

Final Project Code

Kunwu Lyu and Evan Hart

2025-03-16

Data Wrangling

```
## Data from ICPSR
survey <- read_tsv("data/ICPSR_37143/DS0001/37143-0001-Data.tsv") %>%
  janitor::clean_names() # To all lower case
receipt <- read_tsv("data/ICPSR_37143/DS0002/37143-0002-Data.tsv") %>%
  janitor::clean_names()
fast_food <- read_tsv("data/ICPSR_37143/DS0003/37143-0003-Data.tsv") %>%
  janitor::clean_names()
grocery <- read_tsv("data/ICPSR_37143/DS0004/37143-0004-Data.tsv") %>%
  janitor::clean_names()
recall <- read_tsv("data/ICPSR_37143/DS0005/37143-0005-Data.tsv") %>%
  janitor::clean_names()

## Combine multiple surveys
full_data <- survey %>%
  full_join(receipt, relationship = "many-to-many") %>%
  full_join(fast_food, relationship = "many-to-many") %>%
  full_join(grocery, relationship = "many-to-many") %>%
  full_join(recall, relationship = "many-to-many")

## Mutating
full_data <- full_data %>%
  mutate(limit = ordered(q75, levels = c("Never", "Seldom", "Sometimes",
                                         "Often", "Always")))) %>% # for ordinal
  mutate(age = as.numeric(q76),
         gender = if_else(q77 == 0, "M", "F"),
         race = case_when(
           !is.na(q79_1) ~ "Native",
           !is.na(q79_2) ~ "Black",
           !is.na(q79_3) ~ "Asian",
           !is.na(q79_4) ~ "White",
           !is.na(q79_a) ~ "Other"
         ),
         edu = as.numeric(q80),
         location = nemslocationindicator,
         city = q1,
         num_kids = q44,
         surveydate = dmy(surveydate)) %>%
  mutate(days_since_ban =
        as.numeric(interval(as.Date("2013-03-12"), surveydate) / days(1))) %>%
  filter(age > 0)
```

```

# Standardize numerical for prediction
standardize <- function(x, na.rm = TRUE) {
  (x - mean(x, na.rm = na.rm)) /
  sd(x, na.rm = na.rm)
}

# subset of complete dataset
reduced_data <- full_data %>%
  mutate(age_std = standardize(as.numeric(q76))) %>%
  select(c("receiptid", "person_id", "limit", "age", "age_std", "gender",
          "race", "edu", "city", "caff", "location", "round", "nsigns(ssb",
          "num_kids", "surveydate", "days_since_ban", "caloriescal", "fatg",
          "sugarg")) %>%
  group_by(receiptid) %>%
  mutate(black = if_else(race == "Black", "Black", "non-Black")) %>%
  mutate(caff = sum(caff, na.rm = T), # across each receipt
         caloriescal = sum(caloriescal, na.rm = T),
         fatg = sum(fatg, na.rm = T),
         sugarg = sum(sugarg, na.rm = T)) %>%
  drop_na() %>%
  distinct() %>% # Remove duplicate rows because multiple items are on a receipt
  mutate(receiptid = as.factor(receiptid),
         person_id = as.factor(person_id),
         location = as.factor(location),
         round = as.factor(round),
         edu = case_when(
           edu == 1 ~ "Less than High School",
           edu == 2 ~ "Some High School",
           edu == 3 ~ "High School",
           edu == 4 ~ "Some College",
           edu == 5 ~ "Associates Degree",
           edu == 6 ~ "College Degree",
           edu == 7 ~ "Graduate Degree"
         )) %>%
  ungroup() %>%
  mutate(
    caff_std = standardize(caff),
    nsigns(ssb)_std = standardize(nsigns(ssb)),
    days_since_ban_std = standardize(days_since_ban),
    caloriescal_std = standardize(caloriescal),
    fatg_std = standardize(fatg),
    sugarg_std = standardize(sugarg)
  )

# Cleaned data
write_csv(reduced_data, "dietControl.csv")

# One receipt can't appear in multiple locations
multi_receipt_locations <- reduced_data %>%
  group_by(receiptid) %>%
  summarize(n_rounds = n_distinct(location)) %>%
  filter(n_rounds > 1) %>%
  pull(receiptid)

```

```

reduced_data %>%
  filter(receiptid %in% multi_receipt_locations) %>%
  count(receiptid, location)

## # A tibble: 0 x 3
## # i 3 variables: receiptid <fct>, location <fct>, n <int>
## But not all stores have been surveyed three times
multi_round_locations <- reduced_data %>%
  group_by(location) %>%
  summarize(n_rounds = n_distinct(round)) %>%
  filter(n_rounds < 3) %>%
  pull(location)

reduced_data %>%
  filter(location %in% multi_round_locations) %>%
  count(location, round)

## # A tibble: 47 x 3
##   location round     n
##   <fct>    <fct> <int>
## 1 B105      1      11
## 2 B127A     1       2
## 3 B127B     1       8
## 4 B127B     3      22
## 5 B205      1       8
## 6 B217      1      16
## 7 B227      1      25
## 8 B227      3      31
## 9 K234      2       7
## 10 K234     3      34
## # i 37 more rows

```

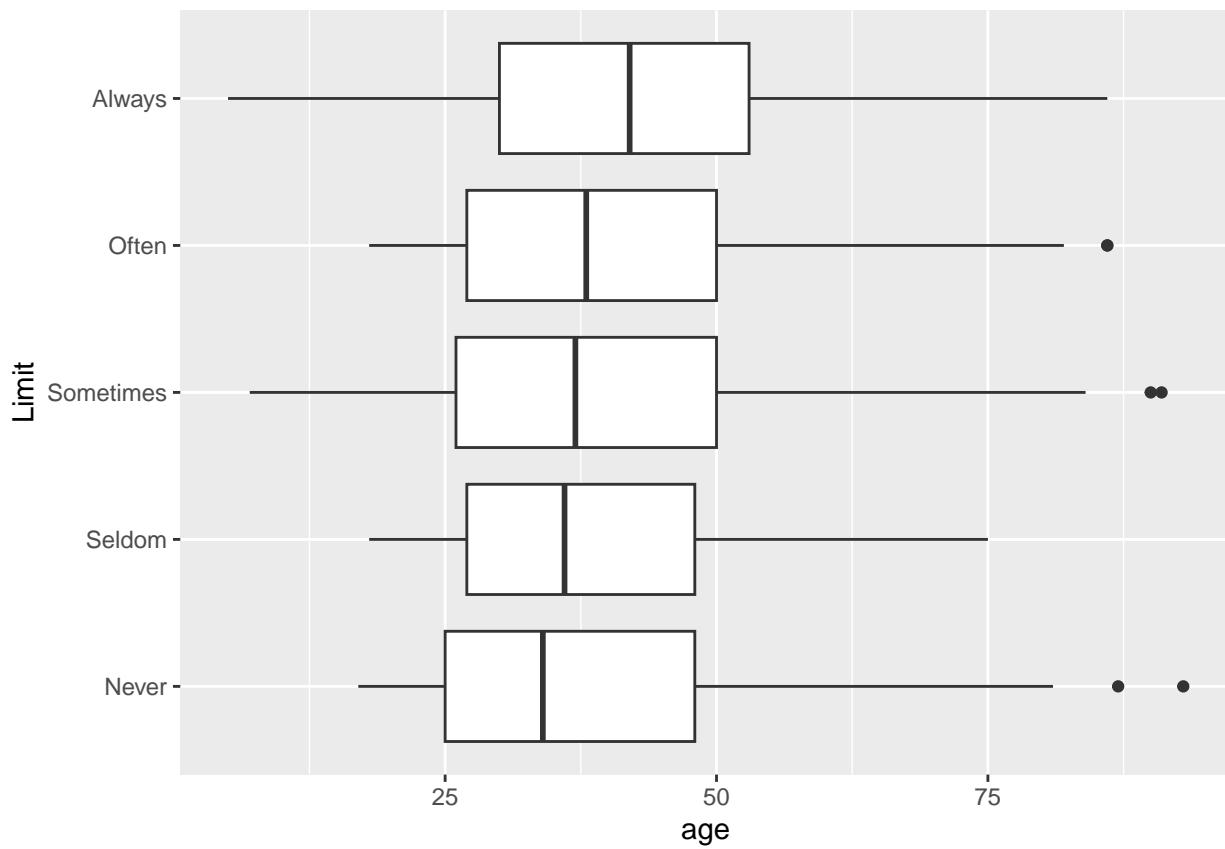
EDA

```

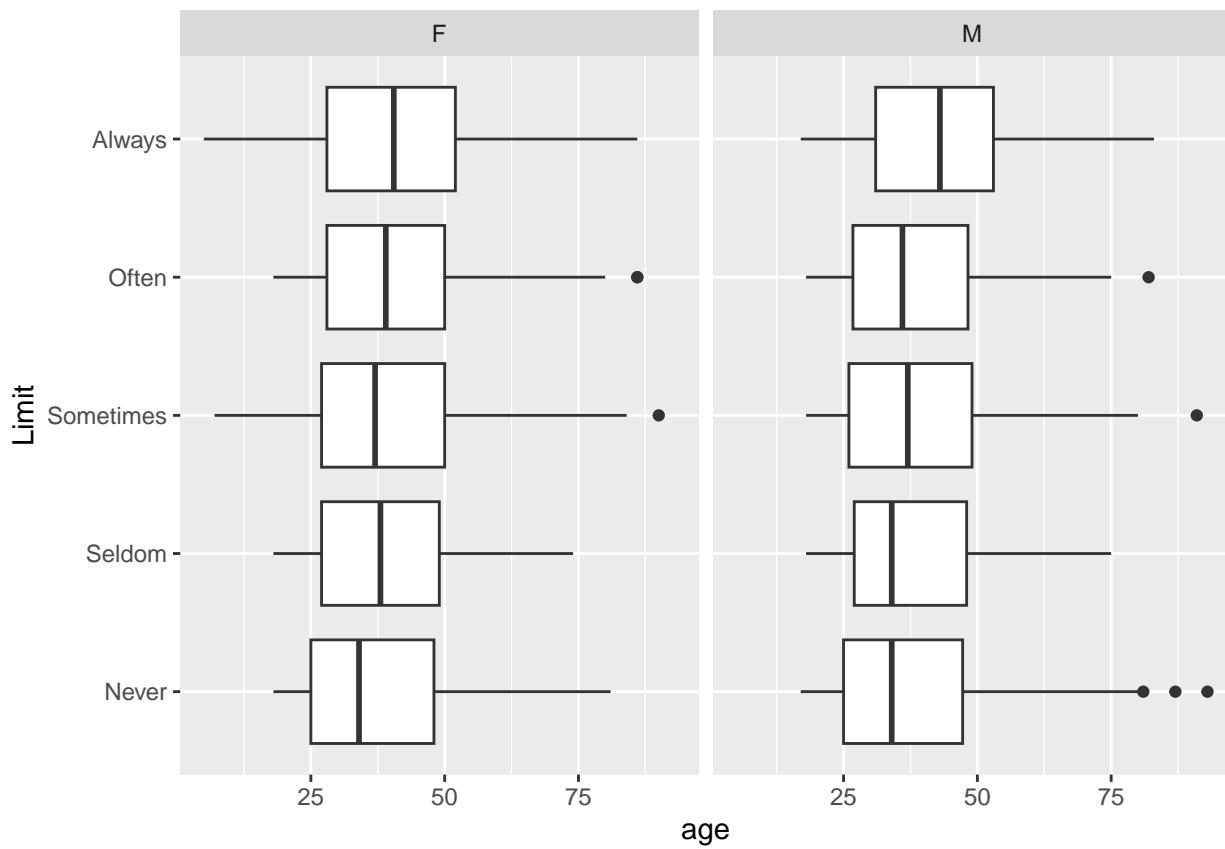
# Single variables, interactions plotted against limit

# Age
ggplot(data = reduced_data, aes(x = age , y = limit)) +
  geom_boxplot() +
  labs(x = "age", y = "Limit")

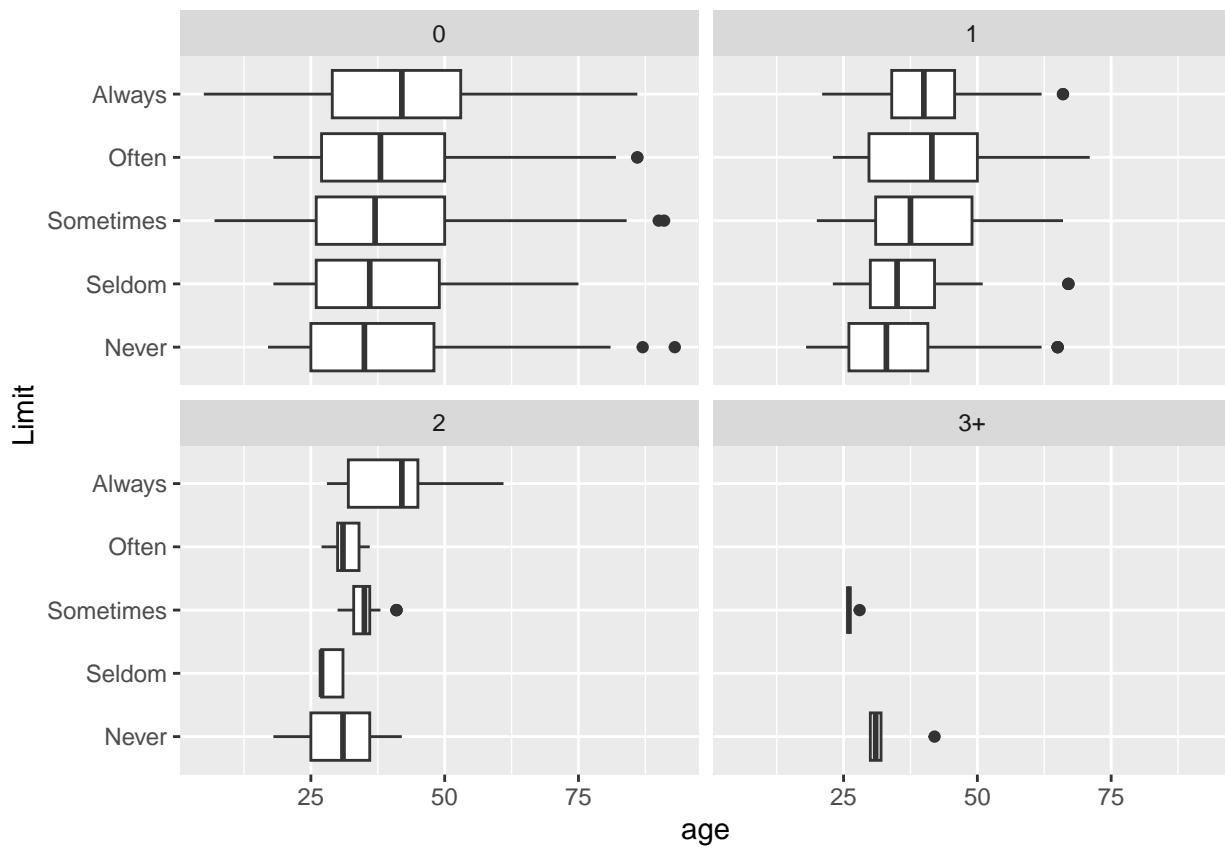
```



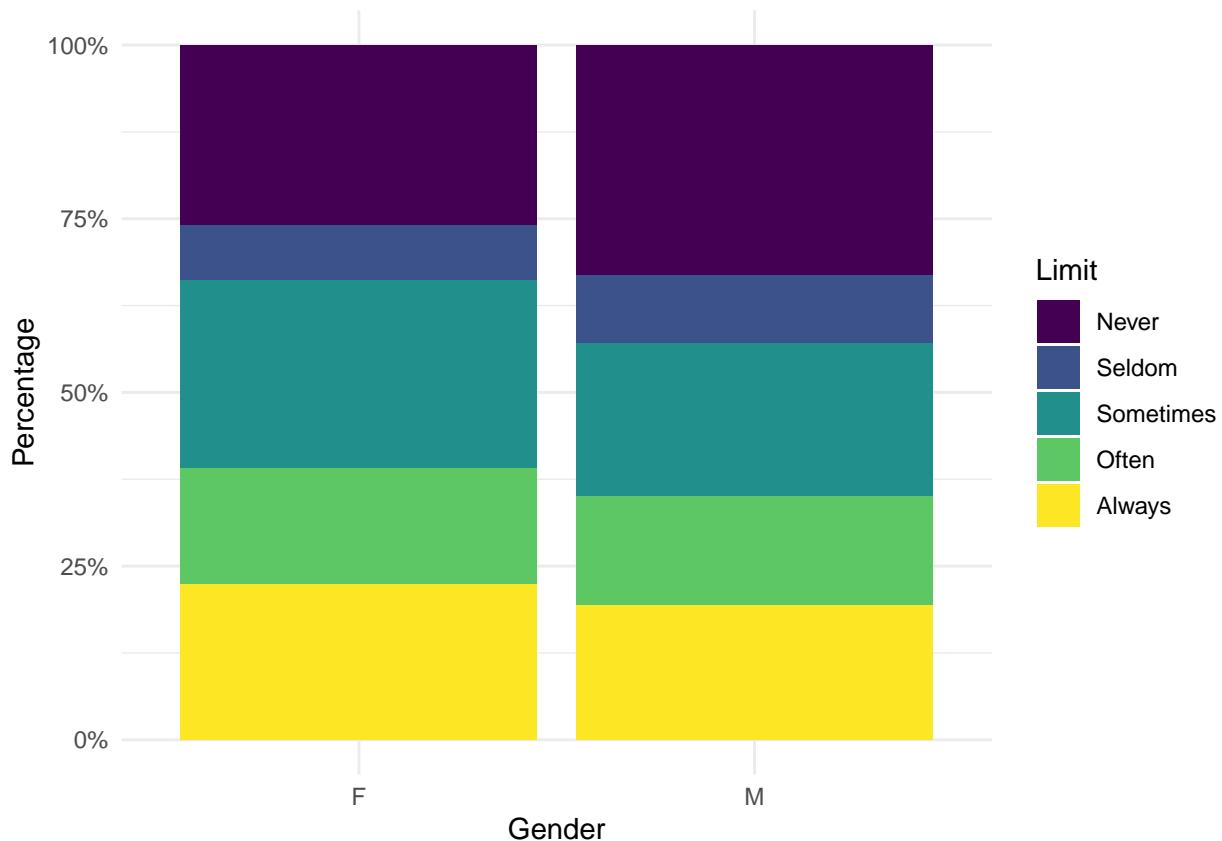
```
# Age faceted by gender
ggplot(data = reduced_data, aes(x = age , y = limit)) +
  geom_boxplot() +
  facet_wrap(~gender) +
  labs(x = "age", y = "Limit")
```



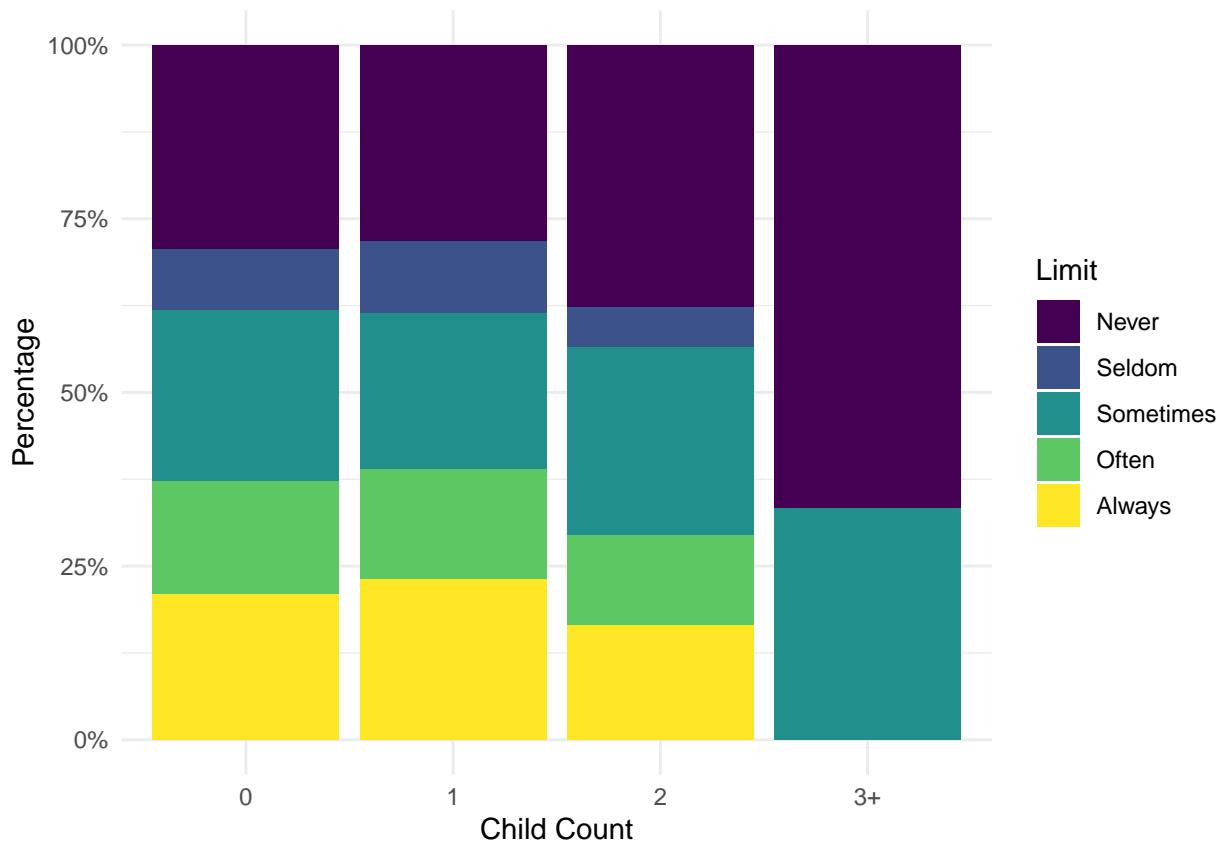
```
# Age faceted by child count
ggplot(data = reduced_data, aes(x = age , y = limit)) +
  geom_boxplot() +
  facet_wrap(~num_kids) +
  labs(x = "age", y = "Limit")
```



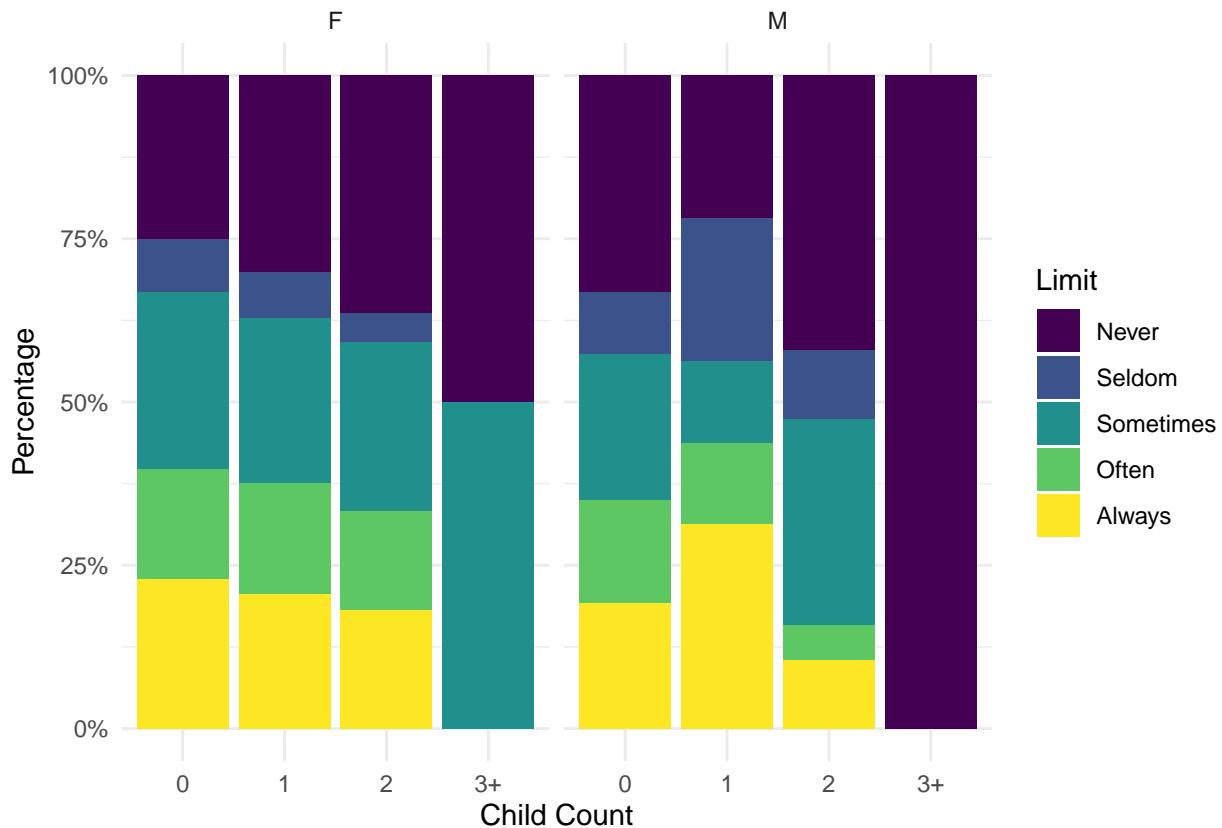
```
ggplot(data = reduced_data, aes(x = gender, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Gender", y = "Percentage", fill = "Limit") +
  theme_minimal()
```



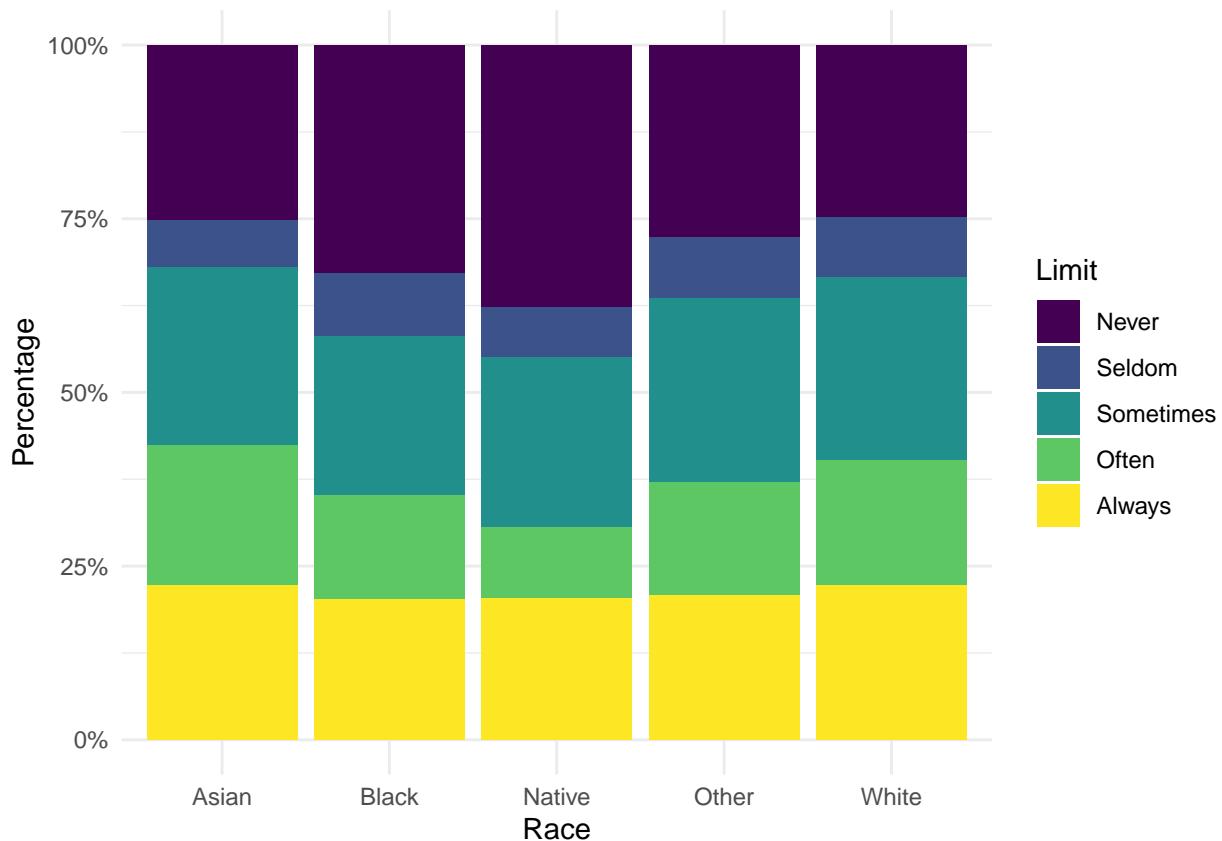
```
# Child count
ggplot(data = reduced_data, aes(x = num_kids, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Child Count", y = "Percentage", fill = "Limit") +
  theme_minimal()
```



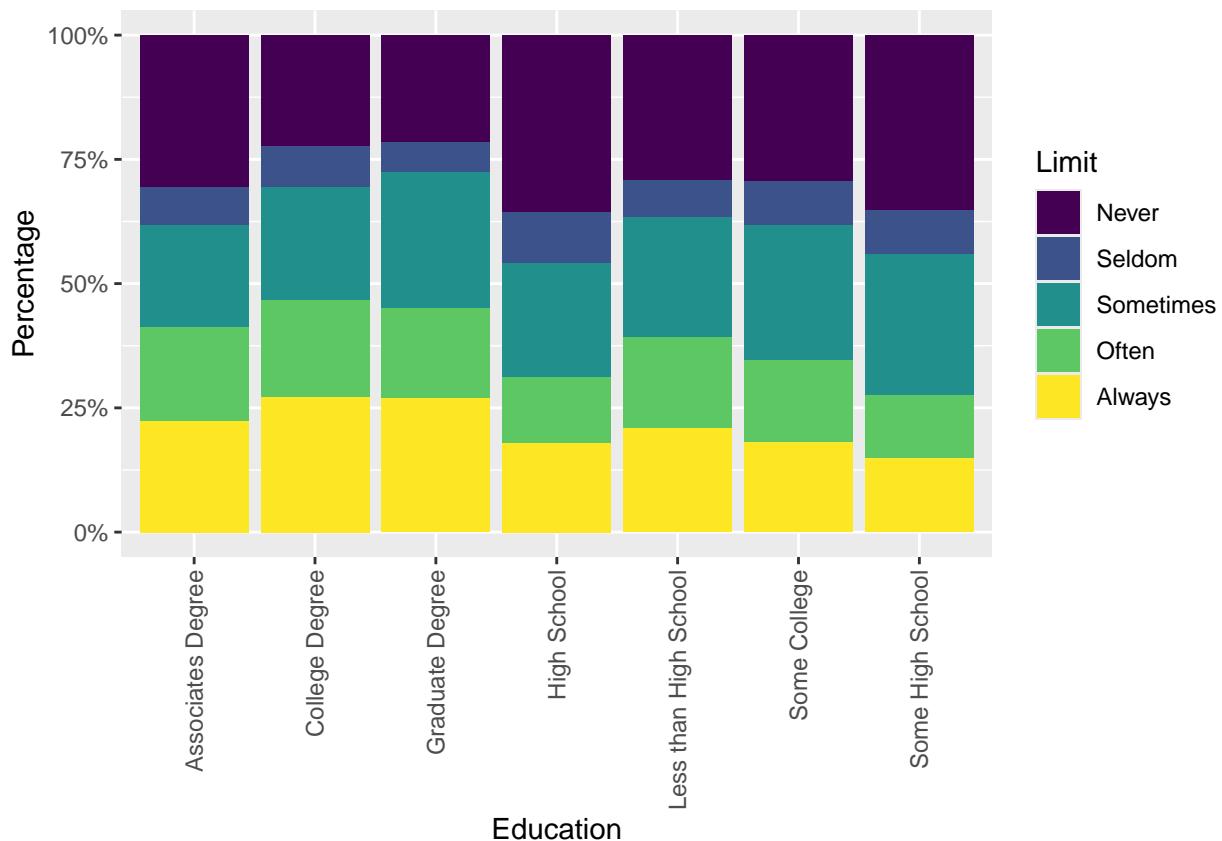
```
# Child count / gender interaction
ggplot(data = reduced_data, aes(x = num_kids, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Child Count", y = "Percentage", fill = "Limit") +
  facet_wrap(~gender) +
  theme_minimal()
```



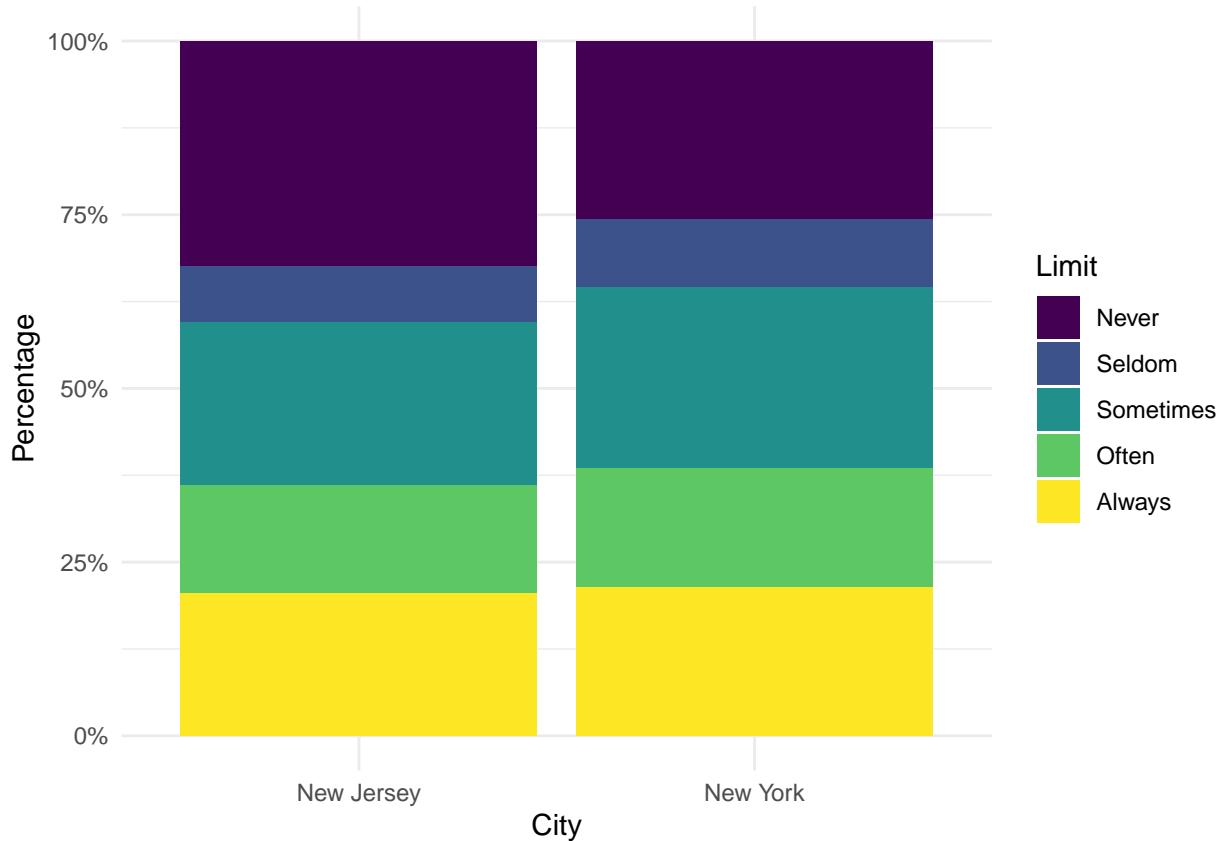
```
# Race
ggplot(data = reduced_data, aes(x = race, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Race", y = "Percentage", fill = "Limit") +
  theme_minimal()
```



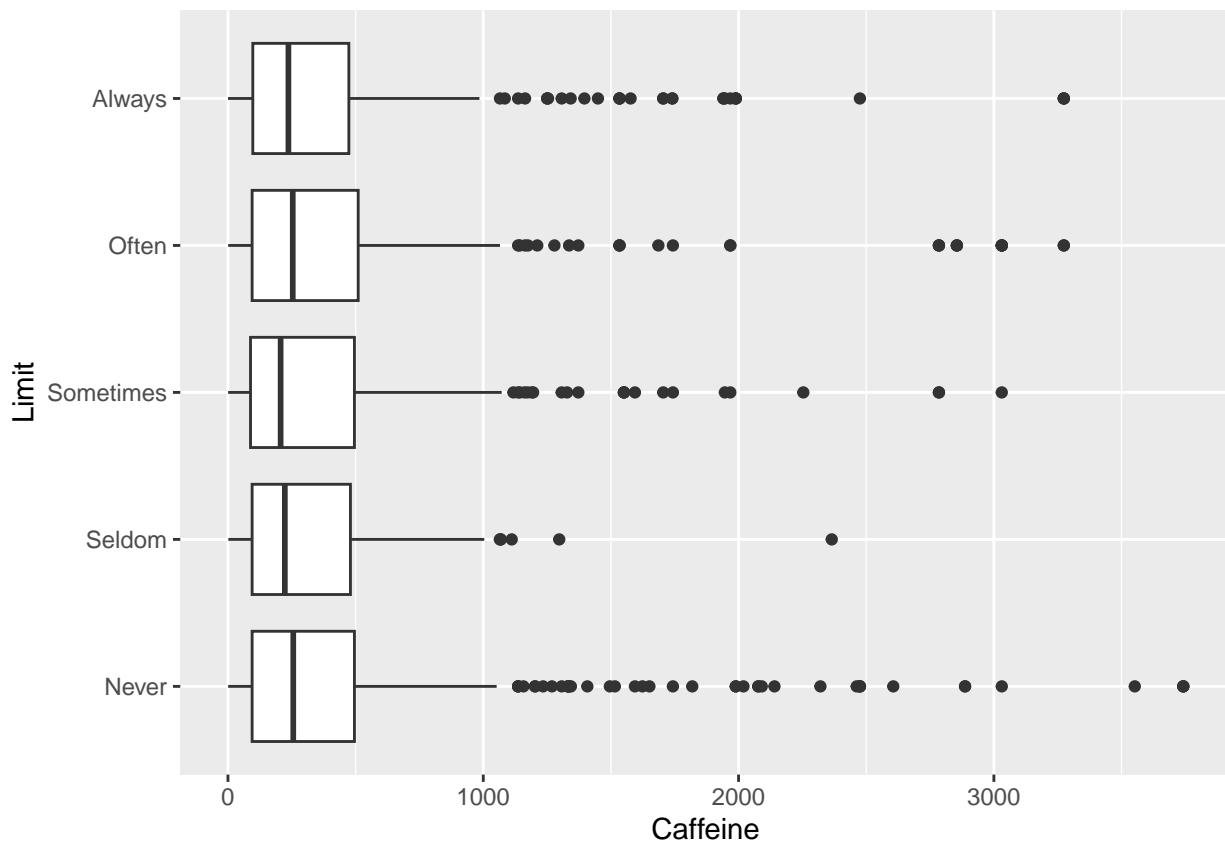
```
# Education
ggplot(data = reduced_data, aes(x = edu, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Education", y = "Percentage", fill = "Limit") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



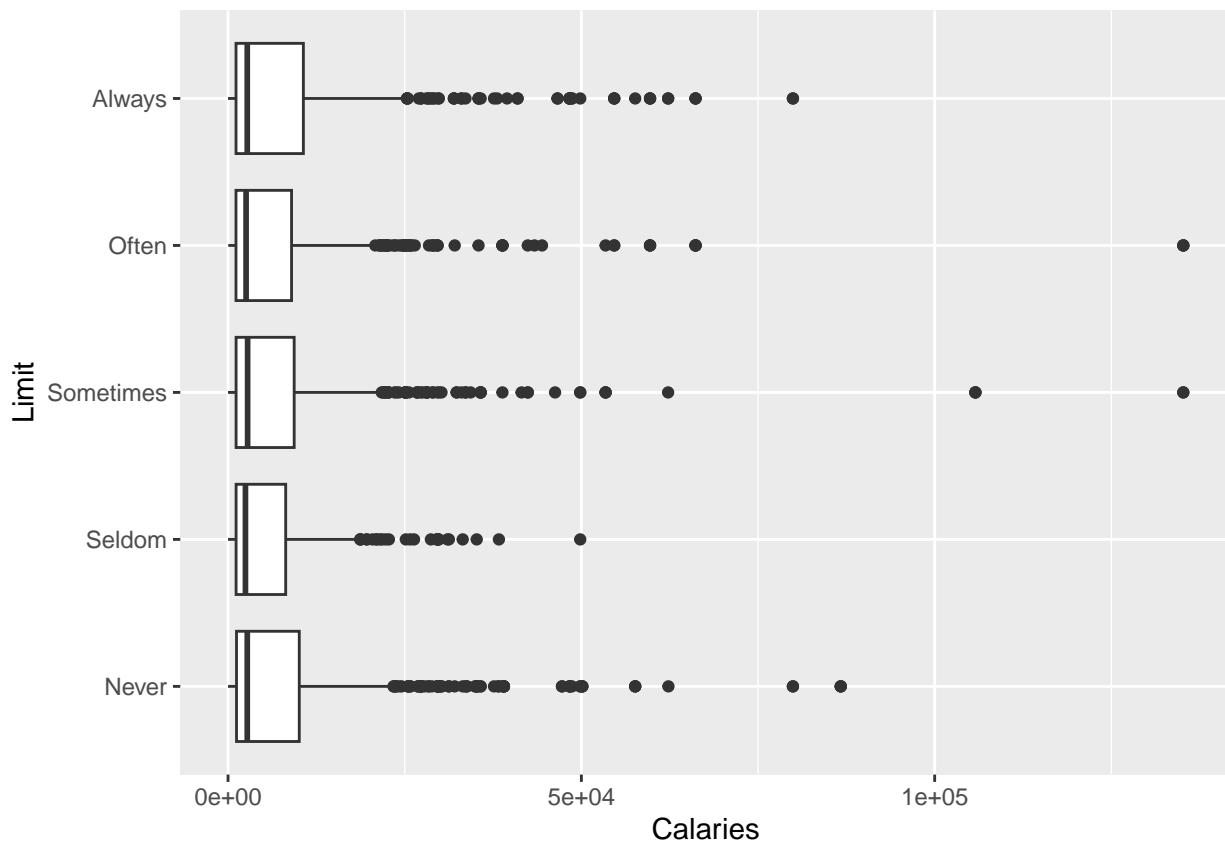
```
# City
ggplot(data = reduced_data, aes(x = city, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "City", y = "Percentage", fill = "Limit") +
  theme_minimal()
```



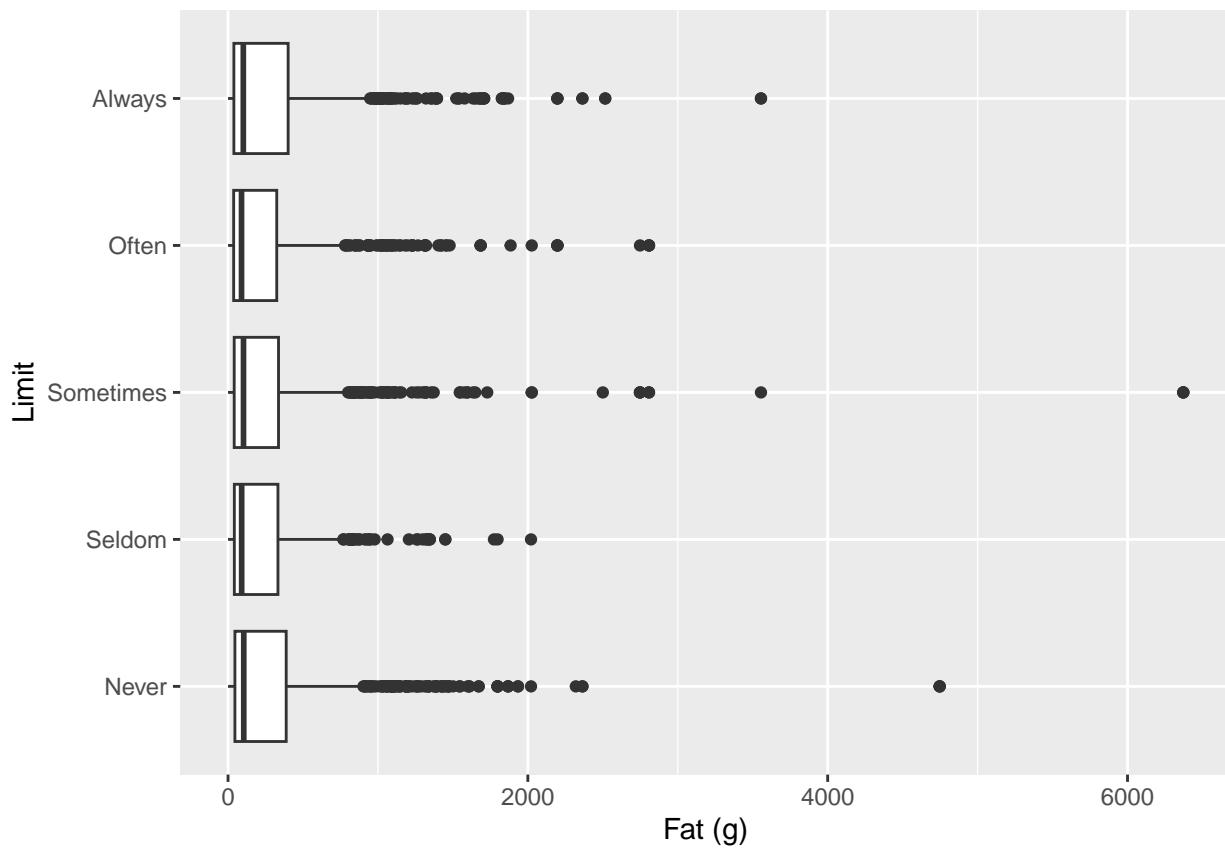
```
# Caffeine
ggplot(data = reduced_data %>% filter(caff > 0), aes(x = caff , y = limit)) +
  geom_boxplot() +
  labs(x = "Caffeine", y = "Limit")
```



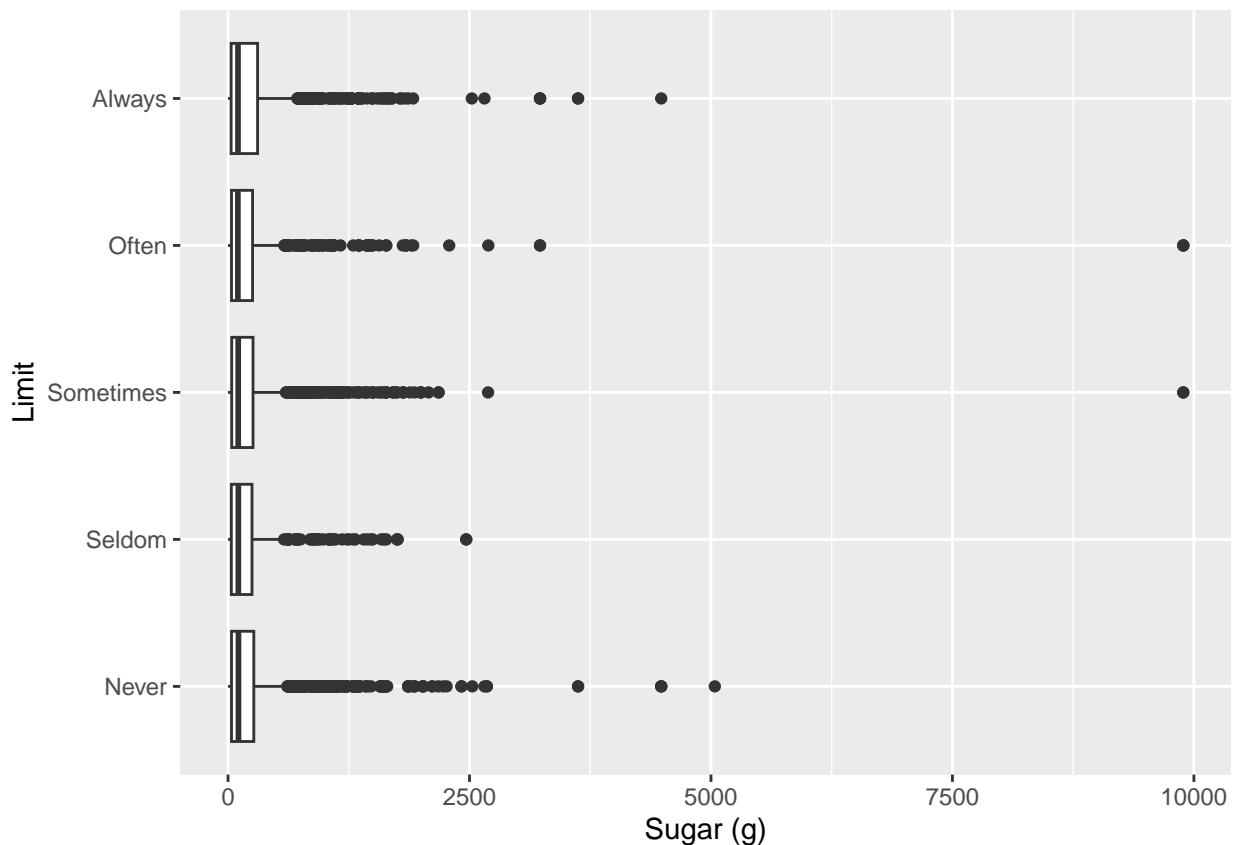
```
# Kcal
ggplot(data = reduced_data, aes(x = caloriescal, y = limit)) +
  geom_boxplot() +
  labs(x = "Calories", y = "Limit")
```



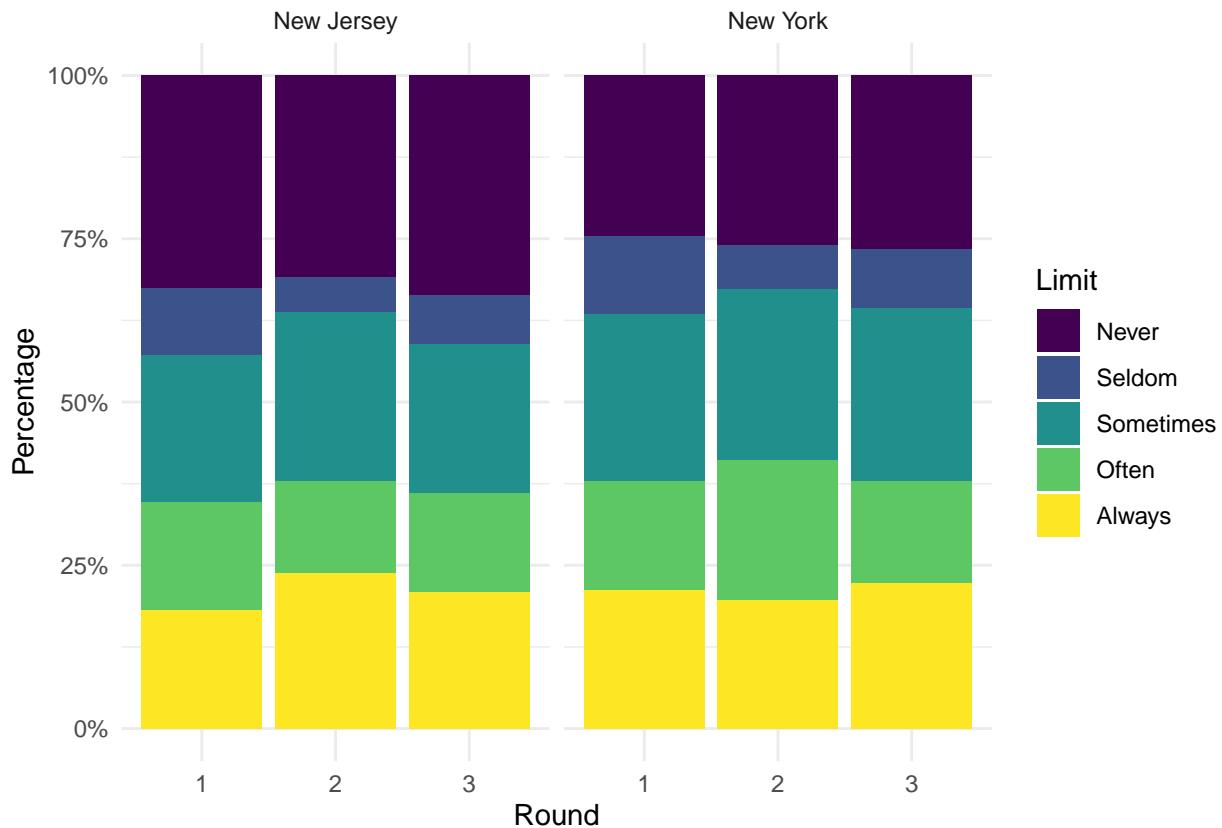
```
# fat
ggplot(data = reduced_data, aes(x = fatg, y = limit)) +
  geom_boxplot() +
  labs(x = "Fat (g)", y = "Limit")
```



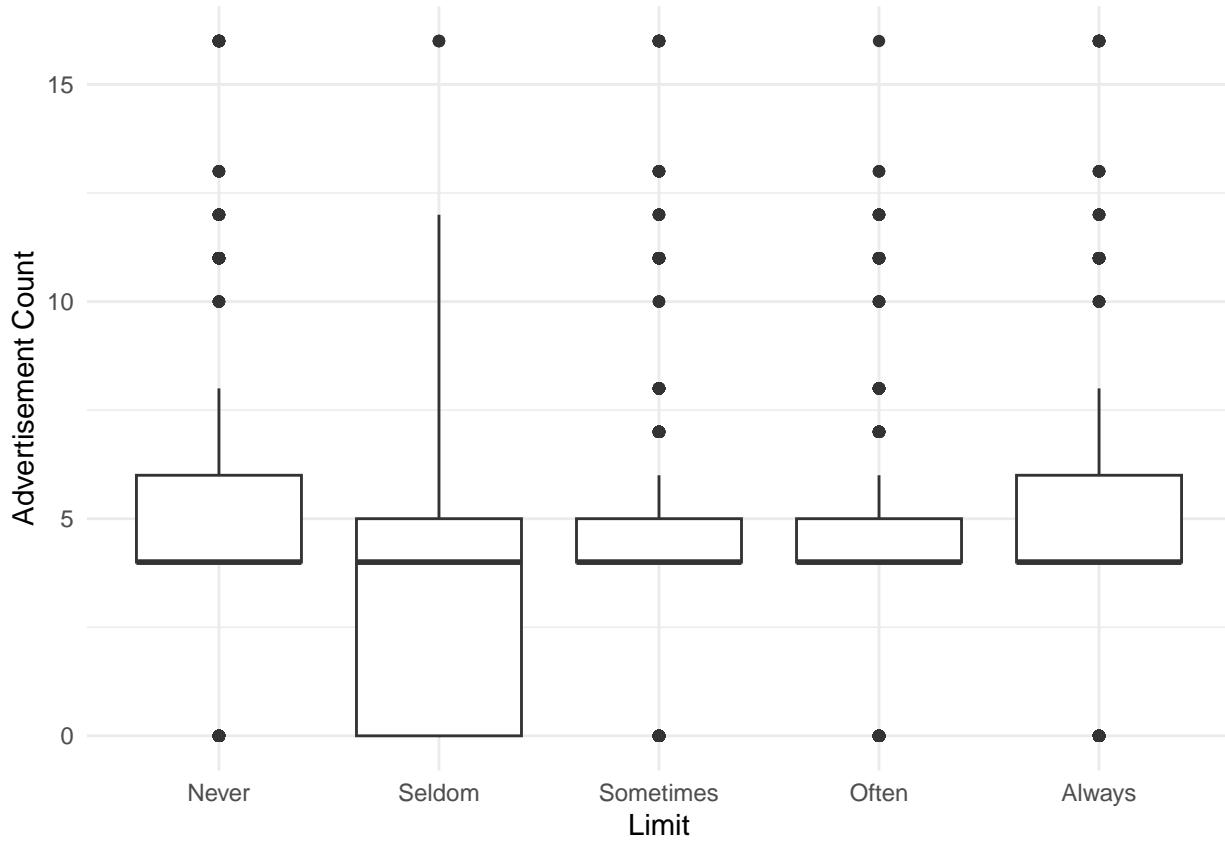
```
# sugar
ggplot(data = reduced_data, aes(x = sugarg, y = limit)) +
  geom_boxplot() +
  labs(x = "Sugar (g)", y = "Limit")
```



```
# Survey round
ggplot(data = reduced_data, aes(x = round, fill = limit)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Round", y = "Percentage", fill = "Limit") +
  facet_wrap(~city) +
  theme_minimal()
```



```
# Advertisement count
ggplot(data = reduced_data, aes(y = nsigns_ssbb, x = limit)) +
  geom_boxplot() +
  labs(y = "Advertisement Count", x = "Limit") +
  theme_minimal()
```



```
# Plot function for interactions

plot_cats <- c("limit", "gender", "race", "city", "round", "num_kids", "edu")
plot_nums <- c("age", "caff", "nsigns(ssb)", "days_since_ban", "caloriescal", "sugarg", "fatg")

library(rlang)

make_plot <- function(var1, var2){
  if(var1 %in% plot_cats & var2 %in% plot_cats){
    print(ret_plot <- ggplot(data = reduced_data, aes(x = !!sym(var1), fill = !!sym(var2))) +
      geom_bar(position = "fill") +
      scale_y_continuous(labels = scales::percent) +
      theme_minimal())
  }

  if(var1 %in% plot_cats & var2 %in% plot_nums){
    print(ret_plot <- ggplot(data = reduced_data, aes(x = !!sym(var1), y = !!sym(var2))) +
      geom_boxplot() +
      theme_minimal())
  }

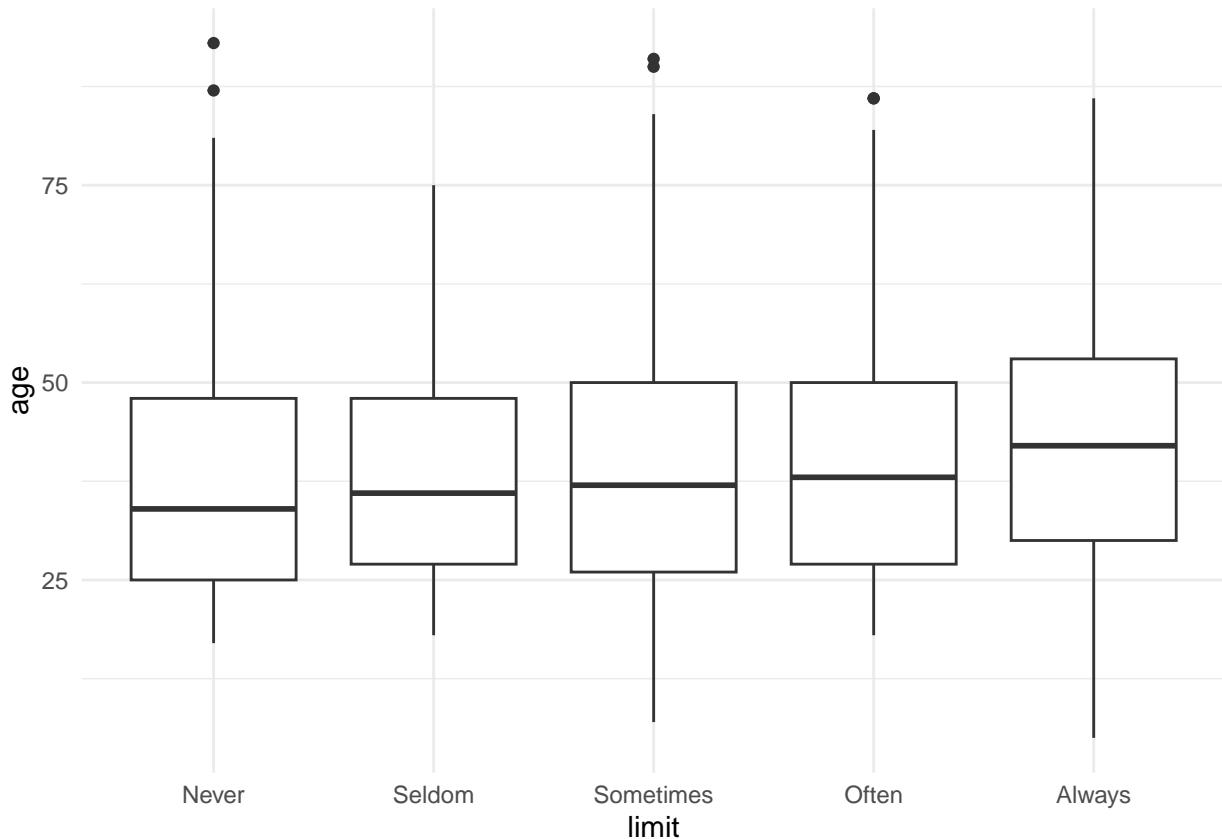
  if(var1 %in% plot_nums & var2 %in% plot_cats){
    print(ret_plot <- ggplot(data = reduced_data, aes(x = !!sym(var2), y = !!sym(var1))) +
      geom_boxplot() +
      theme_minimal())
  }
}
```

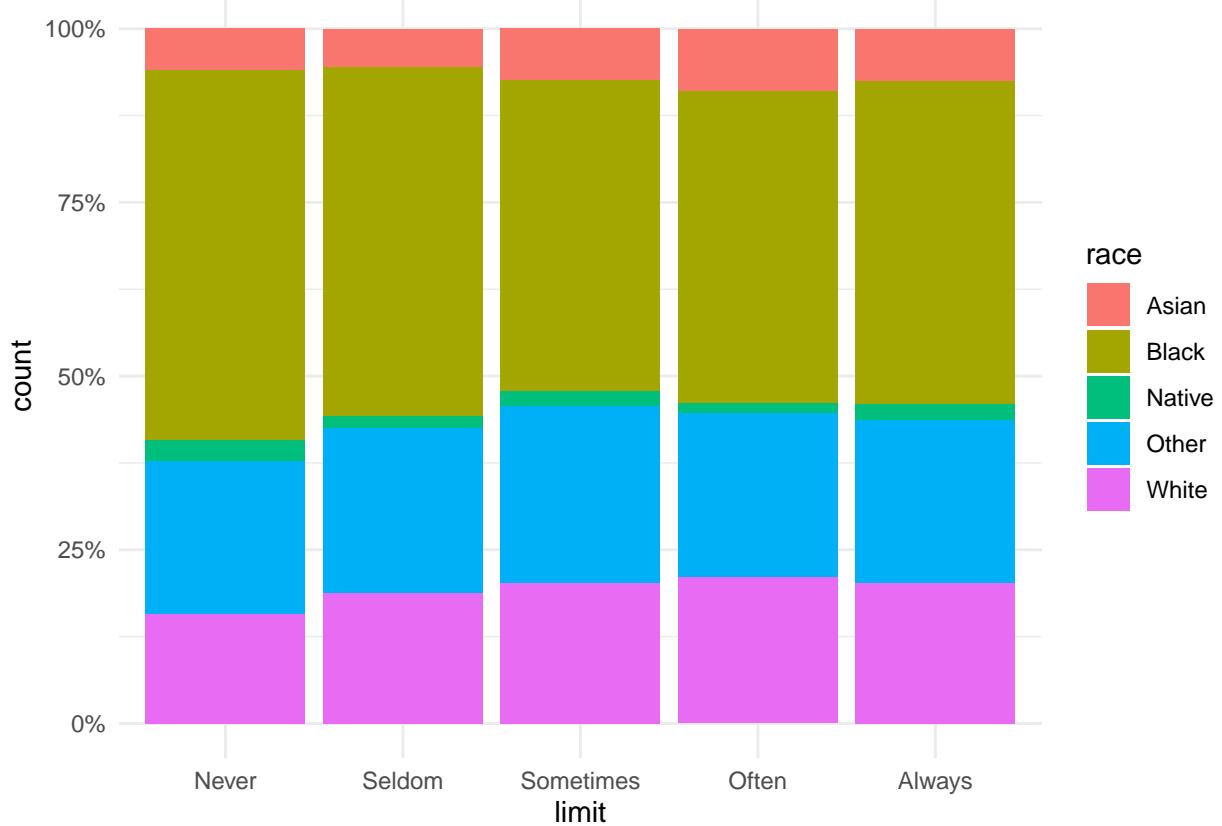
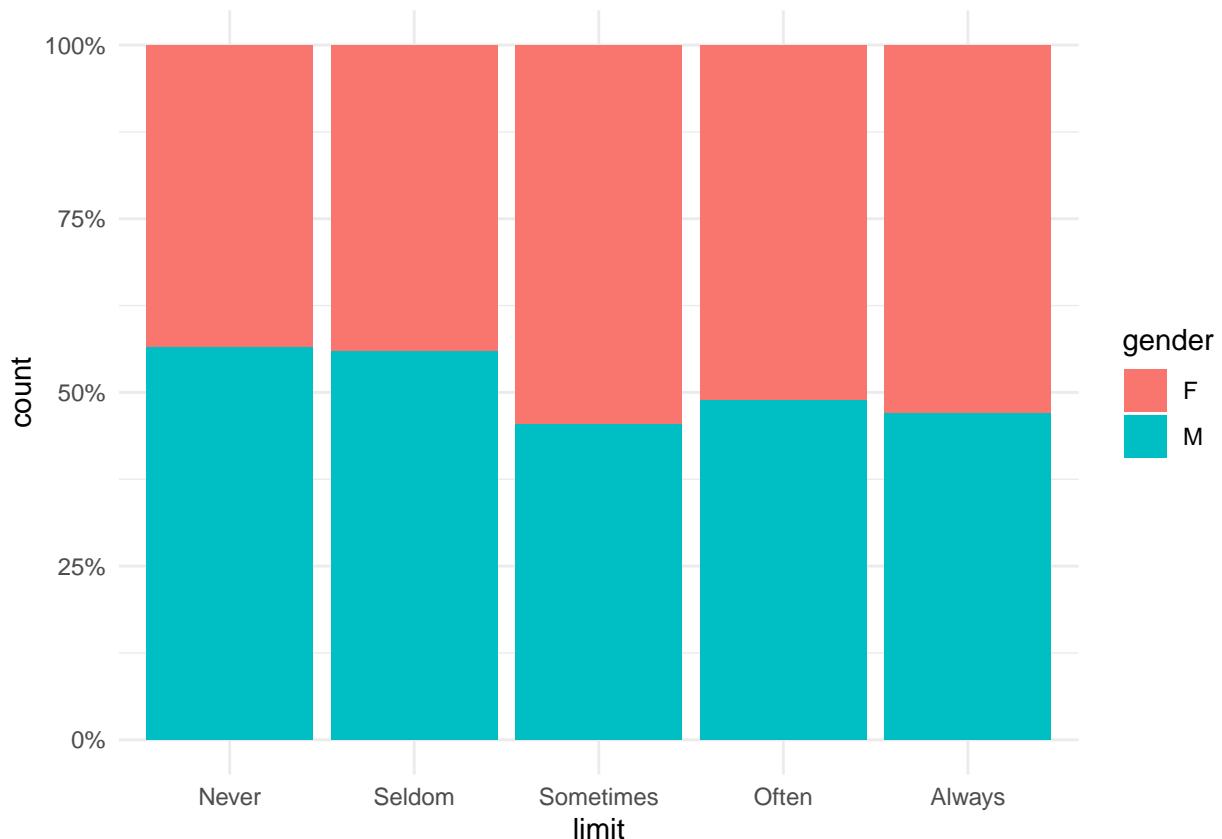
```

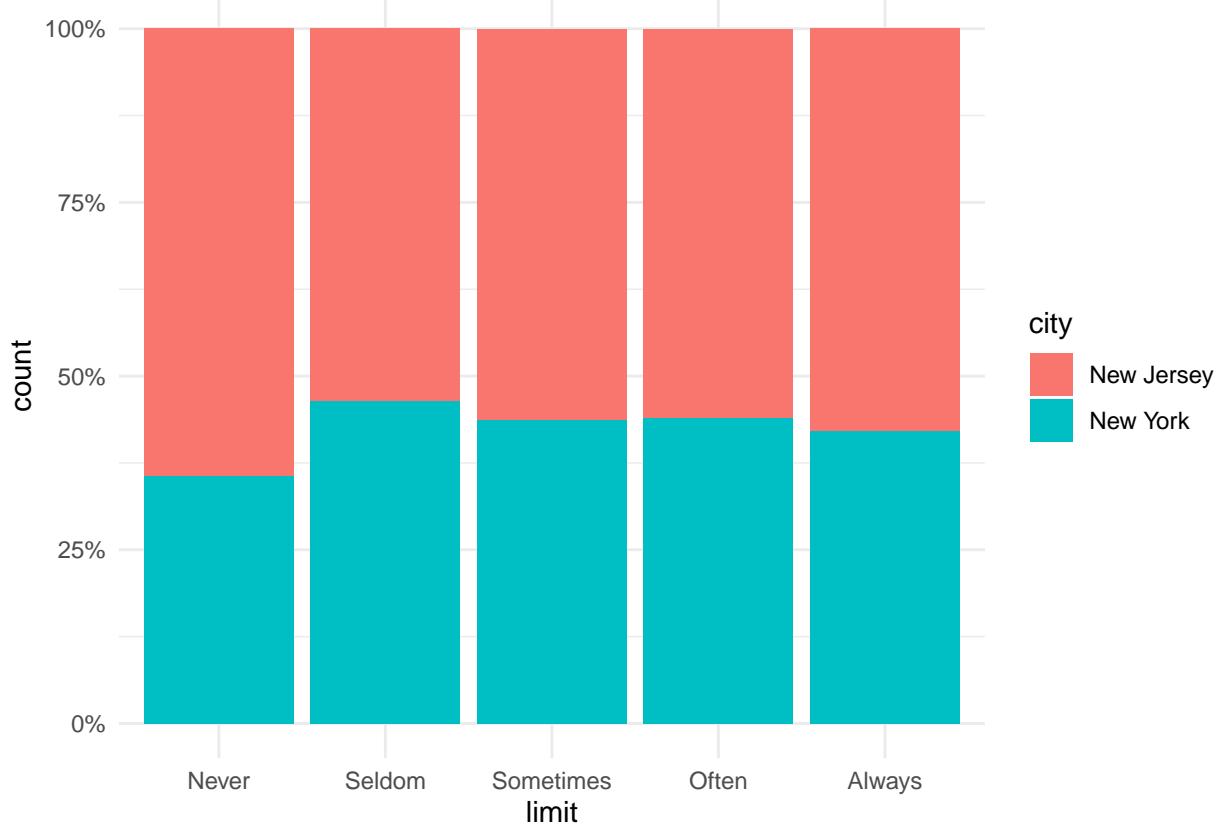
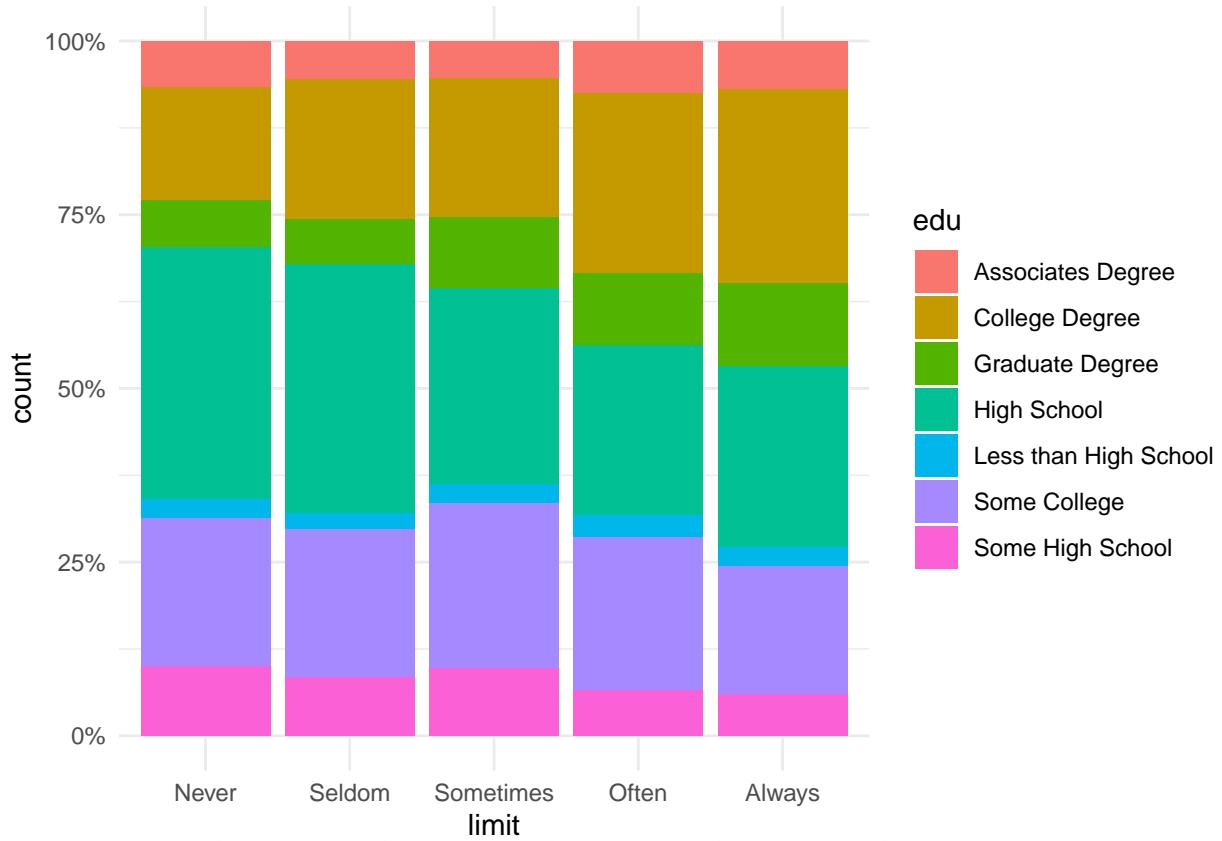
if(var1 %in% plot_nums & var2 %in% plot_nums){
  print(ggplot(data = reduced_data, aes(x = !!sym(var2), y = !!sym(var1))) +
    geom_point() +
    theme_minimal())
}

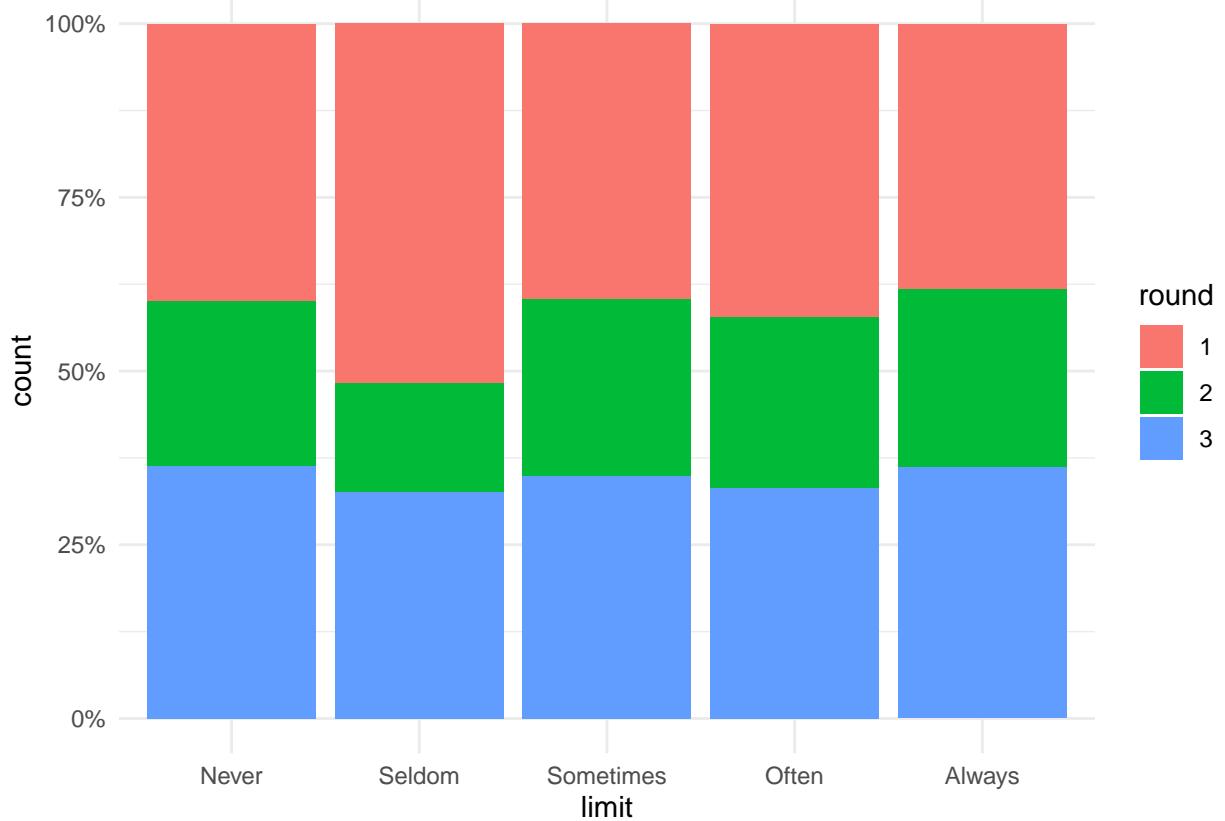
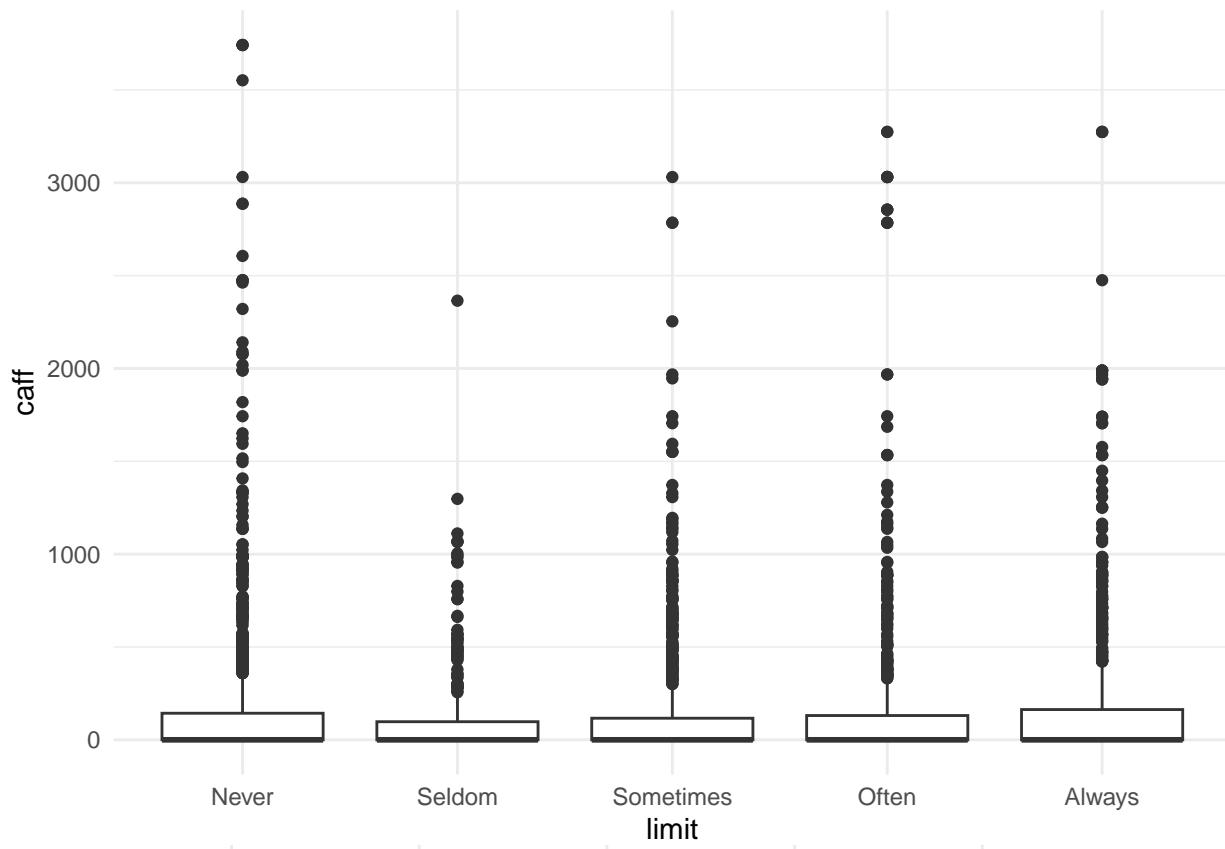
for(i in 1:length(names(reduced_data))){
  if(i != length(reduced_data)){
    for(j in (i+1):length(reduced_data)){
      make_plot(names(reduced_data)[i], names(reduced_data)[j])
    }
  }
}

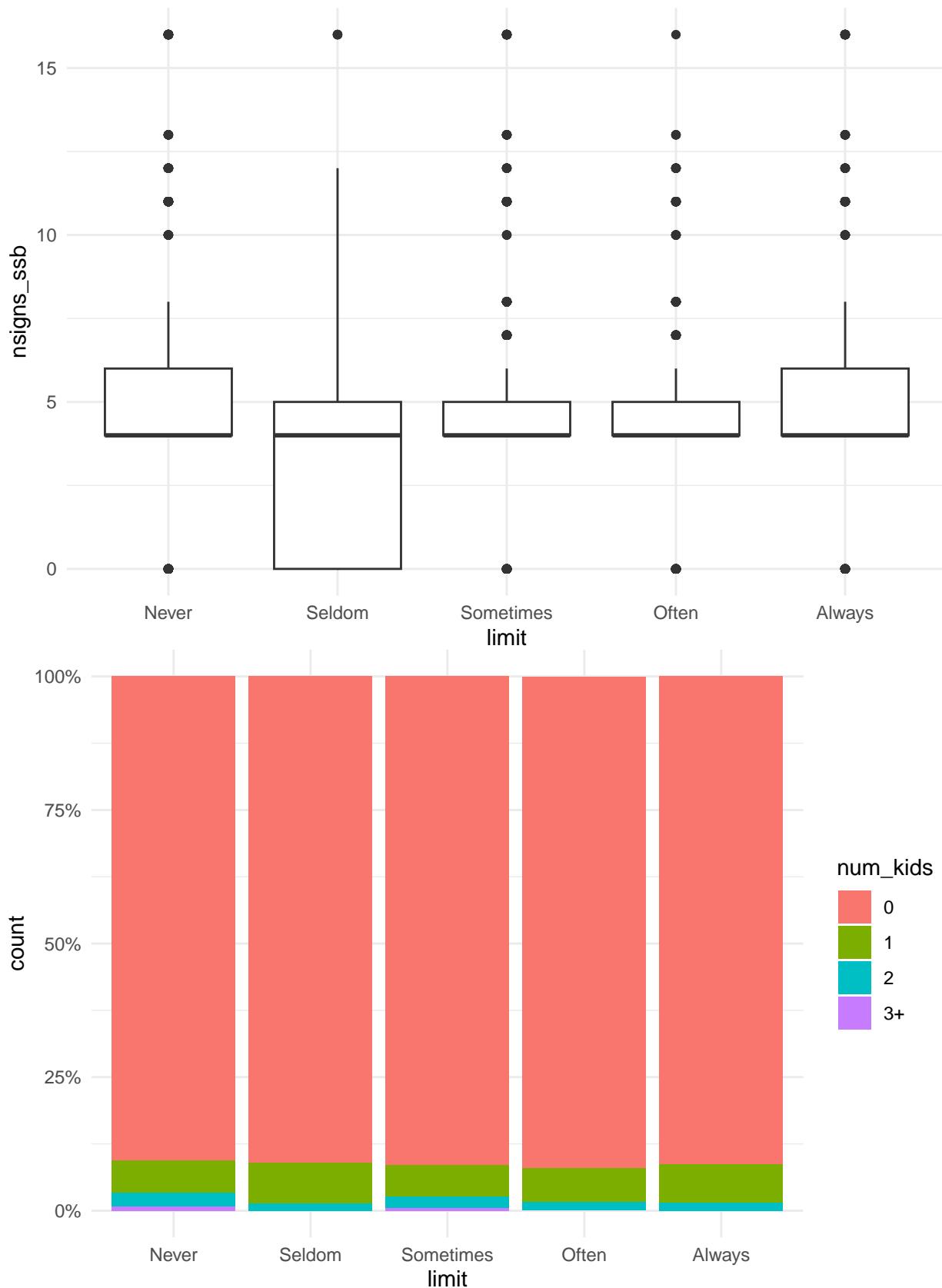
```

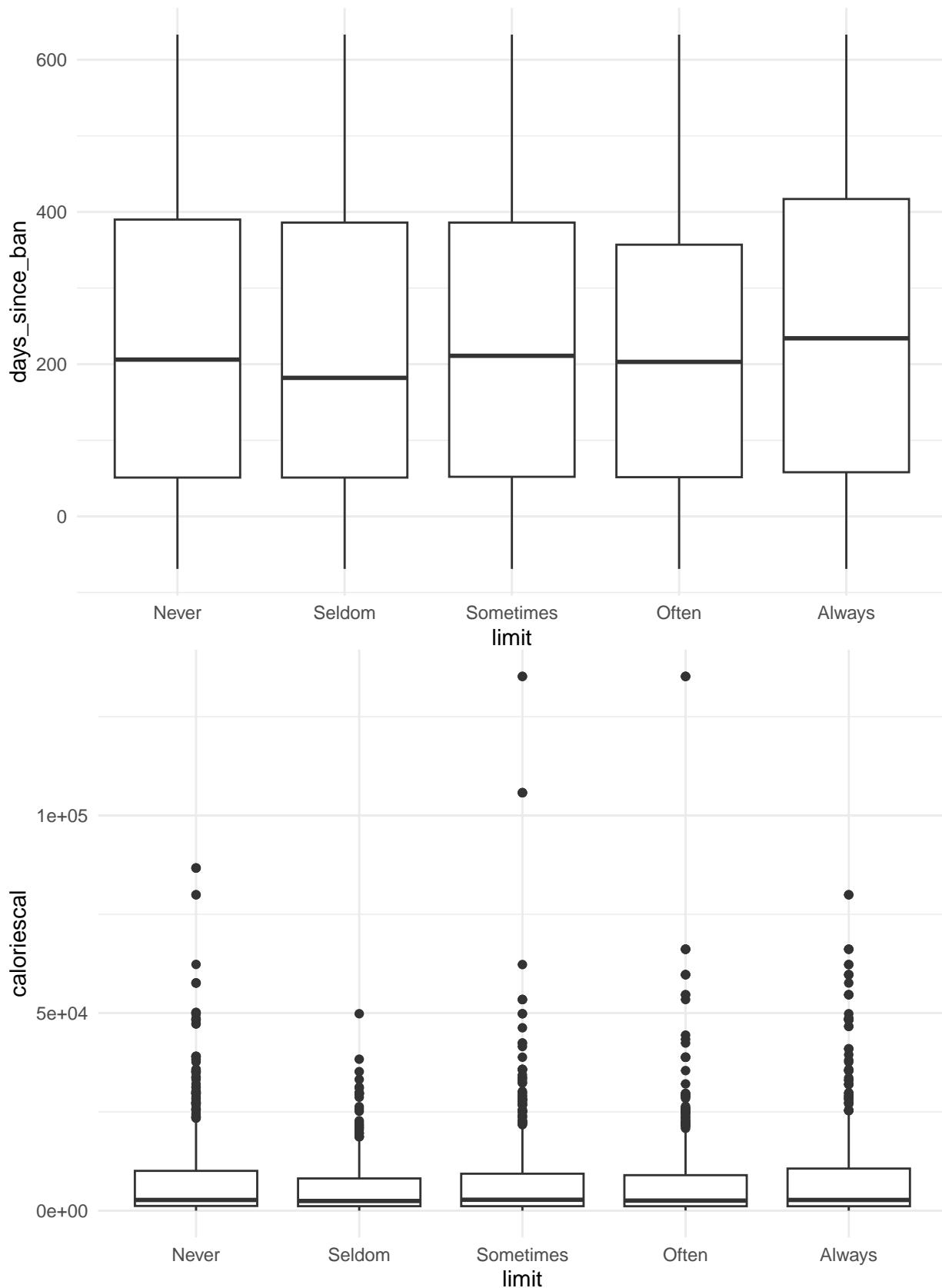


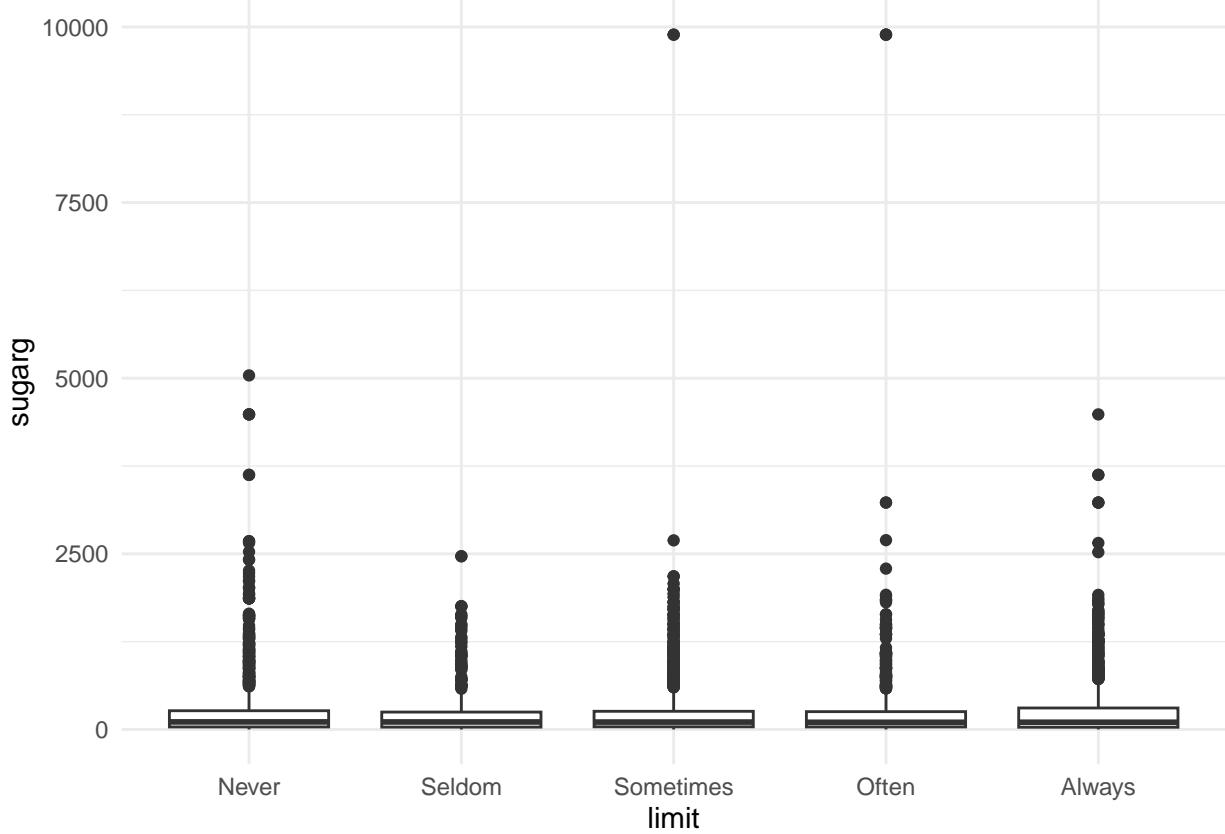
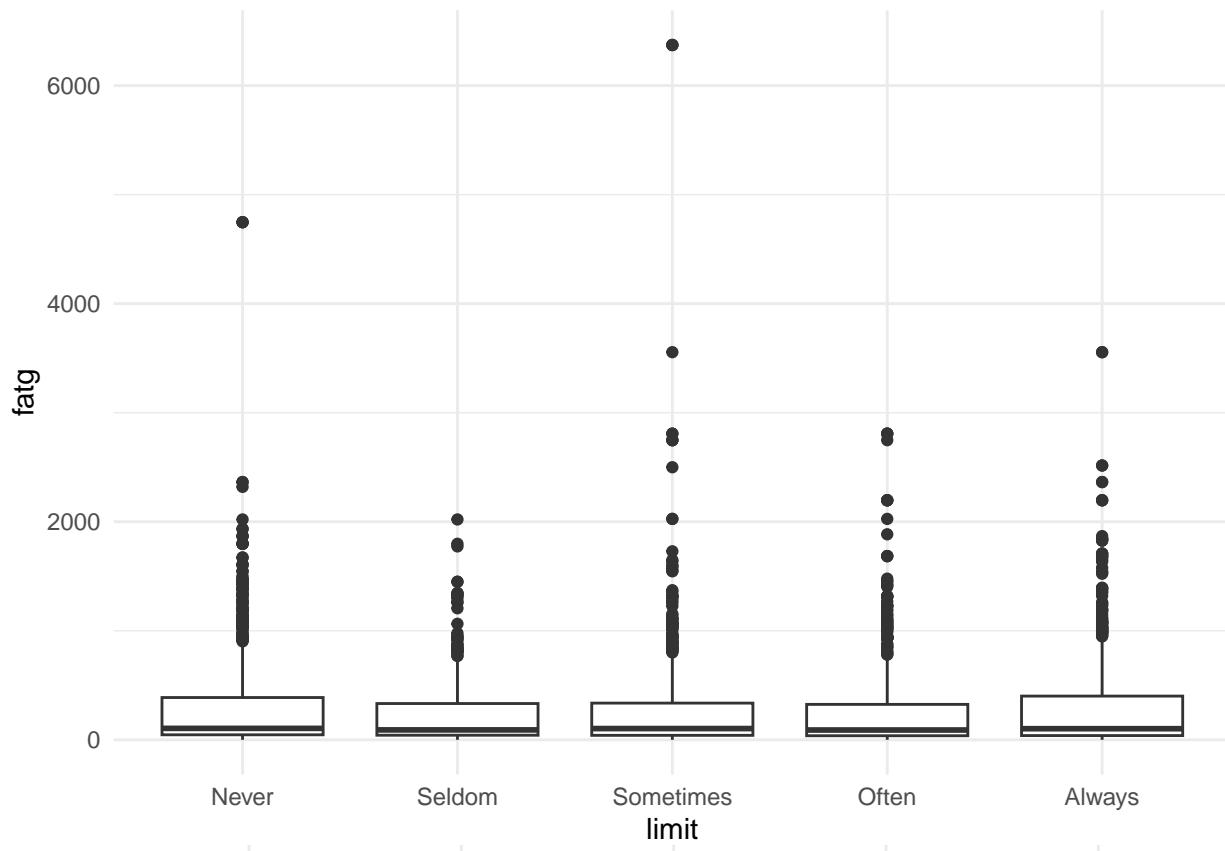


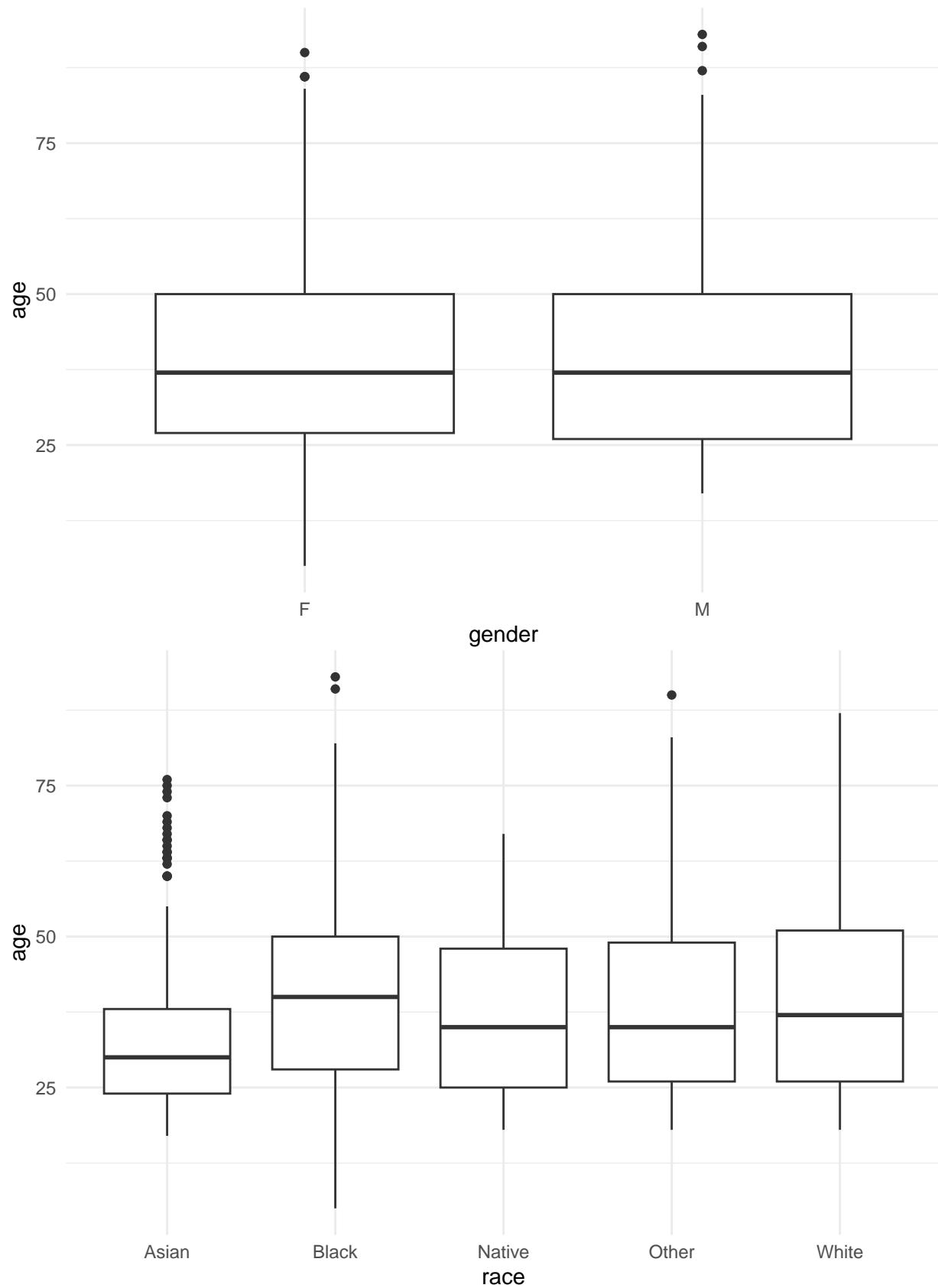


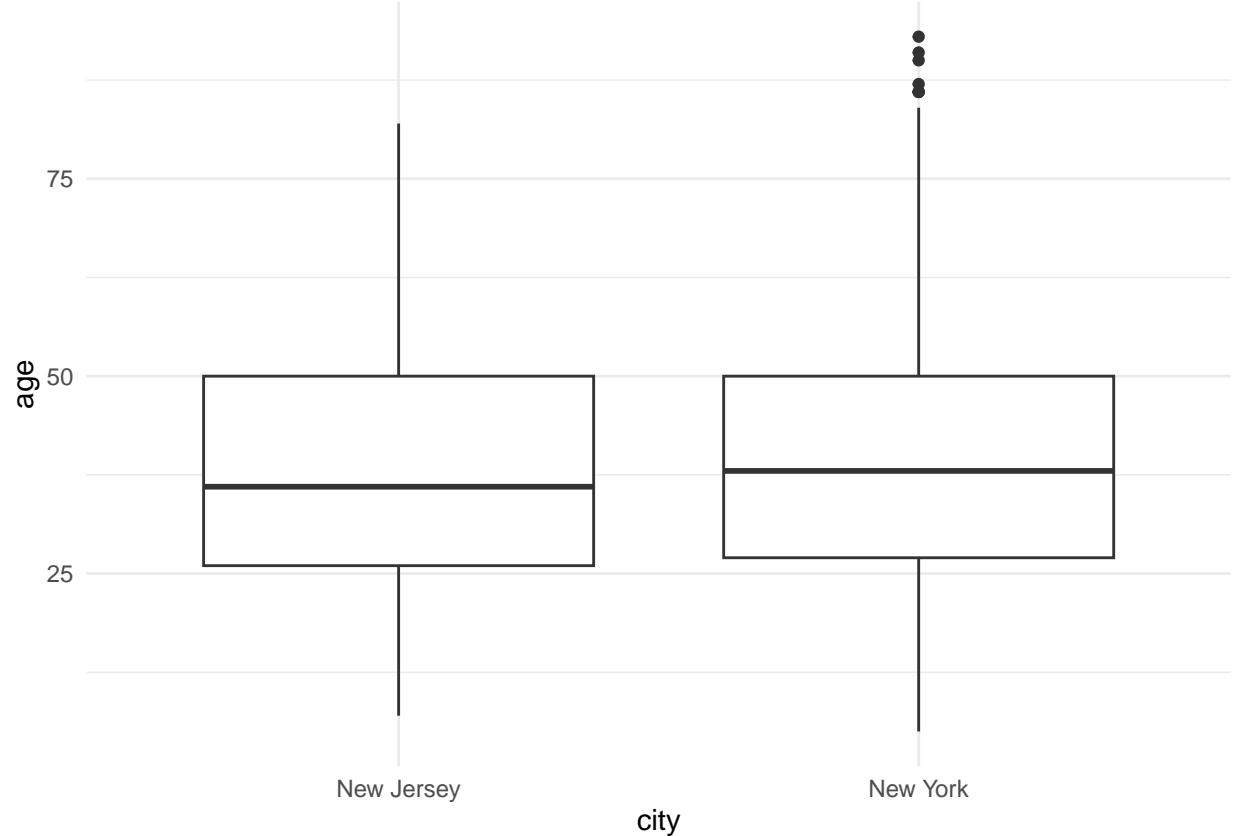
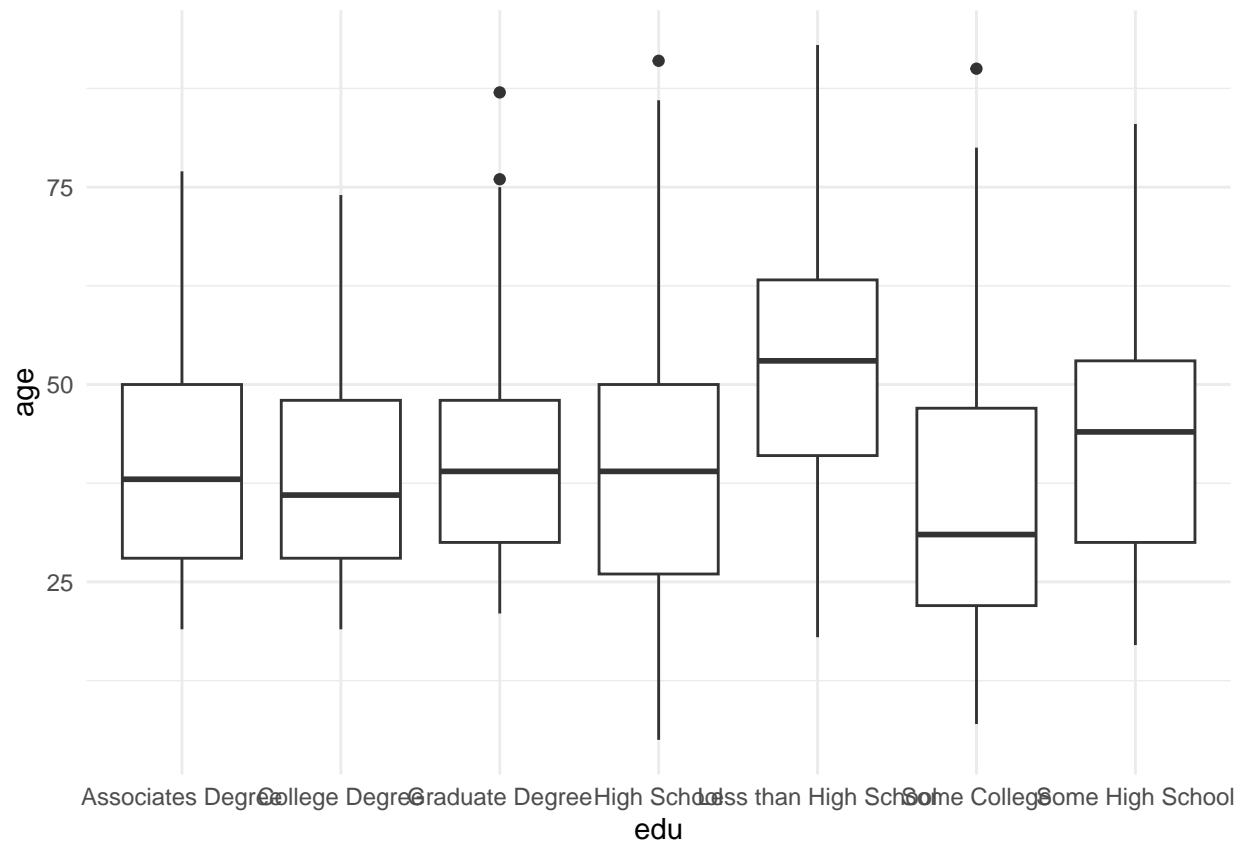


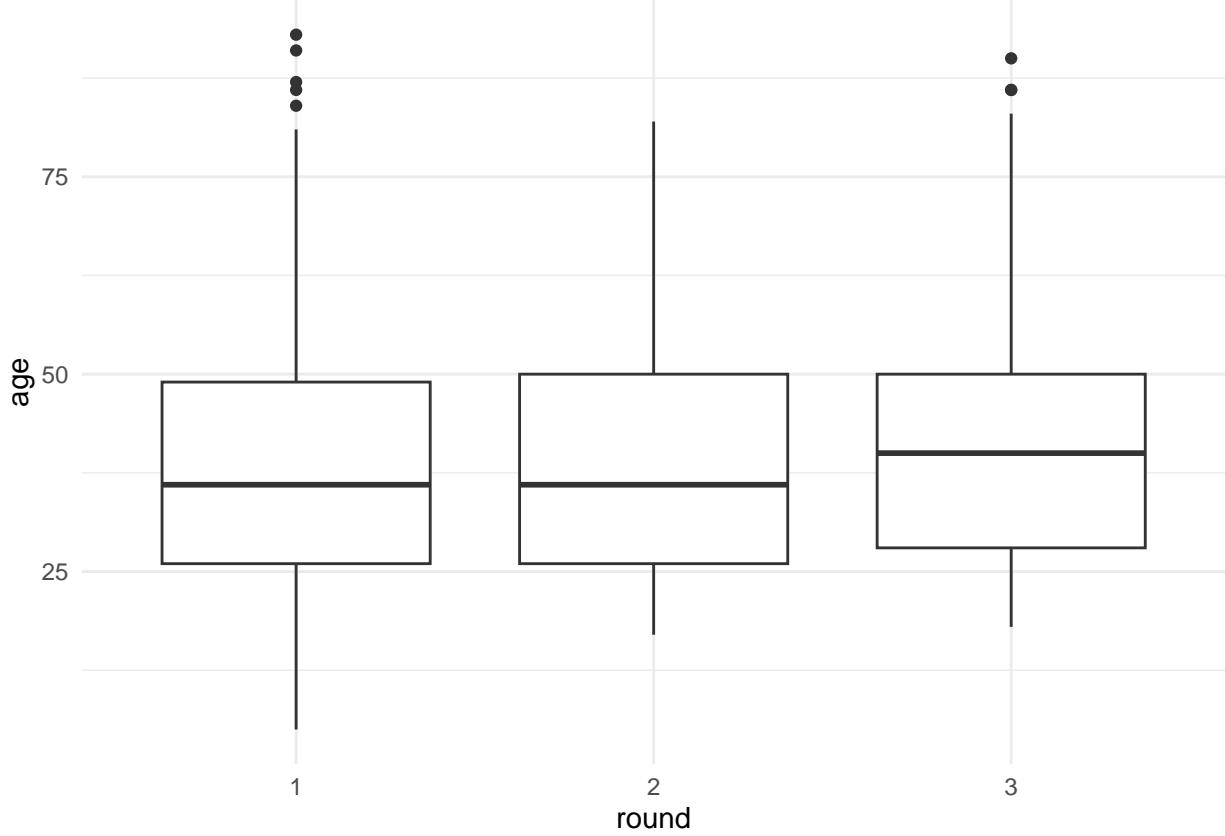
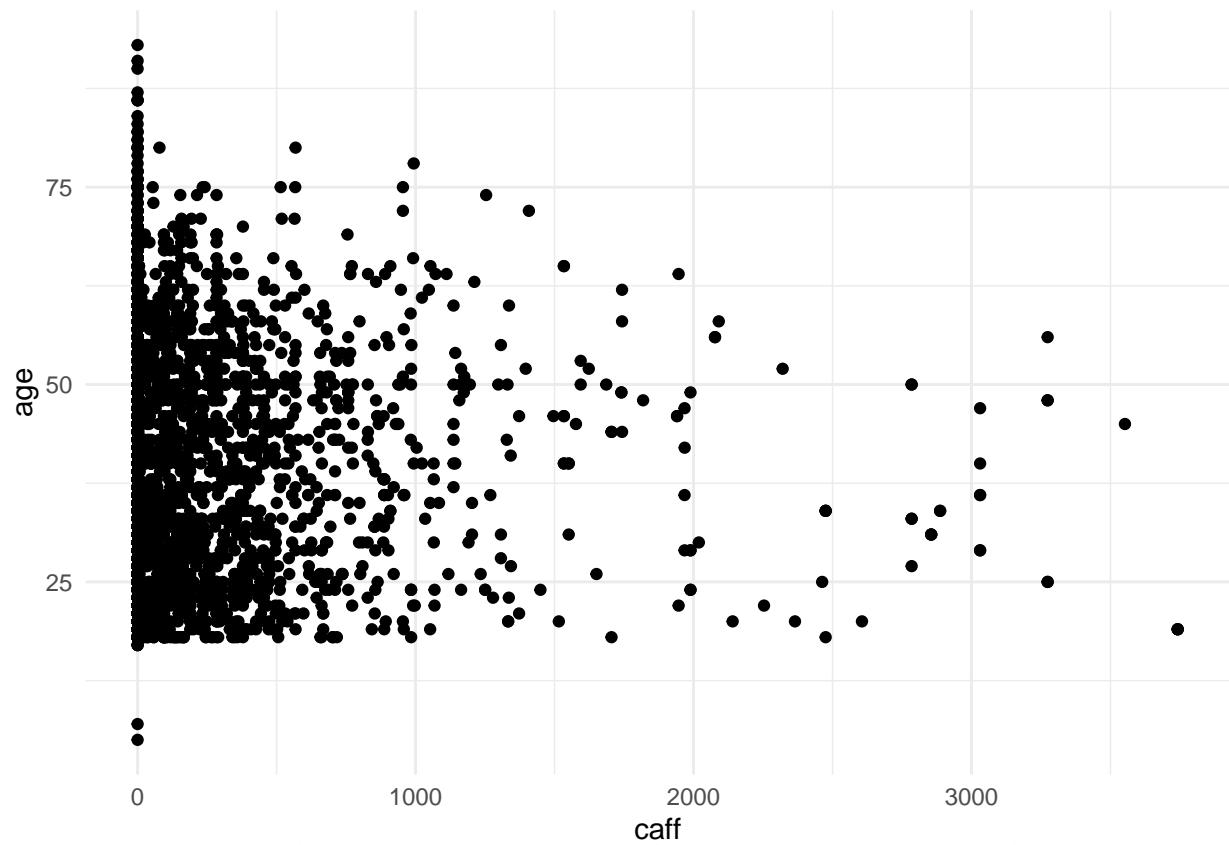


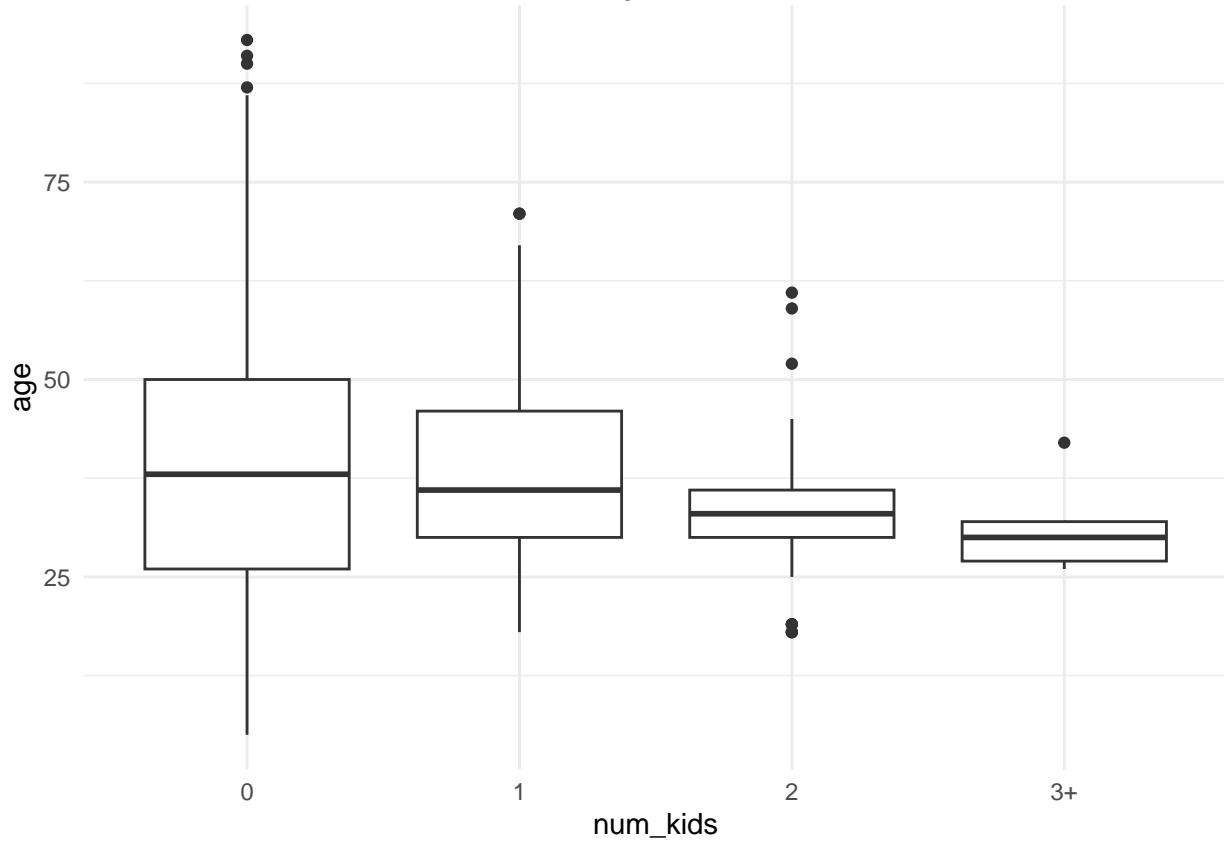
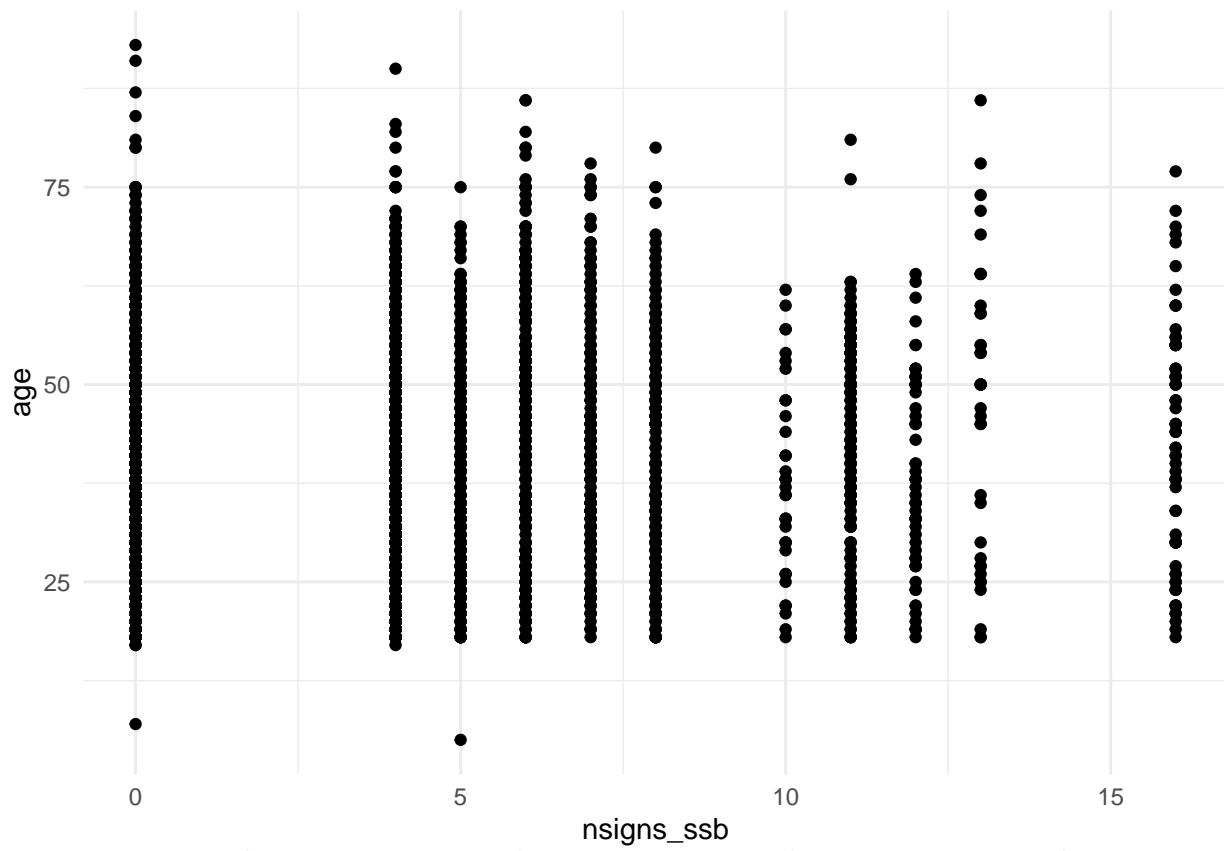


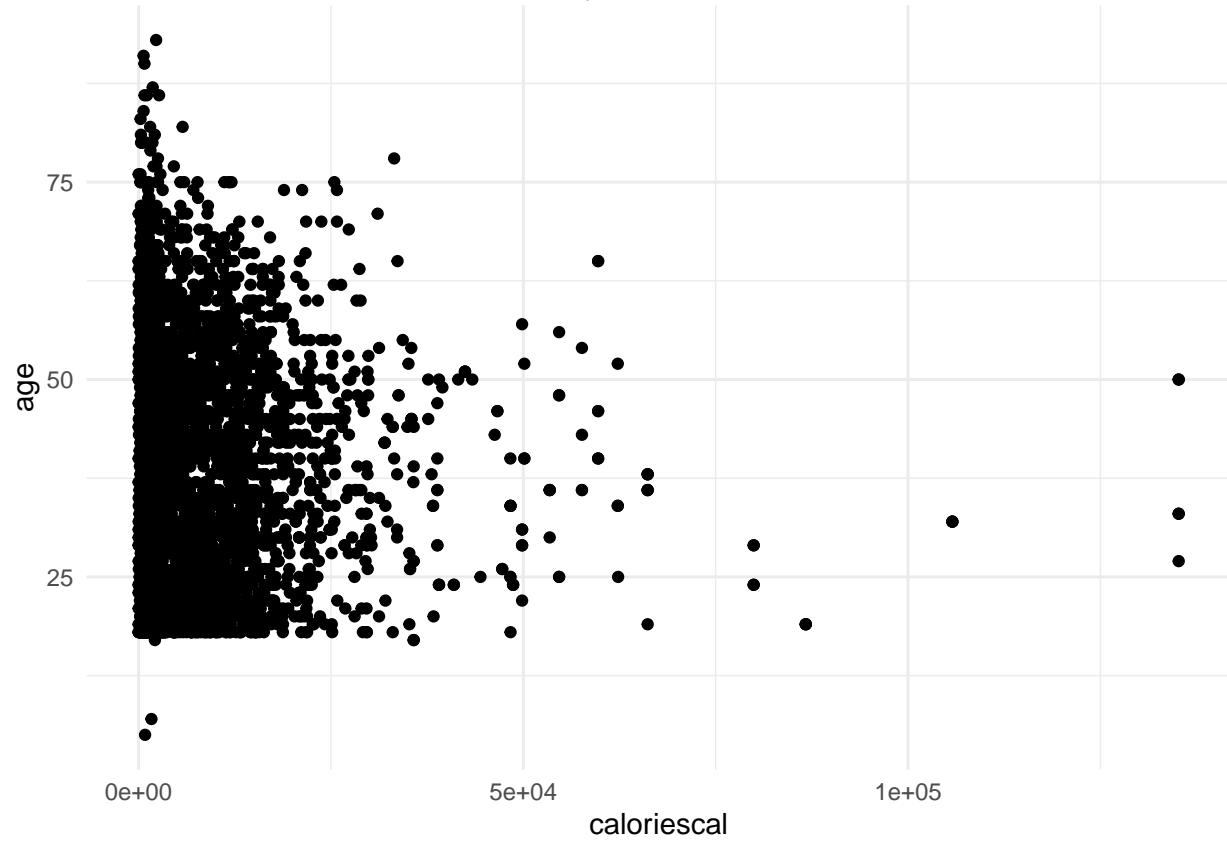
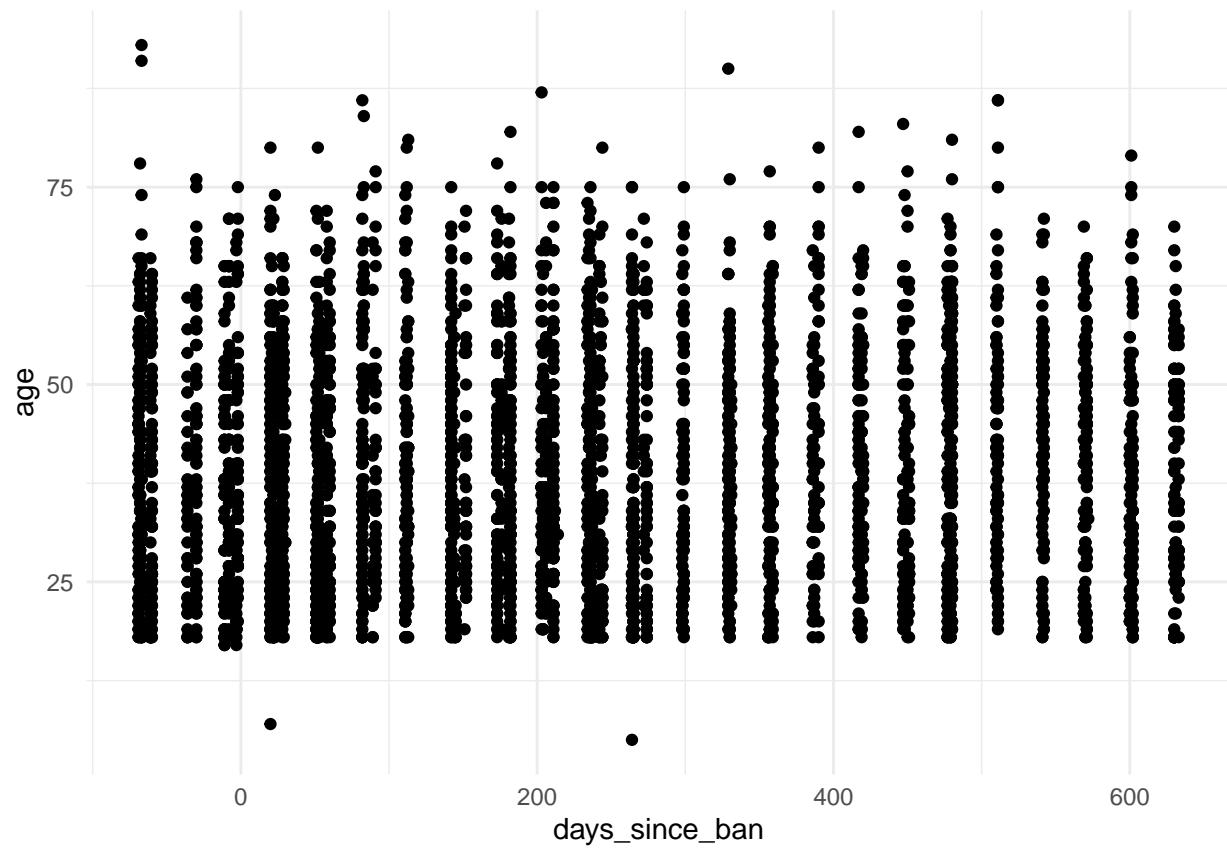


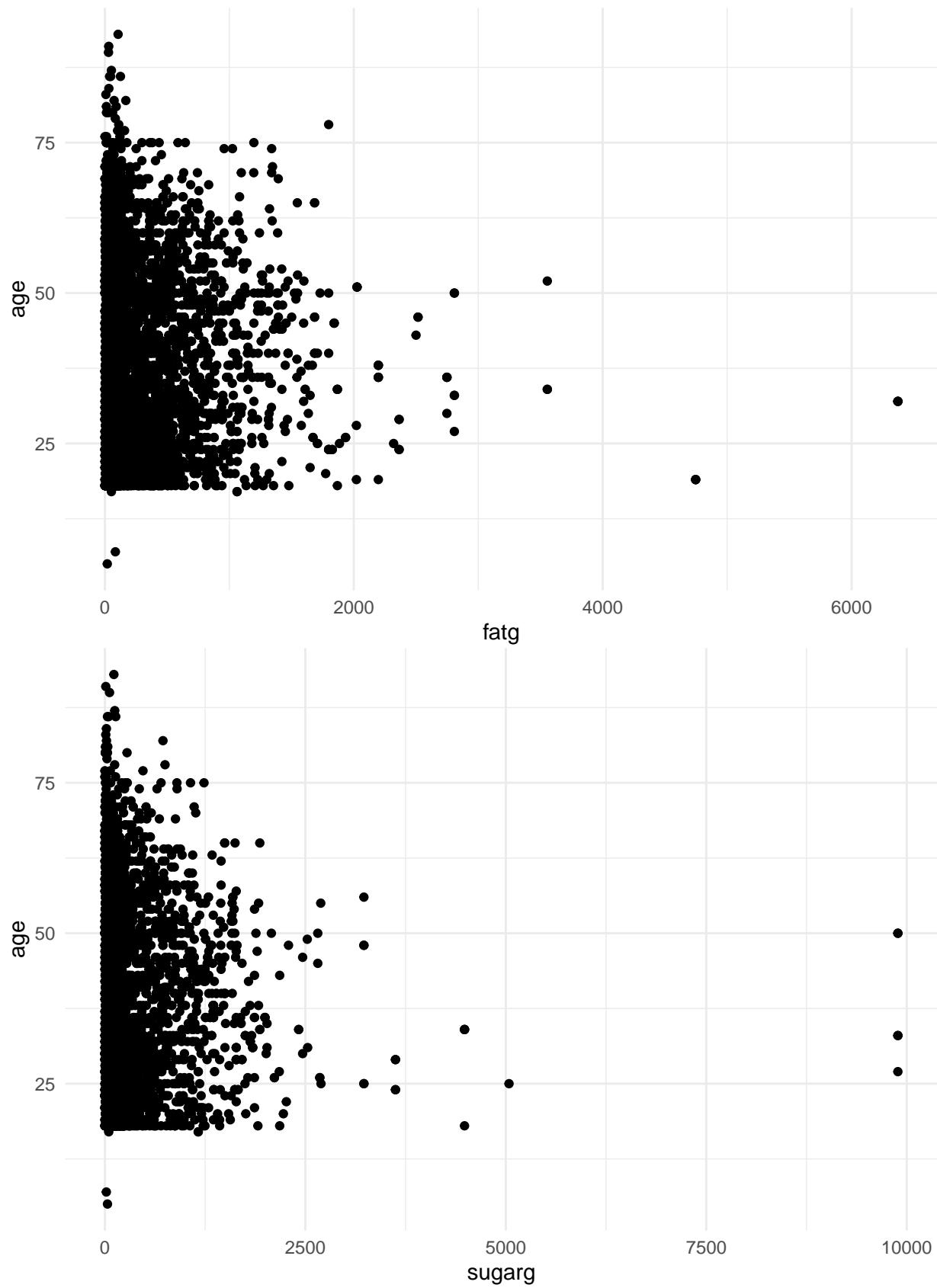


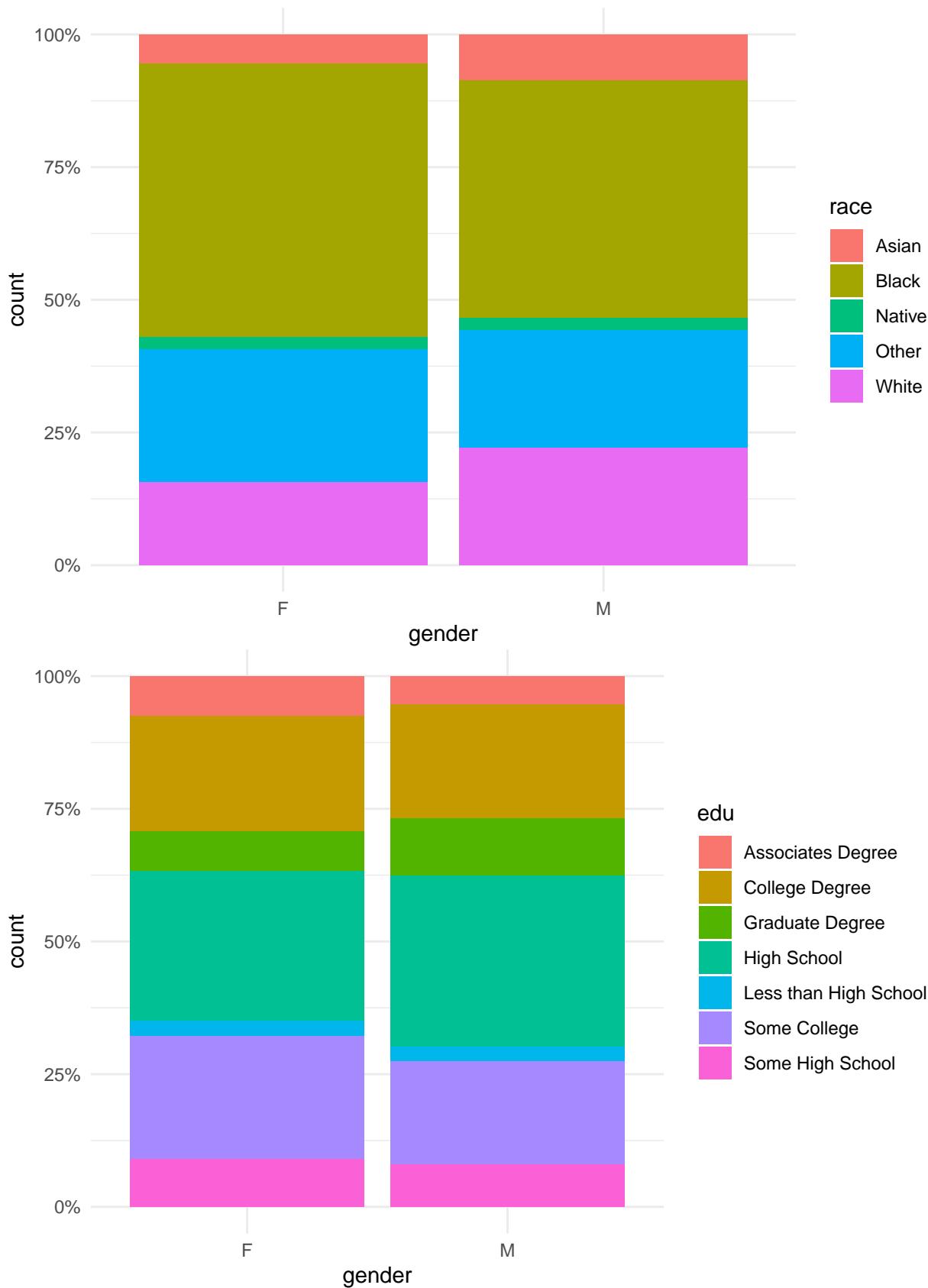


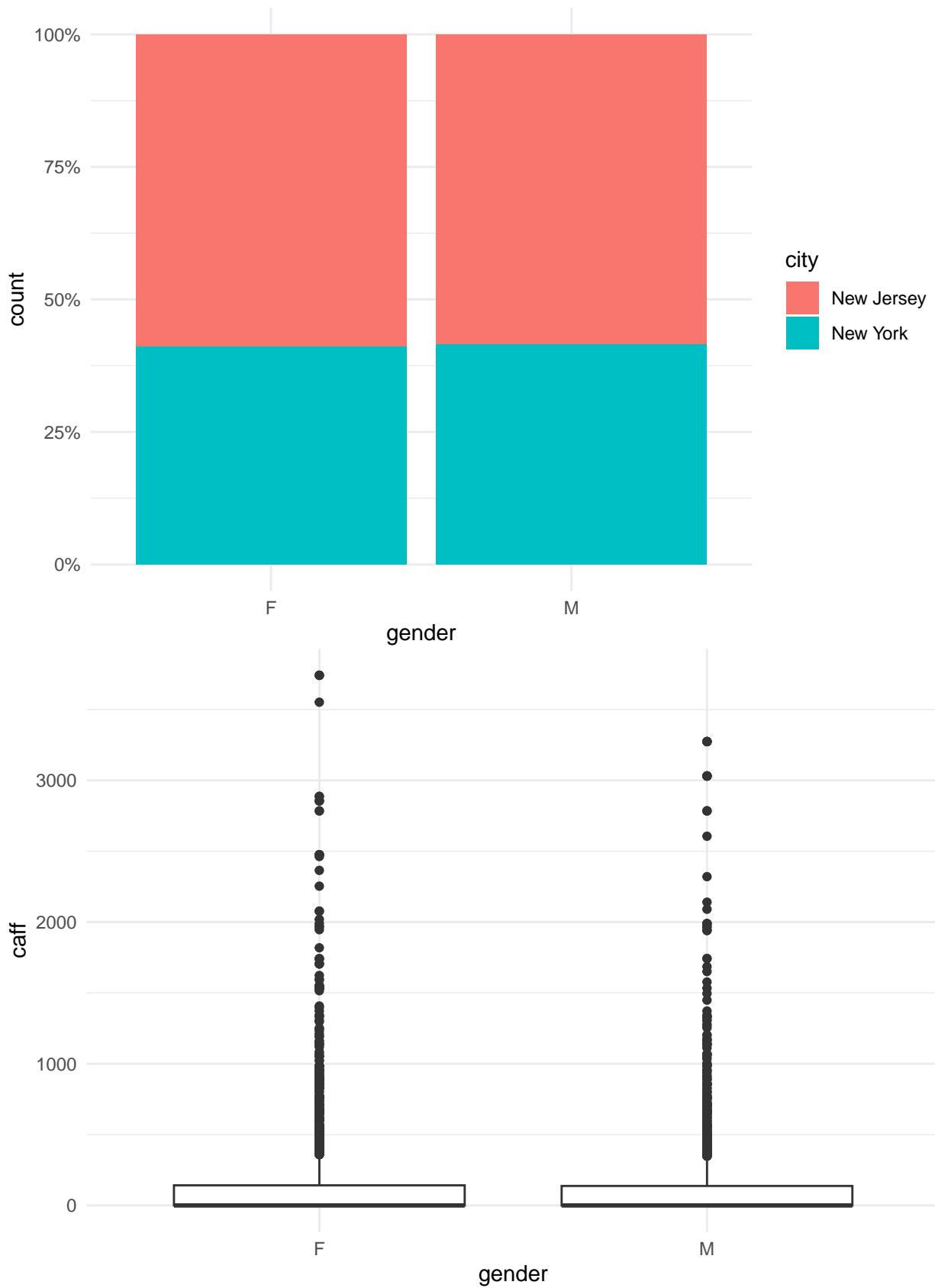


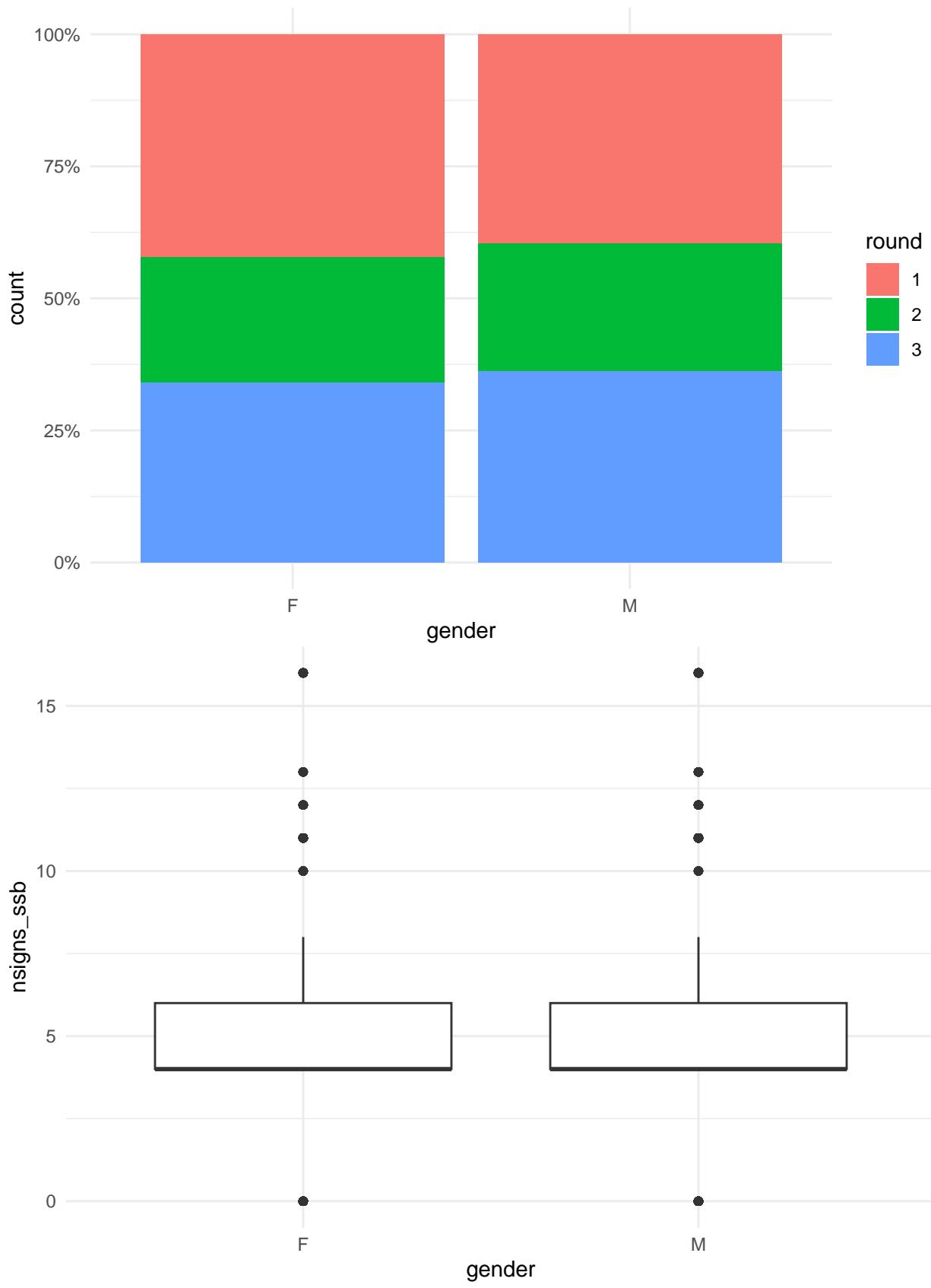


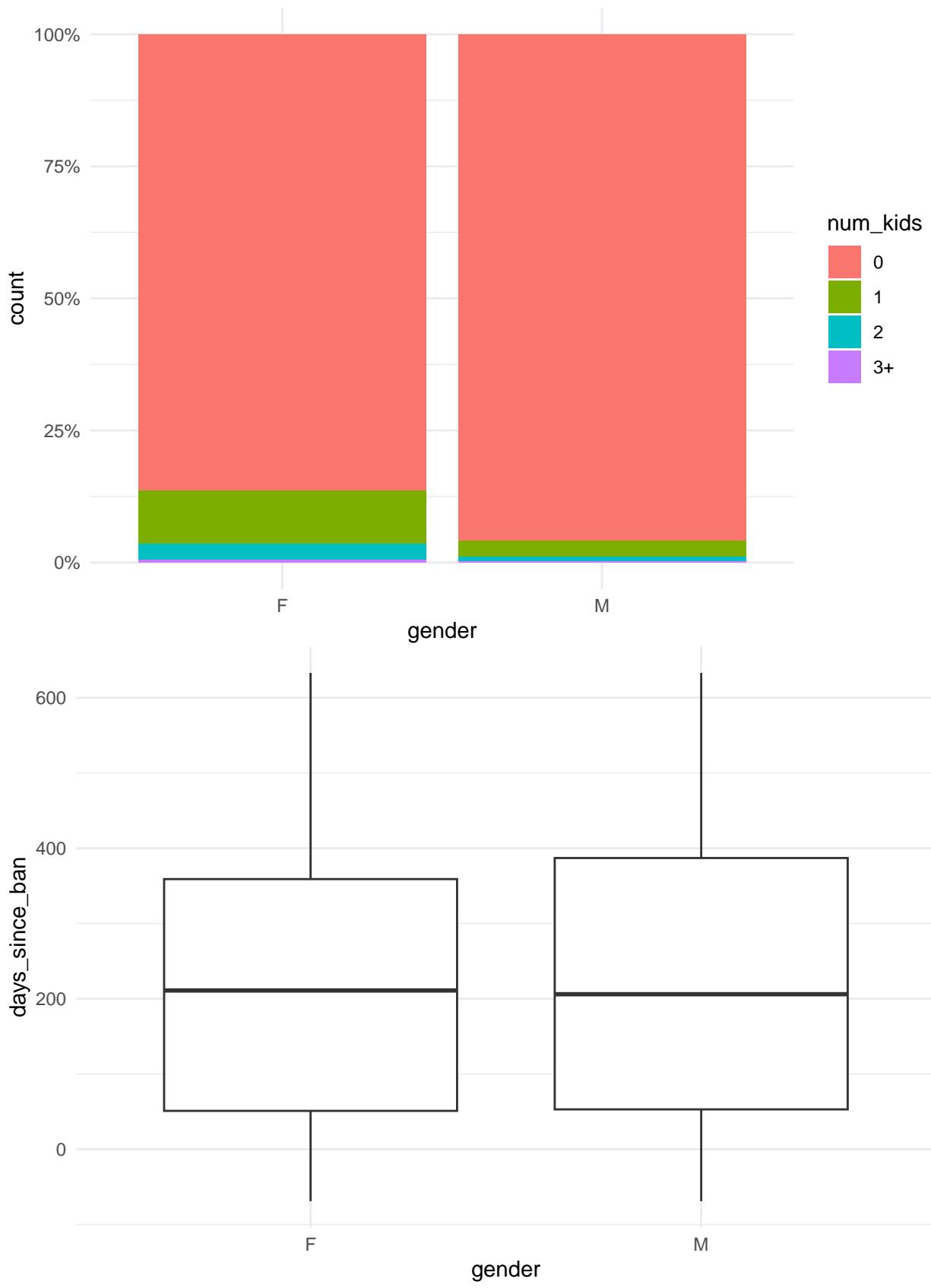


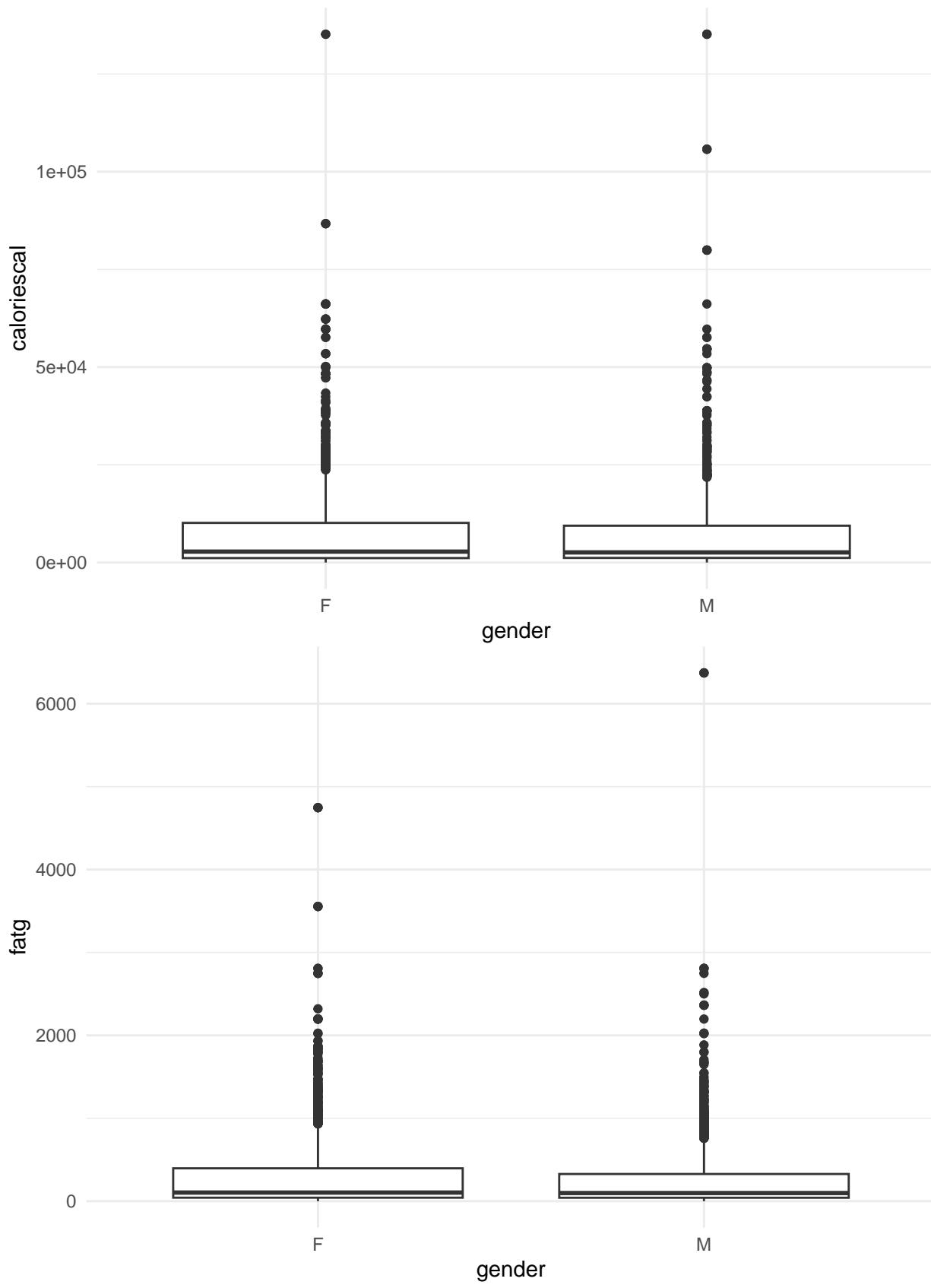


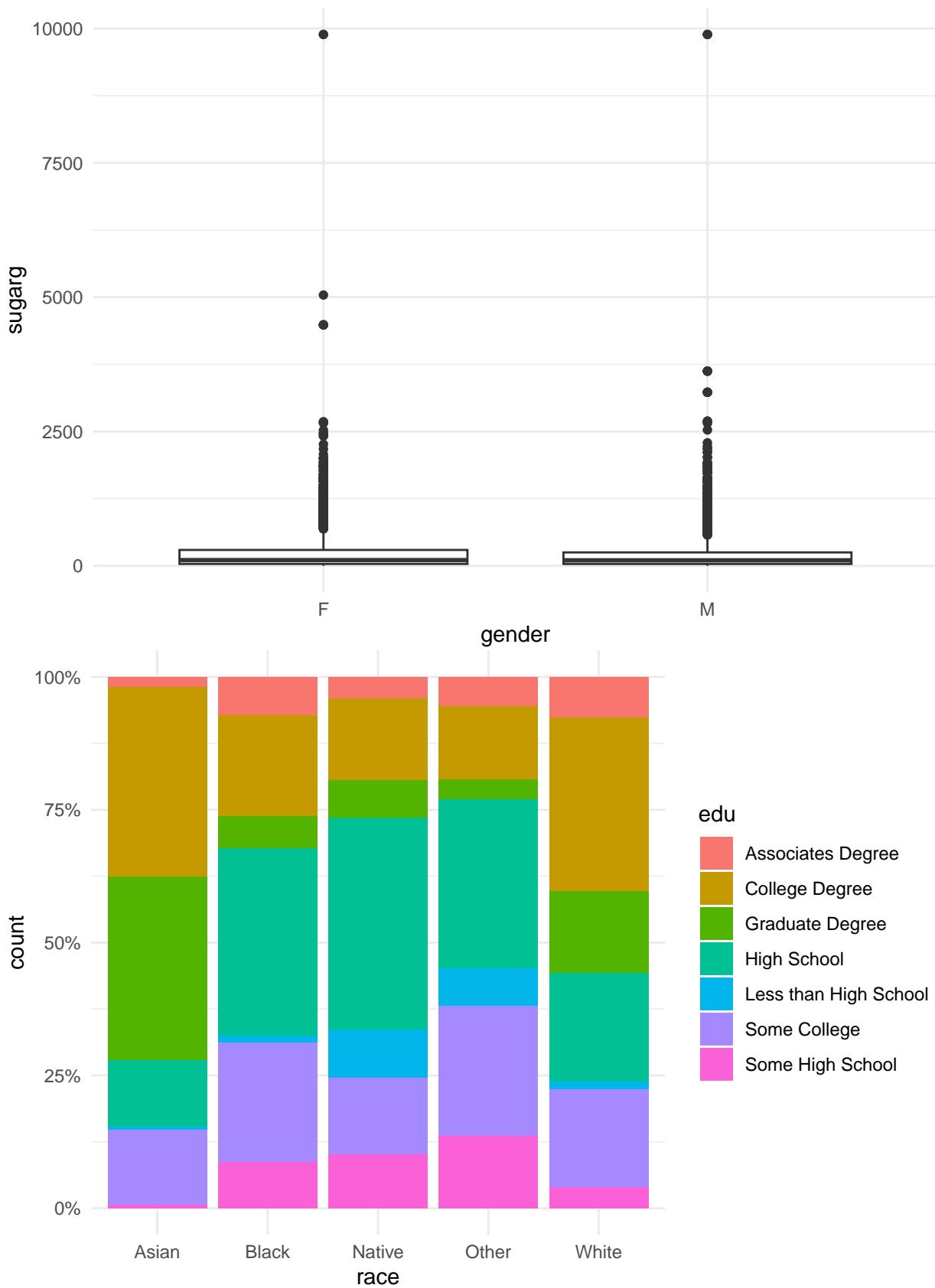


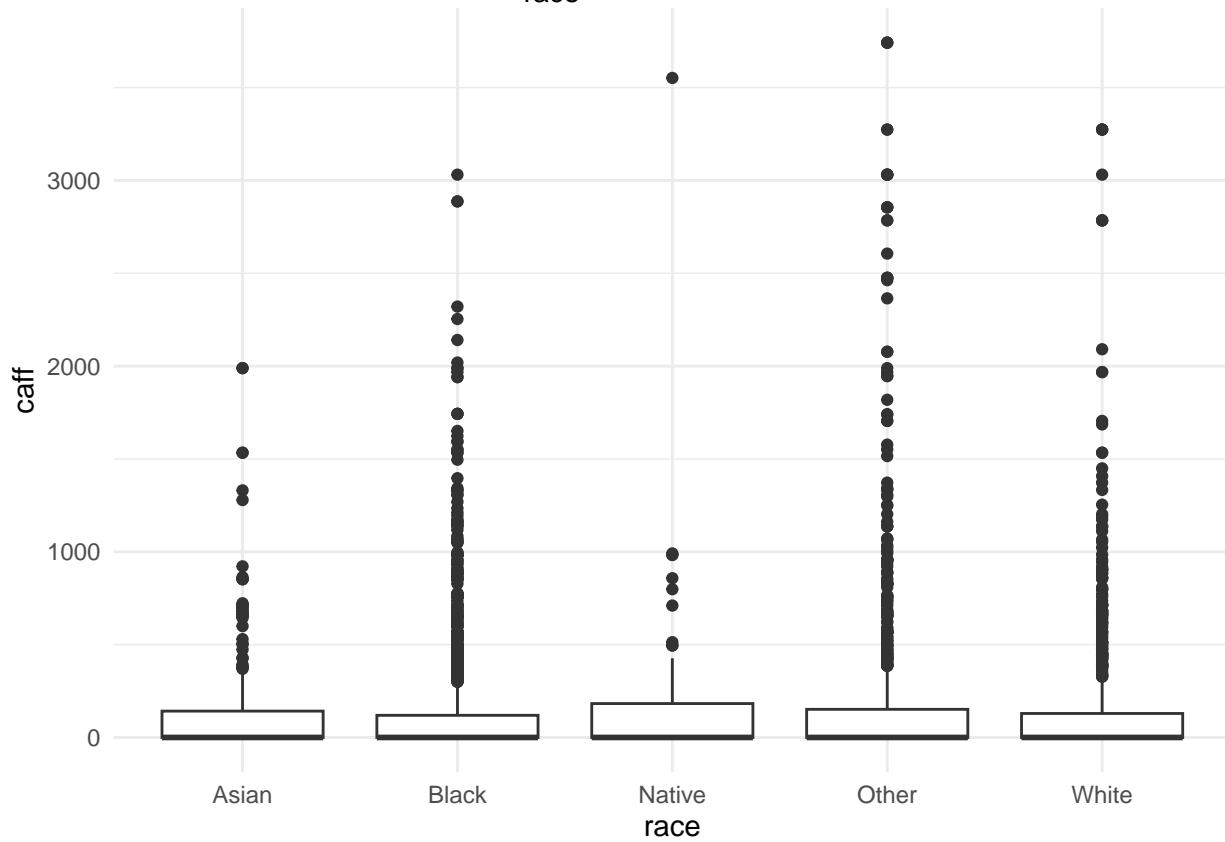
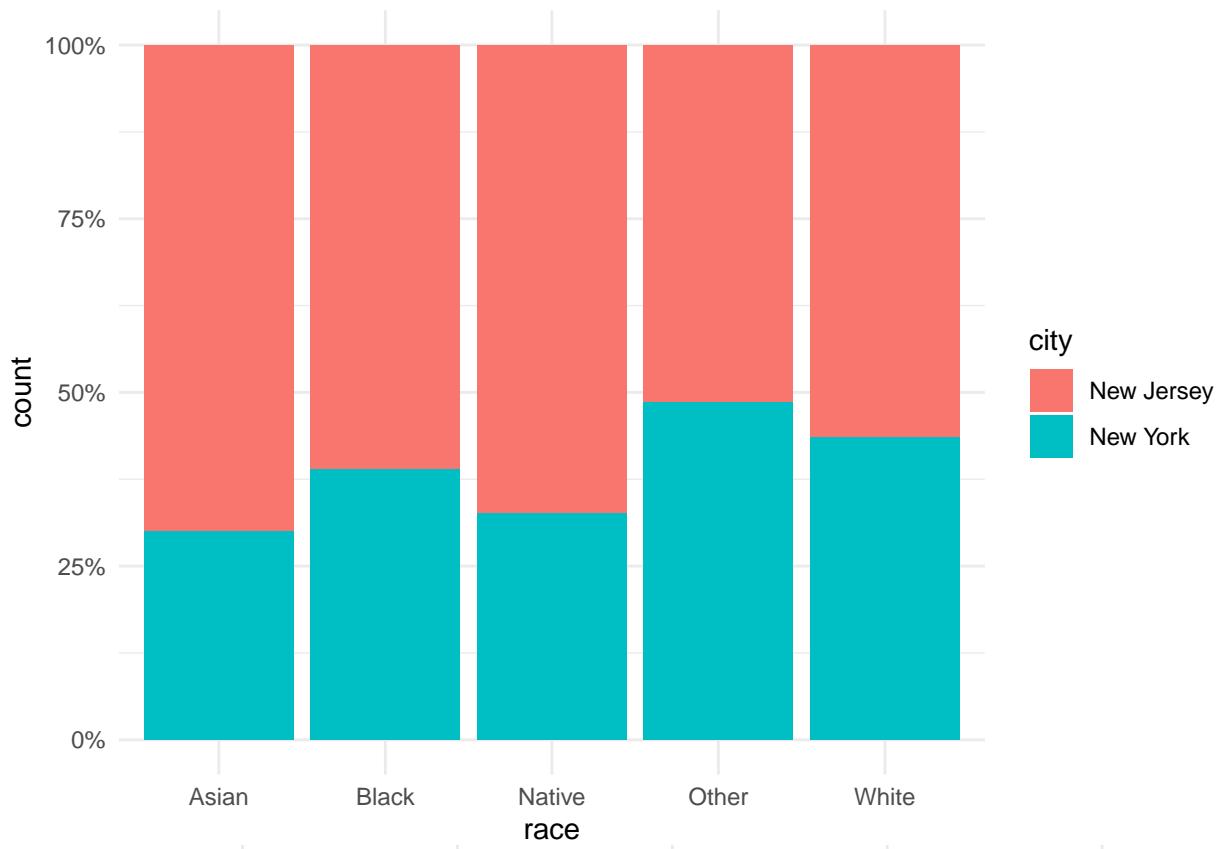


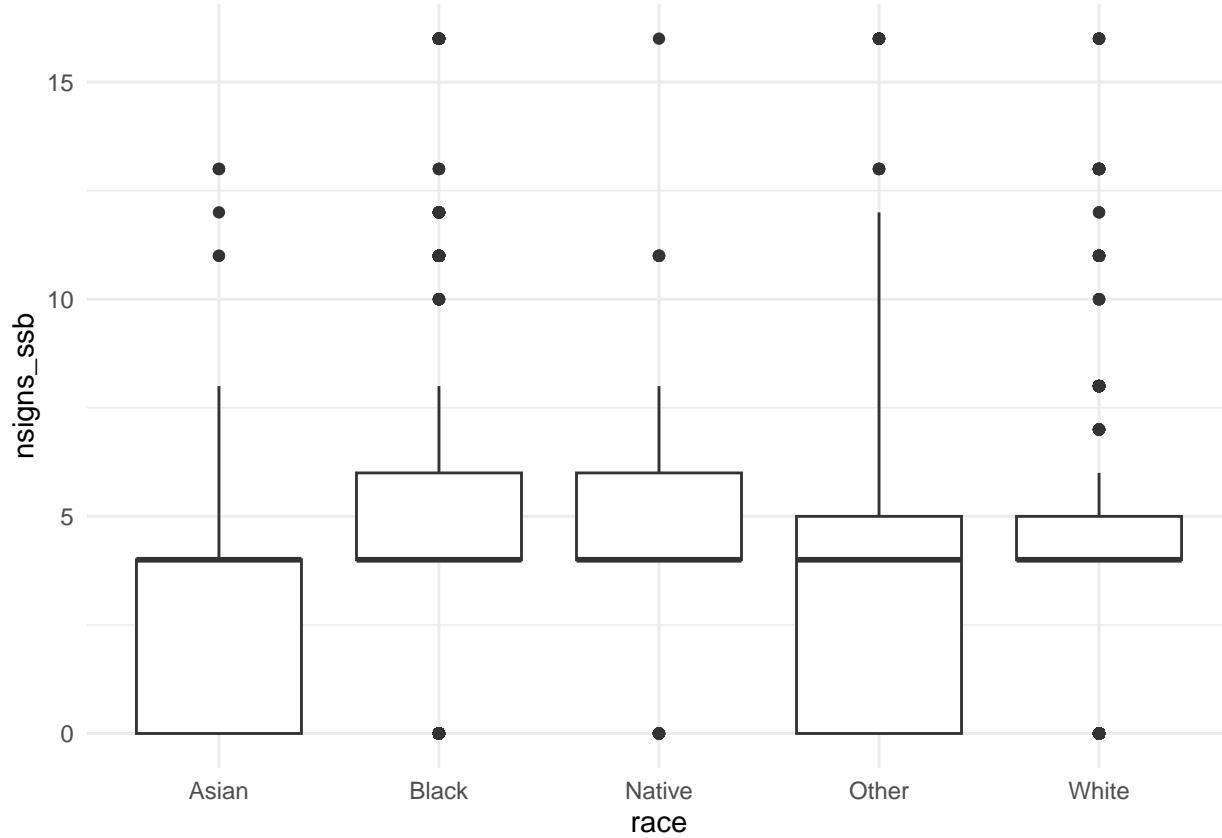
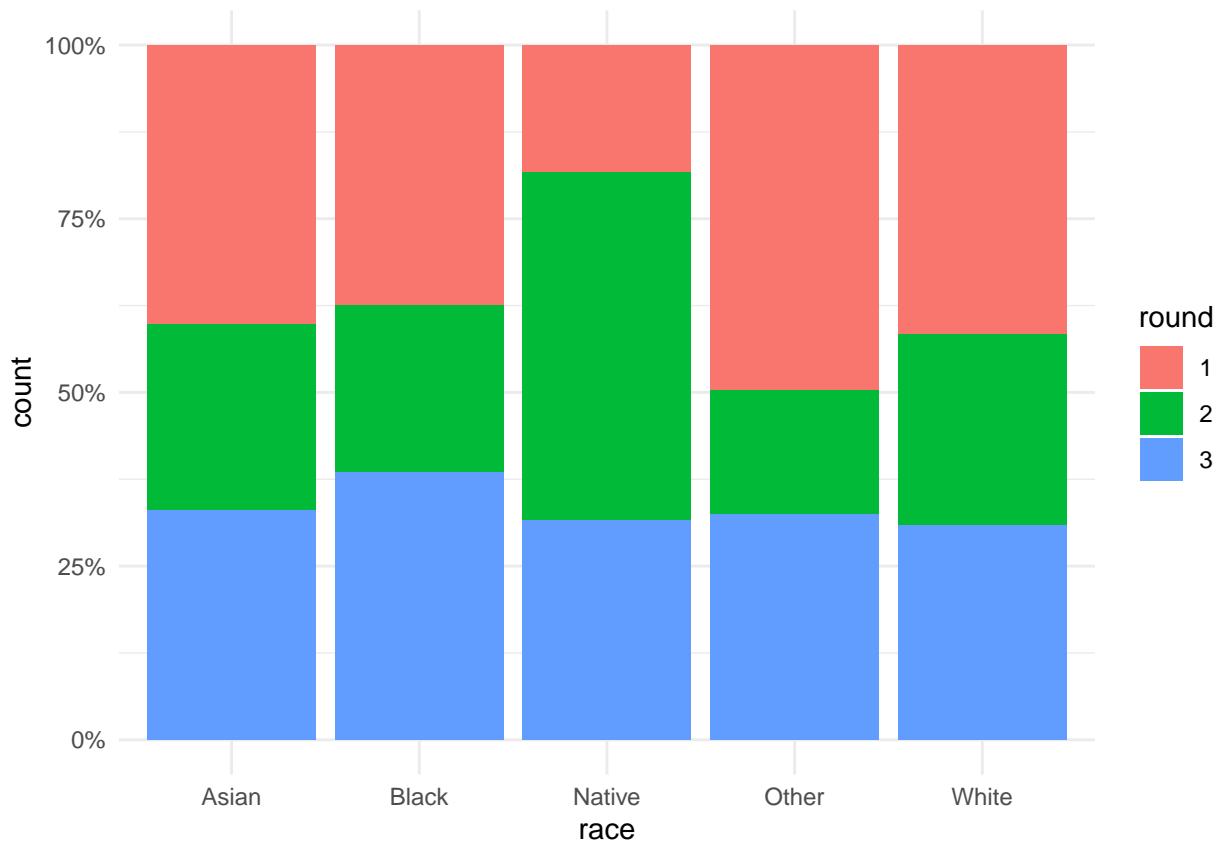


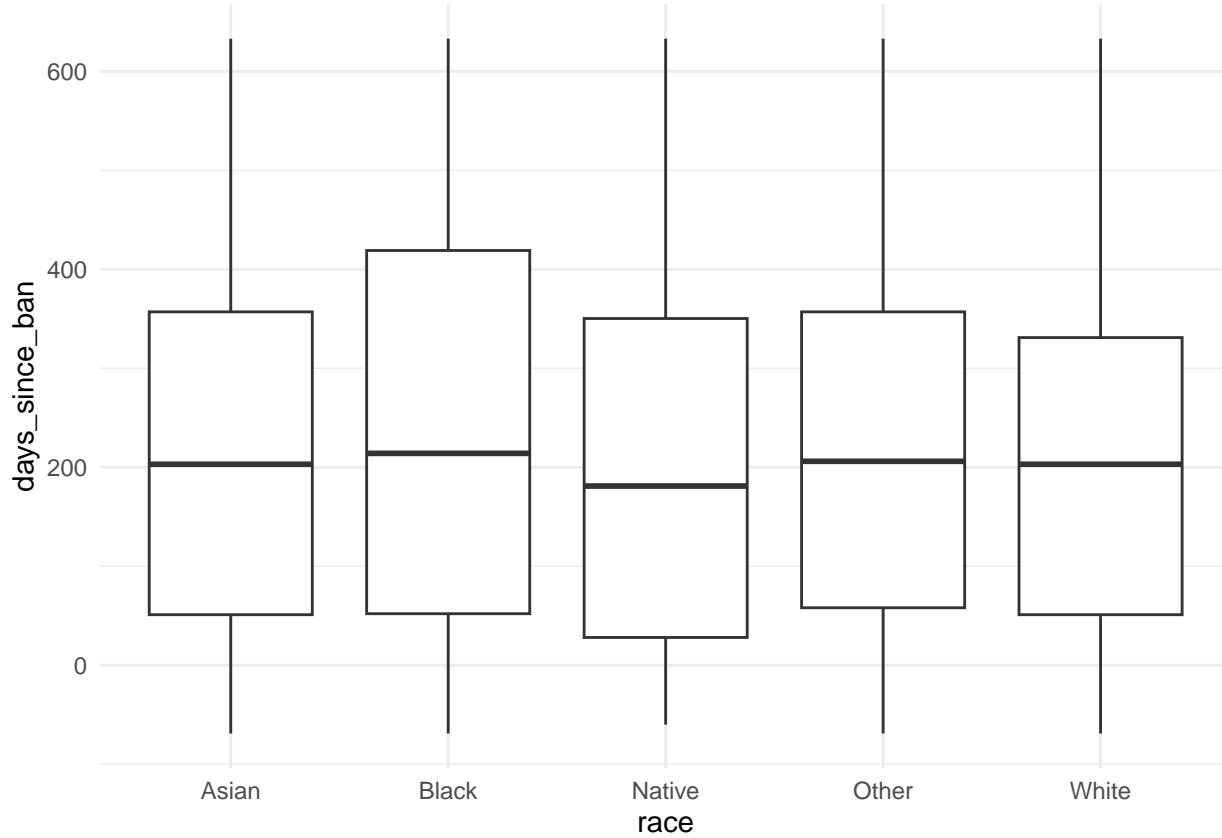
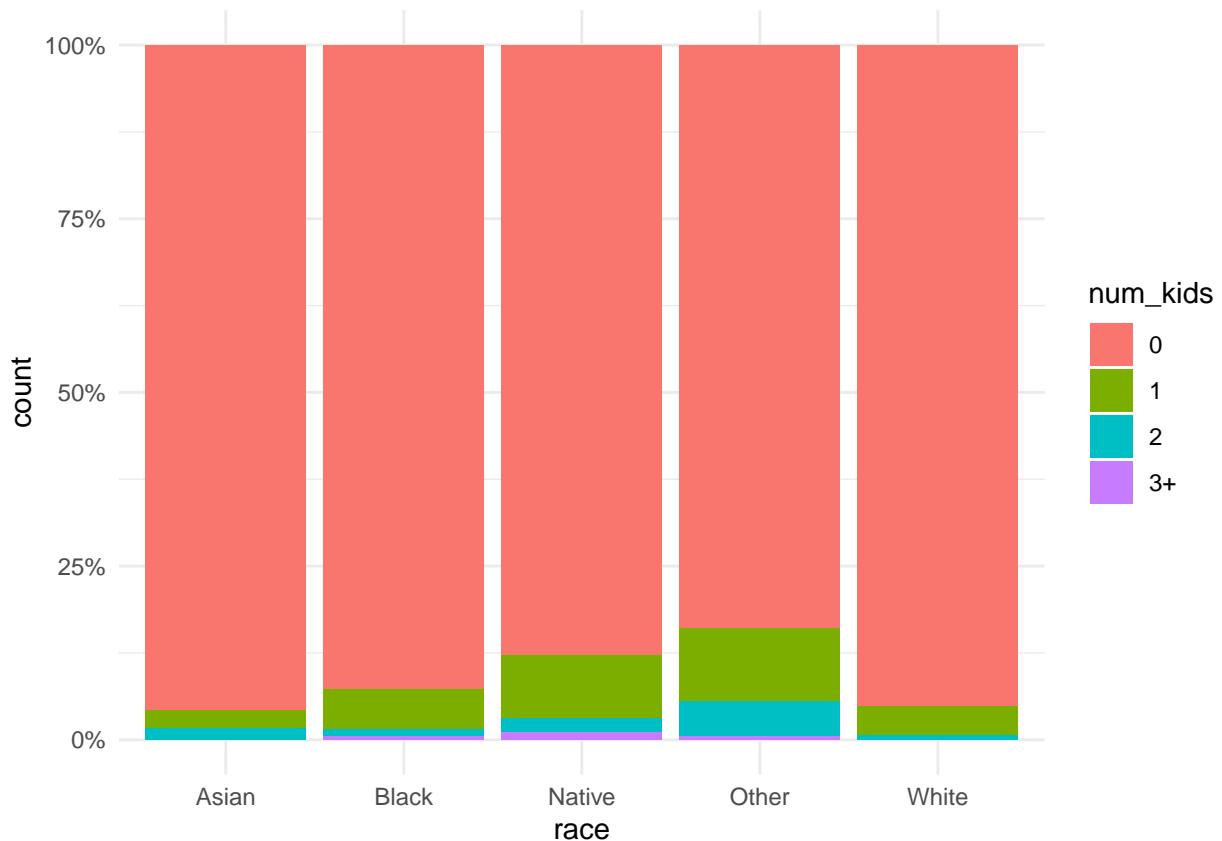


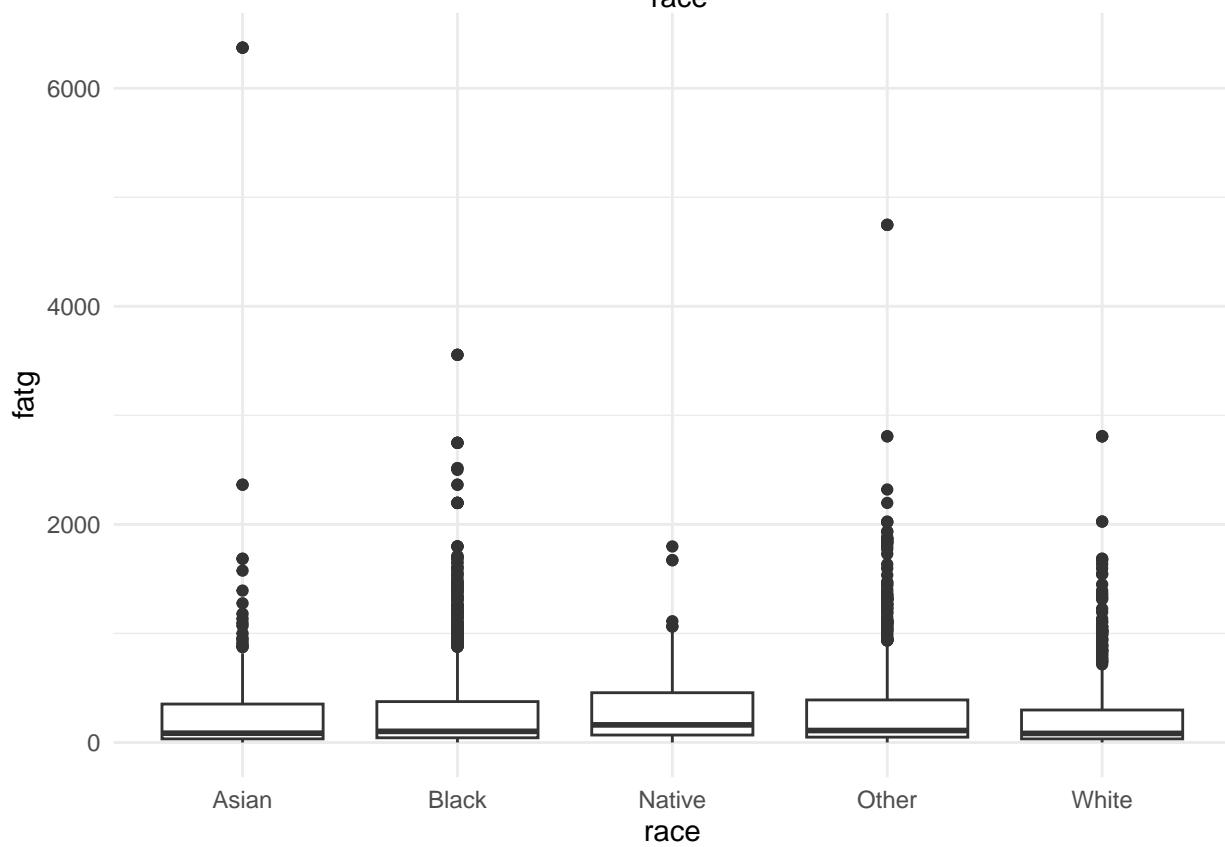
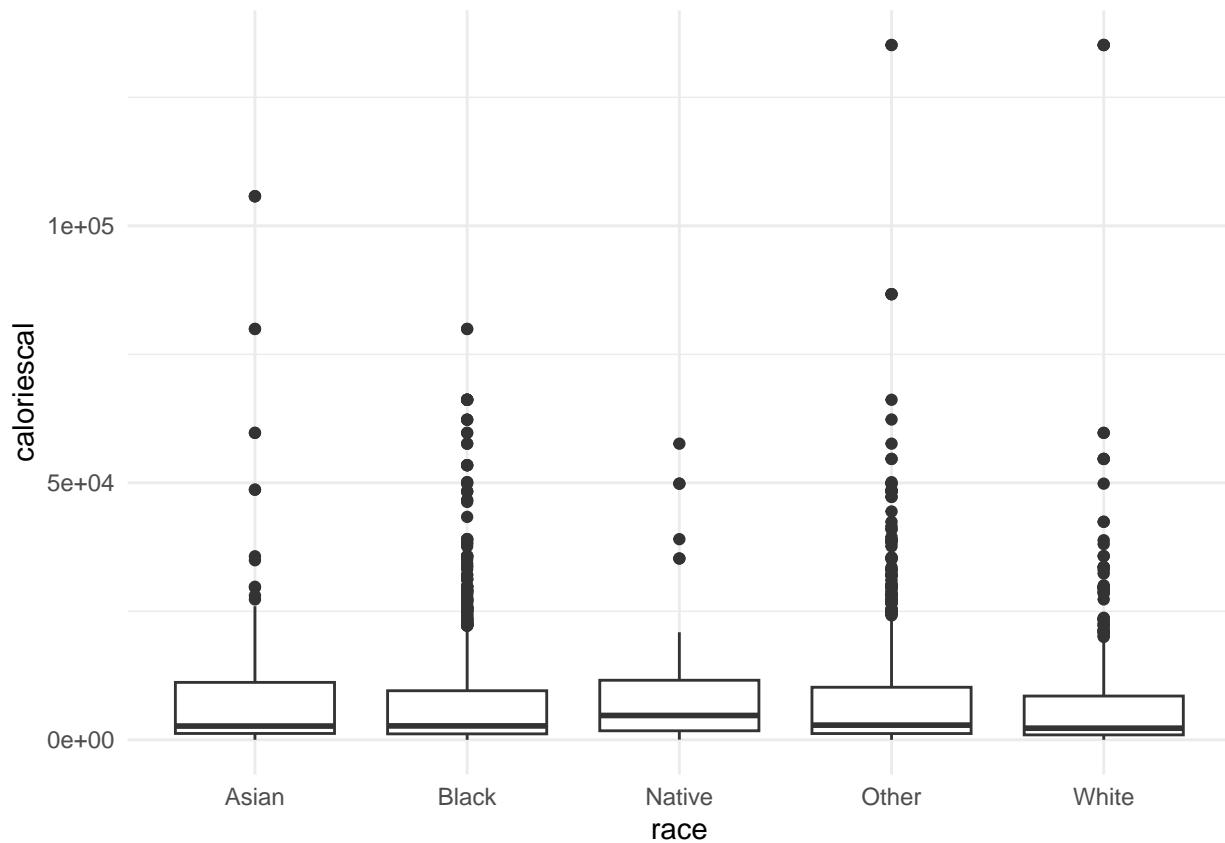


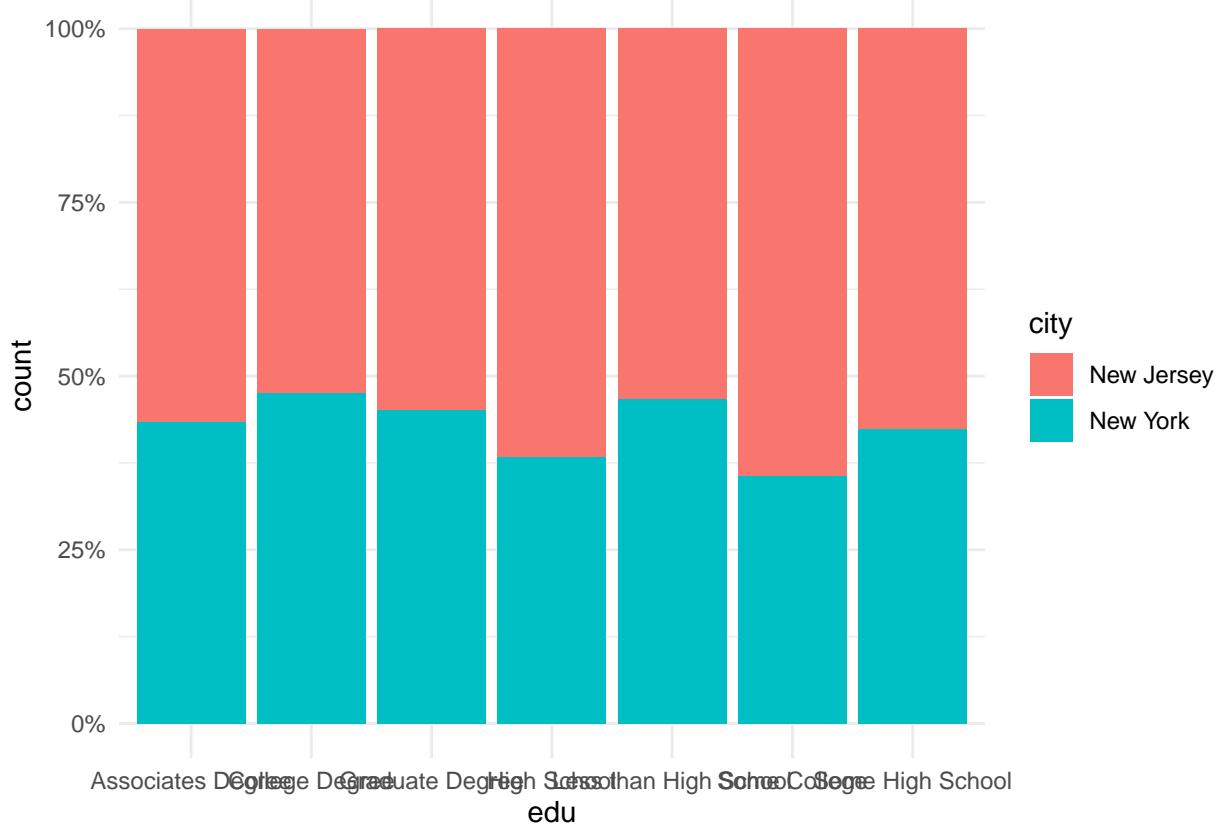
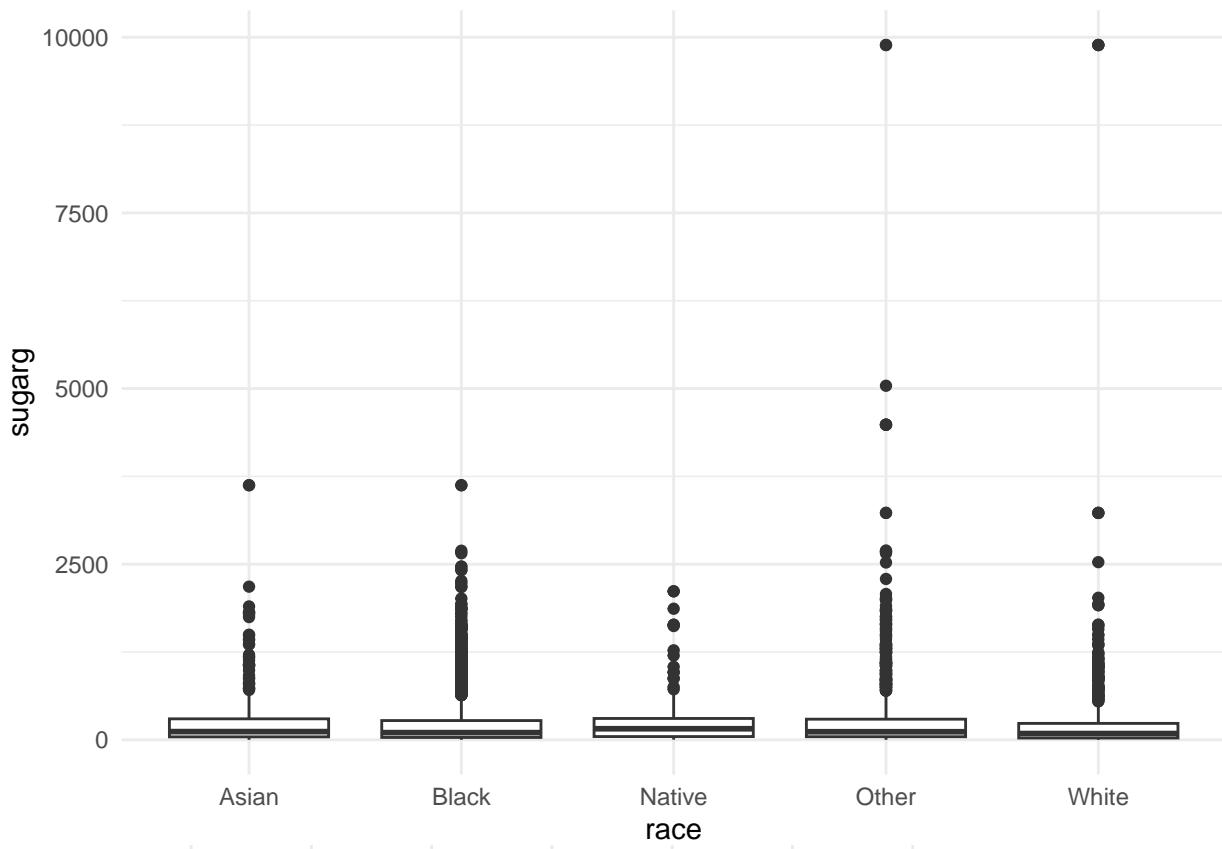


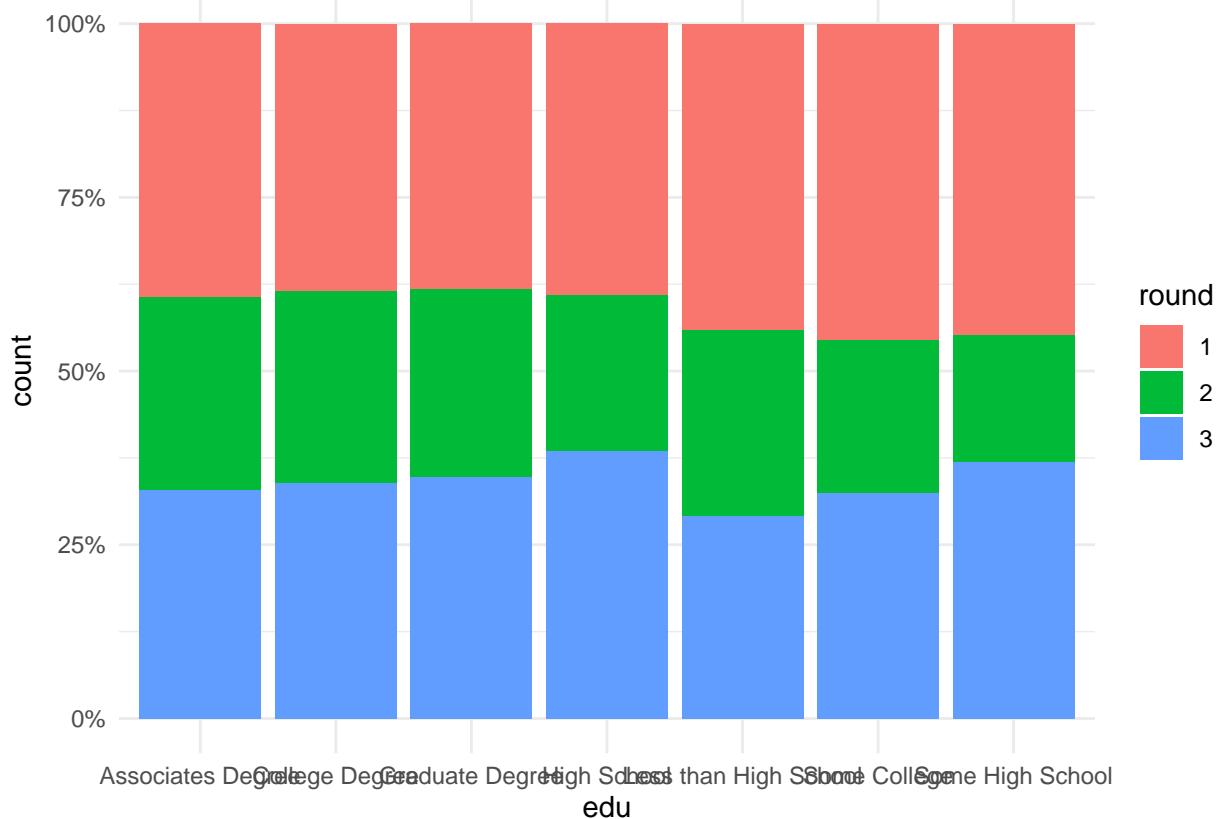
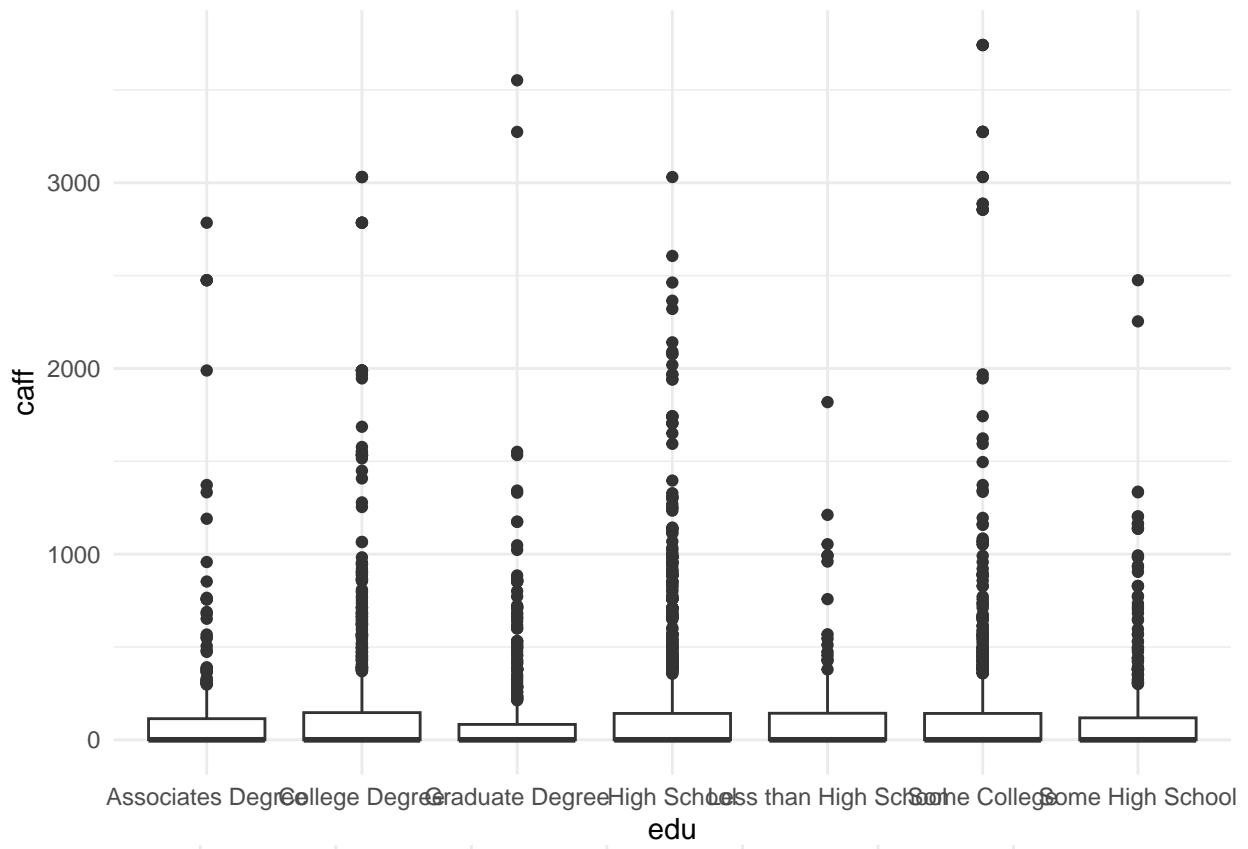


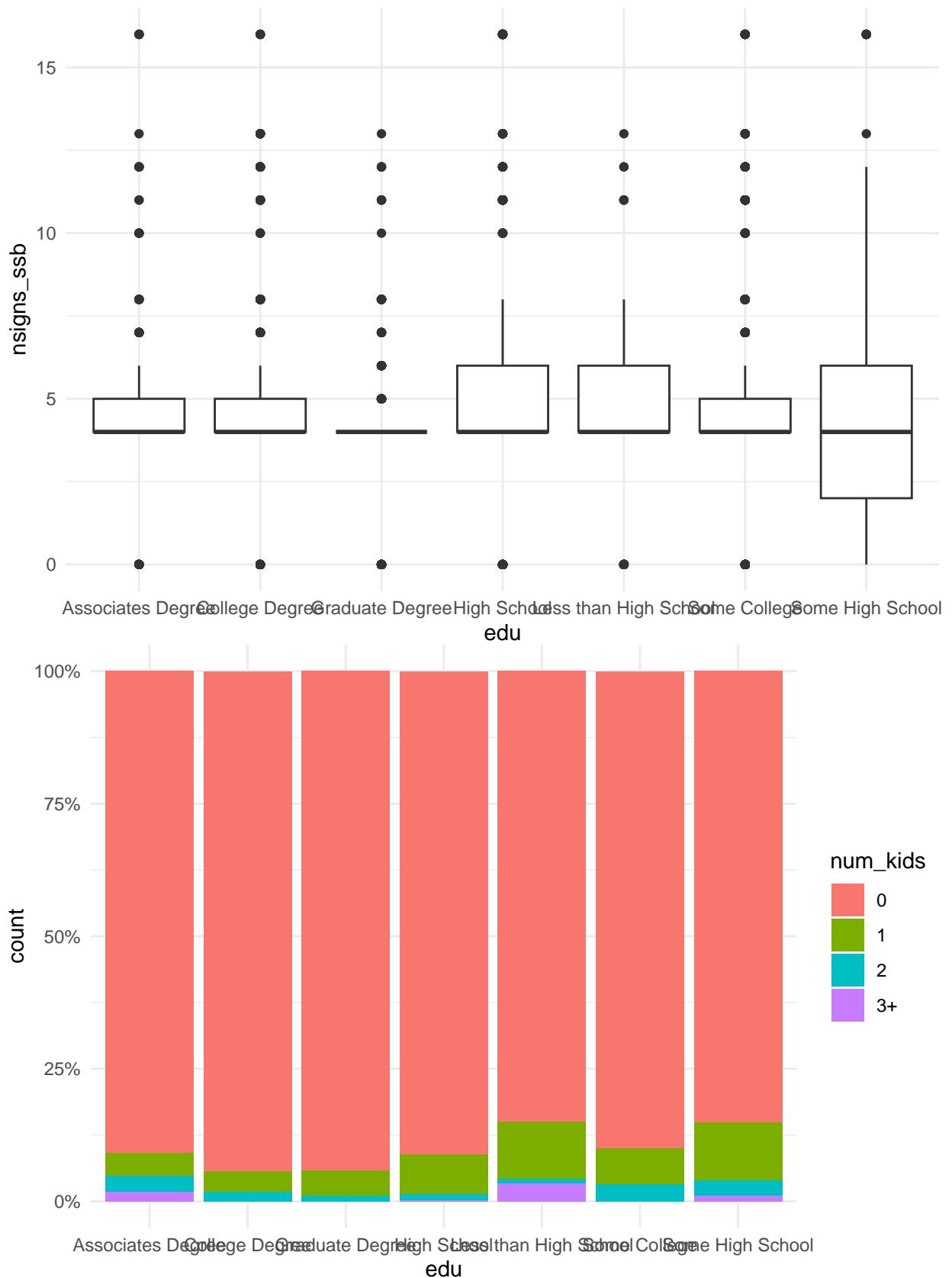


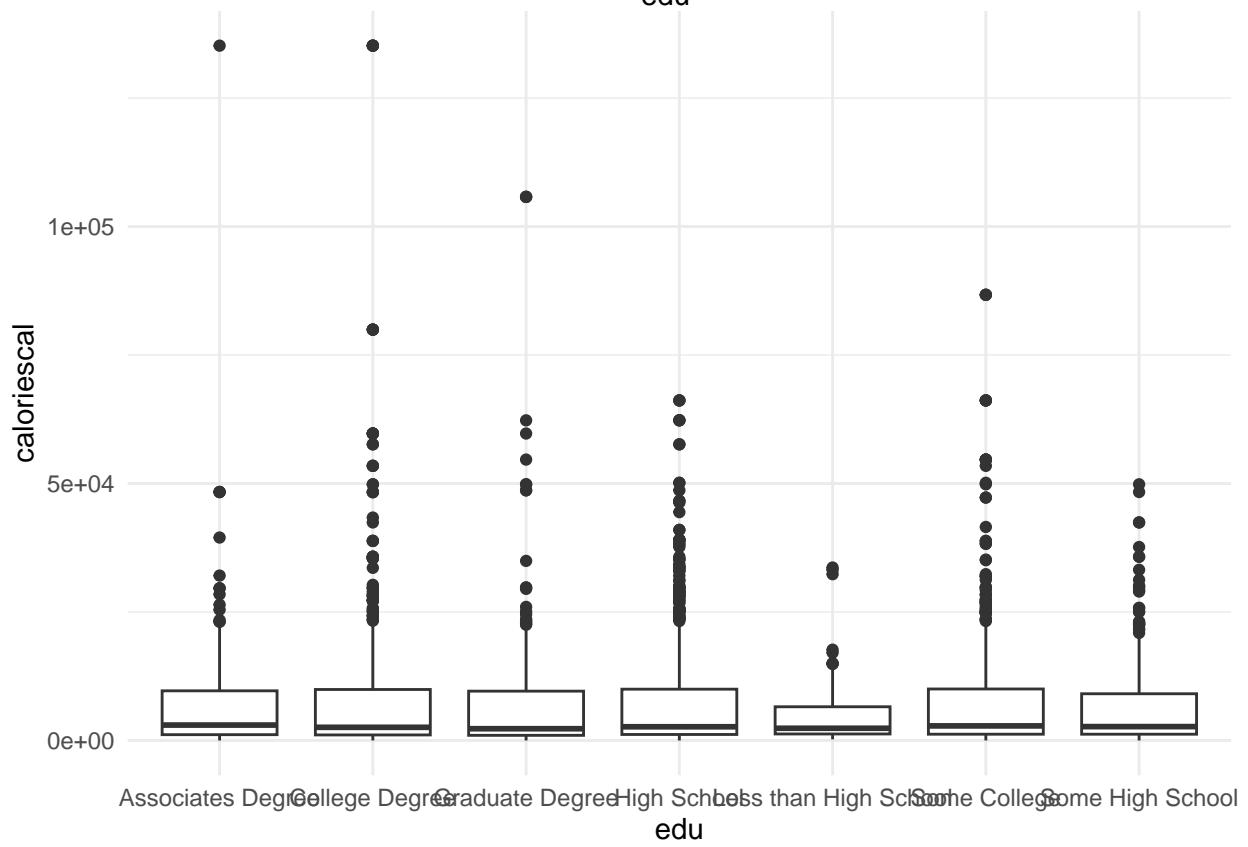
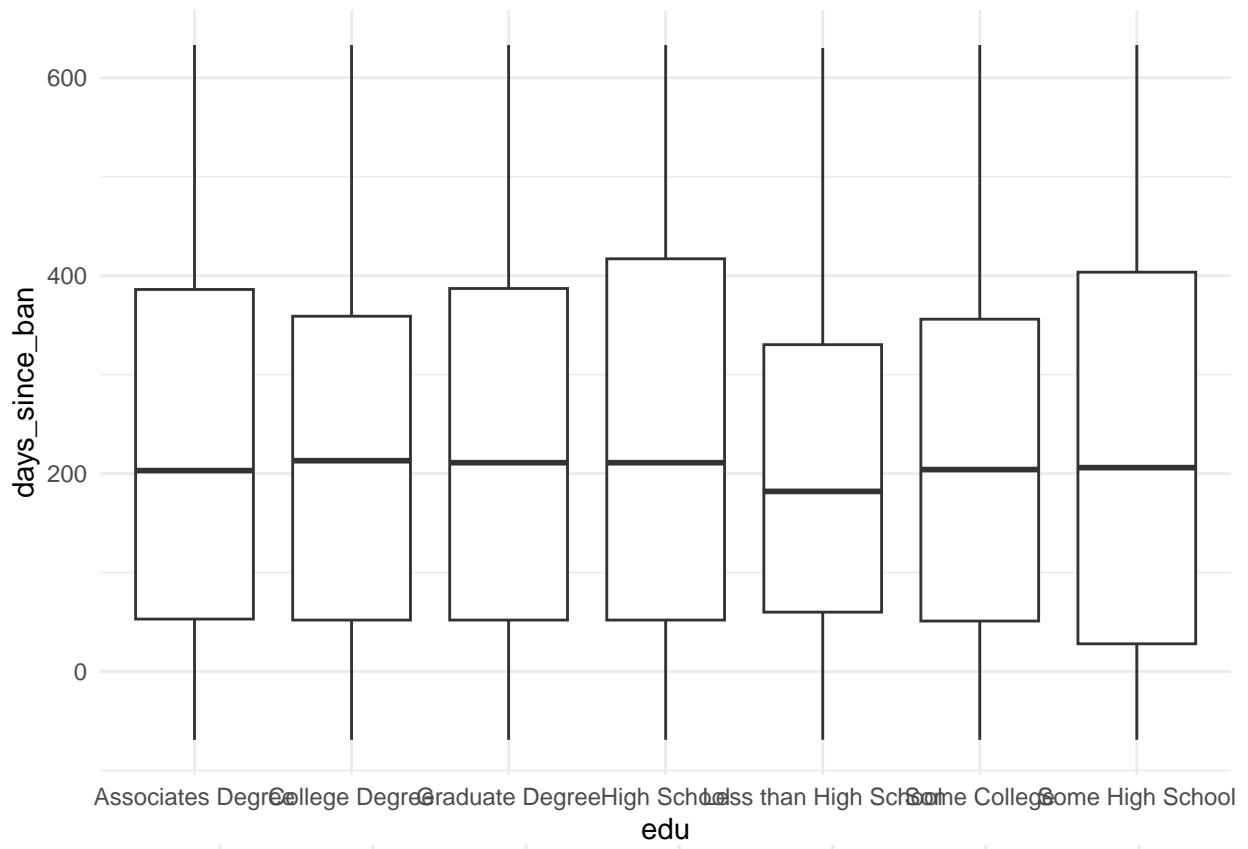


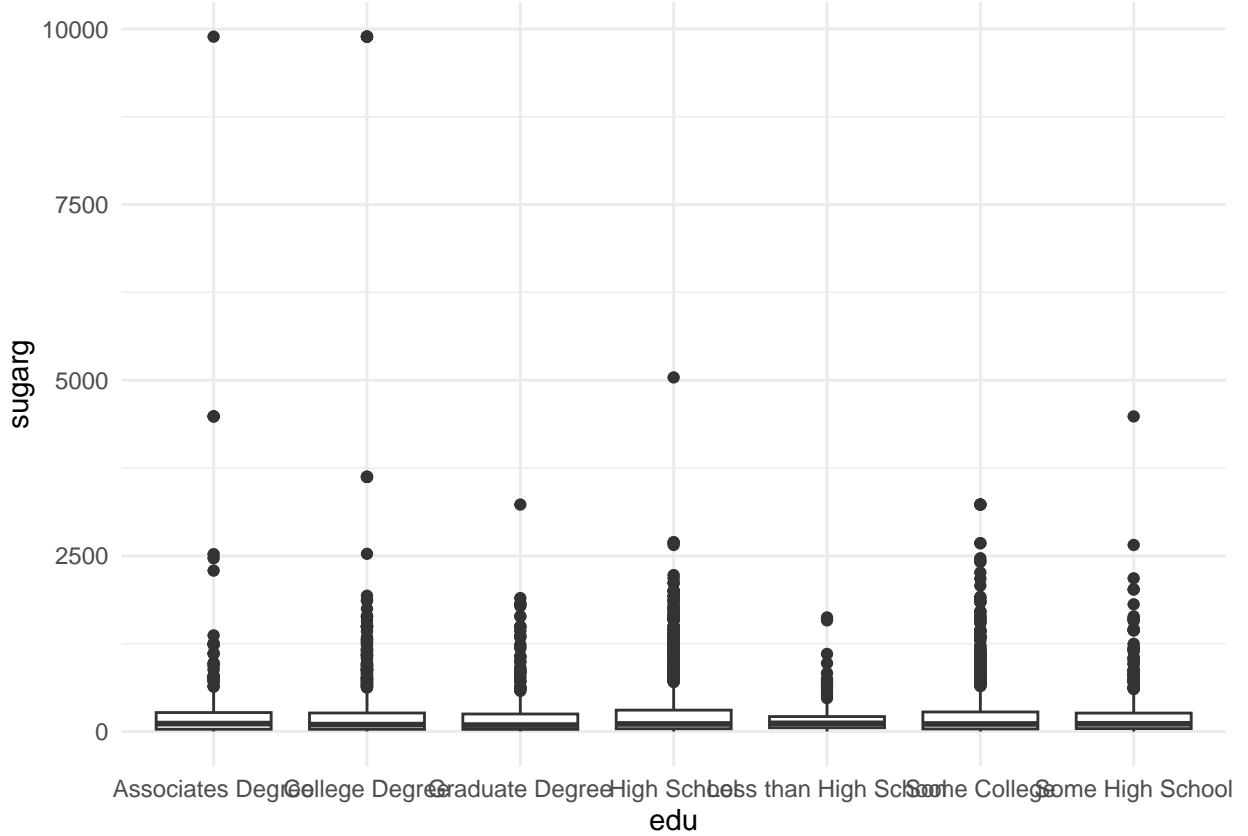
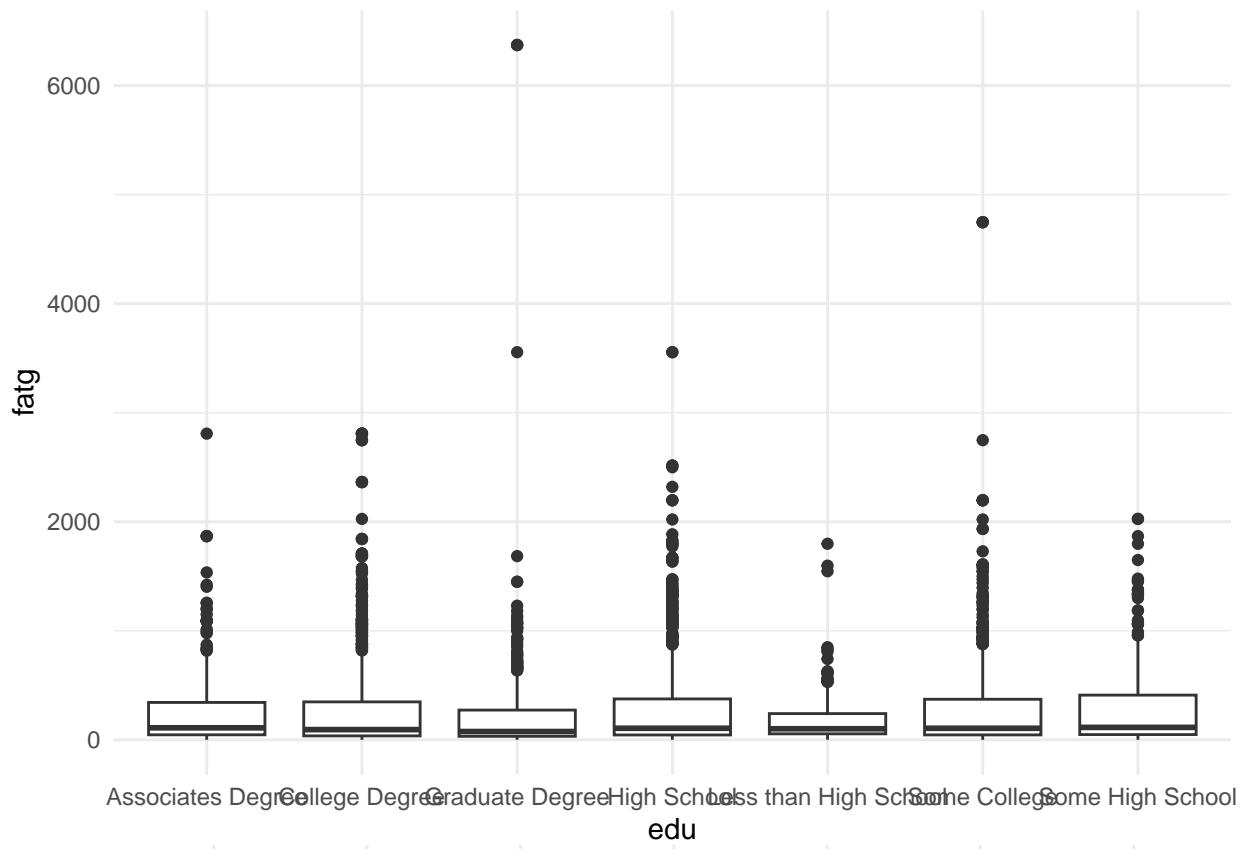


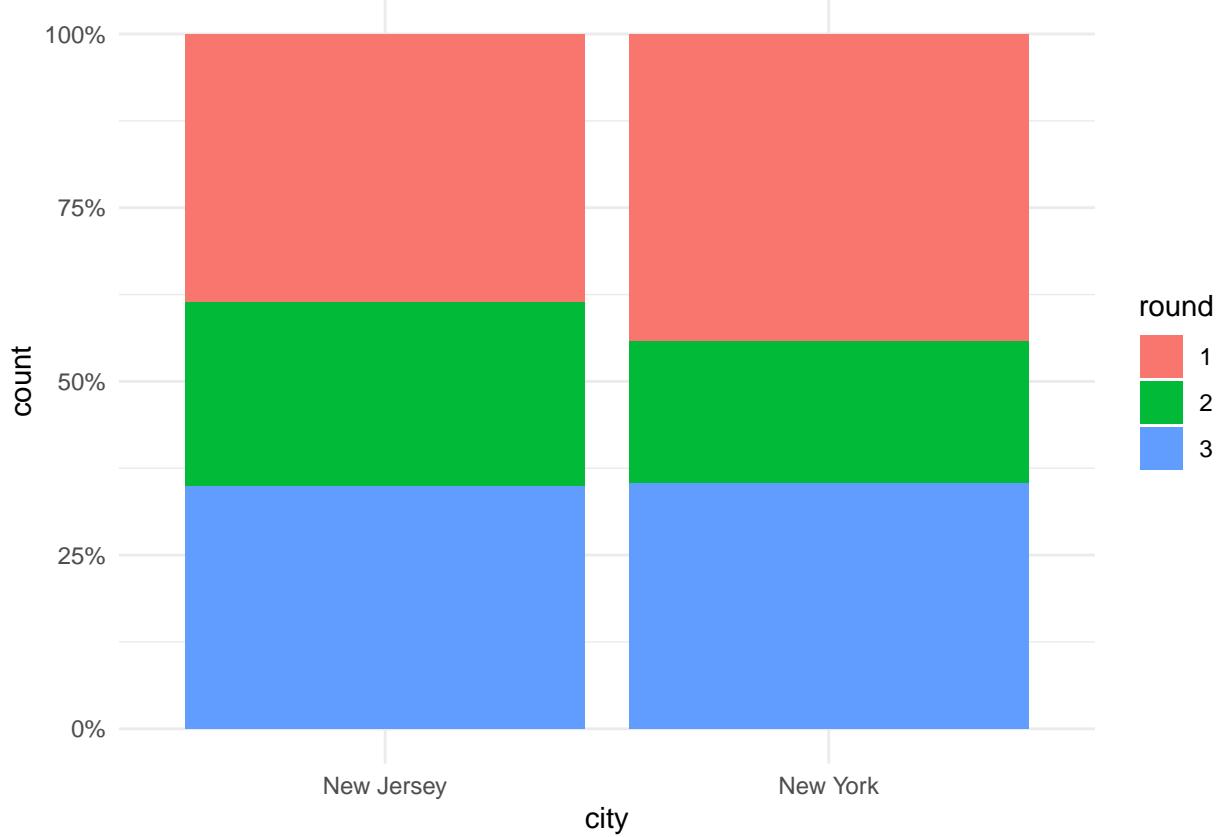
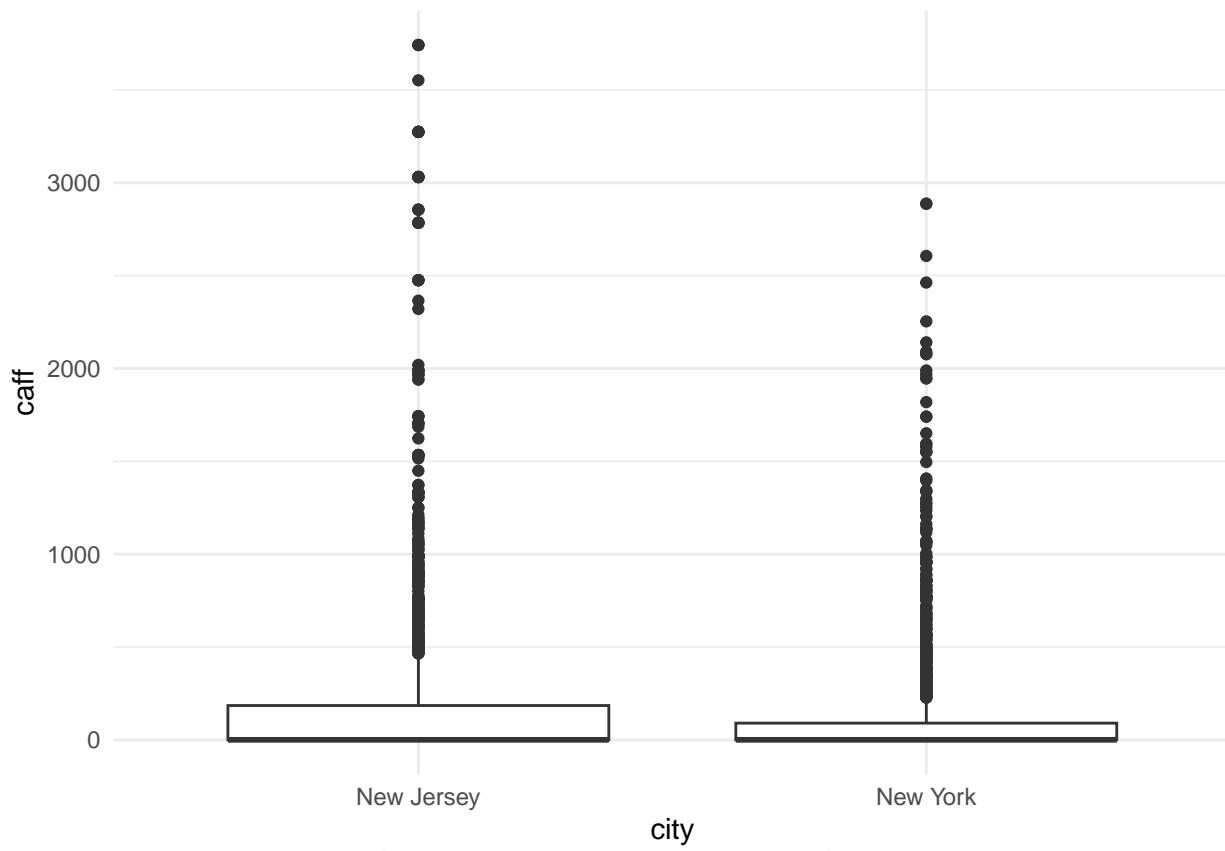


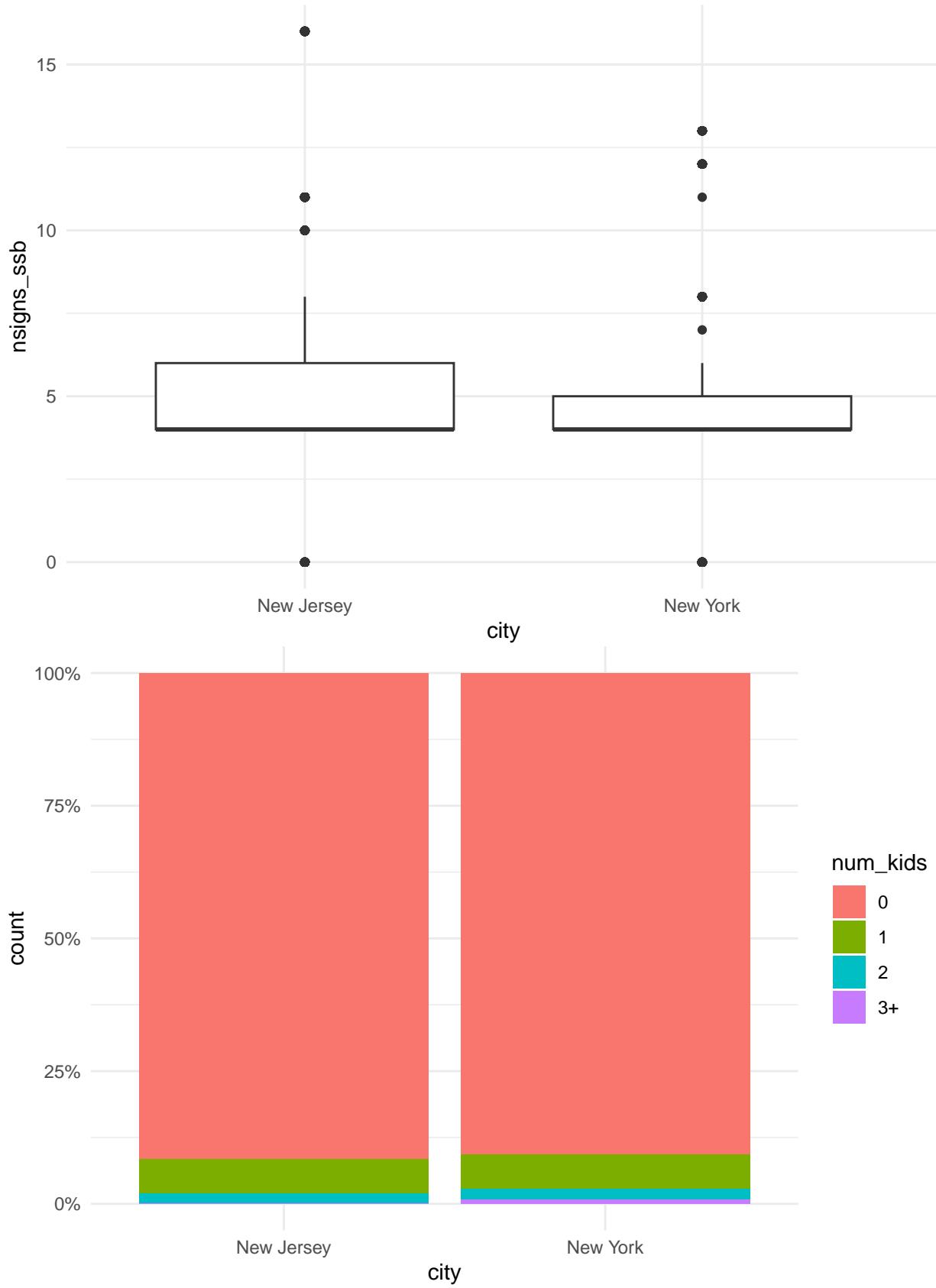


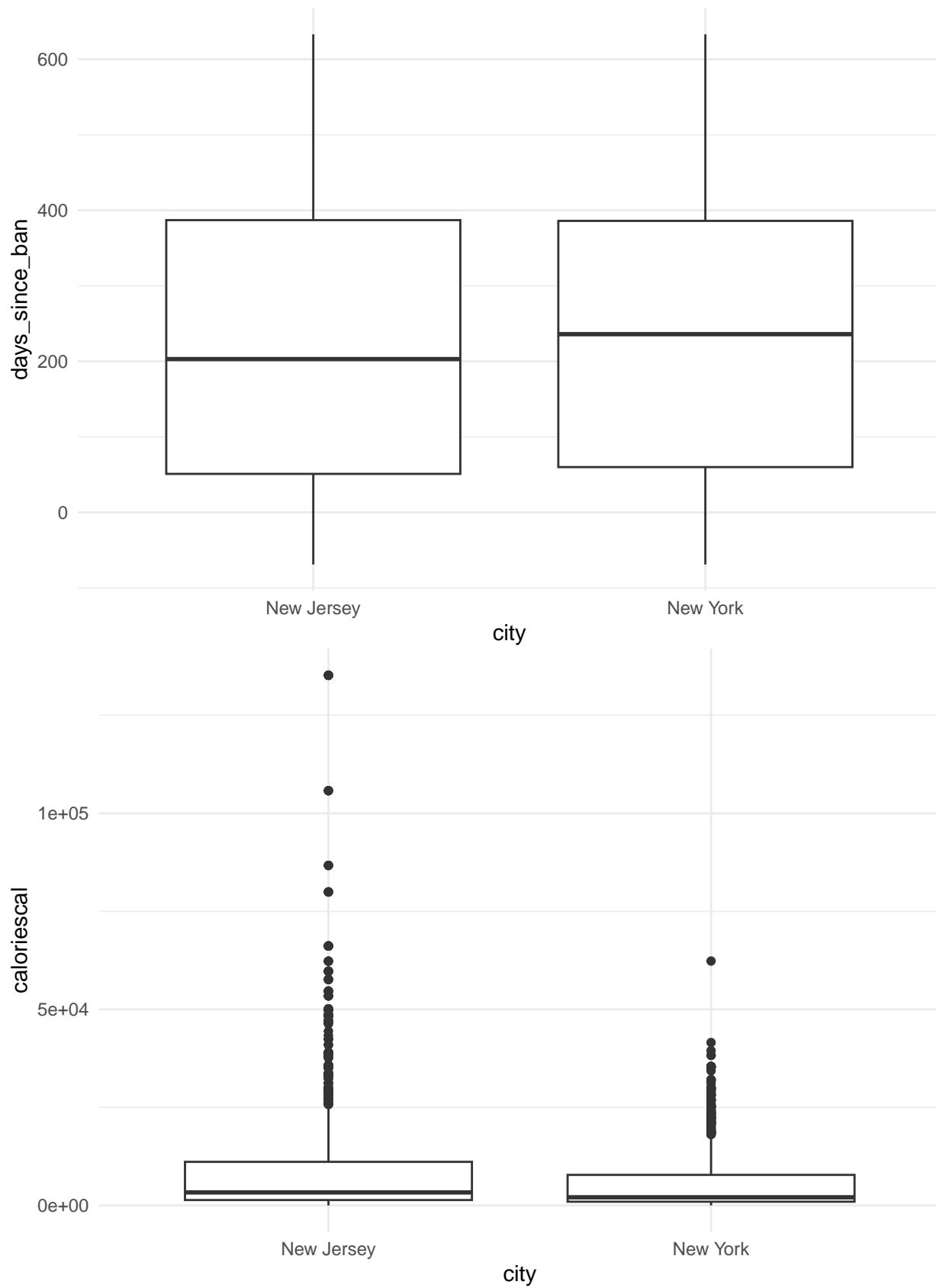


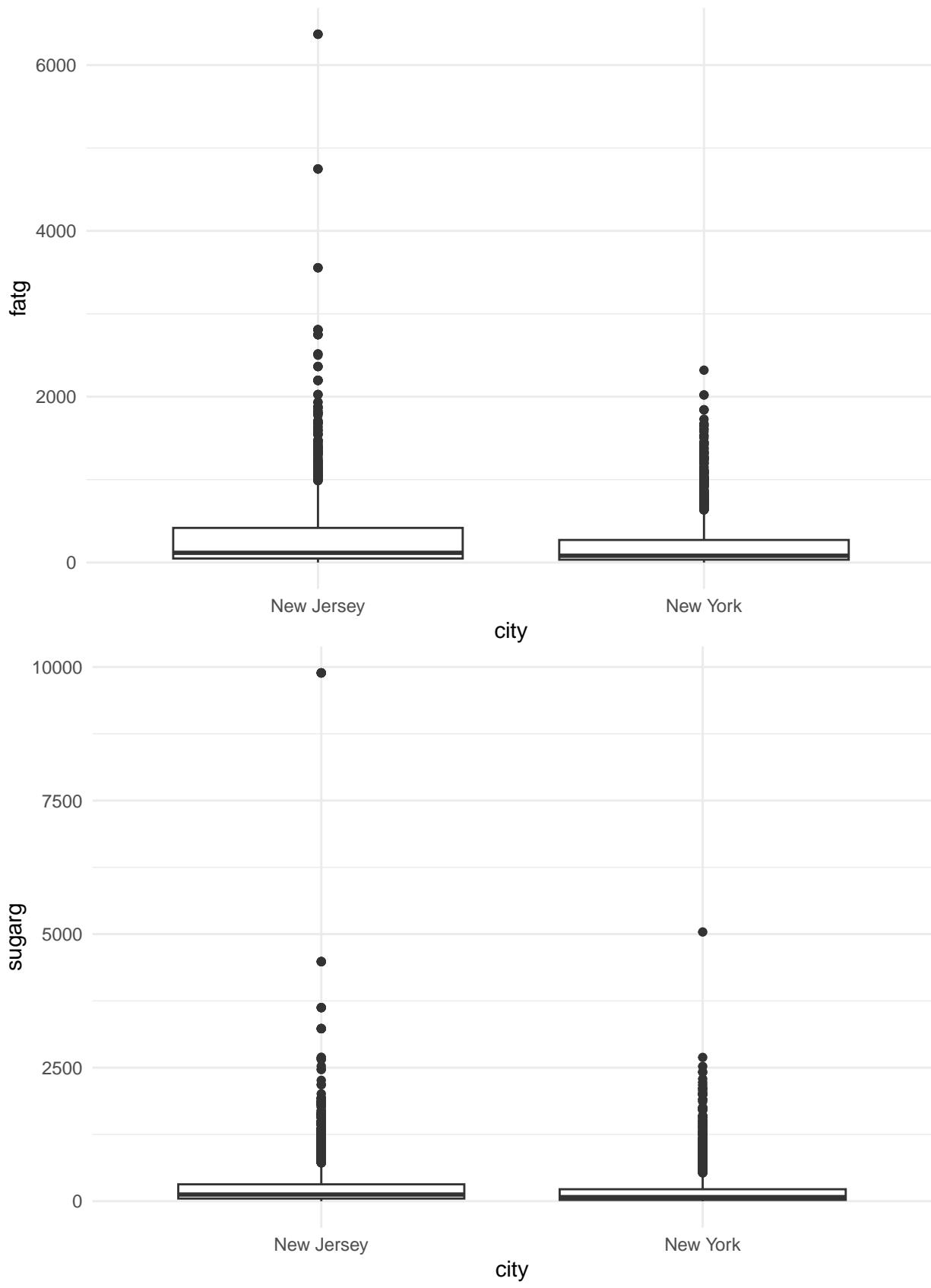


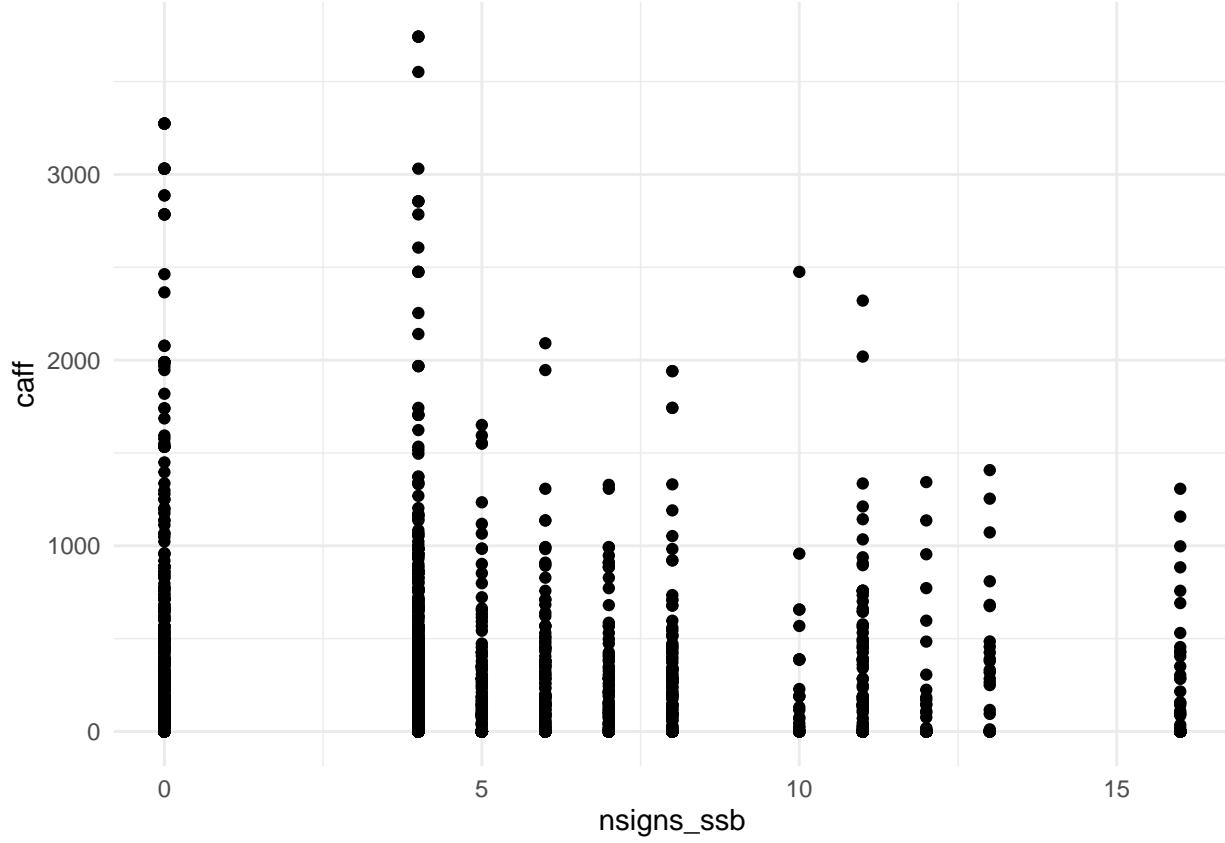
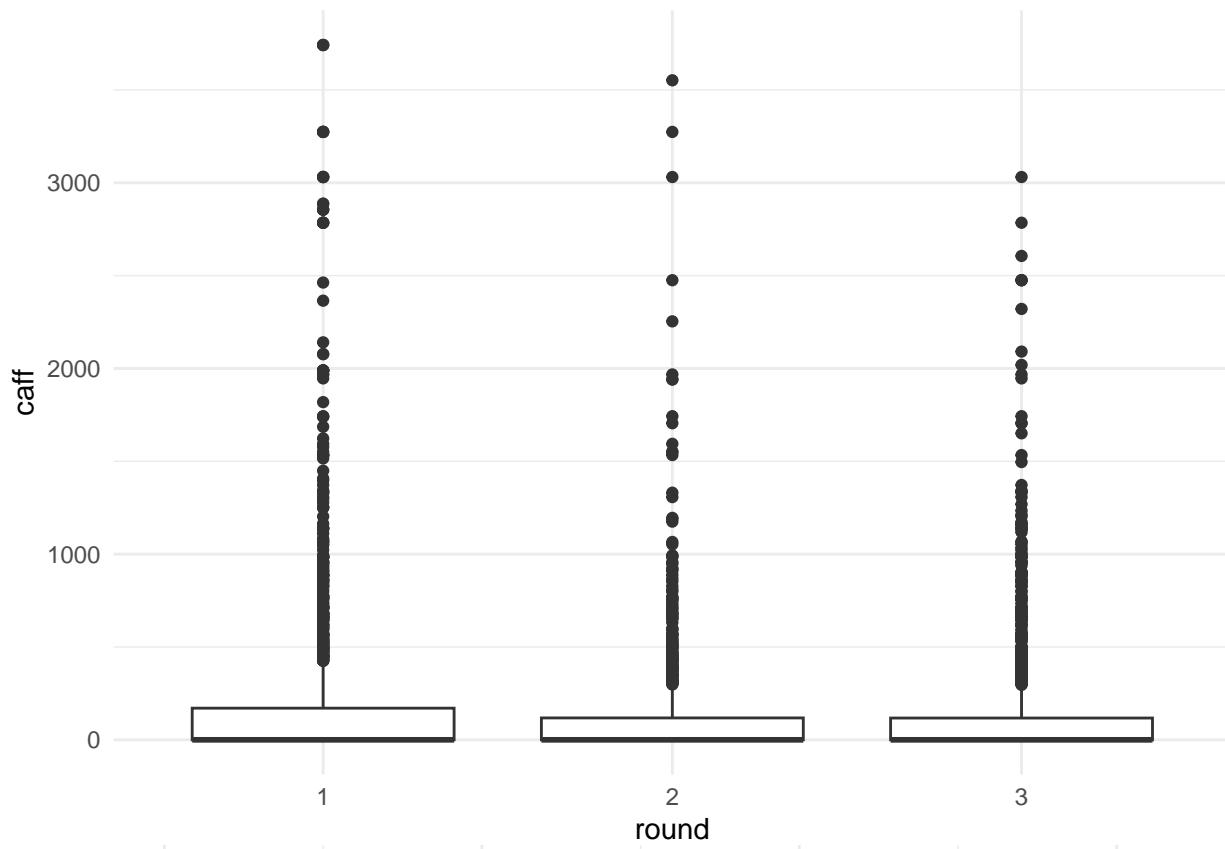


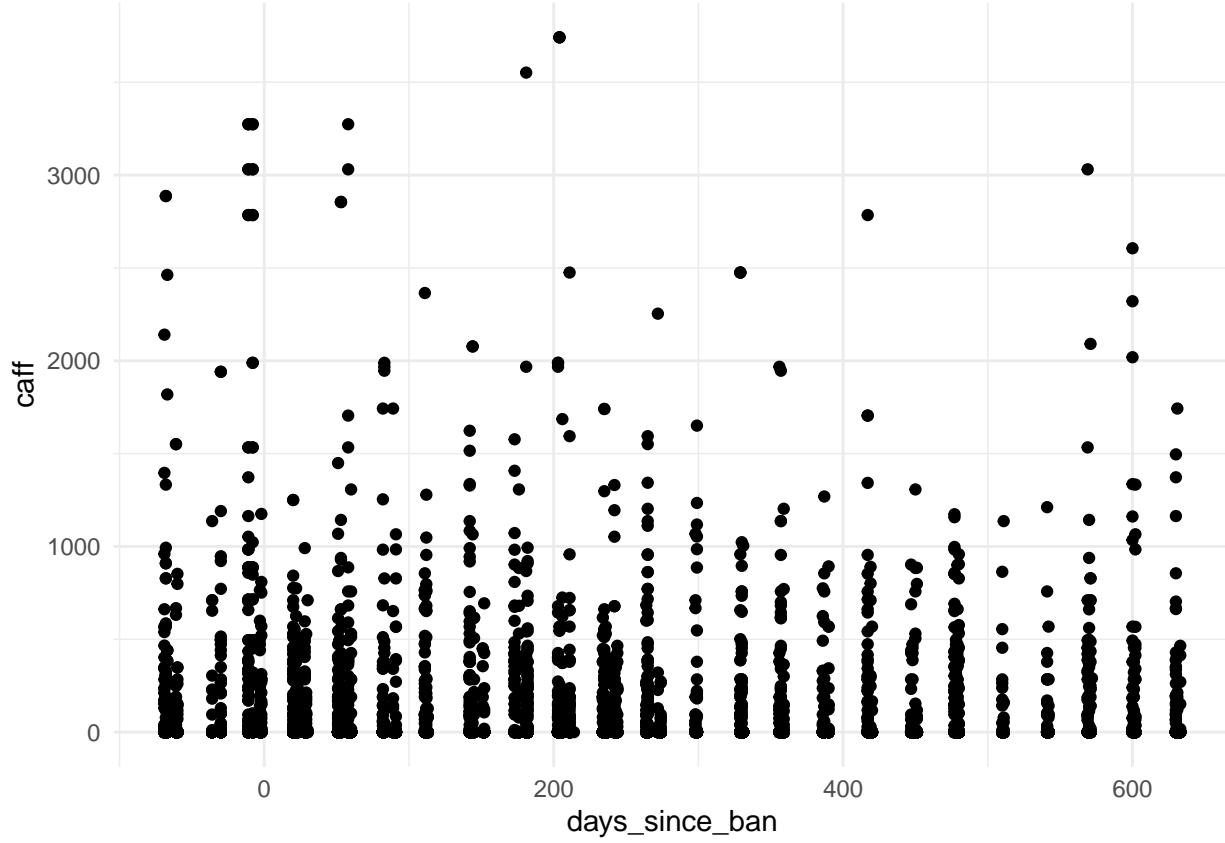
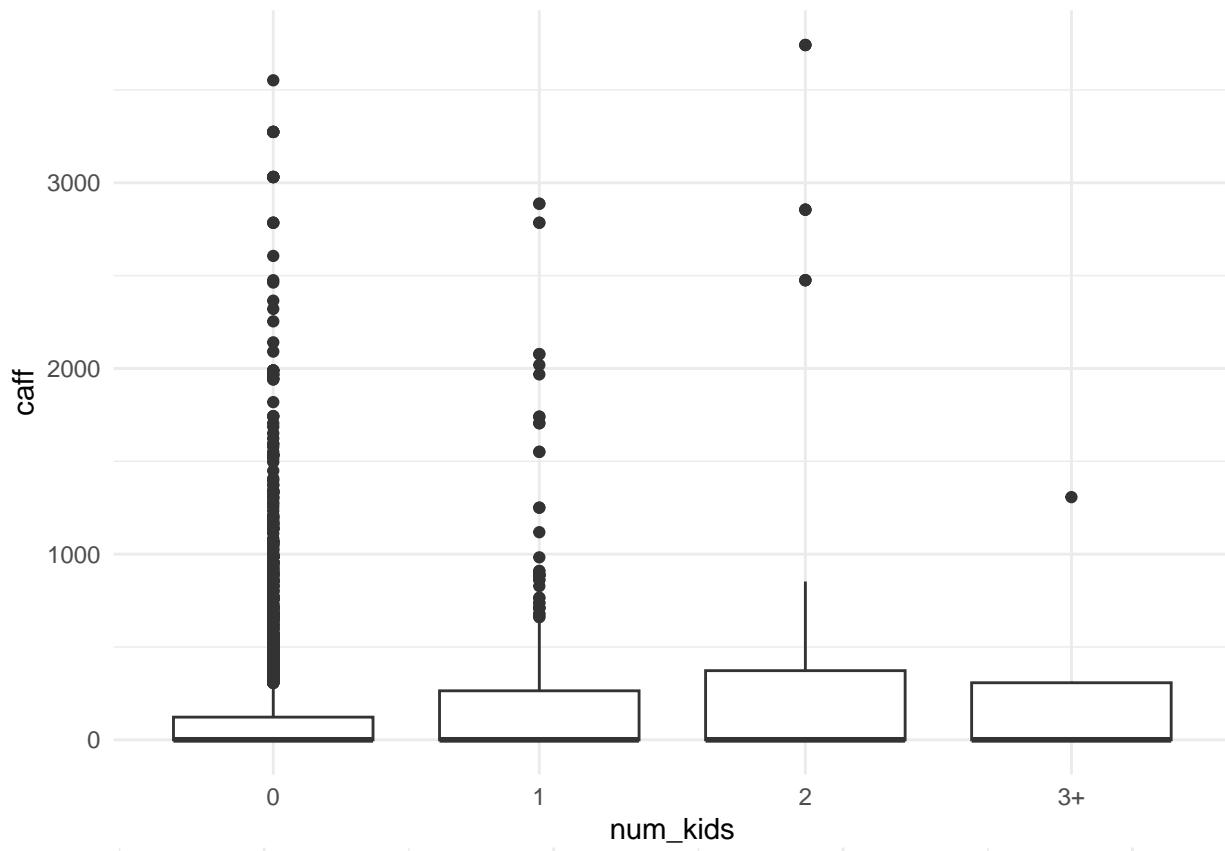


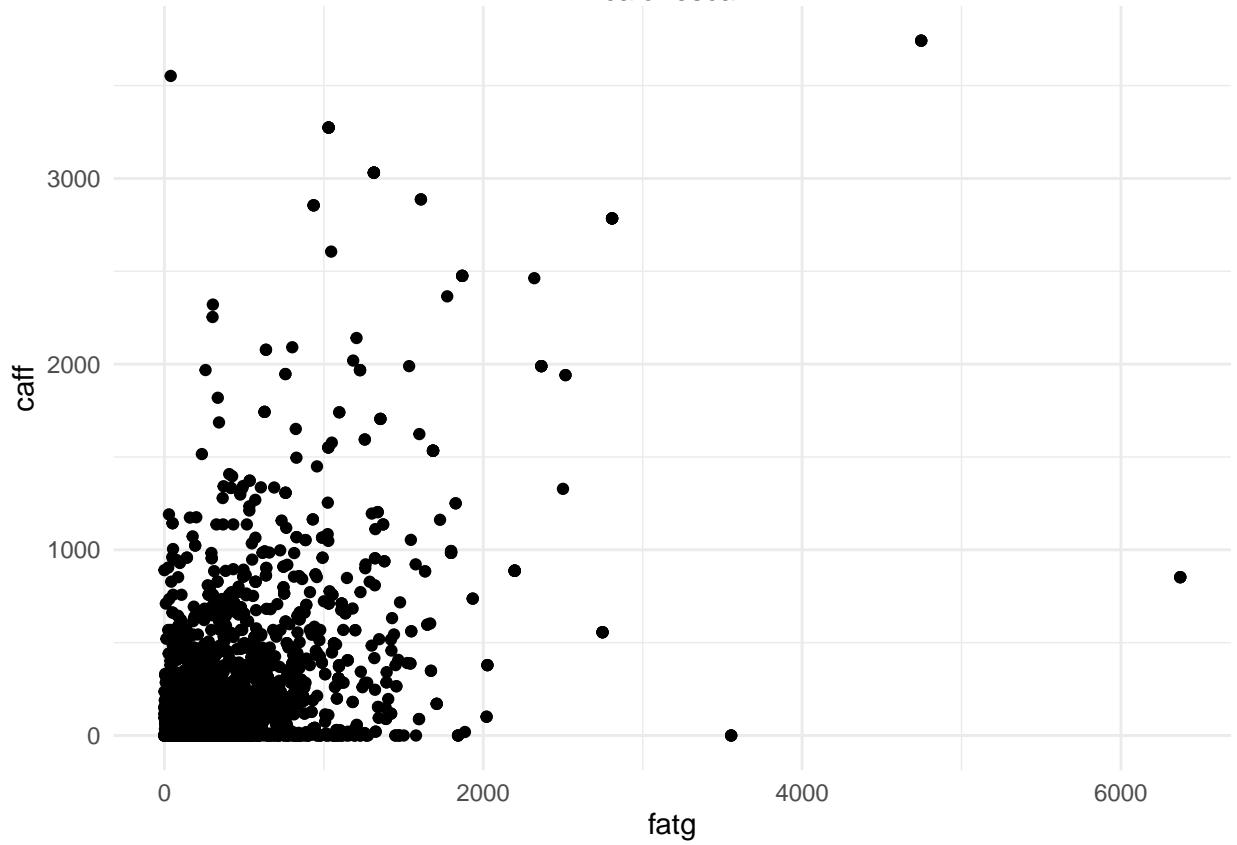
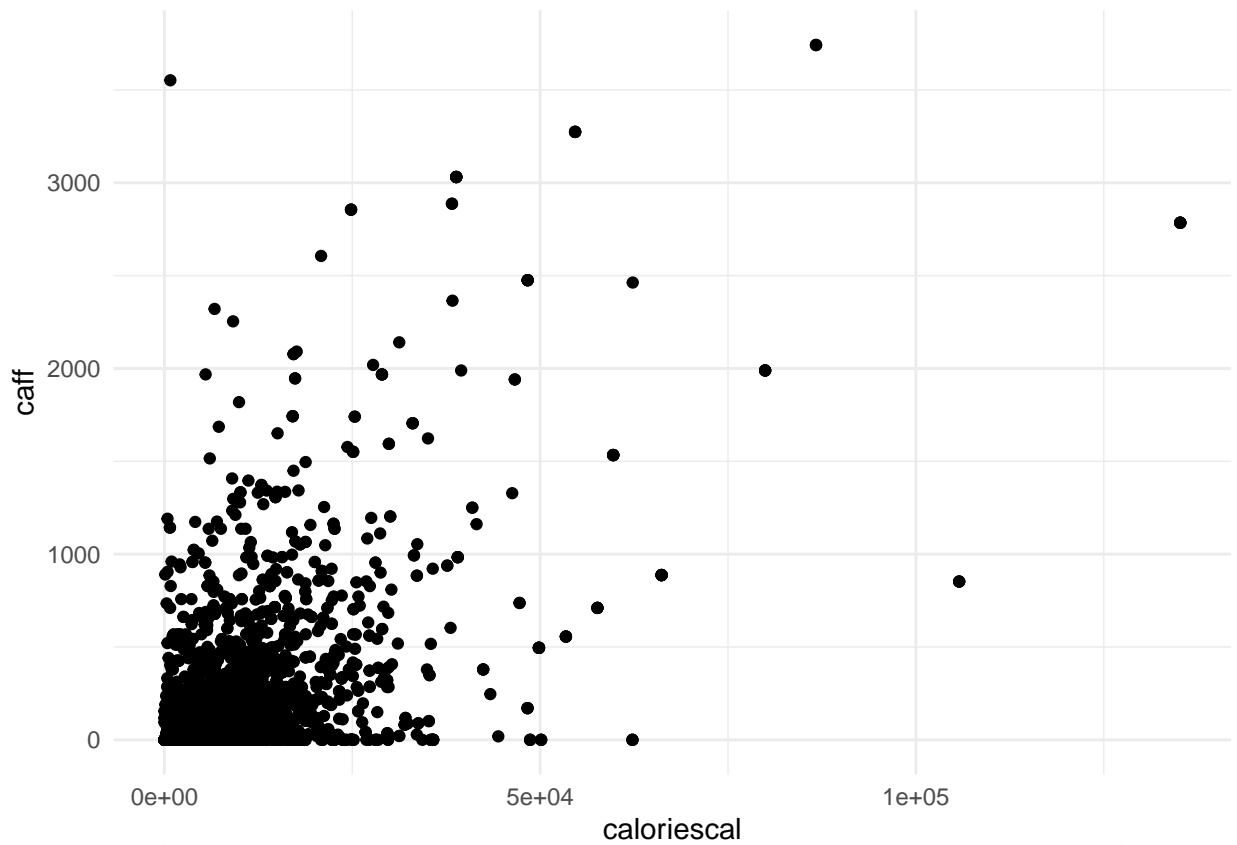


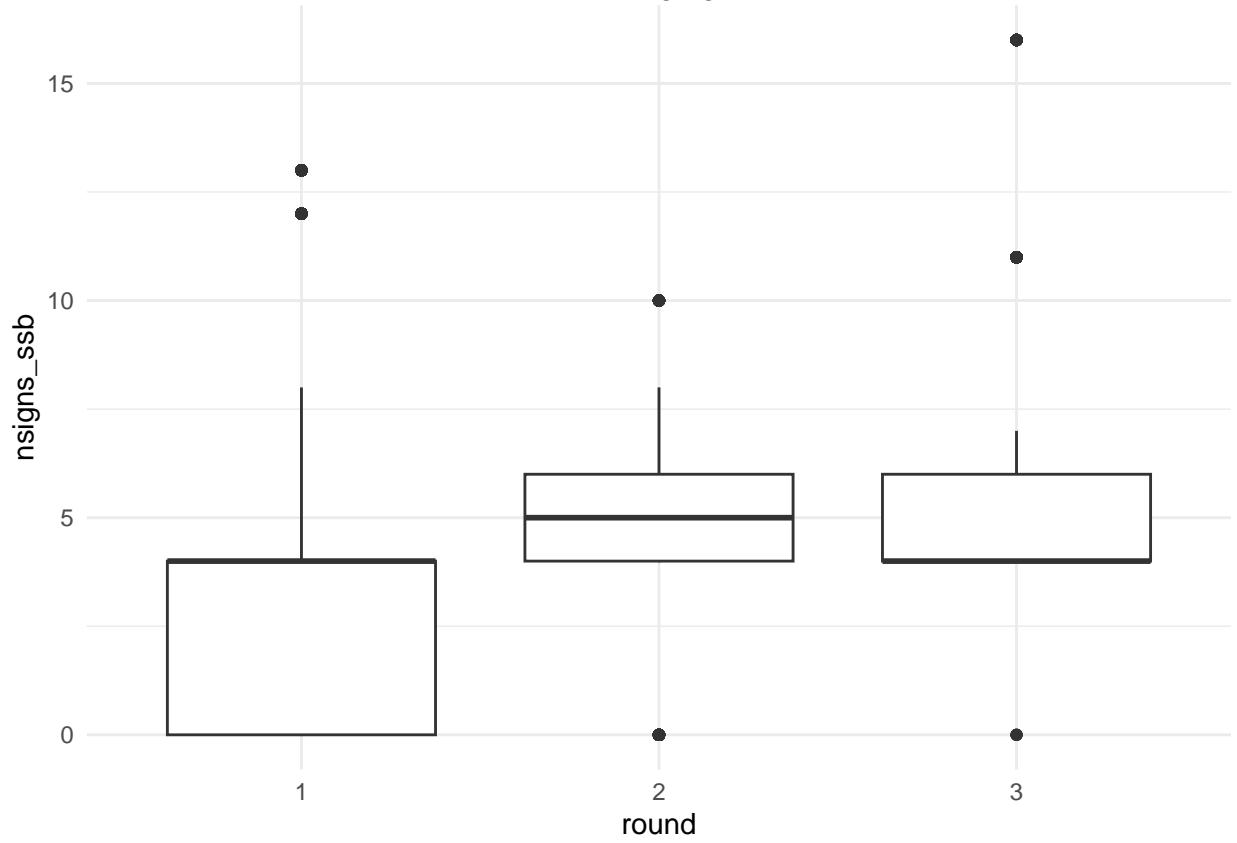
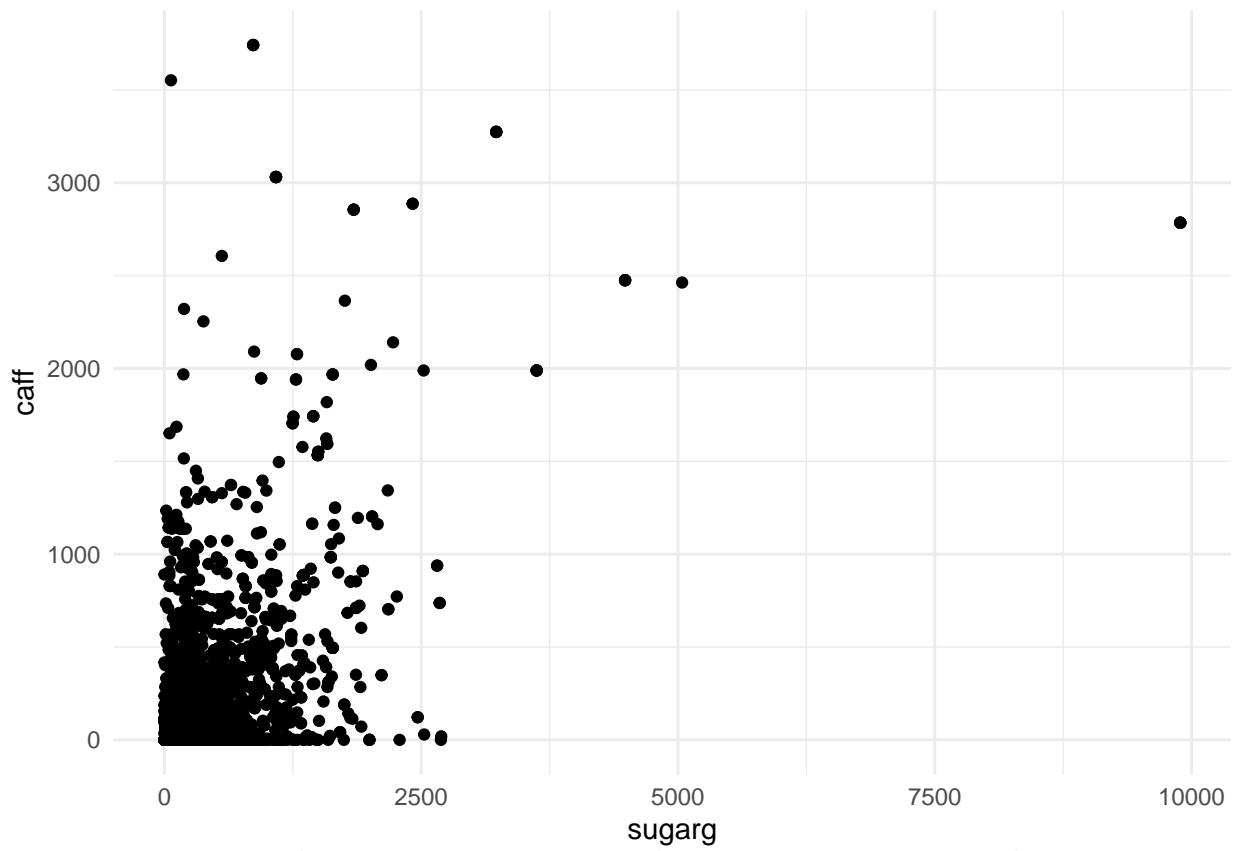


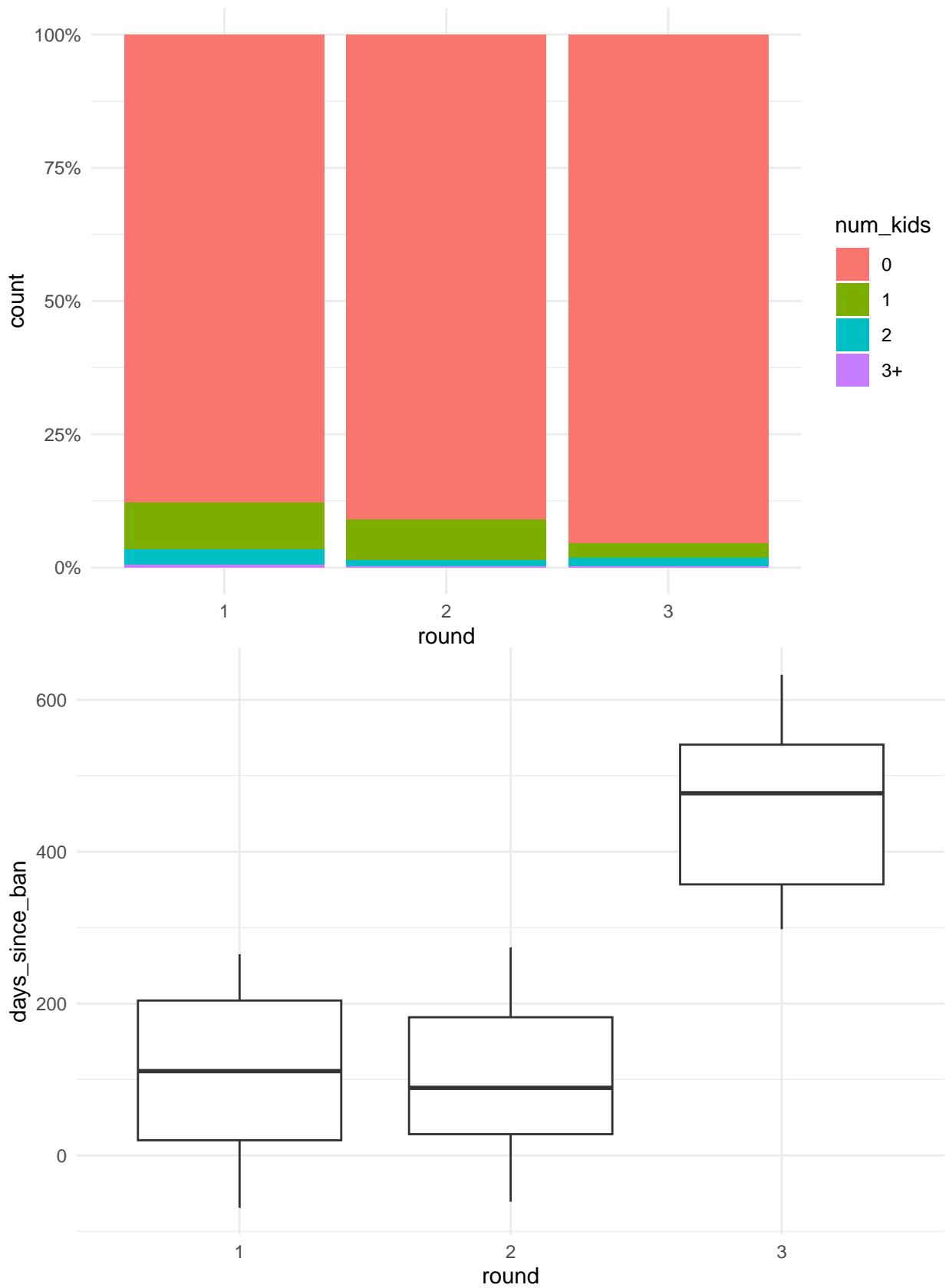


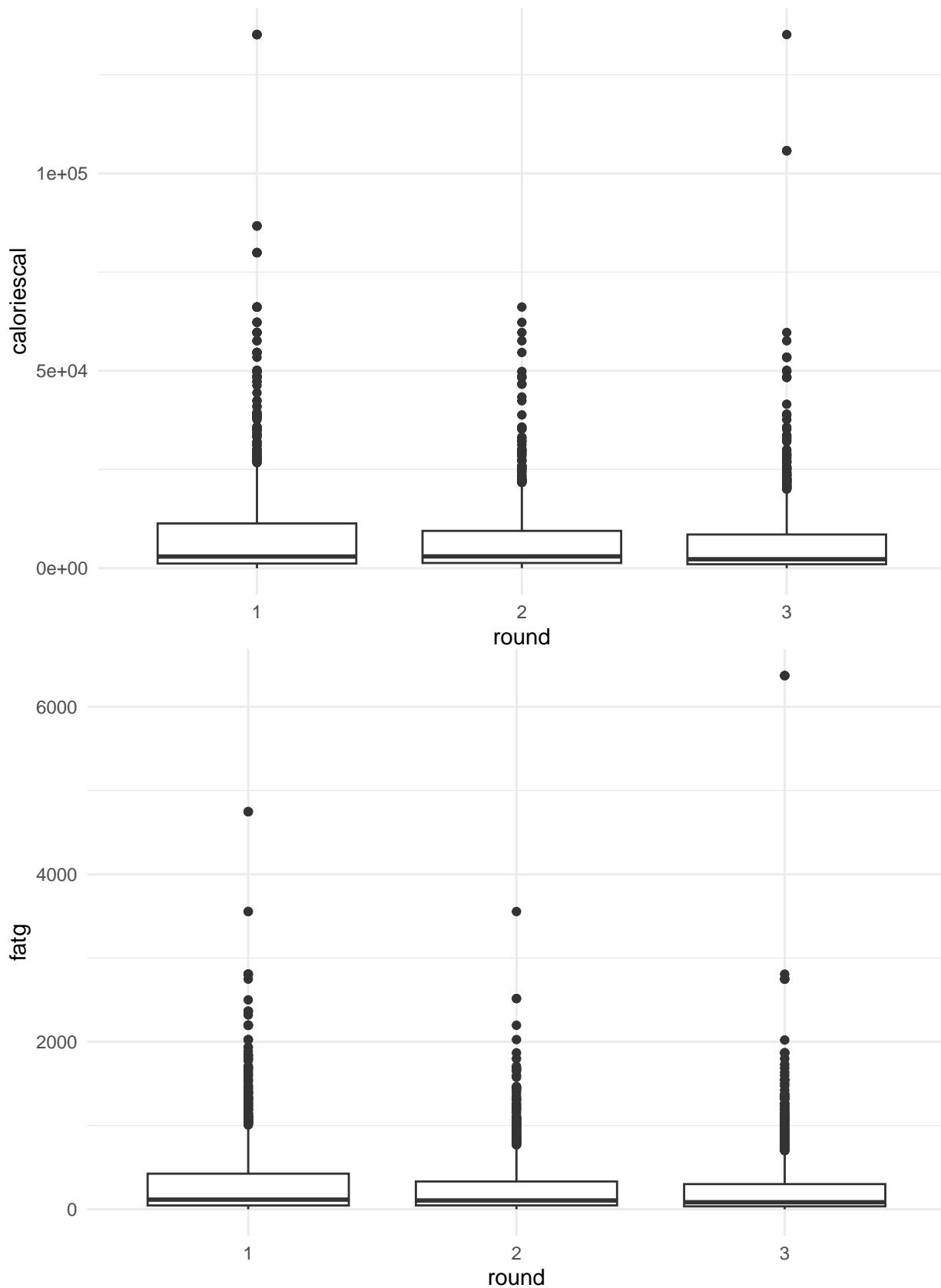


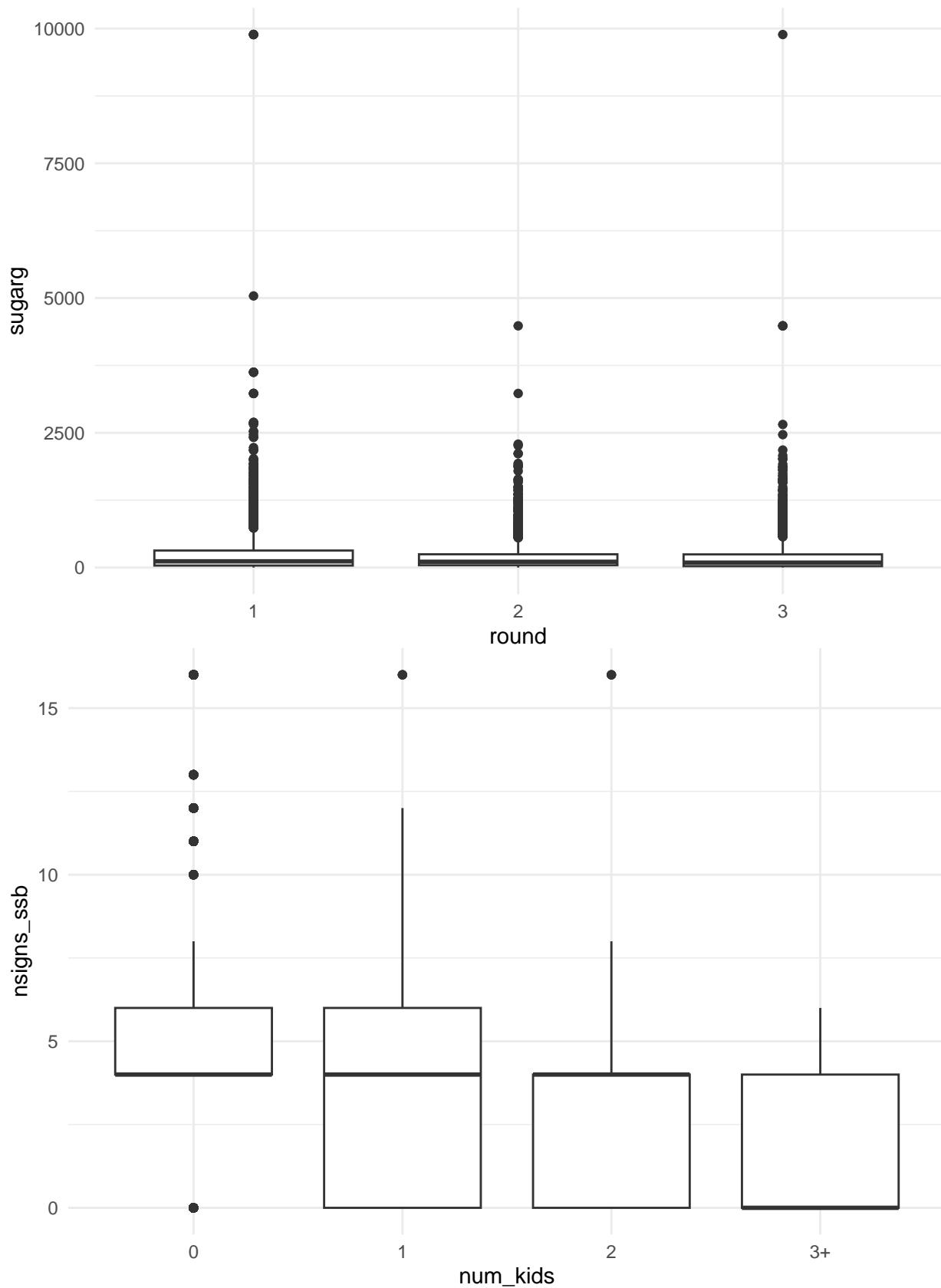


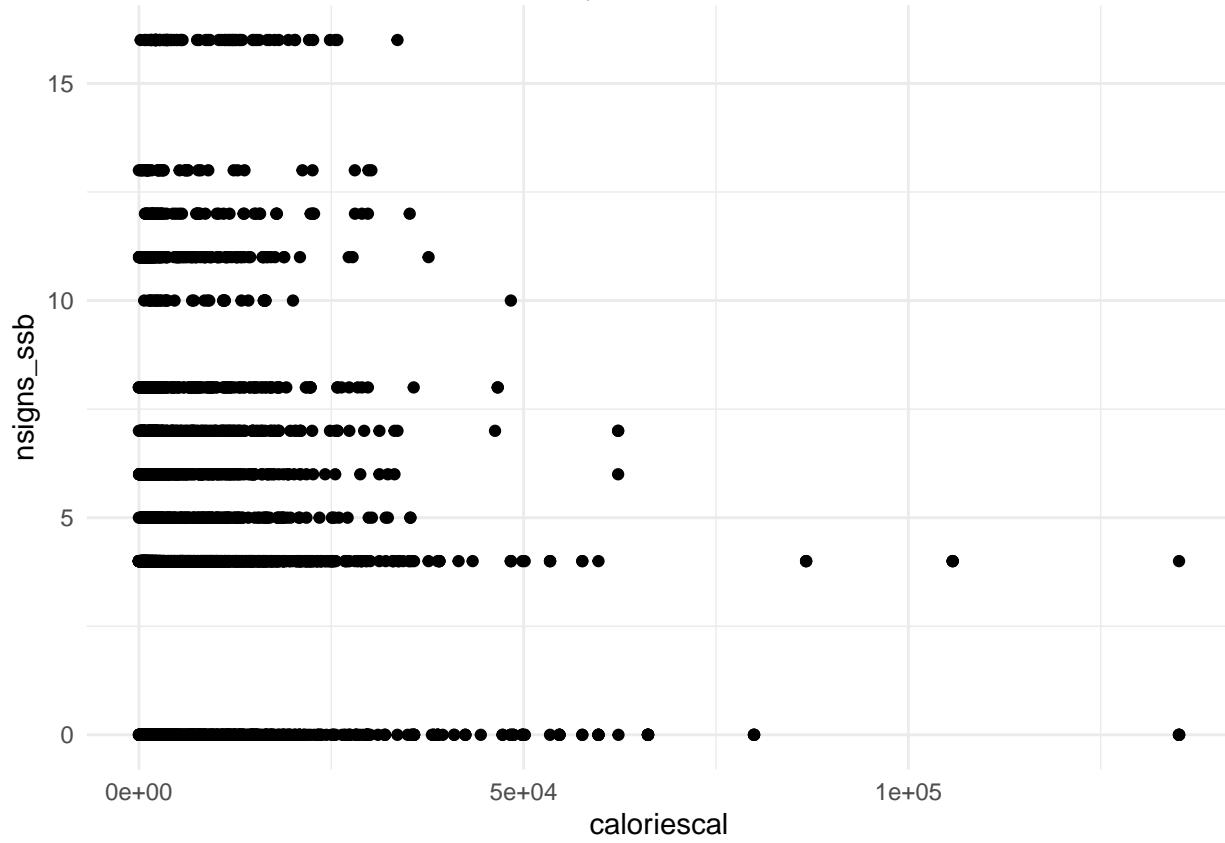
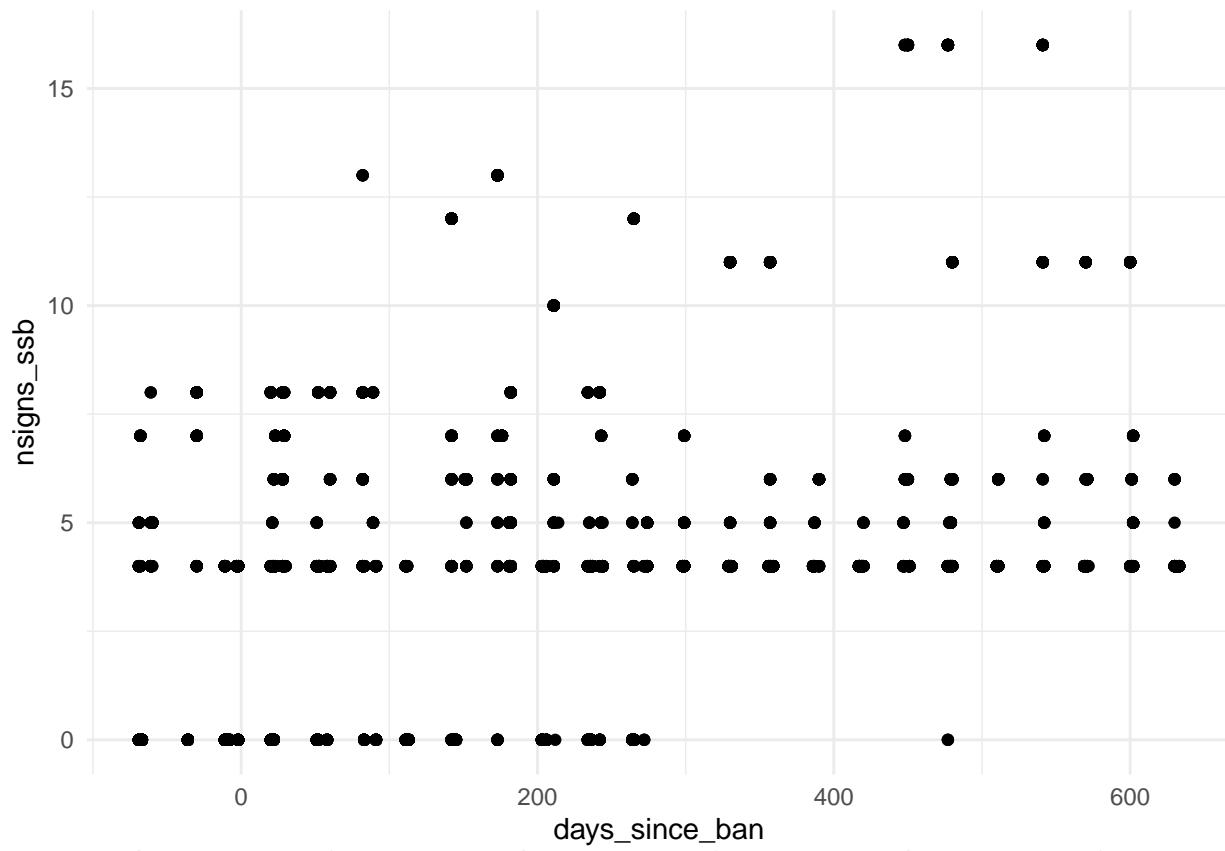


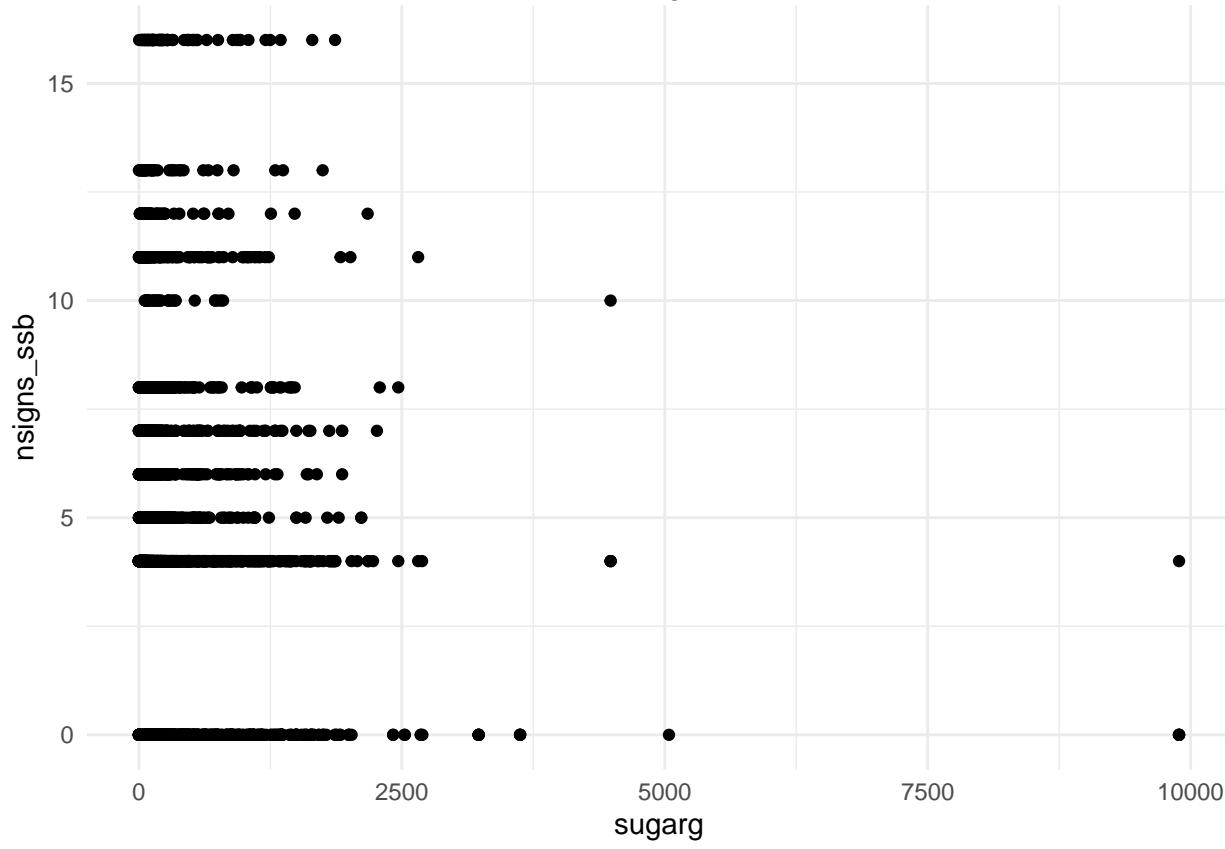
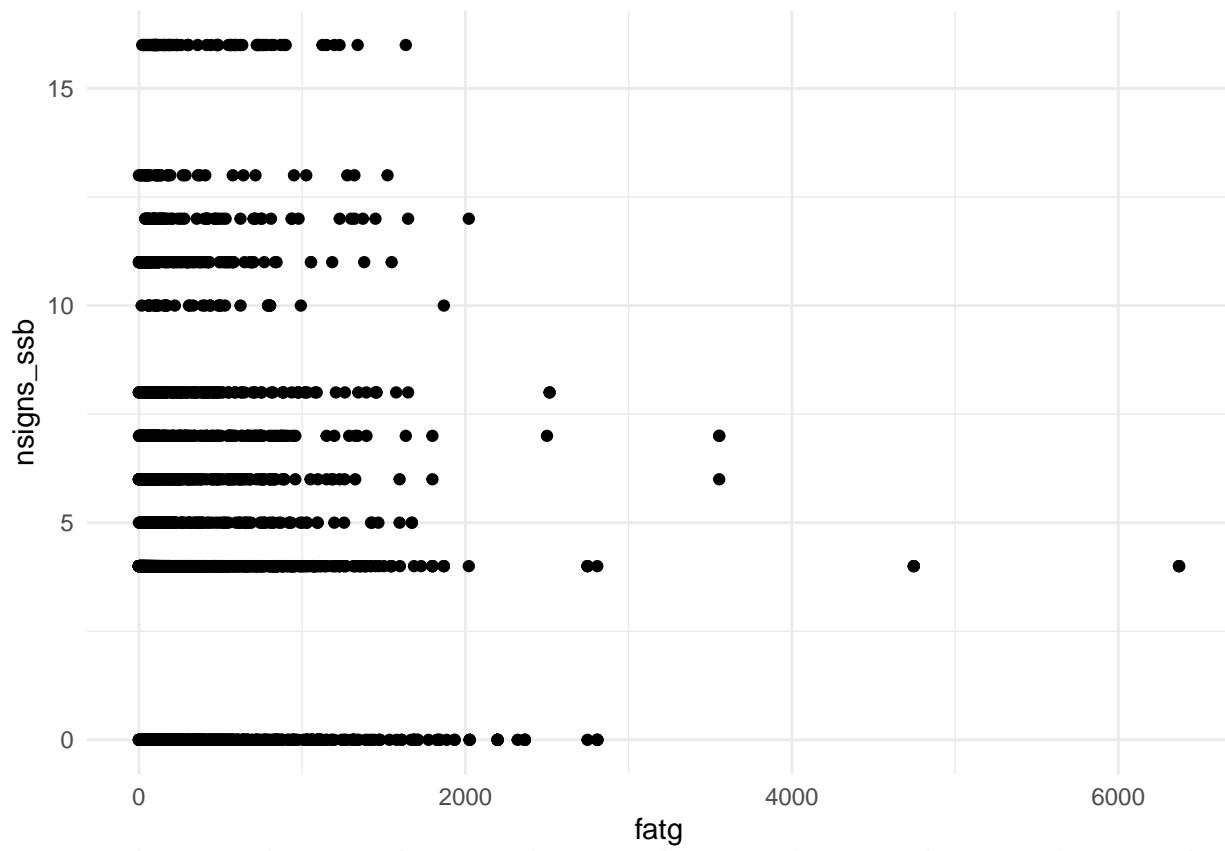


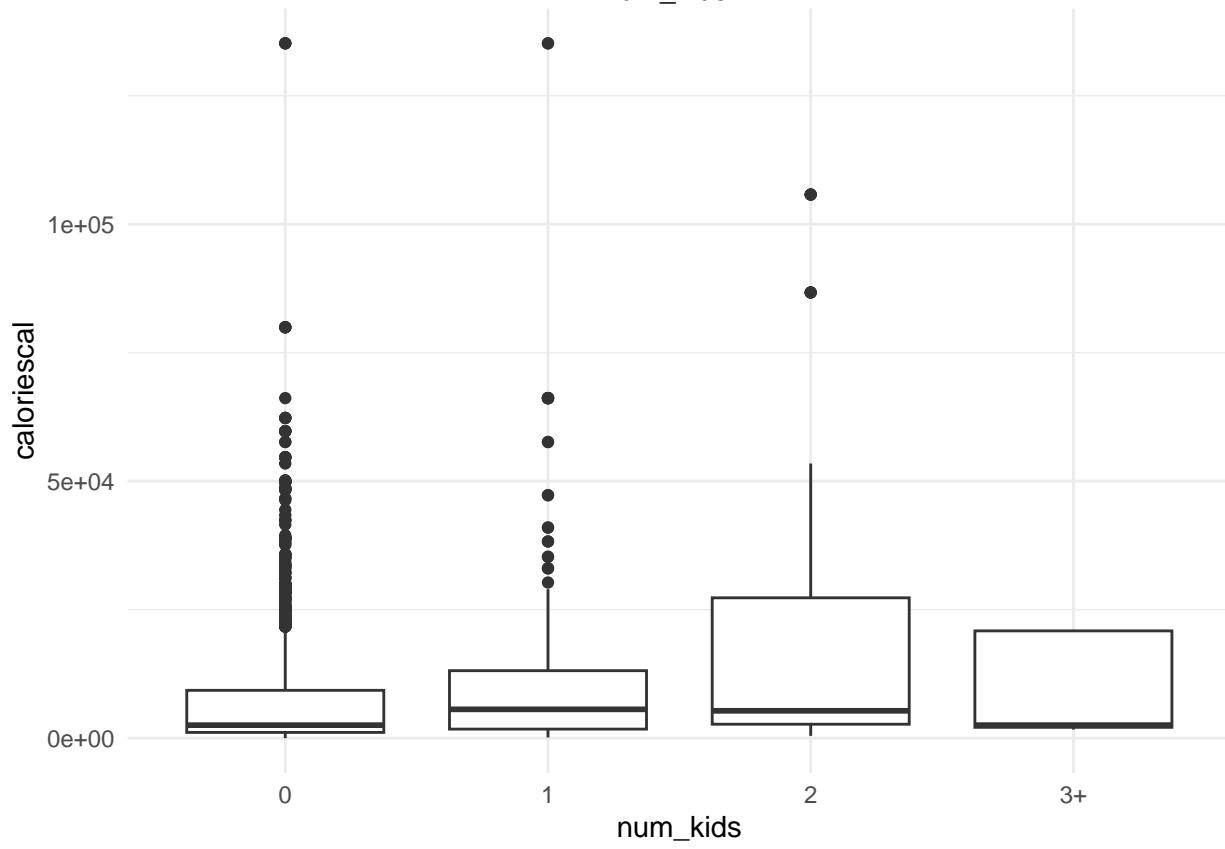
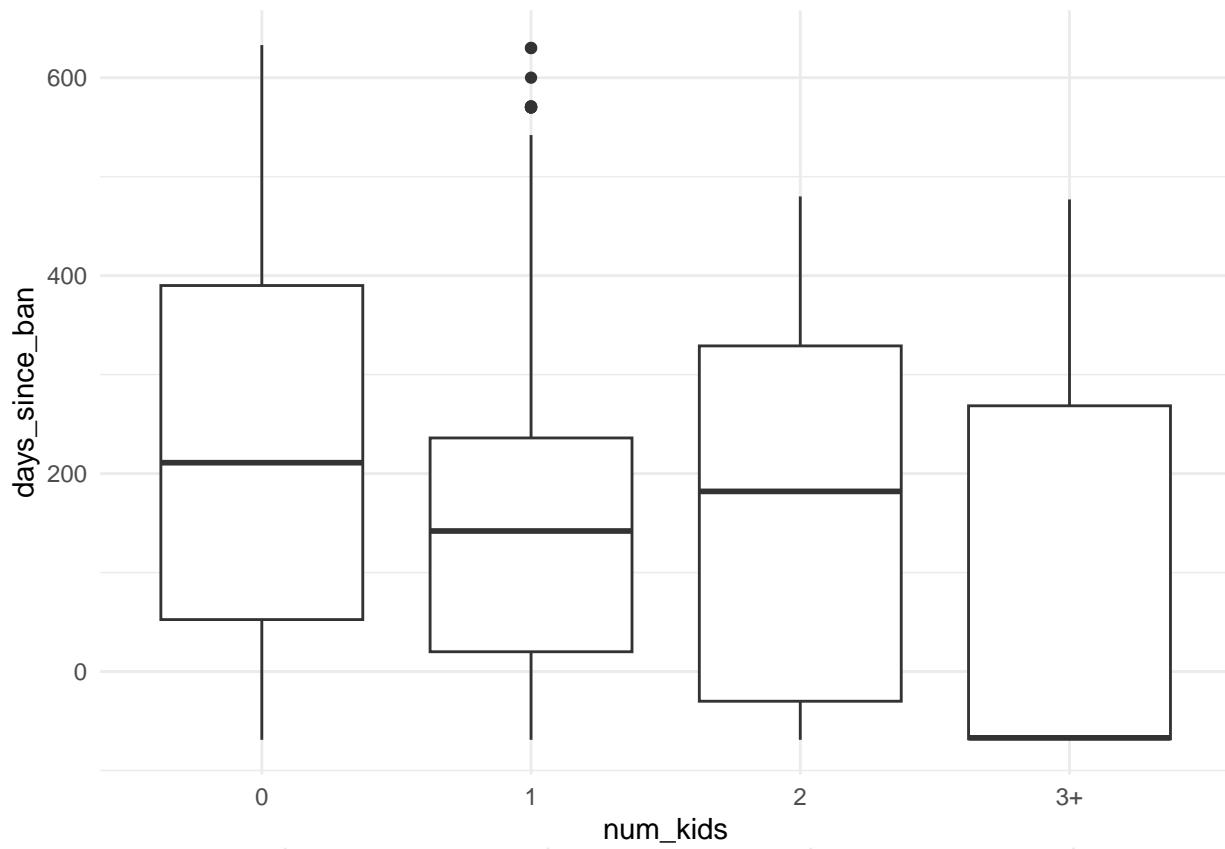


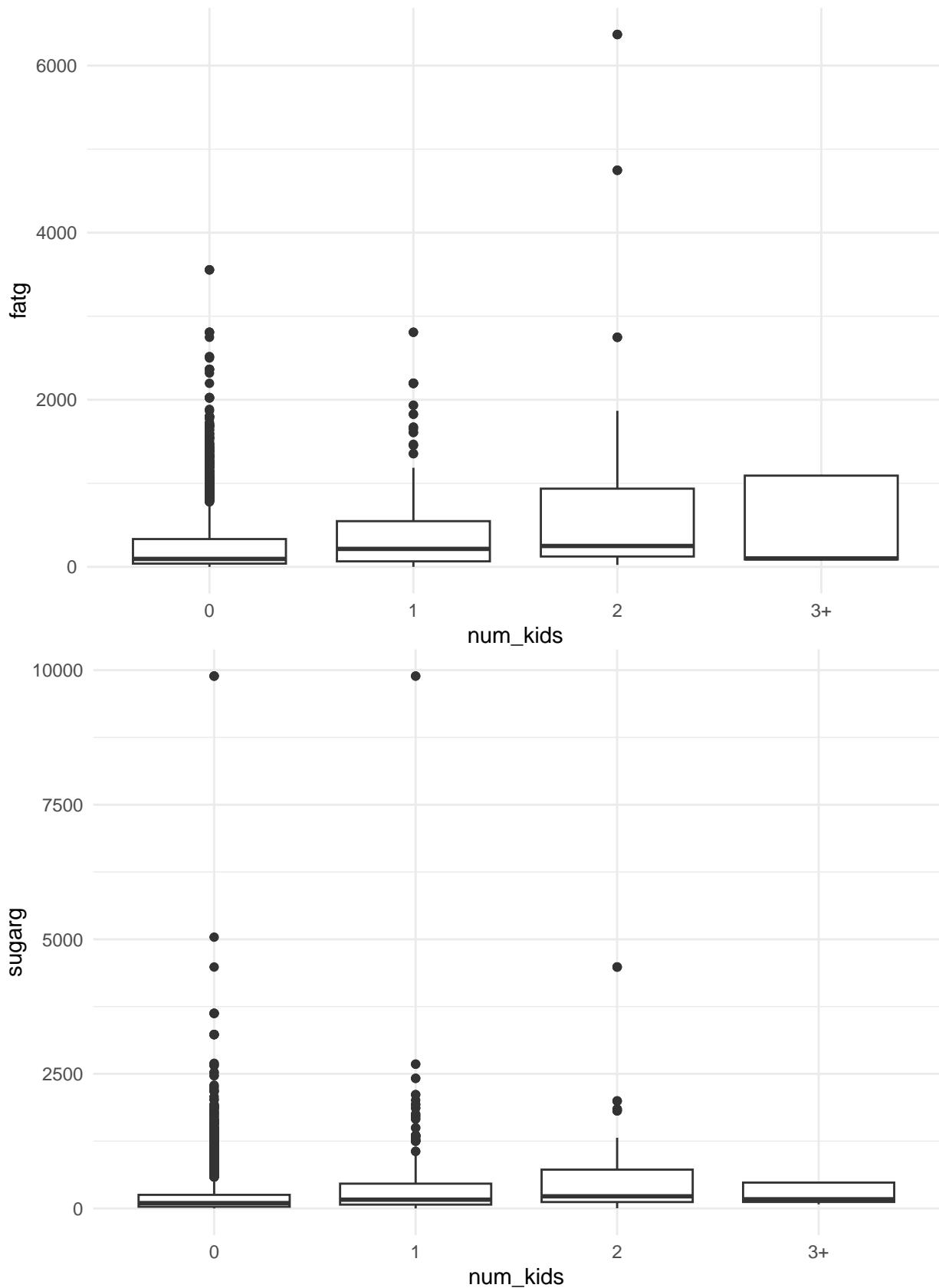


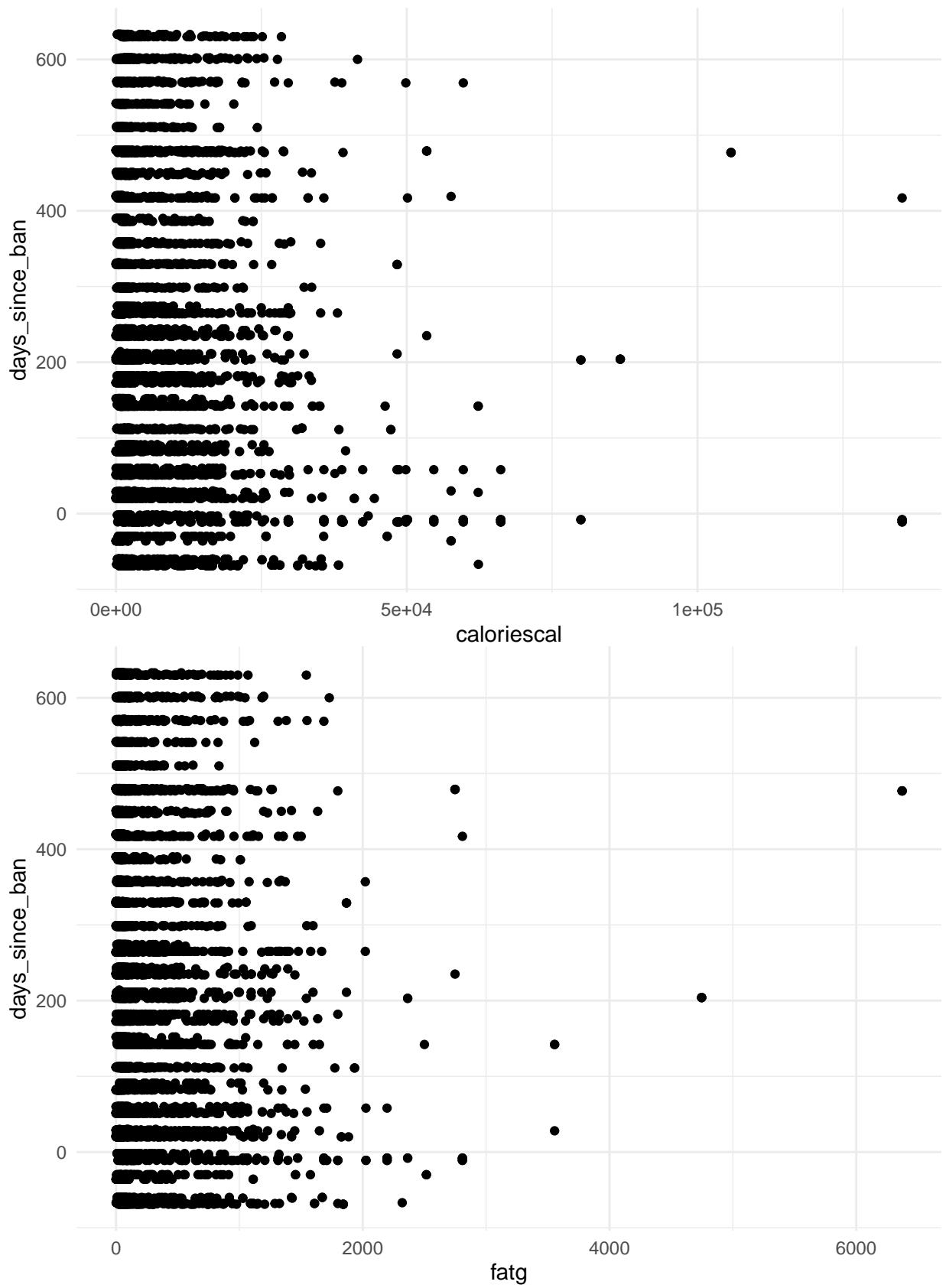


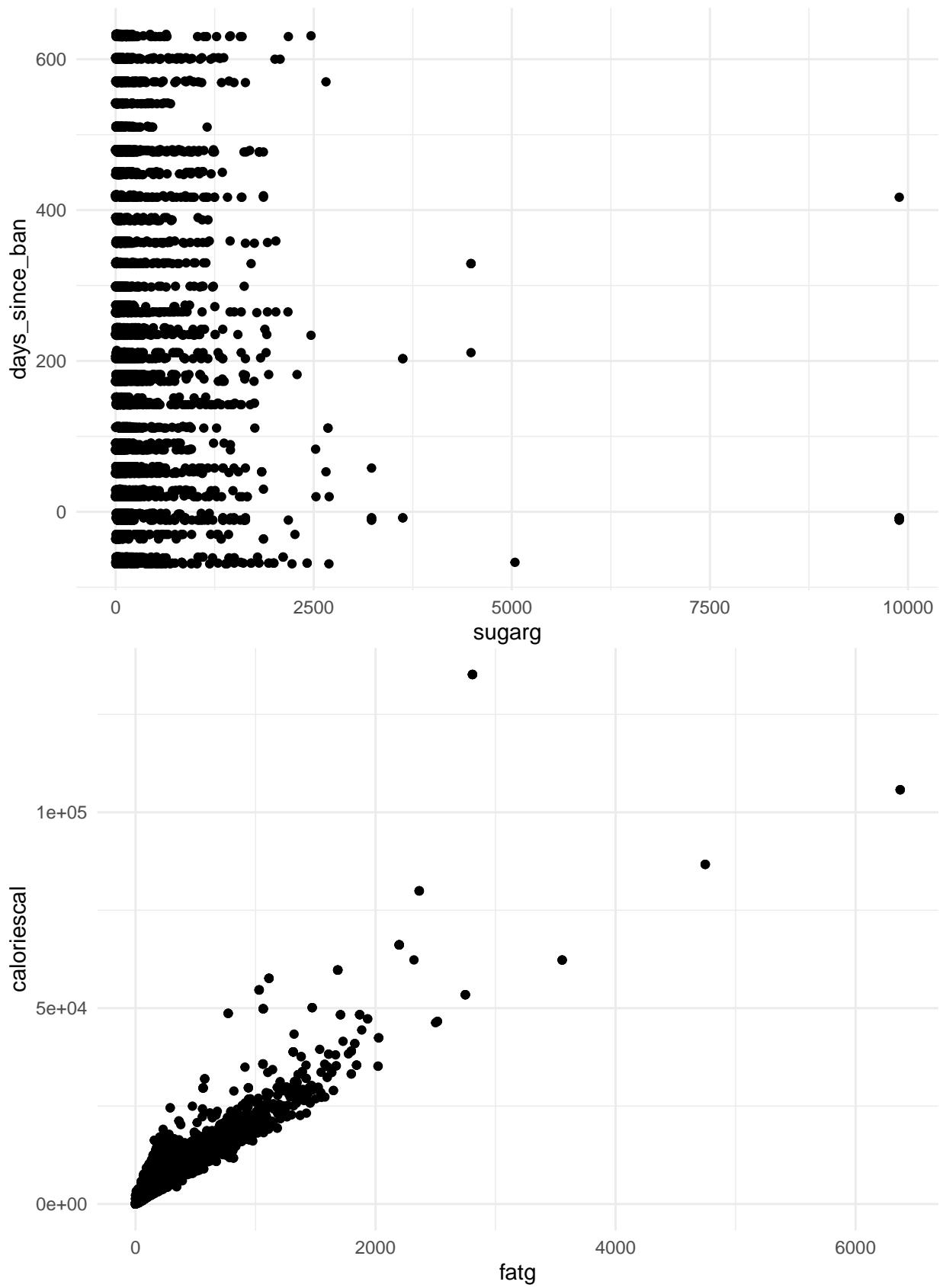


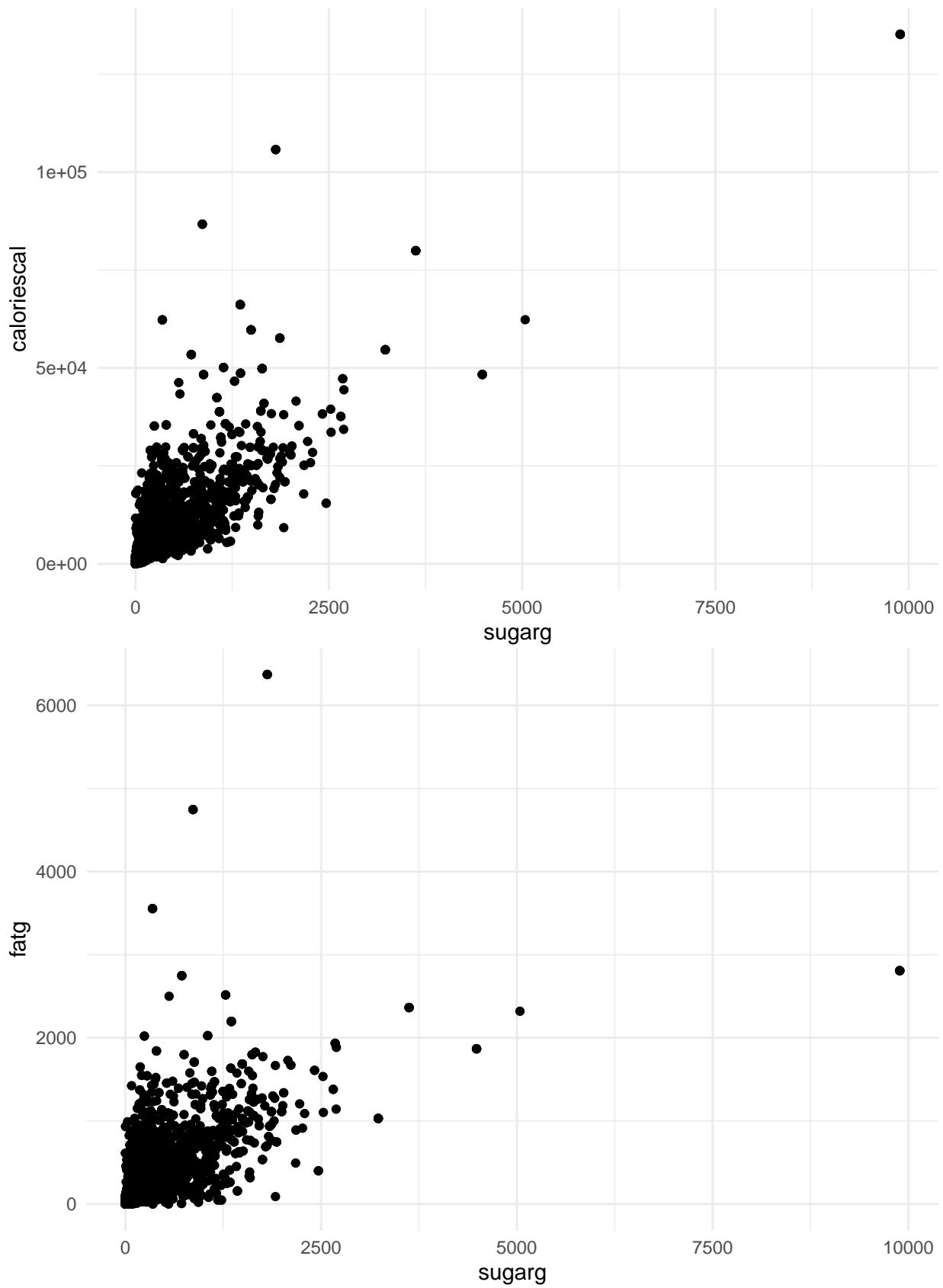












Modeling Process

Testing Different Optimization Methods

For models with no random effects, best to use Newton's approximation. For models with random effects, best to use `nlminb`, which is the default.

```
# No random effects
control_clm_full <- clm(limit ~ 1 + age + gender + race + edu + caff +
                           nsigns(ssb) + num_kids + days_since_ban,
                           data = reduced_data, control = list(
                             maxIter = 10000,
                             maxLineIter = 2000,
                             maxModIter = 2000,
                             method = "Newton",
                             trace = 1))
control_clm <- clm(limit ~ 1 + age + gender + race + edu + caff +
                           nsigns(ssb) + num_kids + days_since_ban, data = reduced_data, control = list(
                             method = "ucminf",
                             stepmax = 1,
                             grad = "central",
                             maxeval = 500000,
                             gradstep = c(1e-10, 1e-12),
                             trace = 1))
control_clm <- clm(limit ~ 1 + age + gender + race + edu + caff +
                           nsigns(ssb) + num_kids + days_since_ban, data = reduced_data, control = list(
                             method = "nlminb",
                             eval.max = 2000,
                             iter.max = 1500,
                             abs.tol = 1e-20,
                             trace = 1))
control_clm <- clm(limit ~ 1 + age + gender + race + edu + caff +
                           nsigns(ssb) + num_kids + days_since_ban, data = reduced_data, control = list(
                             method = "optim",
                             tmax = 100,
                             maxit = 100000,
                             type = 1,
                             ndeps = 1e-10,
                             REPORT = 1,
                             trace = 1))

## Check with alternative packages. Produced the same intercepts
control_vglm <- vglm(limit ~ 1 + age + gender + race + edu + caff +
                           nsigns(ssb) + num_kids + days_since_ban,
                           data = reduced_data, family = cumulative(parallel = TRUE))

## Random effects. Omit the rest for brevity
control_clmm_full <- clmm(limit ~ 1 + age + gender + race + edu + city + caff +
                           nsigns(ssb) + num_kids + days_since_ban +
                           (1 | location) + (1 | round),
                           control = list(method = "nlminb",
                                         useMatrix = T,
                                         maxIter = 200,
                                         gradTol = 1e-4,
                                         maxLineIter = 200,
```

```

        trace = 1),
        data = reduced_data, link = "logit")

# Same intercepts
summary(control_clm)
summary(control_vglm)
coef(control_vglm, matrix = T)

summary(control_clmm_full)
coef(control_clmm_full, matrix = T)

```

Full Model

Note that we also tested the non-standardized model. They both produced the similar conclusions. However, the non-standardized model couldn't fit properly because of the `kcal` variable. We proceeded with the standardized model for predictions.

```

control_clmm_full_std <- clmm(limit ~ 1 + age_std + gender + race + edu + city + caff_std +
                                nsigns(ssb_std + num_kids + days_since_ban_std +
                                caloriescal_std + fatg_std + sugarg_std +
                                (1 | location) + (1 | round),
                                control = list(method = "nlminb",
                                               useMatrix = T,
                                               maxIter = 200,
                                               gradTol = 1e-4,
                                               maxLineIter = 200
                                               # , trace = 1
                                               ),
                                data = reduced_data, link = "logit"))

summary(control_clmm_full_std)

```

```

## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: limit ~ 1 + age_std + gender + race + edu + city + caff_std +
##           nsigns(ssb_std + num_kids + days_since_ban_std + caloriescal_std +
##           fatg_std + sugarg_std + (1 | location) + (1 | round))
## data:     reduced_data
##
##   link threshold nobs logLik    AIC      niter      max.grad cond.H
##   logit flexible  4296 -6498.59 13053.18 6481(13170) 3.83e-03 1.6e+03
## 
## Random effects:
## Groups   Name        Variance Std.Dev.
## location (Intercept) 0.02328  0.1526
## round    (Intercept) 0.00000  0.0000
## Number of groups: location 57, round 3
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## age_std                  0.2259749  0.0296596  7.619 2.56e-14 ***
## genderM                 -0.2933068  0.0566808 -5.175 2.28e-07 ***
## raceBlack                -0.1940720  0.1185131 -1.638  0.10151
## raceNative               -0.2588840  0.2185724 -1.184  0.23624

```

```

## raceOther          0.0470979  0.1266716  0.372  0.71003
## raceWhite         -0.0452073  0.1228145 -0.368  0.71280
## eduCollege Degree 0.2579721  0.1262403  2.044  0.04100 *
## eduGraduate Degree 0.2480524  0.1462447  1.696  0.08986 .
## eduHigh School   -0.3044348  0.1217535 -2.500  0.01240 *
## eduLess than High School -0.1760818  0.2027481 -0.868  0.38513
## eduSome College    -0.0670136  0.1251944 -0.535  0.59246
## eduSome High School -0.4089046  0.1461276 -2.798  0.00514 **
## cityNew York        0.1652177  0.0735426  2.247  0.02467 *
## caff_std            -0.0220947  0.0368587 -0.599  0.54888
## nsigns_ssbb_std     0.0679932  0.0383321  1.774  0.07610 .
## num_kids1           0.0732200  0.1159336  0.632  0.52767
## num_kids2           -0.2779861  0.2079258 -1.337  0.18124
## num_kids3+          -1.4221950  0.5391322 -2.638  0.00834 **
## days_since_ban_std -0.0003282  0.0304117 -0.011  0.99139
## caloriescal_std     0.2954583  0.1209173  2.443  0.01455 *
## fatg_std            -0.1628537  0.0902732 -1.804  0.07123 .
## sugarg_std          -0.1083325  0.0567918 -1.908  0.05645 .

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                               Estimate Std. Error z value
## Never|Seldom      -1.1865    0.1641 -7.230
## Seldom|Sometimes  -0.7759    0.1635 -4.745
## Sometimes|Often    0.2743    0.1630  1.683
## Often|Always       1.1073    0.1640  6.751

## Non-standardized model
# control_clmm_full_non <- clmm(limit ~ 1 + age + gender + race + edu + city + caff +
# nsigns_ssbb + num_kids + days_since_ban + kcal + fv +
# (1 | location) + (1 | round),
#                                     control = list(method = "nlminb",
#                                                 useMatrix = T,
#                                                 maxIter = 200,
#                                                 gradTol = 1e-4,
#                                                 maxLineIter = 200,
#                                                 trace = 1),
#                                     data = reduced_data, link = "logit")

```

Fixed Effects

```

control_clmm_red <- clmm(limit ~ 1 + age_std + gender + race + edu + city +
                           num_kids +
                           (1 | location) + (1 | round),
                           data = reduced_data, link = "logit")
anova(control_clmm_red, control_clmm_full_std)

## Likelihood ratio tests of cumulative link models:
## formula:
## control_clmm_red      limit ~ 1 + age_std + gender + race + edu + city + num_kids + (1 | location) +
## control_clmm_full_std limit ~ 1 + age_std + gender + race + edu + city + caff_std + nsigns_ssbb_std +
##                         link: threshold:

```

```

## control_clmm_red      logit flexible
## control_clmm_full_std logit flexible
##
##           no.par    AIC  logLik LR.stat df Pr(>Chisq)
## control_clmm_red       22 13052 -6504.0
## control_clmm_full_std  28 13053 -6498.6 10.854  6   0.09301 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
summary(control_clmm_red)

## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: limit ~ 1 + age_std + gender + race + edu + city + num_kids +
##           (1 | location) + (1 | round)
## data:     reduced_data
##
## link threshold nobs logLik   AIC      niter      max.grad cond.H
## logit flexible  4296 -6504.02 13052.03 5022(10163) 5.34e-03 1.6e+03
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## location (Intercept) 2.309e-02 1.520e-01
## round     (Intercept) 1.743e-11 4.175e-06
## Number of groups: location 57, round 3
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## age_std      0.22618  0.02964  7.630 2.35e-14 ***
## genderM     -0.28895  0.05662 -5.103 3.34e-07 ***
## raceBlack    -0.18659  0.11856 -1.574  0.11553
## raceNative   -0.24327  0.21806 -1.116  0.26458
## raceOther     0.04228  0.12649  0.334  0.73817
## raceWhite    -0.04580  0.12264 -0.373  0.70884
## eduCollege Degree  0.27544  0.12607  2.185  0.02890 *
## eduGraduate Degree  0.25296  0.14599  1.733  0.08316 .
## eduHigh School -0.30598  0.12168 -2.515  0.01192 *
## eduLess than High School -0.18646  0.20259 -0.920  0.35739
## eduSome College   -0.06847  0.12512 -0.547  0.58420
## eduSome High School -0.41383  0.14602 -2.834  0.00460 **
## cityNew York     0.13928  0.07279  1.914  0.05568 .
## num_kids1        0.07677  0.11518  0.667  0.50508
## num_kids2        -0.28274  0.20261 -1.395  0.16287
## num_kids3+       -1.43678  0.53843 -2.668  0.00762 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##             Estimate Std. Error z value
## Never|Seldom   -1.1827   0.1640 -7.211
## Seldom|Sometimes -0.7726   0.1634 -4.727
## Sometimes|Often   0.2750   0.1629  1.688
## Often|Always     1.1059   0.1640  6.743

```

Random Effects

Note that we couldn't perform bootstrap because the `simulate` command is not implemented in `ordinal`, but the effects are fairly marginal and not significant.

```
control_clmm_loc <- clmm(limit ~ 1 + age_std + gender + race + edu + city +
                           num_kids +
                           (1 | location),
                           data = reduced_data, link = "logit")

lrt_obs_round <- as.numeric(2*(logLik(control_clmm_red) -
                                    logLik(control_clmm_loc)))
.5*(1 - pchisq(lrt_obs_round, 0)) + .5*(1 - pchisq(lrt_obs_round, 1))
```

Level 3 Random Intercept

```
## [1] 0.496914
```

```
control_clm <- clm(limit ~ 1 + age_std + gender + race + edu + city +
                      num_kids,
                      data = reduced_data, link = "logit")
lrt_obs_loc <- as.numeric(2*(logLik(control_clmm_loc) - logLik(control_clm)))
.5*(1 - pchisq(lrt_obs_loc, 0)) + .5*(1 - pchisq(lrt_obs_loc, 1))
```

Level 2 Random Intercept

```
## [1] 0.000985921
```

```
summary(control_clmm_loc)
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: limit ~ 1 + age_std + gender + race + edu + city + num_kids +
##           (1 | location)
## data:     reduced_data
##
##   link threshold nobs logLik    AIC      niter      max.grad cond.H
##   logit flexible  4296 -6504.02 13050.03 3688(7515) 4.27e-03 1.6e+03
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   location (Intercept) 0.02309  0.152
##   Number of groups: location 57
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
##   age_std       0.22617  0.02964  7.630 2.35e-14 ***
##   genderM      -0.28895  0.05663 -5.103 3.35e-07 ***
##   raceBlack    -0.18656  0.11859 -1.573  0.11568
##   raceNative   -0.24324  0.21811 -1.115  0.26476
##   raceOther     0.04231  0.12652  0.334  0.73806
##   raceWhite    -0.04576  0.12267 -0.373  0.70909
##   eduCollege Degree  0.27547  0.12610  2.185  0.02892 *
##   eduGraduate Degree  0.25300  0.14602  1.733  0.08317 .
```

```

## eduHigh School      -0.30595   0.12171  -2.514  0.01194 *
## eduLess than High School -0.18644   0.20260  -0.920  0.35744
## eduSome College     -0.06844   0.12515  -0.547  0.58445
## eduSome High School -0.41381   0.14605  -2.833  0.00461 **
## cityNew York        0.13928   0.07279   1.913  0.05569 .
## num_kids1            0.07678   0.11519   0.667  0.50508
## num_kids2            -0.28275   0.20261  -1.396  0.16285
## num_kids3+           -1.43677   0.53839  -2.669  0.00762 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                         Estimate Std. Error z value
## Never|Seldom       -1.1826    0.1641  -7.208
## Seldom|Sometimes   -0.7725    0.1635  -4.725
## Sometimes|Often     0.2751    0.1630   1.688
## Often|Always        1.1060    0.1641   6.741

```

Separate slopes for each level

Ordinal provides two built-in commands for testing whether we need separate slopes for predictors of each level and whether we need to scale our response by each predictors. None of them showed significance.

```

nominal_test(control_clm)

## Tests of nominal effects
##
## formula: limit ~ 1 + age_std + gender + race + edu + city + num_kids
##          Df  logLik  AIC      LRT Pr(>Chi)
## <none>   -6508.8 13058
## age_std   3 -6504.2 13054  9.1575 0.027269 *
## gender    3 -6501.7 13050 14.1251 0.002740 **
## race      12 -6501.2 13066 15.2722 0.226885
## edu       18 -6498.2 13072 21.2215 0.268386
## city      3 -6500.9 13048 15.8213 0.001234 **
## num_kids
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
scale_test(control_clm)

## Tests of scale effects
##
## formula: limit ~ 1 + age_std + gender + race + edu + city + num_kids
##          Df  logLik  AIC      LRT Pr(>Chi)
## <none>   -6508.8 13058
## age_std   1 -6505.5 13053  6.5930 0.0102379 *
## gender    1 -6506.9 13056  3.8980 0.0483442 *
## race      4 -6502.7 13054 12.1205 0.0164771 *
## edu       6 -6505.6 13063  6.4472 0.3749976
## city      1 -6503.2 13048 11.2105 0.0008134 ***
## num_kids  3 -6507.2 13060  3.1636 0.3670831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

control_clm_nom <- clm(limit ~ 1 + race + edu + num_kids,
                         nominal = ~ age_std + gender + city,
                         data = reduced_data, link = "logit")
anova(control_clm_nom, control_clm)

## Likelihood ratio tests of cumulative link models:
##
##           formula:
## control_clm      limit ~ 1 + age_std + gender + race + edu + city + num_kids
## control_clm_nom limit ~ 1 + race + edu + num_kids
##           nominal:               link: threshold:
## control_clm      ~1                  logit flexible
## control_clm_nom ~age_std + gender + city logit flexible
##
##           no.par  AIC  logLik LR.stat df Pr(>Chisq)
## control_clm      20 13058 -6508.8
## control_clm_nom 29 13035 -6488.6  40.311  9  6.674e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Overall fit Compared to the only intercept model.

```

control_null <- clm(limit ~ 1, data = reduced_data, link = "logit")

## Warning: (-1) Model failed to converge with max|grad| = 5.30767e-06 (tol = 1e-06)
## In addition: iteration limit reached
# Overall fit
anova(control_null, control_clm_nom)

## Likelihood ratio tests of cumulative link models:
##
##           formula:
## control_null      limit ~ 1
## control_clm_nom  limit ~ 1 + race + edu + num_kids + caloriescal_std
##           nominal:               link: threshold:
## control_null      ~1                  logit flexible
## control_clm_nom ~age_std + gender + city + caff_std logit flexible
##
##           no.par  AIC  logLik LR.stat df Pr(>Chisq)
## control_null      4 13243 -6617.6
## control_clm_nom  34 13029 -6480.5  274.31 30  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
control_null_re <- clmm(limit ~ 1 + (1 | location), data = reduced_data, link = "logit")

## Warning: (-1) Model failed to converge with max|grad| = 5.30767e-06 (tol = 1e-06)
## In addition: iteration limit reached
anova(control_null, control_clmm_loc)

## Likelihood ratio tests of cumulative link models:
##
##           formula:
## control_null      limit ~ 1
## control_clmm_loc limit ~ 1 + age_std + gender + race + edu + city + caff_std + num_kids + caloriescal_

```

```

##               link: threshold:
## control_null      logit flexible
## control_clmm_loc logit flexible
##
##          no.par    AIC  logLik LR.stat df Pr(>Chisq)
## control_null         4 13243 -6617.6
## control_clmm_loc     23 13051 -6502.7  229.99 19 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Model Diagnostics

Accuracy Metrics

Because residual analysis are not well understood in ordinal models, we opted for accuracy metrics. Note that our model doesn't predict well.

```

library(tidymodels)
library(workflows)

model_accuracy <- function(model = control_clm, adj = F) {
  comp_metrics <- function(model = model, predict) {
    control_results <- reduced_data %>%
      bind_cols(fit = predict)

    # Confusion matrix
    # table(control_results$limit, control_results$fit)
    conf_mat(control_results, truth = limit, estimate = fit) -> conf

    # accuracy metrics
    accuracy(control_results, truth = limit, estimate = fit) -> acc
    sensitivity(control_results, truth = limit, estimate = fit) -> sen
    specificity(control_results, truth = limit, estimate = fit) -> spe
    # ppu(control_results, truth = limit, estimate = fit)

    # Goodness of fit
    chisq.test(control_results$limit, control_results$fit) -> gof

    return(list(control_results = control_results, conf = conf, acc = acc, sen = sen, spe = spe, gof = gof))
  }
  if (adj) {
    # Predict response
    control_vglm_pred <- predict(model, type = "response")
    level_counts <- table(reduced_data$limit)
    total_counts <- sum(level_counts)
    proportions <- as.numeric(level_counts / total_counts)
    names(proportions) <- names(level_counts)

    adjusted_probs <- control_vglm_pred / proportions[colnames(control_vglm_pred)]
    adjusted_probs <- adjusted_probs / rowSums(adjusted_probs)
    fit <- ordered(colnames(adjusted_probs)[max.col(adjusted_probs)], 
                  levels = c("Never", "Seldom", "Sometimes",
                            "Often", "Always"))
    comp_metrics(model = model, predict = fit) -> result
  }
}

```

```

} else {
  # Predict response
  control_pred <- predict(model, type = "class")
  comp_metrics(model = model, control_pred) -> result
}
return(result)
}

model_accuracy(control_clm_nom)

## $control_results
## # A tibble: 4,296 x 27
##   receiptid person_id limit      age age_std gender race  edu    city    caff
##   <fct>     <fct>   <ord>     <dbl>  <dbl> <chr> <chr> <chr> <dbl>
## 1 B103-098  900563  Sometimes  30  -0.670  F    Black Assoc~ New ~ 0
## 2 B103-022  900076  Sometimes  22  -1.22   F    Other Some C~ New ~ 0
## 3 B103-091  900582  Never     40   0.0172 F    White High S~ New ~ 1137.
## 4 B103-081  900569  Never     20  -1.36   F    Other Some H~ New ~ 0
## 5 B103-080  900568  Sometimes 34  -0.395  M    Other High S~ New ~ 0
## 6 B103-090  900578  Seldom    19  -1.43   F    Other High S~ New ~ 101.
## 7 B103-024  900078  Sometimes 61   1.46   M    Black Some C~ New ~ 75.8
## 8 B103-067  900021  Always    51   0.773  F    Black Colleg~ New ~ 0
## 9 B103-086  900574  Never     58   1.25   F    Black Some C~ New ~ 0
## 10 B103-023  900077  Never    50   0.704  F    Black Colleg~ New ~ 0
## # i 4,286 more rows
## # i 17 more variables: location <fct>, round <fct>, nsigns(ssb) <dbl>,
## # num_kids <chr>, surveydate <date>, days_since_ban <dbl>, caloriescal <dbl>,
## # fatg <dbl>, sugarg <dbl>, black <chr>, caff_std <dbl>,
## # nsigns(ssb_std) <dbl>, days_since_ban_std <dbl>, caloriescal_std <dbl>,
## # fatg_std <dbl>, sugarg_std <dbl>, fit <fct>
##
## $conf
##           Truth
## Prediction Never Seldom Sometimes Often Always
##   Never      893   234      591    358    436
##   Seldom      0     0       0     0     0
##   Sometimes   193   70      250    158    204
##   Often       0     0       0     0     0
##   Always      187   73      213    179    257
##
## $acc
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>        <dbl>
## 1 accuracy multiclass    0.326
##
## $sen
## # A tibble: 1 x 3
##   .metric      .estimator .estimate
##   <chr>        <chr>        <dbl>
## 1 sensitivity macro      0.245
##
## $spe
## # A tibble: 1 x 3

```

```

##   .metric   .estimator .estimate
##   <chr>     <chr>       <dbl>
## 1 specificity macro          0.816
##
## $gof
##
## Pearson's Chi-squared test
##
## data: control_results$limit and control_results$fit
## X-squared = 136.94, df = 8, p-value < 2.2e-16
## Similar results under different model specifications
control_clm_probit <- clm(limit ~ 1 + age_std + gender + race + edu + city +
                           num_kids,
                           data = reduced_data, link = "probit")
model_accuracy(control_clm_probit)

## $control_results
## # A tibble: 4,296 x 27
##   receiptid person_id limit      age age_std gender race  edu    city    caff
##   <fct>     <fct>    <ord>     <dbl>   <dbl> <chr>  <chr> <chr>  <dbl>
## 1 B103-098  900563  Sometimes  30  -0.670  F     Black Assoc~ New ~ 0
## 2 B103-022  900076  Sometimes  22  -1.22   F     Other Some C~ New ~ 0
## 3 B103-091  900582  Never    40   0.0172 F     White High S~ New ~ 1137.
## 4 B103-081  900569  Never    20  -1.36   F     Other Some H~ New ~ 0
## 5 B103-080  900568  Sometimes 34  -0.395  M     Other High S~ New ~ 0
## 6 B103-090  900578  Seldom   19  -1.43   F     Other High S~ New ~ 101.
## 7 B103-024  900078  Sometimes 61   1.46   M     Black Some C~ New ~ 75.8
## 8 B103-067  900021  Always   51   0.773  F     Black Colleg~ New ~ 0
## 9 B103-086  900574  Never    58   1.25   F     Black Some C~ New ~ 0
## 10 B103-023 900077  Never   50   0.704  F     Black Colleg~ New ~ 0
## # i 4,286 more rows
## # i 17 more variables: location <fct>, round <fct>, nsigns(ssb) <dbl>,
## #   num_kids <chr>, surveydate <date>, days_since_ban <dbl>, caloriescal <dbl>,
## #   fatg <dbl>, sugarg <dbl>, black <chr>, caff_std <dbl>,
## #   nsigns(ssb_std) <dbl>, days_since_ban_std <dbl>, caloriescal_std <dbl>,
## #   fatg_std <dbl>, sugarg_std <dbl>, fit <fct>
##
## $conf
##           Truth
## Prediction  Never Seldom Sometimes Often Always
##   Never      991   260      701   437   499
##   Seldom      0     0       0     0     0
##   Sometimes   53    23      67    49    54
##   Often       0     0       0     0     0
##   Always     229   94      286   209   344
##
## $acc
## # A tibble: 1 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>       <dbl>
## 1 accuracy multiclass      0.326
##
## $sen
## # A tibble: 1 x 3

```

```

##   .metric      .estimator .estimate
##   <chr>        <chr>       <dbl>
## 1 sensitivity macro          0.245
##
## $spe
## # A tibble: 1 x 3
##   .metric      .estimator .estimate
##   <chr>        <chr>       <dbl>
## 1 specificity macro          0.815
##
## $gof
##
## Pearson's Chi-squared test
##
## data: control_results$limit and control_results$fit
## X-squared = 133.99, df = 8, p-value < 2.2e-16
control_clm_sym <- clm(limit ~ 1 + age_std + gender + race + edu + city +
                         num_kids,
                         data = reduced_data,
                         link = "probit", threshold = "equidistant")
model_accuracy(control_clm_sym)

## $control_results
## # A tibble: 4,296 x 27
##   receiptid person_id limit      age age_std gender race  edu    city    caff
##   <fct>     <fct>    <ord>     <dbl>   <dbl> <chr>  <chr> <chr>  <dbl>
## 1 B103-098  900563  Sometimes  30  -0.670  F     Black Assoc~ New ~  0
## 2 B103-022  900076  Sometimes  22  -1.22   F     Other Some C~ New ~  0
## 3 B103-091  900582  Never    40   0.0172  F     White High S~ New ~ 1137.
## 4 B103-081  900569  Never    20  -1.36   F     Other Some H~ New ~  0
## 5 B103-080  900568  Sometimes 34  -0.395  M     Other High S~ New ~  0
## 6 B103-090  900578  Seldom   19  -1.43   F     Other High S~ New ~ 101.
## 7 B103-024  900078  Sometimes 61   1.46   M     Black Some C~ New ~ 75.8
## 8 B103-067  900021  Always   51   0.773  F     Black Colleg~ New ~  0
## 9 B103-086  900574  Never    58   1.25   F     Black Some C~ New ~  0
## 10 B103-023 900077  Never   50   0.704  F     Black Colleg~ New ~  0
## # i 4,286 more rows
## # i 17 more variables: location <fct>, round <fct>, nsigns(ssb) <dbl>,
## #   num_kids <chr>, surveydate <date>, days_since_ban <dbl>, caloriescal <dbl>,
## #   fatg <dbl>, sugarg <dbl>, black <chr>, caff_std <dbl>,
## #   nsigns(ssb_std) <dbl>, days_since_ban_std <dbl>, caloriescal_std <dbl>,
## #   fatg_std <dbl>, sugarg_std <dbl>, fit <fct>
##
## $conf
##   Truth
## Prediction Never Seldom Sometimes Often Always
##   Never      951   248     656    403    470
##   Seldom      0      0      0      0      0
##   Sometimes   0      0      0      0      0
##   Often       0      0      0      0      0
##   Always      322   129     398    292    427
##
## $acc
## # A tibble: 1 x 3

```

```

##   .metric   .estimator .estimate
##   <chr>     <chr>          <dbl>
## 1 accuracy multiclass      0.321
##
## $sen
## # A tibble: 1 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>          <dbl>
## 1 sensitivity macro      0.245
##
## $spe
## # A tibble: 1 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>          <dbl>
## 1 specificity macro      0.815
##
## $gof
##
## Pearson's Chi-squared test
##
## data: control_results$limit and control_results$fit
## X-squared = 127.36, df = 4, p-value < 2.2e-16
## Use VGAM to get prob for each level of resp, not implemented in Ordinal
## Similarly inaccurate model
control_vglm_sig <- vglm(limit ~ 1 + age_std + gender + race + edu + city +
                           num_kids,
                           data = reduced_data,
                           family = cumulative(parallel = TRUE))
model_accuracy(control_vglm_sig, adj = T)

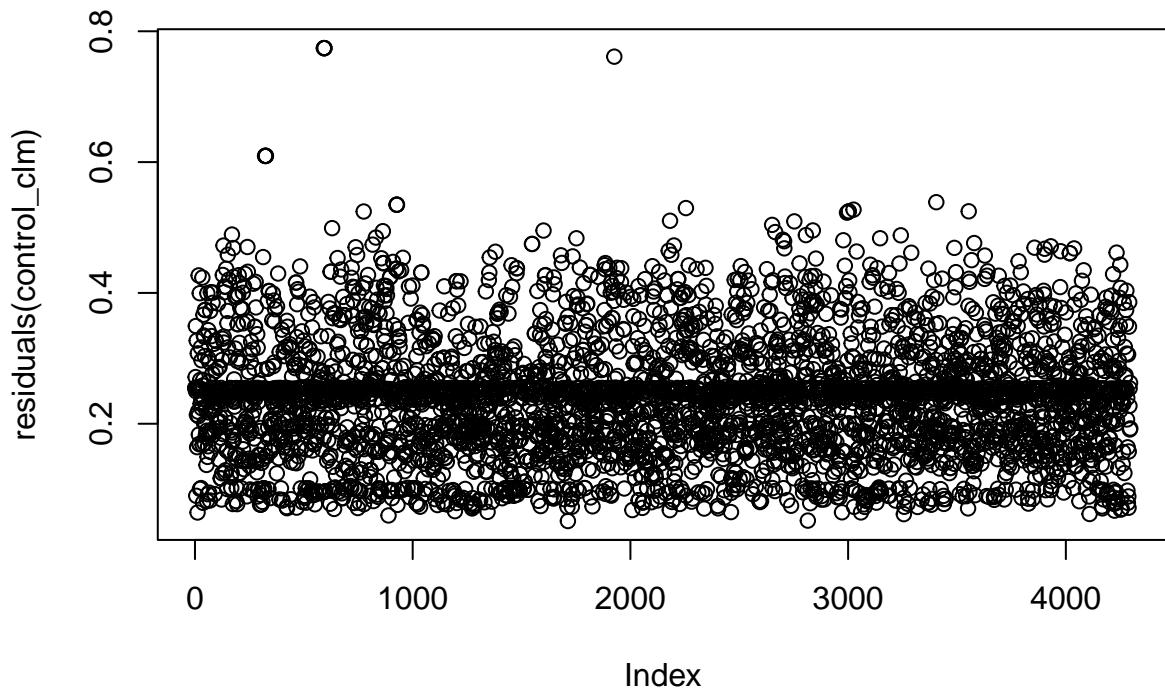
## $control_results
## # A tibble: 4,296 x 27
##   receiptid person_id limit      age age_std gender race  edu    city    caff
##   <fct>     <fct>    <ord>     <dbl>   <dbl>  <chr> <chr> <chr>  <dbl>
## 1 B103-098  900563  Sometimes  30  -0.670  F     Black Assoc~ New ~  0
## 2 B103-022  900076  Sometimes  22  -1.22   F     Other Some C~ New ~  0
## 3 B103-091  900582  Never    40   0.0172  F     White High S~ New ~ 1137.
## 4 B103-081  900569  Never    20  -1.36   F     Other Some H~ New ~  0
## 5 B103-080  900568  Sometimes 34  -0.395  M     Other High S~ New ~  0
## 6 B103-090  900578  Seldom   19  -1.43   F     Other High S~ New ~ 101.
## 7 B103-024  900078  Sometimes 61   1.46    M     Black Some C~ New ~ 75.8
## 8 B103-067  900021  Always   51   0.773   F     Black Colleg~ New ~  0
## 9 B103-086  900574  Never    58   1.25    F     Black Some C~ New ~  0
## 10 B103-023 900077  Never   50   0.704   F     Black Colleg~ New ~  0
## # i 4,286 more rows
## # i 17 more variables: location <fct>, round <fct>, nsigns_ssbb <dbl>,
## #   num_kids <chr>, surveydate <date>, days_since_ban <dbl>, caloriescal <dbl>,
## #   fatg <dbl>, sugarg <dbl>, black <chr>, caff_std <dbl>,
## #   nsigns_ssbb_std <dbl>, days_since_ban_std <dbl>, caloriescal_std <dbl>,
## #   fatg_std <dbl>, sugarg_std <dbl>, fit <ord>
##
## $conf
##           Truth
## Prediction Never Seldom Sometimes Often Always

```

```

##   Never      587     152      427     279     266
##   Seldom      22       8       26      11      16
##   Sometimes   272     66     220     150     193
##   Often       116     37     129      95     138
##   Always      276    114     252     160     284
##
## $acc
## # A tibble: 1 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>        <dbl>
## 1 accuracy  multiclass    0.278
##
## $sen
## # A tibble: 1 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>        <dbl>
## 1 sensitivity macro      0.229
##
## $spe
## # A tibble: 1 x 3
##   .metric   .estimator .estimate
##   <chr>     <chr>        <dbl>
## 1 specificity macro      0.809
##
## $gof
##
## Pearson's Chi-squared test
##
## data: control_results$limit and control_results$fit
## X-squared = 88.466, df = 16, p-value = 4.791e-12
plot(fitted(control_clm), residuals(control_clm))

```

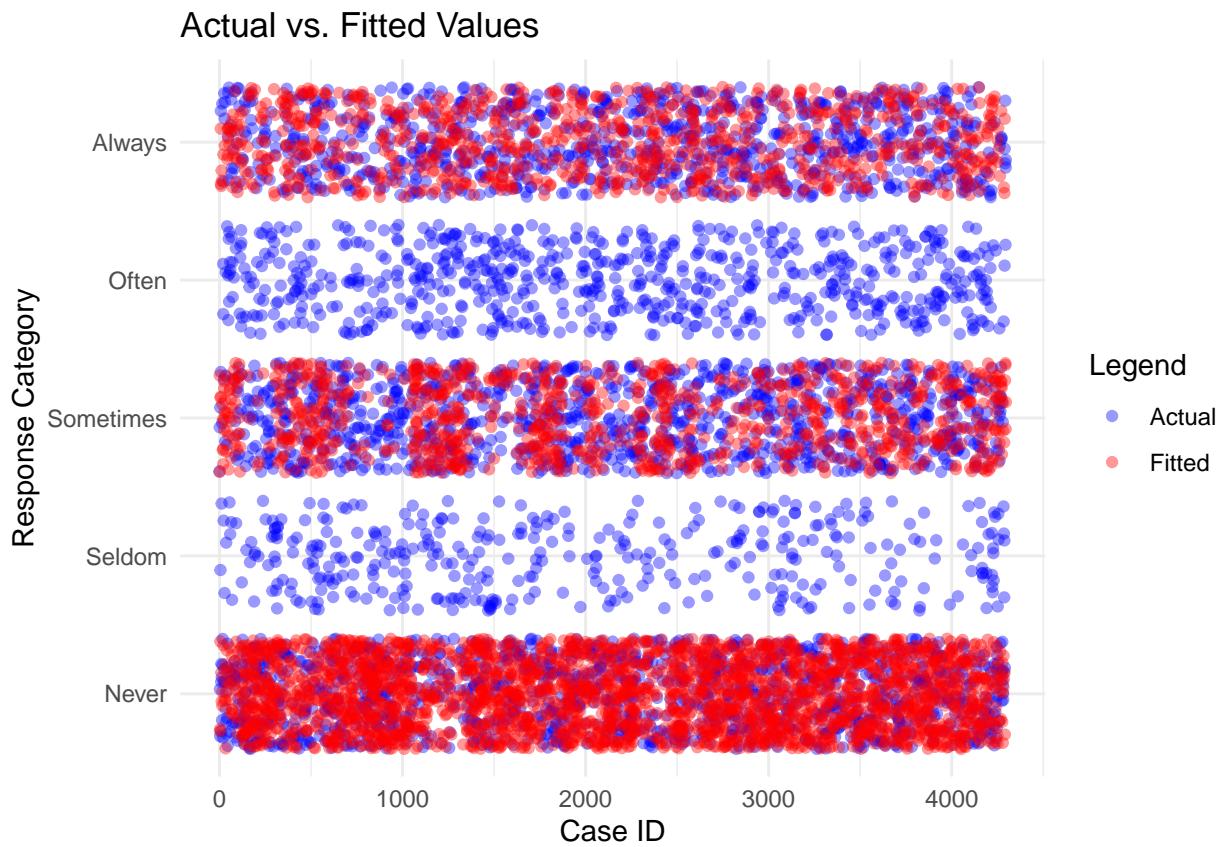


```

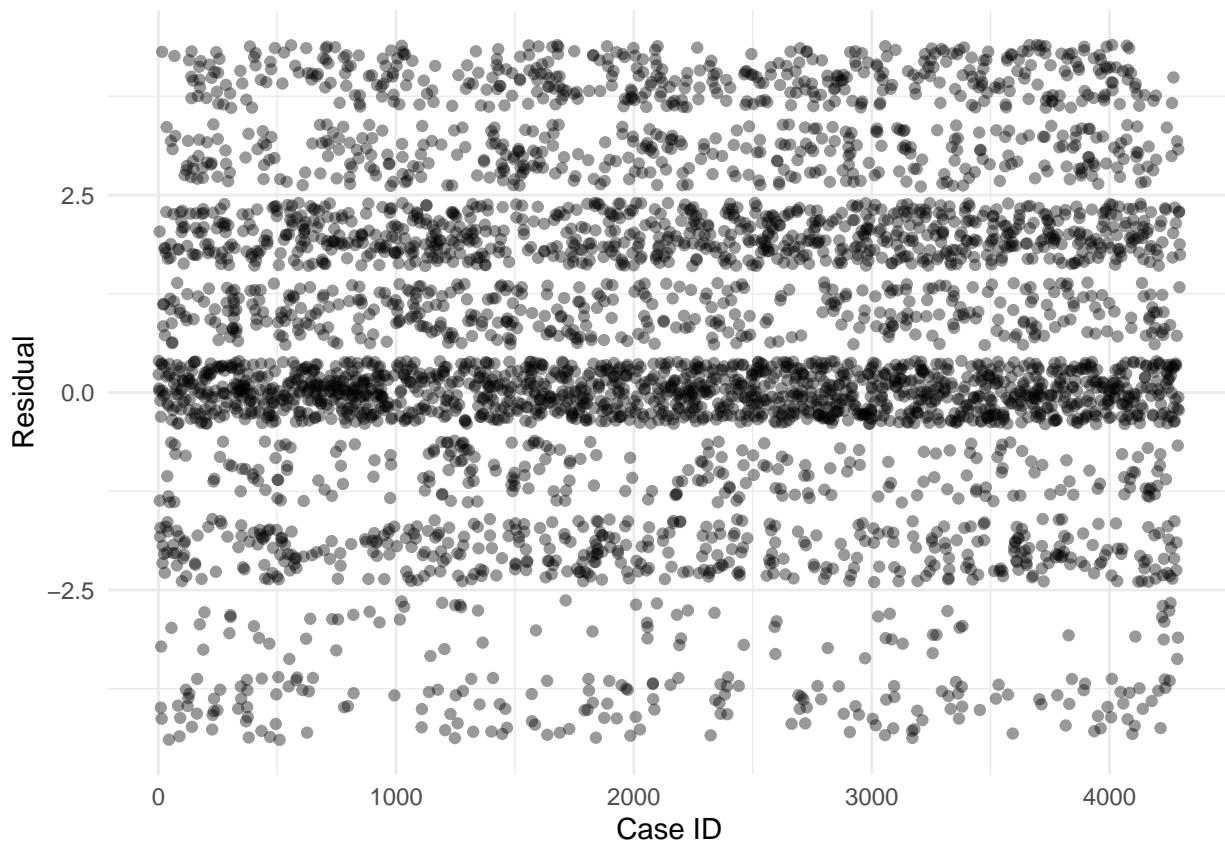
control_resid <- model_accuracy(control_clm_nom)$control_results %>%
  mutate(
    case_id = row_number(),
    fit = ordered(fit, levels = c("Never", "Seldom", "Sometimes", "Often", "Always"))
  ) %>%
  select(case_id, limit, fit) %>%
  mutate(
    limit_num = case_when(
      limit == "Never" ~ 0,
      limit == "Seldom" ~ 1,
      limit == "Sometimes" ~ 2,
      limit == "Often" ~ 3,
      limit == "Always" ~ 4),
    fit_num = case_when(
      fit == "Never" ~ 0,
      fit == "Seldom" ~ 1,
      fit == "Sometimes" ~ 2,
      fit == "Often" ~ 3,
      fit == "Always" ~ 4),
  ) %>%
  mutate(resid = limit_num - fit_num)

ggplot(control_resid, aes(x = case_id)) +
  geom_jitter(aes(y = limit, color = "Actual"), alpha = 0.4) +
  geom_jitter(aes(y = fit, color = "Fitted"), alpha = 0.4) +
  scale_color_manual(values = c("Actual" = "blue", "Fitted" = "red")) +
  labs(
    x = "Case ID",
    y = "Response Category",
    title = "Actual vs. Fitted Values",
    color = "Legend"
  ) +
  theme_minimal()

```



```
ggplot(control_resid, aes(x = case_id, y = resid)) +
  geom_jitter(alpha = 0.4) +
  labs(
    x = "Case ID",
    y = "Residual"
  ) +
  theme_minimal()
```



Effects Interpretation

Confidence Intervals

```
confint(control_clmm_loc) %>% kable(digits = 3)
```

	2.5 %	97.5 %
Never Seldom	-1.504	-0.861
Seldom Sometimes	-1.093	-0.452
Sometimes Often	-0.044	0.595
Often Always	0.784	1.428
age_std	0.168	0.284
genderM	-0.400	-0.178
raceBlack	-0.419	0.046
raceNative	-0.671	0.184
raceOther	-0.206	0.290
raceWhite	-0.286	0.195
eduCollege Degree	0.028	0.523
eduGraduate Degree	-0.033	0.539
eduHigh School	-0.544	-0.067
eduLess than High School	-0.584	0.211
eduSome College	-0.314	0.177
eduSome High School	-0.700	-0.128
cityNew York	-0.003	0.282
num_kids1	-0.149	0.303
num_kids2	-0.680	0.114

	2.5 %	97.5 %
num_kids3+	-2.492	-0.382

```
exp(confint(control_clmm_loc)) %>% kable(digits = 3)
```

	2.5 %	97.5 %
Never Seldom	0.222	0.423
Seldom Sometimes	0.335	0.636
Sometimes Often	0.957	1.812
Often Always	2.191	4.168
age_std	1.183	1.329
genderM	0.670	0.837
raceBlack	0.658	1.047
raceNative	0.511	1.202
raceOther	0.814	1.337
raceWhite	0.751	1.215
eduCollege Degree	1.029	1.686
eduGraduate Degree	0.967	1.715
eduHigh School	0.580	0.935
eduLess than High School	0.558	1.234
eduSome College	0.731	1.193
eduSome High School	0.497	0.880
cityNew York	0.997	1.326
num_kids1	0.862	1.353
num_kids2	0.507	1.121
num_kids3+	0.083	0.683

```
(100*(exp(confint(control_clmm_loc))-1)) %>% kable(digits = 3)
```

	2.5 %	97.5 %
Never Seldom	-77.780	-57.728
Seldom Sometimes	-66.476	-36.369
Sometimes Often	-4.335	81.219
Often Always	119.107	316.830
age_std	18.302	32.880
genderM	-32.964	-16.303
raceBlack	-34.229	4.694
raceNative	-48.866	20.231
raceOther	-18.589	33.680
raceWhite	-24.888	21.489
eduCollege Degree	2.873	68.644
eduGraduate Degree	-3.266	71.463
eduHigh School	-41.986	-6.519
eduLess than High School	-44.208	23.448
eduSome College	-26.928	19.344
eduSome High School	-50.345	-11.975
cityNew York	-0.338	32.570
num_kids1	-13.842	35.329
num_kids2	-49.331	12.116
num_kids3+	-91.725	-31.720