Temperature

In [1]:
```
import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import fisher_exact

data = pd.read_csv(filepath_or_buffer='../../../Archive/HTWTempRatios.csv')
data['HTW'] = (data['DodISm'] >= 0.25) | (data['DodI20m'] >= 0.25)
temp = [21.7, 16.6, 13.6, 25, 18.9, 14.7, 15.1, 13.9, 20, 19.4]
years = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]
runners = list(map (lambda x: len(data.loc[data['Year'] == x].index), years))
female = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['Gender'] == 'F')].index), years))
male = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['Gender'] == 'M')].index), years))
avg_time = list(map (lambda x: ((data.loc[(data['Year'] == x)]['Time']).mean(), years))
avg_time_f = list(map (lambda x: ((data.loc[(data['Year'] == x) & (data['Gender'] == 'F')]['Time']).mean(), years))
avg_time_m = list(map (lambda x: ((data.loc[(data['Year'] == x) & (data['Gender'] == 'M')]['Time']).mean(), years))
htw = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['HTW'] == True)].index), years))
htw_f = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['HTW'] == True) & (data['Gender'] == 'F')].index), years))
htw_m = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['HTW'] == True) & (data['Gender'] == 'M')].index), years))
neg_split = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['SplitRatio'] <= 1)].index), years))
neg_split_f = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['SplitRatio'] <= 1) & (data['Gender'] == 'F')].index), years))
neg_split_m = list(map (lambda x: len(data.loc[(data['Year'] == x) & (data['SplitRatio'] <= 1) & (data['Gender'] == 'M')].index), years))

d =  {'Year': years, 'Temp': temp, 'Runners': runners,'Female': female, 'Male': male,
      'Avg Time' : avg_time, 'Avg Time F' : avg_time_f, 'Avg Time M' : avg_time_m,
      'HTW': htw, 'HTW F': htw_f, 'HTW M' : htw_m, 'Neg Split' : neg_split,
      'Neg Split F' : neg_split_f, 'Neg Split M' : neg_split_m}
df = pd.DataFrame(data=d)
df['HTW%'] = df['HTW'] / df['Runners']
df['F HTW%'] = df['HTW F'] / df['Female']
df['M HTW%'] = df['HTW M'] / df['Male']
df['Neg Split%'] = df['Neg Split'] / df['Runners']
df['F Neg Split%'] = df['Neg Split F'] / df['Female']
df['M Neg Split%'] = df['Neg Split M'] / df['Male']
```

In [2]:
```
df
```

Out[2]:

	Year	Temp	Runners	Female	Male	Avg Time	Avg Time F	Avg Time M	HTW	HTW F	HTW M	Neg Split	Neg Split F	Neg Split M	HTW%	F HTW%	M HTW%	Neg Split%	F Neg Split%	M Neg Split%
0	2010	21.7	37982	10996	26986	7643.674056	8145.278192	7439.285148	5141	622	4519	2171	740	1431	0.135354	0.056566	0.167457	0.057159	0.067297	0.053027
1	2011	16.6	42838	13179	29659	7263.707736	7798.754989	7025.959068	2358	507	1851	5940	1597	4343	0.055045	0.038470	0.062409	0.138662	0.121178	0.146431
2	2012	13.6	44094	13750	30344	7208.821881	7749.826618	6963.672423	2768	607	2161	6297	1725	4572	0.062775	0.044145	0.071217	0.142809	0.125455	0.150672
3	2013	25.0	44919	14814	30105	7747.183441	8205.571756	7521.620761	6189	1123	5066	2997	1178	1819	0.137781	0.075807	0.168278	0.066720	0.079519	0.060422
4	2014	18.9	47187	16323	30864	7458.688389	7989.805060	7177.797466	5007	1179	3828	3165	1058	2107	0.106110	0.072229	0.124028	0.067074	0.064817	0.068267
5	2015	14.7	46207	16086	30121	7335.203281	7845.001865	7062.947379	3339	794	2545	5139	1592	3547	0.072262	0.049360	0.084493	0.111217	0.098968	0.117758
6	2016	15.1	44972	15662	29310	7343.250111	7876.157323	7058.487479	2522	573	1949	4442	1601	2841	0.056079	0.036585	0.066496	0.098773	0.102222	0.096929
7	2017	13.9	42252	14557	27695	7323.356078	7848.933915	7047.102726	2220	559	1661	5803	1834	3969	0.052542	0.038401	0.059975	0.137343	0.125987	0.143311
8	2018	20.0	39911	13775	26136	7519.654431	8079.968494	7224.340450	3614	814	2800	3133	1035	2098	0.090551	0.059093	0.107132	0.078500	0.075136	0.080272
9	2019	19.4	33134	11267	21867	7492.973200	8066.226502	7197.603695	3183	757	2426	2399	611	1788	0.096064	0.067187	0.110943	0.072403	0.054229	0.081767

Average times by temperature

In [3]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['Avg Time'])
print ("Linear regression all runners finish time")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("intercept: ", intercept)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
Linear regression all runners finish time
r-squared: 0.9111089869659573
p:  1.7715249036464247e-05
intercept:  6655.242823063195
slope:  43.510812593271396
std err:  4.805028447513403
```

In [4]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['Avg Time F'])
print ("Linear regression female finish time")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("intercept: ", intercept)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
Linear regression female finish time
r-squared: 0.8979538035167087
p:  3.0945034994583414e-05
intercept:  7248.004616470852
slope:  39.829393778516619
std err:  4.747711897897793
```

In [5]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['Avg Time M'])
print ("Linear regression male finish time")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("intercept: ", intercept)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
Linear regression male finish time
r-squared: 0.9098993795605811
p:  1.8709167068037667e-05
intercept:  6338.671104258585
slope:  46.57409475643498
std err:  5.181632825176719
```

HTW rates by temperature

In [6]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['HTW'])
print("Number of runners HTW by temperature")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
Number of runners HTW by temperature
r-squared: 0.7463542610048088
p:  0.0012689337299441673
slope:  312.5197011887763
std err:  64.41302235121836
```

In [7]:
```
#linear regression for men HTW by temperature.
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['HTW M'])
print("Male runners HTW by temperature")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
Male runners HTW by temperature
r-squared: 0.7585656976834076
p:  0.0010352165980106564
slope:  274.7679637766305
std err:  54.805480353646965
```

In [8]:
```
#linear regression for women HTW by temperature.
#It's not a good fit and women seem to not increase their risk of HTW as much as men.
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['HTW F'])
print("Female runners HTW by temperature")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
Female runners HTW by temperature
r-squared: 0.3666048334437487
p:  0.06359434553891391
slope:  37.751737412143335
std err:  17.544065061839188
```

In [9]:
```
#linear regression for % of men HTW by temperature.
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['M HTW%'])
print("% Male runners HTW by temperature")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
% Male runners HTW by temperature
r-squared: 0.8455525035440344
p:  0.0001662461930069671
slope:  0.010041290656111033
std err:  0.0015172774786787909
```

In [10]:
```
#linear regression for % of women HTW by temperature.
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['F HTW%'])
print("% Female runners HTW by temperature")
print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
% Female runners HTW by temperature
r-squared: 0.6633203737137115
p:  0.004118524114005285
slope:  0.0031697931925571831
std err:  0.0007984223525730542
```

Negative Split Rates by Temperature

For the negative splits, the relationship isn't really linear (see plots), which will be shown below, but rather there are two clusters, below and above 18 degrees, when more runners manage a negative split in the five cooler years.

In [11]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['Neg Split'])

print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
r-squared: 0.6841474625576467
p:  0.003153405979409332
slope:  -343.9905655168061
std err:  82.63591930164884
```

In [12]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['Neg Split M'])

print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
r-squared: 0.6863935996805802
p:  0.0030608845197342205
slope:  -257.5870283744486
std err:  61.55804823171513
```

In [13]:
```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Temp'],df['Neg Split F'])

print ("r-squared:", r_value**2)
print ("p: ", p_value)
print ("slope: ", slope)
print ("std err: ", std_err)
```

```
r-squared: 0.5695167082636373
p:  0.011643943985669154
slope:  -86.40353714236127
std err:  26.55897911788878
```

As the number of negative splits isn't linear, it appear to form two clusers, one for the cold years < 18 degrees, and one for the warm years > 18 degrees. Let's compute the differences between these conditions.

In [14]:
```
ct = pd.crosstab(data['temperature'] < 18, data['SplitRatio'] <= 1)
#temperature = False means the temperature is above 18 degrees C, temperatur = True that it is below
f_ct = pd.crosstab(data.loc[data['Gender'] == 'F']['temperature'] < 18, data['SplitRatio'] <= 1)
m_ct = pd.crosstab(data.loc[data['Gender'] == 'M']['temperature'] < 18, data['SplitRatio'] <= 1)
ct
```

Out[14]:

SplitRatio	False	True
temperature		
False	189268	13865
True	192742	27621

In [15]:
```
f_ct
```

Out[15]:

SplitRatio	False	True
temperature		
False	62553	4622
True	64885	8349

In [16]:
```
m_ct
```

Out[16]:

SplitRatio	False	True
temperature		
False	126715	9243
True	127857	19272

In [17]:
```
oddsr, p = fisher_exact(ct)
oddsrF, pF = fisher_exact(f_ct)
oddsrM, pM = fisher_exact(m_ct)
print("Neagtive Splits warm vs cold years: \n p: ", p, "OR: ", oddsr)
print("Neagtive Splits warm vs cold years (Female): \n p: ", pF, "OR: ", oddsrF)
print("Neagtive Splits warm vs cold years (Men): \n p: ", pM, "OR: ", oddsrM)
```

```
Neagtive Splits warm vs cold years:
 p:  0.0 OR:  1.9562319862232438
Neagtive Splits warm vs cold years (Female):
 p:  1.504591212971079e-190 OR:  1.7414393511243988
Neagtive Splits warm vs cold years (Men):
 p:  0.0 OR:  2.0664140774948905
```

In [ ]:
```

```