Global Region Demographic Analysis

Code to grab the data

In [52]:
```
import pandas as pd
import altair as alt
import vegafusion as vf
import playwright
import os
df = pd.read_csv('gapminder.csv')
df.head()
len(df)
df.head()
```

Out[52]:

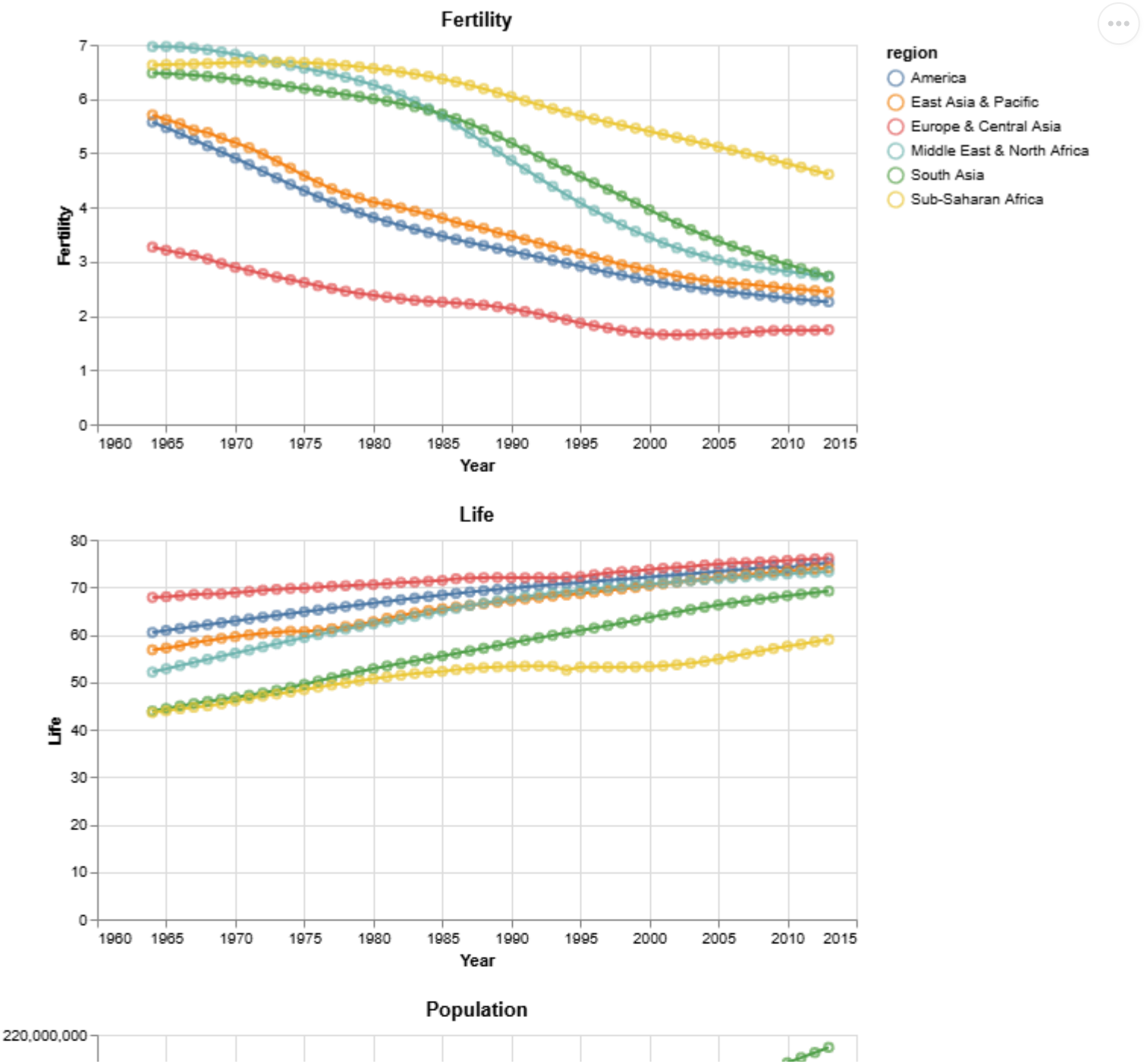
	Country	Year	fertility	life	population	child_mortality	gdp	region
0	Afghanistan	1964	7.671	33.639	10474903.0	339.7	1182.0	South Asia
1	Afghanistan	1965	7.671	34.152	10697983.0	334.1	1182.0	South Asia
2	Afghanistan	1966	7.671	34.662	10927724.0	328.7	1168.0	South Asia
3	Afghanistan	1967	7.671	35.170	11163656.0	323.3	1173.0	South Asia
4	Afghanistan	1968	7.671	35.674	11411022.0	318.1	1187.0	South Asia

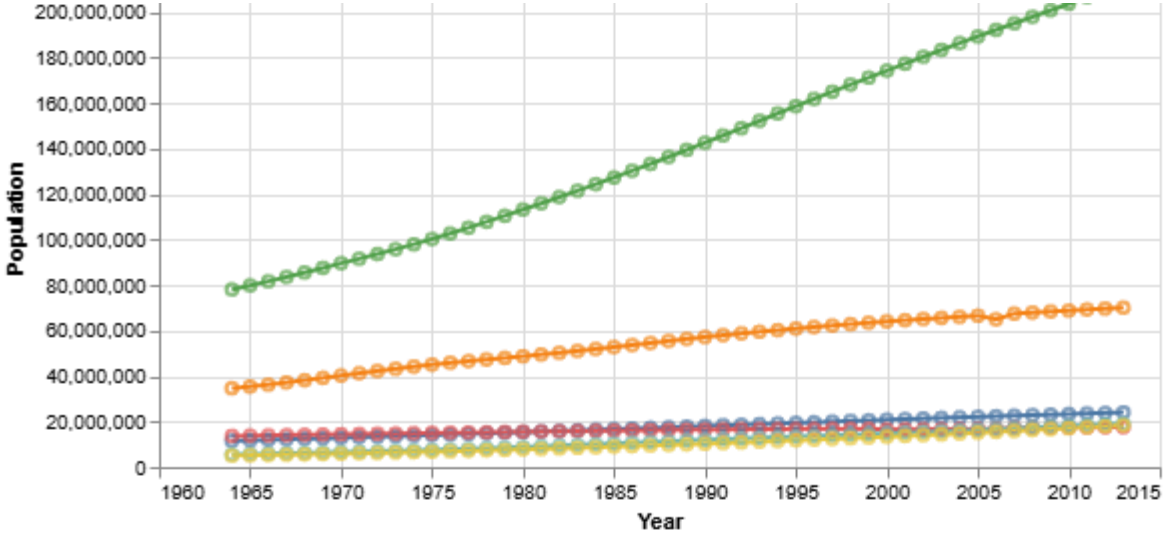
In [2]:
```
# First sort the dataframe by region and Year
data_df = df.sort_values(by=['region', 'Year'])
```

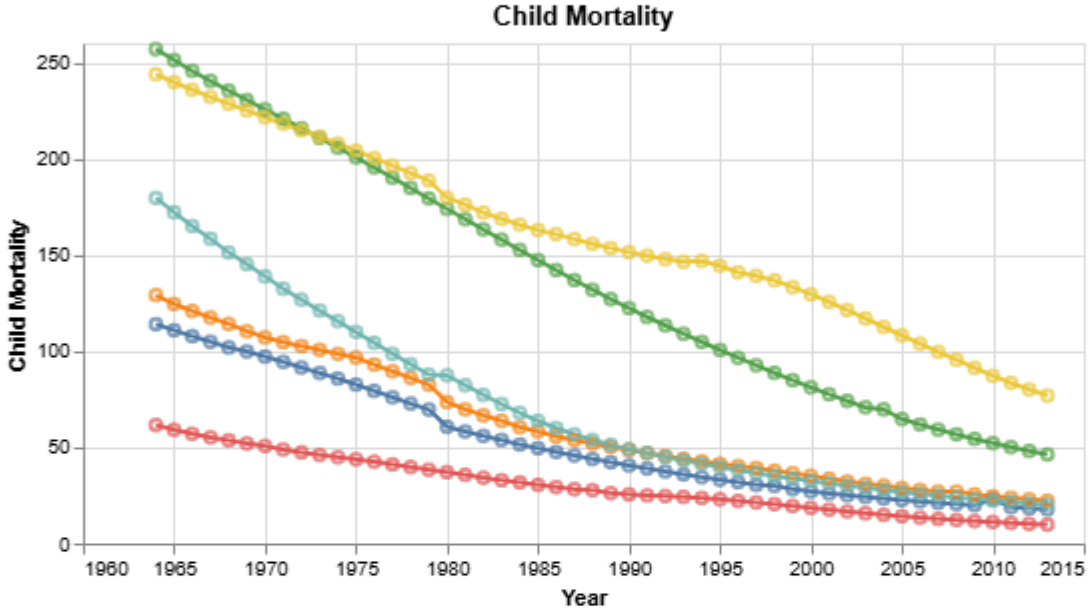
In [3]:
```
import altair as alt

# List of variables to plot
variables = ['fertility', 'life', 'population', 'child_mortality', 'gdp']

# Group data once
grouped = data_df.groupby(['Year', 'region'], as_index=False)[variables].mean()
```

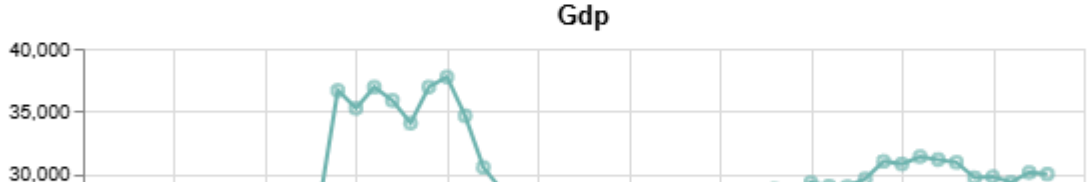
```
# Define selection
brush = alt.selection_interval()

# Create chart list
charts = []

for var in variables:
    line = alt.Chart(grouped).mark_line().encode(
        x=alt.X('Year:Q', axis=alt.Axis(format='d')),
        y=alt.Y(f'{var}:Q', title=var.replace('_', ' ').title()),
        color=alt.condition(brush, 'region:N', alt.value('lightgray')),
        tooltip=['Year:Q', f'{var}:Q', 'region:N']
    )

    point = alt.Chart(grouped).mark_point().encode(
        x=alt.X('Year:Q'),
        y=alt.Y(f'{var}:Q'),
        color=alt.condition(brush, 'region:N', alt.value('lightgray')),
        tooltip=['Year:Q', f'{var}:Q', 'region:N']
    )

    chart = (line + point).properties(
        title=var.replace('_', ' ').title(),
        width=500,
        height=250
    ).add_params(brush)

    charts.append(chart)

# Combine charts vertically
combined = alt.vconcat(*charts)
combined
```

Out[3]:

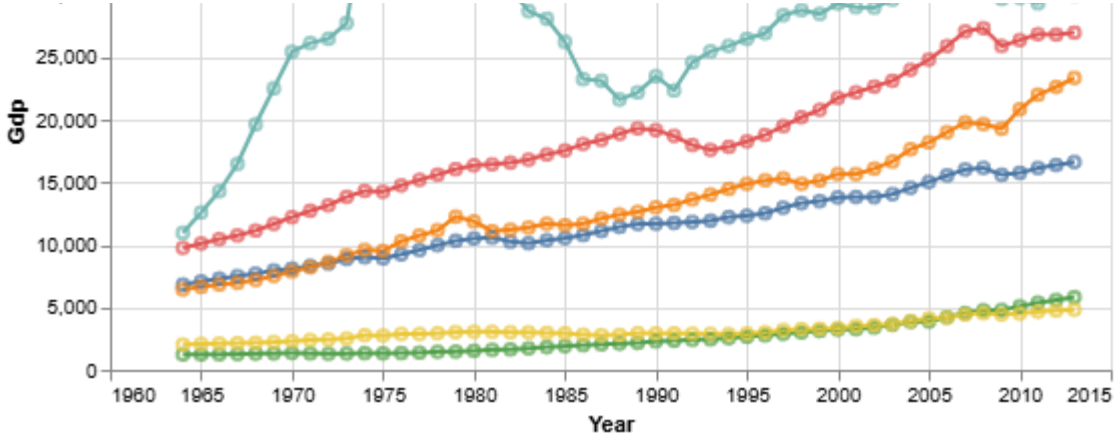

Fertility
Fertility
Year
region
America
East Asia & Pacific
Europe & Central Asia
Middle East & North Africa
South Asia
Sub-Saharan Africa
Life
Life
Year
Population
220,000,000


Population
200,000,000
180,000,000
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
Year


Child Mortality
Child Mortality
250
200
150
100
50
0
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
Year


Gdp
40,000
35,000
30,000

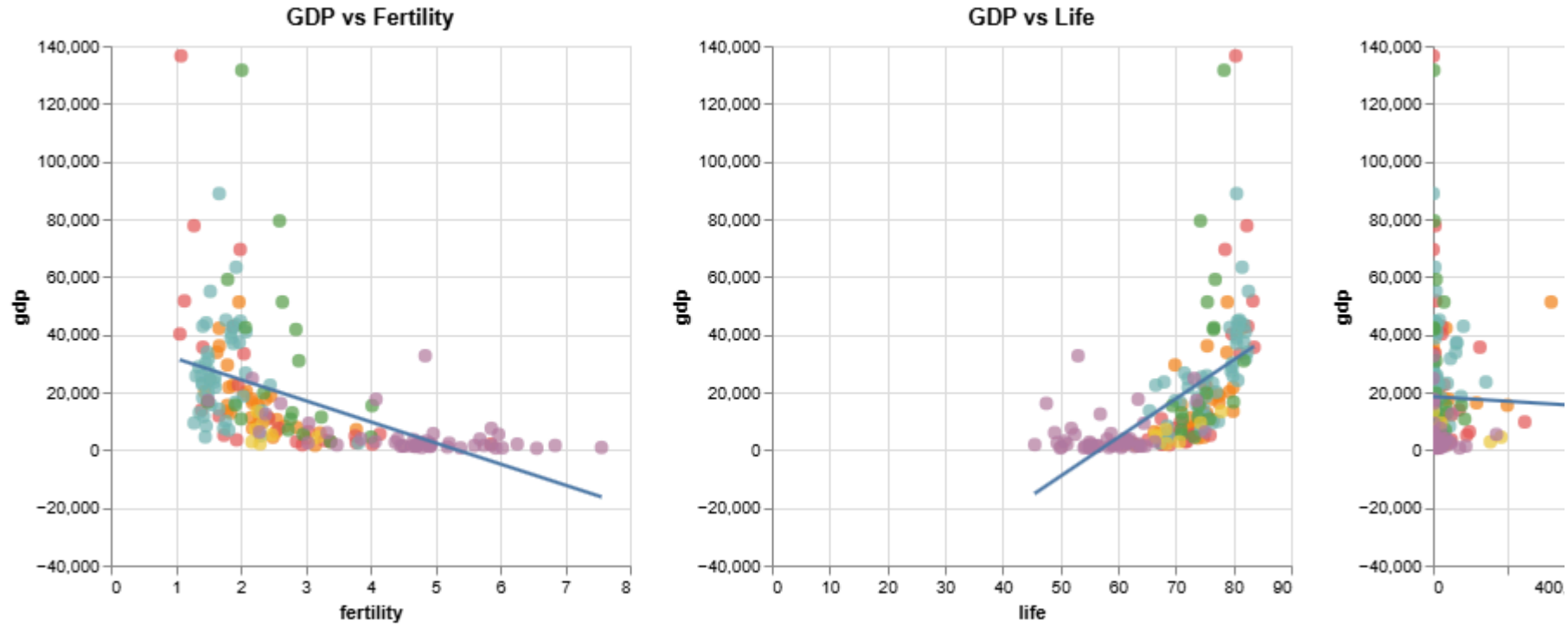
In [7]:



In [22]:

```
charts = []
for var in variables:
    base = alt.Chart(filtered_country).encode(
        x=var,
        y='gdp',
        color='region',
        tooltip=['Country', 'region', 'gdp', var]
    )

    scatter = base.mark_circle(size=60)

    regression = base.transform_regression(
        var, 'gdp'
    ).mark_line(color='black')

    chart = (scatter + regression).properties(
        title=f'GDP vs {var.capitalize()}'
    ).interactive()

    charts.append(chart)

# Combine charts horizontally
combined_chart = alt.hconcat(*charts).resolve_scale(
    y='shared'
)

combined_chart
```

Out[22]:

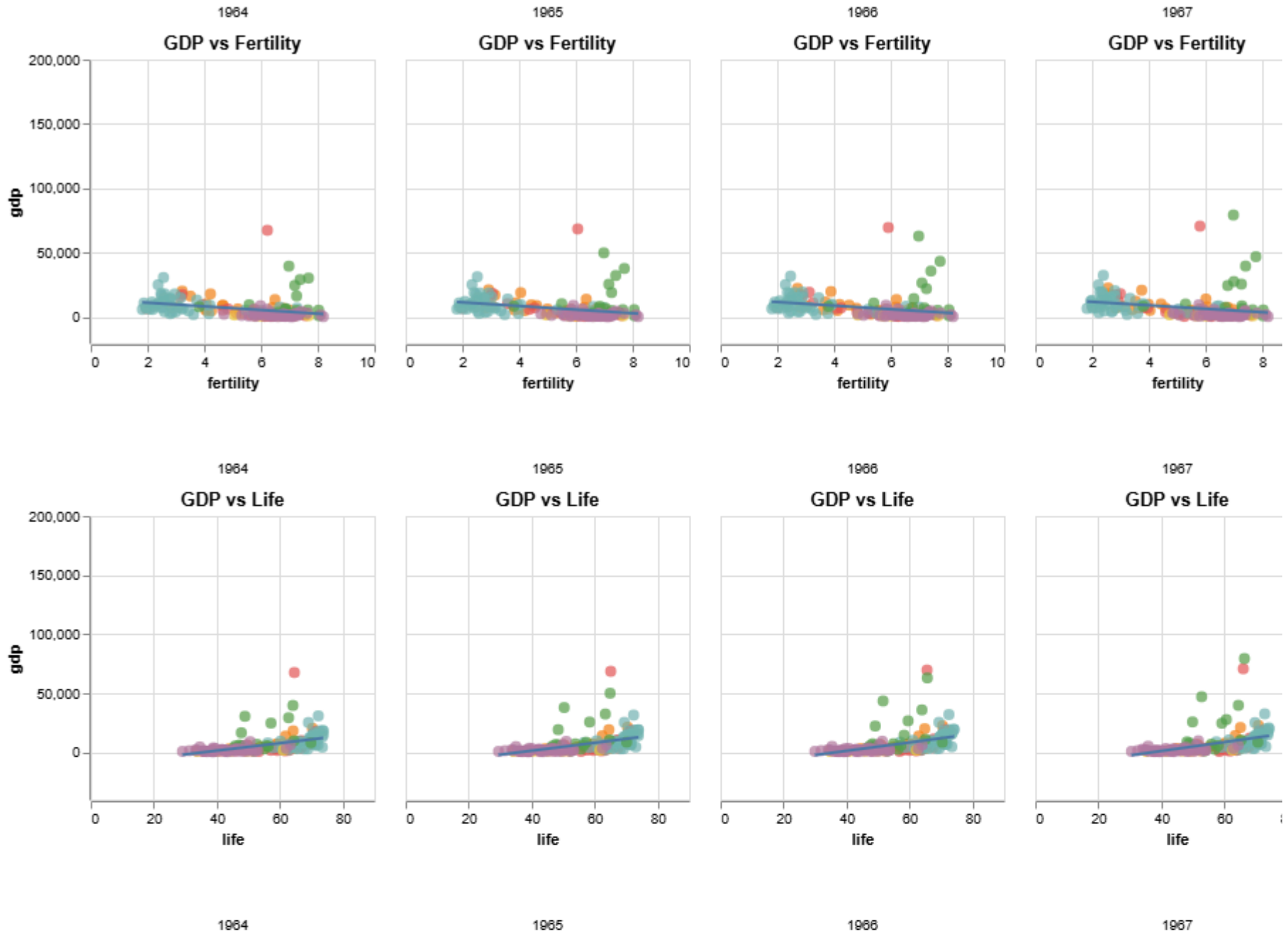


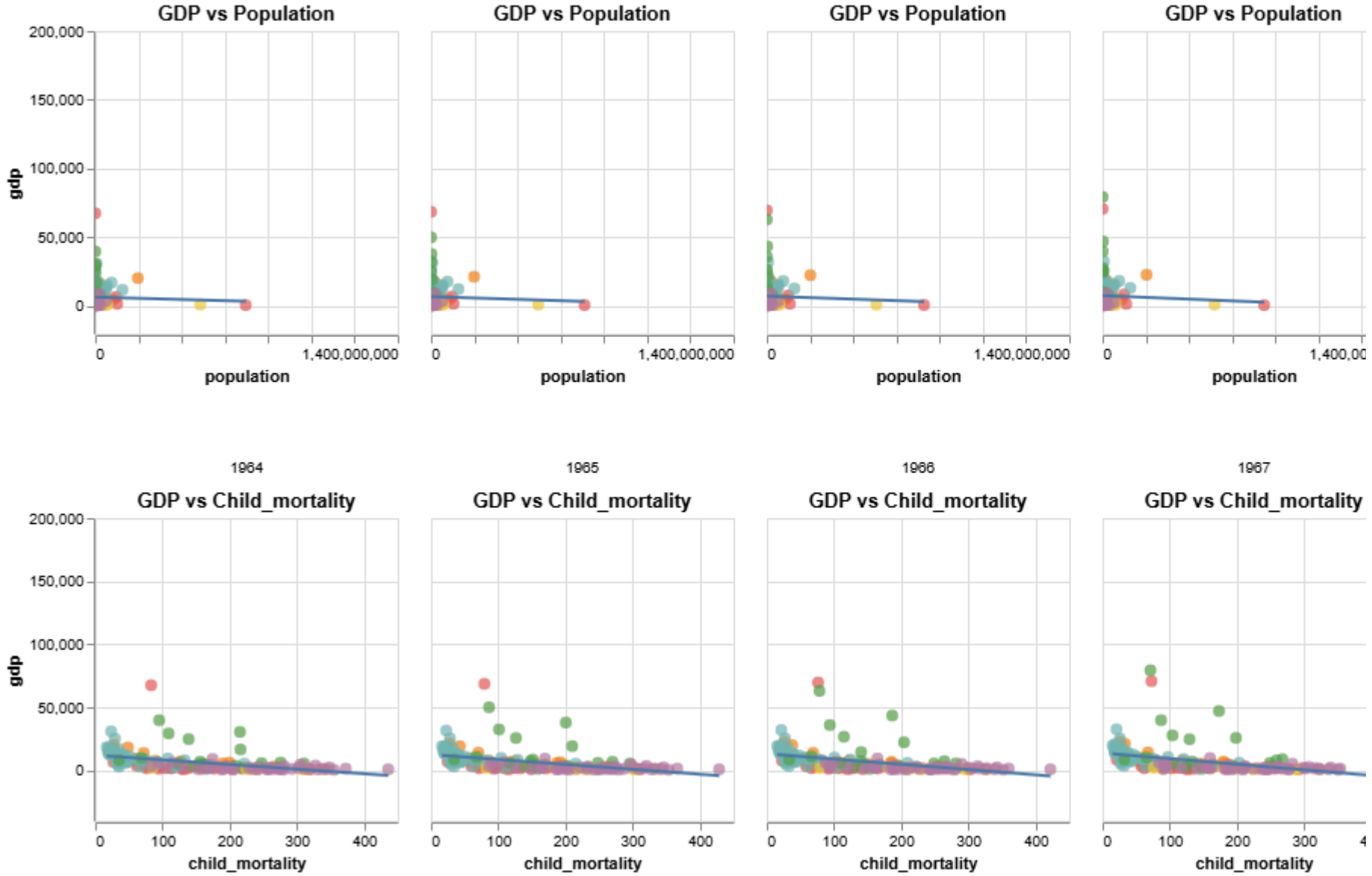
In [23]:

```
charts = []

for var in variables:
    base = alt.Chart(country).encode(
        x=var,
        y='gdp',
        color='region',
        tooltip=['Country', 'region', 'Year', 'gdp', var]
    )

    scatter = base.mark_circle(size=60)

    regression = base.transform_regression(
        var, 'gdp', groupby=['Year']
    ).mark_line(color='black')

    chart = (scatter + regression).properties(
```

```
        width=200,
        height=200,
        title=f'GDP vs {var.capitalize()}'
    ).facet(
        column='Year:N'
    ).interactive()

    charts.append(chart)

# Combine vertically, one row per variable
combined_chart = alt.vconcat(*charts)

combined_chart
```

Out[23]:

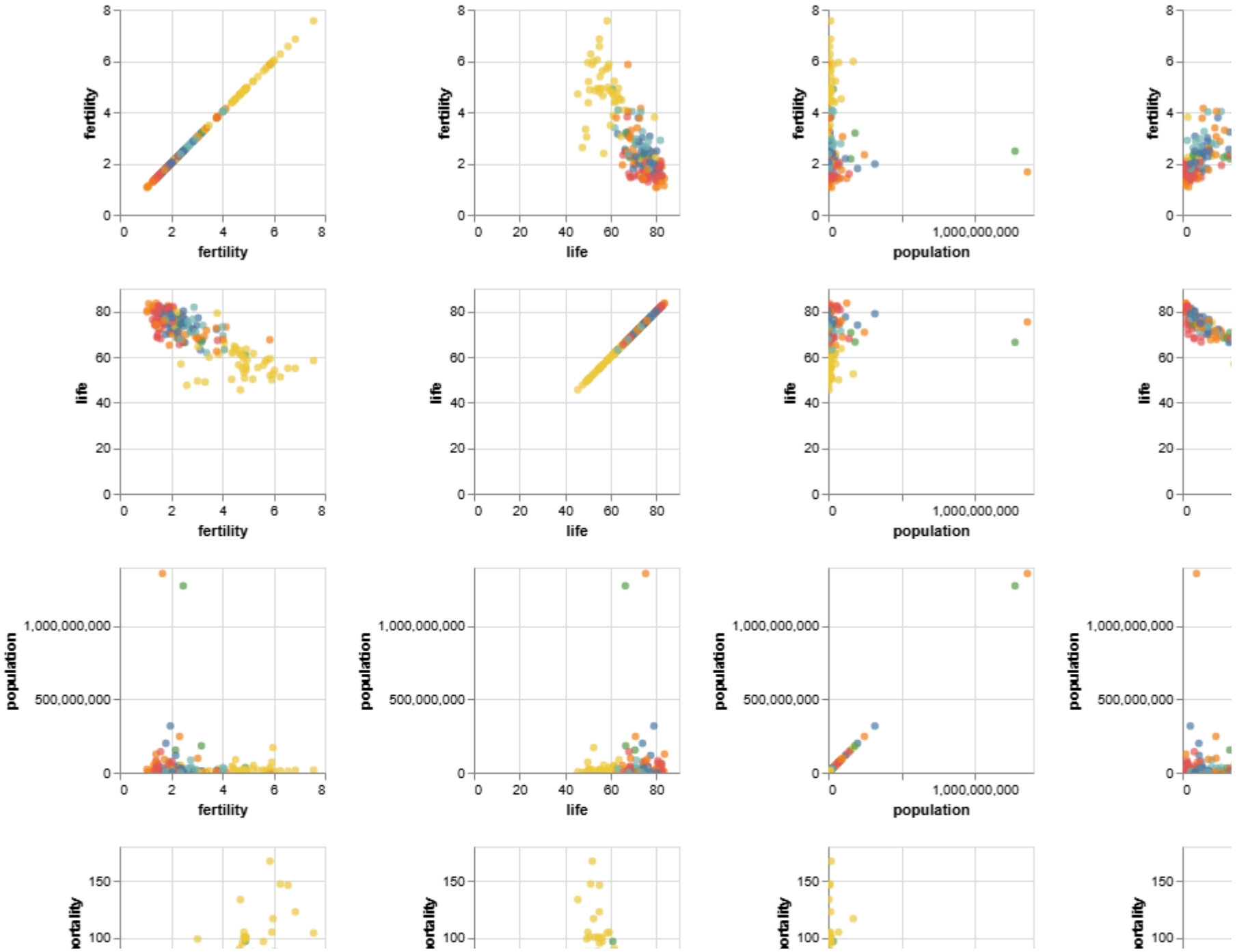

1964
GDP vs Fertility
1965
GDP vs Fertility
1966
GDP vs Fertility
1967
GDP vs Fertility
gdp
200,000
150,000
100,000
50,000
0
0 2 4 6 8 10
fertility
1964
GDP vs Life
1965
GDP vs Life
1966
GDP vs Life
1967
GDP vs Life
gdp
200,000
150,000
100,000
50,000
0
0 20 40 60 80
life
1964
1965
1966
1967

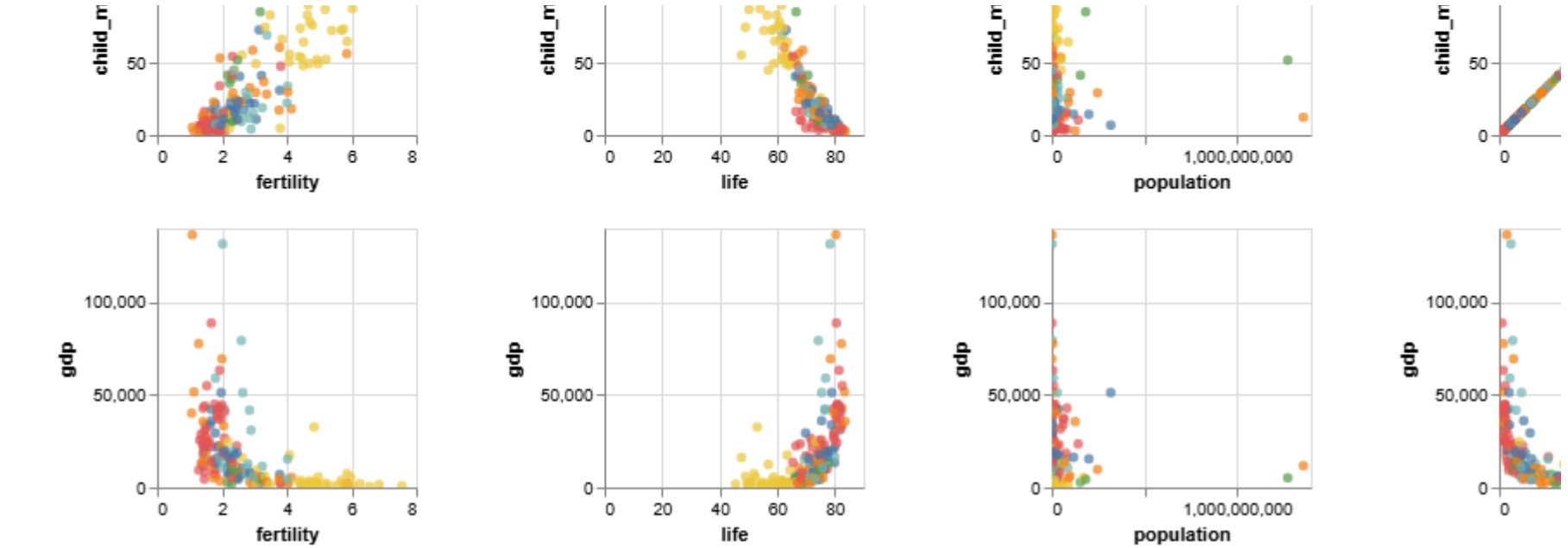


In [25]:
```
#RELAtionship
gapminder_2013 = df[df['Year']==latest_year]

alt.Chart(gapminder_2013).mark_circle().encode(
```

```
    alt.X(alt.repeat("column"), type='quantitative'),
    alt.Y(alt.repeat("row"), type='quantitative'),
    color='region:N'
).properties(
    width=150,
    height=150
).repeat(
    row=['fertility','life','population','child_mortality','gdp'],
    column=['fertility','life','population','child_mortality','gdp']
).interactive()
```

Out[25]:


fertility
life
population
0
2
4
6
8
20
40
60
80
500,000,000
1,000,000,000
150
100



In [27]:
```
df.region.unique()
region_df = grouped
region_df['gdp_per_100']=region_df['gdp']/100
region_df = pd.melt(region_df,id_vars=['Year','region'],value_vars=['gdp_per_100','fertility','child_mortality','life'])
region_df.head()
```

Out[27]:

	Year	region	variable	value
0	1964	America	gdp_per_100	68.138750
1	1964	East Asia & Pacific	gdp_per_100	64.315000
2	1964	Europe & Central Asia	gdp_per_100	97.600625
3	1964	Middle East & North Africa	gdp_per_100	109.621579
4	1964	South Asia	gdp_per_100	12.338750

In [29]:
```
sort_des = grouped.groupby(['region']).sum().reset_index()
sort_des = sort_des.sort_values(by='gdp', ascending=False)
```

```
sort_des
```

Out[29]:

	region	Year	fertility	life	population	child_mortality	gdp	gdp_per_100
3	Middle East & North Africa	99425	248.500952	3259.715048	5.856515e+08	3543.322553	1.375537e+06	13755.365789
2	Europe & Central Asia	99425	110.708874	3596.146672	8.000828e+08	1515.239743	9.221023e+05	9221.022708
1	East Asia & Pacific	99425	186.291786	3305.238612	2.733987e+09	3096.965512	6.668078e+05	6668.078462
0	America	99425	174.303069	3436.112550	8.872860e+08	2637.111889	5.799961e+05	5799.960938
5	Sub-Saharan Africa	99425	297.805240	2583.221290	5.254992e+08	7983.357034	1.576214e+05	1576.214255
4	South Asia	99425	250.208125	2856.885500	7.033908e+09	6888.357500	1.276325e+05	1276.325000

In [49]:
```
region_list = list(sort_des['region'].unique())
high_region = region_list[0:4]
print(high_region)
low_region=region_list[-2:]
print(low_region)
```

```
['Middle East & North Africa', 'Europe & Central Asia', 'East Asia & Pacific', 'America']
['Sub-Saharan Africa', 'South Asia']
```

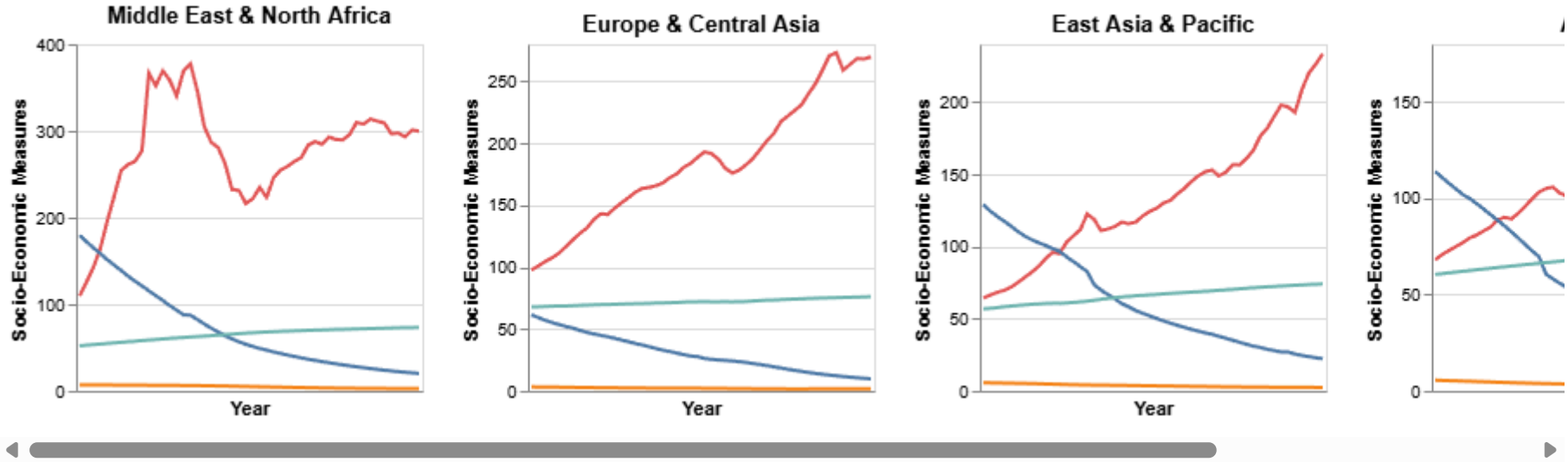
In [48]:
```
region_df_expand = grouped
region_df_expand['gdp_per_1000'] = grouped['gdp']/1000
region_df_expand['pop_per_mil'] = grouped['population']/1000000
region_df_expand = pd.melt(grouped,id_vars=['Year','region'],value_vars=['fertility','child_mortality','life'])
#'gdp_per_1000','pop_per_mil',
```

In [50]:
```
print('\033[1m' + 'HIGH GDP COUNTRIES' + '\033[0m')

chart = alt.hconcat()

for region in high_region:
    base = alt.Chart(region_df, title=region).mark_line().encode(
        alt.X('Year:O', title='Year', axis=alt.Axis(values=list(range(1, 50, 10)))),
        alt.Y('value:Q', title='Socio-Economic Measures'),
        alt.Color('variable:N')
    ).transform_filter(
```

```
        alt.datum.region == region  # ✅ Correct use of `alt.datum`, no import
    ).interactive().properties(
        width=200,
        height=200
    )

    chart |= base

chart
```

HIGH GDP COUNTRIES

Out[50]:



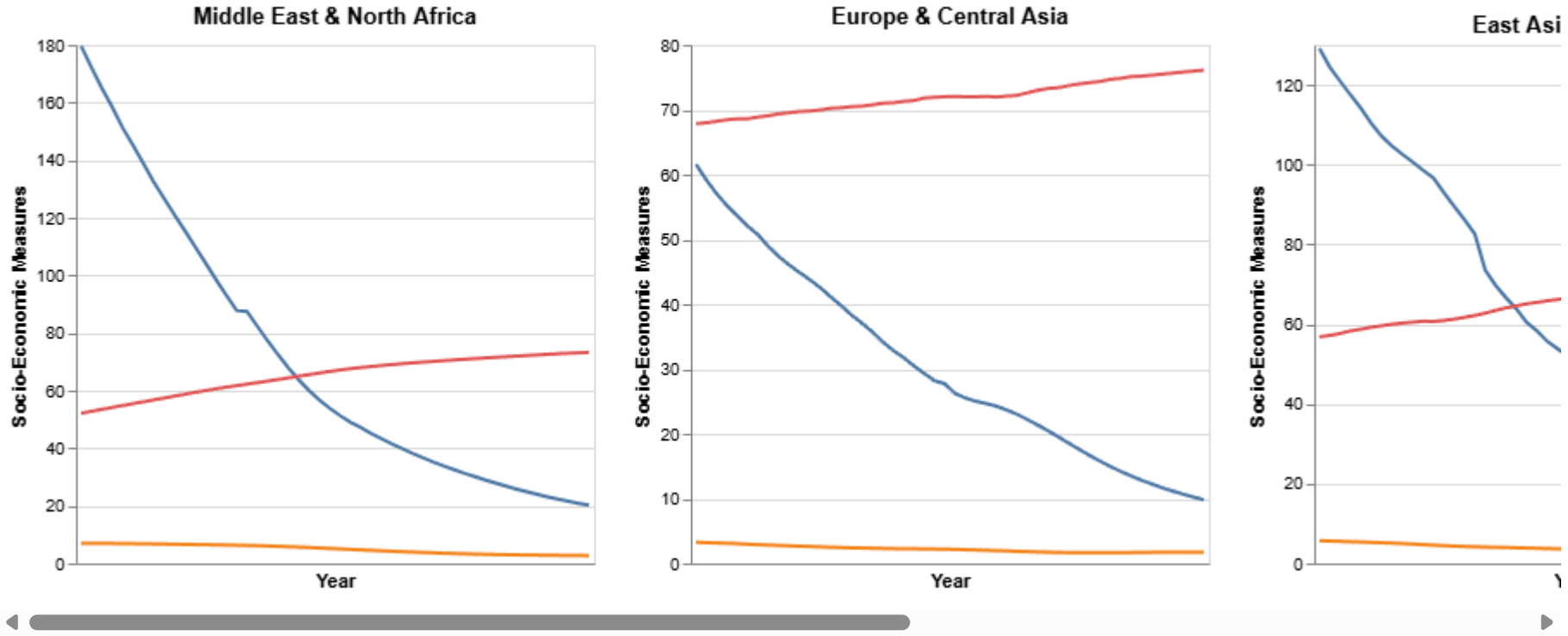
In [43]:

```
print('\033[1m' + 'FERTILITY, MORTALITY, LIFE' + '\033[0m')

chart = alt.hconcat()

for region in high_region:
    base = alt.Chart(region_df_expand, title=region).mark_line().encode(
        alt.X('Year:O', title='Year', axis=alt.Axis(values=list(range(1, 50, 10)))),
        alt.Y('value:Q', title='Socio-Economic Measures'),
        alt.Color('variable:N')
    ).transform_filter(
        alt.datum.region == region
    ).interactive().properties(
        width=300,
```

```
        height=300
    )

    chart |= base  # Concatenate the filtered chart

chart
```

FERTILITY, MORTALITY, LIFE

Out[43]:



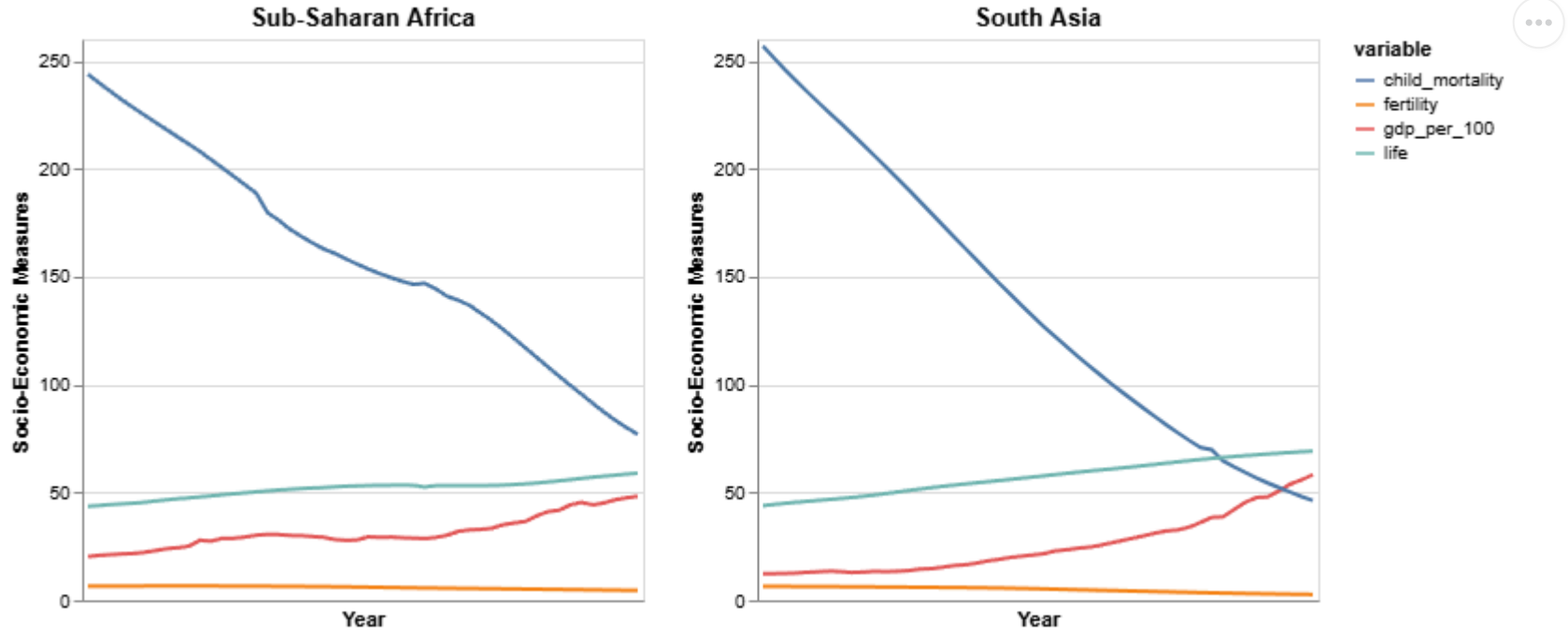
In [44]:

```
print('\033[1m' + 'LOW GDP COUNTRIES' + '\033[0m')

chart = alt.hconcat()

for region in low_region:
    base = alt.Chart(region_df, title=region).mark_line().encode(
        alt.X('Year:O', title='Year', axis=alt.Axis(values=list(range(1, 50, 10)))),
        alt.Y('value:Q', title='Socio-Economic Measures'),
        alt.Color('variable:N')
    ).transform_filter(
```

```
        alt.datum.region == region
    ).interactive().properties(
        width=300,
        height=300
    )

    chart |= base  # Combine charts side by side

chart

#when GDP per capita increases,we see an exponential drop in Child_mortality rate (through time), slight increase in life, fer
#probably also increasing gdp per capita
```

LOW GDP COUNTRIES

Out[44]:



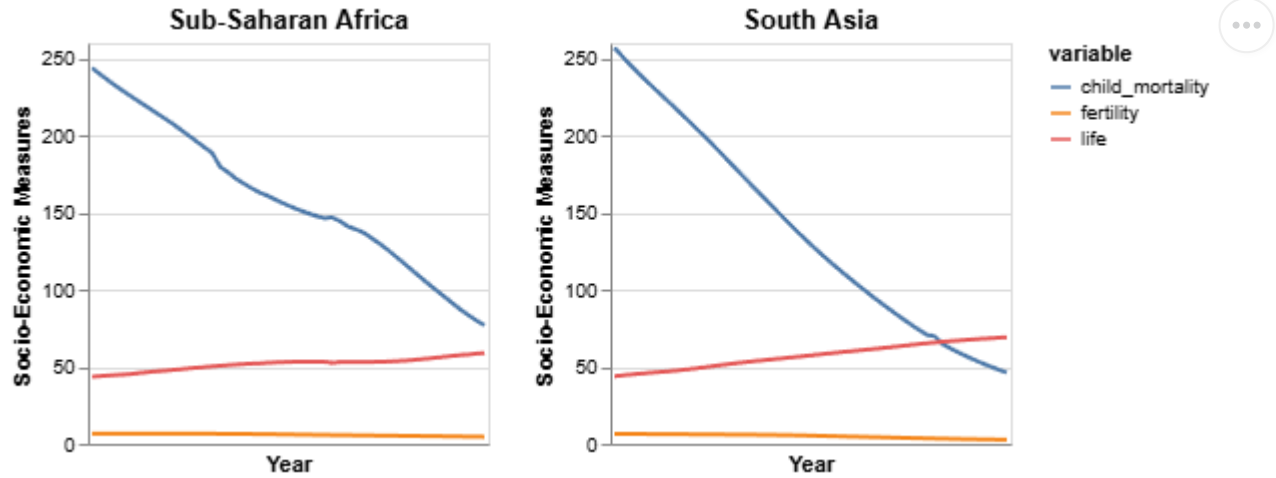
In [46]:

```
print('\033[1m' + 'FERTILITY, MORTALITY, LIFE' + '\033[0m')

chart = alt.hconcat()

for region in low_region:
```

```
    base = alt.Chart(region_df_expand, title=region).mark_line().encode(
        alt.X('Year:O', title='Year', axis=alt.Axis(values=list(range(1, 50, 10)))),
        alt.Y('value:Q', title='Socio-Economic Measures'),
        alt.Color('variable:N')
    ).transform_filter(
        alt.datum.region == region  # ✅ Correct filtering
    ).interactive().properties(
        width=200,
        height=200
    )

    chart |= base

chart
```

FERTILITY, MORTALITY, LIFE

Out[46]:



In [ ]: