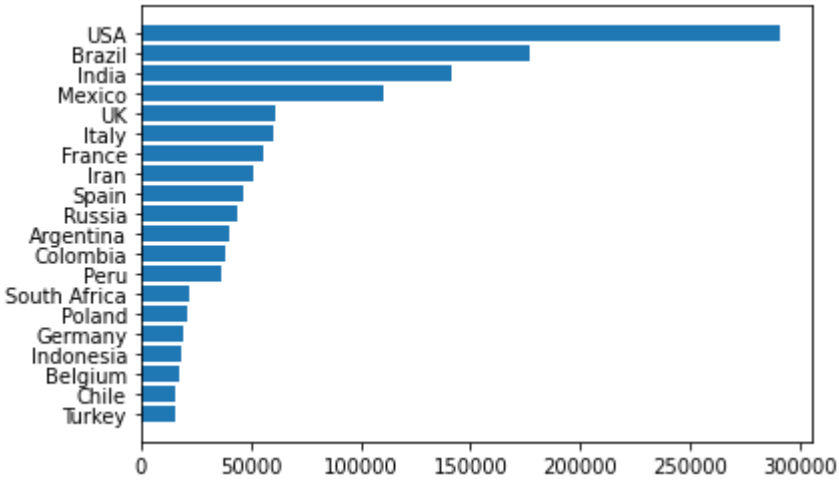
In [67]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline
death_df = pd.read_csv("death.csv")
average_df = pd.read_csv("avg_deaths.csv")
```

In [4]:
```
tot = death_df["Total_Deaths"]
cont = death_df["Country_Other"]
plt.plot(cont, tot);
```

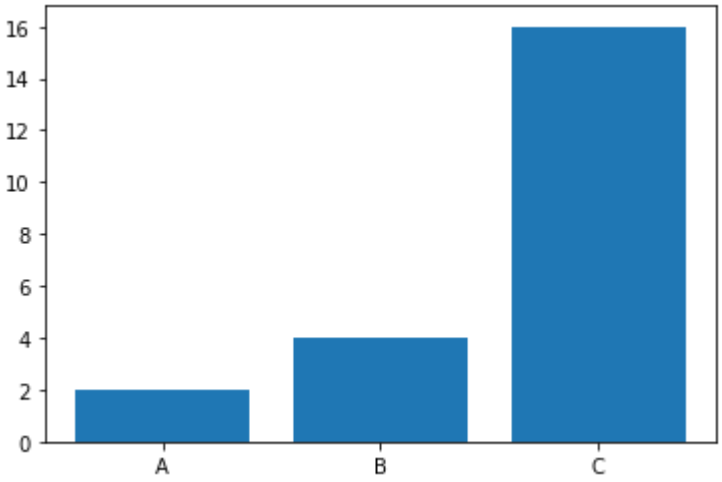
Out[4]:

	Country_Other	Total_Deaths
0	Turkey	15103
1	Chile	15663
2	Belgium	17386
3	Indonesia	18000
4	Germany	19539
5	Poland	20592
6	South Africa	22249
7	Peru	36324
8	Colombia	37995
9	Argentina	39888
10	Russia	44159
11	Spain	46646
12	Iran	50917
13	France	55521
14	Italy	60606
15	UK	61434
16	Mexico	110074
17	India	141005
18	Brazil	177388
19	USA	290798

In [6]:
```
tot = death_df["Total_Deaths"]
cont = death_df["Country_Other"]
plt.barh(cont, tot);
```

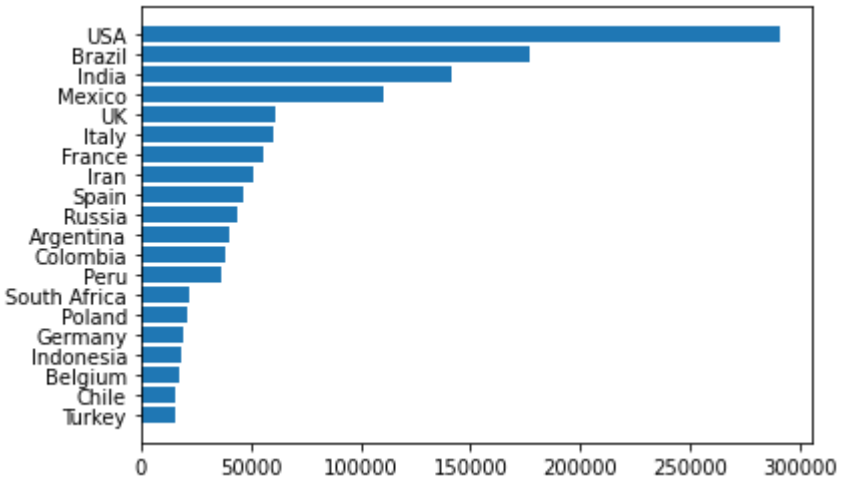


In [18]:
```
fig, ax = plt.subplots()
ax.bar(['A', 'B', 'C'],
       [2, 4, 16]);
```



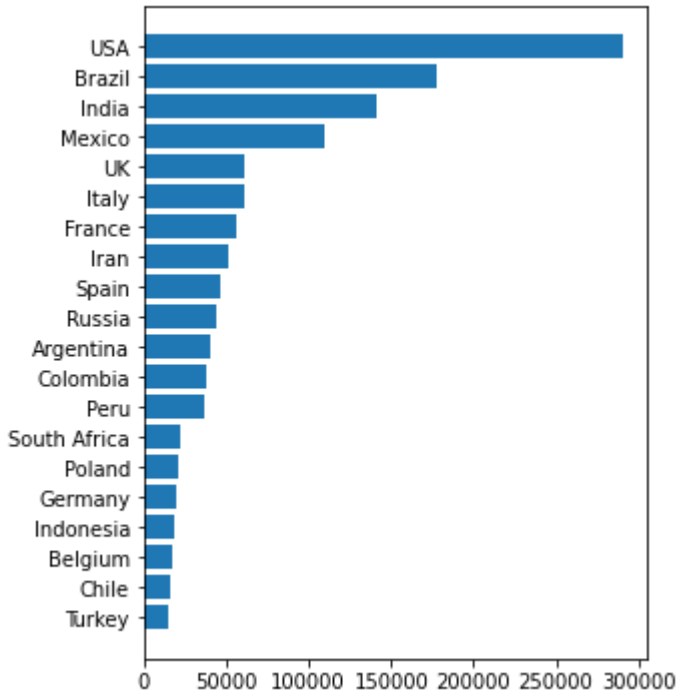
In [39]:
```
tot = death_df["Total_Deaths"]
cont = death_df["Country_Other"]
a = list(cont)
b = list(tot)
fig, ax = plt.subplots()
ax.barh(cont, tot)
```

Out[39]: <BarContainer object of 20 artists>

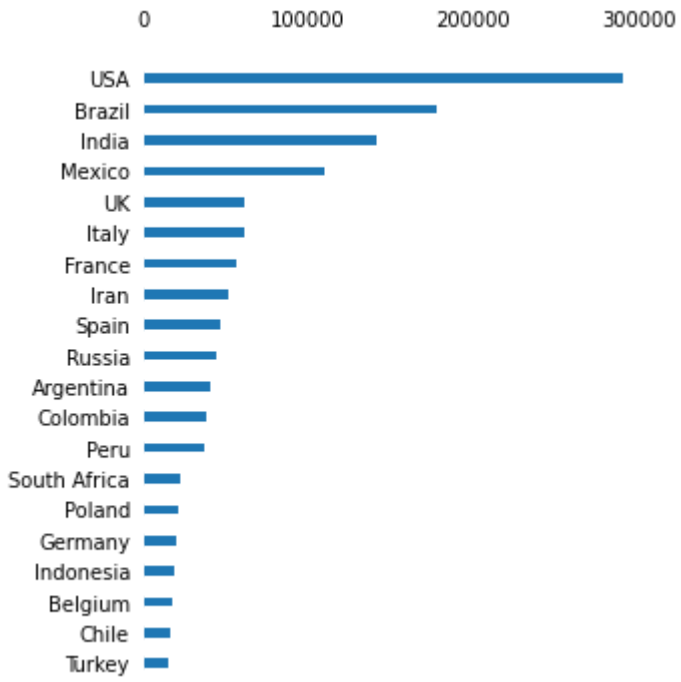


In [41]:
```
tot = death_df["Total_Deaths"]
cont = death_df["Country_Other"]
a = list(cont)
b = list(tot)
fig, ax = plt.subplots(figsize=(4.5, 6))
ax.barh(cont, tot)
```

Out[41]: <BarContainer object of 20 artists>

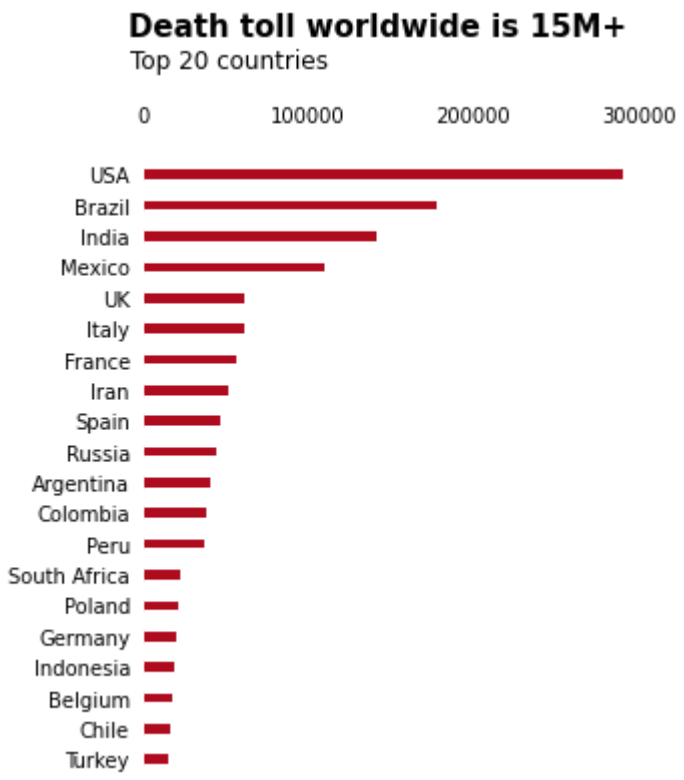


In [50]:
```
tot = death_df["Total_Deaths"]
cont = death_df["Country_Other"]
a = list(cont)
b = list(tot)
fig, ax = plt.subplots(figsize=(4.5, 6))
ax.barh(cont, tot, height=0.30)
ax.spines["left"].set_visible(False)
ax.spines["right"].set_visible(False)
ax.spines["bottom"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.tick_params(bottom=False, left=False)
ax.xaxis.tick_top()
ax.tick_params(top=False, left=False)
ax.set_xticks([0, 100000, 200000, 300000]);
```



In [64]:
```
tot = death_df["Total_Deaths"]
cont = death_df["Country_Other"]
a = list(cont)
b = list(tot)
fig, ax = plt.subplots(figsize=(4.5, 6))
ax.barh(cont, tot, height=0.30, color='#af0b1e')
ax.spines["left"].set_visible(False)
ax.spines["right"].set_visible(False)
ax.spines["bottom"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.tick_params(bottom=False, left=False)
ax.xaxis.tick_top()
ax.tick_params(top=False, left=False)
ax.set_xticks([0, 100000, 200000, 300000])
ax.text(x=-10000, y=23.5, s='Death toll worldwide is 15M+',
        size=15, weight="bold")
ax.text(x=-8000, y=22.5, s='Top 20 countries',
        size=12)
```

Out[64]:
```
Text(-8000, 22.5, 'Top 20 countries')
```

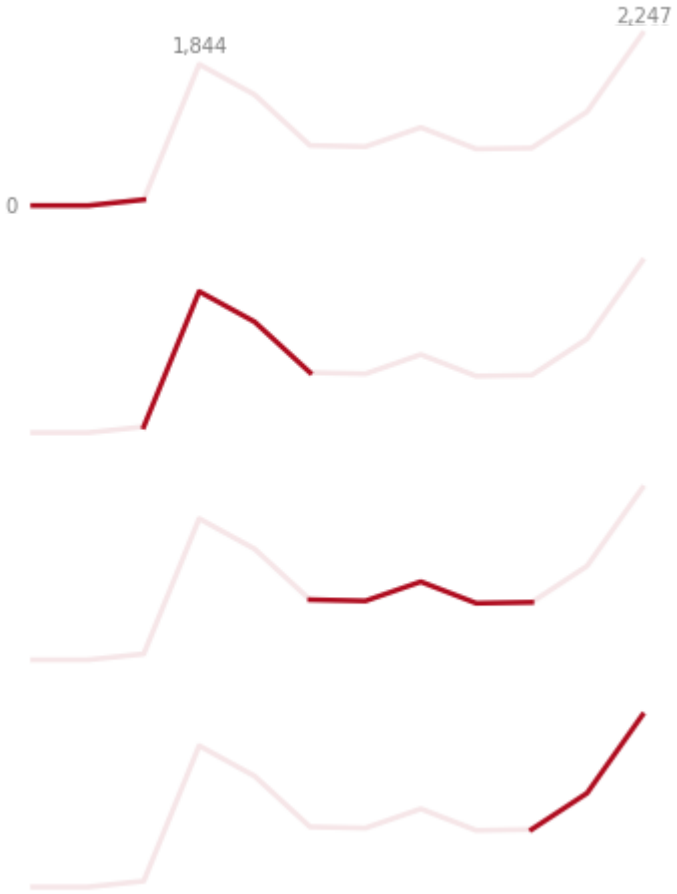


In [68]:
```
average_df
```

Out[68]:

	Month	New_deaths
0	1	0
1	2	0
2	3	77
3	4	1844
4	5	1448
5	6	785
6	7	769
7	8	1020
8	9	739
9	10	751
10	11	1225
11	12	2247

In [110]:
```
jan_mar = average_df["New_deaths"][:3]
january_march = average_df["Month"][:3]
mar_june = average_df["New_deaths"][2:6]
march_june = average_df["Month"][2:6]
june_oct = average_df["New_deaths"][5:10]
June_oct = average_df["Month"][5:10]
oct_dec = average_df["New_deaths"][9:12]
Oct_dec = average_df["Month"][9:12]
fig, (ax1, ax2, ax3, ax4) = plt.subplots(nrows=4, ncols=1, figsize=(6, 8))
ax1.plot(average_df["Month"], average_df["New_deaths"], color='#af0b1e', alpha=0.
ax1.plot(january_march, jan_mar, color='#af0b1e', alpha=1.0, linewidth=2.5)
ax1.set_yticklabels([])
ax1.set_xticklabels([])
ax1.spines["left"].set_visible(False)
ax1.spines["right"].set_visible(False)
ax1.spines["bottom"].set_visible(False)
ax1.spines["top"].set_visible(False)
ax1.tick_params(bottom=False, left=False)
# Assume the rest of the code is written
ax1.text(0.5, -80, '0', alpha=0.5)
ax1.text(3.5, 2000, '1,844', alpha=0.5)
ax1.text(11.5, 2400, '2,247', alpha=0.5)
ax2.plot(average_df["Month"], average_df["New_deaths"], color='#af0b1e', alpha=0.
ax2.plot(march_june, mar_june, color='#af0b1e', alpha=1.0, linewidth=2.5)
ax2.set_yticklabels([])
ax2.set_xticklabels([])
ax2.spines["left"].set_visible(False)
ax2.spines["right"].set_visible(False)
ax2.spines["bottom"].set_visible(False)
ax2.spines["top"].set_visible(False)
ax2.tick_params(bottom=False, left=False)
ax3.plot(average_df["Month"], average_df["New_deaths"], color='#af0b1e', alpha=0.
ax3.plot(June_oct, june_oct, color='#af0b1e', alpha=1.0, linewidth=2.5)
ax3.set_yticklabels([])
ax3.set_xticklabels([])
ax3.spines["left"].set_visible(False)
ax3.spines["right"].set_visible(False)
ax3.spines["bottom"].set_visible(False)
ax3.spines["top"].set_visible(False)
ax3.tick_params(bottom=False, left=False)
ax4.plot(average_df["Month"], average_df["New_deaths"], color='#af0b1e', alpha=0.
ax4.plot(Oct_dec, oct_dec, color='#af0b1e', alpha=1.0, linewidth=2.5)
ax4.set_yticklabels([])
ax4.set_xticklabels([])
ax4.spines["left"].set_visible(False)
ax4.spines["right"].set_visible(False)
ax4.spines["bottom"].set_visible(False)
ax4.spines["top"].set_visible(False)
ax4.tick_params(bottom=False, left=False);
```



In [84]:
```
average_df["New_deaths"].sum() / 12
```

Out[84]: 908.75

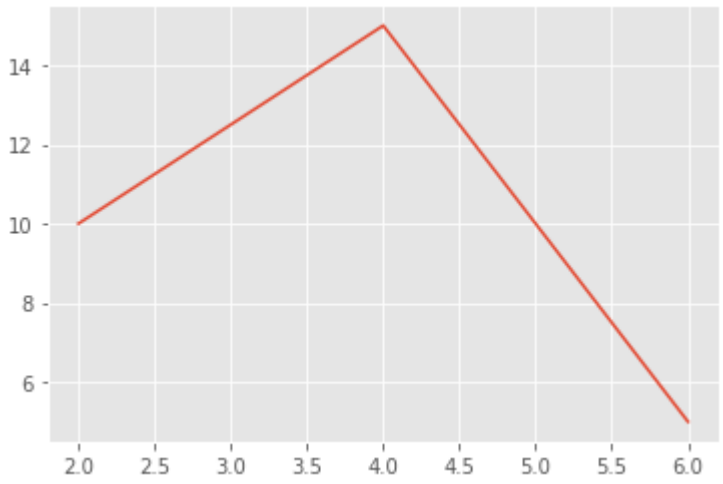
In [111]:
```
import matplotlib.style as style
```

In [112]:
```
style.available
```

Out[112]:
```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn',
 'seaborn-bright',
 'seaborn-colorblind',
 'seaborn-dark',
 'seaborn-dark-palette',
 'seaborn-darkgrid',
 'seaborn-deep',
 'seaborn-muted',
 'seaborn-notebook',
 'seaborn-paper',
 'seaborn-pastel',
 'seaborn-poster',
 'seaborn-talk',
 'seaborn-ticks',
 'seaborn-white',
 'seaborn-whitegrid',
 'tableau-colorblind10']
```

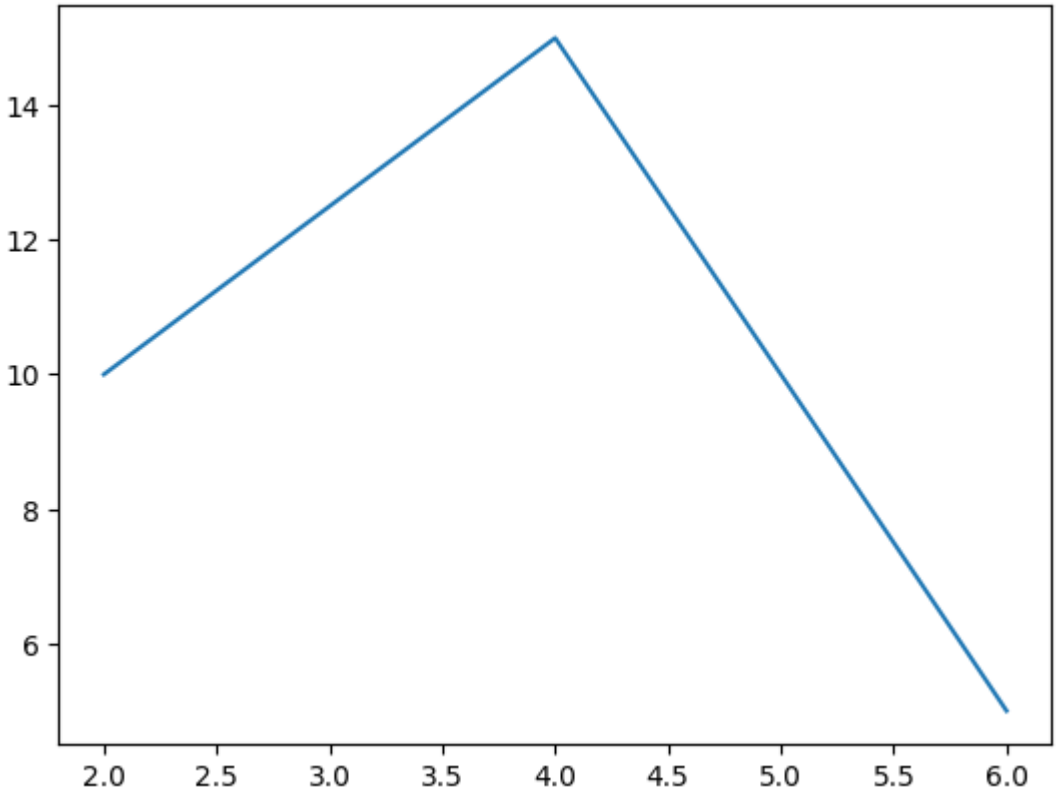
In [115]:
```
style.use("ggplot")
plt.plot([2, 4, 6], [10, 15, 5])
```

Out[115]:
```
[<matplotlib.lines.Line2D at 0x23bf5848>]
```



In [118]:
```
style.use("default")
plt.plot([2, 4, 6], [10, 15, 5])
```

Out[118]:
```
[<matplotlib.lines.Line2D at 0x23d84b08>]
```



In [142]:
```
import pandas as pd
white_wine = pd.read_csv('white_wine.csv', sep = ';')
red_wine = pd.read_csv('red_wine.csv', sep = ';')
```

In [144]:
```
red_wine
```

Out[144]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9
...	...	...	...	...	...	...	...	...	...	...	
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11

1599 rows × 12 columns

In [154]:
```
red_corr = red_wine.corr()["quality"][:-1]
red_corr
```

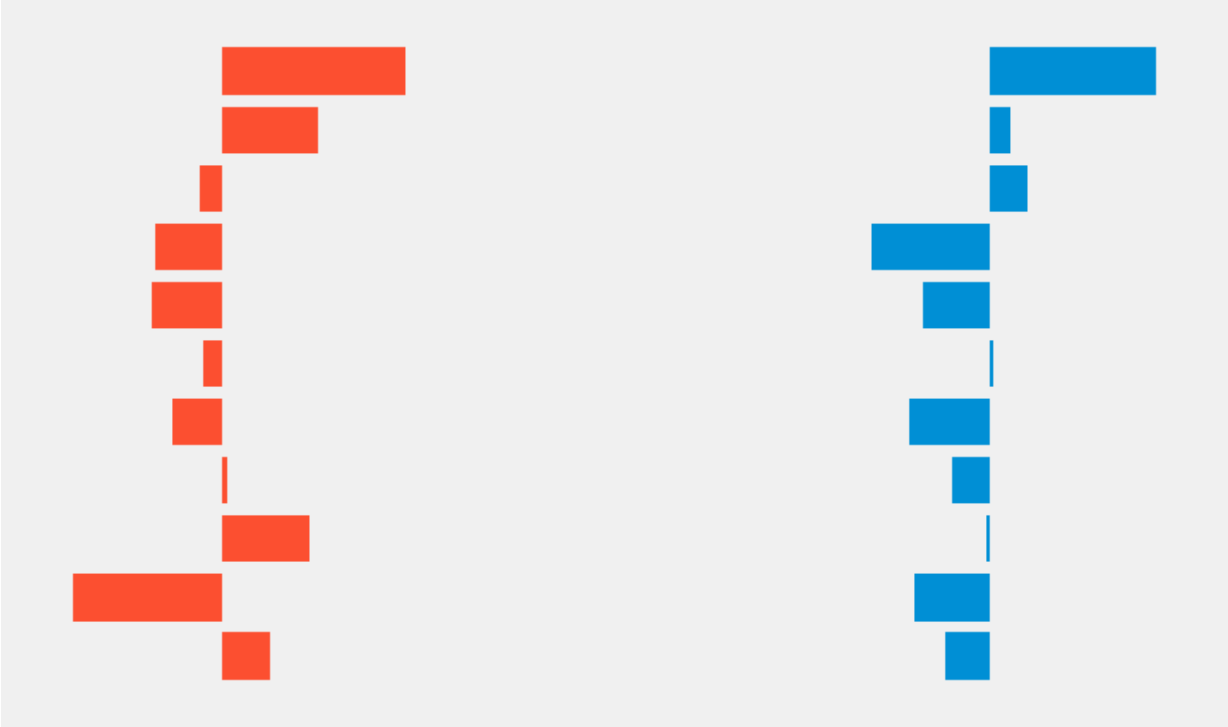
Out[154]:
```
fixed acidity           0.124052
volatile acidity       -0.390558
citric acid             0.226373
residual sugar          0.013732
chlorides              -0.128907
free sulfur dioxide    -0.050656
total sulfur dioxide   -0.185100
density                -0.174919
pH                     -0.057731
sulphates               0.251397
alcohol                 0.476166
Name: quality, dtype: float64
```

In [155]:
```
white_corr = white_wine.corr()["quality"][:-1]
white_corr
```

Out[155]:
```
fixed acidity          -0.113663
volatile acidity       -0.194723
citric acid            -0.009209
residual sugar         -0.097577
chlorides              -0.209934
free sulfur dioxide     0.008158
total sulfur dioxide   -0.174737
density                -0.307123
pH                      0.099427
sulphates               0.053678
alcohol                 0.435575
Name: quality, dtype: float64
```

In [163]:
```
style.use('fivethirtyeight')
fig, ax = plt.subplots(figsize=(8, 5))
ax.barh(white_corr.index, white_corr, left=2)
ax.barh(red_corr.index, red_corr)
ax.grid(b=False)
ax.set_yticklabels([])
ax.set_xticklabels([])
plt.show()
```

```
C:\New\envs\snakes\lib\site-packages\ipykernel_launcher.py:5: MatplotlibDepreca
tionWarning: The 'b' parameter of grid() has been renamed 'visible' since Matpl
otlib 3.5; support for the old name will be dropped two minor releases later.
  """
```



In [ ]:
```
```