CMSC 320 Homework 4: Data Exploration

Jenny Chang

October 23, 2023

1 Data Issues

While exploring the data, I found and corrected the following issues:

1.1 Differing Columns

After merging the data frames I noticed that there were an unequal number of columns between the dataframes I merged since Professor Fardina's section was asked one additional question. As a result, the columns from both Professor Fardina's section and Professor Max's section regarding the "pregnancy rules" scenario were not merged into one. To fix this, I merged them together by using the .fillna command and dropping the second column entirely. I felt this was a valid way to fix it because this combined the answers from both sections into one column without changing any of the responses, thus just making the dataframe easier and more convenient to access.

1.2 Column Names

I found that the names of the columns were quite long, particularly the columns that tell people's personal anecdotes before asking if they are a jerk. This was an issue to me because I knew that once I started performing hypothesis tests and began trying to access the data frames, it would be inconvenient for me to write out the name of the entire column. Additionally, the way the columns were titled originally, it was difficult to remember the premise of the story by quickly looking at the name of the column. I elected to fix it by renaming each column to more succinctly describe its contents or include keywords related to each story. Additionally, columns that were questions were titled QX, where X is the number of which question it was. Since I had also taken the survey, I was familiar with each of the questions already which allowed me to rename them to be even more succinct, allowing me to quickly recall the premise of a story from just glancing at the name of the column. I felt this was a valid way to fix it because this does not change the data, just the name of the columns the data is stored in. While doing so, I also switched the order of some of the columns specifically by moving the "Compassionate" column to be with the other self-description columns. I knew taking these steps would make accessing the data easier for me in the future.

1.3 Missing Values (entire row)

I found that some rows in the data set were completely empty. I knew this was a problem because I would not be able to use techniques to fill in missing values since all the values were missing. I elected to fix it by dropping the completely empty rows entirely using the .dropna(how='all') command. I felt this was a valid way to fix it because this would ensure I am reducing the number

```
Question: Q1: Doctor girlfriend, P-value: 0.2719969890730754
Question: Q2: Daughter married, P-value: 0.44445455006316903
Question: Q3: Trust fund, P-value: 0.6499718441824033
Question: Q4: Kids school, P-value: 0.06340243866679925
Question: Q5: Cat, P-value: 0.2785442569330333
Question: Q6: Niece, P-value: 0.4605445885243139
Question: Q7: Flight seats, P-value: 0.10075287657649135
Question: Q8: Child support, P-value: 0.7419883522691773
Question: Q9: Child support court case, P-value: 0.5718684724164991
Question: Q10: Childs tuition, P-value: 0.232172159220412
Question: Q11: Lawyer in-law, P-value: 0.9599597732834264
Question: Q12: Wedding donation, P-value: 0.7057697285111806
Question: Q13: Pregnancy rules, P-value: 0.142708748725531
Question: Q14: Bridesmaids hair, P-value: 0.41091916220451297
```

of invalid values for when I later access and manipulate the data.

1.4 Column Data Types

I found that all the columns were of the type object. I knew that while I wouldn't need to change the data type for every single column (for instance, I didn't change the type of "Year" to categorical since I knew I would not be investigating if there was a statistical difference in responses depending on year), I changed the data type of Q1-Q14 to be categorical. This is because I knew I would be using the questions to perform my chi squared categorical data hypothesis tests. To do so, I created a custom range of "jerkiness" categories from ['Not a jerk', 'Mildly a jerk', 'Strongly a jerk']. I felt this was a valid way to fix this issue because it makes the data easier to manipulate and does not change the data that was collected in any way.

2 My Questions

2.1 Was Professor Farina's section primed with the question "Would you describe yourself as compassionate?", thus leading for their responses to the survey to be statistically different from that of Max's section?

This question was of interest to me because I knew I wanted to merge the two data frames from Professor Fardina and Max's section together because that would mean I have more data I can conduct my hypothesis testing on. However, I knew that if the students of Professor Fardina's section were primed into responding to the questions differently from those of Professor Max's section, mixing the responses from the potentially distinct groups could lead to incorrect or misleading conclusions in the future. My null hypothesis is that there is no statistically significant difference between the responses to the questions in both Professor Fardina and Professor Max's sections. My alternative hypothesis is that there is a statistically significant difference between the responses in Professor Fardina and Professor Max's sections, indicating that Professor Fardina's section could have been primed.

I investigated this question by first creating two data frames: one with the potentially primed responses (responses from Professor Fardina's section) which were all response that were not .isna() and the other data frame (with responses from Professor Max's section that were not at risk of being primed) that were all .isna(). Then, I created a for loop to loop through every question (Q1-Q14) and create a contingency table comparing responses to each question from both the potentially primed group and the non-primed group. After the table was created, the chi squared test was conducted since this is categorical data.

I selected a significance level of $\alpha = 0.05$. Upon looking at the chi squared p values from each question, every single p value is higher than my selected alpha value. This means I fail to reject the null hypothesis that there is no statistically significant difference between the responses to the questions in both Professor Fardina and Professor Max's sections. As a result, I am able to combine

```
Question: Q1: Doctor girlfriend, P-value: 0.2719969890730754
Question: Q2: Daughter married, P-value: 0.44445455006316903
Question: Q3: Trust fund, P-value: 0.6499718441824033
Question: Q4: Kids school, P-value: 0.06340243866679925
Question: Q5: Cat, P-value: 0.2785442569330333
Question: Q6: Niece, P-value: 0.4605445885243139
Question: Q7: Flight seats, P-value: 0.10075287657649135
Question: Q8: Child support, P-value: 0.7419883522691773
Question: Q9: Child support court case, P-value: 0.5718684724164991
Question: Q10: Childs tuition, P-value: 0.232172159220412
Question: Q11: Lawyer in-law, P-value: 0.9599597732834264
Question: Q12: Wedding donation, P-value: 0.7057697285111806
Question: Q13: Pregnancy rules, P-value: 0.142708748725531
Question: Q14: Bridesmaids hair, P-value: 0.41091916220451297
```

the two sets of data into one knowing that the students of Professor Fardina's section likely were not primed when providing their responses.

2.2 Do people of different sex respond differently to questions about children?

This question was of interest to me because when answering the survey myself, I noticed that many of the questions involved children and whether or not they were being treated fairly. From personal experience, I know that there is a huge divide between those that love kids and those that are not fond of children. Just out of curiosity, I decided to see if there was a statistical difference between people of different genders and their empathy towards situations including small children. My null hypothesis that there is no statistically significant difference in the responses to the questions involving kids between surveyees of different genders. My alternative hypothesis is that there is a statistically significant difference in the responses.

I investigated this by first creating a list of questions that involved the well being of children. Then, for each question in the list, I created a contingency table using the "Gender" column and conducted a chi squared test since this is categorical data.

I selected a significance level of $\alpha = 0.05$. Upon looking at the chi squared p values from each question, every single p value is higher than my selected alpha value. This means I fail to reject the null hypothesis that there is no statistically significant difference between the responses to the questions about kids between surveyees of different genders.

```
Question: Q4: Kids school, P-value: 0.0695681482063424
Question: Q6: Niece, P-value: 0.10616481246665906
Question: Q7: Flight seats, P-value: 0.3052918698023143
Question: Q8: Child support, P-value: 0.7237132278020337
Question: Q9: Child support court case, P-value: 0.08035475558315591
```

2.3 Does different levels of religiousness impact perception of jerkiness?

This question was of interest to me because some religions emphasize compassion, forgiveness, and empathy. I was curious if such religious beliefs would have found their ways into the minds of my peers and impact the way they respond to the survey questions.

I investigated this by looping through all the questions and creating a contingency table with "Religiousness" and answers to the current question. For each question, I then perform a chi squared test since this is categorical data. My null hypothesis is that there is no statistically significant difference between survey answers from people that identify as different levels of religiousness. My alternative hypothesis is that there is a statistically significant difference between survey answers from people that identify as different levels of religiousness.

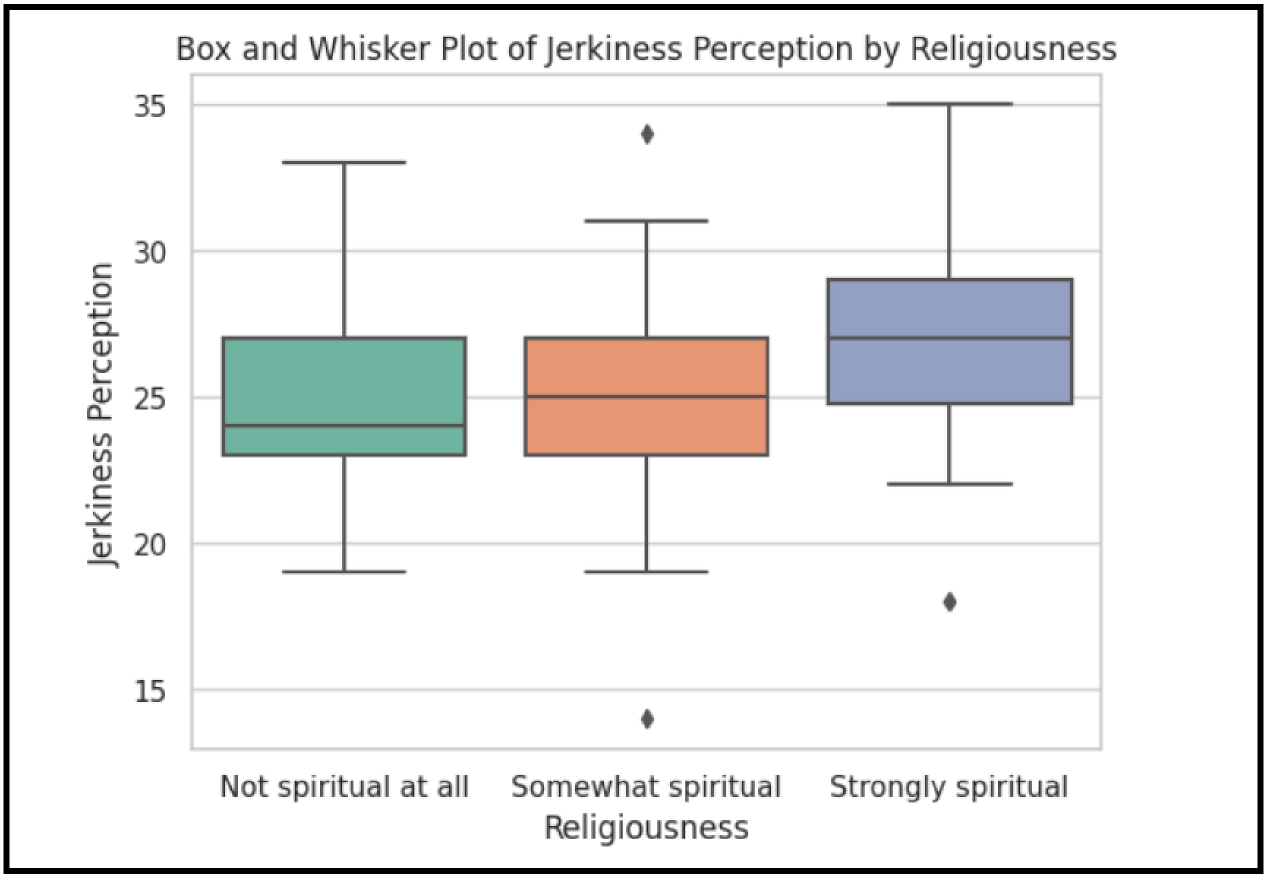
I selected a significance level of $\alpha = 0.05$. Upon looking at the p values from each question, it seems some of the questions (Q7, Q12, and Q13) have a p value that is less than the significance level,

```
Question: Q1: Doctor girlfriend, P-value: 0.2719969890730754
Question: Q2: Daughter married, P-value: 0.44445455006316903
Question: Q3: Trust fund, P-value: 0.6499718441824033
Question: Q4: Kids school, P-value: 0.06340243866679925
Question: Q5: Cat, P-value: 0.2785442569330333
Question: Q6: Niece, P-value: 0.4605445885243139
Question: Q7: Flight seats, P-value: 0.10075287657649135
Question: Q8: Child support, P-value: 0.7419883522691773
Question: Q9: Child support court case, P-value: 0.5718684724164991
Question: Q10: Childs tuition, P-value: 0.232172159220412
Question: Q11: Lawyer in-law, P-value: 0.9599597732834264
Question: Q12: Wedding donation, P-value: 0.7057697285111806
Question: Q13: Pregnancy rules, P-value: 0.142708748725531
Question: Q14: Bridesmaids hair, P-value: 0.41091916220451297
```

while all the other questions have a p value that is greater than the significance level. This means that for Q7, Q12, and Q13, we reject the null hypothesis that states there is no difference between responses from people with different levels of religiousness. Instead, there is evidence that supports the alternative hypothesis which states that there is a statistically significant difference in answers from people that have different levels of religiousness. Since this isn't the case for all the questions, it is hard to say if there is a huge impact of religiousness on the responses to the questions. To better understand this scenario, more research is needed. For starters, we can investigate Q7, Q12, and Q13 in particular. Since Q7 is related to family and children, Q12 is about same-sex marriage, and Q13 is about pregnancy and husband/wife dynamics, we can investigate more into the relationships of those topics and religion to get a better understanding of our results.

```
Question: Q1: Doctor girlfriend, P-value: 0.25776994483232024
Question: Q2: Daughter married, P-value: 0.21689978079243727
Question: Q3: Trust fund, P-value: 0.14660755789686924
Question: Q4: Kids school, P-value: 0.474339271735423
Question: Q5: Cat, P-value: 0.9707353562638525
Question: Q6: Niece, P-value: 0.024617208598755234
Question: Q7: Flight seats, P-value: 0.0005631709000570145
Question: Q8: Child support, P-value: 0.29059655002872126
Question: Q9: Child support court case, P-value: 0.8582099445230069
Question: Q10: Childs tuition, P-value: 0.7077131676559321
Question: Q11: Lawyer in-law, P-value: 0.7944171467002314
Question: Q12: Wedding donation, P-value: 4.044990153542355e-05
Question: Q13: Pregnancy rules, P-value: 0.0049331718824009
Question: Q14: Bridesmaids hair, P-value: 0.2558801291637775
```

```
Question: Q1: Doctor girlfriend, P-value: 0.2719969890730754
Question: Q2: Daughter married, P-value: 0.44445455006316903
Question: Q3: Trust fund, P-value: 0.6499718441824033
Question: Q4: Kids school, P-value: 0.06340243866679925
Question: Q5: Cat, P-value: 0.2785442569330333
Question: Q6: Niece, P-value: 0.4605445885243139
Question: Q7: Flight seats, P-value: 0.10075287657649135
Question: Q8: Child support, P-value: 0.7419883522691773
Question: Q9: Child support court case, P-value: 0.5718684724164991
Question: Q10: Childs tuition, P-value: 0.232172159220412
Question: Q11: Lawyer in-law, P-value: 0.9599597732834264
Question: Q12: Wedding donation, P-value: 0.7057697285111806
Question: Q13: Pregnancy rules, P-value: 0.142708748725531
Question: Q14: Bridesmaids hair, P-value: 0.4109191622045129 7
```




```
Question: Q1: Doctor girlfriend, P-value: 0.2719969890730754
Question: Q2: Daughter married, P-value: 0.44445455006316903
Question: Q3: Trust fund, P-value: 0.6499718441824033
Question: Q4: Kids school, P-value: 0.06340243866679925
Question: Q5: Cat, P-value: 0.2785442569330333
Question: Q6: Niece, P-value: 0.4605445885243139
Question: Q7: Flight seats, P-value: 0.10075287657649135
Question: Q8: Child support, P-value: 0.7419883522691773
Question: Q9: Child support court case, P-value: 0.5718684724164991
Question: Q10: Childs tuition, P-value: 0.232172159220412
Question: Q11: Lawyer in-law, P-value: 0.9599597732834264
Question: Q12: Wedding donation, P-value: 0.7057697285111806
Question: Q13: Pregnancy rules, P-value: 0.142708748725531
Question: Q14: Bridesmaids hair, P-value: 0.41091916220451297
```

```
import pandas as pd
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import seaborn as sea
```

```
df1 = pd.read_csv("/content/Dataset Generation (Max) (Responses) - Form Responses 1.csv")
df2 = pd.read_csv("/content/Dataset Generation (Fardina) (Responses) - Form Responses 1.csv")
df = pd.concat([df1, df2], axis=0)
```

```
column_list = df.columns.tolist()
print(column_list)
```

```
['Timestamp', 'What year are you?', 'How old are you?', 'You could describe the adults you grew up with as...', 'You could describe yours
```

```
df.rename(columns={'What year are you?': 'Year'}, inplace=True)

df.rename(columns={'How old are you?': 'Age'}, inplace=True)

df.rename(columns={'You could describe the adults you grew up with as...': 'Upbringing'}, inplace=True)

df.rename(columns={'You could describe yourself as...': 'Self-Description'}, inplace=True)

df.rename(columns={'How would you rate your religiousness / spirituality?': 'Religiousness'}, inplace=True)

df.rename(columns={'What bests represents your gender?': 'Gender'}, inplace=True)

df.rename(columns={'My girlfriend is a doctor. Lately she\'s been complaining about pain in her right knee and constantly taking TONS of ibupro

df.rename(columns={'My daughter is getting married soon. I only learned about her a few years ago. We\'ve been building a relationship the last

df.rename(columns={'I\'m a trust fund kid; I get a healthy \'allowance\' from my parents, but I mostly sock it away since I don\'t really feel

df.rename(columns={'My wife and I have separate finances, but I pay for almost everything. My son starts school next year, and I\'m planning or

df.rename(columns={'I saw a poster for a lost cat advertising a 500 dollar reward. I saw the cat, tracked it down, and called the owner. When I

df.rename(columns={'My sister\'s nine year old daughter is poorly behaved. One day, my sister dropped the daughter off on my doorstep without m

df.rename(columns={'My parents want us to come out for their anniversary, and bought my sister and I tickets on the same flight. My sister has

df.rename(columns={'I\'m a single mom with four kids, one of whom has a different father from the other three. I get a lot of child support for

df.rename(columns={'I have a child with a mother who never wanted anything to do with them. I make enough money to cover my expenses, but I cor

df.rename(columns={'One of my children wants to go to an expensive school to become a dentist. I told them I\'d be fine paying for it. The othe

df.rename(columns={'I was in a conflict with my mother-in-law\'s boyfriend, in which I made a snide comment about he\'s never paid child suppor

df.rename(columns={'\nSome of my relatives refuse to come to my wedding, since they don\'t approve of our \'lifestyle\'. I would like to donate

df.rename(columns={'My wife has decided that since she can’t drink because she is pregnant that I can’t either. I planned to take my annual lea

df.rename(columns={'My sister is going to be a bridesmaid at my wedding. Her hair was dyed, but she recently decided to grow it out, so parts c

df.rename(columns={'Would you describe yourself as compassionate?': 'Compassionate'}, inplace=True)

df.rename(columns={'My wife has decided that since she can’t drink because she is pregnant that I can’t either. I planned to take my annual lea
```

```
df = df[['Timestamp', 'Year', 'Age', 'Gender', 'Upbringing', 'Self-Description', 'Religiousness', 'Compassionate', 'Q1: Doctor girlfriend', '

df['Q13: Pregnancy rules'].fillna(df['Q14: Pregnancy rules'], inplace=True)
df.drop(columns=['Q14: Pregnancy rules'], inplace=True)
```

```
print(df.dtypes)
```

```
Timestamp                   object
Year                        object
Age                         object
Gender                      object
Upbringing                  object
```

```
Self-Description              object
Religiousness                 object
Compassionate                 object
Q1: Doctor girlfriend         object
Q2: Daughter married          object
Q3: Trust fund                object
Q4: Kids school               object
Q5: Cat                       object
Q6: Niece                     object
Q7: Flight seats              object
Q8: Child support             object
Q9: Child support court case  object
Q10: Childs tuition           object
Q11: Lawyer in-law            object
Q12: Wedding donation         object
Q13: Pregnancy rules          object
Q14: Bridesmaids hair         object
dtype: object
```

```
df = df.dropna(how='all')
```

```
df = df[df['Age'].notna()]
df = df[df['Age'] != 0]
print(df['Age'].value_counts())
```

```
20.0    64
20      48
19.0    31
21.0    25
19      21
21      15
18.0     5
23.0     3
22       3
25.0     2
22.0     2
17       1
23       1
50+      1
18       1
26       1
24       1
29.0     1
17.0     1
40       1
28.0     1
24.0     1
Name: Age, dtype: int64
```

```
df['Age'] = df['Age'].replace('50+', np.nan)

df['Age'] = df['Age'].astype(float).astype(pd.Int64Dtype())

print(df['Age'].value_counts())
print(df.dtypes)
```

```
20    112
19     52
21     40
18      6
22      5
23      4
24      2
17      2
25      2
40      1
26      1
29      1
28      1
Name: Age, dtype: Int64
Timestamp                     object
Year                          object
Age                            Int64
Gender                        object
Upbringing                    object
Self-Description              object
Religiousness                 object
Compassionate                 object
Q1: Doctor girlfriend         object
Q2: Daughter married          object
Q3: Trust fund                object
Q4: Kids school               object
```

```
Q5: Cat                        object
Q6: Niece                      object
Q7: Flight seats               object
Q8: Child support              object
Q9: Child support court case   object
Q10: Childs tuition            object
Q11: Lawyer in-law             object
Q12: Wedding donation          object
Q13: Pregnancy rules           object
Q14: Bridesmaids hair          object
dtype: object
```

```
jerk_categories = ['Not a jerk', 'Mildly a jerk', 'Strongly a jerk']

columns_to_convert = ['Q1: Doctor girlfriend', 'Q2: Daughter married', 'Q3: Trust fund', 'Q4: Kids school',
                      'Q5: Cat', 'Q6: Niece', 'Q7: Flight seats', 'Q8: Child support', 'Q9: Child support court case',
                      'Q10: Childs tuition', 'Q11: Lawyer in-law', 'Q12: Wedding donation', 'Q13: Pregnancy rules', 'Q14: Bridesmaids hair']

for column in columns_to_convert:
    df[column] = pd.Categorical(df[column], categories=jerk_categories, ordered=True)

print(df.dtypes)
```

```
Timestamp                        object
Year                             object
Age                               Int64
Gender                           object
Upbringing                       object
Self-Description                 object
Religiousness                    object
Compassionate                    object
Q1: Doctor girlfriend          category
Q2: Daughter married           category
Q3: Trust fund                 category
Q4: Kids school                category
Q5: Cat                        category
Q6: Niece                      category
Q7: Flight seats               category
Q8: Child support              category
Q9: Child support court case   category
Q10: Childs tuition            category
Q11: Lawyer in-law             category
Q12: Wedding donation          category
Q13: Pregnancy rules           category
Q14: Bridesmaids hair          category
dtype: object
```

```
print(df['Compassionate'].value_counts())
print(df['Compassionate'])
```

```
Yes    120
No      16
Name: Compassionate, dtype: int64
1      NaN
3      NaN
4      NaN
5      NaN
6      NaN
      ...
131     No
132    Yes
133    Yes
134    Yes
135    Yes
Name: Compassionate, Length: 230, dtype: object
```

```
maybe_primed = df[~df['Compassionate'].isna()]
maybe_not_primed = df[df['Compassionate'].isna()]

questions = ['Q1: Doctor girlfriend', 'Q2: Daughter married', 'Q3: Trust fund', 'Q4: Kids school', 'Q5: Cat',
             'Q6: Niece', 'Q7: Flight seats', 'Q8: Child support', 'Q9: Child support court case', 'Q10: Childs tuition',
             'Q11: Lawyer in-law', 'Q12: Wedding donation', 'Q13: Pregnancy rules', 'Q14: Bridesmaids hair']

for question in questions:
    table = pd.crosstab(maybe_primed[question], maybe_not_primed[question])
    chi2, p, _, _ = sp.stats.chi2_contingency(table)
    print(f"Question: {question}, P-value: {p}")
```

```
Question: Q1: Doctor girlfriend, P-value: 0.2719969890730754
Question: Q2: Daughter married, P-value: 0.44445455006316903
```

```
Question: Q3: Trust fund, P-value: 0.6499718441824033
Question: Q4: Kids school, P-value: 0.06340243866679925
Question: Q5: Cat, P-value: 0.2785442569330333
Question: Q6: Niece, P-value: 0.4605445885243139
Question: Q7: Flight seats, P-value: 0.10075287657649135
Question: Q8: Child support, P-value: 0.7419883522691773
Question: Q9: Child support court case, P-value: 0.5718684724164991
Question: Q10: Childs tuition, P-value: 0.232172159220412
Question: Q11: Lawyer in-law, P-value: 0.9599597732834264
Question: Q12: Wedding donation, P-value: 0.7057697285111806
Question: Q13: Pregnancy rules, P-value: 0.142708748725531
Question: Q14: Bridesmaids hair, P-value: 0.41091916220451297
```

```
print(df.columns)
```

```
Index(['Timestamp', 'Year', 'Age', 'Gender', 'Upbringing', 'Self-Description',
       'Religiousness', 'Compassionate', 'Q1: Doctor girlfriend',
       'Q2: Daughter married', 'Q3: Trust fund', 'Q4: Kids school', 'Q5: Cat',
       'Q6: Niece', 'Q7: Flight seats', 'Q8: Child support',
       'Q9: Child support court case', 'Q10: Childs tuition',
       'Q11: Lawyer in-law', 'Q12: Wedding donation', 'Q13: Pregnancy rules',
       'Q14: Bridesmaids hair'],
      dtype='object')
```

```
questions = ['Q1: Doctor girlfriend', 'Q2: Daughter married', 'Q3: Trust fund', 'Q4: Kids school', 'Q5: Cat',
             'Q6: Niece', 'Q7: Flight seats', 'Q8: Child support', 'Q9: Child support court case', 'Q10: Childs tuition',
             'Q11: Lawyer in-law', 'Q12: Wedding donation', 'Q13: Pregnancy rules', 'Q14: Bridesmaids hair']

for question in questions:
    contingency_table = pd.crosstab(df['Religiousness'], df[question])
    chi2, p, _, _ = sp.stats.chi2_contingency(contingency_table)
    print(f"Question: {question}, P-value: {p}")
```

```
Question: Q1: Doctor girlfriend, P-value: 0.25776994483232024
Question: Q2: Daughter married, P-value: 0.21689978079243727
Question: Q3: Trust fund, P-value: 0.14660755789686924
Question: Q4: Kids school, P-value: 0.474339271735423
Question: Q5: Cat, P-value: 0.9707353562638525
Question: Q6: Niece, P-value: 0.024617208598755234
Question: Q7: Flight seats, P-value: 0.0005631709000570145
Question: Q8: Child support, P-value: 0.29059655002872126
Question: Q9: Child support court case, P-value: 0.8582099445230069
Question: Q10: Childs tuition, P-value: 0.7077131676559321
Question: Q11: Lawyer in-law, P-value: 0.7944171467002314
Question: Q12: Wedding donation, P-value: 4.044990153542355e-05
Question: Q13: Pregnancy rules, P-value: 0.0049331718824009
Question: Q14: Bridesmaids hair, P-value: 0.2558801291637775
```

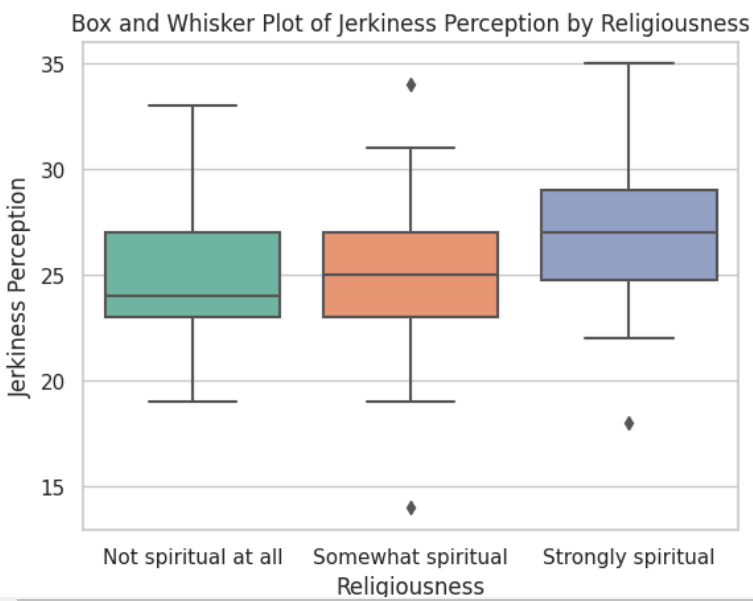
```
print(df['Q5: Cat'])
```

```
1        Mildly a jerk
3           Not a jerk
4           Not a jerk
5      Strongly a jerk
6           Not a jerk
             ...
131         Not a jerk
132    Strongly a jerk
133         Not a jerk
134      Mildly a jerk
135         Not a jerk
Name: Q5: Cat, Length: 230, dtype: category
Categories (3, object): ['Not a jerk' < 'Mildly a jerk' < 'Strongly a jerk']
```

```
questions = ['Q4: Kids school', 'Q6: Niece', 'Q7: Flight seats', 'Q8: Child support', 'Q9: Child support court case']

for question in questions:
  table = pd.crosstab(df['Gender'], df[question])
  chi2, p, _, _ = sp.stats.chi2_contingency(table)
  print(f"Question: {question}, P-value: {p}")
```

```
Question: Q4: Kids school, P-value: 0.0695681482063424
Question: Q6: Niece, P-value: 0.10616481246665906
Question: Q7: Flight seats, P-value: 0.3052918698023143
Question: Q8: Child support, P-value: 0.7237132278020337
Question: Q9: Child support court case, P-value: 0.08035475558315591
```

```
df = df.dropna()

def calculate_jerkiness(row):
    jerkiness = 0
    jerkiness_mapping = {
        "Not a jerk": 1,
        "Mildly a jerk": 2,
        "Strongly a jerk": 3
    }

    questions = ['Q1: Doctor girlfriend', 'Q2: Daughter married', 'Q3: Trust fund', 'Q4: Kids school', 'Q5: Cat',
                 'Q6: Niece', 'Q7: Flight seats', 'Q8: Child support', 'Q9: Child support court case', 'Q10: Childs tuition',
                 'Q11: Lawyer in-law', 'Q12: Wedding donation', 'Q13: Pregnancy rules', 'Q14: Bridesmaids hair']

    for question in questions:
        response = row[question]
        jerkiness += jerkiness_mapping.get(response, 0)

    return jerkiness

sea.set(style="whitegrid")

sea.boxplot(x="Religiousness", y="Jerkiness", data=df, palette="Set2")

plt.title("Box and Whisker Plot of Jerkiness Perception by Religiousness")
plt.xlabel("Religiousness")
plt.ylabel("Jerkiness Perception")

plt.show()
```



Start coding or generate with AI.