431 Class 23

https://thomaselove.github.io/431-2024/

2024-11-19

## Today’s Agenda

* An example from NHANES 8/2021 - 8/2023
* Dealing with Missing Data, Partitioning, etc.
* Most of the things we’ve done previously, plus…
* Automated Variable Selection in Multiple Regression
  + Stepwise regression (AIC-based backwards elimination)
  + Best subsets (based on Mallows’ )
* Some rudimentary cross-validation approaches

## Today’s Packages

library(nhanesA)  
library(naniar)  
library(janitor)  
library(broom)  
library(gt)  
library(glue)  
library(car)  
library(GGally)  
library(mice)  
library(patchwork)  
library(olsrr) ## for best subsets  
library(xfun)   
library(easystats)  
library(tidyverse)  
  
theme\_set(theme\_bw())

## Today’s Data Source

[NHANES August 2021 - August 2023 Data](https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?Cycle=2021-2023)

In the next two slides, I’ll show the work I did to create a raw data set containing data on subjects from this administration of NHANES, using the nhanesA package, but I won’t execute that code.

Instead, I’ve provided an R data set of the raw data (**nh\_L\_raw.Rds**) which we’ll use shortly.

## Outcome and Potential Predictors

Today, we will try to predict waist circumference (in cm) using some combination of these 12 candidate predictors:

* body weight (in kg): expected to be primary driver
* demographics: age, sex, race/ethnicity, marital status, educational attainment, and insurance status
* systolic and diastolic blood pressure (1st reading), total cholesterol, self-rated oral health, sedentary minutes/day

## Today’s Data

Source: [NHANES August 2021 - August 2023 Data](https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?Cycle=2021-2023)

demoL <- nhanes('DEMO\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
bpxoL <- nhanes('BPXO\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
bmxL <- nhanes('BMX\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
tcholL <- nhanes('TCHOL\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
hscrpL <- nhanes('HSCRP\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
hiqL <- nhanes('HIQ\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
ohqL <- nhanes('OHQ\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))  
paqL <- nhanes('PAQ\_L', translated = FALSE) |>   
 tibble() |> mutate(SEQN = as.character(SEQN))

## Aggregating the Data

t12 <- left\_join(demoL, bpxoL, by = "SEQN")  
t123 <- left\_join(t12, bmxL, by = "SEQN")  
t1234 <- left\_join(t123, tcholL, by = "SEQN")  
t12345 <- left\_join(t1234, hiqL, by = "SEQN")  
t123456 <- left\_join(t12345, ohqL, by = "SEQN")  
nh\_L\_raw <- left\_join(t123456, paqL, by = "SEQN")  
  
write\_rds(nh\_L\_raw, "c23/data/nh\_L\_raw.Rds")

## Pruning the Rows

Let’s ingest the raw data from our Rds file.

Then we want to focus on adults ages 21-79 (Age is in the RIDAGEYR variable) who were both interviewed and MEC examined (RIDSTATR = 2)

nh\_L\_raw <- read\_rds("c23/data/nh\_L\_raw.Rds")  
dim(nh\_L\_raw)

[1] 11933 86

nh\_L <- nh\_L\_raw |>   
 filter(RIDSTATR == 2,  
 RIDAGEYR >= 21 & RIDAGEYR <= 79)  
dim(nh\_L)

[1] 5669 86

## Pruning the Variables

… and replacing Refused, or Don’t Know codes with NA.

nh\_L <- nh\_L |>  
 select(SEQN, RIAGENDR, RIDAGEYR, RIDRETH3, DMDEDUC2, DMDMARTZ,  
 BPXOSY1, BPXODI1, BMXWT, BMXWAIST, LBXTC,   
 HIQ011, HIQ032A, HIQ032B, HIQ032D,   
 OHQ845, PAD680, WTINT2YR, WTMEC2YR) |>  
 replace\_with\_na(replace =   
 list(DMDEDUC2 = c(7, 9), DMDMARTZ = c(77, 99), HIQ011 = c(7, 9),   
 HIQ032A = c(77, 99), OHQ845 = c(7, 9), PAD680 = c(7777, 9999)))

## Factor Cleaning (1/2)

nh\_L <- nh\_L |>   
 mutate(sex = fct\_recode(factor(RIAGENDR), "M" = "1", "F" = "2")) |>  
 mutate(race\_eth = fct\_recode(factor(RIDRETH3),   
 "Hispanic" = "1", "Hispanic" = "2", "NH\_White" = "3",   
 "NH\_Black" = "4", "NH\_Asian" = "6", "Other" = "7")) |>  
 mutate(educ = fct\_recode(factor(DMDEDUC2),  
 "NonHSGrad" = "1", "NonHSGrad" = "2", "HSGrad" = "3",  
 "SomeCollege" = "4", "CollegeGrad" = "5")) |>  
 mutate(marital = fct\_recode(factor(DMDMARTZ),  
 "Married" = "1", "Formerly" = "2", "Never" = "3"))

## Factor Cleaning (2/2)

nh\_L <- nh\_L |>   
 mutate(insur = factor(case\_when(  
 HIQ011 == 2 ~ "Uninsured",  
 HIQ011 == 1 & HIQ032D == 4 ~ "Medicaid",  
 HIQ011 == 1 & HIQ032B == 2 ~ "Medicare",  
 HIQ011 == 1 & HIQ032A == 1 ~ "Commercial"))) |>  
 mutate(insur = fct\_relevel(insur, "Medicare")) |>  
 mutate(oral\_h = fct\_recode(factor(OHQ845),  
 "E" = "1", "VG" = "2", "G" = "3", "F" = "4", "P" = "5"))

#### Variable Renaming

nh\_L <- nh\_L |>   
 rename(age = RIDAGEYR, sbp1 = BPXOSY1, dbp1 = BPXODI1,   
 wt\_kg = BMXWT, waist = BMXWAIST, tot\_chol = LBXTC,   
 sedentary = PAD680)

## Final Set: 16 Variables

nh\_L <- nh\_L |>   
 select(SEQN, sex, age, race\_eth, educ, marital, insur,   
 sbp1, dbp1, wt\_kg, waist, tot\_chol, oral\_h,   
 sedentary, WTINT2YR, WTMEC2YR)  
  
dim(nh\_L)

[1] 5669 16

## Variable Descriptions (1/3)

| Variable | Description | NA |
| --- | --- | --- |
| SEQN | Subject identifier code | 0 |
| sex | M or F | 0 |
| age | age in years (21-79) | 0 |
| race\_eth | race/ethnicity (5 categories) | 0 |
| educ | educational attainment (4 categories) | 3 |
| marital | marital status (3 categories) | 4 |

## Variable Descriptions (2/3)

| Variable | Description | NA |
| --- | --- | --- |
| insur | primary insurance (4 categories) | 557 |
| sbp1 | 1st Systolic BP reading (mm Hg) | 177 |
| dbp1 | 1st Diastolic BP reading (mm Hg) | 177 |
| wt\_kg | Body weight (kg) | 70 |
| waist | Waist circumference (cm) | 259 |
| tot\_chol | Total Cholesterol (mg/dl) | 516 |

## Variable Descriptions (3/3)

| Variable | Description | NA |
| --- | --- | --- |
| oral\_h | self-reported oral health (5 levels) | 9 |
| sedentary | typical day: sedentary activity (minutes) | 39 |
| WTINT2YR | sampling weight for interview items | 0 |
| WTMEC2YR | sampling weight for examination items | 0 |

## data\_codebook() results

data\_codebook(nh\_L |> select(-SEQN, -WTINT2YR, -WTMEC2YR))

select(nh\_L, -SEQN, -WTINT2YR, -WTMEC2YR) (5669 rows and 13 variables, 13 shown)  
  
ID | Name | Type | Missings | Values | N  
---+-----------+-------------+------------+---------------+-------------  
1 | sex | categorical | 0 (0.0%) | M | 2531 (44.6%)  
 | | | | F | 3138 (55.4%)  
---+-----------+-------------+------------+---------------+-------------  
2 | age | numeric | 0 (0.0%) | [21, 79] | 5669  
---+-----------+-------------+------------+---------------+-------------  
3 | race\_eth | categorical | 0 (0.0%) | Hispanic | 965 (17.0%)  
 | | | | NH\_White | 3298 (58.2%)  
 | | | | NH\_Black | 723 (12.8%)  
 | | | | NH\_Asian | 316 ( 5.6%)  
 | | | | Other | 367 ( 6.5%)  
---+-----------+-------------+------------+---------------+-------------  
4 | educ | categorical | 3 (0.1%) | NonHSGrad | 688 (12.1%)  
 | | | | HSGrad | 1192 (21.0%)  
 | | | | SomeCollege | 1741 (30.7%)  
 | | | | CollegeGrad | 2045 (36.1%)  
---+-----------+-------------+------------+---------------+-------------  
5 | marital | categorical | 4 (0.1%) | Married | 3112 (54.9%)  
 | | | | Formerly | 1365 (24.1%)  
 | | | | Never | 1188 (21.0%)  
---+-----------+-------------+------------+---------------+-------------  
6 | insur | categorical | 557 (9.8%) | Medicare | 1595 (31.2%)  
 | | | | Commercial | 2217 (43.4%)  
 | | | | Medicaid | 804 (15.7%)  
 | | | | Uninsured | 496 ( 9.7%)  
---+-----------+-------------+------------+---------------+-------------  
7 | sbp1 | numeric | 177 (3.1%) | [75, 232] | 5492  
---+-----------+-------------+------------+---------------+-------------  
8 | dbp1 | numeric | 177 (3.1%) | [40, 142] | 5492  
---+-----------+-------------+------------+---------------+-------------  
9 | wt\_kg | numeric | 70 (1.2%) | [27.9, 248.2] | 5599  
---+-----------+-------------+------------+---------------+-------------  
10 | waist | numeric | 259 (4.6%) | [60, 187] | 5410  
---+-----------+-------------+------------+---------------+-------------  
11 | tot\_chol | numeric | 516 (9.1%) | [62, 438] | 5153  
---+-----------+-------------+------------+---------------+-------------  
12 | oral\_h | categorical | 9 (0.2%) | E | 710 (12.5%)  
 | | | | VG | 1391 (24.6%)  
 | | | | G | 1872 (33.1%)  
 | | | | F | 1011 (17.9%)  
 | | | | P | 676 (11.9%)  
---+-----------+-------------+------------+---------------+-------------  
13 | sedentary | numeric | 39 (0.7%) | [0, 1380] | 5630  
------------------------------------------------------------------------

## Missing Data by Variable

miss\_var\_summary(nh\_L) |>   
 slice(1:8)

# A tibble: 8 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <num>  
1 insur 557 9.83   
2 tot\_chol 516 9.10   
3 waist 259 4.57   
4 sbp1 177 3.12   
5 dbp1 177 3.12   
6 wt\_kg 70 1.23   
7 sedentary 39 0.688  
8 oral\_h 9 0.159

miss\_var\_summary(nh\_L) |>   
 slice(9:16)

# A tibble: 8 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <num>  
1 marital 4 0.0706  
2 educ 3 0.0529  
3 SEQN 0 0   
4 sex 0 0   
5 age 0 0   
6 race\_eth 0 0   
7 WTINT2YR 0 0   
8 WTMEC2YR 0 0

## Missing Data

dim(nh\_L)

[1] 5669 16

miss\_case\_table(nh\_L)

# A tibble: 6 × 3  
 n\_miss\_in\_case n\_cases pct\_cases  
 <int> <int> <dbl>  
1 0 4357 76.9   
2 1 1005 17.7   
3 2 180 3.18   
4 3 75 1.32   
5 4 39 0.688  
6 5 13 0.229

pct\_miss\_case(nh\_L)

[1] 23.14341

## Missingness Mechanism

Can we assume the data are MCAR, per Little’s test?

mcar\_test(nh\_L)

# A tibble: 1 × 4  
 statistic df p.value missing.patterns  
 <dbl> <dbl> <dbl> <int>  
1 1022. 532 0 41

* Looks like we’ll need to assume MAR and do some imputation.

## Outcome and Potential Predictors

Again, our goal is to try to predict waist circumference (in cm) using some combination of these 12 candidate predictors:

* body weight (in kg): expected to be primary driver
* demographics: age, sex, race/ethnicity, marital status, educational attainment, and insurance status
* systolic and diastolic blood pressure, total cholesterol, self-rated oral health, sedentary minutes/day

We’ll generate several possible models.

## Single imputation, then partition

set.seed(202411191)  
nh\_i1 <- mice(nh\_L, m = 1, printFlag = FALSE)

Warning: Number of logged events: 1

nh\_si <- complete(nh\_i1) ## si for single imputation  
n\_miss(nh\_si)

[1] 0

We’ll use this single imputation for decision-making…

#### Partition the Data

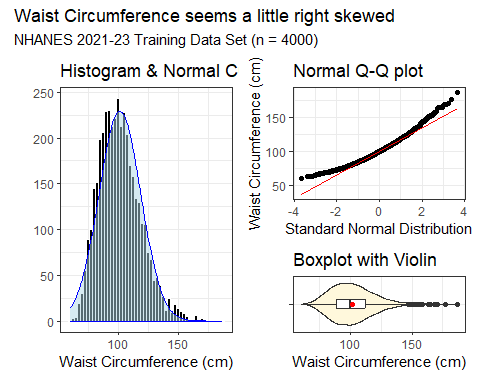
c(nrow(nh\_si), n\_distinct(nh\_si$SEQN)) # check that SEQNs are all distinct

[1] 5669 5669

set.seed(202411192)  
  
nh\_train <- nh\_si |> slice\_sample(n = 4000, replace = FALSE)  
nh\_test <- anti\_join(nh\_si, nh\_train, by = "SEQN")

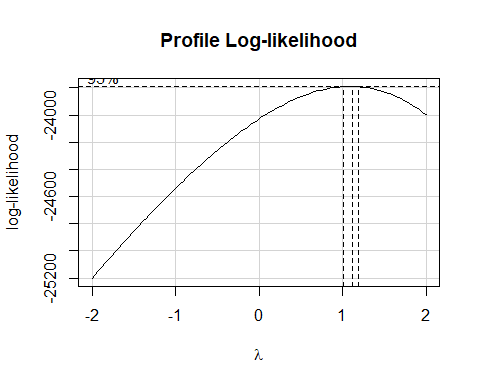
## Waist Circumference (our outcome)

bw = 2.5 # specify width of bins in histogram  
  
p1 <- ggplot(nh\_train, aes(x = waist)) +  
 geom\_histogram(binwidth = bw, fill = "black", col = "white") +  
 stat\_function(fun = function(x)  
 dnorm(x, mean = mean(nh\_train$waist, na.rm = TRUE),  
 sd = sd(nh\_train$waist, na.rm = TRUE)) \*  
 length(nh\_train$waist) \* bw,   
 geom = "area", alpha = 0.5, fill = "lightblue", col = "blue") +  
 labs(x = "Waist Circumference (cm)", y = "",  
 title = "Histogram & Normal Curve")   
  
p2 <- ggplot(nh\_train, aes(sample = waist)) +  
 geom\_qq() + geom\_qq\_line(col = "red") +  
 labs(y = "Waist Circumference (cm)",  
 x = "Standard Normal Distribution",  
 title = "Normal Q-Q plot")  
  
p3 <- ggplot(nh\_train, aes(x = waist, y = "")) +  
 geom\_violin(fill = "cornsilk") +  
 geom\_boxplot(width = 0.2) +  
 stat\_summary(fun = mean, geom = "point",   
 shape = 16, col = "red") +  
 labs(y = "", x = "Waist Circumference (cm)",  
 title = "Boxplot with Violin")  
  
p1 + (p2 / p3 + plot\_layout(heights = c(2, 1))) +  
 plot\_annotation(  
 title = "Waist Circumference seems a little right skewed",  
 subtitle = glue("NHANES 2021-23 Training Data Set (n = ", nrow(nh\_train), ")"))



## Transform our Outcome?

t\_check <- lm(waist ~ wt\_kg + age + sex + race\_eth + marital + educ +   
 insur + sbp1 + dbp1 + tot\_chol + oral\_h + sedentary,  
 data = nh\_train)  
boxCox(t\_check)



## Collinearity?

vif(t\_check)

GVIF Df GVIF^(1/(2\*Df))  
wt\_kg 1.236757 1 1.112096  
age 2.312126 1 1.520568  
sex 1.140010 1 1.067712  
race\_eth 1.447115 4 1.047280  
marital 1.350003 2 1.077913  
educ 1.456866 3 1.064723  
insur 2.744398 3 1.183245  
sbp1 2.204324 1 1.484697  
dbp1 2.003562 1 1.415472  
tot\_chol 1.103807 1 1.050622  
oral\_h 1.225628 4 1.025758  
sedentary 1.118783 1 1.057725

# Automated Predictor Selection

## Stepwise Regrssion Process

We start with a big model, including all candidate predictors.

* R looks at each variable in turn to see what would happen to AIC if we dropped it from the big model.
* It then drops the variable that most reduces the AIC, creating the first step down.
* Then it repeats the process with the newly stepped down model (where one variable has already been dropped) and drops the variable that most reduces the AIC now.
* When it can no longer drop a variable and reduce the AIC, the process stops.

## Stepwise Regression

We’ll start with our 12-predictor model, predicting our untransformed waist circumference.

fit\_all12 <- lm(waist ~ wt\_kg + age + sex + race\_eth + marital + educ +   
 insur + sbp1 + dbp1 + tot\_chol + oral\_h + sedentary,  
 data = nh\_train)

Here’s one way to run a backwards stepwise elimination search of our regression model, using AIC as the measure to tell us when to stop removing variables.

step\_res <- step(fit\_all12, direction = "backward")

Start: AIC=14453.44  
waist ~ wt\_kg + age + sex + race\_eth + marital + educ + insur +   
 sbp1 + dbp1 + tot\_chol + oral\_h + sedentary  
  
 Df Sum of Sq RSS AIC  
- marital 2 20 146612 14450  
- tot\_chol 1 16 146608 14452  
- sbp1 1 35 146627 14452  
<none> 146592 14453  
- sedentary 1 489 147081 14465  
- dbp1 1 639 147231 14469  
- insur 3 1099 147691 14477  
- oral\_h 4 1262 147854 14480  
- educ 3 3259 149851 14535  
- race\_eth 4 5295 151887 14587  
- sex 1 11840 158433 14762  
- age 1 18184 164776 14919  
- wt\_kg 1 851358 997950 22124  
  
Step: AIC=14449.99  
waist ~ wt\_kg + age + sex + race\_eth + educ + insur + sbp1 +   
 dbp1 + tot\_chol + oral\_h + sedentary  
  
 Df Sum of Sq RSS AIC  
- tot\_chol 1 18 146630 14448  
- sbp1 1 36 146648 14449  
<none> 146612 14450  
- sedentary 1 482 147095 14461  
- dbp1 1 639 147252 14465  
- insur 3 1105 147718 14474  
- oral\_h 4 1280 147892 14477  
- educ 3 3298 149910 14533  
- race\_eth 4 5465 152077 14588  
- sex 1 12143 158756 14766  
- age 1 20837 167450 14980  
- wt\_kg 1 851827 998440 22122  
  
Step: AIC=14448.48  
waist ~ wt\_kg + age + sex + race\_eth + educ + insur + sbp1 +   
 dbp1 + oral\_h + sedentary  
  
 Df Sum of Sq RSS AIC  
- sbp1 1 39 146670 14448  
<none> 146630 14448  
- sedentary 1 481 147112 14460  
- dbp1 1 697 147328 14466  
- insur 3 1089 147719 14472  
- oral\_h 4 1273 147904 14475  
- educ 3 3283 149913 14531  
- race\_eth 4 5560 152191 14589  
- sex 1 12325 158955 14769  
- age 1 21210 167840 14987  
- wt\_kg 1 860219 1006849 22153  
  
Step: AIC=14447.55  
waist ~ wt\_kg + age + sex + race\_eth + educ + insur + dbp1 +   
 oral\_h + sedentary  
  
 Df Sum of Sq RSS AIC  
<none> 146670 14448  
- sedentary 1 487 147156 14459  
- dbp1 1 891 147561 14470  
- insur 3 1077 147746 14471  
- oral\_h 4 1277 147946 14474  
- educ 3 3248 149917 14529  
- race\_eth 4 5658 152327 14591  
- sex 1 12980 159650 14785  
- age 1 22396 169066 15014  
- wt\_kg 1 880112 1026781 22230

Results on the next slide.

## So What Happened?

* The big model (12 predictors) had an AIC of 14422.72
* Dropping marital reduced the AIC to 14419.05
* Then dropping tot\_chol reduced the AIC to 14417.92
* Then dropping sbp1 reduced the AIC to 14417.22
* but at that point, no additional variable removals reduce the AIC. The best we can do is to drop sedentary but that increases AIC back to 14426.

## Stepwise Regression Result

So we drop three of the 12 predictors (marital, tot\_chol and sbp1) and keep the remaining nine (wt\_kg, age, sex, race\_eth, educ, income, oral\_h, dbp1, and sedentary).

fit\_step9 <- lm(waist ~ wt\_kg + age + sex + race\_eth + educ +   
 insur + dbp1 + oral\_h + sedentary, data = nh\_train)

It turns out that stepwise regression is a pretty poor approach, even by the standards of automated variable selection procedures. Are there better options?

* Can we generate other competitive candidate predictor sets?

## A Best Subsets Approach

Again, we start with our big model fit\_all12, with all 12 predictors, but now we use a function from the **olsrr** package to complete some automated variable selection[[1]](#footnote-56).

nh\_best\_res <-   
 ols\_step\_best\_subset(fit\_all12, max\_order = 6, metric = "cp")

This code identifies the best subsets of variables according to Mallows’ statistic, while restricting the search to models with 1-6 variables (of our original 12) included. Results on the next slide…

## Results from “Best Subsets” Search

nh\_best\_res

Best Subsets Regression   
-------------------------------------------------  
Model Index Predictors  
-------------------------------------------------  
 1 wt\_kg   
 2 wt\_kg age   
 3 wt\_kg age sex   
 4 wt\_kg age sex educ   
 5 wt\_kg age sex race\_eth educ   
 6 wt\_kg age sex race\_eth educ oral\_h   
-------------------------------------------------  
  
 Subsets Regression Summary   
----------------------------------------------------------------------------------------------------------------------------------------------  
 Adj. Pred   
Model R-Square R-Square R-Square C(p) AIC SBIC SBC MSEP FPE HSP APC   
----------------------------------------------------------------------------------------------------------------------------------------------  
 1 0.8145 0.8144 0.8142 2094.3495 27468.6621 16115.5453 27487.5442 224658.8296 56.1928 0.0141 0.1857   
 2 0.8563 0.8562 0.856 724.3907 26449.7113 15097.2432 26474.8875 174094.0151 43.5561 0.0109 0.1440   
 3 0.8662 0.8661 0.8659 398.5405 26163.6496 14811.4139 26195.1198 162037.9082 40.5500 0.0101 0.1340   
 4 0.8715 0.8713 0.871 232.4954 26009.7543 14653.7227 26060.1067 155727.3687 39.0000 0.0098 0.1288   
 5 0.8756 0.8753 0.8748 106.5218 25888.6589 14526.9014 25964.1875 150819.4446 37.8087 0.0095 0.1248   
 6 0.8770 0.8765 0.876 68.3982 25851.2327 14483.5912 25951.9375 149153.7010 37.4287 0.0094 0.1235   
----------------------------------------------------------------------------------------------------------------------------------------------  
AIC: Akaike Information Criteria   
 SBIC: Sawa's Bayesian Information Criteria   
 SBC: Schwarz Bayesian Criteria   
 MSEP: Estimated error of prediction, assuming multivariate normality   
 FPE: Final Prediction Error   
 HSP: Hocking's Sp   
 APC: Amemiya Prediction Criteria

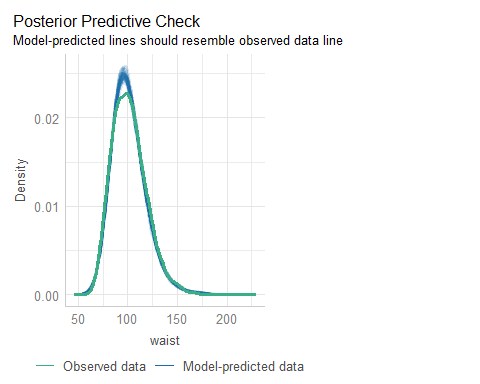
## We’ll compare four options

I’ve somewhat arbitrarily selected[[2]](#footnote-59) these four options.

fit\_12 <- lm(waist ~ wt\_kg + age + sex + race\_eth + marital + educ +   
 insur + sbp1 + dbp1 + tot\_chol + oral\_h + sedentary,  
 data = nh\_train)  
  
fit\_09 <- lm(waist ~ wt\_kg + age + sex + race\_eth + educ +   
 insur + dbp1 + oral\_h + sedentary, data = nh\_train)  
  
fit\_04 <- lm(waist ~ wt\_kg + age + sex + educ, data = nh\_train)  
  
fit\_01 <- lm(waist ~ wt\_kg, data = nh\_train)

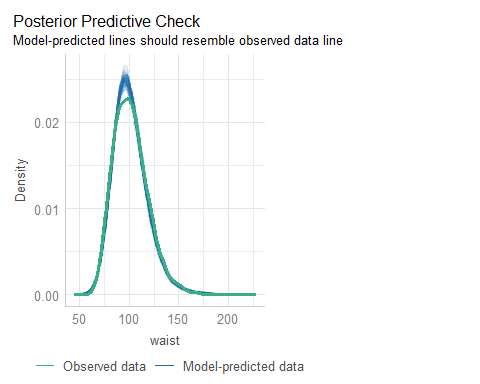
## Posterior Predictive Checks (fit\_12)

set.seed(43101); check\_model(fit\_12, check = "pp\_check")



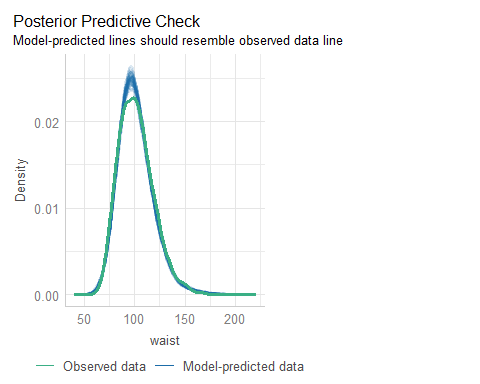
## Posterior Predictive Checks (fit\_09)

set.seed(43102); check\_model(fit\_09, check = "pp\_check")



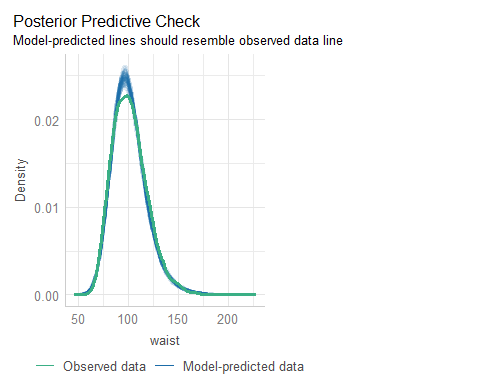
## Posterior Predictive Checks (fit\_04)

set.seed(43103); check\_model(fit\_04, check = "pp\_check")



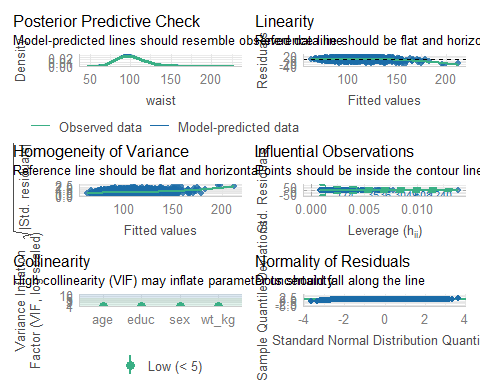
## Posterior Predictive Checks (fit\_01)

set.seed(43104); check\_model(fit\_01, check = "pp\_check")



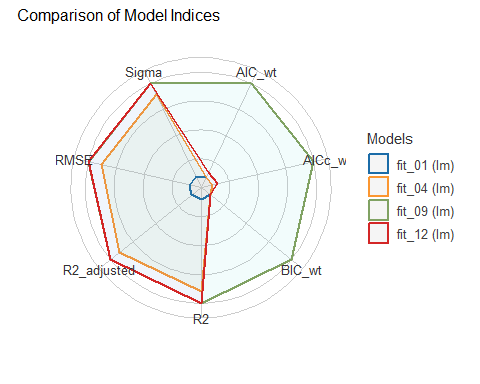
## Checking Model fit\_04

set.seed(43105); check\_model(fit\_04)



## Training Sample Performance

plot(compare\_performance(fit\_12, fit\_09, fit\_04, fit\_01))



## Training Sample Performance

compare\_performance(fit\_12, fit\_09, fit\_04, fit\_01, rank = TRUE) |>   
 gt() |> fmt\_number(decimals = 3) |> tab\_options(table.font.size = 24)

| Name | Model | R2 | R2\_adjusted | RMSE | Sigma | AIC\_wt | AICc\_wt | BIC\_wt | Performance\_Score |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fit\_09 | lm | 0.879 | 0.878 | 6.055 | 6.071 | 0.950 | 0.952 | 1.000 | 1.000 |
| fit\_12 | lm | 0.879 | 0.878 | 6.054 | 6.072 | 0.050 | 0.048 | 0.000 | 0.586 |
| fit\_04 | lm | 0.871 | 0.871 | 6.236 | 6.241 | 0.000 | 0.000 | 0.000 | 0.504 |
| fit\_01 | lm | 0.814 | 0.814 | 7.492 | 7.494 | 0.000 | 0.000 | 0.000 | 0.000 |

## Cross-Validation Performance

Within the training sample, we might try some cross-validation.

set.seed(2024111903)  
performance\_accuracy(fit\_12, method = "cv", k = 5)

# Accuracy of Model Predictions  
  
Accuracy (95% CI): 93.65% [92.64%, 94.15%]  
Method: Correlation between observed and predicted

performance\_cv(fit\_12)

# Cross-validation performance (30% holdout method)  
  
MSE | RMSE | R2  
-----------------  
39 | 6.2 | 0.88

This is something we’ll do in 432, too.

## Cross-Validation Results

across all four models we’re considering…

| Model | Accuracy |  |  |  |
| --- | --- | --- | --- | --- |
| fit\_12 | 0.9366 | 38 | 6.1 | 0.88 |
| fit\_09 | 0.9369 | 36 | 6 | 0.88 |
| fit\_04 | 0.9335 | 38 | 6.2 | 0.87 |
| fit\_01 | 0.9022 | 56 | 7.5 | 0.81 |

* Accuracy = correlation of waist, waist.
* Other summaries use 30% holdout approach.

## Predicting into Test Sample

aug01\_test <- augment(fit\_01, newdata = nh\_test) |> mutate(mod = "fit\_01")  
aug04\_test <- augment(fit\_04, newdata = nh\_test) |> mutate(mod = "fit\_04")  
aug09\_test <- augment(fit\_09, newdata = nh\_test) |> mutate(mod = "fit\_09")  
aug12\_test <- augment(fit\_12, newdata = nh\_test) |> mutate(mod = "fit\_12")  
  
temp14 <- bind\_rows(aug01\_test, aug04\_test)  
temp149 <- bind\_rows(temp14, aug09\_test)  
test\_res <- bind\_rows(temp149, aug12\_test) |>  
 relocate(SEQN, mod, waist, everything()) |> arrange(SEQN, mod)

## Four Summary Measures

test\_summary <- test\_res |>   
 group\_by(mod) |>  
 summarize(MAPE = mean(abs(.resid)),  
 Max\_APE = max(abs(.resid)),  
 RMSE = sqrt(mean(.resid^2)),  
 R2\_val = cor(waist, .fitted)^2)  
  
test\_summary |> gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 28)

| mod | MAPE | Max\_APE | RMSE | R2\_val |
| --- | --- | --- | --- | --- |
| fit\_01 | 5.8849 | 27.6462 | 7.3967 | 0.8191 |
| fit\_04 | 4.8477 | 26.4983 | 6.2469 | 0.8713 |
| fit\_09 | 4.7499 | 27.7847 | 6.1422 | 0.8755 |
| fit\_12 | 4.7504 | 27.8797 | 6.1418 | 0.8756 |

# Multiple Imputation

## We’d like to do 25 imputations

25 imputations on model fit\_04 takes quite a while (7-10 minutes on my machine.)

pct\_miss\_case(nh\_L)

[1] 23.14341

imp25\_ests <- nh\_L |>  
 mice(m = 25, seed = 2024, print = FALSE) |>  
 with(lm(waist ~ wt\_kg + age + sex + educ)) |>  
 pool()

Warning: Number of logged events: 1

## Pooled Results for model fit\_04

glance(imp25\_ests)

nimp nobs r.squared adj.r.squared  
1 25 5669 0.8722111 0.8720756

tidy(imp25\_ests, conf.int = TRUE, conf.level = 0.90) |>  
 select(term, estimate, std.error, conf.low, conf.high, p.value) |>  
 gt() |> fmt\_number(decimals = 3, columns = c(-p.value))

| term | estimate | std.error | conf.low | conf.high | p.value |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | 30.488 | 0.537 | 29.604 | 31.372 | 0.000000e+00 |
| wt\_kg | 0.720 | 0.004 | 0.713 | 0.727 | 0.000000e+00 |
| age | 0.211 | 0.005 | 0.203 | 0.220 | 1.653545e-283 |
| sexF | 3.695 | 0.174 | 3.408 | 3.982 | 9.271242e-95 |
| educHSGrad | -1.584 | 0.305 | -2.085 | -1.082 | 2.122695e-07 |
| educSomeCollege | -2.634 | 0.287 | -3.106 | -2.162 | 6.783848e-20 |
| educCollegeGrad | -3.934 | 0.279 | -4.394 | -3.474 | 5.797537e-44 |

## Training Sample Results for fit\_04

glance(fit\_04) |> select(nobs, r.squared, adj.r.squared)

# A tibble: 1 × 3  
 nobs r.squared adj.r.squared  
 <int> <dbl> <dbl>  
1 4000 0.871 0.871

tidy(fit\_04, conf.int = TRUE, conf.level = 0.90) |>  
 select(term, estimate, std.error, conf.low, conf.high, p.value) |>  
 gt() |> fmt\_number(decimals = 3, columns = c(-p.value))

| term | estimate | std.error | conf.low | conf.high | p.value |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | 30.159 | 0.609 | 29.158 | 31.161 | 0.000000e+00 |
| wt\_kg | 0.717 | 0.005 | 0.710 | 0.725 | 0.000000e+00 |
| age | 0.216 | 0.006 | 0.206 | 0.226 | 1.486143e-239 |
| sexF | 3.712 | 0.205 | 3.375 | 4.049 | 1.432568e-70 |
| educHSGrad | -1.321 | 0.357 | -1.908 | -0.733 | 2.210386e-04 |
| educSomeCollege | -2.444 | 0.337 | -2.998 | -1.890 | 4.743961e-13 |
| educCollegeGrad | -3.746 | 0.328 | -4.286 | -3.205 | 1.166192e-29 |

## Single Imputation Results for fit\_04

For the entire sample after single imputation. Our model fit\_04 uses wt\_kg, age, sex and educ to predict waist. Our identifying code is in SEQN.

fit4\_si <- lm(waist ~ wt\_kg + age + sex + educ, data = nh\_si)  
  
n\_obs(fit4\_si)

[1] 5669

## Single Imputation Results for fit\_04

glance(fit4\_si) |> select(nobs, r.squared, adj.r.squared)

# A tibble: 1 × 3  
 nobs r.squared adj.r.squared  
 <int> <dbl> <dbl>  
1 5669 0.871 0.871

tidy(fit4\_si, conf.int = TRUE, conf.level = 0.90) |>  
 select(term, estimate, std.error, conf.low, conf.high, p.value) |>  
 gt() |> fmt\_number(decimals = 3, columns = c(-p.value))

| term | estimate | std.error | conf.low | conf.high | p.value |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | 30.599 | 0.513 | 29.755 | 31.443 | 0.000000e+00 |
| wt\_kg | 0.718 | 0.004 | 0.712 | 0.724 | 0.000000e+00 |
| age | 0.213 | 0.005 | 0.204 | 0.221 | 0.000000e+00 |
| sexF | 3.716 | 0.172 | 3.433 | 4.000 | 3.985320e-99 |
| educHSGrad | -1.652 | 0.299 | -2.144 | -1.159 | 3.617931e-08 |
| educSomeCollege | -2.745 | 0.282 | -3.209 | -2.280 | 3.666489e-22 |
| educCollegeGrad | -4.007 | 0.276 | -4.460 | -3.553 | 5.009061e-47 |

## Building “Complete Case” Data

Our model fit\_04 uses wt\_kg, age, sex and educ to predict waist. Our identifying code is in SEQN.

nh\_cc <- nh\_L |> select(SEQN, waist, wt\_kg, age, sex, educ) |>  
 drop\_na()  
  
fit4\_cc <- lm(waist ~ wt\_kg + age + sex + educ, data = nh\_cc)  
  
n\_obs(fit4\_cc)

[1] 5397

## “Complete Case” Results for fit\_04

glance(fit4\_cc) |> select(nobs, r.squared, adj.r.squared)

# A tibble: 1 × 3  
 nobs r.squared adj.r.squared  
 <int> <dbl> <dbl>  
1 5397 0.870 0.870

tidy(fit4\_cc, conf.int = TRUE, conf.level = 0.90) |>  
 select(term, estimate, std.error, conf.low, conf.high, p.value) |>  
 gt() |> fmt\_number(decimals = 3, columns = c(-p.value))

| term | estimate | std.error | conf.low | conf.high | p.value |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | 30.194 | 0.526 | 29.329 | 31.059 | 0.000000e+00 |
| wt\_kg | 0.724 | 0.004 | 0.717 | 0.730 | 0.000000e+00 |
| age | 0.211 | 0.005 | 0.202 | 0.219 | 8.528098e-314 |
| sexF | 3.723 | 0.174 | 3.436 | 4.010 | 4.161630e-97 |
| educHSGrad | -1.606 | 0.305 | -2.107 | -1.104 | 1.442240e-07 |
| educSomeCollege | -2.669 | 0.287 | -3.141 | -2.197 | 1.964979e-20 |
| educCollegeGrad | -3.972 | 0.280 | -4.432 | -3.512 | 5.598585e-45 |

## Our 4-predictor model

* Use wt\_kg, age, sex and educ to predict waist.

| Method | Sample |  | wt\_kg (90% CI) |
| --- | --- | --- | --- |
| Complete Cases | 5,397 | 0.870 | 0.724 (0.717, 0.730) |
| Training Sample | 4,000 | 0.871 | 0.717 (0.710, 0.725) |
| Single Imp. (all) | 5,669 | 0.871 | 0.718 (0.712, 0.724) |
| Multiple Imp. (25) | 5,669 | 0.872 | 0.720 (0.713, 0.727) |

## Session Information

xfun::session\_info()

R version 4.4.2 (2024-10-31 ucrt)  
Platform: x86\_64-w64-mingw32/x64  
Running under: Windows 11 x64 (build 22631)  
  
Locale:  
 LC\_COLLATE=English\_United States.utf8   
 LC\_CTYPE=English\_United States.utf8   
 LC\_MONETARY=English\_United States.utf8  
 LC\_NUMERIC=C   
 LC\_TIME=English\_United States.utf8   
  
Package version:  
 abind\_1.4-8 AsioHeaders\_1.22.1.2 askpass\_1.2.1   
 backports\_1.5.0 base64enc\_0.1.3 bayestestR\_0.15.0   
 bigD\_0.3.0 bit\_4.5.0 bit64\_4.5.2   
 bitops\_1.0.9 blob\_1.2.4 boot\_1.3-31   
 broom\_1.0.7 bslib\_0.8.0 cachem\_1.1.0   
 callr\_3.7.6 car\_3.1-3 carData\_3.0-5   
 cellranger\_1.1.0 chromote\_0.3.1 cli\_3.6.3   
 clipr\_0.8.0 coda\_0.19-4.1 codetools\_0.2-20   
 colorspace\_2.1-1 commonmark\_1.9.2 compiler\_4.4.2   
 conflicted\_1.2.0 correlation\_0.8.6 cowplot\_1.1.3   
 cpp11\_0.5.0 crayon\_1.5.3 curl\_6.0.1   
 data.table\_1.16.2 datasets\_4.4.2 datawizard\_0.13.0   
 DBI\_1.2.3 dbplyr\_2.5.0 Deriv\_4.1.6   
 digest\_0.6.37 doBy\_4.6.24 dplyr\_1.1.4   
 dtplyr\_1.3.1 easystats\_0.7.3 effectsize\_0.8.9   
 emmeans\_1.10.5 estimability\_1.5.1 evaluate\_1.0.1   
 fansi\_1.0.6 farver\_2.1.2 fastmap\_1.2.0   
 fontawesome\_0.5.3 forcats\_1.0.0 foreach\_1.5.2   
 foreign\_0.8-87 Formula\_1.2-5 fs\_1.6.5   
 gargle\_1.5.2 generics\_0.1.3 GGally\_2.2.1   
 ggplot2\_3.5.1 ggrepel\_0.9.6 ggstats\_0.7.0   
 glmnet\_4.1-8 glue\_1.8.0 goftest\_1.2-3   
 googledrive\_2.1.1 googlesheets4\_1.1.1 graphics\_4.4.2   
 grDevices\_4.4.2 grid\_4.4.2 gridExtra\_2.3   
 gt\_0.11.1 gtable\_0.3.6 haven\_2.5.4   
 highr\_0.11 hms\_1.1.3 htmltools\_0.5.8.1   
 htmlwidgets\_1.6.4 httpuv\_1.6.15 httr\_1.4.7   
 ids\_1.0.1 insight\_0.20.5 isoband\_0.2.7   
 iterators\_1.0.14 janitor\_2.2.0 jomo\_2.7-6   
 jquerylib\_0.1.4 jsonlite\_1.8.9 juicyjuice\_0.1.0   
 knitr\_1.49 labeling\_0.4.3 later\_1.3.2   
 lattice\_0.22-6 lifecycle\_1.0.4 lme4\_1.1-35.5   
 lubridate\_1.9.3 magrittr\_2.0.3 markdown\_1.13   
 MASS\_7.3-61 Matrix\_1.7-1 MatrixModels\_0.5.3   
 memoise\_2.0.1 methods\_4.4.2 mgcv\_1.9-1   
 mice\_3.16.0 microbenchmark\_1.5.0 mime\_0.12   
 minqa\_1.2.8 mitml\_0.4-5 modelbased\_0.8.9   
 modelr\_0.1.11 multcomp\_1.4-26 munsell\_0.5.1   
 mvtnorm\_1.3-2 naniar\_1.1.0 nhanesA\_1.1   
 nlme\_3.1-166 nloptr\_2.1.1 nnet\_7.3-19   
 norm\_1.0-11.1 nortest\_1.0-4 numDeriv\_2016.8.1.1   
 olsrr\_0.6.1 openssl\_2.2.2 ordinal\_2023.12.4.1   
 pan\_1.9 parallel\_4.4.2 parameters\_0.23.0   
 patchwork\_1.3.0 pbkrtest\_0.5.3 performance\_0.12.4   
 pillar\_1.9.0 pkgconfig\_2.0.3 plyr\_1.8.9   
 prettyunits\_1.2.0 processx\_3.8.4 progress\_1.2.3   
 promises\_1.3.0 ps\_1.8.1 purrr\_1.0.2   
 quantreg\_5.99 R6\_2.5.1 ragg\_1.3.3   
 rappdirs\_0.3.3 RColorBrewer\_1.1-3 Rcpp\_1.0.13-1   
 RcppEigen\_0.3.4.0.2 reactable\_0.4.4 reactR\_0.6.1   
 readr\_2.1.5 readxl\_1.4.3 rematch\_2.0.0   
 rematch2\_2.1.2 report\_0.5.9 reprex\_2.1.1   
 rlang\_1.1.4 rmarkdown\_2.29 rpart\_4.1.23   
 rstudioapi\_0.17.1 rvest\_1.0.4 sandwich\_3.1-1   
 sass\_0.4.9 scales\_1.3.0 see\_0.9.0   
 selectr\_0.4.2 shape\_1.4.6.1 shiny\_1.9.1   
 snakecase\_0.11.1 sourcetools\_0.1.7.1 SparseM\_1.84.2   
 splines\_4.4.2 stats\_4.4.2 stringi\_1.8.4   
 stringr\_1.5.1 survival\_3.7-0 sys\_3.4.3   
 systemfonts\_1.1.0 textshaping\_0.4.0 TH.data\_1.1-2   
 tibble\_3.2.1 tidyr\_1.3.1 tidyselect\_1.2.1   
 tidyverse\_2.0.0 timechange\_0.3.0 tinytex\_0.54   
 tools\_4.4.2 tzdb\_0.4.0 ucminf\_1.2.2   
 UpSetR\_1.4.0 utf8\_1.2.4 utils\_4.4.2   
 uuid\_1.2.1 V8\_6.0.0 vctrs\_0.6.5   
 viridis\_0.6.5 viridisLite\_0.4.2 visdat\_0.6.0   
 vroom\_1.6.5 websocket\_1.4.2 withr\_3.0.2   
 xfun\_0.49 xml2\_1.3.6 xplorerr\_0.2.0   
 xtable\_1.8-4 yaml\_2.3.10 zoo\_1.8-12

1. *Caution*: this process takes a while. [↑](#footnote-ref-56)
2. fit12 is the “kitchen sink”, fit09 is “stepwise”, fit04 has a nice adjusted , and fit01 is the most crucial predictor. [↑](#footnote-ref-59)