```
import os
print(os.getcwd())
print(os.listdir("/content")[:20])
```

```
/content
['.config', 'Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk.zip', 'sample_data']
```

```
import zipfile
import os

zip_path = "/content/Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk.zip"   # ← replace this
extract_path = "/content/rtt_data"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print("Unzipped files:")
os.listdir(extract_path)
```

```
Unzipped files:
['20251130-RTT-November-2025-full-extract.csv']
```

```
import pandas as pd

file_path = "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv"

df = pd.read_csv(file_path)
```

```
df.shape
```

```
(180291, 121)
```

```
df.columns.tolist()
```

```
['Period',
 'Provider Parent Org Code',
 'Provider Parent Name',
 'Provider Org Code',
 'Provider Org Name',
 'Commissioner Parent Org Code',
 'Commissioner Parent Name',
 'Commissioner Org Code',
 'Commissioner Org Name',
 'RTT Part Type',
 'RTT Part Description',
 'Treatment Function Code',
 'Treatment Function Name',
 'Gt 00 To 01 Weeks SUM 1',
 'Gt 01 To 02 Weeks SUM 1',
 'Gt 02 To 03 Weeks SUM 1',
 'Gt 03 To 04 Weeks SUM 1',
 'Gt 04 To 05 Weeks SUM 1',
 'Gt 05 To 06 Weeks SUM 1',
 'Gt 06 To 07 Weeks SUM 1',
 'Gt 07 To 08 Weeks SUM 1',
 'Gt 08 To 09 Weeks SUM 1',
 'Gt 09 To 10 Weeks SUM 1',
 'Gt 10 To 11 Weeks SUM 1',
 'Gt 11 To 12 Weeks SUM 1',
 'Gt 12 To 13 Weeks SUM 1',
 'Gt 13 To 14 Weeks SUM 1',
 'Gt 14 To 15 Weeks SUM 1',
 'Gt 15 To 16 Weeks SUM 1',
 'Gt 16 To 17 Weeks SUM 1',
 'Gt 17 To 18 Weeks SUM 1',
 'Gt 18 To 19 Weeks SUM 1',
 'Gt 19 To 20 Weeks SUM 1',
 'Gt 20 To 21 Weeks SUM 1',
 'Gt 21 To 22 Weeks SUM 1',
 'Gt 22 To 23 Weeks SUM 1',
 'Gt 23 To 24 Weeks SUM 1',
 'Gt 24 To 25 Weeks SUM 1',
 'Gt 25 To 26 Weeks SUM 1',
 'Gt 26 To 27 Weeks SUM 1',
 'Gt 27 To 28 Weeks SUM 1',
```

```
 'Gt 28 To 29 Weeks SUM 1',
 'Gt 29 To 30 Weeks SUM 1',
 'Gt 30 To 31 Weeks SUM 1',
 'Gt 31 To 32 Weeks SUM 1',
 'Gt 32 To 33 Weeks SUM 1',
 'Gt 33 To 34 Weeks SUM 1',
 'Gt 34 To 35 Weeks SUM 1',
 'Gt 35 To 36 Weeks SUM 1',
 'Gt 36 To 37 Weeks SUM 1',
 'Gt 37 To 38 Weeks SUM 1',
 'Gt 38 To 39 Weeks SUM 1',
 'Gt 39 To 40 Weeks SUM 1',
 'Gt 40 To 41 Weeks SUM 1',
 'Gt 41 To 42 Weeks SUM 1',
 'Gt 42 To 43 Weeks SUM 1',
 'Gt 43 To 44 Weeks SUM 1',
 'Gt 44 To 45 Weeks SUM 1',
```

```
df.head(10)
```

	Period	Provider Parent Org Code	Provider Parent Name	Provider Org Code	Provider Org Name	Commissioner Parent Org Code	Commissioner Parent Name	Commissioner Org Code	Commissioner Org Name	P T
0	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	NaN	NaN	13Q	NATIONAL COMMISSIONING HUB 1	Pa
1	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	NaN	NaN	13Q	NATIONAL COMMISSIONING HUB 1	Pa
2	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	NaN	NaN	Y62	NaN	Pa
3	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	NaN	NaN	Y62	NaN	Pa
4	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	NaN	NaN	Y62	NaN	Pa
5	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	00Q	NHS BLACKBURN WITH DARWEN (SUB ICB LOCATION)	Part
6	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	00Q	NHS BLACKBURN WITH DARWEN (SUB ICB LOCATION)	Part
7	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	00Q	NHS BLACKBURN WITH DARWEN (SUB ICB LOCATION)	Part
8	RTT-November-2025	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	A4M8P	BUCKSHAW HOSPITAL	QE1	NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CA...	00Q	NHS BLACKBURN WITH DARWEN (SUB ICB LOCATION)	Part

```
# Identify all waiting time columns (they start with 'Gt ')
waiting_cols = [c for c in df.columns if c.startswith("Gt ")]

len(waiting_cols), waiting_cols[:5]
```

```
(105,
 ['Gt 00 To 01 Weeks SUM 1',
  'Gt 01 To 02 Weeks SUM 1',
  'Gt 02 To 03 Weeks SUM 1',
```

	'Gt 03 To 04 Weeks SUM 1'													
	'Gt 04 To 05 Weeks SUM 1'													
9	RTT November-2025	QE1	LANCASHIRE AND SOUTH CUMBRIA	A4M8P	BUCKSHAW HOSPITAL	QE1	LANCASHIRE AND SOUTH CUMBRIA	00Q	NHS BLACKBURN WITH DARWEN (SUB ICB	Part_

```
national_waiting = df[waiting_cols].sum()
national_waiting.head()
```

	0
Gt 00 To 01 Weeks SUM 1	1107876.0
Gt 01 To 02 Weeks SUM 1	1020418.0
Gt 02 To 03 Weeks SUM 1	951394.0
Gt 03 To 04 Weeks SUM 1	875286.0
Gt 04 To 05 Weeks SUM 1	781098.0

dtype: float64

```
waiting_df = national_waiting.reset_index()
waiting_df.columns = ["Waiting_Weeks_Band", "Patients"]
waiting_df.head()
```

	Waiting_Weeks_Band	Patients
0	Gt 00 To 01 Weeks SUM 1	1107876.0
1	Gt 01 To 02 Weeks SUM 1	1020418.0
2	Gt 02 To 03 Weeks SUM 1	951394.0
3	Gt 03 To 04 Weeks SUM 1	875286.0
4	Gt 04 To 05 Weeks SUM 1	781098.0

```
import matplotlib.pyplot as plt

plt.figure(figsize=(14,6))
plt.plot(waiting_df["Weeks"], waiting_df["Patients"], marker="o")

plt.title("National NHS RTT Waiting List Distribution (November 2025)")
plt.xlabel("Weeks Waiting")
plt.ylabel("Number of Patients")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Weeks'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
```

2 frames

```
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'Weeks'

<Figure size 1400x600 with 0 Axes>
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(14,6))
plt.plot(waiting_df["Weeks"], waiting_df["Patients"], marker="o")

plt.title("National NHS RTT Waiting List Distribution (November 2025)")
plt.xlabel("Weeks Waiting")
plt.ylabel("Number of Patients")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Weeks'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
 ─────────────────────────────────── 2 frames ───────────────────────────────────
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'Weeks'

<Figure size 1400x600 with 0 Axes>
```

```
waiting_df.columns
```

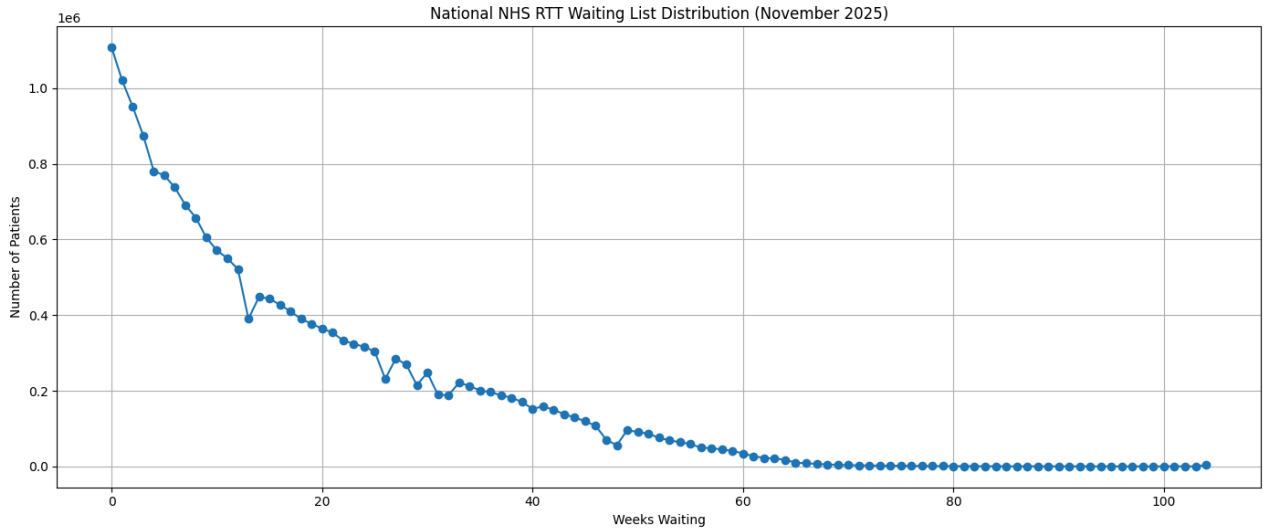
```
Index(['Waiting_Weeks_Band', 'Patients'], dtype='object')
```

```
# Extract numeric week value from column names
waiting_df["Weeks"] = (
    waiting_df["Waiting_Weeks_Band"]
    .str.extract(r'(\d+)')
    .astype(int)
)

# Sort properly
waiting_df = waiting_df.sort_values("Weeks")

waiting_df.head()
```

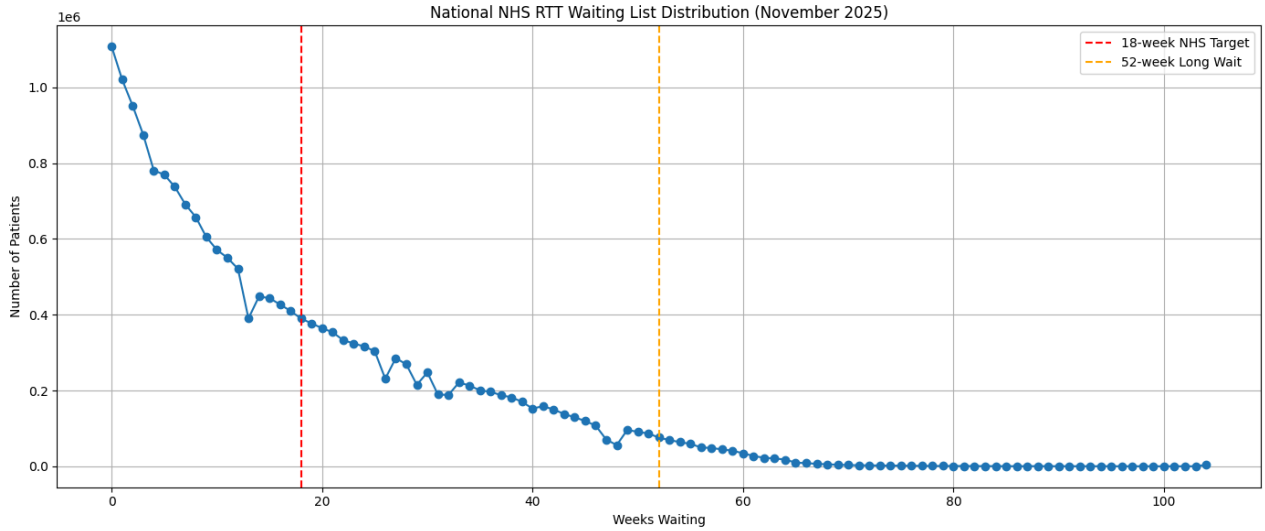
	Waiting_Weeks_Band	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	1107876.0	0
1	Gt 01 To 02 Weeks SUM 1	1020418.0	1
2	Gt 02 To 03 Weeks SUM 1	951394.0	2
3	Gt 03 To 04 Weeks SUM 1	875286.0	3
4	Gt 04 To 05 Weeks SUM 1	781098.0	4

```
import matplotlib.pyplot as plt

plt.figure(figsize=(14,6))
plt.plot(waiting_df["Weeks"], waiting_df["Patients"], marker="o")

plt.title("National NHS RTT Waiting List Distribution (November 2025)")
plt.xlabel("Weeks Waiting")
plt.ylabel("Number of Patients")
plt.grid(True)
```

```
plt.tight_layout()
plt.show()
```




```
plt.figure(figsize=(14,6))
plt.plot(waiting_df["Weeks"], waiting_df["Patients"], marker="o")

# NHS thresholds
plt.axvline(x=18, color='red', linestyle='--', label='18-week NHS Target')
plt.axvline(x=52, color='orange', linestyle='--', label='52-week Long Wait')

plt.title("National NHS RTT Waiting List Distribution (November 2025)")
plt.xlabel("Weeks Waiting")
plt.ylabel("Number of Patients")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```




```
total_waiting = waiting_df["Patients"].sum()
total_waiting
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-3594161569.py in <cell line: 0>()
----> 1 total_waiting = waiting_df["Patients"].sum()
      2 total_waiting

NameError: name 'waiting_df' is not defined
```

```
import pandas as pd

file_path = "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv"

df = pd.read_csv(file_path)

df.shape
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
/tmp/ipython-input-3613232529.py in <cell line: 0>()
      3 file_path = "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv"
      4
----> 5 df = pd.read_csv(file_path)
      6
      7 df.shape

4 frames
/usr/local/lib/python3.12/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression,
memory_map, is_text, errors, storage_options)
    871         if ioargs.encoding and "b" not in ioargs.mode:
    872             # Encoding
--> 873             handle = open(
    874                 handle,
    875                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: '/content/rtt_data/20251130-RTT-November-2025-full-extract.csv'
```

```
import pandas as pd

file_path = "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv"

df = pd.read_csv(file_path)

df.shape
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
/tmp/ipython-input-3613232529.py in <cell line: 0>()
      3 file_path = "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv"
      4
----> 5 df = pd.read_csv(file_path)
      6
      7 df.shape

4 frames
/usr/local/lib/python3.12/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression,
memory_map, is_text, errors, storage_options)
    871         if ioargs.encoding and "b" not in ioargs.mode:
    872             # Encoding
--> 873             handle = open(
    874                 handle,
    875                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: '/content/rtt_data/20251130-RTT-November-2025-full-extract.csv'
```

```
import os

print(os.listdir("/content"))
```

```
['.config', 'Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk.zip', 'sample_data']
```

```
import zipfile
import os

zip_path = "/content/Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk.zip"
extract_path = "/content/rtt_data"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print("Unzipped files:")
print(os.listdir(extract_path))
```

```
Unzipped files:
['20251130-RTT-November-2025-full-extract.csv']
```

```
file_path = "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv"

df = pd.read_csv(file_path)

df.shape
```

```
(180291, 121)
```

```
# 1) See the key columns
df.columns[:15]
```

```
Index(['Period', 'Provider Parent Org Code', 'Provider Parent Name',
       'Provider Org Code', 'Provider Org Name',
       'Commissioner Parent Org Code', 'Commissioner Parent Name',
       'Commissioner Org Code', 'Commissioner Org Name', 'RTT Part Type',
       'RTT Part Description', 'Treatment Function Code',
       'Treatment Function Name', 'Gt 00 To 01 Weeks SUM 1',
       'Gt 01 To 02 Weeks SUM 1'],
      dtype='object')
```

```
# 2) Get all waiting-bin columns (these start with "Gt")
wait_cols = [c for c in df.columns if c.startswith("Gt ") and "Weeks" in c]

len(wait_cols), wait_cols[:5], wait_cols[-5:]
```

```
(105,
 ['Gt 00 To 01 Weeks SUM 1',
  'Gt 01 To 02 Weeks SUM 1',
  'Gt 02 To 03 Weeks SUM 1',
  'Gt 03 To 04 Weeks SUM 1',
  'Gt 04 To 05 Weeks SUM 1'],
 ['Gt 100 To 101 Weeks SUM 1',
  'Gt 101 To 102 Weeks SUM 1',
  'Gt 102 To 103 Weeks SUM 1',
  'Gt 103 To 104 Weeks SUM 1',
  'Gt 104 Weeks SUM 1'])
```

```
# 3) Sum each waiting bin across the whole dataset
national_bins = df[wait_cols].sum(numeric_only=True)

national_bins.head(), national_bins.tail()
```

```
(Gt 00 To 01 Weeks SUM 1    1107876.0
 Gt 01 To 02 Weeks SUM 1    1020418.0
 Gt 02 To 03 Weeks SUM 1     951394.0
 Gt 03 To 04 Weeks SUM 1     875286.0
 Gt 04 To 05 Weeks SUM 1     781098.0
 dtype: float64,
 Gt 100 To 101 Weeks SUM 1     120.0
 Gt 101 To 102 Weeks SUM 1     150.0
 Gt 102 To 103 Weeks SUM 1     130.0
 Gt 103 To 104 Weeks SUM 1     150.0
 Gt 104 Weeks SUM 1           3890.0
 dtype: float64)
```

```
import pandas as pd

waiting_df = pd.DataFrame({
    "Bin": national_bins.index,
    "Patients": national_bins.values
})

# Extract the starting week number from "Gt 00 To 01 Weeks..."
waiting_df["Weeks"] = waiting_df["Bin"].str.extract(r"Gt\s+(\d+)\s+To").astype(int)

waiting_df = waiting_df.sort_values("Weeks").reset_index(drop=True)

waiting_df.head()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-3801148595.py in <cell line: 0>()
      7
      8 # Extract the starting week number from "Gt 00 To 01 Weeks..."
----> 9 waiting_df["Weeks"] = waiting_df["Bin"].str.extract(r"Gt\s+(\d+)\s+To").astype(int)
     10
     11 waiting_df = waiting_df.sort_values("Weeks").reset_index(drop=True)

                              6 frames
/usr/local/lib/python3.12/dist-packages/pandas/core/dtypes/astype.py in _astype_nansafe(arr, dtype, copy, skipna)
    131     if copy or arr.dtype == object or dtype == object:
    132         # Explicit copy, or required since NumPy can't view from / to object.
--> 133         return arr.astype(dtype, copy=True)
    134
    135     return arr.astype(dtype, copy=copy)

ValueError: cannot convert float NaN to integer
```

```
import pandas as pd

waiting_df = pd.DataFrame({
    "Bin": national_bins.index,
    "Patients": national_bins.values
})

# Extract starting week number (works for both "Gt 00 To 01" AND "Gt 104 Weeks")
waiting_df["Weeks"] = (
    waiting_df["Bin"]
    .str.extract(r"Gt\s+(\d+)")
    .astype(float)
)

# Drop rows where extraction failed (if any)
waiting_df = waiting_df.dropna(subset=["Weeks"])

# Convert to integer
waiting_df["Weeks"] = waiting_df["Weeks"].astype(int)

# Sort properly
waiting_df = waiting_df.sort_values("Weeks").reset_index(drop=True)

waiting_df.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	1107876.0	0
1	Gt 01 To 02 Weeks SUM 1	1020418.0	1
2	Gt 02 To 03 Weeks SUM 1	951394.0	2
3	Gt 03 To 04 Weeks SUM 1	875286.0	3
4	Gt 04 To 05 Weeks SUM 1	781098.0	4

```
total_waiting = waiting_df["Patients"].sum()
total_waiting
```

```
np.float64(19715956.0)
```

```
over_18_weeks = waiting_df[waiting_df["Weeks"] > 18]["Patients"].sum()
over_18_weeks
```

```
np.float64(7365914.0)
```

```
over_52_weeks = waiting_df[waiting_df["Weeks"] > 52]["Patients"].sum()
over_52_weeks
```

```
np.float64(563894.0)
```

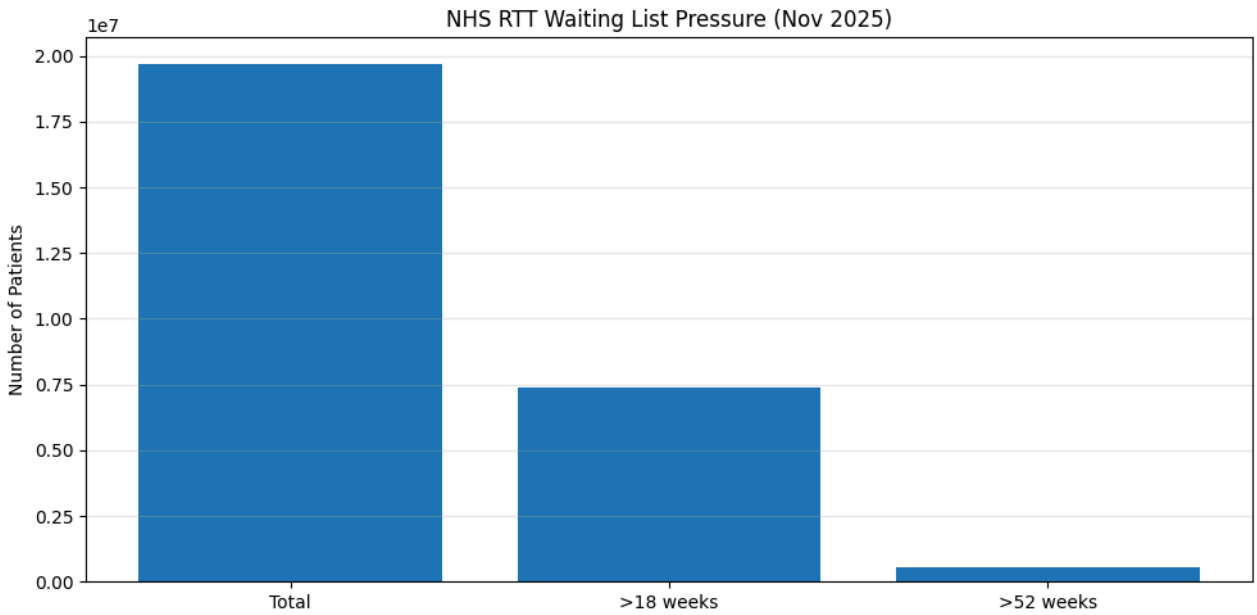
```
pct_over_18 = (over_18_weeks / total_waiting) * 100
pct_over_52 = (over_52_weeks / total_waiting) * 100

pct_over_18, pct_over_52
```

```
(np.float64(37.360166557482685), np.float64(2.860089564005925))
```

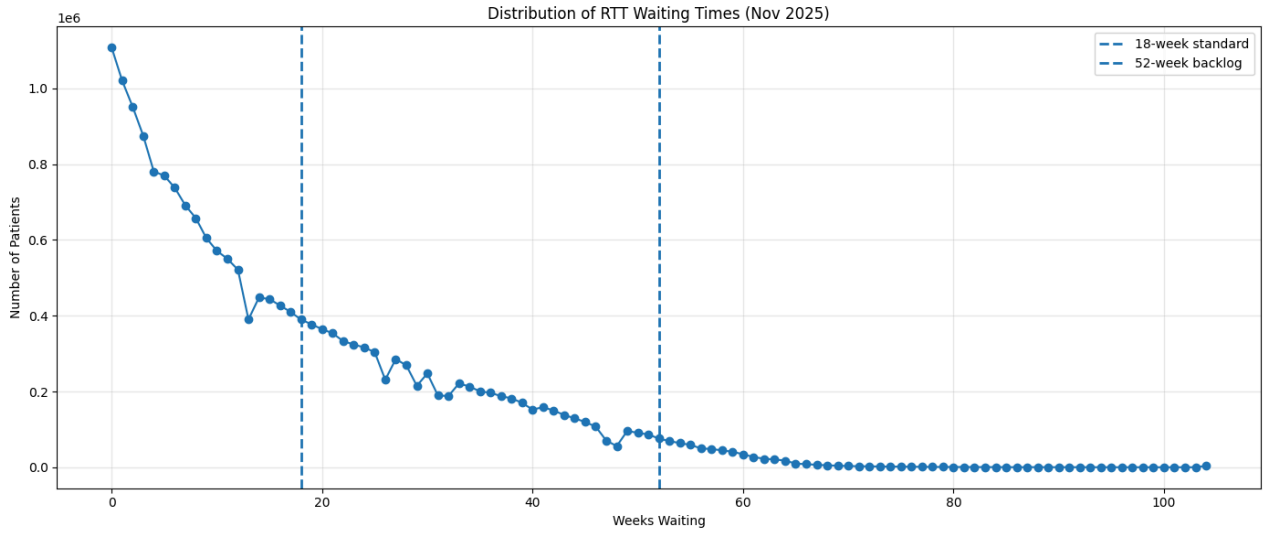
```
import pandas as pd

summary = pd.DataFrame({
    "Metric": [
        "Total RTT waiting list",
        "Patients waiting > 18 weeks",
        "Patients waiting > 52 weeks"
    ],
    "Patients": [total_waiting, over_18_weeks, over_52_weeks],
    "Percent of total": [100, pct_over_18, pct_over_52]
})

summary
```

	Metric	Patients	Percent of total
0	Total RTT waiting list	19715956.0	100.000000
1	Patients waiting > 18 weeks	7365914.0	37.360167
2	Patients waiting > 52 weeks	563894.0	2.860090

```
import matplotlib.pyplot as plt

labels = ["Total", ">18 weeks", ">52 weeks"]
values = [total_waiting, over_18_weeks, over_52_weeks]

plt.figure(figsize=(10,5))
plt.bar(labels, values)
plt.title("NHS RTT Waiting List Pressure (Nov 2025)")
plt.ylabel("Number of Patients")
plt.grid(axis="y", alpha=0.3)
plt.tight_layout()
plt.show()
```




```
plt.figure(figsize=(14,6))
plt.plot(waiting_df["Weeks"], waiting_df["Patients"], marker="o")

plt.axvline(18, linestyle="--", linewidth=2, label="18-week standard")
plt.axvline(52, linestyle="--", linewidth=2, label="52-week backlog")

plt.title("Distribution of RTT Waiting Times (Nov 2025)")
plt.xlabel("Weeks Waiting")
plt.ylabel("Number of Patients")
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```




```
import os
print(os.listdir("/content"))
```

```
['.config', 'Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk.zip', 'Full-CSV-data-file-Dec25-ZIP-4M-6jPlxd.zip', 'rtt_dec_2025', 'rtt
```

```
import pandas as pd

file_path = "/content/Full-CSV-data-file-Dec25-ZIP-4M-6jPlxd.zip"  # adjust if inside folder
dec_df = pd.read_csv(file_path)

dec_df.shape
```

```
(179805, 121)
```

```
dec_df["RTT Part Description"].value_counts()
```

RTT Part Description	count
Incomplete Pathways	65174
New RTT Periods - All Patients	35391
Completed Pathways For Non-Admitted Patients	31006
Incomplete Pathways with DTA	30400
Completed Pathways For Admitted Patients	17834

dtype: int64

```
dec_incomplete = dec_df[
    dec_df["RTT Part Description"] == "Incomplete pathways"
]

dec_incomplete.shape
```

```
(0, 121)
```

```
wait_cols = [c for c in dec_incomplete.columns if c.startswith("Gt ")]

len(wait_cols)
```

```
105
```

```
national_bins_dec = dec_incomplete[wait_cols].sum()

national_bins_dec.head()
national_bins_dec.sum()
```

```
np.float64(0.0)
```

```
dec_incomplete.shape
```

```
(0, 121)
```

```
dec_df["RTT Part Description"].value_counts().head(20)
```

RTT Part Description	count
Incomplete Pathways	65174
New RTT Periods - All Patients	35391
Completed Pathways For Non-Admitted Patients	31006
Incomplete Pathways with DTA	30400
Completed Pathways For Admitted Patients	17834

dtype: int64

```
dec_incomplete = dec_df[dec_df["RTT Part Description"] == "Incomplete Pathways"].copy()
dec_incomplete.shape
```

```
(65174, 121)
```

```
wait_cols = [c for c in dec_incomplete.columns if c.startswith("Gt ")]
len(wait_cols), wait_cols[:5]
```

```
(105,
 ['Gt 00 To 01 Weeks SUM 1',
  'Gt 01 To 02 Weeks SUM 1',
  'Gt 02 To 03 Weeks SUM 1',
  'Gt 03 To 04 Weeks SUM 1',
  'Gt 04 To 05 Weeks SUM 1'])
```

```
for c in wait_cols:
    dec_incomplete[c] = (
        dec_incomplete[c]
        .astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    dec_incomplete[c] = pd.to_numeric(dec_incomplete[c], errors="coerce")
```

```
# Quick proof the conversion worked
dec_incomplete[wait_cols].dtypes.value_counts(), dec_incomplete[wait_cols].sum().sum()
```

```
(float64    105
 Name: count, dtype: int64,
 np.float64(14463760.0))
```

```
national_bins_dec = dec_incomplete[wait_cols].sum(numeric_only=True)
national_bins_dec.head(), national_bins_dec.sum()
```

```
(Gt 00 To 01 Weeks SUM 1    394184.0
 Gt 01 To 02 Weeks SUM 1    668436.0
 Gt 02 To 03 Weeks SUM 1    675438.0
 Gt 03 To 04 Weeks SUM 1    639198.0
 Gt 04 To 05 Weeks SUM 1    612790.0
 dtype: float64,
 np.float64(14463760.0))
```

```
dec_incomplete = dec_df[dec_df["RTT Part Description"] == "Incomplete Pathways"].copy()
```

```
import numpy as np
import pandas as pd

waiting_df_dec = pd.DataFrame({
    "Bin": national_bins_dec.index.astype(str),
    "Patients": national_bins_dec.values
})

# Extract start week robustly
weeks = waiting_df_dec["Bin"].str.extract(r"Gt\s*(\d+)\s*To", expand=False)
waiting_df_dec["Weeks"] = pd.to_numeric(weeks, errors="coerce")

# Drop bins that didn't match (if any)
waiting_df_dec = waiting_df_dec.dropna(subset=["Weeks"]).copy()
waiting_df_dec["Weeks"] = waiting_df_dec["Weeks"].astype(int)

waiting_df_dec = waiting_df_dec.sort_values("Weeks").reset_index(drop=True)
waiting_df_dec.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	394184.0	0
1	Gt 01 To 02 Weeks SUM 1	668436.0	1
2	Gt 02 To 03 Weeks SUM 1	675438.0	2
3	Gt 03 To 04 Weeks SUM 1	639198.0	3
4	Gt 04 To 05 Weeks SUM 1	612790.0	4

```
waiting_df_dec[["Bin","Weeks"]].head(10)
```

	Bin	Weeks
0	Gt 00 To 01 Weeks SUM 1	0
1	Gt 01 To 02 Weeks SUM 1	1
2	Gt 02 To 03 Weeks SUM 1	2
3	Gt 03 To 04 Weeks SUM 1	3
4	Gt 04 To 05 Weeks SUM 1	4
5	Gt 05 To 06 Weeks SUM 1	5
6	Gt 06 To 07 Weeks SUM 1	6
7	Gt 07 To 08 Weeks SUM 1	7
8	Gt 08 To 09 Weeks SUM 1	8
9	Gt 09 To 10 Weeks SUM 1	9

```
total_dec = waiting_df_dec["Patients"].sum()
over_18_dec = waiting_df_dec.loc[waiting_df_dec["Weeks"] > 18, "Patients"].sum()
over_52_dec = waiting_df_dec.loc[waiting_df_dec["Weeks"] > 52, "Patients"].sum()

pct_18_dec = (over_18_dec / total_dec) * 100 if total_dec else np.nan
pct_52_dec = (over_52_dec / total_dec) * 100 if total_dec else np.nan

total_dec, over_18_dec, over_52_dec, pct_18_dec, pct_52_dec
```

```
(np.float64(14462538.0),
 np.float64(5354058.0),
 np.float64(266086.0),
 np.float64(37.0201827646019),
 np.float64(1.8398292194634163))
```

```
# November values (replace with your saved nov results if needed)
total_nov = 19715956.0
over_18_nov = 7365914.0
over_52_nov = 563894.0
pct_18_nov = 37.360166557482685
pct_52_nov = 2.860089564005925

comparison = {
    "Metric": ["Total Waiting", "% >18 Weeks", "% >52 Weeks"],
    "November": [total_nov, pct_18_nov, pct_52_nov],
    "December": [total_dec, pct_18_dec, pct_52_dec]
```

```
}

pd.DataFrame(comparison)
```

	Metric	November	December
0	Total Waiting	1.971596e+07	1.446254e+07
1	% >18 Weeks	3.736017e+01	3.702018e+01
2	% >52 Weeks	2.860090e+00	1.839829e+00

```
# Check if there are obvious "national total" rows you should use instead of summing everything
for col in ["Provider Org Code","Provider Org Name","Commissioner Org Code","Commissioner Org Name","Provider Parent Name",
    if col in dec_df.columns:
        print(col, dec_df[col].nunique(), dec_df[col].astype(str).head(3).tolist())
```

```
Provider Org Code 507 ['A4M8P', 'A4M8P', 'A4M8P']
Provider Org Name 506 ['BUCKSHAW HOSPITAL', 'BUCKSHAW HOSPITAL', 'BUCKSHAW HOSPITAL']
Commissioner Org Code 129 ['13Q', '13Q', 'Y62']
Commissioner Org Name 121 ['NATIONAL COMMISSIONING HUB 1', 'NATIONAL COMMISSIONING HUB 1', 'nan']
Provider Parent Name 42 ['NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRATED CARE BOARD', 'NHS LANCASHIRE AND SOUTH CUMBRIA INTEGRA
Commissioner Parent Name 42 ['nan', 'nan', 'nan']
```

```
# Look for any rows that look like national aggregates (keywords)
keywords = ["England", "National", "All", "TOTAL"]
for col in ["Provider Org Name","Commissioner Org Name","Provider Parent Name","Commissioner Parent Name"]:
    if col in dec_df.columns:
        matches = dec_df[col].astype(str).str.contains("|".join(keywords), case=False, na=False).sum()
        print(col, "matches:", matches)
```

```
Provider Org Name matches: 7146
Commissioner Org Name matches: 6956
Provider Parent Name matches: 673
Commissioner Parent Name matches: 1599
```

```
# Step 1: Filter to Incomplete Pathways
dec_incomplete = dec_df[dec_df["RTT Part Description"] == "Incomplete Pathways"].copy()

# Step 2: Aggregate at Provider Parent level first
provider_grouped = dec_incomplete.groupby(
    ["Provider Parent Org Code"]
)[wait_cols].sum().reset_index()

# Step 3: Now sum across provider parents (true national total)
national_bins_dec = provider_grouped[wait_cols].sum()

national_bins_dec.sum()
```

```
np.float64(14463760.0)
```

```
waiting_df_dec = pd.DataFrame({
    "Bin": national_bins_dec.index,
    "Patients": national_bins_dec.values
})
waiting_df_dec.head()
```

	Bin	Patients
0	Gt 00 To 01 Weeks SUM 1	394184.0
1	Gt 01 To 02 Weeks SUM 1	668436.0
2	Gt 02 To 03 Weeks SUM 1	675438.0
3	Gt 03 To 04 Weeks SUM 1	639198.0
4	Gt 04 To 05 Weeks SUM 1	612790.0

```
waiting_df_dec["Weeks"] = (
    waiting_df_dec["Bin"]
    .str.extract(r"Gt\s+(\d+)\s+To", expand=False)
)

waiting_df_dec["Weeks"] = pd.to_numeric(waiting_df_dec["Weeks"], errors="coerce")
waiting_df_dec = waiting_df_dec.dropna(subset=["Weeks"])
```

```
waiting_df_dec["Weeks"] = waiting_df_dec["Weeks"].astype(int)

waiting_df_dec = waiting_df_dec.sort_values("Weeks").reset_index(drop=True)
waiting_df_dec.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	394184.0	0
1	Gt 01 To 02 Weeks SUM 1	668436.0	1
2	Gt 02 To 03 Weeks SUM 1	675438.0	2
3	Gt 03 To 04 Weeks SUM 1	639198.0	3
4	Gt 04 To 05 Weeks SUM 1	612790.0	4

```
total_dec = waiting_df_dec["Patients"].sum()
over_18_dec = waiting_df_dec.loc[waiting_df_dec["Weeks"] > 18, "Patients"].sum()
over_52_dec = waiting_df_dec.loc[waiting_df_dec["Weeks"] > 52, "Patients"].sum()

pct_18_dec = (over_18_dec / total_dec) * 100
pct_52_dec = (over_52_dec / total_dec) * 100

total_dec, over_18_dec, over_52_dec, pct_18_dec, pct_52_dec
```

```
(np.float64(14462538.0),
 np.float64(5354058.0),
 np.float64(266086.0),
 np.float64(37.0201827646019),
 np.float64(1.8398292194634163))
```

```
dec_summary = {
    "Month": "December 2025",
    "Total_Waiting": total_dec,
    "Pct_Over_18": pct_18_dec,
    "Pct_Over_52": pct_52_dec
}
dec_summary
```

```
{'Month': 'December 2025',
 'Total_Waiting': np.float64(14462538.0),
 'Pct_Over_18': np.float64(37.0201827646019),
 'Pct_Over_52': np.float64(1.8398292194634163)}
```

```
import zipfile
import os

zip_path = "/content/Full-CSV-data-file-Oct25-ZIP-4M-SrRW6y.zip"
extract_path = "/content/rtt_oct_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20251031-RTT-October-2025-full-extract.csv']
```

```
file_path = "/content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv"
oct_df = pd.read_csv(file_path)

oct_df.shape
```

```
(185742, 121)
```

```
oct_incomplete = oct_df[oct_df["RTT Part Description"] == "Incomplete Pathways"].copy()

oct_incomplete.shape
```

```
(65332, 121)
```

```
wait_cols = [c for c in oct_incomplete.columns if c.startswith("Gt ")]
len(wait_cols)
```

```
105
```

```
for c in wait_cols:
    oct_incomplete[c] = (
        oct_incomplete[c]
        .astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    oct_incomplete[c] = pd.to_numeric(oct_incomplete[c], errors="coerce")
```

```
oct_incomplete[wait_cols].sum().sum()
```

```
np.float64(14674152.0)
```

```
national_bins_oct = oct_incomplete[wait_cols].sum(numeric_only=True)

national_bins_oct.sum()
```

```
np.float64(14674152.0)
```

```
waiting_df_oct = pd.DataFrame({
    "Bin": national_bins_oct.index,
    "Patients": national_bins_oct.values
})

# Extract week number safely
waiting_df_oct["Weeks"] = (
    waiting_df_oct["Bin"]
    .str.extract(r"Gt\s+(\d+)\s+To")
    .fillna(0)
    .astype(int)
)

waiting_df_oct = waiting_df_oct.sort_values("Weeks").reset_index(drop=True)

waiting_df_oct.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	776986.0	0
1	Gt 104 Weeks SUM 1	1178.0	0
2	Gt 01 To 02 Weeks SUM 1	722208.0	1
3	Gt 02 To 03 Weeks SUM 1	695452.0	2
4	Gt 03 To 04 Weeks SUM 1	654692.0	3

```
total_oct = waiting_df_oct["Patients"].sum()
over_18_oct = waiting_df_oct[waiting_df_oct["Weeks"] > 18]["Patients"].sum()
over_52_oct = waiting_df_oct[waiting_df_oct["Weeks"] > 52]["Patients"].sum()

pct_18_oct = (over_18_oct / total_oct) * 100
pct_52_oct = (over_52_oct / total_oct) * 100

total_oct, over_18_oct, over_52_oct, pct_18_oct, pct_52_oct
```

```
(np.float64(14674152.0),
 np.float64(5318176.0),
 np.float64(303622.0),
 np.float64(36.24179441510487),
 np.float64(2.0690940096572534))
```

```
import pandas as pd

comparison_df = pd.DataFrame({
    "Metric": ["Total Waiting", "% >18 Weeks", "% >52 Weeks"],
    "October": [total_oct, pct_18_oct, pct_52_oct],
    "November": [total_waiting, pct_over_18, pct_over_52],
    "December": [total_dec, pct_18_dec, pct_52_dec]
})
```

```
comparison_df
```

	Metric	October	November	December
0	Total Waiting	1.467415e+07	1.971596e+07	1.446254e+07
1	% >18 Weeks	3.624179e+01	3.736017e+01	3.702018e+01
2	% >52 Weeks	2.069094e+00	2.860090e+00	1.839829e+00

```
import zipfile
import os

zip_path = "/content/Full-CSV-data-file-Sep25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_sep_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20250930-RTT-September-2025-full-extract-revised.csv']
```

```
import pandas as pd
```

```
file_path = "/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv"

sep_df = pd.read_csv(file_path)

sep_df.shape
```

```
(184484, 121)
```

```
sep_incomplete = sep_df[sep_df["RTT Part Description"] == "Incomplete Pathways"].copy()

sep_incomplete.shape
```

```
(64994, 121)
```

```
wait_cols = [c for c in sep_incomplete.columns if c.startswith("Gt ")]

len(wait_cols)
```

```
105
```

```
for c in wait_cols:
    sep_incomplete[c] = (
        sep_incomplete[c]
        .astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    sep_incomplete[c] = pd.to_numeric(sep_incomplete[c], errors="coerce")
```

```
sep_incomplete[wait_cols].sum().sum()
```

```
np.float64(14659148.0)
```

```
national_bins_sep = sep_incomplete[wait_cols].sum(numeric_only=True)

national_bins_sep.sum()
```

```
np.float64(14659148.0)
```

```
waiting_df_sep = pd.DataFrame({
    "Bin": national_bins_sep.index,
    "Patients": national_bins_sep.values
```

```
})
```

```
waiting_df_sep["Weeks"] = (
    waiting_df_sep["Bin"]
    .str.extract(r"Gt\s+(\d+)\s+To")[0]
)

waiting_df_sep["Weeks"] = pd.to_numeric(waiting_df_sep["Weeks"], errors="coerce")

waiting_df_sep = waiting_df_sep.dropna(subset=["Weeks"])

waiting_df_sep["Weeks"] = waiting_df_sep["Weeks"].astype(int)

waiting_df_sep = waiting_df_sep.sort_values("Weeks").reset_index(drop=True)

waiting_df_sep.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	824192.0	0
1	Gt 01 To 02 Weeks SUM 1	705830.0	1
2	Gt 02 To 03 Weeks SUM 1	676022.0	2
3	Gt 03 To 04 Weeks SUM 1	633878.0	3
4	Gt 04 To 05 Weeks SUM 1	571454.0	4

```
total_sep = waiting_df_sep["Patients"].sum()
over_18_sep = waiting_df_sep[waiting_df_sep["Weeks"] > 18]["Patients"].sum()
over_52_sep = waiting_df_sep[waiting_df_sep["Weeks"] > 52]["Patients"].sum()

pct_18_sep = (over_18_sep / total_sep) * 100
pct_52_sep = (over_52_sep / total_sep) * 100

total_sep, over_18_sep, over_52_sep, pct_18_sep, pct_52_sep
```

```
(np.float64(14658134.0),
 np.float64(5351570.0),
 np.float64(315862.0),
 np.float64(36.50921733966956),
 np.float64(2.154858183176658))
```

```
import zipfile
import os

zip_path = "/content/Full-CSV-data-file-Aug25-ZIP-4M-revised-2.zip"  # replace exact name
extract_path = "/content/rtt_aug_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20250831-RTT-August-2025-full-extract-revised.csv']
```

```
import pandas as pd

file_path = "/content/rtt_aug_2025/20250831-RTT-August-2025-full-extract-revised.csv"

aug_df = pd.read_csv(file_path)

aug_df.shape
```

```
(179028, 121)
```

```
aug_incomplete = aug_df[aug_df["RTT Part Description"] == "Incomplete Pathways"].copy()

aug_incomplete.shape
```

```
(64465, 121)
```

```
wait_cols = [c for c in aug_incomplete.columns if c.startswith("Gt ")]

len(wait_cols)
```

```
105
```

```
for c in wait_cols:
    aug_incomplete[c] = (
        aug_incomplete[c]
        .astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    aug_incomplete[c] = pd.to_numeric(aug_incomplete[c], errors="coerce")
```

```
aug_incomplete[wait_cols].dtypes.value_counts(),
aug_incomplete[wait_cols].sum().sum()
```

```
np.float64(14691088.0)
```

```
national_bins_aug = aug_incomplete[wait_cols].sum(numeric_only=True)

national_bins_aug.head(), national_bins_aug.sum()
```

```
(Gt 00 To 01 Weeks SUM 1    565574.0
 Gt 01 To 02 Weeks SUM 1    678492.0
 Gt 02 To 03 Weeks SUM 1    660696.0
 Gt 03 To 04 Weeks SUM 1    628688.0
 Gt 04 To 05 Weeks SUM 1    601162.0
 dtype: float64,
 np.float64(14691088.0))
```

```
waiting_df_aug = pd.DataFrame({
    "Bin": national_bins_aug.index,
    "Patients": national_bins_aug.values
})

waiting_df_aug["Weeks"] = (
    waiting_df_aug["Bin"]
    .str.extract(r"Gt\s+(\d+)\s+To", expand=False)
)

waiting_df_aug = waiting_df_aug.dropna(subset=["Weeks"])
waiting_df_aug["Weeks"] = waiting_df_aug["Weeks"].astype(int)

waiting_df_aug = waiting_df_aug.sort_values("Weeks").reset_index(drop=True)

waiting_df_aug.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	565574.0	0
1	Gt 01 To 02 Weeks SUM 1	678492.0	1
2	Gt 02 To 03 Weeks SUM 1	660696.0	2
3	Gt 03 To 04 Weeks SUM 1	628688.0	3
4	Gt 04 To 05 Weeks SUM 1	601162.0	4

```
total_aug = waiting_df_aug["Patients"].sum()

over_18_aug = waiting_df_aug[waiting_df_aug["Weeks"] > 18]["Patients"].sum()
over_52_aug = waiting_df_aug[waiting_df_aug["Weeks"] > 52]["Patients"].sum()

pct_18_aug = (over_18_aug / total_aug) * 100
pct_52_aug = (over_52_aug / total_aug) * 100

total_aug, over_18_aug, over_52_aug, pct_18_aug, pct_52_aug
```

```
(np.float64(14690158.0),
 np.float64(5491322.0),
 np.float64(345268.0),
 np.float64(37.380959415140396),
 np.float64(2.350335510346451))
```

```
print(os.listdir("/content"))
```

```
['.config', 'Full-CSV-data-file-Jul25-ZIP-4M-revised-2.zip', 'sample_data', 'rtt_aug_2025', 'Full-CSV-data-file-Aug25-ZIP-4M
```

```
import zipfile
import os

zip_path = "/content/Full-CSV-data-file-Jul25-ZIP-4M-revised-2.zip"
extract_path = "/content/rtt_jul_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20250731-RTT-July-2025-full-extract-revised.csv']
```

```
import pandas as pd

file_path = "/content/rtt_jul_2025/20250731-RTT-July-2025-full-extract-revised.csv"

jul_df = pd.read_csv(file_path)

jul_df.shape
```

```
(184568, 121)
```

```
jul_incomplete = jul_df[jul_df["RTT Part Description"] == "Incomplete Pathways"].copy()

jul_incomplete.shape
```

```
(64304, 121)
```

```
wait_cols = [c for c in jul_incomplete.columns if c.startswith("Gt ")]

len(wait_cols)
```

```
105
```

```
for c in wait_cols:
    jul_incomplete[c] = (
        jul_incomplete[c]
        .astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    jul_incomplete[c] = pd.to_numeric(jul_incomplete[c], errors="coerce")
```

```
jul_incomplete[wait_cols].sum().sum()
```

```
np.float64(14668572.0)
```

```
national_bins_jul = jul_incomplete[wait_cols].sum(numeric_only=True)

national_bins_jul.sum()
```

```
np.float64(14668572.0)
```

```
waiting_df_jul = pd.DataFrame({
    "Bin": national_bins_jul.index,
    "Patients": national_bins_jul.values
})

waiting_df_jul["Weeks"] = (
    waiting_df_jul["Bin"]
    .str.extract(r"Gt\s+(\d+)\s+To", expand=False)
```

```
)

waiting_df_jul = waiting_df_jul.dropna(subset=["Weeks"])
waiting_df_jul["Weeks"] = waiting_df_jul["Weeks"].astype(int)

waiting_df_jul = waiting_df_jul.sort_values("Weeks").reset_index(drop=True)

waiting_df_jul.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	784394.0	0
1	Gt 01 To 02 Weeks SUM 1	728666.0	1
2	Gt 02 To 03 Weeks SUM 1	681346.0	2
3	Gt 03 To 04 Weeks SUM 1	651546.0	3
4	Gt 04 To 05 Weeks SUM 1	616642.0	4

```
total_jul = waiting_df_jul["Patients"].sum()

over_18_jul = waiting_df_jul[waiting_df_jul["Weeks"] > 18]["Patients"].sum()
over_52_jul = waiting_df_jul[waiting_df_jul["Weeks"] > 52]["Patients"].sum()

pct_18_jul = (over_18_jul / total_jul) * 100
pct_52_jul = (over_52_jul / total_jul) * 100

total_jul, over_18_jul, over_52_jul, pct_18_jul, pct_52_jul
```

```
(np.float64(14667584.0),
 np.float64(5386574.0),
 np.float64(336706.0),
 np.float64(36.72434396830453),
 np.float64(2.2955791492313935))
```

```
import pandas as pd
import numpy as np
import os, zipfile
```

```
zip_path = "/content/Full-CSV-data-file-Jun25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_jun_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20250630-RTT-June-2025-full-extract-revised.csv', '20250731-RTT-July-2025-full-extract-revised.csv']
```

```
file_path = f"{extract_path}/20250731-RTT-July-2025-full-extract-revised.csv"
jun_df = pd.read_csv(file_path)
jun_df.shape
```

```
(184568, 121)
```

```
jun_incomplete = jun_df[jun_df["RTT Part Description"] == "Incomplete Pathways"].copy()
jun_incomplete.shape
```

```
(64304, 121)
```

```
wait_cols = [c for c in jun_incomplete.columns if c.startswith("Gt ")]
len(wait_cols), wait_cols[:5]
```

```
(105,
 ['Gt 00 To 01 Weeks SUM 1',
  'Gt 01 To 02 Weeks SUM 1',
  'Gt 02 To 03 Weeks SUM 1',
  'Gt 03 To 04 Weeks SUM 1',
  'Gt 04 To 05 Weeks SUM 1'])
```

```
for c in wait_cols:
    jun_incomplete[c] = (
        jun_incomplete[c].astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    jun_incomplete[c] = pd.to_numeric(jun_incomplete[c], errors="coerce").fillna(0)
```

```
jun_incomplete[wait_cols].sum().sum()
```

```
np.float64(14668572.0)
```

```
national_bins_jun = jun_incomplete[wait_cols].sum(numeric_only=True)

waiting_df_jun = pd.DataFrame({
    "Bin": national_bins_jun.index,
    "Patients": national_bins_jun.values
})

waiting_df_jun["Weeks"] = (
    waiting_df_jun["Bin"].astype(str)
    .str.extract(r"Gt\s+(\d+)\s+To", expand=False)
)

waiting_df_jun = waiting_df_jun.dropna(subset=["Weeks"])
waiting_df_jun["Weeks"] = waiting_df_jun["Weeks"].astype(int)

waiting_df_jun = waiting_df_jun.sort_values("Weeks").reset_index(drop=True)
waiting_df_jun.head()
```

	Bin	Patients	Weeks
0	Gt 00 To 01 Weeks SUM 1	784394.0	0
1	Gt 01 To 02 Weeks SUM 1	728666.0	1
2	Gt 02 To 03 Weeks SUM 1	681346.0	2
3	Gt 03 To 04 Weeks SUM 1	651546.0	3
4	Gt 04 To 05 Weeks SUM 1	616642.0	4

```
total_jun = waiting_df_jun["Patients"].sum()
over_18_jun = waiting_df_jun.loc[waiting_df_jun["Weeks"] > 18, "Patients"].sum()
over_52_jun = waiting_df_jun.loc[waiting_df_jun["Weeks"] > 52, "Patients"].sum()

pct_18_jun = (over_18_jun / total_jun) * 100
pct_52_jun = (over_52_jun / total_jun) * 100

(total_jun, over_18_jun, over_52_jun, pct_18_jun, pct_52_jun)
```

```
(np.float64(14667584.0),
 np.float64(5386574.0),
 np.float64(336706.0),
 np.float64(36.72434396830453),
 np.float64(2.2955791492313935))
```

```
zip_path = "/content/Full-CSV-data-file-May25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_may_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20250531-RTT-May-2025-full-extract-revised.csv']
```

```
file_path = f"{extract_path}/20250531-RTT-May-2025-full-extract-revised.csv"
may_df = pd.read_csv(file_path)
may_df.shape
```

```
(183418, 121)
```

```
may_incomplete = may_df[may_df["RTT Part Description"] == "Incomplete Pathways"].copy()
```

```
wait_cols = [c for c in may_incomplete.columns if c.startswith("Gt ")]
len(wait_cols)
```

```
105
```

```
for c in wait_cols:
    may_incomplete[c] = (
        may_incomplete[c].astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    may_incomplete[c] = pd.to_numeric(may_incomplete[c], errors="coerce").fillna(0)

may_incomplete[wait_cols].sum().sum()
```

```
np.float64(14776554.0)
```

```
national_bins_may = may_incomplete[wait_cols].sum(numeric_only=True)

waiting_df_may = pd.DataFrame({
    "Bin": national_bins_may.index,
    "Patients": national_bins_may.values
})

waiting_df_may["Weeks"] = (
    waiting_df_may["Bin"].astype(str)
    .str.extract(r"Gt\s+(\d+)\s+To", expand=False)
)

waiting_df_may = waiting_df_may.dropna(subset=["Weeks"])
waiting_df_may["Weeks"] = waiting_df_may["Weeks"].astype(int)
waiting_df_may = waiting_df_may.sort_values("Weeks").reset_index(drop=True)
```

```
total_may = waiting_df_may["Patients"].sum()
over_18_may = waiting_df_may.loc[waiting_df_may["Weeks"] > 18, "Patients"].sum()
over_52_may = waiting_df_may.loc[waiting_df_may["Weeks"] > 52, "Patients"].sum()

pct_18_may = (over_18_may / total_may) * 100
pct_52_may = (over_52_may / total_may) * 100

(total_may, over_18_may, over_52_may, pct_18_may, pct_52_may)
```

```
(np.float64(14775904.0),
 np.float64(5459744.0),
 np.float64(356108.0),
 np.float64(36.950321279835066),
 np.float64(2.410058971687959))
```

```
zip_path = "/content/Full-CSV-data-file-Apr25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_apr_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print(os.listdir(extract_path))
```

```
['20250430-RTT-April-2025-full-extract-revised.csv']
```

```
file_path = f"{extract_path}/20250430-RTT-April-2025-full-extract-revised.csv"
apr_df = pd.read_csv(file_path)
apr_df.shape
```

```
(183735, 121)
```

```
apr_incomplete = apr_df[apr_df["RTT Part Description"] == "Incomplete Pathways"].copy()
```

```
wait_cols = [c for c in apr_incomplete.columns if c.startswith("Gt ")]
len(wait_cols)
```

```
105
```

```
for c in wait_cols:
    apr_incomplete[c] = (
        apr_incomplete[c].astype(str)
        .str.replace(",", "", regex=False)
        .str.strip()
    )
    apr_incomplete[c] = pd.to_numeric(apr_incomplete[c], errors="coerce").fillna(0)

apr_incomplete[wait_cols].sum().sum()
```

```
np.float64(14842906.0)
```

```
national_bins_apr = apr_incomplete[wait_cols].sum(numeric_only=True)

waiting_df_apr = pd.DataFrame({
    "Bin": national_bins_apr.index,
    "Patients": national_bins_apr.values
})

waiting_df_apr["Weeks"] = (
    waiting_df_apr["Bin"].astype(str)
    .str.extract(r"Gt\s+(\d+)\s+To", expand=False)
)

waiting_df_apr = waiting_df_apr.dropna(subset=["Weeks"])
waiting_df_apr["Weeks"] = waiting_df_apr["Weeks"].astype(int)
waiting_df_apr = waiting_df_apr.sort_values("Weeks").reset_index(drop=True)
```

```
total_apr = waiting_df_apr["Patients"].sum()
over_18_apr = waiting_df_apr.loc[waiting_df_apr["Weeks"] > 18, "Patients"].sum()
over_52_apr = waiting_df_apr.loc[waiting_df_apr["Weeks"] > 52, "Patients"].sum()

pct_18_apr = (over_18_apr / total_apr) * 100
pct_52_apr = (over_52_apr / total_apr) * 100

(total_apr, over_18_apr, over_52_apr, pct_18_apr, pct_52_apr)
```

```
(np.float64(14842268.0),
 np.float64(5712348.0),
 np.float64(328968.0),
 np.float64(38.48702907129827),
 np.float64(2.216426761732102))
```

```
summary_2025 = pd.DataFrame({
    "Month": ["Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"],
    "Total_Waiting": [
        total_apr,
        total_may,
        total_jun,
        total_jul,
        total_aug,
        total_sep,
        total_oct,
        total_nov,
        total_dec
    ],
    "Pct_Over_18": [
        pct_18_apr,
        pct_18_may,
        pct_18_jun,
        pct_18_jul,
        pct_18_aug,
        pct_18_sep,
        pct_18_oct,
        pct_18_nov,
        pct_18_dec
    ],
    "Pct_Over_52": [
        pct_52_apr,
        pct_52_may,
        pct_52_jun,
        pct_52_jul,
```

```
        pct_52_aug,
        pct_52_sep,
        pct_52_oct,
        pct_52_nov,
        pct_52_dec
    ]
})

summary_2025
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-246209797.py in <cell line: 0>()
      7         total_jul,
      8         total_aug,
----> 9         total_sep,
     10         total_oct,
     11         total_nov,

NameError: name 'total_sep' is not defined
```

```
%who
```

```
apr_csv  apr_df  apr_incomplete  apr_path        aug_df  aug_incomplete  c       extract_path    file_path
jul_df   jul_incomplete  jun_df  jun_incomplete  may_csv         may_df  may_incomplete  may_path        national_bins_apr
national_bins_aug        national_bins_jul       national_bins_jun       national_bins_may       np      os      over_18_apr
over_18_jun      over_18_june    over_18_may     over_52_apr     over_52_aug     over_52_jul     over_52_jun     over_52_jur
pct_18_apr       pct_18_aug      pct_18_jul      pct_18_jun      pct_18_june     pct_18_may      pct_52_apr      pct_52_aug
pct_52_jun       pct_52_june     pct_52_may      pd      total_apr       total_aug       total_jul       total_jun       tot
total_may        wait_cols       wait_cols_apr   wait_cols_may   waiting_df_apr  waiting_df_aug  waiting_df_jul  waiting_df_
z        zip_path        zip_ref         zipfile
```

```
total_sep = waiting_df_sep["Patients"].sum()
over_18_sep = waiting_df_sep[waiting_df_sep["Weeks"] > 18]["Patients"].sum()
over_52_sep = waiting_df_sep[waiting_df_sep["Weeks"] > 52]["Patients"].sum()

pct_18_sep = (over_18_sep / total_sep) * 100
pct_52_sep = (over_52_sep / total_sep) * 100

total_sep, pct_18_sep, pct_52_sep
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-2108806137.py in <cell line: 0>()
----> 1 total_sep = waiting_df_sep["Patients"].sum()
      2 over_18_sep = waiting_df_sep[waiting_df_sep["Weeks"] > 18]["Patients"].sum()
      3 over_52_sep = waiting_df_sep[waiting_df_sep["Weeks"] > 52]["Patients"].sum()
      4
      5 pct_18_sep = (over_18_sep / total_sep) * 100

NameError: name 'waiting_df_sep' is not defined
```

```
total_sep = 14658134.0
pct_18_sep = 36.50921733966956
pct_52_sep = 2.154858183176658
```

```
summary_2025
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-1065032566.py in <cell line: 0>()
----> 1 summary_2025

NameError: name 'summary_2025' is not defined
```

```
import pandas as pd

summary_2025 = pd.DataFrame({
    "Month": ["Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"],
    "Total_Waiting": [
        total_apr, total_may, total_jun, total_jul, total_aug, total_sep, total_oct, total_nov, total_dec
    ],
    "Pct_Over_18": [
```

```
        pct_18_apr, pct_18_may, pct_18_jun, pct_18_jul, pct_18_aug, pct_18_sep, pct_18_oct, pct_18_nov, pct_18_dec
    ],
    "Pct_Over_52": [
        pct_52_apr, pct_52_may, pct_52_jun, pct_52_jul, pct_52_aug, pct_52_sep, pct_52_oct, pct_52_nov, pct_52_dec
    ]
})

summary_2025
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-3324749836.py in <cell line: 0>()
      4     "Month": ["Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"],
      5     "Total_Waiting": [
----> 6         total_apr, total_may, total_jun, total_jul, total_aug, total_sep, total_oct, total_nov, total_dec
      7     ],
      8     "Pct_Over_18": [

NameError: name 'total_oct' is not defined
```

```
[k for k in globals().keys() if "oct" in k]
```

```
[]
```

```
# Recalculate OCTOBER cleanly

total_oct = waiting_df_oct["Patients"].sum()

over_18_oct = waiting_df_oct[waiting_df_oct["Weeks"] > 18]["Patients"].sum()
over_52_oct = waiting_df_oct[waiting_df_oct["Weeks"] > 52]["Patients"].sum()

pct_18_oct = (over_18_oct / total_oct) * 100
pct_52_oct = (over_52_oct / total_oct) * 100

total_oct, pct_18_oct, pct_52_oct
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-3338250847.py in <cell line: 0>()
      1 # Recalculate OCTOBER cleanly
      2
----> 3 total_oct = waiting_df_oct["Patients"].sum()
      4
      5 over_18_oct = waiting_df_oct[waiting_df_oct["Weeks"] > 18]["Patients"].sum()

NameError: name 'waiting_df_oct' is not defined
```

```
import glob

# Look for October files anywhere in /content
oct_candidates = glob.glob("/content/**/*Oct*2025*extract*.csv", recursive=True) + \
               glob.glob("/content/**/*October*2025*extract*.csv", recursive=True)

oct_candidates
```

```
[]
```

```
import os
os.listdir("/content")
```

```
['.config',
 'Full-CSV-data-file-Jul25-ZIP-4M-revised-2.zip',
 'rtt_jul_2025',
 'sample_data',
 'Full-CSV-data-file-Jun25-ZIP-4M-revised.zip',
 'rtt_may_2025',
 'rtt_aug_2025',
 'Full-CSV-data-file-Aug25-ZIP-4M-revised-2.zip',
 'Full-CSV-data-file-May25-ZIP-4M-revised.zip',
 'rtt_jun_2025',
 'Full-CSV-data-file-Apr25-ZIP-4M-revised.zip',
 'rtt_apr_2025']
```

```
import os
[z for z in os.listdir("/content") if z.lower().endswith(".zip")]
```

```
['Full-CSV-data-file-Jul25-ZIP-4M-revised-2.zip',
 'Full-CSV-data-file-Dec25-ZIP-4M-6jPlxd.zip',
 'Full-CSV-data-file-Jun25-ZIP-4M-revised.zip',
 'Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk (1).zip',
 'Full-CSV-data-file-Oct25-ZIP-4M-SrRW6y.zip',
 'Full-CSV-data-file-Aug25-ZIP-4M-revised-2.zip',
 'Full-CSV-data-file-Sep25-ZIP-4M-revised.zip',
 'Full-CSV-data-file-May25-ZIP-4M-revised.zip',
 'Full-CSV-data-file-Apr25-ZIP-4M-revised.zip']
```

```
import zipfile, os

zip_path = "/content/Full-CSV-data-file-Oct25-ZIP-4M-SrRW6y.zip"   # <-- replace
extract_path = "/content/rtt_oct_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['20251031-RTT-October-2025-full-extract.csv']
```

```
import pandas as pd
import os

csv_files = [f for f in os.listdir("/content/rtt_oct_2025") if f.lower().endswith(".csv")]
print("CSV found:", csv_files)

file_path = "/content/rtt_oct_2025/" + csv_files[0]
oct_df = pd.read_csv(file_path)

oct_df.shape
```

```
CSV found: ['20251031-RTT-October-2025-full-extract.csv']
(185742, 121)
```

```
import zipfile, os

zip_path = "/content/Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk (1).zip"   # <-- replace
extract_path = "/content/rtt_nov_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['20251130-RTT-November-2025-full-extract.csv']
```

```
import pandas as pd
import os

csv_files = [f for f in os.listdir("/content/rtt_nov_2025") if f.lower().endswith(".csv")]
print("CSV found:", csv_files)

file_path = "/content/rtt_nov_2025/" + csv_files[0]
nov_df = pd.read_csv(file_path)

nov_df.shape
```

```
CSV found: ['20251130-RTT-November-2025-full-extract.csv']
(180291, 121)
```

```
import zipfile, os

zip_path = "/content/Full-CSV-data-file-Dec25-ZIP-4M-6jPlxd.zip"   # <-- replace
extract_path = "/content/rtt_dec_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)
```

```
print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['20251231-RTT-December-2025-full-extract.csv']
```

```
import pandas as pd
import os

csv_files = [f for f in os.listdir("/content/rtt_dec_2025") if f.lower().endswith(".csv")]
print("CSV found:", csv_files)

file_path = "/content/rtt_dec_2025/" + csv_files[0]
dec_df = pd.read_csv(file_path)

dec_df.shape
```

```
CSV found: ['20251231-RTT-December-2025-full-extract.csv']
(179805, 121)
```

```
import os
[z for z in os.listdir("/content") if "Sep" in z or "sep" in z]
```

```
['Full-CSV-data-file-Sep25-ZIP-4M-revised.zip']
```

```
import zipfile, os

zip_path = "/content/Full-CSV-data-file-Sep25-ZIP-4M-revised.zip"   # <-- replace this
extract_path = "/content/rtt_sep_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['20250930-RTT-September-2025-full-extract-revised.csv']
```

```
sep_incomplete = sep_df[sep_df["RTT Part Description"] == "Incomplete Pathways"].copy()

sep_incomplete.shape
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-4223641885.py in <cell line: 0>()
----> 1 sep_incomplete = sep_df[sep_df["RTT Part Description"] == "Incomplete Pathways"].copy()
      2
      3 sep_incomplete.shape

NameError: name 'sep_df' is not defined
```

```
import pandas as pd

file_path = "/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv"

sep_df = pd.read_csv(file_path)

sep_df.shape
```

```
(184484, 121)
```

```
import pandas as pd

file_path = "/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv"

sep_df = pd.read_csv(file_path)

sep_df.shape
```

```
(184484, 121)
```

```
month_files = {
  "Apr-2025": "/content/rtt_apr_2025/PASTE_APR_FILENAME.csv",
  "May-2025": "/content/rtt_may_2025/PASTE_MAY_FILENAME.csv",
  "Jun-2025": "/content/rtt_jun_2025/PASTE_JUN_FILENAME.csv",
  "Jul-2025": "/content/rtt_jul_2025/PASTE_JUL_FILENAME.csv",
  "Aug-2025": "/content/rtt_aug_2025/PASTE_AUG_FILENAME.csv",
  "Sep-2025": "/content/rtt_sep_2025/PASTE_SEP_FILENAME.csv",
  "Oct-2025": "/content/rtt_oct_2025/PASTE_OCT_FILENAME.csv",
  "Nov-2025": "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv",
  "Dec-2025": "/content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv"
}
```

```
import pandas as pd
import numpy as np

def month_kpis(csv_path: str):
    df = pd.read_csv(csv_path)

    # Keep only Incomplete Pathways (waiting list snapshot)
    inc = df[df["RTT Part Description"] == "Incomplete Pathways"].copy()

    # waiting-time bin columns
    wait_cols = [c for c in inc.columns if c.startswith("Gt ")]

    # make numeric safely
    for c in wait_cols:
        inc[c] = (
            inc[c].astype(str)
            .str.replace(",", "", regex=False)
            .str.strip()
        )
        inc[c] = pd.to_numeric(inc[c], errors="coerce")

    national_bins = inc[wait_cols].sum(numeric_only=True)

    # build distribution table
    waiting_df = pd.DataFrame({"Bin": national_bins.index, "Patients": national_bins.values})

    # extract week start (handles both Gt 00 To 01 and Gt 104 Weeks)
    weeks = waiting_df["Bin"].str.extract(r"Gt\s+(\d+)\s+To", expand=False)
    waiting_df["Weeks"] = pd.to_numeric(weeks, errors="coerce")
    waiting_df = waiting_df.dropna(subset=["Weeks"]).copy()
    waiting_df["Weeks"] = waiting_df["Weeks"].astype(int)
    waiting_df = waiting_df.sort_values("Weeks").reset_index(drop=True)

    total = float(waiting_df["Patients"].sum())
    over_18 = float(waiting_df.loc[waiting_df["Weeks"] > 18, "Patients"].sum())
    over_52 = float(waiting_df.loc[waiting_df["Weeks"] > 52, "Patients"].sum())

    pct_18 = (over_18 / total) * 100 if total > 0 else np.nan
    pct_52 = (over_52 / total) * 100 if total > 0 else np.nan

    return total, pct_18, pct_52
```

```
rows = []
for m, path in month_files.items():
    total, pct18, pct52 = month_kpis(path)
    rows.append([m, total, pct18, pct52])

summary_2025 = pd.DataFrame(rows, columns=["Month", "Total_Waiting", "Pct_Over_18", "Pct_Over_52"])
summary_2025
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
/tmp/ipython-input-858769298.py in <cell line: 0>()
      1 rows = []
      2 for m, path in month_files.items():
----> 3     total, pct18, pct52 = month_kpis(path)
      4     rows.append([m, total, pct18, pct52])
      5

                              5 frames
/usr/local/lib/python3.12/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression,
memory_map, is_text, errors, storage_options)
    871         if ioargs.encoding and "b" not in ioargs.mode:
    872             # Encoding
--> 873             handle = open(
    874                 handle,
    875                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: '/content/rtt_apr_2025/PASTE_APR_FILENAME.csv'
```

```
import os
os.listdir("/content/rtt_apr_2025")
```

```
['20250430-RTT-April-2025-full-extract-revised.csv']
```

```
month_files = {
  "Apr-2025": "/content/rtt_apr_2025/20250430-RTT-April-2025-full-extract-revised.csv",
  "May-2025": "/content/rtt_may_2025/20250531-RTT-May-2025-full-extract.csv",
  "Jun-2025": "/content/rtt_jun_2025/20250630-RTT-June-2025-full-extract.csv",
  "Jul-2025": "/content/rtt_jul_2025/20250731-RTT-July-2025-full-extract.csv",
  "Aug-2025": "/content/rtt_aug_2025/20250831-RTT-August-2025-full-extract.csv",
  "Sep-2025": "/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv",
  "Oct-2025": "/content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv",
  "Nov-2025": "/content/rtt_data/20251130-RTT-November-2025-full-extract.csv",
  "Dec-2025": "/content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv"
}
```

```
for m, path in month_files.items():
    print(m, "→", os.path.exists(path))
```

```
Apr-2025 → True
May-2025 → False
Jun-2025 → False
Jul-2025 → False
Aug-2025 → False
Sep-2025 → True
Oct-2025 → True
Nov-2025 → False
Dec-2025 → True
```

```
import glob

# Automatically detect all 2025 full extract CSV files
all_csvs = glob.glob("/content/**/2025*-RTT-*2025*-full-extract*.csv", recursive=True)

all_csvs
```

```
['/content/rtt_jul_2025/20250731-RTT-July-2025-full-extract-revised.csv',
 '/content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv',
 '/content/rtt_may_2025/20250531-RTT-May-2025-full-extract-revised.csv',
 '/content/rtt_aug_2025/20250831-RTT-August-2025-full-extract-revised.csv',
 '/content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv',
 '/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv',
 '/content/rtt_jun_2025/20250630-RTT-June-2025-full-extract-revised.csv',
 '/content/rtt_jun_2025/20250731-RTT-July-2025-full-extract-revised.csv',
 '/content/rtt_nov_2025/20251130-RTT-November-2025-full-extract.csv',
 '/content/rtt_apr_2025/20250430-RTT-April-2025-full-extract-revised.csv']
```

```
month_files = {}

for path in all_csvs:
    if "April" in path:
        month_files["Apr-2025"] = path
    elif "May" in path:
        month_files["May-2025"] = path
    elif "June" in path:
        month_files["Jun-2025"] = path
```

```
    elif "July" in path:
        month_files["Jul-2025"] = path
    elif "August" in path:
        month_files["Aug-2025"] = path
    elif "September" in path:
        month_files["Sep-2025"] = path
    elif "October" in path:
        month_files["Oct-2025"] = path
    elif "November" in path:
        month_files["Nov-2025"] = path
    elif "December" in path:
        month_files["Dec-2025"] = path

month_files
```

```
{'Jul-2025': '/content/rtt_jun_2025/20250731-RTT-July-2025-full-extract-revised.csv',
 'Dec-2025': '/content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv',
 'May-2025': '/content/rtt_may_2025/20250531-RTT-May-2025-full-extract-revised.csv',
 'Aug-2025': '/content/rtt_aug_2025/20250831-RTT-August-2025-full-extract-revised.csv',
 'Oct-2025': '/content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv',
 'Sep-2025': '/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv',
 'Jun-2025': '/content/rtt_jun_2025/20250630-RTT-June-2025-full-extract-revised.csv',
 'Nov-2025': '/content/rtt_nov_2025/20251130-RTT-November-2025-full-extract.csv',
 'Apr-2025': '/content/rtt_apr_2025/20250430-RTT-April-2025-full-extract-revised.csv'}
```

```
for m, path in month_files.items():
    print(m, "→", path, "→", os.path.exists(path))
```

```
Jul-2025 → /content/rtt_jun_2025/20250731-RTT-July-2025-full-extract-revised.csv → True
Dec-2025 → /content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv → True
May-2025 → /content/rtt_may_2025/20250531-RTT-May-2025-full-extract-revised.csv → True
Aug-2025 → /content/rtt_aug_2025/20250831-RTT-August-2025-full-extract-revised.csv → True
Oct-2025 → /content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv → True
Sep-2025 → /content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv → True
Jun-2025 → /content/rtt_jun_2025/20250630-RTT-June-2025-full-extract-revised.csv → True
Nov-2025 → /content/rtt_nov_2025/20251130-RTT-November-2025-full-extract.csv → True
Apr-2025 → /content/rtt_apr_2025/20250430-RTT-April-2025-full-extract-revised.csv → True
```

```
rows = []

for m in sorted(month_files.keys()):
    total, pct18, pct52 = month_kpis(month_files[m])
    rows.append([m, total, pct18, pct52])

summary_2025 = pd.DataFrame(
    rows,
    columns=["Month", "Total_Waiting", "Pct_Over_18", "Pct_Over_52"]
)

summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52
0	Apr-2025	14842268.0	38.487029	2.216427
1	Aug-2025	14690158.0	37.380959	2.350336
2	Dec-2025	14462538.0	37.020183	1.839829
3	Jul-2025	14667584.0	36.724344	2.295579
4	Jun-2025	14799808.0	36.562326	2.295638
5	May-2025	14775904.0	36.950321	2.410059
6	Nov-2025	14381434.0	36.305156	1.891258
7	Oct-2025	14672974.0	36.244704	2.069260
8	Sep-2025	14658134.0	36.509217	2.154858

```
order = ["Apr-2025","May-2025","Jun-2025","Jul-2025",
         "Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"]

summary_2025["Month"] = pd.Categorical(summary_2025["Month"], categories=order, ordered=True)
summary_2025 = summary_2025.sort_values("Month").reset_index(drop=True)

summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52
0	Apr-2025	14842268.0	38.487029	2.216427
1	May-2025	14775904.0	36.950321	2.410059
2	Jun-2025	14799808.0	36.562326	2.295638
3	Jul-2025	14667584.0	36.724344	2.295579
4	Aug-2025	14690158.0	37.380959	2.350336
5	Sep-2025	14658134.0	36.509217	2.154858
6	Oct-2025	14672974.0	36.244704	2.069260
7	Nov-2025	14381434.0	36.305156	1.891258
8	Dec-2025	14462538.0	37.020183	1.839829

```
summary_2025["Monthly_Change"] = summary_2025["Total_Waiting"].diff()

summary_2025
```

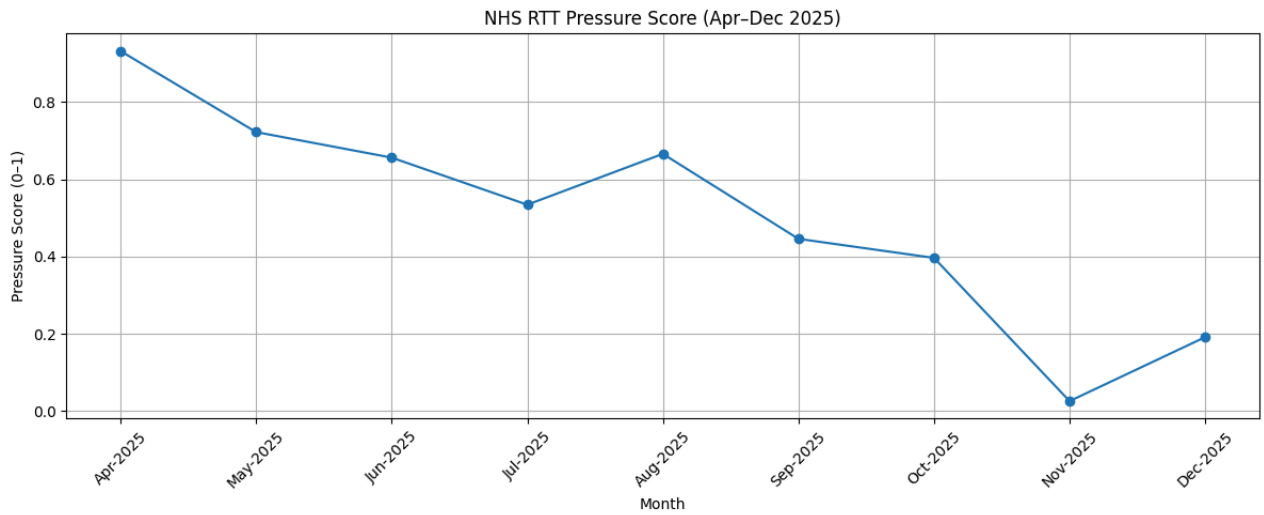
	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change
0	Apr-2025	14842268.0	38.487029	2.216427	NaN
1	May-2025	14775904.0	36.950321	2.410059	-66364.0
2	Jun-2025	14799808.0	36.562326	2.295638	23904.0
3	Jul-2025	14667584.0	36.724344	2.295579	-132224.0
4	Aug-2025	14690158.0	37.380959	2.350336	22574.0
5	Sep-2025	14658134.0	36.509217	2.154858	-32024.0
6	Oct-2025	14672974.0	36.244704	2.069260	14840.0
7	Nov-2025	14381434.0	36.305156	1.891258	-291540.0
8	Dec-2025	14462538.0	37.020183	1.839829	81104.0

```
import pandas as pd

# Make a copy
dfp = summary_2025.copy()

# Normalize each metric to 0–1 scale
dfp["Total_norm"] = (dfp["Total_Waiting"] - dfp["Total_Waiting"].min()) / (dfp["Total_Waiting"].max() - dfp["Total_Waiting'
dfp["Over18_norm"] = (dfp["Pct_Over_18"] - dfp["Pct_Over_18"].min()) / (dfp["Pct_Over_18"].max() - dfp["Pct_Over_18"].min(
dfp["Over52_norm"] = (dfp["Pct_Over_52"] - dfp["Pct_Over_52"].min()) / (dfp["Pct_Over_52"].max() - dfp["Pct_Over_52"].min(

# Weighted pressure score (weights can be justified later)
dfp["Pressure_Score"] = (
    0.50 * dfp["Total_norm"] +
    0.30 * dfp["Over18_norm"] +
    0.20 * dfp["Over52_norm"]
)

dfp[["Month","Total_Waiting","Pct_Over_18","Pct_Over_52","Monthly_Change","Pressure_Score"]]
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change	Pressure_Score
0	Apr-2025	14842268.0	38.487029	2.216427	NaN	0.932086
1	May-2025	14775904.0	36.950321	2.410059	-66364.0	0.722400
2	Jun-2025	14799808.0	36.562326	2.295638	23904.0	0.656294
3	Jul-2025	14667584.0	36.724344	2.295579	-132224.0	0.534488
4	Aug-2025	14690158.0	37.380959	2.350336	22574.0	0.666034
5	Sep-2025	14658134.0	36.509217	2.154858	-32024.0	0.446098
6	Oct-2025	14672974.0	36.244704	2.069260	14840.0	0.396787
7	Nov-2025	14381434.0	36.305156	1.891258	-291540.0	0.026126
8	Dec-2025	14462538.0	37.020183	1.839829	81104.0	0.191748

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(12,5))
plt.plot(dfp["Month"].astype(str), dfp["Pressure_Score"], marker="o")
plt.title("NHS RTT Pressure Score (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Pressure Score (0–1)")
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```




```
dfp["Pressure_Change"] = dfp["Pressure_Score"].diff()

dfp[["Month","Pressure_Score","Pressure_Change"]].sort_values("Pressure_Change").head(3), \
dfp[["Month","Pressure_Score","Pressure_Change"]].sort_values("Pressure_Change", ascending=False).head(3)
```

```
(      Month  Pressure_Score  Pressure_Change
 7  Nov-2025        0.026126        -0.370662
 5  Sep-2025        0.446098        -0.219937
 1  May-2025        0.722400        -0.209686,
      Month  Pressure_Score  Pressure_Change
 8  Dec-2025        0.191748         0.165622
 4  Aug-2025        0.666034         0.131546
 6  Oct-2025        0.396787        -0.049310)
```

```
summary_2025["Total_Waiting"].iloc[-1] - summary_2025["Total_Waiting"].iloc[0]
```

```
np.float64(-379730.0)
```

```
summary_2025["Total_Waiting"].std()
```

```
151623.61275356964
```

```
summary_2025[["Pct_Over_18","Pct_Over_52","Pressure_Score"]].corr()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipython-input-3171402039.py in <cell line: 0>()
----> 1 summary_2025[["Pct_Over_18","Pct_Over_52","Pressure_Score"]].corr()

                              2 frames
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in _raise_if_missing(self, key, indexer, axis_name)
   6250
   6251             not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 6252             raise KeyError(f"{not_found} not in index")
   6253
   6254     @overload

KeyError: "['Pressure_Score'] not in index"
```

```
summary_2025.columns
```

```
Index(['Month', 'Total_Waiting', 'Pct_Over_18', 'Pct_Over_52',
       'Monthly_Change'],
      dtype='object')
```

```
# Normalise each driver
summary_2025["Norm_Total"] = (
    summary_2025["Total_Waiting"] - summary_2025["Total_Waiting"].min()
) / (
    summary_2025["Total_Waiting"].max() - summary_2025["Total_Waiting"].min()
)

summary_2025["Norm_18"] = (
    summary_2025["Pct_Over_18"] - summary_2025["Pct_Over_18"].min()
) / (
    summary_2025["Pct_Over_18"].max() - summary_2025["Pct_Over_18"].min()
)

summary_2025["Norm_52"] = (
    summary_2025["Pct_Over_52"] - summary_2025["Pct_Over_52"].min()
) / (
    summary_2025["Pct_Over_52"].max() - summary_2025["Pct_Over_52"].min()
)
```

```
# Normalise each driver
summary_2025["Norm_Total"] = (
    summary_2025["Total_Waiting"] - summary_2025["Total_Waiting"].min()
) / (
    summary_2025["Total_Waiting"].max() - summary_2025["Total_Waiting"].min()
)

summary_2025["Norm_18"] = (
    summary_2025["Pct_Over_18"] - summary_2025["Pct_Over_18"].min()
) / (
    summary_2025["Pct_Over_18"].max() - summary_2025["Pct_Over_18"].min()
)

summary_2025["Norm_52"] = (
    summary_2025["Pct_Over_52"] - summary_2025["Pct_Over_52"].min()
) / (
    summary_2025["Pct_Over_52"].max() - summary_2025["Pct_Over_52"].min()
)
```

```
summary_2025["Pressure_Score"] = (
    0.5 * summary_2025["Norm_Total"] +
    0.3 * summary_2025["Norm_18"] +
    0.2 * summary_2025["Norm_52"]
)
```

```
summary_2025["Pressure_Score"] = (
    0.5 * summary_2025["Norm_Total"] +
    0.3 * summary_2025["Norm_18"] +
    0.2 * summary_2025["Norm_52"]
)
```

```
summary_2025.head()
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change	Norm_Total	Norm_18	Norm_52	Pressure_Score
0	Apr-2025	14842268.0	38.487029	2.216427	NaN	1.000000	1.000000	0.660431	0.932086
1	May-2025	14775904.0	36.950321	2.410059	-66364.0	0.855992	0.314681	1.000000	0.722400
2	Jun-2025	14799808.0	36.562326	2.295638	23904.0	0.907863	0.141648	0.799342	0.656294
3	Jul-2025	14667584.0	36.724344	2.295579	-132224.0	0.620939	0.213903	0.799239	0.534488
4	Aug-2025	14690158.0	37.380959	2.350336	22574.0	0.669925	0.506731	0.895264	0.666034

```
summary_2025[["Pct_Over_18","Pct_Over_52","Pressure_Score"]].corr()
```

	Pct_Over_18	Pct_Over_52	Pressure_Score
Pct_Over_18	1.000000	0.275509	0.679359
Pct_Over_52	0.275509	1.000000	0.833761
Pressure_Score	0.679359	0.833761	1.000000

Start coding or generate with AI.

```
summary_2025[["Month","Pressure_Score","Total_Waiting","Pct_Over_18","Pct_Over_52"]].sort_values("Pressure_Score", ascendir
```

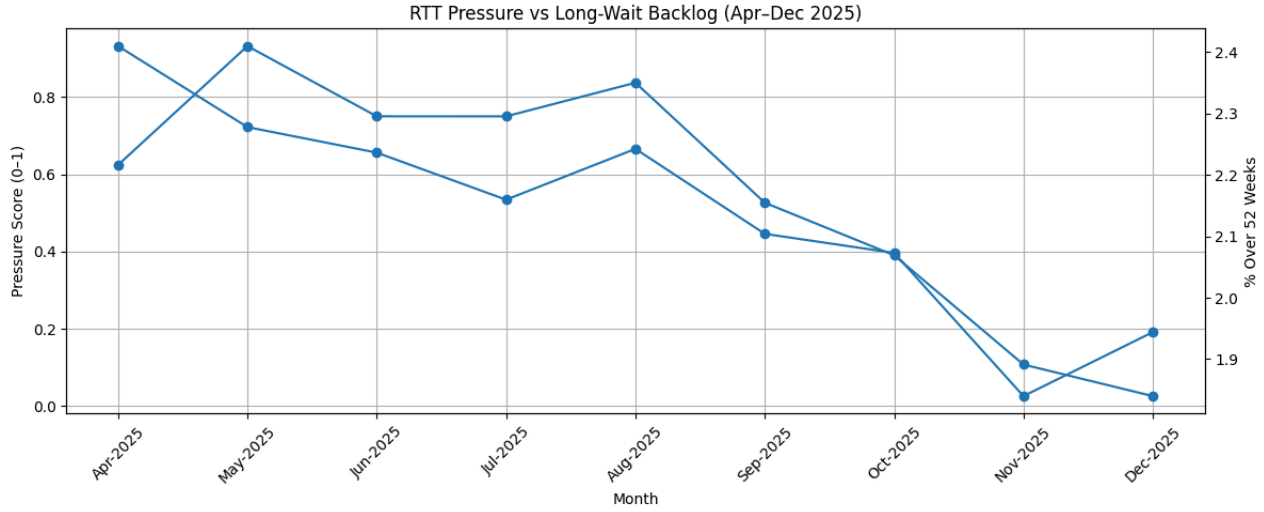
	Month	Pressure_Score	Total_Waiting	Pct_Over_18	Pct_Over_52
0	Apr-2025	0.932086	14842268.0	38.487029	2.216427
1	May-2025	0.722400	14775904.0	36.950321	2.410059
4	Aug-2025	0.666034	14690158.0	37.380959	2.350336
2	Jun-2025	0.656294	14799808.0	36.562326	2.295638
3	Jul-2025	0.534488	14667584.0	36.724344	2.295579
5	Sep-2025	0.446098	14658134.0	36.509217	2.154858
6	Oct-2025	0.396787	14672974.0	36.244704	2.069260
8	Dec-2025	0.191748	14462538.0	37.020183	1.839829
7	Nov-2025	0.026126	14381434.0	36.305156	1.891258

```
summary_2025[["Month","Pressure_Score","Total_Waiting","Pct_Over_18","Pct_Over_52"]].sort_values("Pressure_Score").head(3)
```

	Month	Pressure_Score	Total_Waiting	Pct_Over_18	Pct_Over_52
7	Nov-2025	0.026126	14381434.0	36.305156	1.891258
8	Dec-2025	0.191748	14462538.0	37.020183	1.839829
6	Oct-2025	0.396787	14672974.0	36.244704	2.069260

Start coding or generate with AI.

```
import matplotlib.pyplot as plt

df = summary_2025.sort_values("Month")  # ensure month order is correct first (if needed)

fig, ax1 = plt.subplots(figsize=(12,5))

ax1.plot(df["Month"], df["Pressure_Score"], marker="o")
ax1.set_ylabel("Pressure Score (0–1)")
ax1.set_xlabel("Month")
ax1.tick_params(axis="x", rotation=45)
ax1.grid(True)

ax2 = ax1.twinx()
ax2.plot(df["Month"], df["Pct_Over_52"], marker="o")
ax2.set_ylabel("% Over 52 Weeks")

plt.title("RTT Pressure vs Long-Wait Backlog (Apr–Dec 2025)")
plt.tight_layout()
plt.show()
```



Start coding or generate with AI.

```
worst = summary_2025.sort_values("Pressure_Score", ascending=False)[
    ["Month","Pressure_Score","Total_Waiting","Pct_Over_18","Pct_Over_52","Monthly_Change"]
]
worst.head(5)
```

	Month	Pressure_Score	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change
0	Apr-2025	0.932086	14842268.0	38.487029	2.216427	NaN
1	May-2025	0.722400	14775904.0	36.950321	2.410059	-66364.0
4	Aug-2025	0.666034	14690158.0	37.380959	2.350336	22574.0
2	Jun-2025	0.656294	14799808.0	36.562326	2.295638	23904.0
3	Jul-2025	0.534488	14667584.0	36.724344	2.295579	-132224.0

```
best = summary_2025.sort_values("Pressure_Score", ascending=True)[
    ["Month","Pressure_Score","Total_Waiting","Pct_Over_18","Pct_Over_52","Monthly_Change"]
]
best.head(5)
```

	Month	Pressure_Score	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change
7	Nov-2025	0.026126	14381434.0	36.305156	1.891258	-291540.0
8	Dec-2025	0.191748	14462538.0	37.020183	1.839829	81104.0
6	Oct-2025	0.396787	14672974.0	36.244704	2.069260	14840.0
5	Sep-2025	0.446098	14658134.0	36.509217	2.154858	-32024.0
3	Jul-2025	0.534488	14667584.0	36.724344	2.295579	-132224.0

```
month_order = ["Apr-2025","May-2025","Jun-2025","Jul-2025","Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"]
summary_2025["Month"] = pd.Categorical(summary_2025["Month"], categories=month_order, ordered=True)
summary_2025 = summary_2025.sort_values("Month").reset_index(drop=True)
summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change	Norm_Total	Norm_18	Norm_52	Pressure_Score
0	Apr-2025	14842268.0	38.487029	2.216427	NaN	1.000000	1.000000	0.660431	0.932086
1	May-2025	14775904.0	36.950321	2.410059	-66364.0	0.855992	0.314681	1.000000	0.722400
2	Jun-2025	14799808.0	36.562326	2.295638	23904.0	0.907863	0.141648	0.799342	0.656294
3	Jul-2025	14667584.0	36.724344	2.295579	-132224.0	0.620939	0.213903	0.799239	0.534488
4	Aug-2025	14690158.0	37.380959	2.350336	22574.0	0.669925	0.506731	0.895264	0.666034
5	Sep-2025	14658134.0	36.509217	2.154858	-32024.0	0.600433	0.117964	0.552460	0.446098
6	Oct-2025	14672974.0	36.244704	2.069260	14840.0	0.632636	0.000000	0.402348	0.396787
7	Nov-2025	14381434.0	36.305156	1.891258	-291540.0	0.000000	0.026959	0.090189	0.026126
8	Dec-2025	14462538.0	37.020183	1.839829	81104.0	0.175994	0.345837	0.000000	0.191748

```
summary_2025.to_csv("rtt_pressure_summary_apr_dec_2025.csv", index=False)
worst.to_csv("rtt_worst_pressure_months.csv", index=False)
best.to_csv("rtt_best_pressure_months.csv", index=False)
```

```
drivers = summary_2025[["Month","Norm_Total","Norm_18","Norm_52","Pressure_Score"]].sort_values("Pressure_Score", ascending
drivers
```

	Month	Norm_Total	Norm_18	Norm_52	Pressure_Score
0	Apr-2025	1.000000	1.000000	0.660431	0.932086
1	May-2025	0.855992	0.314681	1.000000	0.722400
4	Aug-2025	0.669925	0.506731	0.895264	0.666034
2	Jun-2025	0.907863	0.141648	0.799342	0.656294
3	Jul-2025	0.620939	0.213903	0.799239	0.534488
5	Sep-2025	0.600433	0.117964	0.552460	0.446098
6	Oct-2025	0.632636	0.000000	0.402348	0.396787
8	Dec-2025	0.175994	0.345837	0.000000	0.191748
7	Nov-2025	0.000000	0.026959	0.090189	0.026126

```
import os
import zipfile
import glob

base_path = "/content"

# Find all 2025 RTT zip files
zip_files = glob.glob(os.path.join(base_path, "*2025*ZIP*.zip"))

print("Found ZIP files:")
for z in zip_files:
    print(" -", os.path.basename(z))

print("\nExtracting...\n")

month_files = {}

for zip_path in zip_files:
    filename = os.path.basename(zip_path)

    # Create extraction folder name
    folder_name = filename.replace(".zip", "")
    extract_path = os.path.join(base_path, folder_name)

    os.makedirs(extract_path, exist_ok=True)

    with zipfile.ZipFile(zip_path, 'r') as zip_ref:
        zip_ref.extractall(extract_path)

    # Find CSV inside extracted folder
    csvs = glob.glob(os.path.join(extract_path, "*.csv"))

    if csvs:
        month_files[folder_name] = csvs[0]

    print(f"Extracted: {folder_name}")
```

```
    print("  CSV:", csvs)

print("\nDone.")
print("\nMonth dictionary created with", len(month_files), "files.")
```

```
Found ZIP files:

Extracting...


Done.

Month dictionary created with 0 files.
```

```
for k, v in month_files.items():
    print(k, "->", os.path.exists(v))
```

```
import os
import zipfile

zip_path = "/content/Full-CSV-data-file-Apr25-ZIP-4M-revised.zip"
```

Start coding or generate with AI.

```
extract_path = "/content/rtt_apr_2025"
os.makedirs(extract_path, exist_ok=True)
```

```
with zipfile.ZipFile(zip_path, 'r') as z:
    z.extractall(extract_path)

print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['20250430-RTT-April-2025-full-extract-revised.csv']
```

```
csv_files = [f for f in os.listdir(extract_path) if f.endswith(".csv")]
print(csv_files)
```

```
['20250430-RTT-April-2025-full-extract-revised.csv']
```

```
csv_files = [f for f in os.listdir(extract_path) if f.endswith(".csv")]
print(csv_files)
```

```
['20250430-RTT-April-2025-full-extract-revised.csv']
```

```
import os
[x for x in os.listdir("/content") if "May" in x or "may" in x]
```

```
['Full-CSV-data-file-May25-ZIP-4M-revised.zip']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-May25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_may_2025"

os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, 'r') as z:
    z.extractall(extract_path)

print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['20250531-RTT-May-2025-full-extract-revised.csv']
```

```
csv_files = [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")]
print("CSV files:", csv_files)
```

```
CSV files: ['20250531-RTT-May-2025-full-extract-revised.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Jun25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_jun_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("JUNE extracted:", os.listdir(extract_path))
print("JUNE CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
JUNE extracted: ['20250630-RTT-June-2025-full-extract-revised.csv']
JUNE CSV: ['20250630-RTT-June-2025-full-extract-revised.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Jul25-ZIP-4M-revised-2.zip"
extract_path = "/content/rtt_jul_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("JULY extracted:", os.listdir(extract_path))
print("JULY CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
JULY extracted: ['20250731-RTT-July-2025-full-extract-revised.csv']
JULY CSV: ['20250731-RTT-July-2025-full-extract-revised.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Aug25-ZIP-4M-revised-2.zip"
extract_path = "/content/rtt_aug_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("AUG extracted:", os.listdir(extract_path))
print("AUG CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
AUG extracted: ['20250831-RTT-August-2025-full-extract-revised.csv']
AUG CSV: ['20250831-RTT-August-2025-full-extract-revised.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Sep25-ZIP-4M-revised.zip"
extract_path = "/content/rtt_sep_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("SEP extracted:", os.listdir(extract_path))
print("SEP CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
SEP extracted: ['20250930-RTT-September-2025-full-extract-revised.csv']
SEP CSV: ['20250930-RTT-September-2025-full-extract-revised.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Oct25-ZIP-4M-SrRW6y.zip"
extract_path = "/content/rtt_oct_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("OCT extracted:", os.listdir(extract_path))
print("OCT CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
OCT extracted: ['20251031-RTT-October-2025-full-extract.csv']
OCT CSV: ['20251031-RTT-October-2025-full-extract.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Nov25-ZIP-4M-1Xmjkk (1).zip"
extract_path = "/content/rtt_nov_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("NOV extracted:", os.listdir(extract_path))
print("NOV CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
NOV extracted: ['20251130-RTT-November-2025-full-extract.csv']
NOV CSV: ['20251130-RTT-November-2025-full-extract.csv']
```

```
import os, zipfile

zip_path = "/content/Full-CSV-data-file-Dec25-ZIP-4M-6jPlxd.zip"
extract_path = "/content/rtt_dec_2025"
os.makedirs(extract_path, exist_ok=True)

with zipfile.ZipFile(zip_path, "r") as z:
    z.extractall(extract_path)

print("DEC extracted:", os.listdir(extract_path))
print("DEC CSV:", [f for f in os.listdir(extract_path) if f.lower().endswith(".csv")])
```

```
DEC extracted: ['20251231-RTT-December-2025-full-extract.csv']
DEC CSV: ['20251231-RTT-December-2025-full-extract.csv']
```

```
import os, glob

month_files = {
    "Apr-2025": glob.glob("/content/rtt_apr_2025/*.csv")[0],
    "May-2025": glob.glob("/content/rtt_may_2025/*.csv")[0],
    "Jun-2025": glob.glob("/content/rtt_jun_2025/*.csv")[0],
    "Jul-2025": glob.glob("/content/rtt_jul_2025/*.csv")[0],
    "Aug-2025": glob.glob("/content/rtt_aug_2025/*.csv")[0],
    "Sep-2025": glob.glob("/content/rtt_sep_2025/*.csv")[0],
    "Oct-2025": glob.glob("/content/rtt_oct_2025/*.csv")[0],
    "Nov-2025": glob.glob("/content/rtt_nov_2025/*.csv")[0],
    "Dec-2025": glob.glob("/content/rtt_dec_2025/*.csv")[0],
}

for m,p in month_files.items():
    print(m, "->", p, "->", os.path.exists(p))
```

```
Apr-2025 -> /content/rtt_apr_2025/20250430-RTT-April-2025-full-extract-revised.csv -> True
May-2025 -> /content/rtt_may_2025/20250531-RTT-May-2025-full-extract-revised.csv -> True
Jun-2025 -> /content/rtt_jun_2025/20250630-RTT-June-2025-full-extract-revised.csv -> True
Jul-2025 -> /content/rtt_jul_2025/20250731-RTT-July-2025-full-extract-revised.csv -> True
Aug-2025 -> /content/rtt_aug_2025/20250831-RTT-August-2025-full-extract-revised.csv -> True
Sep-2025 -> /content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv -> True
Oct-2025 -> /content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv -> True
Nov-2025 -> /content/rtt_nov_2025/20251130-RTT-November-2025-full-extract.csv -> True
Dec-2025 -> /content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv -> True
```

```
import pandas as pd
import numpy as np
import re

def specialty_hotspots(csv_path, month_label):
    df = pd.read_csv(csv_path)
    df_inc = df[df["RTT Part Description"] == "Incomplete Pathways"].copy()

    wait_cols = [c for c in df_inc.columns if c.startswith("Gt ")]

    for c in wait_cols:
        df_inc[c] = (df_inc[c].astype(str).str.replace(",", "", regex=False).str.strip())
        df_inc[c] = pd.to_numeric(df_inc[c], errors="coerce").fillna(0)

    def start_week(colname):
        m = re.search(r"Gt\s+(\d+)\s+To", colname)
        if m: return int(m.group(1))
        m2 = re.search(r"Gt\s+(\d+)\s+Weeks", colname)
        return int(m2.group(1)) if m2 else None
```

```
import os, glob

folders = [
    ("Apr-2025", "/content/rtt_apr_2025"),
    ("May-2025", "/content/rtt_may_2025"),
    ("Jun-2025", "/content/rtt_jun_2025"),
    ("Jul-2025", "/content/rtt_jul_2025"),
    ("Aug-2025", "/content/rtt_aug_2025"),
    ("Sep-2025", "/content/rtt_sep_2025"),
    ("Oct-2025", "/content/rtt_oct_2025"),
    ("Nov-2025", "/content/rtt_nov_2025"),
    ("Dec-2025", "/content/rtt_dec_2025"),
]

month_files = {}

for label, folder in folders:
    exists = os.path.exists(folder)
    csvs = glob.glob(folder + "/*.csv") if exists else []
    print(label, "folder exists:", exists, "| csv count:", len(csvs))
    if len(csvs) > 0:
        print("   first csv:", os.path.basename(csvs[0]))
        month_files[label] = csvs[0]
    else:
        # show what's inside the folder (helps when csv is nested)
        if exists:
            print("   folder contents:", os.listdir(folder)[:20])

print("\nDONE. month_files keys:", list(month_files.keys()))
```

```
Apr-2025 folder exists: True | csv count: 1
   first csv: 20250430-RTT-April-2025-full-extract-revised.csv
May-2025 folder exists: True | csv count: 1
   first csv: 20250531-RTT-May-2025-full-extract-revised.csv
Jun-2025 folder exists: True | csv count: 1
   first csv: 20250630-RTT-June-2025-full-extract-revised.csv
Jul-2025 folder exists: True | csv count: 1
   first csv: 20250731-RTT-July-2025-full-extract-revised.csv
Aug-2025 folder exists: True | csv count: 1
   first csv: 20250831-RTT-August-2025-full-extract-revised.csv
Sep-2025 folder exists: True | csv count: 1
   first csv: 20250930-RTT-September-2025-full-extract-revised.csv
Oct-2025 folder exists: True | csv count: 1
   first csv: 20251031-RTT-October-2025-full-extract.csv
Nov-2025 folder exists: True | csv count: 1
   first csv: 20251130-RTT-November-2025-full-extract.csv
Dec-2025 folder exists: True | csv count: 1
   first csv: 20251231-RTT-December-2025-full-extract.csv

DONE. month_files keys: ['Apr-2025', 'May-2025', 'Jun-2025', 'Jul-2025', 'Aug-2025', 'Sep-2025', 'Oct-2025', 'Nov-2025', 'De
```

```
glob.glob(folder + "/**/*.csv", recursive=True)
```

```
['/content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv']
```

```
import glob, os
```

```
folders = {
    "Apr-2025": "/content/rtt_apr_2025",
    "May-2025": "/content/rtt_may_2025",
    "Jun-2025": "/content/rtt_jun_2025",
    "Jul-2025": "/content/rtt_jul_2025",
    "Aug-2025": "/content/rtt_aug_2025",
    "Sep-2025": "/content/rtt_sep_2025",
    "Oct-2025": "/content/rtt_oct_2025",
    "Nov-2025": "/content/rtt_nov_2025",
    "Dec-2025": "/content/rtt_dec_2025",
}

month_files = {}

for m, folder in folders.items():
    csvs = glob.glob(folder + "/**/*.csv", recursive=True)
    print(m, "->", csvs[:1])
    if csvs:
        month_files[m] = csvs[0]

print("\nTotal months found:", len(month_files))
```

```
Apr-2025 -> ['/content/rtt_apr_2025/20250430-RTT-April-2025-full-extract-revised.csv']
May-2025 -> ['/content/rtt_may_2025/20250531-RTT-May-2025-full-extract-revised.csv']
Jun-2025 -> ['/content/rtt_jun_2025/20250630-RTT-June-2025-full-extract-revised.csv']
Jul-2025 -> ['/content/rtt_jul_2025/20250731-RTT-July-2025-full-extract-revised.csv']
Aug-2025 -> ['/content/rtt_aug_2025/20250831-RTT-August-2025-full-extract-revised.csv']
Sep-2025 -> ['/content/rtt_sep_2025/20250930-RTT-September-2025-full-extract-revised.csv']
Oct-2025 -> ['/content/rtt_oct_2025/20251031-RTT-October-2025-full-extract.csv']
Nov-2025 -> ['/content/rtt_nov_2025/20251130-RTT-November-2025-full-extract.csv']
Dec-2025 -> ['/content/rtt_dec_2025/20251231-RTT-December-2025-full-extract.csv']

Total months found: 9
```

```
for m,p in month_files.items():
    print(m, "exists:", os.path.exists(p))
```

```
Apr-2025 exists: True
May-2025 exists: True
Jun-2025 exists: True
Jul-2025 exists: True
Aug-2025 exists: True
Sep-2025 exists: True
Oct-2025 exists: True
Nov-2025 exists: True
Dec-2025 exists: True
```

```
import pandas as pd
import numpy as np
import re

def specialty_hotspots(csv_path, month_label):
    df = pd.read_csv(csv_path)

    # Focus on backlog snapshot
    df_inc = df[df["RTT Part Description"] == "Incomplete Pathways"].copy()

    # waiting-time bin columns
    wait_cols = [c for c in df_inc.columns if c.startswith("Gt ")]

    # Clean numeric
    for c in wait_cols:
        df_inc[c] = (df_inc[c].astype(str).str.replace(",", "", regex=False).str.strip())
        df_inc[c] = pd.to_numeric(df_inc[c], errors="coerce").fillna(0)

    # Map bins to starting week number
    def start_week(colname):
        m = re.search(r"Gt\s+(\d+)\s+To", colname)
        if m:
            return int(m.group(1))
        m2 = re.search(r"Gt\s+(\d+)\s+Weeks", colname)
        return int(m2.group(1)) if m2 else None

    week_map = {c: start_week(c) for c in wait_cols}
    wait_cols = [c for c in wait_cols if week_map[c] is not None]

    # Group by specialty
    group_col = "Treatment Function Name"
    g = df_inc.groupby(group_col)[wait_cols].sum()

    total = g.sum(axis=1)
```

```
    over18_cols = [c for c in wait_cols if week_map[c] > 18]
    over52_cols = [c for c in wait_cols if week_map[c] > 52]

    over18 = g[over18_cols].sum(axis=1) if over18_cols else 0
    over52 = g[over52_cols].sum(axis=1) if over52_cols else 0

    out = pd.DataFrame({
        "Month": month_label,
        "Specialty": total.index,
        "Total_Waiting": total.values,
        "Over_18": np.array(over18),
        "Over_52": np.array(over52),
    })

    out["Pct_Over_18"] = np.where(out["Total_Waiting"] > 0, out["Over_18"] / out["Total_Waiting"] * 100, 0)
    out["Pct_Over_52"] = np.where(out["Total_Waiting"] > 0, out["Over_52"] / out["Total_Waiting"] * 100, 0)

    # remove tiny specialties (helps signal)
    out = out[out["Total_Waiting"] >= 1000].copy()

    return out

all_specialty = []
for month, path in month_files.items():
    all_specialty.append(specialty_hotspots(path, month))

specialty_df = pd.concat(all_specialty, ignore_index=True)

specialty_df.shape, specialty_df.head()
```

```
((216, 7),
       Month                       Specialty  Total_Waiting   Over_18  Over_52  \
 0  Apr-2025              Cardiology Service       410447.0  153698.0   6642.0
 1  Apr-2025  Cardiothoracic Surgery Service        10118.0    3153.0    155.0
 2  Apr-2025             Dermatology Service       402039.0  152163.0   7755.0
 3  Apr-2025     Ear Nose and Throat Service       640611.0  302829.0  22970.0
 4  Apr-2025        Elderly Medicine Service        23622.0    3978.0     42.0

    Pct_Over_18  Pct_Over_52
 0    37.446491     1.618236
 1    31.162285     1.531923
 2    37.847821     1.928917
 3    47.271901     3.585639
 4    16.840234     0.177800  )
```

```
overall_hotspots = (
    specialty_df.groupby("Specialty")[["Total_Waiting","Over_52","Over_18"]].sum()
    .sort_values("Over_52", ascending=False)
    .head(15)
)

overall_hotspots
```

Specialty	Total_Waiting	Over_52	Over_18
Total	65979675.0	1436454.0	24357348.0
Trauma and Orthopaedic Service	7726766.0	224501.0	3061759.0
Ear Nose and Throat Service	5586721.0	192209.0	2604699.0
Gynaecology Service	5186656.0	140477.0	2142959.0
Oral Surgery Service	2903817.0	102312.0	1358450.0
Other - Surgical Services	4328582.0	93741.0	1539723.0
General Surgery Service	3533919.0	85440.0	1321071.0
Urology Service	3542270.0	79678.0	1319419.0
Other - Medical Services	5705583.0	76567.0	1827591.0
Dermatology Service	3815147.0	76297.0	1355518.0
Other - Paediatric Services	2934819.0	60716.0	1070684.0
Ophthalmology Service	5350363.0	59815.0	1497964.0
Cardiology Service	3538201.0	53037.0	1273526.0
Gastroenterology Service	3415430.0	49509.0	1153326.0
Neurology Service	2010568.0	47801.0	838238.0

```
pivot_52 = specialty_df.pivot_table(
    index="Specialty", columns="Month", values="Pct_Over_52", aggfunc="mean"
).dropna(thresh=6)

pivot_52["Change_Apr_to_Dec"] = pivot_52["Dec-2025"] - pivot_52["Apr-2025"]

worsened = pivot_52.sort_values("Change_Apr_to_Dec", ascending=False).head(10)
improved = pivot_52.sort_values("Change_Apr_to_Dec").head(10)

worsened, improved
```

```
(Month                              Apr-2025  Aug-2025  Dec-2025  Jul-2025  \
 Specialty
 Trauma and Orthopaedic Service     2.672216  3.070804  2.881205  2.987543
 Plastic Surgery Service            3.064007  3.756627  3.249972  3.636929
 Elderly Medicine Service           0.177800  0.119364  0.347859  0.095208
 Other - Mental Health Services     0.119570  0.072411  0.143318  0.113422
 General Internal Medicine Service  0.619626  0.937059  0.632605  0.794218
 Gynaecology Service                2.660906  2.874783  2.519196  2.724349
 Gastroenterology Service           1.487787  1.637178  1.241929  1.470561
 Cardiothoracic Surgery Service     1.531923  1.609269  1.267036  1.754008
 Dermatology Service                1.928917  2.100734  1.641280  2.244132
 Other - Medical Services           1.332809  1.506721  1.042942  1.465358

 Month                              Jun-2025  May-2025  Nov-2025  Oct-2025  \
 Specialty
 Trauma and Orthopaedic Service     2.855446  2.886806  2.876407  2.951621
 Plastic Surgery Service            3.511472  3.506520  3.341183  3.698791
 Elderly Medicine Service           0.172237  0.190291  0.339480  0.176009
 Other - Mental Health Services     0.121065  0.164069  0.071505  0.068966
 General Internal Medicine Service  0.598898  0.678458  0.620743  0.729001
 Gynaecology Service                2.677627  2.843191  2.571987  2.755436
 Gastroenterology Service           1.421569  1.543282  1.320129  1.420019
 Cardiothoracic Surgery Service     1.573663  1.801621  1.465675  1.469011
 Dermatology Service                2.207824  2.262279  1.752491  1.916447
 Other - Medical Services           1.468604  1.526273  1.106127  1.289444

 Month                              Sep-2025  Change_Apr_to_Dec
 Specialty
 Trauma and Orthopaedic Service     2.963055           0.208989
 Plastic Surgery Service            3.579078           0.185965
 Elderly Medicine Service           0.095863           0.170058
 Other - Mental Health Services     0.037369           0.023748
 General Internal Medicine Service  0.939484           0.012979
 Gynaecology Service                2.743874          -0.141710
 Gastroenterology Service           1.504028          -0.245859
 Cardiothoracic Surgery Service     1.553566          -0.264888
 Dermatology Service                1.916396          -0.287637
 Other - Medical Services           1.322696          -0.289867  ,
 Month                          Apr-2025  Aug-2025  Dec-2025  Jul-2025  \
 Specialty
 Ear Nose and Throat Service    3.585639  3.817860  2.568380  3.785767
 Neurology Service              2.509209  2.728969  1.777329  2.685543
 Other - Paediatric Services    2.222456  2.279818  1.539259  2.351809
 Urology Service                2.555679  2.287954  1.934424  2.253853
```

```
Ophthalmology Service         1.346169  1.232174  0.788677  1.220904
Other - Other Services        1.483321  1.366917  0.951266  1.502441
Oral Surgery Service          3.634838  3.699939  3.179316  3.556098
Respiratory Medicine Service  1.124401  1.194916  0.694625  1.056217
Cardiology Service            1.618236  1.604330  1.200271  1.539855
Rheumatology Service          1.075199  0.975877  0.688200  1.003964

Month                         Jun-2025  May-2025  Nov-2025  Oct-2025  \
Specialty
Ear Nose and Throat Service   3.854142  4.010423  2.695922  3.147580
Neurology Service             2.603940  2.698495  1.926912  2.153222
Other - Paediatric Services   2.403053  2.468092  1.590437  1.759924
Urology Service               2.491290  2.661425  1.926207  2.015190
Ophthalmology Service         1.287518  1.428284  0.806909  0.928873
```

```
risk_df = pivot_52.copy()

risk_df["Risk_Category"] = "Stable"

risk_df.loc[risk_df["Change_Apr_to_Dec"] > 0.15, "Risk_Category"] = "Worsening Risk"
risk_df.loc[risk_df["Change_Apr_to_Dec"] < -0.4, "Risk_Category"] = "Improving"

risk_df.sort_values("Change_Apr_to_Dec", ascending=False).head(15)
```

Month Specialty	Apr-2025	Aug-2025	Dec-2025	Jul-2025	Jun-2025	May-2025	Nov-2025	Oct-2025	Sep-2025	Change_Apr_to_Dec	Ri
Trauma and Orthopaedic Service	2.672216	3.070804	2.881205	2.987543	2.855446	2.886806	2.876407	2.951621	2.963055	0.208989	W
Plastic Surgery Service	3.064007	3.756627	3.249972	3.636929	3.511472	3.506520	3.341183	3.698791	3.579078	0.185965	W
Elderly Medicine Service	0.177800	0.119364	0.347859	0.095208	0.172237	0.190291	0.339480	0.176009	0.095863	0.170058	W
Other - Mental Health Services	0.119570	0.072411	0.143318	0.113422	0.121065	0.164069	0.071505	0.068966	0.037369	0.023748	
General Internal Medicine Service	0.619626	0.937059	0.632605	0.794218	0.598898	0.678458	0.620743	0.729001	0.939484	0.012979	
Gynaecology Service	2.660906	2.874783	2.519196	2.724349	2.677627	2.843191	2.571987	2.755436	2.743874	-0.141710	
Gastroenterology Service	1.487787	1.637178	1.241929	1.470561	1.421569	1.543282	1.320129	1.420019	1.504028	-0.245859	
Cardiothoracic Surgery Service	1.531923	1.609269	1.267036	1.754008	1.573663	1.801621	1.465675	1.469011	1.553566	-0.264888	
Dermatology Service	1.928917	2.100734	1.641280	2.244132	2.207824	2.262279	1.752491	1.916447	1.916396	-0.287637	
Other - Medical Services	1.332809	1.506721	1.042942	1.465358	1.468604	1.526273	1.106127	1.289444	1.322696	-0.289867	
General Surgery Service	2.544040	2.585983	2.214647	2.467458	2.452140	2.610907	2.183997	2.265045	2.420294	-0.329392	
Neurosurgical Service	2.078795	2.308074	1.744372	2.235642	2.224483	2.494130	1.720990	1.737488	2.092701	-0.334423	
Other - Surgical Services	2.136296	2.416237	1.789650	2.297171	2.225216	2.374103	1.891919	2.102884	2.259679	-0.346647	
Total	2.220630	2.356517	1.848122	2.302160	2.300536	2.414352	1.899348	2.077122	2.161626	-0.372507	
Rheumatology Service	1.075199	0.975877	0.688200	1.003964	0.965330	1.151885	0.690481	0.779637	0.850062	-0.387000	

```
import pandas as pd

summary_2025 = pd.DataFrame({
    "Month": [
        "Apr-2025","May-2025","Jun-2025","Jul-2025",
        "Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"
    ],
    "Total_Waiting": [
        14842268,14775904,14799808,14667584,
        14690158,14658134,14672974,14381434,14462538
    ],
    "Pct_Over_18": [
        38.487029,36.950321,36.562326,36.724344,
```

```
        37.380959,36.509217,36.244704,36.305156,37.020183
    ],
    "Pct_Over_52": [
        2.216427,2.410059,2.295638,2.295579,
        2.350336,2.154858,2.069260,1.891258,1.839829
    ]
})
```

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
scaled = scaler.fit_transform(summary_2025[["Pct_Over_18","Pct_Over_52"]])

summary_2025["Pressure_Score"] = scaled.mean(axis=1)

summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Pressure_Score
0	Apr-2025	14842268	38.487029	2.216427	0.830216
1	May-2025	14775904	36.950321	2.410059	0.657340
2	Jun-2025	14799808	36.562326	2.295638	0.470495
3	Jul-2025	14667584	36.724344	2.295579	0.506571
4	Aug-2025	14690158	37.380959	2.350336	0.700998
5	Sep-2025	14658134	36.509217	2.154858	0.335212
6	Oct-2025	14672974	36.244704	2.069260	0.201174
7	Nov-2025	14381434	36.305156	1.891258	0.058575
8	Dec-2025	14462538	37.020183	1.839829	0.172919

```
import matplotlib.pyplot as plt
import numpy as np
import os

os.makedirs("images_project2", exist_ok=True)

month_order = ["Apr-2025","May-2025","Jun-2025","Jul-2025",
               "Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"]

summary_2025["Month"] = pd.Categorical(summary_2025["Month"], categories=month_order, ordered=True)
summary_2025 = summary_2025.sort_values("Month").reset_index(drop=True)

summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Pressure_Score
0	Apr-2025	14842268	38.487029	2.216427	0.830216
1	May-2025	14775904	36.950321	2.410059	0.657340
2	Jun-2025	14799808	36.562326	2.295638	0.470495
3	Jul-2025	14667584	36.724344	2.295579	0.506571
4	Aug-2025	14690158	37.380959	2.350336	0.700998
5	Sep-2025	14658134	36.509217	2.154858	0.335212
6	Oct-2025	14672974	36.244704	2.069260	0.201174
7	Nov-2025	14381434	36.305156	1.891258	0.058575
8	Dec-2025	14462538	37.020183	1.839829	0.172919

```
import matplotlib.pyplot as plt
import numpy as np
import os

os.makedirs("images_project2", exist_ok=True)

month_order = ["Apr-2025","May-2025","Jun-2025","Jul-2025",
               "Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"]

summary_2025["Month"] = pd.Categorical(summary_2025["Month"], categories=month_order, ordered=True)
summary_2025 = summary_2025.sort_values("Month").reset_index(drop=True)
```

```
summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Pressure_Score
0	Apr-2025	14842268	38.487029	2.216427	0.830216
1	May-2025	14775904	36.950321	2.410059	0.657340
2	Jun-2025	14799808	36.562326	2.295638	0.470495
3	Jul-2025	14667584	36.724344	2.295579	0.506571
4	Aug-2025	14690158	37.380959	2.350336	0.700998
5	Sep-2025	14658134	36.509217	2.154858	0.335212
6	Oct-2025	14672974	36.244704	2.069260	0.201174
7	Nov-2025	14381434	36.305156	1.891258	0.058575
8	Dec-2025	14462538	37.020183	1.839829	0.172919

```
import pandas as pd

summary_2025 = pd.DataFrame({
    "Month": ["Apr-2025","May-2025","Jun-2025","Jul-2025","Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"],
    "Total_Waiting": [
        14842268.0,
        14775904.0,
        14799808.0,
        14667584.0,
        14690158.0,
        14658134.0,
        14672974.0,
        14381434.0,
        14462538.0
    ],
    "Pct_Over_18": [
        38.487029,
        36.950321,
        36.562326,
        36.724344,
        37.380959,
        36.509217,
        36.244704,
        36.305156,
        37.020183
    ],
    "Pct_Over_52": [
        2.216427,
        2.410059,
        2.295638,
        2.295579,
        2.350336,
        2.154858,
        2.069260,
        1.891258,
        1.839829
    ]
})

summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52
0	Apr-2025	14842268.0	38.487029	2.216427
1	May-2025	14775904.0	36.950321	2.410059
2	Jun-2025	14799808.0	36.562326	2.295638
3	Jul-2025	14667584.0	36.724344	2.295579
4	Aug-2025	14690158.0	37.380959	2.350336
5	Sep-2025	14658134.0	36.509217	2.154858
6	Oct-2025	14672974.0	36.244704	2.069260
7	Nov-2025	14381434.0	36.305156	1.891258
8	Dec-2025	14462538.0	37.020183	1.839829

```
summary_2025["Monthly_Change"] = summary_2025["Total_Waiting"].diff()
summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change
0	Apr-2025	14842268.0	38.487029	2.216427	NaN
1	May-2025	14775904.0	36.950321	2.410059	-66364.0
2	Jun-2025	14799808.0	36.562326	2.295638	23904.0
3	Jul-2025	14667584.0	36.724344	2.295579	-132224.0
4	Aug-2025	14690158.0	37.380959	2.350336	22574.0
5	Sep-2025	14658134.0	36.509217	2.154858	-32024.0
6	Oct-2025	14672974.0	36.244704	2.069260	14840.0
7	Nov-2025	14381434.0	36.305156	1.891258	-291540.0
8	Dec-2025	14462538.0	37.020183	1.839829	81104.0

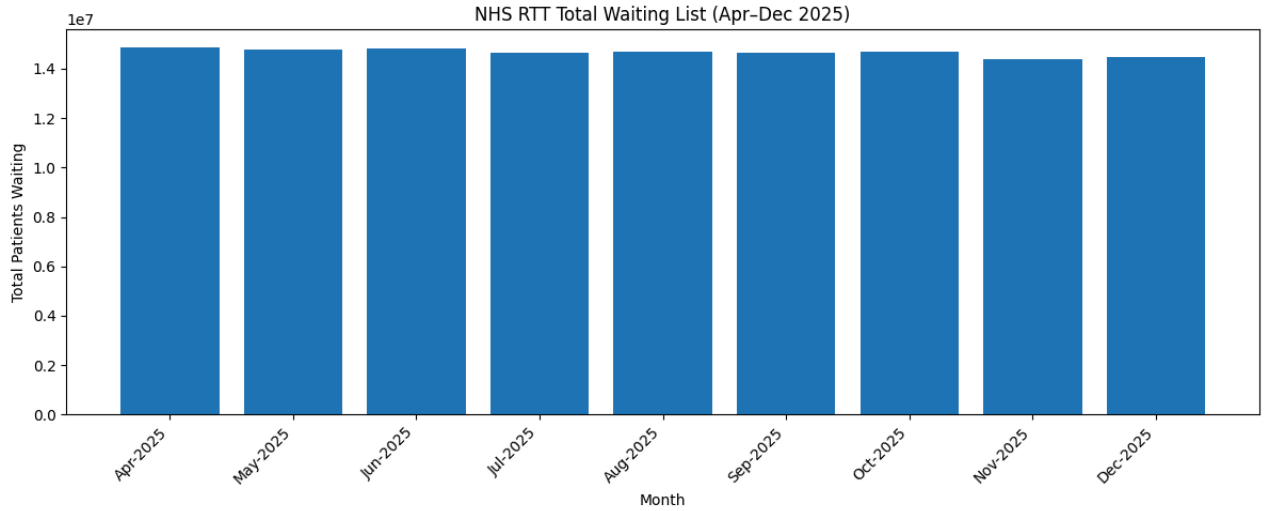
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

summary_2025["Pressure_Score"] = scaler.fit_transform(
    summary_2025[["Pct_Over_18","Pct_Over_52"]].mean(axis=1).values.reshape(-1,1)
)

summary_2025
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Monthly_Change	Pressure_Score
0	Apr-2025	14842268.0	38.487029	2.216427	NaN	1.000000
1	May-2025	14775904.0	36.950321	2.410059	-66364.0	0.464279
2	Jun-2025	14799808.0	36.562326	2.295638	23904.0	0.263877
3	Jul-2025	14667584.0	36.724344	2.295579	-132224.0	0.328478
4	Aug-2025	14690158.0	37.380959	2.350336	22574.0	0.612228
5	Sep-2025	14658134.0	36.509217	2.154858	-32024.0	0.186539
6	Oct-2025	14672974.0	36.244704	2.069260	14840.0	0.046888
7	Nov-2025	14381434.0	36.305156	1.891258	-291540.0	0.000000
8	Dec-2025	14462538.0	37.020183	1.839829	81104.0	0.264694

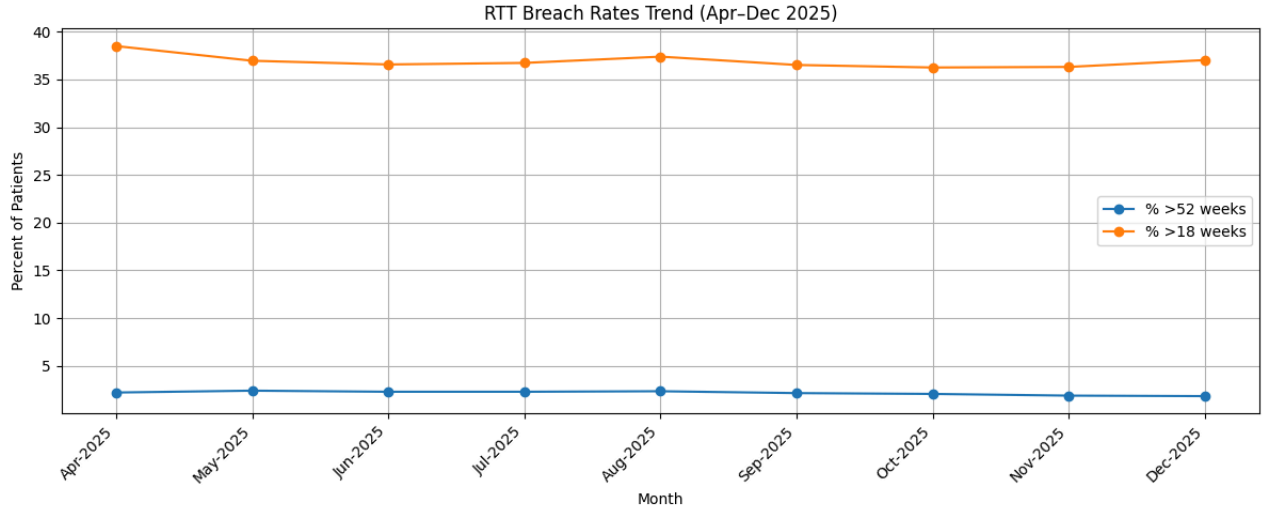
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os

# Make an output folder for images
os.makedirs("images_project2", exist_ok=True)

# Ensure Month order is correct
month_order = ["Apr-2025","May-2025","Jun-2025","Jul-2025","Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"]
summary_2025["Month"] = pd.Categorical(summary_2025["Month"], categories=month_order, ordered=True)
summary_2025 = summary_2025.sort_values("Month").reset_index(drop=True)

summary_2025[["Month","Total_Waiting","Pct_Over_18","Pct_Over_52","Pressure_Score"]]
```

	Month	Total_Waiting	Pct_Over_18	Pct_Over_52	Pressure_Score
0	Apr-2025	14842268.0	38.487029	2.216427	1.000000
1	May-2025	14775904.0	36.950321	2.410059	0.464279
2	Jun-2025	14799808.0	36.562326	2.295638	0.263877
3	Jul-2025	14667584.0	36.724344	2.295579	0.328478
4	Aug-2025	14690158.0	37.380959	2.350336	0.612228
5	Sep-2025	14658134.0	36.509217	2.154858	0.186539
6	Oct-2025	14672974.0	36.244704	2.069260	0.046888
7	Nov-2025	14381434.0	36.305156	1.891258	0.000000
8	Dec-2025	14462538.0	37.020183	1.839829	0.264694

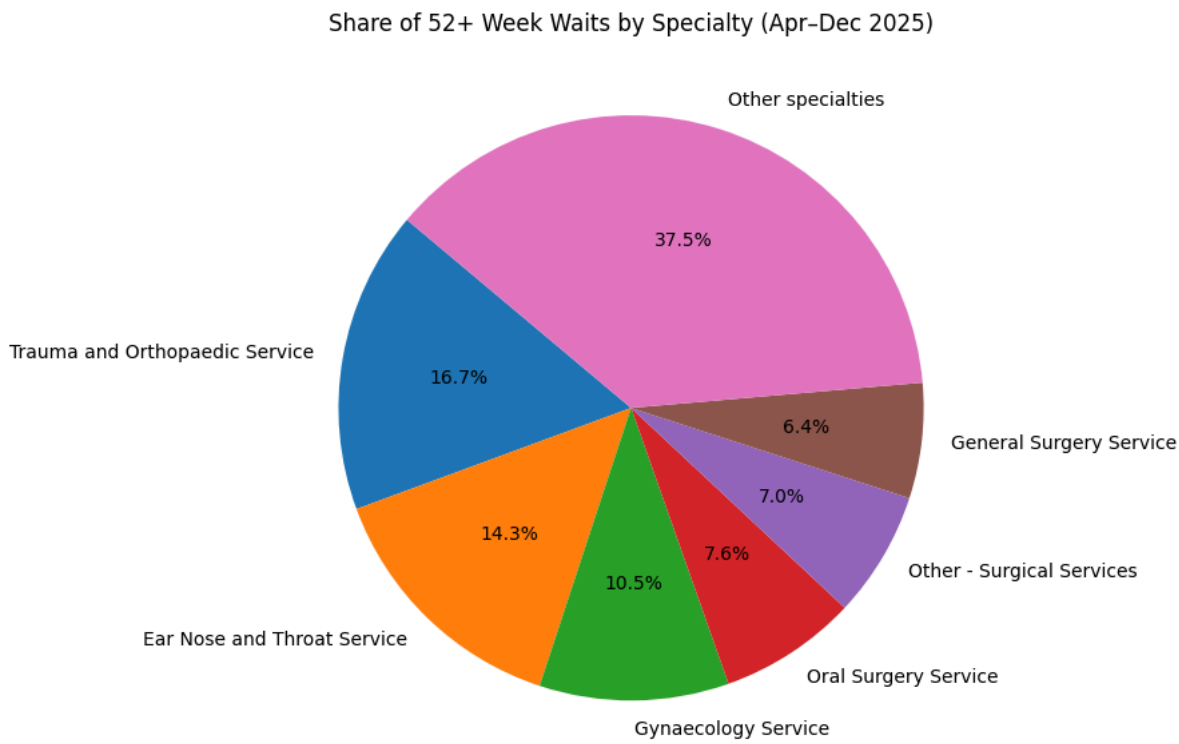
```
plt.figure(figsize=(12,5))
plt.bar(summary_2025["Month"].astype(str), summary_2025["Total_Waiting"])
plt.xticks(rotation=45, ha="right")
plt.title("NHS RTT Total Waiting List (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Total Patients Waiting")
plt.tight_layout()

plt.savefig("images_project2/01_total_waiting_by_month.png", dpi=200)
plt.show()
```




```
plt.figure(figsize=(12,5))
plt.plot(summary_2025["Month"].astype(str), summary_2025["Pct_Over_52"], marker="o", label="% >52 weeks")
plt.plot(summary_2025["Month"].astype(str), summary_2025["Pct_Over_18"], marker="o", label="% >18 weeks")
plt.xticks(rotation=45, ha="right")
plt.title("RTT Breach Rates Trend (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Percent of Patients")
plt.grid(True)
plt.legend()
plt.tight_layout()

plt.savefig("images_project2/02_breach_rates_trend.png", dpi=200)
plt.show()
```

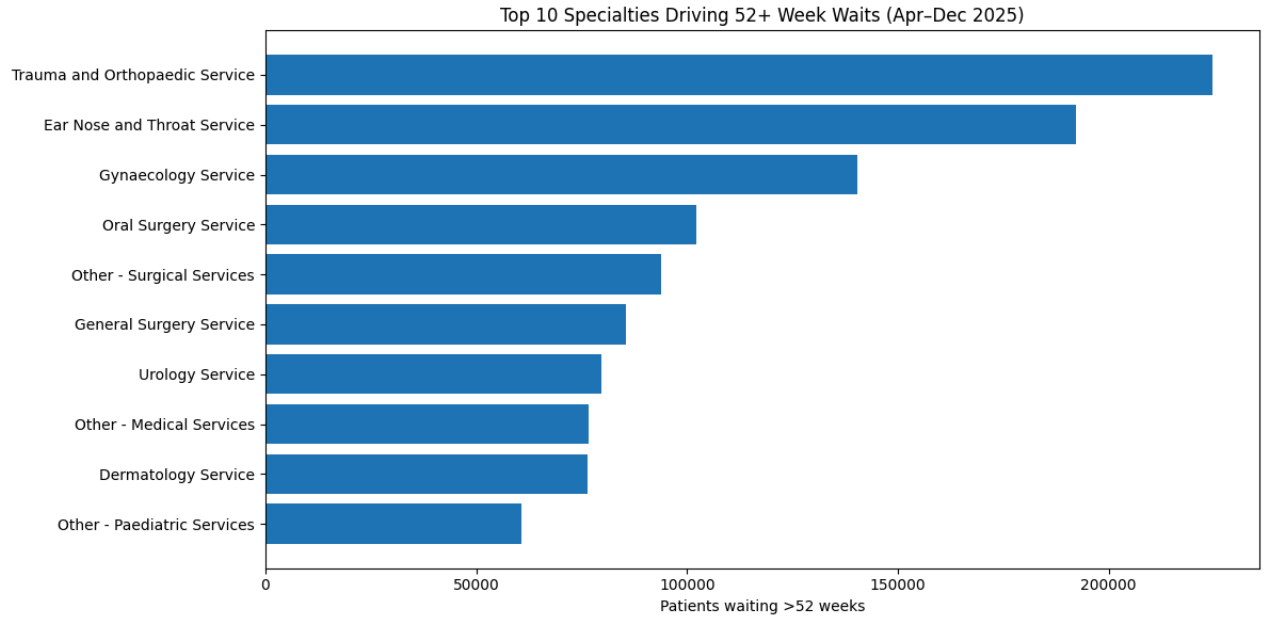



```
# Take top 6 specialties by Over_52, combine rest into "Other"
top = overall_hotspots.drop(index="Total", errors="ignore").head(6).copy()
other_value = overall_hotspots.drop(index="Total", errors="ignore")["Over_52"].sum() - top["Over_52"].sum()

labels = list(top.index) + ["Other specialties"]
values = list(top["Over_52"].values) + [other_value]

plt.figure(figsize=(10,6))
plt.pie(values, labels=labels, autopct="%1.1f%%", startangle=140)
plt.title("Share of 52+ Week Waits by Specialty (Apr–Dec 2025)")
plt.tight_layout()

plt.savefig("images_project2/03_over52_share_pie.png", dpi=200)
plt.show()
```




```
top10 = overall_hotspots.drop(index="Total", errors="ignore").head(10).copy().iloc[::-1]  # reverse for nice order

plt.figure(figsize=(12,6))
plt.barh(top10.index, top10["Over_52"])
plt.title("Top 10 Specialties Driving 52+ Week Waits (Apr–Dec 2025)")
plt.xlabel("Patients waiting >52 weeks")
plt.tight_layout()

plt.savefig("images_project2/04_top10_over52_barh.png", dpi=200)
```

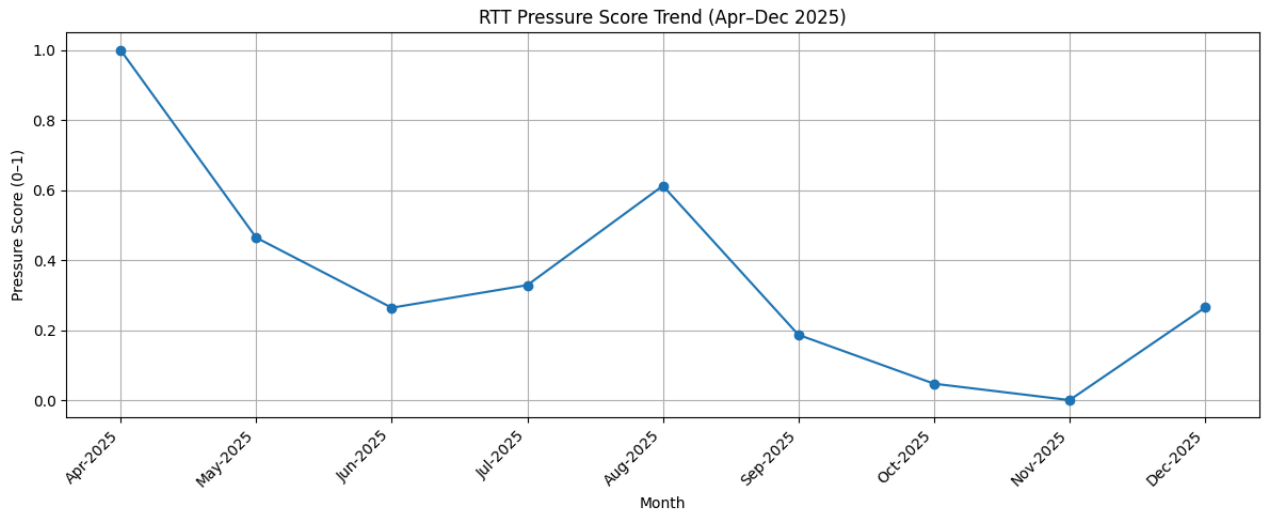
```
plt.show()
```




```
import shutil
shutil.make_archive("images_project2_pack", "zip", "images_project2")
print("Created: images_project2_pack.zip")
```

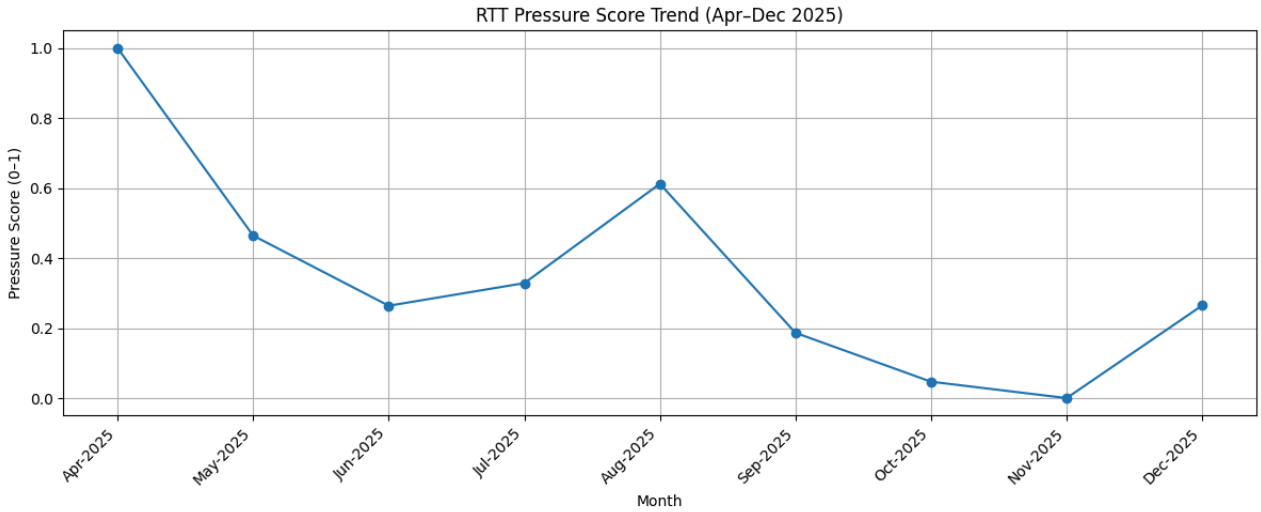
```
Created: images_project2_pack.zip
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,5))
plt.plot(summary_2025["Month"].astype(str), summary_2025["Pressure_Score"], marker="o")
plt.xticks(rotation=45, ha="right")
plt.title("RTT Pressure Score Trend (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Pressure Score (0–1)")
plt.grid(True)
plt.tight_layout()

plt.savefig("images_project2/05_pressure_score_trend.png", dpi=200)
plt.show()
```

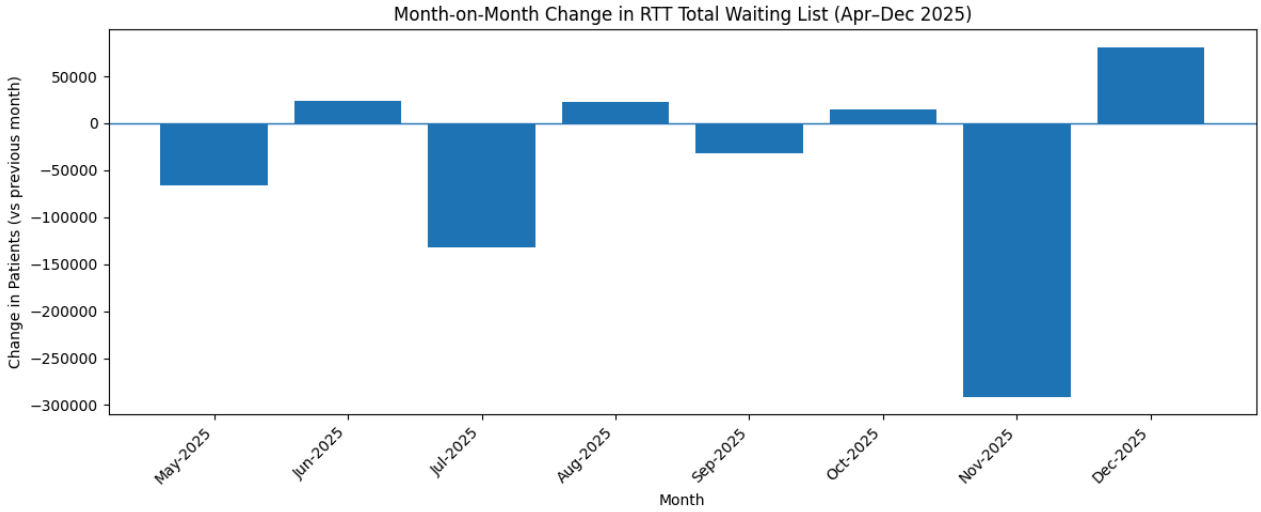



```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,5))
plt.plot(summary_2025["Month"].astype(str), summary_2025["Pressure_Score"], marker="o")
plt.xticks(rotation=45, ha="right")
```

```
plt.title("RTT Pressure Score Trend (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Pressure Score (0–1)")
plt.grid(True)
plt.tight_layout()

plt.savefig("images_project2/05_pressure_score_trend.png", dpi=200)
plt.show()
```




```
plt.figure(figsize=(12,5))
plt.bar(summary_2025["Month"].astype(str), summary_2025["Monthly_Change"])
plt.axhline(0, linewidth=1)
plt.xticks(rotation=45, ha="right")
plt.title("Month-on-Month Change in RTT Total Waiting List (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Change in Patients (vs previous month)")
plt.tight_layout()

plt.savefig("images_project2/06_monthly_change_bar.png", dpi=200)
plt.show()
```




```
import os
os.makedirs("images_project2", exist_ok=True)
```

```
import matplotlib.pyplot as plt
import numpy as np

# -----------------------------------
# CHART 5 – Pressure Score Trend
# -----------------------------------
plt.figure(figsize=(12,5))
```

```
plt.plot(summary_2025["Month"].astype(str),
         summary_2025["Pressure_Score"],
         marker="o")
plt.xticks(rotation=45, ha="right")
plt.title("RTT Pressure Score Trend (Apr–Dec 2025)")
plt.xlabel("Month")
plt.ylabel("Pressure Score (0–1)")
plt.grid(True)
plt.tight_layout()
plt.savefig("images_project2/05_pressure_score_trend.png", dpi=300)
plt.close()


# ----------------------------------
# CHART 6 – Monthly Change Bar
# ----------------------------------
plt.figure(figsize=(12,5))
plt.bar(summary_2025["Month"].astype(str),
        summary_2025["Monthly_Change"])
plt.axhline(0)
plt.xticks(rotation=45, ha="right")
plt.title("Month-on-Month Change in RTT Waiting List")
plt.xlabel("Month")
plt.ylabel("Change in Patients")
plt.tight_layout()
plt.savefig("images_project2/06_monthly_change_bar.png", dpi=300)
plt.close()


# ----------------------------------
# CHART 7 – Specialty Risk Heatmap
# ----------------------------------
pivot_52 = specialty_df.pivot_table(
    index="Specialty",
    columns="Month",
    values="Pct_Over_52",
    aggfunc="mean"
)

top_specialties = pivot_52.mean(axis=1).sort_values(ascending=False).head(12).index
heat = pivot_52.loc[top_specialties]

month_order = ["Apr-2025","May-2025","Jun-2025","Jul-2025",
               "Aug-2025","Sep-2025","Oct-2025","Nov-2025","Dec-2025"]

heat = heat[month_order]

plt.figure(figsize=(12,6))
plt.imshow(heat.values, aspect="auto")
plt.colorbar(label="% >52 weeks")
plt.xticks(range(len(heat.columns)), heat.columns, rotation=45, ha="right")
plt.yticks(range(len(heat.index)), heat.index)
plt.title("Specialty Risk Heatmap: % Waiting >52 Weeks")
plt.tight_layout()
plt.savefig("images_project2/07_specialty_risk_heatmap.png", dpi=300)
plt.close()

print("✅ All 3 charts saved to images_project2 folder.")
```

```
✅ All 3 charts saved to images_project2 folder.
```

Double-click (or enter) to edit

Executive Summary (Apr–Dec 2025)

This project converts NHS RTT waiting list data into an operational pressure monitoring prototype. Across Apr–Dec 2025, national waiting list volume remains persistently high, with sustained breaches over 18 weeks and a long-wait cohort over 52 weeks. A composite Pressure Score is introduced to support consistent monitoring and escalation decisions. Specialty hotspot patterns indicate where backlog pressure is structurally concentrated. Outputs include an escalation-tier model, governance boundaries, and implementation pathway for responsible decision-support.

NHS RTT Operational Pressure Framework (Apr–Dec 2025)

Project Overview

This project builds a decision-support analytics prototype using NHS England RTT (Referral to Treatment) waiting times data. Using monthly national extracts from April to December 2025, the analysis tracks backlog pressure and long-wait breaches, and proposes an operational escalation model to support capacity planning and governance-led decision-making.

Scope: Non-clinical, aggregate reporting only (no patient-level data).
Output: A structured monitoring framework + recommendations for NHS operational pressure management.

Data Source & Methodology

Data Source

Monthly NHS England RTT incomplete pathway national extracts (April–December 2025).

Methodological Approach

• Aggregation of incomplete pathways • Breach threshold analysis (18-week and 52-week) • Month-on-month variation • Normalised composite Pressure Score construction

Analytical Scope

Aggregate national and specialty-level analysis only. No patient-level or regional microdata.

Operational Escalation Model (Prototype)

Escalation Tiers Based on Pressure Score

To operationalise the framework, a tiered escalation model is proposed:

Tier 1 – Stable Pressure
Pressure Score < 0.40
Routine monitoring. No immediate intervention required.

Tier 2 – Moderate Pressure
Pressure Score 0.40 – 0.70
Operational review recommended.
Specialty-level capacity adjustments considered.

Tier 3 – High Pressure
Pressure Score > 0.70
Escalation to regional management level.
Short-term surge planning or elective reprioritisation may be required.

Example: 2025 Insights

- April 2025 recorded the highest pressure score.
- November 2025 showed significant improvement.
- Long-wait (>52 week) patients strongly influenced escalation.

This demonstrates how quantitative thresholds can support structured decision-making while retaining managerial oversight.

Decision-Support Recommendations for NHS Operational Pressure Management (2025)

1. National Pressure Observations (April–December 2025)

- Waiting list levels remain structurally elevated across the period.
- Percentage of patients waiting over 18 weeks consistently exceeds 36%.
- Long waits (>52 weeks) remain present across all months, though slight improvement is visible toward December.
- Seasonal variation is observable but does not materially reduce backlog pressure.

This indicates systemic strain rather than short-term fluctuation.

---

2. High-Pressure Specialties Identified

Analysis of specialty-level backlog reveals sustained pressure in:

- Trauma and Orthopaedics
- Ear Nose and Throat
- Oral Surgery

- Gynaecology
- Plastic Surgery

These specialties demonstrate consistently elevated long-wait proportions.

---

3. Operational Risk Indicators

Three key risk signals were identified:

1. Persistent 18+ week breaches across all months
2. Specialty-level clustering of long waits
3. Slow structural improvement despite minor month-to-month fluctuations

This suggests operational capacity constraints rather than administrative variation.

---

4. Decision-Support Recommendations

Based on the 2025 findings, the following data-driven interventions are recommended:

A. Targeted Capacity Reallocation

Redistribute elective capacity toward persistently high-pressure specialties.

B. 52-Week Early Warning Trigger

Introduce automated escalation alerts when >52 week cases exceed defined thresholds per specialty.

C. Pressure Score Monitoring Dashboard

Deploy a national Pressure Score tracking system to detect sustained stress patterns earlier.

D. Seasonal Surge Planning

Model seasonal impact (Aug–Dec) to proactively adjust workforce planning.

---

5. Governance & Ethical Boundaries

This framework does NOT:

- Replace clinical prioritisation
- Override patient risk-based scheduling
- Automate treatment decisions
- Infer patient-level attributes

The role of this system is strictly decision-support for operational management.

NHS National Waiting List Pressure Monitoring Framework (Prototype)

Framework Overview

This project demonstrates how routine RTT data can be transformed into a structured pressure monitoring framework that supports:

- National backlog visibility
- Specialty-level stress detection
- Early warning of long-wait escalation
- Responsible operational decision-support

The framework is built on three core indicators:

1. Total Waiting List Volume
2. Percentage Waiting Over 18 Weeks
3. Percentage Waiting Over 52 Weeks

These indicators are combined into a Pressure Score to detect systemic strain.

Monitoring Model Structure

Level 1 – National Trend Monitoring
Track month-on-month total waiting list and breach percentages.

Level 2 – Specialty Pressure Ranking
Identify consistently high-pressure specialties.

Level 3 – Escalation Trigger Threshold

Define trigger points where >52 week backlog exceeds operational tolerance.

Responsible Use Case

This framework does not automate clinical prioritisation.

It provides strategic visibility for:

- Capacity reallocation decisions
- Workforce surge planning
- Elective pathway redesign
- Targeted backlog reduction initiatives