analysis

December 8, 2019

rename paths from unix timestamp to iso 8601

```
In [ ]: """
        import os
        from pathlib import Path
        basedir = Path("data")
        for fn in basedir.glob('*.zip'):

            file_split = fn.name.split("_")
            if len(file_split) != 2:
                continue
            try:
                file_split[0] = datetime.utcfromtimestamp(float(file_split[0])).isoformat()
            except Exception as e:
                continue
            new_file = basedir / "_".join(file_split)
            fn.rename(new_file)
        """
```

```
In [ ]: """
        import os
        from pathlib import Path
        basedir = Path("data")
        for fn in os.listdir(basedir):
            fn = basedir / fn
            if not fn.is_dir():
                continue

            file_split = fn.name.split("_")
            if len(file_split) != 2:
                continue
            try:
                file_split[0] = datetime.utcfromtimestamp(float(file_split[0])).isoformat()
            except Exception as e:
                continue
            new_file = basedir / "_".join(file_split)
            fn.rename(new_file)
        """
```

1 Task 3

1.1 Imports

```
In [14]: import pandas as pd
         import matplotlib.pyplot as plt
         import numpy as np
         import seaborn as sbn#
         from sklearn.linear_model import LinearRegression
         import zipfile
         from datetime import datetime
         plt.rcParams['figure.dpi'] = 300
```

1.2 Reading Data

```
In [15]: def _read_zipped_csv(archive, path):
             with archive.open(path,"r") as f:
                 df = pd.read_csv(f,header=None)
                 timestamp = datetime.utcfromtimestamp(float(df.iloc[0,0]))
                 sample_rate = float(df.iloc[1,0])
                 df = df.iloc[2:]
                 if len(df.columns) == 1:
                     df = np.array(df[0])
             return {"timestamp":timestamp
                     ,"sample_rate":sample_rate
                     ,"data": df}

         def read_e4_data(path):
             sensors = ["TEMP", "EDA", "BVP", "ACC", "HR"]#"IBI",
             annotations = ["tags"]
             data = {}
             with zipfile.ZipFile(path, 'r') as archive:
                 for s in sensors:
                     file_name = f"{s}.csv"
                     #print(file_name)
                     data[s] = _read_zipped_csv(archive, file_name)
             return data
```

```
In [16]: contr_1 = read_e4_data("data/2019-10-21T14:21:47_A0208E.zip")
         contr_2 = read_e4_data("data/2019-10-21T14:21:48_A023B2.zip")

         exp_1   = read_e4_data("data/2019-10-21T14:32:26_A0208E.zip")
         exp_2   = read_e4_data("data/2019-10-21T14:32:27_A023B2.zip")
```

1.3 Bonus Hypothesis #1 (ANOVA)

Hypothesis: **Hypothesis: Participants in the experimental condition collaborate more, using more body language and moving their hands more, than participants in the control group.**

In [17]:
```python
ACC_SAMPLE_RATE = int(exp_1["ACC"]["sample_rate"])
def acc_abs(data):
    tmp = np.linalg.norm(data["ACC"]["data"],axis = 1)
    #tmp -= np.mean(tmp)
    return tmp

#only look at last 180ms
def region_of_interest(data):
    return list(data[-200*ACC_SAMPLE_RATE:-20*ACC_SAMPLE_RATE])

e1_acc = acc_abs(exp_1)
e2_acc = acc_abs(exp_2)

c1_acc = acc_abs(contr_1)
c2_acc = acc_abs(contr_2)

from statsmodels.formula.api import ols
groups = np.array(list([0]*ACC_SAMPLE_RATE*180*2) + list([1]*ACC_SAMPLE_RATE*180*2))

data_exp = region_of_interest(e1_acc)+ region_of_interest(e2_acc)
data_controll = region_of_interest(c1_acc)+region_of_interest(c2_acc)

all_data= data_exp + data_controll
all_data = np.array(all_data)
print(np.mean(data_exp))
print(np.mean(data_controll))

df = pd.DataFrame(np.array([all_data,groups]).T,columns=["data","group"])
results = ols('data ~ C(group)', data=df).fit()
results.summary()
```

```
64.34814229087641
63.31739830998151
```

Out[17]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                   data   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     258.6
Date:                Sun, 08 Dec 2019   Prob (F-statistic):           7.10e-58
Time:                        23:40:11   Log-Likelihood:                -69140.
No. Observations:               23040   AIC:                         1.383e+05
Df Residuals:                   23038   BIC:                         1.383e+05
Df Model:                           1
Covariance Type:            nonrobust
```

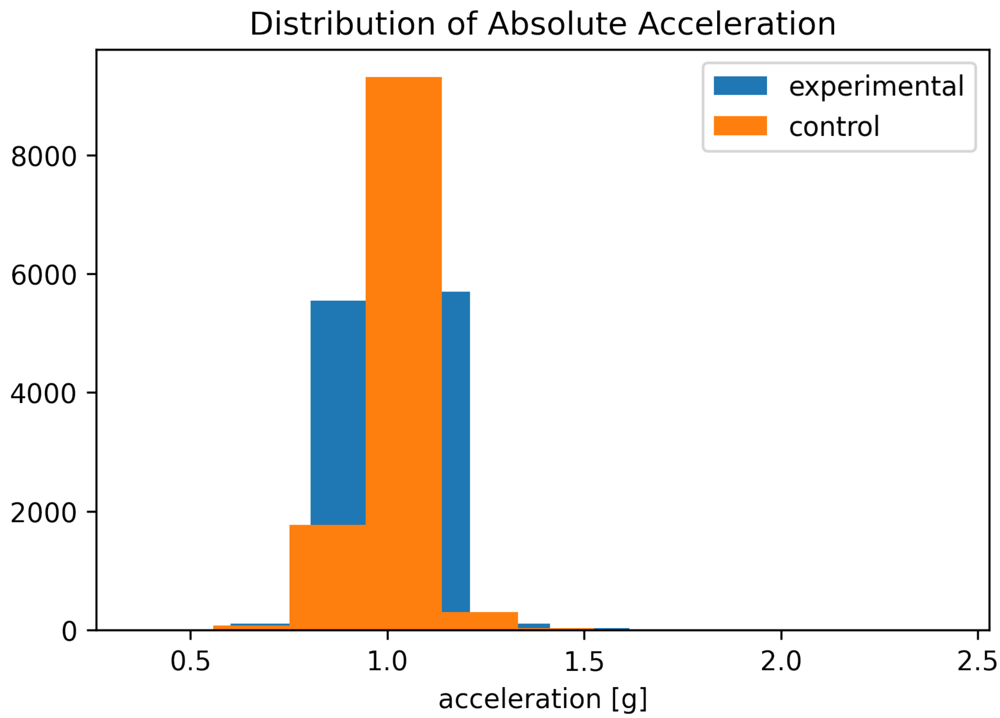
```
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept         64.3481      0.045   1419.804      0.000      64.259      64.437
C(group)[T.1.0]   -1.0307      0.064    -16.082      0.000      -1.156      -0.905
==============================================================================
Omnibus:                    14650.146   Durbin-Watson:                   1.157
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1226252.002
Skew:                           2.252   Prob(JB):                         0.00
Kurtosis:                      38.455   Cond. No.                         2.62
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
"""
```

Result: The anova shows an overall significant difference ($p = 7.10 * 10^{-58}$). Since we only had 2 groups (experiment vs. controll) the anova shows a significant difference between the mean absolute acceleration between these two groups. This might indicate that participants in the experimental group were more creative and used more body language or it might be just by chance, since the sample size is so small.

In [18]:
```python
ACC_UNIT = 1/64
plt.hist(np.array(data_exp)*ACC_UNIT)
plt.hist(np.array(data_controll)*ACC_UNIT)
plt.title("Distribution of Absolute Acceleration")
plt.legend(["experimental", "control"])
plt.xlabel("acceleration [g]")
```

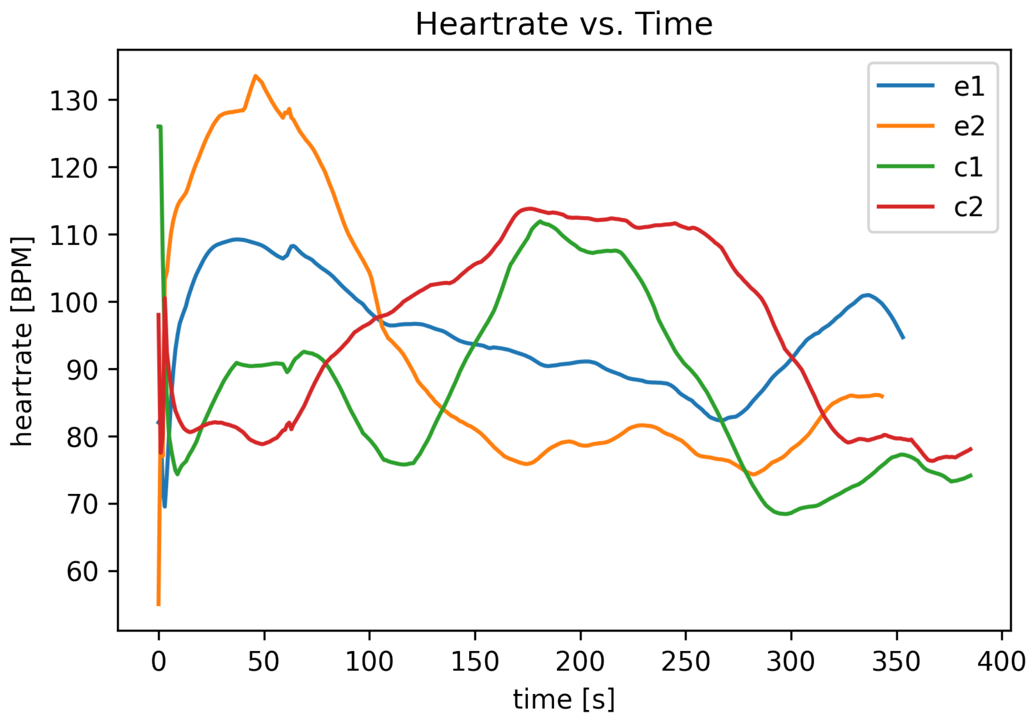
Out[18]:
```
Text(0.5, 0, 'acceleration [g]')
```



1.4 Bonus Hypothesis 2 (LinReg)

Hypothesis: Participants in the experimental group experience less stress and therefore have a lower heartrate than participants in the controll group.

```
In [19]: plt.plot(exp_1["HR"]["data"])
         plt.plot(exp_2["HR"]["data"])
         plt.plot(contr_1["HR"]["data"])
         plt.plot(contr_2["HR"]["data"])
         plt.legend(["e1","e2","c1","c2"])
         plt.xlabel("time [s]")
         plt.ylabel("heartrate [BPM]")
         plt.title("Heartrate vs. Time")
         plt.show()
```

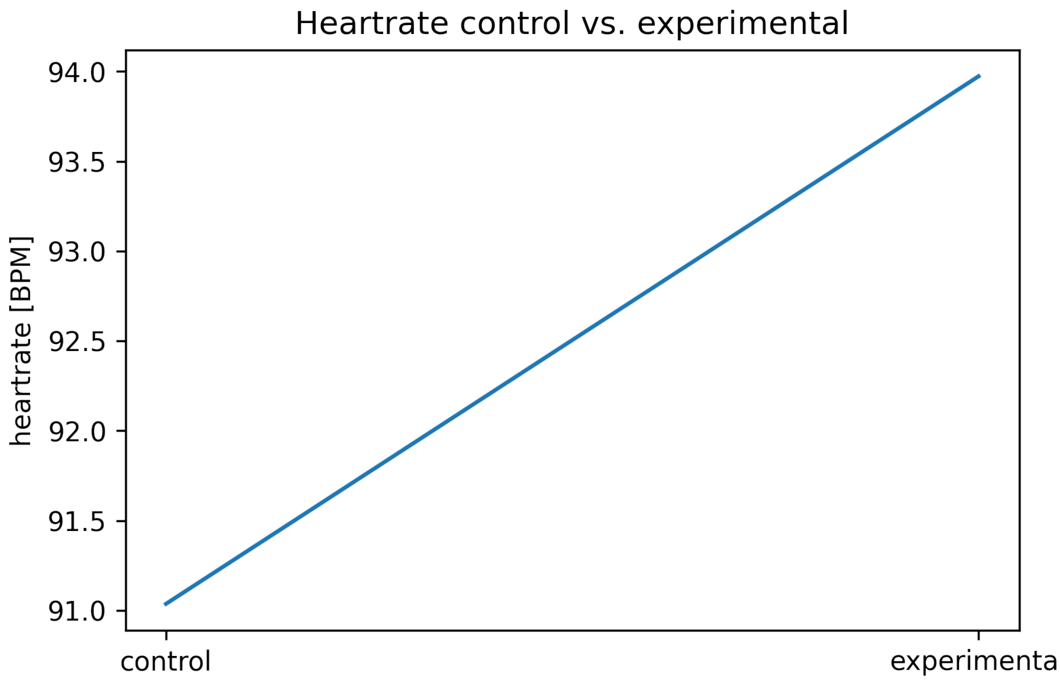


In [20]:
```
hr_exp = list(exp_1["HR"]["data"][10:]) + list(exp_2["HR"]["data"][10:])
hr_contr = list(contr_1["HR"]["data"][10:]) + list(contr_2["HR"]["data"][10:])

y = [1] * len(hr_exp) + [0]*len(hr_contr)

regressor = LinearRegression()
regressor.fit(np.array([y]).T,hr_exp+hr_contr)
pred = regressor.predict([[0],[1]])
plt.plot(pred)
plt.xticks([0,1],["control","experimental"])
plt.ylabel("heartrate [BPM]")
plt.title("Heartrate control vs. experimental")
```

Out[20]:
```
Text(0.5, 1.0, 'Heartrate control vs. experimental')
```



Result: The average heartrate in the control group was 91 and 94 in the experimental condition group. Contrary to our expectations, participants in the experimental group had a higher mean heartrate. This is probably just by chance, since we only had 2 participants in each group and did an between subjects study. Additionally, no baseline heartrate was measured, so it is not clear whether the higher heart rate was due to the activity, or whether the participants simply had higher resting heart rates. The heart rate of healthy humans at rest can range from around 50 to 100 bpm. The largest problem we faced when analysing the data, was not being able to sync the data with the videos, thus not knowing when the actual experiment started and ended. We also did not record baseline measurements before or after the experiment as would be needed for EDA analysis.

```
In [ ]: #plt.plot(e1_acc[-180*ACC_SAMPLE_RATE:])
        """
        plt.hist(region_of_interest(e2_acc))
        plt.hist(region_of_interest(e1_acc))

        plt.hist(region_of_interest(c1_acc))
        plt.hist(region_of_interest(c2_acc))
        #plt.hist(c1_acc[-180*ACC_SAMPLE_RATE:])
        #plt.hist(c2_acc[-180*ACC_SAMPLE_RATE:])
        """

In [ ]:

In [ ]:
```