Income Dynamics: Analyzing the Influence of Education, Gender, and Race

present

`df_adult =`

	age	workclass	fnlwgt	education	education_num	marital_s
1	39	" State-gov"	77516	" Bachelors"	13	" Never-marrie
2	50	" Self-emp-not-inc"	83311	" Bachelors"	13	" Married-civ·
3	38	" Private"	215646	" HS-grad"	9	" Divorced"
4	53	" Private"	234721	" 11th"	7	" Married-civ·
5	28	" Private"	338409	" Bachelors"	13	" Married-civ·
6	37	" Private"	284582	" Masters"	14	" Married-civ·
7	49	" Private"	160187	" 9th"	5	" Married-spou
8	52	" Self-emp-not-inc"	209642	" HS-grad"	9	" Married-civ·
9	31	" Private"	45781	" Masters"	14	" Never-marrie
10	42	" Private"	159449	" Bachelors"	13	" Married-civ·
more						
32561	52	" Self-emp-inc"	287927	" HS-grad"	9	" Married-civ·

1.1

Describing the Dataframe

	variable	mean	min	median	max	
1	:age	38.5816	17	37.0	90	(
2	:workclass	nothing	" ?"	nothing	" Without-pay"	(
3	:fnlwgt	1.89778e5	12285	178356.0	1484705	(
4	:education	nothing	" 10th"	nothing	" Some-college"	(
5	:education_num	10.0807	1	10.0	16	(
6	:marital_status	nothing	" Divorced"	nothing	" Widowed"	(
7	:occupation	nothing	" ?"	nothing	" Transport-moving"	(
8	:relationship	nothing	" Husband"	nothing	" Wife"	(
9	:race	nothing	" Amer-Indian-Eskimo"	nothing	" White"	(
10	:sex	nothing	" Female"	nothing	" Male"	(
11	:capital_gain	1077.65	0	0.0	99999	(
12	:capital_loss	87.3038	0	0.0	4356	(
13	:hours_per_week	40.4375	1	40.0	99	(
14	:native_country	nothing	" ?"	nothing	" Yugoslavia"	(
15	:income	nothing	" <=50K"	nothing	" >50K"	(

Size, Names and types of features

```
["age", "workclass", "fnlwgt", "education", "education_num", "marital_status", "occupati
```

```
(32561, 15)
```

```
Vector{String} (alias for Array{String, 1})
```

— I removed rows that include "?" in any column.

```
df_adult_filtered =
```

	age	workclass	fnlwgt	education	education_num	marital_
1	39	" State-gov"	77516	" Bachelors"	13	" Never-marri
2	50	" Self-emp-not-inc"	83311	" Bachelors"	13	" Married-civ
3	38	" Private"	215646	" HS-grad"	9	" Divorced"
4	53	" Private"	234721	" 11th"	7	" Married-civ
5	28	" Private"	338409	" Bachelors"	13	" Married-civ
6	37	" Private"	284582	" Masters"	14	" Married-civ
7	49	" Private"	160187	" 9th"	5	" Married-spo
8	52	" Self-emp-not-inc"	209642	" HS-grad"	9	" Married-civ
9	31	" Private"	45781	" Masters"	14	" Never-marri
10	42	" Private"	159449	" Bachelors"	13	" Married-civ
more						
30162	52	" Self-emp-inc"	287927	" HS-grad"	9	" Married-civ

— Corelation between features and income can be calculated on the dataset.

— Single Linear Regression can be applied to figure out the effect of each feature on income.

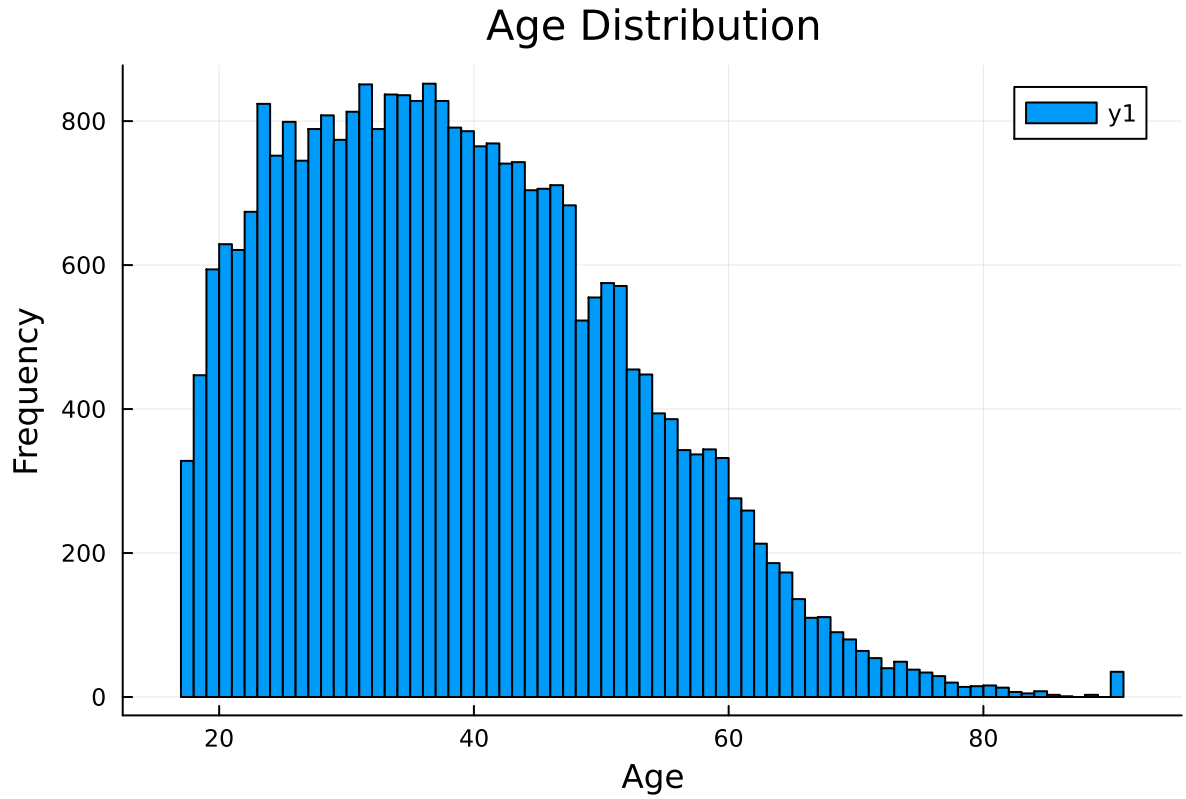
— Multiple Linear Regression can be applied to see whether the effect of one feature changes the efect of another feature.

— Lastly, labeling income greater than 50K as 1 and below 50K as 0, classification method can be studied on dataset.

2.1 Plotting Histogram based on "Age" data

```
Histogram{Int64, 1, Tuple{StepRangeLen{Float64, Base.TwicePrecision{Float64}, Base.TwiceF
edges:
  17.0:1.0:91.0
weights: [328, 447, 594, 629, 621, 674, 824, 752, 799, 745  …  13, 7, 5, 8, 3, 1, 0, 3, 0
closed: left
isdensity: false
```

```
age_hist =
```



2.2 Sorting age data on dataframe

```
df_adult_sorted =
```

	age	workclass	fnlwgt	education	education_num	marital_stat
1	17	" Private"	65368	" 11th"	7	" Never-married
2	17	" Private"	245918	" 11th"	7	" Never-married
3	17	" Private"	191260	" 9th"	5	" Never-married
4	17	" Private"	270942	" 5th-6th"	3	" Never-married
5	17	" Private"	89821	" 11th"	7	" Never-married
6	17	" Private"	175024	" 11th"	7	" Never-married
7	17	" Private"	211870	" 9th"	5	" Never-married
8	17	" Private"	242718	" 11th"	7	" Never-married
9	17	" Private"	169658	" 10th"	6	" Never-married
10	17	" Self-emp-not-inc"	368700	" 11th"	7	" Never-married
more						
30162	90	" Local-gov"	214594	" 7th-8th"	4	" Married-civ-s

Approximate the histogram data by polynomials

```
 [17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0
```

```
edges_int =
 [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,    more ,8
```

```
 (74)
```

```
 [328, 447, 594, 629, 621, 674, 824, 752, 799, 745, 789, 808, 774, 813, 851, 789, 837, 836,
```

```
 (74)
```

degreeofp = 

```
fill_matrix_A (generic function with 1 method)
```

```
A_poly = 74×6 Matrix{Float64}:
 1.0  17.0   289.0    4913.0   83521.0        1.41986e6
 1.0  18.0   324.0    5832.0  104976.0        1.88957e6
 1.0  19.0   361.0    6859.0  130321.0        2.4761e6
 1.0  20.0   400.0    8000.0  160000.0        3.2e6
 1.0  21.0   441.0    9261.0  194481.0        4.0841e6
 1.0  22.0   484.0   10648.0  234256.0        5.15363e6
 1.0  23.0   529.0   12167.0  279841.0        6.43634e6
 ⋮                                            ⋮
 1.0  85.0  7225.0  614125.0       5.22006e7  4.43705e9
 1.0  86.0  7396.0  636056.0       5.47008e7  4.70427e9
 1.0  87.0  7569.0  658503.0       5.72898e7  4.98421e9
 1.0  88.0  7744.0  681472.0       5.99695e7  5.27732e9
 1.0  89.0  7921.0  704969.0       6.27422e7  5.58406e9
 1.0  90.0  8100.0  729000.0       6.561e7    5.9049e9
```

```
1 A_poly = fill_matrix_A(d_of_pol)
```

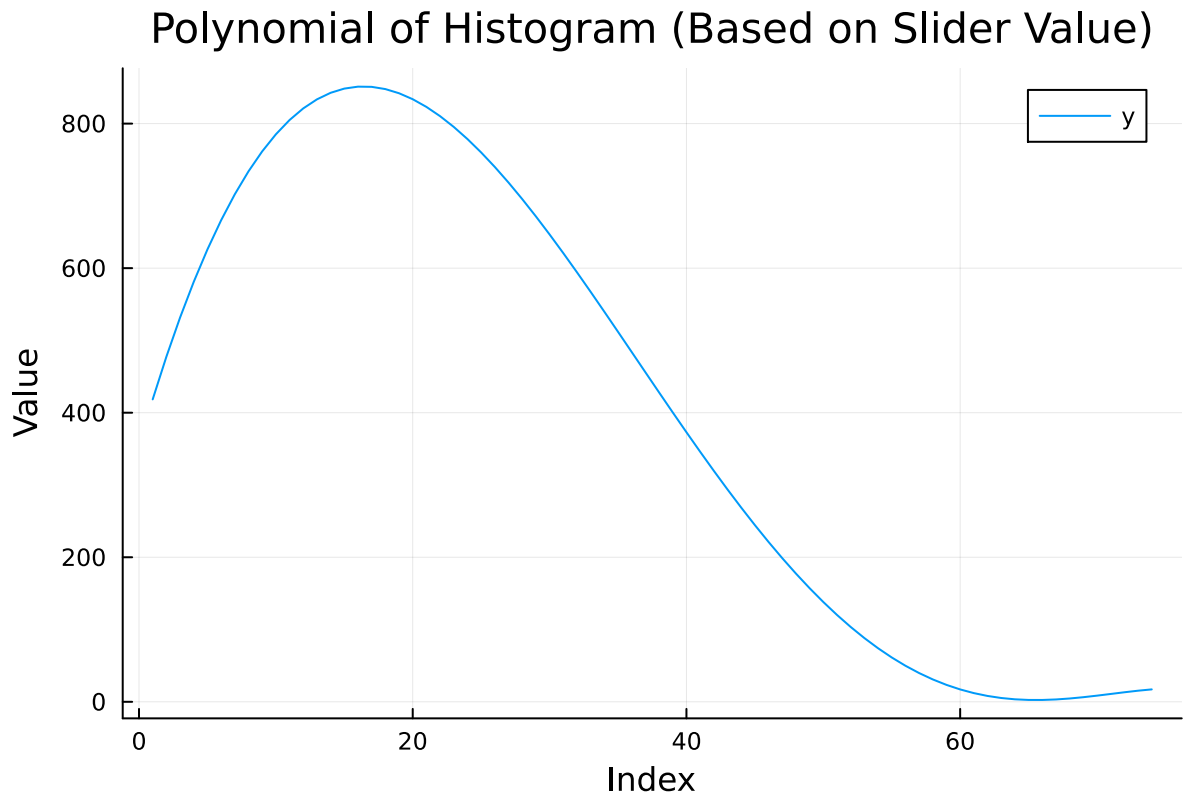
```
(74, 6)
```

```
y_poly =
 [328, 447, 594, 629, 621, 674, 824, 752, 799, 745, 789, 808, 774, 813, 851, 789, 837, 836,
```

```
x_hat =  [-1402.29, 154.904, -2.84363, -0.00443455, 0.00040782, -2.20373e-6]
```

```
y_hat =
 [418.417, 477.433, 531.612, 581.063, 625.9, 666.241, 702.21, 733.932, 761.538, 785.162, 8
```

Slider value is initially 0, don't forget to change it if the plot is not visible



Error Change According to the Polynomial Degree

```
error = 270.3197916431133
```

— Error of degree 5 polynomial is 270.32

— Error rate changes according to the degree value on Slider.

For this part, show your work in the cells below.

```
selected_columns =  [:age, :education_num, :race, :sex, :income]
```

3.1

— Retaining only the age, education_num, race and sex features

df_selected =

	age	education_num	race	sex	income
1	17	7	" White"	" Female"	" <=50K"
2	17	7	" White"	" Male"	" <=50K"
3	17	5	" White"	" Male"	" <=50K"
4	17	3	" White"	" Male"	" <=50K"
5	17	7	" White"	" Male"	" <=50K"
6	17	7	" White"	" Male"	" <=50K"
7	17	5	" White"	" Male"	" <=50K"
8	17	7	" White"	" Male"	" <=50K"
9	17	6	" White"	" Female"	" <=50K"
10	17	7	" White"	" Male"	" <=50K"
more					
30162	90	4	" White"	" Male"	" <=50K"

Converting types to appropriate.

— I converted values of "sex" column to numbers, to be able to categorize them

— 2 represents "Male" and 1 represents "Female"

— I also converted race to numbers so that 2 represents "White" race and 1 represents "non-White"

```
PooledArrays.PooledVector{Any, UInt32, Vector{UInt32}}: [1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2,

[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1,    more ,1, 2, 2, 2, 1, 2, 2, 2

PooledArrays.PooledVector{Any, UInt32, Vector{UInt32}}: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

	age	education_num	race	sex	income
1	17	7	2	1	1
2	17	7	2	2	1
3	17	5	2	2	1
4	17	3	2	2	1
5	17	7	2	2	1
6	17	7	2	2	1
7	17	5	2	2	1
8	17	7	2	2	1
9	17	6	2	1	1
10	17	7	2	2	1
more					
30162	90	4	2	2	1

3.2

— In order to decide on which feature is the strongest indicator, I applied Ordinary Least Square using GLM package.

	age	education_num	race	sex	income
1	17.0	7.0	2.0	1.0	1.0
2	17.0	7.0	2.0	2.0	1.0
3	17.0	5.0	2.0	2.0	1.0
4	17.0	3.0	2.0	2.0	1.0
5	17.0	7.0	2.0	2.0	1.0
6	17.0	7.0	2.0	2.0	1.0
7	17.0	5.0	2.0	2.0	1.0
8	17.0	7.0	2.0	2.0	1.0
9	17.0	6.0	2.0	1.0	1.0
10	17.0	7.0	2.0	2.0	1.0
more					
30162	90.0	4.0	2.0	2.0	1.0

```
ols_age =
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredCh

income ~ 1 + age

Coefficients:
```

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	0.942702	0.00747107	126.18	<1e-99	0.928058	0.957346
age	0.00796663	0.000183926	43.31	<1e-99	0.00760613	0.00832713

ols_education =
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredCh

income ~ 1 + education_num

Coefficients:

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	0.673489	0.00960133	70.15	<1e-99	0.65467	0.692308
education_num	0.0568536	0.00091988	61.81	<1e-99	0.0550506	0.0586566

ols_race =
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredCh

income ~ 1 + race

Coefficients:

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	1.05267	0.013518	77.87	<1e-99	1.02617	1.07916
race	0.105525	0.00714511	14.77	<1e-48	0.0915199	0.119529

ols_sex =
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredCh

income ~ 1 + sex

Coefficients:

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	0.913519	0.00903377	101.12	<1e-99	0.895813	0.931226
sex	0.200159	0.00519229	38.55	<1e-99	0.189982	0.210336

— As it can be seen from tables;

— Age's coefficient is 0.00796663

— Education Number's coefficient is 0.0568536

— Race's coefficient is 0.105525

— Sex's coefficient is 0.200159

— However, it may be a misleading analysis if we do not apply a multiple linear regression.

— For example, an an aged Female worker may be more valued than an young Male or vice versa.

— To verify that, I will apply a linear regression to the data again using GLM package

```
ols_all =
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}, GLM.DensePredCh

income ~ 1 + age + education_num + race + sex

Coefficients:
```

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	0.0325433	0.0169483	1.92	0.0548	-0.000676156	0.0657627
age	0.00694621	0.000169816	40.90	<1e-99	0.00661336	0.00727906
education_num	0.054722	0.000872816	62.70	<1e-99	0.0530112	0.0564328
race	0.0520342	0.00644087	8.08	<1e-15	0.0394099	0.0646586
sex	0.178286	0.00478527	37.26	<1e-99	0.168906	0.187665

— Applying multiple linear regression, we still can see the domination of the sex data on determining the income.

— As the result, We can say that the Sex is more effective than the Race on the income.

— To verify our analysis, I will manually compare the ratio of males above 50K income to all number of males.

— Then I will apply the same operation on Female data

```
6396
```

```
20380
```

```
male_ratio = 0.3138370951913641
```

— Number of males above 50k income / Number of males is 0.314

```
1112
```

```
9782
```

```
female_ratio = 0.11367818442036394
```

— Number of females above 50k income / Number of females is 0.114

Based on LS coefficients and manual calculations, We can see that the **sex** is the strongest indicator for income being above or below 50K.