Australian_Trends

October 28, 2023

CITS2402 Practical Assignment, Semester 2 2023

1 Exploring Trends in Australian Census Data

1.1 Declaration

This declaration should be completed and remain attached to the top of your submission.

I/we am/are aware of the University's policy on academic conduct and I declare that this assignment is entirely the work of the author(s) listed below and that suitable acknowledgement has been made for any sources of information used in preparing it. I have retained a copy for my own records.

- Name 1: Goh Kuang Thai
- Student ID 1: 23467903
- Name 2: Shaden Albalawi
- Student ID 2: 23269191
- Name 3: Aaminah Irfan
- Student ID 3: 23642166
- Date: 11/09/2023

Remove `_template` from your filename for submission.

2 Examine trends using ABS Census data.

Hypothesis: There is a significant difference in the distribution of income categories among different gender groups in the periods 1996 - 2021

2.1 Context:

The issue of gender pay disparity is a persistent concern globally, encompassing a long history of struggle for equal wages and opportunities. Despite significant advancements in women's participation in the Western Australian workforce, recent statistics reveal an unsettling reality. According to a report by ABC News, female participation in the WA workforce stands at 64.5 percent, showcasing women's growing presence in the labor market [2]. However, this promising trend is shadowed by the fact that the gender pay gap in Western Australia remains notably higher than the national average.

In November 2022, Western Australia reported the widest gender pay gap among the Australian states, with a staggering 21.4 percent difference between men's and women's earnings [1]. This stark contrast, as highlighted by the Workplace Gender Equality Agency (WGEA), underscores the urgency of addressing gender-based income disparities in the region.

Our study delves into this issue, aiming to assess whether progress has been made in narrowing the gender pay gap in Western Australia from 1996 to 2021. It seeks to determine if higher-income brackets continue to be male-dominated and if lower-income categories are still predominantly occupied by females. In an era of heightened awareness and evolving gender dynamics, this research contributes to the broader discourse on gender equality, shedding light on the current state of the gender pay gap and the extent of change achieved over the years.

2.1.1 Data Location

- Data Topic: Total Personal Weekly Income by Age and Sex
- Region of Interest: Western Australia (LGA)

Data Sourcing for 1996 and 2001: For the years 1996 and 2001, data was accessed from the Australian Bureau of Statistics (ABS) Census page. Since Census Data Packs were not available for Year 1996 and 2001, 'Find Census Data' feature was used. Using options for Year and Location, desired Year (1996,2001) and location (Western Australia) was selected. 'Basic Community Profile' data was downloaded from the output of the search results. File size was approximately 1 MB. Downloaded files were opened, and specific data of interest:' B13 Weekly Total Personal Income by Sex' was located, extracted and converted to CSV format as a separate file for further analysis. Note that specific region (LGA) data was not provided, so the whole Western Australia data was used for analysis.

Data Sourcing for 2011, 2016, and 2021: Data Packs for individual years were accessed using ABS Census Data Packs page. Using the options on the page, Desired Census year (2011, 2016, or 2021) was selected, "Basic Community Profile" was chosen as the Data Pack type, "LGA" (Local Government Year) was selected as Geography type and the state "Western Australia" was chosen. Output File from Search Results were downloaded which were approximately 5 MB (for each year). Using the Metadata provided in the downloaded data packs, specific data for "Weekly Personal Income by Sex" was located, extracted, converted to CSV format and saved as a separate file for each year to facilitate analysis. "All Persons" data was not extracted since it is not in the scope of the analysis.

2.1.2 Data Differences:

Since, data packs were not available for 1996 and 2001 data, they were extracted from General Community Profile data of whole of Western Australia. On the other hand, data for 2011-2021 of Western Australia was split into different regions one of them being LGA (Local Government Area). LGA is a region that falls within the boundaries of a state. Hence, LGA was chosen as it covers most of Western Australia compared to other geographical areas provided in ABS data which focus on smaller regions of Western Australia such as the main city, remote areas, suburbs etc. Data Format for data 1996 - 2001 and 2011-2021 were also in different format. The data format for 1996 and 2001 is tabular and organized by age groups and sex. It provides counts of individuals falling into various income categories for each age and sex group. The format includes rows for each income category and columns for different age groups, such as "15-19 years," "20-24 years," and so on, as well as separate tables for males and females. Each cell in the table contains the count of individuals falling into a specific income category within a particular age and sex group. On the other hand, data for 2011-2021 is organized at the level of Local Government Areas (LGAs) within Western Australia. Each row represents a specific LGA. The data includes various columns that break down count of people by income, age groups and gender for each LGA. For example, you have columns like "M_Neg_Nil_income_15_19_yrs," and so on, which provide income information on count of "Males" in different age (15-19) and income groups (Neg/Nil Income) groups. Given that the data for 2011-2021 was initially segmented into various Local Government Area (LGA) regions, and the objective of this project is to provide an overview of the entire Western Australia region rather than specific LGAs, a Python script was employed to aggregate or sum all the rows, effectively consolidating the data for the different LGAs into a comprehensive representation of Western Australia as a whole. In the process of analyzing the data from various years, variations in the categorization of income and age groups between the earlier dataset (1996-2001) and the more recent datasets (2011, 2016, and 2021) were encountered These differences can stem from changes in data collection and reporting methods over time, which are important to acknowledge for the sake of accurate analysis and comparison.

2.1.3 Assumptions:

- The data for 1996 and 2001, extracted from Basic Community Profile data, is representative of the entire Western Australia region since specific LGA-level data was not available.
- For 2011-2021 data, it is assumed that the LGA-level data, which was aggregated to represent whole Western Australia, accurately reflects the income distribution for the entire state.
- The data collected by ABS is assumed to be accurate and free from significant errors or biases. The income distribution data collected in each respective year is assumed to be temporally relevant and accurately represents the income distribution for that year or at-least up to that year.

2.1.4 Packages:

In this project, the following packages were used:

- csv
- numpy
- re
- matplotlib/pyplot

2.2 Methodology

Essentially how we approached the data and we handled the visualisations is we broken it down to two parts.

2.2.1 Part one

- This is where we handle the cleaning of the overall data, into a format we could use for visualisation of the data in each year.
- We approached visualisation of the data by plotting line charts for the cleaned data to illustrate the overall trends we were traying to explore

2.2.2 Part two

- In this part, we are going for the approach of showing the aggregate discrepancies that we were trying to explore over a certain income range, to illustrate how bad is the difference between the number of employed men and women over a certain income range.
- So therefore the code for part two is more so to combined the data, so that it allows us to use it in a standardised form, and therefor allow for visualisation to be made for all the categories on the same number context.

2.3 Data cleaning from raw to preferred format Part One:

- Because the formatting of (1996 - 2001) data sets and (2011, 2016 and 2021) data sets were different, two different cleaning functions are used for the respective ranges of the data sets.

For 1996 and 2001:

1. The function ***cleaned(data)*** reads into the data, extracts, and cleans age categories, and converts it into integers. It also extracts the Male and Females data into separate lists.
2. The function ***re_format(male_cat, female_cat)*** conducts data reformatting, so the list appears to be in the same format as the 2021 data set. That is done by extracting and replacing some characters in the data.

For 2011 - 2021:

1. The function ***clean(data)*** reads into the data, removes any empty lists and values, converts all numbers to integers, separates categories and numbers, and stores them into a list.
2. The function ***get_sum(numbers)*** gets the sum of the columns into one list.
3. Join the data parts into one list for the data sets that are divided into multiple files.
4. The function ***get_format(data)*** takes in the data, re-formats it, and separates female and male data. Additionally, it removes the Neg / Nil income category.

[3]:
```
# IMPORTING ALL THE FILES AND ALSO THE LIBRARIES WE USED
import csv
import re
import numpy as np
from matplotlib import pyplot as plt
A_1996 = '1996_A.csv'
A_2001 = '2001_A.csv'
B_2001 = '2001_B.csv'
A_2011 = '2011_A.csv'
B_2011= '2011_B.csv'
A_2016 = '2016_A.csv'
B_2016 = '2016_B.csv'
C_2016 = '2016_C.csv'
A_2021 = '2021_A.csv'
B_2021 = '2021_B.csv'
C_2021 = '2021_C.csv'
```

[4]:
```
## DIFFERENT CLEANING FUNCTION FOR 1996 AND 2001
def cleaned(data):
```

```
    """
    Read into the data, extract and clean age , extract and clean Male and␣
↪female data
    """
    csv.register_dialect('myDialect',
    delimiter=',',
    skipinitialspace=True,
    quoting=csv.QUOTE_ALL)

    original = []
    # READ INTO THE DATA
    with open(data, newline='') as file:
        reader = csv.reader(file, dialect = 'excel')
        for row in reader:
            row = [item.replace(' ', '').replace('\t', '') for item in row if␣
↪item != '']
            if len(row) == 0:
                continue
            else:
                original.append(row)
    # EXTRACT AGE CATEGORIES
    rawages = original[4] + original[5]
    ages = [] # has all the age categories
    for x in rawages:
        if x not in ages:
            ages.append(x)
    ages.pop(-3)
    ages[7:9] = [''.join(ages[7 : 9])]

    original = original[6:]
    # CONVERT ALL NUMBER STRINGS TO INT
    for i in range(len(original)):
        for j in range(len(original[i])):
            if isinstance(original[i][j], str) and original[i][j].replace(",",␣
↪"").isdigit():
                original[i][j] = int(original[i][j].replace(",", ""))
    for row in original:
        if len(row)==1 and (row[0]=='MALES' or row[0]=='MALE'):
            male_idx = original.index(row)
        if len(row)==1 and (row[0]=='FEMALES' or row[0]=='FEMALE'):
            female_idx = original.index(row)
    male_data = original[male_idx+2: female_idx]
    female_data = original[female_idx+2:]

    return  male_data, female_data
```

```python
def re_format(male_cat, female_cat):
    """
    formatting specifically for the 1996 and 2001 data to look look like 2021
    """

    # data reformatting so the list appears to be the same as the data_2021

    for lst in male_cat:
        lst[0]='M_'+ lst[0].replace('$','').replace('-','_').
 ↪replace('ormore','_more').replace(',','').replace('Notstated','PI_NS').
 ↪replace('Total','Tot')

    for lst in female_cat:
        lst[0]='F_'+ lst[0].replace('$','').replace('-','_').
 ↪replace('ormore','_more').replace(',','').replace('Notstated','PI_NS').
 ↪replace('Total','Tot')

    # fixing the list so that it exlcudes the overseas cat and also makes it so␣
 ↪that the total count is less of overseas count

    oversea_values_M=male_cat[-2][1:]
    Total_value_M=male_cat[-1][1:]
    Total_less_oversea_M=[]

    oversea_values_F=female_cat[-2][1:]
    Total_value_F=female_cat[-1][1:]
    Total_less_oversea_F=[]

    # REMOVING THE VALUES OF THE OVERSEAS CATEGORY
    for oversea,total in zip(oversea_values_M,Total_value_M):
        value=total-oversea
        Total_less_oversea_M.append(value)

    Total_less_oversea_M.insert(0,'M_Tot')
    male_cat.insert(-2,Total_less_oversea_M)

    male_cat_no_overseas=male_cat[:-2]

    for oversea,total in zip(oversea_values_F,Total_value_F):
        value=total-oversea
        Total_less_oversea_F.append(value)

    Total_less_oversea_F.insert(0,'F_Tot')
    female_cat.insert(-2,Total_less_oversea_F)
```

```
    female_cat_no_overseas=female_cat[:-2]

    return male_cat_no_overseas,female_cat_no_overseas


male_cat2001, female_cat2001 =cleaned(A_2001)
male_cat1996, female_cat1996 =cleaned(A_1996)

# ASSIGNING OF VARIABLES
data_2001 = re_format(male_cat2001, female_cat2001)
data_1996 = re_format(male_cat1996, female_cat1996)

# FOR CLEANING PURPOSES, AS THE FIRST INDEX OF EACH LIST HAD A CATEGORY THAT␣
 ↪WAS NOT ACCOUNTED FOR IN THE CODE
for x in data_1996:
    x.pop(0)
```

[5]:

```
# CLEANING FOR DATA SETS, 2011 2016 AND 2021
def clean(data):
    """
    Read in the data, remove any empty lists / values,
    convert all numbers to integers, separated categories,
    and numbers and store them into lists.
    """
    # DEFINE YOUR OWN DIALECT
    csv.register_dialect('myDialect',
                     delimiter=',',
                     skipinitialspace=True,
                     quoting=csv.QUOTE_ALL)
    # READ INTO THE DATA
    original = []
    with open(data, newline='') as file:
        reader = csv.reader(file, dialect = 'excel')
        for row in reader:
            row  = [item.replace(' ', '').replace('\t', '') for item in row if␣
 ↪item != '']
            if len(row) == 0:
                continue
            else:
                row = [int(value) if value.isdigit() else value for value in␣
 ↪row]
                original.append(row)
    categories = original[0][1:]
    numbers = original[1:]
    return categories, numbers
```

```python
# GET THE SUM OF THE COLUMNS INTO ONE LIST
def get_sum(numbers):
    new = []
    for x in numbers:
        z = x.pop(0)
        new.append(x)

    summed = [sum(x) for x in zip(*new)]
    return summed

# JOIN DATA PARTS INTO ONE LIST
data_2021 = []

categoriesA, numbersA = clean(A_2021)
categoriesB, numbersB = clean(B_2021)
categoriesC, numbersC = clean(C_2021)
data_2021.append(categoriesA)
data_2021[0].extend(categoriesB)
data_2021[0].extend(categoriesC)

data_2021.append(get_sum(numbersA))
data_2021[1].extend(get_sum(numbersB))
data_2021[1].extend(get_sum(numbersC))

data_2016 = []

categoriesA, numbersA = clean(A_2016)
categoriesB, numbersB = clean(B_2016)
categoriesC, numbersC = clean(C_2016)
data_2016.append(categoriesA)
data_2016[0].extend(categoriesB)
data_2016[0].extend(categoriesC)

data_2016.append(get_sum(numbersA))
data_2016[1].extend(get_sum(numbersB))
data_2016[1].extend(get_sum(numbersC))

data_2011 = []

categoriesA, numbersA = clean(A_2011)
categoriesB, numbersB = clean(B_2011)
data_2011.append(categoriesA)
data_2011[0].extend(categoriesB)

data_2011.append(get_sum(numbersA))
data_2011[1].extend(get_sum(numbersB))
```

```python
def get_format(data):
    """
    takes in the data and re-formats it and separate male and female data
    """
    categories = data[0]
    numbers = data[1]
    male = []
    female = []
    for i in range(0, len(categories), 10): # iterate over every 10th element␣
 ↪(10 age categories)
        category = categories[i]
        new_cat = category[:-10] # remove age categories from new category name
        if new_cat[0] == "M": # separate male data
            new_numbers = numbers[i:i + 10]
            row = [new_cat] + new_numbers
            male.append(row)
        elif new_cat[0] == "F": # separate female data
            new_numbers = numbers[i:i + 10]
            row = [new_cat] + new_numbers
            female.append(row)
    # remove Neg/ Nil income category:
    male_data = male[1:]
    female_data = female[1:]
    for i in (male_data, female_data):
        for x in i:
            if len(x) > 8:
                summed_value = x[-2] + x[-3]
                x[-2] = summed_value
                x.pop(-3)


    return male_data, female_data

# Re-format dta (2011-2021) and assign to a new variable
data_2021 = get_format(data_2021)
data_2016 = get_format(data_2016)
data_2011 = get_format(data_2011)
```

3 Visualisation of the data in Part One

-

3.0.1 Line Plots

- With the data for all the years cleaned but not yet standardized to the ranges we preferred, we first made line plots of each year's data to see what are the trends that are happening with the genders across the income range

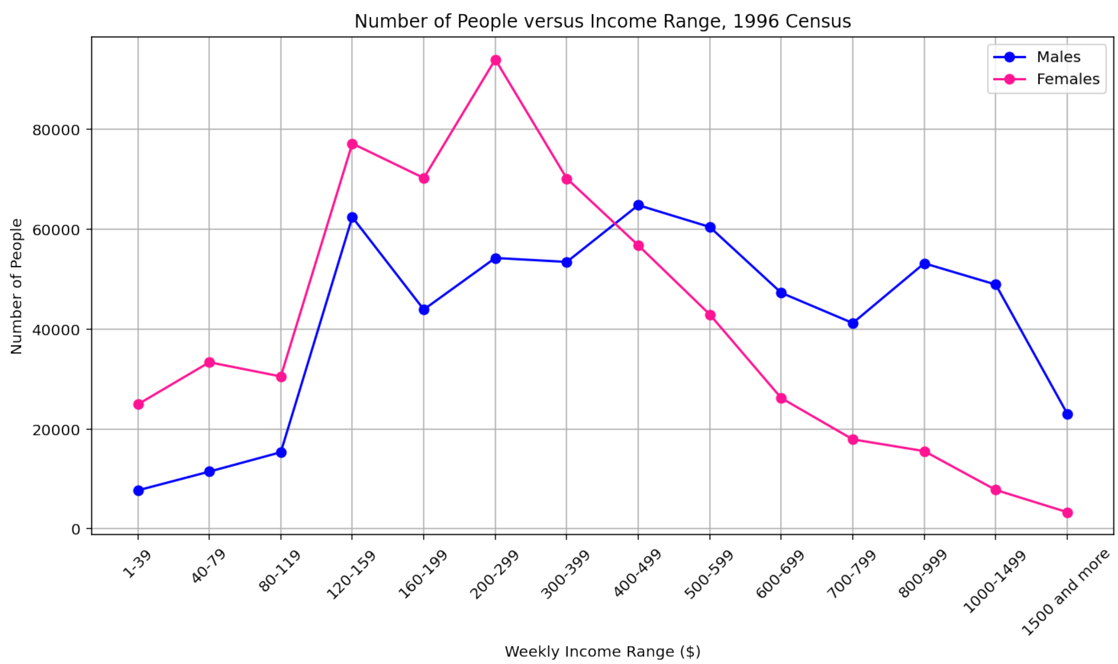
- Inside each years' data, we used the income range as the x-axis and the number of people as the y-axis
- The dots represent the number of people in that income range, and the colour of the dots follows the colour of the lines that are a part of
  * There are two colours, one blue line for the MALE gender and one pink line for the FEMALE gender
- With this, we are able to visualise
  * which gender dominates more than the other in each category
  * Across the x-axis, where the income range increases, we get to see what changes are happening to the amount of people in the income range based on gender.
- A brief description of the graphs is provided in the Results section.

[6]:
```
### RAW INCOME RANGE ##

# 1996 & 2001 income range
income_ranges_1996 = ["1-39", "40-79", "80-119", "120-159", "160-199",␣
↪"200-299", "300-399", "400-499", "500-599", "600-699", "700-799", "800-999",␣
↪"1000-1499","1500 and more"]

# 2011 income range
income_ranges_2011 = ["1-199", "200-299", "300-399", "400-599", "600-799",␣
↪"800-999", "1000-1249", "1250-1499", "1500-1999", "2000 and more"]

# 2016 income range
income_ranges_2016 = ["1-149", "150-299", "300-399", "400-499", "500-649",␣
↪"650-799", "800-999", "1000-1249", "1250-1499", "1500-1749", "1750-1999",␣
↪"2000-2999", "3000 and more"]

# 2021 income ranges
income_ranges_2021 = ["1-149", "150-299", "300-399", "400-499", "500-649",␣
↪"650-799", "800-999", "1000-1249", "1250-1499", "1500-1749", "1750-1999",␣
↪"2000-2999", "3000-3499", "3500 and more"]
```

[7]:
```
# 1996 DATA PLOT - TOTAL NO. OF PEOPLE (Y) VS. INCOME RANGE(X)
import matplotlib.pyplot as plt

# Extracting income ranges and 1996 data
income_ranges = ["1-39", "40-79", "80-119", "120-159", "160-199", "200-299",␣
↪"300-399", "400-499", "500-599", "600-699", "700-799", "800-999",␣
↪"1000-1499","1500 and more"]
males_count, females_count = data_1996


males_values = [] # store the total value of each income category for males
for item in males_count[:-2]: # exclude the last two lists
    total_m = item[-1]
    males_values.append(total_m)
```

```
females_values = [] # store the total value of each income category for females
for item2 in females_count[:-2]:
    total_f = item2[-1]
    females_values.append(total_f)


# Creating the plot
plt.figure(figsize=(12, 6))

plt.plot(income_ranges, males_values, marker = 'o', label='Males',color='b')
plt.plot(income_ranges, females_values, marker = 'o',␣
 ↪label='Females',color='deeppink')

plt.xlabel('Weekly Income Range ($)')
plt.ylabel('Number of People')
plt.title('Number of People versus Income Range, 1996 Census')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

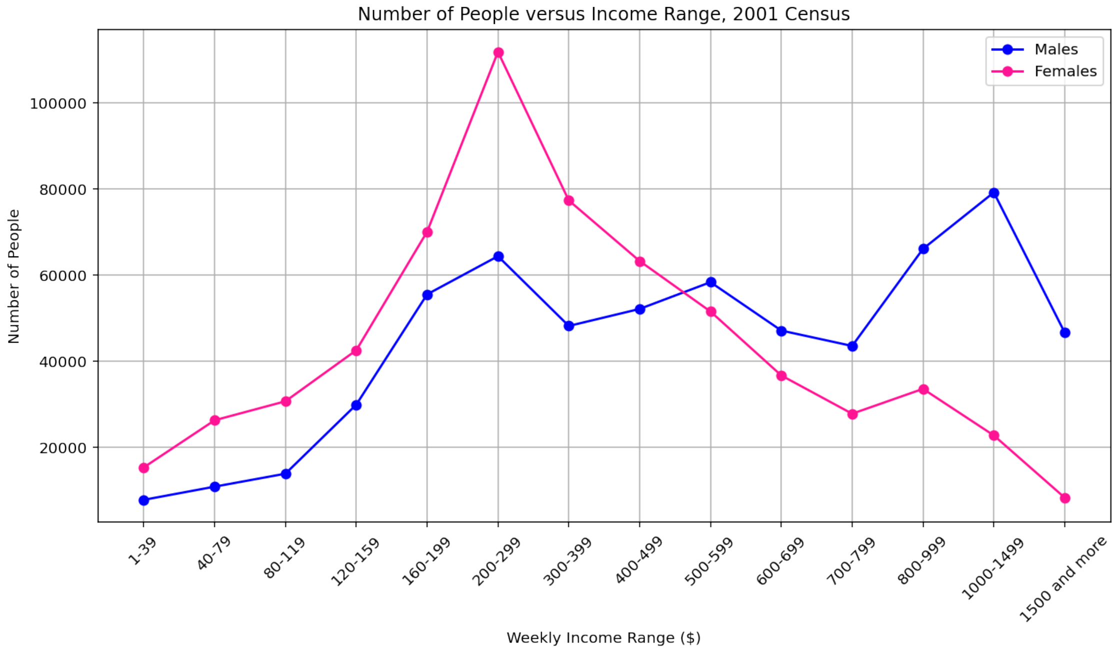
[7]:



[8]:

```python
# 2001 DATA PLOT - TOTAL NO. OF PEOPLE (Y) VS. INCOME RANGE(X)
import matplotlib.pyplot as plt

# Extracting income range and 2001 data
income_ranges = ["1-39", "40-79", "80-119", "120-159", "160-199", "200-299", "300-399", "400-499", "500-599", "600-699", "700-799", "800-999", "1000-1499","1500 and more"]
males_count, females_count = data_2001


males_values = [] # store the total value of each income category for males
for item in males_count[:-2]: # exclude the last two lists
    total_m = item[-1]
    males_values.append(total_m)

females_values = [] # store the total value of each income category for females
for item2 in females_count[:-2]:
    total_f = item2[-1]
    females_values.append(total_f)



# Creating the plot
plt.figure(figsize=(12, 6))

plt.plot(income_ranges, males_values, marker = 'o', label='Males',color='b')
plt.plot(income_ranges, females_values, marker = 'o', label='Females',color='deeppink')

plt.xlabel('Weekly Income Range ($)')
plt.ylabel('Number of People')
plt.title('Number of People versus Income Range, 2001 Census')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

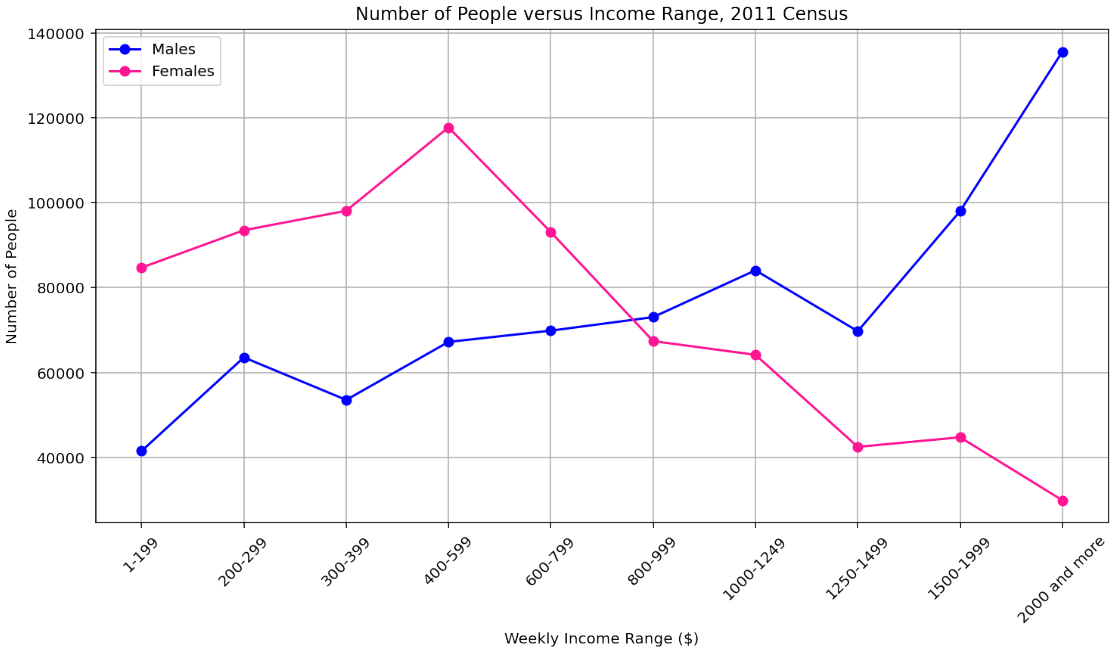
[8]:



[9]:
```
# 2011 DATA PLOT - TOTAL NO. OF PEOPLE (Y) VS. INCOME RANGE(X)
import matplotlib.pyplot as plt

# Extracting income category and 2011 data
income_ranges = ["1-199", "200-299", "300-399", "400-599", "600-799",␣
 ↪"800-999", "1000-1249", "1250-1499", "1500-1999", "2000 and more"]
males_count, females_count = data_2011

males_values = [] # store the total value of each income category for males
for item in males_count[:-2]: # exclude the last two lists
    total_m = item[-1]
    males_values.append(total_m)

females_values = [] # store the total value of each income category for females
for item2 in females_count[:-2]:
    total_f = item2[-1]
    females_values.append(total_f)


# Creating the plot
plt.figure(figsize=(12, 6))

plt.plot(income_ranges, males_values, marker = 'o', label='Males',color='b')
plt.plot(income_ranges, females_values, marker = 'o',␣
 ↪label='Females',color='deeppink')
```

```
plt.xlabel('Weekly Income Range ($)')
plt.ylabel('Number of People')
plt.title('Number of People versus Income Range, 2011 Census')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

[9]:

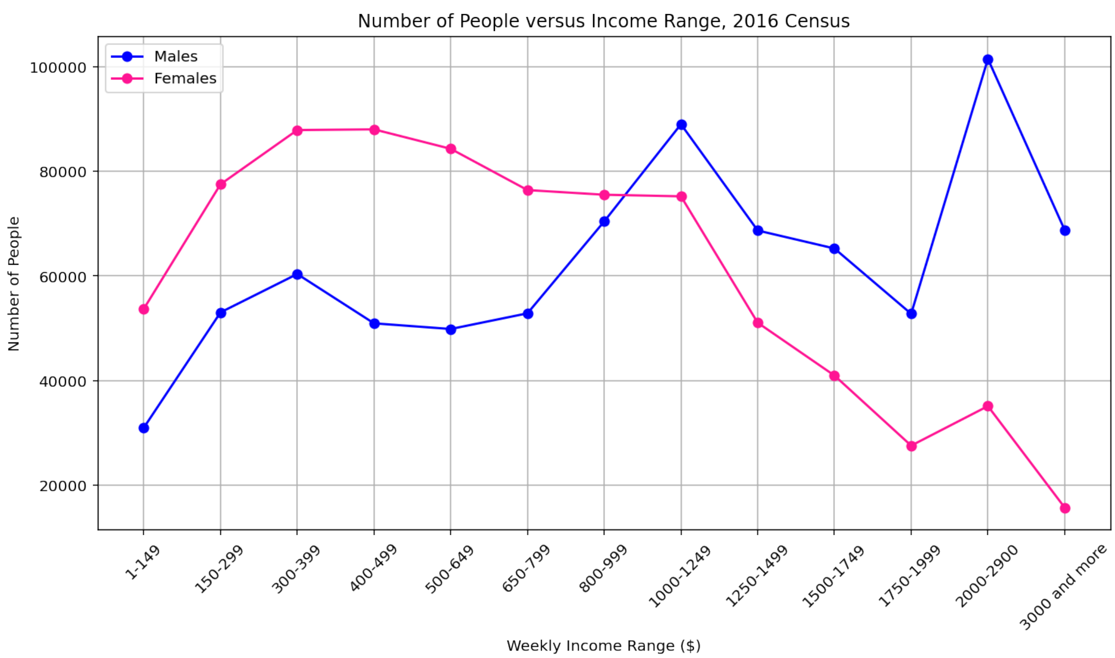


[10]:

```
# 2016 DATA PLOT - TOTAL NO. OF PEOPLE (Y) VS. INCOME RANGE(X)
import matplotlib.pyplot as plt

# Extracting income ranges and 2016 data
income_ranges = ["1-149", "150-299", "300-399", "400-499", "500-649",␣
↪"650-799", "800-999", "1000-1249", "1250-1499", "1500-1749", "1750-1999",␣
↪"2000-2900", "3000 and more"]
males_count, females_count = data_2016


males_values = [] # store the total value of each income category for males
for item in males_count[:-2]: # exclude the last two lists
    total_m = item[-1]
    males_values.append(total_m)
```

```
females_values = [] # store the total value of each income category for females
for item2 in females_count[:-2]:
    total_f = item2[-1]
    females_values.append(total_f)



# Creating the plot
plt.figure(figsize=(12, 6))

plt.plot(income_ranges, males_values, marker = 'o', label='Males',color='b')
plt.plot(income_ranges, females_values, marker = 'o',␣
 ↪label='Females',color='deeppink')

plt.xlabel('Weekly Income Range ($)')
plt.ylabel('Number of People')
plt.title('Number of People versus Income Range, 2016 Census')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

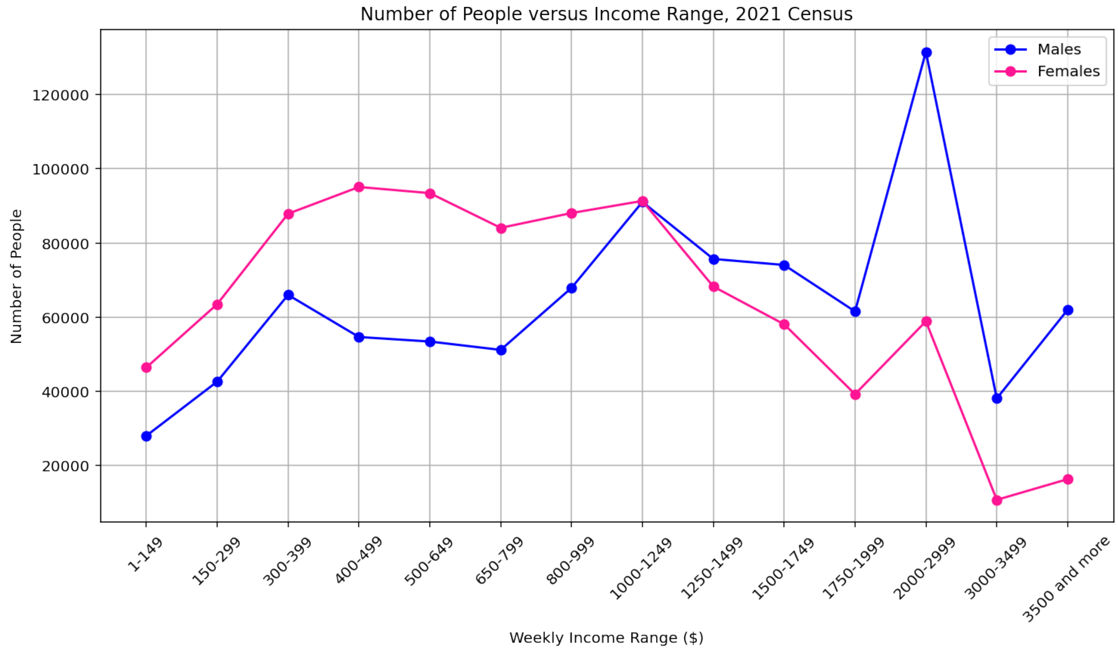
[10]:



[11]:

```python
# 2021 DATA PLOT - TOTAL NO. OF PEOPLE (Y) VS. INCOME RANGE(X)
import matplotlib.pyplot as plt

# Extracting income ranges and 2021 data
income_ranges = ["1-149", "150-299", "300-399", "400-499", "500-649",␣
↪"650-799", "800-999", "1000-1249", "1250-1499", "1500-1749", "1750-1999",␣
↪"2000-2999", "3000-3499", "3500 and more"]
males_count, females_count = data_2021


males_values = [] # store the total value of each income category for males
for item in males_count[:-2]: # exclude the last two lists
    total_m = item[-1]
    males_values.append(total_m)

females_values = [] # store the total value of each income category for females
for item2 in females_count[:-2]:
    total_f = item2[-1]
    females_values.append(total_f)


# Creating the plot
plt.figure(figsize=(12, 6))

plt.plot(income_ranges, males_values, marker = 'o', label='Males',color='b')
plt.plot(income_ranges, females_values, marker = 'o',␣
↪label='Females',color='deeppink')

plt.xlabel('Weekly Income Range ($)')
plt.ylabel('Number of People')
plt.title('Number of People versus Income Range, 2021 Census')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

[11]:



4 Data cleaning and formatting Part Two:

This section combines all the data into desired income categories so that all categories across all data sets are consistent.

This is done by extracting the data from each income range, removing the category name, re-assigning the new format for the income category, and re-assigning new data into a new list.

1. The function ***new_categories(data)*** re-formats the income category for 1996 and 2001.
2. The function ***cat_combine(data_year)*** re-formats the 2011, 2016 and 2021 income category.

[12]:
```
## COMBINING THE DATA INTO DESIRED CATEGORIES, AND SO THAT ALL OF THEM HAVE THE␣
↪SAME CATEGORIES AS EACH OTHER, AND ARE STANDARDISED THE WAY WE WANT IT



# re-format income categories for 1996 and 2001 data:
def new_categories(data):
    # FIX MALE DATA
    new_data = []
    cat_16 = data[0][0:6] # extract data for income range $ 1- $299
    cat_811 = data[0][7:11] # extract data for income range $ 400-$799
    for x in cat_16: # remove category name
        x.pop(0)
    for x in cat_811:
        x.pop(0)
```

```
    new_M = []
    new_M_cat16 = []
    new_M_cat16.append('M_1_299') # re-assign new income category
    summed = np.sum(cat_16,0) # sum all counts into one list
    summed = list(summed)
    new_M_cat16.extend(summed) # re-assign new data to a new list
    new_M.append(new_M_cat16)
    new_M.append(data[0][6])

    new_M_cat811 = []
    new_M_cat811.append('M_400_799')
    summed = np.sum(cat_811,0)
    summed = list(summed)
    new_M_cat811.extend(summed)
    new_M.append(new_M_cat811)
    new_M.extend(data[0][11:])

    new_data.append(new_M)

    # FIX FEMALE DATA
    cat_16 = data[1][0:6]
    cat_811 = data[1][7:11]
    for x in cat_16:
        x.pop(0)
    for x in cat_811:
        x.pop(0)
    new_f = []
    new_f_cat16 = []
    new_f_cat16.append('F_1_299')
    summed = np.sum(cat_16,0)
    summed = list(summed)
    new_f_cat16.extend(summed)
    new_f.append(new_f_cat16)
    new_f.append(data[1][6])

    new_f_cat811 = []
    new_f_cat811.append('F_400_799')
    summed = np.sum(cat_811,0)
    summed = list(summed)
    new_f_cat811.extend(summed)
    new_f.append(new_f_cat811)
    new_f.extend(data[1][11:])

    new_data.append(new_f)

    return new_data
```

```
# !!!!!!!!!!!  BELOW ARE FUNCTIONS FOR RE FORMATTING data_2011, data_2016 and
↪data_2021 !!!!!!!!!!!!!!!

# AN EXAMPLE FOR THE data_2011

# TRYING TO GET FROM
# [['M_1_199', 18411, 5978, 2945, 2288, 2447, 3629, 3742, 2060, 41500],
↪['M_200_299', 4875, 6205, 5453, 4692, 5088, 8212, 15554, 13463, 63542],
↪['M_300_399', 4787, 4970, 4351, 3901, 4982, 7278, 12233, 11031, 53533],
↪['M_400_599', 6347, 8410, 7998, 6425, 7084, 9952, 12410, 8568,
↪67194],['M_600_799', 3365, 10720, 13794, 10311, 10332, 10991, 6745, 3588,
↪69846],....]]

# TO HERE, WHERE THE RANGES ARE WHAT WE PREFERRED TO USE
#[['M_1_299', 23286, 12183, 8398, 6980, 7535, 11841, 19296, 15523, 105042],
↪['M_300_399', 4787, 4970, 4351, 3901, 4982, 7278, 12233, 11031, 53533],
↪['M_400_799', 9712, 19130, 21792, 16736, 17416, 20943, 19155, 12156,
↪137040],....]]

# brute forcing here, because it seems to be the most simplest fix

for_comb_data_2011=([['M_1_199', 18411, 5978, 2945, 2288, 2447, 3629, 3742,
↪2060, 41500], ['M_200_299', 4875, 6205, 5453, 4692, 5088, 8212, 15554,
↪13463, 63542], ['M_300_399', 4787, 4970, 4351, 3901, 4982, 7278, 12233,
↪11031, 53533], ['M_400_599', 6347, 8410, 7998, 6425, 7084, 9952, 12410,
↪8568, 67194], ['M_600_799', 3365, 10720, 13794, 10311, 10332, 10991, 6745,
↪3588, 69846], ['M_800_999', 1384, 9110, 17283, 13690, 13260, 11839, 4553,
↪1913, 73032], ['M_1000_1249', 783, 7883, 21310, 18230, 17165, 13748, 3729,
↪1211, 84059], ['M_1250_1499', 313, 4737, 18532, 17205, 15500, 10500, 2316,
↪612, 69715], ['M_1500_1999', 278, 4409, 23911, 27043, 24417, 14863, 2475,
↪595, 97991], ['M_2000_more', 192, 3607, 26581, 41264, 38020, 21382, 3685,
↪831, 135562], ['M_PI_NS', 8299, 8975, 18661, 16201, 13708, 10057, 5949,
↪6614, 88464], ['M_Tot', 76472, 82185, 165962, 165033, 155804, 127319, 75833,
↪51630, 900238]], [['F_1_199', 22721, 7976, 13427, 14697, 8501, 8964, 5027,
↪3374, 84687], ['F_200_299', 5929, 7831, 10197, 11619, 10210, 15439, 18581,
↪13696, 93502], ['F_300_399', 4257, 6808, 10042, 12052, 11299, 13307, 18667,
↪21634, 98066], ['F_400_599', 5137, 11078, 17851, 21224, 20635, 17308, 13028,
↪11467, 117728], ['F_600_799', 2413, 12103, 17394, 18991, 20058, 13605, 4945,
↪3554, 93063], ['F_800_999', 726, 8274, 15305, 13845, 15555, 9786, 2395,
↪1498, 67384], ['F_1000_1249', 257, 5978, 16943, 13533, 15333, 9575, 1722,
↪810, 64151], ['F_1250_1499', 118, 2191, 12505, 10032, 10325, 5847, 1011,
↪442, 42471], ['F_1500_1999', 70, 1215, 11723, 11446, 12102, 6796, 963, 418,
↪44733], ['F_2000_more', 57, 599, 6537, 8797, 8167, 4210, 984, 542, 29893],
↪['F_PI_NS', 6637, 5580, 11333, 10843, 9321, 8132, 5718, 11333, 68897],
↪['F_Tot', 71732, 76824, 158133, 163220, 154429, 127110, 76713, 70703,
↪898864]])
```

```python
for_comb_data_2016=([['M_1_149', 14762, 4948, 1963, 1346, 1579, 2553, 2434,
1127, 30943], ['M_150_299', 6327, 8510, 7557, 5192, 5446, 7014, 7651, 5190,
53029], ['M_300_399', 3023, 5364, 5119, 3708, 4586, 6777, 16240, 15437,
60373], ['M_400_499', 2988, 5289, 5461, 3840, 4504, 6211, 12260, 10250,
50933], ['M_500_649', 2842, 6933, 7352, 4762, 5018, 6361, 9914, 6479,
49847], ['M_650_799', 1659, 7993, 10725, 6788, 6533, 7055, 7340, 4577,
52849], ['M_800_999', 972, 8511, 17879, 11693, 11419, 10394, 6690, 2709,
70451], ['M_1000_1249', 486, 7306, 24148, 17779, 16489, 13905, 6648, 2263,
89056], ['M_1250_1499', 146, 3836, 19070, 15729, 14177, 10655, 3908, 1050,
68707], ['M_1500_1749', 64, 2340, 17287, 16078, 15129, 10610, 2924, 707,
65266], ['M_1750_1999', 22, 1265, 12829, 14175, 12920, 8808, 2097, 465,
52755], ['M_2000_2999', 38, 1691, 22845, 29514, 27071, 16292, 3348, 744,
101570], ['M_3000_more', 70, 615, 9723, 20479, 21504, 12290, 3159, 866,
68812], ['M_PI_NS', 7309, 7882, 21004, 17251, 15981, 12596, 9436, 9544,
101121], ['M_Tot', 77141, 82558, 190959, 173491, 168148, 139001, 98766,
63742, 993722]], [['F_1_149', 18162, 5235, 6773, 7149, 5235, 5904, 3399,
1678, 53704], ['F_150_299', 8037, 9224, 11674, 10991, 9549, 11698, 9847,
6409, 77560], ['F_300_399', 3084, 6622, 10074, 9795, 9434, 11630, 20734,
16513, 87940], ['F_400_499', 2355, 6508, 10556, 10260, 10227, 10799, 17626,
19683, 88076], ['F_500_649', 2165, 8126, 14976, 14357, 13680, 11702, 11071,
8150, 84347], ['F_650_799', 1204, 8519, 15689, 14267, 14560, 11019, 6846,
4350, 76449], ['F_800_999', 547, 7637, 17710, 14813, 16125, 11329, 4583,
2685, 75556], ['F_1000_1249', 205, 5542, 19951, 15040, 16700, 12293, 3707,
1811, 75252], ['F_1250_1499', 50, 2962, 14693, 11074, 11758, 7840, 1978,
569, 51051], ['F_1500_1749', 29, 1051, 11493, 9638, 10006, 6553, 1475, 484,
40938], ['F_1750_1999', 6, 416, 6860, 6922, 7190, 4687, 917, 326, 27516],
['F_2000_2999', 3, 327, 7555, 9528, 9813, 5989, 1261, 506, 35081],
['F_3000_more', 29, 155, 1980, 4552, 4525, 2633, 995, 619, 15662],
['F_PI_NS', 6441, 6538, 15768, 12411, 12284, 11211, 9829, 15602, 90215],
['F_Tot', 72896, 77779, 188222, 171552, 166853, 143056, 100947, 82810,
1004054]])
```

```
for_comb_data_2021=([['M_1_149', 14722, 2912, 1159, 895, 1066, 2136, 3233,␣
↪1727, 28042], ['M_150_299', 8331, 6589, 4370, 3354, 3576, 4848, 6818, 4671,␣
↪42644], ['M_300_399', 3327, 5704, 5603, 4325, 4908, 7171, 17116, 17622,␣
↪66012], ['M_400_499', 3233, 4996, 4598, 3402, 4114, 6127, 14249, 13753,␣
↪54661], ['M_500_649', 3979, 6951, 6355, 4273, 4573, 6008, 11880, 9299,␣
↪53444], ['M_650_799', 2720, 8170, 8236, 5420, 4758, 6230, 8851, 6622,␣
↪51190], ['M_800_999', 1903, 9514, 15271, 10049, 8920, 9362, 8258, 4426,␣
↪67918], ['M_1000_1249', 1093, 8803, 22808, 17172, 14424, 14370, 8944, 3434,␣
↪91074], ['M_1250_1499', 425, 5310, 19642, 16588, 13739, 12160, 5863, 1863,␣
↪75700], ['M_1500_1749', 237, 3353, 18340, 17802, 15073, 12877, 4725, 1548,␣
↪74077], ['M_1750_1999', 96, 1952, 14097, 16218, 14239, 10825, 3298, 789,␣
↪61550], ['M_2000_2999', 113, 2639, 26616, 38079, 33582, 23482, 5644, 1182,␣
↪131525], ['M_3000_3499', 3, 391, 6485, 11601, 10572, 6771, 1662, 357,␣
↪38100], ['M_3500_more', 52, 293, 6053, 17918, 20084, 12732, 3570, 1104,␣
↪62002], ['M_PI_NS', 4995, 7125, 18976, 17660, 15488, 12210, 8099, 8861,␣
↪93384], ['M_Tot', 78733, 81170, 184219, 188897, 173948, 154764, 120462,␣
↪81753, 1063950]], [['F_1_149', 17635, 3344, 4094, 5022, 3750, 5221, 4874,␣
↪2273, 46448], ['F_150_299', 10213, 7496, 7608, 7881, 6539, 8513, 9248, 5957,␣
↪63522], ['F_300_399', 4332, 7022, 8677, 8916, 8561, 11670, 21647, 16968,␣
↪87893], ['F_400_499', 3050, 6434, 9305, 9365, 8873, 11051, 21371, 25392,␣
↪95101], ['F_500_649', 2905, 8700, 14671, 14310, 12633, 12393, 14913, 12796,␣
↪93447], ['F_650_799', 1838, 8422, 15971, 15707, 13880, 11864, 9673, 6533,␣
↪84084], ['F_800_999', 1189, 8923, 19077, 17676, 16618, 13177, 7234, 3964,␣
↪88080], ['F_1000_1249', 593, 7516, 21858, 19446, 18739, 15115, 5835, 2220,␣
↪91349], ['F_1250_1499', 219, 4635, 17861, 15329, 14365, 11022, 3443, 1088,␣
↪68268], ['F_1500_1749', 132, 2348, 15414, 13785, 13057, 9489, 2655, 1120,␣
↪58083], ['F_1750_1999', 23, 819, 9843, 10055, 9589, 6477, 1704, 466, 39250],␣
↪['F_2000_2999', 32, 866, 11689, 16750, 16254, 10117, 2353, 680, 58892],␣
↪['F_3000_3499', 0, 123, 1732, 3151, 2981, 1931, 553, 268, 10769],␣
↪['F_3500_more', 40, 92, 1479, 4692, 5169, 2922, 1101, 644, 16347],␣
↪['F_PI_NS', 4433, 5235, 12694, 10899, 9634, 8522, 7782, 13208, 72484],␣
↪['F_Tot', 74591, 77761, 188516, 190925, 174615, 159000, 127189, 99531,␣
↪1092165]])

def cat_combine(data_year):
    # changing order based on the data set
    if data_year==data_2016:
        order=[2,1,3,1,2,5]
    elif data_year==data_2021:
        order=[2,1,3,1,2,6]
    else:
        order=[2,1,2,1,2,3]


    formatted_M_F=[]
    for cat in data_year:
        # removing the not_stated category
```

```
        cat.pop(-2)
        n=0
        cat_format=[]
        # formatting the data set so it falls in the bins we want to work with
        for idx in order:
            cat_format.append(cat[n:n+idx])
            n=n+idx
        formatted_M_F.append(cat_format)


    # sperating out the the list that contains data about total amount of people from the data list we want to work with
    Total_M_F=[]
    for cats in formatted_M_F:
        Total_M_F.append(cats[-1].pop(-1))

    # header range formatting and also values combination
    cat_head_M_F=[]
    cat_value_combined=[]
    for group in formatted_M_F:
        for cat in group:
            cat_split_list=[]
            cat_value_lst=[]
            for i in cat: # iterating through the each single list of data
                cat_value=i[1:] # taking out the values of the category
                cat_value_lst.append(cat_value) # put them in a list

                # category header reformatting for desired range
                cat_head=i[0][2:]
                cat_split_list.extend(cat_head.split('_'))
                cat_min_max='_'.join([min(cat_split_list),max(cat_split_list)])

            # summation of the values of the categories that got combined
            value_summed = [sum(values) for values in zip(*cat_value_lst)]

            # put them all into a list
            cat_value_combined.append(value_summed)
            cat_head_M_F.append(cat_min_max)

    # separating the data into Male and Female
    M_header=['M_'+header for header in cat_head_M_F[:6]]
    F_header=['F_'+header for header in cat_head_M_F[6:]]

    M_value=[value for value in cat_value_combined[:6]]
    F_value=[value for value in cat_value_combined[6:]]

    # combining the new category headers with the combined values
```

```
    M_header_value_total=[]
    for head,lst in zip(M_header,M_value):
        lst.insert(0,head)
        M_header_value_total.append(lst)

    F_header_value_total=[]
    for head,lst in zip(F_header,F_value):
        lst.insert(0,head)
        F_header_value_total.append(lst)

    M_header_value_total.append(Total_M_F[0])
    F_header_value_total.append(Total_M_F[1])

    return M_header_value_total,F_header_value_total


new_data_2011=cat_combine(for_comb_data_2011)
new_data_2016=cat_combine(for_comb_data_2016)
new_data_2021=cat_combine(for_comb_data_2021)
new_data_1996 = new_categories(data_1996)
new_data_2001 = new_categories(data_2001)
```

5 Part Two: Visualisation of the reformatted data for the bar chart and the line graph

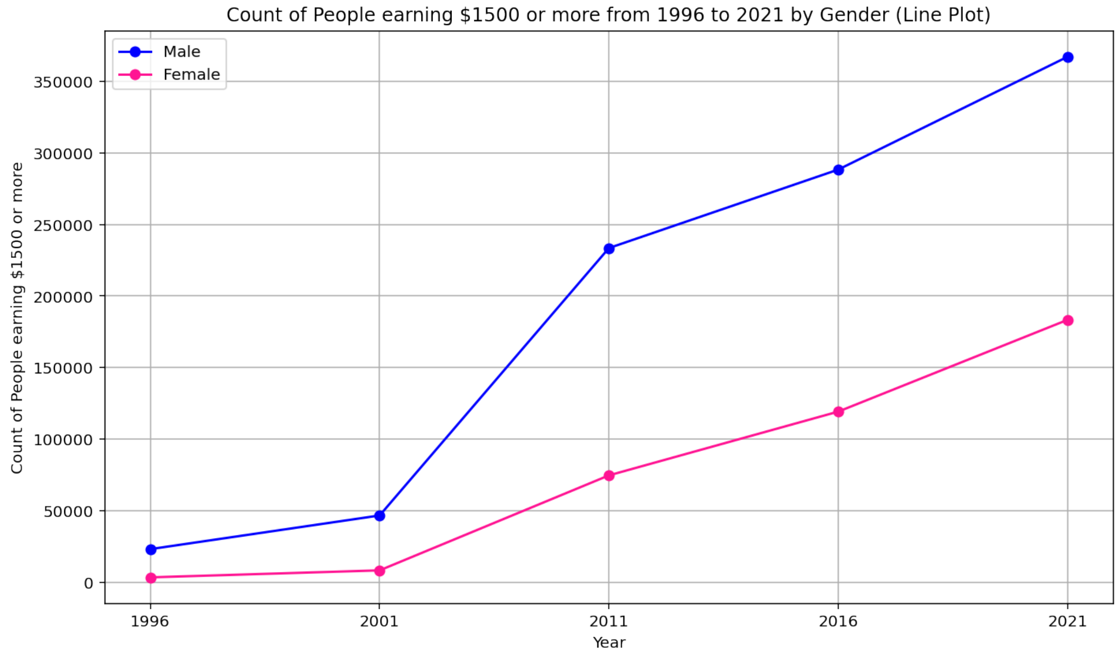
[13]:
```
######################### CREATE THE BAR GRAPH SHOWING NUMBER OF PEOPLE EARNING␣
↪$1500 OR MORE FROM 1996-2021 BY GENDE #######################

# Extract data for '1500 or more' category for each year
male_counts = []
female_counts = []

for data_year in [new_data_1996, new_data_2001, new_data_2011, new_data_2016,␣
↪new_data_2021]:
    male_count_year = 0
    female_count_year = 0
    for category_data in data_year[0]:  # Male data
        if category_data[0] == 'M_1500_more':
            male_count_year += category_data[-1]  # Sum the count for males
    for category_data in data_year[1]:  # Female data
        if category_data[0] == 'F_1500_more':
            female_count_year += category_data[-1]  # Sum the count for females
    male_counts.append(male_count_year)
    female_counts.append(female_count_year)
```

```
# Create the bar graph
years = ['1996', '2001', '2011', '2016', '2021']
bar_width = 0.35  # Width of each bar

fig, ax = plt.subplots(figsize=(10, 6))

# Calculate the x positions for the bars
x = range(len(years))

# Plot male counts as one set of bars
bar1 = ax.bar(x, male_counts, bar_width, label='Male', color = "skyblue")

# Plot female counts as another set of bars
bar2 = ax.bar([i + bar_width for i in x], female_counts, bar_width,␣
 ↪label='Female', color = "lightpink")

ax.set_xlabel('Year')
ax.set_ylabel('Number of People Earning > $1500  ')
ax.set_title('Number of People Earning > $1500 from 1996-2021 by Gender')
ax.set_xticks([i + bar_width / 2 for i in x])
ax.set_xticklabels(years)
ax.legend()

# Display the graph
plt.tight_layout()
plt.show()

############## LINE PLOT ######################

male_counts = []
female_counts = []

for data_year in [new_data_1996, new_data_2001, new_data_2011, new_data_2016,␣
 ↪new_data_2021]:
    male_count_year = 0
    female_count_year = 0
    for category_data in data_year[0]:  # Male data
        if category_data[0] == 'M_1500_more':
            male_count_year += category_data[-1]  # Sum the count for males
    for category_data in data_year[1]:  # Female data
        if category_data[0] == 'F_1500_more':
            female_count_year += category_data[-1]  # Sum the count for females
    male_counts.append(male_count_year)
    female_counts.append(female_count_year)

# Create the bar graph
years = ['1996', '2001', '2011', '2016', '2021']
```

```
fig, ax = plt.subplots(figsize=(10, 6))

# Plot male counts as a line
ax.plot(years, male_counts, label='Male', marker='o', linestyle='-', color =␣
↪"b")

# Plot female counts as another line
ax.plot(years, female_counts, label='Female', marker='o', linestyle='-', color␣
↪= "deeppink")

ax.set_xlabel('Year')
ax.set_ylabel('Count of People earning $1500 or more')
ax.set_title('Count of People earning $1500 or more from 1996 to 2021 by Gender␣
↪(Line Plot)')
ax.legend()

# Display the graph
plt.tight_layout()
plt.grid(True)
plt.show()
```

[13]:

[13]:



6 Results

6.0.1 Part One:

Graph 1 (1996):

In 1996, the income distribution graph reveals the number of people in each weekly income category. Initially, females led the way, peaking at more than 90,000 individuals in the $200 - $299 category. However, a shift occurs beyond the $399 level as the number of males increases, surpassing that of females. This highlights that as income levels climb, the number of males within that income level also increases. In comparison, the number of females decreases at each higher income level.

Graph 2 (2001):

The 2001 weekly income distribution graph follows the 1996 trend. However, the notable distinction in gender numbers occurs at a higher income threshold, specifically beyond $499. Moreover, the gender gap difference significantly widens at $800 weekly income and higher, marking a remarkable divergence in male and female representation.

Graph 3 (2011):

The weekly income data for both genders for 2011 is portrayed above. The graph shows a considerable difference in weekly income between males and females. Women in the lowest income levels seem more dominant than males. On the other hand, the trend reverses after exceeding $800 weekly income, where the number of males significantly increases as the payment gets higher and females gradually decrease, reaching less than 30,000 females with weekly income higher than $2000 and above.

Graph 4 (2016):

The 2021 weekly income distribution shows a parallel increase for both genders, with females leading in the number of individuals in the $1 to $399 weekly income range. A similar parallel decline is observed in the number of males and females with a payment between $1249 to $1999 weekly. Nevertheless, the real twist happens as the number of males peaked at approximately 100,000 men with a weekly income between $2000 to $2900, compared to less than 40,000 females. This difference amounts to more than 60,000 persons, illustrating a significant gender gap.

Graph 5 (2021):

The 2021 graph mirrors the trend observed in the 2016 data regarding the number of people versus their weekly income range. The graph still shows females as more dominant in the lower income level, with a shift point beyond $1249 weekly income. At this point, both groups meet at roughly 90,000 individuals with incomes ranging from $1000 to $1249 weekly. The male group reaches its highest point of 120,000 males with income between $2000 to $2999. Subsequently, there is a sharp decline in higher income levels, where females remain lower than males.

On an overall basis,

- Over the 5 graphs, we can see that the number of women is always on a downtrend while the income range increases and the number of men is always increasing with the income range.
- There are always more men and women with increasing income ranges throughout 1996 and 2021, hence showing there is a disparity between men and women when it comes to higher-income jobs
- **Based on Graphs of 1996 and 2001 data**
  - The above statements are more prominent in 1996 and 2001, where women's numbers peaked at around the $200-$600 weekly income range, and men peaked at $1000 and above while maintaining the huge difference in men employed in high-income jobs. Whereas women are more employed in low-income jobs
- **Based on Graphs of 2011 to 2021**
  - As the line plots have shown for all the years, men always have the higher count in high-income ranges, more so after the $1500 weekly income mark, as can be seen in the plots from 2011, 2016, and 2021.
  - Also noting that the sharp increase can be justified by the median weekly salary at about $1700 for a full-time adult in Perth, in reference to (Statistics, 2013) and (Average Salary in Perth 2021 - the Complete Guide, n.d.). Therefore, even that shows women are far behind in terms of how many of them are in those median pay ranges when compared to men

6.0.2 Part Two:

This section plots the number of Males and Females earning a weekly income of $1500 and more for all years.

Graph 1 (Bar Graph):

This bar graph illustrates the number of individuals, categorized by gender (Male and Female), who earned a weekly income exceeding $1500 across the years 1996 to 2021. Each bar on the x-axis represents a specific year, while the y-axis represents the count of people falling within this income bracket. One notable observation from this graph is the persistent gender gap in income. Over the entire period, it is evident that males consistently outnumber females in earning above $1500 weekly. Additionally, the gap appears to grow wider over time. While both genders have experienced an increase in this income bracket, the rate of increase among males seems to outpace that of females. This trend highlights the persistence of gender-based income disparities and suggests a growing divide, particularly after the year 2001.

Graph 2 (Line Graph):

This line graph visualizes the evolution of the gender-based income disparity in the category of weekly earnings exceeding $1,500, covering the years from 1996 to 2021. The x-axis denotes the years, while the y-axis represents the count of individuals earning above $1500 weekly. The analysis of this graph reveals an interesting pattern. Initially, from 1996 to 2001, there is a gradual increase in the income gap between males and females, indicating that more males were consistently earning higher than females. However, after the year 2001, there is a sharp and noticeable spike in this gap. What stands out is that, after this point, the gap remains large and appears to grow at a consistent rate for both males and females. This suggests that the gender income disparity, especially in the higher income brackets, not only persisted but accelerated significantly after 2001.

6.0.3 Discussion:

Our analysis of income distribution trends across genders from 1996 to 2021 provided substantial insights directly related to our original hypothesis. The hypothesis stated that higher-income brackets would be more dominated by males, while lower-income brackets would see a higher prevalence of females. Our findings robustly support this hypothesis, reaffirming the persistent existence of gender-based income discrepancies. In examining the line plots from Graph 1 (1996) to Graph 3 (2011), we observe a consistent pattern: more females tend to occupy lower income categories than males. However, after a certain threshold and as income levels rise, the number of males within those income categories increases while the number of females decreases. This trend aligns perfectly with our initial hypothesis, indicating that gender indeed plays a significant role in determining income distribution. This shift also highlights the enduring challenges women face in accessing higher-paying roles and underscores the urgent need to address these disparities. Moreover, the line graph showing the count of people in higher income categories by gender from 1996 to 2021 clearly illustrates that these income disparities have not diminished over time; rather, they have intensified. This finding emphasises that gender-based income disparities, particularly in higher income brackets, not only persist but are growing despite changes in societal norms, policies and legislations, and economic conditions over the years. This observation serves as a stark reminder of the ongoing gender pay gap, a long-standing issue that requires continued attention and action.

According to Workplace Gender Equality Agency (WGEA), there can be several factors affecting such income disparities and gender pay gap. The Western Australian full-time workforce is characterized by a significant presence in industries like mining and construction, which are known for their relatively high earnings but also for having low representation of women [7]. This industry-specific composition contributes to the gender pay gap as these sectors tend to offer fewer opportunities for women to access higher-paying roles. Discrimination and bias in hiring and pay decisions also play a role in perpetuating the gender pay gap. This bias can lead to women being paid less than their male counterparts for similar work, even within the same industries. Additionally, the phenomenon of occupational segregation, where women and men work in different industries and jobs, has a direct impact on the gender pay gap. Female-dominated industries and jobs often attract lower wages, exacerbating income disparities. Women's disproportionate share of unpaid caregiving and domestic work further compounds the gender pay gap. This additional workload can limit women's availability for full-time employment or career advancement, affecting their overall earnings potential. In senior roles, the lack of workplace flexibility to accommodate caring and other responsibilities can act as a barrier to women's career progression. Furthermore, women's greater time out of the workforce, often due to caregiving responsibilities, can significantly impact their career progression and opportunities for higher-paying positions. Career interruptions can result in reduced work experience and seniority, which can translate into lower earnings.

To address the gender pay gap and income disparities in Western Australia and beyond, it is essential to tackle these multifaceted challenges. Strategies should include combating discrimination and bias in employment practices, promoting pay equity in all industries, offering workplace flexibility, and recognizing and valuing unpaid caregiving and domestic work. Additionally, addressing industry-specific gender imbalances and fostering career development opportunities for women in underrepresented sectors can contribute to narrowing the gender pay gap over time.

6.1 Conclusion :

We were able to show that even with the changing of modern times and the societal improvement we have undergone, over the past two decades, WA women are preferred less in high-income categories and more so in low-income categories. This is evident that our hypothesis, which states there is a significant difference in the distribution of income categories among different gender groups in the periods 1996 - 2021, is true in our analysis through the visualisations we provided and the justifications we stated.

6.1.1 Limitations:

- **Variation in Geographic Scope:** One notable limitation in our analysis stems from the variation in the geographic scope of the data sources. Data from 1996 and 2001 encompassed the entirety of Western Australia, while data for 2011, 2016, and 2021 focused exclusively on Local Government Areas (LGA) within the state. Hence, data for 2011-2021 is likely to have larger amount of data than data for 1996-2001 and is likely to be focused on LGA regions rather than the whole of Western Australia.
- **Quality and Representation of Early Data:** Another limitation pertains to the quality and representativeness of the 1996 and 2001 data. These datasets were collected during a period when technological advancements in data collection were not as sophisticated as in later years. Hence, the size of the sample may not be as robust as in more recent years, potentially affecting the reliability of the findings. These factors should be considered when interpreting trends and conclusions drawn from the early data.
- **Not Stated Categories:** A further limitation arises from the presence of "not stated" categories within the dataset, particularly in the weekly income category. The inclusion of these categories can introduce bias into our analysis, as they may not accurately reflect the true income distribution. This limitation requires careful consideration when drawing conclusions about income disparities between genders.
- **Societal Factors:** It is essential to acknowledge that societal factors, such as historical gender roles and stereotypes, can influence income disparities. Stereotypical expectations that men should be primary breadwinners may contribute to the observed income disparity. While our analysis provides a quantitative view of the gender pay gap, it cannot fully capture the complex sociocultural factors at play.
- **Income Range Discrepancies:** Variations in income range categories between different years' datasets present a limitation. These differences can affect the granularity of our anal-

ysis and make direct comparisons across years more challenging. Careful attention to these variations is essential to ensure meaningful interpretations of the data.

- **No Job Specifications:** Without the specifications of jobs in the data set we are unable to view the data, with the careers of the man and woman in the data set in mind, therefore we are only able to perform analysis with the raw number of people in income categories, and jobs might be a crucial factor that plays out in the gender income discrepancy we were not able to capture with our analysis.

6.2 References:

1. Gender pay gap data | WGEA. (n.d.). www.wgea.gov.au. https://www.wgea.gov.au/pay\-and\-gender/gender\-pay\-gap\-data\#:~:text=The gender pay gap in each state&text=In November 2022%3A
2. A snapshot of WA women has found they still suffer the biggest gender pay gap in the country. (2022, August 4). ABC News. https://www.abc.net.au/news/2022-08-04/wa-gender-pay-gap-still-worst-despite-more-women-working-report/101298200
3. Australian Bureau of Statistics. (2021). Search Census data | Australian Bureau of Statistics. www.abs.gov.au. https://www.abs.gov.au/census/find-census-data/search-by-area
4. Census DataPacks | Australian Bureau of Statistics. (n.d.). Www.abs.gov.au. https://www.abs.gov.au/census/find-census-data/datapacks
5. Average Salary in Perth 2021 - The Complete Guide. (n.d.). Www.salaryexplorer.com. http://www.salaryexplorer.com/salary-survey.php?loc=90&loctype=3
6. Statistics, c=AU; o=Commonwealth of A. ou=Australian B. of. (2013, December 20). Main Features - Main Features. Www.abs.gov.au. https://www.abs.gov.au/ausstats/abs@.nsf/mf/5673.0.55.003
7. Gender pay gap data | WGEA. (n.d.). www.wgea.gov.au. https://www.wgea.gov.au/pay-and-gender/gender-pay-gap-data#:~:text=This difference can be partly