DS 223: Marketing Analytics

Homework 3 - Survival Analysis

Alexander Shahramanyan

April 28, 2024

To build AFT models, we first need to load the data. We will be working with the Telco Customer Churn dataset, which has the following columns:

- ID: subscriber's ID
- region: region code
- tenure: lifetime (in months)
- age: subscriber's age
- marital: subscriber's marital status
- address: number of years living in the same address
- income: subscriber's annual income (K)
- ed: subscriber's education level
- retire: retired (Yes/No)
- gender: subscriber's gender (Male/Female)
- voice: voice service (Yes/No)
- internet: internet service (Yes/No)
- forward: call forwarding (Yes/No)
- custcat: customer category
- churn: whether the customer churned (Yes/No)

```r
# Read the CSV file
telco <- read.csv("telco.csv")
telco$churn = ifelse(telco$churn=='Yes', 1, 0)

head(telco)
```

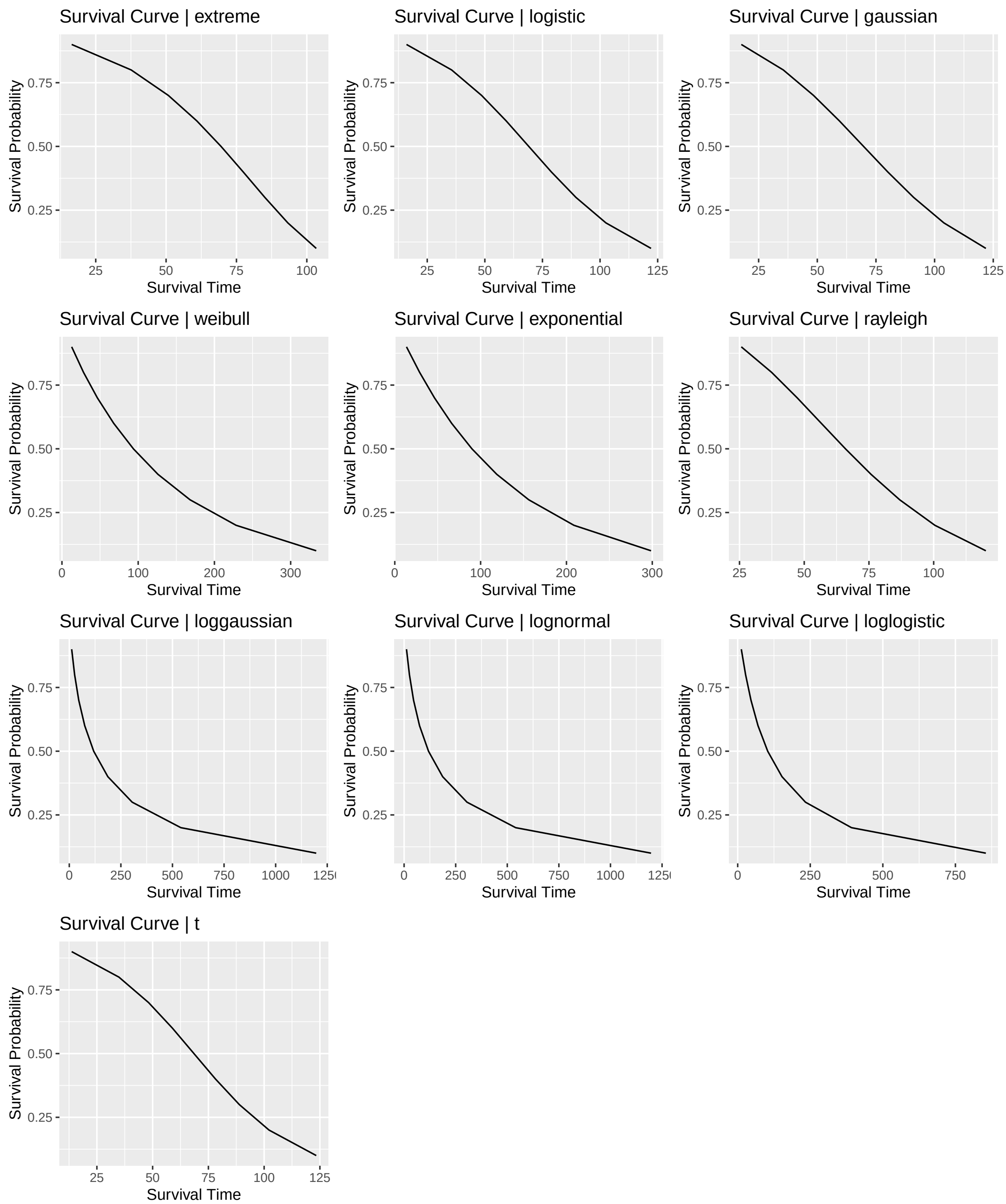
```
##   ID region tenure age   marital address income                           ed
## 1  1 Zone 2     13  44   Married       9     64              College degree
## 2  2 Zone 3     11  33   Married       7    136    Post-undergraduate degree
## 3  3 Zone 3     68  52   Married      24    116 Did not complete high school
## 4  4 Zone 2     33  33 Unmarried      12     33          High school degree
## 5  5 Zone 2     23  30   Married       9     30 Did not complete high school
## 6  6 Zone 2     41  39 Unmarried      17     78          High school degree
##   retire gender voice internet forward       custcat churn
## 1     No   Male    No       No     Yes Basic service     1
## 2     No   Male   Yes       No     Yes Total service     1
## 3     No Female    No       No      No  Plus service     0
## 4     No Female    No       No      No Basic service     1
## 5     No   Male    No       No     Yes  Plus service     0
## 6     No Female    No       No      No  Plus service     0
```

Now, we will build basic models (intercept-only) with all the different distributions available in `survreg` package.

```r
surv_obj = Surv(time=telco$tenure, event=telco$churn)
reg_models <- list()

for(distribution in names(survreg.distributions)){
    # get the regression model
    reg_m = survreg(formula=surv_obj~1, dist=distribution)

    # print the summary
    # summary(reg_m)

    # add reg_m to reg_models
    reg_models[[distribution]] <- reg_m
}
```

As we have the models now, let's visualize the probability of churn during customer lifetime using the models in different plots and have an initial look at them.

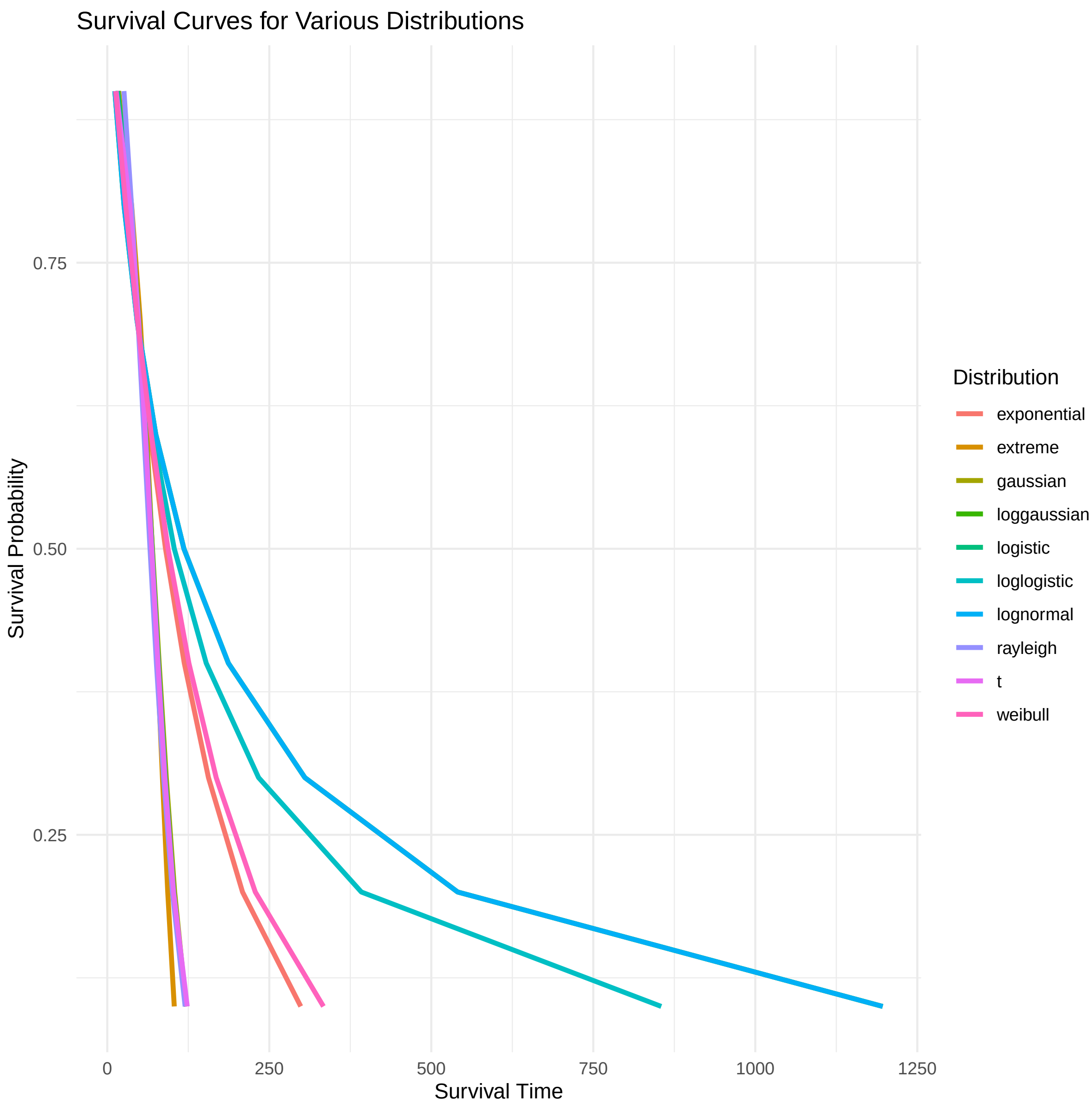
```r
# Initialize an empty list for storing plots
plot_list <- list()

for (distribution in names(survreg.distributions)) {
    reg_m <- reg_models[[distribution]]

    probs <- seq(.1, .9, length=9)
    pred <- predict(reg_m, type="quantile", p=1-probs, newdata=data.frame(1))

    df <- data.frame(Time=pred, Probabilities=probs)

    # Generate the plot for current distribution
    p <- ggplot(df, aes(x = Time, y = Probabilities)) +
        geom_line() +
        labs(title = paste("Survival Curve |", distribution),
             x = "Survival Time",
             y = "Survival Probability")

    # Store the plot in the list
    plot_list[[distribution]] <- p
}

# Combine the plots into a grid (4x3) and leave the last two positions blank
plot_grid <- wrap_plots(plot_list, nrow = 4, ncol = 3) +
             plot_spacer() + plot_spacer()

# Print the combined plot grid
print(plot_grid)
```



As we can see, there are indeed some differences between the models. We can plot all the model curves in one graph to be able to compare the models.

```
# Initialize an empty data frame for storing combined data
combined_df <- data.frame()

for (distribution in names(survreg.distributions)) {
    reg_m <- reg_models[[distribution]]

    probs <- seq(.1, .9, length=9)
    pred <- predict(reg_m, type="quantile", p=1-probs, newdata=data.frame(1))
```

```
    df <- data.frame(Time = pred, Probabilities = probs, Distribution = distribution)

    # Combine data
    combined_df <- rbind(combined_df, df)
}

# Generate a single plot with curves for all distributions
p <- ggplot(combined_df, aes(x = Time, y = Probabilities, color = Distribution)) +
    geom_line(linewidth=1.2) +
    labs(title = "Survival Curves for Various Distributions",
         x = "Survival Time",
         y = "Survival Probability") +
    theme_minimal()

# Print the plot
print(p)
```



Lognormal seems to be the better one, however, let's also compare the AIC and BIC.

```r
combined_scores <- data.frame(Distribution = character(), AIC = numeric(), BIC = numeric())


for(distribution in names(survreg.distributions)){
    reg_m <- reg_models[[distribution]]

    extracted_scores <- extract(
      reg_m,
      include.aic = TRUE,
      include.bic = TRUE
    )

    # Extract AIC and BIC names
    score_names <- extracted_scores@gof.names

    # Find indices of AIC and BIC in the names
    aic_index <- which(score_names == "AIC")
    bic_index <- which(score_names == "BIC")

    # Extract AIC and BIC scores
    aic <- extracted_scores@gof[aic_index]
    bic <- extracted_scores@gof[bic_index]

    combined_scores <- rbind(combined_scores,
                             tibble(Distribution = distribution,
                                    AIC = aic,
                                    BIC = bic))
}
# Order the scores by AIC and BIC
combined_scores_ordered <- combined_scores[order(combined_scores$AIC, combined_scores$BIC), ]

# Print the combined scores data frame
print(combined_scores_ordered)
```

```
## # A tibble: 10 x 3
##    Distribution   AIC   BIC
##    <chr>        <dbl> <dbl>
##  1 loggaussian  3209. 3219.
##  2 lognormal    3209. 3219.
##  3 loglogistic  3214. 3224.
##  4 exponential  3216. 3221.
##  5 weibull      3217. 3227.
##  6 gaussian     3433. 3443.
##  7 logistic     3472. 3482.
##  8 rayleigh     3481. 3486.
##  9 extreme      3498. 3508.
## 10 t            3500. 3510.
```

As we can see the models with `loggaussian` and `lognormal` distributions have lower AIC and BIC. Let's pick the `lognormal` one go on with it. We'll train a new model adding some of the variables to it. But first, we'll define the order to some of the factor variables.

```r
# Define the education order
ed_order <- c("Did not complete high school", "High school degree", "Some college", "College degree"

# Apply the education order to the respective variable
telco$ed <- factor(telco$ed, levels = ed_order)
```

Let's add gender all the columns to the model and then remove those that are not statistically significant to the model (we assume that the p-values of the models with only one covariate and the model with said covariate and some others are not very different).

```
reg_f= survreg(surv_obj ~ region + age + marital + address + income + ed + retire + gender + voice +
               data=telco, dist="lognormal")
summary(reg_f)
```

```
##
## Call:
## survreg(formula = surv_obj ~ region + age + marital + address +
##     income + ed + retire + gender + voice + internet + forward +
##     custcat, data = telco, dist = "lognormal")
##                                Value Std. Error     z       p
## (Intercept)                  2.73588    0.31345  8.73 < 2e-16
## regionZone 2                -0.09704    0.14277 -0.68   0.497
## regionZone 3                 0.04822    0.14154  0.34   0.733
## age                          0.03267    0.00725  4.50 6.7e-06
## maritalUnmarried            -0.45515    0.11543 -3.94 8.0e-05
## address                      0.04254    0.00890  4.78 1.8e-06
## income                       0.00140    0.00092  1.52   0.129
## edHigh school degree        -0.05768    0.18724 -0.31   0.758
## edSome college              -0.10129    0.20103 -0.50   0.614
## edCollege degree            -0.37361    0.20159 -1.85   0.064
## edPost-undergraduate degree -0.40797    0.26920 -1.52   0.130
## retireYes                    0.02248    0.44407  0.05   0.960
## genderMale                   0.05188    0.11429  0.45   0.650
## voiceYes                    -0.43379    0.16895 -2.57   0.010
## internetYes                 -0.77150    0.14348 -5.38 7.6e-08
## forwardYes                  -0.19813    0.18004 -1.10   0.271
## custcatE-service             1.06642    0.17053  6.25 4.0e-10
## custcatPlus service          0.92495    0.21575  4.29 1.8e-05
## custcatTotal service         1.19860    0.25045  4.79 1.7e-06
## Log(scale)                   0.27577    0.04600  6.00 2.0e-09
##
## Scale= 1.32
##
## Log Normal distribution
## Loglik(model)= -1457   Loglik(intercept only)= -1602.5
##  Chisq= 291.01 on 18 degrees of freedom, p= 3.4e-51
## Number of Newton-Raphson Iterations: 5
## n= 1000
```

As we can see, only the coefficients of `age`, `marital`, `voice`, `internet`, and `custcat` are statistically significant. Let's rebuild the model using only those. Let's also add `ed`, because, I suppose, it might also have some impact on the model (the p-value of some education levels are almost statistically significant).

```
reg_f= survreg(surv_obj ~ age + marital +  voice + internet + custcat,
               data=telco, dist="lognormal")
summary(reg_f)
```

```
##
## Call:
## survreg(formula = surv_obj ~ age + marital + voice + internet +
##     custcat, data = telco, dist = "lognormal")
##                         Value Std. Error     z       p
## (Intercept)            2.1416     0.2346  9.13 < 2e-16
```

```
## age                    0.0576     0.0053 10.86 < 2e-16
## maritalUnmarried      -0.4227     0.1167 -3.62 0.00029
## voiceYes              -0.5279     0.1707 -3.09 0.00198
## internetYes           -0.8980     0.1412 -6.36 2.0e-10
## custcatE-service       1.0905     0.1719  6.34 2.2e-10
## custcatPlus service    0.8823     0.1729  5.10 3.4e-07
## custcatTotal service   1.1313     0.2141  5.28 1.3e-07
## Log(scale)             0.3090     0.0462  6.69 2.3e-11
##
## Scale= 1.36
##
## Log Normal distribution
## Loglik(model)= -1474.1   Loglik(intercept only)= -1602.5
##  Chisq= 256.9 on 7 degrees of freedom, p= 9.4e-52
## Number of Newton-Raphson Iterations: 5
## n= 1000
```

Now, as we have the model, let's have a look at AIC and BIC.

```
extract(
    reg_f,
    include.aic = TRUE,
    include.bic = TRUE
)
```

```
##
##
##                             coef.       s.e.            p
## (Intercept)            2.14158992 0.23455078 6.811916e-20
## age                    0.05760635 0.00530462 1.794220e-27
## maritalUnmarried      -0.42269964 0.11670057 2.922356e-04
## voiceYes              -0.52785517 0.17067406 1.982995e-03
## internetYes           -0.89799989 0.14122107 2.033043e-10
## custcatE-service       1.09048902 0.17190558 2.245524e-10
## custcatPlus service    0.88228952 0.17293336 3.362522e-07
## custcatTotal service   1.13131652 0.21410967 1.265190e-07
## Log(scale)             0.30901747 0.04621185 2.278320e-11
##
##                       GOF dec. places
## AIC              2966.131        TRUE
## BIC              3010.301        TRUE
## Log Likelihood -1474.066        TRUE
## Num. obs.        1000.000       FALSE
```

As we can see the AIC and BIC are better than of the intercept-only model. Now, let's interpret the coefficients. Since we're using a `lognormal` distribution for the model, we'll need to exponentiate the coefficients returned by the model to understand the real effect of the covariates.

```
exp(coef(reg_f))
```

```
##          (Intercept)                  age     maritalUnmarried
##            8.5129618            1.0592979            0.6552754
##             voiceYes          internetYes     custcatE-service
##            0.5898688            0.4073837            2.9757289
##  custcatPlus service custcatTotal service
##            2.4164258            3.0997347
```

Report

As we can see the coefficients for `maritalUnmarried`, `voiceYes`, and `internetYes` are less than one. From this we can conclude that unot married individuals, as well as individuals using voice and/or internet services have less life time, that is, they are more prone to churn earlier compared to married, individuals not using voice and internet services, respectively.

On the contrary, `age` has a coefficient, which is greater than 1. This means as people get older, they are less prone to churn (the lifetime is longer). Same goes for the different customer categories; those using Plus service tend to have longer lifetime (2.4 times more), those using E-service and Total service are even less prone to churn (about 3 times longer lifetime), compared to individuals with Basic service.

CLV

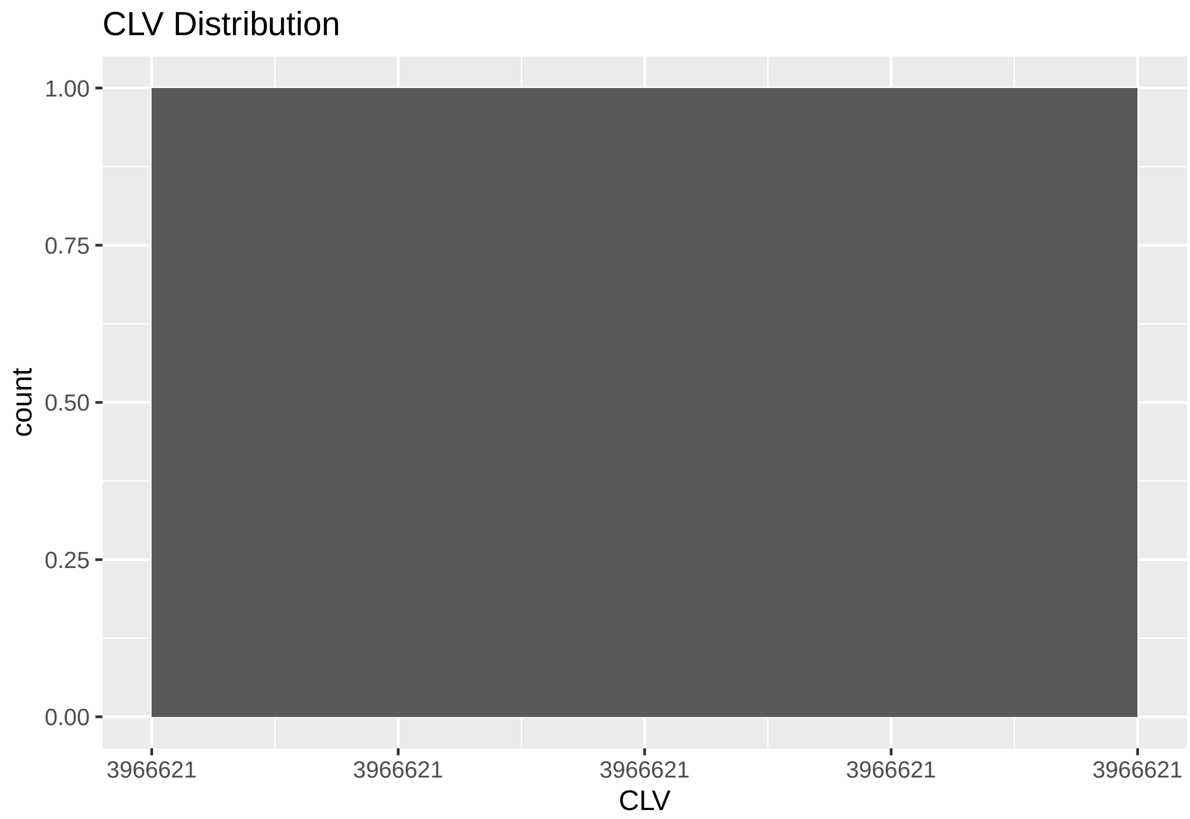
(Something is off here. I didn't have time to understand what the problem is before midnight. My next commits will solve the problem.)

```
pred=predict(reg_f, type="response")

pred_data=data.frame(t(pred))[,0:24]

sequence = seq(1,length(colnames(pred_data)),1)
MM = 1300
r = 0.1

for (num in sequence) {
    pred_data[,num]=pred_data[,num]/(1+r/12)^(sequence[num]-1)
}
```

```
pred_data$CLV=MM*rowSums(pred_data)
summary(pred_data$CLV)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 3966621 3966621 3966621 3966621 3966621 3966621
```

```
ggplot(pred_data,aes(x=CLV))+labs(title = "CLV Distribution")+
geom_histogram()
```




```
telco$CLV = pred_data$CLV
ggplot(telco,aes(x=CLV, color=gender))+
labs(title = "CLV Density By Gender")+
geom_density()
```