Regression Code Templates

2024-02-29

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(broom)
library(patchwork)
```

Data

National Health and Nutrition Examination Survey is conducted by the National Center for Health Statistics (NCHS). The goal is to "assess the health and nutritional status of adults and children in the United States".

```
library(NHANES)
data(NHANES)
```

Preprocessing data before regression analysis

1. Check missingness

```
NHANES %>% drop_na(HealthGen) %>% head(5)
```

```
## # A tibble: 5 x 76
##      ID SurveyYr Gender   Age AgeDecade AgeMonths Race1 Race3 Education
##   <int> <fct>    <fct>  <int> <fct>         <int> <fct> <fct> <fct>
## 1 51624 2009_10  male      34 " 30-39"        409 White <NA>  High School
## 2 51624 2009_10  male      34 " 30-39"        409 White <NA>  High School
## 3 51624 2009_10  male      34 " 30-39"        409 White <NA>  High School
## 4 51630 2009_10  female    49 " 40-49"        596 White <NA>  Some College
## 5 51647 2009_10  female    45 " 40-49"        541 White <NA>  College Grad
## # i 67 more variables: MaritalStatus <fct>, HHIncome <fct>, HHIncomeMid <int>,
```

```
## #   Poverty <dbl>, HomeRooms <int>, HomeOwn <fct>, Work <fct>, Weight <dbl>,
## #   Length <dbl>, HeadCirc <dbl>, Height <dbl>, BMI <dbl>,
## #   BMICatUnder20yrs <fct>, BMI_WHO <fct>, Pulse <int>, BPSysAve <int>,
## #   BPDiaAve <int>, BPSys1 <int>, BPDia1 <int>, BPSys2 <int>, BPDia2 <int>,
## #   BPSys3 <int>, BPDia3 <int>, Testosterone <dbl>, DirectChol <dbl>,
## #   TotChol <dbl>, UrineVol1 <int>, UrineFlow1 <dbl>, UrineVol2 <int>, ...
```

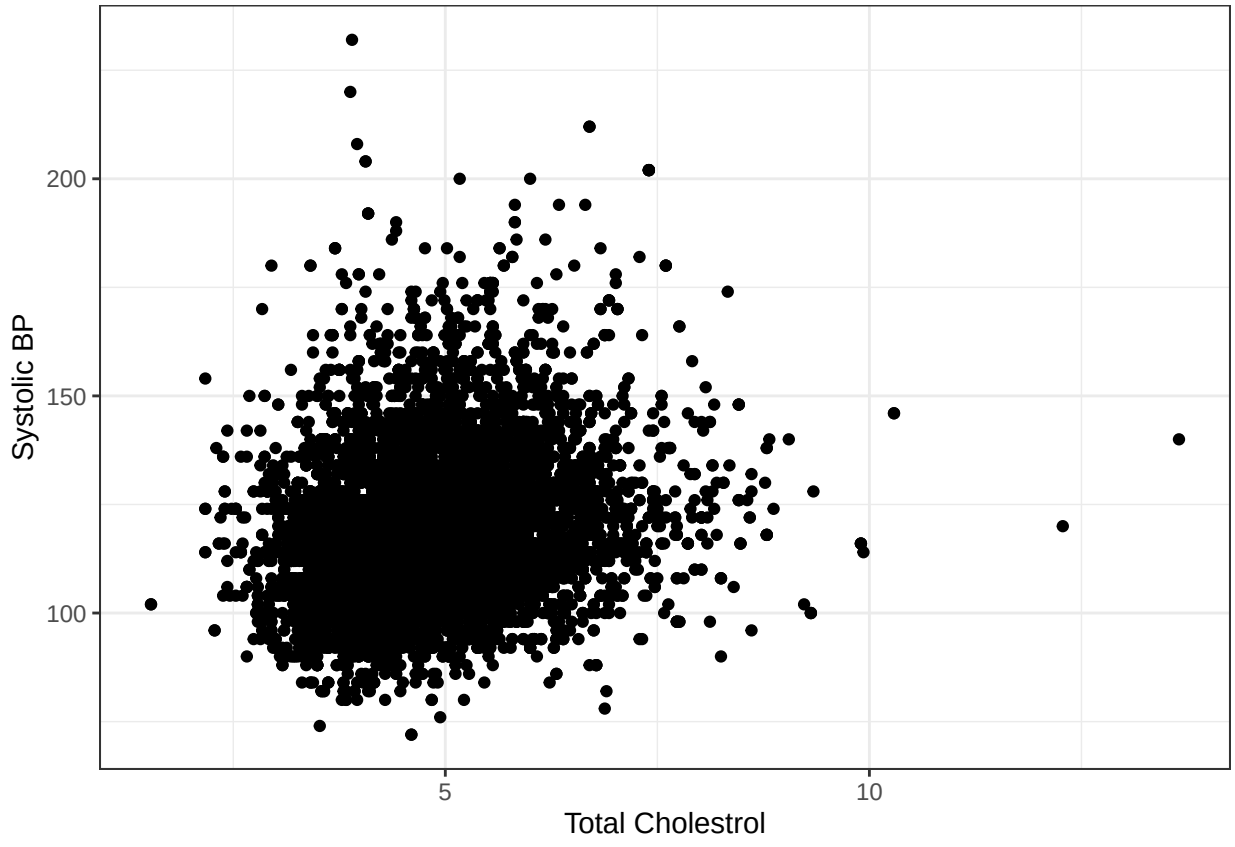
2. Ensure variables are the right type

```
#sapply(NHANES, class)
```

Simple Linear Regression:

```
ggplot(data = NHANES, mapping = aes(x =TotChol, y = BPSys1)) +
  geom_point() +
  labs(x = "Total Cholestrol", y = "Systolic BP") +
  theme_bw()
```

```
## Warning: Removed 2306 rows containing missing values (‘geom_point()‘).
```



```r
# fitting the model
slr <- lm(BPSys1 ~ TotChol, data = NHANES)

# showing regression output option 1, the * indicate statistical significance
summary(slr)
```

```
##
## Call:
## lm(formula = BPSys1 ~ TotChol, data = NHANES)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.885 -11.276  -2.262   8.933 116.021
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.0146     0.9025  114.14   <2e-16 ***
## TotChol       3.3242     0.1800   18.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.96 on 7692 degrees of freedom
##   (2306 observations deleted due to missingness)
## Multiple R-squared:  0.04247,    Adjusted R-squared:  0.04234
## F-statistic: 341.2 on 1 and 7692 DF,  p-value: < 2.2e-16
```

```r
# showing regression output option 2, can automatically generate confidence intervals, 95% by default.
tidy(slr, conf.int = TRUE)
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)   103.       0.903     114.  0          101.      105.
## 2 TotChol         3.32     0.180      18.5 1.44e-74     2.97      3.68
```

Hypothesis testing:

- Formula: Systolic BP = 103.0146 + 3.32 Total Cholesterol
- Each coefficient is a hypothesis test (t-test)
  - Standard error = standard deviation / sqrt(n)
  - T-test statistics = (Estimated value - hypothesized value) / standard error = Estimated value / standard error
  - P-value = the probability of observing a t-test that's bigger or equal to than this value given the null hypothesis is true
  - Confidence interval = [Estimated value -1.96 x standard error; Estimated value + 1.96 x standard error]
- Note: If the sample size is large enough, the test will likely result in rejecting (e.g. > 1000 patients). Consider the practical significance of the result not just the statistical significance. If the sample size is small, there may not be enough evidence to reject

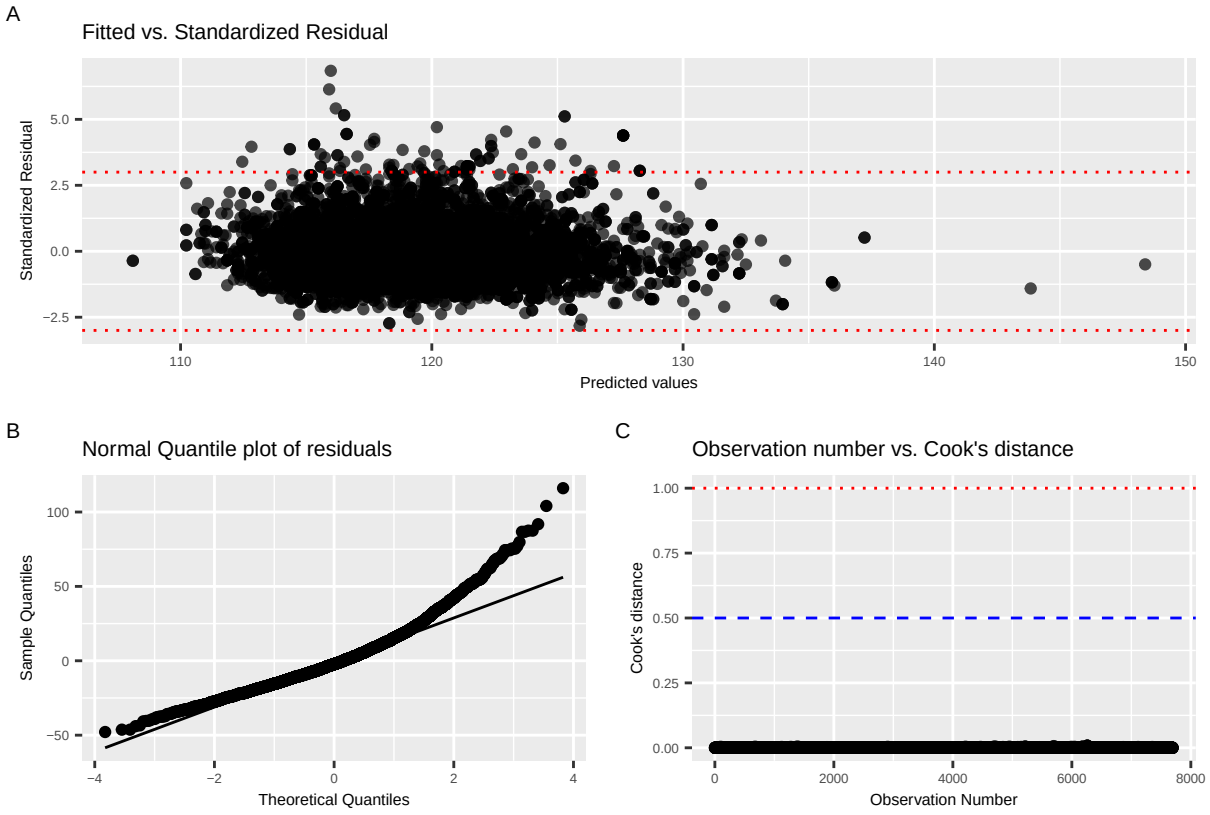
Interpretation:

- Slope: For every one point increase in total cholesterol, we expect the systolic blood pressure to increase by 3.32 points, on average.
- Intercept: If the total cholesterol is 0, we expect the systolic blood pressure to be 103.014.
  - Only interpret the interpret if 1) the predictor can feasibly take values equal to or near zero; 2) There are values near zero in the data

Checking the assumption for your regression models:

- 1. Linearity: There is a linear relationship between the response and predictor variable.
- 2. Constant Variance: The variability of the errors is equal for all values of the predictor variable.
- 3. Normality: The errors follow a normal distribution.
- 4. Independence: The errors are independent from each other.

```
diag_plot(slr)
```



Multiple Linear Regression - adjusting for confoundings!

```r
# converting a categorical variable to the right type before fitting the model
NHANES <- NHANES %>%
  mutate(Gender = as.factor(Gender))

# fitting the model
mlr <- lm(BPSys1 ~ TotChol + Age + Gender + SleepHrsNight, data = NHANES)

tidy(mlr, conf.int = TRUE)
```

```
## # A tibble: 5 x 7
##   term          estimate std.error statistic   p.value conf.low conf.high
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)     97.2      1.36       71.4  0           94.5    99.9
## 2 TotChol          0.968    0.175       5.52 3.43e-  8    0.624   1.31
## 3 Age              0.417    0.0104     40.0  6.25e-314    0.397   0.437
## 4 Gendermale       4.52     0.367      12.3  1.44e- 34    3.81    5.24
## 5 SleepHrsNight   -0.273    0.136      -2.01 4.44e-  2   -0.540  -0.00679
```

Interpretation:

- Numerical variables: for every mg/dL increase in total cholesterol, we expect the systolic blood pressure to increase by 0.96, on average, **holding all other predictor variables constant**.
- Categorical variables: compared to females, males on average have 4.52mmHg higher systolic blood pressure, holding all other predictor variables constant.

Comparing models:

R-squared = variance explained by the model / total variance

```r
summary(mlr)$r.squared
```

```
## [1] 0.2188266
```

```r
mlr2 <- lm(BPSys1 ~ TotChol + Age + Gender + SleepHrsNight + Work + Weight +
             Height + Pulse, data = NHANES)
summary(mlr2)$r.squared
```

```
## [1] 0.2340117
```

```r
summary(mlr2)$adj.r.squared
```

```
## [1] 0.232991
```

R2 will always increase as we add more variables to the model. Adjusted R2: measure that includes a penalty for unnecessary predictor variables.

Alternatively, you can use AIC or BIC. The AIC and BIC values themselves are not meaningful, but you can use them to compare models.

```r
glance(mlr) %>%
  dplyr::select(AIC, BIC)
```

```
## # A tibble: 1 x 2
##       AIC    BIC
##     <dbl>  <dbl>
## 1 56330. 56371.
```

Interaction terms:

Question of itnerest: Does the association between total cholestrol and SBP varies based on the patient's gender?

```r
mlr <- lm(BPSys1 ~ TotChol + Gender + TotChol*Gender, data = NHANES)

tidy(mlr, conf.int = TRUE)
```

```
## # A tibble: 4 x 7
##   term                estimate std.error statistic  p.value conf.low conf.high
##   <chr>                  <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)            95.9      1.30      73.9  0           93.3      98.4
## 2 TotChol                 4.34     0.255     17.0  6.08e-64     3.84      4.84
## 3 Gendermale             12.6      1.79       7.01 2.61e-12     9.06     16.1
## 4 TotChol:Gendermale     -1.71     0.358     -4.79 1.72e- 6    -2.41     -1.01
```

Interpretation:

- The effect of total cholesterol on systolic BP differs by -1.712 when the patient is male compared to when the patient is female, holding all else constant.
- If the patient is female, we expect the systolic BP to increase by 4.33 for each point increase in total cholesterol, holding all else constant. If the patient is male, we expect the total cholesterol to increase by 2.62 (= 4.33 - 1.71) for each point increase in total cholesterol, holding all else constant.
- Note:
  - 1) when adding interaction terms, always make sure the main effect is in the model.
  - 2) don't do >2 way interactions because they are very hard to interpret.

Logistic Regression

Formulation of logistic regression:

$$Log(\frac{p}{1-p}) = mx + b$$

Note:

- p is the probability and $\frac{p}{1-p}$ is the odds
- logs here means natural log

```r
# make sure the outcome is binary (factor)
NHANES <- NHANES %>%
  mutate(Diabetes = as.factor(Diabetes))

# fitting the model
lr <- glm(Diabetes ~ Gender + BMI + TotChol, data = NHANES, family = "binomial")

# showing regression output
# for logistic regression, this shows the **un-exponentiated** output (log odds ratio)
tidy(lr, conf.int = TRUE)
```

```
## # A tibble: 4 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)  -4.54     0.253      -18.0  4.69e-72  -5.04     -4.05
## 2 Gendermale    0.246    0.0829       2.97 3.00e- 3   0.0838    0.409
## 3 BMI           0.0945   0.00513     18.4  7.14e-76   0.0845    0.105
## 4 TotChol      -0.161    0.0402      -4.01 6.12e- 5  -0.240    -0.0829
```

```r
# for logistic regression, this shows the **exponentiated** output (odds ratio)
tidy(lr, exponentiate = TRUE, conf.int = TRUE)
```

```
## # A tibble: 4 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)   0.0106   0.253      -18.0  4.69e-72  0.00645    0.0174
## 2 Gendermale    1.28     0.0829       2.97 3.00e- 3  1.09       1.51
## 3 BMI           1.10     0.00513     18.4  7.14e-76  1.09       1.11
## 4 TotChol       0.851    0.0402      -4.01 6.12e- 5  0.786      0.920
```

Interpretation:

- For each additional points on BMI, the odds of having diabetes are expected to **multiply** by a factor of 1.09 (exp(0.094)), holding all else constant.
- The odds of having diabetes for those who are male is expected to be 1.27 (exp(0.24)) times the odds for female patients, holding all else constant.

Multinomial logistic Regression:

```r
library(nnet) # package for multinomial logistic regression

NHANES %>% count(HealthGen)
```

```
## # A tibble: 6 x 2
##   HealthGen     n
##   <fct>     <int>
## 1 Excellent   878
## 2 Vgood      2508
## 3 Good       2956
```

```
## 4 Fair       1010
## 5 Poor        187
## 6 <NA>       2461
```

```
NHANES <- NHANES %>% mutate(HealthGen = as.factor(HealthGen),
                            PhysActive = as.factor(PhysActive))
```

```
health_m <- multinom(HealthGen ~ Age + PhysActive,
                     data = NHANES)
```

```
## # weights:  20 (12 variable)
## initial  value 12131.942984
## iter  10 value 10065.191767
## iter  20 value 9823.413287
## final  value 9823.413154
## converged
```

```
tidy(health_m, conf.int = TRUE, exponentiate = TRUE)
```

```
## # A tibble: 12 x 8
##    y.level term          estimate std.error statistic  p.value conf.low conf.high
##    <chr>   <chr>            <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 Vgood   (Intercept)     3.82     0.126      10.7   1.68e-26   2.98       4.88
##  2 Vgood   Age             0.998    0.00210    -0.794 4.27e- 1   0.994      1.00
##  3 Vgood   PhysActiveY~    0.732    0.0897     -3.48  5.06e- 4   0.614      0.873
##  4 Good    (Intercept)     6.89     0.123      15.7   8.78e-56   5.42       8.76
##  5 Good    Age             0.998    0.00207    -1.17  2.43e- 1   0.994      1.00
##  6 Good    PhysActiveY~    0.378    0.0871    -11.2   6.57e-29   0.319      0.449
##  7 Fair    (Intercept)     2.36     0.145       5.93  3.00e- 9   1.78       3.13
##  8 Fair    Age             1.00     0.00251     1.40  1.62e- 1   0.999      1.01
##  9 Fair    PhysActiveY~    0.205    0.103     -15.3   5.76e-53   0.167      0.251
## 10 Poor    (Intercept)     0.240    0.263      -5.41  6.16e- 8   0.144      0.403
## 11 Poor    Age             1.02     0.00448     4.55  5.36e- 6   1.01       1.03
## 12 Poor    PhysActiveY~    0.0698   0.228     -11.7   1.96e-31   0.0446     0.109
```

Interpretation:

- The baseline category for the model is Excellent.
- For each additional year in age, the odds a person rates themselves as having fair health versus excellent health are expected to multiply by 1.003 (exp(0.003)), holding physical activity constant.
- The odds a person who does physical activity will rate themselves as having fair health versus excellent health are expected to be 0.204 (exp(-1.58)) times the odds for a person who doesn't do physical activity, holding age constant.

Changing the baseline level:

```
NHANES <- NHANES %>%
  mutate(HealthGen = fct_relevel(HealthGen,
                                 c("Poor","Fair","Good", "Vgood", "Excellent")))

health_m <- multinom(HealthGen ~ Age + PhysActive,
                     data = NHANES)
```

```
## # weights:  20 (12 variable)
## initial  value 12131.942984
## iter  10 value 10063.345462
## iter  20 value 9823.417980
## final  value 9823.413154
## converged
```

```
tidy(health_m, conf.int = TRUE, exponentiate = TRUE)
```

```
## # A tibble: 12 x 8
##    y.level   term       estimate std.error statistic  p.value conf.low conf.high
##    <chr>     <chr>         <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 Fair      (Intercep~    9.81    0.254        8.98 2.75e-19    5.96     16.1
##  2 Fair      Age           0.983   0.00437     -3.86 1.13e- 4    0.975     0.992
##  3 Fair      PhysActiv~    2.93    0.225        4.79 1.68e- 6    1.89      4.56
##  4 Good      (Intercep~   28.7     0.244       13.8  4.66e-43   17.8      46.2
##  5 Good      Age           0.977   0.00417     -5.46 4.82e- 8    0.970     0.986
##  6 Good      PhysActiv~    5.42    0.218        7.76 8.53e-15    3.54      8.30
##  7 Vgood     (Intercep~   15.9     0.247       11.2  4.08e-29    9.78     25.7
##  8 Vgood     Age           0.978   0.00422     -5.22 1.80e- 7    0.970     0.986
##  9 Vgood     PhysActiv~   10.5     0.219       10.7  6.66e-27    6.83     16.1
## 10 Excellent (Intercep~    4.16    0.263        5.41 6.15e- 8    2.48      6.97
## 11 Excellent Age           0.980   0.00448     -4.55 5.34e- 6    0.971     0.988
## 12 Excellent PhysActiv~   14.3     0.228       11.7  1.96e-31    9.15     22.4
```

Ordinal Regression (Proportional odds model)

```
library(MASS)
```

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

```
## The following object is masked from 'package:patchwork':
##
##     area
```

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

```
ordinal_reg <- polr(HealthGen ~ Age + PhysActive, data = NHANES)

tidy(ordinal_reg, conf.int = TRUE, exponentiate = TRUE)
```

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

```
## # A tibble: 6 x 7
##   term            estimate std.error statistic conf.low conf.high coef.type
```

```
##   <chr>              <dbl>     <dbl>     <dbl>    <dbl>     <dbl> <chr>
## 1 Age               0.997    0.00113     -2.42    0.995     0.999 coefficient
## 2 PhysActiveYes     2.69     0.0455      21.7     2.46      2.94  coefficient
## 3 Poor|Fair         0.0351   0.0943     -35.5    NA        NA     scale
## 4 Fair|Good         0.272    0.0665     -19.6    NA        NA     scale
## 5 Good|Vgood        1.96     0.0648      10.4    NA        NA     scale
## 6 Vgood|Excellent  13.0      0.0719      35.7    NA        NA     scale
```

Note:

- A coefficient taking on a **negative** value means it increases the odds of being **less than or equal to** class J.

Poisson Regression

```r
Poisson_r <- glm(SleepHrsNight ~ Age + Gender, data = NHANES,
                 family = "poisson")

# Print the summary of the Poisson regression model
summary(Poisson_r)
```

```
##
## Call:
## glm(formula = SleepHrsNight ~ Age + Gender, family = "poisson",
##     data = NHANES)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.948e+00  1.251e-02 155.782   <2e-16 ***
## Age         -7.925e-05  2.396e-04  -0.331   0.7408
## Gendermale  -1.877e-02  8.637e-03  -2.173   0.0298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2108.7  on 7754  degrees of freedom
## Residual deviance: 2103.9  on 7752  degrees of freedom
##   (2245 observations deleted due to missingness)
## AIC: 31405
##
## Number of Fisher Scoring iterations: 4
```

Survival Analysis

Use case: time to event data in the presence of censoring

- Not all units are observed until their event times - (i.e. the outcome is not observed for everyone)
- In these cases, observations are said to be censored. We know that they survived until at least their censoring time, but do not know any further information.

```r
library(survival)
library(survminer)
```

```
## Loading required package: ggpubr
```

```
##
## Attaching package: 'survminer'
```

```
## The following object is masked from 'package:survival':
##
##     myeloma
```

```r
head(lung, 5)
```

```
##   inst time status age sex ph.ecog ph.karno pat.karno meal.cal wt.loss
## 1    3  306      2  74   1       1       90       100     1175      NA
## 2    3  455      2  68   1       0       90        90     1225      15
## 3    3 1010      1  56   1       0       90        90       NA      15
## 4    5  210      2  57   1       1       90        60     1150      11
## 5    1  883      2  60   1       0      100        90       NA       0
```

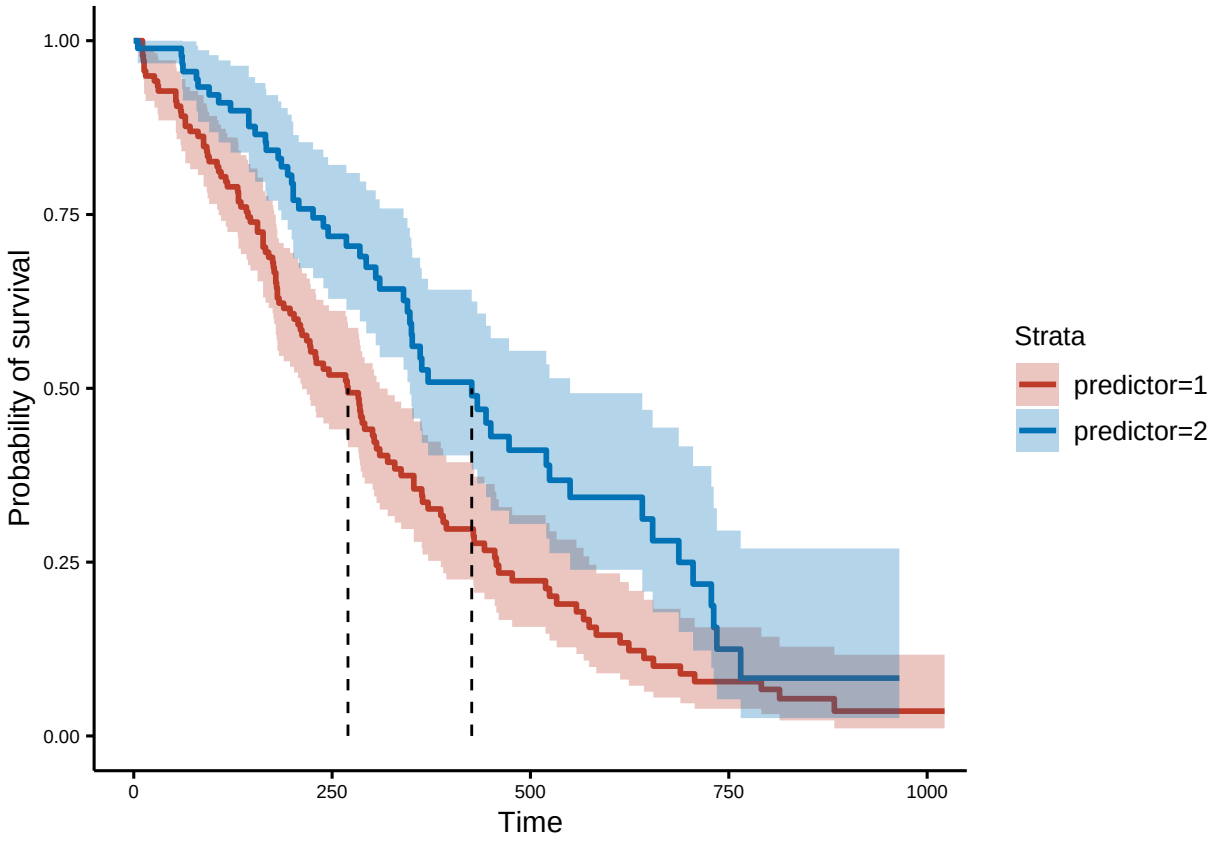
Kaplan Meier Survival Curve Analysis

The Kaplan Meier estimate is a non-parametric estimation of survival

- Probability of survival beyond time X = Probability of surviving to time X-1 * Probability of surviving at time X

```r
time_to_event <- lung$time
outcome <- lung$status
predictor <- lung$sex

ggsurvplot(survfit(Surv(time_to_event, outcome) ~ predictor, data = lung),
     xlab = "Time", ylab = "Probability of survival",
     ylim = c(0, 1),
     #risk.table = T,
     #tables.height = 0.25,
     conf.int = T,
     censor = F,
     palette = "nejm",
     #legend.labs = c("xxx", "yyy"),
     legend = "right",
     surv.median.line = "v", font.x = 11, font.y = 11,
     font.tickslab = c(7),
     font.legend = 10)
```




```
# log rank test
survdiff(Surv(time_to_event, outcome) ~ predictor)
```

```
## Call:
## survdiff(formula = Surv(time_to_event, outcome) ~ predictor)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## predictor=1 138      112     91.6      4.55      10.3
## predictor=2  90       53     73.4      5.68      10.3
##
##  Chisq= 10.3  on 1 degrees of freedom, p= 0.001
```

Note: If the Kaplan-Meier survival curves cross, then this is clear departure from proportional hazards, and the log rank test should not be used.

Cox Proportional Hazards Regression

```
lung <- lung %>%
  mutate(sex = as.factor(sex))

coxm1 <- coxph(Surv(time, status) ~ age + sex, data = lung)

tidy(coxm1, conf.int = TRUE, conf.level = 0.95, exponentiate = TRUE)
```

```
## # A tibble: 2 x 7
##   term  estimate std.error statistic p.value conf.low conf.high
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 age      1.02    0.00922      1.85 0.0646     0.999     1.04
## 2 sex2     0.599   0.167       -3.06 0.00218    0.431     0.831
```

Interpretation:

- Older age is a not significant risk factor for mortality. Each additional years in age is associated with 1% additional hazard for mortality, holding all else constant.
- Female patients are associated with 41% lower hazard of mortality compared to male patients, after adjusting for age.

A document with more technical details: https://docs.google.com/document/d/1dwgSMnwTl8B-CY-o6ZSrZJCeiIps4Qx3zomIjK6KfZ0/edit?usp=sharing

Takeaways

- Use the right type of model for the right type of data
- Use the appropriate metrics to compare models
- Keep the practical and clinical significance in mind when comparing models
- "All models are wrong but some are useful"

Other future topics?

- Data cleaning in R
- Machine learning
- Github
- Figure making

Reference:

- https://cran.r-project.org/web/packages/NHANES/NHANES.pdf
- https://sta210-fa20.netlify.app
- https://bioconnector.github.io/workshops/r-survival.html