Preliminary Analyses of Actigraph Data for BIS and KPS students

```
md"""
## Preliminary Analyses of Actigraph Data for BIS and KPS students

"""
```

```
Enter cell code...
```

```
using CSV , DataFrames , XLSX , Gadfly , Statistics , GLM
```

```
bisdf = XLSXFile("/tmp/jl_WbIDGVA5oB") containing 1 Worksheet
                    sheetname size          range
        -------------------------------------------------
                         Sheet1 1379x98       A1:CT1379
```

```
# Read the excel file

bisdf =
XLSX.readxlsx(download("https://github.com/arinbasu/activation_study/raw/master/BIS_
collated_v1.xlsx"))
```

	W_Filename	W_Epoch	W_Date	W_Hour	W_Day of Week	V
1	"932001W.agd"	60	"31/03/2023"	"09:00"	"Friday"	5
2	"932001W.agd"	60	"31/03/2023"	"10:00"	"Friday"	5
3	"932001W.agd"	60	"31/03/2023"	"11:00"	"Friday"	5
4	"932001W.agd"	60	"31/03/2023"	"12:00"	"Friday"	5
5	"932001W.agd"	60	"31/03/2023"	"13:00"	"Friday"	5
6	"932001W.agd"	60	"31/03/2023"	"14:00"	"Friday"	5
7	"932001W.agd"	60	"31/03/2023"	"15:00"	"Friday"	5
8	"932001W.agd"	60	"31/03/2023"	"16:00"	"Friday"	5
9	"932001W.agd"	60	"31/03/2023"	"17:00"	"Friday"	5
10	"932001W.agd"	60	"31/03/2023"	"18:00"	"Friday"	5
more						
1378	"932020W (2022-11-24)60sec.agd"	60	"30/11/2022"	"21:00"	"Wednesday"	3

```
# Read the excel file for further processing
bis =
DataFrame(XLSX.readtable(download("https://github.com/arinbasu/activation_study/raw/
master/BIS_collated_v1.xlsx"), "Sheet1"))
```

```
# Print out the names of the variables contained in the data frame

foreach(x -> println(x), names(bis))
```

```
W_Axis 2 Average Counts
W_Axis 3 Average Counts
W_Axis 1 Max Counts
W_Axis 2 Max Counts
W_Axis 3 Max Counts
W_Axis 1 CPM
W_Axis 2 CPM
W_Axis 3 CPM
W_Vector Magnitude Counts
W_Vector Magnitude Average Counts
W_Vector Magnitude Max Counts
W_Vector Magnitude CPM
W_Steps Counts
W_Steps Average Counts
W_Steps Max Counts
W_Steps Per Minute
W_Lux Average Counts
W_Lux Max Counts
W_Number of Epochs
W_Time
W_Calendar Days
H_Filename
H_Epoch
H_Date
H_Hour
H_Day of Week
H_Day of Week Num
H_Number of Sedentary Bouts occurring on this hour
H_Number of Sedentary Bouts Starting on the hour
H_Number of Sedentary Bouts Ending on the hour
H_Total time of Sedentary Bouts Occuring on this hour
H_Number of Sedentary Breaks Occurring on the hour
H_Number of Sedentary Breaks Starting on the hour
H_Number of Sedentary Breaks Ending on the hour
H_Total time of Sedentary Breaks Occuring on this hour
H_Sedentary
```

Selection deleted

bis1 =

	W_Filename	W_Hour	W_Sedentary	W_Light	W_Moderate	W
1	"932001W.agd"	"09:00"	1	53	6	0
2	"932001W.agd"	"10:00"	1	20	13	2(
3	"932001W.agd"	"11:00"	0	35	15	1(
4	"932001W.agd"	"12:00"	0	25	18	1;
5	"932001W.agd"	"13:00"	0	11	14	3!
6	"932001W.agd"	"14:00"	0	18	26	1(
7	"932001W.agd"	"15:00"	3	20	19	1{
8	"932001W.agd"	"16:00"	2	41	13	4
9	"932001W.agd"	"17:00"	9	36	14	1
10	"932001W.agd"	"18:00"	2	31	19	8
	more					
1378	"932020W (2022-11-24)60sec.agd"	"21:00"	46	13	1	0

```
# select relevant variables from bis and store in data frame bis1

bis1 = select(bis, ["W_Filename", "W_Hour", "W_Sedentary",
"W_Light", "W_Moderate", "W_Vigorous", "W_Steps Counts", "H_Filename",
"H_Hour", "H_Sedentary", "H_Light", "H_Moderate", "H_Vigorous",
"H_Steps Counts"])
```

14

```
ncol(bis1) # number of variables
```

1378

```
nrow(bis1) # number of cases or children
```

```
foreach(x -> println(x), names(bis1)) # LIst of variables
```

```
W_Filename
W_Hour
W_Sedentary
W_Light
W_Moderate
W_Vigorous
W_Steps Counts
H_Filename
H_Hour
H_Sedentary
H_Light
H_Moderate
H_Vigorous
H_Steps Counts
```

[59, 59, 60, 60, 60, 60, 57, 58, 51, 58, 57, 53, 25, 8, 0, 0, 0, 0, 0, 0, more ,58, 59,

```
# For BIS, calculate total activity level for Wrist EXCLUDING sedenatary levels

bis1.W_total = bis1.W_Light + bis1.W_Moderate + bis1.W_Vigorous
```

[6, 39, 25, 35, 49, 42, 37, 17, 15, 27, 2, 3, 2, 4, 0, 0, 0, 0, 0, 0, more ,36, 50, 39,

```
# Calculate MVPA for BIS students for Wrist

bis1.W_mvpa = bis1.W_Moderate + bis1.W_Vigorous
```

```
Enter cell code...
```

[10, 18, 23, 34, 50, 39, 46, 38, 30, 35, 26, 14, 9, 7, 0, 0, 0, 0, 0, 0, more ,47, 57, 5

```
# Calculate total activity levels, i.e., light+moderate+vigorous for Hip

bis1.H_total = bis1.H_Light + bis1.H_Moderate + bis1.H_Vigorous
```

```
Enter cell code...
```

[0, 0, 0, 1, 5, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, more ,3, 13, 2, 15, 0, 0, 0,

```
# Calculate and store a variable for Hip MVPA
bis1.H_mvpa = bis1.H_Moderate + bis1.H_Vigorous
```

The next set of analyses will involve total activity registered in the wrist or hip based devices for BIS and also for KPS as separate analyses

```
md"""

The next set of analyses will involve total activity registered in the wrist or hip based devices for BIS and also for KPS as separate analyses

"""
```

bis1_filtered =

	W_Filename	W_Hour	W_Sedentary	W_Light	W_Moderate	W_
1	"932001W.agd"	"09:00"	1	53	6	0
2	"932001W.agd"	"10:00"	1	20	13	26
3	"932001W.agd"	"11:00"	0	35	15	10
4	"932001W.agd"	"12:00"	0	25	18	17
5	"932001W.agd"	"13:00"	0	11	14	35
6	"932001W.agd"	"14:00"	0	18	26	16
7	"932001W.agd"	"15:00"	3	20	19	18
8	"932001W.agd"	"16:00"	2	41	13	4
9	"932001W.agd"	"17:00"	9	36	14	1
10	"932001W.agd"	"18:00"	2	31	19	8
more						
843	"932020W (2022-11-24)60sec.agd"	"21:00"	46	13	1	0

```
# Filter BIS data based on total activities at Wrist where total activities > 0

bis1_filtered = filter(:W_total => (x -> x > 0 ) , bis1)
```

843

```
nrow(bis1_filtered) # total number of hours registered = 843
```

bis1_filtered1 =

	W_Filename	W_Hour	W_Sedentary	W_Light	W_Moderate	W_
1	"932001W.agd"	"09:00"	1	53	6	0
2	"932001W.agd"	"10:00"	1	20	13	26
3	"932001W.agd"	"11:00"	0	35	15	10
4	"932001W.agd"	"12:00"	0	25	18	17
5	"932001W.agd"	"13:00"	0	11	14	35
6	"932001W.agd"	"14:00"	0	18	26	16
7	"932001W.agd"	"15:00"	3	20	19	18
8	"932001W.agd"	"16:00"	2	41	13	4
9	"932001W.agd"	"17:00"	9	36	14	1
10	"932001W.agd"	"18:00"	2	31	19	8
more						
831	"932020W (2022-11-24)60sec.agd"	"21:00"	46	13	1	0

```
## Further filter this data by filtering only those whose total activity at Hip > 0
bis1_filtered1 = filter(:H_total => (x -> x > 0), bis1_filtered)
```

We will use this filtered data set for BIS for the next set of tables and data

```
md"""
We will use this filtered data set for BIS for the next set of tables and data
"""
```

831

```
nrow(bis1_filtered1) # Total number of hours registered = 831
```

[5.9, 3.27778, 2.6087, 1.76471, 1.2, 1.53846, 1.23913, 1.52632, 1.7, 1.65714, 2.19231, 3.

```
# calculate a variable named w_h_ratio i.e., wrist: hip ratio for BIS
# based on Wrist total activities and Hip total activities
bis1_filtered1.w_h_ratio = bis1_filtered1.W_total ./ bis1_filtered1.H_total
```

av_whr =

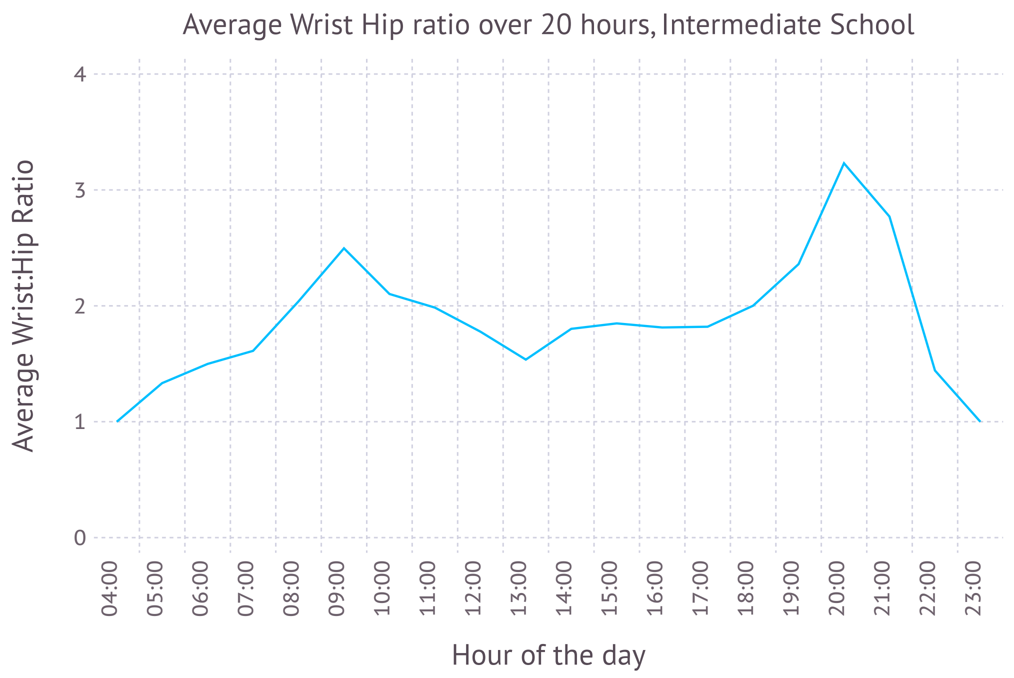
	W_Hour	mean_whr
1	"09:00"	2.49607
2	"10:00"	2.10213
3	"11:00"	1.98505
4	"12:00"	1.77764
5	"13:00"	1.53626
6	"14:00"	1.80143
7	"15:00"	1.84876
8	"16:00"	1.81295
9	"17:00"	1.81995
10	"18:00"	2.00131
	more	
20	"05:00"	1.33333

```
# Aggregate this data set by activities registered at each hour
av_whr = combine(groupby(bis1_filtered1, :W_Hour), :w_h_ratio => mean => :mean_whr )
```

av_whr1 =

	W_Hour	mean_whr
1	"04:00"	1.0
2	"05:00"	1.33333
3	"06:00"	1.49851
4	"07:00"	1.61193
5	"08:00"	2.03815
6	"09:00"	2.49607
7	"10:00"	2.10213
8	"11:00"	1.98505
9	"12:00"	1.77764
10	"13:00"	1.53626
	more	
20	"23:00"	1.0

```
# Sort this data set based on the values of Hours at the Wrist, from lowest to
highest
av_whr1 = sort(av_whr, :W_Hour)
```




```
# Plot how average Wrist to Hip Ratio varies over hours of activity

plot(av_whr1,
x = :W_Hour,
y = :mean_whr,
Geom.line,
Guide.title("Average Wrist Hip ratio over 20 hours, Intermediate School"),
Guide.xlabel("Hour of the day"),
Guide.ylabel("Average Wrist:Hip Ratio"))
```

This plot shows that for BIS, Wrist:Hip ratio drops between 8AM in the morning till about 12 PM then rises between 1:30 PM in the afternoon till about 6 PM in the evening, then sharply rises between 6 PM in the evening till about 10:00 PM at night, then sharply drops.

```
md"""
This plot shows that for BIS, Wrist:Hip ratio drops between 8AM in the morning till about 12 PM then rises between 1:30 PM in the afternoon till about 6 PM in the evening, then sharply rises between 6 PM in the evening till about 10:00 PM at night, then sharply drops.
"""
```

ag_bis_whr =

	W_Filename	mean_whr
1	"932001W.agd"	2.30321
2	"932002W.agd"	1.85222
3	"932006W.agd"	2.91995
4	"932015W.agd"	1.42174
5	"932026W.agd"	1.48725
6	"932027W.agd"	1.80314
7	"932030W.agd"	2.45617
8	"932033W.agd"	1.43169
9	"932034W.agd"	2.25976
10	"932009W (2022-11-25)60sec.agd"	2.79527
11	"932013W (2022-11-25)60sec.agd"	1.64018
12	"932020W (2022-11-24)60sec.agd"	1.59576

```
# aggregate wrist-hip ratio by child for BIS
ag_bis_whr = combine(groupby(bis1_filtered1, :W_Filename), :w_h_ratio => mean =>
:mean_whr)
```

min_whr_bis = 1.4217426901750032

```
# minimum value for Wrist Hip ratio BIS students
min_whr_bis = minimum(ag_bis_whr.mean_whr)
```

max_whr_bis = 2.919949221566353

```
# maximum value for Wrist Hip Ratio, BIS students
max_whr_bis = maximum(ag_bis_whr.mean_whr)
```

mean_whr_bis = 1.997195614564407

```
# Average of the aggregated averages for Wrist Hip ratio for BIS
mean_whr_bis = mean(ag_bis_whr.mean_whr)
```

std_whr_bis = 0.5325669785692786

```
# Standard deviation of the aggregated averages for BIS students
std_whr_bis = std(ag_bis_whr.mean_whr)
```

```
0.15262013716979964
```

```
# assess if the Wrist is statistically significantly higher than Hip
begin
mean_bis_wrist = mean(bis1_filtered1.W_total)
    std_bis_wrist = std(bis1_filtered1.W_total)
    mean_bis_hip = mean(bis1_filtered1.H_total)
    std_bis_hip = std(bis1_filtered1.H_total)
    n_bis_wrist = nrow(bis1_filtered1)
    n_bis_hip = nrow(bis1_filtered1)
    function pooled_sd(n1, n2, s1, s2)
        Pooled_std = sqrt((n1-1) * s1 +  (n2-1) * s2 /  (n1+n2-2))
        return(Pooled_std)
    end
    pooled_std_both = pooled_sd(n_bis_wrist, n_bis_hip, std_bis_wrist, std_bis_hip)
    cohens_d_value = (mean_bis_wrist - mean_bis_hip) / pooled_std_both
    cohens_d_value # 0.16 very low
end
```

```
114.3838598811891
```

```
pooled_sd(n_bis_wrist, n_bis_hip, std_bis_wrist, std_bis_hip)
```

In the above analyses, if we analyse hour by hour and calculate the mean, and standard deviation of the total activity registered in the devices at wrist and hip, we see that there is a small but statistically significant difference between these values for the devices worn at the wrist and hip. But as the values are correlated between one hour and another, therefore, any statistical analysis that does not take into account the multilevel nature of the hour, is uninterpretable. A better way to interpret such findings if we aggregate the children by their identity code, so we know that this is one child who wears the device at wrist and hip and then we can compare.

```
md"""
In the above analyses, if we analyse hour by hour and calculate the mean, and standard deviation of the total activity registered in the devices at wrist and hip, we see that there is a small but statistically significant difference between these values for the devices worn at the wrist and hip. But as the values are correlated between one hour and another, therefore, any statistical analysis that does not take into account the multilevel nature of the hour, is uninterpretable. A better way to interpret such findings if we aggregate the children by their identity code, so we know that this is one child who wears the device at wrist and hip and then we can compare.

"""
```

ag_bis =

	W_Filename	wrist	hip
1	"932001W.agd"	49.3158	25.6316
2	"932002W.agd"	45.92	30.6
3	"932006W.agd"	48.9639	24.012
4	"932015W.agd"	49.8933	37.4933
5	"932026W.agd"	47.2222	33.0556
6	"932027W.agd"	48.169	31.7887
7	"932030W.agd"	44.2571	24.7143
8	"932033W.agd"	45.6027	34.5205
9	"932034W.agd"	45.675	28.0
10	"932009W (2022-11-25)60sec.agd"	44.4359	20.7564
11	"932013W (2022-11-25)60sec.agd"	45.7606	31.4366
12	"932020W (2022-11-24)60sec.agd"	44.9884	32.5814

```
# aggregate over children
ag_bis = combine(groupby(bis1_filtered1, :W_Filename),
[:W_total, :H_total] .=> mean .=> [:wrist, :hip])
```

["wrist_mean: 46.68366083565052, hip_mean: 29.549208950536926, std_md: 3.49356800310333

```
# Calculate the values of total activities at devices in wrist and hip
# and test their statistical difference and standardised values
begin
wrist_b_mean = mean(ag_bis.wrist)
    wrist_b_sd = std(ag_bis.wrist)
    hip_b_mean = mean(ag_bis.hip)
    hip_b_sd = std(ag_bis.hip)
    n_1_b = nrow(ag_bis)
    n_2_b = nrow(ag_bis)
    pooled_sd_b = pooled_sd(n_1_b, n_2_b, wrist_b_sd, hip_b_sd)
    cohens_d_bis = (wrist_b_mean - hip_b_mean) / pooled_sd_b
    ["wrist_mean: $wrist_b_mean, hip_mean: $hip_b_mean, std_md: $cohens_d_bis"]
end
```

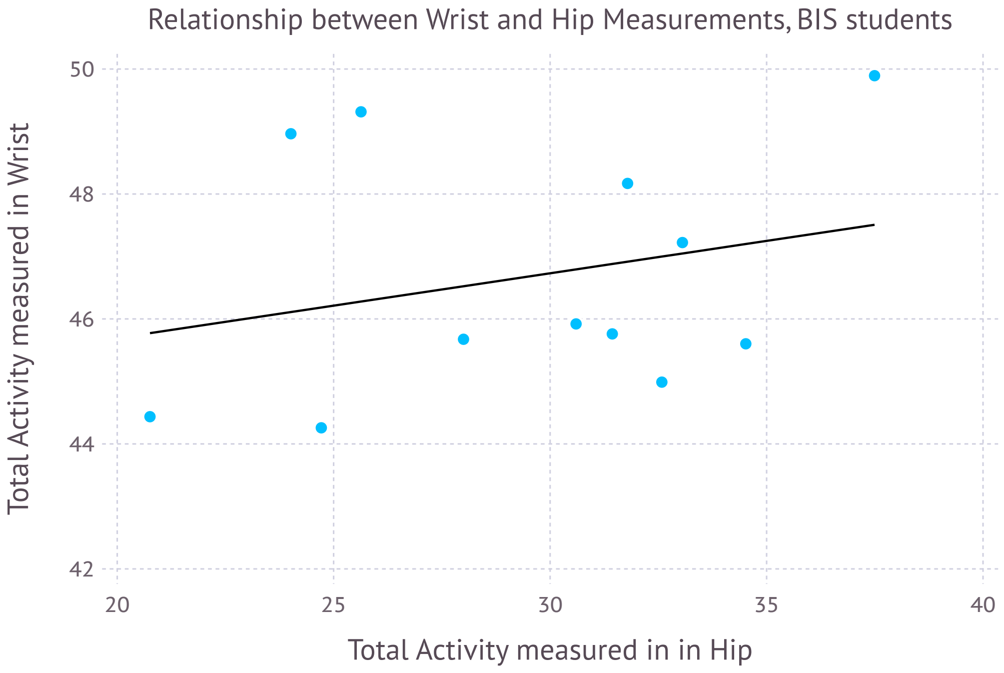
As can be seen, with the pooled sd of 4.9 and mean difference of 19, the standardised mean difference between wrist and hip is 3.49, which is very high for bis students. With 12 participants (11 degrees of freedom), a one-sample t test to measure if these differences are statistically signficant will return a p-value < 0.001

```
md"""

As can be seen, with the pooled sd of 4.9 and mean difference of 19, the
standardised mean difference between wrist and hip is 3.49, which is very high for
bis students.  With 12 participants (11 degrees of freedom), a one-sample t test to
measure if these differences are statistically signficant will return a p-value <
0.001

"""
```

```
using HypothesisTests
```

```
diff_w_h = One sample t-test
           -----------------
           Population details:
               parameter of interest:   Mean
               value under h_0:         0
               point estimate:          17.1345
               95% confidence interval: (14.08, 20.19)

           Test summary:
               outcome with 95% confidence: reject h_0
               two-sided p-value:           <1e-07

           Details:
               number of observations:   12
               t-statistic:              12.330210052441839
               degrees of freedom:       11
               empirical standard error: 1.389631791529807
```

```
diff_w_h = OneSampleTTest(ag_bis.wrist, ag_bis.hip) # < 0.001 (11 degrees of
freedom)
```




```julia
# Plot the graph between Wrist_total and Hip_total
plot(ag_bis,
x = :hip,
y = :wrist,
Geom.point,
layer(Geom.line(Stat.smooth(method = :lm)),
    color=[colorant"black"]
),
    Guide.title("Relationship between Wrist and Hip Measurements, BIS students"),
    Guide.xlabel("Total Activity measured in in Hip"),
    Guide.ylabel("Total Activity measured in Wrist")
)
```

The above plot suggests a linear relationship between Wrist and Hip measurements on the same child. We have also seen that the measurement on the Wrist are higher than the measurements on hip, and such measurements are statistically significant.

```julia
md"""
The above plot suggests a linear relationship between Wrist and Hip measurements on the same child. We have also seen that the measurement on the Wrist are higher than the measurements on hip, and such measurements are statistically significant.

"""
```

Measurement of Total Activity for Primary School Students

```julia
md"""
## Measurement of Total Activity for Primary School Students
"""
```

We examine similar analyses but this time with the primary school students.

```
md"""

We examine similar analyses but this time with the primary school students.

"""
```

```
kpsdf = XLSXFile("/tmp/jl_SRwfevoLSf") containing 7 Worksheets
                        sheetname size          range
        -------------------------------------------------
                        Variables 14x4          A1:D14
                           Hourly 3920x98       A1:CT3920
                            Daily 396x95        A1:CQ396
                          Summary 52x54         A1:BB52
               Sedentary Analysis 1447x35       A1:AI1447
         Wear Time Validation… 52x8             A1:H52
                      Definitions 52x2          A1:B52
```

```
# read the data from the primary school actigraph outputs
kpsdf =
XLSX.readxlsx(download("https://github.com/arinbasu/activation_study/raw/master/KPS_collated_v1.xlsx"))
```

We notice that the hourly data are in the "Hourly" sheet in the spreadsheet. In the next block, we will import the "Hourly" sheet from the spreadsheet.

```
md"""
We notice that the hourly data are in the "Hourly" sheet in the spreadsheet. In the next block, we will import the "Hourly" sheet from the spreadsheet.
"""
```

	W_Filename	W_Epoch	W_Date	W_Hour	W_Day of Week	V
1	"931001W (2023-03-01)60sec.agd"	60	2023-03-01	"09:00"	"Wednesday"	3
2	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"10:00"	"Wednesday"	3
3	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"11:00"	"Wednesday"	3
4	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"12:00"	"Wednesday"	3
5	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"13:00"	"Wednesday"	3
6	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"14:00"	"Wednesday"	3
7	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"15:00"	"Wednesday"	3
8	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"16:00"	"Wednesday"	3
9	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"17:00"	"Wednesday"	3
10	"931001W (2023-03-01)60sec.agd"	60	"01/03/2023"	"18:00"	"Wednesday"	3
more						
3919	"931076W (2023-03-01)60sec.agd"	60	"02/03/2023"	"13:00"	"Thursday"	4

```
# Import the data from the hourly spreadsheet and prepare for analyses
kps =
DataFrame(XLSX.readtable(download("https://github.com/arinbasu/activation_study/raw/master/KPS_collated_v1.xlsx"), "Hourly"))
```

```
foreach(x -> println(x), names(kps))
```

```
H_Light
H_Moderate
H_Vigorous
H_% in Sedentary
H_% in Light
H_% in Moderate
H_% in Vigorous
H_Total MVPA
H_% in MVPA
H_Axis 1 Counts
H_Axis 2 Counts
H_Axis 3 Counts
H_Axis 1 Average Counts
H_Axis 2 Average Counts
H_Axis 3 Average Counts
H_Axis 1 Max Counts
H_Axis 2 Max Counts
H_Axis 3 Max Counts
H_Axis 1 CPM
H_Axis 2 CPM
H_Axis 3 CPM
H_Vector Magnitude Counts
H_Vector Magnitude Average Counts
H_Vector Magnitude Max Counts
H_Vector Magnitude CPM
H_Steps Counts
H_Steps Average Counts
H_Steps Max Counts
H_Steps Per Minute
H_Lux Average Counts
H_Lux Max Counts
H_Number of Epochs
H_Time
H_Calendar Days
```

kps1 =

	W_Filename	W_Hour	W_Sedentary	W_Light	W_Moderate	W
1	"931001W (2023-03-01)60sec.agd"	"09:00"	22	30	8	0
2	"931001W (2023-03-01)60sec.agd"	"10:00"	4	49	7	0
3	"931001W (2023-03-01)60sec.agd"	"11:00"	9	33	4	1
4	"931001W (2023-03-01)60sec.agd"	"12:00"	3	29	8	2(
5	"931001W (2023-03-01)60sec.agd"	"13:00"	3	29	12	1(
6	"931001W (2023-03-01)60sec.agd"	"14:00"	1	15	17	2
7	"931001W (2023-03-01)60sec.agd"	"15:00"	1	11	11	3
8	"931001W (2023-03-01)60sec.agd"	"16:00"	16	23	13	8
9	"931001W (2023-03-01)60sec.agd"	"17:00"	5	36	13	6
10	"931001W (2023-03-01)60sec.agd"	"18:00"	14	16	11	1
more						
3919	"931076W (2023-03-01)60sec.agd"	"13:00"	0	2	8	5(

```
# select variables from kps
kps1 = select(kps, ["W_Filename", "W_Hour", "W_Sedentary",
"W_Light", "W_Moderate", "W_Vigorous", "W_Steps Counts", "H_Filename",
"H_Hour", "H_Sedentary", "H_Light", "H_Moderate", "H_Vigorous",
"H_Steps Counts"])
```

3919

```
# Number of hours
nrow(kps1) # shows 3919 hours of data
```

18

```
# number of variables
ncol(kps1) # we have selected 14 variables for this analyses
```

```
# List all the variables in this new data
foreach(x -> println(x), names(kps1))
```

```
W_Filename
W_Hour
W_Sedentary
W_Light
W_Moderate
W_Vigorous
W_Steps Counts
H_Filename
H_Hour
H_Sedentary
H_Light
H_Moderate
H_Vigorous
H_Steps Counts
```

[38, 56, 51, 57, 57, 59, 59, 44, 55, 46, 59, 27, 55, 51, 57, 55, 57, 59, 58, 58, more ,6

```julia
# create a variable for total activities for Wrist Worn
kps1.W_total = kps1.W_Light + kps1.W_Moderate + kps1.W_Vigorous
```

[16, 23, 31, 36, 33, 48, 52, 28, 34, 34, 55, 6, 29, 13, 12, 11, 40, 39, 31, 33, more ,50

```julia
# create a variable for total activities for hip worn device
kps1.H_total = kps1.H_Light + kps1.H_Moderate + kps1.H_Vigorous
```

[8, 7, 18, 28, 28, 44, 48, 21, 19, 30, 45, 6, 22, 4, 5, 11, 30, 28, 20, 25, more ,28, 43

```julia
# Create a variable for MVPA for Wrist
kps1.W_mvpa = kps1.W_Moderate + kps1.W_Vigorous
```

[0, 0, 2, 2, 0, 1, 7, 3, 1, 8, 22, 0, 0, 0, 0, 0, 1, 0, 1, 0, more ,2, 8, 25, 0, 16, 8, 1

```julia
# Create a variable for MVPA for Hip
kps1.H_mvpa = kps1.H_Moderate + kps1.H_Vigorous
```

kps2 =

	W_Filename	W_Hour	W_Sedentary	W_Light	W_Moderate	V
1	"931001W (2023-03-01)60sec.agd"	"09:00"	22	30	8	0
2	"931001W (2023-03-01)60sec.agd"	"10:00"	4	49	7	0
3	"931001W (2023-03-01)60sec.agd"	"11:00"	9	33	4	1
4	"931001W (2023-03-01)60sec.agd"	"12:00"	3	29	8	2
5	"931001W (2023-03-01)60sec.agd"	"13:00"	3	29	12	1
6	"931001W (2023-03-01)60sec.agd"	"14:00"	1	15	17	2
7	"931001W (2023-03-01)60sec.agd"	"15:00"	1	11	11	3
8	"931001W (2023-03-01)60sec.agd"	"16:00"	16	23	13	8
9	"931001W (2023-03-01)60sec.agd"	"17:00"	5	36	13	6
10	"931001W (2023-03-01)60sec.agd"	"18:00"	14	16	11	1
more						
3844	"931076W (2023-03-01)60sec.agd"	"13:00"	0	2	8	5

```julia
# remove missing values and store the data to kps2
kps2 = dropmissing(kps1)
```

find_both (generic function with 1 method)

```julia
# define a function where BOTH X AND Y must be > 0
function find_both(x,y)
    z = (x > 0 && y > 0)
    return(z)
end
```

	W_Filename	W_Hour	W_Sedentary	W_Light	W_Moderate	W
1	"931001W (2023-03-01)60sec.agd"	"09:00"	22	30	8	0
2	"931001W (2023-03-01)60sec.agd"	"10:00"	4	49	7	0
3	"931001W (2023-03-01)60sec.agd"	"11:00"	9	33	4	1
4	"931001W (2023-03-01)60sec.agd"	"12:00"	3	29	8	2
5	"931001W (2023-03-01)60sec.agd"	"13:00"	3	29	12	1
6	"931001W (2023-03-01)60sec.agd"	"14:00"	1	15	17	2
7	"931001W (2023-03-01)60sec.agd"	"15:00"	1	11	11	3
8	"931001W (2023-03-01)60sec.agd"	"16:00"	16	23	13	8
9	"931001W (2023-03-01)60sec.agd"	"17:00"	5	36	13	6
10	"931001W (2023-03-01)60sec.agd"	"18:00"	14	16	11	1
more						
3703	"931076W (2023-03-01)60sec.agd"	"13:00"	0	2	8	5

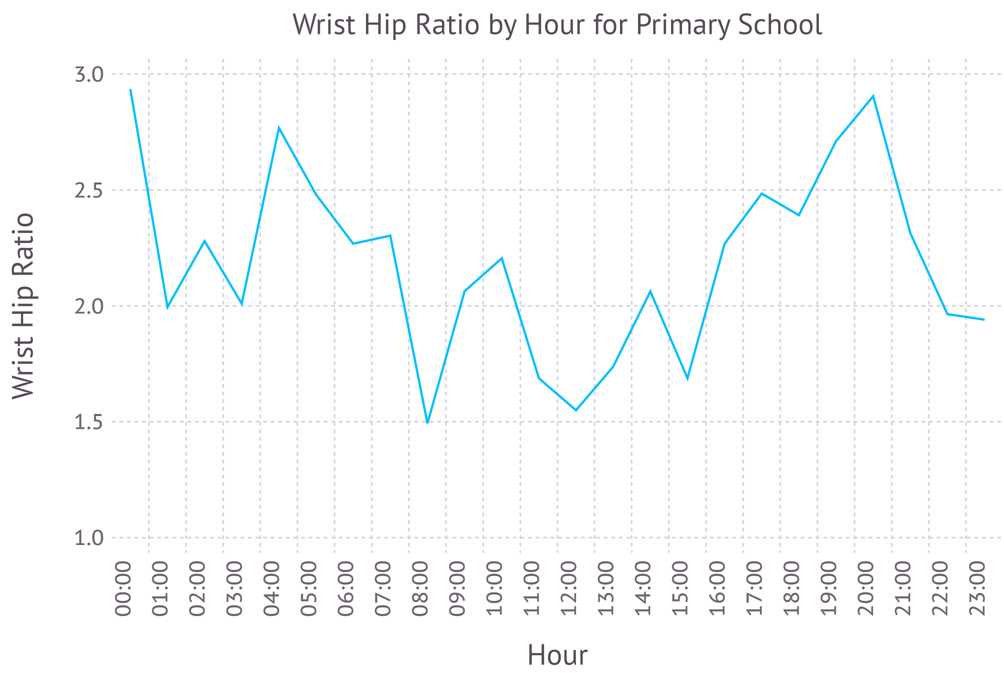
```
# now filter kps to remove 0s from Wrist and Hip data
begin
    kps_filter = filter([:W_total, :H_total] => find_both , kps2)
end
```

```
[2.375, 2.43478, 1.64516, 1.58333, 1.72727, 1.22917, 1.13462, 1.57143, 1.61765, 1.35294,
```

```
# Calculate wrist:hip ratio and save to a variable named w_h_ratio
kps_filter.w_h_ratio = kps_filter.W_total ./ kps_filter.H_total
```

	W_Hour	mean_wh_ratio
1	"00:00"	2.93558
2	"01:00"	1.9947
3	"02:00"	2.27949
4	"03:00"	2.00937
5	"04:00"	2.76744
6	"05:00"	2.48011
7	"06:00"	2.26799
8	"07:00"	2.3028
9	"08:00"	1.49283
10	"09:00"	2.06317
	more	
24	"23:00"	1.94023

```
# Now aggregate average of wrist:hip ratio by the hours and sort the values
begin
av_w_h_ratio_kps = combine(groupby(kps_filter, :W_Hour), :w_h_ratio => mean =>
:mean_wh_ratio)
    av_whr_kps_sorted = sort(av_w_h_ratio_kps, :W_Hour)
end
```




```
# plot the Ratio by Hour
plot(av_whr_kps_sorted,
x = :W_Hour,
y = :mean_wh_ratio,
Geom.line,
Guide.title("Wrist Hip Ratio by Hour for Primary School"),
Guide.xlabel("Hour"),
Guide.ylabel("Wrist Hip Ratio"))
```

As can be seen in the above plot the wrist:hip activity ratio varies over time in a day when the child is active. It follows a complex non-stationary cyclical pattern such that between 12 PM and 8 PM the ratio between Wrist:Hip continues to rise and then reaches a peak around 9 PM, and then drops throughout the night (when theere is little activity). This pattern is similar to what we have seen in BIS students.

```
md"""
As can be seen in the above plot the wrist:hip activity ratio varies over time in a day when the child is active. It follows a complex non-stationary cyclical pattern such that between 12 PM and 8 PM the ratio between Wrist:Hip continues to rise and then reaches a peak around 9 PM, and then drops throughout the night (when theere is little activity).
This pattern is similar to what we have seen in BIS students.

"""
```

ag_child_whr_k =

	W_Filename	mean_wh_ratio
1	"931001W (2023-03-01)60sec.agd"	1.9
2	"931003W (2023-03-01)60sec.agd"	1.55153
3	"931007W (2023-03-01)60sec.agd"	1.43171
4	"931009W (2022-11-25)60sec.agd"	2.00926
5	"931010W (2022-11-25)60sec.agd"	1.79295
6	"931011W (2023-03-01)60sec.agd"	6.08505
7	"931013W (2022-11-25)60sec.agd"	2.12735
8	"931014W (2022-11-25)60sec.agd"	2.2074
9	"931015W (2022-11-25)60sec.agd"	2.06044
10	"931016W (2022-11-25)60sec.agd"	2.42287
more		
48	"931076W (2023-03-01)60sec.agd"	1.46358

```
# Calculate the Wrist Hip Ratio by participant for all hours
# in this case, we aggregate the Wrist:Hip Ratio for each participant

ag_child_whr_k = combine(groupby(kps_filter, :W_Filename), :w_h_ratio => mean =>
:mean_wh_ratio)


```

min_whr_k = 1.2150182492966677

```
# minimum ag_child_whr_k
min_whr_k = minimum(ag_child_whr_k.mean_wh_ratio)
```

max_whr_k = 6.0850468056720945

```
# maximum ag_child_whr_k
max_whr_k = maximum(ag_child_whr_k.mean_wh_ratio)
```

mean_whr_kps = 2.0895390652433643

```
# mean value
mean_whr_kps = mean(ag_child_whr_k.mean_wh_ratio)
```

std_whr_kps = 0.8731940595443839

```
# standard deviation of the ratio
std_whr_kps = std(ag_child_whr_k.mean_wh_ratio)
```

```
0.06281732987488366
```

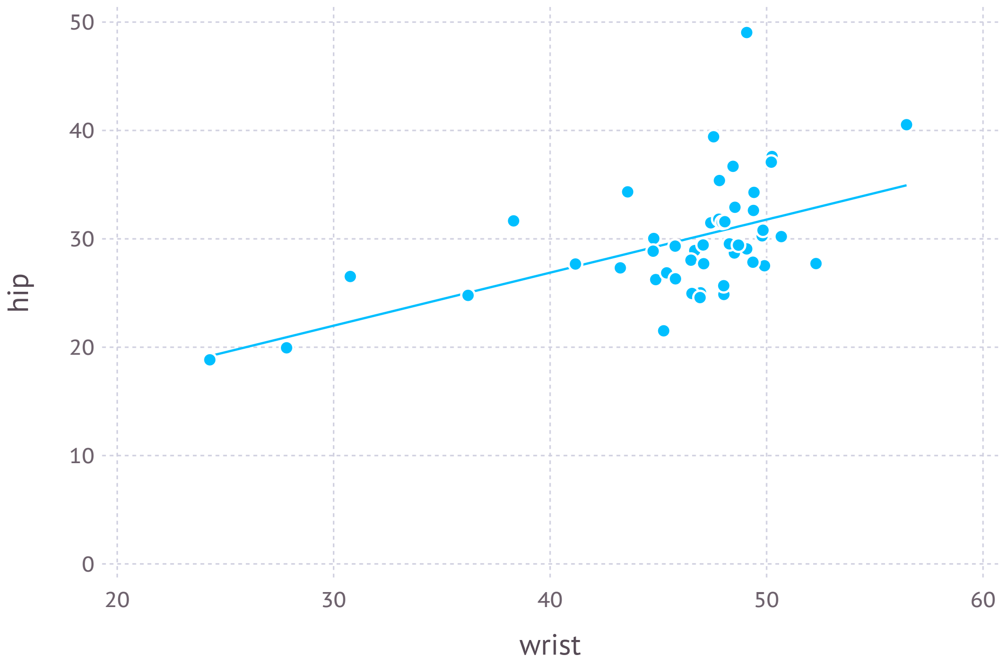
```julia
# using the kps_filter data set, calculate the mean and std for W_total and H_total
begin
    mean_w_total = mean(kps_filter.W_total)
    std_w_total = std(kps_filter.W_total)
    mean_h_total = mean(kps_filter.H_total)
    std_h_total = std(kps_filter.H_total)
    n_total1 = nrow(kps_filter)
    n_total2 = nrow(kps_filter)
    pooled_sd_total = pooled_sd(n_total1, n_total2, std_w_total, std_h_total)
    cohens_d_kps_total = (mean_w_total - mean_h_total) / pooled_sd_total
end

```

```
["wrist_mean: 46.05637969397765, hip_mean: 29.841552260610445, std_md: 0.97448758898239
```

```julia
# do the same but this time with aggregating over W_Filename
begin
    ag_kps = combine(groupby(kps_filter, :W_Filename),
    [:W_total, :H_total] .=> mean .=> [:wrist, :hip])
    wrist_mean = mean(ag_kps.wrist)
    wrist_sd = std(ag_kps.wrist)
    hip_mean = mean(ag_kps.hip)
    hip_sd = std(ag_kps.hip)
    n_1 = nrow(ag_kps)
    n_2 = nrow(ag_kps)
    pooled_sd_all = pooled_sd(n_1, n_2, wrist_sd, hip_sd)
    cohens_d_kps = (wrist_mean - hip_mean) / pooled_sd_all
    ["wrist_mean: $wrist_mean, hip_mean: $hip_mean, std_md: $cohens_d_kps"]
end

```

```
One sample t-test
-----------------
Population details:
    parameter of interest:   Mean
    value under h_0:         0
    point estimate:          16.2148
    95% confidence interval: (14.64, 17.79)

Test summary:
    outcome with 95% confidence: reject h_0
    two-sided p-value:           <1e-24

Details:
    number of observations:   48
    t-statistic:              20.688306337417195
    degrees of freedom:       47
    empirical standard error: 0.7837677559927076
```

```julia
# is this difference statistically significant?
OneSampleTTest(ag_kps.wrist, ag_kps.hip)
```




```
# Let's do a scatterplot of the wrist versus hip
plot(ag_kps,
x = :wrist,
y = :hip,
Geom.point,
Geom.line(Stat.smooth(method = :lm)))
```

As can be seen with KPS, the findings are similar to what we have seen with BIS students. The values are higher in the Wrist than at the hip. There is a variation in the ration of Wrist:Hip over time, and that the value differences are statistically significant.

```
md"""
As can be seen with KPS, the findings are similar to what we have seen with BIS students. The values are higher in the Wrist than at the hip. There is a variation in the ration of Wrist:Hip over time, and that the value differences are statistically significant.

"""
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
Enter cell code...
```

```
# select variables from bis

```

This instance of Pluto runs on JuliaHub!

🚀 A static version of this notebook is **automagically** deployed every time you save & run!

You can extend the time limit of the job serving this pluto notebook by clicking below:

1hour 56 minutes left

extend by 1 hour

Click on the JuliaHub logo to toggle the visibility of this control panel!