Mini Project Samantha Maticka

A

Describe the dataset you have selected. Explain how the data was collected, and explain the meaning of the columns. Do you have any concerns about the data collection process, or about the completeness and accuracy of the data itself?

*Note*: This is also a good time to go through some basic *data cleaning*: if there are columns that are obviously extraneous to the data analysis (e.g., IDs or metadata that have no bearing on your analysis), you can remove those now to make your life easier.

The dataset I’m using was compiled from a collection of GPS-tracked activities, namely runs, bikes, swims, and walks. The activities are from 1 of 2 athletes over the past 5 years. Each activity represents a single observation. The variables associated with each activity are parameters representing the distribution of the entire activity as a whole (e.g. average speed, total distance traveled, etc).

Each .gpx file consists of a latitude, longitude, elevation, with a respective time stamp. These variables were then used to calculate the interval-distance traveled (i.e. from point sample 1 to 2, from 2 to 3, etc.), interval-speeds, and displacement from initial location. Most of the covariates for each observation were derived from these 3 base variables. All covariates in the data set are described in the table below.

|  |  |  |
| --- | --- | --- |
| **Covariate** | **Description** | **Calculation** |
| Continuous | | |
| Umean | Average Speed | mean(speed) |
| Umed | Median Speed | Median(speed) |
| Umax | Maximum Speed | Max(speed) |
| Dtotal | Total Distance Covered | Sum(Distances) |
| dEmax | Maximum Elevation difference | Max(Elevation) – min(Elevation) |
| Grademed | Median Grade [path slope] | Median( |(ΔElevation)/Distance| )\* |
| Ttotal | Total moving-time of activity | Sum(Δt)\*\* |
| Turnsmed | Curviness/small proximity of route  (calculated per 10-min bins) - median | Median( max displacement within 10 min / total distance within 10 min) |
| Turnsmin | Curviness/small proximity of route  (calculated per 10-min bins) - minimum | Min( max displacement within 10 min / total distance within 10 min) |
| Turnsmean | Curviness/small proximity of route  (calculated per 10-min bins) - mean | Mean( max displacement within 10 min / total distance within 10 min) |
| Dispmed | Max displacement within a 10-min window - median | Max(displacement within 10 min) |
| Categorical | | |
| Activity | Type of activity (run, swim, bike, walk) | Known prior to processing |
| Iswim | Swim Indicator | Logical: (Umed < Swim Threshold)\*\*\* |
| Ibike | Bike Indicator | Logical: (Umed > Run Threshold)\*\*\*\* |

\*this is irrespective to whether it was an ascent or descent.

\*\*time steps that indicated no movement (speed < 0.3 m/s) were removed before processing. Therefor the sum of Δt is used as opposed to the difference between end and start time, to only account for moving time.

\*\*\*The median speed is used to account for outliers (e.g. if someone forgot to turn off there app after swimming and before biking home). Threshold choice: Michael Phelps swam ~1.9 m/s for 200 meters - assume if the median speed of activity is greater than this, the athlete is probably not swimming. Note: people can also walk this slow.

\*\*\*\*Usain Bolt ran 12.5 m/s for 100 meters – assume if the median speed of activity is greater than this, the athlete is probably not running, and thus, biking. Note: this is still fast for a biking pace, so may be too strict.

**Concerns about data collection process/ completeness/ accuracy:**

Some things that I’ll have to keep in mind regarding the data collection process, data completion and accuracy are:

1. The scope of athletic ability is very narrow for the training data since only two athletes are used. This will likely make it harder to predict on athletes of different calibers.
2. There are more runs (~700) and bikes (~600) than swims and walks (~100). So the resulting model may be more strongly associated with bike and run covariates than swim and walk covariate vectors.
3. User-introduced error: There is a little bit of uncertainty introduced by the app users. Some activities may be mislabeled (e.g. I found a ‘river rafting’ activity labeled as a swim, and a tour bus ride labeled as a ride). Thankfully some of the mislabeled activities show up as outliers, but that’s not necessarily true for all. Also, sometimes the athlete forgets to stop their watch after, for example swimming, and then they bike home; these impurities make it harder to predict the activity. To account for this, I’m hoping analyzing median values will disregard these extremes.
4. Equipment-introduced error: sometimes satellites signal is not great, or satellite resolution may not be refined, so the data in the .gpx file has some inherent inaccuracy (e.g. any given measured Lat-Lon pair is within plus or minus some degree from the true location, and thus the interpolated elevation will also have some built-in error)
5. The start time of activity is defined in GMT time. The time was able to be adjusted to the proper time zone (since not all activities are in CA), however, daylight saving time is different throughout the world, so the DST adjustment was not accounted for. Therefore, the start time of activity may be off by 1 hour (unless the athlete was in NT, Australia, where it could be off by 1.5 hours – annoying..). It isn’t included in this write up, b/c I just figured out how to adjust it per location. So, hopefully I can add it in for part II.
6. Some of the covariates used may be redundant and/or irrelevant, but they were used in order to determine what is the better metric for prediction. For instance, I’d expect the median speed to be a better metric than the mean speed, since there are likely outliers in every data set.
7. There is likely some, as of yet, unforeseen source of perturbation in the assimilation of the dataset that I’ll keep my eyes peeled for.

B

Are any values in your dataset NULL or NA? Think of what you will do with rows with such entries: do you plan to delete them, or still work with the remaining columns for such rows? (You don’t need to report anything to us for this part.)

There are no NULL or NA values in the dataset. There were a few activities that had no elevation data, and they have already been excluded from the data.

C

*Randomly* choose a test set (representing 20% of your rows), and keep it for later. You will not touch this test set again until the end of the course! Fix this set from the beginning, and use the remaining 80% for exploration, model selection, and validation. (You don’t need to report anything to us for this part.)  *Note*: It may be harder to do this properly with some type of datasets, like time series; if you have selected time series data, let us know and we can help you find a testing strategy. In general, for such data, you want to train on earlier data and test on later data.

D

Compute the mean and variance for each of the columns (you don’t need to report this to us). Are there any columns that appear to be random noise?

There doesn’t appear to be noise. None of the means are near zero, aside from the bike indicator. The threshold used was too strict and not even bike rides registered as a bike ride. There are some covariates (Umax and Dtotal) that have a very large variance. The Umax is possibly from artificial outliers, but the Dtotal is probably due to the large variability arising from 100+ mile bike rides to ~1 mile swims (see below).

Added after submission:

**Umean Umax Umed Tstart Ttotal Dtotal Dispmed dEmax Iswim**

**Mean** 5.2194 29.754 5.2438 11.917 1.3262 27567. 774.54 148.87 0.076857

**SD** 2.2828 323.21 2.4158 3.7562 1.3670 36136. 806.49 248.86 0.26648

**Turnsmed Turnsmin Turnsmean Grademed**

**Mean** 0.62270 0.24086 0.60237 0.019259  
**SD** 0.19145 0.18747 0.16288 0.022421

E

Suggest at least one possibility for a continuous response variable. Explain your choice. Compute the mean and variance of this variable.

Dtotal, total distance covered is a possible response variable that may be able to be inferred given other information about the activity. For example, we might expect a fast run to be a shorter length than a slow run.

mean(df$Dtotal) = 27864.51 meters

var(df$Dtotal) = 1341373455 meters^2

F

Suggest at least one possibility for a binary response variable. Compute the mean of this variable (i.e., the fraction of rows for which this variable is 1.) Explain your choice.  *Note*: You can always create a binary response variable by starting with a continuous variable Y , and then defining Z = 1 if Y exceeds a fixed threshold, and Z = 0 otherwise.

Activity, type of activity is a possibility for a multiclass response variable. While this isn’t binary, the type of classifier needed for multiclass is broken down into binary classification. Below is the mean for the 4 separate cases (i.e. case 1: run = 1, swim/bike/walk = 0, etc).

**Case Mean**

Swim 0.0487

Walk 0.0145

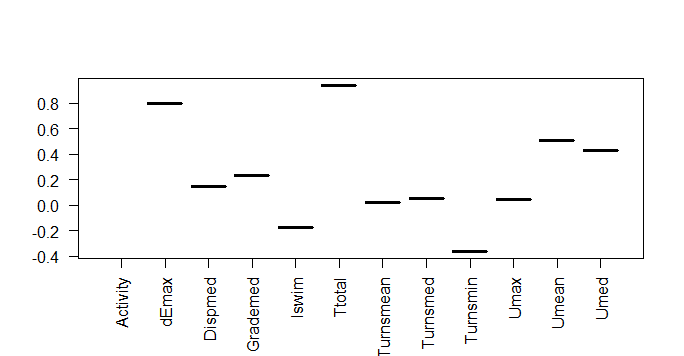
Run 0.5158

Ride 0.4210

Each class likely have key traits about it that separate it from the others, but there is certainly overlap among each of the activities. For example, a swim will be slow, and no elevation change, but this could also be true for a walk, that’s when displacement may be a distinguishing trait.

G

Find the five covariates that are most strongly positively correlated, as well as most strongly negatively correlated, with your choice of continuous response variable. (You should do the same for your choice of binary response variable, but you don’t need to report the results.)  Are there variables you think should affect your response variable, that nevertheless have weak correlation with your response variable?



The top five positively-correlated covariates are Ttotal, dEmax, Umean, Umed, and Grademed.

The fact that dEmax is a strongly associated covariate may not be true for athletes that workout in less hilly regions, like Florida.

The top five negatively-correlated covariates are Turnsmin, Iswim, Turnsmean, Umax, and Turnsmed. For Turn variables, the lower the number the more curvy the path was. This possibly suggests that a route that has more turns (e.g. laps in a pool) or a curvy road/hike path is associated with a shorter activity.

I thought Turnsmed would have the strongest of the Turns variables since it would be more robust to outliers. I thought Iswim would be a stronger indicator, but it may be low since there were far more non-swim activities than there were swims. Lastly, I expected Umed to be a better metric than Umean, and I think for a more general purpose (e.g. to hold up against cases I’m not testing against), Umed still would be better.

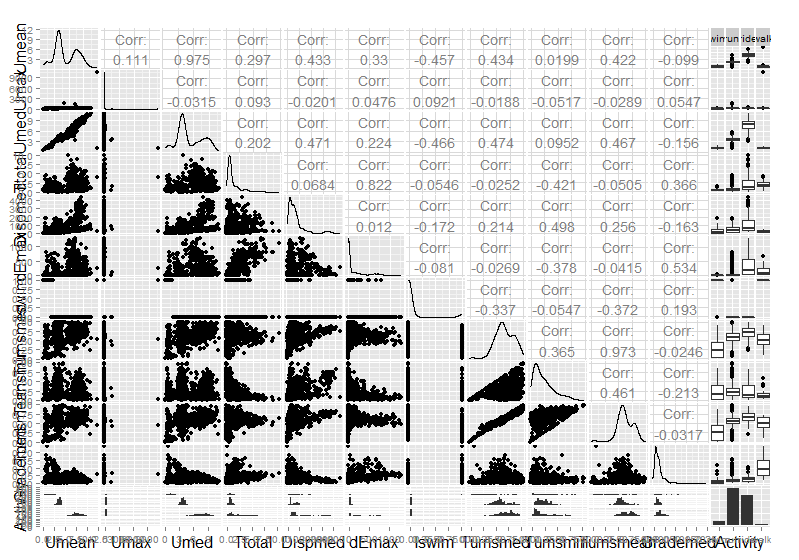
I wasn’t sure how to find correlations between multiclass categorical covariates and the continuous response, so I was unable to report a correlation coefficient for activity type.

H

Now find mutual correlations between the ten variables you identified in the last part. Create a scatterplot for every pair of covariates you believe correlates well to the response variable.  Are correlations *associative* in your data? That is, if A is correlated strongly with B, and B with C, is A also correlated strongly with C in your data?

The ggpairs plot below is of all of the covariates, not including the continuous response variable, Dtotal. Some notable correlations among the covariates are: Umean & Umed (0.975), Ttotal & dEmax (0.822), Turnsmin & Turnsmed (0.973). The strong correlations may suggest collinearity of the covariates. Some of the covariates have a skewed distribution, and transforming them may result in better prediction.

Umed is correlated fairly well with Dispmed (0.471) and with Turnsmean (0.467), however, Dispmed and Turnsmean are not strongly correlated (0.256).



I

Are there variables you would like to *add* to your dataset as you embark on your analysis? For example, are there interactions or higher order terms that might be relevant? (You don’t need to report your answer to this part.)

Yes, I think interaction terms will be helpful in discerning activity type, since it’s the combination of certain traits that distinguish an activity. For example, speed and elevation change.

J

Suggest one or two population models that you think might be relevant for your chosen continuous response variable. (You should do the same for your chosen binary response variable, but don’t need to report the results.) Does your suggestion depend on the desired goal (prediction vs. inference)?  Note that there is no right answer to this question! We just want you to start thinking about what kinds of models might be reasonable to capture relationships between variables in your

data. At this point you don’t have to fit any regressions; it will just be useful to refer back to your answer to this question as you move forward and actually start building models in the next two problem sets.

One population model I would consider is to narrow the single covariates down to ‘Activity’, ‘dEmax’, ‘Iswim’, ‘Ttotal’, ‘Turnsmin’, and ‘Umed’. Then, test the log transform of some of the skewed covariates, against a simple linear regression of all single covariates, with no transform. Then, using the transformed variables, the population model would include all single covariates, all double interactions, and lastly include triple interactions only with categorical variables (i.e. activity and Iswim). Then I would run a backwards regression to decide which model is most fitting. By doing a backward regression, the main goal is prediction.

*Note*: These steps are just the tip of the iceberg! Ideally, you will look at your data many different ways; for example, it’s useful to look at means and variances of columns, grouped based on the level of a categorical variable. (E.g., in the College Scorecard data, you might look at how future earnings differ for public vs. private colleges.)

Try to play with and understand your data as much as you can *before* you start building models!

**MATLAB-code file: GPXprocessing.m**

% this script reads in a gpx file and creates variables

tic

FolderBase **=** 'C:\Users\smaticka\Box Sync\stanford\Classes\MS&E 226 small data\mini project\Adam\_activities\'**;**

% FolderBase = 'C:\Users\smaticka\Box Sync\stanford\Classes\MS&E 226 small data\mini project\Sam\_activities\';

Athlete **=** 'Adam'**;**

Fext **=** **{**'swim'**,**'ride'**,**'run'**,**'walk'**};**

**for** f **=** 1:4

clear activities

cd**(**strcat**(**FolderBase**,** Fext**{**f**}))**

Activity **=** Fext**{**f**};** % activity type variable

BinPoints **=** 90**;** % points in each bin. 90 is 9-10 minutes ish.

SwimThresh **=** 1.9**;** % m/s, Michael Phelps swam for 200 meters

RunThresh **=** 11**;** % m/s, 11 m/s = 24.5mph. Bolt ran 28mph for 100m.. downhill, yea

DispThresh **=** 100**;**

% get list of files in the directory to upload data from

allFiles **=** dir**();**

FileNames **=** **{**allFiles**.**name**};**

FileNames **=** FileNames**(**3**:end);**

**for** k **=** 1**:**length**(**FileNames**)**

k

FileName **=** FileNames**{**k**};**

FileID **=** FileName**(**1**:**15**);**

data **=** gpxread**(**FileName**,** 'FeatureType'**,** 'track'**);**

% pull variables from map data structure

Lat **=** data**.**Latitude**;**

Lon **=** data**.**Longitude**;**

Elevation **=** data**.**Elevation**;** % meters, elevation

% make into column vectors

Lat **=** Lat**(:);**

Lon **=** Lon**(:);**

Elevation **=** Elevation**(:);**

Time **=** strrep**(**data**.**Time**,** 'T'**,** ' '**);**

Time **=** strrep**(**Time**,** 'Z'**,** ''**);**

offset **=** round**(**Lon**(**1**)** **\*** 24**/**360**);**

Time **=** datenum**(**Time**,** 31**)** **+** offset**/**24**;** % convert to local Time from GMT

% Convert time to decimal Hour of day

**[**Y**,**M**,**D**,**h**,**m**,**s**]** **=** datevec**(**Time**(**1**));**

Time **=** **(**Time **-** datenum**(**Y**,**M**,**D**))** **\***24**;** % remove year, month, day

% calculate distance from lat lon and reference ellipsoid

e **=** wgs84Ellipsoid**;**

Distance **=** distance**(**Lat**(**2**:end),**Lon**(**2**:end),**Lat**(**1**:end-**1**),**Lon**(**1**:end-**1**),** e**);** % meters

Displacement **=** distance**(**Lat**(**2**:end),**Lon**(**2**:end),**Lat**(**1**),**Lon**(**1**),** e**);** % meters

Elevation **=** Elevation**(**2**:end);**

% calculate time intervals and resulting speeds

dT **=** **(**Time**(**2**:end)-**Time**(**1**:end-**1**))\***60**\***60**;** % s, time interval of samples

Speed **=** Distance**./**dT**;**

% store variables in a table (data frame)

Time **=** Time**(**2**:end);**

df **=** table**(**Time**,** Distance**,** dT**,** Elevation**,** Speed**,** Displacement**);**

% so i don't accidentally call uncleansed data:

clear Distance Displacement Elevation Speed dT Lat Lon e data Time Y M D h m s

% remove stand-still points

df **=** df**((**df**.**Speed **>** .3**),:);**% .2 m/s:1.3-1.8 is ave walk. .2m/s = 8 min/100 m swim

% calculate total-activity variables

Umean **=** mean**(**df**.**Speed**);** % m/s, average speed

Umed **=** median**(**df**.**Speed**);** % m/s, median speed

Umax **=** max**(**df**.**Speed**);** % m/s, max speed

Dtotal **=** sum**(**df**.**Distance**);** % m, total distance covered

dEmax **=** max**(**df**.**Elevation**)** **-** min**(**df**.**Elevation**);** % m, elevation change

Grademed **=** median**(** abs**(** **(** df**.**Elevation**(**2**:end)** **-** df**.**Elevation**(**1**:end-**1**)** **)** **./(** df**.**Distance**(**2**:end)** **)));**

Tstart **=** df**.**Time**(**1**);** % hrs, start time of activity

Ttotal **=** sum**(**df**.**dT**/**60**/**60**);** % hrs, sum of moving time intervals (not elapsed)

Iswim **=** double**(**Umed **<** SwimThresh**);**% binary, True: speed < than physical swim limit

Ibike **=** double**(**Umed **>** RunThresh**);** % binary, True if speed is more than physical run limit

%% calculate 10-min bin stats (use entire activity if not greater than 10 minutes)

% truncate so that there are constant size bins

**if** size**(**df**,**1**)** **>** BinPoints

df\_bin **=** df**(**1**:end-**rem**(**size**(**df**,**1**),** BinPoints**)** **,** **:);**

Nobs **=** size**(**df\_bin**,**1**);**

% Calculate relevant variables over bin period

Displacement **=** reshape**(**df\_bin**.**Displacement**,** BinPoints**,** Nobs**/**BinPoints**);**

Displacement **=** Displacement **-** meshgrid**(**Displacement**(**1**,:),** 1**:**size**(**Displacement**,**1**));**

Distance **=** reshape**(**df\_bin**.**Distance**,** BinPoints**,** Nobs**/**BinPoints**);**

Elevation **=** reshape**(**df\_bin**.**Elevation**,** BinPoints**,** Nobs**/**BinPoints**);**

Turns **=** max**(**abs**(**Displacement**))./**sum**(**Distance**(**1**:end-**1**,:));**

Turnsmed **=** median**(**Turns**);**

Turnsmin **=** min**(**Turns**);**

Turnsmean **=** mean**(**Turns**);**

Dispmed **=** median**(**max**(**abs**(**Displacement**)));**

% Grademed = median(mean(abs((Elevation(2:end,:)-Elevation(1:end-1,:))./(Distance(2:end,:)))));

**else**

Turns **=** max**(**abs**(**df**.**Displacement**))./**sum**(**df**.**Distance**);**

Turnsmed **=** median**(**Turns**);**

Turnsmin **=** min**(**Turns**);**

Turnsmean **=** mean**(**Turns**);**

Dispmed **=** max**(**abs**(**df**.**Displacement**));**

% Grademed = mean( abs( ( df.Elevation(2:end) - df.Elevation(1:end-1) ) ./( df.Distance(2:end) )));

**end**

activities**(**k**,:)** **=** table**(**Umean**,** Umax**,** Umed**,** Tstart**,** Ttotal**,** Dtotal**,** Dispmed**,**...

dEmax**,** Iswim**,** Ibike**,** Turnsmed**,** Turnsmin**,** Turnsmean**,**...

Grademed**,** **{**Activity**},** **{**Athlete**},** **{**FileID**});**

**end**

writetable**(**activities**,** strcat**(**Athlete**,**Fext**{**f**},**'.csv'**))**

**end**

toc

% Adam's swims took 34 seconds to process (33 files)

% Adam's walks took 131 seconds (24 files)

% Adam's runs took ~1 hr

% Adam's rides took 2 hr 25 min

**R-code file 1: ProcessData.R**

setwd**(**'C:\\Users\\smaticka\\Box Sync\\stanford\\Classes\\MS&E 226 small data\\mini project\\csv files'**)**

df.Sam.swim **<-** read.csv**(**'Samswim.csv', header **=** **TRUE)**

df.S.swim.start **<-** read.csv**(**'SamStartTimeswim.csv',header **=** **TRUE)**

df.Sam.swim **=** cbind**(**df.Sam.swim, df.S.swim.start**)**

df.Adam.swim **<-** read.csv**(**'Adamswim.csv', header **=** **TRUE)**

df.A.swim.start **<-** read.csv**(**'AdamStartTimeswim.csv',header **=** **TRUE)**

df.Adam.swim **=** cbind**(**df.Adam.swim, df.A.swim.start**)**

df.Adam.ride **<-** read.csv**(**'Adamride.csv', header **=** **TRUE)**

df.A.ride.start **<-** read.csv**(**'AdamStartTimeride.csv',header **=** **TRUE)**

df.Adam.ride **=** cbind**(**df.Adam.ride, df.A.ride.start**)**

df.Adam.run **<-** read.csv**(**'Adamrun.csv', header **=** **TRUE)**

df.A.run.start **<-** read.csv**(**'AdamStartTimerun.csv',header **=** **TRUE)**

df.Adam.run **=** cbind**(**df.Adam.run, df.A.run.start**)**

df.Adam.walk **<-** read.csv**(**'Adamwalk.csv', header **=** **TRUE)**

df.A.walk.start **<-** read.csv**(**'AdamStartTimewalk.csv',header **=** **TRUE)**

df.Adam.walk **=** cbind**(**df.Adam.walk, df.A.walk.start**)**

# combine all activities into one data frame

df **=** rbind**(**df.Sam.swim, df.Adam.run, df.Adam.ride, df.Adam.swim, df.Adam.walk**)**

# replace the start time covariate with time

df**$**Tstart **=** df**$**Time

df **=** df**[**,colnames**(**df**)!=**"Time"**]**

# shuffle the data frame (random sample without replacement)

# (column means were the same before and after shuffling. good.)

n **=** dim**(**df**)[**1**]**

df **=** df**[**sample.int**(**n, size **=** n, replace **=** **FALSE)**,**]**

# separate the data into training and test

# use a seed in case I need to rerun

set.seed**(**123**)**

# divide total number of total observations into training and testing

train.ind **=** sample**(**1**:**n, 0.8**\***n**)**

df.train **=** df**[**train.ind,**]**

df.test **=** df**[-**train.ind,**]**

saveRDS**(**df.train, file**=**"..\\TrainingData.rds"**)**

saveRDS**(**df.test, file**=**"..\\TestData.rds"**)**

**R-code file 2: TrainingDataDescription\_partI.R**

# read in training data

setwd**(**'C:\\Users\\smaticka\\Box Sync\\stanford\\Classes\\MS&E 226 small data\\mini project'**)**

.libPaths**(**"C:\\Users\\smaticka\\Documents\\R\\win-library\\3.2"**)**

library**(**ggplot2**)**

library**(**GGally**)**

df **<-** readRDS**(**"TrainingData.rds"**)**

ID **<-** as.data.frame**(**cbind**(**df**$**Athlete, df**$**FileID**))**

#reassign df without IDs

df **=** df**[**,colnames**(**df**)!=**"Athlete"**]**

df **=** df**[**,colnames**(**df**)!=**"FileID"**]**

df **=** df**[**,colnames**(**df**)!=**"Ibike"**]**

# convert Iswim integer to numeric to apply mean to data frame as matrix

df**$**Iswim **=** as.numeric**(**df**$**Iswim**)**

# calculate mean manually (need to change class)

ncov **=** dim**(**df**)[**2**]**

M **<-** apply**(**df**[**,**-**ncov**]**, 2, mean**)**

SD **<-** apply**(**df**[**,**-**ncov**]**, 2, sd**)**

print**(**formatC**(**M,digits**=**5,format**=**"fg",flag**=**"#"**))**

print**(**formatC**(**SD,digits**=**5,format**=**"fg",flag**=**"#"**))**

# separate activities

n **=** nrow**(**df**)**

mean.swim **=** nrow**(**df**[**df**$**Activity **==** "swim",**])/**n

mean.walk **=** nrow**(**df**[**df**$**Activity **==** "walk",**])/**n

mean.run **=** nrow**(**df**[**df**$**Activity **==** "run",**])/**n

mean.ride **=** nrow**(**df**[**df**$**Activity **==** "ride",**])/**n

#calculate cor. coefficient for each covariate Dtotal

df.cov **=** df**[**,colnames**(**df**)!=**"Dtotal"**]**

Ncov **=** length**(**df.cov**)**

coef **=** rep**(NA**,Ncov**)**

**for** **(**i **in** 1**:**Ncov**)** **{**

coef**[**i**]** **=** cor**(**df.cov**[**,i**]**, df**$**Dtotal**)**

**}**

# associate variable name to the coefficient

covar **=** colnames**(**df.cov**)**

plot**(**factor**(**covar**)**, coef, las **=** 2**)**

# mutual correlations

#ggpairs(df.cov, columns = 1:Ncov, las =0)