Factors that contribute to hypertension

Kexin Wang, M.S.

Contents

Motivation

The link between hypertension and some physical measurements has been well-established in previous studies. In this case study, moving beyond the traditional aspects, we will explore other risk factors and their potential influence on hypertension.

This case study also introduces logistic regression and survey-weighted logistic regression, focusing on the comparison between them. The result plot indicates that standard error of coefficients calculated by logistic regression is not accurate. So when we use survey data, survey-weight logistic regression is a good choice of model in this setting.

What is the Data

NYC HANES 2013-14 Blood Pressure Data: NYC HANES Analytics Datasets is downloaded here. Other useful resources can be found through it, such as 'Data Documents', 'Variable List','Analytics Guideline','Questionnaires'. All of them enable data users to understand the meaning and encoding of the variables better, and then complete the analysis more efficiently.

Other Resource

1. National Health and Nutrition Examination Survey (NHANES): A program of studies designed to assess the health and nutritional status of adults and children in the United States. NYC HANES is a local version of NHANES, which implies it mainly focus on New York area. Modeling on NHANES, NYC HANES collected data from a physical examination and laboratory tests, as well as interviews.
2. R package nhanesA: This package was developed to enable fully customizable retrieval of data from NHANES, such as load data, list variables in NHANES table.

Data Preprocessing

Load packages

Since this is the first R code chunk, we will load the necessary libraries.

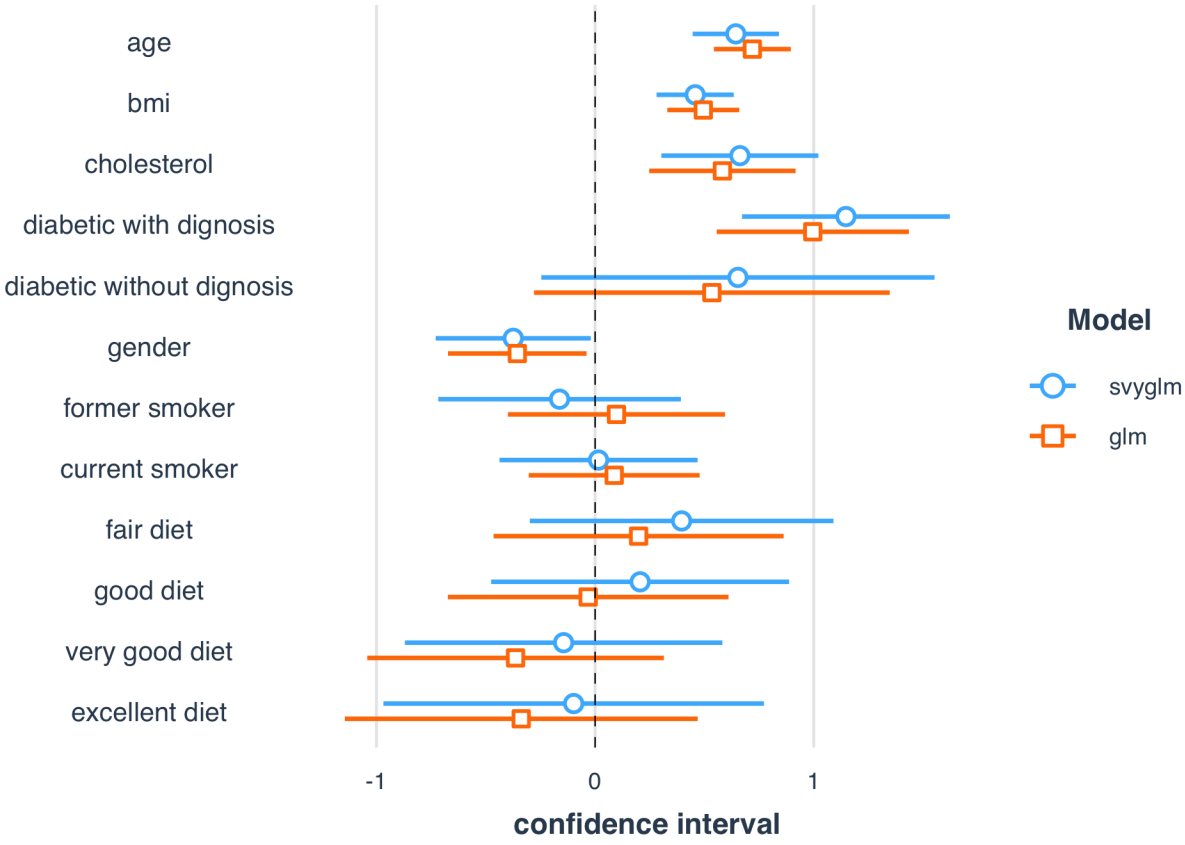

age
bmi
cholesterol
diabetic with dignosis
diabetic without dignosis
gender
former smoker
current smoker
fair diet
good diet
very good diet
excellent diet
-1
0
1
confidence interval
Model
svyglm
glm

Figure 1:

```
library(knitr)
library(ggplot2)
library(ggpubr)
library(ggrepel)
library(tableone)
library(kableExtra)
library(survey)
library(broom)
library(Rmisc)
library(tidyverse)
library(haven)
library(jtools)
library(plotrix)
```

Load Data

It is a SAS formatted file so we would like to use the function `read_sas` from the `haven` library to read it into a data frame in R. `haven` library is useful to import and export 'SAS', 'STATA', and 'SPSS' file.

```
dat <- read_sas('./data/dat.sas7bdat')
```

This data frame contains 1527 observations of 704 different variables. In our analysis, only a subset of the variables will be selected.

Select the variables that we are interested in

As we mentioned above, this is a survey data set based on interview or questionnaires with 704 variables. Some variables are meaningless to our research interest, such as 'LAQ1: What language is being used to conduct this interview'.

Previous people showed hypertension has relationship with drink, smoking, cholesterol values, triglyceride. Except for them, we still want to choose other covariates which might not be highly-related, such as income, and try to see whether they have a potential association with hypertension. Therefore, based on our interest, finally selected 13 covariates are kept.

Here, we use the `select` function to choose and rename the columns that we want.

```
hypertension_DF <- dat %>%
  select(
id = KEY,
age = SPAGE,
race = DMQ_14_1,
gender = GENDER,
diet = DBQ_1,
income = INC20K,
diabetes = DX_DBTS,
bmi = BMI,
cholesterol = BPQ_16,
drink = ALQ_1_UNIT,
smoking = SMOKER3CAT,
hypertension = BPQ_2,
surveyweight = CAPI_WT
)
```

Here are explanations for these variables:

-Non Categorical

- id: Sample case ID
- age: Sample age, range 22-115
- bmi: BMI = kg/m2 where kg is a person's weight in kilograms and m2 is their height in meters squared
- surveyweight: surveyweight

-Categorical

- race:
  - 100 = White
  - 110 = Black/African American
  - 120 = Indian
  - 140 = Native Hawaiian/Other Pacific Islander
  - 180 = Asian
  - 250 = Other race
- gender:
  - 1 = Male
  - 2 = Female
- born:
  - 1 = Us born
  - 2 = Other country
- diet:
  - 1 = Excellent
  - 2 = Very good
  - 3 = Good
  - 4 = Fair
  - 5 = Poor
- diabetes: Previously diagnosed with diabetes
  - 1 = Diabetic with dignosis
  - 2 = Diabetic without dignosis
  - 3 = Not diabetic
- cholesterol: An oil-based substance. If concentrations get too high, it puts people at risk of heart diseases
  - 1 = High cholesterol value
  - 2 = Low cholesterol value
- drink: In the past 12 months, how often did sample drink any type of alcoholic beverage
  - 1 = Weekly
  - 2 = Monthly
  - 3 = Yearly
- smoke:
  - 1 = Never smoker
  - 2 = Current smoker
  - 3 = Former smoker
- income:
  - 1 = Less than $20,000
  - 2 = $20,000 - $39,999
  - 3 = $40,000 - $59,999
  - 4 = $60,000 - $79,999
  - 5 = $80,000 - $99,999
  - 6 = $100,000 or more
- hypertension: Previously diagnosed as hypertension
  - 1 = Yes
  - 2 = No

Data Wrangling

The first step of any data analysis should be to explore the data through calculating various summary statistics. There are several ways that you can have a glance at your data. Plotting the data or using the `summary` or `head` functions are excellent ways to help you have a quick judgement on the data set.

The `summary` function tabulates categorical variables and provides summary statistics for continuous ones, while also including a count of missing values, which can be very important in deciding what variables to consider in downstream analysis.

```
summary(hypertension_DF)
```

```
##       id                 age              race           gender
##  Length:1527        Min.   :20.00   Min.   :100.0   Min.   :1.00
##  Class :character   1st Qu.:30.00   1st Qu.:100.0   1st Qu.:1.00
##  Mode  :character   Median :42.00   Median :110.0   Median :2.00
##                     Mean   :44.55   Mean   :136.8   Mean   :1.58
##                     3rd Qu.:57.00   3rd Qu.:180.0   3rd Qu.:2.00
##                     Max.   :97.00   Max.   :250.0   Max.   :2.00
##                                     NA's   :59
##       diet           income         diabetes          bmi
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :12.31
##  1st Qu.:2.00   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:23.33
##  Median :3.00   Median :2.000   Median :3.000   Median :26.52
##  Mean   :2.92   Mean   :2.985   Mean   :2.719   Mean   :27.73
##  3rd Qu.:4.00   3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:30.71
##  Max.   :5.00   Max.   :6.000   Max.   :3.000   Max.   :69.17
##  NA's   :3      NA's   :161     NA's   :281     NA's   :38
##   cholesterol        drink          smoking       hypertension
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :1.000   Median :1.000   Median :2.000
##  Mean   :1.719   Mean   :1.726   Mean   :1.598   Mean   :1.726
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :2.000   Max.   :3.000   Max.   :3.000   Max.   :2.000
##  NA's   :15      NA's   :412     NA's   :3       NA's   :3
##   surveyweight
##  Min.   : 1030
##  1st Qu.: 2875
##  Median : 3648
##  Mean   : 4116
##  3rd Qu.: 4989
##  Max.   :15422
##
```

We find NA's occupy big parts of the dataset, even 412 for 'drink' and 281 for 'diabetes'. Directly removing rows containing missing data is not appropriate considering its large amount. So we try to check the variable list again and find another variable 'AlQ_1' : how often did SP drink any type of alcoholic beverage? 0 means they never drink.

Let's see its distribution with function `table()`.

```
table(dat$ALQ_1)
```

```
##
##   0   1   2   3   4   5   6   7   8  10  12  14  15  16  17  20  24  25
## 406 267 276 183 100  80  40  68   7  25   9   6   6   1   1  14   6   2
```

```
##  28  30  40  50  60 100 144 180 189 200 365
##   1   8   1   1   1   1   1   1   1   1   7
```

The result answers why there are so many missing value for 'drink'. Among these 412 missing values, 406 samples never drink and there are just 6 real missing values. Therefore, merging these two variables as one is a better choice to avoid losing too many observations.

```
hypertension_DF$drink[which(dat$ALQ_1==0)] <- 4
summary(hypertension_DF$drink)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   1.000   2.000   2.333   4.000   4.000       6
```

With the help of function `which()`, 6 missing value for 'drink' is left now.

In next step, we aim to remove rows containing missing data with a nice function `drop_na()` in library `tidyr` and store in a new dataframe:

```
DF <- hypertension_DF %>%
  drop_na()
```

Finally we remain 1065 observations of 13 different variables.

Adjust Data Type

From the data summaries above, we can see that there are several categorical variables like race, gender, born, diet, income, diabetes, BMI, drink, and smoke, which are currently being treated as numerical values.

For these variables, it is important for us to adjust the coding, prior to doing any analysis in R. Since different value represent nothing but different categories. Sometimes people forget to change the type of data, and usually, it will give you a totally wrong result.

For example, in `lm()` and `glm()`, the function will treat numerically-coded categorical variables as continuous variables, which will give the wrong result. Instead, we want to convert these categorical variables to factors. Before doing this, we want to get a better understanding of exactly what values are stored in the different variables.

The first step for these categorical variables is to change them to factors. We refer to the codebook to get the correct mapping of the numerical values to the category labels.

We can use the `factor` function to convert each variable and assign the correct levels. Any values that are not included in the 'levels' argument will get set to 'NA' values.

```
DF$race <- factor(DF$race, levels=c(100, 110, 120, 140, 180, 250),
labels=c('White', 'Black/African American',
'Indian /Alaska Native',
'Pacific Islander', 'Asian', 'Other Race'))

DF$gender <-  factor(DF$gender, levels=c(1,2),
labels=c('Male', 'Female'))

DF$diet <-  factor(DF$diet, levels=c(5:1),
labels=c('Poor', 'Fair', 'Good', 'Very good','Excellent'))

DF$income <-  factor(DF$income, levels=c(1:6),
labels=c('Less than $20,000','$20,000 - $39,999',
'$40,000 - $59,999','$60,000 - $79,999',
'$80,000 - $99,999','$100,000 or more'))
```

```
DF$diabetes <-  factor(DF$diabetes, levels=c(3,1,2),
labels=c('Not diabetic','Diabetic dx','Diabetic but no dx'))

DF$cholesterol <-  factor(DF$cholesterol, levels=c(2,1),
labels=c('Low value','High value'))

DF$drink <-  factor(DF$drink, levels=c(4,1,2,3),
labels=c('Never','Weekly', 'Monthly', 'Yearly'))

DF$smoking <-  factor(DF$smoking, levels=c(3:1),
labels=c('Never smoker','Former smoker','Current smoker'))

DF$hypertension <-  factor(DF$hypertension, levels=c(2,1),
labels=c('No','Yes'))
```

```
summary(DF)
```

```
##       id                 age                             race
##  Length:1065        Min.   :20.00   White                  :491
##  Class :character   1st Qu.:30.00   Black/African American:258
##  Mode  :character   Median :42.00   Indian /Alaska Native :  7
##                     Mean   :44.54   Pacific Islander      :  5
##                     3rd Qu.:57.00   Asian                  :143
##                     Max.   :94.00   Other Race             :161
##     gender           diet                    income
##  Male  :460   Poor     : 72   Less than $20,000:309
##  Female:605   Fair     :235   $20,000 - $39,999:246
##               Good     :391   $40,000 - $59,999:124
##               Very good:264   $60,000 - $79,999:110
##               Excellent:103   $80,000 - $99,999: 75
##                               $100,000 or more :201
##                diabetes        bmi            cholesterol      drink
##  Not diabetic      :911   Min.   :15.02   Low value :758   Never  :255
##  Diabetic dx       :126   1st Qu.:23.40   High value:307   Weekly :424
##  Diabetic but no dx: 28   Median :26.47                    Monthly:201
##                           Mean   :27.76                    Yearly :185
##                           3rd Qu.:30.47
##                           Max.   :69.17
##            smoking    hypertension  surveyweight
##  Never smoker  :230   No :775      Min.   : 1030
##  Former smoker :204   Yes:290      1st Qu.: 2864
##  Current smoker:631                Median : 3559
##                                    Mean   : 4028
##                                    3rd Qu.: 4786
##                                    Max.   :15422
```

EXplore Data Analysis

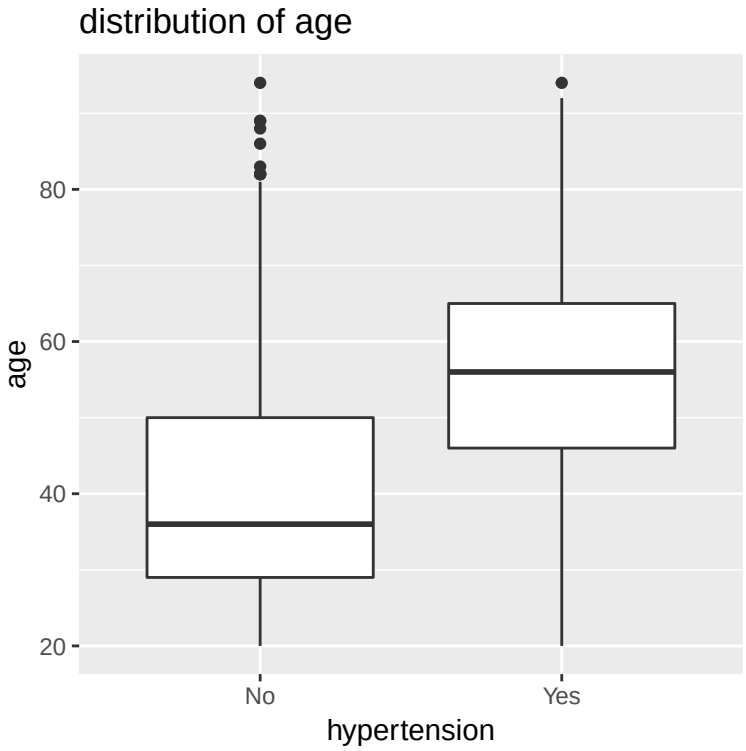
Simple data visualization will help you to make a first step judgement and provide much information. Plots indicate the trend, or pattern of the distribution of variables you interested in, and inspire you how to do the next step in data analysis. We will mainly use package `ggplot2`, a powerful tool for data visualization, and here is the link for its cheat sheet: https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf.

Plotting numerical data is something you may be familiar with. This time we are going to incorporate some

of the categorical variables into the plots. Although going from raw numerical data to categorical data bins does give you less precision, it can make drawing conclusions from plots much easier.

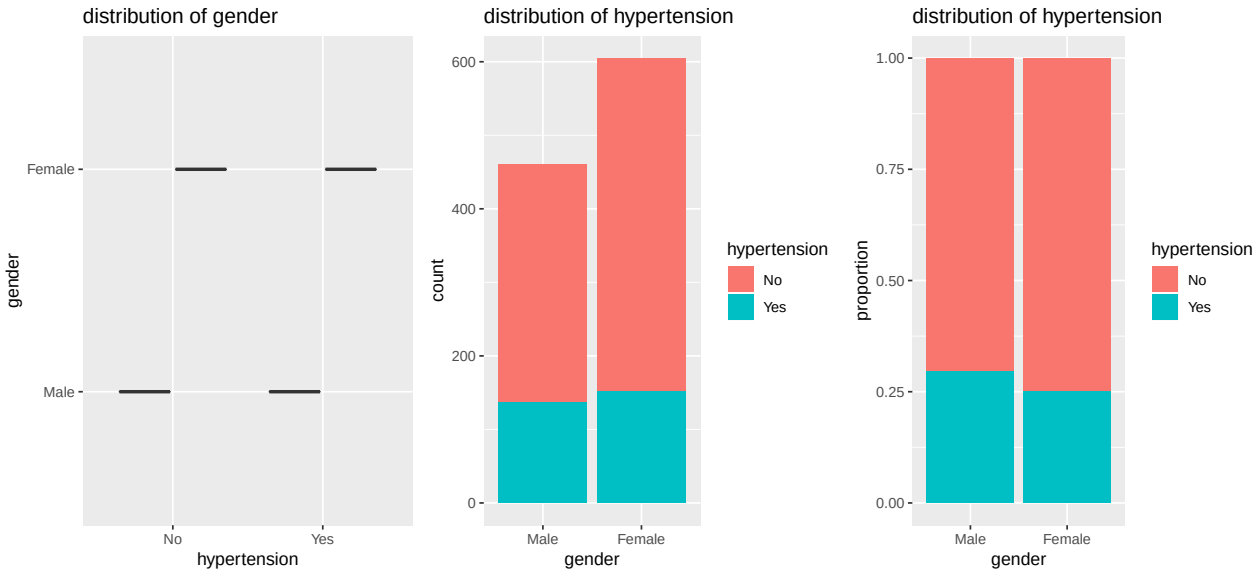
First, we try to plot one categorical variable 'hypertension', with one numerical variable 'age'.

```
p1 <- ggplot(DF, aes(x = hypertension, y=age))+
  geom_boxplot()+ggtitle('distribution of age')
p1
```



What about hypertension with gender? Now let's try three different ways to plot categorical variable 'gender'.

```
p2 <- ggplot(DF, aes(x = hypertension, y=gender))+
  geom_boxplot()+ggtitle('distribution of gender')
p3 <- ggplot(DF, aes(x = gender, fill = hypertension)) +
  geom_bar()+ggtitle('distribution of hypertension')
p4 <- ggplot(DF, aes(x = gender, fill = hypertension)) +
  geom_bar(position = "fill")+ggtitle('distribution of hypertension') + ylab('proportion')
ggarrange(p2,p3,p4,ncol=3,nrow=1)
```



The left plot uses the same way as 'age', but it fails to show the relationship! It seems not what we want so we need a correct way instead of just applying traditional plotting methods.

Then, we think about other ways, as shown in the other two plots. This time it works! We switch the x-axis and y-axis. The middle plot shows the count of 'hypertension' for male and female and you can compare the skewness of distribution. But to more clearly see the proportion of male and female for each level, we use `position='fill'`. y-axis in the right plot is proportion rather than count. It is obvious that male are more likely to get hypertension. You can try to use this method on other categorical variables.

Simple data visualization will help you to make a first step judgement and provide much information. Plots indicate the trend, or pattern of the distribution of variables you interested in, and inspire you how to do the next step in data analysis.

Survey Weighted Data

So what model can we use in our case?

- binary outcomes

You might think of logistic regression. Yes, you're pretty close already. However, think of the nature of our dataset. It's a data obtained from survey, and there is an important point to consider for analysis of survey data.

What is Survey Weighted Data

It possible and indeed often happens to a perfectly designed sampling plan ends up with too many samples in a category. For example, too many women and not enough men, or too many white and not enough other races or both. Data weighting make sense for this kind of data. If we want the data to reflect the whole population, instead of counting the weight of each data point as one, we weight the data so that the sample we have can better reflect the entire community.

What is the Weight of Data

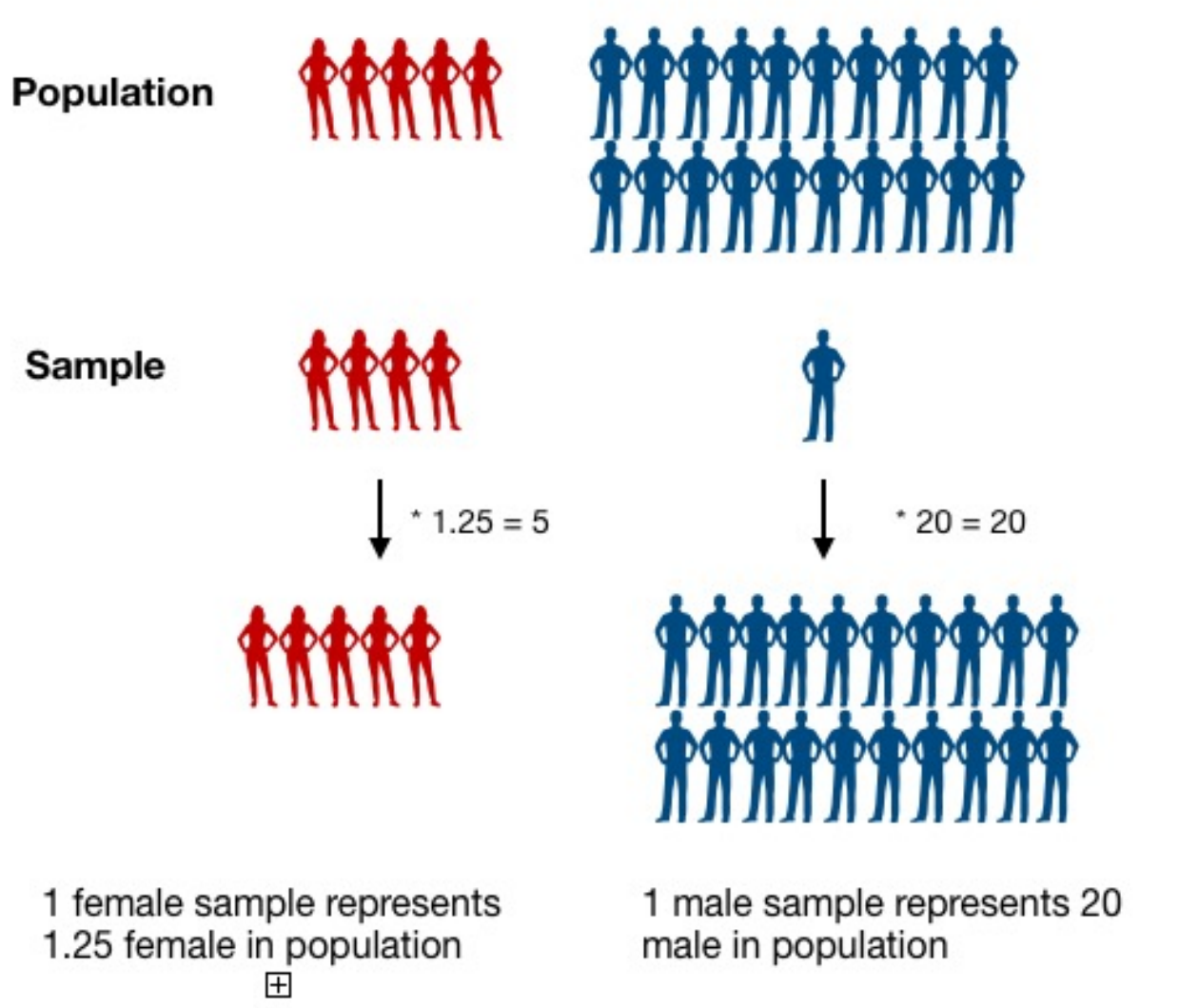
Assuming that you have 25 students (20 male and 5 female) in your class, and you want to talk with 5 of them to know their understanding of the biostatistics class. By sampling 5 students from the total 25

students, you might get 5 all female students or 4 female and 1 male in your sample. Do you expect this sample to represent the population? Definitely not since there is a higher proportion of female in sample than population. The way of calculating the weight is:

$$Weight = \frac{Proportion\ in\ population}{Proportion\ in\ sample}$$

$$Male\ Weight = \frac{20/25}{1/25} = 20$$

$$Female\ Weight = \frac{5/25}{4/25} = 1.25$$



By weighting the observation, we can make the sample better represent the population.

When we have multiple strata on the data, it might be troublesome to calculate the weight. However, for most of the survey data, the weight is calculated and included in the dataset. In our case study, the weight is calculated and we can simply apply the weight in the survey-weighted logistic regression.

The weight we use

NYC HANES 2013-2014 data are weighted in order to compensate for unequal probability of selection and 5 sets of survey weights have been constructed: CAPI weight, Physical weight, Blood Lab result weight, Urine Lab results weight and Salica Lab results weight. The determination of the most appropriate weight to use for a specific analysis depends upon the variables selected by the data analyst. When an analysis involves variables from different components of the survey, the analyst should decide whether the outcome is inclusive or exclusive, and then choose certain weights. The website (http://nychanes.org/data/) provides a guideline about how to use weight with different purposes, and you can find them through 'Analytic Guideline' and 'Other Training Materials'.

As the weight is given in the origin dataset, we use it as variable 'surveyweight' directly in our study. Here we choose CAPI weight, which should be used to analyze participants responses to all interview questions. All 1527 survey participants have a CAPI_WT. We define hypertension as the prior diagnosis of high blood pressure, so we should use the most inclusive one, CAPI weight, to get an inclusive outcome.

What is Finite Population Correction Factor

$$FPC = (\frac{N-n}{N-1})^{\frac{1}{2}}$$

- N = population size
- n = sample size

The finite population correction (fpc) is used to reduce the variance when a substantial fraction of the total population of interest has been sampled.

```
N <-  6825749
n <- 1065
((N-n)/(N-1))^0.5
```

```
## [1] 0.9999221
```

fpc of our data set is almost close to 1 and in general, you can ingore it. But if you want to get a without replacement sample, it is more appropriate to use it.

Create the Survey Weight Data

There is a function `svydesign()` in R package `survey`. The function combines a data frame and all design information needed to specify a survey design. Here is the list of options provided in this function:

- `ids`: Specify it for cluster sampling, ~0 or ~1 is a formula for no clusters. Cluster sampling is a multi-stage sampling, the total population are divided into several clusters and a simple random sample of clusters are selected. Each element in these clusters are then sampled.
- `data`: Data frame to look up variables in the formula arguments, or database table name
- `weights`: Formula or vector specifying sampling weights as an alternative to `prob`
- `fpc`: Finite population correction, `~rep(N,n)` generates a vector of length n where each entry is N (the population size). Default value is 1. The use of fpc derives a without replacement sample, otherwise with replacement sample.
- `strata`: Specify it for stratified sampling, which divides members of the population into homogeneous subgroups and then sample independently in these subpopulations. It is advantageous when subpopulations within an overall population vary.

Now we use some options to create a design relative to our dataset:

```
hypertension_design <- svydesign(
  id = ~1,
  #fpc = ~rep(N,n),
   weights = ~DF$surveyweight,
  data = DF[,-c(1,13)]
)
```

The arguments are interpreted as the following:

- `ids = ~1` means there is no clustering.
- `data = DF[,-c(1,13)]` tells `svydesign` where to find the actual data.
- `weights= ~DF$surveyweight` tells it where to find the weight.

`summary()` shows the results:

```
summary(hypertension_design)
```

```
## Independent Sampling design (with replacement)
## svydesign(id = ~1, weights = ~DF$surveyweight, data = DF[, -c(1,
##     13)])
## Probabilities:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 6.484e-05 2.089e-04 2.810e-04 2.921e-04 3.492e-04 9.705e-04
## Data variables:
##  [1] "age"          "race"         "gender"       "diet"
##  [5] "income"       "diabetes"     "bmi"          "cholesterol"
##  [9] "drink"        "smoking"      "hypertension"
```

"Independent sampling design" means our sampling design is a simple random sample. When the population size is specified (via the fpc argument) it is assumed that the SRS is without replacement. By setting other parameters you can also design differenct kinds of design, such as stratified sampling etc.

Survey Weight Data Analysis

Once we created a `survey.design` object, we can do further analysis. It is very convenient to use `svy*` functions which account for survey design features.

To calculate the mean and its standard error, use function `svymean()`. Compare the result of traditional way with the help of `mean()` and `std.error()`:

```
svymean(~bmi, hypertension_design)
```

```
##       mean     SE
## bmi 27.897 0.2188
```

```
mean(DF$bmi)
```

```
## [1] 27.7636
```

```
std.error(DF$bmi)
```

```
## [1] 0.2010068
```

It seems that there is not a significant differene between them. To calculate the confidence interval, use function `confint()` directly:

```
confint(svymean(~bmi, hypertension_design))
```

```
##        2.5 %   97.5 %
## bmi 27.46845 28.32625
```

Subgroup statistics is also easy to calculate with function `svyby()`:

```
svyby(~bmi, by=~diet, design=hypertension_design, FUN = svymean)
```

```
##                diet      bmi        se
## Poor           Poor 29.50139 1.0511178
## Fair           Fair 30.30994 0.6104274
## Good           Good 27.51282 0.3045348
## Very good Very good 26.51232 0.3048522
## Excellent Excellent 26.04319 0.6233014
```

If you are particularly interested in one group, you can use function `subset()` :

```
svymean(~bmi, subset(hypertension_design,gender=="Female"))
```

```
##       mean     SE
## bmi 28.075 0.3284
```

```
svymean(~bmi, subset(hypertension_design,gender=="Male"))
```

```
##       mean     SE
## bmi 27.703 0.2843
```

It seems that female's mean bmi and sde are a little higher.

Fit Generalized Linear Regression

Use `svyglm` in R

Logistic regression is widely used to deal with binary response variable. As we mentioned above, our dataset is survey weight dataset, therefore we need to fit the model with the survey weighted data. Just use `svyglm()` function from the `survey` package.

It's similar to fit a normal logistic regression, the only difference is that, instead of using the original data set in the `data` argument, you should input the object from the `svydesign()` to the `design` argument in the function.

```
hypertension_design$variables$hypertension <-
  ifelse(hypertension_design$variables$hypertension == 'No', yes = 0, no =1)
```

```
g1 <- svyglm(hypertension ~
    age+race+gender+diet+income+diabetes+bmi+cholesterol+drink+smoking,
    family = quasibinomial(), design = hypertension_design)
summ(g1)
```

Observations	1065
Dependent variable	hypertension
Type	Survey-weighted generalized linear model
Family	quasibinomial
Link	logit

Pseudo-R^2 (Cragg-Uhler)	0.08
Pseudo-R^2 (McFadden)	0.24
AIC	NA

	Est.	S.E.	t val.	p
(Intercept)	-4.67	0.71	-6.56	0.00
age	0.04	0.01	5.65	0.00
raceBlack/African American	0.72	0.24	3.03	0.00
raceIndian /Alaska Native	-1.49	1.07	-1.40	0.16
racePacific Islander	2.46	1.38	1.78	0.08
raceAsian	0.27	0.28	0.96	0.34
raceOther Race	0.17	0.31	0.56	0.58
genderFemale	-0.46	0.20	-2.27	0.02
dietFair	0.22	0.35	0.63	0.53
dietGood	0.22	0.34	0.63	0.53
dietVery good	-0.07	0.37	-0.19	0.85
dietExcellent	-0.04	0.45	-0.10	0.92
income$20,000 - $39,999	-0.56	0.24	-2.36	0.02
income$40,000 - $59,999	-0.84	0.35	-2.42	0.02
income$60,000 - $79,999	-0.31	0.30	-1.03	0.30
income$80,000 - $99,999	-0.78	0.53	-1.47	0.14
income$100,000 or more	-0.71	0.29	-2.40	0.02
diabetesDiabetic dx	0.94	0.26	3.59	0.00
diabetesDiabetic but no dx	0.43	0.43	1.00	0.32
bmi	0.07	0.01	4.86	0.00
cholesterolHigh value	0.86	0.19	4.52	0.00
drinkWeekly	-0.02	0.26	-0.07	0.95
drinkMonthly	-0.28	0.29	-0.97	0.33
drinkYearly	0.08	0.28	0.29	0.77
smokingFormer smoker	-0.35	0.29	-1.22	0.22
smokingCurrent smoker	-0.10	0.24	-0.40	0.69

Standard errors: Robust

Model Selection

Not all variables in g1 are significant so we would like to remove some of them to get a reduced model. We find 'race','income', 'diet','drink' and 'smoke' are not that sigifincant. Contrary to our common sense, 'diet','drink' and 'smoke' have minimum influence on hypertension. We believe this is because that their influence may be explained by other covariates, such as 'bmi'. Therefore, here we just remove 'race' and 'income'.

```
g2 <- svyglm(hypertension ~
        age+bmi+cholesterol+diabetes+gender+smoking+diet,
        family=binomial(), design = hypertension_design)
#summ(g2)$coeftable[,1:2]
```

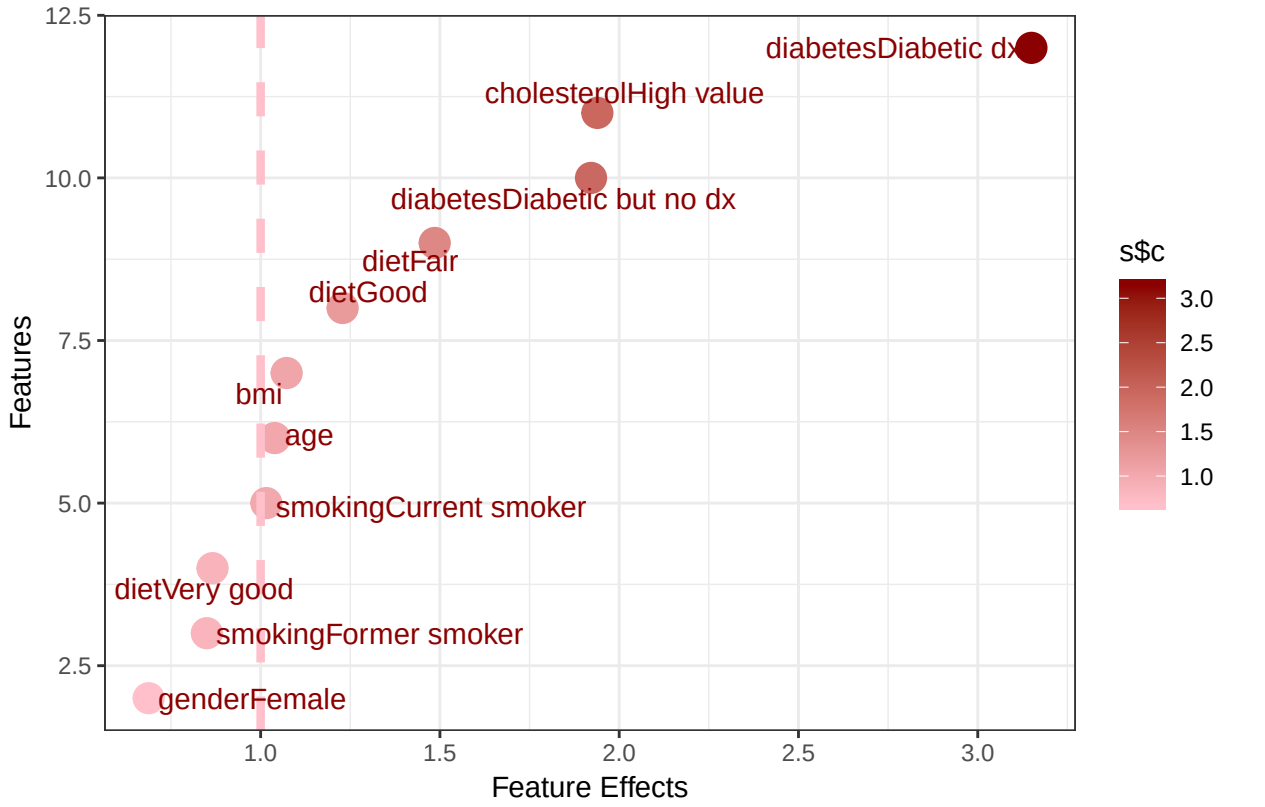
We would further study which covariate has larger influence on hypertension, thus we compare their influence and the result is shown in a plot below.

```
s <- data.frame('c'=exp(g2$coefficients[2:12]),
                'n'=names(g2$coefficients)[2:12])
```

```
s <- s[order(s$c),]
l=seq(from=2, to=12, by=1)

plt<- ggplot(s,aes(x = l, y=c,col=s$c)) +
  geom_point(size=5)+
  scale_colour_gradient(low ='pink', high = 'darkred')+
  geom_hline(yintercept = 1,col='pink',size=1.5,linetype=2)+
  theme_bw() + theme(plot.title = element_text(hjust = 2,vjust=-2))+
  geom_text_repel(aes(y = c,label=n),col='darkred')+
  ggtitle("Effect on Hypertension") +
  ylab("Feature Effects") +
  xlab("Features")+
  coord_flip()

plt
```



Here we plot the exponential form of coefficients. If the value is larger than 1, this corvariate is positive on the probability of getting hypertension, and vice versa. Besides, larger absolute value indicates larger influence on response variable.

We can conclude that people with diabetes, high value of cholesterol and fair diet have higher risk of getting hypertension. Also, among them, diabetes and cholesterol have the strongest relationship with hypertension, which is an accurate reflection of real-world situation. In all, with the help of this satisfactory regression model, research organization is able to make an initial judgement about whether the sample has high risk of hypertension once they collect information.

Compare `glm` and `svyglm`

Now, thinking about what would happen if we use general logistic regression?

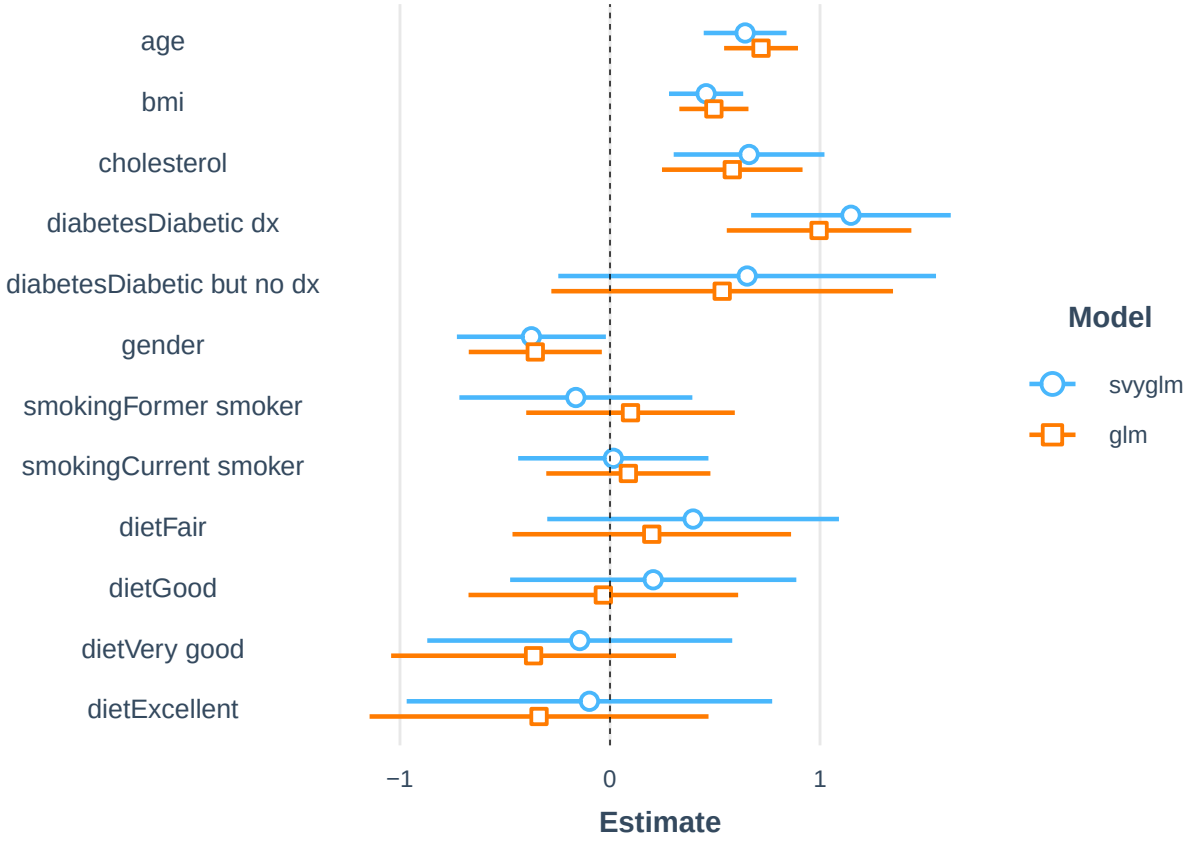
```
g3 <- glm(hypertension~ age+bmi+cholesterol+diabetes+gender+smoking+diet,
          family = binomial(link = 'logit'),data = DF)
#summ(g3)$coeftable[,1:2]
```

Function `plot_summs`

Function `plot_summs` is provided by library `jtools`, which plots regression coefficients and their uncertainty in a visually appealing way. The most wonderful thing is that you can do the same procedure for multiple models simultaneously, and compare directly in one plot. We would like to introduce some basic options here:

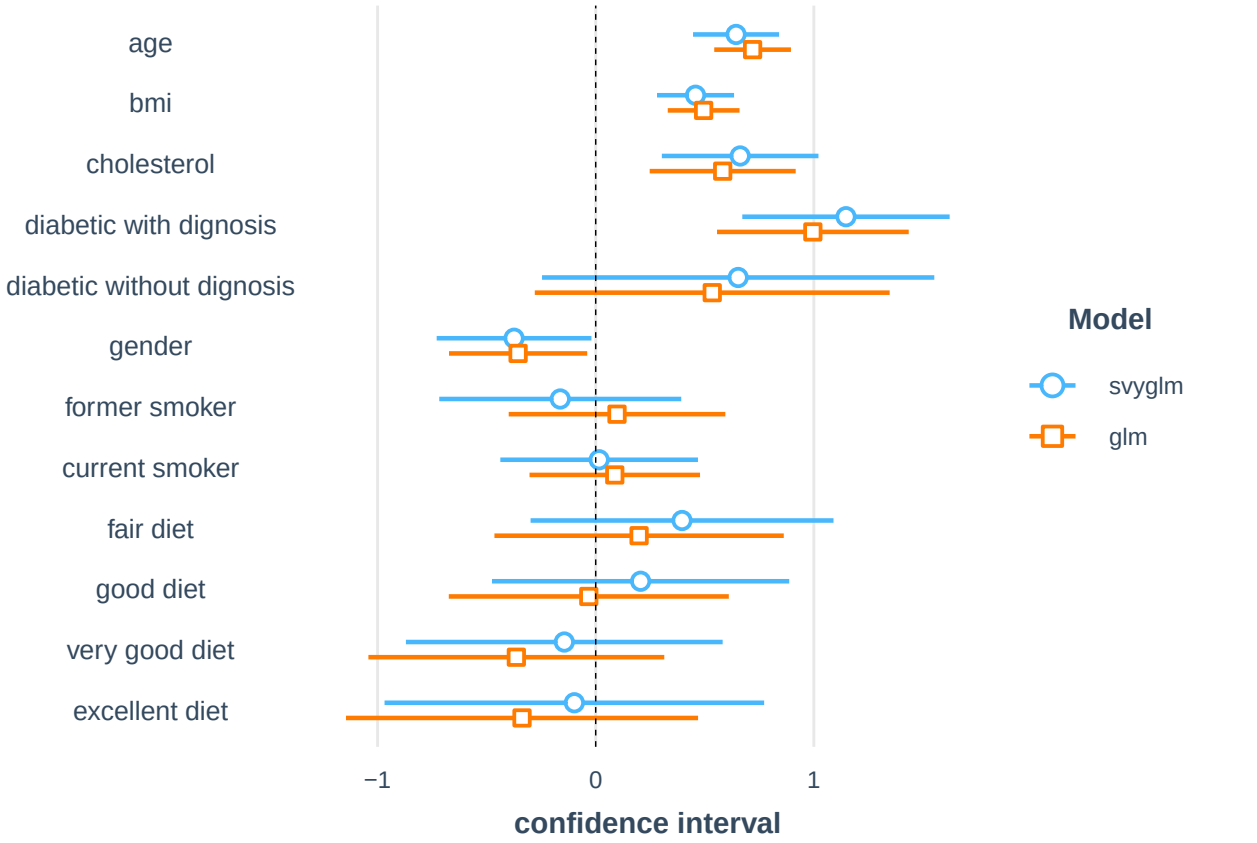
- `...` : regression model(s)
- `coef` : If you'd like to include only certain coefficients, provide them as a vector. If it is a named vector, then the names will be used in place of the variable names. Default: NULL
- `model.names` : If plotting multiple models simultaneously, you can provide a vector of names here. Default: NULL
- `ci_level`: The desired width of confidence intervals for the coefficients. Default: 0.95
- `scale`: If we set it as TRUE, it will make all variable in the same scale, making it pretty easy to make a quick assessment

```
plot_summs(g2, g3,scale = TRUE, model.names = c("svyglm", "glm"))
```



The label names are default setting, but to make it more readable, we prefer to specify the label name via `coefs=`.

```
plot_summs(g2,g3,
           coefs = c('age'='age','bmi'='bmi', 'cholesterol'='cholesterol',
                     'diabetic with dignosis'='diabetesDiabetic dx',
                     'diabetic without dignosis'='diabetesDiabetic but no dx',
                     'gender'='gender','former smoker'='smokingFormer smoker',
                     'current smoker'='smokingCurrent smoker',
                     'fair diet'='dietFair','good diet'='dietGood',
                     'very good diet'='dietVery good','excellent diet'='dietExcellent'),
           model.names = c("svyglm", "glm"), scale=TRUE)+ xlab('confidence interval')
```




```
ggsave("data/Finalplot.png", plot = last_plot(), device = "png")
```

We can see that 'cholesterol', 'diabetes', 'smoke' and 'drink' have larger difference in confidence interval plot while 'age', 'bmi' and 'gender' have slight difference.

It is strange that the confidence interval computed by `svyglm` is larger than that computed by `glm`. One possible reason behind this is `survey` computes the standard errors with consideration of the loss of precision introduced by sampling weights. Weights in `glm` simply adjust the weight given to the errors in the least square estimation, so the standard errors aren't correct.