Applied_Stats_project_1

Payden Bullis, Hayoung Chen, Solomon Mathew

2025-02-02

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
hospitaldataoriginal <- read.csv("C:\\Users\\payde\\Desktop\\Homework and data sets\\HospitalDurations_2.csv")

#na explore
sum(is.na(hospitaldataoriginal))
```

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

```
#no NA's found.

# Convert Region to factor
hospitaldataoriginal$Region <- as.factor(hospitaldataoriginal$Region)
hospitaldataoriginal$Med.Sc.Aff <- as.factor(hospitaldataoriginal$Med.Sc.Aff)
```

##Plots with color palette

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.4.2
```

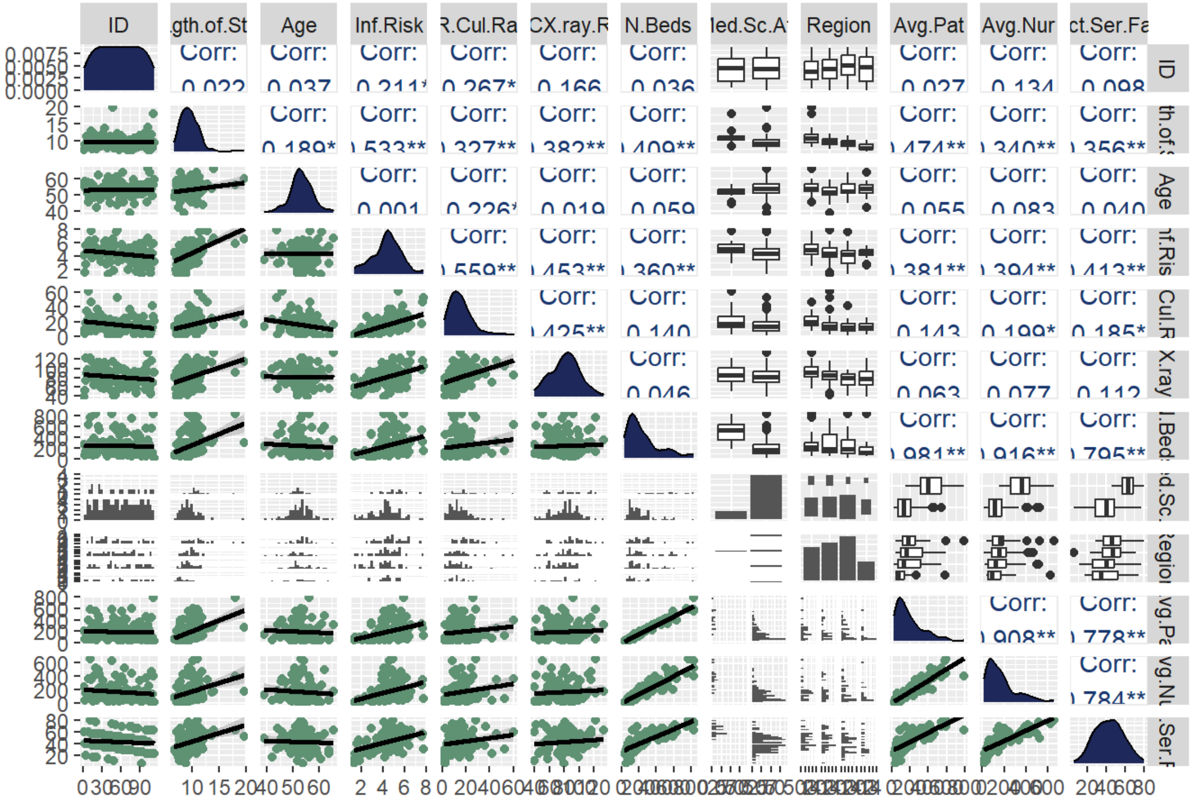
```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
hosp_color <- c("#609175", "#21295C", "#1B3B6F", "#6F7273", "#DCEDFF")

ggpairs(hospitaldataoriginal,
        title = "Correlation Scatterplot Matrix",
        lower = list(continuous = wrap("smooth", color = hosp_color[1])), # Apply custom color to lower panels
        diag = list(continuous = wrap("densityDiag", fill = hosp_color[2])), # Apply custom color to diagonal panels
        upper = list(continuous = wrap("cor", color = hosp_color[3]))) # Apply custom color to upper panels
```

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

Correlation Scatterplot Matrix

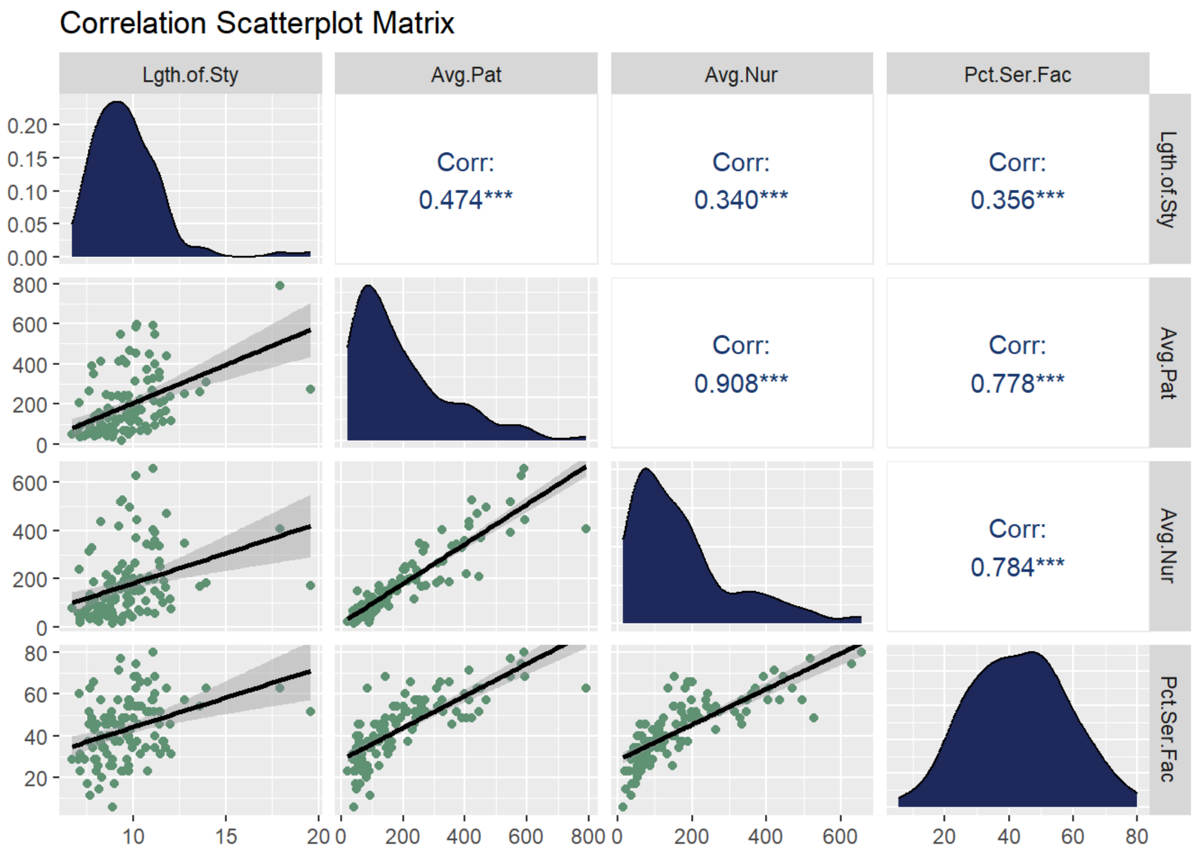


EDA continued

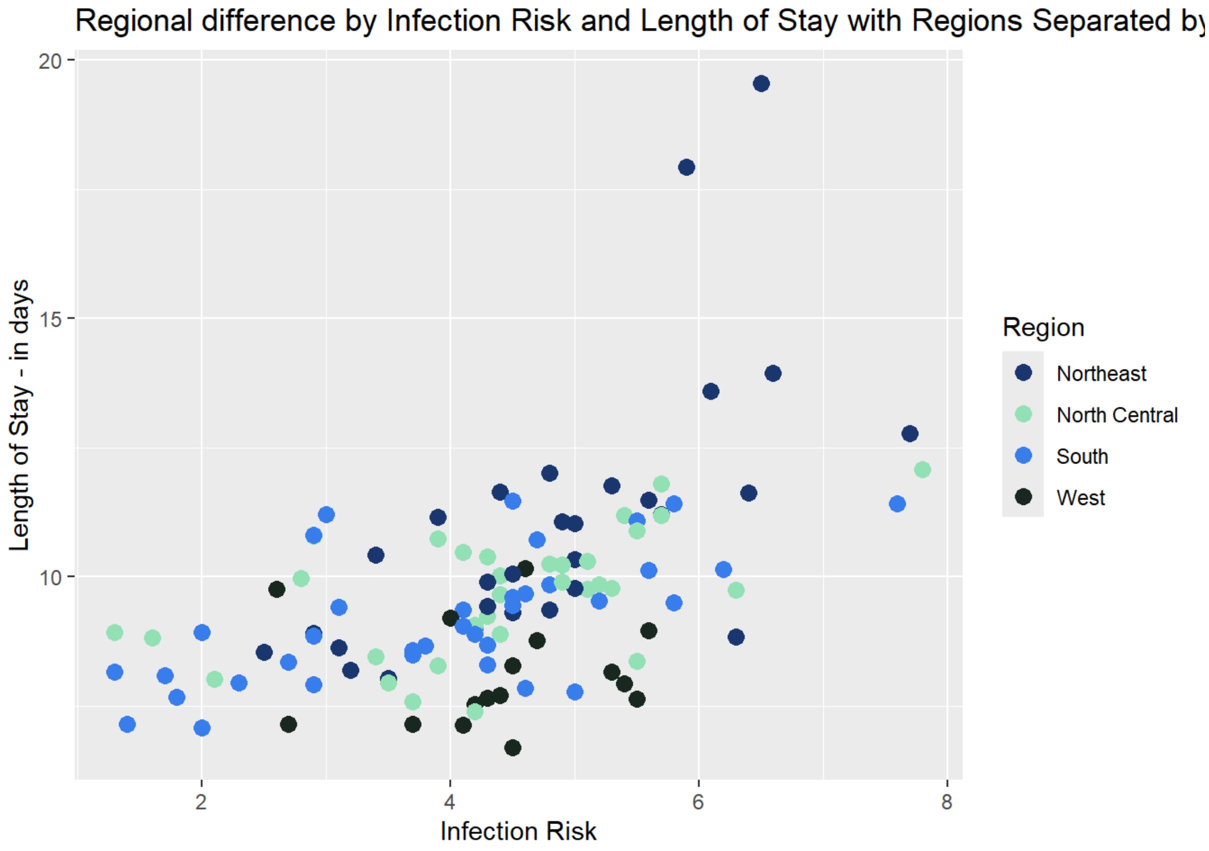
```
#troubleshoot column issues
str(hospitaldataoriginal[, c(2,10,11,12)])
```

```
## 'data.frame':    113 obs. of  4 variables:
##  $ Lgth.of.Sty: num  7.13 8.82 8.34 8.95 11.2 ...
##  $ Avg.Pat    : int  207 51 82 53 134 147 151 399 130 59 ...
##  $ Avg.Nur    : int  241 52 54 148 151 106 129 360 118 66 ...
##  $ Pct.Ser.Fac: num  60 40 20 40 40 40 40 60 40 40 ...
```

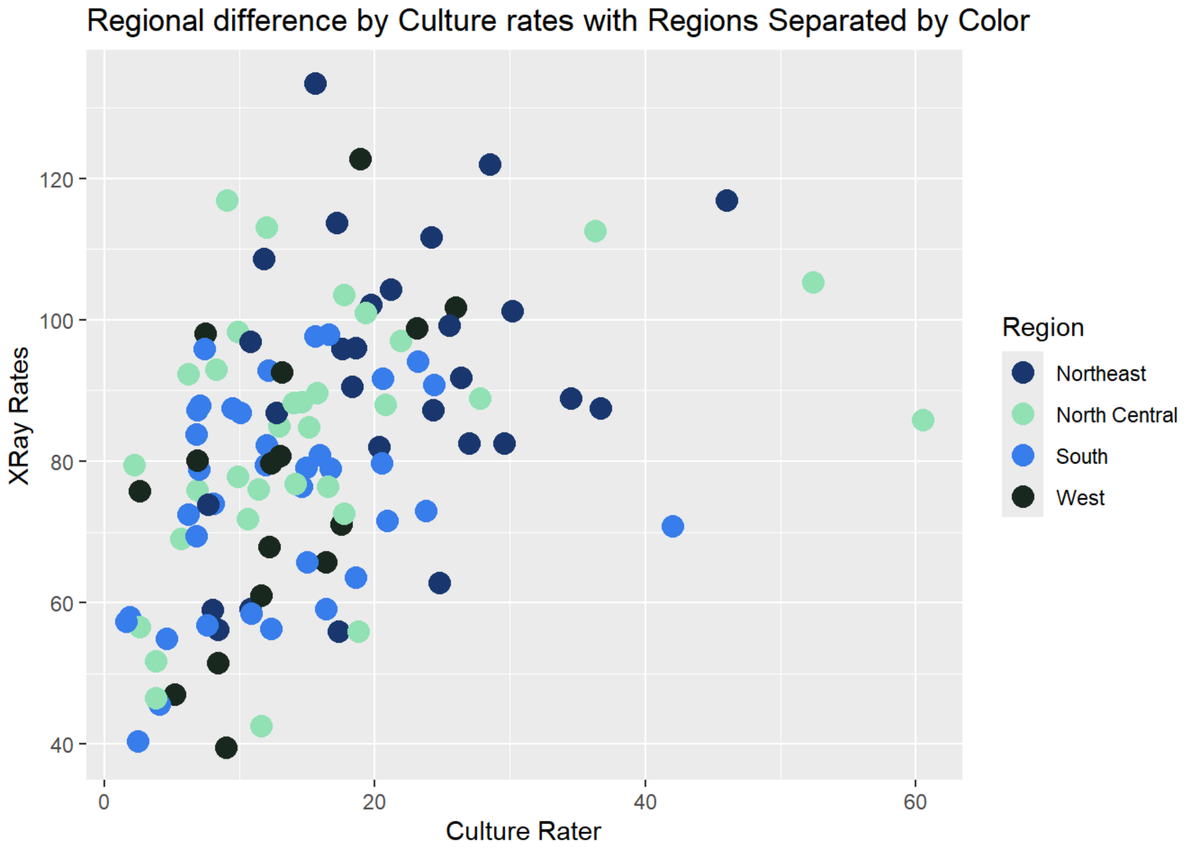
```
# Convert columns to numeric if necessary
hospitaldataoriginal[, c(2,10,11,12)] <- lapply(hospitaldataoriginal[, c(2,10,11,12)], function(x) as.numeric(as.character(x)))

ggpairs(hospitaldataoriginal,
        columns = c(2,10,11,12),
        title = "Correlation Scatterplot Matrix",
        lower = list(continuous = wrap("smooth", color = hosp_color[1])), # Apply custom color to lower panels
        diag = list(continuous = wrap("densityDiag", fill = hosp_color[2])), # Apply custom color to diagonal panels
        upper = list(continuous = wrap("cor", color = hosp_color[3]))) # Apply custom color to upper panels
```



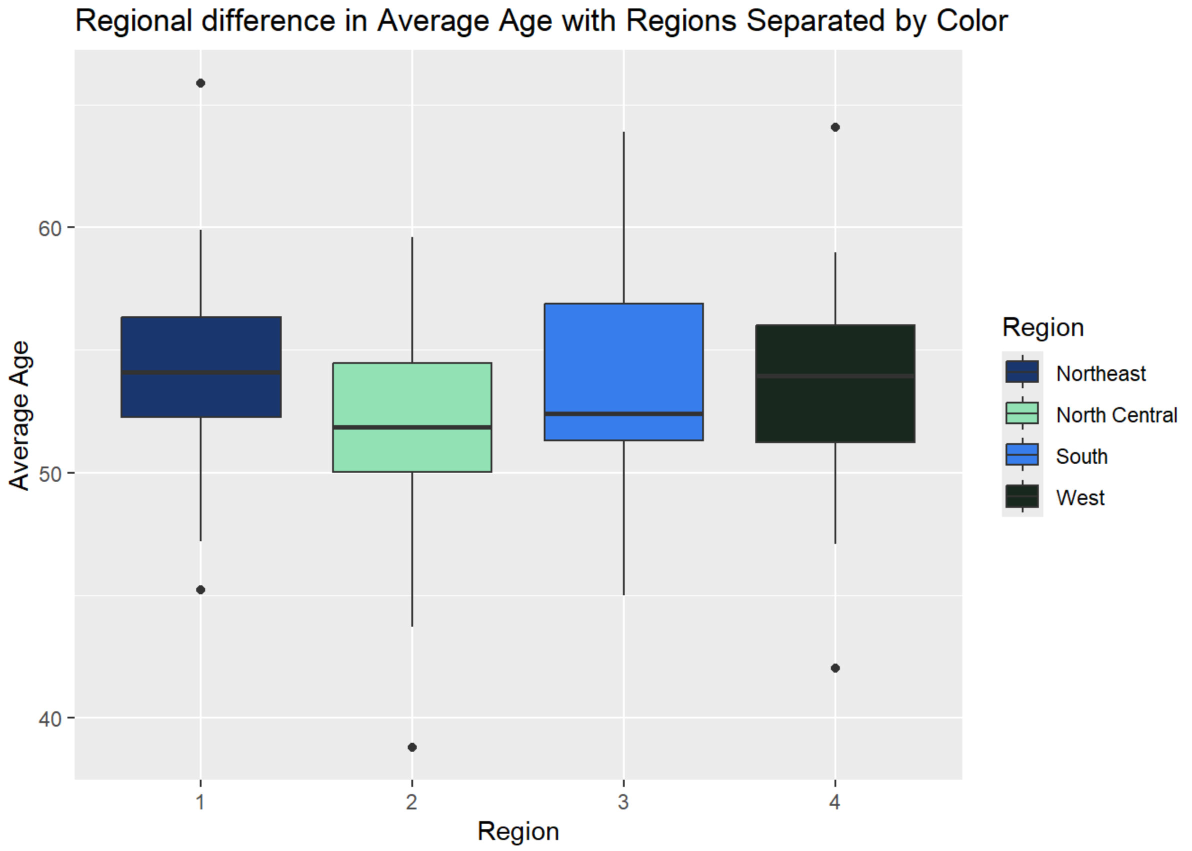
```
#we can see there are different risk profiles based on region
#exploratory chart with interaction terms
library(ggplot2)

# Convert Region to factor
hospitaldataoriginal$Region <- as.factor(hospitaldataoriginal$Region)
hospitaldataoriginal$Med.Sc.Aff <- as.factor(hospitaldataoriginal$Med.Sc.Aff)

ggplot(hospitaldataoriginal, aes(x=Inf.Risk, y=Lgth.of.Sty, color=Region)) +
  geom_point(size=3) +
  labs(title="Regional difference by Infection Risk and Length of Stay with Regions Separated by Color",
       x="Infection Risk", y="Length of Stay - in days") +
  scale_color_manual(values = c("4" = "#1A2820", "3" = "#387DEC", "2" = "#95E2B7", "1" = "#1B3B6F"),
labels = c("4" = "West", "3" = "South", "2" = "North Central", "1" = "Northeast"))
```



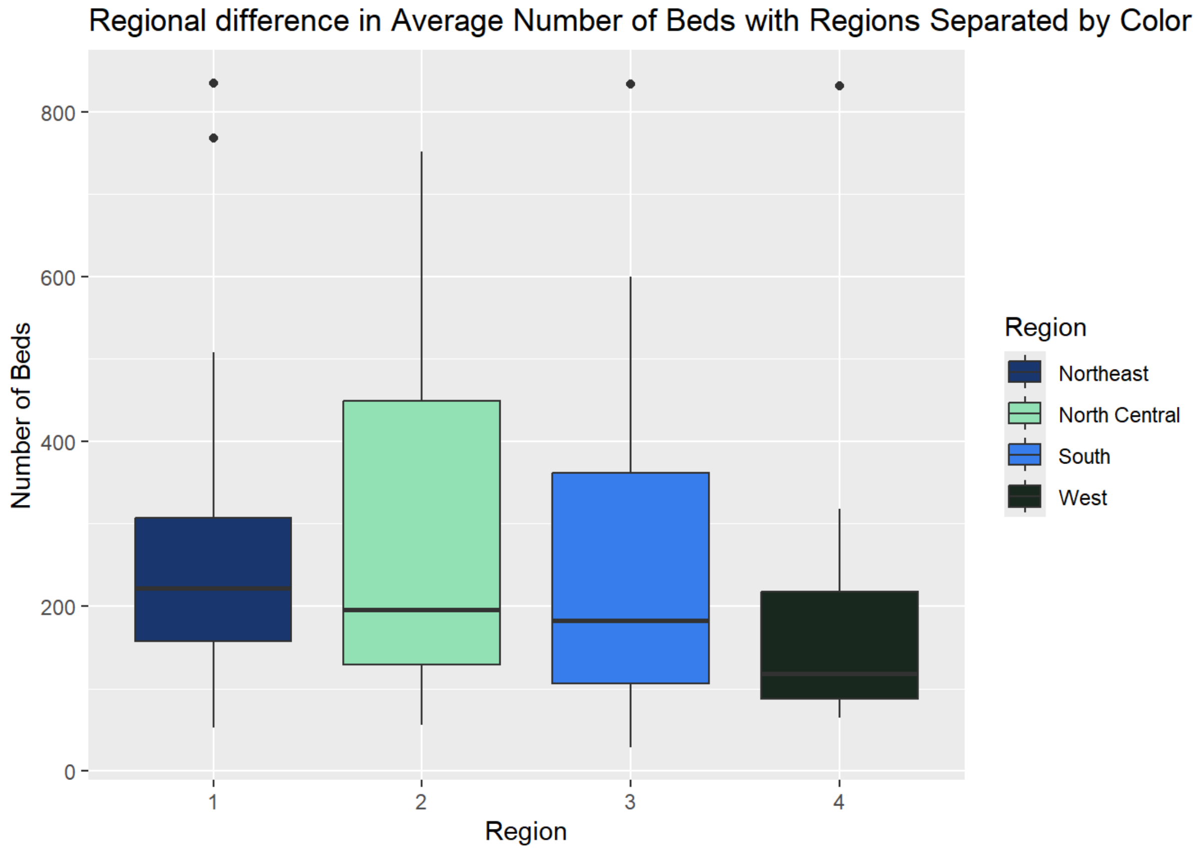
```
#further exploration
ggplot(hospitaldataoriginal, aes(x=R.Cul.Rat, y=R.CX.ray.Rat, color=Region)) +
  geom_point(size=4) +
  labs(title="Regional difference by Culture rates with Regions Separated by Color",
       x="Culture Rater", y="XRay Rates") +
  scale_color_manual(values = c("4" = "#1A2820", "3" = "#387DEC", "2" = "#95E2B7", "1" = "#1B3B6F"),
                     labels = c("4" = "West", "3" = "South", "2" = "North Central", "1" = "Northeast"))
```



```
#risk factors by region
ggplot(hospitaldataoriginal, aes(x=factor(Region), y=Age, fill=Region)) +
  geom_boxplot() +
  labs(title="Regional difference in Average Age with Regions Separated by Color",
       x="Region", y="Average Age") +
  scale_fill_manual(values = c("4" = "#1A2820", "3" = "#387DEC", "2" = "#95E2B7", "1" = "#1B3B6F"),
                    labels = c("4" = "West", "3" = "South", "2" = "North Central", "1" = "Northeast"))
```

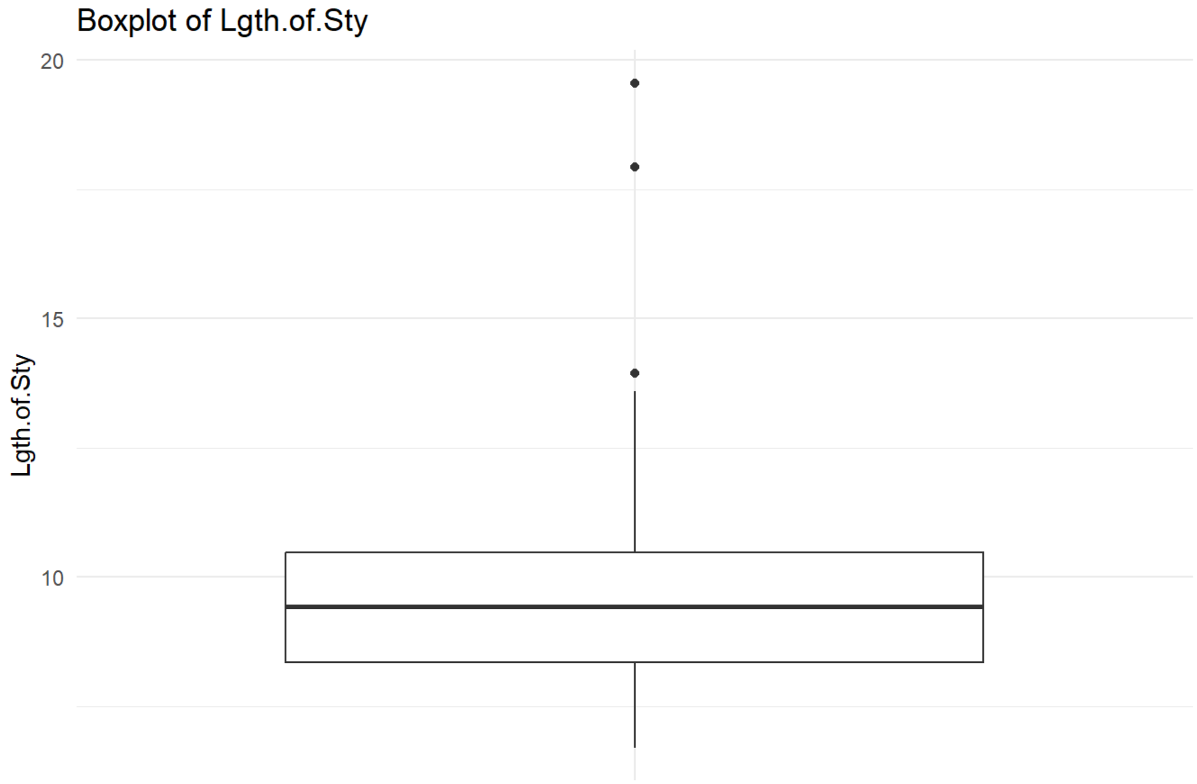


```r
ggplot(hospitaldataoriginal, aes(x=factor(Region), y=N.Beds, fill=Region)) +
  geom_boxplot() +
  labs(title="Regional difference in Average Number of Beds with Regions Separated by Color",
       x="Region", y="Number of Beds") +
  scale_fill_manual(values = c("4" = "#1A2820", "3" = "#387DEC", "2" = "#95E2B7", "1" = "#1B3B6F"),
                    labels = c("4" = "West", "3" = "South", "2" = "North Central", "1" = "Northeast"))
```

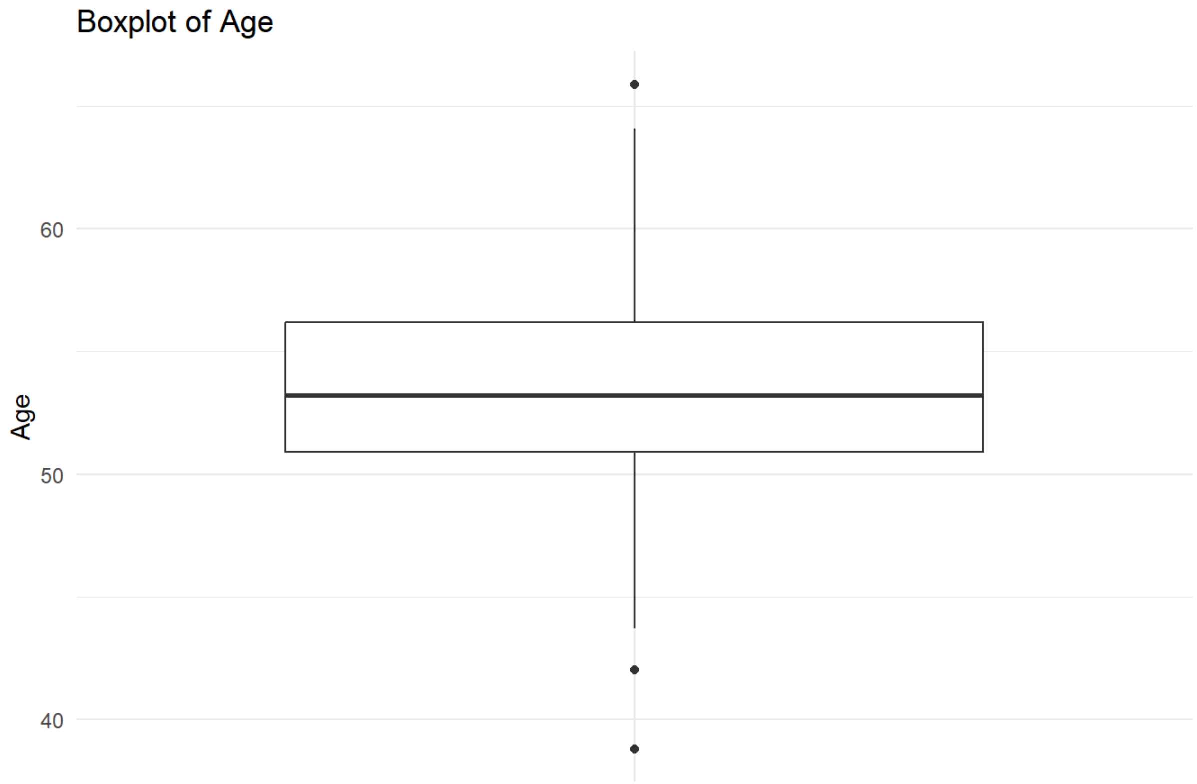


```
#reviewing the data of the correlations we see that typically there is an association of more patients and m
ore beds with medical school
#SW appears to have a longer stay and infection risk
#W has the lowest infection risk

#need help with data wrangling for nures ratio.
#create new variable Nurse/patient ratio to reflect number of nurse to patients to address collinearity and
possible lapses in coverage.

# Create a loop that displays all variables of boxplots except for ID.
for (i in 2:12) {
  # Ensure the column index is valid
  if (i <= ncol(hospitaldataoriginal)) {
    p <- ggplot(hospitaldataoriginal, aes(x = "", y = hospitaldataoriginal[[i]])) +
      geom_boxplot() +
      labs(y = colnames(hospitaldataoriginal)[i], x = "") +
      ggtitle(paste("Boxplot of", colnames(hospitaldataoriginal)[i])) +
      theme_minimal()

    print(p)
  }
}
```

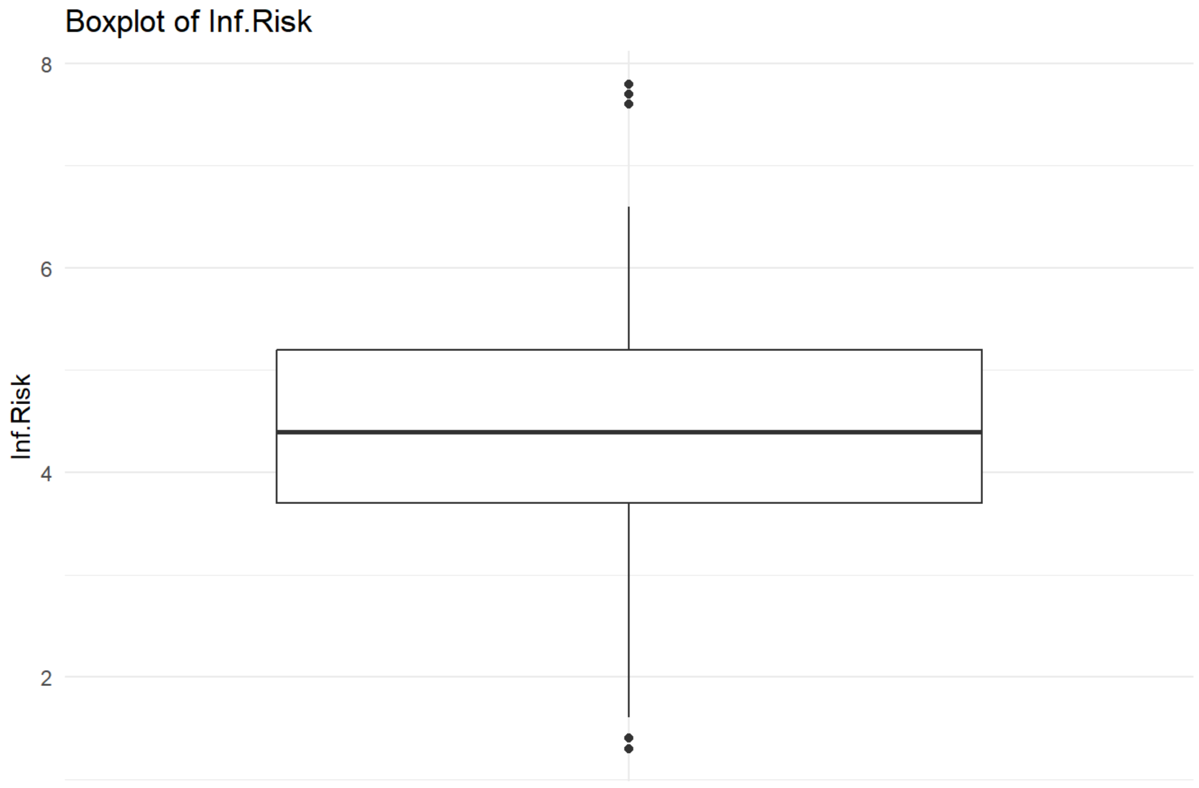
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



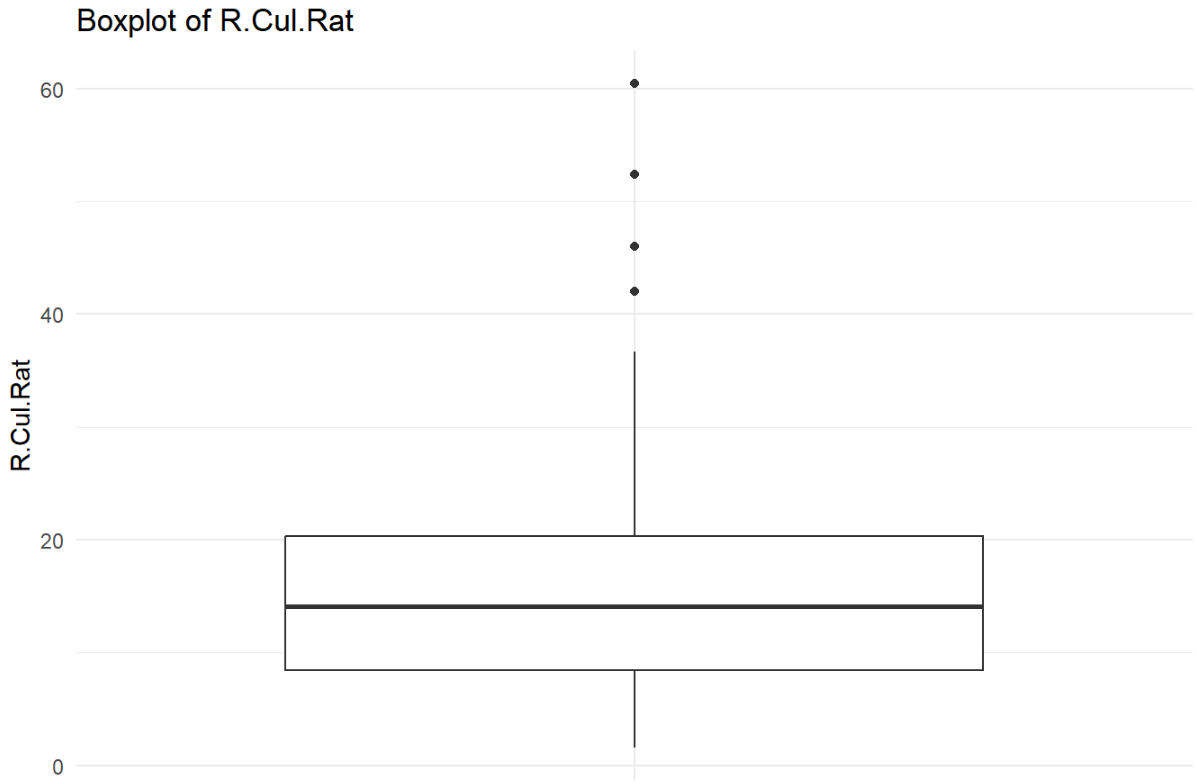
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



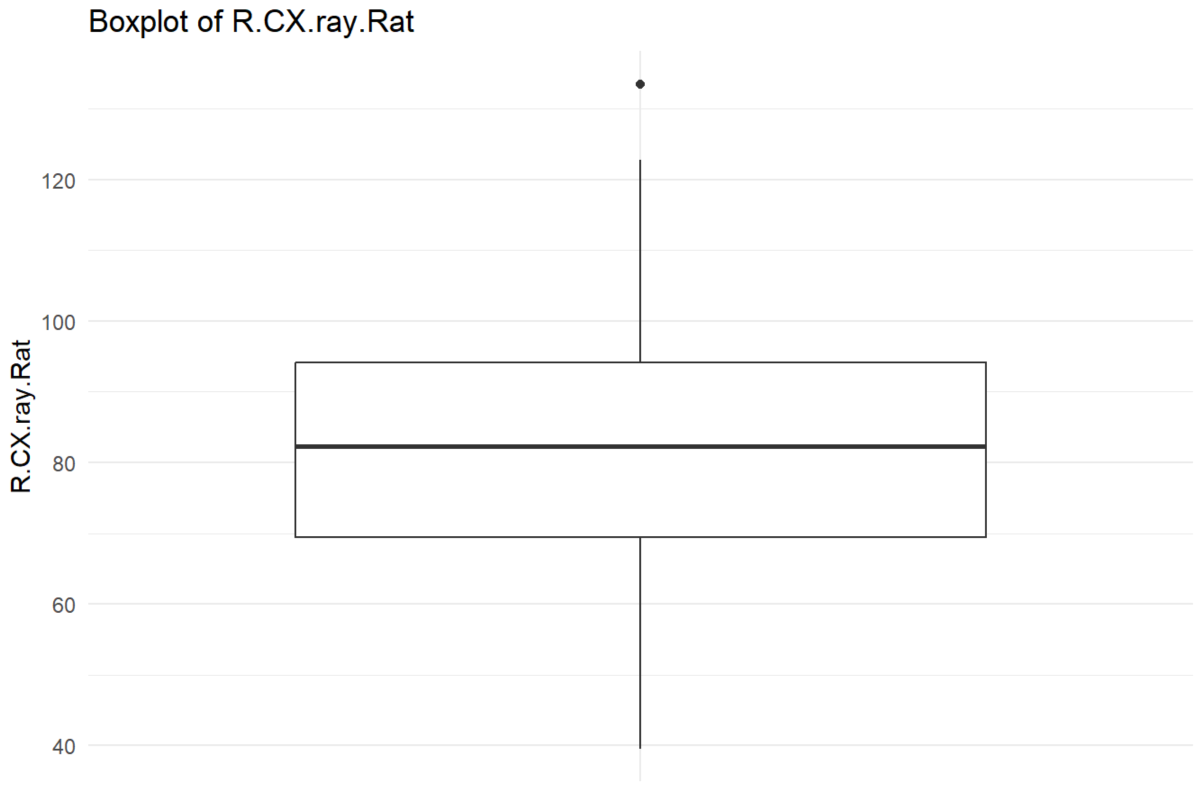
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



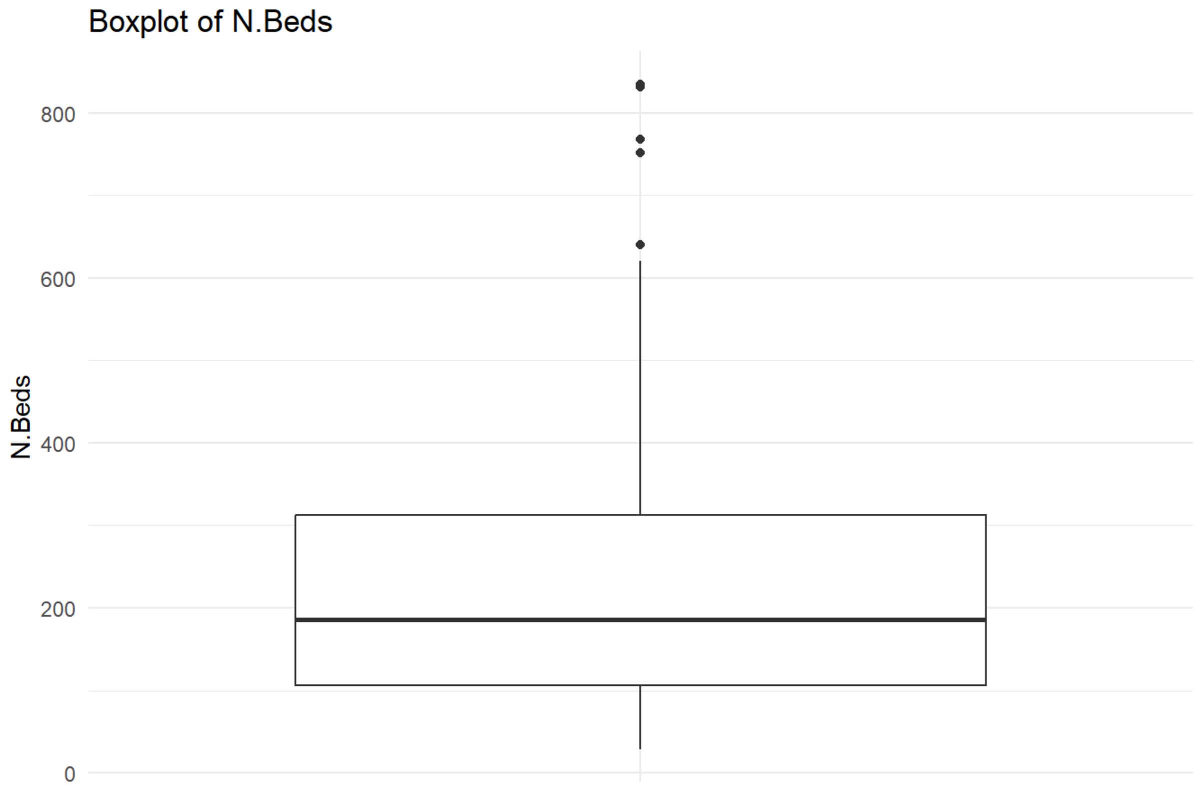
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



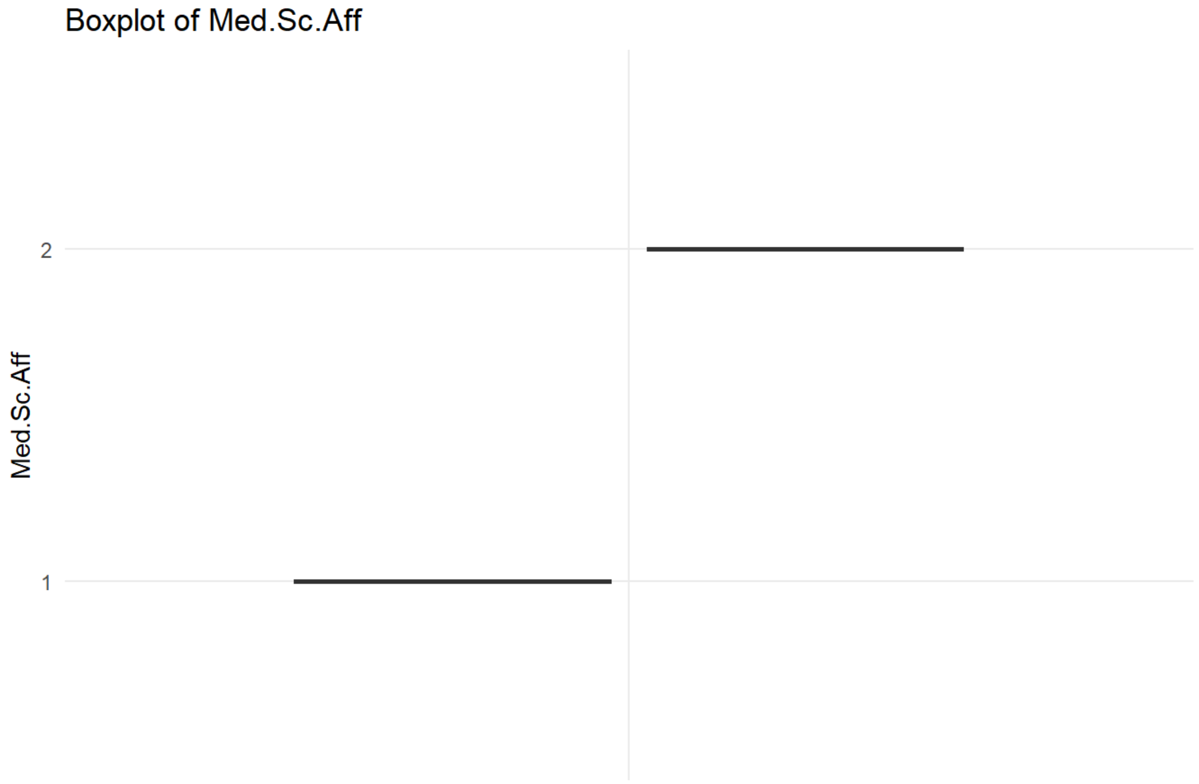
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



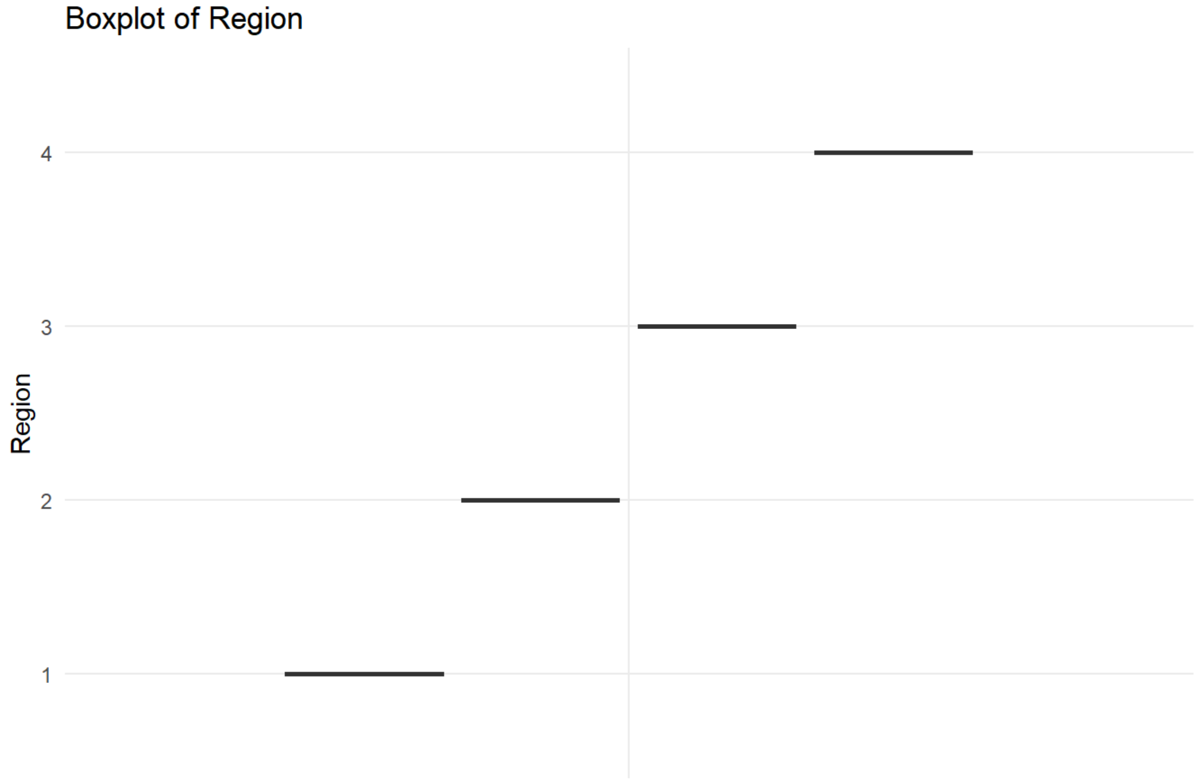
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



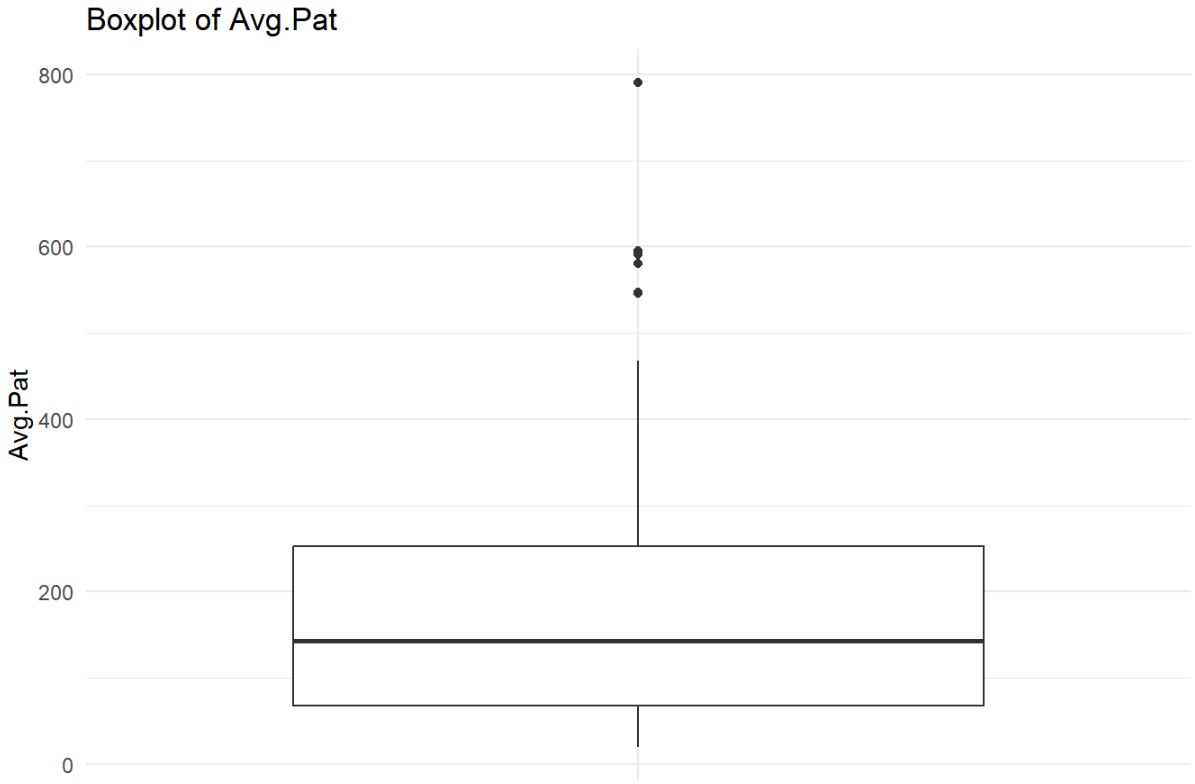
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



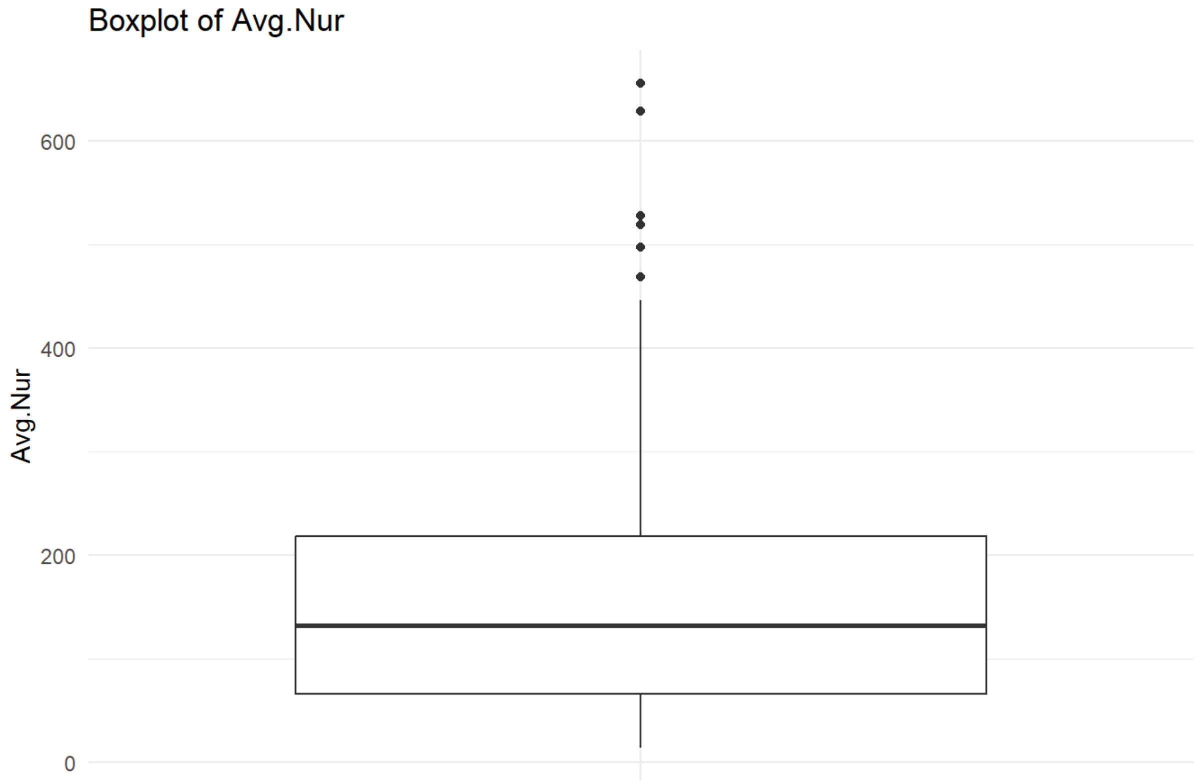
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



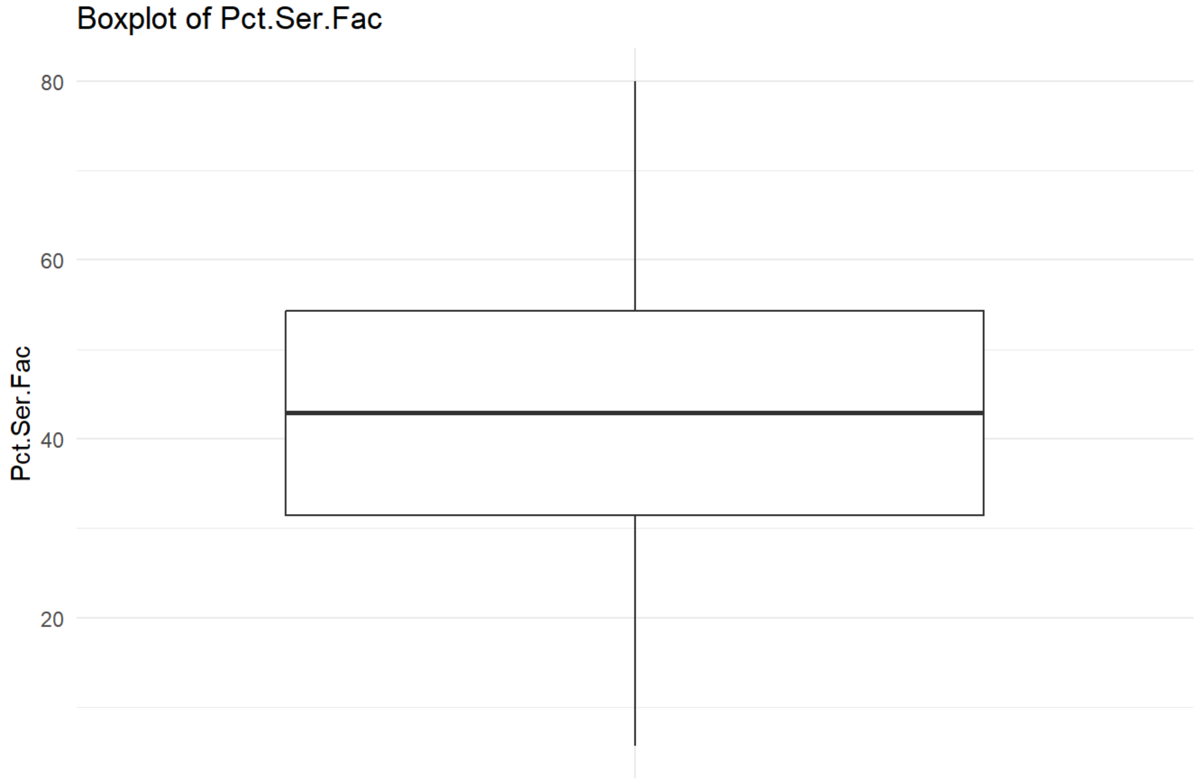
```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```

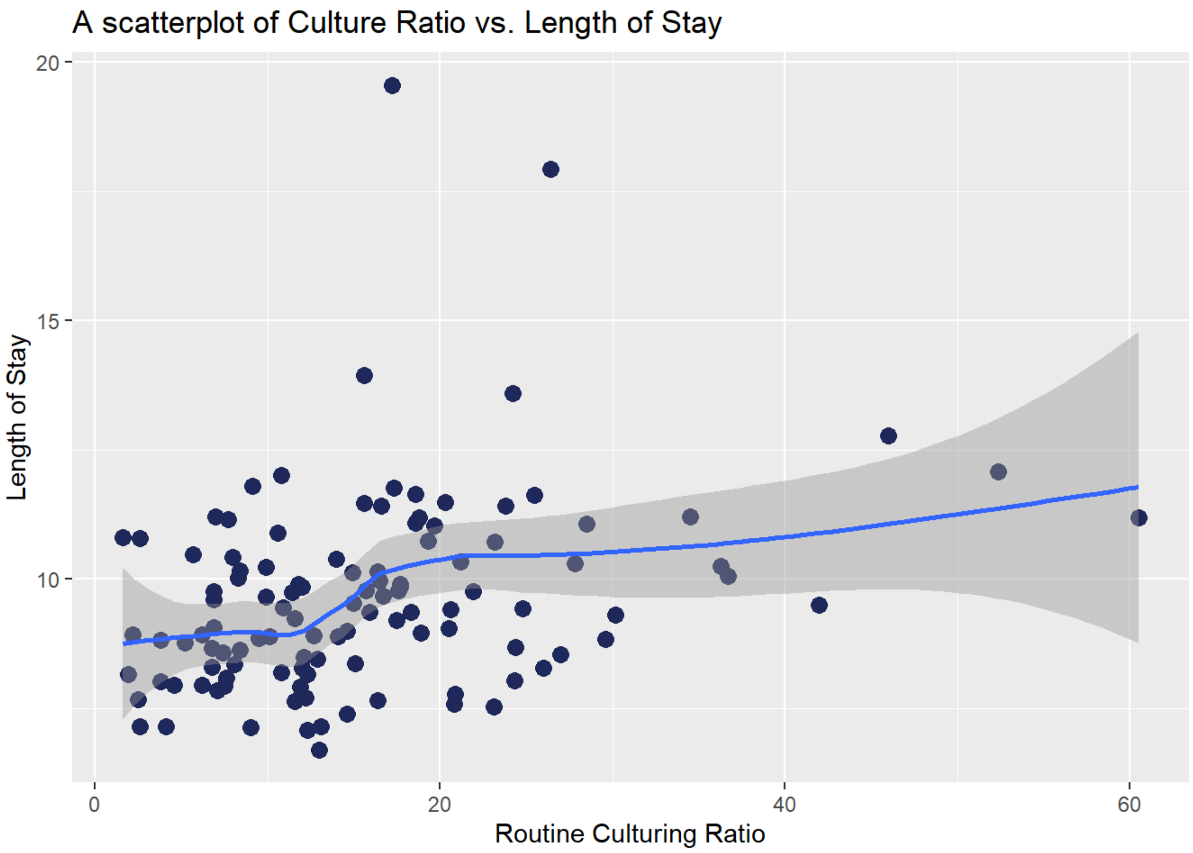


```
## Warning: Use of `hospitaldataoriginal[[i]]` is discouraged.
## ℹ Use `.data[[i]]` instead.
```



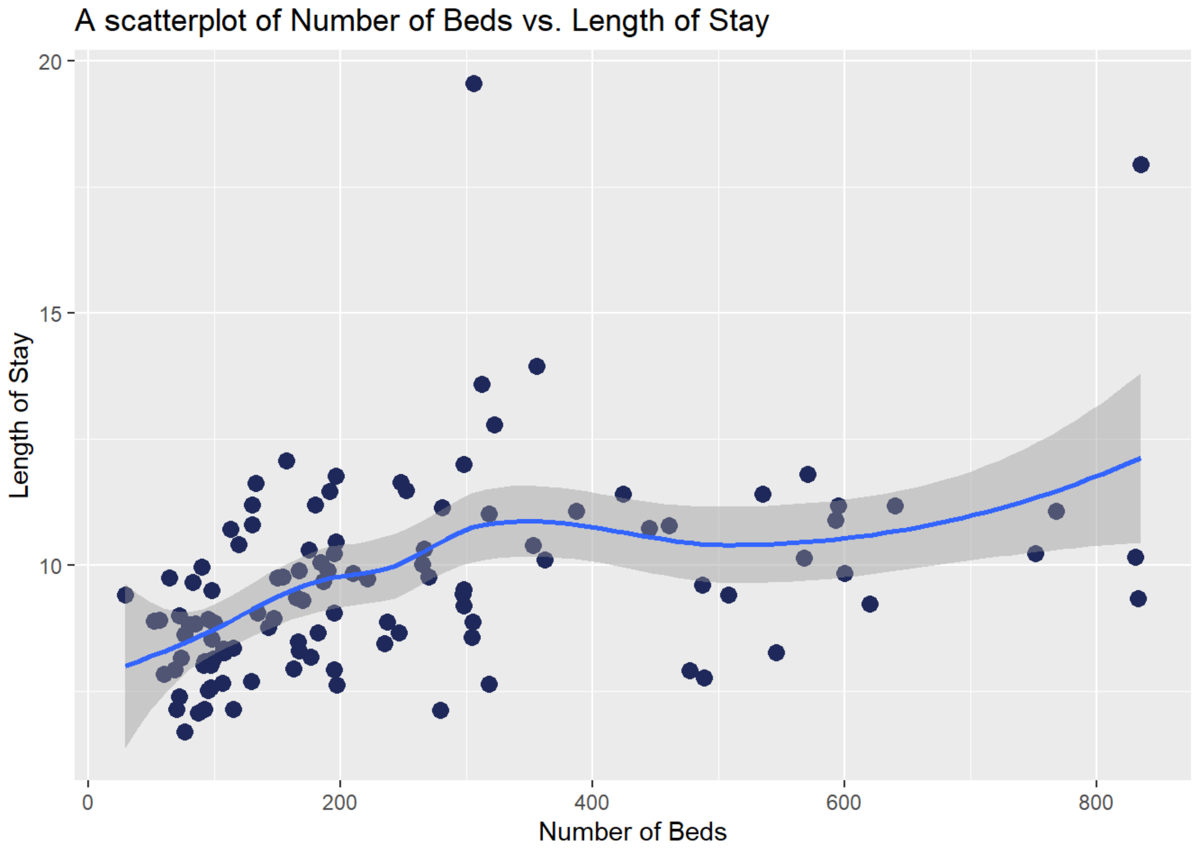
```
#log transforms
ggplot(hospitaldataoriginal, aes(x = R.Cul.Rat, y = Lgth.of.Sty )) +
      geom_point(size=3, color="#21295C") +
      geom_smooth() +
      labs(x = "Routine Culturing Ratio", y = "Length of Stay") +
      ggtitle("A scatterplot of Culture Ratio vs. Length of Stay")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
ggplot(hospitaldataoriginal, aes(x = N.Beds, y = Lgth.of.Sty )) +
      geom_point(size=3, color="#21295C") +
      geom_smooth() +
      labs(x = "Number of Beds", y = "Length of Stay") +
      ggtitle("A scatterplot of Number of Beds vs. Length of Stay")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
#Model Building and Correlation - objective 1
library(caret)
```

```
## Loading required package: lattice
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.4.2
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.4.2
```

```
#model and testing.
hospitaldataoriginal$Region <- factor(hospitaldataoriginal$Region)
hospitaldataoriginal$Med.Sc.Aff <- factor(hospitaldataoriginal$Med.Sc.Aff)
model <- lm(Lgth.of.Sty ~ Inf.Risk + R.Cul.Rat + R.CX.ray.Rat + Age +N.Beds + Med.Sc.Aff + Region + Avg.Pat
+ Avg.Nur + Pct.Ser.Fac, data =hospitaldataoriginal)
summary(model)
```

```
##
## Call:
## lm(formula = Lgth.of.Sty ~ Inf.Risk + R.Cul.Rat + R.CX.ray.Rat +
##     Age + N.Beds + Med.Sc.Aff + Region + Avg.Pat + Avg.Nur +
##     Pct.Ser.Fac, data = hospitaldataoriginal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3048 -0.6608 -0.0272  0.5862  6.3001
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.322292   1.782122   1.864 0.065222 .
## Inf.Risk      0.439665   0.127298   3.454 0.000812 ***
## R.Cul.Rat     0.005546   0.015982   0.347 0.729299
## R.CX.ray.Rat  0.012688   0.007147   1.775 0.078892 .
## Age           0.079922   0.028266   2.827 0.005668 **
## N.Beds       -0.004851   0.003603  -1.346 0.181224
## Med.Sc.Aff2  -0.266644   0.441089  -0.605 0.546872
## Region2      -0.812966   0.351406  -2.313 0.022744 *
## Region3      -1.158277   0.351704  -3.293 0.001370 **
## Region4      -1.880560   0.444136  -4.234  5.1e-05 ***
## Avg.Pat       0.015182   0.004424   3.432 0.000872 ***
## Avg.Nur      -0.005891   0.002218  -2.656 0.009203 **
## Pct.Ser.Fac  -0.012179   0.013774  -0.884 0.378698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.231 on 100 degrees of freedom
## Multiple R-squared:  0.6299, Adjusted R-squared:  0.5855
## F-statistic: 14.18 on 12 and 100 DF,  p-value: < 2.2e-16
```

```
modelall <- lm(Lgth.of.Sty ~.,data = hospitaldataoriginal)
summary(modelall)
```

```
##
## Call:
## lm(formula = Lgth.of.Sty ~ ., data = hospitaldataoriginal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2041 -0.6967 -0.0619  0.5284  6.3268
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.020361   1.827912   1.652 0.101630
## ID            0.002960   0.003828   0.773 0.441235
## Age           0.080410   0.028330   2.838 0.005504 **
## Inf.Risk      0.444081   0.127682   3.478 0.000752 ***
## R.Cul.Rat     0.007403   0.016193   0.457 0.648529
## R.CX.ray.Rat  0.012896   0.007166   1.799 0.074987 .
## N.Beds       -0.004929   0.003612  -1.365 0.175445
## Med.Sc.Aff2  -0.239421   0.443379  -0.540 0.590416
## Region2      -0.802425   0.352378  -2.277 0.024927 *
## Region3      -1.157860   0.352413  -3.286 0.001409 **
## Region4      -1.887197   0.445114  -4.240 5.03e-05 ***
## Avg.Pat       0.014953   0.004442   3.366 0.001086 **
## Avg.Nur      -0.005501   0.002279  -2.414 0.017618 *
## Pct.Ser.Fac  -0.011895   0.013806  -0.862 0.391015
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.233 on 99 degrees of freedom
## Multiple R-squared:  0.6321, Adjusted R-squared:  0.5838
## F-statistic: 13.09 on 13 and 99 DF,  p-value: 2.799e-16
```

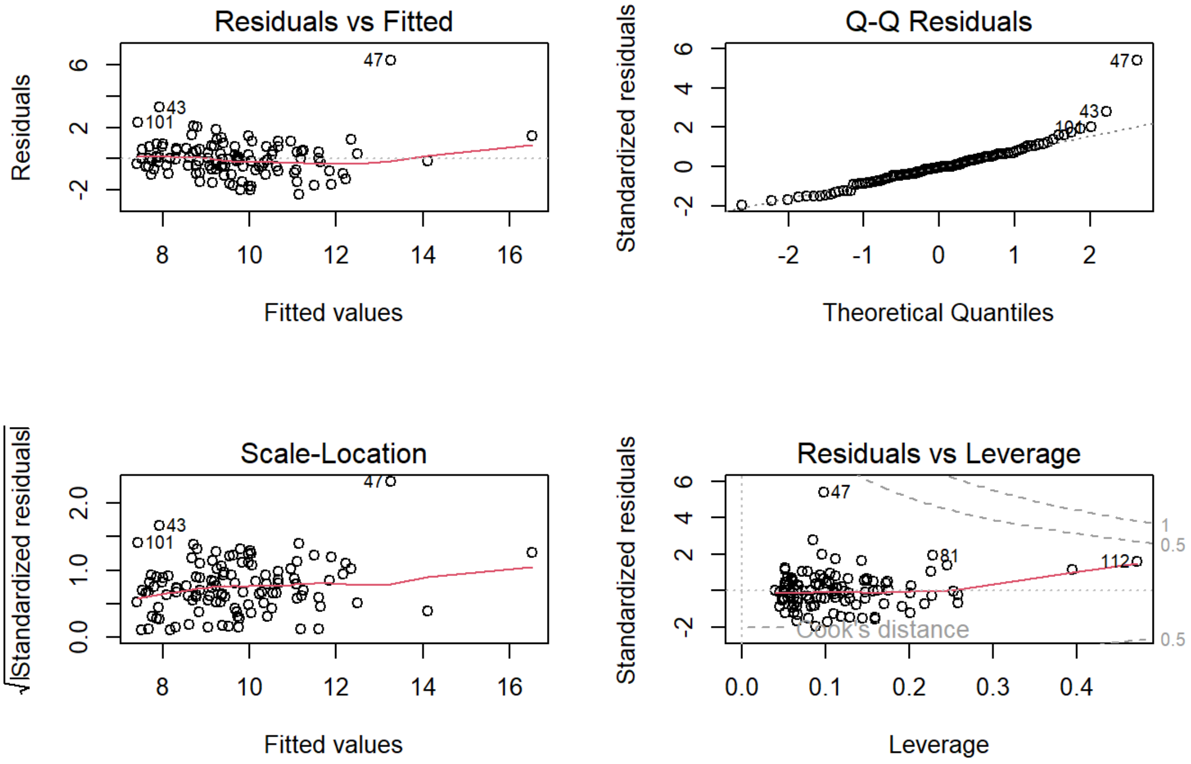
```
vif(modelall)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## ID            1.158549  1        1.076359
## Age           1.176754  1        1.084783
## Inf.Risk      2.159012  1        1.469358
## R.Cul.Rat     2.023038  1        1.422335
## R.CX.ray.Rat  1.418264  1        1.190909
## N.Beds       35.726926  1        5.977201
## Med.Sc.Aff    1.867108  1        1.366421
## Region        1.721747  3        1.094783
## Avg.Pat      34.363637  1        5.862051
## Avg.Nur       7.418642  1        2.723718
## Pct.Ser.Fac   3.244109  1        1.801141
```

```
modellog <- lm(Lgth.of.Sty ~ Inf.Risk + R.Cul.Rat + R.CX.ray.Rat + Age +N.Beds + Med.Sc.Aff + Region + Avg.P
at + Avg.Nur + Pct.Ser.Fac, data =hospitaldataoriginal)
summary(modellog)
```

```
##
## Call:
## lm(formula = Lgth.of.Sty ~ Inf.Risk + R.Cul.Rat + R.CX.ray.Rat +
##     Age + N.Beds + Med.Sc.Aff + Region + Avg.Pat + Avg.Nur +
##     Pct.Ser.Fac, data = hospitaldataoriginal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3048 -0.6608 -0.0272  0.5862  6.3001
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.322292   1.782122   1.864 0.065222 .
## Inf.Risk      0.439665   0.127298   3.454 0.000812 ***
## R.Cul.Rat     0.005546   0.015982   0.347 0.729299
## R.CX.ray.Rat  0.012688   0.007147   1.775 0.078892 .
## Age           0.079922   0.028266   2.827 0.005668 **
## N.Beds       -0.004851   0.003603  -1.346 0.181224
## Med.Sc.Aff2  -0.266644   0.441089  -0.605 0.546872
## Region2      -0.812966   0.351406  -2.313 0.022744 *
## Region3      -1.158277   0.351704  -3.293 0.001370 **
## Region4      -1.880560   0.444136  -4.234  5.1e-05 ***
## Avg.Pat       0.015182   0.004424   3.432 0.000872 ***
## Avg.Nur      -0.005891   0.002218  -2.656 0.009203 **
## Pct.Ser.Fac  -0.012179   0.013774  -0.884 0.378698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.231 on 100 degrees of freedom
## Multiple R-squared:  0.6299, Adjusted R-squared:  0.5855
## F-statistic: 14.18 on 12 and 100 DF,  p-value: < 2.2e-16
```

```
#residual plot
par(mfrow=c(2,2))
plot(model) #how to add color on this plot format?
```



```
par(mfrow=c(1,1))

#All not transform
# Set up the K-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation
# Train the model using K-fold cross-validation
modellog <- train(Lgth.of.Sty ~ ., data = hospitaldataoriginal, method = "lm", trControl = train_control)
# Print the results
print(modellog)
```

```
## Linear Regression
##
## 113 samples
##  11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 101, 102, 102, 102, 101, 101, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.273548  0.5527278  0.9645423
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
#model building objective 2
# full model
#lasso for significance check
# Define the fit control
fitControl <- trainControl(
  method = "cv", # cross-validation
  number = 10    # number of folds
)
complex.glmnet.fit<-train(log(Lgth.of.Sty) ~
                            Age +
                            Inf.Risk  +
                            log(R.Cul.Rat) +
                            R.CX.ray.Rat +
                            log(N.Beds) +
                            Pct.Ser.Fac +
                            Region +
                            Med.Sc.Aff ,
                          data = hospitaldataoriginal,
                  method="glmnet",
                  trControl=fitControl,
                  tuneGrid=expand.grid(data.frame(alpha=1,lambda=seq(0,0.05,0.001)))
)
opt.pen<-complex.glmnet.fit$finalModel$lambdaOpt
coef(complex.glmnet.fit$finalModel,opt.pen)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                           s1
## (Intercept)     1.392381699
## Age             0.007493821
## Inf.Risk        0.049804895
## log(R.Cul.Rat) -0.025491800
## R.CX.ray.Rat    0.001297288
## log(N.Beds)     0.089803786
## Pct.Ser.Fac    -0.002569695
## Region2        -0.085900038
## Region3        -0.117215037
## Region4        -0.239122417
## Med.Sc.Aff2    -0.075859848
```

```
complex.glmnet.fit
```

```
## glmnet
##
## 113 samples
##   8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 101, 101, 101, 101, 101, 101, ...
## Resampling results across tuning parameters:
##
##   lambda  RMSE       Rsquared   MAE
##   0.000   0.1195688  0.5994991  0.09332822
##   0.001   0.1197579  0.5988740  0.09338920
##   0.002   0.1202951  0.5957051  0.09376343
##   0.003   0.1210823  0.5898832  0.09420030
##   0.004   0.1217428  0.5828032  0.09426100
##   0.005   0.1222928  0.5769104  0.09405825
##   0.006   0.1225708  0.5731090  0.09383439
##   0.007   0.1229314  0.5698274  0.09392464
##   0.008   0.1235777  0.5651592  0.09425346
##   0.009   0.1243199  0.5597846  0.09461095
##   0.010   0.1250209  0.5543290  0.09495526
##   0.011   0.1256027  0.5496166  0.09530283
##   0.012   0.1260402  0.5462259  0.09561996
##   0.013   0.1264226  0.5436467  0.09595650
##   0.014   0.1268502  0.5406878  0.09628846
##   0.015   0.1273667  0.5366964  0.09670084
##   0.016   0.1278906  0.5329181  0.09719985
##   0.017   0.1284367  0.5291485  0.09776838
##   0.018   0.1289716  0.5259372  0.09833180
##   0.019   0.1295294  0.5226472  0.09889134
##   0.020   0.1301025  0.5194257  0.09944884
##   0.021   0.1306858  0.5160198  0.10006726
##   0.022   0.1312790  0.5124468  0.10067965
##   0.023   0.1318818  0.5086447  0.10126066
##   0.024   0.1324833  0.5048272  0.10188013
##   0.025   0.1331069  0.5007130  0.10252515
##   0.026   0.1337340  0.4966537  0.10317434
##   0.027   0.1343684  0.4926319  0.10380243
##   0.028   0.1350357  0.4883066  0.10441924
##   0.029   0.1356776  0.4842188  0.10497644
##   0.030   0.1363107  0.4805928  0.10551178
##   0.031   0.1369446  0.4771321  0.10602007
##   0.032   0.1375426  0.4741076  0.10647020
##   0.033   0.1381326  0.4713518  0.10695965
##   0.034   0.1386925  0.4691173  0.10744205
##   0.035   0.1392270  0.4671432  0.10793461
##   0.036   0.1397358  0.4656634  0.10838532
##   0.037   0.1402588  0.4640237  0.10883183
##   0.038   0.1407913  0.4621118  0.10927307
##   0.039   0.1413285  0.4603329  0.10973385
##   0.040   0.1418820  0.4584319  0.11019549
##   0.041   0.1424392  0.4564401  0.11064967
##   0.042   0.1430055  0.4543336  0.11110010
##   0.043   0.1435842  0.4521124  0.11155030
##   0.044   0.1441765  0.4497468  0.11203094
##   0.045   0.1447780  0.4473724  0.11250873
##   0.046   0.1453911  0.4449037  0.11298289
##   0.047   0.1460148  0.4423626  0.11345583
##   0.048   0.1466518  0.4396479  0.11392869
##   0.049   0.1473010  0.4367892  0.11439804
##   0.050   0.1479617  0.4337934  0.11486361
## 
## Tuning parameter 'alpha' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 0.
```

```
#model with stats
complexMLR <- lm(log(Lgth.of.Sty) ~
                   Age +
                   Inf.Risk  +
                   log(R.Cul.Rat) +
                   R.CX.ray.Rat +
                   log(N.Beds) +
                   Pct.Ser.Fac +
                   Region +
                   Med.Sc.Aff ,
                 data = hospitaldataoriginal)

summary(complexMLR)
```

```
## 
## Call:
## lm(formula = log(Lgth.of.Sty) ~ Age + Inf.Risk + log(R.Cul.Rat) + 
##     R.CX.ray.Rat + log(N.Beds) + Pct.Ser.Fac + Region + Med.Sc.Aff, 
##     data = hospitaldataoriginal)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.26135 -0.07948 -0.00408  0.05218  0.42573 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     1.3893418  0.2277768   6.100 1.93e-08 ***
## Age             0.0075147  0.0027352   2.747 0.007103 ** 
## Inf.Risk        0.0500108  0.0124553   4.015 0.000114 ***
## log(R.Cul.Rat) -0.0263510  0.0236476  -1.114 0.267761    
## R.CX.ray.Rat    0.0013049  0.0006845   1.907 0.059397 .  
## log(N.Beds)     0.0914489  0.0311439   2.936 0.004105 ** 
## Pct.Ser.Fac    -0.0026640  0.0014236  -1.871 0.064169 .  
## Region2        -0.0875336  0.0320911  -2.728 0.007512 ** 
## Region3        -0.1190709  0.0321072  -3.709 0.000339 ***
## Region4        -0.2407594  0.0388238  -6.201 1.21e-08 ***
## Med.Sc.Aff2    -0.0766780  0.0368444  -2.081 0.039926 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1154 on 102 degrees of freedom
## Multiple R-squared:  0.6174, Adjusted R-squared:  0.5798 
## F-statistic: 16.46 on 10 and 102 DF,  p-value: < 2.2e-16
```

```
vif(complexMLR)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## Age            1.253306  1        1.119511
## Inf.Risk       2.347541  1        1.532169
## log(R.Cul.Rat) 2.275025  1        1.508319
## R.CX.ray.Rat   1.478355  1        1.215876
## log(N.Beds)    4.371156  1        2.090731
## Pct.Ser.Fac    3.941102  1        1.985221
## Region         1.507269  3        1.070776
## Med.Sc.Aff     1.473240  1        1.213771
```

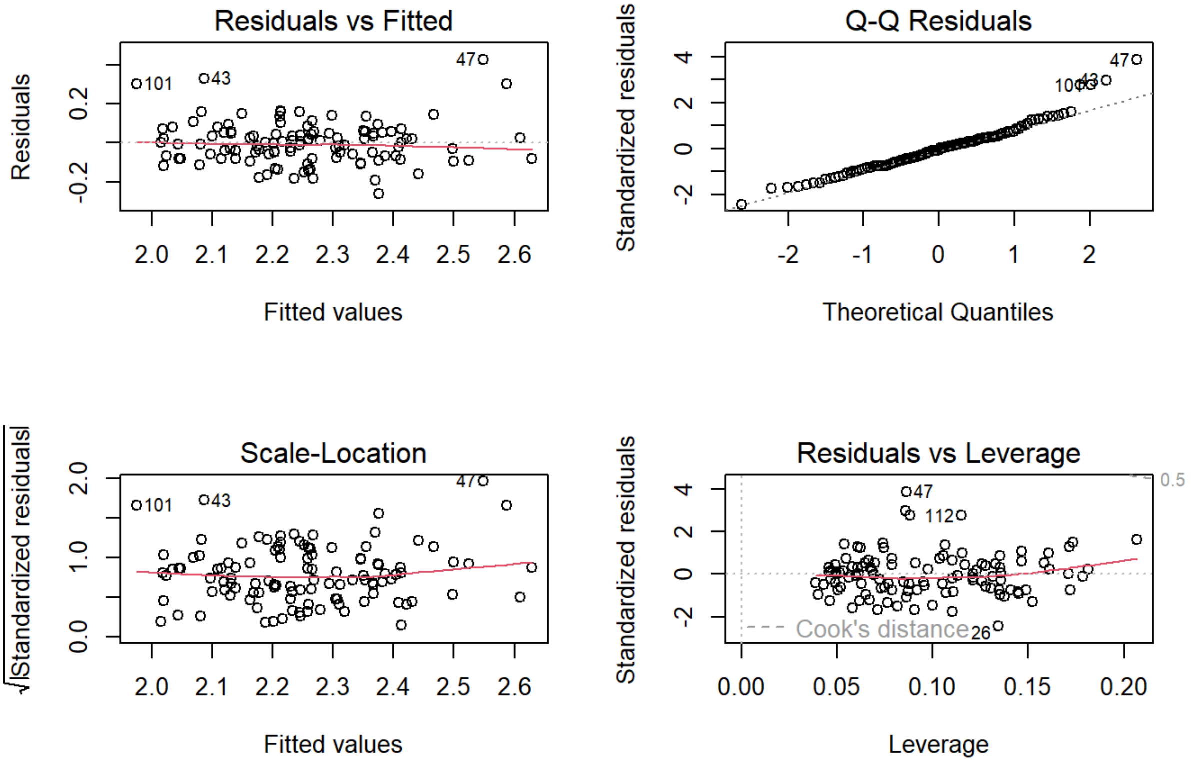
```
AIC(complexMLR)
```

```
## [1] -154.9845
```

```
BIC(complexMLR)
```

```
## [1] -122.2558
```

```
par(mfrow=c(2,2))
plot(complexMLR)
```



```
#All not transform
# Set up the K-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation
# Train the model using K-fold cross-validation
model <- train(Lgth.of.Sty ~ ., data = hospitaldataoriginal, method = "lm", trControl = train_control)
# Print the results
print(model)
```

```
## Linear Regression
##
## 113 samples
##  11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 101, 101, 102, 102, 102, 102, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.241145  0.5266838  0.9521896
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
#complex
# Set up the K-fold cross-validation
train_control <- trainControl(method = "cv", number = 5)  # 5-fold cross-validation due to data size
# Train the model using K-fold cross-validation
complexMLR <- train(log(Lgth.of.Sty) ~ Age +Inf.Risk  + log(R.Cul.Rat) + R.CX.ray.Rat + log(N.Beds) + Pct.Ser.Fac + Region + Med.Sc.Aff,
                 data = hospitaldataoriginal, method = "lm", trControl = train_control)
# Print the results
print(complexMLR)
```

```
## Linear Regression
##
## 113 samples
##   8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 91, 90, 90, 91, 90
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1180721  0.5627052  0.08904771
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
#KNN
hospitaldataoriginal = subset(hospitaldataoriginal, select = -c(ID))

set.seed(123)
splitPerc = .7
trainIndices = sample(1:dim(hospitaldataoriginal)[1],round(splitPerc * dim(hospitaldataoriginal)[1]))
train = hospitaldataoriginal[trainIndices,]
test = hospitaldataoriginal[-trainIndices,]
#fitControl<-trainControl(method="repeatedcv",number=10,repeats=1)
knn.fit<-train(Lgth.of.Sty~.,
                  data=hospitaldataoriginal,
                  method="knn",
                  trControl=fitControl,
                  tuneGrid=expand.grid(k=c(1:30)))
knn.fit
```

```
## k-Nearest Neighbors
##
## 113 samples
##  10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 101, 102, 102, 101, 102, 103, ...
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##    1  1.931874  0.2035353  1.461433
##    2  1.835081  0.1537217  1.367330
##    3  1.712071  0.1744278  1.249423
##    4  1.709318  0.1713712  1.260177
##    5  1.670932  0.2386659  1.248104
##    6  1.623353  0.2768401  1.215800
##    7  1.625698  0.2547864  1.210476
##    8  1.627995  0.2610006  1.208975
##    9  1.609004  0.2708242  1.204524
##   10  1.595395  0.2764421  1.193722
##   11  1.610189  0.2697781  1.201789
##   12  1.624102  0.2639921  1.216427
##   13  1.621095  0.2646163  1.209822
##   14  1.613763  0.2667539  1.199333
##   15  1.611689  0.2638428  1.193311
##   16  1.609333  0.2715710  1.189467
##   17  1.606180  0.2744751  1.188110
##   18  1.597008  0.2761880  1.169145
##   19  1.611736  0.2624505  1.172820
##   20  1.610582  0.2669988  1.168927
##   21  1.605600  0.2736155  1.166018
##   22  1.608602  0.2642584  1.165232
##   23  1.607435  0.2645986  1.164905
##   24  1.608091  0.2644755  1.161224
##   25  1.600322  0.2718914  1.157588
##   26  1.597892  0.2757529  1.156870
##   27  1.584482  0.2884484  1.152622
##   28  1.588025  0.2933965  1.163574
##   29  1.588752  0.2895604  1.164008
##   30  1.593788  0.2872139  1.160700
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 27.
```

```
knn.fit$results
```

```
##     k     RMSE  Rsquared      MAE    RMSESD RsquaredSD     MAESD
## 1   1 1.931874 0.2035353 1.461433 0.6979680  0.1777741 0.2770183
## 2   2 1.835081 0.1537217 1.367330 0.8088568  0.1971560 0.4034611
## 3   3 1.712071 0.1744278 1.249423 0.7567335  0.1899604 0.3734956
## 4   4 1.709318 0.1713712 1.260177 0.7351722  0.1788759 0.3500292
## 5   5 1.670932 0.2386659 1.248104 0.6569790  0.1751006 0.3137085
## 6   6 1.623353 0.2768401 1.215800 0.6602331  0.1856511 0.3203075
## 7   7 1.625698 0.2547864 1.210476 0.6505152  0.1495626 0.3290280
## 8   8 1.627995 0.2610006 1.208975 0.6320418  0.1574146 0.3122166
## 9   9 1.609004 0.2708242 1.204524 0.6447646  0.1634360 0.3187603
## 10 10 1.595395 0.2764421 1.193722 0.6197940  0.1694124 0.3085731
## 11 11 1.610189 0.2697781 1.201789 0.6262054  0.1609877 0.3253841
## 12 12 1.624102 0.2639921 1.216427 0.6422669  0.1455946 0.3271783
## 13 13 1.621095 0.2646163 1.209822 0.6445161  0.1422989 0.3245422
## 14 14 1.613763 0.2667539 1.199333 0.6592942  0.1394791 0.3318230
## 15 15 1.611689 0.2638428 1.193311 0.6742624  0.1340311 0.3394510
## 16 16 1.609333 0.2715710 1.189467 0.6761658  0.1409613 0.3434679
## 17 17 1.606180 0.2744751 1.188110 0.6818061  0.1616404 0.3451567
## 18 18 1.597008 0.2761880 1.169145 0.6982730  0.1679167 0.3556517
## 19 19 1.611736 0.2624505 1.172820 0.6968963  0.1833894 0.3459264
## 20 20 1.610582 0.2669988 1.168927 0.6964919  0.1849379 0.3561728
## 21 21 1.605600 0.2736155 1.166018 0.7026222  0.1796564 0.3530010
## 22 22 1.608602 0.2642584 1.165232 0.6902601  0.1744242 0.3538911
## 23 23 1.607435 0.2645986 1.164905 0.6843779  0.1676605 0.3544878
## 24 24 1.608091 0.2644755 1.161224 0.6819871  0.1668138 0.3520622
## 25 25 1.600322 0.2718914 1.157588 0.6756445  0.1699803 0.3524991
## 26 26 1.597892 0.2757529 1.156870 0.6758481  0.1625604 0.3568347
## 27 27 1.584482 0.2884484 1.152622 0.6638921  0.1674895 0.3518263
## 28 28 1.588025 0.2933965 1.163574 0.6563762  0.1602739 0.3455494
## 29 29 1.588752 0.2895604 1.164008 0.6536832  0.1695549 0.3466031
## 30 30 1.593788 0.2872139 1.160700 0.6575934  0.1747443 0.3444715
```

```
knn.fit$finalModel$k
```

```
## [1] 27
```