

EDA_R_Markdown

Chhavi Verma, Shruthi Rao

17-05-2023

R Markdown

This is an R Markdown document. The github link is
<https://github.com/ChhaviHere/CollegeProjects/>.

THIS IS CHECK * 5 (If this works then proceed peacefully)

```
telco_data <- read.csv(file.choose())
str(telco_data)

## 'data.frame':    7043 obs. of  21 variables:
## $ customerID      : chr  "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "77
95-CFOCW" ...
## $ gender          : chr  "Female" "Male" "Male" "Male" ...
## $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 ...
## $ Partner          : chr  "Yes" "No" "No" "No" ...
## $ Dependents       : chr  "No" "No" "No" "No" ...
## $ tenure           : int  1 34 2 45 2 8 22 10 28 62 ...
## $ PhoneService     : chr  "No" "Yes" "Yes" "No" ...
## $ MultipleLines    : chr  "No phone service" "No" "No" "No phone ser
vice" ...
## $ InternetService  : chr  "DSL" "DSL" "DSL" "DSL" ...
## $ OnlineSecurity   : chr  "No" "Yes" "Yes" "Yes" ...
## $ OnlineBackup      : chr  "Yes" "No" "Yes" "No" ...
## $ DeviceProtection: chr  "No" "Yes" "No" "Yes" ...
## $ TechSupport       : chr  "No" "No" "No" "Yes" ...
## $ StreamingTV       : chr  "No" "No" "No" "No" ...
## $ StreamingMovies   : chr  "No" "No" "No" "No" ...
## $ Contract          : chr  "Month-to-month" "One year" "Month-to-mont
h" "One year" ...
## $ PaperlessBilling: chr  "Yes" "No" "Yes" "No" ...
## $ PaymentMethod     : chr  "Electronic check" "Mailed check" "Mailed
check" "Bank transfer (automatic)" ...
## $ MonthlyCharges   : num  29.9 57 53.9 42.3 70.7 ...
## $ TotalCharges      : num  29.9 1889.5 108.2 1840.8 151.7 ...
## $ Churn             : chr  "No" "No" "Yes" "No" ...

head(telco_data)

##   customerID gender SeniorCitizen Partner Dependents tenure PhoneSer
vice
## 1 7590-VHVEG Female          0      Yes        No       1
## 2 7590-VHVEG Female          1      Yes        Yes      12
## 3 7590-VHVEG Female          0      Yes        Yes      12
## 4 7590-VHVEG Female          0      Yes        Yes      12
## 5 7590-VHVEG Female          0      Yes        Yes      12
```

## 2	5575-GNVDE	Male	0	No	No	34
Yes						
## 3	3668-QPYBK	Male	0	No	No	2
Yes						
## 4	7795-CFOCW	Male	0	No	No	45
No						
## 5	9237-HQITU	Female	0	No	No	2
Yes						
## 6	9305-CDSKC	Female	0	No	No	8
Yes						
## MultipleLines InternetService OnlineSecurity OnlineBackup DeviceProtection						
## 1	No phone service		DSL	No	Yes	
Yes	No		DSL	Yes	No	
## 2		No	DSL	Yes	No	
Yes		No	DSL	Yes	Yes	
## 3		No	DSL	Yes	Yes	
Yes	No		DSL	Yes	No	
## 4	No phone service		DSL	Yes	No	
Yes		No	Fiber optic	No	No	
## 5		No	Fiber optic	No	No	
Yes		Yes	Fiber optic	No	No	
## 6		Yes	Fiber optic	No	No	
## TechSupport StreamingTV StreamingMovies Contract PaperlessBilling						
## 1		No	No	No	Month-to-month	
Yes		No	No	No	One year	
## 2		No	No	No	Month-to-month	
No		No	No	No	Month-to-month	
## 3		No	No	No	One year	
Yes		No	No	No	Month-to-month	
## 4		Yes	No	No	Month-to-month	
No		No	No	No	Month-to-month	
## 5		Yes	No	No	Month-to-month	
Yes		No	No	No	Month-to-month	
## 6		Yes	Yes	Yes	Month-to-month	
Yes		Yes	Yes	Yes	Month-to-month	
## PaymentMethod MonthlyCharges TotalCharges Churn						
## 1	Electronic check	29.85	29.85	No		
## 2	Mailed check	56.95	1889.50	No		
## 3	Mailed check	53.85	108.15	Yes		
## 4	Bank transfer (automatic)	42.30	1840.75	No		
## 5	Electronic check	70.70	151.65	Yes		
## 6	Electronic check	99.65	820.50	Yes		

Installing and loading required packages

```
library(tidyverse)
```

```
## └─ Attaching core tidyverse packages ─────────────────── tidyve
rse 2.0.0 ─
## ✓ dplyr     1.1.2      ✓ readr     2.1.4
## ✓forcats   1.0.0      ✓ stringr   1.5.0
## ✓ ggplot2    3.4.2      ✓ tibble    3.2.1
## ✓ lubridate  1.9.2      ✓ tidyrr    1.3.0
## ✓ purrr     1.0.1
## └─ Conflicts ─────────────────── tidyverse_co
nflicts() ─
## ✘ dplyr::filter() masks stats::filter()
## ✘ dplyr::lag()    masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to fo
rce all conflicts to become errors

library(ggplot2)
library(dplyr)
library(rmarkdown)
library(markdown)
library(knitr)
library(readr)
library(viridis)

## Loading required package: viridisLite

library(plot3D)
library(plotly)

##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout

library(scales)

##
## Attaching package: 'scales'
##
## The following object is masked from 'package:viridis':
##
##     viridis_pal
##
```

```
## The following object is masked from 'package:purrr':  
##      discard  
##  
## The following object is masked from 'package:readr':  
##  
##      col_factor  
  
library(igraph)  
  
##  
## Attaching package: 'igraph'  
##  
## The following object is masked from 'package:plotly':  
##  
##      groups  
##  
## The following objects are masked from 'package:lubridate':  
##  
##      %--%, union  
##  
## The following objects are masked from 'package:dplyr':  
##  
##      as_data_frame, groups, union  
##  
## The following objects are masked from 'package:purrr':  
##  
##      compose, simplify  
##  
## The following object is masked from 'package:tidyr':  
##  
##      crossing  
##  
## The following object is masked from 'package:tibble':  
##  
##      as_data_frame  
##  
## The following objects are masked from 'package:stats':  
##  
##      decompose, spectrum  
##  
## The following object is masked from 'package:base':  
##  
##      union  
  
library(plot3D)  
library(corrplot)  
  
## corrplot 0.92 loaded
```

```
library(viridis)
library(ggplot2movies)
library(fields)

## Loading required package: spam
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
##
## Try help(fields) to get started.

library(dendextend)

##
## -----
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##     https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
##
##
## Attaching package: 'dendextend'
##
## The following object is masked from 'package:stats':
##
##     cutree

library(rgl)
library(RColorBrewer)
```

```
library(graphics)
library(scales)

Count the number of churned and non-churned customers:
telco_data %>% group_by(Churn) %>% summarise(count = n())

## # A tibble: 2 × 2
##   Churn count
##   <chr> <int>
## 1 No     5174
## 2 Yes    1869
```

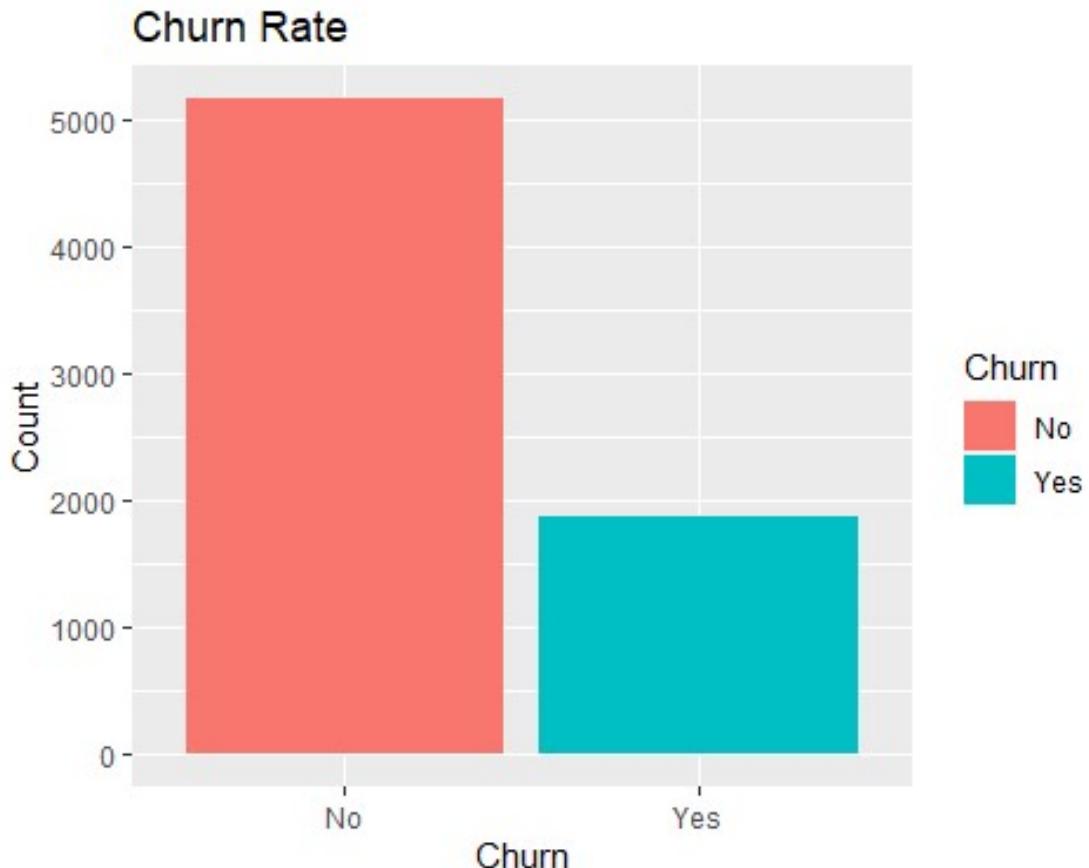
Calculate the churn rate:

```
churn_rate <- telco_data %>%
  filter(Churn == "Yes") %>%
  nrow() / nrow(telco_data)
churn_rate

## [1] 0.2653699
```

Creating a bar plot of churn rate

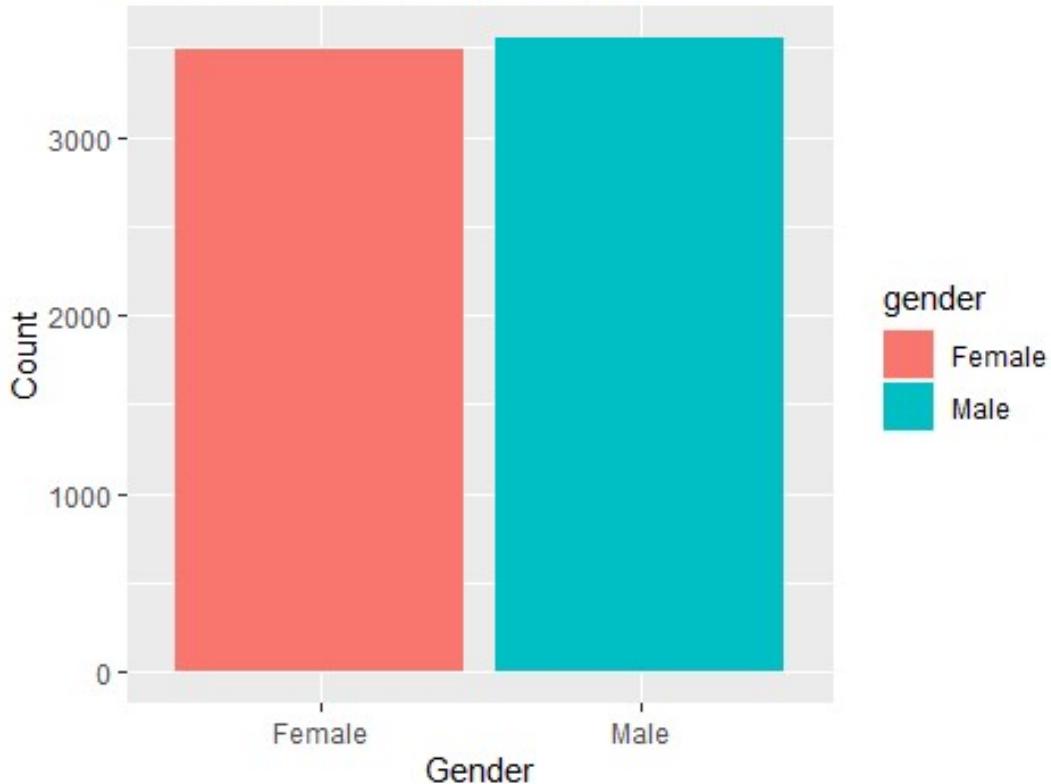
```
telco_data %>%
  ggplot(aes(x = Churn, fill = Churn)) +
  geom_bar() +
  labs(title = "Churn Rate",
       x = "Churn",
       y = "Count")
```



Analyze the distribution of customer genders:

```
telco_data %>%
  ggplot(aes(x = gender, fill = gender)) +
  geom_bar() +
  labs(title = "Customer Gender Distribution",
       x = "Gender",
       y = "Count")
```

Customer Gender Distribution



```
### Calculate the average monthly charges for churned and non-churned customers:
```

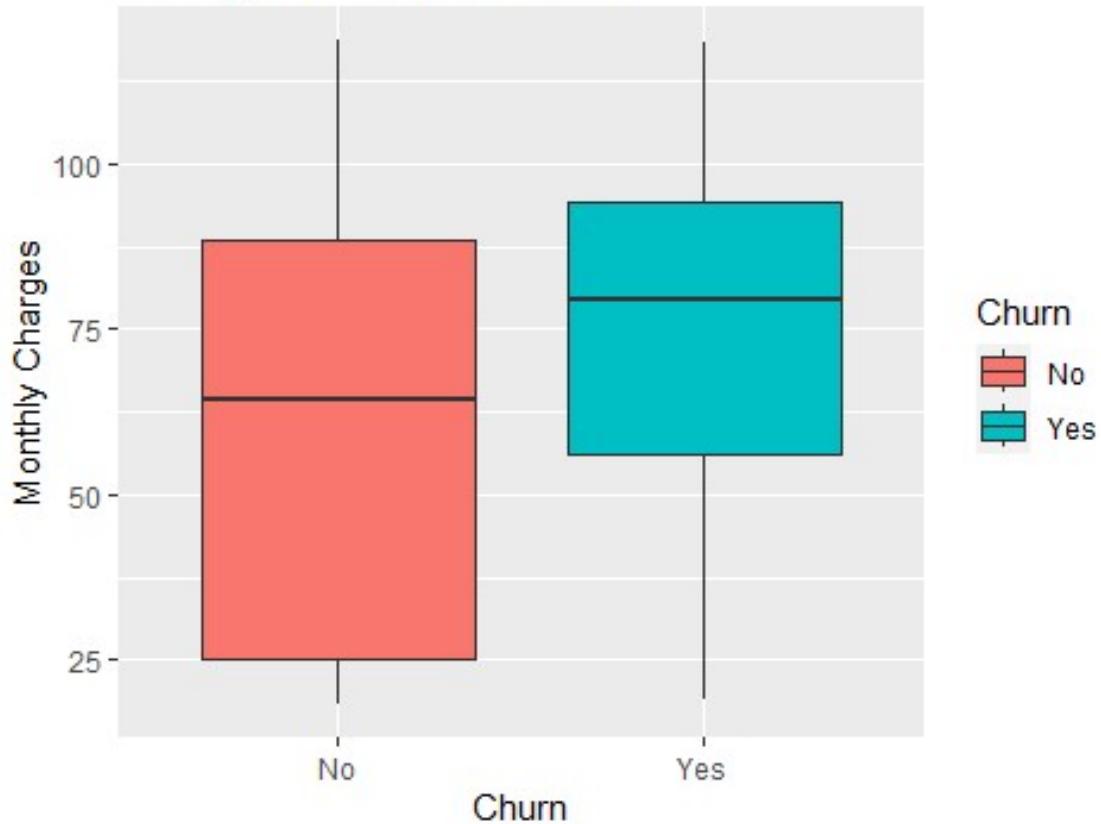
```
telco_data %>%
  group_by(Churn) %>%
  summarise(avg_monthly_charges = mean(MonthlyCharges))

## # A tibble: 2 × 2
##   Churn avg_monthly_charges
##   <chr>          <dbl>
## 1 No            61.3
## 2 Yes           74.4
```

Visualize the average monthly charges for churned and non-churned customers:

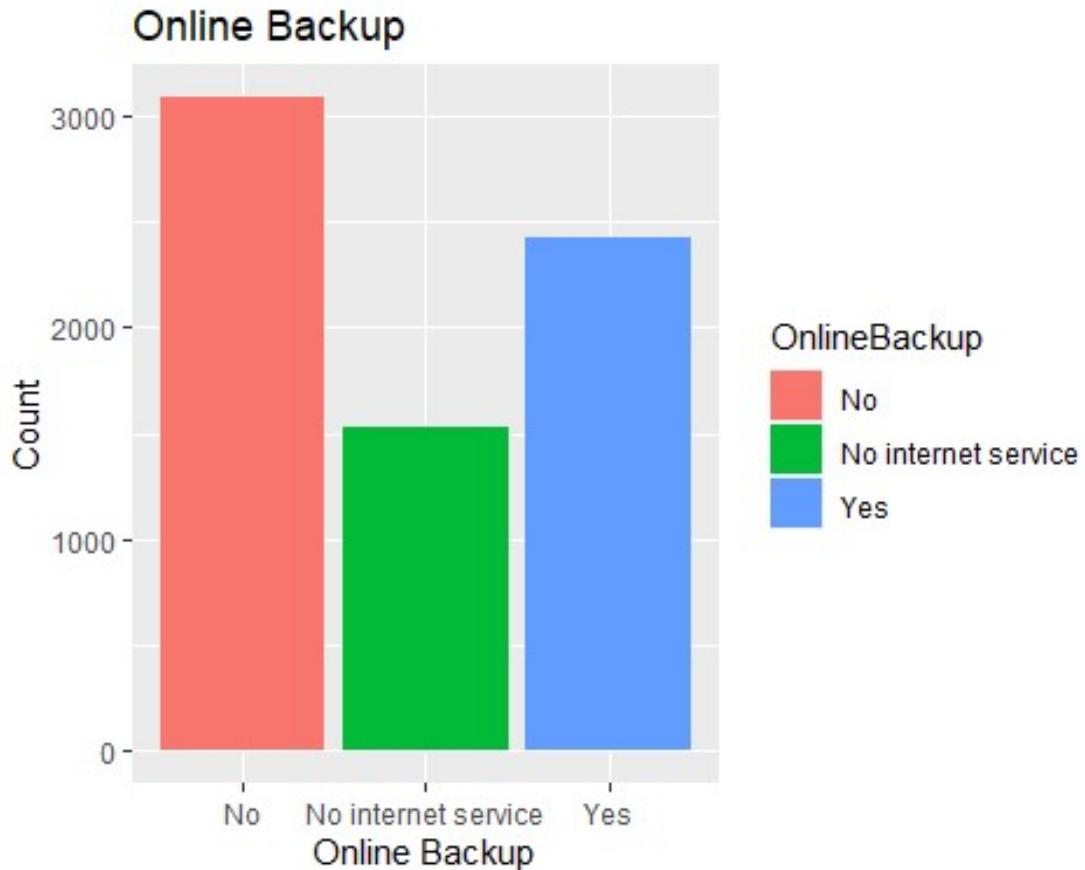
```
telco_data %>%
  ggplot(aes(x = Churn, y = MonthlyCharges, fill = Churn)) +
  geom_boxplot() +
  labs(title = "Average Monthly Charges",
       x = "Churn",
       y = "Monthly Charges")
```

Average Monthly Charges



Visualize the distribution of customers with and without online backup:

```
telco_data %>%
  ggplot(aes(x = OnlineBackup, fill = OnlineBackup)) +
  geom_bar() +
  labs(title = "Online Backup",
       x = "Online Backup",
       y = "Count")
```



Visualize the distribution of contract types:

```
telco_data %>%
  ggplot(aes(x = Contract, fill = Contract)) +
  geom_bar() +
  labs(title = "Contract Type Distribution",
       x = "Contract Type",
       y = "Count")
```



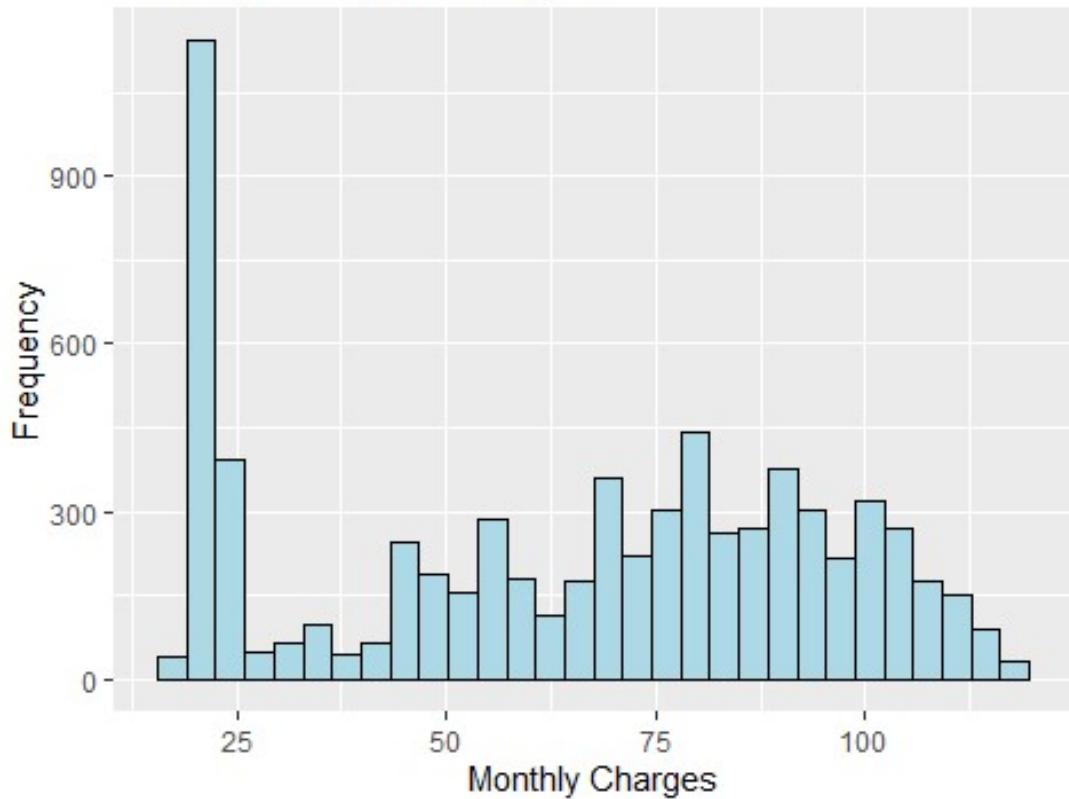
```
colnames(telco_data)
```

```
## [1] "customerID"      "gender"          "SeniorCitizen"   "Partn
er"
## [5] "Dependents"       "tenure"          "PhoneService"    "Multi
pleLines"
## [9] "InternetService"  "OnlineSecurity"   "OnlineBackup"    "Devic
eProtection"
## [13] "TechSupport"      "StreamingTV"     "StreamingMovies" "Contr
act"
## [17] "PaperlessBilling" "PaymentMethod"   "MonthlyCharges"  "Total
Charges"
## [21] "Churn"
```

Basic Histogram

```
ggplot(telco_data, aes(x = MonthlyCharges)) +
  geom_histogram(fill = "lightblue", color = "black") +
  labs(title = "Monthly Charges Histogram", x = "Monthly Charges", y =
"Frequency")
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

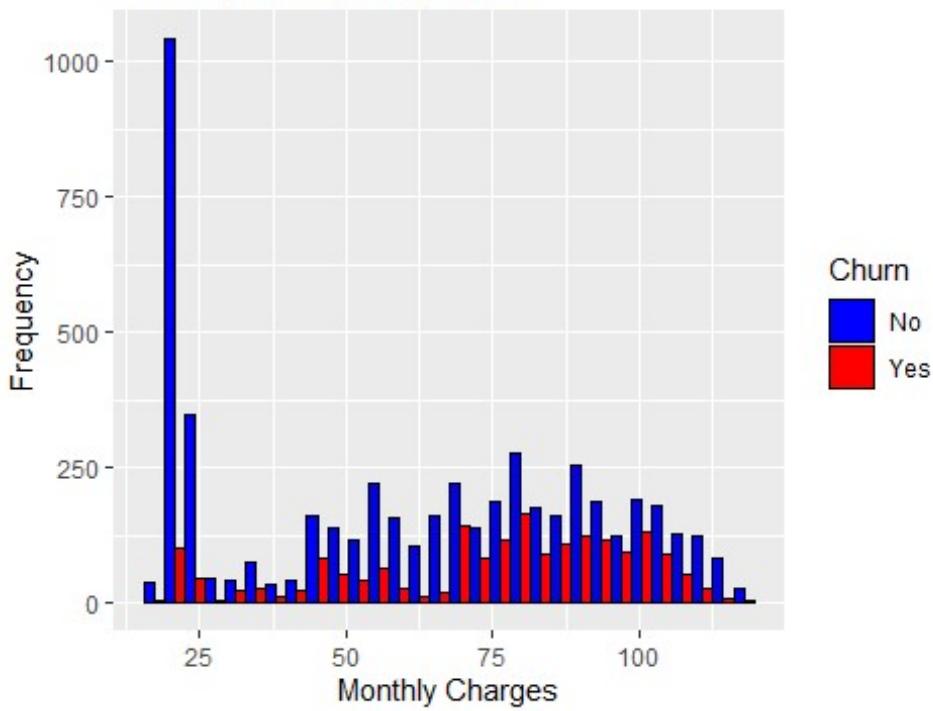
Monthly Charges Histogram



Multiple Histograms from Grouped Data

```
ggplot(telco_data, aes(x = MonthlyCharges, fill = Churn)) +  
  geom_histogram(position = "dodge", color = "black") +  
  labs(title = "Monthly Charges by Churn", x = "Monthly Charges", y = "  
Frequency") +  
  scale_fill_manual(values = c("blue", "red"))  
## `stat_bin()` using `bins = 30` . Pick better value with `binwidth` .
```

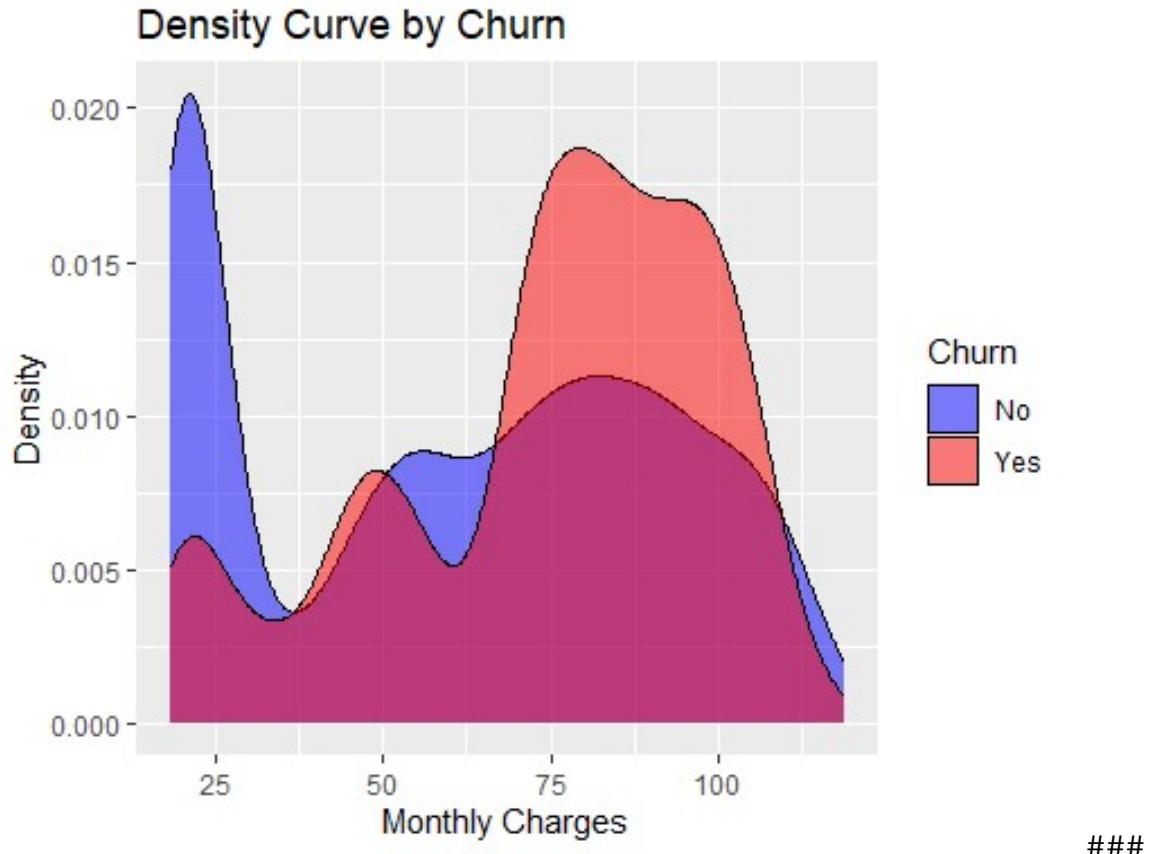
Monthly Charges by Churn



#

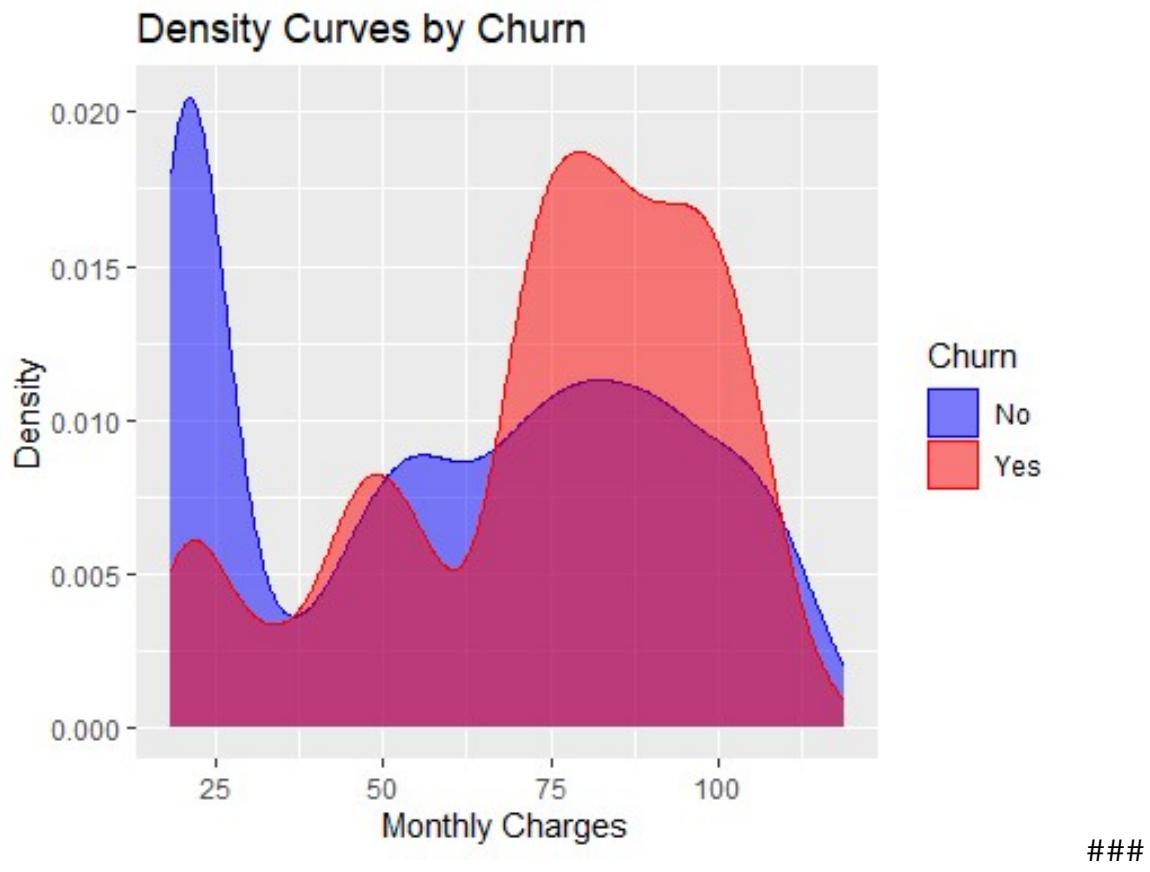
Density Curve

```
ggplot(telco_data, aes(x = MonthlyCharges, fill = Churn)) +  
  geom_density(alpha = 0.5) +  
  labs(title = "Density Curve by Churn", x = "Monthly Charges", y = "De  
nsity") +  
  scale_fill_manual(values = c("blue", "red"))
```



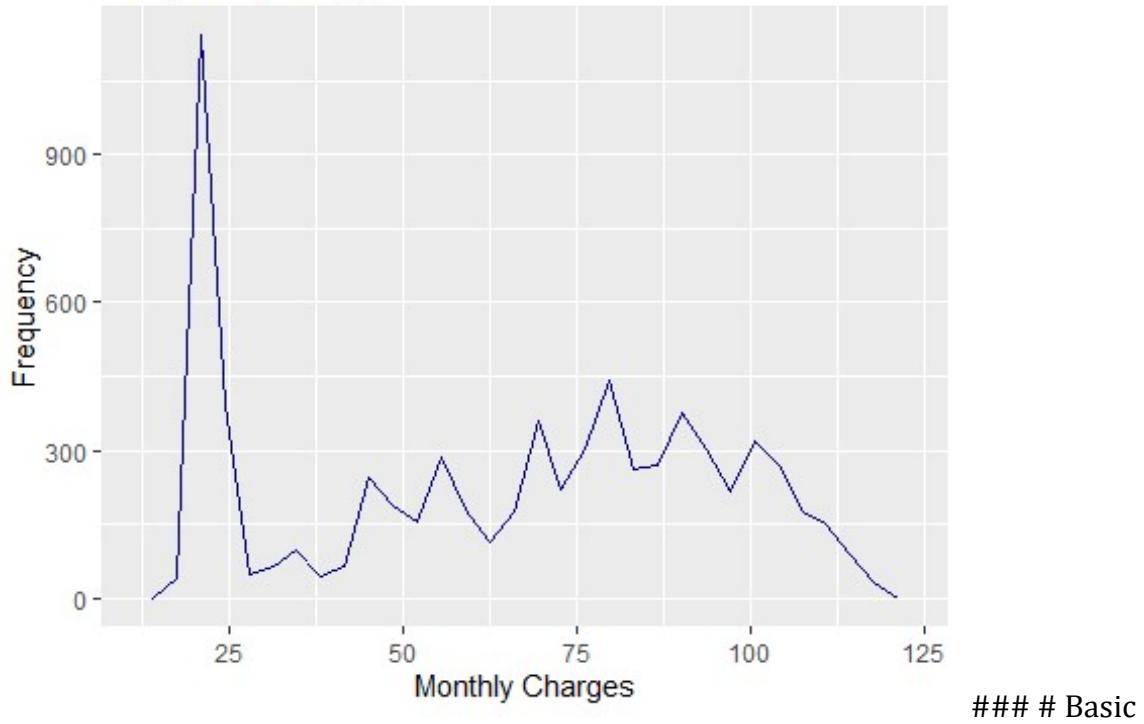
Multiple Density Curves from Grouped Data

```
ggplot(telco_data, aes(x = MonthlyCharges, color = Churn, fill = Churn))  
+  
  geom_density(alpha = 0.5) +  
  labs(title = "Density Curves by Churn", x = "Monthly Charges", y = "Density") +  
  scale_fill_manual(values = c("blue", "red")) +  
  scale_color_manual(values = c("blue", "red"))
```



```
# Frequency Polygon  
###  
# Frequency Polygon  
ggplot(telco_data, aes(x = MonthlyCharges)) +  
  geom_freqpoly(color = "darkblue") +  
  labs(title = "Frequency Polygon", x = "Monthly Charges", y = "Frequency")  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

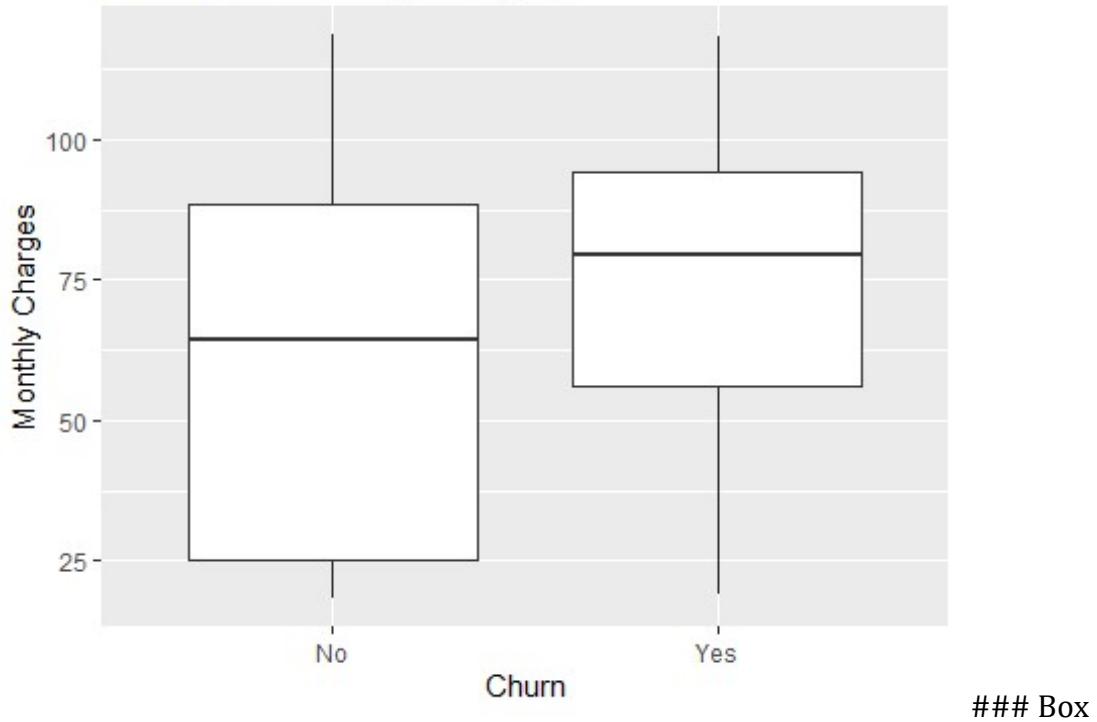
Frequency Polygon



Box Plot

```
ggplot(telco_data, aes(x = Churn, y = MonthlyCharges)) +  
  geom_boxplot() +  
  labs(title = "Box Plot of Monthly Charges", x = "Churn", y = "Monthly  
  Charges")
```

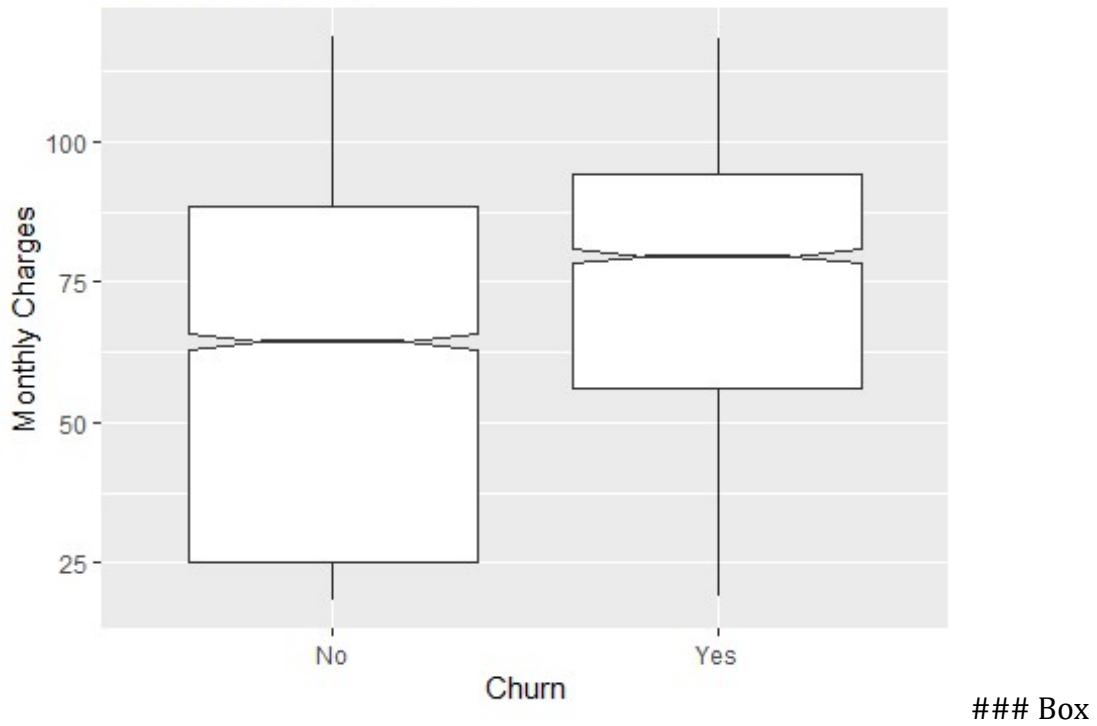
Box Plot of Monthly Charges



Plot with Notches

```
ggplot(telco_data, aes(x = Churn, y = MonthlyCharges)) +  
  geom_boxplot(notch = TRUE) +  
  labs(title = "Box Plot with Notches", x = "Churn", y = "Monthly Charges")
```

Box Plot with Notches

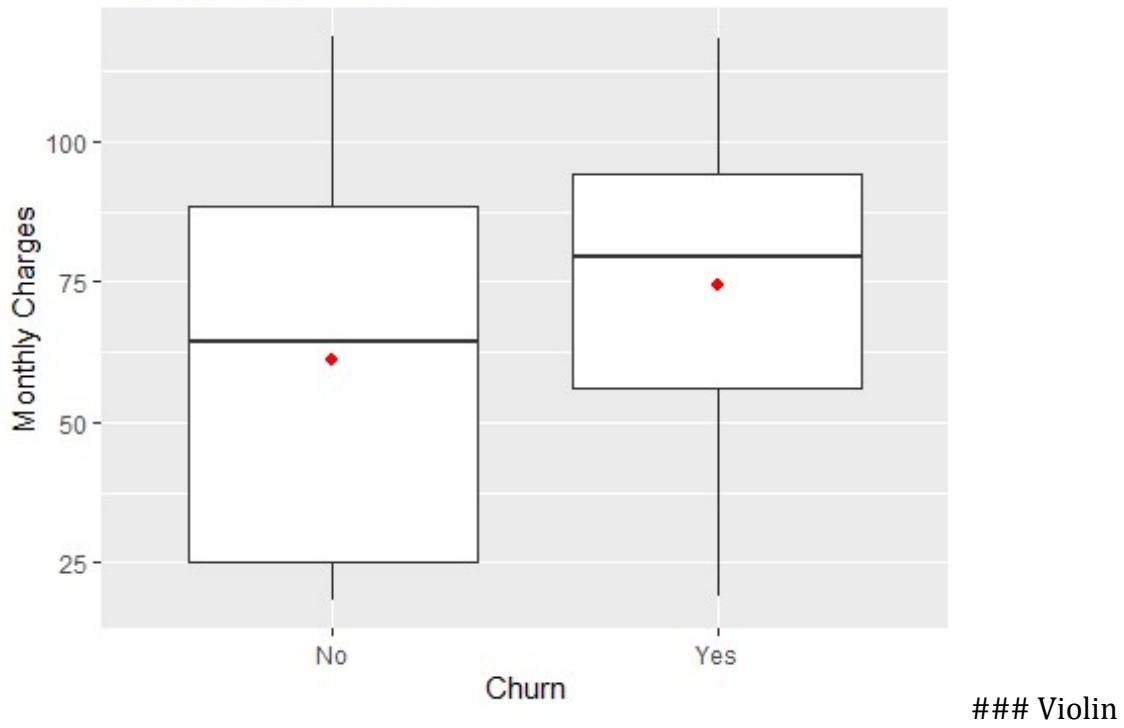


Box

Plot with Means

```
ggplot(telco_data, aes(x = Churn, y = MonthlyCharges)) +  
  geom_boxplot() +  
  stat_summary(fun = mean, geom = "point", shape = 20, size = 3, color  
= "red") +  
  labs(title = "Box Plot with Means", x = "Churn", y = "Monthly Charges  
)
```

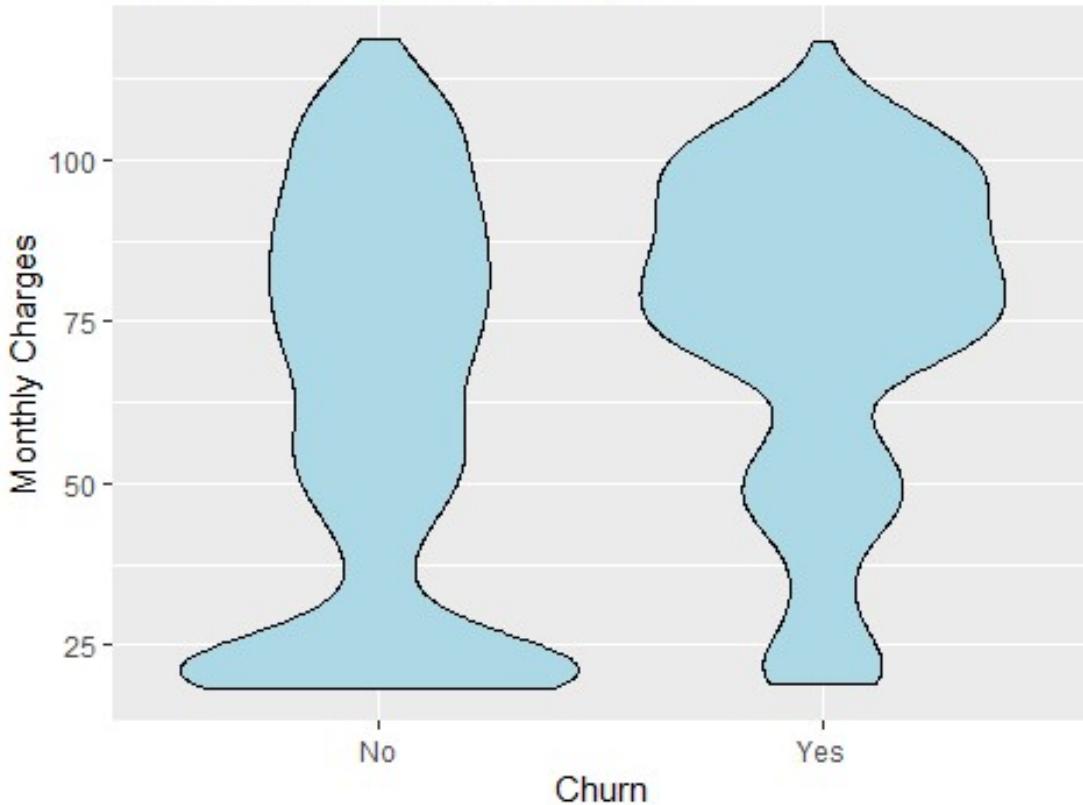
Box Plot with Means



Plot

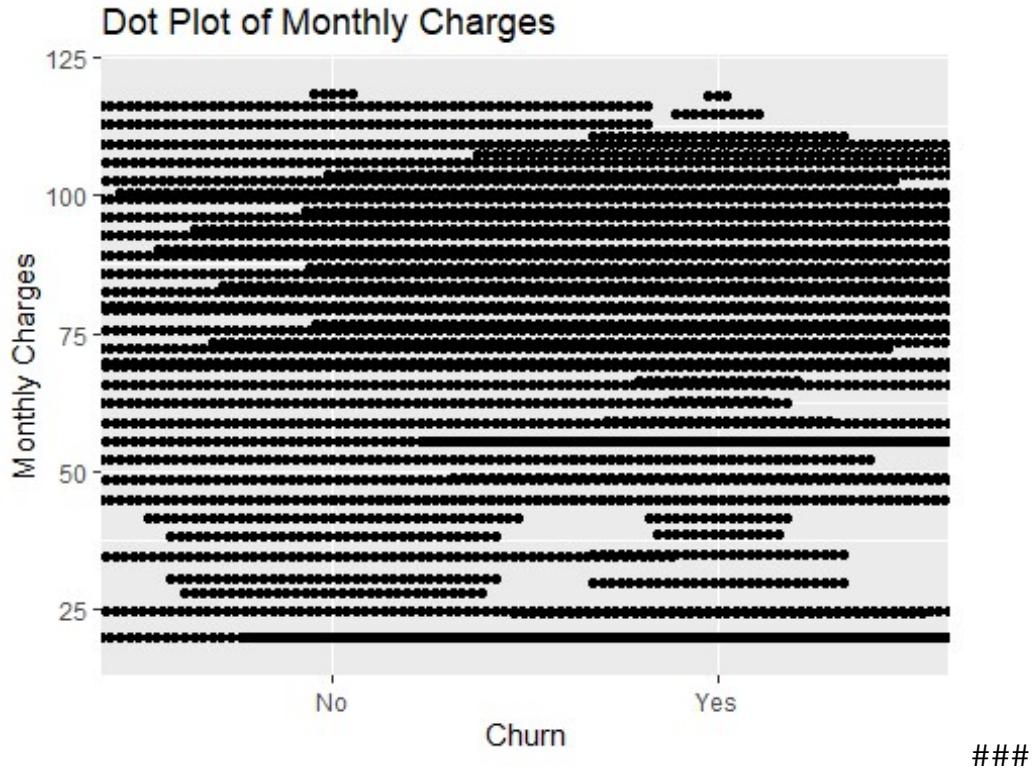
```
ggplot(telco_data, aes(x = Churn, y = MonthlyCharges)) +  
  geom_violin(fill = "lightblue", color = "black") +  
  labs(title = "Violin Plot of Monthly Charges", x = "Churn", y = "Mont  
hly Charges")
```

Violin Plot of Monthly Charges



Dot Plot

```
ggplot(telco_data, aes(x = Churn, y = MonthlyCharges)) +  
  geom_dotplot(binaxis = "y", stackdir = "center", dotsize = 0.5) +  
  labs(title = "Dot Plot of Monthly Charges", x = "Churn", y = "Monthly  
  Charges")  
  
## Bin width defaults to 1/30 of the range of the data. Pick better val  
ue with  
## `binwidth`.
```

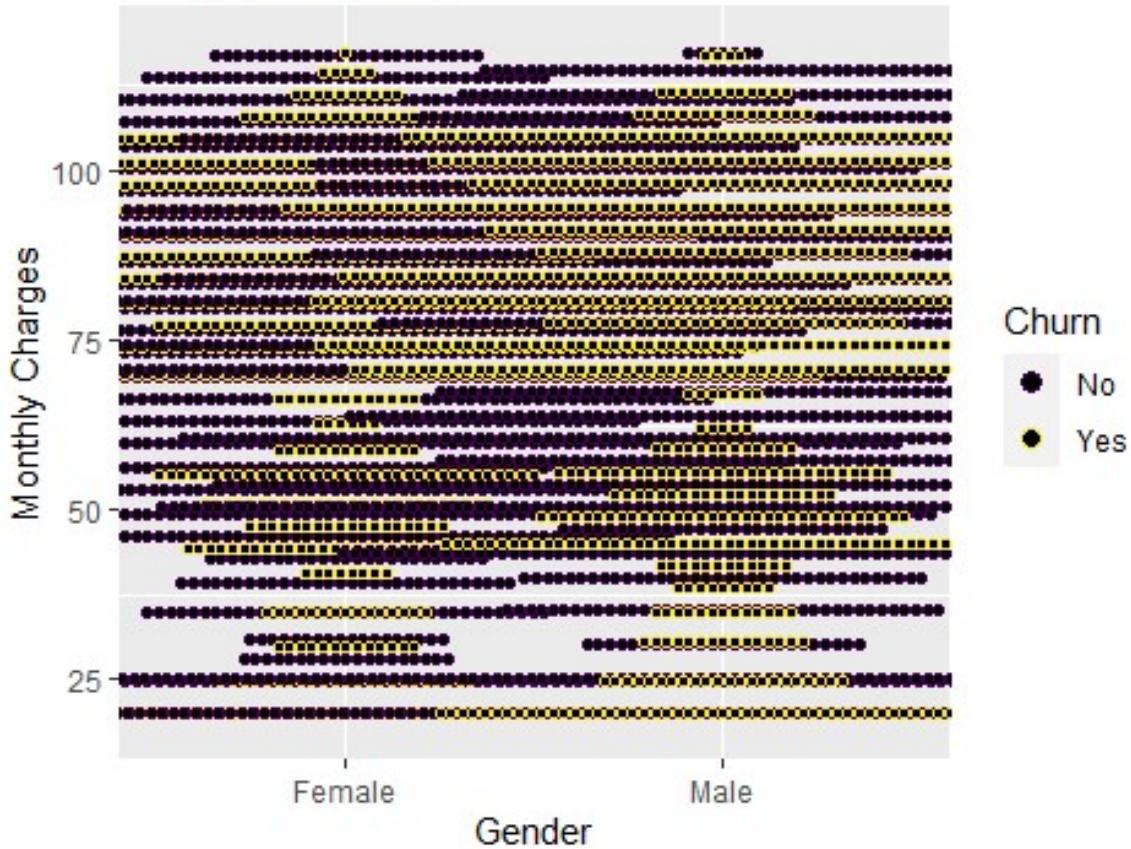


Multiple dot plot from grouped data

```
ggplot(telco_data, aes(x = gender, y = MonthlyCharges, color = Churn))
+
  geom_dotplot(binaxis = "y", stackdir = "center", dotsize = 0.5) +
  scale_color_manual(values = viridis(2)) +
  labs(title = "Multiple Dot Plot",
       x = "Gender",
       y = "Monthly Charges")

## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```

Multiple Dot Plot



Density plot of 2D data

```
ggplot(telco_data, aes(x = MonthlyCharges, y = tenure)) +  
  geom_density_2d(color = "green", alpha = 0.6) +  
  scale_fill_viridis() +  
  labs(title = "Density Plot",  
       x = "Monthly Charges",  
       y = "Tenure")
```

Density Plot



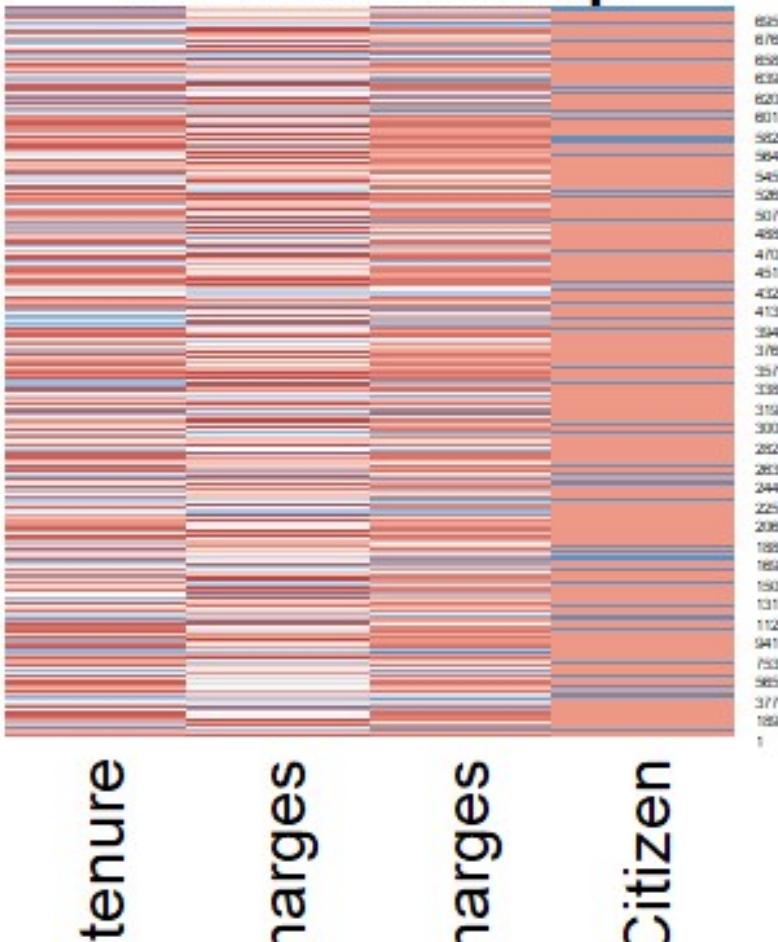
Correlation matrix

```
# Select the columns for the heatmap
heatmap_data <- telco_data[, c("tenure", "MonthlyCharges", "TotalCharges", "SeniorCitizen")]

# Convert the selected columns to a numeric matrix
heatmap_matrix <- as.matrix(heatmap_data)

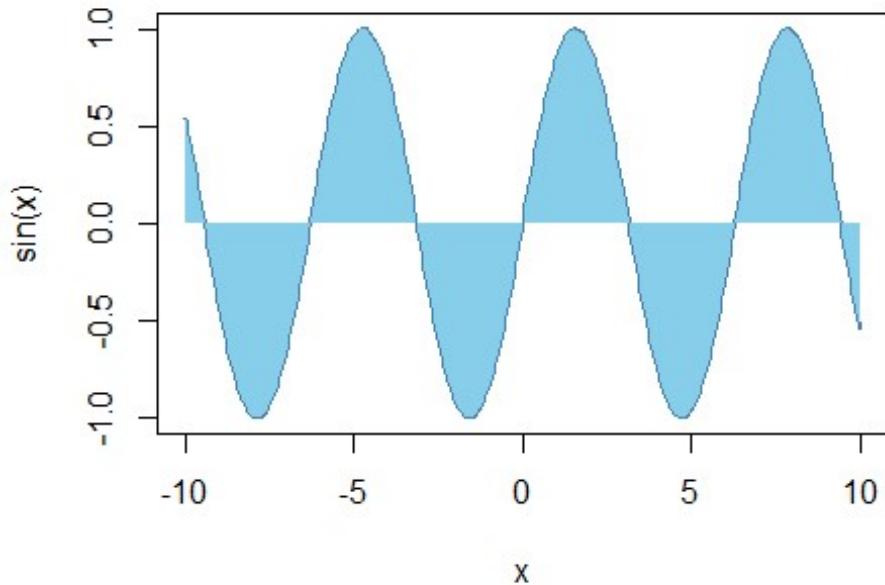
# Create the heatmap
heatmap(heatmap_matrix, scale = "column", Colv = NA, Rowv = NA,
        col = colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77ADD", "#4477AA"))(200),
        labCol = colnames(heatmap_matrix), labRow = rownames(heatmap_matrix),
        main = "Labeled Heatmap")
```

Labeled Heatmap

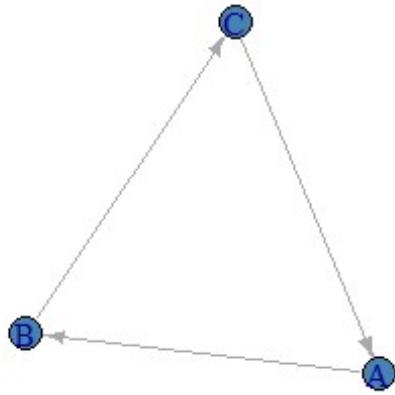


```
x <- seq(-10, 10, by = 0.1)
y <- sin(x)
plot(x, y, type = "l", col = "steelblue", lwd = 2, xlab = "x", ylab = "sin(x)", main = "Plot of sin(x)")
x <- seq(-10, 10, by = 0.1)
y <- sin(x)
plot(x, y, type = "l", col = "steelblue", lwd = 2, xlab = "x", ylab = "sin(x)", main = "Plot of sin(x)")
polygon(c(-10, x, 10), c(0, y, 0), col = "skyblue", border = NA)
```

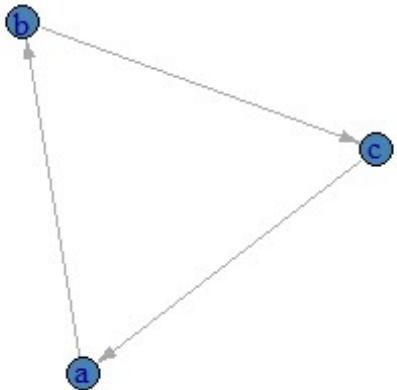
Plot of $\sin(x)$



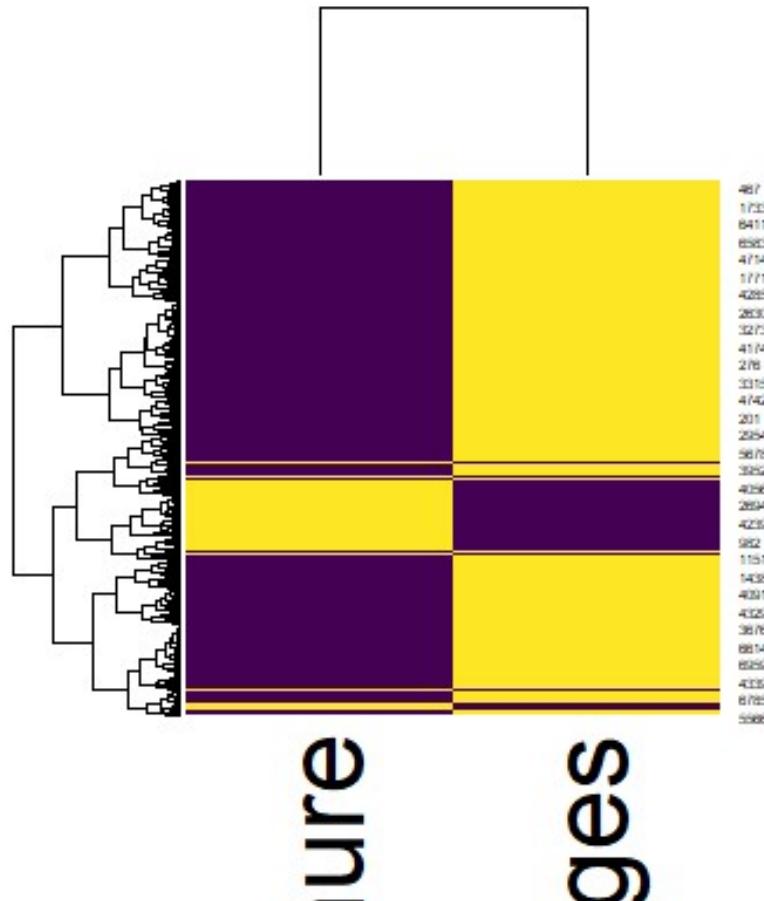
```
edges <- data.frame(from = c("A", "B", "C"), to = c("B", "C", "A"))
graph <- graph_from_data_frame(edges)
plot(graph, edge.arrow.size = 0.5, edge.color = "darkgray", vertex.col
r = "steelblue", vertex.size = 20)
```



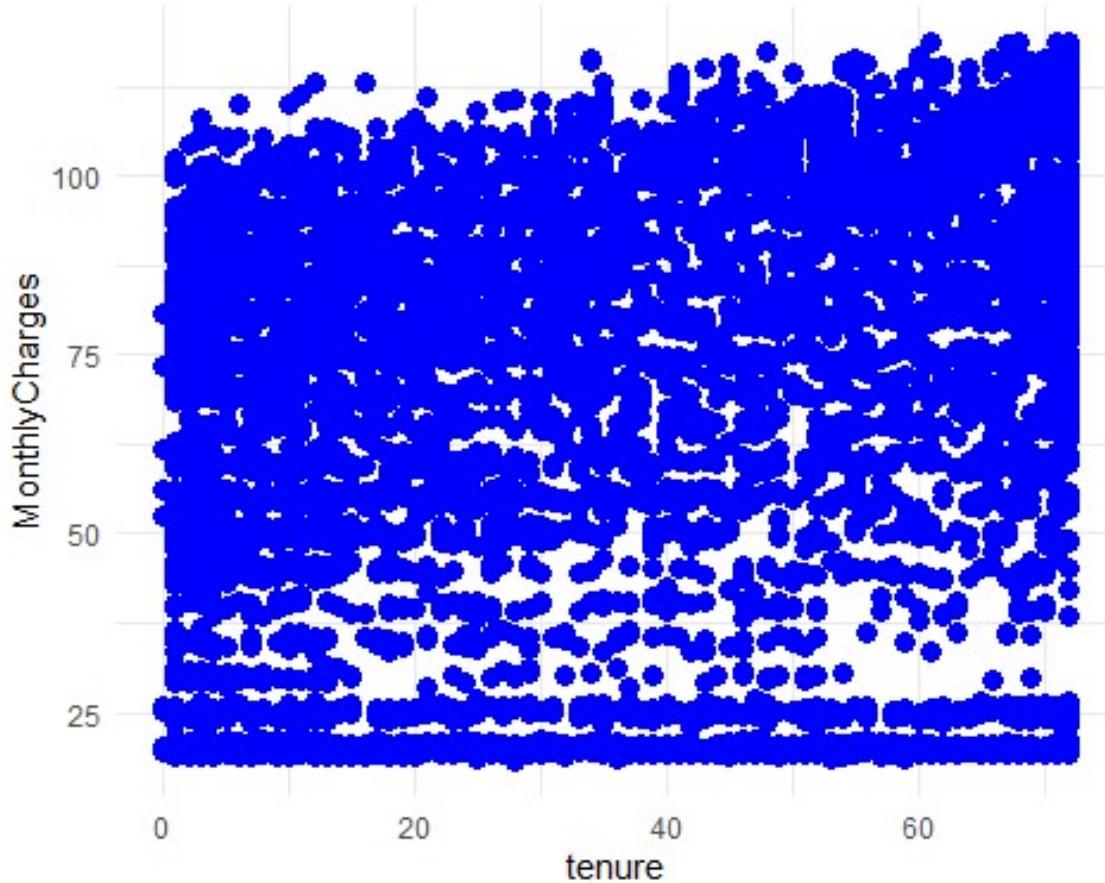
```
edges <- data.frame(from = c("A", "B", "C"), to = c("B", "C", "A"))
graph <- graph_from_data_frame(edges)
plot(graph, edge.arrow.size = 0.5, edge.color = "darkgray", vertex.colour = "steelblue", vertex.size = 20, vertex.label = letters[1:3])
```



```
heatmap(data.matrix(telco_data[, c("tenure", "MonthlyCharges")]), col = viridis(100))
```



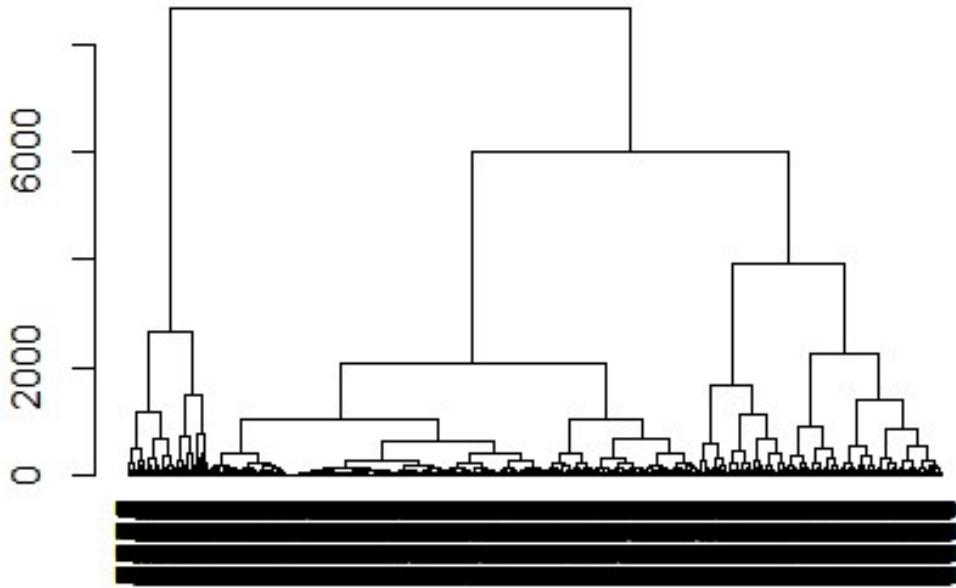
```
scatter_plot <- ggplot(telco_data, aes(x = tenure, y = MonthlyCharges,
z = TotalCharges)) +
  geom_point(color = "blue", size = 3) +
  theme_minimal()
scatter_plot
```



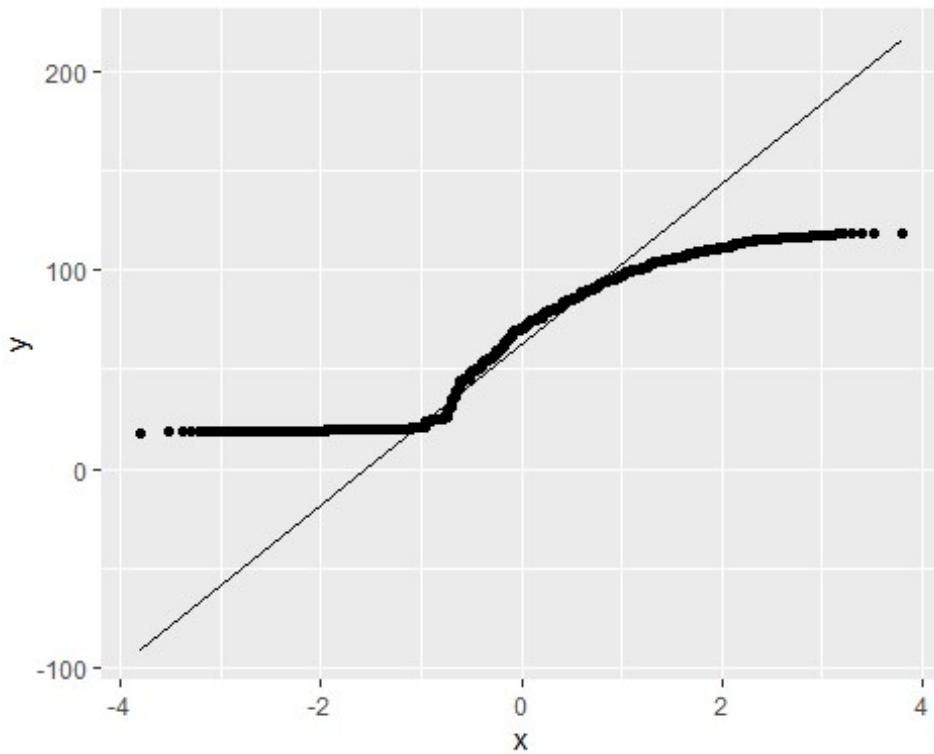
```
ggsave("scatter_plot.png", plot = scatter_plot)

## Saving 5 x 4 in image

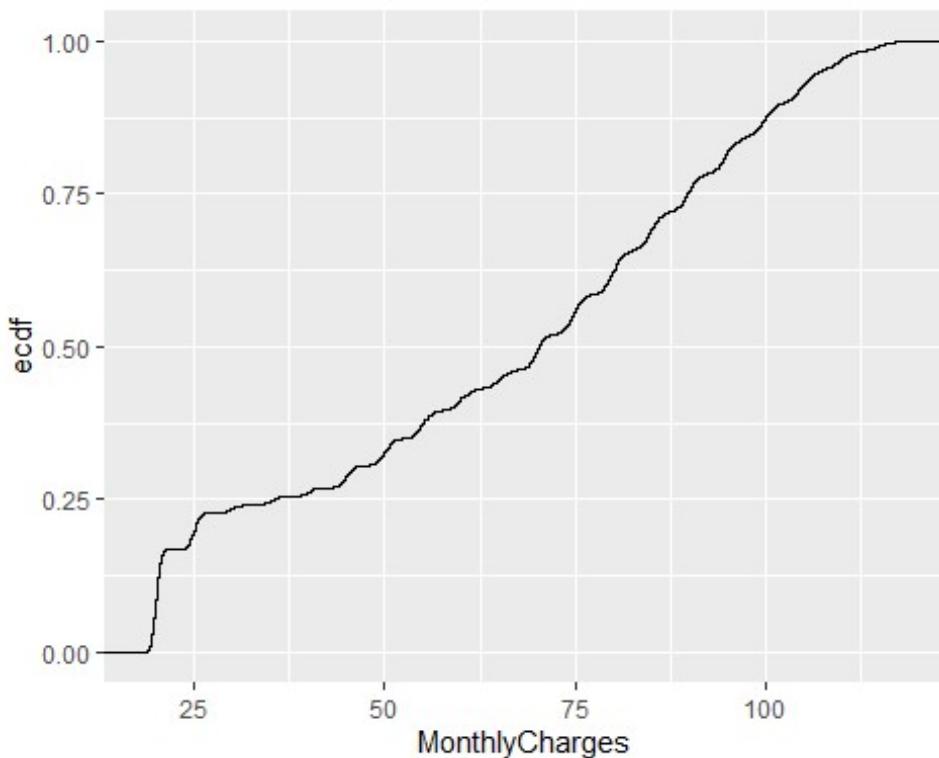
dendrogram <- telco_data[, c("tenure", "MonthlyCharges", "TotalCharges")]
dend <- as.dendrogram(hclust(dist(dendrogram)))
plot(dend)
```



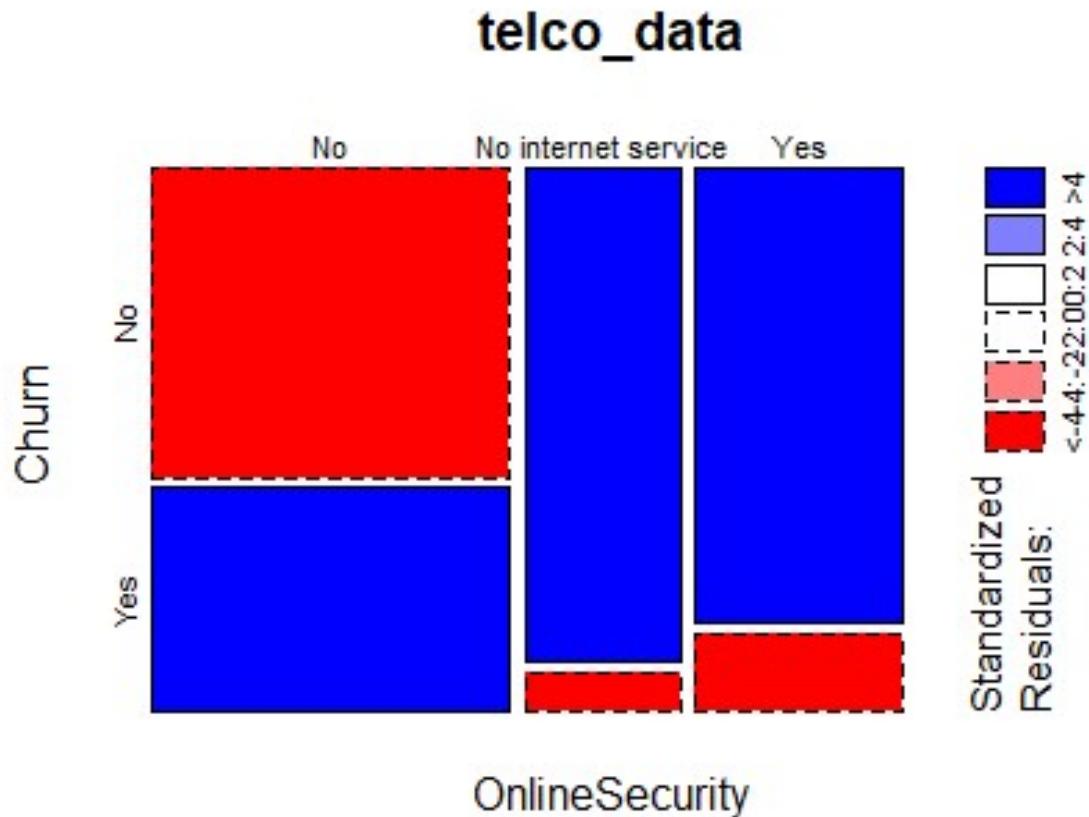
```
qq_plot <- ggplot(telco_data, aes(sample = MonthlyCharges)) +  
  stat_qq() +  
  stat_qq_line()  
qq_plot
```



```
ecdf_graph <- ggplot(telco_data, aes(x = MonthlyCharges)) +  
  stat_ecdf(geom = "step")  
ecdf_graph
```



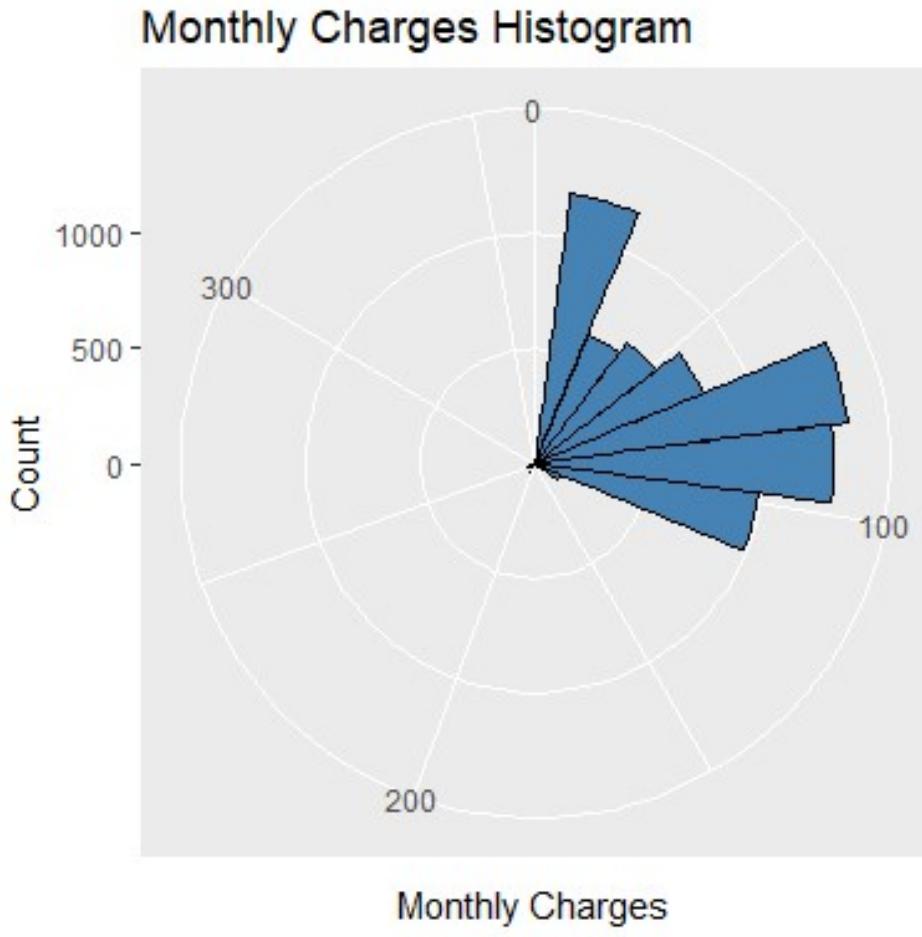
```
mosaic_plot <- mosaicplot(~OnlineSecurity + Churn, data = telco_data, shade = TRUE, color = TRUE)
```



```
ggplot(telco_data, aes(x = MonthlyCharges)) +
  geom_histogram(binwidth = 15, origin = -7.5, color = "black", fill = "steelblue") +
  labs(title = "Monthly Charges Histogram",
       x = "Monthly Charges",
       y = "Count") +
  coord_polar() +
  scale_x_continuous(limit = c(0, 360))

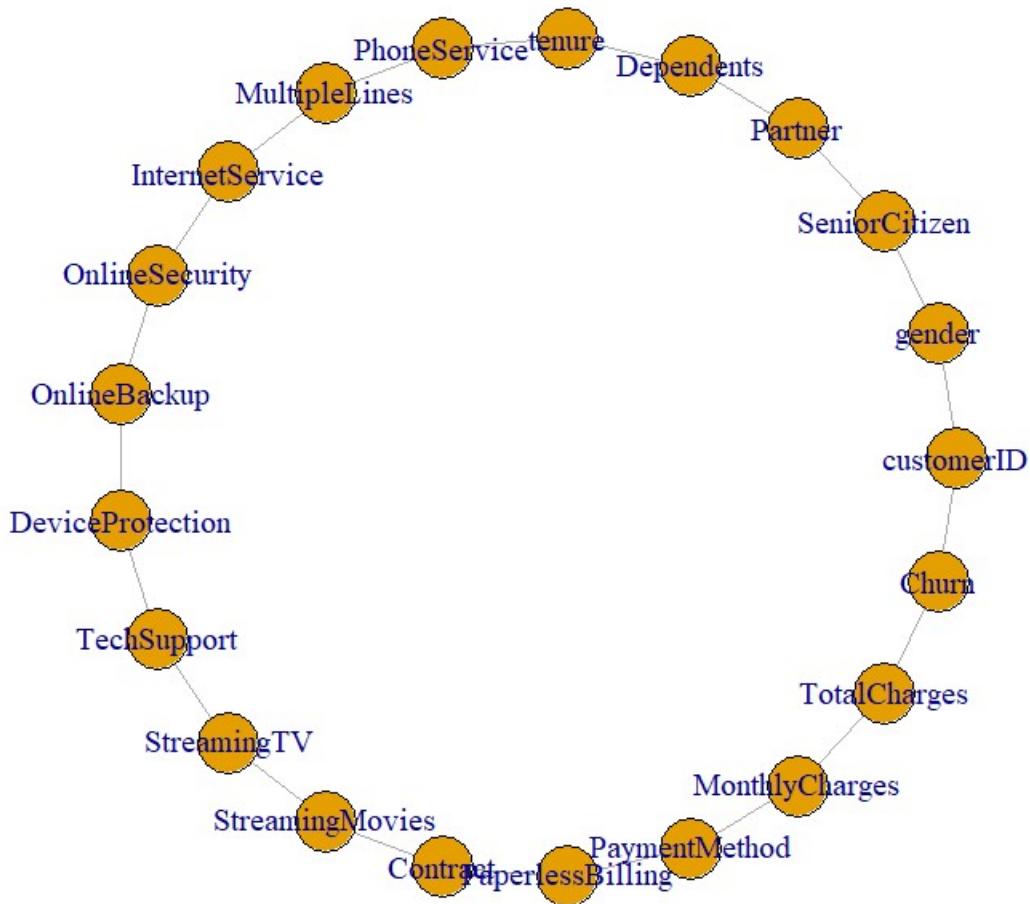
## Warning: The `origin` argument of `stat_bin()` is deprecated as of ggplot2 2.1.0.
## Please use the `boundary` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



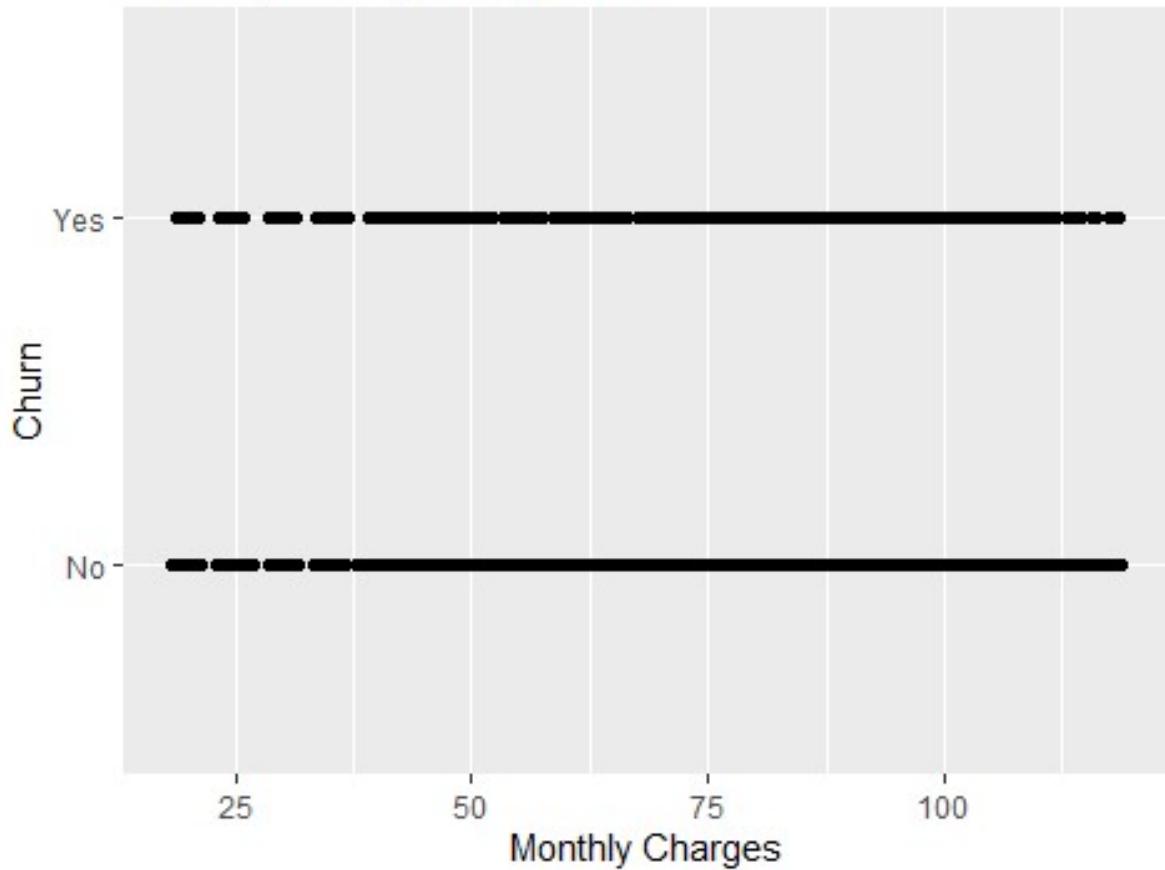
```
column_names <- c("customerID", "gender", "SeniorCitizen", "Partner", "Dependents", "tenure", "PhoneService", "MultipleLines", "InternetService", "OnlineSecurity", "OnlineBackup", "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies", "Contract", "PaperlessBilling", "PaymentMethod", "MonthlyCharges", "TotalCharges", "Churn")  
  
graph <- graph.empty(n = length(column_names), directed = FALSE)  
  
V(graph)$name <- column_names  
  
for (i in 1:(length(column_names) - 1)) {  
  graph <- add_edges(graph, c(i, i + 1))  
}  
  
graph <- add_edges(graph, c(length(column_names), 1))
```

```
plot(graph, layout = layout.circle(graph), vertex.label = V(graph)$name)
```



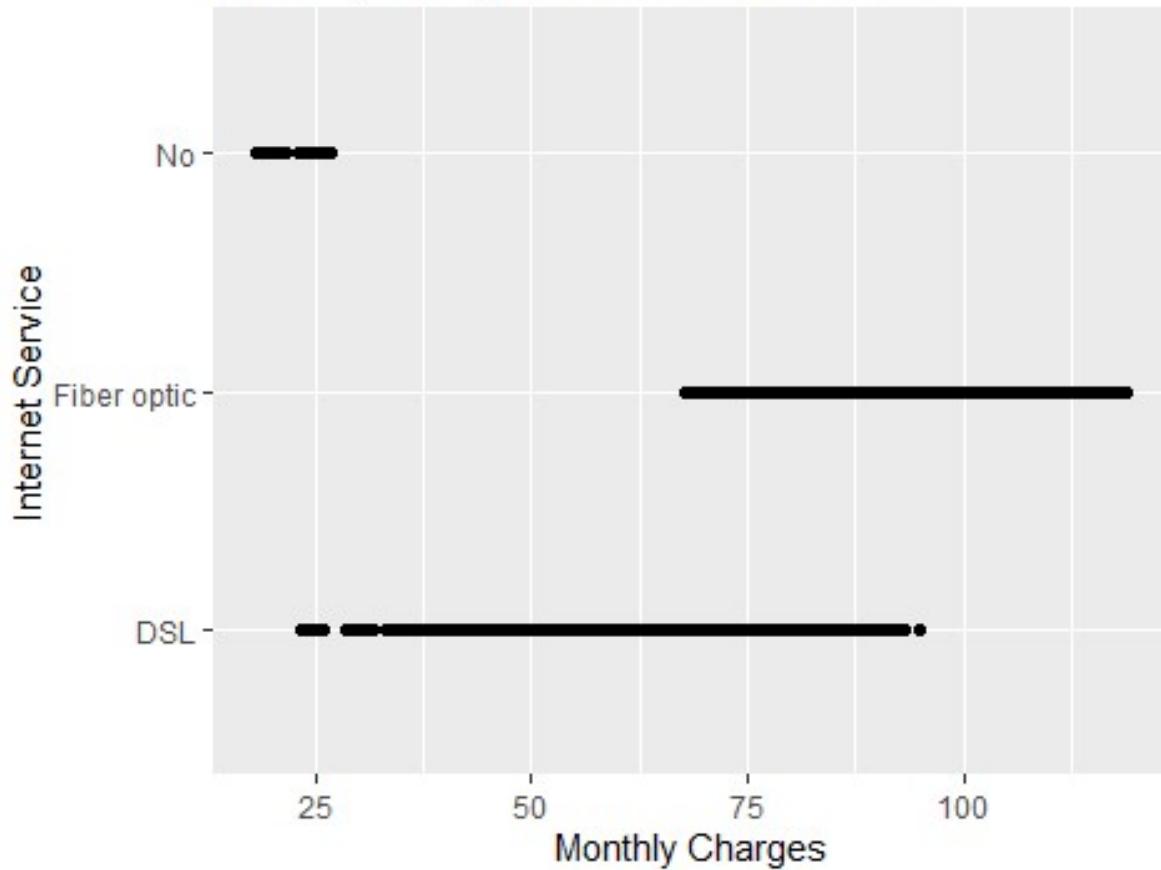
```
ggplot(telco_data, aes(x = MonthlyCharges, y = Churn)) +  
  geom_point() +  
  labs(title = "Monthly Charges vs. Churn",  
       x = "Monthly Charges",  
       y = "Churn")
```

Monthly Charges vs. Churn



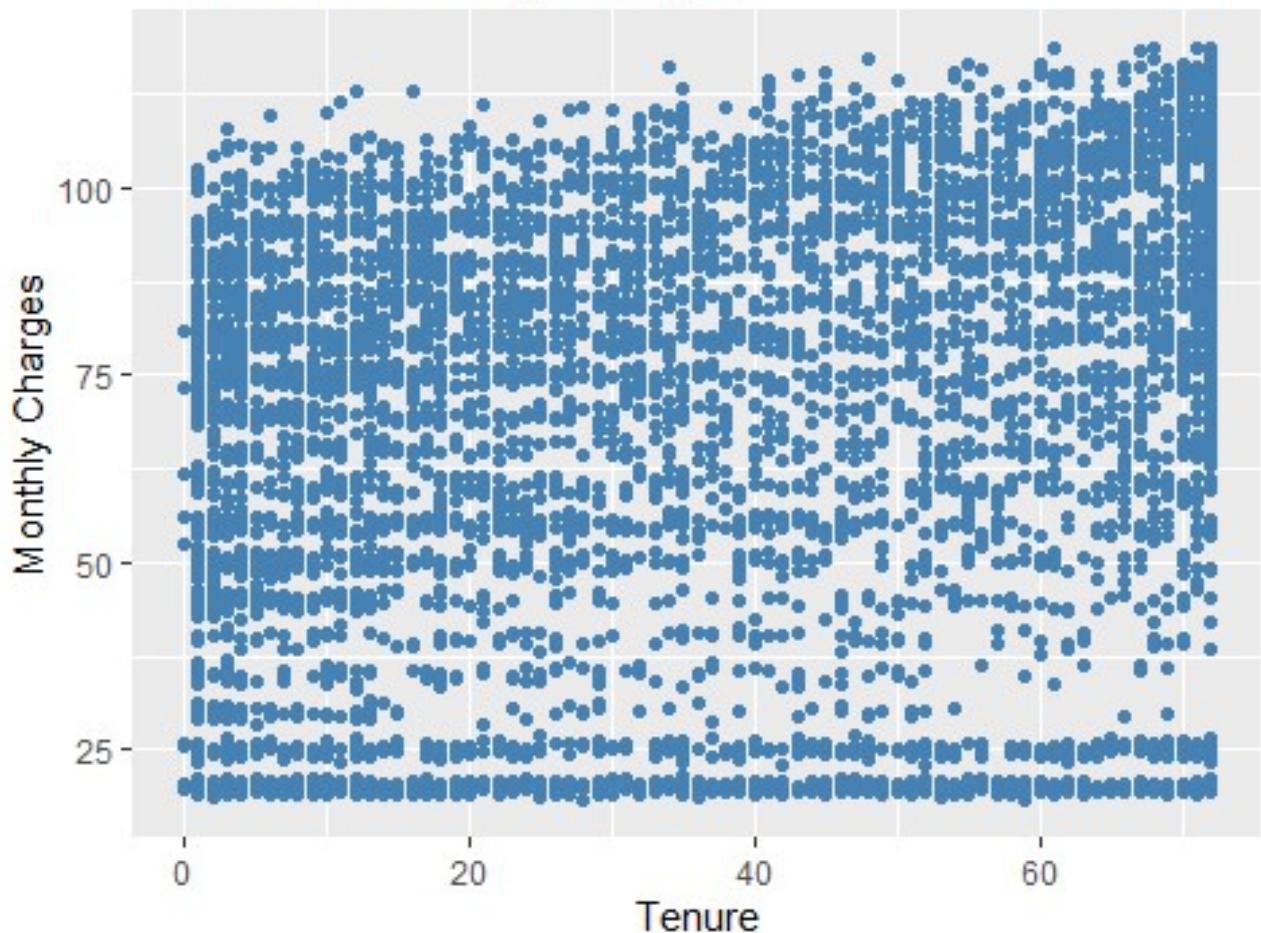
```
ggplot(telco_data, aes(x = MonthlyCharges, y = InternetService)) +  
  geom_point() +  
  labs(title = "Monthly Charges vs. Internet Service",  
       x = "Monthly Charges",  
       y = "Internet Service")
```

Monthly Charges vs. Internet Service



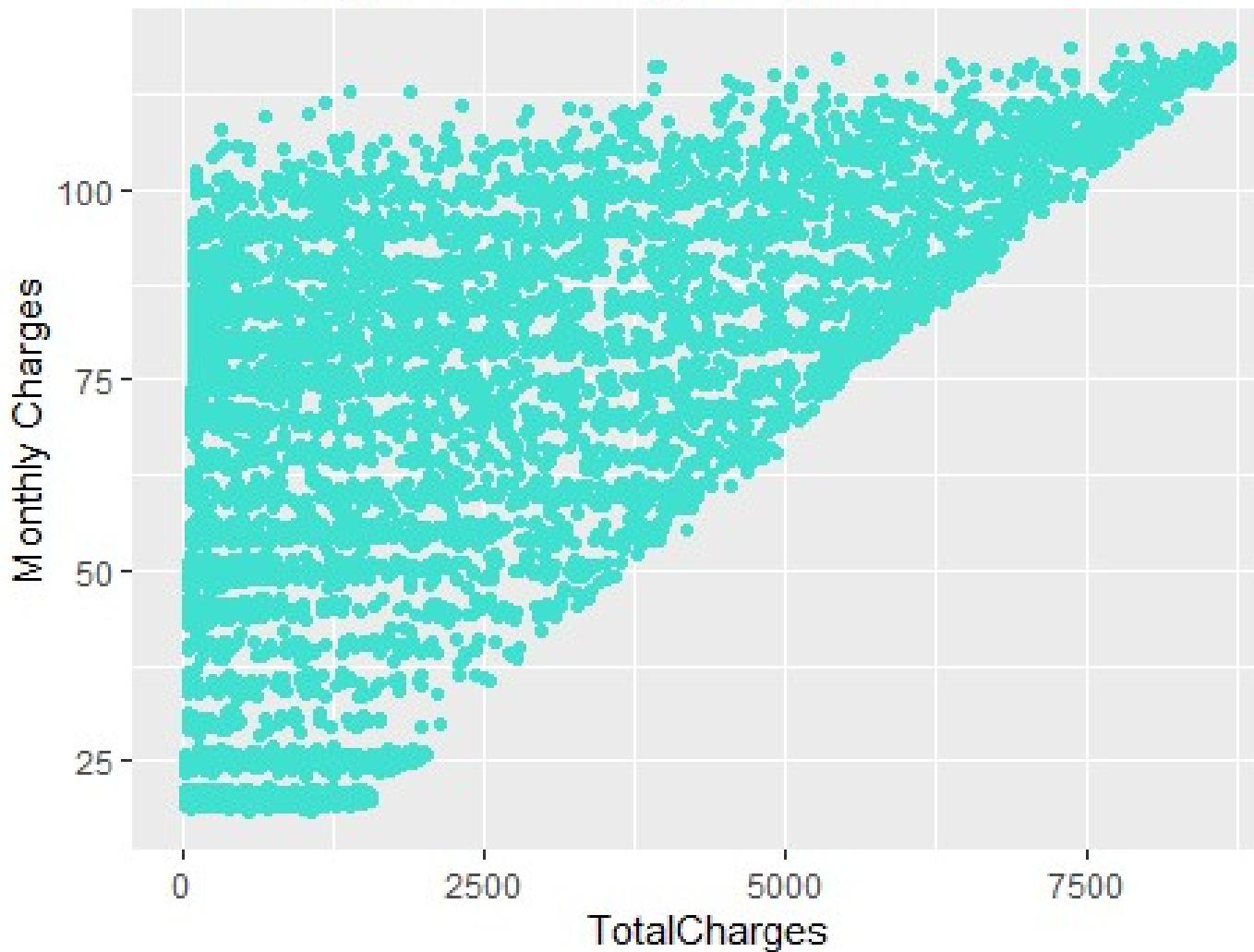
```
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges)) +  
  geom_point(color = "steelblue") +  
  labs(title = "Tenure vs. Monthly Charges",  
       x = "Tenure",  
       y = "Monthly Charges")
```

Tenure vs. Monthly Charges



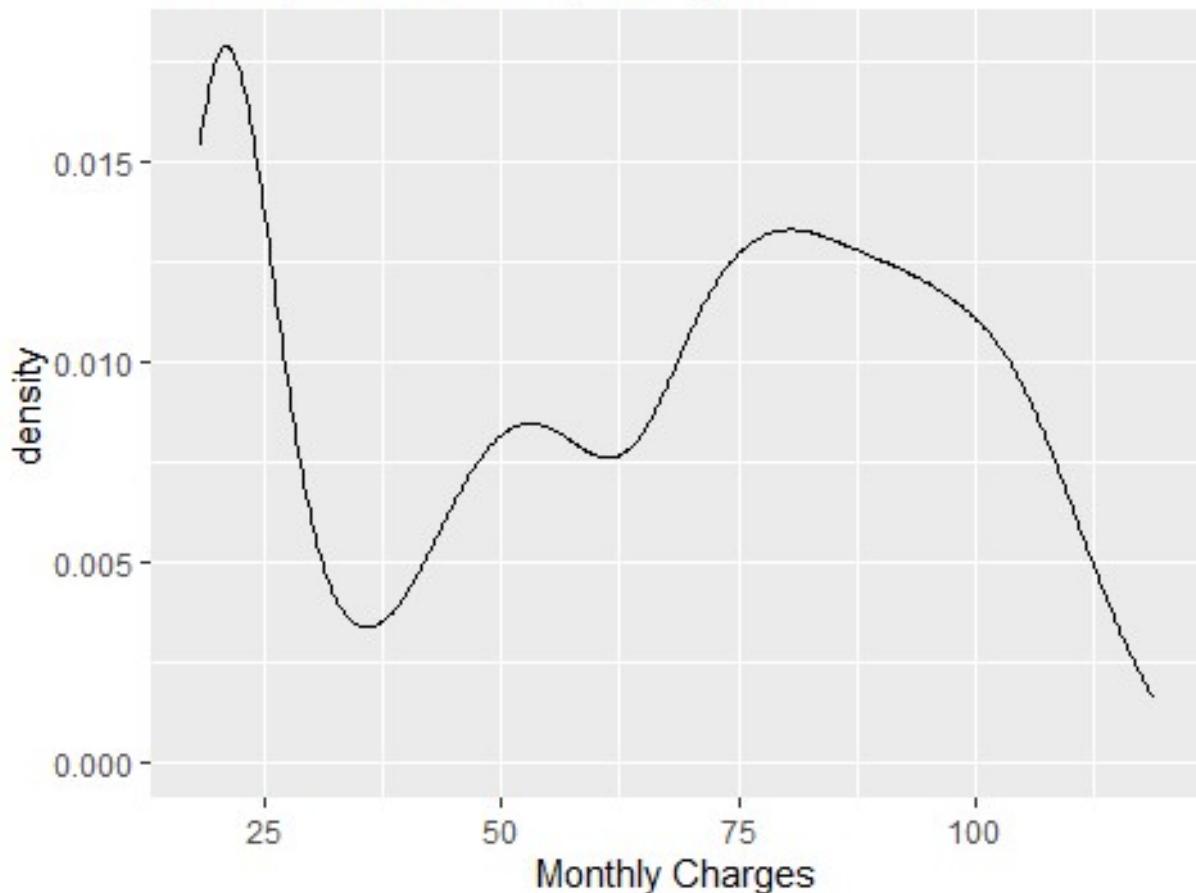
```
ggplot(telco_data, aes(x = TotalCharges, y = MonthlyCharges)) +  
  geom_point(color = "turquoise") +  
  labs(title = "TotalCharges vs MonthlyCharges",  
       x = "TotalCharges",  
       y = "Monthly Charges")  
  
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```

TotalCharges vs MonthlyCharges



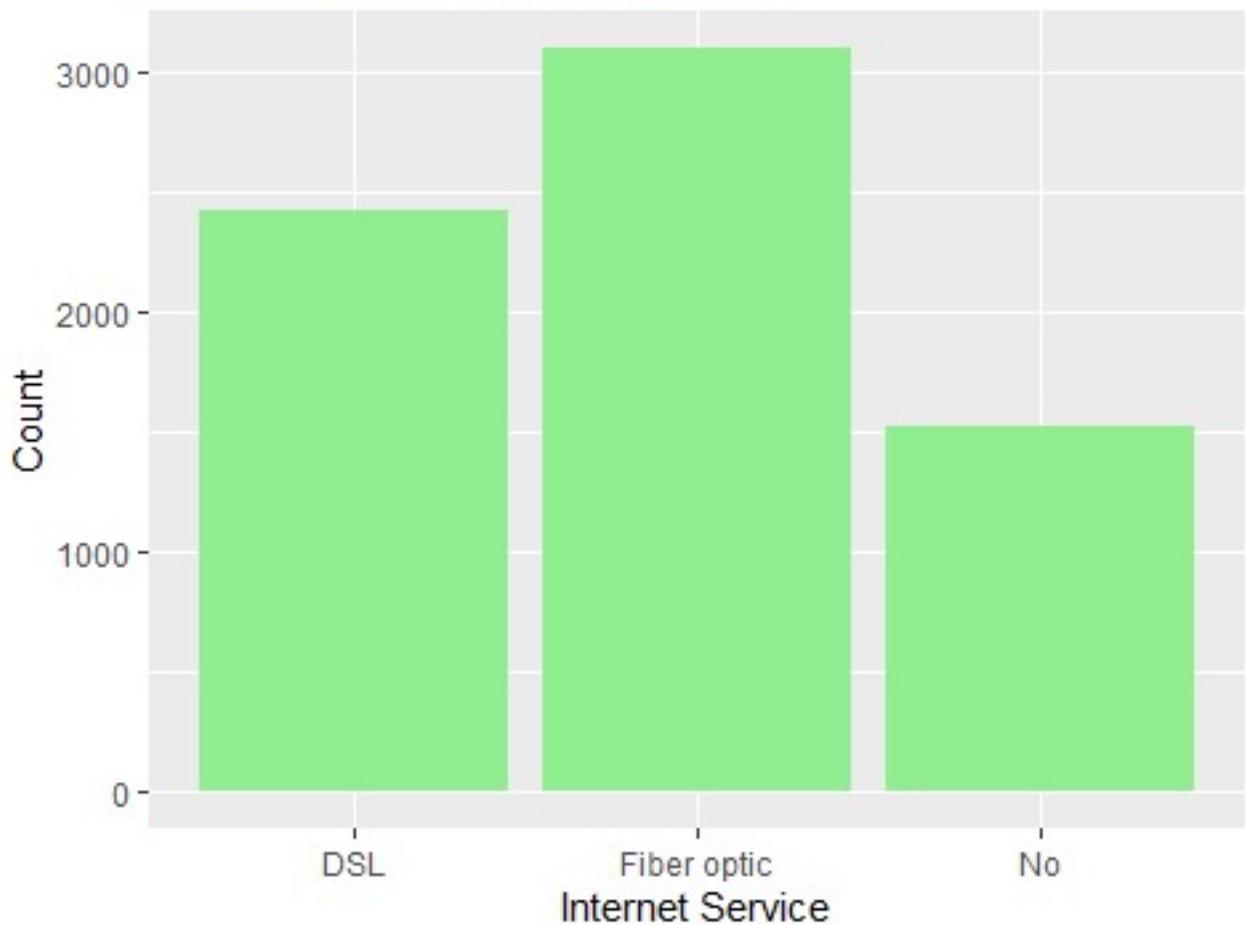
```
# Density plot: MonthlyCharges
ggplot(telco_data, aes(x = MonthlyCharges)) +
  geom_density() +
  labs(title = "Density Plot: Monthly Charges",
       x = "Monthly Charges")
```

Density Plot: Monthly Charges

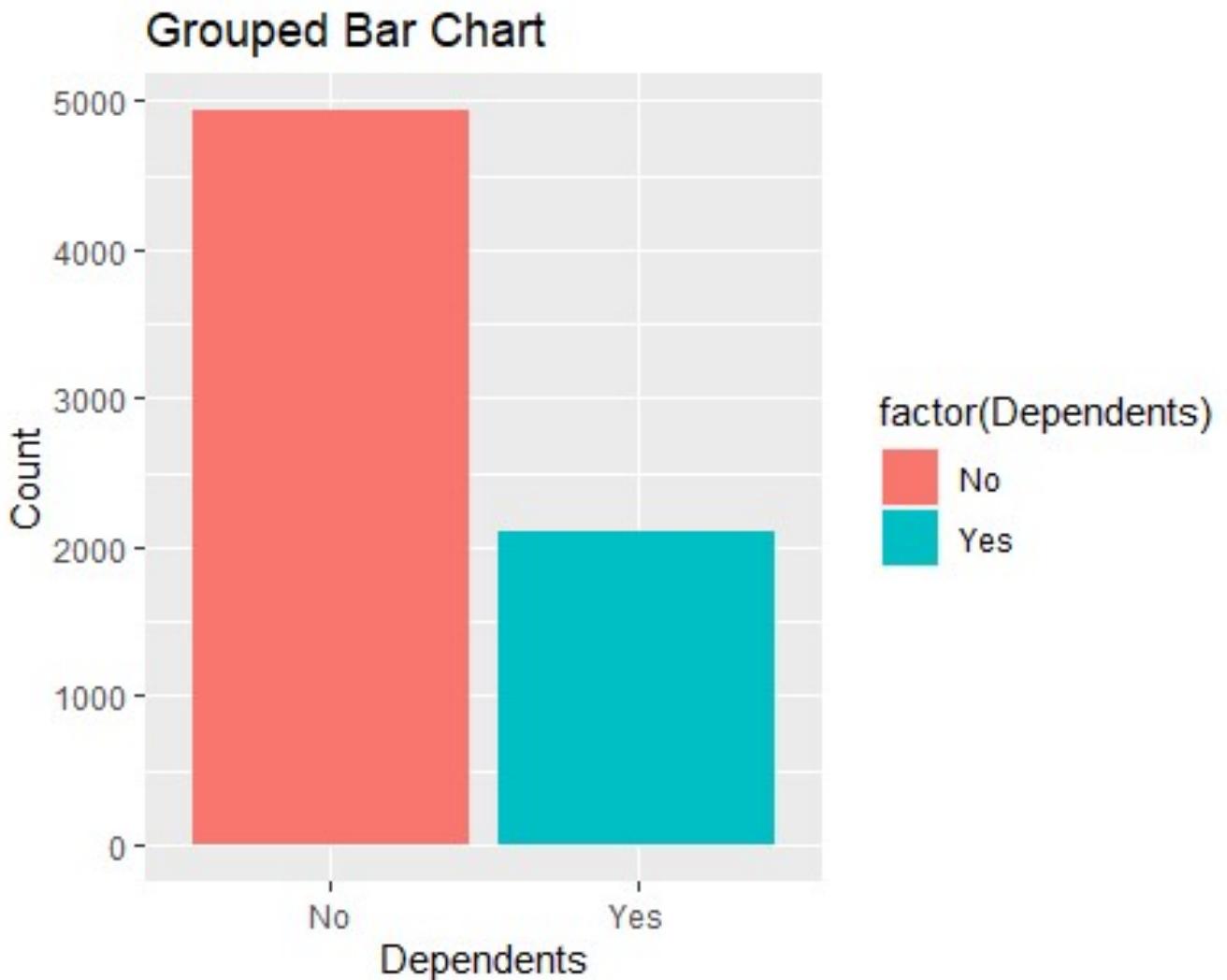


```
# Bar plot: InternetService distribution
ggplot(telco_data, aes(x = factor(InternetService))) +
  geom_bar(fill = "lightgreen") +
  labs(title = "Distribution of Internet Service",
       x = "Internet Service",
       y = "Count")
```

Distribution of Internet Service

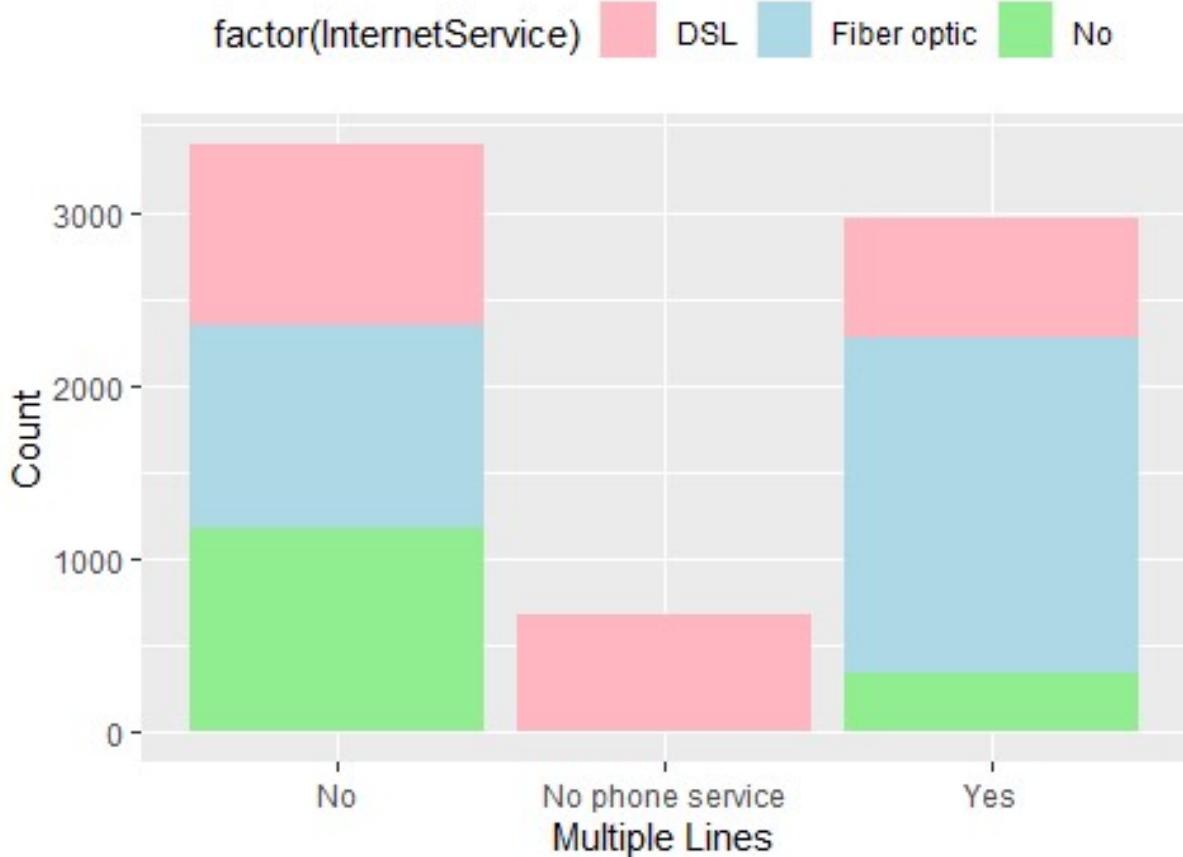


```
# Grouped bar chart: Dependents
ggplot(telco_data, aes(x = factor(Dependents), fill = factor(Dependents))) +
  geom_bar() +
  labs(title = "Grouped Bar Chart",
       x = "Dependents",
       y = "Count")
```



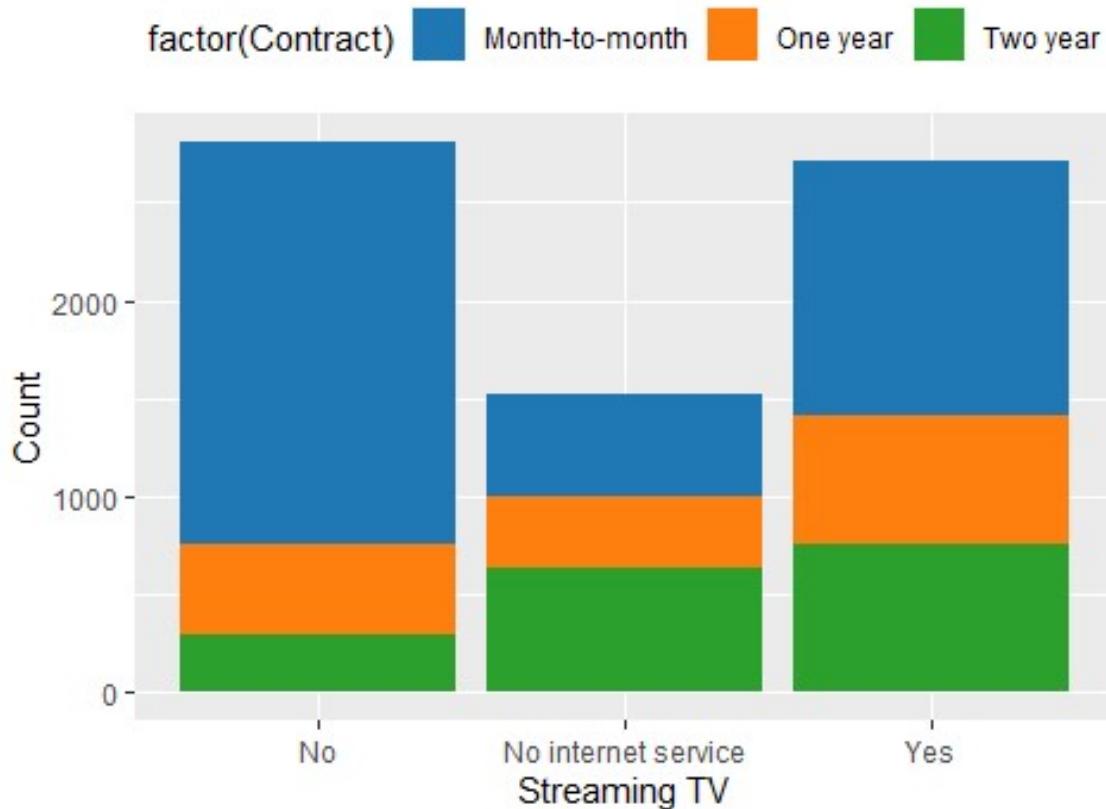
```
# Stack plot: MultipleLines and InternetService
ggplot(telco_data, aes(x = factor(MultipleLines), fill = factor(InternetService))) +
  geom_bar() +
  labs(title = "Stacked Bar Plot of Multiple Lines and Internet Service",
       x = "Multiple Lines",
       y = "Count") +
  scale_fill_manual(values = c("lightpink", "lightblue", "lightgreen",
"lightyellow")) +
  theme(legend.position = "top")
```

Stacked Bar Plot of Multiple Lines and Internet Service



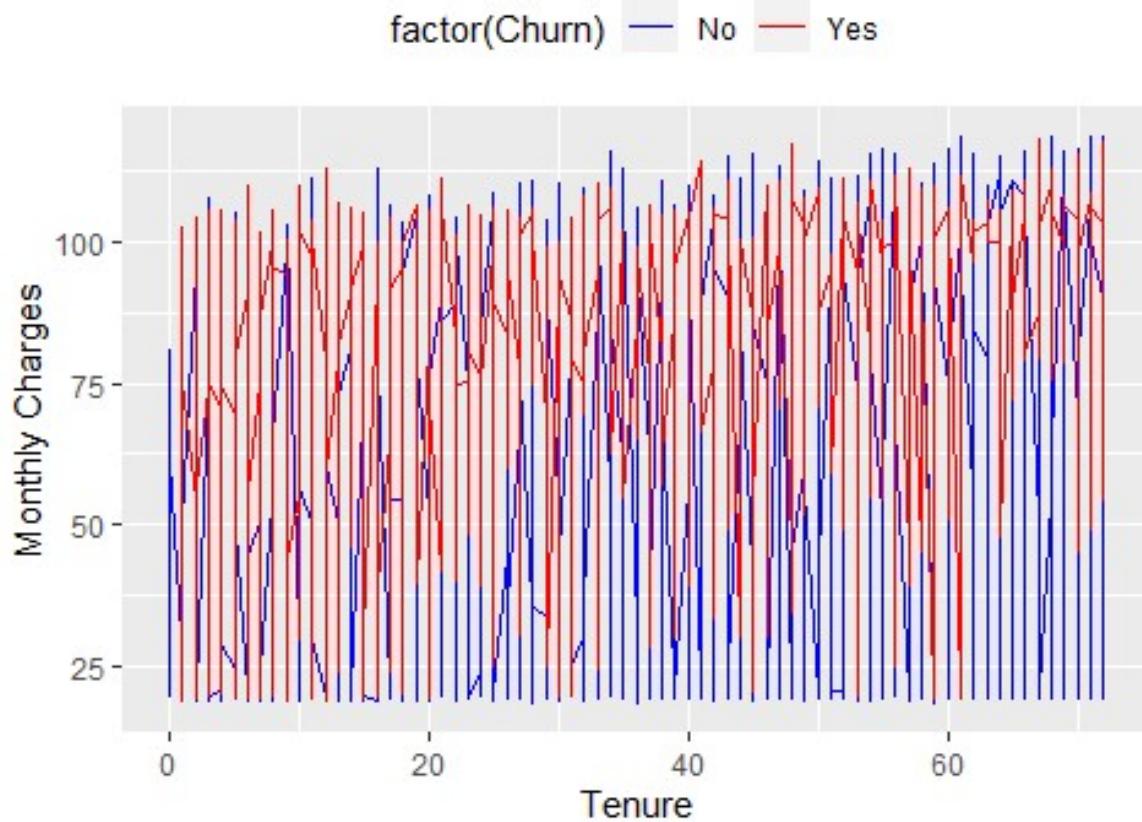
```
# Stacked bar plot: StreamingTV and Contract
ggplot(telco_data, aes(x = factor(StreamingTV), fill = factor(Contract))) +
  geom_bar() +
  labs(title = "Stacked Bar Plot of Streaming TV and Contract",
       x = "Streaming TV",
       y = "Count") +
  scale_fill_manual(values = c("#1f77b4", "#ff7f0e", "#2ca02c")) +
  theme(legend.position = "top")
```

Stacked Bar Plot of Streaming TV and Contract



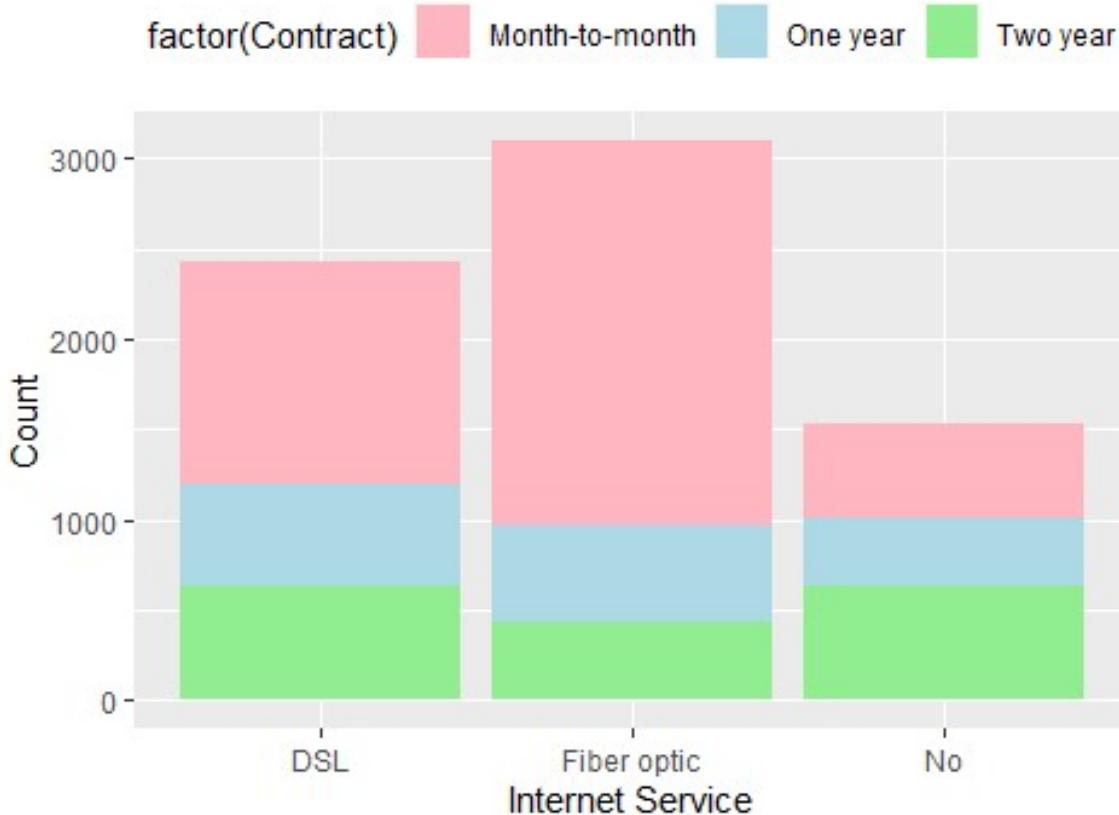
```
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, color = factor(C  
hurn))) +  
  geom_line() +  
  labs(title = "Monthly Charges Over Tenure",  
       x = "Tenure",  
       y = "Monthly Charges") +  
  scale_color_manual(values = c("blue", "red")) +  
  theme(legend.position = "top")
```

Monthly Charges Over Tenure



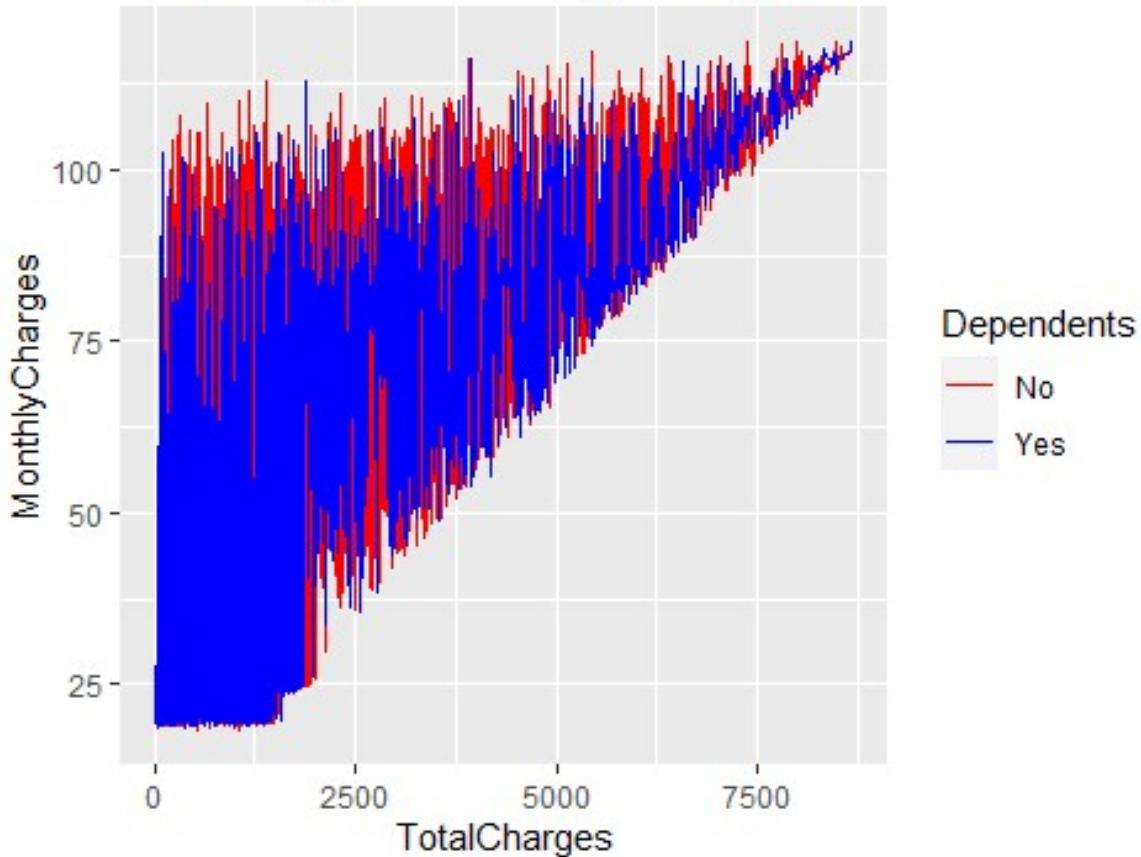
```
ggplot(telco_data, aes(x = factor(InternetService), fill = factor(Contract))) +  
  geom_bar() +  
  labs(title = "Stacked Bar Plot of Internet Service and Contract",  
       x = "Internet Service",  
       y = "Count") +  
  scale_fill_manual(values = c("lightpink", "lightblue", "lightgreen"))  
+  
  theme(legend.position = "top")
```

Stacked Bar Plot of Internet Service and Contract



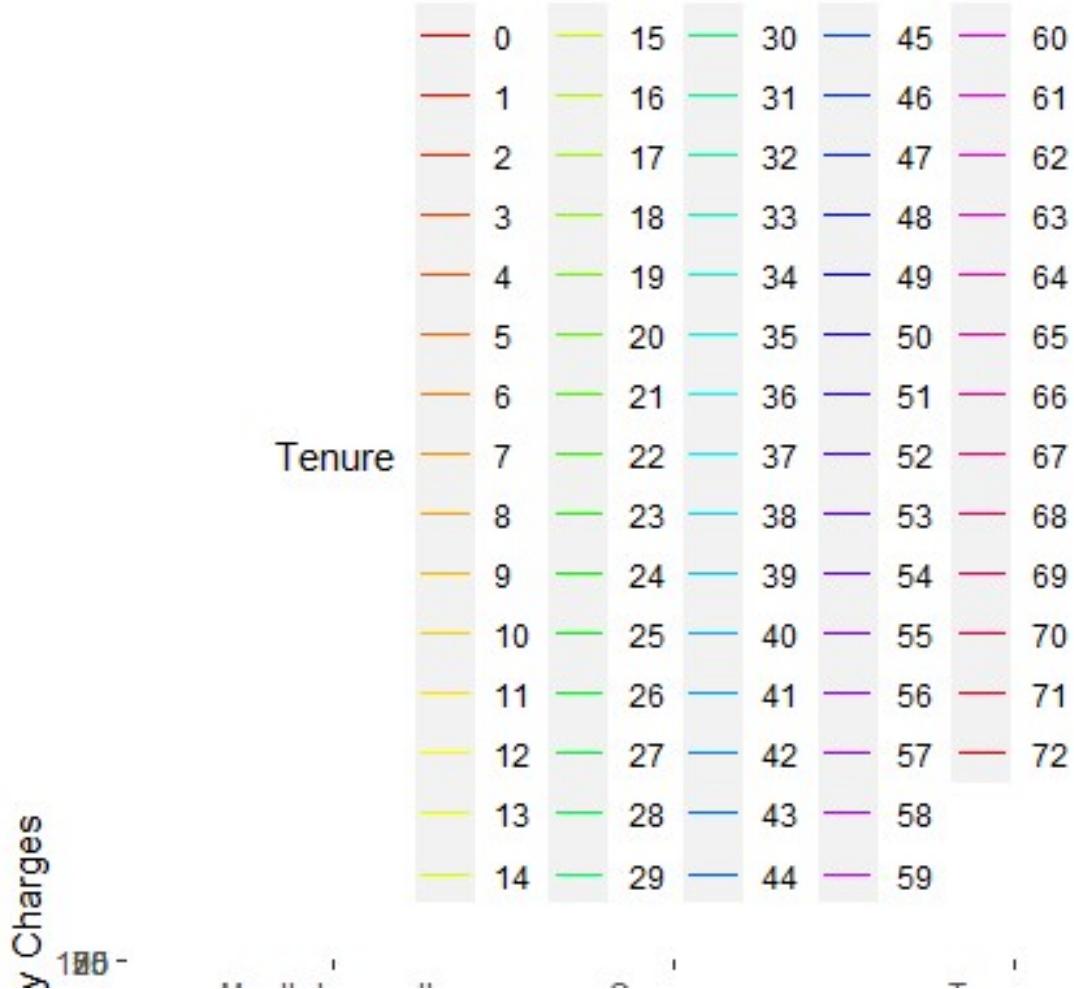
```
# Create the Line graph
ggplot(telco_data, aes(x = TotalCharges, y = MonthlyCharges, color = as.factor(Dependents))) +
  geom_line() +
  labs(title = "TotalCharges vs Monthly Charges",
       x = "TotalCharges",
       y = "MonthlyCharges",
       color = "Dependents") +
  scale_color_manual(values = c("red", "blue", "green", "orange", "purple",
                               "cyan", "brown", "pink"))
## Warning: Removed 11 rows containing missing values (`geom_line()`).
```

TotalCharges vs Monthly Charges



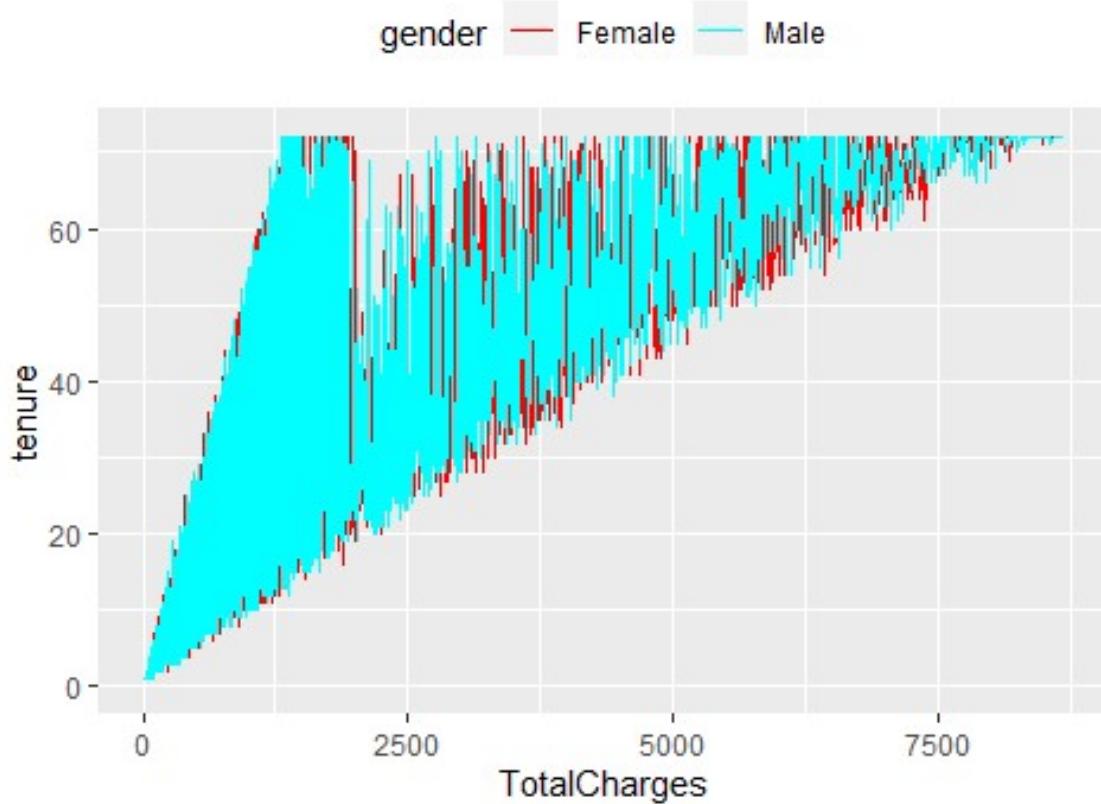
```
library(ggplot2)
library(dplyr)

# Line plot with multiple lines for 'MonthlyCharges' and 'tenure' over 'Contract'
ggplot(telco_data, aes(x = Contract, y = MonthlyCharges, color = as.factor(tenure))) +
  geom_line() +
  labs(title = "Monthly Charges and Tenure by Contract",
       x = "Contract",
       y = "Monthly Charges",
       color = "Tenure") +
  scale_color_manual(values = rainbow(length(unique(telco_data$tenure)))) +
  theme(legend.position = "top")
```



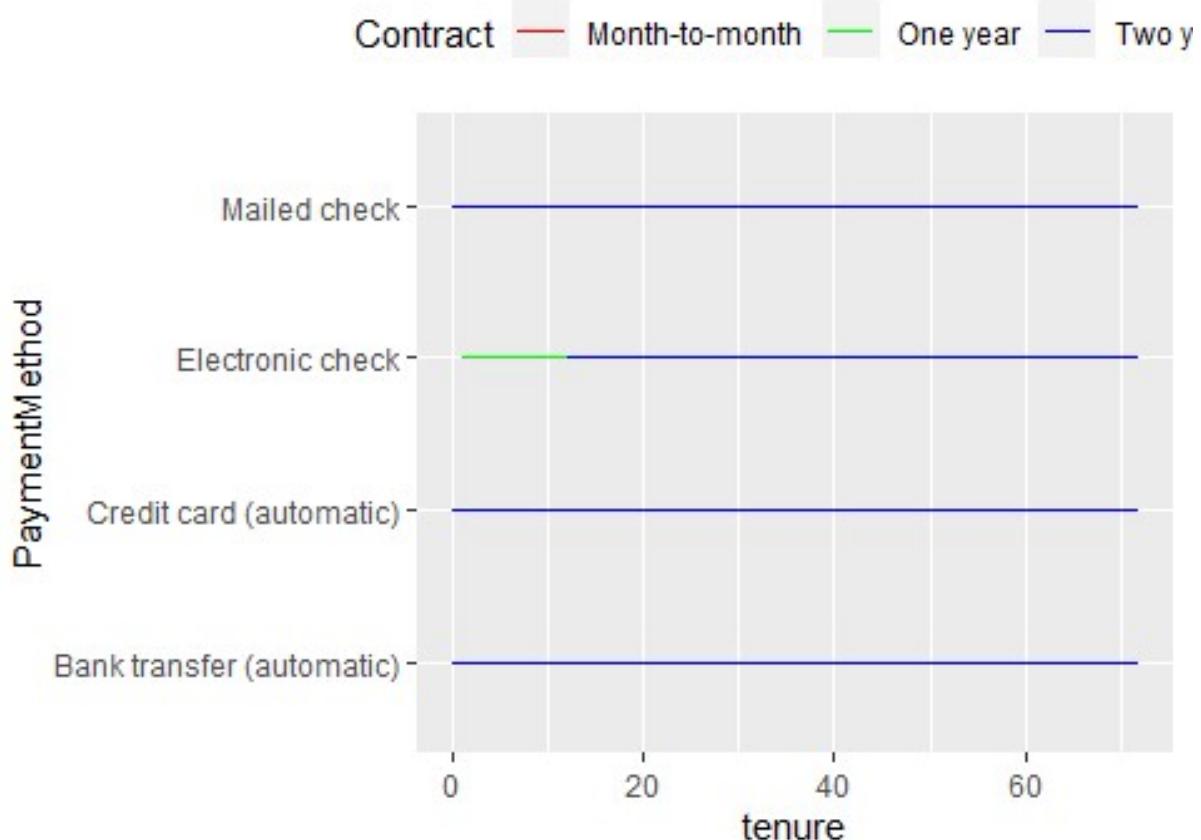
```
# Line plot with multiple lines for 'MonthlyCharges' and 'Dependents' over 'Contract'
ggplot(telco_data, aes(x = TotalCharges, y = tenure, color = as.factor(gender))) +
  geom_line() +
  labs(title = "TotalCharges and tenure by gender",
       x = "TotalCharges",
       y = "tenure",
       color = "gender") +
  scale_color_manual(values = rainbow(length(unique(telco_data$gender)))) +
  theme(legend.position = "top")
## Warning: Removed 11 rows containing missing values (`geom_line()`).
```

TotalCharges and tenure by gender



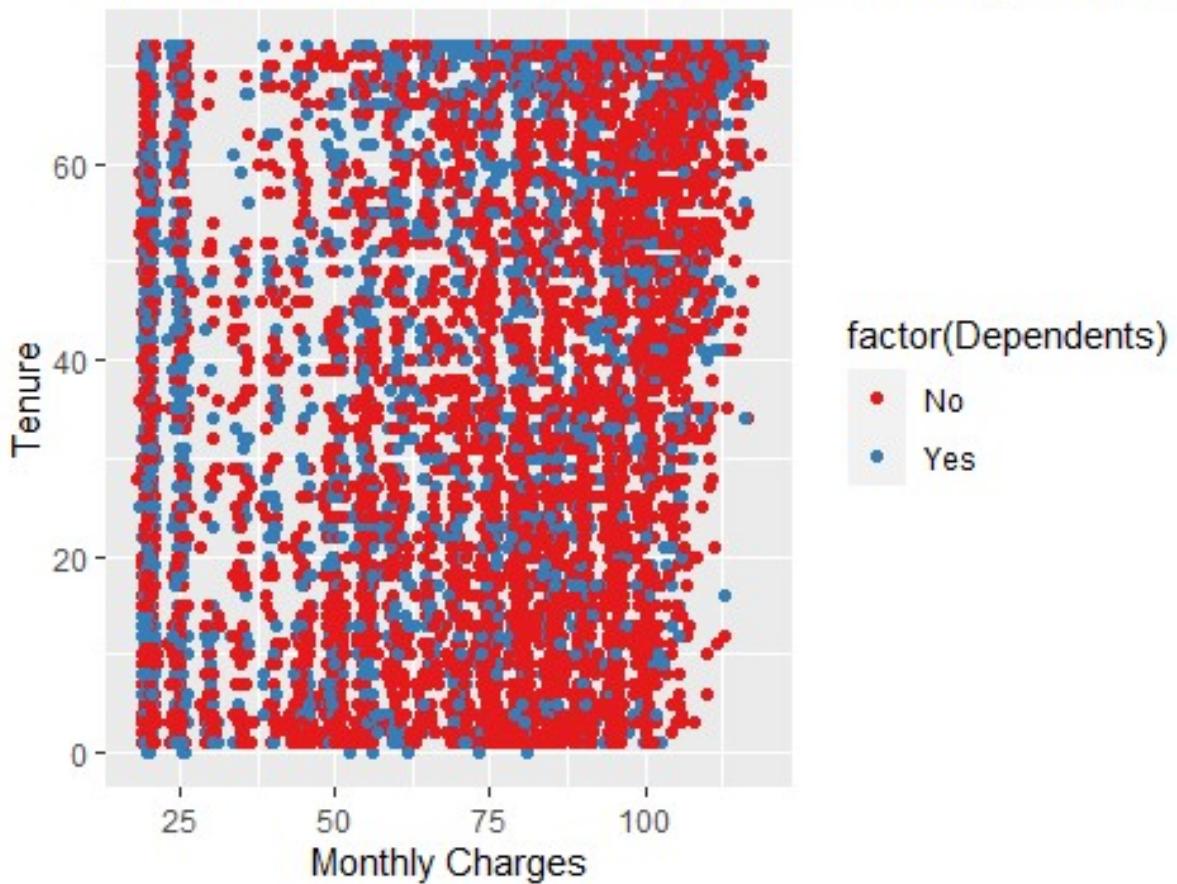
```
# Line plot with multiple lines for 'PaymentMethod' and 'Contract' over
# 'tenure'
ggplot(telco_data, aes(x = tenure, y = PaymentMethod, color = as.factor
(Contract))) +
  geom_line() +
  labs(title = "Monthly Charges and Contract by Dependents",
       x = "tenure",
       y = "PaymentMethod",
       color = "Contract") +
  scale_color_manual(values = rainbow(length(unique(telco_data$Contract)))) +
  theme(legend.position = "top")
```

Monthly Charges and Contract by Dependent



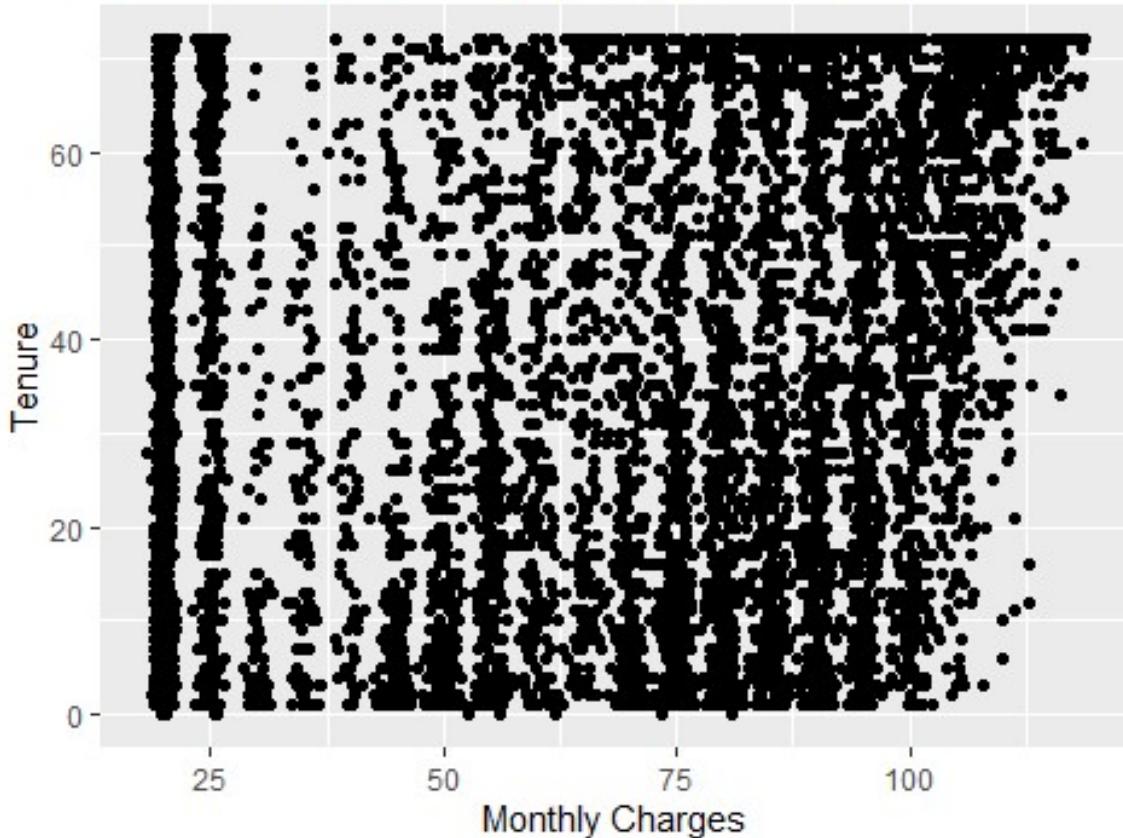
```
# Scatter plot of grouped data
ggplot(telco_data, aes(x = MonthlyCharges, y = tenure, color = factor(Dependents))) +
  geom_point() +
  labs(title = "Scatter Plot of Monthly Charges vs. Tenure (Grouped by Dependents)",
       x = "Monthly Charges",
       y = "Tenure") +
  scale_color_brewer(palette = "Set1")
```

Scatter Plot of Monthly Charges vs. Tenure (Grouped by Dependents)



```
# Basic scatter plot
ggplot(telco_data, aes(x = MonthlyCharges, y = tenure)) +
  geom_point() +
  labs(title = "Scatter Plot of Monthly Charges vs. Tenure",
       x = "Monthly Charges",
       y = "Tenure")
```

Scatter Plot of Monthly Charges vs. Tenure

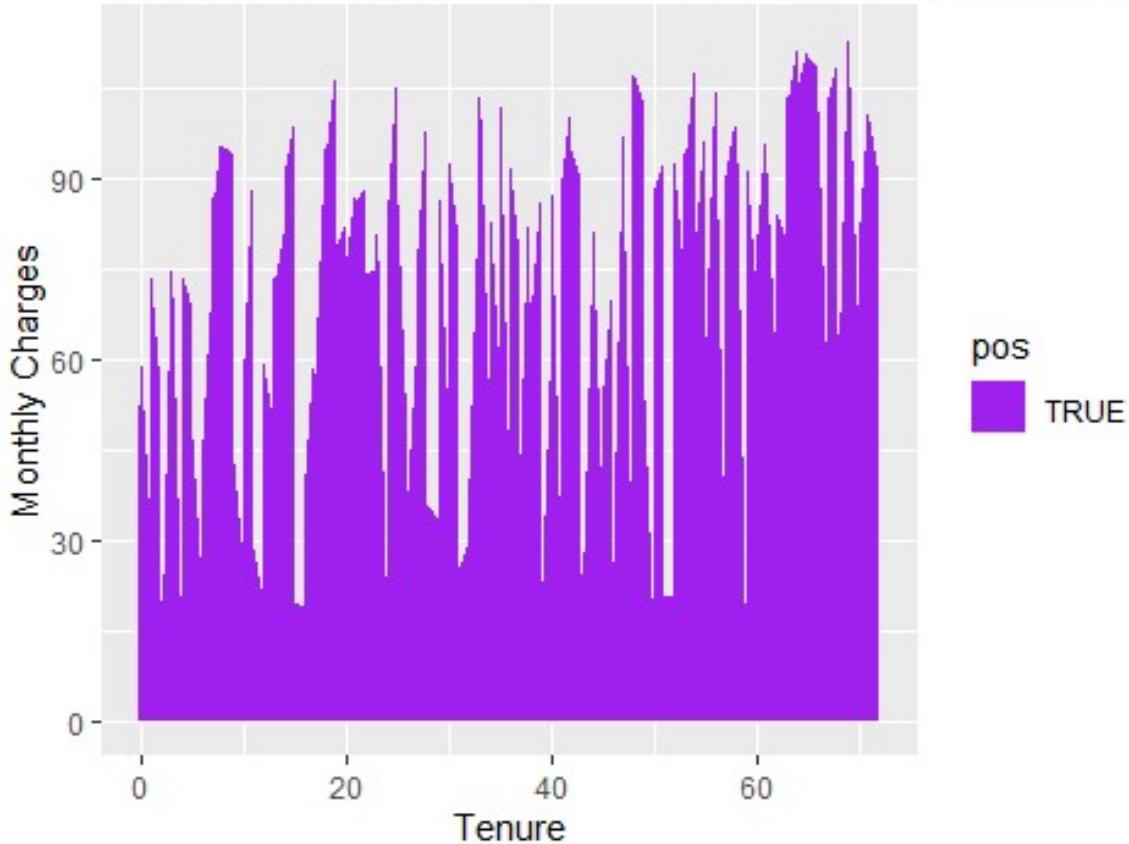


```
# Create a subset of telco_data for visualization
telco_sub <- telco_data %>%
  select(tenure, MonthlyCharges, Churn) # Choose relevant columns for v
isualization

# Calculate positive and negative values
telco_sub <- telco_sub %>%
  mutate(pos = MonthlyCharges >= 0)

# Stacked area graph
ggplot(telco_sub, aes(x = tenure, y = MonthlyCharges, fill = pos)) +
  geom_area(position = "stack") +
  scale_fill_manual(values = c("purple", "green")) + # Set colors for
positive and negative values
  labs(title = "Stacked Area Graph of Tenure and Monthly Charges",
       x = "Tenure",
       y = "Monthly Charges")
```

Stacked Area Graph of Tenure and Monthly Charges



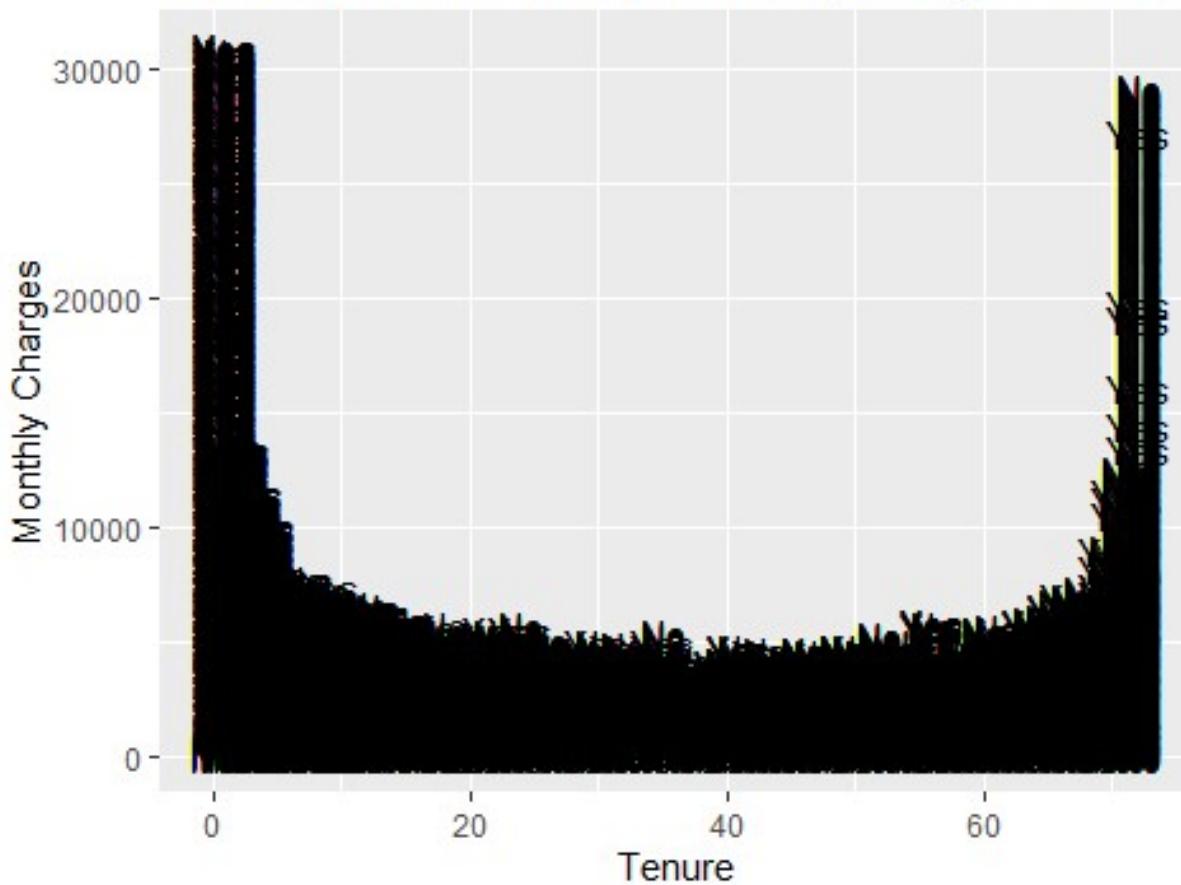
```
# Bar graph with labels
ggplot(telco_sub, aes(x = tenure, y = MonthlyCharges, fill = pos, label
= Churn)) +
  geom_col(position = "identity", colour = "black", size = 0.25) +
  scale_fill_manual(values = c("red", "green"), guide = FALSE) + # Set
  colors for positive and negative values
  geom_text(position = position_stack(vjust = 0.5)) + # Add Labels to
  the bars
  labs(title = "Bar Graph of Tenure and Monthly Charges with Labels",
        x = "Tenure",
        y = "Monthly Charges")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2
3.4.0.
## [i] Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warnin
g was
## generated.

## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This
was deprecated in
```

```
## ggplot2 3.3.4.
## ⚠ Please use "none" instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

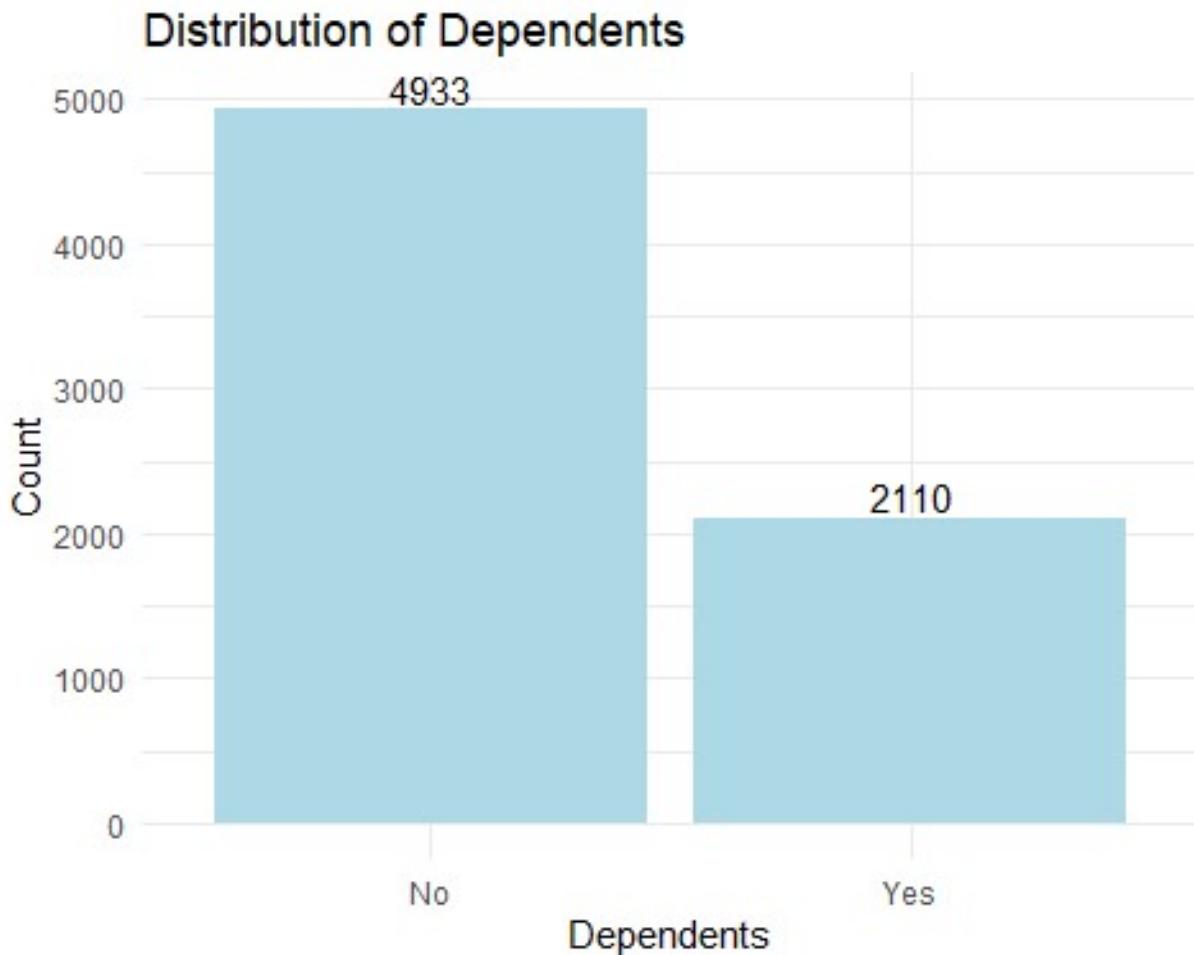
Bar Graph of Tenure and Monthly Charges with Labels



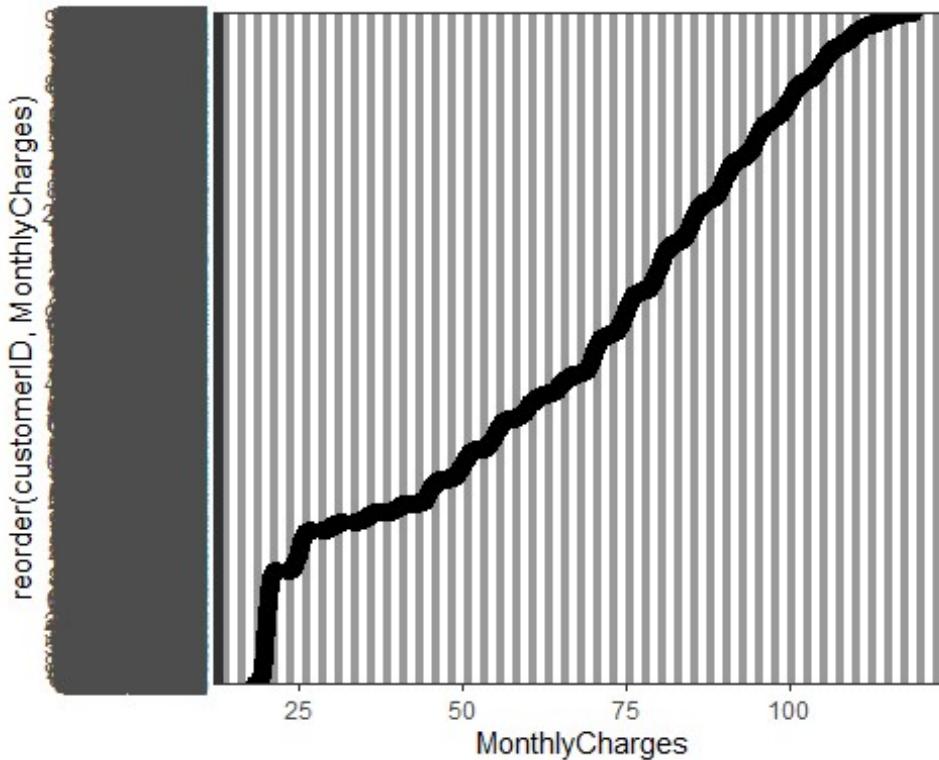
```
# Create a bar graph with Labels
ggplot(telco_data, aes(x = factor(Dependents))) +
  geom_bar(fill = "lightblue") +
  labs(title = "Distribution of Dependents",
       x = "Dependents",
       y = "Count") +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.2, color
= "black") +
  theme_minimal()

## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot
## 2.3.4.0.
## ⚠ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
# Create a Cleveland Dot Plot
ggplot(telco_data, aes(x = MonthlyCharges, y = reorder(customerID, MonthlyCharges))) +
  geom_point(size = 3) +
  theme_bw() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey60", linetype = "dashed")
  )
```

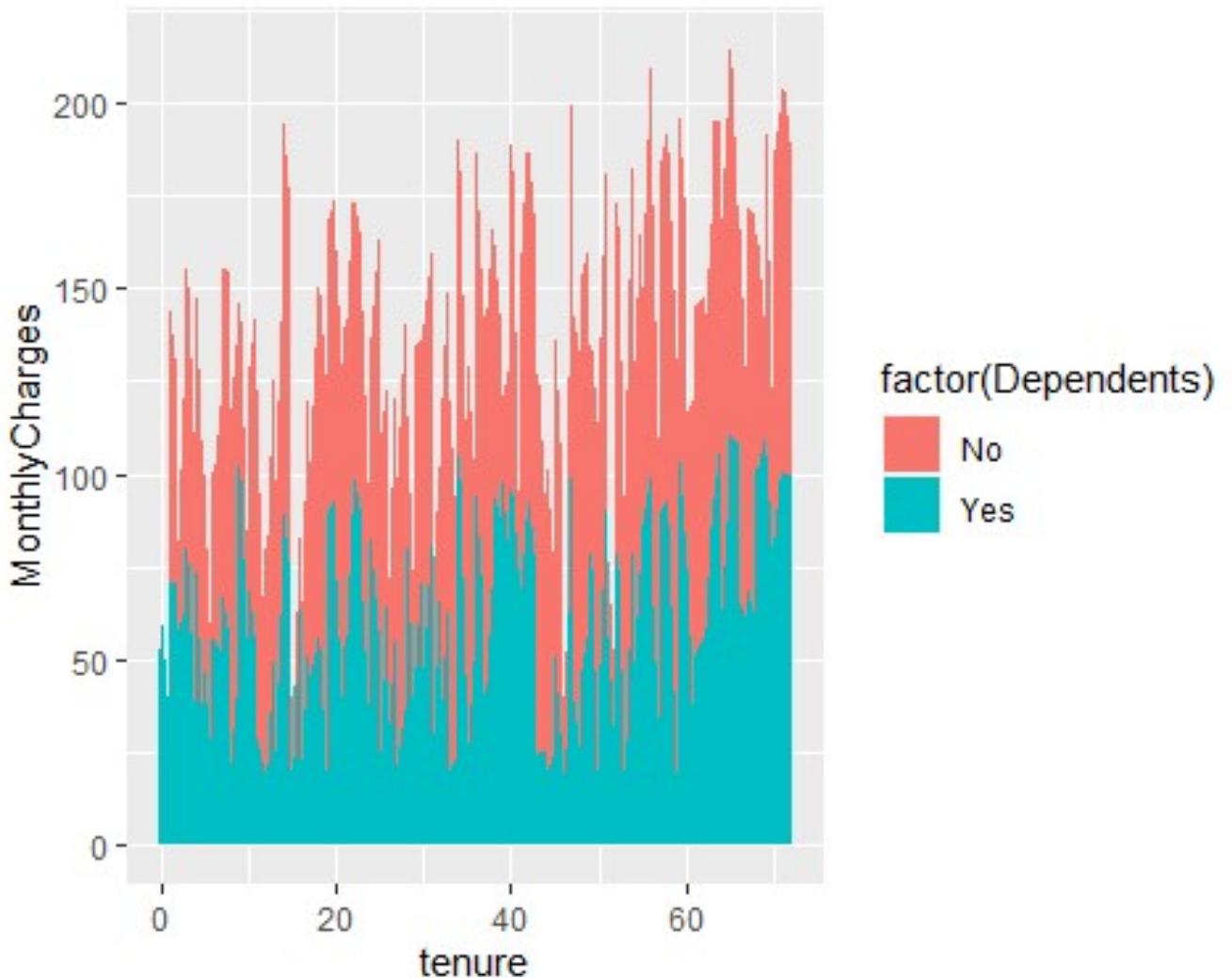


```
library(gcookbook) # Load gcookbook for the uspopage data set

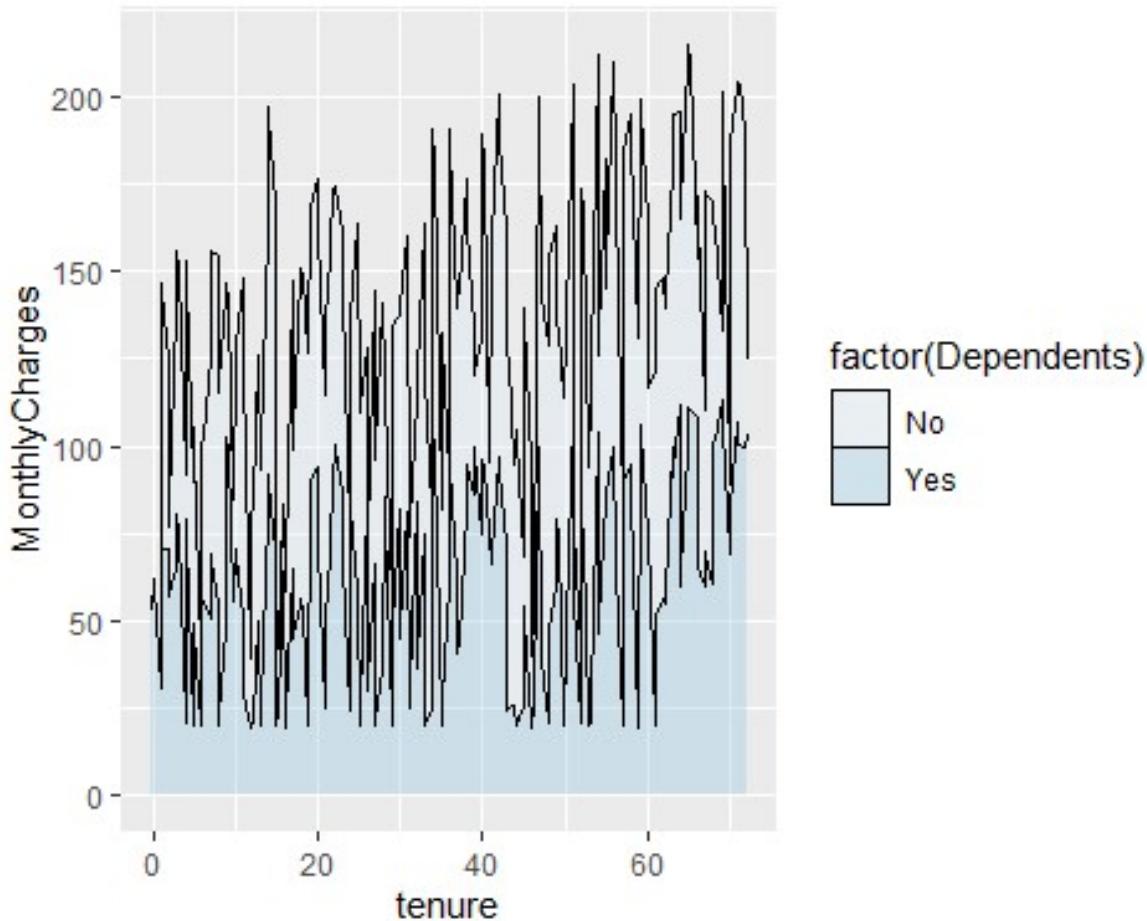
## 
## Attaching package: 'gcookbook'

## The following object is masked from 'package:plotly':
## 
##     wind

ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, fill = factor(De-
pendents))) +
  geom_area()
```

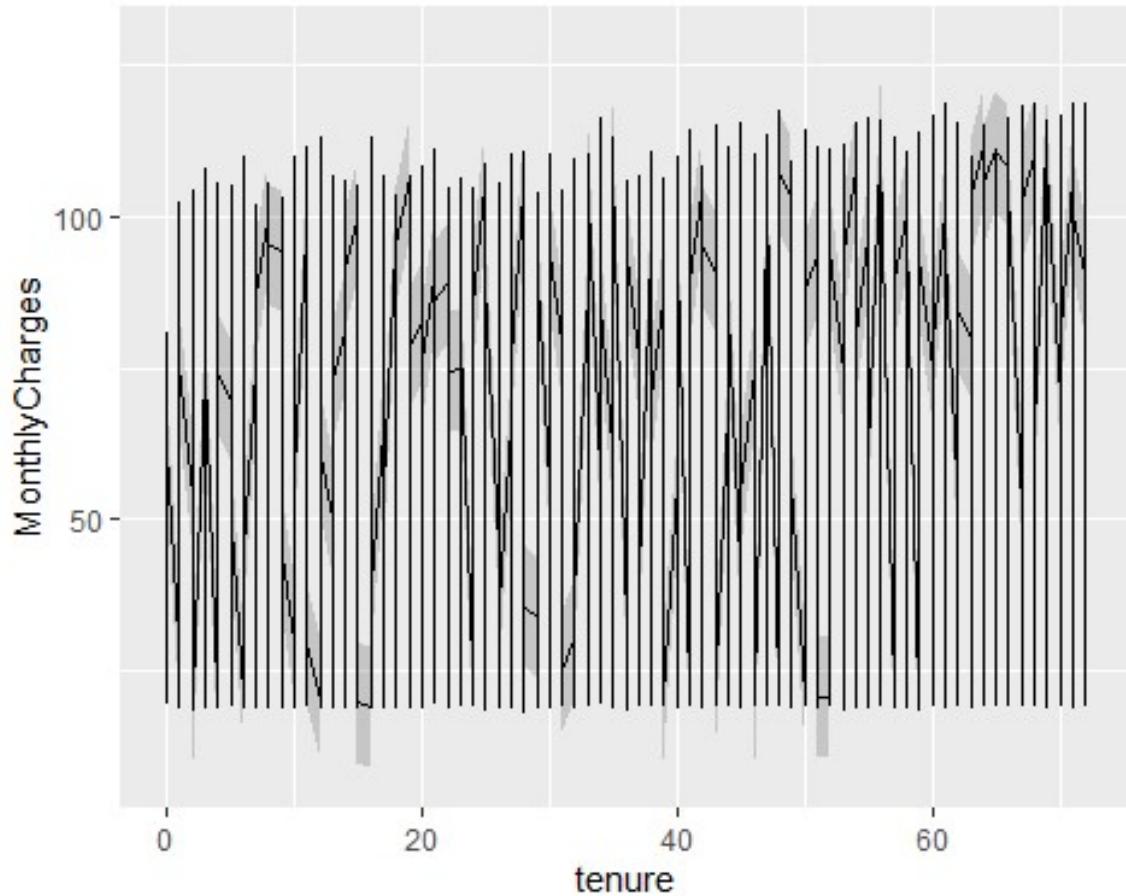


```
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, fill = factor(Dependents))) +  
  geom_area(colour = "black", size = 0.2, alpha = 0.4) +  
  scale_fill_brewer(palette = "Blues")
```



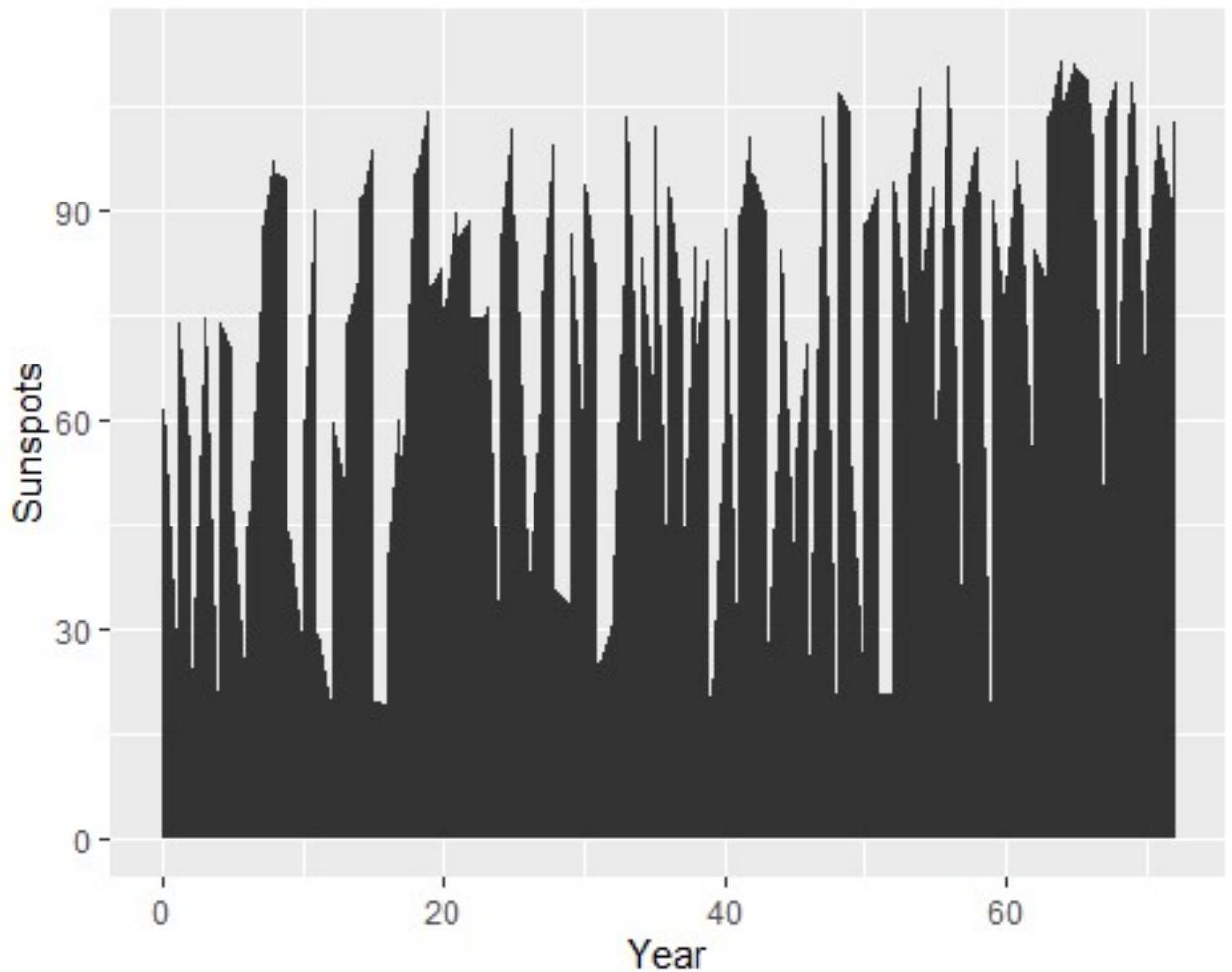
```
# Grab a subset of the telco_data for the example
telco_subset <- subset(telco_data, select = c(tenure, MonthlyCharges, C
hurn))

ggplot(telco_subset, aes(x = tenure, y = MonthlyCharges)) +
  geom_ribbon(aes(ymin = MonthlyCharges - 10, ymax = MonthlyCharges + 1
0), alpha = 0.2) +
  geom_line()
```

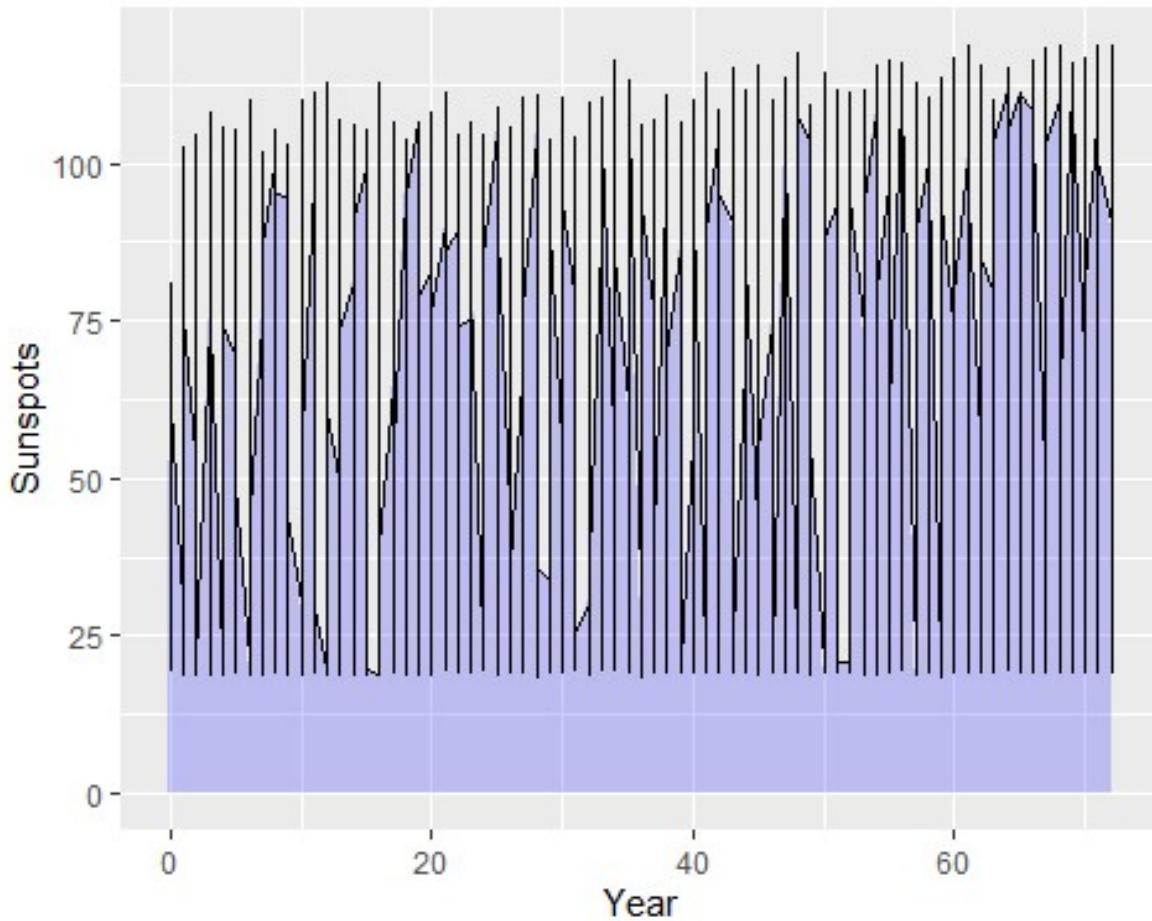


```
# Convert the telco_data dataset into a data frame for this example
telco_df <- data.frame(
  Year = as.numeric(telco_data$tenure),
  Sunspots = as.numeric(telco_data$MonthlyCharges)
)

ggplot(telco_df, aes(x = Year, y = Sunspots)) +
  geom_area()
```

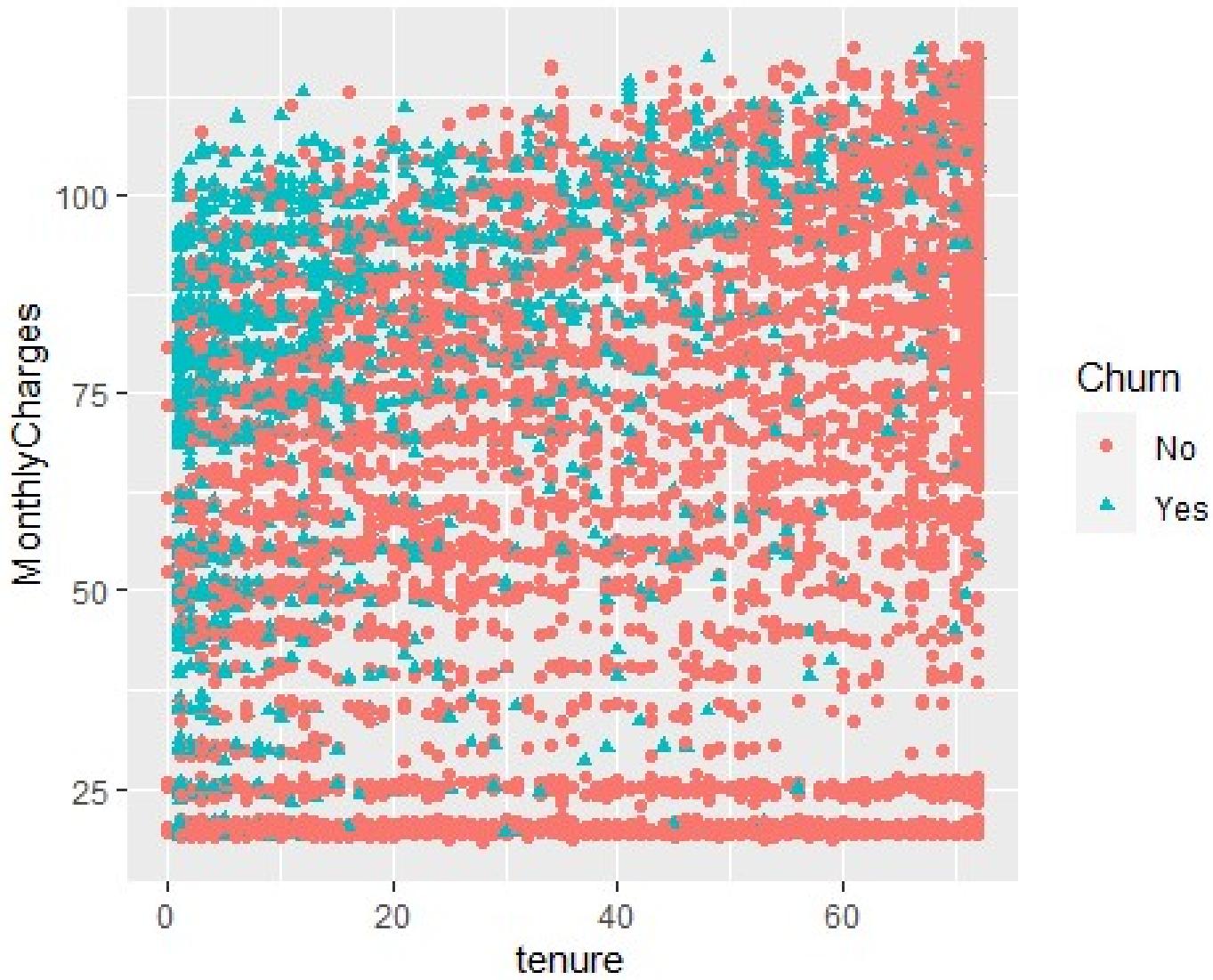


```
ggplot(telco_df, aes(x = Year, y = Sunspots)) +  
  geom_area(fill = "blue", alpha = 0.2) +  
  geom_line()
```

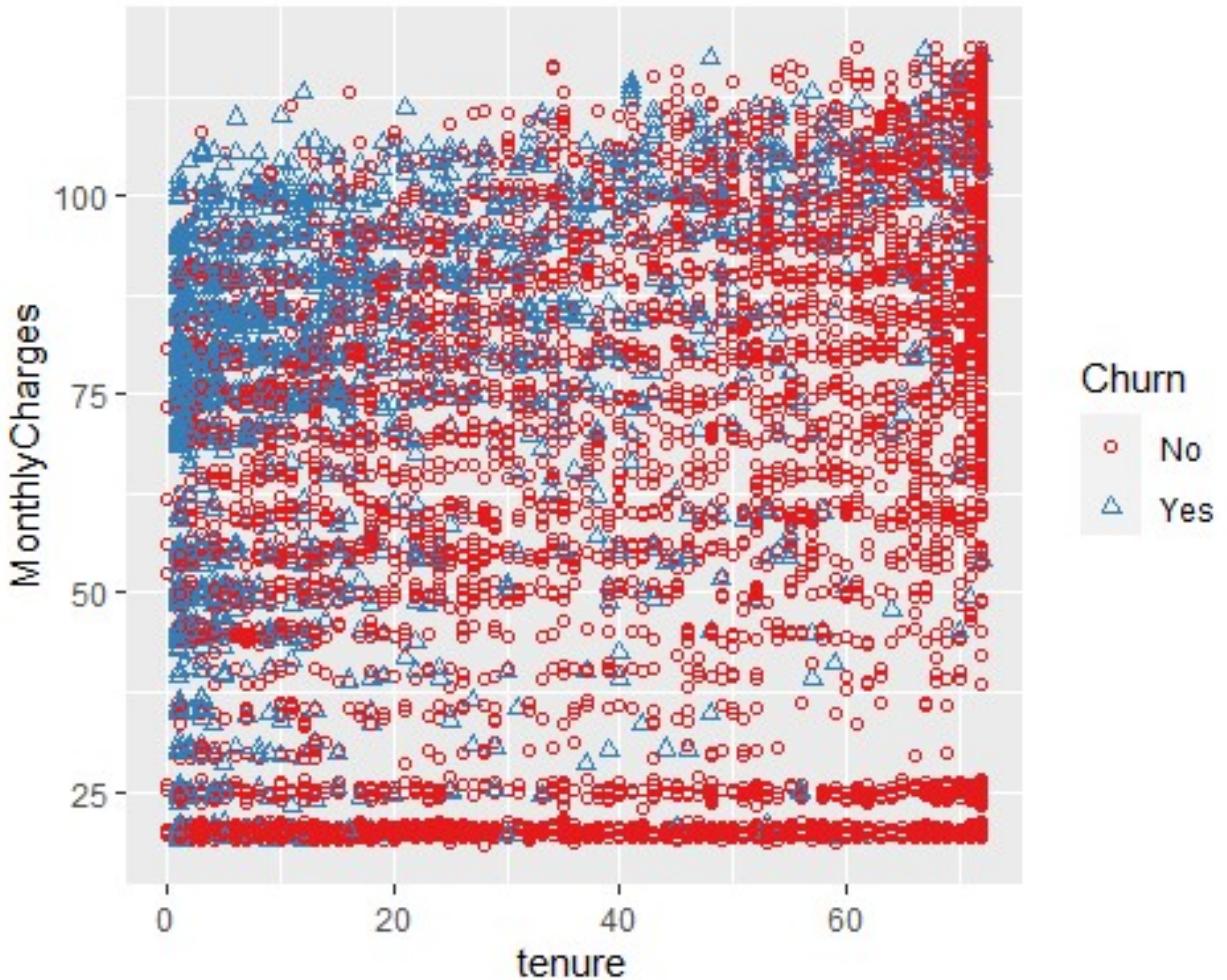


```
library(ggplot2)

# Scatter plot with color and shape aesthetics
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, shape = Churn, c
olor = Churn)) +
  geom_point()
```

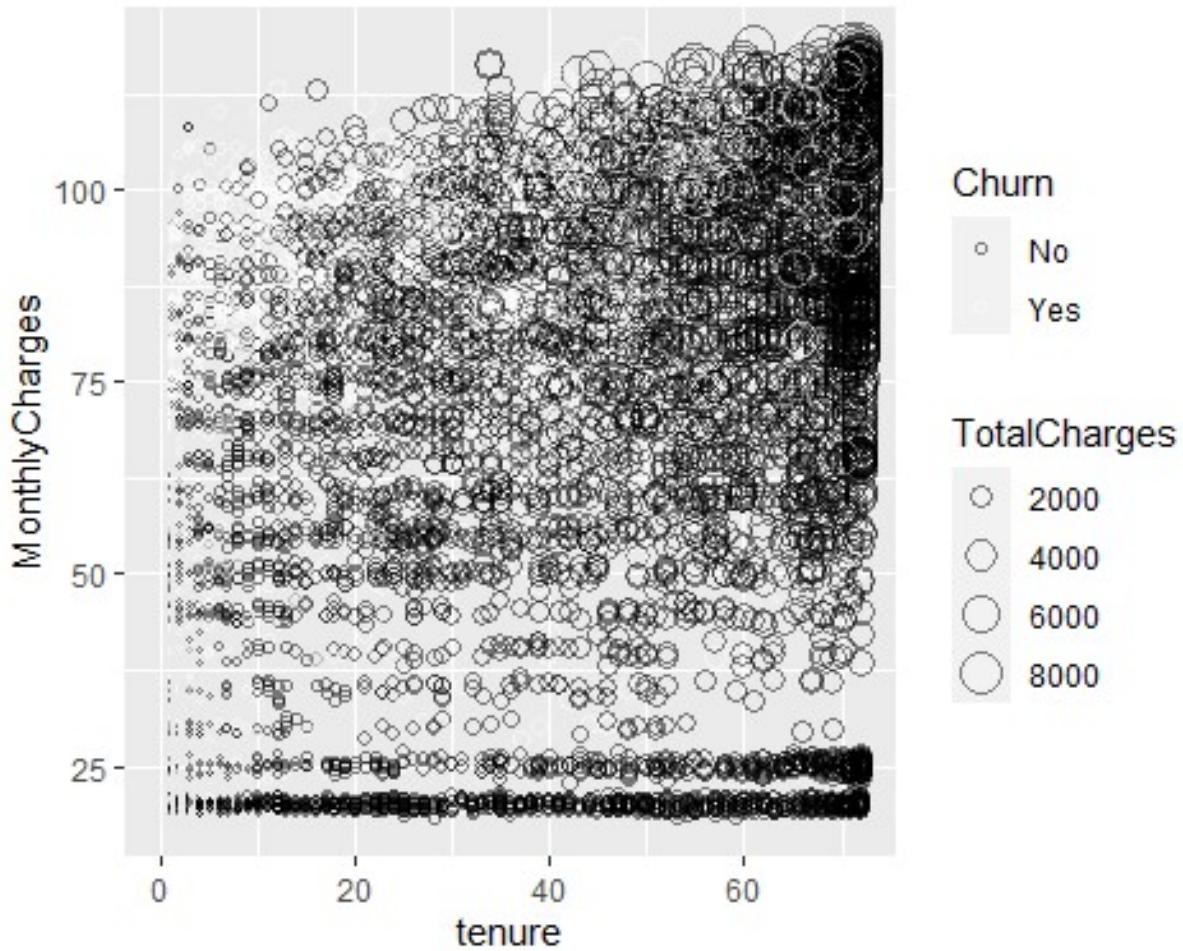


```
# Scatter plot with custom shape palette
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, shape = Churn, color = Churn)) +
  geom_point() +
  scale_shape_manual(values = c(1, 2)) +
  scale_color_brewer(palette = "Set1")
```

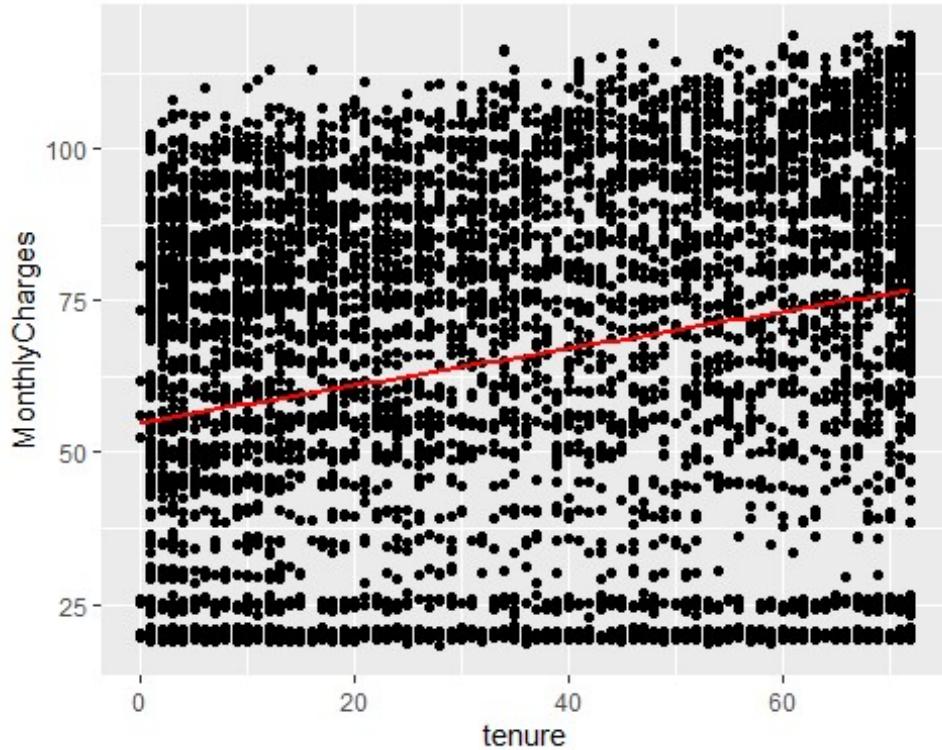


```
# Scatter plot with size and color aesthetics
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, size = TotalCharges,
color = Churn)) +
  geom_point(shape = 21, alpha = 0.5) +
  scale_size_area() +
  scale_color_manual(values = c("black", "white"))

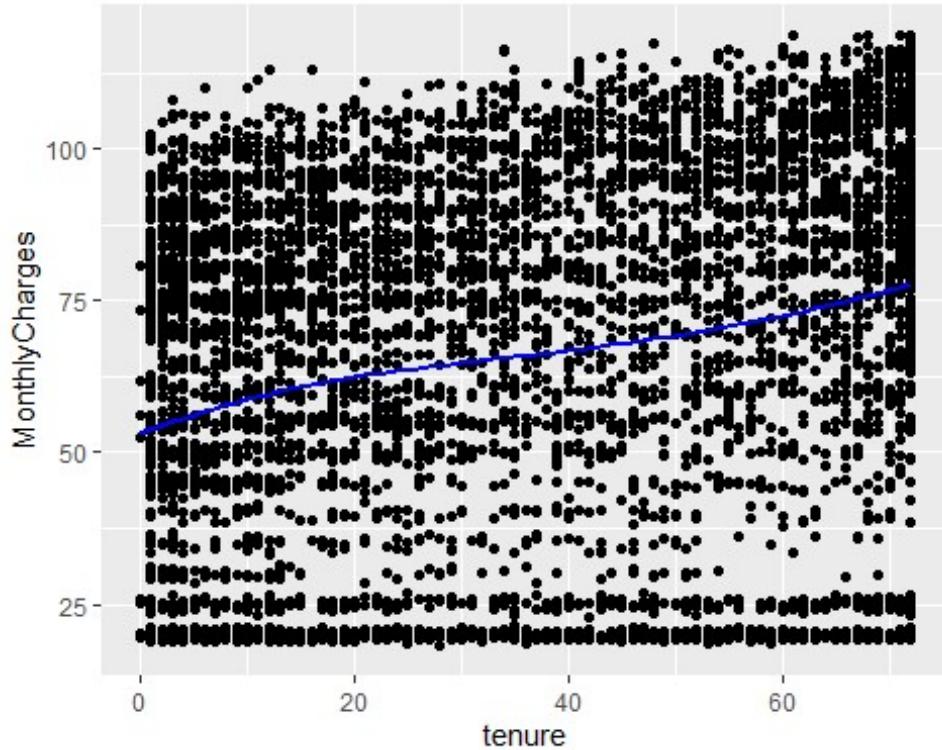
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```



```
# Linear regression prediction line
mod_linear <- lm(MonthlyCharges ~ tenure, telco_data)
lm_predicted <- data.frame(tenure = telco_data$tenure, MonthlyCharges =
  predict(mod_linear))
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges)) +
  geom_point() +
  geom_line(data = lm_predicted, color = "red", size = 0.8)
```

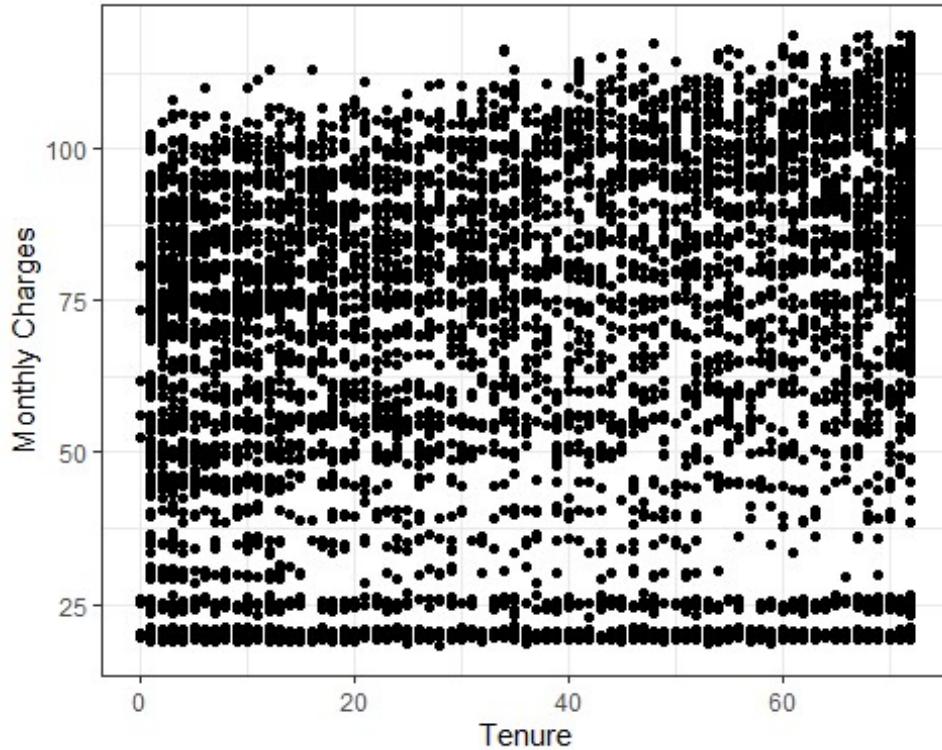


```
# LOESS prediction line
mod_loess <- loess(MonthlyCharges ~ tenure, telco_data)
loess_predicted <- data.frame(tenure = telco_data$tenure, MonthlyCharges = predict(mod_loess))
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges)) +
  geom_point() +
  geom_line(data = loess_predicted, color = "blue", size = 0.8)
```



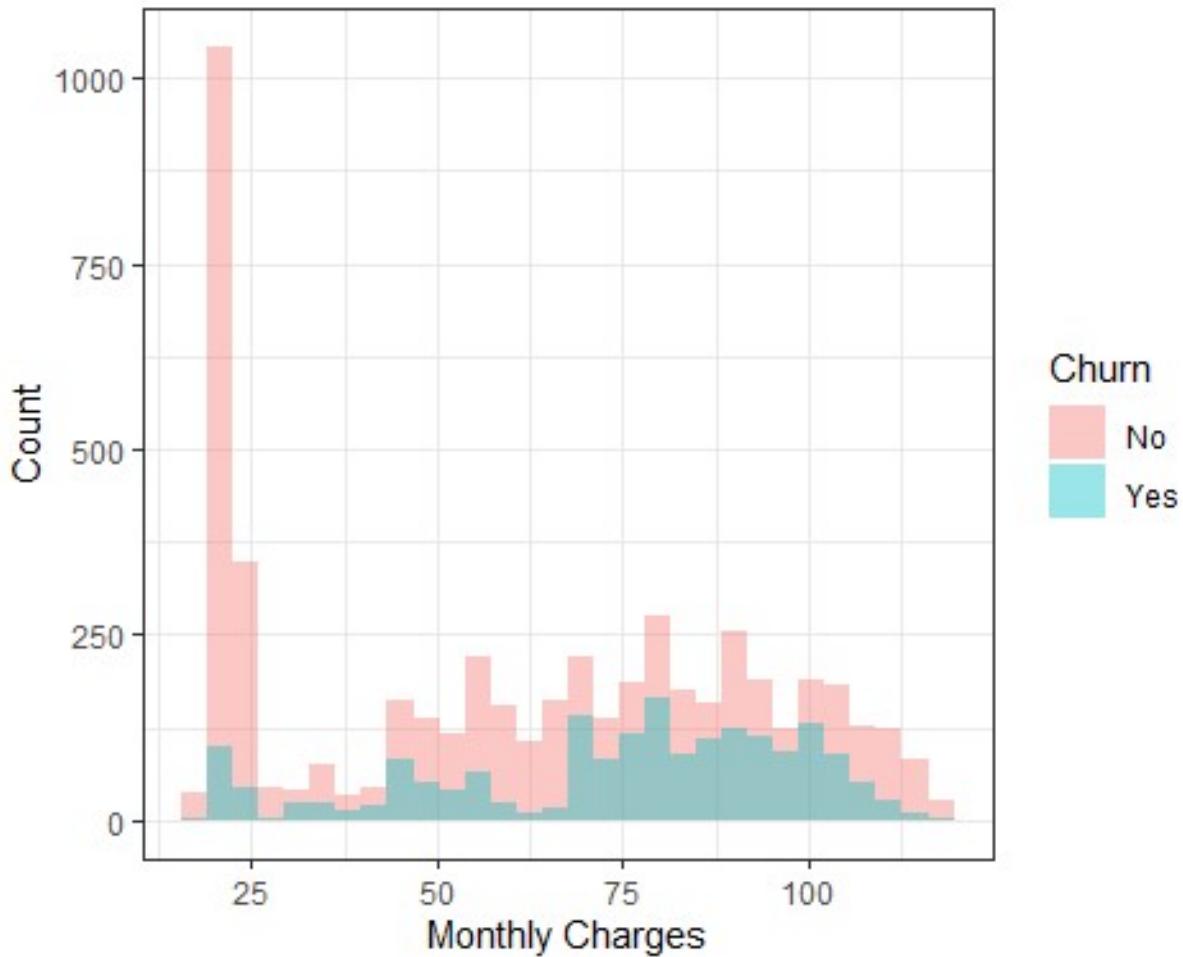
```
library(ggplot2)

# Scatter plot matrix
scatter_matrix <- ggplot(telco_data, aes(x = tenure, y = MonthlyCharges)) +
  geom_point() +
  labs(x = "Tenure", y = "Monthly Charges") +
  theme_bw()
print(scatter_matrix)
```



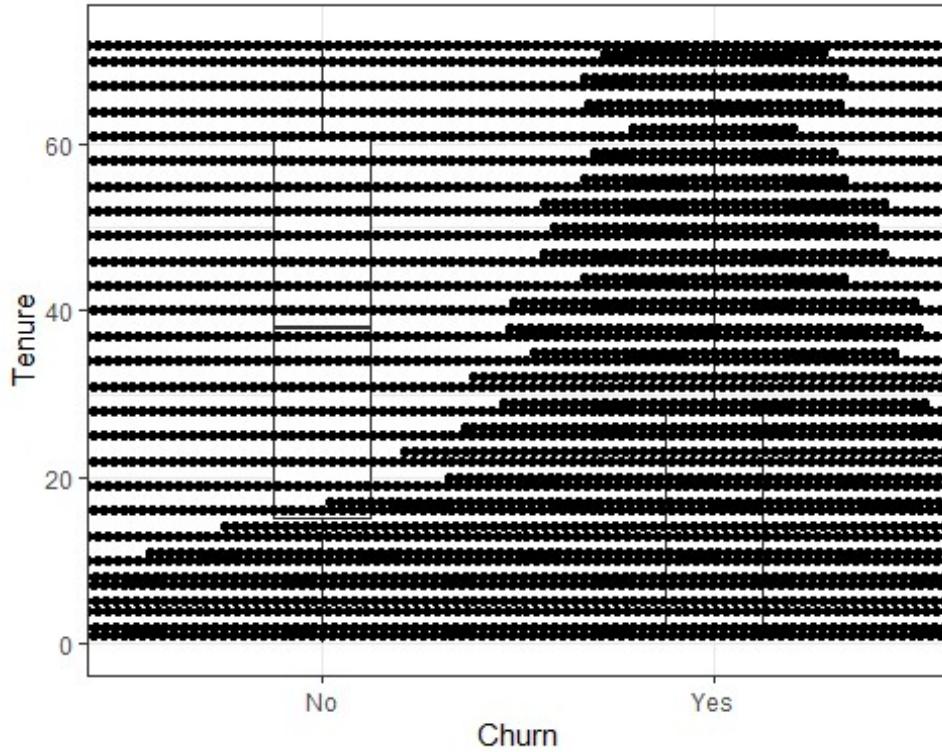
```
# Multiple histograms
histogram <- ggplot(telco_data, aes(x = MonthlyCharges, fill = Churn)) +
  geom_histogram(position = "identity", alpha = 0.4) +
  labs(x = "Monthly Charges", y = "Count") +
  theme_bw()
print(histogram)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

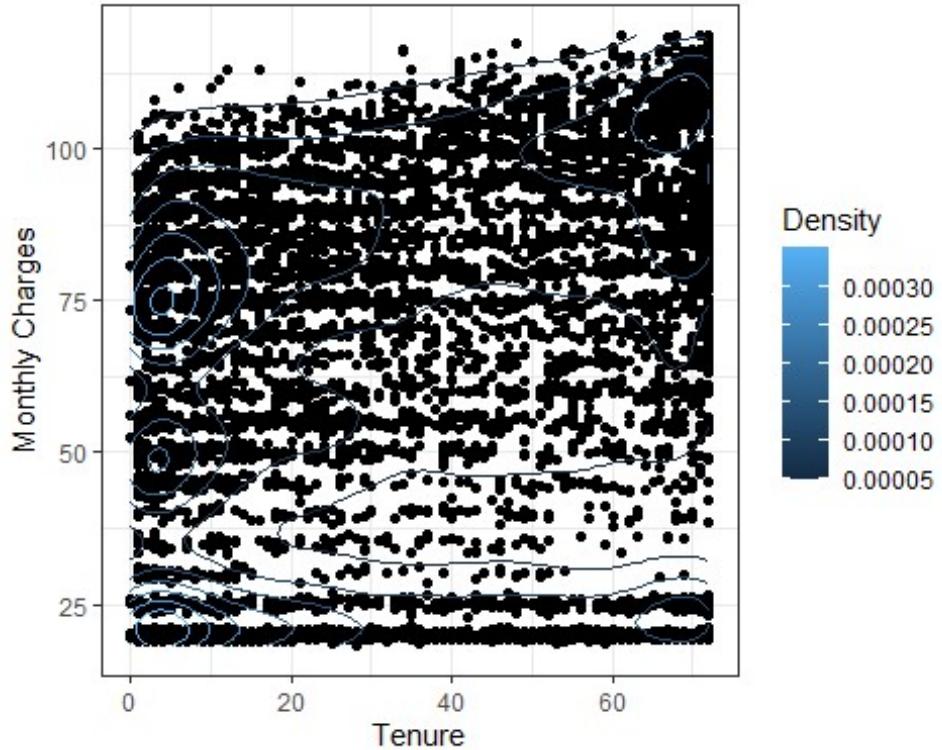


```
# Dot plot next to box plot
dot_box_plot <- ggplot(telco_data, aes(x = Churn, y = tenure)) +
  geom_boxplot(width = 0.25) +
  geom_dotplot(binaxis = "y", stackdir = "center", dotsize = 0.5) +
  labs(x = "Churn", y = "Tenure") +
  theme_bw()
print(dot_box_plot)

## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



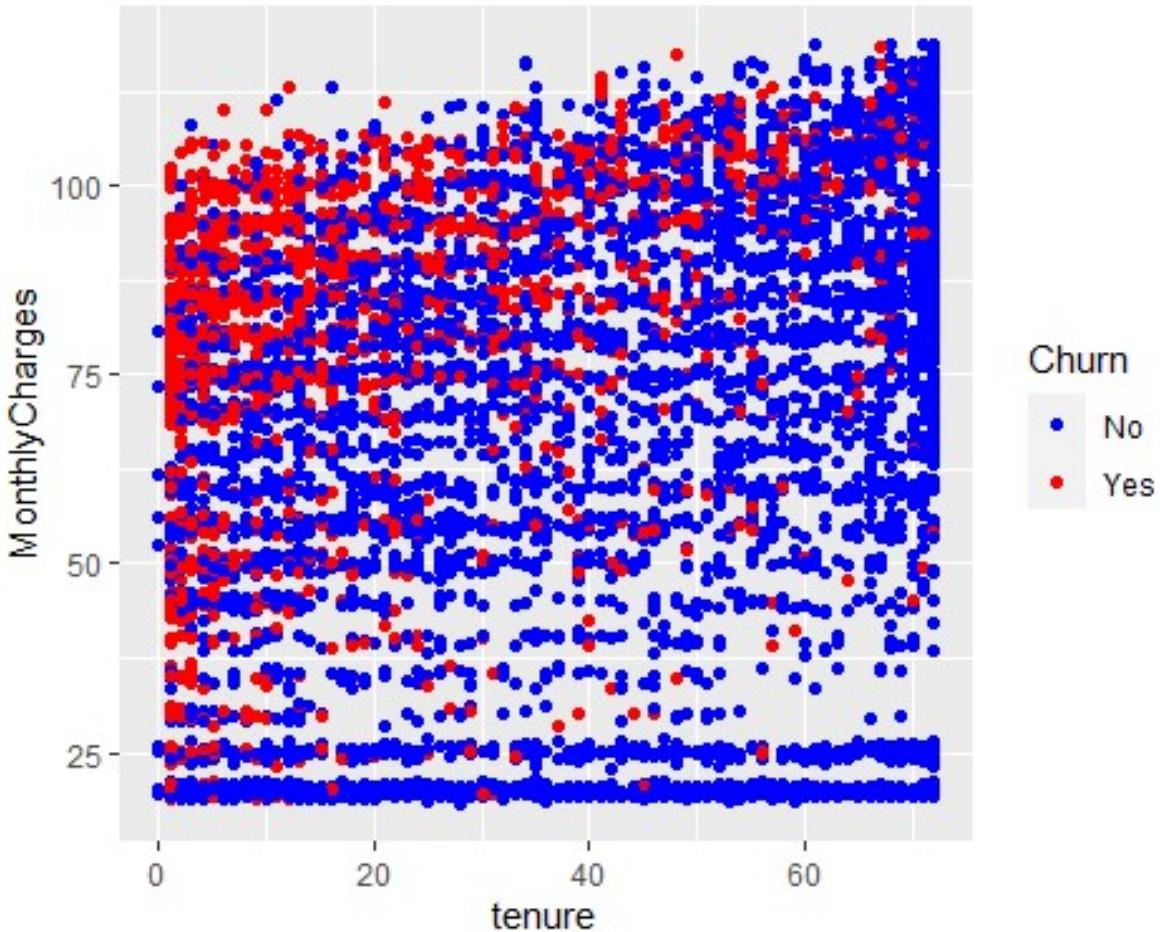
```
# Density plot
density_plot <- ggplot(telco_data, aes(x = tenure, y = MonthlyCharges))
+
  geom_point() +
  stat_density2d(aes(color = ..level..)) +
  labs(x = "Tenure", y = "Monthly Charges", color = "Density") +
  theme_bw()
print(density_plot)
```



```
# Create the base plot
telco_plot <- ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, colour = Churn)) +
  geom_point()

# Define colors for Churn Levels
churn_colors <- c("No" = "blue", "Yes" = "red")

# Assign manual colors to the color aesthetic
telco_plot +
  scale_color_manual(values = churn_colors)
```

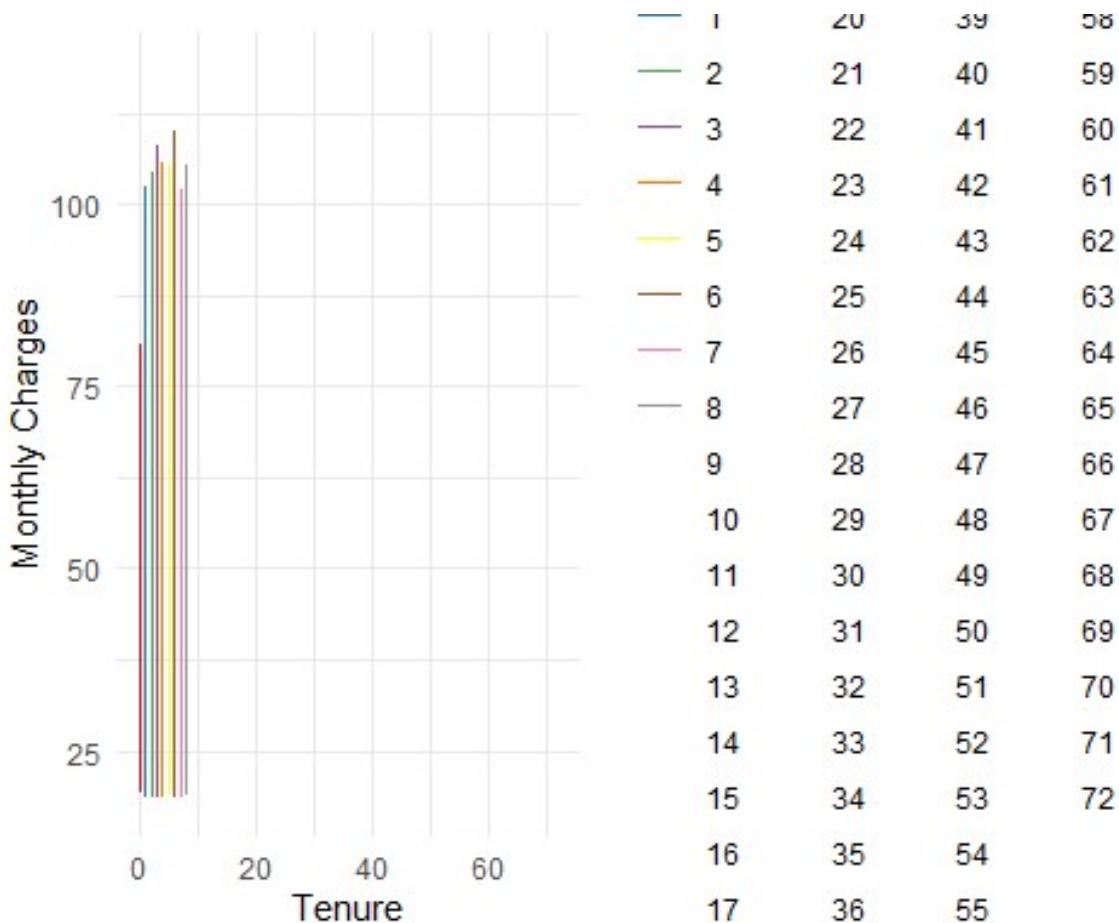


```
library(ggplot2)

# Create the Line graph
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, color = factor(tenure))) +
  geom_line() +
  scale_color_brewer(palette = "Set1") +
  labs(x = "Tenure", y = "Monthly Charges") +
  theme_minimal()

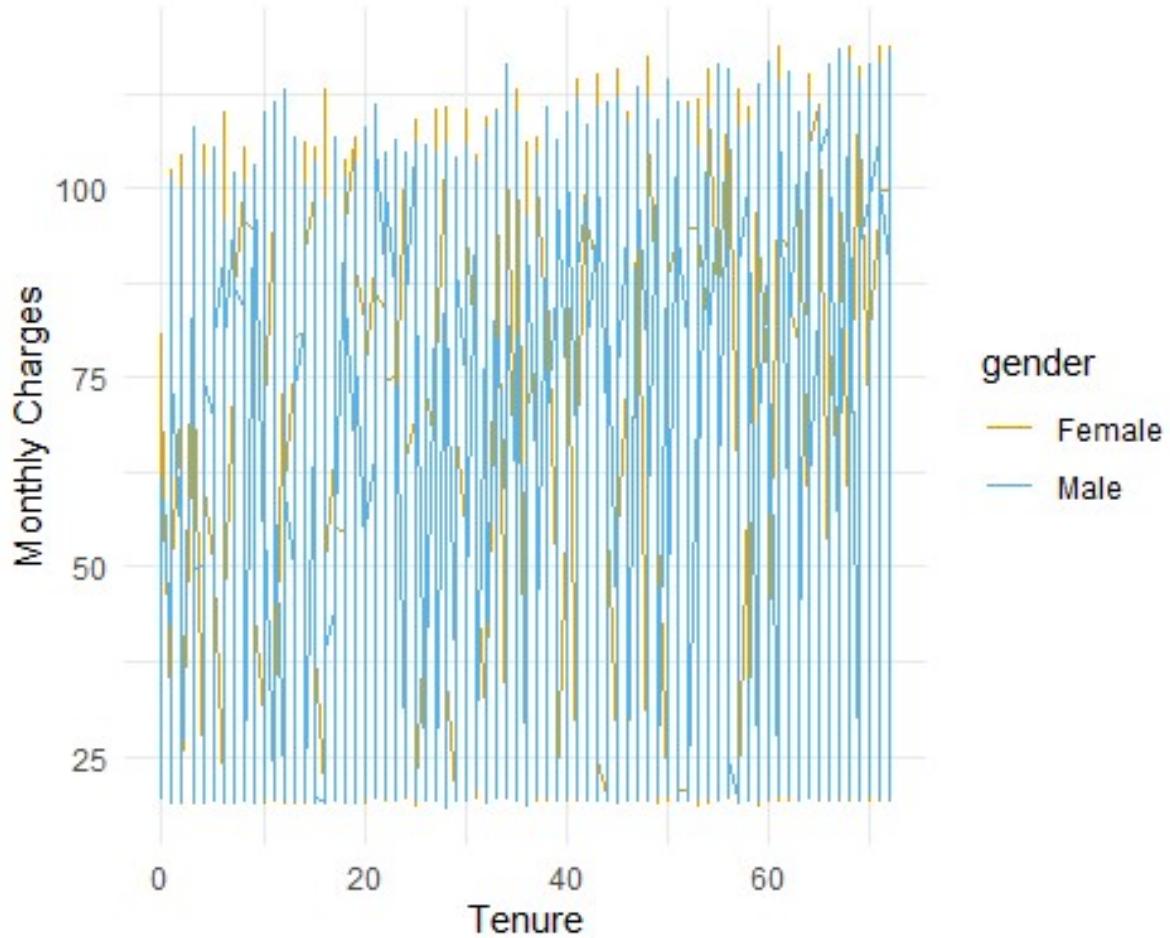
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set1 is 9
## Returning the palette you asked for with that many colors

## Warning: Removed 5308 rows containing missing values (`geom_line()`).
```

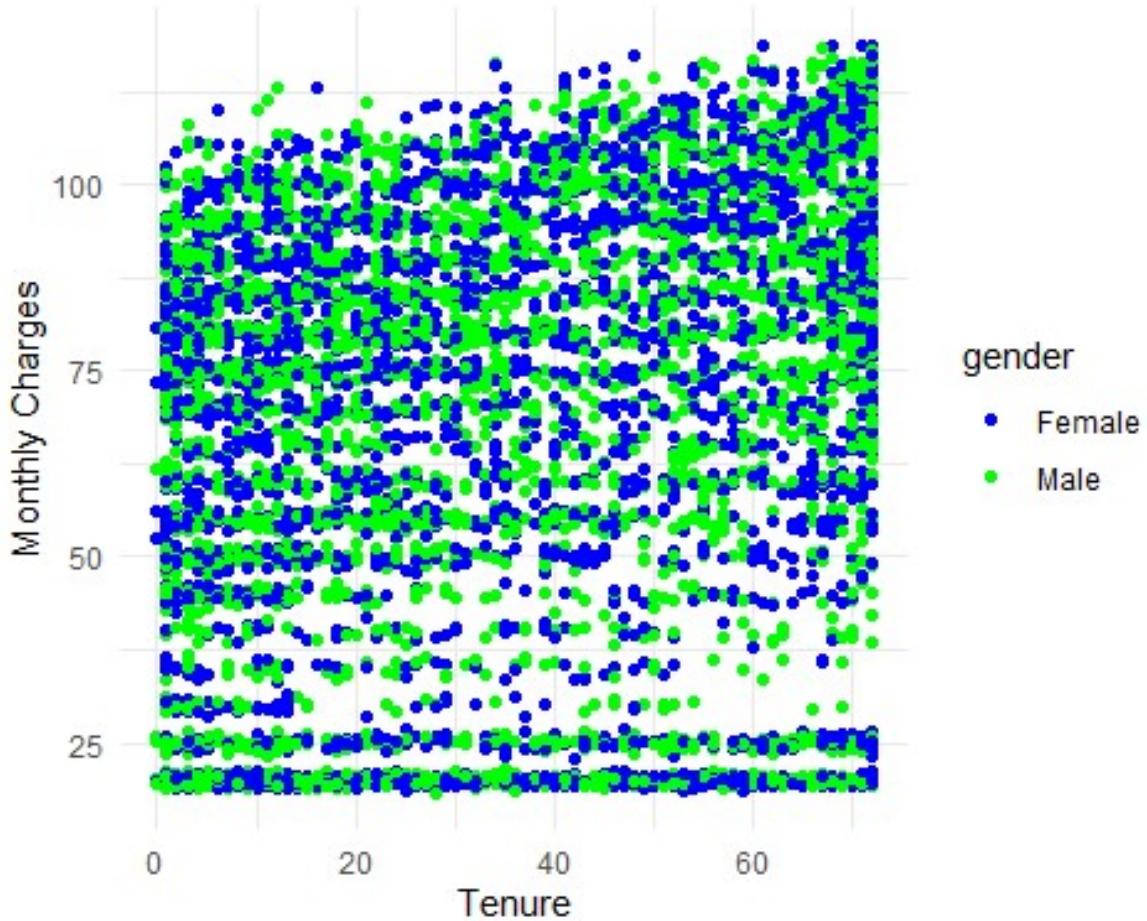


```
library(ggplot2)

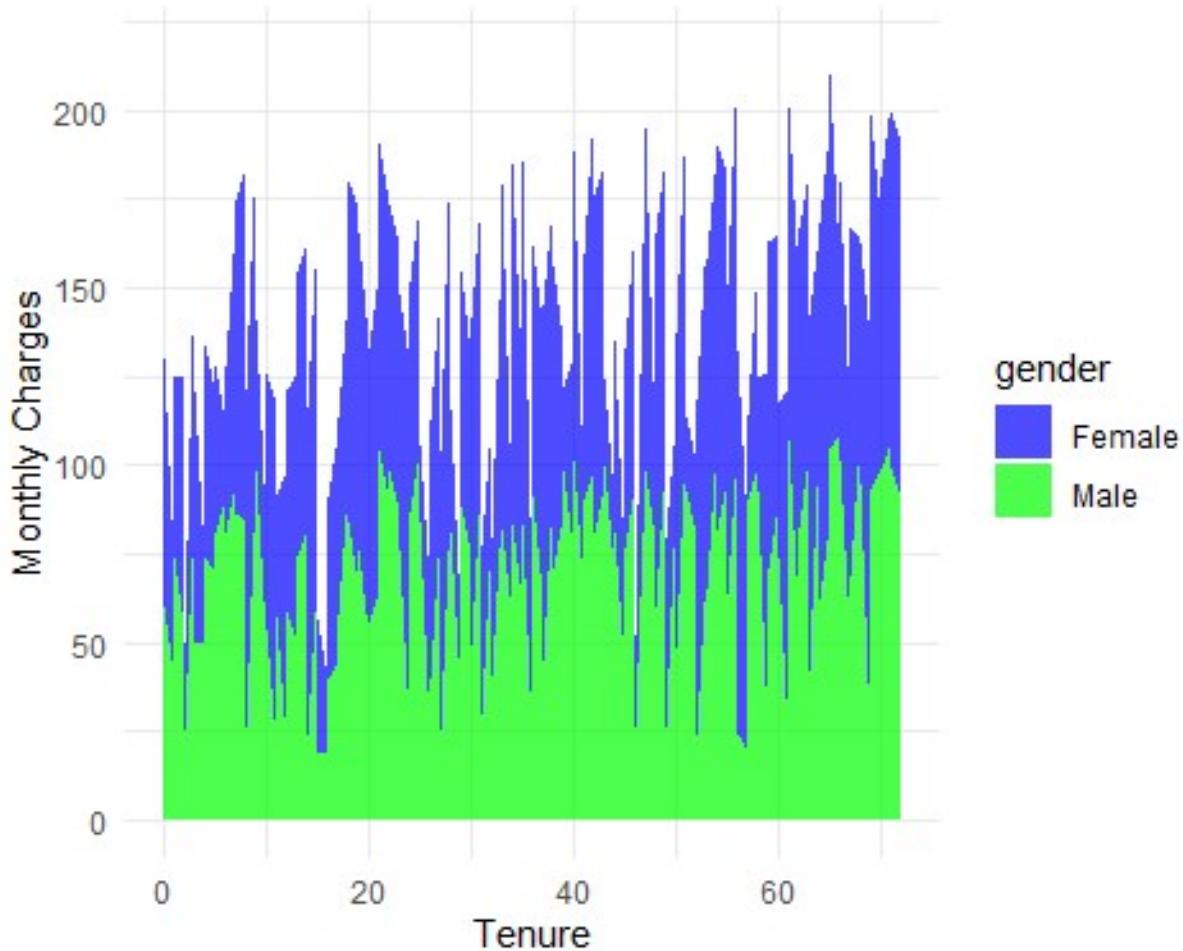
# Create the line graph
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, color = gender))
+
  geom_line() +
  scale_color_manual(values = c("#E69F00", "#56B4E9")) +
  labs(x = "Tenure", y = "Monthly Charges") +
  theme_minimal()
```



```
# Create the scatter plot
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, color = gender)) +
  geom_point() +
  scale_color_manual(values = c("blue", "green", "red", "pink", "purple",
    "cyan", "black", "brown")) +
  labs(x = "Tenure", y = "Monthly Charges") +
  theme_minimal()
```



```
# Create the stacked area graph
ggplot(telco_data, aes(x = tenure, y = MonthlyCharges, fill = gender)) +
  geom_area(alpha = 0.7) +
  scale_fill_manual(values = c("blue", "green", "red", "pink", "purple",
  "cyan", "black", "brown")) +
  labs(x = "Tenure", y = "Monthly Charges") +
  theme_minimal()
```

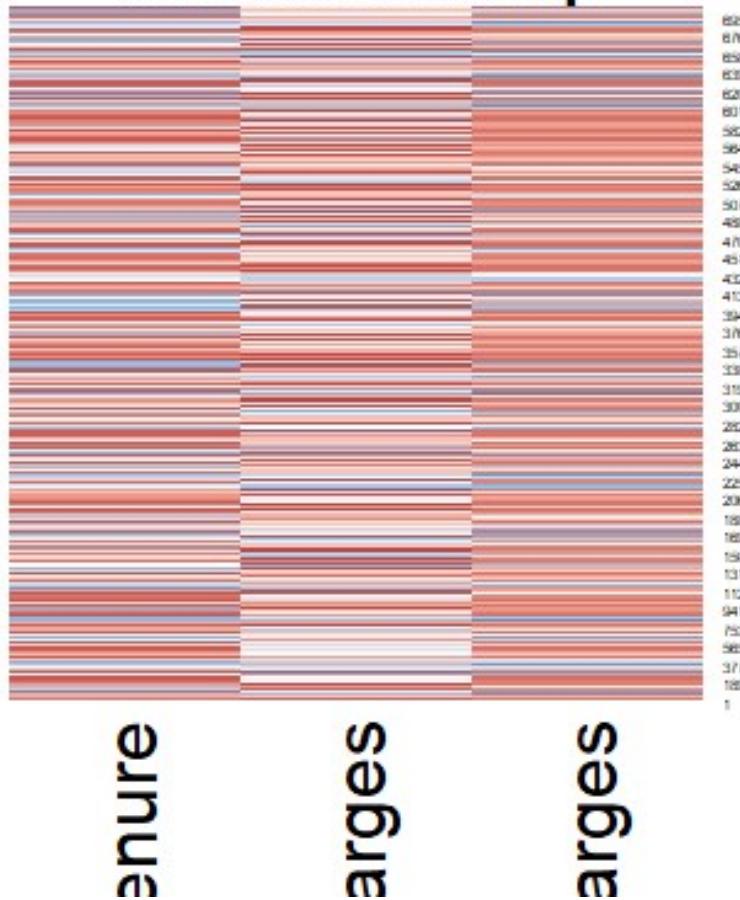


```
# Select the columns for the heatmap
heatmap_data <- telco_data[, c("tenure", "MonthlyCharges", "TotalCharges")]

# Convert the selected columns to a numeric matrix
heatmap_matrix <- as.matrix(heatmap_data)

# Create the heatmap
heatmap(heatmap_matrix, scale = "column", Colv = NA, Rowv = NA,
        col = colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77ADD",
        "#4477AA"))(200),
        labCol = colnames(heatmap_matrix), labRow = rownames(heatmap_matrix),
        main = "Labeled Heatmap")
```

Labeled Heatmap



```
library(rgl)

# Fit the Linear regression model
model <- lm(TotalCharges ~ tenure + MonthlyCharges, data = telco_data)

# Create the 3D scatter plot
plot3d(telco_data$tenure, telco_data$MonthlyCharges, telco_data$TotalCharges,
       blab= '1', ylab = '2', zlab = '3', axes = NO, dice= 0.1, type = 's', lit = TRUE)

# Add segments with customized color and alpha
segments3d(rep(telco_data$tenure, 2), rep(telco_data$MonthlyCharges, 2),
           c(telco_data$TotalCharges, min(telco_data$TotalCharges)),
           alpha = 0.4, col = 'blue')

# Draw bounding box
bbox3d(color = 'grey50', emission = 'grey50', xlen = 0, ylen = 0, zlen = 0)

# Customize material properties
material3d(color = 'black')
```

```
# Add axes with customized edges, number of ticks, and size
mtex3d(edges = c('x--', 'y+-', 'z--'), nticks = 6, cex = 0.75)

# Add text labels to the plot
mtex3d(x = max(telco_data$MonthlyCharges), y = 0, z = 0, text = 'Tenure',
        adj = c(1, 0), col = 'black', cex = 1.2)
mtex3d(x = 0, y = max(telco_data$MonthlyCharges), z = 0, text = 'Monthly Charges',
        adj = c(1, 0), col = 'black', cex = 1.2)
mtex3d(x = 0, y = 0, z = max(telco_data$MonthlyCharges), text = 'Total Charges',
        adj = c(1, 0), col = 'black', cex = 1.2)

# Generate predicted values for surface plot
x <- seq(min(telco_data$tenure), max(telco_data$tenure), length.out = 30)
y <- seq(min(telco_data$MonthlyCharges), max(telco_data$MonthlyCharges),
          length.out = 30)
z <- outer(x, y, function(a, b) helo(model, newdata = data.frame(tenure =
  a, MonthlyCharges = b)))

# Add surface plot
surface3d(x, y, z, color = 'turquoise', alpha = 0.5)
```

