R Notebook

Eagle Xuhui Ying

10/24/2022

Libraries

```
options(warn = -1)
library(tidyverse)
library(tidymodels)
library(janitor)
library(skimr)
library(kknn)
library(ggplot2)
```

1. Import Data

```
churn <- read_csv("Churn_training.csv") %>% clean_names()

head(churn)
```

monthly_minutes <dbl>	customer_service_calls <dbl>	streaming_minutes <dbl>	total_billed <dbl>	prev_balance <dbl>	late_pa
22604	2	26525	285	47	

monthly_minutes <dbl>	customer_service_calls <dbl>	streaming_minutes <dbl>	total_billed <dbl>	prev_balance <dbl>	late_pa
17059	2	16887	201	45	
25848	2	26783	264	44	
22080	3	23649	274	49	
23871	3	7705	236	61	
28098	3	12062	307	58	

6 rows | 1-6 of 34 columns

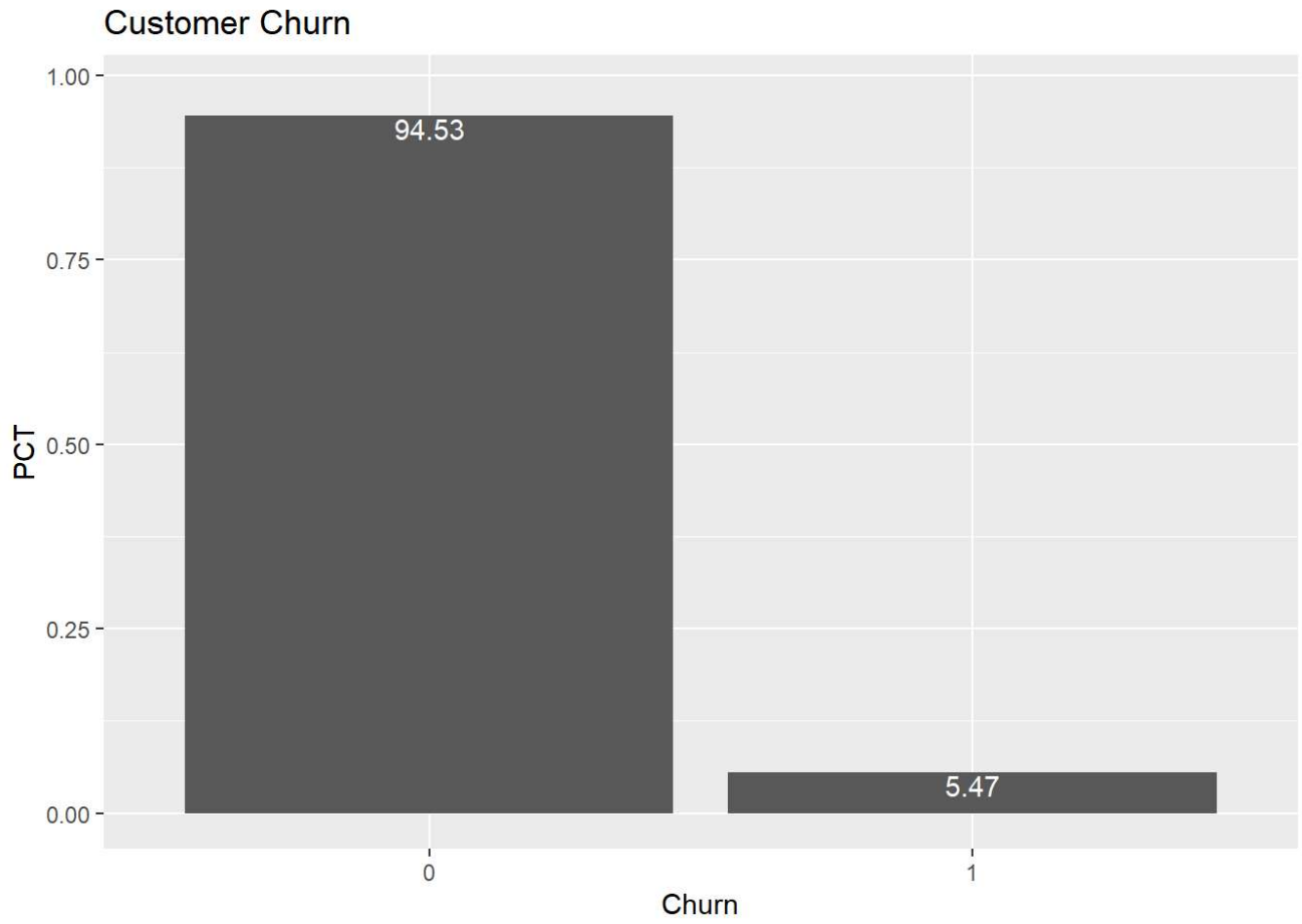
```
churn_kaggle <- read_csv("Churn_holdout.csv") %>% clean_names()


head(churn_kaggle)
```

monthly_minutes <dbl>	customer_service_calls <dbl>	streaming_minutes <dbl>	total_billed <dbl>	prev_balance <dbl>	late_pa
21721	1	14283	220	80	
20412	1	14789	202	43	
21228	1	35096	255	41	
18925	3	15776	282	48	
18488	1	9845	182	62	
17218	2	25357	291	58	

6 rows | 1-6 of 33 columns

2. Explore Target

```
churn_summary <- churn %>%
  count(churn) %>%
  mutate(pct = n/sum(n))


churn_summary %>%
  ggplot(aes(x=factor(churn),y=pct)) +
  geom_col()  +
  geom_text(aes(x=factor(churn),y=pct+0.034, label = round(pct*100,2)), vjust = 2.25, colour = "white") +
  labs(title="Customer Churn ", x="Churn", y="PCT")
```



3. Explore your data

```
churn %>% skimr::skim_without_charts()
```

Data summary

Name	Piped data
Number of rows	90901
Number of columns	34
_____	
Column type frequency:	
character	22
numeric	12
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
customer_reg_date	27	1	8	10	0	308	0
email_domain	28	1	9	11	0	3	0
phone_model	25	1	9	29	0	15	0
billing_city	29	1	6	24	0	8140	0
billing_state	26	1	4	14	0	48	0
partner	25	1	2	3	0	2	0
phone_service	25	1	2	3	0	2	0
multiple_lines	24	1	2	3	0	2	0
streaming_plan	28	1	3	8	0	4	0
mobile_hotspot	36	1	2	3	0	2	0
wifi_calling_text	32	1	2	3	0	2	0
online_backup	29	1	2	18	0	3	0
device_protection	29	1	1	1	0	26	0
contract_code	26	1	1	1	0	26	0
currency_code	29	1	3	3	0	3	0
maling_code	31	1	1	1	0	26	0
paperless_billing	31	1	2	3	0	2	0
payment_method	24	1	11	16	0	4	0
customer_id	0	1	7	20	0	90901	0
billing_address	20	1	10	38	0	90880	0
gender	27	1	4	6	0	2	0
network_speed	27	1	2	5	0	2	0

Variable type: numeric

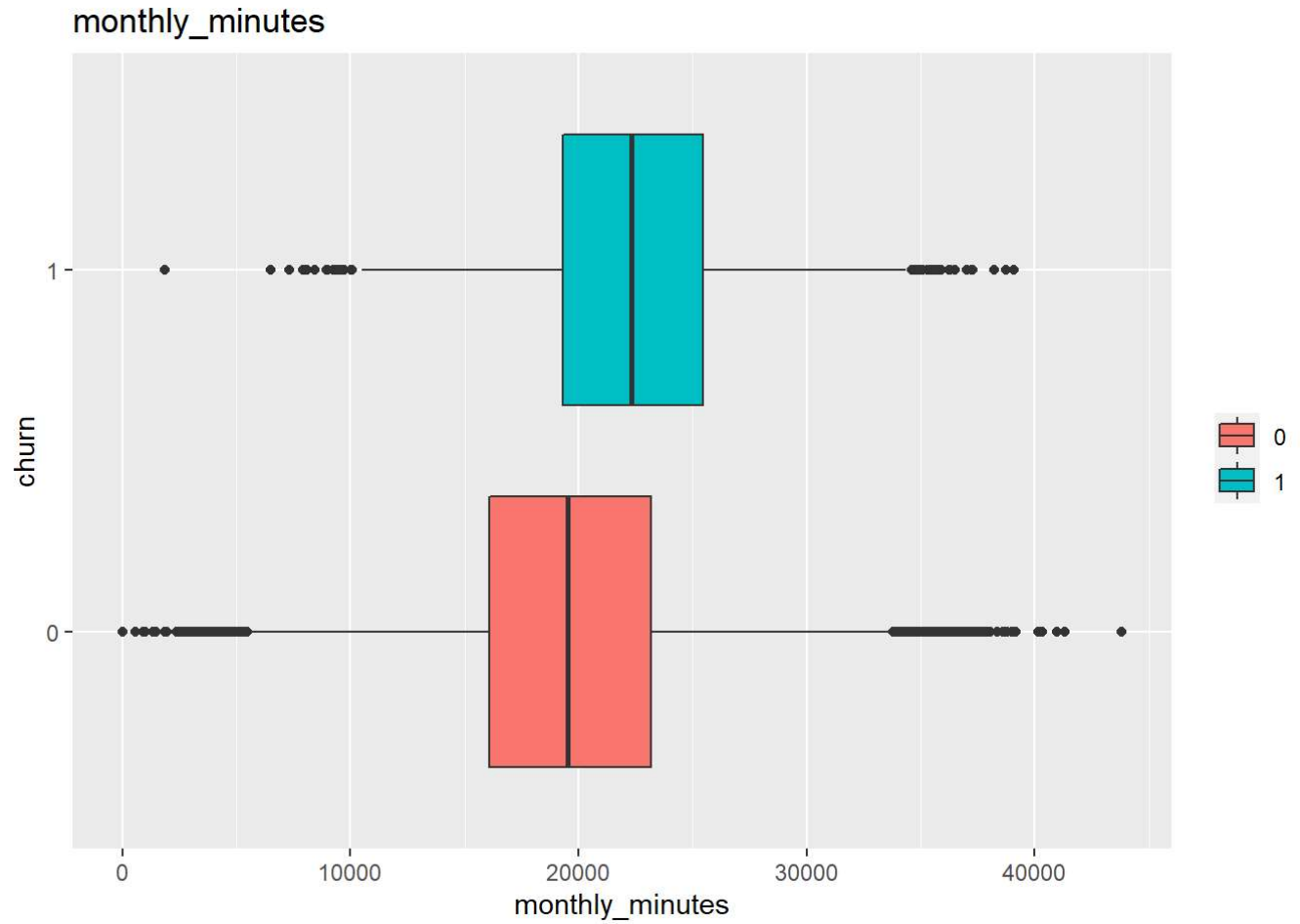
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
monthly_minutes	20	1	19851.97	5117.73	0	16244	19694	23337	43799
customer_service_calls	22	1	1.65	0.66	0	1	2	2	4
streaming_minutes	22	1	20696.93	4988.01	0	17327	20671	24023	43799
total_billed	34	1	250.25	35.58	100	226	251	274	399

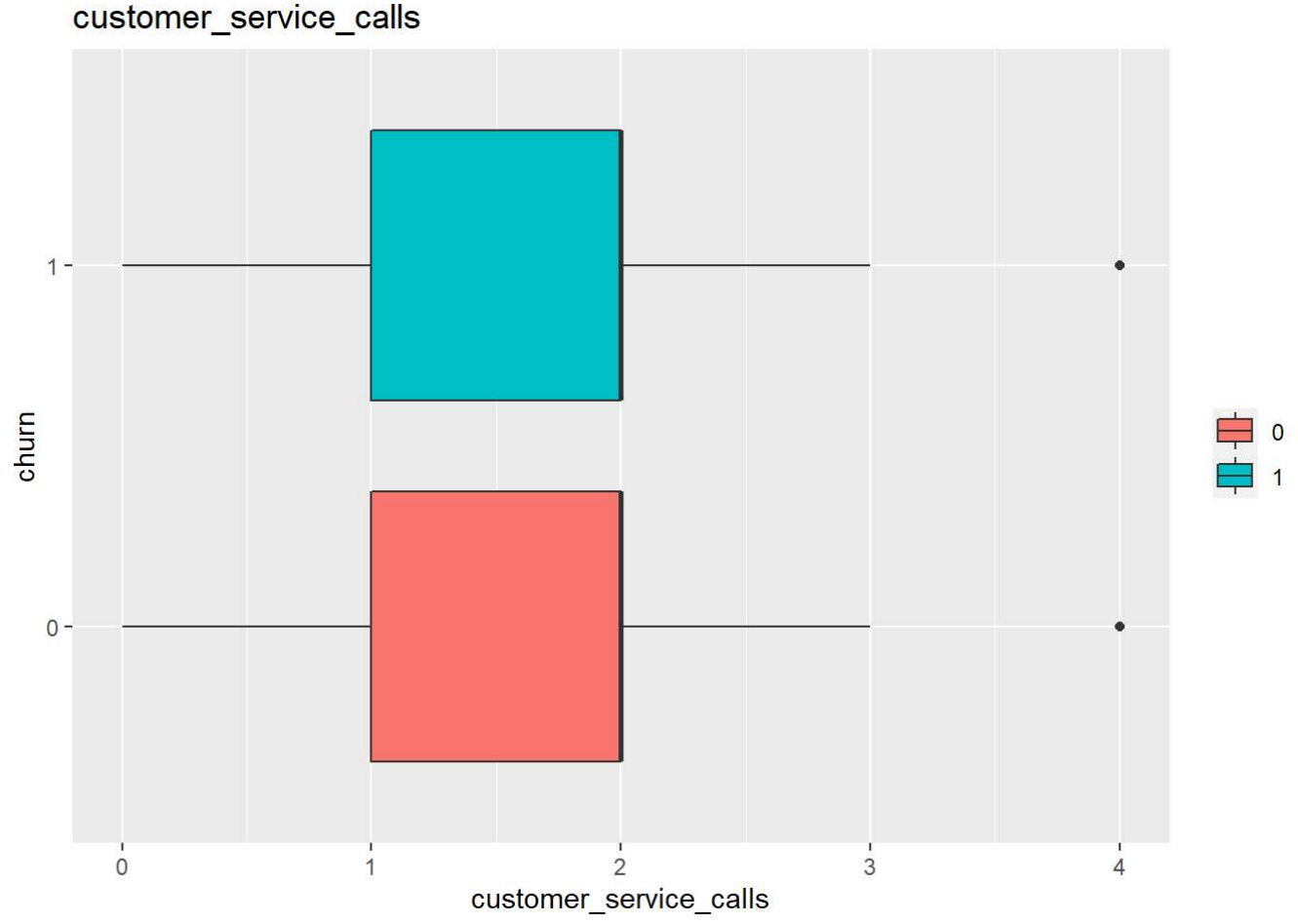
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
prev_balance	22	1	51.46	11.92	0	43	51	59	99
late_payments	20	1	4.80	1.32	0	4	5	6	9
ip_address_asn	17	1	34846.93	16862.57	2013	18773	26969	51472	65533
phone_area_code	28	1	247.56	10.66	200	240	248	255	289
billing_postal	28	1	50124.16	28586.74	502	25325	49849	75058	99922
number_phones	30	1	5.31	1.09	0	5	5	6	10
senior_citizen	35	1	0.50	0.50	0	0	0	1	1
churn	0	1	0.05	0.23	0	0	0	0	1

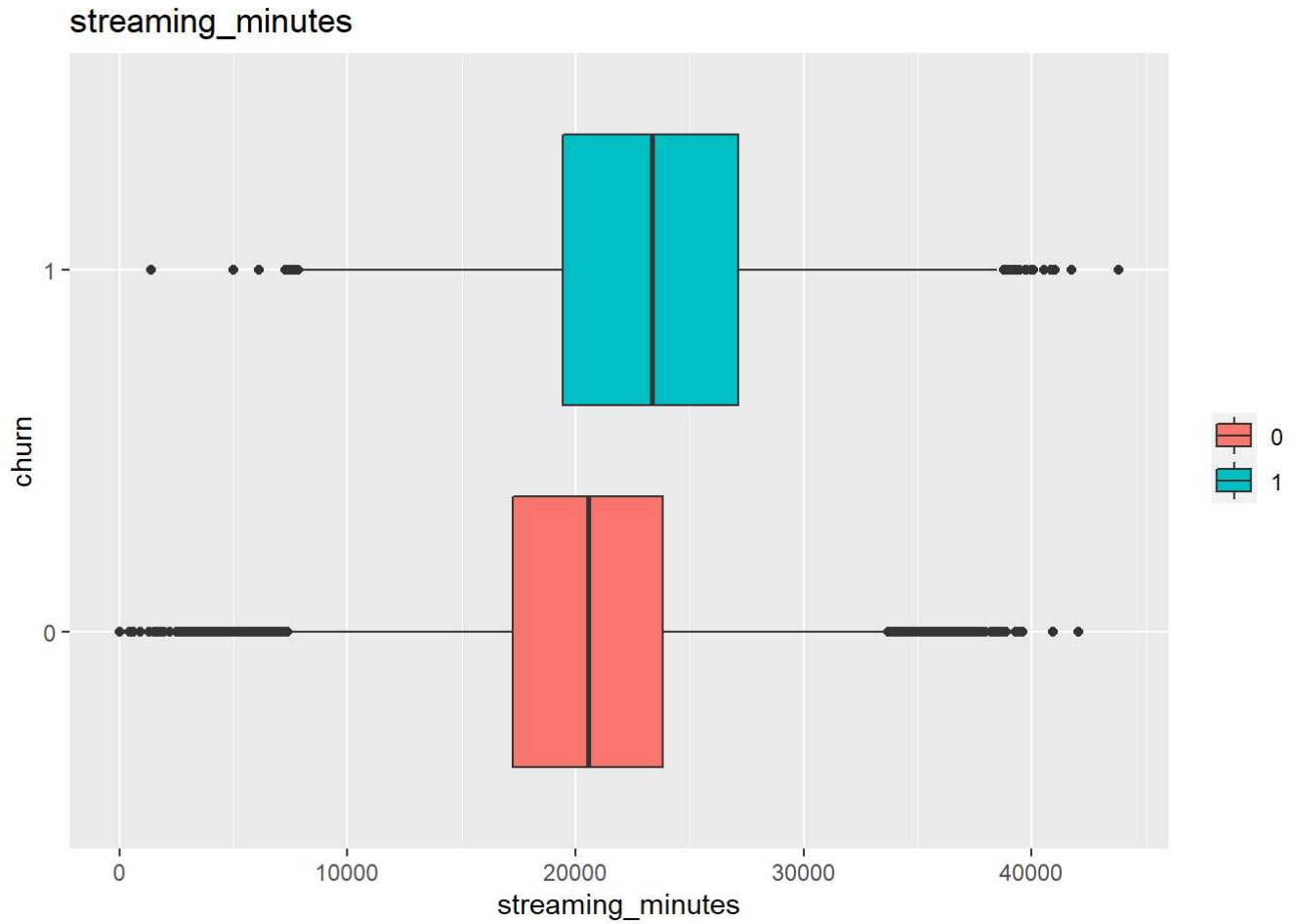
Explore numerics

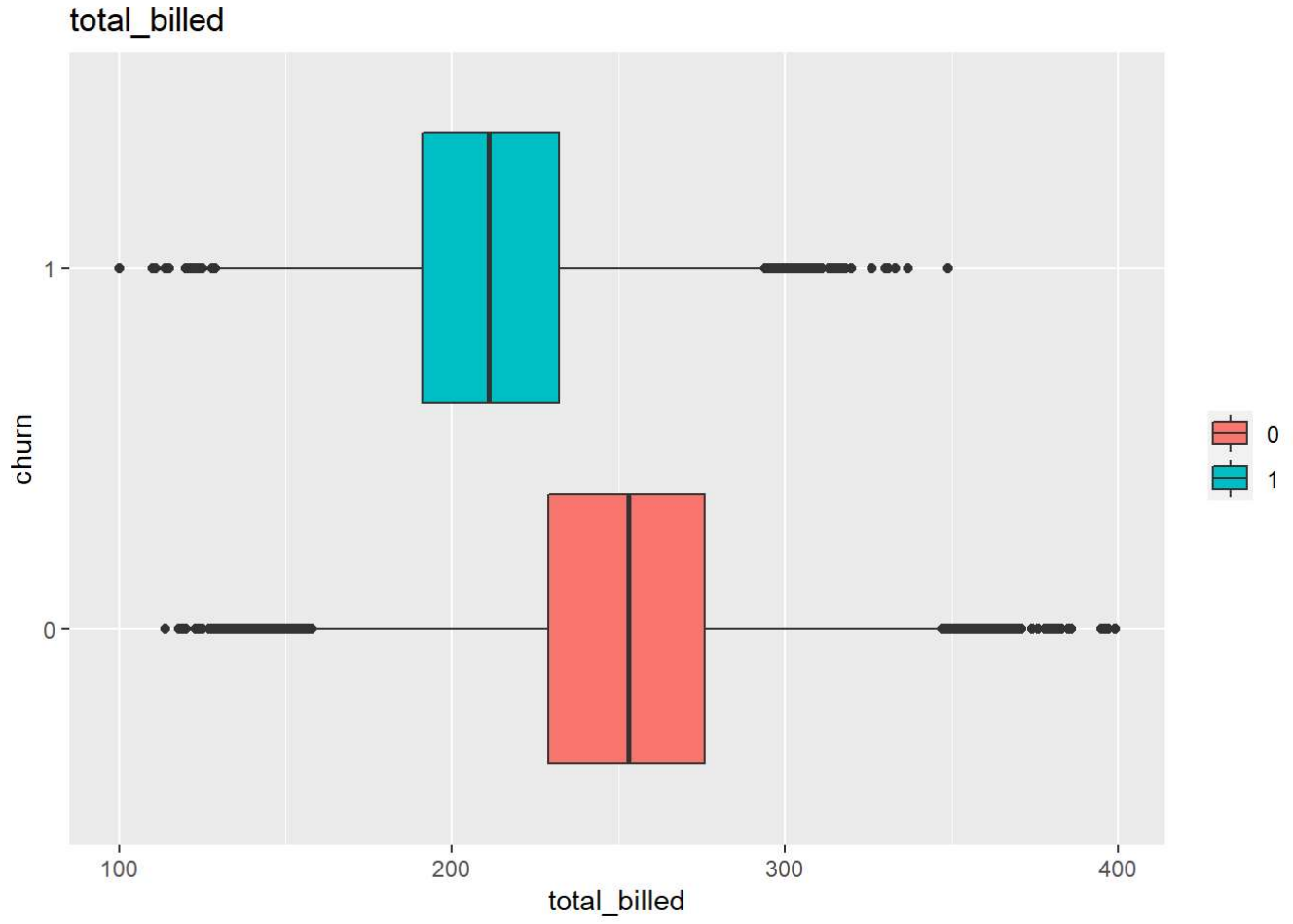
numeric variables: monthly_minutes, customer_service_calls, streaming_minutes, total_billed, prev_balance, late_payments, number_phones

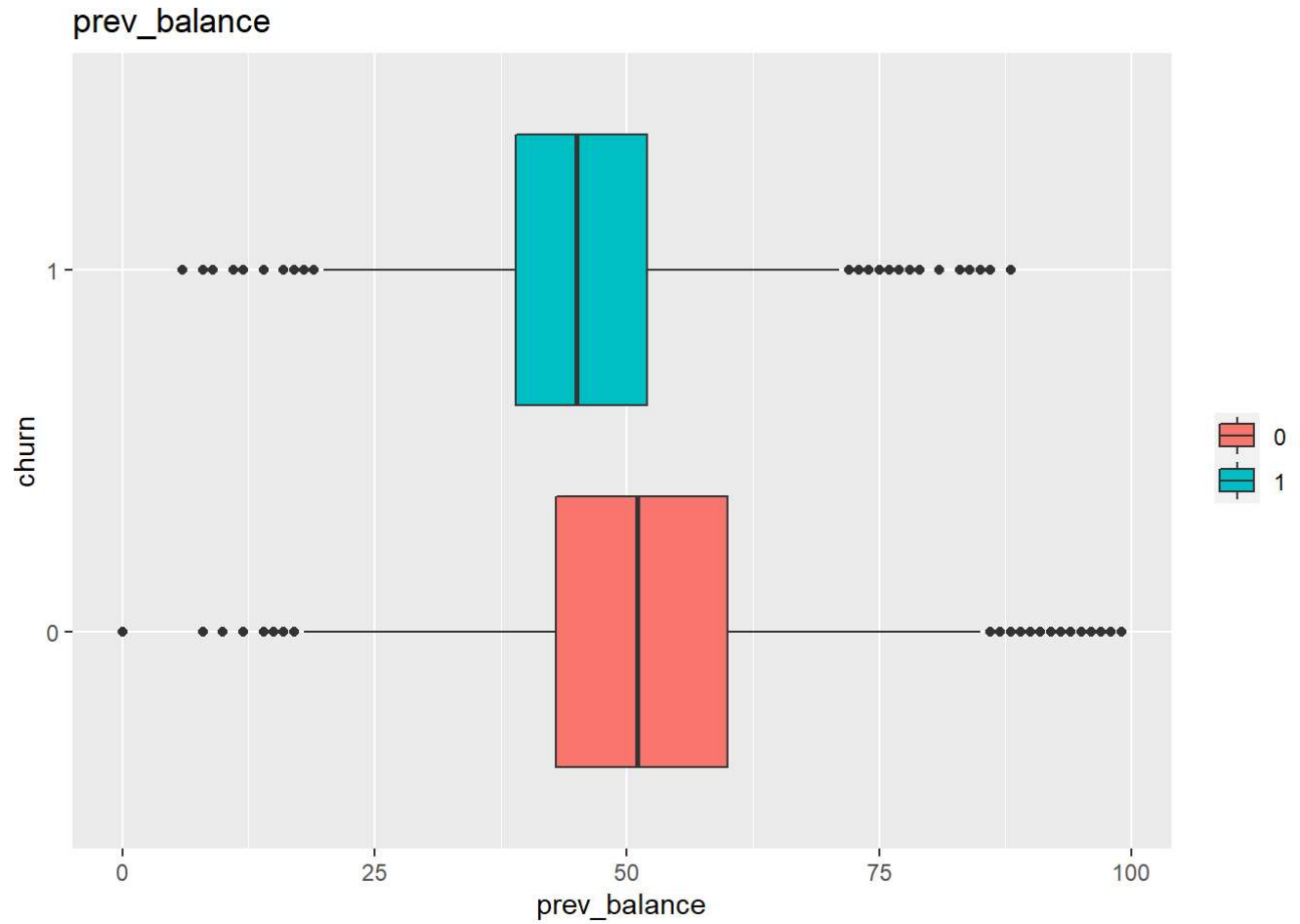
```
# -- comparative boxplots

boxplot <- function(m){
    ggplot(churn, aes(x=!!as.name(m), y=as.factor(churn), fill=as.factor(churn))) +
    geom_boxplot() +
    labs(title = as.character(m), y = 'churn') +
    theme(legend.title = element_blank())
}

numerics <- c('monthly_minutes', 'customer_service_calls', 'streaming_minutes', 'total_billed', 'prev_balanc
         e', 'late_payments', 'number_phones')

for (c in numerics){
    print(boxplot(c))
}
```

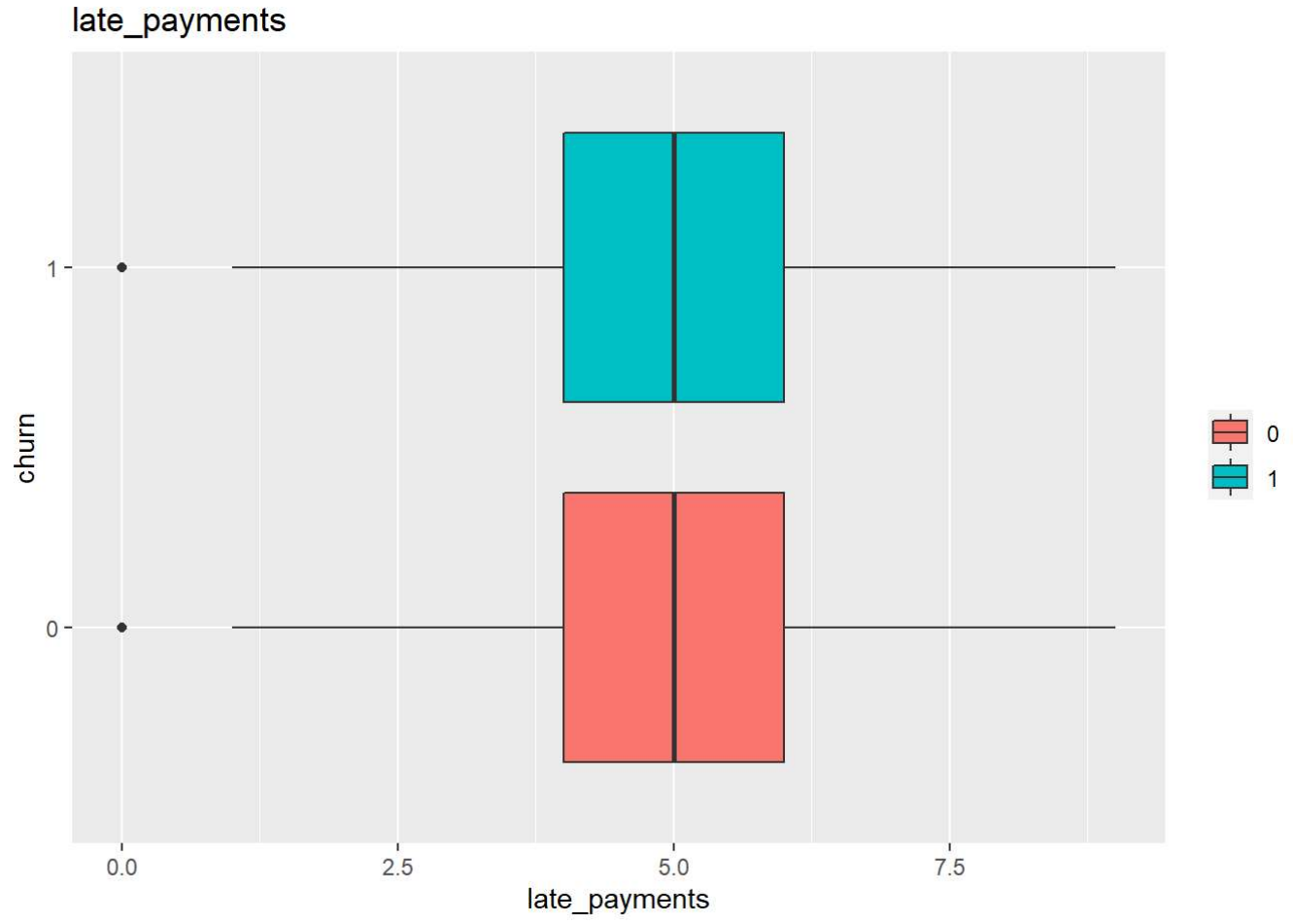

monthly_minutes
churn
1
0
0
10000
20000
30000
40000
monthly_minutes
0
1

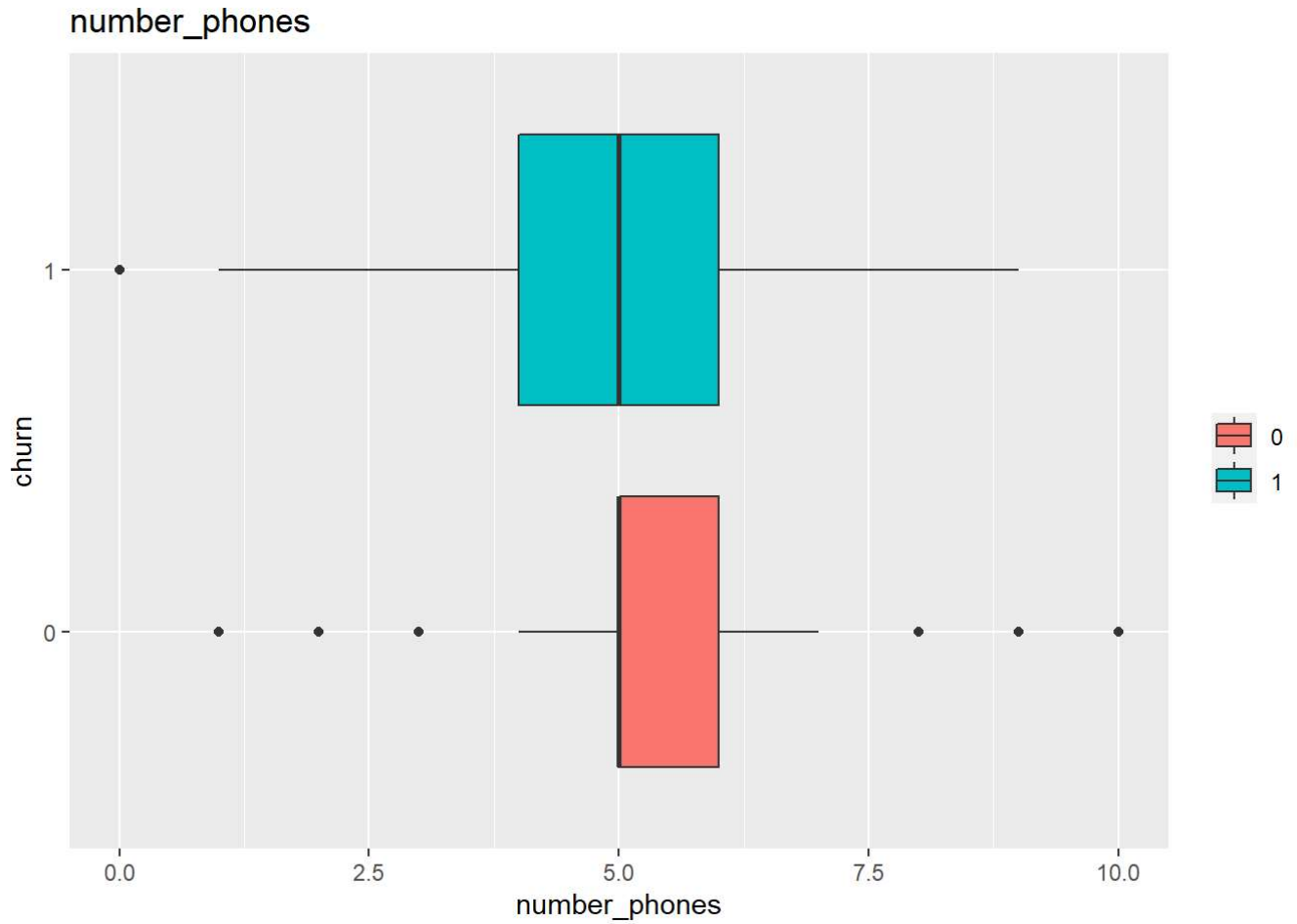

customer_service_calls
churn
1
0
0
1
2
3
4
customer_service_calls
0
1


streaming_minutes
churn
1
0
0
10000
20000
30000
40000
streaming_minutes
0
1


total_billed
churn
1
0
100
200
300
400
total_billed
0
1


prev_balance
churn
1
0
0
25
50
75
100
prev_balance
0
1

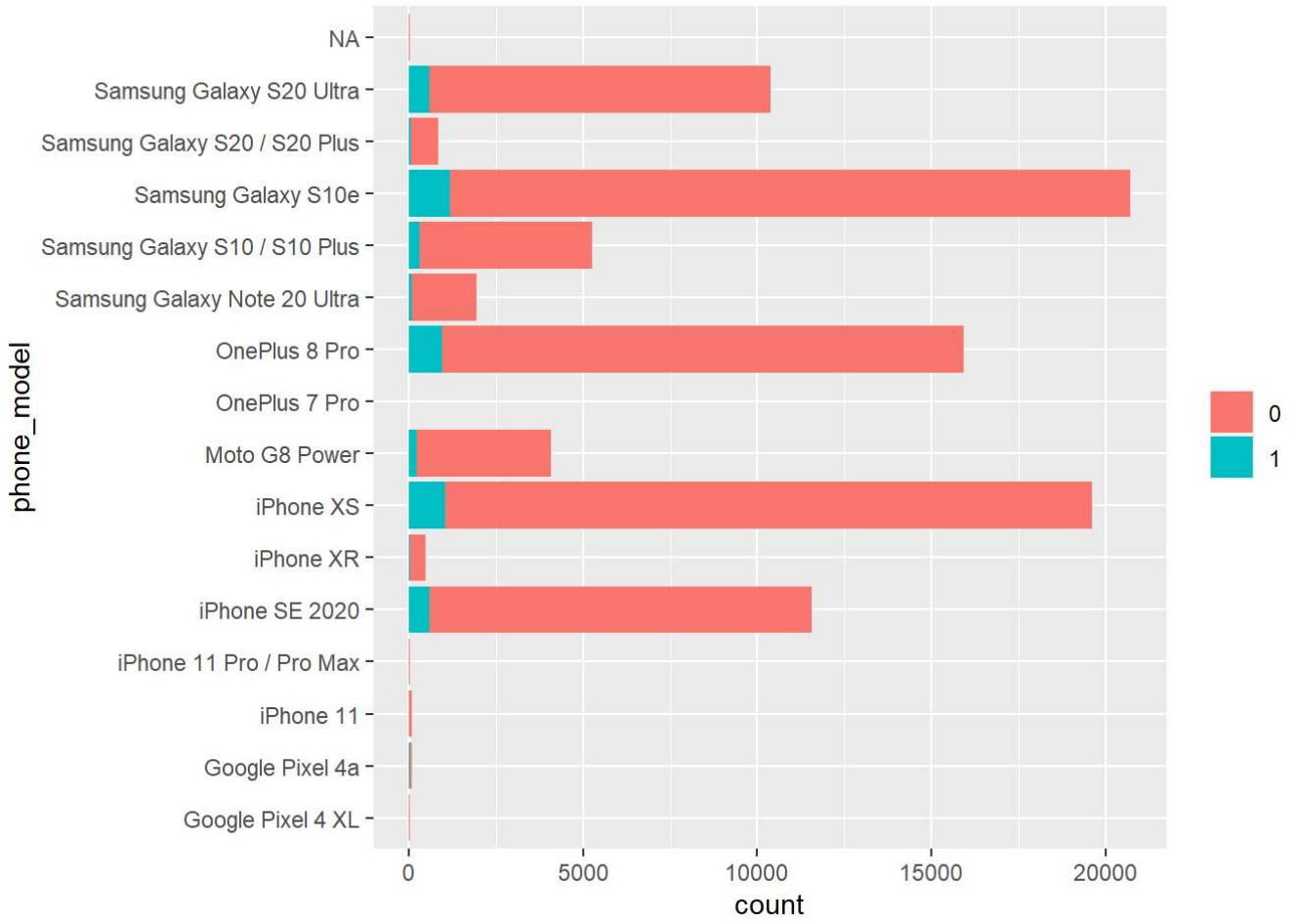

late_payments
churn
1
0
0.0
2.5
5.0
7.5
late_payments
0
1

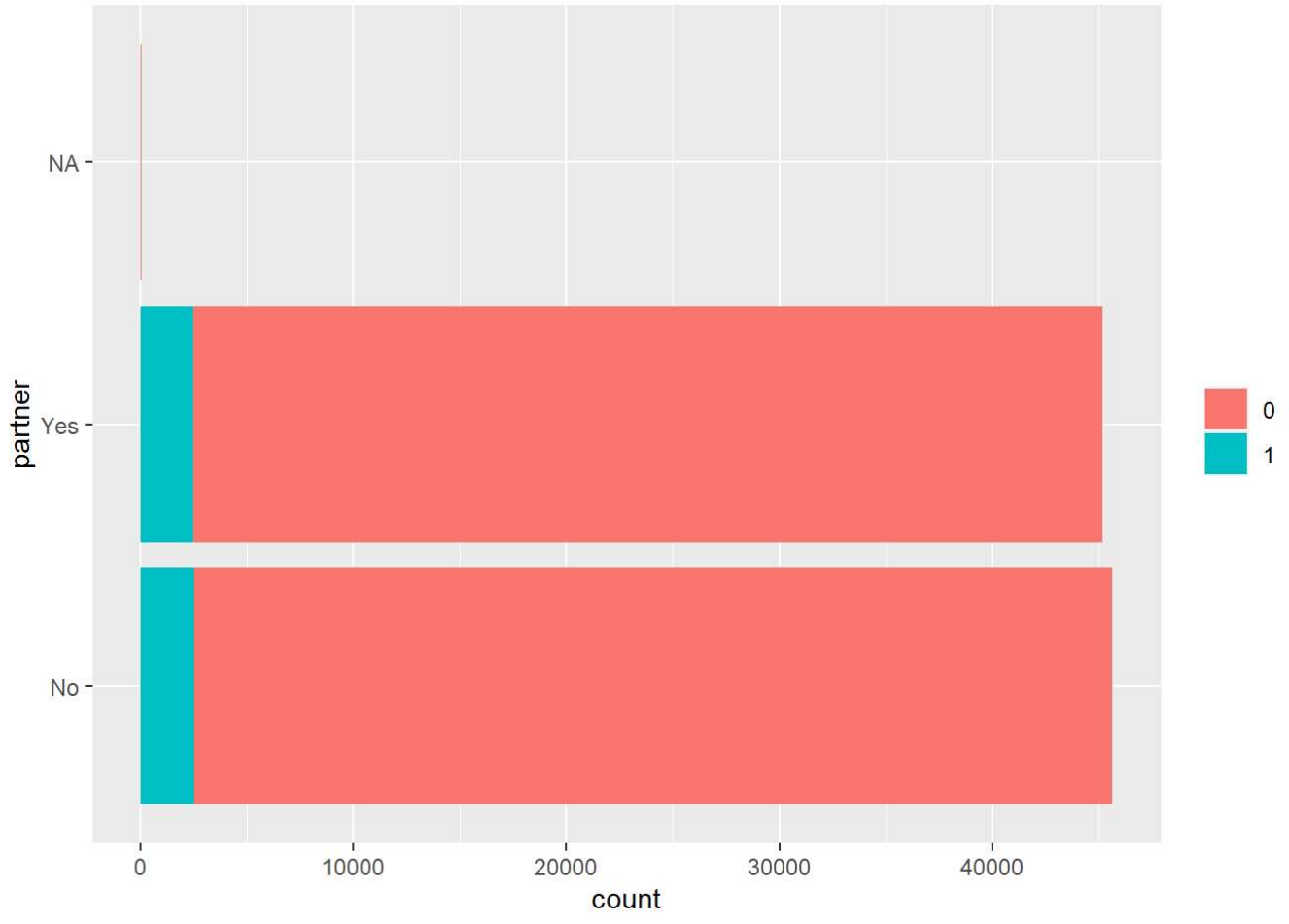


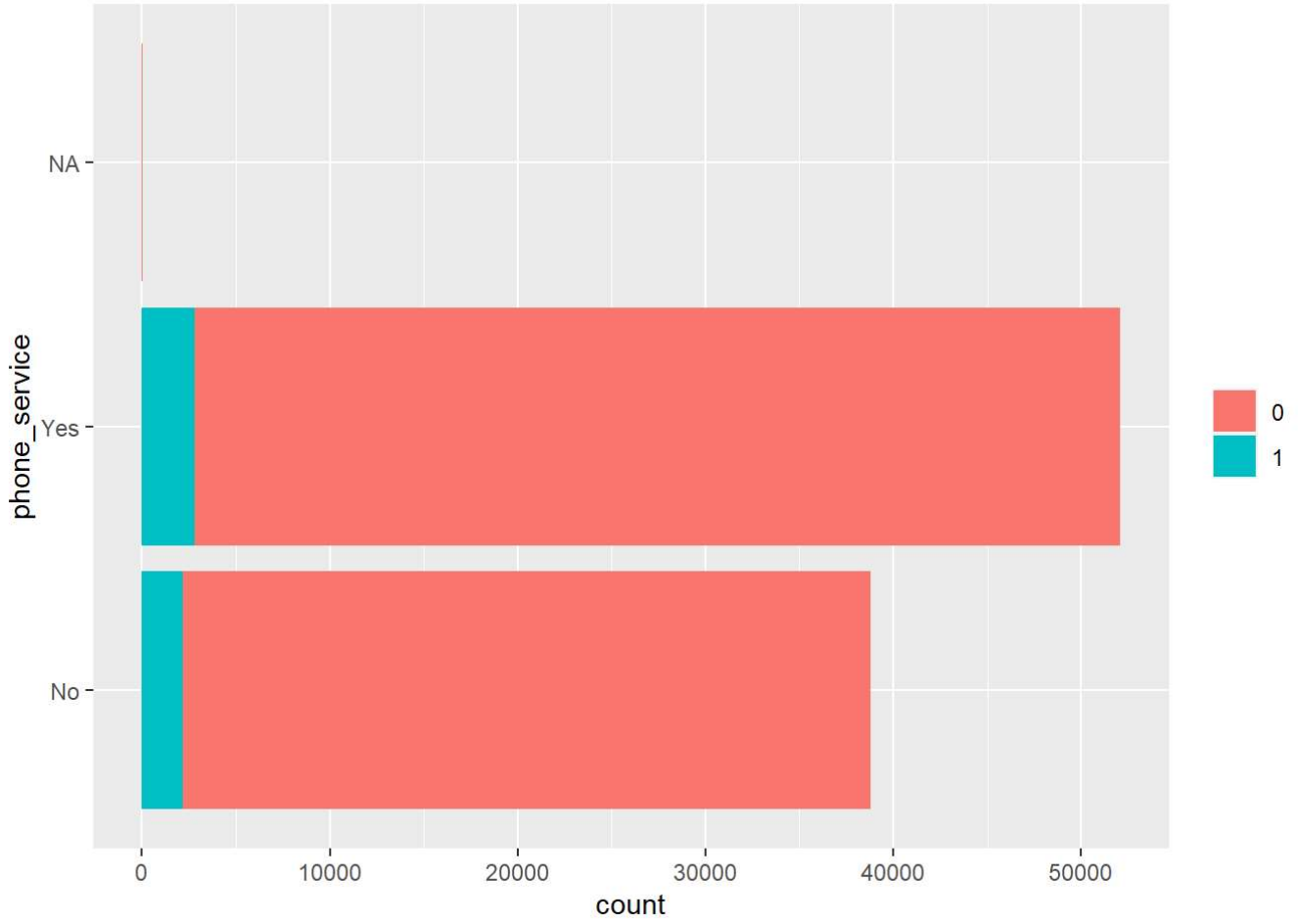
Explore character variables

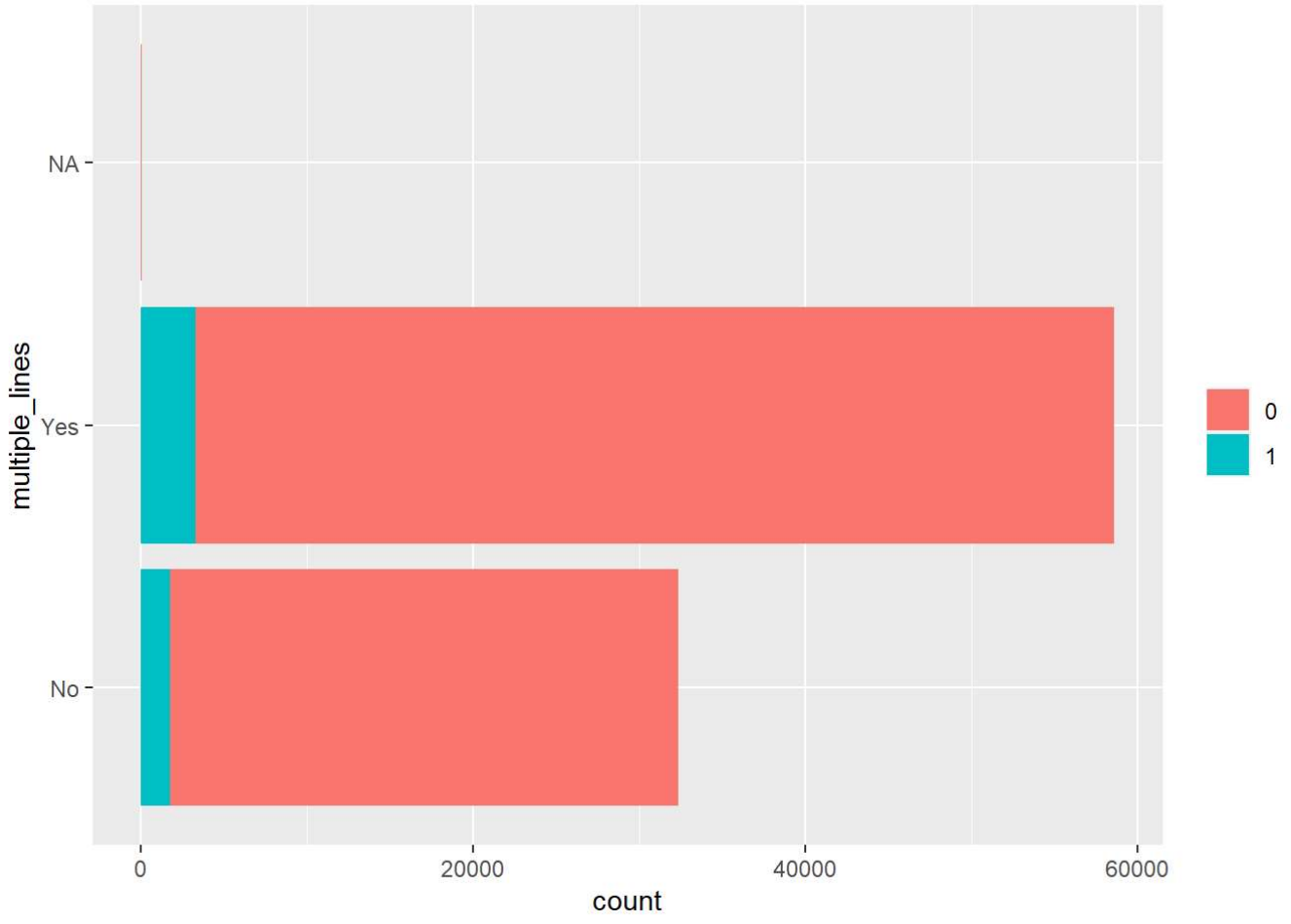
categorical variables: phone_model, partner, phone_service, multiple_lines, streaming_plan, mobile_hotspot, wifi_calling_text, online_backup, paperless_billing, payment_method, gender, network_speed, senior_citizen

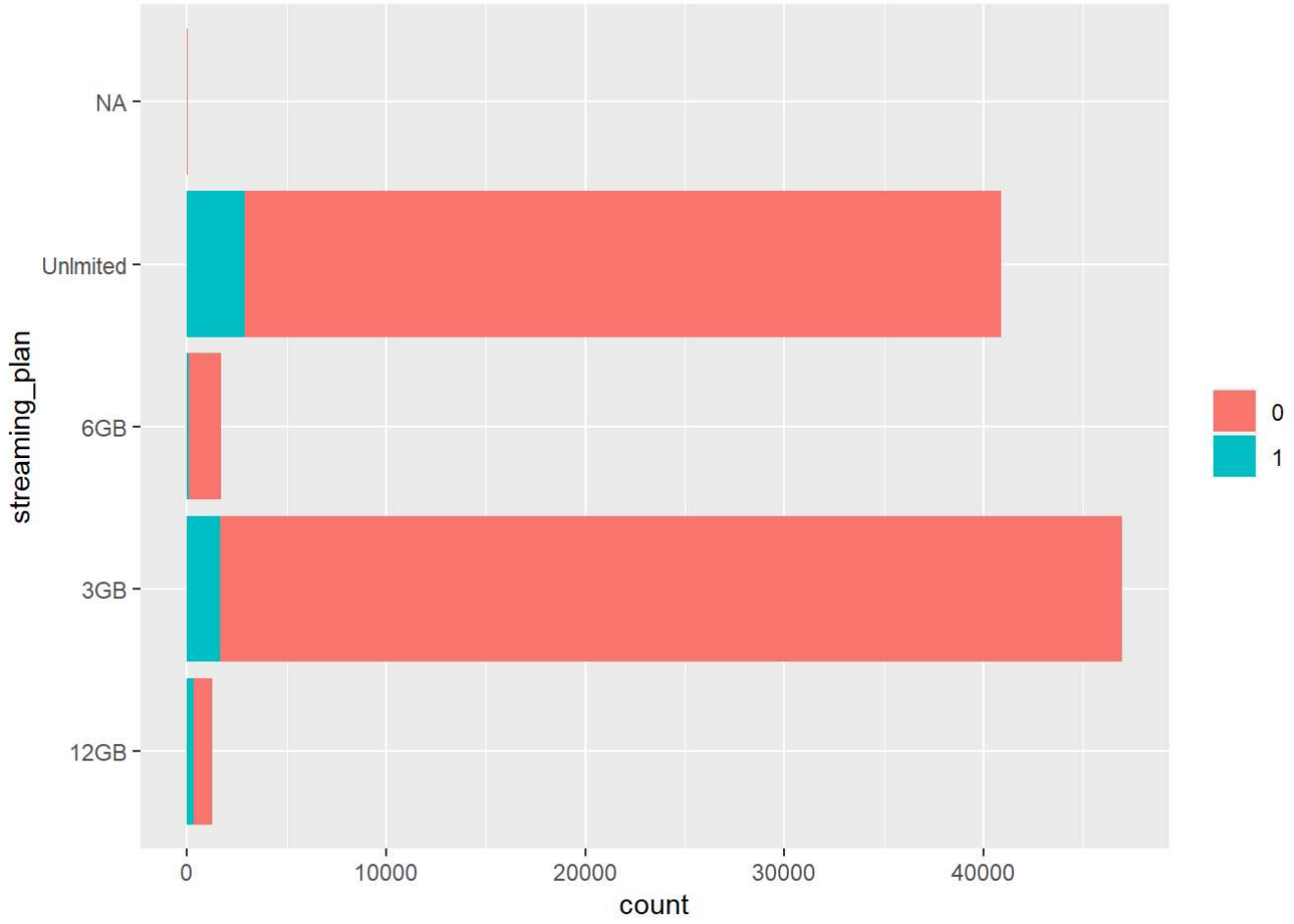
```
churn$senior_citizen <- as.factor(churn$senior_citizen)
churn_kaggle$senior_citizen <- as.factor(churn_kaggle$senior_citizen)


char_explore <- function(col){
    churn %>%
    ggplot(aes(!!as.name(col))) +
    geom_bar(aes(fill = as.factor(churn))) +
    coord_flip() +
    theme(legend.title = element_blank())
}


dummy <- c('phone_model', 'partner', 'phone_service', 'multiple_lines', 'streaming_plan', 'mobile_hotspot',
          'wifi_calling_text', 'online_backup', 'paperless_billing', 'payment_method', 'gender', 'network_sp
          eed', 'senior_citizen')


# -- for each character column, create a chart
for (column in dummy){
    chrt <- char_explore(column)
    print(chrt)
}
```

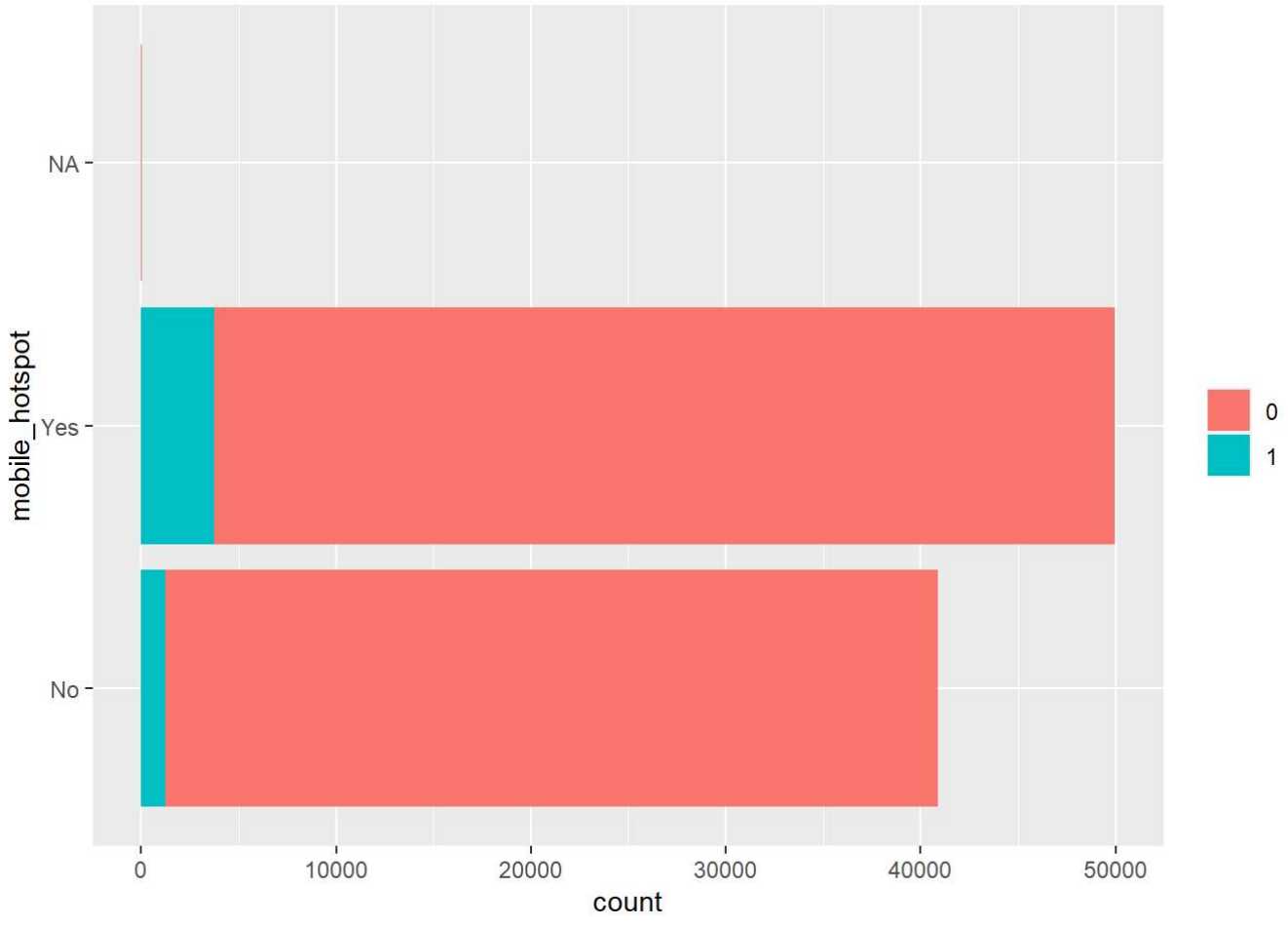

NA
Samsung Galaxy S20 Ultra
Samsung Galaxy S20 / S20 Plus
Samsung Galaxy S10e
Samsung Galaxy S10 / S10 Plus
Samsung Galaxy Note 20 Ultra
OnePlus 8 Pro
OnePlus 7 Pro
Moto G8 Power
iPhone XS
iPhone XR
iPhone SE 2020
iPhone 11 Pro / Pro Max
iPhone 11
Google Pixel 4a
Google Pixel 4 XL
phone_model
0
5000
10000
15000
20000
count
0
1

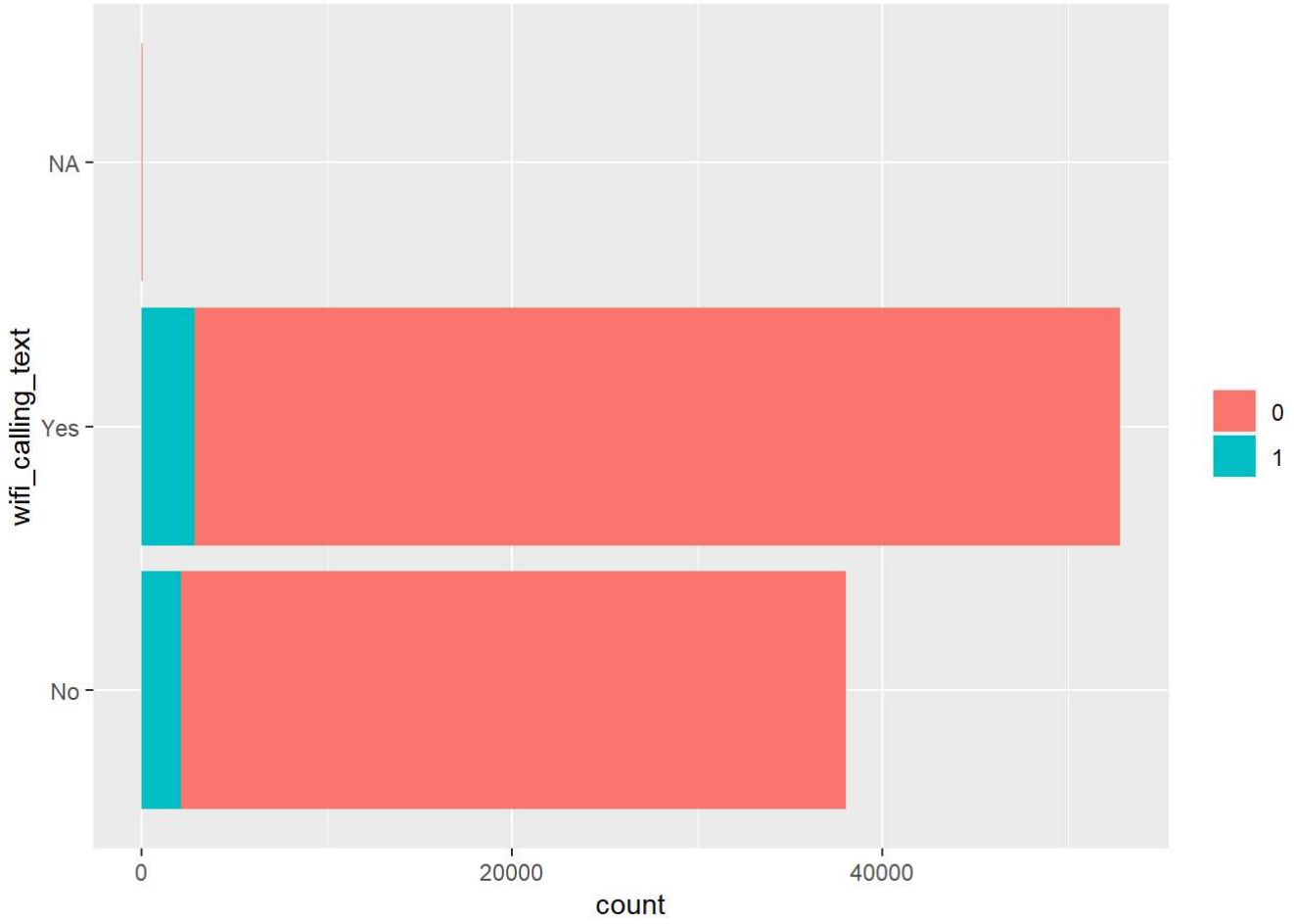

NA
Yes
No
partner
0
10000
20000
30000
40000
count
0
1

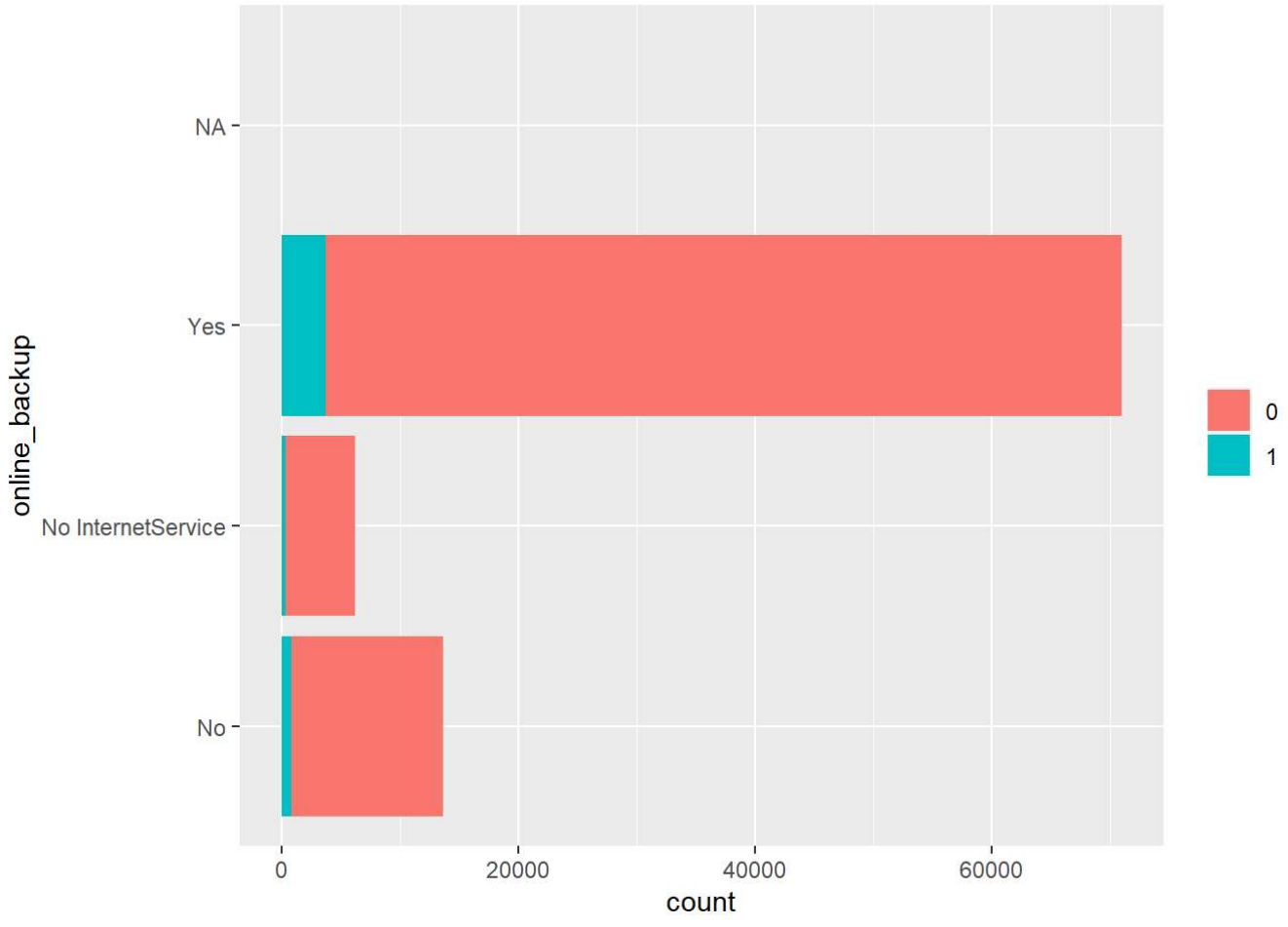

phone_service
NA
Yes
No
0
10000
20000
30000
40000
50000
count
0
1

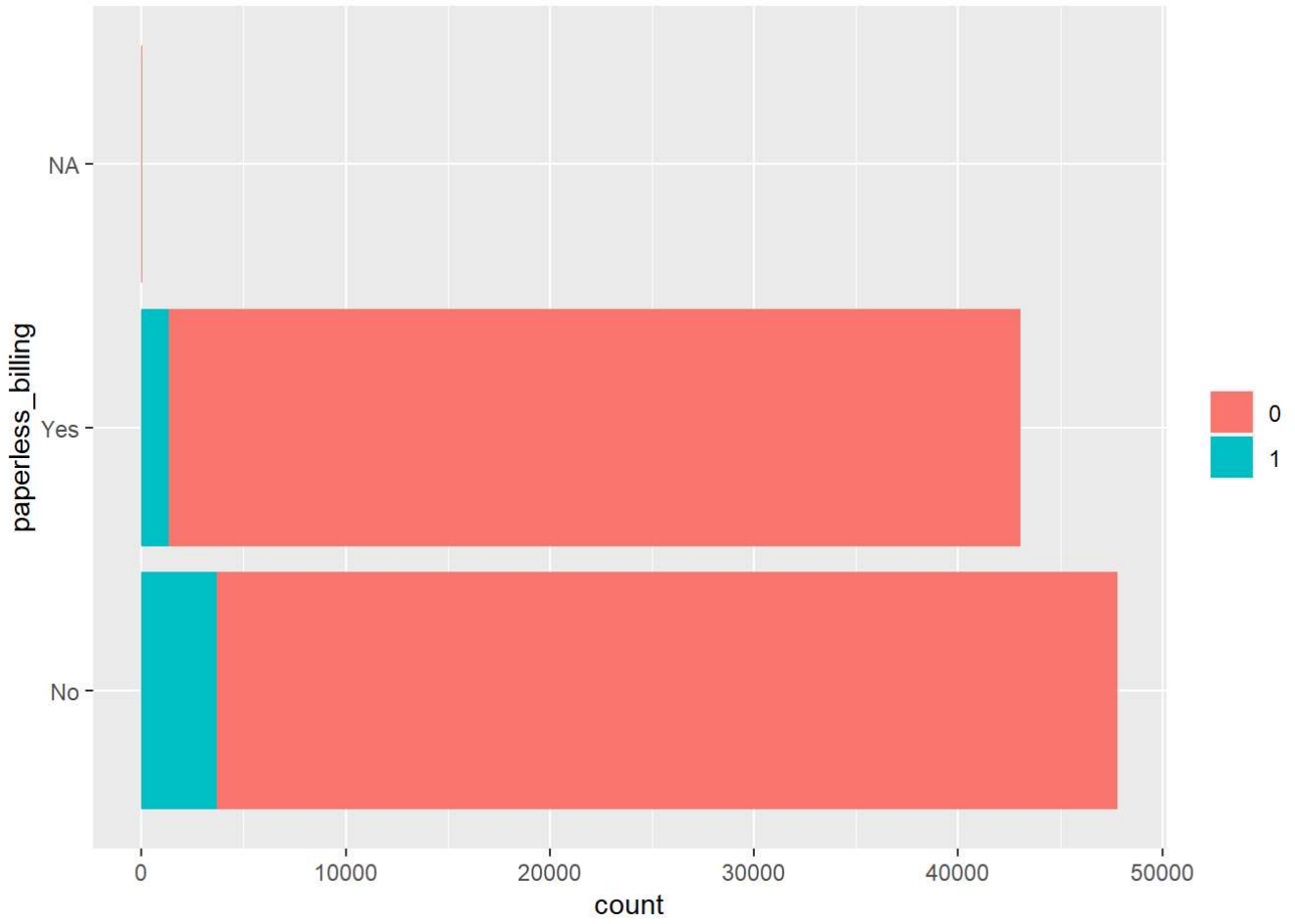

multiple_lines
NA
Yes
No
0
20000
40000
60000
count
0
1

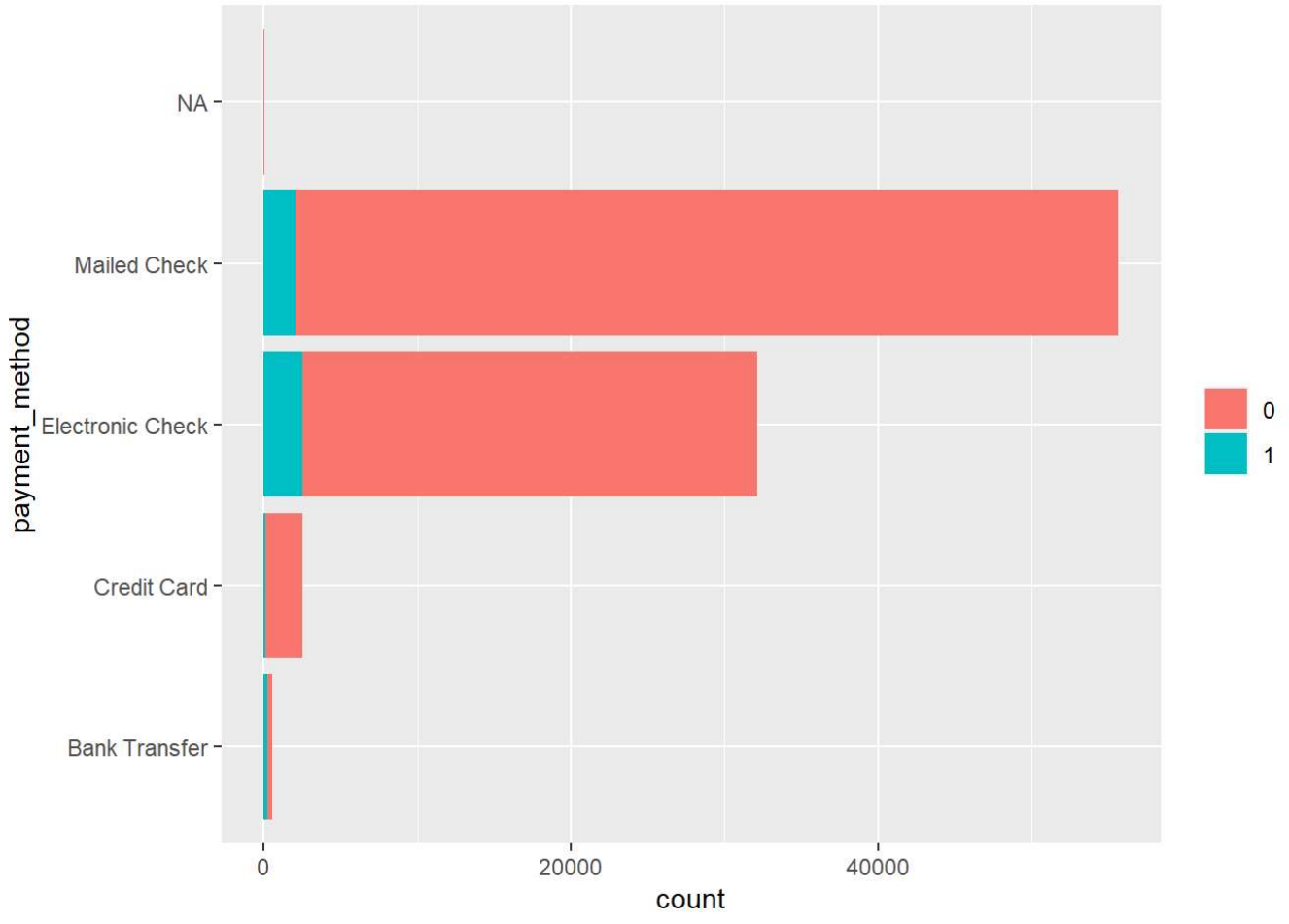

NA
Unlmited
6GB
3GB
12GB
streaming_plan
0
10000
20000
30000
40000
count
0
1

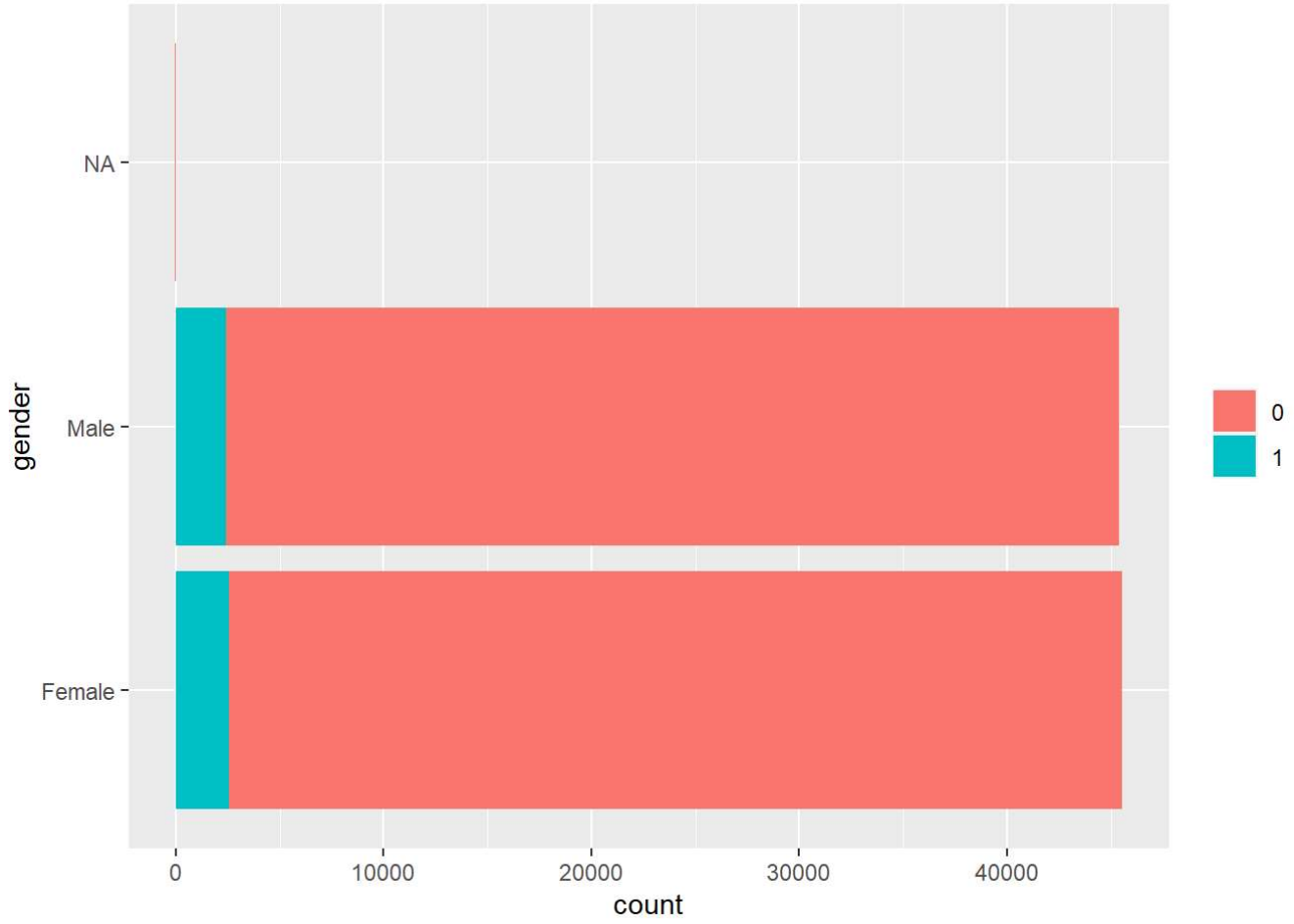

NA
Yes
No
mobile_hotspot
0
10000
20000
30000
40000
50000
count
0
1

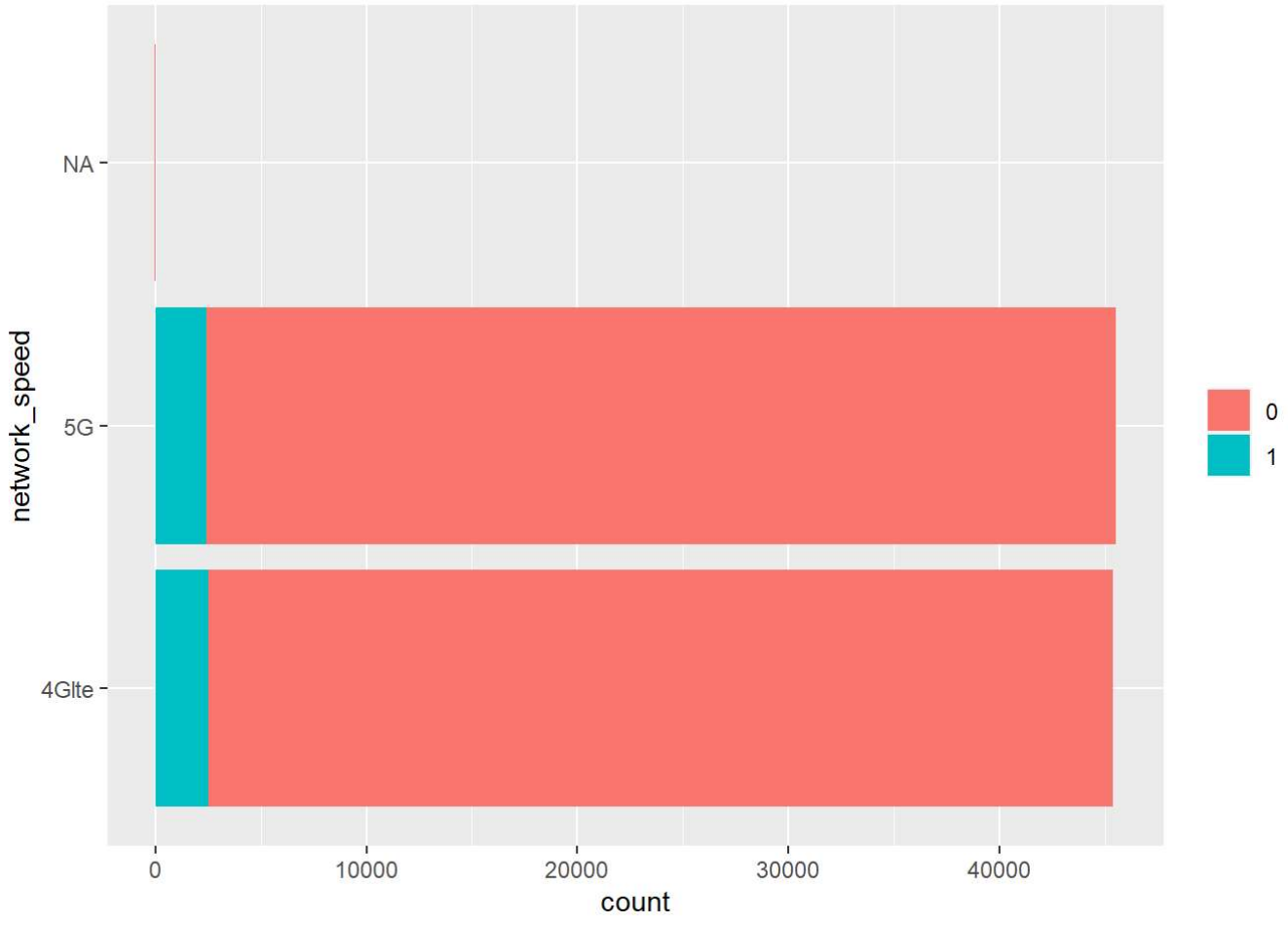

wifi_calling_text
NA
Yes
No
0
20000
40000
count
0
1

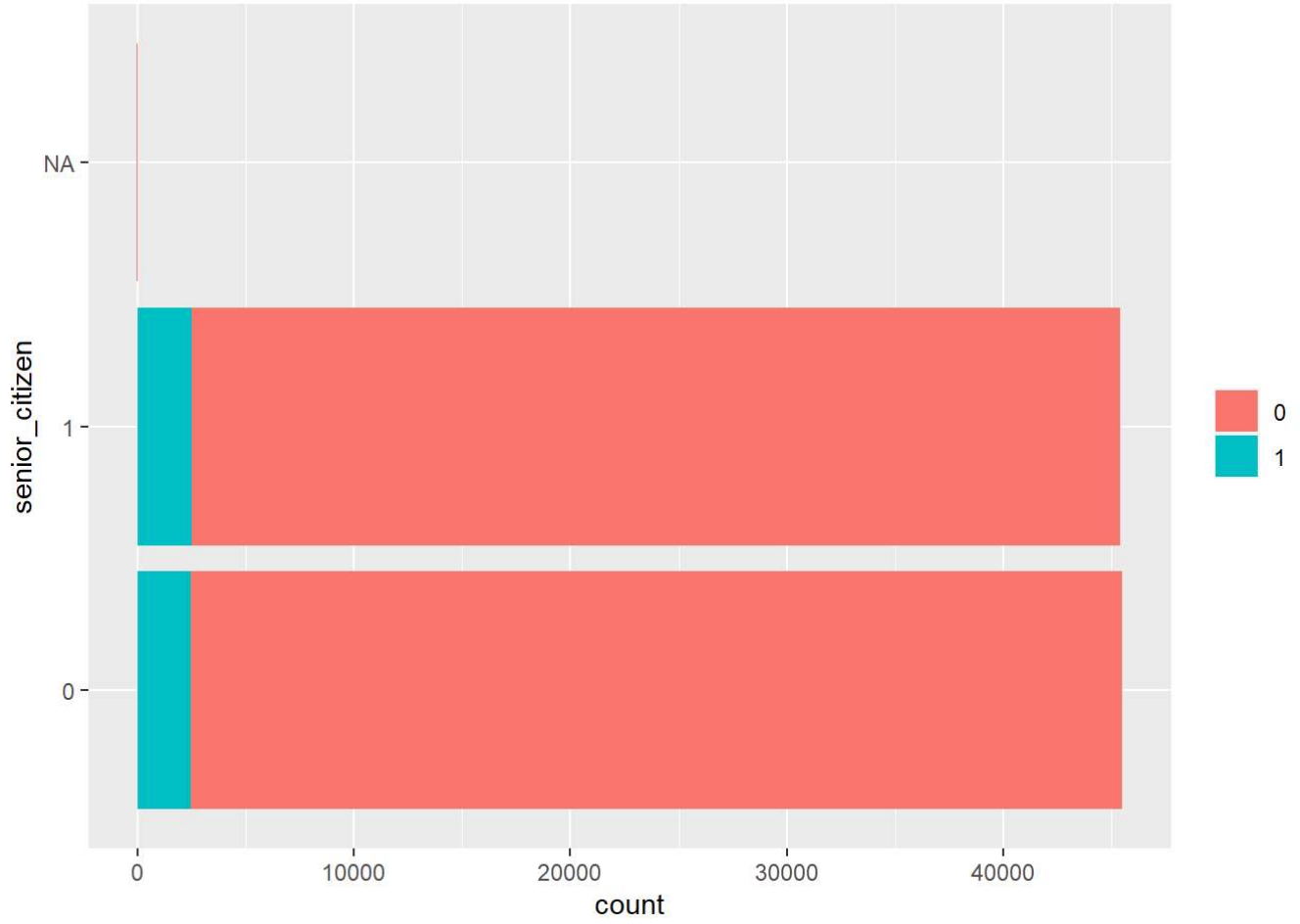

online_backup
NA
Yes
No InternetService
No
0
20000
40000
60000
count
0
1


paperless_billing
NA
Yes
No
0
10000
20000
30000
40000
50000
count
0
1


payment_method
NA
Mailed Check
Electronic Check
Credit Card
Bank Transfer
0
20000
40000
count
0
1


gender
NA
Male
Female
0
10000
20000
30000
40000
count
0
1


network_speed
NA
5G
4Glte
0
10000
20000
30000
40000
count
0
1



4. Transform

Convert categories to factors

```
churn_prep <- churn %>%
   mutate(churn = as.factor(churn)) %>%
   mutate_if(is.character, factor)

churn_prep %>% head()
```

monthly_minutes <dbl>	customer_service_calls <dbl>	streaming_minutes <dbl>	total_billed <dbl>	prev_balance <dbl>	late_pa
22604	2	26525	285	47	
17059	2	16887	201	45	
25848	2	26783	264	44	
22080	3	23649	274	49	
23871	3	7705	236	61	
28098	3	12062	307	58	

6 rows | 1-6 of 34 columns

```
churn_kaggle <- churn_kaggle %>%
   mutate_if(is.character, factor)
```

5. Partition your data into 70/30 train/test split

```
set.seed(123)
x <- initial_split(churn_prep, prop = 0.7, strata = churn)
train <- training(x)
test  <- testing(x)

sprintf("Train PCT : %1.2f%%", nrow(train)/nrow(churn_prep) * 100)
```

```
## [1] "Train PCT : 70.00%"
```

```
sprintf("Test  PCT : %1.2f%%", nrow(test)/nrow(churn_prep) * 100)
```

```
## [1] "Test  PCT : 30.00%"
```

6. Define Recipe

```
knn_recipe <- recipe(churn ~ monthly_minutes + customer_service_calls + streaming_minutes + total_billed + p
          rev_balance + late_payments + phone_model + partner + phone_service + multiple_lines + streaming_p
          lan + mobile_hotspot + wifi_calling_text + online_backup + number_phones + paperless_billing + pay
          ment_method + gender + network_speed + senior_citizen, data=train) %>%
    step_impute_median(all_numeric_predictors()) %>%
    step_unknown(all_nominal_predictors()) %>%
    step_scale(all_numeric_predictors()) %>%
    step_dummy(all_nominal_predictors())

knn_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         20
##
## Operations:
##
## Median imputation for all_numeric_predictors()
## Unknown factor level assignment for all_nominal_predictors()
## Scaling for all_numeric_predictors()
## Dummy variables from all_nominal_predictors()
```

```
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  2939195 157.0    4934144 263.6  4934144 263.6
## Vcells 12210633  93.2   21049997 160.6 17474998 133.4
```

```
# eyeball recipe results
bake(knn_recipe %>% prep(), train, composition = "tibble") %>% head()
```

monthly_minutes <dbl>	customer_service_calls <dbl>	streaming_minutes <dbl>	total_billed <dbl>	prev_balance <dbl>	late_pa
3.329799	3.019829	3.387319	5.658620	3.772390	3.
5.045352	3.019829	5.372332	7.432218	3.688559	4.
4.309864	4.529744	4.743691	7.713741	4.107714	4.
2.804535	1.509915	3.827608	7.967112	5.951994	3.
4.702397	3.019829	4.162589	6.700257	3.269405	4.
3.827347	3.019829	5.863370	4.785898	3.269405	5.

6 rows | 1-6 of 52 columns

7. Define your Model(s)

```
knn_model <- nearest_neighbor(neighbors = 7) %>%
    set_mode("classification") %>%
    set_engine("kknn")

knn_model_2 <- nearest_neighbor(neighbors = 10) %>%
    set_mode("classification") %>%
    set_engine("kknn")
```

8. Workflow

```
knn_workflow <- workflow() %>%
    add_recipe(knn_recipe) %>%
    add_model(knn_model) %>%
    fit(train)

knn_workflow_2 <- workflow() %>%
    add_recipe(knn_recipe) %>%
    add_model(knn_model_2) %>%
    fit(train)
```

9. Score the model

```
options(yardstick.event_first = TRUE)
# -- score training
scored_train <- predict(knn_workflow, train, type="prob") %>%
    bind_cols(predict(knn_workflow, train, type="class")) %>%
    bind_cols(.,train)
# -- score testing
scored_test <- predict(knn_workflow, test, type="prob") %>%
    bind_cols(predict(knn_workflow, test, type="class")) %>%
    bind_cols(.,test)

scored_train_2 <- predict(knn_workflow_2, train, type="prob") %>%
    bind_cols(predict(knn_workflow_2, train, type="class")) %>%
    bind_cols(.,train)
# -- score testing
scored_test_2 <- predict(knn_workflow_2, test, type="prob") %>%
    bind_cols(predict(knn_workflow_2, test, type="class")) %>%
    bind_cols(.,test)
```

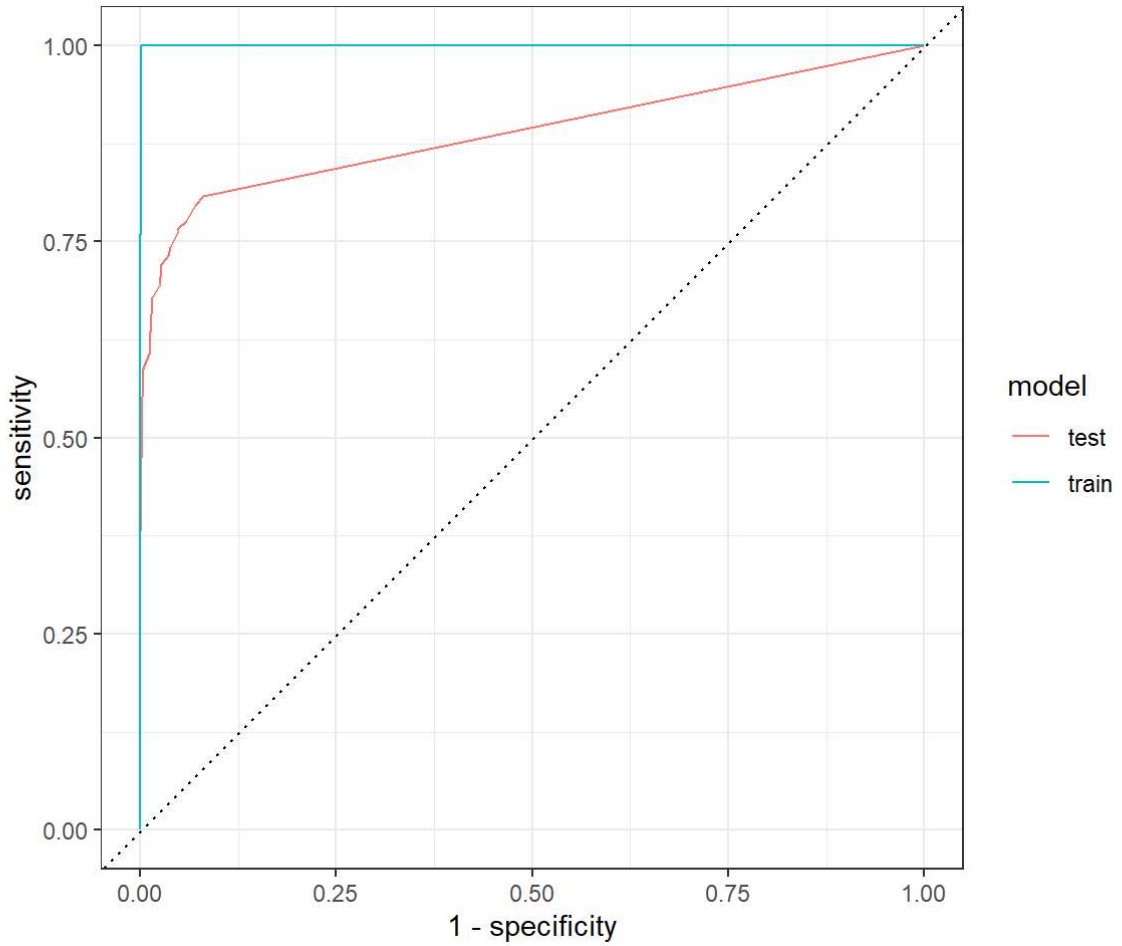
10. Evaluate (KNN = 7)

```
options(yardstick.event_first = FALSE)
# -- Metrics: Train and Test
scored_train %>%
    metrics(churn, .pred_1, estimate = .pred_class) %>%
    mutate(part="training") %>%
    bind_rows(scored_test %>%
                metrics(churn, .pred_1, estimate = .pred_class) %>%
                mutate(part="testing") ) %>%
    filter(.metric %in% c('accuracy','roc_auc')) %>%
    pivot_wider(names_from = .metric, values_from=.estimate)
```

.estimator	part	accuracy	roc_auc
<chr>	<chr>	<dbl>	<dbl>
binary	training	0.9841427	0.9997585
binary	testing	0.9704081	0.8894098

2 rows

```
# -- ROC Charts
scored_train %>%
  mutate(model = "train") %>%
  bind_rows(scored_test %>%
              mutate(model="test")) %>%
  group_by(model) %>%
  roc_curve(churn, .pred_1) %>%
  autoplot()
```




```
scored_train %>%
  precision(churn, .pred_class) %>%
  mutate(part="training") %>%
  bind_rows(
  scored_test %>%
  precision(churn,.pred_class) %>%
    mutate(part="testing"))
```

.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
precision	binary	0.9922354	training
precision	binary	0.9232506	testing

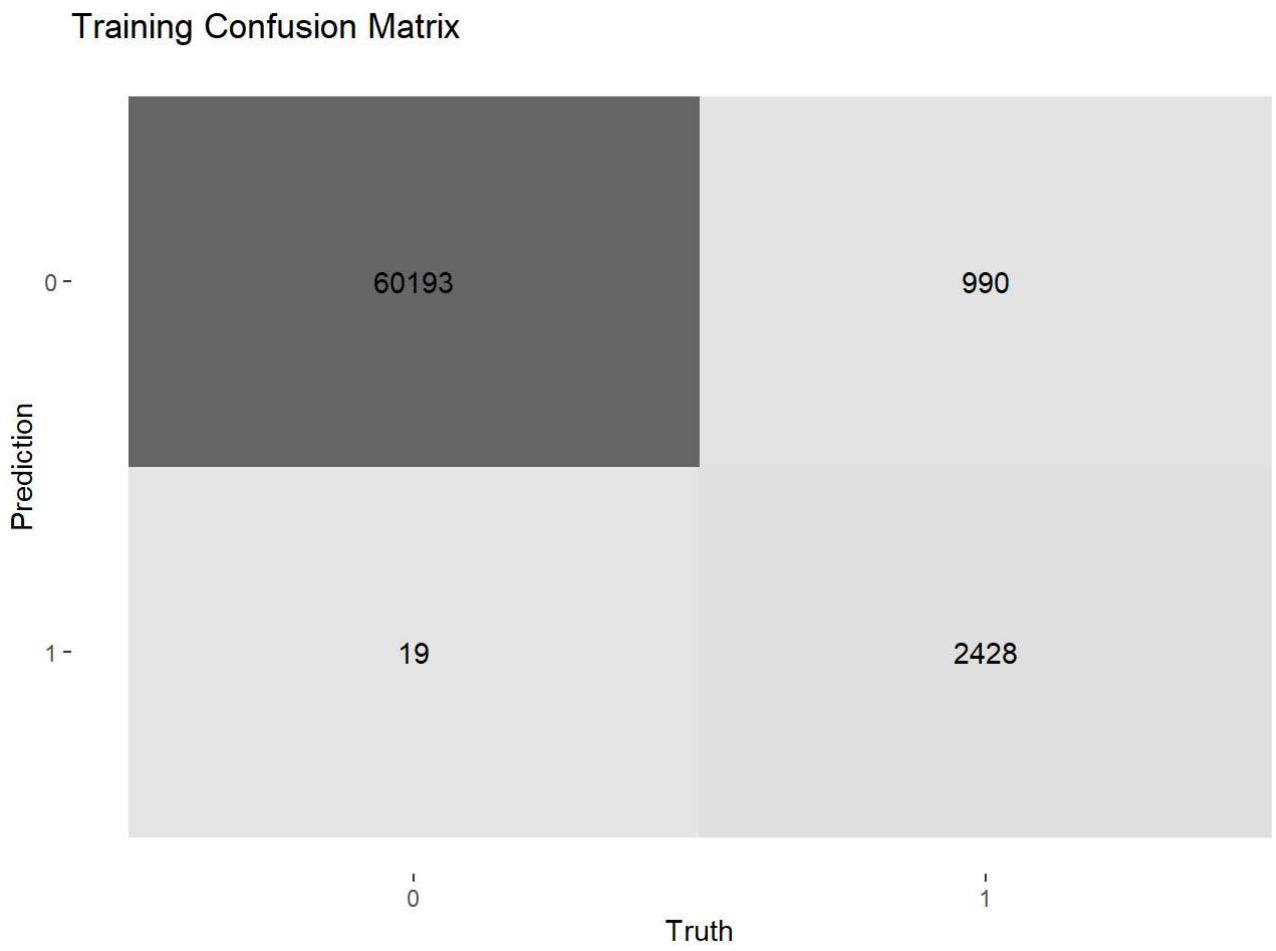
2 rows

```
scored_train %>%
  recall(churn, .pred_class) %>%
  mutate(part="training") %>%
  bind_rows(
  scored_test %>%
  recall(churn,.pred_class) %>%
    mutate(part="testing")
  )
```

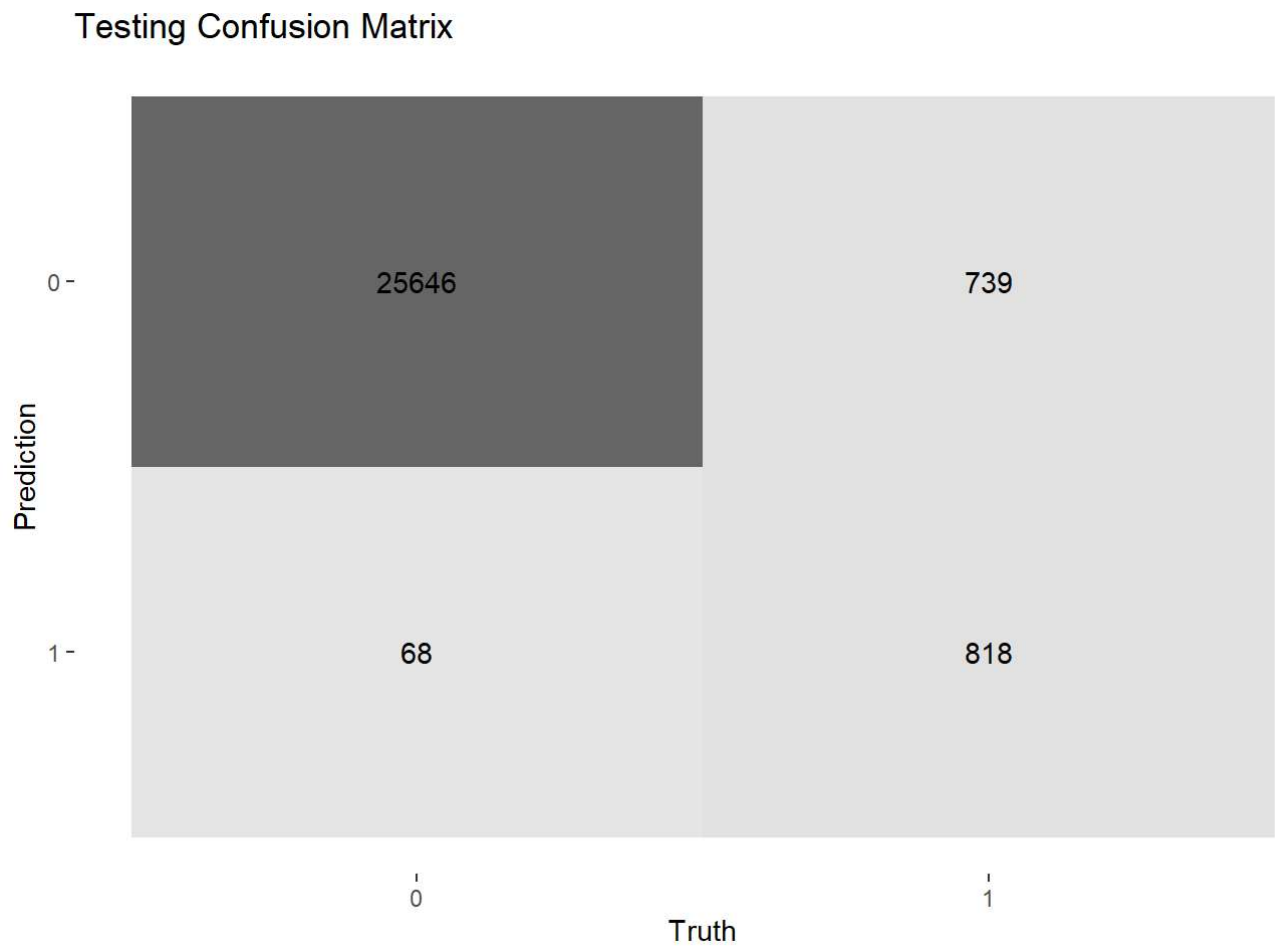
.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
recall	binary	0.7103569	training
recall	binary	0.5253693	testing

2 rows

```
scored_train %>%
  conf_mat(
  truth = churn,
  estimate = .pred_class,
  dnn = c("Prediction", "Truth")
) %>%
  autoplot(type = "heatmap") +
  labs(title="Training Confusion Matrix")
```




```
scored_test %>%
  conf_mat(
  truth = churn,
  estimate = .pred_class,
  dnn = c("Prediction", "Truth")
) %>%
  autoplot(type = "heatmap") +
  labs(title="Testing Confusion Matrix")
```



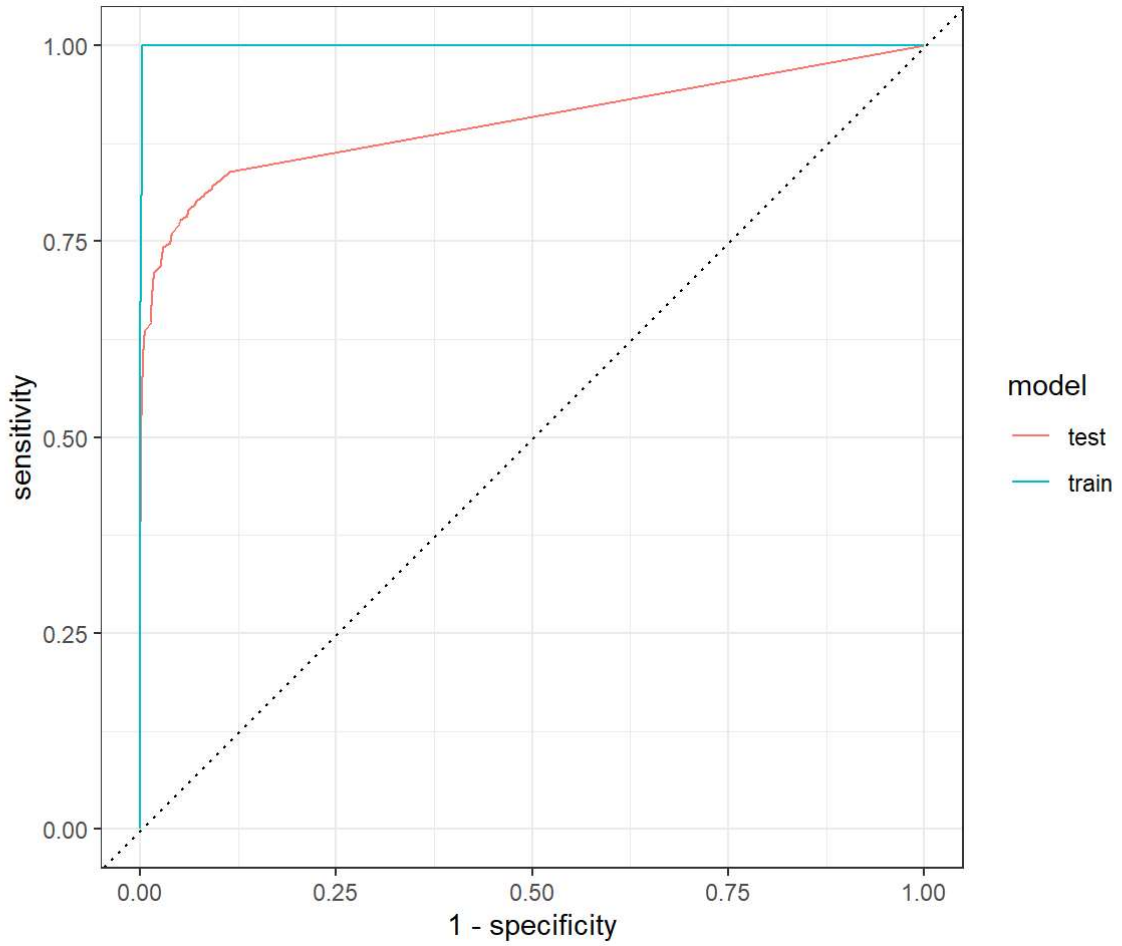
Evaluate (KNN = 10)

```
options(yardstick.event_first = FALSE)
# -- Metrics: Train and Test
scored_train_2 %>%
    metrics(churn, .pred_1, estimate = .pred_class) %>%
    mutate(part="training") %>%
    bind_rows(scored_test_2 %>%
                metrics(churn, .pred_1, estimate = .pred_class) %>%
                mutate(part="testing") ) %>%
    filter(.metric %in% c('accuracy','roc_auc')) %>%
    pivot_wider(names_from = .metric, values_from=.estimate)
```

.estimator	part	accuracy	roc_auc
<chr>	<chr>	<dbl>	<dbl>
binary	training	0.9809052	0.9994449
binary	testing	0.9703715	0.9013333

2 rows

```
# -- ROC Charts
scored_train_2 %>%
  mutate(model = "train") %>%
  bind_rows(scored_test_2 %>%
              mutate(model="test")) %>%
  group_by(model) %>%
  roc_curve(churn, .pred_1) %>%
  autoplot()
```




```
scored_train_2 %>%
  precision(churn, .pred_class) %>%
  mutate(part="training") %>%
  bind_rows(
  scored_test_2 %>%
  precision(churn,.pred_class) %>%
    mutate(part="testing"))
```

.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
precision	binary	0.9876051	training
precision	binary	0.9485030	testing

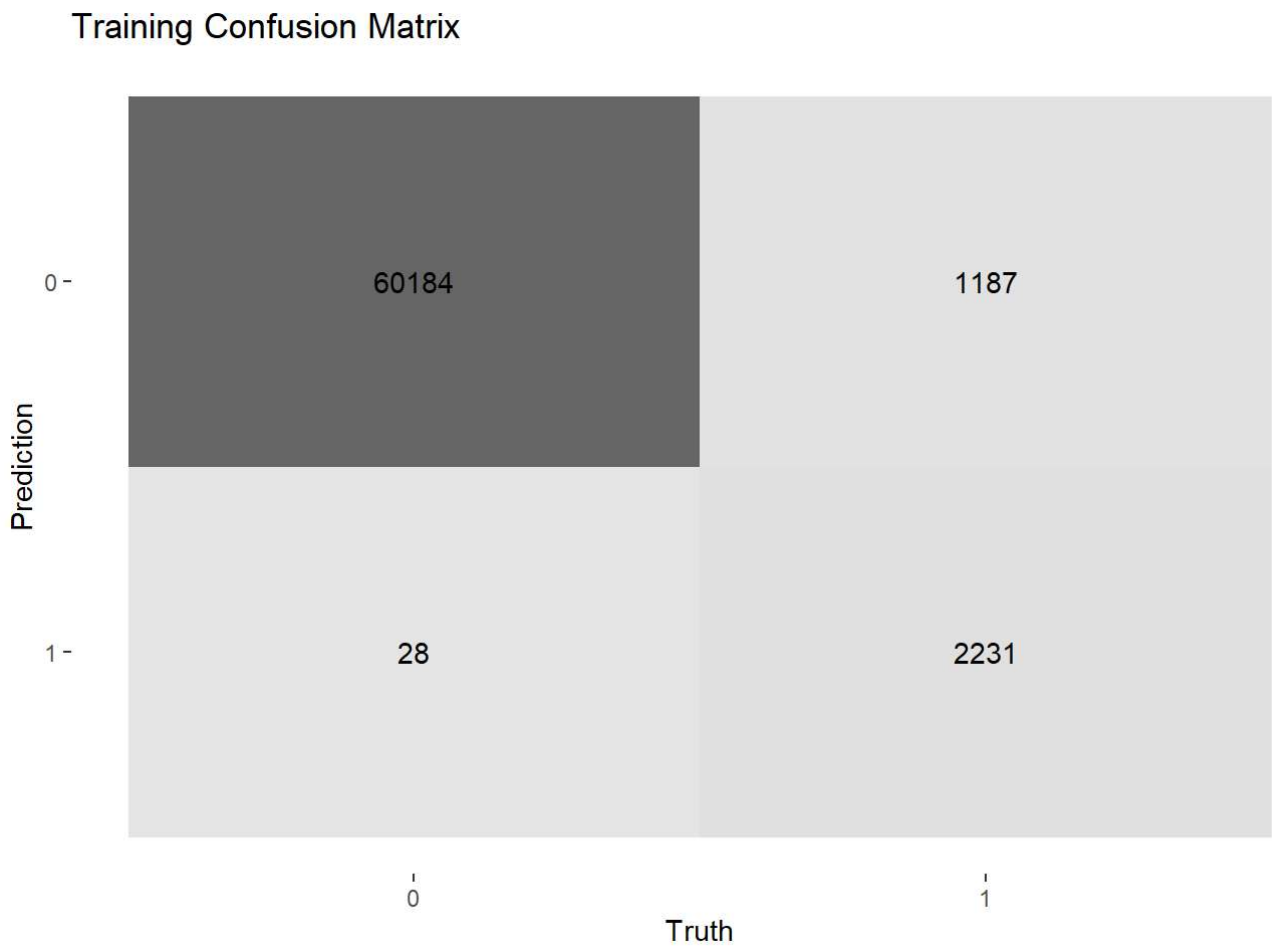
2 rows

```
scored_train_2 %>%
  recall(churn, .pred_class) %>%
  mutate(part="training") %>%
  bind_rows(
  scored_test_2 %>%
  recall(churn,.pred_class) %>%
    mutate(part="testing")
  )
```

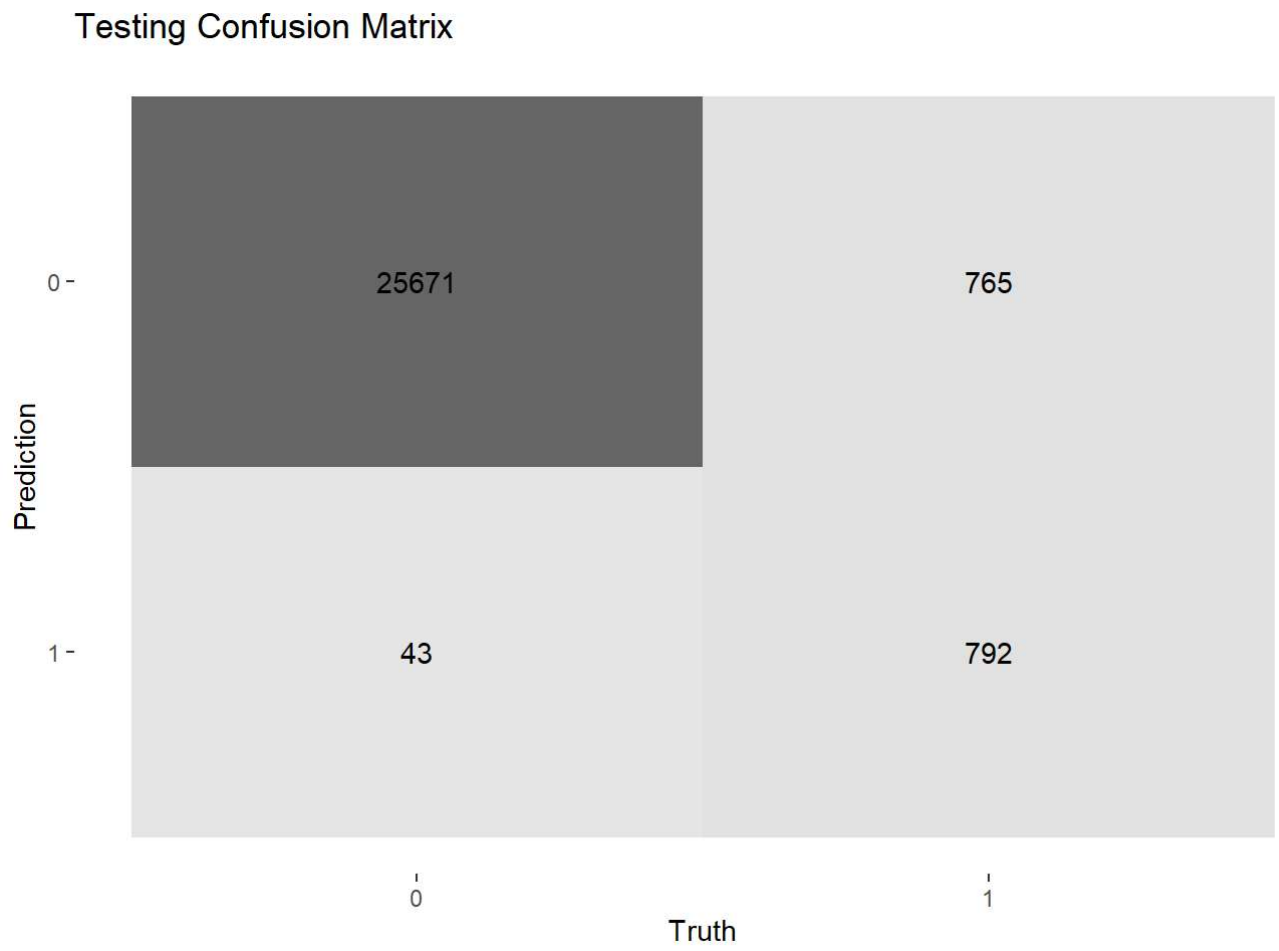
.metric	.estimator	.estimate	part
<chr>	<chr>	<dbl>	<chr>
recall	binary	0.6527209	training
recall	binary	0.5086705	testing

2 rows

```
scored_train_2 %>%
  conf_mat(
  truth = churn,
  estimate = .pred_class,
  dnn = c("Prediction", "Truth")
) %>%
  autoplot(type = "heatmap") +
  labs(title="Training Confusion Matrix")
```




```
scored_test_2 %>%
  conf_mat(
  truth = churn,
  estimate = .pred_class,
  dnn = c("Prediction", "Truth")
) %>%
  autoplot(type = "heatmap") +
  labs(title="Testing Confusion Matrix")
```



Kaggle

```
# -- score testing
kaggle_prediction <- predict(knn_workflow, churn_kaggle, type="class") %>%
    bind_cols(., churn_kaggle) %>%
    select(customer_id, churn=.pred_class)

kaggle_prediction %>%
  write_csv("challenge_1_kaggle.csv")
```