Healtcare Appointment No-Show

DESCRIPTION

A patient is considered to be a no-show when they fail to be present for a scheduled appointment. For any healthcare organization, no-shows lead to higher costs and underutilization of resources, which affects the quality of service healthcare organizations provide.

In order to solve this problem, the organizations need to be able to understand why no-show patients do so.

Task 1 : Explore the data to check for missing values or erroneous entries, comment on redundant features, and add additional ones if needed.

```
# Import excel
library(readxl)
appointments <- read_excel("appointments.xls")
# check structure of the data
str(appointments)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    65535 obs. of  13 variables:
##  $ Age                    : num  19 24 4 5 38 5 46 4 20 51 ...
##  $ Gender                 : chr  "M" "F" "F" "M" ...
##  $ AppointmentRegistration: chr  "2014-12-16T14:46:25Z" "2015-08-18T07:01:26Z" "2014-02-17T12:53:46Z
##  $ AppointmentDate        : chr  "2015-01-14T00:00:00Z" "2015-08-19T00:00:00Z" "2014-02-18T00:00:00Z
##  $ Diabetes               : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Alcoholism             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ HyperTension           : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Handicap               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Smokes                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Scholarship            : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Tuberculosis           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sms_Reminder           : num  0 0 0 1 1 1 1 1 0 1 ...
##  $ Status                 : chr  "Show-Up" "Show-Up" "Show-Up" "Show-Up" ...
```

```
# column names
names(appointments)
```

```
##  [1] "Age"                    "Gender"
##  [3] "AppointmentRegistration" "AppointmentDate"
##  [5] "Diabetes"               "Alcoholism"
##  [7] "HyperTension"           "Handicap"
##  [9] "Smokes"                 "Scholarship"
## [11] "Tuberculosis"           "Sms_Reminder"
## [13] "Status"
```

```
# convert date columns into character
appointments$AppointmentRegistration = as.POSIXct(appointments$AppointmentRegistration)
appointments$AppointmentDate = as.POSIXct(appointments$AppointmentDate)
```

```r
# Convert columns to factors
appointments$Gender = as.factor(appointments$Gender)
appointments$Status = as.factor(appointments$Status)
appointments$Diabetes = as.factor(appointments$Diabetes)
appointments$Alcoholism = as.factor(appointments$Alcoholism)
appointments$HyperTension = as.factor(appointments$HyperTension)
appointments$Handicap = as.factor(appointments$Handicap)
appointments$Smokes = as.factor(appointments$Smokes)
appointments$Scholarship = as.factor(appointments$Scholarship)
appointments$Tuberculosis = as.factor(appointments$Tuberculosis)
appointments$Sms_Reminder = as.factor(appointments$Sms_Reminder)

summary(appointments)  # summary of the data
```

```
##       Age          Gender    AppointmentRegistration
##  Min.   : -1.00   F:43765   Min.   :2013-05-29 00:00:00
##  1st Qu.: 19.00   M:21770   1st Qu.:2014-06-20 00:00:00
##  Median : 38.00             Median :2014-12-02 00:00:00
##  Mean   : 37.75             Mean   :2014-12-13 15:26:42
##  3rd Qu.: 56.00             3rd Qu.:2015-06-10 00:00:00
##  Max.   :113.00             Max.   :2015-12-29 00:00:00
##  AppointmentDate               Diabetes  Alcoholism HyperTension Handicap
##  Min.   :2014-01-02 00:00:00   0:60418   0:63898    0:51446      0:64293
##  1st Qu.:2014-07-03 00:00:00   1: 5117   1: 1637    1:14089      1: 1129
##  Median :2014-12-15 00:00:00                                     2:  100
##  Mean   :2014-12-27 10:18:31                                     3:   11
##  3rd Qu.:2015-06-25 00:00:00                                     4:    2
##  Max.   :2015-12-30 00:00:00
##  Smokes    Scholarship Tuberculosis Sms_Reminder     Status
##  0:62071   0:59160     0:65510      0:28255      No-Show:19871
##  1: 3464   1: 6375     1:   25      1:37092      Show-Up:45664
##                                     2:  188
##
##
##
```

```r
# check for null values
sapply(appointments, function(x) sum(is.na(x)))
```

```
##                      Age                   Gender AppointmentRegistration
##                        0                        0                        0
##          AppointmentDate                 Diabetes               Alcoholism
##                        0                        0                        0
##             HyperTension                 Handicap                   Smokes
##                        0                        0                        0
##              Scholarship             Tuberculosis             Sms_Reminder
##                        0                        0                        0
##                   Status
##                        0
```

Task 2 : Create a new feature called HourOfTheDay, which will indicate the hour of the day at which the appointment was booked.

```r
# create 3 new features for AppointmentRegistration
appointments <- transform(appointments, Day_Reg = format(AppointmentRegistration, "%d"))
appointments <-transform(appointments, Month_Reg = format(AppointmentRegistration, "%m"))
appointments <- transform(appointments, Year_Reg = format(AppointmentRegistration, "%Y"))

appointments$Day_Reg = as.factor(appointments$Day_Reg)
appointments$Month_Reg = as.factor(appointments$Month_Reg)
appointments$Year_Reg = as.factor(appointments$Year_Reg)

summary(appointments)  # summary of the data
```

```
##       Age          Gender    AppointmentRegistration
##  Min.   : -1.00   F:43765   Min.   :2013-05-29 00:00:00
##  1st Qu.: 19.00   M:21770   1st Qu.:2014-06-20 00:00:00
##  Median : 38.00             Median :2014-12-02 00:00:00
##  Mean   : 37.75             Mean   :2014-12-13 15:26:42
##  3rd Qu.: 56.00             3rd Qu.:2015-06-10 00:00:00
##  Max.   :113.00             Max.   :2015-12-29 00:00:00
##
##  AppointmentDate               Diabetes  Alcoholism HyperTension Handicap
##  Min.   :2014-01-02 00:00:00   0:60418   0:63898    0:51446      0:64293
##  1st Qu.:2014-07-03 00:00:00   1: 5117   1: 1637    1:14089      1: 1129
##  Median :2014-12-15 00:00:00                                     2:  100
##  Mean   :2014-12-27 10:18:31                                     3:   11
##  3rd Qu.:2015-06-25 00:00:00                                     4:    2
##  Max.   :2015-12-30 00:00:00
##
##  Smokes    Scholarship Tuberculosis Sms_Reminder     Status
##  0:62071   0:59160     0:65510      0:28255      No-Show:19871
##  1: 3464   1: 6375     1:   25      1:37092      Show-Up:45664
##                                     2:  188
##
##
##
##
##     Day_Reg        Month_Reg     Year_Reg
##  24     : 2523   07     : 6048   2013:  844
##  27     : 2486   10     : 5928   2014:33904
##  20     : 2427   05     : 5902   2015:30787
##  10     : 2393   09     : 5698
##  06     : 2373   08     : 5693
##  03     : 2328   01     : 5628
##  (Other):51005   (Other):30638
```

Task 3 : Identify and remove outliers from the age column and explain the reason behind the selected outlier treatment using an appropriate plot.

```r
# We see from the summary(appointments) that the minimum age is -1.
# So we check all the values less than 0.
appointments[appointments$Age < 0, ]
```

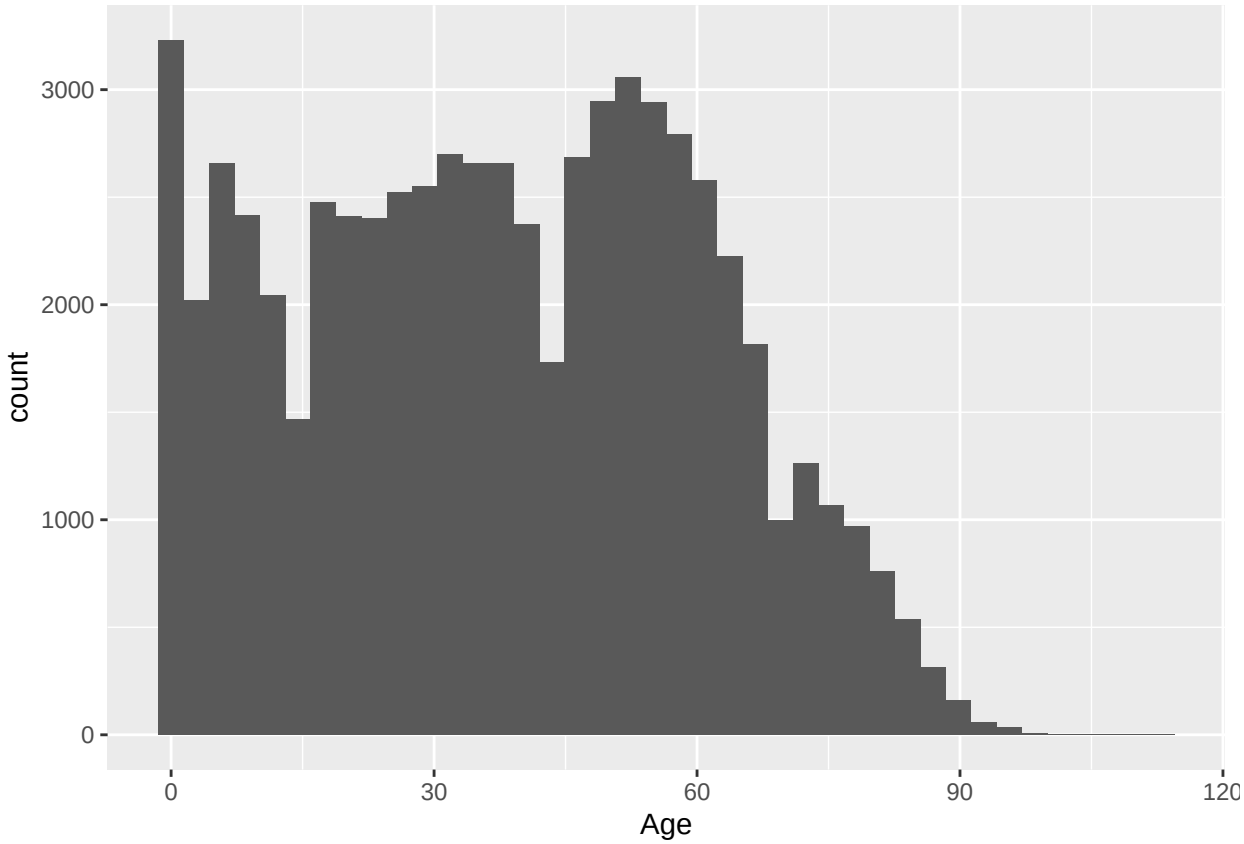
```
##       Age Gender AppointmentRegistration AppointmentDate Diabetes Alcoholism
```

```
## 63391  -1      F              2014-03-14      2014-03-21        0          0
##       HyperTension Handicap Smokes Scholarship Tuberculosis Sms_Reminder
## 63391            0        0      0           0             0            1
##        Status Day_Reg Month_Reg Year_Reg
## 63391 No-Show      14        03     2014
```

```
# It is only one and we will drop it.
appointments <-appointments[!(appointments$Age<0),]
summary(appointments$Age) # summary of the column Age
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   19.00   38.00   37.75   56.00  113.00
```

```
# Plot
library(ggplot2)
ggplot(appointments, aes(x=Age)) + geom_histogram(bins=40)
```



Task 4 : Analyze the probability of showing up with respect to different features. Create a scatter plot and trend lines to analyze the relation between the probability of showing up with respect to age or hour of the day, and describe your findings.

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

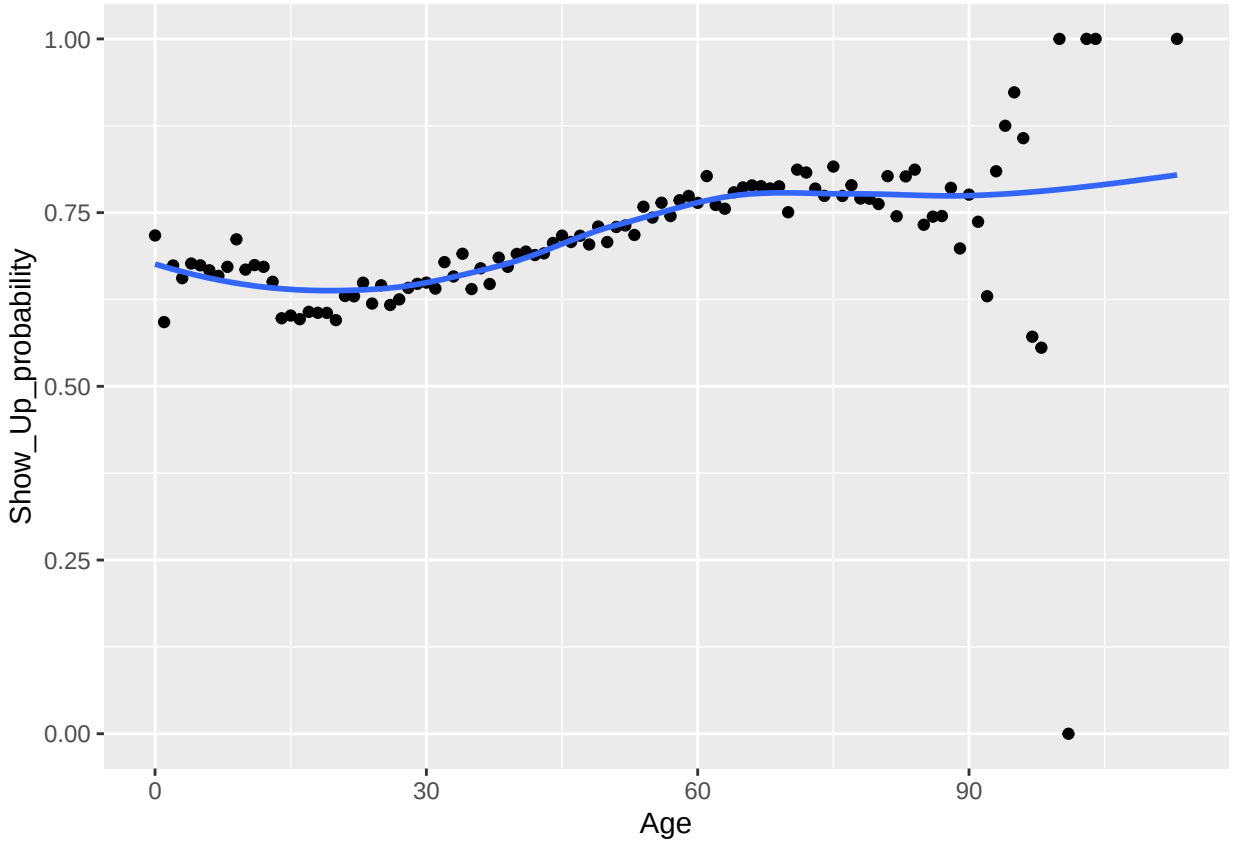
```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
appointments %>% group_by(Age) %>%
  summarise(Show_Up_probability=sum(Status=="Show-Up")/n()) %>%
ggplot(aes(x=Age, y=Show_Up_probability)) + geom_point() + geom_smooth(se=F)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



We see that the older people show-up more often that younger ones.

Task 5 : Create a bar graph to depict the probability of showing up for diabetes, alcoholism, hypertension, TB, smokes, and scholarship.

```
# check probability stats about these features
prop.table(table(appointments$Diabetes, appointments$Status))
```

```
##
```

```
##        No-Show    Show-Up
##   0 0.28318125 0.63873714
##   1 0.02002014 0.05806146
```

```r
prop.table(table(appointments$Alcoholism, appointments$Status))
```

```
##
##         No-Show     Show-Up
##   0 0.294091617 0.680928983
##   1 0.009109775 0.015869625
```

```r
prop.table(table(appointments$HyperTension, appointments$Status))
```

```
##
##        No-Show    Show-Up
##   0 0.24962615 0.53538621
##   1 0.05357524 0.16141240
```

```r
prop.table(table(appointments$Tuberculosis, appointments$Status))
```

```
##
##          No-Show      Show-Up
##   0 0.3030182806 0.6966002380
##   1 0.0001831111 0.0001983703
```

```r
prop.table(table(appointments$Smokes, appointments$Status))
```

```
##
##        No-Show    Show-Up
##   0 0.28510392 0.66203803
##   1 0.01809748 0.03476058
```

```r
prop.table(table(appointments$Scholarship, appointments$Status))
```

```
##
##        No-Show    Show-Up
##   0 0.26819666 0.63452559
##   1 0.03500473 0.06227302
```

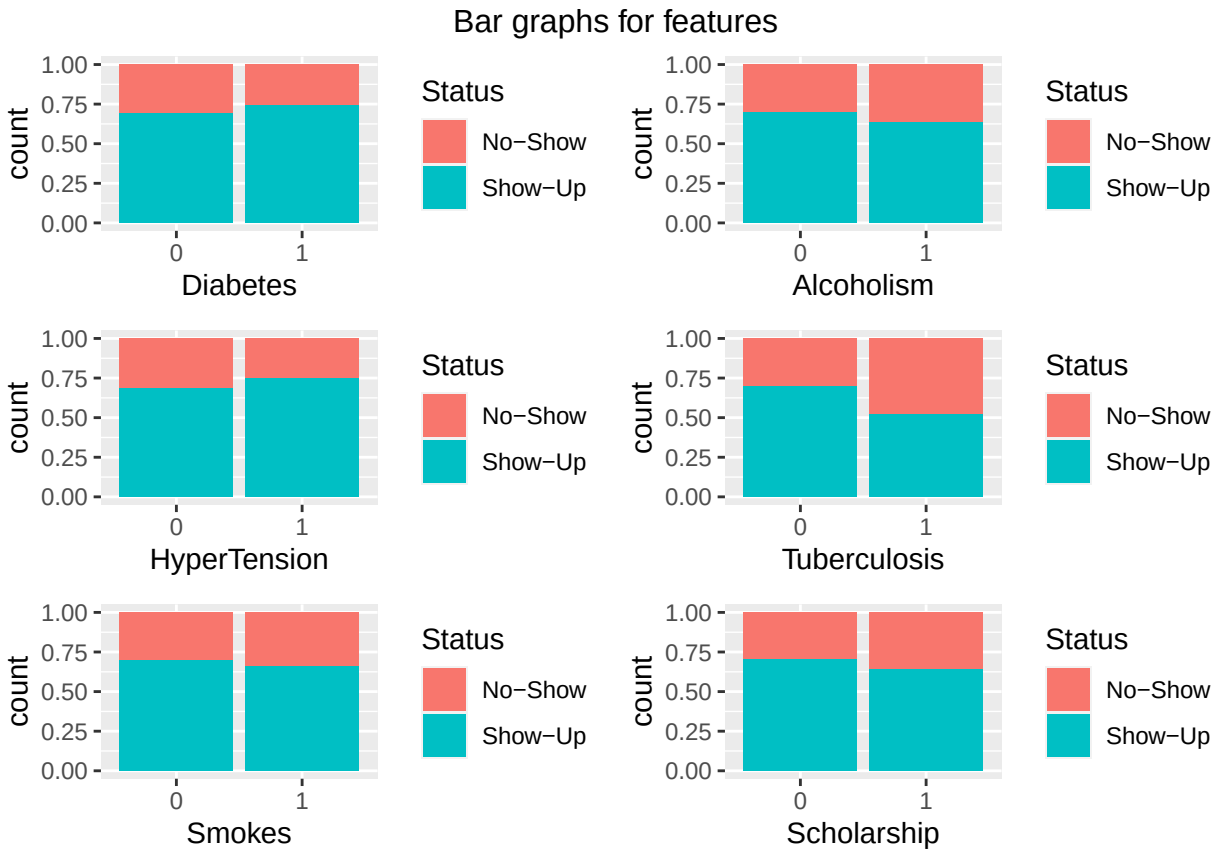
```r
# Plots
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

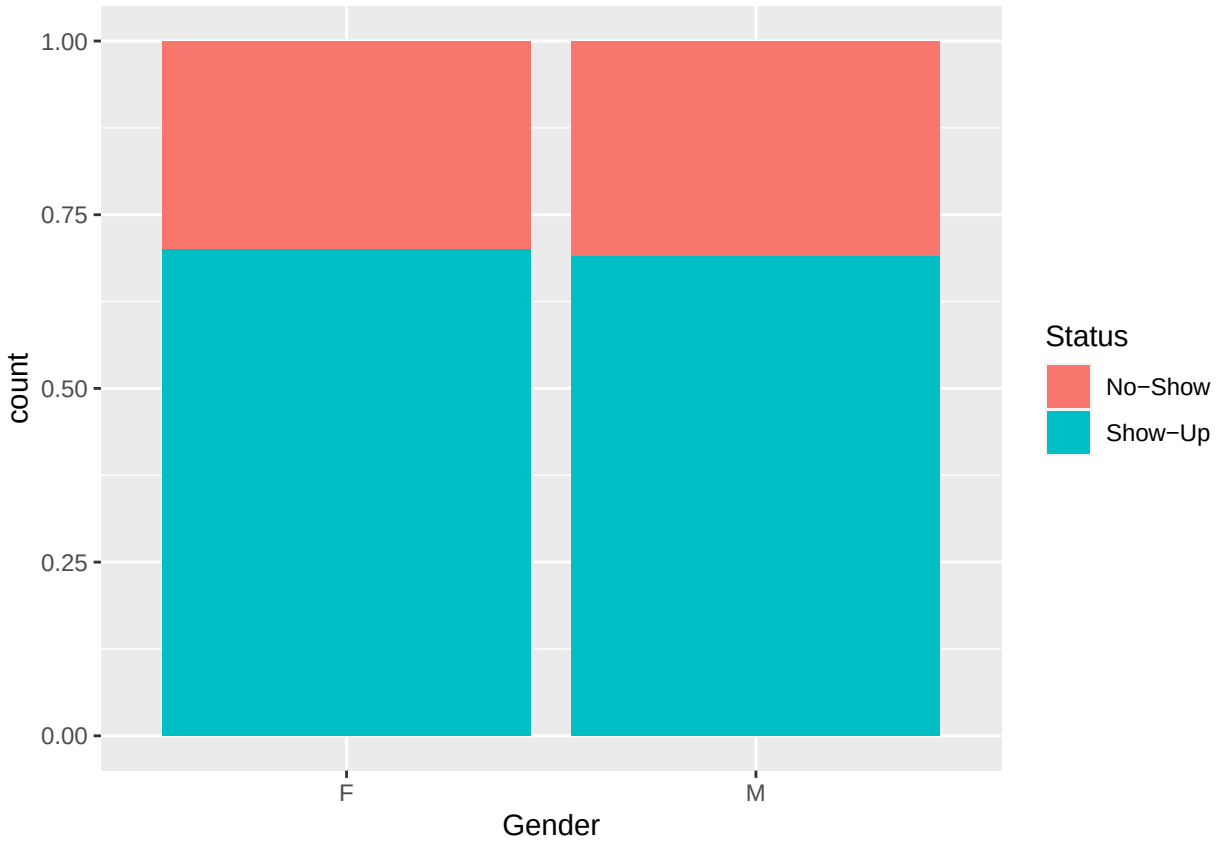
```
library(ggplot2)
g_Diabetes <- ggplot(appointments, aes(x=Diabetes, fill=Status)) + geom_bar(position="fill")
g_Alcoholism <- ggplot(appointments, aes(x=Alcoholism, fill=Status)) + geom_bar(position="fill")
g_Hypertension <- ggplot(appointments, aes(x=HyperTension, fill=Status)) + geom_bar(position="fill")
g_TB <- ggplot(appointments, aes(x=Tuberculosis, fill=Status)) + geom_bar(position="fill")
g_Smokes <- ggplot(appointments, aes(x=Smokes, fill=Status)) + geom_bar(position="fill")
g_Scholarship <- ggplot(appointments, aes(x=Scholarship, fill=Status)) + geom_bar(position="fill")

grid.arrange(g_Diabetes, g_Alcoholism, g_Hypertension, g_TB, g_Smokes,  g_Scholarship, ncol=2, top='Bar
```



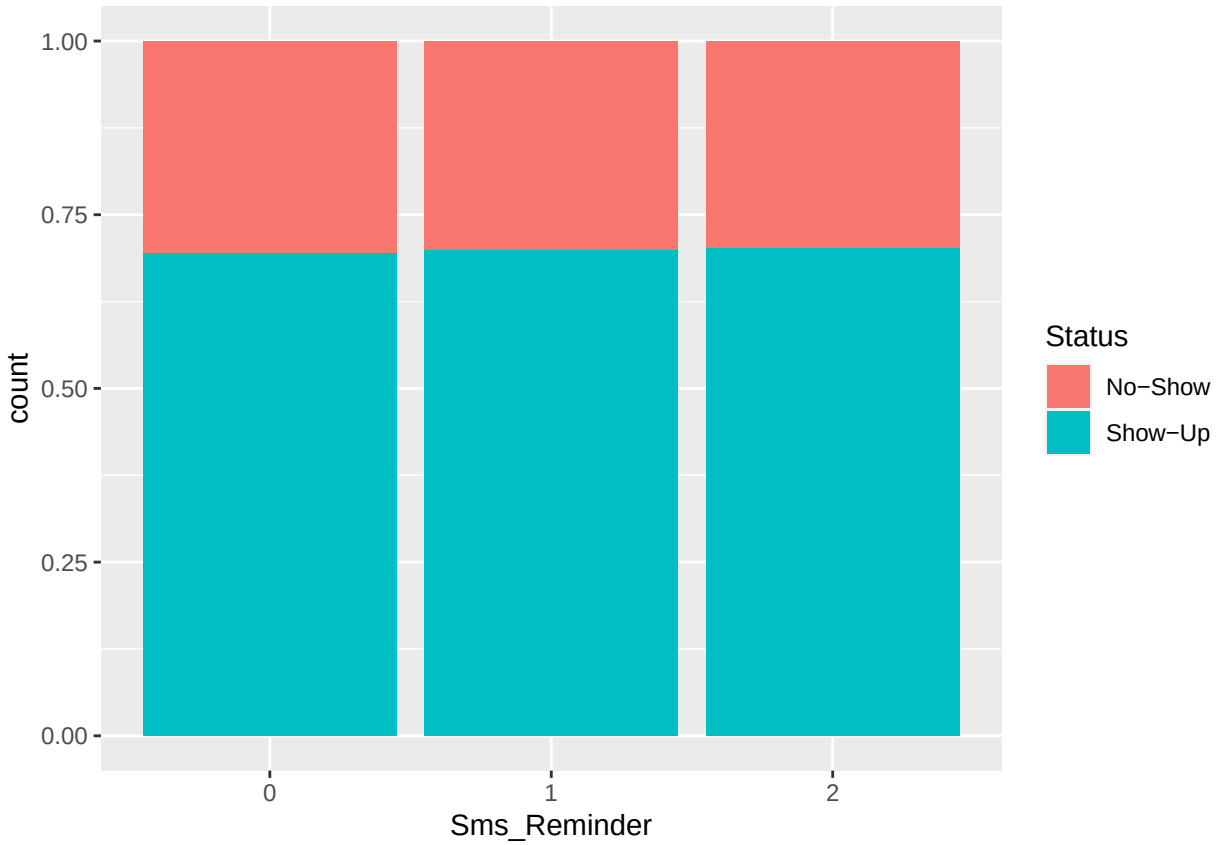
Task 6 : Create separate bar graphs to show the probability of showing up with respect to male or female, day of the week, and SMS reminder columns and describe your findings.

```
# Bar graph to show the probability of showing up with respect to male or female
ggplot(appointments, aes(x=Gender, fill=Status)) + geom_bar(position="fill")
```



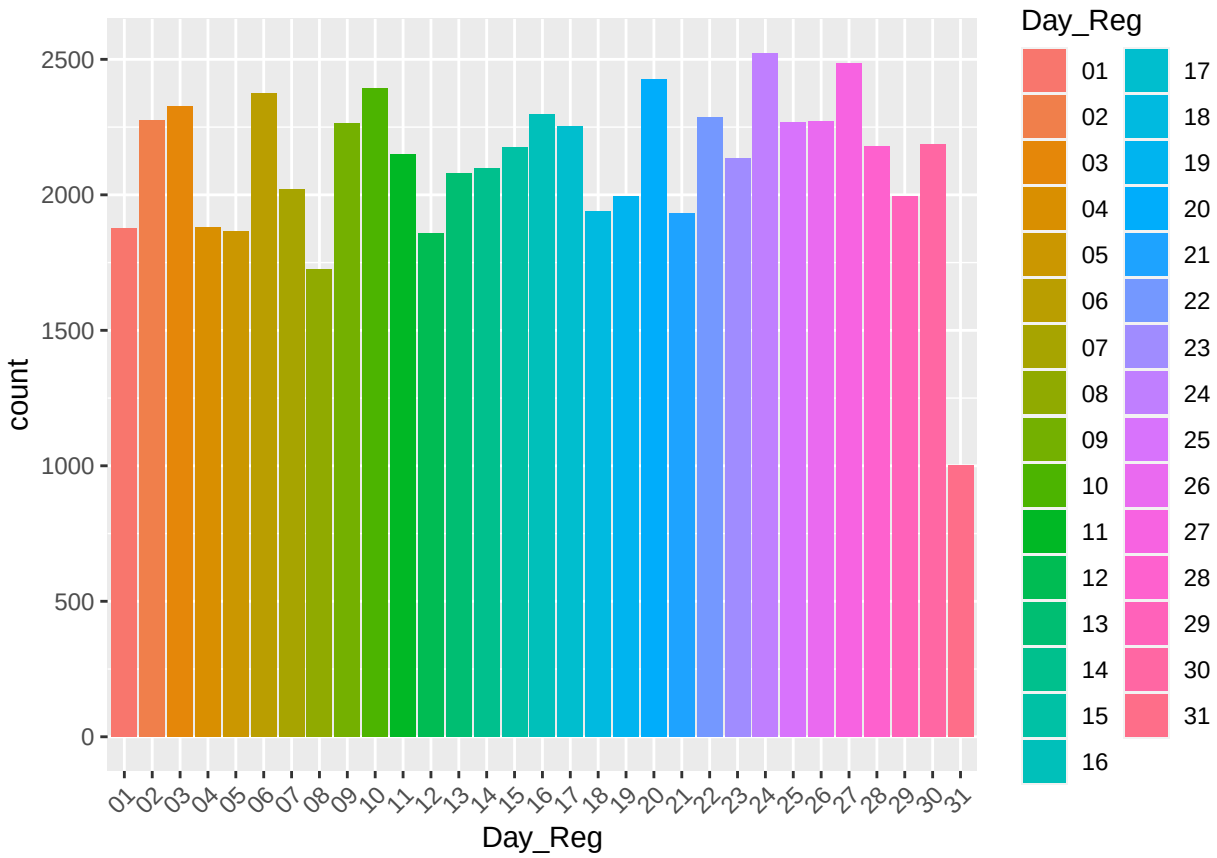
It seems like the probability both for men and women to Show Up is similar.

```
# Bar graph to show the probability of showing up with respect to SMS reminder
ggplot(appointments, aes(x=Sms_Reminder, fill=Status)) + geom_bar(position="fill")
```



It seems like Sms reminder is not important enough.

```
ggplot(appointments, aes(x=Day_Reg, fill=Day_Reg )) + geom_bar() + theme(axis.text.x = element_text(ang
```



We see that the day with the maximum probability to show-up in the month is the 24th.

Task 7 : Use different classification models to predict the show or no-show status based on the features that display the most variation in the probability of showing up.

Logistic Regression Model

```
appointments_2 <- select(appointments, Age, Gender, Scholarship, HyperTension, Diabetes, Alcoholism, Ha
```

```
appointments_2 <- mutate_at(appointments_2, vars(Status), as.factor)
```

```
log_model <- glm(Status ~ . ,family = binomial(link = 'logit'),  data = appointments_2 )
```

```
summary(log_model)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial(link = "logit"),
##     data = appointments_2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8513  -1.4290   0.7801   0.8823   1.1661
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)    0.4949346  0.0216372  22.874  < 2e-16 ***
## Age            0.0089252  0.0004489  19.882  < 2e-16 ***
## GenderM       -0.0023025  0.0185141  -0.124   0.9010
## Scholarship1  -0.2060499  0.0281378  -7.323 2.43e-13 ***
## HyperTension1  0.1082088  0.0268170   4.035 5.46e-05 ***
## Diabetes1     -0.0477623  0.0371552  -1.285   0.1986
## Alcoholism1   -0.3882182  0.0529176  -7.336 2.20e-13 ***
## Handicap1      0.0456762  0.0688012   0.664   0.5068
## Handicap2     -0.1520866  0.2184526  -0.696   0.4863
## Handicap3      0.8186787  0.7838481   1.044   0.2963
## Handicap4      9.0012787 51.2167987   0.176   0.8605
## Sms_Reminder1  0.0391626  0.0173186   2.261   0.0237 *
## Sms_Reminder2 -0.0025007  0.1610072  -0.016   0.9876
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 80417  on 65533  degrees of freedom
## Residual deviance: 79610  on 65521  degrees of freedom
## AIC: 79636
##
## Number of Fisher Scoring iterations: 8
```

```
# Logistic Regression with training and test data set
library(caTools)
set.seed(100)

split = sample.split(appointments_2$Status, SplitRatio = 0.70)

train = subset(appointments_2, split == TRUE)
test = subset(appointments_2, split == FALSE)
```

```
logit_model <- glm(formula = Status ~ . ,  data = train, family =binomial(link = 'logit') )
summary(logit_model)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8462  -1.4297   0.7799   0.8835   1.1792
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.4991944  0.0258605  19.303  < 2e-16 ***
## Age            0.0088294  0.0005365  16.457  < 2e-16 ***
## GenderM        0.0072848  0.0221585   0.329 0.742339
## Scholarship1  -0.2072324  0.0336343  -6.161 7.21e-10 ***
## HyperTension1  0.1124068  0.0320492   3.507 0.000453 ***
## Diabetes1     -0.0297095  0.0445645  -0.667 0.504986
```

```
## Alcoholism1   -0.4539482  0.0624797  -7.266 3.72e-13 ***
## Handicap1     -0.0248609  0.0811840  -0.306 0.759430
## Handicap2     -0.2234274  0.2562901  -0.872 0.383331
## Handicap3      0.7595818  1.1184428   0.679 0.497048
## Handicap4      8.9930732 51.2130135   0.176 0.860607
## Sms_Reminder1  0.0340616  0.0206887   1.646 0.099684 .
## Sms_Reminder2  0.1065692  0.1920153   0.555 0.578892
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 56292  on 45873  degrees of freedom
## Residual deviance: 55720  on 45861  degrees of freedom
## AIC: 55746
##
## Number of Fisher Scoring iterations: 8
```

Task 8 : Evaluate the models and choose the best one for the data.

```
fitted_p <- predict(logit_model,newdata=test,type='response')
head(fitted_p)
```

```
##         6         8        13        14        15        19
## 0.6404696 0.5893650 0.7557121 0.7466222 0.7060045 0.7651336
```

```
pred_test <-  ifelse(fitted_p>0.5,1,0)
```

```
tab <- table(predicted = pred_test, actual = test$Status)
tab
```

```
##          actual
## predicted No-Show Show-Up
##         0       2       2
##         1    5959   13697
```

We see that there is nearly 80 % chance that the patient will show up.