

20200326_FS_Beacon_over_time.

March 30, 2020

- 1 This script is designed to graph beacon times over multiple sessions over an animal, taking differences between visible and invisible beacon times.

1.0.1 imports

```
In [117]: import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
from matplotlib.patches import Circle
import matplotlib.tri as tri
import numpy as np
from scipy.spatial.transform import Rotation as R

root = 'C:/Users/Fabian/Desktop/Analysis/Multiple_trial_analysis/Data/Raw/'
figures = 'C:/Users/Fabian/Desktop/Analysis/Multiple_trial_analysis/Figures/'

#Data with beacon metadata

beacon_Day86_fs2 = pd.read_csv(root+'beacons 20200128-151826.txt',sep=" ", header=Nor
beacon_Day86_fs1 = pd.read_csv(root+'beacons 20200128-160013.txt',sep=" ", header=Nor

beacon_Day87_fs2 = pd.read_csv(root+'beacons 20200129-153534.txt',sep=" ", header=Nor
beacon_Day87_fs1 = pd.read_csv(root+'beacons 20200129-161806.txt',sep=" ", header=Nor

beacon_Day88_fs2 = pd.read_csv(root+'beacons 20200130-102126.txt',sep=" ", header=Nor
beacon_Day88_fs1 = pd.read_csv(root+'beacons 20200130-111741.txt',sep=" ", header=Nor

beacon_Day89_fs2 = pd.read_csv(root+'beacons 20200130-161126.txt',sep=" ", header=Nor
beacon_Day89_fs1 = pd.read_csv(root+'beacons 20200130-151829.txt',sep=" ", header=Nor

beacon_Day90_fs2 = pd.read_csv(root+'beacons 20200203-154441.txt',sep=" ", header=Nor
beacon_Day90_fs1 = pd.read_csv(root+'beacons 20200203-145842.txt',sep=" ", header=Nor

beacon_Day91_fs2 = pd.read_csv(root+'beacons 20200204-125552.txt',sep=" ", header=Nor
beacon_Day91_fs1 = pd.read_csv(root+'beacons 20200204-133905.txt',sep=" ", header=Nor
```

```

beacon_Day92_fs2 = pd.read_csv(root+'beacons 20200205-143220.txt',sep=" ", header=None)
beacon_Day92_fs1 = pd.read_csv(root+'beacons 20200205-151052.txt',sep=" ", header=None)

beacon_Day93_fs2 = pd.read_csv(root+'beacons 20200206-133529.txt',sep=" ", header=None)
beacon_Day93_fs1 = pd.read_csv(root+'beacons 20200206-125706.txt',sep=" ", header=None)


Day46_fs1 = pd.read_csv(root+'position 20190923-174441.txt',sep=" ", header=None)
Day46_fs2 = pd.read_csv(root+'position 20190923-171112.txt',sep=" ", header=None)
Day47_fs1 = pd.read_csv(root+'position 20191001-112411.txt',sep=" ", header=None)
Day47_fs2 = pd.read_csv(root+'position 20191001-115127.txt',sep=" ", header=None)
Day48_fs1 = pd.read_csv(root+'position 20191002-115000.txt',sep=" ", header=None)
Day48_fs2 = pd.read_csv(root+'position 20191002-111038.txt',sep=" ", header=None)
Day51_fs1 = pd.read_csv(root+'position 20191106-170809.txt',sep=" ", header=None)
Day52_fs2 = pd.read_csv(root+'position 20191107-174215.txt',sep=" ", header=None)
Day52_fs1 = pd.read_csv(root+'position 20191107-183857.txt',sep=" ", header=None)
Day53_fs2 = pd.read_csv(root+'position 20191108-142321.txt',sep=" ", header=None)
Day53_fs1 = pd.read_csv(root+'position 20191108-145125.txt',sep=" ", header=None)
Day66_fs1 = pd.read_csv(root+'position 20191118-161325.txt',sep=" ", header=None)
Day66_fs2 = pd.read_csv(root+'position 20191118-171209.txt',sep=" ", header=None)
Day72_fs1 = pd.read_csv(root+'position 20191127-122008.txt',sep=" ", header=None)
Day72_fs2 = pd.read_csv(root+'position 20191127-132223.txt',sep=" ", header=None)


Day79_fs2 = pd.read_csv(root+'position 20200121-154004.txt',sep=" ", header=None)
Day79_fs1 = pd.read_csv(root+'position 20200121-161359.txt',sep=" ", header=None)


Day80_fs2 = pd.read_csv(root+'position 20200122-141738.txt',sep=" ", header=None)
Day80_fs1 = pd.read_csv(root+'position 20200122-133022.txt',sep=" ", header=None)


Day81_fs2 = pd.read_csv(root+'position 20200123-141930.txt',sep=" ", header=None)
Day81_fs1 = pd.read_csv(root+'position 20200123-150059.txt',sep=" ", header=None)


Day82_fs2 = pd.read_csv(root+'position 20200124-151642.txt',sep=" ", header=None)
Day82_fs1 = pd.read_csv(root+'position 20200124-160826.txt',sep=" ", header=None)


Day83_fs2 = pd.read_csv(root+'position 20200126-183810.txt',sep=" ", header=None)
Day83_fs1 = pd.read_csv(root+'position 20200126-180200.txt',sep=" ", header=None)


Day84_fs2 = pd.read_csv(root+'position 20200127-205615.txt',sep=" ", header=None)
Day84_fs1 = pd.read_csv(root+'position 20200127-155645.txt',sep=" ", header=None)


Day85_fs2 = pd.read_csv(root+'position 20200128-112255.txt',sep=" ", header=None)
Day85_fs1 = pd.read_csv(root+'position 20200128-104637.txt',sep=" ", header=None)


Day86_fs2 = pd.read_csv(root+'position 20200128-160013.txt',sep=" ", header=None)
Day86_fs1 = pd.read_csv(root+'position 20200128-151826.txt',sep=" ", header=None)

```

```

Day87_fs2 = pd.read_csv(root+'position 20200129-153534.txt',sep=" ", header=None)
Day87_fs1 = pd.read_csv(root+'position 20200129-161806.txt',sep=" ", header=None)

Day88_fs2 = pd.read_csv(root+'position 20200130-102126.txt',sep=" ", header=None)
Day88_fs1 = pd.read_csv(root+'position 20200130-111741.txt',sep=" ", header=None)

Day89_fs2 = pd.read_csv(root+'position 20200130-161126.txt',sep=" ", header=None)
Day89_fs1 = pd.read_csv(root+'position 20200130-151829.txt',sep=" ", header=None)

Day90_fs2 = pd.read_csv(root+'position 20200203-154441.txt',sep=" ", header=None)
Day90_fs1 = pd.read_csv(root+'position 20200203-145842.txt',sep=" ", header=None)

Day91_fs2 = pd.read_csv(root+'position 20200204-125552.txt',sep=" ", header=None)
Day91_fs1 = pd.read_csv(root+'position 20200204-133905.txt',sep=" ", header=None)

Day92_fs2 = pd.read_csv(root+'position 20200205-143220.txt',sep=" ", header=None)
Day92_fs1 = pd.read_csv(root+'position 20200205-151052.txt',sep=" ", header=None)

Day93_fs2 = pd.read_csv(root+'position 20200206-133529.txt',sep=" ", header=None)
Day93_fs1 = pd.read_csv(root+'position 20200206-125706.txt',sep=" ", header=None)

```

In [130]: *#first time is in between visible and invisible, not from start.*

```

beacon_time= beacon_Day86_fs1[0]
first=beacon_time[1]-beacon_time[0]
diff=np.diff(beacon_time)
print(first)
print(diff)
diff[1::2]

```

43.670000076293945

```

[ 43.67000008  29.42000008  26.15999985  50.80999994  45.74000001
  24.73000002 104.09000015  14.42999983   5.49000001   3.60000014
  39.24000001   5.12999988  93.88000011   5.32999992  49.25999999
  21.70000005  53.1500001  15.65999985   6.95000005  22.17000008
 414.38999987  31.23000002   9.17000008  66.17000008  11.5999999
  28.57999992 264.29999995   6.45000005  26.03999996  20.25
 15.5400002   90.12999988]

```

```

Out[130]: array([29.42000008, 50.80999994, 24.73000002, 14.42999983,  3.60000014,
                  5.12999988,  5.32999992, 21.70000005, 15.65999985, 22.17000008,
                  31.23000002, 66.17000008, 28.57999992,  6.45000005, 20.25      ,
                  90.12999988])

```

1.1 Scatter plot per session

```

In [143]: fig, ax = plt.subplots(dpi= 500)
          ax.scatter(list(np.arange(1,len(list(diff[1::2]))+1,1)),diff[1::2], marker='v',c='c' )

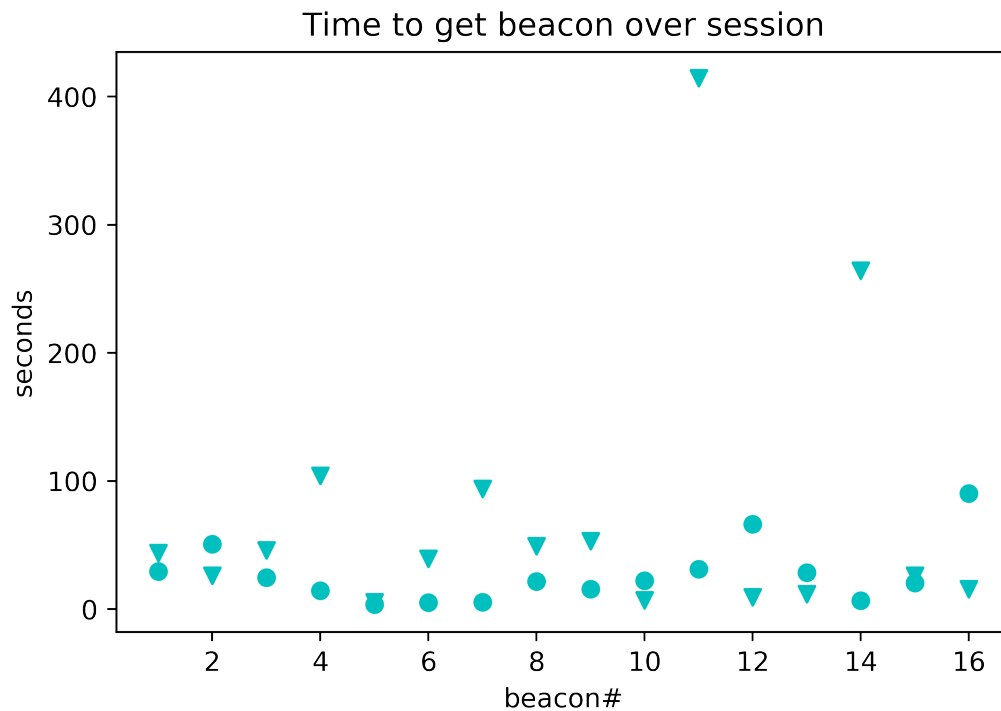
```

```

ax.scatter(list(np.arange(1,len(list(diff[1::2]))+1,1)),diff[1::2], marker='o',c='c')
ax.set_ylabel('seconds')
ax.set_xlabel('beacon#')
ax.set_title('Time to get beacon over session')
ax.imshow()

```

Out[143]: Text(0.5, 1.0, 'Time to get beacon over session')



1.2 Now Graph beacons visible vs. invisible over particular session.

```

In [119]: def beacons_over_session (beacons_vis,beacons_invis,list_of_number_of_beacons,session)
           """this function takes lists of days for each animal and plots a distance covered"""

           x = np.arange(len(list_of_number_of_beacons)) # the label locations
           width = 0.35 # the width of the bars

           fig, ax = plt.subplots(dpi= 500)
           FS1 = ax.bar(x - width/2, beacons_vis, width, label='visible '+ str(int(np.sum(beacons_vis))))
           FS2 = ax.bar(x + width/2, beacons_invis, width, fill=False, label='invisible '+ str(int(np.sum(beacons_invis))))

           ax.set_ylabel('seconds')
           ax.set_xlabel('beacon#')

```

```

ax.set_title('Time to get beacon over session')
ax.set_xticks(x)
ax.set_xticklabels(list_of_number_of_beacons)
ax.legend(loc='upper left',prop={'size': 6})

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(int(height)),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 0), # 3 points vertical offset - set to 0
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(FS1)
autolabel(FS2)

fig.tight_layout()
plt.savefig(figures+'beacons_over_session_'+ session +'.png', dpi = 1000)
plt.show()

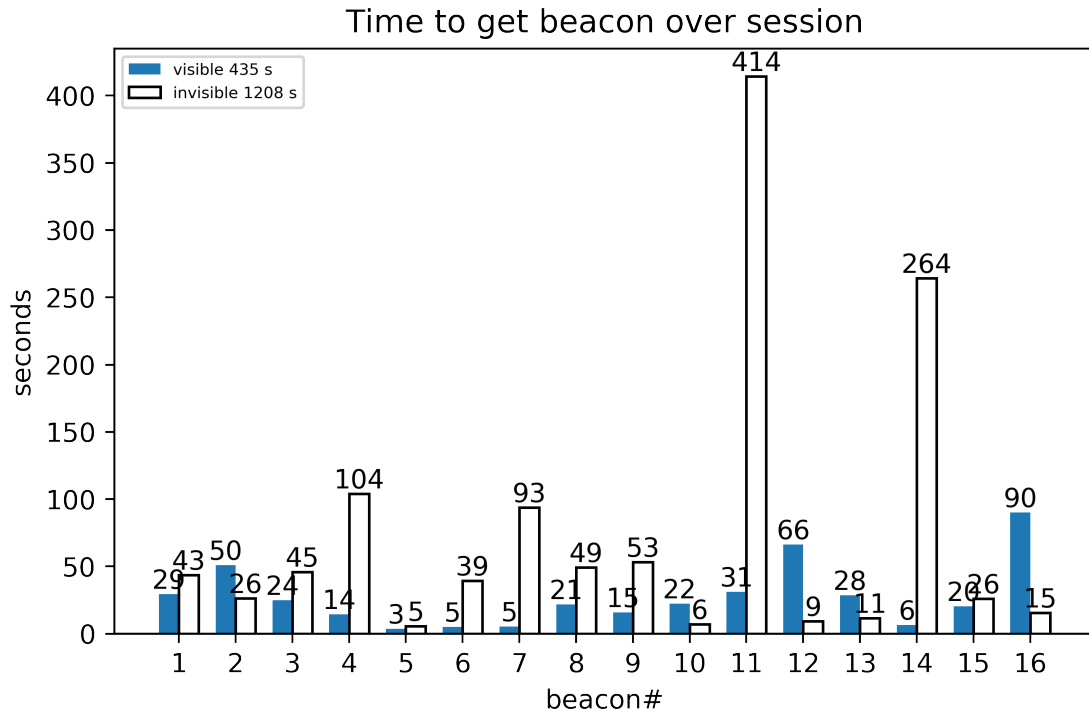
```

1.3 Graph Day 86

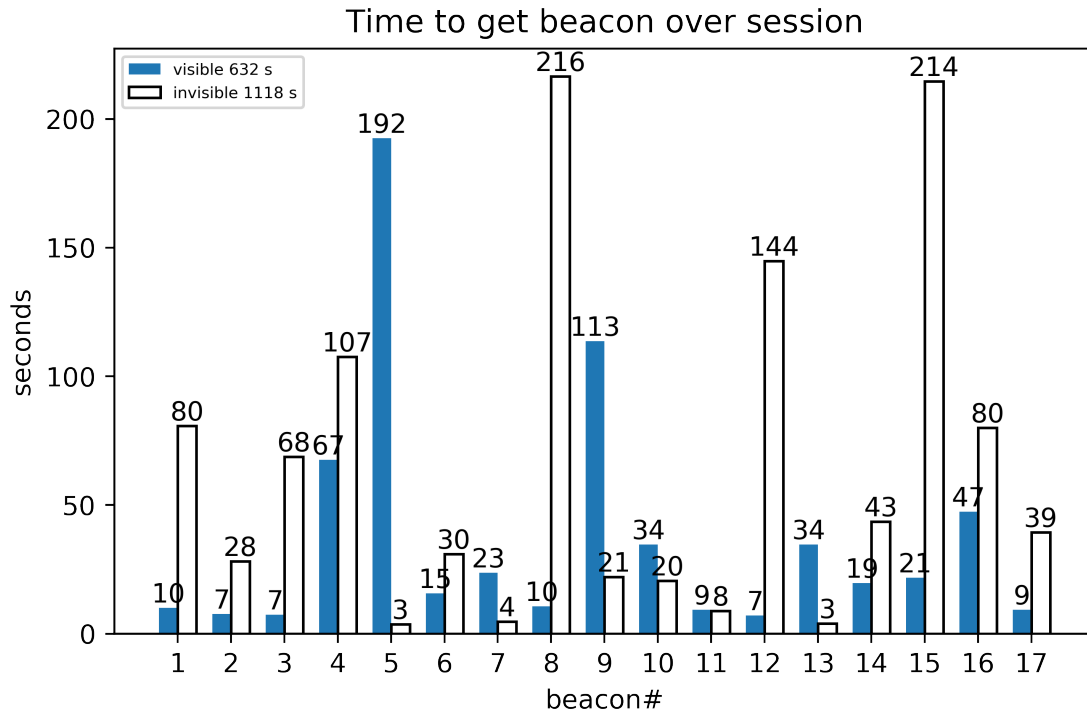
```

In [120]: diff=np.diff(beacon_Day86_fs1[0])
          session = 'FS1_86'
          Beacons_vis = list(diff[1::2])
          Beacons_invis = list(diff[::2])
          Beacon_number =list(np.arange(1,len(list(diff[::2]))+1,1))
          beacons_over_session(Beacons_vis, Beacons_invis, Beacon_number,session)

```



```
In [127]: diff=np.diff(beacon_Day86_fs2[0])
          session = 'FS2_86'
          Beacons_vis = list(diff[1::2])
          Beacons_invis = list(diff[:,2])
          Beacon_number =list(np.arange(1,len(list(diff[:,2]))+1,1))
          beacons_over_session(Beacons_vis, Beacons_invis, Beacon_number,session)
```



1.4 PS: Can have mishape problem if beaocn list is odd! - might need to change Beacon_number = list(np.arange(1,len(list(diff[:,2]))+1,1)) so it fits the shape

2 Now graphing overall beacons over many sessions

2.1 first: make a summary of all of them and how to add together

```
In [122]: visible=[]
invisible = []
beacons = [beacon_Day86_fs1,beacon_Day87_fs1,beacon_Day88_fs1,beacon_Day89_fs1,beacon_Day90_fs1,beacon_Day91_fs1,beacon_Day92_fs1,beacon_Day93_fs1,beacon_Day94_fs1,beacon_Day95_fs1,beacon_Day96_fs1,beacon_Day97_fs1,beacon_Day98_fs1,beacon_Day99_fs1,beacon_Day100_fs1]

for beacon in beacons:
    diff=np.diff(beacon[0])
    visible.append(sum(diff[1::2]))
    invisible.append (sum(diff[0::2]))
print (sum(visible))
print (sum(invisible))
```

```
1764.4599997997284
12228.280000209808
```

2.2 Now graphing function

```
In [123]: beacons = [beacon_Day86_fs1, beacon_Day87_fs1, beacon_Day88_fs1, beacon_Day89_fs1, beacon_Day90_fs1]
Day_number_list = ('86', '87', '88', '89', '90')

def Total_beacons_over_sessions (list_of_beacon_days, list_of_number_of_days, animal_ID):
    """this function takes lists of days for each animal and plots a distance covered"""

    visible = []
    invisible = []

    for beacon in list_of_beacon_days:
        diff = np.diff(beacon[0])
        visible.append(sum(diff[1::2]))
        invisible.append(sum(diff[0::2]))

    x = np.arange(len(list_of_number_of_days)) # the label locations
    width = 0.35 # the width of the bars

    fig, ax = plt.subplots(dpi= 500)
    FS1 = ax.bar(x - width/2, visible, width, label='visible ' + str(int(np.sum(visible))))
    FS2 = ax.bar(x + width/2, invisible, width, fill=False, label='invisible ' + str(int(np.sum(invisible))))

    ax.set_ylabel('seconds looking for beacon')
    ax.set_xlabel('session#')
    ax.set_title('Time to get beacon over session ' + animal_ID)
    ax.set_xticks(x)
    ax.set_xticklabels(list_of_number_of_days)
    ax.legend(loc='upper left', prop={'size': 5})

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}' .format(int(height)),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 0), # 3 points vertical offset - set to 0
                    textcoords="offset points",
                    ha='center', va='bottom')

    autolabel(FS1)
    autolabel(FS2)
```



```

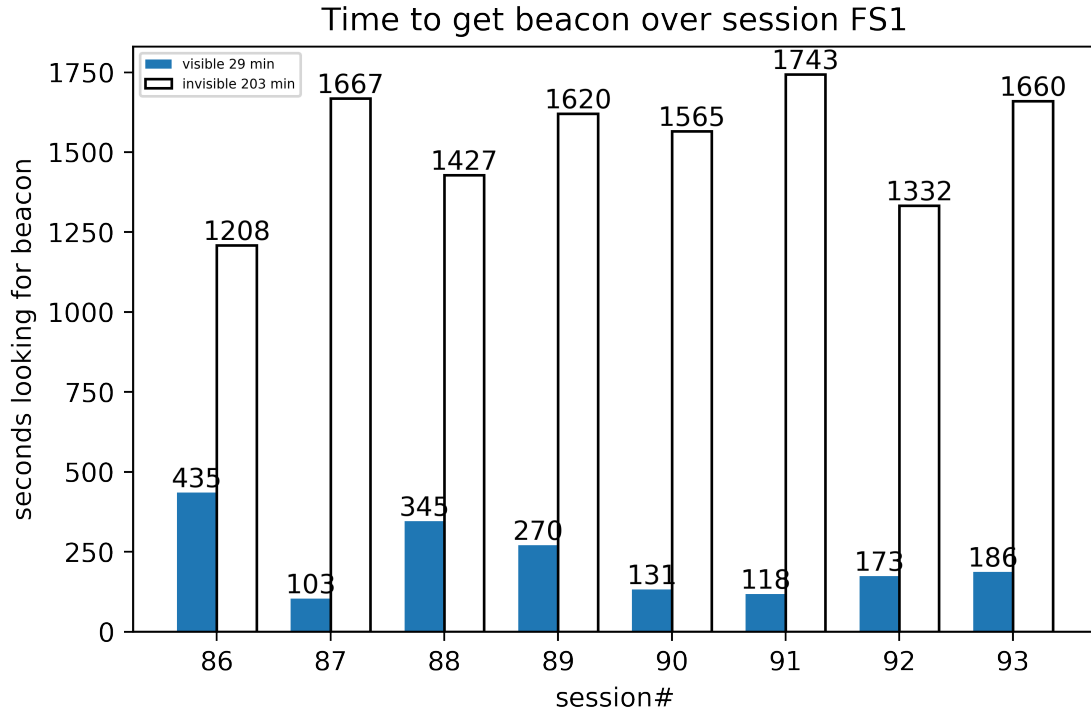
fig.tight_layout()
plt.savefig(figures+'Total_beacons_over_session_'+ animal_ID + '_' +Day_number_list)
plt.show()

```

```

In [124]: beacons = [beacon_Day86_fs1,beacon_Day87_fs1,beacon_Day88_fs1,beacon_Day89_fs1,beacon_Day90_fs1,beacon_Day91_fs1,beacon_Day92_fs1,beacon_Day93_fs1]
Day_number_list = ('86','87','88','89','90','91','92','93')
Animal = 'FS1'
Total_beacons_over_sessions(beacons,Day_number_list,Animal)

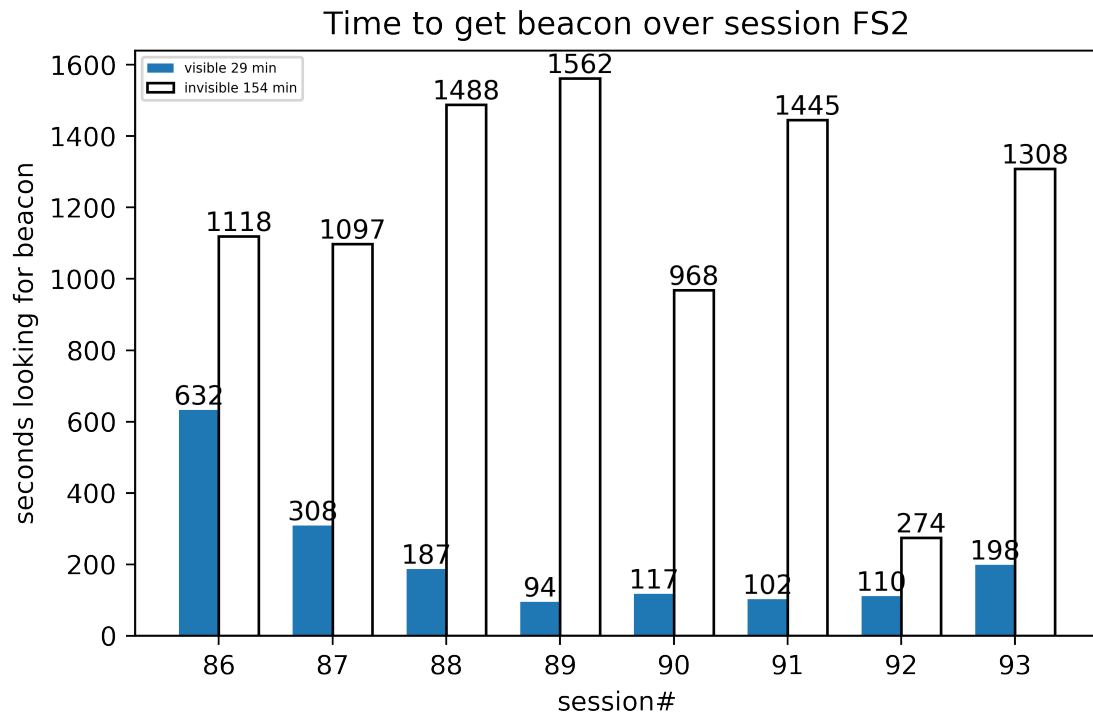
```



```

In [125]: beacons = [beacon_Day86_fs2,beacon_Day87_fs2,beacon_Day88_fs2,beacon_Day89_fs2,beacon_Day90_fs2,beacon_Day91_fs2,beacon_Day92_fs2,beacon_Day93_fs2]
Day_number_list = ('86','87','88','89','90','91','92','93')
Animal = 'FS2'
Total_beacons_over_sessions(beacons,Day_number_list,Animal)

```



In []: