An exercise on survival analysis and multiple imputation techniques using stroke data

Lui Yiu Wa

2025-11-04

Data source and goal

Source of data

https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data

Provided by user Fedesoriano

Introduction

The primary goal of this project was a pedagogical exercise to implement and analyze a series of survival analysis and missing-data handling techniques. The main focus of this project is on the techniques themselves rather than establishing a definitive scientific conclusion on the risk of getting a stroke even though it will be used as a heuristic motivation to guide this project. Another objective of this project is to recognize the limitations of this data set and why it is not ideal for a survival analysis.

The data set used in this analysis contains multiple attributes including the variables about their health as well as the status of their occupation and personal life. There are missing values on smoking status and BMI of the respondents. It provides us an opportunity to apply multiple imputation technique and comparing it to complete case analysis.

I will take the age at which a stroke could occur as the response variable that we would like to learn more about or predict using the covariates available.

Goal of the analysis:

1. Use non-parametric methods to visualize the empirical survival function
2. Use semi-parametric methods to investigate what variables significantly increase the risk of getting a stroke
3. Handling sparse events
4. Investigate whether the common proportional hazard assumption holds. If not, what could the remedy be?
5. Compare analysis with multiple imputation data set vs complete case analysis
6. Recognize the limitation of our analysis
7. Have fun!

Limitation on our data set

The dataset that I will be using for this project provides the age at event or censoring, but not the age at study entry. The sampling method was not specified by the author, and without further information, the best available course is to assume every subject entered the study at birth. However, this assumption has a significant chance of being wrong, as a study lasting over 90 years is unlikely, and it would introduce a bias, specifically, an underestimation of risk. Ideally, we could account for this left truncation with entry age information, this means we will only include a subject in our risk set when they have entered our study. But for now, we will move forward while keeping this limitation in mind.

Initial data assessment and data cleaning

First, we need to load the libraries that we will need for this analysis

```r
library(readr)
library(survival)
library(tidyverse)
library(mice)
library(MASS)
library(AMR)
library(car)
```

Load the data set used in this analysis

```r
data <- read_csv("archive/healthcare-dataset-stroke-data.csv")
```

```
## Rows: 5110 Columns: 12
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (6): gender, ever_married, work_type, Residence_type, bmi, smoking_status
## dbl (6): id, age, hypertension, heart_disease, avg_glucose_level, stroke
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

We can first have an overview of the data itself

```r
print(data, width = Inf)
```

```
## # A tibble: 5,110 x 12
##       id gender   age hypertension heart_disease ever_married work_type
##    <dbl> <chr>  <dbl>        <dbl>         <dbl> <chr>        <chr>
##  1  9046 Male      67            0             1 Yes          Private
##  2 51676 Female    61            0             0 Yes          Self-employed
##  3 31112 Male      80            0             1 Yes          Private
##  4 60182 Female    49            0             0 Yes          Private
##  5  1665 Female    79            1             0 Yes          Self-employed
##  6 56669 Male      81            0             0 Yes          Private
##  7 53882 Male      74            1             1 Yes          Private
##  8 10434 Female    69            0             0 No           Private
##  9 27419 Female    59            0             0 Yes          Private
## 10 60491 Female    78            0             0 Yes          Private
##    Residence_type avg_glucose_level bmi   smoking_status  stroke
##    <chr>                      <dbl> <chr> <chr>            <dbl>
##  1 Urban                      229.  36.6  formerly smoked      1
##  2 Rural                      202.  N/A   never smoked         1
##  3 Rural                      106.  32.5  never smoked         1
##  4 Urban                      171.  34.4  smokes               1
##  5 Rural                      174.  24    never smoked         1
##  6 Urban                      186.  29    formerly smoked      1
##  7 Rural                       70.1 27.4  never smoked         1
##  8 Urban                       94.4 22.8  never smoked         1
##  9 Rural                       76.2 N/A   Unknown              1
## 10 Urban                       58.6 24.2  Unknown              1
## # i 5,100 more rows
```

```
summary(data)
```

```
##        id           gender               age         hypertension
##  Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17741   Class  :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36932   Mode   :character   Median :45.00   Median :0.00000
##  Mean   :36518                      Mean   :43.23   Mean   :0.09746
##  3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                      Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married       work_type          Residence_type
##  Min.   :0.00000   Length:5110        Length:5110        Length:5110
##  1st Qu.:0.00000   Class  :character  Class  :character  Class  :character
##  Median :0.00000   Mode   :character  Mode   :character  Mode   :character
##  Mean   :0.05401
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level     bmi            smoking_status          stroke
##  Min.   : 55.12    Length:5110        Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    Class  :character  Class  :character  1st Qu.:0.00000
##  Median : 91.89    Mode   :character  Mode   :character  Median :0.00000
##  Mean   :106.15                                          Mean   :0.04873
##  3rd Qu.:114.09                                          3rd Qu.:0.00000
##  Max.   :271.74                                          Max.   :1.00000
```

We see some variables are not using the data types that we are expecting

Let's quickly convert them to the ones that we need

```
data_clean <- data %>% mutate(
  id = as.character(id),
  age = as.numeric(age),
  avg_glucose_level = as.numeric(avg_glucose_level),
  bmi = as.numeric(bmi),
  across(c(gender, hypertension, heart_disease, ever_married,
           work_type, Residence_type, smoking_status), as.factor)

)
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘bmi = as.numeric(bmi)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
print(data, n = 5, width = Inf)
```

```
## # A tibble: 5,110 x 12
##      id gender   age hypertension heart_disease ever_married work_type
##   <dbl> <chr>  <dbl>        <dbl>         <dbl> <chr>        <chr>
## 1  9046 Male      67            0             1 Yes          Private
## 2 51676 Female    61            0             0 Yes          Self-employed
## 3 31112 Male      80            0             1 Yes          Private
## 4 60182 Female    49            0             0 Yes          Private
## 5  1665 Female    79            1             0 Yes          Self-employed
##   Residence_type avg_glucose_level bmi   smoking_status  stroke
##   <chr>                      <dbl> <chr> <chr>            <dbl>
## 1 Urban                       229. 36.6  formerly smoked      1
## 2 Rural                       202. N/A   never smoked         1
## 3 Rural                       106. 32.5  never smoked         1
## 4 Urban                       171. 34.4  smokes               1
## 5 Rural                       174. 24    never smoked         1
## # i 5,105 more rows
```

We can also see the tables for all the factor variables

```r
table(data_clean$gender)
```

```
## 
## Female   Male  Other 
##   2994   2115      1
```

```r
table(data_clean$hypertension, useNA = "always")
```

```
## 
##    0    1 <NA> 
## 4612  498    0
```

```r
table(data_clean$heart_disease, useNA = "always")
```

```
## 
##    0    1 <NA> 
## 4834  276    0
```

```r
table(data_clean$ever_married, useNA = "always")
```

```
## 
##   No  Yes <NA> 
## 1757 3353    0
```

```
table(data_clean$work_type, useNA = "always")
```

```
##
##      children      Govt_job  Never_worked       Private Self-employed
##           687           657            22          2925           819
##          <NA>
##             0
```

```
table(data_clean$Residence_type, useNA = "always")
```

```
##
## Rural Urban  <NA>
##  2514  2596     0
```

```
table(data_clean$smoking_status, useNA = "always")
```

```
##
## formerly smoked    never smoked          smokes         Unknown            <NA>
##             885            1892             789            1544               0
```

```
table(data_clean$stroke, useNA = "always")
```

```
##
##    0    1 <NA>
## 4861  249    0
```

No NA values appear.

But we notice gender == “Other” has only 1 observation, we should disregard that observation because it will likely produce unstable result in our analysis

Challenge: Checking for NA values, sparse events and converting implicitly missing data into explicit NA values

```
data_clean <- data_clean %>% filter(gender != "Other")
data_clean <- data_clean %>% droplevels.data.frame()
```

And we also notice for smoking status, we have an “Unknown” level, which we can treat as having NA values

```
data_clean <- data_clean %>% mutate(
  smoking_status = ifelse(smoking_status == "Unknown", NA,
                          as.character(smoking_status)) %>% as.factor()
)
```

We should also check for missing values in numeric variables

```
sapply(data_clean %>% dplyr::select(age, avg_glucose_level, bmi), function(x) sum(is.na(x)))
```

```
##               age avg_glucose_level               bmi
##                 0                 0               201
```

We have 201 missing values in bmi

Handling missing values

Now that we have noticed that there are a good amount of observations that of NA values. We have 2 common ways to deal with such observations. We can either do a multiple imputation or we can simply drop the observations with NA values, a so called complete case analysis.

Or we can do both and compare the results. Such practice is called a sensitivity analysis.

We can start with either one but I opted to do my initial analysis with the multiple imputation approach.

Imputing the missing values

We may use the MICE library to help us do the imputation. Note that for imputation to be valid, we assume the data to be missing at random (MAR). Which means the probability of a data point missing is dependent on the observed data, but not the missing data. If the data is not missing at random (MNAR) meaning the probability of a data point missing also depends on the missing data, then this problem becomes exceptionally difficult to handle, at least, too difficult for me to handle. But let's just assume MAR is what we are dealing with for now.

I decided to use a simple default chain with 5 chains, 50 iterations each and used predictive mean matching. This will work well enough for a simple data set like ours.

Let's see if our imputation has worked properly

```
sum(is.na(imputed_complete_data$smoking_status))
```

```
## [1] 0
```

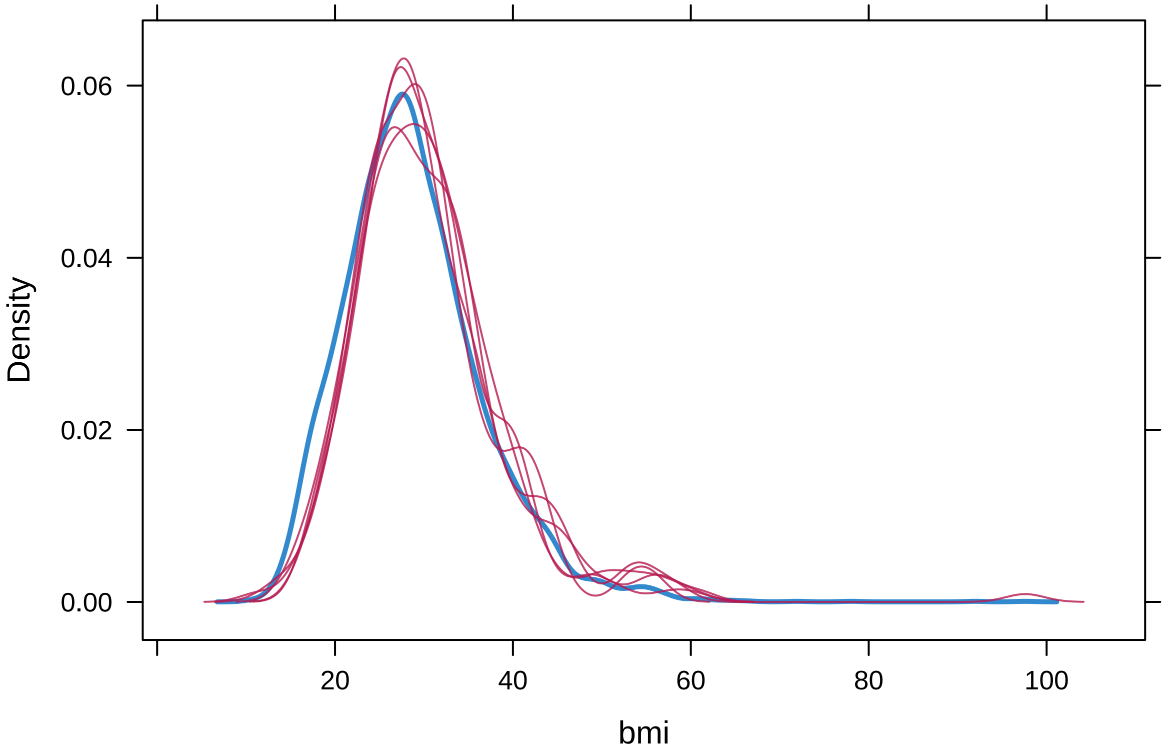
```
sum(is.na(imputed_complete_data$bmi))
```

```
## [1] 0
```

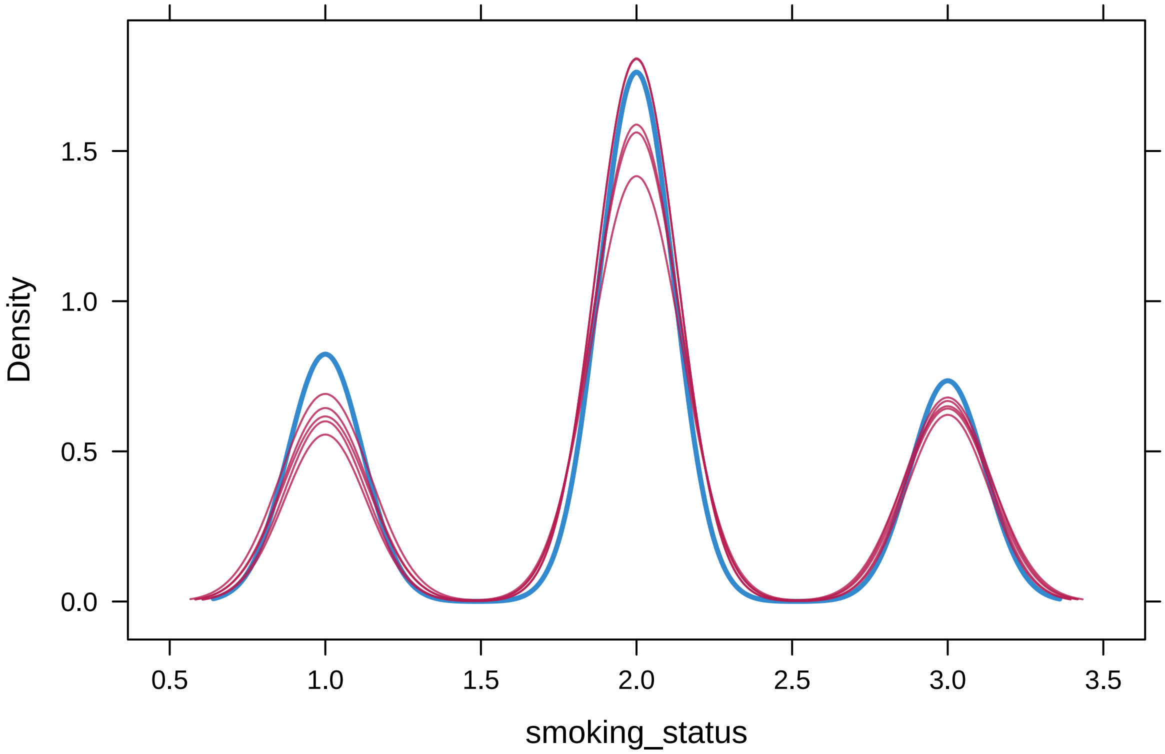
MICE diagnostics

We need to see if the imputed data make sense. Ideally, the imputed data should have similiar distribution as the observed data.

```
mice::densityplot(imputed_data, ~bmi)
```



```
mice::densityplot(imputed_data, ~smoking_status)
```



It seems that the distribution of the imputed data and observed data are quite similar, we can be confident that our imputed data are reasonable.

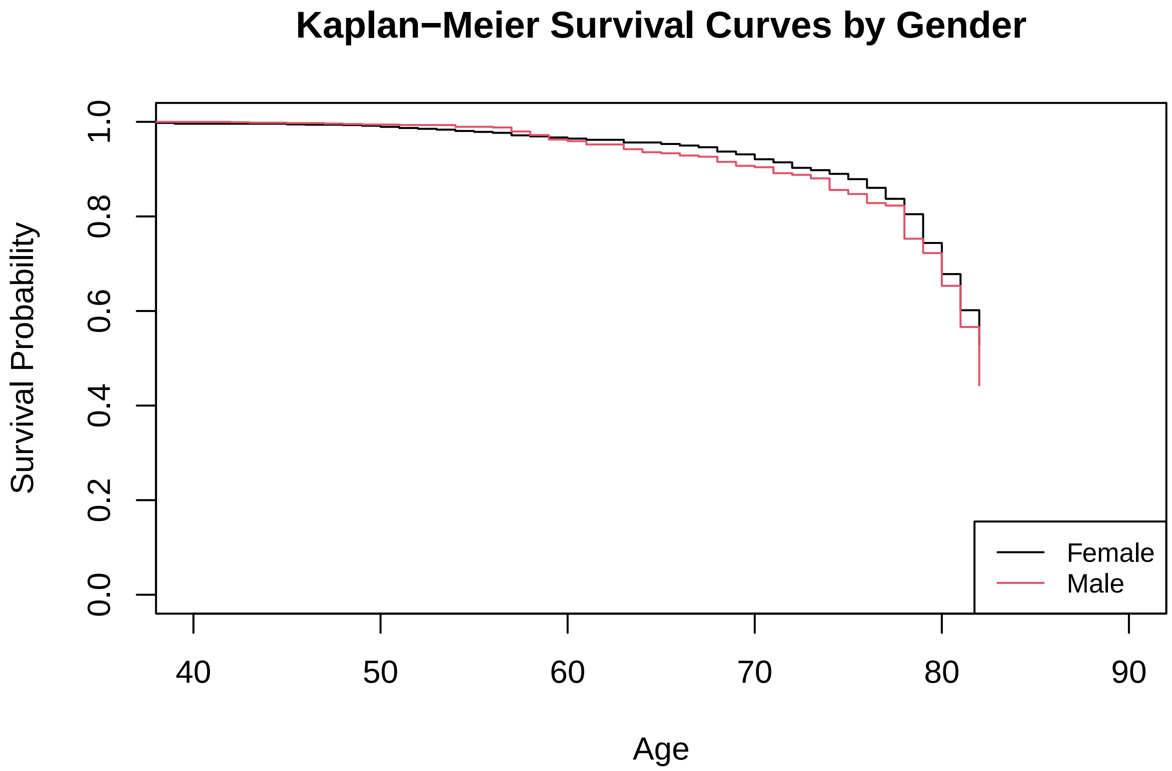
Initial exploration and visualization

We can first create a survival object

```
stroke.surv <- Surv(imputed_complete_data$age, imputed_complete_data$stroke)
```

We can visualize the empirical survival function with Kaplan-Meier curve, and one thing we can do is to stratify our data into groups and show their respective survival curves. Take gender for example

```
km_fit_gender <- survfit(stroke.surv ~ gender, data = imputed_complete_data)
plot(km_fit_gender, col = 1:2, xlab = "Age", ylab = "Survival Probability",
     main = "Kaplan-Meier Survival Curves by Gender", xlim = c(40,90))
legend("bottomright", legend = levels(imputed_complete_data$gender),
       col = 1:2, lty = 1, cex = 0.8)
```



It doesn't seem like there is that big of a difference, but remember it is only a visual test.

We can also do the same for smoking status. However, note that we have imputed smoking status, and we have produced 5 data sets with different imputed values, we can either plot the curves 5 times or just arbitrarily pick one data set and plot the curve that corresponds to it. Neither of these are very helpful. It is better for us to use other techniques.

Initial assessment the effect of smoking and BMI on the risk of getting a stroke

To see if smoking status has an effect on the survival function, we can investigate how smoking affects the risk of one getting a stroke. One popular way to do it is by the Cox Proportional Hazard model.

Again we have 5 data sets so we need to fit the model 5 times. Then we can average the estimates and compute the total variance which is contributed by the inherent randomness of the data as well as the different imputed values between data sets using Rubin's rules.

It sounds tedious but fortunate for us, some incredibly intelligent people have already done the hard work for us, all we need is to call some functions

```
imputed_smokes_ph <- with(imputed_data, coxph(stroke.surv ~ smoking_status))
pooled_smokes_ph <- pool(imputed_smokes_ph)
summary(pooled_smokes_ph)
```

```
##                          term   estimate std.error statistic        df   p.value
## 1 smoking_statusnever smoked -0.1763112 0.1541691 -1.143622  98.91239 0.2555407
## 2       smoking_statussmokes  0.1882760 0.1861524  1.011408 161.25908 0.3133364
```

Oof, it seems like our p-value is not great. But it is not the end of the world. Let's take a look at the point estimate.

Remember, the base level is formerly smoked, so we are comparing the relative risk of never_smoked (or smokes) to formerly_smoked. And the result is simply e^(estimate).

So people who never smoked have roughly e^(-0.176) (or 0.839) the risk of people who formerly smoked. And people who smokes have roughly e^(0.188) (or 1.21) the risk.

Seems reasonable. And it also aligns with the general consensus that smoking increases your risk of having a stroke.

Even though the p value is not amazing, I would argue it might still be worth it to keep smoking status as a variable in our model for now.

We can also do the same for bmi

```
imputed_bmi_ph <- with(imputed_data, coxph(stroke.surv ~ bmi))
pooled_bmi_ph <- pool(imputed_bmi_ph)
summary(pooled_bmi_ph)
```

```
##   term   estimate  std.error statistic       df     p.value
## 1  bmi 0.03276682 0.01038324  3.155741 162.4569 0.001908171
```

This time, it is pretty clear cut that bmi is a significant variable, we can feel pretty confident to proceed with our modeling.

Further investigation on other covariates.

Let's now try to fit a multivariate model. For practicality, I am inclined to use only one instance of the imputation, we have to keep in mind that our estimation will likely be overconfident.

Note that using all the covariates in our model and see what are significant is NOT a good practice in general. It is called a kitchen sink model. Ideally, we should have some specific hypothesis we wish to test before running the analysis. But here, I am just trying to see if anything "funky" is going on so I can further clean up the data. The goal here is not to make any inference.

```
imputed_full_ph <- coxph(stroke.surv ~ . - id - age - stroke, data = imputed_complete_data)
summary(imputed_full_ph)
```

```
## Call:
## coxph(formula = stroke.surv ~ . - id - age - stroke, data = imputed_complete_data)
##
##   n= 5109, number of events= 249
##
##                                  coef  exp(coef)   se(coef)      z Pr(>|z|)
## genderMale                  8.291e-02  1.086e+00  1.325e-01  0.626   0.5316
## hypertension1               1.790e-01  1.196e+00  1.471e-01  1.217   0.2238
## heart_disease1             -7.102e-02  9.314e-01  1.676e-01 -0.424   0.6718
## ever_marriedYes            -1.454e-01  8.647e-01  2.057e-01 -0.707   0.4797
## work_typeGovt_job          -1.857e+01  8.644e-09  1.157e+03 -0.016   0.9872
## work_typeNever_worked      -1.845e+01  9.683e-09  1.722e+04 -0.001   0.9991
## work_typePrivate           -1.856e+01  8.690e-09  1.157e+03 -0.016   0.9872
## work_typeSelf-employed     -1.914e+01  4.857e-09  1.157e+03 -0.017   0.9868
## Residence_typeUrban        -3.127e-02  9.692e-01  1.285e-01 -0.243   0.8077
## avg_glucose_level           2.214e-03  1.002e+00  1.080e-03  2.050   0.0404 *
## bmi                         2.465e-02  1.025e+00  1.068e-02  2.308   0.0210 *
## smoking_statusnever smoked -1.725e-01  8.416e-01  1.445e-01 -1.194   0.2325
## smoking_statussmokes        1.685e-01  1.183e+00  1.823e-01  0.924   0.3555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                            exp(coef) exp(-coef) lower .95 upper .95
## genderMale                 1.086e+00  9.204e-01    0.8379     1.409
## hypertension1              1.196e+00  8.361e-01    0.8964     1.596
## heart_disease1             9.314e-01  1.074e+00    0.6706     1.294
## ever_marriedYes            8.647e-01  1.156e+00    0.5778     1.294
## work_typeGovt_job          8.644e-09  1.157e+08    0.0000       Inf
## work_typeNever_worked      9.683e-09  1.033e+08    0.0000       Inf
## work_typePrivate           8.690e-09  1.151e+08    0.0000       Inf
## work_typeSelf-employed     4.857e-09  2.059e+08    0.0000       Inf
## Residence_typeUrban        9.692e-01  1.032e+00    0.7535     1.247
## avg_glucose_level          1.002e+00  9.978e-01    1.0001     1.004
## bmi                        1.025e+00  9.757e-01    1.0037     1.047
## smoking_statusnever smoked 8.416e-01  1.188e+00    0.6341     1.117
## smoking_statussmokes       1.183e+00  8.450e-01    0.8279     1.692
##
## Concordance= 0.634  (se = 0.024 )
## Likelihood ratio test= 50.23  on 13 df,   p=3e-06
## Wald test            = 21.69  on 13 df,   p=0.06
## Score (logrank) test = 66.34  on 13 df,   p=4e-09
```

We see the work_type is behaving very strangely, let's investigate further!

Further data cleaning

```
table(imputed_complete_data$work_type, imputed_complete_data$stroke)
```

```
##
##                   0    1
##   children      685    2
##   Govt_job      624   33
##   Never_worked   22    0
##   Private      2775  149
##   Self-employed  754   65
```

Ha, we see that our base case children has very low risk of getting a stroke (surprise to no one), so the relative risk becomes very unstable. Also, work_type == "Never_worked" has no observation that has ever had a stroke. That can also lead to unstable results.

So, it is apparent that cox_ph may not handle this variable well without doing a lot of pre-processing, but we also want to see if work_type has a significant effect on stroke risk.

In this case, I will simply use the likelihood test on the contingency table. But, there are cells that have very little observation, we need to be careful with the results

```
work_type_stroke_table <- table(imputed_complete_data$work_type, imputed_complete_data$stroke)
g.test(work_type_stroke_table)
```

```
## Warning in g.test(work_type_stroke_table): G-statistic approximation may be
## incorrect due to E < 5
```

```
##
##  G-test of independence
##
## data:  work_type_stroke_table
## X-squared = 69.761, p-value = 2.55e-14
```

Alternatively, we can artificially conflate the observation of each cells by 5, 10 and 15. Or we could use a Fischer's exact test.

```
g.test(work_type_stroke_table + 5)
```

```
## Warning in g.test(work_type_stroke_table + 5): G-statistic approximation may be
## incorrect due to E < 5
```

```
##
##  G-test of independence
##
## data:  work_type_stroke_table + 5
## X-squared = 56.681, p-value = 1.443e-11
```

```
g.test(work_type_stroke_table + 10)
```

```
## Warning in g.test(work_type_stroke_table + 10): G-statistic approximation may
## be incorrect due to E < 5
```

```
##
##  G-test of independence
##
## data:  work_type_stroke_table + 10
## X-squared = 58.74, p-value = 5.335e-12
```

```
g.test(work_type_stroke_table + 15)
```

```
## Warning in g.test(work_type_stroke_table + 15): G-statistic approximation may
## be incorrect due to E < 5
```

```
##
##  G-test of independence
##
## data:  work_type_stroke_table + 15
## X-squared = 65.106, p-value = 2.444e-13
```

It looks like no matter what we get a significant result, that is to say, work type probably has a significant effect on the risk of getting a stroke.

We can try to use a different base level for our regression.

```
imputed_complete_data$work_type <- relevel(imputed_complete_data$work_type, ref = "Govt_job")
imputed_full_ph <- coxph(stroke.surv ~ . - id - age - stroke,
                         data = imputed_complete_data)
```

```
## Warning in coxph.fit(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 5 ; coefficient may be infinite.
```

```
summary(imputed_full_ph)
```

```
## Call:
## coxph(formula = stroke.surv ~ . - id - age - stroke, data = imputed_complete_data)
##
##   n= 5109, number of events= 249
##
##                                  coef  exp(coef)   se(coef)      z Pr(>|z|)
## genderMale                  8.291e-02  1.086e+00  1.325e-01  0.626  0.53164
## hypertension1               1.790e-01  1.196e+00  1.471e-01  1.217  0.22375
## heart_disease1             -7.102e-02  9.314e-01  1.676e-01 -0.424  0.67184
## ever_marriedYes            -1.454e-01  8.647e-01  2.057e-01 -0.707  0.47970
## work_typechildren           1.857e+01  1.157e+08  1.157e+03  0.016  0.98719
## work_typeNever_worked       1.136e-01  1.120e+00  1.725e+04  0.000  0.99999
## work_typePrivate            5.343e-03  1.005e+00  1.933e-01  0.028  0.97795
## work_typeSelf-employed     -5.763e-01  5.620e-01  2.164e-01 -2.664  0.00773 **
## Residence_typeUrban        -3.127e-02  9.692e-01  1.285e-01 -0.243  0.80772
## avg_glucose_level           2.214e-03  1.002e+00  1.080e-03  2.050  0.04040 *
## bmi                         2.465e-02  1.025e+00  1.068e-02  2.308  0.02100 *
## smoking_statusnever smoked -1.725e-01  8.416e-01  1.445e-01 -1.194  0.23254
## smoking_statussmokes        1.685e-01  1.183e+00  1.823e-01  0.924  0.35553
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##                            exp(coef) exp(-coef) lower .95 upper .95
## genderMale                 1.086e+00  9.204e-01    0.8379    1.4087
## hypertension1              1.196e+00  8.361e-01    0.8964    1.5957
## heart_disease1             9.314e-01  1.074e+00    0.6706    1.2938
## ever_marriedYes            8.647e-01  1.156e+00    0.5778    1.2940
## work_typechildren          1.157e+08  8.644e-09    0.0000       Inf
## work_typeNever_worked      1.120e+00  8.926e-01    0.0000       Inf
## work_typePrivate           1.005e+00  9.947e-01    0.6883    1.4684
## work_typeSelf-employed     5.620e-01  1.779e+00    0.3677    0.8588
## Residence_typeUrban        9.692e-01  1.032e+00    0.7535    1.2467
## avg_glucose_level          1.002e+00  9.978e-01    1.0001    1.0043
## bmi                        1.025e+00  9.757e-01    1.0037    1.0466
## smoking_statusnever smoked 8.416e-01  1.188e+00    0.6341    1.1170
## smoking_statussmokes       1.183e+00  8.450e-01    0.8279    1.6919
##
## Concordance= 0.634  (se = 0.024 )
## Likelihood ratio test= 50.23  on 13 df,   p=3e-06
## Wald test            = 21.69  on 13 df,   p=0.06
## Score (logrank) test = 66.34  on 13 df,   p=4e-09
```

This time we can capture that there is a significant difference between Self_employed and Govt_job.

I have one more idea, and that is to regroup them and see if there is a difference

```
imputed_complete_data_collapse <- imputed_complete_data %>% mutate(
 work_type = fct_collapse(work_type,
   Govt_job = "Govt_job",
    'Self-employed' = "Self-employed",
   Other = c("children", "Never_worked", "Private")
  )
)
```

```
imputed_collapse_full_ph <- coxph(stroke.surv ~ . - id - age - stroke, data = imputed_complete_data_col
summary(imputed_collapse_full_ph)
```

```
## Call:
## coxph(formula = stroke.surv ~ . - id - age - stroke, data = imputed_complete_data_collapse)
##
##   n= 5109, number of events= 249
##
##                                 coef exp(coef)  se(coef)      z Pr(>|z|)
## genderMale                  0.086694  1.090563  0.132496  0.654   0.5129
## hypertension1               0.176302  1.192798  0.146922  1.200   0.2301
## heart_disease1             -0.076983  0.925905  0.167570 -0.459   0.6459
## ever_marriedYes            -0.202380  0.816785  0.199619 -1.014   0.3107
## work_typeOther              0.017914  1.018076  0.193046  0.093   0.9261
## work_typeSelf-employed     -0.573896  0.563326  0.216397 -2.652   0.0080 **
## Residence_typeUrban        -0.032463  0.968059  0.128468 -0.253   0.8005
## avg_glucose_level           0.002241  1.002243  0.001079  2.078   0.0377 *
## bmi                         0.022860  1.023123  0.010665  2.143   0.0321 *
## smoking_statusnever smoked -0.169992  0.843671  0.144430 -1.177   0.2392
## smoking_statussmokes        0.166997  1.181750  0.182318  0.916   0.3597
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##                            exp(coef) exp(-coef) lower .95 upper .95
## genderMale                    1.0906     0.9170    0.8411    1.4139
## hypertension1                 1.1928     0.8384    0.8943    1.5908
## heart_disease1                0.9259     1.0800    0.6667    1.2859
## ever_marriedYes               0.8168     1.2243    0.5523    1.2079
## work_typeOther                1.0181     0.9822    0.6974    1.4863
## work_typeSelf-employed        0.5633     1.7752    0.3686    0.8609
## Residence_typeUrban           0.9681     1.0330    0.7526    1.2452
## avg_glucose_level             1.0022     0.9978    1.0001    1.0044
## bmi                           1.0231     0.9774    1.0020    1.0447
## smoking_statusnever smoked    0.8437     1.1853    0.6357    1.1197
## smoking_statussmokes          1.1818     0.8462    0.8267    1.6893
##
## Concordance= 0.62  (se = 0.022 )
## Likelihood ratio test= 37.21  on 11 df,   p=1e-04
## Wald test            = 36.56  on 11 df,   p=1e-04
## Score (logrank) test = 37.2  on 11 df,   p=1e-04
```

This avoid some categories from having 0 or very sparse stroke events, and any models that uses work_type as a covariate should behave better now.

Checking collinearity.

Now we also need to check for collinearity, we can do so by running a linear regression on a made up response with the covariates, and calculate the variance inflation factor.

```
lm_for_vif <- lm(rnorm(n = nrow(imputed_complete_data_collapse)) ~ . - id - age - stroke,
                 data = imputed_complete_data_collapse)
vif(lm_for_vif)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## gender            1.021420  1        1.010653
## hypertension      1.078263  1        1.038395
## heart_disease     1.054608  1        1.026941
## ever_married      1.229029  1        1.108616
## work_type         1.087891  2        1.021284
## Residence_type    1.000636  1        1.000318
## avg_glucose_level 1.086364  1        1.042288
## bmi               1.166712  1        1.080145
## smoking_status    1.024625  2        1.006100
```

VIF looks good as they are all below the common cut off of 5, collinearity should not be a concern. We can proceed.

Model selection

Now we want to select variables that best explain the response. And there are some modern methods such as LASSO, RIDGE, principle component analysis and data scientists' favorite: Cross validation.

These are all fantastic choice they are a bit cumbersome to implement. Because model selection is not the main focus of this project I will opt for AIC/BIC stepwise selection for an easier and quicker implementation. But note that these 2 are almost always inferior choices as they don't deal with collinearity as well, and less stable variable selection. Between the 2, AIC is better than BIC for prediction as it has a more lenient penalty term for extra variables. Where as BIC is better for identifying the true model given that the true model is in our set of candidate models. And they are asymptotically equivalent to cross validation with different values of k.

AIC model:

```
aic_model <- step(imputed_collapse_full_ph, k = 2, direction = "both")
```

```
summary(aic_model)
```

```
## Call:
## coxph(formula = stroke.surv ~ work_type + avg_glucose_level +
##     bmi, data = imputed_complete_data_collapse)
```

```
## 
##   n= 5109, number of events= 249 
## 
##                              coef exp(coef)  se(coef)      z Pr(>|z|)  
## work_typeOther           0.038575  1.039329  0.192695  0.200   0.8413  
## work_typeSelf-employed -0.547063  0.578647  0.215799 -2.535   0.0112 *
## avg_glucose_level        0.002388  1.002391  0.001047  2.280   0.0226 *
## bmi                      0.022248  1.022497  0.010515  2.116   0.0344 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
##                        exp(coef) exp(-coef) lower .95 upper .95
## work_typeOther            1.0393     0.9622    0.7124    1.5163
## work_typeSelf-employed    0.5786     1.7282    0.3791    0.8833
## avg_glucose_level         1.0024     0.9976    1.0003    1.0045
## bmi                       1.0225     0.9780    1.0016    1.0438
## 
## Concordance= 0.592  (se = 0.023 )
## Likelihood ratio test= 30.66  on 4 df,   p=4e-06
## Wald test            = 30.05  on 4 df,   p=5e-06
## Score (logrank) test = 30.6  on 4 df,   p=4e-06
```

BIC model:

```r
bic_model <- step(imputed_collapse_full_ph,
                  k = log(nrow(imputed_complete_data_collapse)),
                  direction = "both")
```

```r
summary(bic_model)
```

```
## Call:
## coxph(formula = stroke.surv ~ work_type, data = imputed_complete_data_collapse)
## 
##   n= 5109, number of events= 249 
## 
##                             coef exp(coef) se(coef)      z Pr(>|z|)   
## work_typeOther           0.04887   1.05009  0.19259  0.254  0.79968   
## work_typeSelf-employed -0.56044   0.57096  0.21570 -2.598  0.00937 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
##                        exp(coef) exp(-coef) lower .95 upper .95
## work_typeOther             1.050     0.9523    0.7199    1.5317
## work_typeSelf-employed     0.571     1.7515    0.3741    0.8714
## 
## Concordance= 0.573  (se = 0.019 )
## Likelihood ratio test= 18.54  on 2 df,   p=9e-05
## Wald test            = 17.1  on 2 df,   p=2e-04
## Score (logrank) test = 17.6  on 2 df,   p=2e-04
```

As expected, BIC imposes a much stricter penalty term and only takes work type as the only predictor whereas AIC takes work type, average glucose level and bmi as predictors.

There is nothing that stops us from adding smoking into either of this model as long as it is emprically and clinically valid (Let’s assume it is). I am more inclined to add smoking status to the AIC model for the better prediction performance.

```
aic_model_with_smoke <- coxph(stroke.surv ~ work_type
                              + avg_glucose_level + bmi + smoking_status,
                     data = imputed_complete_data_collapse)
summary(aic_model_with_smoke)
```

```
## Call:
## coxph(formula = stroke.surv ~ work_type + avg_glucose_level +
##     bmi + smoking_status, data = imputed_complete_data_collapse)
##
##   n= 5109, number of events= 249
##
##                                 coef exp(coef)  se(coef)      z Pr(>|z|)
## work_typeOther              0.027207  1.027580  0.192820  0.141   0.8878
## work_typeSelf-employed     -0.553682  0.574830  0.215850 -2.565   0.0103 *
## avg_glucose_level           0.002369  1.002372  0.001049  2.258   0.0239 *
## bmi                         0.023558  1.023837  0.010551  2.233   0.0256 *
## smoking_statusnever smoked -0.166570  0.846564  0.142008 -1.173   0.2408
## smoking_statussmokes        0.162461  1.176403  0.180976  0.898   0.3693
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##                            exp(coef) exp(-coef) lower .95 upper .95
## work_typeOther                1.0276     0.9732    0.7042    1.4995
## work_typeSelf-employed        0.5748     1.7396    0.3765    0.8775
## avg_glucose_level             1.0024     0.9976    1.0003    1.0044
## bmi                           1.0238     0.9767    1.0029    1.0452
## smoking_statusnever smoked    0.8466     1.1812    0.6409    1.1182
## smoking_statussmokes          1.1764     0.8500    0.8251    1.6773
##
## Concordance= 0.615  (se = 0.023 )
## Likelihood ratio test= 34.41  on 6 df,   p=6e-06
## Wald test            = 33.87  on 6 df,   p=7e-06
## Score (logrank) test = 34.5  on 6 df,   p=5e-06
```

So smoking is still not that significant, but leaving it in our model does have a nice interpretation in our model. Never smokes has the lowest risk of getting a stroke, formerly smoked is somewhere in between and smoking has the highest risk. One may consider leaving it in our model.

And the work type “Other” does not have significantly different effect on the risk compared to government jobs.

And all the other covariates are significant.

Before we are done here, there is still one thing that we must do, and that is to check if our Cox-proportional-hazard model is appropriate. That is to say, are the proportional hazard assumption violated?

Checking the validity of proportional hazard assumption

First thing to check is that are the covariate time invariant, that is to say, after the subject has entered our study, can the covariates’ values change? If so, we should account for them by using time variant analysis.

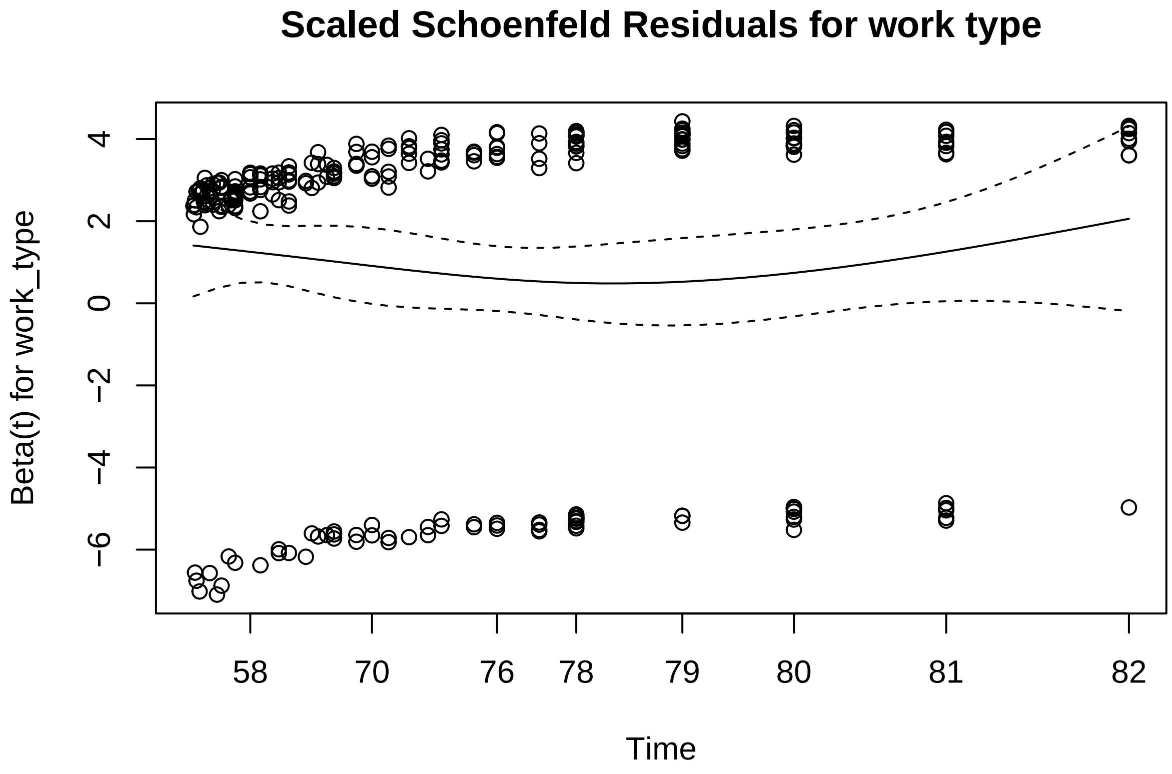
However, in this data set, we don't have that luxury to know if this assumption hold because we don't know how the study is conducted. Not much we can do other than to recognize this limitation.

Second thing to check is that are the coefficient time invariant? i.e. given 2 groups, their relative hazard should be proportional at all time. This we can actually check by scaled Schoenfeld residual.

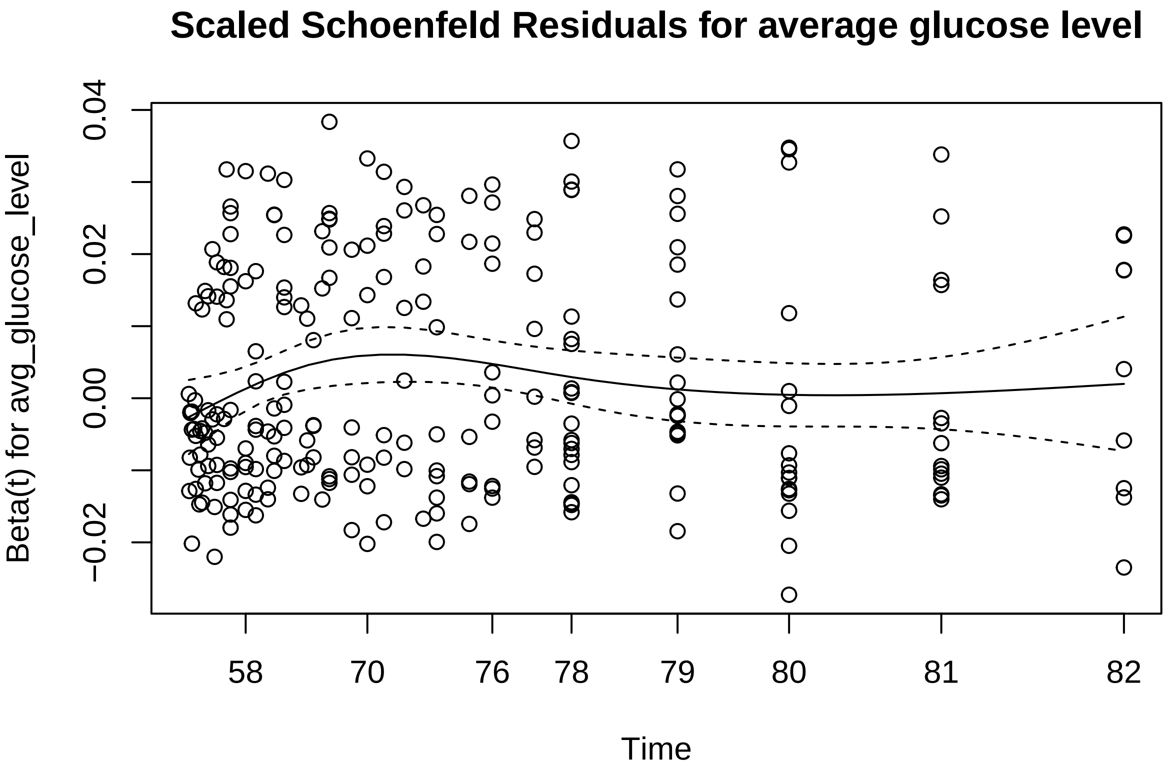
```
aic_model_with_smoke_zph <- cox.zph(aic_model_with_smoke)
aic_model_with_smoke_zph$table
```

```
##                        chisq df          p
## work_type          0.6917060  2 0.70761650
## avg_glucose_level  0.4445405  1 0.50493904
## bmi                8.4325755  1 0.00368558
## smoking_status     6.2054533  2 0.04492654
## GLOBAL            16.7593434  6 0.01020960
```

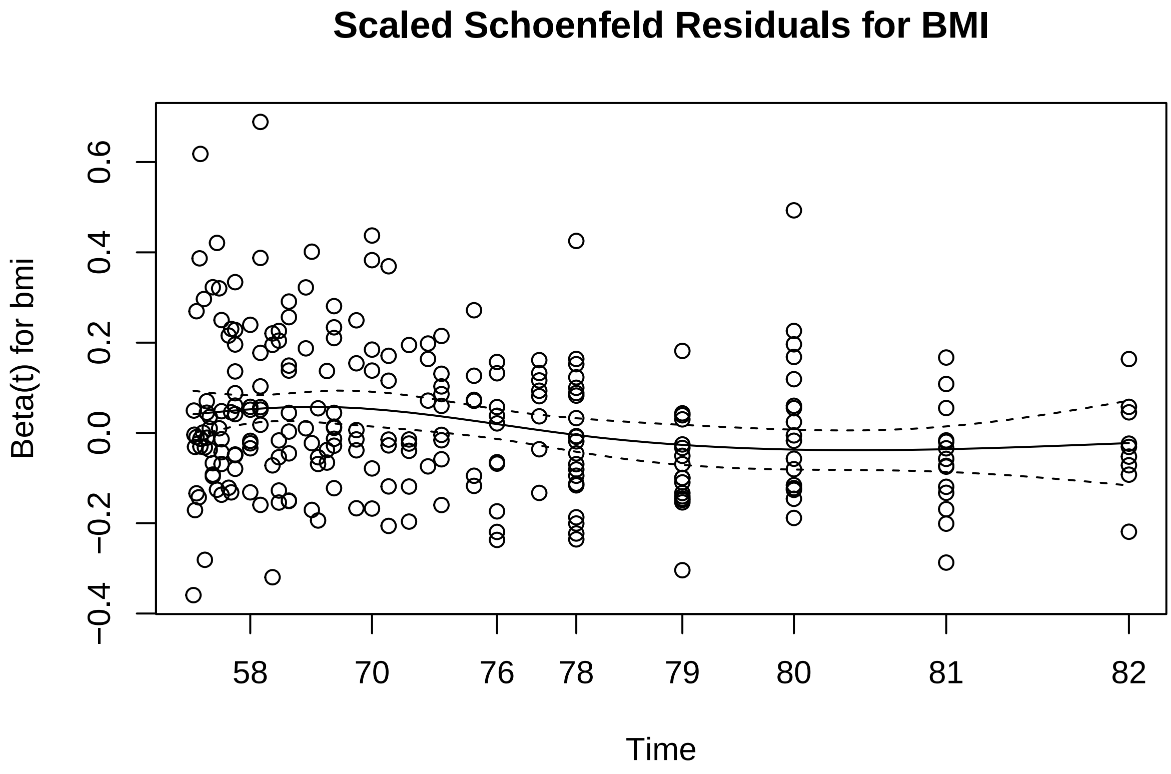
```
plot(aic_model_with_smoke_zph[1], main = "Scaled Schoenfeld Residuals for work type")
```



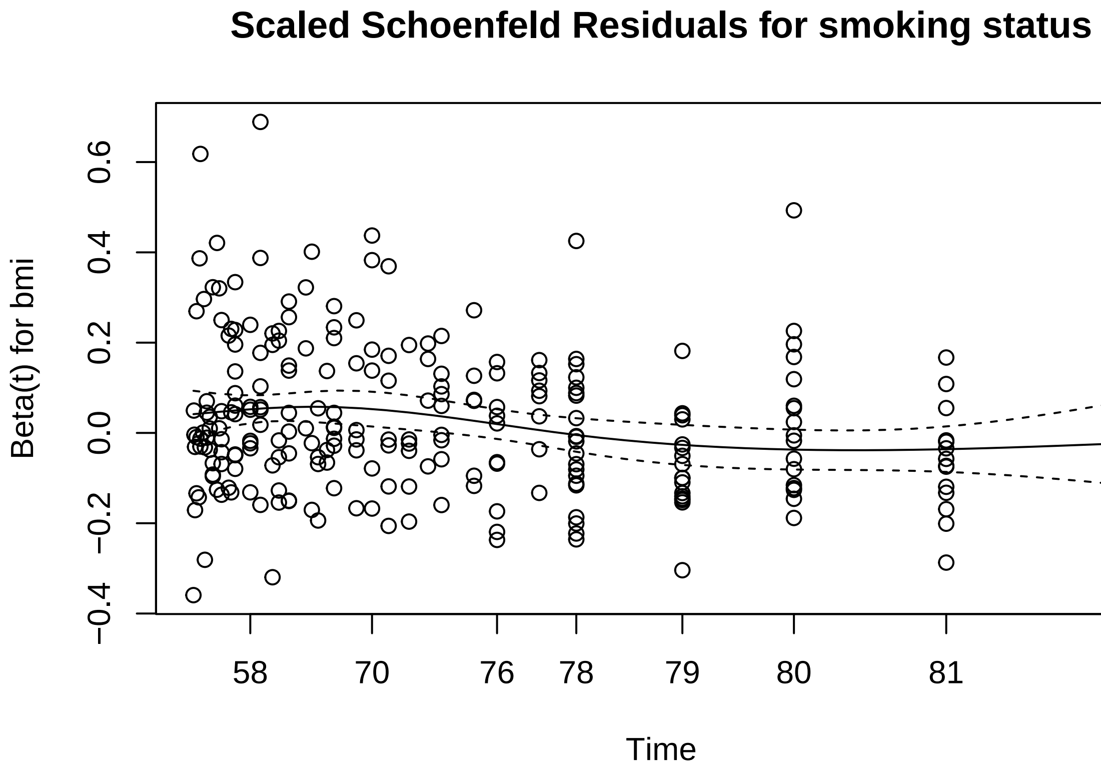
```
plot(aic_model_with_smoke_zph[2], main = "Scaled Schoenfeld Residuals for average glucose level")
```



```
plot(aic_model_with_smoke_zph[3], main = "Scaled Schoenfeld Residuals for BMI")
```



```
plot(aic_model_with_smoke_zph[3], main = "Scaled Schoenfeld Residuals for smoking status")
```



bmi and smoking status have very small p values, suggesting that they might violate the proportional hazard assumption.

Adressing the violation of PH assumption

We can try use a time variant coefficient and try to fix it but it is hard to tell from the plot what the trend is like.

Or we can also try a simpler approach, we can try stratify both of these variables, and fit different Cox-ph models on these strata. It is like saying, I know the hazard ratio of group A vs B changes over time, so I might as well not bother with their hazard ratio and just estimate the hazard ratio of the other covariates while adjusting their baseline risk according to whether the sample falls in group A or B.

Or… and I might sound crazy, but the simplest approach is to leave it be. The violation of the proportional hazard assumption does not always pose a concern for our analysis. The estimates given for the variables where proportional hazard assumption does not hold can be interpreted as the average effect.

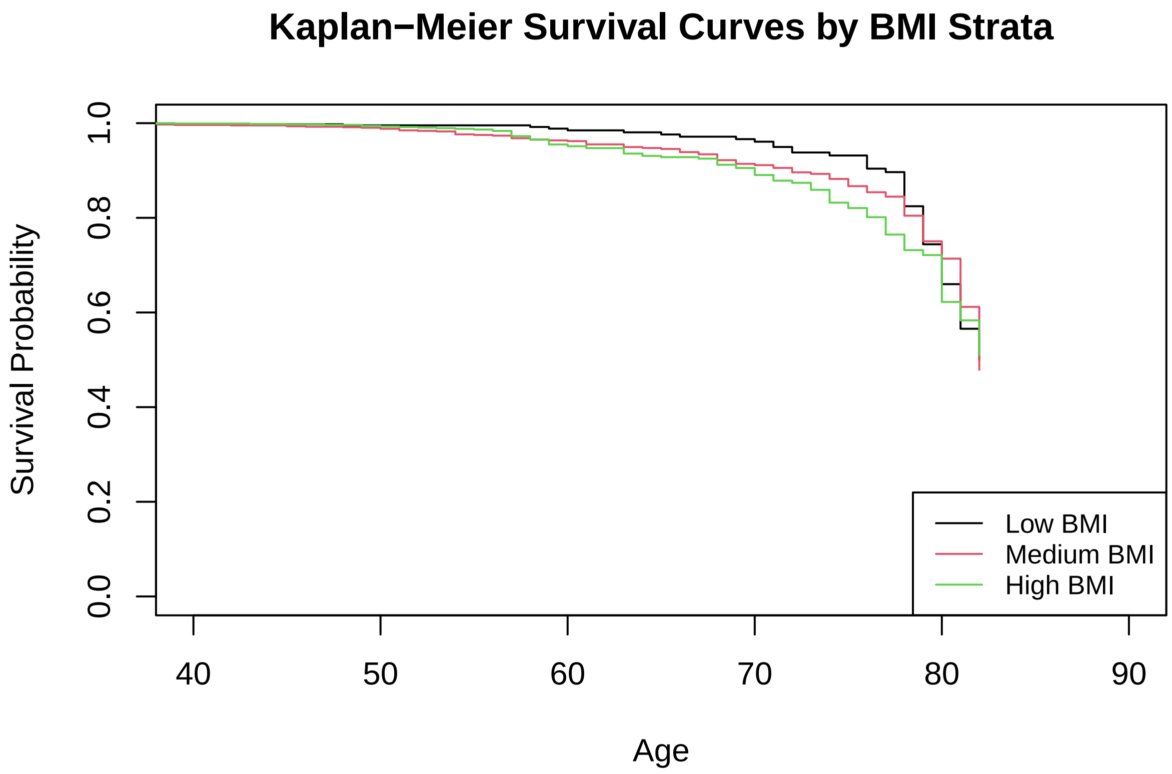
Before we decide which strategy we should go for, we can plot out the survival curves of each strata and decide.

We can first stratify BMI into say 3 strata

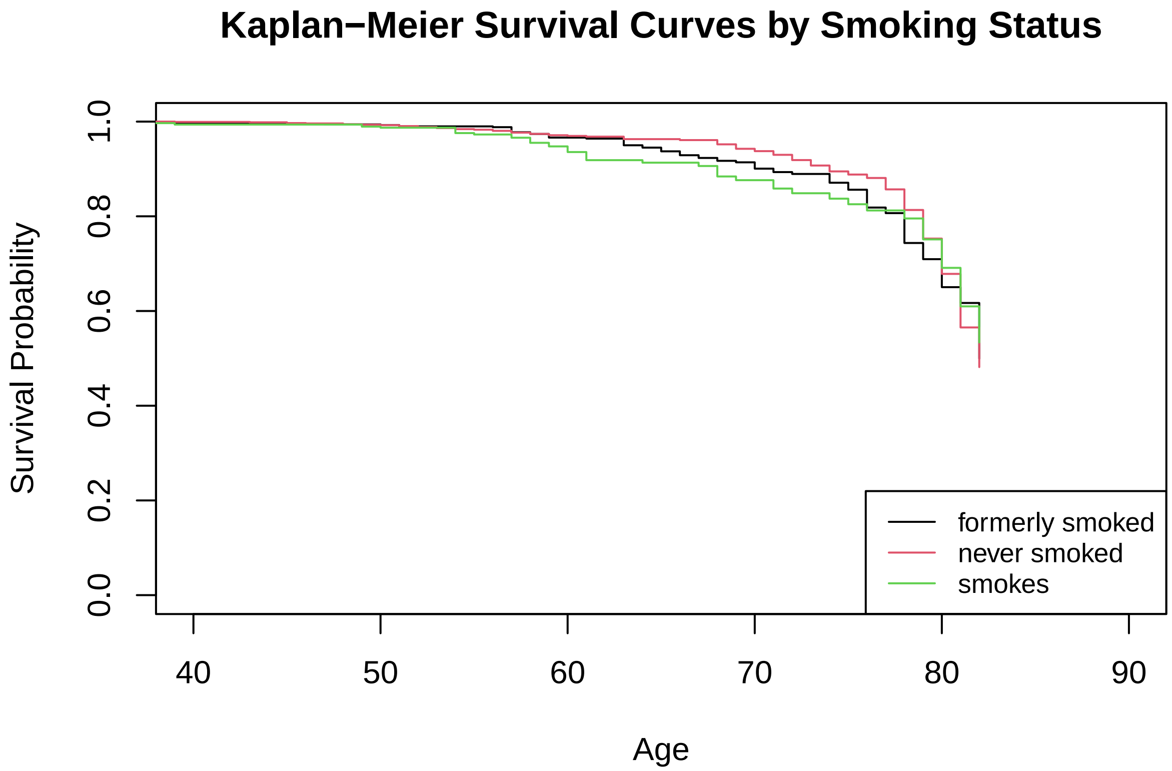
```
imputed_complete_data_collapse_stratified <-
  imputed_complete_data_collapse %>% mutate(
  bmi_strata = ntile(bmi, 3) %>%
    factor(labels = c("Low BMI", "Medium BMI", "High BMI"))
)
```

I know, the name is getting ridiculous but now we can plot out the survival curves for different strata of bmi and smoking status

```
km_fit_bmi <- survfit(stroke.surv ~ bmi_strata,
                      data = imputed_complete_data_collapse_stratified)
plot(km_fit_bmi, col = 1:3, xlab = "Age", ylab = "Survival Probability",
     main = "Kaplan-Meier Survival Curves by BMI Strata", xlim = c(40,90))
legend("bottomright", legend = levels(imputed_complete_data_collapse_stratified$bmi_strata),
       col = 1:3, lty = 1, cex = 0.8)
```



```
km_fit_smoke <- survfit(stroke.surv ~ smoking_status, data = imputed_complete_data_collapse_stratified)
plot(km_fit_smoke, col = 1:3, xlab = "Age", ylab = "Survival Probability",
     main = "Kaplan-Meier Survival Curves by Smoking Status", xlim = c(40,90))
legend("bottomright", legend =
         levels(imputed_complete_data_collapse_stratified$smoking_status),
       col = 1:3, lty = 1, cex = 0.8)
```



Oof, looks like both plots show clear convergence of curves, which is a tale-tell sign that the proportional hazard violation is severe. Given our decent sample size, it is likely more than just noisy data.

It might be worth it to try the stratified Cox-ph model. and do a side by side comparison

Comparing stratified model and un-stratified model

```
## stratified_ph_model summary:

##                                  coef exp(coef)    se(coef)          z   Pr(>|z|)
## work_typeOther          0.054382302 1.0558882 0.193330477  0.2812919 0.77848650
## work_typeSelf-employed -0.543708185 0.5805913 0.216140252 -2.5155341 0.01188522
## avg_glucose_level       0.002577503 1.0025808 0.001039908  2.4785874 0.01319038

## --------------------------------------------------------------------------------

## aic_model_with_smoke summary:

##                                      coef exp(coef)    se(coef)          z
## work_typeOther              0.027206925 1.0275804 0.192820426  0.1410998
## work_typeSelf-employed     -0.553681690 0.5748296 0.215850149 -2.5651207
## avg_glucose_level           0.002369139 1.0023719 0.001049019  2.2584334
## bmi                         0.023557633 1.0238373 0.010551066  2.2327254
## smoking_statusnever smoked -0.166569621 0.8465639 0.142007523 -1.1729634
## smoking_statussmokes        0.162461386 1.1764029 0.180976197  0.8976948
##                              Pr(>|z|)
## work_typeOther             0.88779110
## work_typeSelf-employed     0.01031399
## avg_glucose_level          0.02391866
## bmi                        0.02556706
## smoking_statusnever smoked 0.24081050
## smoking_statussmokes       0.36934830
```

We see, we have a much more significant p-values for the stratified model. It is because we have accounted for different base line risk function between each strata and prevented the skewness introduced in the old model.

But it is a trade-off. Even though the stratified_ph_model better models the relationship of the covariates and the risk, we do lose the information on how bmi and smoking status can affect the risk. Even if the estimates in the old model are technically incorrect, they can still be useful because as mentioned before, they can be interpreted as the average effect.

If we still want that information, we can employ a time transform function on bmi and smoking status. However, modeling this interaction requires a lot more effort and some specialty knowledge. That is beyond the scope of this project and honestly, it is kind of above my pay-grade without a lot more reading :(

So let's just move on…

But now we need to choose whether we should use the stratified model or not, and there is arguments to be made for both. In this case I will go on with the stratified model.

Now that we are done with all the diagnostic, I want to give you a friendly reminder that all the model fitted so far used only one instance of the imputed data, to get a valid estimation, we need to consider all the imputed data set.

Obtaining the valid estimates by pooling all the imputed data sets

But first we need to collapse the factor of work_type like we did for imputed_complete_data_collapse across all the data set.

```
completed_datasets <- complete(imputed_data, "long", include = TRUE)

completed_datasets_collapsed <- completed_datasets %>%
  mutate(
    work_type = fct_collapse(work_type,
      Govt_job = "Govt_job",
      'Self-employed' = "Self-employed",
      Other = c("children", "Never_worked", "Private")
    ),
    work_type = relevel(work_type, ref = "Govt_job"),
    bmi_strata = ntile(bmi, 3) %>%
    factor(labels = c("Low BMI", "Medium BMI", "High BMI"))
  )


imputed_data_collapsed <- as.mids(completed_datasets_collapsed)
```

Now we can use the full imputed data set in our final model

```
imputed_final_ph <- with(imputed_data_collapsed,
                         coxph(stroke.surv ~ work_type + avg_glucose_level + strata(bmi_strata) + strat

pooled_final_ph <- pool(imputed_final_ph)
```

```
## pooled_final_ph summary:
```

```
##                      term     estimate   std.error  statistic       df
## 1          work_typeOther  0.069927927 0.193992523  0.3604671 241.9241
## 2 work_typeSelf-employed -0.520762645 0.217359410 -2.3958597 240.3499
## 3       avg_glucose_level  0.002511112 0.001049212  2.3933319 243.2061
##      p.value
## 1 0.71881236
## 2 0.01734716
## 3 0.01745518
```

```
## --------------------------------------------------------------------------------
```

```
## Estimated 95% CI
```

```
##                      term   Lower_95_CI  Upper_95_CI
## 1          work_typeOther -0.3102974190  0.450153273
## 2 work_typeSelf-employed -0.9467870877 -0.094738202
## 3       avg_glucose_level  0.0004546569  0.004567567
```

Phew, that was a lot of work, but at last we obtained the final model that we have by using the imputation approach.

Comparing our results to complete case analysis

Now that we are done with multiple imputation, we can now do a complete case analysis where all the observations containing NA values are dropped.

For this to be statistically valid, we need to assume the data is missing completely at random (MCAR), which is a strong assumption that states the probability of the data missing does not depend on the observed data. This could be unrealistic but this section only serves as a sensitivity analysis so it does not pose a major concern for us.

```
data_dropped <- data_clean %>% drop_na()
sapply(data_dropped %>% dplyr::select(age,
  avg_glucose_level, bmi, smoking_status), function(x) sum(is.na(x)))
```

```
##               age avg_glucose_level               bmi    smoking_status
##                 0                 0                 0                 0
```

And then we also need the same releveling and factor collapsing for data_dropped as in imputed_complete_data_collapse for a fair comparison.

```
data_dropped_collapse <- data_dropped %>% mutate(
  work_type = fct_collapse(work_type,
                           Govt_job = "Govt_job",
                           'Self-employed' = "Self-employed",
                           Other = c("children", "Never_worked", "Private")
  ),
  work_type = relevel(work_type, ref = "Govt_job"),
  bmi_strata = ntile(bmi, 3) %>%
    factor(labels = c("Low BMI", "Medium BMI", "High BMI"))
)
```

Now we can run the same model as before on this complete data set and do a side by side comparison

```
stroke.surv_dropped <- Surv(data_dropped_collapse$age, data_dropped_collapse$stroke)
complete_case_ph <- coxph(stroke.surv_dropped ~ work_type + avg_glucose_level +
                            strata(bmi_strata) + strata(smoking_status),
                          data = data_dropped_collapse)
```

```
## pooled_final_ph summary:

##                      term     estimate   std.error  statistic    p.value
## 1          work_typeOther  0.069927927 0.193992523  0.3604671 0.71881236
## 2 work_typeSelf-employed -0.520762645 0.217359410 -2.3958597 0.01734716
## 3       avg_glucose_level  0.002511112 0.001049212  2.3933319 0.01745518

## --------------------------------------------------------------------------------

## complete_case_ph summary:

##                                coef    se(coef)          z    Pr(>|z|)
## work_typeOther          0.178688516 0.231549513  0.7717076 0.440287635
## work_typeSelf-employed -0.413854964 0.256607343 -1.6127947 0.106789117
## avg_glucose_level       0.003324793 0.001192001  2.7892527 0.005282983
```

Because these 2 summary are coming from 2 different R objects, the names are going to be a bit different. But each column of one table are representing the same things to the corresponding column from the other table. i.e. coef == estimate.

With that out of the way, we do see some very interesting results here. We see a drastically different p-value across all covariates. That said, the direction of the effects agree with each other (e.g. avg_glucose_level increases the risk in both level).

The difference in p-value could be due to a few potential issues:

1. We used too little imputations, 20-40 imputations is the standard, we used 5 only.
2. It could be the case that smoking status cannot be accurately predicted by our simple imputation scheme, and hence we produced noisy imputed data that skews the result. For example, we might mispecified bmi as having a linear association with other covariate when the real association is non linear. Leading to the discrepency.
3. It could also be the case that the data is genuinely not missing at random (NMAR) and our prediction systematically bias toward certain values, leading to poor predictions. While at the same time, dropping observations with NA values also introduce a different bias.

4. We could simply have too many missing data, rendering our imputed model unstable and perhaps by chance, had a very different result from the model that runs on the complete data set.
5. We have different number of observations in the 2 data sets, leading to the different estimates of the std.error.

But overall, they seem largely consistent. Usually, the imputed model is preferred because it is potentially less biased than the complete case model. Complete case case could yield worse result if the missing probability depends on the response variable, and only similar result to imputed model otherwise. So imputed data model generally outperforms complete case model.

Lesson learned and summary

In this project, various techniques in survival modeling and multiple imputation are employed including KM curves, proportional hazard model and its stratified variant, multiple imputation, diagnostics and sensitivity analysis.

Upon reviewing it and thanks to the feed back from Onur Ramazan, I have identified several potential pitfall.

1. More chains should be used when imputing the data, 20 to 40 is generally the standard,
2. I started this project without a hypothesis in mind even though the number of variables is not that big. Instead I did a data driven approach which could inflate type I error. If the number of variables is not that big, confirmatory analysis is the gold standard.
3. I trusted the p-value after stepwise selection is run. However the p-value is already invalid after the model selection event. Selective inference could be performed if valid p-value is required after model selection (using LASSO).
4. The way the model is picked is based off of only one imputed data set. This ignores the variability between different data set as different models could be picked in other data set. MI-LASSO, majority rule, stacked analysis and Wald's multivariate test could be used to rigorously pick a model accounting for all the different data set.
5. When performing G test on the independence of work_type, Fisher's exact test should have been used for a more robust result given the small cell count.

These mistakes have been noted and will be surely be avoided in my coming analysis.

This project has been a fruitful one, I have put many of the theory that I have learned into use and learn how these techniques interact with imperfect data. It is also intellectually stimulating to constantly contemplate different approaches to a problem and the assumptions that allow our techniques to work. And upon reviewing my work, I have discovered many new things, some of which (such as MI-LASSO and selective inference) are really technical graduate level material, and researching them has been incredibly eye opening.

If you have made it this far to my analysis, I want to say thank you and hopefully it has been as entertaining for you as it is for me to work on it. If you have any feedback or spotted any other mistakes that I did not realize, feel free to point them out! I am still learning.

That's it for now!