

# 1. Data Ingestion & Orchestration

```
In[*]:= (*Automation through terminal and shell script to move CSV file to target folder*)
In[*]:= terminal[string_String] := RunProcess[{"zsh", "-i", "-c", string}];
terminal["updateManu"];
```

## 2. Data Transformation & Validation

---

### Data & Sample

```
In[*]:= (*Import the data*)
In[*]:= data = Import["/Users/andrefreitas/andre/Manuela/manu.csv", "RawData"];
In[*]:= (*Small sample of the data filtered by type "sleep"*)
In[*]:= {First[data], Select[data, Function[#[[1]] === "Sleep"]][[1 ;; 3]]} // TableForm
Out[*] // TableForm =
```

Type	Start	End	Duration	Start Condition
Sleep	Sleep	Sleep		
2026-01-24 18:28	2026-01-24 14:08	2026-01-24 10:23		
2026-01-25 06:57	2026-01-24 14:53	2026-01-24 11:05		
12:29	00:45	00:42		

---

### Variables1

```
In[*]:= (*Putting data into Tabular format and
selecting only the relevant columns (first 4)*)
In[*]:= sleepTabular1 =
Tabular[Select[data, Function[#[[1]] === "Sleep"]], First[data]][[All, 1 ;; 4]]
In[*]:= (*adding a column 'Date' to indicate the day in which the sleeping started*)
```

```
In[*]:= sleepTabular2 = TransformColumns[sleepTabular1,
  {"Type", "Date" → Function[StringSplit[#Start, " "] /. {a_, b_} → a]]]
```

Out[\*] =

	Type	Date	Start	End	Duration
12	Sleep	2026-01-21	2026-01-21 18:28	2026-01-22 07:22	12:54
13	Sleep	2026-01-21	2026-01-21 13:33	2026-01-21 14:11	00:38
14	Sleep	2026-01-21	2026-01-21 09:44	2026-01-21 10:25	00:41
15	Sleep	2026-01-20	2026-01-20 19:13	2026-01-21 07:16	12:03
16	Sleep	2026-01-20	2026-01-20 14:53	2026-01-20 15:27	00:34
17	Sleep	2026-01-20	2026-01-20 10:38	2026-01-20 11:25	00:47
18	Sleep	2026-01-20	2026-01-20 07:04	2026-01-20 08:04	01:00

```
In[*]:= (*Variable that list all days in string ISODate format from when sleeping
  data started to be collected (17/05/2025) up to the current day*)

dates =
  (DateString[#, "ISODate"] & /@ DateRange[DateObject["2025-05-17"], Today, "Day"]);
```

## Functions

```
In[*]:= (*Function that adds column "Duration2" and "Time Range" which gives
  the duration of each sleeping session restricted to a specific day*)
```

```
In[*]:= f[date_String] := TransformColumns[Select[sleepTabular2,
  Function[(StringSplit[#End, " "] /. {a_, b_} → a) == date || #Date == date]],
  {"Duration2" → Function[If[(StringSplit[#End, " "] /. {a_, b_} → a) ≠ date,
    DateObject[StringJoin[date, " 23:59"]] - DateObject[#Start],
    If[(StringSplit[#Start, " "] /. {a_, b_} → a) ≠ date,
      DateObject[#End] - DateObject[StringJoin[date, " 00:01"]], DateObject[#End] -
      DateObject[#Start]], DateObject[#End] - DateObject[#Start]]],
  "TimeRange" → Function[If[(StringSplit[#End, " "] /. {a_, b_} → a) ≠ date,
    (StringSplit[#Start, " "] /. {a_, b_} → b) <> "-23:59",
    If[(StringSplit[#Start, " "] /. {a_, b_} → a) ≠ date,
      "00:00-" <> (StringSplit[#End, " "] /. {a_, b_} → b),
      (StringSplit[#Start, " "] /. {a_, b_} → b) <> "-" <>
      (StringSplit[#End, " "] /. {a_, b_} → b)]]]]]
```

```
In[*]:= (*example: the column "duration2" displays the amount of sleep only up until 23:
  59 on the 21/05/2025*)
```

```

In[*]:= f["2025-05-21"]

In[*]:= (*testing I can use the function to calculate total amount of sleep from 00:
        00 to 23:59 on any given day*)

In[*]:= f["2025-05-21"][[All, "Duration2"]] // Normal // Total
Out[*]=
932 min

In[*]:= (*Function Gives the total amount of sleep for the inputed day,
        includint partial days*)

In[*]:= totalDaySleep[date_String] :=
        (f[date] [[All, 6]] // Normal // Total) /. Quantity[x_, "Minutes"] →
        Quantity[MixedMagnitude[{0, x}], MixedUnit[{"Hours", "Minutes"}]]

In[*]:= (*Example*)

In[*]:= totalDaySleep["2026-01-25"]
Out[*]=
6h 56min

In[*]:= (*Function gives last x days (including current day) as string "ISODate"*)

In[*]:= listOfDates[daysback_Integer] := (DateString[#, "ISODate"] & /@
        DateRange[DatePlus[Today, -daysback], Yesterday, "Day"])

In[*]:= (*Example*)

In[*]:= listOfDates[7]
Out[*]=
{2026-01-18, 2026-01-19, 2026-01-20, 2026-01-21, 2026-01-22, 2026-01-23, 2026-01-24}

In[*]:= (*Function Gives the total amount of sleep per
        day in the last "daysback"s not including current day*)

In[*]:= totalDaySleep2[daysback_Integer] := Rule @@@ Partition[
        Riffle[listOfDates[daysback], Map[totalDaySleep[#] &, listOfDates[daysback]]], 2]

In[*]:= (*Example*)

In[*]:= totalDaySleep2[3]
Out[*]=
{2026-01-22 → 13h 52min, 2026-01-23 → 14h 28min, 2026-01-24 → 14h 6min}

```

---

## Variables 2

Defining secondary feature set based on the primary variables established in Variables 1 subsection and on the functions established in the Functions section.

```

In[*]:= (*All data on total sleep (in hours and minutes) per day as an ASSOCIATION*)

```

```

In[*]:= sleepData = Rule @@@ Partition[Riffle[dates, Map[totalDaySleep, dates]], 2]
In[*]:= (*All data on total sleep per day as list of decimal hours*)
In[*]:= sleepData2 = Drop[Function[UnitConvert[#, "Hours"]]/@Values[sleepData], -1] // N

```

# Insights & Visual Monitoring

## Total Hours per day

```

In[*]:= (*Function to manually calculate the total amount of sleep in a day,
        for partial days calculations*)

In[*]:= manualSleeping[hours_List, minutes_List] :=
        Quantity[MixedMagnitude[{Total[Join[hours, {IntegerPart[Total[minutes] / 60}]]],
        FractionalPart[Total[minutes] / 60] * 60}], MixedUnit[{"Hours", "Minutes"}]]

In[*]:= (*Example*)

In[*]:= (*2026-01-24*)

In[*]:= manualSleeping[{1, 4}, {23, 34}]
Out[*]=
        5h 57min

In[*]:= (*Total sleep in current day - Almost always partial*)

In[*]:= totalDaySleep[DateString[Today, "ISODate"]]
Out[*]=
        6h 56min

In[*]:= (*Total sleep per day from the last 8 days*)

In[*]:= totalDaySleep2[7]
Out[*]=
        {2026-01-18 → 13h 41min, 2026-01-19 → 13h 50min,
        2026-01-20 → 13h 25min, 2026-01-21 → 14h 5min,
        2026-01-22 → 13h 52min, 2026-01-23 → 14h 28min, 2026-01-24 → 14h 6min}

In[*]:= (*Average sleep from the last 7 days*)

```

```

In[*]:= (Mean[
  UnitConvert[#, "Minutes"] & /@
  (⌘[All, 2]])
// N)
/ 60.0) /. Quantity[a_, "Minutes"] =>
Quantity[MixedMagnitude[{IntegerPart[a], IntegerPart[FractionalPart[a] * 60]}],
MixedUnit[{"Hours", "Minutes"}]]

Out[*]:=
13h 55min

```

## Naps per day

```

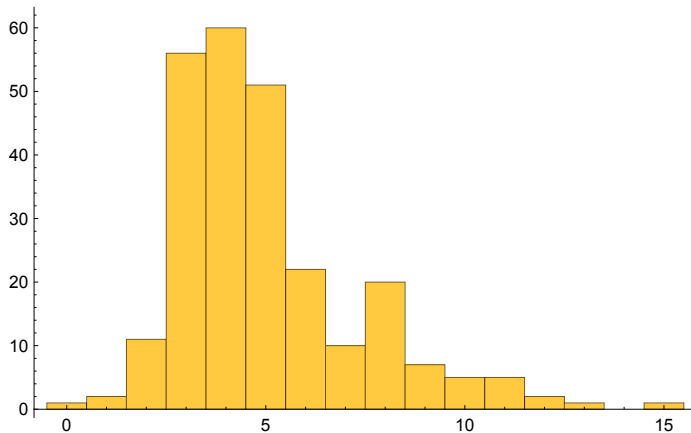
In[*]:= (*tracking the number of naps (under 120 min) per day*)

In[*]:= Function[Length[Select[f[#], Function[#Duration2 < 120]]]] /@ dates

In[*]:= Histogram[⌘]

Out[*]:=

```



## Distribution of Naps along specific days

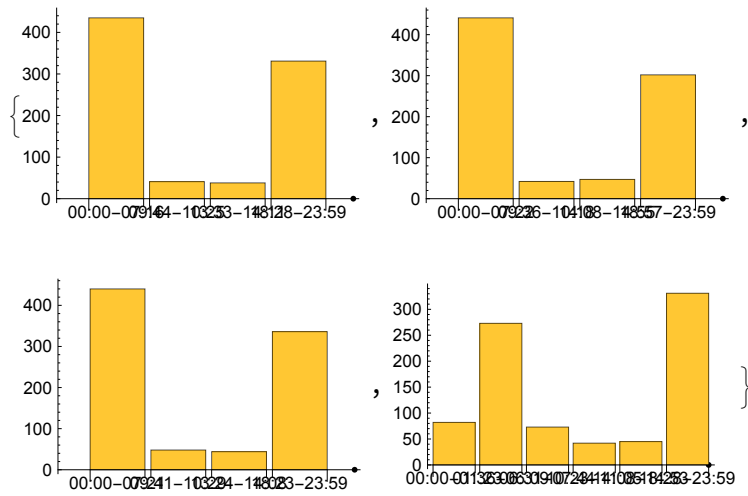
```

In[*]:= barChartSleep[date_String] :=
  BarChart[f[date] ⌘[All, "Duration2"] // QuantityMagnitude // Normal // Reverse,
  ChartLabels -> (f[date] ⌘[All, "TimeRange"] // Normal // Reverse)]

```

```
In[*]:= barChartSleep[#] & /@ listOfDates[4]
```

```
Out[*]:=
```

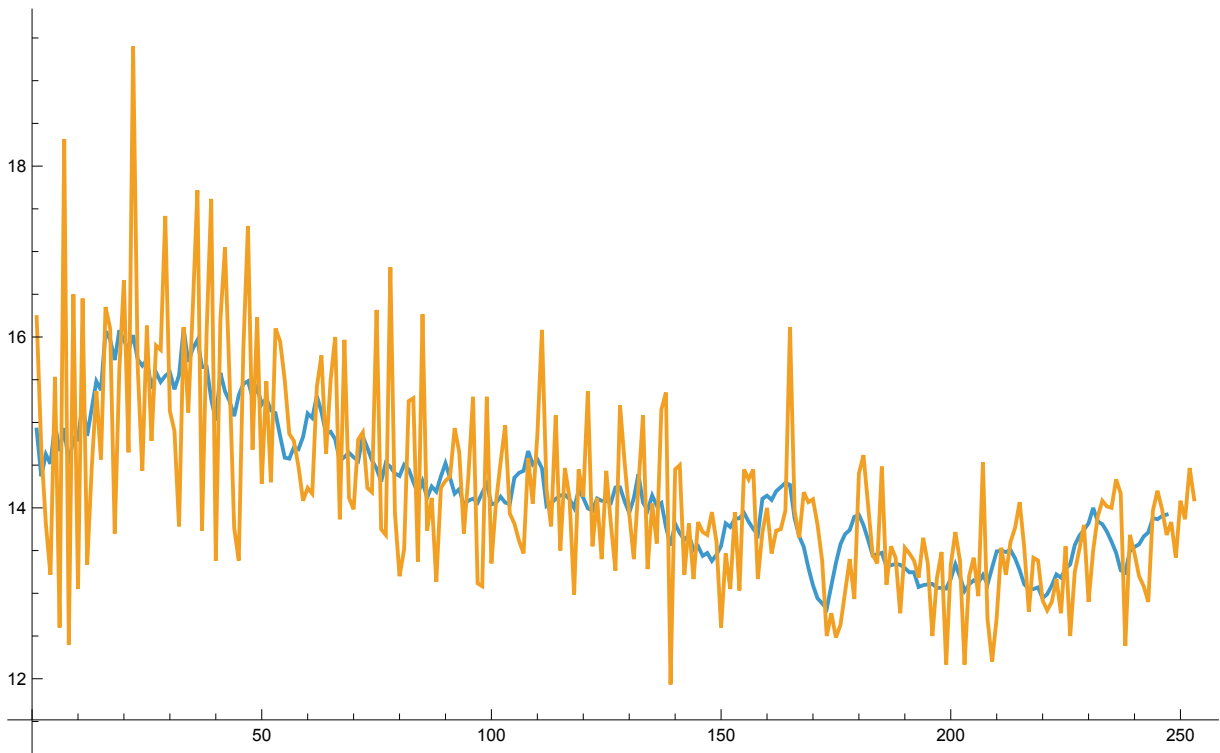


## Moving average total hours per day

```
In[*]:= (*Moving average for 7 days against raw data*)
```

```
In[*]:= ListLinePlot[{MovingAverage[sleepData2, 7], sleepData2}]
```

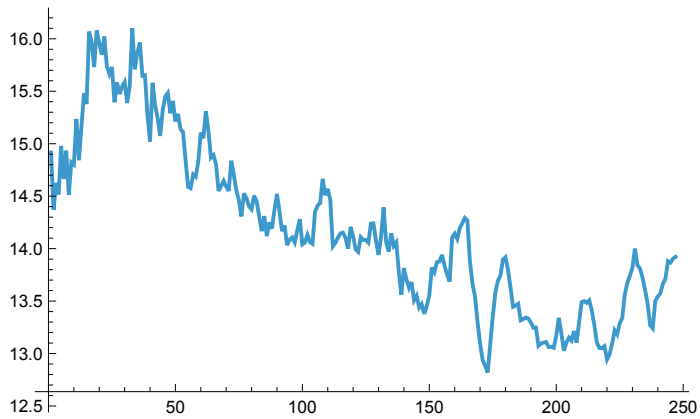
```
Out[*]:=
```



```
In[*]:= (*Moving average only*)
```

```
In[*]:= ListLinePlot[MovingAverage[sleepData2, 7]]
```

```
Out[*]=
```



```
In[*]:= (*Fitting the data with ListFitPlot[]*)
```

```
In[*]:= ListFitPlot[sleepData2[[1 ;; -2]], PlotRange -> {Automatic, {12, 20}}]
```

```
Out[*]=
```

