International car accident database

This dataset contains registered car accidents from several countries of Europe (2000-2011).

The fields are: * **accident_id**: unique ID of the registers
* **accident_date**, time_of_the_day and year, month, day: time information of the accident
* **cost**: (calculated) cost of the renovation after the accident
* **car_brand** (and brand, type): car manufacturer
* **car_type**: gas / diesel / electric
* **country**: country name
* **weather**: wheather condition, rainy
* **reported**: nature of accident being reported (1) or not (0), toward authorities or insurance companies
* and least important field as **factory_id**: parts of original car factory ID

Data analysis

Some topics and questions are elaborated in this notebook, some are presented in a separate file.

Understanding the dataset

Loading the data from a csv file

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

In [2]:
```
accidents = pd.read_csv ('accidents.csv', delimiter =',')
```

In [27]:
```python
accidents.head()
```

Out[27]:

	car_type	city	factory_id	accident_date	accident_id	car_brand	country	reported	weather	time_of_the_day	cost	yeai
0	Gas	SALONIKA	C3532xx	2000-01-01	535167	FORD KUGA	Greece	False	Rainy	Day	0	200(
1	Gas	UNKNOWN	C3333xx	2000-01-01	547337	VOLKSWAGEN GOLF	NaN	False	NaN	NaN	0	200(
2	Gas	COPENHAGEN	C3333xx	2000-01-01	547339	VOLKSWAGEN GOLF	Denmark	False	NaN	NaN	0	200(
3	Gas	KIEV	C3232xx	2000-01-01	542341	VOLKSWAGEN GOLF	Ukraine	False	NaN	NaN	0	200(
4	Gas	UNKNOWN	C55x	2000-01-01	540395	BMW 1	France	False	NaN	Day	0	200(

For specific plots of time periods we split the **accident_date** string into year, month, day data.

In [3]:
```python
accidents[['year', 'month', 'day']]= accidents['accident_date'].str.split("-", expand=True)
```

1) Seasonality of accidents

In which month did the most accidents occur between January 1, 2000 and December 31, 2011? Monthly list is calculated as percentage of total accident amount.

In [57]:
```python
step = accidents[['month', 'cost']]
step = step.groupby(['month']).count()
step['sum'] = step['cost'].sum()
step ['percent'] = step ['cost'] / step ['sum']
step = step [['percent']]
step
```

Out[57]:

	percent
month	
01	0.033405
02	0.028930
03	0.049663
04	0.078701
05	0.106283
06	0.076770
07	0.116289
08	0.134217
09	0.137404
10	0.130922
11	0.070181
12	0.037236

2) In what weather are more expensive accidents?

Is the average accident damage higher in rainy, windy or sunny weather? (average of non-zero values are compared)

In [16]:

```
tfw = accidents[['weather', 'cost']][accidents.cost >0]
tfw = tfw.groupby('weather').agg (['sum','mean', 'count', 'max', 'median'])
tfw
```

Out[16]:

	cost				
	sum	mean	count	max	median
weather					
Rainy	278397	1062.583969	262	43928	84
Sunny	479809	647.515520	741	75681	43
Windy	260447	566.189130	460	29253	66

As generally would be expected, rainy days cause the highest accident costs, however the number of registered accidents on rainy days is the lowest (in this dataset).

3) In which year were the most *reported* accidents?

Yearly data is calculated (count), including basic statistical values of the costs (sum, mean, median, minimum, maximum, standard error).

In [18]:

```
step = accidents[accidents.reported == True]
step = step[['year', 'cost']][accidents.cost > 0]
step = step.groupby('year')['cost'].agg(['sum', 'mean','count', 'median', 'min', 'max', 'std'])
step = step.sort_values (by= ['sum'], ascending = False)
step
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: UserWarning: Boolean Series key will be reindexed to match Data
Frame index.
```

Out[18]:

year	sum	mean	count	median	min	max	std
2011	144909	1575.097826	92	40.5	1	30107	5516.109355
2009	117214	1698.753623	69	110.0	1	75681	9216.142026
2005	93292	2392.102564	39	180.0	1	43928	7391.847553
2006	65254	1553.666667	42	161.5	1	29253	5059.975965
2010	62060	838.648649	74	83.5	1	29374	3500.238166
2008	61694	907.264706	68	60.5	1	17760	3219.275531
2007	58973	1072.236364	55	113.0	1	12018	2443.683361
2002	52157	1372.552632	38	72.0	1	9751	2755.206567
2003	39406	772.666667	51	126.0	3	11656	1948.869392
2000	31006	1240.240000	25	68.0	2	17269	3489.690486
2004	29353	815.361111	36	155.5	1	15418	2581.293487
2001	21480	795.555556	27	166.0	6	5646	1555.858910

Above only those accidents are considered which are reported and have non-zero (repair) costs.
In general, for the complete list of accidents this query provides the answer for the question:

In [ ]:

```
step = accidents['year']
step = step.groupby('year')['year'].agg(['count'])
```

4 A) What ratio of the accidents are reported?

In [70]:

```
step = accidents[['reported']][accidents.reported == True].count() / accidents[['reported']].count()
step
```

Out[70]:
```
reported    0.187695
dtype: float64
```

Which means, in general (for all Europe) ~18,8% of the cases are reported.

A yearly tendency report is included in the separate presentation file.

4 B) What ratio of the accidents with costs are reported?

In [71]:
```
step = accidents[['reported']][(accidents.reported == True) & (accidents.cost > 0)].count() / accidents[['reported']][accident
step
```

Out[71]:
```
reported    0.299611
dtype: float64
```

Compared to previous calculation this result shows higher reporting rate in the case of costly crashes.

4 C) In which year was the highest rate of reporting?

no: not reported, yes: reported cases

In [73]:
```
step = accidents[['year', 'cost','reported']]
step = step.groupby(['year', 'reported']).count()
step = step.unstack('reported')
step = step.cost.reset_index()
step.columns = ['year', 'no', 'yes']
step['ratio'] = (step.yes / (step.no + step.yes))
step
```

Out[73]:

	year	no	yes	ratio
0	2000	4625	160	0.033438
1	2001	4308	205	0.045424
2	2002	4438	509	0.102891
3	2003	4177	691	0.141947
4	2004	4494	1094	0.195777
5	2005	4874	599	0.109446
6	2006	4629	818	0.150174
7	2007	4794	1000	0.172592
8	2008	4374	1118	0.203569
9	2009	4868	1742	0.263540
10	2010	4028	2140	0.346952
11	2011	3402	2173	0.389776

5) For which car brands, the average damage cost value of accidents is the highest?

Values greater than zero were involved.

In [74]:

```
accidents[['brand', 'type1']]= accidents['car_brand'].str.split(" ", n=1, expand=True)
tfw = accidents[['cost', 'brand']][accidents.cost >0]
tfw = tfw.groupby('brand').agg(['sum', 'count', 'mean' , 'max'])
tfw = tfw.cost.reset_index()
tfw.columns = ['brand', 'total_cost', 'no_of_cases', 'mean' , 'max']
tfw.sort_values(by = ['mean'], inplace= True, ascending = False)
tfw
```

Out[74]:

	brand	total_cost	no_of_cases	mean	max
14	OPEL	31947	14	2281.928571	27846
9	LADA	17382	8	2172.750000	15418
4	FIAT	42015	21	2000.714286	15653
2	CITROEN	24155	18	1341.944444	17269
21	VOLKSWAGEN	893806	1004	890.245020	75681
6	HYUNDAI	7328	9	814.222222	5128
18	SKODA	41487	58	715.293103	18689
19	SUZUKI	685	1	685.000000	685
15	PEUGEOT	51751	77	672.090909	17760
12	MINI	1141	2	570.500000	1139
10	MAZDA	569	1	569.000000	569
0	AUDI	32650	62	526.612903	22694
8	KIA	5315	12	442.916667	2925
11	MERCEDES	15309	36	425.250000	5022
5	FORD	90186	224	402.616071	16769
3	DACIA	18620	49	380.000000	4749
13	NISSAN	16614	49	339.061224	4250
7	JEEP	1674	5	334.800000	993
16	RENAULT	80764	314	257.210191	5646
20	TOYOTA	6982	41	170.292683	948
17	SEAT	2667	16	166.687500	1267
1	BMW	4758	34	139.941176	2047

	brand	total_cost	no_of_cases	mean	max
22	VOLVO	2	1	2.000000	2

Opel, Lada, Fiat concerns' cars made the highest damage.

Weekday names has to be defined for each day given in the register.

In [23]:
```
from datetime import datetime
accidents['accident_date'] =  pd.to_datetime(accidents['accident_date'], format='%Y-%m-%d')
```

In [26]:
```
accidents['day_of_week'] = accidents['accident_date'].dt.weekday_name
```

In [ ]:
```
##  6) Milyen napon történik a legtöbb baleset?
```

In [28]:
```
step = accidents[['day_of_week', 'cost']]
step = step.groupby(['day_of_week']).count()
step['sum'] = step['cost'].sum()
step ['percent'] = step ['cost'] / step ['sum']
step
```

Out[28]:

day_of_week	cost	sum	percent
Friday	9640	65260	0.147717
Monday	9113	65260	0.139641
Saturday	7414	65260	0.113607
Sunday	7617	65260	0.116718
Thursday	10717	65260	0.164220
Tuesday	10445	65260	0.160052
Wednesday	10314	65260	0.158045

7) Which day of the week there are the most accidents?

In [75]:
```
step = accidents[['day_of_week', 'cost']][accidents.cost > 0]
step = step.groupby(['day_of_week']).agg(['sum', 'count', 'mean', 'max'])
step
```

Out[75]:

	cost			
	sum	**count**	**mean**	**max**
day_of_week				
Friday	200486	325	616.880000	27846
Monday	158201	277	571.122744	29374
Saturday	155773	212	734.778302	17760
Sunday	183472	225	815.431111	56195
Thursday	165803	341	486.225806	16650
Tuesday	336450	349	964.040115	75681
Wednesday	187622	327	573.767584	29253

In the above (non-ordered) list the count value of Tuesday is the highest.

8) Fun fact question: is Friday the 13th the most dangerous day?

Let's compare with other day results as well.

In [80]:
```
df = accidents[(accidents.day == '13') & (accidents.day_of_week == "Friday")]
df = df['cost'] .mean()
df
```

Out[80]:
```
47.20503597122302
```

In [81]:
```
df = accidents[(accidents.day != '13') & (accidents.day_of_week != "Friday")]
df = df['cost'].mean()
df
```

Out[81]: 21.781015510082483

In [82]:
```
df = accidents[(accidents.day == '13') & (accidents.day_of_week == "Friday")]
df = df['cost'] .count()
df
```

Out[82]: 278

In [83]:
```
df = accidents[(accidents.day == '13') & (accidents.day_of_week == "Friday")]
df = df['cost'][df.cost >0] .count()
df
```

Out[83]: 13

In [86]:
```
df = accidents[(accidents.day != '13') & (accidents.day_of_week != "Friday")]
df = df['cost'] .count()
df
```

Out[86]: 53707

In [85]:
```
df = accidents[(accidents.day != '13') & (accidents.day_of_week != "Friday")]
df = df['cost'][df.cost >0] .count()
df
```

Out[85]: 1662

The average cost of accidents on 278 events on 13th Friday was 47.2 (unit), while Fridays, not being the 13th has lower average. Further analysis results may be found in the separate presentation file.

In [ ]:
```

```