

# 1. Data Ingestion & Orchestration

```
In[1]:= (*Automation through terminal and shell script to move CSV file to target folder*)  
In[2]:= terminal[string_String] := RunProcess[{"zsh", "-i", "-c", string}];  
terminal["updateManu"];
```

# 2. Data Transformation & Validation

## Data & Sample

```
In[1]:= (*Import the data*)  
In[2]:= data = Import["/Users/andrefreitas/andre/Manuela/manu.csv", "RawData"];  
In[3]:= (*Small sample of the data filtered by type "sleep"*)  
In[4]:= {First[data], Select[data, Function[#\[Equal] "Sleep"]][1 ;; 3]} // TableForm  
Out[4]//TableForm=
```

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

## Variables1

```
In[1]:= (*Putting data into Tabular format and  
selecting only the relevant columns (first 4)*)  
In[2]:= sleepTabular1 =  
Tabular[Select[data, Function[#\[Equal] "Sleep"]], First[data]]\[All, 1 ;; 4]  
In[3]:= (*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[1]:= f["2025-05-21"]

In[2]:= (*testing I can use the function to calculate total amount of sleep from 00:
          00 to 23:59 on any given day*)

In[3]:= f["2025-05-21"][[All, "Duration2"]] // Normal // Total
Out[3]=
932 min

In[4]:= (*Function Gives the total amount of sleep for the inputed day,
           includint partial days*)

In[5]:= totalDaySleep[date_String] :=
  (f[date][[All, 6]] // Normal // Total) /. Quantity[x_, "Minutes"] \[Rule]
    Quantity[MixedMagnitude[{0, x}], MixedUnit[{"Hours", "Minutes"}]]

In[6]:= (*Example*)

In[7]:= totalDaySleep["2026-01-25"]
Out[7]=
6h 56min

In[8]:= (*Function gives last x days (including current day) as string "ISODate"*)
In[9]:= listOfDates[daysback_Integer] := (DateString[#, "ISODate"] & /@
  DateRange[DatePlus[Today, -daysback], Yesterday, "Day"])

In[10]:= (*Example*)

In[11]:= listOfDates[7]
Out[11]=
{2026-01-18, 2026-01-19, 2026-01-20, 2026-01-21, 2026-01-22, 2026-01-23, 2026-01-24}

In[12]:= (*Function Gives the total amount of sleep per
           day in the last "daysback"s not including current day*)

In[13]:= totalDaySleep2[daysback_Integer] := Rule @@@ Partition[
  Riffle[listOfDates[daysback], Map[totalDaySleep[#] &, listOfDates[daysback]]], 2]

In[14]:= (*Example*)

In[15]:= totalDaySleep2[3]
Out[15]=
{2026-01-22 \[Rule] 13h 52min, 2026-01-23 \[Rule] 14h 28min, 2026-01-24 \[Rule] 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[1]:= (*All data on total sleep (in hours and minutes) per day as an ASSOCIATION*)
```

```
In[1]:= sleepData = Rule @@ Partition[Riffle[dates, Map[totalDaySleep, dates]], 2]
In[2]:= (*All data on total sleep per day as list of decimal hours*)
In[3]:= sleepData2 = Drop[Function[UnitConvert[#, "Hours"]], -1] // N
```

# Insights & Visual Monitoring

---

## Total Hours per day

```
In[1]:= (*Function to manually calculate the total amount of sleep in a day,
for partial days calculations*)
In[2]:= manualSleeping[hours_List, minutes_List] :=
Quantity[MixedMagnitude[{Total[Join[hours, {IntegerPart[Total[minutes] / 60]}]],

FractionalPart[Total[minutes] / 60] * 60}], MixedUnit[{"Hours", "Minutes"}]]
In[3]:= (*Example*)
In[4]:= (*2026-01-24*)
In[5]:= manualSleeping[{1, 4}, {23, 34}]
Out[5]=
5h 57min

In[6]:= (*Total sleep in current day - Almost always partial*)
In[7]:= totalDaySleep[DateString[Today, "ISODate"]]
Out[7]=
6h 56min

In[8]:= (*Total sleep per day from the last 8 days*)
In[9]:= totalDaySleep2[7]
Out[9]=
{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[10]:= (*Average sleep from the last 7 days*)
```

```
In[1]:= (Mean[
  UnitConvert[#, "Minutes"] & /@
  (#[[All, 2]])]
 // N)
/60.0 /. Quantity[a_, "Minutes"] :>
Quantity[MixedMagnitude[{IntegerPart[a], IntegerPart[FractionalPart[a]*60]}],
 MixedUnit[{"Hours", "Minutes"}]]
```

Out[1]= 13h 55min

---

## Naps per day

```
In[2]:= (*tracking the number of naps (under 120 min) per day*)
In[3]:= Function[Length[Select[f[#], Function[#Duration2 < 120]]]] /@ dates
In[4]:= Histogram[dates]
```

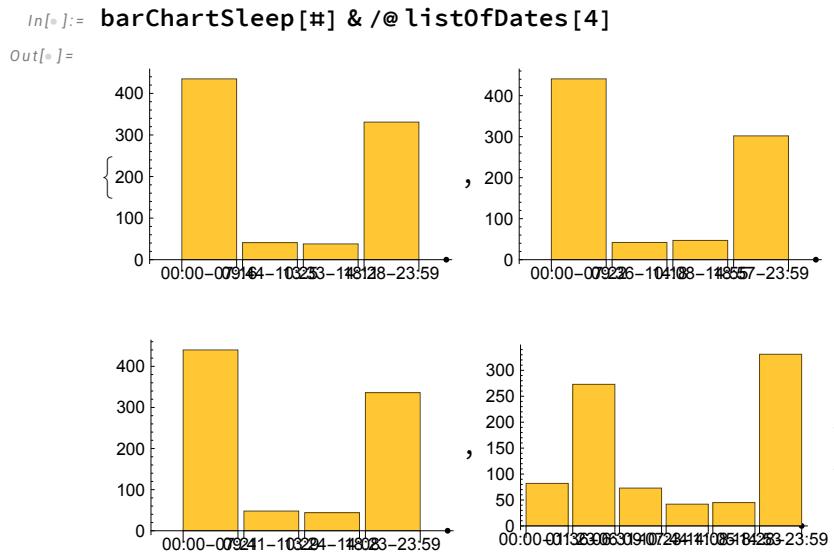
Out[4]=

Naps	Frequency
0	1
1	2
2	3
3	56
4	60
5	51
6	22
7	10
8	20
9	8
10	5
11	5
12	2
13	1
14	1
15	1

---

## Distribution of Naps along specific days

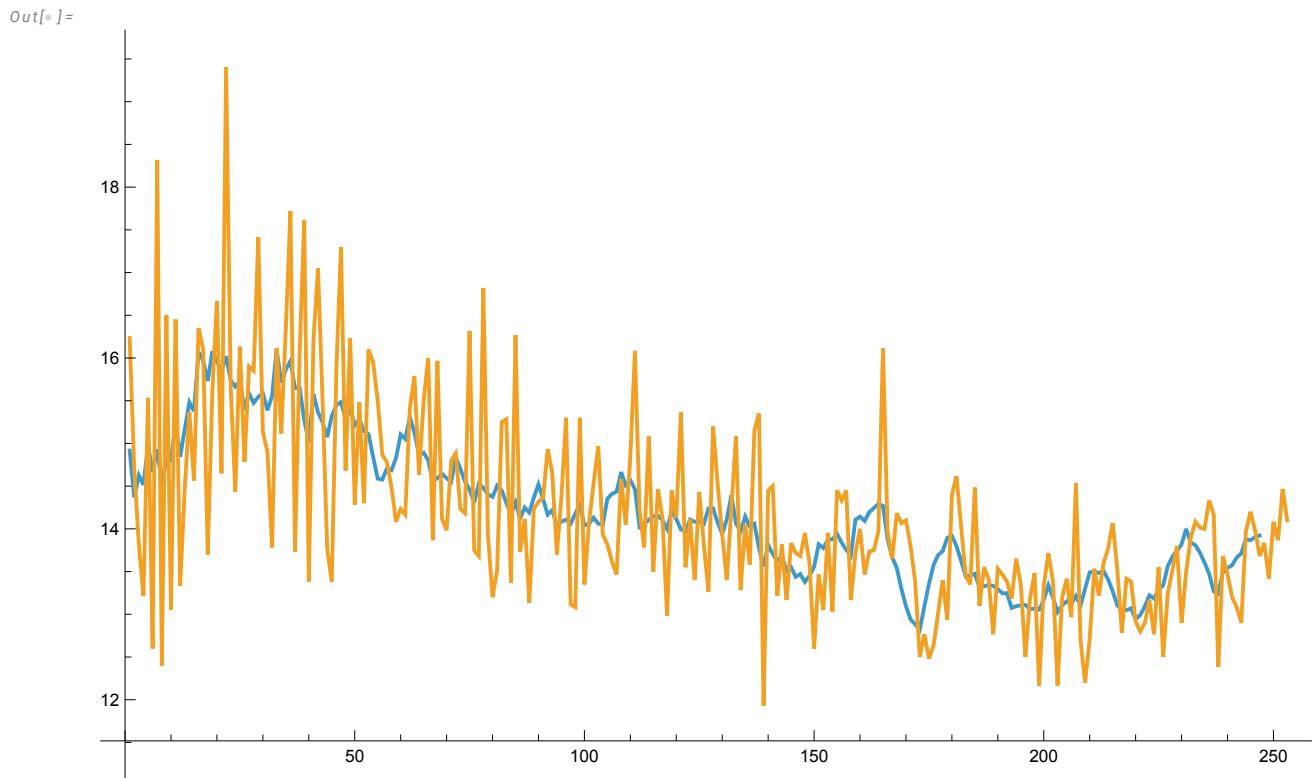
```
In[5]:= barChartSleep[date_String] :=
BarChart[f[date][[All, "Duration2"]] // QuantityMagnitude // Normal // Reverse,
ChartLabels -> (f[date][[All, "TimeRange"]] // Normal // Reverse)]
```



## Moving average total hours per day

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

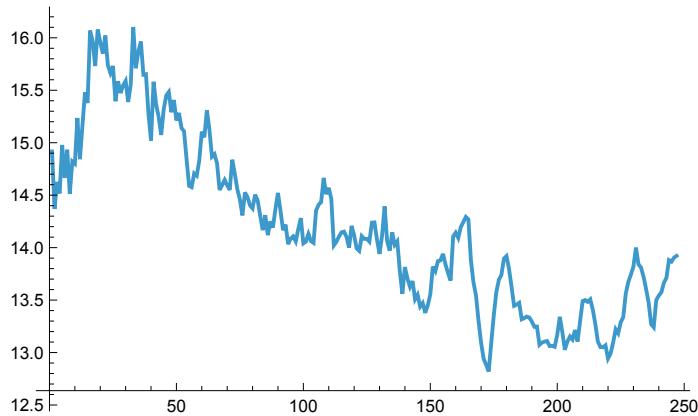
In[8]:= `ListLinePlot[{MovingAverage[sleepData2, 7], sleepData2}]`



In[9]:= (\*Moving average only\*)

```
In[6]:= ListLinePlot[MovingAverage[sleepData2, 7]]
```

```
Out[6]=
```



```
In[7]:= (*Fitting the data with ListFitPlot[]*)
```

```
In[8]:= ListFitPlot[sleepData2[[1 ;; -2]], PlotRange -> {Automatic, {12, 20}}]
```

```
Out[8]=
```

