# LogCheck

*Chuan Hong*

*3/15/2017*

### Data import and munge

```r
library(car)

POST <- read.csv("REVISED_FINAL_POST_OM.csv",header = TRUE,sep = ",",stringsAsFactors = FALSE)
PRE <- read.csv("REVISED_FINAL_PRE_OM.csv",header = TRUE,sep = ",",stringsAsFactors = FALSE)

Post_proc <- lapply(POST, FUN = function(foo) car::recode(foo, "'Never' = 0 ; 'Annually' = 1 ; 'Seasona
Post_proc <- data.frame(Post_proc)
IsExperimental = Post_proc[2:nrow(Post_proc),c(1,ncol(Post_proc))]
cutdown <- ncol(Post_proc)-3
Post_proc <- Post_proc[,1:cutdown]
best_prac <- as.vector(Post_proc[1,2:cutdown],mode='numeric')
best_prac = append(1, best_prac)
Post_proc <- Post_proc[2:nrow(Post_proc), ]

Pre_proc <- lapply(PRE, FUN = function(foo) car::recode(foo, "'Never' = 0 ; 'never' = 0 ; 'Annually' = 
Pre_proc <- data.frame(Pre_proc)
cutdown <- ncol(Pre_proc)-2
Pre_proc <- Pre_proc[2:nrow(Pre_proc),1:cutdown]

Post_proc2 <- subset(Post_proc, BPL.BLD.ID %in% Pre_proc$BPL.BLD.ID)
Pre_proc2 <- subset(Pre_proc, BPL.BLD.ID %in% Post_proc$BPL.BLD.ID)
Post_score <- data.frame(mapply(`/`,Post_proc2,best_prac))
Pre_score <- data.frame(mapply(`/`,Pre_proc2,best_prac))
Pre_score <- Pre_score[with(Pre_score, order(BPL.BLD.ID)), ]
Post_score <- Post_score[with(Post_score, order(BPL.BLD.ID)), ]
Pre_score[1000 > Pre_score & Pre_score > 1] <- 1
Post_score[1000 > Post_score & Post_score > 1] <- 1

Pre_post = Post_score - Pre_score
Pre_post[,1] = Pre_score[,1]

IsExperimental <- lapply(POST[2:nrow(POST),c(1,ncol(POST))], FUN = function(foo) car::recode(foo, "'' =
IsExperimental <- data.frame(IsExperimental)
IsExperimental <- subset(IsExperimental, BPL.BLD.ID %in% Pre_score$BPL.BLD.ID)
IsExperimental <- IsExperimental[with(IsExperimental, order(BPL.BLD.ID)), ]
```

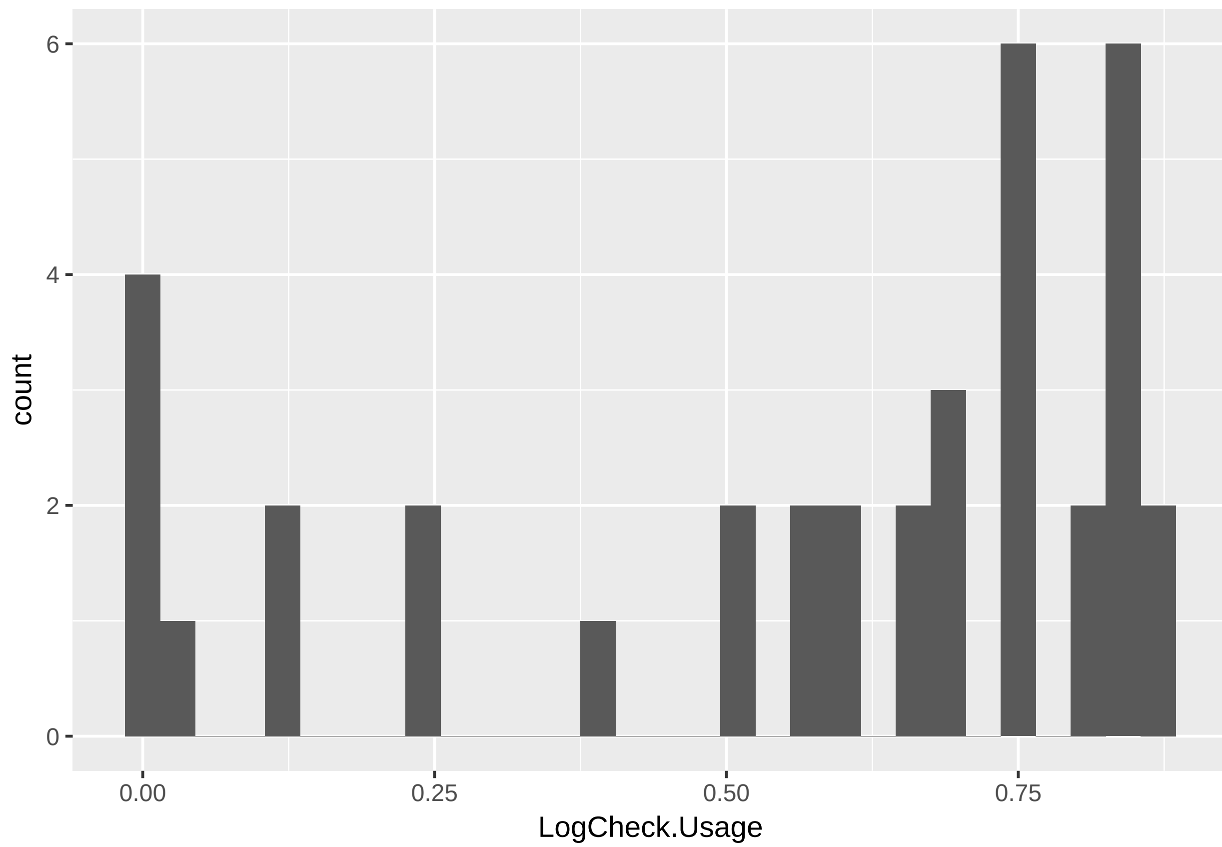
### Load libraries and data

```r
library(data.table)
library(dplyr)
library(ggplot2)
```

```
logcheck.usage <- fread(input = "./LOGCHECK USERS WITH POST.csv",
                        header = TRUE)
```

**- Check the distribution of LogCheck.Usage**

```
ggplot(logcheck.usage,aes(LogCheck.Usage))+geom_histogram()
```




```
summary(logcheck.usage$LogCheck.Usage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3900  0.6800  0.5624  0.8100  0.8700
```

**- Merge Pre_post and logcheck.usage, and list obs having LogCheck.Usage value**

```
pre_post.usage <- merge(Pre_post,logcheck.usage,by='BPL.BLD.ID',all.x = TRUE)
intersect(Pre_post$BPL.BLD.ID,logcheck.usage$BPL.BLD.ID)
```

```
##  [1] 1003 1031 1033 1043 1052 1055 1058 1101 1102 1108 1111 1115 1125 1130
## [15] 1132 1148 1150 1153 1158 1160 1175 1188 1195 1203 1206 1224 1226 1227
## [29] 1282 1300 1313 1321 1322 1323 1328
```

**- Merge Pre_post.usage and IsExperimental, and list TRUE and FALSE**

```
# Merge Pre_post.usage and IsExperimental
pre_post.usage.isexp <- merge(pre_post.usage,IsExperimental,by='BPL.BLD.ID', all.x = TRUE)
```

```
summary(pre_post.usage.isexp$LogCheck)
```

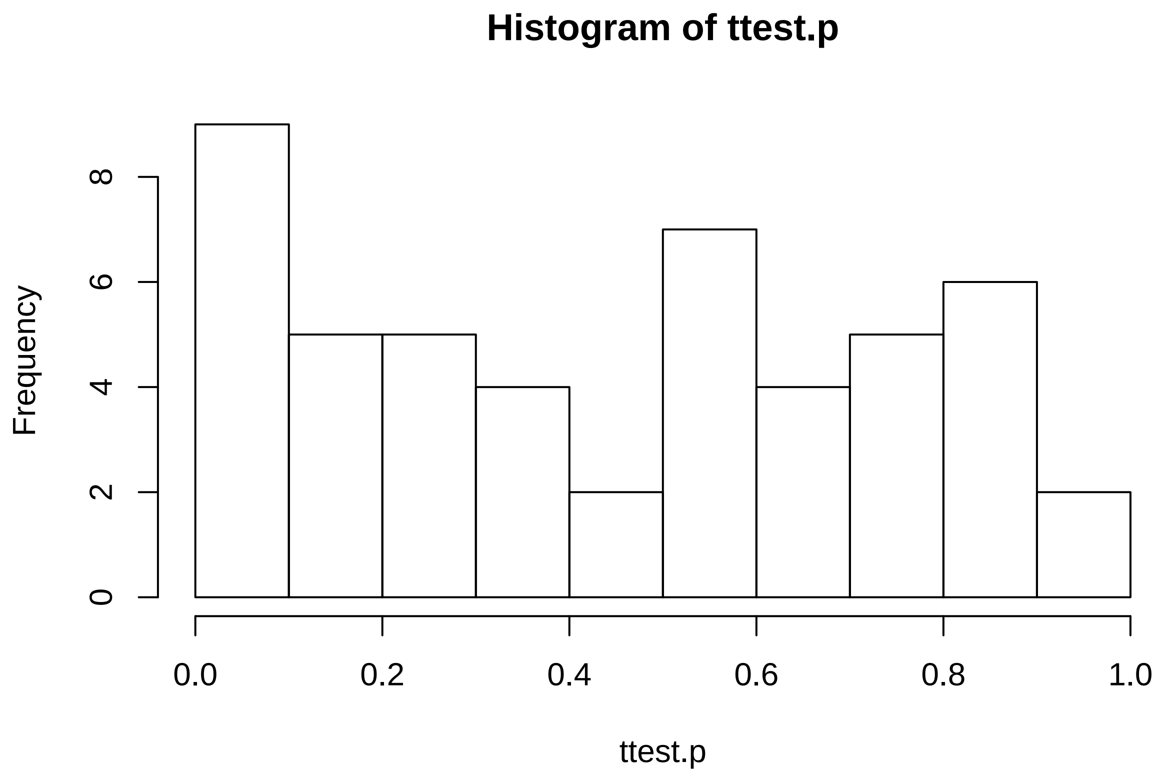
```
## FALSE  TRUE
##   102    35
```

**- Create new df call pre_post.all with all info: usage, isExperimental, group mean (X1, X2, . . . , X8), grand mean (Xall)**

```
pre_post.all <- pre_post.usage.isexp %>%
  mutate(X1 = rowMeans(select(pre_post.usage.isexp, starts_with("X1")), na.rm = TRUE),
         X2 = rowMeans(select(pre_post.usage.isexp, starts_with("X2")), na.rm = TRUE),
         X3 = rowMeans(select(pre_post.usage.isexp, starts_with("X3")), na.rm = TRUE),
         X4 = rowMeans(select(pre_post.usage.isexp, starts_with("X4")), na.rm = TRUE),
         X5 = rowMeans(select(pre_post.usage.isexp, starts_with("X5")), na.rm = TRUE),
         X6 = rowMeans(select(pre_post.usage.isexp, starts_with("X6")), na.rm = TRUE),
         X7 = rowMeans(select(pre_post.usage.isexp, starts_with("X7")), na.rm = TRUE),
         X8 = rowMeans(select(pre_post.usage.isexp, starts_with("X8")), na.rm = TRUE),
         Xall = rowMeans(select(pre_post.usage.isexp, starts_with("X")), na.rm = TRUE))
```

## One Tailed Two-sample t-tests

```
ttest <- function(coln,group){
  col_C <- coln[group == 'FALSE']
  col_E <- coln[group == 'TRUE']
  t.test(col_E,col_C,alternative = 'greater')$p.value
}

ttest.p <- sapply(pre_post.all %>% select(-BPL.BLD.ID, -LogCheck.Usage, -LogCheck),
                  function(x) ttest(x, pre_post.usage.isexp$LogCheck))

hist(ttest.p,width=5)
```



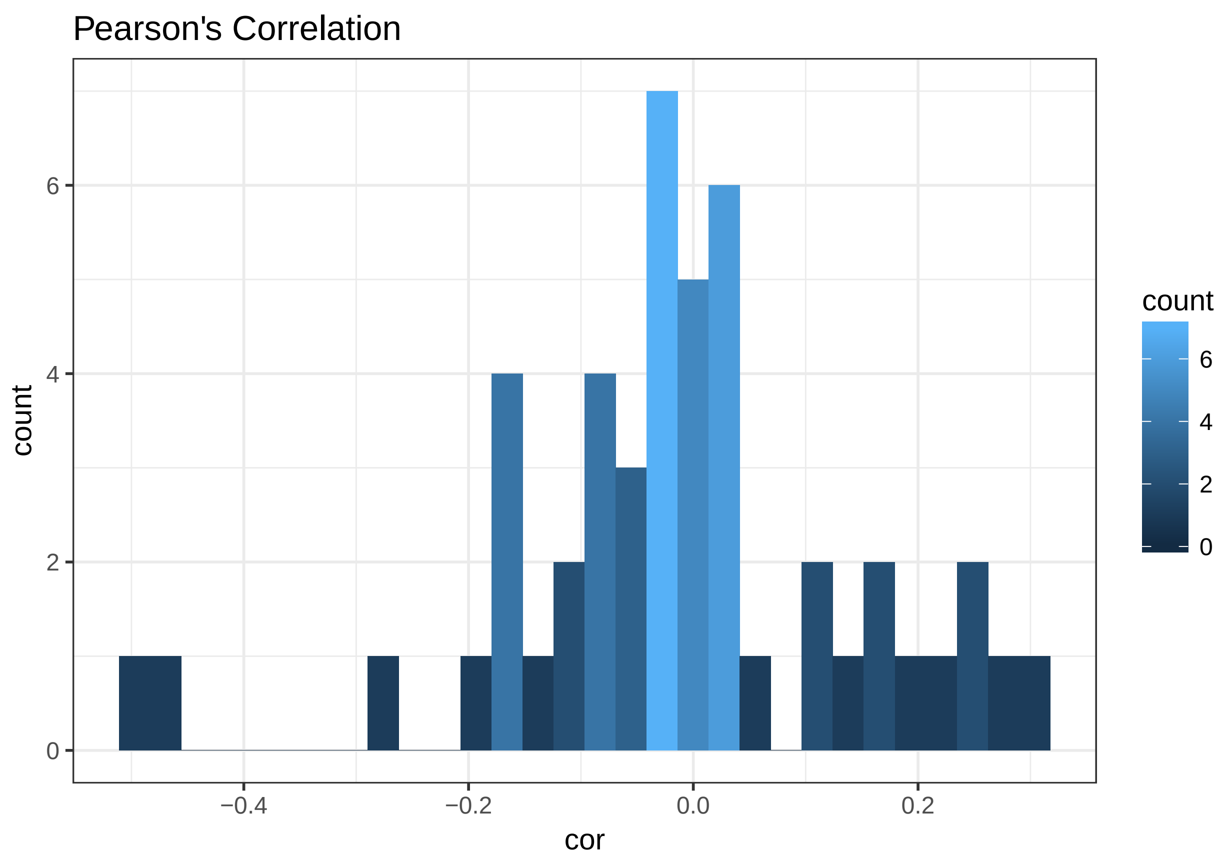

```
signif_p <- Filter(function(x) x < .05, ttest.p)
signif_p
```

```
##         X3.1         X3.6         X4.5         X5.4         X7.2
## 0.0335961958 0.0351338586 0.0109343908 0.0233137261 0.0003613601
##         X7.3           X7
## 0.0130736916 0.0006313622
```

### Pearson's Correlation

Note: Weakly positive correlations are observed between LogCheck.Usage and X1.1, X2.5, X4.3, X4.6, X8.1.

```
pre_post.all_E <- pre_post.all %>%
  filter(LogCheck == "TRUE") %>%
  select(-BPL.BLD.ID, -LogCheck)

pcor_e <- sapply(pre_post.all_E %>% select(-LogCheck.Usage),
       function(x) cor(x, pre_post.all_E$LogCheck.Usage,
                       method = "pearson",
                       use = "pairwise.complete.obs"))

ggplot(data.frame(cor = pcor_e), aes(cor)) +
  geom_histogram(aes(fill = ..count..))+
  theme_bw() + ggtitle("Pearson's Correlation")
```




```
Filter(function(x) x > .2, pcor_e)
```

```
##       X1.1      X2.5      X4.3      X4.6      X8.1
## 0.2307243 0.2393168 0.2666647 0.2388505 0.2952502
```

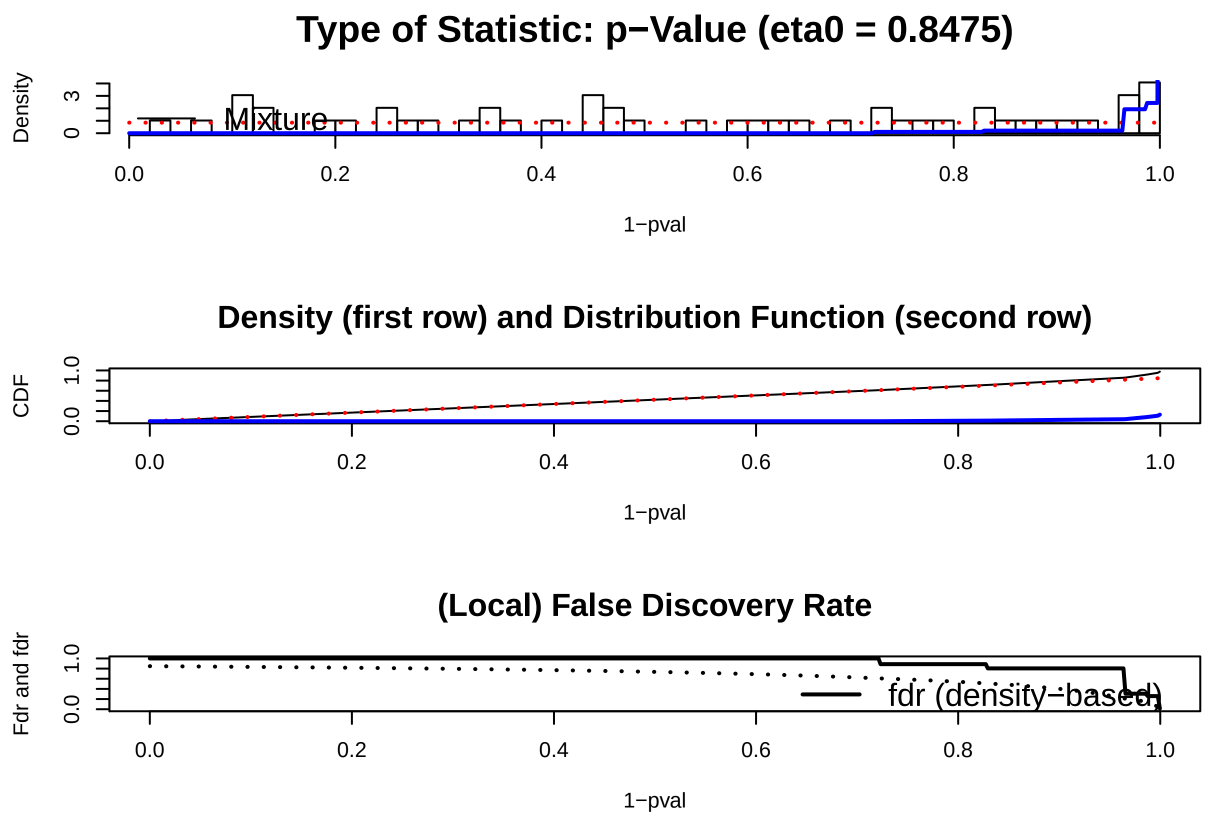
### Linear Regression

NOT EXECUTED.

### FDR

```
library(fdrtool)
fdr.t_pval <- fdrtool(ttest.p[complete.cases(ttest.p)], statistic = "pvalue")
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
## Step 5... prepare for plotting
```




```
data.frame(qval = fdr.t_pval$qval, lfdr = fdr.t_pval$lfdr)[names(signif_p),]
```

```
##             qval       lfdr
## X3.1 0.20544743 0.30536895
## X3.6 0.20843236 0.80390084
## X4.5 0.12419693 0.25835034
## X5.4 0.17953704 0.30536895
## X7.2 0.01310949 0.01310949
## X7.3 0.13572992 0.30536895
## X7   0.01310949 0.25835034
```

## 3/15/17 update

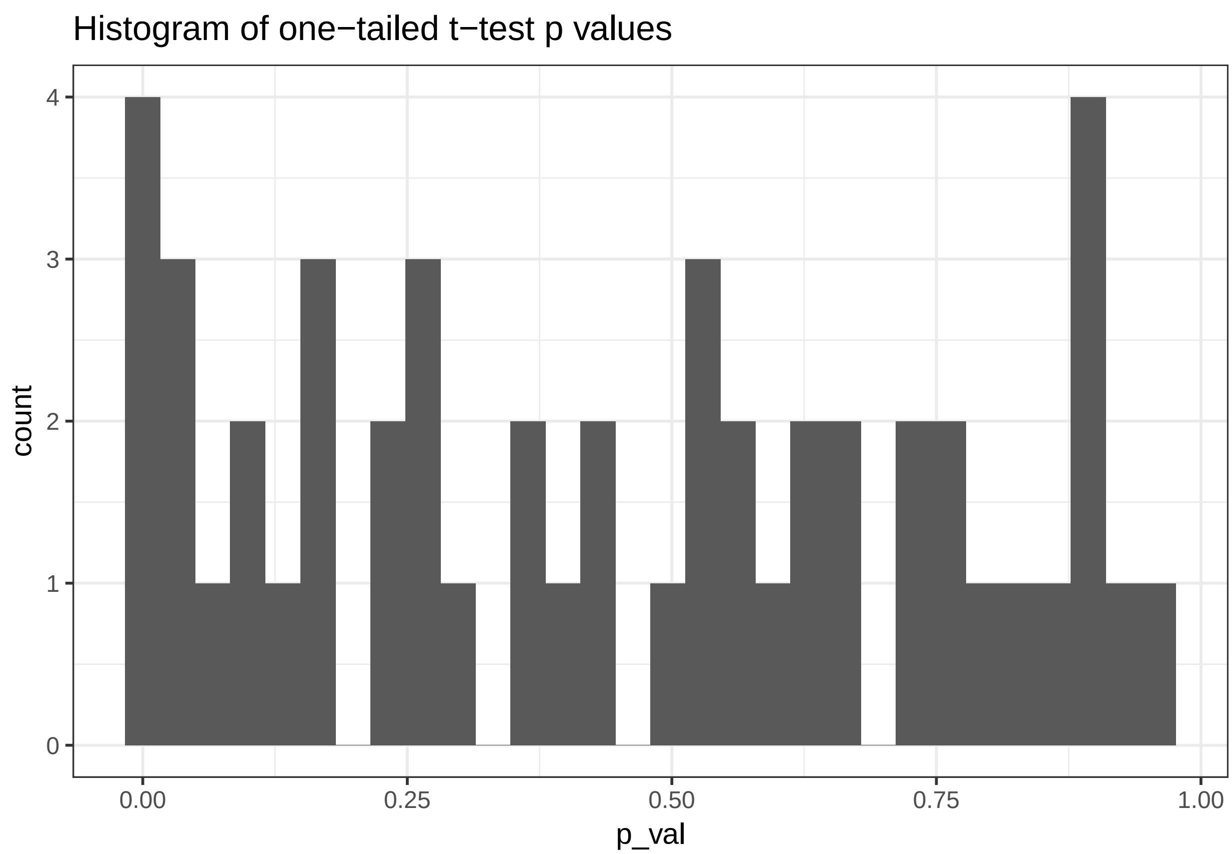
### 1) Table of p-values for each of the 40+ individual behaviors and the groups of behaviors and histogram

```
ttest_pvals <- data.frame(name = names(ttest.p), p_val = ttest.p)
ttest_pvals
```

```
##      name        p_val
## X1.1 X1.1 0.6752092783
## X1.2 X1.2 0.8188472416
## X1.3 X1.3          NaN
## X1.4 X1.4 0.2770147784
## X2.1 X2.1 0.8833367399
```

```
## X2.2 X2.2 0.8906634807
## X2.3 X2.3 0.7551953705
## X2.4 X2.4 0.3987297689
## X2.5 X2.5 0.7499575319
## X3.1 X3.1 0.0335961958
## X3.2 X3.2 0.5324208861
## X3.3 X3.3 0.3784371543
## X3.4 X3.4 0.6486770297
## X3.5 X3.5 0.7195982149
## X3.6 X3.6 0.0351338586
## X3.7 X3.7 0.5422702201
## X4.1 X4.1 0.7867334778
## X4.2 X4.2 0.0880695562
## X4.3 X4.3 0.3059903237
## X4.5 X4.5 0.0109343908
## X4.6 X4.6 0.5238069475
## X4.7 X4.7 0.5912952448
## X4.8 X4.8 0.2155312294
## X5.1 X5.1 0.1590053715
## X5.2 X5.2 0.4454687570
## X5.3 X5.3 0.9239509268
## X5.4 X5.4 0.0233137261
## X5.5 X5.5 0.0769886238
## X5.6 X5.6 0.5580231422
## X6.1 X6.1 0.5071686482
## X6.2 X6.2 0.1706413136
## X6.3 X6.3 0.7336872971
## X7.1 X7.1 0.2628859148
## X7.2 X7.2 0.0003613601
## X7.3 X7.3 0.0130736916
## X7.4 X7.4 0.1600108144
## X8.1 X8.1 0.5494683891
## X8.2 X8.2 0.3550563669
## X8.3 X8.3 0.8862151568
## X8.4 X8.4 0.9601037838
## X8.5 X8.5 0.6414532975
## X1     X1 0.6208418334
## X2     X2 0.8654965583
## X3     X3 0.2376303241
## X4     X4 0.2559034073
## X5     X5 0.1126502793
## X6     X6 0.4172467798
## X7     X7 0.0006313622
## X8     X8 0.8794451004
## Xall Xall 0.1229331990
```

```r
ggplot(ttest_pvals,aes(p_val))+
  geom_histogram()+
  theme_bw()+
  ggtitle("Histogram of one-tailed t-test p values")
```




**3) Table of all means, SDs and SEMs**

**4) Table of all mean differences**

```
library(tibble)
library(dplyr)

sem <- function(x, ...) {
  sd(x, ...)/sqrt(length(na.omit(x)))
}

pre_post.stat <- pre_post.all %>%
  group_by(LogCheck) %>%
  summarise_at(vars(matches("X")), funs(mean, sd, sem), na.rm = TRUE) %>%
  mutate(logc = ifelse(LogCheck == TRUE, "Exp", "Ctl")) %>%
  column_to_rownames(var="logc") %>%
  select(-LogCheck)
```

```
library(data.table)
library(tidyr)

pre_post.stat.rshaped <- as.data.frame(t(pre_post.stat)) %>%
  rownames_to_column("feature")  %>%
  separate(col = feature, into = c("feature", "measure"), sep = "_") %>%
  setDT() %>%
```

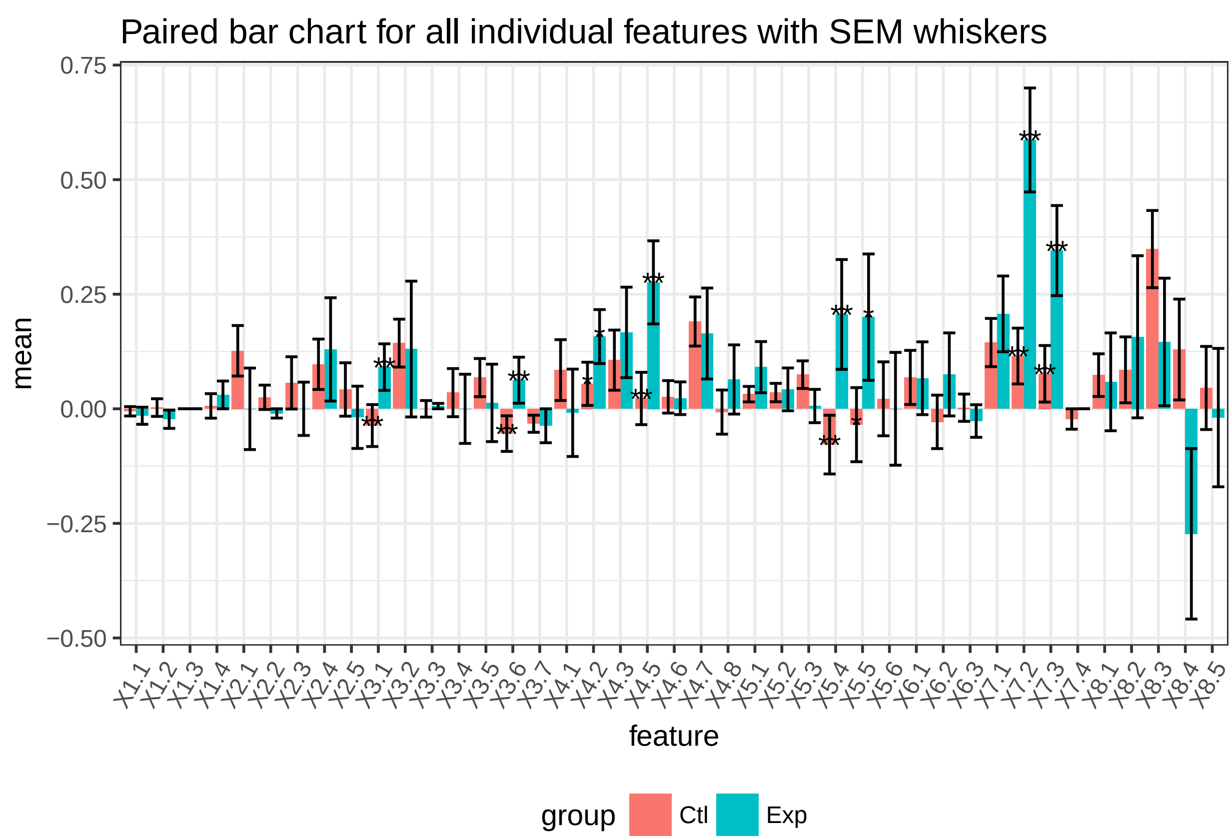
```r
  dcast(feature ~ measure, value.var = c("Exp", "Ctl")) %>%
  mutate(mean_diff = Exp_mean - Ctl_mean)

pre_post.stat.rshaped #Show all means, SDs, SEMs, mean differences in Exp and Ctl group
```

```
##    feature     Exp_mean     Exp_sd     Exp_sem      Ctl_mean     Ctl_sd
## 1       X1 -0.001838235 0.06791904 0.011648019  2.878007e-03 0.09735797
## 2     X1.1 -0.015151515 0.10715413 0.018653140 -5.434783e-03 0.09790569
## 3     X1.2 -0.022727273 0.11462637 0.019953890  2.688172e-03 0.18611672
## 4     X1.3  0.000000000 0.00000000 0.000000000  0.000000e+00 0.00000000
## 5     X1.4  0.030303030 0.17407766 0.030303030  6.315789e-03 0.26003704
## 6       X2  0.021813725 0.20210138 0.034660101  6.921769e-02 0.24416342
## 7     X2.1  0.000000000 0.42640143 0.088910845  1.265823e-01 0.49008342
## 8     X2.2 -0.010101010 0.05802589 0.010101010  2.508961e-02 0.25649326
## 9     X2.3  0.000000000 0.30223526 0.058165203  5.654762e-02 0.52260813
## 10    X2.4  0.129629630 0.58577025 0.112731536  9.722222e-02 0.46639831
## 11    X2.5 -0.018518519 0.35304939 0.067944388  4.216867e-02 0.53078876
## 12      X3  0.025200680 0.15156706 0.025619509  2.065826e-03 0.19774057
## 13    X3.1  0.090909091 0.23836565 0.050819727 -3.658537e-02 0.41410686
## 14    X3.2  0.130434783 0.71059180 0.148168634  1.433333e-01 0.45242092
## 15    X3.3  0.005882353 0.03429972 0.005882353 -2.192557e-18 0.18110770
## 16    X3.4  0.000000000 0.39223227 0.075485136  3.529412e-02 0.48420428
## 17    X3.5  0.012903226 0.47027789 0.084464402  6.808511e-02 0.40380256
## 18    X3.6  0.062500000 0.28398092 0.050201208 -5.412371e-02 0.38226777
## 19    X3.7 -0.037037037 0.19245009 0.037037037 -3.260870e-02 0.17858337
## 20      X4  0.106394558 0.25268253 0.042711144  7.111111e-02 0.32587784
## 21    X4.1 -0.008695652 0.45717815 0.095328233  8.450704e-02 0.55924640
## 22    X4.2  0.157575758 0.33822173 0.058876846  5.454545e-02 0.41375859
## 23    X4.3  0.166666667 0.54096551 0.098766337  1.061728e-01 0.59083536
## 24    X4.5  0.275862069 0.48818554 0.090653779  2.253521e-02 0.48112879
## 25    X4.6  0.023076923 0.18178601 0.035651170  2.608696e-02 0.33964091
## 26    X4.7  0.164285714 0.52508503 0.099231743  1.904762e-01 0.49078718
## 27    X4.8  0.064000000 0.37735925 0.075471849 -7.142857e-03 0.44173418
## 28      X5  0.076380952 0.23878385 0.040361837  2.228333e-02 0.17608022
## 29    X5.1  0.090909091 0.32051096 0.055793796  3.191489e-02 0.16409594
## 30    X5.2  0.042424242 0.26813045 0.046675521  3.541667e-02 0.19680563
## 31    X5.3  0.006060606 0.20907444 0.036395188  7.441860e-02 0.27957364
## 32    X5.4  0.205882353 0.49398961 0.119810079 -7.812500e-02 0.44409917
## 33    X5.5  0.200000000 0.55136195 0.137840488 -3.461538e-02 0.58305639
## 34    X5.6  0.000000000 0.42640143 0.123091491  2.173913e-02 0.54728138
## 35      X6  0.029947917 0.32699525 0.057805140  1.570175e-02 0.34658897
## 36    X6.1  0.066666667 0.43516413 0.079449736  6.845238e-02 0.54101532
## 37    X6.2  0.075000000 0.49632268 0.090615710 -2.848101e-02 0.51885497
## 38    X6.3 -0.026666667 0.19464084 0.035536393  2.409639e-03 0.27139023
## 39      X7  0.245182292 0.26284030 0.046464040  5.926418e-02 0.28162180
## 40    X7.1  0.207142857 0.43710205 0.082604523  1.445783e-01 0.47913871
## 41    X7.2  0.586538462 0.57854260 0.113461538  1.151316e-01 0.53141445
## 42    X7.3  0.345238095 0.45052879 0.098313442  7.638889e-02 0.52346168
## 43    X7.4  0.000000000 0.00000000 0.000000000 -2.222222e-02 0.21081851
## 44      X8  0.002352941 0.33691207 0.081713179  1.132823e-01 0.30242712
## 45    X8.1  0.058823529 0.44025394 0.106777264  7.346939e-02 0.32581661
## 46    X8.2  0.157142857 0.66183229 0.176882119  8.500000e-02 0.45491617
## 47    X8.3  0.145833333 0.48216290 0.139188440  3.484848e-01 0.48387835
## 48    X8.4 -0.272727273 0.61699418 0.186030745  1.293103e-01 0.59244006
```

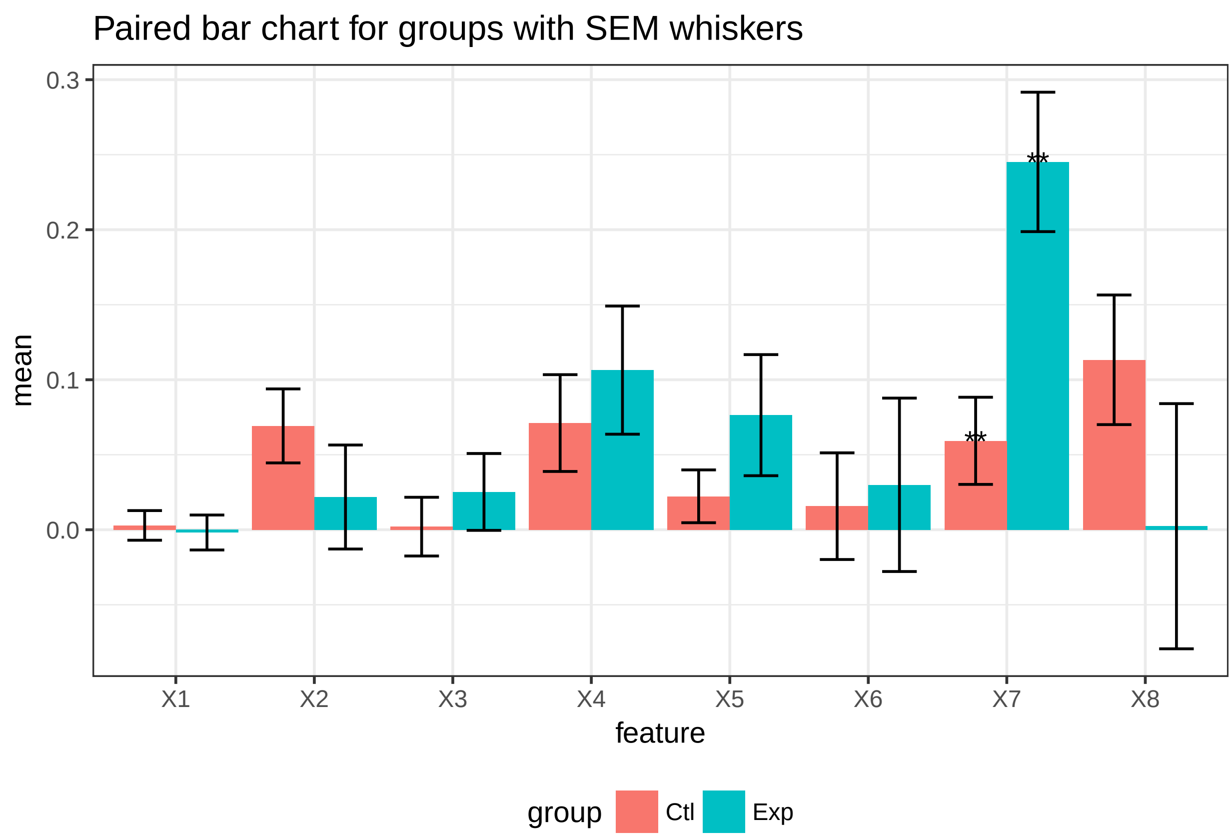
```
## 49    X8.5 -0.019230769 0.54449459 0.151015628  4.545455e-02 0.52087121
## 50    Xall  0.069722007 0.11902739 0.020119300  4.147969e-02 0.13437991
##         Ctl_sem    mean_diff
## 1  0.009885204 -0.004716242
## 2  0.010207373 -0.009716733
## 3  0.019299405 -0.025415445
## 4  0.000000000  0.000000000
## 5  0.026679237  0.023987241
## 6  0.024664230 -0.047403962
## 7  0.055138693 -0.126582278
## 8  0.026597112 -0.035190616
## 9  0.057021222 -0.056547619
## 10 0.054965568  0.032407407
## 11 0.058261635 -0.060687193
## 12 0.019579234  0.023134854
## 13 0.045730453  0.127494457
## 14 0.052241067 -0.012898551
## 15 0.018020890  0.005882353
## 16 0.052519328 -0.035294118
## 17 0.041649054 -0.055181881
## 18 0.038813412  0.116623711
## 19 0.018618604 -0.004428341
## 20 0.032266714  0.035283447
## 21 0.066370337 -0.093202694
## 22 0.047152168  0.103030303
## 23 0.065648373  0.060493827
## 24 0.057099482  0.253326858
## 25 0.035410013 -0.003010033
## 26 0.053549272 -0.026190476
## 27 0.048197151  0.071142857
## 28 0.017608022  0.054097619
## 29 0.016925204  0.058994197
## 30 0.020086391  0.007007576
## 31 0.030147201 -0.068357999
## 32 0.064100193  0.284007353
## 33 0.080855373  0.234615385
## 34 0.080692237 -0.021739130
## 35 0.035559278  0.014246162
## 36 0.059029611 -0.001785714
## 37 0.058375745  0.103481013
## 38 0.029788948 -0.029076305
## 39 0.029047071  0.185918107
## 40 0.052592306  0.062564544
## 41 0.060957418  0.471406883
## 42 0.061690550  0.268849206
## 43 0.022222222  0.022222222
## 44 0.043203875 -0.110929372
## 45 0.046545230 -0.014645858
## 46 0.071928561  0.072142857
## 47 0.084232409 -0.202651515
## 48 0.110013357 -0.402037618
## 49 0.090672039 -0.064685315
## 50 0.013305594  0.028242319
```

**2) Paired bar chart with SEM whiskers or significant and marginally significant results – a separate one for groups, and markings for significance and high significance. (Bar chart should have means.)**

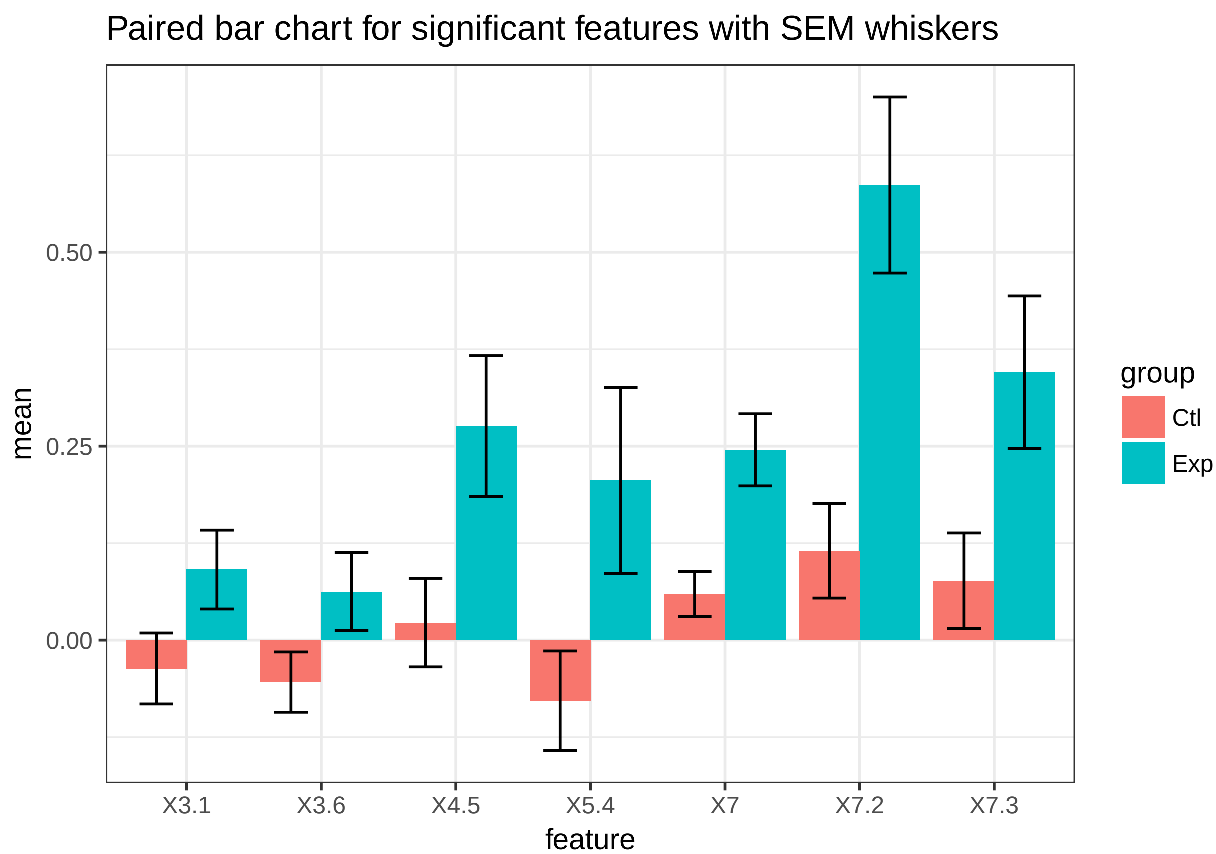
```
library(ggplot2)

pre_post.stat.mean <- pre_post.stat.rshaped %>%
  select(feature, Exp_mean, Ctl_mean, Exp_sem, Ctl_sem) %>%
  gather(Exp_mean, Ctl_mean, key = "group", value = "mean") %>%
  mutate(sem = ifelse(group == "Exp_mean", Exp_sem, Ctl_sem),
         group = substr(group, 1, 3)) %>%
  select(-Exp_sem, -Ctl_sem) %>%
  merge(data.frame(feature = names(ttest.p), p_val = ttest.p),by='feature', all.x = TRUE) %>%
  mutate(sig_level = ifelse(p_val > .1, "", ifelse(p_val > .05, "*", "**")))

# Barchart for individuals (*:p<0.1 **:p<0.05)
pre_post.stat.mean %>%
  filter(grepl("^X\\d\\.\\d$", feature)) %>%
  ggplot(aes(x = feature, y = mean, fill = group)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = mean - sem, ymax = mean + sem), position = position_dodge(0.9)) +
  geom_text(aes(label=sig_level), position = position_dodge(0.9)) +
  theme_bw()+theme(axis.text.x = element_text(angle = 60, hjust = 1))+theme(legend.position="bottom") +
  ggtitle("Paired bar chart for all individual features with SEM whiskers")
```



```r
# Barchart for groups (*:p<0.1 **:p<0.05)
pre_post.stat.mean %>%
  filter(grepl("^X\\d$", feature)) %>%
  ggplot(aes(x = feature, y = mean, fill = group)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = mean - sem, ymax = mean + sem),
                position = position_dodge(0.9),
                width =.5) +
  geom_text(aes(label=sig_level),position = position_dodge(0.9))+
  theme_bw()+theme(legend.position="bottom") +
  ggtitle("Paired bar chart for groups with SEM whiskers")
```




```r
# Barchart for individuals and groups with p < 0.05
pre_post.stat.mean %>%
  filter(feature %in% c("X3.1", "X3.6", "X4.5", "X5.4", "X7.2", "X7.3", "X7")) %>%
  ggplot(aes(x = feature, y = mean, fill = group)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = mean - sem, ymax = mean + sem),
                position = position_dodge(0.9),
                width = .5) +
  theme_bw()+
  ggtitle("Paired bar chart for significant features with SEM whiskers")
```




**5) Table of frequencies for the responses in the PRE dataset only? It can be from the CORRECTED PRE data file and have daily, weekly, etc. instead of the numerical values. Also, with the PRE table of frequencies, flag the Experimental participants in the table?**

1. For all observations

```
freq <- function(x) table(x) / length(x)
PRE_cleaned <- PRE %>%
  select(starts_with("X")) %>%
  mutate_all(function(x) ifelse(x %in% c("Daily", "Weekly", "Monthly", "Seasonally", "Annually", "Never

do.call(gtools::smartbind, lapply(PRE_cleaned, freq)) %>%
  select(Daily, Weekly, Monthly, Seasonally, Annually, Never, `NA`)
```

```
##           Daily      Weekly     Monthly  Seasonally    Annually
## X1.1 0.74271845 0.116504854 0.048543689          NA          NA
## X1.2 0.67475728 0.106796117 0.121359223          NA          NA
## X1.3 0.95631068          NA          NA          NA          NA
## X1.4 0.87864078          NA          NA 0.009708738          NA
## X2.1 0.43203883 0.111650485 0.097087379 0.014563107 0.009708738
## X2.2 0.63592233 0.121359223 0.082524272 0.038834951          NA
## X2.3 0.26213592 0.140776699 0.165048544 0.131067961 0.024271845
## X2.4 0.05825243 0.063106796 0.033980583 0.077669903 0.004854369
## X2.5 0.08252427 0.101941748 0.116504854 0.325242718 0.067961165
## X3.1 0.39805825 0.165048544 0.033980583 0.116504854          NA
## X3.2 0.13592233 0.048543689 0.024271845 0.063106796          NA
```

```
## X3.3 0.94660194 0.033980583          NA          NA          NA
## X3.4 0.13106796 0.004854369 0.009708738 0.004854369          NA
## X3.5 0.75728155 0.033980583 0.009708738 0.024271845 0.004854369
## X3.6 0.38834951 0.150485437 0.131067961 0.218446602 0.024271845
## X3.7 0.39805825 0.140776699 0.092233010 0.213592233 0.029126214
## X4.1 0.17475728 0.024271845 0.019417476 0.082524272 0.014563107
## X4.2 0.60679612 0.058252427 0.063106796 0.058252427 0.009708738
## X4.3 0.48543689 0.024271845 0.019417476 0.033980583          NA
## X4.5 0.38349515 0.048543689 0.048543689 0.121359223 0.009708738
## X4.6 0.60679612 0.077669903 0.150485437 0.077669903          NA
## X4.7 0.45631068 0.033980583 0.063106796 0.043689320 0.004854369
## X4.8 0.37864078 0.101941748 0.116504854 0.189320388 0.009708738
## X5.1 0.83495146 0.019417476 0.019417476          NA          NA
## X5.2 0.86893204 0.009708738 0.009708738          NA          NA
## X5.3 0.74757282 0.024271845 0.014563107 0.043689320 0.004854369
## X5.4 0.34951456 0.048543689 0.048543689 0.048543689 0.004854369
## X5.5 0.29611650 0.029126214 0.009708738 0.043689320          NA
## X5.6 0.08737864 0.053398058 0.111650485 0.179611650 0.014563107
## X6.1 0.57281553 0.077669903 0.014563107 0.004854369 0.004854369
## X6.2 0.53398058 0.077669903 0.024271845 0.014563107          NA
## X6.3 0.77184466 0.009708738 0.004854369 0.053398058 0.004854369
## X7.1 0.48543689 0.058252427 0.033980583 0.014563107          NA
## X7.2 0.15048544 0.024271845 0.043689320 0.019417476 0.004854369
## X7.3 0.09223301 0.033980583 0.029126214 0.097087379 0.024271845
## X7.4 0.01456311 0.019417476 0.169902913 0.388349515 0.266990291
## X8.1 0.46601942 0.082524272 0.038834951 0.019417476 0.004854369
## X8.2 0.11165049 0.019417476 0.009708738 0.004854369          NA
## X8.3 0.07281553 0.033980583 0.038834951 0.009708738 0.004854369
## X8.4 0.06796117 0.024271845 0.009708738 0.009708738 0.004854369
## X8.5 0.07766990 0.043689320 0.019417476 0.063106796 0.009708738
##            Never          NA
## X1.1 0.009708738 0.082524272
## X1.2 0.024271845 0.072815534
## X1.3 0.009708738 0.033980583
## X1.4 0.058252427 0.053398058
## X2.1 0.179611650 0.155339806
## X2.2 0.029126214 0.092233010
## X2.3 0.160194175 0.116504854
## X2.4 0.611650485 0.150485437
## X2.5 0.155339806 0.150485437
## X3.1 0.121359223 0.165048544
## X3.2 0.563106796 0.165048544
## X3.3 0.009708738 0.009708738
## X3.4 0.776699029 0.072815534
## X3.5 0.126213592 0.043689320
## X3.6 0.048543689 0.038834951
## X3.7 0.043689320 0.082524272
## X4.1 0.480582524 0.203883495
## X4.2 0.101941748 0.101941748
## X4.3 0.330097087 0.106796117
## X4.5 0.218446602 0.169902913
## X4.6 0.014563107 0.072815534
## X4.7 0.310679612 0.087378641
## X4.8 0.092233010 0.111650485
```

```
## X5.1 0.038834951 0.087378641
## X5.2 0.033980583 0.077669903
## X5.3 0.043689320 0.121359223
## X5.4 0.121359223 0.378640777
## X5.5 0.237864078 0.383495146
## X5.6 0.097087379 0.456310680
## X6.1 0.184466019 0.140776699
## X6.2 0.184466019 0.165048544
## X6.3 0.024271845 0.131067961
## X7.1 0.237864078 0.169902913
## X7.2 0.543689320 0.213592233
## X7.3 0.490291262 0.233009709
## X7.4 0.029126214 0.111650485
## X8.1 0.067961165 0.320388350
## X8.2 0.485436893 0.368932039
## X8.3 0.368932039 0.470873786
## X8.4 0.359223301 0.524271845
## X8.5 0.320388350 0.466019417
```

2. Experimental group

```
PRE_cleaned_E <- PRE %>%
  filter(BPL.BLD.ID %in% logcheck.usage$BPL.BLD.ID) %>%
  select(starts_with("X")) %>%
  mutate_all(function(x) ifelse(x %in% c("Daily", "Weekly", "Monthly", "Seasonally", "Annually", "Never
do.call(gtools::smartbind, lapply(PRE_cleaned_E, freq)) %>%
  select(Daily, Weekly, Monthly, Seasonally, Annually, Never, `NA`)
```

```
##            Daily     Weekly    Monthly Seasonally   Annually      Never
## X1.1 0.77142857 0.14285714 0.02857143         NA         NA         NA
## X1.2 0.74285714 0.11428571 0.08571429         NA         NA         NA
## X1.3 0.97142857         NA         NA         NA         NA         NA
## X1.4 0.91428571         NA         NA         NA         NA 0.02857143
## X2.1 0.51428571 0.08571429 0.11428571         NA         NA 0.11428571
## X2.2 0.80000000 0.14285714 0.02857143         NA         NA         NA
## X2.3 0.20000000 0.22857143 0.22857143 0.14285714         NA 0.08571429
## X2.4 0.05714286 0.11428571 0.05714286 0.05714286         NA 0.57142857
## X2.5 0.11428571 0.14285714 0.20000000 0.31428571 0.02857143 0.08571429
## X3.1 0.45714286 0.11428571 0.05714286 0.02857143         NA 0.05714286
## X3.2 0.22857143 0.05714286         NA 0.05714286         NA 0.40000000
## X3.3 0.94285714 0.02857143         NA         NA         NA         NA
## X3.4 0.14285714         NA         NA         NA         NA 0.74285714
## X3.5 0.74285714 0.02857143         NA         NA 0.02857143 0.08571429
## X3.6 0.42857143 0.14285714 0.17142857 0.20000000         NA 0.05714286
## X3.7 0.40000000 0.14285714 0.05714286 0.22857143         NA 0.08571429
## X4.1 0.25714286         NA 0.05714286 0.08571429         NA 0.40000000
## X4.2 0.65714286 0.02857143 0.08571429 0.08571429         NA 0.14285714
## X4.3 0.51428571 0.02857143         NA 0.08571429         NA 0.28571429
## X4.5 0.31428571 0.02857143 0.08571429 0.22857143         NA 0.25714286
## X4.6 0.60000000 0.05714286 0.17142857         NA         NA         NA
## X4.7 0.54285714         NA 0.05714286 0.02857143         NA 0.25714286
## X4.8 0.34285714 0.11428571 0.11428571 0.22857143         NA 0.02857143
## X5.1 0.80000000 0.02857143 0.02857143         NA         NA 0.08571429
## X5.2 0.88571429         NA         NA         NA         NA 0.05714286
## X5.3 0.91428571         NA         NA         NA         NA 0.02857143
```

```
## X5.4 0.31428571 0.02857143 0.05714286 0.08571429         NA 0.20000000
## X5.5 0.28571429 0.02857143         NA 0.02857143         NA 0.22857143
## X5.6 0.08571429 0.02857143 0.17142857 0.17142857         NA 0.02857143
## X6.1 0.65714286 0.05714286 0.02857143         NA 0.02857143 0.14285714
## X6.2 0.60000000 0.14285714         NA         NA         NA 0.17142857
## X6.3 0.85714286         NA         NA 0.02857143         NA         NA
## X7.1 0.51428571 0.02857143         NA 0.02857143         NA 0.31428571
## X7.2 0.05714286 0.02857143         NA 0.02857143         NA 0.65714286
## X7.3 0.11428571         NA 0.02857143 0.08571429 0.02857143 0.51428571
## X7.4 0.02857143 0.02857143 0.14285714 0.51428571 0.17142857         NA
## X8.1 0.42857143 0.05714286 0.02857143         NA         NA 0.08571429
## X8.2 0.08571429 0.02857143 0.02857143         NA         NA 0.40000000
## X8.3 0.02857143 0.02857143 0.02857143         NA         NA 0.45714286
## X8.4 0.05714286 0.05714286 0.02857143 0.02857143         NA 0.31428571
## X8.5 0.11428571 0.02857143         NA 0.05714286         NA 0.31428571
##              NA
## X1.1 0.05714286
## X1.2 0.05714286
## X1.3 0.02857143
## X1.4 0.05714286
## X2.1 0.17142857
## X2.2 0.02857143
## X2.3 0.11428571
## X2.4 0.14285714
## X2.5 0.11428571
## X3.1 0.28571429
## X3.2 0.25714286
## X3.3 0.02857143
## X3.4 0.11428571
## X3.5 0.11428571
## X3.6         NA
## X3.7 0.08571429
## X4.1 0.20000000
## X4.2         NA
## X4.3 0.08571429
## X4.5 0.08571429
## X4.6 0.17142857
## X4.7 0.11428571
## X4.8 0.17142857
## X5.1 0.05714286
## X5.2 0.05714286
## X5.3 0.05714286
## X5.4 0.31428571
## X5.5 0.42857143
## X5.6 0.51428571
## X6.1 0.08571429
## X6.2 0.08571429
## X6.3 0.11428571
## X7.1 0.11428571
## X7.2 0.22857143
## X7.3 0.22857143
## X7.4 0.11428571
## X8.1 0.40000000
## X8.2 0.45714286
```

```
## X8.3 0.45714286
## X8.4 0.51428571
## X8.5 0.48571429
```

3. Control group

```
PRE_cleaned_C <- PRE %>%
  filter(!BPL.BLD.ID %in% logcheck.usage$BPL.BLD.ID) %>%
  select(starts_with("X")) %>%
  mutate_all(function(x) ifelse(x %in% c("Daily", "Weekly", "Monthly", "Seasonally", "Annually", "Never

do.call(gtools::smartbind, lapply(PRE_cleaned_C, freq)) %>%
  select(Daily, Weekly, Monthly, Seasonally, Annually, Never, `NA`)
```

```
##            Daily      Weekly     Monthly  Seasonally    Annually      Never
## X1.1 0.73684211 0.111111111 0.052631579          NA          NA 0.01169591
## X1.2 0.66081871 0.105263158 0.128654971          NA          NA 0.02923977
## X1.3 0.95321637          NA          NA          NA          NA 0.01169591
## X1.4 0.87134503          NA          NA 0.011695906          NA 0.06432749
## X2.1 0.41520468 0.116959064 0.093567251 0.017543860 0.011695906 0.19298246
## X2.2 0.60233918 0.116959064 0.093567251 0.046783626          NA 0.03508772
## X2.3 0.27485380 0.122807018 0.152046784 0.128654971 0.029239766 0.17543860
## X2.4 0.05847953 0.052631579 0.029239766 0.081871345 0.005847953 0.61988304
## X2.5 0.07602339 0.093567251 0.099415205 0.327485380 0.076023392 0.16959064
## X3.1 0.38596491 0.175438596 0.029239766 0.134502924          NA 0.13450292
## X3.2 0.11695906 0.046783626 0.029239766 0.064327485          NA 0.59649123
## X3.3 0.94736842 0.035087719          NA          NA          NA 0.01169591
## X3.4 0.12865497 0.005847953 0.011695906 0.005847953          NA 0.78362573
## X3.5 0.76023392 0.035087719 0.011695906 0.029239766          NA 0.13450292
## X3.6 0.38011696 0.152046784 0.122807018 0.222222222 0.029239766 0.04678363
## X3.7 0.39766082 0.140350877 0.099415205 0.210526316 0.035087719 0.03508772
## X4.1 0.15789474 0.029239766 0.011695906 0.081871345 0.017543860 0.49707602
## X4.2 0.59649123 0.064327485 0.058479532 0.052631579 0.011695906 0.09356725
## X4.3 0.47953216 0.023391813 0.023391813 0.023391813          NA 0.33918129
## X4.5 0.39766082 0.052631579 0.040935673 0.099415205 0.011695906 0.21052632
## X4.6 0.60818713 0.081871345 0.146198830 0.093567251          NA 0.01754386
## X4.7 0.43859649 0.040935673 0.064327485 0.046783626 0.005847953 0.32163743
## X4.8 0.38596491 0.099415205 0.116959064 0.181286550 0.011695906 0.10526316
## X5.1 0.84210526 0.017543860 0.017543860          NA          NA 0.02923977
## X5.2 0.86549708 0.011695906 0.011695906          NA          NA 0.02923977
## X5.3 0.71345029 0.029239766 0.017543860 0.052631579 0.005847953 0.04678363
## X5.4 0.35672515 0.052631579 0.046783626 0.040935673 0.005847953 0.10526316
## X5.5 0.29824561 0.029239766 0.011695906 0.046783626          NA 0.23976608
## X5.6 0.08771930 0.058479532 0.099415205 0.181286550 0.017543860 0.11111111
## X6.1 0.55555556 0.081871345 0.011695906 0.005847953          NA 0.19298246
## X6.2 0.52046784 0.064327485 0.029239766 0.017543860          NA 0.18713450
## X6.3 0.75438596 0.011695906 0.005847953 0.058479532 0.005847953 0.02923977
## X7.1 0.47953216 0.064327485 0.040935673 0.011695906          NA 0.22222222
## X7.2 0.16959064 0.023391813 0.052631579 0.017543860 0.005847953 0.52046784
## X7.3 0.08771930 0.040935673 0.029239766 0.099415205 0.023391813 0.48538012
## X7.4 0.01169591 0.017543860 0.175438596 0.362573099 0.286549708 0.03508772
## X8.1 0.47368421 0.087719298 0.040935673 0.023391813 0.005847953 0.06432749
## X8.2 0.11695906 0.017543860 0.005847953 0.005847953          NA 0.50292398
## X8.3 0.08187135 0.035087719 0.040935673 0.011695906 0.005847953 0.35087719
## X8.4 0.07017544 0.017543860 0.005847953 0.005847953 0.005847953 0.36842105
```

```
## X8.5 0.07017544 0.046783626 0.023391813 0.064327485 0.011695906 0.32163743
##                NA
## X1.1 0.087719298
## X1.2 0.076023392
## X1.3 0.035087719
## X1.4 0.052631579
## X2.1 0.152046784
## X2.2 0.105263158
## X2.3 0.116959064
## X2.4 0.152046784
## X2.5 0.157894737
## X3.1 0.140350877
## X3.2 0.146198830
## X3.3 0.005847953
## X3.4 0.064327485
## X3.5 0.029239766
## X3.6 0.046783626
## X3.7 0.081871345
## X4.1 0.204678363
## X4.2 0.122807018
## X4.3 0.111111111
## X4.5 0.187134503
## X4.6 0.052631579
## X4.7 0.081871345
## X4.8 0.099415205
## X5.1 0.093567251
## X5.2 0.081871345
## X5.3 0.134502924
## X5.4 0.391812865
## X5.5 0.374269006
## X5.6 0.444444444
## X6.1 0.152046784
## X6.2 0.181286550
## X6.3 0.134502924
## X7.1 0.181286550
## X7.2 0.210526316
## X7.3 0.233918129
## X7.4 0.111111111
## X8.1 0.304093567
## X8.2 0.350877193
## X8.3 0.473684211
## X8.4 0.526315789
## X8.5 0.461988304
```