**Pipeline for naïve imputation.**

1. Data\_processing.r:
   1. Naively imputes the missing data and creates the interaction terms.
   2. *Outputs*
      1. *Data/Train\_val\_processed.csv*
      2. *Data/Test\_from\_train\_processed.csv*
2. Stable Regression (Naïve Impute)
   1. Imports train\_val\_processed.csv from data\_processing.r
   2. Uses optimization to find the toughest training set.
   3. *Outputs* 
      1. *Data/x\_training\_stable.csv*
      2. *Data/y\_training\_stable.csv*
      3. *Data/x\_valid\_set.csv*
      4. *Data/y\_valid\_set.csv*
3. Holistic Regression (Naïve Impute)
   1. Reads in
      1. Data/X\_training\_stable.csv (from stable)
      2. Data/Y\_training\_Stable.csv (from stable)
      3. Data/X\_Val
      4. Data/Y\_val
   2. Conducts transformations of our four numeric variables
   3. Create holistic regression function
   4. Conduct cross validation to find the best gamma (regularization term).
   5. Get betas for that gamma 0.08
   6. Get model performance on test set.
   7. *Outputs*
      1. *Identify store 13* 
         1. *Data/Export store\_13.csv*
      2. *Combine beta0 and beta vector.*
         1. *Export Data/betas.csv*
4. Visibility\_optimization.jl
   1. Reads in Data/store3\_13.csv
   2. Reads in Data/betas.csv
   3. Find best changes for visibility

**ONLY DATA NEEDED INITIALLY IS ORIGINAL TRAIN AND TEST FROM KAGGLE.**

**Pipeline for optimal imputation**

1. Data\_processing\_imputation.r
   1. Reads in
      1. Data/Test.csv
      2. Data/Train.csv
   2. Preprocesses data to make missings “999” to make for easier imputation in Julia.
   3. Other data cleaning tasks(fat content and years.
   4. Create dummies
   5. *Outputs*
      1. *Data/data\_preimpute.csv*
      2. Can be used for both opt imput and knn impute
2. Imputation Work.pynb
   1. Reads in
      1. Data/data\_preimpute.csv
   2. Converts 9999 to missing object
   3. Implement optimal imputation
   4. *Outputs*
      1. *Data/Optimal\_Impute\_Data.csv*
3. Imputed\_data\_post\_processing.R
   1. Reads in
      1. Data/Optimal\_Impute\_Data.csv
   2. Creates all the interaction variables.
   3. *Outputs*
      1. *Data/train\_val\_optimpute.csv*
      2. *Data/test\_optimpute.csv (never used)*
4. Stable\_Reg\_Project\_optimal\_impute (maybe change name lol)
   1. Reads in
      1. Data/train\_val\_optimpute.csv
   2. Conduct stable regression to get toughest train and test set.
   3. *Outputs*
      1. *Data/x\_training\_stable\_optimpute.csv*
      2. *Data/y\_training\_stable\_optimpute.csv*
      3. *Data/x\_valid\_set\_optimpute.csv*
      4. *Data/y\_valid\_set\_optimpute.csv*
5. Hollistic\_Project\_optimal\_impute
   1. Reads in
      1. Data/x\_training\_stable\_optimpute.csv
      2. Data/y\_training\_stable\_optimpute.csv
      3. Data/x\_valid\_set\_optimpute.csv
      4. Data/y\_valid\_set\_optimpute.csv
   2. Implements holistic regression
   3. *Outputs*
      1. *Store/store\_13\_impute.csv*
      2. *Data/betas.csv*
6. Visibility\_optimization\_optimpute.jl
   1. Reads in Data/store3\_13.csv
   2. Reads in Data/betas.csv
   3. Find best changes for visibility