

Some Cars from the gtcars Dataset

Five Cars are shown here

Everything but the cost				
Make and Model		Performance		
mfr	model	hp	trq	msrp
Ford	GT	647	550	\$447,000.00
Ferrari	458 Speciale	597	398	\$291,744.00
Ferrari	458 Spider	562	398	\$263,553.00
Ferrari	458 Italia	562	398	\$233,509.00
Ferrari	488 GTB	661	561	\$245,400.00

Cars are all 2015 models.

Horsepower and Torque values are estimates.

```
import polars as pl
from great_tables import GT, md, html
from great_tables.data import islands

islands_mini = pl.from_pandas(islands).sort("size", descending=True).head(10)

print(
    GT(islands_mini)
    .tab_header(title="Large Landmasses of the World", subtitle="The top ten largest are presented")
    # .tab_stub(rowname_col="name")
    .tab_source_note(source_note="Source: The World Almanac and Book of Facts, 1975, page 400")
    .tab_source_note(
        # source_note=md("Reference: McNeil, D. R. (1977) *Interactive Data Analysis*. Wiley")
        source_note=html("Reference: McNeil, D. R. (1977) *Interactive Data Analysis*. Wiley")
    )
    .tab_stubhead(label="landmass")
    .fmt_image(columns="size")
    .as_latex()
)
```



```
\begin{table}
\caption*{
{\large Large Landmasses of the World} \\
{\small The top ten largest are presented}
}
```

Large Landmasses of the World
The top ten largest are presented

name	size
Africa	11,506
Antarctica	5,500
Asia	16,988
Australia	2,968
Axel Heiberg	16
Baffin	184
Banks	23
Borneo	280
Britain	84
Celebes	73

Source: The World Almanac and Book of Facts, 1975, page 406.

Reference: McNeil, D. R. (1977) *Interactive Data Analysis*. Wiley.

New York Air Quality Measurements
Daily measurements in New York City (May 1-10, 1973)

Ozone	Solar_R	Wind	Temp	Month	Day	Year
41.0	190.0	7.4	67	5	1	1973
36.0	118.0	8.0	72	5	2	1973
12.0	149.0	12.6	74	5	3	1973
18.0	313.0	11.5	62	5	4	1973
nan	nan	14.3	56	5	5	1973
28.0	nan	14.9	66	5	6	1973
23.0	299.0	8.6	65	5	7	1973
19.0	99.0	13.8	59	5	8	1973
8.0	19.0	20.1	61	5	9	1973
nan	194.0	8.6	69	5	10	1973

Physical Constants Having a Molar Basis

name	value
Molar Planck Constant	3.990×10^{-10}
Electron Molar Mass	5.486×10^{-7}
Molar Volume of Silicon	1.206×10^{-5}
Muon Molar Mass	1.134×10^{-4}
Molar Mass Constant	1.000×10^{-3}
Proton Molar Mass	1.007×10^{-3}
Neutron Molar Mass	1.009×10^{-3}
Tau Molar Mass	1.908×10^{-3}
Deuteron Molar Mass	2.014×10^{-3}
Helion Molar Mass	3.015×10^{-3}
Triton Molar Mass	3.016×10^{-3}
Alpha Particle Molar Mass	4.002×10^{-3}
Molar Mass of Carbon-12	1.200×10^{-2}
Molar Volume of Ideal Gas (273.15 K, 101.325 kpa)	2.241×10^{-2}
Molar Volume of Ideal Gas (273.15 K, 100 kpa)	2.271×10^{-2}
Molar Gas Constant	8.314

num	date	time	currency
111 B	Thursday, January 15, 2015	[13:35]	49.95
2.2 KiB	Sunday, February 15, 2015	[14:40]	17.95
32.5 KiB	Sunday, March 15, 2015	[15:45]	\$_{1.39}
434 KiB	Wednesday, April 15, 2015	[16:50]	\$_{65,100.00}
5.3 MiB	Friday, May 15, 2015	[17:55]	\$_{1,325.81}

```

\fontsize{12.0pt}{14.4pt}\selectfont

\begin{tabular*}{\linewidth}{@{\extracolsep{\fill}}lr}
\toprule
name & size \\
\midrule
Asia & 16988 \\
Africa & 11506 \\
North America & 9390 \\
South America & 6795 \\
Antarctica & 5500 \\
Europe & 3745 \\
Australia & 2968 \\
Greenland & 840 \\
New Guinea & 306 \\
Borneo & 280 \\
\bottomrule
\end{tabular*}

\begin{minipage}{\linewidth}
Source: The World Almanac and Book of Facts, 1975, page 406.\\
Reference: McNeil, D. R. (1977) *Interactive Data Analysis*. Wiley.\$\\
\end{minipage}

\end{table}

/opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/great_tables/_formats.py
    warn("fmt_image() is not currently implemented in LaTeX output.")
/opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/great_tables/_utils_rendering.py
    warnings.warn(msg)

from great_tables import GT, html
from great_tables.data import airquality

airquality_mini = airquality.head(10).assign(Year=1973)

print(
    GT(airquality_mini)
    .tab_header(
        title="New York Air Quality Measurements",
        subtitle="Daily measurements in New York City (May 1-10, 1973)",
    )
    .tab_spinner(label="Time", columns=["Year", "Month", "Day"])
)

```

```

.tab_spinner(label="Measurement", columns=["Ozone", "Solar_R", "Wind", "Temp"])
.cols_move_to_start(columns=["Year", "Month", "Day"])
.cols_label(
    Ozone=html("Ozone,<br>ppbV"),
    Solar_R=html("Solar R.,<br>cal/m<sup>2</sup>"),
    Wind=html("Wind,<br>mph"),
    Temp=html("Temp,<br>&deg;F"),
)
.as_latex()
)

\begin{table}
\caption*{
{\large New York Air Quality Measurements} \\
{\small Daily measurements in New York City (May 1-10, 1973)}
}

\fontsize{12.0pt}{14.4pt}\selectfont

\begin{tabular*}{\linewidth}{@{\extracolsep{\fill}}rrrrrrr}
\toprule
\multicolumn{3}{c}{Time} & \multicolumn{4}{c}{Measurement} \\
\cmidrule(lr){1-3} \cmidrule(lr){4-7}
Year & Month & Day & Ozone,<br>ppbV & Solar R.,<br>cal/m<sup>2</sup> & Wind,<br>mph & Temp,<br>
\midrule\addlinespace[2.5pt]
1973 & 5 & 1 & 41.0 & 190.0 & 7.4 & 67 \\
1973 & 5 & 2 & 36.0 & 118.0 & 8.0 & 72 \\
1973 & 5 & 3 & 12.0 & 149.0 & 12.6 & 74 \\
1973 & 5 & 4 & 18.0 & 313.0 & 11.5 & 62 \\
1973 & 5 & 5 & nan & nan & 14.3 & 56 \\
1973 & 5 & 6 & 28.0 & nan & 14.9 & 66 \\
1973 & 5 & 7 & 23.0 & 299.0 & 8.6 & 65 \\
1973 & 5 & 8 & 19.0 & 99.0 & 13.8 & 59 \\
1973 & 5 & 9 & 8.0 & 19.0 & 20.1 & 61 \\
1973 & 5 & 10 & nan & 194.0 & 8.6 & 69 \\
\bottomrule
\end{tabular*}

\end{table}

/opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/great_tables/_utils_rend
warnings.warn(msg)

```

```

from great_tables import GT
from great_tables.data import countrypops
import polars as pl
import polars.selectors as cs

# Get vectors of 2-letter country codes for each region of Oceania
oceania = {
    "Australasia": ["AU", "NZ"],
    "Melanesia": ["NC", "PG", "SB", "VU"],
    "Micronesia": ["FM", "GU", "KI", "MH", "MP", "NR", "PW"],
    "Polynesia": ["PF", "WS", "TO", "TV"],
}

# Create a dictionary mapping country to region (e.g. AU -> Australasia)
country_to_region = {
    country: region for region, countries in oceania.items() for country in countries
}

wide_pops = (
    pl.from_pandas(countrypops)
    .filter(
        pl.col("country_code_2").is_in(list(country_to_region))
        & pl.col("year").is_in([2000, 2010, 2020])
    )
    .with_columns(pl.col("country_code_2").replace(country_to_region).alias("region"))
    .pivot(index=["country_name", "region"], on="year", values="population")
    .sort("2020", descending=True)
)

print(
    GT(wide_pops)
    .tab_header(title="Populations of Oceania's Countries in 2000, 2010, and 2020")
    .tab_spinner(label="Total Population", columns=cs.all())
    #.tab_stub(rowname_col="country_name", groupname_col="region")
    .fmt_integer() # example fails because of this method
    .as_latex()
)

```

towny example

```

from great_tables import GT, html
from great_tables.data import sza

```

```

import polars as pl
import polars.selectors as cs

sza_pivot = (
    pl.from_pandas(sza)
    .filter((pl.col("latitude") == "20") & (pl.col("tst") <= "1200"))
    .select(pl.col("*").exclude("latitude"))
    .drop_nulls()
    .pivot(values="sza", index="month", on="tst", sort_columns=True)
)

```

```

print(
    GT(
        sza_pivot,
        #rowname_col="month"
    )
    .data_color(
        domain=[90, 0],
        palette=["rebeccapurple", "white", "orange"],
        na_color="white",
    )
    .tab_header(
        title="Solar Zenith Angles from 05:30 to 12:00",
        subtitle=html("Average monthly values at latitude of 20&deg;N."),
    )
    .sub_missing(missing_text="")
    .as_latex()
)

```

```

\begin{table}
\caption*{
{\large Solar Zenith Angles from 05:30 to 12:00} \\
{\small Average monthly values at latitude of 20\&deg;N.}
}

\fontsize{12.0pt}{14.4pt}\selectfont

\begin{tabular*}{\linewidth}{@{\extracolsep{\fill}}lrrrrrrrrrrrrr}
\toprule
month & 0530 & 0600 & 0630 & 0700 & 0730 & 0800 & 0830 & 0900 & 0930 & 1000 & 1030 & 1100 & \\
\midrule
jan & None & None & None & 84.9 & 78.7 & 72.7 & 66.1 & 61.5 & 56.5 & 52.1 & 48.3 & 45.5 & 43

```

```

feb & None & None & 88.9 & 82.5 & 75.8 & 69.6 & 63.3 & 57.7 & 52.2 & 47.4 & 43.1 & 40.0 & 37
mar & None & None & 85.7 & 78.8 & 72.0 & 65.2 & 58.6 & 52.3 & 46.2 & 40.5 & 35.5 & 31.4 & 28
apr & None & 88.5 & 81.5 & 74.4 & 67.4 & 60.3 & 53.4 & 46.5 & 39.7 & 33.2 & 26.9 & 21.3 & 17
may & None & 85.0 & 78.2 & 71.2 & 64.3 & 57.2 & 50.2 & 43.2 & 36.1 & 29.1 & 26.1 & 15.2 & 8.9
jun & 89.2 & 82.7 & 76.0 & 69.3 & 62.5 & 55.7 & 48.8 & 41.9 & 35.0 & 28.1 & 21.1 & 14.2 & 7.3
jul & 88.8 & 82.3 & 75.7 & 69.1 & 62.3 & 55.5 & 48.7 & 41.8 & 35.0 & 28.1 & 21.2 & 14.3 & 7.1
aug & None & 83.8 & 77.1 & 70.2 & 63.3 & 56.4 & 49.4 & 42.4 & 35.4 & 28.3 & 21.3 & 14.3 & 7.3
sep & None & 87.2 & 80.2 & 73.2 & 66.1 & 59.1 & 52.1 & 45.1 & 38.1 & 31.3 & 24.7 & 18.6 & 13
oct & None & None & 84.1 & 77.1 & 70.2 & 63.3 & 56.5 & 49.9 & 43.5 & 37.5 & 32.0 & 27.4 & 24
nov & None & None & 87.8 & 81.3 & 74.5 & 68.3 & 61.8 & 56.0 & 50.2 & 45.3 & 40.7 & 37.4 & 35
dec & None & None & None & 84.3 & 78.0 & 71.8 & 66.1 & 60.5 & 55.6 & 50.9 & 47.2 & 44.2 & 42
\bottomrule
\end{tabular*}

\end{table}

\begin{table}[ht]
\caption*{\large 2023 Mean **Carbon Intensity** (gCO2eq/kWh) and **Power Consumption** Breakdown (\%)}

\begin{tabular*}{\textwidth}{@{\extracolsep{\fill}}lccccccccccc}
\toprule
Zone & CO2 Intensity & Hydro & Nuclear & Wind & Solar & Geothermal & Biomass & Gas & Coal & Others
\midrule
Sweden & 26 & 39.1\% & 26.8\% & 27.7\% & 0.1\% & 0.0\% & 0.4\% & 0.4\% & 0.8\% & 0.0\% & 4.6\%
Iceland & 28 & 69.4\% & 0.0\% & 0.0\% & 0.0\% & 30.6\% & 0.0\% & 0.0\% & 0.0\% & 0.0\% & 0.0\%
Quebec & 35 & 90.1\% & 2.1\% & 4.4\% & 0.0\% & 0.0\% & 1.9\% & 1.4\% & 0.0\% & 0.0\% & 0.0\%
France & 46 & 12.3\% & 65.4\% & 10.3\% & 1.8\% & 0.0\% & 1.0\% & 7.1\% & 0.3\% & 0.3\% & 0.1\%
Ontario & 104 & 23.3\% & 49.4\% & 8.7\% & 0.1\% & 0.0\% & 0.2\% & 18.1\% & 0.0\% & 0.0\% & 0.0\%
New Zealand & 106 & 60.5\% & 0.0\% & 7.7\% & 0.1\% & 19.0\% & 0.0\% & 6.8\% & 3.7\% & 0.0\% & 0.0\%
Finland & 107 & 20.2\% & 36.5\% & 24.1\% & 0.1\% & 0.0\% & 6.2\% & 3.0\% & 8.1\% & 0.0\% & 1.0\%
\bottomrule
\end{tabular*}
\end{table}

```

```

South Australia & 132 & 0.7\% & 0.0\% & 42.6\% & 33.7\% & 0.0\% & 0.0\% & 13.3\% & 9.0\% & 0
Spain & 132 & 17.1\% & 24.2\% & 25.1\% & 8.0\% & 0.0\% & 2.0\% & 18.8\% & 1.3\% & 0.2\% & 0
Belgium & 147 & 1.3\% & 39.6\% & 25.2\% & 3.6\% & 0.0\% & 2.8\% & 19.4\% & 1.7\% & 0.1\% & 4
Tasmania & 162 & 49.0\% & 0.0\% & 22.6\% & 10.8\% & 0.0\% & 0.0\% & 1.5\% & 16.1\% & 0.0\% &
East Denmark & 184 & 6.4\% & 5.5\% & 48.4\% & 1.3\% & 0.0\% & 16.8\% & 7.7\% & 10.8\% & 1.4\%
West Denmark & 188 & 8.8\% & 2.2\% & 56.3\% & 1.6\% & 0.0\% & 7.6\% & 8.5\% & 13.0\% & 0.9\%
Great Britain & 214 & 3.8\% & 12.4\% & 35.9\% & 2.7\% & 0.0\% & 6.2\% & 35.1\% & 2.0\% & 0.0\%
Netherlands & 218 & 1.1\% & 3.9\% & 46.7\% & 10.8\% & 0.0\% & 4.6\% & 22.4\% & 8.6\% & 0.8\%
New York ISO & 275 & 23.7\% & 22.8\% & 4.9\% & 0.0\% & 0.0\% & 0.1\% & 46.9\% & 0.0\% & 0.0\%
Italy (North) & 307 & 22.7\% & 14.5\% & 3.9\% & 2.9\% & 0.2\% & 3.1\% & 38.4\% & 1.5\% & 0.2\%
California & 328 & 8.4\% & 12.7\% & 7.9\% & 12.0\% & 3.0\% & 1.8\% & 48.5\% & 2.1\% & 0.0\%
Germany & 389 & 4.4\% & 2.8\% & 39.7\% & 3.3\% & 0.0\% & 8.7\% & 14.4\% & 23.3\% & 0.6\% & 0
Ireland & 389 & 3.7\% & 0.8\% & 38.5\% & 0.2\% & 0.0\% & 2.5\% & 42.4\% & 9.7\% & 2.0\% & 0.1\%
Western Australia & 417 & 0.0\% & 0.0\% & 14.1\% & 33.8\% & 0.0\% & 0.3\% & 24.2\% & 27.1\% & 0
Texas & 432 & 0.0\% & 9.1\% & 22.3\% & 6.0\% & 0.0\% & 0.0\% & 46.1\% & 16.1\% & 0.0\% & 0.4\%
Alberta & 447 & 1.9\% & 0.0\% & 12.4\% & 1.1\% & 0.0\% & 2.5\% & 70.7\% & 7.2\% & 0.0\% & 4.1\%
Victoria & 508 & 3.9\% & 0.0\% & 17.5\% & 19.0\% & 0.0\% & 0.0\% & 0.3\% & 59.1\% & 0.0\% & 0
New South Wales & 578 & 3.2\% & 0.0\% & 9.5\% & 23.7\% & 0.0\% & 0.2\% & 0.7\% & 62.6\% & 0.0\%
Queensland & 662 & 1.9\% & 0.0\% & 3.8\% & 21.1\% & 0.0\% & 0.0\% & 7.2\% & 65.7\% & 0.2\% & 0.1\%
South Africa & 685 & 2.2\% & 4.3\% & 5.8\% & 3.8\% & 0.0\% & 0.0\% & 0.0\% & 79.9\% & 2.0\% & 0.0\%
India (North) & 693 & 9.3\% & 2.2\% & 0.1\% & 10.6\% & 0.0\% & 0.0\% & 1.8\% & 75.2\% & 0.0\% & 0
\bottomrule
\end{tabular*}

```

```
\end{table}
```

```

import polars as pl
import polars.selectors as cs
from great_tables import GT, loc, style

coffee_sales = pl.read_ndjson("../examples/_data/coffee-sales.ndjson")

sel_rev = cs.starts_with("revenue")
sel_prof = cs.starts_with("profit")

# yo

print(
    GT(coffee_sales)
        .tab_header("Sales of Coffee Equipment")
        .tab_spinner(label="Revenue", columns=sel_rev)
        .tab_spinner(label="Profit", columns=sel_prof)

```

```

.cols_label(
    revenue_dollars="Amount",
    profit_dollars="Amount",
    revenue_pct="Percent",
    profit_pct="Percent",
    monthly_sales="Monthly Sales",
    icon="",
    product="Product",
)
# formatting ----
(fmt_number(
    columns=cs.ends_with("dollars"),
    compact=True,
    pattern="${x}",
    n_sigfig=3,
)
(fmt_percent(columns=cs.ends_with("pct"), decimals=0)
# style ----
.tab_style(
    style=style.fill(color="aliceblue"),
    locations=loc.body(columns=sel_rev),
)
.tab_style(
    style=style.fill(color="papayawhip"),
    locations=loc.body(columns=sel_prof),
)
.tab_style(
    style=style.text(weight="bold"),
    locations=loc.body(rows=pl.col("product") == "Total"),
)
# .fmt_nanoplot("monthly_sales", plot_type="bar")
# .fmt_image("icon", path="docs/examples/_data/coffee-table-icons/")
.sub_missing(missing_text="")
.as_latex()
)

\begin{table}
\caption*{
{\large Sales of Coffee Equipment}
}

```

```
\fontsize{12.0pt}{14.4pt}\selectfont
```

```

\begin{tabular*}{\linewidth}{@{\extracolsep{\fill}}llrrrrc}
\toprule
& \multicolumn{2}{c}{Revenue} & \multicolumn{2}{c}{Profit} & \\
\cmidrule(lr){2-3} \cmidrule(lr){4-5}
& Product & Amount & Percent & Amount & Percent & Monthly Sales \\
\midrule\addlinespace[2.5pt]
grinder.png & Grinder & \$904K & 3\% & \$568K & 4\% & shape: (12,)
Series: '' [i64]
[
521
494
596
613
667
...
686
607
594
568
751
] \\
moka-pot.png & Moka pot & \$2.05M & 7\% & \$181K & 1\% & shape: (12,)
Series: '' [i64]
[
4726
4741
4791
5506
6156
...
6026
5304
4884
4648
6283
] \\
cold-brew.png & Cold brew & \$289K & 1\% & \$242K & 2\% & shape: (12,)
Series: '' [i64]
[
244
249
438

```

```
981
1774
...
2348
1741
896
499
244
] \\
filter.png & Filter & \$404K & 1% & $70.0K & 0% & shape: (12,)
Series: '' [i64]
[
 2067
 1809
 1836
 2123
 2252
...
 2367
 2164
 2195
 2070
 2744
] \\
drip-machine.png & Drip machine & $2.63M & 9% & $1.37M & 9% & shape: (12,)
Series: '' [i64]
[
 2137
 1623
 1971
 2097
 2580
...
 2316
 2052
 1967
 1837
 2328
] \\
aeropress.png & AeroPress & $2.60M & 9% & $1.29M & 9% & shape: (12,)
Series: '' [i64]
[
 6332
```

```
5199
6367
7024
7906
...
7797
6828
6963
6877
9270
] \\
pour-over.png & Pour over & \$846K & 3% & \$365K & 2% & shape: (12,)
Series: '' [i64]
[
  1562
  1291
  1511
  1687
  1940
  ...
  1856
  1715
  1806
  1601
  2165
] \\
french-press.png & French press & \$1.11M & 4% & \$748K & 5% & shape: (12,)
Series: '' [i64]
[
  3507
  2880
  3346
  3792
  3905
  ...
  4428
  3279
  3420
  3297
  4819
] \\
cezve.png & Cezve & \$2.51M & 9% & \$1.97M & 13% & shape: (12,)
Series: '' [i64]
```

```
[  
 12171  
 11469  
 11788  
 13630  
 15391  
 ...  
 14433  
 12985  
 12935  
 11598  
 15895  
] \\  
chemex.png & Chemex & \$3.14M & 11% & \$818K & 6% & shape: (12,)  
Series: '' [i64]  
[  
 4938  
 4167  
 5235  
 6000  
 6358  
 ...  
 6249  
 5605  
 6076  
 4980  
 7220  
] \\  
scale.png & Scale & \$3.80M & 13% & \$2.91M & 20% & shape: (12,)  
Series: '' [i64]  
[  
 1542  
 1566  
 1681  
 2028  
 2425  
 ...  
 2232  
 2036  
 2089  
 1693  
 3180  
] \\
```

```

kettle.png & Kettle & \$756K & 3% & \$618K & 4% & shape: (12,)
Series: '' [i64]
[
  1139
  1023
  1087
  1131
  1414
  ...
  1304
  1140
  1233
  1193
  1529
] \\
espresso-machine.png & Espresso Machine & \$8.41M & 29% & \$3.64M & 25% & shape: (12,)
Series: '' [i64]
[
  686
  840
  618
  598
  2148
  ...
  996
  1002
  668
  858
  2577
] \\
None & Total & \$29.4M & 100% & \$14.8M & 100% & None \\
\bottomrule
\end{tabular*}

\end{table}

/opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/great_tables/_utils_rend
warnings.warn(msg)

```