In [285…

```
import json
import altair as alt
from altair import expr, datum
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import requests
```

In [286…

```
import colorsys
from matplotlib.colors import to_hex, to_rgb


def scale_lightness(rgb, scale_l):
    rgbhex = False
    if "#" in rgb:
        rgb = to_rgb(rgb)
        rgbhex = True
    # convert rgb to hls
    h, l, s = colorsys.rgb_to_hls(*rgb)
    # manipulate h, l, s values and return as rgb
    c = colorsys.hls_to_rgb(h, min(1, l * scale_l), s=s)
    if rgbhex:
        c = to_hex(c)
    return c
```

In [287…

```
    LOCAL = False

if LOCAL:
    local_suffix = "_local"
else:
    local_suffix = ""
```

In [288…

```
%%capture pwd
!pwd
```

In [289…

```
uid = pwd.stdout.split("/")[-1].split("\r")[0]
eco_git_home = (
    "https://raw.githubusercontent.com/EconomicsObservatory/ECOvisualisations/main/
)
eco_git_path = eco_git_home + "articles/" + uid + "/data/"
vega_embed = requests.get(eco_git_home + "guidelines/html/vega-embed.html").text
colors = json.loads(
    requests.get(eco_git_home + "guidelines/colors/eco-colors.json").content
)
category_color = json.loads(
    requests.get(eco_git_home + "guidelines/colors/eco-category-color.json").conten
)
hue_color = json.loads(
    requests.get(eco_git_home + "guidelines/colors/eco-single-hue-color.json").cont
)
mhue_color = json.loads(
    requests.get(eco_git_home + "guidelines/colors/eco-multi-hue-color.json").conte
)
```

```
div_color = json.loads(
    requests.get(eco_git_home + "guidelines/colors/eco-diverging-color.json").conte
)
config = json.loads(
    requests.get(eco_git_home + "guidelines/charts/eco-global-config.json").content
)
height = config["height"]
width = config["width"]
uid, height, width
```

Out[289…

```
('what-are-the-options-for-reforming-the-resourcing-of-adult-social-care-in-the-u
k',
 300,
 500)
```

Fig 1

In [290…

```
df = pd.read_excel("raw/figs.xlsx", skiprows=4, usecols="B,F,I").dropna(axis=0)
df.columns = ["Year", "Total", "65+"]
df = df.round(1)
df["65+"] = df["65+"].astype(int)
```

In [291…

```
f = "fig1_expenditure"
f1 = eco_git_path + f + ".csv"
df.to_csv("data/" + f + ".csv")
f += local_suffix
open("visualisation/" + f + ".html", "w").write(
    vega_embed.replace(
        "JSON_PATH", f1.replace("/data/", "/visualisation/").replace(".csv", ".json
    )
)
if LOCAL:
    f1 = df
readme = "### " + f + '\n!["' + f + '"](visualisation/' + f + '.png "' + f + '")\n\
df.head()
```

Out[291…

	Year	Total	65+
0	2006.0	19.0	2222
1	2007.0	19.2	2224
2	2008.0	19.1	2186
3	2009.0	19.6	2202
4	2010.0	20.2	2225

In [292…

```
base = alt.Chart(f1).encode(
    tooltip=["Year:Q", "Total:Q", "65+:Q"],
    x=alt.X(
        "Year:Q",
        axis=alt.Axis(
            grid=False,
```

```
            titleAlign="right",
            titleAnchor="end",
            title="",
            titleY=-15,
            labelColor=colors["eco-gray"],
            titleColor=colors["eco-gray"],
            tickColor=colors["eco-gray"],
            domainColor=colors["eco-gray"],
            tickCount=6,
            format=".0f",
            orient="bottom",
            labelAngle=0,
        ),
    ),
)
line1 = base.mark_line(color=colors["eco-turquiose"]).encode(
    y=alt.Y(
        "Total:Q",
        sort=[],
        axis=alt.Axis(
            grid=False,
            title="Total expenditure (£B)",
            titleX=-5,
            titleY=-5,
            titleBaseline="bottom",
            titleAngle=0,
            titleAlign="left",
            labelColor=colors["eco-turquiose"],
            titleColor=colors["eco-turquiose"],
            tickColor=colors["eco-turquiose"],
            domainColor=colors["eco-turquiose"],
        ),
        scale=alt.Scale(domain=[17, 21]),
    )
)
line2 = base.mark_line(color=colors["eco-dot"]).encode(
    y=alt.Y(
        "65+:Q",
        sort=[],
        axis=alt.Axis(
            grid=False,
            title="per person, 65+ and disabled population (£)",
            titleX=5,
            titleY=-5,
            titleBaseline="bottom",
            titleAngle=0,
            titleAlign="right",
            labelColor=colors["eco-dot"],
            titleColor=colors["eco-dot"],
            tickColor=colors["eco-dot"],
            domainColor=colors["eco-dot"],
        ),
        scale=alt.Scale(domain=[1600, 2800]),
    )
)
layer1 = (
```

```
    ((line1 + line2).properties(height=300, width=400))
    .configure_view(stroke=None)
    .properties(title="")
).resolve_scale(y="independent")
layer1.save("visualisation/" + f + ".json")
layer1.save("visualisation/" + f + ".png")
open("README.md", "w").write(readme)
layer1
```

```
Fontconfig error: Cannot load default config file
```

Out[292…

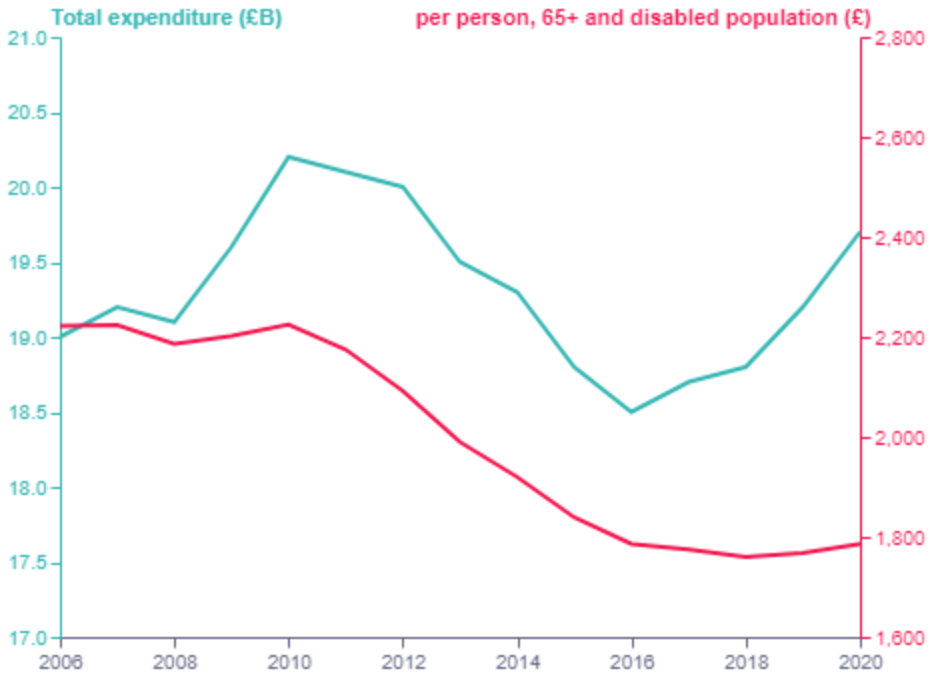


Fig 2

In [293…

```
df = pd.read_excel("raw/figs.xlsx", sheet_name="fig2")
df.columns = ["cat", "2020", "2050 (projected)"]
df = pd.DataFrame(df.set_index("cat").stack()).reset_index()
df.columns = ["cat", "timeframe", "perc"]
```

In [294…

```
f = "fig2_perc"
f2 = eco_git_path + f + ".csv"
df.to_csv("data/" + f + ".csv")
f += local_suffix
open("visualisation/" + f + ".html", "w").write(
    vega_embed.replace(
        "JSON_PATH", f2.replace("/data/", "/visualisation/").replace(".csv", ".json
    )
)
if LOCAL:
    f2 = df
readme = "### " + f + '\n!["' + f + '"](visualisation/' + f + '.png "' + f + '")\n\
df.head()
```

Out[294…

	cat	timeframe	perc
0	65+ population (% of total)	2020	18.5
1	65+ population (% of total)	2050 (projected)	24.7
2	85+ population (% of total)	2020	2.5
3	85+ population (% of total)	2050 (projected)	5.0
4	65+ population (% 15-64)	2020	29.2

In [295…

```
base = (
    alt.Chart(f2)
    .encode(
        x=alt.X(
            "cat:N",
            sort=[],
            axis=alt.Axis(
                grid=False,
                titleAlign="center",
                titleAnchor="middle",
                title="",
                titleY=-15,
                titleX=207,
                labelColor=colors["eco-gray"],
                titleColor=colors["eco-gray"],
                tickColor=colors["eco-gray"],
                domainColor=colors["eco-gray"],
                tickCount=10,
                orient="bottom",
                labelAngle=-15,
                labelOffset=30,
            ),
        ),
        y=alt.Y(
            "y:Q",
            sort=[],
            axis=alt.Axis(
                grid=False,
                title="",
                titleX=-5,
                titleY=-5,
                titleBaseline="bottom",
                titleAngle=0,
                format=".0%",
                titleAlign="left",
                labelColor=colors["eco-gray"],
                titleColor=colors["eco-gray"],
                tickColor=colors["eco-gray"],
                domainColor=colors["eco-gray"],
            ),
            scale=alt.Scale(domain=[0, 0.5]),
        ),
    )
    .transform_calculate(y="datum.perc/100")
```

```
)
bar1 = (
    base.mark_bar(width=20, xOffset=-11, color=colors["eco-turquiose"])
    .encode()
    .transform_filter("datum.timeframe==2020")
)
bar2 = (
    base.mark_bar(width=20, xOffset=11, color=colors["eco-light-blue"])
    .encode()
    .transform_filter("datum.timeframe=='2050 (projected)'")
)
text1 = bar1.mark_text(
    color=colors["eco-turquiose"], align="center", xOffset=-11, yOffset=-6
).encode(text="perc:Q")
text2 = bar2.mark_text(
    color=colors["eco-light-blue"], align="center", xOffset=11, yOffset=-6
).encode(text="perc:Q")
text1a = (
    text1.mark_text(
        color=colors["eco-turquiose"], align="right", xOffset=-25, yOffset=30
    )
    .encode(text="t:N")
    .transform_calculate(t="2020")
    .transform_filter("datum.cat=='65+ population (% 15-64)'")
)
text1b = (
    text1.mark_text(
        color=colors["eco-light-blue"], align="left", xOffset=25, yOffset=30
    )
    .encode(text="t:N")
    .transform_calculate(t="'2050 (projected)'")
    .transform_filter("datum.cat=='65+ population (% 15-64)'")
)
layer1 = (
    ((bar1 + bar2 + text1 + text2 + text1a + text1b).properties(height=300, width=4
    .configure_view(stroke=None)
    .properties(title="")
)
layer1.save("visualisation/" + f + ".json")
layer1.save("visualisation/" + f + ".png")
open("README.md", "a").write(readme)
layer1
```

```
Fontconfig error: Cannot load default config file
```

Out[295…

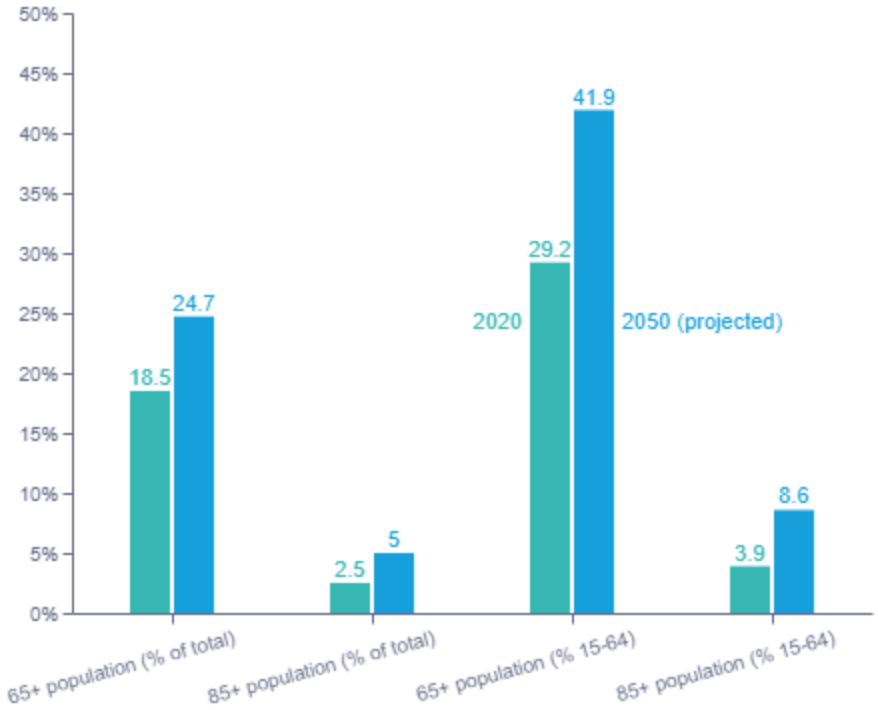


Fig 3

In [296…

```
df = pd.read_excel("raw/figs.xlsx", sheet_name="fig3_CB.2", usecols="D:BC", skiprow
df=df.T.reset_index()
df['index']='20'+df['index'].str.split('-').str[1]
df.columns=['Year','Spending','Caseload']
```

In [297…

```
f = "fig3_caseload"
f3 = eco_git_path + f + ".csv"
df.to_csv("data/" + f + ".csv")
f += local_suffix
open("visualisation/" + f + ".html", "w").write(
    vega_embed.replace(
        "JSON_PATH", f3.replace("/data/", "/visualisation/").replace(".csv", ".json
    )
)
if LOCAL:
    f3 = df
readme = "### " + f + '\n!["' + f + '"](visualisation/' + f + '.png "' + f + '")\n\
df.head()
```

Out[297…

	Year	Spending	Caseload
0	2017	1.088076	1.550471
1	2018	1.164926	1.571808
2	2019	1.193050	1.594030
3	2020	1.182044	1.617381
4	2021	1.208137	1.641975

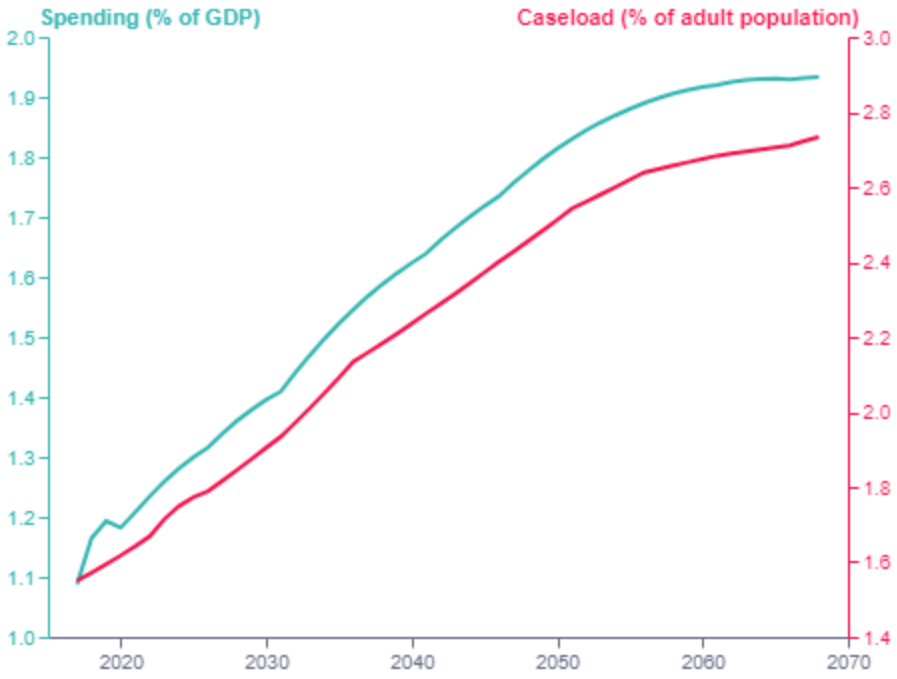
In [298…

```
base = alt.Chart(f3).encode(
    tooltip=["Year:Q", "Spending:Q", "Caseload:Q"],
    x=alt.X(
        "Year:Q",
        axis=alt.Axis(
            grid=False,
            titleAlign="right",
            titleAnchor="end",
            title="",
            titleY=-15,
            labelColor=colors["eco-gray"],
            titleColor=colors["eco-gray"],
            tickColor=colors["eco-gray"],
            domainColor=colors["eco-gray"],
            tickCount=6,
            format=".0f",
            orient="bottom",
            labelAngle=0,
        ),
    ),
)
line1 = base.mark_line(color=colors["eco-turquiose"]).encode(
    y=alt.Y(
        "Spending:Q",
        sort=[],
        axis=alt.Axis(
            grid=False,
            title="Spending (% of GDP)",
            titleX=-5,
            titleY=-5,
            titleBaseline="bottom",
            titleAngle=0,
            titleAlign="left",
            labelColor=colors["eco-turquiose"],
            titleColor=colors["eco-turquiose"],
            tickColor=colors["eco-turquiose"],
            domainColor=colors["eco-turquiose"],
        ),
        scale=alt.Scale(domain=[1, 2]),
    )
)
line2 = base.mark_line(color=colors["eco-dot"]).encode(
    y=alt.Y(
        "Caseload:Q",
```

```
        sort=[],
        axis=alt.Axis(
            grid=False,
            title="Caseload (% of adult population)",
            titleX=5,
            titleY=-5,
            titleBaseline="bottom",
            titleAngle=0,
            titleAlign="right",
            labelColor=colors["eco-dot"],
            titleColor=colors["eco-dot"],
            tickColor=colors["eco-dot"],
            domainColor=colors["eco-dot"],
        ),
        scale=alt.Scale(domain=[1.5, 3]),
    )
)
layer1 = (
    ((line1 + line2).properties(height=300, width=400))
    .configure_view(stroke=None)
    .properties(title="")
).resolve_scale(y="independent")
layer1.save("visualisation/" + f + ".json")
layer1.save("visualisation/" + f + ".png")
open("README.md", "a").write(readme)
layer1
```

```
Fontconfig error: Cannot load default config file
```

Out[298…



Metadata

In [299…

```
meta=open("README.md", "r").read()
```

Figs

In [300…

```
figs=[i.split('\n')[0] for i in meta.split('### ')[1:]]
config=json.loads(open('config.json','r').read())
config['charts']=figs
open('config.json','w').write(json.dumps(config))
```

Out[300… 584

In [ ]: