Exam 1 Info

February 5, 2019

1 Exam 1 Info

Exam 1 will be in class on Tuesday, February 5. You will have the entire class period (170 minutes) to complete the exam. It is open-book, open-Internet. The only rule is that you cannot communicate with another person during the exam.

Your Exam 1 will consist of eight questions about a data set of used cars, available here: `https://raw.githubusercontent.com/dlsun/data-science-book/master/data/usedcars.csv`. I encourage you to explore this data set, come up with interesting questions, and answer those questions to prepare for the exam.

Good luck!

```
In [1]: %matplotlib inline
        import pandas as pd
        import numpy as np

        pd.options.display.max_rows = 30

        cars = pd.read_csv("https://raw.githubusercontent.com/dlsun/data-science-book/master/da
```

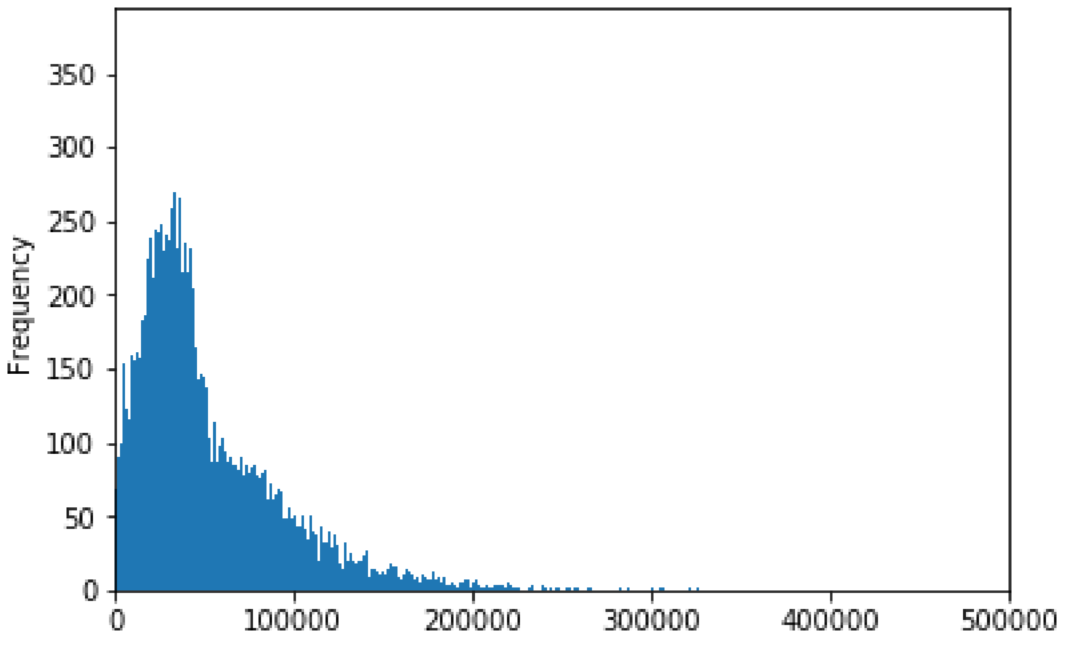
```
In [2]: len(cars.Make.unique())
```

```
Out[2]: 56
```

```
In [3]: cars.State = cars.State.str.upper()
```

```
In [4]: cars.Mileage.plot.hist(xlim=(0,500000), bins=5000)
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7f90532e7748>
```



In [5]: `cars.groupby(["Year"]).Price.describe()`

```
Out[5]:       count          mean           std      min       25%      50%  \
        Year
        1997   43.0   8819.418605  20682.446459   1550.0   3979.50   4975.0
        1998   55.0   7430.090909   9838.256032   1995.0   3503.00   4499.0
        1999   91.0   6723.439560   5409.042616   1600.0   3497.00   4995.0
        2000  116.0   6365.155172   5311.307530   1950.0   3898.75   4920.0
        2001  148.0   7476.817568   7997.241798   1733.0   3268.50   4988.0
        2002  202.0   8517.910891   9236.519396   1500.0   3995.00   5794.5
        2003  331.0   7513.223565   6548.992509   1500.0   3996.00   5995.0
        2004  487.0   9415.178645   8055.512985   1995.0   5224.50   7499.0
        2005  623.0   9838.422151   7324.293090   1550.0   5797.50   7799.0
        2006  851.0   9762.663925   6478.939973   1988.0   5995.00   7900.0
        2007 1203.0  11897.819618   7543.487725   1999.0   6996.50   9995.0
        2008 1448.0  12778.387431  12110.727644   2499.0   7991.00  10742.5
        2009 1168.0  13040.249144   9582.083451   2972.0   8075.00  10995.0
        2010 1557.0  14150.176622   8334.857093   3000.0   9344.00  12495.0
        2011 2317.0  16064.495037   8082.013969   2900.0  10943.00  14495.0
        2012 2864.0  17493.777933  10368.118810   3995.0  11205.25  14995.0
        2013 4498.0  19225.613384   9868.689991   4258.0  12900.00  16997.0
        2014 9480.0  21895.390612  10345.856580   5500.0  14993.00  19814.0
        2015 9164.0  25130.852030  13887.110725   5999.0  16000.00  21995.0
        2016 7957.0  25394.173181  14394.852331   4289.0  15999.00  21700.0
        2017 5339.0  29641.928264  15393.637924   8850.0  19078.00  25900.0
        2018   58.0  42542.586207  11944.668675  19950.0  34541.25  44145.5
```

```
           75%       max
Year
1997   7499.00  139900.0
1998   7990.50   69988.0
1999   7992.50   29999.0
2000   6419.00   35990.0
2001   7908.75   67995.0
2002   8995.00   85914.0
2003   8970.50   89950.0
2004  10998.00   99990.0
2005  11347.50   78960.0
2006  11479.00   90999.0
2007  14981.00  113500.0
2008  14995.00  315000.0
2009  15846.75  234900.0
2010  16995.00  155555.0
2011  19237.00  145777.0
2012  21330.75  169957.0
2013  23700.00  209988.0
2014  26995.00  252000.0
2015  30991.00  299900.0
2016  31206.00  319900.0
2017  35943.00  209899.0
2018  49333.25   79988.0
```

```
In [6]: cars.groupby(["Make"]).Price.mean().sort_values()
```

```
Out[6]: Make
        Oldsmobile        3597.750000
        Isuzu             5090.000000
        Suzuki            6028.550000
        Saturn            6134.230769
        Saab              6552.238095
        Mercury           7067.986486
        Pontiac           7468.397959
        smart             7965.230769
        Scion            11862.988827
        FIAT             12391.021739
        Mitsubishi       12810.701681
        Hyundai          14526.449830
        Volkswagen       14767.844175
        Mazda            15620.998684
        Kia              15687.983020
                             ...
        Genesis          39188.285714
        Land             42245.180258
        Fisker           42997.000000
```

```
Maserati          49671.087719
Porsche           55846.110132
Tesla             59477.400000
Lotus             59950.000000
AM                67995.000000
Alfa              70299.666667
Aston             87333.333333
Bentley          105573.173913
Lamborghini      166398.800000
McLaren          195990.000000
Rolls-Royce      197014.000000
Ferrari          203851.869565
Name: Price, Length: 56, dtype: float64
```

```
In [7]: len(cars.Model.unique())
```

```
Out[7]: 1855
```

```
In [8]: pd.crosstab(cars.Make, cars.State)
```

```
Out[8]: State         AK   AL   AR   AZ   CA   CO   CT   DC   DE   FL ...   SD   TN  \
        Make                                                        ...
        AM             0    0    0    0    0    0    0    0    0    0 ...    0    0
        Acura          0    5    4   11   66   31   11    0    0   63 ...    0   15
        Alfa           0    0    0    0    0    0    0    0    0    0 ...    0    0
        Aston          0    0    0    0    2    0    0    0    0    0 ...    0    0
        Audi           0    3    0   14   80   39   13    0    3   76 ...    0    9
        BMW            3   22    4   49  286   44   26    0    4  209 ...    1   23
        Bentley        0    0    0    1    6    0    0    0    0    5 ...    0    0
        Buick          1    9    6   29   27   15    6    0    1   60 ...    2   11
        Cadillac       1   10    5   16   59   24    5    0    3   97 ...    0   21
        Chevrolet     15  115   54  165  460  130   42    1   14  456 ...   15  151
        Chrysler       0   15    7   16   92   24   10    0    6   75 ...    4   28
        Dodge          4   34   30   63  143   53   20    1    7  135 ...    7   54
        FIAT           0    1    1    6   18    4    1    0    0   11 ...    0    1
        Ferrari        0    0    0    1    5    0    0    0    0    4 ...    0    0
        Fisker         0    0    0    0    0    0    0    0    0    0 ...    0    1
        ...          ...  ...  ...  ...  ...  ...  ...  ...  ...  ... ...  ...  ...
        Plymouth       0    0    0    0    0    0    0    0    0    1 ...    0    0
        Pontiac        0    0    1    3    3    4    2    0    0    5 ...    2    1
        Porsche        1    1    0    5   50    3    0    1    0   22 ...    0    3
        Ram            2   15   14   40   69   39   14    0    4   77 ...    5   31
        Rolls-Royce    0    0    0    0    2    0    0    0    0    2 ...    0    0
        Saab           0    0    0    1    0    0    1    0    0    2 ...    0    1
        Saturn         0    0    0    1    3    3    2    0    0    4 ...    0    1
        Scion          0    5    2    6   31    2    1    0    0   16 ...    1    2
        Subaru         1    4    6   20   54   68   53    1    8   36 ...    1   10
        Suzuki         0    0    0    2    0    2    3    0    0    0 ...    0    0
        Tesla          0    0    0    0    4    0    0    0    0    0 ...    0    2
```

```
Toyota          10    75    50   106   550   106    47     0    13   426  ...     3   112
Volkswagen       1    15     4    43   150    44    16     0     5   138  ...     1    10
Volvo            0     3     2     6    30     8    14     0     1    27  ...     0     4
smart            0     0     3     5     5     3     0     0     0     5  ...     0     1

State           TX    UT    VA    VT    WA    WI    WV    WY
Make
AM               0     0     0     0     0     0     0     0
Acura           66     7    37     0    17     4     0     0
Alfa             1     0     0     0     1     0     0     0
Aston            0     0     0     0     0     0     0     0
Audi            95     9    33     0    30     3     1     0
BMW            221    11    85     0    68    15     3     1
Bentley          3     0     1     0     0     0     0     0
Buick           57     6    17     0    10    16     1     0
Cadillac       121     5    36     0    16     9     1     2
Chevrolet      772    75   194     3   119    74    10     5
Chrysler        83    17    51     2    25    16     1     1
Dodge          242    25    75     3    56    34     1     3
FIAT             9     1     3     0     1     1     0     0
Ferrari          1     0     3     0     2     0     0     0
Fisker           0     0     0     0     0     0     0     0
...            ...   ...   ...   ...   ...   ...   ...   ...
Plymouth         0     0     0     0     0     0     0     0
Pontiac          6     1     6     0     3     1     0     1
Porsche         20     1     6     0     9     0     0     0
Ram            165    16    35     1    31    22     0     3
Rolls-Royce      0     0     1     0     0     0     0     0
Saab             2     0     3     1     0     0     0     0
Saturn           5     2     2     0     3     2     0     0
Scion           15     2    14     0    12     4     0     0
Subaru          43    22    51     7    38    17     4     1
Suzuki           0     0     1     0     2     0     0     0
Tesla            1     0     3     0     0     0     0     0
Toyota         545    42   192    13   139    53     4     2
Volkswagen     132    13    84     3    65    19     2     0
Volvo           27     2    17     0     8     1     0     0
smart            5     0     2     0     2     0     0     0

[56 rows x 51 columns]
```

```
In [9]: toyota_df = cars[cars.Make == "Toyota"]
        #pd.crosstab(toyota_df.Model, toyota_df.State)
        #pd.crosstab(toyota_df.Model, toyota_df.Price)
```

```
In [10]: toyota_pivot = toyota_df.pivot_table(index="Model", columns=["City", "State"],
                                  values="Price", aggfunc=np.mean).fillna(0)
```

```
In [11]: toyota_pivot.sum(axis=1)
```

```
Out[11]: Model
         4Runner2WD       2.724260e+05
         4Runner4WD       1.963568e+06
         4Runner4dr       2.314915e+05
         4Runner4x2       9.658157e+05
         4Runner4x4       2.080082e+06
         4RunnerRWD       9.643220e+05
         86Manual         2.398000e+04
         Avalon           2.848005e+05
         Avalon4dr        5.643240e+05
         AvalonLimited    5.628290e+05
         AvalonPremium    9.842700e+04
         AvalonTouring    1.332960e+05
         AvalonXLE        1.103129e+06
         Camry            6.983550e+05
         Camry2014.5      3.251500e+04
                              ...
         VenzaLE          9.887700e+04
         VenzaLE,         3.610430e+05
         VenzaLimited     5.027800e+04
         VenzaLimited,    1.734890e+05
         VenzaXLE         1.200600e+05
         VenzaXLE,        3.080800e+05
         Yaris            1.379300e+05
         Yaris3-Door      2.208200e+04
         Yaris3dr         4.856900e+04
         Yaris4DR         8.672000e+03
         Yaris4dr         5.714200e+04
         Yaris5-Door      2.289378e+05
         Yaris5dr         1.099500e+04
         YarisBase        7.499000e+03
         YarisFleet       8.995000e+03
         Length: 115, dtype: float64
```

```
In [12]: cars.State.value_counts() / len(cars)
         len(cars[cars.State == " TX"])/len(cars)
```

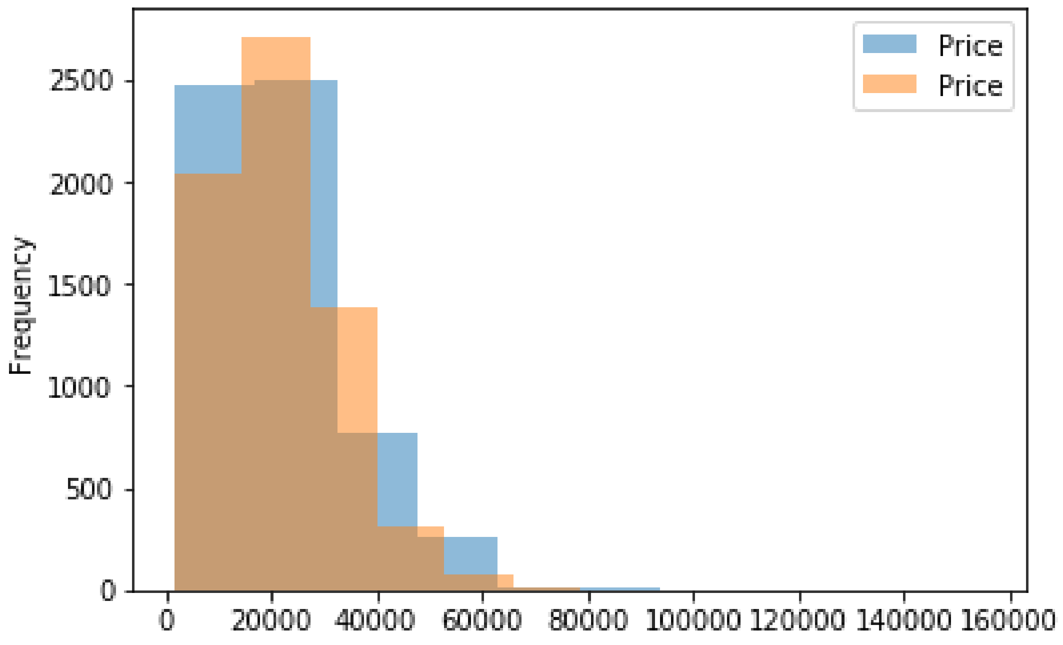
```
Out[12]: 0.11436
```

```
In [13]: cars.Make.value_counts() / len(cars)
         len(cars[cars.Make == "Ford"])/len(cars)
```

```
Out[13]: 0.1308
```

```
In [14]: cars[cars.Make == "Chevrolet"].Price.plot.hist(legend=True, alpha=.5)
         cars[cars.Make == "Ford"].Price.plot.hist(legend=True, alpha=.5)
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9052434ef0>
```



```
In [15]: cars.Make.value_counts().head()

Out[15]: Ford          6540
         Chevrolet     6040
         Toyota        4567
         Nissan        3897
         Honda         2852
         Name: Make, dtype: int64

In [16]: cars.groupby(["Make", "State"])["Price"].describe()

Out[16]:              count          mean           std      min       25%      50%  \
         Make  State
         AM    OR       1.0  67995.000000           NaN  67995.0  67995.00  67995.0
         Acura AL       5.0  13278.000000   8593.805443   3900.0   5000.00  16512.0
               AR       4.0  22018.750000   8627.987072  13988.0  15496.25  20793.5
               AZ      11.0  36832.909091  51755.321070   4995.0  10491.50  24678.0
               CA      66.0  22183.000000   9259.422040   2995.0  15922.50  20446.5
               CO      31.0  24741.870968  14652.434826   2950.0  10799.50  26495.0
               CT      11.0  19457.727273   7018.591099   6990.0  16724.50  20619.0
               FL      63.0  22324.285714  10174.869498   2350.0  14245.00  21994.0
               GA      43.0  23077.697674   9217.018291   2788.0  17999.50  24500.0
               HI       5.0  15304.600000  11321.977424   3199.0   5839.00  13495.0
               ID       3.0  24136.666667   9792.581597  12921.0  20710.00  28499.0
               IL      51.0  24770.647059  11262.176878   3917.0  17454.00  23969.0
               IN       5.0  20115.800000   9590.128425  11495.0  12795.00  15395.0
```

```
       KS      7.0  31233.428571   8431.691801  16999.0  26588.50  31993.0
       KY      5.0  24190.400000   4376.939833  19500.0  20250.00  23995.0
...            ...           ...           ...      ...       ...      ...
smart  HI      1.0   6412.000000           NaN   6412.0   6412.00   6412.0
       IL      1.0   7995.000000           NaN   7995.0   7995.00   7995.0
       KS      2.0   8995.000000      0.000000   8995.0   8995.00   8995.0
       KY      2.0   9875.000000   4065.863992   7000.0   8437.50   9875.0
       LA      5.0   9598.200000   4693.953898   7499.0   7499.00   7499.0
       MD      1.0   5999.000000           NaN   5999.0   5999.00   5999.0
       NJ      2.0   8941.500000   4166.980262   5995.0   7468.25   8941.5
       NV      1.0   6999.000000           NaN   6999.0   6999.00   6999.0
       NY      1.0   7498.000000           NaN   7498.0   7498.00   7498.0
       OK      1.0   7284.000000           NaN   7284.0   7284.00   7284.0
       OR      4.0   8206.500000   2838.876010   5995.0   5995.00   7447.0
       TN      1.0   5765.000000           NaN   5765.0   5765.00   5765.0
       TX      5.0   8300.000000    912.672997   6822.0   7998.00   8890.0
       VA      2.0   6247.000000   2477.702161   4495.0   5371.00   6247.0
       WA      2.0   8432.500000   2209.708691   6870.0   7651.25   8432.5

                   75%       max
Make  State
AM    OR     67995.00   67995.0
Acura AL     16990.00   23988.0
      AR     27316.00   32500.0
      AZ     35409.50  188300.0
      CA     27998.75   43000.0
      CO     38493.50   46000.0
      CT     21496.50   33995.0
      FL     30435.00   47033.0
      GA     27895.00   46491.0
      HI     26995.00   26995.0
      ID     29744.50   30990.0
      IL     32427.00   51771.0
      IN     28995.00   31899.0
      KS     38735.00   38995.0
      KY     28364.00   28843.0
...               ...       ...
smart HI      6412.00    6412.0
      IL      7995.00    7995.0
      KS      8995.00    8995.0
      KY     11312.50   12750.0
      LA      7499.00   17995.0
      MD      5999.00    5999.0
      NJ     10414.75   11888.0
      NV      6999.00    6999.0
      NY      7498.00    7498.0
      OK      7284.00    7284.0
      OR      9658.50   11937.0
```

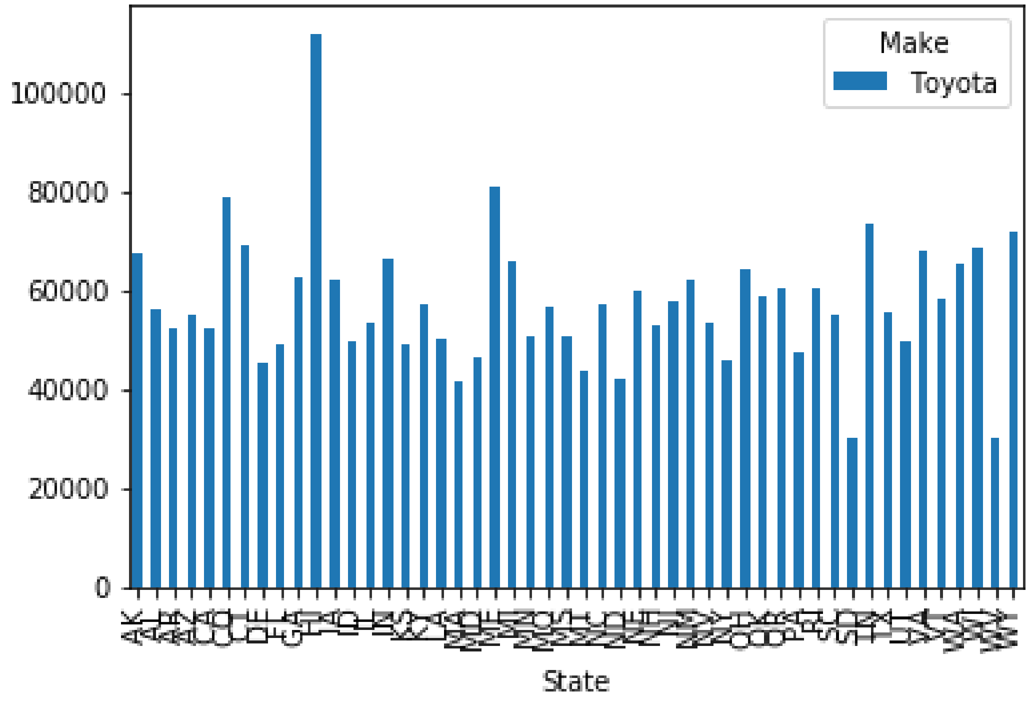
```
       TN     5765.00    5765.0
       TX     8890.00    8900.0
       VA     7123.00    7999.0
       WA     9213.75    9995.0

[1648 rows x 8 columns]
```

```
In [17]: toyota_df.groupby(["Make"])["Model"].value_counts()
```

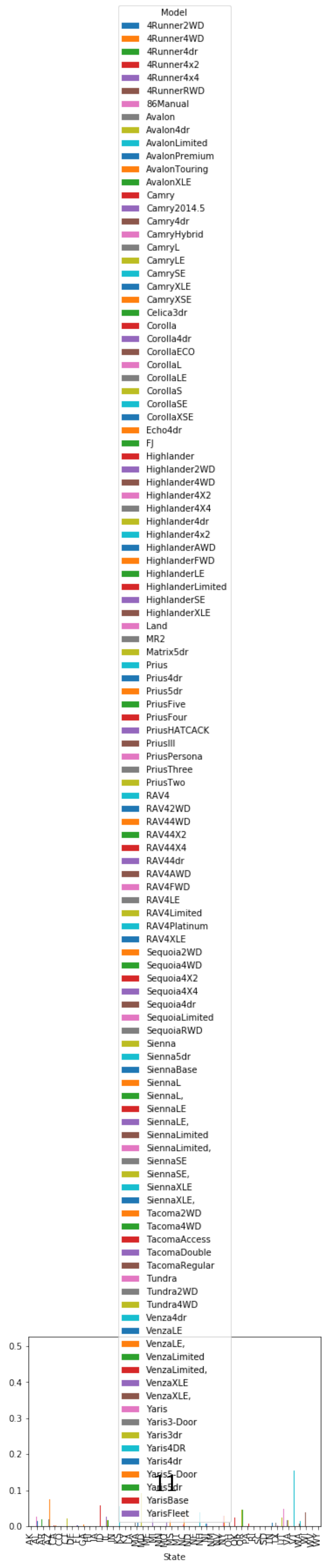
```
Out[17]: Make    Model
         Toyota  Tundra            323
                 CamrySE           288
                 CorollaLE         262
                 RAV44X4           228
                 Camry4dr          194
                 Tacoma2WD         181
                 Tacoma4WD         174
                 CamryLE           166
                 CorollaS          158
                 RAV44X2           124
                 Corolla4dr        121
                 Sienna5dr         100
                 Prius              94
                 TacomaDouble       87
                 Prius5dr           85
                                  ...
                 VenzaLimited        2
                 Yaris3-Door         2
                 86Manual            1
                 Echo4dr             1
                 HighlanderSE        1
                 PriusHATCACK        1
                 PriusIII            1
                 Sequoia2WD          1
                 SequoiaLimited      1
                 Sienna              1
                 SiennaBase          1
                 Yaris4DR            1
                 Yaris5dr            1
                 YarisBase           1
                 YarisFleet          1
         Name: Model, Length: 115, dtype: int64
```

```
In [20]: toyota_df.pivot_table(index="State", columns="Make", values="Mileage",
                                 aggfunc=np.mean).plot.bar()
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9052bd0128>
```

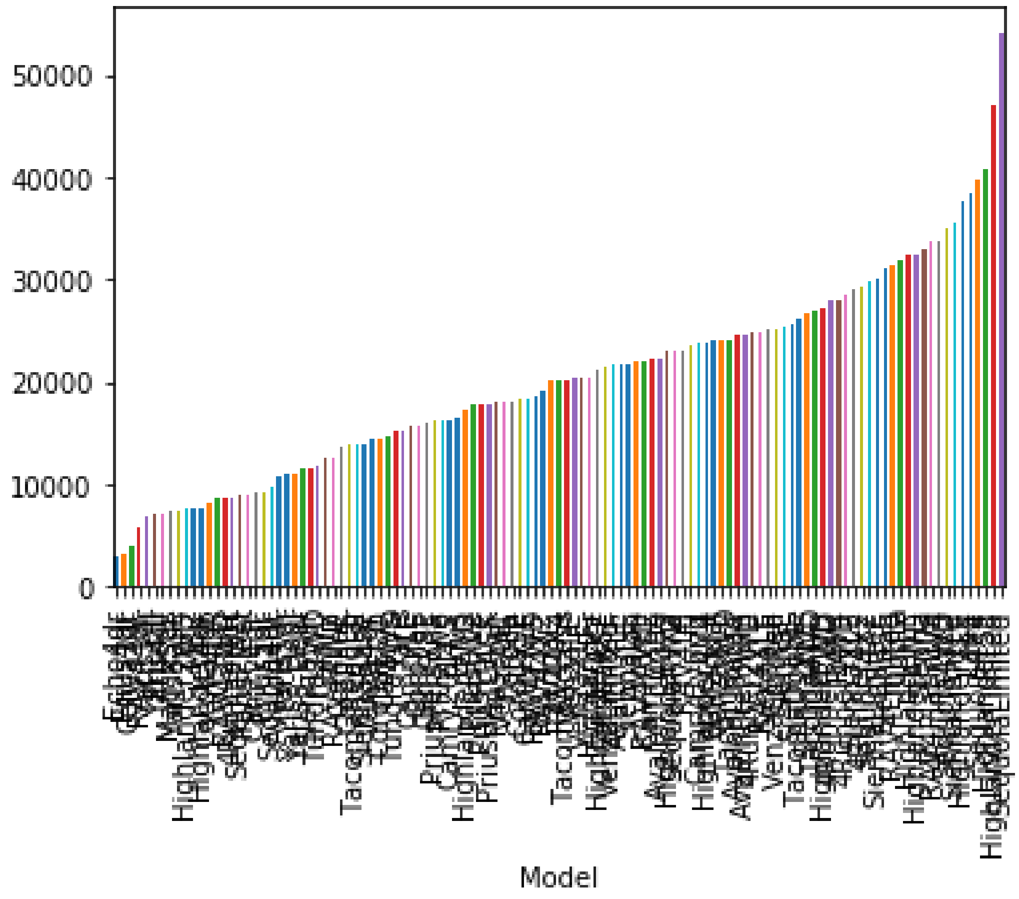


```
In [25]: states_model_counts = pd.crosstab(toyota_df.State, toyota_df.Model)
         states_model_counts

         state_counts = states_model_counts.sum(axis=1)
         states_given_model = states_model_counts.divide(state_counts, axis=0)
         states_given_model.plot.bar()

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7f90493e0358>
```


Model
4Runner2WD
4Runner4WD
4Runner4dr
4Runner4x2
4Runner4x4
4RunnerRWD
86Manual
Avalon
Avalon4dr
AvalonLimited
AvalonPremium
AvalonTouring
AvalonXLE
Camry
Camry2014.5
Camry4dr
CamryHybrid
CamryL
CamryLE
CamrySE
CamryXLE
CamryXSE
Celica3dr
Corolla
Corolla4dr
CorollaECO
CorollaL
CorollaLE
CorollaS
CorollaSE
CorollaXSE
Echo4dr
FJ
Highlander
Highlander2WD
Highlander4WD
Highlander4X2
Highlander4X4
Highlander4dr
Highlander4x2
HighlanderAWD
HighlanderFWD
HighlanderLE
HighlanderLimited
HighlanderSE
HighlanderXLE
Land
MR2
Matrix5dr
Prius
Prius4dr
Prius5dr
PriusFive
PriusFour
PriusHATCACK
PriusIII
PriusPersona
PriusThree
PriusTwo
RAV4
RAV42WD
RAV44WD
RAV44X2
RAV44X4
RAV44dr
RAV4AWD
RAV4FWD
RAV4LE
RAV4Limited
RAV4Platinum
RAV4XLE
Sequoia2WD
Sequoia4WD
Sequoia4X2
Sequoia4X4
Sequoia4dr
SequoiaLimited
SequoiaRWD
Sienna
Sienna5dr
SiennaBase
SiennaL
SiennaL,
SiennaLE
SiennaLE,
SiennaLimited
SiennaLimited,
SiennaSE
SiennaSE,
SiennaXLE
SiennaXLE,
Tacoma2WD
Tacoma4WD
TacomaAccess
TacomaDouble
TacomaRegular
Tundra
Tundra2WD
Tundra4WD
Venza4dr
VenzaLE
VenzaLE,
VenzaLimited
VenzaLimited,
VenzaXLE
VenzaXLE,
Yaris
Yaris3-Door
Yaris3dr
Yaris4DR
Yaris4dr
Yaris5-Door
Yaris5dr
YarisBase
YarisFleet
0.5
0.4
0.3
0.2
0.1
0.0
State

```
In [26]: toyota_df.groupby(["Model"])["Price"].mean().sort_values().plot.bar()
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f904880df60>
```



```
In [29]: cars.Make.value_counts().loc["Jeep"]
```

```
Out[29]: 2377
```

```
In [31]: cars.Make.value_counts().loc["Ferrari"]
```

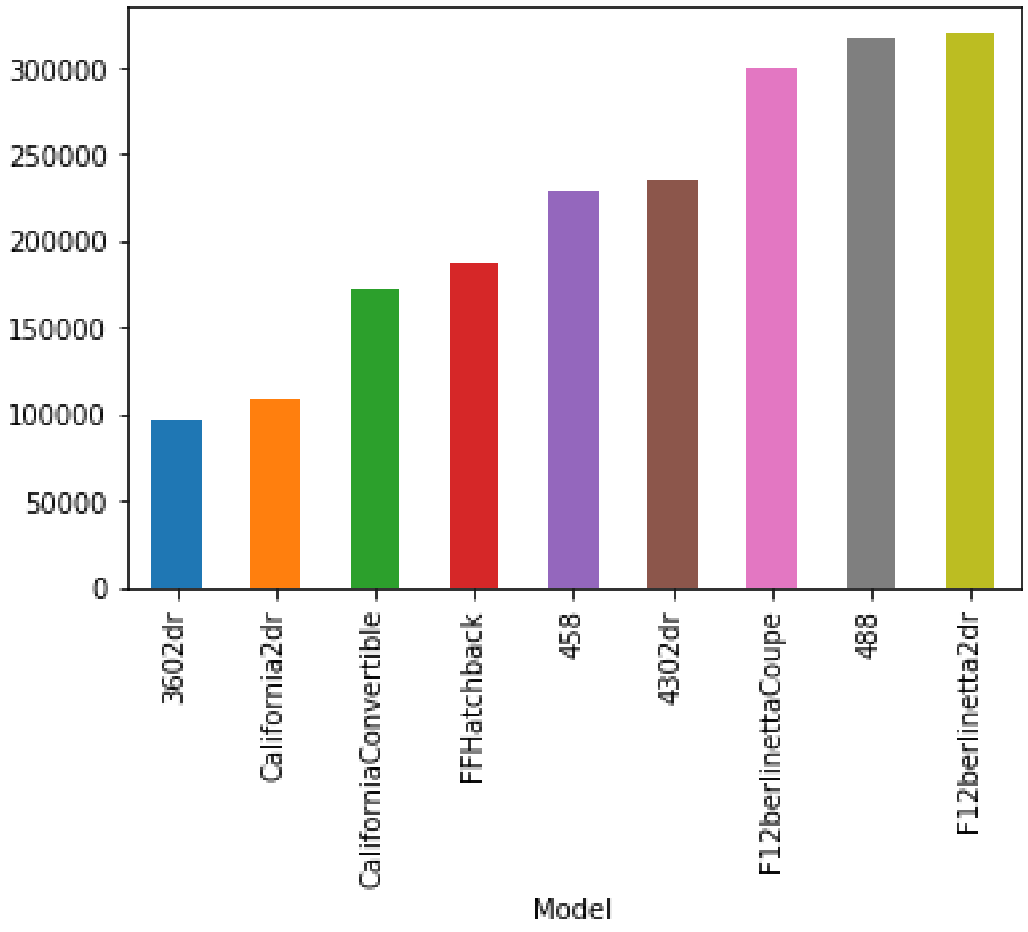
```
Out[31]: 23
```

```
In [32]: ferrari_df = cars[cars.Make == "Ferrari"]
         ferrari_df.head()
```

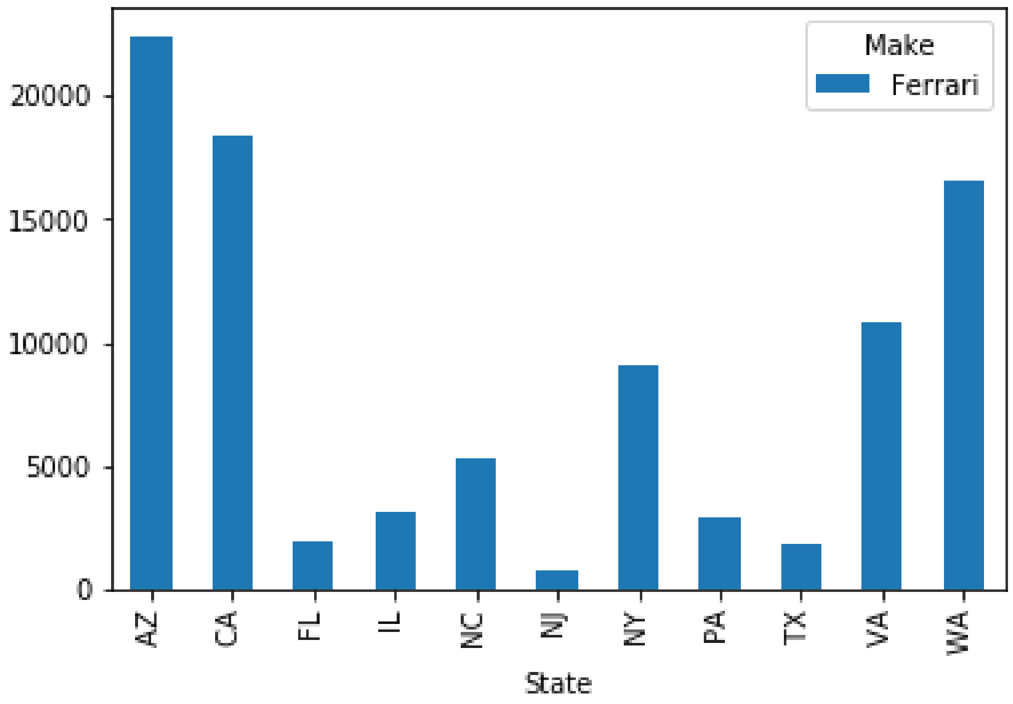
```
Out[32]:        Price  Year  Mileage      City State                Vin     Make  \
         439   246709  2014     4992  Hinsdale    IL  ZFF68NHA3E0199633  Ferrari
         913    89950  2003    22406   Phoenix    AZ  ZFFYT53A130131696  Ferrari
         1650  179985  2013    11760    Dulles    VA  ZFF73SKA2D0189054  Ferrari
         3227  209988  2013    17695    Dulles    VA  ZFF68NHA4D0190941  Ferrari
         5402  169350  2015     1843    Dallas    TX  ZFF77XJA5F0206001  Ferrari

                               Model
         439                     458
         913                  3602dr
         1650             FFHatchback
         3227                    458
         5402  CaliforniaConvertible
```

```
In [33]: ferrari_df.groupby(["Model"])["Price"].mean().sort_values().plot.bar()
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7f903f5c7f98>
```



```
In [35]: ferrari_df.pivot_table(index="State", columns="Make", values="Mileage",
                                aggfunc=np.mean).plot.bar()

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x7f903d0d5e10>
```



```
In [36]: ferrari_df.groupby(["Make", "State"])["Price"].describe()

Out[36]:                 count           mean           std       min       25%  \
         Make    State
         Ferrari  AZ       1.0   89950.000000           NaN   89950.0   89950.0
                  CA       5.0  192331.400000  83220.266461  108880.0  145777.0
                  FL       4.0  260800.000000  60331.639019  188500.0  223300.0
                  IL       3.0  251232.666667  63625.224482  189990.0  218349.5
                  NC       1.0  252000.000000           NaN  252000.0  252000.0
                  NJ       1.0  289000.000000           NaN  289000.0  289000.0
                  NY       1.0  145900.000000           NaN  145900.0  145900.0
                  PA       1.0   99990.000000           NaN   99990.0   99990.0
                  TX       1.0  169350.000000           NaN  169350.0  169350.0
                  VA       3.0  196652.666667  15276.561666  179985.0  189985.0
                  WA       2.0  146945.000000  67818.611384   98990.0  122967.5

                       50%       75%       max
         Make    State
         Ferrari  AZ      89950.0   89950.0   89950.0
```

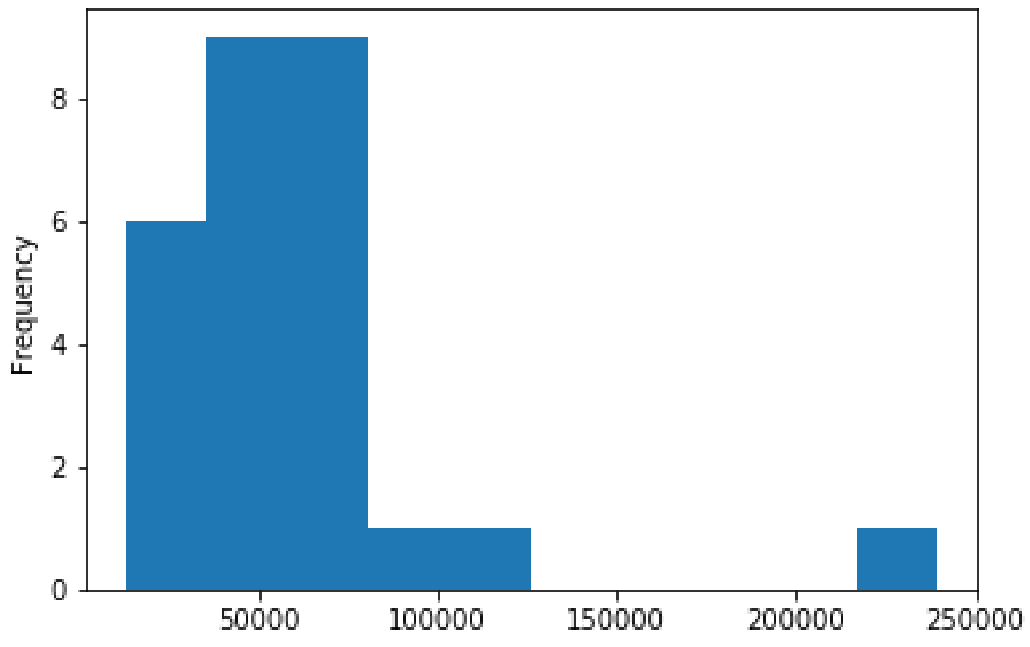
```
CA    158500.0  229500.0  319000.0
FL    267400.0  304900.0  319900.0
IL    246709.0  281854.0  316999.0
NC    252000.0  252000.0  252000.0
NJ    289000.0  289000.0  289000.0
NY    145900.0  145900.0  145900.0
PA     99990.0   99990.0   99990.0
TX    169350.0  169350.0  169350.0
VA    199985.0  204986.5  209988.0
WA    146945.0  170922.5  194900.0
```

```
In [40]: cars.City.value_counts().loc["Fresno"]
```

```
Out[40]: 27
```

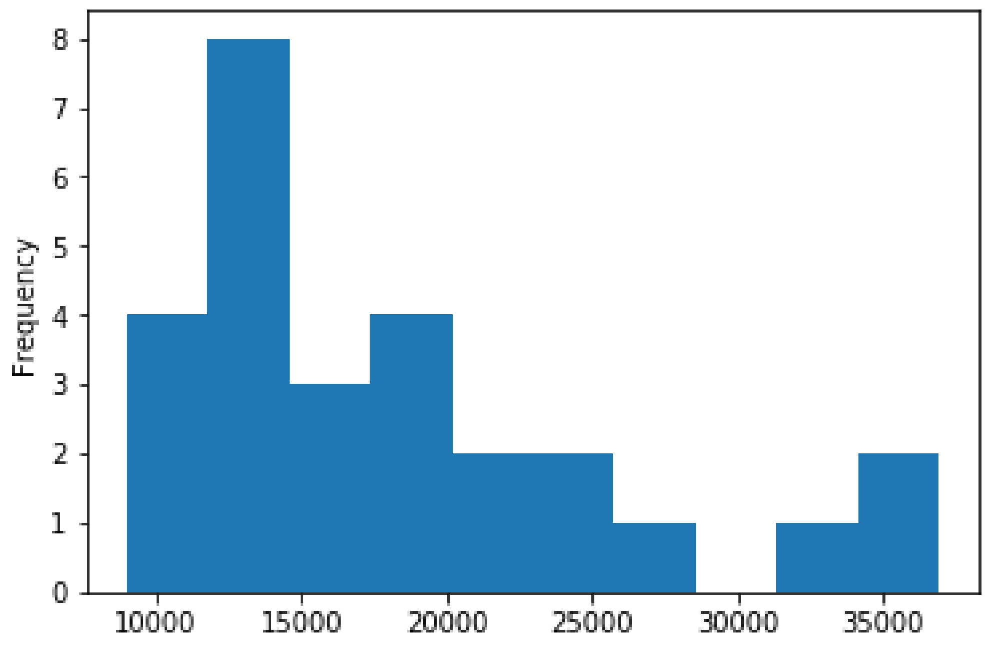
```
In [45]: fresno_df = cars[cars.City == "Fresno"]
         fresno_df.Mileage.plot.hist()
```

```
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x7f903cb9a0f0>
```



```
In [46]: fresno_df.Price.plot.hist()
```

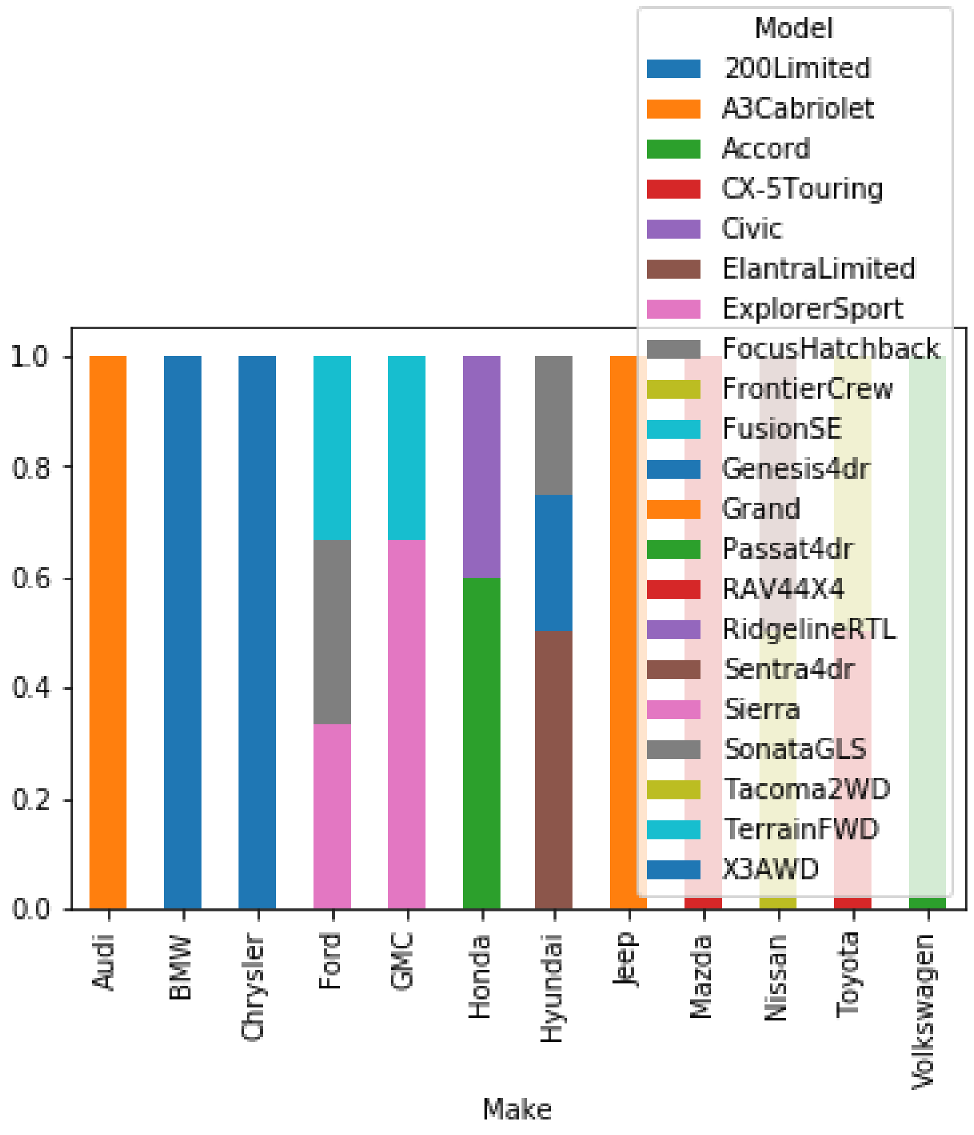
```
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9049727320>
```



```
In [49]: fresno_df.Make.value_counts(), fresno_df.Model.value_counts()

Out[49]: (Honda          5
          Hyundai        4
          GMC            3
          Ford           3
          Volkswagen     2
          Nissan         2
          Toyota         2
          Chrysler       2
          BMW            1
          Audi           1
          Jeep           1
          Mazda          1
          Name: Make, dtype: int64, Accord            3
          Passat4dr          2
          200Limited         2
          Sierra             2
          ElantraLimited     2
          Sentra4dr          1
          CX-5Touring        1
          RAV44X4            1
          FrontierCrew       1
          RidgelineRTL       1
          ExplorerSport      1
          TerrainFWD         1
          X3AWD             1
          SonataGLS         1
          Tacoma2WD         1
          Grand             1
          Civic             1
          Genesis4dr        1
          A3Cabriolet       1
          FocusHatchback    1
          FusionSE          1
          Name: Model, dtype: int64)
```

```
In [52]: make_model_counts = pd.crosstab(fresno_df.Make, fresno_df.Model)
         make_model_counts

         make_counts = make_model_counts.sum(axis=1)
         model_given_make = make_model_counts.divide(make_counts, axis=0)
         model_given_make.plot.bar(stacked=True)
```

```
Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9051f30630>
```



```
In [54]: fresno_df.groupby(["Make"])["Price", "Mileage"].mean()
```

```
Out[54]:                  Price        Mileage
         Make
         Audi        25777.000000   39438.000000
         BMW         11222.000000  110851.000000
         Chrysler    13995.000000   38872.000000
         Ford        18996.333333   59606.000000
         GMC         27084.666667   36468.666667
         Honda       14965.600000   80925.000000
         Hyundai     14144.250000   59747.500000
         Jeep        36901.000000   24599.000000
```

```
Mazda       18997.000000   39328.000000
Nissan      16765.000000   40089.000000
Toyota      21399.500000   81239.500000
Volkswagen  14722.500000   54316.000000
```