Assignment 1 - STATS 720

Question 1: Olympic medals

Analyze the Olympic data set found here (raw download from here). The variables are:

- `team`: (approximately) country
- `year`
- `medal` (bronze/gold/silver)
- `n`: medal count
- `gdp`: GDP in const 2015 US$ (billions)
- `pop`: population size (millions)

a. State which possible predictor variables you're going to include; justify your choice (refer to Harrell chapter 4 for rules of thumb about appropriate numbers of predictors).

   - decide whether you're going to predict gold medals only, total medal count, or some weighted average of medals (e.g. `4*G+5*S+2*B`).

Load the data

```
library(RCurl)
library(readr)

urlfile <- "https://raw.githubusercontent.com/bbolker/stats720/main/data/olymp1.csv"

data<-read.csv(url(urlfile))

head(data)
```

```
         team year  medal n      gdp      pop
1 Afghanistan 2000 Bronze 0 6.206548 19.54298
2 Afghanistan 2000   Gold 0 6.206548 19.54298
3 Afghanistan 2000 Silver 0 6.206548 19.54298
4 Afghanistan 2004 Bronze 0 7.978516 23.55355
5 Afghanistan 2004   Gold 0 7.978516 23.55355
6 Afghanistan 2004 Silver 0 7.978516 23.55355
```

```r
str(data)
```

```
'data.frame':	2435 obs. of  6 variables:
 $ team : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
 $ year : int  2000 2000 2000 2004 2004 2004 2008 2008 2008 2012 ...
 $ medal: chr  "Bronze" "Gold" "Silver" "Bronze" ...
 $ n    : int  0 0 0 0 0 0 1 0 0 1 ...
 $ gdp  : num  6.21 6.21 6.21 7.98 7.98 ...
 $ pop  : num  19.5 19.5 19.5 23.6 23.6 ...
```

```r
#View(data)
```

a. State which possible predictor variables you're going to include; justify your choice (refer to Harrell chapter 4 for rules of thumb about appropriate numbers of predictors).

```r
library(tidyverse)

wt_data<-data |>
    mutate(across(medal, ~ factor(., levels = c("Bronze", "Silver", "Gold")))) |>
    group_by(team, year) |>
    arrange(medal) |>
    summarise(n_wt = sum(c(1,2,4)*n)/4,
              gdp = mean(gdp),
              pop = mean(pop),
              .groups = "drop")|>
    na.omit()

#View(wt_data)
summary(wt_data)
```

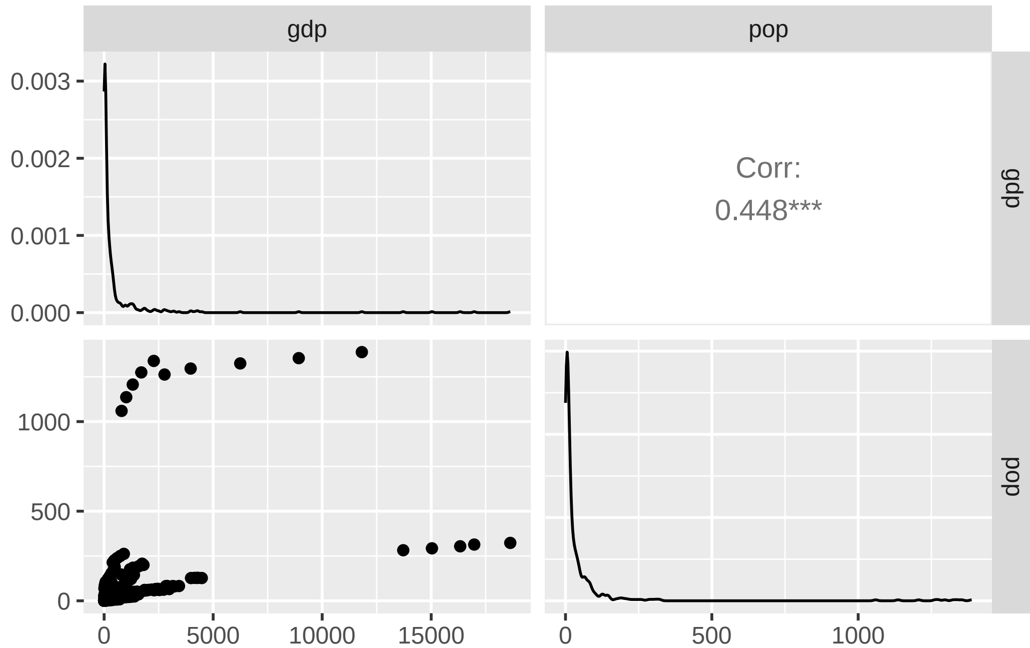
```
     team                year          n_wt             gdp          
 Length:541         Min.   :2000   Min.   :  0.00   Min.   :    0.717  
 Class :character   1st Qu.:2004   1st Qu.:  0.00   1st Qu.:   21.592  
 Mode  :character   Median :2008   Median :  1.50   Median :   87.348  
                    Mean   :2008   Mean   : 10.57   Mean   :  553.276  
                    3rd Qu.:2012   3rd Qu.:  7.50   3rd Qu.:  331.035  
                    Max.   :2016   Max.   :202.00   Max.   :18627.888  
      pop          
 Min.   :   0.1074  
 1st Qu.:   3.9273  
 Median :  10.5148  
 Mean   :  53.2954  
 3rd Qu.:  38.2586  
 Max.   :1387.7900  
```

Descriptive statistics for predictors

```
library(ggplot2)
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from   
  +.gg   ggplot2
```

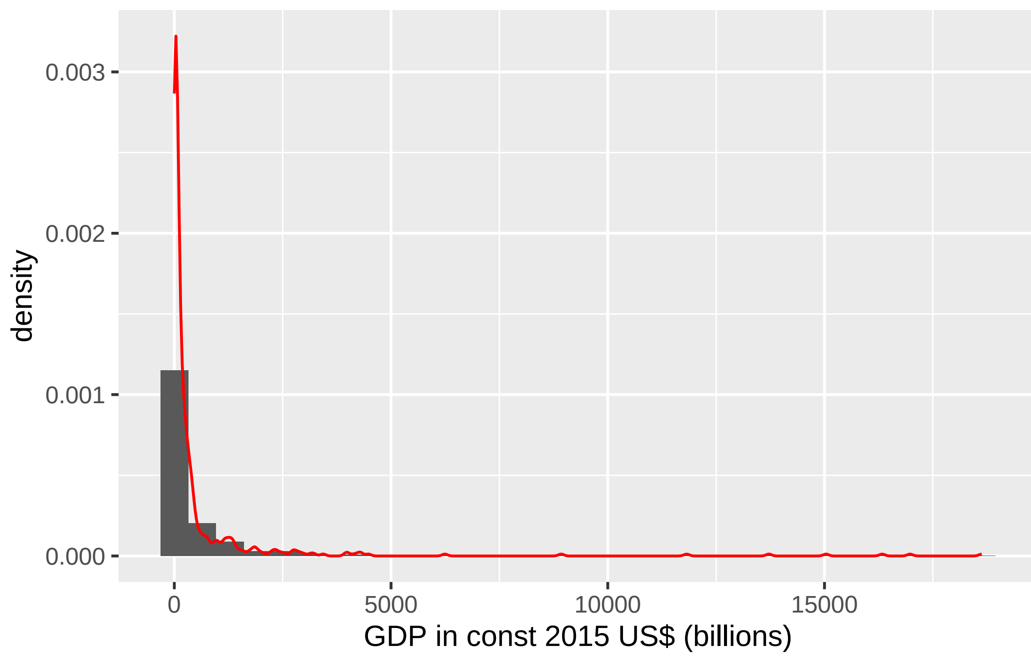
```
# Pairs plot between GDp and population
ggpairs(wt_data[,4:5])
```




```
#Histogram of GDP
ggplot(wt_data, aes(gdp)) +
  geom_histogram(aes(y=..density..)) +  # scale histogram y
  geom_density(col = "red")+
  labs(x = "GDP in const 2015 US$ (billions)")
```

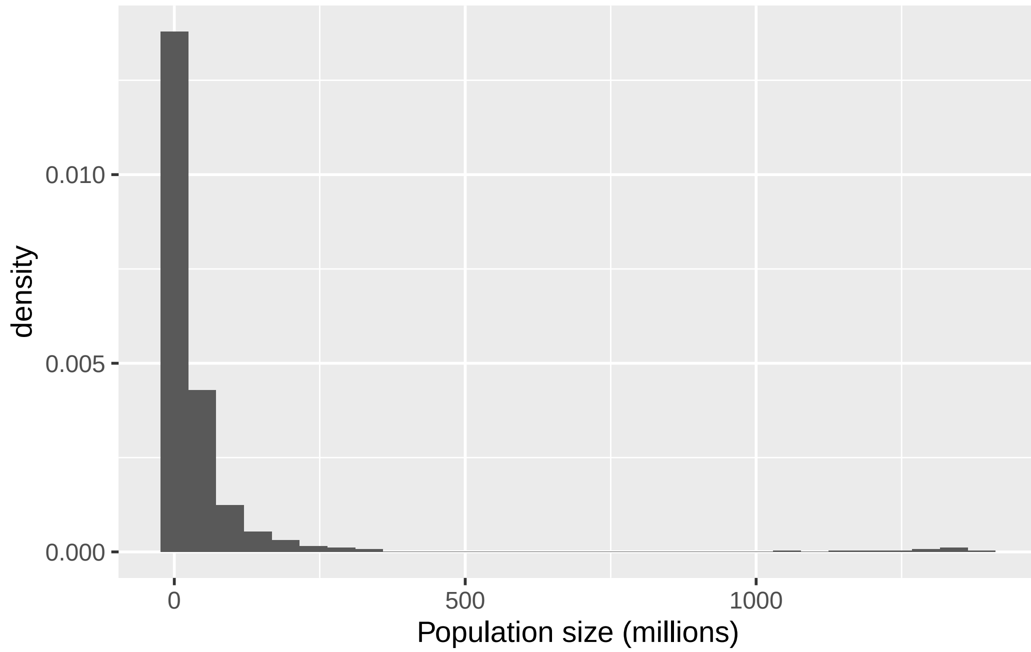
```
Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Histogram of Population
ggplot(wt_data, aes(pop)) +
  geom_histogram(aes(y=..density..)) +  # scale histogram y   geom_density(col = "red")+
  labs(x = "Population size (millions)")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Predictors:

To predict the medals, I am going to use Year, GDP, and population of the country. Natural spline with 5 df is used for GDP and Population. I also expect the interaction between GDP and year also population and Year. Since the GDP and population change over time. Total sample size is 541 which satisfy the Harrell rule of thumb of co-variate and sample size ratio 1:15

Decide whether you're going to predict gold medals only, total medal count, or some weighted average of medals (e.g. `4*G+5*S+2*B`). You can derive these different responses as follows:

Outcome:

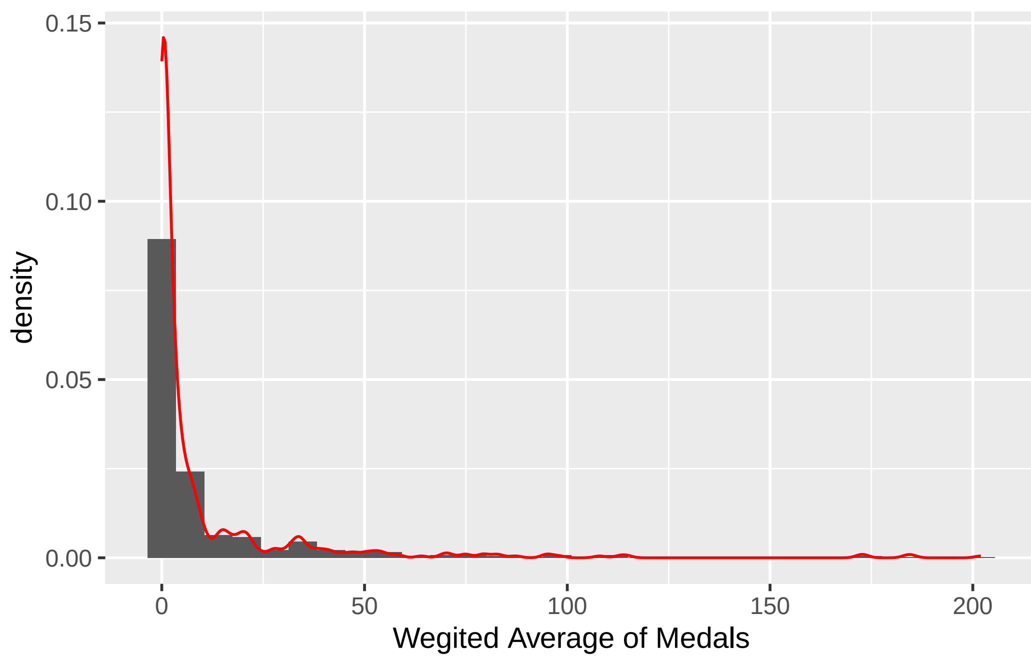
In this analysis, I use weighted average of medals as the outcome, because gold, silver, and bronze medals have different values of importance.

b. State the units of the response variable and of each predictor variable you plan to include; for each variable, state what you would consider as a reasonable threshold for a small change in that variable, *or* for a small slope (regression coefficient)

Histogram of Weighted average of medal

```
ggplot(wt_data, aes(n_wt)) +
  geom_histogram(aes(y=..density..)) +  # scale histogram y
  geom_density(col = "red")+
  labs(x = "Wegited Average of Medals")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

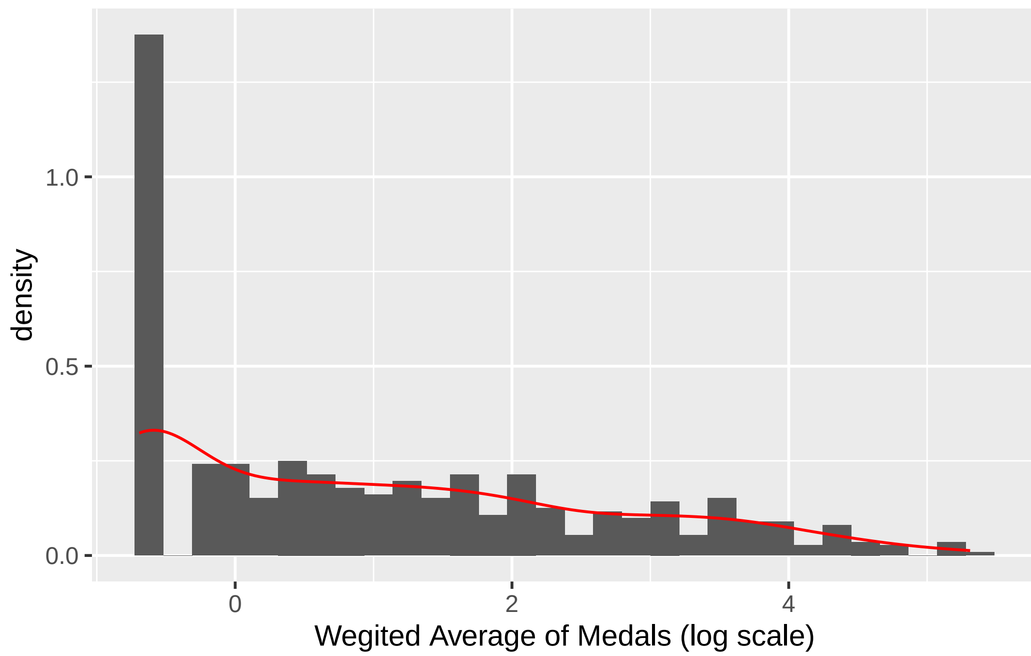


As the outcome is skewed, log transformation is used for in the linear model. And the outcome doesn't have any units.

- GDP is consider US$ (billions) and
- Population size is consider in millions

```r
wt_data$ln.n_wt<-log(wt_data$n_wt+0.5)

ggplot(wt_data, aes(ln.n_wt)) +
  geom_histogram(aes(y=..density..)) +  # scale histogram y
  geom_density(col = "red")+
  labs(x = "Wegited Average of Medals (log scale)")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Descriptive Statistics

```r
cat("Summary of weighted number of medal:","\n")
```

```
Summary of weighted number of medal:
```

```r
summary(wt_data$n_wt)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    0.00    1.50   10.57    7.50  202.00 
```

```r
cat("Summary of weighted number of medal (log scale):","\n")
```

```
Summary of weighted number of medal (log scale): 
```

```r
summary(wt_data$ln.n_wt)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-0.6931 -0.6931  0.6931  1.0076  2.0794  5.3107 
```

```r
cat("Summary of GDP:","\n")
```

```
Summary of GDP: 
```

```r
summary(wt_data$gdp)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
    0.717    21.592    87.348   553.276   331.035 18627.888 
```

```r
cat("Summary of Population:","\n")
```

```
Summary of Population: 
```

```r
summary(wt_data$pop)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
   0.1074    3.9273   10.5148   53.2954   38.2586 1387.7900 
```

c. Fit Model

Linear Model

```r
library(splines)

Mod<-lm(ln.n_wt~year*ns(gdp, df=5)+year*ns(pop, df=5),wt_data)

summary(Mod)
```

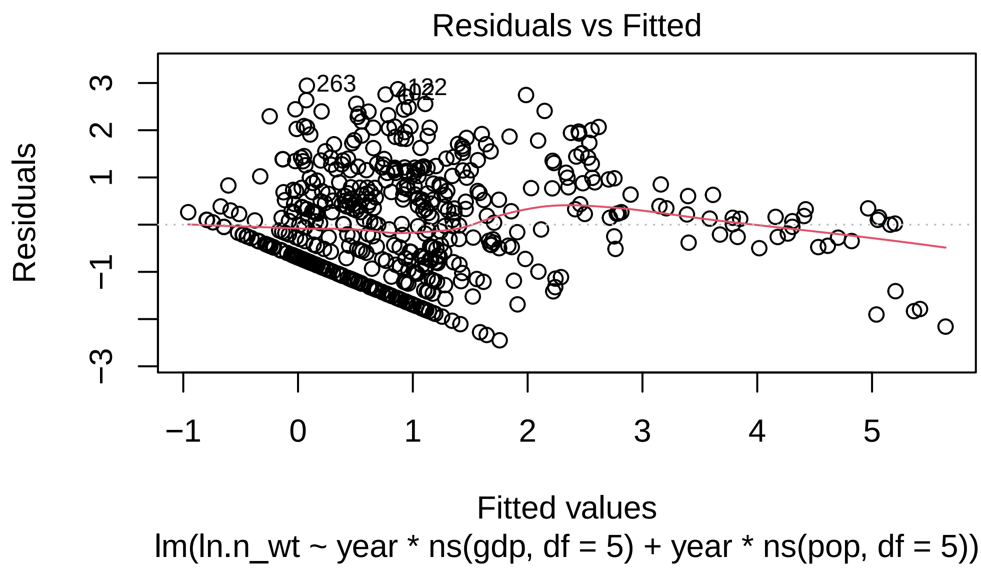
```
Call:
lm(formula = ln.n_wt ~ year * ns(gdp, df = 5) + year * ns(pop, 
    df = 5), data = wt_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4495 -0.8121 -0.1225  0.7688  2.9436 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)  
(Intercept)           -1.545e+02  7.800e+01  -1.981   0.0481 *
year                   7.679e-02  3.886e-02   1.976   0.0487 *
ns(gdp, df = 5)1      -2.725e+01  9.696e+01  -0.281   0.7788  
ns(gdp, df = 5)2       1.583e+02  9.045e+01   1.750   0.0807 .
ns(gdp, df = 5)3      -1.019e+02  2.665e+02  -0.382   0.7023  
ns(gdp, df = 5)4       9.327e+01  2.332e+02   0.400   0.6894  
ns(gdp, df = 5)5      -3.071e+02  2.930e+02  -1.048   0.2950  
ns(pop, df = 5)1       1.659e+02  9.069e+01   1.829   0.0680 .
ns(pop, df = 5)2       7.663e+01  9.903e+01   0.774   0.4394  
ns(pop, df = 5)3       1.473e+01  3.250e+02   0.045   0.9639  
ns(pop, df = 5)4       1.993e+02  2.501e+02   0.797   0.4258  
ns(pop, df = 5)5       3.476e+02  1.860e+02   1.869   0.0622 .
year:ns(gdp, df = 5)1  1.405e-02  4.830e-02   0.291   0.7713  
year:ns(gdp, df = 5)2 -7.808e-02  4.506e-02  -1.733   0.0837 .
year:ns(gdp, df = 5)3  5.583e-02  1.327e-01   0.421   0.6741  
year:ns(gdp, df = 5)4 -4.243e-02  1.161e-01  -0.365   0.7151  
year:ns(gdp, df = 5)5  1.552e-01  1.457e-01   1.065   0.2875  
year:ns(pop, df = 5)1 -8.246e-02  4.518e-02  -1.825   0.0685 .
year:ns(pop, df = 5)2 -3.812e-02  4.933e-02  -0.773   0.4400  
year:ns(pop, df = 5)3 -7.749e-03  1.618e-01  -0.048   0.9618  
year:ns(pop, df = 5)4 -1.000e-01  1.246e-01  -0.803   0.4224  
year:ns(pop, df = 5)5 -1.742e-01  9.259e-02  -1.882   0.0604 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

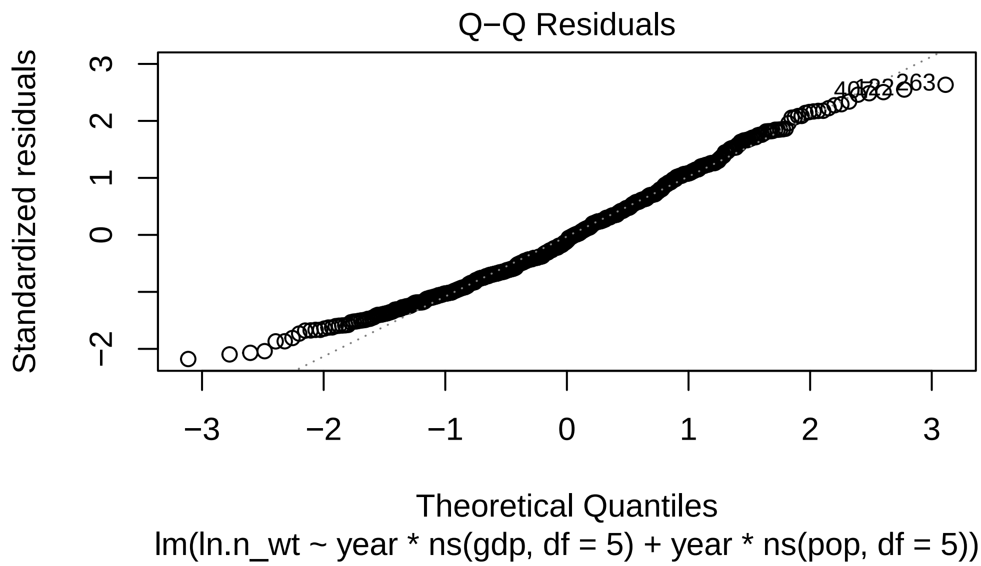
```
Residual standard error: 1.136 on 519 degrees of freedom
Multiple R-squared:  0.5142,	Adjusted R-squared:  0.4945 
F-statistic: 26.16 on 21 and 519 DF,  p-value: < 2.2e-16
```

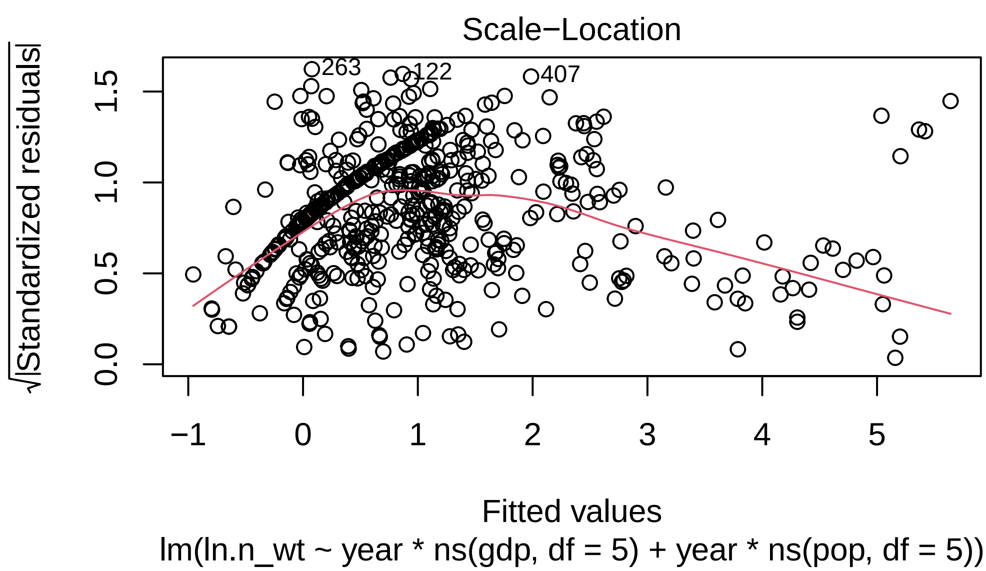
d. Diagnose the model

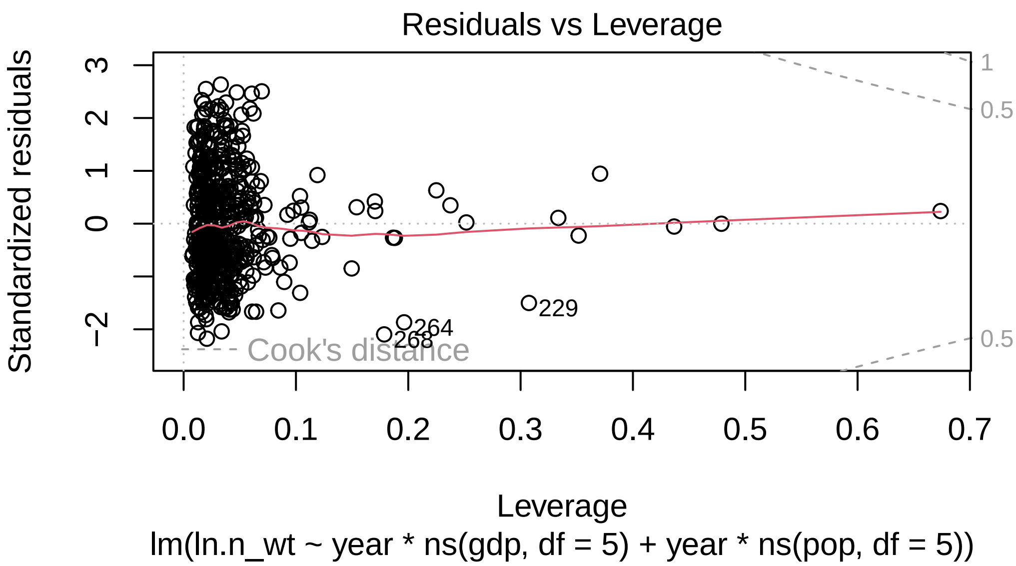
Performance plot

```r
#par(mfrow=c(2,2))
plot(Mod)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(ln.n_wt ~ year * ns(gdp, df = 5) + year * ns(pop, df = 5))
263
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
263
122
407
0.0
0.5
1.0
1.5
−1
0
1
2
3
4
5
Fitted values
lm(ln.n_wt ~ year * ns(gdp, df = 5) + year * ns(pop, df = 5))



Interpretation:

Linearity Assumption:

As there is no clear pattern in the scatter plot of fitted values vs Residual. Therefore linearity assumption is satisfied.

Normality:

From the Q-Q plot, we can see that the points at the lower tail are slightly deviates from the diagonal line. However, in my opinion the normality assumption is satisfied approximately.

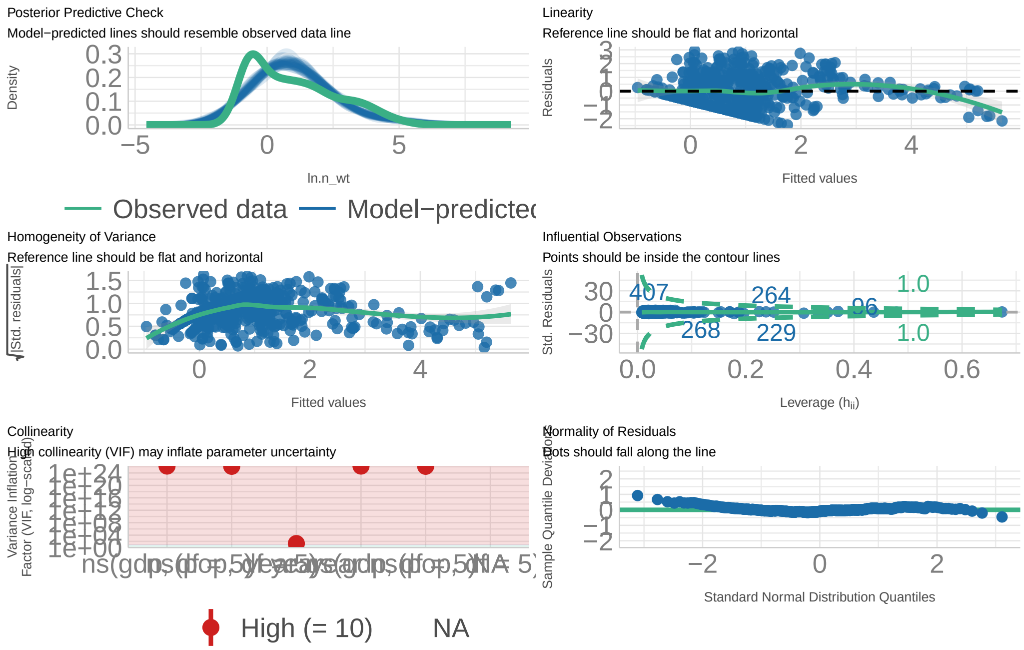
Homoscedasticity:

Scale-Location plot is dense upto 3 and then the spread is wider. Also, the red line is not horizontal until 3. It suggests that the homoscedasticity assumption is violated slightly.

Residual vs Leverage:

The plot depicts that there is no influential cases since all the points are inside the boundary.

```
library(performance)

check_model(Mod,panel=T,check='all',title_size=5,base_size=5,axis_title_size=5)
```



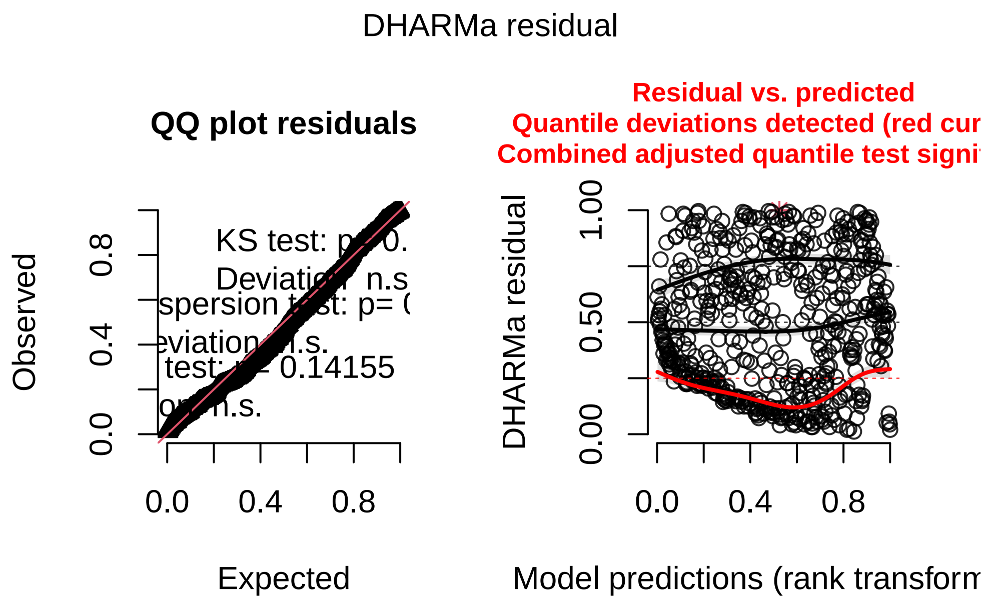
Performance plot is also says the same story of linearity, homogeneity of variance, normality of residuals, and influential points. In addition to that, The posterior predictive check plot illustrate observed and predicted density curve from the model which means, there is a slight deviation in the curve when it reaches the peak. VIF plot illustrates that there is a high collinearity among the predictors. There I drop both interaction term from the model.

DHARMa

```r
library(DHARMa)
```

```
This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'D
```

```r
simulationOutput <- simulateResiduals(fittedModel = Mod)

plot(simulationOutput)
```



The Q-Q residual plot suggests that the model satisfies the normality of residual, homogeneity assumption and there is no influential points.

The residual vs predicted plot suggests that more residuals are in the lower tail of the distribution then we expect.

e. If the model has any problems, make adjustments

To correct multi-colinearity, I drop the interaction term. In order to compare the most important predictor, I would scale the predictors and make unitless.

Scale

```
wt_data$s.gdp<-scale(wt_data$gdp,center = FALSE,scale = TRUE)

wt_data$s.pop<-scale(wt_data$pop,center = FALSE,scale = TRUE)
```

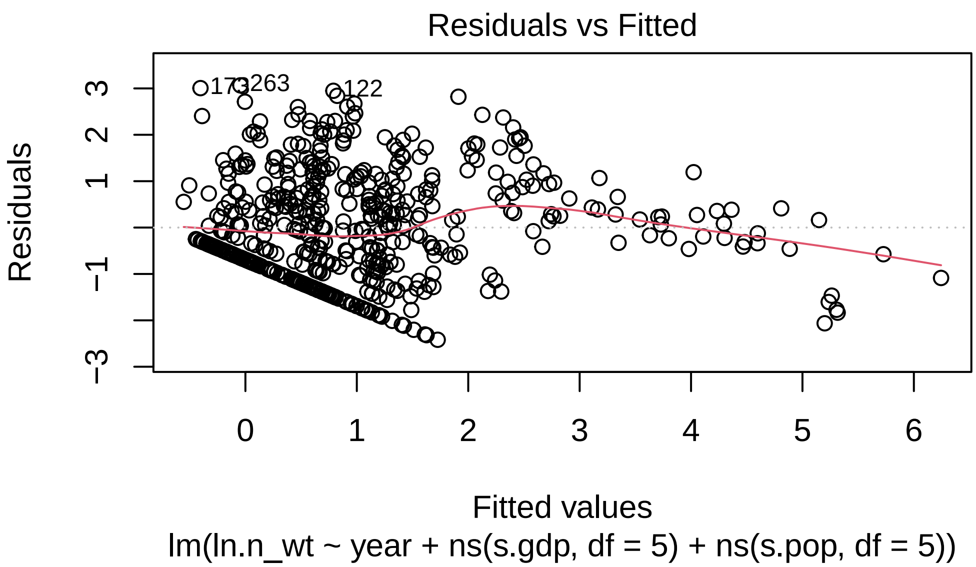
Fitting the Model

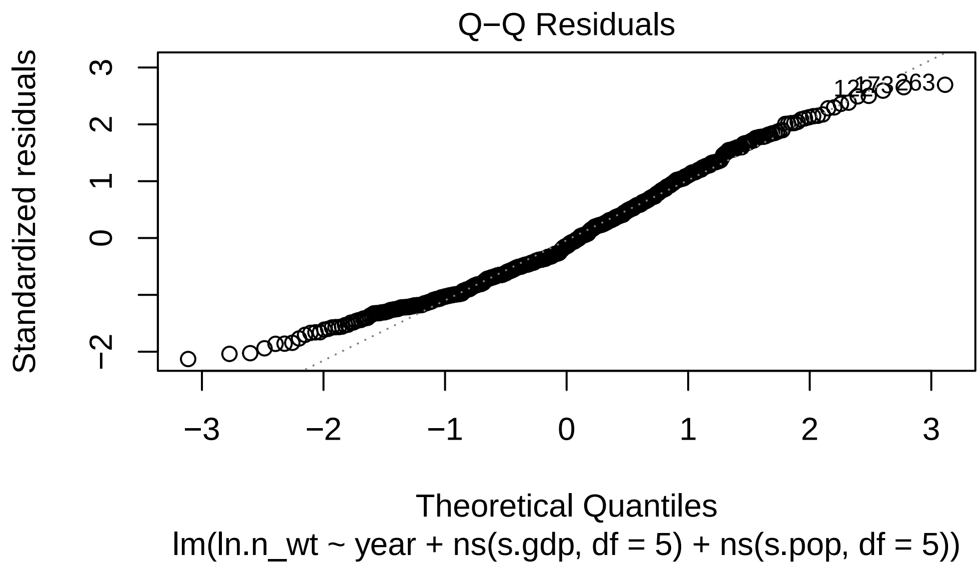
```r
Mod.Scaled<-lm(ln.n_wt~year+ns(s.gdp, df=5)+ns(s.pop, df=5),wt_data)

summary(Mod.Scaled)
```

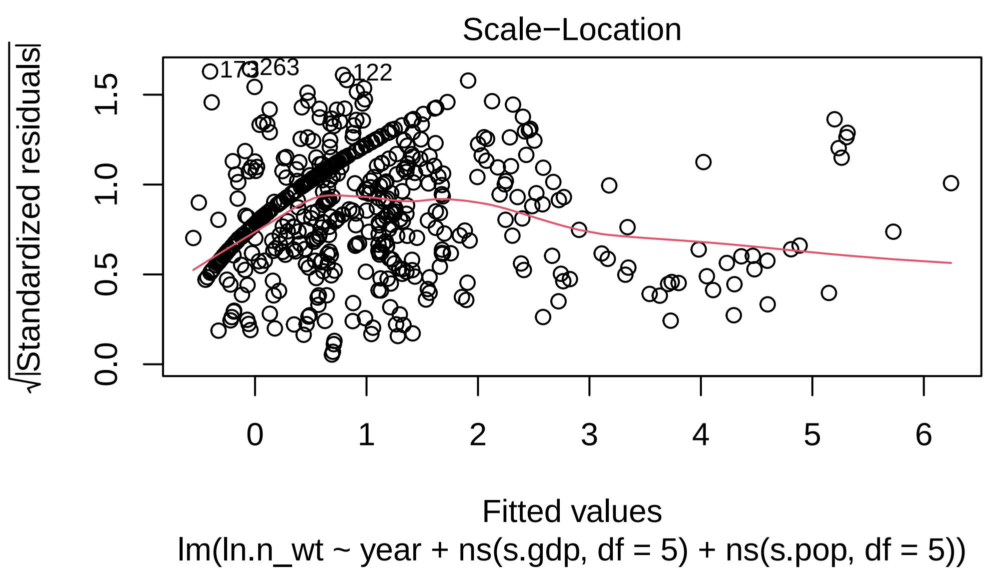
```
Call:
lm(formula = ln.n_wt ~ year + ns(s.gdp, df = 5) + ns(s.pop, df = 5),
    data = wt_data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4189 -0.8536 -0.1652  0.7673  3.0655

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        28.130200  17.625388   1.596  0.11108
year               -0.014234   0.008781  -1.621  0.10559
ns(s.gdp, df = 5)1  1.007928   0.270050   3.732  0.00021 ***
ns(s.gdp, df = 5)2  1.585695   0.250029   6.342 4.87e-10 ***
ns(s.gdp, df = 5)3  9.952518   0.730633  13.622  < 2e-16 ***
ns(s.gdp, df = 5)4  8.312507   0.641493  12.958  < 2e-16 ***
ns(s.gdp, df = 5)5  4.603230   0.764216   6.023 3.20e-09 ***
ns(s.pop, df = 5)1  0.324111   0.254214   1.275  0.20288
ns(s.pop, df = 5)2  0.064812   0.277035   0.234  0.81512
ns(s.pop, df = 5)3 -0.558101   0.913552  -0.611  0.54152
ns(s.pop, df = 5)4 -1.391665   0.705981  -1.971  0.04922 *
ns(s.pop, df = 5)5 -2.191894   0.508931  -4.307 1.97e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.145 on 529 degrees of freedom
Multiple R-squared:  0.4972,    Adjusted R-squared:  0.4868
F-statistic: 47.56 on 11 and 529 DF,  p-value: < 2.2e-16
```

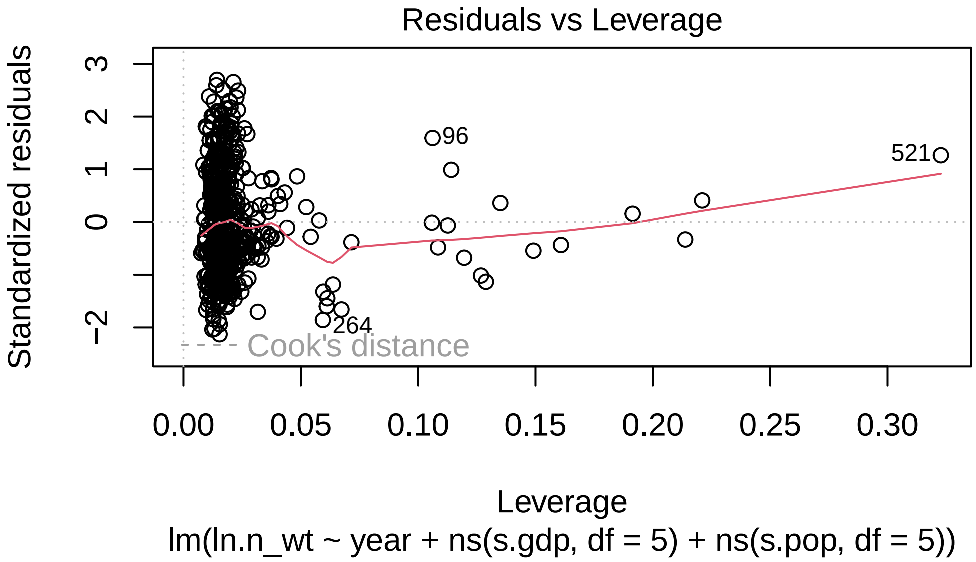
Plots

```r
plot(Mod.Scaled)
```



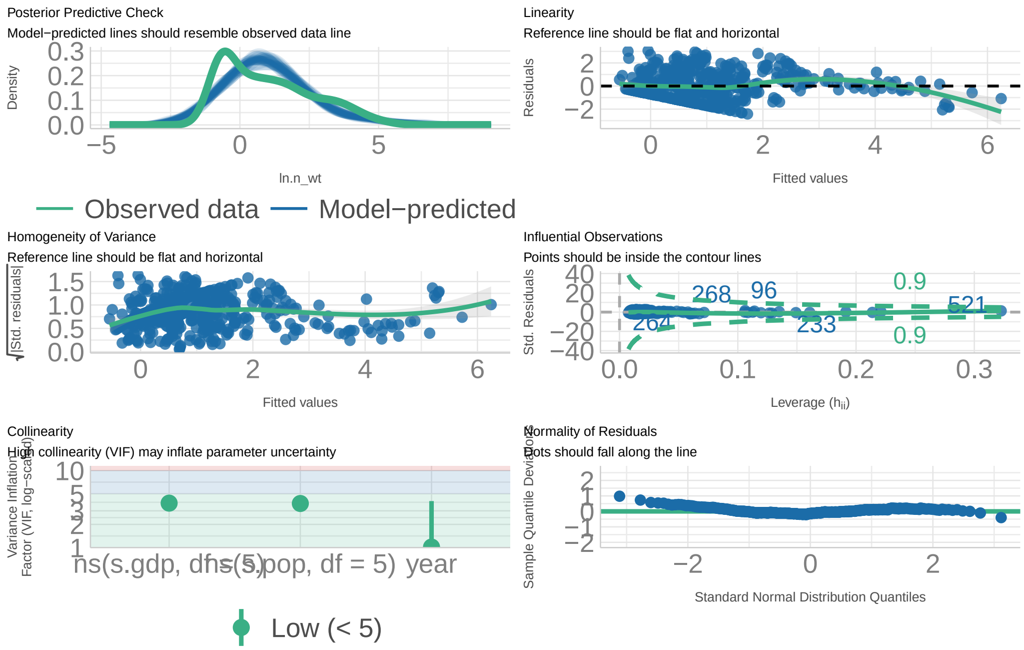

Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(ln.n_wt ~ year + ns(s.gdp, df = 5) + ns(s.pop, df = 5))


Scale−Location
√|Standardized residuals|
Fitted values
lm(ln.n_wt ~ year + ns(s.gdp, df = 5) + ns(s.pop, df = 5))




```r
check_model(Mod.Scaled,panel=T,check='all',title_size=5,base_size=5,axis_title_size=5)
```

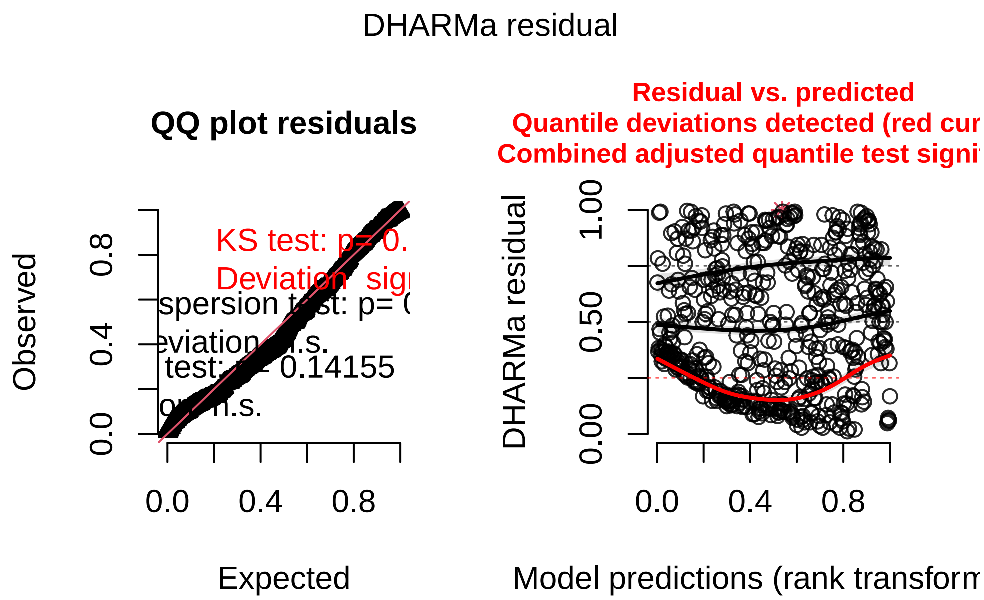
```
Some of the variables were in matrix-format - probably you used
  `scale()` on your data?
  If so, and you get an error, please try `datawizard::standardize()` to
  standardize your data.
Some of the variables were in matrix-format - probably you used
  `scale()` on your data?
  If so, and you get an error, please try `datawizard::standardize()` to
  standardize your data.
```



From the VIF plot, we can see that there is no collinear problem among predictors.

DHARMa

```r
simulationOutput.scaled <- simulateResiduals(fittedModel = Mod.Scaled)

plot(simulationOutput.scaled)
```

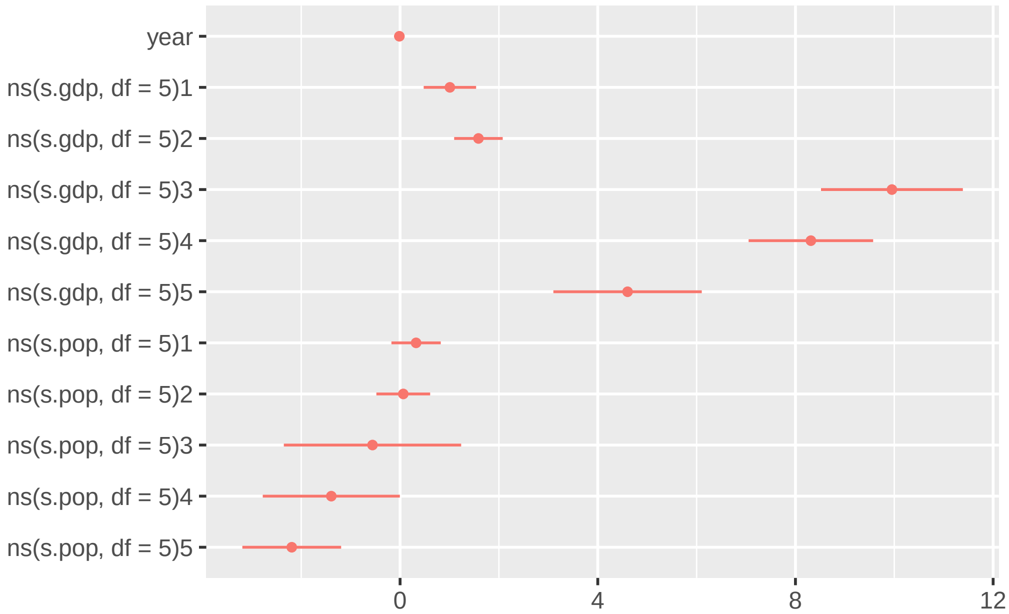


The Q-Q residual plot suggests that the model satisfies the homogeneity assumption and there is no influential points. The residual vs predicted plot suggests that more residuals are in the lower tail of the distribution then we expect.

Since the sample size is large it can detect a small deviation of the normality. KS test is sensitive to sample size.

f. Show a coefficient plot of the results

```
library(dotwhisker)

dwplot(Mod.Scaled)
```



From the plot we can that, GDP has non-linear effect on average number of medals. One year increase, reduces the average number of medals by 1% (exp(-0.01)). As GDP and population are included as spline term, the regression coefficients can't be interpreted directly.

g. Show an effects plot (predicted values or effects

```r
library(effects)
```

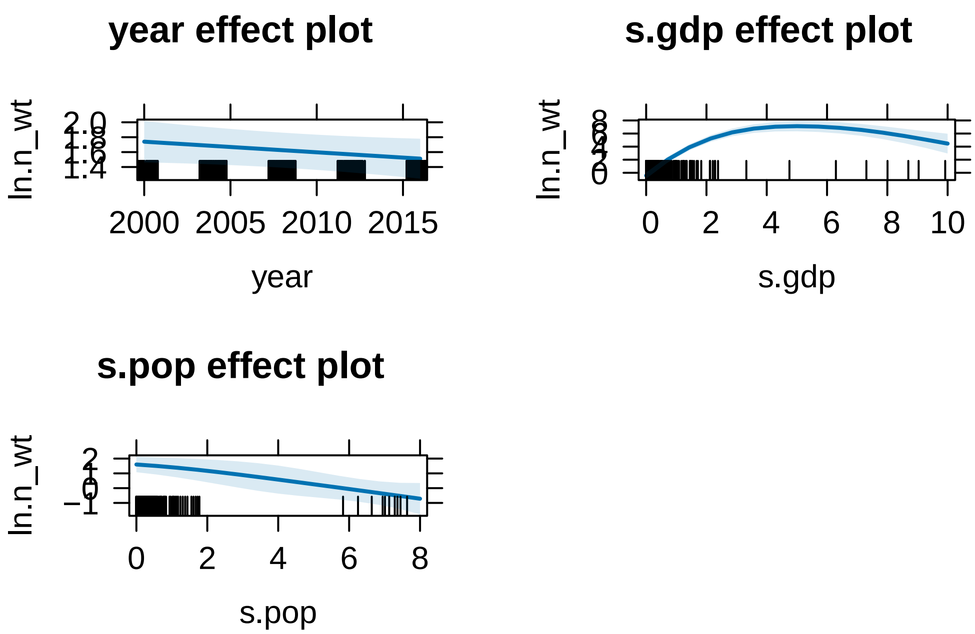
```
Loading required package: carData
```

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

```r
#effects::allEffects(Mod)

effect.plot<-effects::allEffects(Mod.Scaled)
```

```
Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
predictors s.gdp, s.pop are one-column matrices that were converted to vectors
Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
predictors s.gdp, s.pop are one-column matrices that were converted to vectors
Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
predictors s.gdp, s.pop are one-column matrices that were converted to vectors
```

```r
plot(effect.plot)
```



The marginal effect plots show that the year and population has negative effect on average medals, wehereas GDP has quadratic effect.

Estimate Marginal means

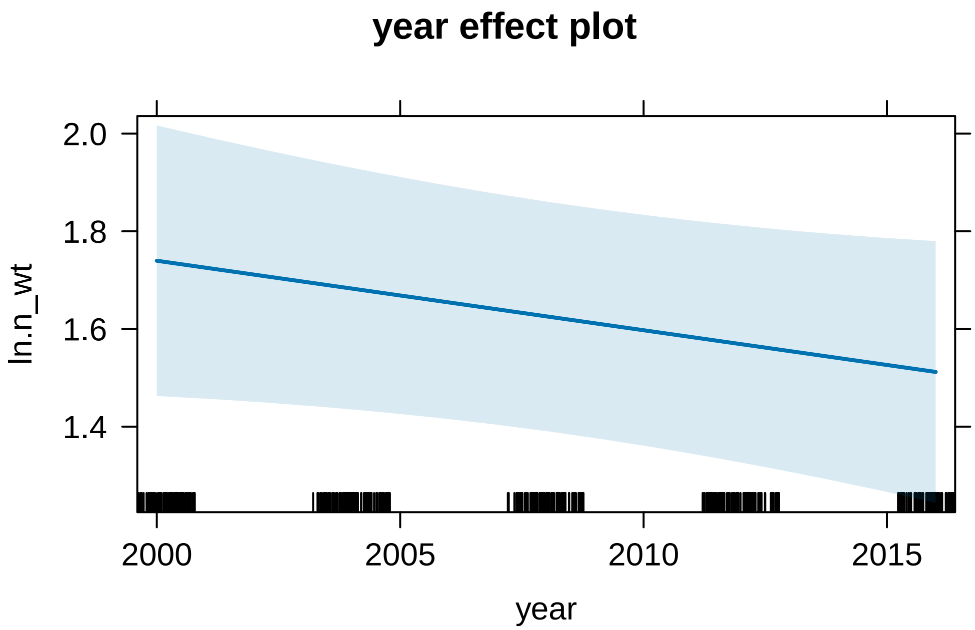
```r
library(emmeans)
```

```
Welcome to emmeans.
Caution: You lose important information if you filter this package's results.
See '? untidy'
```

```
Attaching package: 'emmeans'


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

    pigs
```

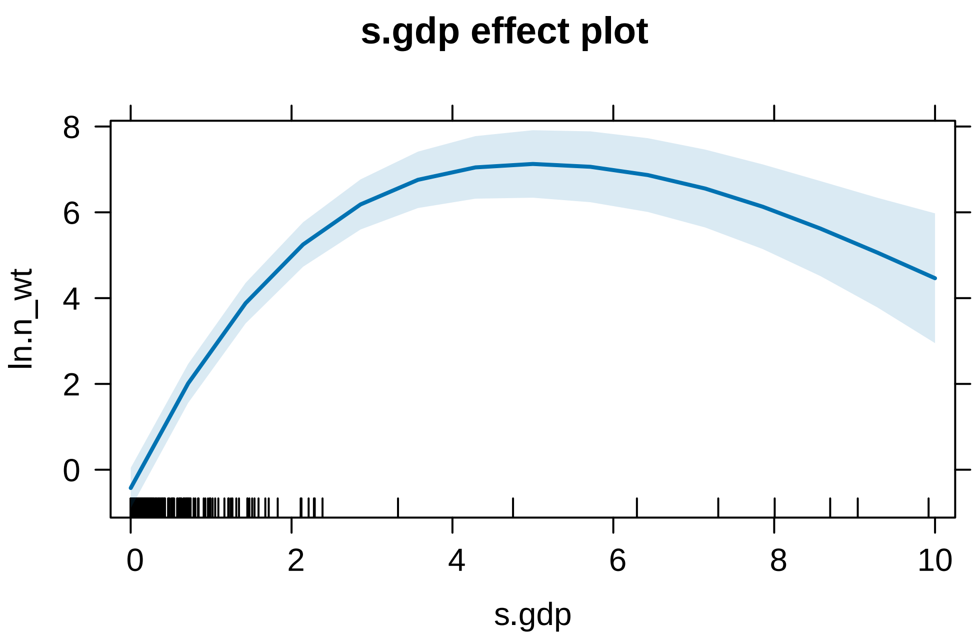
```
plot(effect("year",Mod.Scaled))
```

```
Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
predictors s.gdp, s.pop are one-column matrices that were converted to vectors
```




```
plot(effect("s.gdp",Mod.Scaled))
```

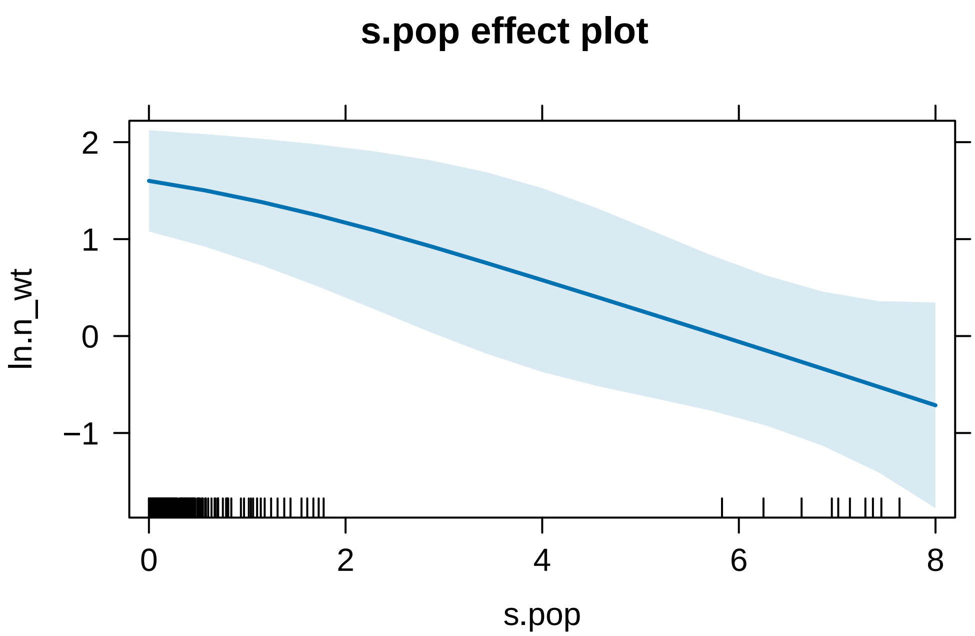
```
NOTE: s.gdp does not appear in the model


Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
predictors s.gdp, s.pop are one-column matrices that were converted to vectors
```




```
plot(effect("s.pop",Mod.Scaled))
```

```
NOTE: s.pop does not appear in the model
```

```
Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
predictors s.gdp, s.pop are one-column matrices that were converted to vectors
```



Test for non-linearity

GDP

$$H_0 : \beta_3 = \beta_4 = \beta_5 = \beta_6$$

```
library(car)
```

```
Attaching package: 'car'

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

    recode

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

    some
```

```
linearHypothesis(Mod, names(coef(Mod))[4:7])
```

```
Linear hypothesis test

Hypothesis:
ns(gdp, df = 5)2 = 0
ns(gdp, df = 5)3 = 0
ns(gdp, df = 5)4 = 0
ns(gdp, df = 5)5 = 0

Model 1: restricted model
Model 2: ln.n_wt ~ year * ns(gdp, df = 5) + year * ns(pop, df = 5)

  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    523 679.33
2    519 670.13  4    9.1992 1.7811 0.1312
```

ANOVA/F test suggests that the full model is has significant lower RSS, means natural spline of GDP has non-linear effect on log transformed weighted average medal.

Population

$$H_0 : \beta_9 = \beta_{10} = \beta_{11} = \beta_{12}$$

```r
linearHypothesis(Mod, names(coef(Mod))[9:12])
```

```
Linear hypothesis test

Hypothesis:
ns(pop, df = 5)2 = 0
ns(pop, df = 5)3 = 0
ns(pop, df = 5)4 = 0
ns(pop, df = 5)5 = 0

Model 1: restricted model
Model 2: ln.n_wt ~ year * ns(gdp, df = 5) + year * ns(pop, df = 5)

  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    523 675.19
2    519 670.13  4    5.0537 0.9785 0.4188
```

ANOVA/F test suggests that the full model is has significant lower RSS, means natural spline of population has non-linear effect on log transformed weighted average medal.

Question 2: contrasts

Suppose we have an experiment with four levels: control (C) and three increasing levels of the treatment (I, II, III). We are interested in:

- the difference between the control and the *average* of the treatment levels
- *successive differences* (I vs II, II vs III) among the non-control treatments.

Construct a set of contrasts to quantify these effects. Test your results by making up a minimal data frame with just one observation per treatment. Fit the linear model and show that the coefficients match what you intended

```
trt<-c("control","I","II","III")
mu<-c(0.5,0.9,1.2,0.7)

set.seed(100)
y<-rnorm(length(trt),mu,0.5)

da<-data.frame(y=y,x=trt)
da
```

```
          y       x
1 0.2489038 control
2 0.9657656       I
3 1.1605415      II
4 1.1433924     III
```

```
mod<-lm(y~as.factor(x),da)
mod
```

```

Call:
lm(formula = y ~ as.factor(x), data = da)

Coefficients:
    (Intercept)    as.factor(x)I   as.factor(x)II  as.factor(x)III
         0.2489           0.7169           0.9116           0.8945
```

```r
Mean.eff<-predict(mod,da)

# 3 contrasts:
# 1. the difference between the control and the average of the treatment levels
# 2. difference of I vs II
# 3. difference II vs III among the non-control treatments



C.inv<-matrix(c(1,-1/3,-1/3,-1/3,
                0,1,-1,0,
                0,0,1,-1),
                byrow = T,nrow = 3)
C.inv
```

```
     [,1]       [,2]       [,3]       [,4]
[1,]    1 -0.3333333 -0.3333333 -0.3333333
[2,]    0  1.0000000 -1.0000000  0.0000000
[3,]    0  0.0000000  1.0000000 -1.0000000
```

```r
C.inv%*%Mean.eff
```

```
            [,1]
[1,] -0.84099599
[2,] -0.19477587
[3,]  0.01714905
```

```r
# C matrix with inercept

C<-matrix(c(1,1,1,1,
               1,-1/3,-1/3,-1/3,
                0,1,-1,0,
                0,0,1,-1),
                byrow = F,nrow = 4)

cat("C matrix","\n")
```

```
C matrix
```

```r
C
```

```
     [,1]       [,2] [,3] [,4]
[1,]    1  1.0000000    0    0
[2,]    1 -0.3333333    1    0
[3,]    1 -0.3333333   -1    1
[4,]    1 -0.3333333    0   -1
```

```r
library(MASS)
```

```

Attaching package: 'MASS'

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

    select
```

```r
# C-inverse matrix

cat("C-inverse matrix","\n")
```

```
C-inverse matrix
```

```r
Contrast.mat<-fractions(solve(C))
Contrast.mat[,-1]
```

```
     [,1] [,2] [,3]
[1,]  1/4  1/4  1/4
[2,] -1/4 -1/4 -1/4
[3,]  2/3 -1/3 -1/3
[4,]  1/3  1/3 -2/3
```

Question 3: simulations to evaluate the effects of model misspecification

Function to Simulate data from t-distribution

```r
sim_fun <- function(n = 100, slope = 1, sd = 1, intercept = 0,df=2) {
    x <- runif(n)
    mu<-intercept + slope * x
    #y <- rnorm(n, intercept + slope * x, sd = sd)
    y<-mu+sd*rt(n, df)
    data.frame(x, y)
}

#sim<-sim_fun (n = 100, slope = 1, sd = 1, intercept = 0)
```

Evaluate coverage for model mis-specification

```r
Model_misp<-function(n=100,t.df=2,slope=1,sd=1,intercept=0,B=1000,alpha=0.05)
  {

  out<-data.frame(matrix(0,nrow=B,ncol = 3))
  colnames(out)<-c("slope","p.val","Coverage")


  for(i in 1:B)
  {
    #cat("Iteration number is:",i,"\n")

  sim.dat<-sim_fun (n = n, slope = slope, sd = sd, intercept = intercept,df=t.df)

  head(sim.dat)

  lm.mod<-lm(y~x,sim.dat)
  out$slope[i] <- coef(lm.mod)[2]
  out$p.val[i]<-coef(summary(lm.mod))[2, "Pr(>|t|)"]

  between <- function(a, b) (b[1] < a & a < b[2]);
  out$Coverage[i]<- between(slope, confint(lm.mod)[2,])
  }

  Bias<-mean(out$slope-slope)
  SE<-sd(out$slope)
  RMSE<-sqrt(mean((out$slope-slope)^2))

  Power<-mean(out$p.val<alpha)
```

```
  Coverage<-mean(out$Coverage)

  #Metrics<-list(Bias=Bias,SE=SE,RMSE=RMSE,Power=Power,Coverage=Coverage)

  return(results=list(Bias=Bias,SE=SE,RMSE=RMSE,Power=Power,Coverage=Coverage,Estimates=ou

}

res<-Model_misp(n=100,t.df=2,slope=1,sd=1,intercept=0,B=1000,alpha=0.05)

n<-seq(10,100,by=10)
df<-seq(2, 50, by = 6)

results<-expand.grid(n=n,df=df)


for(i in 1:nrow(results)){

     set.seed(1000+i)
     res<-Model_misp(n=results$n[i],t.df=results$df[i],slope=1,sd=1,intercept=0,B=1000,alp

results$Bias[i]<-res$Bias
results$SE[i]<-res$SE
results$RMSE[i]<-res$RMSE
results$Power[i]<-res$Power
results$Coverage[i]<-res$Coverage

}

results
```

```
     n df          Bias        SE      RMSE Power Coverage
1   10  2  0.0218595214 3.5895902 3.5878615 0.076    0.937
2   20  2 -0.0073709893 2.6444793 2.6431670 0.105    0.958
3   30  2 -0.0778527608 2.0815568 2.0819719 0.116    0.958
4   40  2 -0.1017981934 1.8502557 1.8521300 0.147    0.940
5   50  2 -0.0347609093 1.4947178 1.4943746 0.145    0.954
6   60  2  0.0968219980 1.8208437 1.8225067 0.192    0.956
7   70  2  0.0024580897 1.1240820 1.1235225 0.199    0.961
8   80  2  0.0237283434 1.4402335 1.4397088 0.223    0.956
9   90  2 -0.0002820687 1.2096921 1.2090871 0.223    0.953
10 100  2 -0.0126257102 1.1320592 1.1315635 0.250    0.951
```

11	10	8	-0.0515679820	1.4259321	1.4261515	0.110	0.955
12	20	8	-0.0067655281	0.9552187	0.9547649	0.196	0.942
13	30	8	-0.0151260646	0.7546581	0.7544323	0.254	0.949
14	40	8	0.0125071126	0.6630010	0.6627874	0.349	0.944
15	50	8	0.0068930966	0.5953288	0.5950710	0.408	0.940
16	60	8	0.0204083386	0.5302600	0.5303876	0.495	0.952
17	70	8	0.0224191729	0.4769822	0.4772704	0.550	0.953
18	80	8	0.0122020768	0.4629180	0.4628474	0.616	0.944
19	90	8	0.0157803038	0.4466383	0.4466937	0.655	0.946
20	100	8	-0.0042286131	0.3908542	0.3906816	0.684	0.959
21	10	14	-0.0518997522	1.2932170	1.2936118	0.097	0.951
22	20	14	-0.0058549002	0.8709583	0.8705424	0.198	0.944
23	30	14	0.0060676230	0.7146047	0.7142731	0.270	0.949
24	40	14	-0.0186884651	0.6068555	0.6068398	0.362	0.950
25	50	14	0.0093959287	0.5514083	0.5512126	0.447	0.952
26	60	14	0.0133625731	0.4939028	0.4938366	0.522	0.950
27	70	14	-0.0071733354	0.4473936	0.4472274	0.588	0.953
28	80	14	-0.0104992466	0.4275915	0.4275066	0.618	0.949
29	90	14	-0.0156276852	0.4031473	0.4032486	0.695	0.941
30	100	14	-0.0021582989	0.3623533	0.3621785	0.780	0.947
31	10	20	-0.0568342173	1.3381028	1.3386407	0.117	0.935
32	20	20	-0.0279860007	0.8242102	0.8242732	0.199	0.961
33	30	20	-0.0122992554	0.6808521	0.6806227	0.297	0.952
34	40	20	-0.0091135232	0.5890949	0.5888708	0.399	0.961
35	50	20	0.0161912074	0.5185226	0.5185161	0.477	0.960
36	60	20	-0.0095241911	0.4772801	0.4771365	0.535	0.955
37	70	20	-0.0342428118	0.4665877	0.4676098	0.591	0.939
38	80	20	0.0037466861	0.4134585	0.4132687	0.668	0.952
39	90	20	-0.0132804259	0.4013590	0.4013781	0.694	0.944
40	100	20	-0.0021710612	0.3669792	0.3668021	0.770	0.950
41	10	26	-0.0187496641	1.2642145	1.2637213	0.128	0.949
42	20	26	0.0124275774	0.8517176	0.8513823	0.203	0.939
43	30	26	0.0006904761	0.6807617	0.6804216	0.320	0.945
44	40	26	-0.0081542237	0.5873802	0.5871431	0.406	0.946
45	50	26	0.0192417729	0.5073518	0.5074630	0.491	0.965
46	60	26	-0.0064943529	0.4851123	0.4849132	0.562	0.934
47	70	26	-0.0213907719	0.4534566	0.4537343	0.606	0.949
48	80	26	-0.0158201673	0.3939339	0.3940546	0.681	0.959
49	90	26	0.0073036795	0.3824970	0.3823754	0.735	0.956
50	100	26	-0.0104583120	0.3568978	0.3568725	0.778	0.954
51	10	32	-0.0241962664	1.2978057	1.2973823	0.118	0.937
52	20	32	-0.0031347924	0.8392026	0.8387887	0.209	0.946
53	30	32	0.0071653371	0.6923543	0.6920451	0.325	0.946

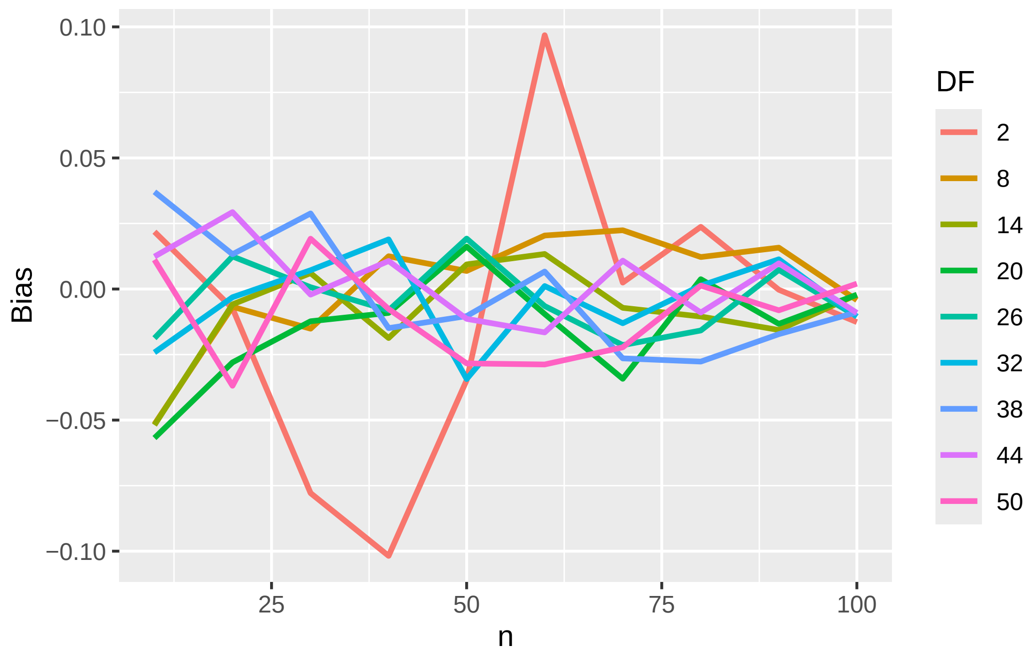
```
54  40 32  0.0189375934 0.5769600 0.5769823 0.413    0.953
55  50 32 -0.0343354944 0.5160919 0.5169753 0.462    0.946
56  60 32  0.0011451290 0.4690567 0.4688235 0.564    0.953
57  70 32 -0.0130525207 0.4376707 0.4376465 0.622    0.946
58  80 32  0.0012771787 0.4012028 0.4010042 0.695    0.950
59  90 32  0.0113366553 0.3866452 0.3866180 0.739    0.936
60 100 32 -0.0105588159 0.3662593 0.3662284 0.766    0.949
61  10 38  0.0370772057 1.2700155 1.2699217 0.128    0.935
62  20 38  0.0132638985 0.8245994 0.8242937 0.221    0.954
63  30 38  0.0288349296 0.6487288 0.6490452 0.306    0.954
64  40 38 -0.0149106178 0.5799881 0.5798898 0.404    0.954
65  50 38 -0.0103223888 0.4728997 0.4727758 0.493    0.967
66  60 38  0.0066717373 0.4693540 0.4691667 0.559    0.945
67  70 38 -0.0264588717 0.4250136 0.4256242 0.604    0.958
68  80 38 -0.0276799035 0.4077135 0.4084486 0.661    0.955
69  90 38 -0.0172105351 0.3859669 0.3861576 0.736    0.953
70 100 38 -0.0088593634 0.3452754 0.3452164 0.787    0.949
71  10 44  0.0123797648 1.2827362 1.2821544 0.123    0.946
72  20 44  0.0293393573 0.8272464 0.8273531 0.222    0.951
73  30 44 -0.0021412368 0.6462148 0.6458952 0.310    0.964
74  40 44  0.0107049715 0.5673222 0.5671395 0.422    0.952
75  50 44 -0.0113795459 0.5254795 0.5253399 0.487    0.943
76  60 44 -0.0165485090 0.4600446 0.4601123 0.553    0.956
77  70 44  0.0108335607 0.4288320 0.4287544 0.652    0.953
78  80 44 -0.0089833849 0.4035550 0.4034532 0.705    0.944
79  90 44  0.0097965436 0.3738644 0.3738058 0.750    0.949
80 100 44 -0.0090846682 0.3606650 0.3605991 0.786    0.941
81  10 50  0.0111526674 1.1860166 1.1854759 0.098    0.970
82  20 50 -0.0369001075 0.8618221 0.8621811 0.229    0.943
83  30 50  0.0191935240 0.6680284 0.6679701 0.322    0.961
84  40 50 -0.0076266623 0.5715790 0.5713441 0.416    0.949
85  50 50 -0.0283692453 0.5309160 0.5314083 0.483    0.937
86  60 50 -0.0287830245 0.4701709 0.4708164 0.544    0.940
87  70 50 -0.0222349591 0.4392776 0.4396206 0.606    0.951
88  80 50  0.0013181087 0.3987025 0.3985053 0.686    0.954
89  90 50 -0.0081176928 0.3758140 0.3757137 0.740    0.959
90 100 50  0.0020334640 0.3599674 0.3597931 0.792    0.948
```

```
#View(results)
```

Plot - Results

Bias

```
#cols <- c("gray", "gray", "gray","gray", "gray", "gray","gray", "gray", "blue")

ggplot(results, aes(x=n, y=Bias)) +
  geom_line(aes(color=as.factor(df)),linewidth =1)+
  guides(color = guide_legend(title = "DF")) #+
```



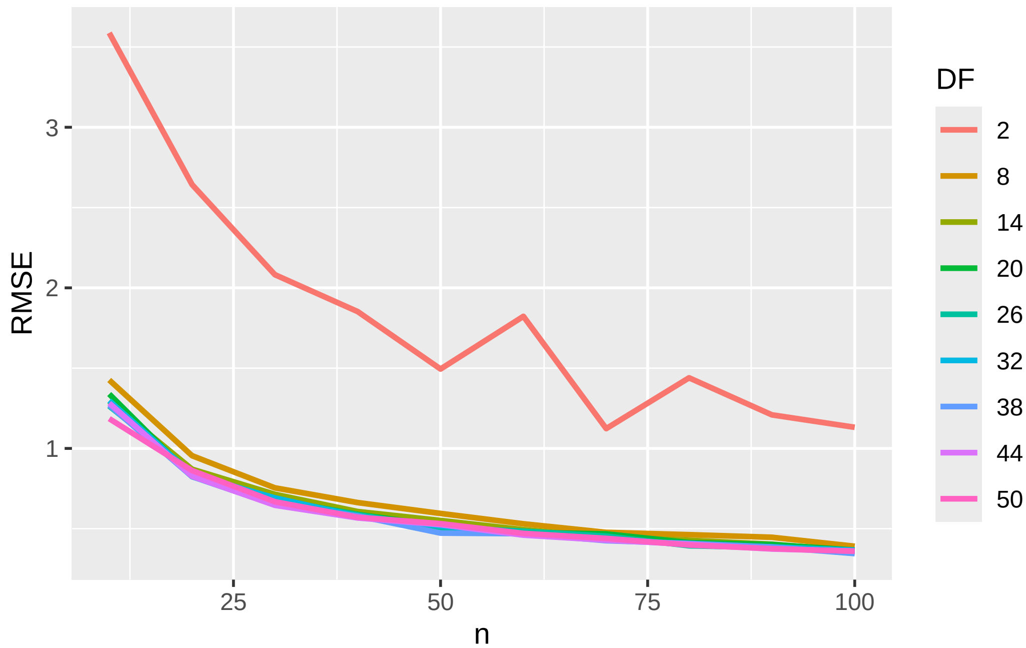

```
    #scale_color_manual(values = cols)
```

Bias plot suggests that as sample size increases the bias converges to zero irrespective of the df.

RMSE

```
#cols <- c("gray", "gray", "gray","gray", "gray", "gray","gray", "gray", "blue")

ggplot(results, aes(x=n, y=RMSE)) +
  geom_line(aes(color=as.factor(df)),linewidth =1)+
```

```
  guides(color = guide_legend(title = "DF"))#+
```

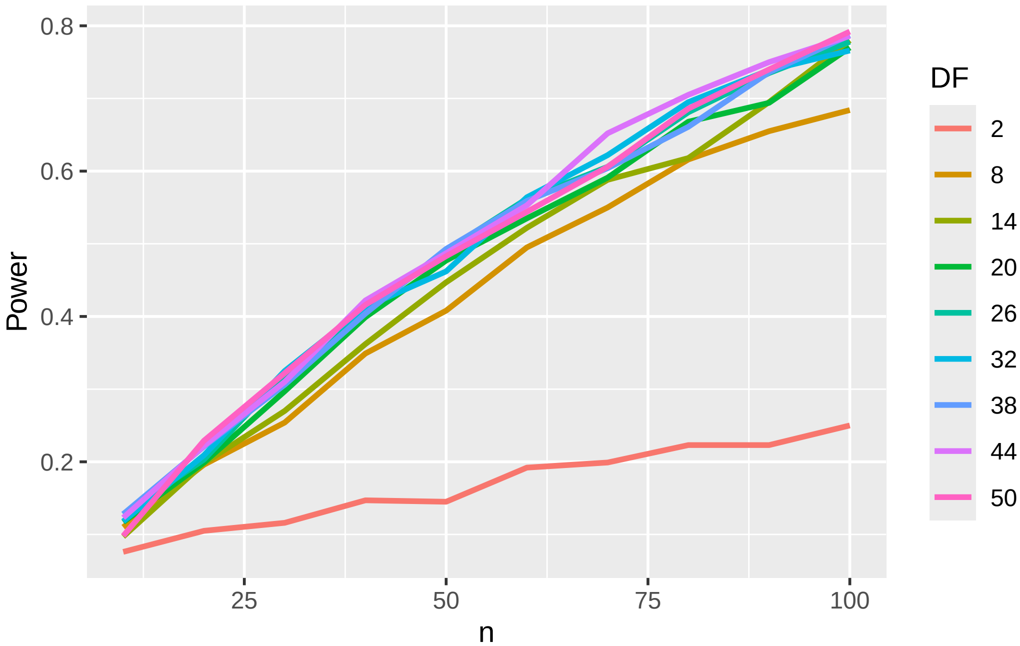



```
   #scale_color_manual(values = cols)
```

RMSE depicts that except df=2, RMSE are almost eqal irrespective of the df.

Power

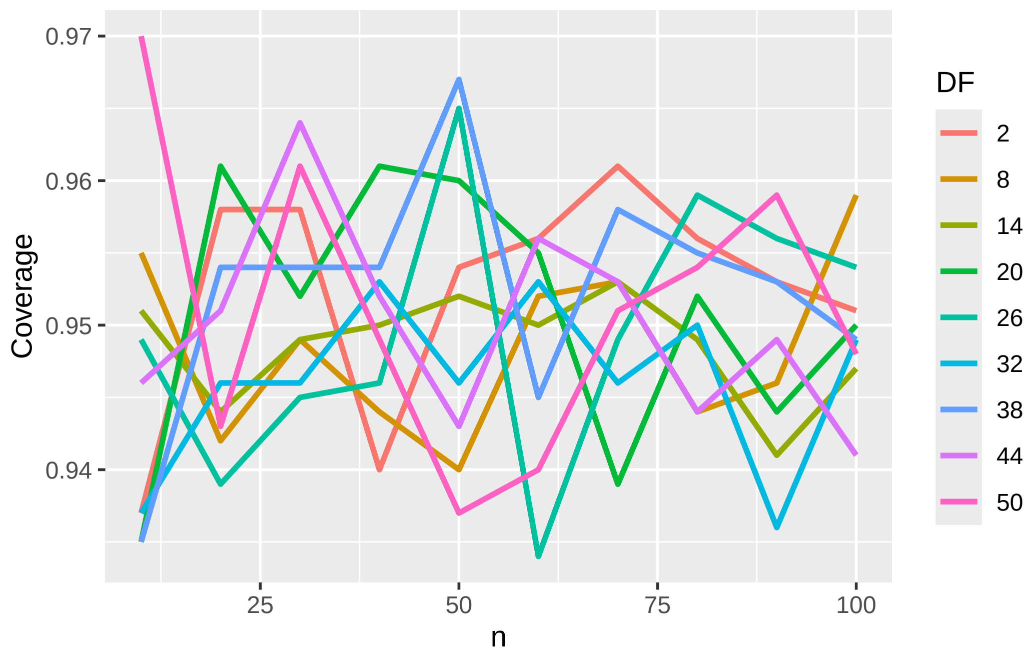
```
cols <- c("gray", "gray", "gray","gray", "gray", "gray","gray", "gray", "blue")

ggplot(results, aes(x=n, y=Power)) +
  geom_line(aes(color=as.factor(df)),linewidth =1)+
  guides(color = guide_legend(title = "DF"))
```



The plot suggest that as sample size and df increases the power increases.

Coverage

```
cols <- c("gray", "gray", "gray","gray", "gray", "gray","gray", "gray", "blue")

ggplot(results, aes(x=n, y=Coverage)) +
  geom_line(aes(color=as.factor(df)),linewidth =1)+
  guides(color = guide_legend(title = "DF"))
```



The coverage is highly oscillating irrespective of df.