

```
In [221]: 1 import pandas as pd
2 import numpy as np
3 pd.set_option('display.max_columns', None)
4 df = pd.read_csv("table-1.csv")
5 df = df.drop(df.filter(like='Unnamed', axis=1), axis=1)
6 df = df.drop("Row Number", axis=1)
7 df = df.drop(["16-44 years", "45-59 years", "60 + years"], axis=1)
8 df["Industries"] = pd.read_csv("table-1.csv")["Unnamed: 1"]
9 df.insert(0, "Industries", df.pop("Industries"))
10 df = df.rename(columns={'With a disability': 'Median age With a disability', 'No disability': 'Median age No disability',
11                  'With a disability.1': 'Percent working full-time, year-round With a disability', 'No disability.1': 'Percent working full-time, year-round No disability',
12                  'With a disability.2': 'Median earnings of full-time, year-round workers With a disability', 'No disability.2': 'Median earnings of full-time, year-round workers
13 for i, row in df.iterrows():
14    try:
15        df.loc[i, 'Median earnings of full-time, year-round workers With a disability'] = int(str(df.loc[i, 'Median earnings of full-time, year-round workers With a disability']).replace(',',
16        df.loc[i, 'Median earnings of full-time, year-round workers No disability'] = int(str(df.loc[i, 'Median earnings of full-time, year-round workers No disability']).replace(',',
17        df.loc[i, 'Percent working full-time, year-round With a disability'] = float(df.loc[i, 'Percent working full-time, year-round With a disability'])
18        df.loc[i, 'Percent working full-time, year-round No disability'] = float(df.loc[i, 'Percent working full-time, year-round No disability'])
19    except ValueError:
20        pass
21 df.replace('-', np.nan)
22 df['Percent of workers in occupation with a disability'] = df['Percent of workers in occupation with a disability'].astype(float)
23 df['Median age With a disability'] = df['Median age With a disability'].astype(float)
24 df['Median age No disability'] = df['Median age No disability'].astype(float)
25 df['Median earnings of full-time, year-round workers With a disability'] = df['Median earnings of full-time, year-round workers With a disability'].astype(float)
26 df['Median earnings of full-time, year-round workers No disability'] = df['Median earnings of full-time, year-round workers No disability'].astype(float)
27 df['Percent working full-time, year-round With a disability'] = df['Percent working full-time, year-round With a disability'].astype(float)
28 df['Percent working full-time, year-round No disability'] = df['Percent working full-time, year-round No disability'].astype(float)
29 df = df.dropna()
30 df
```

Out[221]:

	Industries	Total employed workers	Workers with a disability	Percent of workers in occupation with a disability	Median age With a disability	Median age No disability	Percent working full-time, year-round With a disability	Percent working full-time, year-round No disability	Median earnings of full-time, year-round workers With a disability	Median earnings of full-time, year-round workers No disability
0	Total	155,041,900	9,085,980	5,9	51,5	41,3	58,3	71,3	41332,0	47279,0
1	Management, Business, Science, and Arts Occupa...	59,155,285	2,692,860	4,6	53,4	43,2	68,8	77,9	60233,0	66310,0
2	Management, Business, and Financial Occupations	24,008,060	1,114,300	4,6	54,6	45,2	76,4	86,1	62620,0	71924,0
3	Management Occupations	16,331,700	767,440	4,7	55,0	46,0	78,3	87,7	65623,0	75918,0
4	Chief executives	1,229,375	59,150	4,8	62,0	52,1	77,2	89,9	105849,0	136412,0
...	...	...	...	...	...	...	...	...	...	...
548	Industrial truck and tractor operators	616,000	37,390	6,1	49,7	39,7	78,0	84,3	32608,0	33018,0
549	Cleaners of vehicles and equipment	378,000	27,650	7,3	43,9	33,2	45,3	60,6	24759,0	26212,0
550	Laborers and freight, stock, and material move...	2,338,100	203,460	8,7	41,9	34,9	45,6	63,1	30405,0	31104,0
552	Packers and packagers, hand	520,790	48,085	9,2	44,0	37,8	40,4	64,3	24028,0	24703,0
554	Refuse and recyclable material collectors	95,325	11,780	12,4	52,7	40,6	30,9	70,6	40083,0	33665,0

268 rows × 10 columns

In [174]: 1 df.sort\_values(by=["Percent of workers in occupation with a disability"])

Out[174]:

	Industries	Total employed workers	Workers with a disability	Percent of workers in occupation with a disability	Median age With a disability	Median age No disability	Percent working full-time, year-round With a disability	Percent working full-time, year-round No disability	Median earnings of full-time, year-round workers With a disability	Median earnings of full-time, year-round workers No disability
113	Physical scientists, all other	243,825	6,125	2,5	40,6	37,8	74,7	82,8	64559,0	80759,0
49	Market research analysts and marketing special...	32,600	8,570	2,6	43,2	36,9	67,2	78,6	60916,0	71373,0
179	Physicians and surgeons	954,830	25,575	2,7	62,1	46,4	67,2	80,3	221607,0	202763,0
69	Software developers, applications and systems ...	1,363,955	38,825	2,9	44,9	38,4	86,7	88,9	92271,0	102312,0
8	Marketing and sales managers	989,695	30,530	3,1	47,7	41,7	80,5	87,5	70050,0	85541,0
...	...	...	...	...	...	...	...	...	...	...
235	Crossing guards	57,785	7,470	12,9	66,4	54,7	8,4	22,2	31031,0	30305,0
294	Models, demonstrators, and product promoters	60,750	8,325	13,7	60,0	44,6	7,6	26,1	28182,0	38286,0
529	Motor vehicle operators, all other	51,210	7,290	14,2	62,3	54,6	18,3	43,9	25822,0	30579,0
249	Dishwashers	347,200	50,385	14,5	36,2	28,5	24,7	33,1	17482,0	20937,0
500	Cutting workers	67,225	9,950	14,8	43,4	38,9	35,9	77,8	27143,0	31872,0

268 rows × 10 columns

In [145]: 1 df.dtypes

Out[145]:

Industries	object
Total employed workers	object
Workers with a disability	object
Percent of workers in occupation with a disability	object
Median age With a disability	object
Median age No disability	object
Percent working full-time, year-round With a disability	object
Percent working full-time, year-round No disability	object
Median earnings of full-time, year-round workers With a disability	object
Median earnings of full-time, year-round workers No disability	object
dtype: object	

```
In [51]: 1 import altair as alt
2 import warnings
3
4 warnings.filterwarnings('ignore')
5
6 df_2 = df[["Industries","Percent working full-time, year-round With a disability", "Median earnings of full-time, year-round workers With a disability"]]
7 df_2["disability"] = "disabled"
8 df_2.rename(columns={'Percent working full-time, year-round With a disability': 'Percent working full-time, year-round', 'Median earnings of full-time, year-round workers With a disability': 'Median earnings of full-time, year-round workers'}, inplace=True)
9 df_3 = df[["Industries","Percent working full-time, year-round No disability", "Median earnings of full-time, year-round workers No disability"]]
10 df_3["disability"] = "Not disabled"
11 df_3.rename(columns={'Percent working full-time, year-round No disability': 'Percent working full-time, year-round', 'Median earnings of full-time, year-round workers No disability': 'Median earnings of full-time, year-round workers'}, inplace=True)
12 df_4 = pd.concat([df_2, df_3], axis=0)
13 df_4.reset_index(drop=True)
14 for i, row in df_4.iterrows():
15     try:
16         df_4.loc[i, 'Median earnings of full-time, year-round workers'] = int(df_4.loc[i, 'Median earnings of full-time, year-round workers'].replace(',', ''))
17     except ValueError:
18         pass
19 df_4
```

Out[51]:

	Industries	Percent working full-time, year-round	Median earnings of full-time, year-round workers	disability
0	Total	58.3	41332	disabled
1	Management, Business, Science, and Arts Occupations	68.8	60233	disabled
2	Management, Business, and Financial Occupations	76.4	62620	disabled
3	Management Occupations	78.3	65623	disabled
4	Chief executives	77.2	105849	disabled
...	...	...	...	...
1111	Pumping station operators	-	-	- Not disabled
1112	Refuse and recyclable material collectors	70.6	33665	Not disabled
1113	Mine shuttle car operators	-	-	- Not disabled
1114	Tank car, truck, and ship loaders	-	-	- Not disabled
1115	Material moving workers, all other	-	-	- Not disabled

1116 rows × 4 columns

```
In [138]: 1 df_4.dtypes
```

Out[138]:

Industries	object
Percent working full-time, year-round	object
Median earnings of full-time, year-round workers	object
disability	object
dtype:	object



In [222]:

```

1 import altair as alt
2 from vega_datasets import data
3
4 y_domain = [1800,230000]
5
6 brush = alt.selection_interval(resolve='global')
7
8 base = alt.Chart(df).mark_point(filled=True).add_selection(
9     brush
10 ).encode(
11     opacity=alt.condition(brush, alt.value(1), alt.value(0.6)),
12     size=alt.SizeValue(30),
13     tooltip=['Industries']
14 ).properties(
15     width=400,
16     height=400
17 )
18
19 vline_disability=alt.Chart(df).transform_filter(
20     brush
21 ).transform_aggregate(
22     mean_perc_working_v_dis = "mean(Percent working full-time, year-round With a disability)",
23 ).mark_rule().encode(
24     x=alt.X("mean_perc_working_v_dis:Q", title=' '),
25     color= alt.value('#964000'),
26     size=alt.SizeValue(3),
27 )
28
29 vline_no_disability=alt.Chart(df).transform_filter(
30     brush
31 ).transform_aggregate(
32     mean_perc_working_v_no_dis = "mean(Percent working full-time, year-round No disability)",
33 ).mark_rule().encode(
34     x=alt.X("mean_perc_working_v_no_dis:Q", title=' '),
35     color= alt.value('blue'),
36     size=alt.SizeValue(3),
37 )
38
39 hline_disability=alt.Chart(df).transform_filter(
40     brush
41 ).transform_aggregate(
42     mean_perc_working_h_dis = "mean(Median earnings of full-time, year-round workers With a disability)",
43 ).mark_rule().encode(
44     y=alt.Y("mean_perc_working_h_dis:Q", title=' '),
45     color= alt.value('#964000'),
46     size=alt.SizeValue(3),
47 )
48
49 hline_no_disability=alt.Chart(df).transform_filter(
50     brush
51 ).transform_aggregate(
52     mean_perc_working_h_no_dis = "mean(Median earnings of full-time, year-round workers No disability)",
53 ).mark_rule().encode(
54     y=alt.Y("mean_perc_working_h_no_dis:Q", title=' '),
55     color= alt.value('blue'),
56     size=alt.SizeValue(3),
57 )
58
59 text_x_dis = alt.Chart(df).transform_filter(
60     brush
61 ).transform_aggregate(
62     mean_perc_working_v_dis = "mean(Percent working full-time, year-round With a disability)",
63     mean_perc_working_h_dis = "mean(Median earnings of full-time, year-round workers With a disability)"
64 ).transform_calculate(
65     x_position_dis=alt.datum.mean_perc_working_v_dis + 10,
66     y_position_dis=alt.datum.mean_perc_working_h_dis - 10000
67 ).mark_text().encode(
68     x='x_position_dis:Q',
69     y='y_position_dis:Q',
70     text = alt.Text('mean_perc_working_v_dis:Q', format='.4'),
71     size=alt.SizeValue(20),
72     color = alt.value('#1b0000')
73 )
74
75 text_y_dis = alt.Chart(df).transform_filter(
76     brush
77 ).transform_aggregate(
78     mean_perc_working_v_dis = "mean(Percent working full-time, year-round With a disability)",
79     mean_perc_working_h_dis = "mean(Median earnings of full-time, year-round workers With a disability)"
80 ).transform_calculate(
81     x_position_dis=alt.datum.mean_perc_working_v_dis + 10,
82     y_position_dis=alt.datum.mean_perc_working_h_dis + 10000
83 ).mark_text().encode(
84     x='x_position_dis:Q',
85     y='y_position_dis:Q',
86     text = alt.Text('mean_perc_working_h_dis:Q', format='.6'),
87     size=alt.SizeValue(20),
88     color = alt.value('#1b0000')
89 )
90
91 text_x_no_dis = alt.Chart(df).transform_filter(
92     brush
93 ).transform_aggregate(
94     mean_perc_working_v_no_dis = "mean(Percent working full-time, year-round No disability)",
95     mean_perc_working_h_no_dis = "mean(Median earnings of full-time, year-round workers No disability)"
96 ).transform_calculate(
97     x_position_no_dis=alt.datum.mean_perc_working_v_no_dis + 10,
98     y_position_no_dis=alt.datum.mean_perc_working_h_no_dis - 10000
99 ).mark_text().encode(
100    x=alt.X("x_position_no_dis:Q"),
101    y=alt.Y("y_position_no_dis:Q"),
102    text = alt.Text('mean_perc_working_v_no_dis:Q', format='.4'),
103    size=alt.SizeValue(20),
104    color = alt.value('#000000')
105 )
106
107 text_y_no_dis = alt.Chart(df).transform_filter(
108     brush
109 ).transform_aggregate(
110     mean_perc_working_v_no_dis = "mean(Percent working full-time, year-round No disability)",
111     mean_perc_working_h_no_dis = "mean(Median earnings of full-time, year-round workers No disability)"
112 ).transform_calculate(
113     x_position_no_dis=alt.datum.mean_perc_working_v_no_dis + 10,
114     y_position_no_dis=alt.datum.mean_perc_working_h_no_dis + 10000
115 ).mark_text().encode(
116    x=alt.X("x_position_no_dis:Q"),
117    y=alt.Y("y_position_no_dis:Q"),
118    text = alt.Text('mean_perc_working_h_no_dis:Q', format='.6'),
119    size=alt.SizeValue(20),
120    color = alt.value('#000000')
121 )
122
123 text_age_no_dis = alt.Chart(df).transform_filter(
124     brush
125 ).transform_aggregate(
126     mean_age_dis = "mean(Median age With a disability)",
127     mean_age_no_dis = "mean(Median age No disability)"
128 ).transform_calculate(
129     x_position_no_dis=alt.datum.mean_age_dis + 4,
130     y_position_no_dis=alt.datum.mean_age_no_dis + 4
131 ).mark_text(color='black').encode(
132    x=alt.X("x_position_no_dis:Q"),
133    y=alt.Y("y_position_no_dis:Q"),
134    text = alt.Text('mean_age_no_dis:Q', format='.4'),
135    size=alt.SizeValue(20),
136    color = alt.value('#152238')
137 )

```

```

137 )
138 text_age_dis = alt.Chart(df).transform_filter(
139   brush
140 ).transform_aggregate(
141   mean_age_dis = "mean(Median age With a disability)",
142   mean_age_no_dis = "mean(Median age No disability)"
143 ).transform_calculate(
144   x_position_no_dis=alt.datum.mean_age_dis + 4,
145   y_position_no_dis=alt.datum.mean_age_no_dis - 4
146 ).mark_text(color='black').encode(
147   x=alt.X("x_position_no_dis:Q"),
148   y=alt.Y("y_position_no_dis:Q"),
149   text = alt.Text('mean_age_dis:Q', format='.4'),
150   size=alt.SizeValue(20),
151   color = alt.value('#152238')
152 )
153 )
154
155 vline_age=alt.Chart(df).transform_filter(
156   brush
157 ).transform_aggregate(
158   mean_age_dis = "mean(Median age With a disability)",
159 ).mark_rule().encode(
160   x=alt.X("mean_age_dis:Q", title=' '),
161   color= alt.value("#013220"),
162   size=alt.SizeValue(3),
163 )
164
165 hline_age=alt.Chart(df).transform_filter(
166   brush
167 ).transform_aggregate(
168   mean_age_no_dis = "mean(Median age No disability)",
169 ).mark_rule().encode(
170   y=alt.Y("mean_age_no_dis:Q", title=' '),
171   color= alt.value("#013220"),
172   size=alt.SizeValue(3),
173 )
174
175 age = base.encode(x=alt.X('Median age With a disability',
176                           title='Median age of people with disability',
177                           scale=alt.Scale(domain=[15,70])),
178                           y=alt.Y("Median age No disability",
179                           title='Median age of people without disability',
180                           scale=alt.Scale(domain=[15,70])),
181                           color=alt.condition(brush, alt.value('#597d35'), alt.value('gray'))).properties(
182   width=400,
183   height=400
184 )+alt.Chart(pd.DataFrame({'x': [15, 70], 'y': [15, 70]})).mark_line(color='red').encode(
185   x='x',
186   y='y'
187 )+text_age_dis+text_age_no_dis+vline_age+hline_age
188
189 text_perc_no_dis = alt.Chart(df).transform_filter(
190   brush
191 ).transform_aggregate(
192   mean_perc_dis = "mean(Percent working full-time, year-round With a disability)",
193   mean_perc_no_dis = "mean(Percent working full-time, year-round No disability)"
194 ).transform_calculate(
195   x_position_no_dis=alt.datum.mean_perc_dis - 8,
196   y_position_no_dis=alt.datum.mean_perc_no_dis + 8
197 ).mark_text().encode(
198   x=alt.X("x_position_no_dis:Q"),
199   y=alt.Y("y_position_no_dis:Q"),
200   text = alt.Text('mean_perc_no_dis:Q', format='.4'),
201   size=alt.SizeValue(20),
202   color = alt.value('#152238')
203 )
204
205 text_perc_dis = alt.Chart(df).transform_filter(
206   brush
207 ).transform_aggregate(
208   mean_perc_dis = "mean(Percent working full-time, year-round With a disability)",
209   mean_perc_no_dis = "mean(Percent working full-time, year-round No disability)"
210 ).transform_calculate(
211   x_position_no_dis=alt.datum.mean_perc_dis + 8,
212   y_position_no_dis=alt.datum.mean_perc_no_dis - 8
213 ).mark_text().encode(
214   x=alt.X("x_position_no_dis:Q"),
215   y=alt.Y("y_position_no_dis:Q"),
216   text = alt.Text('mean_age_dis:Q', format='.4'),
217   size=alt.SizeValue(20),
218   color = alt.value('#152238')
219 )
220
221 vline_perc=alt.Chart(df).transform_filter(
222   brush
223 ).transform_aggregate(
224   mean_perc_dis = "mean(Percent working full-time, year-round With a disability)",
225 ).mark_rule().encode(
226   x=alt.X("mean_perc_dis:Q", title=' '),
227   color= alt.value("#5C4033"),
228   size=alt.SizeValue(3),
229 )
230
231 hline_perc=alt.Chart(df).transform_filter(
232   brush
233 ).transform_aggregate(
234   mean_perc_no_dis = "mean(Percent working full-time, year-round No disability)",
235 ).mark_rule().encode(
236   y=alt.Y("mean_perc_no_dis:Q", title=' '),
237   color= alt.value("#5C4033"),
238   size=alt.SizeValue(3),
239 )
240
241 perc = base.encode(x=alt.X('Percent working full-time, year-round With a disability',
242                             title='Percent working full-time, year-round of people with disability',
243                             ),
244                             y=alt.Y('Percent working full-time, year-round No disability',
245                             title='Percent working full-time, year-round of people without disability',
246                             ),
247                             color=alt.condition(brush, alt.value('#C4A484'), alt.value('gray'))).properties(
248   width=400,
249   height=400
250 )+alt.Chart(pd.DataFrame({'x': [0,100], 'y': [0,100]})).mark_line(color='red').encode(
251   x='x',
252   y='y'
253 )+text_perc_dis+text_perc_no_dis+vline_perc+hline_perc
254
255 text_median_no_dis = alt.Chart(df).transform_filter(
256   brush
257 ).transform_aggregate(
258   mean_median_dis = "mean(Median earnings of full-time, year-round workers With a disability)",
259   mean_median_no_dis = "mean(Median earnings of full-time, year-round workers No disability)"
260 ).transform_calculate(
261   x_position_no_dis=alt.datum.mean_median_dis - 10000,
262   y_position_no_dis=alt.datum.mean_median_no_dis + 10000
263 ).mark_text().encode(
264   x=alt.X("x_position_no_dis:Q"),
265   y=alt.Y("y_position_no_dis:Q"),
266   text = alt.Text('mean_median_no_dis:Q', format='.7'),
267   size=alt.SizeValue(20),
268   color = alt.value('#152238')
269 )
270
271 text_median_dis = alt.Chart(df).transform_filter(
272   brush

```

```

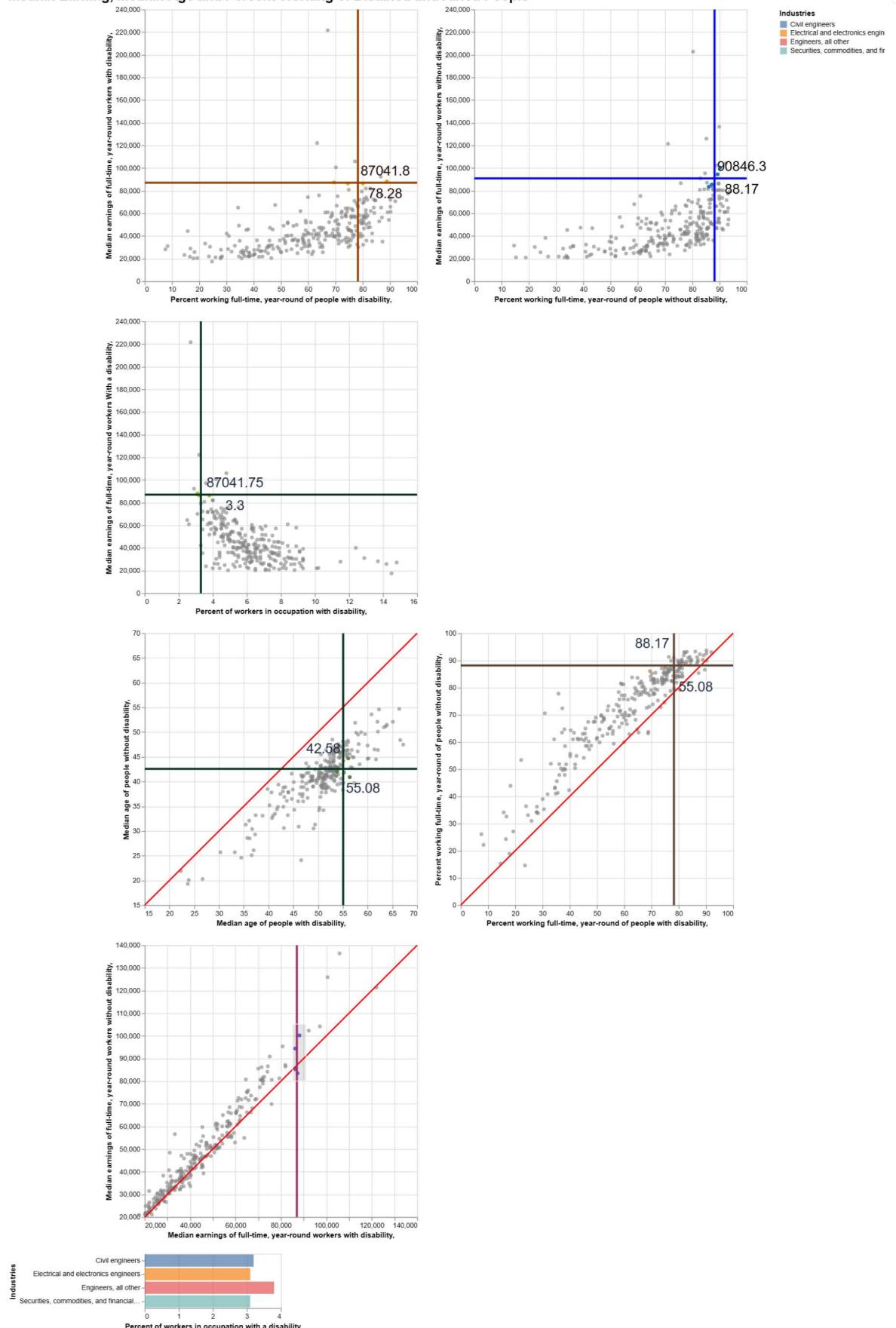
273 ).transform_aggregate(
274     mean_median_dis = "mean(Median earnings of full-time, year-round workers With a disability)",
275     mean_median_no_dis = "mean(Median earnings of full-time, year-round workers No disability)"
276 ).transform_calculate(
277     x_position_no_dis=alt.datum.mean_median_dis + 10000,
278     y_position_no_dis=alt.datum.mean_median_no_dis - 10000
279 ).mark_text().encode(
280     x=alt.X("x_position_no_dis:Q"),
281     y=alt.Y("y_position_no_dis:Q"),
282     text = alt.Text('mean_median_dis:Q', format='.7'),
283     size=alt.SizeValue(20),
284     color = alt.value('#152238')
285 )
286
287 vline_median=alt.Chart(df).transform_filter(
288     brush
289 ).transform_aggregate(
290     mean_median_dis = "mean(Median earnings of full-time, year-round workers With a disability)",
291 ).mark_rule().encode(
292     x=alt.X("mean_median_dis:Q", title=' '),
293     color= alt.value("#F2B68"),
294     size=alt.SizeValue(3),
295 )
296
297 hline_median=alt.Chart(df).transform_filter(
298     brush
299 ).transform_aggregate(
300     mean_median_no_dis = "mean(Median earnings of full-time, year-round workers No disability)",
301 ).mark_rule().encode(
302     y=alt.Y("mean_median_no_dis:Q", title=' '),
303     color= alt.value("#F2B68"),
304     size=alt.SizeValue(3),
305 )
306
307 median = base.transform_filter(
308     alt.datum['Median earnings of full-time, year-round workers With a disability'] < 200000
309 ).encode(x=alt.X("Median earnings of full-time, year-round workers With a disability",
310     title='Median earnings of full-time, year-round workers with disability',
311     scale=alt.Scale(domain=[20000,140000])),
312     y=alt.Y("Median earnings of full-time, year-round workers No disability",
313     title='Median earnings of full-time, year-round workers without disability',
314     scale=alt.Scale(domain=[20000,140000])),
315     color=alt.condition(brush, alt.value('#5c62d6'), alt.value('gray'))).properties(
316     width=400,
317     height=400
318 ).+alt.Chart(pd.DataFrame({'x': [20000,140000], 'y': [20000,140000]}).mark_line(color='red').encode(
319     x='x',
320     y='y'
321 ))\n+text_median_dis+text_median_no_dis+vline_median+hline_median
322
323 bar_1 = alt.Chart(df).transform_filter(
324     brush
325 ).transform_window(
326     sort=[alt.SortField('Percent of workers in occupation with a disability', order='descending')],
327     percent_rank='rank(*)'
328 ).transform_filter(
329     alt.datum.percent_rank < 20
330 ).mark_bar(opacity=0.7).encode(
331     x=alt.X("Percent of workers in occupation with a disability:Q",title="Percent of workers in occupation with a disability"),
332     y=alt.Y('Industries:N', title="Industries", sort='y'),
333     color = 'Industries:N'
334 )
335
336
337 text_indus_perc_dis = alt.Chart(df).transform_filter(
338     brush
339 ).transform_aggregate(
340     mean_1 = "mean(Percent of workers in occupation with a disability)",
341     mean_2 = "mean(Median earnings of full-time, year-round workers With a disability)"
342 ).transform_calculate(
343     x_position_no_dis=alt.datum.mean_1+2,
344     y_position_no_dis=alt.datum.mean_2-10000
345 ).mark_text(color='black').encode(
346     x=alt.X("x_position_no_dis:Q"),
347     y=alt.Y("y_position_no_dis:Q"),
348     text = alt.Text('mean_1:Q', format='.3'),
349     size=alt.SizeValue(20),
350     color = alt.value('#152238')
351 )
352
353 text_earning_dis = alt.Chart(df).transform_filter(
354     brush
355 ).transform_aggregate(
356     mean_1 = "mean(Percent of workers in occupation with a disability)",
357     mean_2 = "mean(Median earnings of full-time, year-round workers With a disability)"
358 ).transform_calculate(
359     x_position_no_dis=alt.datum.mean_1+2,
360     y_position_no_dis=alt.datum.mean_2 +10000
361 ).mark_text(color='black').encode(
362     x=alt.X("x_position_no_dis:Q"),
363     y=alt.Y("y_position_no_dis:Q"),
364     text = alt.Text('mean_2:Q', format='.7'),
365     size=alt.SizeValue(20),
366     color = alt.value('#152238')
367 )
368
369 vline_indus_perc=alt.Chart(df).transform_filter(
370     brush
371 ).transform_aggregate(
372     mean_indus_perc_dis = "mean(Percent of workers in occupation with a disability)",
373 ).mark_rule().encode(
374     x=alt.X("mean_indus_perc_dis:Q", title=' '),
375     color= alt.value("#013220"),
376     size=alt.SizeValue(3),
377 )
378
379 hline_earning=alt.Chart(df).transform_filter(
380     brush
381 ).transform_aggregate(
382     mean_earning_no_dis = "mean(Median earnings of full-time, year-round workers With a disability)",
383 ).mark_rule().encode(
384     y=alt.Y("mean_earning_no_dis:Q", title=' '),
385     color= alt.value("#013220"),
386     size=alt.SizeValue(3),
387 )
388
389 final_plot = alt.vconcat(alt.vconcat(alt.vconcat(base.encode(x=alt.X('Percent working full-time, year-round With a disability',
390     title='Percent working full-time, year-round of people with disability'),
391     y=alt.Y("Median earnings of full-time, year-round workers With a disability",
392     title='Median earnings of full-time, year-round workers with disability',
393     scale=alt.Scale(domain=y_domain)),
394     color=alt.condition(brush, alt.value('orange'), alt.value('gray'))))\
395     +vline_disability+hline_disability+text_y_dis+text_x_dis \
396 , base.encode(x=alt.X("Percent working full-time, year-round of people without disability",
397     title='Percent working full-time, year-round of people without disability'),
398     y=alt.Y("Median earnings of full-time, year-round workers No disability",
399     title='Median earnings of full-time, year-round workers without disability',
400     scale=alt.Scale(domain=y_domain)),
401     color=alt.condition(brush, alt.value('#f77b4'), alt.value('gray')))\\
402     +vline_no_disability+hline_no_disability+text_y_no_dis+text_x_no_dis,
403 base.encode(x=alt.X("Percent of workers in occupation with a disability",
404     title='Percent of workers in occupation with a disability',
405     color=alt.condition(brush, alt.value('#152238'), alt.value('gray')))\\
406     +y=alt.Y("Median earnings of full-time, year-round workers With a disability",
407     title='Median earnings of full-time, year-round workers With a disability',
408     scale=alt.Scale(domain=y_domain)),

```

```
409     color=alt.condition(brush, alt.value('#9acd32'), alt.value('gray')))\`  
410     +vline_indus_perc+hline_earning+text_earning_dis+text_indus_perc_dis, alt.hconcat(age,perc)),median),bar_1  
411     ).properties(title="Median Earning, Median Age and Percent Working of Disabled and Abled People"  
412     ).configure_title(fontSize=20)  
413 final_plot
```

Out[222]:

## Median Earning, Median Age and Percent Working of Disabled and Abled People



```
In [228]: 1 import streamlit as st
2
3 st.set_page_config(layout="wide")
4 st.altair_chart(final_plot, use_container_width=True)
5 st.altair_chart(pie, use_container_width=True)

Out[228]: DeltaGenerator(_root_container=0, _provided_cursor=None, _parent=None, _block_type=None, _form_data=None)
```

```
In [117]: 1 df_5 = pd.read_csv("table_d7_2010.csv").dropna()
2 df_5["15 years and older"] = df_5["15 years and older"].str.replace(",","").astype(int)
3 df_5["25 to 64 years and older"] = df_5["25 to 64 years and older"].str.replace(",","").astype(int)
4 df_5["65 years and older"] = df_5["65 years and older"].str.replace(",","").astype(int)
5 df_5["total"] = df_5["15 years and older"] + df_5["25 to 64 years and older"] + df_5["65 years and older"]
6 df_5["total percent"] = df_5["total"]
7 df_5["class"] = df_5["total"]
8 df_5 = df_5.reset_index(drop=True)
9 for i in range(len(df_5)):
10     if i % 6 != 0:
11         df_5.loc[i,"total percent"] = df_5.loc[i,"total percent"]/df_5.loc[(i//6)*6,"total percent"]
12         df_5.loc[i,"Disability / Health Status"] = df_5.loc[i,"Disability / Health Status"].replace(". . .","")
13     df_5.loc[i,"class"] = df_5.loc[(i//6)*6,"Disability / Health Status"]
```

```
In [118]: 1 df_6 = df_5.drop([i for i in range(len(df_5)) if i % 6 == 0])
2 df_6 = df_6.reset_index(drop=True)
3 df_6
```

```
Out[118]: Disability / Health Status 15 years and older Unnamed: 2 25 to 64 years and older Unnamed: 4 65 years and older Unnamed: 6 total total percent class
0 Excellent 69998 28,9 43212 26,8 4007 10,4 117117 0,265368 All disability statuses
1 Very Good 80857 33,5 58464 36,3 9769 25,3 149090 0,337814 All disability statuses
2 Good 59526 24,6 40288 25,0 13682 35,4 113496 0,257163 All disability statuses
3 Fair 22447 9,3 13853 8,6 7628 19,8 43928 0,099534 All disability statuses
4 Poor 8954 3,7 5239 3,3 3513 9,1 17706 0,040119 All disability statuses
5 Excellent 66473 34,9 41598 31,2 3359 17,3 111420 0,325097 No disability
6 Very Good 73276 38,5 54567 41,0 7188 37,1 135031 0,393989 No disability
7 Good 43324 22,8 31772 23,9 7284 37,6 82380 0,240366 No disability
8 Fair 6635 3,5 4850 3,6 1393 7,2 12878 0,037575 No disability
9 Poor 521 0,3 358 0,3 141 0,7 1020 0,002976 No disability
10 Excellent 2074 13,2 923 10,7 376 7,4 3373 0,114496 Disability, not severe
11 Very Good 3939 25,0 2169 25,2 1165 22,9 7273 0,246886 Disability, not severe
12 Good 6158 39,0 3449 40,1 2145 42,1 11752 0,398927 Disability, not severe
13 Fair 3031 19,2 1702 19,8 1224 24,0 5957 0,202213 Disability, not severe
14 Poor 569 3,6 351 4,1 186 3,7 1106 0,037544 Disability, not severe
15 Excellent 1351 3,8 702 3,6 273 1,9 2326 0,033638 Severe disability
16 Very Good 3643 10,2 1728 8,9 1416 10,0 6787 0,098150 Severe disability
17 Good 10045 28,2 5067 26,2 4253 30,1 19365 0,280047 Severe disability
18 Fair 12782 35,8 7301 37,8 5011 35,4 25094 0,362898 Severe disability
19 Poor 7863 22,0 4531 23,4 3186 22,5 15580 0,225311 Severe disability
```

```
In [225]: 1 dropdown = alt.binding_select(options=list(df_5["class"][:6].unique()), name=" ")
2 selection_1 = alt.selection_single(fields=["class"], bind=dropdown)
3
4 selection = alt.selection_interval(encodings=[{"y": " "})
5
6 bar_chart = alt.Chart(df_6).transform_filter(
7 selection_1
8 ).mark_bar().encode(
9 x=alt.X('total', scale=alt.Scale(domain=[0,160000])),
10 y=alt.Y('Disability / Health Status', sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
11 color='Disability / Health Status'
12 ).add_selection(selection_1).properties(
13 width=300,
14 height=400
15 )
16
17 pie_chart = alt.Chart(df_6).transform_filter(
18 selection_1
19 ).mark_arc().encode(
20 theta='total',
21 color='Disability / Health Status'
22 ).properties(
23 width=400,
24 height=400
25 ).add_selection(selection_1)
26 pie = (bar_chart|pie_chart).properties(title="Health Status of Disabled and Abled People"
27 ).configure_title(fontSize=20)
28 pie
```

Out[225]: **Health Status of Disabled and Abled People**

All disability statuses ▾

```
In [219]: 1 df_7 = pd.concat([df_6[["Disability / Health Status","15 years and older","class"]].rename(columns={"15 years and older":"total"}),
2                         df_6[["Disability / Health Status","25 to 64 years and older","class"]].rename(columns={"25 to 64 years and older":"total"}),
3                         df_6[["Disability / Health Status","65 years and older","class"]].rename(columns={"65 years and older":"total"})),axis=0)
4 df_7["age"] = ["15 years and older"]*20+["25 to 64 years and older"]*20+["65 years and older"]*20
5 df_7 = df_7.reset_index(drop=True)
6 df_7 = df_7[["Disability / Health Status","total","age"]]
7 df_7.head()
```

```
Out[219]:
```

	Disability / Health Status	total	age
0	Excellent	69898	15 years and older
1	Very Good	80857	15 years and older
2	Good	59526	15 years and older
3	Fair	22447	15 years and older
4	Poor	8954	15 years and older

```
In [214]: 1 df_7.dtypes
```

```
Out[214]:
```

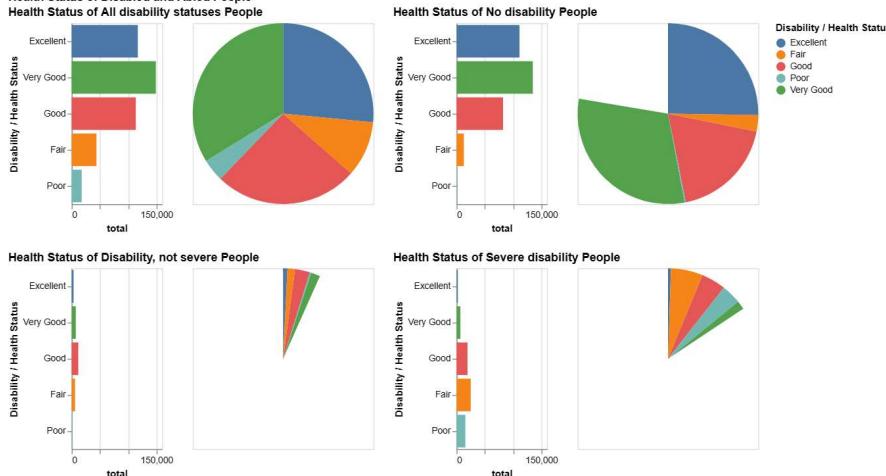
Disability / Health Status	object
total	int32
class	object
age	object
dtype:	object

```

In [229]:
1 bar_chart_1 = alt.Chart(df_6).transform_filter(
2 alt.datum['class'] == "All disability statuses"
3 ).mark_bar().encode(
4 x=alt.X('total',scale=alt.Scale(domain=[0,160000])),
5 y=alt.Y('Disability / Health Status',sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
6 color='Disability / Health Status'
7 ).properties(
8 width=100,
9 height=200
10 )
11 pie_chart_1 = alt.Chart(df_6).transform_filter(
12 alt.datum['class'] == "All disability statuses"
13 ).mark_arc().encode(
14 theta='total',
15 color='Disability / Health Status'
16 ).properties(
17 width=200,
18 height=200
19 )
20 )
21 chart_1 = (bar_chart_1|pie_chart_1).properties(title="Health Status of All disability statuses People"
22 )
23
24 bar_chart_2 = alt.Chart(df_6).transform_filter(
25 alt.datum['class'] == "No disability"
26 ).mark_bar().encode(
27 x=alt.X('total',scale=alt.Scale(domain=[0,160000])),
28 y=alt.Y('Disability / Health Status',sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
29 color='Disability / Health Status'
30 ).properties(
31 width=100,
32 height=200
33 )
34
35 pie_chart_2 = alt.Chart(df_6).transform_filter(
36 alt.datum['class'] == "No disability"
37 ).mark_arc().encode(
38 theta='total',
39 color='Disability / Health Status'
40 ).properties(
41 width=200,
42 height=200
43 )
44 chart_2 = (bar_chart_2|pie_chart_2).properties(title="Health Status of No disability People"
45 )
46
47 bar_chart_3 = alt.Chart(df_6).transform_filter(
48 alt.datum['class'] == "Disability, not severe"
49 ).mark_bar().encode(
50 x=alt.X('total',scale=alt.Scale(domain=[0,160000])),
51 y=alt.Y('Disability / Health Status',sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
52 color='Disability / Health Status'
53 ).properties(
54 width=100,
55 height=200
56 )
57
58 pie_chart_3 = alt.Chart(df_6).transform_filter(
59 alt.datum['class'] == "Disability, not severe"
60 ).mark_arc().encode(
61 theta='total',
62 color='Disability / Health Status',
63 # startAngle=alt.value(280)
64 ).properties(
65 width=200,
66 height=200
67 )
68 chart_3 = (bar_chart_3|pie_chart_3).properties(title="Health Status of Disability, not severe People"
69 )
70
71 bar_chart_4 = alt.Chart(df_6).transform_filter(
72 alt.datum['class'] == "Severe disability"
73 ).mark_bar().encode(
74 x=alt.X('total',scale=alt.Scale(domain=[0,160000])),
75 y=alt.Y('Disability / Health Status',sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
76 color='Disability / Health Status'
77 ).properties(
78 width=100,
79 height=200
80 )
81
82 pie_chart_4 = alt.Chart(df_6).transform_filter(
83 alt.datum['class'] == "Severe disability"
84 ).mark_arc().encode(
85 theta='total',
86 color='Disability / Health Status',
87 # startAngle=alt.value(305)
88 ).properties(
89 width=200,
90 height=200
91 )
92 chart_4 = (bar_chart_4|pie_chart_4).properties(title="Health Status of Severe disability People"
93 )
94
95 alt.vconcat(alt.hconcat(chart_1,chart_2),alt.hconcat(chart_3,chart_4),
96 ).properties(title="Health Status of Disabled and Abled People",
97 )

```

Out[229]:

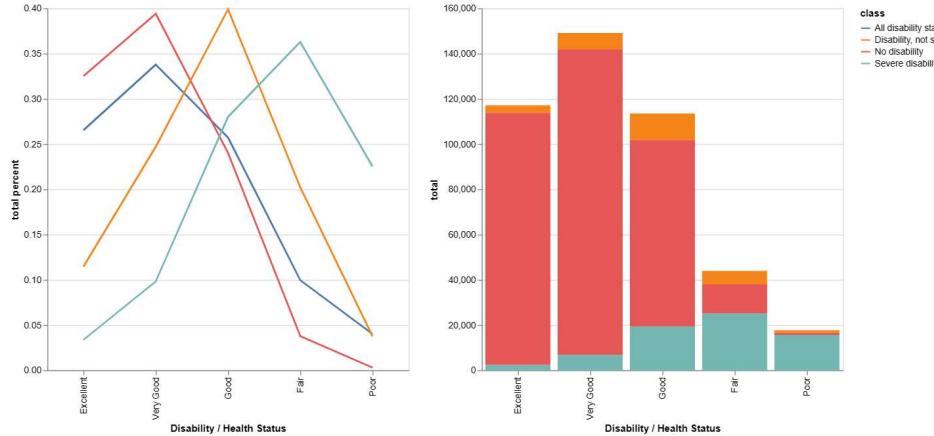


```

In [230]: 1 line_plot = alt.Chart(df_6).mark_line().encode(
2     y='total_percent:Q',
3     x=alt.X('Disability / Health Status:N',sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
4     color="class"
5 ).properties(
6     width=400,
7     height=400
8 )
9
10 bar_plot = alt.Chart(df_6).transform_filter(
11     alt.datum['class'] != "All disability statuses"
12 ).mark_bar().encode(
13     y='total:Q',
14     x=alt.X('Disability / Health Status:N',sort=['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']),
15     color="class"
16 ).properties(
17     width=400,
18     height=400
19 )
20
21 (line_plot|bar_plot).properties(title="Health Status of Disabled and Abled People")

```

Out[230]: Health Status of Disabled and Abled People



In [ ]:

1