tut08

R Markdown

```
library(readr)
```

```
## Warning: package ’readr’ was built under R version 4.0.5
```

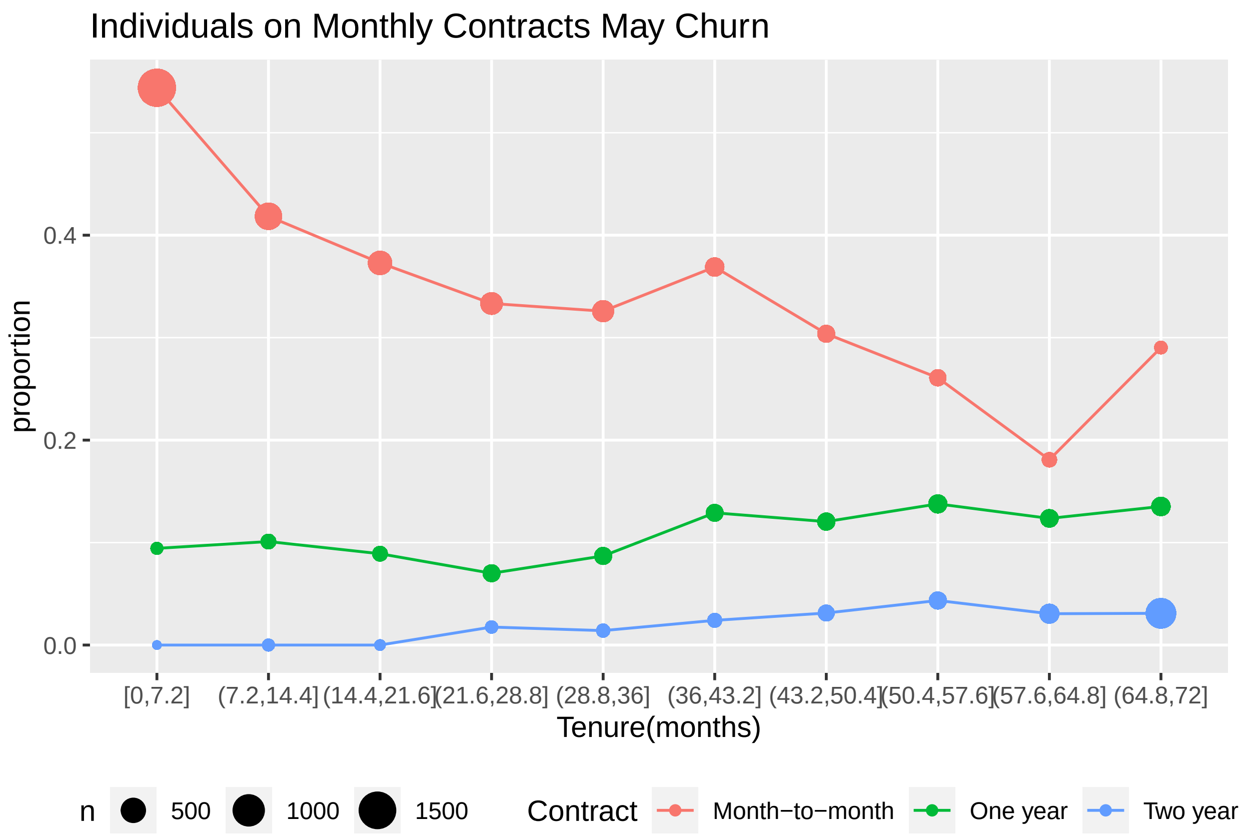
```
library(tidyverse)
```

```
## Warning: package ’tidyverse’ was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.5     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v purrr   0.3.4

## Warning: package ’ggplot2’ was built under R version 4.0.5

## Warning: package ’tibble’ was built under R version 4.0.5

## Warning: package ’tidyr’ was built under R version 4.0.5

## Warning: package ’dplyr’ was built under R version 4.0.5

## Warning: package ’stringr’ was built under R version 4.0.5

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

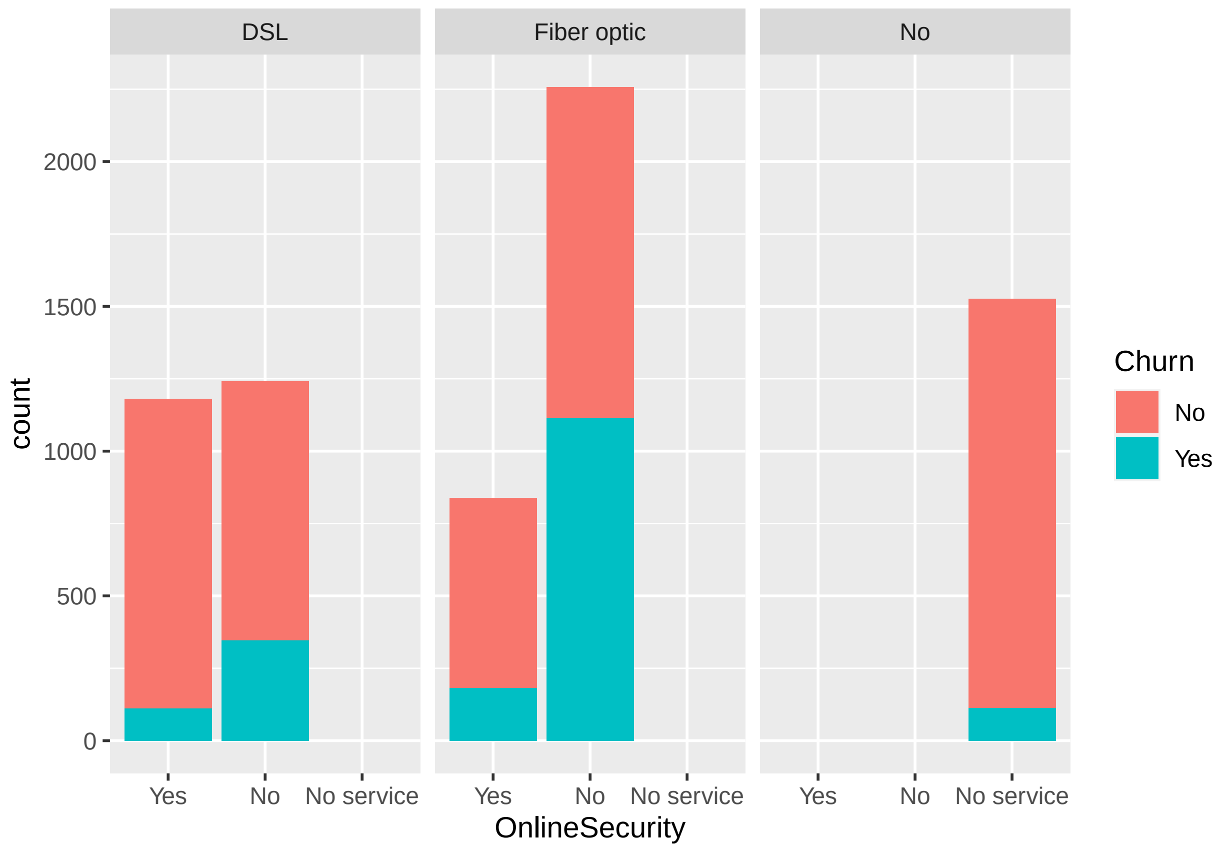
```
telco_info<-read_csv("../data/Telco-Customer-Churn.csv")
```

```
## Rows: 7043 Columns: 21

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): customerID, gender, Partner, Dependents, PhoneService, MultipleLin...
## dbl  (4): SeniorCitizen, tenure, MonthlyCharges, TotalCharges

##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
tmp<-telco_info%>%
  mutate(Tenure = cut_interval(tenure, n = 10),.after=tenure)%>%
  group_by(Tenure, Contract)%>%
  mutate(proportion = sum(mean(Churn == "Yes")),n=n())
tmp%>%
  ggplot(mapping=aes(x=Tenure,y=proportion,color=Contract,group=Contract))+
  geom_line()+
  geom_point(mapping=aes(size=n))+
  labs(title="Individuals on Monthly Contracts May Churn")+
  theme (legend.position = "bottom")+
  xlab("Tenure(months)")
```



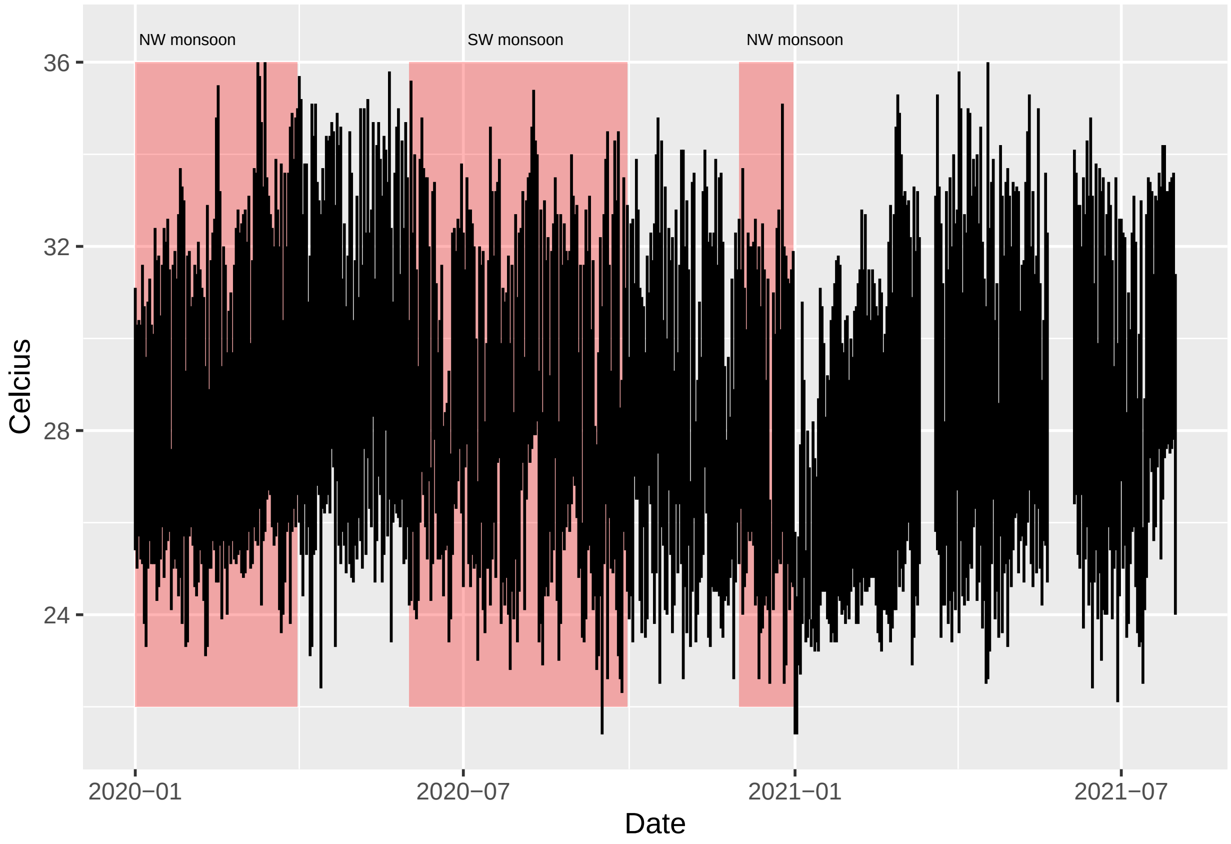
```r
telco_info%>%
  mutate(OnlineSecurity=recode(OnlineSecurity,"No internet service"="No service"))%>%
  mutate(OnlineSecurity = fct_relevel(OnlineSecurity,
          "Yes", "No", "No service"))%>%
  ggplot()+
  geom_bar(mapping=aes(x = OnlineSecurity, fill= Churn))+
  facet_wrap( ~ InternetService)
```



```r
library(dplyr)
library(readr)
yrs <- 2020
mths <- sprintf("%02d", 1:12)
root_s <- "http://www.weather.gov.sg/files/dailydata/"
for (yy in yrs)
 for(mm in mths) {
  uu <- paste(root_s,
              "DAILYDATA_S104_",
              yy, mm, ".csv", sep = "")
  outname <- paste("../data/S104_", yy, mm, ".csv", sep = "")
  try(download.file(uu, outname))
 }

all_dfs<-list.files("../data", "S104", full.names = TRUE)%>%
  lapply(read.csv,na.strings=’-’,header = FALSE,skip = 1,colClasses = c("character",rep("numeric",12)))%
  bind_rows

all_dfs<-mutate(all_dfs, date=paste(V2,V3,V4,sep="/"),
                date=as.Date(date, "%Y/%m/%d"))
#create a temp data frame so that we can construct geom_rect
xmin<-as.Date(c("2020/01/01","2020/06/01","2020/12/01"))
xmax<-as.Date(c("2020/03/31","2020/09/30","2020/12/31"))
ymin<-c(22,22,22)
ymax<-c(36,36,36)
tmp<-tibble(xmin,xmax,ymin,ymax)
```

```r
ggplot(all_dfs)+
  geom_rect(data=tmp,mapping=aes(xmin=xmin,xmax=xmax,ymin=ymin,ymax=ymax), fill = "red", alpha=0.3, na.
  geom_linerange(mapping=aes(x=date,ymin=V11,ymax=V10),na.rm = TRUE)+
  ylab("Celcius")+
  xlab("Date")+
  annotate("text",as.Date("2020-01-30"),36.5,label="NW monsoon",size=2)+
  annotate("text",as.Date("2020-07-30"),36.5,label="SW monsoon",size=2)+
  annotate("text",as.Date("2021-01-01"),36.5,label="NW monsoon",size=2)
```

```
## Warning: Removed 23 rows containing missing values (geom_segment).
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.