# Import the “train” and “test” datafiles in my working directory

setwd("`/R/Titanic")

train <- read.csv("~/R/Titanic/train.csv")

View(train)

test <- read.csv("~/R/Titanic/test.csv")

View(test)

train <- read.csv("train.csv", stringsAsFactors=FALSE)

# select pdf file as the output target

pdf("my\_report.pdf")

# create plot

plot(mtcars$mpg, main="Kernel Density of Miles Per Gallon", type="l")

# save pdf

dev.off()file

**Find the instances in train dataset with missing values**

train[!complete.cases(train),]

PassengerId Survived Pclass Name Sex Age AgeGr SibSp Parch Ticket

426 141 0 3 Boulos, Mrs. Joseph (Sultana) female NA 0 2 2678

427 594 0 3 Bourke, Miss. Mary female NA 0 2 364848

441 569 0 3 Doharr, Mr. Tannous male NA 0 0 2686

444 27 0 3 Emir, Mr. Farred Chehab male NA 0 0 2631

582 421 0 3 Gheorgheff, Mr. Stanio male NA 0 0 349254

628 525 0 3 Kassem, Mr. Fared male NA 0 0 2700

670 43 0 3 Kraeff, Mr. Theodor male NA 0 0 349253

671 523 0 3 Lahoud, Mr. Sarkis male NA 0 0 2624

673 879 0 3 Laleff, Mr. Kristo male NA 0 0 349217

684 37 1 3 Mamee, Mr. Hanna male NA 0 0 2677

686 20 1 3 Masselmani, Mrs. Fatima female NA 0 0 2649

758 698 1 3 Mullens, Miss. Katherine "Katie" female NA 0 0 35852

770 48 1 3 O'Driscoll, Miss. Bridget female NA 0 0 14311

784 585 0 3 Paulner, Mr. Uscher male NA 0 0 3411

799 860 0 3 Razi, Mr. Raihed male NA 0 0 2629

Fare Cabin Embarked

426 15.2458 C

427 7.7500 Q

441 7.2292 C

444 7.2250 C

582 7.8958 C

628 7.2292 C

670 7.8958 C

671 7.2250 C

673 7.8958 S

684 7.2292 C

686 7.2250 C

758 7.7333 Q

770 7.7500 Q

784 8.7125 C

799 7.2292 C

**Remove instances with missing values in the train dataset. After fixing 177 records with missing values I removed only 8 records that weren’t able to find aby information about their age.**

na.omit(train)

**Look at the dataframe structure**

> str(train)

'data.frame': 876 obs. of 9 variables:

$ PassengerId: int 259 680 738 28 89 342 439 312 743 119 ...

$ Survived : int 1 1 1 0 1 1 0 1 1 0 ...

$ Pclass : int 1 1 1 1 1 1 1 1 1 1 ...

$ SexN : int 1 2 2 2 1 1 2 1 1 2 ...

$ Age : num 35 36 35 19 23 24 64 18 21 24 ...

$ AgeN : int 40 40 40 20 30 30 70 20 30 30 ...

$ SibSp : int 0 0 0 3 3 3 1 2 2 0 ...

$ Parch : int 0 1 0 2 2 2 4 2 2 1 ...

$ Embarked : Factor w/ 4 levels "","C","Q","S": 2 2 2 4 4 4 4 2 2 2 ...

**Find how many people survived and died in Titanic disaster:**

table(train$Survived)

0 1

538 338 - We see that in the training set, 338 passengers survived, while 538 died

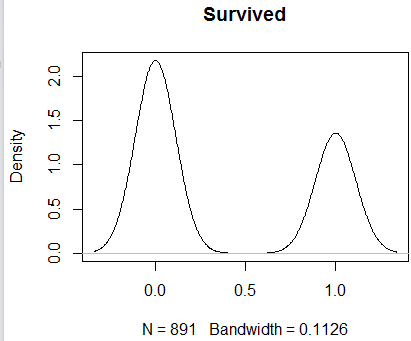
> prop.table(table(train$Survived))

0 1

0.6141553 0.3858447 38% of passengers survived the disaster in the training set and 61% died.

**Histogram and density**

plot(density(train$Survived), main="Survived")



**Finding Correlation Coefficients Pearson:**

> cor(train$Survived, train$SexN, method = "pearson")

[1] -0.5436527

> cor(train$Survived, train$Pclass, method = "pearson")

[1] -0.338481

> cor(train$Survived, train$Age, method = "pearson")

[1] -0.051832

> cor(train$Survived, train$SibSp, method = "pearson")

[1] -0.0353225

Pearson **Correlation Coefficient Calculator**. The Pearson **correlation coefficient** is used to measure the strength of a linear association between two variables, where the value r = 1 means a perfect positive **correlation** and the value r = -1 means a perfect negataive **correlation**.

for example, 0.92 or -0.97 would show, respectively, a very strong positive and negative correlation. As with all statistics that demonstrate correlation, this does not prove causation.

> cor(train[,1:7])

Survived Pclass SexN Age SibSp Parch

Survived 1.00000000 -0.33959068 -0.54365274 -0.05183200 -0.03763407 0.08525835

Pclass -0.33959068 1.00000000 0.13320941 -0.38969350 0.09051013 0.02071297

SexN -0.54365274 0.13320941 1.00000000 0.08914444 -0.11547931 -0.24108181

Age -0.05183200 -0.38969350 0.08914444 1.00000000 -0.30108803 -0.20504350

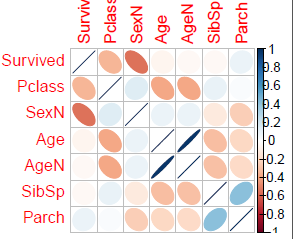
AgeN -0.03919002 -0.38446242 0.08438302 0.97848311 -0.29422982 -0.19641856

SibSp -0.03763407 0.09051013 -0.11547931 -0.30108803 1.00000000 0.41705931

Parch 0.08525835 0.02071297 -0.24108181 -0.20504350 0.41705931 1.00000000

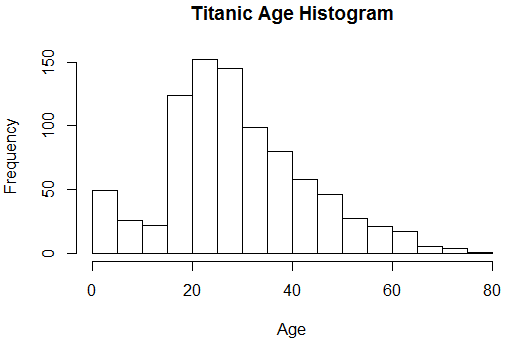
**Visualizing Correlation**

corrplot(cor(train[, c(1,2,3,4,5,6,7)]), method="ellipse")]



In this corrplot negative correclation are showed in brown color and positive correlations are showed in blue color. Darker color and xize of the circles are showing proportional correlation coefficient.

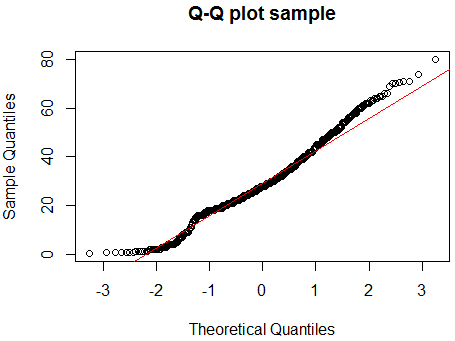
hist(train$Age, xlab= "Age",main="Histogram for Age", breaks=20)

 From this histogram we can see that Titanic has more passaged whose ages are between 20 - 30.

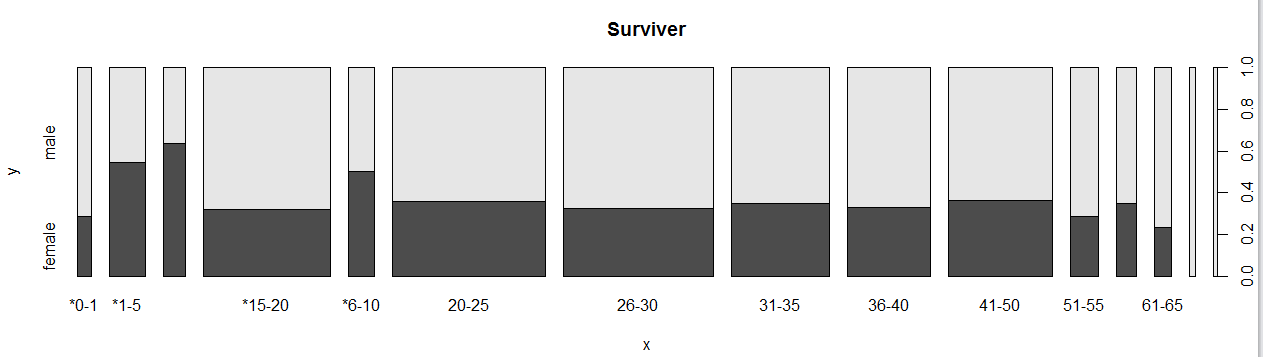
Careate a Quantile-Quantile plat to evaluate the fit of my dataset to the normal distribution.

> qqnorm(train$Age, main="Q-Q plot sample")

> qqline(train$Age, col="red")

 Based on my QQ Plot the quantiles of Age data represented by a straight line and considered as standard normal distribution.

plot(train$AgeGr,train$Sex, main="Titanic Age Group By Sex", type="l", type="2")



**Titanic Age Groups by Sex**

> prop.table(table(train$AgeGr, train$Sex))

female male

\*0-1 0.004566210 0.011415525

\*1-5 0.021689498 0.018264840

\*11-15 0.015981735 0.009132420

\*15-20 0.045662100 0.097031963

\*6-10 0.014840183 0.014840183

20-25 0.061643836 0.110730594

26-30 0.054794521 0.113013699

31-35 0.038812785 0.071917808

36-40 0.030821918 0.062785388

41-50 0.042237443 0.074200913

51-55 0.009132420 0.022831050

56-60 0.007990868 0.014840183

61-65 0.004566210 0.014840183

66-70 0.000000000 0.006849315

71-80 0.000000000 0.004566210

summary(train$Sex)

female male

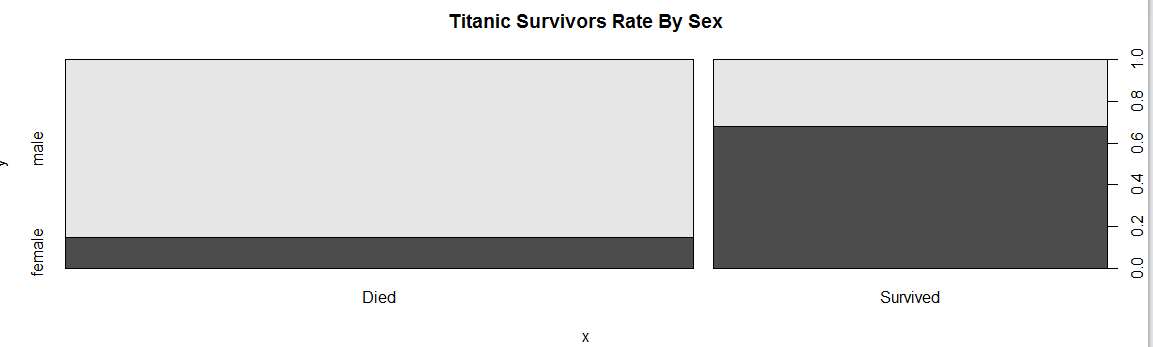
309 567 – Take a look at summary of Sex in the dataset

> prop.table(table(train$Sex))

female male

0.3527397 0.6472603 64% of Titanic passengers are man

> plot(train$Survived,train$Sex, main="Titanic Survivors Rate By Sex", type="l", type="2")



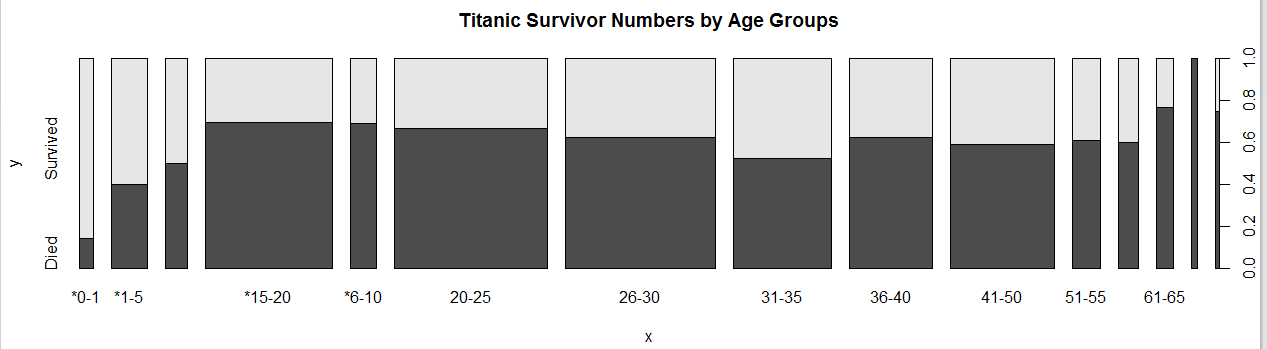
> prop.table(table(train$Sex, train$Survived),1)

0 1

female 0.2556634 0.7443366

male 0.8095238 0.1904762 # Look at Survivor gender patterns

> plot(train$AgeGr,train$SurvivedW, main="Titanic Survivor Numbers by Age Group", type="l", type="2")



> prop.table(table(train$AgeGr, train$Survived))

0 1

\*0-1 0.002283105 0.013698630

\*1-5 0.015981735 0.023972603

\*11-15 0.012557078 0.012557078

\*15-20 0.099315068 0.043378995

\*6-10 0.020547945 0.009132420

20-25 0.115296804 0.057077626

26-30 0.105022831 0.062785388

31-35 0.058219178 0.052511416

36-40 0.058219178 0.035388128

41-50 0.068493151 0.047945205

51-55 0.019406393 0.012557078

56-60 0.013698630 0.009132420

61-65 0.014840183 0.004566210

66-70 0.006849315 0.000000000

71-80 0.003424658 0.001141553

summary(train$Age)

Min. 1st Qu. Median Mean 3rd Qu. Max.

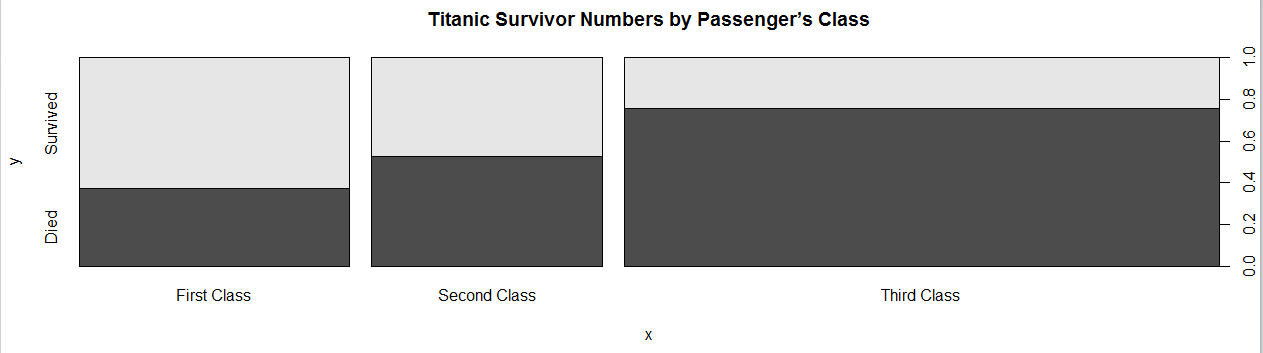
0.42 20.00 28.00 29.51 38.00 80.00 My Median Survivor Age in the dataset is **28** and **Min-0.42** and **Max-80**

> summary(train$AgeGr)

\*0-1 \*1-5 \*11-15 \*15-20 \*6-10 20-25 26-30 31-35 36-40 41-50 51-55 56-60 61-65 66-70 71-80

14 35 22 125 26 151 147 97 82 102 28 20 17 6 4

plot(train$PclassW,train$SurvivedW, main="Titanic Survivor Numbers by Passenger’s Class", type="l", type="2")



> prop.table(table(train$Pclass, train$Survived))

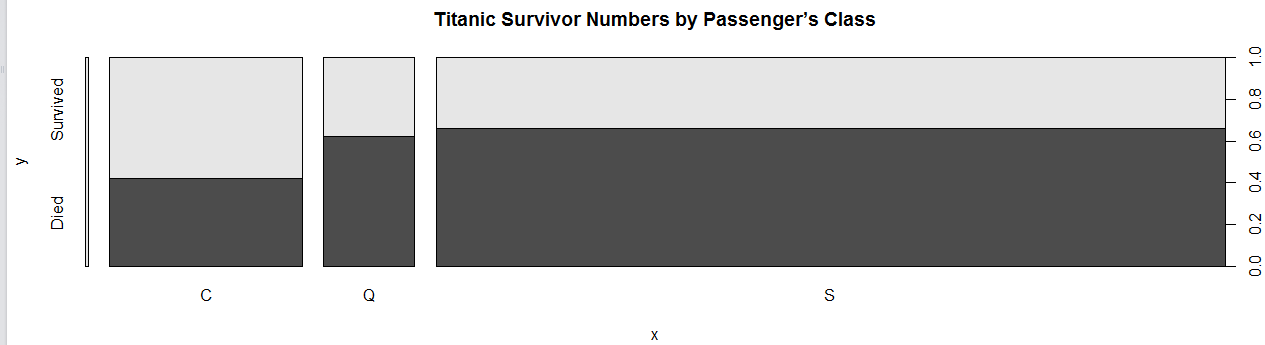
0 1

1 0.09132420 0.15525114

2 0.11073059 0.09931507

3 0.41210046 0.13127854

> plot(train$Embarked,train$SurvivedW, main="Titanic Survivor Numbers by Embarked Area ", type="l", type="2")



**Titanic Survivor Numbers by Embarked Area**

> prop.table(table(train$Embarked, train$Survived))

0 1

0.000000000 0.002283105

C 0.075342466 0.103881279

Q 0.052511416 0.031963470

S 0.486301370 0.247716895

> summary(train$Embarked)

C Q S

2 157 74 643

Southampton passengers have a bigger death rate compare to passengers from Cherbourg and Queenstown.

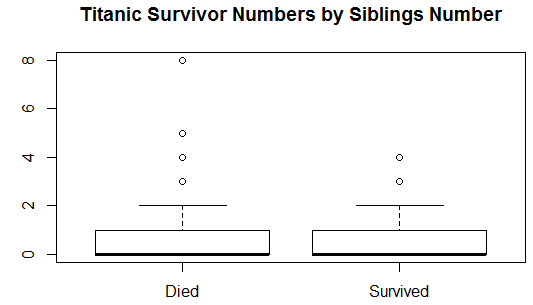
> prop.table(table(train$Embarked))

C Q S

0.002283105 0.179223744 0.084474886 0.734018265 73% of passengers from Southampton died

**Created a Boxplot to see frequencies on Titanic survivor numbers by Sibling number.**

plot(train$SurvivedW,train$SibSp, main="Titanic Survivor Numbers by Siblings Number", type="l", type="2")



> prop.table(table(train$Survived, train$SibSp))

0 1 2 3 4 5

0 0.441780822 0.110730594 0.017123288 0.013698630 0.017123288 0.005707763

1 0.235159817 0.127853881 0.014840183 0.004566210 0.003424658 0.000000000

8

0 0.007990868

1 0.000000000

> summary(train$SibSp)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.000 0.000 0.000 0.532 1.000 8.000

**Significance of Correlation Coefficients**

cor.test(train$Survived, train$SexN, method = c("pearson"))

Pearson's product-moment correlation

data: train$Survived and train$SexN

t = -19.149, df = 874, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.5886914 -0.4952492

sample estimates:

cor

-0.5436527

> cor.test(train$Survived, train$Age, method = c("pearson"))

Pearson's product-moment correlation

data: train$Survived and **train$Age**

t = -1.5344, df = 874, p-value = 0.1253 (P value is statistically significant)

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.11766563 0.01445524

sample estimates:

cor

-0.051832

> cor.test(train$Survived, train$Pclass, method = c("pearson"))

Pearson's product-moment correlation

data: train$Survived and train$Pclass

t = -10.674, df = 874, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.3969005 -0.2796433

sample estimates:

cor

-0.3395907

> cor.test(train$Survived, train$EmbarkedN, method = c("pearson"))

Pearson's product-moment correlation

data: train$Survived and **train$EmbarkedN**

t = 3.4993, df = 874, p-value = 0.0004901 (P value is **VERY** statistically significant)

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.05170904 0.18236181

sample estimates:

cor

0.117544

> cor.test(train$Survived, train$SibSp, method = c("pearson"))

Pearson's product-moment correlation

data: train$Survived and **train$SibSp**

t = -1.1134, df = 874, p-value = 0.2659 (P value is statistically significant)

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.10361338 0.02867502

sample estimates:

cor

-0.03763407

**Choose a single variable to predict Survival rate and build an univariate linear regression**

**Model**

Linear Regression: Fitting the Model

> model\_titanic <- lm(Survived~SexN, data=train)

> summary(model\_titanic)

Call:

lm(formula = Survived ~ SexN, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.7443 -0.1905 -0.1905 0.2557 0.8095

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.29820 0.04961 26.17 <2e-16 \*\*\*

SexN -0.55386 0.02892 -19.15 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.409 on 874 degrees of freedom

Multiple R-squared: 0.2956, Adjusted R-squared: 0.2948

F-statistic: 366.7 on 1 and 874 DF, p-value: < 2.2e-16

|  |
| --- |
| The regression equation is *minutes* = 4.162 + 15.509\**units*. The "*t* values" test the hypotheses that the corresponding population parameters are 0. Usually we test whether the slope is zero because if it is then the model is not much use. Here the  p  -value for that test is "8.92e-13" which is to say 8.92X10-13 or 0.000000000000892, so we would reject the hypothesis that the slope is zero. If you wish to test a nonzero value, subtract it from the coefficient in the regression output (15.509) and divide the result by the coefficient's s.e. (0.505). (Use a calculator for this.) Similarly, if you want confidence intervals, use the coefficient plus or minus the product of its s.e. with a *t*-value for the desired confidence level and 12 degrees of freedom. (Use a calculator for this.) This also works for the intercept (4.162) using its s.e. (3.355). |

> model\_titanic <- lm(Survived~**Age**, data=train)

> summary(model\_titanic)

Call:

lm(formula = Survived ~ Age, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.4363 -0.3938 -0.3638 0.6044 0.7035

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.438029 0.037777 11.595 <2e-16 \*\*\*

Age -0.001769 0.001153 -1.534 0.125

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4867 on 874 degrees of freedom

Multiple R-squared: 0.002687, Adjusted R-squared: 0.001545

F-statistic: 2.354 on 1 and 874 DF, p-value: 0.1253

> model\_titanic <- lm(**Survived~EmbarkedN**, data=train)

> summary(model\_titanic)

Call:

lm(formula = Survived ~ EmbarkedN, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.5345 -0.3540 -0.3540 0.5557 0.6460

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.26374 0.03854 6.844 1.45e-11 \*\*\*

EmbarkedN 0.09027 0.02580 3.499 0.00049 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.484 on 874 degrees of freedom

Multiple R-squared: 0.01382, Adjusted R-squared: 0.01269

F-statistic: 12.24 on 1 and 874 DF, p-value: 0.0004901

> model\_titanic <- lm(**Survived~SibSp**, data=train)

> summary(model\_titanic)

Call:

lm(formula = Survived ~ SibSp, data = train)

Residuals:

Min 1Q Median 3Q Max

-0.3946 -0.3946 -0.3781 0.6054 0.6714

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.39463 0.01825 21.626 <2e-16 \*\*\*

SibSp -0.01651 0.01483 -1.113 0.266

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.487 on 874 degrees of freedom

Multiple R-squared: 0.001416, Adjusted R-squared: 0.0002738

F-statistic: 1.24 on 1 and 874 DF, p-value: 0.2659

# Linear Regression: Prediction

> my\_train <- sample(nrow(train),

+ floor(nrow(train)\*0.8))

> train <- train[my\_train,c("Survived","Age")]

> test <- train[-my\_train,]

> model\_ulm <- lm(Survived~Age, data=train)

> prediction <- predict(model\_ulm, interval="prediction",

+ newdata =test)

> errors <- prediction[,"fit"] - test$Survived

> hist(errors)

****

Linear Regression: RMSE

> sqrt(sum((prediction[,"fit"] - test$Survived)^2)/nrow(test))

[1] 0.4917012

Linear Regression: PRED (25) – find number of records with 25% error

> rel\_change <- 1 - ((test$Survived - abs(errors)) / test$Survived)

> table(rel\_change<0.25)["TRUE"] / nrow(test)

<NA>

NA