Chapter 3- Regression Prediction

Created by Shivakumar Panuganti, spanugan@asu.edu

Simple Regression Model

Using the Amusement park data provided in the Book

```
df<-read.csv("http://goo.gl/HKnl74")
head(df)
```

```
##   weekend num.child  distance rides games wait clean overall
## 1     yes         0 114.64826    87    73   60    89      47
## 2     yes         2  27.01410    87    78   76    87      65
## 3      no         1  63.30098    85    80   70    88      61
## 4     yes         0  25.90993    88    72   66    89      37
## 5      no         4  54.71831    84    87   74    87      68
## 6      no         5  22.67934    81    79   48    79      27
```

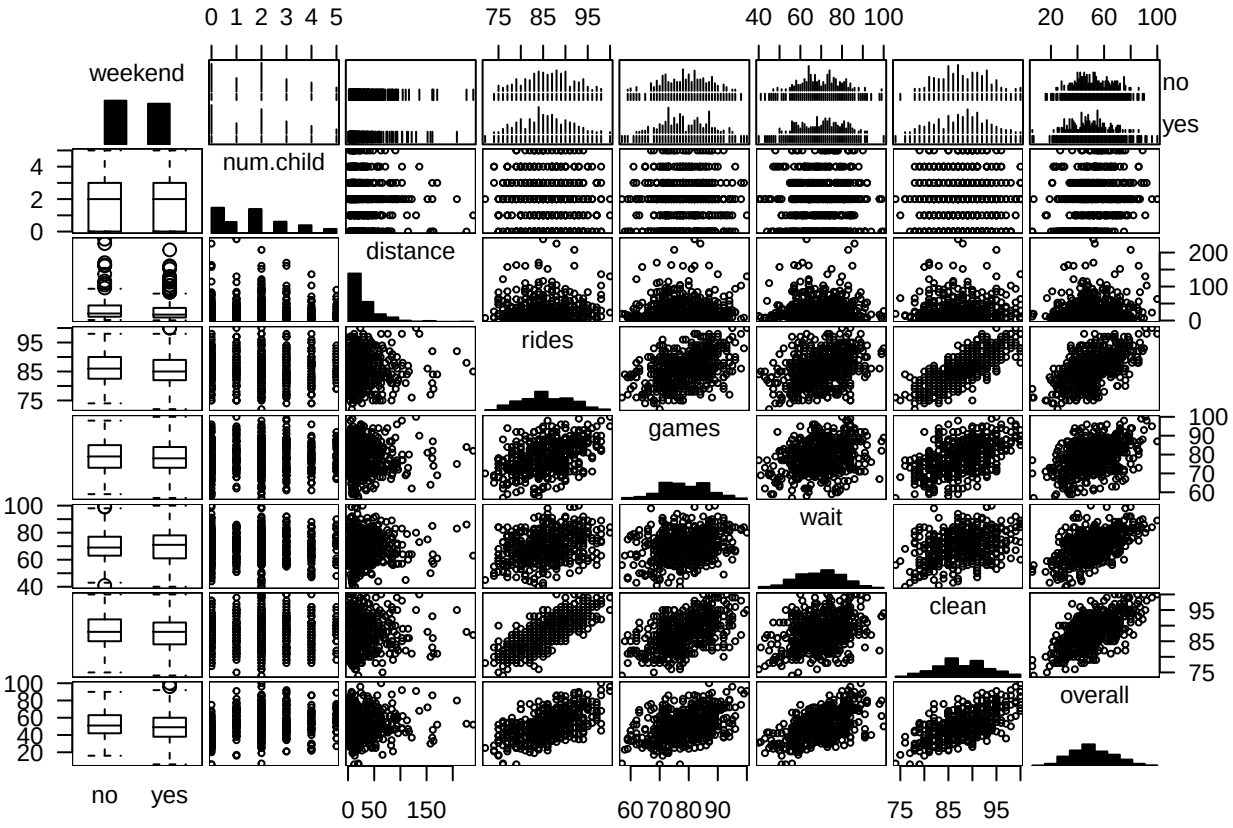
```
dim(df)
```

```
## [1] 500   8
```

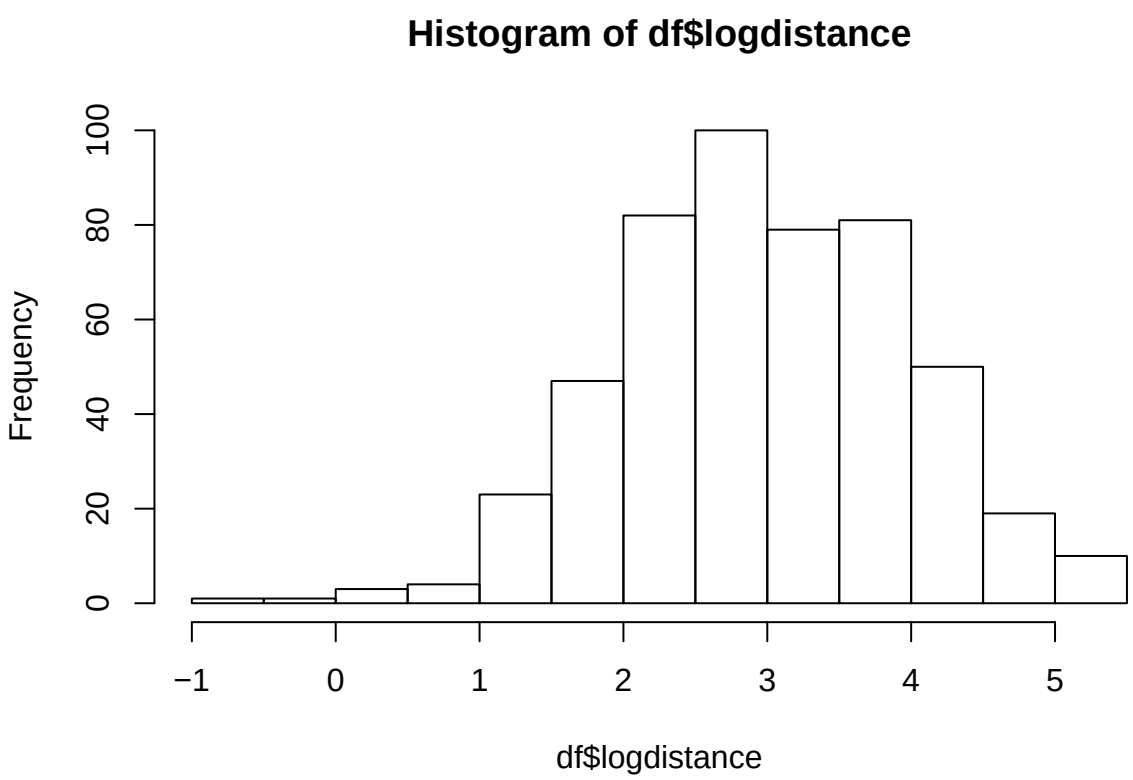
```
library(gpairs)
gpairs(df) #Graphical Summary
```

```
## Loading required package: grid
```

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




```
#distance is skwed, so we can apply log transform for it
df$logdistance<- log(df$distance)
hist(df$logdistance)
```



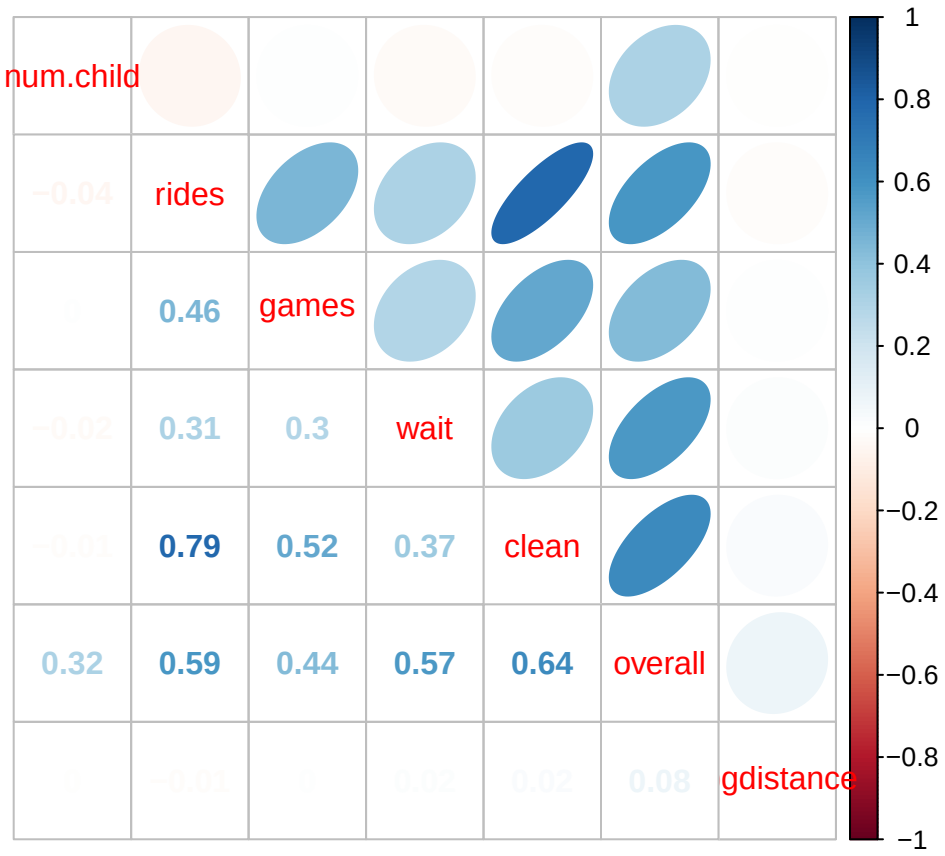

```
head(df)
```

```
##   weekend num.child  distance rides games wait clean overall logdistance
## 1     yes         0 114.64826    87    73   60    89      47    4.741869
## 2     yes         2  27.01410    87    78   76    87      65    3.296359
## 3      no         1  63.30098    85    80   70    88      61    4.147901
## 4     yes         0  25.90993    88    72   66    89      37    3.254626
## 5      no         4  54.71831    84    87   74    87      68    4.002198
## 6      no         5  22.67934    81    79   48    79      27    3.121454
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot.mixed(cor(df[,c(2,4:9)]), upper="ellipse")
```

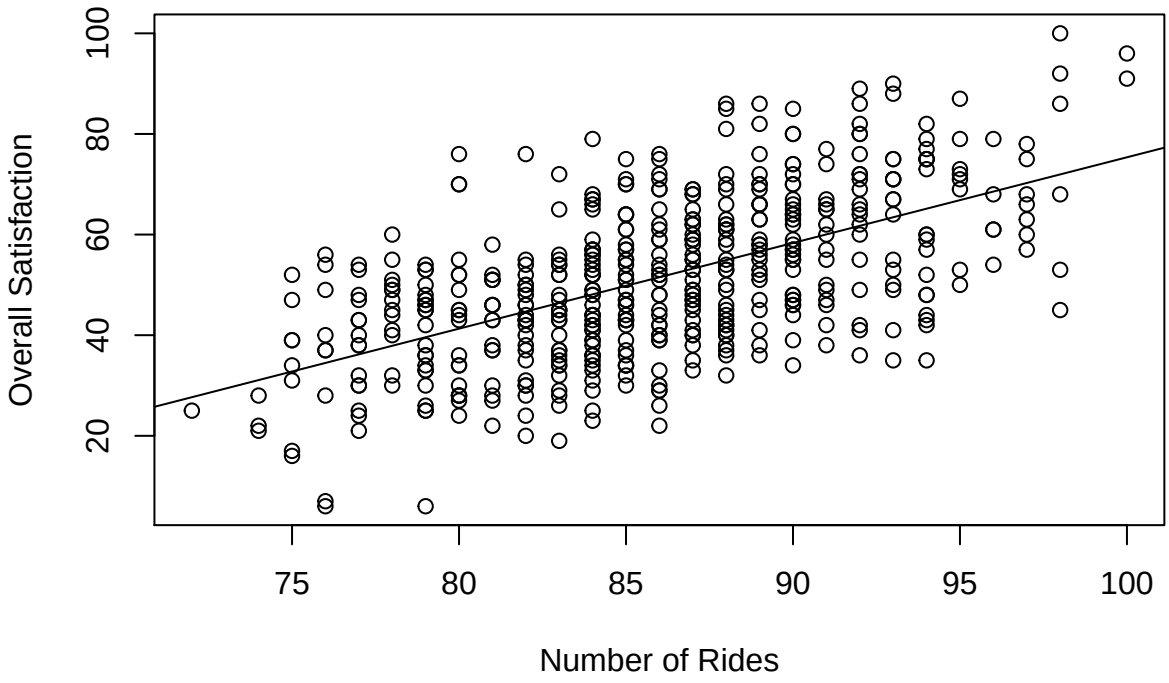


Regression with single independent variable #### Let's consider the variable to be rides

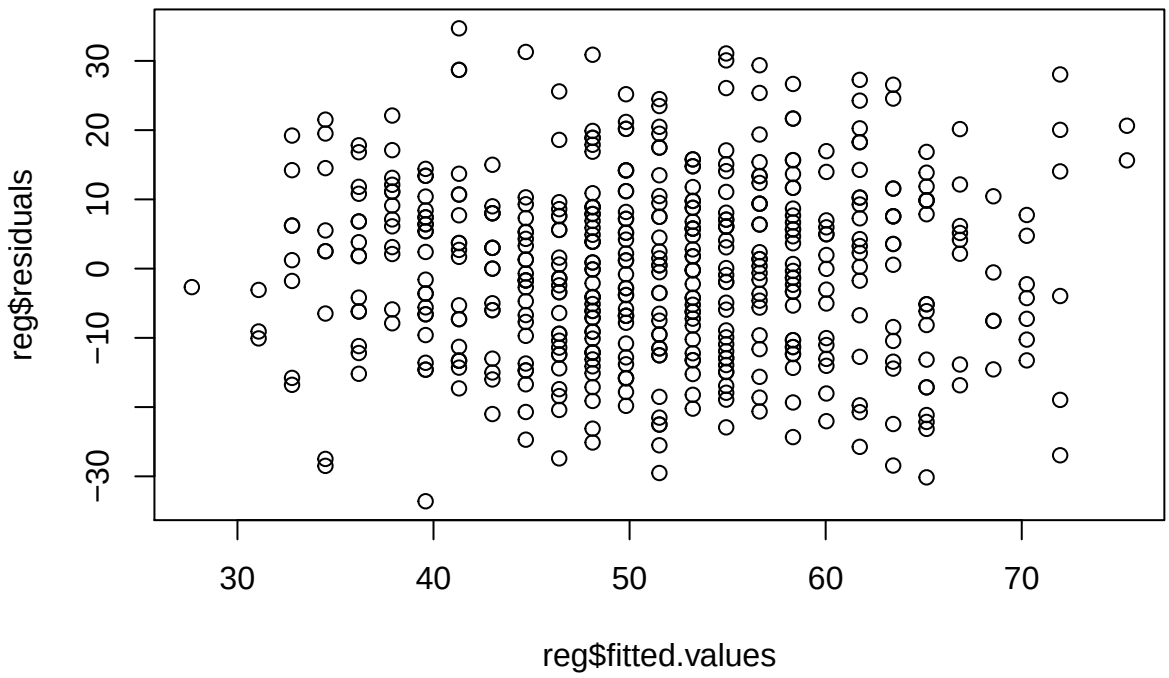
```
reg=lm(overall~rides, data=df)
summary(reg)
```

```
##
## Call:
## lm(formula = overall ~ rides, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.597 -10.048   0.425   8.694  34.699
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -94.9622     9.0790  -10.46   <2e-16 ***
## rides         1.7033     0.1055   16.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.88 on 498 degrees of freedom
## Multiple R-squared:  0.3434, Adjusted R-squared:  0.3421
## F-statistic: 260.4 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
plot(overall~rides,data=df, xlab="Number of Rides", ylab= "Overall Satisfaction")
reg=lm(overall~rides, data=df)
abline(reg)
```




```
plot(reg$fitted.values,reg$residuals) #residual plot
```



Prediction‘

```
names(reg)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
coef(reg)
```

```
## (Intercept)       rides
##  -94.962246    1.703285
```

```
confint(reg)
```

```
##                    2.5 %     97.5 %
## (Intercept) -112.800120 -77.124371
## rides          1.495915   1.910656
```

```
predict(reg, data.frame(rides=c(95)), interval="confidence")
```

```
##        fit      lwr      upr
## 1 66.84988 64.63986 69.05989
```

```
predict(reg, data.frame(rides=c(95)), interval="prediction")
```

```
##        fit      lwr      upr
## 1 66.84988 41.44827 92.25148
```

The 95% confidence interval associated with a ride value of 95 is (64.63,,69.05), and the 95% prediction interval is (41.44,92.25)

Multiple Linear Regression

Data Normalization

```
df_std<-df[,-3] #Drop the distance column
dim(df_std)
```

```
## [1] 500   8
```

```
head(df_std)
```

```
##   weekend num.child rides games wait clean overall logdistance
## 1     yes         0    87    73   60    89      47    4.741869
## 2     yes         2    87    78   76    87      65    3.296359
## 3      no         1    85    80   70    88      61    4.147901
## 4     yes         0    88    72   66    89      37    3.254626
## 5      no         4    84    87   74    87      68    4.002198
## 6      no         5    81    79   48    79      27    3.121454
```

```
df_std[,3:8]<- scale(df_std[,3:8]) # Data Normalization
head(df_std)
```

```
##   weekend num.child      rides       games         wait       clean
## 1     yes         0  0.2112477 -0.69750817 -0.918784090  0.21544189
## 2     yes         2  0.2112477 -0.08198737  0.566719693 -0.17555973
## 3      no         1 -0.1548662  0.16422095  0.009655775  0.01994108
## 4     yes         0  0.3943047 -0.82061233 -0.361720171  0.21544189
## 5      no         4 -0.3379232  1.02595006  0.381031720 -0.17555973
## 6      no         5 -0.8870941  0.04111679 -2.032911927 -1.73956621
##      overall logdistance
## 1 -0.2681587   1.7886823
## 2  0.8654385   0.3226360
## 3  0.6135280   1.1862757
## 4 -0.8979350   0.2803106
## 5  1.0543714   1.0385034
## 6 -1.5277112   0.1452467
```

```
m_reg<- lm(overall~ rides+ games+ wait+ clean+ weekend+ logdistance+ num.child, data= df_std)
summary(m_reg)
```

```
##
## Call:
## lm(formula = overall ~ rides + games + wait + clean + weekend +
##     logdistance + num.child, data = df_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51427 -0.40271  0.01142  0.41613  1.69000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.37271    0.04653  -8.009 8.41e-15 ***
## rides        0.21288    0.04197   5.073 5.57e-07 ***
```

```
## games        0.07066    0.03026   2.335   0.0199 *
## wait         0.38138    0.02777  13.734  < 2e-16 ***
## clean        0.29690    0.04415   6.725 4.89e-11 ***
## weekendyes  -0.04589    0.05141  -0.893   0.3725
## logdistance  0.06470    0.02572   2.516   0.0122 *
## num.child    0.22717    0.01711  13.274  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5709 on 492 degrees of freedom
## Multiple R-squared:  0.6786, Adjusted R-squared:  0.674
## F-statistic: 148.4 on 7 and 492 DF,  p-value: < 2.2e-16
```

One of the most important uses of factors is in statistical modeling; since categorical variables enter into statistical models differently than continuous variables, storing data as factors insures that the modeling functions will treat such data correctly.

```
factor(df_std$num.child)
```

```
##   [1] 0 2 1 0 4 5 1 0 0 3 1 4 2 2 2 2 0 0 2 2 2 3 4 0 3 2 3 4 0 0 2 1 0 0 0
##  [36] 0 4 0 2 0 2 1 3 2 2 0 0 0 2 0 1 2 4 2 2 2 2 0 0 0 2 0 2 1 2 3 4 5 1 4
##  [71] 2 0 5 2 4 2 2 3 4 4 2 3 0 1 3 1 0 0 2 0 5 0 2 0 2 0 0 3 1 4 1 4 2 2 0
## [106] 0 0 1 2 0 0 4 2 3 0 1 0 0 5 0 0 2 2 2 5 2 1 3 0 4 1 0 2 0 3 0 0 3 2 3
## [141] 3 2 2 3 0 1 4 2 5 1 3 0 4 0 1 4 2 3 5 4 0 1 3 2 1 0 0 4 2 3 0 5 1 2 3
## [176] 1 1 3 0 0 0 4 3 2 2 1 0 4 2 2 5 2 2 3 3 3 3 2 4 3 0 2 2 2 2 3 0 0 5 0
## [211] 4 2 0 0 1 2 0 1 2 2 0 0 1 1 1 0 1 4 2 0 1 2 2 2 3 3 1 2 4 0 2 0 3 0 1
## [246] 5 2 2 1 2 2 0 2 2 5 2 4 1 1 2 2 0 2 3 0 2 2 0 3 0 0 2 2 0 4 0 0 0 2 4
## [281] 3 5 3 0 2 0 1 4 2 1 2 0 2 3 0 0 1 1 2 2 2 2 0 2 2 1 2 3 5 2 2 0 1 0 1
## [316] 3 0 3 2 2 0 0 2 5 2 0 0 0 0 3 0 2 2 3 1 3 3 0 2 2 0 0 1 2 0 2 1 2 0 2
## [351] 2 0 5 1 3 1 1 4 0 3 3 2 0 4 2 1 0 0 5 0 1 0 4 4 0 2 0 3 3 0 3 2 2 2 0
## [386] 2 0 2 2 0 3 5 0 0 0 2 2 3 4 0 2 0 4 2 0 2 1 3 1 3 4 0 3 1 2 0 5 1 0 3
## [421] 2 3 0 4 0 0 2 1 1 0 1 5 2 1 0 0 2 2 5 4 0 2 3 3 2 0 4 0 4 2 2 0 1 3 3
## [456] 0 4 0 0 2 1 0 4 0 0 3 0 3 3 0 0 0 3 4 0 2 0 0 2 5 5 2 1 4 2 2 4 1 3 4
## [491] 2 0 2 1 5 0 0 2 3 1
## Levels: 0 1 2 3 4 5
```

```
#factor as an Independent Variable
df_std$num.child.factor<- factor(df_std$num.child)
```

```
df_std[1:5,] #This is same as head(df_std)
```

```
##   weekend num.child      rides       games         wait       clean
## 1     yes         0  0.2112477 -0.69750817 -0.918784090  0.21544189
## 2     yes         2  0.2112477 -0.08198737  0.566719693 -0.17555973
## 3      no         1 -0.1548662  0.16422095  0.009655775  0.01994108
## 4     yes         0  0.3943047 -0.82061233 -0.361720171  0.21544189
## 5      no         4 -0.3379232  1.02595006  0.381031720 -0.17555973
##      overall logdistance num.child.factor
## 1 -0.2681587   1.7886823                0
## 2  0.8654385   0.3226360                2
## 3  0.6135280   1.1862757                1
## 4 -0.8979350   0.2803106                0
## 5  1.0543714   1.0385034                4
```

```
m_reg1<- lm(overall~ rides+ games+ wait+ clean+ weekend+ logdistance+ num.child.factor, data= df_std)
summary(m_reg1)
```

```
##
## Call:
## lm(formula = overall ~ rides + games + wait + clean + weekend +
##     logdistance + num.child.factor, data = df_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25923 -0.35048 -0.00154  0.31400  1.52690
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.69100    0.04488 -15.396  < 2e-16 ***
## rides              0.22313    0.03541   6.301 6.61e-10 ***
## games              0.04258    0.02551   1.669   0.0958 .
## wait               0.38472    0.02338  16.453  < 2e-16 ***
## clean              0.30917    0.03722   8.308 9.72e-16 ***
## weekendyes        -0.02227    0.04322  -0.515   0.6065
## logdistance        0.03187    0.02172   1.467   0.1429
## num.child.factor1  1.01610    0.07130  14.250  < 2e-16 ***
## num.child.factor2  1.03732    0.05640  18.393  < 2e-16 ***
## num.child.factor3  0.98000    0.07022  13.955  < 2e-16 ***
## num.child.factor4  0.93154    0.08032  11.598  < 2e-16 ***
## num.child.factor5  1.00193    0.10369   9.663  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4795 on 488 degrees of freedom
## Multiple R-squared:  0.7751, Adjusted R-squared:   0.77
## F-statistic: 152.9 on 11 and 488 DF,  p-value: < 2.2e-16
```

```
AIC(m_reg); AIC(m_reg1)  #information criteria
```

```
## [1] 868.3958
```

```
## [1] 697.8504
```

```
BIC(m_reg); BIC(m_reg1)
```

```
## [1] 906.3272
```

```
## [1] 752.6403
```

When comparing models fitted by maximum likelihood to the same data, the smaller the AIC or BIC, the better the fit.

```
# Model with binary has.child variable
df_std$has.child<- factor(df_std$num.child>0)
head(df_std)
```

```
##   weekend num.child      rides       games         wait       clean
## 1     yes         0  0.2112477 -0.69750817 -0.918784090  0.21544189
## 2     yes         2  0.2112477 -0.08198737  0.566719693 -0.17555973
## 3      no         1 -0.1548662  0.16422095  0.009655775  0.01994108
## 4     yes         0  0.3943047 -0.82061233 -0.361720171  0.21544189
## 5      no         4 -0.3379232  1.02595006  0.381031720 -0.17555973
```

```
## 6      no         5 -0.8870941  0.04111679 -2.032911927 -1.73956621
##      overall logdistance num.child.factor has.child
## 1 -0.2681587   1.7886823                0     FALSE
## 2  0.8654385   0.3226360                2      TRUE
## 3  0.6135280   1.1862757                1      TRUE
## 4 -0.8979350   0.2803106                0     FALSE
## 5  1.0543714   1.0385034                4      TRUE
## 6 -1.5277112   0.1452467                5      TRUE
```

```
m_reg2<- lm(overall~ rides+ games+ wait+ clean+ weekend+ logdistance+ has.child, data= df_std)
summary(m_reg2)
```

```
##
## Call:
## lm(formula = overall ~ rides + games + wait + clean + weekend +
##     logdistance + has.child, data = df_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22366 -0.35107 -0.01747  0.31852  1.45703
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.69039    0.04478 -15.418  < 2e-16 ***
## rides          0.22193    0.03517   6.310 6.24e-10 ***
## games          0.04409    0.02541   1.735   0.0833 .
## wait           0.38603    0.02328  16.583  < 2e-16 ***
## clean          0.30904    0.03699   8.354 6.75e-16 ***
## weekendyes    -0.02280    0.04311  -0.529   0.5971
## logdistance    0.03404    0.02159   1.576   0.1156
## has.childTRUE  1.00485    0.04689  21.428  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4785 on 492 degrees of freedom
## Multiple R-squared:  0.7742, Adjusted R-squared:  0.771
## F-statistic:   241 on 7 and 492 DF,  p-value: < 2.2e-16
```

```
AIC(m_reg); AIC(m_reg1); AIC(m_reg2)   #information criteria
```

```
## [1] 868.3958
```

```
## [1] 697.8504
```

```
## [1] 691.8489
```

```
BIC(m_reg); BIC(m_reg1); BIC(m_reg2)
```

```
## [1] 906.3272
```

```
## [1] 752.6403
```

```
## [1] 729.7804
```

Combining factors to binary variable gave better AIC and BIC. This is a type of model selection procedure.

Bayesian Linear Models

```
library(MCMCpack)
```

```
## Loading required package: coda
```

```
## Loading required package: MASS
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
m_reg3<- MCMCregress(overall~rides + games + wait + clean + weekend +
    logdistance + has.child, data = df_std)
summary(m_reg3)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                   Mean      SD  Naive SE Time-series SE
## (Intercept)   -0.69058 0.04499 0.0004499      0.0004499
## rides          0.22191 0.03558 0.0003558      0.0003501
## games          0.04425 0.02547 0.0002547      0.0002547
## wait           0.38602 0.02327 0.0002327      0.0002327
## clean          0.30865 0.03731 0.0003731      0.0003731
## weekendyes    -0.02325 0.04327 0.0004327      0.0004327
## logdistance    0.03395 0.02185 0.0002185      0.0002185
## has.childTRUE  1.00557 0.04674 0.0004674      0.0004674
## sigma2         0.22959 0.01471 0.0001471      0.0001471
##
## 2. Quantiles for each variable:
##
##                    2.5%      25%      50%       75%    97.5%
## (Intercept)   -0.776710 -0.72108 -0.69075 -0.660552 -0.60153
## rides          0.150788  0.19799  0.22220  0.245657  0.29144
## games         -0.006029  0.02724  0.04422  0.060862  0.09511
## wait           0.340876  0.37030  0.38583  0.401745  0.43173
## clean          0.235079  0.28341  0.30858  0.334191  0.38148
## weekendyes    -0.108241 -0.05269 -0.02324  0.006212  0.06022
## logdistance   -0.009039  0.01914  0.03413  0.048719  0.07671
## has.childTRUE  0.912580  0.97429  1.00596  1.036403  1.09823
## sigma2         0.202587  0.21921  0.22888  0.239034  0.26023
```