**Python数据处理编程**

**实验报告**

****

|  |  |
| --- | --- |
| **姓名** |  |
| **学号** |  |
| **班级** |  |
| **专业** |  |
| **年级** |  |

目录

[一、 问题描述](#_Toc16459_WPSOffice_Level1) [1](#_Toc16459_WPSOffice_Level1)

[1.1 实验目的](#_Toc16459_WPSOffice_Level2) [1](#_Toc16459_WPSOffice_Level2)

[1.2 实验开发环境和工具](#_Toc18983_WPSOffice_Level2) [1](#_Toc18983_WPSOffice_Level2)

[1.3 实验内容](#_Toc26830_WPSOffice_Level2) [1](#_Toc26830_WPSOffice_Level2)

[二、 需求和规格说明](#_Toc18983_WPSOffice_Level1) [1](#_Toc18983_WPSOffice_Level1)

[三、 概要设计](#_Toc26830_WPSOffice_Level1) [5](#_Toc26830_WPSOffice_Level1)

[四、详细设计](#_Toc23828_WPSOffice_Level1) [5](#_Toc23828_WPSOffice_Level1)

[4.1设计思想](#_Toc18863_WPSOffice_Level2) [5](#_Toc18863_WPSOffice_Level2)

[五、结果分析：](#_Toc16578_WPSOffice_Level1) [12](#_Toc16578_WPSOffice_Level1)

[5.1实现注释程序清单](#_Toc29867_WPSOffice_Level2) [12](#_Toc29867_WPSOffice_Level2)

[5.2结果分析](#_Toc3801_WPSOffice_Level2) [13](#_Toc3801_WPSOffice_Level2)

[六、用户手册](#_Toc166_WPSOffice_Level1) [26](#_Toc166_WPSOffice_Level1)

[七、调试及总结](#_Toc15069_WPSOffice_Level1) [26](#_Toc15069_WPSOffice_Level1)

**实验一 基于大学专业的收入可视分析**

1. **问题描述** 
   1. **实验目的**

学习数据可视化的基础知识，了解和掌握通过matplotlib库画折线图（line chart）、多重绘图（multiple plots）、条形图（bar plot）、散点图（scatter plot）、直方图（histogram）、箱型图（box plot）的方法。并学会综合使用matplotlib库的上述绘图方法并结合pandas，在Jupyter 笔记本上掌握可视化快速浏览数据的方法和技巧。

* 1. **实验开发环境和工具**

可以在Windows或Linux操作系统上搭建开发环境，使用Anaconda科学计算包安装Python 3，里面会包含大多数科研常用的库。.ipynb文件推荐使用Anaconda内置的web版Jupyter Notebook。

* 1. **实验内容**

在Educoder平台“数据可视化探索”的折线图（line chart）、多重绘图（multiple plots）、条形图（bar plot）、散点图（scatter plot）、直方图（histogram）、箱型图（box plot）等部分一直都是直接使用pyplot和matplotlib创建图。 当我们想通过快速创建可视化文件来探索新的数据集时，直接使用这些工具会很麻烦。 值得庆幸的是，pandas有许多方法可以根据DataFrames中的数据快速生成公共图。 与pyplot一样，pandas中的绘图功能是matplotlib的包装。 这意味着我们可以在必要时通过访问基础的Figure，Axes和其他matplotlib对象来自定义绘图。在这个指导项目中，我们将探讨如何使用pandas,matplotlib以及 Jupyter 笔记本界面使我们能够使用可视化快速浏览数据。

1. **需求和规格说明**

1.数据清洗[1]：

我们将使用一个数据集，研究 2010 年至 2012 年间大学毕业的学生的工作成果。就业结果的原始数据是由[美国社区调查](https://www.census.gov/programs-surveys/acs/)公布的，调查并汇总了数据,清理了数据集，并在其 [Github](https://github.com/fivethirtyeight/data/tree/master/college-majors) 上发布。数据集中的每一行代表大学中的不同专业，包含有关性别多样性、就业率、工资中位数等的信息。下面是数据集中的一些列：

* Rank- 按收入中位数排名（数据集按此列排序）。
* Major\_code-主要代码。
* Major- 主要描述。
* Major\_category- 专业类别。
* Total- 有少校的总人数。
* Sample\_size- 全职的样本大小（无加权）。
* Men-男毕业生
* Women-女毕业生
* ShareWomen- 妇女占总数的一部分。
* Employed- 就业人数。
* Median- 全职全年工人工资中位数。
* Low\_wage\_jobs- 低工资服务职位的数量。
* Full\_time- 就业人数为35小时以上。
* Part\_time- 就业人数少于35小时。

在开始创建数据可视化之前，让我们导入所需的库并删除包含 null 值的行。具体要求如下：

* 让我们通过导入所需的库并运行必要的Jupyter magic来设置环境，以便以内嵌方式显示图。
  + 将pandas和matplotlib导入环境。
  + 运行Jupyter magic ％matplotlib inline，以便内联显示绘图。
* 将数据集读入DataFrame并开始探索数据。
  + 读入news-grads.csv到pandas中，然后将结果数据框分配给recent\_grads。
  + 使用DataFrame.iloc []返回格式化为表格的第一行。
  + 使用DataFrame.head（）和DataFrame.tail（）来熟悉数据的结构。
  + 使用DataFrame.describe（）为所有数字列生成摘要统计信息。
* 删除缺少值的行。 Matplotlib期望我们传入的值的列具有匹配的长度，而缺少的值将导致matplotlib引发错误。
  + 在recent\_grads索引中查找行数，并将该值分配给raw\_data\_count。
  + 使用DataFrame.dropna（）可以删除包含缺失值的行，并将所得的DataFrame分配回recent\_grads。
  + 现在查看recent\_grads中的行数，并将其值分配给cleaned\_data\_count。如果比较cleaned\_data\_count和raw\_data\_count，您会注意到只有一行包含缺失值并被删除。

2.利用散点图探索[2]：

pandas中的大多数绘图功能都包含在DataFrame.plot（）方法内。 调用此方法时，我们指定要绘制的数据以及绘制的类型。 我们使用kind参数指定所需的绘图类型。 我们使用x和y来指定每个轴上所需的数据。 您可以在[文档](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.html)中阅读有关不同参数的信息。

（1）recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter')

如果您在jupyter笔记本中创建一个新单元格并运行上面的代码，则散点图将立即显示。 此功能是运行jupyter magic ％matplotlib inline的结果。 这意味着我们可以编写一行代码来生成散点图，使用键盘快捷键运行单元格，检查该图并重复。 DataFrame.plot（）方法具有一些可用于调整散点图的参数：

（1）recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter', title='Employed vs. Sample\_size', figsize=(5,10))

我们可以通过将返回值分配给变量来访问底层的matplotlib Axes对象：

（1）ax = recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter')

（2）ax.set\_title('Employed vs. Sample\_size')

当你在jupyter笔记本单元中运行以上代码时，将像以前一样以inline方式返回图。

具体要求如下：

* 在单独的jupyter笔记本单元中生成散点图，以探索以下关系：
  + Sample\_size 和 Median
  + Sample\_size 和 Unemployment\_rate
  + Full\_time 和 Median
  + ShareWomen 和 Unemployment\_rate
  + Men 和 Median
  + Women 和 Median
* 使用图表探索以下问题：
  + 热门专业的学生会赚更多钱吗？
  + 主修女性的专业学生赚钱更多吗？
  + 全职雇员人数与工资中位数之间有联系吗？

3.探索列中值的分布

要探索列中值的分布，我们可以从DataFrame中选择它，调用[Series.plot（）](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.plot.html)，并将kind参数设置为hist：

（1）recent\_grads['Sample\_size'].plot(kind='hist')

DataFrame.plot（）和Series.plot（）方法具有许多相同的参数，但用于不同的用例。 我们使用Series.plot（）绘制特定列，并使用DataFrame.plot（）生成使用多个列中的值的图。 例如，由于散点图是使用2组值（每个轴一组）生成的，因此无法使用Series.plot（）创建散点图。

如果查看了Series.plot（）的文档，您会发现无法控制直方图的合并策略。 幸运的是，我们可以使用[Series.hist（）](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.hist.html)控制直方图的合并策略，该策略包含特定于自定义直方图的参数：

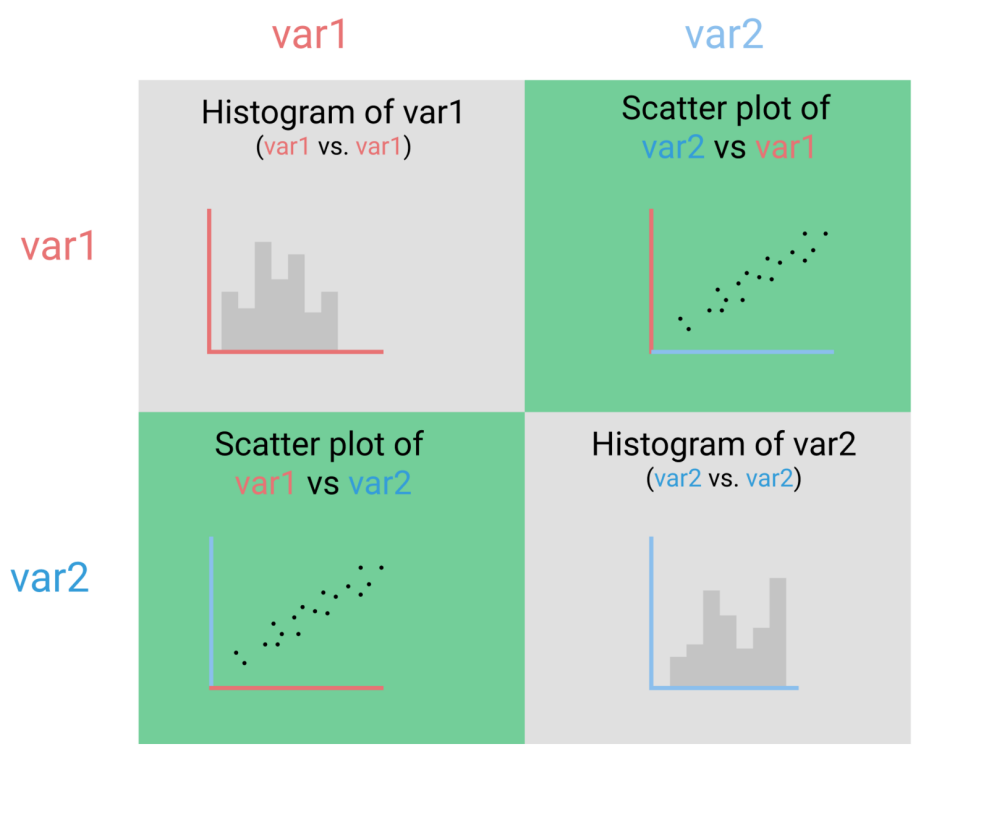
（1）recent\_grads['Sample\_size'].hist(bins=25, range=(0,5000))

具体要求如下：

* 在单独的jupyter笔记本单元格中生成直方图，以探索以下各列的分布： -Sample\_size
  + Median
  + Employed
  + Full\_time
  + ShareWomen
  + Unemployment\_rate
  + Men
  + Women
* 我们鼓励您在生成这些直方图时尝试使用不同的bin大小和范围。
* 使用图表探索以下问题：
  + 专业中占百分之几的主要是男性？ 主要是女性？
  + 最常见的工资中位数是多少？

4.散点矩阵图

在最后两个步骤中，我们创建了单个散点图，以可视化列与直方图之间的潜在关系，并可视化单个列的分布。散点矩阵图将散点图和直方图都组合到一个图网格中，使我们能够同时探索潜在的关系和分布。 散布矩阵图由网格上的n×n图组成，其中n是列数，对角线上的图是直方图，非对角线上的图是散点图。



由于探索性数据分析中经常使用散点矩阵图，因此pandas包含一个名为scatter\_matrix（）的函数，可为我们生成这些图。 此功能是pandas.plotting模块的一部分，需要单独导入。 要生成2列的散点矩阵图，请仅选择这2列，并将结果DataFrame传递到scatter\_matrix（）函数中。

（1）scatter\_matrix(recent\_grads[['Women', 'Men']], figsize=(10,10))

在传递带有2列的DataFrame时，将返回2 x 2散点图（总共4个图），而传递3则返回3 x 3散点图（总共9个图）。 这意味着生成的图的数量以2的倍数成比例缩放，而不是线性的。 如果将列数增加到4或更多，则生成的绘图网格将变得不可读且难以解释（即使使用figsize参数增加了绘图区域）。

如果您想了解有关该函数接受的参数的更多信息，请转至[文档](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.plotting.scatter_matrix.html)。

具体要求如下：

* 从pandas.plotting模块导入scatter\_matrix。
* 使用Sample\_size和Median列创建2 x 2散布矩阵图。
* 使用Sample\_size，Median和Unemployment\_rate列创建3 x 3散点图。
* 使用这些散点图，探索最后几个步骤中的问题。 您可能需要创建更多的散点图。

5.条形图

要在matplotlib中创建条形图，我们必须自己指定条形图的许多方面。 我们必须指定条的位置，标签，长度和宽度。 使用pandas创建条形图时，我们仅需要指定我们希望条形表示的数据以及每个条形的标签。 以下代码返回Women列中前5个值的条形图：

（1）recent\_grads[:5]['Women'].plot(kind='bar')

默认情况下，pandas将对matplotlib中的每个条（1到n）在x轴上使用默认标签。 如果改为使用[DataFrame.plot.bar（）](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.bar.html)方法，则可以使用x参数指定标签，使用y参数指定钢筋的数据：

（2）recent\_grads[:5].plot.bar(x='Major', y='Women')

具体要求如下：

* 使用条形图来比较recent\_grads数据帧的前十行和后十行中的女性（ShareWomen）百分比。
* 使用条形图来比较recent\_grads数据帧的前十行和后十行的失业率（Unemployment\_rate）。

1. **概要设计**
2. **数据清洗**

运用DataFrame.dropna（）可以删除包含缺失值的行。 Matplotlib期望我们传入的值的列具有匹配的长度，而缺少的值将导致matplotlib引发错误。并查看recent\_grads删除空行后的行数如果比较cleaned\_data\_count和raw\_data\_count，注意到只有一行包含缺失值并被删除，则清理成功。

1. **利用散点图探索**

pandas中的大多数绘图功能都包含在DataFrame.plot（）方法内。 调用此方法时，我们指定要绘制的数据以及绘制的类型。 我们使用kind参数指定所需的绘图类型。 我们使用x和y来指定每个轴上所需的数据。 您可以在[文档](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.html)中阅读有关不同参数的信息。

1. **探索列中值的分布**

要探索列中值的分布，我们可以从DataFrame中选择它，调用[Series.plot（）](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.plot.html)，并将kind参数设置为hist。DataFrame.plot（）和Series.plot（）方法具有许多相同的参数，但用于不同的用例。 我们使用Series.plot（）绘制特定列，并使用DataFrame.plot（）生成使用多个列中的值的图。

1. **散点矩阵图**

散点矩阵图将散点图和直方图都组合到一个图网格中，使我们能够同时探索潜在的关系和分布。 散布矩阵图由网格上的n×n图组成，其中n是列数，对角线上的图是直方图，非对角线上的图是散点图。

1. **条形图**

要在matplotlib中创建条形图，我们必须自己指定条形图的许多方面。 我们必须指定条的位置，标签，长度和宽度。 使用pandas创建条形图时，我们仅需要指定我们希望条形表示的数据以及每个条形的标签。

**四、详细设计**

**4.1设计思想**

1. **数据清洗**

根据分析需要我们将数据进行显示，并清洗删除空行。

首先导入需要的包：

%matplotlib inline

import pandas as pd

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

进行数据前几行的显示，确保我们读取数据准确。

recent\_grads = pd.read\_csv('recent-grads.csv')

df = pd.DataFrame(recent\_grads)

print("数据的第一行为:\n",df.iloc[0])

#输出CSV文件的前5行和最后5行，pandas默认的输出5行

print("\n数据的前5行为:\n",df.head())

print("\n数据的前5行为:\n",df.tail())

print("\n数据摘要统计:\n",df.describe())

接下来我们进行数据空行的删除，先运用函数size()记录清洗前数据的行数：

raw\_data\_count = df.iloc[:,0].size

print(raw\_data\_count)#打印行数

由输出结果我们可以知道原数据共有173行。

recent\_grads = pd.DataFrame.dropna(recent\_grads)#删除缺失值

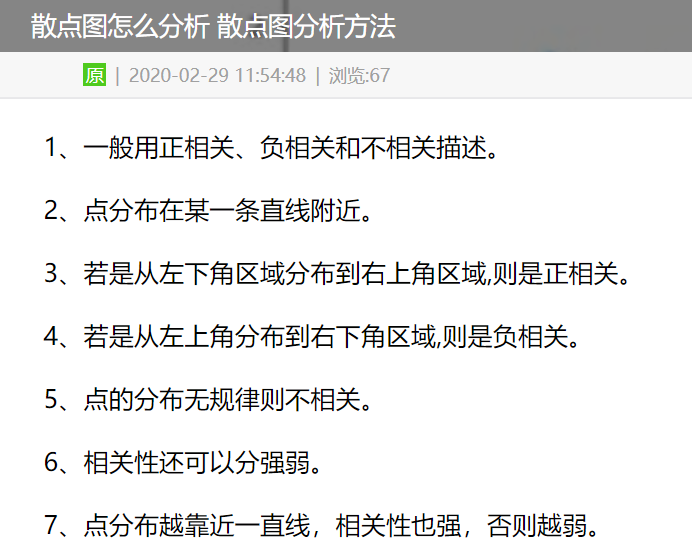
cleaned\_data\_count = recent\_grads.iloc[:,0].size

print(cleaned\_data\_count)#打印行数

删除缺失值后再次输出数据的行数，这次我们会发现，只有172行，所以共删除了1个含有缺失的行。

1. **利用散点图探索**

散点图分析的方法：

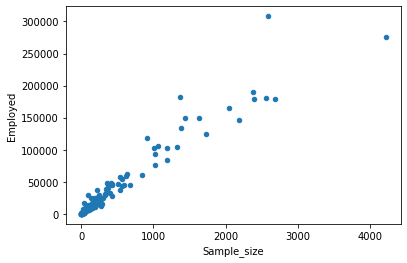
****

散点图是指在[回归分析](https://baike.baidu.com/item/%E5%9B%9E%E5%BD%92%E5%88%86%E6%9E%90/2625498" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)中，数据点在直角坐标系平面上的[分布图](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%9B%BE/10396618" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)，散点图表示因变量随[自变量](https://baike.baidu.com/item/%E8%87%AA%E5%8F%98%E9%87%8F/6895256" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)而[变化](https://baike.baidu.com/item/%E5%8F%98%E5%8C%96/33159" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)的大致趋势，据此可以选择合适的函数[对数](https://baike.baidu.com/item/%E5%AF%B9%E6%95%B0" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)据点进行[拟合](https://baike.baidu.com/item/%E6%8B%9F%E5%90%88/7505059" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)。

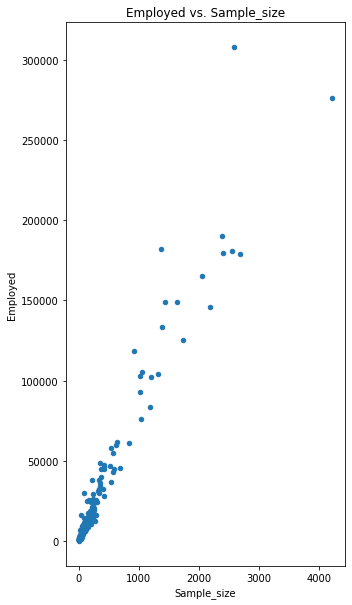
用两组数据构成多个坐标点，考察坐标点的分布，判断两变量之间是否存在某种关联或总结坐标点的分布模式。散点图将序列显示为一组点。值由点在[图表](https://baike.baidu.com/item/%E5%9B%BE%E8%A1%A8/1252386" \t "https://baike.baidu.com/item/%E6%95%A3%E7%82%B9%E5%9B%BE/_blank)中的位置表示。类别由图表中的不同标记表示。散点图通常用于比较跨类别的聚合数据。

首先根据题目提供的方法：

recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter')



recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter', title='Employed vs. Sample\_size', figsize=(5,10))#可以自行调节输出图片大小，并用title增加了图的题头



ax = recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter')

ax.set\_title('Employed vs. Sample\_size')

接下来在单独的jupyter笔记本单元中生成散点图，以探索以下关系：

Sample\_size 和 Median

recent\_grads.plot(x='Sample\_size', y='Median', kind='scatter',title='Sample\_size vs. Median')

Sample\_size 和 Unemployment\_rate

代码实现：recent\_grads.plot(x='Sample\_size',y='Unemployment\_rate',kind='scatter',title='Sample\_size vs. Unemployment\_rate')

Full\_time 和 Median

代码实现：recent\_grads.plot(x='Full\_time', y='Median', kind='scatter',title='Full\_time vs. Median')

ShareWomen 和 Unemployment\_rate

代码实现：recent\_grads.plot(x='ShareWomen',y='Unemployment\_rate',kind='scatter',title='ShareWomen vs. Unemployment\_rate')

Men 和 Median

代码实现：recent\_grads.plot(x='Men', y='Median', kind='scatter',title='Men vs. Median')

Women 和 Median

代码实现：recent\_grads.plot(x='Women', y='Median', kind='scatter',title='Women vs. Median')

根据散点图分析一下实际问题：

* + 热门专业的学生会赚更多钱吗？

代码实现：recent\_grads.plot(x='Total', y='Median', kind='scatter',title='Total vs. Median',figsize=(10,5))

* + 主修女性的专业学生赚钱更多吗？

代码实现：recent\_grads.plot(x='ShareWomen', y='Median', kind='scatter',title='ShareWomen vs. Median',figsize=(10,5))

* + 全职雇员人数与工资中位数之间有联系吗？

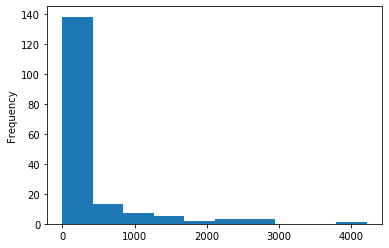
代码实现：

recent\_grads.plot(x='Sample\_size',y='Median',kind='scatter',title='Sample\_size vs. Median')

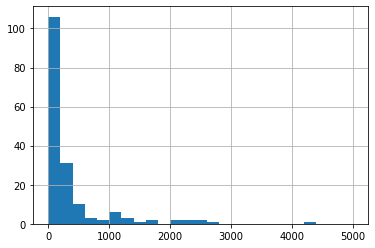
**3.探索列中值的分布**

根据提供的函数绘制直方图如下：

recent\_grads['Sample\_size'].plot(kind='hist')



recent\_grads['Sample\_size'].hist(bins=25, range=(0,5000))



以探索以下各列的分布：

Sample\_sizeMedian

代码实现 ：recent\_grads['Median'].hist(bins=10, range=(0,120000))

Employed

代码实现：recent\_grads['Employed'].hist(bins=10, range=(0,350000))

Full\_time

代码实现：recent\_grads['Full\_time'].hist(bins=10, range=(0,280000))

ShareWomen

代码实现：recent\_grads['ShareWomen'].hist(bins=10, range=(0,1))

Unemployment\_rate

代码实现：recent\_grads['Unemployment\_rate'].hist(bins=10, range=(0,0.18))

Men

代码实现：recent\_grads['Men'].hist(bins=10, range=(0,180000))

Women

代码实现：recent\_grads['Women'].hist(bins=10, range=(0,350000))

我们鼓励您在生成这些直方图时尝试使用不同的bins和range。

使用图表探索以下问题：

主要是男生的专业占百分之几？主要是女生的专业占百分之几？

recent\_grads['ShareWomen'].hist(bins=25, range=(0,1))

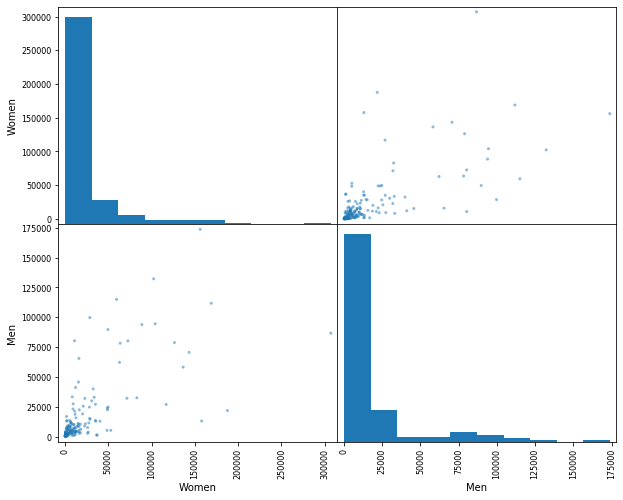
最常见的工资中位数在哪个范围？

代码实现：recent\_grads['Median'].hist(bins=10, range=(0,120000))

**4.散点矩阵图**

import pandas.plotting as pl

pl.scatter\_matrix(recent\_grads[['Women', 'Men']], figsize=(10,8))条形图

****

* 从pandas.plotting模块导入scatter\_matrix。
* 使用Sample\_size和Median列创建2 x 2散点图。

代码实现：pl.scatter\_matrix(recent\_grads[['Sample\_size', 'Median']], figsize=(10,8))

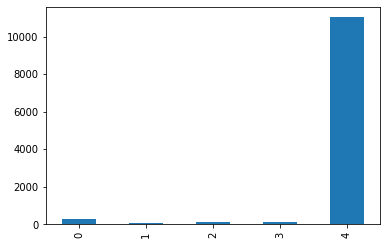
* 使用Sample\_size，Median和Unemployment\_rate列创建3 x 3散点图。

代码实现：pl.scatter\_matrix(recent\_grads[['Sample\_size','Median','Unemployment\_rate']], figsize=(10,8))

* 使用这些散点图，探索最后几个步骤中的问题。 您可能需要创建更多的散点图。

**5.条形图**

recent\_grads[:5]['Women'].plot(kind='bar')

****

recent\_grads[:5].plot.bar(x='Major', y='Women')

****

\*使用条形图来比较recent\_grads数据帧的前十行和后十行中的女性（ShareWomen）百分比。

recent\_grads[:10]['ShareWomen'].plot(kind='bar')#前十行女性百分比

recent\_grads[-10:]['ShareWomen'].plot(kind='bar')#后十行女性百分比

使用条形图来比较recent\_grads数据帧的前十行和后十行的失业率（Unemployment\_rate）。

recent\_grads[:10]['Unemployment\_rate'].plot(kind='bar')#数据帧前十行的失业率

recent\_grads[-10:]['Unemployment\_rate'].plot(kind='bar')#数据帧后十行的失业率

**五、结果分析：**

**5.1实现注释程序清单**

%matplotlib inline

import pandas as pd

import numpy as np

import matplotlib

import matplotlib.pyplot as plt#导入所需的各类包

recent\_grads = pd.read\_csv('recent-grads.csv')

df = pd.DataFrame(recent\_grads)

print("数据的第一行为:\n",df.iloc[0])

#输出CSV文件的前5行和最后5行，pandas默认的输出5行

print("\n数据的前5行为:\n",df.head())

print("\n数据的前5行为:\n",df.tail())

print("\n数据摘要统计:\n",df.describe())

raw\_data\_count = df.iloc[:,0].size

print(raw\_data\_count)#打印行数

recent\_grads = pd.DataFrame.dropna(recent\_grads)#删除缺失值

cleaned\_data\_count = recent\_grads.iloc[:,0].size

print(cleaned\_data\_count)#打印行数

###############################################################

recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter')

recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter', title='Employed vs. Sample\_size', figsize=(5,10))

ax = recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter')

ax.set\_title('Employed vs. Sample\_size')

recent\_grads.plot(x='Sample\_size', y='Median',kind='scatter',title='Sample\_size vs. Median')

recent\_grads.plot(x='Sample\_size',y='Unemployment\_rate', kind='scatter',title='Sample\_size vs. Unemployment\_rate')

recent\_grads.plot(x='Full\_time',y='Median',kind='scatter',title='Full\_time vs. Median')

recent\_grads.plot(x='ShareWomen',y='Unemployment\_rate', kind='scatter',title='ShareWomen vs. Unemployment\_rate')

recent\_grads.plot(x='Men', y='Median', kind='scatter',title='Men vs. Median')

recent\_grads.plot(x='Women', y='Median', kind='scatter',title='Women vs. Median')

recent\_grads.plot(x='Total',y='Median',kind='scatter',title='Totalvs.Median',figsize=(10,5))

recent\_grads.plot(x='ShareWomen', y='Median', kind='scatter',title='ShareWomen vs. Median',figsize=(10,5))

recent\_grads.plot(x='Sample\_size', y='Median', kind='scatter',title='Sample\_size vs. Median')

#################################################################

recent\_grads['Sample\_size'].plot(kind='hist')

recent\_grads['Sample\_size'].hist(bins=25, range=(0,5000))#range()可自行调节生成图的数据范围，bins将range改定范围的数据划分的个数

recent\_grads['Median'].hist(bins=10, range=(0,120000))

recent\_grads['Employed'].hist(bins=10, range=(0,350000))

recent\_grads['Full\_time'].hist(bins=10, range=(0,280000))

recent\_grads['ShareWomen'].hist(bins=10, range=(0,1))

recent\_grads['Unemployment\_rate'].hist(bins=10, range=(0,0.18))

recent\_grads['Men'].hist(bins=10, range=(0,180000))

recent\_grads['Women'].hist(bins=10, range=(0,350000))

recent\_grads['ShareWomen'].hist(bins=25, range=(0,1))

recent\_grads['Median'].hist(bins=10, range=(0,120000))

################################################################

import pandas.plotting as pl

pl.scatter\_matrix(recent\_grads[['Women', 'Men']], figsize=(10,8))

pl.scatter\_matrix(recent\_grads[['Sample\_size', 'Median']], figsize=(10,8))

pl.scatter\_matrix(recent\_grads[['Sample\_size','Median','Unemployment\_rate']], figsize=(10,8))

#################################################################

recent\_grads[:5]['Women'].plot(kind='bar')

recent\_grads[:5].plot.bar(x='Major', y='Women')

recent\_grads[:10]['ShareWomen'].plot(kind='bar')#前十行女性百分比

recent\_grads[-10:]['ShareWomen'].plot(kind='bar')#后十行女性百分比

recent\_grads[:10]['Unemployment\_rate'].plot(kind='bar')#数据帧前十行的失业率

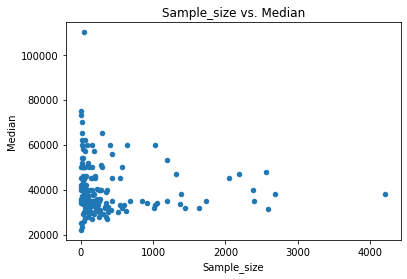
recent\_grads[-10:]['Unemployment\_rate'].plot(kind='bar')#数据帧后十行的失业率

**5.2结果分析**

由上述设计详述中的列中值的分布、散点图、散点矩阵图以及条状图的分析可知以下结论：

**Sample\_size 和 Median**

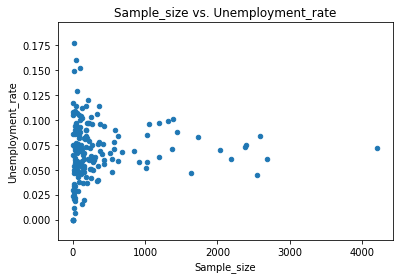
代码实现：recent\_grads.plot(x='Sample\_size', y='Median', kind='scatter',title='Sample\_size vs. Median')



由散点图的分析可知，样本容量越大样本的中位数越稳定，在样本容量趋于零的部分我们可以观察到，中位数的分布是没有规律的。

**Sample\_size 和 Unemployment\_rate**

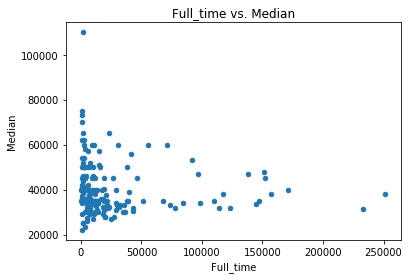
代码实现：recent\_grads.plot(x='Sample\_size',y='Unemployment\_rate',kind='scatter',title='Sample\_size vs. Unemployment\_rate')



由散点图的分析可知，样本容量越小样本的失业率越不具有代表性，在样本容量趋于零的部分我们可以观察到，失业率的分布是没有规律的。

**Full\_time 和 Median**

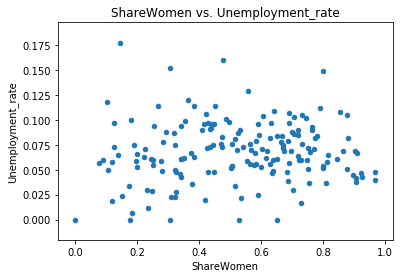
代码实现：recent\_grads.plot(x='Full\_time', y='Median', kind='scatter',title='Full\_time vs. Median')



由散点图的分析可知，工作时间在35小时以上的总人数越小样本的薪资中位数越不具有代表性，工作时间在35小时以上的总人数不断增大我们在图中可以发现该专业找到全年全职的工作的毕业生的薪资中位数集中在40000左右。

**ShareWomen 和 Unemployment\_rate**

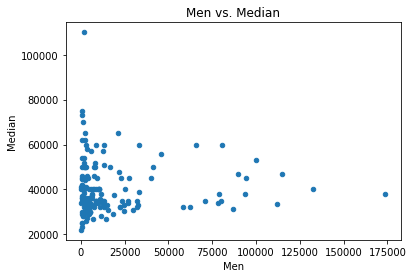
代码实现：recent\_grads.plot(x='ShareWomen',y='Unemployment\_rate',kind='scatter',title='ShareWomen vs. Unemployment\_rate')



由散点图的分析可知，女毕业生占比和失业率关系不明显。

Men 和 Median

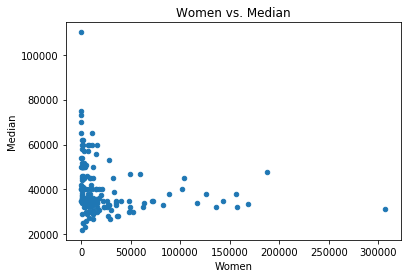
代码实现：recent\_grads.plot(x='Men', y='Median', kind='scatter',title='Men vs. Median')



由散点图的分析可知，男毕业生总人数越小样本的薪资中位数越不具有代表性，男毕业生总人数不断增大我们在图中可以发现该专业找到全年全职的工作的毕业生的薪资中位数集中在40000左右。

Women 和 Median

代码实现：recent\_grads.plot(x='Women', y='Median', kind='scatter',title='Women vs. Median')

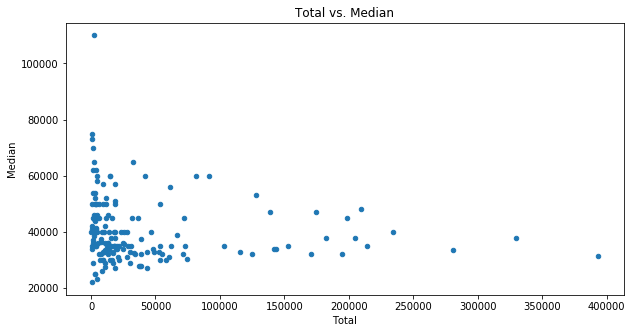


由散点图的分析可知，女毕业生总人数越小样本的薪资中位数越不具有代表性，女毕业生总人数不断增大我们在图中可以发现该专业找到全年全职的工作的毕业生的薪资中位数集中在40000偏下略少于男毕业生。

根据散点图分析一下实际问题：

* + 热门专业的学生会赚更多钱吗？

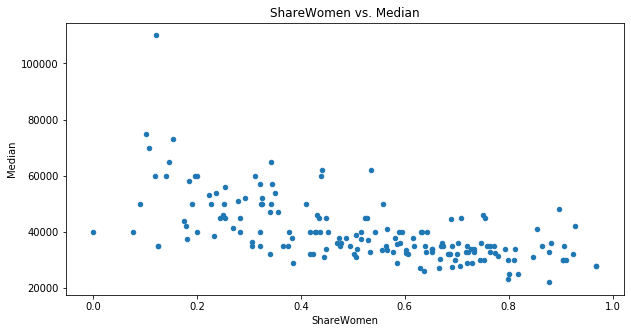
代码实现：recent\_grads.plot(x='Total', y='Median', kind='scatter',title='Total vs. Median',figsize=(10,5))



该散点图并没有体现热门专业和薪资成正比，由图可知薪资中位数的最大值出现在专业人数较少的一端（我们认为选择该专业人数与是否热门成正比）。

* + 主修女性的专业学生赚钱更多吗？

代码实现：recent\_grads.plot(x='ShareWomen', y='Median', kind='scatter',title='ShareWomen vs. Median',figsize=(10,5))

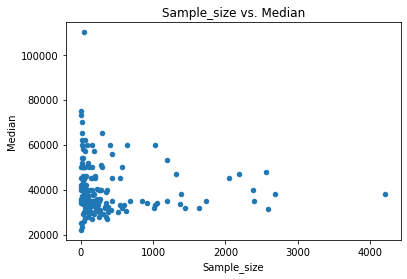


由散点图的分析可知，女毕业生占比越高薪资中位数越低，但下降速率较缓慢，总体呈递减趋势。

* + 全职雇员人数与工资中位数之间有联系吗？

代码实现：

recent\_grads.plot(x='Sample\_size',y='Median',kind='scatter',title='Sample\_size vs. Median')



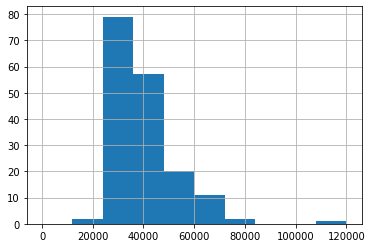
由散点图的分析可知，全职员工样本容量越大样本的中位数越稳定，在全职员工样本容量趋于零的部分我们可以观察到，中位数的分布是没有规律的。

**3.探索列中值的分布**

以探索以下各列的分布：

Median

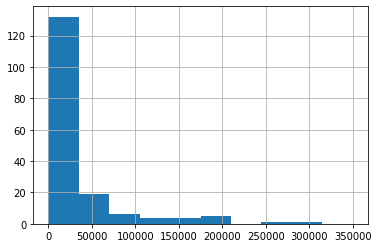
代码实现 ：recent\_grads['Median'].hist(bins=10, range=(0,120000))



由Median的频率分布直方图可知，该专业找到全年全职的工作的毕业生的薪资中位数集中在25000~48000元之间。

Employed

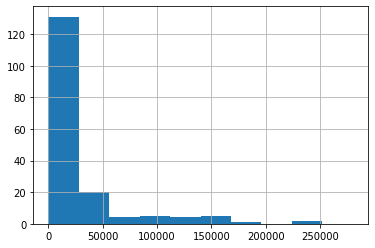
代码实现：recent\_grads['Employed'].hist(bins=10, range=(0,350000))



由Employed的频率分布直方图可知，毕业生就业人数集中在0~30000人之间。

Full\_time

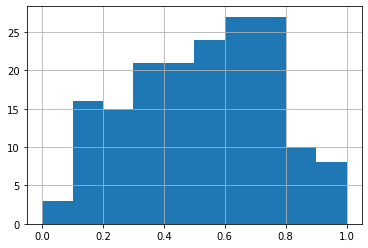
代码实现：recent\_grads['Full\_time'].hist(bins=10, range=(0,280000))



由Full\_time的频率分布直方图可知，工作时长大于35h的人数集中在0~25000人之间。

ShareWomen

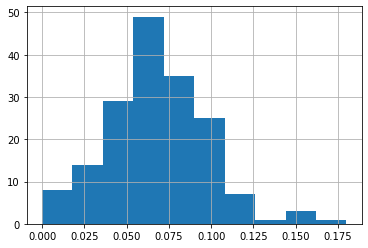
代码实现：recent\_grads['ShareWomen'].hist(bins=10, range=(0,1))



由ShareWomen的频率分布直方图可知，女毕业生占比偏向0.7但分布较为平均，两端较少。

Unemployment\_rate

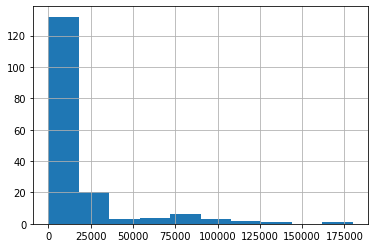
代码实现：recent\_grads['Unemployment\_rate'].hist(bins=10, range=(0,0.18))



由Unemployment\_rate的频率分布直方图可知，失业率集中在0.05~0.07左右，可以认为毕业生失业率较低。

Men

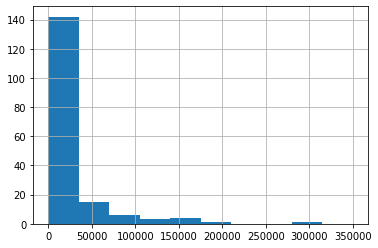
代码实现：recent\_grads['Men'].hist(bins=10, range=(0,180000))



由Men的频率分布直方图可知，毕业生中男生人数集中在0~20000左右。

Women

代码实现：recent\_grads['Women'].hist(bins=10, range=(0,350000))



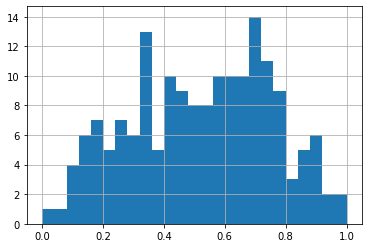
由Women的频率分布直方图可知，毕业生中女生人数集中在0~20000左右。

我们鼓励您在生成这些直方图时尝试使用不同的bins和range。

使用图表探索以下问题：

主要是男生的专业占百分之几？主要是女生的专业占百分之几？

recent\_grads['ShareWomen'].hist(bins=25, range=(0,1))

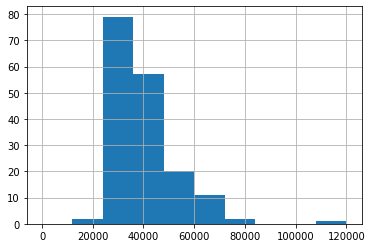


将bins、

设置为25时进行了细分，可以发现主要是女生的专业占70%

最常见的工资中位数在哪个范围？

代码实现：recent\_grads['Median'].hist(bins=10, range=(0,120000))

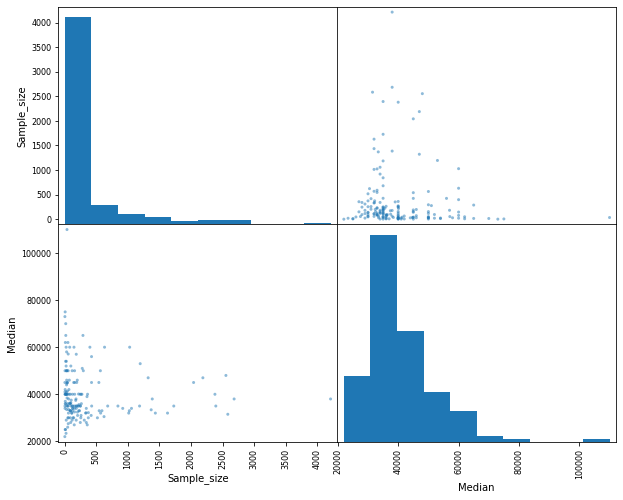


最常见的工资中位数在22000~35000之间。

**4.散点矩阵图**

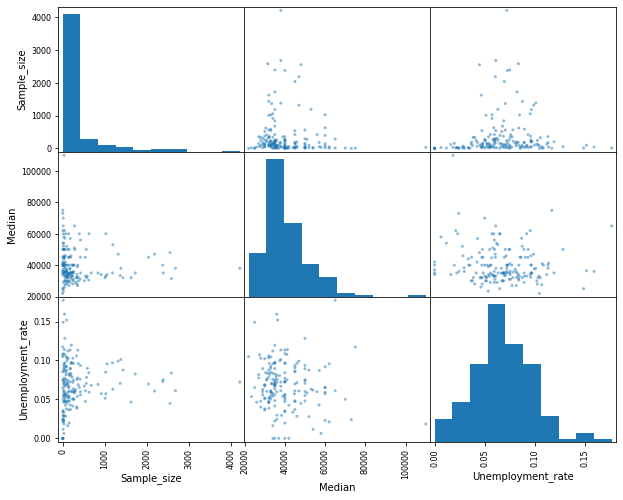
* 从pandas.plotting模块导入scatter\_matrix。
* 使用Sample\_size和Median列创建2 x 2散点图。

代码实现：pl.scatter\_matrix(recent\_grads[['Sample\_size', 'Median']], figsize=(10,8))



* 使用Sample\_size，Median和Unemployment\_rate列创建3 x 3散点图。

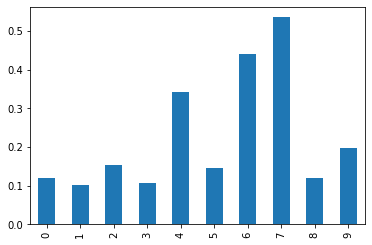
代码实现：pl.scatter\_matrix(recent\_grads[['Sample\_size','Median','Unemployment\_rate']], figsize=(10,8))



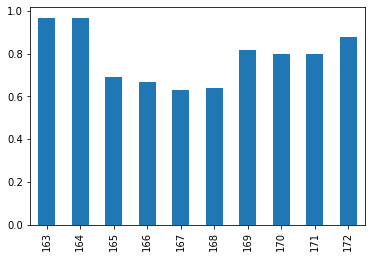
**5.条形图**

\*使用条形图来比较recent\_grads数据帧的前十行和后十行中的女性（ShareWomen）百分比。

recent\_grads[:10]['ShareWomen'].plot(kind='bar')#前十行女性百分比



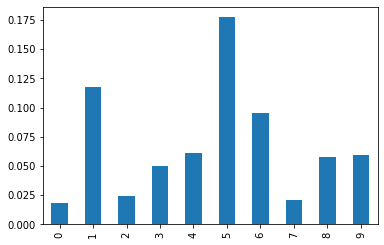
recent\_grads[-10:]['ShareWomen'].plot(kind='bar')#后十行女性百分比



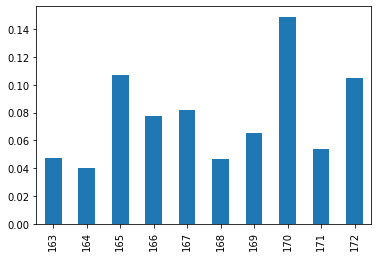
可以明显看出后10行女性占比较为集中，平均在0.7以上。

使用条形图来比较recent\_grads数据帧的前十行和后十行的失业率（Unemployment\_rate）。

recent\_grads[:10]['Unemployment\_rate'].plot(kind='bar')#数据帧前十行的失业率



recent\_grads[-10:]['Unemployment\_rate'].plot(kind='bar')#数据帧后十行的失业率



后十行的平均失业率高于前十行的平均失业率。

1. **用户手册**

若用户需要对类似问题进行分析时，只需将先进行数据清洗，并将以上图表的数据进行修改即可。分析散点图时可以参照在第四节提到的散点图分析方法。

**七、调试及总结**

1.%matplotlib inline

在除jupyter notebook之外python编译器上运行会报错，%matplotlib具体作用是当你调用matplotlib.pyplot的绘图函数plot()进行绘图的时候，或者生成一个figure画布的时候，可以直接在你的python console里面生成图像。而在spyder或者pycharm实际运行代码的时候，可以直接注释掉这一句。

2.recent\_grads.plot(x='Sample\_size', y='Employed', kind='scatter', title='Employed vs. Sample\_size', figsize=(5,10))

在该绘图语句中我们可以将坐标标注、图注、以及图比例大小进行自行调整。

1. recent\_grads['Median'].hist(bins=10, range=(0,120000))

在该绘图语句中我们可以将bins划分个数、range数据范围进行自行设定。

通过本次实验，我对多重绘图（multiple plots）、条形图（bar plot）、散点图（scatter plot）、直方图（histogram）进行了更深入的了解，并且进行了实践操作，所以对知识的掌握动手环节十分重要。