Winning Characteristics in the Olympics

Noah Obuya and Tamya Davidson

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6     v purrr   0.3.4
v tibble  3.1.8     v dplyr   1.0.9
v tidyr   1.2.0     v stringr 1.4.1
v readr   2.1.2     v forcats 0.5.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
-- Attaching packages --------------------------------------- tidymodels 1.0.0 --
v broom        1.0.1     v rsample      1.1.0
v dials        1.0.0     v tune         1.0.1
v infer        1.0.3     v workflows    1.1.0
v modeldata    1.0.1     v workflowsets 1.0.0
v parsnip      1.0.2     v yardstick    1.1.0
v recipes      1.0.2
-- Conflicts ------------------------------------------ tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Dig deeper into tidy modeling with R at https://www.tmwr.org
```

```
library(formatR)
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```
library(nnet)
library(car)
```

```
Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':

    recode

The following object is masked from 'package:purrr':

    some
```

```
library(lme4)
```

```
Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack
```

```
library(glmnet)
```

```
Loaded glmnet 4.1-6
```

```
Rows: 271116 Columns: 15
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (10): name, sex, team, noc, games, season, city, sport, event, medal
dbl  (5): id, age, height, weight, year

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.
```

Introduction and Data

Research Question

What are the most influential characteristics (between sex, age, height, weight, and country) when it comes to predicting gold medals in the Summer Olympics, and do these characteristics change over the course of a decade?

We chose Olympics data from TidyTuesday's github repository (https://github.com/rfordatascience/tidytuesday 07-27/readme.md). The data were collected by scraping www.sports-reference.com and was created in May 2018. The data contains 271,116 observations of 15 variables. The variables of interest in our research include sex, age, height, weight, noc (country), year, season, and medals (Gold, Silver, Bronze). Based on these variables, we will answer the question of what are the most influential variables that influence an athlete receiving a gold medal , and do these variables change over time. From the data set we will only observe the more recent Olympic games (including the years 2004, 2008, 2012, 2016), and we will analyze our research question through subsets of the data. There are many NA values corresponding to medals, and because this is our variable of interest we will drop all NA values corresponding to medals. After doing this we are left with a case study of 39,783 observations of 15 variables. The motivation behind this project is to analyze what athletes can do to better prepare for the Olympic games, and see which factors are more influential than others.

Variables of Interest

Exploratory Data Analysis

```
# A tibble: 3 x 3
  medal      n   per
  <chr>  <int> <dbl>
1 Bronze   706 0.347
2 Gold     664 0.326
3 Silver   665 0.327
```

```
olympics12 %>%
  count(medal) %>%
  mutate(per = n/sum(n))
```

```
# A tibble: 3 x 3
  medal      n   per
  <chr>  <int> <dbl>
1 Bronze   669 0.349
2 Gold     622 0.325
3 Silver   624 0.326
```

```r
olympics16 %>%
  count(medal) %>%
  mutate(per = n/sum(n))
```

```
# A tibble: 3 x 3
  medal      n   per
  <chr>  <int> <dbl>
1 Bronze   700 0.348
2 Gold     662 0.329
3 Silver   652 0.324
```

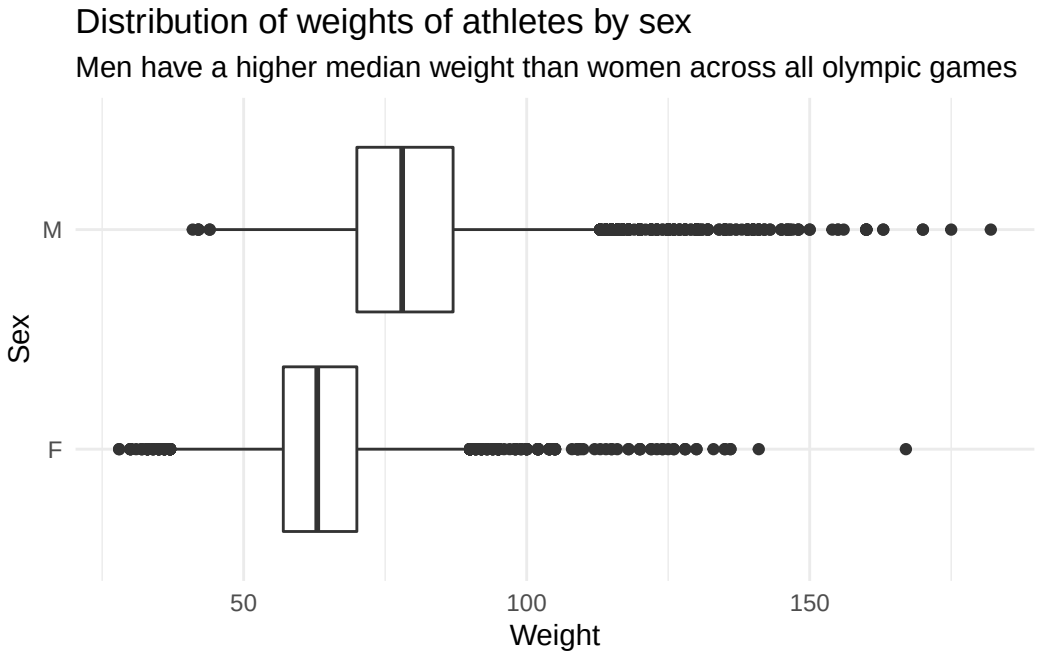
In 2004, the number of bronze medals handed out to individuals was 676 which was 33.8% of the total medals, the number of silver medals was 660 which was 33% of the total medals , and the number of gold medals was 664 which was 33.2% of the total medals .

In 2008, the number of bronze medals handed out to individuals was 706 which was 34.7% of the total medals , the number of silver medals was 665 which was 32.7% of the total medals, and the number of gold medals was 664 which was 32.6% of the total medals.

In 2012, the number of bronze medals handed out to individuals was 669 which was 35% of the total medals , the number of silver medals was 624 which was 32.6% of the total medals, and the number of gold medals was 622 which was 32.4% of the total medals.

In 2016, the number of bronze medals handed out to individuals was 700 which was 34.8% of the total medals, the number of silver medals was 652 which was 32.3% of the total medals, and the number of gold medals was 662 which was 32.8% of the total medals.

```r
ggplot(olympics , mapping = aes(x = weight , y = sex )) +
  geom_boxplot() +
  theme_minimal() +
  labs(x = "Weight", y = "Sex", title = "Distribution of weights of athletes by sex", subt
```



As we can see from the boxplots above, the distribution of weight for men and women athletes competing in the olympics are both skewed to the right, while it appears that the men are skewed heavier. We were interested in the one female athlete who is considered an outlier because her weight is above 150. We have found the athlete to be Olha Vasylivna Korobka who actually got a silver medal in the 2008 summer games in weight lifting. (Code shown below).

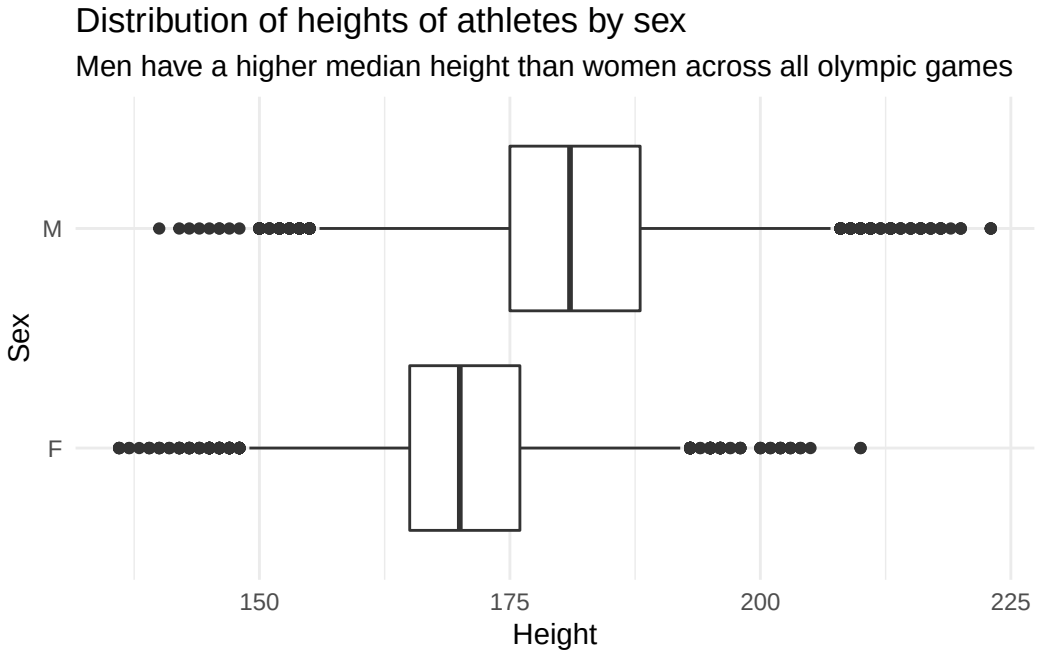
```
olympics %>%
  filter(sex == "F") %>%
  filter(weight > 150)
```

```
# A tibble: 1 x 15
     id name      sex     age height weight team  noc   games  year season city
  <dbl> <chr>     <chr> <dbl>  <dbl>  <dbl> <chr> <chr> <chr> <dbl> <chr>  <chr>
1 62843 Olha Vas~ F        22    181    167 Ukra~ UKR   2008~  2008 Summer Beij~
# ... with 3 more variables: sport <chr>, event <chr>, medal <chr>
```

Height vs Sex BoxPlots

```
ggplot(olympics , mapping = aes(x =height , y = sex )) +
  geom_boxplot() +
```

```
theme_minimal() +
labs(x = "Height", y = "Sex", title = "Distribution of heights of athletes by sex", subt
```



Similar to the results that we saw in the boxplots comparing distributions of weights between men and women, we can see that men also have a higher median height than women who have completed in the Olympics.

We chose to use all years when analyzing the distribution of heights and weights because over the course of our time frame (2004 - 2016) there have been many rule changes about allowed and not allowed substances, and analyzing these two variables through all of the years can give us a better idea of distributions.

Now that we have analyzed the data and got some idea of the distribution of specific parameters of interest, we are interested in analyzing which variables are the biggest factor in predicting gold medals for Summer Olympic games, and whether or not these variables (and their influence) change over time.

Logistic Regression

First we will fit a logistic regression model that predicts the probability of receiving a gold medal (for the purpose of this model, we will use the goldMedal? column that gives us a 1 if someone received a gold medal for their event, and gives us a 0 if someone did not receive

a gold medal (they received either gold or silver) this is due to the characteristics of logistic regression and how it works best when predicting a binary outcome.)

```
# A tibble: 147 x 5
   term          estimate std.error  statistic p.value
   <chr>            <dbl>     <dbl>      <dbl>   <dbl>
 1 (Intercept) -15.0      624.       -0.0241     0.981
 2 sexM          0.0535     0.0319    1.67       0.0942
 3 age           0.00165    0.00257   0.640      0.522
 4 height        0.00201    0.00201   0.999      0.318
 5 weight        0.000230   0.00148   0.156      0.876
 6 nocALG       13.9      624.        0.0223     0.982
 7 nocANZ        0.00897  764.        0.0000117  1.00
 8 nocARG       13.9      624.        0.0222     0.982
 9 nocARM       12.7      624.        0.0204     0.984
10 nocAUS       13.6      624.        0.0217     0.983
# ... with 137 more rows
```

The expected log odds of someone achieving a gold medal if their sex is male is 0.0535 times higher than if someone is a female when holding all other variables constant. For every one year increase in age, the expected log odds of someone achieving a gold medal is expected to increase by .00164 when all other variables are held constant. For every one unit increase in height, we expect the logs odds of someone achieving a gold medal to increase by approximately 0.0020 when all other variables are held constant. . For every one unit increase in weight, we expect the log odds of someone achieving a gold medal to increase by approximately 0.00022 when all other variables are held constant. For each respective noc, the expected log odds of someone achieving a gold medal to [increase or decrease] by X when all other variables are held constant.

Ordinal Regression

```
Re-fitting to get Hessian
```

```
# A tibble: 148 x 5
   term    estimate std.error statistic coef.type
   <chr>      <dbl>     <dbl>     <dbl> <chr>
 1 sexM   0.0227     0.0167      1.36   coefficient
 2 age    0.00128    0.00134     0.957  coefficient
 3 height 0.00126    0.00105     1.20   coefficient
 4 weight 0.0000409  0.000771    0.0531 coefficient
```

```
 5 nocALG 5.89       0.0254    232.     coefficient
 6 nocANZ 4.77       0.00141  3373.     coefficient
 7 nocARG 5.94       0.0768     77.3    coefficient
 8 nocARM 5.41       0.0217    249.     coefficient
 9 nocAUS 5.80       0.0364    160.     coefficient
10 nocAUT 5.85       0.0691     84.7    coefficient
# ... with 138 more rows
```

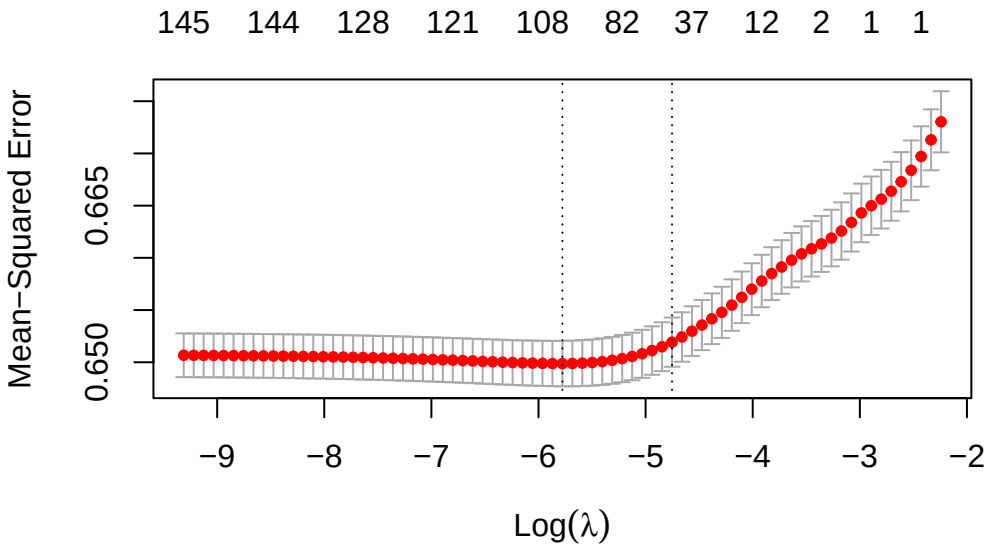
```
        sexM          age       height       weight       nocALG       nocANZ
1.022939e+00 1.001284e+00 1.001258e+00 1.000041e+00 3.609319e+02 1.179668e+02
      nocARG       nocARM       nocAUS       nocAUT       nocAZE       nocBAH
3.808297e+02 2.242517e+02 3.312331e+02 3.486949e+02 2.188170e+02 3.807950e+02
      nocBAR       nocBDI       nocBEL       nocBER       nocBLR       nocBOT
9.992425e-01 8.015282e+02 3.151169e+02 9.918638e-01 2.523201e+02 3.960174e+02
      nocBRA       nocBRN       nocBUL       nocCAN       nocCHI       nocCHN
3.213881e+02 4.087391e+02 2.831901e+02 4.004654e+02 1.526940e+02 4.380718e+02
      nocCIV       nocCMR       nocCOL       nocCRC       nocCRO       nocCUB
3.952125e+02 1.901743e+03 2.488254e+02 2.770645e+02 4.555103e+02 4.534841e+02
      nocCYP       nocCZE       nocDEN       nocDJI       nocDOM       nocECU
3.916621e+02 3.093963e+02 4.215064e+02 9.904270e-01 6.314113e+02 7.862872e+02
      nocEGY       nocERI       nocESP       nocEST       nocETH       nocEUN
1.901097e+02 1.027824e+00 3.368914e+02 3.155498e+02 4.014900e+02 5.128218e+02
      nocFIJ       nocFIN       nocFRA       nocFRG       nocGAB       nocGBR
1.720031e+04 2.888531e+02 3.462567e+02 3.323209e+02 3.874859e+02 3.824089e+02
      nocGDR       nocGEO       nocGER       nocGHA       nocGRE       nocGRN
4.701884e+02 2.482348e+02 3.875267e+02 9.115933e+01 3.616318e+02 7.812284e+02
      nocGUA       nocGUY       nocHAI       nocHKG       nocHUN       nocINA
3.949911e+02 1.013035e+00 3.917255e+02 4.013889e+02 4.038941e+02 3.872837e+02
      nocIND       nocIOA       nocIRI       nocIRL       nocISL       nocISR
4.019611e+02 2.172336e+02 3.498324e+02 4.168905e+02 3.345285e+02 1.100025e+02
      nocISV       nocITA       nocJAM       nocJOR       nocJPN       nocKAZ
3.905083e+02 3.571072e+02 3.775851e+02 1.567645e+04 3.289852e+02 3.211689e+02
      nocKEN       nocKGZ       nocKOR       nocKOS       nocKSA       nocKUW
4.367505e+02 1.458709e+02 4.183413e+02 1.606087e+04 1.004137e+02 9.791206e-01
      nocLAT       nocLIB       nocLIE       nocLTU       nocLUX       nocMAR
2.696198e+02 1.985573e+02 2.809772e+02 1.229116e+02 5.974742e+02 2.712552e+02
      nocMAS       nocMDA       nocMEX       nocMGL       nocMKD       nocMNE
2.676303e+02 1.571235e+02 3.556825e+02 2.005172e+02 9.981059e-01 4.019868e+02
      nocMOZ       nocMRI       nocNAM       nocNED       nocNGR       nocNIG
4.079326e+02 9.977590e-01 3.910636e+02 3.799121e+02 2.869115e+02 3.803270e+02
      nocNOR       nocNZL       nocPAK       nocPAN       nocPAR       nocPER
4.114726e+02 4.134593e+02 4.964070e+02 2.310660e+02 3.911467e+02 4.015484e+02
```

```
      nocPHI       nocPOL       nocPOR       nocPRK       nocPUR       nocQAT
1.609344e+02 2.836537e+02 2.858389e+02 2.949953e+02 1.881955e+02 1.004130e+02
      nocROU       nocRSA       nocRUS       nocSCG       nocSEN       nocSGP
3.150833e+02 3.340651e+02 3.873419e+02 3.006391e+02 3.848809e+02 2.575954e+02
      nocSLO       nocSRB       nocSRI       nocSUD       nocSUI       nocSUR
2.266431e+02 2.526747e+02 4.075605e+02 3.863946e+02 3.085770e+02 3.926000e+02
      nocSVK       nocSWE       nocSYR       nocTAN       nocTCH       nocTGA
4.148622e+02 3.431885e+02 3.936583e+02 3.903963e+02 2.811018e+02 3.877919e+02
      nocTHA       nocTJK       nocTOG       nocTPE       nocTTO       nocTUN
3.358591e+02 2.766558e+02 1.001161e+00 2.664786e+02 2.629928e+02 2.763449e+02
      nocTUR       nocUAE       nocUGA       nocUKR       nocURS       nocURU
4.637544e+02 3.876200e+02 3.973555e+02 2.741526e+02 4.942570e+02 5.386796e+02
      nocUSA       nocUZB       nocVEN       nocVIE       nocWIF       nocYUG
5.686591e+02 2.926607e+02 1.769305e+02 5.413833e+02 1.003835e+00 4.604019e+02
      nocZAM       nocZIM
3.891662e+02 1.342010e+03
```

Variable Selection

```
[1] 0.003098408
```

```r
plot(m_lasso_cv)
```




```
m_best <- glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
147 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept)  .
sexM         0.0068664920
age          0.0001888558
height       0.0007798195
weight       .
nocALG       .
nocANZ      -0.4312942616
nocARG       .
nocARM      -0.2075725595
nocAUS      -0.0619561290
nocAUT      -0.0050631105
nocAZE      -0.2740426879
nocBAH       .
nocBAR      -0.4155635845
nocBDI       0.1778569347
nocBEL      -0.0607687848
nocBER      -0.4196495465
```

```
nocBLR      -0.2204886764
nocBOT       .
nocBRA      -0.0726397838
nocBRN       .
nocBUL      -0.1654722668
nocCAN       0.0440867793
nocCHI      -0.4411521348
nocCHN       0.1040796797
nocCIV       .
nocCMR       0.7580256494
nocCOL      -0.1712833020
nocCRC       .
nocCRO       0.1159548552
nocCUB       0.1253675138
nocCYP       .
nocCZE      -0.0739127003
nocDEN       0.0663801357
nocDJI      -0.4152565196
nocDOM       0.1663496151
nocECU       0.1765738086
nocEGY      -0.2954310637
nocERI      -0.3979599685
nocESP      -0.0413497077
nocEST      -0.0095118328
nocETH       .
nocEUN       0.2000844142
nocFIJ       0.8932562676
nocFIN      -0.1483812577
nocFRA      -0.0247624195
nocFRG      -0.0493914268
nocGAB       .
nocGBR       0.0135141722
nocGDR       0.1579942587
nocGEO      -0.1672990819
nocGER       0.0251224460
nocGHA      -0.5624961948
nocGRE       .
nocGRN       0.1679027268
nocGUA       .
nocGUY      -0.4079533024
nocHAI       .
nocHKG       .
nocHUN       0.0502709174
```

```
nocINA       .
nocIND       0.0107005070
nocIOA      -0.1079476779
nocIRI       .
nocIRL       .
nocISL       .
nocISR      -0.4638446002
nocISV       .
nocITA      -0.0023116760
nocJAM       .
nocJOR       0.5120223216
nocJPN      -0.0638289249
nocKAZ      -0.0333834716
nocKEN       0.0716622241
nocKGZ      -0.2968054319
nocKOR       0.0680575919
nocKOS       0.5304491775
nocKSA      -0.5094857278
nocKUW      -0.5726641823
nocLAT      -0.1384473323
nocLIB      -0.0670768253
nocLIE      -0.0011425182
nocLTU      -0.5813761445
nocLUX       0.0728723378
nocMAR      -0.0945010962
nocMAS      -0.1205251094
nocMDA      -0.3820612317
nocMEX       .
nocMGL      -0.3156370166
nocMKD      -0.4141035135
nocMNE       .
nocMOZ       .
nocMRI      -0.4109598168
nocNAM       .
nocNED       0.0023071590
nocNGR      -0.1215946731
nocNIG       .
nocNOR       0.0588295991
nocNZL       0.0493248605
nocPAK       0.1693918111
nocPAN       .
nocPAR       .
nocPER       .
```

```
nocPHI      -0.3730526477
nocPOL      -0.1626057356
nocPOR      -0.0777091724
nocPRK      -0.0879716815
nocPUR      -0.2589679442
nocQAT      -0.5130270254
nocROU      -0.0921154883
nocRSA      -0.0036787346
nocRUS       0.0206940917
nocSCG      -0.0746484288
nocSEN       .
nocSGP      -0.0890302958
nocSLO      -0.2545363154
nocSRB      -0.2065315467
nocSRI       .
nocSUD       .
nocSUI      -0.0965895098
nocSUR       .
nocSVK       0.0126252158
nocSWE      -0.0295020394
nocSYR       .
nocTAN       .
nocTCH      -0.1719166349
nocTGA       .
nocTHA       .
nocTJK       .
nocTOG      -0.4110503792
nocTPE      -0.1715835368
nocTTO      -0.1281211019
nocTUN      -0.0434936490
nocTUR       0.1046955361
nocUAE       .
nocUGA       .
nocUKR      -0.1682745796
nocURS       0.2012641195
nocURU       0.1491175915
nocUSA       0.3020235759
nocUZB      -0.0621028917
nocVEN      -0.3382137505
nocVIE       0.0411318346
nocWIF      -0.7094396716
nocYUG       0.1363231897
nocZAM       .
```

```
nocZIM       0.6785615265


Subset selection object
Call: regsubsets.formula(medals ~ sex + age + height + weight + noc,
    data = olympics_ord, nbest = 1, nvmax = 5, really.big = T)
146 Variables  (and intercept)
       Forced in Forced out
sexM       FALSE      FALSE
age        FALSE      FALSE
height     FALSE      FALSE
weight     FALSE      FALSE
nocALG     FALSE      FALSE
nocANZ     FALSE      FALSE
nocARG     FALSE      FALSE
nocARM     FALSE      FALSE
nocAUS     FALSE      FALSE
nocAUT     FALSE      FALSE
nocAZE     FALSE      FALSE
nocBAH     FALSE      FALSE
nocBAR     FALSE      FALSE
nocBDI     FALSE      FALSE
nocBEL     FALSE      FALSE
nocBER     FALSE      FALSE
nocBLR     FALSE      FALSE
nocBOT     FALSE      FALSE
nocBRA     FALSE      FALSE
nocBRN     FALSE      FALSE
nocBUL     FALSE      FALSE
nocCAN     FALSE      FALSE
nocCHI     FALSE      FALSE
nocCHN     FALSE      FALSE
nocCIV     FALSE      FALSE
nocCMR     FALSE      FALSE
nocCOL     FALSE      FALSE
nocCRC     FALSE      FALSE
nocCRO     FALSE      FALSE
nocCUB     FALSE      FALSE
nocCYP     FALSE      FALSE
nocCZE     FALSE      FALSE
nocDEN     FALSE      FALSE
nocDJI     FALSE      FALSE
nocDOM     FALSE      FALSE
```

```
nocECU     FALSE      FALSE
nocEGY     FALSE      FALSE
nocERI     FALSE      FALSE
nocESP     FALSE      FALSE
nocEST     FALSE      FALSE
nocETH     FALSE      FALSE
nocEUN     FALSE      FALSE
nocFIJ     FALSE      FALSE
nocFIN     FALSE      FALSE
nocFRA     FALSE      FALSE
nocFRG     FALSE      FALSE
nocGAB     FALSE      FALSE
nocGBR     FALSE      FALSE
nocGDR     FALSE      FALSE
nocGEO     FALSE      FALSE
nocGER     FALSE      FALSE
nocGHA     FALSE      FALSE
nocGRE     FALSE      FALSE
nocGRN     FALSE      FALSE
nocGUA     FALSE      FALSE
nocGUY     FALSE      FALSE
nocHAI     FALSE      FALSE
nocHKG     FALSE      FALSE
nocHUN     FALSE      FALSE
nocINA     FALSE      FALSE
nocIND     FALSE      FALSE
nocIOA     FALSE      FALSE
nocIRI     FALSE      FALSE
nocIRL     FALSE      FALSE
nocISL     FALSE      FALSE
nocISR     FALSE      FALSE
nocISV     FALSE      FALSE
nocITA     FALSE      FALSE
nocJAM     FALSE      FALSE
nocJOR     FALSE      FALSE
nocJPN     FALSE      FALSE
nocKAZ     FALSE      FALSE
nocKEN     FALSE      FALSE
nocKGZ     FALSE      FALSE
nocKOR     FALSE      FALSE
nocKOS     FALSE      FALSE
nocKSA     FALSE      FALSE
nocKUW     FALSE      FALSE
```

```
nocLAT     FALSE      FALSE
nocLIB     FALSE      FALSE
nocLIE     FALSE      FALSE
nocLTU     FALSE      FALSE
nocLUX     FALSE      FALSE
nocMAR     FALSE      FALSE
nocMAS     FALSE      FALSE
nocMDA     FALSE      FALSE
nocMEX     FALSE      FALSE
nocMGL     FALSE      FALSE
nocMKD     FALSE      FALSE
nocMNE     FALSE      FALSE
nocMOZ     FALSE      FALSE
nocMRI     FALSE      FALSE
nocNAM     FALSE      FALSE
nocNED     FALSE      FALSE
nocNGR     FALSE      FALSE
nocNIG     FALSE      FALSE
nocNOR     FALSE      FALSE
nocNZL     FALSE      FALSE
nocPAK     FALSE      FALSE
nocPAN     FALSE      FALSE
nocPAR     FALSE      FALSE
nocPER     FALSE      FALSE
nocPHI     FALSE      FALSE
nocPOL     FALSE      FALSE
nocPOR     FALSE      FALSE
nocPRK     FALSE      FALSE
nocPUR     FALSE      FALSE
nocQAT     FALSE      FALSE
nocROU     FALSE      FALSE
nocRSA     FALSE      FALSE
nocRUS     FALSE      FALSE
nocSCG     FALSE      FALSE
nocSEN     FALSE      FALSE
nocSGP     FALSE      FALSE
nocSLO     FALSE      FALSE
nocSRB     FALSE      FALSE
nocSRI     FALSE      FALSE
nocSUD     FALSE      FALSE
nocSUI     FALSE      FALSE
nocSUR     FALSE      FALSE
nocSVK     FALSE      FALSE
```

```
nocSWE     FALSE      FALSE
nocSYR     FALSE      FALSE
nocTAN     FALSE      FALSE
nocTCH     FALSE      FALSE
nocTGA     FALSE      FALSE
nocTHA     FALSE      FALSE
nocTJK     FALSE      FALSE
nocTOG     FALSE      FALSE
nocTPE     FALSE      FALSE
nocTTO     FALSE      FALSE
nocTUN     FALSE      FALSE
nocTUR     FALSE      FALSE
nocUAE     FALSE      FALSE
nocUGA     FALSE      FALSE
nocUKR     FALSE      FALSE
nocURS     FALSE      FALSE
nocURU     FALSE      FALSE
nocUSA     FALSE      FALSE
nocUZB     FALSE      FALSE
nocVEN     FALSE      FALSE
nocVIE     FALSE      FALSE
nocWIF     FALSE      FALSE
nocYUG     FALSE      FALSE
nocZAM     FALSE      FALSE
nocZIM     FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive


Subset selection object
Call: regsubsets.formula(medals ~ sex + age + height + weight + noc,
    data = olympics_ord, nbest = 1, nvmax = 5, really.big = T)
146 Variables  (and intercept)
       Forced in Forced out
sexM       FALSE      FALSE
age        FALSE      FALSE
height     FALSE      FALSE
weight     FALSE      FALSE
nocALG     FALSE      FALSE
nocANZ     FALSE      FALSE
nocARG     FALSE      FALSE
nocARM     FALSE      FALSE
nocAUS     FALSE      FALSE
```

```
nocAUT     FALSE      FALSE
nocAZE     FALSE      FALSE
nocBAH     FALSE      FALSE
nocBAR     FALSE      FALSE
nocBDI     FALSE      FALSE
nocBEL     FALSE      FALSE
nocBER     FALSE      FALSE
nocBLR     FALSE      FALSE
nocBOT     FALSE      FALSE
nocBRA     FALSE      FALSE
nocBRN     FALSE      FALSE
nocBUL     FALSE      FALSE
nocCAN     FALSE      FALSE
nocCHI     FALSE      FALSE
nocCHN     FALSE      FALSE
nocCIV     FALSE      FALSE
nocCMR     FALSE      FALSE
nocCOL     FALSE      FALSE
nocCRC     FALSE      FALSE
nocCRO     FALSE      FALSE
nocCUB     FALSE      FALSE
nocCYP     FALSE      FALSE
nocCZE     FALSE      FALSE
nocDEN     FALSE      FALSE
nocDJI     FALSE      FALSE
nocDOM     FALSE      FALSE
nocECU     FALSE      FALSE
nocEGY     FALSE      FALSE
nocERI     FALSE      FALSE
nocESP     FALSE      FALSE
nocEST     FALSE      FALSE
nocETH     FALSE      FALSE
nocEUN     FALSE      FALSE
nocFIJ     FALSE      FALSE
nocFIN     FALSE      FALSE
nocFRA     FALSE      FALSE
nocFRG     FALSE      FALSE
nocGAB     FALSE      FALSE
nocGBR     FALSE      FALSE
nocGDR     FALSE      FALSE
nocGEO     FALSE      FALSE
nocGER     FALSE      FALSE
nocGHA     FALSE      FALSE
```

```
nocGRE     FALSE      FALSE
nocGRN     FALSE      FALSE
nocGUA     FALSE      FALSE
nocGUY     FALSE      FALSE
nocHAI     FALSE      FALSE
nocHKG     FALSE      FALSE
nocHUN     FALSE      FALSE
nocINA     FALSE      FALSE
nocIND     FALSE      FALSE
nocIOA     FALSE      FALSE
nocIRI     FALSE      FALSE
nocIRL     FALSE      FALSE
nocISL     FALSE      FALSE
nocISR     FALSE      FALSE
nocISV     FALSE      FALSE
nocITA     FALSE      FALSE
nocJAM     FALSE      FALSE
nocJOR     FALSE      FALSE
nocJPN     FALSE      FALSE
nocKAZ     FALSE      FALSE
nocKEN     FALSE      FALSE
nocKGZ     FALSE      FALSE
nocKOR     FALSE      FALSE
nocKOS     FALSE      FALSE
nocKSA     FALSE      FALSE
nocKUW     FALSE      FALSE
nocLAT     FALSE      FALSE
nocLIB     FALSE      FALSE
nocLIE     FALSE      FALSE
nocLTU     FALSE      FALSE
nocLUX     FALSE      FALSE
nocMAR     FALSE      FALSE
nocMAS     FALSE      FALSE
nocMDA     FALSE      FALSE
nocMEX     FALSE      FALSE
nocMGL     FALSE      FALSE
nocMKD     FALSE      FALSE
nocMNE     FALSE      FALSE
nocMOZ     FALSE      FALSE
nocMRI     FALSE      FALSE
nocNAM     FALSE      FALSE
nocNED     FALSE      FALSE
nocNGR     FALSE      FALSE
```

```
nocNIG     FALSE      FALSE
nocNOR     FALSE      FALSE
nocNZL     FALSE      FALSE
nocPAK     FALSE      FALSE
nocPAN     FALSE      FALSE
nocPAR     FALSE      FALSE
nocPER     FALSE      FALSE
nocPHI     FALSE      FALSE
nocPOL     FALSE      FALSE
nocPOR     FALSE      FALSE
nocPRK     FALSE      FALSE
nocPUR     FALSE      FALSE
nocQAT     FALSE      FALSE
nocROU     FALSE      FALSE
nocRSA     FALSE      FALSE
nocRUS     FALSE      FALSE
nocSCG     FALSE      FALSE
nocSEN     FALSE      FALSE
nocSGP     FALSE      FALSE
nocSLO     FALSE      FALSE
nocSRB     FALSE      FALSE
nocSRI     FALSE      FALSE
nocSUD     FALSE      FALSE
nocSUI     FALSE      FALSE
nocSUR     FALSE      FALSE
nocSVK     FALSE      FALSE
nocSWE     FALSE      FALSE
nocSYR     FALSE      FALSE
nocTAN     FALSE      FALSE
nocTCH     FALSE      FALSE
nocTGA     FALSE      FALSE
nocTHA     FALSE      FALSE
nocTJK     FALSE      FALSE
nocTOG     FALSE      FALSE
nocTPE     FALSE      FALSE
nocTTO     FALSE      FALSE
nocTUN     FALSE      FALSE
nocTUR     FALSE      FALSE
nocUAE     FALSE      FALSE
nocUGA     FALSE      FALSE
nocUKR     FALSE      FALSE
nocURS     FALSE      FALSE
nocURU     FALSE      FALSE
```

```
nocUSA     FALSE      FALSE
nocUZB     FALSE      FALSE
nocVEN     FALSE      FALSE
nocVIE     FALSE      FALSE
nocWIF     FALSE      FALSE
nocYUG     FALSE      FALSE
nocZAM     FALSE      FALSE
nocZIM     FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive
         sexM age height weight nocALG nocANZ nocARG nocARM nocAUS nocAUT
1  ( 1 ) " "  " " " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "  " " " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "  " " " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "  " " " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "  " " " "    " "    " "    " "    " "    " "    " "    " "
         nocAZE nocBAH nocBAR nocBDI nocBEL nocBER nocBLR nocBOT nocBRA nocBRN
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocBUL nocCAN nocCHI nocCHN nocCIV nocCMR nocCOL nocCRC nocCRO nocCUB
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    "*"    " "    " "    " "    " "    " "    " "
         nocCYP nocCZE nocDEN nocDJI nocDOM nocECU nocEGY nocERI nocESP nocEST
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocETH nocEUN nocFIJ nocFIN nocFRA nocFRG nocGAB nocGBR nocGDR nocGEO
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    "*"    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    "*"    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    "*"    " "
         nocGER nocGHA nocGRE nocGRN nocGUA nocGUY nocHAI nocHKG nocHUN nocINA
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
```

```
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocIND nocIOA nocIRI nocIRL nocISL nocISR nocISV nocITA nocJAM nocJOR
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocJPN nocKAZ nocKEN nocKGZ nocKOR nocKOS nocKSA nocKUW nocLAT nocLIB
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocLIE nocLTU nocLUX nocMAR nocMAS nocMDA nocMEX nocMGL nocMKD nocMNE
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    "*"    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    "*"    " "    " "    " "    " "    " "    " "    " "    " "
         nocMOZ nocMRI nocNAM nocNED nocNGR nocNIG nocNOR nocNZL nocPAK nocPAN
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocPAR nocPER nocPHI nocPOL nocPOR nocPRK nocPUR nocQAT nocROU nocRSA
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocRUS nocSCG nocSEN nocSGP nocSLO nocSRB nocSRI nocSUD nocSUI nocSUR
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocSVK nocSWE nocSYR nocTAN nocTCH nocTGA nocTHA nocTJK nocTOG nocTPE
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
```

```
4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
         nocTTO nocTUN nocTUR nocUAE nocUGA nocUKR nocURS nocURU nocUSA nocUZB
1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    "*"    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "    "*"    " "    "*"    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "    "*"    " "    "*"    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "    "*"    " "    "*"    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "    "*"    " "    "*"    " "
         nocVEN nocVIE nocWIF nocYUG nocZAM nocZIM
1  ( 1 ) " "    " "    " "    " "    " "    " "
2  ( 1 ) " "    " "    " "    " "    " "    " "
3  ( 1 ) " "    " "    " "    " "    " "    " "
4  ( 1 ) " "    " "    " "    " "    " "    " "
5  ( 1 ) " "    " "    " "    " "    " "    " "
```

```r
m_all <- regsubsets(medals ~ sex  + age + height +  weight,
                    data = olympics_ord,
                    nbest = 1, nvmax = 5, really.big = T)
m_all
```

```
Subset selection object
Call: regsubsets.formula(medals ~ sex + age + height + weight, data = olympics_ord,
    nbest = 1, nvmax = 5, really.big = T)
4 Variables  (and intercept)
       Forced in Forced out
sexM       FALSE      FALSE
age        FALSE      FALSE
height     FALSE      FALSE
weight     FALSE      FALSE
1 subsets of each size up to 4
Selection Algorithm: exhaustive
```

```r
summary(m_all)
```

```
Subset selection object
Call: regsubsets.formula(medals ~ sex + age + height + weight, data = olympics_ord,
    nbest = 1, nvmax = 5, really.big = T)
4 Variables  (and intercept)
       Forced in Forced out
sexM       FALSE      FALSE
```

```
age         FALSE      FALSE
height      FALSE      FALSE
weight      FALSE      FALSE
1 subsets of each size up to 4
Selection Algorithm: exhaustive
         sexM age height weight
1  ( 1 ) " "  " " "*"    " "
2  ( 1 ) " "  "*" "*"    " "
3  ( 1 ) "*"  "*" "*"    " "
4  ( 1 ) "*"  "*" "*"    "*"
```

```
summary(m_all)$cp
```

```
[1] 8.033983 2.634207 3.027055 5.000000
```

Final Model

```
ordMod_final <-
  polr(factor(medals) ~ sex  + age + height , data = olympics_ord , method = "probit")
tidy(ordMod_final) %>%
  mutate(estimate = round(estimate , 9))
```

```
Re-fitting to get Hessian
```

```
# A tibble: 5 x 5
  term   estimate std.error statistic coef.type
  <chr>     <dbl>     <dbl>     <dbl> <chr>
1 sexM   -0.0196   0.0155      -1.27  coefficient
2 age    -0.00336  0.00129     -2.60  coefficient
3 height  0.00304  0.000673     4.52  coefficient
4 1|2     0.0192   0.118        0.163 scale
5 2|3     0.863    0.118        7.30  scale
```