The efficacy of treatment on liver tumours in a cohort*

Alan Ho

The text of your abstract. 150 – 250 words.

Setup

A good setup of a project requires the consideration of reproducibility that regardless of who or when a project is run the project can still run. There are two principles required this outcome. Firstly, is the consideration of R packages. Each project uses different packages and their versions which may change over time as different functions or classes within the package are added to or depreciated. Virtual environments such as `renv` assure that a snapshot of project package library is taken which records all the package versions of the project. This allows analysts and developers to develop R code on a common virtual environment and even deploy R code (such as R Shiny apps) to servers although the latter may require more sophisticated system containerisation beyond just managing R packages.

The second principle is a simple one of using relative paths mainly in this case where all file paths are relative to something everyone will have access to such as the ,rproj file when setting up an RStudio project. This is why we use the find_rstudio_file() to obtain this path. The goal is whenever a person clones this repository from Github, Bitbucket or some other repository system they should be able restore the virtual environment using `renv::restore()` and then render the R markdown document without changing any of the file paths.

```
library(rprojroot)
library(tidyverse)
library(survival)
library(Hmisc)
library(ggsurvfit)
library(broom)
```

*Grants or other notes about the article that should go on the front page should be placed here. General acknowledgments should be placed at the end of the article.

```r
library(cowplot)
library(compareGroups)
library(glue)
library(encryptr)

ROOT <- find_rstudio_root_file()
```

Loading Data

When it comes to clinical data, we need to consider data governance, privacy and security principles. Privacy is extremely important in all medical research requiring researchers to securely store data, rigously control access and use deidentified data if possible.

If the data is being accessed from a database, we need to ensure that we control the access and logins. Flat files such as csv, xlsx or .dta files should be stored securely and not shared through unsecure channels. Ideally, these files should be contained in a secure platform such as SeRP giving an analysis platfrom to researchers where data custodians have control on what data and files enters and leaves. Some researchers like to create passwords to these files. I am giving an example using file encryption in R.

```r
# Start by generating ssh keys if you don't have these already. Create and secure password
# password: qcifiscool
genkeys()
```

```r
# encrypt the file and securely store the origtinal file using the ssh keys and password.
liver_tumour_file <- file.path(ROOT, 'data', 'Case_study_data.csv')
encrypt_file(liver_tumour_file)

# we create two new directories to store this data in encrypted and backup
if(!dir.exists(file.path(ROOT, 'data', 'encrypted'))) dir.create(file.path(ROOT, 'data', '

if(!dir.exists(file.path(ROOT, 'data', 'backup'))) dir.create(file.path(ROOT, 'data', 'bac

# now we move the file to another folder just to keep the encrypted and unencrypted files
file.copy(
  from = file.path(ROOT, 'data', 'Case_study_data.csv.encryptr.bin'),
  to = file.path(ROOT, 'data', 'encrypted','Case_study_data.csv.encryptr.bin'),
)

file.copy(
```

```
  from = file.path(ROOT, 'data', 'Case_study_data.csv'),
  to = file.path(ROOT, 'data', 'backup','Case_study_data.csv'),
)
```

Now we can load the data into memory. The below example shows how we might decrypt the file if we have the ssh keys and password. I have also included an non-encrypted version which runs by default for demonstration.

```
USE_ENCRYPTED <- FALSE

if(USE_ENCRYPTED){

  # decrypt data file prior to use
  encrptyed_file_path <- file.path(ROOT, 'data', 'encrypted',
                                   "Case_study_data.csv.encryptr.bin")
  decrypt_file(
    encrptyed_file_path,
    file_name = file.path(ROOT, 'data', 'encrypted', 'Case_study_data.csv')
  )

  liver_tumor_file <- file.path(ROOT, 'data', 'encrypted', 'Case_study_data.csv')

} else {
  # use an unencrypted file if working in a secure environment such as SeRP
  liver_tumour_file <- file.path(ROOT, 'data', 'backup', 'Case_study_data.csv')
}

liver_tumour <- read_csv(liver_tumour_file) %>%
  mutate(
    # initial cleaning steps (which I apply here to save time)

    # ensure that date variables are correctly formatted (in this case as dmy).
    # we can use lubridate package for ease
    across(c(Date_of_Death, Date_of_surgery, Last_follow.up), \(x) dmy(x)),

    # now we calculate a time until death (in days)
    Time_until_Death = as.numeric(Last_follow.up - Date_of_surgery)

  )
```

```
Rows: 60 Columns: 16
-- Column specification --------------------------------------------------------
```

```
Delimiter: ","
chr  (5): Patient_ID, Gender, Date_of_Death, Date_of_surgery, Last_follow.up
dbl (11): Age, Height, Weight, BMI, Obese, Smoking.Status, Alcohol.Consumpti...

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.
```

```
head(liver_tumour)
```

```
# A tibble: 6 x 17
  Patient_ID Gender   Age Height Weight   BMI Obese Smoking.Status
  <chr>      <chr>  <dbl>  <dbl>  <dbl> <dbl> <dbl>          <dbl>
1 P54        F         86    180     51  15.7     0              2
2 P57        F         89    194     62  16.5     0              1
3 P46        F         72    184     57  16.8     0              1
4 P41        F         83    183     61  18.2     0              1
5 P7         F         75    157     46  18.7     0              2
6 P52        F         66    177     60  19.2     0              1
# i 9 more variables: Alcohol.Consumption <dbl>, Treatment <dbl>,
#   Multiple_tumors <dbl>, Diam_Tumor <dbl>, Death_status <dbl>,
#   Date_of_Death <date>, Date_of_surgery <date>, Last_follow.up <date>,
#   Time_until_Death <dbl>
```

First thing to examine is whether there are duplicated `Patient_ID` in this data. In this particular dataset we are examining each `Patient_ID` to be unique as person can only have this outcome once. In other types of research questions such as longitudinal or multilevel data structures these types of duplicates are extended and modelled using more sophisicated methods. Even in survival analysis it may be possible to have stratified outcomes such as multiple outcome types such as stages of a disease or the patient having the outcome multiple times at different locations. Nevertheless, we see for this data set that there are no duplicated rows for `Patient_ID` so we can go ahead with the rest of our analysis.

```
liver_tumour %>%
  filter(duplicated(Patient_ID))
```

```
# A tibble: 0 x 17
# i 17 variables: Patient_ID <chr>, Gender <chr>, Age <dbl>, Height <dbl>,
#   Weight <dbl>, BMI <dbl>, Obese <dbl>, Smoking.Status <dbl>,
#   Alcohol.Consumption <dbl>, Treatment <dbl>, Multiple_tumors <dbl>,
#   Diam_Tumor <dbl>, Death_status <dbl>, Date_of_Death <date>,
#   Date_of_surgery <date>, Last_follow.up <date>, Time_until_Death <dbl>
```

Categorical Predictors

Let's begin our description of the data with categorical predictors `Treatment`, `Obese`, `Smoking.Status`, `Alcohol.Consumption` and `Multiple_tumors`.

```
cat_predictors <-
  liver_tumour %>%
    transmute(
      Patient_ID,
      Gender,
      across(
        c(
          Treatment,
          Multiple_tumors,
          Obese,
          Smoking.Status,
          Alcohol.Consumption
        ),
        \(x) as.character(x)
      )
    ) %>%
    pivot_longer(
      cols = -Patient_ID,
      names_to = 'Predictor',
      values_to = 'Value'
    ) %>%
  group_by(Predictor, Value) %>%
  summarise(
    n = n()
  ) %>%
  group_by(Predictor) %>%
  mutate(
    percent = n/(sum(n))*100,
    label = glue('{n} ({round(percent, 1)})')
  ) %>%
  nest() %>%
  mutate(
    plot = pmap(
      list(Predictor, data),

      .f = function(predictor, df){
```

```
        df %>%
          ggplot(
            aes(x = factor(Value), y = n)
          ) +
            geom_text(aes(label = label), nudge_y = 10) +
            geom_bar(
              stat = 'identity',
              fill = 'blue'
            ) +
            labs(x = predictor) +
            scale_y_continuous(limits = c(0, 60)) +
            theme_bw()

      }
    )
  )
```

```
`summarise()` has grouped output by 'Predictor'. You can override using the
`.groups` argument.
```

```
cowplot::plot_grid(
  plotlist = cat_predictors$plot,
  ncol = 2
)
```

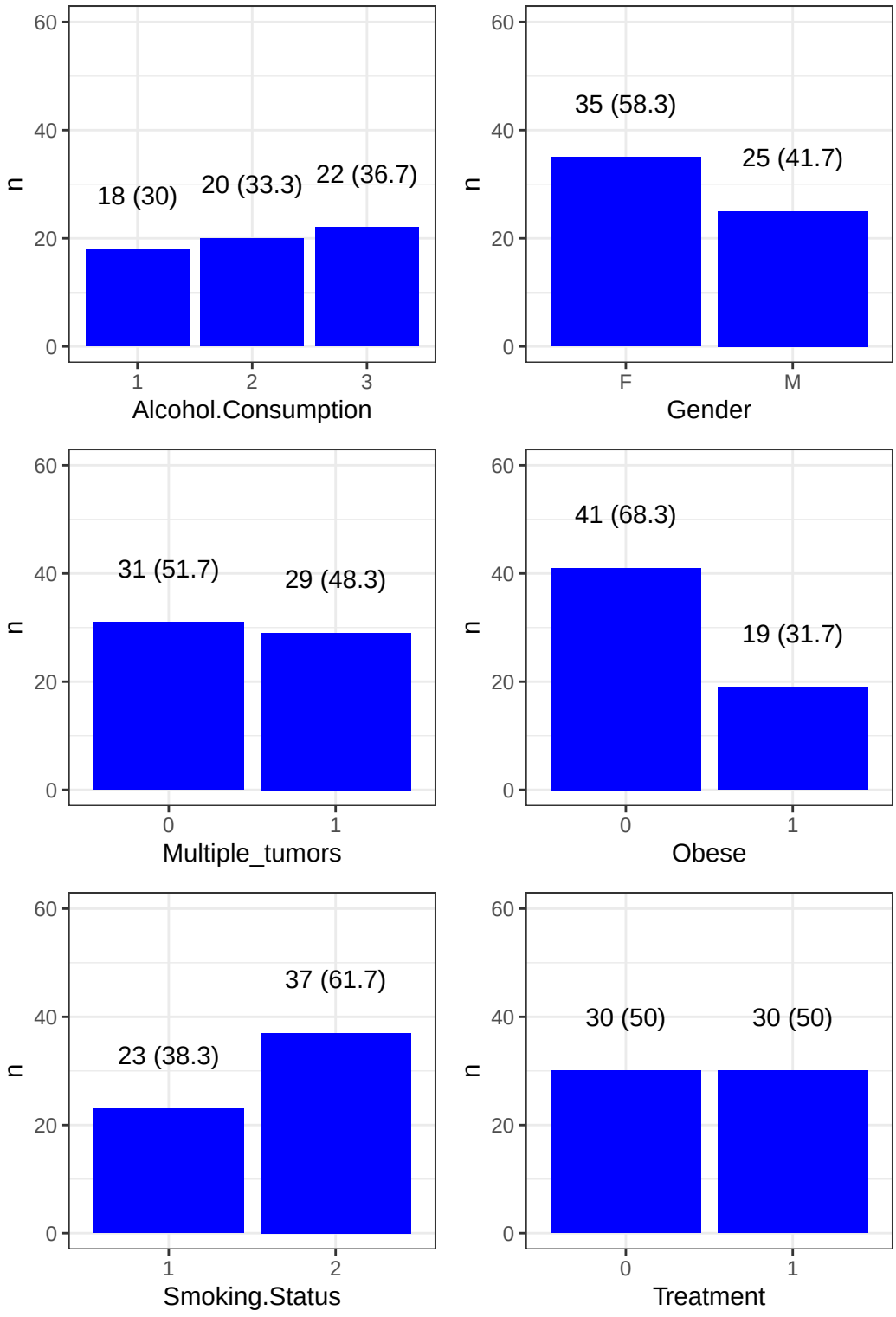


Figure 1: Distribution of Categorical Predictors

The most notable aspects is that alcohol consumption across non-drinker, occasional drinker and regular drinker is fairly evenly distributed with no-drinkers consisting (30%), occasional drinkers (33.3%) and the highest regular drinkers (36.7%). Having multiple tumors is also quite evenly split between those with no multiple tumors (Multiple_tumors = 0, 51.7%) and those with multiple tumors (Multiple_tumors = 1, 48.3%). Smokers are also more common in this dataset with 61.7% reporting being smokers whereas the majority are not obese (68.3%). There are more female patients (58.3%) in this dataset than male patients.

After initial investigation, we might consider adding factor labels to the categorical variables so they are easier to intrepret within figures and tables. We need to check though that each one is correctly refactored before changing these new factor versions of these predictors.

```
# A tibble: 9 x 4
  Predictor            Code Label                 n
  <chr>               <dbl> <fct>             <int>
1 Treatment               0 No                   30
2 Treatment               1 Yes                  30
3 Smoking.Status          1 non-smoker           23
4 Smoking.Status          2 smoker               37
5 Alcohol.Consumption     1 don't drink          18
6 Alcohol.Consumption     2 occasional drinker   20
7 Alcohol.Consumption     3 regular drinker      22
8 Multiple_tumors         0 No                   31
9 Multiple_tumors         1 Yes                  29
```

We should be be pretty confident we have recoded these correctly and can replace these variables if desired. This can also help us develop a data dictionary if necessary.

```
# echo: true

liver_tumour_refactored <-
  liver_tumour %>%
    select(
      -Treatment, -Obese, -Alcohol.Consumption, -Smoking.Status, -Multiple_tumors
    ) %>%
    rename(
      Treatment = Treatment.factor,
      Obese = Obese.factor,
      Alcohol.Consumption = Alcohol.Consumption.factor,
      Smoking.Status = Smoking.Status.factor,
      Multiple_tumors = Multiple_tumors.factor
    )
```

Continuous Predictors

Now we can describe the data in more detail. Below shows one way to describe the continuous predictors using `summarise` with some reshaping of the data used as demonstration. Of course, it is simplier to do `describe` but this is in a nice table format useful for publications and reports.

```
liver_tumor_con.predictors <-
  liver_tumour_refactored %>%
    transmute(
      Patient_ID,
      Age,
      Height,
      Weight,
      BMI,
      Diam_Tumor,
      Time_until_Death
    ) %>%
    pivot_longer(
      cols = -Patient_ID,
      names_to = 'Predictor',
      values_to = 'Value'
    )

liver_tumor_con.predictors %>%
  group_by(Predictor) %>%
  summarise(
    minimum = min(Value, na.rm = T),
    Q1 = quantile(Value, prob = 0.25, na.rm = T),
    median = median(Value, na.rm = T),
    Q3 = quantile(Value, prob = 0.75, na.rm = T),
    max = max(Value),
    missing = sum(is.na(Value))
  ) %>%
  arrange(
    match(
      Predictor,
      c(
        'Age',
```

```
        'Height',
        'Weight',
        'BMI',
        'Diam_Tumor',
        'Time_until_Death'
      )
    )
  )
```

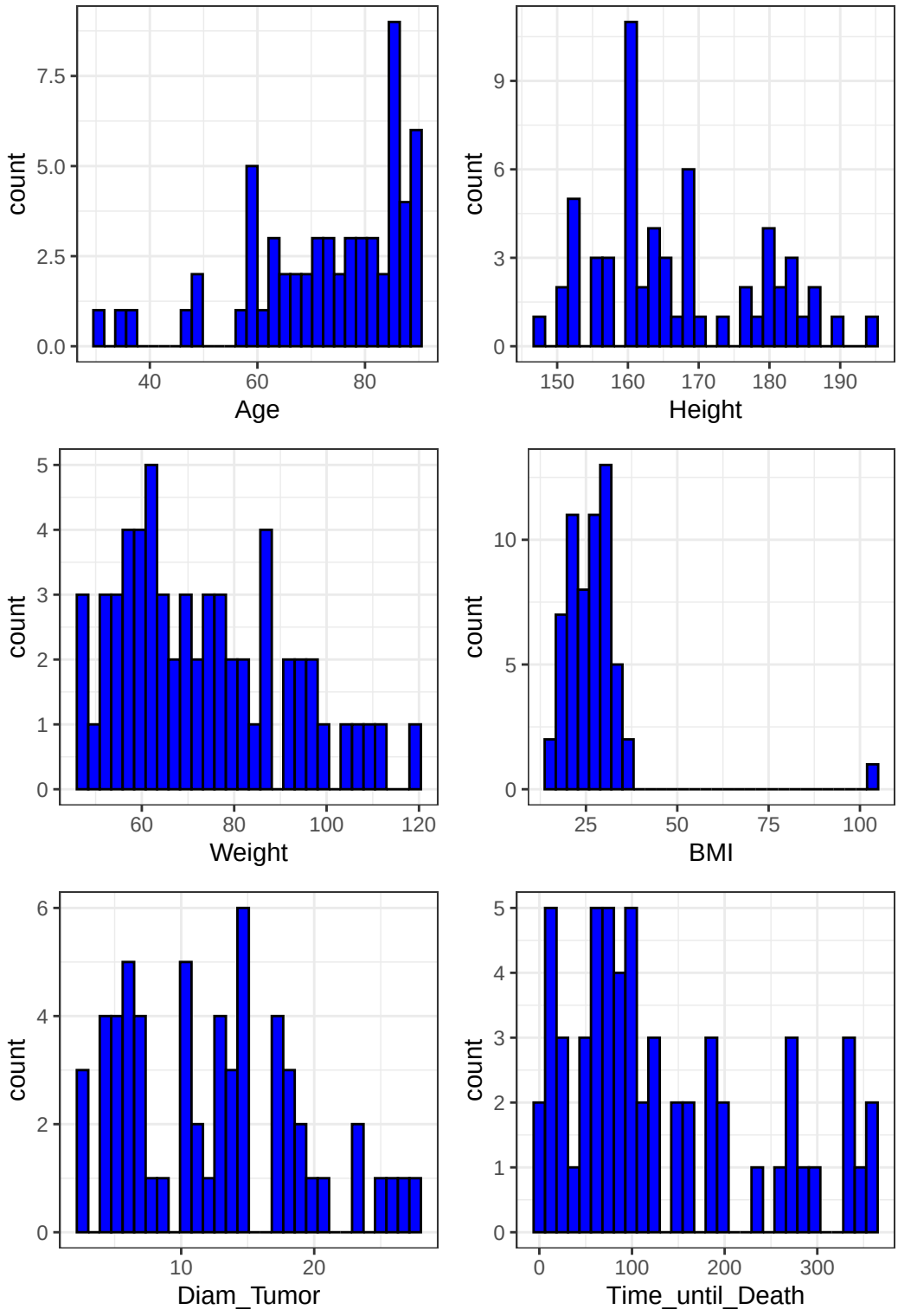
```
# A tibble: 6 x 7
  Predictor        minimum    Q1 median    Q3   max missing
  <chr>              <dbl> <dbl>  <dbl> <dbl> <dbl>   <int>
1 Age                 31    64     76    85.2    90       0
2 Height             147   160    164.  178.    194       0
3 Weight              46    59     69.5  86     118       0
4 BMI                 15.7  21.5   26.7  30.1   104       0
5 Diam_Tumor           3     6     12.5  17      28       0
6 Time_until_Death     4    63     96.5 189.    363       0
```

What we observe in the data is that there are no notable missing values for any of the continuous predictors. The youngest age is 31 whereas the oldest is 90. The descriptive statistics for height and weight seem all reasonable. Nevertheless, the Body Mass Index (BMI) has a maximum value of 104 which does not sound possible. Let's investigate the distributions further and then examine why this value might be happening. We can generate all plots using some nice features of `tidyverse` and `purrr` by looping over each predictor to generate and store each ggplot histogram.

```
liver_tumor_con.predictors.nested <-
  liver_tumor_con.predictors %>%
    group_by(Predictor) %>%
    nest() %>%
    mutate(
      histogram =
        pmap(
          list(Predictor, data),

          .f = function(predictor, df){

            df %>%
              ggplot(
                aes(x = Value)
```

```
                ) +
                geom_histogram(fill = 'blue', linetype = 1, color = 'black') +
                labs(x = predictor) +
                theme_bw()


            })
    )

cowplot::plot_grid(
  plotlist = liver_tumor_con.predictors.nested$histogram,
  ncol = 2
)
```

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


count
Age
Height
Weight
BMI
Diam_Tumor
Time_until_Death

This graphic presentation shows age is right skewed to older patients whereas height and weight appear to be slightly left skewed. Likewise the time until death after surgery appears to be left skewed as well to shorter times which is concerning. We will see what affects this later. Finally, we note that one patient with that very high > 100 BMI. Let's find this patient.

```
liver_tumour_refactored %>%
  filter(BMI > 40) %>%
  transmute(
    Patient_ID,
    Height,
    Weight,
    BMI,
    BMI_recalculated = Weight/(Height/100)^2
  )
```

```
# A tibble: 1 x 5
  Patient_ID Height Weight   BMI BMI_recalculated
  <chr>       <dbl>  <dbl> <dbl>            <dbl>
1 P47           153    105   104             44.9
```

This patient appears to have their BMI incorrectly calculated so we can recalculate (or check the original calculation) just to be sure. I am using the formula provided by CDC https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmiage/page5_1.html#:~:text=With%20the%20metric%20system%2C%20the,multiply%20the%20result%20by%2010%2C000.

```
liver_tumour_cleaned <-
  liver_tumour_refactored %>%
    mutate(
      BMI_recalculated = Weight/(Height/100)^2
    )
```

Now just check that this new formula is similar within one decimal place of the original (with the exception of the incorrectly BMI). We use an arbitrary threshold of 0.1 to find any BMI recalculations that don't match the original. We see that this is only for that one incorrect case.

```
liver_tumour_cleaned %>%
  mutate(
    BMI_difference = round(BMI_recalculated - BMI, 1)
  ) %>%
  filter(
```

```
    abs(BMI_difference) > 0.1
  )
```

```
# A tibble: 1 x 19
  Patient_ID Gender   Age Height Weight   BMI Diam_Tumor Death_status
  <chr>      <chr>  <dbl>  <dbl>  <dbl> <dbl>      <dbl>        <dbl>
1 P47        F         72    153    105   104         11            0
# i 11 more variables: Date_of_Death <date>, Date_of_surgery <date>,
#   Last_follow.up <date>, Time_until_Death <dbl>, Treatment <fct>,
#   Obese <fct>, Smoking.Status <fct>, Alcohol.Consumption <fct>,
#   Multiple_tumors <fct>, BMI_recalculated <dbl>, BMI_difference <dbl>
```

Now we feel confident to replace the BMI column with the recalculated / corrected version. We also bin the Diam_tumor into three bins based on this paper for later use https://onlinelibrary.wiley.com/doi/10.1155/2023/1106975#:~:text=The%20results%20using%20univariate%20analysis,had%20a%20lower%20survival%20rate.

```
liver_tumour_corrected <-
  liver_tumour_cleaned %>%
    select(
      -BMI
    ) %>%
  rename('BMI' = 'BMI_recalculated') %>%
  mutate(
    Diam_Tumor.binned = case_when(
      Diam_Tumor <2 ~ '<2 cm',
      between(Diam_Tumor, 2, 5) ~ '2-5 cm',
      Diam_Tumor > 5 ~ '>5 cm'
    )

  )

head(liver_tumour_corrected)
```

```
# A tibble: 6 x 18
  Patient_ID Gender   Age Height Weight Diam_Tumor Death_status Date_of_Death
  <chr>      <chr>  <dbl>  <dbl>  <dbl>      <dbl>        <dbl> <date>
1 P54        F         86    180     51         13            1 2017-06-30
2 P57        F         89    194     62          3            0 NA
3 P46        F         72    184     57         17            1 2018-01-17
4 P41        F         83    183     61          5            0 NA
```

```
5 P7         F         75    157     46         14            1 2017-10-21
6 P52        F         66    177     60         25            1 2018-03-05
# i 10 more variables: Date_of_surgery <date>, Last_follow.up <date>,
#   Time_until_Death <dbl>, Treatment <fct>, Obese <fct>, Smoking.Status <fct>,
#   Alcohol.Consumption <fct>, Multiple_tumors <fct>, BMI <dbl>,
#   Diam_Tumor.binned <chr>
```

Predictors of death outcomes

Below shows the effect of treatment, presence of multiple tumours and diameter of tumour on percentage of patients who died at their final follow up. While of all these seem to indicate that death is not associated with the treatment, having multiple tumours or tumour diameter based on the percentage of death at the end point. However, this is deceptive and we should look to see what their survival experience over their entire follow up period. Additionally, many of these patients are right censored at their final observation so they may or may not have died.

```
liver_tumour_corrected %>%
  group_by(Treatment) %>%
  summarise(
    number_died = n_distinct(Patient_ID)
  ) %>%
  ungroup() %>%
  mutate(
    `% Died` = number_died/sum(number_died)*100
  ) %>%
  ggplot(
    aes(x = Treatment, y = `% Died`)
  ) +
    geom_bar(stat = 'identity', fill = 'blue') +
    theme_bw()
```

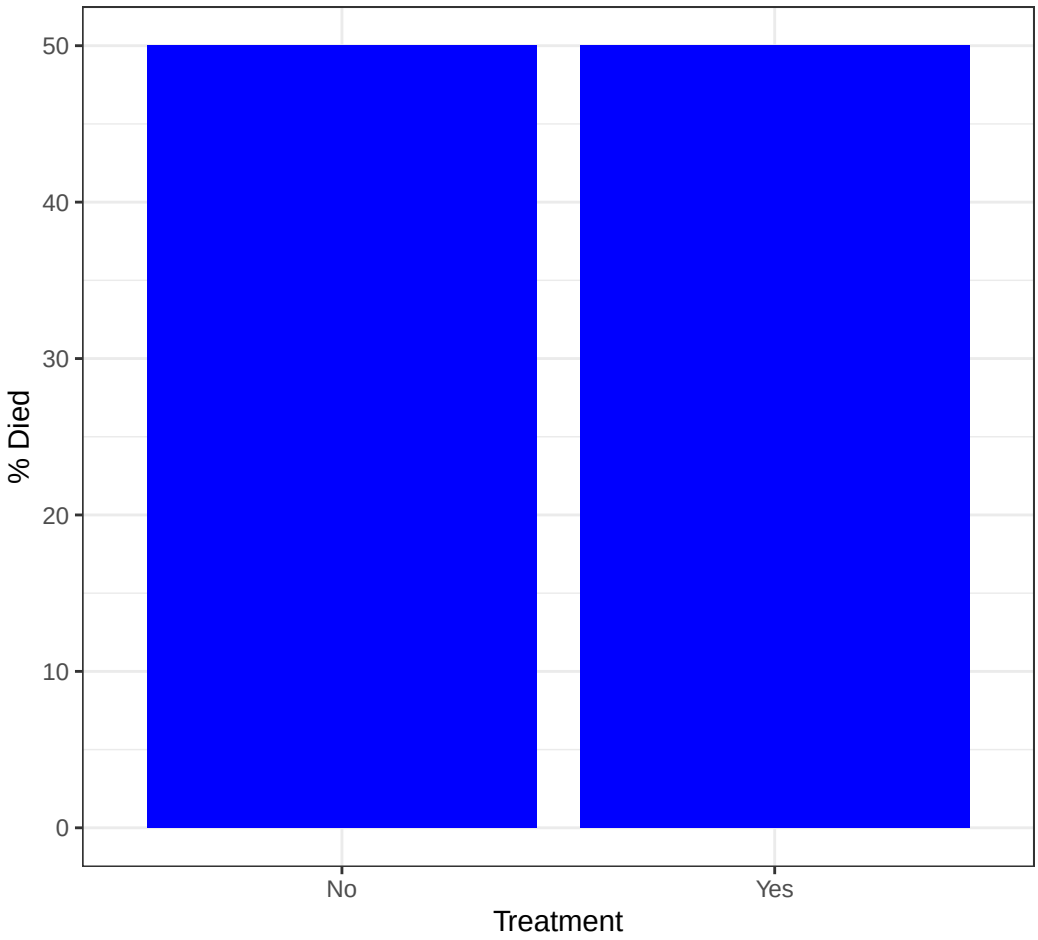


Figure 2: Percentage of death by presence of multiple tumours

```
liver_tumour_corrected %>%
  group_by(Multiple_tumors) %>%
  summarise(
    number_died = n_distinct(Patient_ID)
  ) %>%
  ungroup() %>%
  mutate(
    `% Died` = number_died/sum(number_died)*100
  ) %>%
  ggplot(
    aes(x = Multiple_tumors, y = `% Died`)
  ) +
```

```
    geom_bar(stat = 'identity', fill = 'blue') +
    theme_bw()
```

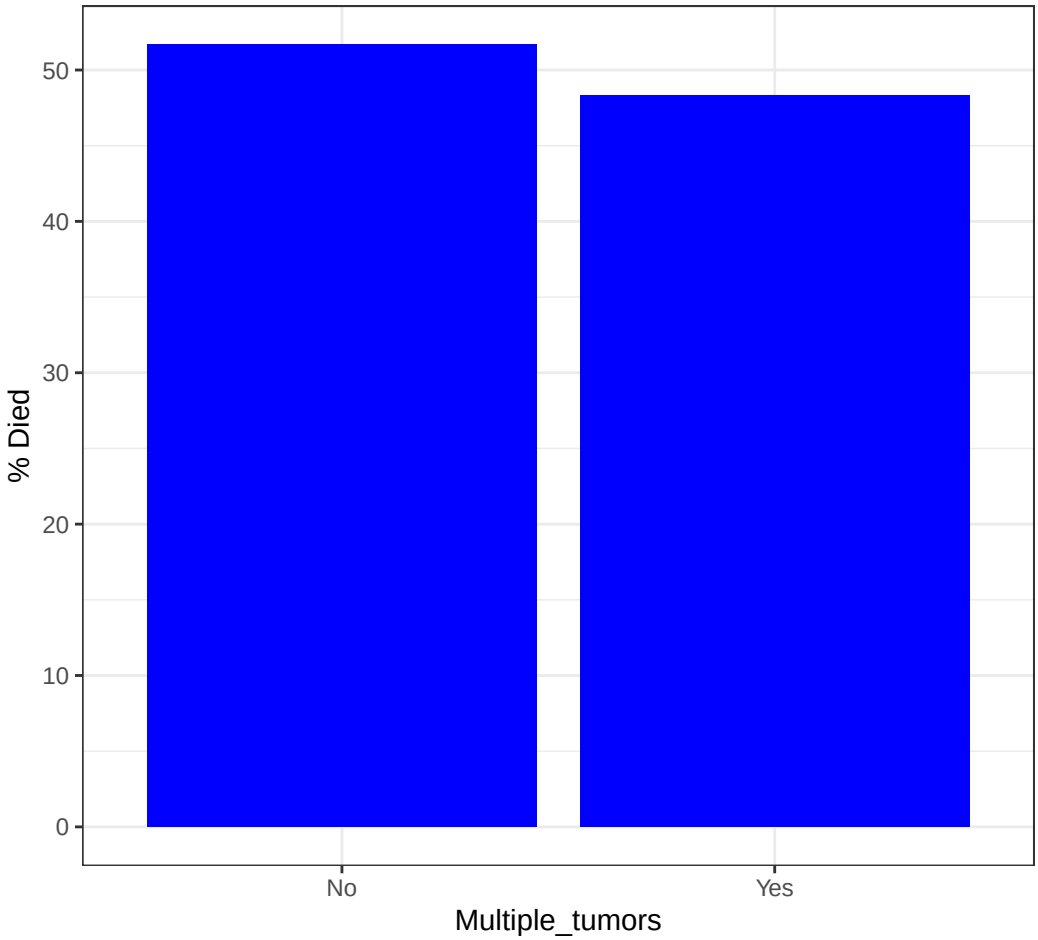
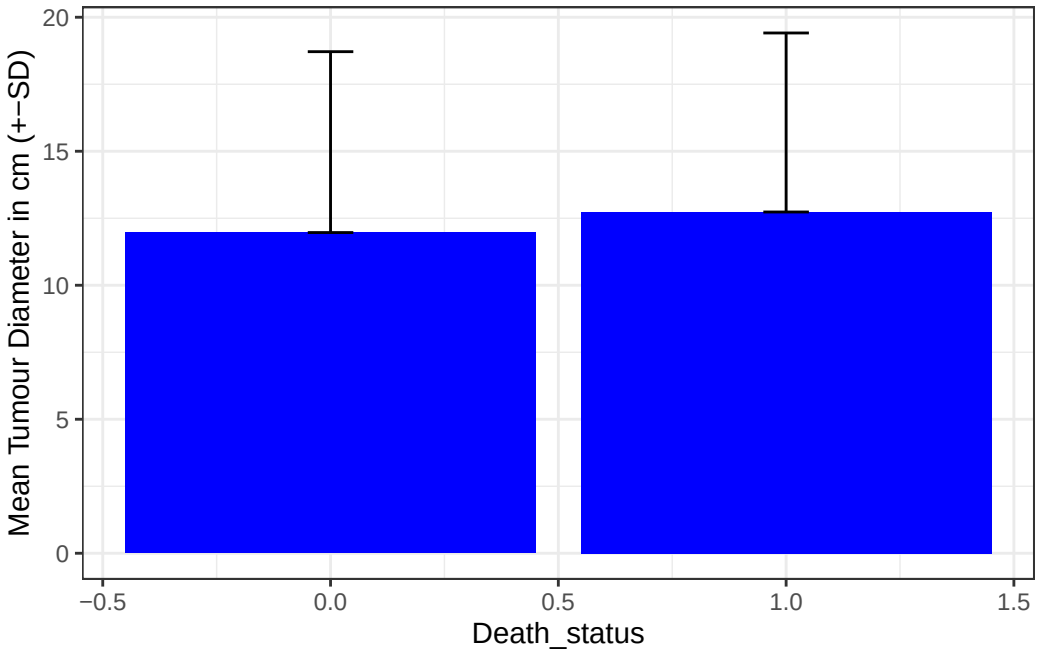


Figure 3: Percentage of death by presence of multiple tumours

```
liver_tumour_corrected %>%
  group_by(Death_status) %>%
  summarise(
    mean_diameter = mean(Diam_Tumor),
    sd_diameter = sd(Diam_Tumor)
  ) %>%
  ungroup() %>%
  ggplot(
```

```
    aes(x = Death_status, y = mean_diameter)
  ) +
    geom_bar(stat = 'identity', fill = 'blue') +
    geom_errorbar(aes(ymin = mean_diameter, ymax = mean_diameter+ sd_diameter), width = 0.
    theme_bw() +
    labs(y = 'Mean Tumour Diameter in cm (+-SD)')
```



Before we do that though we might want to look at the overall survival experience of patients with liver tumors. Let's do this using a Kaplan-Meier estimate

```
surv_fit <- survfit(Surv(Time_until_Death, Death_status) ~ 1, data = liver_tumour_correcte


surv_fit %>%
  ggsurvfit(type = "survival") +
  ggsurvfit::add_risktable() +
  add_confidence_interval() +
  labs(y = 'Prop Survival')
```

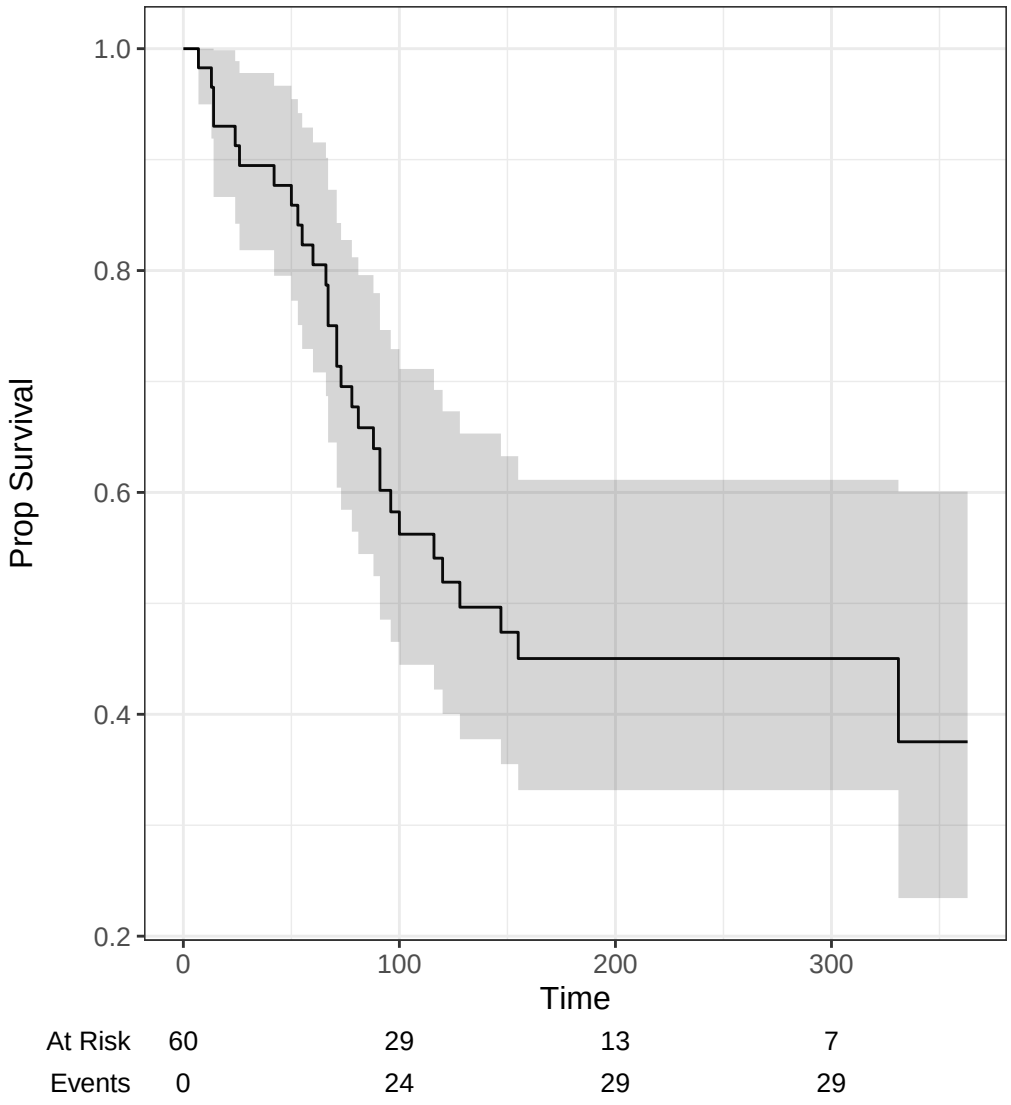


Figure 4: Kaplan-Meier Survival Curves of all patients

From the survival fit, we observe that there 30 events out a possible 60 patients with a median survival time of 128 days post surgery.

```
surv_fit
```

```
Call: survfit(formula = Surv(Time_until_Death, Death_status) ~ 1, data = liver_tumour_correct
```

```
      n events median 0.95LCL 0.95UCL
[1,] 60     30    128      91      NA
```

Below shows the Kaplan-Meier survival curves for each predictor (including Diameter of Tumor binned). The results are much clearer as Treatment clearly has an effect on survival with those without the treatment had survival rates drop rapidly after 100 days post surgery whereas those with the treatment had survival rates remain somewhat flat at 75% after 100 days until their final follow-up. The presence of multiple tumors also had an impact on survival rates where those with multiple tumors also had rapid declines in survival after surgery compared to those without. It is less clear about the impact of wider tumor diameter though higher >5 cm diameter tumors also lead to significantly worse survival rates.

```r
predictors <- c('Treatment', 'Gender', 'Multiple_tumors', 'Diam_Tumor.binned', 'Smoking.St

unadjusted_results <-
  tibble(
    predictor = predictors,
    survformula = map(predictor, ~{
      as.formula(glue('Surv(Time_until_Death, Death_status) ~ {.x}'))
    }),

    survival_plot = map(survformula, ~{
      (.x) %>%
        survfit(data = liver_tumour_corrected) %>%
        ggsurvfit(type = "survival") +
        add_confidence_interval() +
        labs(y = 'Prop Survival')
      }),

    unadjusted_model = map(survformula, ~{
      coxph(.x, data = liver_tumour_corrected)
    }),

    unadjusted_model_tidy = map(unadjusted_model, ~tidy(., exponentiate = T, conf.int = T)

  )

cowplot::plot_grid(
  plotlist = unadjusted_results$survival_plot,
  ncol = 2
)
```

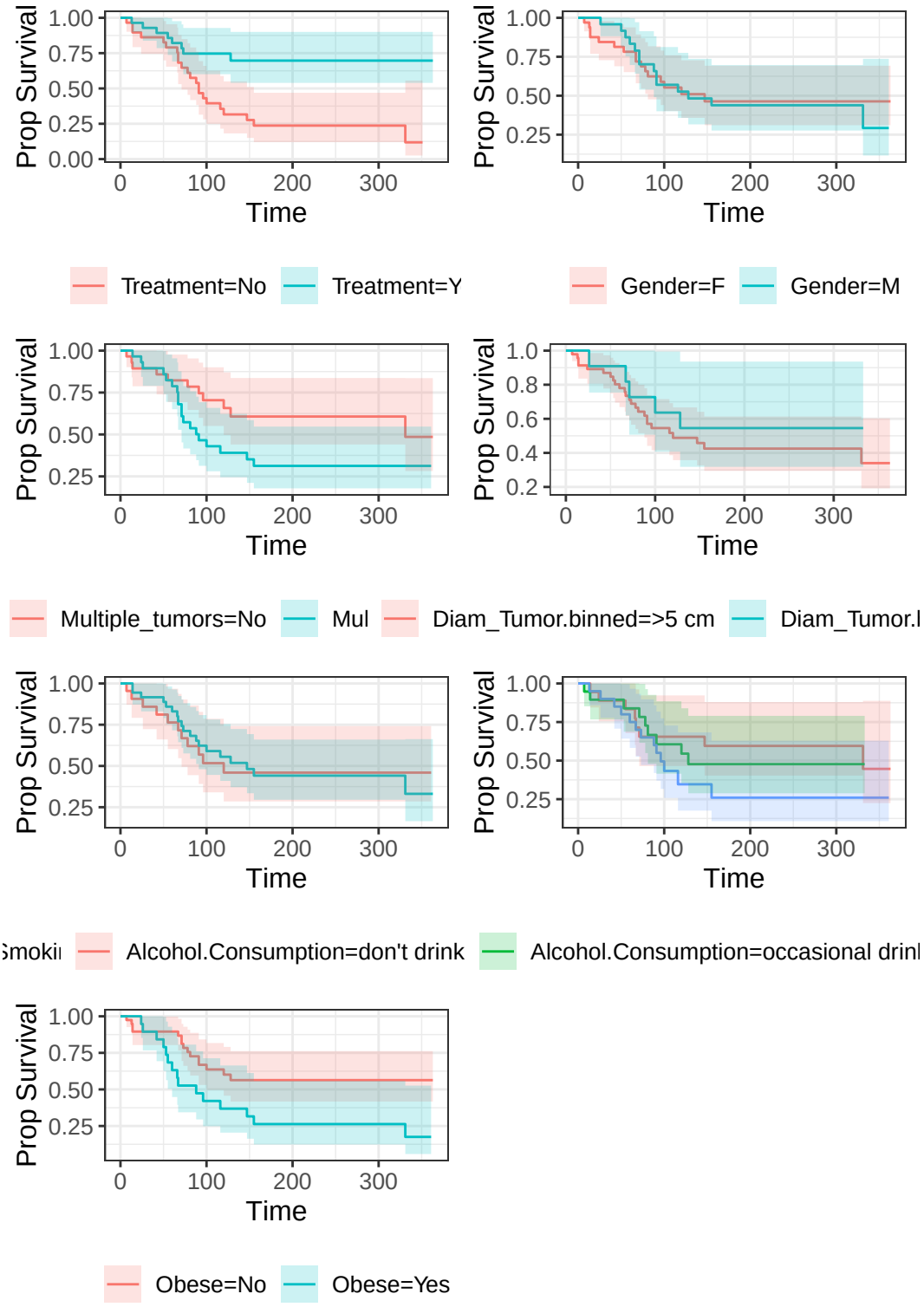


Figure 5: Kaplan Meier Estimates across predictors

Below we note that multiple tumours appears to related to other predictors most notably treatment with higher proportion of multiple tumors in the no treatment group, in males who had treatment and in those who are obese.

```
res <- compareGroups(Multiple_tumors~Treatment + Diam_Tumor + Gender + Age + BMI + Obese +

createTable(res)
```

```
--------Summary descriptives table by 'Multiple_tumors'---------

_____________________________________________________________
                           No          Yes     p.overall
                          N=31        N=29
_____________________________________________________________
Treatment:                                        0.010
    No                 10 (32.3%)  20 (69.0%)
    Yes                21 (67.7%)   9 (31.0%)
Diam_Tumor             12.5 (6.05) 12.2 (7.37)   0.845
Gender:                                           0.021
    F                  23 (74.2%)  12 (41.4%)
    M                   8 (25.8%)  17 (58.6%)
Age                    70.5 (14.3) 75.5 (15.1)   0.188
BMI                    25.0 (5.65) 27.6 (5.77)   0.083
Obese:                                            0.065
    No                 25 (80.6%)  16 (55.2%)
    Yes                 6 (19.4%)  13 (44.8%)
Smoking.Status:                                   0.390
    non-smoker         14 (45.2%)   9 (31.0%)
    smoker             17 (54.8%)  20 (69.0%)
Alcohol.Consumption:                              0.343
    don't drink         8 (25.8%)  10 (34.5%)
    occasional drinker 13 (41.9%)   7 (24.1%)
    regular drinker    10 (32.3%)  12 (41.4%)
_____________________________________________________________
```

Tumor diameter is also found to be wider in the no treated group with a greater proportion of those without treatment having >5 cm tumors (57.1% vs 18.2%).

```
res <- compareGroups(Diam_Tumor.binned~Multiple_tumors + Treatment + Gender + Age + BMI +
```

```
Warning in chisq.test(xx, correct = FALSE): Chi-squared approximation may be
incorrect

Warning in chisq.test(xx, correct = FALSE): Chi-squared approximation may be
incorrect

Warning in chisq.test(xx, correct = FALSE): Chi-squared approximation may be
incorrect

Warning in chisq.test(xx, correct = FALSE): Chi-squared approximation may be
incorrect

Warning in chisq.test(xx, correct = FALSE): Chi-squared approximation may be
incorrect
```

```r
createTable(res)
```

```
--------Summary descriptives table by 'Diam_Tumor.binned'---------

_____________________________________________________________
                          >5 cm      2-5 cm    p.overall
                          N=49        N=11
-------------------------------------------------------------
Multiple_tumors:                                     0.903
    No                 26 (53.1%)   5 (45.5%)
    Yes                23 (46.9%)   6 (54.5%)
Treatment:                                           0.045
    No                 28 (57.1%)   2 (18.2%)
    Yes                21 (42.9%)   9 (81.8%)
Gender:                                              1.000
    F                  29 (59.2%)   6 (54.5%)
    M                  20 (40.8%)   5 (45.5%)
Age                    71.6 (15.4) 78.6 (10.4)   0.081
BMI                    26.3 (5.90) 25.9 (5.64)   0.825
Obese:                                               0.476
    No                 32 (65.3%)   9 (81.8%)
    Yes                17 (34.7%)   2 (18.2%)
Smoking.Status:                                      0.734
    non-smoker         18 (36.7%)   5 (45.5%)
    smoker             31 (63.3%)   6 (54.5%)
Alcohol.Consumption:                                 0.179
```

```
    don't drink        17 (34.7%)   1 (9.09%)
    occasional drinker 14 (28.6%)   6 (54.5%)
    regular drinker    18 (36.7%)   4 (36.4%)
-------------------------------------------------------
```

Multiple tumors and tumor diameter could be confounding our results examining the effect of treatment. We need to statistically control for these effects using the model below.

Modelling survival using Cox Proportional Hazards Regression

In this example, we will create an adjusted model to examine effect of the various predictors on survival using a Cox Proportional Hazards Regression model.

```
predictors <- c(
  'Treatment',
  'Multiple_tumors',
  'Diam_Tumor',
  'Gender',
  'Age',
  'Obese',
  'Smoking.Status',
  'Alcohol_consumption.Status'
)

adjusted_formula = as.formula(
Surv(Time_until_Death, Death_status) ~ Treatment + Multiple_tumors + Diam_Tumor + Gender +

adjusted_cox_model <- coxph(
  adjusted_formula,
  data = liver_tumour_corrected
)

adjusted_cox_model %>%
  tidy(exponentiate = T, conf.int = T) %>%
  select(
    term,
    estimate,
    conf.low,
    conf.high,
    p.value
  )
```

```
# A tibble: 9 x 5
  term                                estimate conf.low conf.high p.value
  <chr>                                  <dbl>    <dbl>     <dbl>   <dbl>
1 TreatmentYes                           0.317    0.128     0.787  0.0133
2 Multiple_tumorsYes                     1.46     0.622     3.43   0.385
3 Diam_Tumor                             1.01     0.940     1.08   0.839
4 GenderM                                0.706    0.281     1.77   0.459
5 Age                                    1.01     0.977     1.04   0.631
6 ObeseYes                               2.28     0.978     5.33   0.0564
7 Smoking.Statussmoker                   0.713    0.303     1.68   0.439
8 Alcohol.Consumptionoccasional drinker  1.85     0.654     5.25   0.246
9 Alcohol.Consumptionregular drinker     1.65     0.600     4.55   0.332
```

The results indicate that Treatment after controlling for all other predictors had a significant effect on survival (HR_adjusted = 0.283, 95% CI: 0.109 to 0.735, p = 0.009). Put differently, those without treatment die at at rate 71.7% higher with the estimate rate being as low as 26.5 times higher to as high as 89.1. According to our unadjusted results, the effect of treatment was actually pretty similar before adjusted for other confounding variables such as tumor diameter. So they may not have had that much impact after all

Interestingly, we note that multiple tumours and tumour diameter do not have a significant effect on survival after accounting other variables such as treatment, smoking, alcohol consumption and obese status. Compare this to the unadjusted hazard ratios where multiple tumours had a significant impact on survival rates with higher death rates for those with multiple tumours.

```
unadjusted_results %>% select(unadjusted_model_tidy) %>% unnest()
```

```
Warning: `cols` is now required when using `unnest()`.
i Please use `cols = c(unadjusted_model_tidy)`.
```

```
# A tibble: 8 x 7
  term                   estimate std.error statistic p.value conf.low conf.high
  <chr>                     <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 TreatmentYes              0.292     0.415    -2.97  0.00300    0.129     0.658
2 GenderM                   1.04      0.367     0.106 0.915      0.507     2.13
3 Multiple_tumorsYes        2.08      0.383     1.91  0.0567     0.980     4.40
4 Diam_Tumor.binned2-5 ~    0.675     0.491    -0.800 0.424      0.258     1.77
5 Smoking.Statussmoker      0.961     0.379    -0.105 0.916      0.457     2.02
6 Alcohol.Consumptionoc~    1.22      0.490     0.408 0.683      0.467     3.19
7 Alcohol.Consumptionre~    1.88      0.458     1.38  0.168      0.766     4.62
8 ObeseYes                  2.32      0.367     2.29  0.0221     1.13      4.76
```

In both unadjusted and adjusted models, obesity has a significant impact on survival with those who are obese having significantly higher hazard / death rates (HR_adjusted = 2.32, 95% CI: 1.13 to 4.76).

Assessing model fit

NOTE: normally we should be doing this step while we are developing the models but we are doing it here for demonstration.

The proportional hazards assumption of Cox regression assumes that the effect of any predictor remains constant over time. In other words, the hazard ratio of 0.283 where those who were treated have 71% higher average survival rates than those without the treatment regardless whether we look at their survival rates just a few weeks after post surgery or several years after surgery. In other words, the benefits of treatment have the same impact regardless when they are assessed. The impact of some predictors on survival may have dependent on time. That is, they have greater impact early on or perhaps later on. For instance, there may be little impact of some treatments early on in liver cancer but does impact the survival rates of those in late stage cancer. This is known as a time-dependent covariate and requires different types of statistical models to describe. In recent years, more sophisticated models such as joint models have been used to model how a predictor changes over time and how this change impacts survival e.g. blood pressure, growth or decline in tumours, etc.

So how do we assess whether this assumption is met. While there are different approaches basically we examine whether the . Residuals are leftovers from any statistical model we fit to data. If the model we fitted is a pretty good explanation of the data, then these residuals (left overs) will contain no useful information that hasn't been captured by the model. In the case of Cox regression, we use approximate Schoenfeld residuals to assess how much proportional hazards have captured the essence of the data.

According to goodness of fit tests (we are only using time here though we can also use log(time), rank(time), etc to better assess this assumption. Nevertheless, we note that treatment is significant indicating that there is some relationship between treatment and time on hazard rates violating our proportional hazards assumption.

```
# echo: true

adjusted_fit_diagnostics <- cox.zph(adjusted_cox_model)

adjusted_fit_diagnostics
```

```
                  chisq df     p
Treatment        4.8405  1 0.028
Multiple_tumors  0.4064  1 0.524
```

```
Diam_Tumor           0.0752  1 0.784
Gender               1.0245  1 0.311
Age                  1.0215  1 0.312
Obese                0.0830  1 0.773
Smoking.Status       1.2957  1 0.255
Alcohol.Consumption 0.9589  2 0.619
GLOBAL               8.9287  9 0.444
```

Examining our residual plot, we observe that the Schoenfeld residuals for our hazard rates (listed as beta(t) here) are not randomly scattered around 0 and appear to become more negative as time goes along. In other words, the difference in survival becomes greater the longer the patients are followed up. This makes sense as we saw in the Kaplan Meier curves where treated patients had a flat survival curve whereas those without treatment kept declining. This suggests that our treatment is better than estimated here by the Cox model but requires some other extensions such as separating treatment into treatment at different time points (time varying predictor).

```r
plot(adjusted_fit_diagnostics[1]) # plot curves

abline (h = coef(adjusted_fit_diagnostics)[1], lty = "dotted", lwd = 1)
```