Poking around the `titanic` dataset

First in Python

```python
import requests
from pathlib import Path

def download(url, output_file):
    if not Path(output_file).exists():
        r = requests.get(url)
        with open(output_file, "wb") as f:
            f.write(r.content)
    print("Download finished")

download("https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv",
"examples/titanic.csv")
```

```
Download finished
```

```python
import pandas as pd
df = pd.read_csv("examples/titanic.csv")
df["Name"] = df["Name"].str.split(" ").str.slice(0, 3).str.join(" ")
df = df.drop(df.filter(like="Aboard", axis=1).columns, axis=1)

print(df.head(5))
```

```
   Survived  Pclass                     Name     Sex   Age     Fare
0         0       3         Mr. Owen Harris    male  22.0   7.2500
1         1       1       Mrs. John Bradley  female  38.0  71.2833
2         1       3  Miss. Laina Heikkinen  female  26.0   7.9250
3         1       1     Mrs. Jacques Heath  female  35.0  53.1000
4         0       3      Mr. William Henry    male  35.0   8.0500
```

Can we do it in Typst?

```
#let csv-to-tabledata(file, n-rows: -2) = {
  let data = csv(file)
  let headers = data.at(0)
  let rows = data.slice(1, n-rows + 1)
  tada.from-rows(rows, field-info: headers)
}
#import tada: TableData, subset, chain, filter, update-fields, agg, sort-values
#let td = chain(
  csv-to-tabledata("/examples/titanic.csv"),
  // Shorten long names
  tada.add-expressions.with(
    Name: `Name.split(" ").slice(1, 3).join(" ")`,
  ),
)
#output[
  Data loaded!
  #chain(
    td,
    subset.with(
      fields: ("Name", "Age", "Fare"), indexes: range(3)
    ),
    to-tablex
  )
]
```

Data loaded!

Name	Age	Fare
Owen Harris	22	7.25
John Bradley	38	71.2833
Laina Heikkinen	26	7.925

Make it prettier

```
#let row-fmt(index, row) = {
  let fill = none
  if index == 0 {
    fill = rgb("#8888")
  } else if calc.odd(index) {
    fill = rgb("#1ea3f288")
  }
  row.map(cell => (..cell, fill: fill))
}
#let title-fmt(name) = heading(outlined: false, name)
#td.tablex-kwargs.insert("map-rows", row-fmt)
#td.field-defaults.insert("title", title-fmt)
#to-tablex(subset(td, fields: ("Name", "Age", "Fare"), indexes: range(0, 5)))
```

Name	Age	Fare
Owen Harris	22	7.25
John Bradley	38	71.2833
Laina Heikkinen	26	7.925
Jacques Heath	35	53.1
William Henry	35	8.05

Convert types & clean data

```
#let usd = tada.display.format-usd

#let td = chain(
  td,
  tada.add-expressions.with(
    Pclass: `int(Pclass)`,
    Name: `Name.slice(0, Name.position("("))`,
    Sex: `upper(Sex.at(0))`,
    Age: `float(Age)`,
    Fare: `float(Fare)`,
  ),
  update-fields.with(
    Fare: (display: usd),
  ),
  subset.with(
    fields: ("Pclass", "Name", "Age", "Fare")
  ),
  sort-values.with(by: "Fare", descending: true),
)
#to-tablex(subset(td, indexes: range(0, 10)))
```

Pclass	Name	Age	Fare
1	John Bradley	38	$71.28
1	Jacques Heath	35	$53.10
1	Timothy J	54	$51.86
2	Nicholas	14	$30.07
3	Gosta Leonard	2	$21.08
3	Oscar W	27	$11.13
3	James Moran	27	$8.46
3	William Henry	35	$8.05
3	Laina Heikkinen	26	$7.93
3	Owen Harris	22	$7.25

Find just the passengers over 30 paying over $230

```
#to-tablex(filter(td, expression: `Age > 30 and Fare > 230`))
```

Pclass	Name	Age	Fare
1	Gustave J	35	$512.33
1	Thomas Drake	36	$512.33
1	Anna Ward	35	$512.33
1	Mark Fortune	64	$263.00
1	James	50	$247.52

See how much each class paid and their average age

```
#let fares-per-class = tada.group-by(
  td,
  by: "Pclass",
  aggs: (
    "Total Fare": `Fare.sum()`,
    "Avg Age": `int(Age.sum()/Age.len())`,
  ),
  field-info: ("Total Fare": (display: usd)),
)
#to-tablex(fares-per-class)
```

Pclass	Total Fare	Avg Age
1	$18,177.41	38
2	$3,801.84	29
3	$6,667.90	25