[redacted]

My exploratory data analysis (EDA) project is based on the Wages File, which regroups data on employees of a firm. I will examine the characteristics of the data, analyze the data to identify some patterns, while creating some graphs, in order to know more about the employees of this firm. I will formulate possible hypotheses and also test them.

At the end, I implemented a multiple linear regression model to predict the hourly wage of a hypothetical new employee and examine its validity. The quality of this model is unfortunately low.

In [1]:
```
import pandas as pd
#I import the pandas library while storing it in the "pd" alias, because I may need that later.
import numpy as np
#I import the numpy library while storing it in the "np" alias, because I may need that later.
%matplotlib inline
#This inline is run in order to display matplotlib plots in the notebook.
```

First, i will load the dataset from the file.

In [2]:
```
df = pd.read_csv('wage1.csv', index_col=0)
# I loaded the csv file and read it. I considered the first column as the index.
```

Once the dataset is loaded, I can examine it.

We would like to know, how many rows and columns are in the database.

In [3]:
```
df.shape
```

Out[3]:
```
(525, 7)
```

The database contains **seven information on 525 employees.** Each row represents an employee and a column a piece of information.

Let's have more information about the columns and their data types.

In [4]:
```
#What are the columns and their data types?
df.dtypes
```

Out[4]:
```
married                  bool
hourly_wage           float64
years_in_education      int64
years_in_employment     int64
num_dependents          int64
gender                 object
race                   object
dtype: object
```

The column "hourly_wage", which stands for the hourly wage, contains float numbers, which are numbers with a decimal point.

The columns "years_in_education", "years_in_employment", "num_dependents" (number of dependents) are integers, that is to say a whole number without a decimal point.

The column "married" is a boolean, which takes either the value "True"(married) or "False" (not married).

The columns "gender" and "race" contain objects.

Let's have a look at the first 10 rows of the dataset.

In [5]:
```
df.head(10)
#I only print the 10 first rows
```

Out[5]:

	married	hourly_wage	years_in_education	years_in_employment	num_dependents	gender	race
0	True	3.24	12	2	3	female	white
1	False	3.00	11	0	2	male	white
2	True	6.00	8	28	0	male	white
3	True	5.30	12	2	1	male	white
4	True	8.75	16	8	0	male	white
5	False	11.25	18	7	0	male	white
6	False	5.00	12	3	0	female	white
7	False	3.60	12	4	2	female	white
8	True	18.18	17	21	0	male	white
9	False	6.25	16	2	0	female	white

1. Descriptive Statistics about the Dataset

In order to display descriptive statistics about the dataset, I will use the method "describe" and show only 2 decimals.

In [6]:
```
df.describe().round(2)
```

Out[6]:

	hourly_wage	years_in_education	years_in_employment	num_dependents
count	525.00	525.00	525.00	525.00
mean	5.90	12.57	5.11	1.04
std	3.69	2.77	7.23	1.26
min	0.53	0.00	0.00	0.00
25%	3.33	12.00	0.00	0.00
50%	4.65	12.00	2.00	1.00
75%	6.88	14.00	7.00	2.00
max	24.98	18.00	44.00	6.00

We can observe that for example:

```
- the mean value of the "hourly_wage" column is 5.9, which means in average the employees of this firm earn
£5.9 per hour;
- the median suggests that at least 50% of the employees spent 12 years in education;
- 75% of the employees have less than 7 years of work experience;
- The employee with the biggest number of dependents has 6 of them (max variable);
- Some employees did not go to university and have never worked before this job at this firm (min variable).
```

2. Data Proofing and Cleaning

Before going further with the analysis of the data, we need first to check whether there any missing values.

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
married                 0
hourly_wage             0
years_in_education      0
years_in_employment     0
num_dependents          0
gender                  0
race                    0
dtype: int64
```

No data is null.

In [8]:
```
df.isna().sum()
```

Out[8]:
```
married                 0
hourly_wage             0
years_in_education      0
years_in_employment     0
num_dependents          0
gender                  0
race                    0
dtype: int64
```

And **no data is missing.**

So, we can start with the analysis of the data.

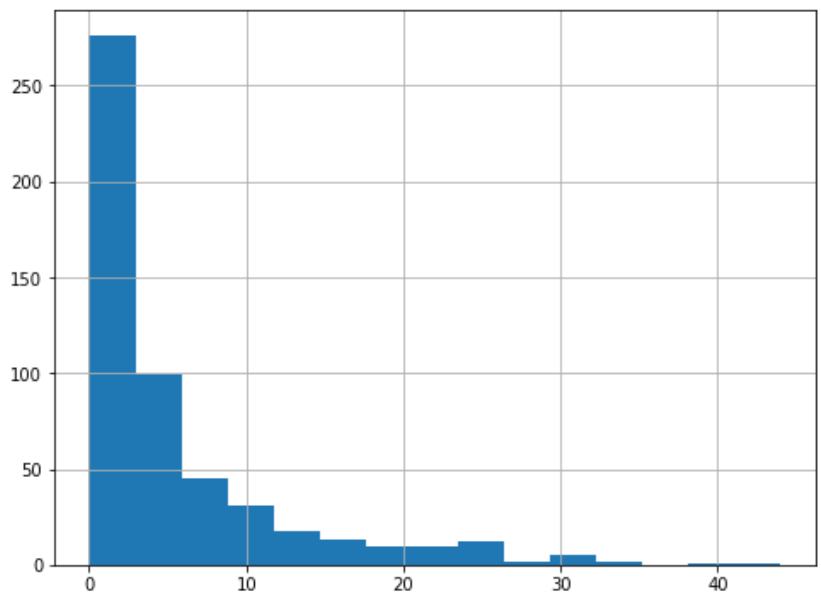
3. Distribution of some Numerical Variables

A. Distribution of the Years of Employment

Let's have a look at the distribution of the years of employment.

In [46]:
```
df['years_in_employment'].hist(bins=15, figsize=(8,6))
```

Out[46]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1de36d2c080>
```



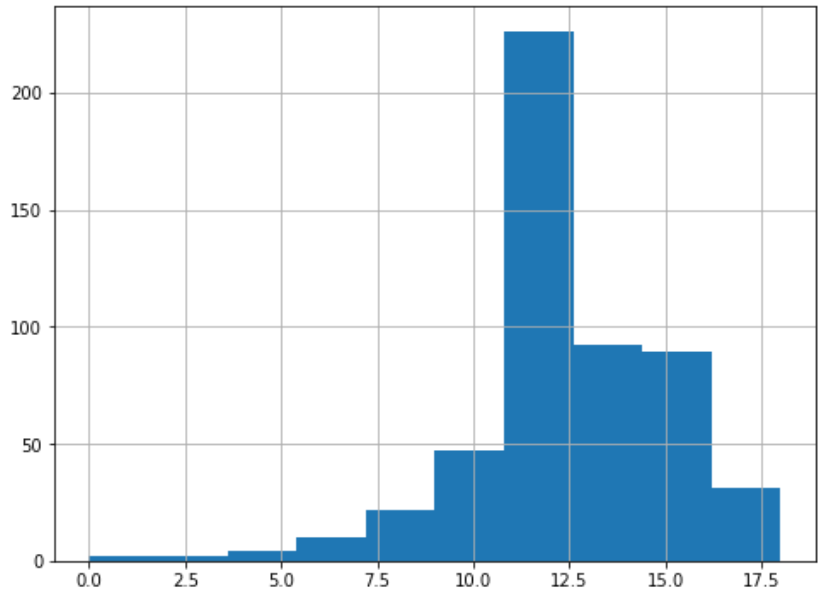
We can see that **most people of this firm (more than 250 people) have less than 5 years of employment.**

Let's see the distribution of the years in education.

B. Distribution of the Years in Education

```
In [47]: df['years_in_education'].hist(bins=10,figsize=(8,6))
```

```
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x1de36d94fd0>
```



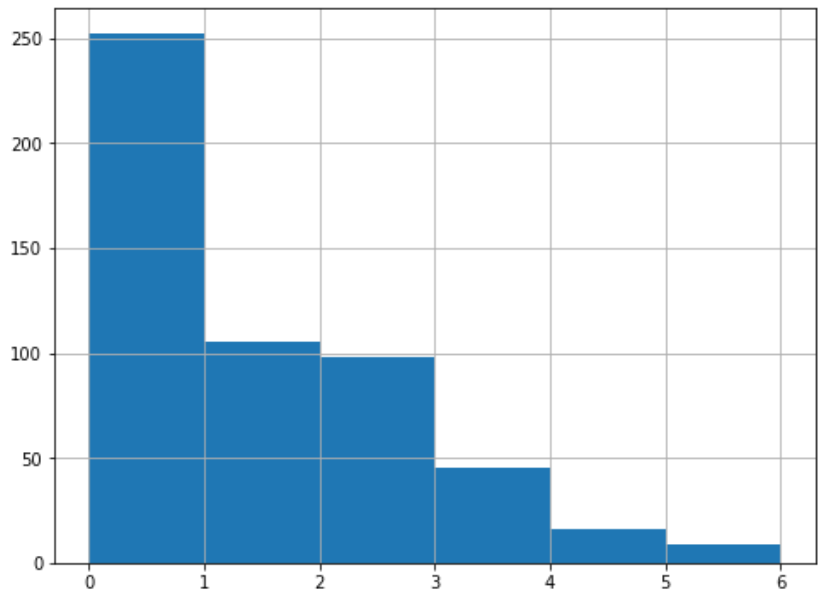
More than 200 people (almost half of the firm) have spent more than 11 years in education. It may be because in this country, education is mandatory for a certain period or this firm has specific hiring criteria for education.

Now, let's have a look at the distribution of the number of dependents.

C. Distribution of the Number of Dependents

```
In [48]: df['num_dependents'].hist(bins=6, figsize=(8,6))
```

```
Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x1de36e795f8>
```



More than 250 employees (almost half of the employees of this firm) have no dependent, a bit more than 100 employees have one dependent (we think it may be a child). Approximately 100 of the employees have 2 dependents and more than 50 people have more than 3 dependents.

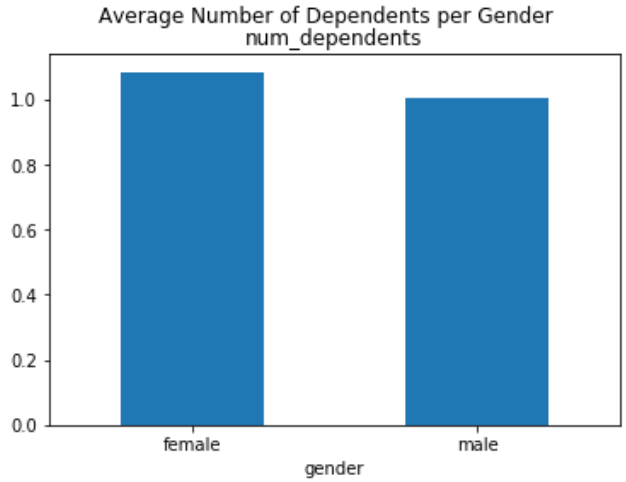
4. Data Visualization of some Numerical Variables of the Dataset

First, let's build a graph representing the average number of dependents per gender.

A. The Average Number of Dependents per Gender

In [12]:
```
df.groupby("gender")['num_dependents'].mean().plot(kind="bar", title="Average Number of Dependents per
 Gender", subplots=True,
                                                   rot=0)
```

Out[12]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000001DE317DAC18>],
      dtype=object)
```



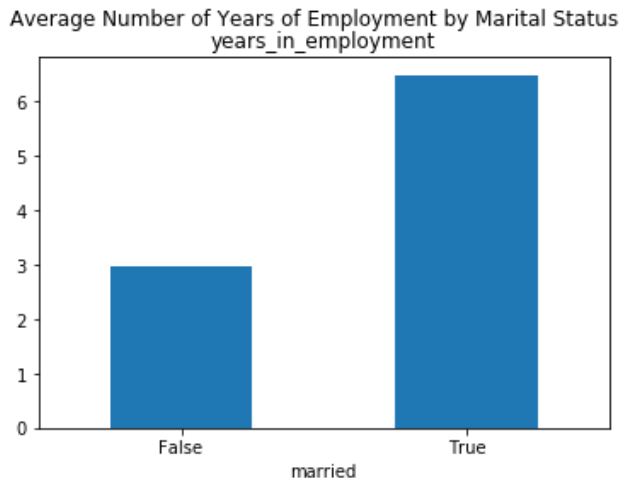
We can see on the graph that **the average number of dependents per gender is quite similar for women and men in this firm.**

B. The Average Number of Years of Employment by Marital Status

Second, let's visualize the average number of years of work experience by marital status.

In [13]:
```
df.groupby(["married"])['years_in_employment'].mean().plot(kind="bar",
                                                           title="Average Number of Years of Employment
by Marital Status",
                                                           subplots=True, rot=0)
```

Out[13]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000001DE3182EA58>],
      dtype=object)
```



On the graph, we can observe that **married employees have in average more work experience than the non-married ones.**

5. Unique Values of the Variable "Gender"

Now, we will display unique values for the categorical variable "gender".

In other words, how many women and men work in this firm?

In [14]:
```
# value_counts counts unique values in a column
df.gender.value_counts()
```

Out[14]:
```
male      274
female    251
Name: gender, dtype: int64
```

There are **251 women and 274 men** in this firm.

Let's have a look at the gender proportion in percentages.

In [15]:
```
df.gender.value_counts(normalize=True).round(3)
```

Out[15]:
```
male      0.522
female    0.478
Name: gender, dtype: float64
```

52.2% of the employees are men and 47.8% are women.

6. Contingency Table and Chi-square Test for the Marital Status and the Gender

Marital status and the gender are both categorical variables (non-numeric). We will analyse in this part whether they are related.

First, let's have a look at both variables with a contingency table. We will be able to find out, how many women and men in this firm are married or not. I think that these two variables may be related.

A. Contingency Table

In [16]:
```
#We create a contingency table using the crosstab method
cont_table = pd.crosstab(df.married, df.gender)
# print the contingency table
cont_table
```

Out[16]:

gender	female	male
married		
False	119	86
True	132	188

The contingency table represents the observed frequencies of co-occurrences of different values of the marital status and gender. For example, 119 women in this firm are not married and 188 male employees are married.

In [17]:
```
cont_table = pd.crosstab(df.married, df.gender, normalize='index').round(2)
# print the contingency table in percentage with 2 decimals
cont_table
```

Out[17]:

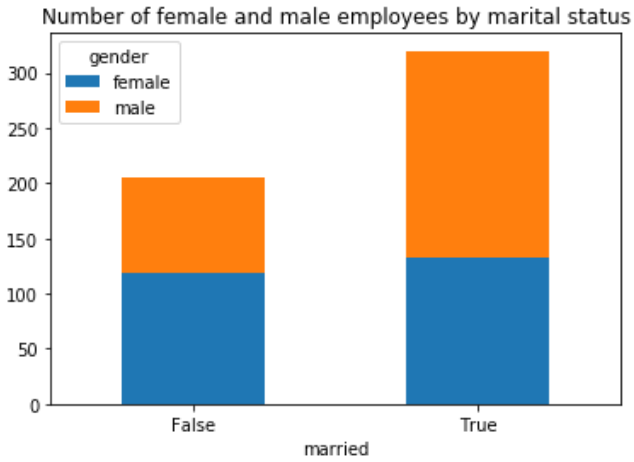
gender	female	male
married		
False	0.58	0.42
True	0.41	0.59

It seems that there are more married male employees than married female employees.

Let's visualize the contingency table as stacked bars.

In [18]:
```
cont_table = pd.crosstab(df.married, df.gender)
cont_table.plot(kind="bar", stacked=True, rot=0, title="Number of female and male employees by marital
 status")
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1de3199f518>
```



The plot suggests that, for example, males constitute a greater share of married employees.

Let's conduct a Chi-square Test, a statistical test, to determine whether the marital status and the gender are independent.

B. Chi-square Test

In [19]:
```
from scipy import stats
# import the statistical functions
```

The null hypothesis of the chi-square test is always that the two variables are independent. The alternative hypothesis is that they are dependent.

We will use directly the chi2-contingency function.

In [20]:
```
chi2, p_val, dof, expected = stats.chi2_contingency(cont_table)
print(f"p-value: {p_val}")
```

```
p-value: 0.0002428643213722774
```

The p-value is less than the usual significance level of 0.05. Therefore, we reject the null hypothesis that there is no dependence between the gender and the marital status. In other words, the **marital status and the gender are related.** Men are more likely to be married in this firm than women.

7. A Subset of the Data based on two or more Criteria with Descriptive Statistics

Let's select a subset of observations. For example, **the number of non-white women** in this firm.

In [21]:
```
print(f"There are", len(df[(df.gender == "female") & (df.race=='nonwhite')] ), "non-white female employ
ees in this firm.")
```

```
There are 25 non-white female employees in this firm.
```

What is their average hourly wage?

In [22]:
```
df[(df.gender == "female") & (df.race=='nonwhite')]['hourly_wage'].mean()
```

Out[22]:
```
4.2376000000000005
```

On average, non-white female employees earn £4.24 per hour in this firm.

In [23]:
```
df[(df.gender == "female") & (df.race=='nonwhite')]['years_in_employment'].mean()
```

Out[23]:
```
3.56
```

On average, they have 4 years of employment.

What are the years of employment of non-white women?

In [24]:
```
df[(df.gender == "female") & (df.race=='nonwhite')]['years_in_employment'].value_counts()
```

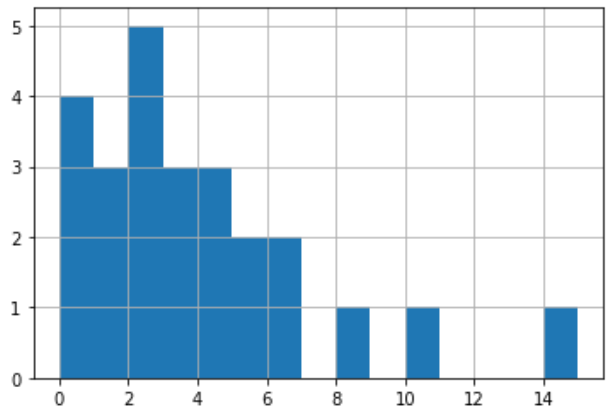
Out[24]:
```
2     5
0     4
4     3
3     3
1     3
6     2
5     2
15    1
10    1
8     1
Name: years_in_employment, dtype: int64
```

For example, 5 non-white women have 2 years of employment.

We can create a histogram to visualize the distribution of the non-white women employees by the number of years of employment.

In [25]:
```
df[(df.gender == "female") & (df.race=='nonwhite')]['years_in_employment'].hist(bins=15)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1de3347d470>
```



8. Independent two-sample t-test for married and unmarried employees

Let's see if the average hourly wage of married employees is greater than the average hourly wage for unmarried employees. If it is, then it would mean that being married increases the chances of earning more.

In [26]:
```
# creation of a serie with the hourly wage of non-married employees
Not_Married=df[df['married'] == False]['hourly_wage']

# What is the average hourly wage for non-married people?
Not_Married.mean()
```

Out[26]:
```
4.852390243902436
```

Non-married people earn on average £4.85 per hour.

In [27]:
```
# creation of a serie with the hourly wage of married employees
Married=df[df['married'] == True]['hourly_wage']

# What is the average hourly wage for married people?
Married.mean()
```

Out[27]:
```
6.5734687500000035
```

Married people earn on average £6.57 per hour.

It seems that **married employees earn more than the non-married ones.**

Let's see if the difference between the two means is statistically significant. We will use the **independent sample t-test**, implemented in the "scipy" package.

The null hypothesis is that there is no difference between the means of the two subsets.

The alternative hypothesis is that there is a significant difference between them.

In [28]:
```
# independent t-test from the scipy package
from scipy import stats
from scipy.stats import ttest_ind

# ttest_ind expects two NumPy arrays as input,
# so, we need to input the `values` of the two series
t_val, p_val = stats.ttest_ind(Not_Married.values, Married.values)

# ttest_ind returns the t-value and p-value
print(f"t-value: {t_val}, p-value: {p_val}")
```

```
t-value: -5.342324014459073, p-value: 1.3716089568681514e-07
```

The p-value is smaller than the significance level of 0.05. We can reject the null hypothesis that there is no difference between the means of the two samples. **So, being married has a significant effect on the hourly wage.**

9. Pivot tables

Pandas makes it possible to group observations by values of a certain column. We will use the groupby method to create a pivot table.

A. Pivot Table for the Gender

In [29]:
```
# Creation of the pivot table to see several information by gender, with 3 decimals.
df.groupby('gender').aggregate({"hourly_wage":['mean', 'min', 'max'],"years_in_education":['mean', 'mi
n', 'max'],
                                "years_in_employment":['mean', 'min', 'max'],"num_dependents":['mean',
'min', 'max']}).round(3)
```

Out[29]:

	hourly_wage			years_in_education			years_in_employment			num_dependents		
	mean	min	max	mean	min	max	mean	min	max	mean	min	max
gender												
female	4.594	0.53	21.63	12.323	0	18	3.629	0	34	1.084	0	5
male	7.099	1.50	24.98	12.788	2	18	6.474	0	44	1.004	0	6

We can observe in this pivot table, for example, that:

- female employees have on average an hourly wage of £4.6, versus £7.1 for male employees in this firm;
- the female employee with the lowest hourly wage is three times lower (£0.5) versus the male collegues'lowest one (£1.5);
- the maximum years spent in education for women and men is the same : 18 years;
- on average, male employees spent a bit more time in education than the female employees : 12.8 years vs 12.3 years;
- on average, male employees have more years of work experience (6.5 years) than the female employees (3.6 years). The main reason would be pregnancy for women;
- On average, male employees have less dependents than their female colleages : 1.0 versus 1.1. But, the male employee
with the highest number of dependents has more dependents than the female colleague with the highest number of
dependents : 6 versus 5.

B. Pivot Table for the Marital Status

In [30]:
```
# Creation of the pivot table to see several information by marital status, with 3 decimals.
df.groupby('married').aggregate({"hourly_wage":['mean', 'min', 'max'],"years_in_education":['mean', 'min', 'max'],
                                 "years_in_employment":['mean', 'min', 'max'],"num_dependents":['mean', 'min', 'max']}).round(3)
```

Out[30]:

	hourly_wage			years_in_education			years_in_employment			num_dependents		
	mean	min	max	mean	min	max	mean	min	max	mean	min	max
married												
False	4.852	0.53	21.63	12.332	0	18	2.961	0	30	0.795	0	5
True	6.573	1.43	24.98	12.716	0	18	6.494	0	44	1.200	0	6

We can observe in this pivot table, for example, that:

- unmarried employees have on average a lower hourly wage (£4.9) than married employees (£6.6) in this firm;
- The married employee's lowest hourly wage (£1.4) is higher than the unmarried one (£0.5);
- On average, married employees spent a bit more time at school and university than their unmarried collegues: 12.7
years versus 12.3 years;
- Married and unmarried employees have both at least one colleague who did not go to school and one colleague who spent the same maximum amount of time in education (18 years);
- On average, married employees have spent more years employed (6.5 years versus 3 years);
- The unmarried employee with the maximum number of work experience is 30 years. It is lower than the maximum number of work experience for one married employee (44 years).
- On average, married people have more dependents than unmarried employees : 1.2 versus 0.8;
- The maximum number of dependents for a married employee is 6, versus 5 for an unmarried employee.

C. Pivot Table for the Ethnicity of the Employees

In [31]:
```
# Creation of the pivot table to see several information by ethnicity, with 3 decimals.
df.groupby('race').aggregate({"hourly_wage":['mean', 'min', 'max'],"years_in_education":['mean', 'min', 'max'],
                              "years_in_employment":['mean', 'min', 'max'],"num_dependents":['mean', 'min', 'max']}).round(3)
```

Out[31]:

	hourly_wage			years_in_education			years_in_employment			num_dependents		
	mean	min	max	mean	min	max	mean	min	max	mean	min	max
race												
nonwhite	5.476	1.96	15.00	11.870	3	18	5.352	0	30	1.333	0	6
white	5.950	0.53	24.98	12.645	0	18	5.087	0	44	1.008	0	6

We can observe in this pivot table, for example, that:

- non-white employees have on average a lower hourly wage (£5.5) than white employees (£6.0) in this firm;
- The white employee's lowest hourly wage (£0.5) is lower than the non-white one (£2.0);
- On average, white employees spent more time at school and university than their non-white collegues: 12.6 years versus 11.9 years;
- The lowest time spent in education for non-white employees is 3 years vs zero for white employees;
- The maximum time spent in education for both types of employees is the same;
- On average, non-white employees have spent more years employed (5.4 years versus 5.1 years);
- The white employee with the maximum number of work experience is 44 years. It is higher than the maximum number of
work experience for one non-white employee (30 years).
- On average, non-white employees have more dependents than white employees : 1.3 versus 1.0;
- The maximum number of dependents for a non-white employee is the same as his/her white colleague (6).

D. Pivot Table for the Gender and the Marital Status

In [32]:
```
# Creation of the pivot table to see several information by gender and marital status, with 3 decimals.
df.groupby(['gender','married']).mean().round(3)
```

Out[32]:

		hourly_wage	years_in_education	years_in_employment	num_dependents
gender	married				
female	False	4.624	12.176	2.748	1.000
	True	4.566	12.455	4.424	1.159
male	False	5.168	12.547	3.256	0.512
	True	7.983	12.899	7.947	1.229

We can observe in this pivot table, for example, that on average:

- married female employees have on average a lower hourly-wage than their unmarried male colleagues (£4.6 versus £5.2);
- married male employees spent more time in education than their unmarried female colleagues (12.9 years versus 12.2
years);
- married female employees have more work experience (4.4 years) than their unmarried male colleagues (3.3 years);
- unmarried female employees have more dependents than their unmarried colleagues: 1 versus 0.5.

10. Correlation

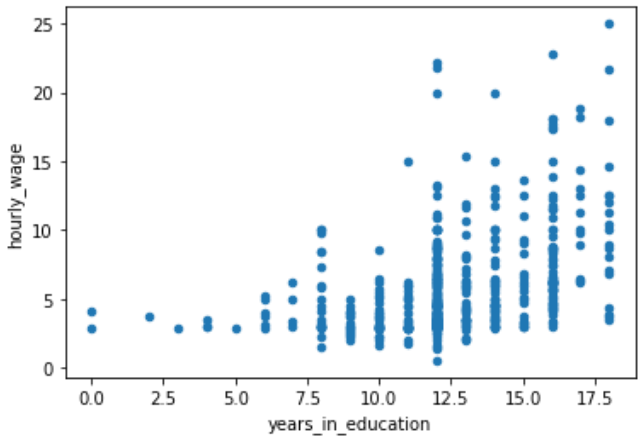
We would like to examine the relationship between the number of years in education and the hourly wage.

First, let's examine the relationship between the two variables with a scatterplot.

A. Correlation between the Number of Education Years and the Hourly Wage

```
In [33]: df.plot(kind='scatter', x='years_in_education', y='hourly_wage')
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1de33509860>
```



It does not look like there is much of a relationship.

Let's calculate **Pearson's correlation coefficient**, which measures linear relationship between two continuous variables. The coefficient ranges between -1 and +1: r = +1: a perfect positive linear relationship r = 0: there is no linear relationship r = -1: a perfect negative linear relationship

If the Pearson's Correlation coefficients are bigger than 0.7 or less than -0.7, there are a relationship between the two independents variables.

To do that, we call the corr method of one Series and pass the other series as the argument to the method.

```
In [34]: df['years_in_education'].corr(df['hourly_wage'], method='pearson')
```

```
Out[34]: 0.4054328824295751
```

The correlation value is lower than 0.5. So, it appears there is **a weak correlation between the number of years in education and the hourly wage.**

B. Variable Transformation

To use the categorical variables "gender" and "race" later on in the correlation and the regression model, we must first convert them to numerical variables by using the replace method.

Race variable:

- 0 will represent white employees;
- 1 non-white employees

Gender variable:

- 0 will represent male employees;
- 1 female employees

We do not have to replace the values for the Married variable because "True" equals to 1 and "False" represents 0.

Race Variable - Transformation

In [35]:
```
df["race"] = df["race"].replace("white", 0)
df["race"] = df["race"].replace("nonwhite", 1)
```

Gender Variable - Transformation

In [36]:
```
df["gender"] = df["gender"].replace("male", 0)
df["gender"] = df["gender"].replace("female", 1)
```

In [37]:
```
# let's have a look at the 10 first rows of our dataset to see whether the replacements worked out
df.head(10)
```

Out[37]:

	married	hourly_wage	years_in_education	years_in_employment	num_dependents	gender	race
0	True	3.24	12	2	3	1	0
1	False	3.00	11	0	2	0	0
2	True	6.00	8	28	0	0	0
3	True	5.30	12	2	1	0	0
4	True	8.75	16	8	0	0	0
5	False	11.25	18	7	0	0	0
6	False	5.00	12	3	0	1	0
7	False	3.60	12	4	2	1	0
8	True	18.18	17	21	0	0	0
9	False	6.25	16	2	0	1	0

C. Correlation Matrix

Instead of calculating for each variable the Pearson's Correlation coefficient, I can calculate it for every variable and put it directly on a matrix. This is what I will calculate just below.

As I did before, I will call the corr method and use it on the whole dataframe.

In [38]:
```
# I call the corr method and use it on the whole dataframe. I want to calculate the Pearson's coefficie
nt. Hence,
# the "method = pearson"
corr = df.corr(method='pearson')

# I applied to the matrix a color scheme and I also set up the number of decimals to 4.
corr.style.background_gradient(cmap='coolwarm').set_precision(4)
```

Out[38]:

	married	hourly_wage	years_in_education	years_in_employment	num_dependents	gender	race
married	1	0.2275	0.06766	0.2387	0.1566	-0.1641	-0.06316
hourly_wage	0.2275	1	0.4054	0.3462	-0.05272	-0.3391	-0.03903
years_in_education	0.06766	0.4054	1	-0.05698	-0.2147	-0.08402	-0.08505
years_in_employment	0.2387	0.3462	-0.05698	1	-0.02604	-0.1968	0.01114
num_dependents	0.1566	-0.05272	-0.2147	-0.02604	1	0.03169	0.07824
gender	-0.1641	-0.3391	-0.08402	-0.1968	0.03169	1	-0.01026
race	-0.06316	-0.03903	-0.08505	0.01114	0.07824	-0.01026	1

We can note that we find in the correlation matrix the same Pearson's Correlation Coefficient between the years in education and the hourly wage as before.

Since there are no Pearson's Correlation coefficient bigger than 0.7 and less than -0.7, we can say that **there are no relationships between the independents variables (every variable but the hourly wage).**

11. Multiple Linear Regression Model

We would like to build a regression model to predict the hourly wage from all other variables.

We will use the formula below:

Hourly_wage = α + β1∗married + β2∗years_in_education + β3∗years_of_employment + β4∗num_dependents + β5∗gender + β5∗race + e

A. Building and Interpreting the Multiple Linear Regression Model

In [39]:
```
# To build linear regression models
import statsmodels.api as sm

# We'll use Bokeh to plot the figures
from bokeh.io import output_notebook
output_notebook()
from bokeh.plotting import figure
from bokeh.io import show
```

(https://bokeh.pydata.org)
BokehJS 1.2.0 successfully loaded.

1. Model 1

In [40]:
```
#Implement a linear regression model and interpret its output.
model = sm.OLS.from_formula(
    'hourly_wage ~ married + years_in_education + years_in_employment + num_dependents + gender + race'
, data=df).fit()


#To interpret the model, we access the model summary
model.summary()
```

Out[40]:

OLS Regression Results

Dep. Variable:	hourly_wage	**R-squared:**	0.366
Model:	OLS	**Adj. R-squared:**	0.358
Method:	Least Squares	**F-statistic:**	49.80
Date:	Thu, 09 Jan 2020	**Prob (F-statistic):**	2.71e-48
Time:	23:06:30	**Log-Likelihood:**	-1311.0
No. Observations:	525	**AIC:**	2636.
Df Residuals:	518	**BIC:**	2666.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.3427	0.699	-1.920	0.055	-2.716	0.031
married[T.True]	0.6266	0.282	2.225	0.026	0.073	1.180
years_in_education	0.5399	0.048	11.136	0.000	0.445	0.635
years_in_employment	0.1558	0.019	8.282	0.000	0.119	0.193
num_dependents	0.1088	0.107	1.014	0.311	-0.102	0.319
gender	-1.7202	0.267	-6.444	0.000	-2.245	-1.196
race	-0.0979	0.429	-0.228	0.820	-0.940	0.745

Omnibus:	187.184	**Durbin-Watson:**	1.792
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	708.583
Skew:	1.612	**Prob(JB):**	1.36e-154
Kurtosis:	7.690	**Cond. No.**	78.0

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Let's refresh our memory with the replacement of the categorical variables:

Race variable:

- 0 will represent white employees;
- 1 non-white employees

Gender variable:

- 0 will represent male employees;
- 1 female employees

(1) Coefficients on the variables. Our model is thus described by the line:

$$HourlyWage = -1.3427 + 0.6266 * Married(iftrue) - 1.7202 * Gender - 0.0979 * Race + 0.5399 * YearsInEducation + 0 * YearsOfEmployment + 0.1088 * NumDependents + e$$

Considering the signs on the coefficients, we can state that the hourly wage is positively affected by being married, the years in education (the greater the number of years in education is, the more likely the hourly wage is high) and employment (the greater the number of years of employment is, the more likely the hourly wage is high) and the number of dependents (the greater the number of dependents is, the more likely the hourly wage is high). It is negatively affected by the gender (if you are a woman, you are more likely to receive a lower hourly wage) and the race (if you are non-white, you are more likely to receive a lower hourly wage).

(2) Significance of the variables. Which predictor variables are the most important? The p-values on all the coefficients (except the intercept, the number of dependents and the race) indicate that the variables are significant, i.e., the factors do have a significant effect on the hourly wage. Indeed, their p value is lower than the usual significance level of 0.05. So, the **most important predictor variables are marriage, gender, years in education and years of employment.** It means that we can omit the other variables (race and number of dependents).

(3) Quality of the model. The coefficient of determination $R2$ and the adjusted R2 describe the amount of variation in the dependent variable (hourly wage) that is explained by the model. It ranges between 0 and 1, with one being the perfect fit. The adjusted R2 removes the dependence of R2 to the number of predictors (the greater the number of predictors is, the higher will be R2). R2 and the adjusted R2 values are around 0.36. So, **the quality of the model is rather low.**

Let's create another multiple linear regression model without the race and number of dependents.

2. Model 2

In [41]:
```
model2 = sm.OLS.from_formula(
    'hourly_wage ~ married + years_in_education + years_in_employment + gender', data=df).fit()


#To interpret the model, we access the model summary
model2.summary()
```

Out[41]:

OLS Regression Results

Dep. Variable:	hourly_wage	**R-squared:**	0.365
Model:	OLS	**Adj. R-squared:**	0.360
Method:	Least Squares	**F-statistic:**	74.58
Date:	Thu, 09 Jan 2020	**Prob (F-statistic):**	6.07e-50
Time:	23:06:31	**Log-Likelihood:**	-1311.5
No. Observations:	525	**AIC:**	2633.
Df Residuals:	520	**BIC:**	2654.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.1394	0.656	-1.737	0.083	-2.428	0.150
married[T.True]	0.6857	0.275	2.492	0.013	0.145	1.226
years_in_education	0.5294	0.047	11.255	0.000	0.437	0.622
years_in_employment	0.1542	0.019	8.232	0.000	0.117	0.191
gender	-1.7109	0.267	-6.420	0.000	-2.234	-1.187

Omnibus:	188.407	**Durbin-Watson:**	1.797
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	723.304
Skew:	1.618	**Prob(JB):**	8.64e-158
Kurtosis:	7.753	**Cond. No.**	73.1

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

(1) Coefficients on the variables. Our model is thus described by the line:

$$HourlyWage = -1.1394 + 0.6857 * Married(iftrue) - 1.7109 * Gender + 0.5294 * YearsInEducation + 0.1542 * YearsOfEmployment + e$$

Considering the signs on the coefficients, we can state that the hourly wage is positively affected by being married, the years in education (the greater the number of years in education is, the more likely the hourly wage is high) and employment (the greater the number of years in employment is, the more likely the hourly wage is high). It is negatively affected by the gender (if you are a woman, you are more likely to receive a lower hourly wage).

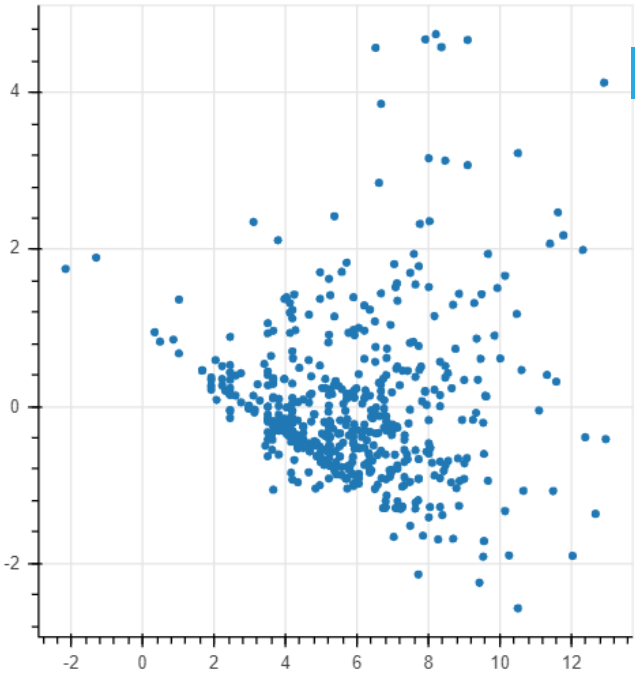
(2) Significance of the variables. The p-values on all the coefficients, except the intercept, indicate that the variables are significant. Indeed, their p value is lower than the usual significance level of 0.05. So, **the most important predictor variables are marriage, gender, years in education and employment.**

(3) Quality of the model. The $R2$ and the adjusted R2 values are around 0.36. The adjusted R2 value even increased a bit versus the previous model. But, **the quality of the model is still low.**

B. Checking the assumptions of normality and zero mean of residuals

We can plot the standardized residuals and their histogram to confirm that the assumptions of normality of the distribution of residuals and of the zero mean of residuals are valid with this model.

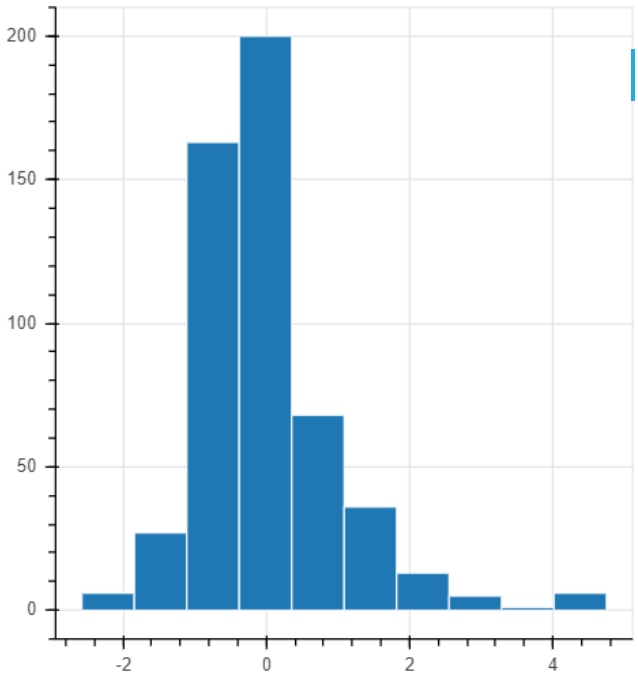
In [42]:
```
# Check the assumptions of normality and zero mean of residuals
# with plotting the standardized residuals and their histogram

fig = figure(height=400, width=400)

# the x axis is the fitted values
# the y axis is the standardized residuals
st_resids = model2.get_influence().resid_studentized_internal
fig.circle(model2.fittedvalues, st_resids)

show(fig)
```



(http

It seems that there is a pattern on the scatterplot, which suggests the residuals are not equally and normally distributed around 0.

In [43]:
```
# We create a histogram with 10 bins
hist, edges = np.histogram(st_resids, bins=10)
fig = figure(height=400, width=400)
fig.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:], line_color="white")
show(fig)
```



(http

The histogram also suggests that the residuals are not equally and normally distributed around 0.

The results of the Jarque-Bera test on the residuals:

The p-value is really small, close to 0, so really below the level of significance (0.05). So, we can reject the null hypothesis of normal distribution. To conclude, **the errors are not normally and equally distributed.**

C. Conclusion

To sum up, **I would not use this model to predict hourly wages because its quality is poor and the assumptions of normality of the distribution of residuals and of the zero mean of residuals are not met with this model.** If we use this model, we would not have accurate results.

In [ ]: