

Victor Poughon

[Home](#) [Blog](#) [GitHub](#) [LinkedIn](#)

All clocks are 30 seconds late

06 Jan, 2025

OK, this is going to sound crazy: I believe all clocks are 30 seconds late.

Let's get a few things out of the way: this is neither about time zones, nor leap seconds. It's not about synchronization of clocks, and it's not about relativity or any obscure corner of physics. I'm talking about everyday clocks. The ones that you might have around in your house, like this one that sits on my desk:



Or this one that stands in London:



(Photo from Wikipedia)

Or even the clock on your phone! Maybe they're all correctly set very precisely to a very accurate reference clock, but here I will argue that they are still exactly 30 seconds late. In other words:

It would be more accurate if they were thirty seconds ahead.

What are you on about?

All the above clocks have one thing in common: they don't show seconds. They truncate down to the nearest lower whole minute, right? So when it's actually 14:15:45, they'll show 14:15. And when the actual time goes from 14:15:59 to 14:16:00, then that's when your clock changes from 14:15 to 14:16.

They apply the [floor function](#)!

Ok great. Now hear me out! Let's compute the average error between a usual (truncating) clock, and the precise current time.

```
import datetime
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

```

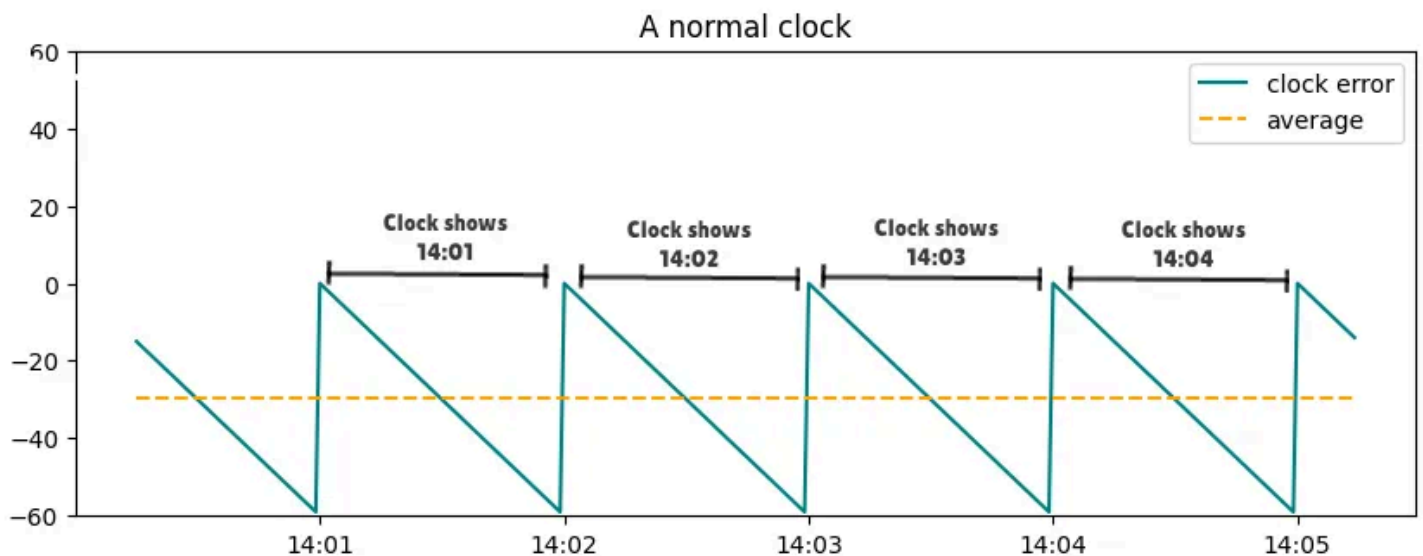
# A datetime array from 14:00:15pm for 300 seconds
true_time = np.array([datetime.datetime(2024, 12, 20, 14, 00, 15) + datetime.timedelta(sec

# The clock truncates seconds
shown_time = [t.replace(second=0) for t in true_time]

# The clock error is the shown time minus the true time
error = [dt.total_seconds() for dt in (shown_time - true_time)]

fig, ax = plt.subplots(figsize=(10, 3.5))
ax.plot(true_time, error, label="clock error", color="teal")
ax.hlines(np.mean(error), xmin=true_time[0], xmax=true_time[-1], label="average", linestylel
ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
ax.set_ylim([-60, 60])
ax.set_title("A normal clock")
ax.legend()

```



- Sometimes the error is 1 or 2 seconds, like when the clock shows 14:02 and it's actually 14:02:01
- Sometimes the error is almost 60 seconds, like when the clock shows 14:02 and it's actually 14:02:57

So, the average error of a truncating clock is 30 seconds. Let that sink in for a moment: **the average error of a clock is 30 seconds**. If clocks rounded to the nearest minute instead of truncating, the average error would be 0. Therefore all clocks are 30 seconds late!

Similarly of course, all clocks that show seconds but not milliseconds are half a second late.

```

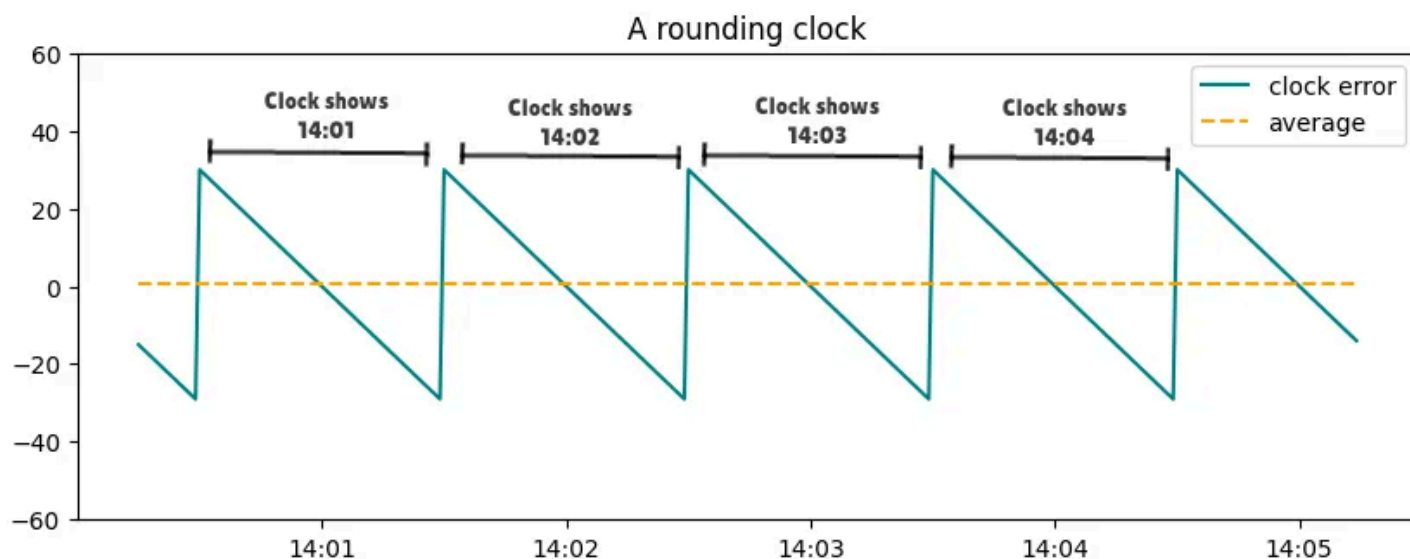
# The rounding clock! aka +30s
shown_time = [(t + datetime.timedelta(seconds=30)).replace(second=0) for t in true_time]

# The clock error is the shown time minus the true time

```

```
error = [dt.total_seconds() for dt in (shown_time - true_time)]
```

```
fig, ax = plt.subplots(figsize=(10, 3.5))  
ax.plot(true_time, error, label="clock error", color="teal")  
ax.hlines(np.mean(error), xmin=true_time[0], xmax=true_time[-1], label="average", linestyle='--')  
ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))  
ax.set_ylim([-60, 60])  
ax.set_title("A rounding clock")  
ax.legend()
```



This is just a convention

But this is just a convention! Truncating, rounding or even ceiling are all valid, as long as we pick one and stick to it. By your logic we should say that it's Tuesday as soon as it's one second past noon on Monday.

Well... kind of.

It would be weird if we rounded for years, months and days, that's for sure. I think most people think of those scales as intervals. In other words, July is a period of time, with a start and an end. So are years, centuries, seasons. We are inside of it or outside.

Is it the same for the current minute of the current hour ? I'm not so sure.

Basically I'm arguing that rounding for clocks would be more useful than flooring. This is especially apparent when you're trying to calculate "how much time until my next meeting?", and your next meeting is at noon. If it's 11:55, you would usually mentally subtract and conclude: the meeting is in 5 minutes. That's how I always do it myself anyway! But the most probable estimate given the available information is actually 4'30"!

If you had a rounding clock instead, 5 minutes would be the correct estimation.

In fact, as we go down the various scales of time all the way from years, months, days and we reach hours, that's when we start truncating! At hours! For example: if it's 10:43 and you are asked the time and (for some context appropriate reason) you reply with just hours, you would say it's 11! Personally, I would never say that it's 10 if the clock shows anything past 10:30. I realize that's probably very different in other cultures, though! And I'd be interested to learn about how different cultures approximate time in language.

So, looking at the various scales of time and how I mentally view them:

- **Years:** truncate. If the full date has 2019 in it, I'll say "the year is 2019" - all the way to the end. Saying "it's 2020" in September 2019 is super weird.
- **Months:** truncate. It's January all the way until it's February.
- **Days:** truncate. It gets a little weird if it's past midnight and I haven't slept yet. Is tomorrow today? Or is it the day after that? I think collectively we still haven't quite figured that one out yet. And that's ok. Kind of like how "next Friday" becomes ambiguous on Thursdays. Some cultures use times beyond 24:00, like 25:30 to indicate one hour and a half past midnight, and I think that's lovely!
- **Hours:** That's when mentally, I switch to rounding! At 15:48 I definitely feel like it's pretty much 16:00.
- **Minutes:** At 15:48:57, I definitely feel like it's 15:49! But clocks switch back to flooring for for some reason! But mentally we round! This is why I feel like all clocks are 30 seconds late: because the truncating convention doesn't match my intuition at the minute scale.
- **Seconds and below:** definitely rounding.

Please someone tell me I'm not crazy 😞