Code for Track 3 - Men's

```
#filter out specific event
m100M <- results %>% filter(Event == "100M Men")

#transform result (time) to double
m100M <- m100M %>% mutate(Result = as.double(Result))

#manually input toyko 2021 data, webscraping unavailable for olympics.com
m100M <- m100M %>% add_row(Gender = c("M","M","M"),
                           Event = c("100M Men", "100M Men", "100M Men"),
                           Location = c("Tokyo","Tokyo","Tokyo"),
                           Year = c(2020,2020,2020),
                           Medal = c("G","S","B"),
                           Name = c("Lamont MARCELL JACOBS","Fred KERLEY","Andre DE GRASSE"),
                           Nationality = c("ITA","USA","CAN"),
                           Result = c(9.80,9.84,9.89)
                           )

#manually add DUMMY variable for missing years (due to WW2) in order for time analysis to be valid
#1988 olympics also gets a DUMMY variable due to the results being invalid due to wind.
#to to fix this, we took the average of the prior and following year as the input.
m100M <- m100M %>%
  add_row(Gender = c("M","M","M","M","M","M","M","M","M","M"),
          Event = c("100M Men", "100M Men", "100M Men","100M Men", "100M Men", "100M Men","100M Men", "
          Location = c("DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY"),
          Year = c(1940,1940,1940,1944,1944,1944,1988,1988,1988),
          Medal = c("G","S","B","G","S","B","G","S","B"),
          Name = c("DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY"),
          Nationality = c("DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY","DUMMY"),
          Result = c(10.30,10.40,10.55,10.30,10.40,10.55,9.98,10.11,10.13)
                           )

#fixing the results of specifics to match olympics.com
m100M[38,"Result"] <- 11.0
m100M[39,"Result"] <- 11.0
```

```
#pull out gold medal winners
m100M_G <- m100M %>% filter(Medal == "G",Year >= 1928) %>% arrange(Year)
#pull out silver medal winners
m100M_S <- m100M %>% filter(Medal == "S",Year >= 1928) %>% arrange(Year)
#pull out bronze medal winners
m100M_B <- m100M %>% filter(Medal == "B",Year >= 1928) %>% arrange(Year)
```

```
#make individual time series objects for gold, silver, bronze
gts <- ts(m100M_G[8], start = 1928, end = 2020, deltat = 4)
sts <- ts(m100M_S[8], start = 1928, end = 2020, deltat = 4)
```

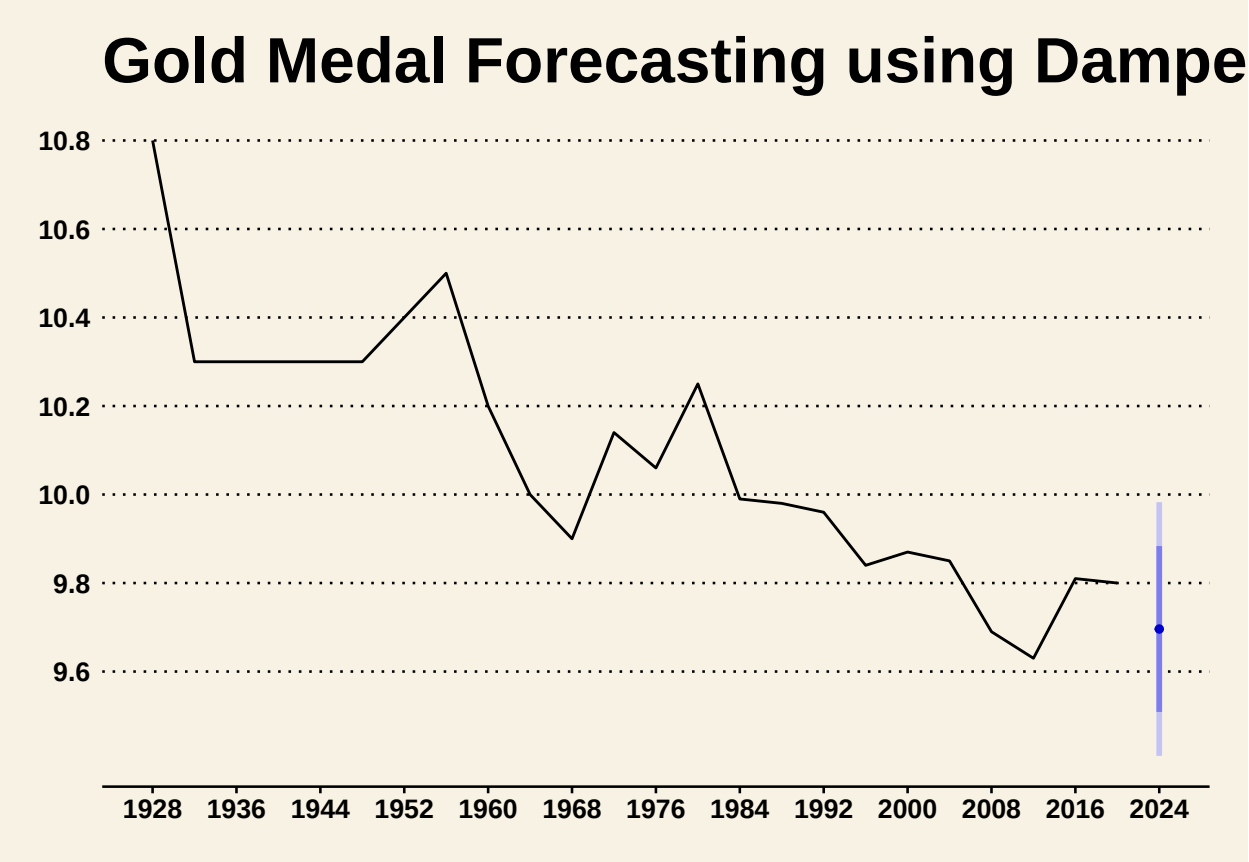
```r
bts <- ts(m100M_B[8], start = 1928, end = 2020, deltat = 4)

#prediction graph and summary using Holt's method for exponential time series smoothing for gold medal
g_model <- holt(gts, h = 1, damped = TRUE)
summary(g_model)
```

```
##
## Forecast method: Damped Holt's method
##
## Model Information:
## Damped Holt's method
##
## Call:
## holt(y = gts, h = 1, damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##     phi   = 0.9762
##
##   Initial states:
##     l = 10.5761
##     b = -0.0474
##
##   sigma:  0.1462
##
##       AIC      AICc       BIC
## -9.615436 -4.674259 -2.547113
##
## Error measures:
##                         ME      RMSE        MAE         MPE      MAPE
## Training set 3.824094e-05 0.1301135 0.09890066 -0.01569171 0.9712747
##                      MASE      ACF1
## Training set 0.009801444 0.1573719
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2024       9.696104 9.508697 9.883512 9.409489 9.982719
```

```r
autoplot(g_model) +
  ggtitle("Gold Medal Forecasting using Damped Holt's Method") +
  xlab("Olympic Year") +
  ylab("Time in Seconds") +
  scale_x_continuous(breaks=seq(1928,2024,8)) +
  scale_y_continuous(breaks=seq(9.6,10.8,0.2)) +
  theme_wsj(base_size = 10, title_family = "sans")
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```



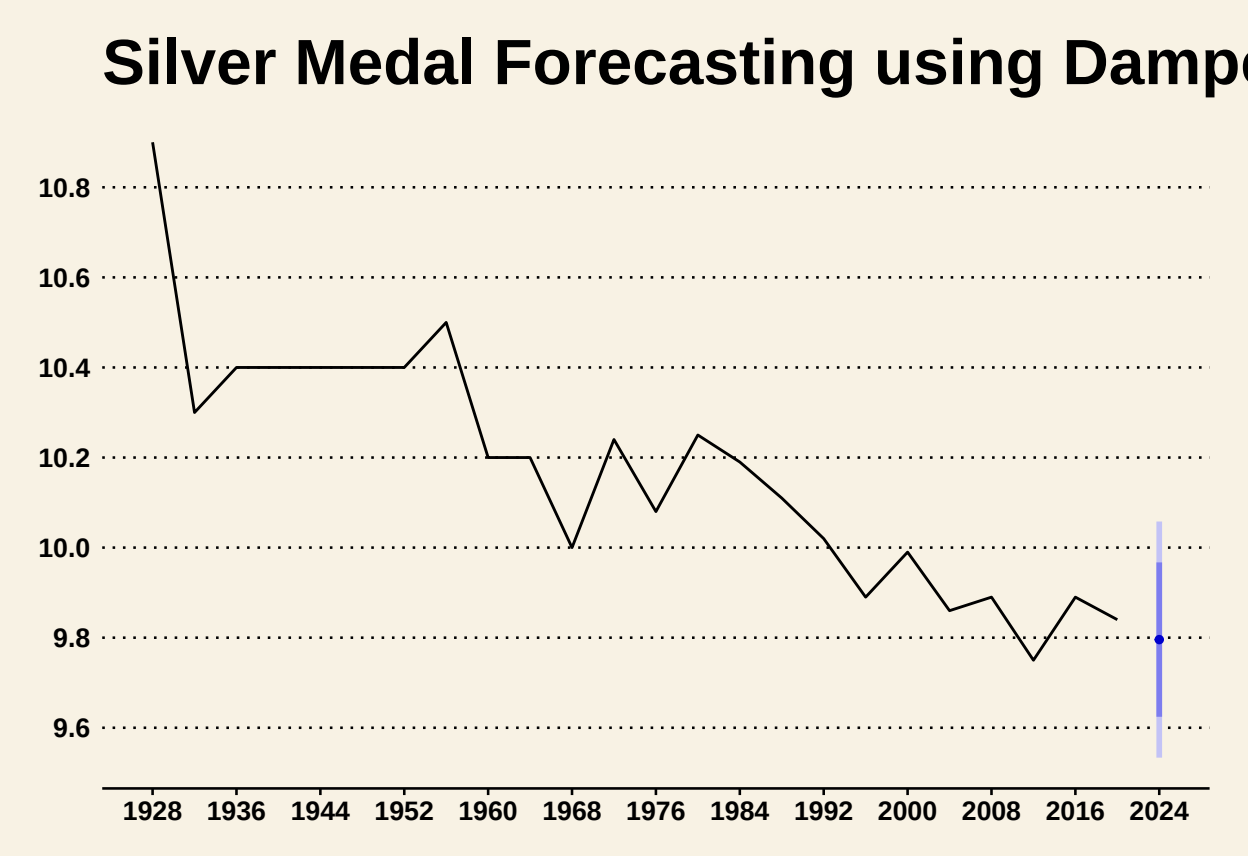

```
#prediction graph and summary using Holt's method for exponential time series smoothing for silver meda
s_model <- holt(sts, h = 1, damped = TRUE)
summary(s_model)
```

```
##
## Forecast method: Damped Holt's method
##
## Model Information:
## Damped Holt's method
##
## Call:
## holt(y = sts, h = 1, damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##     phi   = 0.9736
##
##   Initial states:
##     l = 10.6436
##     b = -0.0471
##
##   sigma:  0.1337
##
##        AIC       AICc        BIC
```

```
## -13.901503  -8.960327  -6.833180
##
## Error measures:
##                          ME      RMSE        MAE        MPE      MAPE
## Training set -6.514133e-05 0.1189988 0.09120502 -0.0138198 0.8883477
##                      MASE       ACF1
## Training set 0.008967311 -0.1250213
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95   Hi 95
## 2024        9.79577 9.624371 9.967169 9.533638 10.0579
```

```
autoplot(s_model) +
  ggtitle("Silver Medal Forecasting using Damped Holt's Method") +
  xlab("Olympic Year") +
  ylab("Time in Seconds") +
  scale_x_continuous(breaks=seq(1928,2024,8)) +
  scale_y_continuous(breaks=seq(9.6,10.8,0.2)) +
  theme_wsj(base_size = 10, title_family = "sans")
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

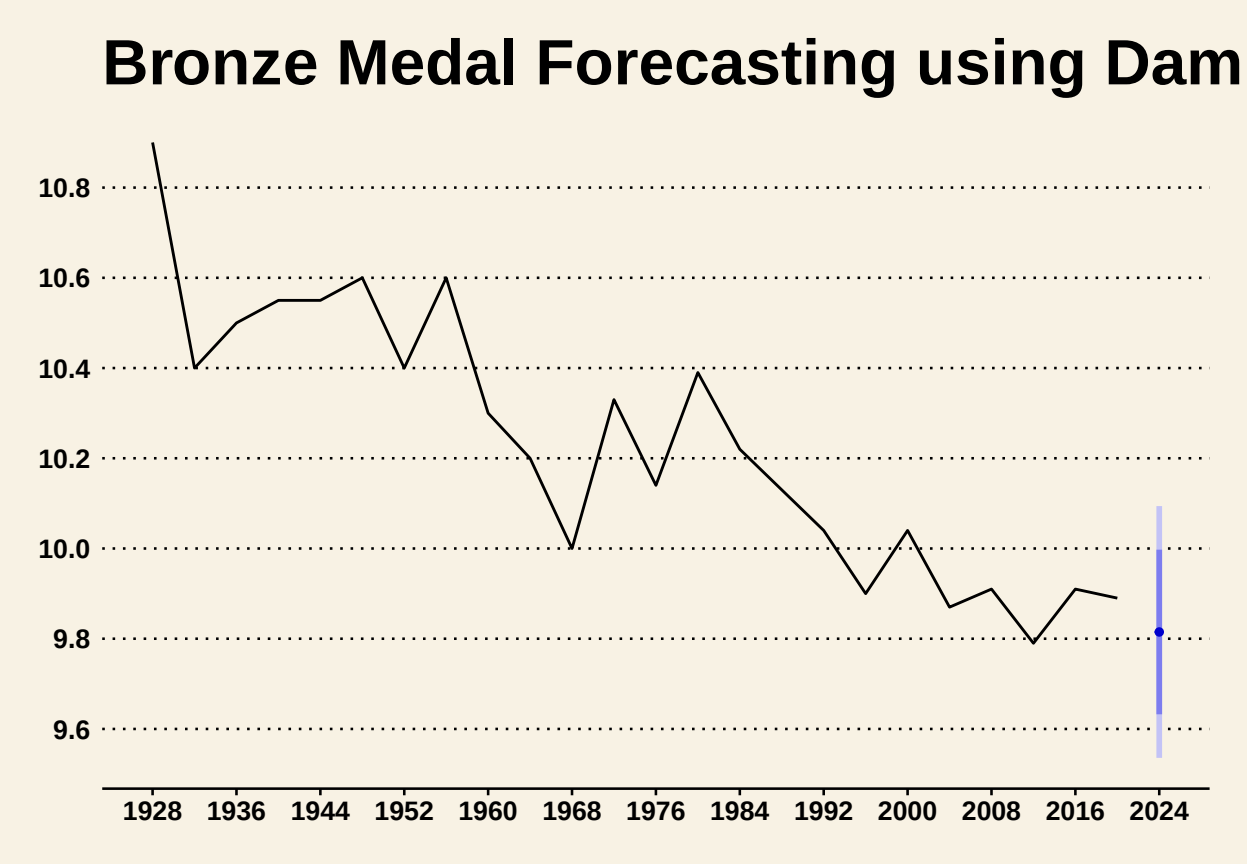


```r
#prediction graph and summary using Holt's method for exponential time series smoothing for bronze meda
b_model <- holt(bts, h = 1, damped = TRUE)
summary(b_model)
```

```
##
## Forecast method: Damped Holt's method
##
## Model Information:
## Damped Holt's method
##
## Call:
## holt(y = bts, h = 1, damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##     phi   = 0.9726
##
##   Initial states:
##     l = 10.7591
##     b = -0.0531
##
##   sigma:  0.1424
##
##        AIC       AICc        BIC
## -10.902448  -5.961272  -3.834125
##
## Error measures:
##                       ME      RMSE       MAE          MPE      MAPE        MASE
## Training set 0.001043916 0.1266711 0.1001852 -0.004607104 0.9736656 0.009791682
##                      ACF1
## Training set -0.07074429
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2024       9.814956 9.632507 9.997405 9.535924 10.09399
```

```r
autoplot(b_model) +
  ggtitle("Bronze Medal Forecasting using Damped Holt's Method") +
  xlab("Olympic Year") +
  ylab("Time in Seconds") +
  scale_x_continuous(breaks=seq(1928,2024,8)) +
  scale_y_continuous(breaks=seq(9.6,10.8,0.2)) +
  theme_wsj(base_size = 10, title_family = "sans")
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```


Bronze Medal Forecasting using Dam
10.8
10.6
10.4
10.2
10.0
9.8
9.6
1928
1936
1944
1952
1960
1968
1976
1984
1992
2000
2008
2016
2024