DIGHUM101- Contraceptive Data

July 4, 2020

1 Digital Humanities 101 Individual Project - Contraceptive Data

1.1 Introduction

Contraceptive use varies greatly across the world. Between 1994 and today, contraceptive use in the world overall grew by 8.3%. Access to effective contraceptives is greatly tied to women's autonomy and well-being. Furthermore, contraceptives are important because they can reduce the risk of sexually transmitted diseases and help women's health in many other ways.

1.2 Research Question

I am interested in finding general patterns with contraceptive use, education, standard of living, etc. Specifically, I am looking for the answer to the question:

What factors most influence a women's decision of contraceptive use, and to what extent?

From studying this, I can hopefully learn more about contraceptive use and expand to learning more about women's rights.

1.3 Materials

I retrieved the data from the UCI Machine Learning Repository, but the data is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. There samples are from married women who were either not pregnant or did not know if they were pregnant at the time of the interview. The different attributes were as follows:

1) Wife's age
2) Wife's education
3) Husband's education
4) Number of children ever born
5) Wife's religion
6) Wife's work status (currently)
7) Husband's occupation
8) Standard of living index

9) Media Exposure

10) Contraceptive use

*Note: Instead of the datasets categorization of long term, short term, and no contraceptive use, I decided to look at a binary variable: whether they used contraceptives or not.

1.4 Data and Results

1.4.1 Part 1: Set Up

[4]:
```
import seaborn as sns
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.linear_model import LogisticRegression
import zipfile
import plotly.express as px
from pathlib import Path
```

[5]:
```
#Loading in data
contraceptive = pd.read_csv("contraceptive.csv")
features = ['wife_age', 'wife_education','husband_education', 'num_child',␣
↪'wife_religion', 'wife_work','husband_occupation', 'standard_living',␣
↪'media_exposure']
```

[6]:
```
new_contraceptive = contraceptive['contraceptive'].replace(1, 0).replace(2, 1).
↪replace(3, 1)
contraceptive['used_contraceptive'] = new_contraceptive
contraceptive = contraceptive.drop(['contraceptive'], axis = 1)
contraceptive.head(10)
```

[6]:

	wife_age	wife_education	husband_education	num_child	wife_religion	\
0	24	2	3	3	1	
1	45	1	3	10	1	
2	43	2	3	7	1	
3	42	3	2	9	1	
4	36	3	3	8	1	
5	19	4	4	0	1	
6	38	2	3	6	1	
7	21	3	3	1	1	
8	27	2	3	3	1	
9	45	1	1	8	1	

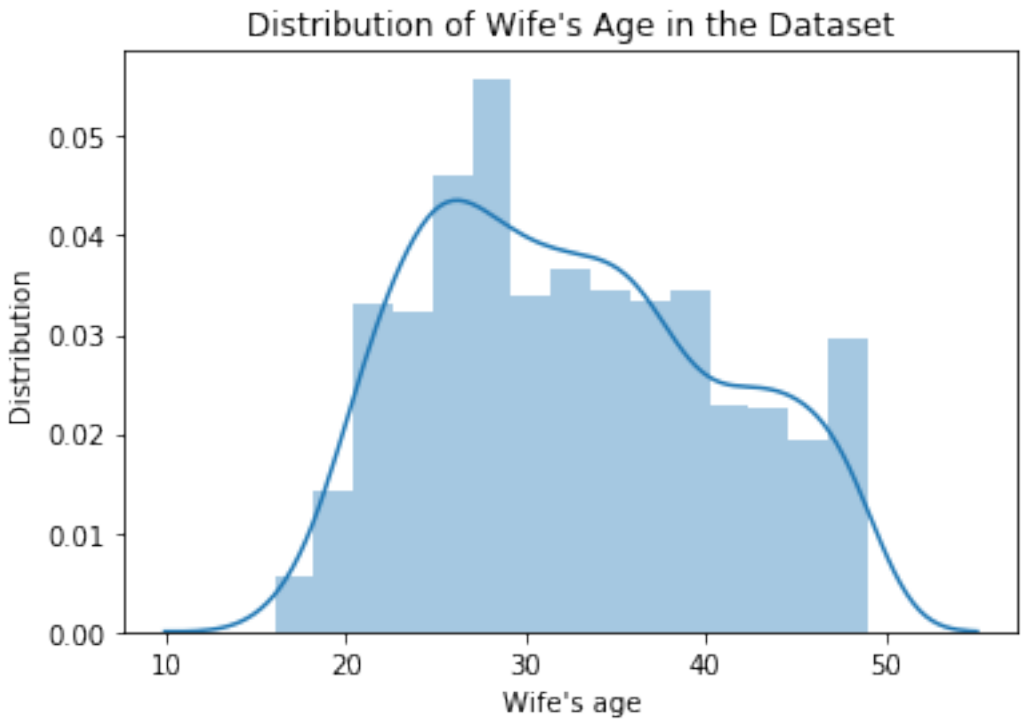
```
   wife_work  husband_occupation  standard_living  media_exposure  \
0          1                   2                3               0
1          1                   3                4               0
2          1                   3                4               0
3          1                   3                3               0
4          1                   3                2               0
5          1                   3                3               0
6          1                   3                2               0
7          0                   3                2               0
8          1                   3                4               0
9          1                   2                2               1

   used_contraceptive
0                   0
1                   0
2                   0
3                   0
4                   0
5                   0
6                   0
7                   0
8                   0
9                   0
```

1.4.2 Part 2: Exploratory Data Analysis

To start off, I wanted to learn a little bit more about the data. I started off looking at the distribution of the wife's age, to see if this data was mostly on younger or older women, or if there was an even distribution of all ages. I also looked at the distribution of number of children had by each woman.

[7]:
```
sns.distplot(contraceptive['wife_age'])
plt.xlabel("Wife's age")
plt.ylabel("Distribution")
plt.title("Distribution of Wife's Age in the Dataset")
```

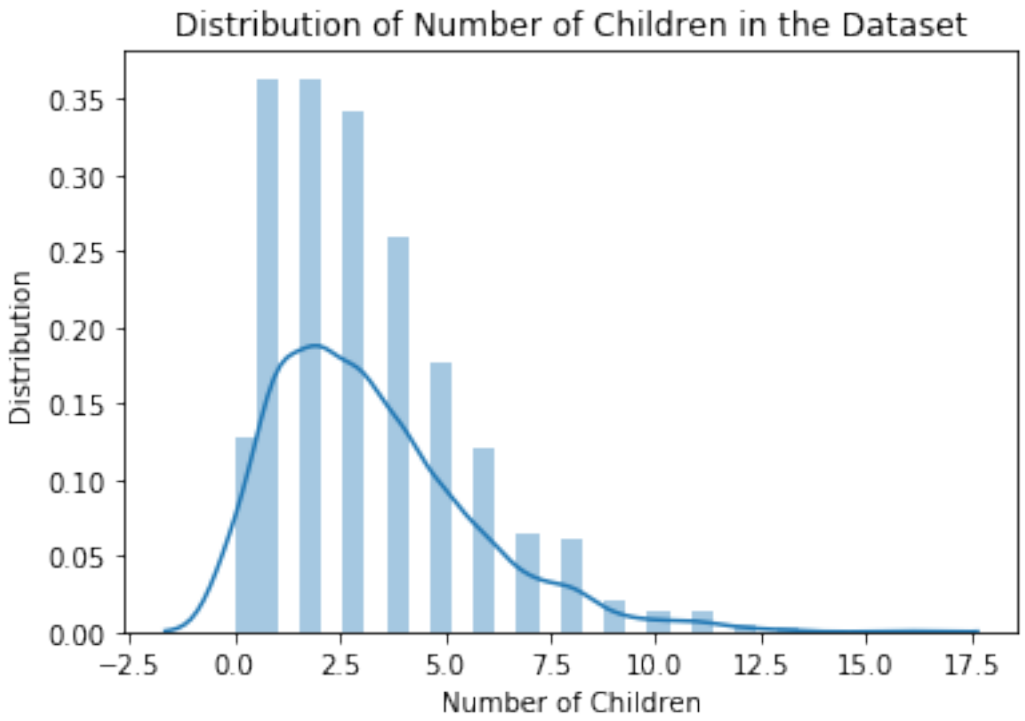
[7]:
```
Text(0.5, 1.0, "Distribution of Wife's Age in the Dataset")
```



[8]:
```
wife_age_median = contraceptive['wife_age'].quantile([.5])
```

[9]:
```
sns.distplot(contraceptive['num_child'])
plt.xlabel("Number of Children")
plt.ylabel("Distribution")
plt.title("Distribution of Number of Children in the Dataset")
```

[9]:
```
Text(0.5, 1.0, 'Distribution of Number of Children in the Dataset')
```



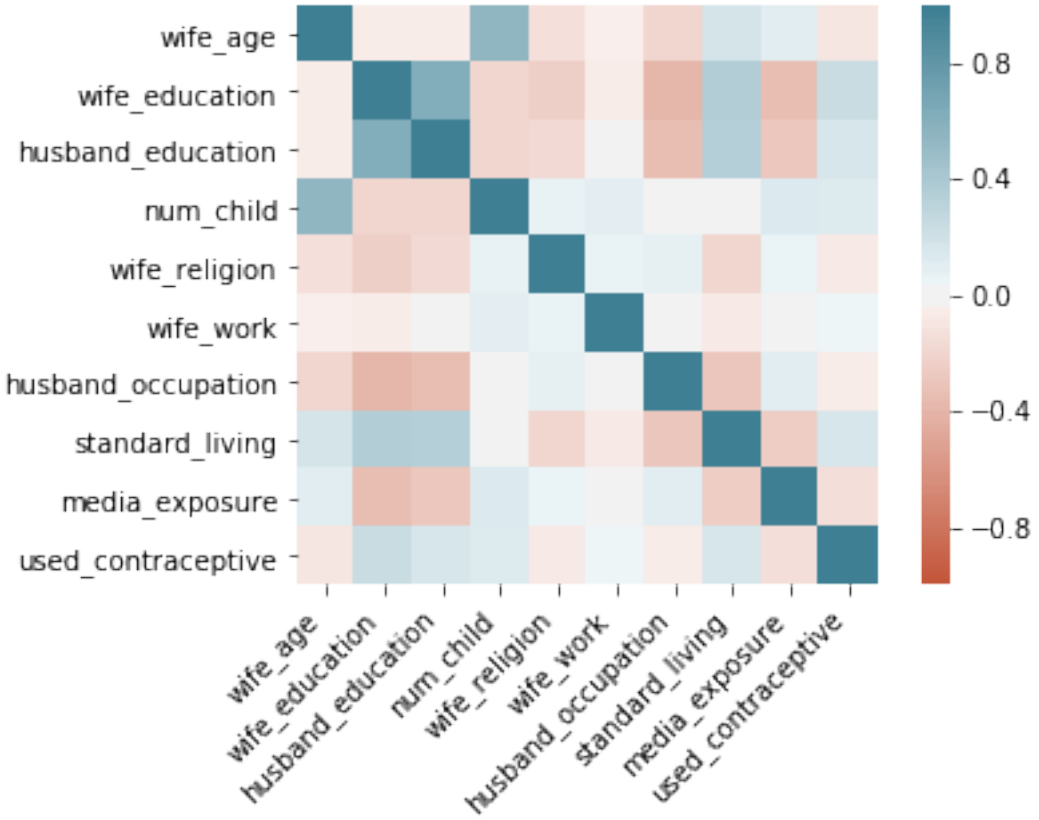
[10]:
```
num_child_median = contraceptive['num_child'].quantile([.5])
```

Next, I wanted to dive deeper into what causes women to choose certain types of birth control. I made a correlation map to see the relationship between many variables.

[11]:
```
corr = contraceptive.corr()
ax = sns.heatmap(
    corr,
    vmin=-1, vmax=1, center=0,
    cmap=sns.diverging_palette(20, 220, n=200),
    square=True
)

ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right'
)
```

[11]:
```
[Text(0.5, 0, 'wife_age'),
 Text(1.5, 0, 'wife_education'),
 Text(2.5, 0, 'husband_education'),
 Text(3.5, 0, 'num_child'),
 Text(4.5, 0, 'wife_religion'),
```

```
Text(5.5, 0, 'wife_work'),
Text(6.5, 0, 'husband_occupation'),
Text(7.5, 0, 'standard_living'),
Text(8.5, 0, 'media_exposure'),
Text(9.5, 0, 'used_contraceptive')]
```



Next, I wanted to create a random forrest to predict the relationship between contraceptive use and other variables. I created this random forrest as to confirm and further extend the results that I got through the heatmap.

[12]:
```
from sklearn.model_selection import train_test_split

X = contraceptive[features]

y = contraceptive['used_contraceptive']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.10)
```

[13]:
```
from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=100)
```

```
clf.fit(X_train,y_train)

y_pred=clf.predict(X_test)

feature_importances = clf.feature_importances_

data = {'Features': features, 'Feature Importances': feature_importances}
features_df = pd.DataFrame(data).sort_values('Feature Importances', ascending =␣
↪False)
features_df
```

[13]:

	Features	Feature Importances
0	wife_age	0.354317
3	num_child	0.258162
1	wife_education	0.085685
7	standard_living	0.081894
6	husband_occupation	0.074903
2	husband_education	0.061043
5	wife_work	0.040036
4	wife_religion	0.026131
8	media_exposure	0.017830

1.4.3 Part 3: Models and Deeper Analysis

From there, I selected some columns that I thought would be useful in my linear model. I created the model using a pipeline, and tested the model on my test data to see how accurate I was.

[14]:
```
def make_categories(data, col1, col2, val1, val2):
    categories = np.array([])
    for i in np.arange(data.shape[0]):
        if (data[col1][i] >= val1) and (data[col2][i] >= val2):
            categories = np.append(categories, 4)
        elif (data[col1][i] >= val1) and (data[col2][i] < val2):
            categories = np.append(categories, 3)
        elif (data[col1][i] < val1) and (data[col2][i] >= val2):
            categories = np.append(categories, 2)
        else:
            categories = np.append(categories, 1)
    return categories
```

[19]:
```
model = LogisticRegression(solver = 'lbfgs', max_iter = 1000)

def select_columns(data, *columns):
    """Select only columns passed as arguments."""
    return data.loc[:, columns]
```

```
def onehot(data, column):
    vec_enc = DictVectorizer()
    vec_enc.fit(data[[column]].to_dict(orient='records'))
    column_data = vec_enc.transform(data[[column]].to_dict(orient='records')).
 ↪toarray()
    column_cats = vec_enc.get_feature_names()
    column = pd.DataFrame(column_data, columns=column_cats)
    data = pd.concat([data, column], axis=1)
    return data

def process_data_fm(data):
    #Encode mother's information
    data['mother_info'] = make_categories(data, 'wife_age', 'num_child', 32, 3)

    # Transform Data, Select Features
    data = select_columns(data,
                          'wife_age',
                          'wife_education',
                          'husband_education',
                          'num_child',
                          'husband_occupation',
                          'wife_work',
                          'media_exposure',
                          'standard_living',
                          'mother_info',
                          'used_contraceptive'
                         )

    # Return predictors and response variables separately
    X = data.drop(['used_contraceptive'], axis = 1)
    y = data.loc[:, 'used_contraceptive']
    return X, y
```

[20]:
```
X_train, y_train = process_data_fm(contraceptive)
X_test, y_test = process_data_fm(contraceptive)

model.fit(X_train, y_train)
```

[20]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

[21]:
```
training_accuracy = model.score(X_train, y_train)
print("Training Accuracy: ", training_accuracy)
```

```
Training Accuracy:  0.6836388323150034
```

I thought the accuracy was fine, but I wanted to ensure my model was a good fit. Thus, I performed cross-validation on my model to make sure the model was adapted to fit a variety of datasets. The results are below.

[22]:
```
scores = cross_val_score(model, process_data_fm(contraceptive)[0],␣
 ↪process_data_fm(contraceptive)[1], cv = 20)
print(scores, np.mean(scores))
```

```
[0.69333333 0.73333333 0.73333333 0.73333333 0.71621622 0.7027027
 0.68918919 0.64864865 0.75675676 0.5890411  0.61643836 0.57534247
 0.60273973 0.71232877 0.7260274  0.73972603 0.65753425 0.60273973
 0.64383562 0.69863014] 0.6785615204245341
```

1.5 Conclusion and Extentions

In this dataset, I found that factors such as a woman's age and their number of children played a large role in their use of contraceptives. Other factors, such as religion, did not play as large of an impact. In general, it does not seem like there are outside forces that play an overbearing role in a woman's choice to use contraceptives. For example, from the model, it does not seem like a woman with a low standard of living is necessarily unlikely to use contraceptives.

If I were to expand on this project, I would like to gather more attributes. For example, I think it could be useful to have data on a woman's income, whether they live in a rural or urban location, etc. This may give further insight on contraceptive use.

Furthermore, I would want to use data with a more international scope. I think it is important to note that this data was done in Indonesia, which is a newly industrialized country. While it is good that there does not seem to be glaring inequalities, this result can not be extrapolated to the rest of the world. In some less developed countries, the inequalities might be greater than Indonesia.

Overall, it seems like there are definitely patterns in determining a woman's contraceptive use (or lack thereof). These patterns seem to be more honed in on a woman's personal life. Thus, this implies that women are probably largely able to make a choice on whether they use contraceptives or not, as opposed to being forced to by factors such as religion or poverty. This is definitely an improvement and likely coincides with women gaining more rights in general; hopefully this trend will proceed in the future.