Isaac's stuff

Scraping

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(rvest)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.2

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.0     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   1.4.0

## Warning: package 'ggplot2' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'forcats' was built under R version 3.6.2

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

```r
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.6.2

##
## Attaching package: 'kableExtra'

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

```
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

wnba scraping

```
wilson <- 'https://www.basketball-reference.com/wnba/players/w/wilsoa01w/gamelog/2022/'
wil_doc <- rvest::read_html(wilson)

wil_doc %>%
  rvest::html_elements(., xpath = "//*[(@id = 'div_wnba_pgl_basic')]") %>%
  rvest::html_table() -> wil
wil <- wil[[1]]
head(wil)
```

```
## # A tibble: 6 x 28
##   Rk    Date  Age   Tm    ``    Opp   ``    GS    MP    FG    FGA   `FG%` `3P`
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 1     2022-~ 25-2~ LVA   "@"   PHO   W (+~ 1     28:35 5     8     .625  0
## 2 2     2022-~ 25-2~ LVA   ""    SEA   W (+~ 1     35:06 8     14    .571  1
## 3 3     2022-~ 25-2~ LVA   "@"   WAS   L (-~ 1     29:56 4     11    .364  0
## 4 4     2022-~ 25-2~ LVA   "@"   ATL   W (+~ 1     29:08 6     11    .545  0
## 5 5     2022-~ 25-2~ LVA   ""    PHO   W (+~ 1     33:45 4     8     .500  0
## 6 6     2022-~ 25-2~ LVA   ""    MIN   W (+~ 1     31:16 5     9     .556  1
## # ... with 15 more variables: 3PA <chr>, 3P% <chr>, FT <chr>, FTA <chr>,
## #   FT% <chr>, ORB <chr>, DRB <chr>, TRB <chr>, AST <chr>, STL <chr>,
## #   BLK <chr>, TOV <chr>, PF <chr>, PTS <chr>, GmSc <chr>
```

```
#wil2 <- mutate_all(wil, function(x) as.numeric(as.character(x)))
#mean(wil2['PTS'])

#wil$eFG<- (wil['FG'] + (0.5*wil['3P']))/wil['FGA']
#wil$eFG ![Screenshot]('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/images/scraping1')
```

EDA/Probability

Baseball

WAR comparison (Prob)

Link to WAR explaination: https://www.mlb.com/glossary/advanced-stats/wins-above-replacement

Player X has a projected mean WAR of 3 with standard deviation of 2 and player Y has a projected mean WAR of 1.5 with a standard deviation of 3. Assume projected WAR is normally distributed. Q: What is the probability that Player X outperforms Player Y? A: We want Pr(X>Y) or Pr(X-Y>0).
Let Z = X-Y.
E[Z]=1.5 Var(Z)=5 Pr(Z>0)=1-Pr($Z \leq 0$)

```r
#Calculate probability Z<=0
pr <- pnorm(0,1.5,sqrt(5))
print(1-pr)
```

```
## [1] 0.7488325
```

The Probability that Player X outperforms Player Y is 0.7488.

Injured Baserunner (Prob)

A runner on first base with 2 out and nobody else on base will attempt to steal second base on the first pitch 70% of the time if he is fully healthy but only 10% of the time if he is playing through an injury. Assume that 80% of the player population is healthy. You see a randomly selected runner not attempt a steal in this situation. Q: What is the probability that the runner is playing through an injury? A: From Bayes Theorem:

Pr(Injury given No Steal) = Pr(No Steal given Injury)*Pr(Injury)/P(No Steal).

Pr(No Steal given Injury) = 1 - Pr(Steal given Injury) = 0.9.

Pr(Injury) = 1- Pr(Healthy) = 0.2.

Pr(No Steal) = Pr(No Steal given Injury)*Pr(Injury)+Pr(No Steal given Healthy)*Pr(Healthy).

Pr(No Steal) = 0.9*0.2+0.7*0.8 = 0.74.

Therefore Pr(Injury given No Steal) = 0.9*0.2/0.74 = 0.243.

OPS (EDA)

Q: Using the dataset, plot the leagues average OPS from every year in the data to see the progression. A:

```r
mlb = read.csv('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/data/mlb_team_stats_history
head(mlb)
```

```
##   yearID lgID teamID franchID divID Rank   G Ghome  W  L DivWin WCWin LgWin
## 1   1976   NL    ATL      ATL     W    6 162    81 70 92      N           N
## 2   1976   AL    BAL      BAL     E    2 162    81 88 74      N           N
## 3   1976   AL    BOS      BOS     E    3 162    81 83 79      N           N
## 4   1976   AL    CAL      ANA     W    4 162    81 76 86      N           N
## 5   1976   AL    CHA      CHW     W    6 161    80 64 97      N           N
## 6   1976   NL    CHN      CHC     E    4 162    81 75 87      N           N
##   WSWin   R   AB    H  X1B X2B X3B  HR  BB  SO  SB CS HBP SF  RA    BA  ER  ERA
## 1     N 620 5345 1309 1027 170  30  82 589 811  74 61  19 47 700 0.245 617 3.86
## 2     N 619 5457 1326  966 213  28 119 519 883 150 61  23 35 598 0.243 541 3.32
## 3     N 716 5511 1448 1004 257  53 134 500 832  95 70  29 59 660 0.263 571 3.52
## 4     N 550 5385 1265  969 210  23  63 534 812 126 80  42 48 631 0.235 551 3.36
## 5     N 586 5532 1410 1082 209  46  73 471 739 120 53  34 55 745 0.255 684 4.25
## 6     N 611 5519 1386 1041 216  24 105 490 834  74 74  30 41 728 0.251 643 3.93
##   CG SHO SV IPouts   HA HRA BBA SOA   E  DP    FP              name
## 1 33  13 27   4314 1435  86 564 818 167 151 0.973    Atlanta Braves
## 2 59  16 23   4406 1396  80 489 678 118 157 0.982 Baltimore Orioles
## 3 49  13 27   4374 1495 109 409 673 141 148 0.978    Boston Red Sox
## 4 64  15 17   4432 1323  95 553 992 150 139 0.977 California Angels
## 5 54  10 22   4344 1460  87 600 802 130 155 0.979 Chicago White Sox
## 6 27  12 33   4414 1511 123 490 850 140 145 0.978      Chicago Cubs
##                           park attendance BPF PPF teamIDBR teamIDlahman45
## 1 Atlanta-Fulton County Stadium     818179 106 108      ATL            ATL
## 2              Memorial Stadium    1058609  94  93      BAL            BAL
## 3                Fenway Park II    1895846 113 112      BOS            BOS
```

```
## 4                Anaheim Stadium    1006774   93   94       CAL               CAL
## 5                  Comiskey Park     914945  101  102       CHW               CHA
## 6                  Wrigley Field    1026217  108  109       CHC               CHN
##   teamIDretro
## 1         ATL
## 2         BAL
## 3         BOS
## 4         CAL
## 5         CHA
## 6         CHN
```

```r
# make new variables
mlb=mutate(mlb,SLG=(X1B+2*X2B+3*X3B+4*HR)/(AB))
mlb=mutate(mlb,OBP=(H+BB+HBP)/(AB+BB+HBP+SF))
mlb=mutate(mlb,OPS=OBP+SLG)

# get avg ops
summarize(mlb, Average = mean(OPS,na.rm=T))
```

```
##     Average
## 1 0.7330384
```

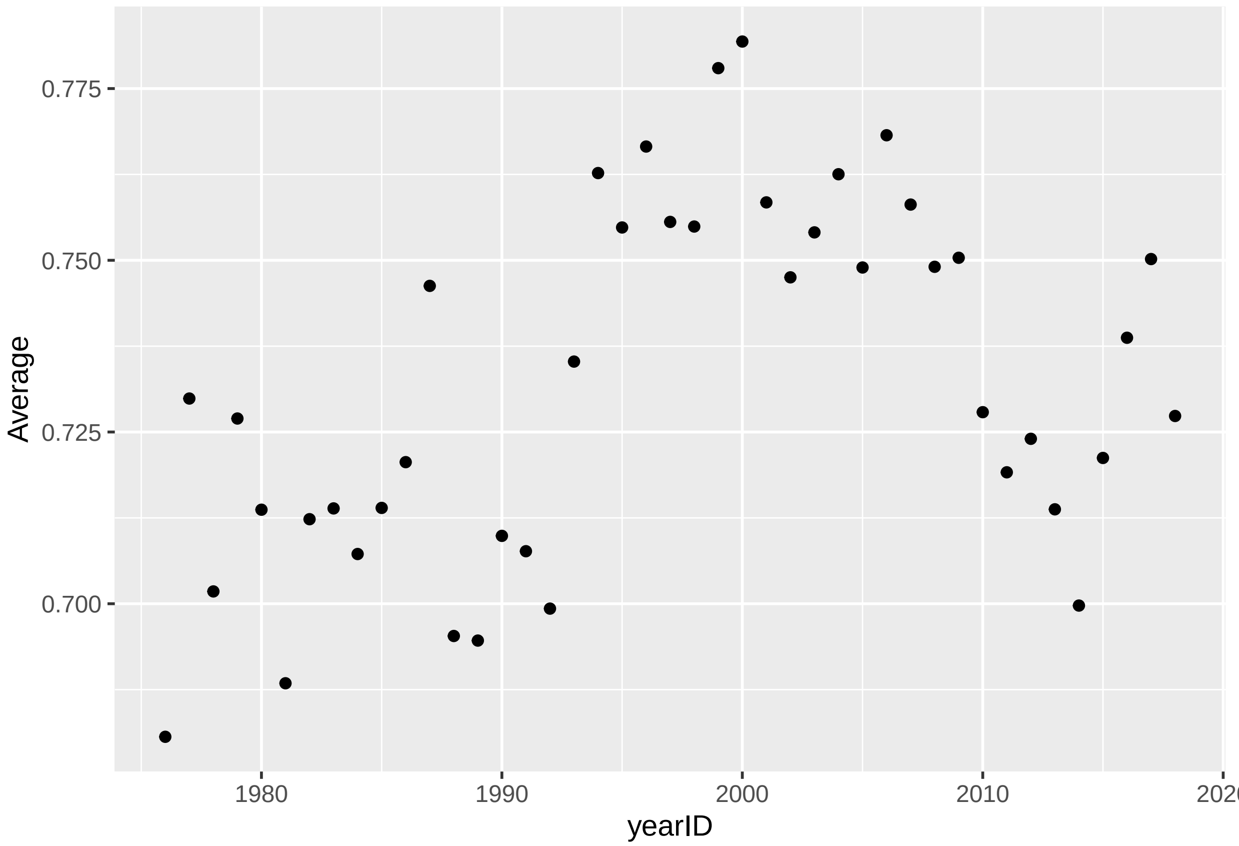
```r
# get avg ops by year
group_by(mlb, yearID)%>%
summarize(Average = mean(OPS, na.rm=T))
```

```
## # A tibble: 43 x 2
##    yearID Average
##     <int>   <dbl>
##  1   1976   0.681
##  2   1977   0.730
##  3   1978   0.702
##  4   1979   0.727
##  5   1980   0.714
##  6   1981   0.688
##  7   1982   0.712
##  8   1983   0.714
##  9   1984   0.707
## 10   1985   0.714
## # ... with 33 more rows
```

```r
group_by(mlb, yearID)%>%
summarize(Average = mean(OPS, na.rm=T))%>%View

#create new dataset
mlbYr=group_by(mlb, yearID)%>%
summarize(Average = mean(OPS, na.rm=T))

#plot it
ggplot(mlbYr, aes(x=yearID, y= Average))+geom_point()
```



Followup Q: What would cause the data to peak around the year 2000? A: PED's

Run Variance (Probability)

Runs Scored	Probability
0	0.55
1	0.25
2	0.15
3	0.05

Q: Using the probability table provided, calculate the variance for runs scored in an inning A: $E(X) = 1 * 0.25 + 2 * 0.15 + 3 * 0.05 = 0.7$
$E(X^2) = 1 * 0.25 + 4 * 0.15 + 9 * 0.05 = 1.3$
$Var(X) = E(X^2) - E(X) = 1.3 - 0.7 = 0.6$

Tennis

Link for brief explanation of tennis scoring: https://www.sportingnews.com/us/tennis/news/tennis-scoring-explained-rules-syst
7uzp2evdhbd11obdd59p3p1cx

Probability of Winning a Game (Prob)

The formula for the probability of a tennis player winning a game (from Analyzing Wimbledon) is given by $\frac{p^4*(-8*p^3+28*p^2-34*p+15)}{p^2+(1-p)^2}$ where p is the probability of a player winning their service point. Q: If a player wins their service points 62% of the time, what is the probability they win the game? A:

```r
p <- 0.62
pr_game <- (p^4*(-8*p^3+28*p^2-34*p+15))/(p^2+(1-p)^2)
pr_game
```

```
## [1] 0.7758627
```

Graph Example of Probability of Winning Point vs Probability of Winning Game (Prob)

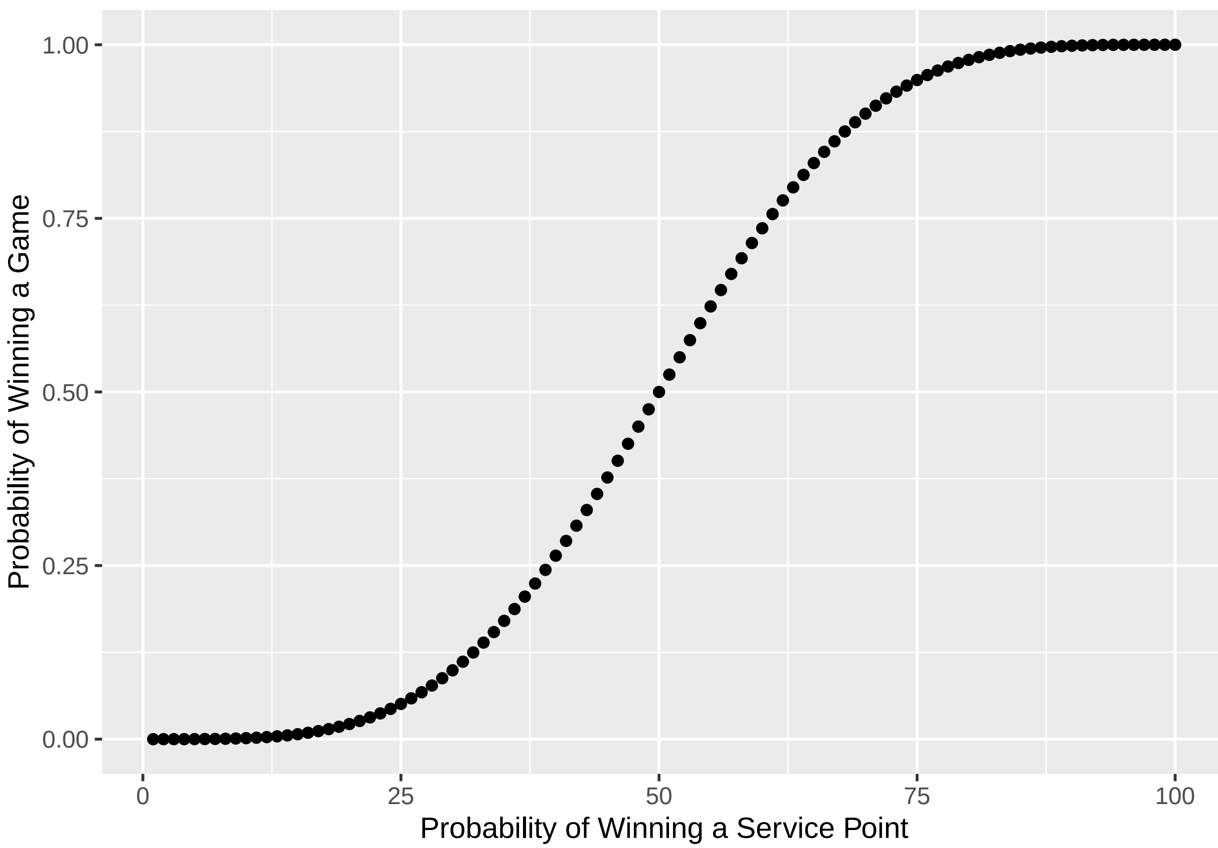
```r
game <- c(0)
pr <- 1:100
for(x in pr) {
  p <- pr/100
  pr_game <- (p^4*(-8*p^3+28*p^2-34*p+15))/(p^2+(1-p)^2)
  game <- c(game,pr_game)
}
game[1]
```

```
## [1] 0
```

```r
game <- game[2:101]
game[1]
```

```
## [1] 1.495898e-07
```

```r
df <- do.call(rbind, Map(data.frame, point_pr=pr, game_pr=game))
ggplot(df, aes(x=point_pr, y=game_pr)) +
  geom_point()+xlab('Probability of Winning a Service Point')+ylab('Probability of Winning a Game')
```



WNBA Scores (EDA)

Q: What is the difference in PPG for a winning team at home vs a winning team away? A:

```
wnba=read.csv('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/data/WNBA_Games2019_Scores.c
head(wnba)
```

```
##   Game         HomeTeam           AwayTeam Winner PTSwin PTSlose
## 1    1    Atlanta Dream       Dallas Wings   Home     76      72
## 2    2 New York Liberty      Indiana Fever   Away     81      80
## 3    3  Connecticut Sun Washington Mystics   Home     84      69
## 4    4   Minnesota Lynx        Chicago Sky   Home     89      71
## 5    5    Seattle Storm    Phoenix Mercury   Home     77      68
## 6    6   Las Vegas Aces Los Angeles Sparks   Home     83      70
##       WinningTeam
## 1   Atlanta Dream
## 2   Indiana Fever
## 3 Connecticut Sun
## 4  Minnesota Lynx
## 5   Seattle Storm
## 6  Las Vegas Aces
```

```
group_by(wnba, Winner)%>%
  summarize(Count=n())%>%
  mutate(Percent=Count/sum(Count))
```

```
## # A tibble: 2 x 3
##   Winner Count Percent
##   <fct>  <int>   <dbl>
## 1 Away      80   0.392
## 2 Home     124   0.608
```

```
group_by(wnba, Winner)%>%
  summarize(Average=mean(PTSwin,na.rm=T),sd=sd(PTSwin,na.rm=T))
```

```
## # A tibble: 2 x 3
##   Winner Average    sd
##   <fct>    <dbl> <dbl>
## 1 Away      83.8  9.20
## 2 Home      84.8 10.8
```

```
84.822-83.787
```

```
## [1] 1.035
```

A home team winner scores on average 1.035 PPG more than an away team winner.

NFL

```
nfl=read.csv('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/data/nfl_pbp.csv')
nfl2 <- select(nfl, c('Date','GameID','qtr','down','time','yrdline100','ydstogo','Yards.Gained','Touchdo
head(nfl2)
```

```
##         Date     GameID qtr down  time yrdline100 ydstogo Yards.Gained
## 1 2009-09-10 2009091000   1   NA 15:00         30       0           39
## 2 2009-09-10 2009091000   1    1 14:53         58      10            5
## 3 2009-09-10 2009091000   1    2 14:16         53       5           -3
## 4 2009-09-10 2009091000   1    3 13:35         56       8            0
```

```
## 5 2009-09-10 2009091000    1     4 13:27          56       8             0
## 6 2009-09-10 2009091000    1     1 13:16          98      10             0
##   Touchdown PlayType FieldGoalResult FieldGoalDistance ScoreDiff Season
## 1         0  Kickoff            <NA>                NA         0   2009
## 2         0     Pass            <NA>                NA         0   2009
## 3         0      Run            <NA>                NA         0   2009
## 4         0     Pass            <NA>                NA         0   2009
## 5         0     Punt            <NA>                NA         0   2009
## 6         0      Run            <NA>                NA         0   2009
```

4th Down Analysis (EDA)

Q: Using NFL Play by Play data, what percentage of the time do coaches choose to go for it on 4th down? And what percentage of 4th down attempts are successful? A:

```
# add indicator column for successful first down attempt
nfl2 <- nfl2 %>%
  mutate(FirstDown = case_when(
    ydstogo < Yards.Gained ~ 1,
    ydstogo > Yards.Gained ~ 0
    ))
# filter by only plays on 4th down
down4 = filter(nfl2, nfl2['down']==4)

#see what play types are run on fourth down and remove the noise
group_by(down4,PlayType) %>%
  summarize(Count=n())%>%
  mutate(Percentage=Count/sum(Count))
```

```
## # A tibble: 8 x 3
##   PlayType   Count Percentage
##   <fct>      <int>      <dbl>
## 1 Field Goal  7265  0.226
## 2 No Play     1433  0.0446
## 3 Pass        2239  0.0698
## 4 Punt       19551  0.609
## 5 QB Kneel      22  0.000685
## 6 Run         1424  0.0444
## 7 Sack         164  0.00511
## 8 Timeout        1  0.0000312
```

```
down4 = filter(down4, down4['PlayType']!='No Play' || down4['PlayType']!= 'QB Kneel' || down4['PlayType

# add indicator column for going for it on 4th
down4 <- down4 %>%
  mutate(GoForIt = case_when(
    PlayType == 'Pass' ~ 1,
    PlayType == 'Run' ~ 1,
    PlayType == 'Sack' ~ 1,
    PlayType == 'Field Goal' ~ 0,
    PlayType == 'Punt' ~ 0
  ))
# get percentage of 4th downs are gone for
group_by(down4,GoForIt) %>%
  summarize(Count=n())%>%
```

```
  mutate(Percentage=Count/sum(Count))
```

```
## # A tibble: 3 x 3
##   GoForIt Count Percentage
##     <dbl> <int>      <dbl>
## 1       0 26816     0.835
## 2       1  3827     0.119
## 3      NA  1456     0.0454
```

```
# get percentage of successful attempted 4th downs
down4 %>%
  filter(down4['GoForIt']==1) %>%
  group_by(FirstDown) %>%
    summarize(Count=n())%>%
    mutate(Percentage=Count/sum(Count))
```

```
## # A tibble: 3 x 3
##   FirstDown Count Percentage
##       <dbl> <int>      <dbl>
## 1         0  1971     0.515
## 2         1  1553     0.406
## 3        NA   303     0.0792
```

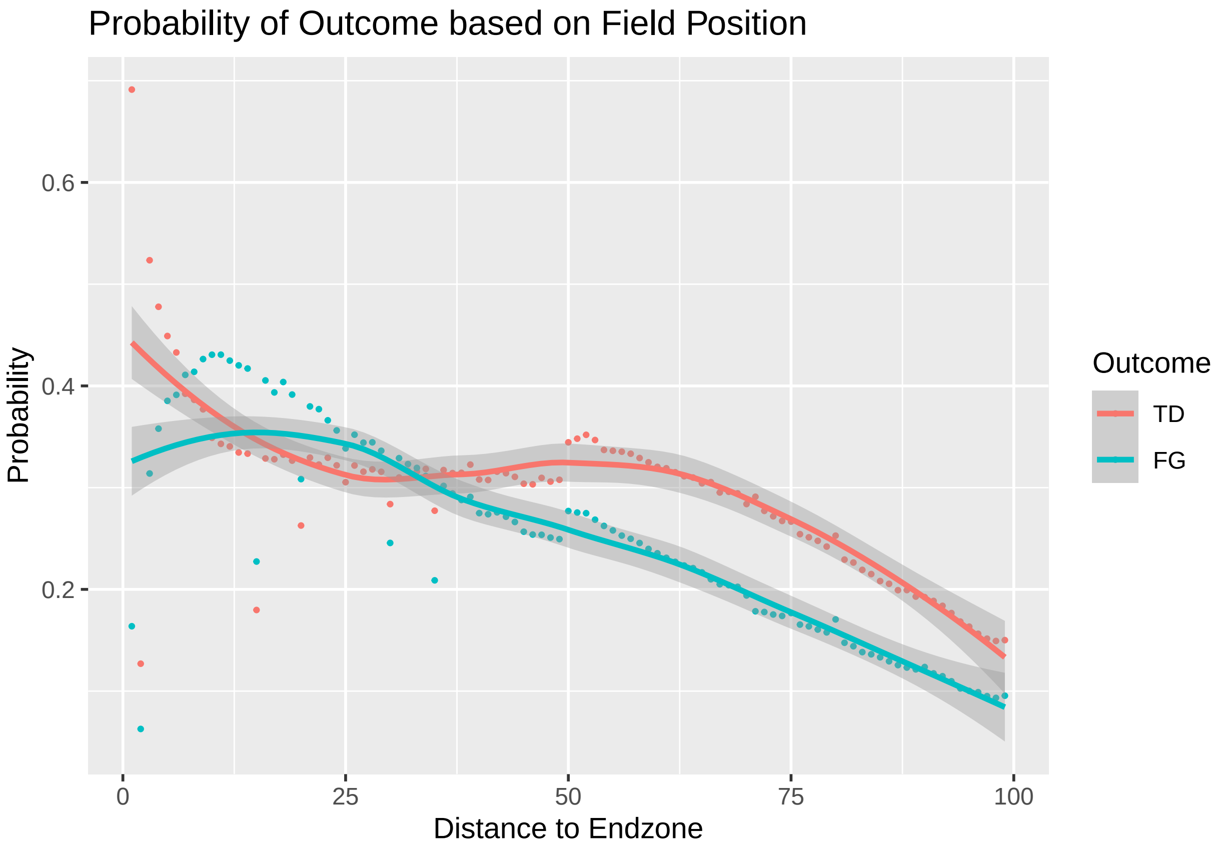
11% of 4th downs are gone for and 40% of those are successful, regardless of how many yards to go there are

Probability of Outcome based on Field Position Graph

```
nfl=read.csv('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/data/nfl_pbp.csv')
nfl2 <- select(nfl, c('yrdline100','ydstogo','No_Score_Prob','Field_Goal_Prob','Touchdown_Prob'))
nfl2 %>%
  na.omit() -> nfl2
```

```
nfl2 %>%
  group_by(yrdline100) %>%
    summarize(mean(Touchdown_Prob, na.rm=T)) -> td_prob
nfl2 %>%
  group_by(yrdline100) %>%
    summarize(mean(Field_Goal_Prob, na.rm=T)) -> fg_prob
nfl2 %>%
  group_by(yrdline100) %>%
    summarize(mean(No_Score_Prob, na.rm=T)) -> no_prob
x <- c('yrdline100','probability', 'Outcome')
colnames(td_prob) <- x
ind <- data.frame(ncol = 1, nrow=nrow(td_prob))
ind= 'TD'
td_prob <- cbind(td_prob,ind)
ind2 <- data.frame(ncol=1, nrow=nrow(fg_prob))
ind2='FG'
fg_prob <- cbind(fg_prob, ind2)
colnames(td_prob) <- x
colnames(fg_prob) <- x
prob_df <- rbind(td_prob, fg_prob)
colnames(prob_df) <- x
ggplot(prob_df, aes(yrdline100,probability, col=Outcome)) +
  geom_point(size=0.5)+geom_smooth()+xlab('Distance to Endzone')+ylab('Probability')+ggtitle('Probabili
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#+facet_wrap('ydstogo')
```

Q: Why does the probability of scoring a field goal get lower as a team is within 10 yards of the endzone?

A: When a team is close to the endzone, they probability of scoring a touchdown goes way up so teams are less likely to attempt field goals since the expected value of attempting a touchdown is higher than the expected value of attempting a field goal.

IN PROGRESS Should they go for it? (Based on historical data)

```
yrd = 45
togo = 3


nfl=read.csv('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/data/nfl_pbp.csv')
nfl2 <- select(nfl, c('down','yrdline100','ydstogo','Yards.Gained','Touchdown','PlayType','FieldGoalResu
```

```
nfl2 <- nfl2 %>%
  mutate(FirstDown = case_when(
    ydstogo < Yards.Gained ~ 1,
    ydstogo > Yards.Gained ~ 0
    ))
# filter by only plays on 4th down
down4 = filter(nfl2, nfl2['down']==4)
down4 <- down4 %>%
  filter(PlayType!= 'Punt') %>%
  filter(PlayType!= 'No Play') %>%
```

```r
  filter(PlayType!= 'QB Kneel') %>%
  filter(PlayType!= 'Timeout') %>%
  mutate(GoForIt = case_when(
    PlayType == 'Pass' ~ 1,
    PlayType == 'Run' ~ 1,
    PlayType == 'Sack' ~ 1,
    PlayType == 'Field Goal' ~ 0
  ))

down4 <- down4 %>%
  mutate(Success = case_when(
    FieldGoalResult == 'Good' ~ 1,
    FirstDown == 1 ~ 1,
    TRUE ~ 0
  ))
down5 <- down4 %>%
  group_by(yrdline100) %>%
        group_by(Success) %>%
          summarize(Count = n()) %>%
          mutate(Percentage=Count/sum(Count))

head(down5)
```

```
## # A tibble: 2 x 3
##   Success Count Percentage
##     <dbl> <int>      <dbl>
## 1       0  3341      0.301
## 2       1  7751      0.699
```

```r
'
down5<- down4 %>%
  group_by(yrdline100) %>%
    group_by(ydstogo) %>%
      group_by(GoForIt) %>%
        group_by(Success) %>%
          summarize(Count = n()) %>%
          mutate(Percentage=Count/sum(Count))
'
```

```
## [1] "\ndown5<- down4 %>%\n  group_by(yrdline100) %>%\n    group_by(ydstogo) %>%\n      group_by(GoFo
```

```r
# create list of plays (list)
# for i in 0-100:
#   count = 0
#   if yrdline100 = i:
#       count = count+1
#       list[i][count] = c(ydstogo, success, goforit)

# group by yards to go

# create list of probabilities (list2)
#create final list (list_f)
# for i in 0-100:
#   create sublist (sublist)
#   lst = list2[i]
```

```
#   for i in length(lst):
#       add lst[i][]


## want to find every play with same situation and group by playtype and find success rate for going fo
```

Football Sample Space (Probability)

A sample space contains all possible outcomes. An american football game can either end with a win (W), loss (L) or a tie (T) which means our sample space is $\Omega = \{W, L, T\}$ and an event, E would be one of the possible outcomes. If a team wins the game, the event for that game would be $E = \{W\}$ or if we want the event of the 2021 CSU football season, it would be $E = \{L, L, W, L, W, W, L, L, L, L, L, L\}$.

Gambling

Sports Betting Bankroll Management

To prevent problematic gambling many people use bankrolls, or money set aside with the sole purpose of gambling. It is often suggested that people partaking in sports betting set aside an amount they are comfortable gambling with and betting 1-5% of that per play, especially considering minimum bets for online sportsbooks are often less than $1. One of the big risks in gambling is chasing losses so a popular strategy is called flat betting, where you bet the same amount on every game to help minimize losses. In addition, parlays can have incredibly attractive payoffs but high reward comes with high risk so it can be quite difficult to find long term profit.

Hold Percentage/Breaking Even

In a perfect world, if a baseball game has probability 0.5 of either team winning, the odds would be +100 (or -100), 50% of bettors would be on one side, 50% on the other, and at the end of the game half of the bettors would double their money and half of them would lose their money. Unfortunately, this would mean that the casinos make no money so to combat this they introduce whats called a hold percentage. Essentially sportsbooks will give you slightly worse odds on bets in order to make money. In this example, when both sides have probability 0.5, the offered lines may both be -110 that way when one team wins, half the bettors win slightly less than double the money (bet 110 dollars to win 100) and the sportsbook collects the rest. As a result, winning percentages have to be higher than expected to show a profit. For -110 odds (the Vegas equivalent of probability = 0.5), a bettor must win at a rate of 52.23% of the time in order to show a profit. Similarly you can convert any American odds into implied probability to get a breakeven percentage, since if you have a win rate higher than the implied probability, the expected value of your bet is positive.

Q: The given odds for the Avs to win the 2022 Stanley Cup are +500 and you've concluded that the bet has a 15% chance of happening. Is it worth making this bet?

A: The implied probability on a +500 bet is 16.66%, therefore if the bet has a 15% chance of happening, it is not a good bet as the expected value is negative but lets look at a simulation.

```
# lets say we start with $1 and we want to bet $1
dollars = 1
set.seed(1)
# we want to simulate the bet 100,000 times
n.sims <- 100000
for(i in 1:n.sims){
# Simulate whether the Avs win based on our probability of 15%
  win <- sum(rbinom(1,1,0.15))
  # change our dollar amount based off the odds
```

```
  if(win == 1){
    dollars = dollars + 5
  }
  if(win == 0){
    dollars = dollars - 1
  }
}
print(dollars)
```

```
## [1] -9573
```

After 100,000 bets, we would be down 9,573 dollars just off of 1 dollar bets.

Gamblers Fallacy

The Gambler's Fallacy is the common misbelief that if independent events fail to happen, theyre more likely to happen in the future. For example, say we are playing Roulette and betting on whether the ball lands on black or red. Theres roughly a 50% chance of either outcome, yet if we somehow get 10 reds in a row, the natural inclination is to assume a black is "due" and must come soon. However, both outcomes still have just a 50% chance of happening regardless of the history. This can be seen in the sports world as well, especially when talking about something like the 'hot hand' in basketball or hitting streaks in baseball, although sports does differ from conventional gambling as recent performance in sports can be useful for predictive purposes.

Kelly Criterion

While flat betting is a common and effective bet sizing strategy, a more advanced technique is called the Kelly Criterion. The Kelly Criterion adjusts each bet size to the specific bet based on bankroll size, given odds, and the predicted probability that the bettor gives the outcome. The formula for calculating the bet size is $f = p - \frac{1-p}{b}$ where f is the fraction of your current bankroll you should wager, p is the probability the bettor gives themselves of winning the bet, and b is the proportion of the bet you win back (+200 odds pays 2 to 1 therefore the proportion is 2.0). This is certainly more complicated than flat betting (especially in large volumes), but it has been proven to provide theoretically optimal bet sizes.

SANDBOX (Can be ignored)

```
nfl=read.csv('~/Google Drive/My Drive/Sports Analytics/SportsAnalyticsBook/data/nfl_pbp.csv')
nfl2 <- select(nfl, c('down','yrdline100','ydstogo','Touchdown','PlayType','FieldGoalResult','FieldGoal
head(nfl2)
```

```
##   down yrdline100 ydstogo Touchdown PlayType FieldGoalResult FieldGoalDistance
## 1   NA         30       0         0  Kickoff            <NA>                NA
## 2    1         58      10         0     Pass            <NA>                NA
## 3    2         53       5         0      Run            <NA>                NA
## 4    3         56       8         0     Pass            <NA>                NA
## 5    4         56       8         0     Punt            <NA>                NA
## 6    1         98      10         0      Run            <NA>                NA
```

```
## for each (grouping of 10 yards?) we get all the 4th down playtypes of fieldgoal and normal plays and
# filter by only plays on 4th down
down4 = filter(nfl2, nfl2['down']==4)

#see what play types are run on first down and remove the noise
group_by(down4,PlayType) %>%
  summarize(Count=n())%>%
  mutate(Percentage=Count/sum(Count))
```

```
## # A tibble: 8 x 3
##   PlayType    Count Percentage
##   <fct>       <int>      <dbl>
## 1 Field Goal  7265  0.226
## 2 No Play      1433  0.0446
## 3 Pass         2239  0.0698
## 4 Punt        19551  0.609
## 5 QB Kneel       22  0.000685
## 6 Run          1424  0.0444
## 7 Sack          164  0.00511
## 8 Timeout         1  0.0000312
```

```r
down3 = filter(down4, down4['PlayType']!='No Play' || down4['PlayType']!= 'QB Kneel' || down4['PlayType
head(down3)
```

```
##   down yrdline100 ydstogo Touchdown   PlayType FieldGoalResult
## 1    4         56       8         0       Punt            <NA>
## 2    4         96       8         0       Punt            <NA>
## 3    4         41      21         0       Punt            <NA>
## 4    4         19       7         0 Field Goal         No Good
## 5    4         79      16         0       Punt            <NA>
## 6    4         44      22         0       Punt            <NA>
##   FieldGoalDistance
## 1                NA
## 2                NA
## 3                NA
## 4                37
## 5                NA
## 6                NA
```

```r
## create 10 grouping dataframes (99-90,89-80,79-70,69-60,59-50,49-40,39-30,29-20,19-10,9-0)
x <- c("playtype", "outcome")
df99_90 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df99_90) <- x
df89_80 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df89_80) <- x
df79_70 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df79_70) <- x
df69_60 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df69_60) <- x
df59_50 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df59_50) <- x
df49_40 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df49_40) <- x
df39_30 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df39_30) <- x
df29_20 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df29_20) <- x
df19_10 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df19_10) <- x
df09_00 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(df09_00) <- x
```

```r
## create final dataframe with 3 columns: distance to goal, field goal probability, td probability
#df_prob <- data.frame(matrix(ncol = 3, nrow = 0))
#y = c('distance_to_goal','fg_prob','td_prob')
#colnames(df_prob) <- x
#down3 %>%
#  group_split(yrdline100)

down3 %>%
  mutate(distance = case_when(down3['yrdline100']<100 & down3['yrdline100'] >89 ~ 90,
                             down3['yrdline100']<90 & down3['yrdline100'] >79 ~ 80,
                             down3['yrdline100']<80 & down3['yrdline100'] >69 ~ 70,
                             down3['yrdline100']<70 & down3['yrdline100'] >59 ~ 60,
                             down3['yrdline100']<60 & down3['yrdline100'] >49 ~ 50,
                             down3['yrdline100']<50 & down3['yrdline100'] >39 ~ 40,
                             down3['yrdline100']<40 & down3['yrdline100'] >29 ~ 30,
                             down3['yrdline100']<30 & down3['yrdline100'] >19 ~ 20,
                             down3['yrdline100']<20 & down3['yrdline100'] >9 ~ 10,
                             down3['yrdline100']<10 ~ 0
    )) -> down3
down3 %>%
  group_split(distance) -> yrd_df

df99_90 <- yrd_df[[10]]
df89_80 <- yrd_df[[9]]
df79_70 <- yrd_df[[8]]
df69_60 <- yrd_df[[7]]
df59_50 <- yrd_df[[6]]
df49_40 <- yrd_df[[5]]
df39_30 <- yrd_df[[4]]
df29_20 <- yrd_df[[3]]
df19_10 <- yrd_df[[2]]
df09_00 <- yrd_df[[1]]
df09_00 %>%
  filter(df09_00['PlayType']=='Field Goal') %>%
  group_by(FieldGoalResult) %>%
    summarize(Count=n())%>%
    mutate(Percentage=Count/sum(Count))
```

```
## # A tibble: 4 x 3
##   FieldGoalResult Count Percentage
##   <fct>           <int>      <dbl>
## 1 Blocked            19    0.0116
## 2 Good             1593    0.974
## 3 No Good            21    0.0128
## 4 <NA>                2    0.00122
```

```r
df09_00 %>%
  filter(df09_00['PlayType']!='Field Goal' && df09_00['PlayType']!='Punt') %>%
  group_by(Touchdown) %>%
    summarize(Count=n())%>%
    mutate(Percentage=Count/sum(Count))
```

```
## # A tibble: 2 x 3
##   Touchdown Count Percentage
```

```
##        <int> <int>      <dbl>
## 1          0  2128     0.910
## 2          1   210     0.0898
```