BUDA 530 Assignment 3

Collin Edwards

2025-02-10

Complete the assignment below. Make sure you answer each part of the questions. Submit your responses on eCampus as both an .RMD file and a knitted .pdf file. This assignment is due by 2/4.

Problem 1

The Billionaires Statistics Dataset is a dataset from Kaggle (.csv included in eCampus attachments) that contains information on the world's billionaires and the countries in which they reside. You can find more information on the dataset using the link above.

```
library(readr)
Billionaires_Statistics_Dataset <- read_csv("~/Downloads/Billionaires Statistics Dataset.csv")
```

```
## Rows: 2640 Columns: 35
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (18): category, personName, country, city, source, industries, countryOf...
## dbl (16): rank, finalWorth, age, birthYear, birthMonth, birthDay, cpi_countr...
## lgl  (1): selfMade
##
## 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.
```

```
str(Billionaires_Statistics_Dataset)
```

```
## spc_tbl_ [2,640 x 35] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ rank                                       : num [1:2640] 1 2 3 4 5 6 7 8 9 10 ...
##  $ finalWorth                                 : num [1:2640] 211000 180000 114000 107000 106000 10400
##  $ category                                   : chr [1:2640] "Fashion & Retail" "Automotive" "Technolo
##  $ personName                                 : chr [1:2640] "Bernard Arnault & family" "Elon Musk" "
##  $ age                                        : num [1:2640] 74 51 59 78 92 67 81 83 65 67 ...
##  $ country                                    : chr [1:2640] "France" "United States" "United States"
##  $ city                                       : chr [1:2640] "Paris" "Austin" "Medina" "Lanai" ...
##  $ source                                     : chr [1:2640] "LVMH" "Tesla, SpaceX" "Amazon" "Oracle"
##  $ industries                                 : chr [1:2640] "Fashion & Retail" "Automotive" "Technolo
##  $ countryOfCitizenship                       : chr [1:2640] "France" "United States" "United States"
##  $ organization                               : chr [1:2640] "LVMH Moët Hennessy Louis Vuitton" "Tesl
##  $ selfMade                                   : logi [1:2640] FALSE TRUE TRUE TRUE TRUE TRUE ...
##  $ status                                     : chr [1:2640] "U" "D" "D" "U" ...
##  $ gender                                     : chr [1:2640] "M" "M" "M" "M" ...
##  $ birthDate                                 : chr [1:2640] "3/5/1949 0:00" "6/28/1971 0:00" "1/12/19
##  $ lastName                                  : chr [1:2640] "Arnault" "Musk" "Bezos" "Ellison" ...
##  $ firstName                                 : chr [1:2640] "Bernard" "Elon" "Jeff" "Larry" ...
##  $ title                                     : chr [1:2640] "Chairman and CEO" "CEO" "Chairman and Fo
##  $ date                                      : chr [1:2640] "4/4/2023 5:01" "4/4/2023 5:01" "4/4/2023
##  $ state                                     : chr [1:2640] NA "Texas" "Washington" "Hawaii" ...
##  $ residenceStateRegion                      : chr [1:2640] NA "South" "West" "West" ...
##  $ birthYear                                 : num [1:2640] 1949 1971 1964 1944 1930 ...
##  $ birthMonth                                : num [1:2640] 3 6 1 8 8 10 2 1 4 3 ...
##  $ birthDay                                  : num [1:2640] 5 28 12 17 30 28 14 28 19 24 ...
##  $ cpi_country                               : num [1:2640] 110 117 117 117 117 ...
##  $ cpi_change_country                        : num [1:2640] 1.1 7.5 7.5 7.5 7.5 7.5 7.5 3.6 7.7 7.5
##  $ gdp_country                               : chr [1:2640] "$2,715,518,274,227" "$21,427,700,000,000
##  $ gross_tertiary_education_enrollment       : num [1:2640] 65.6 88.2 88.2 88.2 88.2 88.2 88.2 40.2 1
##  $ gross_primary_education_enrollment_country: num [1:2640] 102 102 102 102 102 ...
##  $ life_expectancy_country                   : num [1:2640] 82.5 78.5 78.5 78.5 78.5 78.5 78.5 75 69
##  $ tax_revenue_country_country               : num [1:2640] 24.2 9.6 9.6 9.6 9.6 9.6 9.6 13.1 11.2 9
##  $ total_tax_rate_country                    : num [1:2640] 60.7 36.6 36.6 36.6 36.6 36.6 36.6 55.1 4
##  $ population_country                        : num [1:2640] 6.71e+07 3.28e+08 3.28e+08 3.28e+08 3.28e
##  $ latitude_country                          : num [1:2640] 46.2 37.1 37.1 37.1 37.1 ...
##  $ longitude_country                         : num [1:2640] 2.21 -95.71 -95.71 -95.71 -95.71 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   rank = col_double(),
##   ..   finalWorth = col_double(),
##   ..   category = col_character(),
##   ..   personName = col_character(),
##   ..   age = col_double(),
##   ..   country = col_character(),
##   ..   city = col_character(),
##   ..   source = col_character(),
##   ..   industries = col_character(),
##   ..   countryOfCitizenship = col_character(),
##   ..   organization = col_character(),
##   ..   selfMade = col_logical(),
##   ..   status = col_character(),
##   ..   gender = col_character(),
##   ..   birthDate = col_character(),
##   ..   lastName = col_character(),
##   ..   firstName = col_character(),
##   ..   title = col_character(),
##   ..   date = col_character(),
##   ..   state = col_character(),
##   ..   residenceStateRegion = col_character(),
##   ..   birthYear = col_double(),
##   ..   birthMonth = col_double(),
##   ..   birthDay = col_double(),
##   ..   cpi_country = col_double(),
##   ..   cpi_change_country = col_double(),
##   ..   gdp_country = col_character(),
##   ..   gross_tertiary_education_enrollment = col_double(),
##   ..   gross_primary_education_enrollment_country = col_double(),
##   ..   life_expectancy_country = col_double(),
##   ..   tax_revenue_country_country = col_double(),
##   ..   total_tax_rate_country = col_double(),
##   ..   population_country = col_double(),
##   ..   latitude_country = col_double(),
##   ..   longitude_country = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
summary(Billionaires_Statistics_Dataset)
```

```
##       rank        finalWorth       category          personName
##  Min.   :   1   Min.   :  1000   Length:2640        Length:2640
##  1st Qu.: 659   1st Qu.:  1500   Class :character   Class :character
##  Median :1312   Median :  2300   Mode  :character   Mode  :character
##  Mean   :1289   Mean   :  4624
##  3rd Qu.:1905   3rd Qu.:  4200
##  Max.   :2540   Max.   :211000
##
##       age           country              city               source
##  Min.   : 18.00   Length:2640        Length:2640        Length:2640
##  1st Qu.: 56.00   Class :character   Class :character   Class :character
##  Median : 65.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 65.14
##  3rd Qu.: 75.00
##  Max.   :101.00
##  NA’s   :65
##   industries        countryOfCitizenship organization       selfMade
##  Length:2640        Length:2640          Length:2640        Mode :logical
##  Class :character   Class :character     Class :character   FALSE:828
##  Mode  :character   Mode  :character     Mode  :character   TRUE :1812
##
##
##
##
##     status             gender           birthDate           lastName
##  Length:2640        Length:2640        Length:2640        Length:2640
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   firstName            title              date               state
##  Length:2640        Length:2640        Length:2640        Length:2640
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  residenceStateRegion   birthYear      birthMonth       birthDay
##  Length:2640          Min.   :1921   Min.   : 1.00   Min.   : 1.0
##  Class :character     1st Qu.:1948   1st Qu.: 2.00   1st Qu.: 1.0
##  Mode  :character     Median :1957   Median : 6.00   Median :11.0
##                       Mean   :1957   Mean   : 5.74   Mean   :12.1
##                        3rd Qu.:1966   3rd Qu.: 9.00   3rd Qu.:21.0
##                        Max.   :2004   Max.   :12.00   Max.   :31.0
##                        NA's   :76     NA's   :76      NA's   :76
##   cpi_country     cpi_change_country gdp_country
##  Min.   : 99.55   Min.   :-1.900     Length:2640
##  1st Qu.:117.24   1st Qu.: 1.700     Class :character
##  Median :117.24   Median : 2.900     Mode  :character
##  Mean   :127.76   Mean   : 4.364
##  3rd Qu.:125.08   3rd Qu.: 7.500
##  Max.   :288.57   Max.   :53.500
##  NA's   :184      NA's   :184
##  gross_tertiary_education_enrollment gross_primary_education_enrollment_country
##  Min.   :  4.00                      Min.   : 84.7
##  1st Qu.: 50.60                      1st Qu.:100.2
##  Median : 65.60                      Median :101.8
##  Mean   : 67.23                      Mean   :102.9
##  3rd Qu.: 88.20                      3rd Qu.:102.6
##  Max.   :136.60                      Max.   :142.1
##  NA's   :182                         NA's   :181
##  life_expectancy_country tax_revenue_country_country total_tax_rate_country
##  Min.   :54.30           Min.   : 0.10               Min.   :  9.90
##  1st Qu.:77.00           1st Qu.: 9.60               1st Qu.: 36.60
##  Median :78.50           Median : 9.60               Median : 41.20
##  Mean   :78.12           Mean   :12.55               Mean   : 43.96
##  3rd Qu.:80.90           3rd Qu.:12.80               3rd Qu.: 59.10
##  Max.   :84.20           Max.   :37.20               Max.   :106.30
##  NA's   :182             NA's   :183                 NA's   :182
##  population_country  latitude_country longitude_country
##  Min.   :3.802e+04   Min.   :-40.90   Min.   :-106.35
##  1st Qu.:6.683e+07   1st Qu.: 35.86   1st Qu.: -95.71
##  Median :3.282e+08   Median : 37.09   Median :  10.45
##  Mean   :5.102e+08   Mean   : 34.90   Mean   :  12.58
##  3rd Qu.:1.366e+09   3rd Qu.: 40.46   3rd Qu.: 104.20
##  Max.   :1.398e+09   Max.   : 61.92   Max.   : 174.89
##  NA's   :164         NA's   :164      NA's   :164
```

For this analysis, consider a scenario where you work for a wealth management firm that is looking to expand its operations to new countries. Your job is to understand what factors are associated with the number of billionaires in a country. Ultimately your firm will cross reference this information with industry forecasts on country GDP growth, etc. to determine which countries are most likely to have the highest growth in the number of billionaires in the next 10 years (outside the scope of this assignment). For now, you will build a model focusing on the factors associated with the number of billionaires in a country.

Before we can model this, we must summarize the data by country. We will count the number of billionaires per country and use the first observation for each country to summarize the country specific statistics. To do this we use the `tidyverse` package (note, you do not have to understand this code but if you are interested in learning more about `tidyverse` see the corresponding Module X section).

Now you have the summarized dataset `BillionairesByCountry`. Use this dataset to:

(1) Build a Poisson regression model to predict the number of billionaires per country (`count_billionaires`) using the other variables in the dataset (except for `country`). Note: It is industry standard to use a `log` transform when using large financial metrics such as `gdp_country` and `tax_revenue_country_country` as predictors so make sure you do that. Note: you will have to deal with "NAs", do so using `na.omit`.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
# summarize the dataset by country and fix the GDP column
BillionairesByCountry <- Billionaires_Statistics_Dataset %>%
  select(country,
         gdp_country,
         gross_tertiary_education_enrollment,
         gross_primary_education_enrollment_country,
         life_expectancy_country,
         tax_revenue_country_country,
         total_tax_rate_country,
         population_country,
         latitude_country,
         longitude_country) %>%  # select only the columns we are interested in
  group_by(country) %>%       # group by country
  summarise(count_billionaires = n(),
            gdp_country = first(gdp_country),
            gross_tertiary_education_enrollment = first(gross_tertiary_education_enrollment),
            gross_primary_education_enrollment_country = first(gross_primary_education_enrollment_count
            life_expectancy_country = first(life_expectancy_country),
            tax_revenue_country_country = first(tax_revenue_country_country),
            total_tax_rate_country = first(total_tax_rate_country),
            population_country = first(population_country),
            latitude_country = first(latitude_country),
            longitude_country = first(longitude_country)) %>%  # summarize by country
  mutate(gdp_country = as.numeric(gsub("\\$|\\,", "", gdp_country)))  # fix a problem with GDP being a

# checking the dataset structure and summary
str(BillionairesByCountry)
```

```
## tibble [79 x 11] (S3: tbl_df/tbl/data.frame)
##  $ country                                   : chr [1:79] "Algeria" "Andorra" "Argentina" "Armenia"
##  $ count_billionaires                        : int [1:79] 1 1 4 1 43 11 2 1 3 2 ...
##  $ gdp_country                               : num [1:79] 1.70e+11 3.15e+09 4.50e+11 1.37e+10 1.39e+
##  $ gross_tertiary_education_enrollment       : num [1:79] 51.4 NA 90 54.6 113.1 ...
##  $ gross_primary_education_enrollment_country: num [1:79] 109.9 106.4 109.7 92.7 100.3 ...
##  $ life_expectancy_country                   : num [1:79] 76.7 NA 76.5 74.9 82.7 81.6 NA 77.2 81.6 N
##  $ tax_revenue_country_country               : num [1:79] 37.2 NA 10.1 20.9 23 25.4 NA 4.2 24 NA ...
##  $ total_tax_rate_country                    : num [1:79] 66.1 NA 106.3 22.6 47.4 ...
##  $ population_country                        : num [1:79] 43053054 77142 44938712 2957731 25766605 .
##  $ latitude_country                          : num [1:79] 28 42.5 -38.4 40.1 -25.3 ...
##  $ longitude_country                         : num [1:79] 1.66 1.52 -63.62 45.04 133.78 ...
```

```r
summary(BillionairesByCountry)
```

```
##    country          count_billionaires  gdp_country
##  Length:79          Min.   :  1.00     Min.   :3.154e+09
##  Class :character   1st Qu.:  1.50     1st Qu.:1.154e+11
##  Mode  :character   Median :  5.00     Median :3.359e+11
##                     Mean   : 33.42     Mean   :1.269e+12
##                     3rd Qu.: 26.00     3rd Qu.:7.931e+11
##                     Max.   :754.00     Max.   :2.143e+13
##                                        NA’s   :11
##  gross_tertiary_education_enrollment gross_primary_education_enrollment_country
##  Min.   :  4.00                      Min.   : 84.7
##  1st Qu.: 36.42                      1st Qu.:100.0
##  Median : 60.85                      Median :102.5
##  Mean   : 57.48                      Mean   :103.5
##  3rd Qu.: 80.38                      3rd Qu.:106.2
##  Max.   :136.60                      Max.   :142.1
##  NA’s   :13                          NA’s   :12
##  life_expectancy_country tax_revenue_country_country total_tax_rate_country
##  Min.   :54.30           Min.   : 0.1                Min.   :  9.90
##  1st Qu.:75.08           1st Qu.:12.5                1st Qu.: 28.90
##  Median :77.60           Median :17.1                Median : 38.00
##  Mean   :77.33           Mean   :17.5                Mean   : 39.10
##  3rd Qu.:81.67           3rd Qu.:23.0                3rd Qu.: 46.92
##  Max.   :84.20           Max.   :37.2                Max.   :106.30
##  NA’s   :13              NA’s   :14                  NA’s   :13
##  population_country  latitude_country longitude_country
##  Min.   :3.802e+04   Min.   :-40.90   Min.   :-106.347
##  1st Qu.:5.790e+06   1st Qu.: 15.42   1st Qu.:   5.086
##  Median :2.256e+07   Median : 36.65   Median :  22.381
##  Mean   :8.565e+07   Mean   : 28.69   Mean   :  29.040
##  3rd Qu.:6.193e+07   3rd Qu.: 47.64   3rd Qu.:  58.128
##  Max.   :1.398e+09   Max.   : 61.92   Max.   : 174.886
##  NA’s   :11          NA’s   :11       NA’s   :11
```

```r
View(BillionairesByCountry)

# counting NAs before removal
sum(is.na(BillionairesByCountry))
```

```
## [1] 110
```

```r
# removing observations with NA values
BillionairesByCountry_clean <- na.omit(BillionairesByCountry)

# create new columns for the log-transformed variables
BillionairesByCountry_clean <- BillionairesByCountry_clean %>%
  mutate(log_gdp_country = log(gdp_country),
         log_tax_revenue = log(tax_revenue_country_country))

# verifying that the new variables exist
names(BillionairesByCountry_clean)
```

```
##  [1] "country"
##  [2] "count_billionaires"
##  [3] "gdp_country"
##  [4] "gross_tertiary_education_enrollment"
##  [5] "gross_primary_education_enrollment_country"
##  [6] "life_expectancy_country"
##  [7] "tax_revenue_country_country"
##  [8] "total_tax_rate_country"
##  [9] "population_country"
## [10] "latitude_country"
## [11] "longitude_country"
## [12] "log_gdp_country"
## [13] "log_tax_revenue"
```

```
# Expected to see "log_gdp_country" and "log_tax_revenue" in the output

# fitting the Poisson model using the pre-transformed variables
model_poisson <- glm(count_billionaires ~ log_gdp_country +
                                      gross_tertiary_education_enrollment +
                                      gross_primary_education_enrollment_country +
                                      life_expectancy_country +
                                      log_tax_revenue +
                                      total_tax_rate_country +
                                      population_country +
                                      latitude_country +
                                      longitude_country,
                     data = BillionairesByCountry_clean,
                     family = poisson())
summary(model_poisson)
```

```
##
## Call:
## glm(formula = count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment +
##     gross_primary_education_enrollment_country + life_expectancy_country +
##     log_tax_revenue + total_tax_rate_country + population_country +
##     latitude_country + longitude_country, family = poisson(),
##     data = BillionairesByCountry_clean)
##
## Coefficients:
##                                               Estimate Std. Error z value
## (Intercept)                                 -2.618e+01  1.147e+00 -22.827
## log_gdp_country                              9.490e-01  2.755e-02  34.441
## gross_tertiary_education_enrollment          8.596e-03  1.679e-03   5.119
## gross_primary_education_enrollment_country   4.402e-02  4.995e-03   8.812
## life_expectancy_country                     -1.097e-02  7.978e-03  -1.375
## log_tax_revenue                             -6.102e-03  4.075e-02  -0.150
## total_tax_rate_country                      -2.736e-02  2.521e-03 -10.850
## population_country                           6.916e-10  9.256e-11   7.471
## latitude_country                             1.689e-03  1.338e-03   1.262
## longitude_country                           -5.303e-05  3.914e-04  -0.135
##                                             Pr(>|z|)
## (Intercept)                                  < 2e-16 ***
## log_gdp_country                              < 2e-16 ***
## gross_tertiary_education_enrollment         3.08e-07 ***
## gross_primary_education_enrollment_country  < 2e-16 ***
## life_expectancy_country                       0.169    
## log_tax_revenue                               0.881    
## total_tax_rate_country                      < 2e-16 ***
## population_country                         7.93e-14 ***
## latitude_country                              0.207    
## longitude_country                             0.892    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7527.27  on 64  degrees of freedom
## Residual deviance:  362.29  on 55  degrees of freedom
## AIC: 637.97
##
## Number of Fisher Scoring iterations: 5
```

Problem 1 Answer I modeled counted the `billionaires` using the remaining predictors in the dataset. I used a Poisson regression model because the response variable is a count variable. I used a log transform for the financial metrics `gdp_country` and `tax_revenue_country_country` because they are large financial metrics. *I omitted "NAs" from the dataset because I did not want to include missing data in the model and since there were not a lot of NA's I omitted them instead of interpolating.* I did not use `country` as a predictor because it is a categorical variable with many levels and would not be appropriate for a Poisson regression model.

(2) Use `step` for model feature selection.

```
model_poisson_step <- step(model_poisson)
```

```
## Start:  AIC=637.97
## count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment + 
##     gross_primary_education_enrollment_country + life_expectancy_country + 
##     log_tax_revenue + total_tax_rate_country + population_country + 
##     latitude_country + longitude_country
##
##                                              Df Deviance     AIC
## - longitude_country                           1   362.31  635.99
## - log_tax_revenue                             1   362.32  635.99
## - latitude_country                            1   363.90  637.58
## - life_expectancy_country                     1   364.15  637.82
## <none>                                            362.29  637.97
## - gross_tertiary_education_enrollment         1   387.97  661.65
## - population_country                          1   418.82  692.50
## - gross_primary_education_enrollment_country  1   436.75  710.42
## - total_tax_rate_country                      1   492.98  766.65
## - log_gdp_country                             1  1930.98 2204.65
##
## Step:  AIC=635.99
## count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment + 
##     gross_primary_education_enrollment_country + life_expectancy_country + 
##     log_tax_revenue + total_tax_rate_country + population_country + 
##     latitude_country
## 
##                                              Df Deviance     AIC
## - log_tax_revenue                             1   362.33  634.01
## - latitude_country                            1   363.99  635.67
## <none>                                            362.31  635.99
## - life_expectancy_country                     1   364.48  636.16
## - gross_tertiary_education_enrollment         1   388.77  660.44
## - population_country                          1   436.85  708.53
## - gross_primary_education_enrollment_country  1   449.00  720.67
## - total_tax_rate_country                      1   497.91  769.59
## - log_gdp_country                             1  2881.38 3153.05
## 
## Step:  AIC=634.01
## count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment + 
##     gross_primary_education_enrollment_country + life_expectancy_country + 
##     total_tax_rate_country + population_country + latitude_country
## 
##                                              Df Deviance    AIC
## - latitude_country                            1   364.00  633.7
## <none>                                            362.33  634.0
## - life_expectancy_country                     1   364.55  634.2
## - gross_tertiary_education_enrollment         1   389.46  659.1
## - population_country                          1   436.91  706.6
## - gross_primary_education_enrollment_country  1   449.03  718.7
## - total_tax_rate_country                      1   509.24  778.9
## - log_gdp_country                             1  3152.69 3422.4
## 
## Step:  AIC=633.67
## count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment + 
##     gross_primary_education_enrollment_country + life_expectancy_country + 
##     total_tax_rate_country + population_country
## 
##                                              Df Deviance    AIC
## - life_expectancy_country                     1    365.3  632.9
## <none>                                             364.0  633.7
## - gross_tertiary_education_enrollment         1    390.1  657.8
## - population_country                          1    439.3  707.0
## - gross_primary_education_enrollment_country  1    449.4  717.1
## - total_tax_rate_country                      1    522.2  789.9
## - log_gdp_country                             1   3252.9 3520.5
## 
## Step:  AIC=632.94
## count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment + 
##     gross_primary_education_enrollment_country + total_tax_rate_country + 
##     population_country
## 
##                                              Df Deviance    AIC
## <none>                                             365.3  632.9
## - gross_tertiary_education_enrollment         1    390.2  655.8
## - population_country                          1    453.4  719.0
## - gross_primary_education_enrollment_country  1    459.8  725.5
## - total_tax_rate_country                      1    543.3  809.0
## - log_gdp_country                             1   3333.5 3599.1
```

```r
summary(model_poisson_step)
```

```
##
## Call:
## glm(formula = count_billionaires ~ log_gdp_country + gross_tertiary_education_enrollment +
##     gross_primary_education_enrollment_country + total_tax_rate_country +
##     population_country, family = poisson(), data = BillionairesByCountry_clean)
##
## Coefficients:
##                                               Estimate Std. Error z value
## (Intercept)                                 -2.731e+01  7.920e-01 -34.476
## log_gdp_country                              9.572e-01  2.116e-02  45.245
## gross_tertiary_education_enrollment          7.852e-03  1.562e-03   5.028
## gross_primary_education_enrollment_country   4.564e-02  4.504e-03  10.133
## total_tax_rate_country                      -2.859e-02  2.244e-03 -12.739
## population_country                           7.125e-10  7.646e-11   9.319
##                                             Pr(>|z|)
## (Intercept)                                  < 2e-16 ***
## log_gdp_country                              < 2e-16 ***
## gross_tertiary_education_enrollment         4.96e-07 ***
## gross_primary_education_enrollment_country   < 2e-16 ***
## total_tax_rate_country                       < 2e-16 ***
## population_country                           < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7527.27  on 64  degrees of freedom
## Residual deviance:  365.27  on 59  degrees of freedom
## AIC: 632.94
##
## Number of Fisher Scoring iterations: 5
```

Problem 2 Answer In order to remove the non-significant predictors we went with the `step` function to perform model feature selection like we did in the previous assignment. This function removes the non-significant predictors from the model using AIC as the criterion. Occram’s razor(our favorite principle) states that the simplest model that explains the data is the best model. This is why we used the `step` function to remove the non-significant predictors from the model.

(3) Check for overdispersion/ underdispersion and account for it in your final model if necessary.

```r
# Calculate the dispersion parameter
dispersion <- sum(residuals(model_poisson_step, type = "pearson")^2) / model_poisson_step$df.residual
dispersion
```

```
## [1] 7.034681
```

```r
# If overdispersion is present (e.g., dispersion > 1.5), refit using a quasi-Poisson family.
if(dispersion > 1.5) {
  model_final <- glm(count_billionaires ~ ., data = model.frame(model_poisson_step),
                     family = quasipoisson())
} else {
  model_final <- model_poisson_step
}
summary(model_final)
```

```
##
## Call:
## glm(formula = count_billionaires ~ ., family = quasipoisson(),
##     data = model.frame(model_poisson_step))
##
## Coefficients:
##                                               Estimate Std. Error t value
## (Intercept)                                 -2.731e+01  2.101e+00 -12.999
## log_gdp_country                              9.572e-01  5.611e-02  17.059
## gross_tertiary_education_enrollment          7.852e-03  4.142e-03   1.896
## gross_primary_education_enrollment_country   4.564e-02  1.195e-02   3.820
## total_tax_rate_country                      -2.859e-02  5.952e-03  -4.803
## population_country                           7.125e-10  2.028e-10   3.514
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## log_gdp_country                              < 2e-16 ***
## gross_tertiary_education_enrollment         0.062908 .
## gross_primary_education_enrollment_country  0.000323 ***
## total_tax_rate_country                      1.11e-05 ***
## population_country                          0.000855 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for quasipoisson family taken to be 7.034681)
##
##     Null deviance: 7527.27  on 64  degrees of freedom
## Residual deviance:  365.27  on 59  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

Problem 3 Answer I checked for overdispersion by calculating the dispersion parameter. If the dispersion parameter is greater than 1.5, then overdispersion is present and I refit the model using a quasi-Poisson family. If the dispersion parameter is less than 1.5, then I used the original Poisson model. This accounts for overdispersion/ underdispersion in the final model.

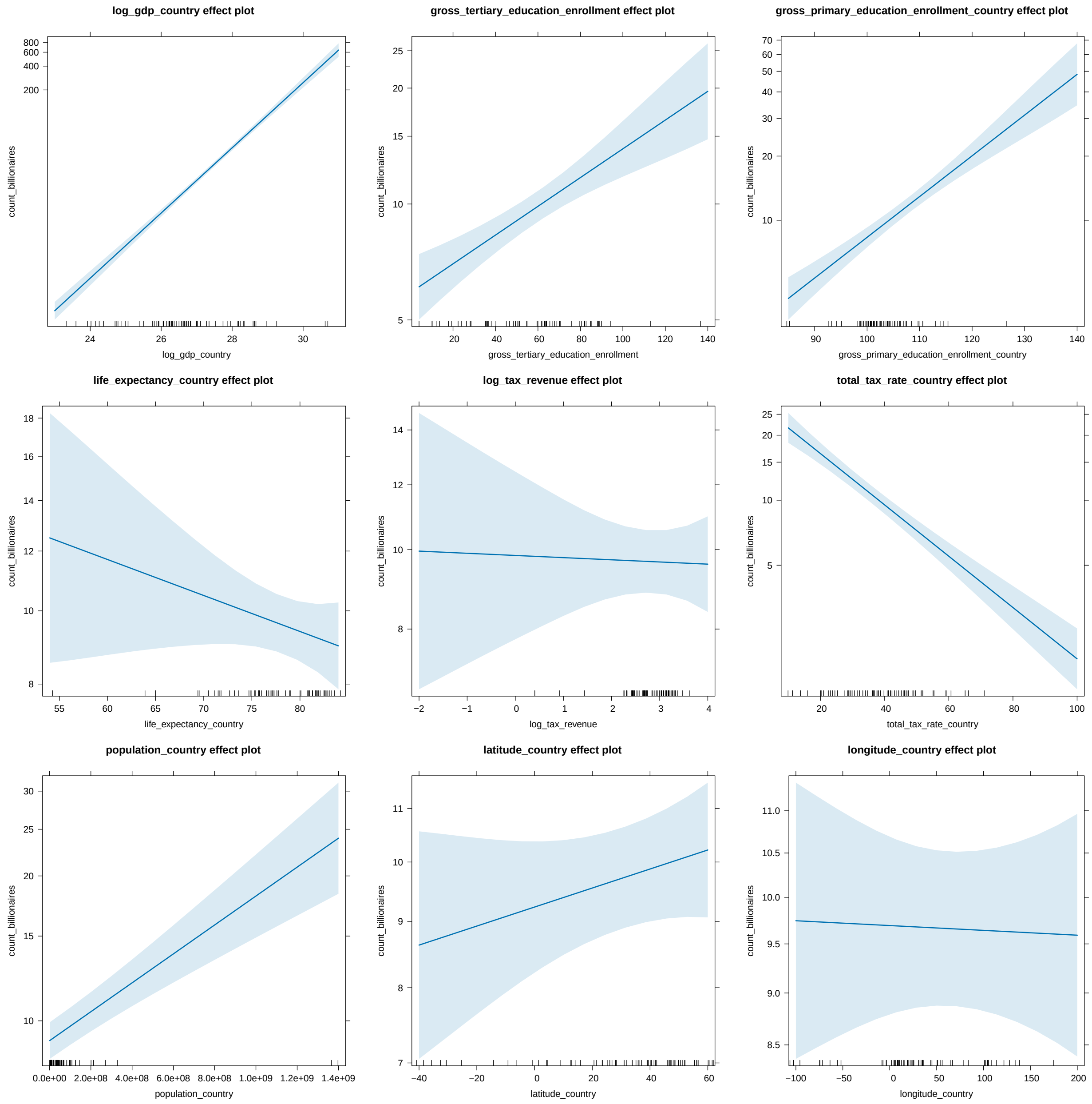
(3b) Use the `effects` library to create effects plots for the final model (remember to adjust the `fig.height` and `fig.width` chunk options so that the plots look nice).

```r
library(effects)
```

```
## Loading required package: carData
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```r
plot(allEffects(model_poisson))
```



Problem 3b Answer I modified the figure size and the graph size thanks to the collab notes. I also used the `effects` library to create effects plots for the final model. The effects plots show the relationship between the predictors and the response variable. The effects plots are easier to interpret than the coefficients because they show the relationship between the predictors and the response variable in a visual way.

(4) Write a summary of your methodology for your direct supervisor. You direct supervisor has a similar statistical background as you, but does not use it on a daily basis so you will need to briefly refresh them on the statistical concepts you used and explain your methodology in detail. You know from previous experience that your supervisor is will be interested in why you did not use `country` as a predictor, why you used a `log` transform for large financial metrics, why you chose to omit “NAs”, and how you accounted for overdispersion/ underdispersion.

Problem 4 Answer *Methodology Summary:*

I built a Poisson regression model to predict the number of billionaires per country based on several country-specific metrics. Financial metrics like GDP and tax revenue were log-transformed to account for their large scale and to linearize their relationships with the outcome. The country variable was excluded as it is a categorical identifier that does not generalize to new countries. I removed missing data using na.omit to ensure the integrity of the model fitting.

```
After fitting the initial model, I used stepwise selection (step()) to eliminate non-significant predict

Finally, I generated effects plots to visualize the influence of each predictor on the number of billion
```

(5) Write a summary of the key findings for the VP of your firm that you report under who is particularly interested in this project. This VP also has a similar statistical background to you and your supervisor, but has not used in years. Further your VP is an extremely busy person and does not like reading long reports; however they will ask an annoying number of questions if they do not understand something. Strike a balance between including how you arrived at the results and the results themselves. Are there certain details you can footnote rather include directly in the body of the report? Are their resources on the web you can hyperlink for additional information rather than recreate the wheel (i.e. Poisson regression - Wikipedia).

Problem 5 Answer *Key Findings:*

The analysis shows that several country-level factors are significantly associated with the number of billionaires. In particular, financial metrics (GDP and tax revenue, both log-transformed) and other socio-economic indicators such as education enrollment, life expectancy, and population have strong predictive power.

A notable technical challenge was overdispersion, which we addressed by switching to a quasi-Poisson model. This adjustment ensures that our standard errors and confidence intervals are accurate.

For further details on Poisson regression and overdispersion, please refer to Poisson regression - Wikipedia Poisson regression - Wikipedia. Additional in-depth methodology is available in our internal documentation if required.

Please let me know if you have any questions.

Note: The full methodology and model details are available upon request.

(6) Suppose your VP responds to your report with the email below. Write a brief response to this email (do not actually write any additional code or do any additional analysis for this part).

"Great Analysis!

As a follow-up, I'd be interested in extending the model to counties without billionaires so that we don't miss out on emerging opportunities. A junior analyst should be able to pull the metrics you need for most of the countries not listed here. I don't think you'll be able to use Poisson regression for this since the response variable will be 0 for all of these countries. It's been a while since I've had stats, can you remind me what model can be used for this case?

Additionally, I'd be interested in a comparison between the models with and without the additional countries. I'm not sure how to do this (although cross validation seems to be ringing a bell). I'm sure you can figure it out; let me know what you plan to try and I'll see if jogs my memory.

Lastly, I know metrics like "GDP" aren't going to be available for all countries (i.e. Hong Kong), but I'd like to see if we can estimate a value for these countries so we don't exclude them from consideration. I'm going to reach out to some consultants to see if they have any ideas; do you have any concerns I should bring up with them? In particular, would you want to use these estimates for training your model or just for prediction?

Thanks!"

Problem 6 Answer *Response:*

Thank you for your feedback. To extend the model to include countries without any billionaires (where the observed count is zero), a Zero-Inflated Poisson (ZIP) model would be more appropriate because it can handle excess zeros by modeling both a binary process (for structural zeros) and a count process.

For comparing models that include and exclude these additional countries, I would suggest using cross-validation to assess predictive performance. Cross-validation would allow us to compare model accuracy on hold-out data, which is more appropriate than comparing AIC values across models fitted with different datasets.

Regarding missing financial metrics such as GDP for some countries, my recommendation would be to use the estimated values solely for prediction. Incorporating them into the training set might introduce bias unless the estimates are highly reliable. I would be happy to discuss this further.

Thanks!

Question 2

The dataset `happy` in the `faraway` package is about 39 students from the University of Chicago MBA cohort.

```
library(faraway)
data(happy)
str(happy)
```

```
## ’data.frame':    39 obs. of  5 variables:
##  $ happy: num  10 8 8 8 4 9 8 6 5 4 ...
##  $ money: num  36 47 53 35 88 175 175 45 35 55 ...
##  $ sex  : num  0 1 0 1 1 1 1 0 1 1 ...
##  $ love : num  3 3 3 3 1 3 3 2 2 1 ...
##  $ work : num  4 1 5 3 2 4 4 3 2 4 ...
```

```
summary(happy)
```

```
##      happy            money             sex              love      
##  Min.   : 2.000   Min.   :  0.00   Min.   :0.0000   Min.   :1.000  
##  1st Qu.: 5.000   1st Qu.: 42.50   1st Qu.:0.0000   1st Qu.:2.000  
##  Median : 7.000   Median : 50.00   Median :1.0000   Median :3.000  
##  Mean   : 6.744   Mean   : 62.15   Mean   :0.6923   Mean   :2.462  
##  3rd Qu.: 8.000   3rd Qu.: 78.00   3rd Qu.:1.0000   3rd Qu.:3.000  
##  Max.   :10.000   Max.   :175.00   Max.   :1.0000   Max.   :3.000  
##       work      
##  Min.   :1.000  
##  1st Qu.:3.000  
##  Median :4.000  
##  Mean   :3.359  
##  3rd Qu.:4.000  
##  Max.   :5.000  
```

```
help(happy)
```

We want to explain the effects of the other information on the happiness of the students. The variable `happy` is a numeric variable that ranges from 0 to 10, with 10 being the happiest. This is recorded as a number, so we must first convert it an ordered factor variable using the code below:

```r
# This code converts the happy variable from a numeric variable to an ordered factor variable
myHappy <-happy %>%
  mutate(happy = factor(happy, ordered = TRUE))
```

Consider the following models:

```r
# Ordinal Regression
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

```r
mod1<-polr(happy~.,data=myHappy)
summary(mod1)
```

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

## Call:
## polr(formula = happy ~ ., data = myHappy)
##
## Coefficients:
##          Value Std. Error t value
## money  0.02246    0.01066  2.1064
## sex   -0.47344    0.79498 -0.5955
## love   3.60765    0.80114  4.5031
## work   0.88751    0.40826  2.1739
##
## Intercepts:
##      Value   Std. Error t value
## 2|3   5.4708  1.9891     2.7504
## 3|4   6.4684  1.9223     3.3650
## 4|5   9.1591  2.1698     4.2212
## 5|6  10.9725  2.3213     4.7268
## 6|7  11.5113  2.3720     4.8530
## 7|8  13.5433  2.6673     5.0776
## 8|9  17.2909  3.1454     5.4971
## 9|10 19.0112  3.3270     5.7142
##
## Residual Deviance: 94.86029
## AIC: 118.8603
```

```r
# Multinomial Regression
library(nnet)
mod2<-multinom(happy~.,data=myHappy)
```

```
## # weights:  54 (40 variable)
## initial  value 85.691759
## iter  10 value 68.212870
## iter  20 value 38.631288
## iter  30 value 28.889527
## iter  40 value 27.437462
## iter  50 value 26.714973
## iter  60 value 26.708293
## iter  70 value 26.703682
## final  value 26.703644
## converged
```

```
summary(mod2)
```

```
## Call:
## multinom(formula = happy ~ ., data = myHappy)
##
## Coefficients:
##    (Intercept)    money        sex        love      work
## 3     95.34717 8.207435   47.53907 -121.974569 -83.37503
## 4    108.15356 6.448616  126.62067 -144.128458 -19.45492
## 5    103.43665 6.504835   17.26409  -89.127605 -18.72953
## 6    -56.16590 6.632862  -29.02349   -9.832305 -20.71266
## 7     23.22477 6.557004   16.88244  -51.573474 -17.95274
## 8    -94.78326 6.586027  -39.30831    6.352961 -17.99809
## 9   -213.80005 6.596050   16.52875  -14.640589  13.18524
## 10  -149.75016 4.278169 -142.97177   95.281251 -45.18974
##
## Std. Errors:
##    (Intercept)       money          sex        love        work
## 3    0.3254856 27.67909763 3.254856e-01 0.328049419 0.328049254
## 4    0.7997975  4.62411790 7.997975e-01 1.571938267 1.210306605
## 5    0.7465953  4.62366455 1.682088e+00 1.493190570 0.865609267
## 6    2.3786859  4.62378395 4.087284e+00 1.201823878 1.931873301
## 7    1.6205287  4.62359602 1.651471e+00 0.829190888 0.703236472
## 8    1.2289108  4.62360855 1.892117e+00 1.036603125 0.814853388
## 9    0.1035101  4.62362077 1.035101e-01 0.310530188 0.414040251
## 10   0.0015474  0.06183351 9.725159e-08 0.004642199 0.007414518
##
## Residual Deviance: 53.40729
## AIC: 133.4073
```

For this problem:

(1) Compare the two models summaries. You do not need to interpret each coefficient, but pick a particular variable and explain how the interpretation of the coefficient differs between the two models. What is different about these models? What is the same/ similar?

Problem 1 Answer In the ordinal regression model (using `polr`), the coefficients represent the change in the log odds of being in a higher happiness category per unit increase in the predictor under the proportional odds assumption. In the multinomial regression model (using `multinom`), each non-baseline category receives its own set of coefficients so that for a given predictor the effect is estimated separately for each comparison with the baseline. For instance, if we examine a variable like `health` (if present), the ordinal model gives one coefficient that applies across all thresholds, whereas the multinomial model yields multiple coefficients corresponding to each level of happiness relative to the baseline.

(2) Can AIC and Deviance be used to compare these two models (Hint: NO!)? Why or why not? What can be used to compare these two models (you do not need to write the code you this, just explain the methodology you would use)?

Problem 2 Answer AIC and Deviance cannot be used to compare the ordinal and multinomial regression models directly because they are different types of models with different assumptions. AIC and Deviance are used to compare models of the same type (e.g., two Poisson models or two linear regression models). To compare the ordinal and multinomial models, we can use a likelihood ratio test (LRT) to determine if the more complex model (multinomial) provides a significantly better fit than the simpler model (ordinal). The LRT compares the log-likelihoods of the two models to assess whether the additional complexity of the multinomial model is justified by the data.

(3) Notice that we changed `happy` from a numeric variable to an ordered factor variable. What is the difference between an ordered factor variable and a numeric variable? Why is it important to use an ordered factor variable for `happy` in this case? Are there other variables that can benefit from a similar conversion? If so, what are they and what do they need converted to? (You do not need to write any code for this question, just answer conceptually).

Problem 3 Answer An ordered factor variable is a categorical variable with a natural ordering (e.g., low, medium, high) that is treated as a single variable with multiple levels. In contrast, a numeric variable is a continuous variable that can take on any value within a range. It is important to use an ordered factor variable for `happy` in this case because the happiness variable is ordinal, meaning the levels have a meaningful order (e.g., $1 < 2 < 3$). Using an ordered factor variable ensures that the model treats the levels as ordered categories rather than arbitrary numbers.

(4) Create effects plots for each model (remember to adjust the `fig.height` and `fig.width` chunk options so that the plots look nice). What do you notice about the effects plots? What is different between these two models? What is the same? Are the effects easier or harder to interpret than the coefficients? Why or why not?

Problem 4 Answer The effects plots for both models display how the predictors influence the probability of each happiness level. In the ordinal model, the effect is assumed to be consistent across thresholds (proportional odds), while the multinomial model shows separate curves for each non-baseline category. Effects plots are generally easier to interpret because they visualize predicted probabilities rather than raw coefficients. The ordinal model's effects plot shows a single curve for each predictor, while the multinomial model's effects plot displays multiple curves for each predictor, making it more complex to interpret.

(5) Which model do you prefer for this case? Why?

Problem 5 Answer For the happy dataset, I prefer the ordinal regression model because the response is naturally ordered. The ordinal model is more parsimonious, relies on the proportional odds assumption (which is reasonable in this context), and provides a more straightforward interpretation compared to the multinomial model. The ordinal model also has the advantage of treating the happiness levels as ordered categories, which aligns with the nature of the response variable.