of Monthly Death Rate Before and After Handwashing')
9. plt.xlim(0, 0.40)
10. plt.show()



Now that we have an idea of what the two distributions look like, we can further strengthen our argument for handwashing by using a statistical test. We can test whether our distributions ended up looking so different purely by chance (i.e., the lower death rate is just an accident) or if the 8.4% difference in the average death rate is **statistically significant**.

Challenge 5: Use a T-Test to Show Statistical Significance

Use a t-test to determine if the differences in the means are statistically significant or purely due to chance.

If the p-value is less than 1% then we can be 99% certain that handwashing has made a difference to the average monthly death rate.

* Import stats from scipy
* Use the [.ttest\_ind() function](https://docs.scipy.org/%5Ddoc/scipy/reference/generated/scipy.stats.ttest_ind.html) to calculate the t-statistic and the p-value
* Is the difference in the average proportion of monthly deaths statistically significant at the 99% level?

.

.

..

...

..

.

.

**Solution to Challenge 5**

The first step is to import stats from scipy

1. import scipy.stats as stats

When we calculate the p\_value we see that it is 0.0000002985 or .00002985% which is far below even 1%. In other words, the difference in means is highly statistically significant and we can go ahead on publish our research paper 😊

1. t\_stat, p\_value = stats.ttest\_ind(a=before\_washing.pct\_deaths,
2. b=after\_washing.pct\_deaths)
3. print(f'p-palue is {p\_value:.10f}')
4. print(f't-statstic is {t\_stat:.4}')