Mini Project Samantha Maticka

Part 1:

Relevant files: *PullGPX_CreateFeatures.m, 1_AssembleData.R, 2_DescribeData_ggpairs.R*

Dataset Description:

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 athlete over the past 5 years (activities from an additional athlete were also used for 'swims' to increase data points for that category). 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 series of sample points; each point contains latitude, longitude, elevation, and a respective time stamp. These 4 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 4 base variables. All covariates in the data set are described in Table 1.

Including covariates with information about the curviness of an activity is an attempt to capture pool swims and track runs where the displacement is capped but the distance is not. (e.g. 25 yard pool).

Initial Concerns with 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) 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.

6) 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.

Table 1: Covariate Descriptions

*The absolute value makes this is irrespective to whether it was an ascent or descent.
**Time steps with no movement (speed<0.3m/s) were removed. Therefore 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 m - assume if the median speed of activity is greater than this, the athlete is probably not swimming. Note: people can also walk this slow, so *Iswim=1* may also be true for a walk.
****Usain Bolt ran 12.5 m/s for 100 m – assume if the median speed of activity is greater than this, the athlete is probably not running, and is biking. Note: this is fast for biking, so may be too strict.

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)****

Mini Project Samantha Maticka

Categorical Prediction (Logisitic Regression):

Activity type 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). Another way to interpret this is that 4.9% of the activities were swims, 1.5% were walks, 51.6% runs, and 42.1% rides.

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 with 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 (if it is a pool swim).

Data Separation & Basic Covariate Stats:

A few activities had no elevation data, and were excluded from the data. 20% of the remaining data were set aside for testing once the best predictive model has been developed. The remaining 80% of the data was used for training the model. The mean and variance of each of the covariates (training data only) are summarized below.

	Umean (m/s)	**Umax (m/s)**	**Umed (m/s)**	**Tstart (hr)**	**Ttotal (hr)**	**Dtotal (m)**	**Dispmed (m)**
Mean	5.22	29.75	5.24	11.92	1.33	27567	774.54
Stand. Dev.	2.28	323.21	2.42	3.76	1.37	36136	806.49
	dEmax (m)	**Iswim**	**Turnsmed**	**Turnsmin**	**Turnsmean**	**Grademed**	
Mean	148.9	0.08	0.62	0.24	0.60	0.02	
Stand. Dev.	248.86	0.27	0.19	0.19	0.16	0.02	

There does not appear to be random noise, since none of the means are near zero, aside from binary covariates. The threshold used for the bike indicator was too strict and bike rides did not registered as a 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).

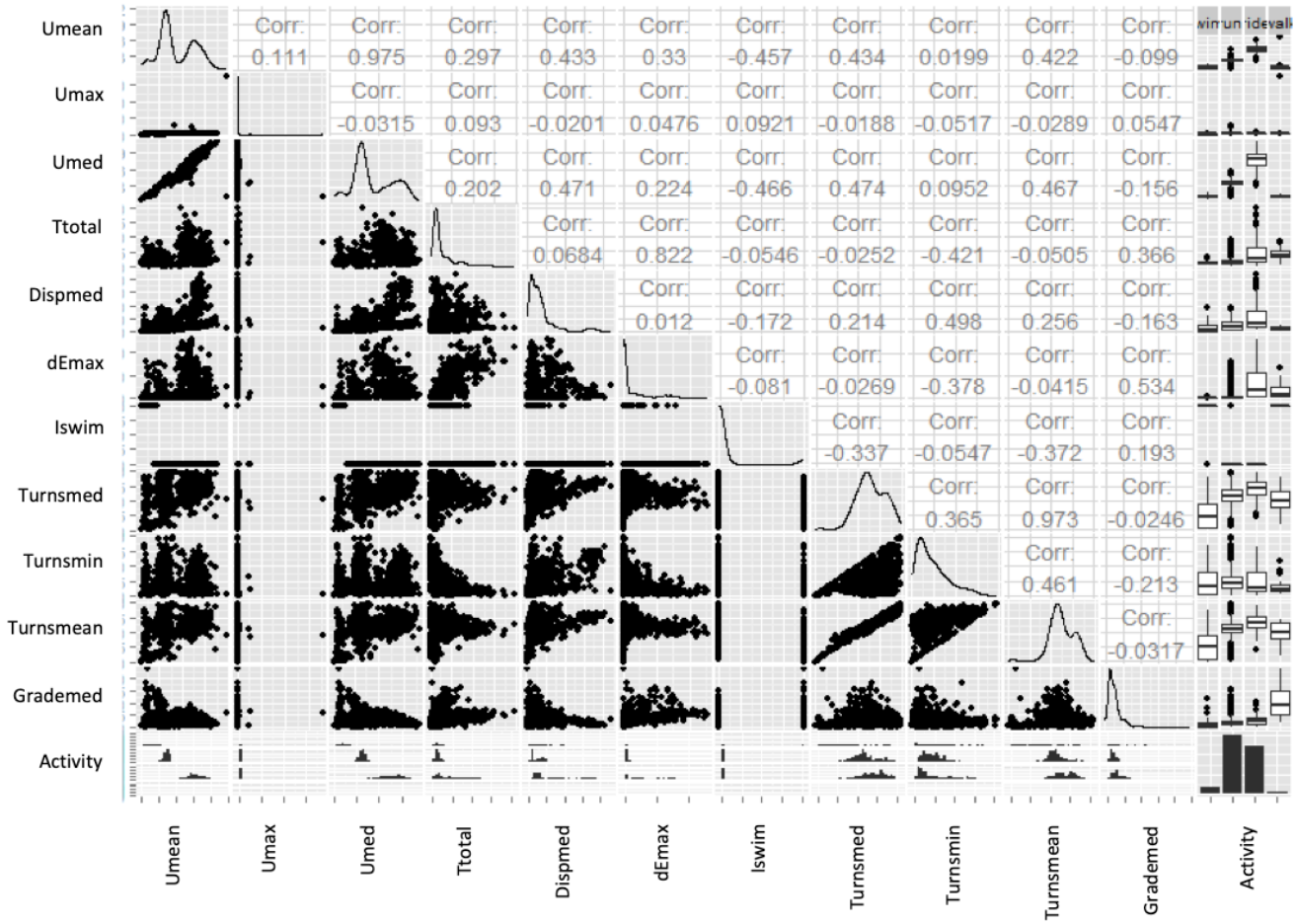
Mini Project Samantha Maticka

Mutual Correlations of Covariates:

The *ggpairs* plot below is of all of the covariates. 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 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.



Part 2:

Relevant files: *3_RemoveCovar_ggpairs.R, 1_BuildingModels.R*

Recap:

My project is looking at the relationship between covariates that describe different physical activities – swimming, biking, running, and walking. The covariates consist of variables detailing speeds, distance, elevation change, etc. throughout the respective activity. Each activity is a single observation - there were ~700 runs, ~600 bikes, ~70 swims, and ~30 walks total. I will be doing a logistic regression (categorical prediction) trying to predict activity type.

Baseline (dummy) Model

Classification problem *(4 classifications: walk, swim, run, bike):*

Since there are 4 separate activities, the activity with the highest empirical frequency is predicted as the dummy model. *Runs* accounts for 51.58% of the observations. When *run* is predicted every time, the Accuracy is 51.58%.

For categorical prediction, accuracy is a good description of how well a model performs. For a binary prediction, this is the sum of true positives and true negatives, divided by the total positives and negatives, Acc=(TP+TN)/(P+N). For multiple categories this accounts for the correct classifications across all cases (i.e. true 'case 1' + true 'case 2' +…). This tells us what percent of the time the model is able to accurately predict an activity. However, we care about the accuracy of prediction per activity as well (e.g. $TP_{run}/(P_{run})$). One thing to strive for is maximum average accuracy (i.e. in the base model, total accuracy is 51%, but the model is 0% accurate at predicted 3 of the 4 activities and 100% at predicting 1, therefore the average accuracy is 25%).

Building the model:

Logistic Regression: Multinomial Classification

To determine the best multinomial classification model, the R-package *glmnet* was used. The model-selection function, *cv.glmnet* determines the *best* model through an iterative process. Different lambdas are used to penalize larger coefficients in the maximum likelihood calculation, and the model corresponding to a lambda that yields the lowest mean CV error is chosen. I used 10 folds, a ridge regression method for lambda, and the iteration was conducted 100 times. As another simple base model, I used 4 covariates (Tstart, Umed, Ttotal, & Turnsmin) that were chosen through a mix of having correlations to activity type, and trial and error. When all individual terms were used in the model (no interaction terms), the error appeared to converge as the *glmnet* method solved for lambda (see below). The final model was chosen as 9 of the original covariates, plus all interaction terms with the 3 covariates Grademed, Iswim, and Umed. See Table 2 for model details.

Training Model Results

Accuracy was calculated by performing cross-validations across 10 folds and the average error metric is what is listed in Table 2. This should be an unbiased estimates of prediction error, since cross-validation was used; hence, the errors are calculated on different data than what was used to train the models.

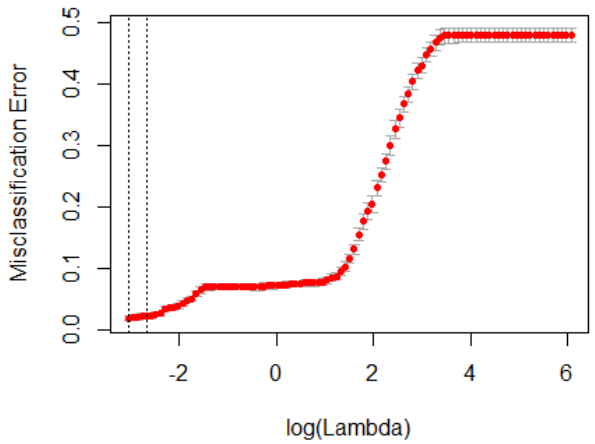


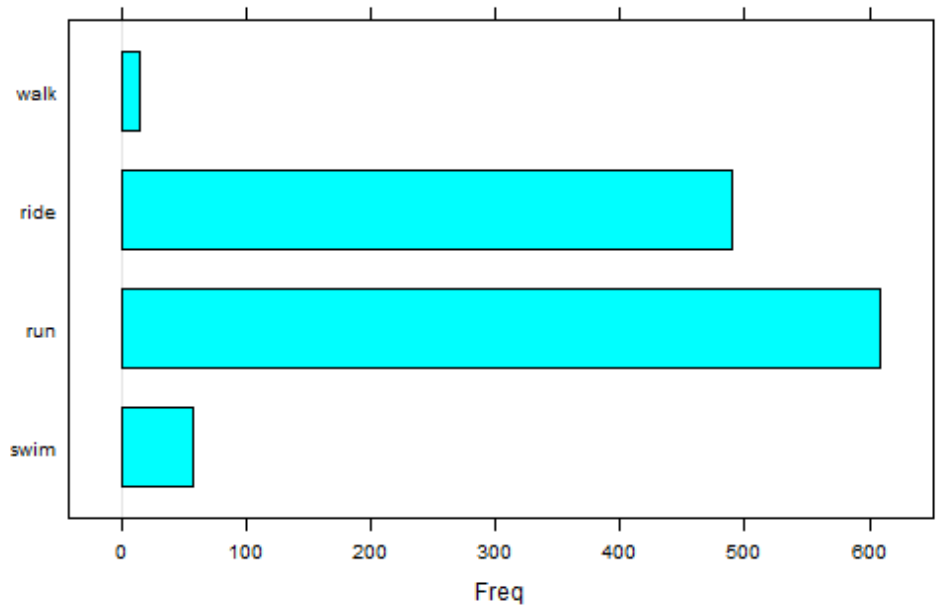
Figure 1: Error convergence for lambda selection

Table 2: Model Descriptions and Training Error

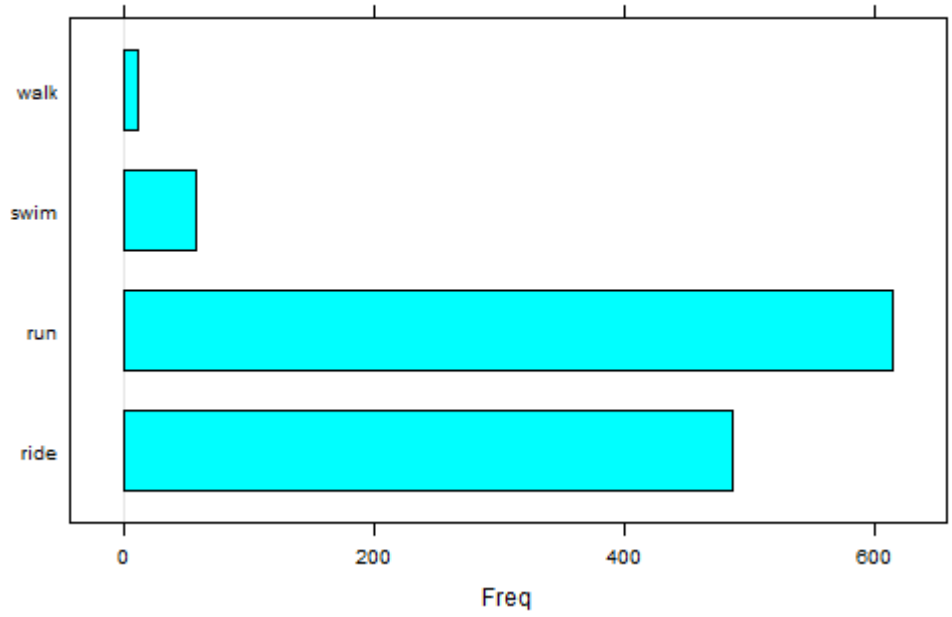
Model Name	Accuracy	Model
Base 1	Total: 51.58% Walk: 0 % Swim: 0 % Run: 100% Ride: 0 % Avg. Accuracy: 25 %	$\widehat{Act} = \{Act_i : P(Act_i) = \max(Act_1, Act_2, Act_3, Act_4)\}$ (Always guess: $Act_i = Act_3$ = 'run')
Base 2	Total: 92.49% Walk: 0 % Swim: 0 % Run: 100% Ride: 96.73 % Avg. Accuracy: 49.18 %	Population model: $P(Y=1\mid\bar{X}) = \frac{\exp(\bar{X}\beta)}{1+\exp(\bar{X}\beta)} = g^{-1}(\bar{X}\beta)$ Likelihood: $P(Y\mid\beta,X) = \prod_{i=1}^{n} g^{-1}(X_i\beta)^{Y_i}(1-g^{-1}(X_i\beta))^{1-Y_i}$ Objective function with ridge regression to minimize: $\frac{-\log(likelihood)}{N_{obs}} + \frac{\lambda}{2}\|\beta\|_2^2$ Base2 model: $\vec{X}\beta = \beta_0 + \beta_1 T_{start} + \beta_2 U_{med} + \beta_3 T_{total} + \beta_4 Turns_{min}$
Chosen Model	Total: 98.21% Walk: 80 % Swim: 91.23 % Run: 99.01% Ride: 98.57 % Avg. Accuracy: 98.57 %	Population model, likelihood, and objective function are same as above (Base 2). Chosen model: $\bar{X}\beta = \beta_0 + \beta_1 T_{start} + \beta_2 U_{med} + \beta_3 T_{total} + \beta_4 Turns_{med} + \beta_5 I_{swim} + \beta_6 Grade_{med} + \beta_7 D_{total} + \ldots$ $\beta_8 Disp_{med} + \beta_9 dE_{\max} +$ *All Interaction Terms with* $Grade_{med}, I_{swim}, U_{med}$

Comparison with observed:

Observed



Modeled



Reflection: Even though CV was used, I do think the estimate for error is optimistic, since my training data set may not be the best representation of the population. More training data would improve this estimate.

Part 3:

Relevant files: *2_TestingModels.R*

Model Testing

Prediction error (accuracy) was calculated by performing cross-validations across 10 folds of the *training data*. Test error (using the reserved 20% of original data) and estimates of prediction error are listed in Table 3.

Since, we care about the accuracy of prediction per activity (e.g. $TP_{run}/(P_{runs})$), and not just overall model accuracy, the metric used to measure the model's prediction ability was the average accuracy of the 4 individual accuracies. The chosen model does estimate an optimistic average accuracy (**98%**) of the test accuracy (**83%**), but the 2 base models are both well below (**50%** and **25%**) the test accuracy. The test accuracy of the base models compares fairly well to the estimated test error of each model.

Summary: The prediction ability of the classifier does quite well. It is much better at predicting *runs* and *rides* than *walks* and *swims*. This is likely due to the smaller availability of the latter 2 activities, as well as the similarity of characteristics between them (e.g. both are slow speeds, and may both be on flat land, etc.).

Table 3: Prediction Error from Training Data & Test Error

Model Name	Estimate of Prediction Error (Accuracy, %)	Test Error (Accuracy, %)
Base 1 (mean)	Total: 51.58 % Walk: 0 % Swim: 0 % Run: 100 % Ride: 0 % **Avg Accuracy: 25 %**	Total: 50.17 % Walk: 0 % Swim: 0 % Run: 100 % Ride: 0 % **Avg Accuracy: 25 %**
Base 2 (a few covariates)	Total: 92.66 % Walk: 0 % Swim: 5.26 % Run: 99.83 % Ride: 96.73 % **Avg Accuracy: 50.46 %**	Total: 91.81 % Walk: 0 % Swim: 0 % Run: 100 % Ride: 96.06 % **Avg Accuracy: 50.46 %**
Chosen model	Total: 98.21 % Walk: 80 % Swim: 91.23 % Run: 99.01 % Ride: 98.57 % **Avg Accuracy: 98.57 %**	Total: 96.25% Walk: 37.50 % Swim: 100 % Run: 98.64% Ride: 96.85 % **Avg Accuracy: 83.24 %**

Ridge regression was used, and the lambda that yielded the lowest cv MSE was chosen, but the subset of covariates was chosen by trial and error of adding/removing covariates to maximize average accuracy. It's likely that this favorable selection led to the optimistic prediction error estimate and possible removal of relevant covariates.

The model was designed only with prediction in mind, rather than inference. Overall, I think this model is a good starting point and a beacon of hope that it could evolve to something more robust, but it has its limitations. To summarize, I'll break it into 1) the limitations I've discovered and 2) some potential extensions that could improve the predictive power.

Limitations:

1) *Sampling bias per athlete*: this model was only trained on 1 athlete. So, while there is clustering per activity type, if more athletes are considered there is a second dimension of clustering related to athletic ability.
2) *Sampling bias per location:* While this is a well-traveled triathlete, most of his activities reside in the Bay Area, where there are ample hills. Take an athlete from Florida, and elevation won't have the same ability to predict.
3) *Sampling bias activity:* There are far more runs and rides that walks and bikes. More of the latter are need for better tuning.

Extension:

1) Guide a predictive model that takes user-defined input. For instance, when an athletes starts using Strava the model will make a guess and based on certainty, it could ask the user 'was this a swim?' This would require training *per athlete*. It may be wise to re-train occasionally over the lifetime of app-usage in case an athlete ramps-up or drops-off their fitness. Not too frequent to annoy users with questions.
2) Geo Mapping the routes could help immensely with predictive power, by associated water and roads to swims and bikes, respectively.

If I were to tackle the same dataset again, I would 1) give myself more time (R took more time than I had anticipated), 2) have employed a naïve Bayes Classifier for continuous variables (assigning a normal distribution to continuous covariates, rather than empirical frequencies) mostly to have a behind the scenes look at the regression process, and 3) extend the dataset by collecting data from more friends.