# Gender Inequality

### Fidan Aydamirova

### Radhika Magaji

### Pooja Bhatia

In [1]:

```
import pandas as pd
import altair as alt
from IPython.display import HTML
import matplotlib.pyplot as plt
import geopandas
```

In [2]:

```
alt.data_transformers.enable('default', max_rows=None)
```

Out[2]:

```
DataTransformerRegistry.enable('default')
```

## Load all Datasets

In [3]:

```
jobs = pd.read_csv('JobsData.csv')
parliament = pd.read_csv('Par_Women_Data.csv')
women_wage_perc = pd.read_excel('wage_per_occupation.xlsx', sheet_name="Table 14")
lp = pd.read_csv("Labor Force Participation Rate of Mothers and Fathers by Age of Youngest Child.csv",
                 skiprows=1)
world_data = pd.read_csv("WDIData.csv")
mortality  = pd.read_csv("Maternal_Mortality_ratio.csv")
inequality  = pd.read_csv("gender-inequality-index-from-the-human-development-report.csv")
```

In [4]:

```
jobs = jobs.rename(columns = {"Indicator Name":"Variables"})
```

In [5]:

```
job_list_of_values = ["Employment in agriculture (% of total employment) (modeled ILO estimate)",
                  "Employment in agriculture, female (% of female employment) (modeled ILO estimate)",
                  "Employment in agriculture, male (% of male employment) (modeled ILO estimate)",
                  "Employment in industry (% of total employment) (modeled ILO estimate)",
                  "Employment in industry, female (% of female employment) (modeled ILO estimate)",
                  "Employment in industry, male (% of male employment) (modeled ILO estimate)",
                  "Employment in services (% of total employment) (modeled ILO estimate)",
                  "Employment in services, female (% of female employment) (modeled ILO estimate)",
                  "Employment in services, male (% of male employment) (modeled ILO estimate)",
                  "Labor force with advanced education, female (% of female working-age population with advanced educati
                  "Labor force with basic education, female (% of female working-age population with basic education)",
                  "Labor force with intermediate education, female (% of female working-age population with intermediate
                  "Labor force participation rate, female (% of female population ages 15+) (modeled ILO estimate)",
                  "Fertility rate, total (births per woman)",
           "Literacy rate, adult female (% of females ages 15 and above)",
           "Literacy rate, adult male (% of males ages 15 and above)",
           "Self-employed, female (% of female employment) (modeled ILO estimate)",
           "Self-employed, male (% of male employment) (modeled ILO estimate)",
            ]
jobs_df = jobs[jobs['Variables'].isin(job_list_of_values)]
```

In [6]:

```
jobs_df_small = jobs_df.reset_index()
jobs_df_small = jobs_df_small.drop(columns = ['Indicator Code','index'])
jobs_dfp = jobs_df_small.pivot(index='Variables', columns=['Country Name', 'Country Code']).T
```

In [7]:

```
jDF = jobs_dfp
jDF = jobs_dfp.rename(columns={"Employment in agriculture (% of total employment) (modeled ILO estimate)":"Agriculture_
                  "Employment in agriculture, female (% of female employment) (modeled ILO estimate)":"Agriculture_Fema
                  "Employment in agriculture, male (% of male employment) (modeled ILO estimate)":"Agriculture_Male",
                  "Employment in industry (% of total employment) (modeled ILO estimate)":"Industry_Total",
                  "Employment in industry, female (% of female employment) (modeled ILO estimate)":"Industry_Female",
                  "Employment in industry, male (% of male employment) (modeled ILO estimate)":"Industry_Male",
                  "Employment in services (% of total employment) (modeled ILO estimate)":"Service_Total",
                  "Employment in services, female (% of female employment) (modeled ILO estimate)":"Service_Female",
                  "Employment in services, male (% of male employment) (modeled ILO estimate)":"Service_Male",

                  "Labor force with advanced education, female (% of female working-age population with advanced educati
                  "Labor force with basic education, female (% of female working-age population with basic education)":
                  "Labor force with intermediate education, female (% of female working-age population with intermediate
                  "Labor force participation rate, female (% of female population ages 15+) (modeled ILO estimate)":"la

            "Fertility rate, total (births per woman)":'Fertility',
           "Literacy rate, adult female (% of females ages 15 and above)":'lit_F',
           "Literacy rate, adult male (% of males ages 15 and above)":'lit_m',
           "Self-employed, female (% of female employment) (modeled ILO estimate)":'self_Emp_F',
           "Self-employed, male (% of male employment) (modeled ILO estimate)":'self_Emp_M'})
```

In [8]:

```
jDF.reset_index(inplace=True)
```

In [9]:

```
jDF.head()
```

Out[9]:

| Variables | level_0 | Country Name | Country Code | Agriculture_Total | Agriculture_Female | Agriculture_Male | Industry_Total | Industry_Female | Industry_Male |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1990 | Arab World | ARB | NaN | NaN | NaN | NaN | NaN | NaN |
| **1** | 1990 | East Asia & Pacific | EAS | NaN | NaN | NaN | NaN | NaN | NaN |
| **2** | 1990 | East Asia & Pacific (excluding high income) | EAP | NaN | NaN | NaN | NaN | NaN | NaN |
| **3** | 1990 | Euro area | EMU | NaN | NaN | NaN | NaN | NaN | NaN |
| **4** | 1990 | Europe & Central Asia | ECS | NaN | NaN | NaN | NaN | NaN | NaN |

5 rows × 21 columns

In [10]:

```
jDF = jDF.rename(columns={'level_0':'Year',
                         "Country Name":"Country",
                         "Country Code":"CODE"})
```

In [ ]:

In [11]:

```
years = jDF['Year'].unique() # get unique field values
years = list(filter(lambda x:  x > '2000', years)) # filter out None values
years.sort() # sort alphabetically
```

In [12]:

```
selectYear = alt.selection_single(
    name='Select',
    fields=['Year'],
    init={'Year': '2016'},
    #bind=alt.binding_range(min=1990, max=2016)
    bind={'Year': alt.binding_select(options=years)},
)
```

- Agriculture : Most South Asian women (about 60%) are employed in the field of agriculture and less than 1% of women from North America region are employed in Agriculture.
- Industry : Between 8-20% of women from these regions are employed in the industry field.
- Service : A whopping 90% of women from North America are employed in the Service field.
- Overall, except for South Asian women, most women over the world are employed mostly in service fields.

In [13]:

```
display(HTML("""
<style>
form.vega-bindings {
  position: absolute;
  left: 0px;
  top: 0px;
}
</style>
"""))
```

In [14]:

```
legend_labels = ("datum.label == 'Agriculture_Female' ? 'Agriculture' : datum.label == 'Industry_Female' ? 'Industry' :
axis_labels = ("datum.label == 'Agriculture_Female' ? 'Female' : datum.label == 'Industry_Female' ? 'Female' : datum.lab


color_category =['#3A2A51','#52A675','#FF595E'] #3 distinct
color_category1_light = ['#3A2A51','#BFAED5'] #2 lighter shade of 1 category color
color_category2_light = ['#52A675','#9FD0B4']
color_category3_light = ['#FF595E','#FFADB0']
heatmap = ['#3A2A51', '#FFC2C4']
heatmap1 = ['#FFC2C4','#3A2A51']
color_two_category = ['#3A2A51','#FF595E'] #2 distinct
#['#6A4C93','#1982C4','#FF924C']
#['#FF6B6B','#4ECDC4','#1A535C']#, '#638ccc'] #distinct; category
#['#000075','#f58231','#800000']
```

## What is the share of women employment by sectors?

In [15]:

```
stackedbar = alt.Chart(jDF).mark_bar().add_selection(selectYear).transform_filter(selectYear
).transform_fold(
    ['Agriculture_Female','Industry_Female','Service_Female']
).transform_filter(alt.FieldOneOfPredicate(field='Country',
                                           oneOf=['India','Azerbaijan','United States',
                                                  'Afghanistan','Yemen, Rep.']) #'Yemen, Rep.'
).encode(
    alt.Y('Country:N',
          sort=['Afghanistan','Yemen, Rep.','India','Azerbaijan','United States'], title=None),
    alt.X('value:Q',
          title="Female share(%)", axis=alt.Axis(tickMinStep = 100),
          scale= alt.Scale(domain=[0,100])),
    alt.Color('key:N',
              legend=alt.Legend(orient='right', titleOrient='top',
                                title='Employment Sector',labelExpr=legend_labels),
              scale=alt.Scale(#domain=['Agriculture_Female','Industry_Female','Service_Female'],
                              range= color_category)),
    alt.Order('key:N', sort='ascending'),
    alt.Tooltip('value:Q',format='.1f')
).properties(
    width = 750,
    height = 120,
    title = 'Share of Female Employment in Sectors(%)'
)


text = alt.Chart(jDF).mark_text(color='white',align='center',dx=-14,dy=0,fontSize=11
).transform_filter(
    selectYear
).transform_fold(
    ['Agriculture_Female','Industry_Female','Service_Female']
).transform_filter(alt.FieldOneOfPredicate(field='Country',
                                           oneOf=['India','Azerbaijan','United States',
                                                  'Afghanistan','Yemen, Rep.'])
).encode(
    alt.Y('Country:N',sort=['Afghanistan','Yemen, Rep.','India','Azerbaijan','United States']),
    alt.X('value:Q', stack='zero', scale= alt.Scale(domain=[0,100])),
    alt.Text('value:N',format='.1f'),
    alt.Order('key:N', sort='ascending'),
)


stackedbarsector = alt.layer(
    stackedbar,text
).resolve_scale(
    color='independent'
)
```

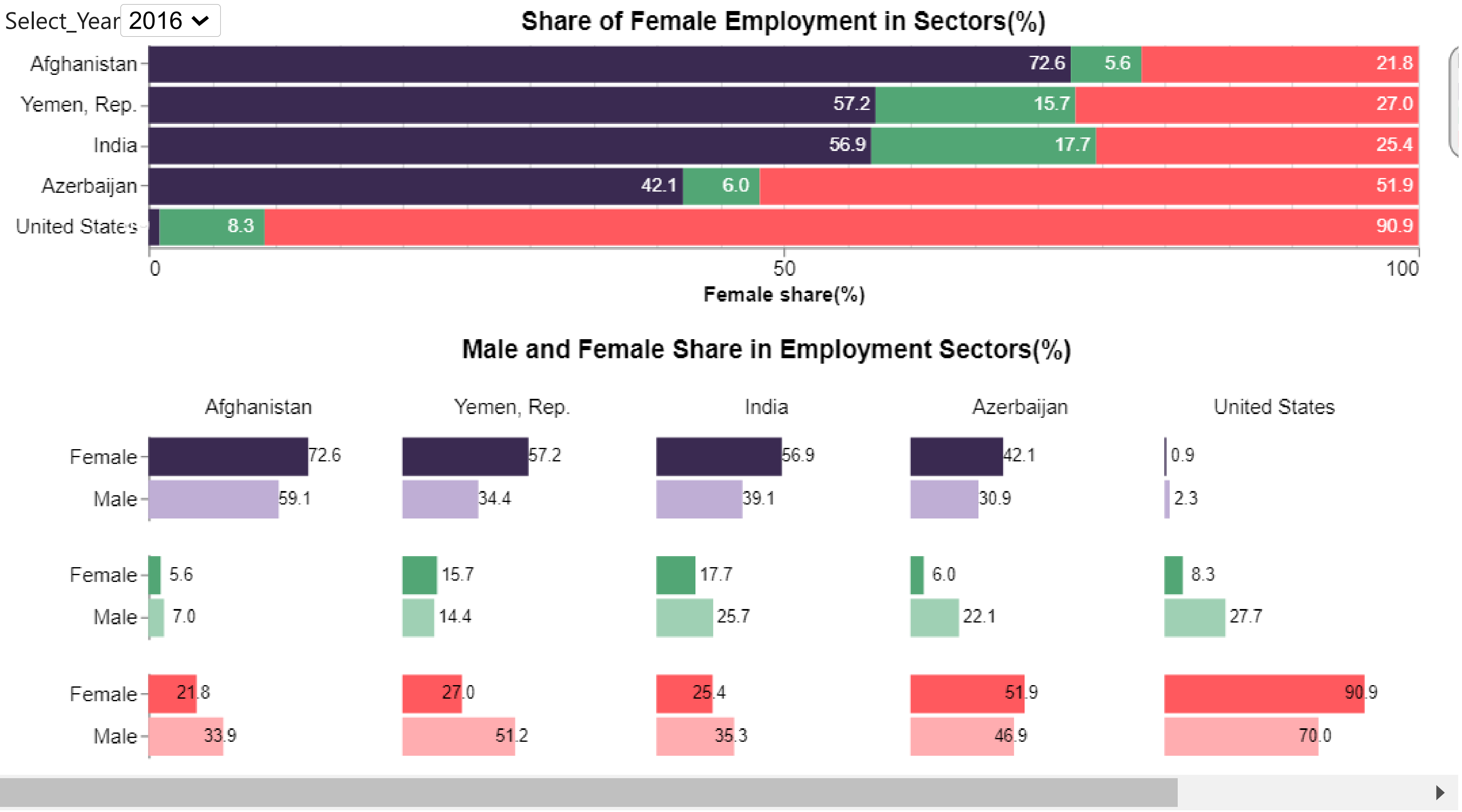
In [16]:

```
agri = alt.layer(
  alt.Chart().mark_bar().transform_fold(
    ['Agriculture_Male','Agriculture_Female']
    ).encode(
        alt.Y('key:N',stack='zero',axis=alt.Axis(labelExpr=axis_labels), title = None),
        alt.X('value:Q',
              title = None, axis=None,
           #    axis=alt.Axis(tickMinStep = 100),
               scale=alt.Scale(domain=[0,100])),
        alt.Color('key:N',scale=alt.Scale(range=color_category1_light),legend=None),
      alt.Tooltip('value:Q',format='.1f')

      )
    ,
  alt.Chart().mark_text(color='black',align='center',dx=9.5,dy=0,fontSize=10
    ).transform_fold(
      ['Agriculture_Male','Agriculture_Female']
    ).encode(
        alt.Y('key:N',stack='zero', title = None),
        alt.X('value:Q',stack='zero', title = None),
        alt.Text('value:N',format='.1f')
    )
).properties(
     width = 130,
    height = 50
).facet(
  data=jDF,
     columns=5,
  column =alt.Column('Country:N', title='Male and Female Share in Employment Sectors(%)',
                     header=alt.Header(titleFontSize=15, labelFontSize=12),
                     sort=['Afghanistan','Yemen, Rep.','India','Azerbaijan','United States'])
)

indu = alt.layer(
  alt.Chart().mark_bar().transform_fold(
    ['Industry_Male','Industry_Female']
    ).encode(
        alt.Y('key:N',stack='zero', axis=alt.Axis(labelExpr=axis_labels),title = None),
        alt.X('value:Q',title = None, axis=None,
#               axis=alt.Axis(tickMinStep = 100),
           scale=alt.Scale(domain=[0,100])),
        alt.Color('key:N',scale=alt.Scale(range=color_category2_light),legend=None),
      alt.Tooltip('value:Q',format='.1f')
      )
    ,
  alt.Chart().mark_text(color='black',align='center',dx=12,dy=0,fontSize=10
    ).transform_fold(
    ['Industry_Male','Industry_Female']
    ).encode(
        alt.Y('key:N',stack='zero', title = None),
        alt.X('value:Q',stack='zero', title = None),
        alt.Text('value:N',format='.1f')
    )
).properties(
     width = 130,
    height = 50
).facet(
  data=jDF,
     columns=5,
  column =alt.Column('Country:N',title=None,header=alt.Header(labels=False),
                     sort=['Afghanistan','Yemen, Rep.','India','Azerbaijan','United States'])
)

serv = alt.layer(
  alt.Chart().mark_bar().transform_fold(
    ['Service_Male','Service_Female']
    ).encode(
        alt.Y('key:N',stack='zero', axis=alt.Axis(labelExpr=axis_labels),title = None),
        alt.X('value:Q',title = None, axis=None,
               #axis=alt.Axis(tickMinStep = 100),
              scale=alt.Scale(domain=[0,100])),
        alt.Color('key:N',scale=alt.Scale(range=color_category3_light),legend=None),
      alt.Tooltip('value:Q',format='.1f')
    )
    ,
  alt.Chart().mark_text(color='black',align='center',dx=-2,dy=0,fontSize=10,
      ).transform_fold(
    ['Service_Male','Service_Female']
    ).encode(
        alt.Y('key:N',stack='zero', title = None),
        alt.X('value:Q',stack='zero'),
        alt.Text('value:N',format='.1f')
    )
).properties(
     width = 130,
    height = 50,
).facet(
  data=jDF,
     columns=5,
  column =alt.Column('Country:N',title=None,header=alt.Header(labels=False),
                     sort=['Afghanistan','Yemen, Rep.','India','Azerbaijan','United States'])

)
```

In [17]:

```
employment_sector = alt.vconcat(stackedbarsector , agri , indu, serv
).resolve_scale(
    color='independent'
).transform_filter(
    alt.FieldOneOfPredicate(field='Country', oneOf=['Afghanistan','India','Azerbaijan','United States','Yemen, Rep.'])
).add_selection(selectYear).transform_filter(selectYear
).configure_title(
    anchor='middle',
    fontSize = 15
).configure_axis(
    labelFontSize=12,
     titleFontSize=12
).configure_legend(
    labelFontSize=12,
    titleFontSize =12,
    strokeColor='gray',
    fillColor='#EEEEEE',
    padding=5,
    cornerRadius=10,
    orient='bottom-right'
).configure_view(stroke=None)
employment_sector
```

Out[17]:



## What is the share of women in Parliament seats?

In [18]:

```
parliament.head()
```

Out[18]:

| | Year | Azerbaijan | Afghanistan | India | Yemen, Rep. | United States | World |
|---|---|---|---|---|---|---|---|
| **0** | 2020 | 17.355372 | 27.016129 | 14.364641 | 0.332226 | 27.464789 | 25.580431 |
| **1** | 2019 | 16.806723 | 27.868852 | 14.391144 | 0.332226 | 23.433875 | 24.636604 |
| **2** | 2018 | 16.800000 | NaN | 11.808118 | 0.000000 | 23.502304 | 24.097878 |
| **3** | 2017 | 16.800000 | 27.710843 | 11.808118 | 0.000000 | 19.354839 | 23.590337 |
| **4** | 2016 | 16.800000 | 27.710843 | 11.970534 | 0.000000 | 19.168591 | 23.091367 |

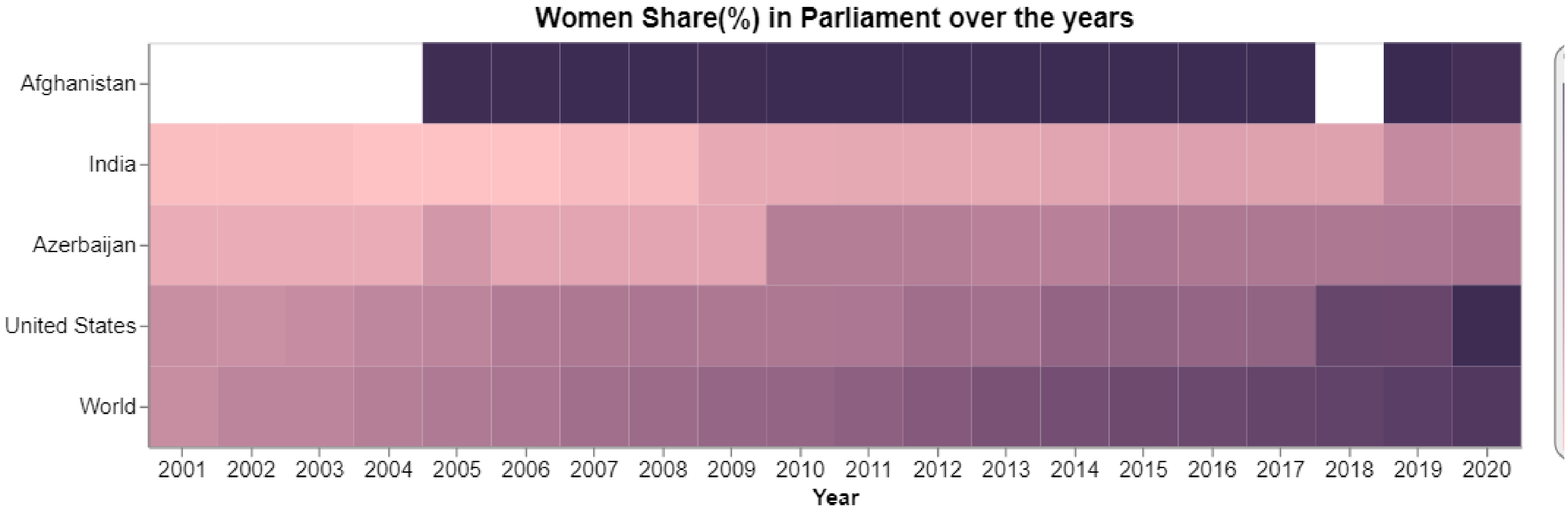
In [19]:

```
line = alt.Chart(parliament).mark_line(point=True).transform_fold(
     ['Azerbaijan','United States','India','Afghanistan','World']).encode(
    alt.X('Year:N', stack=None),
    alt.Y('value:Q',
          impute=alt.ImputeParams(method='mean'),
          axis=alt.Axis(tickMinStep = 5),
          scale=alt.Scale(domain=[0,30]),
          title = '% of Women in Parliament'),
    alt.Color('key:N'),
    alt.Tooltip('value:Q')
).properties(
    title ='Women % in Parliament over the years',
    width=700
)
```

In [20]:

```
parl_hm = alt.Chart(parliament).mark_rect().transform_fold(
     ['Azerbaijan','United States','India','Afghanistan','World']).encode(
    alt.X('Year:N'),
    alt.Y('key:N',sort=['Afghanistan','India','Azerbaijan','United States','World'], title=None),
    alt.Color('value:Q',
              scale=alt.Scale(range=heatmap1),
              legend=alt.Legend(orient='right', titleOrient='top',
                                title='%')),
    tooltip= alt.Tooltip('value:Q', format='.1f')
    #alt.Size('value:Q')
).properties(
    width= 750,
    height=220,
    title ='Women Share(%) in Parliament over the years'
).transform_filter(
    'datum.Year > 2000'
).configure_title(
    anchor='middle',
    fontSize = 15
).configure_axis(
    labelFontSize=12,
    labelAngle=0,
     titleFontSize=12
).configure_legend(
    labelFontSize=9,
    titleFontSize =12,
    strokeColor='gray',
    fillColor='#EEEEEE',
    padding=5,
    cornerRadius=10,
    orient='bottom-right'
)
parl_hm
```

Out[20]:



## What percentage of parents return to the workforce after having a child?

In [21]:
```
labor_parent=lp[:4]
labor_parent = labor_parent.rename(columns={"Age of youngest child ":"child_age"})
```

In [22]:
```
labor_parent=pd.melt(labor_parent,id_vars=['child_age'],var_name='metrics', value_name='values')
labor_parent.head()
```

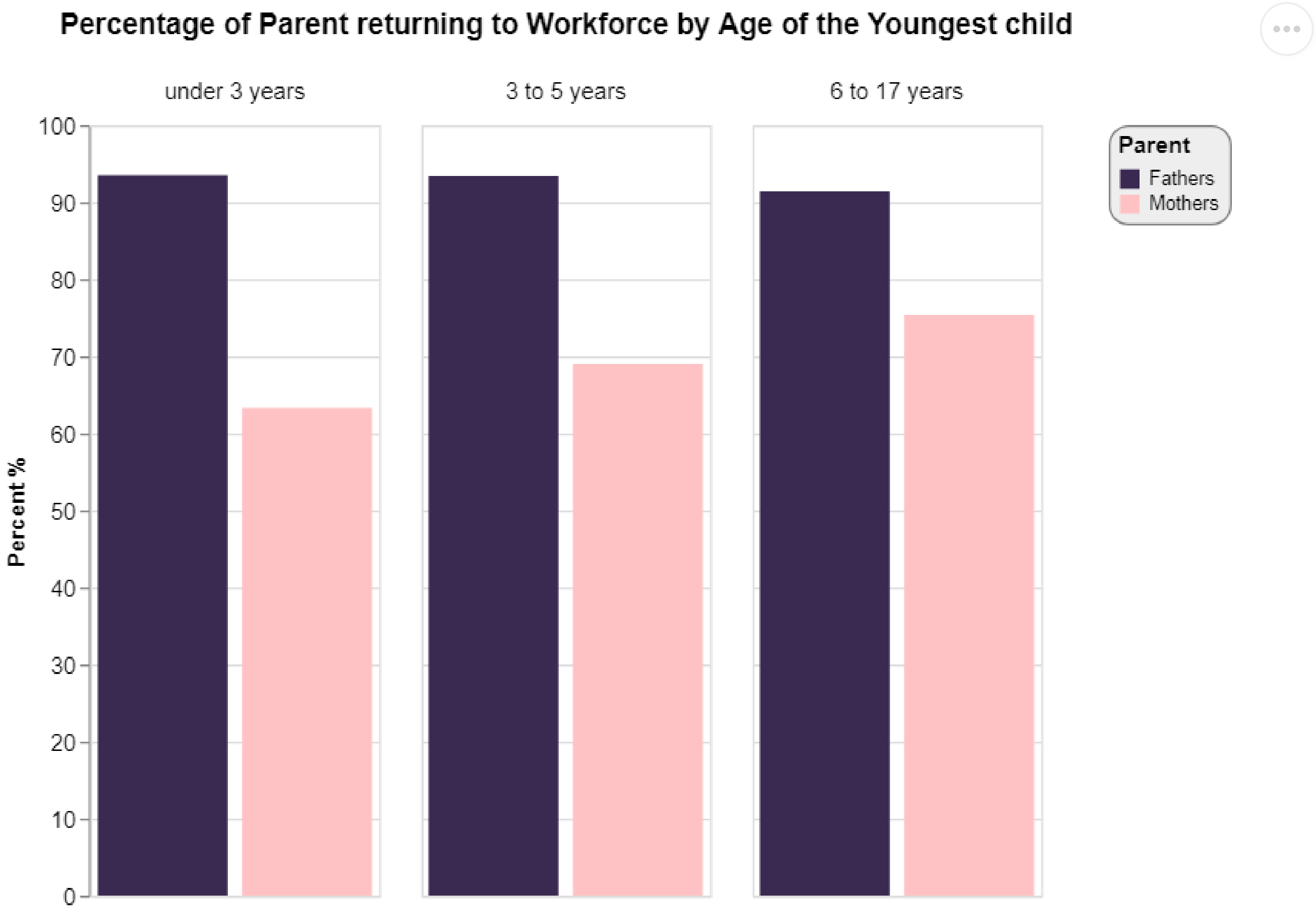
Out[22]:

| | child_age | metrics | values |
|---|---|---|---|
| **0** | under 3 years | Mothers | 63.3 |
| **1** | 3 to 5 years | Mothers | 69.0 |
| **2** | 6 to 17 years | Mothers | 75.4 |
| **3** | under 18 years | Mothers | 71.2 |
| **4** | under 3 years | Fathers | 93.5 |

In [23]:

```
parentperc = alt.Chart(labor_parent).mark_bar().encode(
    alt.Y('values:Q', title='Percent %'),
    x = alt.X("metrics:N", title=None, axis=None),
    color=alt.Color('metrics:N', scale=alt.Scale(range =heatmap), title='Parent'),
    tooltip = ['values'],
    column=alt.Column('child_age:N',title=("Percentage of Parent returning to Workforce by Age of the Youngest child"),
                      sort=["under 3 years", "3 to 5 years", "6 to 17 years","under 18 years"])
).transform_filter(
    'datum.child_age != "under 18 years"'
).properties(
    height = 400,
    width=150
).configure_axis(
    labelFontSize=12,
    titleFontSize=12
).configure_title(
    anchor='middle',
    fontSize = 15
).configure_header(
    titleFontSize=15,
    labelFontSize=12
).configure_legend(
    labelFontSize=10,
    titleFontSize =12,
    strokeColor='gray',
    fillColor='#EEEEEE',
    padding=5,
    cornerRadius=10,
    orient='right'
)

parentperc
```

Out[23]:



**Comparing population with advanced degree, which countries have higher male employment over female? Does the employment rate with advanced degrees give equal opportunity?**

In [24]:

```
#Map Plot Data Manipulation
wdf = world_data.rename(columns = {"Indicator Name":"Variables"})

list_of_values = [
                "Labor force with advanced education, female (% of female working-age population with advanced educati
                "Labor force with advanced education, male (% of male working-age population with advanced education)"
test = wdf[wdf['Variables'].isin(list_of_values)]

test_small = test.reset_index()
test_small = test_small.drop(columns = ['Indicator Code', 'index','Unnamed: 66'])
```

```
wdf_small = test_small.pivot(index='Variables', columns=['Country Name', 'Country Code']).T

wdf_small = wdf_small.rename(columns={
                        "Labor force with advanced education, female (% of female working-age population with advanced e
                         "Labor force with advanced education, male (% of male working-age population with advanced edu

                         )

wdf_small.reset_index(inplace=True)

wdf_small = wdf_small.rename(columns={'level_0':'Year',
                         "Country Name":"Country",
                         "Country Code":"CODE"})

# first let us merge geopandas data with our data
# 'naturalearth_lowres' is geopandas datasets so we can use it directly
world = geopandas.read_file(geopandas.datasets.get_path('naturalearth_lowres'))

# rename the columns so that we can merge with our data
world.columns=['pop_est', 'continent', 'name', 'CODE', 'gdp_md_est', 'geometry']
# then merge with our data
merge=pd.merge(world,wdf_small,on='CODE')

list_of_years = merge['Year'].unique()
merge['Year'] = pd.to_numeric(merge['Year'])
#we are focusing on years after 1990, there are almost no missing data
merge[merge['Year'] > 1990]['advanced_female_empl'].isna().mean()

#gender gap would represent difference between male and female labor force with advaced degree(%)
#negative means male with the advanced degree have higher empl rate than female
merge['gender_gap']= merge['advanced_female_empl'] - merge['advanced_male_empl']

merge.head()
```

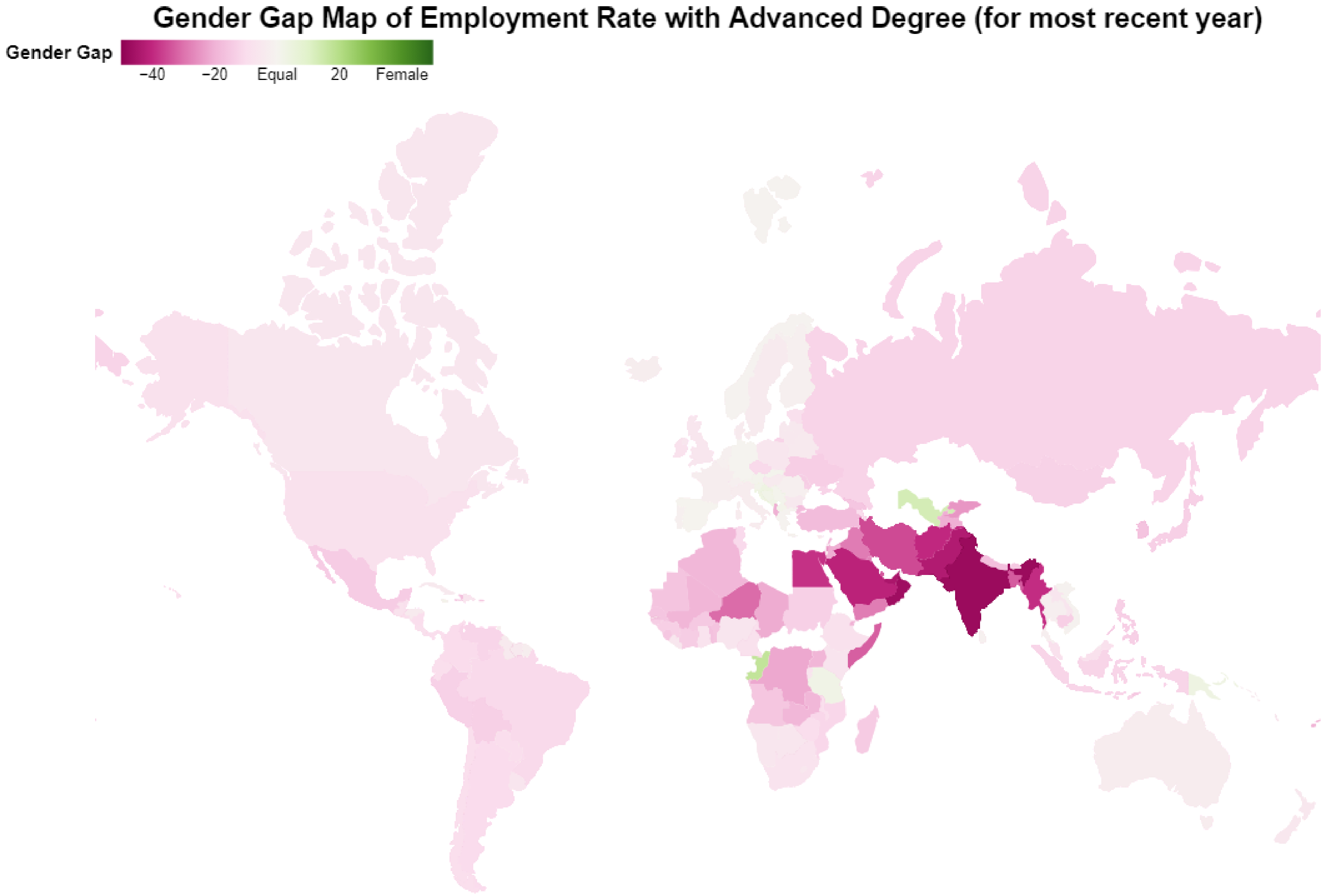
Out[24]:

| | pop_est | continent | name | CODE | gdp_md_est | geometry | Year | Country | advanced_female_empl | advanced_male_empl | gender_gap |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... | 1960 | Fiji | NaN | NaN | NaN |
| **1** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... | 1961 | Fiji | NaN | NaN | NaN |
| **2** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... | 1962 | Fiji | NaN | NaN | NaN |
| **3** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... | 1963 | Fiji | NaN | NaN | NaN |
| **4** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... | 1964 | Fiji | NaN | NaN | NaN |

In [25]:

```
legend_labels1 = ("-1 * datum.label == 40 ? 'Male' :  datum.label == 40 ? 'Female' : datum.label == 0 ? 'Equal' : datum


#World Map Plot
EmploymentGap_Map = ( alt.Chart(merge).mark_geoshape(
).encode(
    color=alt.Color('gender_gap:Q',
                   scale=alt.Scale(domain=[-50, 50],
                       scheme='pinkyellowgreen'),
                   #range = wage_map_color),
                    legend=alt.Legend(orient='top', titleOrient='left',
                               title='Gender Gap',labelExpr=legend_labels1
                                    ),
                   #title = ['green - women','purple - men','yellow - equal']
                   ),
    tooltip = ["name" , alt.Tooltip('advanced_female_empl:Q', format='.2'),
              alt.Tooltip('advanced_male_empl', format='.2')]
)).properties(
    width=900,
    height=500,
    title  = "Gender Gap Map of Employment Rate with Advanced Degree (for most recent year)"
).configure_view(
    stroke=None
).configure_title(fontSize=18, anchor="middle"
                 ).configure_legend(titleColor='black', titleFontSize=12)
EmploymentGap_Map
```

Out[25]:



Gender Gap Map of Employment Rate with Advanced Degree (for most recent year)

### In which occupations women are being paid more than men?

In [26]:

```
#Wage per Occupation Data Manipulation
occupation = pd.read_excel('wage_per_occupation.xlsx', sheet_name="Table 2")
occupation = occupation[3:]
```

```
data=occupation.reset_index()

data = data[4:]

data.columns = ['new_col1','Occupation', 'Number of workers/total', 'Median weekly earnings/total',
                'Standard error of median/total', 'Number of workers/women',
                'Median weekly earnings/women', 'Standard error of median/women',
               'Number of workers/men','Median weekly earnings/men','Standard error of median/men',
                "Women's earnings as a percentage of men's"]
data = data.reset_index()
data = data.drop(columns=['new_col1'])

occup_data = pd.wide_to_long(data,
                             stubnames=['Number of workers', 'Median weekly earnings','Standard error of median'],
                             i='index', j='group',
                             sep='/', suffix=r'\w+')
occup_data = occup_data.reset_index()

occup_data = occup_data.drop(columns=['index'])

occup_data = occup_data.rename(columns={"Women's earnings as a percentage of men's":'women_earn_percentage',
                           "Occupation":"occupation",
                           "Number of workers":'num_work',
                           "Median weekly earnings":'median_week_earn',
                           "Standard error of median":'std_error_med'})

# filter missing/invalid values
occup_data = occup_data[(occup_data['women_earn_percentage'] != '-') & (occup_data['group'] != 'total')]

occup_data.fillna(value = -1, inplace = True)

occup_data = occup_data[(occup_data['occupation']!= -1) & (occup_data['median_week_earn'] != -1) ]
```

In [27]:

```
occup_data
```

Out[27]:

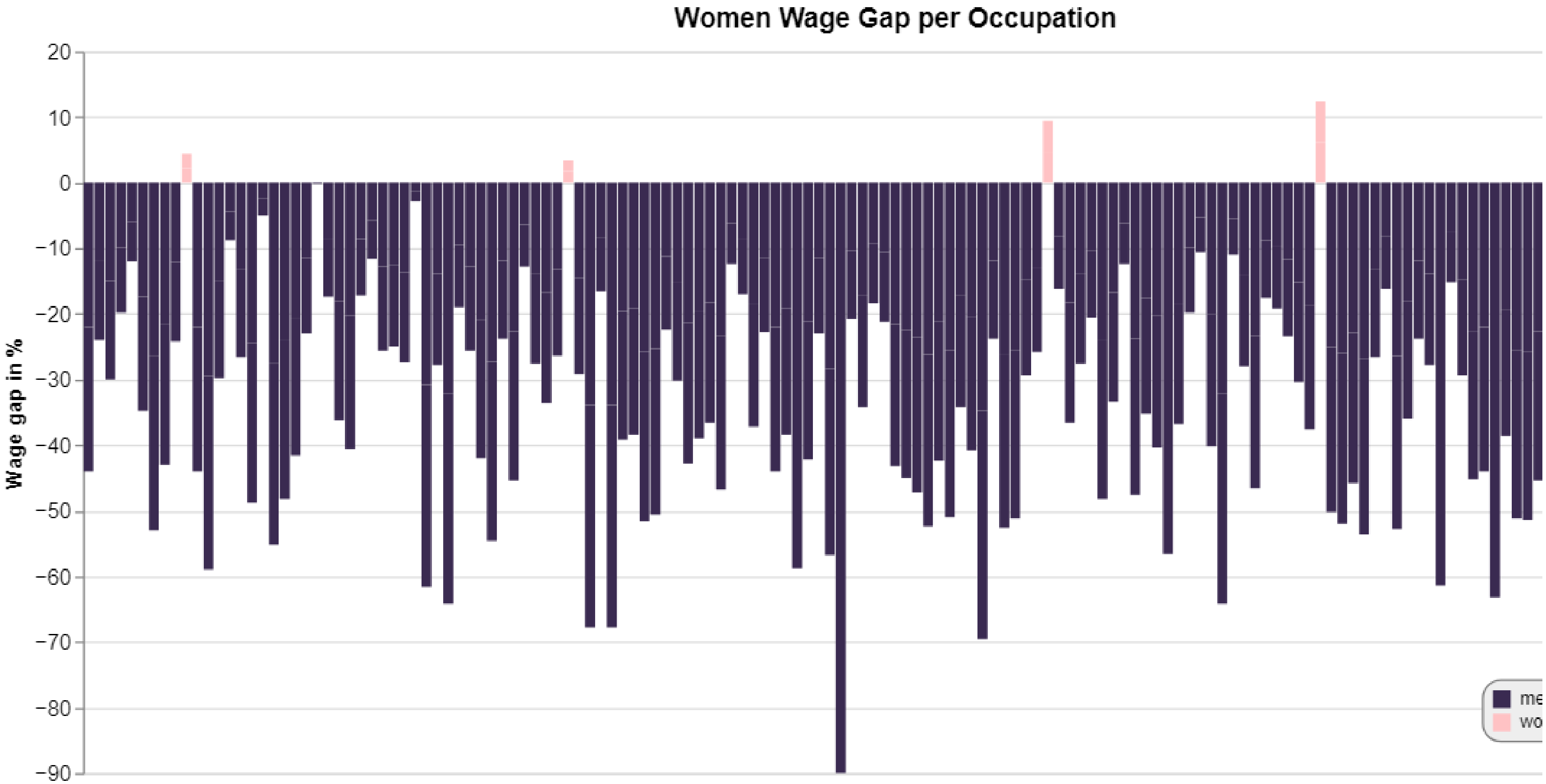
| | group | occupation | women_earn_percentage | num_work | median_week_earn | std_error_med |
|---|---|---|---|---|---|---|
| **598** | women | Management, professional, and related occupations | 73.8 | 25933 | 1164 | 4 |
| **599** | women | Management, business, and financial operations... | 76.4 | 9729 | 1274 | 12 |
| **600** | women | Management occupations | 77.5 | 5747 | 1347 | 12 |
| **601** | women | Chief executives | 75.6 | 363 | 2051 | 91 |
| **602** | women | General and operations managers | 80.5 | 281 | 1241 | 30 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1763** | men | Bus drivers, transit and intercity | 102.2 | 89 | 774 | 54 |
| **1764** | men | Driver/sales workers and truck drivers | 72.7 | 2409 | 916 | 14 |
| **1783** | men | Laborers and freight, stock, and material move... | 88.5 | 1268 | 672 | 9 |
| **1785** | men | Packers and packagers, hand | 90.1 | 205 | 604 | 8 |
| **1786** | men | Stockers and order fillers | 95.7 | 714 | 602 | 8 |

298 rows × 6 columns

In [28]:

```python
# Wage Gap Bar Chart
bar_chart = alt.Chart(occup_data).mark_bar().transform_calculate(
    wage_gap = 'datum.women_earn_percentage - 100',
    gender_high_pay = 'datum.wage_gap > 0 ? "women earn more": "men earn more"'
).encode(
    x=alt.X("occupation:N", title ='Occupation', axis = None),
    y=alt.Y("wage_gap:Q",title ='Wage gap in %'),
    tooltip = ['occupation','women_earn_percentage'],
    color=alt.Color('gender_high_pay:N', scale=alt.Scale(range =heatmap), title=None)

).properties(title = 'Women Wage Gap per Occupation',width=1000)




bar_chart_wage_gap = bar_chart.properties(
    height = 400,
    width=900
).configure_axis(
    labelFontSize=12,
    titleFontSize=12
).configure_title(
    anchor='middle',
    fontSize = 15
).configure_header(
    titleFontSize=15,
    labelFontSize=12
).configure_legend(
    labelFontSize=10,
    titleFontSize =12,
    strokeColor='gray',
    fillColor='#EEEEEE',
    padding=5,
    cornerRadius=10,
    orient='bottom-right'
)
bar_chart_wage_gap
```

Out[28]:



In [29]:

```
occup_data
```

Out[29]:

| | group | occupation | women_earn_percentage | num_work | median_week_earn | std_error_med |
|---|---|---|---|---|---|---|
| **598** | women | Management, professional, and related occupations | 73.8 | 25933 | 1164 | 4 |
| **599** | women | Management, business, and financial operations... | 76.4 | 9729 | 1274 | 12 |
| **600** | women | Management occupations | 77.5 | 5747 | 1347 | 12 |
| **601** | women | Chief executives | 75.6 | 363 | 2051 | 91 |
| **602** | women | General and operations managers | 80.5 | 281 | 1241 | 30 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1763** | men | Bus drivers, transit and intercity | 102.2 | 89 | 774 | 54 |
| **1764** | men | Driver/sales workers and truck drivers | 72.7 | 2409 | 916 | 14 |
| **1783** | men | Laborers and freight, stock, and material move... | 88.5 | 1268 | 672 | 9 |
| **1785** | men | Packers and packagers, hand | 90.1 | 205 | 604 | 8 |
| **1786** | men | Stockers and order fillers | 95.7 | 714 | 602 | 8 |

298 rows × 6 columns

In [30]:

```
# Weekly Pay Gap per Occupation
men_wage_occ = alt.Chart(occup_data).encode(
    alt.Y('occupation:N'),
    alt.X('median_week_earn:Q'),
    color = alt.Color('group',scale=alt.Scale(range =color_two_category)),
    shape = 'group',
    tooltip = ['median_week_earn', 'occupation']
).transform_filter("datum.group != 'total'"and ('datum.women_earn_percentage < 100')).properties(width=200)

women_wage_occ = alt.Chart(occup_data).encode(
    alt.Y('occupation:N', title = None),
    alt.X('median_week_earn:Q', stack = True),
    color = alt.Color('group',scale=alt.Scale(range =color_two_category)),
    shape = 'group',
    tooltip = ['median_week_earn', 'occupation']
).transform_filter(("datum.group != 'total'") and ('datum.women_earn_percentage >= 100')
                  ).properties(width=200)

scatter_pay_gap = men_wage_occ.mark_point() | women_wage_occ.mark_point()
```

### What is the Adolescent Fertility Rate and Maternal Mortality rate ? Can there be any relation for factors with enrolment of women into secondary Education?

In [31]:

```
color_5_category =['#3A2A51','#FF7075' ,"#FFD35C",'#52A675',"#FFADB0"] #3 distinct
W = 430
sort_cty=['Yemen, Rep.','Afghanistan','India','Azerbaijan','United States']
```

In [32]:

```
jobs  = pd.read_csv("JobsData.csv")
inequality  = pd.read_csv("gender-inequality-index-from-the-human-development-report.csv")

inequality_cty =inequality[inequality["Entity"].isin(["India","United States"
                                          ,"Yemen, Rep."
                                                   ,"Afghanistan"
                                              ,"Azerbaijan"
                                             ])]
```

In [33]:

```
inequality_2005_2021 = inequality_cty[inequality_cty["Year"]>= 2005]

inequality_2021 =  inequality_cty[inequality_cty["Year"]== 2021]
inequality_2021
inequality_world_2021 = inequality[inequality["Year"]== 2021]
```

In [34]:

```
world = geopandas.read_file(geopandas.datasets.get_path('naturalearth_lowres'))
world.head()
```

Out[34]:

| | pop_est | continent | name | iso_a3 | gdp_md_est | geometry |
|---|---|---|---|---|---|---|
| **0** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... |
| **1** | 58005463.0 | Africa | Tanzania | TZA | 63177 | POLYGON ((33.90371 -0.95000, 34.07262 -1.05982... |
| **2** | 603253.0 | Africa | W. Sahara | ESH | 907 | POLYGON ((-8.66559 27.65643, -8.66512 27.58948... |
| **3** | 37589262.0 | North America | Canada | CAN | 1736425 | MULTIPOLYGON (((-122.84000 49.00000, -122.9742... |
| **4** | 328239523.0 | North America | United States of America | USA | 21433226 | MULTIPOLYGON (((-122.84000 49.00000, -120.0000... |

In [35]:

```
merge_DF = pd.merge(world, inequality_world_2021, left_on='iso_a3', right_on='Code')
merge_DF.columns =['pop_est', 'continent', 'name', 'iso_a3', 'gdp_md_est', 'geometry',
       'Entity', 'Code', 'Year',
       'GDI']
```

```
merge_DF
```

Out[35]:

| | pop_est | continent | name | iso_a3 | gdp_md_est | geometry | Entity | Code | Year | GDI |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 889953.0 | Oceania | Fiji | FJI | 5496 | MULTIPOLYGON (((180.00000 -16.06713, 180.00000... | Fiji | FJI | 2021 | 0.318 |
| **1** | 58005463.0 | Africa | Tanzania | TZA | 63177 | POLYGON ((33.90371 -0.95000, 34.07262 -1.05982... | Tanzania | TZA | 2021 | 0.560 |
| **2** | 37589262.0 | North America | Canada | CAN | 1736425 | MULTIPOLYGON (((-122.84000 49.00000, -122.9742... | Canada | CAN | 2021 | 0.069 |
| **3** | 328239523.0 | North America | United States of America | USA | 21433226 | MULTIPOLYGON (((-122.84000 49.00000, -120.0000... | United States | USA | 2021 | 0.179 |
| **4** | 18513930.0 | Asia | Kazakhstan | KAZ | 181665 | POLYGON ((87.35997 49.21498, 86.59878 48.54918... | Kazakhstan | KAZ | 2021 | 0.161 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **153** | 2083459.0 | Europe | North Macedonia | MKD | 12547 | POLYGON ((22.38053 42.32026, 22.88137 41.99930... | North Macedonia | MKD | 2021 | 0.134 |
| **154** | 6944975.0 | Europe | Serbia | SRB | 51475 | POLYGON ((18.82982 45.90887, 18.82984 45.90888... | Serbia | SRB | 2021 | 0.131 |
| **155** | 622137.0 | Europe | Montenegro | MNE | 5542 | POLYGON ((20.07070 42.58863, 19.80161 42.50009... | Montenegro | MNE | 2021 | 0.119 |
| **156** | 1394973.0 | North America | Trinidad and Tobago | TTO | 24269 | POLYGON ((-61.68000 10.76000, -61.10500 10.890... | Trinidad and Tobago | TTO | 2021 | 0.344 |
| **157** | 11062113.0 | Africa | S. Sudan | SSD | 11998 | POLYGON ((30.83385 3.50917, 29.95350 4.17370, ... | South Sudan | SSD | 2021 | 0.587 |

158 rows × 10 columns

In [36]:

```
GDI_Trend =  ( alt.Chart(inequality_2005_2021).mark_line(
).encode(
    alt.X("Year:N"  )
    ,alt.Y( "Gender Inequality Index:Q")
#     ,column = "Name:N"
#     longitude='longitude:Q', # apply the field named 'longitude' to the longitude channel
#     latitude='latitude:Q'    # apply the field named 'latitude' to the latitude channel
    ,color  =  alt.Color("Entity:N"
    ,  scale = alt.Scale(range = color_5_category)
                       ,sort =  sort_cty)
#     , tooltip = ["name" , "GDI"]
)).properties(
    width=W,
#    / height=500
    title  = "Gender Inequality Index"
)
```

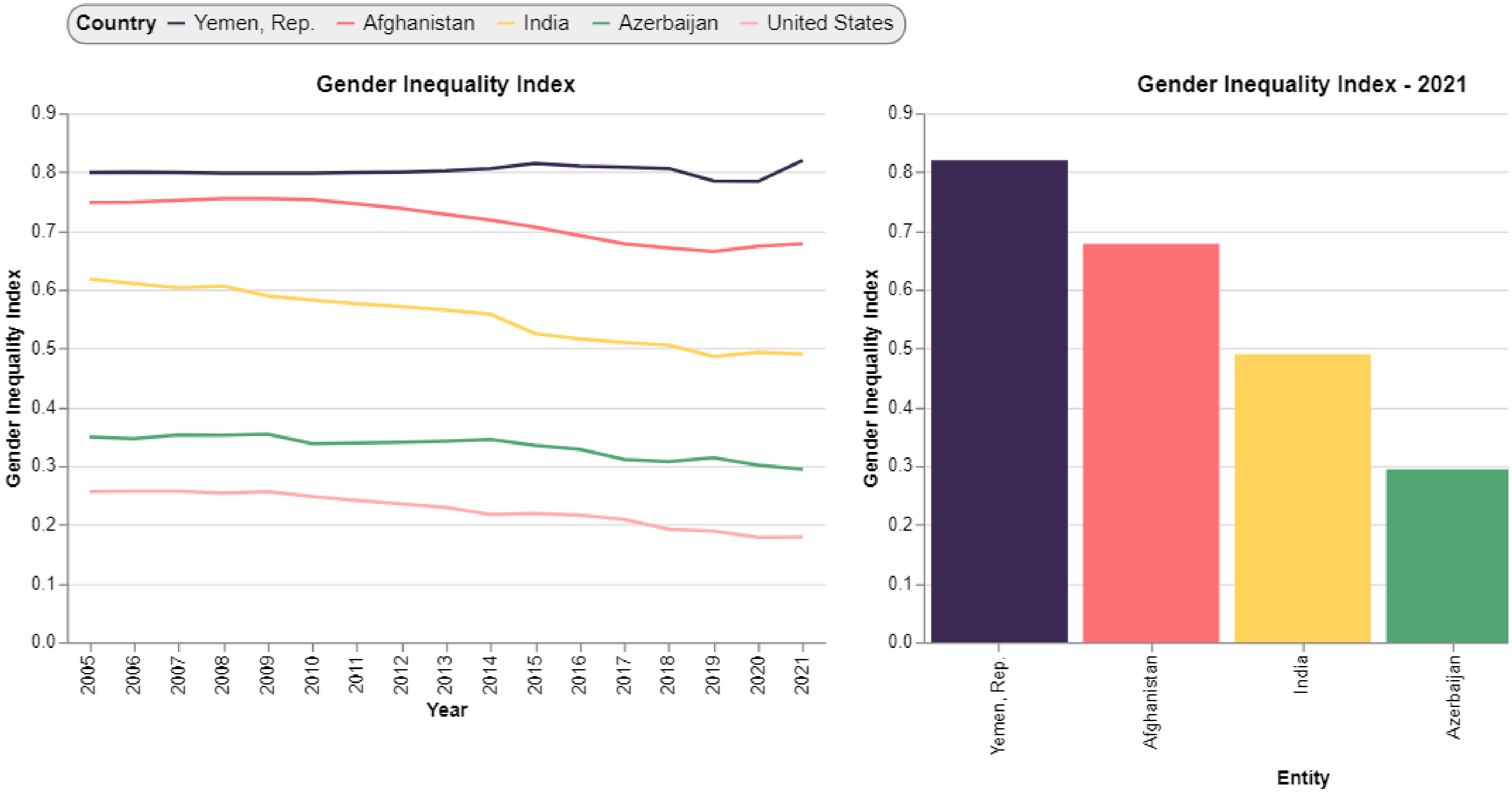
In [37]:

```
GDI_bar =  ( alt.Chart(inequality_2021).mark_bar(
).encode(
    alt.X("Entity:N" ,sort =  sort_cty )
    ,alt.Y( "Gender Inequality Index:Q")
#     ,column = "Name:N"
#     longitude='longitude:Q', # apply the field named 'longitude' to the longitude channel
#     latitude='latitude:Q'    # apply the field named 'latitude' to the latitude channel
    ,color  =  alt.Color("Entity:N"
    ,  scale = alt.Scale(range = color_5_category)
                       ,sort =  sort_cty
                        ,  legend=alt.Legend(orient='top', titleOrient='left',
                            title='Country'
                  ))
#     , tooltip = ["name" , "GDI"]
)).properties(
    width=W,
#    / height=500
    title  = "Gender Inequality Index - 2021"
)
```

In [38]:

```
(GDI_Trend | GDI_bar).configure_view(
    stroke=None
).configure_legend(
    labelFontSize=12,
    titleFontSize =12,
    strokeColor='gray',
    fillColor='#EEEEEE',
    padding=5,
    cornerRadius=10,
    orient='top-right'
)
```

Out[38]:



In [39]:

```
Jobs = jobs[['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
        '2011']]
Jobs.columns = Jobs.columns.astype(str)
Stats5countries =  Jobs[Jobs["Country Name"].isin(sort_cty)]
```

In [40]:

```
Female_secodary_enrolment = Stats5countries[Stats5countries["Indicator Code"].isin([
                                                                     "SE.SEC.ENRR.FE"])]
```

In [41]:

```
Secondary_bar = alt.Chart(Female_secodary_enrolment).mark_line( stroke = "#65605D"  , color = "#1B3727"  ).encode(
alt.X("Country Name:N", title = None ,sort=sort_cty, axis=alt.Axis(labels=False))
, alt.Y("2011:Q" , title = "School enrollment, secondary, female (% gross)" , scale=alt.Scale(domain=[0,100]))
)
```

In [42]:

```
Fertility  = pd.read_csv("Adolescent_fertilirt.csv")
Fertility_2017 = Fertility[Fertility["Year"] == 2017]
Fertility_2017
```

Out[42]:

| | Year | Adolescent fertility rate (births per 1,000 women ages 15-19) | Country |
|---|---|---|---|
| **3** | 2017 | 55.838 | Azerbaijan |
| **24** | 2017 | 68.957 | Afghanistan |
| **45** | 2017 | 60.352 | Yemen, Rep. |
| **66** | 2017 | 13.177 | India |
| **87** | 2017 | 19.860 | United States |

In [43]:

```
fertility_bar = alt.Chart(Fertility_2017).mark_bar().encode(
alt.X("Country:N", title = None ,sort=sort_cty)
, alt.Y("Adolescent fertility rate (births per 1,000 women ages 15-19):Q"
        , title = "Adolescent fertility rate" )
    ,alt.Color("Country:N"  )
 ).transform_filter("datum.Country != 'World'").properties(width =W , title = "Adolescent fertility rate (births per 1,


P1 = (fertility_bar + Secondary_bar.encode(
alt.Y("2011:Q" ,title = None , axis=alt.Axis(labels=False)))).resolve_scale(
    y="independent"
    , x = "independent"
).properties(width =W
            )
```

In [44]:

```
mortality  = pd.read_csv("Maternal_Mortality_ratio.csv")
mortality_2017 = mortality[mortality["Year"] == 2017]
mortality_2017
```

Out[44]:

|  | Year | Country | Maternal mortality ratio (per 100,000 live births) |
|---|---|---|---|
| **0** | 2017 | World | 211 |
| **18** | 2017 | Afghanistan | 638 |
| **36** | 2017 | Azerbaijan | 26 |
| **54** | 2017 | India | 145 |
| **72** | 2017 | Yemen, Rep. | 164 |
| **90** | 2017 | United States | 19 |

In [45]:

```
mortality_bar = alt.Chart(mortality_2017).mark_bar().encode(
alt.X("Country:N", title = None ,sort=sort_cty )
, alt.Y("Maternal mortality ratio (per 100,000 live births):Q"  , title = "Maternal mortality ratio" )
    ,alt.Color("Country:N" , legend = None , scale = alt.Scale(range = color_5_category))
).transform_filter("datum.Country != 'World'").properties(width =W , title = "Maternal mortality ratio (per 100,000 liv
mortality_bar

p2 = (mortality_bar + Secondary_bar
     ).resolve_scale(
    y="independent"
    , x = "independent"
).properties(width =W , title = "Maternal mortality ratio (per 100,000 live births) - 2017")
```

In [46]:

```
mortality_trend = alt.Chart(mortality).mark_line().encode(
alt.X("Year:N")
,alt.Y("Maternal mortality ratio (per 100,000 live births)" , title ="Maternal mortality ratio")
,alt.Color("Country"
                   ,  scale = alt.Scale(range = color_5_category)
           , legend=alt.Legend(orient='top', titleOrient='left',
                                title='Country'
                  ))).transform_filter("datum.Country != 'World'"
                                    ).properties(width =W , title = "Trend of Maternal mortality ratio (per 100,000 liv
```

In [47]:

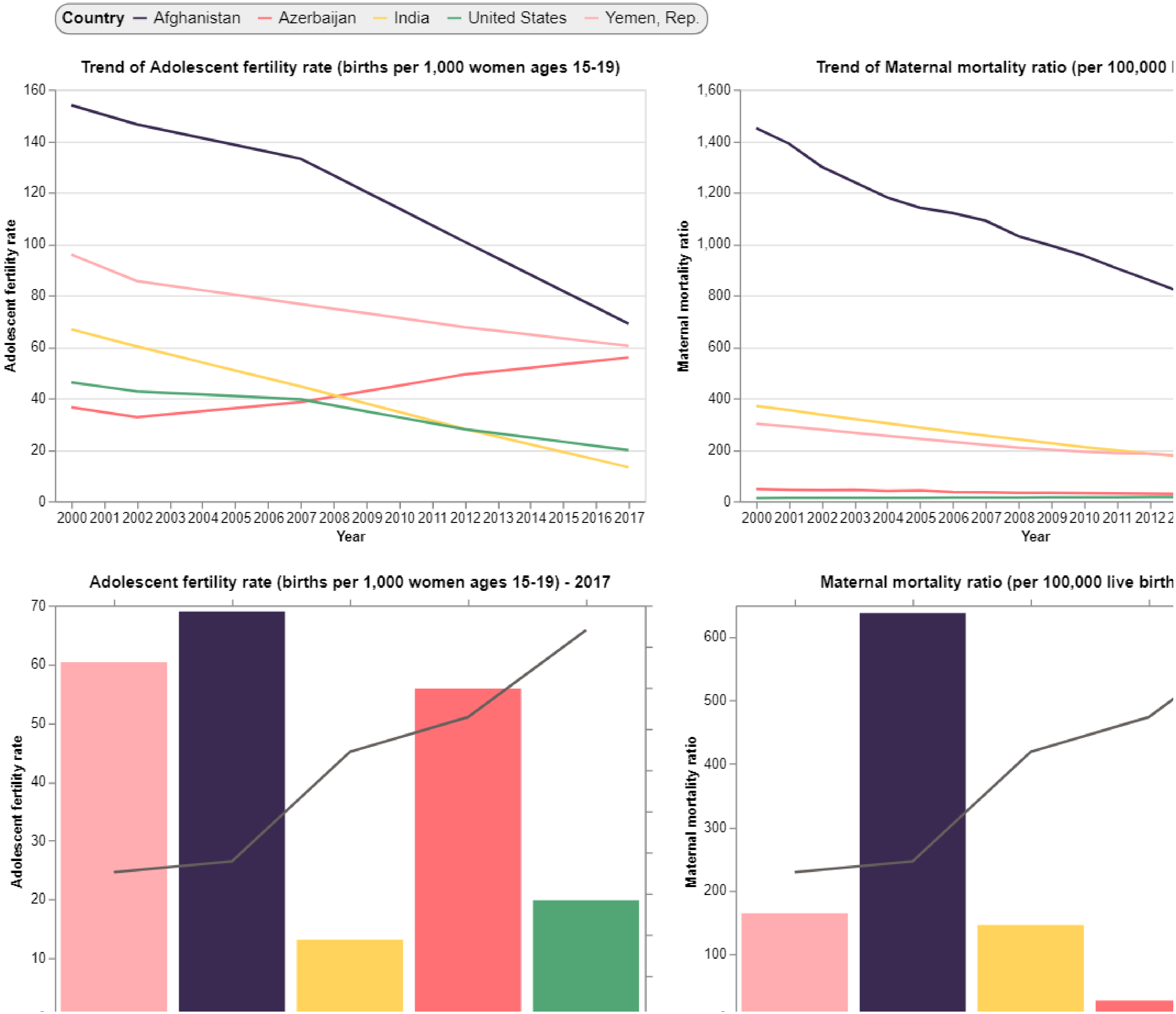
```
Fertility_trend = alt.Chart(Fertility).mark_line(
).encode(
alt.X("Year:N")
,alt.Y("Adolescent fertility rate (births per 1,000 women ages 15-19)"
       , title = "Adolescent fertility rate")
    ,alt.Color("Country")
).transform_filter("datum.Country != 'World'"
```

```
                  ).transform_filter("datum.Year <='2017'"
                                    ).properties(width =W
                                                 , title ="Trend of Adolescent fertility rate (births per 1,000 women ag
```

In [48]:

```
Ferti_Mortalilty = (( Fertility_trend | mortality_trend) & (P1 | p2)
).resolve_scale(color = "independent").configure_legend(
    labelFontSize=12,
    titleFontSize =12,
    strokeColor='gray',
    fillColor='#EEEEEE',
    padding=5,
    cornerRadius=10,
    orient='top-right'
).configure_axis(
    labelFontSize=10,
     titleFontSize=10
    ,labelAngle=0
).configure_title(
    anchor='middle',
    fontSize = 12
)

Ferti_Mortalilty
```

Out[48]:



Trend of Adolescent fertility rate (births per 1,000 women ages 15-19)

Trend of Maternal mortality ratio (per 100,000 l

Adolescent fertility rate (births per 1,000 women ages 15-19) - 2017

Maternal mortality ratio (per 100,000 live birth

In [ ]:

## Summary of all visuals

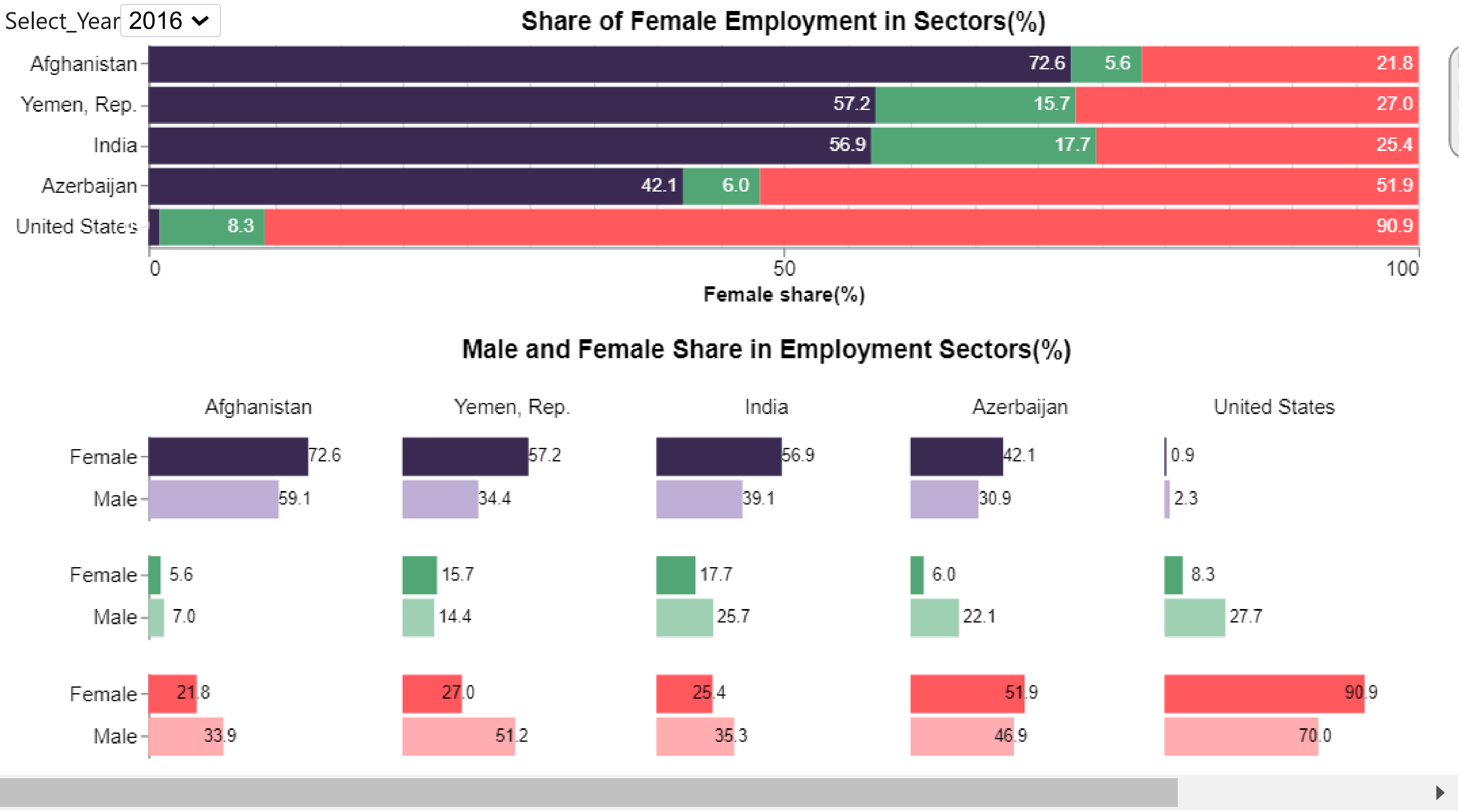
In [ ]:

## What is the share of women employment by sectors?

In [49]:

```
employment_sector
```

Out[49]:

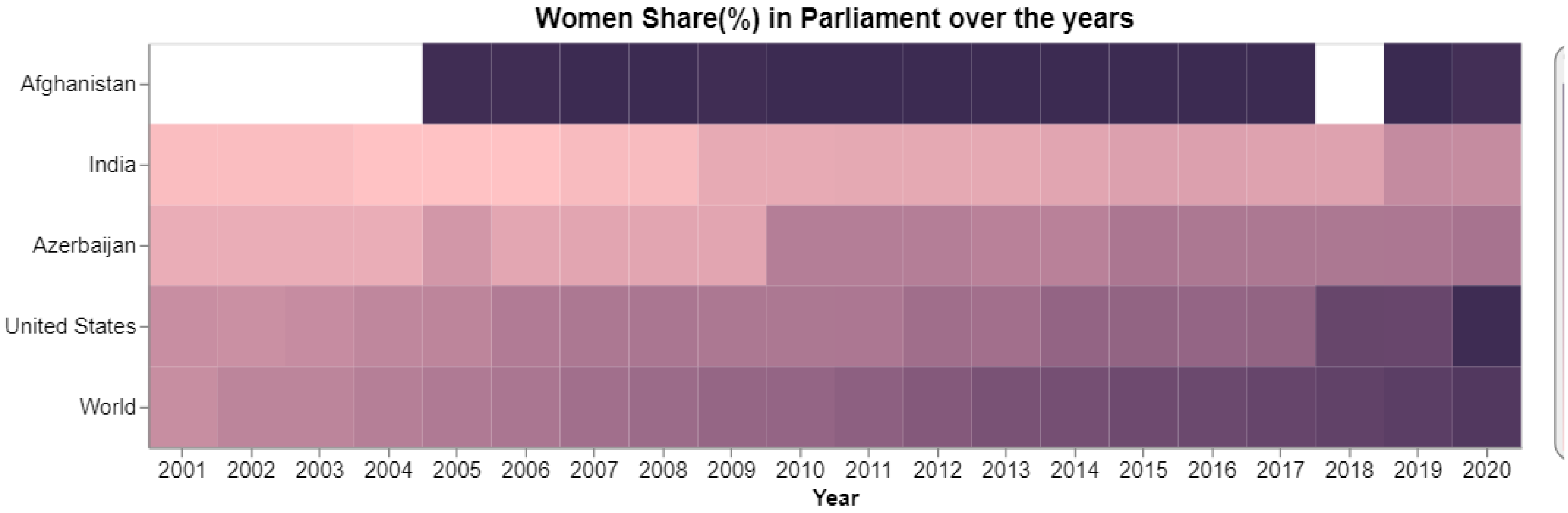


### What is the share of women in Parliament seats?

In [50]:

```
parl_hm
```

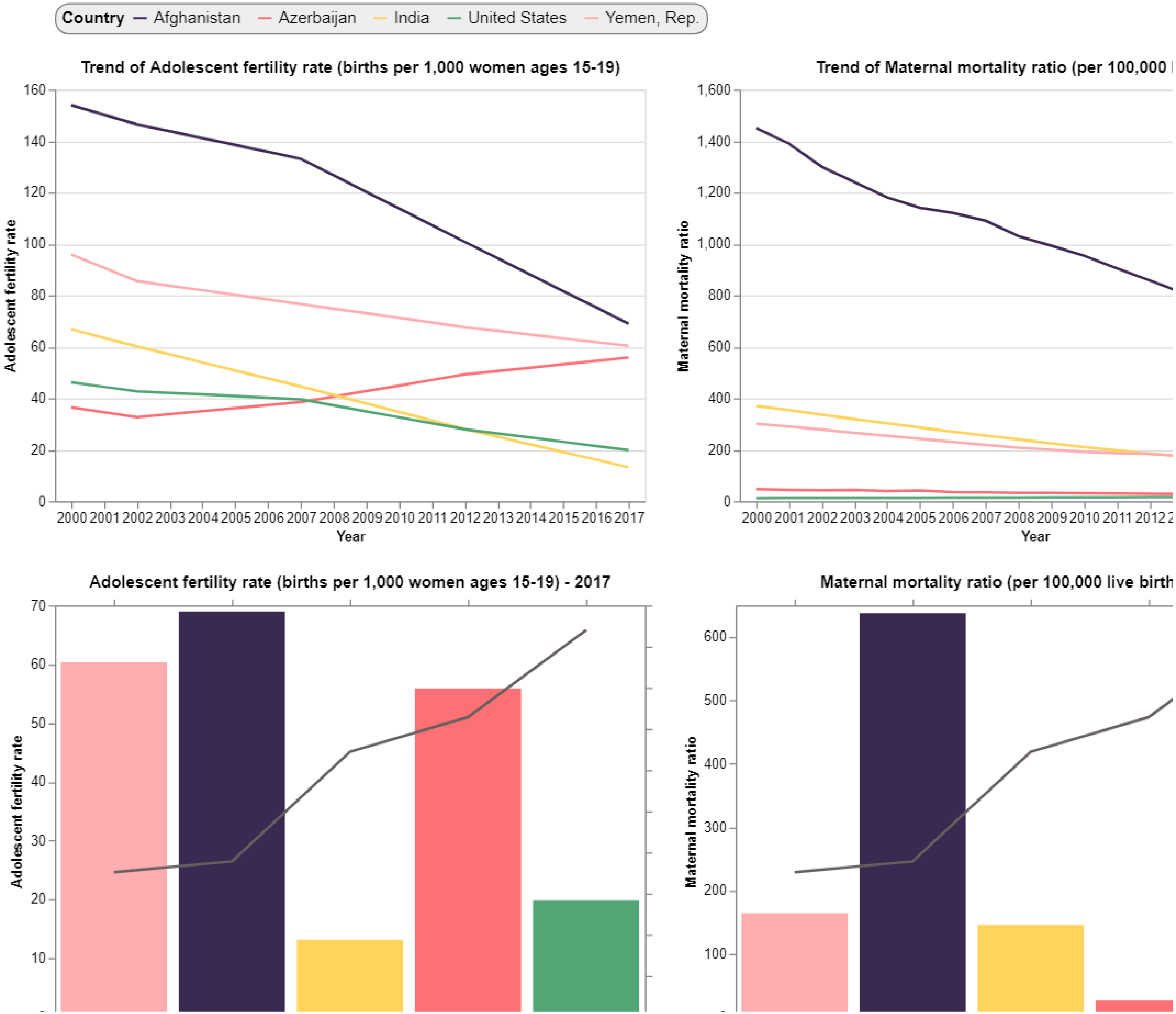
Out[50]:

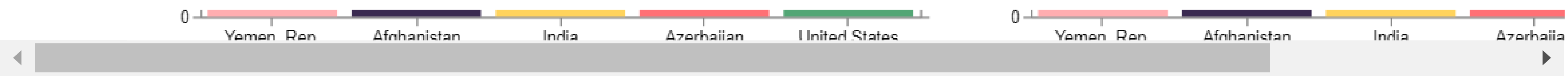


### What is the Adolescent Fertility Rate and Maternal Mortality rate ? Can there be any relation for factors with enrolment of women into secondary Education?

In [51]:
```
Ferti_Mortalilty
```

```
Out[51]:
```


Country
Afghanistan
Azerbaijan
India
United States
Yemen, Rep.
Trend of Adolescent fertility rate (births per 1,000 women ages 15-19)
Adolescent fertility rate
160
140
120
100
80
60
40
20
0
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
Year
Trend of Maternal mortality ratio (per 100,000 l
Maternal mortality ratio
1,600
1,400
1,200
1,000
800
600
400
200
0
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2
Year
Adolescent fertility rate (births per 1,000 women ages 15-19) - 2017
Adolescent fertility rate
70
60
50
40
30
20
10
Maternal mortality ratio (per 100,000 live birth
Maternal mortality ratio
600
500
400
300
200
100

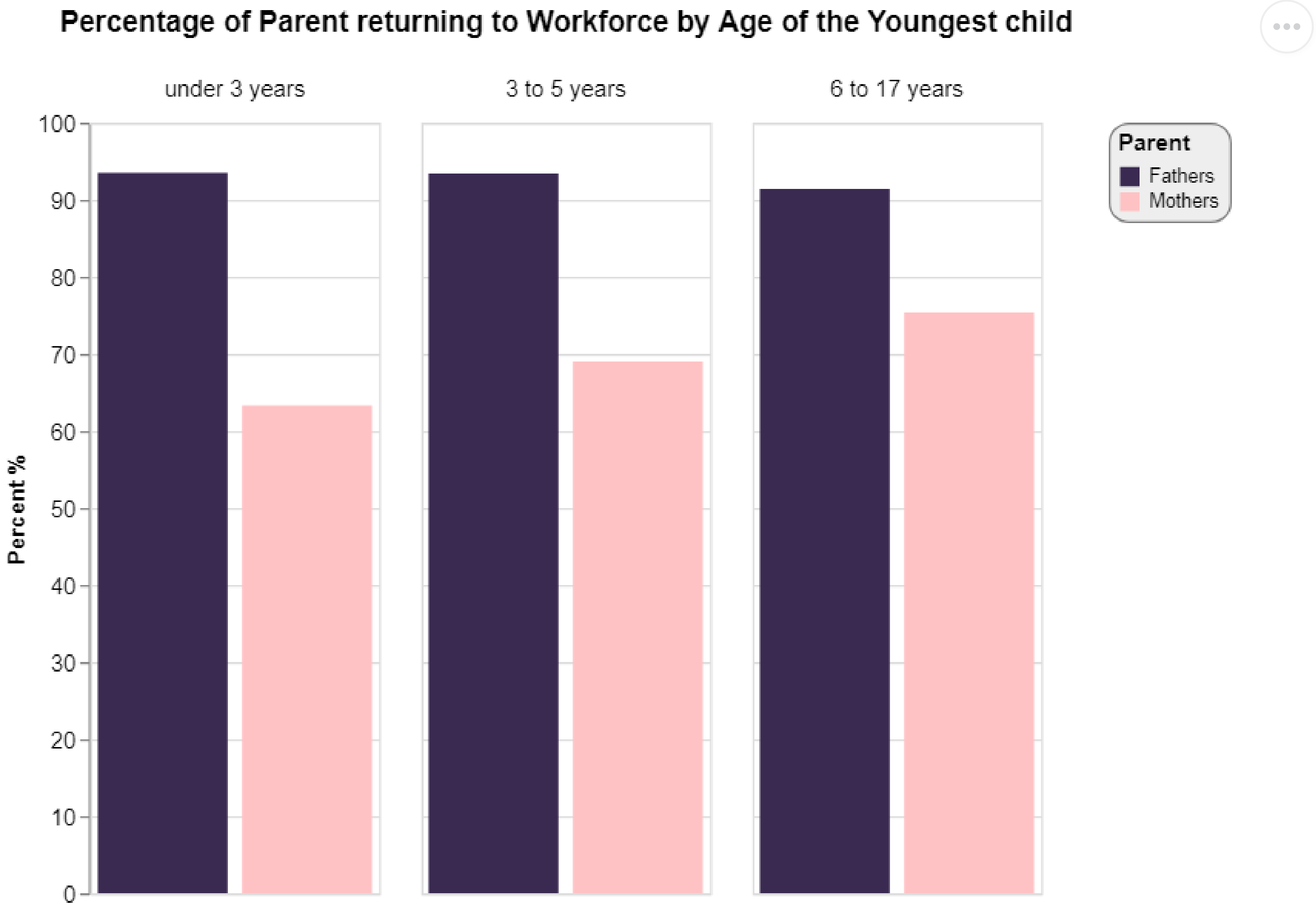


## What percentage of parents return to the workforce after having a child?

In [52]:

```
parentperc
```

Out[52]:

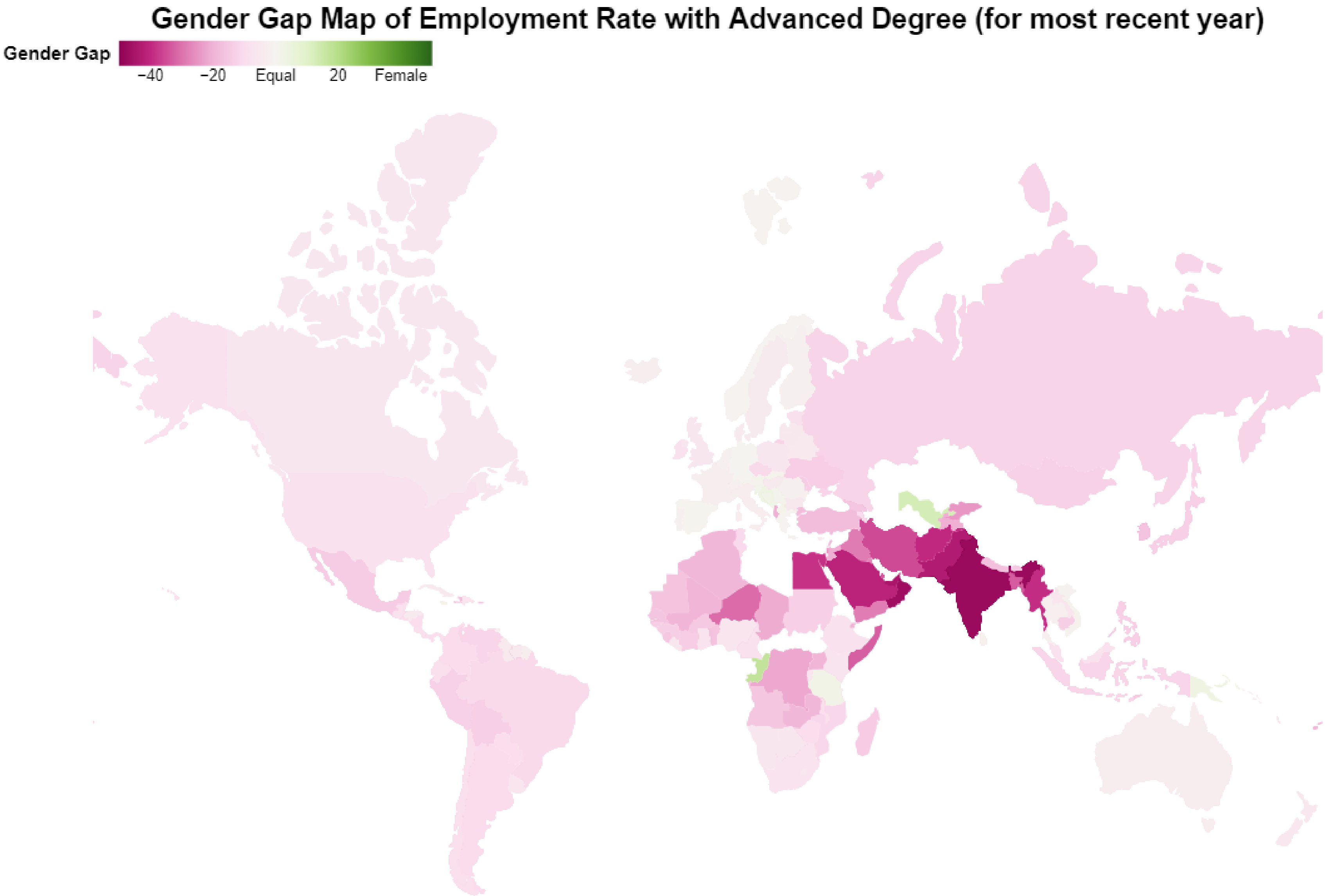


## Comparing population with advanced degree, which countries have higher male employment over female? Does the employment rate with advanced degrees give equal opportunity?

In [53]:

```
EmploymentGap_Map
```

Out[53]:

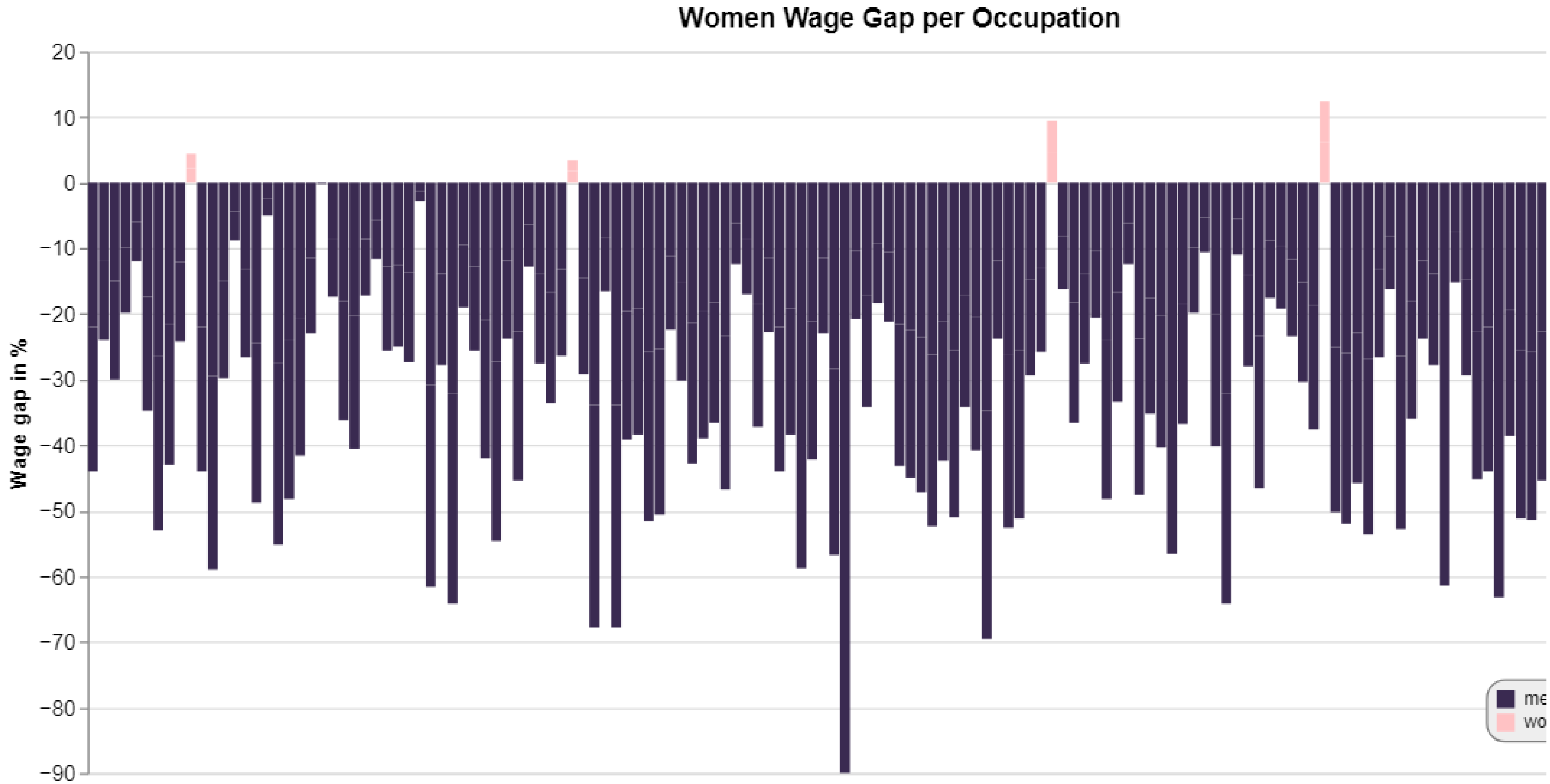

Gender Gap Map of Employment Rate with Advanced Degree (for most recent year)
Gender Gap
−40
−20
Equal
20
Female

## In which occupations women are being paid more than men?

In [54]:

```
bar_chart_wage_gap
```

Out[54]:



In [ ]:

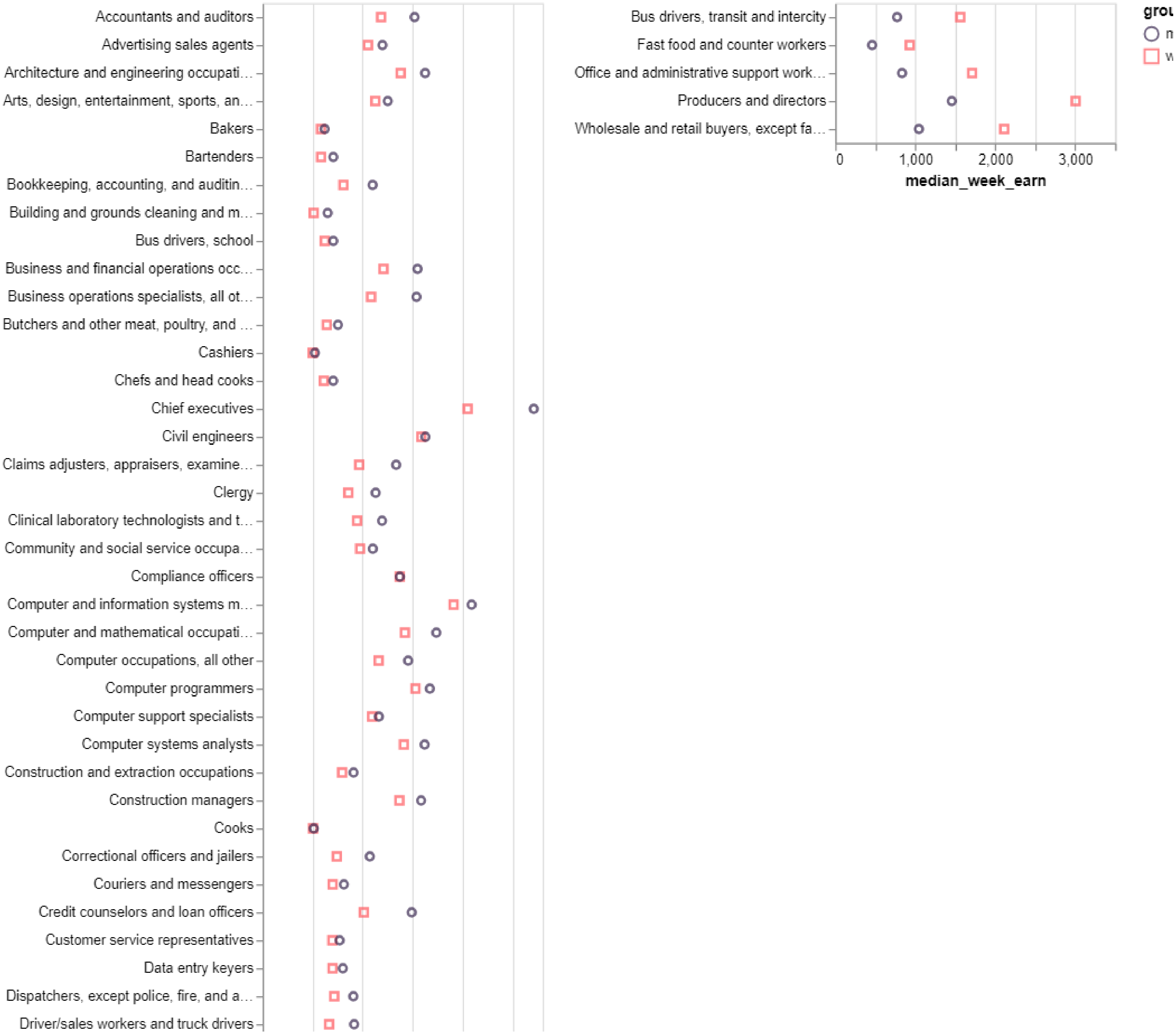
```
```

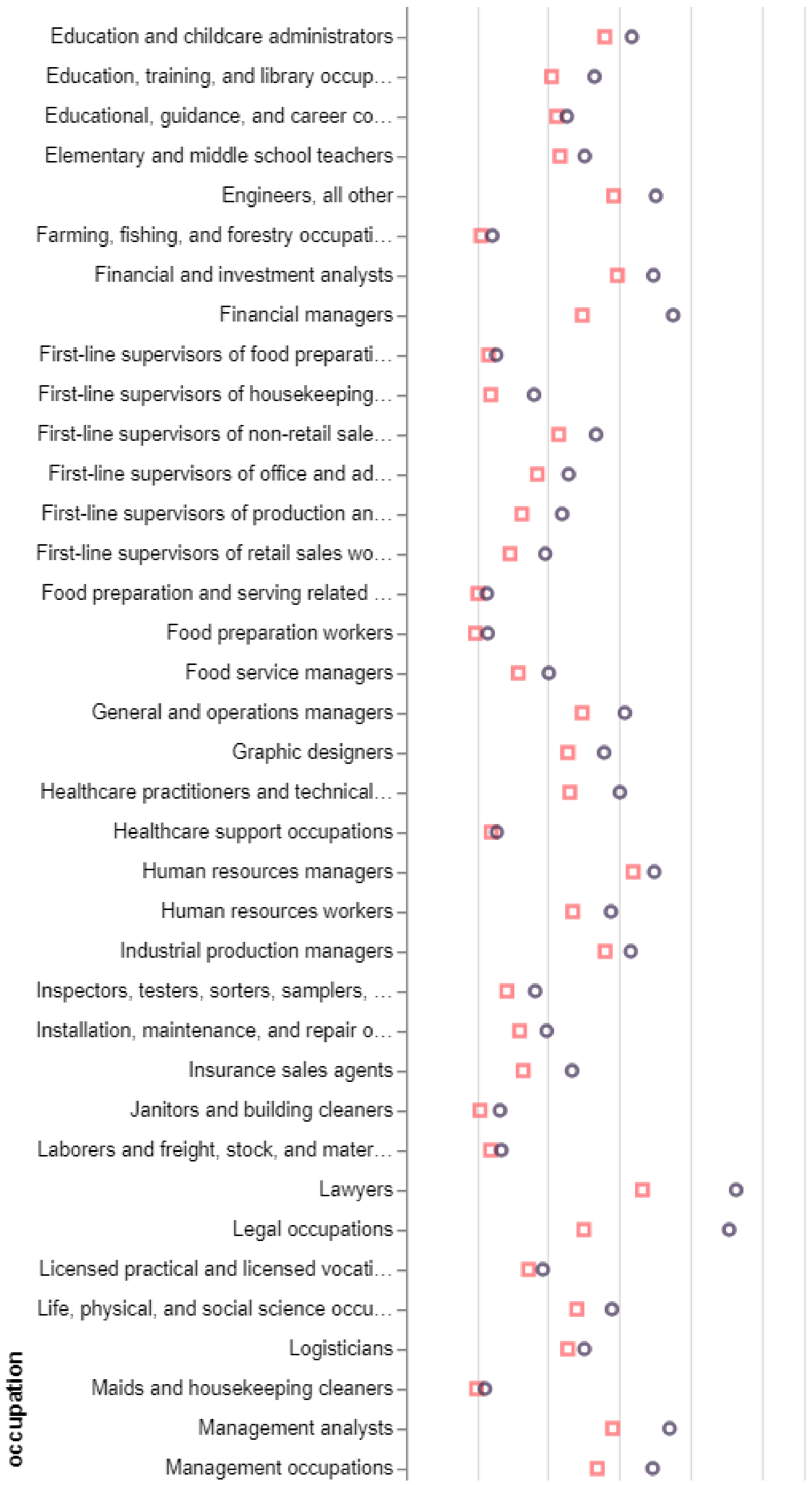
## Not used in presentation / just exploration

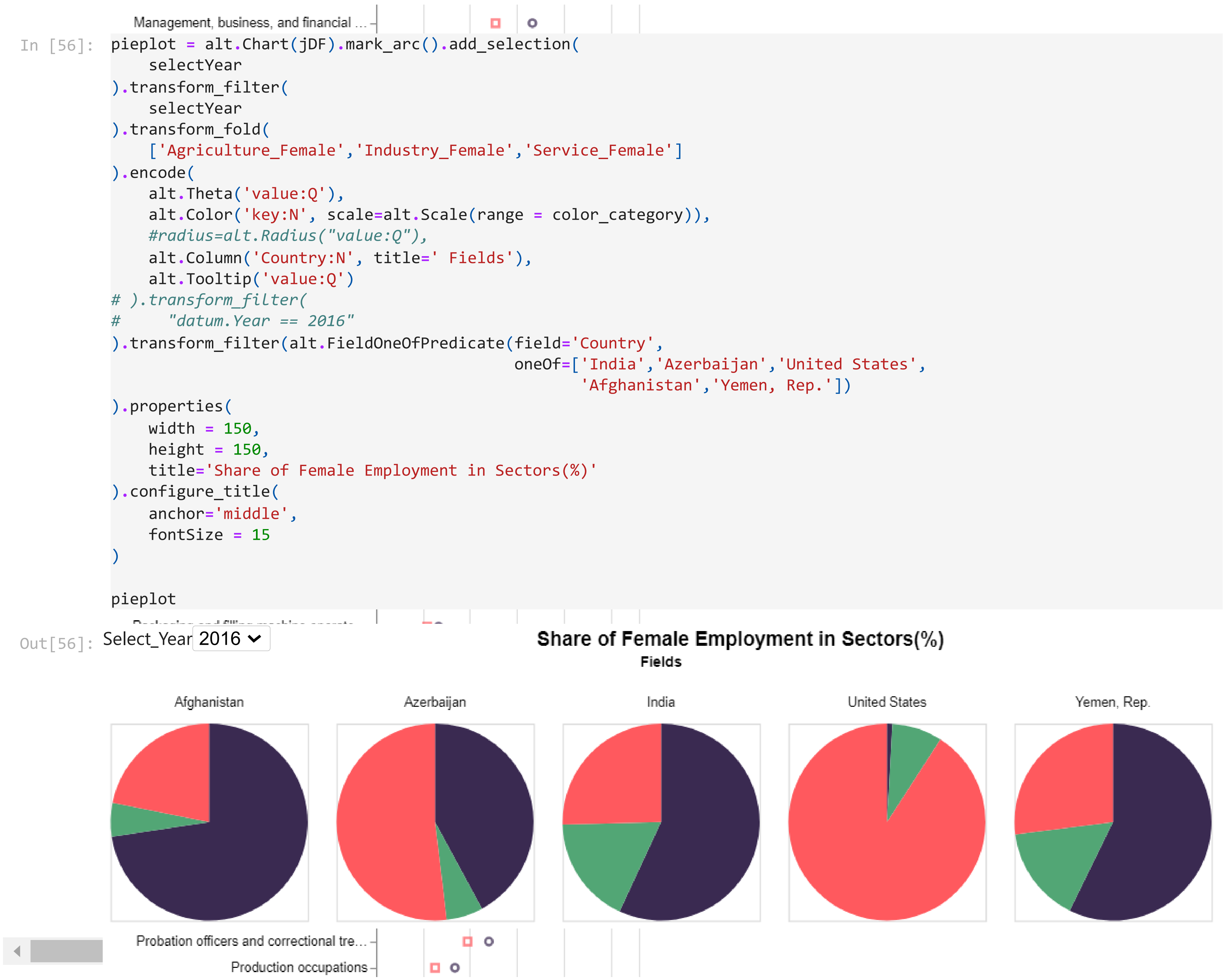
In [55]:

```
scatter_pay_gap
```

Out[55]:


Accountants and auditors
Advertising sales agents
Architecture and engineering occupati…
Arts, design, entertainment, sports, an…
Bakers
Bartenders
Bookkeeping, accounting, and auditin…
Building and grounds cleaning and m…
Bus drivers, school
Business and financial operations occ…
Business operations specialists, all ot…
Butchers and other meat, poultry, and …
Cashiers
Chefs and head cooks
Chief executives
Civil engineers
Claims adjusters, appraisers, examine…
Clergy
Clinical laboratory technologists and t…
Community and social service occupa…
Compliance officers
Computer and information systems m…
Computer and mathematical occupati…
Computer occupations, all other
Computer programmers
Computer support specialists
Computer systems analysts
Construction and extraction occupations
Construction managers
Cooks
Correctional officers and jailers
Couriers and messengers
Credit counselors and loan officers
Customer service representatives
Data entry keyers
Dispatchers, except police, fire, and a…
Driver/sales workers and truck drivers
Bus drivers, transit and intercity
Fast food and counter workers
Office and administrative support work…
Producers and directors
Wholesale and retail buyers, except fa…
0
1,000
2,000
3,000
median_week_earn
grou
○ n
□ w


occupation
Education and childcare administrators
Education, training, and library occup...
Educational, guidance, and career co...
Elementary and middle school teachers
Engineers, all other
Farming, fishing, and forestry occupati...
Financial and investment analysts
Financial managers
First-line supervisors of food preparati...
First-line supervisors of housekeeping...
First-line supervisors of non-retail sale...
First-line supervisors of office and ad...
First-line supervisors of production an...
First-line supervisors of retail sales wo...
Food preparation and serving related ...
Food preparation workers
Food service managers
General and operations managers
Graphic designers
Healthcare practitioners and technical...
Healthcare support occupations
Human resources managers
Human resources workers
Industrial production managers
Inspectors, testers, sorters, samplers, ...
Installation, maintenance, and repair o...
Insurance sales agents
Janitors and building cleaners
Laborers and freight, stock, and mater...
Lawyers
Legal occupations
Licensed practical and licensed vocati...
Life, physical, and social science occu...
Logisticians
Maids and housekeeping cleaners
Management analysts
Management occupations



Management, business, and financial ...
Packaging and filling machine operato...
Probation officers and correctional tre...
Production occupations

In [56]:

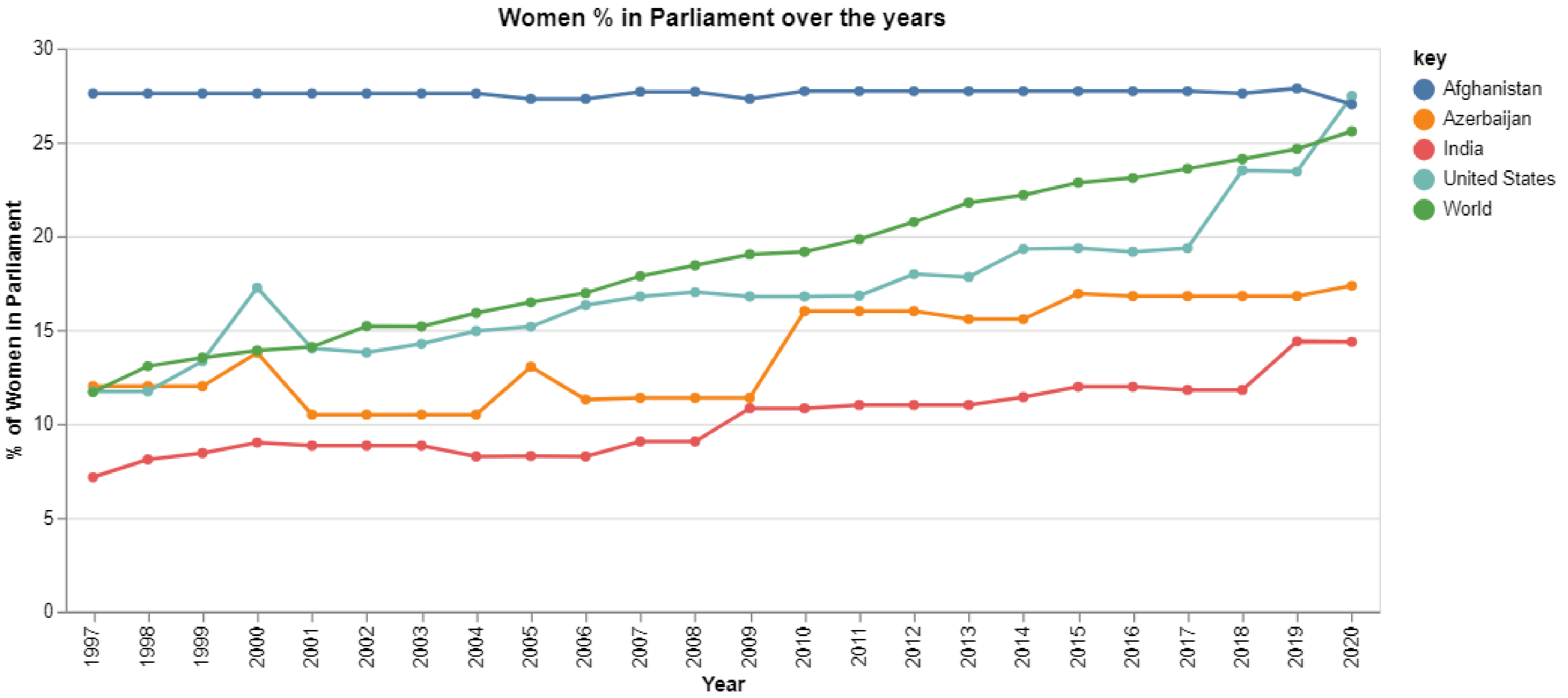
```
pieplot = alt.Chart(jDF).mark_arc().add_selection(
    selectYear
).transform_filter(
    selectYear
).transform_fold(
    ['Agriculture_Female','Industry_Female','Service_Female']
).encode(
    alt.Theta('value:Q'),
    alt.Color('key:N', scale=alt.Scale(range = color_category)),
    #radius=alt.Radius("value:Q"),
    alt.Column('Country:N', title=' Fields'),
    alt.Tooltip('value:Q')
# ).transform_filter(
#     "datum.Year == 2016"
).transform_filter(alt.FieldOneOfPredicate(field='Country',
                                           oneOf=['India','Azerbaijan','United States',
                                                  'Afghanistan','Yemen, Rep.'])

).properties(
    width = 150,
    height = 150,
    title='Share of Female Employment in Sectors(%)'
).configure_title(
    anchor='middle',
    fontSize = 15
)

pieplot
```

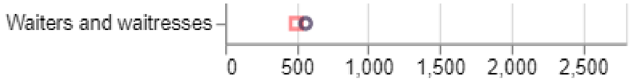
Out[56]:

In [57]:

```
parl_line = alt.Chart(parliament).mark_line(point=True).transform_fold(
     ['Azerbaijan','United States','India','Afghanistan','World']).encode(
    alt.X('Year:N', stack=None),
    alt.Y('value:Q',
          impute=alt.ImputeParams(method='mean'),
          axis=alt.Axis(tickMinStep = 5),
          scale=alt.Scale(domain=[0,30]),
          title = '% of Women in Parliament'),
    alt.Color('key:N'),
    alt.Tooltip('value:Q')
).properties(
    title ='Women % in Parliament over the years',
    width=700
)
parl_line
```

Out[57]:

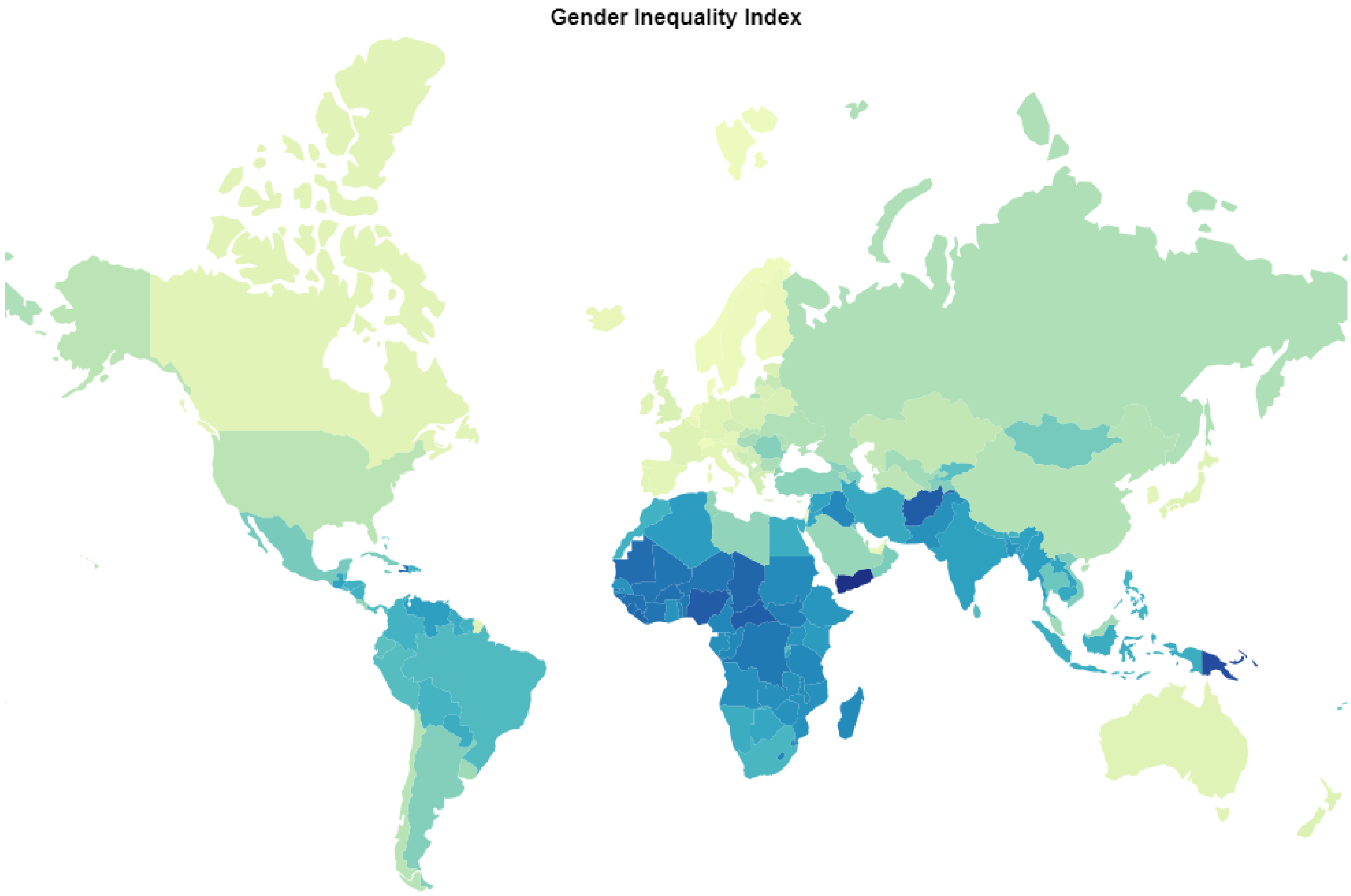




In [58]:

```
GDI_Map = ( alt.Chart(merge_DF).mark_geoshape(
).encode(
#     longitude='longitude:Q', # apply the field named 'longitude' to the longitude channel
#     latitude='latitude:Q'    # apply the field named 'latitude' to the latitude channel
    color  =  "GDI:Q"
    , tooltip = ["name" , "GDI"]
)).properties(
    width=900,
    height=500
    ,title  = "Gender Inequality Index"
).configure_view(
    stroke=None
)

GDI_Map
```

Out[58]:



In [ ]: