dnd

September 7, 2024

0.1 DnD Great Weapon Master Comparison

Helper Functions

[1]:

```python
# Function definitions

import random as rand
import pandas as pd

ROLLS_TO_MAKE = 1_000_000

def roll(die=20):
    result = rand.randrange(1,die + 1)
    return result

def to_hit(dc=10, d20s=1):
    hit = False
    crit = False
    rolls = []
    for i in range(d20s):
        res = roll(20)
        rolls.append(res)
        if res == 20:
            hit = True
            crit = True
        elif res >= dc:
            hit = True

    return(hit, crit, rolls)

def damage(crit=False, sides=8, mod=4, dice=1):
    res = 0
    if crit:
        dice = dice * 2
    for i in range(dice):
        res += roll(sides)
    res += mod
    return res
```

```
def attack(dc=10, advantage=False, sup=False, dmg_die=10, dmg_mod=0,␣
 ↪dmg_dice=1):
    dice_to_roll = 1
    if advantage:
        dice_to_roll = 2
    if sup:
        dice_to_roll = 3

    hit, critical, rolls = to_hit(dc, dice_to_roll)
    if hit:
        dmg = damage(crit=critical, sides=dmg_die, mod=dmg_mod, dice=dmg_dice)
    else:
        dmg = 0

    roll_type = "Normal"
    if advantage and not sup:
        roll_type = "Advantage"
    elif advantage and sup:
        roll_type= "Super"

    return {"rolls": rolls, "hit": hit, "damage": dmg, "crit": critical,␣
 ↪"advantage": advantage, "super_advantage": sup, "roll_type": roll_type}
```

Exploration

2024 Rules

[2]:

```
# 2024 Normal Rolls
new_norm = []
for i in range(ROLLS_TO_MAKE):
    new_norm.append(attack(10, False, False, 8, 7, 1))

new_norm_df = pd.DataFrame(new_norm)
print(f"2024 Normal Rolls {new_norm_df.shape}")
print(f"Damage per roll: {new_norm_df['damage'].mean()}")
print(new_norm_df.head())

# 2024 Advantage Rolls
new_adv = []
for i in range(ROLLS_TO_MAKE):
    new_adv.append(attack(10, True, False, 8, 7, 1))

new_adv_df = pd.DataFrame(new_adv)
print(f"\n2024 Advantage Rolls {new_adv_df.shape}")
print(f"Damage per roll: {new_adv_df['damage'].mean()}")
print(new_adv_df.head())

# 2024 Super Advantage Rolls
```

```
new_sup = []
for i in range(ROLLS_TO_MAKE):
    new_sup.append(attack(10, True, True, 8, 7, 1))

new_sup_df = pd.DataFrame(new_sup)
print(f"\n2024 Super Advantage Rolls {new_sup_df.shape}")
print(f"Damage per roll: {new_sup_df['damage'].mean()}")
print(new_sup_df.head())
```

```
2024 Normal Rolls (1000000, 7)
Damage per roll: 6.547295
  rolls    hit  damage   crit  advantage  super_advantage roll_type
0  [18]   True      11  False      False            False    Normal
1  [12]   True       9  False      False            False    Normal
2   [7]  False       0  False      False            False    Normal
3  [15]   True       8  False      False            False    Normal
4  [13]   True      13  False      False            False    Normal

2024 Advantage Rolls (1000000, 7)
Damage per roll: 9.613134
     rolls   hit  damage   crit  advantage  super_advantage  roll_type
0  [3, 20]  True      20   True       True            False  Advantage
1  [15, 8]  True      10  False       True            False  Advantage
2  [1, 11]  True      11  False       True            False  Advantage
3  [9, 19]  True       9  False       True            False  Advantage
4  [9, 20]  True      15   True       True            False  Advantage

2024 Super Advantage Rolls (1000000, 7)
Damage per roll: 11.096072
          rolls   hit  damage   crit  advantage  super_advantage roll_type
0   [5, 11, 11]  True       9  False       True             True     Super
1  [19, 14, 20]  True      13   True       True             True     Super
2    [12, 6, 4]  True      15  False       True             True     Super
3  [12, 14, 11]  True      11  False       True             True     Super
4    [14, 4, 3]  True       8  False       True             True     Super
```

2014 Rules

[3]:

```
# 2014 Normal Rolls
old_norm = []
for i in range(ROLLS_TO_MAKE):
    old_norm.append(attack(15, False, False, 8, 14, 1))

old_norm_df = pd.DataFrame(old_norm)
print(f"2014 Normal Rolls {old_norm_df.shape}")
print(f"Damage per roll: {old_norm_df['damage'].mean()}")
print(old_norm_df.head())
```

```
# 2024 Advantage Rolls
old_adv = []
for i in range(ROLLS_TO_MAKE):
    old_adv.append(attack(15, True, False, 8, 14, 1))

old_adv_df = pd.DataFrame(old_adv)
print(f"\n2014 Advantage Rolls {old_adv_df.shape}")
print(f"Damage per roll: {old_adv_df['damage'].mean()}")
print(old_adv_df.head())

# 2024 Super Advantage Rolls
old_sup = []
for i in range(ROLLS_TO_MAKE):
    old_sup.append(attack(15, True, True, 8, 14, 1))

old_sup_df = pd.DataFrame(old_sup)
print(f"\n2014 Super Advantage Rolls {old_sup_df.shape}")
print(f"Damage per roll: {old_sup_df['damage'].mean()}")
print(old_sup_df.head())
```

```
2014 Normal Rolls (1000000, 7)
Damage per roll: 5.79152
  rolls    hit  damage   crit  advantage  super_advantage roll_type
0   [1]  False       0  False      False            False    Normal
1  [16]   True      15  False      False            False    Normal
2   [8]  False       0  False      False            False    Normal
3  [17]   True      19  False      False            False    Normal
4  [12]  False       0  False      False            False    Normal

2014 Advantage Rolls (1000000, 7)
Damage per roll: 9.873522
      rolls    hit  damage   crit  advantage  super_advantage  roll_type
0  [18, 18]   True      18  False       True            False  Advantage
1   [9, 14]  False       0  False       True            False  Advantage
2  [16, 11]   True      22  False       True            False  Advantage
3  [16, 18]   True      18  False       True            False  Advantage
4    [8, 4]  False       0  False       True            False  Advantage

2014 Super Advantage Rolls (1000000, 7)
Damage per roll: 12.801061
          rolls    hit  damage   crit  advantage  super_advantage roll_type
0    [6, 11, 2]  False       0  False       True             True     Super
1  [19, 17, 11]   True      21  False       True             True     Super
2    [3, 7, 12]  False       0  False       True             True     Super
3     [3, 5, 1]  False       0  False       True             True     Super
4   [19, 2, 13]   True      21  False       True             True     Super
```

Impressions

[4]:

```
new_norm_hit_perc = new_norm_df[new_norm_df['hit']].size / new_norm_df.size *␣
  ↪100
old_norm_hit_perc = old_norm_df[old_norm_df['hit']].size / old_norm_df.size *␣
  ↪100
print(f"2014 vs 2024 Normal Roll Hit : {old_norm_hit_perc:.2f}% =>␣
  ↪{new_norm_hit_perc:.2f}% \
for {old_norm_df['damage'].mean():.2f} => {new_norm_df['damage'].mean():.2f}␣
  ↪damage.")

new_adv_hit_perc = new_adv_df[new_adv_df['hit']].size / new_adv_df.size * 100
old_adv_hit_perc = old_adv_df[old_adv_df['hit']].size / old_adv_df.size * 100
print(f"2014 vs 2024 Advantage Roll Hit : {old_adv_hit_perc:.2f}% =>␣
  ↪{new_adv_hit_perc:.2f}% \
for {old_adv_df['damage'].mean():.2f} => {new_adv_df['damage'].mean():.2f}␣
  ↪damage.")

new_sup_hit_perc = new_sup_df[new_sup_df['hit']].size / new_sup_df.size * 100
old_sup_hit_perc = old_sup_df[old_norm_df['hit']].size / old_sup_df.size * 100
print(f"2014 vs 2024 Super Advantage Roll Hit : {old_sup_hit_perc:.2f}% =>␣
  ↪{new_sup_hit_perc:.2f}% \
for {old_sup_df['damage'].mean():.2f} => {new_sup_df['damage'].mean():.2f}␣
  ↪damage.")
```

```
2014 vs 2024 Normal Roll Hit : 30.08% => 54.96% for 5.79 => 6.55 damage.
2014 vs 2024 Advantage Roll Hit : 51.01% => 79.76% for 9.87 => 9.61 damage.
2014 vs 2024 Super Advantage Roll Hit : 30.08% => 90.89% for 12.80 => 11.10
damage.
```

Analysis

[5]:

```
# Get Consolidated Data Set
old_frames = [old_norm_df, old_adv_df, old_sup_df]
new_frames = [new_norm_df, new_adv_df, new_sup_df]

old_df = pd.concat(old_frames)
new_df = pd.concat(new_frames)

old_df['rules'] = '2014'
new_df['rules'] = '2024'

all_df = pd.concat([old_df, new_df])
sample_df = all_df.sample(10)
sample_df.head(10)
```

[5]:

```
              rolls    hit  damage   crit  advantage  super_advantage  \
979151         [13]   True       8  False      False            False
998853      [11, 5]  False       0  False       True            False
245747    [1, 5, 4]  False       0  False       True             True
```

```
697574   [6, 9, 16]   True      17  False       True             True
915816  [10, 1, 16]   True      15  False       True             True
487902     [15, 20]   True      21   True       True            False
744960  [14, 19, 6]   True      13  False       True             True
357874         [13]   True      11  False      False            False
137632  [4, 11, 13]  False       0  False       True             True
710463  [6, 11, 12]   True      14  False       True             True

        roll_type rules
979151     Normal  2024
998853  Advantage  2014
245747      Super  2024
697574      Super  2014
915816      Super  2014
487902  Advantage  2014
744960      Super  2024
357874     Normal  2024
137632      Super  2014
710463      Super  2024
```

Viaualizations

[6]:

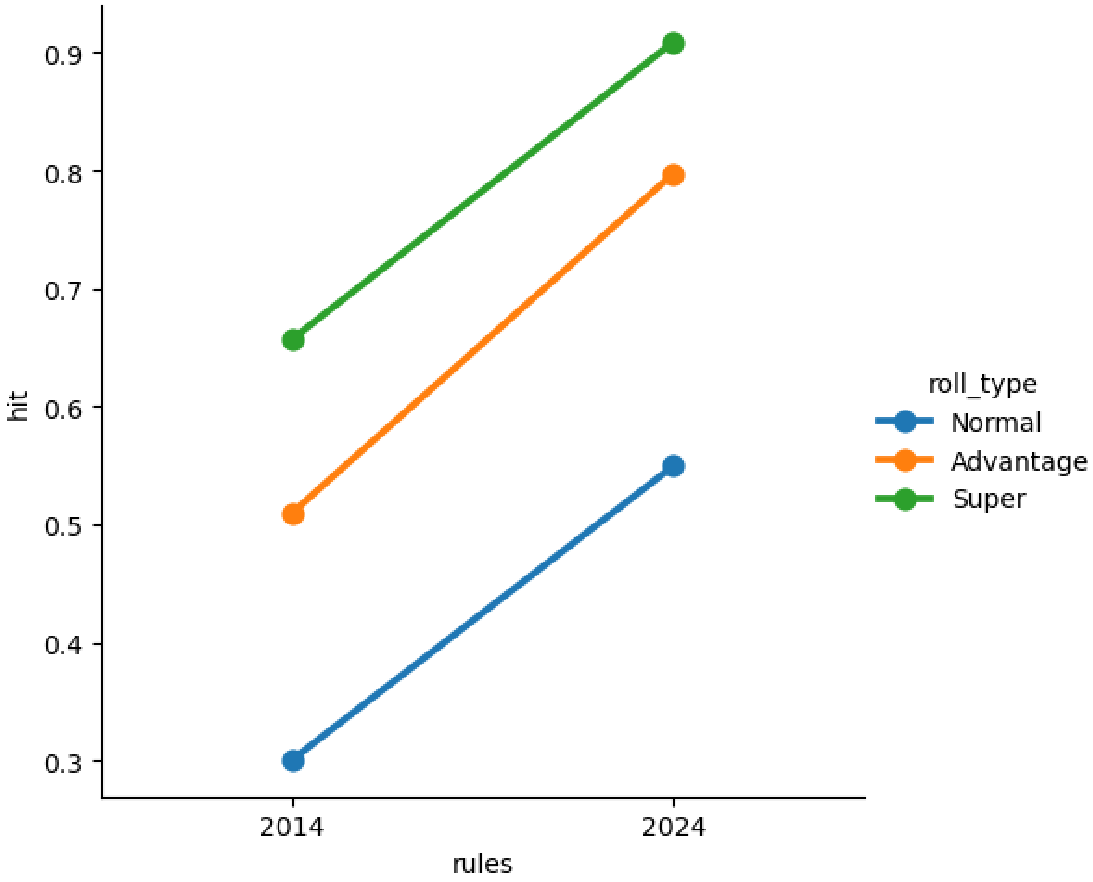
```
# Drops some columns to clean up the DataFrame
#all_df.drop(['advantage', 'super_advantage', 'rolls'], axis=1, inplace=True)
#all_df.sample(10)
```

[7]:

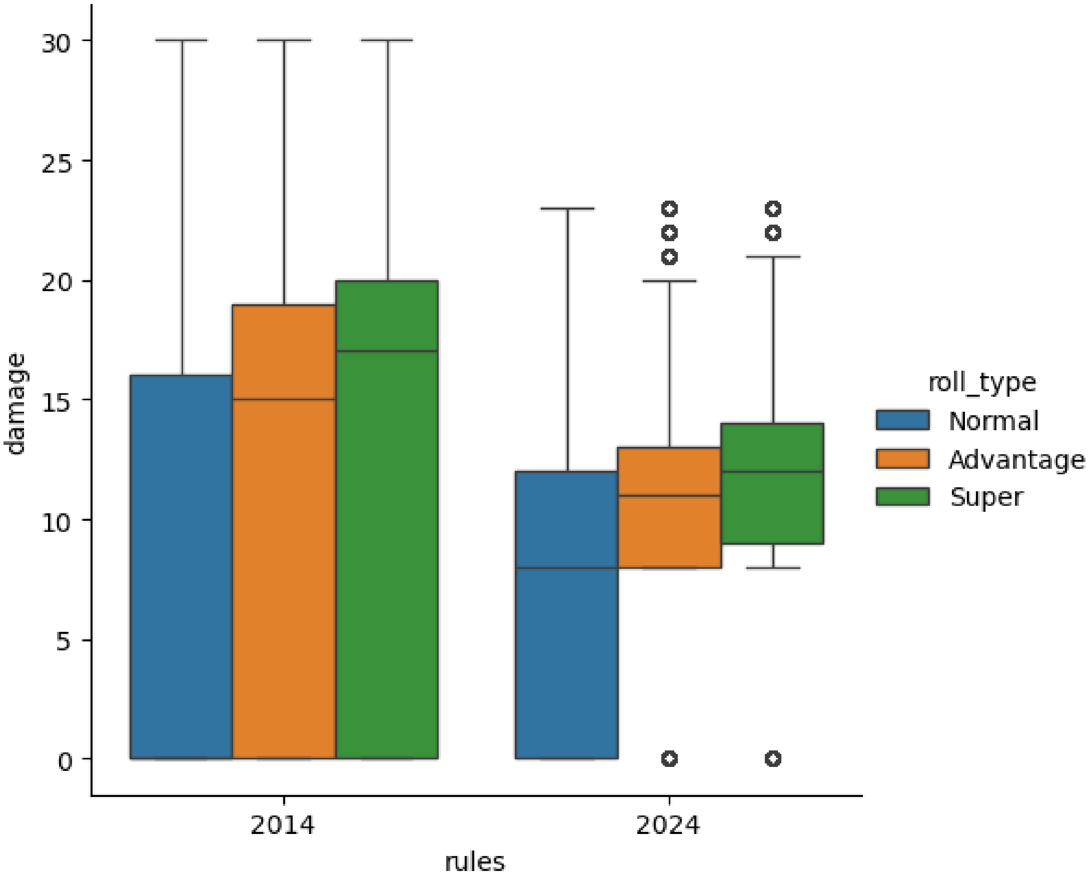
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

cat_1 = sns.catplot(data=all_df, x='rules', y='hit', hue='roll_type',␣
 ↪kind='point')
cat_2 = sns.catplot(data=all_df, x='rules', y='damage', hue='roll_type',␣
 ↪kind='box')

plt.show()
```



0.9
0.8
0.7
0.6
0.5
0.4
0.3
hit
2014
2024
rules
roll_type
Normal
Advantage
Super



[8]:

```
# Setup a couple of new DataFrames to sum the damage
old_df = all_df[all_df['rules'] == '2014'].copy()
old_df['cumulative'] = old_df['damage'].cumsum()
old_df.head()

new_df = all_df[all_df['rules'] == '2024'].copy()
new_df['cumulative'] = new_df['damage'].cumsum()
new_df.head()
```

[8]:

```
  rolls    hit  damage   crit  advantage  super_advantage roll_type rules  \
0  [18]   True      11  False      False            False    Normal  2024
1  [12]   True       9  False      False            False    Normal  2024
2   [7]  False       0  False      False            False    Normal  2024
3  [15]   True       8  False      False            False    Normal  2024
4  [13]   True      13  False      False            False    Normal  2024

   cumulative
0          11
1          20
```

```
2          20
3          28
4          41
```

[9]:

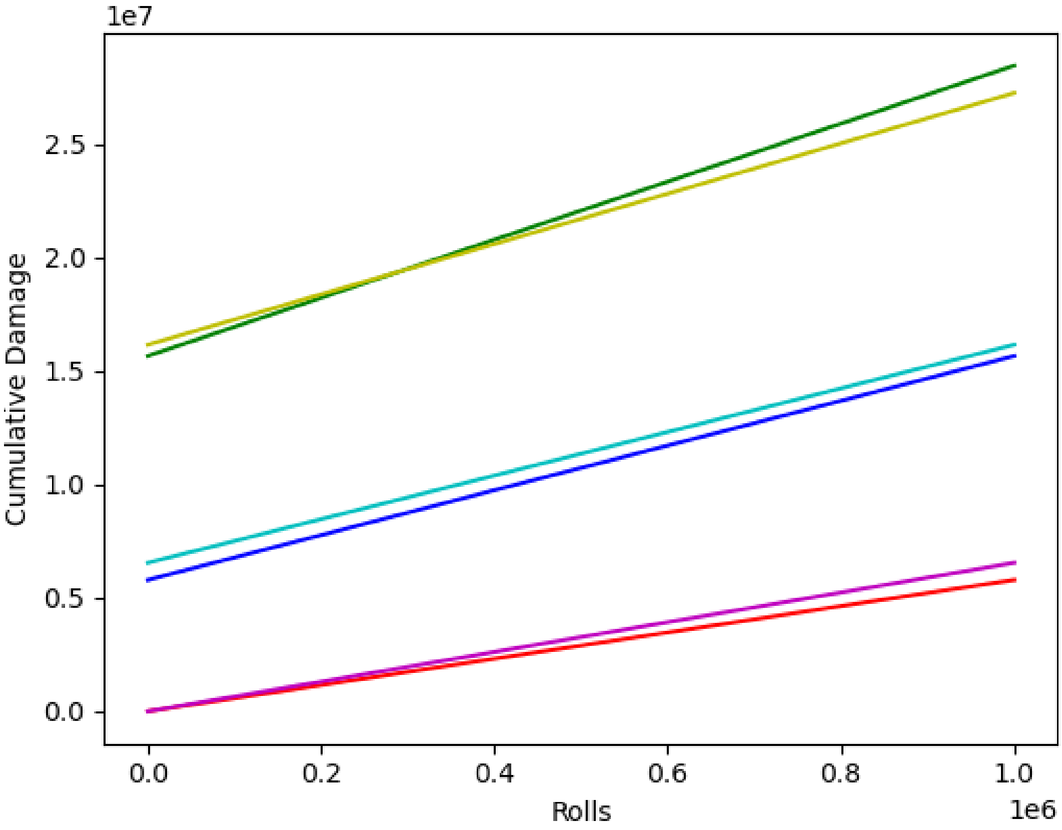
```python
old_norm_dmg = old_df[old_df['roll_type'] == 'Normal']['cumulative']
old_adv_dmg = old_df[old_df['roll_type'] == 'Advantage']['cumulative']
old_sup_dmg = old_df[old_df['roll_type'] == 'Super']['cumulative']

new_norm_dmg = new_df[old_df['roll_type'] == 'Normal']['cumulative']
new_adv_dmg = new_df[old_df['roll_type'] == 'Advantage']['cumulative']
new_sup_dmg = new_df[old_df['roll_type'] == 'Super']['cumulative']

plt.plot(old_norm_dmg, 'r')
plt.plot(old_adv_dmg, 'b')
plt.plot(old_sup_dmg, 'g')

plt.plot(new_norm_dmg, 'm')
plt.plot(new_adv_dmg, 'c')
plt.plot(new_sup_dmg, 'y')

plt.xlabel('Rolls')
plt.ylabel('Cumulative Damage')
plt.show()
```



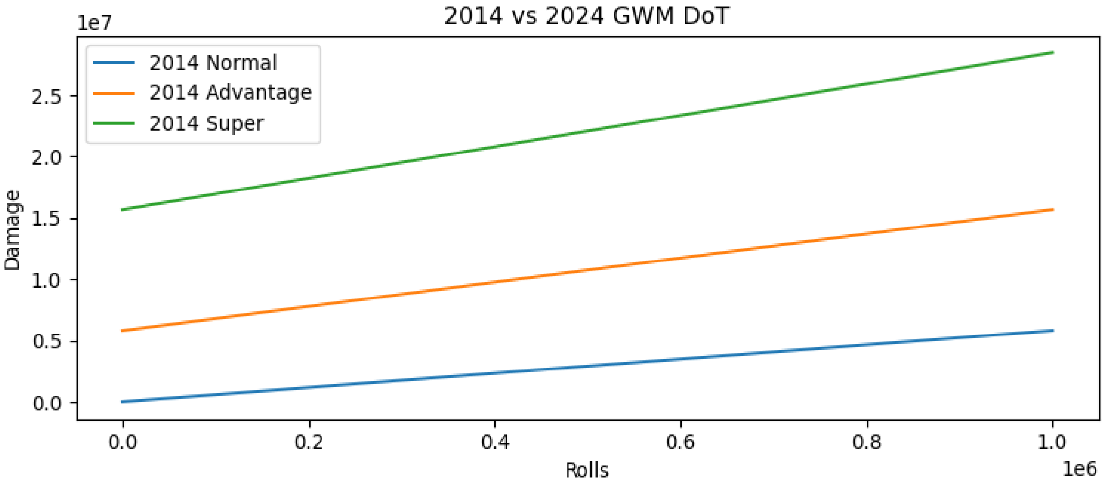
[10]:

```
from cProfile import label

from matplotlib.pyplot import show


fig_1 = plt.figure(figsize=(8,30), dpi=96)
axes_1 = fig_1.add_axes([0.1, 0.1, 0.9,.09]) # Canvas needs a border so bring it in 10%
axes_1.set_xlabel('Rolls')
axes_1.set_ylabel('Damage')
axes_1.set_title('2014 vs 2024 GWM DoT')
axes_1.plot(old_norm_dmg, label='2014 Normal')
axes_1.plot(old_adv_dmg, label='2014 Advantage')
axes_1.plot(old_sup_dmg, label='2014 Super')


axes_1.legend(loc=0)
plt.show()
```


2014 vs 2024 GWM DoT
1e7
2014 Normal
2014 Advantage
2014 Super
Damage
1e6
Rolls