In [1]:
```python
import pandas as pd
```

In [2]:
```python
teachST = pd.read_csv("public-school-teacher-salaries.csv")
```

In [3]:
```python
teachST.head(10)
```

Out[3]:

	State	1990	1991	1992	1993	1994	1995	1996	1997	1998	...	2009	2010	2011	2012	2013	2014
0	United States	31,367	33,061	34,015	34,974	35,705	36,757	37,585	38,413	39,346	...	54,354	55,225	55,586	55,522	56,065	56,648
1	Alabama	24,828	26,874	26,971	26,953	28,705	31,144	31,313	32,549	32,818	...	46,879	47,571	47,803	48,003	47,949	48,720
2	Alaska	43,153	43,427	44,661	46,701	47,512	47,951	49,620	50,647	47,601	...	58,395	60,732	62,918	62,425	65,468	65,891
3	Arizona	29,402	30,773	31,176	31,352	31,800	32,574	32,483	33,685	34,411	...	46,358	46,952	46,637	45,193	45,264	45,335
4	Arkansas	22,352	23,611	27,070	27,433	28,098	28,934	29,533	30,578	31,837	...	45,738	46,045	45,998	46,314	46,631	47,319
5	California	37,998	39,598	39,922	40,035	40,264	41,078	42,259	42,992	44,585	...	66,995	67,932	67,871	69,755	69,435	71,396
6	Colorado	30,758	31,819	33,072	33,541	33,826	34,571	35,364	36,271	37,240	...	48,485	49,181	49,228	49,049	49,844	49,615
7	Connecticut	40,461	43,808	46,971	48,343	49,769	50,045	50,254	50,426	50,730	...	65,830	68,096	69,165	69,465	69,397	70,583
8	Delaware	33,377	35,246	34,548	36,217	37,469	39,076	40,533	41,436	42,439	...	56,667	57,080	57,934	58,800	59,679	59,305
9	District of Columbia	38,402	39,497	38,798	38,702	42,543	41,835	43,700	44,834	47,076	...	62,557	64,548	66,601	68,720	70,906	73,162

10 rows × 30 columns

In [4]:
```python
teachAZ = pd.read_csv("AZ School District.csv")
teachAZ.head()
```

Out[4]:

	District name	Fiscal year	County	Operational efficiency peer group	Transportation efficiency peer group	Student achievement peer group	Legislative district(s)	District size	Location	Students attending	...	Spending exceeded operating/capital budgets (2015 through 2017)
0	Agua Fria Union High School District	2017	Maricopa	3	T-4	8	4, 13, 19, and 29	Medium-large	Suburb	7,721	...	No overspending
1	Aguila Elementary School District	2017	Maricopa	11	T-11	19	13	Very small	Rural	148	...	Operating and capital
2	Ajo Unified School District	2017	Pima	6	T-9	5	4	Small	Town	419	...	No overspending
3	Alhambra Elementary School District	2017	Maricopa	7	T-1	15	29 and 30	Large	City	12,524	...	No overspending
4	Alpine Elementary School District	2017	Apache	11	T-11	18	7	Very small	Rural	57	...	No overspending

5 rows × 134 columns

In [5]:
```python
print(teachAZ.columns.tolist())
```

```
['District name', 'Fiscal year', 'County', 'Operational efficiency peer group', 'Transportation effic
iency peer group', 'Student achievement peer group', 'Legislative district(s)', 'District size', 'Loc
ation', 'Students attending', 'Number of schools', 'Instruction', 'Instruction (peer average)', 'Admi
nistration', 'Plant operations', 'Food service', 'Transportation', 'Student support', 'Instruction su
pport', 'Administration cost per pupil', 'Peer average: Administration cost per pupil', 'Administrati
on cost per pupil relative to peer average', 'Students per administrative position', 'Peer average: S
tudents per administrative position', 'Overall administration measure relative to peer average', 'Pla
nt operations cost per square foot', 'Peer average: Plant operations cost per square foot', 'Plant op
erations cost per square foot relative to peer average', 'Square footage per student', 'Peer average:
Square footage per student', 'Square footage per student relative to peer average', 'Overall plant op
erations measures relative to peer average', 'Food service cost per meal', 'Peer average: Food servic
e cost per meal', 'Food service cost per meal relative to peer average', 'Overall food service measur
e relative to peer average', 'Transportation cost per mile', 'Peer average: Transportation cost per m
ile', 'Transportation cost per mile relative to peer average', 'Transportation cost per rider', 'Peer
average: Transportation cost per rider', 'Transportation cost per rider relative to peer average', 'O
verall transportation measures relative to peer average', 'Instruction.1', 'Peer average: Instructio
n', 'Administration.1', 'Peer average: Administration', 'Plant operations.1', 'Peer average: Plant op
erations', 'Food service.1', 'Peer average: Food service', 'Transportation.1', 'Peer average: Transpo
rtation', 'Student support.1', 'Peer average: Student support', 'Instruction support.1', 'Peer averag
e: Instruction support', 'Total operational spending per pupil', 'Peer average: Total operational spe
nding', 'Land and buildings', 'Peer average: Land and buildings', 'Equipment', 'Peer average: Equipme
nt', 'Interest', 'Peer average: Interest', 'Other', 'Peer average: Other', 'Total nonoperational spen
ding per pupil', 'Peer average total nonoperational spending', 'Total per pupil spending', 'Peer aver
age: Total per pupil spending', 'Math', 'Peer group: Math', 'English Language Arts', 'Peer group: Eng
lish Language Arts', 'Science', 'Peer group: Science', 'Attendance rate', 'Peer average: Attendance r
ate', 'Graduation rate (FY 2016)', 'Peer average: Graduation rate (FY 2016)', 'Poverty rate (FY 201
6)', 'Peer average: Poverty rate (FY 2016)', 'Special education population', 'Peer average: Special e
ducation population', 'Students per teacher', 'Peer average: Students per teacher', 'Average teacher
salary', 'Peer average: Average teacher salary', 'Average teacher salary amount from Prop 301', 'Peer
average: Average teacher salary amount from Prop 301', 'Average years of teacher experience', 'Peer a
verage: Average years of teacher experience', 'Percent of teachers in first 3 years', 'Peer average:
Percent of teachers in first 3 years', 'Federal revenues', 'Peer average: Federal revenues', 'State r
evenues', 'Peer average: State revenues', 'Local revenues', 'Peer average: Local revenues', 'Total re
venues per pupil', 'Peer average: Total revenues per pupil', 'Equalization formula funding revenues',
'Peer average: Equalization formula funding revenues', 'Equalization formula funding amount from Prop
123', 'Peer average: Equalization formula funding amount from Prop 123', 'Prop 123 additional fundin
g', 'Peer average: Prop 123 additional funding', 'Grant revenues', 'Peer average: Grant revenues', 'D
onation and tax credit revenues', 'Peer average: Donation and tax credit revenues', 'Desegregation re
venues', 'Number of peers receiving desegregation revenues', 'Small school adjustment revenues', 'Num
ber of peers receiving small school adjustment revenues', 'Federal impact aid revenues', 'Number of p
eers receiving federal impact aid revenues', 'Voter-approved levy increase revenues', 'Number of peer
s receiving voter-approved levy increase revenues', 'Overall financial stress level', 'Change in numb
er of district students (2015 through 2017)', 'Financial stress level for change in number of distric
t students', 'Spending exceeded operating/capital budgets (2015 through 2017)', 'Financial stress lev
el for spending exceeded operating/capital budgets', 'Spending increase election results  (2015 throu
gh 2017)', 'Financial stress level for spending increase election results', 'Operating reserve percen
tage, Trend (2015 through 2017)', 'Financial stress level for operating reserve percentage, Trend',
'Years of capital reserve held (2015 through 2017)', 'Financial stress level for years of capital res
erve held', 'Current financial and internal control status (2015 through 2017)', 'Financial stress le
vel for current financial and internal control status']
```

In [6]:
```python
len(teachAZ.columns.tolist())
```

Out[6]: 134

In [7]:
```python
keep = ['District name', 'County', 'Administration cost per pupil', 'Average teacher salary']
t_salariesAZ = teachAZ[keep]
t_salariesAZ.head()
```

Out[7]:

	District name	County	Administration cost per pupil	Average teacher salary
0	Agua Fria Union High School District	Maricopa	$710	$40,425
1	Aguila Elementary School District	Maricopa	$1,375	$47,114
2	Ajo Unified School District	Pima	$1,306	$44,386
3	Alhambra Elementary School District	Maricopa	$777	$58,362
4	Alpine Elementary School District	Apache	$3,222	$60,612

In [8]:
```python
incomesAZ = pd.read_csv("CAINC1_AZ_1969_2019.csv")
incomesAZ.head()
```

Out[8]:

	GeoFIPS	GeoName	Region	TableName	LineCode	IndustryClassification	Description	Unit	1969	1970	...	2010
0	"04000"	Arizona	6.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	6327071	7216133	...	215523641.0
1	"04000"	Arizona	6.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	1737000	1794912	...	6407172.0
2	"04000"	Arizona	6.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	3643	4020	...	33638.0
3	"04001"	Apache, AZ	6.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	80748	89332	...	1939941.0
4	"04001"	Apache, AZ	6.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	34200	32883	...	71829.0

5 rows × 59 columns

We need to remove rows to get just personal income as well as clean up county name to remove AZ

In [9]:
```python
incomesAZ.tail()
```

Out[9]:

	GeoFIPS	GeoName	Region	TableName	LineCode	IndustryClassification	Description	Unit	1969	1970	...	2010	2011
47	"04027"	Yuma, AZ*	6.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	3503	3828	...	27236.0	27964.0
48	Note: See the included footnote file.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
49	CAINC1: Personal Income Summary: Personal Inco...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
50	Last updated: November 17, 2020--new statisti...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
51	Source: U.S. Department of Commerce / Bureau o...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

5 rows × 59 columns

In [10]:
```python
incomesAZ.drop(incomesAZ.tail(4).index, inplace = True)
incomesAZ.tail()
```

Out[10]:

	GeoFIPS	GeoName	Region	TableName	LineCode	IndustryClassification	Description	Unit	1969	1970	...	2010
43	"04025"	Yavapai, AZ	6.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	35800	37570	...	210983.0
44	"04025"	Yavapai, AZ	6.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	3456	3843	...	28911.0
45	"04027"	Yuma, AZ*	6.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	209506	235094	...	5368937.0
46	"04027"	Yuma, AZ*	6.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	59800	61415	...	197127.0
47	"04027"	Yuma, AZ*	6.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	3503	3828	...	27236.0

5 rows × 59 columns

In [11]:
```python
t_salariesNJ = pd.read_csv("nj_teacher_salaries_2018.csv")
t_salariesNJ.head()
```

Out[11]:

	CONAME	DIST	DISTNAME	salary_2018	salary_2017	salary_2016
0	Bergen	3710	Northern Valley Regional	$108,867	$105,650	$95,418
1	Middlesex	1290	Edison Township	$96,650	$92,432	$92,194
2	Atlantic	110	Atlantic City	$96,488	$88,318	$94,135
3	Passaic	3995	Passaic County Vocational	$95,549	$95,303	$89,522
4	Atlantic	3020	Margate City	$93,275	$91,045	$89,766

In [12]:
```python
t_salariesNJ.rename(columns = {'CONAME':'COUNTY', 'DISTNAME':'DISTRICT'},inplace = True)
t_salariesNJ.head()
```

Out[12]:

	COUNTY	DIST	DISTRICT	salary_2018	salary_2017	salary_2016
0	Bergen	3710	Northern Valley Regional	$108,867	$105,650	$95,418
1	Middlesex	1290	Edison Township	$96,650	$92,432	$92,194
2	Atlantic	110	Atlantic City	$96,488	$88,318	$94,135
3	Passaic	3995	Passaic County Vocational	$95,549	$95,303	$89,522
4	Atlantic	3020	Margate City	$93,275	$91,045	$89,766

In [13]:
```python
incomesNJ = pd.read_csv("CAINC1_NJ_1969_2019.csv")
incomesNJ.head()
```

Out[13]:

	GeoFIPS	GeoName	Region	TableName	LineCode	IndustryClassification	Description	Unit	1969	1970	...
0	"34000"	New Jersey	2.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	32504175.0	35294519.0	...
1	"34000"	New Jersey	2.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	7095000.0	7190282.0	...
2	"34000"	New Jersey	2.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	4581.0	4909.0	...
3	"34001"	Atlantic, NJ	2.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	694634.0	782866.0	...
4	"34001"	Atlantic, NJ	2.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	174603.0	175908.0	...

5 rows × 59 columns

In [14]:
```python
incomesNJ.drop(incomesNJ.tail(4).index, inplace = True)
incomesNJ.tail()
```

Out[14]:

	GeoFIPS	GeoName	Region	TableName	LineCode	IndustryClassification	Description	Unit	1969	1970	...	2010
61	"34039"	Union, NJ	2.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	538294.0	543400.0	...	537369.0
62	"34039"	Union, NJ	2.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	5240.0	5602.0	...	51042.0
63	"34041"	Warren, NJ	2.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	283277.0	309939.0	...	4806788.0
64	"34041"	Warren, NJ	2.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	72760.0	74130.0	...	108576.0
65	"34041"	Warren, NJ	2.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	3893.0	4181.0	...	44271.0

5 rows × 59 columns

In [15]:
```python
sorted(incomesNJ['GeoName'].unique())
```

Out[15]:
```
['Atlantic, NJ',
 'Bergen, NJ',
 'Burlington, NJ',
 'Camden, NJ',
 'Cape May, NJ',
 'Cumberland, NJ',
 'Essex, NJ',
 'Gloucester, NJ',
 'Hudson, NJ',
 'Hunterdon, NJ',
 'Mercer, NJ',
 'Middlesex, NJ',
 'Monmouth, NJ',
 'Morris, NJ',
 'New Jersey',
 'Ocean, NJ',
 'Passaic, NJ',
 'Salem, NJ',
 'Somerset, NJ',
 'Sussex, NJ',
 'Union, NJ',
 'Warren, NJ']
```

In [16]:
```python
sorted(t_salariesNJ['COUNTY'].unique())
```

Out[16]:
```
['Atlantic',
 'Bergen',
 'Burlington',
 'Camden',
 'Cape May',
 'Cumberland',
 'Essex',
 'Gloucester',
 'Hudson',
 'Hunterdon',
 'Mercer',
 'Middlesex',
 'Monmouth',
 'Morris',
 'Ocean',
 'Passaic',
 'Salem',
 'Somerset',
 'Sussex',
 'Union',
 'Warren']
```

In [17]:
```python
sorted(incomesAZ['GeoName'].unique())
```

Out[17]:
```
['Apache, AZ',
 'Arizona',
 'Cochise, AZ',
 'Coconino, AZ',
 'Gila, AZ',
 'Graham, AZ',
 'Greenlee, AZ',
 'La Paz, AZ*',
 'Maricopa, AZ',
 'Mohave, AZ',
 'Navajo, AZ',
 'Pima, AZ',
 'Pinal, AZ',
 'Santa Cruz, AZ',
 'Yavapai, AZ',
 'Yuma, AZ*']
```

In [18]:
```python
sorted(t_salariesAZ['County'].unique())
```

Out[18]:
```
['Apache',
 'Cochise',
 'Coconino',
 'Gila',
 'Graham',
 'Greenlee',
 'La Paz',
 'Maricopa',
 'Mohave',
 'Navajo',
 'Pima',
 'Pinal',
 'Santa Cruz',
 'Yavapai',
 'Yuma']
```

Next we clean our data:

1. Fixing the naming convention for counties

In [19]:
```python
#in remove rows for Arizona/New Jersey in general
incomesAZ.drop(incomesAZ.head(3).index, inplace = True)
incomesNJ.drop(incomesNJ.head(3).index, inplace = True)
```

In [20]:
```python
incomesAZ['CountyList'] = incomesAZ['GeoName'].str.split(',')
incomesAZ.loc[:, 'County'] = incomesAZ.CountyList.map(lambda x: x[0])
incomesAZ.head()
```

Out[20]:

	GeoFIPS	GeoName	Region	TableName	LineCode	IndustryClassification	Description	Unit	1969	1970	...	2012
3	"04001"	Apache, AZ	6.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	80748	89332	...	1985377.0
4	"04001"	Apache, AZ	6.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	34200	32883	...	72229.0
5	"04001"	Apache, AZ	6.0	CAINC1	3.0	...	Per capita personal income (dollars) 2/	Dollars	2361	2717	...	27487.0
6	"04003"	Cochise, AZ	6.0	CAINC1	1.0	...	Personal income (thousands of dollars)	Thousands of dollars	235225	258346	...	4535292.0
7	"04003"	Cochise, AZ	6.0	CAINC1	2.0	...	Population (persons) 1/	Number of persons	60500	62770	...	132017.0

5 rows × 61 columns

In [21]:
```python
incomesNJ['CountyList'] = incomesNJ['GeoName'].str.split(',')
incomesNJ.loc[:, 'County'] = incomesNJ.CountyList.map(lambda x: x[0])
```

Dealing with OES Data

In [22]:
```python
import pandas as pd

odf97 = pd.read_excel('OES/national_1997_dl.xls')
```

In [23]:
```python
codes = ['13-1071', '13-2010','13-2081', '21-1022', '33-2010', '33-3051', '43-5052', '25-2031']
```

In [24]:
```python
odf01 = pd.read_excel('OES/oes01nat/national_2001_dl.xls')
odf01 = odf01.drop(columns=['annual', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_wpct25', 'h_wpct7
5', 'a_wpct25', 'a_wpct75'])

odf01 = odf01.loc[odf01['occ_code'].isin(codes)]
odf01
```

Out[24]:

	occ_code	occ_title	group	tot_emp	h_wpct10	h_median	h_wpct90	a_wpct10	a_median	a_wpct90	year
41	13-1071	Employment, Recruitment, and Placement Special...	NaN	173940	11.3	18.27	34.95	23500	38010	72690	2001
149	21-1022	Medical and Public Health Social Workers	NaN	103490	11.35	17.5	26.61	23600	36410	55350	2001
207	25-2031	Secondary School Teachers, Except Special and ...	NaN	980730	*	*	*	27980	43280	67940	2001
326	33-3051	Police and Sheriff's Patrol Officers	NaN	599550	11.78	19.7	29.74	24490	40970	61870	2001
444	43-5052	Postal Service Mail Carriers	NaN	355120	14.62	18.61	21.16	30410	38700	44010	2001

In [25]:
```python
odf02 = pd.read_excel('OES/oes02nat/national_2002_dl.xls')
odf02 = odf02.drop(columns=['annual', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_wpct25', 'h_wpct7
5', 'a_wpct25', 'a_wpct75'])
odf02 = odf02.rename(columns={'occ_titl':'occ_title'})
odf02
```

Out[25]:

	occ_code	occ_title	group	tot_emp	h_wpct10	h_median	h_wpct90	a_wpct10	a_median	a_wpct90
0	00-0000	All Occupations	major	127523760	6.95	13.31	31.2	14450	27690	64900
1	11-0000	Management Occupations	major	7092460	15.31	32.27	69.53	31850	67120	144620
2	11-1011	Chief Executives	NaN	452400	24.83	60.7	#	51650	126260	#
3	11-1021	General and Operations Managers	NaN	1998350	15.72	32.8	#	32700	68210	#
4	11-1031	Legislators	NaN	64650	5.83	7.32	33.36	12130	15220	69380
...	...	...	...	...	...	...	...	...	...	...
729	53-7072	Pump Operators, Except Wellhead Pumpers	NaN	12360	9.47	17.53	28.17	19700	36470	58590
730	53-7073	Wellhead Pumpers	NaN	10280	9.33	16.24	23.74	19420	33770	49370
731	53-7081	Refuse and Recyclable Material Collectors	NaN	132290	6.28	11.6	21.07	13070	24130	43820
732	53-7111	Shuttle Car Operators	NaN	3070	13.09	18.44	22.17	27230	38360	46120
733	53-7121	Tank Car, Truck, and Ship Loaders	NaN	16960	9.52	15.63	27.38	19810	32500	56940

734 rows × 10 columns

In [26]:
```python
odf03 = pd.read_excel('OES/oesm03nat/national_may2003_dl.xls')
odf03 = odf03.drop(columns=['annual', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25', 'h_pct75'
, 'a_pct25', 'a_pct75'])
odf03.insert(0, 'year', 2003)
odf03
```

Out[26]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2003	00-0000	All Occupations	NaN	127567910	7.04	13.53	31.97	14640	28140	66500
1	2003	11-0000	Management occupations	major	6653480	16.15	34.07	#	33590	70870	#
2	2003	11-1011	Chief executives	NaN	389880	26.79	64.78	#	55720	134740	#
3	2003	11-1021	General and operations managers	NaN	1892060	16.75	35.00	#	34850	72800	#
4	2003	11-1031	Legislators	NaN	65280	5.87	7.90	34.92	12220	16440	72630
...	...	...	...	...	...	...	...	...	...	...	...
729	2003	53-7072	Pump operators, except wellhead pumpers	NaN	12260	9.68	18.00	27.85	20130	37430	57920
730	2003	53-7073	Wellhead pumpers	NaN	8560	8.88	14.89	23.99	18470	30970	49900
731	2003	53-7081	Refuse and recyclable material collectors	NaN	138480	6.31	11.56	21.05	13130	24040	43790
732	2003	53-7111	Shuttle car operators	NaN	3040	14.10	18.79	22.22	29330	39080	46210
733	2003	53-7121	Tank car, truck, and ship loaders	NaN	16210	9.75	15.90	27.20	20280	33060	56580

734 rows × 11 columns

In [27]:
```python
odf04 = pd.read_excel('OES/oesm04nat/national_may2004_dl.xls')
odf04 = odf04.drop(columns=['annual', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25', 'h_pct75'
, 'a_pct25', 'a_pct75'])
odf04.insert(0, 'year', 2004)
odf04
```

Out[27]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90	hourly
0	2004	00-0000	All Occupations	NaN	128127360	7.16	13.83	32.94	14880	28770	68510	NaN
1	2004	11-0000	Management occupations	major	6200940	17.19	35.77	#	35760	74390	#	NaN
2	2004	11-1011	Chief executives	NaN	346590	27.76	67.47	#	57740	140350	#	NaN
3	2004	11-1021	General and operations managers	NaN	1752910	18.15	37.22	#	37760	77420	#	NaN
4	2004	11-1031	Legislators	NaN	63440	*	*	*	11970	16190	71010	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
818	2004	53-7073	Wellhead pumpers	NaN	10040	9.12	16.31	24.21	18980	33930	50350	NaN
819	2004	53-7081	Refuse and recyclable material collectors	NaN	139920	6.58	12.38	22.40	13680	25760	46590	NaN
820	2004	53-7111	Shuttle car operators	NaN	3000	13.95	18.08	21.70	29030	37610	45130	NaN
821	2004	53-7121	Tank car, truck, and ship loaders	NaN	16530	9.95	15.59	24.92	20690	32440	51840	NaN
822	2004	53-7199	Material moving workers, all other	NaN	57390	8.23	13.87	25.16	17110	28850	52320	NaN

823 rows × 12 columns

In [28]:
```python
odf05 = pd.read_excel('OES/oesm05nat/national_may2005_dl.xls')
odf05 = odf05.drop(columns=['annual', 'hourly', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25',
 'h_pct75', 'a_pct25', 'a_pct75'])
odf05.insert(0, 'year', 2005)
odf05
```

Out[28]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2005	00-0000	All Occupations	NaN	130307840	7.26	14.15	33.74	15110	29430	70180
1	2005	11-0000	Management occupations	major	5960560	18.17	37.32	#	37800	77630	#
2	2005	11-1011	Chief executives	NaN	321300	28.84	68.48	#	59990	142440	#
3	2005	11-1021	General and operations managers	NaN	1663810	19.26	39.17	#	40060	81480	#
4	2005	11-1031	Legislators	NaN	61060	*	*	*	11820	15740	72700
...	...	...	...	...	...	...	...	...	...	...	...
817	2005	53-7073	Wellhead pumpers	NaN	10190	9.96	18.12	25.90	20720	37690	53880
818	2005	53-7081	Refuse and recyclable material collectors	NaN	133930	7.63	13.68	23.33	15880	28460	48520
819	2005	53-7111	Shuttle car operators	NaN	3100	14.43	18.42	22.14	30010	38310	46050
820	2005	53-7121	Tank car, truck, and ship loaders	NaN	15950	9.36	15.06	26.26	19480	31310	54630
821	2005	53-7199	Material moving workers, all other	NaN	52970	8.09	14.53	25.36	16830	30220	52740

822 rows × 11 columns

In [29]:
```python
odf06 = pd.read_excel('OES/oesm06nat/national_may2006_dl.xls')
odf06 = odf06.drop(columns=['annual', 'hourly', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25',
 'h_pct75', 'a_pct25', 'a_pct75'])
odf06.insert(0, 'year', 2006)
odf06
```

Out[29]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2006	00-0000	All Occupations	NaN	132604980	7.45	14.61	35.08	15500	30400	72960
1	2006	11-0000	Management occupations	major	5892900	18.93	38.93	#	39380	80980	#
2	2006	11-1011	Chief executives	NaN	299520	29.79	#	#	61970	#	#
3	2006	11-1021	General and operations managers	NaN	1663280	20.29	40.97	#	42210	85230	#
4	2006	11-1031	Legislators	NaN	62020	*	*	*	12190	15660	75270
...	...	...	...	...	...	...	...	...	...	...	...
817	2006	53-7073	Wellhead pumpers	NaN	13280	10.63	17.38	25.94	22120	36150	53950
818	2006	53-7081	Refuse and recyclable material collectors	NaN	125770	7.98	13.93	24.42	16600	28970	50790
819	2006	53-7111	Shuttle car operators	NaN	2860	14.91	18.78	23.23	31020	39060	48320
820	2006	53-7121	Tank car, truck, and ship loaders	NaN	15360	9.55	15.37	25.64	19850	31970	53330
821	2006	53-7199	Material moving workers, all other	NaN	52120	8.13	14.55	25.32	16920	30270	52670

822 rows × 11 columns

In [30]:
```python
odf07 = pd.read_excel('OES/oesm07nat/national_May2007_dl.xls')
odf07 = odf07.drop(columns=['annual', 'hourly', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25',
 'h_pct75', 'a_pct25', 'a_pct75'])
odf07.insert(0, 'year', 2007)
odf07
```

Out[30]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2007	00-0000	All Occupations	NaN	134354250	7.72	15.10	36.49	16060	31410	75910
1	2007	11-0000	Management occupations	major	6003930	19.64	40.60	#	40850	84440	#
2	2007	11-1011	Chief executives	NaN	299160	31.02	#	#	64530	#	#
3	2007	11-1021	General and operations managers	NaN	1655410	21.15	42.64	#	43990	88700	#
4	2007	11-1031	Legislators	NaN	61110	*	*	*	13090	16220	76260
...	...	...	...	...	...	...	...	...	...	...	...
818	2007	53-7073	Wellhead pumpers	NaN	15780	10.75	17.65	26.72	22350	36700	55580
819	2007	53-7081	Refuse and recyclable material collectors	NaN	126270	8.21	14.15	24.19	17070	29420	50320
820	2007	53-7111	Shuttle car operators	NaN	2660	16.06	19.67	23.90	33410	40920	49720
821	2007	53-7121	Tank car, truck, and ship loaders	NaN	14870	9.41	15.93	27.69	19570	33140	57590
822	2007	53-7199	Material moving workers, all other	NaN	43840	8.30	14.74	25.02	17270	30650	52030

823 rows × 11 columns

In [31]:
```python
odf08 = pd.read_excel('OES/oesm08nat/national__M2008_dl.xls')
odf08 = odf08.drop(columns=['annual', 'hourly', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25',
 'h_pct75', 'a_pct25', 'a_pct75'])
odf08.insert(0, 'year', 2008)
odf08
```

Out[31]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2008	00-0000	All Occupations	NaN	135185230	8.02	15.57	37.99	16680	32390	79020
1	2008	11-0000	Management occupations	major	6152650	20.40	42.15	#	42440	87670	#
2	2008	11-1011	Chief executives	NaN	301930	33.02	76.23	#	68680	158560	#
3	2008	11-1021	General and operations managers	NaN	1697690	21.83	44.02	#	45410	91570	#
4	2008	11-1031	Legislators	NaN	64650	*	*	*	14080	16920	81170
...	...	...	...	...	...	...	...	...	...	...	...
818	2008	53-7073	Wellhead pumpers	NaN	17050	11.60	18.20	27.35	24120	37860	56880
819	2008	53-7081	Refuse and recyclable material collectors	NaN	129080	8.62	14.93	24.68	17930	31050	51330
820	2008	53-7111	Shuttle car operators	NaN	3050	16.53	20.29	24.80	34390	42200	51580
821	2008	53-7121	Tank car, truck, and ship loaders	NaN	12330	10.75	18.14	29.65	22360	37730	61680
822	2008	53-7199	Material moving workers, all other	NaN	41140	8.45	15.68	26.06	17580	32620	54210

823 rows × 11 columns

In [32]:
```python
odf09 = pd.read_excel('OES/oesm09nat/national_dl.xls')
odf09 = odf09.drop(columns=['annual', 'hourly', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25',
 'h_pct75', 'a_pct25', 'a_pct75'])
odf09.insert(0, 'year', 2009)
odf09
```

Out[32]:

	year	occ_code	occ_title	group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2009	00-0000	All Occupations	total	130647610	8.24	15.95	39.02	17140	33190	81160
1	2009	11-0000	Management occupations	major	6116380	20.86	42.95	#	43400	89330	#
2	2009	11-1011	Chief executives	NaN	297640	34.13	77.27	#	71000	160720	#
3	2009	11-1021	General and operations managers	NaN	1689680	22.01	44.55	#	45780	92650	#
4	2009	11-1031	Legislators	NaN	65750	*	*	*	14830	18810	81150
...	...	...	...	...	...	...	...	...	...	...	...
818	2009	53-7073	Wellhead pumpers	NaN	15360	11.42	18.48	27.93	23760	38430	58090
819	2009	53-7081	Refuse and recyclable material collectors	NaN	128940	8.93	15.42	25.09	18580	32070	52190
820	2009	53-7111	Shuttle car operators	NaN	3520	17.63	21.91	28.17	36670	45560	58590
821	2009	53-7121	Tank car, truck, and ship loaders	NaN	11560	10.79	18.76	30.50	22450	39020	63450
822	2009	53-7199	Material moving workers, all other	NaN	32180	8.72	15.39	26.21	18140	32020	54520

823 rows × 11 columns

In [33]:
```python
odf10 = pd.read_excel('OES/oesm10nat/national_M2010_dl.xls')
odf10 = odf10.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25',
 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf10.insert(0, 'year', 2010)
odf10
```

Out[33]:

	year	OCC_CODE	OCC_TITLE	GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2010	00-0000	All Occupations	total	127097160	8.51	16.27	39.97	17690	33840	83140
1	2010	11-0000	Management Occupations	major	6022860	21.57	43.96	#	44860	91440	#
2	2010	11-1011	Chief Executives	NaN	273500	36.14	79.37	#	75160	165080	#
3	2010	11-1021	General and Operations Managers	NaN	1708080	22.73	45.38	#	47280	94400	#
4	2010	11-1031	Legislators	NaN	65710	*	*	*	15790	19260	84320
...	...	...	...	...	...	...	...	...	...	...	...
814	2010	53-7073	Wellhead Pumpers	NaN	12960	11.83	19.54	29.67	24610	40640	61720
815	2010	53-7081	Refuse and Recyclable Material Collectors	NaN	126360	9	15.69	25.75	18730	32640	53560
816	2010	53-7111	Mine Shuttle Car Operators	NaN	3080	18.88	23.13	28.7	39280	48110	59700
817	2010	53-7121	Tank Car, Truck, and Ship Loaders	NaN	10390	10.98	20.57	33.82	22830	42780	70340
818	2010	53-7199	Material Moving Workers, All Other	NaN	28040	8.79	16.35	31.57	18290	34010	65680

819 rows × 11 columns

In [34]:
```python
import pandas as pd
odf11 = pd.read_excel('OES/oesm11nat/national_M2011_dl.xls')
odf11 = odf11.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25',
 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf11.insert(0, 'year', 2011)
odf11
```

Out[34]:

	year	OCC_CODE	OCC_TITLE	GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2011	00-0000	All Occupations	total	128278550	8.65	16.57	41	18000	34460	85280
1	2011	11-0000	Management Occupations	major	6183820	21.98	44.65	#	45720	92880	#
2	2011	11-1011	Chief Executives	NaN	267370	36.47	80.25	#	75860	166910	#
3	2011	11-1021	General and Operations Managers	NaN	1805030	22.87	45.74	#	47580	95150	#
4	2011	11-1031	Legislators	NaN	62180	*	*	*	16280	19630	85920
...	...	...	...	...	...	...	...	...	...	...	...
814	2011	53-7073	Wellhead Pumpers	NaN	13190	12.26	19.87	28.36	25490	41320	58990
815	2011	53-7081	Refuse and Recyclable Material Collectors	NaN	123160	8.92	15.52	26.69	18560	32280	55510
816	2011	53-7111	Mine Shuttle Car Operators	NaN	3080	19.74	24.43	29.26	41060	50820	60860
817	2011	53-7121	Tank Car, Truck, and Ship Loaders	NaN	10960	11.21	19.16	34.06	23310	39860	70840
818	2011	53-7199	Material Moving Workers, All Other	NaN	28370	9.58	16.9	32.42	19930	35160	67440

819 rows × 11 columns

In [35]:
```python
odf12 = pd.read_excel('OES/oesm12nat/national_M2012_dl.xls')
odf12 = odf12.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25',
 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf12.insert(0, 'year', 2012)
odf12
```

Out[35]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2012	00-0000	All Occupations	total	130287700	8.7	16.71	41.74	18090	34750	86810
1	2012	11-0000	Management Occupations	major	6390430	22.12	45.15	#	46000	93910	#
2	2012	11-1000	Top Executives	minor	2212150	21.81	47.86	#	45370	99550	#
3	2012	11-1010	Chief Executives	broad	255940	36.65	80.84	#	76220	168140	#
4	2012	11-1011	Chief Executives	detailed	255940	36.65	80.84	#	76220	168140	#
...	...	...	...	...	...	...	...	...	...	...	...
1389	2012	53-7111	Mine Shuttle Car Operators	detailed	2990	20.01	25.05	28.79	41620	52110	59890
1390	2012	53-7120	Tank Car, Truck, and Ship Loaders	broad	12390	11.49	21.2	34.79	23900	44100	72360
1391	2012	53-7121	Tank Car, Truck, and Ship Loaders	detailed	12390	11.49	21.2	34.79	23900	44100	72360
1392	2012	53-7190	Miscellaneous Material Moving Workers	broad	27260	9.4	17.94	34.57	19550	37320	71900
1393	2012	53-7199	Material Moving Workers, All Other	detailed	27260	9.4	17.94	34.57	19550	37320	71900

1394 rows × 11 columns

In [36]:
```python
odf13 = pd.read_excel('OES/oesm13nat/national_M2013_dl.xls')
odf13 = odf13.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25',
 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf13.insert(0, 'year', 2013)
odf13
```

Out[36]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2013	00-0000	All Occupations	total	132588810	8.74	16.87	42.47	18190	35080	88330
1	2013	11-0000	Management Occupations	major	6542950	22.28	45.96	#	46340	95600	#
2	2013	11-1000	Top Executives	minor	2278260	21.46	48.22	#	44630	100310	#
3	2013	11-1010	Chief Executives	broad	248760	36.07	82.5	#	75030	171610	#
4	2013	11-1011	Chief Executives	detailed	248760	36.07	82.5	#	75030	171610	#
...	...	...	...	...	...	...	...	...	...	...	...
1389	2013	53-7111	Mine Shuttle Car Operators	detailed	2730	20.42	25.49	29.38	42480	53020	61110
1390	2013	53-7120	Tank Car, Truck, and Ship Loaders	broad	12560	12.12	20.33	34.3	25210	42280	71350
1391	2013	53-7121	Tank Car, Truck, and Ship Loaders	detailed	12560	12.12	20.33	34.3	25210	42280	71350
1392	2013	53-7190	Miscellaneous Material Moving Workers	broad	24250	8.9	16.64	32.61	18510	34620	67820
1393	2013	53-7199	Material Moving Workers, All Other	detailed	24250	8.9	16.64	32.61	18510	34620	67820

1394 rows × 11 columns

In [37]:
```python
odf14 = pd.read_excel('OES/oesm14nat/national_M2014_dl.xlsx')
odf14 = odf14.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25', 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf14.insert(0, 'year', 2014)
odf14
```

Out[37]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2014	00-0000	All Occupations	total	135128260	8.82	17.09	43.3	18350	35540	90060
1	2014	11-0000	Management Occupations	major	6741640	22.33	46.75	#	46440	97230	#
2	2014	11-1000	Top Executives	minor	2351130	20.94	48.51	#	43570	100910	#
3	2014	11-1010	Chief Executives	broad	246240	34.97	83.33	#	72750	173320	#
4	2014	11-1011	Chief Executives	detailed	246240	34.97	83.33	#	72750	173320	#
...	...	...	...	...	...	...	...	...	...	...	...
1389	2014	53-7111	Mine Shuttle Car Operators	detailed	2630	20.94	26.44	31.92	43560	55000	66400
1390	2014	53-7120	Tank Car, Truck, and Ship Loaders	broad	12490	11.02	19.8	34.57	22920	41180	71900
1391	2014	53-7121	Tank Car, Truck, and Ship Loaders	detailed	12490	11.02	19.8	34.57	22920	41180	71900
1392	2014	53-7190	Miscellaneous Material Moving Workers	broad	22910	8.91	14.82	33.12	18530	30830	68890
1393	2014	53-7199	Material Moving Workers, All Other	detailed	22910	8.91	14.82	33.12	18530	30830	68890

1394 rows × 11 columns

In [38]:
```python
odf15 = pd.read_excel('OES/oesm15nat/national_M2015_dl.xlsx')
odf15 = odf15.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25', 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf15.insert(0, 'year', 2015)
odf15
```

Out[38]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2015	00-0000	All Occupations	total	137896660	9.07	17.4	44.29	18870	36200	92110
1	2015	11-0000	Management Occupations	major	6936990	22.29	47.38	#	46370	98560	#
2	2015	11-1000	Top Executives	minor	2439900	20.47	48.53	#	42580	100940	#
3	2015	11-1010	Chief Executives	broad	238940	32.98	84.19	#	68600	175110	#
4	2015	11-1011	Chief Executives	detailed	238940	32.98	84.19	#	68600	175110	#
...	...	...	...	...	...	...	...	...	...	...	...
1389	2015	53-7111	Mine Shuttle Car Operators	detailed	2310	20.78	26.59	32.28	43230	55320	67150
1390	2015	53-7120	Tank Car, Truck, and Ship Loaders	broad	11960	10.76	17.63	33.57	22370	36660	69820
1391	2015	53-7121	Tank Car, Truck, and Ship Loaders	detailed	11960	10.76	17.63	33.57	22370	36660	69820
1392	2015	53-7190	Miscellaneous Material Moving Workers	broad	21740	9.3	14.6	32.22	19340	30360	67010
1393	2015	53-7199	Material Moving Workers, All Other	detailed	21740	9.3	14.6	32.22	19340	30360	67010

1394 rows × 11 columns

In [39]:
```python
odf16 = pd.read_excel('OES/oesm16nat/national_M2016_dl.xlsx')
odf16 = odf16.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25', 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf16.insert(0, 'year', 2016)
odf16
```

Out[39]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2016	00-0000	All Occupations	total	140400040	9.27	17.81	45.45	19290	37040	94540
1	2016	11-0000	Management Occupations	major	7090790	22.76	48.46	#	47330	100790	#
2	2016	11-1000	Top Executives	minor	2465800	20.58	49.19	#	42810	102320	#
3	2016	11-1010	Chief Executives	broad	223260	33.55	87.12	#	69780	181210	#
4	2016	11-1011	Chief Executives	detailed	223260	33.55	87.12	#	69780	181210	#
...	...	...	...	...	...	...	...	...	...	...	...
1389	2016	53-7111	Mine Shuttle Car Operators	detailed	1590	21.21	27.14	32.73	44120	56450	68080
1390	2016	53-7120	Tank Car, Truck, and Ship Loaders	broad	10920	10.82	17.2	31.82	22510	35770	66190
1391	2016	53-7121	Tank Car, Truck, and Ship Loaders	detailed	10920	10.82	17.2	31.82	22510	35770	66190
1392	2016	53-7190	Miscellaneous Material Moving Workers	broad	23880	10	13.64	28.29	20810	28370	58840
1393	2016	53-7199	Material Moving Workers, All Other	detailed	23880	10	13.64	28.29	20810	28370	58840

1394 rows × 11 columns

In [40]:
```python
odf17 = pd.read_excel('OES/oes17/nat17.xlsx')
odf17 = odf17.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25', 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf17.insert(0, 'year', 2017)
odf17
```

Out[40]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2017	00-0000	All Occupations	total	142549250	9.6	18.12	46.23	19970	37690	96150
1	2017	11-0000	Management Occupations	major	7280330	23.19	49.32	#	48220	102590	#
2	2017	11-1000	Top Executives	minor	2473740	20.74	49.58	#	43140	103120	#
3	2017	11-1010	Chief Executives	broad	210160	32.74	88.11	#	68110	183270	#
4	2017	11-1011	Chief Executives	detailed	210160	32.74	88.11	#	68110	183270	#
...	...	...	...	...	...	...	...	...	...	...	...
1377	2017	53-7111	Mine Shuttle Car Operators	detailed	1550	21.28	27.35	32.16	44260	56890	66890
1378	2017	53-7120	Tank Car, Truck, and Ship Loaders	broad	10300	11.39	17.72	31.82	23700	36860	66180
1379	2017	53-7121	Tank Car, Truck, and Ship Loaders	detailed	10300	11.39	17.72	31.82	23700	36860	66180
1380	2017	53-7190	Miscellaneous Material Moving Workers	broad	25990	10.36	14.23	28.39	21540	29610	59050
1381	2017	53-7199	Material Moving Workers, All Other	detailed	25990	10.36	14.23	28.39	21540	29610	59050

1382 rows × 11 columns

In [41]:
```python
odf18 = pd.read_excel('OES/oesm18nat/national_M2018_dl.xlsx')
odf18 = odf18.drop(columns=['ANNUAL', 'HOURLY', 'A_MEAN', 'H_MEAN', 'EMP_PRSE', 'MEAN_PRSE', 'H_PCT25', 'H_PCT75', 'A_PCT25', 'A_PCT75'])
odf18.insert(0, 'year', 2018)
odf18
```

Out[41]:

	year	OCC_CODE	OCC_TITLE	OCC_GROUP	TOT_EMP	H_PCT10	H_MEDIAN	H_PCT90	A_PCT10	A_MEDIAN	A_PCT90
0	2018	00-0000	All Occupations	total	144733270	9.95	18.58	47.31	20690	38640	98410
1	2018	11-0000	Management Occupations	major	7616650	23.68	50.11	#	49260	104240	#
2	2018	11-1000	Top Executives	minor	2535640	20.86	49.73	#	43400	103450	#
3	2018	11-1010	Chief Executives	broad	195530	32.86	91.15	#	68360	189600	#
4	2018	11-1011	Chief Executives	detailed	195530	32.86	91.15	#	68360	189600	#
...	...	...	...	...	...	...	...	...	...	...	...
1374	2018	53-7111	Mine Shuttle Car Operators	detailed	1690	21.17	27.09	31.86	44030	56340	66270
1375	2018	53-7120	Tank Car, Truck, and Ship Loaders	broad	9000	12.12	18.38	33.99	25210	38220	70690
1376	2018	53-7121	Tank Car, Truck, and Ship Loaders	detailed	9000	12.12	18.38	33.99	25210	38220	70690
1377	2018	53-7190	Miscellaneous Material Moving Workers	broad	27010	10.76	14.14	27.95	22380	29410	58140
1378	2018	53-7199	Material Moving Workers, All Other	detailed	27010	10.76	14.14	27.95	22380	29410	58140

1379 rows × 11 columns

In [42]:
```python
odf19 = pd.read_excel('oes/oesm19nat/national_m2019_dl.xlsx')
odf19 = odf19.drop(columns=['annual', 'area_title', 'area', 'area_type', 'hourly', 'a_mean', 'h_mean', 'emp_prse', 'mean_prse', 'h_pct25', 'h_pct75', 'a_pct25', 'a_pct75', 'jobs_1000','loc_quotient', 'pct_total', 'own_code', 'naics', 'naics_title', 'i_group'])
odf19.insert(0, 'year', 2019)
odf19
```

Out[42]:

	year	occ_code	occ_title	o_group	tot_emp	h_pct10	h_median	h_pct90	a_pct10	a_median	a_pct90
0	2019	00-0000	All Occupations	total	146875480	10.35	19.14	48.57	21530	39810	101020
1	2019	11-0000	Management Occupations	major	8054120	24.03	50.8	#	49990	105660	#
2	2019	11-1000	Top Executives	minor	2658440	21.01	49.63	#	43710	103230	#
3	2019	11-1010	Chief Executives	broad	205890	29.95	88.68	#	62290	184460	#
4	2019	11-1011	Chief Executives	detailed	205890	29.95	88.68	#	62290	184460	#
...	...	...	...	...	...	...	...	...	...	...	...
1324	2019	53-7081	Refuse and Recyclable Material Collectors	detailed	121330	10.95	18.19	32.39	22780	37840	67370
1325	2019	53-7120	Tank Car, Truck, and Ship Loaders	broad	11620	13.51	20.36	36.56	28090	42360	76050
1326	2019	53-7121	Tank Car, Truck, and Ship Loaders	detailed	11620	13.51	20.36	36.56	28090	42360	76050
1327	2019	53-7190	Miscellaneous Material Moving Workers	broad	28240	11.36	15.28	27.86	23620	31770	57960
1328	2019	53-7199	Material Moving Workers, All Other	detailed	28240	11.36	15.28	27.86	23620	31770	57960

1329 rows × 11 columns

things we can discard from each data set

Get rid of hourly and annual

hourly and annual for each dataset drop everything BUT median hourly wage

We want to add YEAR to every single data file, that way when we merge and reshape the data, we get a nice column that identifies that year on every row

So far, I've done:

Dataset 2001, 2005, 2011, 2013, 2014, 2015, 2016, 2017, 2018, 2019

In [43]:
```python
def rename_caps(df):
    return df.rename(columns=str.lower)
odf10,odf11,odf12,odf13,odf14,odf15,odf16,odf17,odf18= [rename_caps(df) for df in (odf10,odf11,odf12,odf13,odf14,odf15,odf16,odf17,odf18)]


def filter_jobs(df):
    return df.loc[df['occ_code'].isin(codes)]

odf02,odf03,odf04,odf05,odf06,odf07,odf08,odf09,odf10,odf11,odf12,odf13,odf14,odf15,odf16,odf17,odf18,odf19= [filter_jobs(df) for df in (odf02,odf03,odf04,odf05,odf06,odf07,odf08,odf09,odf10,odf11,odf12,odf13,odf14,odf15,odf16,odf17,odf18,odf19)]
```

In [44]:
```python
def fix_w(df):
    return df.rename(columns={'a_wpct10':'a_pct10','a_wpct90':'a_pct90','h_wpct10':'h_pct10','h_wpct90':'h_pct90',})
odf01,odf02 = [fix_w(df)for df in (odf01,odf02)]
```

In [45]:
```python
odfs = [odf01,odf02,odf03,odf04,odf05,odf06,odf07,odf08,odf09,odf10,odf11,odf12,odf13,odf14,odf15,odf16,odf17,odf18,odf19]

odfAll = pd.concat(odfs, sort=True)
```

In [46]:
```python
odfAll.head()
```

Out[46]:

	a_median	a_pct10	a_pct90	group	h_median	h_pct10	h_pct90	hourly	o_group	occ_code	occ_group	occ_title	tot_emp	y
41	38010	23500	72690	NaN	18.27	11.3	34.95	NaN	NaN	13-1071	NaN	Employment, Recruitment, and Placement Special...	173940	200
149	36410	23600	55350	NaN	17.5	11.35	26.61	NaN	NaN	21-1022	NaN	Medical and Public Health Social Workers	103490	200
207	43280	27980	67940	NaN	*	*	*	NaN	NaN	25-2031	NaN	Secondary School Teachers, Except Special and ...	980730	200
326	40970	24490	61870	NaN	19.7	11.78	29.74	NaN	NaN	33-3051	NaN	Police and Sheriff's Patrol Officers	599550	200
444	38700	30410	44010	NaN	18.61	14.62	21.16	NaN	NaN	43-5052	NaN	Postal Service Mail Carriers	355120	200

In [47]:
```python
odfAll = odfAll.drop(columns=['group','o_group','occ_group'])
odfAll
```

Out[47]:

	a_median	a_pct10	a_pct90	h_median	h_pct10	h_pct90	hourly	occ_code	occ_title	tot_emp	year
41	38010	23500	72690	18.27	11.3	34.95	NaN	13-1071	Employment, Recruitment, and Placement Special...	173940	2001.0
149	36410	23600	55350	17.5	11.35	26.61	NaN	21-1022	Medical and Public Health Social Workers	103490	2001.0
207	43280	27980	67940	*	*	*	NaN	25-2031	Secondary School Teachers, Except Special and ...	980730	2001.0
326	40970	24490	61870	19.7	11.78	29.74	NaN	33-3051	Police and Sheriff's Patrol Officers	599550	2001.0
444	38700	30410	44010	18.61	14.62	21.16	NaN	43-5052	Postal Service Mail Carriers	355120	2001.0
...	...	...	...	...	...	...	...	...	...	...	...
294	56750	35000	86130	27.29	16.83	41.41	NaN	21-1022	Healthcare Social Workers	174890	2019.0
384	61660	40540	99660	*	*	*	NaN	25-2031	Secondary School Teachers, Except Special and ...	1035850	2019.0
594	50850	25550	92020	24.45	12.29	44.24	NaN	33-2010	Firefighters	324620	2019.0
610	63150	36960	105230	30.36	17.77	50.59	NaN	33-3051	Police and Sheriff's Patrol Officers	665280	2019.0
846	51310	36990	65620	24.67	17.79	31.55	NaN	43-5052	Postal Service Mail Carriers	339650	2019.0

117 rows × 11 columns

In [ ]:

In [48]:
```python
odfAll[['a_median','a_pct10','a_pct90']] = odfAll[['a_median','a_pct10','a_pct90']].apply(pd.to_numeric,errors='coerce')
```

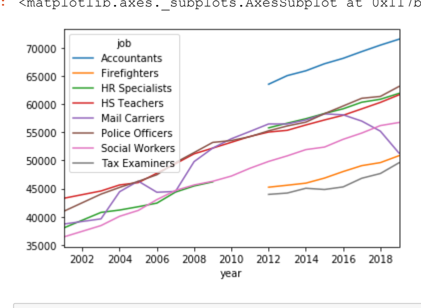
In [49]:
```python
odfAll = odfAll.dropna(subset=['a_median','a_pct10','a_pct90'])
```

In [52]:
```python
myMap ={'13-1071':'HR Specialists',
        '13-2010':'Accountants',
        '13-2081':'Tax Examiners',
        '21-1022':'Social Workers',
        '33-2010':'Firefighters',
        '33-3051':'Police Officers',
        '43-5052':'Mail Carriers',
        '25-2031':'HS Teachers'}

tempJobs = odfAll['occ_code'].map(myMap)
odfAll['job'] = tempJobs
```

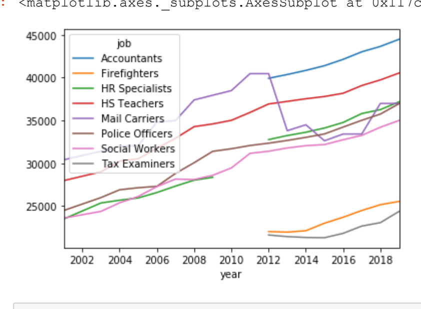
In [56]:
```python
medianAnnuals = odfAll.groupby(['year', 'job'])['a_median'].mean()
medianAnnuals.unstack().plot()
```

Out[56]: <matplotlib.axes._subplots.AxesSubplot at 0x117b85c90>



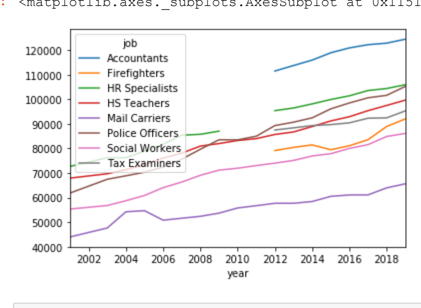
In [57]:
```python
bottomAnnuals = odfAll.groupby(['year', 'job'])['a_pct10'].mean()
bottomAnnuals.unstack().plot()
```

Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x117c9c1d0>



In [58]:
```python
topAnnuals = odfAll.groupby(['year', 'job'])['a_pct90'].mean()
topAnnuals.unstack().plot()
```

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x11517dc10>



In [ ]: