Data Understanding and Data Exploration

Daniel-Mateus-Pires

PDF config

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

EuroCom

Dependencies

```
# install.packages('ggplot2')
library(ggplot2)
```

Reading the dataset

```
phones <- read.csv("./eurocomPHONEchurners.csv")
head(phones)
```

```
##   CUST_ID AREA_CODE MINUTES_CURR_MONTH MINUTES_PREV_MONTH
## 1     129     45987                 60                456
## 2     130     15563                  2                  0
## 3     131     10040                  2                  0
## 4     132     21750                678               1222
## 5     133     55166                110                 98
## 6     134     36785                 97                 56
##   MINUTES_3MONTHS_AGO CUST_MOS LONGDIST_FLAG CALLWAITING_FLAG NUM_LINES
## 1                 398       13             0                1         1
## 2                   4        4             0                0         1
## 3                   0        1             0                0         1
## 4                 598       30             1                1         2
## 5                  56       15             1                0         1
## 6                  97        8             0                0         1
##   VOICEMAIL_FLAG MOBILE_PLAN CONVERGENT_BILLING GENDER INCOME
## 1              1           0                Yes      M  88000
## 2              0           0                Yes      M  53000
## 3              1           0                 No      F  29000
## 4              0           1                Yes      M  46000
## 5              1           0                 No      M  98000
## 6              0           1                 No      M 125000
##      PHONE_PLAN   EDUCATION TOT_MINUTES_USAGE CHURNER
## 1 International     Masters               914     yes
## 2 International   Bachelors                 6      no
## 3      National High School                 0      no
## 4 International High School              2498     yes
## 5    Promo_plan High School               264      no
```

```
## 6      National                          250      no
```

1

Data pre-processing

1.a

Getting how many null values, or empty string values there is per column.

```
count_na <- sapply(phones, function(y) sum(length(which(is.na(y) |
    y == ""))))
na_df <- data.frame(count_na)

subset(na_df, na_df$count_na > 0)
```

```
##                      count_na
## MINUTES_3MONTHS_AGO         3
## CUST_MOS                    3
## PHONE_PLAN                  4
## EDUCATION                   8
## TOT_MINUTES_USAGE           4
```

1.b

Replacing na numerics with medians

```
replace_na_with_median <- function(col) {
    median_without_na <- median(col, na.rm = TRUE)
    col[is.na(col)] <- median_without_na
    return(col)
}
```

MINUTES_3MONTHS_AGO

```
phones$MINUTES_3MONTHS_AGO <- replace_na_with_median(phones$MINUTES_3MONTHS_AGO)
```

CUST_MOS

```
phones$CUST_MOS <- replace_na_with_median(phones$CUST_MOS)
```

TOT_MINUTES_USAGE

```
phones$TOT_MINUTES_USAGE <- replace_na_with_median(phones$TOT_MINUTES_USAGE)
```

1.c

Getting the mode for categorical columns PER GENDER

```
get_mode <- function(x) {
    xtable <- table(x)
    idx <- xtable == max(xtable)
    names(xtable)[idx]
}
```

Function to get all modes from a data frame

```
get_modes <- function(x) {
    if (class(x) == "numeric" | class(x) == "integer")
        return("X")
    xtable <- table(x)
    idx <- xtable == max(xtable)
    names(xtable)[idx]
}
```

Displaying modes for males

```
phones_male <- phones[phones$GENDER == "M", ]

modes_male <- data.frame(sapply(phones_male, get_modes))
names(modes_male)[1] <- "MODE_MALE"
modes_male <- subset(modes_male, MODE_MALE != "X")

modes_male
```

```
##                        MODE_MALE
## CONVERGENT_BILLING           Yes
## GENDER                         M
## PHONE_PLAN         International
## EDUCATION           Post Primary
## CHURNER                      yes
```

Displaying modes for females

```
phones_female <- phones[phones$GENDER == "F", ]

modes_female <- data.frame(sapply(phones_female, get_modes))
names(modes_female)[1] <- "MODE_FEMALE"
modes_female <- subset(modes_female, MODE_FEMALE != "X")

modes_female
```

```
##                      MODE_FEMALE
## CONVERGENT_BILLING            No
## GENDER                         F
## PHONE_PLAN         International
## EDUCATION              Bachelors
## CHURNER                       no
```

PHONE_PLAN

```
phones$PHONE_PLAN[phones$PHONE_PLAN == "" & phones$GENDER ==
    "M"] <- get_mode(phones$PHONE_PLAN[phones$GENDER == "M"])
phones$PHONE_PLAN[phones$PHONE_PLAN == "" & phones$GENDER ==
    "F"] <- get_mode(phones$PHONE_PLAN[phones$GENDER == "F"])
```

EDUCATION

```
phones$EDUCATION[phones$EDUCATION == "" & phones$GENDER == "M"] <- get_mode(phones$EDUCATION[phones$GEN
    "M"])
phones$EDUCATION[phones$EDUCATION == "" & phones$GENDER == "F"] <- get_mode(phones$EDUCATION[phones$GEN
    "F"])
```

AREA_CODE

```
phones$AREA_CODE[phones$AREA_CODE == "" & phones$GENDER == "M"] <- get_mode(phones$AREA_CODE[phones$GEN
    "M"])
phones$AREA_CODE[phones$AREA_CODE == "" & phones$GENDER == "F"] <- get_mode(phones$AREA_CODE[phones$GEN
    "F"])
```

LONGDIST_FLAG

```
phones$LONGDIST_FLAG[phones$LONGDIST_FLAG == "" & phones$GENDER ==
    "M"] <- get_mode(phones$LONGDIST_FLAG[phones$GENDER == "M"])
phones$LONGDIST_FLAG[phones$LONGDIST_FLAG == "" & phones$GENDER ==
    "F"] <- get_mode(phones$LONGDIST_FLAG[phones$GENDER == "F"])
```

CALLWAITING_FLAG

```
phones$CALLWAITING_FLAG[phones$CALLWAITING_FLAG == "" & phones$GENDER ==
    "M"] <- get_mode(phones$CALLWAITING_FLAG[phones$GENDER ==
    "M"])
phones$CALLWAITING_FLAG[phones$CALLWAITING_FLAG == "" & phones$GENDER ==
    "F"] <- get_mode(phones$CALLWAITING_FLAG[phones$GENDER ==
    "F"])
```

VOICEMAIL_FLAG

```
phones$VOICEMAIL_FLAG[phones$VOICEMAIL_FLAG == "" & phones$GENDER ==
    "M"] <- get_mode(phones$VOICEMAIL_FLAG[phones$GENDER == "M"])
phones$VOICEMAIL_FLAG[phones$VOICEMAIL_FLAG == "" & phones$GENDER ==
    "F"] <- get_mode(phones$VOICEMAIL_FLAG[phones$GENDER == "F"])
```

MOBILE_PLAN

```
phones$MOBILE_PLAN[phones$MOBILE_PLAN == "" & phones$GENDER ==
    "M"] <- get_mode(phones$MOBILE_PLAN[phones$GENDER == "M"])
phones$MOBILE_PLAN[phones$MOBILE_PLAN == "" & phones$GENDER ==
    "F"] <- get_mode(phones$MOBILE_PLAN[phones$GENDER == "F"])
```

2

Discretising Income predictor values

```
head(phones$INCOME)
```

```
## [1]  88000  53000  29000  46000  98000 125000
```

```
phones$INCOME <- cut(phones$INCOME, breaks = c(0, 37999, 88000,
    max(phones$INCOME)), include.lowest = TRUE, labels = c("Low Income",
    "Medium Income", "High Income"))

head(phones$INCOME)
```

```
## [1] Medium Income Medium Income Low Income    Medium Income High Income
## [6] High Income
## Levels: Low Income Medium Income High Income
```

3.c

```
get_mode(phones$AREA_CODE)
```

```
## [1] "10040"
```

```
summary(phones$CUST_MOS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.00   11.00   16.05   26.00   50.00
```

```
get_mode(phones$LONGDIST_FLAG)
```

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

```
get_mode(phones$CALLWAITING_FLAG)
```

```
## [1] "0"
```

```
summary(phones$NUM_LINES)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.391   2.000   3.000
```

```
get_mode(phones$VOICEMAIL_FLAG)
```

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

```
get_mode(phones$MOBILE_PLAN)
```

```
## [1] "0"
```

```
get_mode(phones$CONVERGENT_BILLING)
```

```
## [1] "No"
```

```
get_mode(phones$GENDER)
```

```
## [1] "M"
```

```
get_mode(phones$INCOME)
```

```
## [1] "Medium Income"
```

```
get_mode(phones$PHONE_PLAN)
```

```
## [1] "International"
```

```
get_mode(phones$EDUCATION)
```

```
## [1] "Post Primary"
```

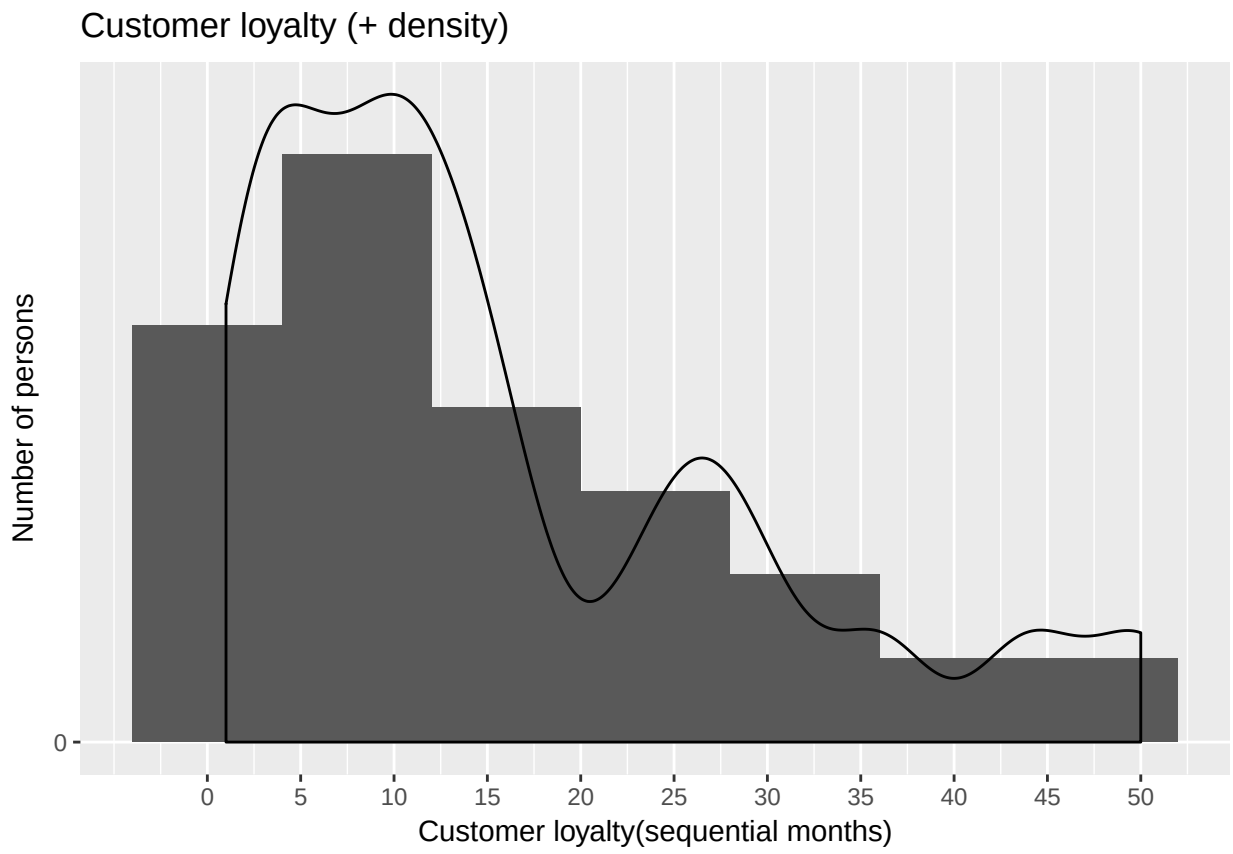
```
summary(phones$TOT_MINUTES_USAGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     116     264    2036    1677   36240
```

3.d

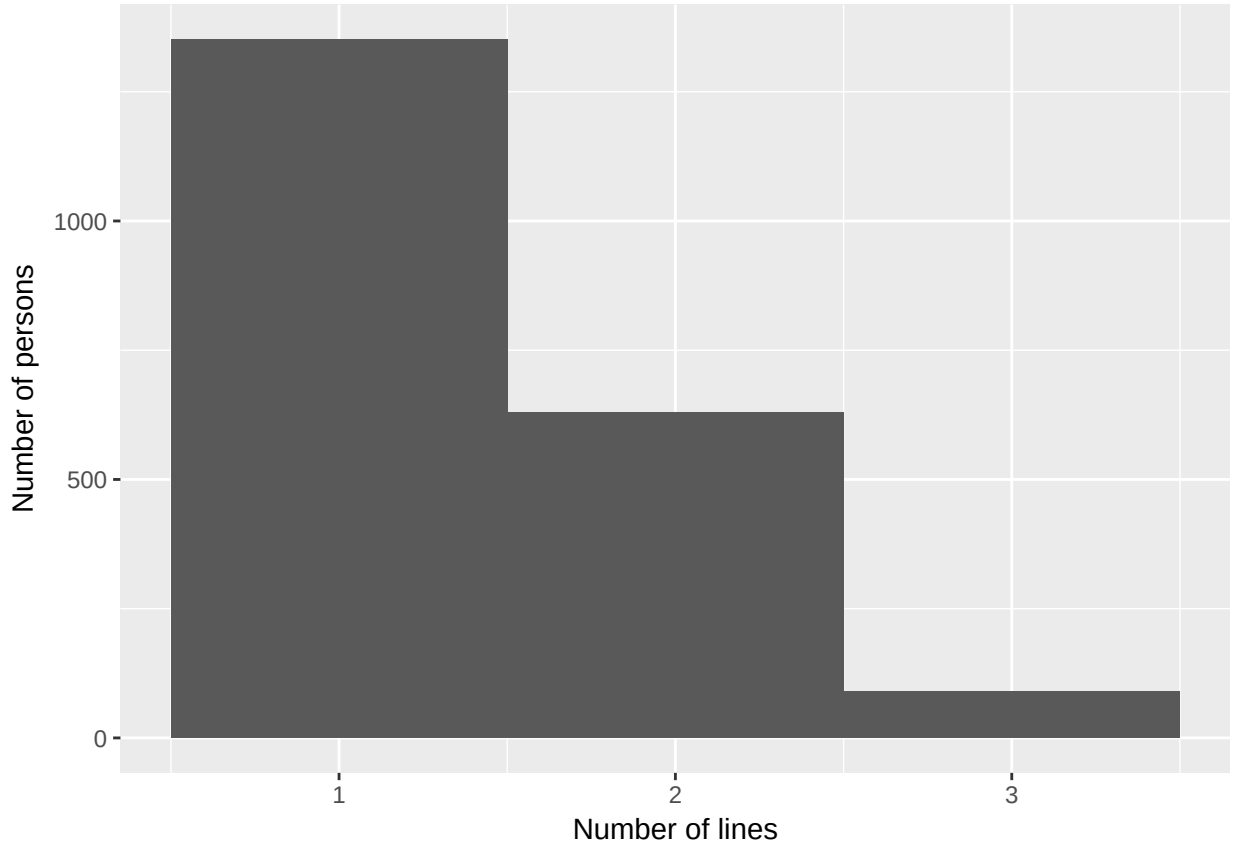
CUST_MOS

```
ggplot(data = phones, aes(phones$CUST_MOS)) + geom_histogram(binwidth = 8,
    aes(y = ..density..)) + scale_x_continuous(breaks = seq(0,
    60, 5)) + scale_y_continuous(breaks = seq(0, 1000, 50)) +
    labs(x = "Customer loyalty(sequential months)", y = "Number of persons",
        title = "Customer loyalty (+ density)") + geom_density()
```



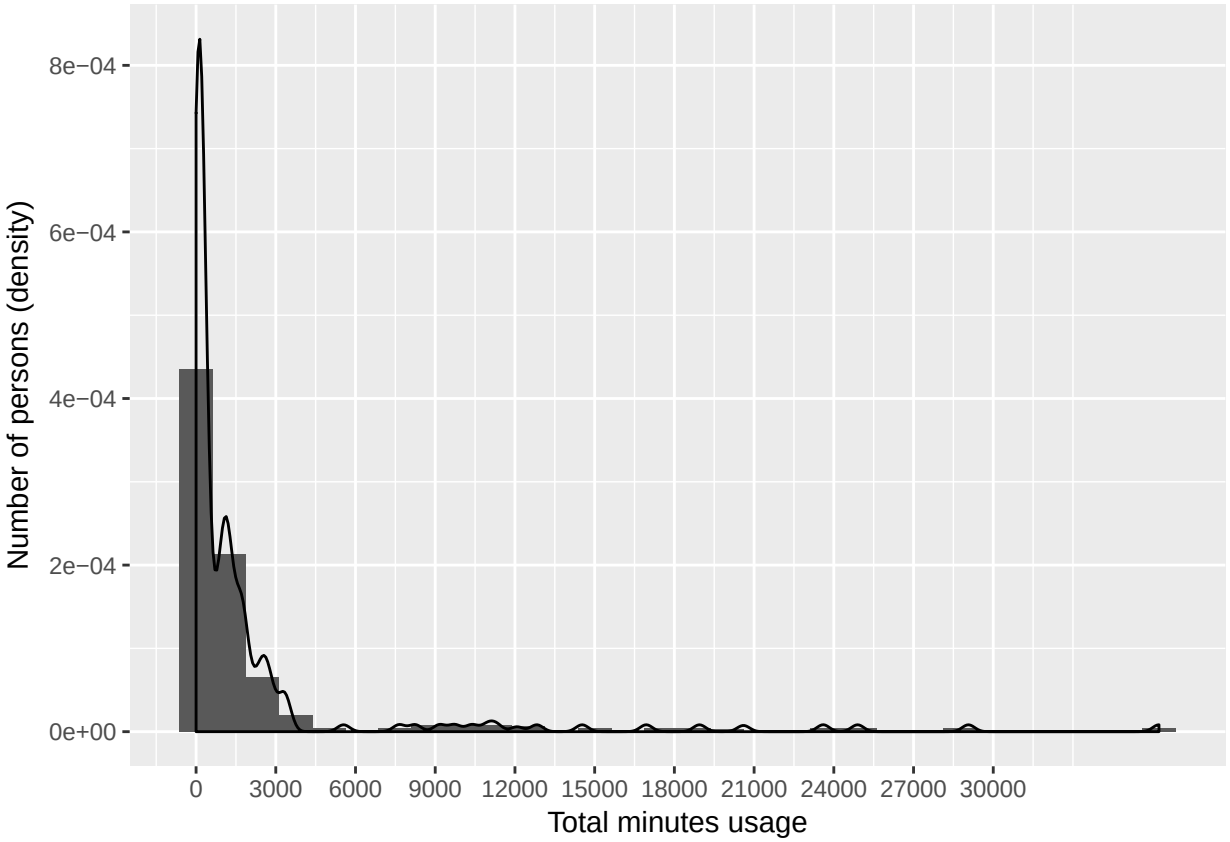
NUM_LINES

```
ggplot(data = phones, aes(phones$NUM_LINES)) + geom_histogram(binwidth = 1) +
    scale_x_continuous(breaks = 0:3) + labs(x = "Number of lines",
    y = "Number of persons")
```



TOT_MINUTES_USAGE

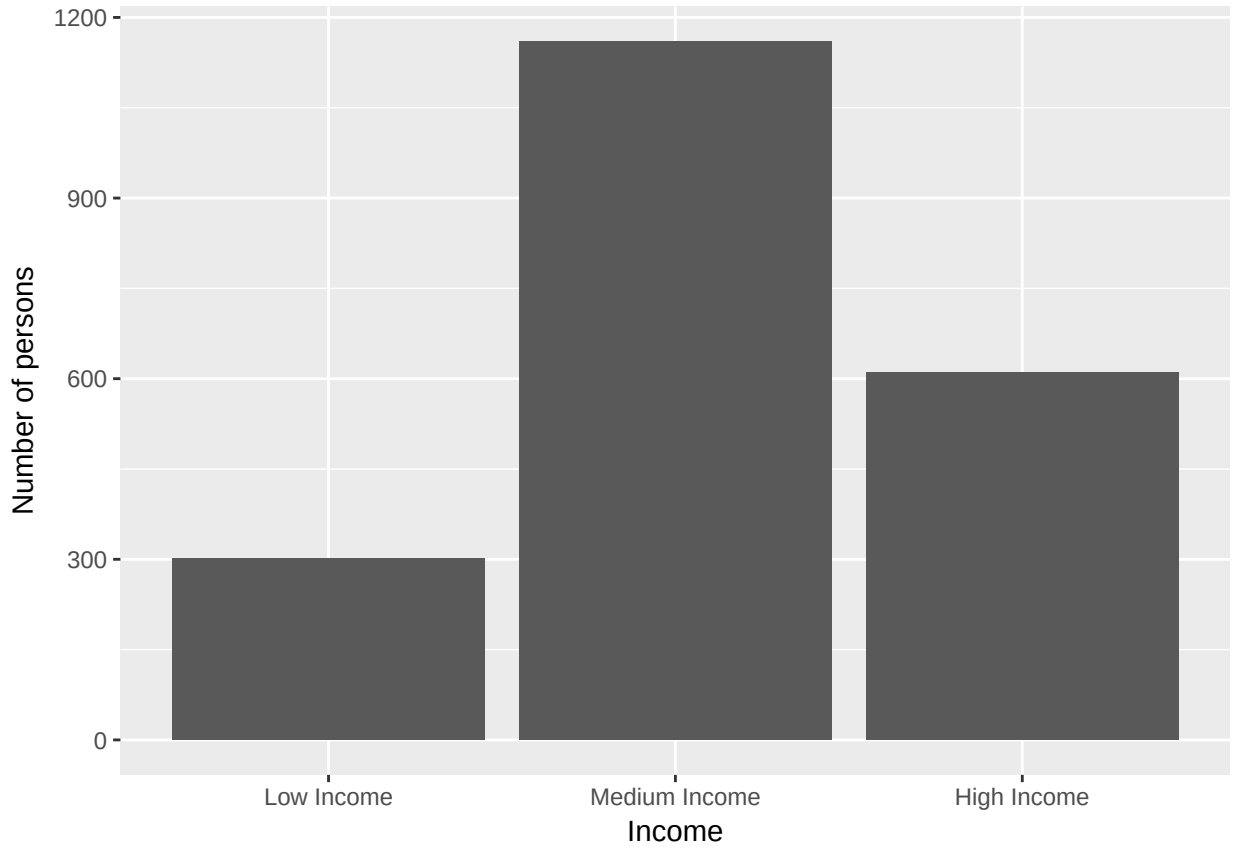
```
ggplot(data = phones, aes(phones$TOT_MINUTES_USAGE)) + geom_histogram(bins = 30,
    aes(y = ..density..)) + scale_x_continuous(breaks = seq(0,
    30000, 3000)) + labs(x = "Total minutes usage", y = "Number of persons (density)") +
    geom_density()
```



INCOME

```
ggplot(data = phones, aes(phones$INCOME)) + geom_histogram(stat = "count") +
    labs(x = "Income", y = "Number of persons")
```

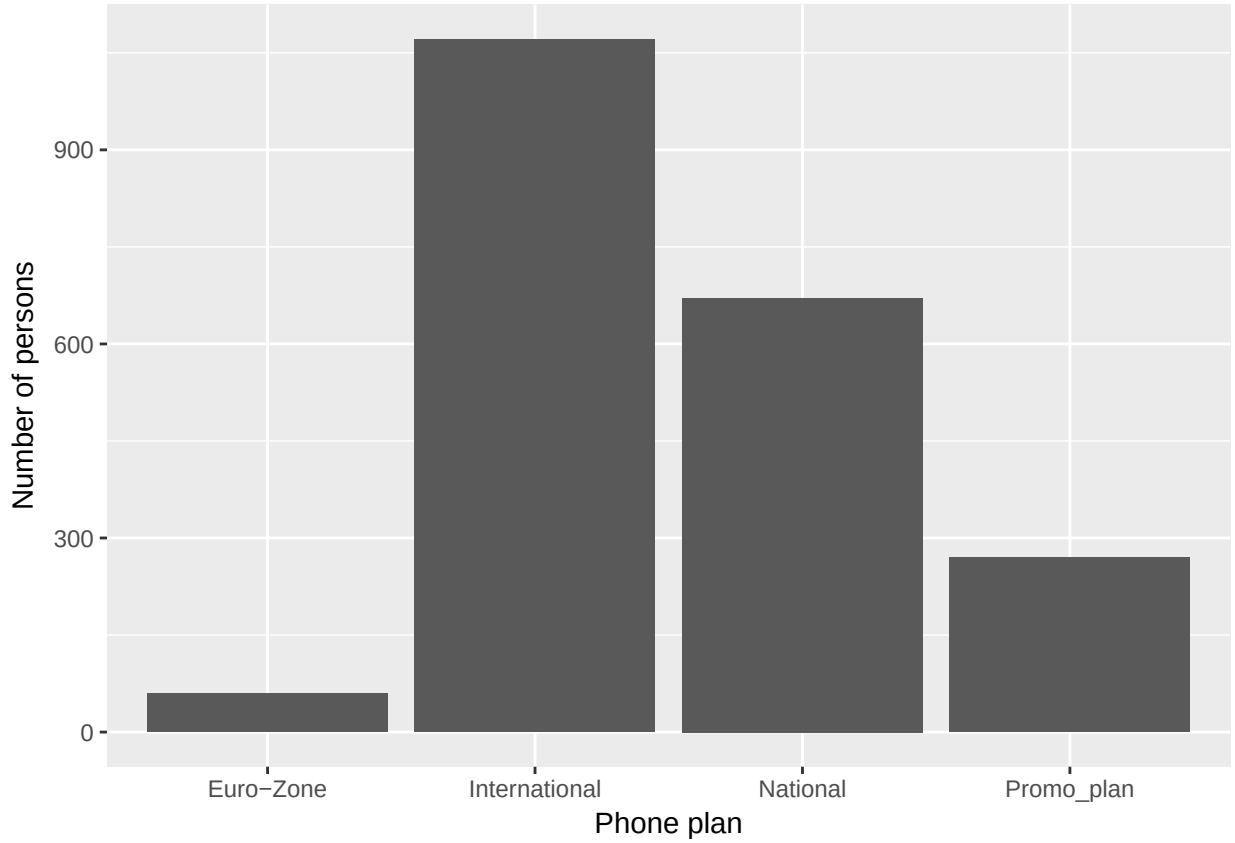
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



PHONE_PLAN

```
ggplot(data = phones, aes(phones$PHONE_PLAN)) + geom_histogram(stat = "count") +
    labs(x = "Phone plan", y = "Number of persons")
```

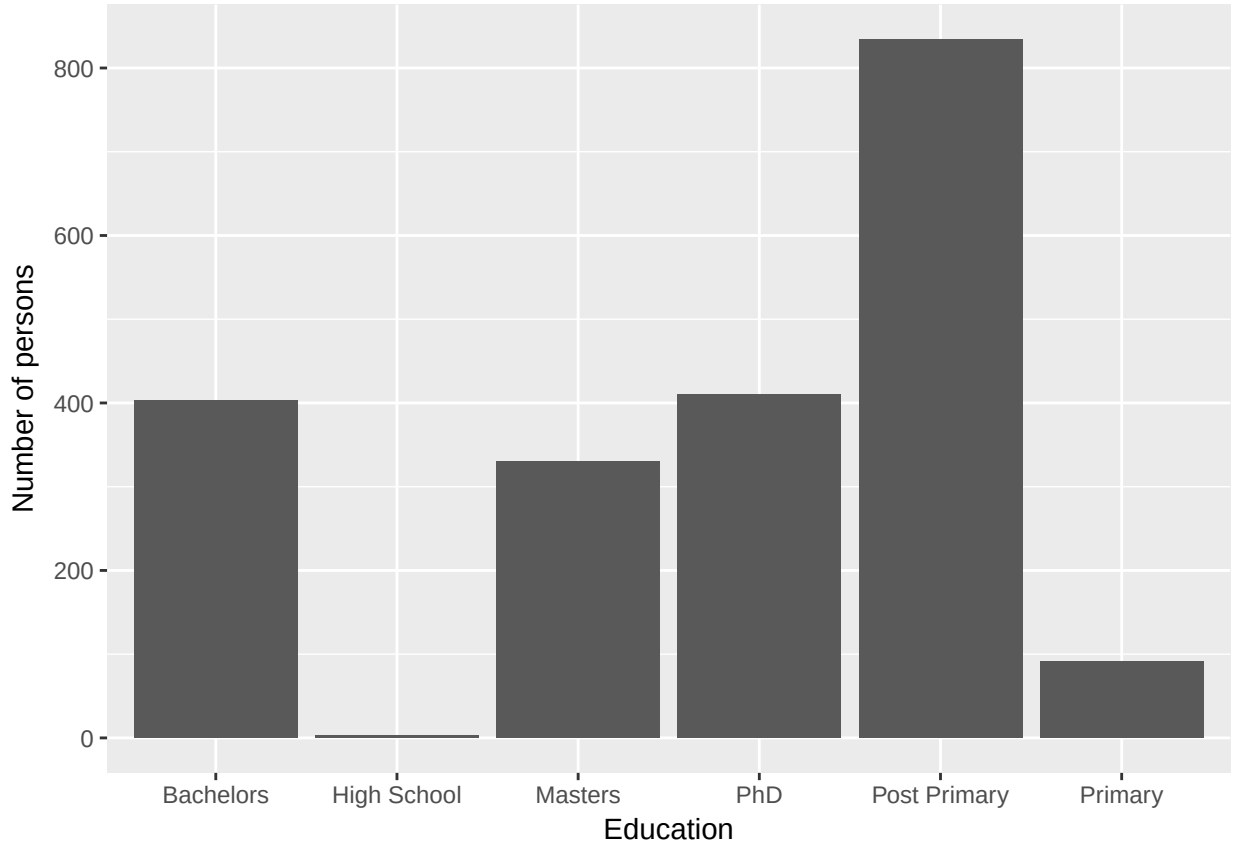
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



EDUCATION

```
ggplot(data = phones, aes(phones$EDUCATION)) + geom_histogram(stat = "count") +
    labs(x = "Education", y = "Number of persons")
```

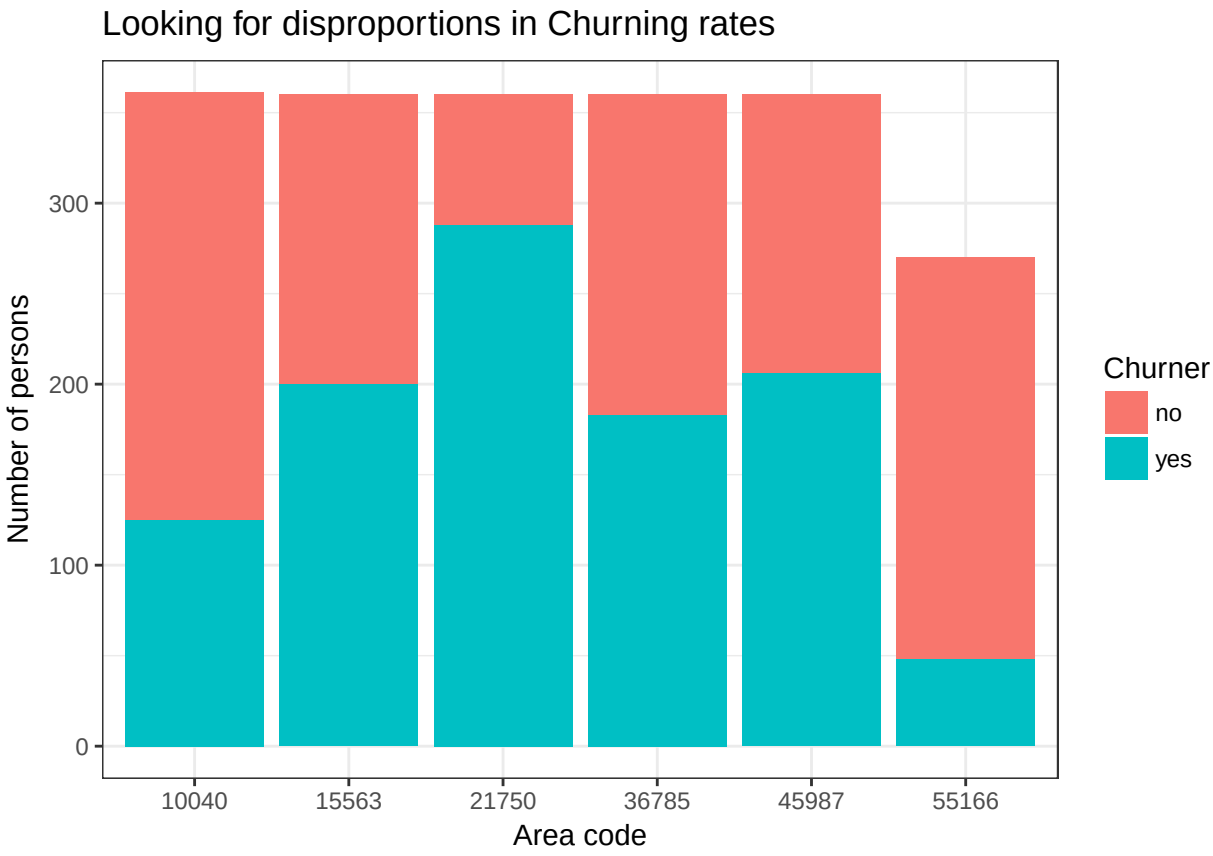
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



3.e ### AREA_CODE

```
ggplot(data = phones, aes(x = phones$AREA_CODE, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(stat = "count") +
    theme_bw() + labs(x = "Area code", y = "Number of persons",
    title = "Looking for disproportions in Churning rates", fill = "Churner")
```

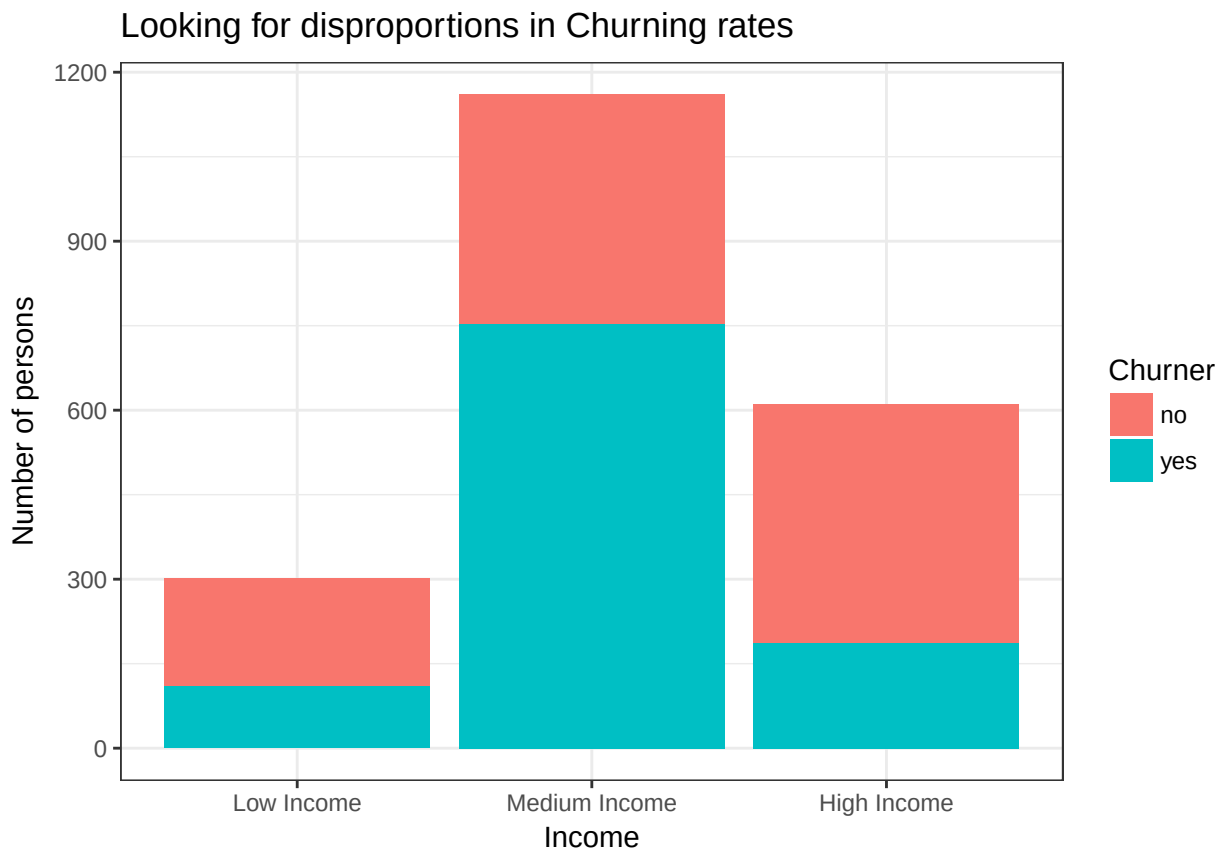
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



INCOME

```
ggplot(data = phones, aes(x = phones$INCOME, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(stat = "count") +
    theme_bw() + labs(x = "Income", y = "Number of persons",
    title = "Looking for disproportions in Churning rates", fill = "Churner")
```

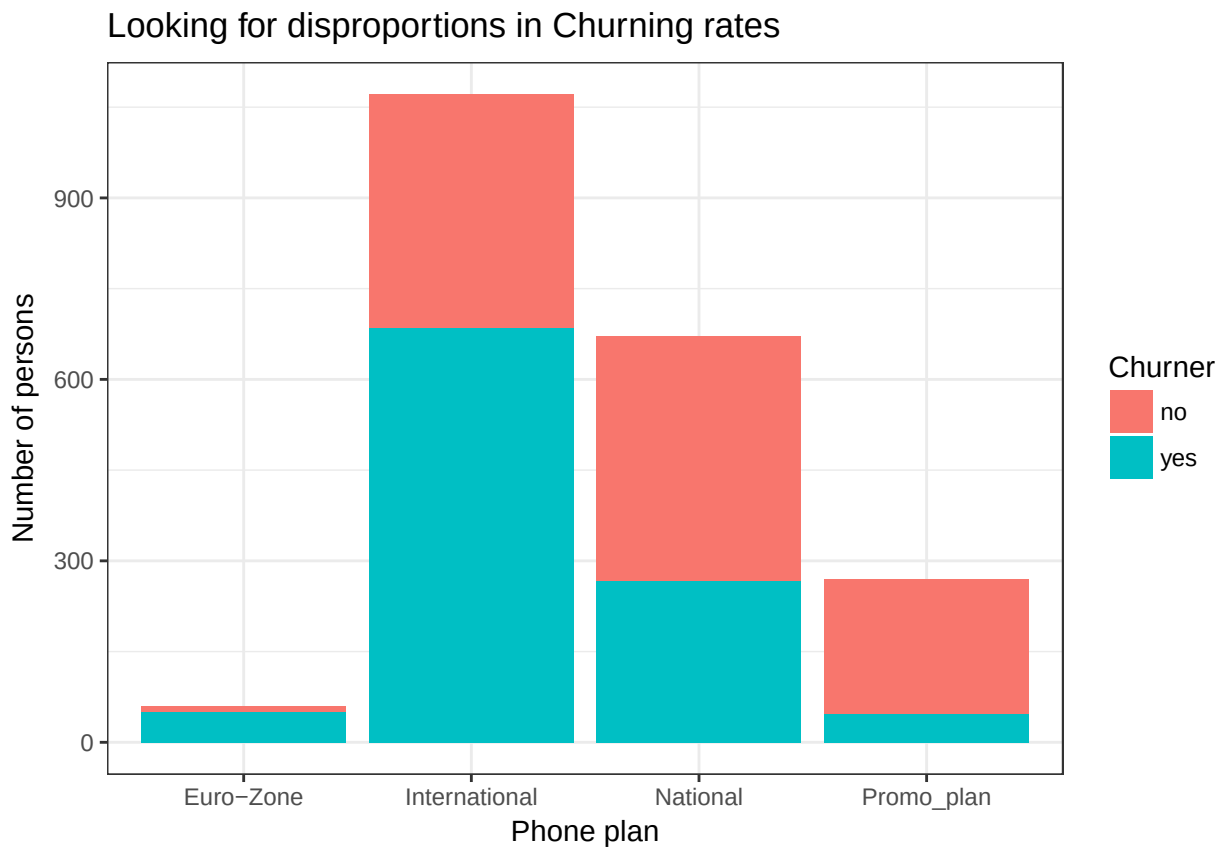
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



PHONE_PLAN

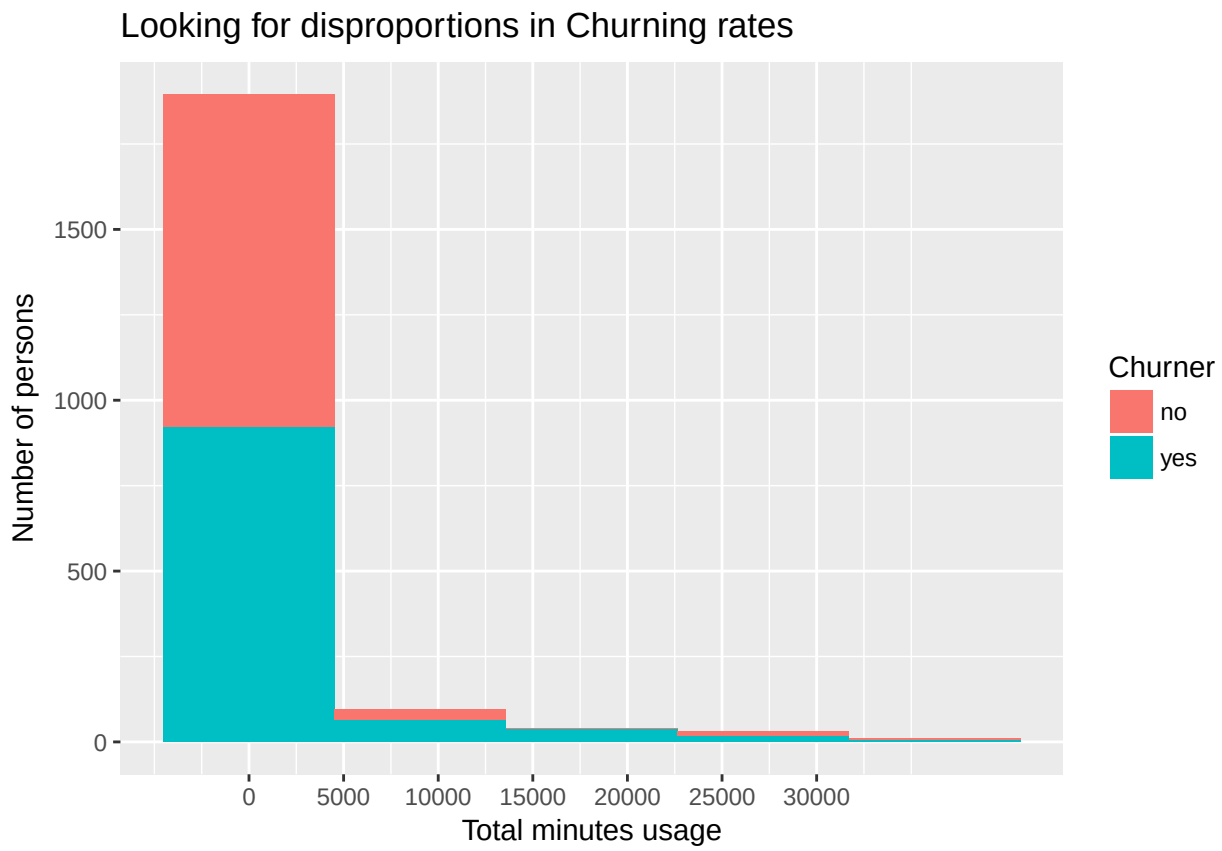
```
ggplot(data = phones, aes(x = phones$PHONE_PLAN, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(stat = "count") +
    theme_bw() + labs(x = "Phone plan", y = "Number of persons",
    title = "Looking for disproportions in Churning rates", fill = "Churner")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



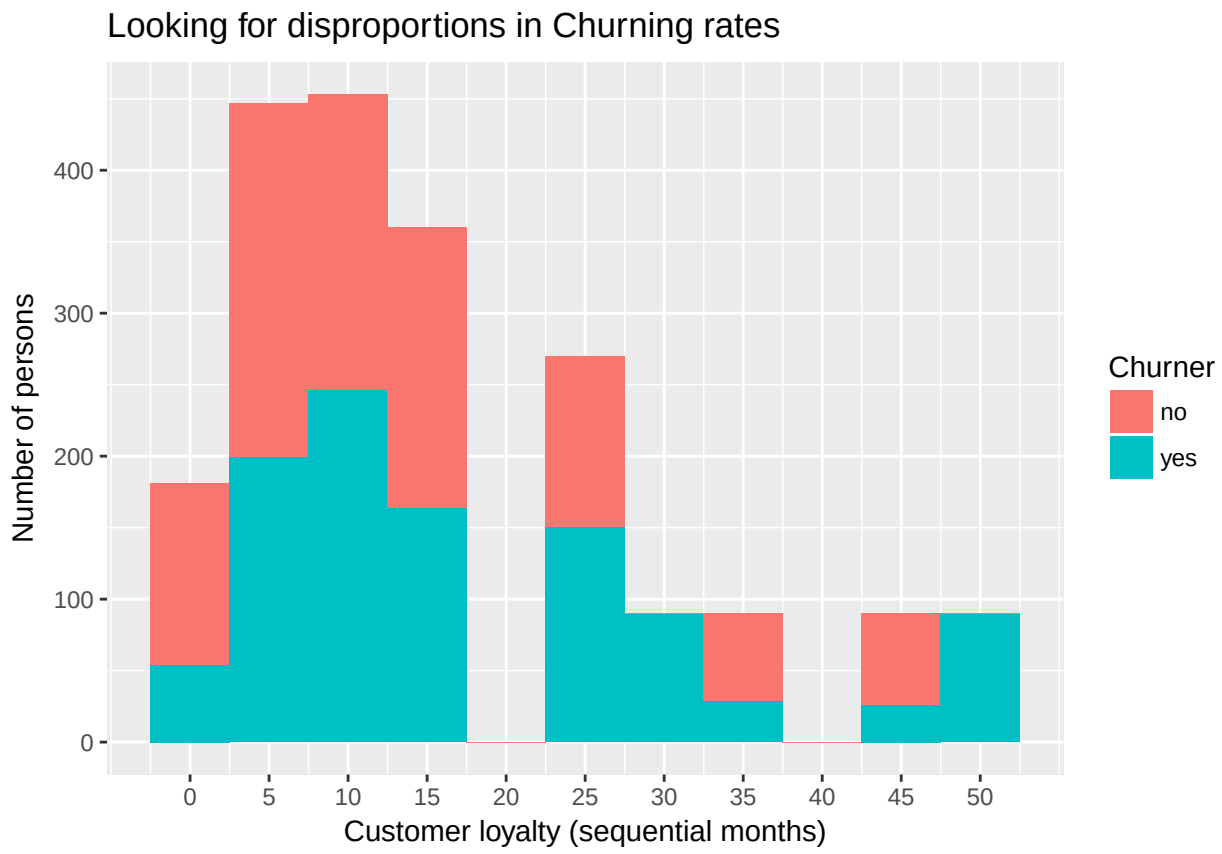
TOT_MINUTES_USAGE

```
ggplot(data = phones, aes(x = phones$TOT_MINUTES_USAGE, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(bins = 5) + scale_x_continuous(breaks = seq(0,
    30000, 5000)) + labs(x = "Total minutes usage", y = "Number of persons",
    title = "Looking for disproportions in Churning rates", fill = "Churner")
```



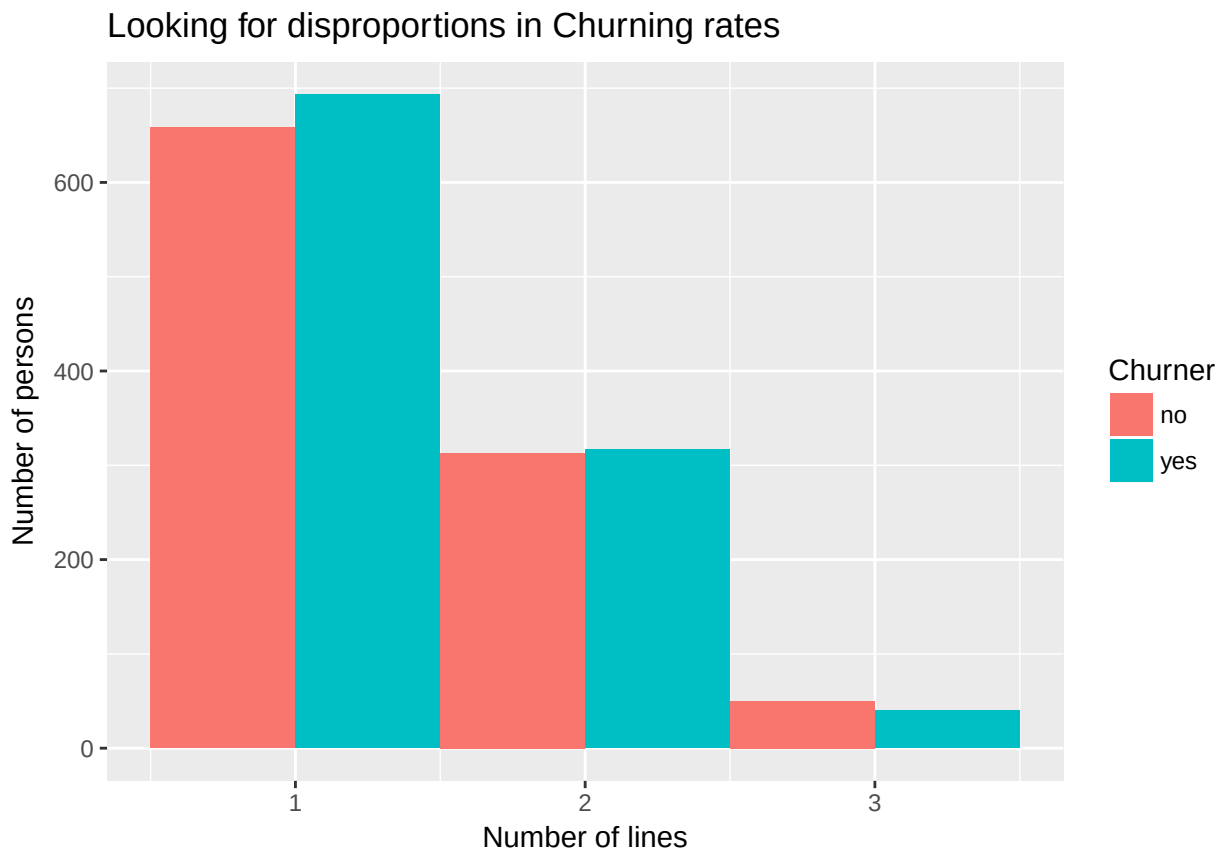
CUST_MOS

```
ggplot(data = phones, aes(x = phones$CUST_MOS, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(binwidth = 5) +
    scale_x_continuous(breaks = seq(0, 50, 5)) + labs(x = "Customer loyalty (sequential months)",
    y = "Number of persons", title = "Looking for disproportions in Churning rates",
    fill = "Churner")
```



NUM_LINES

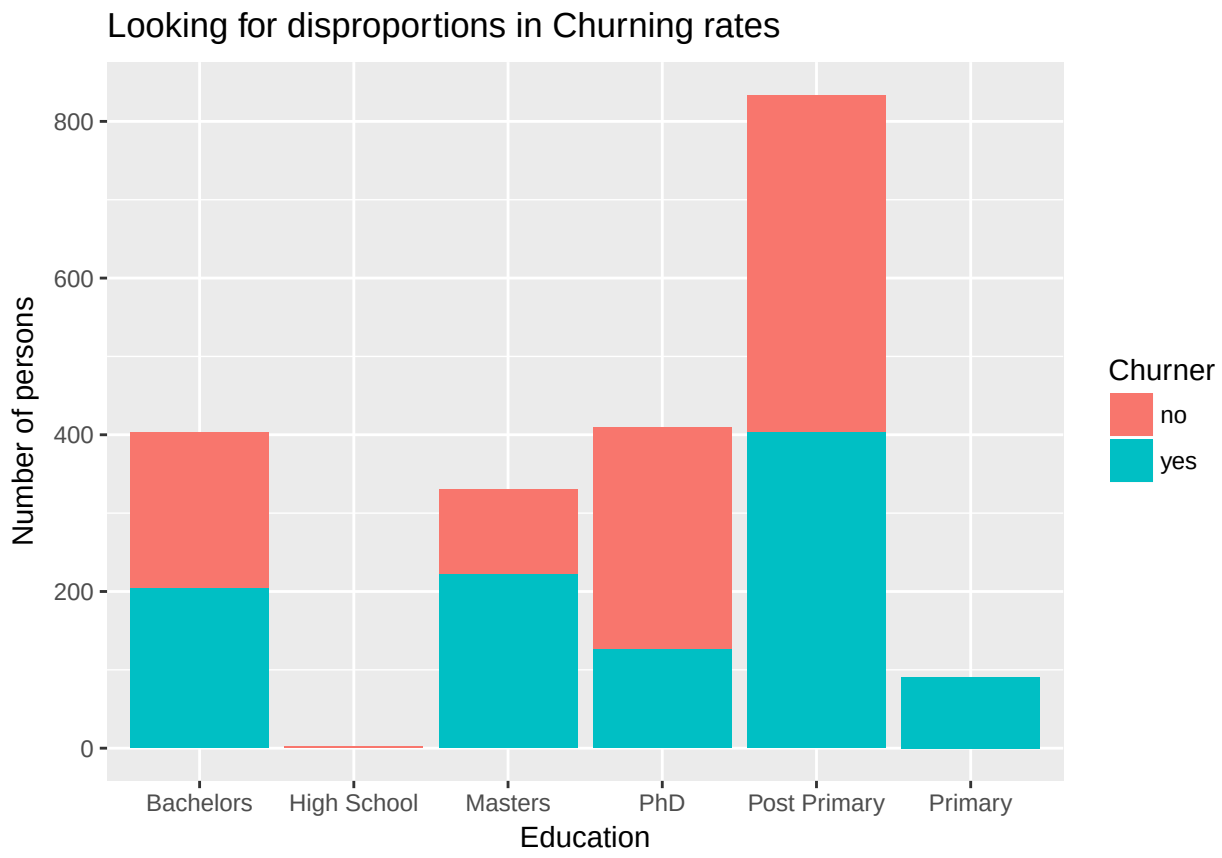
```
ggplot(data = phones, aes(x = phones$NUM_LINES, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(binwidth = 1, position = "dodge") +
    scale_x_continuous(breaks = 0:3) + labs(x = "Number of lines",
    y = "Number of persons", title = "Looking for disproportions in Churning rates",
    fill = "Churner")
```



EDUCATION

```
ggplot(data = phones, aes(x = phones$EDUCATION, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(stat = "count") +
    labs(x = "Education", y = "Number of persons", title = "Looking for disproportions in Churning rates
        fill = "Churner")
```

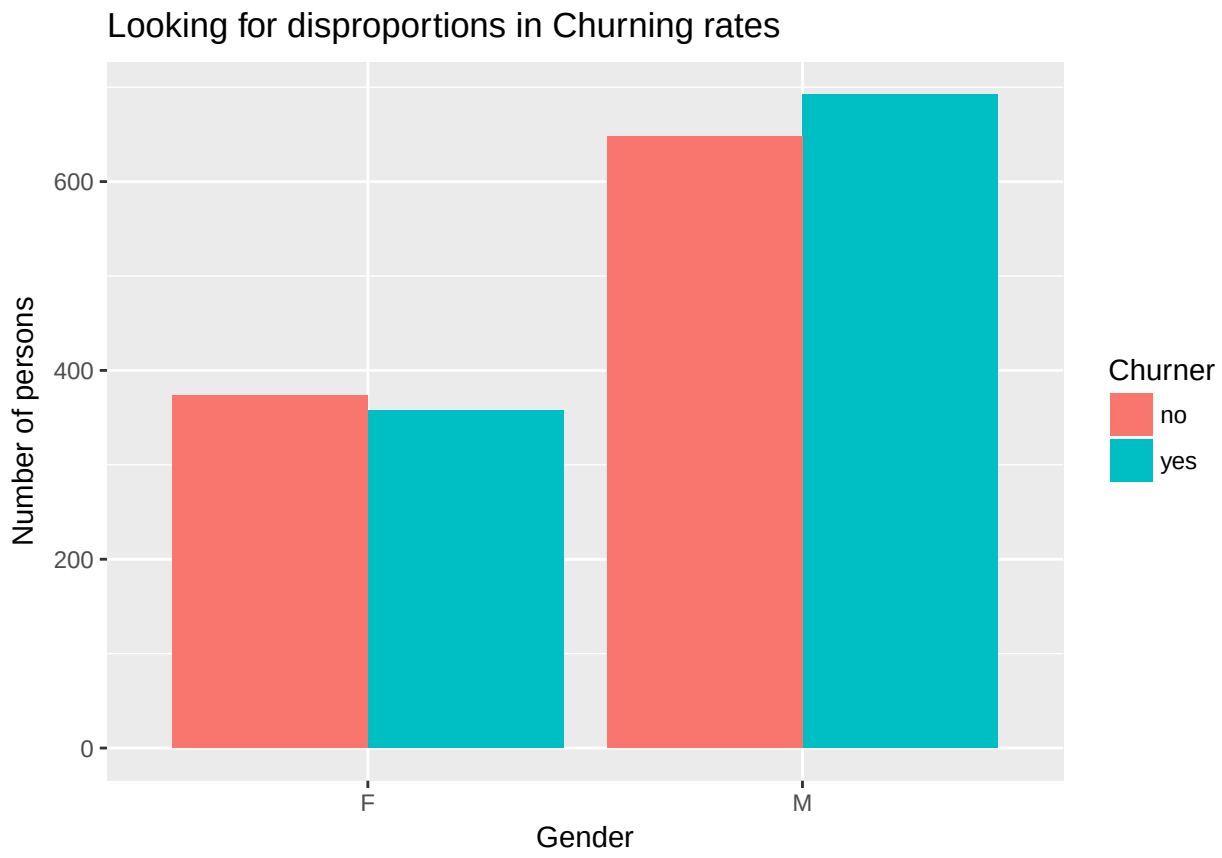
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



GENDER

```
ggplot(data = phones, aes(x = phones$GENDER, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(stat = "count",
    position = "dodge") + labs(x = "Gender", y = "Number of persons",
    title = "Looking for disproportions in Churning rates", fill = "Churner")
```

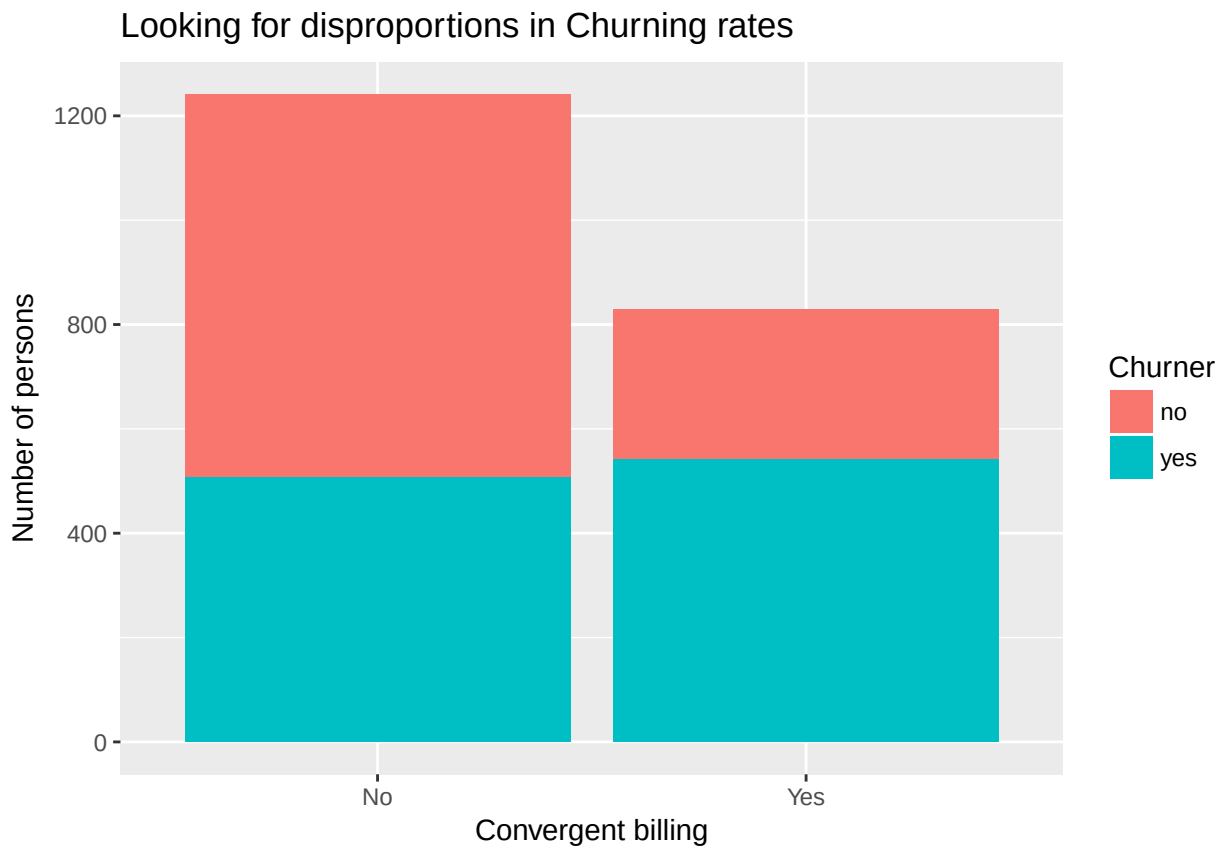
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



CONVERGENT_BILLING

```
ggplot(data = phones, aes(x = phones$CONVERGENT_BILLING, group = phones$CHURNER,
    fill = phones$CHURNER)) + geom_histogram(stat = "count") +
    labs(x = "Convergent billing", y = "Number of persons", title = "Looking for disproportions in Churn
        fill = "Churner")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



3.f ### CUST_MOS

```
cust_mos_skew <- (3 * (mean(phones$CUST_MOS) - median(phones$CUST_MOS)))/sd(phones$CUST_MOS)
cust_mos_skew
```

```
## [1] 1.131224
```

NUM_LINES

```
num_lines_skew <- (3 * (mean(phones$NUM_LINES) - median(phones$NUM_LINES)))/sd(phones$NUM_LINES)
num_lines_skew
```

```
## [1] 2.057503
```

TOT_MINUTES_USAGE

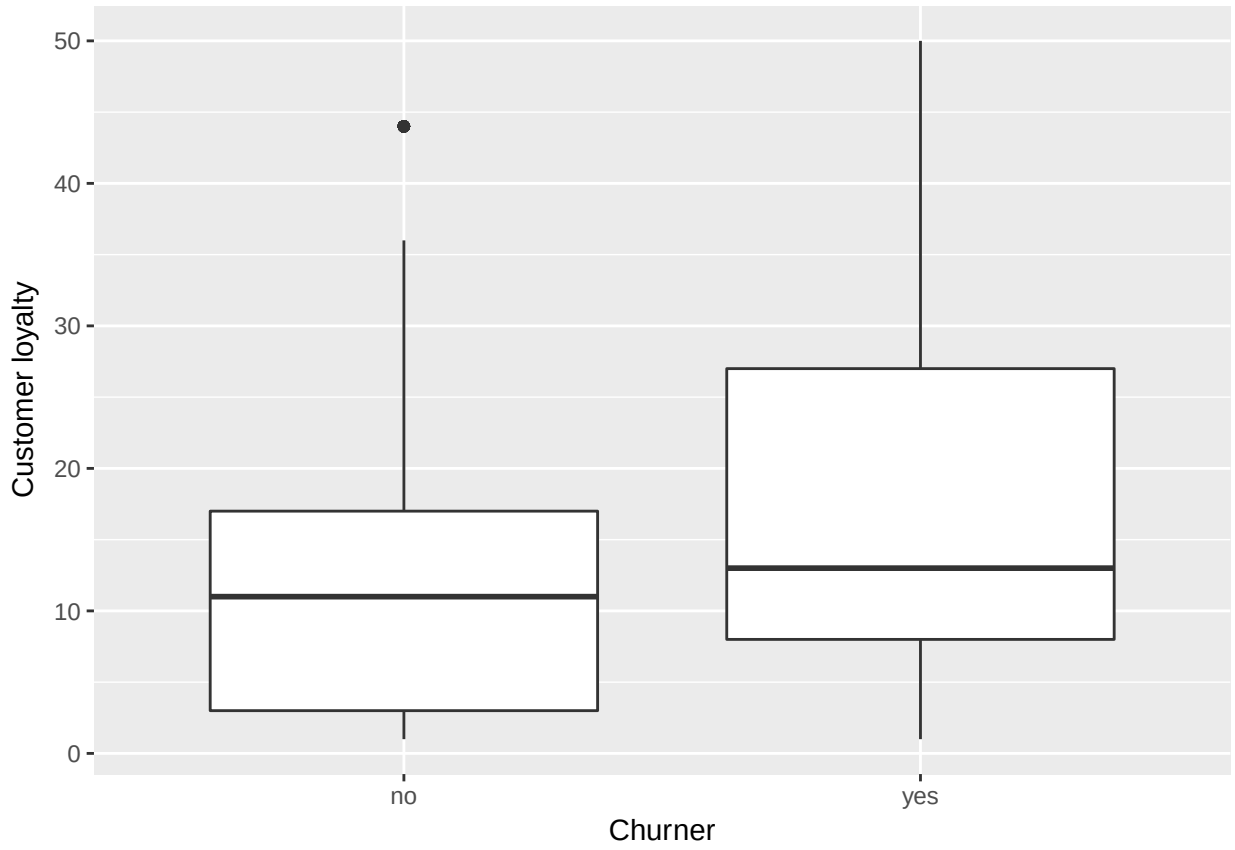
```
tot_minutes_usage_skew <- (3 * (mean(phones$TOT_MINUTES_USAGE) -
    median(phones$TOT_MINUTES_USAGE)))/sd(phones$TOT_MINUTES_USAGE)
tot_minutes_usage_skew
```

```
## [1] 1.088757
```

3.g

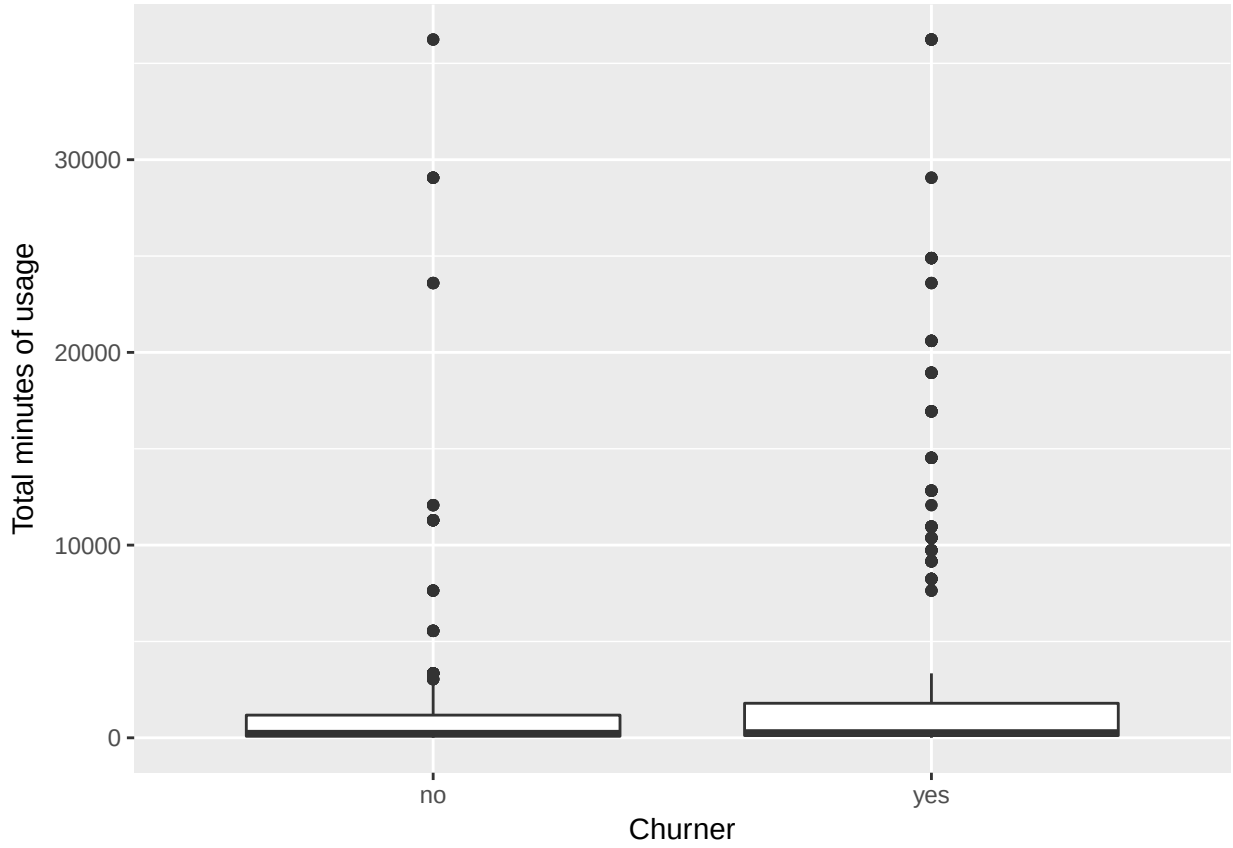
CUST_MOS

```
ggplot(data = phones, aes(phones$CHURNER, phones$CUST_MOS)) +
    geom_boxplot() + labs(x = "Churner", y = "Customer loyalty")
```



TOT_MINUTES_USAGE

```
ggplot(data = phones, aes(phones$CHURNER, phones$TOT_MINUTES_USAGE)) +
    geom_boxplot() + labs(x = "Churner", y = "Total minutes of usage")
```



4. Finding outliers mathematically in TOT_MINUTES_USAGE

IQR method

```
summary(phones$TOT_MINUTES_USAGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     116     264    2036    1677   36240
```

```
IQR <- 1677 - 116
lower_bound <- 116 - (IQR * 1.5)
upper_bound <- 1677 + (IQR * 1.5)

nrow(phones[phones$TOT_MINUTES_USAGE < lower_bound | phones$TOT_MINUTES_USAGE >
    upper_bound, ])
```

```
## [1] 176
```

Z standardisation method

```
# Z score
z_score_tot_minutes_usage <- scale(phones$TOT_MINUTES_USAGE,
    center = TRUE, scale = TRUE)
# same as (phones$TOT_MINUTES_USAGE -
# mean(phones$TOT_MINUTES_USAGE))/sd(phones$TOT_MINUTES_USAGE)
summary(z_score_tot_minutes_usage)
```

```
##        V1
##  Min.   :-0.41698
##  1st Qu.:-0.39323
```

```
##  Median :-0.36292
##  Mean   : 0.00000
##  3rd Qu.:-0.07354
##  Max.   : 7.00417
```

```
z_range <- table(z_score_tot_minutes_usage > -3 & z_score_tot_minutes_usage <
    3)
z_range[names(z_range) == FALSE]
```

```
## FALSE
##    69
```

5

```
tot_mins_before_transfo <- (3 * (mean(phones$TOT_MINUTES_USAGE) -
    median(phones$TOT_MINUTES_USAGE)))/sd(phones$TOT_MINUTES_USAGE)
tot_mins_before_transfo
```

```
## [1] 1.088757
```

5.a

Z-score standardisation see above, we reduced the number of outliers from 176 to 69

```
tot_mins_z_score <- (3 * (mean(z_score_tot_minutes_usage) - median(z_score_tot_minutes_usage)))/sd(z_sc
tot_mins_z_score
```

```
## [1] 1.088757
```

5.b

Natural log

```
natural_log_transfo <- log(phones$TOT_MINUTES_USAGE[phones$TOT_MINUTES_USAGE !=
    0])
natural_log_transfo_skewness <- (3 * (mean(natural_log_transfo) -
    median(natural_log_transfo)))/sd(natural_log_transfo)

natural_log_transfo_skewness
```

```
## [1] -0.7042918
```

5.c

Square root

```
square_root_transfo <- sqrt(phones$TOT_MINUTES_USAGE)
square_root_transfo_skewness <- (3 * (mean(square_root_transfo) -
    median(square_root_transfo)))/sd(square_root_transfo)

square_root_transfo_skewness
```

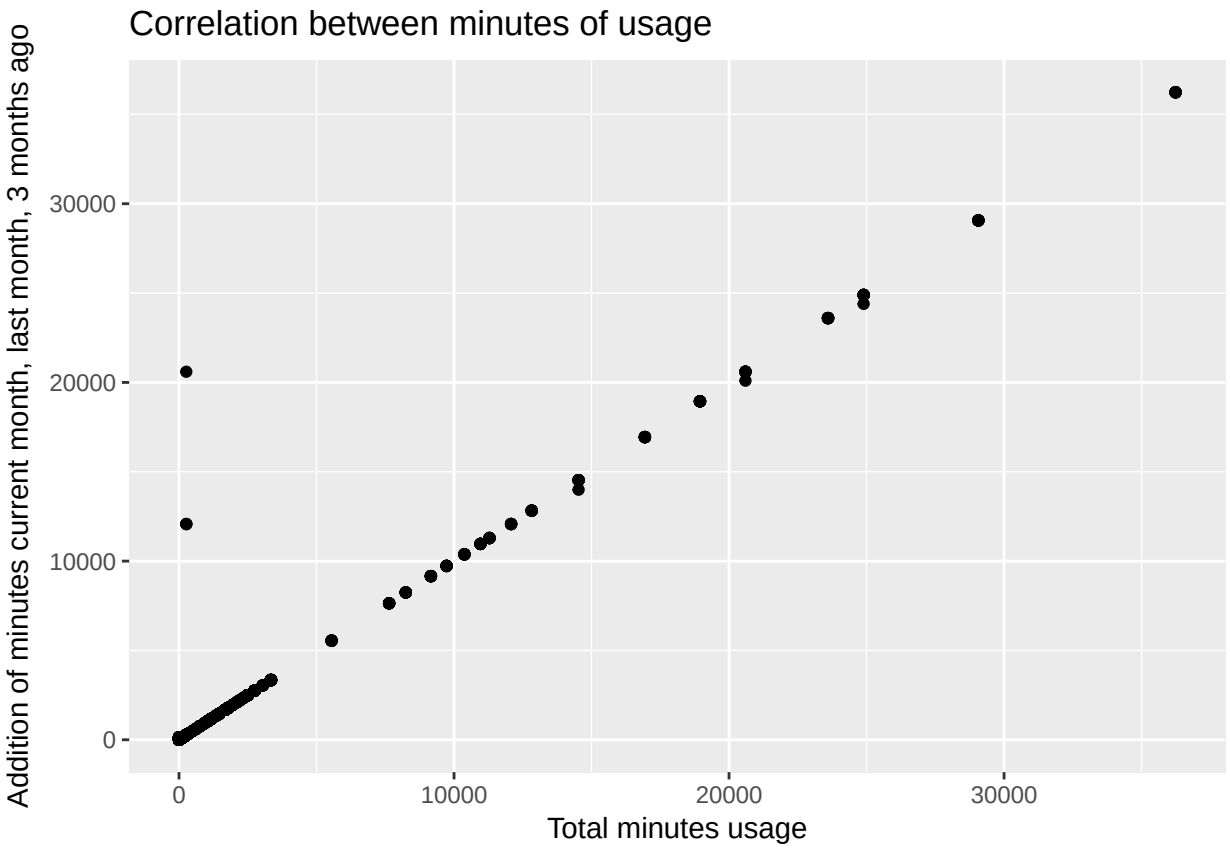
```
## [1] 1.288432
```

7.a.

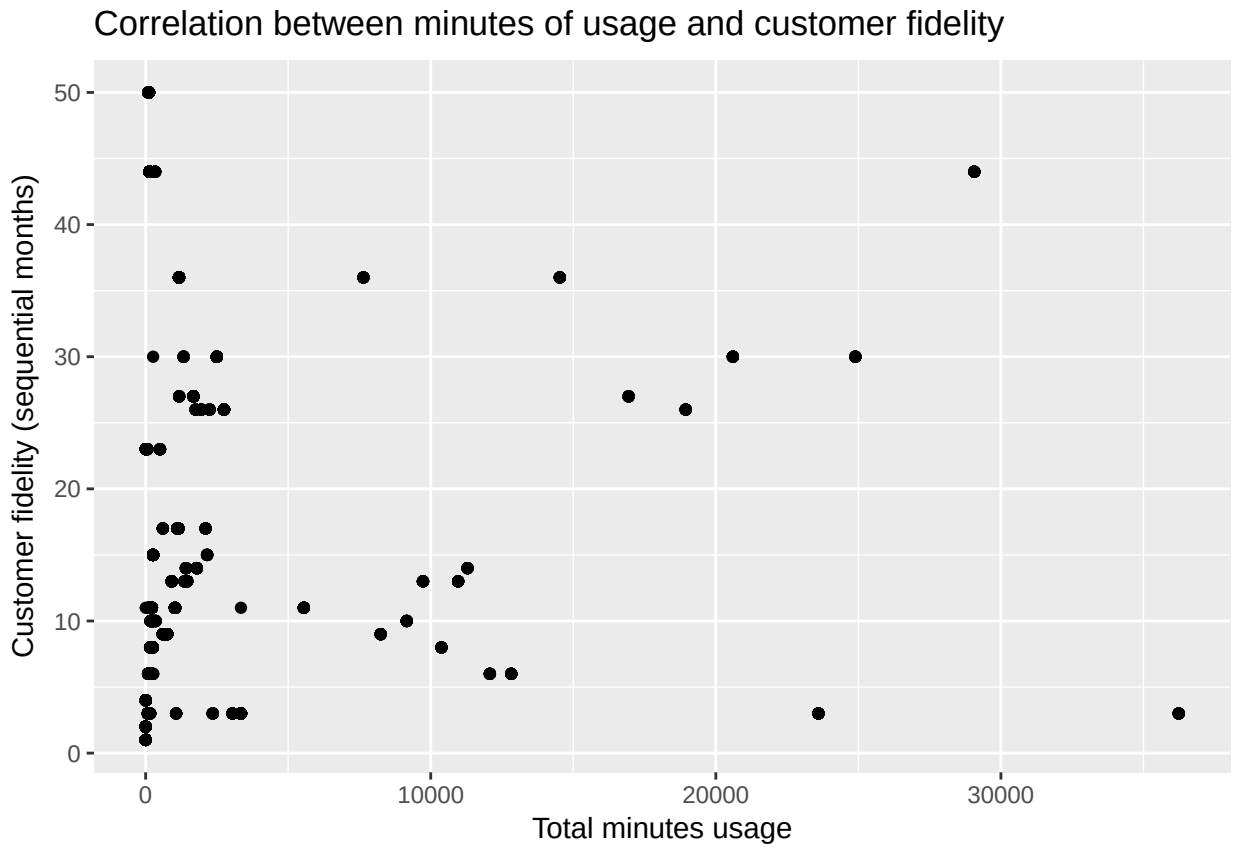
Correlation

Minutes of usage

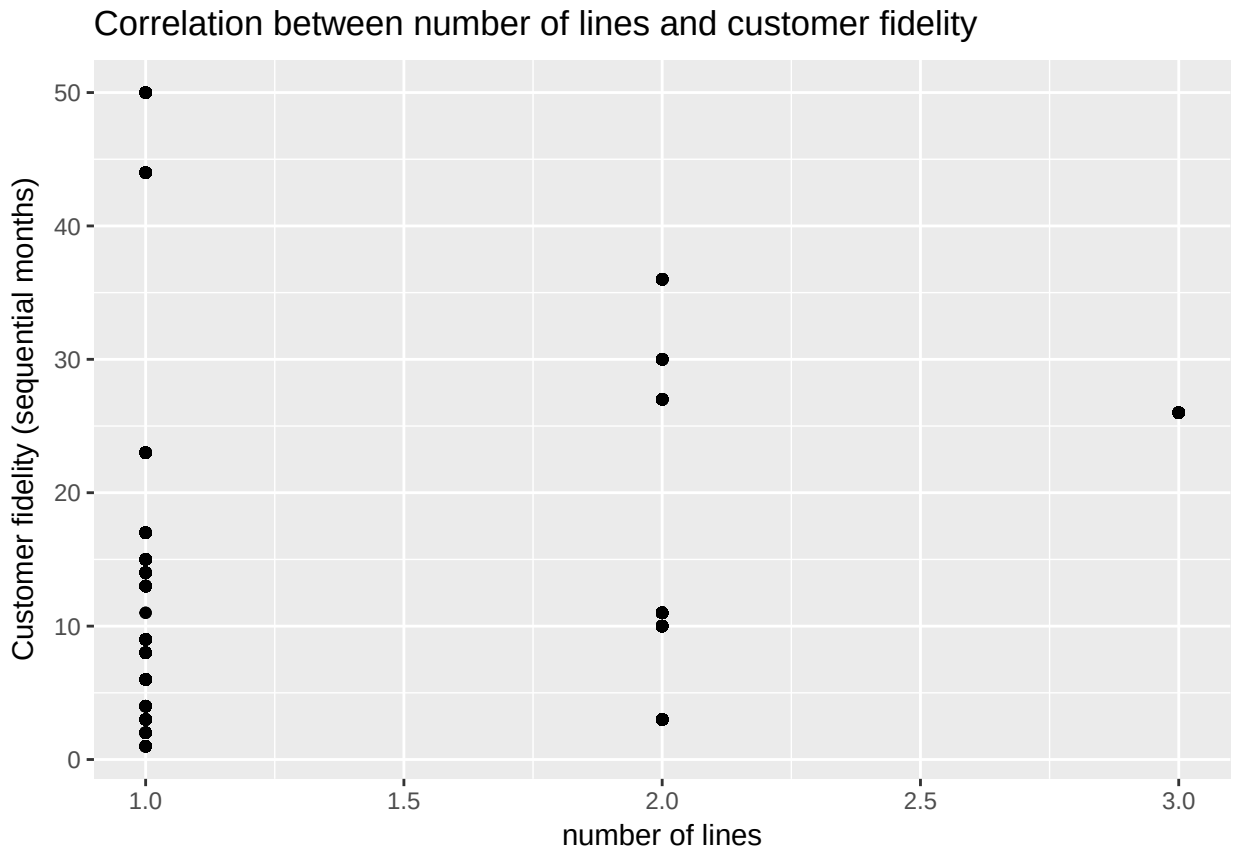
```
ggplot(data = phones, aes(x = phones$TOT_MINUTES_USAGE, y = phones$MINUTES_CURR_MONTH +
    phones$MINUTES_PREV_MONTH + phones$MINUTES_3MONTHS_AGO)) +
    geom_point() + labs(x = "Total minutes usage", y = "Addition of minutes current month, last month, 3
    title = "Correlation between minutes of usage")
```



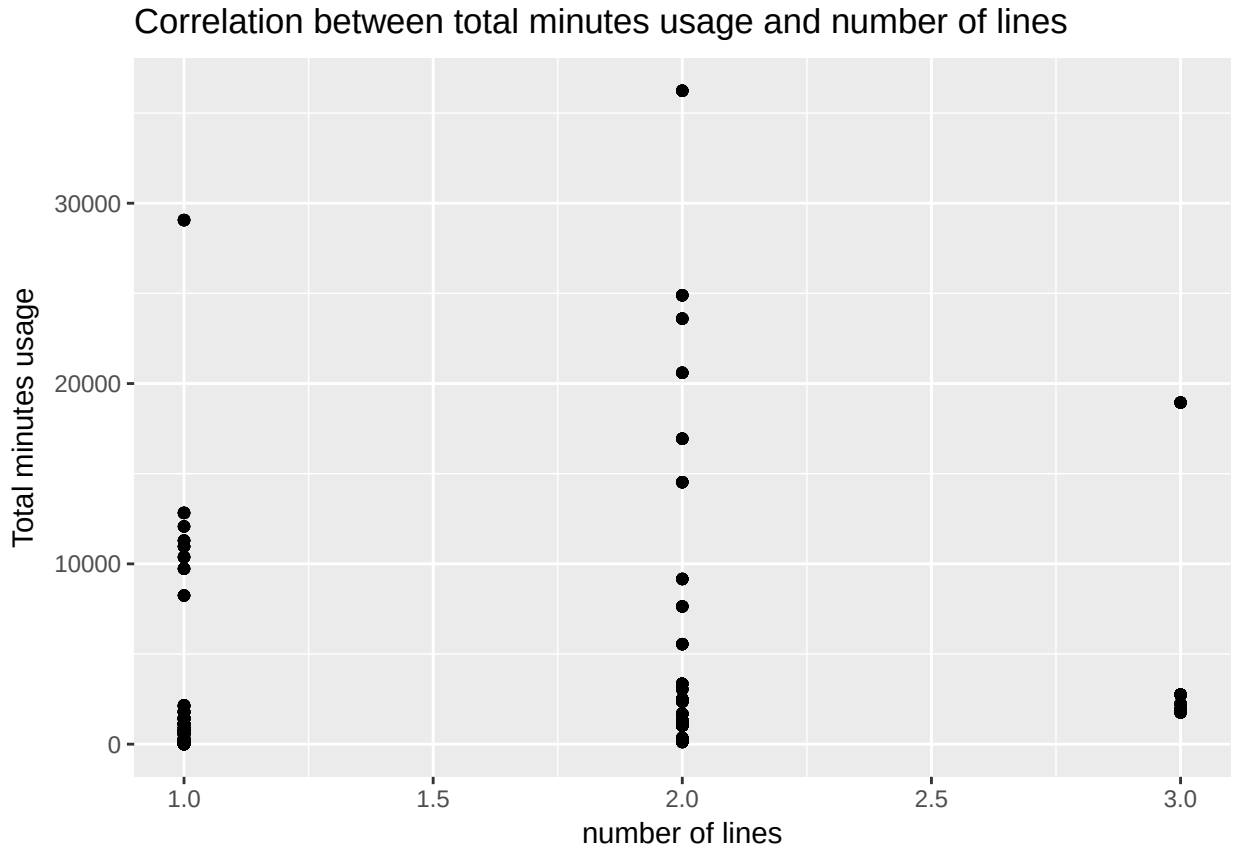

```
ggplot(data = phones, aes(x = phones$TOT_MINUTES_USAGE, y = phones$CUST_MOS)) +
    geom_point() + labs(y = "Customer fidelity (sequential months)",
    x = "Total minutes usage", title = "Correlation between minutes of usage and customer fidelity")
```




```
ggplot(data = phones, aes(x = phones$NUM_LINES, y = phones$CUST_MOS)) +
    geom_point() + labs(y = "Customer fidelity (sequential months)",
    x = "number of lines", title = "Correlation between number of lines and customer fidelity")
```




```
ggplot(data = phones, aes(x = phones$NUM_LINES, y = phones$TOT_MINUTES_USAGE)) +
    geom_point() + labs(y = "Total minutes usage", x = "number of lines",
    title = "Correlation between total minutes usage and number of lines")
```



7.b

Minutes usage metrics correlation

```
covariance_minutes <- cov(phones$TOT_MINUTES_USAGE, phones$MINUTES_CURR_MONTH +
    phones$MINUTES_PREV_MONTH + phones$MINUTES_3MONTHS_AGO)
covariance_minutes
```

```
## [1] 23778254
```

```
correlation_minutes <- covariance_minutes/(sd(phones$TOT_MINUTES_USAGE) *
    sd(phones$MINUTES_CURR_MONTH + phones$MINUTES_PREV_MONTH +
        phones$MINUTES_3MONTHS_AGO))
correlation_minutes
```

```
## [1] 0.9916396
```

Usage and customer fidelity

```
covariance_minutes_fid <- cov(phones$TOT_MINUTES_USAGE, phones$CUST_MOS)
covariance_minutes_fid
```

```
## [1] 5931.69
```

```
correlation_minutes_fid <- covariance_minutes_fid/(sd(phones$TOT_MINUTES_USAGE) *
    sd(phones$CUST_MOS))
correlation_minutes_fid
```

```
## [1] 0.09075367
```

```
covariance_lines_fid <- cov(phones$NUM_LINES, phones$CUST_MOS)
covariance_lines_fid
```

```
## [1] 1.550566
```

```
correlation_lines_fid <- covariance_lines_fid/(sd(phones$NUM_LINES) *
    sd(phones$CUST_MOS))
correlation_lines_fid
```

```
## [1] 0.2031285
```

```
covariance_lines_minutes <- cov(phones$NUM_LINES, phones$TOT_MINUTES_USAGE)
covariance_lines_minutes
```

```
## [1] 685.4576
```

```
correlation_lines_minutes <- covariance_lines_minutes/(sd(phones$NUM_LINES) *
    sd(phones$TOT_MINUTES_USAGE))
correlation_lines_minutes
```

```
## [1] 0.2461581
```

Part 2

Preparating Data for learning

```
keep <- c("INCOME", "PHONE_PLAN", "EDUCATION", "AREA_CODE", "CUS_MOS",
    "CHURNER", "CONVERGENT_BILLING")
phones_learning <- phones[, (names(phones) %in% keep)]
phones_learning$AREA_CODE <- as.factor(phones_learning$AREA_CODE)
head(phones_learning)
```

```
##   AREA_CODE CONVERGENT_BILLING        INCOME    PHONE_PLAN    EDUCATION
## 1     45987                Yes Medium Income International      Masters
## 2     15563                Yes Medium Income International    Bachelors
## 3     10040                 No    Low Income      National  High School
## 4     21750                Yes Medium Income International  High School
## 5     55166                 No   High Income    Promo_plan  High School
## 6     36785                 No   High Income      National Post Primary
##   CHURNER
## 1     yes
## 2      no
## 3      no
## 4     yes
## 5      no
## 6      no
```

Writing the learning data to csv

```
write.csv(phones_learning, file = "./learning_churners.csv")
```