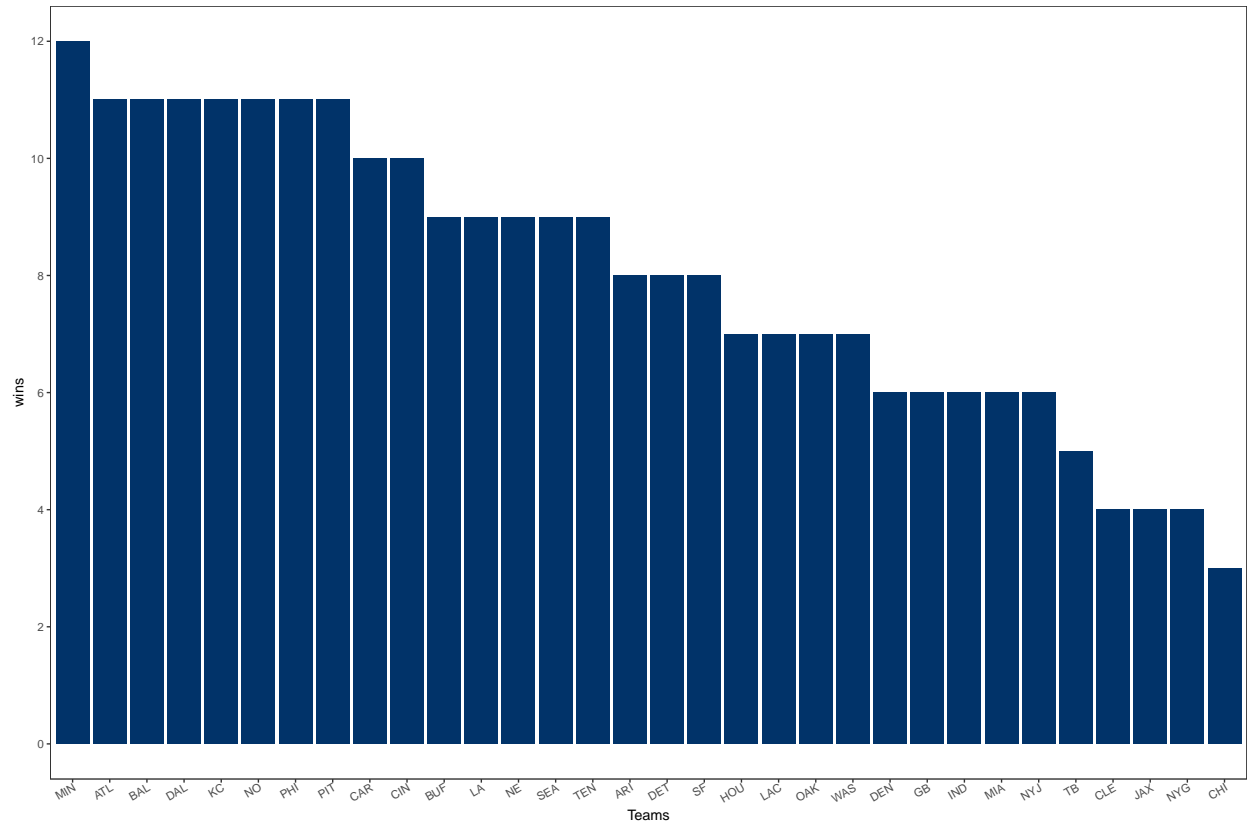
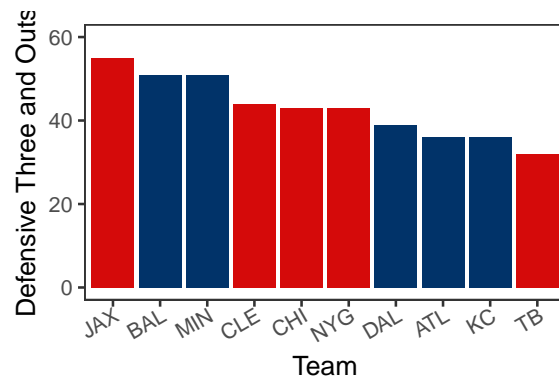
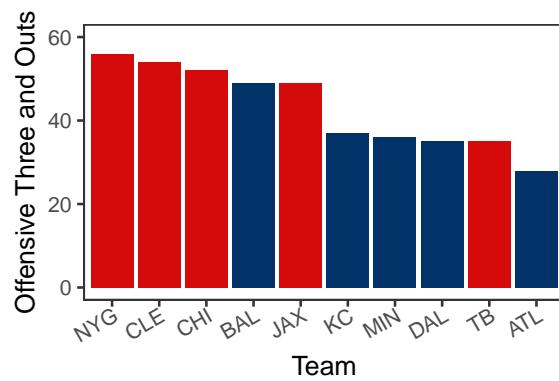
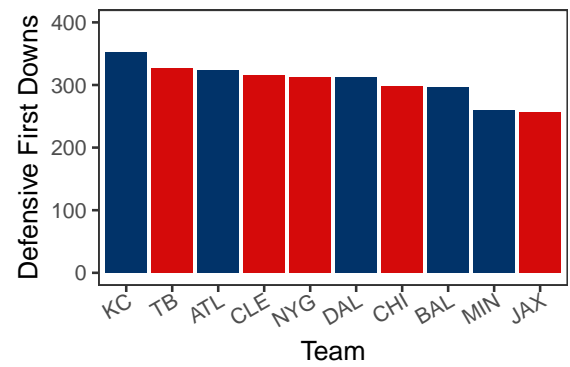
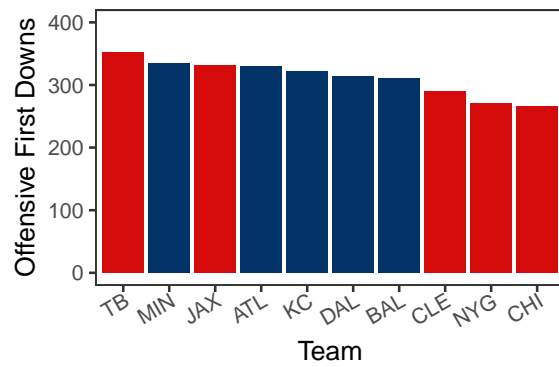
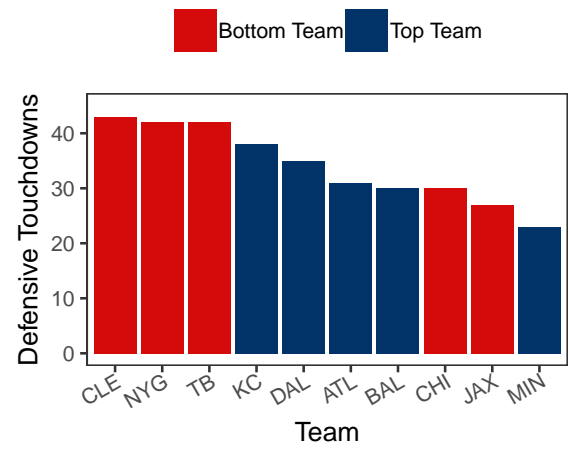
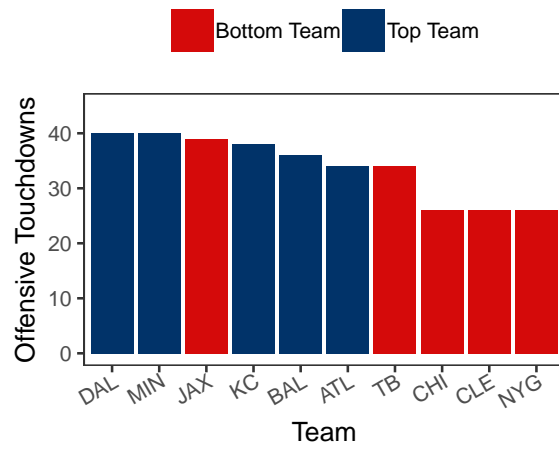


Final Project

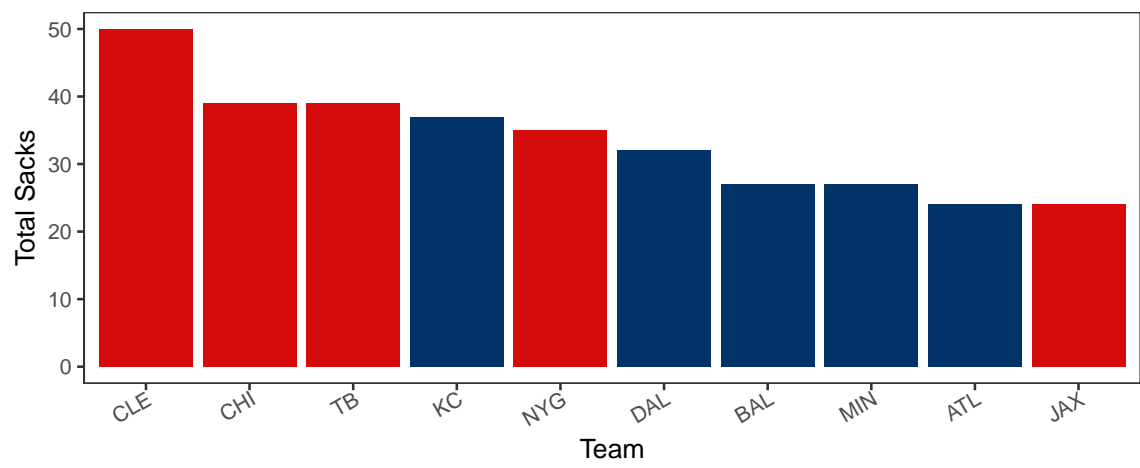
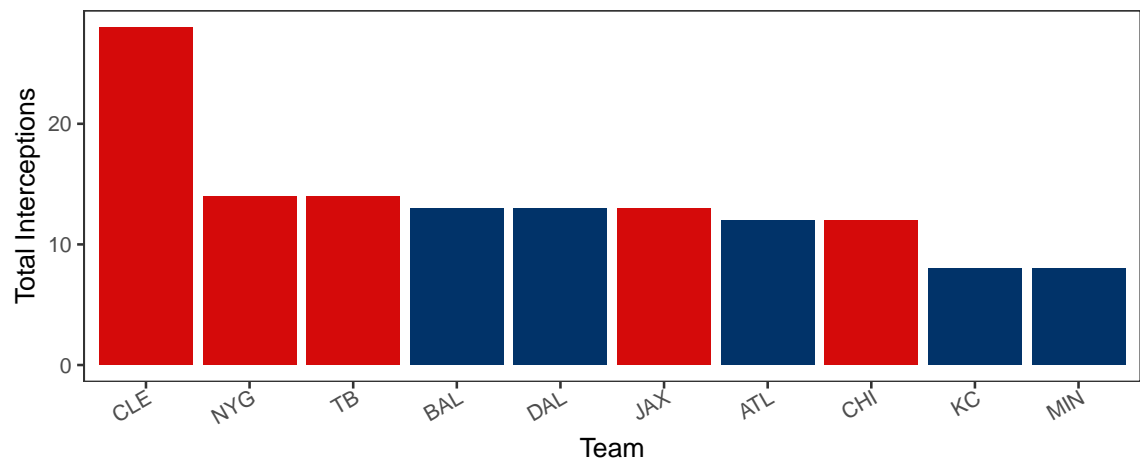
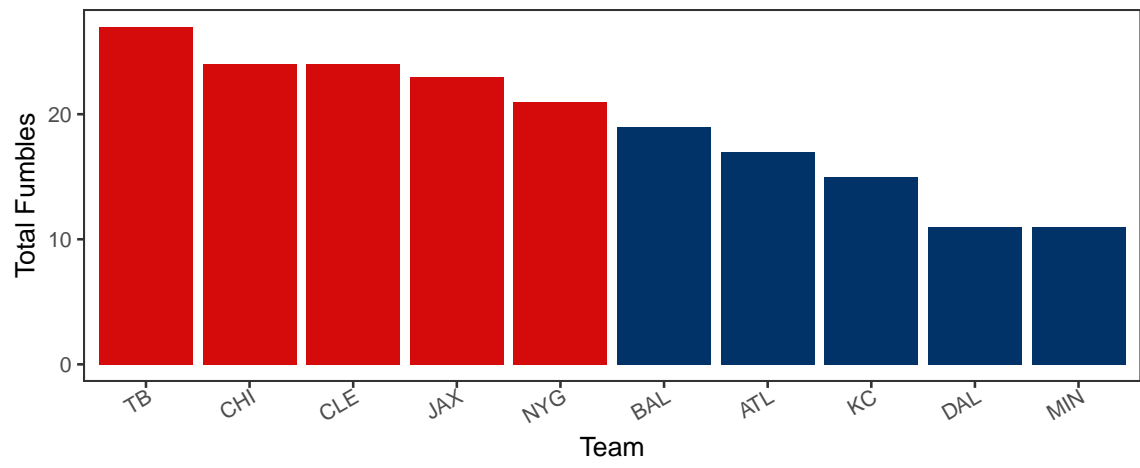
Wins By Team



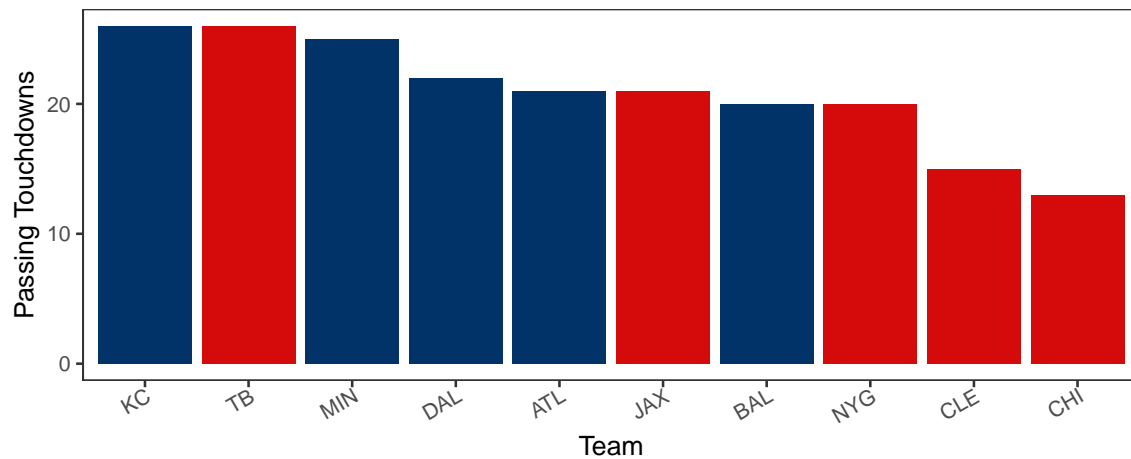
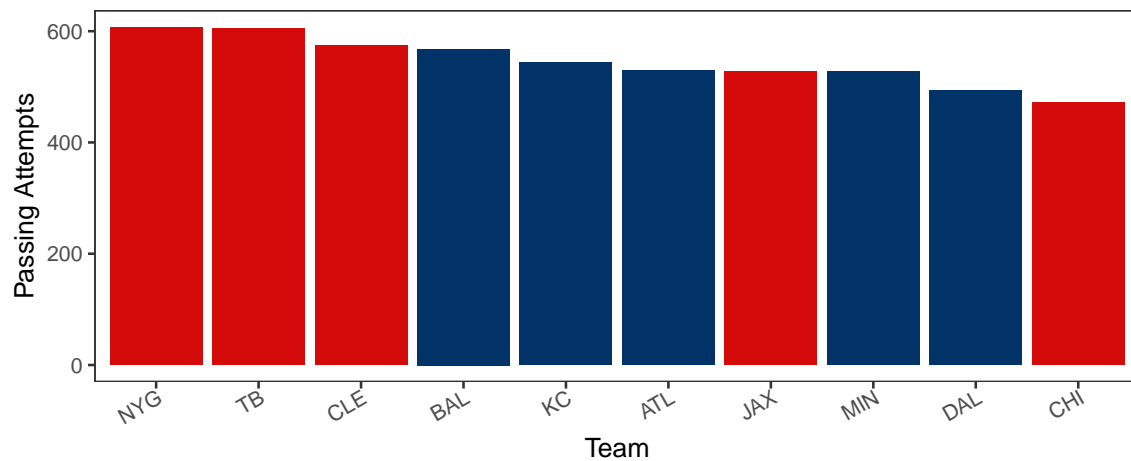
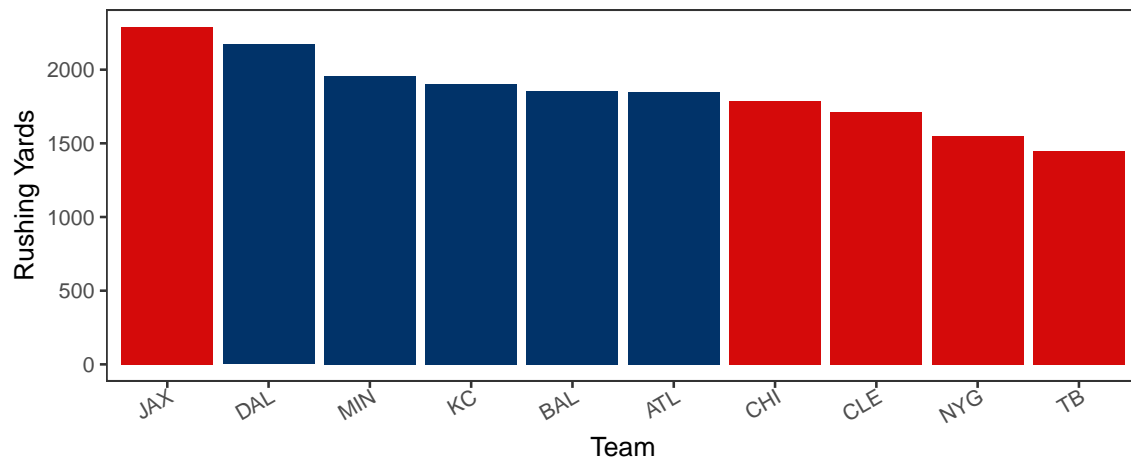
Touchdowns, Firstdowns and Three and Outs



Protect the Rock



Extras



To start I first looked at the records for all of the teams, and I determined I should look through the available stats to see if there were any trends that seemed meaningful between the top performing teams, and the bottom ones.

I decided that these three stats on offense and defense could be clear indicators of a team that is likely to win games, as a team that has more first downs would have more offensive opportunities, and vice versa on

defense. Clearly a team with more touchdowns and limits the touchdowns of their opponents will win more games, and minimizing three and outs on offense or maximizing them on defense is the counterpoint to first downs. A three and out is defined as any drive that did not contain a first down and ended in a punt.

Interestingly the offensive stats did have some correlation with wins, even though Jacksonville did have the third most TDs and FDs, and TB had the most FDs and least three and outs by and large the top and bottom teams occupied opposite ends of the plots. Defensive touchdowns followed a similar pattern, with the top and bottom teams clustered, with a few exceptions. Suprisingly for me, seemingly strong indicators of a weak or strong defense did not seem to have any correlation with being in the top or bottom 5 teams. Time to dig deeper.

Here as we dig deeper a pattern begins to show, teams that do better do a better job of protecting the ball, and their quarterback. This makes perfect sense as turnovers give the other team more chances to score, and limit your own opportunities. Sacks can cause multiple potential issues for a team. For one, a sack is generally accompanied by a loss of yards, which can make it more difficult to get a firstdown, and in addition to that when a QB is hit a lot in a game, their fatigue can quickly increase, and any rough collision could cause soreness that could lead to poor play down the line.

These last 3 plots I find to be a bit fun, because it shows some potentially misleading stats. The plots showing that rushing yards with the winning teams (generally) being above the losing teams could mean that having a very good ground game is essential to winning football games, but that is not truly the case, it is a side effect of strategy. When a team is ahead in the games, they tend to pass more, and rush less. This is also shown in the passing attempts plot. The duality of this struggle to catch back up by increasing passing attempts is shown in the final plot, indicating that for most of these teams, the increase in attempts is not also linked to an increased amount of scoring through the air.

```

#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
#   http://shiny.rstudio.com/
#

library(shiny)
library(tidyverse)
library(sqldf)
library(scales)
drive<-readxl::read_xlsx('NFL.xlsx',sheet='Drive')
game<-readxl::read_xlsx('NFL.xlsx',sheet='Game')
play<-readxl::read_xlsx('NFL.xlsx',sheet='Player-Play')
game_stats<-readxl::read_xlsx('NFL.xlsx',sheet='Stats')
#rsconnect::setAccountInfo(name='shnipes', token='6A6195725697F3D8AB463C0B34FC3209', secret='QGjr4Mn1z0

teamWin <- group_by(game, winner) %>%
  summarise(wins = n())
teamWin<-teamWin[order(teamWin$wins,decreasing=T),]
selectedTeam = teamWin$winner[1]
teamWin<-teamWin[order(teamWin$winner,decreasing=F),]

uniquestat<-game_stats$cat

play<- merge(play,game,by="gameid")
play<- merge(play,game_stats,by='cat')
Pri=list()
Sec=list()

Pri['ARI']=rgb(151/255,35/255,63/255)
Pri['ATL']=rgb(167/255,25/255,48/255)
Pri['BAL']=rgb(26/255,25/255,95/255)
Pri['BUF']=rgb(0/255,51/255,141/255)
Pri['CAR']=rgb(0/255,133/255,202/255)
Pri['CHI']=rgb(200/255,56/255,3/255)
Pri['CIN']=rgb(0/255,85/255,100/255)
Pri['CLE']=rgb(49/255,29/255,0/255)
Pri['DAL']=rgb(0/255,53/255,148/255)
Pri['DEN']=rgb(251/255,79/255,20/255)
Pri['DET']=rgb(0/255,118/255,182/255)
Pri['GB']=rgb(24/255,48/255,40/255)
Pri['HOU']=rgb(167/255,25/255,48/255)
Pri['IND']=rgb(0/255,44/255,95/255)
Pri['JAX']=rgb(215/255,162/255,42/255)
Pri['KC']=rgb(255/255,184/255,28/255)
Pri['LA']=rgb(134/255,109/255,75/255)
Pri['LAC']=rgb(255/255,194/255,14/255)
Pri['MIA']=rgb(242/255,106/255,36/255)
Pri['MIN']=rgb(255/255/255, 198/255/255,47/255)
Pri['NE']=rgb(198/255,12/255,48/255)

```

```

Pri['NO']=rgb(211/255,188/255,141/255)
Pri['NYG']=rgb(163/255,13/255,45/255)
Pri['NYJ']=rgb(200/255,200/255,200/255)
Pri['OAK']=rgb(165/255,172/255,175/255)
Pri['PHI']=rgb(165/255,172/255,175/255)
Pri['PIT']=rgb(255/255,182/255,18/255)
Pri['SEA']=rgb(173/255,153/255,93/255)
Pri['SF']=rgb(165/255,172/255,175/255)
Pri['TB']=rgb(213/255,10/255,10/255)
Pri['TEN']=rgb(68/255,149/255,209/255)
Pri['WAS']=rgb(255/255,182/255,18/255)
Sec['ARI']=rgb(255/255,182/255,18/255)
Sec['ATL']=rgb(0/255,0/255,0/255)
Sec['BAL']=rgb(0/255,0/255,0/255)
Sec['BUF']=rgb(198/255,12/255,48/255)
Sec['CAR']=rgb(16/255,24/255,32/255)
Sec['CHI']=rgb(11/255,22/255,42/255)
Sec['CIN']=rgb(0/255,0/255,0/255)
Sec['CLE']=rgb(255/255,60/255,0/255)
Sec['DAL']=rgb(134/255,147/255,151/255)
Sec['DEN']=rgb(0/255,34/255,68/255)
Sec['DET']=rgb(176/255,183/255,188/255)
Sec['GB']=rgb(255/255,184/255,28/255)
Sec['HOU']=rgb(3/255,32/255,47/255)
Sec['IND']=rgb(162/255,170/255,173/255)
Sec['JAX']=rgb(0/255,103/255,120/255)
Sec['KC']=rgb(227/255,24/255,55/255)
Sec['LA']=rgb(0/255,34/255,68/255)
Sec['LAC']=rgb(0/255,42/255,94/255)
Sec['MIA']=rgb(0/255,87/255,120/255)
Sec['MIN']=rgb(79/255,38/255,131/255)
Sec['NE']=rgb(0/255,34/255,68/255)
Sec['NO']=rgb(16/255,24/255,31/255)
Sec['NYG']=rgb(1/255,35/255,82/255)
Sec['NYJ']=rgb(0/255,63/255,45/255)
Sec['OAK']=rgb(166/255,174/255,176/255)
Sec['PHI']=rgb(0/255,76/255,84/255)
Sec['PIT']=rgb(16/255,24/255,32/255)
Sec['SEA']=rgb(170/255,0/255,0/255)
Sec['SF']=rgb(0/255,34/255,68/255)
Sec['TB']=rgb(10/255,10/255,8/255)
Sec['TEN']=rgb(0/255,42/255,92/255)
Sec['WAS']=rgb(63/255,16/255,16/255)

playCount <- play %>%
  group_by(team,cat,playername)%>%
  group_by(team,cat)%>%summarise(playername=n())
theme_NFL <- function () {
  theme_bw(base_size=10, base_family="sans") %+replace%
  theme(
    axis.text.x=element_text(angle=30,
                              hjust=1,
                              vjust=1),

```

```

    panel.background = element_blank(),
    plot.background = element_blank(),
    panel.grid = element_blank(),
    complete=TRUE
  )
}

# Define UI for application that draws a histogram
ui <- fluidPage(

  # Application title
  titlePanel("NFL"),

  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
      sliderInput("week",
        "Week:",
        min = 1,
        max = 17,
        value = 17),
      selectInput("team",
        "Team:",
        choices = teamWin$winner,
        multiple=FALSE,
        selected = selectedTeam
      ),selectInput("stat",
        "Stat:",
        choices = uniquestat,
        multiple=FALSE)
    ),

    # Show a plot of the generated distribution
    mainPanel(
      tabsetPanel(type = "tabs",
        tabPanel("Win Record", plotOutput("winPlot")),
        tabPanel("League Stats",plotOutput('statPlot')),
        tabPanel("League Leader", plotOutput("llPlot"))
        , tabPanel("Team Stats", plotOutput(outputId = "teamstats", height= '1000px'))
        #
      )
    )
  )

)

# Define server logic required to draw a histogram
server <- function(input, output) {

  wins<-reactive({
    game2<- group_by(game,week, winner) %>%

```



```

    summarise(wins = n())
game2<-sqldf(paste('select * from game2 where week <= ',input$week,sep=' ' ))
game2<-group_by(game2,winner) %>%
  summarise(wins=sum(wins))

game2$highlight <- ifelse(game2$winner == input$team , '1', '0')
game2
})
limweek <- reactive({
  tackles<-play
  tackles<-sqldf(paste('select * from tackles where week <= ',input$week,sep=' ' ))

  tackles<- group_by(tackles,team, cat,G_B,O_D) %>%
    summarise(val = sum(val))
  tackles$highlight <- ifelse(tackles$team == input$team , '1', '0')
  tackles
})
fbStats<-reactive({
  tackles<-limweek()
  tackles$selcat <- ifelse(tackles$cat == input$stat , '1', '0')
  tackles<-tackles %>% filter(selcat == '1')

  tackles
})

leagueLeader <-reactive({
  LL<-sqldf(paste('select * from play where week <= ',input$week,sep=' ' ))
  LL <- group_by(LL,team,playername,cat) %>%
    summarise(val = sum(val))
  LL$selcat <- ifelse(LL$cat == input$stat , '1', '0')
  LL$highlight <- ifelse(LL$team == input$team , '1', '0')
  LL<-LL %>% filter(selcat == '1')

  LL<-LL[order(LL$val,decreasing=T),]
  LL = LL[1:10,]

  LL
})
teamStat <-reactive({
  tStt <-play
  tStt<-sqldf(paste('select * from tStt where week <= ',input$week,sep=' ' ))
  tStt$highlight <- ifelse(tStt$team == input$team , '1', '0')
  tStt<-tStt %>% filter(highlight == '1')
  tStt <-group_by(tStt,week,playername,cat,G_B,O_D,highlight)%>%
    summarise(val=sum(val))
  tStt$selcat <- ifelse(tStt$cat == input$stat , '1', '0')
  tStt<-tStt %>% filter(selcat == '1')
  tStt$running_total <- ave(tStt$val, tStt$playername, FUN=cumsum)

  tStt
})
facGraphs<- reactive({

```

```

length(unique(teamStat()$playername))
})

pCol <-reactive({ toString(Pri[input$team])})
sCol <-reactive({ toString(Sec[input$team])})

output$winPlot <- renderPlot({
  ggplot(wins()[order(wins()$wins,decreasing=T),],aes(x=reorder(winner,-wins, function(x){ sum(x) })))
  geom_bar(stat='identity',aes(fill=highlight))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  scale_y_continuous(breaks=pretty_breaks(),limits=c(0,16))+
  geom_hline(yintercept=sum(wins()$wins)/32)+
  scale_fill_manual(name='Top 5 Team',values=setNames(c(pCol(),sCol()),c(1,0)))+
  theme_NFL()+theme(legend.position="none")+
  xlab('Team')

})

output$statPlot <- renderPlot({
  ggplot(fbStats(),aes(x=reorder(team,-val, function(x){ sum(x) })),val))+
  geom_bar(stat = 'identity',aes(fill=highlight))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ylab(input$stat)+
  scale_y_continuous(breaks=pretty_breaks())+
  geom_hline(yintercept=mean(fbStats()$val))+
  scale_fill_manual(name='Top 5 Team',values=setNames(c(pCol(),sCol()),c(1,0)))+
  theme_NFL()+theme(legend.position="none")+
  xlab('Team')

})

output$llPlot <- renderPlot({
  ggplot(leagueLeader(),aes(x=reorder(paste(playername,team,sep=' | '),-val, function(x){ sum(x) })))
  geom_bar(stat = 'identity',aes(fill=highlight))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ylab(input$stat)+
  scale_y_continuous(breaks=pretty_breaks())+
  geom_hline(yintercept=mean(leagueLeader()$val))+
  scale_fill_manual(name='Top 5 Team',values=setNames(c(pCol(),sCol()),c(1,0)))+
  theme_NFL()+theme(legend.position="none")+
  xlab('Player')

})

output$teamstats <- renderPlot({
  print(facGraphs())
  ggplot(teamStat(),aes(week,running_total))+
  geom_line(color=pCol())+
  geom_point(aes(week,running_total),color=sCol())+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  facet_grid(playername~.,shrink=FALSE )+
  scale_y_continuous(breaks=pretty_breaks())+
  #scale_x_discrete()+
  scale_fill_manual(name='Top 5 Team',values=setNames(c(pCol(),sCol()),c(1,0)))+
  theme_NFL()+theme(legend.position="none")+
  xlab('Week')
})

```

```
  })  
  
}  
  
# Run the application  
shinyApp(ui = ui, server = server)  
#library(rsconnect)  
  
#rsconnect::deployApp(appName='NFL_STATS')
```