

Bank EDA

Megan Riley

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse
## v ggplot2 3.2.1      v purrr  0.3.2
## v tibble  2.1.3      v dplyr  0.8.3
## v tidyr   1.0.2      v stringr 1.4.0
## v ggplot2 3.2.1      v forcats 0.4.0

## -- Conflicts ----- tidyverse
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

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

```
## here() starts at /Users/zmartygirl/data/MSDSR/Stats6372Project/Stats-6372-Project-Two
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

##
## Attaching package: 'GGally'

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

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5    2019-07-22
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

```
## The following object is masked from 'package:purrr':
##
##      some

library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##      lift

root = here()

bank_20 = read.csv(paste(root, "/data/bank-additional/bank-additional-full.csv", sep = ""), sep=";")
bank_17 = read.csv(paste(root, "/data/bank/bank-full.csv", sep = ""), sep=";")
```

Summary

Unknown whether we should work with both data sets or if Dr. Turner is good with us choosing one. My vote is for bank_20 if we can choose.

Variable Notes: -Duration is a variable not known until Y is determined, duration is the duration of the call when attempting to sell the term deposit package.

- No NAs, uses unknown in places otherwise -Campaign is # of contacts, minimum 1 b/c it includes this contact in the data, even if the contact was unsuccessful -pdays needs to be potentially cleaned where 999 should equal NA or potentially switched to a categorical variable -Do not understand some of the later variables, seem to be more socially based.

#Dr Turner is heavily requesting summary stats

```
summary(bank_20)
```

```
##      age                job                marital
##  Min.   :17.00   admin.       :10422   divorced: 4612
##  1st Qu.:32.00   blue-collar: 9254   married  :24928
##  Median :38.00   technician : 6743   single   :11568
##  Mean   :40.02   services   : 3969   unknown  :   80
##  3rd Qu.:47.00   management : 2924
##  Max.   :98.00   retired    : 1720
##                (Other)    : 6156
##      education      default      housing
##  university.degree :12168   no       :32588   no       :18622
##  high.school        : 9515   unknown: 8597   unknown:  990
##  basic.9y           : 6045   yes      :    3   yes      :21576
##  professional.course: 5243
##  basic.4y           : 4176
##  basic.6y           : 2292
##  (Other)            : 1749
##      loan      contact      month      day_of_week
##  no       :33950   cellular :26144   may       :13769   fri:7827
##  unknown:  990   telephone:15044   jul       : 7174   mon:8514
```

```
## yes      : 6248          aug      : 6178    thu:8623
##                                     jun      : 5318    tue:8090
##                                     nov      : 4101    wed:8134
##                                     apr       : 2632
##                                     (Other): 2016
##      duration      campaign      pdays      previous
## Min.      : 0.0    Min.      : 1.000    Min.      : 0.0    Min.      :0.000
## 1st Qu.: 102.0    1st Qu.: 1.000    1st Qu.:999.0    1st Qu.:0.000
## Median : 180.0    Median : 2.000    Median :999.0    Median :0.000
## Mean      : 258.3    Mean      : 2.568    Mean      :962.5    Mean      :0.173
## 3rd Qu.: 319.0    3rd Qu.: 3.000    3rd Qu.:999.0    3rd Qu.:0.000
## Max.      :4918.0    Max.      :56.000    Max.      :999.0    Max.      :7.000
##
##      poutcome      emp.var.rate      cons.price.idx      cons.conf.idx
## failure      : 4252    Min.      :-3.40000    Min.      :92.20    Min.      :-50.8
## nonexistent:35563    1st Qu.: -1.80000    1st Qu.:93.08    1st Qu.: -42.7
## success      : 1373    Median : 1.10000    Median :93.75    Median : -41.8
##                                     Mean      : 0.08189    Mean      :93.58    Mean      : -40.5
##                                     3rd Qu.: 1.40000    3rd Qu.:93.99    3rd Qu.: -36.4
##                                     Max.      : 1.40000    Max.      :94.77    Max.      : -26.9
##
##      euribor3m      nr.employed      y
## Min.      :0.634    Min.      :4964    no :36548
## 1st Qu.:1.344    1st Qu.:5099    yes: 4640
## Median :4.857    Median :5191
## Mean      :3.621    Mean      :5167
## 3rd Qu.:4.961    3rd Qu.:5228
## Max.      :5.045    Max.      :5228
##
```

```
#Does not look like any NAs in either data set
sapply(bank_20, function(x) sum(is.na(x)))
```

```
##      age      job      marital      education      default
##      0      0      0      0      0
##      housing      loan      contact      month      day_of_week
##      0      0      0      0      0
##      duration      campaign      pdays      previous      poutcome
##      0      0      0      0      0
##      emp.var.rate      cons.price.idx      cons.conf.idx      euribor3m      nr.employed
##      0      0      0      0      0
##      y
##      0
```

```
sapply(bank_17, function(x) sum(is.na(x)))
```

```
##      age      job      marital      education      default      balance      housing
##      0      0      0      0      0      0      0
##      loan      contact      day      month      duration      campaign      pdays
##      0      0      0      0      0      0      0
##      previous      poutcome      y
##      0      0      0
```

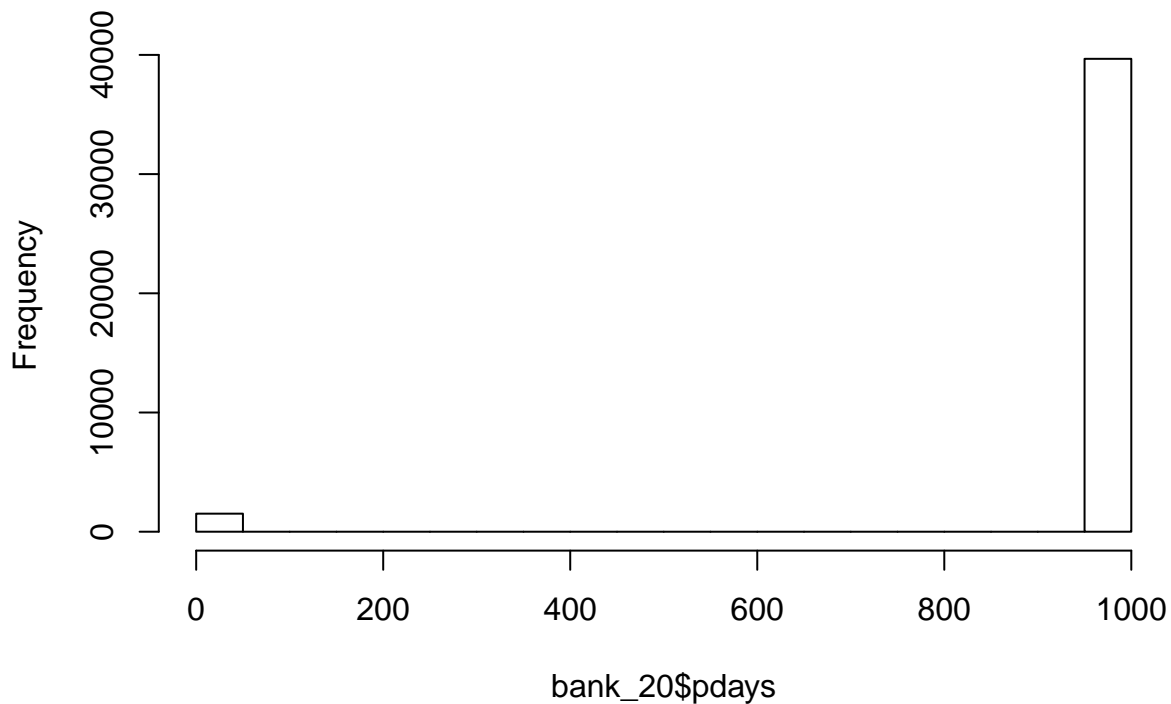
```
#Set any predictions without using Duration or Y
```

Data Preprocessing

Pdays is a major variable that needs correcting, the large inclusion of the 999 category requires either splitting the variable or turning it into a categorical with bins. The first option will be used to create one boolean variable that indicates whether a customer has been contacted before (where the 999 code indicates they have not been) and another variable with

```
clean_bank_20 = as.data.frame(bank_20)
#pdays- about 40k of the 41k are at level 999, no previous contact
#could bin this data
hist(bank_20$pdays)
```

Histogram of bank_20\$pdays

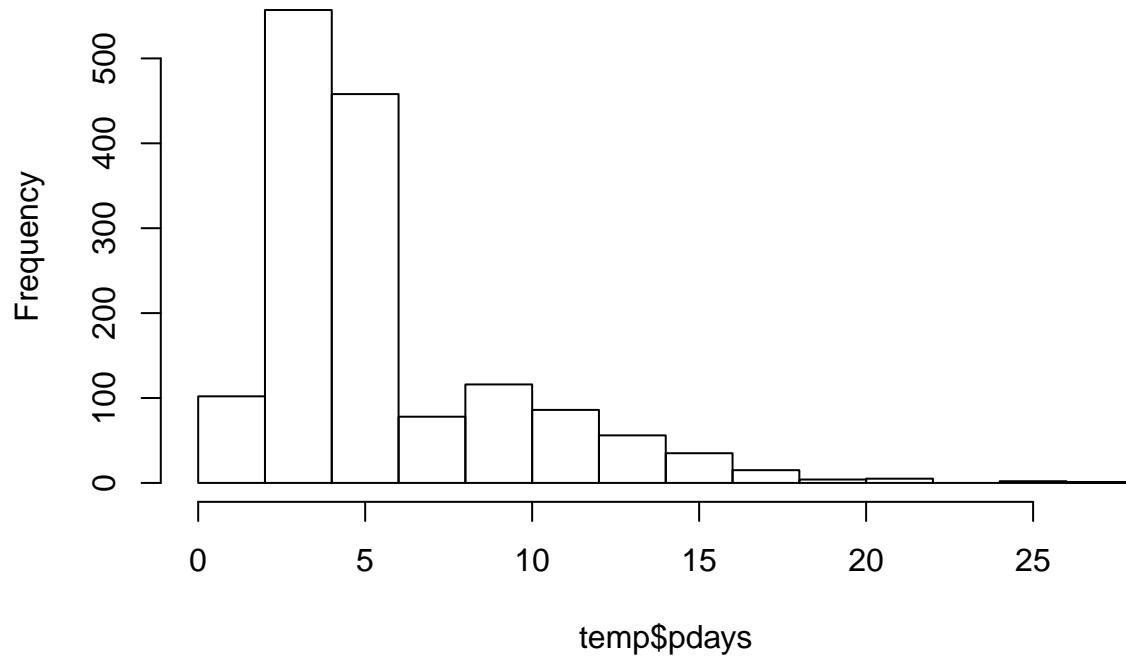


```
temp = bank_20 %>% filter(pdays != 999)
dim(temp)
```

```
## [1] 1515 21
```

```
hist(temp$pdays)
```

Histogram of temp\$pdays



```
summary(temp$pdays)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   3.000   6.000   6.015   7.000  27.000
```

```
#within 5 days, 10 , 15, 30 and never
```

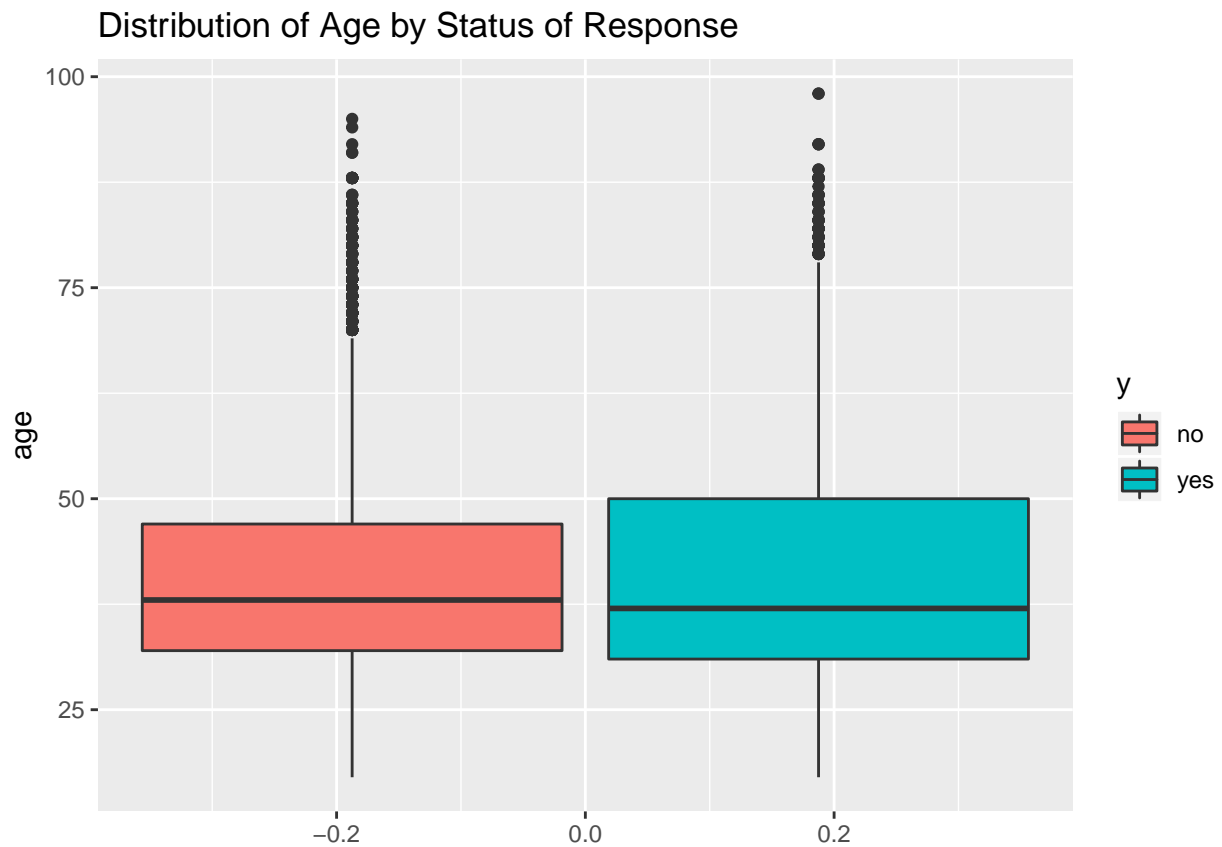
```
clean_bank_20$newpdays = case_when(bank_20$pdays == 999 ~ "Never",
                                   bank_20$pdays >= 15 ~ "Within 30 Days",
                                   bank_20$pdays >= 10 & bank_20$pdays < 15 ~ "Within 15 Days",
                                   bank_20$pdays >= 5 & bank_20$pdays < 10 ~ "Within 10 Days",
                                   bank_20$pdays < 5 ~ "Within 5 Days")
```

```
clean_bank_20 = dplyr::select(clean_bank_20, -pdays)
```

EDA Plots of Specific Variables by Y Response

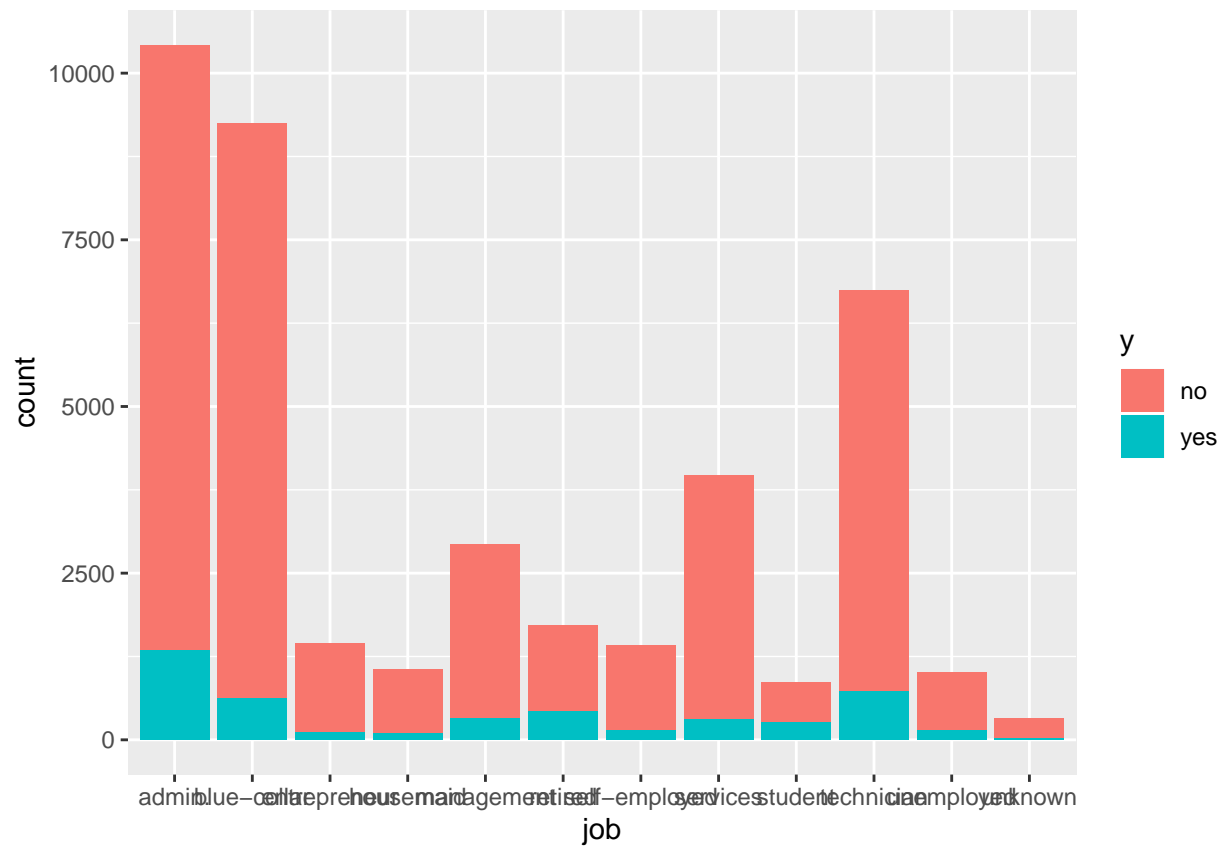
```
#Age
```

```
clean_bank_20 %>% ggplot(aes(y= age, fill = y)) + geom_boxplot() + ggtitle("Distribution of Age by Status")
```

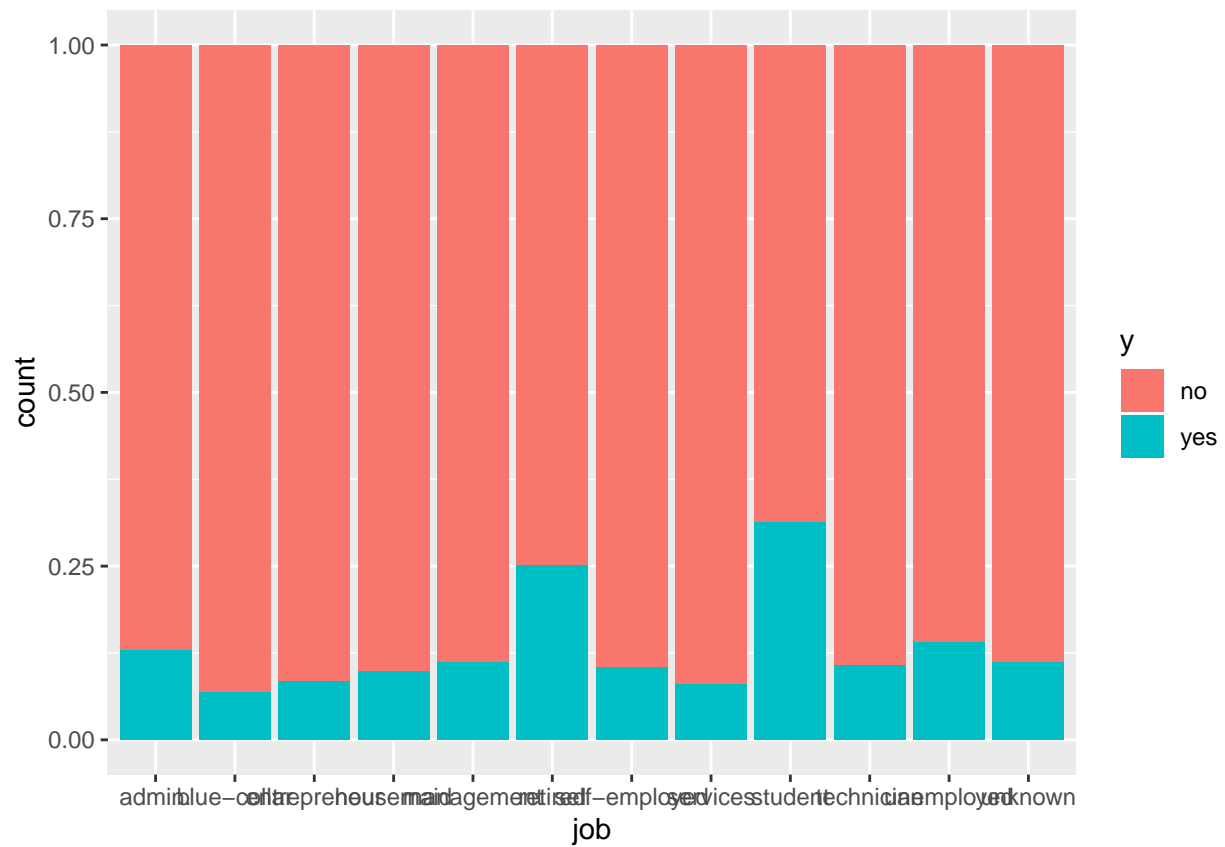


```
#Job  
clean_bank_20 %>% ggplot(aes(x = job, fill = y)) + geom_histogram(stat = "count")
```

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

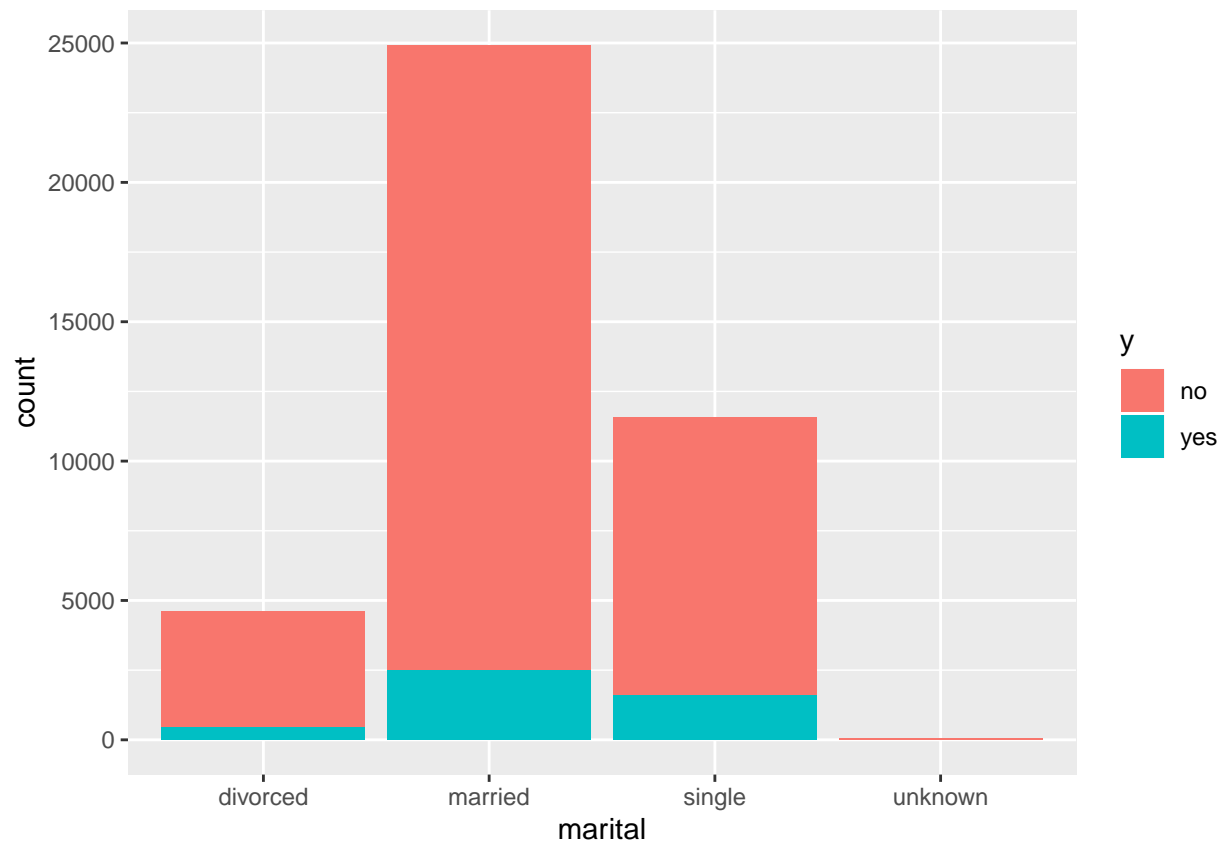


```
clean_bank_20 %>% ggplot(aes(x = job, fill = y)) + geom_bar(position = "fill")
```

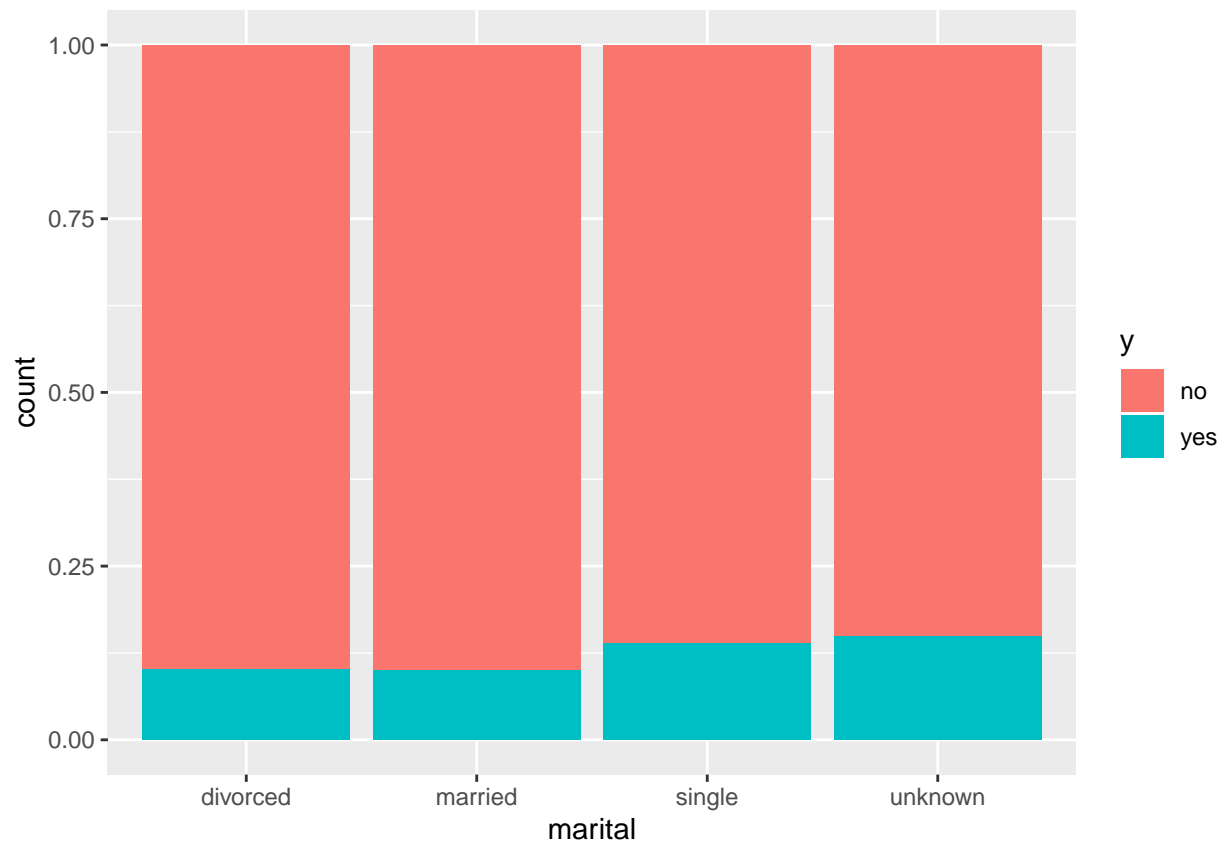


```
#Marital
clean_bank_20 %>% ggplot(aes(x = marital, fill = y)) + geom_histogram(stat = "count")
```

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

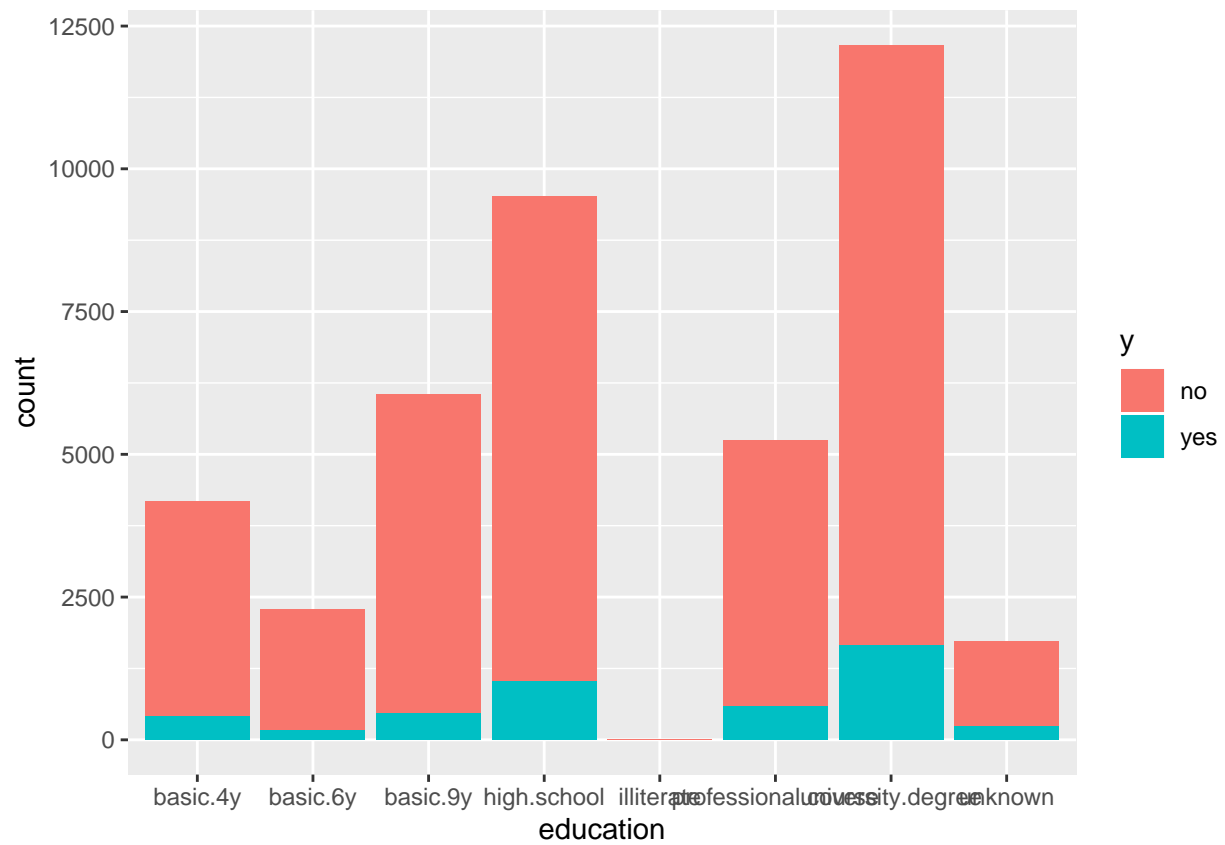



```
clean_bank_20 %>% ggplot(aes(x = marital, fill = y)) + geom_bar(position = "fill")
```

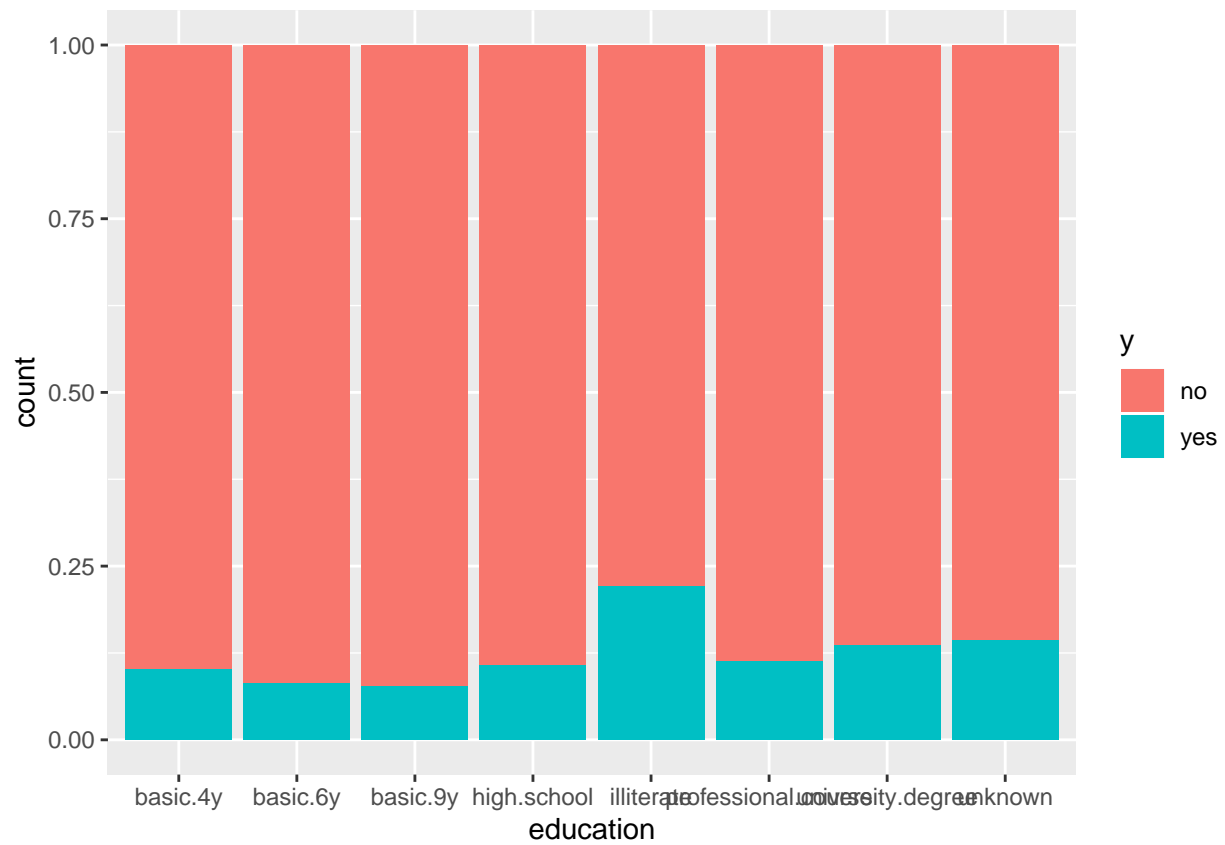


```
#Education  
clean_bank_20 %>% ggplot(aes(x = education, fill = y)) + geom_histogram(stat = "count")
```

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

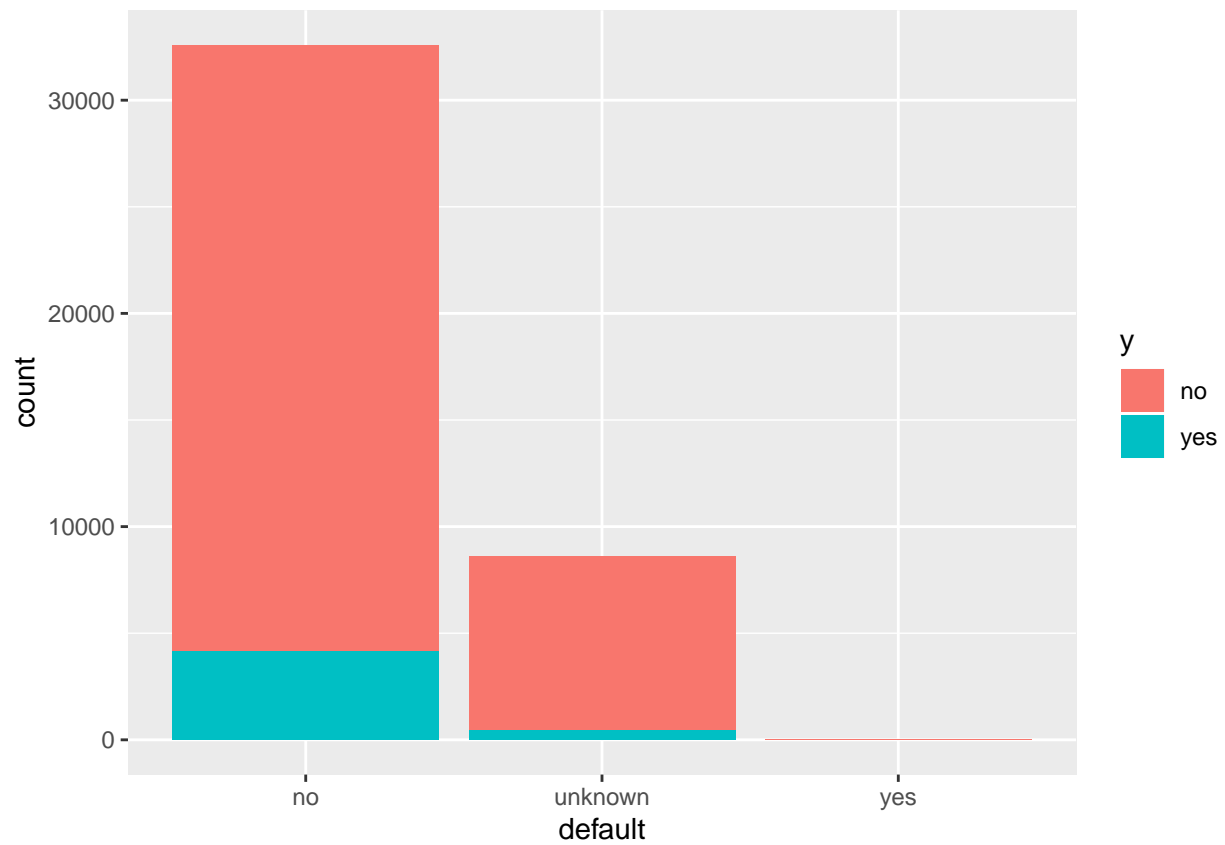


```
clean_bank_20 %>% ggplot(aes(x = education, fill = y)) + geom_bar(position = "fill")
```

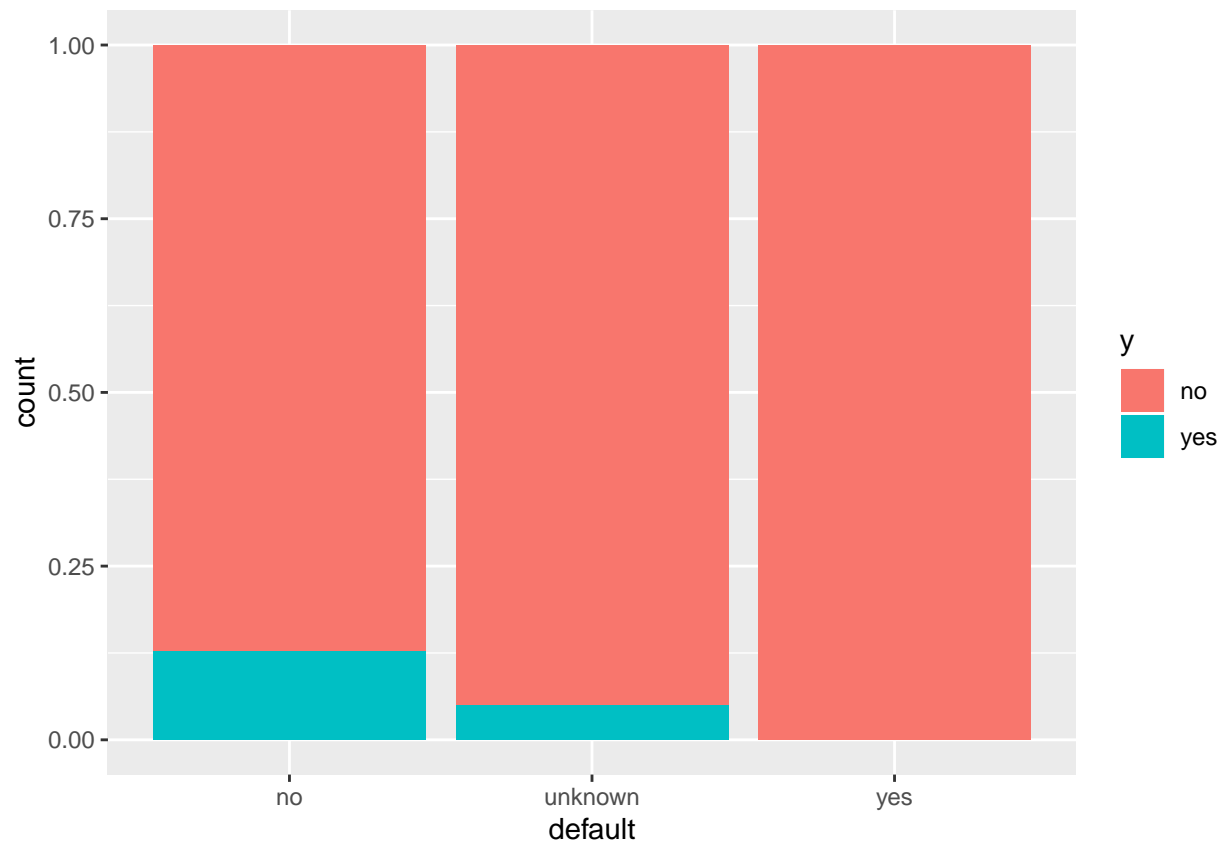


```
#Default
clean_bank_20 %>% ggplot(aes(x = default, fill = y)) + geom_histogram(stat = "count")
```

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

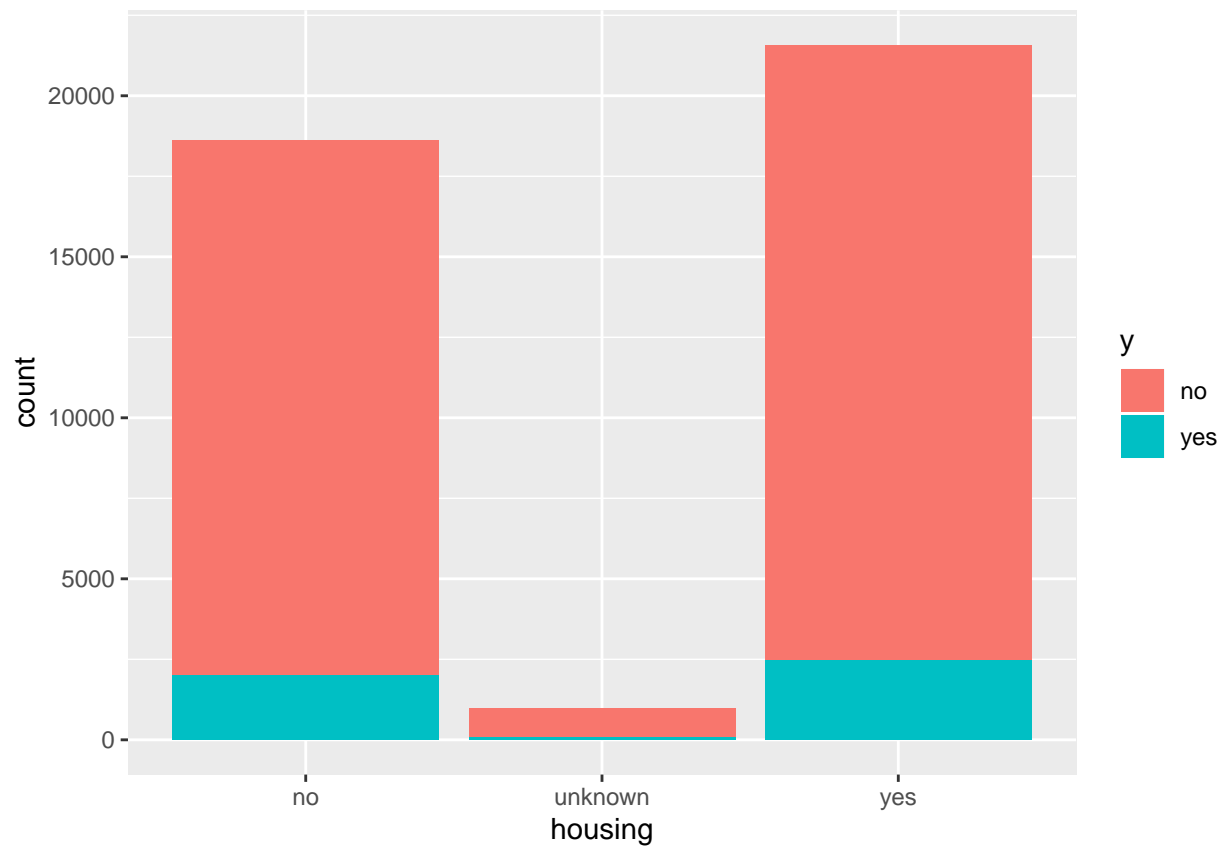


```
clean_bank_20 %>% ggplot(aes(x = default, fill = y)) + geom_bar(position = "fill")
```

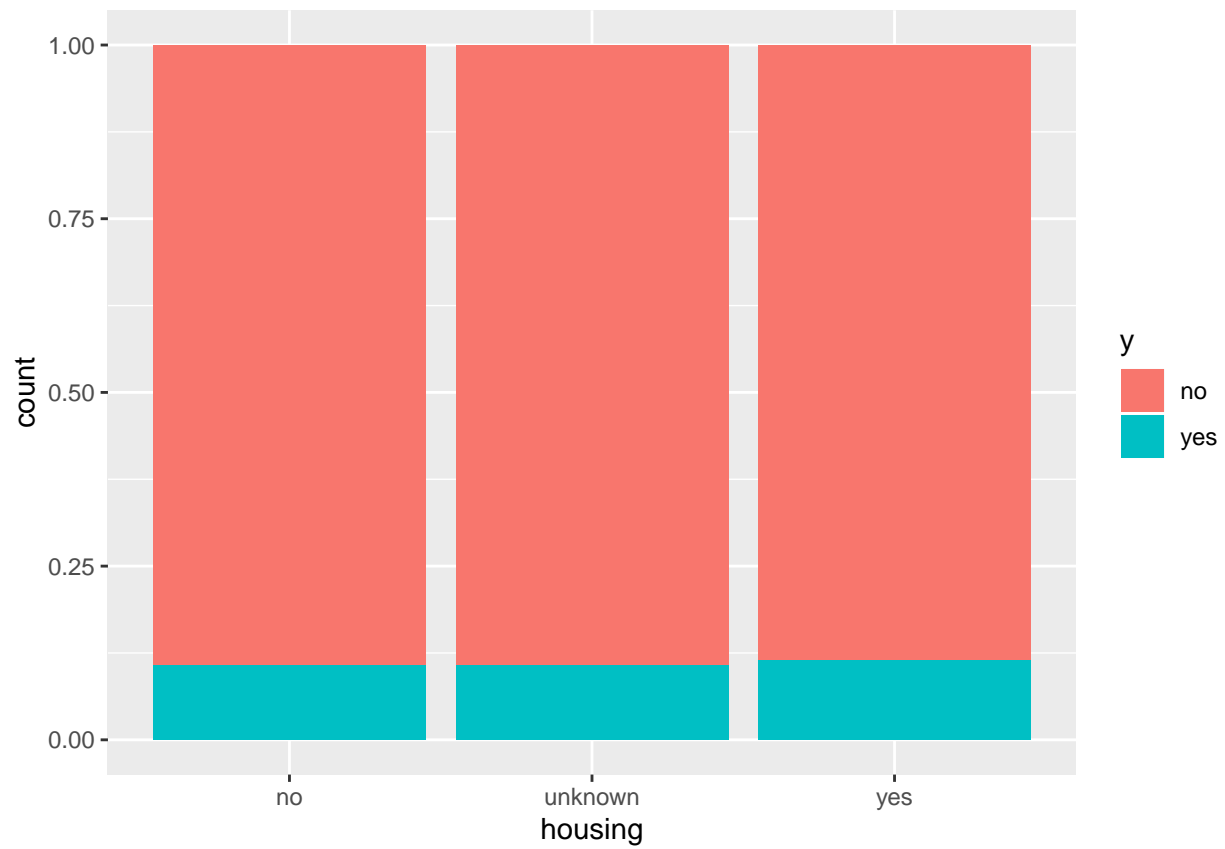


```
#Housing  
clean_bank_20 %>% ggplot(aes(x = housing, fill = y)) + geom_histogram(stat = "count")
```

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



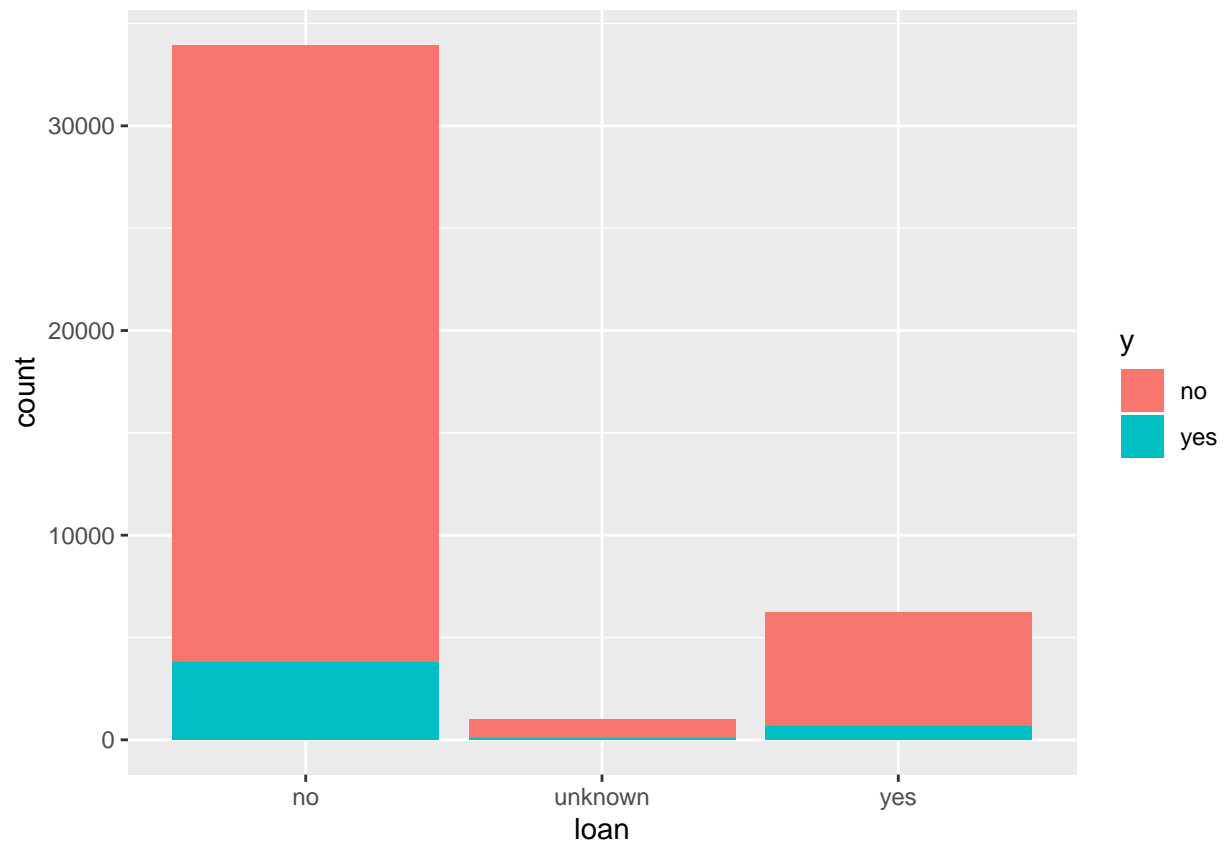
```
clean_bank_20 %>% ggplot(aes(x = housing, fill = y)) + geom_bar(position = "fill")
```



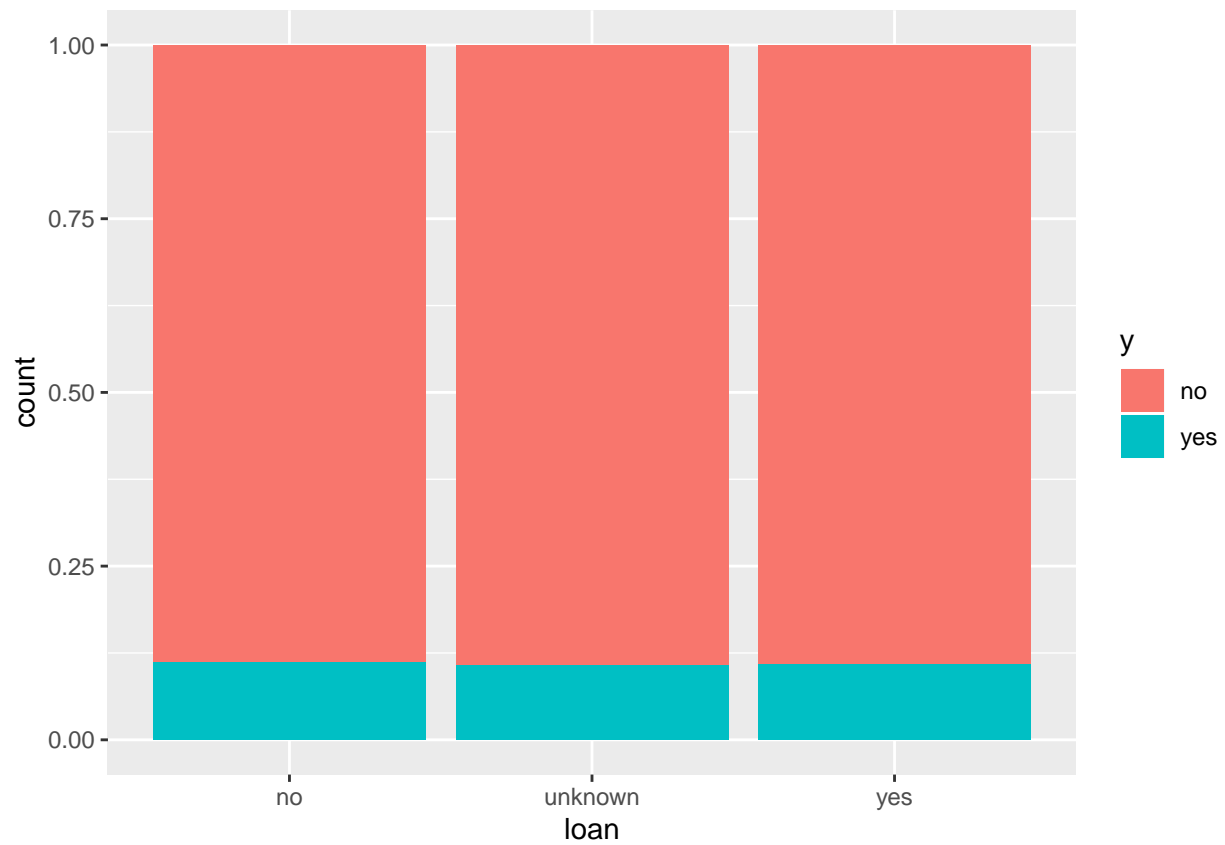
```
#Loan
```

```
clean_bank_20 %>% ggplot(aes(x = loan, fill = y)) + geom_histogram(stat = "count")
```

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

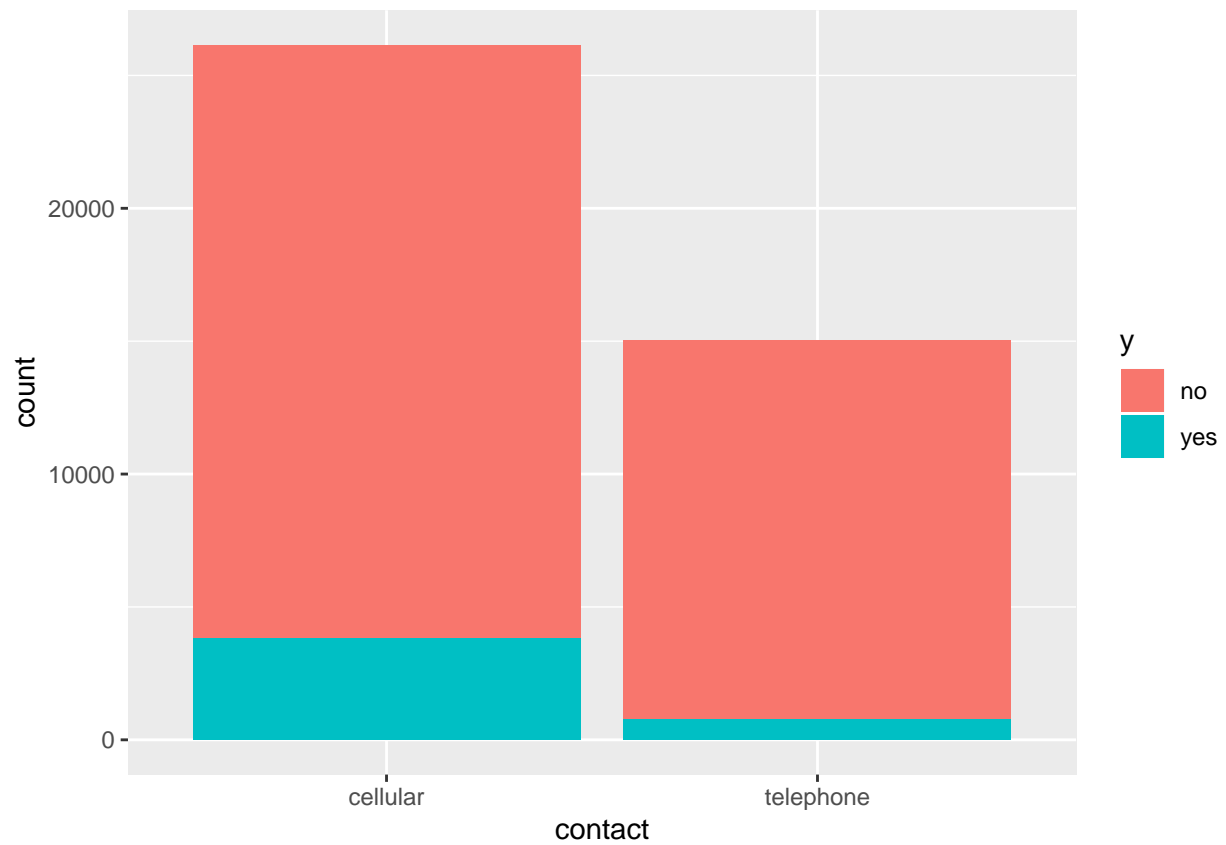



```
clean_bank_20 %>% ggplot(aes(x = loan, fill = y)) + geom_bar(position = "fill")
```

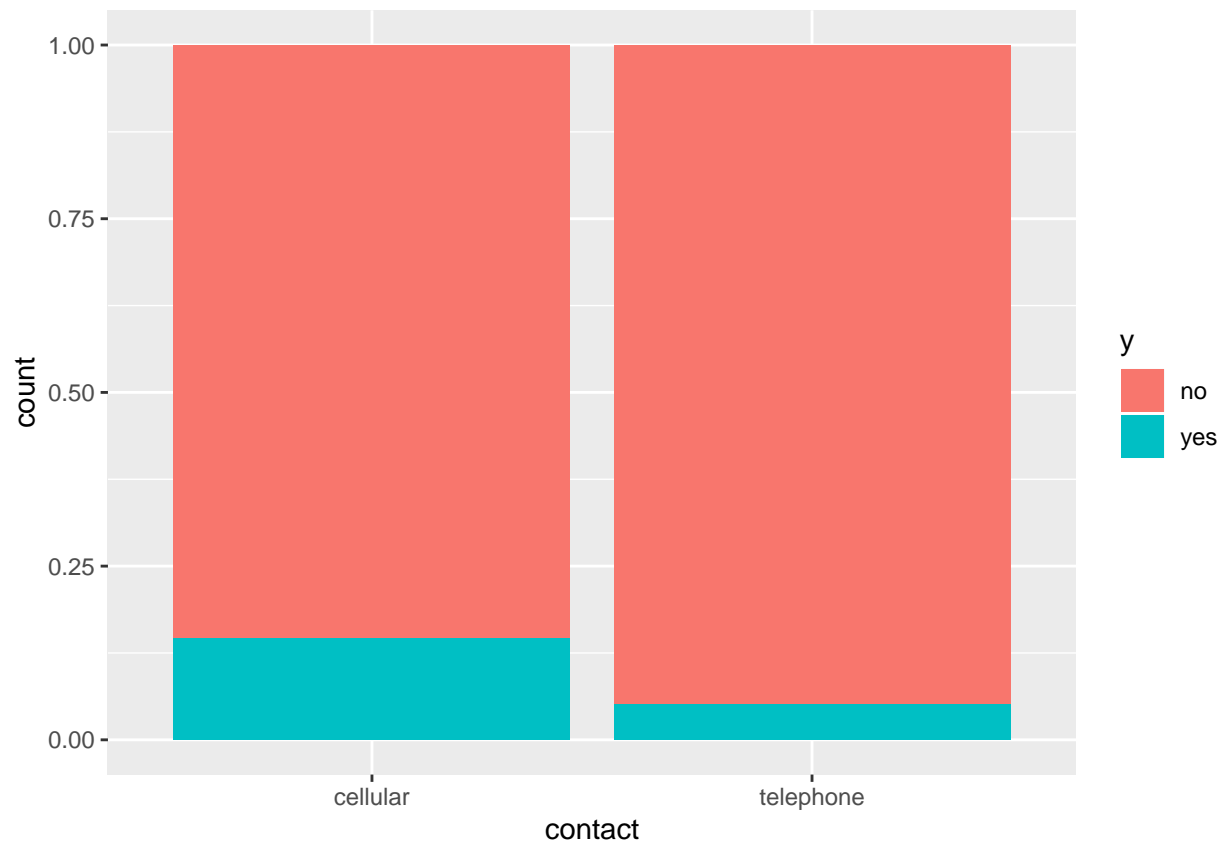


```
#Contact  
clean_bank_20 %>% ggplot(aes(x = contact, fill = y)) + geom_histogram(stat = "count")
```

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

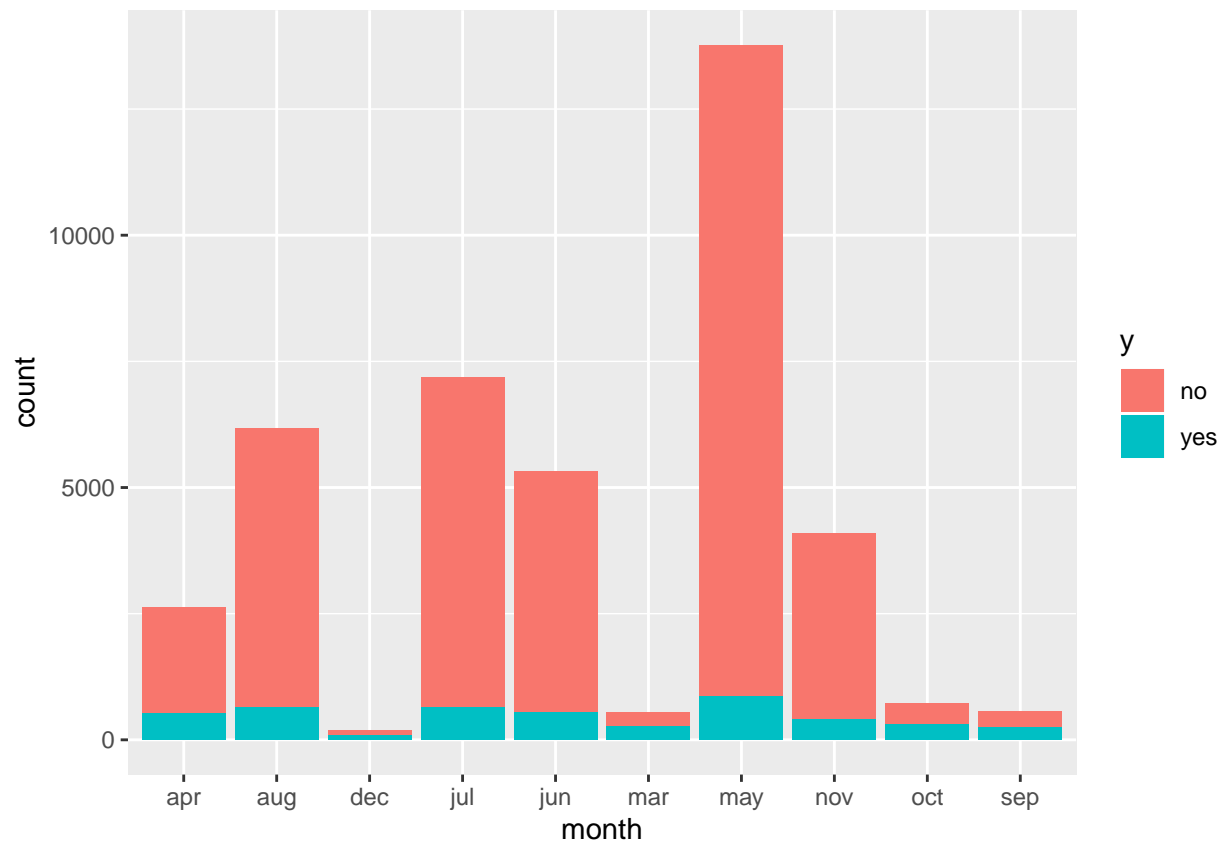


```
clean_bank_20 %>% ggplot(aes(x = contact, fill = y)) + geom_bar(position = "fill")
```

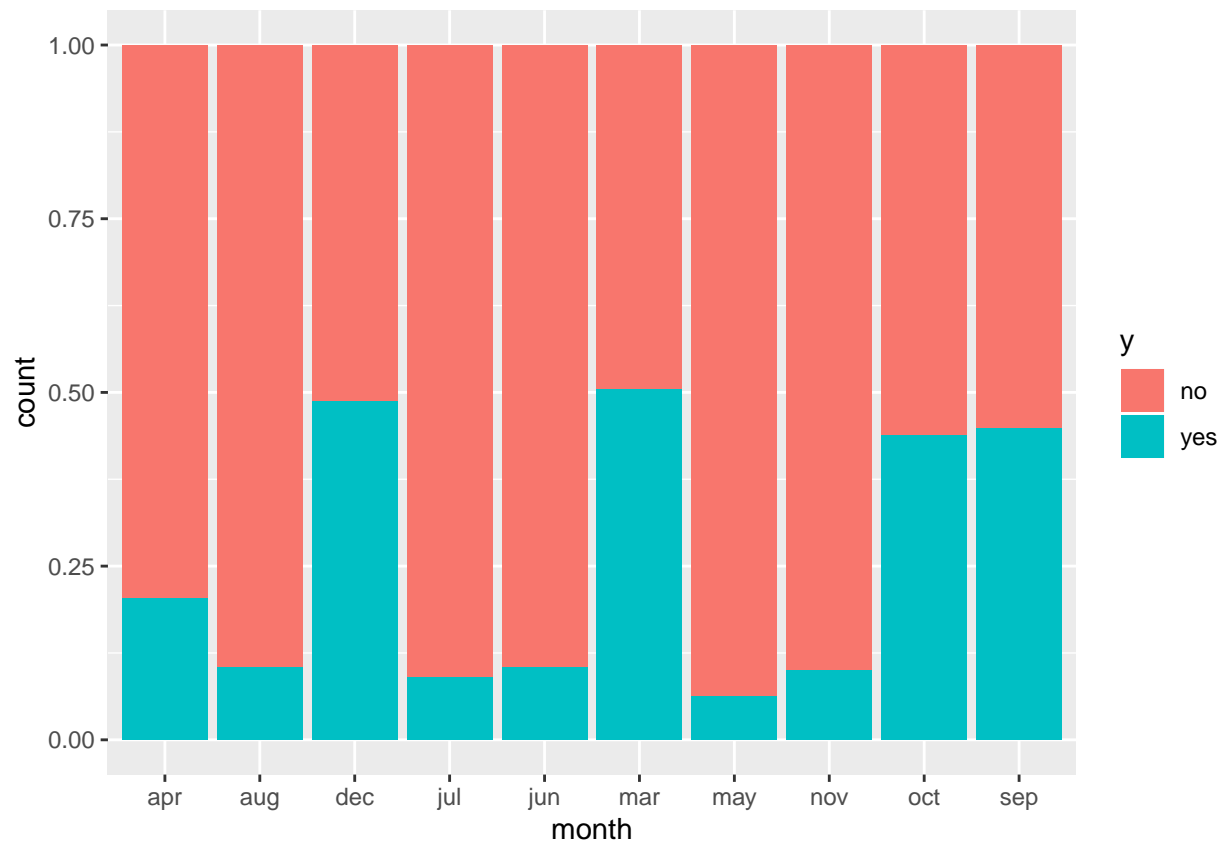


```
#Month
clean_bank_20 %>% ggplot(aes(x = month, fill = y)) + geom_histogram(stat = "count")
```

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

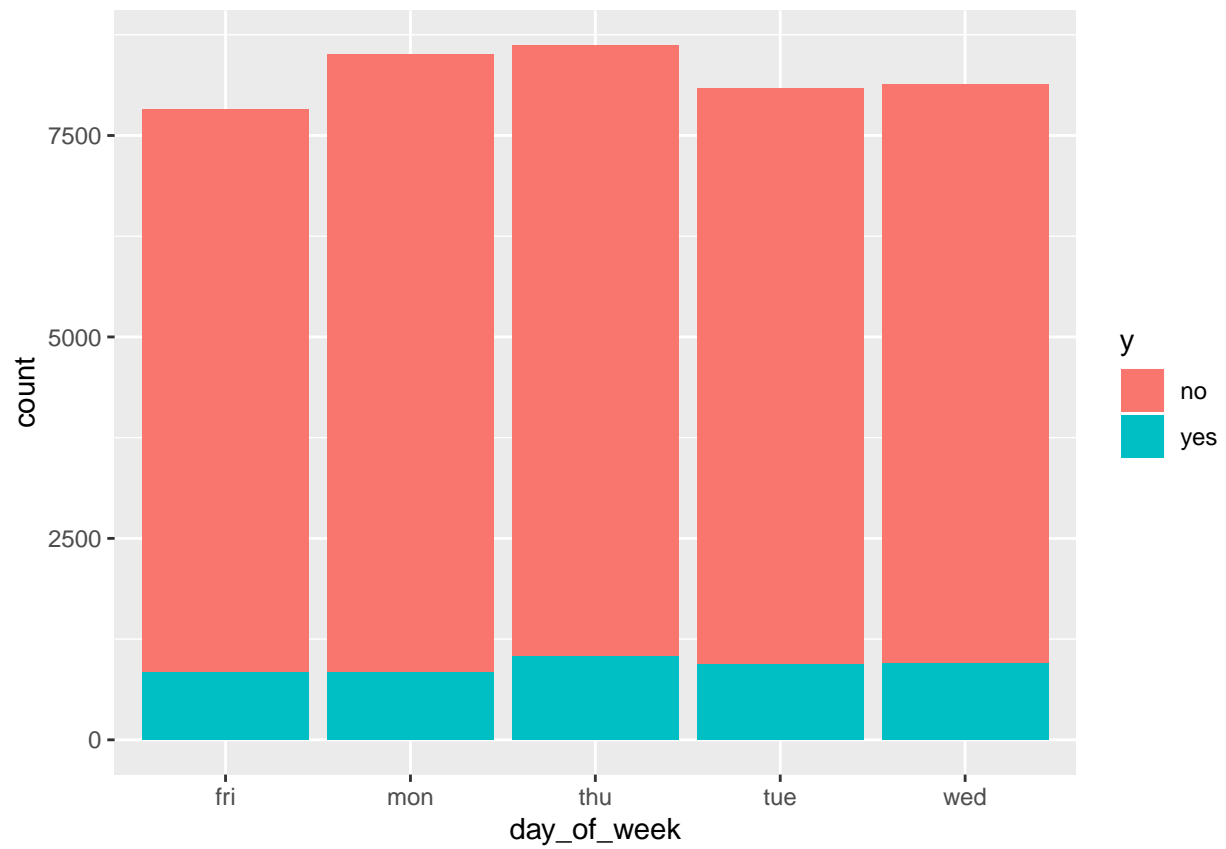


```
clean_bank_20 %>% ggplot(aes(x = month, fill = y)) + geom_bar(position = "fill")
```

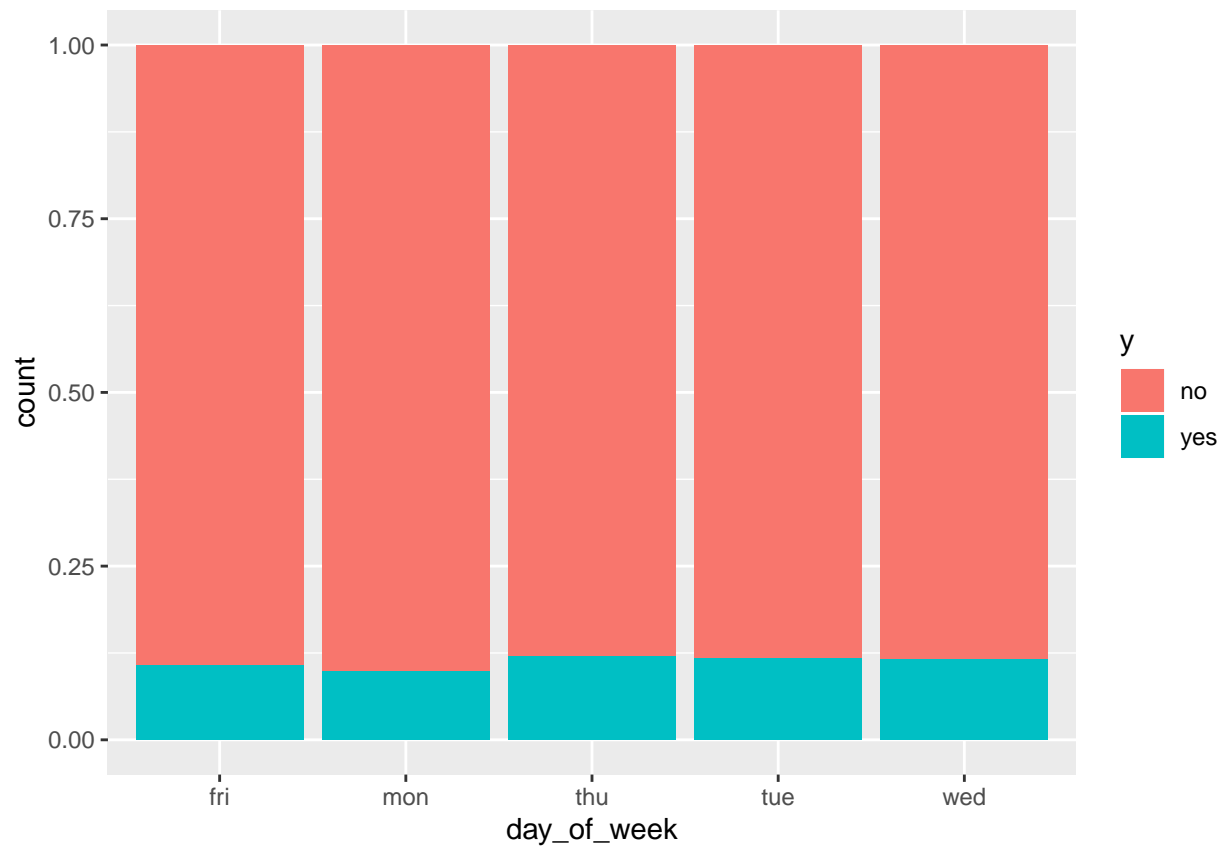


```
#Day_of_week
clean_bank_20 %>% ggplot(aes(x = day_of_week, fill = y)) + geom_histogram(stat = "count")
```

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

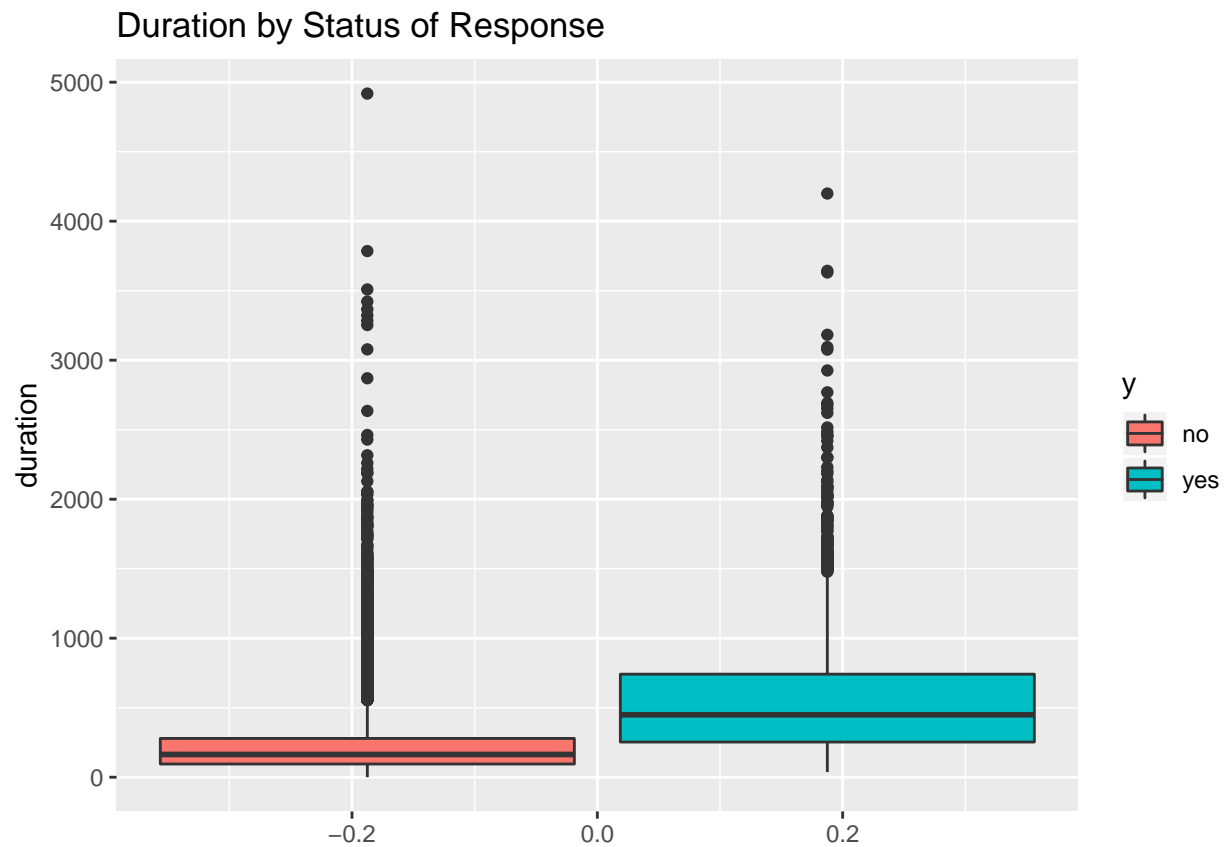


```
clean_bank_20 %>% ggplot(aes(x = day_of_week, fill = y)) + geom_bar(position = "fill")
```

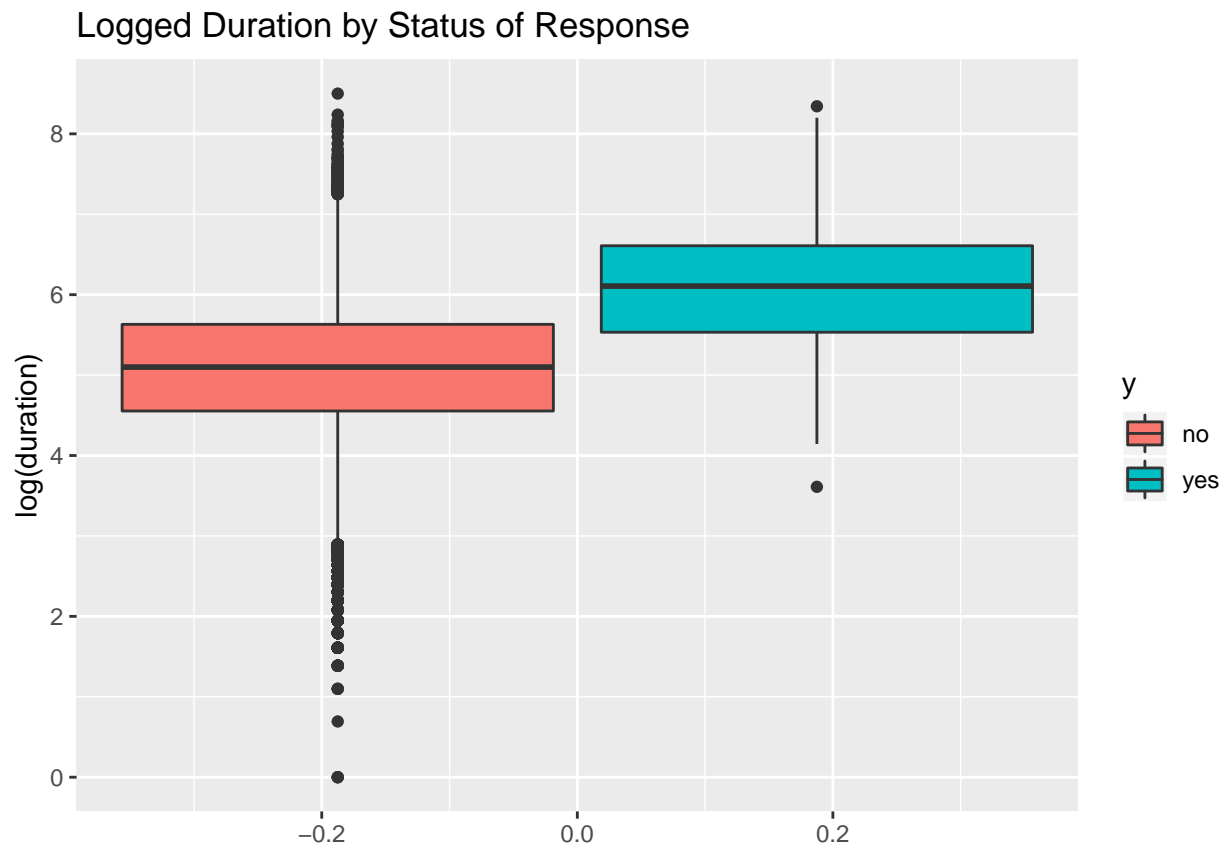


```
#duration
```

```
clean_bank_20 %>% ggplot(aes(y = duration, fill = y)) + geom_boxplot() + ggtitle("Duration by Status of
```

```
clean_bank_20 %>% ggplot(aes(y = log(duration), fill = y)) + geom_boxplot() + ggtitle("Logged Duration by Status of Response")
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
```

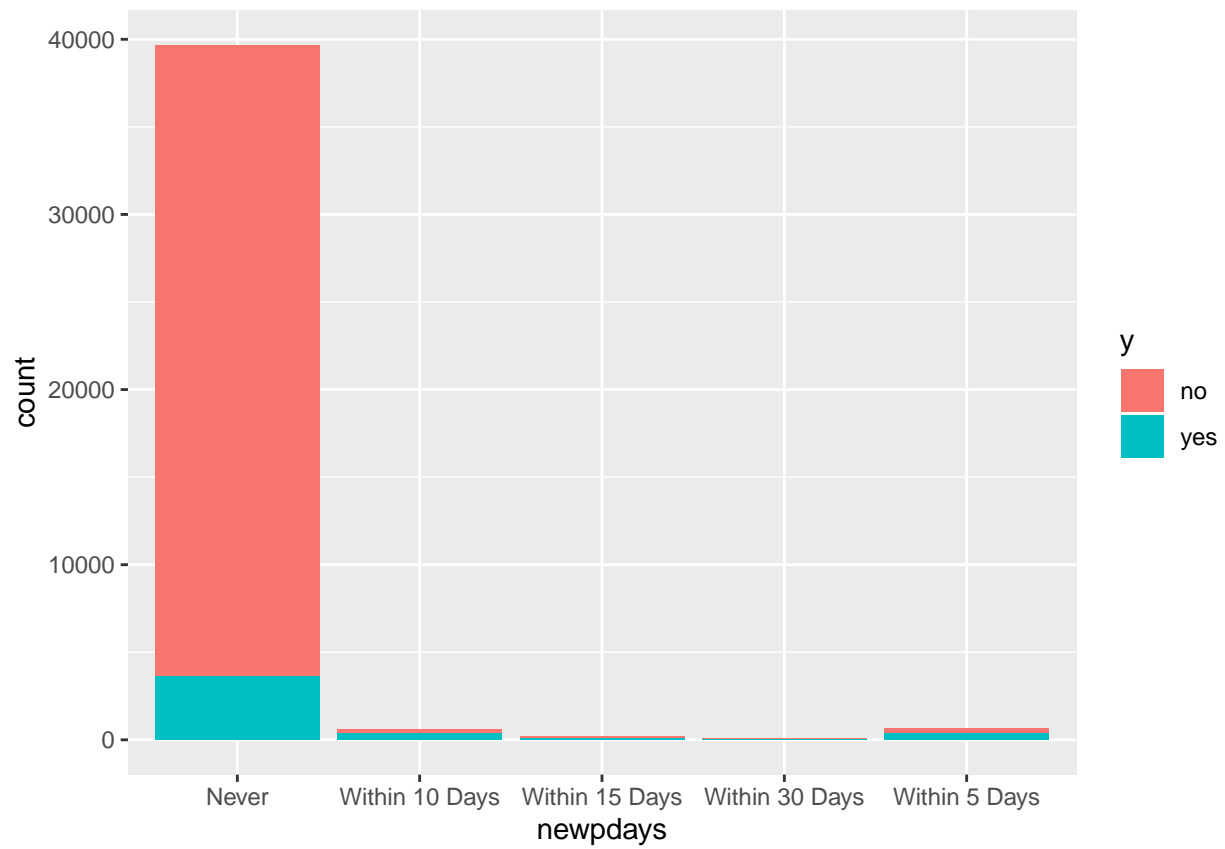


```
#Campaign

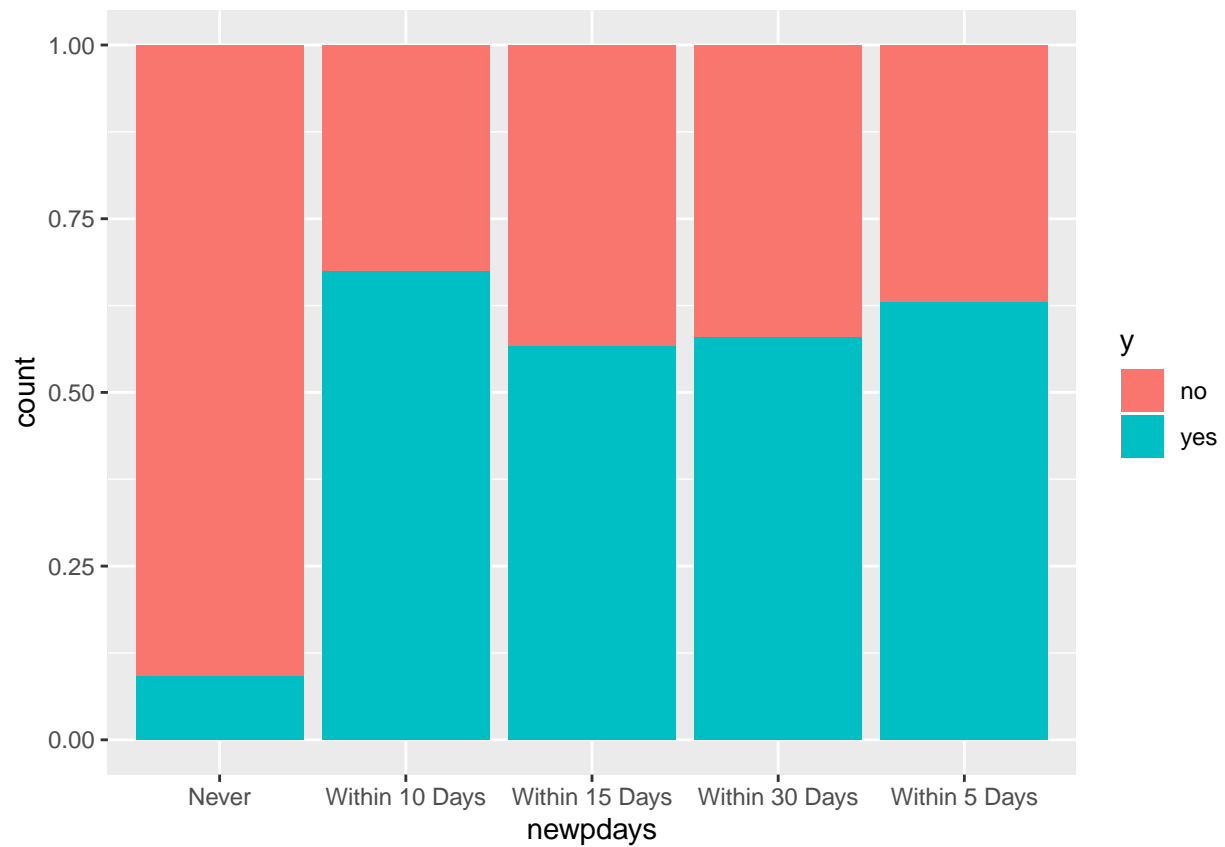
#newPdays

clean_bank_20 %>% ggplot(aes(x =newpdays, fill = y)) + geom_histogram(stat = "count")

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



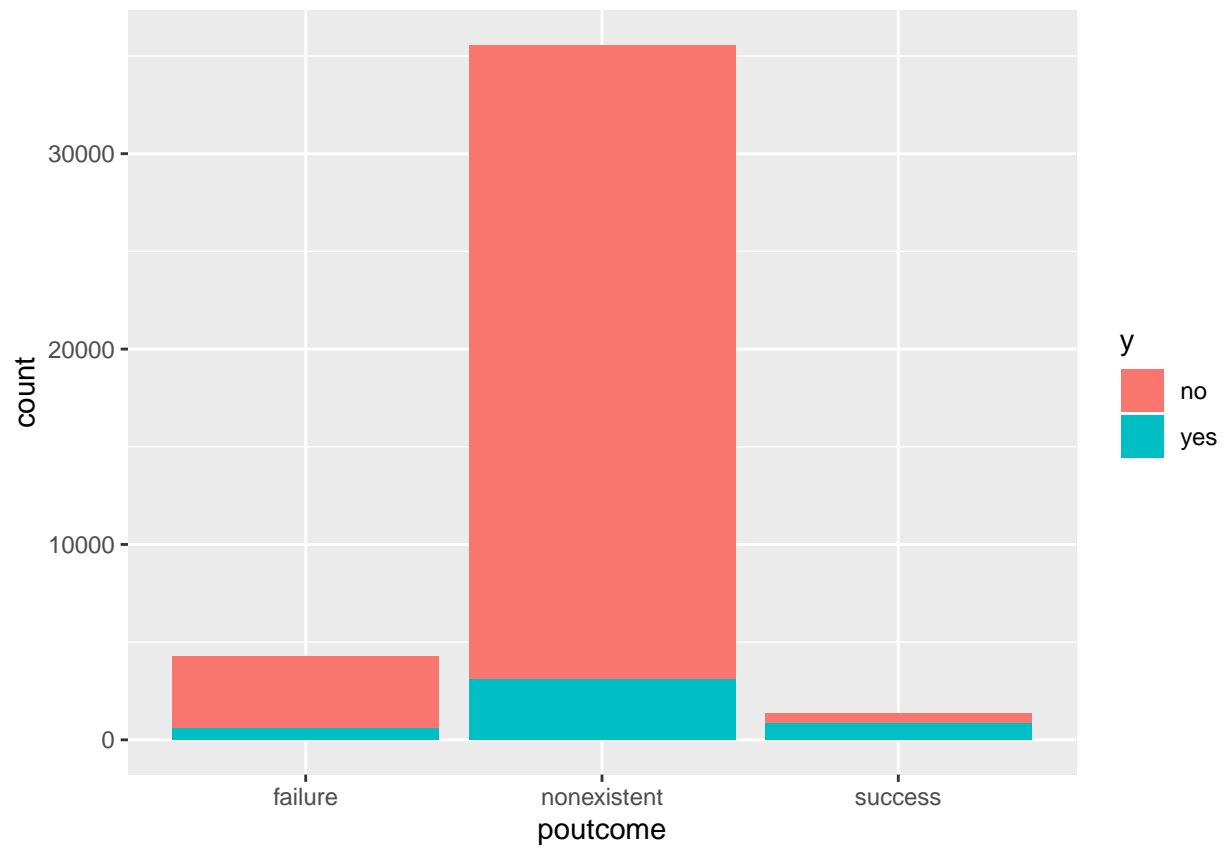
```
clean_bank_20 %>% ggplot(aes(x = newpdays, fill = y)) + geom_bar(position = "fill")
```



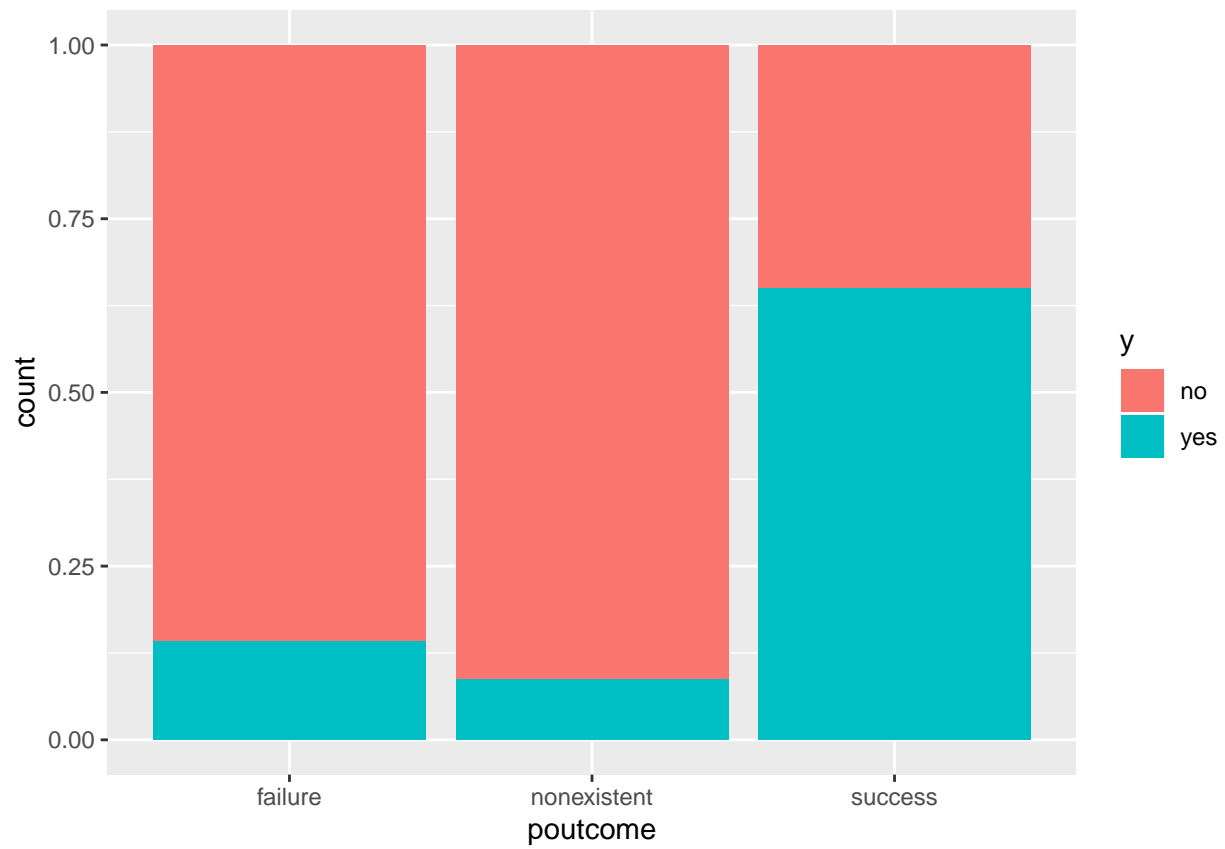
#Previous

```
#poutcome
clean_bank_20 %>% ggplot(aes(x =poutcome, fill = y)) + geom_histogram(stat = "count")
```

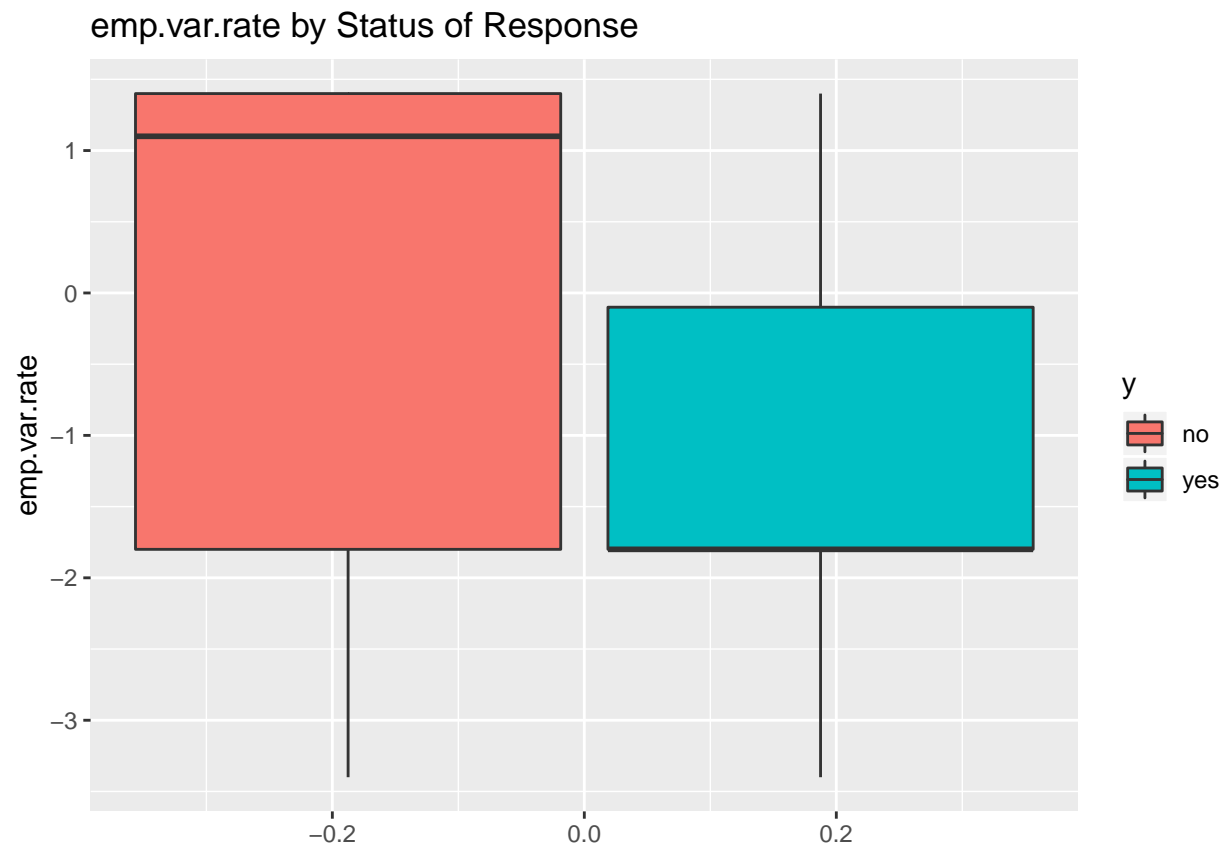
Warning: Ignoring unknown parameters: binwidth, bins, pad



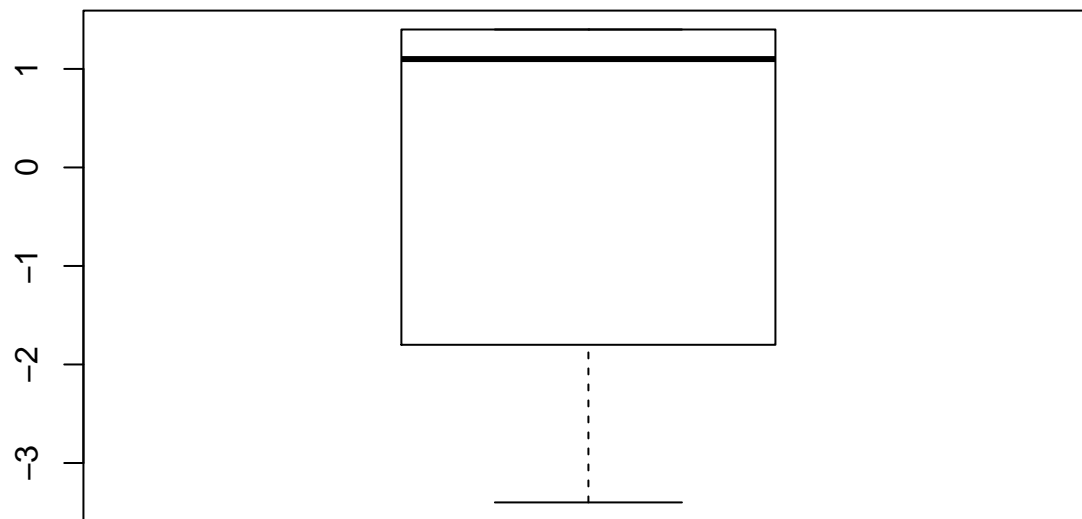
```
clean_bank_20 %>% ggplot(aes(x = poutcome, fill = y)) + geom_bar(position = "fill")
```



```
#emp.var.rate
clean_bank_20 %>% ggplot(aes(y = emp.var.rate, fill = y)) + geom_boxplot() + ggtitle("emp.var.rate by S
```

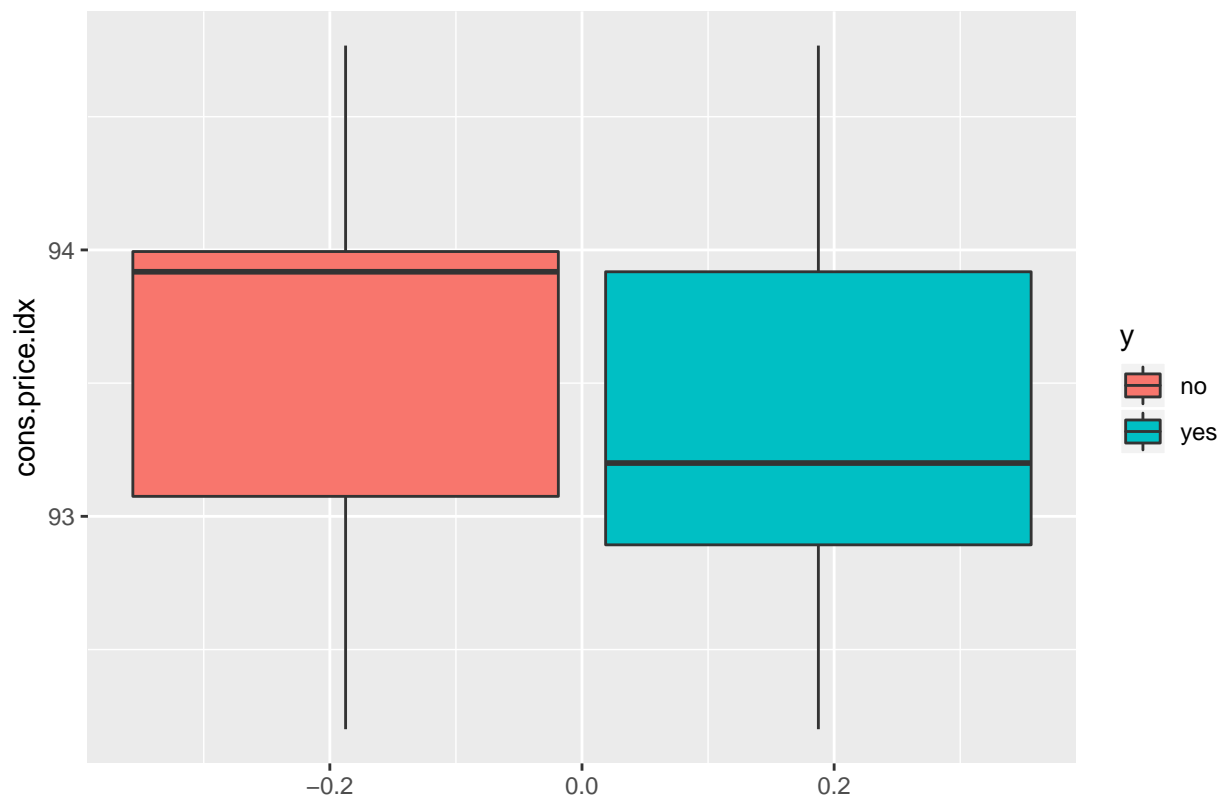


```
boxplot(clean_bank_20$emp.var.rate)
```



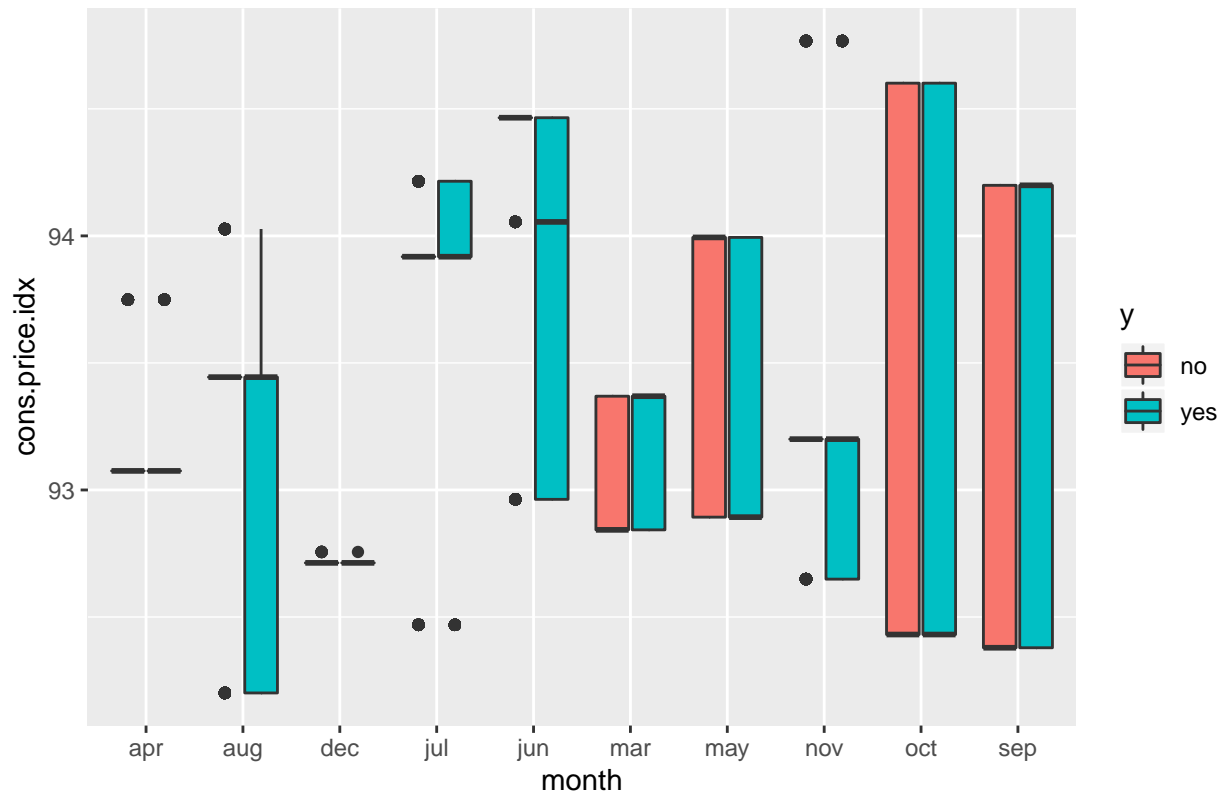
```
#cons.price.idx
clean_bank_20 %>% ggplot(aes(y = cons.price.idx, fill = y)) + geom_boxplot() + ggtitle("CPI by Status of Response")
```

CPI by Status of Response



```
#something here potentially, but is messy
clean_bank_20 %>% ggplot(aes(x = month, y = cons.price.idx, fill = y)) + geom_boxplot() + ggtitle("CPI by Status of Response")
```

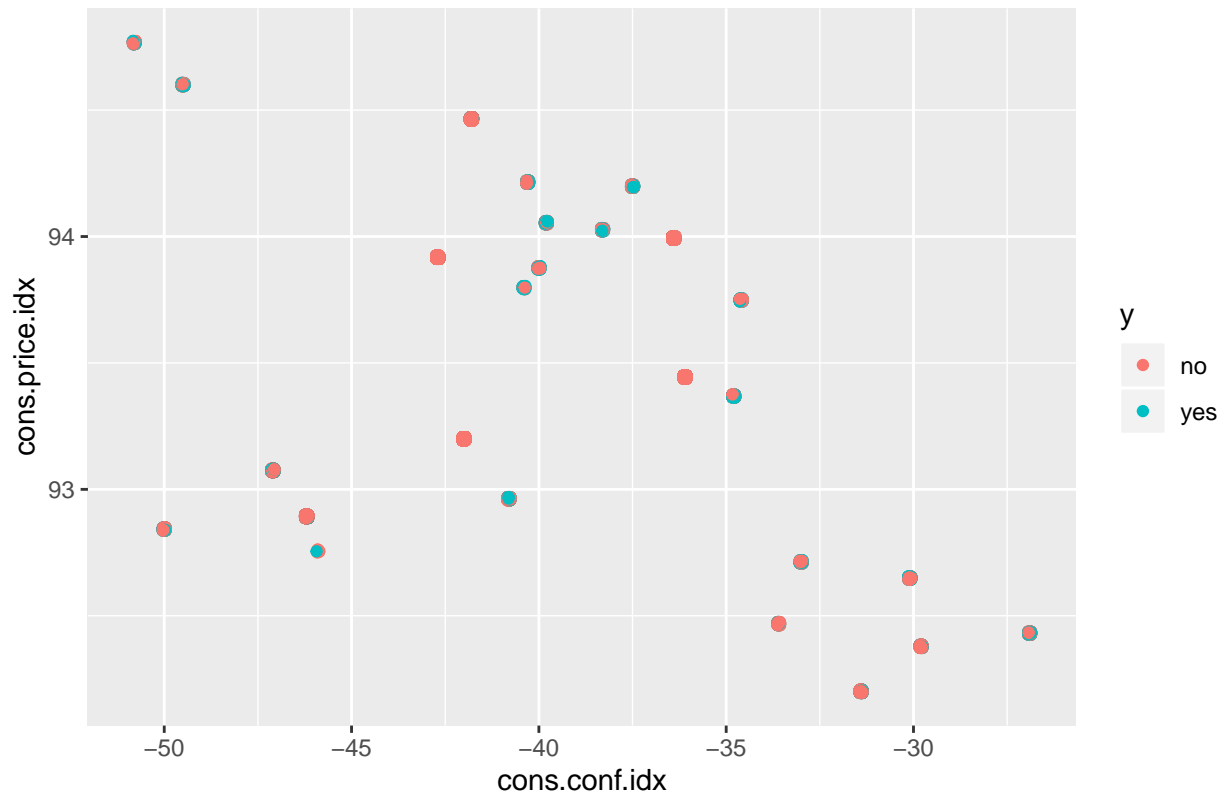

CPI by month by Status of Response



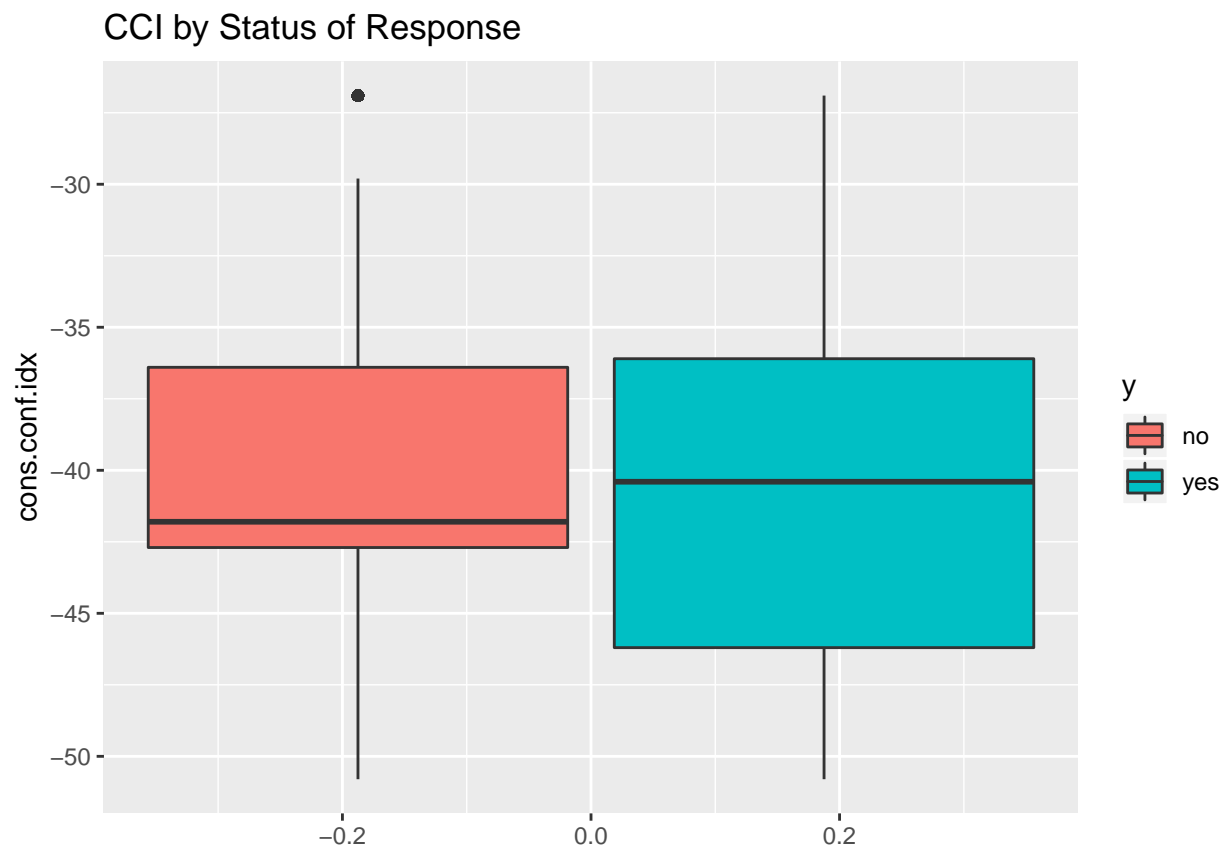
#CCI and CPI

```
clean_bank_20 %>% ggplot(aes(x = cons.conf.idx, y = cons.price.idx, color = y)) + geom_jitter() + ggtitle("CPI by month by Status of Response")
```

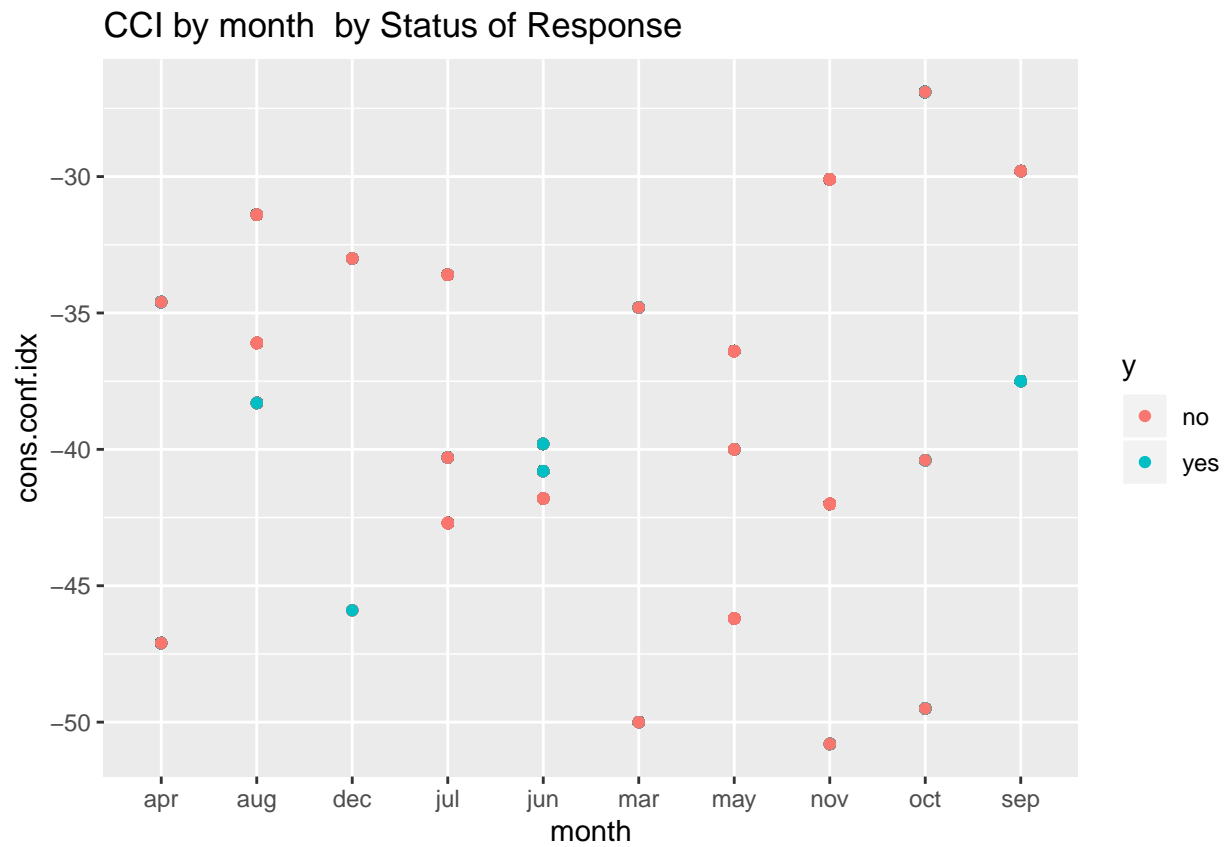
CPI by month by Status of Response



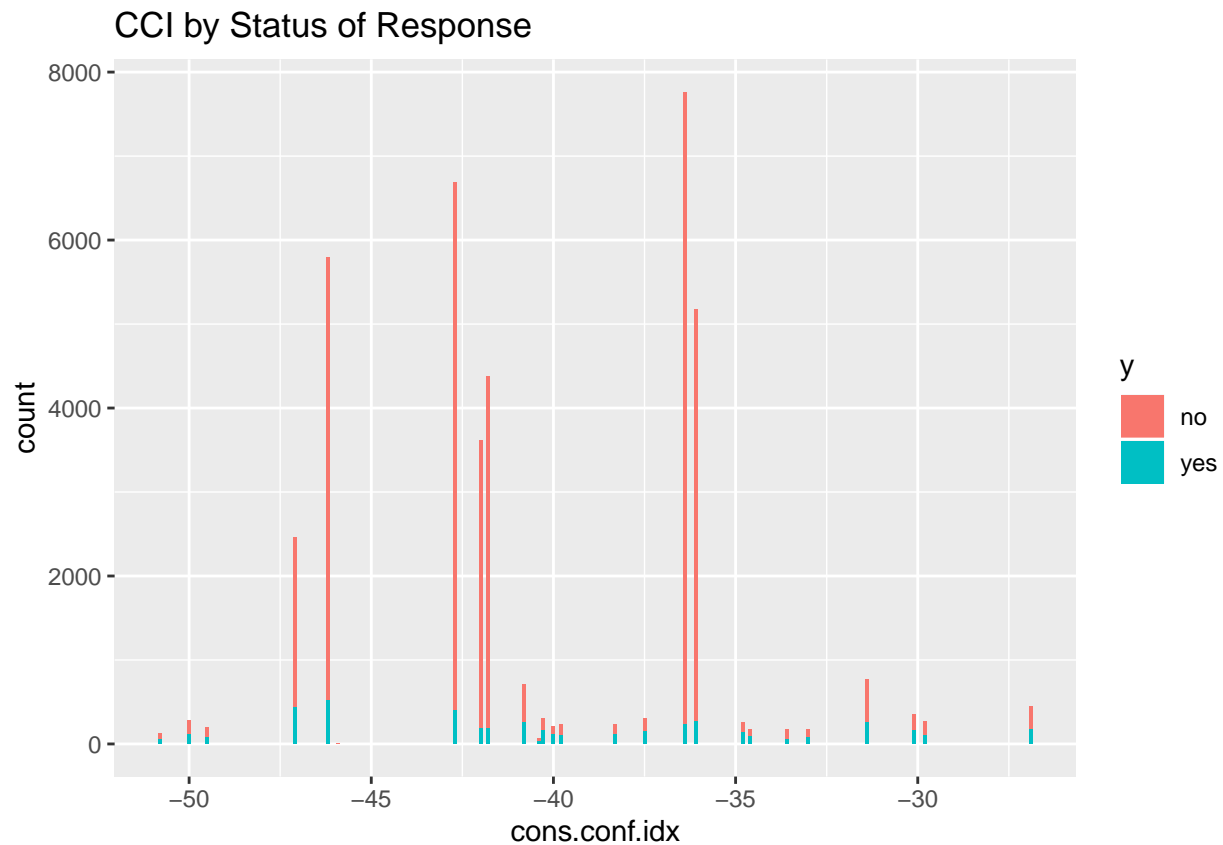
```
#cons.conf.idx
clean_bank_20 %>% ggplot(aes(y = cons.conf.idx, fill = y)) + geom_boxplot() + ggtitle("CCI by Status of
```



```
clean_bank_20 %>% ggplot(aes(x = month, y = cons.conf.idx, color = y)) + geom_point() + ggtitle("CCI by
```



```
clean_bank_20 %>% ggplot(aes(x = cons.conf.idx, fill = y)) + geom_histogram(stat = "count") + ggtitle("CCI by month by Status of Response")
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



```
#euribor3m

#nr.employed

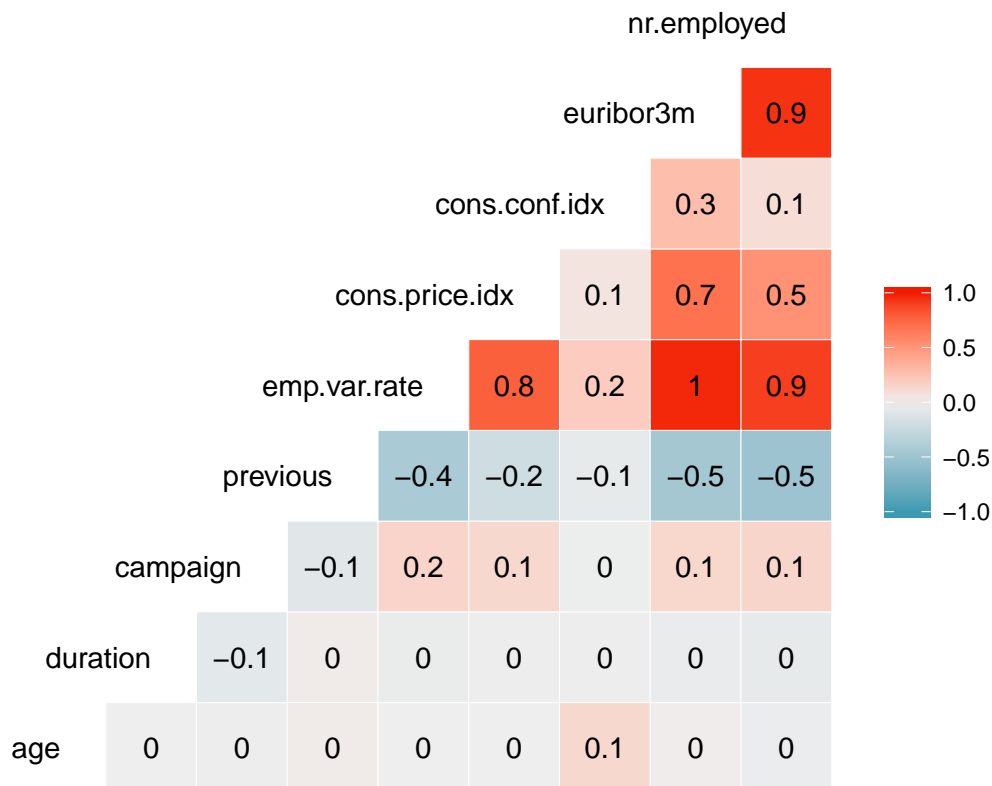
#working with the ftables

#probs needs to be a factor
#ftable(addmargins(clean_bank_20$y, clean_bank_20$month))
```

Multicollinearity and Interactions

We need to check our multicollinearity between predictor variables as well as interactions, for example, month information of a cyclical nature may be reflected in some of the economic indicators or dependent on them.

```
#Multicollinearity in the continuous variables.
cont_bank = clean_bank_20[,c(1,11,12,13,15,16,17,18,19)]
ggcorr(cont_bank, label = TRUE, hjust = 1 )
```



Predicting Duration, in order to predict calls.

Basically if we know there is a strong relationship between duration and the result we are predicting, we can use information that explains duration to therefore explain response. Since it is invalid to use duration in my opinion.

- Is this valid? Possibly?
- Is it helpful? Maybe?

```
duration_model = lm(duration ~ ., data = bank_20)
```

Uneven Split in outcomes

Yes happens about 10% of the time, where no is the response 90% of the time. This unbalance makes it difficult to predict. - Can balance the train/test split, what else have we learned about predicting unbalanced outcomes?

```
yes_answer = bank_20 %>% filter(y == "yes")

no_answer_all = bank_20 %>% filter(y == "no")
no_indices = sample(dim(no_answer_all)[1], dim(yes_answer)[1])
no_answer = no_answer_all[no_indices,]

balanced_bank_20 = rbind(yes_answer, no_answer)

balanced_indices = sample(dim(balanced_bank_20)[1], round(dim(balanced_bank_20)[1] * .1))
balanced_test = balanced_bank_20[balanced_indices,]
balanced_train = balanced_bank_20[-balanced_indices,]
```

Simple Baseline Predictions

```
#using a handful of vars I identified as probably useful on a first pass
simple.model = glm(y~ job + marital + month + newpdays + poutcome, data = clean_bank_20, family = binom)

#VIF of simple model

(vif(simple.model)[,3])^2

##      job  marital    month newpdays poutcome
## 1.024109 1.054595 1.024809 1.826933 3.427882

#Prediction of simple model
pred = ifelse(simple.model$fitted.values > .5, 1, 0)
predtable = table(pred, clean_bank_20$y )
predtable

##
## pred    no    yes
##    0 36114  3797
##    1   434   843

#prediction of full model

full.model = glm(y~ age+ job + marital + education + default + housing + loan + contact
                + month + day_of_week + campaign + previous + poutcome + emp.var.rate +
                cons.price.idx + cons.conf.idx + euribor3m + nr.employed + newpdays , data = clean_b

fullpred = ifelse(full.model$fitted.values > .5, 1,0)
fullpredtable = table(fullpred, clean_bank_20$y)
fullpredtable

##
## fullpred    no    yes
##           0 36009  3570
##           1   539  1070
```