2-exploration_analyses.R

oomeir

Mon Dec 11 20:33:20 2017

```
library(ggplot2)
library(ggthemes)
suppressPackageStartupMessages(library(dplyr))
```

```
## Warning: package 'dplyr' was built under R version 3.4.1
```

```
library(stringr)
library(reshape2)
library(knitr)
library(xtable)
suppressWarnings(library(lubridate))
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
dat <- read.csv("data/data-ready.text", sep = "|", check.names = F)
```

Split quant. data into groups for time, satisfaction, and preference

```
dat.time <- dat[c("participant-number", "house-type", "t1", "t2", "t3", "t4", "t5")]
names(dat.time) <- c("p", "h", "shelf", "list", "fb", "m_return", "fb_return")
dat.satisfaction <- dat[c("participant-number", "t1q", "t2q",
                          "t3q", "t4q", "t5q", "house-type")]
names(dat.satisfaction) <- c("p", "shelf", "list", "fb", "m_return", "fb_return", "h")
dat.preference <- dat[c("participant-number", "q1", "q2", "q3", "q4", "house-type")]
names(dat.preference) <- c("p", "machine_usefulness",
                           "fb_usefulness", "shelf_vs_list", "app_vs_fb", "h")
dat.error <- dat[c("participant-number", "house-type")]
```

Create data.frame for Errors from observed data. t1e is task 1 errors, t2e is task 2 errors.

```
error <- data.frame(c(8245, 2589, 466, 5115, 9893, 1557, 46, 5511,
                      2757, 7994, 2105, 9500, 3682, 7019, 5665, 2747),
                    c(0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2),
                    c(2, 2, 2, 0, 3, 0, 2, 0, 2, 1, 2, 2, 2, 2, 0, 0))
names(error) <- c("p", "t1e", "t2e")
names(dat.error) <- c("p", "h")
dat.er <- merge(dat.error, error)

# Plots for most important measures
means.time <- melt(dat.time[2:5] %>%
  group_by(h) %>%
  summarise_all(mean))
```

```
## Using h as id variables
```

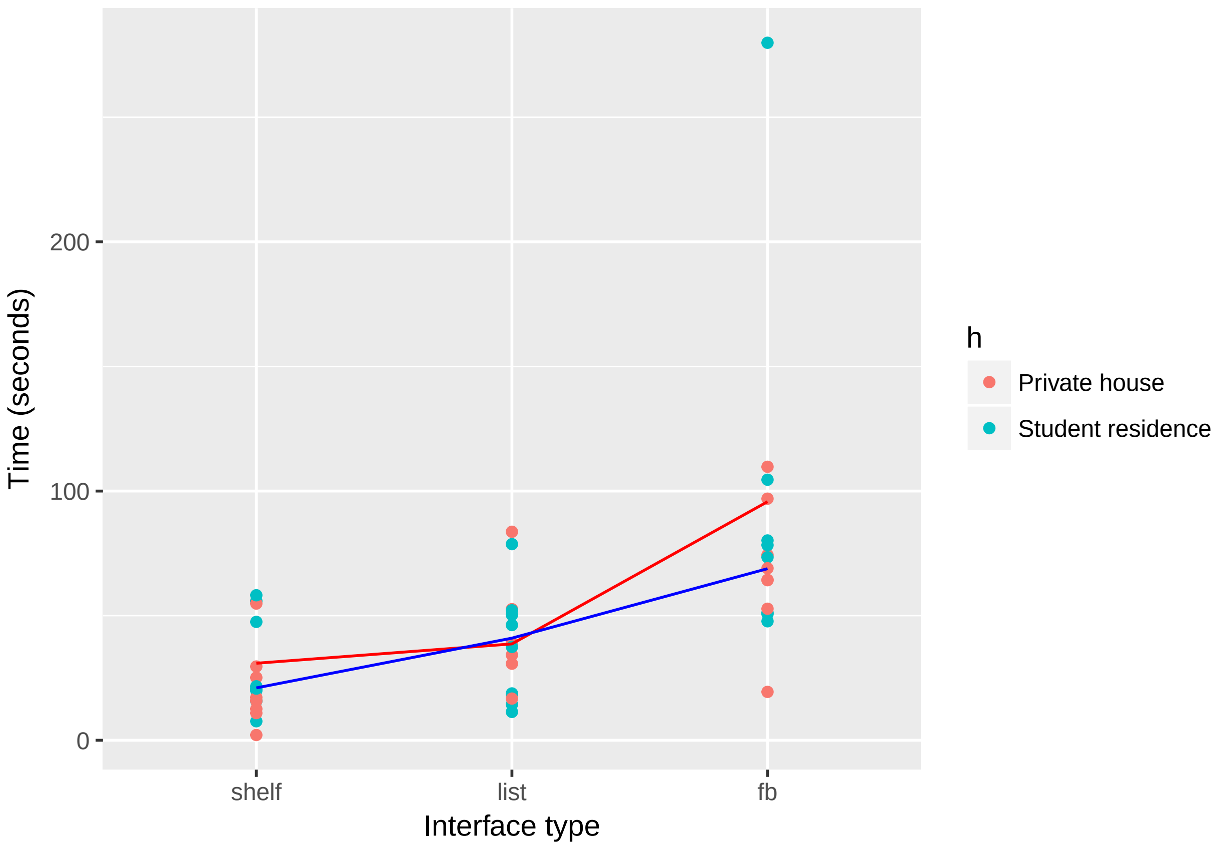
```
means.time.sr <- subset(means.time, h == "Student residence")
means.time.pr <- subset(means.time, h == "Private house")
```

Time for borrow tasks with different interface type

```
ggplot(melt(dat.time[2:5]),aes(x=variable,y=value, color=h)) +
  geom_point() +
  stat_summary(data = means.time.sr, aes(y = value), colour="red", geom="line", group=1)+
  stat_summary(data = means.time.pr, aes(y = value), colour="blue", geom="line", group=2)+
  ylab("Time (seconds)") + xlab("Interface type")
```

```
## Using h as id variables
```

```
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
```



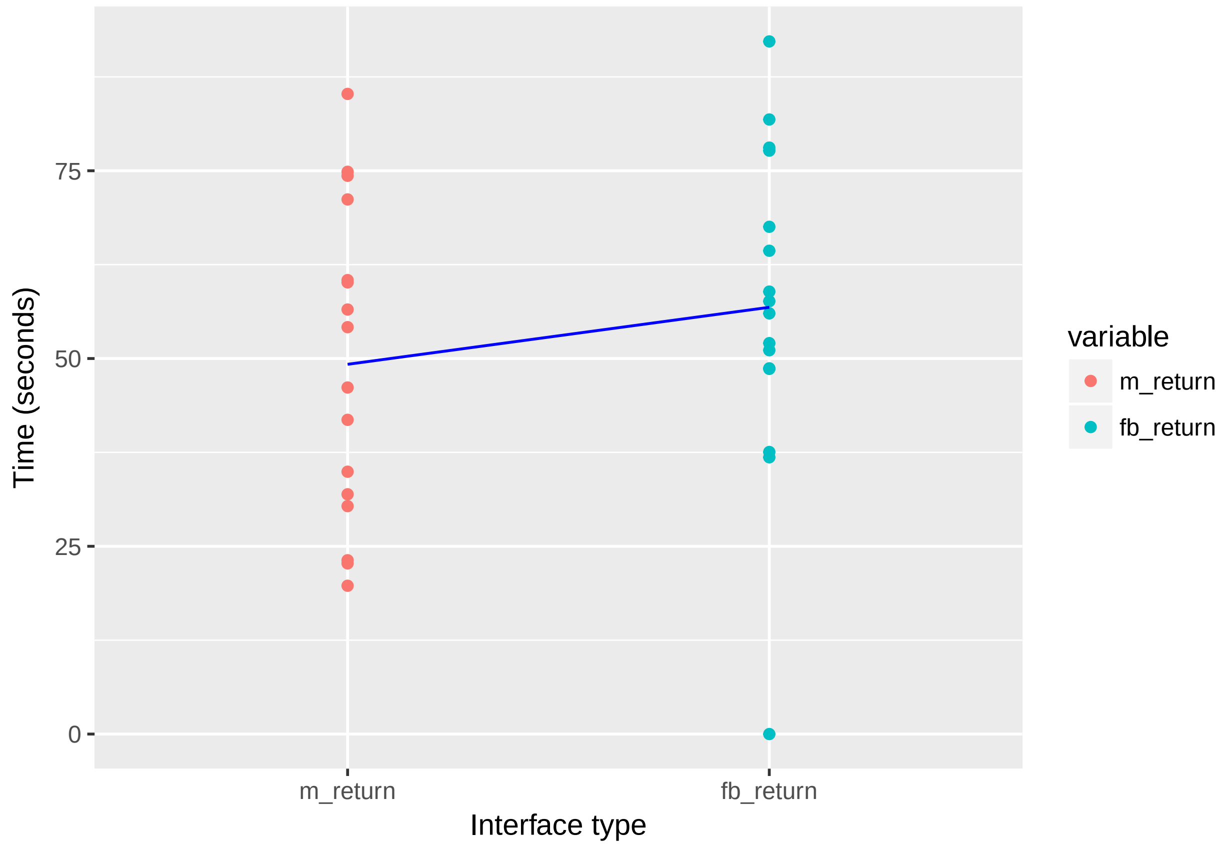

```
ggsave("time-borrow.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
```

Time for return tasks

```
ggplot(melt(dat.time[6:7]),aes(x=variable,y=value, color=variable)) +
  geom_point() +
  stat_summary(aes(y = value), fun.y = mean, colour="blue", geom="line", group=1)+
  ylab("Time (seconds)") + xlab("Interface type")
```

```
## No id variables; using all as measure variables
```



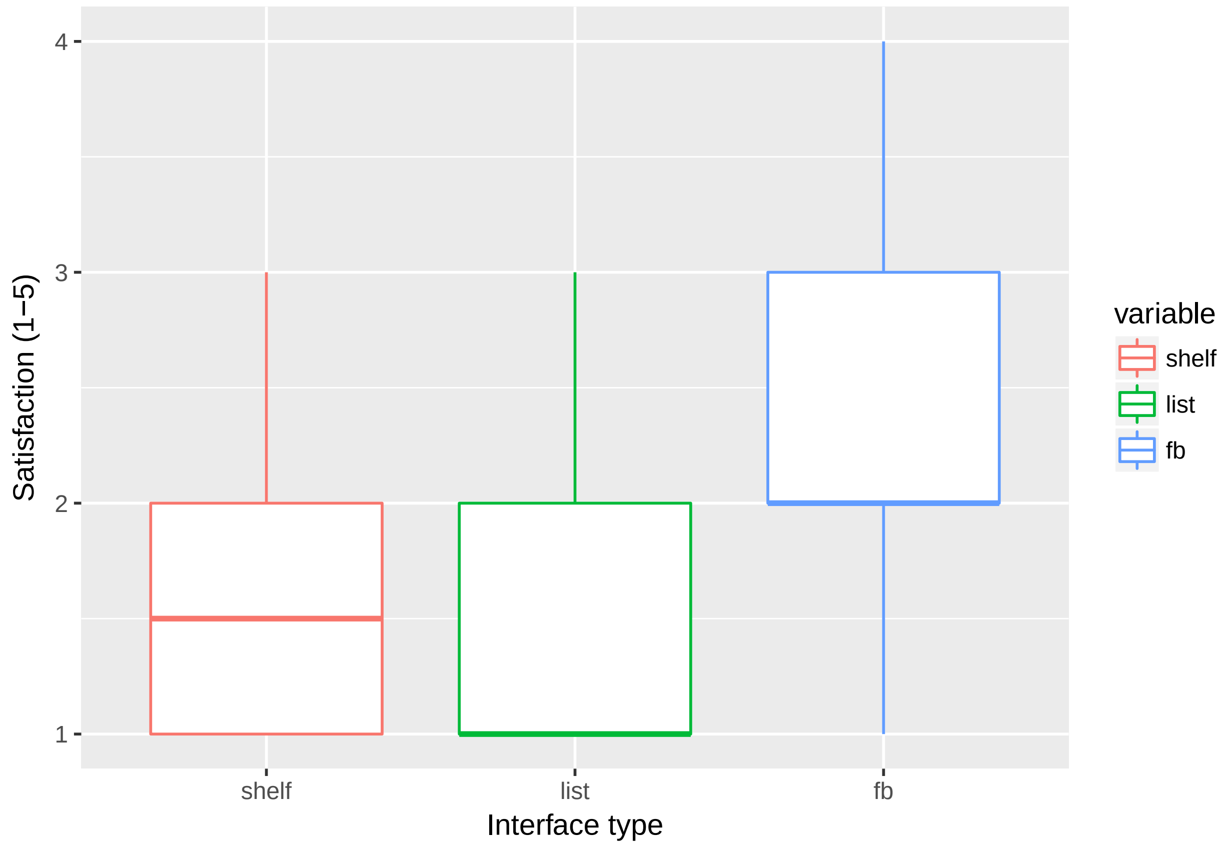

```r
ggsave("time-return.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

Satisfaction for borrow tasks

```r
ggplot(melt(dat.satisfaction[2:4]),aes(x=variable,y=value, color=variable)) +
  geom_boxplot() +
  ylab("Satisfaction (1-5)") + xlab("Interface type")
```

```
## No id variables; using all as measure variables
```



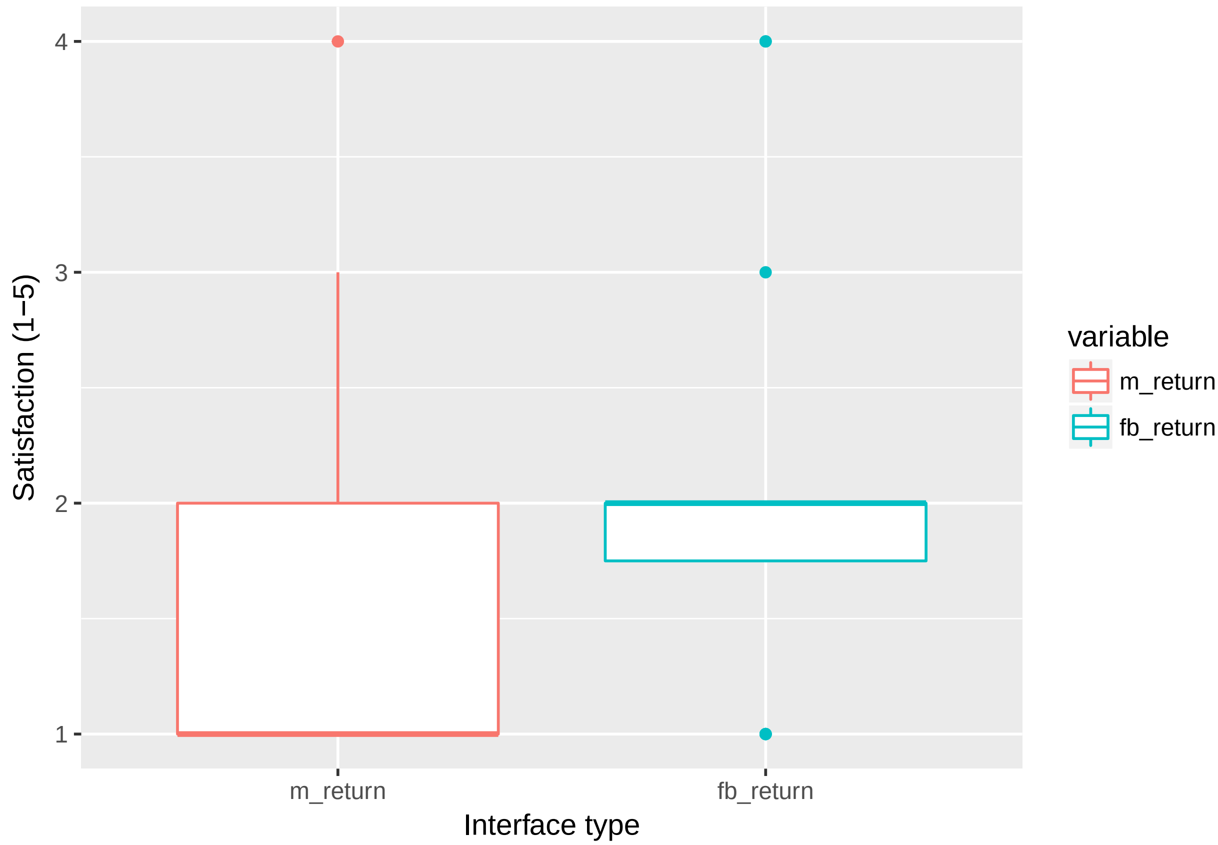

```
ggsave("sat-borrow.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

Satisfaction for return tasks (Facebook task depended heavily on how fast people type)

```
ggplot(melt(dat.satisfaction[5:6]),aes(x=variable,y=value, color=variable)) +
  geom_boxplot() +
  ylab("Satisfaction (1-5)") + xlab("Interface type")
```

```
## No id variables; using all as measure variables
```



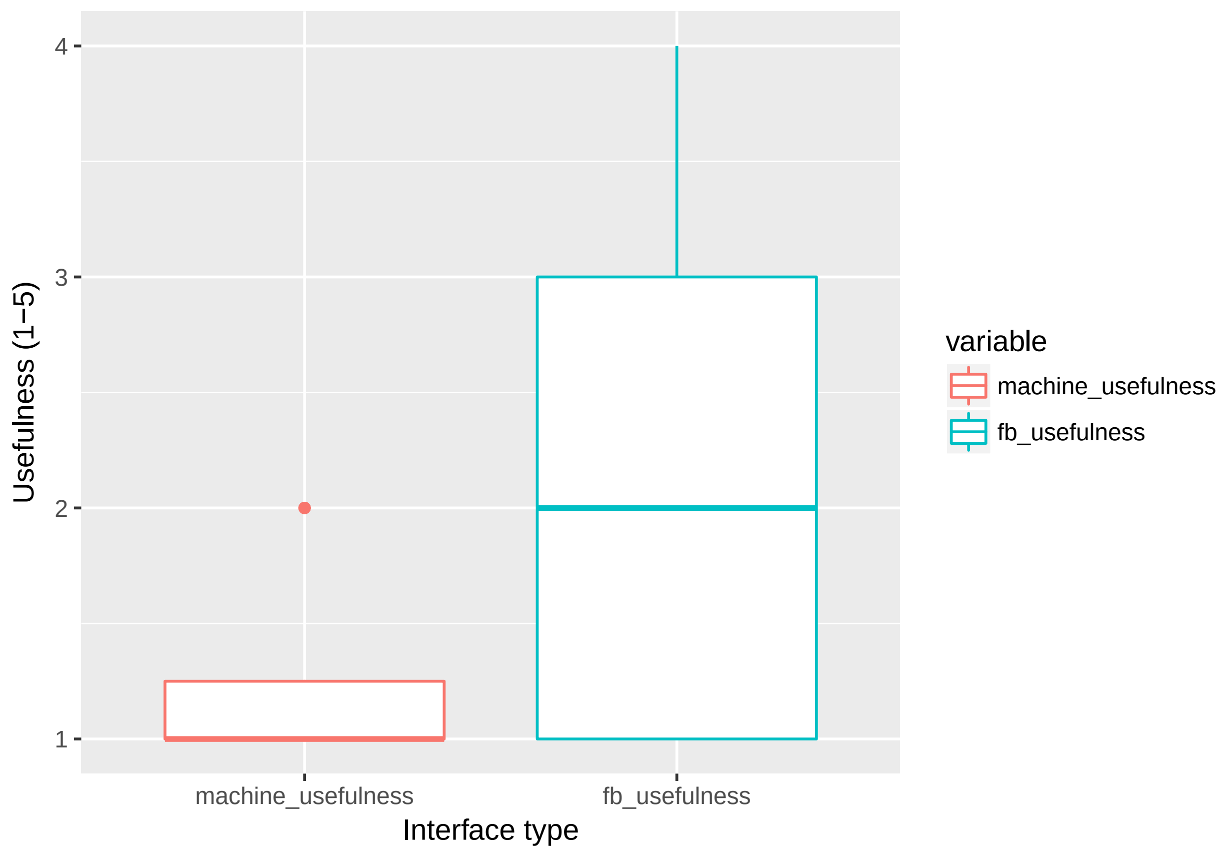

```
ggsave("sat-return.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

Usefulness rating

```
ggplot(melt(dat.preference[2:3]),aes(x=variable,y=value, color=variable)) +
  geom_boxplot() +
  ylab("Usefulness (1-5)") + xlab("Interface type")
```

```
## No id variables; using all as measure variables
```




```
ggsave("usefulness.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

Summary stats

```
kable(dat.er[3:4] %>%
  summarise_all(funs(mean, median, sd)))
```

t1e_mean	t2e_mean	t1e_median	t2e_median	t1e_sd	t2e_sd
0.4375	1.375	0	2	0.813941	1.024695

```
xtable(melt(dat.er[3:4]) %>%
         group_by(variable) %>%
        summarise_all(funs(mean, median, sd)))
```

```
## No id variables; using all as measure variables
```

```
## % latex table generated in R 3.4.0 by xtable 1.8-2 package
## % Mon Dec 11 20:33:28 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrr}
##   \hline
##  & variable & mean & median & sd \\
##   \hline
## 1 & t1e & 0.44 & 0.00 & 0.81 \\
##   2 & t2e & 1.38 & 2.00 & 1.02 \\
```

```
##    \hline
## \end{tabular}
## \end{table}
```

```r
kable(dat.satisfaction[2:6] %>%
  summarise_all(funs(mean)))
```

shelf	list	fb	m_return	fb_return
1.625	1.375	2.1875	1.625	2.0625

```r
kable(dat.satisfaction[2:6] %>%
        summarise_all(funs(median)))
```

shelf	list	fb	m_return	fb_return
1.5	1	2	1	2

```r
kable(dat.satisfaction[2:6] %>%
        summarise_all(funs(sd)))
```

shelf	list	fb	m_return	fb_return
0.7187953	0.6191392	0.8341663	0.8850612	0.9287088

```r
kable(dat.time[3:7] %>%
  summarise_all(funs(mean)))
```

shelf	list	fb	m_return	fb_return
25.95362	39.8145	82.29094	49.23006	56.8215

```r
kable(dat.time[3:7] %>%
        summarise_all(funs(median)))
```

shelf	list	fb	m_return	fb_return
20.3175	38.238	71.22	50.1585	56.8215

```r
kable(dat.time[3:7] %>%
        summarise_all(funs(sd)))
```

shelf	list	fb	m_return	fb_return
18.1288	21.71433	57.43662	20.91993	21.80287

```r
xtable(melt(dat.time[3:7]) %>%
  group_by(variable) %>%
  summarise_all(funs(mean, median, sd)))
```

```
## No id variables; using all as measure variables
```

```
## % latex table generated in R 3.4.0 by xtable 1.8-2 package
## % Mon Dec 11 20:33:29 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrr}
##   \hline
##  & variable & mean & median & sd \\
##   \hline
## 1 & shelf & 25.95 & 20.32 & 18.13 \\
##   2 & list & 39.81 & 38.24 & 21.71 \\
##   3 & fb & 82.29 & 71.22 & 57.44 \\
##   4 & m\_return & 49.23 & 50.16 & 20.92 \\
##   5 & fb\_return & 56.82 & 56.82 & 21.80 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
kable(dat.preference[2:3] %>%
        summarise_all(funs(mean, median)))
```

machine_usefulness_mean	fb_usefulness_mean	machine_usefulness_median	fb_usefulness_median
1.25	1.9375	1	2

```
xtable(melt(dat.satisfaction[2:6]) %>%
         group_by(variable) %>%
         summarise_all(funs(mean, median, sd)))
```

```
## No id variables; using all as measure variables
```

```
## % latex table generated in R 3.4.0 by xtable 1.8-2 package
## % Mon Dec 11 20:33:29 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrr}
##   \hline
##  & variable & mean & median & sd \\
##   \hline
## 1 & shelf & 1.62 & 1.50 & 0.72 \\
##   2 & list & 1.38 & 1.00 & 0.62 \\
##   3 & fb & 2.19 & 2.00 & 0.83 \\
##   4 & m\_return & 1.62 & 1.00 & 0.89 \\
##   5 & fb\_return & 2.06 & 2.00 & 0.93 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
xtable(melt(dat.preference[2:3]) %>%
        group_by(variable) %>%
        summarise_all(funs(mean, median, sd)))
```

```
## No id variables; using all as measure variables
```

```
## % latex table generated in R 3.4.0 by xtable 1.8-2 package
## % Mon Dec 11 20:33:29 2017
## \begin{table}[ht]
```

```
## \centering
## \begin{tabular}{rlrrr}
##   \hline
##  & variable & mean & median & sd \\
##   \hline
## 1 & machine\_usefulness & 1.25 & 1.00 & 0.45 \\
##   2 & fb\_usefulness & 1.94 & 2.00 & 1.00 \\
##    \hline
## \end{tabular}
## \end{table}
```

To prepare time data before running mixed factorial ANOVA, run tests to determine if ANOVA assumptions hold. If data has homogenuos variance (Bartlett) and is normally distributed (Shapiro Wilk).

```r
bartlett.test(dat.time[2:5])
```

```
## Warning in FUN(X[[i]], ...): Calling var(x) on a factor x is deprecated and will become an error.
##   Use something like 'all(duplicated(x)[-1L])' to test for a constant vector.
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  dat.time[2:5]
## Bartlett's K-squared = 131.33, df = 3, p-value < 2.2e-16
```

Put data in the correct format

```r
dat.aov <- melt(dat.time[2:5])
```

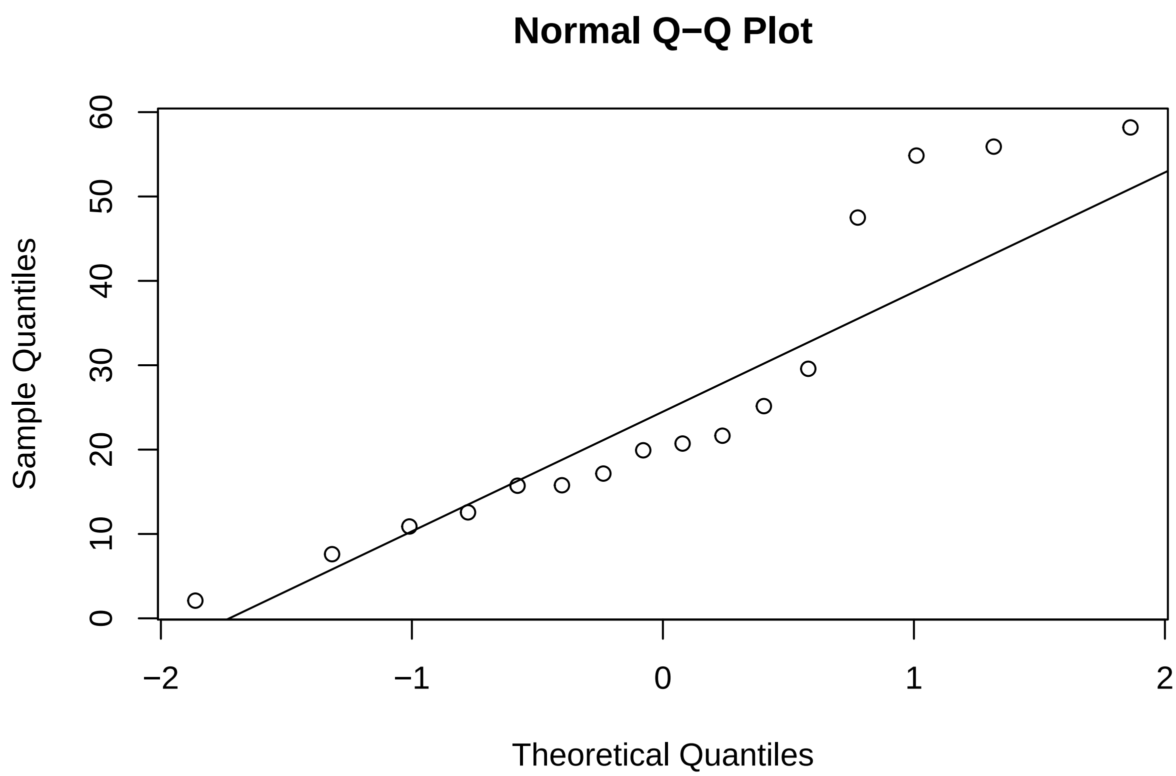
```
## Using h as id variables
```

```r
shapiro.test(dat.aov$value)
```

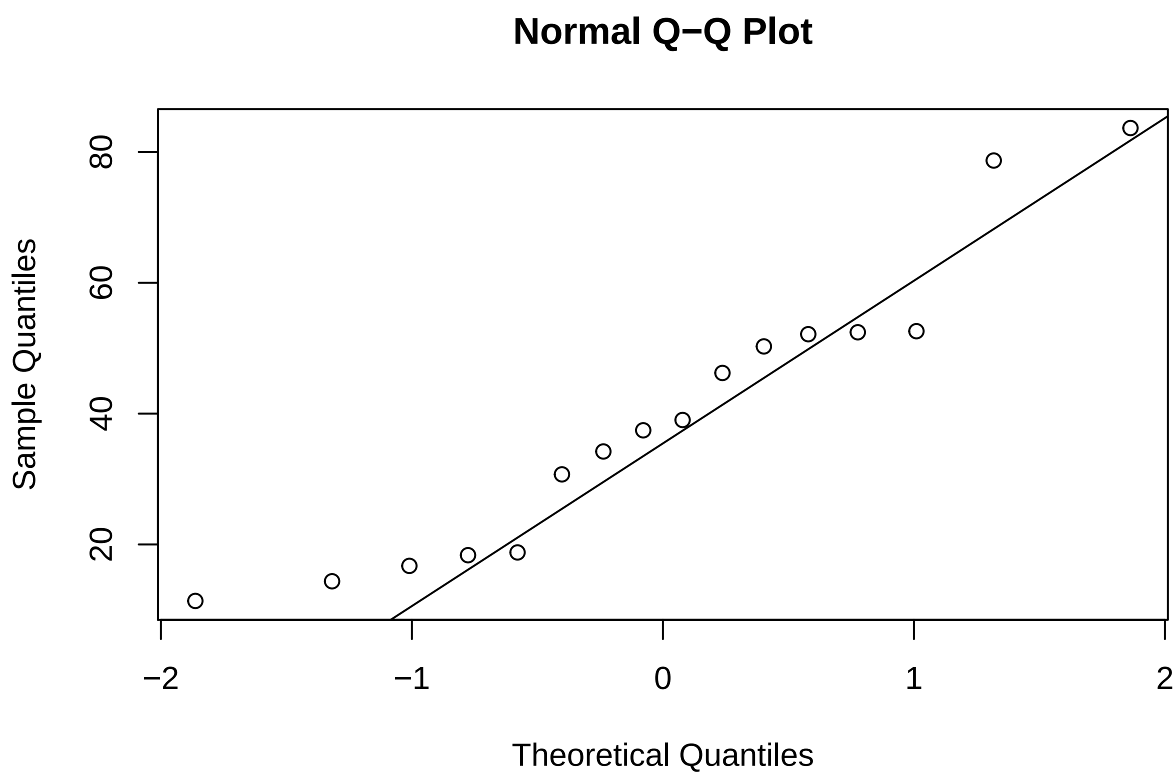
```
##
##  Shapiro-Wilk normality test
##
## data:  dat.aov$value
## W = 0.7122, p-value = 2.194e-08
```

Plot quantile-quantile to see how far data deviates from normal. Normal assumption seems to hold so we can run our ANOVA.

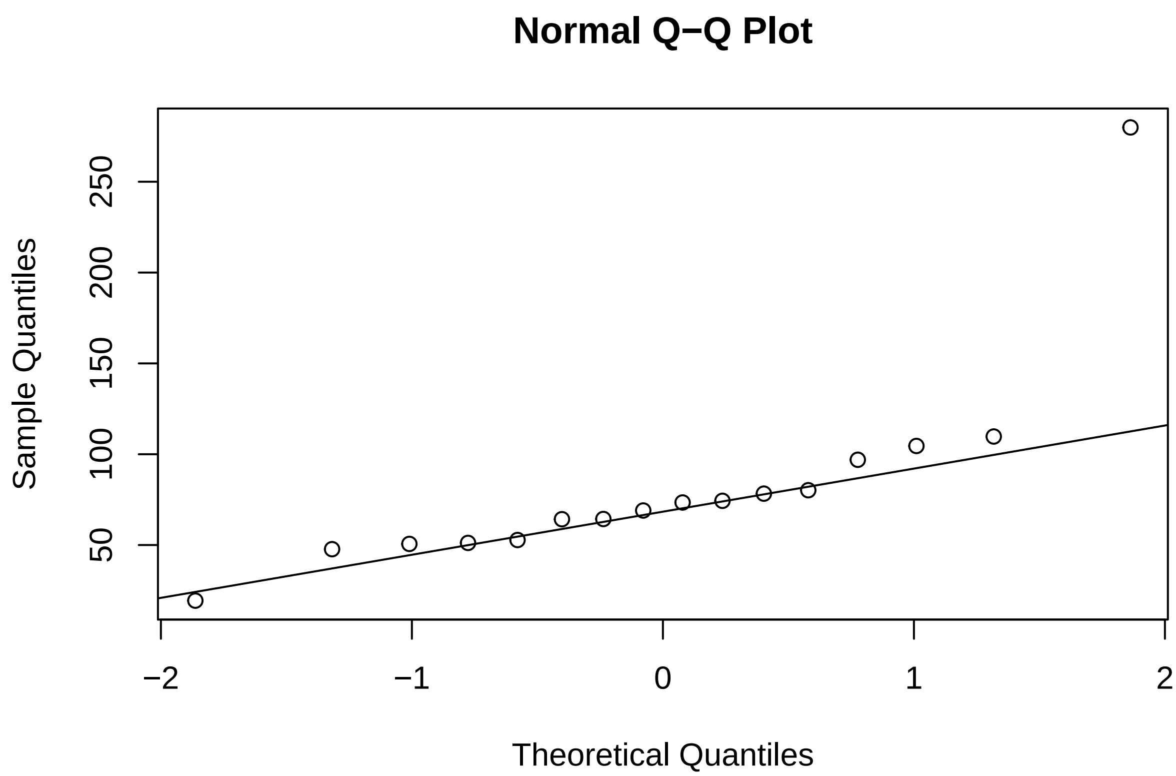
```r
qqnorm(subset(dat.aov, variable == "shelf")$value);qqline(subset(dat.aov, variable == "shelf")$value)
```




```r
qqnorm(subset(dat.aov, variable == "list")$value);qqline(subset(dat.aov, variable == "list")$value)
```




```r
qqnorm(subset(dat.aov, variable == "fb")$value);qqline(subset(dat.aov, variable == "fb")$value)
```



Fit the linear model (value ~ variable * h) value is the measurement (time), variable is the UI type, and h is the home type.

```
fit <- lm(formula = value ~ variable * h, data = dat.aov)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: value
##             Df Sum Sq Mean Sq F value    Pr(>F)
## variable     2  27575 13787.4  9.9525 0.0002897 ***
## h            1   1584  1584.3  1.1436 0.2910005
## variable:h   2   1719   859.6  0.6205 0.5425175
## Residuals   42  58183  1385.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This is the same as above but using the ANOVA native function of R

```
aov.fit <- aov(formula = value ~ variable * h, data = dat.aov)
summary(aov.fit)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## variable     2  27575   13787   9.952 0.00029 ***
## h            1   1584    1584   1.144 0.29100
## variable:h   2   1719     860   0.621 0.54252
## Residuals   42  58183    1385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tukey Honest Significant Differences is a post-hoc test to determine the difference in means in terms of the UI type factor (variable).

```r
TukeyHSD(aov.fit, "variable", ordered = TRUE)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = value ~ variable * h, data = dat.aov)
##
## $variable
##                 diff       lwr      upr     p adj
## list-shelf 13.86087 -18.10937 45.83112 0.5479598
## fb-shelf   56.33731  24.36707 88.30755 0.0003055
## fb-list    42.47644  10.50620 74.44668 0.0067032
```

Significant effect of interface, but no interaction effect of housing type and interface Difference between FB and shelf and FB and list is significant.

```r
dat.ret <- melt(dat.time[dat.time$p != 2589, 6:7])
```

```
## No id variables; using all as measure variables
```

```r
t.test(subset(dat.ret, variable == "m_return")$value, subset(dat.ret, variable == "fb_return")$value, pa
```

```
##
##  Paired t-test
##
## data:  subset(dat.ret, variable == "m_return")$value and subset(dat.ret, variable == "fb_return")$val
## t = -1.4812, df = 14, p-value = 0.1607
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -23.599834   4.319168
## sample estimates:
## mean of the differences
##               -9.640333
```

No significant effect of return task time!

Prepare data to run friedman test on satisfaction (3 factor)

```r
dat.fr <- as.matrix(dat.satisfaction[2:4])
friedman.test(dat.fr)
```

```
##
##  Friedman rank sum test
##
## data:  dat.fr
## Friedman chi-squared = 9.2558, df = 2, p-value = 0.009775
```

Significant difference detected, to see what it is we conduct Wilcoxon tests with "less" hypotheses"

```r
wilcox.test(dat.satisfaction$list, dat.satisfaction$fb, paired = T, alternative = "less")
```

```
## Warning in wilcox.test.default(dat.satisfaction$list, dat.satisfaction
## $fb, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(dat.satisfaction$list, dat.satisfaction
## $fb, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
```

```
##
## data:  dat.satisfaction$list and dat.satisfaction$fb
## V = 9, p-value = 0.008271
## alternative hypothesis: true location shift is less than 0
```

```
wilcox.test(dat.satisfaction$shelf, dat.satisfaction$fb, paired = T, alternative = "less")
```

```
## Warning in wilcox.test.default(dat.satisfaction$shelf, dat.satisfaction
## $fb, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(dat.satisfaction$shelf, dat.satisfaction
## $fb, : cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat.satisfaction$shelf and dat.satisfaction$fb
## V = 14, p-value = 0.04284
## alternative hypothesis: true location shift is less than 0
```

All FB tests show significant difference from FB.

```
wilcox.test(dat.satisfaction$list, dat.satisfaction$shelf, paired = T, alternative = "less")
```

```
## Warning in wilcox.test.default(dat.satisfaction$list, dat.satisfaction
## $shelf, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(dat.satisfaction$list, dat.satisfaction
## $shelf, : cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat.satisfaction$list and dat.satisfaction$shelf
## V = 7, p-value = 0.1201
## alternative hypothesis: true location shift is less than 0
```

No significant difference between list and shelf though. Test usefulness. Significant results.

```
wilcox.test(dat.preference$machine_usefulness, dat.preference$fb_usefulness, paired = T)
```

```
## Warning in wilcox.test.default(dat.preference$machine_usefulness,
## dat.preference$fb_usefulness, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(dat.preference$machine_usefulness,
## dat.preference$fb_usefulness, : cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat.preference$machine_usefulness and dat.preference$fb_usefulness
## V = 3.5, p-value = 0.02355
## alternative hypothesis: true location shift is not equal to 0
```

Alternative hypothesis is machine usefulness is less (better) than fb usefulness

```
wilcox.test(dat.preference$machine_usefulness, dat.preference$fb_usefulness, paired = T, alternative =
```

```
## Warning in wilcox.test.default(dat.preference$machine_usefulness,
## dat.preference$fb_usefulness, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(dat.preference$machine_usefulness,
## dat.preference$fb_usefulness, : cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat.preference$machine_usefulness and dat.preference$fb_usefulness
## V = 3.5, p-value = 0.01178
## alternative hypothesis: true location shift is less than 0
```

Error rate is simply counts so we assume it's not normally distributed

```
wilcox.test(dat.er$t1e, dat.er$t2e, paired = T)
```

```
## Warning in wilcox.test.default(dat.er$t1e, dat.er$t2e, paired = T): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(dat.er$t1e, dat.er$t2e, paired = T): cannot
## compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat.er$t1e and dat.er$t2e
## V = 10, p-value = 0.01874
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(dat.er$t1e, dat.er$t2e, paired = T, alternative = "less")
```

```
## Warning in wilcox.test.default(dat.er$t1e, dat.er$t2e, paired = T,
## alternative = "less"): cannot compute exact p-value with ties

## Warning in wilcox.test.default(dat.er$t1e, dat.er$t2e, paired = T,
## alternative = "less"): cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  dat.er$t1e and dat.er$t2e
## V = 10, p-value = 0.009371
## alternative hypothesis: true location shift is less than 0
```

Instead of testing for interaction with non-parametric tests, we run tests comparing student housing for shelf with private residence for shelf, compare error, usefulness and satisfaction with wilcox

```
wilcox.test(subset(dat.satisfaction, h == "Student residence")$shelf, subset(dat.satisfaction, h == "Pri
```

```
## Warning in wilcox.test.default(subset(dat.satisfaction, h == "Student
## residence")$shelf, : cannot compute exact p-value with ties

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  subset(dat.satisfaction, h == "Student residence")$shelf and subset(dat.satisfaction, h == "Pr
## W = 28, p-value = 0.6854
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(subset(dat.satisfaction, h == "Student residence")$list, subset(dat.satisfaction, h == "Pri
```

```
## Warning in wilcox.test.default(subset(dat.satisfaction, h == "Student
## residence")$list, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  subset(dat.satisfaction, h == "Student residence")$list and subset(dat.satisfaction, h == "Pr
## W = 37, p-value = 0.5613
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(subset(dat.satisfaction, h == "Student residence")$fb, subset(dat.satisfaction, h == "Priva
```

```
## Warning in wilcox.test.default(subset(dat.satisfaction, h == "Student
## residence")$fb, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  subset(dat.satisfaction, h == "Student residence")$fb and subset(dat.satisfaction, h == "Priva
## W = 24.5, p-value = 0.4268
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(subset(dat.preference, h == "Student residence")$machine_usefulness, subset(dat.preference,
```

```
## Warning in wilcox.test.default(subset(dat.preference, h == "Student
## residence")$machine_usefulness, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  subset(dat.preference, h == "Student residence")$machine_usefulness and subset(dat.preference
## W = 24, p-value = 0.2946
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(subset(dat.er, h == "Student residence")$t1e, subset(dat.er, h == "Private house")$t1e)
```

```
## Warning in wilcox.test.default(subset(dat.er, h == "Student residence")
## $t1e, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  subset(dat.er, h == "Student residence")$t1e and subset(dat.er, h == "Private house")$t1e
## W = 33, p-value = 0.9447
## alternative hypothesis: true location shift is not equal to 0
```

No main effect of home type on error, usefulness or satisfaction. No real need to test interaction. When we asked people which one they prefer, the answers look like this:

```
kable(dat.preference[4:5])
```

shelf_vs_list	app_vs_fb
3	4
5	5
2	5
2	4
3	4
5	5
2	3
2	1
1	1

shelf_vs_list	app_vs_fb
1	1
2	5
1	1
4	4
2	5
4	5
2	5

```r
kable(dat.preference[4:5] %>%
        summarise_all(funs(mean, median)))
```

shelf_vs_list_mean	app_vs_fb_mean	shelf_vs_list_median	app_vs_fb_median
2.5625	3.625	2	4

```r
kable(melt(dat.preference[4:5]) %>%
        group_by(variable) %>%
        summarise_all(funs(mean, median, sd)))
```

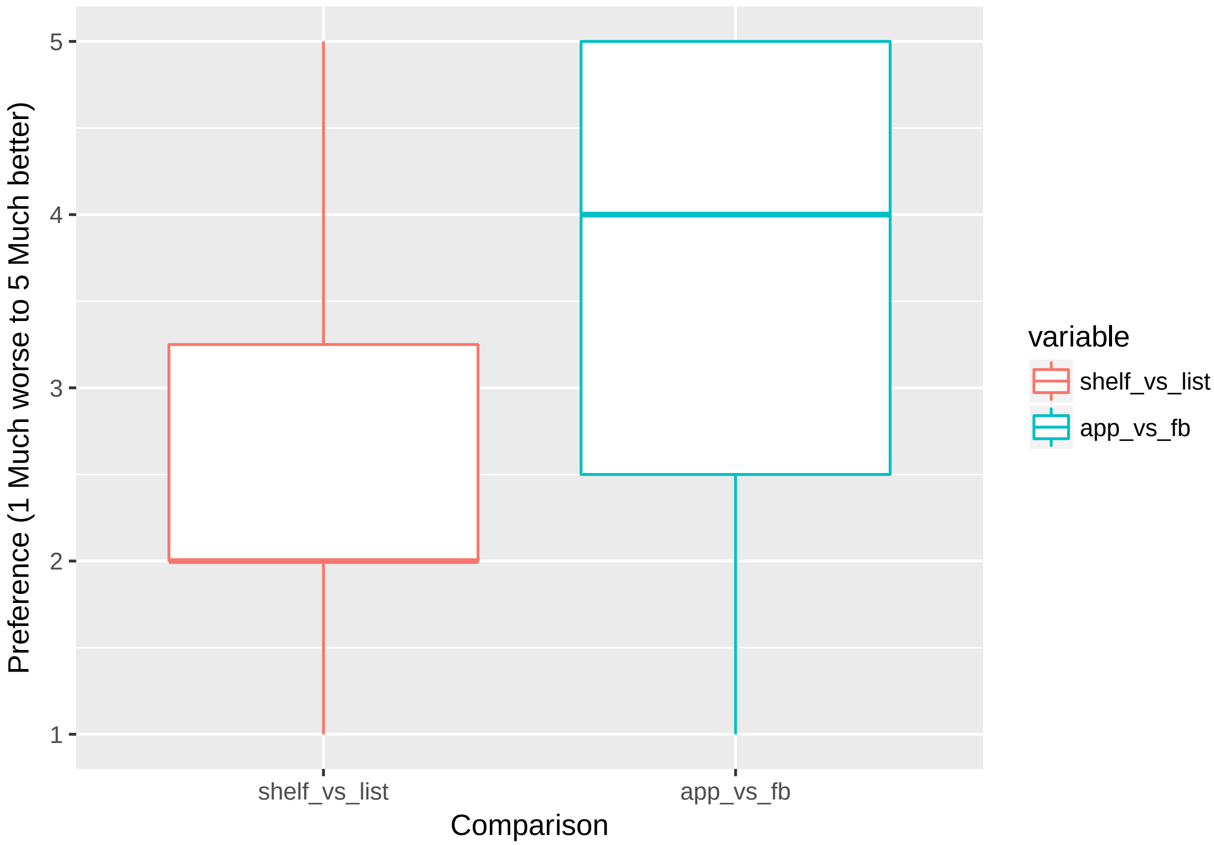
```
## No id variables; using all as measure variables
```

variable	mean	median	sd
shelf_vs_list	2.5625	2	1.314978
app_vs_fb	3.6250	4	1.668332

There's no need to analyze this. A plot summarizes the results.

```r
ggplot(melt(dat.preference[4:5]),aes(x=variable,y=value, color=variable)) +
  geom_boxplot() +
  ylab("Preference (1 Much worse to 5 Much better)") + xlab("Comparison")
```

```
## No id variables; using all as measure variables
```




```
ggsave("pref-summary.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

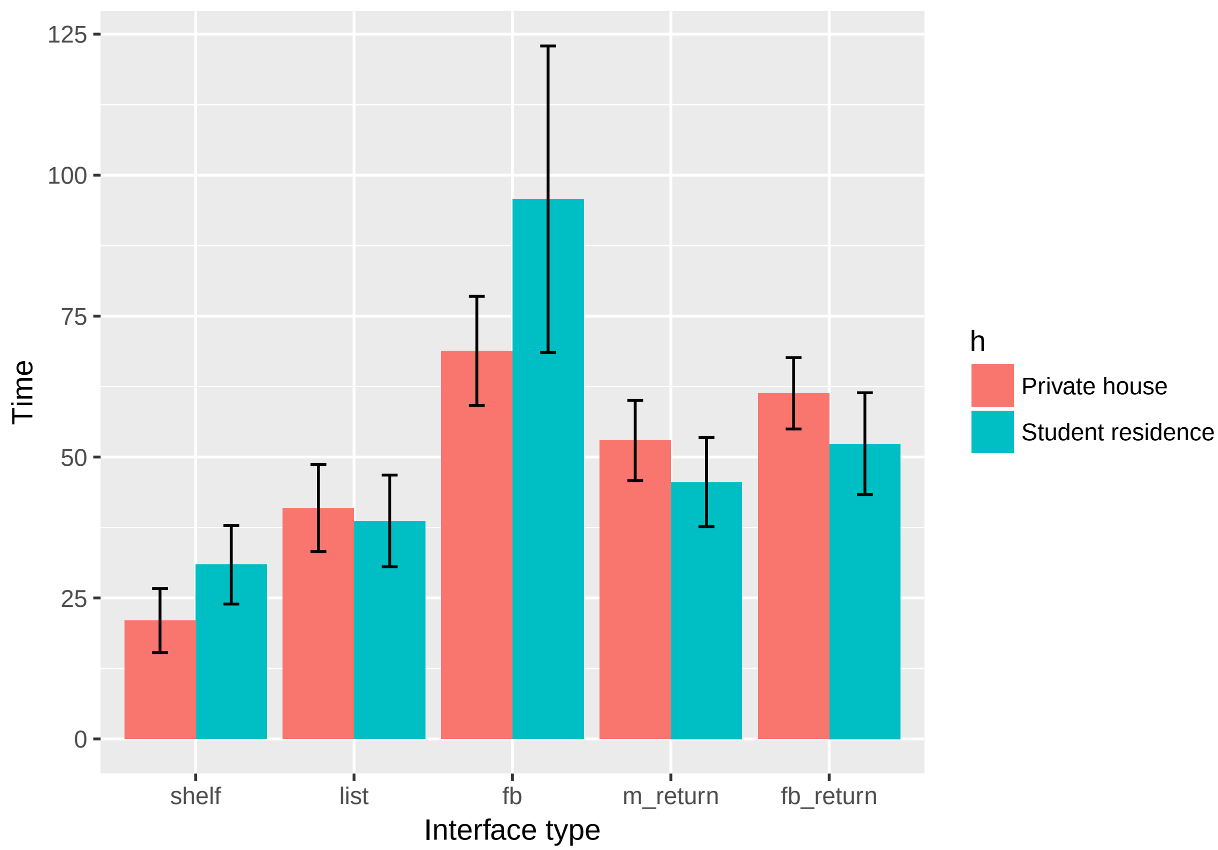
Barcharts with CIs

```
std.err <- function(x) sd(x)/sqrt(length(x))

timedat <- melt(dat.time[2:7]) %>%
  group_by(variable, h) %>%
  summarise_all(funs(mean, std.err))
```

```
## Using h as id variables
```

```
ggplot(timedat,aes(x=variable,y=mean, fill=h)) +
  geom_bar(stat="identity",position="dodge") +
  geom_errorbar(aes(ymin=mean-std.err, ymax=mean+std.err), width=.2, position=position_dodge(.9)) +
  ylab("Time") + xlab("Interface type")
```



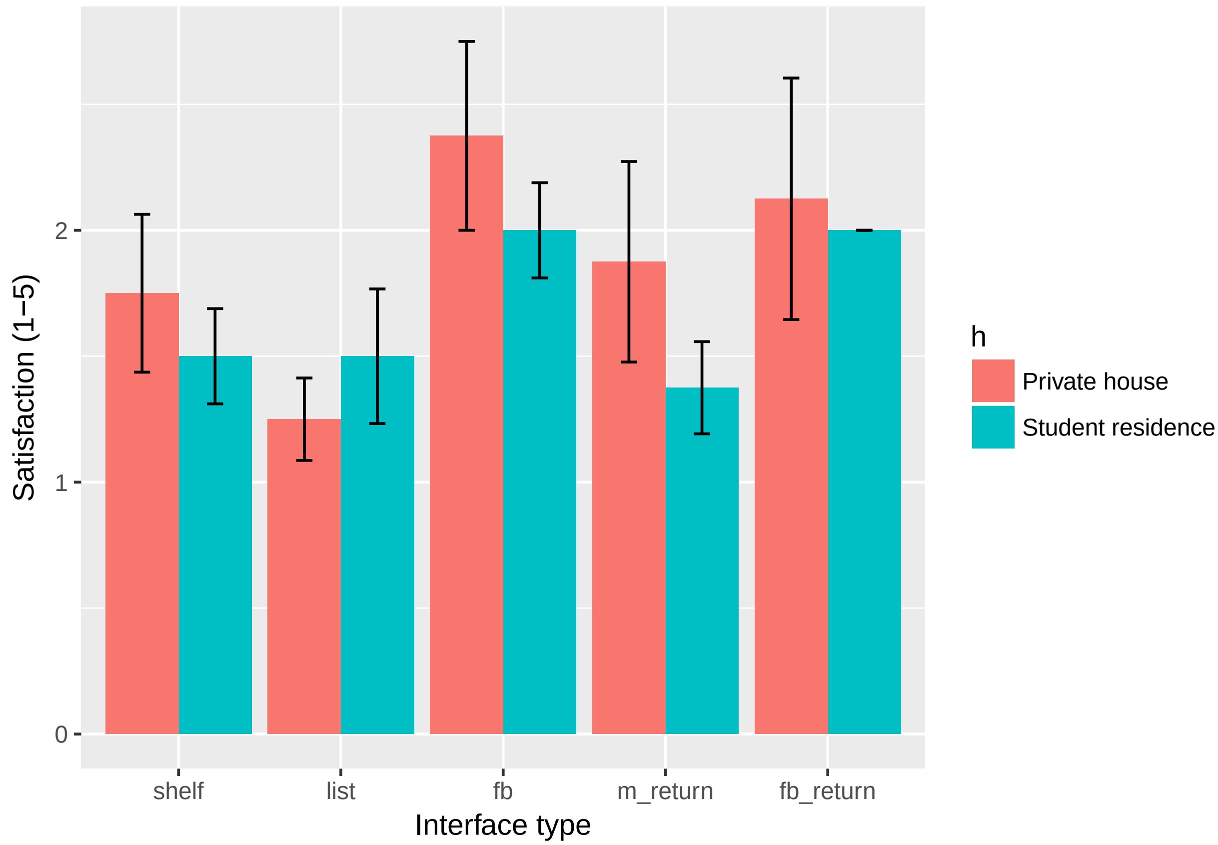

```
ggsave("bar-time.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

```
satdat <- melt(dat.satisfaction[2:7]) %>%
  group_by(variable, h) %>%
  summarise_all(funs(mean, std.err))
```

```
## Using h as id variables
```

```
ggplot(satdat,aes(x=variable,y=mean, fill=h)) +
  geom_bar(stat="identity",position="dodge") +
  geom_errorbar(aes(ymin=mean-std.err, ymax=mean+std.err), width=.2, position=position_dodge(.9)) +
  ylab("Satisfaction (1-5)") + xlab("Interface type")
```



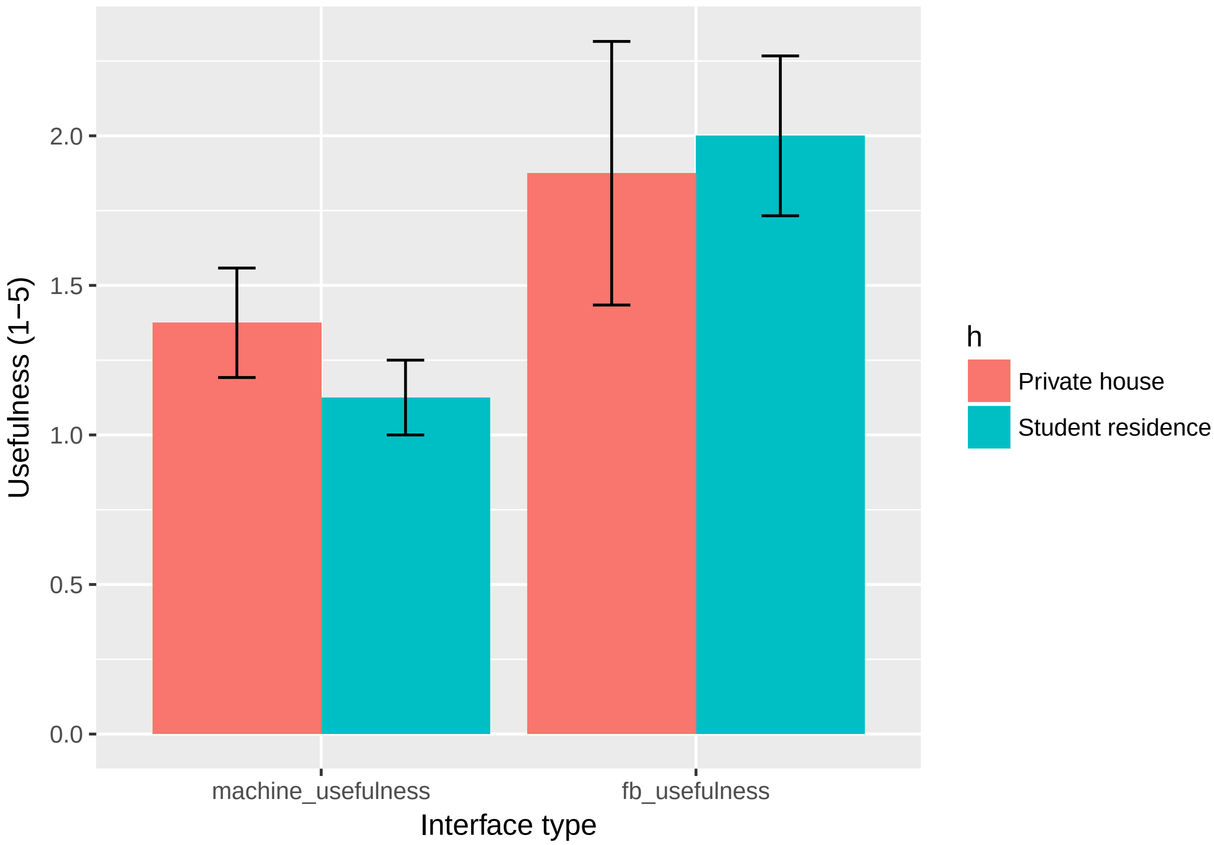

```
ggsave("bar-satisfaction.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

```
prefdat <- melt(dat.preference[c(2, 3, 6)]) %>%
  group_by(variable, h) %>%
  summarise_all(funs(mean, std.err))
```

```
## Using h as id variables
```

```
ggplot(prefdat,aes(x=variable,y=mean, fill=h)) +
  geom_bar(stat="identity",position="dodge") +
  geom_errorbar(aes(ymin=mean-std.err, ymax=mean+std.err), width=.2, position=position_dodge(.9)) +
  ylab("Usefulness (1-5)") + xlab("Interface type")
```




```r
ggsave("bar-usefulness.png", scale = 1, path = "plots/")
```

```
## Saving 6.5 x 4.5 in image
```

Adjusted p-values for pairwise tests The values are of the following tests: 1) Satisfaction list vs Facebook 2) satisfaction shelf vs Facebook 3) satisfaction list vs shelf 4) usefulness machine vs FB 5) Errors shelf vs list

```r
p.adjust(c(0.008271, 0.04284,0.1201, 0.01178, 0.009371), method = "bonferroni")
```

```
## [1] 0.041355 0.214200 0.600500 0.058900 0.046855
```