DMC@ISU: The 2015 Iowa State University Data Mining Cup Team

Creating Feature Matrix Version 0.1

Spring 2015, A Team as Strong as Steel

Last Day: May 19, 2015

I am using the following packages:

```
library(magrittr)
library(dplyr)
library(tidyr)
library(lubridate)
library(ggplot2)
library(rCharts)
library(xtable)
library(foreach)
library(gtools)
library(knitr)
library(utils)
source("~/dmc2015/ian/R/renm.R")
```

0.1 I am cleaning the original features

There are some things that I realized last night:

- `sim_ftd` and `sim_lnorm` are identical. They are just scaled versions of the same thing and we can drop them.
- We missed several files, either by their location or structure being problematic.
- We would all feel more comfortable if the orderIDs are on the data.

So I am recreating the feature matrices:

0.1.1 Universal Features

These are the files that I know contain universal features:

```
path.to.files = "../universal/individual/"
files = list.files(path.to.files)
cat(paste(files, collapse = "\n"))
```

```
## basePrice_price_ratio.rds
## basePrice_price_ratio2.rds
## ftd_invmax_similarity.rds
## ftd_ivf_similarity.rds
## ftd_lnorm_similarity.rds
## ftd_pif_similarity.rds
## item_price_basePrice_reward_long.rds
## item_price_basePrice_reward_wide.rds
## jaccard_similarity.rds
## lnorm_invmax_similarity.rds
## lnorm_ivf_similarity.rds
## lnorm_lnorm_similarity.rds
```

```
## lnorm_pif_similarity.rds
## order_match_class.rds
## peteCoupFreq.rds
## user_info.rds
## userTimeFeatures.rds
```

We can read each in as follows:

```
files = list.files("~/dmc2015/features/feature_files/universal/individual/",
    full.name = TRUE)
features = lapply(files, function(x) readRDS(x))
```

Keep in mind that long files have 20,166 rows and wide files have 6722 rows. Wide files should have columns identified as `couponCol`:

```
wideSet = c()
longSet = c()
noSet = c()
for (i in 1:length(features)) {
    if (nrow(features[[i]]) == 20166) {
        longSet = c(longSet, i)
    } else {
        if (nrow(features[[i]]) == 6722) {
            wideSet = c(wideSet, i)
        } else {
            noSet = c(noSet, i)
        }
    }
}

## wide format sets:
wideSet

##  [1]  2  3  4  5  6  8  9 10 11 12 13 14 15 16 17

## long format sets:
longSet

## [1] 1 7

## no format sets:
noSet

## NULL
```

in this case, only the second set /Users/user/dmc2015/features/feature_files/universal/individual//basePrice_price_ratio.rds
/Users/user/dmc2015/features/feature_files/universal/individual//item_price_basePrice_reward_long.rds
is long, and it is kind of broken:

```
badlong = grep("basePrice_price_ratio.rds", files)
features[[badlong]] %>% head

##   orderID bPr2pr_ratio1 bPr2pr_ratio2 bPr2pr_ratio3 bPr2pr_approx_ratio1
## 1       1     1.6666667            NA            NA                   1.7
## 2       1            NA     0.1098266            NA                    NA
## 3       1            NA            NA    1.00000000                    NA
## 4       2     0.6853448            NA            NA                   0.7
```

```
## 5       2            NA     0.5000000            NA                    NA
## 6       2            NA            NA    0.01542416                    NA
##   bPr2pr_approx_ratio2 bPr2pr_approx_ratio3
## 1                   NA                   NA
## 2                  0.1                   NA
## 3                   NA                    1
## 4                   NA                   NA
## 5                  0.5                   NA
## 6                   NA                    0
```

so I will just convert the long format.

```
features[[badlong]] = features[[badlong]] %>% gather(columnName, value, -orderID) %>%
    filter(!is.na(value)) %>% mutate(varName = gsub("ratio[0-9]", "ratio", columnName)) %>%
    mutate(couponCol = 1 * grepl("ratio1", columnName) + 2 * grepl("ratio2",
        columnName) + 3 * grepl("ratio3", columnName)) %>% select(-columnName) %>%
    spread(varName, value) %>% arrange(orderID, couponCol)
```

Combine Wide Sets:

```
wideFrame = features[[wideSet[1]]] %>% select(orderID)
for (x in wideSet) {
    dsn.i = features[[x]]
    dropCols = which(names(dsn.i) %in% names(wideFrame) & names(dsn.i) != "orderID")
    if (length(dropCols) == 0) {
        wideFrame = wideFrame %>% left_join(dsn.i, by = "orderID")
    } else {
        wideFrame = wideFrame %>% left_join(dsn.i[, -dropCols], by = "orderID")
    }
}
names(wideFrame) = gsub("cos\\.sim", "cosSim", names(wideFrame))
names(wideFrame) = gsub("([np])Coup(\\d)(Col\\d)", "\\1times\\3_\\2", names(wideFrame))
names(wideFrame) = gsub("(^.*[^_0-9])(\\d+)$", "\\1_\\2", names(wideFrame))
wideFrame = wideFrame[, which(!grepl("sim_lnorm_.*_", names(wideFrame)))]
```

Long Version

We can also convert this set to the long format. Notice that not all of the files are the same "wide" as others, but we would still like to use them:

```
coln = names(wideFrame)
non_numbered_cols = which(!grepl("_\\d+$", coln))
coupon1_cols = c(1, grep("_(\\d*)1(\\d*)$", coln))
coupon2_cols = c(1, grep("_(\\d*)2(\\d*)$", coln))
coupon3_cols = c(1, grep("_(\\d*)3(\\d*)$", coln))

## coupon 1
coupon1 = wideFrame[, coupon1_cols]
names(coupon1) = gsub("_13$", "_to3Col", names(coupon1))
names(coupon1) = gsub("_12$", "_to2Col", names(coupon1))
names(coupon1) = gsub("_1$", "", names(coupon1))
coupon1$couponCol = 1

## coupon 2
coupon2 = wideFrame[, coupon2_cols]
```

```
names(coupon2) = gsub("_23$", "_to3Col", names(coupon2))
names(coupon2) = gsub("_12$", "_to1Col", names(coupon2))
names(coupon2) = gsub("_2$", "", names(coupon2))
coupon2$couponCol = 2

## coupon 3
coupon3 = wideFrame[, coupon3_cols]
names(coupon3) = gsub("_23$", "_to2Col", names(coupon3))
names(coupon3) = gsub("_13$", "_to3Col", names(coupon3))
names(coupon3) = gsub("_3$", "", names(coupon3))
coupon3$couponCol = 3

coupons = bind_rows(coupon1, coupon2, coupon3)
for (i in grep("_to\\dCol", names(coupons))) is.na(coupons[, i]) = 1

longFrame = wideFrame[, non_numbered_cols] %>% left_join(coupons, by = "orderID") %>%
    arrange(orderID, couponCol)
```

Check Universal Features

We can check the final results:

```
dim(wideFrame)
```

```
## [1] 6722  172
```

```
dim(longFrame)
```

```
## [1] 20166   101
```

Save Universal

These are our universal features:

```
saveRDS(wideFrame, file = "../universal/combined/universalFeaturesCombined_wide.rds")
saveRDS(longFrame, file = "../universal/combined/universalFeaturesCombined_long.rds")
```

0.2 Feature Matrix for Set 1

Again we need to read in the files. In addition to the `llrs` we have:

```
path.to.files = "../set1/individual/"
files = list.files(path.to.files, full.name = TRUE)
cat(paste(files, collapse = "\n"))
```

```
## ../set1/individual//clean_coupon_used_wide.rds
## ../set1/individual//HTVmelt1_Combn_UniqueUser.rds
## ../set1/individual//llrs
```

We repeat the process of creating the wide set and then the long set: We can read each in as follows:

```
path.to.files = "../set1/individual/"
path.to.llrs = "../set1/individual/llrs/"
files = c(list.files(path.to.files, pattern = ".rds", full.name = TRUE), list.files(path.to.llrs,
    pattern = "wide", full.name = TRUE))
features = lapply(files, function(x) readRDS(x))

# make one set of adjustments
features[[2]] = do.call("rbind", features[[2]])
```

Keep in mind that long files have 20,166 rows and wide files have 6722 rows. Wide files should have columns identified as `couponCol`:

```
wideSet = c()
longSet = c()
noSet = c()
for (i in 1:length(features)) {
    if (nrow(features[[i]]) == 20166) {
        longSet = c(longSet, i)
    } else {
        if (nrow(features[[i]]) == 6722) {
            wideSet = c(wideSet, i)
        } else {
            noSet = c(noSet, i)
        }
    }
}

## wide format sets:
wideSet
```

```
##   [1]   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  [18]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [35]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
##  [52]  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
##  [69]  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87
##  [86]  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104
## [103] 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
## [120] 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138
## [137] 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
## [154] 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
## [171] 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
## [188] 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206
## [205] 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
## [222] 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
## [239] 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257
## [256] 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274
## [273] 275 276 277 278
```

```
## long format sets:
longSet
```

```
## NULL
```

```
## no format sets:
noSet
```

```
## [1] 1 2
```

in this case, sets ../set1/individual//clean_coupon_used_wide.rds, ../set1/individual//HTVmelt1_Combn_UniqueUser.rds have no proper place. We can figure out why this is by doing the following:

```
features[[noSet[1]]] %>% names
```

```
##  [1] "orderID"        "nCoupon1Used"   "pCoupon1Used"   "nCoup1Col1Used"
##  [5] "nCoup1Col2Used" "nCoup1Col3Used" "pCoup1Col1Used" "pCoup1Col2Used"
```

```
##  [9] "pCoup1Col3Used" "pCoupon1in1"    "nCoupon2Used"   "pCoupon2Used"
## [13] "nCoup2Col1Used" "nCoup2Col2Used" "nCoup2Col3Used" "pCoup2Col1Used"
## [17] "pCoup2Col2Used" "pCoup2Col3Used" "pCoupon2in2"    "nCoupon3Used"
## [21] "pCoupon3Used"   "nCoup3Col1Used" "nCoup3Col2Used" "nCoup3Col3Used"
## [25] "pCoup3Col1Used" "pCoup3Col2Used" "pCoup3Col3Used" "pCoupon3in3"
```

The first set appears to be wide and should join well. All of the `NA`s would be in the historical rows and wouldn't make their way to our feature matrix.

```
features[[noSet[2]]] %>% names
```

```
##   [1] "orderID"
##   [2] "orderTime"
##   [3] "userID"
##   [4] "couponsReceived"
##   [5] "basketValue"
##   [6] "couponsReceivedDate"
##   [7] "couponsReceivedTime"
##   [8] "couponsReceivedDoW"
##   [9] "couponsReceivedWeekend"
##  [10] "couponsReceivedFriSat"
##  [11] "orderTimeDate"
##  [12] "orderTimeTime"
##  [13] "orderTimeDoW"
##  [14] "orderTimeWeekend"
##  [15] "orderTimeFriSat"
##  [16] "batchID"
##  [17] "couponsExpire"
##  [18] "couponsSent"
##  [19] "TimeBtwnSentRec"
##  [20] "TimeBtwnRecExpire"
##  [21] "TimeBtwnRecOrder"
##  [22] "TimeBtwnOrderExpire"
##  [23] "couponID"
##  [24] "price"
##  [25] "basePrice"
##  [26] "reward"
##  [27] "premiumProduct"
##  [28] "brand"
##  [29] "productGroup"
##  [30] "categoryIDs"
##  [31] "couponUsed"
##  [32] "couponCol"
##  [33] "TimeBtwnRecOrder_disc"
##  [34] "ratio_bp_p"
##  [35] "ratio_bp_p_round"
##  [36] "couponID_nUser"
##  [37] "couponID_nUserUsed"
##  [38] "couponID_Twice"
##  [39] "couponID_prob"
##  [40] "couponsReceivedTime_nUser"
##  [41] "couponsReceivedTime_nUserUsed"
##  [42] "couponsReceivedTime_Twice"
##  [43] "couponsReceivedTime_prob"
##  [44] "couponsReceivedDoW_nUser"
```

```
##  [45] "couponsReceivedDoW_nUserUsed"
##  [46] "couponsReceivedDoW_prob"
##  [47] "orderTimeTime_nUser"
##  [48] "orderTimeTime_nUserUsed"
##  [49] "orderTimeTime_Twice"
##  [50] "orderTimeTime_prob"
##  [51] "orderTimeDoW_nUser"
##  [52] "orderTimeDoW_nUserUsed"
##  [53] "orderTimeDoW_prob"
##  [54] "TimeBtwnRecOrder_disc_nUser"
##  [55] "TimeBtwnRecOrder_disc_nUserUsed"
##  [56] "TimeBtwnRecOrder_disc_Twice"
##  [57] "TimeBtwnRecOrder_disc_prob"
##  [58] "price_nUser"
##  [59] "price_nUserUsed"
##  [60] "price_Twice"
##  [61] "price_prob"
##  [62] "basePrice_nUser"
##  [63] "basePrice_nUserUsed"
##  [64] "basePrice_Twice"
##  [65] "basePrice_prob"
##  [66] "productGroup_nUser"
##  [67] "productGroup_nUserUsed"
##  [68] "productGroup_Twice"
##  [69] "productGroup_prob"
##  [70] "categoryIDs_nUser"
##  [71] "categoryIDs_nUserUsed"
##  [72] "categoryIDs_Twice"
##  [73] "categoryIDs_prob"
##  [74] "ratio_bp_p_round_nUser"
##  [75] "ratio_bp_p_round_nUserUsed"
##  [76] "ratio_bp_p_round_Twice"
##  [77] "ratio_bp_p_round_prob"
##  [78] "reward_couponsReceivedTime_nUser"
##  [79] "reward_couponsReceivedTime_nUserUsed"
##  [80] "reward_couponsReceivedTime_Twice"
##  [81] "reward_couponsReceivedTime_prob"
##  [82] "reward_couponsReceivedDoW_nUser"
##  [83] "reward_couponsReceivedDoW_nUserUsed"
##  [84] "reward_couponsReceivedDoW_Twice"
##  [85] "reward_couponsReceivedDoW_prob"
##  [86] "reward_orderTimeTime_nUser"
##  [87] "reward_orderTimeTime_nUserUsed"
##  [88] "reward_orderTimeTime_Twice"
##  [89] "reward_orderTimeTime_prob"
##  [90] "reward_orderTimeDoW_nUser"
##  [91] "reward_orderTimeDoW_nUserUsed"
##  [92] "reward_orderTimeDoW_Twice"
##  [93] "reward_orderTimeDoW_prob"
##  [94] "reward_TimeBtwnRecOrder_disc_nUser"
##  [95] "reward_TimeBtwnRecOrder_disc_nUserUsed"
##  [96] "reward_TimeBtwnRecOrder_disc_Twice"
##  [97] "reward_TimeBtwnRecOrder_disc_prob"
##  [98] "reward_price_nUser"
```

```
##  [99] "reward_price_nUserUsed"
## [100] "reward_price_Twice"
## [101] "reward_price_prob"
## [102] "reward_basePrice_nUser"
## [103] "reward_basePrice_nUserUsed"
## [104] "reward_basePrice_Twice"
## [105] "reward_basePrice_prob"
## [106] "reward_productGroup_nUser"
## [107] "reward_productGroup_nUserUsed"
## [108] "reward_productGroup_Twice"
## [109] "reward_productGroup_prob"
## [110] "reward_categoryIDs_nUser"
## [111] "reward_categoryIDs_nUserUsed"
## [112] "reward_categoryIDs_Twice"
## [113] "reward_categoryIDs_prob"
## [114] "reward_ratio_bp_p_round_nUser"
## [115] "reward_ratio_bp_p_round_nUserUsed"
## [116] "reward_ratio_bp_p_round_Twice"
## [117] "reward_ratio_bp_p_round_prob"
## [118] "reward_brand_nUser"
## [119] "reward_brand_nUserUsed"
## [120] "reward_brand_Twice"
## [121] "reward_brand_prob"
## [122] "reward_premiumProduct_nUser"
## [123] "reward_premiumProduct_nUserUsed"
## [124] "reward_premiumProduct_Twice"
## [125] "reward_premiumProduct_prob"
## [126] "couponID_couponsReceivedTime_nUserUsed"
## [127] "couponID_couponsReceivedTime_Twice"
## [128] "couponID_couponsReceivedTime_prob"
## [129] "couponID_couponsReceivedDoW_nUser"
## [130] "couponID_couponsReceivedDoW_nUserUsed"
## [131] "couponID_couponsReceivedDoW_Twice"
## [132] "couponID_couponsReceivedDoW_prob"
## [133] "couponID_orderTimeTime_nUserUsed"
## [134] "couponID_orderTimeTime_prob"
## [135] "couponID_orderTimeDoW_nUser"
## [136] "couponID_orderTimeDoW_nUserUsed"
## [137] "couponID_orderTimeDoW_Twice"
## [138] "couponID_orderTimeDoW_prob"
## [139] "couponID_TimeBtwnRecOrder_disc_nUser"
## [140] "couponID_TimeBtwnRecOrder_disc_nUserUsed"
## [141] "couponID_TimeBtwnRecOrder_disc_Twice"
## [142] "couponID_TimeBtwnRecOrder_disc_prob"
## [143] "couponsReceivedTime_couponsReceivedDoW_nUser"
## [144] "couponsReceivedTime_couponsReceivedDoW_nUserUsed"
## [145] "couponsReceivedTime_couponsReceivedDoW_Twice"
## [146] "couponsReceivedTime_couponsReceivedDoW_prob"
## [147] "couponsReceivedTime_orderTimeTime_nUserUsed"
## [148] "couponsReceivedTime_orderTimeTime_Twice"
## [149] "couponsReceivedTime_orderTimeTime_prob"
## [150] "couponsReceivedTime_orderTimeDoW_nUser"
## [151] "couponsReceivedTime_orderTimeDoW_nUserUsed"
## [152] "couponsReceivedTime_orderTimeDoW_Twice"
```

```
## [153] "couponsReceivedTime_orderTimeDoW_prob"
## [154] "couponsReceivedTime_TimeBtwnRecOrder_disc_nUser"
## [155] "couponsReceivedTime_TimeBtwnRecOrder_disc_nUserUsed"
## [156] "couponsReceivedTime_TimeBtwnRecOrder_disc_Twice"
## [157] "couponsReceivedTime_TimeBtwnRecOrder_disc_prob"
## [158] "couponsReceivedTime_price_nUser"
## [159] "couponsReceivedTime_price_nUserUsed"
## [160] "couponsReceivedTime_price_Twice"
## [161] "couponsReceivedTime_price_prob"
## [162] "couponsReceivedTime_basePrice_nUser"
## [163] "couponsReceivedTime_basePrice_nUserUsed"
## [164] "couponsReceivedTime_basePrice_Twice"
## [165] "couponsReceivedTime_basePrice_prob"
## [166] "couponsReceivedTime_productGroup_nUser"
## [167] "couponsReceivedTime_productGroup_nUserUsed"
## [168] "couponsReceivedTime_productGroup_Twice"
## [169] "couponsReceivedTime_productGroup_prob"
## [170] "couponsReceivedTime_categoryIDs_nUser"
## [171] "couponsReceivedTime_categoryIDs_nUserUsed"
## [172] "couponsReceivedTime_categoryIDs_Twice"
## [173] "couponsReceivedTime_categoryIDs_prob"
## [174] "couponsReceivedTime_ratio_bp_p_round_nUser"
## [175] "couponsReceivedTime_ratio_bp_p_round_nUserUsed"
## [176] "couponsReceivedTime_ratio_bp_p_round_Twice"
## [177] "couponsReceivedTime_ratio_bp_p_round_prob"
## [178] "couponsReceivedTime_brand_nUser"
## [179] "couponsReceivedTime_brand_nUserUsed"
## [180] "couponsReceivedTime_brand_Twice"
## [181] "couponsReceivedTime_brand_prob"
## [182] "couponsReceivedTime_premiumProduct_nUser"
## [183] "couponsReceivedTime_premiumProduct_nUserUsed"
## [184] "couponsReceivedTime_premiumProduct_Twice"
## [185] "couponsReceivedTime_premiumProduct_prob"
## [186] "couponsReceivedDoW_orderTimeTime_nUser"
## [187] "couponsReceivedDoW_orderTimeTime_nUserUsed"
## [188] "couponsReceivedDoW_orderTimeTime_Twice"
## [189] "couponsReceivedDoW_orderTimeTime_prob"
## [190] "couponsReceivedDoW_orderTimeDoW_nUser"
## [191] "couponsReceivedDoW_orderTimeDoW_nUserUsed"
## [192] "couponsReceivedDoW_orderTimeDoW_Twice"
## [193] "couponsReceivedDoW_orderTimeDoW_prob"
## [194] "couponsReceivedDoW_TimeBtwnRecOrder_disc_nUser"
## [195] "couponsReceivedDoW_TimeBtwnRecOrder_disc_nUserUsed"
## [196] "couponsReceivedDoW_TimeBtwnRecOrder_disc_Twice"
## [197] "couponsReceivedDoW_TimeBtwnRecOrder_disc_prob"
## [198] "couponsReceivedDoW_price_nUser"
## [199] "couponsReceivedDoW_price_nUserUsed"
## [200] "couponsReceivedDoW_price_Twice"
## [201] "couponsReceivedDoW_price_prob"
## [202] "couponsReceivedDoW_basePrice_nUser"
## [203] "couponsReceivedDoW_basePrice_nUserUsed"
## [204] "couponsReceivedDoW_basePrice_Twice"
## [205] "couponsReceivedDoW_basePrice_prob"
## [206] "couponsReceivedDoW_productGroup_nUser"
```

```
## [207] "couponsReceivedDoW_productGroup_nUserUsed"
## [208] "couponsReceivedDoW_productGroup_Twice"
## [209] "couponsReceivedDoW_productGroup_prob"
## [210] "couponsReceivedDoW_categoryIDs_nUser"
## [211] "couponsReceivedDoW_categoryIDs_nUserUsed"
## [212] "couponsReceivedDoW_categoryIDs_Twice"
## [213] "couponsReceivedDoW_categoryIDs_prob"
## [214] "couponsReceivedDoW_ratio_bp_p_round_nUser"
## [215] "couponsReceivedDoW_ratio_bp_p_round_nUserUsed"
## [216] "couponsReceivedDoW_ratio_bp_p_round_Twice"
## [217] "couponsReceivedDoW_ratio_bp_p_round_prob"
## [218] "couponsReceivedDoW_brand_nUser"
## [219] "couponsReceivedDoW_brand_nUserUsed"
## [220] "couponsReceivedDoW_brand_Twice"
## [221] "couponsReceivedDoW_brand_prob"
## [222] "couponsReceivedDoW_premiumProduct_nUser"
## [223] "couponsReceivedDoW_premiumProduct_nUserUsed"
## [224] "couponsReceivedDoW_premiumProduct_prob"
## [225] "orderTimeTime_orderTimeDoW_nUser"
## [226] "orderTimeTime_orderTimeDoW_nUserUsed"
## [227] "orderTimeTime_orderTimeDoW_Twice"
## [228] "orderTimeTime_orderTimeDoW_prob"
## [229] "orderTimeTime_TimeBtwnRecOrder_disc_nUser"
## [230] "orderTimeTime_TimeBtwnRecOrder_disc_nUserUsed"
## [231] "orderTimeTime_TimeBtwnRecOrder_disc_Twice"
## [232] "orderTimeTime_TimeBtwnRecOrder_disc_prob"
## [233] "orderTimeTime_price_nUser"
## [234] "orderTimeTime_price_nUserUsed"
## [235] "orderTimeTime_price_Twice"
## [236] "orderTimeTime_price_prob"
## [237] "orderTimeTime_basePrice_nUser"
## [238] "orderTimeTime_basePrice_nUserUsed"
## [239] "orderTimeTime_basePrice_Twice"
## [240] "orderTimeTime_basePrice_prob"
## [241] "orderTimeTime_productGroup_nUser"
## [242] "orderTimeTime_productGroup_nUserUsed"
## [243] "orderTimeTime_productGroup_Twice"
## [244] "orderTimeTime_productGroup_prob"
## [245] "orderTimeTime_categoryIDs_nUser"
## [246] "orderTimeTime_categoryIDs_nUserUsed"
## [247] "orderTimeTime_categoryIDs_Twice"
## [248] "orderTimeTime_categoryIDs_prob"
## [249] "orderTimeTime_ratio_bp_p_round_nUser"
## [250] "orderTimeTime_ratio_bp_p_round_nUserUsed"
## [251] "orderTimeTime_ratio_bp_p_round_Twice"
## [252] "orderTimeTime_ratio_bp_p_round_prob"
## [253] "orderTimeTime_brand_nUser"
## [254] "orderTimeTime_brand_nUserUsed"
## [255] "orderTimeTime_brand_Twice"
## [256] "orderTimeTime_brand_prob"
## [257] "orderTimeTime_premiumProduct_nUser"
## [258] "orderTimeTime_premiumProduct_nUserUsed"
## [259] "orderTimeTime_premiumProduct_Twice"
## [260] "orderTimeTime_premiumProduct_prob"
```

```
## [261] "orderTimeDoW_TimeBtwnRecOrder_disc_nUser"
## [262] "orderTimeDoW_TimeBtwnRecOrder_disc_nUserUsed"
## [263] "orderTimeDoW_TimeBtwnRecOrder_disc_Twice"
## [264] "orderTimeDoW_TimeBtwnRecOrder_disc_prob"
## [265] "orderTimeDoW_price_nUser"
## [266] "orderTimeDoW_price_nUserUsed"
## [267] "orderTimeDoW_price_Twice"
## [268] "orderTimeDoW_price_prob"
## [269] "orderTimeDoW_basePrice_nUser"
## [270] "orderTimeDoW_basePrice_nUserUsed"
## [271] "orderTimeDoW_basePrice_Twice"
## [272] "orderTimeDoW_basePrice_prob"
## [273] "orderTimeDoW_productGroup_nUser"
## [274] "orderTimeDoW_productGroup_nUserUsed"
## [275] "orderTimeDoW_productGroup_Twice"
## [276] "orderTimeDoW_productGroup_prob"
## [277] "orderTimeDoW_categoryIDs_nUser"
## [278] "orderTimeDoW_categoryIDs_nUserUsed"
## [279] "orderTimeDoW_categoryIDs_Twice"
## [280] "orderTimeDoW_categoryIDs_prob"
## [281] "orderTimeDoW_ratio_bp_p_round_nUser"
## [282] "orderTimeDoW_ratio_bp_p_round_nUserUsed"
## [283] "orderTimeDoW_ratio_bp_p_round_Twice"
## [284] "orderTimeDoW_ratio_bp_p_round_prob"
## [285] "orderTimeDoW_brand_nUser"
## [286] "orderTimeDoW_brand_nUserUsed"
## [287] "orderTimeDoW_brand_Twice"
## [288] "orderTimeDoW_brand_prob"
## [289] "orderTimeDoW_premiumProduct_nUser"
## [290] "orderTimeDoW_premiumProduct_nUserUsed"
## [291] "orderTimeDoW_premiumProduct_prob"
## [292] "TimeBtwnRecOrder_disc_price_nUser"
## [293] "TimeBtwnRecOrder_disc_price_nUserUsed"
## [294] "TimeBtwnRecOrder_disc_price_Twice"
## [295] "TimeBtwnRecOrder_disc_price_prob"
## [296] "TimeBtwnRecOrder_disc_basePrice_nUser"
## [297] "TimeBtwnRecOrder_disc_basePrice_nUserUsed"
## [298] "TimeBtwnRecOrder_disc_basePrice_Twice"
## [299] "TimeBtwnRecOrder_disc_basePrice_prob"
## [300] "TimeBtwnRecOrder_disc_productGroup_nUser"
## [301] "TimeBtwnRecOrder_disc_productGroup_nUserUsed"
## [302] "TimeBtwnRecOrder_disc_productGroup_Twice"
## [303] "TimeBtwnRecOrder_disc_productGroup_prob"
## [304] "TimeBtwnRecOrder_disc_categoryIDs_nUser"
## [305] "TimeBtwnRecOrder_disc_categoryIDs_nUserUsed"
## [306] "TimeBtwnRecOrder_disc_categoryIDs_Twice"
## [307] "TimeBtwnRecOrder_disc_categoryIDs_prob"
## [308] "TimeBtwnRecOrder_disc_ratio_bp_p_round_nUser"
## [309] "TimeBtwnRecOrder_disc_ratio_bp_p_round_nUserUsed"
## [310] "TimeBtwnRecOrder_disc_ratio_bp_p_round_Twice"
## [311] "TimeBtwnRecOrder_disc_ratio_bp_p_round_prob"
## [312] "TimeBtwnRecOrder_disc_brand_nUser"
## [313] "TimeBtwnRecOrder_disc_brand_nUserUsed"
## [314] "TimeBtwnRecOrder_disc_brand_Twice"
```

```
## [315] "TimeBtwnRecOrder_disc_brand_prob"
## [316] "TimeBtwnRecOrder_disc_premiumProduct_nUser"
## [317] "TimeBtwnRecOrder_disc_premiumProduct_nUserUsed"
## [318] "TimeBtwnRecOrder_disc_premiumProduct_Twice"
## [319] "TimeBtwnRecOrder_disc_premiumProduct_prob"
## [320] "price_basePrice_nUser"
## [321] "price_basePrice_nUserUsed"
## [322] "price_basePrice_Twice"
## [323] "price_basePrice_prob"
## [324] "price_productGroup_nUser"
## [325] "price_productGroup_nUserUsed"
## [326] "price_productGroup_Twice"
## [327] "price_productGroup_prob"
## [328] "price_categoryIDs_nUser"
## [329] "price_categoryIDs_nUserUsed"
## [330] "price_categoryIDs_Twice"
## [331] "price_categoryIDs_prob"
## [332] "price_ratio_bp_p_round_nUser"
## [333] "price_ratio_bp_p_round_nUserUsed"
## [334] "price_ratio_bp_p_round_Twice"
## [335] "price_ratio_bp_p_round_prob"
## [336] "price_brand_nUser"
## [337] "price_brand_nUserUsed"
## [338] "price_brand_Twice"
## [339] "price_brand_prob"
## [340] "price_premiumProduct_nUser"
## [341] "price_premiumProduct_nUserUsed"
## [342] "price_premiumProduct_Twice"
## [343] "price_premiumProduct_prob"
## [344] "basePrice_productGroup_nUser"
## [345] "basePrice_productGroup_nUserUsed"
## [346] "basePrice_productGroup_Twice"
## [347] "basePrice_productGroup_prob"
## [348] "basePrice_categoryIDs_nUser"
## [349] "basePrice_categoryIDs_nUserUsed"
## [350] "basePrice_categoryIDs_Twice"
## [351] "basePrice_categoryIDs_prob"
## [352] "basePrice_ratio_bp_p_round_nUser"
## [353] "basePrice_ratio_bp_p_round_nUserUsed"
## [354] "basePrice_ratio_bp_p_round_Twice"
## [355] "basePrice_ratio_bp_p_round_prob"
## [356] "basePrice_brand_nUser"
## [357] "basePrice_brand_nUserUsed"
## [358] "basePrice_brand_Twice"
## [359] "basePrice_brand_prob"
## [360] "basePrice_premiumProduct_nUser"
## [361] "basePrice_premiumProduct_nUserUsed"
## [362] "basePrice_premiumProduct_Twice"
## [363] "basePrice_premiumProduct_prob"
## [364] "productGroup_categoryIDs_nUser"
## [365] "productGroup_categoryIDs_nUserUsed"
## [366] "productGroup_categoryIDs_Twice"
## [367] "productGroup_categoryIDs_prob"
## [368] "productGroup_ratio_bp_p_round_nUser"
```

```
## [369] "productGroup_ratio_bp_p_round_nUserUsed"
## [370] "productGroup_ratio_bp_p_round_Twice"
## [371] "productGroup_ratio_bp_p_round_prob"
## [372] "productGroup_brand_nUser"
## [373] "productGroup_brand_nUserUsed"
## [374] "productGroup_brand_Twice"
## [375] "productGroup_brand_prob"
## [376] "productGroup_premiumProduct_nUser"
## [377] "productGroup_premiumProduct_nUserUsed"
## [378] "productGroup_premiumProduct_Twice"
## [379] "productGroup_premiumProduct_prob"
## [380] "categoryIDs_ratio_bp_p_round_nUser"
## [381] "categoryIDs_ratio_bp_p_round_nUserUsed"
## [382] "categoryIDs_ratio_bp_p_round_Twice"
## [383] "categoryIDs_ratio_bp_p_round_prob"
## [384] "categoryIDs_brand_nUser"
## [385] "categoryIDs_brand_nUserUsed"
## [386] "categoryIDs_brand_Twice"
## [387] "categoryIDs_brand_prob"
## [388] "categoryIDs_premiumProduct_nUser"
## [389] "categoryIDs_premiumProduct_nUserUsed"
## [390] "categoryIDs_premiumProduct_Twice"
## [391] "categoryIDs_premiumProduct_prob"
## [392] "ratio_bp_p_round_brand_nUser"
## [393] "ratio_bp_p_round_brand_nUserUsed"
## [394] "ratio_bp_p_round_brand_Twice"
## [395] "ratio_bp_p_round_brand_prob"
## [396] "ratio_bp_p_round_premiumProduct_nUser"
## [397] "ratio_bp_p_round_premiumProduct_nUserUsed"
## [398] "ratio_bp_p_round_premiumProduct_Twice"
## [399] "ratio_bp_p_round_premiumProduct_prob"
## [400] "brand_premiumProduct_nUser"
## [401] "brand_premiumProduct_nUserUsed"
## [402] "brand_premiumProduct_Twice"
## [403] "brand_premiumProduct_prob"
```

The second set appears to be long, so I will just convert the long format.

```
badlong = noSet[2]
features[[badlong]] = features[[badlong]][, -(2:31)] %>% gather(columnName,
    value, -orderID, -couponCol) %>% mutate(varName = paste0(columnName, "_",
    couponCol)) %>% select(orderID, varName, value) %>% spread(varName, value) %>%
    arrange(orderID)
```

Combine Wide Sets:

```
wideFrame = features[[badlong]]
for (x in wideSet) {
    dsn.i = features[[x]]
    dropCols = which(names(dsn.i) %in% names(wideFrame) & names(dsn.i) != "orderID")
    if (length(dropCols) == 0) {
        wideFrame = wideFrame %>% full_join(dsn.i, by = "orderID")
    } else {
        wideFrame = wideFrame %>% full_join(dsn.i[, -dropCols], by = "orderID")
    }
```

```
}
names(wideFrame) = gsub("_[Cc]ol(\\d+)$", "_\\1", names(wideFrame))
names(wideFrame) = gsub("(^.*[^_0-9])(\\d+)$", "\\1_\\2", names(wideFrame))
wideFrame = wideFrame[, which(!grepl("times(N*o*t*)Used*_", names(wideFrame)))]
```

Long Version

We can also convert this set to the long format. Notice that not all of the files are the same "wide" as others, but we would still like to use them:

```
coln = names(wideFrame)
non_numbered_cols = which(!grepl("_\\d+$", coln))
coupon1_cols = c(1, grep("_(\\d*)1(\\d*)$", coln))
coupon2_cols = c(1, grep("_(\\d*)2(\\d*)$", coln))
coupon3_cols = c(1, grep("_(\\d*)3(\\d*)$", coln))

## coupon 1
coupon1 = wideFrame[, coupon1_cols]
names(coupon1) = gsub("_13$", "_to3Col", names(coupon1))
names(coupon1) = gsub("_12$", "_to2Col", names(coupon1))
names(coupon1) = gsub("_1$", "", names(coupon1))
coupon1$couponCol = 1

## coupon 2
coupon2 = wideFrame[, coupon2_cols]
names(coupon2) = gsub("_23$", "_to3Col", names(coupon2))
names(coupon2) = gsub("_12$", "_to1Col", names(coupon2))
names(coupon2) = gsub("_2$", "", names(coupon2))
coupon2$couponCol = 2

## coupon 3
coupon3 = wideFrame[, coupon3_cols]
names(coupon3) = gsub("_23$", "_to2Col", names(coupon3))
names(coupon3) = gsub("_13$", "_to3Col", names(coupon3))
names(coupon3) = gsub("_3$", "", names(coupon3))
coupon3$couponCol = 3

coupons = bind_rows(coupon1, coupon2, coupon3)

longFrame = coupons %>% arrange(orderID, couponCol)
```

Check Universal Features

We can check the final results:

```
dim(wideFrame)
```

```
## [1] 6722 3598
```

```
dim(longFrame)
```

```
## [1] 20166  1201
```

Save Universal

These are our universal features:

```
saveRDS(wideFrame, file = "../set1/combined/set1FeaturesCombined_wide.rds")
saveRDS(longFrame, file = "../set1/combined/set1FeaturesCombined_long.rds")
```

0.3 Feature Matrix for Set 2

Again we need to read in the files. In addition to the `llrs` we have:

```
path.to.files = "../set2/individual/"
files = list.files(path.to.files, full.name = TRUE)
cat(paste(files, collapse = "\n"))
```

```
## ../set2/individual//coupon_usr_info.rds
## ../set2/individual//HTVmelt2_Combn_UniqueUser.rds
## ../set2/individual//llrs
```

We repeat the process of creating the wide set and then the long set: We can read each in as follows:

```
path.to.files = "../set2/individual/"
path.to.llrs = "../set2/individual/llrs/"
files = c(list.files(path.to.files, pattern = ".rds", full.name = TRUE), list.files(path.to.llrs,
    pattern = "wide", full.name = TRUE))
features = lapply(files, function(x) readRDS(x))

# make one set of adjustments
features[[2]] = do.call("rbind", features[[2]])
```

Keep in mind that long files have 20,166 rows and wide files have 6722 rows. Wide files should have columns identified as `couponCol`:

```
wideSet = c()
longSet = c()
noSet = c()
for (i in 1:length(features)) {
    if (nrow(features[[i]]) == 20166) {
        longSet = c(longSet, i)
    } else {
        if (nrow(features[[i]]) == 6722) {
            wideSet = c(wideSet, i)
        } else {
            noSet = c(noSet, i)
        }
    }
}

## wide format sets:
wideSet
```

```
##   [1]   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  [18]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [35]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
##  [52]  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
##  [69]  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87
```

```
##  [86]  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104
## [103] 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
## [120] 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138
## [137] 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
## [154] 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172
## [171] 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
## [188] 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206
## [205] 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
## [222] 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
## [239] 241 242
```

```
## long format sets:
longSet
```

```
## NULL
```

```
## no format sets:
noSet
```

```
## [1] 1 2
```

in this case, sets ../set2/individual//coupon_usr_info.rds, ../set2/individual//HTVmelt2_Combn_UniqueUser.rds have no proper place. We can figure out why this is by doing the following:

```
features[[noSet[1]]] %>% names
```

```
## [1] "couponID"  "nUserSent" "nUserUsed" "UsedTwice" "prop"      "prob"
```

The first set appears to be long. We can join by couponID:

```
features[[noSet[1]]] = features[[noSet[2]]][, -c(2:22, 24:31)] %>% arrange(orderID,
    couponCol) %>% left_join(features[[noSet[1]]], by = "couponID") %>% select(-couponID) %>%
    gather(varname, value, -couponCol, -orderID) %>% mutate(colname = paste0(varname,
    "_", couponCol)) %>% select(orderID, colname, value) %>% spread(colname,
    value)
```

```
features[[noSet[1]]] %>% arrange(orderID) %>% select(orderID) %>% head
```

```
##   orderID
## 1       1
## 2       2
## 3       3
## 4       4
## 5       5
## 6       6
```

Combine Wide Sets:

```
wideFrame = features[[noSet[1]]]
for (x in wideSet) {
    dsn.i = features[[x]]
    dropCols = which(names(dsn.i) %in% names(wideFrame) & names(dsn.i) != "orderID")
    dropCols = c(dropCols, which(grepl("times(N*o*t*)Used*_", names(dsn.i))))
    if (length(dropCols) == 0) {
        wideFrame = wideFrame %>% full_join(dsn.i, by = "orderID")
    } else {
        wideFrame = wideFrame %>% full_join(dsn.i[, -dropCols], by = "orderID")
```

```
    }
}
names(wideFrame) = gsub("_[Cc]ol(\\d+)$", "_\\1", names(wideFrame))
names(wideFrame) = gsub("(^.*[^_0-9])(\\d+)$", "\\1_\\2", names(wideFrame))
wideFrame = wideFrame[, which(!grepl("times(N*o*t*)Used*_", names(wideFrame)))]
```

Long Version

We can also convert this set to the long format. Notice that not all of the files are the same "wide" as others, but we would still like to use them:

```
coln = names(wideFrame)
non_numbered_cols = which(!grepl("_\\d+$", coln))
coupon1_cols = c(1, grep("_(\\d*)1(\\d*)$", coln))
coupon2_cols = c(1, grep("_(\\d*)2(\\d*)$", coln))
coupon3_cols = c(1, grep("_(\\d*)3(\\d*)$", coln))

## coupon 1
coupon1 = wideFrame[, coupon1_cols]
names(coupon1) = gsub("_13$", "_to3Col", names(coupon1))
names(coupon1) = gsub("_12$", "_to2Col", names(coupon1))
names(coupon1) = gsub("_1$", "", names(coupon1))
coupon1$couponCol = 1

## coupon 2
coupon2 = wideFrame[, coupon2_cols]
names(coupon2) = gsub("_23$", "_to3Col", names(coupon2))
names(coupon2) = gsub("_12$", "_to1Col", names(coupon2))
names(coupon2) = gsub("_2$", "", names(coupon2))
coupon2$couponCol = 2

## coupon 3
coupon3 = wideFrame[, coupon3_cols]
names(coupon3) = gsub("_23$", "_to2Col", names(coupon3))
names(coupon3) = gsub("_13$", "_to3Col", names(coupon3))
names(coupon3) = gsub("_3$", "", names(coupon3))
coupon3$couponCol = 3

coupons = bind_rows(coupon1, coupon2, coupon3)

longFrame = coupons %>% arrange(orderID, couponCol)
```

Check Universal Features

We can check the final results:

```
dim(wideFrame)
```

```
## [1] 6722 3286
```

```
dim(longFrame)
```

```
## [1] 20166  1097
```

Save Universal

These are our universal features:

```
saveRDS(wideFrame, file = "../set2/combined/set2FeaturesCombined_wide.rds")
saveRDS(longFrame, file = "../set2/combined/set2FeaturesCombined_long.rds")
```

0.4 Feature Matrix for Set 3

Again we need to read in the files. In addition to the `llrs` we have:

```
path.to.files = "../set3/individual/"
files = list.files(path.to.files, full.name = TRUE)
cat(paste(files, collapse = "\n"))
```

```
## ../set3/individual//HTVmelt3_Combn_UniqueUser.rds
## ../set3/individual//llrs
```

We repeat the process of creating the wide set and then the long set: We can read each in as follows:

```
path.to.files = "../set3/individual/"
path.to.llrs = "../set3/individual/llrs/"
files = c(list.files(path.to.files, pattern = ".rds", full.name = TRUE), list.files(path.to.llrs,
    pattern = "wide", full.name = TRUE))
features = lapply(files, function(x) readRDS(x))

# make one set of adjustments
features[[1]] = do.call("rbind", features[[1]])
```

Keep in mind that long files have 20,166 rows and wide files have 6722 rows. Wide files should have columns identified as `couponCol`:

```
wideSet = c()
longSet = c()
noSet = c()
for (i in 1:length(features)) {
    if (nrow(features[[i]]) == 20166) {
        longSet = c(longSet, i)
    } else {
        if (nrow(features[[i]]) == 6722) {
            wideSet = c(wideSet, i)
        } else {
            noSet = c(noSet, i)
        }
    }
}

## wide format sets:
wideSet
```

```
##   [1]   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [18]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
##  [35]  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52
##  [52]  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69
##  [69]  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86
##  [86]  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103
```

```
## [103] 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
## [120] 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137
## [137] 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154
## [154] 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171
## [171] 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188
## [188] 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205
## [205] 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222
## [222] 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
## [239] 240 241
```

```
## long format sets:
longSet
```

```
## NULL
```

```
## no format sets:
noSet
```

```
## [1] 1
```

in this case, sets ../set3/individual//HTVmelt3_Combn_UniqueUser.rds have no proper place. We can figure out why this is by doing the following:

```
features[[noSet[1]]] %>% select(orderID) %>% arrange(orderID) %>% head
```

```
##   orderID
## 1       1
## 2       1
## 3       1
## 4       2
## 5       2
## 6       2
```

The first set appears to be long. We can join by couponID:

```
features[[noSet[1]]] = features[[noSet[1]]][, -c(2:22, 24:31)] %>% arrange(orderID,
    couponCol) %>% select(-couponID) %>% gather(varname, value, -couponCol,
    -orderID) %>% mutate(colname = paste0(varname, "_", couponCol)) %>% select(orderID,
    colname, value) %>% spread(colname, value)

features[[noSet[1]]] %>% arrange(orderID) %>% select(orderID) %>% head
```

```
##   orderID
## 1       1
## 2       2
## 3       3
## 4       4
## 5       5
## 6       6
```

Combine Wide Sets:

```
wideFrame = features[[noSet[1]]]
for (x in wideSet) {
    dsn.i = features[[x]]
    dropCols = which(names(dsn.i) %in% names(wideFrame) & names(dsn.i) != "orderID")
    dropCols = c(dropCols, which(grepl("times(N*o*t*)Used*_", names(dsn.i))))
```

```
    if (length(dropCols) == 0) {
        wideFrame = wideFrame %>% full_join(dsn.i, by = "orderID")
    } else {
        wideFrame = wideFrame %>% full_join(dsn.i[, -dropCols], by = "orderID")
    }
}
names(wideFrame) = gsub("_[Cc]ol(\\d+)$", "_\\1", names(wideFrame))
names(wideFrame) = gsub("(^.*[^_0-9])(\\d+)$", "\\1_\\2", names(wideFrame))
wideFrame = wideFrame[, which(!grepl("times(N*o*t*)Used*_", names(wideFrame)))]
```

Long Version

We can also convert this set to the long format. Notice that not all of the files are the same "wide" as others, but we would still like to use them:

```
coln = names(wideFrame)
non_numbered_cols = which(!grepl("_\\d+$", coln))
coupon1_cols = c(1, grep("_(\\d*)1(\\d*)$", coln))
coupon2_cols = c(1, grep("_(\\d*)2(\\d*)$", coln))
coupon3_cols = c(1, grep("_(\\d*)3(\\d*)$", coln))

## coupon 1
coupon1 = wideFrame[, coupon1_cols]
names(coupon1) = gsub("_13$", "_to3Col", names(coupon1))
names(coupon1) = gsub("_12$", "_to2Col", names(coupon1))
names(coupon1) = gsub("_1$", "", names(coupon1))
coupon1$couponCol = 1

## coupon 2
coupon2 = wideFrame[, coupon2_cols]
names(coupon2) = gsub("_23$", "_to3Col", names(coupon2))
names(coupon2) = gsub("_12$", "_to1Col", names(coupon2))
names(coupon2) = gsub("_2$", "", names(coupon2))
coupon2$couponCol = 2

## coupon 3
coupon3 = wideFrame[, coupon3_cols]
names(coupon3) = gsub("_23$", "_to2Col", names(coupon3))
names(coupon3) = gsub("_13$", "_to3Col", names(coupon3))
names(coupon3) = gsub("_3$", "", names(coupon3))
coupon3$couponCol = 3

coupons = bind_rows(coupon1, coupon2, coupon3)

longFrame = coupons %>% arrange(orderID, couponCol)
```

Check Universal Features

We can check the final results:

```
dim(wideFrame)
```

```
## [1] 6722 3277
```

```
dim(longFrame)
```

```
## [1] 20166  1094
```

Save Universal

These are our universal features:

```
saveRDS(wideFrame, file = "../set3/combined/set3FeaturesCombined_wide.rds")
saveRDS(longFrame, file = "../set3/combined/set3FeaturesCombined_long.rds")
```