Old Dataset(Staircase)

Importing Jury data csv file to data and renaming the columns

In [10]:
```
# coding: utf-8
import pandas as pd
import numpy as np

data = pd.read_csv('jury_data.csv', encoding= 'ISO-8859-1', skiprows=[0,2])
data.rename(columns={"Was defendant Mesa Management negligent?": "Mesa_Negligent",
                         "Was Mesa Management's negligence a substantial factor in cau
sing harm to  Mackenzie Dunn?":"Liability",
                         "What are the total damages that you find that MacKenzie Dunn
 sufferered?":"damages" ,
                         "What is your sex?": "gender",
                         "Please write your answer to the preceding damages question i
n words (quality check).":"damages_word",
                         "What percentage of responsibility for Mackenzie Dunn's injur
ies was each party responsible for? (Answers should add up to 100%) - Mesa Management
 Co":"Mesa_reponsible_percentage",
                         "Path":"Scenario",
                         "Was MacKenzie Dunn negligent?":"Dunn_negligent",
                         "Unnamed: 63":"perc_calc"
                         },inplace=True)
#data['mm_perc'] = np.where(data['Mesa_reponsible_percentage']>=1, data['perc_calc'],d
ata['Mesa_reponsible_percentage'])
data['mm_perc']=data['Mesa_reponsible_percentage']
req_data = pd.DataFrame(data[["Mesa_Negligent","damages","Liability",
                 "gender",
                 "damages_word",
                 "Scenario","Dunn_negligent","perc_calc","Start Date","End Date","mm_p
erc"]])


req_data['Liability'] = req_data['Liability'].map({'Yes': 1, 'No': 0})


print(req_data.columns)
```

```
Index(['Mesa_Negligent', 'damages', 'Liability', 'gender', 'damages_word',
       'Scenario', 'Dunn_negligent', 'perc_calc', 'Start Date', 'End Date',
       'mm_perc'],
      dtype='object')
```

Replacing Null values and changing the data types of required columns

In [11]:
```
print(pd.isnull(req_data).any())
print(pd.isnull(req_data['Scenario']).any())
req_data = req_data[np.isfinite(data['Scenario'])]
print(pd.isnull(req_data['Scenario']).any())
req_data['damages'].fillna(0,inplace=True)
req_data['damages_word'].fillna(0,inplace=True)
req_data['mm_perc'].fillna(1,inplace=True)
req_data['perc_calc'].fillna(0,inplace=True)
#Dropping the last two rows which has null values
#data[pd.isnull(data['Path'])]
#data['Path']=data.Path.dropna(inplace= True)
#data[pd.isnull(data['Path'])]
print(pd.isnull(req_data).any())



#Changing data types of columns
req_data['End Date'] = pd.to_datetime(data['End Date'])
req_data['Start Date'] = pd.to_datetime(data['Start Date'])
req_data['Scenario']= req_data.Scenario.astype(int)
req_data['Liability']= req_data.Liability.astype(int)
req_data.dtypes

# Getting the id of the column
data.columns.get_loc("Liability")
```

```
Mesa_Negligent     True
damages            True
Liability          True
gender             True
damages_word       True
Scenario           True
Dunn_negligent     True
perc_calc          True
Start Date        False
End Date          False
mm_perc            True
dtype: bool
True
False
Mesa_Negligent    False
damages           False
Liability         False
gender            False
damages_word      False
Scenario          False
Dunn_negligent     True
perc_calc         False
Start Date        False
End Date          False
mm_perc           False
dtype: bool
```

Out[11]: 55

Cleaning Damages and perc_calc column

In [12]:
```
print(req_data.isnull().any())
req_data['damages'] = req_data['damages'].str.replace(',', '')
req_data['perc_calc'] = req_data['perc_calc'].str.replace('$', '')
req_data['perc_calc'] = req_data['perc_calc'].str.replace(',', '')
req_data['perc_calc'] = req_data['perc_calc'].str.replace('-', '')
req_data['perc_calc'] = req_data['perc_calc'].str.replace("  ", '')
#req_data['mm_perc'] = req_data['mm_perc'].str.replace("$", '')
#req_data['mm_perc'] = req_data['mm_perc'].str.replace(",", '')
#req_data['mm_perc'] = req_data['mm_perc'].str.replace("  ", '')
#req_data.damages=pd.to_numeric(req_data['damages'].str.replace(',', ''))
#req_data.perc_calc=pd.to_numeric(req_data.perc_calc)

req_data.damages=pd.to_numeric(req_data['damages'])
req_data.perc_calc=pd.to_numeric(req_data.perc_calc)
req_data['damages'].fillna(0,inplace=True)
req_data['mm_perc'].fillna(1,inplace=True)
req_data['perc_calc'].fillna(0,inplace=True)
#print(req_data.damages)
print(req_data.isnull().any())

print(req_data[pd.isnull(req_data['Dunn_negligent'])])
```

```
Mesa_Negligent    False
damages           False
Liability         False
gender            False
damages_word      False
Scenario          False
Dunn_negligent     True
perc_calc         False
Start Date        False
End Date          False
mm_perc           False
dtype: bool
Mesa_Negligent    False
damages           False
Liability         False
gender            False
damages_word      False
Scenario          False
Dunn_negligent     True
perc_calc         False
Start Date        False
End Date          False
mm_perc           False
dtype: bool
    Mesa_Negligent  damages  Liability  gender damages_word  Scenario  \
0               No      0.0          0  Female             0         1
1               No      0.0          0    Male             0         1
2               No      0.0          0    Male             0         1
3               No      0.0          0    Male             0         1
4               No      0.0          0    Male             0         1
5               No      0.0          0    Male             0         1
6               No      0.0          0    Male             0         1
7               No      0.0          0  Female             0         1
8               No      0.0          0    Male             0         1
9               No      0.0          0  Female             0         1
10              No      0.0          0    Male             0         1
11              No      0.0          0  Female             0         1
12              No      0.0          0    Male             0         1
13              No      0.0          0  Female             0         1
14              No      0.0          0    Male             0         1
15              No      0.0          0    Male             0         1
16              No      0.0          0    Male             0         1
17              No      0.0          0    Male             0         1
18              No      0.0          0    Male             0         1
19              No      0.0          0    Male             0         1
20              No      0.0          0    Male             0         1
21              No      0.0          0  Female             0         1
22              No      0.0          0  Female             0         1
23              No      0.0          0  Female             0         1
24              No      0.0          0    Male             0         1
25              No      0.0          0    Male             0         1
26              No      0.0          0    Male             0         1
27              No      0.0          0    Male             0         1
28              No      0.0          0  Female             0         1
29              No      0.0          0    Male             0         1
..             ...      ...        ...     ...           ...       ...
397            Yes      0.0          0  Female             0         2
398            Yes      0.0          0    Male             0         2
399            Yes      0.0          0  Female             0         2
400            Yes      0.0          0  Female             0         2
401            Yes      0.0          0  Female             0         2
402            Yes      0.0          0  Female             0         2
```

403	Yes	0.0	0	Female	0	2
404	Yes	0.0	0	Female	0	2
405	Yes	0.0	0	Female	0	2
406	Yes	0.0	0	Female	0	2
407	Yes	0.0	0	Male	0	3
408	Yes	0.0	0	Female	0	3
409	Yes	0.0	0	Male	0	3
410	Yes	0.0	0	Female	0	4
411	Yes	0.0	0	Male	0	4
412	Yes	0.0	0	Male	0	4
413	Yes	0.0	0	Male	0	4
414	Yes	0.0	0	Male	0	4
415	Yes	0.0	0	Male	0	4
416	Yes	0.0	0	Male	0	4
417	Yes	0.0	0	Female	0	5
418	Yes	0.0	0	Male	0	5
419	Yes	0.0	0	Female	0	5
420	Yes	0.0	0	Female	0	5
421	Yes	0.0	0	Male	0	5
422	Yes	0.0	0	Female	0	5
423	Yes	0.0	0	Female	0	5
424	Yes	0.0	0	Female	0	5
425	Yes	0.0	0	Female	0	5
426	Yes	0.0	0	Male	0	5

	Dunn_negligent	perc_calc	Start Date	End Date	mm_perc
0	NaN	0.0	2017-09-29 13:58:00	2017-09-29 14:16:00	1.0
1	NaN	0.0	2017-09-29 14:00:00	2017-09-29 14:18:00	1.0
2	NaN	0.0	2017-09-29 13:57:00	2017-09-29 14:19:00	1.0
3	NaN	0.0	2017-09-29 14:01:00	2017-09-29 14:21:00	1.0
4	NaN	0.0	2017-09-29 14:04:00	2017-09-29 14:23:00	1.0
5	NaN	0.0	2017-09-29 14:06:00	2017-09-29 14:25:00	1.0
6	NaN	0.0	2017-09-29 14:06:00	2017-09-29 14:26:00	1.0
7	NaN	0.0	2017-09-29 14:16:00	2017-09-29 14:35:00	1.0
8	NaN	0.0	2017-09-29 14:27:00	2017-09-29 14:49:00	1.0
9	NaN	0.0	2017-09-29 14:44:00	2017-09-29 15:02:00	1.0
10	NaN	0.0	2017-09-29 15:09:00	2017-09-29 15:27:00	1.0
11	NaN	0.0	2017-09-29 15:55:00	2017-09-29 16:16:00	1.0
12	NaN	0.0	2017-09-29 18:28:00	2017-09-29 18:51:00	1.0
13	NaN	0.0	2017-09-29 19:54:00	2017-09-29 20:14:00	1.0
14	NaN	0.0	2017-09-29 20:56:00	2017-09-29 21:16:00	1.0
15	NaN	0.0	2017-09-30 11:47:00	2017-09-30 12:04:00	1.0
16	NaN	0.0	2017-09-30 12:56:00	2017-09-30 13:13:00	1.0
17	NaN	0.0	2017-09-30 14:15:00	2017-09-30 14:26:00	1.0
18	NaN	0.0	2017-09-30 16:19:00	2017-09-30 16:38:00	1.0
19	NaN	0.0	2017-10-01 21:22:00	2017-10-01 21:40:00	1.0
20	NaN	0.0	2017-10-01 21:47:00	2017-10-01 22:07:00	1.0
21	NaN	0.0	2017-10-02 02:43:00	2017-10-02 03:02:00	1.0
22	NaN	0.0	2017-10-02 06:47:00	2017-10-02 07:11:00	1.0
23	NaN	0.0	2017-10-02 07:12:00	2017-10-02 07:31:00	1.0
24	NaN	0.0	2017-10-02 07:21:00	2017-10-02 07:38:00	1.0
25	NaN	0.0	2017-10-02 07:57:00	2017-10-02 08:15:00	1.0
26	NaN	0.0	2017-10-02 08:31:00	2017-10-02 08:50:00	1.0
27	NaN	0.0	2017-10-02 09:01:00	2017-10-02 09:19:00	1.0
28	NaN	0.0	2017-10-02 09:20:00	2017-10-02 09:43:00	1.0
29	NaN	0.0	2017-10-02 10:41:00	2017-10-02 10:59:00	1.0
..	...	...	...	...	...
397	NaN	0.0	2017-09-29 14:00:00	2017-09-29 14:19:00	1.0
398	NaN	0.0	2017-09-29 18:28:00	2017-09-29 18:48:00	1.0
399	NaN	0.0	2017-09-30 11:49:00	2017-09-30 12:10:00	1.0
400	NaN	0.0	2017-09-30 14:52:00	2017-09-30 15:12:00	1.0
401	NaN	0.0	2017-10-01 10:25:00	2017-10-01 10:42:00	1.0

402	NaN	0.0	2017-10-03 13:07:00	2017-10-03 13:26:00	1.0
403	NaN	0.0	2017-10-05 18:01:00	2017-10-05 18:20:00	1.0
404	NaN	0.0	2017-10-05 18:16:00	2017-10-05 18:35:00	1.0
405	NaN	0.0	2017-10-05 18:17:00	2017-10-05 18:40:00	1.0
406	NaN	0.0	2017-10-05 18:20:00	2017-10-05 18:49:00	1.0
407	NaN	0.0	2017-09-30 20:10:00	2017-09-30 20:30:00	1.0
408	NaN	0.0	2017-10-05 17:45:00	2017-10-05 18:12:00	1.0
409	NaN	0.0	2017-10-05 18:03:00	2017-10-05 18:25:00	1.0
410	NaN	0.0	2017-09-29 14:14:00	2017-09-29 14:44:00	1.0
411	NaN	0.0	2017-09-29 18:35:00	2017-09-29 18:58:00	1.0
412	NaN	0.0	2017-10-01 04:32:00	2017-10-01 04:54:00	1.0
413	NaN	0.0	2017-10-02 09:26:00	2017-10-02 09:48:00	1.0
414	NaN	0.0	2017-10-05 17:32:00	2017-10-05 17:52:00	1.0
415	NaN	0.0	2017-10-05 17:33:00	2017-10-05 17:54:00	1.0
416	NaN	0.0	2017-10-05 17:39:00	2017-10-05 18:02:00	1.0
417	NaN	0.0	2017-09-30 17:52:00	2017-09-30 18:11:00	1.0
418	NaN	0.0	2017-10-03 07:48:00	2017-10-03 08:08:00	1.0
419	NaN	0.0	2017-10-04 10:56:00	2017-10-04 11:17:00	1.0
420	NaN	0.0	2017-10-05 17:25:00	2017-10-05 17:47:00	1.0
421	NaN	0.0	2017-10-05 17:29:00	2017-10-05 17:50:00	1.0
422	NaN	0.0	2017-10-05 17:25:00	2017-10-05 17:52:00	1.0
423	NaN	0.0	2017-10-05 17:32:00	2017-10-05 17:55:00	1.0
424	NaN	0.0	2017-10-05 17:32:00	2017-10-05 17:57:00	1.0
425	NaN	0.0	2017-10-05 17:58:00	2017-10-05 18:19:00	1.0
426	NaN	0.0	2017-10-05 18:10:00	2017-10-05 18:30:00	1.0

[427 rows x 11 columns]

EDA

Calculating Discounted Damages stats for case expected value

In [13]:
```
req_data['Discounted_damages']=req_data.damages*pd.to_numeric(req_data.mm_perc)

req_data['winrate_percentage']=req_data.Liability
req_data['Discounted_damages_mean']=req_data['Discounted_damages']
req_data['Discounted_damages_median']=req_data['Discounted_damages']
req_data['Discounted_damages_sd']=req_data['Discounted_damages']

winrate_damages_expected=req_data.groupby('Scenario').aggregate(
    {'winrate_percentage': np.mean, 'Discounted_damages_mean': np.mean,'Discounted_dam
ages_median':np.median,'Discounted_damages_sd':np.std})

winrate_damages_expected['winrate_percentage']=winrate_damages_expected['winrate_perce
ntage']*100.0
winrate_damages_expected
```

Out[13]:

	winrate_percentage	Discounted_damages_mean	Discounted_damages_median	Discou
Scenario				
1	47.263682	344885.572139	0.0	1.5364
2	48.730964	460532.994924	0.0	2.2140
3	68.817204	398266.129032	110000.0	2.5616
4	60.294118	108101.715686	100000.0	1.0667
5	57.591623	92761.780105	80000.0	1.0090

Calculating Discounted Damages stats when plaintiff wins

In [14]:
```
#req_data['winrate_percentage']=np.mean(req_data.Juror_Response)
#print(req_data)
req_data['mm_perc'].fillna(1,inplace=True)

req_data['Discounted_damages_mean1']=req_data['Discounted_damages']
req_data['Discounted_damages_median1']=req_data['Discounted_damages']
req_data['Discounted_damages_sd1']=req_data['Discounted_damages']
#print(req_data.mm_perc)

winrate_damages_plaintiffwin=req_data.loc[(req_data['Dunn_negligent']=='No') & (req_da
ta['Liability']==1)].groupby('Scenario').aggregate({'Discounted_damages_mean1': np.mea
n,'Discounted_damages_median1':np.median,'Discounted_damages_sd1':np.std})


winrate_damages_plaintiffwin
```

Out[14]:

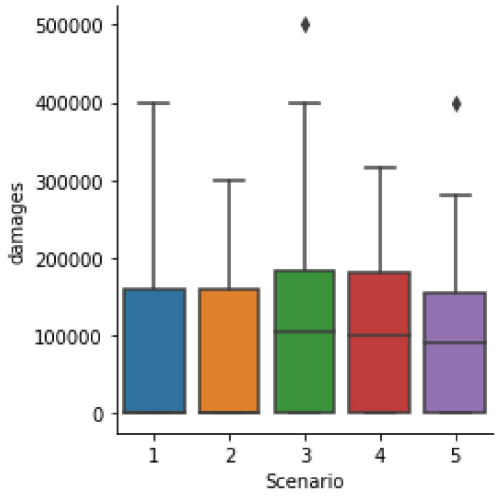
	Discounted_damages_mean1	Discounted_damages_median1	Discounted_damages_sd
Scenario			
1	185125.000000	180000.0	78852.429869
2	177743.902439	160000.0	72767.821367
3	186972.477064	180000.0	80274.466527
4	189429.906542	180000.0	73657.559546
5	173118.279570	150000.0	78781.184228

Plotting graph for Liability vs Path

In [15]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

sns.factorplot(x='Scenario', y='damages', kind='box',data=req_data)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x24892ab6828>



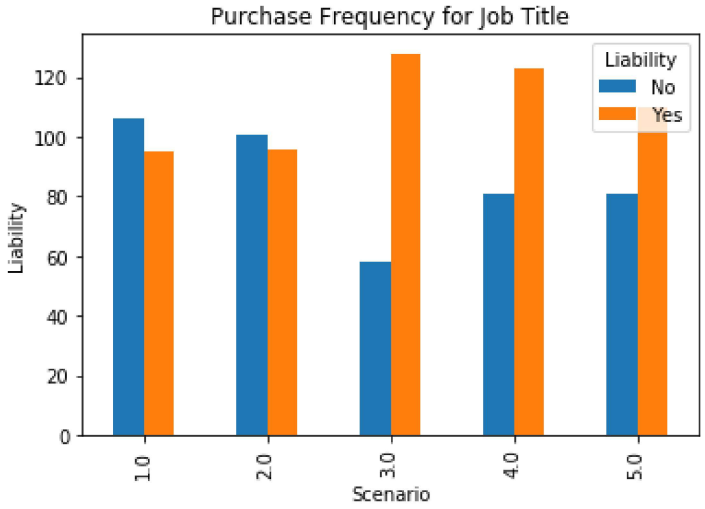
In [16]:
```
pd.crosstab(data.Scenario,data.Liability).plot(kind='bar')
plt.title('Purchase Frequency for Job Title')
plt.xlabel('Scenario')
plt.ylabel('Liability')
plt.savefig('Juror Response per each Scenario')



a = req_data['Scenario']
b = req_data['Liability']
pd.crosstab(a,b)
```

Out[16]:

Liability	0	1
Scenario		
1	106	95
2	101	96
3	58	128
4	81	123
5	81	110



From the graph we can see that

- Winrate increases when subsequent remedial measures introduced at scenario3.
- Winrate decreases slightly when limited jury instruction presented at Scenario4.
- Similarly winrate and damages decreases more when more explanation on limiting jury instruction provided at Scenario5.But not decreased as in Scenario2(before limiting jury instructions)

Importing data to "old_data.csv" file

In [17]:
```
req_data.to_csv("old_data.csv",sep=',')
```