In [152]:
```
# Import Libraries
import pandas as pd
import numpy as np

# Read Data
df1 = pd.read_csv('Data_Washington Fatal Crash Survey.csv')
county_info = pd.read_csv('usa_zipcode_of_wa-1564j.csv')
```

```
/Users/chaitanyapohnerkar/anaconda3/lib/python3.6/site-packages/IPython/core/
interactiveshell.py:2698: DtypeWarning: Columns (21,22,23,301,303) have mixed
types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [153]:
```
county_info.head(20)
```

Out[153]:

	zip	Zipcode name	City	State	County Name
0	98520	ABERDEEN, WA	ABERDEEN	WA	GRAYS HARBOR
1	98220	ACME, WA	ACME	WA	WHATCOM
2	99101	ADDY, WA	ADDY	WA	STEVENS
3	98522	ADNA, WA	ADNA	WA	LEWIS
4	99001	AIRWAY HEIGHTS, WA	AIRWAY HEIGHTS	WA	SPOKANE
5	99102	ALBION, WA	ALBION	WA	WHITMAN
6	98524	ALLYN, WA	ALLYN	WA	MASON
7	99103	ALMIRA, WA	ALMIRA	WA	LINCOLN
8	98526	AMANDA PARK, WA	AMANDA PARK	WA	GRAYS HARBOR
9	98601	AMBOY, WA	AMBOY	WA	CLARK
10	98221	ANACORTES, WA	ANACORTES	WA	SKAGIT
11	99401	ANATONE, WA	ANATONE	WA	ASOTIN
12	98303	ANDERSON ISLAND, WA	ANDERSON ISLAND	WA	PIERCE
13	98602	APPLETON, WA	APPLETON	WA	KLICKITAT
14	98811	ARDENVOIR, WA	ARDENVOIR	WA	CHELAN
15	98603	ARIEL, WA	ARIEL	WA	COWLITZ
16	98223	ARLINGTON, WA	ARLINGTON	WA	SNOHOMISH
17	98304	ASHFORD, WA	ASHFORD	WA	PIERCE
18	99402	ASOTIN, WA	ASOTIN	WA	ASOTIN
19	98001	AUBURN, WA	AUBURN	WA	KING

In [154]:
```
# Identify High Risk Factors

df1['isHighRisk'] = np.where((df1['dd_inv']==1) | (df1['dpd_inv']==1) | (df1['
```

In [155]:
```
# Group by resident zipcode and count number of high risk drivers and drivers

df2 = df1.groupby(["dzip"]).agg(
    count_high_risk=pd.NamedAgg(column="isHighRisk", aggfunc="sum"),
    count_drivers = pd.NamedAgg(column="dzip", aggfunc="count"))
df2.reset_index()
```

Out[155]:

	dzip	count_high_risk	count_drivers
0	0.0	21	27
1	1060.0	1	1
2	2863.0	1	1
3	3869.0	1	1
4	7021.0	0	1
...	...	...	...
695	99508.0	0	2
696	99611.0	1	1
697	99692.0	0	1
698	99998.0	6	9
699	99999.0	38	95

700 rows × 3 columns

In [156]:
```
# sort zipcodes by most number of high risk drivers
df2=df2.sort_values(by='count_high_risk', ascending=False)
```

In [157]:
```
df2.reset_index(inplace=True)
```

In [158]:
```
# make zipcode as interger
df2 = df2.astype({"dzip":'int'})
```

In [159]:
```
# remove invalid zipcodes
df2.drop(df2[df2.dzip == 0].index, inplace=True)
df2.drop(df2[df2.dzip == 99999].index, inplace=True)
```

In [160]:
```
# Find proportion of high risk drivers in total crashes
df2['%_at_high_risk'] = df2['count_high_risk']/df2['count_drivers']
```

In [161]:
```
# view zipcodes where high risk drivers are greater than 20
df2[df2['count_high_risk']>20]
```

Out[161]:

	dzip	count_high_risk	count_drivers	%_at_high_risk
1	98444	34	43	0.790698
2	98387	28	35	0.800000
3	99301	27	47	0.574468
4	99206	27	30	0.900000
5	98837	27	32	0.843750
6	98391	27	32	0.843750
7	98258	26	31	0.838710
8	98223	25	31	0.806452
9	98404	25	32	0.781250
10	98584	24	30	0.800000
11	99344	22	25	0.880000
12	98901	22	31	0.709677

In [162]:
```
# export dataframe
df2.to_csv('/Users/chaitanyapohnerkar/Downloads/High_Risk_Drivers.csv')
```

In [163]:
```
county_info.rename(columns = {'zip':'dzip'}, inplace = True)
```

In [164]:

```
county_info = county_info.astype({"dzip":'int'})
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-164-34d3ef929c9a> in <module>()
      1
----> 2 county_info = county_info.astype({"dzip":'int'})

~/anaconda3/lib/python3.6/site-packages/pandas/core/generic.py in astype(sel
f, dtype, copy, errors)
   5531                 if col_name in dtype:
   5532                     results.append(
-> 5533                         col.astype(dtype=dtype[col_name], copy=copy,
errors=errors)
   5534                     )
   5535                 else:

~/anaconda3/lib/python3.6/site-packages/pandas/core/generic.py in astype(sel
f, dtype, copy, errors)
   5546         else:
   5547             # else, only a single dtype is given
-> 5548             new_data = self._mgr.astype(dtype=dtype, copy=copy, error
s=errors,)
   5549             return self._constructor(new_data).__finalize__(self, met
hod="astype")
   5550

~/anaconda3/lib/python3.6/site-packages/pandas/core/internals/managers.py in
astype(self, dtype, copy, errors)
    602         self, dtype, copy: bool = False, errors: str = "raise"
    603     ) -> "BlockManager":
--> 604         return self.apply("astype", dtype=dtype, copy=copy, errors=er
rors)
    605
    606     def convert(

~/anaconda3/lib/python3.6/site-packages/pandas/core/internals/managers.py in
apply(self, f, align_keys, **kwargs)
    407                 applied = b.apply(f, **kwargs)
    408             else:
--> 409                 applied = getattr(b, f)(**kwargs)
    410             result_blocks = _extend_blocks(applied, result_blocks)
    411

~/anaconda3/lib/python3.6/site-packages/pandas/core/internals/blocks.py in as
type(self, dtype, copy, errors)
    593             vals1d = values.ravel()
    594             try:
--> 595                 values = astype_nansafe(vals1d, dtype, copy=True)
    596             except (ValueError, TypeError):
    597                 # e.g. astype_nansafe can fail on object-dtype of str
ings

~/anaconda3/lib/python3.6/site-packages/pandas/core/dtypes/cast.py in astype_
nansafe(arr, dtype, copy, skipna)
    972         # work around NumPy brokenness, #1987
    973         if np.issubdtype(dtype.type, np.integer):
--> 974             return lib.astype_intsafe(arr.ravel(), dtype).reshape(ar
r.shape)
```

```
    975
    976         # if we have a datetime/timedelta array of objects

pandas/_libs/lib.pyx in pandas._libs.lib.astype_intsafe()

ValueError: cannot convert float NaN to integer
```

In [165]:
```
print(county_info[county_info['dzip'].isnull()])
```

```
     dzip Zipcode name City State County Name
693   NaN          NaN  NaN   NaN         NaN
694   NaN          NaN  NaN   NaN         NaN
```

In [166]:
```
county_info['dzip'] = pd.to_numeric(county_info['dzip'], errors='coerce')
```

In [168]:
```
df3 = pd.merge(df2, county_info, on='dzip', how='inner')
```

In [169]:
```
df3.to_csv('/Users/chaitanyapohnerkar/Downloads/High_Risk_Drivers1.csv')
```

In [170]:
```
df3.head(20)
```

Out[170]:

	dzip	count_high_risk	count_drivers	%_at_high_risk	Zipcode name	City	State	
0	98444	34	43	0.790698	TACOMA, WA	TACOMA	WA	
1	98387	28	35	0.800000	SPANAWAY, WA	SPANAWAY	WA	
2	99301	27	47	0.574468	PASCO, WA	PASCO	WA	F
3	99206	27	30	0.900000	SPOKANE, WA	SPOKANE	WA	S
4	98837	27	32	0.843750	MOSES LAKE, WA	MOSES LAKE	WA	
5	98391	27	32	0.843750	BONNEY LAKE, WA	BONNEY LAKE	WA	
6	98258	26	31	0.838710	LAKE STEVENS, WA	LAKE STEVENS	WA	SN
7	98223	25	31	0.806452	ARLINGTON, WA	ARLINGTON	WA	SN
8	98404	25	32	0.781250	TACOMA, WA	TACOMA	WA	
9	98584	24	30	0.800000	SHELTON, WA	SHELTON	WA	
10	99344	22	25	0.880000	OTHELLO, WA	OTHELLO	WA	
11	98901	22	31	0.709677	YAKIMA, WA	YAKIMA	WA	
12	98022	20	29	0.689655	ENUMCLAW, WA	ENUMCLAW	WA	
13	98003	20	32	0.625000	FEDERAL WAY, WA	FEDERAL WAY	WA	
14	98284	19	22	0.863636	SEDRO WOOLLEY, WA	SEDRO WOOLLEY	WA	
15	98168	19	27	0.703704	SEATTLE, WA	SEATTLE	WA	
16	98951	19	32	0.593750	WAPATO, WA	WAPATO	WA	
17	98208	18	22	0.818182	EVERETT, WA	EVERETT	WA	SN
18	98597	18	22	0.818182	YELM, WA	YELM	WA	TH
19	98273	18	24	0.750000	MOUNT VERNON, WA	MOUNT VERNON	WA	

In [ ]: