Survival and Longitudinal Data Analysis – Project

Jaad Belhouari (jaad.belhouari@gmail.com)

December 6th 2024

```
setwd("C:/Users/jaadb/Documents/Cours/M2DS/Survival/Projet/code")
```

A. Exploratory Data Analysis (EDA)

Question 1 : Data loading and summary

```
# Load the data
hiv_data <- read.csv("hiv.csv")

# Display the structure of the dataset
str(hiv_data)
```

```
## 'data.frame':    1405 obs. of  9 variables:
##  $ subject       : int  1 1 1 2 2 2 2 3 3 3 ...
##  $ time          : num  17 17 17 19 19 ...
##  $ death         : int  0 0 0 0 0 0 0 1 1 1 ...
##  $ cd4           : num  10.68 8.43 9.43 6.32 8.12 ...
##  $ time_obs      : int  0 6 12 0 6 12 18 0 2 6 ...
##  $ treatment     : chr  "ddC" "ddC" "ddC" "ddI" ...
##  $ sex           : chr  "male" "male" "male" "male" ...
##  $ prev_infection: chr  "AIDS" "AIDS" "AIDS" "noAIDS" ...
##  $ azt           : chr  "intolerance" "intolerance" "intolerance" "intolerance" ...
```

```
# Preview the first few rows
head(hiv_data)
```

```
##   subject  time death       cd4 time_obs treatment  sex prev_infection
## 1       1 16.97     0 10.677078        0       ddC male           AIDS
## 2       1 16.97     0  8.426150        6       ddC male           AIDS
## 3       1 16.97     0  9.433981       12       ddC male           AIDS
## 4       2 19.00     0  6.324555        0       ddI male         noAIDS
## 5       2 19.00     0  8.124038        6       ddI male         noAIDS
## 6       2 19.00     0  4.582576       12       ddI male         noAIDS
##           azt
## 1 intolerance
## 2 intolerance
## 3 intolerance
```

```
## 4 intolerance
## 5 intolerance
## 6 intolerance
```

Question 2 : Check if data contains NA or duplicate lines

```
# Check for missing values
na_counts <- colSums(is.na(hiv_data))
na_counts
```

```
##        subject           time          death            cd4       time_obs
##              0              0              0              0              0
##      treatment            sex prev_infection            azt
##              0              0              0              0
```

```
# Check for duplicate rows
duplicates <- nrow(hiv_data) - nrow(distinct(hiv_data))
duplicates
```

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

Thus, there is no missing values in our data.

```
# unique(hiv_data$subject)
```

Modifying our data with a start stop format as the variable 'cd4' vary over time for each patient

```
df <- data.frame(
  subject = hiv_data$subject,
  start = hiv_data$time_obs,
  stop = c(hiv_data$time_obs[-1], NA),
  time = hiv_data$time
)

for (i in unique(df$subject)) {
  last_index <- max(which(df$subject == i))
  df$stop[last_index] <- df$time[last_index]
}

hiv_data$time_obs = as.double(df$start)
hiv_data$stop <- df$stop
hiv_data <- hiv_data %>%
  rename(start = time_obs) %>%
  select(subject, time, death, cd4, start, stop, treatment, sex, prev_infection, azt)  # 'start' direct
head(hiv_data)
```

```
##   subject  time death       cd4 start  stop treatment  sex prev_infection
## 1       1 16.97     0 10.677078     0  6.00       ddC male           AIDS
## 2       1 16.97     0  8.426150     6 12.00       ddC male           AIDS
## 3       1 16.97     0  9.433981    12 16.97       ddC male           AIDS
```

```
## 4       2 19.00     0  6.324555     0  6.00       ddI male         noAIDS
## 5       2 19.00     0  8.124038     6 12.00       ddI male         noAIDS
## 6       2 19.00     0  4.582576    12 18.00       ddI male         noAIDS
##           azt
## 1 intolerance
## 2 intolerance
## 3 intolerance
## 4 intolerance
## 5 intolerance
## 6 intolerance
```

Converting categorical variables into factors

```r
# Convert variables to factors if necessary
hiv_data$death <- as.factor(hiv_data$death)
hiv_data$treatment <- as.factor(hiv_data$treatment)
hiv_data$sex <- as.factor(hiv_data$sex)
hiv_data$prev_infection <- as.factor(hiv_data$prev_infection)
hiv_data$azt <- as.factor(hiv_data$azt)

# Verify the changes
str(hiv_data)
```

```
## 'data.frame':    1405 obs. of  10 variables:
##  $ subject       : int  1 1 1 2 2 2 2 3 3 3 ...
##  $ time          : num  17 17 17 19 19 ...
##  $ death         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
##  $ cd4           : num  10.68 8.43 9.43 6.32 8.12 ...
##  $ start         : num  0 6 12 0 6 12 18 0 2 6 ...
##  $ stop          : num  6 12 17 6 12 ...
##  $ treatment     : Factor w/ 2 levels "ddC","ddI": 1 1 1 2 2 2 2 2 2 2 ...
##  $ sex           : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 1 1 1 ...
##  $ prev_infection: Factor w/ 2 levels "AIDS","noAIDS": 1 1 1 2 2 2 2 1 1 1 ...
##  $ azt           : Factor w/ 2 levels "failure","intolerance": 2 2 2 2 2 2 2 2 2 2 ...
```

Question 3 : Data Summurization and Vizualization

```r
# Summarize categorical variables
summary(hiv_data$treatment)
```

```
## ddC ddI
## 717 688
```

```r
summary(hiv_data$sex)
```

```
## female   male
##    117   1288
```

```r
summary(hiv_data$prev_infection)
```

```
##   AIDS noAIDS
##    863    542
```

```r
summary(hiv_data$azt)
```

```
##     failure intolerance
##         491         914
```

```r
# Summarize continuous variables
summary(hiv_data$time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.47   12.23   14.07   13.89   17.00   21.40
```

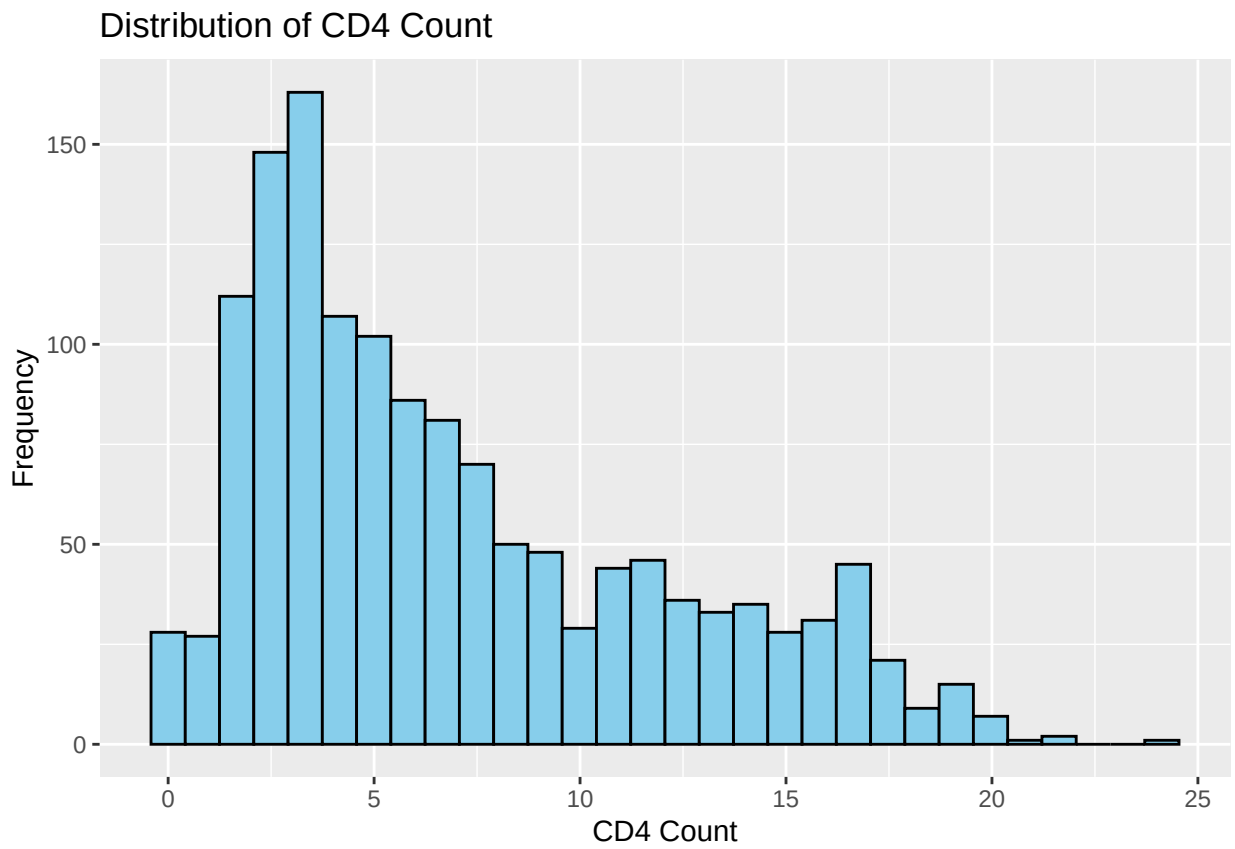
```r
summary(hiv_data$cd4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.162   5.477   7.023  10.440  24.125
```

Continious data variable

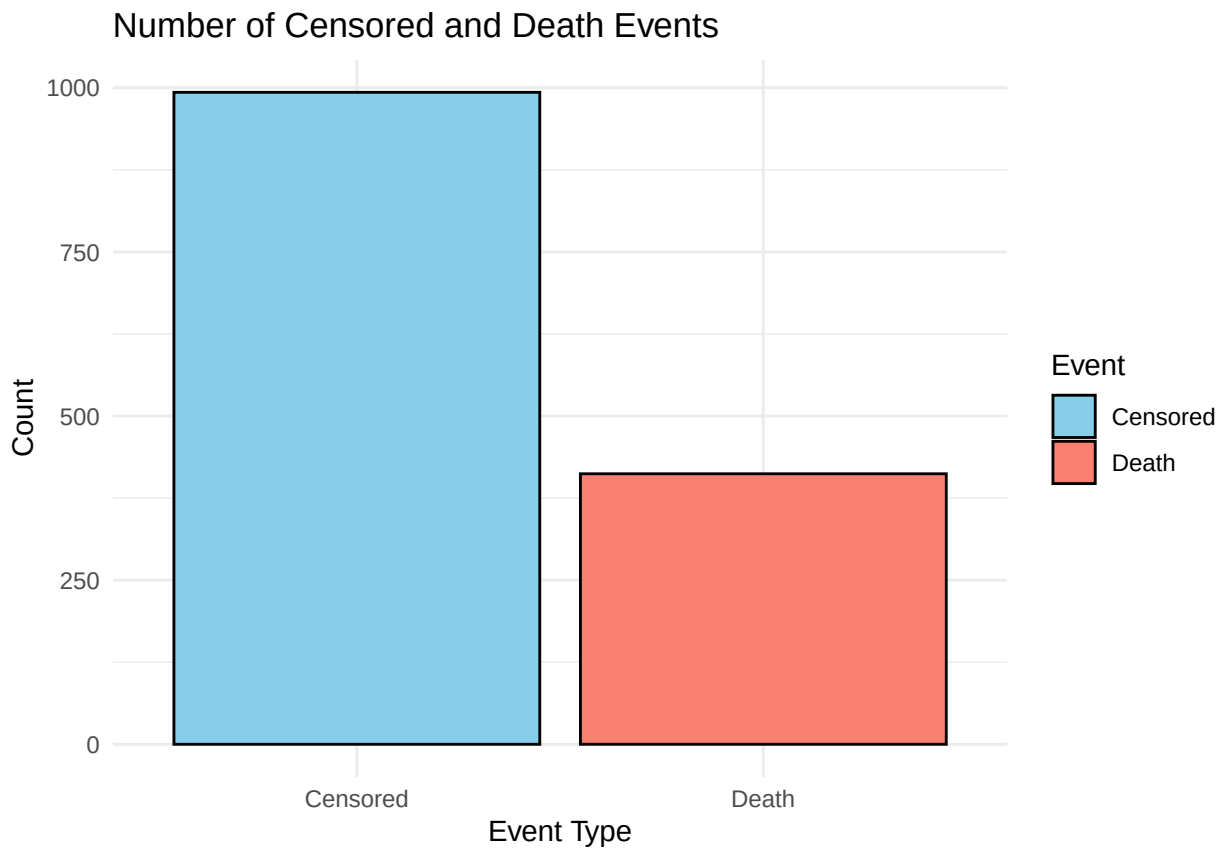
Let's visualize the distribution of our continuous variable 'dc4', regarding the values taken.

```r
ggplot(hiv_data, aes(x = cd4)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  labs(title = "Distribution of CD4 Count", x = "CD4 Count", y = "Frequency")
```



Counting the number of censored and death person in our data

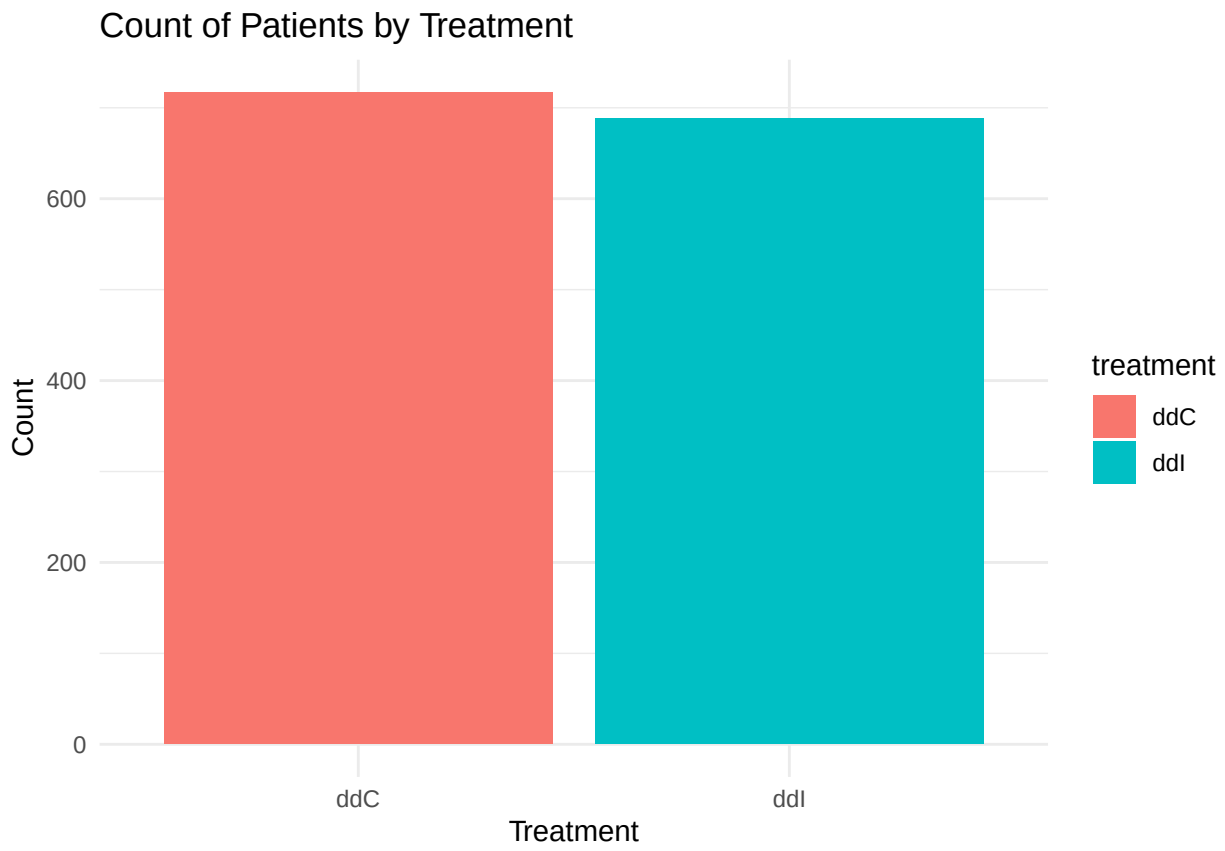
```
# Count the number of censored and death events
event_counts <- table(hiv_data$death)

# Convert to a data frame for plotting
event_counts_df <- as.data.frame(event_counts)
colnames(event_counts_df) <- c("Event", "Count")

# Replace 0 and 1 with meaningful labels
event_counts_df$Event <- ifelse(event_counts_df$Event == 0, "Censored", "Death")

# Load ggplot2 for plotting
library(ggplot2)

# Create the bar plot
ggplot(event_counts_df, aes(x = Event, y = Count, fill = Event)) +
  geom_bar(stat = "identity", color = "black") +
  labs(title = "Number of Censored and Death Events",
       x = "Event Type",
       y = "Count") +
  scale_fill_manual(values = c("Censored" = "skyblue", "Death" = "salmon")) +
  theme_minimal()
```



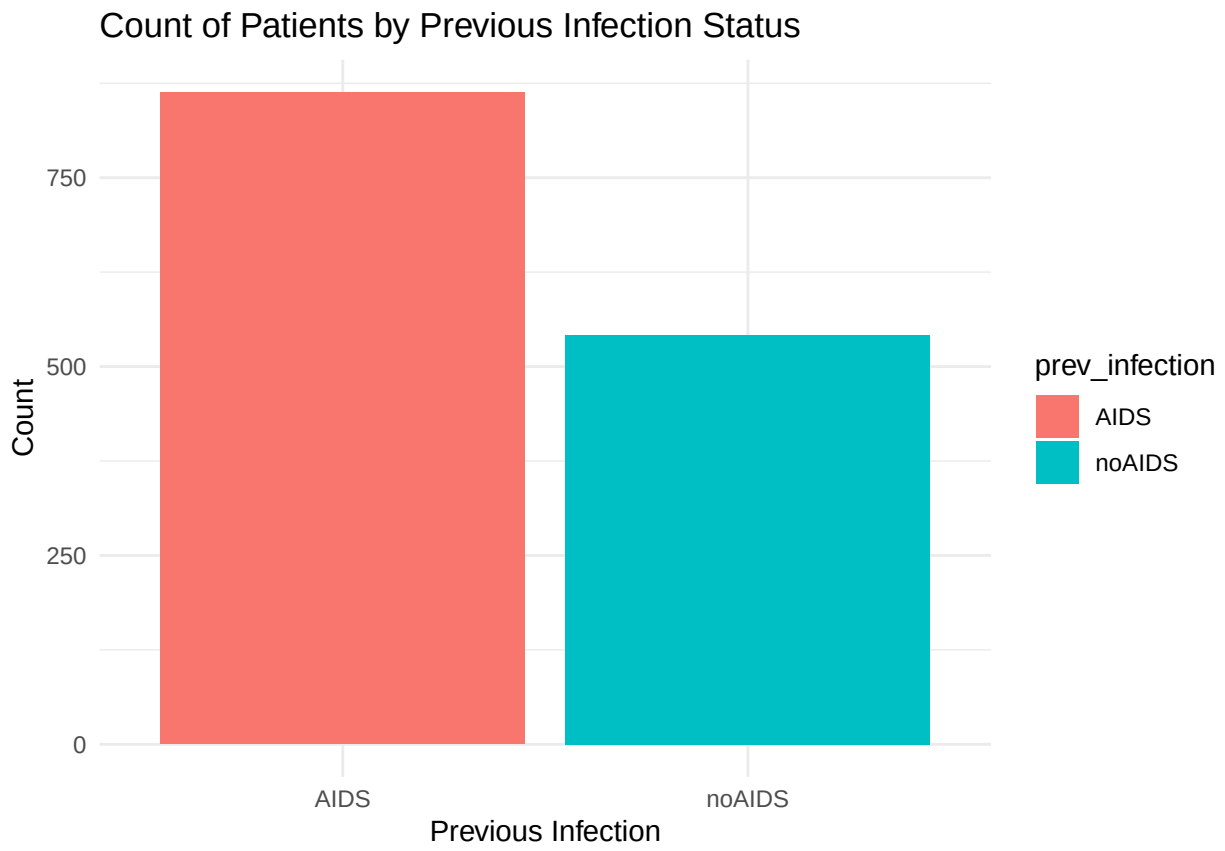
Categorical variable

Let's now count our categorical variable regarding the different groups.

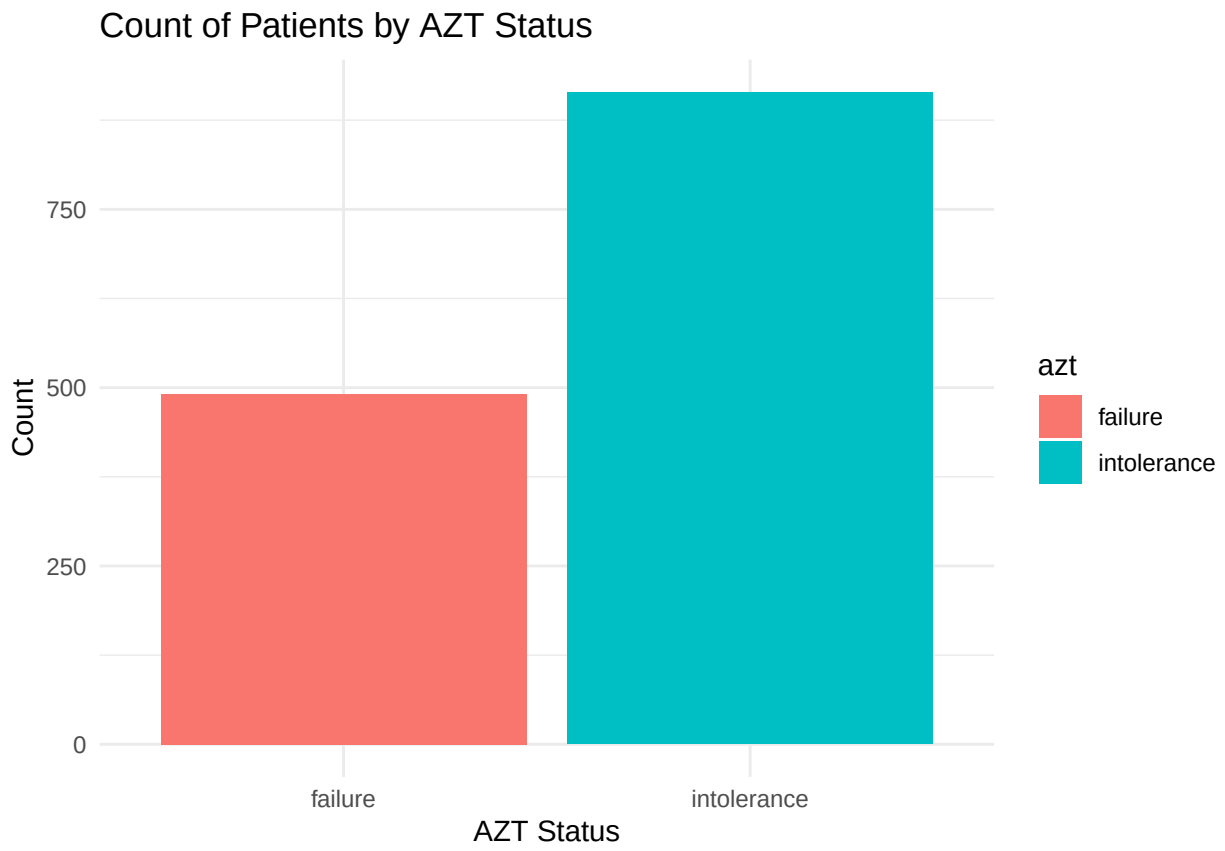
```
# Treatment
ggplot(hiv_data, aes(x = treatment, fill = treatment)) +
  geom_bar() +
  labs(title = "Count of Patients by Treatment",
       x = "Treatment",
       y = "Count") +
  theme_minimal()
```



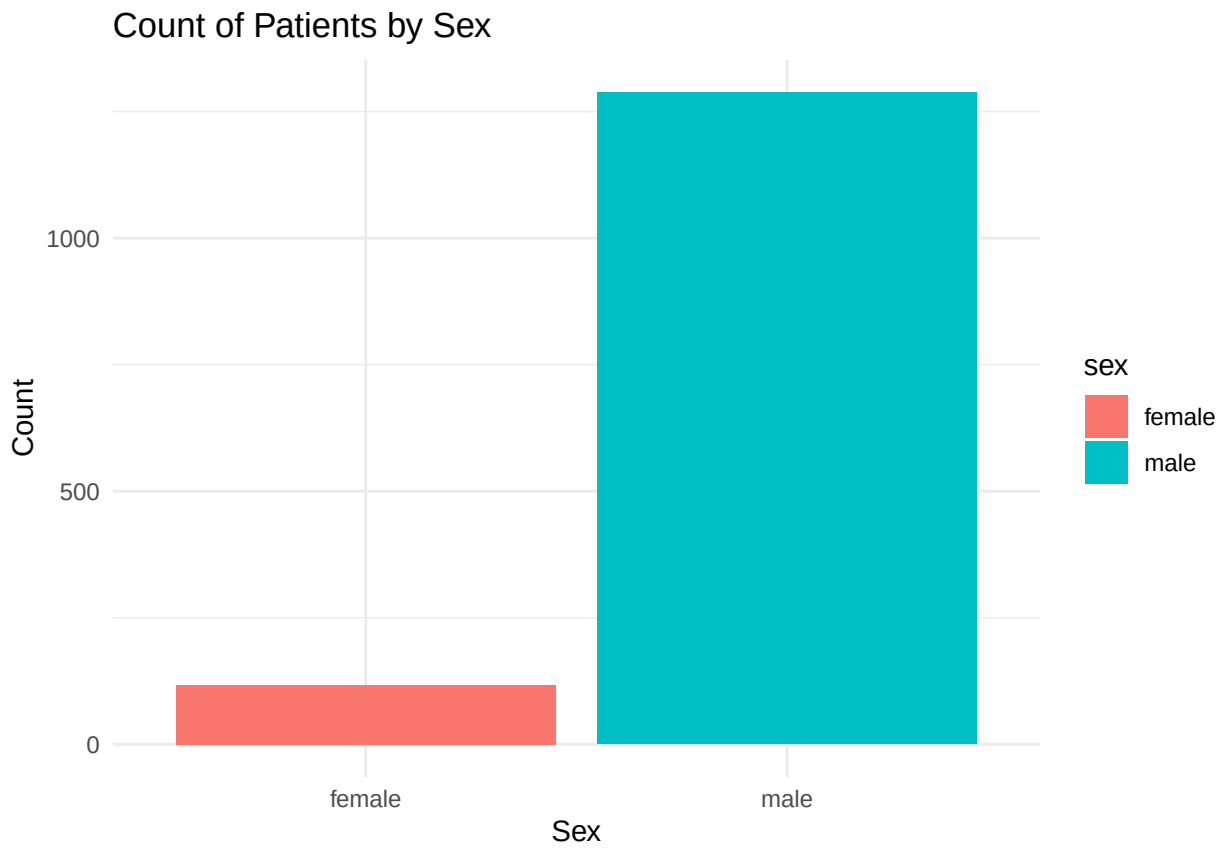

```
# Previous Infection
ggplot(hiv_data, aes(x = prev_infection, fill = prev_infection)) +
  geom_bar() +
  labs(title = "Count of Patients by Previous Infection Status",
       x = "Previous Infection",
       y = "Count") +
  theme_minimal()
```




```
# AZT Status
ggplot(hiv_data, aes(x = azt, fill = azt)) +
  geom_bar() +
  labs(title = "Count of Patients by AZT Status",
       x = "AZT Status",
       y = "Count") +
  theme_minimal()
```

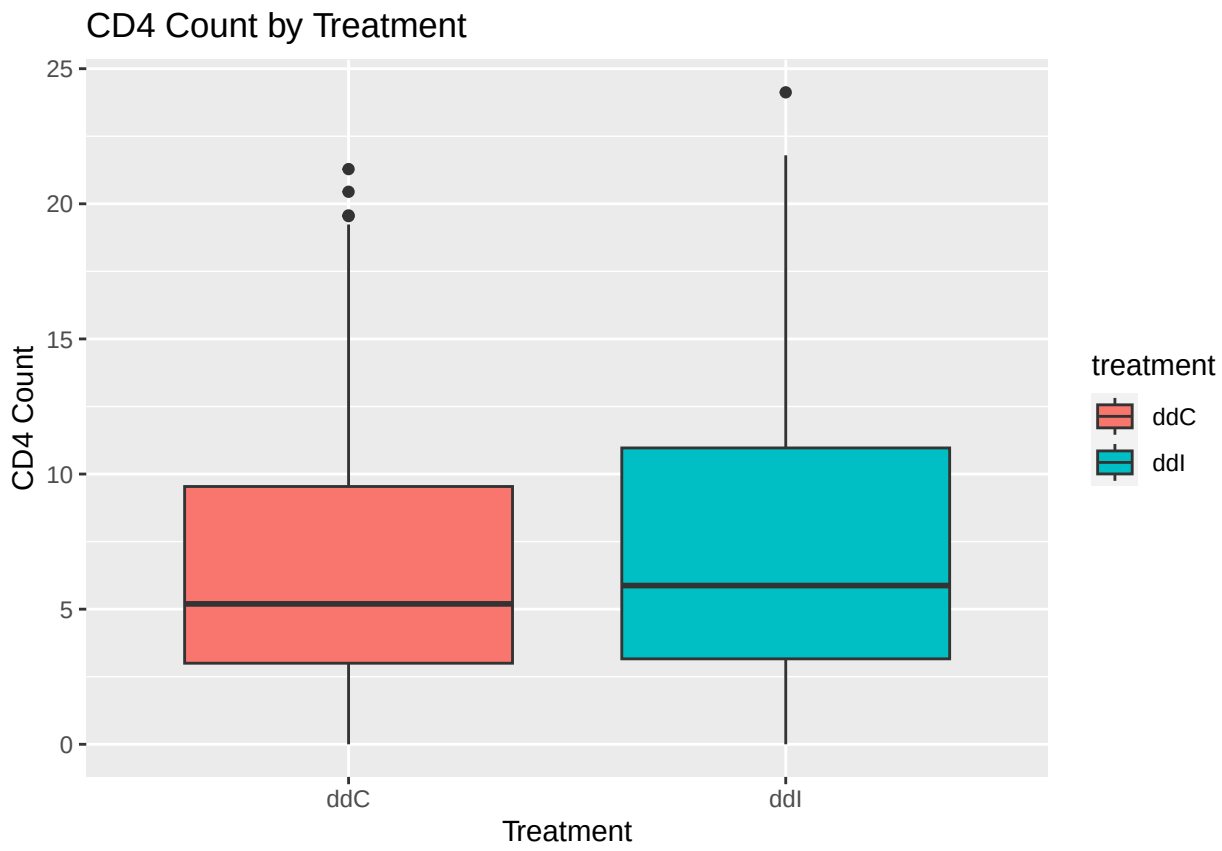



```
# Sex
ggplot(hiv_data, aes(x = sex, fill = sex)) +
  geom_bar() +
  labs(title = "Count of Patients by Sex",
       x = "Sex",
       y = "Count") +
  theme_minimal()
```

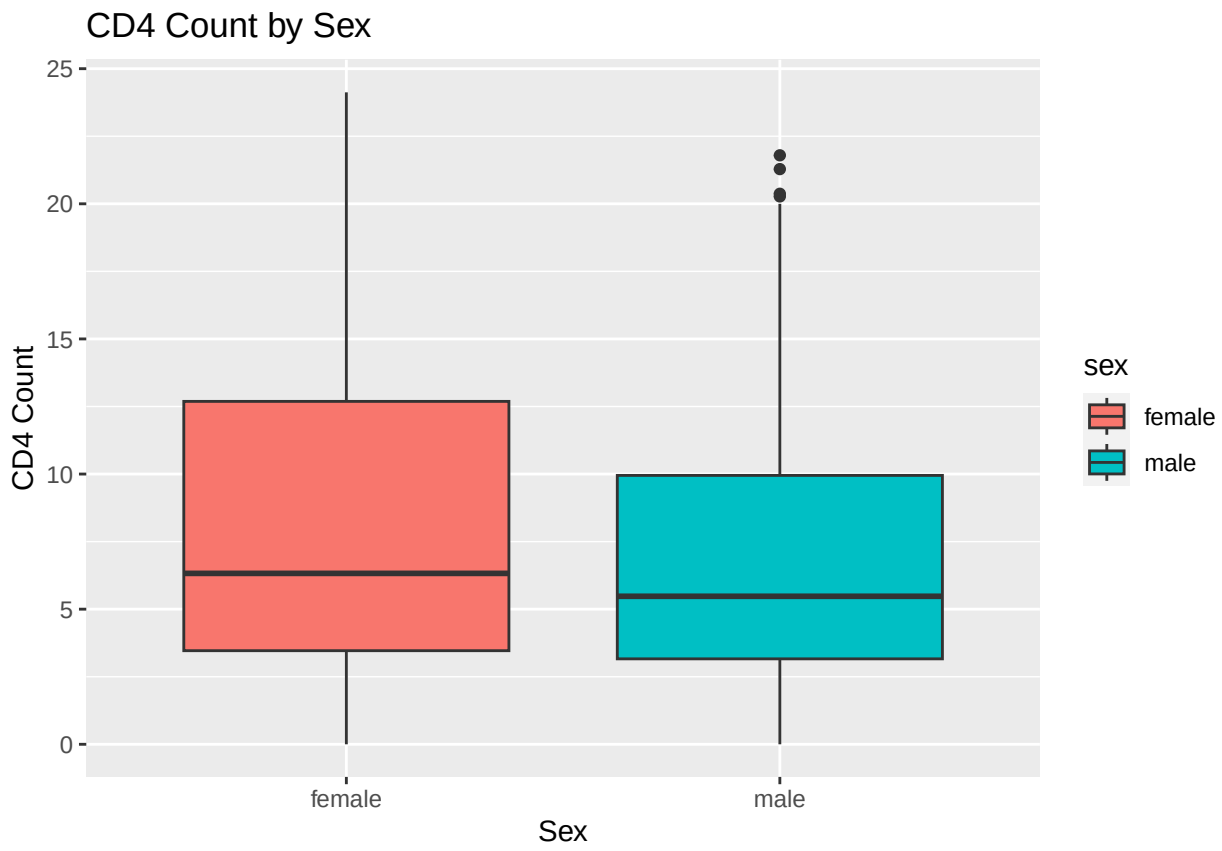


While the assessed treatment is equally distributed in term of counting examples in our data, we observe a strong difference regarding the gender of the individuals. Indeed, our data set contains almost only data related to male. For the rest of our study, we will have to pay attention to this unbalanced groups, and keep in mind that the number of female individuals in our data might be not significant.

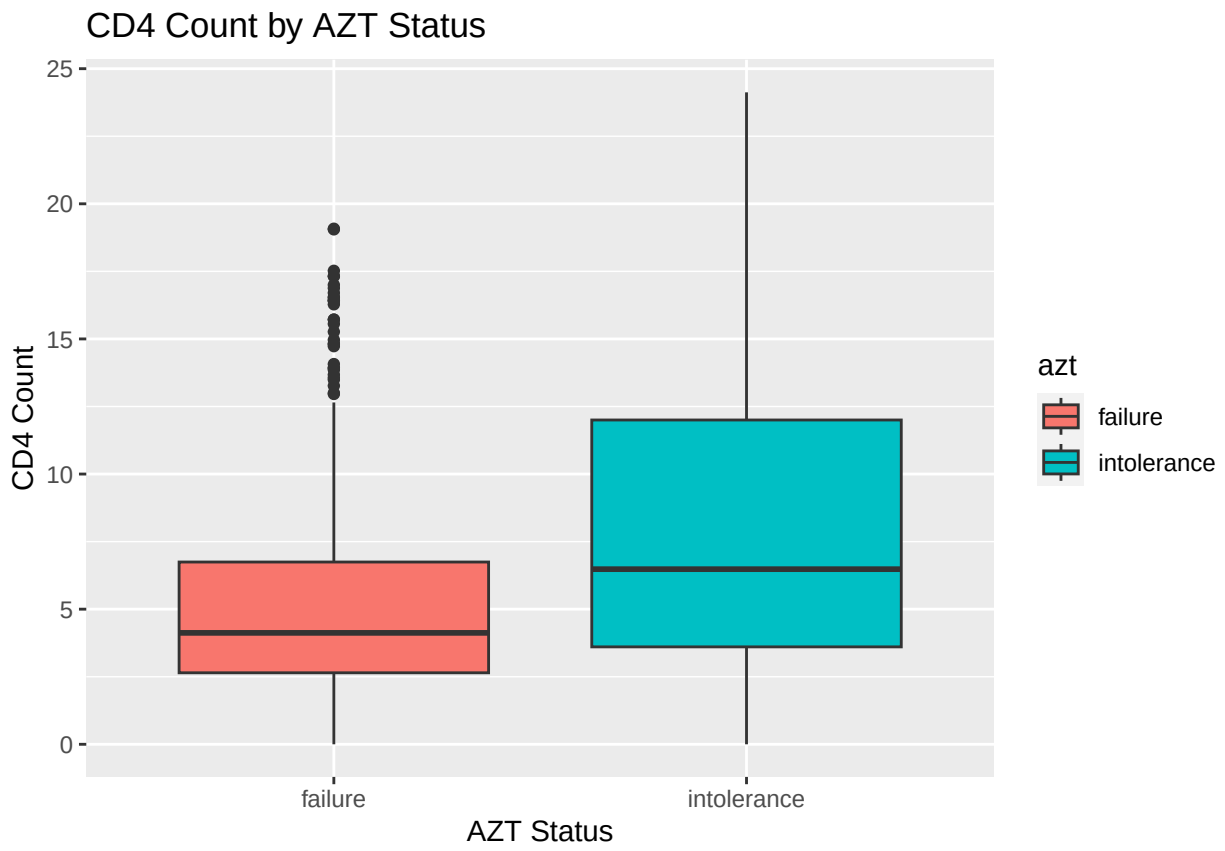
```
# CD4 by Treatment
ggplot(hiv_data, aes(x = treatment, y = cd4, fill = treatment)) +
  geom_boxplot() +
  labs(title = "CD4 Count by Treatment", x = "Treatment", y = "CD4 Count")
```



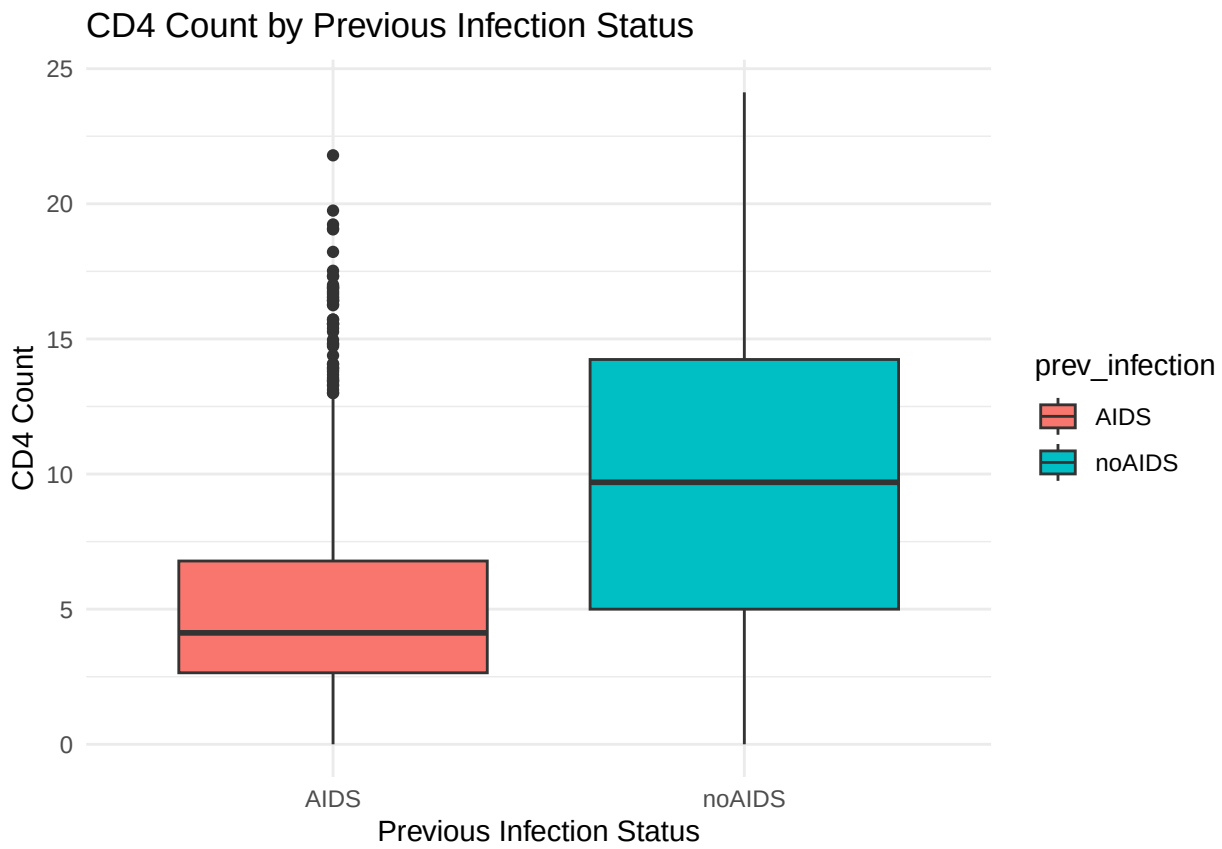

```
# CD4 by Sex
ggplot(hiv_data, aes(x = sex, y = cd4, fill = sex)) +
  geom_boxplot() +
  labs(title = "CD4 Count by Sex", x = "Sex", y = "CD4 Count")
```




```
# CD4 by AZT Status
ggplot(hiv_data, aes(x = azt, y = cd4, fill = azt)) +
  geom_boxplot() +
  labs(title = "CD4 Count by AZT Status", x = "AZT Status", y = "CD4 Count")
```

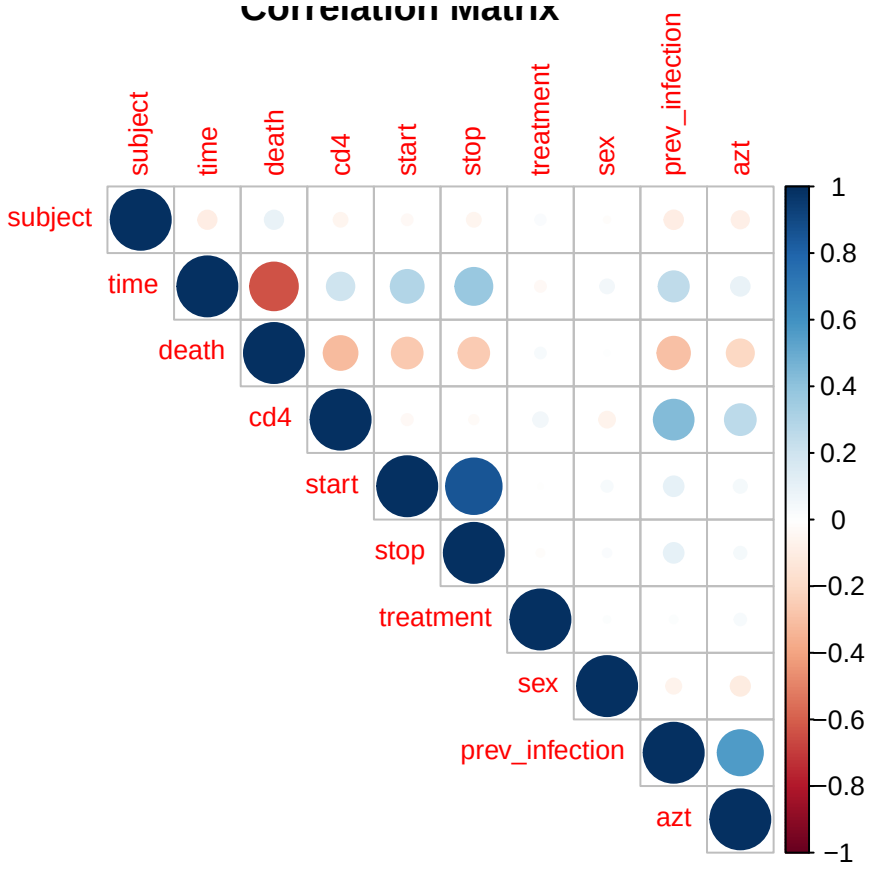



```
# CD4 by Previous Infection Status
ggplot(hiv_data, aes(x = prev_infection, y = cd4, fill = prev_infection)) +
  geom_boxplot() +
  labs(title = "CD4 Count by Previous Infection Status",
       x = "Previous Infection Status",
       y = "CD4 Count") +
  theme_minimal()
```



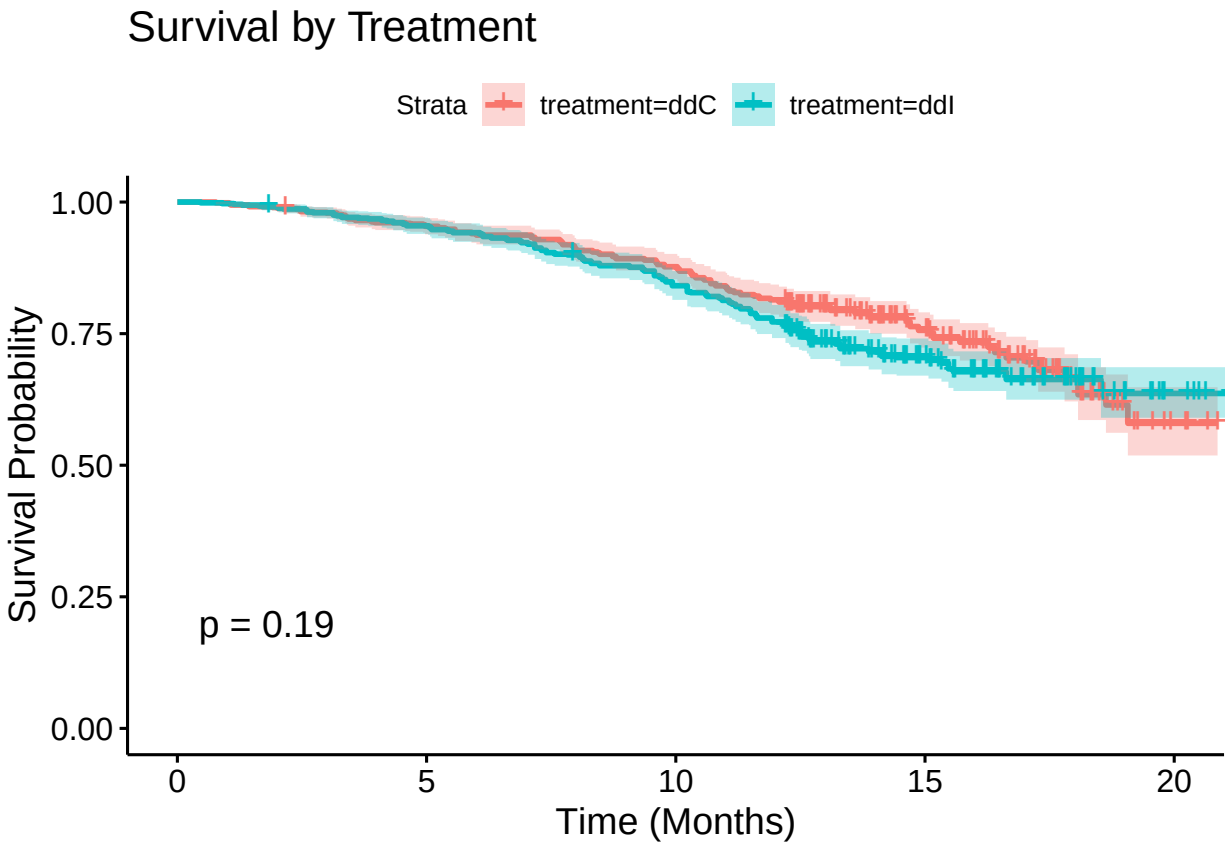
Question 4 : Correlation between covariates

```
# Create dummy variables for categorical variables
hiv_data_numeric <- hiv_data %>%
  mutate(across(c(treatment, sex, prev_infection, azt, death), as.numeric))

# Compute correlation matrix
cor_matrix <- cor(hiv_data_numeric, use = "complete.obs")

# Visualize the correlation matrix
corrplot(cor_matrix, method = "circle", type = "upper", tl.cex = 0.8, main = "Correlation Matrix")
```

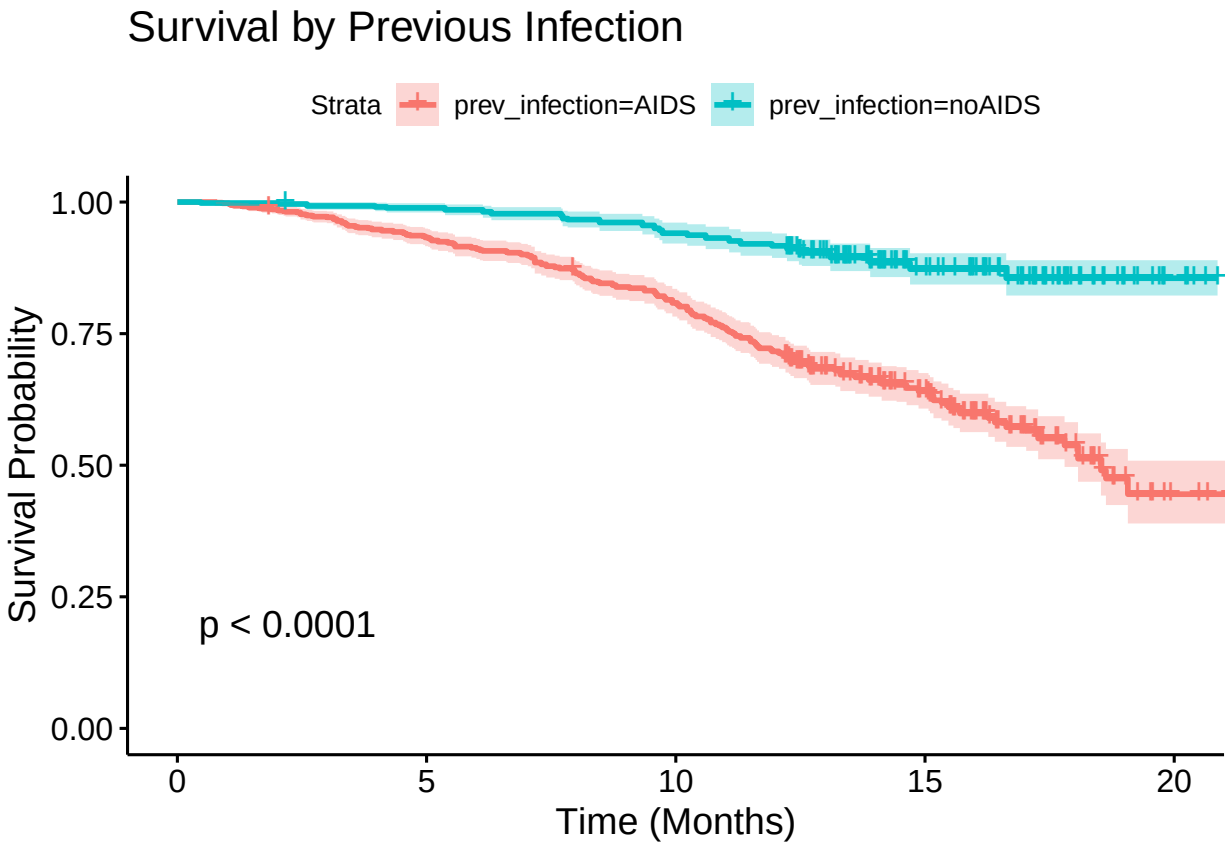


Question 5 : Survival function by sub-groups

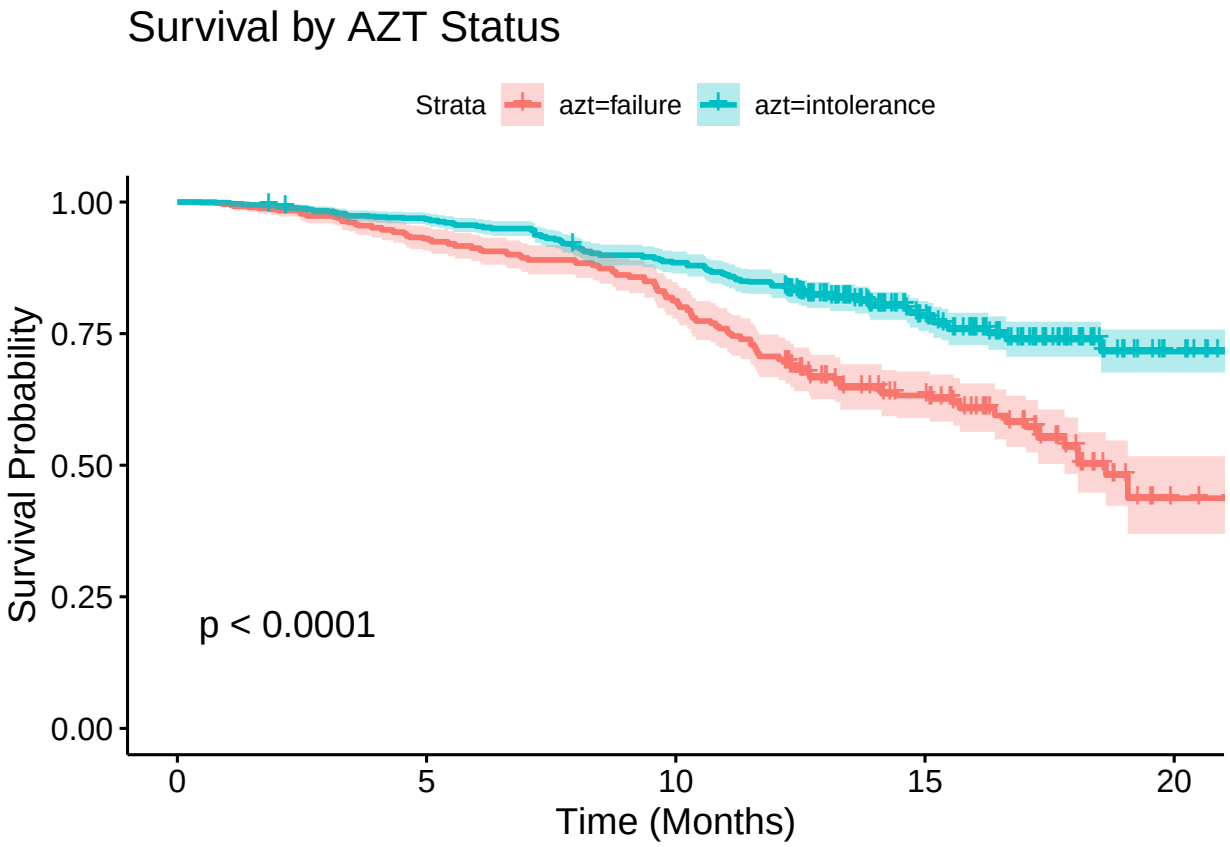
```
# Create a Surv object
surv_object <- Surv(time = hiv_data$time, event = as.numeric(as.character(hiv_data$death)))

# Survival by Treatment
km_treatment <- survfit(surv_object ~ treatment, data = hiv_data)
ggsurvplot(km_treatment, data = hiv_data,
           pval = TRUE, conf.int = TRUE,
           title = "Survival by Treatment",
           xlab = "Time (Months)", ylab = "Survival Probability")
```



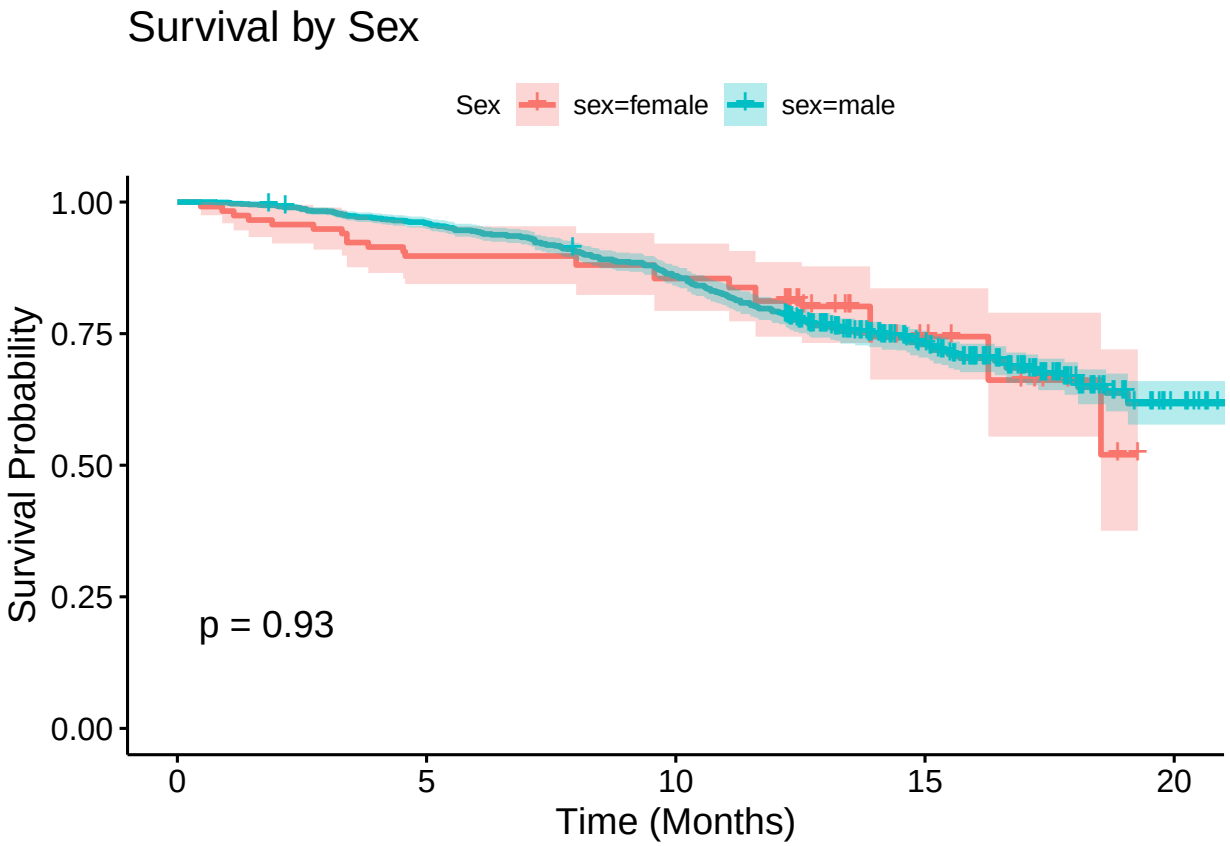

```
# Survival by Previous Infection
km_prev_infection <- survfit(surv_object ~ prev_infection, data = hiv_data)
ggsurvplot(km_prev_infection, data = hiv_data,
           pval = TRUE, conf.int = TRUE,
           title = "Survival by Previous Infection",
           xlab = "Time (Months)", ylab = "Survival Probability")
```




```
# Survival by AZT Status
km_azt <- survfit(surv_object ~ azt, data = hiv_data)
ggsurvplot(km_azt, data = hiv_data,
           pval = TRUE, conf.int = TRUE,
           title = "Survival by AZT Status",
           xlab = "Time (Months)", ylab = "Survival Probability")
```




```r
# Survival by Sex
km_sex <- survfit(surv_object ~ sex, data = hiv_data)
ggsurvplot(km_sex, data = hiv_data,
           pval = TRUE, conf.int = TRUE,
           title = "Survival by Sex",
           xlab = "Time (Months)",
           ylab = "Survival Probability",
           legend.title = "Sex",
           )  # Optional custom colors for groups
```



Remarks: - At first sight, it doesn't seem that there is a significant difference in survival functions among the groups characterized by the sex of the individuals or the treatment given to them. However, as the number of female individuals in our data is not statistically big enough, we should pay attention to the variability of our the survival function regarding the group female. Thus, for the groups characterized by the factor 'sex', we couldn't really say anything qualitatively, by just comparing the survival function visually, we need more precise analysis. - Besides, we might have significant survival functions among the individuals who had a previous infection or not. Individuals having a previous AIDS infection, seem to have shorter life expectancy compared to the others who doesn't, which is logical taking a medical point of view. - Same for the AZT status. Individuals who failed for AZT therapy seem to have a shorter survival probability, than the ones who were intolerant.

Question 6 : Test for differences in survival function

Let's apply a log-rank test to check if there is any difference in survival functions among the groups, with the significance level set at 0.05. #### By Treatment

```
# Log-rank test
survdiff(Surv(time, as.numeric(as.character(death))) ~ treatment, data = hiv_data)
```

```
## Call:
## survdiff(formula = Surv(time, as.numeric(as.character(death))) ~
##     treatment, data = hiv_data)
##
##                  N Observed Expected (O-E)^2/E (O-E)^2/V
## treatment=ddC 717      200      213     0.843      1.75
## treatment=ddI 688      212      199     0.906      1.75
```

```
##
##  Chisq= 1.8  on 1 degrees of freedom, p= 0.2
```

```r
# Log-rank test
survdiff(Surv(time, as.numeric(as.character(death))) ~ prev_infection, data = hiv_data)
```

By Previous Infection

```
## Call:
## survdiff(formula = Surv(time, as.numeric(as.character(death))) ~
##     prev_infection, data = hiv_data)
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## prev_infection=AIDS    863      344      232      53.9       125
## prev_infection=noAIDS 542       68      180      69.6       125
##
##  Chisq= 125  on 1 degrees of freedom, p= <2e-16
```

```r
# Log-rank test
survdiff(Surv(time, as.numeric(as.character(death))) ~ azt, data = hiv_data)
```

By AZT Status

```
## Call:
## survdiff(formula = Surv(time, as.numeric(as.character(death))) ~
##     azt, data = hiv_data)
##
##                    N Observed Expected (O-E)^2/E (O-E)^2/V
## azt=failure      491      206      137      35.4      53.1
## azt=intolerance 914      206      275      17.5      53.1
##
##  Chisq= 53.1  on 1 degrees of freedom, p= 3e-13
```

```r
# Log-rank test
survdiff(Surv(time, as.numeric(as.character(death))) ~ sex, data = hiv_data)
```

By Sex status

```
## Call:
## survdiff(formula = Surv(time, as.numeric(as.character(death))) ~
##     sex, data = hiv_data)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=female   117       33     32.5  0.007689   0.00838
```

```
## sex=male   1288      379    379.5  0.000658   0.00838
##
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

Based on the available data the null hypothesis is rejected for the survival functions among the groups "azt" and "prev_infection", as the obtained p-value is lower than the the significance level set at 0.05. Which quantitatively confirm our previous qualitative observations.

B. Statistical Modelling

Question 7 : Train and Test split

```
# Check the distribution of the censorship variable
table(hiv_data$death)
```

```
##
##   0   1
## 993 412
```

```
# Set a seed for reproducibility
set.seed(123)

# Create an 80/20 partition stratified by the censorship variable
train_index <- createDataPartition(hiv_data$death, p = 0.8, list = FALSE)

# Split the data into training and testing sets
train_data <- hiv_data[train_index, ]
test_data <- hiv_data[-train_index, ]

# Verify the stratification by comparing the censorship proportions
train_censorship <- table(train_data$death) / nrow(train_data)
test_censorship <- table(test_data$death) / nrow(test_data)

# Display the results
cat("Censorship proportions in training data:\n")
```

```
## Censorship proportions in training data:
```

```
print(train_censorship)
```

```
##
##         0         1
## 0.7066667 0.2933333
```

```
cat("\nCensorship proportions in testing data:\n")
```

```
##
## Censorship proportions in testing data:
```

```
print(test_censorship)
```

```
##
##         0         1
## 0.7071429 0.2928571
```

Here, we have our divided data into train and test conserving the censorship proportions.

Question 8 : Training linear and generalized Cox model

Linear Model with Start/Stop Format associate to 'cd4' Variable

```
# Create the Surv object for survival analysis
surv_object_train <- Surv(train_data$start, train_data$stop, event = as.numeric(as.character(train_data$

# 1. Linear Cox Proportional Hazards Model
# Fit the Cox model with linear relationships for covariates
cox_model_linear <- coxph(surv_object_train ~ cd4 + sex + treatment + prev_infection + azt, data = train

# Summary of the linear model
summary(cox_model_linear)
```

```
## Call:
## coxph(formula = surv_object_train ~ cd4 + sex + treatment + prev_infection +
##     azt, data = train_data, x = TRUE, y = TRUE)
##
##   n= 1125, number of events= 330
##
##                          coef exp(coef) se(coef)      z Pr(>|z|)
## cd4                   -0.12368   0.88366  0.01668 -7.414 1.22e-13 ***
## sexmale               -0.16626   0.84682  0.18994 -0.875    0.381
## treatmentddI           0.14832   1.15988  0.11054  1.342    0.180
## prev_infectionnoAIDS -0.70378   0.49471  0.16944 -4.153 3.27e-05 ***
## aztintolerance        -0.14840   0.86208  0.12335 -1.203    0.229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## cd4                      0.8837     1.1317    0.8552    0.9130
## sexmale                  0.8468     1.1809    0.5836    1.2288
## treatmentddI             1.1599     0.8622    0.9340    1.4405
## prev_infectionnoAIDS     0.4947     2.0214    0.3549    0.6896
## aztintolerance           0.8621     1.1600    0.6770    1.0978
##
## Concordance= 0.711  (se = 0.014 )
## Likelihood ratio test= 159.8  on 5 df,   p=<2e-16
## Wald test            = 120.7  on 5 df,   p=<2e-16
## Score (logrank) test = 135.8  on 5 df,   p=<2e-16
```

Let's train a new linear model by keeping only the most significant covariates : 'cd4' and 'prev_infection'

```r
# 2. Second Linear Cox Proportional Hazards Model
# Fit the Cox model
cox_model_linear2 <- coxph(surv_object_train ~ cd4 +  prev_infection, data = train_data, x = TRUE, y = T

# Summary of the linear model
summary(cox_model_linear2)
```

```
## Call:
## coxph(formula = surv_object_train ~ cd4 + prev_infection, data = train_data,
##     x = TRUE, y = TRUE)
##
##   n= 1125, number of events= 330
##
##                          coef exp(coef) se(coef)      z Pr(>|z|)
## cd4                  -0.12440   0.88303  0.01674 -7.430 1.09e-13 ***
## prev_infectionnoAIDS -0.78370   0.45671  0.15484 -5.061 4.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## cd4                     0.8830      1.132    0.8545    0.9125
## prev_infectionnoAIDS    0.4567      2.190    0.3372    0.6186
##
## Concordance= 0.708  (se = 0.015 )
## Likelihood ratio test= 156  on 2 df,   p=<2e-16
## Wald test            = 115.4  on 2 df,   p=<2e-16
## Score (logrank) test = 129.9  on 2 df,   p=<2e-16
```

Observation : The concordance index is slightly worse with this new model. We will have to check with other metrics to have better understanding of our model performances.

Let's now check the hypothesis that the 'cd4' covariate has a linear effect, by plotting the martingale residuals :

It seems that we better have to add non linear relationship to the continious variable cd4. For this purpose, let's use the splines functions.

```r
# 2. Generalized (Nonlinear) Cox Model using splines for `cd4` (as an example of nonlinear relationship
cox_model_nonlinear <- coxph(surv_object_train ~ pspline(cd4) + sex + treatment + prev_infection + azt,

# Summary of the nonlinear model
summary(cox_model_nonlinear)
```

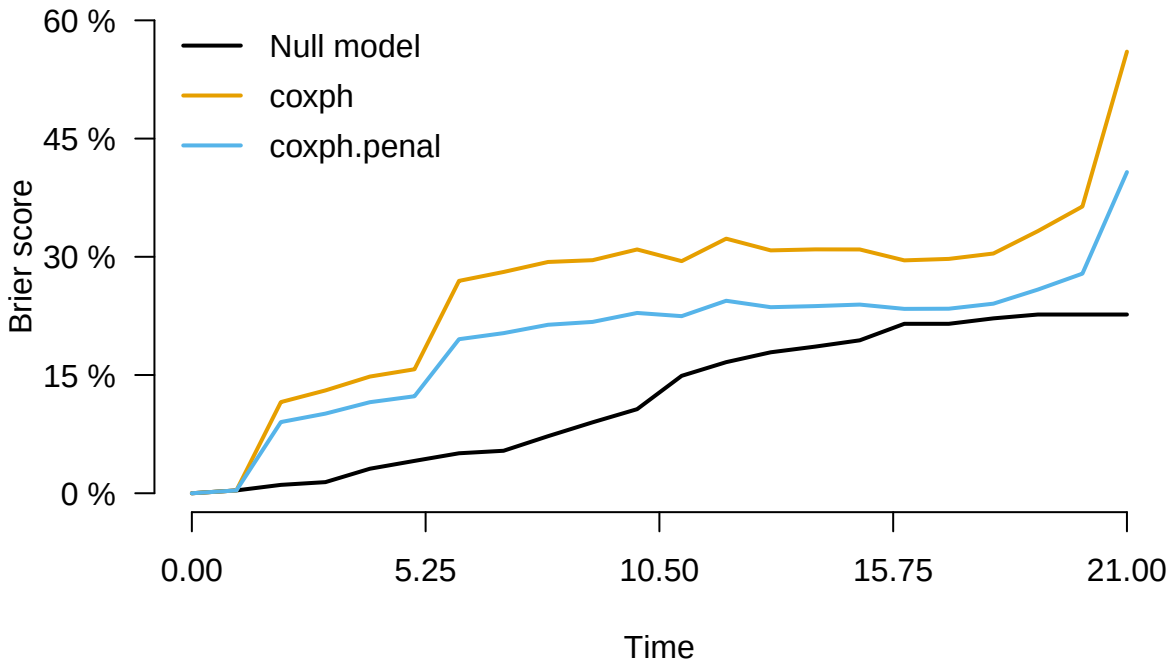
```
## Call:
## coxph(formula = surv_object_train ~ pspline(cd4) + sex + treatment +
##     prev_infection + azt, data = train_data, x = TRUE, y = TRUE)
##
##   n= 1125, number of events= 330
##
##                       coef    se(coef) se2     Chisq DF   p
## pspline(cd4), linear -0.1240 0.01735  0.01724 51.12 1.00 8.7e-13
## pspline(cd4), nonlin                          2.38 3.07 5.1e-01
## sexmale              -0.1767 0.19061  0.19052  0.86 1.00 3.5e-01
```

```
## treatmentddI          0.1500 0.11059  0.11057  1.84 1.00 1.8e-01
## prev_infectionnoAIDS -0.6984 0.17003  0.16994 16.87 1.00 4.0e-05
## aztintolerance       -0.1550 0.12421  0.12412  1.56 1.00 2.1e-01
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## ps(cd4)3               1.04882     0.9535 0.3700023    2.9730
## ps(cd4)4               0.93453     1.0701 0.2681024    3.2575
## ps(cd4)5               0.61396     1.6288 0.1874172    2.0112
## ps(cd4)6               0.50426     1.9831 0.1530458    1.6615
## ps(cd4)7               0.43471     2.3004 0.1273209    1.4842
## ps(cd4)8               0.25557     3.9129 0.0723711    0.9025
## ps(cd4)9               0.15604     6.4085 0.0423391    0.5751
## ps(cd4)10              0.12058     8.2933 0.0297406    0.4889
## ps(cd4)11              0.08846    11.3052 0.0150917    0.5185
## ps(cd4)12              0.06120    16.3394 0.0045573    0.8219
## ps(cd4)13              0.04180    23.9237 0.0008864    1.9710
## ps(cd4)14              0.02850    35.0886 0.0001221    6.6511
## sexmale                0.83799     1.1933 0.5767583    1.2176
## treatmentddI           1.16182     0.8607 0.9354206    1.4430
## prev_infectionnoAIDS   0.49739     2.0105 0.3564179    0.6941
## aztintolerance         0.85638     1.1677 0.6713342    1.0924
##
## Iterations: 4 outer, 12 Newton-Raphson
##      Theta= 0.779737
## Degrees of freedom for terms= 4.1 1.0 1.0 1.0 1.0
## Concordance= 0.712  (se = 0.014 )
## Likelihood ratio test= 163.6  on 8.06 df,   p=<2e-16
```

Question 9

Here are the plot of the brier score of the three Cox models just developed. As the Brier is an error metric, we want it to be as low as possible. So here, the best model seems to be the third, with non-linear relations.

```
library(rms)
library(riskRegression)

brier <- Score(list(cox_model_linear,  cox_model_nonlinear), Hist(time, death) ~ 1, data = test_data, m
plot_brier_cox <- plotBrier(brier, ylim=c(0,0.6))
```



We can compute the embedded score on the test sample by writing a function which compute the area under the curve thanks to the 'integrate' function. We also want it to be low.

```
# Fonction to compute the embeded Brier score
calculate_embedded_score <- function(models, data, time_range, metrics = "brier") {
  # Computig Brier score for each model
  brier_scores <- Score(
    models,
    Hist(time, death) ~ 1,
    data = data,
    metrics = metrics,
    times = time_range
  )

  # Extraire les données de Brier
  brier_data <- brier_scores$Brier$score

  # Computing the Air Under the Curve (Brier AUC)
  calculate_auc <- function(x, y) {
    interp_func <- approxfun(x, y, method = "linear", rule = 2)
    auc_value <- integrate(interp_func, min(x), max(x))$value
    return(auc_value)
  }

  # Compution IBS for each model
  results <- list()
  for (model_name in names(models)) {
```

```
    model_brier <- subset(brier_data, model == model_name)
    auc_brier <- calculate_auc(model_brier$times, model_brier$Brier)
    results[[model_name]] <- list(
      IBS = auc_brier,
      Details = model_brier
    )
  }

  return(results)
}
```

```
# Make the computation on our data
time_range <- seq(0, 21.4, by = 0.1) # Gamme de temps
models <- list(
  "Linear Cox Model" = cox_model_linear,
  "Nonlinear Cox Model" = cox_model_nonlinear
)

embedded_scores <- calculate_embedded_score(models, test_data, time_range)

# Printing the results
for (model_name in names(embedded_scores)) {
  cat("Model:", model_name, "\n")
  cat("Integrated Brier Score (IBS):", embedded_scores[[model_name]]$IBS, "\n")
}
```

```
## Model: Linear Cox Model
## Integrated Brier Score (IBS): 5.44647
## Model: Nonlinear Cox Model
## Integrated Brier Score (IBS): 4.155653
```

```
calculate_auc <- function(x, y) {
  interp_func <- approxfun(x, y, method = "linear", rule = 2)
  auc_value <- integrate(interp_func, min(x), max(x))$value
  return(auc_value)
}

brier_score_cox <- data.frame(x = plot_brier_cox$times, y = plot_brier_cox$Brier)

calculate_auc(brier_score_cox$x, brier_score_cox$y)
```

```
## [1] 4.037133
```

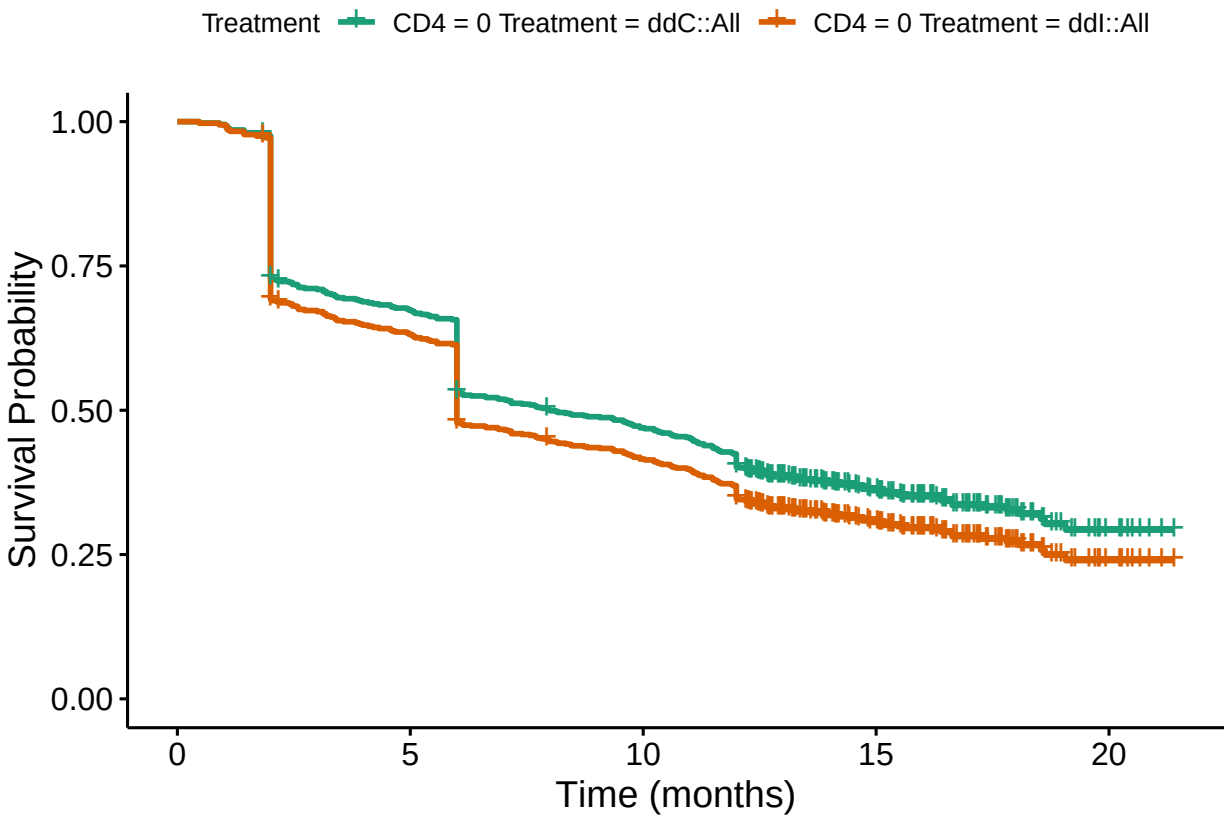
Question 10

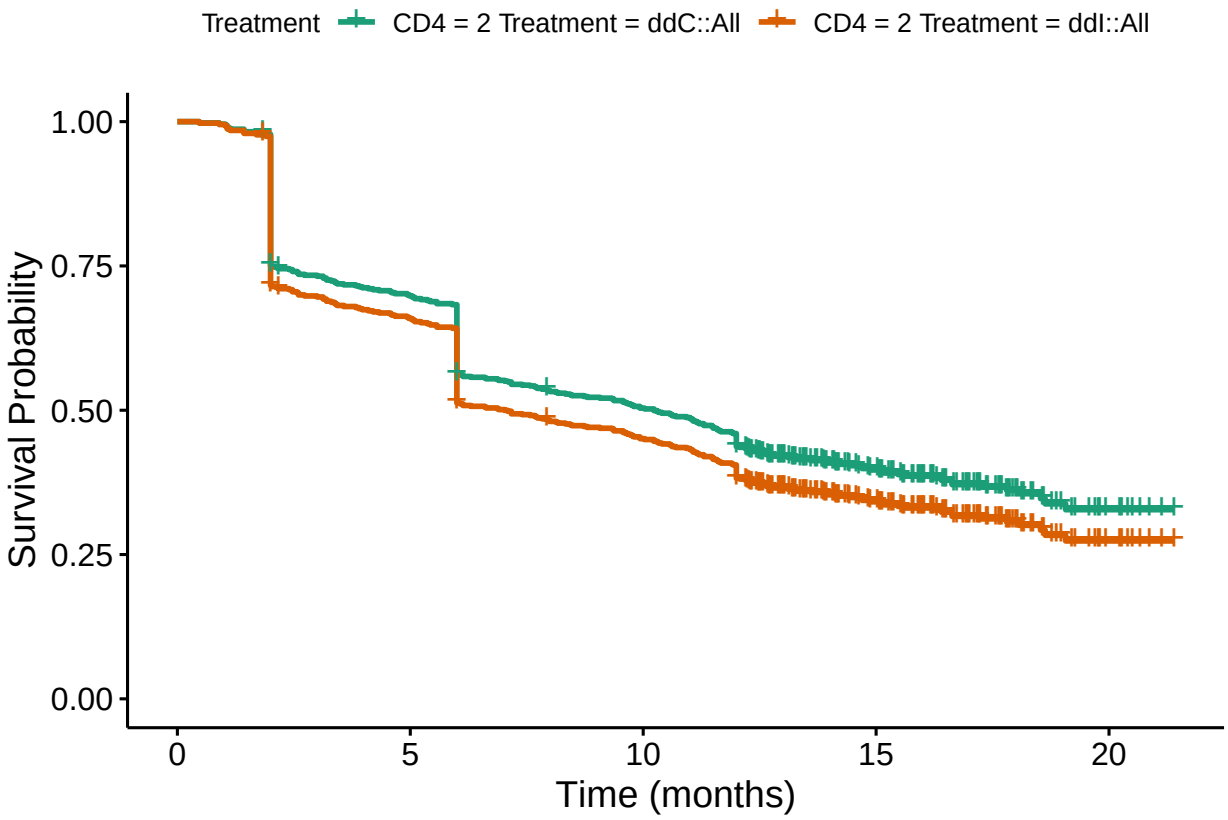
Let's take randomly three subjects:

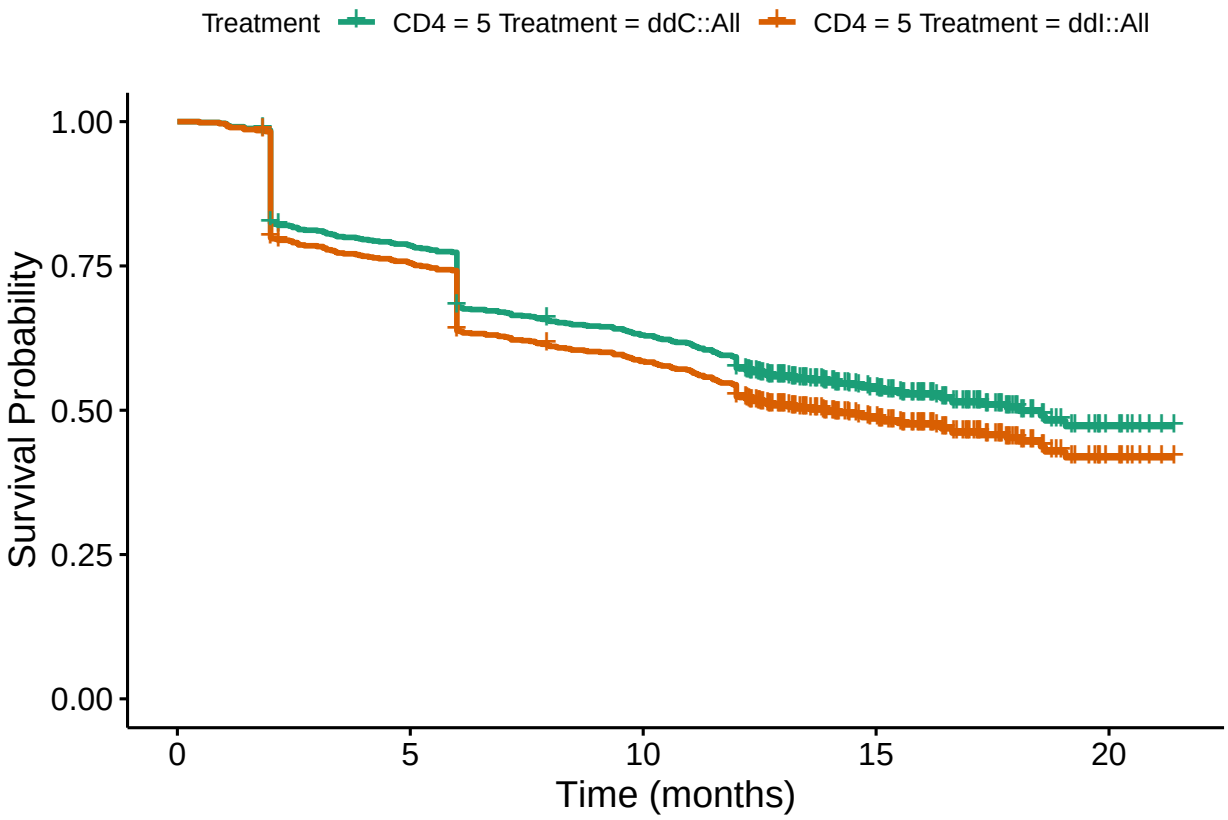
```
library(survival)
library(MASS)
library(splines)
library(survminer)
```

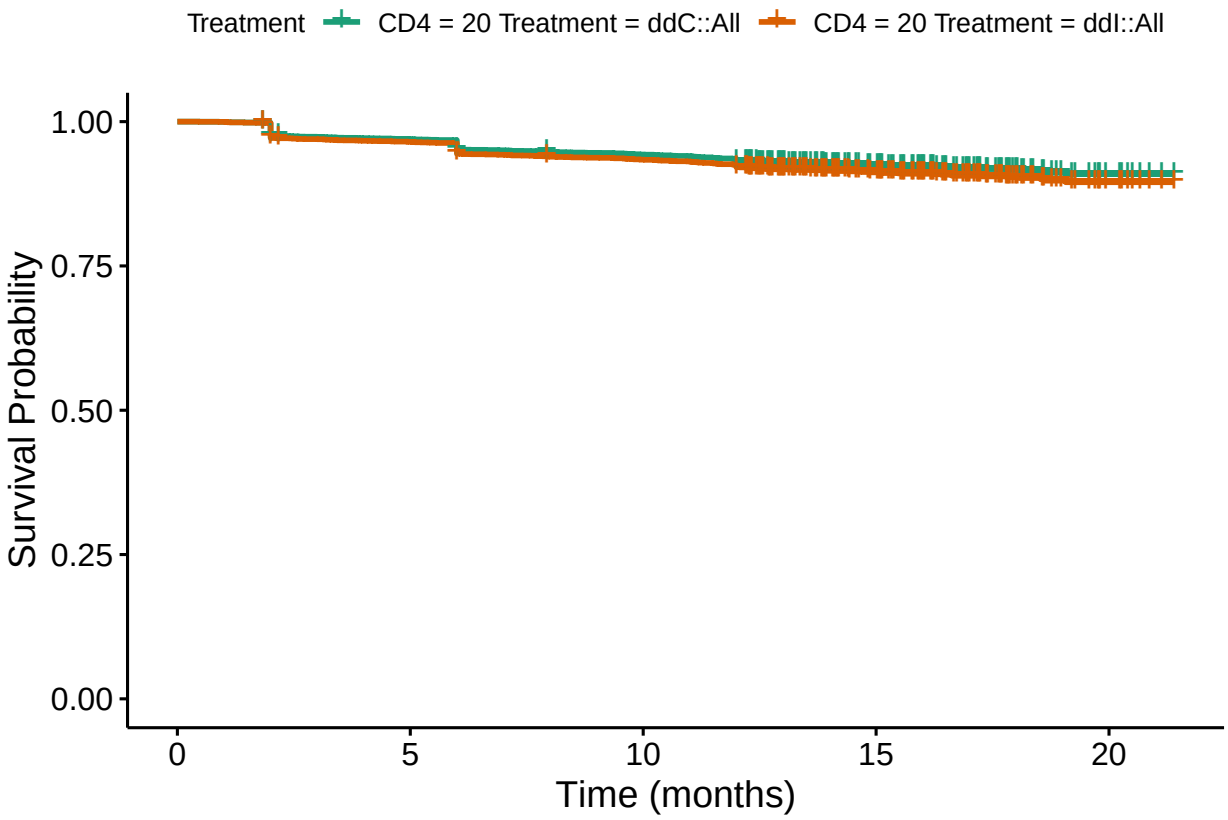
```r
subject_rows <- c(3, 5, 7)

all_survival_results <- data.frame(
  subject = numeric(),
  cd4 = numeric(),
  treatment = character(),
  time = numeric(),
  survival_prob = numeric()
)

for (subject_row in subject_rows) {
  # cat("\nAnalyzing Subject:", subject_row, "\n")

  base_data <- test_data[subject_row, ]
  cd4_values <- c(0, 2, base_data$cd4, 20, 50)

  for (cd4_value in cd4_values) {
    new_data_cd4 <- base_data
    new_data_cd4$cd4 <- cd4_value

    survival_curves <- list()

    for (treatment_value in c('ddC', 'ddI')) {
      new_data_cd4$treatment <- treatment_value

      survival_curve <- survfit(cox_model_nonlinear, newdata = new_data_cd4)

      survival_prob_18_months <- summary(survival_curve, times = 18)$surv

      # Ajouter les résultats au tableau
      all_survival_results <- rbind(
        all_survival_results,
        data.frame(
          subject = subject_row,
          cd4 = cd4_value,
          treatment = treatment_value,
          time = 18,
          survival_prob = survival_prob_18_months
        )
      )

      curve_label <- paste("CD4 =", cd4_value, "Treatment =", treatment_value)
      survival_curves[[curve_label]] <- survival_curve

      #cat("Subject:", subject_row,
      #    "- CD4:", cd4_value,
      #    "- Treatment:", treatment_value,
      #    "- Survival Probability at 18 months:", survival_prob_18_months, "\n")
    }

    plot_cd4 <- ggsurvplot(
      survival_curves,
      combine = TRUE,
```

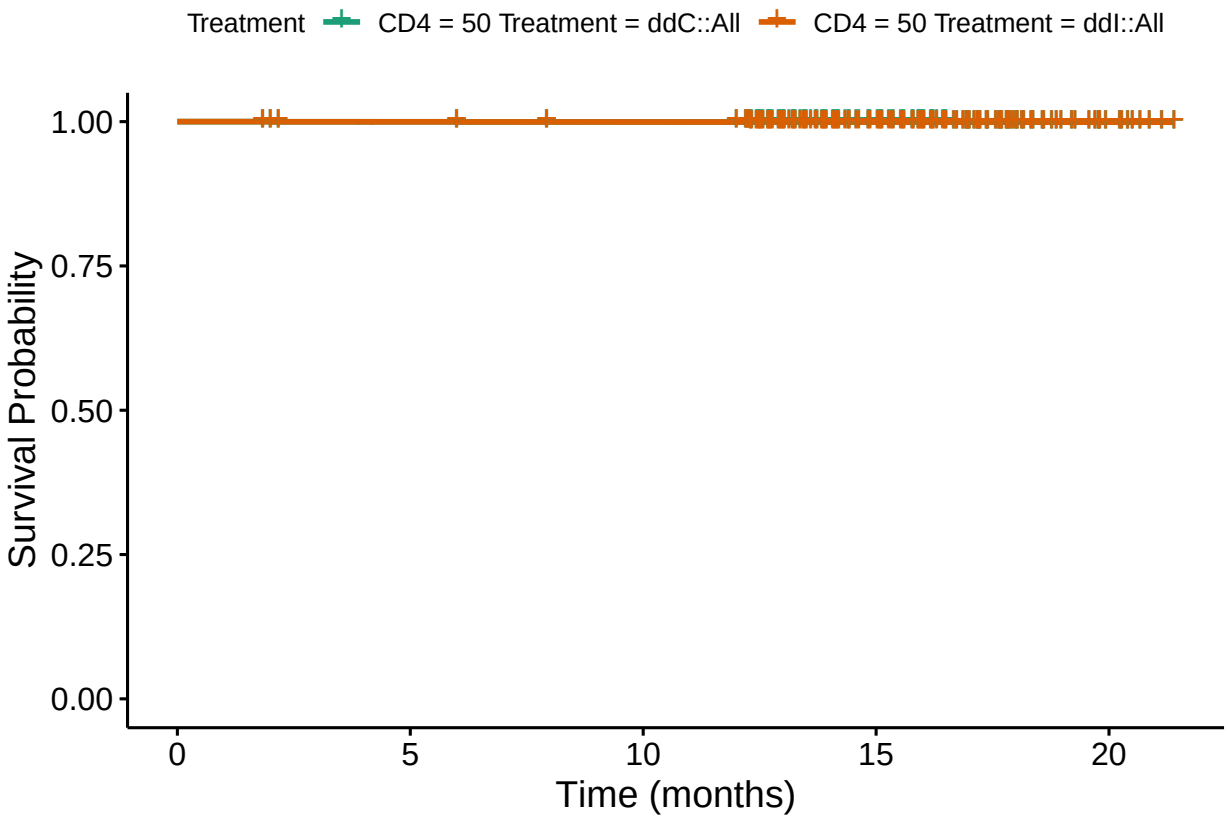
```
      main = paste("Survival Curves for Subject", subject_row, "and CD4 =", cd4_value),
      xlab = "Time (months)",
      ylab = "Survival Probability",
      legend.title = "Treatment",
      palette = "Dark2"
    )

    print(plot_cd4)
  }
}
```

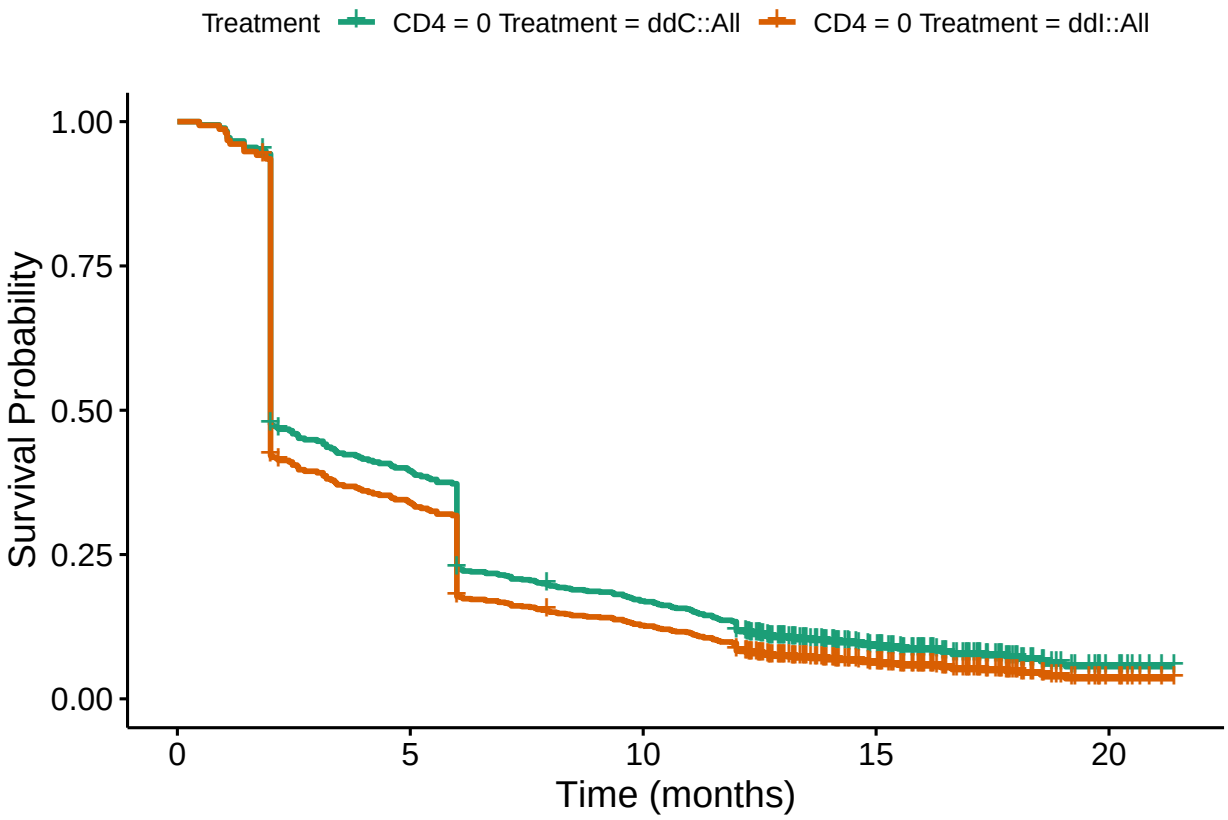


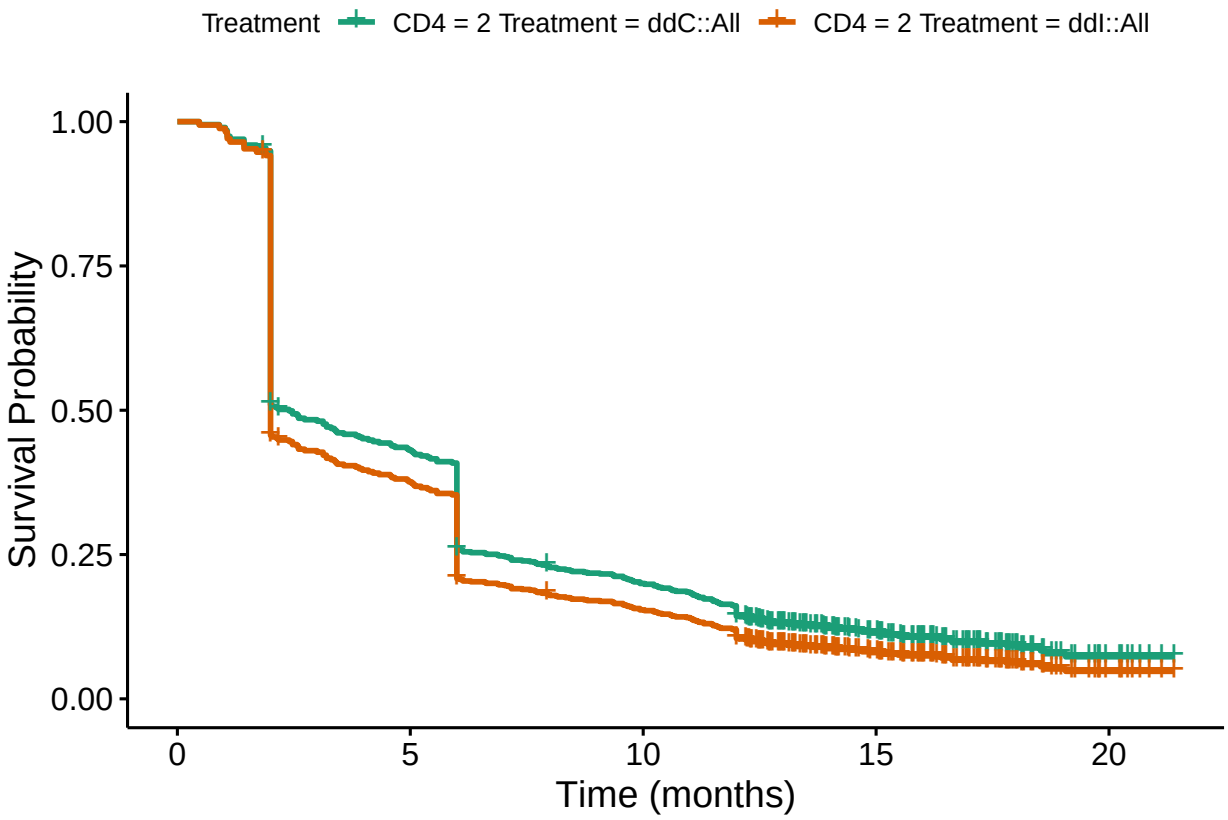

Treatment
CD4 = 2 Treatment = ddC::All
CD4 = 2 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)

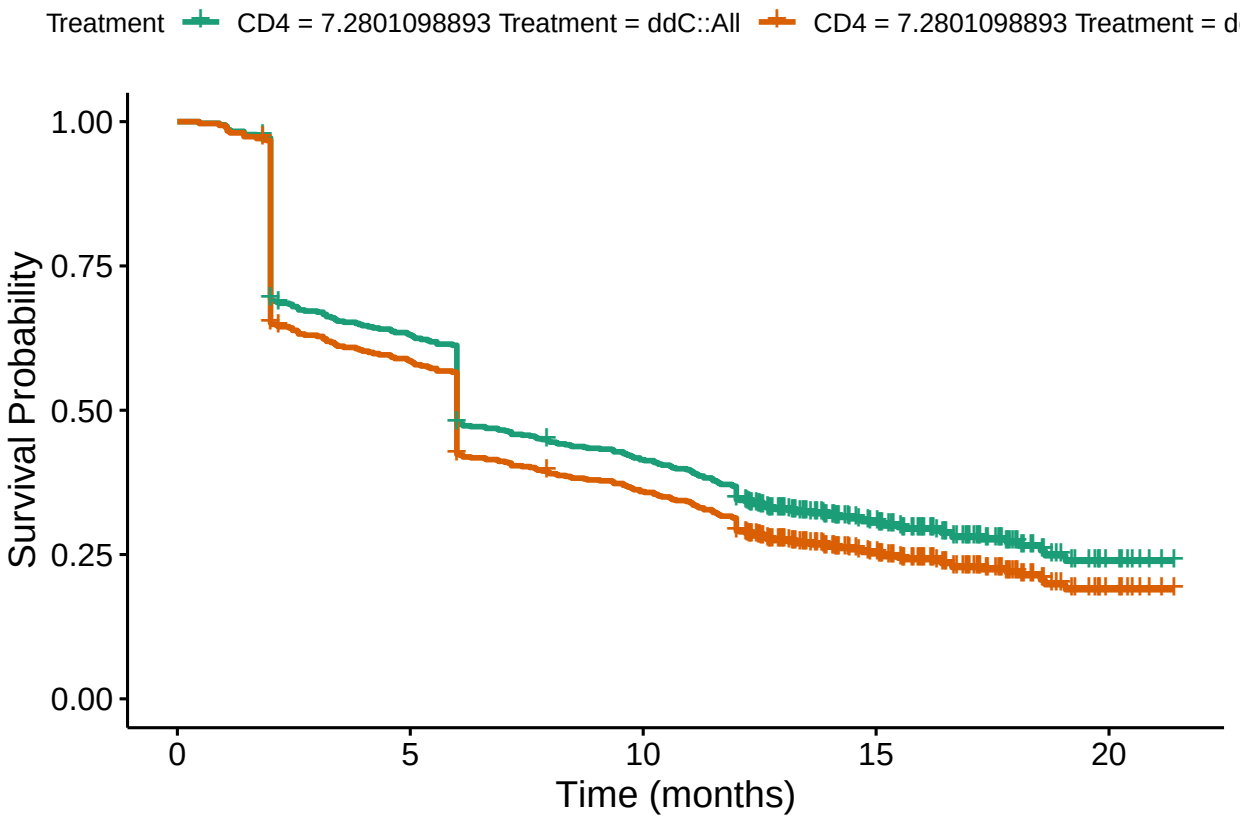

Treatment
CD4 = 5 Treatment = ddC::All
CD4 = 5 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)

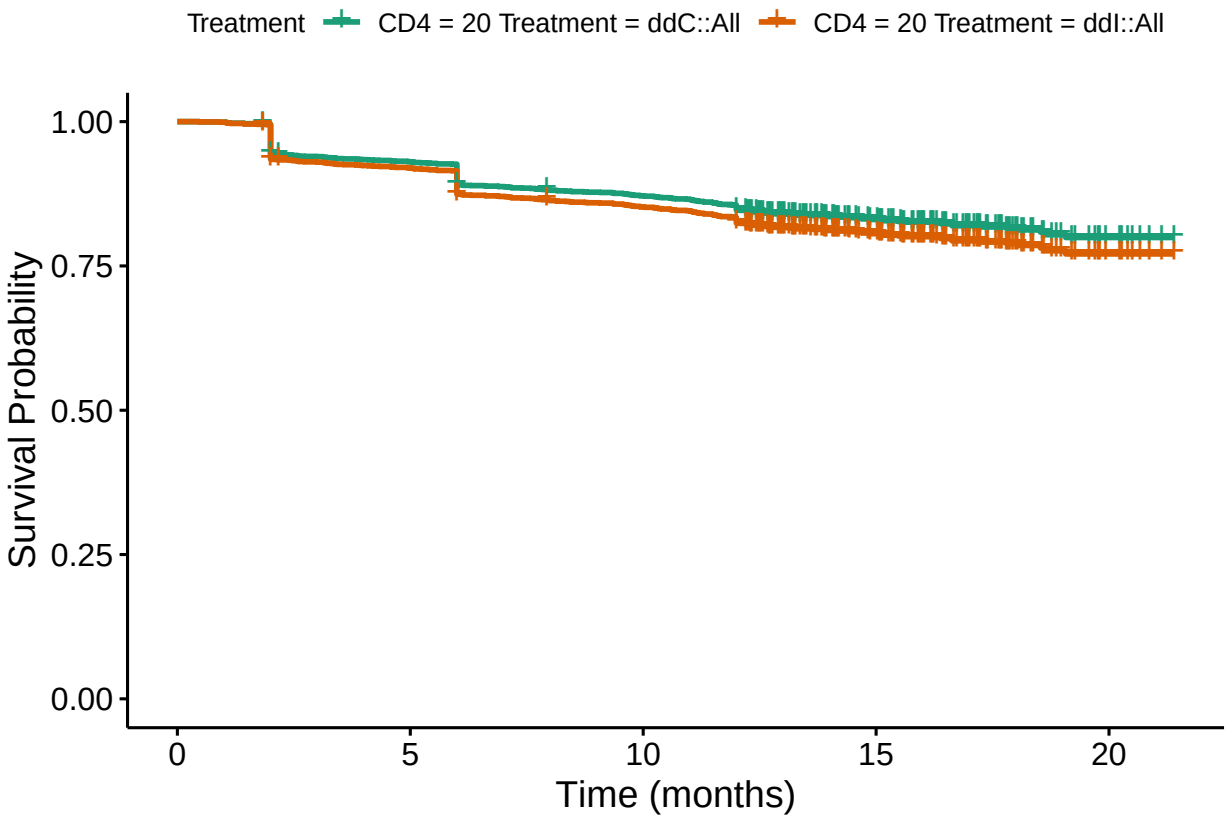

Treatment
CD4 = 20 Treatment = ddC::All
CD4 = 20 Treatment = ddI::All
1.00
0.75
0.50
0.25
0.00
Survival Probability
0
5
10
15
20
Time (months)

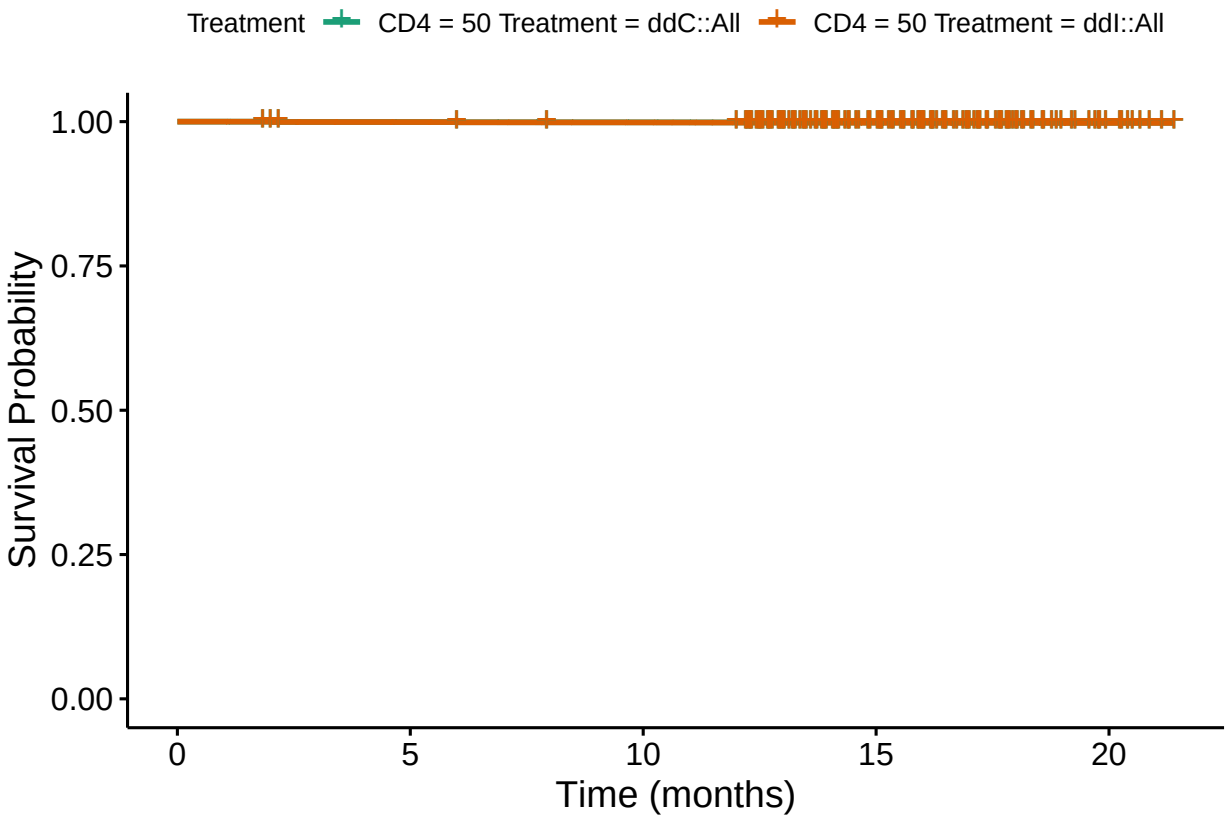

Treatment
CD4 = 50 Treatment = ddC::All
CD4 = 50 Treatment = ddI::All
1.00
0.75
0.50
0.25
0.00
Survival Probability
0
5
10
15
20
Time (months)

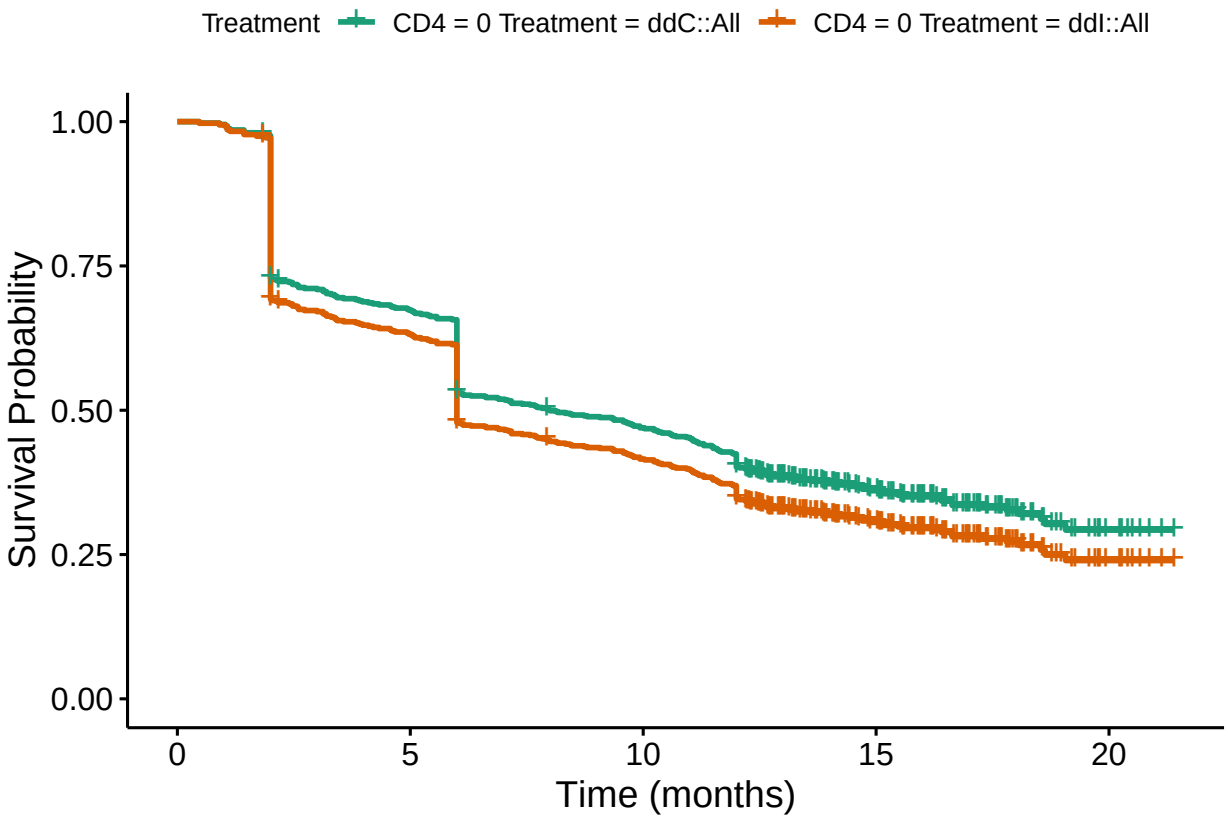

Treatment
CD4 = 0 Treatment = ddC::All
CD4 = 0 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)

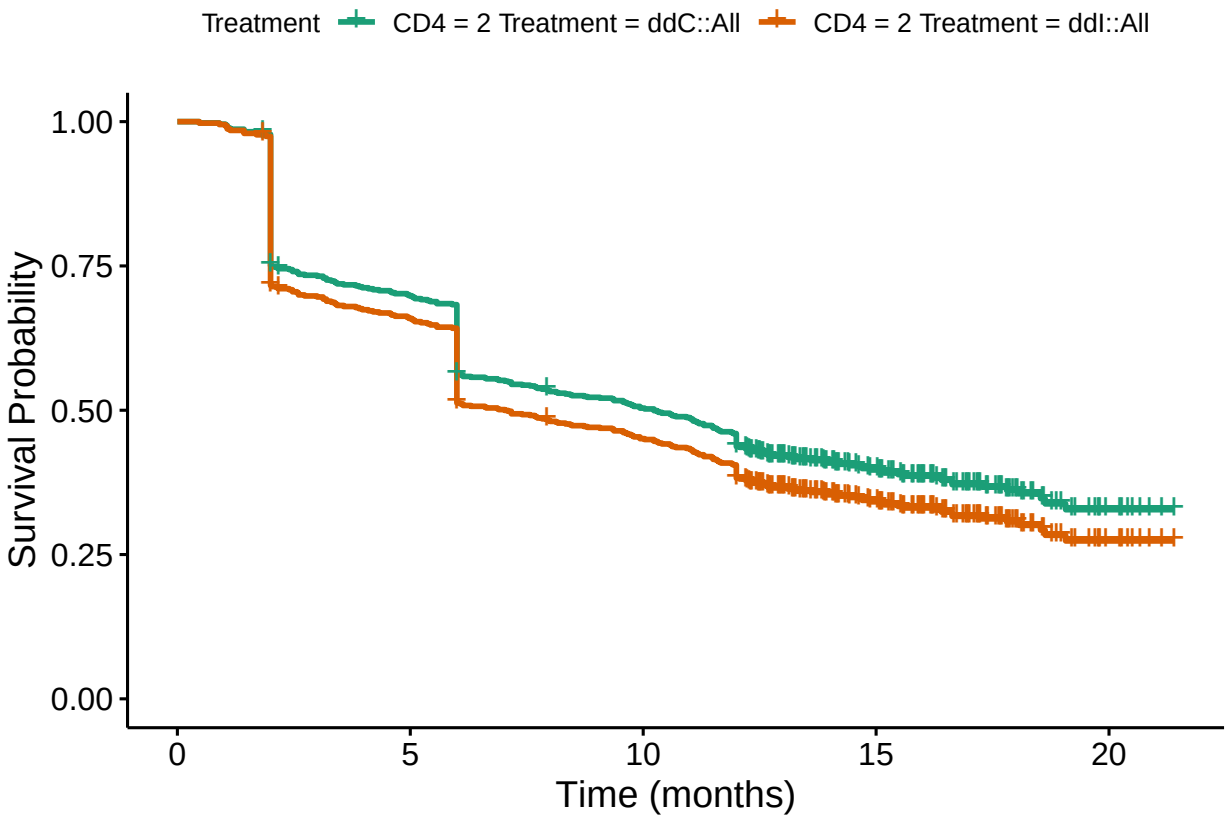

Treatment
CD4 = 2 Treatment = ddC::All
CD4 = 2 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)

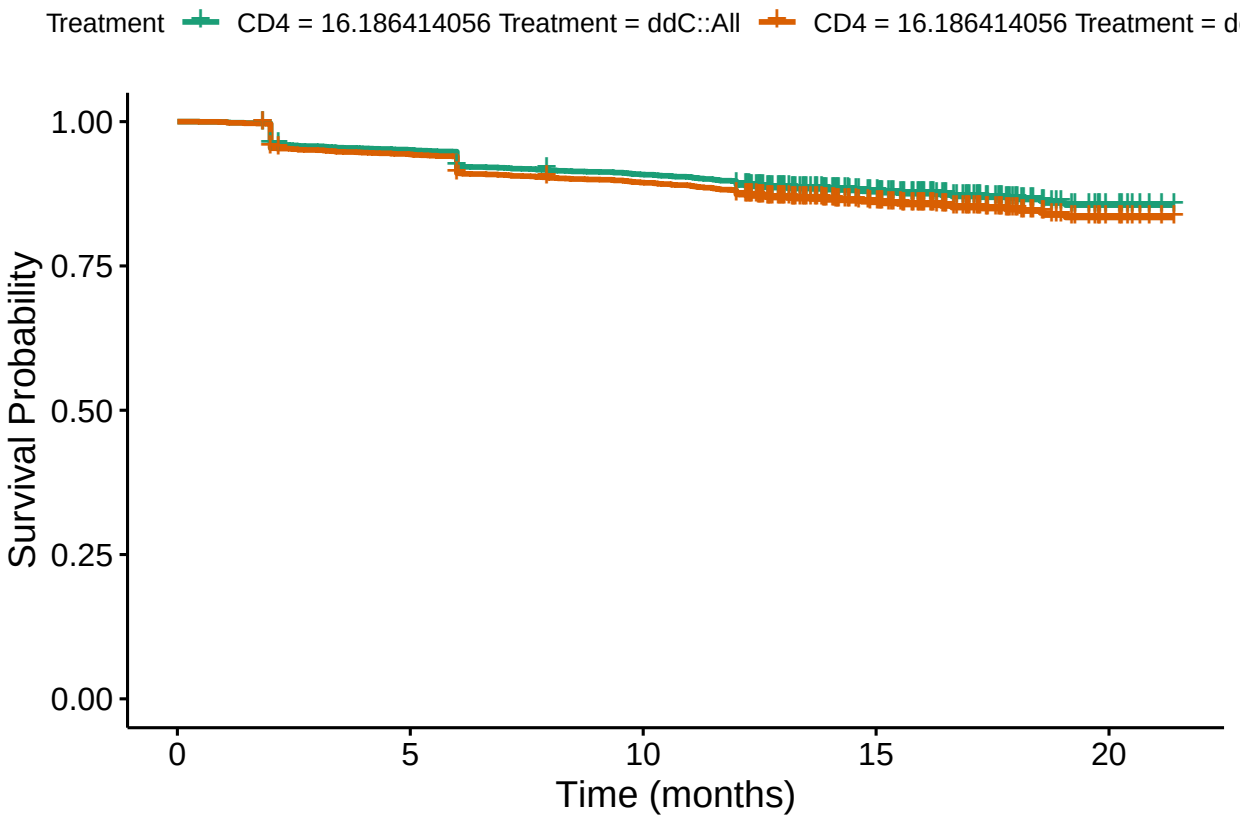

Treatment
CD4 = 7.2801098893 Treatment = ddC::All
CD4 = 7.2801098893 Treatment = d
1.00
0.75
0.50
0.25
0.00
Survival Probability
0
5
10
15
20
Time (months)

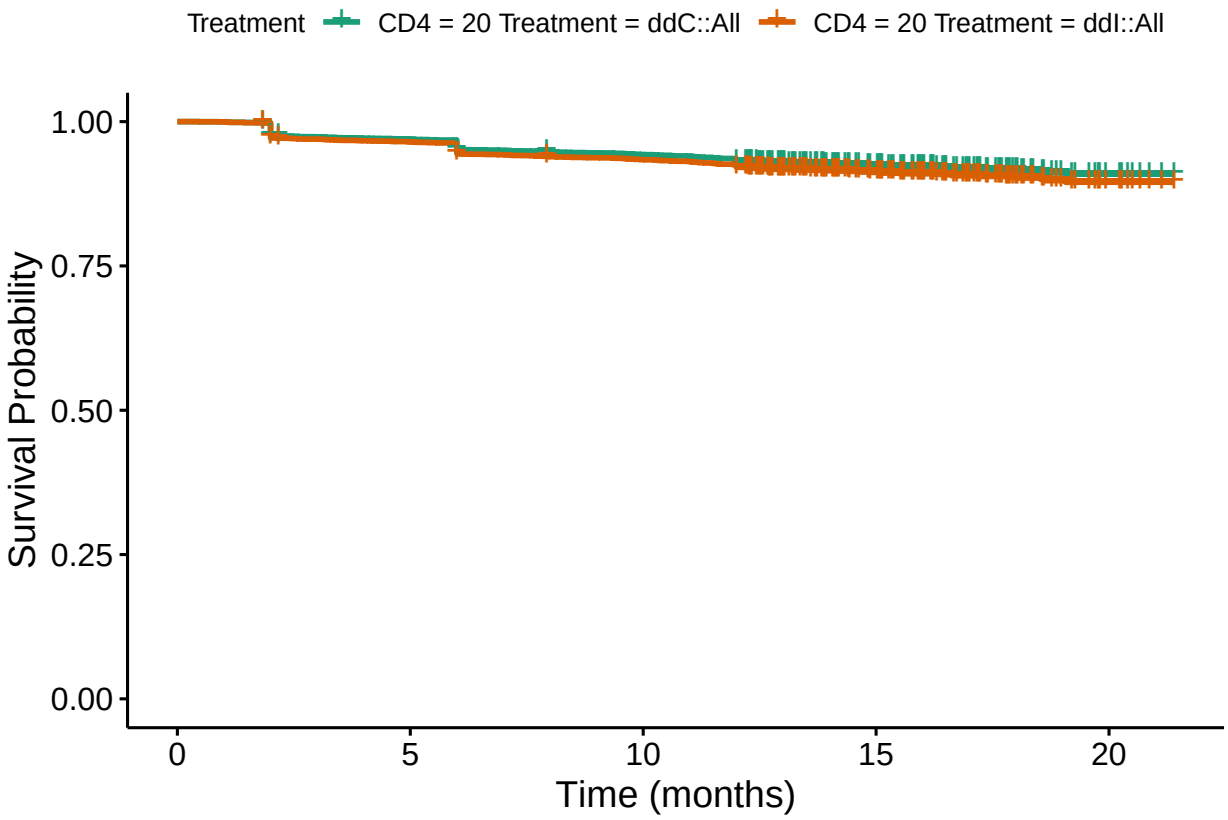

Treatment
CD4 = 20 Treatment = ddC::All
CD4 = 20 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)

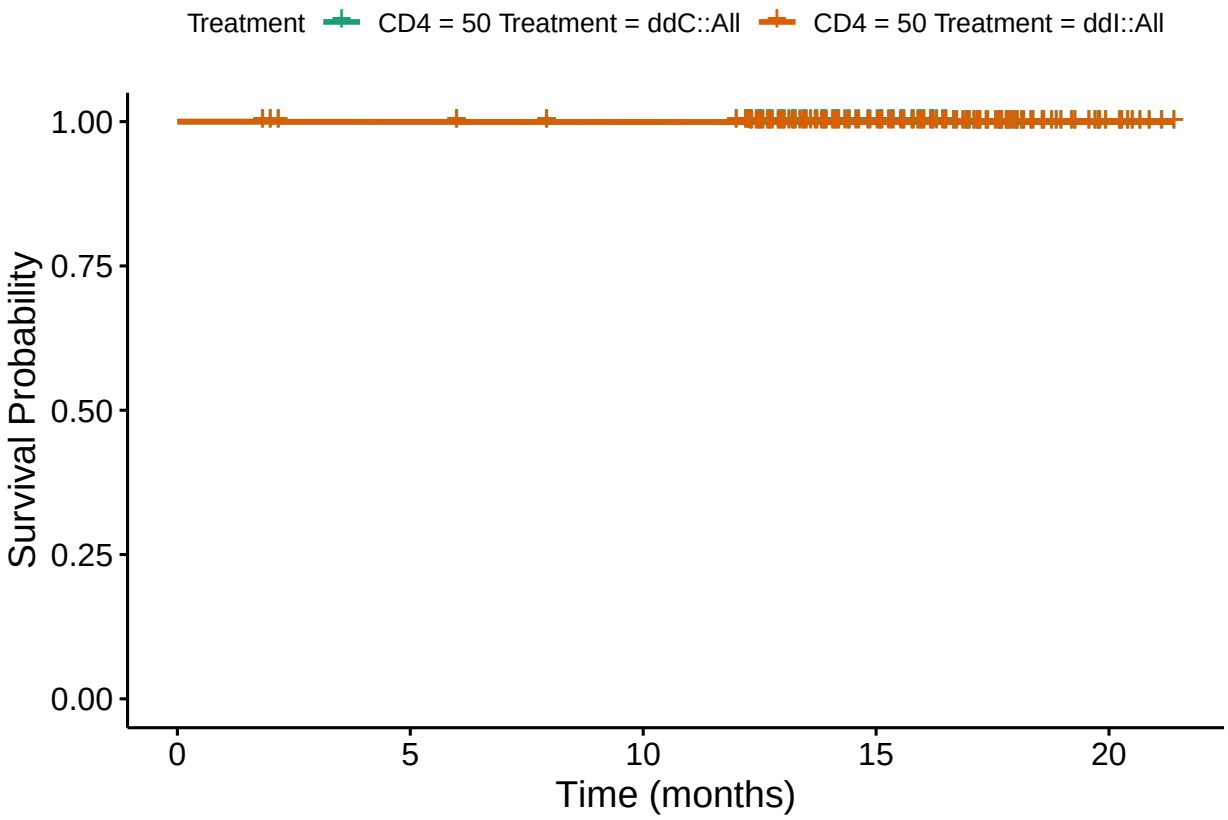

Treatment
CD4 = 50 Treatment = ddC::All
CD4 = 50 Treatment = ddI::All
1.00
0.75
0.50
0.25
0.00
Survival Probability
0
5
10
15
20
Time (months)


Treatment
CD4 = 0 Treatment = ddC::All
CD4 = 0 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)


Treatment
CD4 = 2 Treatment = ddC::All
CD4 = 2 Treatment = ddI::All
Survival Probability
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
Time (months)


Treatment
CD4 = 16.186414056 Treatment = ddC::All
CD4 = 16.186414056 Treatment = d
1.00
0.75
0.50
0.25
0.00
Survival Probability
0
5
10
15
20
Time (months)


Treatment
CD4 = 20 Treatment = ddC::All
CD4 = 20 Treatment = ddI::All
1.00
0.75
0.50
0.25
0.00
Survival Probability
0
5
10
15
20
Time (months)




```r
# We take only one subject and then we vary the cd4 value and the treamtment used
subject_rows <- c(3)

all_survival_results <- data.frame(
  subject = numeric(),
  cd4 = numeric(),
  treatment = character(),
  time = numeric(),
  survival_prob = numeric()
)

for (subject_row in subject_rows) {
  cat("\nAnalyzing Subject:", subject_row, "\n")

  base_data <- test_data[subject_row, ]
  cd4_values <- seq(from = 0, to = 50, length.out = 100)

  for (cd4_value in cd4_values) {
    new_data_cd4 <- base_data
    new_data_cd4$cd4 <- cd4_value

    survival_curves <- list()

    for (treatment_value in c('ddC', 'ddI')) {
      new_data_cd4$treatment <- treatment_value
```

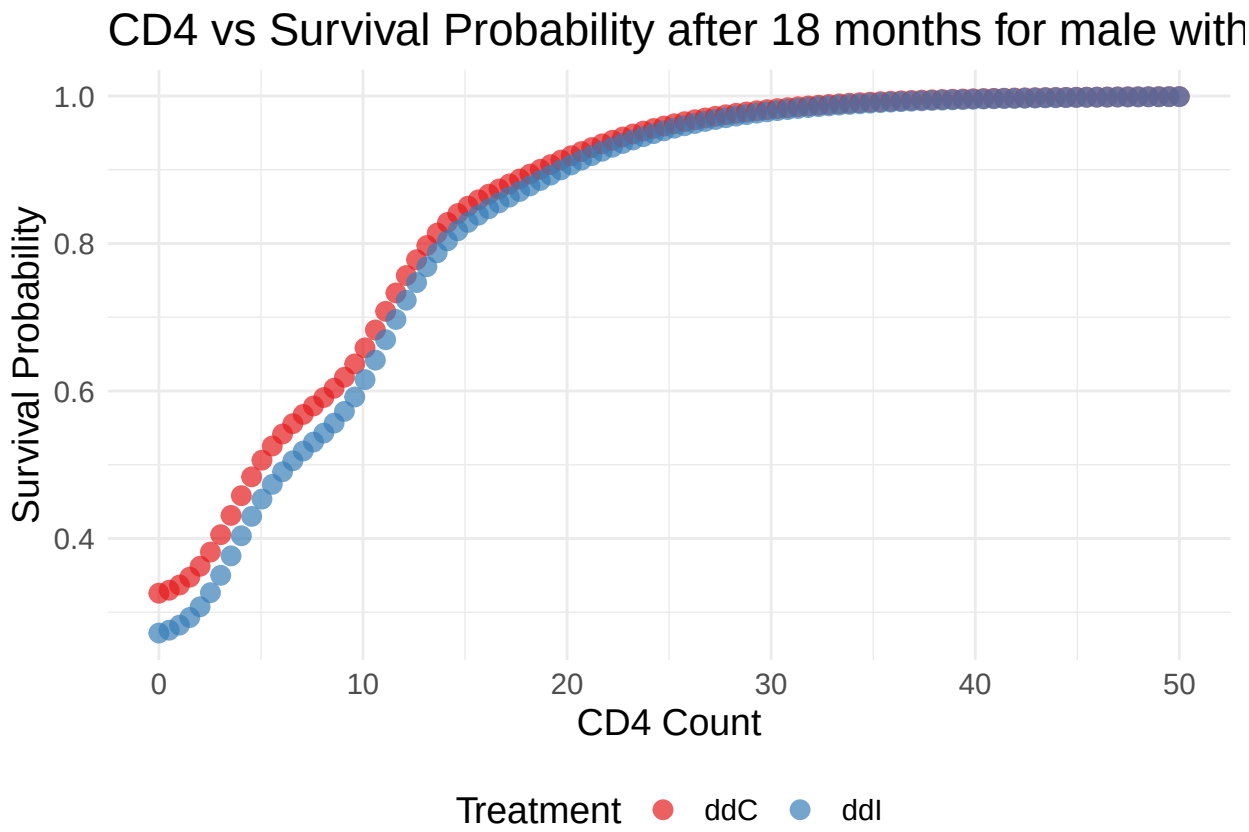
```
      survival_curve <- survfit(cox_model_nonlinear, newdata = new_data_cd4)

      survival_prob_18_months <- summary(survival_curve, times = 18)$surv

      # Ajouter les résultats au tableau
      all_survival_results <- rbind(
        all_survival_results,
        data.frame(
          subject = subject_row,
          cd4 = cd4_value,
          treatment = treatment_value,
          time = 18,
          survival_prob = survival_prob_18_months
        )
      )

      curve_label <- paste("CD4 =", cd4_value, "Treatment =", treatment_value)
      survival_curves[[curve_label]] <- survival_curve

      # cat("Subject:", subject_row,
      #   "- CD4:", cd4_value,
      #    "- Treatment:", treatment_value,
      #    "- Survival Probability at 18 months:", survival_prob_18_months, "\n")
    }

  }
}
```

```
##
## Analyzing Subject: 3
```

The probability that these individuals will survive beyond 18 months is stocked in the data : all_survival_results

Let's analyse this results on a plot :

```
# Plot cd4 vs survival_prob with treatment as color
ggplot(all_survival_results, aes(x = cd4, y = survival_prob, color = treatment)) +
  geom_point(size = 3, alpha = 0.7) +  # Points with size and transparency
  labs(
    title = "CD4 vs Survival Probability after 18 months for male with noAIDS",
    x = "CD4 Count",
    y = "Survival Probability",
    color = "Treatment"
  ) +
  scale_color_manual(
    values = c("ddC" = "#E41A1C", "ddI" = "#377EB8")  # Custom colors
  ) +
  theme_minimal() +  # Clean theme
  theme(
    text = element_text(size = 14),  # Adjust text size
    legend.position = "bottom"      # Move legend to the bottom
  )
```



Here, we can observe that for the subject 2, with the following fixed characteristics : - prev_infection = noAIDS - sex = male - AZT = intolerance

And the following varying covariates: - cd4 - treatment

We can see that the impact of the treatment ddC or ddI depends on the cd4 value. Indeed, when cd4 value exceeds 30, there is no distinction between the 2 treatment. However, when the cd4 value is between 0 and 10, we observe a slightly but significant improvement for the subjects who have taken the ddC treatment in terms of survival probability.

Let's now try to plot the same curve for an individual with the following fixed characteristics : - prev_infection = AIDS - sex = male - AZT = intolerance

```
# We keep the subject number 25 in our test set as he is a male with AIDS.
test_data[25, ]
```

```
##     subject  time death      cd4 start stop treatment  sex prev_infection
## 149      55 12.27     0 3.162278     2    6       ddC male           AIDS
##             azt
## 149 intolerance
```

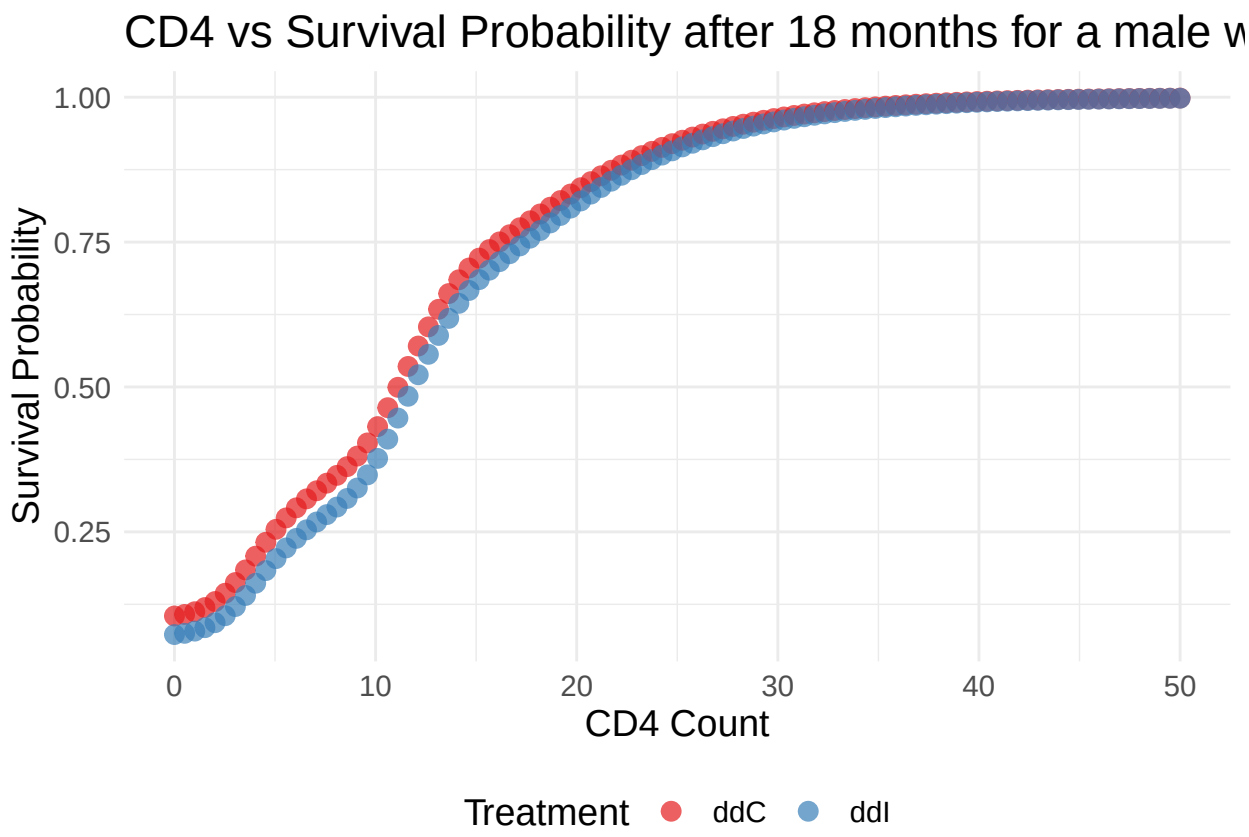
```
# We take only one subject and then we vary the cd4 value and the treamtment used
subject_rows <- c(25)

all_survival_results <- data.frame(
  subject = numeric(),
  cd4 = numeric(),
  treatment = character(),
```

```
  time = numeric(),
  survival_prob = numeric()
)

for (subject_row in subject_rows) {
  cat("\nAnalyzing Subject:", subject_row, "\n")

  base_data <- test_data[subject_row, ]
  cd4_values <- seq(from = 0, to = 50, length.out = 100)

  for (cd4_value in cd4_values) {
    new_data_cd4 <- base_data
    new_data_cd4$cd4 <- cd4_value

    survival_curves <- list()

    for (treatment_value in c('ddC', 'ddI')) {
      new_data_cd4$treatment <- treatment_value

      survival_curve <- survfit(cox_model_nonlinear, newdata = new_data_cd4)

      survival_prob_18_months <- summary(survival_curve, times = 18)$surv

      # Ajouter les résultats au tableau
      all_survival_results <- rbind(
        all_survival_results,
        data.frame(
          subject = subject_row,
          cd4 = cd4_value,
          treatment = treatment_value,
          time = 18,
          survival_prob = survival_prob_18_months
        )
      )

      curve_label <- paste("CD4 =", cd4_value, "Treatment =", treatment_value)
      survival_curves[[curve_label]] <- survival_curve

      # cat("Subject:", subject_row,
      #   "- CD4:", cd4_value,
      #    "- Treatment:", treatment_value,
      #    "- Survival Probability at 18 months:", survival_prob_18_months, "\n")
    }

  }
}
```

```
##
## Analyzing Subject: 25
```

```
# Plot cd4 vs survival_prob with treatment as color
ggplot(all_survival_results, aes(x = cd4, y = survival_prob, color = treatment)) +
  geom_point(size = 3, alpha = 0.7) +  # Points with size and transparency
```

```r
labs(
  title = "CD4 vs Survival Probability after 18 months for a male with AIDS",
  x = "CD4 Count",
  y = "Survival Probability",
  color = "Treatment"
) +
scale_color_manual(
  values = c("ddC" = "#E41A1C", "ddI" = "#377EB8")  # Custom colors
) +
theme_minimal() +  # Clean theme
theme(
  text = element_text(size = 14),  # Adjust text size
  legend.position = "bottom"       # Move legend to the bottom
)
```



We here still observe a slightly improvement by taking the treatment ddC compared to the ddI while having a cd4 value under 20, for a male having AIDS.

Thus, for both subject with or without AIDS and a CD4 level under 20, the best treatment according to our non linear model seems to be the ddC one. However, when the subject has a CD4 level of 30 and above, there is no significant differences between both treatments.

Question 11

For a patient as the previous case, but who has been alive for 12 months since the beginning of the study meaning that the time points at which the CD4 cells count was recorded is 12 months, the estimated

probability of being alive for another 6 months is 0.9413551.

The difference between this probability and the previous scenario, where the patient had no time alive before the study, and then the CD4 cells count was recorded is 0 months, is 0.07864922.

In theory, the addition of the time already survived reduces the time frame for estimating survival, impacting the probability. In practice, this signifies that patients who have already survived a certain duration exhibit a higher likelihood of continued survival within a shorter time horizon compared to those without a history of infection.

```
# Patient profile

new_covariates <- data.frame(
  subject = 2,
  time = 19,
  death = as.factor(0),
  cd4 = 0, # cd4 will move
  start = 12, # We start at 0 month
  stop = 18,
  treatment = as.factor('ddI'), # treatment will move
  sex = as.factor('male'),
  prev_infection = as.factor("noAIDS"),
  azt = as.factor('intolerance')
)

new_covariates
```

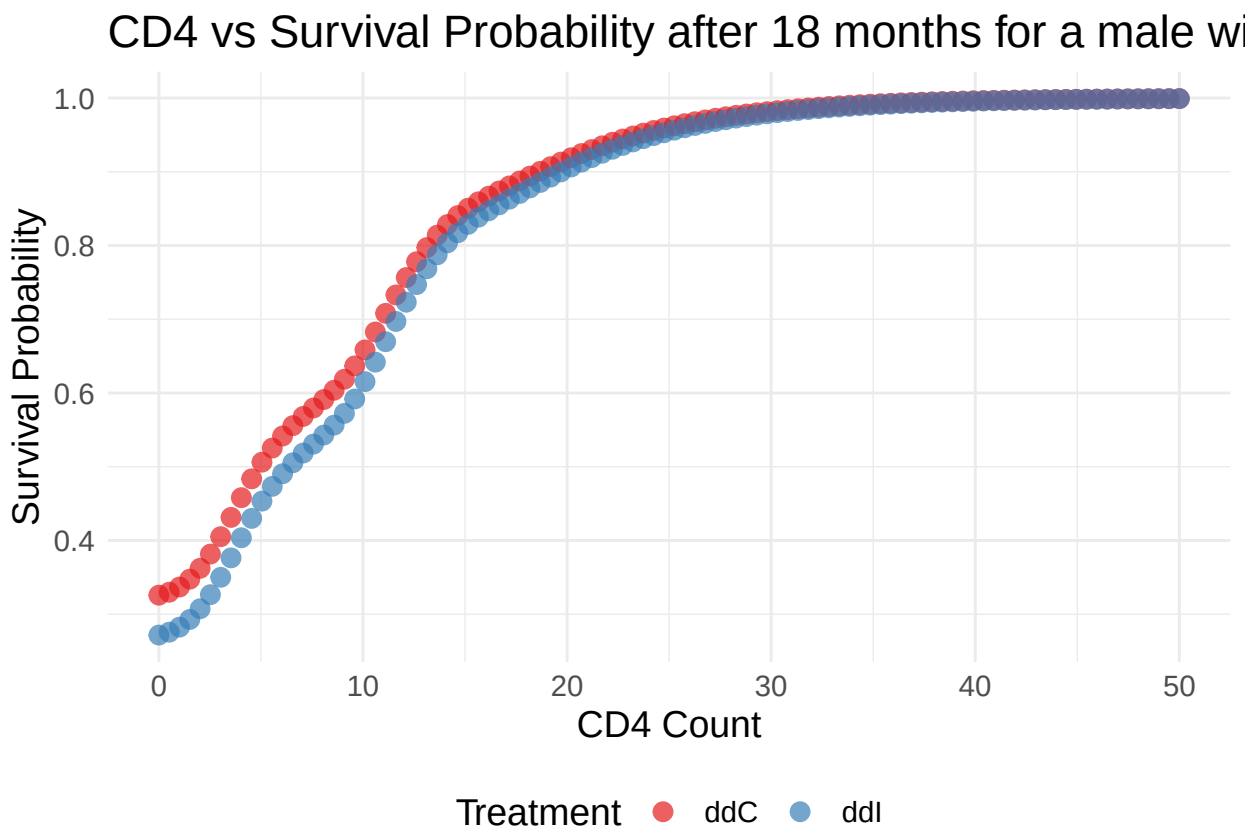
```
##   subject time death cd4 start stop treatment  sex prev_infection         azt
## 1       2   19     0   0    12   18       ddI male         noAIDS intolerance
```

```
# We take only one subject and then we vary the cd4 value and the treamtment used
subject_rows <- c(25)

all_survival_results <- data.frame(
  subject = numeric(),
  cd4 = numeric(),
  treatment = character(),
  time = numeric(),
  survival_prob = numeric()
)

for (subject_row in subject_rows) {
  cat("\nAnalyzing Subject:", subject_row, "\n")

  base_data <- new_covariates
  cd4_values <- seq(from = 0, to = 50, length.out = 100)

  for (cd4_value in cd4_values) {
    new_data_cd4 <- base_data
    new_data_cd4$cd4 <- cd4_value

    survival_curves <- list()

    for (treatment_value in c('ddC', 'ddI')) {
```

```
      new_data_cd4$treatment <- treatment_value

      survival_curve <- survfit(cox_model_nonlinear, newdata = new_data_cd4)

      survival_prob_18_months <- summary(survival_curve, times = 18)$surv

      # Ajouter les résultats au tableau
      all_survival_results <- rbind(
        all_survival_results,
        data.frame(
          subject = subject_row,
          cd4 = cd4_value,
          treatment = treatment_value,
          time = 18,
          survival_prob = survival_prob_18_months
        )
      )

      curve_label <- paste("CD4 =", cd4_value, "Treatment =", treatment_value)
      survival_curves[[curve_label]] <- survival_curve

      # cat("Subject:", subject_row,
      #   "- CD4:", cd4_value,
      #    "- Treatment:", treatment_value,
      #    "- Survival Probability at 18 months:", survival_prob_18_months, "\n")
    }

  }
}
```

```
##
## Analyzing Subject: 25
```

```
# Plot cd4 vs survival_prob with treatment as color
ggplot(all_survival_results, aes(x = cd4, y = survival_prob, color = treatment)) +
  geom_point(size = 3, alpha = 0.7) +  # Points with size and transparency
  labs(
    title = "CD4 vs Survival Probability after 18 months for a male with noAIDS",
    x = "CD4 Count",
    y = "Survival Probability",
    color = "Treatment"
  ) +
  scale_color_manual(
    values = c("ddC" = "#E41A1C", "ddI" = "#377EB8")  # Custom colors
  ) +
  theme_minimal() +  # Clean theme
  theme(
    text = element_text(size = 14),  # Adjust text size
    legend.position = "bottom"      # Move legend to the bottom
  )
```



New patient profile, with the first "time obs" after 12 months being alive.

```
# Patient profile

new_covariates <- data.frame(
  subject = 2,
  time = 0,
  death = as.factor(0),
  cd4 = 0, # cd4 will move
  start = 12, # We start at 12 months
  stop = 20,
  treatment = as.factor('ddI'), # treatment will move
  sex = as.factor('male'),
  prev_infection = as.factor("noAIDS"),
  azt = as.factor('intolerance')
)

new_covariates
```

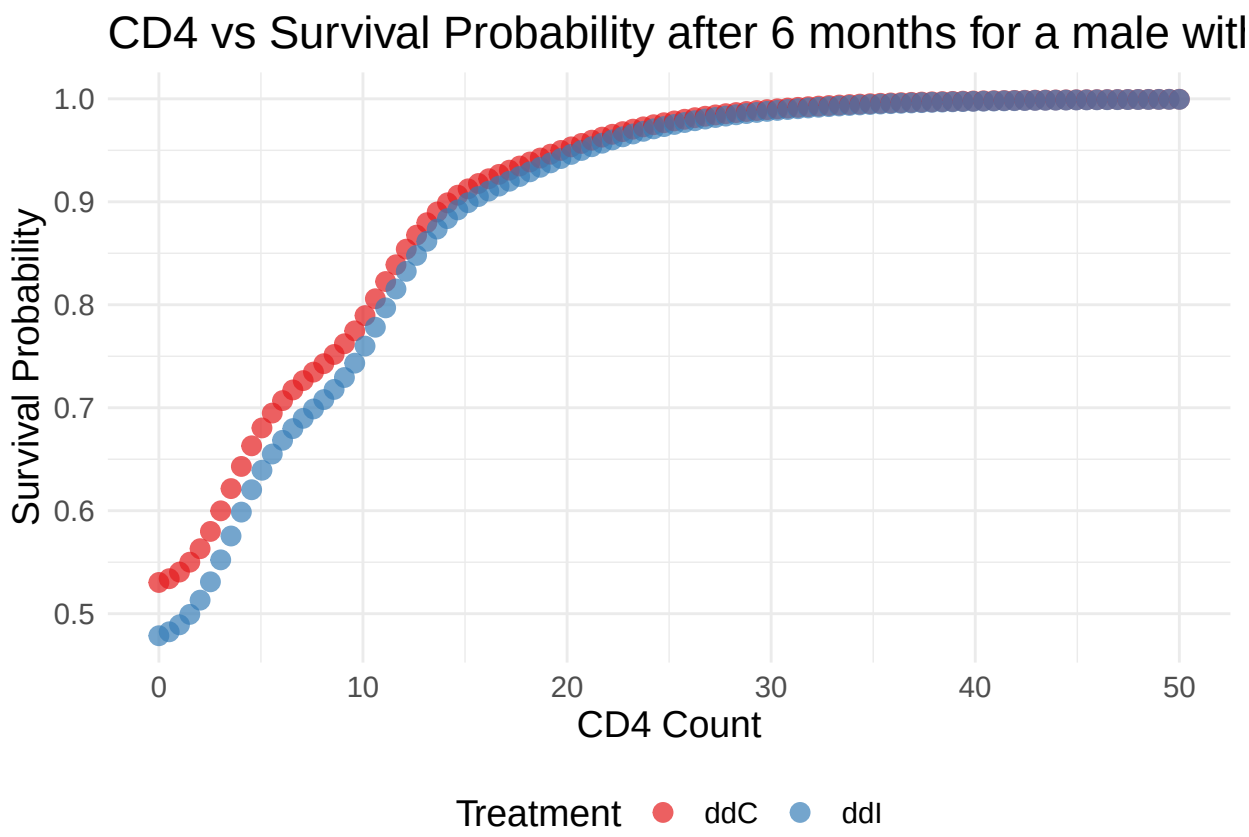
```
##   subject time death cd4 start stop treatment  sex prev_infection         azt
## 1       2    0     0   0    12   20       ddI male         noAIDS intolerance
```

```
# We take only one subject and then we vary the cd4 value and the treamtment used
subject_rows <- c(25)

all_survival_results <- data.frame(
```

```r
  subject = numeric(),
  cd4 = numeric(),
  treatment = character(),
  time = numeric(),
  survival_prob = numeric()
)

for (subject_row in subject_rows) {
  cat("\nAnalyzing Subject:", subject_row, "\n")

  base_data <- new_covariates
  cd4_values <- seq(from = 0, to = 50, length.out = 100)

  for (cd4_value in cd4_values) {
    new_data_cd4 <- base_data
    new_data_cd4$cd4 <- cd4_value

    survival_curves <- list()

    for (treatment_value in c('ddC', 'ddI')) {
      new_data_cd4$treatment <- treatment_value

      survival_curve <- survfit(cox_model_nonlinear, newdata = new_data_cd4)

      survival_prob_6_months <- summary(survival_curve, times = 6)$surv

      # Ajouter les résultats au tableau
      all_survival_results <- rbind(
        all_survival_results,
        data.frame(
          subject = subject_row,
          cd4 = cd4_value,
          treatment = treatment_value,
          time = 6,
          survival_prob = survival_prob_6_months
        )
      )

      curve_label <- paste("CD4 =", cd4_value, "Treatment =", treatment_value)
      survival_curves[[curve_label]] <- survival_curve

      # cat("Subject:", subject_row,
      #   "- CD4:", cd4_value,
      #    "- Treatment:", treatment_value,
      #    "- Survival Probability at 6 months:", survival_prob_18_months, "\n")
    }

  }
}
```

```
##
## Analyzing Subject: 25
```

```r
# Plot cd4 vs survival_prob with treatment as color
ggplot(all_survival_results, aes(x = cd4, y = survival_prob, color = treatment)) +
  geom_point(size = 3, alpha = 0.7) +  # Points with size and transparency
  labs(
    title = "CD4 vs Survival Probability after 6 months for a male with noAIDS",
    x = "CD4 Count",
    y = "Survival Probability",
    color = "Treatment"
  ) +
  scale_color_manual(
    values = c("ddC" = "#E41A1C", "ddI" = "#377EB8")  # Custom colors
  ) +
  theme_minimal() +  # Clean theme
  theme(
    text = element_text(size = 14),  # Adjust text size
    legend.position = "bottom"       # Move legend to the bottom
  )
```



Question 12

Let's create a Random Forest model, with the same formula as the second Cox model, as Random Forest is supposed to take in account non linear relations.

```r
# We only use the right censoring as the model do not take start/stop format
random_forest_model <- rfsrc(
  Surv(time, death) ~ pspline(cd4) + sex + treatment + prev_infection + azt,
  data = train_data
```

```
)

# Print the model summary
print(random_forest_model)
```

```
##                          Sample size: 1125
##                     Number of events: 795, 330
##                      Number of trees: 500
##            Forest terminal node size: 15
##        Average no. of terminal nodes: 54.004
## No. of variables tried at each split: 3
##               Total no. of variables: 5
##        Resampling used to grow trees: swor
##     Resample size used to grow trees: 711
##                             Analysis: RSF
##                               Family: surv-CR
##                       Splitting rule: logrankCR *random*
##        Number of random split points: 10
##    (OOB) Requested performance error: 0.49564026, 0.46242977
```

The model seems to be satisfying as the requested performance error is estimate to 0.49-0.46

Let's fine tune our model :

```
param_grid <- expand.grid(
  ntree = c(200, 500, 700),
  mtry = c(2, 4, 6),
  maxdepth = c(5, 10, 15),
  nodesize = c(5, 10, 20)
)

tuned_rf_model <- tune.rfsrc(
  formula = Surv(time, death) ~ pspline(cd4) + sex + treatment + prev_infection + azt,
  data = train_data,
  paramgrid = param_grid,
  nodedepth = 15
)

best_rf_model <- tuned_rf_model$optimal
```

With the grid search method, we optimize the hyper-parameters of our Random Forest model, especially about the number of trees, the depth, the size of node and the number of random variables selected at each tree division

```
tuned_rf_model$optimal
```

```
## nodesize     mtry
##        1        5
```

Then we create an other random forest model with the best parameters we just found.

```
formula = Surv(time, death) ~ pspline(cd4) + sex + treatment + prev_infection + azt
random_forest_model_2 <- rfsrc(formula , data = train_data, ntree=500, mtry = 4, nodesize = 15, nodedep
random_forest_model_2
```

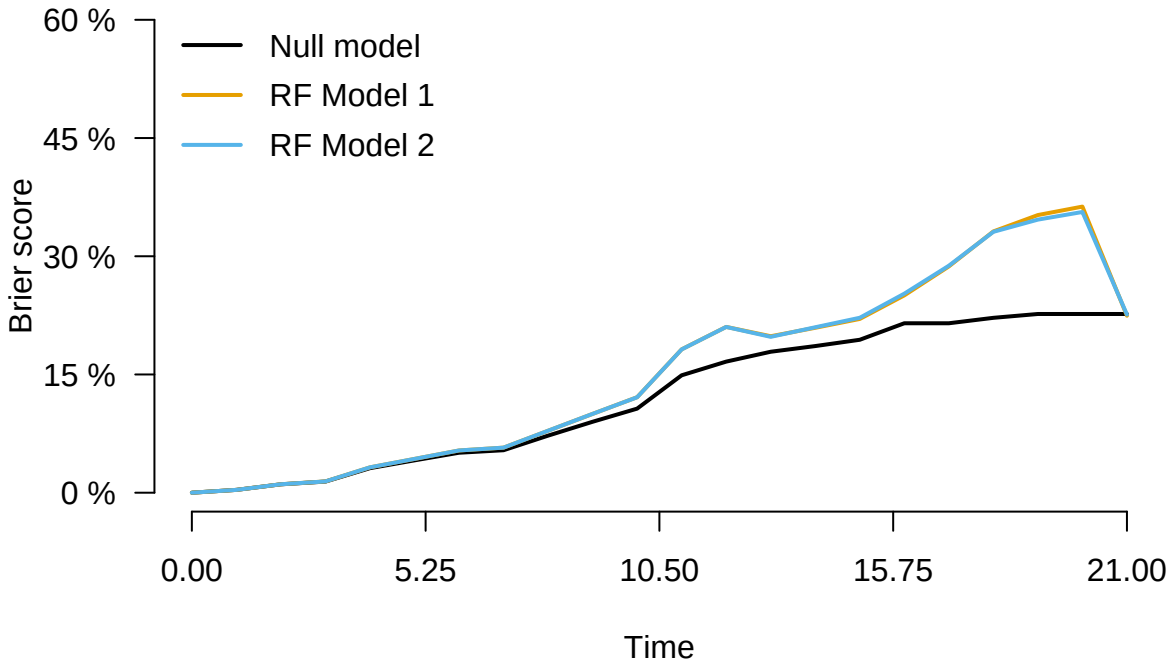
```
##                          Sample size: 1125
##                     Number of events: 795, 330
##                      Number of trees: 500
##            Forest terminal node size: 15
##        Average no. of terminal nodes: 53.474
## No. of variables tried at each split: 4
##               Total no. of variables: 5
##        Resampling used to grow trees: swor
##     Resample size used to grow trees: 711
##                             Analysis: RSF
##                               Family: surv-CR
##                       Splitting rule: logrankCR *random*
##        Number of random split points: 10
##    (OOB) Requested performance error: 0.49388179, 0.45861656
```

As expected, results is a bit better than before, according to the error.

Let's create a wrapper to be able to use the brier score :

```
predictRisk.custom_rfsrc <- function(object, newdata, times, cause = 1) {
  # Ensure cause is not NULL
  if (is.null(cause)) {
    stop("The 'cause' parameter must be provided for competing risks models.")
  }
  # Ensure cause is numeric
  cause <- as.numeric(cause)
  # Call predictRisk from randomForestSRC
  predictRisk(object$model, newdata = newdata, times = times, cause = cause)
}

rf_model_1 <- list(model = random_forest_model)
class(rf_model_1) <- "custom_rfsrc"

rf_model_2 <- list(model = random_forest_model_2)
class(rf_model_2) <- "custom_rfsrc"

# Test risk predictions for the first model
risk_preds_1 <- predictRisk(rf_model_1, newdata = test_data, times = 0:21.4, cause = 1)
# print(head(risk_preds_1))

# Test risk predictions for the second model
risk_preds_2 <- predictRisk(rf_model_2, newdata = test_data, times = 0:21.4, cause = 1)
# print(head(risk_preds_2))

brier_rf <- Score(
  list("RF Model 1" = rf_model_1, "RF Model 2" = rf_model_2),
  formula = Hist(time, death) ~ 1,
  data = test_data,
  times = 0:21.4,
```

```
  metrics = "brier",
  cause = 1 # Pass cause explicitly for competing risks
)

# Plot the Brier scores
plot_brier_rf <- plotBrier(brier_rf, ylim = c(0, 0.6))
```




```
# print(plot_brier_rf)
```

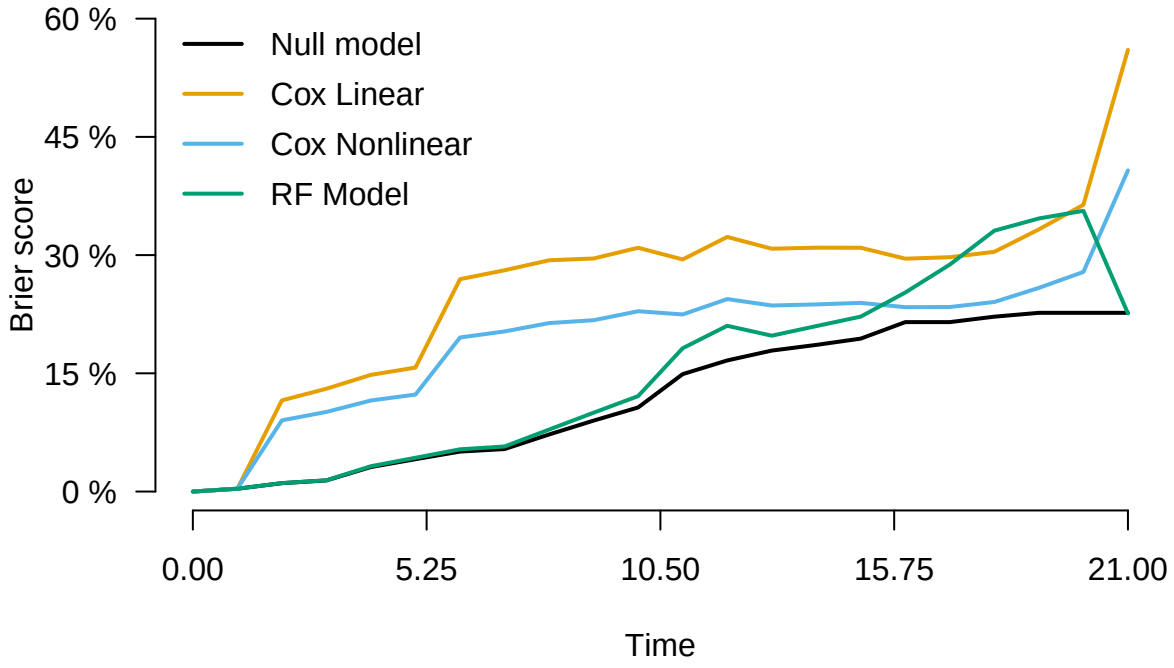
The brier plot also shows that the second model is slightly better.

Question 13

Let's plot our best random forest model vs our best non linear cox model :

```
# Compute the Brier scores for all models
brier <- Score(
  list(
    "Cox Linear" = cox_model_linear,
    "Cox Nonlinear" = cox_model_nonlinear,
    # "RF Model 1" = rf_model_1,
    "RF Model" = rf_model_2
  ),
  formula = Hist(time, death) ~ 1,
```

```
  data = test_data,
  times = 0:21.4,
  metrics = "brier",
  cause = 1
)

# Plot the Brier score comparison
plot_brier <- plotBrier(brier, ylim = c(0, 0.6))
```




```
# Display the plot
# print(plot_brier)
```

By comparing visually the brier plots, Random Forest seems to be the best model for prediction. Furthermore, relations between variables are not linear and Random Forest is better to handle this type of relation.

```
# Make the computation on our data
time_range <- seq(0, 21.4, by = 0.1) # Gamme de temps
models <- list(
  "Random Forest" = rf_model_1,
  "Random Forest2" = rf_model_2
)

embedded_scores <- calculate_embedded_score(models, test_data, time_range)
```

```r
# Printing the results
for (model_name in names(embedded_scores)) {
  cat("Model:", model_name, "\n")
  cat("Integrated Brier Score (IBS):", embedded_scores[[model_name]]$IBS, "\n")
}
```

```
## Model: Random Forest
## Integrated Brier Score (IBS): 3.373954
## Model: Random Forest2
## Integrated Brier Score (IBS): 3.36624
```