Raw Work for Demographic Influences on Consumer Perceptions of Zero-Emission Vehicles Research Paper

January 6, 2025

```
[67]: import numpy as np
      from datascience import *

      # These lines do some fancy plotting magic.
      import matplotlib
      %matplotlib inline
      import matplotlib.pyplot as plt
      plt.style.use('fivethirtyeight')
      import warnings
      warnings.simplefilter('ignore', FutureWarning)
      from matplotlib import patches
      from ipywidgets import interact, interactive, fixed
      import ipywidgets as widgets
```

```
[68]: zev = Table().read_table("CAZevSurvey.csv")
      zev
      zevCleaned = zev.select("Respondent Education", "Respondent Gender", "rIncome",␣
        ↪"Respondent's vehicle's monthly miles", "Commute", "Garage or Carport",␣
        ↪"Personal interest in ZEV tech", "Should government offer incentives",␣
        ↪"Replacement: Electricity", "Replacement: Hydrogen", "Consider an EV",␣
        ↪"Familiarity: Gasoline Vehicles", "Familiarity: EVs", "Familiarity: HEVs",␣
        ↪"Incentives: Federal", "Home natural gas", "EV: safety", "EV: reliability",␣
        ↪"Global warming: certainty")
      zevCleaned.show()
```

```
<IPython.core.display.HTML object>
```

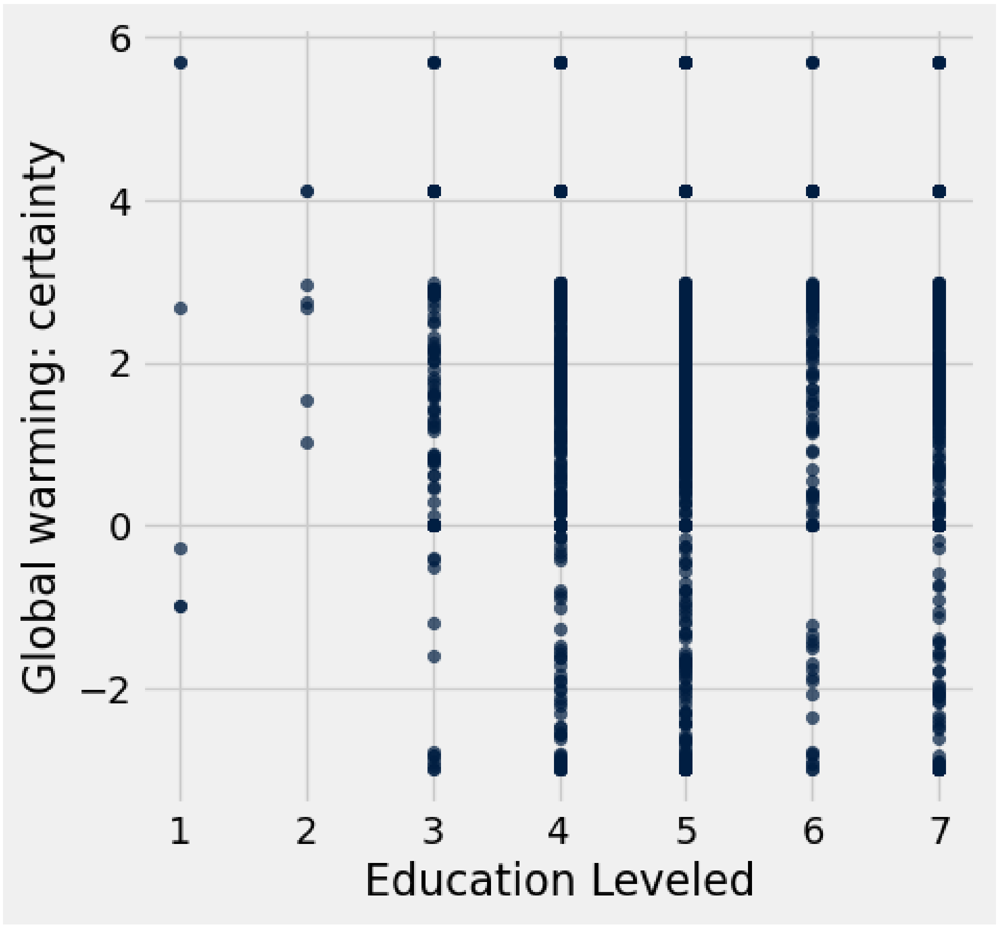
```
[69]: def educationLeveled (edLevel):
          if (edLevel == "Masters, Doctorate, or Professional Degree"):
              level = 7
          if (edLevel == "Some Graduate School"):
              level = 6
          if (edLevel == "College Graduate"):
              level = 5
          if (edLevel == "Some College"):
              level = 4
```

```
    if (edLevel == "High School Graduate or GED"):
        level = 3
    if (edLevel == "Some High School"):
        level = 2
    if (edLevel == "Grade 8 or less"):
        level = 1
    return level

removePreferNotToAnswer = zevCleaned.where("Respondent Education", are.
 ↪not_equal_to ("Prefer not to answer"))
levels = removePreferNotToAnswer.apply(educationLeveled, "Respondent Education")
zevCleanedWithEdLevel = removePreferNotToAnswer.with_column("Education␣
 ↪Leveled", levels)
zevCleanedWithEdLevel.show(5)
```

```
<IPython.core.display.HTML object>
```

[113]:
```
zevCleanedWithEdLevel.scatter("Education Leveled", "Global warming: certainty")
```



[71]:
```
groupedED = zevCleaned.group("Respondent Education")
groupedED
```

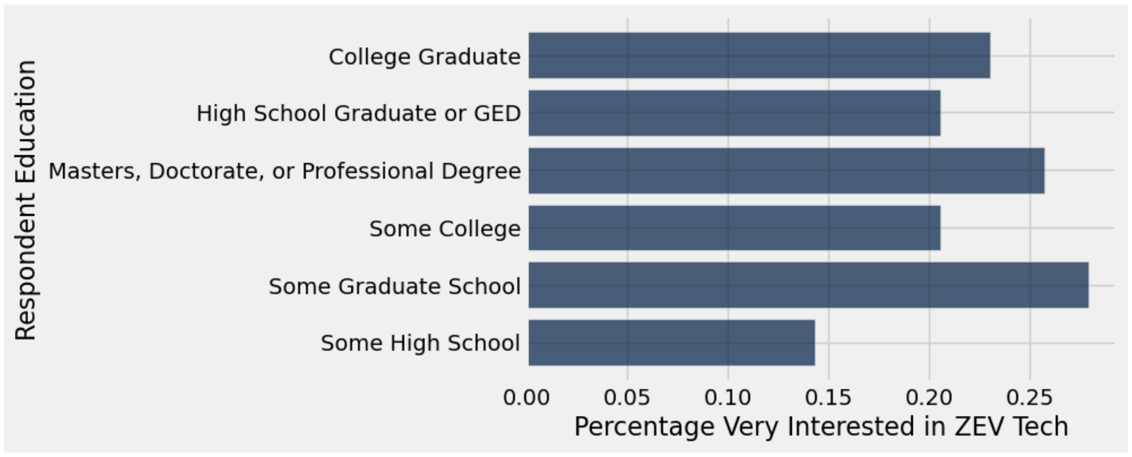
[71]:
```
Respondent Education                          | count
College Graduate                              | 638
Grade 8 or less                               | 6
High School Graduate or GED                   | 112
Masters, Doctorate, or Professional Degree | 381
Prefer not to answer                          | 9
Some College                                  | 389
Some Graduate School                          | 129
Some High School                              | 7
```

[103]:
```
def percentageED (edLevel):
    percentage = (veryInterested.where("Respondent Education", are.
 ↪equal_to(edLevel)).num_rows)/zevCleaned.where("Respondent Education", are.
 ↪equal_to(edLevel)).num_rows
    return percentage

arrayED = groupedED.apply(percentageED, "Respondent Education")
edWithPercentage = groupedED.with_column("Percentage Very Interested in ZEV␣
 ↪Tech", arrayED)
withoutCount5 = edWithPercentage.take(0, 2, 3, 5, 6, 7).drop("count")
edWithPercentage
```

[103]:
```
Respondent Education                          | count | Percentage Very Interested
in ZEV Tech
College Graduate                              | 638   | 0.230408
Grade 8 or less                               | 6     | 0.833333
High School Graduate or GED                   | 112   | 0.205357
Masters, Doctorate, or Professional Degree | 381   | 0.257218
Prefer not to answer                          | 9     | 0
Some College                                  | 389   | 0.205656
Some Graduate School                          | 129   | 0.27907
Some High School                              | 7     | 0.142857
```

[104]:
```
withoutCount5.barh("Respondent Education")
```



```
[77]: groupedConsider = zevCleaned.group("Consider an EV")
      groupedConsider
```

```
[77]: Consider an EV                                             | count
      I (we) already have a vehicle powered by electricity       | 51
      I (we) have not considered buying a vehicle that runs on … | 480
      I (we) have not  and would not  consider buying a vehicl … | 245
      Shopped for an electric vehicle, including a visit to at … | 78
      Started to gather some information, but haven  not reall … | 249
      The idea has occurred, but no real steps have been taken … | 568
```

```
[78]: alrOwn = zevCleaned.where("Consider an EV", are.equal_to("I (we) already have a␣
        ↪vehicle powered by electricity"))
```

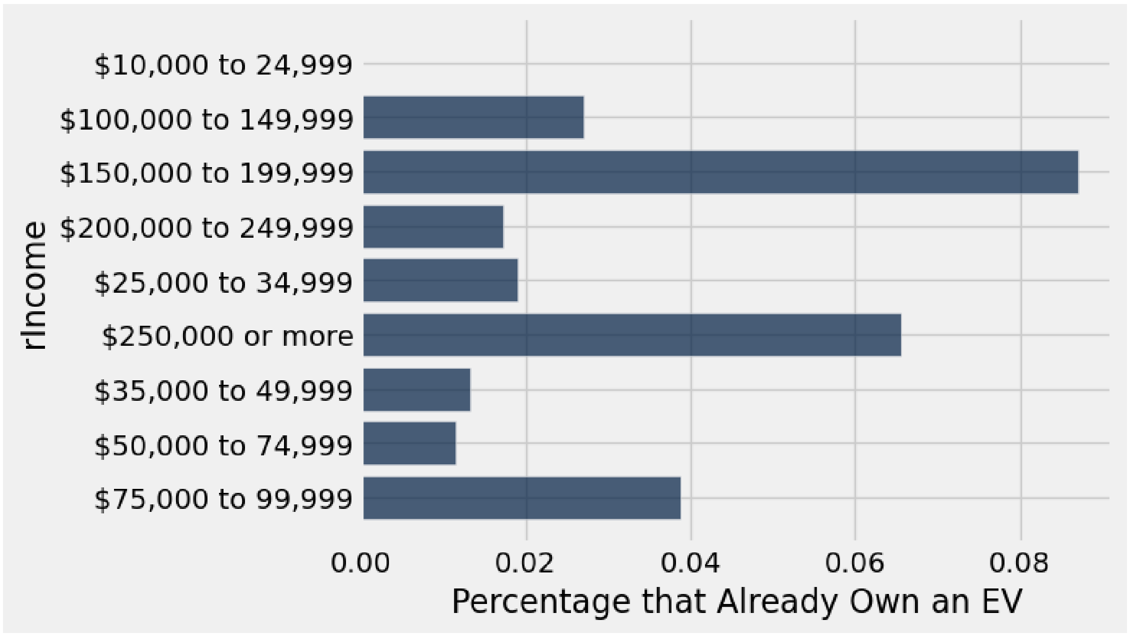
```
[105]: def percentageAlrOwn (incomeLevel):
           percentage = (alrOwn.where("rIncome", are.equal_to(incomeLevel)).num_rows)/
         ↪zevCleaned.where("rIncome", are.equal_to(incomeLevel)).num_rows
           return percentage

       groupedIncome = zevCleaned.group("rIncome")
       incomeArray = groupedIncome.apply(percentageAlrOwn, "rIncome")
       incomesWithPercentageOwn = groupedIncome.with_column("Percentage that Already␣
         ↪Own an EV", incomeArray)
       withoutCount1 = incomesWithPercentageOwn.take(1, 2, 3, 4, 5, 6, 7, 8, 9).
         ↪drop("count")
       incomesWithPercentageOwn
```

```
[105]: rIncome              | count | Percentage that Already Own an EV
       $0 to 9,999          | 26    | 0.0769231
       $10,000 to 24,999    | 52    | 0
       $100,000 to 149,999  | 370   | 0.027027
```

```
$150,000 to 199,999 | 138   | 0.0869565
$200,000 to 249,999 | 58    | 0.0172414
$25,000 to 34,999   | 105   | 0.0190476
$250,000 or more    | 61    | 0.0655738
$35,000 to 49,999   | 152   | 0.0131579
$50,000 to 74,999   | 348   | 0.0114943
$75,000 to 99,999   | 361   | 0.0387812
```

[106]:
```
withoutCount1.barh("rIncome")
```



[80]:
```
neverConsidered = zevCleaned.where("Consider an EV", are.equal_to("I (we) have␣
↪not and would not consider buying a vehicle that runs on electricity"))
```

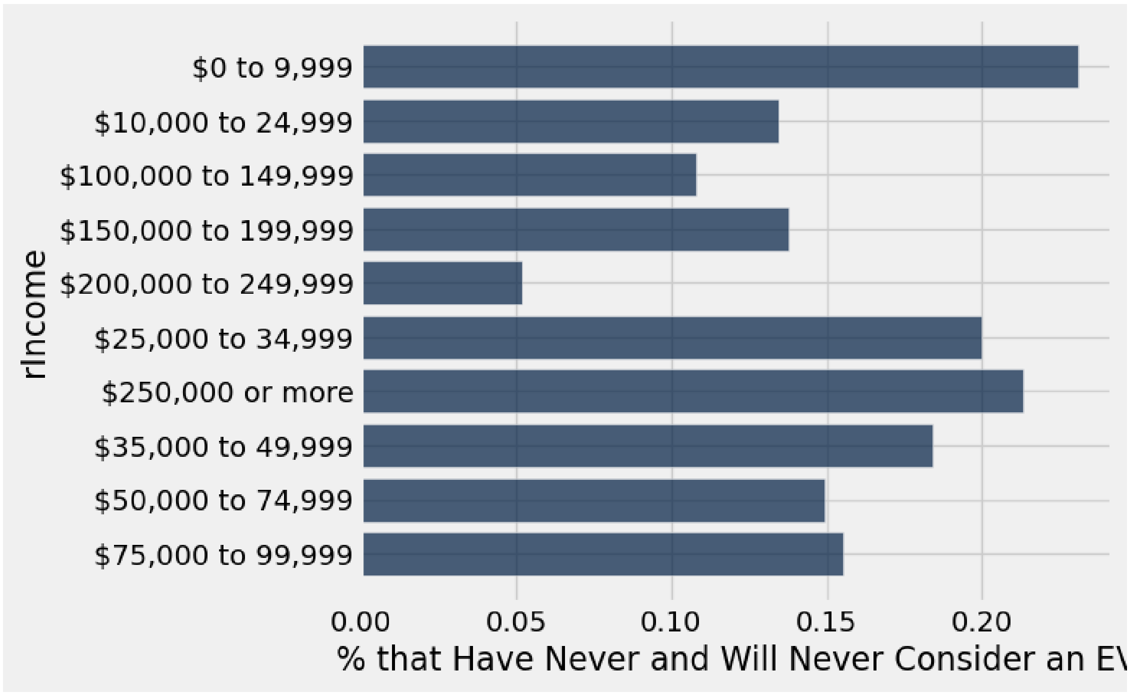
[107]:
```
def percentageNeverConsidered (incomeLevel):
    percentage = (neverConsidered.where("rIncome", are.equal_to(incomeLevel)).
↪num_rows)/zevCleaned.where("rIncome", are.equal_to(incomeLevel)).num_rows
    return percentage

incomeArray = groupedIncome.apply(percentageNeverConsidered, "rIncome")
incomesWithPercentageNever = groupedIncome.with_column("% that Have Never and␣
↪Will Never Consider an EV", incomeArray)
withoutCount2 = incomesWithPercentageNever.drop("count")
incomesWithPercentageNever
```

[107]:
```
rIncome             | count | % that Have Never and Will Never Consider an EV
$0 to 9,999         | 26    | 0.230769
```

```
$10,000 to 24,999   | 52    | 0.134615
$100,000 to 149,999 | 370   | 0.108108
$150,000 to 199,999 | 138   | 0.137681
$200,000 to 249,999 | 58    | 0.0517241
$25,000 to 34,999   | 105   | 0.2
$250,000 or more    | 61    | 0.213115
$35,000 to 49,999   | 152   | 0.184211
$50,000 to 74,999   | 348   | 0.149425
$75,000 to 99,999   | 361   | 0.155125
```

[108]:
```
withoutCount2.barh("rIncome")
```



[82]:
```
def percentageIncome (incomeLevel):
    percentage = (veryInterested.where("rIncome", are.equal_to(incomeLevel)).
 ↪num_rows)/zevCleaned.where("rIncome", are.equal_to(incomeLevel)).num_rows
    return percentage

incomeArray = groupedIncome.apply(percentageIncome, "rIncome")
incomesWithPercentage = groupedIncome.with_column("Percentage Very Interested␣
 ↪in ZEV Tech", incomeArray)
incomesWithPercentage
```

[82]:
```
rIncome             | count | Percentage Very Interested in ZEV Tech
$0 to 9,999         | 26    | 0.230769
$10,000 to 24,999   | 52    | 0.288462
```

```
$100,000 to 149,999 | 370   | 0.208108
$150,000 to 199,999 | 138   | 0.275362
$200,000 to 249,999 | 58    | 0.241379
$25,000 to 34,999   | 105   | 0.209524
$250,000 or more    | 61    | 0.245902
$35,000 to 49,999   | 152   | 0.223684
$50,000 to 74,999   | 348   | 0.241379
$75,000 to 99,999   | 361   | 0.235457
```

[83]:
```
groupedGender = zevCleaned.group("Respondent Gender")
```

[84]:
```
def percentageGender (gender):
    percentage = (veryInterested.where("Respondent Gender", are.
 ↪equal_to(gender)).num_rows)/zevCleaned.where("Respondent Gender", are.
 ↪equal_to(gender)).num_rows
    return percentage

genderArray = groupedGender.apply(percentageGender, "Respondent Gender")
genderWithPercentage = groupedGender.with_column("Percentage Very Interested in␣
 ↪ZEV Tech", genderArray)
genderWithPercentage
```

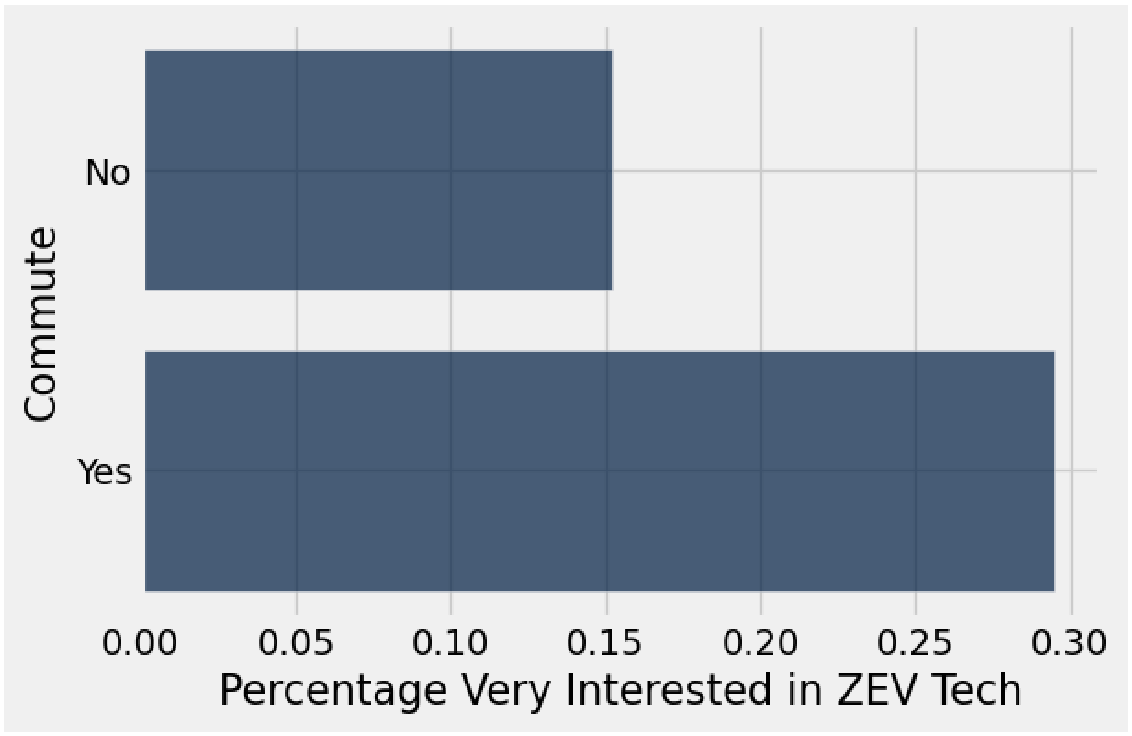
[84]:
```
Respondent Gender | count | Percentage Very Interested in ZEV Tech
Declined To State | 5     | 0.2
Female            | 820   | 0.153659
Male              | 842   | 0.308789
nan               | 4     | 0.75
```

[85]:
```
groupedCommute = zevCleaned.group("Commute")
```

[109]:
```
def percentageCommute (commute):
    percentage = (veryInterested.where("Commute", are.equal_to(commute)).
 ↪num_rows)/zevCleaned.where("Commute", are.equal_to(commute)).num_rows
    return percentage
commuteArray = groupedCommute.apply(percentageCommute, "Commute")
commuteWithPercentage = groupedCommute.with_column("Percentage Very Interested␣
 ↪in ZEV Tech", commuteArray)
withoutCount = commuteWithPercentage.drop("count")
commuteWithPercentage
```

[109]:
```
Commute | count | Percentage Very Interested in ZEV Tech
No      | 718   | 0.151811
Yes     | 953   | 0.294858
```

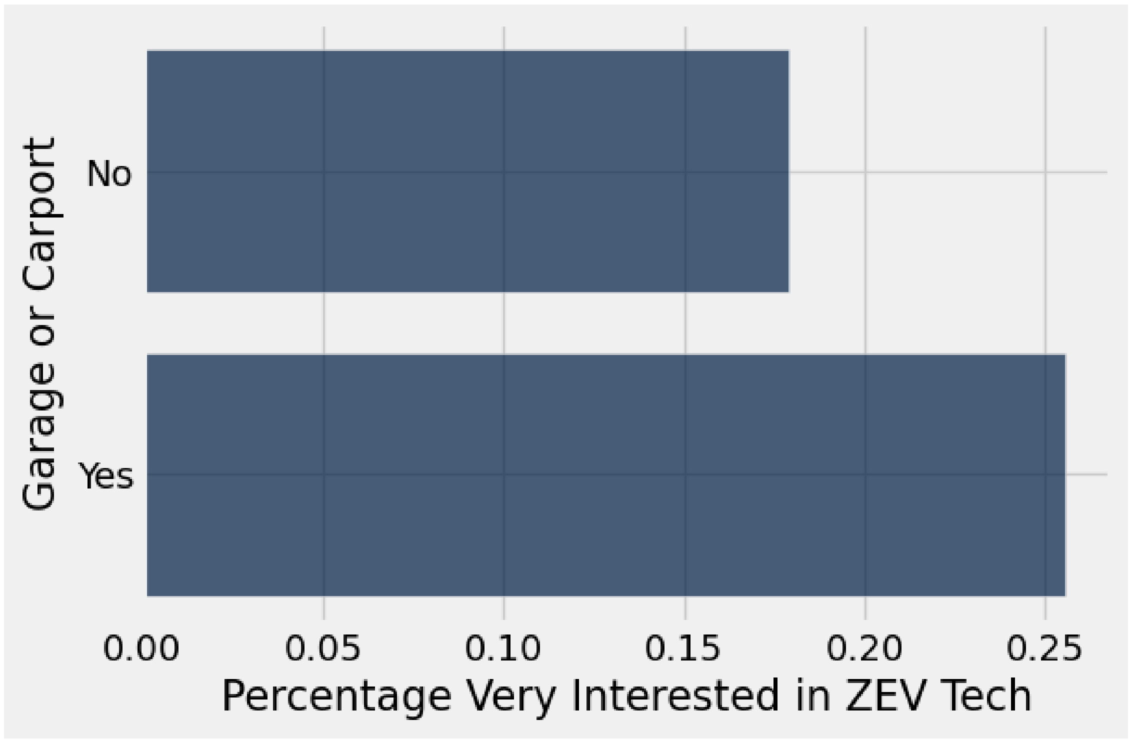
[110]:
```
withoutCount.barh("Commute")
```



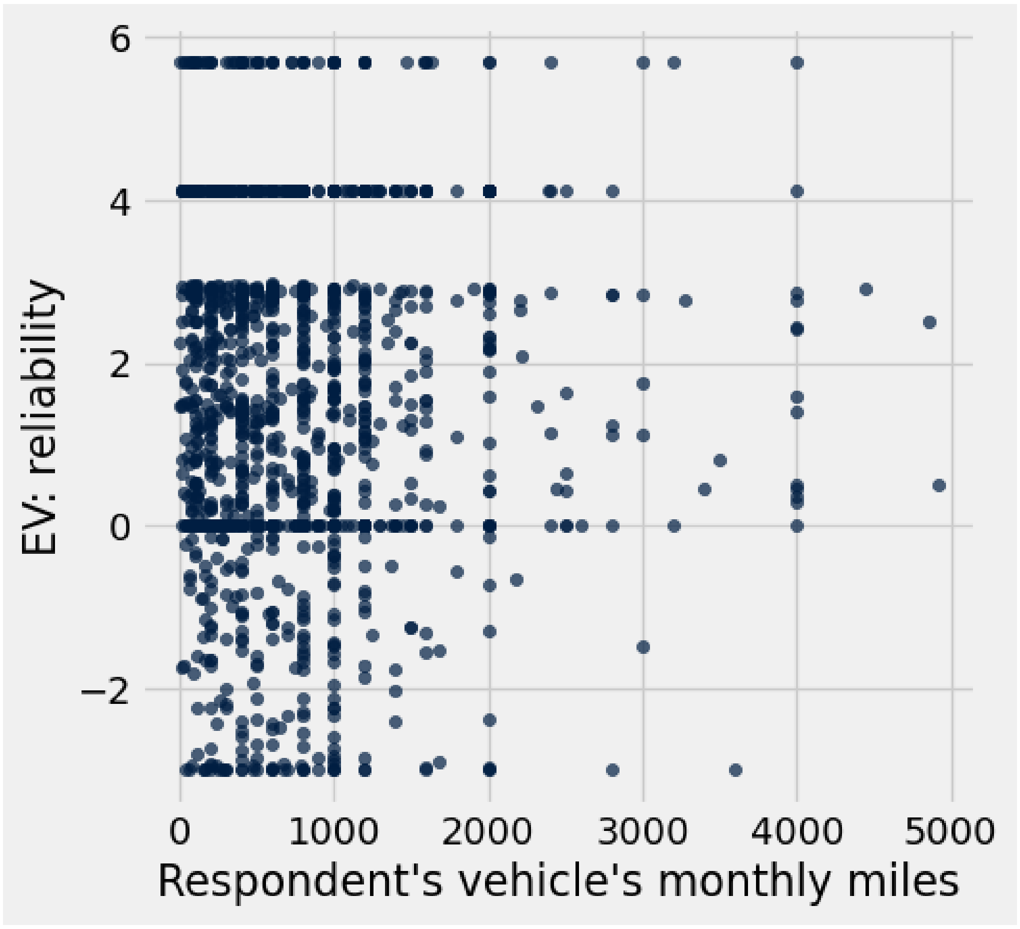
[111]:
```python
groupedGarage = zevCleaned.group("Garage or Carport")
def percentageGarage (garage):
    percentage = (veryInterested.where("Garage or Carport", are.
 ↪equal_to(garage)).num_rows)/zevCleaned.where("Garage or Carport", are.
 ↪equal_to(garage)).num_rows
    return percentage
garageArray = groupedGarage.apply(percentageGarage, "Garage or Carport")
garageWithPercentage = groupedGarage.with_column("Percentage Very Interested in␣
 ↪ZEV Tech", garageArray)
withoutCount3 = garageWithPercentage.drop("count")
garageWithPercentage
```

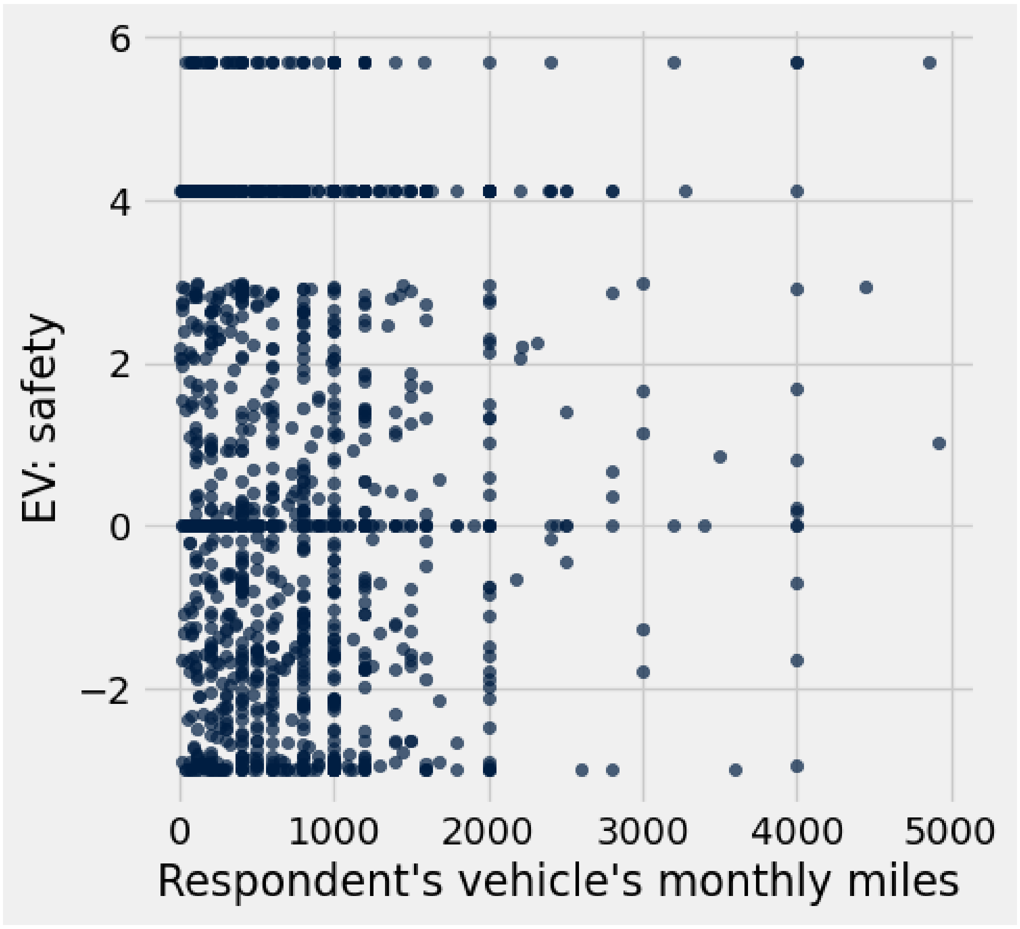
[111]:
```
Garage or Carport | count | Percentage Very Interested in ZEV Tech
No                | 481   | 0.178794
Yes               | 1190  | 0.255462
```

[112]:
```python
withoutCount3.barh("Garage or Carport")
```



```
[88]: zevCleanedShortened = zevCleaned.where("Respondent's vehicle's monthly miles",␣
        ↪are.below(5000))
      zevCleanedShortened.scatter("Respondent's vehicle's monthly miles", "EV:␣
        ↪reliability")
      zevCleanedShortened.scatter("Respondent's vehicle's monthly miles", "EV:␣
        ↪safety")
```


EV: reliability
Respondent's vehicle's monthly miles
6
4
2
0
−2
0
1000
2000
3000
4000
5000



```
[89]: groupedIncentives = zevCleaned.group("Should government offer incentives")
      def percentageIncentives (incentives):
          percentage = (veryInterested.where("Should government offer incentives",␣
       ↪are.equal_to(incentives)).num_rows)/zevCleaned.where("Should government␣
       ↪offer incentives", are.equal_to(incentives)).num_rows
          return percentage
      incentivesArray = groupedIncentives.apply(percentageIncentives, "Should␣
       ↪government offer incentives")
      incentivesWithPercentage = groupedIncentives.with_column("Percentage Very␣
       ↪Interested in ZEV Tech", incentivesArray)
      incentivesWithPercentage
```

```
[89]: Should government offer incentives | count | Percentage Very Interested in ZEV
      Tech
      I'm not sure                       | 246   | 0.0853659
      No, neither one                    | 223   | 0.134529
      Yes, both electricity and hydrogen | 931   | 0.24275
      Yes, but only electricity          | 218   | 0.431193
```

```
Yes, but only hydrogen              | 53    | 0.358491
```

[90]:
```
(veryInterested.where("Should government offer incentives", are.equal_to("Yes,␣
 ↪but only electricity"))).num_rows/veryInterested.num_rows
```

[90]:
```
0.24102564102564103
```

[91]:
```
groupedGas = zevCleaned.group("Home natural gas")
def percentageGas (gas):
    percentage = (veryInterested.where("Home natural gas", are.equal_to(gas)).
 ↪num_rows)/zevCleaned.where("Home natural gas", are.equal_to(gas)).num_rows
    return percentage
gasArray = groupedGas.apply(percentageGas, "Home natural gas")
gasWithPercentage = groupedGas.with_column("Percentage Very Interested in ZEV␣
 ↪Tech", gasArray)
gasWithPercentage
```

[91]:
```
Home natural gas              | count | Percentage Very Interested in ZEV Tech
No, we don't have natural gas | 405   | 0.254321
Yes, we do have natural gas   | 1266  | 0.226698
```

[92]:
```
veryInterested.num_rows
```

[92]:
```
390
```

[93]:
```
#Null Hypothesis: There is no significance to say that a higher proportion of␣
 ↪non-gas users tend to be very interested in ZEV
#compared to the proportion of gas-users. The observation was merely due to␣
 ↪chance

#Alternative Hypothesis: There is significance to say that a higher proportion␣
 ↪of non-gas users tend to be very interested in
#ZEV tech compared to the proportion of gas-users.

#Test Statistic: The difference in the proportion of very interested users that␣
 ↪don't use gas and the proportion that do use gas

#Observed Statistic
observedStatisticGas = (gasWithPercentage.column("Percentage Very Interested in␣
 ↪ZEV Tech").item(0)) - gasWithPercentage.column("Percentage Very Interested␣
 ↪in ZEV Tech").item(1)
```

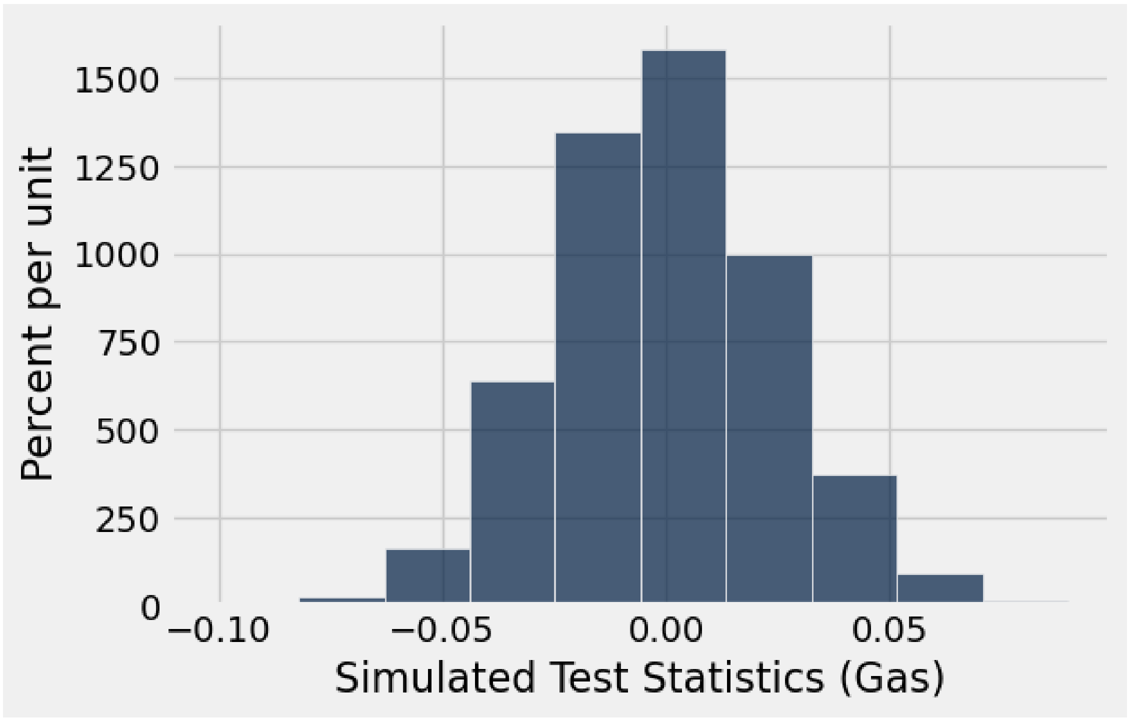
[94]:
```
#10000 Simulations under null hypothesis. Both gas users and non-gas users have␣
 ↪a 1/4 chance that they are very interested in ZEV Tech
modelProportions = make_array(1/4, 3/4)
appendedGasProportions = make_array()
```

```
for i in np.arange(10000):
    oneSimulation = (sample_proportions(405, modelProportions)).
 ↪item(0)-(sample_proportions(1266, modelProportions)).item(0)
    appendedGasProportions = np.append(appendedGasProportions, oneSimulation)

appendedGasProportions
```

```
[94]: array([ 0.02614241,  0.00629571,  0.05853958, …, -0.00042127,
              0.01864139,  0.04727634])
```

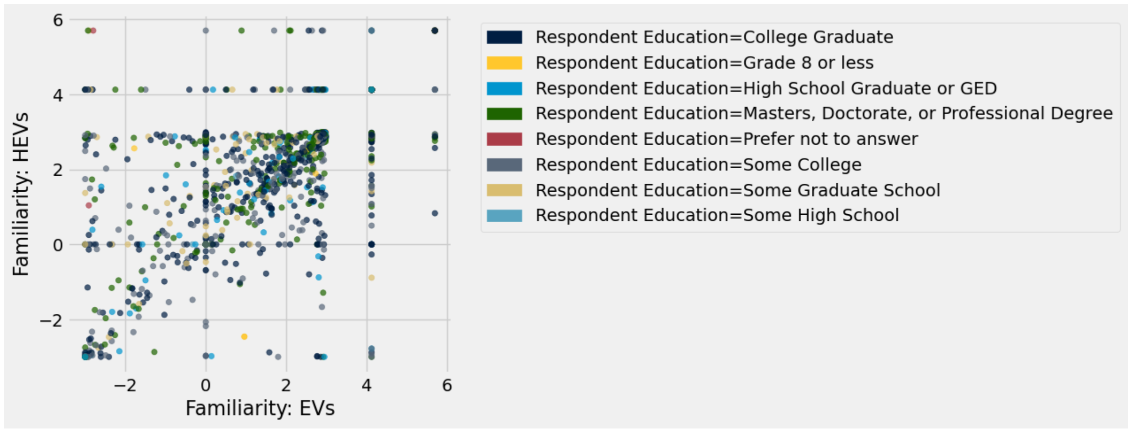
```
[95]: gasVI = Table().with_column("Simulated Test Statistics (Gas)",␣
       ↪appendedGasProportions)
      gasVI.hist()
```



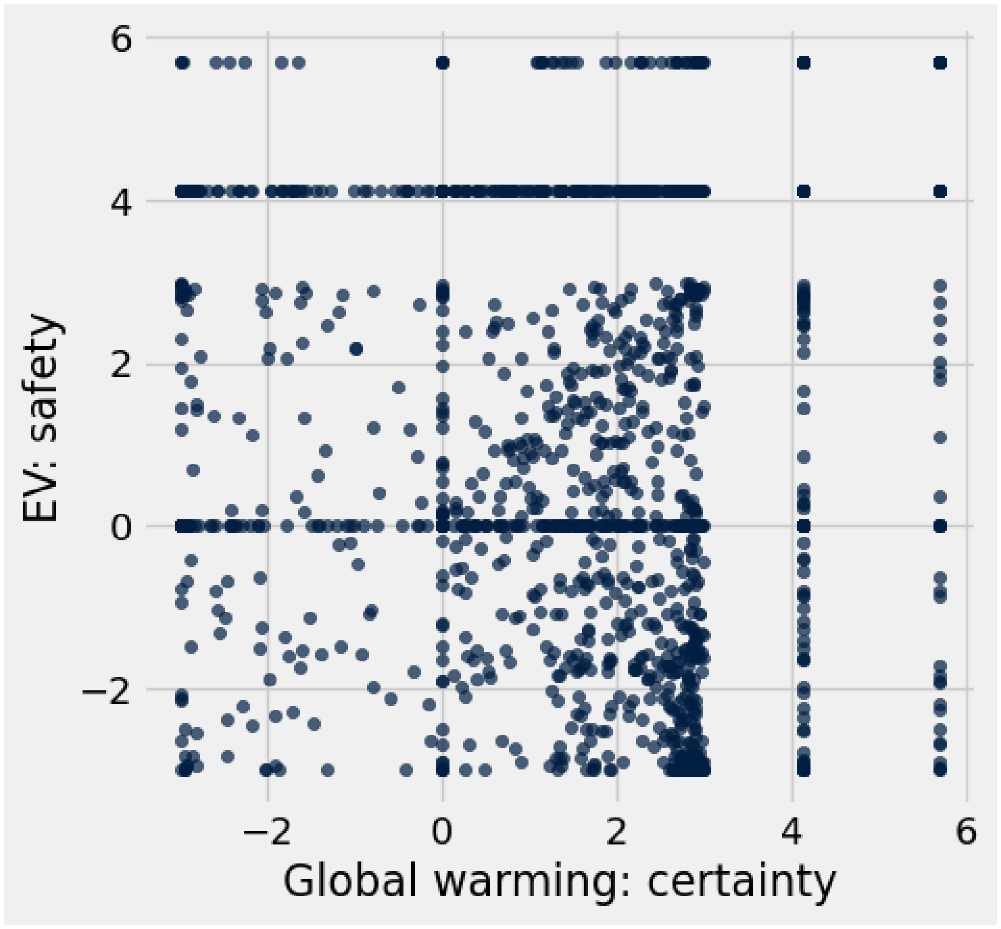
```
[96]: pValueGas = (np.count_nonzero(gasVI.column("Simulated Test Statistics (Gas)")␣
       ↪>= observedStatisticGas)/10000) * 100
      pValueGas
```

```
[96]: 12.76
```

```
[97]: zevCleaned.scatter("Familiarity: EVs", "Familiarity: HEVs", group = "Respondent␣
       ↪Education")
```



```
[98]: zevCleaned.scatter("Global warming: certainty", "EV: safety")
```



```
[99]: notInterested = zevCleaned.where("Personal interest in ZEV tech", are.
      ↪equal_to("Not interested"))
```

[100]:
```
def percentageIncome (incomeLevel):
    percentage = (notInterested.where("rIncome", are.equal_to(incomeLevel)).
 ↪num_rows)/zevCleaned.where("rIncome", are.equal_to(incomeLevel)).num_rows
    return percentage

incomeArray = groupedIncome.apply(percentageIncome, "rIncome")
incomesWithPercentage = groupedIncome.with_column("Percentage Not Interested in␣
 ↪ZEV Tech", incomeArray)
incomesWithPercentage
```

[100]:
```
rIncome             | count | Percentage Not Interested in ZEV Tech
$0 to 9,999         | 26    | 0.269231
$10,000 to 24,999   | 52    | 0.0961538
$100,000 to 149,999 | 370   | 0.116216
$150,000 to 199,999 | 138   | 0.108696
$200,000 to 249,999 | 58    | 0.0689655
$25,000 to 34,999   | 105   | 0.133333
$250,000 or more    | 61    | 0.114754
$35,000 to 49,999   | 152   | 0.151316
$50,000 to 74,999   | 348   | 0.114943
$75,000 to 99,999   | 361   | 0.108033
```

[101]:
```
def percentageED (edLevel):
    percentage = (notInterested.where("Respondent Education", are.
 ↪equal_to(edLevel)).num_rows)/zevCleaned.where("Respondent Education", are.
 ↪equal_to(edLevel)).num_rows
    return percentage

arrayED = groupedED.apply(percentageED, "Respondent Education")
edWithPercentage = groupedED.with_column("Percentage Not Interested in ZEV␣
 ↪Tech", arrayED)
edWithPercentage.take(0, 1, 2, 3, 5, 6, 7)
```

[101]:
```
Respondent Education                          | count | Percentage Not Interested
in ZEV Tech
College Graduate                              | 638   | 0.105016
Grade 8 or less                               | 6     | 0
High School Graduate or GED                   | 112   | 0.133929
Masters, Doctorate, or Professional Degree    | 381   | 0.115486
Some College                                  | 389   | 0.14653
Some Graduate School                          | 129   | 0.0852713
Some High School                              | 7     | 0
```