Trends in 80s Prenatal Care Viz Project

Douglas Williams

10/20/2022

```
ax = sums_by_race_plot.plot.barh(stacked=True, figsize=(14,10), width=0.8,\
                                 color={"1st-2nd Month": "#1A3179", \
                                        "3rd Month": "#863A85",\
                                       "4th-6th Month": "#CF4E7B", \
                                        "7th-9th Month": "#FC7B66",\
                                       "No Care": "#FFB859", "No Data": "#2F4858"})
ax.set_xlabel('Percentage of 1985 Mothers', fontsize=24)
ax.set_ylabel('Race of Mother', fontsize=25.5)
ax.set_title('Month of Pregnancy 1985 Mothers Began Prenatal Care by Race', \
             fontsize=26, x=0.25, y=1.1)
ax.set_yticks(yticks, ylabels, fontsize=20)
ax.set_xticks(xticks, xlabels, fontsize=20)
ax.legend(bbox_to_anchor=(-0.11, 0), loc="lower right", prop={'size': 18})
ax.grid(axis='x', color='black', alpha=0.25)
plt.show()
```

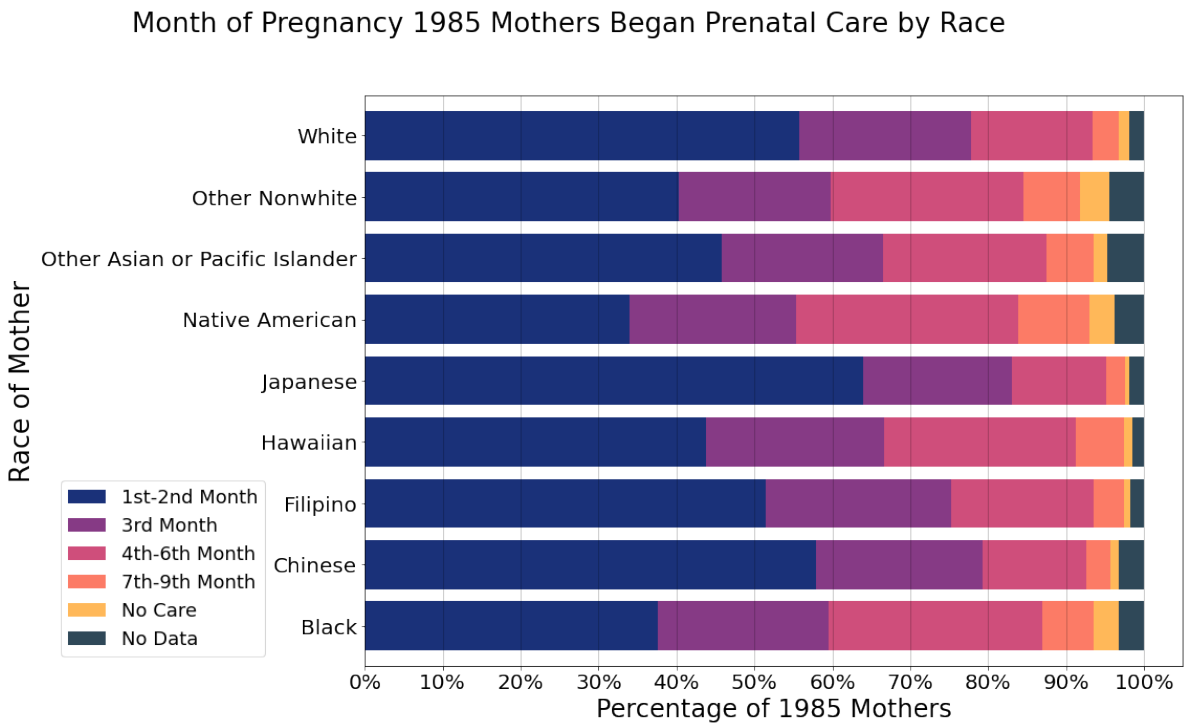

Figure 1: Data Source: 1985 Birth Data File. National Center for Health Statistics, Centers for Disease Control. https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm.

Caption

Above is an exploration of how early different mothers in 1985 began prenatal care visits, if at all. The data is broken up by race, labeled on the vertical, and quantifies the percentage of women of each race who began care at different color coded intervals. It is important to note we cannot draw any causal inference from this, as time of first prenatal care visit is influenced by a variety of factors such as health issues, healthcare access, cultural comfort with the medical profession, etc.

Regardless, this can help direct exploration. For example, Native American and Black mothers clearly engage with prenatal care later than other groups. Meanwhile, Japanese Americans quickly go in. It would be interesting to delve into more demographic data from 1985 to try and tease out reasons for these differences and see if they have any effect on the health of the mother and newborn.

Code for Production

```
birth_df = pd.read_fwf('Natl1985.txt', header=None, widths=[202])
```

```
birth_df.head()
```

	0
0	5010 1101002999917716301002631111112108 10...
1	5010 1101002999917716301002632111113017 15...
2	5010 1101002999917716301002631111112411 13...
3	5010 1101002999917716301002631111112411 13...
4	5010 1101002999917716301002631111112209 11...

```
birth_df.shape
```

```
(3765064, 1)
```

```
def pull_mother_race(row):
    race_choose = row[37]
    if race_choose == ' ':
        race_choose = np.nan
    return race_choose

def pull_father_race(row):
    race_choose = row[36]
    if race_choose == ' ':
        race_choose = np.nan
    return race_choose

def pull_prenatal_care(row):
    start = row[108]
    start_recoded = row[111]
    if start == ' ':
        start = np.nan
    if start_recoded == ' ':
        start_recoded = np.nan
    return start, start_recoded
```

```
def pull_mother_ed(row):
    ed_choose = row[97:99]
    ed_recode = row[101]
    if ed_choose == '  ':
        ed_choose = np.nan
    if ed_recode == ' ':
        ed_recode = np.nan
    return ed_choose, ed_recode

def pull_father_ed(row):
    ed_choose = row[102:104]
    ed_code14 = row[104:106]
    if ed_choose == '  ':
        ed_choose = np.nan
    if ed_code14 == '  ':
        ed_code14 = np.nan
    return ed_choose, ed_code14

def pull_num_births(row):
    live_choose = row[51:53]
    dead_choose = row[53:55]
    total_recoded = row[57:59]
    if live_choose == '  ':
        live_choose = np.nan
    if dead_choose == '  ':
        dead_choose = np.nan
    if total_recoded == '  ':
        total_recoded = np.nan
    return live_choose, dead_choose, total_recoded

def pull_mom_age(row):
    age_choose = row[42:44]
    age_recoded = row[48]
    if age_choose == '  ':
        age_choose = np.nan
    if age_recoded == ' ':
        age_recoded = np.nan
    return age_choose, age_recoded

def pull_all_data(row):
    mom_race = pull_mother_race(row)
```

```
    dad_race = pull_father_race(row)
    care_start, care_start_recoded = pull_prenatal_care(row)
    mom_age_choose, mom_age_recoded = pull_mom_age(row)
    mom_ed, mom_ed_recode = pull_mother_ed(row)
    dad_ed, dad_ed_code14 = pull_father_ed(row)
    live_births, dead_births, total_births = pull_num_births(row)
    value_list = [mom_race, dad_race, care_start, care_start_recoded,\
                  mom_age_choose, mom_age_recoded, mom_ed, mom_ed_recode,\
                  dad_ed, dad_ed_code14, live_births, dead_births, total_births]
    return value_list

def appending_loop(variable_arrays, value_list):
    for var_array, value in zip(variable_arrays, value_list):
        var_array.append(value)
    return variable_arrays

def extracting_loop(df, variable_arrays):
    loop_end = len(df)
    for index in range(loop_end):
        row = df.iloc[index].values[0]
        value_list = pull_all_data(row)
        variable_arrays = appending_loop(variable_arrays, value_list)
    return variable_arrays

def extract_from_df(df):
    mom_race = []
    dad_race = []
    care_start = []
    care_start_recoded = []
    mom_age_choose = []
    mom_age_recoded = []
    mom_ed = []
    mom_ed_recode = []
    dad_ed = []
    dad_ed_code14 = []
    live_births = []
    dead_births = []
    total_births = []
    variable_arrays = [mom_race, dad_race, care_start, care_start_recoded,\
                  mom_age_choose, mom_age_recoded, mom_ed, mom_ed_recode,\
                  dad_ed, dad_ed_code14, live_births, dead_births, total_births]
```

```
    variable_arrays = extracting_loop(df, variable_arrays)
    return variable_arrays

def format_raw_dfs(variable_arrays):
    source = variable_arrays
    previous_children_df = pd.DataFrame({'Prior Births (Alive)': source[10],\
                                         'Prior Births (Deceased)': source[11],\
                                        'Total Prior Births': source[12], \
                                         "Mother's Education": source[6],\
                                         "Mother's Education (Recoded)": source[7], \
                                         "Father's Education": source[8],\
                                         "Father's Education (Code14)": source[9], \
                                         "Mother's Age": source[4],\
                                        "Mother's Age (Recoded)": source[5]})
    parents_demographics_df = pd.DataFrame({"Mother's Education": source[6], \
                                            "Mother's Education (Recoded)": source[7],\
                                         "Father's Education": source[8], \
                                            "Father's Education (Code14)": source[9],\
                                        "Mother's Race": source[0], \
                                            "Father's Race": source[1]})
    prenatal_care_df = pd.DataFrame({"Mother's Race": source[0], \
                                     "Father's Race": source[1],\
                                   "Care Start": source[2], \
                                     "Care Start (Recoded)": source[3]})
    return previous_children_df, parents_demographics_df, prenatal_care_df

def extract_raw_plot_data(df):
    variable_arrays = extract_from_df(df)
    previous_children_df, parents_demographics_df, \
    prenatal_care_df = format_raw_dfs(variable_arrays)
    return previous_children_df, parents_demographics_df, prenatal_care_df
```

```
previous_children_df, parents_demographics_df, \
prenatal_care_df = extract_raw_plot_data(birth_df)
```

```
prenatal_care_df.to_csv('prenatal_care_df.csv', na_rep='NaN')
parents_demographics_df.to_csv('parents_demographics_df.csv', na_rep='NaN')
previous_children_df.to_csv('previous_children_df.csv', na_rep='NaN')
```

```
prenatal_care_df = pd.read_csv('prenatal_care_df.csv', index_col=0)
parents_demographics_df = pd.read_csv('parents_demographics_df.csv',index_col=0)
previous_children_df = pd.read_csv('previous_children_df.csv', index_col=0)
```

```
def decode_race(col):
    race_decoded = []
    decoding_dict = {0: 'Other Asian or Pacific Islander', 1: 'White', 2: 'Black',\
                     3: 'Native American', 4: 'Chinese',  5: 'Japanese', 6: 'Hawaiian',\
                     7: 'Other Nonwhite', 8: 'Filipino', 9: np.nan}
    for val in col:
        val = int(val)
        decoded_race = decoding_dict[val]
        race_decoded.append(decoded_race)
    race_decoded = np.array(race_decoded)
    return race_decoded

def raw_care_start_decode(col):
    raw_start_decoded = []
    decoding_dict = {1: '1st Month', 2: '2nd Month', 3: '3rd Month',\
                     4: '4th Month', 5: '5th Month',  6: '6th Month',\
                     7: '7th Month', 8: '8th Month', 9: '9th Month',\
                     0: 'No Care', '-': np.nan}
    for val in col:
        try:
            val = int(val)
        except:
            val = '-'
        decoded_start = decoding_dict[val]
        raw_start_decoded.append(decoded_start)
    raw_start_decoded = np.array(raw_start_decoded)
    return raw_start_decoded

def recoded_care_start_decode(col):
    recoded_start_decoded = []
    decoding_dict = {1: '1st-2nd Month', 2: '3rd Month', 3: '4th-6th Month',\
                     4: '7th-9th Month', 5: 'No Care', 6: np.nan}
    for val in col:
        try:
            val = int(val)
        except:
            val = '-'
```

```
        decoded_start = decoding_dict[val]
        recoded_start_decoded.append(decoded_start)
    recoded_start_decoded = np.array(recoded_start_decoded)
    return recoded_start_decoded
```

```
def zero_or_grade(val):
    if val == 0:
        return 'None'
    decoded = f'Grade {val}'
    return decoded

def decode_raw_ed(col):
    ed_decoded = []
    college_dict = {13: '1 Year College', 14: '2 Years College', 15: '3 Years College',\
                    16: '4 Years College', 17: '5+ Years Higher Ed', 99: np.nan}
    for val in col:
        if val < 13:
            decoded_ed = zero_or_grade(val)
            ed_decoded.append(decoded_ed)
            continue
        decoded_ed = college_dict[val]
        ed_decoded.append(decoded_ed)
    ed_decoded = np.array(ed_decoded)
    return ed_decoded

def recode_ed(col):
    recoded_ed = []
    for val in col:
        if val <= 8:
            recoded_ed.append('0-8 years')
            continue
        if val <= 11:
            recoded_ed.append('9-11 years')
            continue
        if val == 12:
            recoded_ed.append('12 years')
            continue
        if val <= 15:
            recoded_ed.append('13-15 years')
            continue
        if val == 16:
```

```
            recoded_ed.append('16 years')
            continue
        if val < 99:
            recoded_ed.append('17+ years')
            continue
        if val == 99:
            recoded_ed.append(np.nan)
    recoded_ed = np.array(recoded_ed)
    return recoded_ed
```

```
def decode_age_buckets(col):
    decoded_age = []
    decoding_dict = {1: "Under 15", 2: "15-19", 3: "20-24", 4: "25-29", 5: "30-34",\
                     6: "35-39", 7: "40-44", 8: "45+"}
    for val in col:
        decoded_age_bucket = decoding_dict[val]
        decoded_age.append(decoded_age_bucket)
    return decoded_age
```

```
care_vis_df = pd.DataFrame()
care_vis_df["Mother's Race"] = decode_race(prenatal_care_df["Mother's Race"])
care_vis_df["Father's Race"] = decode_race(prenatal_care_df["Father's Race"])
care_vis_df["Care Start"] = raw_care_start_decode(prenatal_care_df["Care Start"])
care_reoded_short = ecoded_care_start_decode(prenatal_care_df["Care Start (Recoded)"])
care_vis_df["Care Start Buckets"] = care_reoded_short
care_vis_df.head()
```

	Mother's Race	Father's Race	Care Start	Care Start Buckets
0	White	White	3rd Month	3rd Month
1	White	White	2nd Month	1st-2nd Month
2	White	White	2nd Month	1st-2nd Month
3	White	White	2nd Month	1st-2nd Month
4	White	White	2nd Month	1st-2nd Month

```
# row index is race
# columns are percentage of race that began care at each time bucket
```

```
care_vis_df["Mother's Race"].unique()
```

```
array(['White', 'Other Asian or Pacific Islander', 'Black', 'Japanese',
       'Filipino', 'Native American', 'Chinese', 'nan', 'Other Nonwhite',
       'Hawaiian'], dtype=object)
```

```
care_category_counts = care_vis_df.groupby(["Mother's Race",'Care Start Buckets'])
care_category_counts = care_category_counts.size().reset_index(name='counts')
```

```
sums_by_race = care_category_counts.groupby("Mother's Race").sum()
```

```
care_vis_df["Care Start Buckets"].unique()
```

```
array(['3rd Month', '1st-2nd Month', '4th-6th Month', '7th-9th Month',
       'nan', 'No Care'], dtype=object)
```

```
one_two_month = []
third_month = []
four_six_month = []
seven_nine_month = []
no_care = []
nan = []

cols = [one_two_month, third_month, four_six_month, seven_nine_month, no_care, nan]

for index in range(0, 10):
    total = sums_by_race['counts'][index]
    start = (index) * 6
    stop = start + 6
    for row, col in zip(range(start, stop), cols):
        val = care_category_counts["counts"][row]
        pct = round((val / total) * 100, 2)
        col.append(pct)
```

```
sums_by_race["1st-2nd Month"] = one_two_month
sums_by_race["3rd Month"] = third_month
sums_by_race["4th-6th Month"] = four_six_month
sums_by_race["7th-9th Month"] = seven_nine_month
sums_by_race["No Care"] = no_care
sums_by_race["No Data"] = nan
#sums_by_race
```

```
sums_by_race_plot = sums_by_race.copy()
sums_by_race_plot = sums_by_race_plot.drop('counts', axis=1)
sums_by_race_plot = sums_by_race_plot.drop('nan')
```

```
sums_by_race_plot.head()
```

Mother's Race	1st-2nd Month	3rd Month	4th-6th Month	7th-9th Month	No Care	No Data
Black	37.56	21.93	27.47	6.51	3.31	3.22
Chinese	57.85	21.43	13.26	3.17	1.04	3.25
Filipino	51.44	23.75	18.33	3.88	0.83	1.77
Hawaiian	43.81	22.86	24.54	6.24	1.01	1.54
Japanese	63.91	19.17	12.00	2.43	0.60	1.88

```
ylabels = list(sums_by_race_plot.index)
yticks = range(len(ylabels))
xticks = [0,10,20,30,40,50,60,70,80,90,100]
xlabels = [f'{tick}%' for tick in xticks]
```