Superheroes

April 1, 2019

1 The Statistics of Superheroes

CHE 116: Numerical Methods and Statistics
Luke Oluoch

1.1 Overview

In recent years, it seems as though media has gone through a superhero renaissance. During the 1990s, which had Marvel Comics filing for bankruptcy and DC Comics relying on sensationalism (stories like the Death of Superman and the Knightfall, wherein Batman gets paralyzed), the medium struggled to stay relevant in the eyes of the public . Resurgence began in the 2000s with a modern, cinematic take on the heroes, but it wasn't until 2007 and 2008 that they became a dominant genre, with the releases of Spider-Man 3, Iron Man, and the Dark Knight. Now each year is accompanied by at least three superhero films, and they consistently rank among the highest grossing movies per year. But how does this affect the sales of the actual comic books where the source material comes from?

The purpose of this project is to explore the relationship between the release of superhero movies and the number of comic book units sold of the company making the movies. The data being used provides number of individual units sold by Marvel Comics and DC Comics each month. All of the information refers to issues distributed to North American comic book vendors given out by Diamond Comics Distributors. In this scenario we will be considering the years 2012 to 2016. We will also count Star Wars: The Force Awakens, because Marvel sells Star Wars Comics and they do so well, they have actually saved the company's profits in the past.

For the first part, a couple of tests will be performed. First, I will take the data of the month after the movie comes out and use a t-test (because there are only 12 months per year) to compare it to the other months that had in theory would have no superhero movie release. Then once that is done, a Wilcoxon Sum of Ranks test will be used to compare the sales between each year to make sure that the data is not erratically different. Wilcoxon will be used because they are using similar data of different sizes, but with the same type of data. The goal of the first test is to see how the the number of sales a company had is truly affected by what happens in their cinematic counterparts. The goal of the second is to see the ensure that the sales are all in the same distribution and to see whether or not certain years do quite different than other years.

For the second part, a linear regression will be performed to see if there is a linear growth that has happened alongside the boom of superhero media

```
In [3]: %matplotlib inline
        import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as ss
import seaborn
seaborn.set_context("poster")
```

2 Part One: Hypothesis testing

2.1 Hypothesis Test: t-Test

The goal of the t-Test is to take a single point of data and see if it is part of a t-distribution that. If it is not, then it is a sign that the data point is unique (in this case since we already know it's part of the distribution). In other words, Each superhero movie month will have its comic book sales compared to the rest of the year, and if it fails the null-hypothesis test, then it's a sign that the movie helped boost sales tremendously. In python, the t - Test is accessed through `scipy.stats.t`.

In this case, the **null hypothesis** is that the movie month data point is from the same distribution as non-movie months. Now, by testing the null hypothesis, a p-value will be obtained, therefore either supporting or disproving the null hypothesis.

Equation:

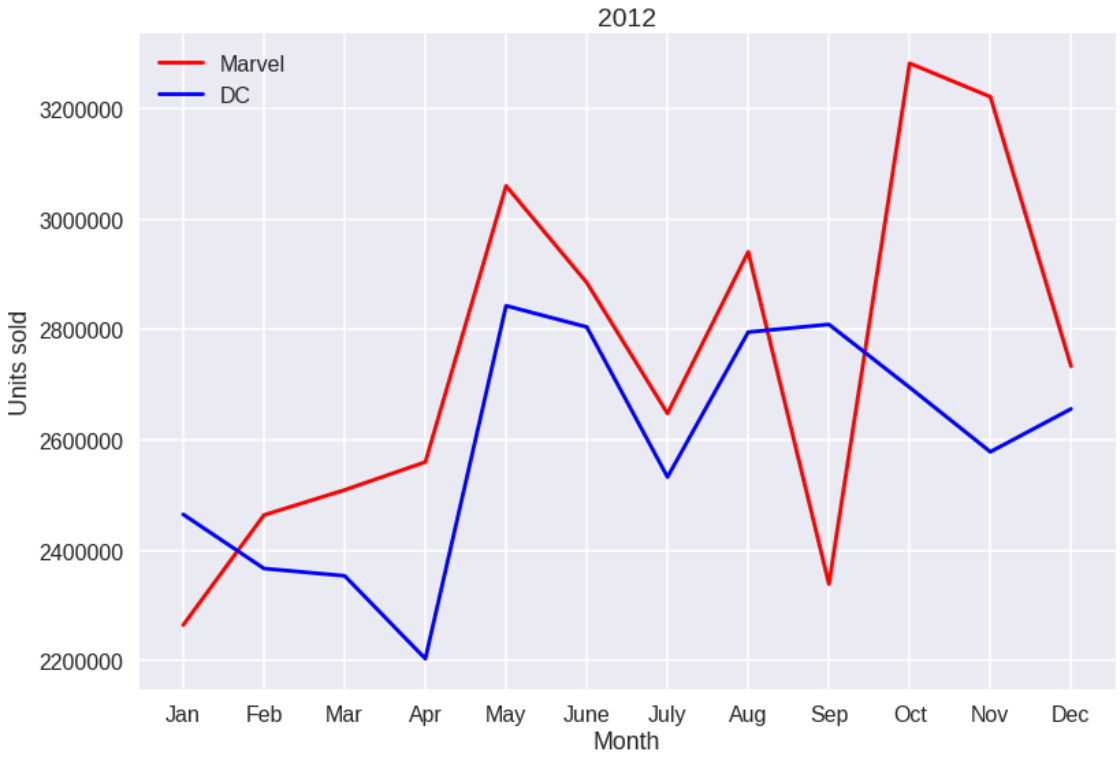
$$t = \frac{Z}{s} = \frac{(\bar{X} - \mu)/(\sigma/\sqrt{n})}{s}$$

The p-value is obtained by comparing t to a standard t-distribution of length n. If the p value is less than 0.05, then the null hypothesis is rejected.

```
In [ ]: #2012: Marvel: Ghostrider (Feb), Avengers(May), Amazing Spider-Man(July). DC: Dark Kni
        #2013: Marvel: Iron Man 3(May), Wolverine(July), Thor 2(Nov). DC: Man of Steel(Jun)
        #2014: Marvel: Cap 2(April), TASM 2(May), X-Men DOFP(May), Guardians of Galaxy (August
        #2015: Marvel: Avengers 2(May),Ant-Man(July), Fant4stic(Aug), StarWars7(Dec)
        #2016: Marvel: Deadpool(Feb), Cap 3(May), X-Men Apocalypse(May), Doctor Strange (Nov)
```

3 2012

```
In [36]: data=pd.read_excel('AllData.xlsx')

         #print(data)

         M2012=np.array(data['Marvel 2012'])
         D2012=np.array(data['DC 2012'])
         plt.plot(range(len(M2012)),M2012,'r', label="Marvel")
         plt.plot(range(len(D2012)),D2012, 'b', label="DC")
         plt.xticks(np.arange(12),('Jan','Feb','Mar','Apr','May','June','July','Aug','Sep','Oct
         plt.xlabel('Month')
         plt.ylabel('Units sold')
         plt.title('2012')
         plt.legend(loc='best')
         plt.show()
         #2012: Marvel: Ghostrider (Feb), Avengers(May), Amazing Spider-Man(July). DC: Dark Kn
```




```
In [50]: Ghostrider=M2012[2:3]
         Avenge=M2012[5:6]
         TASM=M2012[7:8]
         TDKR=D2012[7:8]


         pureM2012=np.hstack((M2012[0:2],M2012[3:5],M2012[6:7],M2012[8:12]))
         pureD2012=np.hstack((D2012[0:7],D2012[8:12]))

         print('Null Hypothesis: Ghost Rider had no massive impact on Marvels sales')
         #GhostRider
         mean=np.mean(pureM2012)
         #print(mean)

         std=np.std(pureM2012)
         #print(std)
         z=(abs(Ghostrider[0]-mean))/std
         pval=2*ss.t.cdf(-z,len(pureM2012))
         print('Ghost Rider\nT: {}\nP: {}'.format(z,pval))

         #Avengers
         print()
         print('Null Hypothesis: Avengers had no massive impact on Marvels sales')
```

```
z2=(Avenge[0]-mean)/std
pval2=2*ss.t.cdf(-z2,len(pureM2012))
print('Avengers\nT: {}\nP: {}'.format(z2,pval2))


#TASM
print()
print('Null Hypothesis: The Amazing Spider-Man had no massive impact on Marvels sales
z3=(TASM[0]-mean)/std
pval3=2*ss.t.cdf(-z3,len(pureM2012))
print('The Amazing Spider-Man\nT: {}\nP: {}'.format(z3,pval3))


#TDKR
print()
mean2=np.mean(pureD2012)
#print(mean2)


std4=np.std(pureD2012)
#print(std4)
print('Null Hypothesis: The Dark Knight Rises had no massive impact on DCs sales')
z4=(TDKR[0]-mean)/std4
pval4=2*ss.t.cdf(-z4,len(pureD2012))
print('The Dark Knight Rises\nT: {}\nP: {}'.format(z4,pval4))
```

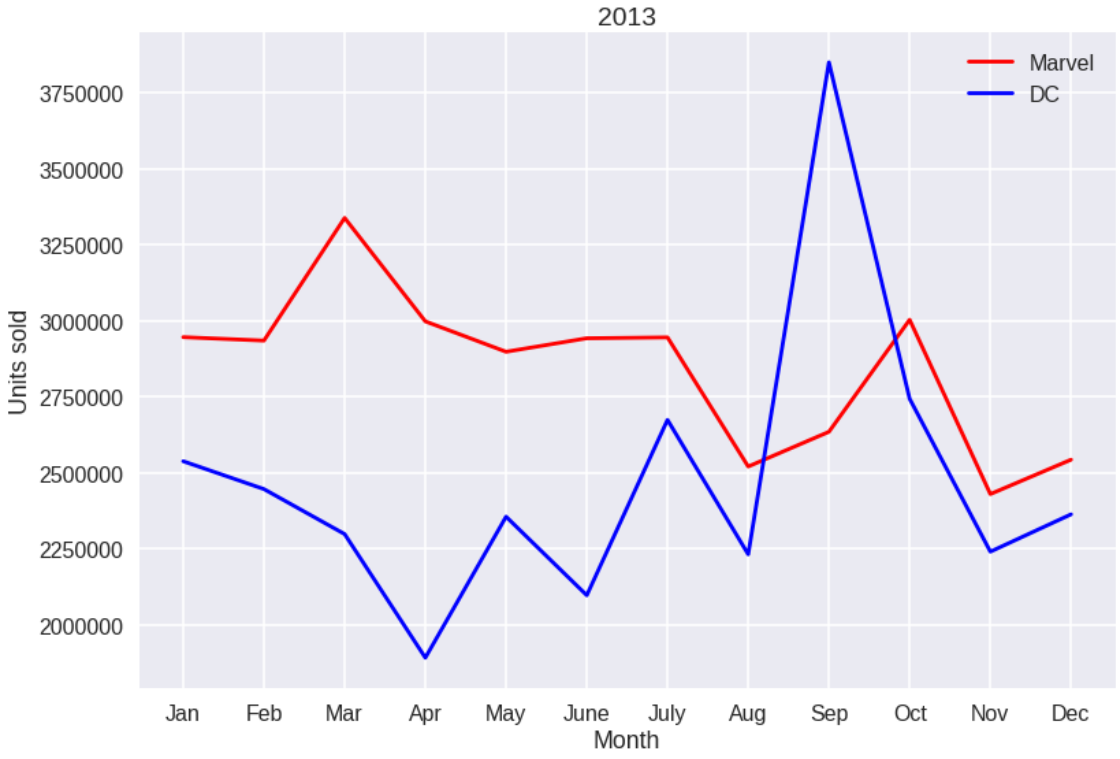
```
Null Hypothesis: Ghost Rider had no massive impact on Marvels sales
Ghost Rider
T: 0.6229758241688508
P: 0.5487590698550218

Null Hypothesis: Avengers had no massive impact on Marvels sales
Avengers
T: 0.4358745457998073
P: 0.6731953315776069

Null Hypothesis: The Amazing Spider-Man had no massive impact on Marvels sales
The Amazing Spider-Man
T: 0.5934050270818033
P: 0.5675217003556514

Null Hypothesis: The Dark Knight Rises had no massive impact on DCs sales
The Dark Knight Rises
T: 0.32382914495768506
P: 0.7521410102278739
```

3.1 2012 Round-Up

Ghost-Rider had a p-value of 0.548, The Avengers had a p-value of 0.673, The Amazing Spider-Man had a p-value of 0.568. It appears as though the Marvel movies of this year don't fail the Null Hypothesis and seem to have **not** had an impact on the sales of their comics. DC shares a similar fate, with the Dark Knight Rises having a p-value of 0.752

4 2013

```
In [38]: M2013=np.array(data['Marvel 2013'])
         D2013=np.array(data['DC 2013'])
         plt.plot(range(len(M2013)),M2013,'r', label="Marvel")
         plt.plot(range(len(D2013)),D2013, 'b', label="DC")
         plt.xticks(np.arange(12),('Jan','Feb','Mar','Apr','May','June','July','Aug','Sep','Oct
         plt.xlabel('Month')
         plt.ylabel('Units sold')
         plt.title('2013')
         plt.legend(loc='best')
         plt.show()
```




```
In [65]: #2013: Marvel: Iron Man 3(May), Wolverine(July), Thor 2(Nov). DC: Man of Steel(Jun)
         IM3=M2013[5:6]
         Snikt=M2013[7:8]
```

```
Thor=M2013[11:12]
Supe=D2013[6:7]


pureM2013=np.hstack((M2013[0+1:4+1],M2013[5+1:6+1],M2013[7+1:10+1]))
pureD2013=np.hstack((D2013[0+1:5+1],D2013[6+1:12+1]))

print('Null Hypothesis: Iron Man 3 had no massive impact on Marvels sales')
#Iron Man 3
mean=np.mean(pureM2013)
#print(mean)

std=np.std(pureM2013)
#print(std)
z=(IM3[0]-mean)/std
pval=2*ss.t.cdf(-z,len(pureM2013))
print('T: {}\nP: {}'.format(z,pval))

#Wolverine
print()
print('Null Hypothesis: The Wolverine had no massive impact on Marvels sales')
z2=(Snikt[0]-mean)/std
pval2=2*ss.t.cdf(z2,len(pureM2013))
print('T: {}\nP: {}'.format(z2,pval2))

#Thor2
print()
print('Null Hypothesis: Thor 2 had no massive impact on Marvels sales')
z3=(Thor[0]-mean)/std
pval3=2*ss.t.cdf(z3,len(pureM2013))
print('T: {}\nP: {}'.format(z3,pval3))

#TDKR
print()
mean2=np.mean(pureD2013)
print(mean2)


std4=np.std(pureD2013)
print(std4)
print('Null Hypothesis: Man of Steel had no massive impact on DCs sales')
z4=(Supe[0]-mean)/std4
pval4=2*ss.t.cdf(z4,len(pureD2013))
print('T: {}\nP: {}'.format(z4,pval4))
```

```
Null Hypothesis: Iron Man 3 had no massive impact on Marvels sales
T: 0.17748083803425868
P: 0.8635421297379777
```

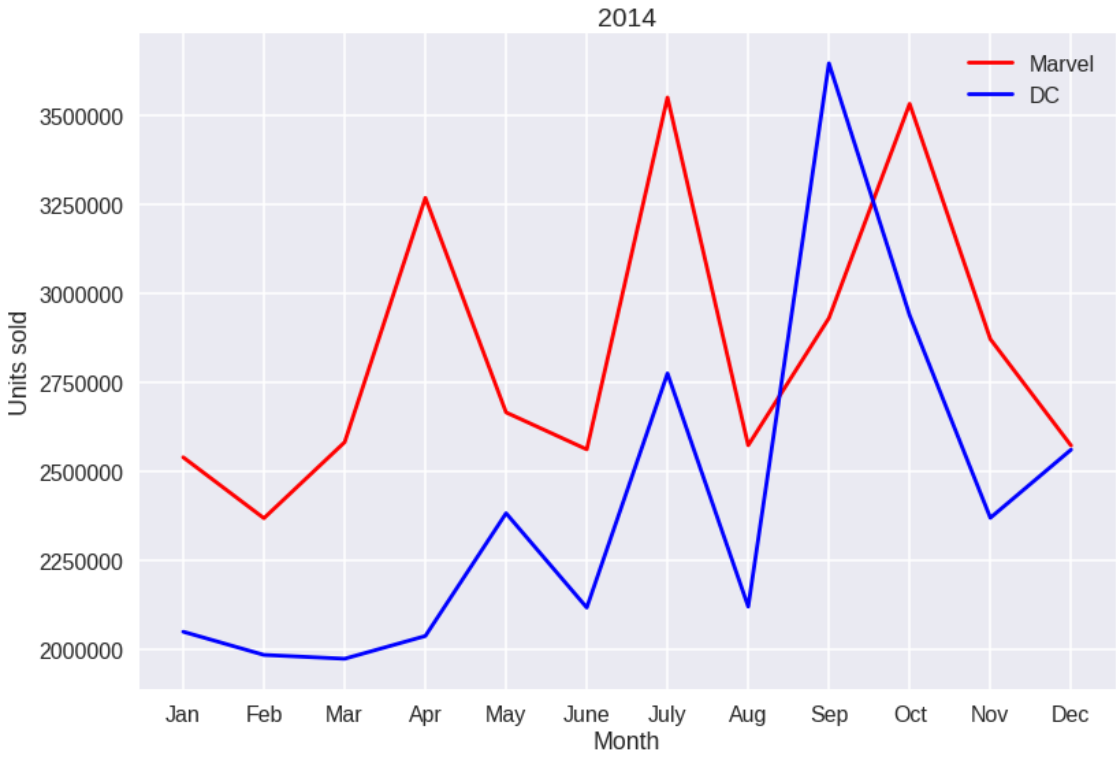
```
Null Hypothesis: The Wolverine had no massive impact on Marvels sales
T: -1.4986733029345005
P: 0.17233995992951845

Null Hypothesis: Thor 2 had no massive impact on Marvels sales
T: -1.4090719240270875
P: 0.19647713416743617

2452291.7
511434.845086
Null Hypothesis: Man of Steel had no massive impact on DCs sales
T: -0.43762685931626133
P: 0.6709575388353938
```

4.1 2013 Round-Up

The year where all the Marvel properties produced were sequels and DC rebooted Superman to start the DC Extended Universe. Man of Steel had a p-value of 0.671, The Wolverine had a p-value of 0.172, and the MCU movies had p-values of 0.864 and 0.196 for Iron Man 3 and Thor: The Dark World, respectively

5 2014

```
In [40]: M2014=np.array(data['Marvel 2014'])
         D2014=np.array(data['DC 2014'])
         plt.plot(range(len(M2014)),M2014,'r', label="Marvel")
         plt.plot(range(len(D2014)),D2014, 'b', label="DC")
         plt.xticks(np.arange(12),('Jan','Feb','Mar','Apr','May','June','July','Aug','Sep','Oct
         plt.xlabel('Month')
         plt.ylabel('Units sold')
         plt.title('2014')
         plt.legend(loc='best')
         plt.show()
```



In [68]:
```
#2014: Marvel: Cap 2(April), TASM 2(May), X-Men DOFP(May), Guardians of Galaxy (Augus
Cap=M2014[3+1:4+1]
TASM2=M2014[4+1:5+1]
DOFP=M2014[4+1:5+1]
Quill=M2014[7+1:8+1]


pureM2014=np.hstack((M2014[0+1:3+1],M2014[5+1:7+1],M2014[7+1:12]))


print('Null Hypothesis: Winter Soldier had no massive impact on Marvels sales')
#Cap2
mean=np.mean(pureM2014)
#print(mean)

std=np.std(pureM2014)
#print(std)
z=(Cap[0]-mean)/std
pval=2*ss.t.cdf(z,len(pureM2014))
print('T: {}\nP: {}'.format(z,pval))

#Wolverine
print()
```

```
print('Null Hypothesis: Amazing Spider-Man 2 had no massive impact on Marvels sales')
z2=(TASM2[0]-mean)/std
pval2=2*ss.t.cdf(z2,len(pureM2014))
print('T: {}\nP: {}'.format(z2,pval2))

#Thor2
print()
print('Null Hypothesis: Days of Future Past had no massive impact on Marvels sales')
z3=(DOFP[0]-mean)/std
pval3=2*ss.t.cdf(z3,len(pureM2014))
print('T: {}\nP: {}'.format(z3,pval3))


print()
print('Null Hypothesis: Guardians of the Galaxy had no massive impact on Marvels sales
z4=(Quill[0]-mean)/std
pval4=2*ss.t.cdf(-z4,len(pureM2014))
print('T: {}\nP: {}'.format(z4,pval4))
```

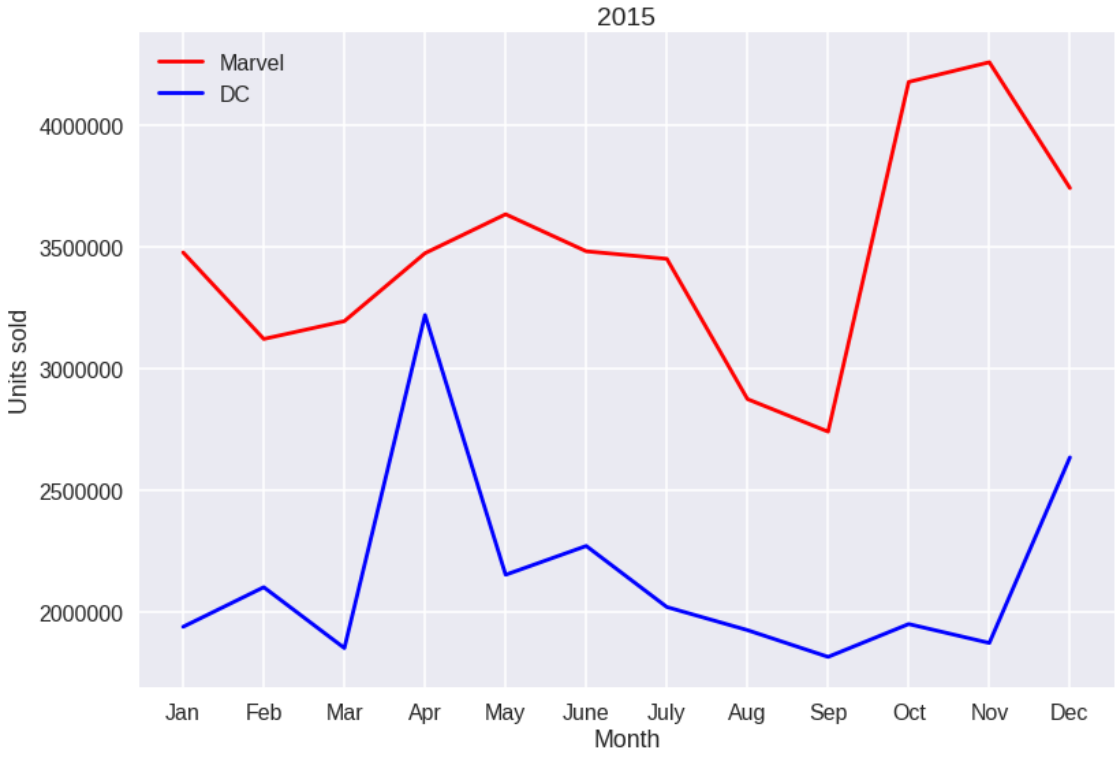
```
Null Hypothesis: Winter Soldier had no massive impact on Marvels sales
T: -0.6035159806965158
P: 0.5610658534224425

Null Hypothesis: Amazing Spider-Man 2 had no massive impact on Marvels sales
T: -0.8534453948075873
P: 0.4155482681203807

Null Hypothesis: Days of Future Past had no massive impact on Marvels sales
T: -0.8534453948075873
P: 0.4155482681203807

Null Hypothesis: Guardians of the Galaxy had no massive impact on Marvels sales
T: 0.033602024937097345
P: 0.9739279339970002
```

5.1 2014 Round-Up

A year without a DC release, so let's see how Marvel faired alone. Captain America: The Winter Soldier seemed to have not had a large impact on Marvel's sales as we can't reject the hypothesis test with it's p-value of 0.561. The Amazing Spider 2 and X-Men: Days of Future Past came out during the same month, but it seems they did not mobilize fans to come out and by, as it saw the sales go down in the next month. This was'nt a large impact though, as they shared a p-value of 0.4155. The sleeper hit, Guardians of the Galaxy also failed to have a large impact on the comic industry as it has a p-value of 0.974. The graph shows however that DC did have a spike after Guardians came out, which is interesting to note.

6 2015

In [42]:
```
M2015=np.array(data['Marvel 2015'])
D2015=np.array(data['DC 2015'])
plt.plot(range(len(M2015)),M2015,'r', label="Marvel")
plt.plot(range(len(D2015)),D2015, 'b', label="DC")
plt.xticks(np.arange(12),('Jan','Feb','Mar','Apr','May','June','July','Aug','Sep','Oct
plt.xlabel('Month')
plt.ylabel('Units sold')
plt.title('2015')
plt.legend(loc='best')
plt.show()
```



In [69]:
```
#2015: Marvel: Avengers 2(May),Ant-Man(July), Fant4stic(Aug), StarWars7(Dec)
M2016=np.array(data['Marvel 2016'])

Ultron=M2015[3+1:4+1]
Pym=M2015[6+1:7+1]
Doom=M2015[7+1:8+1]
Rey=M2016[1:2] #We will check Star Wars with January of 2016


pureM2015=np.hstack((M2015[0+1:3+1],M2015[4+1:6+1],M2015[8+1:11+1]))
```

```
print('Null Hypothesis: Age of Ultron had no massive impact on Marvels sales')
#Cap2
mean=np.mean(pureM2015)
#print(mean)

std=np.std(pureM2015)
#print(std)
z=(Ultron[0]-mean)/std
pval=2*ss.t.cdf(-z,len(pureM2015))
print('T: {}\nP: {}'.format(z,pval))

#Wolverine
print()
print('Null Hypothesis: Ant-Man had no massive impact on Marvels sales')
z2=(Pym[0]-mean)/std
pval2=2*ss.t.cdf(z2,len(pureM2015))
print('T: {}\nP: {}'.format(z2,pval2))

#Thor2
print()
print('Null Hypothesis: Fantastic Four had no massive impact on Marvels sales')
z3=(Doom[0]-mean)/std
pval3=2*ss.t.cdf(z3,len(pureM2015))
print('T: {}\nP: {}'.format(z3,pval3))


print()
print('Null Hypothesis: The Force Awakens had no massive impact on Marvels sales')
z4=(Rey[0]-mean)/std
pval4=2*ss.t.cdf(z4,len(pureM2015))
print('T: {}\nP: {}'.format(z4,pval4))
```

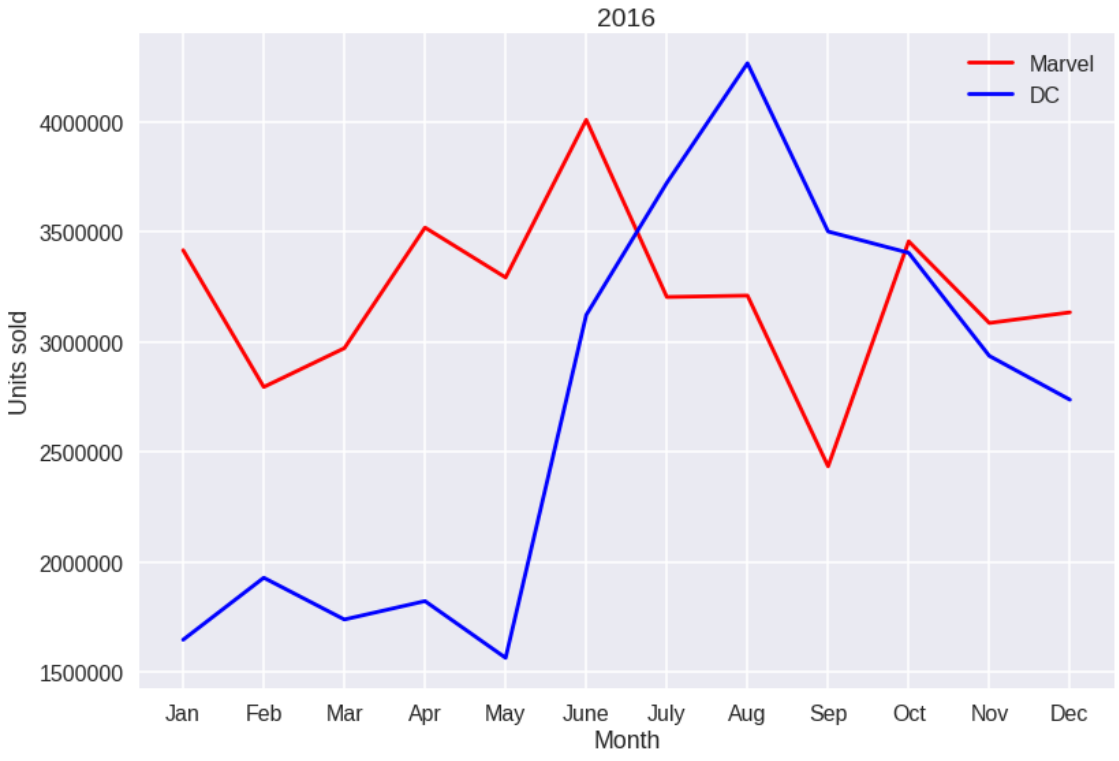
```
Null Hypothesis: Age of Ultron had no massive impact on Marvels sales
T: 0.054497724058251235
P: 0.9578750239785352

Null Hypothesis: Ant-Man had no massive impact on Marvels sales
T: -1.8811688723911995
P: 0.09673145313092209

Null Hypothesis: Fantastic Four had no massive impact on Marvels sales
T: -2.221824983772612
P: 0.057019550244482825

Null Hypothesis: The Force Awakens had no massive impact on Marvels sales
```

```
T: -2.0811147566510337
P: 0.07099205454808104
```

7 2015 Round-Up

The year Star-Wars came back to the silver screen. How did it impact Marvel's sales? Pretty decently, however, with a p-value of 0.071, The Force Awakens still doesn't fail the Null Hypothesis Test. Ant-Man and Fantastic Four came close to as well, scoring p-values of 0.097 and 0.057, and oddly enough, corresponding with the year's weakest sales from Marvel. The MCU's block-buster Avengers: Age of Ultron had a p-value of 0.958, so it also had no large impact on the comic book sales.

```
In [44]: M2016=np.array(data['Marvel 2016'])
         D2016=np.array(data['DC 2016'])
         plt.plot(range(len(M2016)),M2016,'r', label="Marvel")
         plt.plot(range(len(D2016)),D2016, 'b', label="DC")
         plt.xticks(np.arange(12),('Jan','Feb','Mar','Apr','May','June','July','Aug','Sep','Oct
         plt.xlabel('Month')
         plt.ylabel('Units sold')
         plt.title('2016')
         plt.legend(loc='best')
         plt.show()
```



8 2016

In [72]:
```
#2016: Marvel: Deadpool(Feb), Cap 3(May), X-Men Apocalypse(May), Doctor Strange (Nov)
Wade=M2016[1+1:2+1]
Cap3=M2016[4+1:5+1]
Xmen=M2016[4+1:5+1]
DocS=M2016[10+1:11+1]
BVS=D2016[2+1:3+1]
SS=D2016[7+1:8+1]

pureM2016=np.hstack((M2016[0+1:1+1],M2014[2+1:4+1],M2012[5+1:10+1]))
pureD2016=np.hstack((D2016[0+1:2+1],D2016[3+1:7+1],D2016[7+1:12]))


print('Null Hypothesis: Deadpool had no massive impact on Marvels sales')
#Deadpool
mean=np.mean(pureM2016)
#print(mean)

std=np.std(pureM2016)
#print(std)
z=(Wade[0]-mean)/std
pval=2*ss.t.cdf(-z,len(pureM2016))
print('T: {}\nP: {}'.format(z,pval))

#Civil War
print()
print('Null Hypothesis: Civil War had no massive impact on Marvels sales')
z2=(Cap3[0]-mean)/std
pval2=2*ss.t.cdf(-z2,len(pureM2016))
print('T: {}\nP: {}'.format(z2,pval2))

#AgeofA
print()
print('Null Hypothesis: Age of Apocalypse had no massive impact on Marvels sales')
z3=(Xmen[0]-mean)/std
pval3=2*ss.t.cdf(-z3,len(pureM2016))
print('T: {}\nP: {}'.format(z3,pval3))


print()
print('Null Hypothesis: Doctor Strange had no massive impact on Marvels sales')
z4=(DocS[0]-mean)/std
pval4=2*ss.t.cdf(-z4,len(pureM2016))
print('T: {}\nP: {}'.format(z4,pval4))


mean2=np.mean(pureD2016)
```

```
print(mean2)
print('Null Hypothesis: BvS had no massive impact on DCs sales')
std2=np.std(pureD2016)
print(std2)
z6=(BVS[0]-mean2)/std2
pval6=2*ss.t.cdf(z6,len(pureD2016))
print('T: {}\nP: {}'.format(z6,pval6))
print()
print('Null Hypothesis: Suicide Squad had no massive impact on DCs sales')

z7=(SS[0]-mean2)/std2
pval7=2*ss.t.cdf(-z7,len(pureD2016))
print('T: {}\nP: {}'.format(z7,pval7))
```

```
Null Hypothesis: Deadpool had no massive impact on Marvels sales
T: 0.2406699460014454
P: 0.8158639355142541

Null Hypothesis: Civil War had no massive impact on Marvels sales
T: 3.4608125196092723
P: 0.008557555133180387

Null Hypothesis: Age of Apocalypse had no massive impact on Marvels sales
T: 3.4608125196092723
P: 0.008557555133180387

Null Hypothesis: Doctor Strange had no massive impact on Marvels sales
T: 0.7459377170287573
P: 0.47704895330743724
2893217.9
Null Hypothesis: BvS had no massive impact on DCs sales
855079.163
T: -1.251188131221435
P: 0.2393448207283938

Null Hypothesis: Suicide Squad had no massive impact on DCs sales
T: 0.7125037380898083
P: 0.49244242685216333
```

8.1 2016 Round-Up

This was the year with some of the highest grossing movies. Deadpool had a p-value of 0.816, Doctor Strange had a value of 0.239, Batman v. Superman: Dawn of Justice had one of 0.239, and Suicide Squad had one of 0.492. All of them didn't fail the null hypothesis, however, X-Men: Apocalypse and Captain America: Civil War both came out the same month, and gave rise to the next month having a p-value of 0.009! Below the threshold of 0.05, these two movies fail the Null Hypothesis. It looks like it took these two heavy hitters to release simultaneously to have a large

impact on comic sales!

8.1.1 Of the 22 movies that came out in the past five years, only X-Men: Apocalypse and Captain America: Civil War had the impact to strongly influence comic book sales.

9 Hypothesis Test: Wilxocon Signed Ranks

This test is one that takes pairs of data and compares them to each other to test whether or not they are in the same distribution. This one is used as opposed to the sum of ranks test, as they both come from the same source (the comic book industry) and both have the same amount of entries (1 point of summed up dales per month). The Null hypothesis that accompanies th signed Ranks test is that they are from the same distribution. Get a p-value below 0.05, and the two data sets are from different distributions.

In [123]:
```
M_Match=[]

M_Match=np.append(M_Match,ss.wilcoxon(M2012, M2013).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2013, M2014).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2014, M2015).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2015, M2016).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2012, M2016).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2013, M2016).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2014, M2016).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2015, M2012).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2015, M2013).pvalue)
M_Match=np.append(M_Match,ss.wilcoxon(M2012, M2014).pvalue)




# multiple retests to make sure there are no surprises

for i in M_Match:
    if(i>0.05):
        print(i,'True. In distribution')
    else:
        print(i,'False')
```

```
0.346521711706 True. In distribution
0.937472922362 True. In distribution
0.00474176803841 False
0.084379442594 True. In distribution
0.00370174940669 False
0.0120633234189 False
0.0341704726922 False
0.00287341392785 False
0.00370174940669 False
0.753683717726 True. In distribution
```

It appears that for Marvel, the years 2012, 2013, and 2014 are in the same distribution, however, 2015 and 2016 are in a distribution of their own, and looking at the graphs, it looks like their distibution is more highest in in sales.

```
In [124]: D_Match=[]

          D_Match=np.append(D_Match,ss.wilcoxon(D2012, D2013).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2013, D2014).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2014, D2015).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2015, D2016).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2012, D2016).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2013, D2016).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2014, D2016).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2015, D2012).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2014, D2012).pvalue)
          D_Match=np.append(D_Match,ss.wilcoxon(D2015, D2013).pvalue)


          # multiple retests to make sure there are no surprises

          for i in D_Match:
              if(i>0.05):
                  print(i,'True. In distribution')
              else:
                  print(i,'False')
```

```
0.272095182736 True. In distribution
0.582919547269 True. In distribution
0.136097381114 True. In distribution
0.182338354181 True. In distribution
0.753683717726 True. In distribution
0.432767580668 True. In distribution
0.346521711706 True. In distribution
0.0341704726922 False
0.1579393105 True. In distribution
0.0597390154554 True. In distribution
```

DC tends to have it be more similar in terms of distribution, with the only year-pair that failed the Signed-Rank test sharing no similar distribution being 2015 and 2012

10 Part 2: Linear Regression

As we have seen, a big success at the box office doesn't necessarily mean a direct and immediate success in the comic book business. However, have the movies cause the mediums to grow overall? To find out, a linear regression can be performed and an R^2 value can be extracted to see whether or not there is an upwards trend, and if there is, how well it fits the data.

The Linear Regression technique that will be used here will be the Ordinary Least Squares Regression in 1D. Here we will attempt to model the equation:

$$y = \alpha + \beta x + \epsilon$$

where $\hat{\beta}$ is an approximation of β calculated by

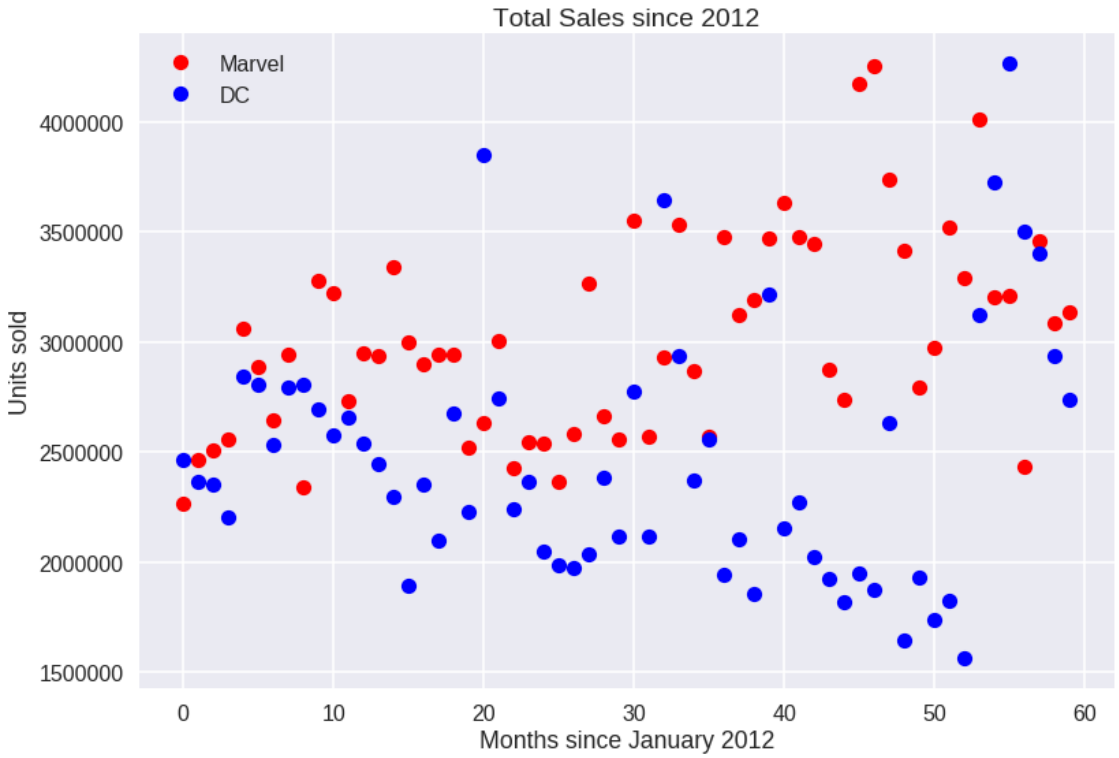
$$\hat{\beta} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{\sigma_{xy}}{\sigma_x^2}$$

(σ_{xy} is the sample covariance of x and y and σ_x^2 is the sample variance of x)
and $\hat{\alpha}$ is and approximation of α calculated by

$$\hat{\alpha} = \frac{1}{N} \sum_i (y_i - \hat{\beta} x_i)$$

This will be done twice, with x representing the amount of months since January 2012 and y representing the sales of the respective companies

```
In [22]: Marvel=np.hstack((M2012,M2013,M2014,M2015,M2016))
         DC=np.hstack((D2012,D2013,D2014,D2015,D2016))
         plt.plot(range(len(Marvel)),Marvel,'ro', label="Marvel")
         plt.plot(range(len(DC)),DC,'bo', label="DC")
         plt.legend(loc='best')
         plt.xlabel('Months since January 2012')
         plt.ylabel('Units sold')
         plt.title('Total Sales since 2012')
         plt.show()
```



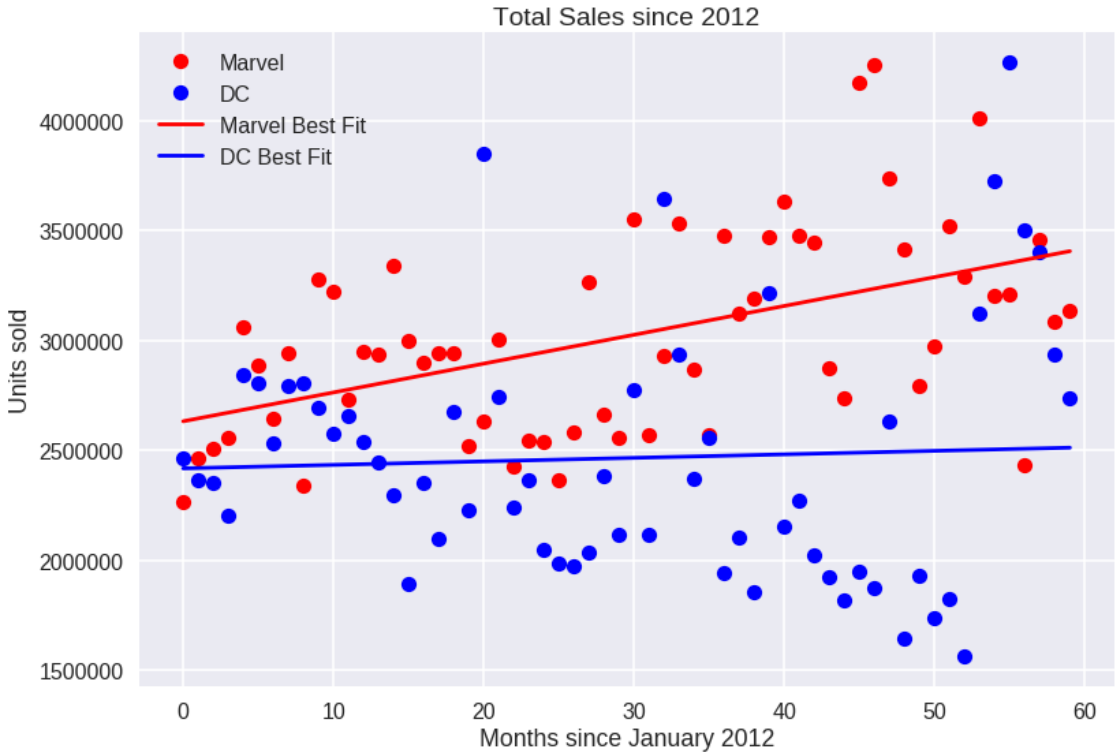
First, a Shapiro-Wilk Test to confirm that the data is not normally distributed. This is good, so that we know that if it is able to be fit normally, we don't have to form a linear relationship. The Null Hypothesis is that the samples are from an unknown normal distribution. We shall try and prove that wrong to continue on.

```
In [125]: print(ss.shapiro(Marvel))
          print(ss.shapiro(DC))
```

```
(0.9598690867424011, 0.04636183753609657)
(0.9300827383995056, 0.00200202246196568)
```

The Marvel Data has a p-value of 0.046 and DC has one of 0.002, so we can reject the null hypothesis and eliminate the possibility of these being normal distributions.

```
In [130]: Month=np.linspace(1,60,60)
          cov1 = np.cov(Month,Marvel, ddof=2)
          cov2 = np.cov(Month,DC,ddof=2)
          beta_hat_M = cov1[0,1] / cov1[0,0]
          beta_hat_D = cov2[0,1] / cov2[0,0]
          alpha_hat_M = np.mean( Marvel - beta_hat_M * Month)
          alpha_hat_D = np.mean( DC - beta_hat_D * Month)


          plt.plot(range(len(Marvel)),Marvel,'ro', label="Marvel")
          plt.plot(range(len(DC)),DC,'bo', label="DC")
          plt.plot(range(len(Marvel)),alpha_hat_M + beta_hat_M * Month,'r', label="Marvel Best
          plt.plot(range(len(DC)),alpha_hat_D + beta_hat_D * Month,'b', label="DC Best Fit")
          plt.xlabel('Months since January 2012')
          plt.ylabel('Units sold')
          plt.title('Total Sales since 2012')
          plt.legend(loc='best')
          plt.show()

          print ('For Marvel: y= ',alpha_hat_M,' + ', beta_hat_M,'x')
          print ('For DC: y= ',alpha_hat_D,' + ', beta_hat_D,'x')
```




```
For Marvel: y=  2620077.14237  +  13117.5297583 x
For DC: y=  2417406.91469  +  1595.77929981 x
```

Marvel had $y = 2620077.14237 + 13117.5297583x$
DC had $y = 2417406.91469 + 1595.77929981x$

According to linear regression, they seem to be gradually selling more comics as time goes on. But how accurate are these lines to their true data counterparts? An R-Squared test and error determination is needed here. R-Squared works by taking the SSR or the sum of square residuals and divide it over the total sum of squares. It's equation is this:

$$R^2 = 1 - \frac{\text{SSR}}{\text{TSS}} = 1 - \frac{\sum_i (\hat{y}_i - y)^2}{\sum_i (\bar{y} - y)^2}$$

The answer is subtracted from 1 so that 1 can represent a perfect fit and 0 can represent no fit at all.

The error for all terms are calculated by

$$S_\epsilon^2 = \frac{\sigma_\epsilon^2}{N - D} = \frac{1}{N - D} \sum_i (\hat{y}_i - y_i)^2$$

$$S_\alpha^2 = S_\epsilon^2 \left[\frac{1}{N - D} + \frac{\bar{x}^2}{\sum_i (x_i - \bar{x})^2} \right]$$

$$S_\beta^2 = \frac{S_\epsilon^2}{\sum_i (x_i - \bar{x})^2}$$

where $S^2_{epsilon}$ is the square of the noise

In [137]:
```
ssrM = np.sum((Marvel - alpha_hat_M - beta_hat_M * Month)**2)
tssM = np.sum((np.mean(Marvel) - Marvel)**2)
rsqM = 1 - ssrM / tssM

ssrD = np.sum((DC - alpha_hat_D - beta_hat_D * Month)**2)
tssD = np.sum((np.mean(DC) - DC)**2)
rsqD = 1 - ssrD / tssD

print(rsqM,rsqD)

df = len(Month) - 2
s2_epsilonM = np.sum((Marvel - alpha_hat_M - beta_hat_M * Month) ** 2) / df
s2_alphaM =  s2_epsilonM * (1. / df + np.mean(Month) ** 2 / (np.sum((np.mean(Month)

print('The standard error for the intercept of Marvel is', np.sqrt(s2_alphaM))



s2_epsilonD = np.sum((DC- alpha_hat_D - beta_hat_D * Month) ** 2) / df
s2_alphaD =  s2_epsilonD * (1. / df + np.mean(Month) ** 2 / (np.sum((np.mean(Month)

print('The standard error for the intercept of DC is', np.sqrt(s2_alphaD))

s2_betaM = s2_epsilonM / np.sum((Month - np.mean(Month))**2)
T = ss.t.ppf(0.975, len(Month) - 2)



print('The error margin for beta for Marvel is plus or minus', T * np.sqrt(s2_betaM)

s2_betaD = s2_epsilonD / np.sum((Month - np.mean(Month))**2)
T = ss.t.ppf(0.975, len(Month) - 2)



print('The error margin for beta for DC is plus or minus', T * np.sqrt(s2_betaD), ' w
```

```
0.25089186788 0.00236092375179
The standard error for the intercept of Marvel is 104825.494086
The standard error for the intercept of DC is 151706.597777
The error margin for beta for Marvel is plus or minus 5957.57896012  with 95% confidence
The error margin for beta for DC is plus or minus 8621.98688313  with 95% confidence
```

Oh no! It looks like Marvel's equation is barely a fit, being 25% accurate, and DC's is completely terrible, being close to zero! The sales are too erratic and inconsistent to be able to plot a reliable increase. And looking at the error margins, on can see why. They are very large and have to incorporate all the crazy changes in sales.

11 In Conclusion

What we learned from the first test is that our off 22 movies, the biggest influence that the superhero movie craze could have on the sales only came from the release of the third Captain America and X-Men: Apocalypse. Otherwise, there was no large connection between them. This shows that the comic book culture and the movie watching culture are two separate beasts. Comic book publishers can't rely on the popularity of their movies, but rather on their own written stories and niche fanbases. The sum of ranks test showed that comic book sales aren't always going to have the same kind of sales year after year. You can have surges and droughts that appear out of seemingly nowhere that causes the sales figures of one year to look alien compared to another.

Lastly, the final conclusion to be drawn is that the month to month and year to year sales of comics are erratic, spontaneous, and do not fit well from a linear perspective. Sales look more akin to a persons heart beat, then a steady increase, and while they are growing in popularity, that does not stop the sheer variety of sales that comes and goes in the industry.

Bibliography

Miller, John J. "Comic Book Sales by Month." Comichron: Comic Book Sales by Month. Amazon Services, Apr. 2017. Web. 01 May 2017. http://www.comichron.com/monthlycomicssales.html.

Staff, Beat, Todd Allen, and Heidi MacDonald. "Home." The Beat. N.p., 02 May 2017. Web. 03 May 2017. http://www.comicsbeat.com/category/sales-charts/.

```
In [141]: class GIFError(Exception): pass

          def get_gif_num_frames(filename):
              frames = 0
              with open(filename, 'rb') as f:
                  if f.read(6) not in ('GIF87a', 'GIF89a'):
                      raise GIFError('not a valid GIF file')
                  f.seek(4, 1)
                  def skip_color_table(flags):
                      if flags & 0x80: f.seek(3 << ((flags & 7) + 1), 1)
                  flags = ord(f.read(1))
                  f.seek(2, 1)
                  skip_color_table(flags)
                  while True:
                      block = f.read(1)
                      if block == ';': break
                      if block == '!': f.seek(1, 1)
                      elif block == ',':
                          frames += 1
                          f.seek(8, 1)
                          skip_color_table(ord(f.read(1)))
```

```
                    f.seek(1, 1)
                else: raise GIFError('unknown block type')
                while True:
                    l = ord(f.read(1))
                    if not l: break
                    f.seek(l, 1)
            return frames
```

In [148]:
```
import tkinter

root = tkinter.Tk()
canvas = tkinter.Canvas(root)
canvas.grid(row = 0, column = 0)
photo = tkinter.PhotoImage(file = './test.gif')
canvas.create_image(0, 0, image=photo)
root.mainloop()
```

```
---------------------------------------------------------------------------

TclError                                    Traceback (most recent call last)

<ipython-input-148-6e77c9ea4784> in <module>()
  1 import tkinter
  2
----> 3 root = tkinter.Tk()
  4 canvas = tkinter.Canvas(root)
  5 canvas.grid(row = 0, column = 0)


/opt/conda/lib/python3.5/tkinter/__init__.py in __init__(self, screenName, baseName, cl
1875                 baseName = baseName + ext
1876         interactive = 0
-> 1877         self.tk = _tkinter.create(screenName, baseName, className, interactive, wan
1878         if useTk:
1879             self._loadtk()


TclError: no display name and no $DISPLAY environment variable
```

In [149]:
```
from __future__ import division # Just because division doesn't work right in 2.7.4
from tkinter import *
from PIL import Image,ImageTk
import threading
from time import sleep

def anim_gif(name):
```

```
        ## Returns { 'frames', 'delay', 'loc', 'len' }
        im = Image.open(name)
        gif = { 'frames': [],
                'delay': 100,
                'loc' : 0,
                'len' : 0 }
        pics = []
        try:
            while True:
                pics.append(im.copy())
                im.seek(len(pics))
        except EOFError: pass

        temp = pics[0].convert('RGBA')
        gif['frames'] = [ImageTk.PhotoImage(temp)]
        temp = pics[0]
        for item in pics[1:]:
            temp.paste(item)
            gif['frames'].append(ImageTk.PhotoImage(temp.convert('RGBA')))

        try: gif['delay'] = im.info['duration']
        except: pass
        gif['len'] = len(gif['frames'])
        return gif

    def ratio(a,b):
        if b < a: d,c = a,b
        else: c,d = a,b
        if b == a: return 1,1
        for i in reversed(xrange(2,int(round(a / 2)))):
            if a % i == 0 and b % i == 0:
                a /= i
                b /= i
        return (int(a),int(b))

    class App(Frame):
        def show(self,image=None,event=None):
            self.display.create_image((0,0),anchor=NW,image=image)

        def animate(self,event=None):
            self.show(image=self.gif['frames'][self.gif['loc']])
            self.gif['loc'] += 1
            if self.gif['loc'] == self.gif['len']:
                self.gif['loc'] = 0
            if self.cont:
                threading.Timer((self.gif['delay'] / 1000),self.animate).start()

        def kill(self,event=None):
```

```
            self.cont = False
            sleep(0.1)
            self.quit()

        def __init__(self,master):
            Frame.__init__(self,master)
            self.grid(row=0,sticky=N+E+S+W)
            self.rowconfigure(1,weight=2)
            self.rowconfigure(3,weight=1)
            self.columnconfigure(0,weight=1)
            self.title = Label(self,text='No title')
            self.title.grid(row=0,sticky=E+W)
            self.display = Canvas(self)
            self.display.grid(row=1,sticky=N+E+S+W)
            self.user = Label(self,text='Posted by No Username')
            self.user.grid(row=2,sticky=E+W)
            self.comment = Text(self,height=4,width=40,state=DISABLED)
            self.comment.grid(row=3,sticky=N+E+S+W)
            self.cont = True
            self.gif = anim_gif('test.gif')
            self.animate()

            root.protocol("WM_DELETE_WINDOW",self.kill)


    root = Tk()
    root.rowconfigure(0,weight=1)
    root.columnconfigure(0,weight=1)
    app = App(root)
    app.mainloop()
```

```
---------------------------------------------------------------------------

TclError                                  Traceback (most recent call last)

<ipython-input-149-ab0e46dfde6a> in <module>()
     79
     80
---> 81 root = Tk()
     82 root.rowconfigure(0,weight=1)
     83 root.columnconfigure(0,weight=1)


/opt/conda/lib/python3.5/tkinter/__init__.py in __init__(self, screenName, baseName, cl
   1875                 baseName = baseName + ext
   1876         interactive = 0
-> 1877         self.tk = _tkinter.create(screenName, baseName, className, interactive, wan
```

```
1878         if useTk:
1879             self._loadtk()
```

```
TclError: no display name and no $DISPLAY environment variable
```

In [ ]: