Deliverable4

Introduction

- Interest from a business perspective: helps bike rental businesses meet demands
- City planning perspective: helps cities to adapt to the change of number of bikers to enforce better traffic laws
- A way to sense mobility in the city

Backgrounds

The data is a two-year historical log corresponding to years 2011 and 2012 from Capital Bikeshare system, Washington D.C. containing the following datas: weathersit: 1: Clear, Few clouds, Partly cloudy, 2: Mist and Cloudy, Mist and Broken clouds, Mist and Few clouds, Mist 3: Light Snow, Light Rain and Thunderstorm and Scattered clouds, Light Rain an dScattered clouds 4: Heavy Rain and Ice Pallets and Thunderstorm and Mist, Snow and Fog instant: record index

dteday: date

season: season (1:spring, 2:summer, 3:fall, 4:winter)

yr: year (0: 2011, 1:2012)

mnth: month (1 to 12)

holiday: weather day is holiday or not (extracted from http://dchr.dc.gov/page/holiday-schedule)

weekday: day of the week

workingday: if day is neither weekend nor holiday is 1, otherwise is 0.

temp: Normalized temperature in Celsius. The values are divided to 41 (max)

atemp: Normalized feeling temperature in Celsius. The values are divided to 50 (max)

hum: Normalized humidity. The values are divided to 100 (max)

windspeed: Normalized wind speed. The values are divided to 67 (max)

casual: count of casual users

registered: count of registered users

cnt: count of total rental bikes including both casual and registered

Our goal is to use data in 2011 to predict bike rential behaviour in 2012.

Preprocessing

```r
bikedata <- read.csv("day.csv",header=T)
names(bikedata)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"
##  [6] "holiday"    "weekday"    "workingday" "weathersit" "temp"
## [11] "atemp"      "hum"        "windspeed"  "casual"     "registered"
## [16] "cnt"
```

```r
#Transform temp, atemp, windspeed, and humidity to actual values
bikedata <-
  bikedata %>% mutate(actual.temp = temp*41) %>%
  mutate(actual.atemp = atemp*50) %>%
  mutate(actual.windspeed = windspeed*67) %>%
  mutate(actual.hum = hum*100)

#Combining summer, fall, and spring, winter
bikedata <- bikedata %>% mutate(season.2 = if_else(season == 2|season==3|season==4,0,if_else(season ==1

#process factor data
bikedata$season <- factor(format(bikedata$season, format="%A"),
                   levels = c("1", "2","3","4") ,
                   labels = c("Spring","Summer","Fall","Winter"))

bikedata$spring <- factor(format(bikedata$season.2, format="%A"),
                   levels = c("0","1") ,
                   labels = c("Not Spring","Spring"))

bikedata$holiday <-factor(format(bikedata$holiday, format="%A"),
                          levels = c("0", "1") ,
                          labels = c("Not Holiday","Holiday"))

bikedata$weathersit <- factor(format(bikedata$weathersit, format="%A"),
                       levels = c("1", "2","3","4") ,
                       labels = c("Good:Clear/Sunny","Moderate:Cloudy/Mist","Bad: Rain/Snow/Fog","Worse

bikedata$workingday <- factor(format(bikedata$workingday, format = "%A"),
                              levels = c("0", "1"),
                              labels = c("Not WorkingDay", "WorkingDay"))

bikedata$yr <- factor(format(bikedata$yr, format="%A"),
                          levels = c("0", "1") , labels = c("2011","2012"))

bikedata <- bikedata %>% mutate(weekend = if_else(weekday == 0|weekday==6,0,if_else(weekday ==1|weekday

bikedata$weekend <- factor(format(bikedata$weekend, format = "%A"),
                           levels = c(0,1),
                           labels = c("Weekend", "Weekday"))

bikedata$mnth <- as.factor(bikedata$mnth)



#Generate days from start date values
```

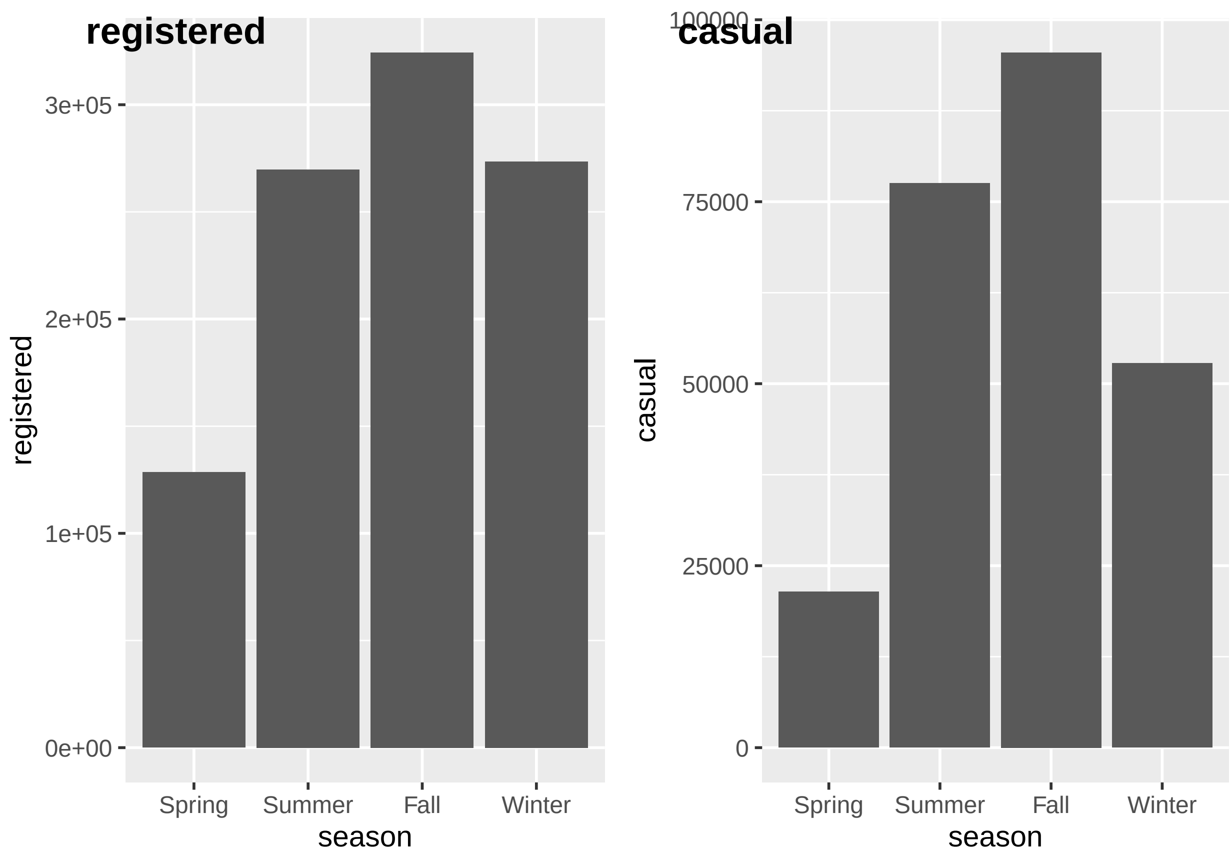
```r
start = "2011-01-01"
bikedata$date_diff <- as.Date(as.character(bikedata$dteday), format="%Y-%m-%d")-
                  as.Date(start, format="%Y-%m-%d")
```

```r
#Seperate training and validating datas base on year

training_d = bikedata %>%  filter(yr == "2011")
set.seed(42)
#partitiontraining <- createDataPartition(y = train$cnt, p = 0.8, list = F)
#training_d <- train[partitiontraining, ]
#test_d <- train[-partitiontraining, ]
validate_d <- bikedata %>% filter(yr == "2012")
```

```r
training.workingday = filter(training_d, workingday == "WorkingDay")
training.nworkingday = filter(training_d, workingday == "Not WorkingDay")
validate.workingday = filter(validate_d, workingday == "WorkingDay")
validate.nworkingday = filter(validate_d, workingday == "Not WorkingDay")
```

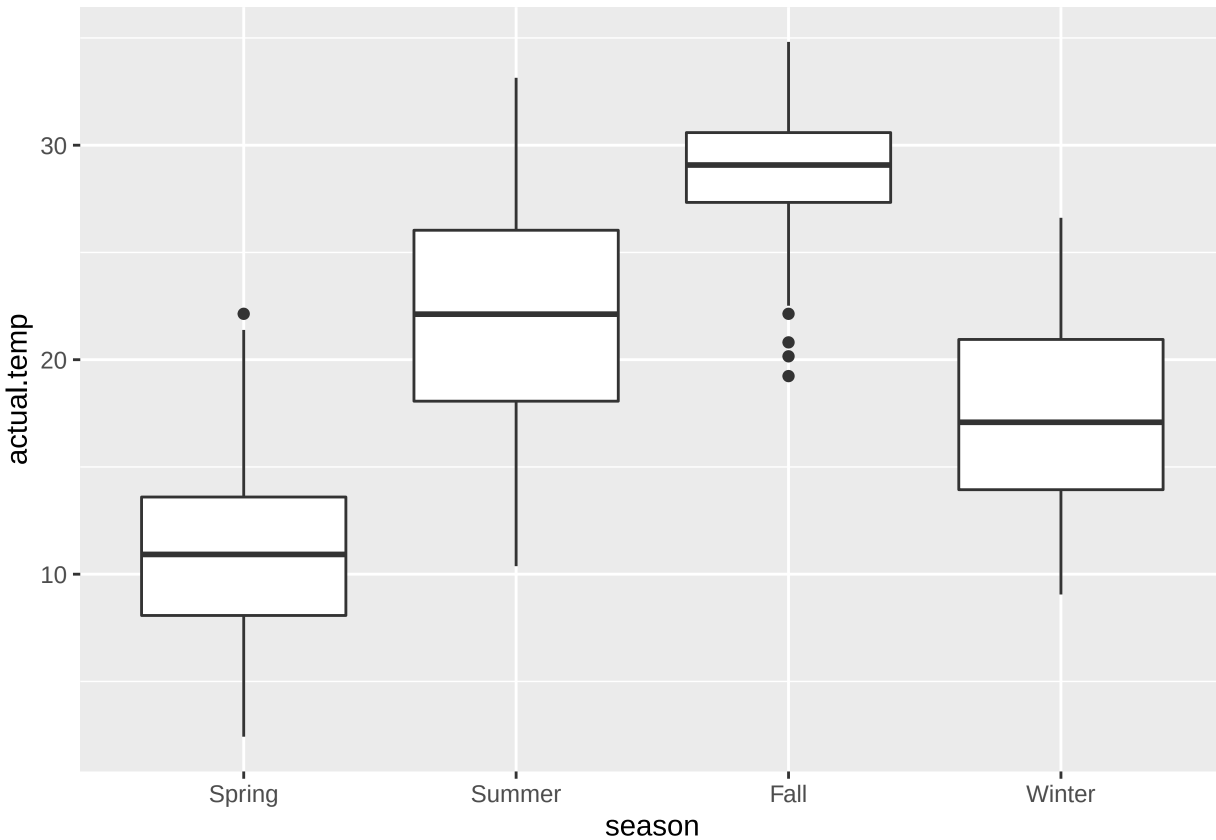
Season

```r
plot1<- ggplot(training_d,aes(x=season,y=registered))+geom_col()
plot2<- ggplot(training_d,aes(x=season,y=casual ))+geom_col()
plot_grid(plot1, plot2, labels = c("registered", "casual"))
```



The graphs show that for both casual and registered bikers, there are the most rental counts during autumn season and the least during the spring season. However, for registered, there are about the same amount of count during summer and winter while for casual there are significantly less counts during winter than during summer. Therefore we think that we should fit different models for registered and casual.

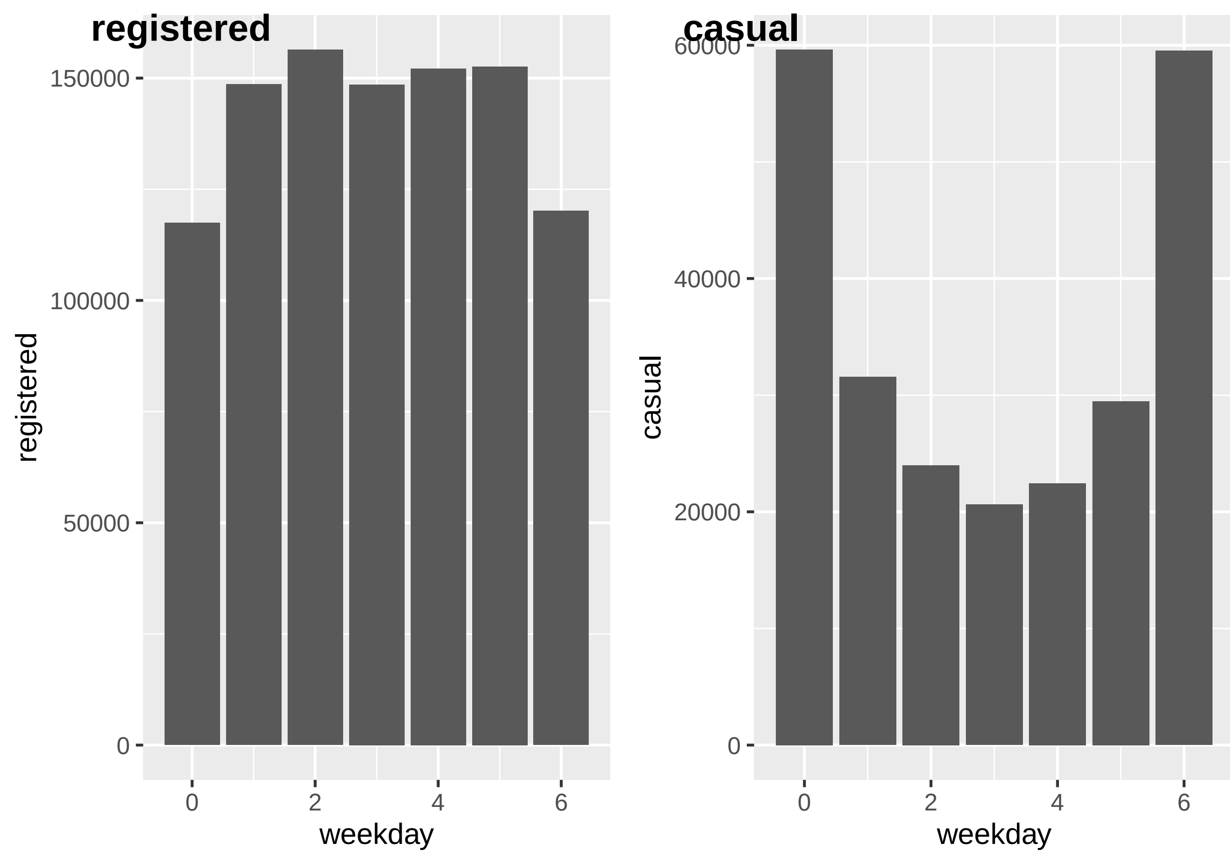
```
ggplot(training_d,aes(x=season,y=actual.temp))+geom_boxplot()
```



Temperature and seasons are strongly correlated. Spring has the lowest temperature while fall has the highest temperature.

Holiday, Weekday, Workingday

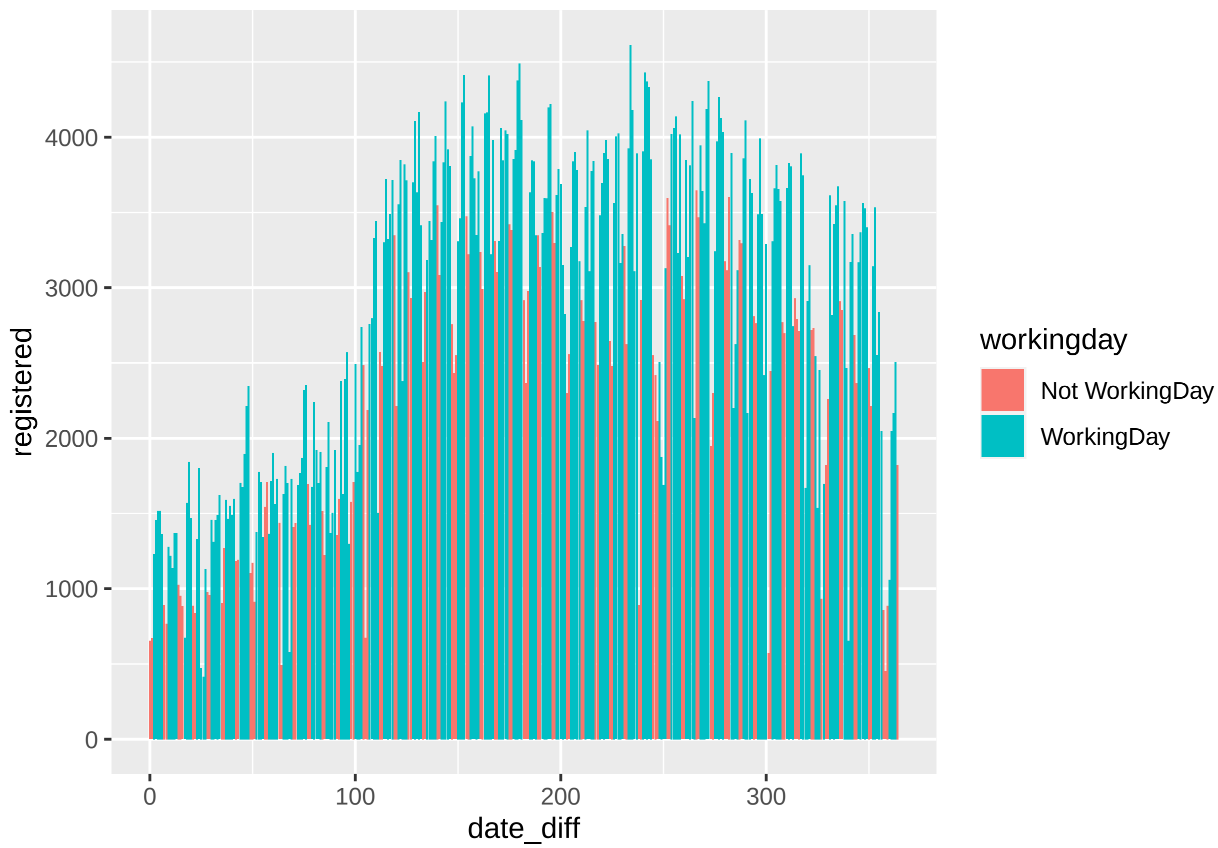
```
plot1<- ggplot(training_d,aes(x=weekday,y=registered))+geom_col()
plot2 <- ggplot(training_d,aes(x=weekday,y=casual))+geom_col()
plot_grid(plot1, plot2, labels = c("registered", "casual"))
```



Casual rental counts are higher on weekends compared to on weekdays while registered rental counts are lower on weekends than on weekdays.

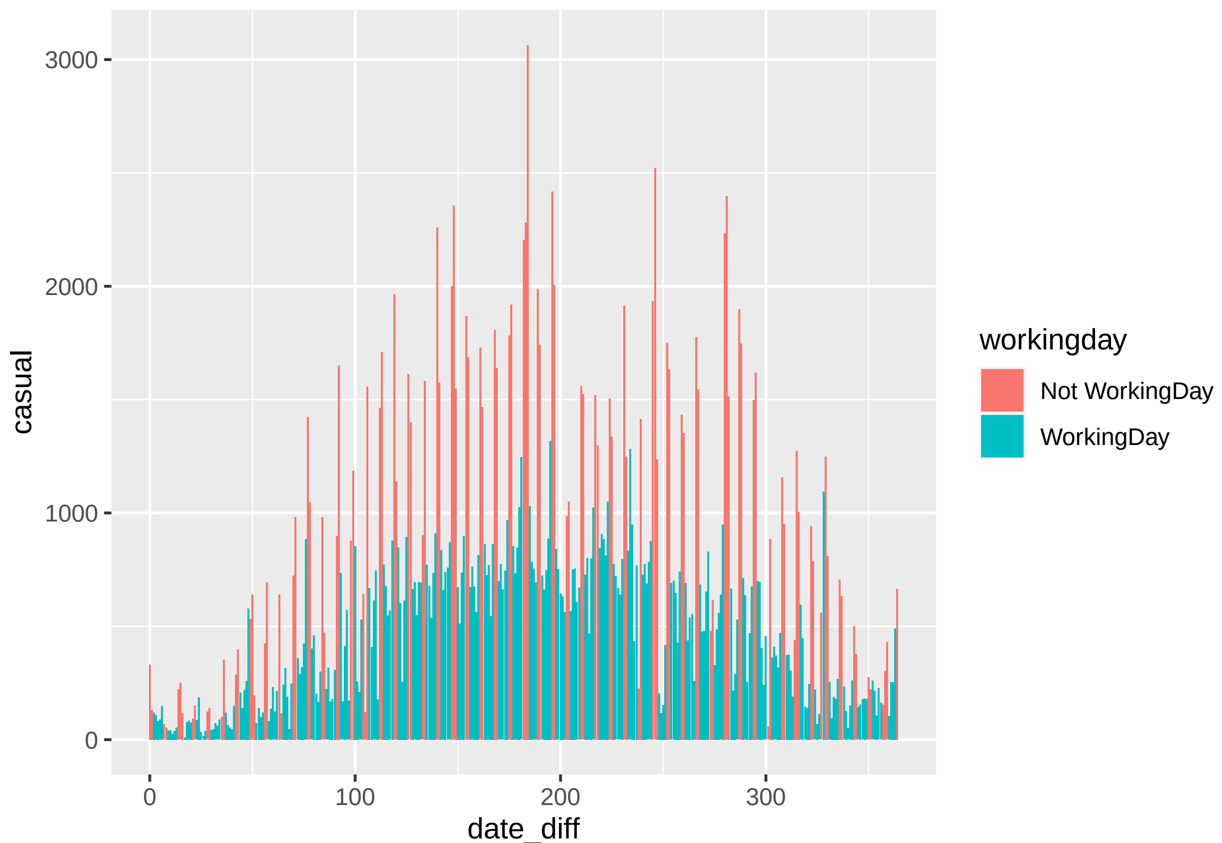
```
plot1 <- ggplot(data = training_d, aes(x=date_diff, y = registered)) +  geom_col(aes(fill = workingday)
plot2 <- ggplot(data = training_d, aes(x=date_diff, y = casual)) +  geom_col(aes(fill = workingday))
plot1
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
plot2
```

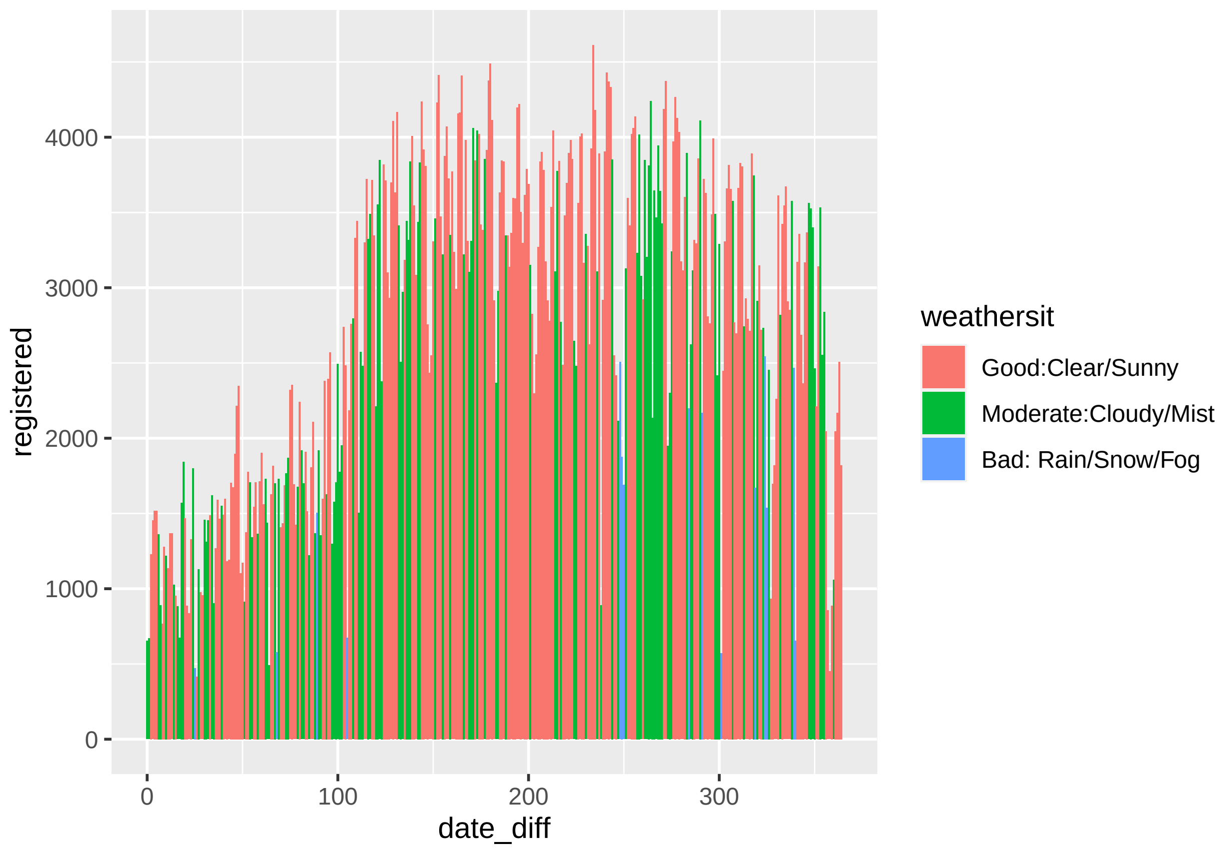
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



There are more rental counts on not working days than on working days for casual bikers while there are more rental registered rental counts on working days than on not workingdays. There are also less rental counts for both registered and casual in the beginning of the year, then we see an increase of bikers during the summer and fall seasons, then a decrease during the end of the year. We suspect that this trend is due to temperature and other weather conditions.

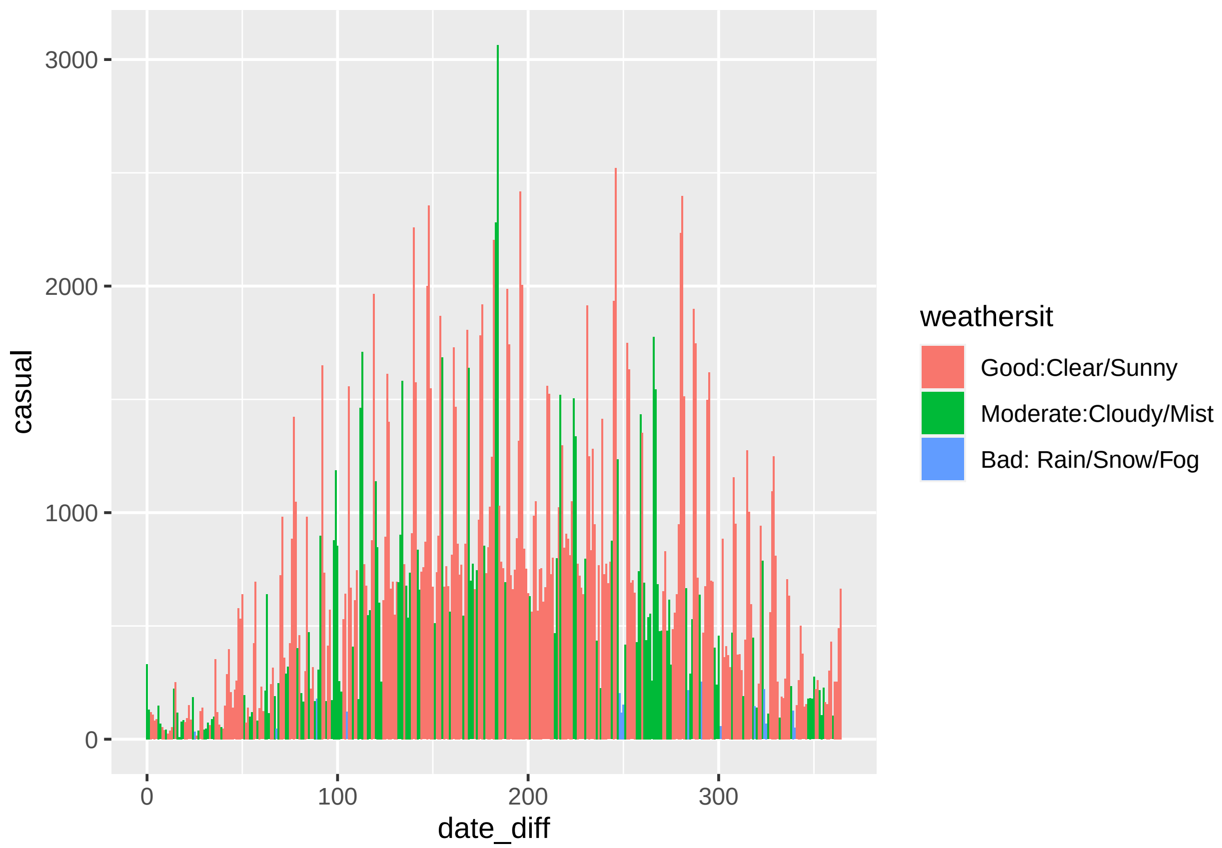
```
plot1 <- ggplot(data = training_d, aes(x=date_diff, y = registered)) +  geom_col(aes(fill = weathersit)
plot2 <- ggplot(data = training_d, aes(x=date_diff, y = casual)) +  geom_col(aes(fill = weathersit))
plot1
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



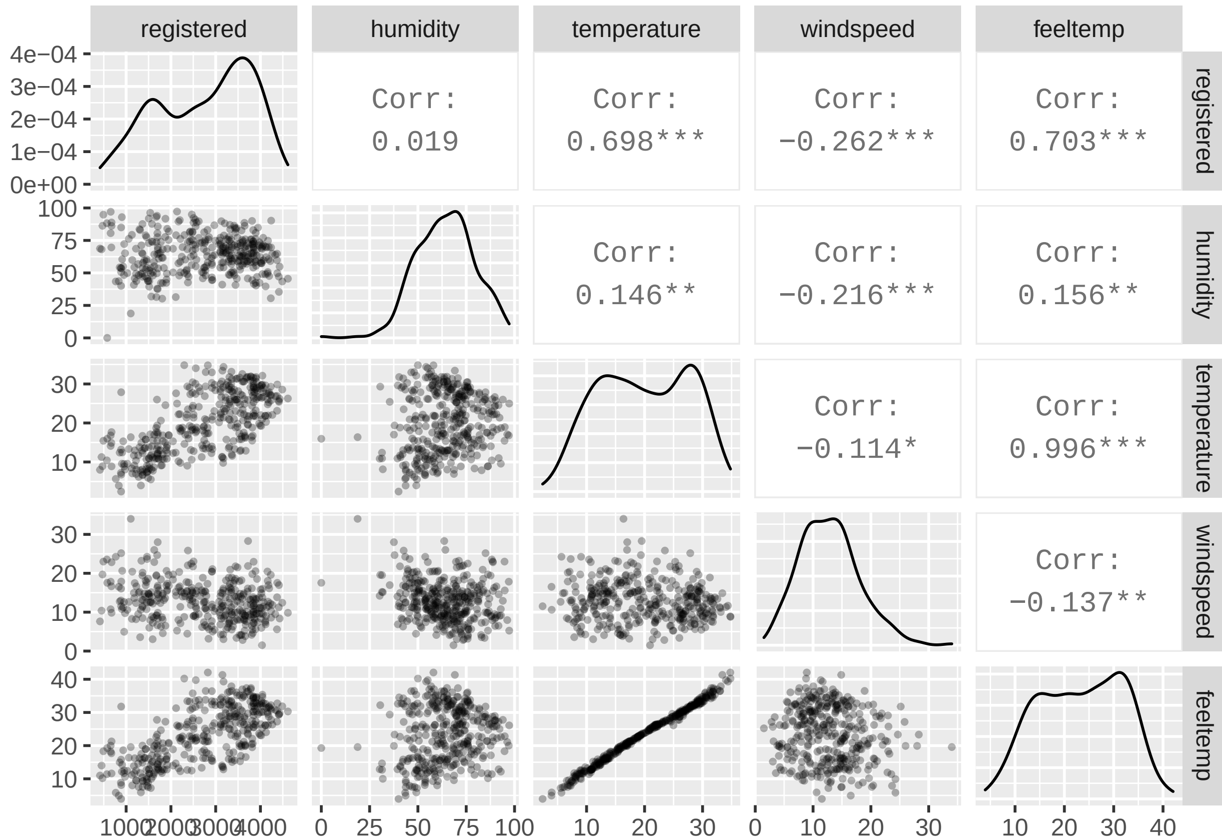

```
plot2
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

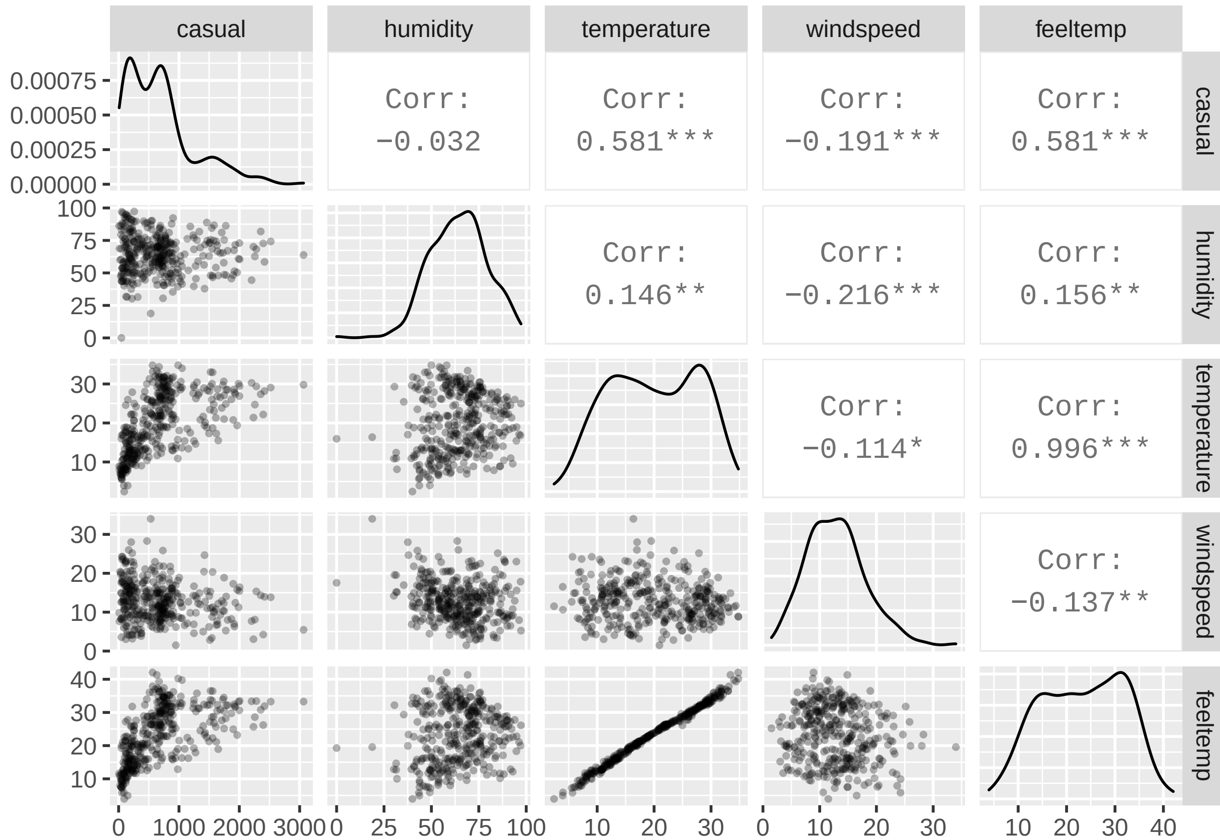


More bikers tend to bike on days with good and moderate weather conditions than on bad weather conditions.

```
data <- data.frame(training_d$registered, training_d$actual.hum, training_d$actual.temp, training_d$actu
data = data%>% rename( registered = training_d.registered, humidity= training_d.actual.hum,  temperatur
plot1 <- ggpairs(data, lower = list(continuous = wrap("points", alpha = 0.3, size= 0.7)))
data <- data.frame(training_d$casual, training_d$actual.hum, training_d$actual.temp, training_d$actual.
data = data%>% rename( casual = training_d.casual, humidity= training_d.actual.hum,  temperature= train
plot2 <- ggpairs(data, lower = list(continuous = wrap("points", alpha = 0.3, size= 0.7)))
plot1
```

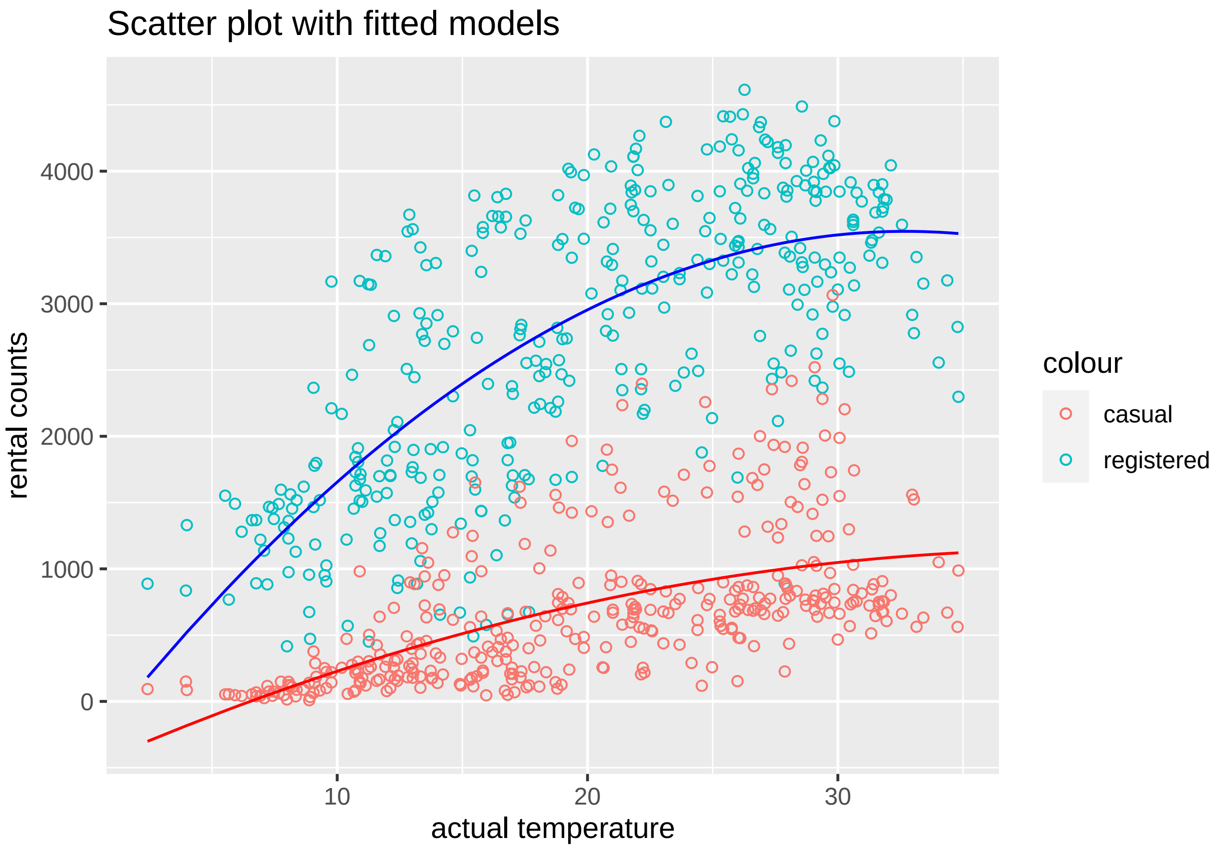

registered
humidity
temperature
windspeed
feeltemp
Corr:
0.019
Corr:
0.698***
Corr:
−0.262***
Corr:
0.703***
Corr:
0.146**
Corr:
−0.216***
Corr:
0.156**
Corr:
−0.114*
Corr:
0.996***
Corr:
−0.137**

```
plot2
```

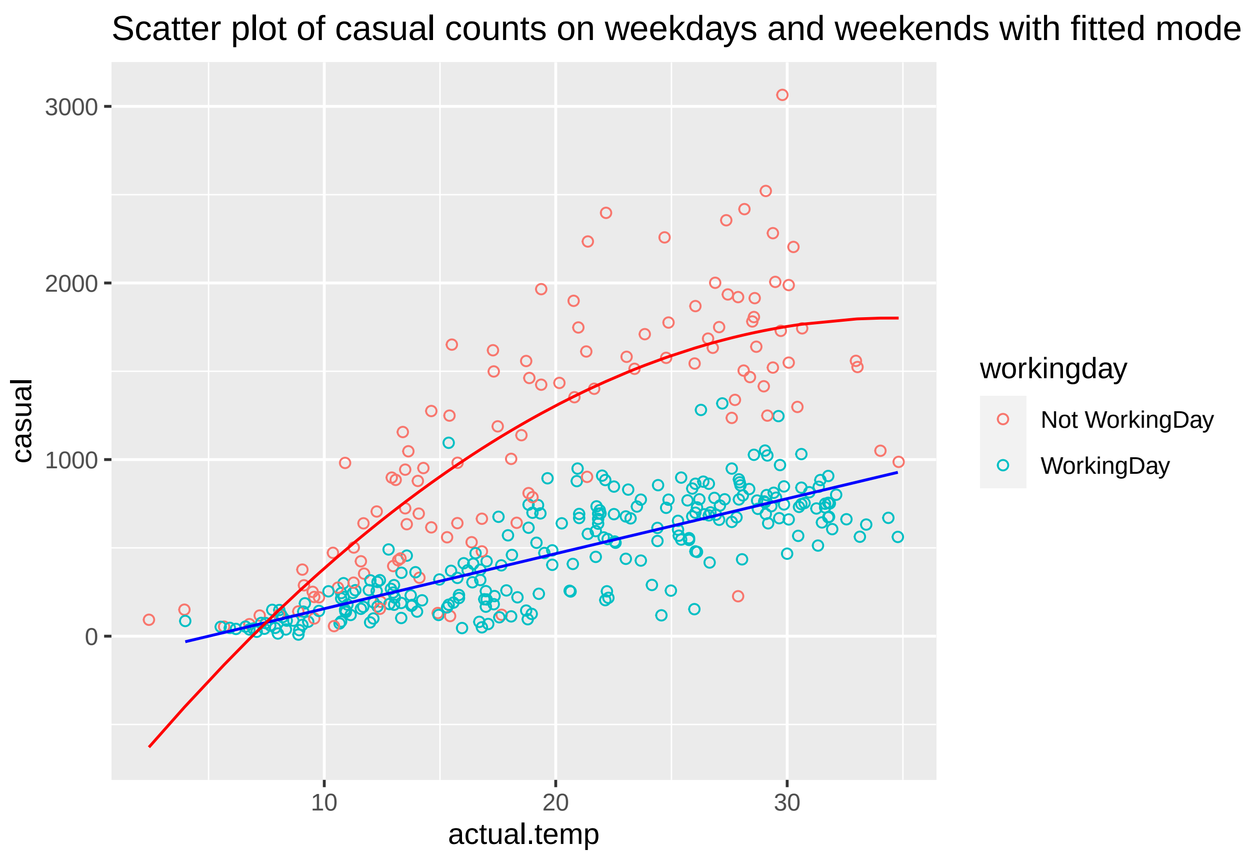


The graphs suggest that for both registered and casual bikers, there is a high correlation between temperature, windspeed and rental counts. There is strong correlation between temperature and feel temperature, so we decided to omit feel temperature to avoid collinearity.

```
m.quadls_casual <- lm(training_d$casual ~ training_d$actual.temp + I(training_d$actual.temp^2))
m.quadls_registered <- lm(training_d$registered ~ training_d$actual.temp + I(training_d$actual.temp^2))

ggplot(training_d, aes(x = actual.temp)) + geom_point(aes(y = registered,  color = "registered"), shape
```



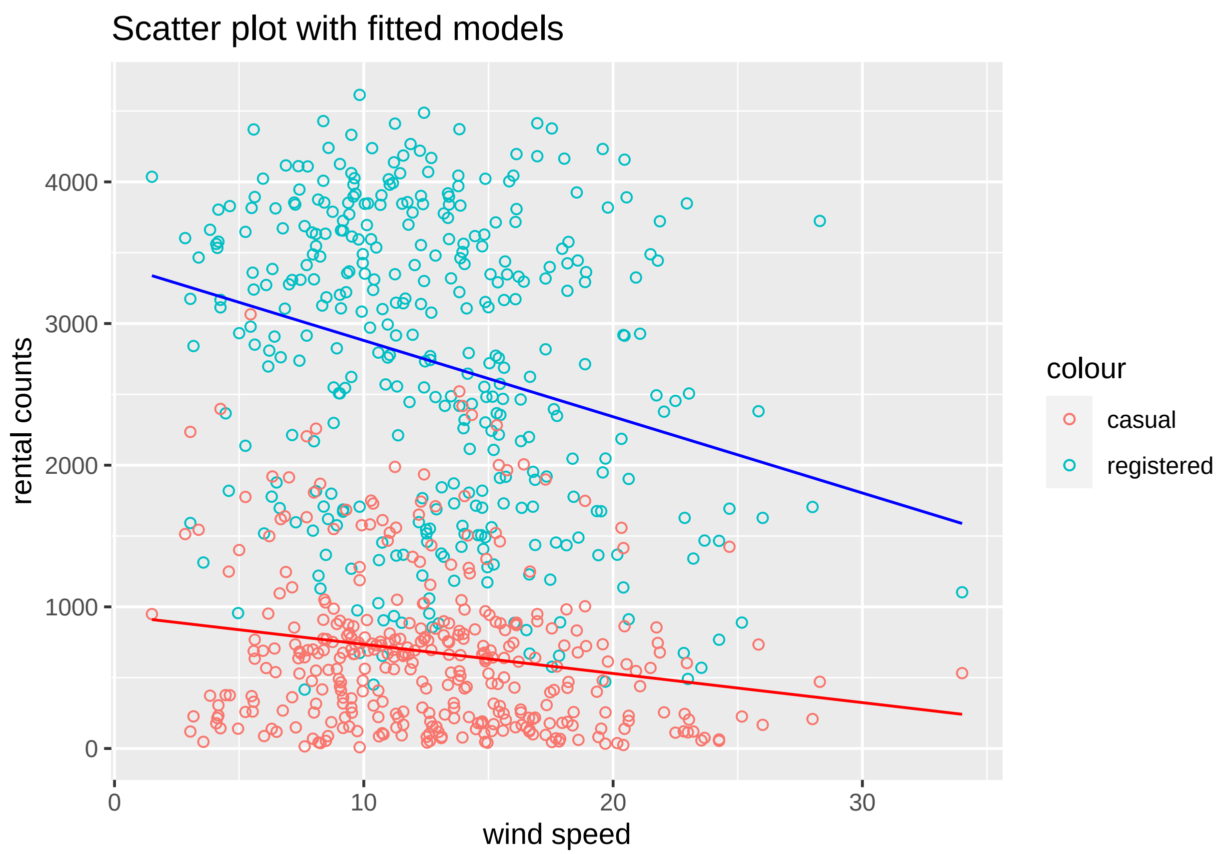

```
m.casual.workingday <- lm(training.workingday$casual ~ training.workingday$actual.temp)

m.quadls_casual.nworkingday <- lm(training.nworkingday$casual ~ training.nworkingday$actual.temp + I(tra

m.registered.nworkingday <- lm(training.nworkingday$registered ~ training.nworkingday$actual.temp + I(ti
ggplot(training_d, aes(x = actual.temp)) + geom_point(aes(y = casual,  color = workingday), shape = 1) +
```




```
m.registered.workingday <- lm(training.workingday$registered ~ training.workingday$actual.temp + I(trai

m.registered.nworkingday <- lm(training.nworkingday$registered ~ training.nworkingday$actual.temp + I(t
ggplot(training_d, aes(x = actual.temp)) + geom_point(aes(y = registered,  color = workingday), shape =
```

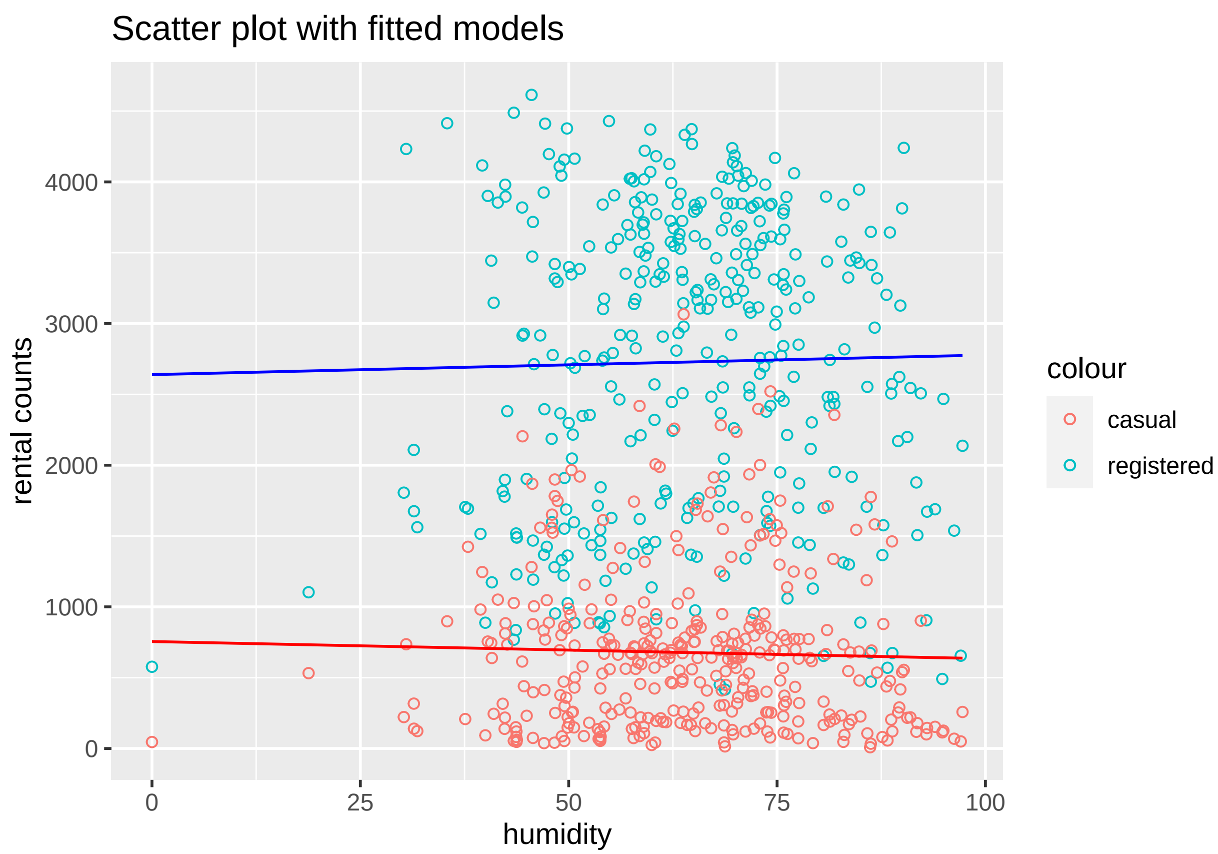
Wind speed and rental counts

```
m.lin_casual <- lm(training_d$casual ~ training_d$actual.windspeed)
m.lin_registered <- lm(training_d$registered ~ training_d$actual.windspeed)

ggplot(training_d, aes(x = actual.windspeed)) + geom_point(aes(y = registered,  color = "registered"), s
```

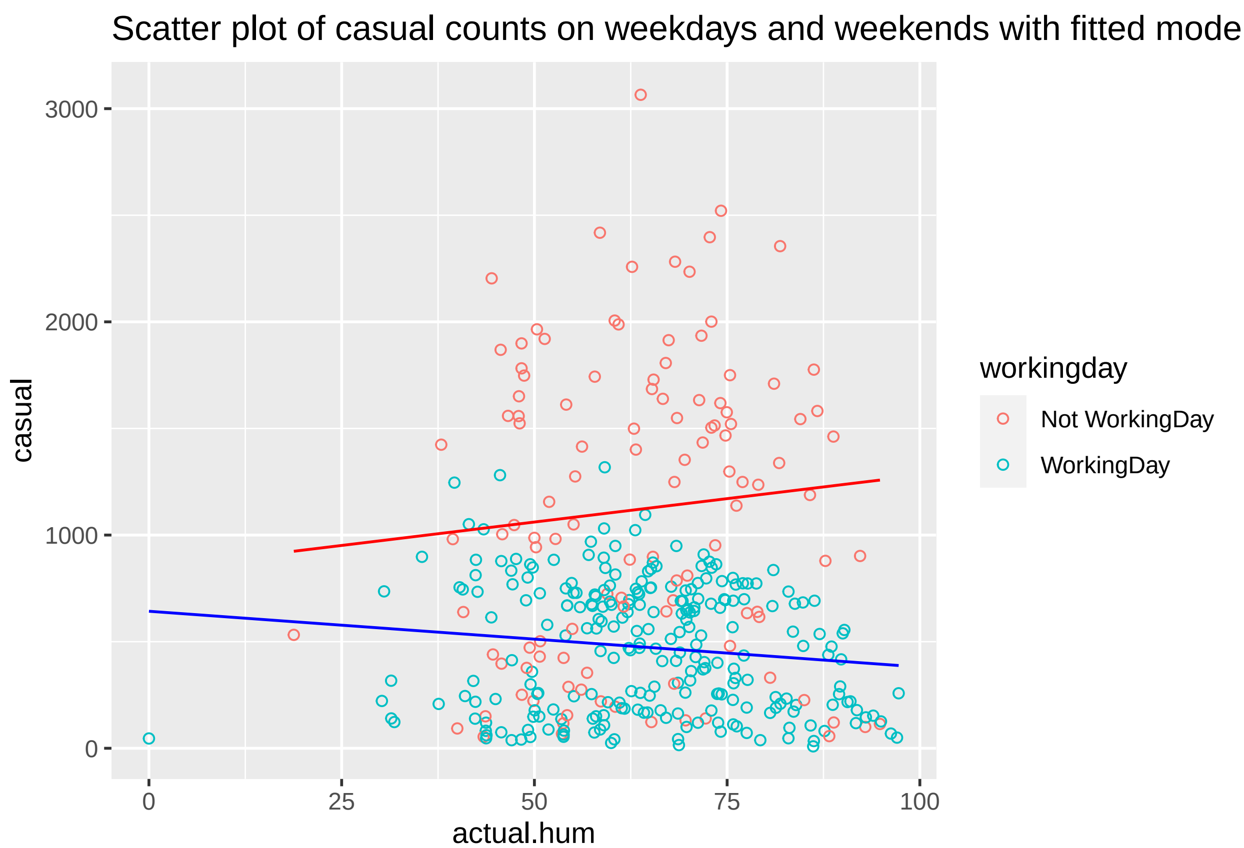


Humidity

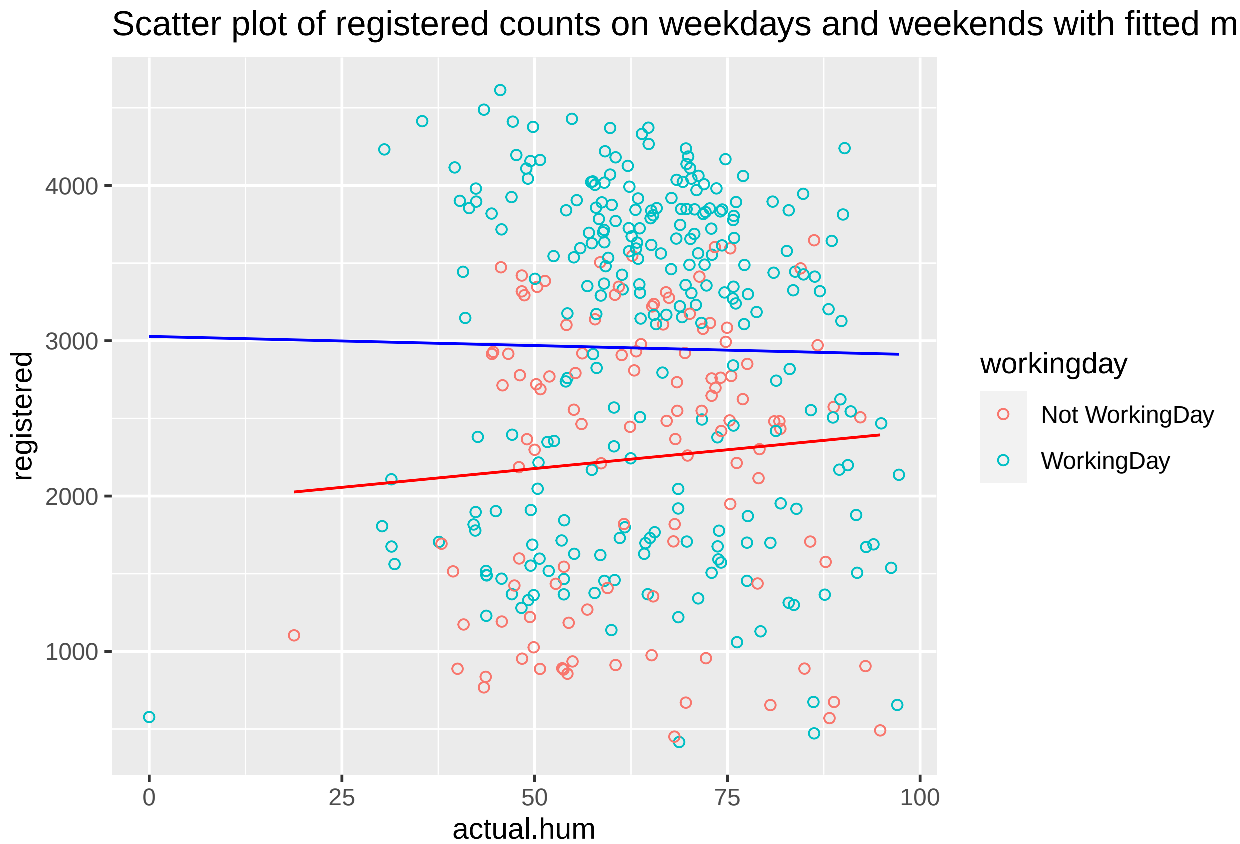
```
m.lin_casual <- lm(training_d$casual ~ training_d$actual.hum)
m.lin_registered <- lm(training_d$registered ~ training_d$actual.hum)

ggplot(training_d, aes(x = actual.hum)) + geom_point(aes(y = registered,  color = "registered"), shape =
```




```
m.hum_casual.workingday <- lm(training.workingday$casual ~ training.workingday$actual.hum)

m.hum_casual.nworkingday <- lm(training.nworkingday$casual ~ training.nworkingday$actual.hum)

ggplot(training_d, aes(x = actual.hum)) + geom_point(aes(y = casual,  color = workingday), shape = 1) +
```




```
m.hum_registered.workingday <- lm(training.workingday$registered ~ training.workingday$actual.hum)

m.hum_registered.nworkingday <- lm(training.nworkingday$registered ~ training.nworkingday$actual.hum)

ggplot(training_d, aes(x = actual.hum)) + geom_point(aes(y = registered,  color = workingday), shape = 
```



Model

```
model.casual.workingday <- lm(casual ~actual.windspeed + actual.temp +I(actual.temp^2) + weathersit, da

model.registered.workingday <- lm(registered ~   actual.temp + I(actual.temp^2)+actual.windspeed + weat
```

```
summary(model.casual.workingday)
```

```
##
## Call:
## lm(formula = casual ~ actual.windspeed + actual.temp + I(actual.temp^2) +
##     weathersit, data = training.workingday)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -369.62  -95.15  -20.24   70.12  636.87
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -316.7199    67.7510  -4.675 4.87e-06 ***
## actual.windspeed               -2.8500     1.9268  -1.479     0.14
## actual.temp                    65.6497     7.2837   9.013  < 2e-16 ***
```

```
## I(actual.temp^2)                 -0.9122     0.1828  -4.990 1.15e-06 ***
## weathersitModerate:Cloudy/Mist -144.9753    21.0026  -6.903 4.38e-11 ***
## weathersitBad: Rain/Snow/Fog   -390.4630    45.1814  -8.642 7.39e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 152.7 on 244 degrees of freedom
## Multiple R-squared:  0.7506, Adjusted R-squared:  0.7455
## F-statistic: 146.9 on 5 and 244 DF,  p-value: < 2.2e-16
```

```
summary(model.registered.workingday)
```

```
##
## Call:
## lm(formula = registered ~ actual.temp + I(actual.temp^2) + actual.windspeed +
##     weathersit + date_diff, data = training.workingday)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1760.06  -291.00    26.68   344.05  1084.55
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      50.2880   219.6676   0.229   0.8191
## actual.temp                     221.4958    26.4657   8.369 4.60e-15 ***
## I(actual.temp^2)                 -3.5440     0.6492  -5.459 1.18e-07 ***
## actual.windspeed                -16.6444     6.5696  -2.534   0.0119 *
## weathersitModerate:Cloudy/Mist  -389.8799    67.7165  -5.758 2.56e-08 ***
## weathersitBad: Rain/Snow/Fog   -1651.5825   146.5403 -11.270  < 2e-16 ***
## date_diff                         2.8943     0.3639   7.953 6.90e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 491.7 on 243 degrees of freedom
## Multiple R-squared:  0.7839, Adjusted R-squared:  0.7786
## F-statistic: 146.9 on 6 and 243 DF,  p-value: < 2.2e-16
```

```
model.casual.nworkingday <- lm(casual ~ actual.windspeed + actual.temp +I(actual.temp^2) + weathersit, d

model.registered.nworkingday <- lm(registered ~ actual.temp +  actual.windspeed + weathersit, data = tra
```

```
summary(model.casual.nworkingday)
```

```
##
## Call:
## lm(formula = casual ~ actual.windspeed + actual.temp + I(actual.temp^2) +
##     weathersit, data = training.nworkingday)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -998.6 -272.4  -51.1  258.2 1329.4
##
```

```
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -649.3944   253.0927  -2.566   0.0117 *
## actual.windspeed               -24.2122     7.2201  -3.353   0.0011 **
## actual.temp                    170.4207    25.8870   6.583 1.67e-09 ***
## I(actual.temp^2)                -2.6486     0.6468  -4.095 8.13e-05 ***
## weathersitModerate:Cloudy/Mist -209.2066    80.9685  -2.584   0.0111 *
## weathersitBad: Rain/Snow/Fog   -536.2792   291.3608  -1.841   0.0684 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 391.3 on 109 degrees of freedom
## Multiple R-squared:  0.7062, Adjusted R-squared:  0.6927
## F-statistic:  52.4 on 5 and 109 DF,  p-value: < 2.2e-16
```

```r
summary(model.registered.nworkingday)
```

```
##
## Call:
## lm(formula = registered ~ actual.temp + actual.windspeed + weathersit,
##     data = training.nworkingday)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1402.68  -413.18   -12.15   435.30  1424.07
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1509.210    225.096   6.705 9.02e-10 ***
## actual.temp                     73.602      7.372   9.984  < 2e-16 ***
## actual.windspeed               -46.649     11.182  -4.172 6.06e-05 ***
## weathersitModerate:Cloudy/Mist -293.689    124.600  -2.357   0.0202 *
## weathersitBad: Rain/Snow/Fog   -839.067    450.831  -1.861   0.0654 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 606.1 on 110 degrees of freedom
## Multiple R-squared:  0.5861, Adjusted R-squared:  0.571
## F-statistic: 38.94 on 4 and 110 DF,  p-value: < 2.2e-16
```

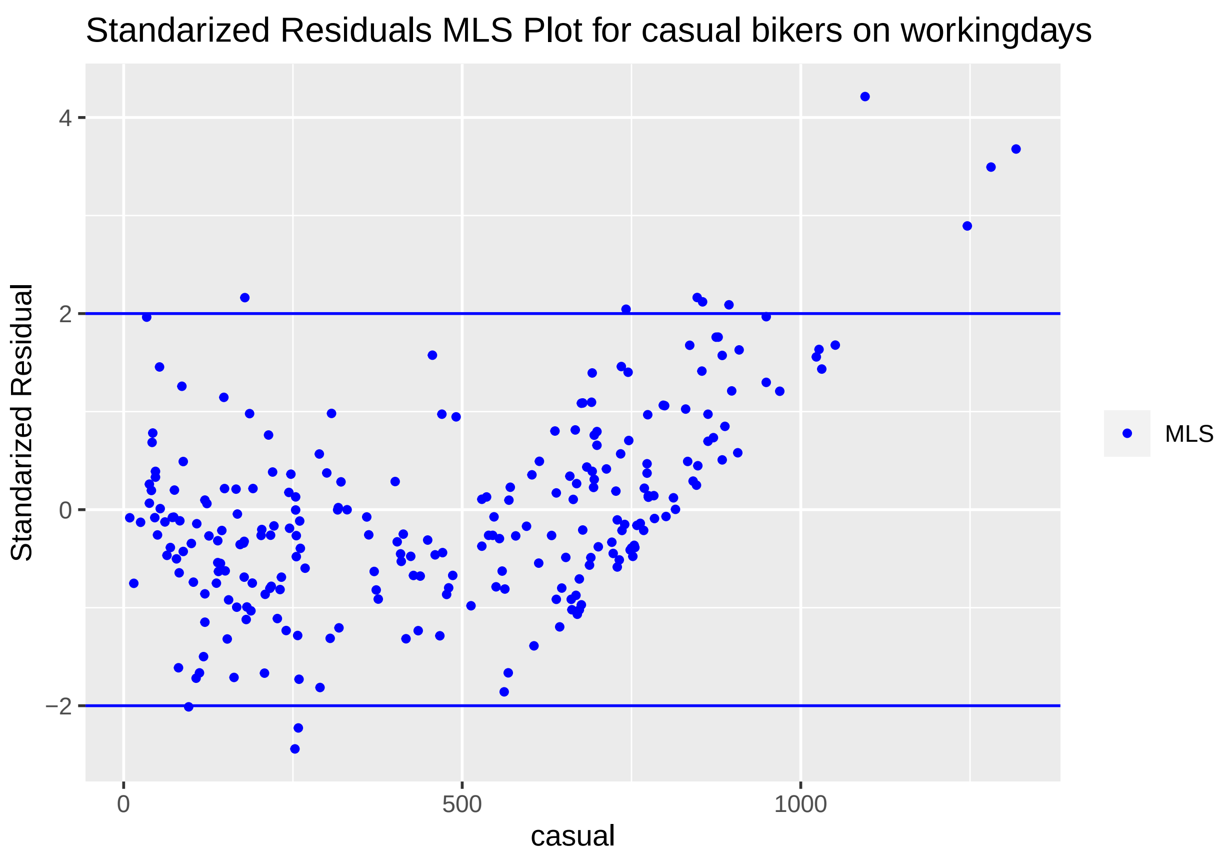
All of the p values on the coefficients of the regressors are less than 0.005. Therefore we are confident that all the regressors have an effect on the rental counts individually. Furthermore, the p value of the F-statistic is less than 0.005. Therefore we are very confident that all the regressors are jointly significant. The R^2 value is arount 0.7, so the models explain around 70 percent of the variation in rental counts. (explain more in paper).

Model diagnosis

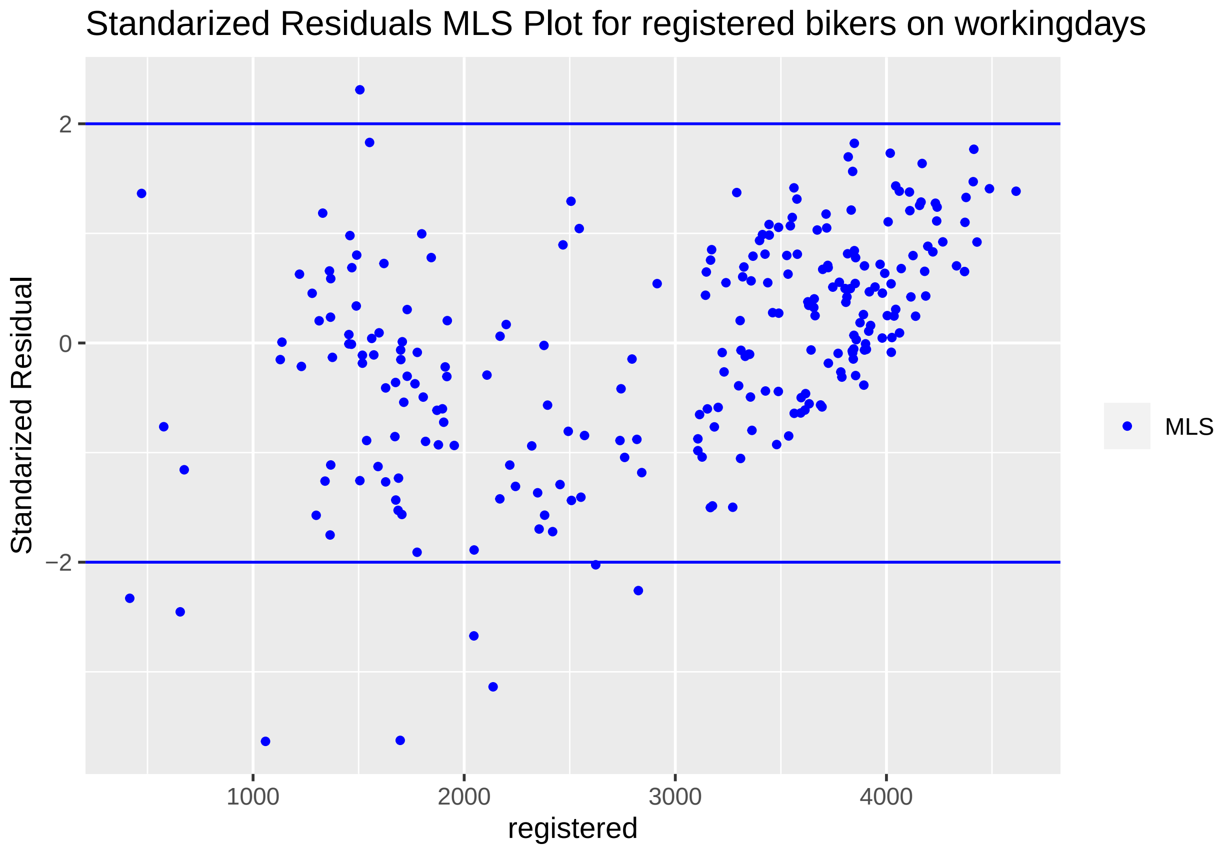
```r
StanRes.casual.workingday <- rstandard(model.casual.workingday)
StanRes.registered.workingday <- rstandard(model.registered.workingday)
```

```
StanRes.casual.nworkingday <- rstandard(model.casual.nworkingday)
StanRes.registered.nworkingday <- rstandard(model.registered.nworkingday)
```

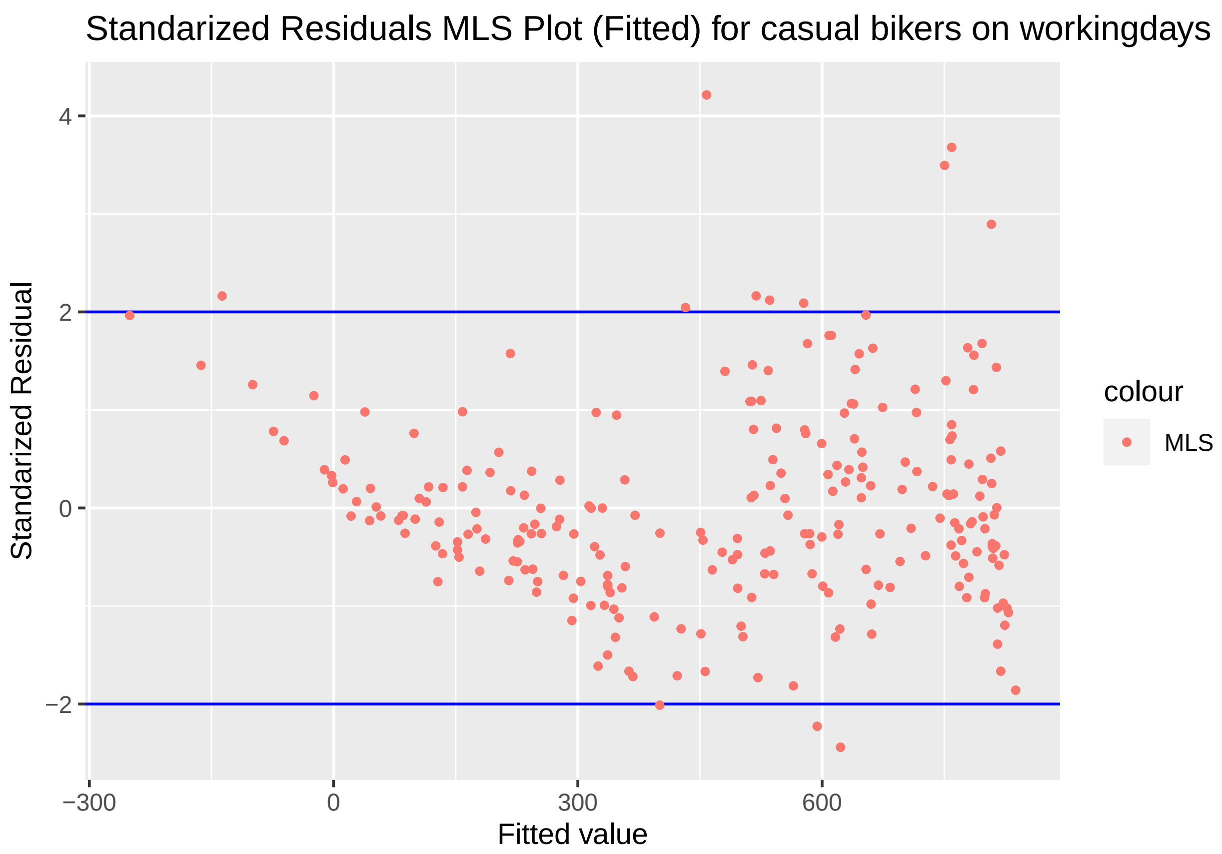
```
ggplot() +
geom_point(data=training.workingday, aes(x=casual, y=StanRes.casual.workingday, color = "MLS"), size = 1
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("blue")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for casual bikers on working
```



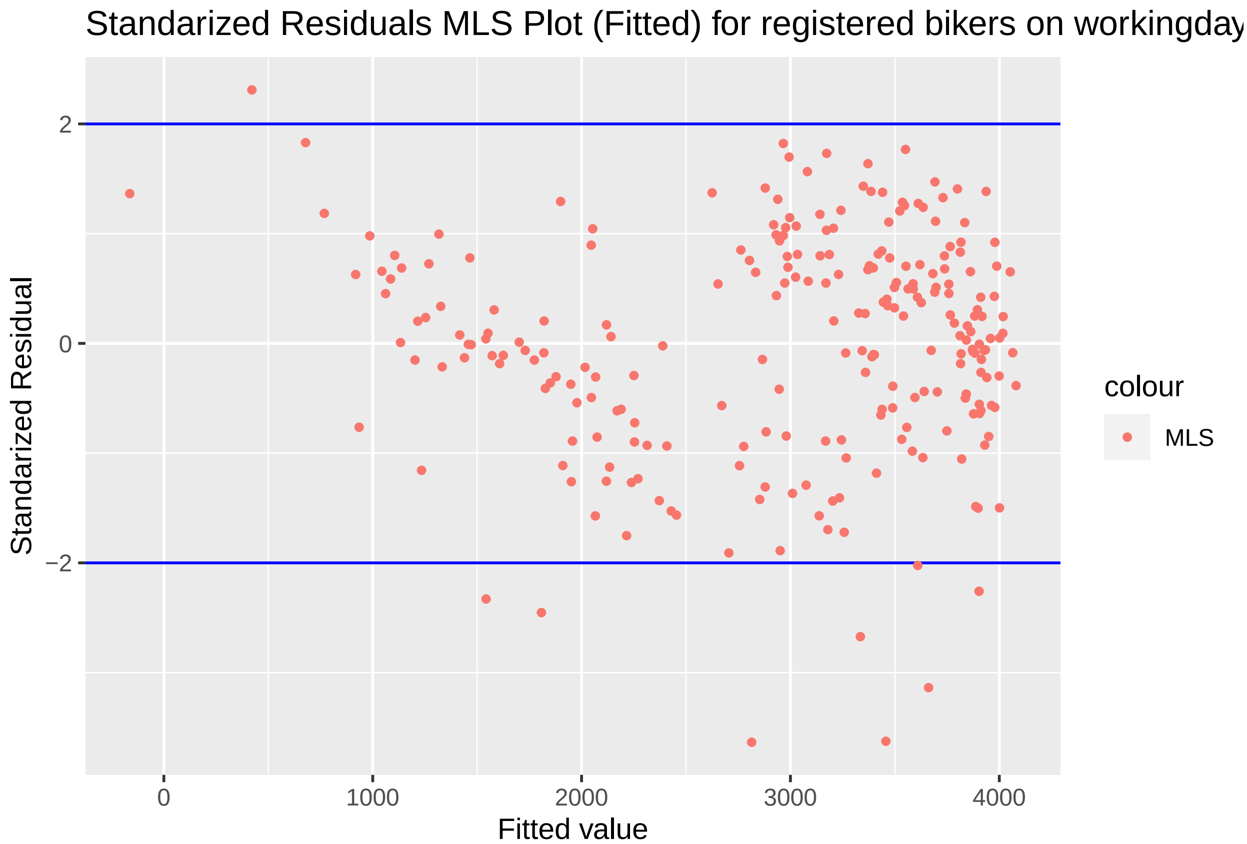

```
ggplot() +
geom_point(data=training.workingday, aes(x=registered, y=StanRes.registered.workingday, color = "MLS"),
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("blue")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for registered bikers on work
```



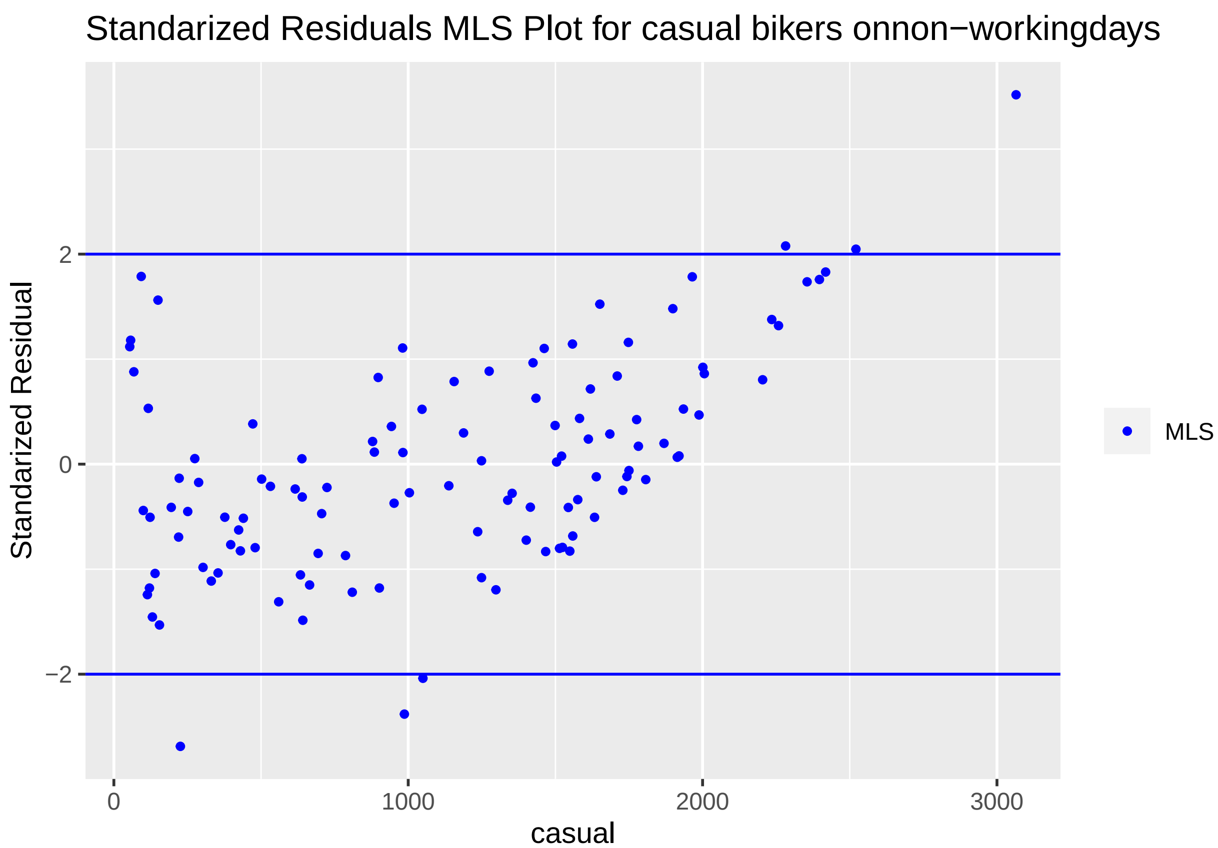

```
Fitted_casual.workingday = fitted(model.casual.workingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitte
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals MLS Plot (Fitted) for casual bikers on workingdays")
```



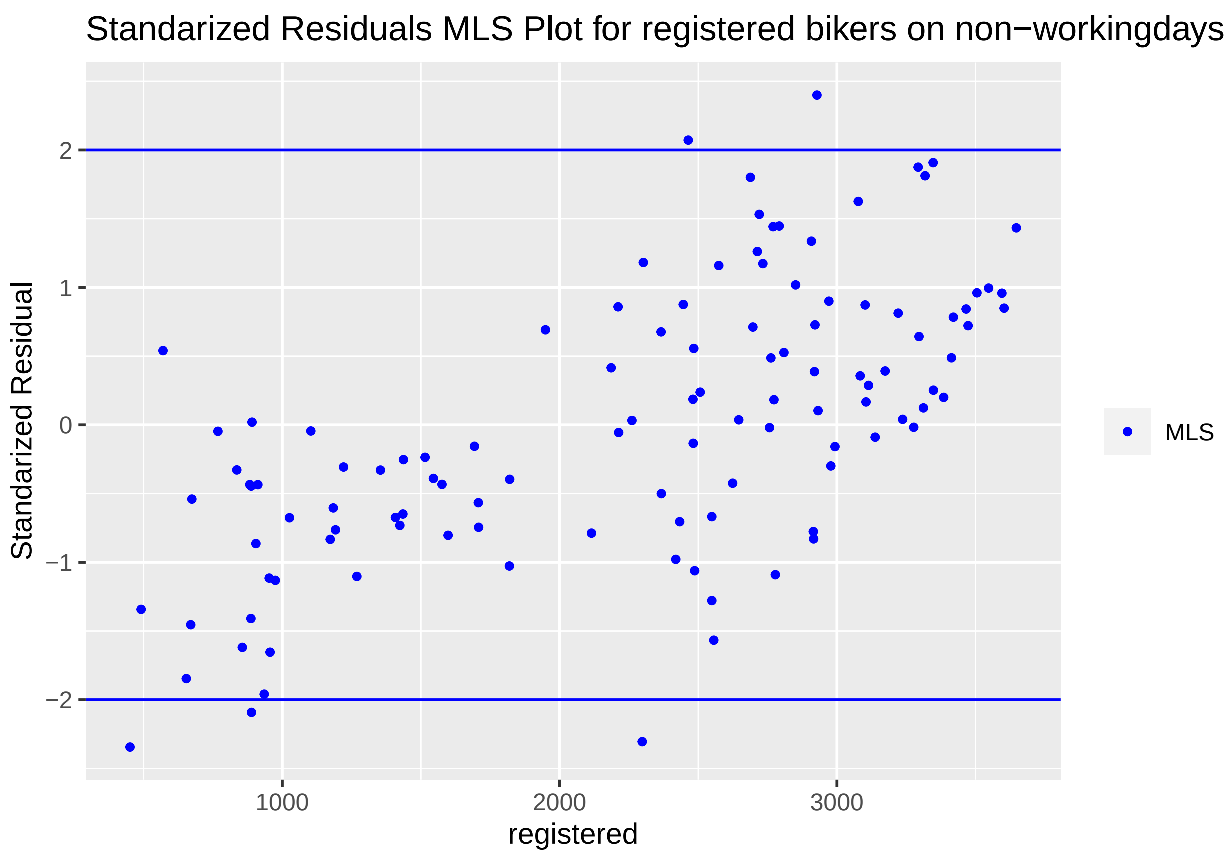

```r
Fitted_registered.workingday = fitted(model.registered.workingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitt
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals MLS Plot (Fitted) for registered bikers on workingdays")
```



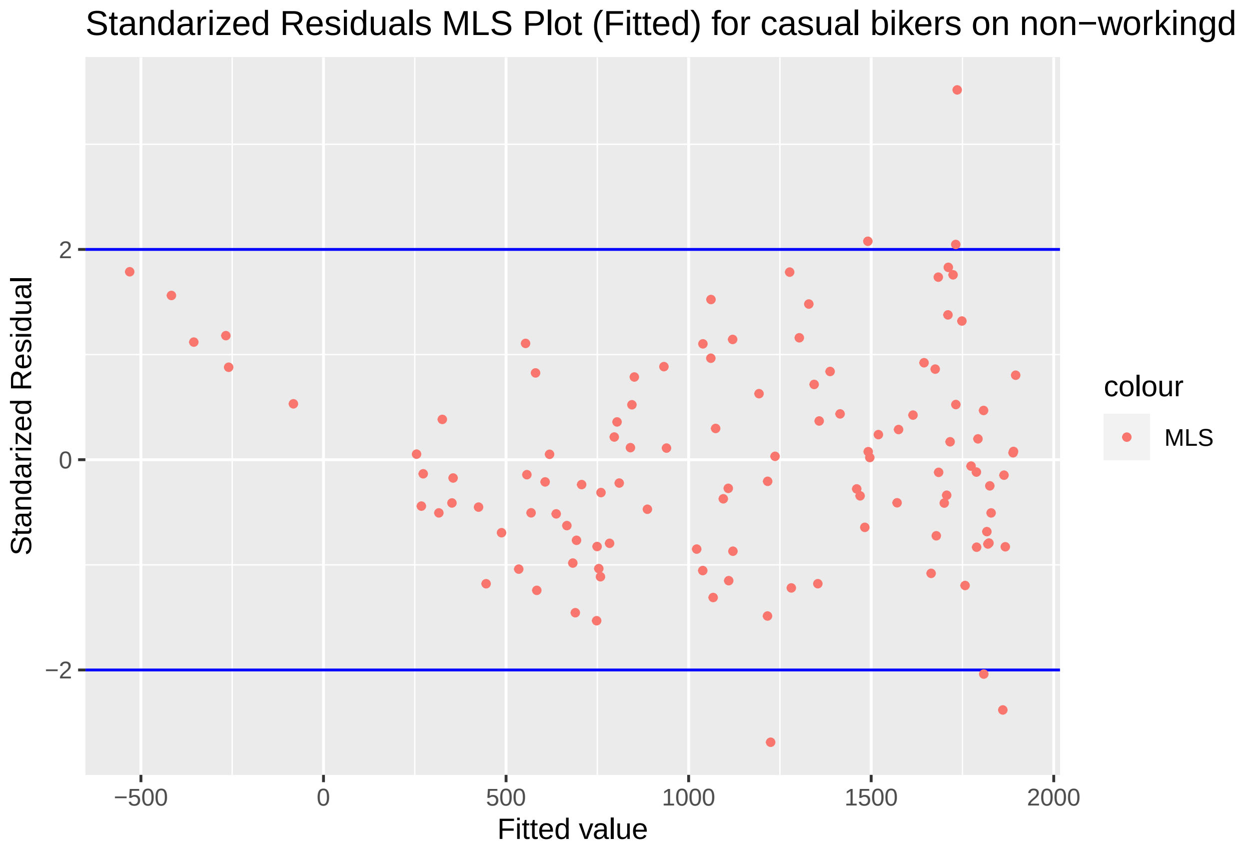

```
ggplot() +
geom_point(data=training.nworkingday, aes(x=casual, y=StanRes.casual.nworkingday, color = "MLS"), size =
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("blue")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for casual bikers onnon-worki
```



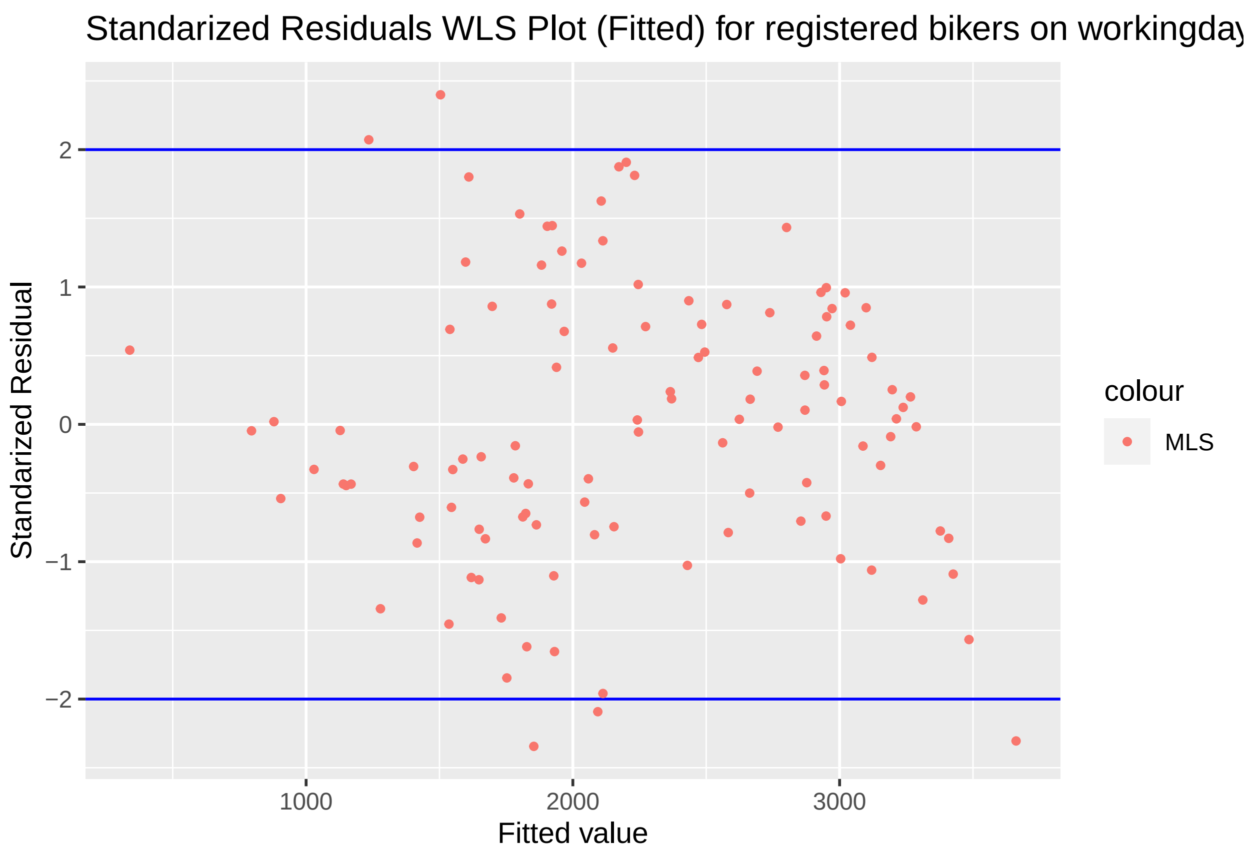

```
ggplot() +
geom_point(data=training.nworkingday, aes(x=registered, y=StanRes.registered.nworkingday, color = "MLS"
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("blue")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for registered bikers on non
```



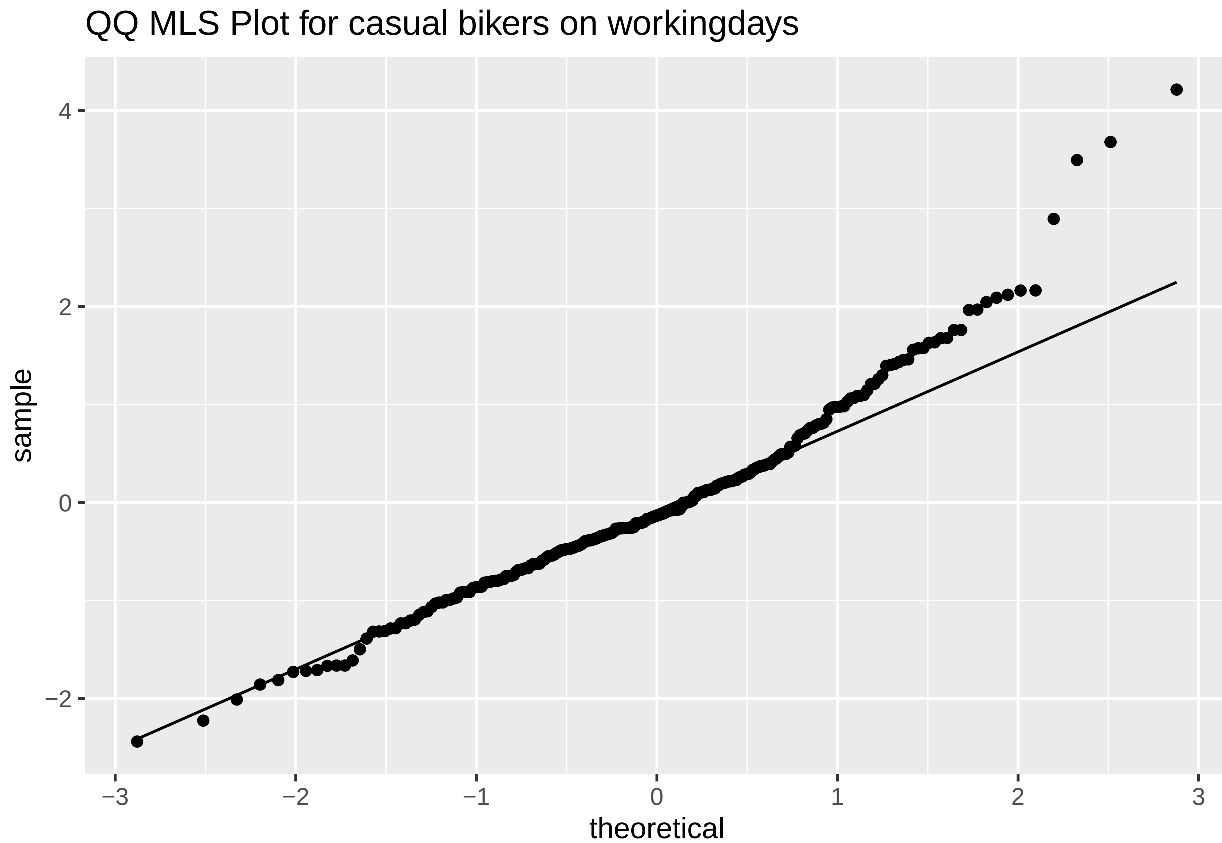

```
Fitted_casual.nworkingday = fitted(model.casual.nworkingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitte
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals MLS Plot (Fitted) for casual bikers on non-workingdays")
```



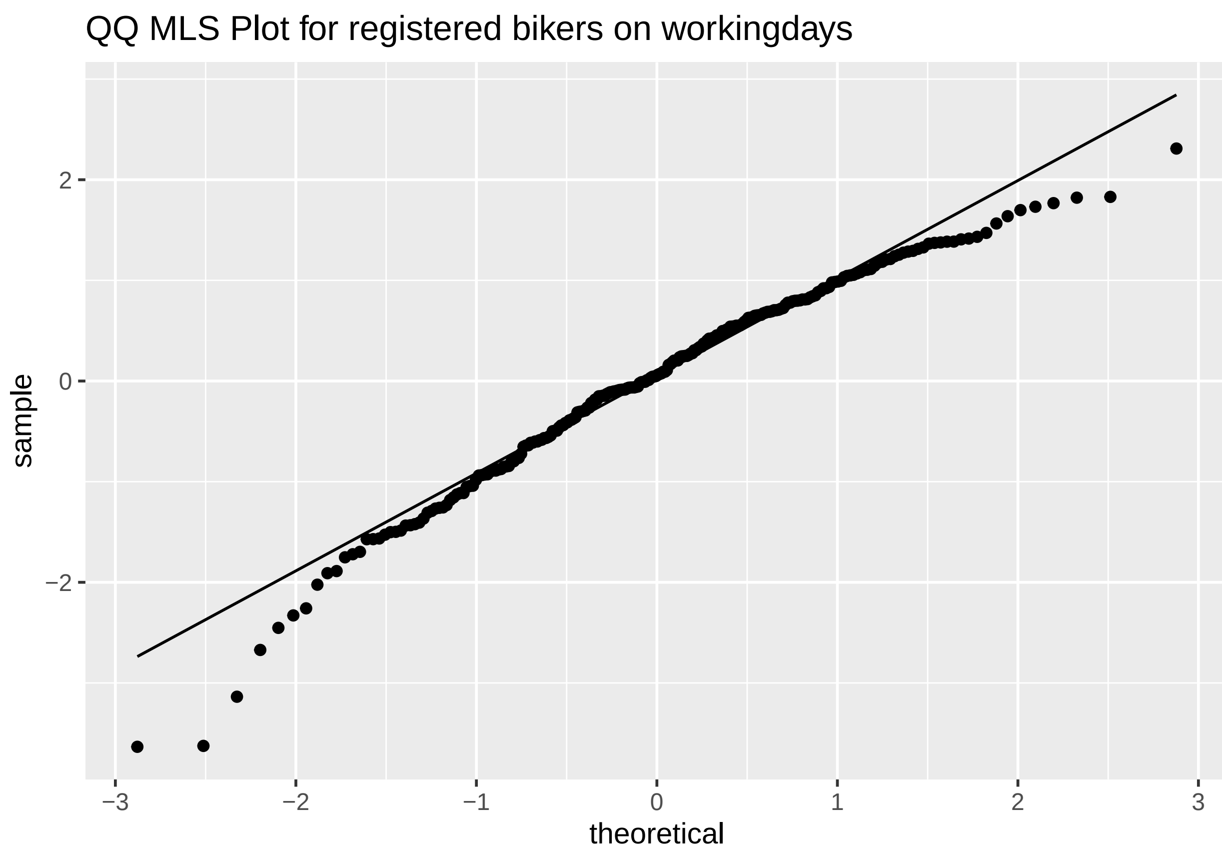

```
Fitted_registered.nworkingday = fitted(model.registered.nworkingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitte
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals WLS Plot (Fitted) for registered bikers on workingdays")
```



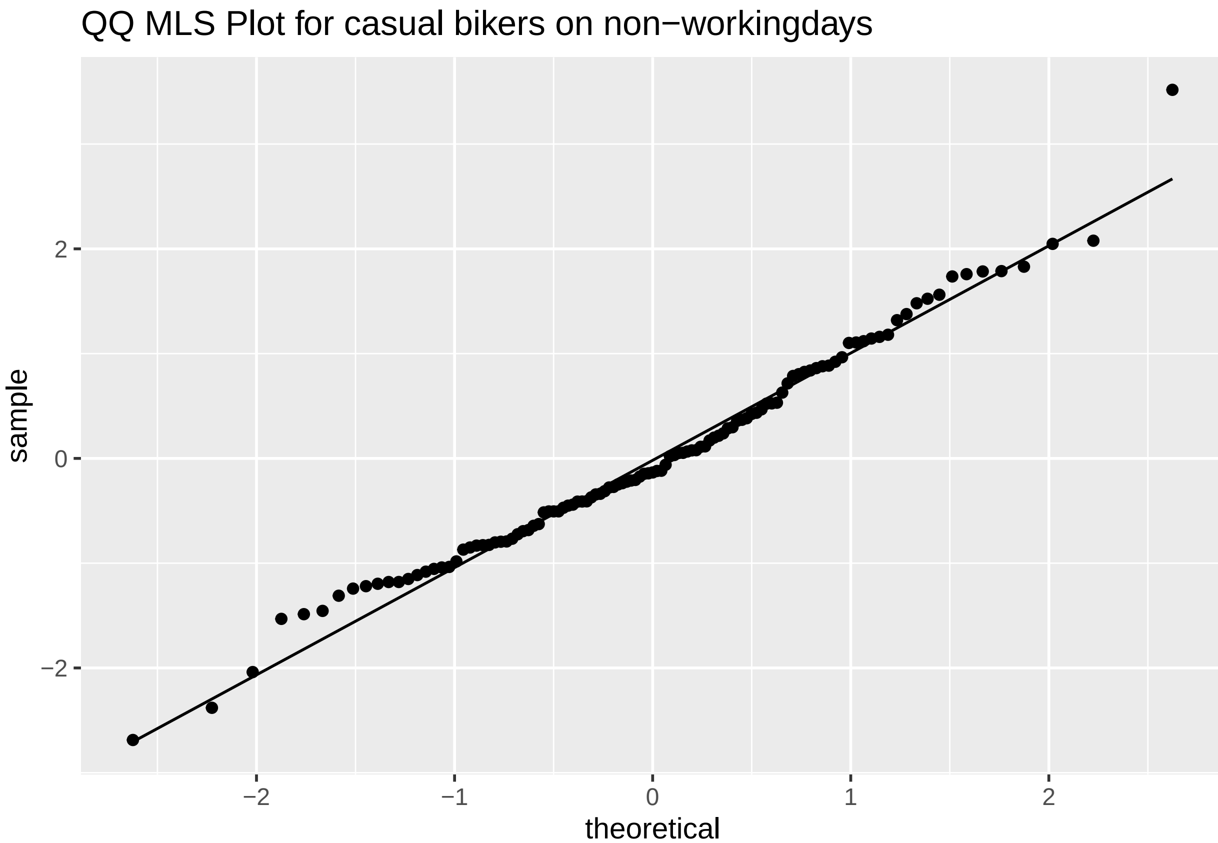

```r
p <- ggplot(data.frame(StanRes.casual.workingday), aes(sample = StanRes.casual.workingday)) +
ggtitle("QQ MLS Plot for casual bikers on workingdays")
p + stat_qq() + stat_qq_line()
```



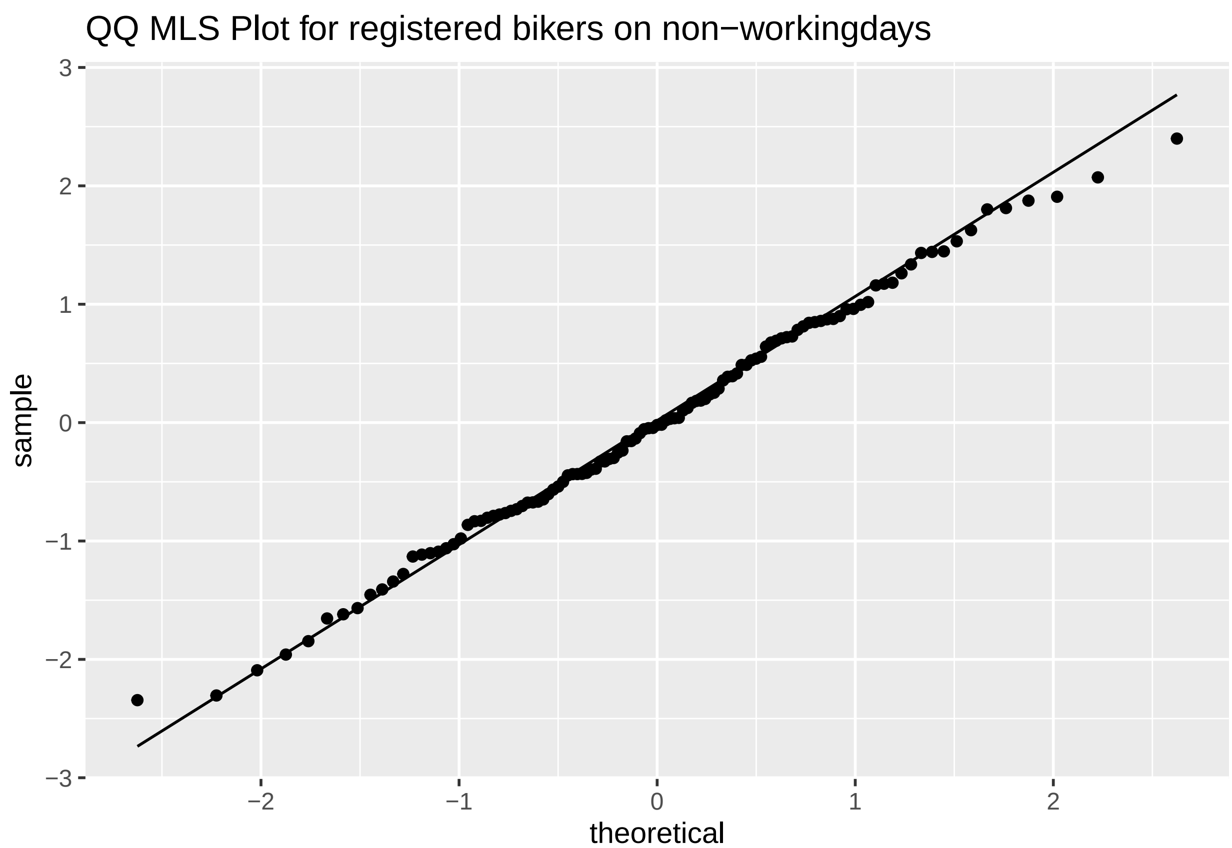

```r
p <- ggplot(data.frame(StanRes.registered.workingday), aes(sample = StanRes.registered.workingday)) +
ggtitle("QQ MLS Plot for registered bikers on workingdays")
p + stat_qq() + stat_qq_line()
```




```
p <- ggplot(data.frame(StanRes.casual.nworkingday), aes(sample = StanRes.casual.nworkingday)) +
ggtitle("QQ MLS Plot for casual bikers on non-workingdays")
p + stat_qq() + stat_qq_line()
```



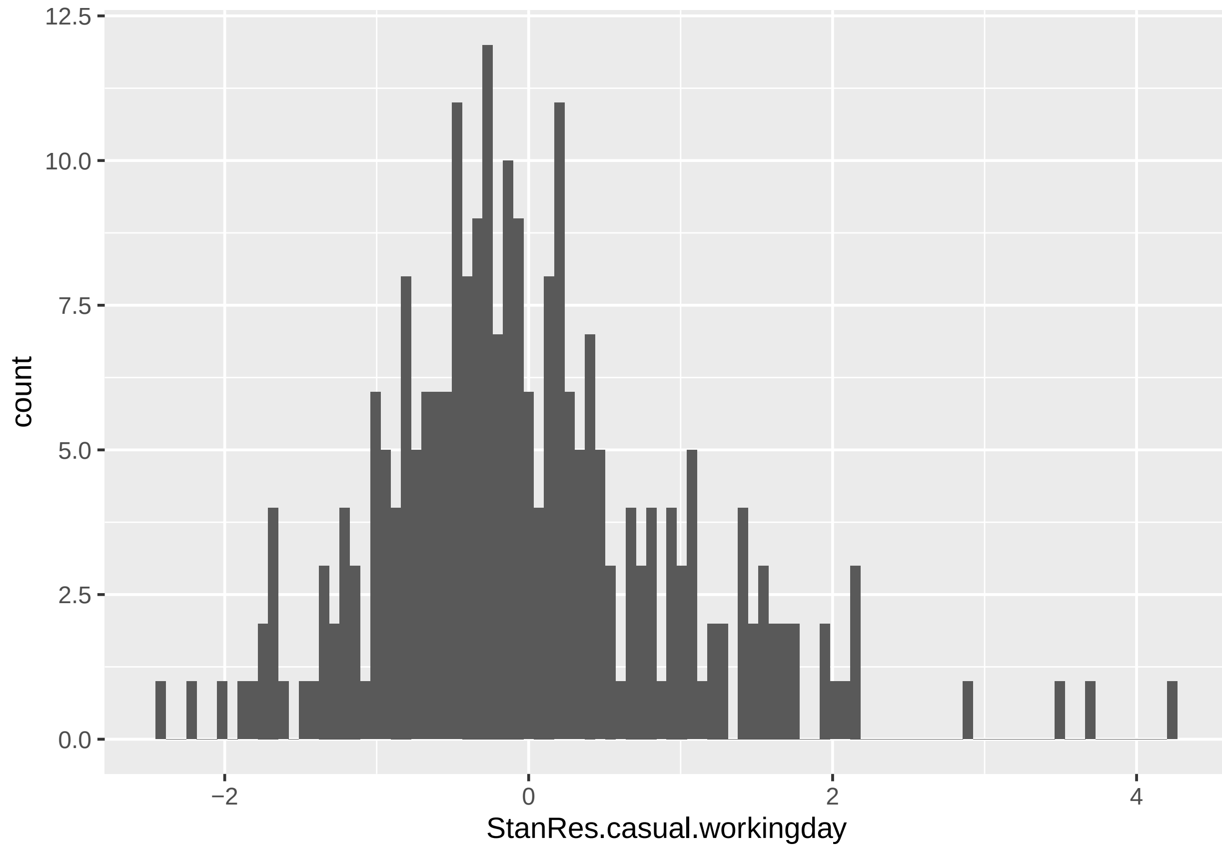

```
p <- ggplot(data.frame(StanRes.registered.nworkingday), aes(sample = StanRes.registered.nworkingday)) +
ggtitle("QQ MLS Plot for registered bikers on non-workingdays")
p + stat_qq() + stat_qq_line()
```



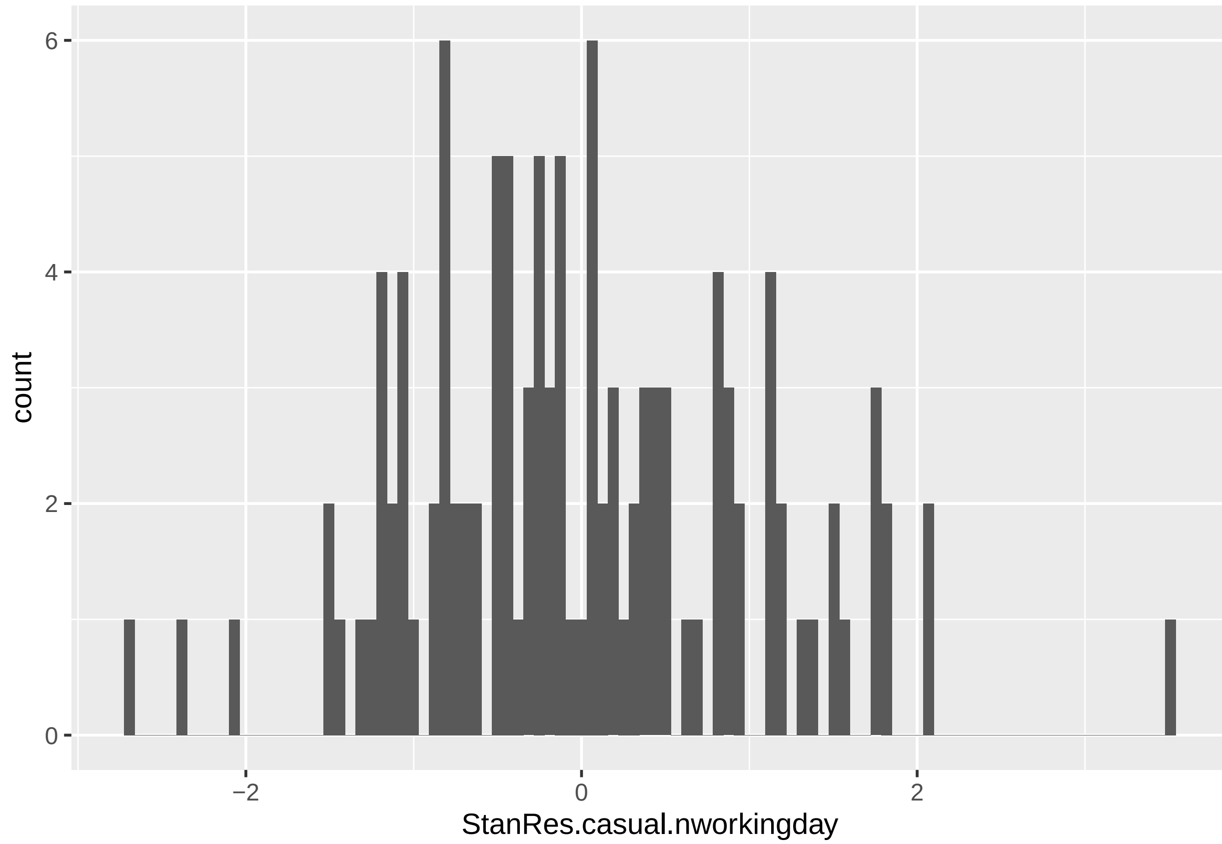
The fitted residual plot and the residual plot suggest that there are extreme outliers in the casual model and that the residual for both models are not evenly distributed around 0, therefore suggesting that there exists heterogeneity in the models.

The QQ plots show a line that is roughly straight, therefore we conclude that the data of registered bikers come from a normally distributed sample. We can also conclude the same for casual bikers, however, there exists some data points that do not come from a normal distribution as indicated by the few datapoints that deviate significantly from the straight line.

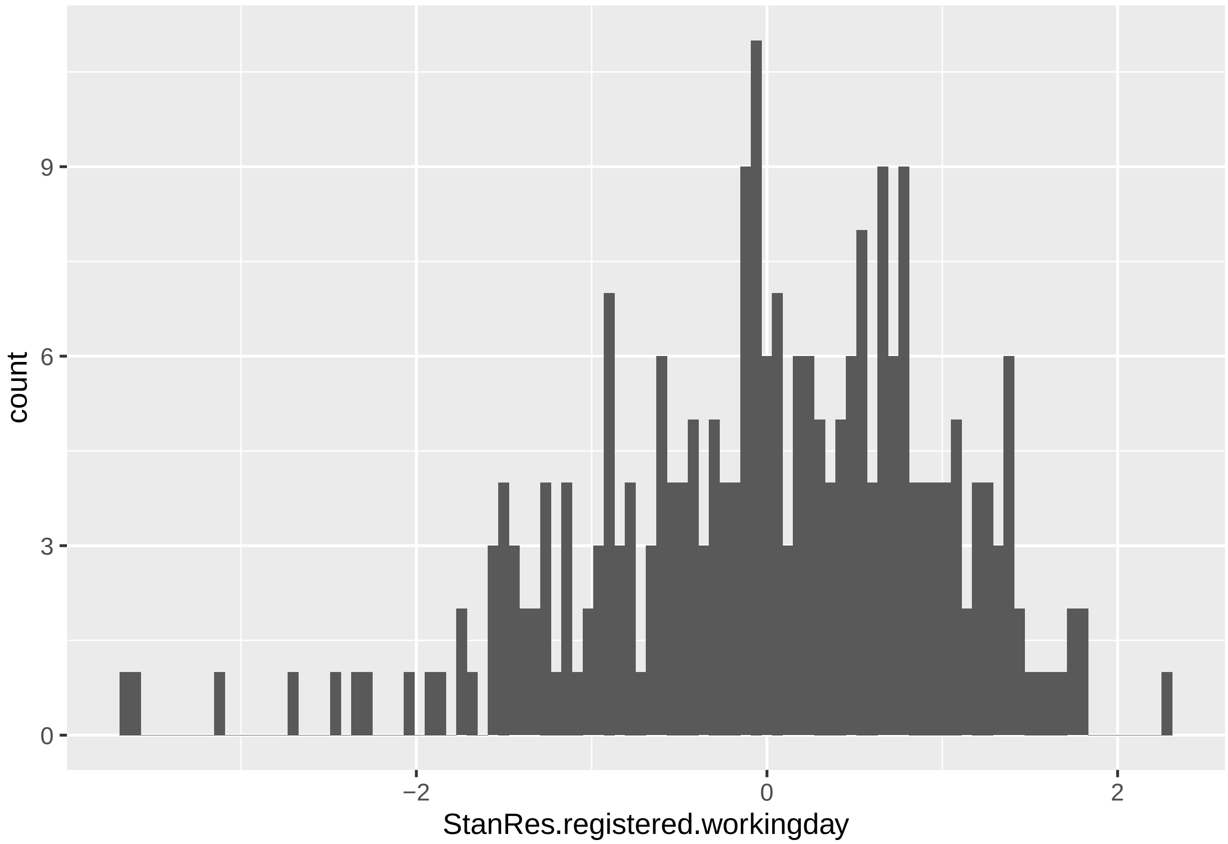
```
p1 <- ggplot(data = data.frame(StanRes.casual.workingday), aes(x = StanRes.casual.workingday)) + geom_hi

p1
```



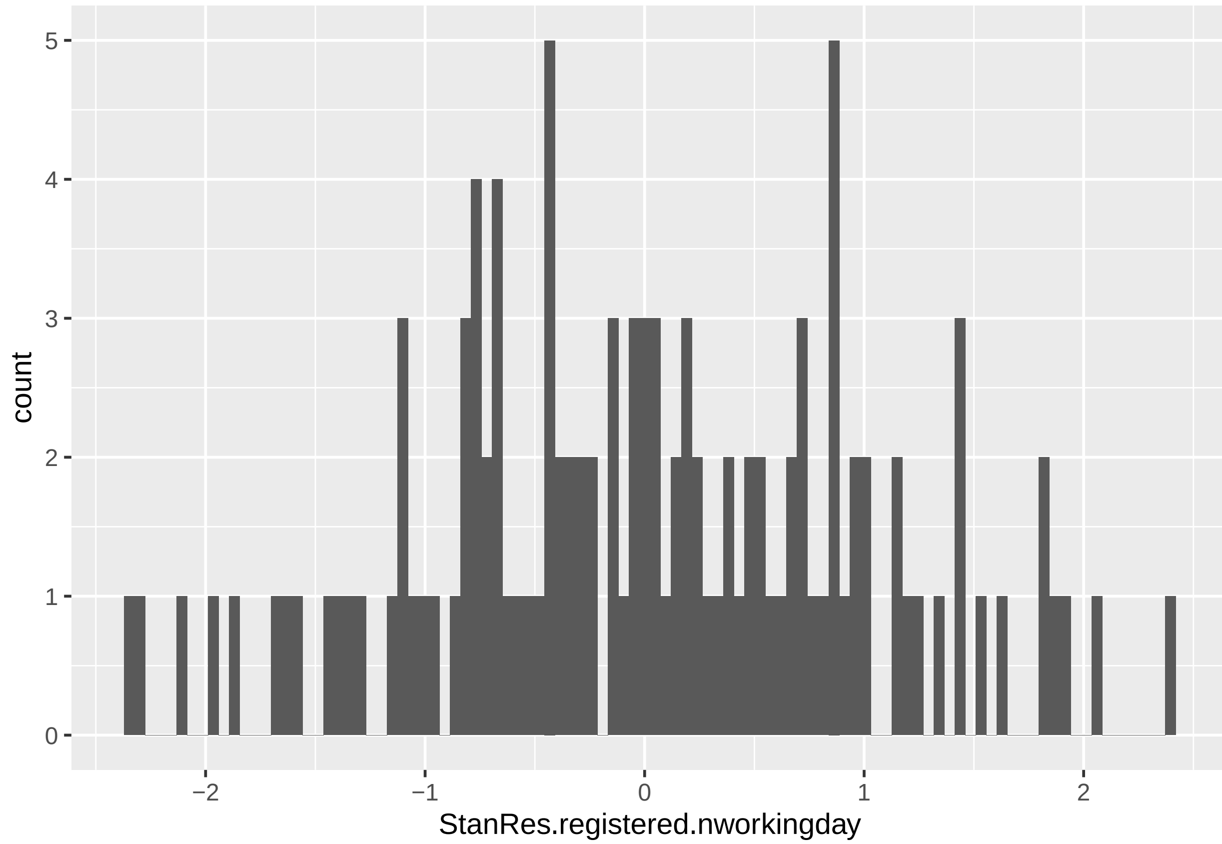

```
p2 <- ggplot(data = data.frame(StanRes.casual.nworkingday), aes(x = StanRes.casual.nworkingday)) + geom_

p2
```



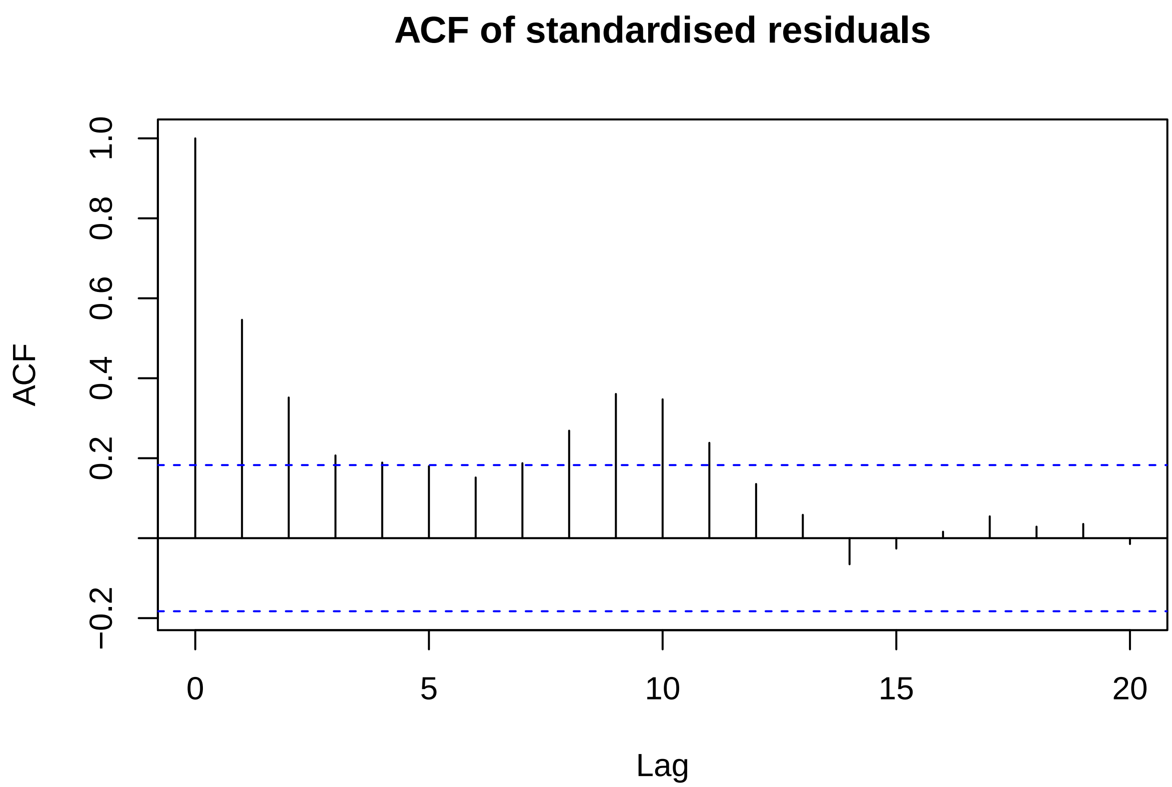

```
p3 <- ggplot(data = data.frame(StanRes.registered.workingday), aes(x = StanRes.registered.workingday))

p3
```



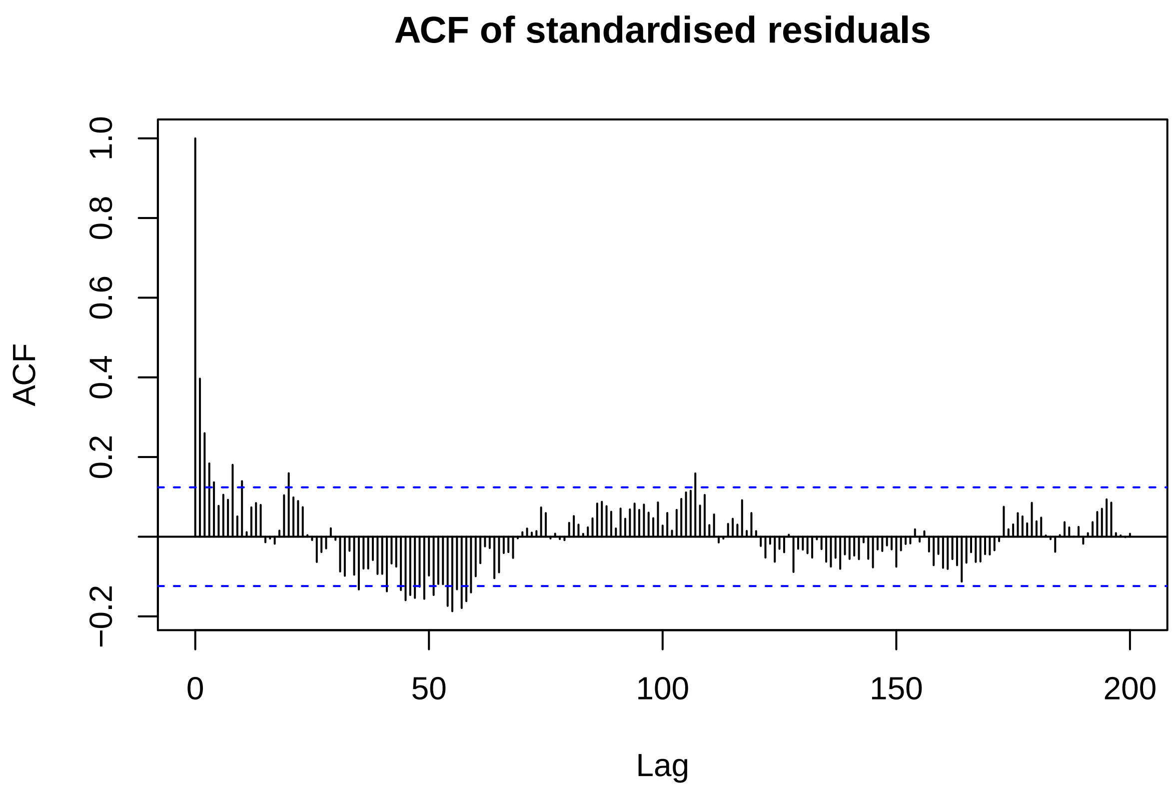

```
p4 <- ggplot(data = data.frame(StanRes.registered.nworkingday), aes(x = StanRes.registered.nworkingday)
p4
```



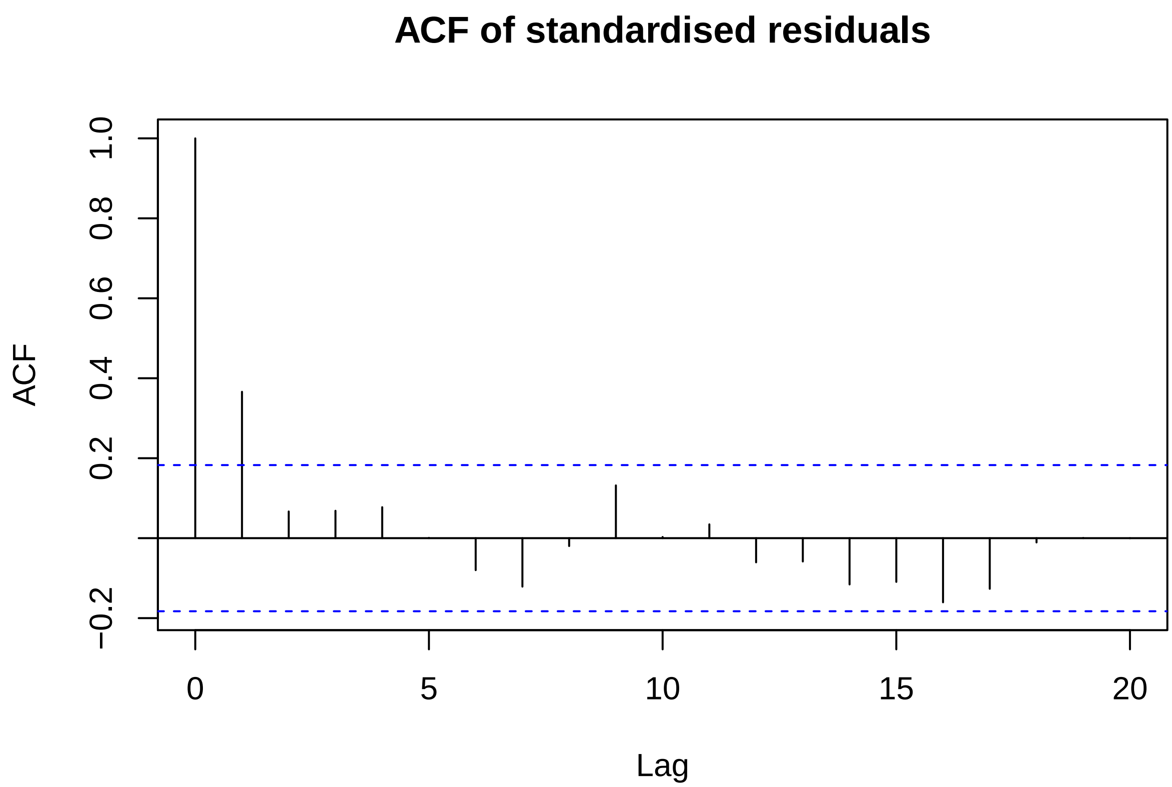

```r
acf(StanRes.registered.nworkingday, main="ACF of standardised residuals")
```



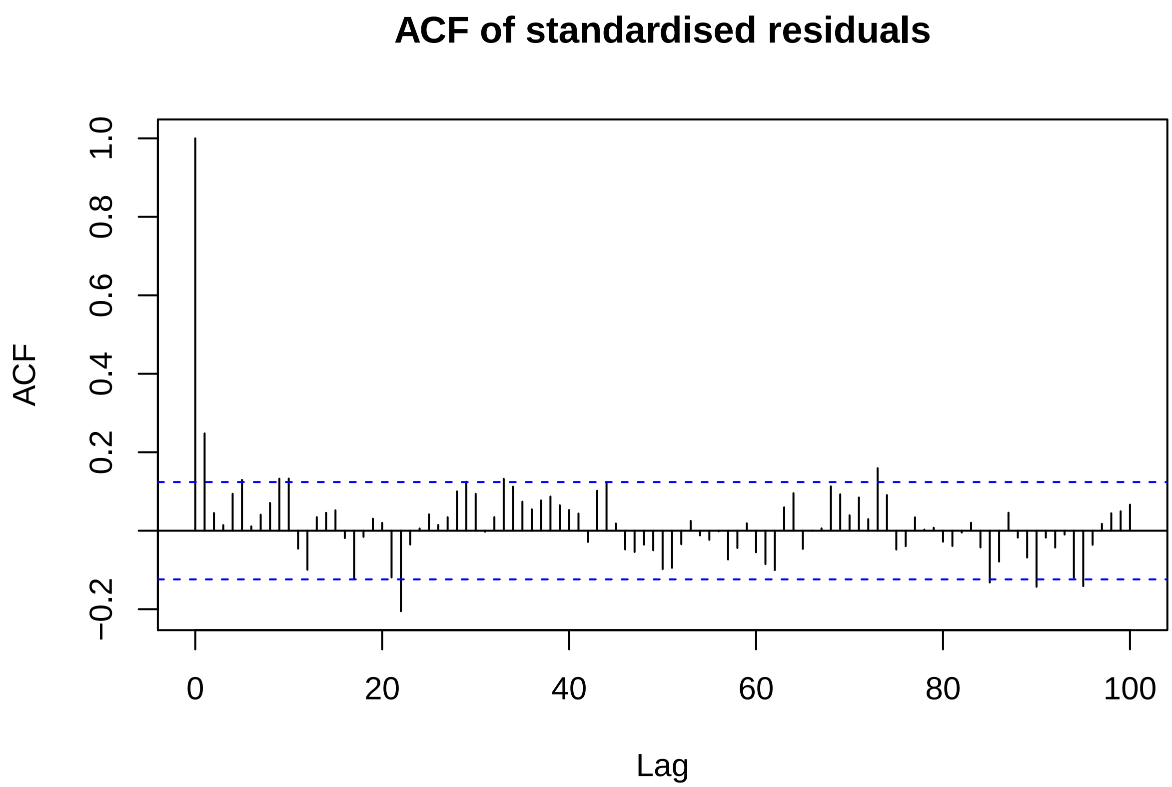

```r
acf(StanRes.registered.workingday, main="ACF of standardised residuals", 200)
```




```r
acf(StanRes.casual.nworkingday, main="ACF of standardised residuals")
```




```r
acf(StanRes.casual.workingday, main="ACF of standardised residuals", 100)
```



Therefore using a gls With corrAR1 to correct correlations between y values in different periods.

model 2

```
m.gls.casual.workingday <- gls(casual ~ actual.windspeed + actual.temp +I(actual.temp^2) + weathersit, d
correlation=corAR1(form=~instant), method="ML")

summary(m.gls.casual.workingday)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: casual ~ actual.windspeed + actual.temp + I(actual.temp^2) +      weathersit
##   Data: training.workingday
##          AIC      BIC    logLik
##   3208.902 3237.073 -1596.451
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~instant
##  Parameter estimate(s):
##     Phi1
## 0.409339
##
## Coefficients:
##                                   Value Std.Error   t-value p-value
## (Intercept)                   -292.1801  83.90200 -3.482397  0.0006
```

```
## actual.windspeed                 -4.3055   1.83322 -2.348629  0.0196
## actual.temp                      66.7114   9.19742  7.253271  0.0000
## I(actual.temp^2)                 -0.9598   0.23163 -4.143686  0.0000
## weathersitModerate:Cloudy/Mist -137.8642  19.66188 -7.011750  0.0000
## weathersitBad: Rain/Snow/Fog   -341.1413  44.81133 -7.612836  0.0000
##
##  Correlation:
##                                (Intr) actl.w actl.t I(.^2) wM:C/M
## actual.windspeed               -0.216
## actual.temp                    -0.906 -0.075
## I(actual.temp^2)                0.841  0.083 -0.982
## weathersitModerate:Cloudy/Mist  0.049 -0.017 -0.171  0.188
## weathersitBad: Rain/Snow/Fog    0.109 -0.067 -0.161  0.167  0.299
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.4735214 -0.6998380 -0.1837750  0.3880913  3.9953681
##
## Residual standard error: 154.4065
## Degrees of freedom: 250 total; 244 residual
```

```r
m.gls.registered.workingday <- gls(registered ~  actual.temp + I(actual.temp^2)+actual.windspeed + weat
correlation=corAR1(form=~instant), method="ML")

summary(m.gls.registered.workingday)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: registered ~ actual.temp + I(actual.temp^2) + actual.windspeed +      weathersit + date_dif
##   Data: training.workingday
##        AIC      BIC    logLik
##   3782.694 3814.387 -1882.347
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~instant
##  Parameter estimate(s):
##     Phi1
## 0.494695
##
## Coefficients:
##                                    Value Std.Error    t-value p-value
## (Intercept)                     232.7873 281.54230   0.826829  0.4091
## actual.temp                     183.7581  33.35502   5.509157  0.0000
## I(actual.temp^2)                 -2.7413   0.82643  -3.317067  0.0010
## actual.windspeed                 -9.0304   5.78612  -1.560709  0.1199
## weathersitModerate:Cloudy/Mist  -349.0268  60.52713  -5.766452  0.0000
## weathersitBad: Rain/Snow/Fog   -1545.0791 138.93462 -11.120908  0.0000
## date_diff                          3.2526   0.52415   6.205531  0.0000
##
##  Correlation:
##                                (Intr) actl.t I(.^2) actl.w wM:C/M wB:R/S
## actual.temp                    -0.862
## I(actual.temp^2)                0.810 -0.982
## actual.windspeed               -0.173 -0.148  0.145
## weathersitModerate:Cloudy/Mist  0.056 -0.167  0.184 -0.008
```

```
## weathersitBad: Rain/Snow/Fog    0.110 -0.133  0.141 -0.074  0.313
## date_diff                       0.051 -0.376  0.331  0.205  0.032 -0.039
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -3.6582184 -0.4978623  0.1207504  0.7009100  1.9261065
##
## Residual standard error: 503.9427
## Degrees of freedom: 250 total; 243 residual
```

```r
m.gls.casual.nworkingday <- gls(casual ~ actual.windspeed + actual.temp +I(actual.temp^2) + weathersit
correlation=corAR1(form=~instant), method="ML")

summary(m.gls.casual.nworkingday)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: casual ~ actual.windspeed + actual.temp + I(actual.temp^2) +      weathersit
##   Data: training.nworkingday
##        AIC      BIC    logLik
##   1693.409 1715.369 -838.7046
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~instant
##  Parameter estimate(s):
##      Phi1
## 0.4745946
##
## Coefficients:
##                                    Value Std.Error   t-value p-value
## (Intercept)                    -565.7639 284.55073 -1.988271  0.0493
## actual.windspeed                -23.3300   7.12136 -3.276056  0.0014
## actual.temp                     162.5514  30.38286  5.350100  0.0000
## I(actual.temp^2)                 -2.5204   0.76242 -3.305775  0.0013
## weathersitModerate:Cloudy/Mist -243.8728  76.67194 -3.180730  0.0019
## weathersitBad: Rain/Snow/Fog   -567.7572 239.91721 -2.366471  0.0197
##
##  Correlation:
##                                (Intr) actl.w actl.t I(.^2) wM:C/M
## actual.windspeed               -0.281
## actual.temp                    -0.896 -0.060
## I(actual.temp^2)                0.830  0.080 -0.982
## weathersitModerate:Cloudy/Mist  0.002 -0.040 -0.088  0.095
## weathersitBad: Rain/Snow/Fog    0.004 -0.249  0.054 -0.044  0.044
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.49539283 -0.70285493 -0.07174161  0.74973266  3.66719315
##
## Residual standard error: 380.7387
## Degrees of freedom: 115 total; 109 residual
```

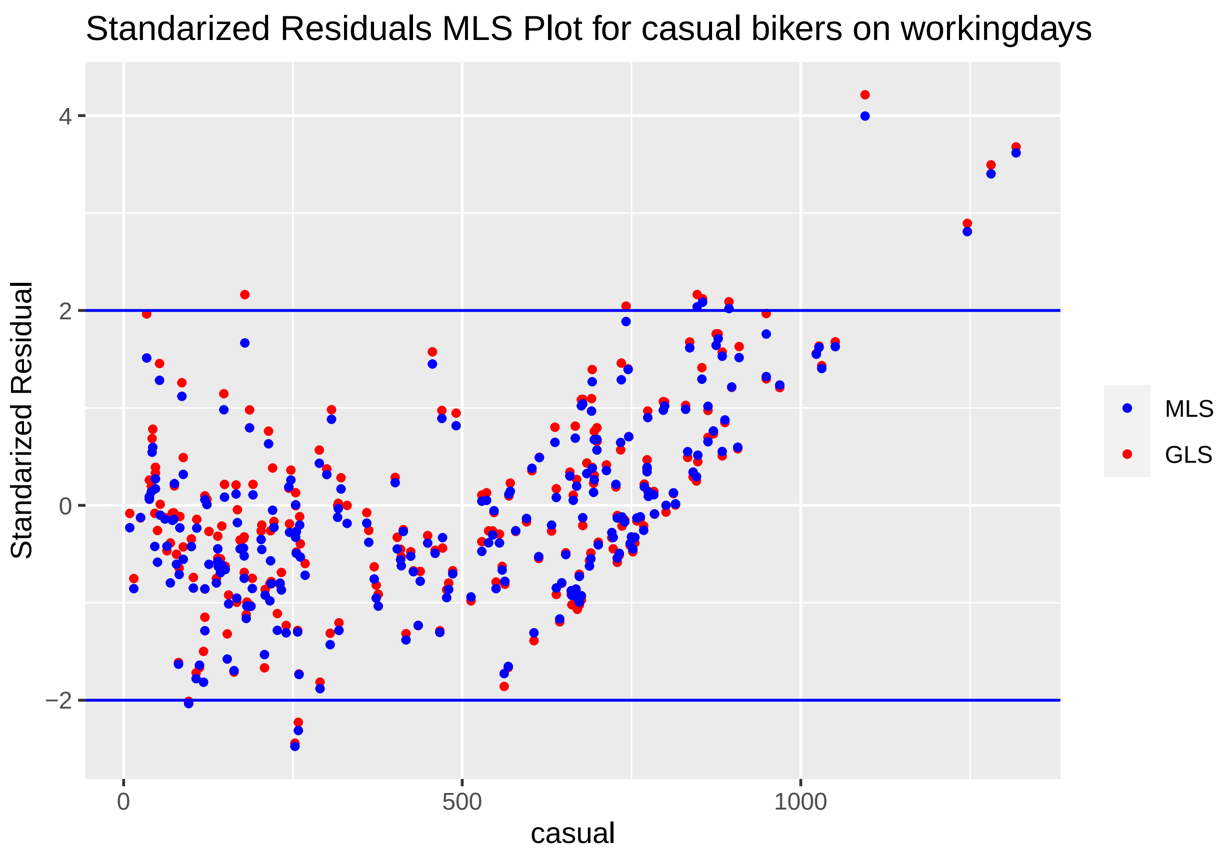
```r
m.gls.registered.nworkingday <- gls(registered ~ actual.temp +  actual.windspeed + weathersit, data = t

summary(m.gls.registered.nworkingday)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: registered ~ actual.temp + actual.windspeed + weathersit
##   Data: training.nworkingday
##        AIC      BIC    logLik
##   1741.781 1760.996 -863.8907
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~instant
##  Parameter estimate(s):
##      Phi1
## 0.8274103
##
## Coefficients:
##                                   Value Std.Error   t-value p-value
## (Intercept)                   1303.4975 272.08118  4.790841  0.0000
## actual.temp                     67.1992  11.88276  5.655186  0.0000
## actual.windspeed               -20.1243   7.90017 -2.547327  0.0122
## weathersitModerate:Cloudy/Mist  -265.4322  80.92342 -3.280042  0.0014
## weathersitBad: Rain/Snow/Fog   -1463.8807 226.05018 -6.475910  0.0000
##
##  Correlation:
##                                (Intr) actl.t actl.w wM:C/M
## actual.temp                    -0.834
## actual.windspeed               -0.321 -0.048
## weathersitModerate:Cloudy/Mist -0.107  0.044 -0.060
## weathersitBad: Rain/Snow/Fog   -0.007  0.116 -0.280  0.026
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.44652281 -0.72636708  0.03613507  0.81099354  1.86484084
##
## Residual standard error: 619.706
## Degrees of freedom: 115 total; 110 residual
```

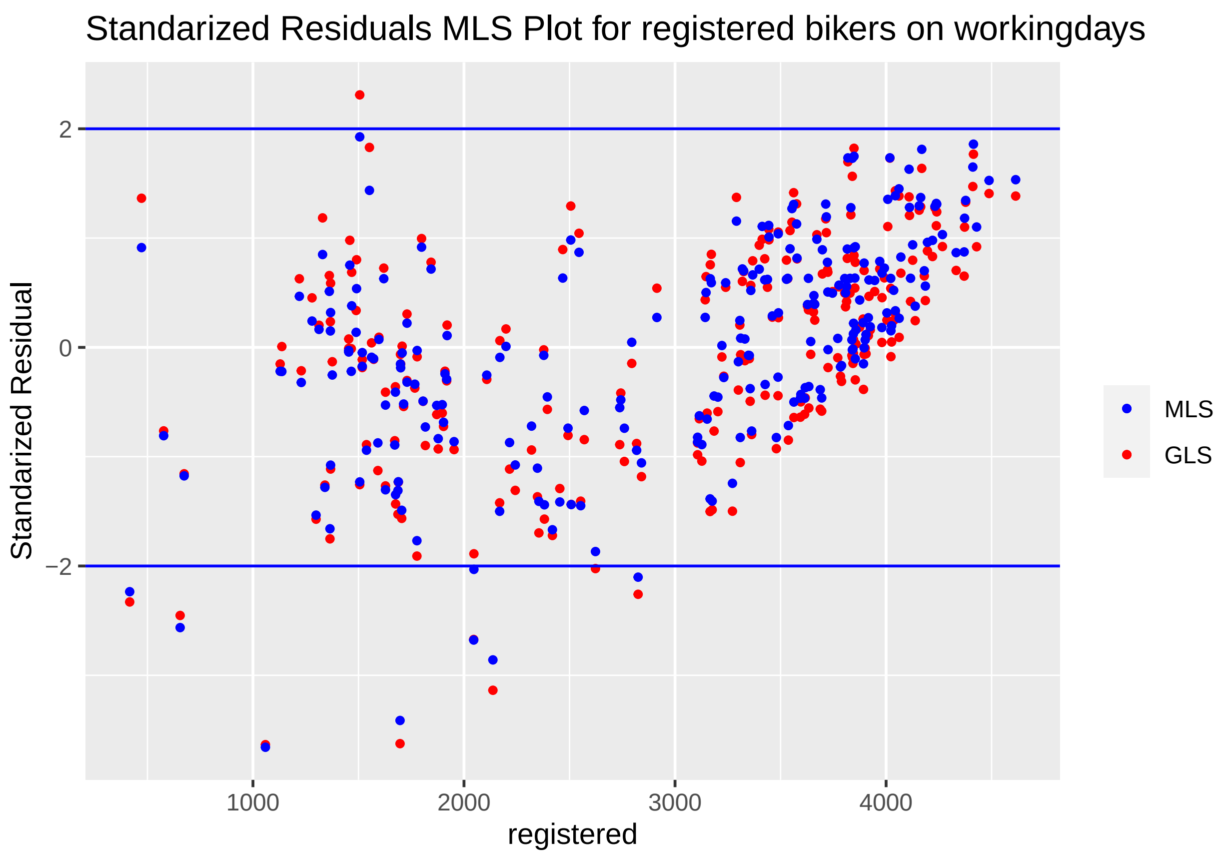
Model2 diagnosis

```
StanResGLS.casual.nworkingday <- residuals(m.gls.casual.nworkingday,"pearson")
StanResGLS.casual.workingday <- residuals(m.gls.casual.workingday,"pearson")
StanResGLS.registered.nworkingday <- residuals(m.gls.registered.nworkingday,"pearson")
StanResGLS.registered.workingday <- residuals(m.gls.registered.workingday,"pearson")
```

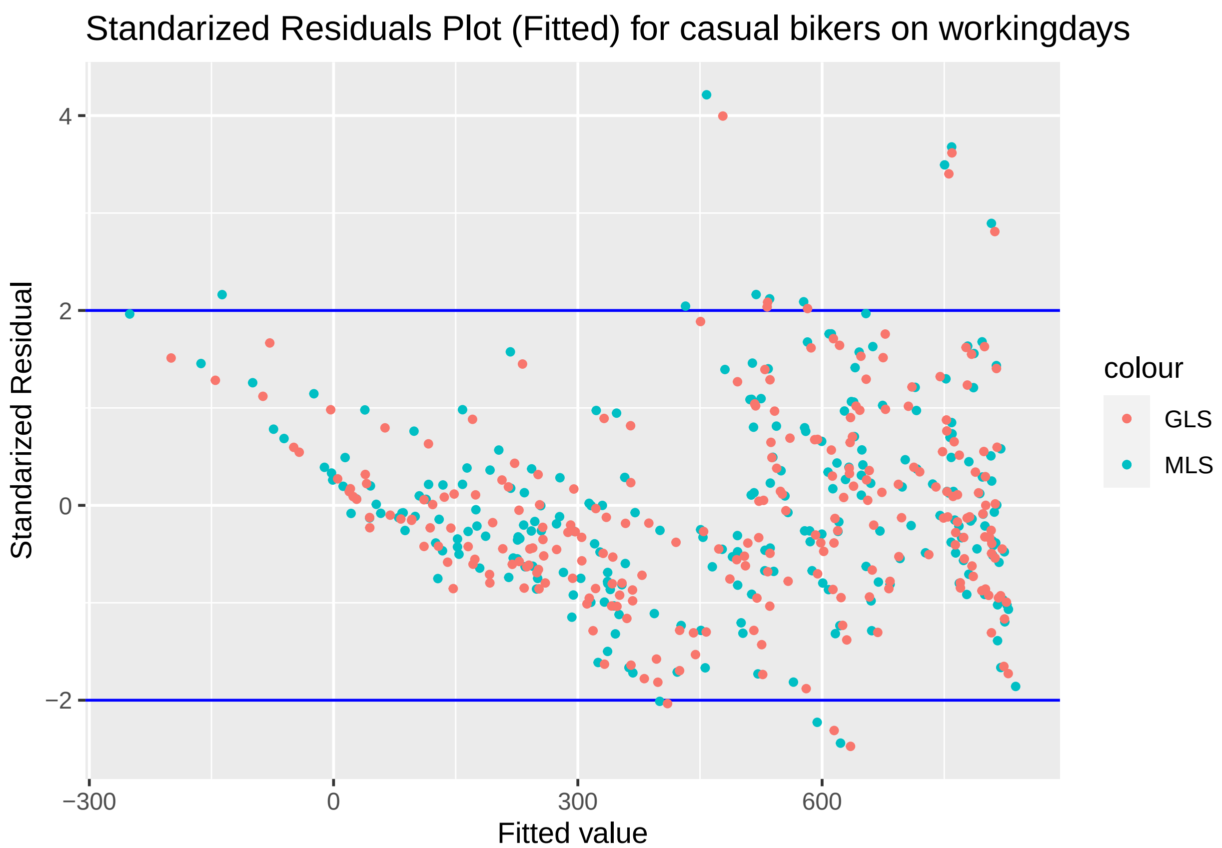
```
ggplot(data=training.workingday, aes(x=casual)) +
geom_point(aes(y=StanRes.casual.workingday, color = "MLS"), size = 1) +
geom_point(aes(y=StanResGLS.casual.workingday, color = "GLS"), size = 1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
scale_color_manual(name = element_blank(), labels = c("MLS", "GLS"), values = c("blue", "red")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for casual bikers on working
```



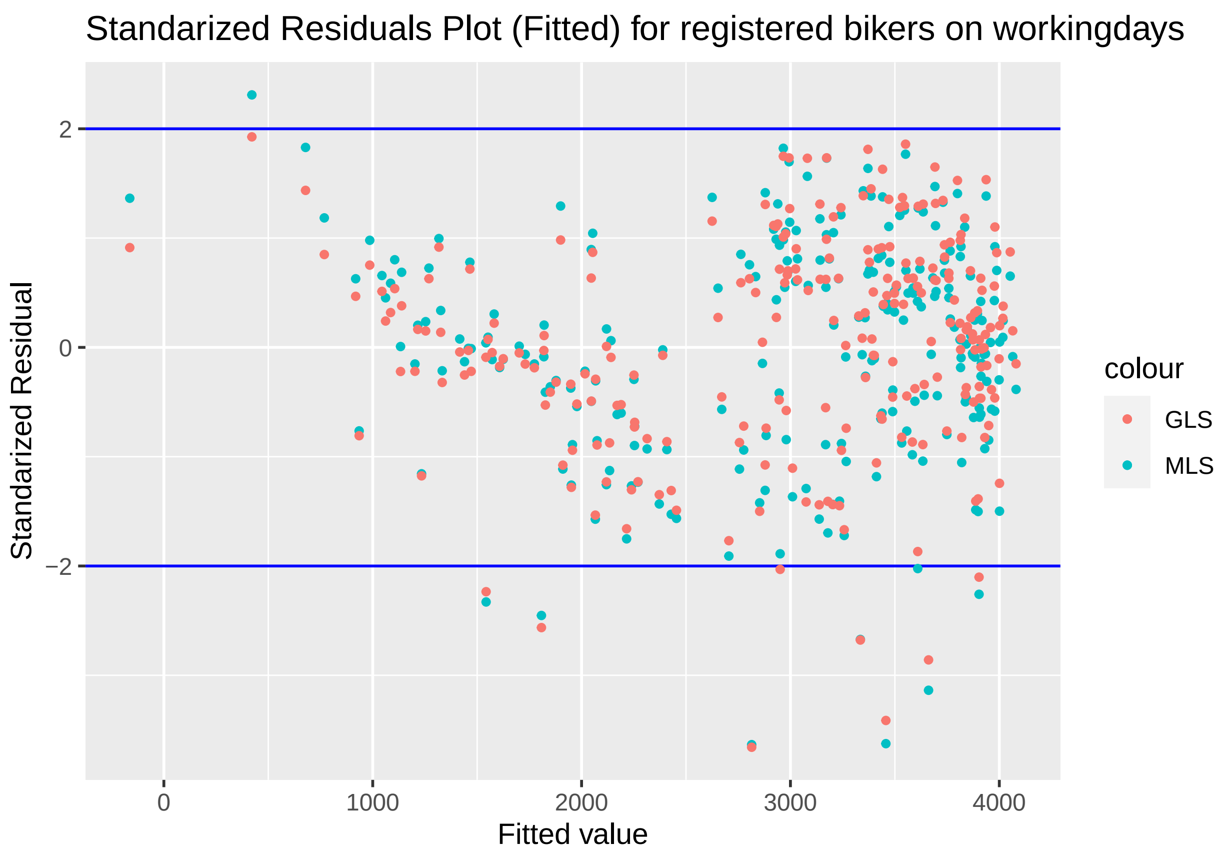

```r
ggplot(data=training.workingday, aes(x=registered)) +
geom_point(aes(y=StanRes.registered.workingday, color = "MLS"), size = 1) +
  geom_point(aes(y=StanResGLS.registered.workingday, color = "GLS"), size = 1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
scale_color_manual(name = element_blank(), labels = c("MLS", "GLS"), values = c("blue", "red")) +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for registered bikers on worl
```



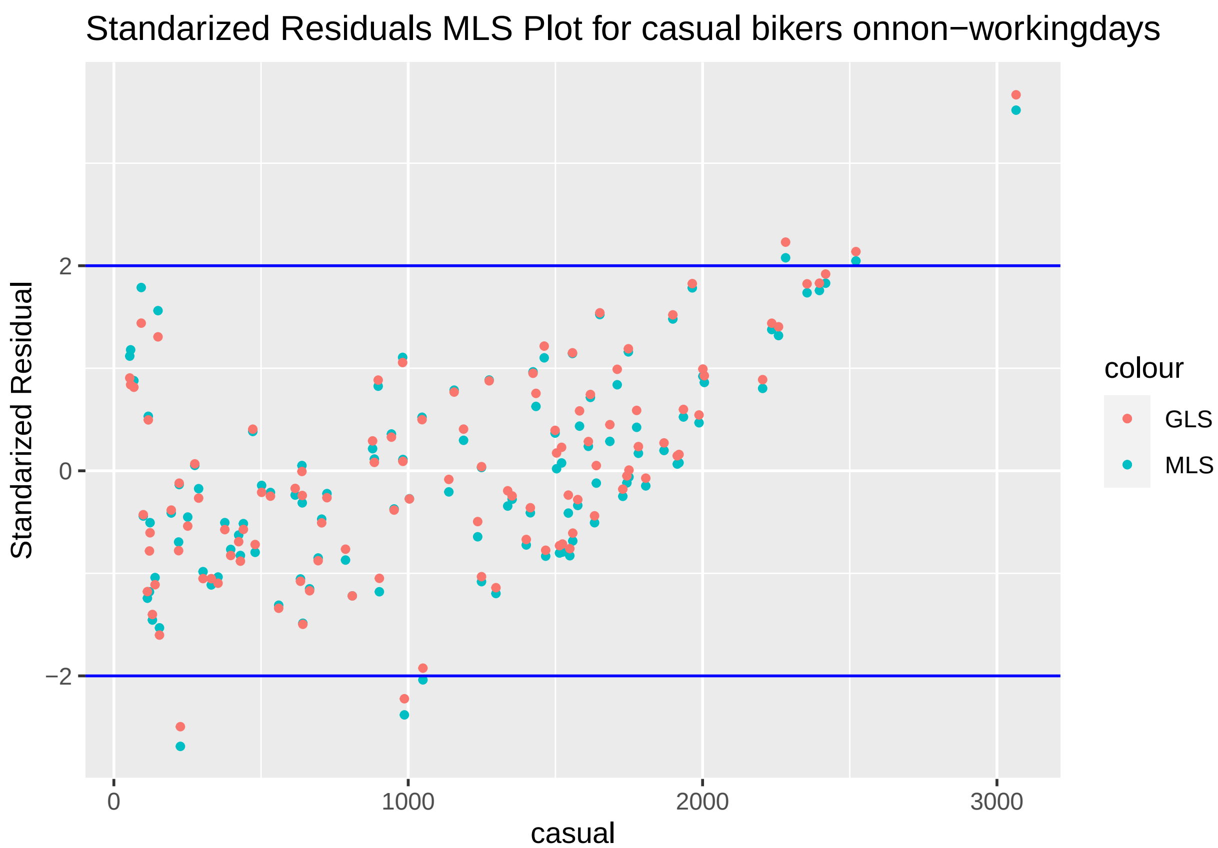

```
FittedGLS_casual.workingday = fitted(m.gls.casual.workingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitt
geom_point(aes(x=FittedGLS_casual.workingday, y=StanResGLS.casual.workingday, color = "GLS"), size = 1)
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals Plot (Fitted) for casual bikers on workingdays")
```



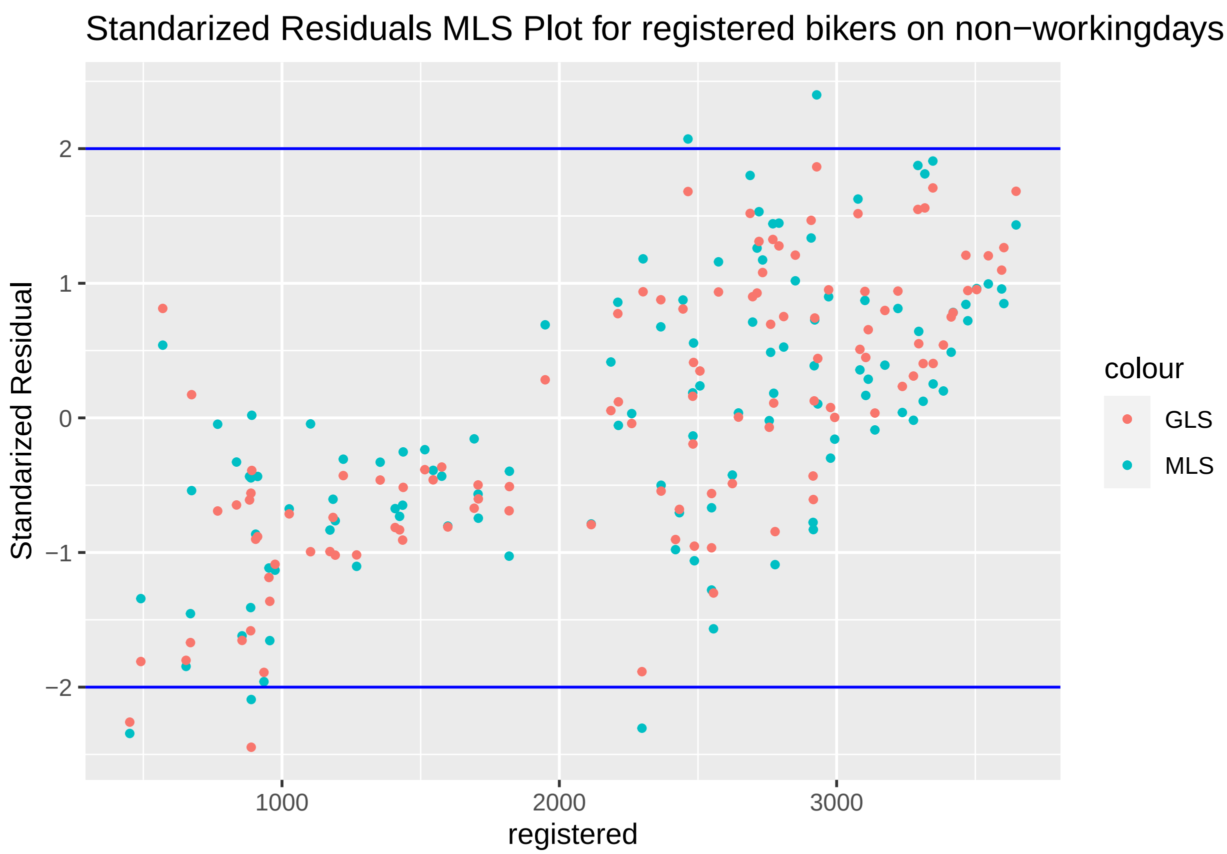

```
FittedGLS_registered.workingday = fitted(model.registered.workingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitte
geom_point(aes(x=FittedGLS_registered.workingday, y=StanResGLS.registered.workingday, color = "GLS"), si
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals Plot (Fitted) for registered bikers on workingdays")
```



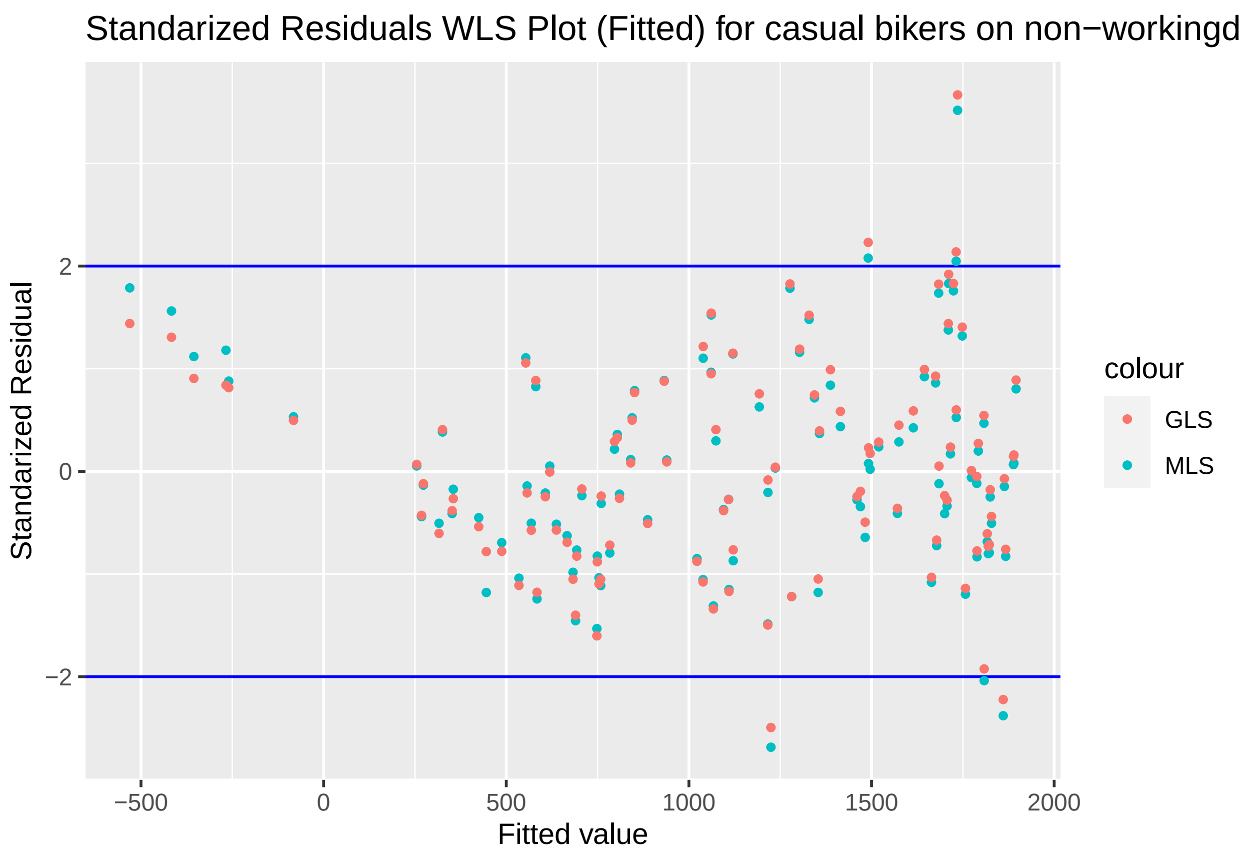

```
ggplot(data=training.nworkingday, aes(x=casual)) +
geom_point(aes(y=StanRes.casual.nworkingday, color = "MLS"), size = 1) +
geom_point(aes(y=StanResGLS.casual.nworkingday, color = "GLS"), size = 1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for casual bikers onnon-worki
```



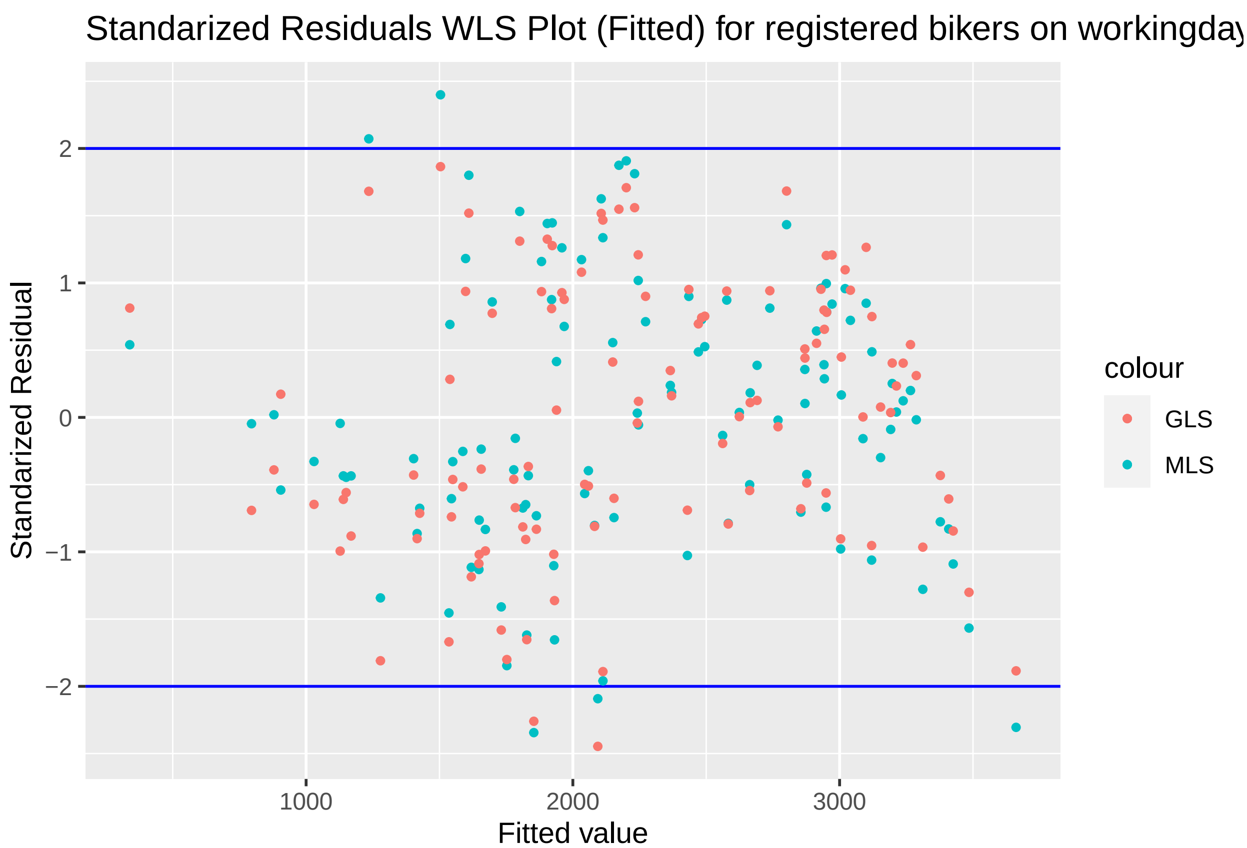

```r
ggplot(data=training.nworkingday, aes(x=registered)) +
geom_point(aes(y=StanRes.registered.nworkingday, color = "MLS"), size = 1) +
geom_point(aes(y=StanResGLS.registered.nworkingday, color = "GLS"), size = 1) +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot for registered bikers on non
```



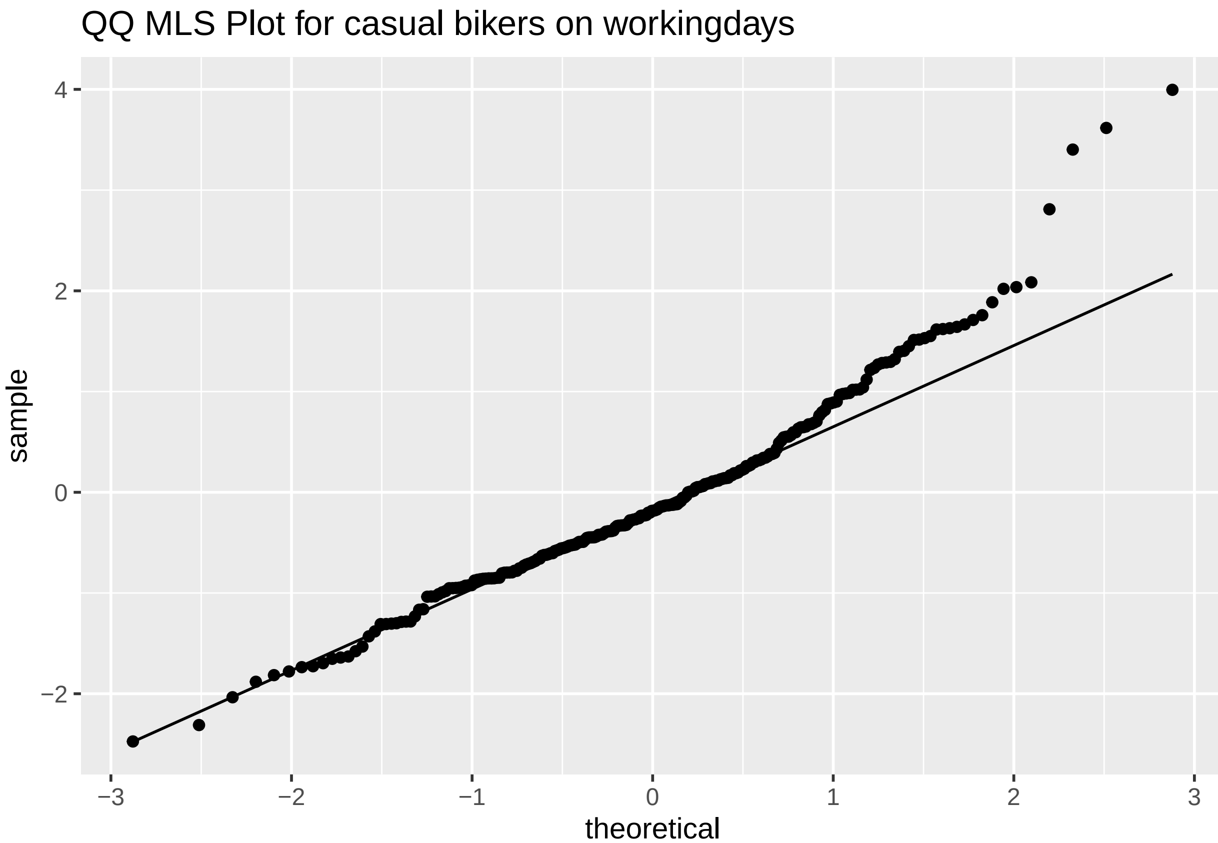

```
FittedGLS_casual.nworkingday = fitted(model.casual.nworkingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitte
geom_point(aes(x=FittedGLS_casual.nworkingday, y=StanResGLS.casual.nworkingday, color = "GLS"), size =
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals WLS Plot (Fitted) for casual bikers on non-workingdays")
```



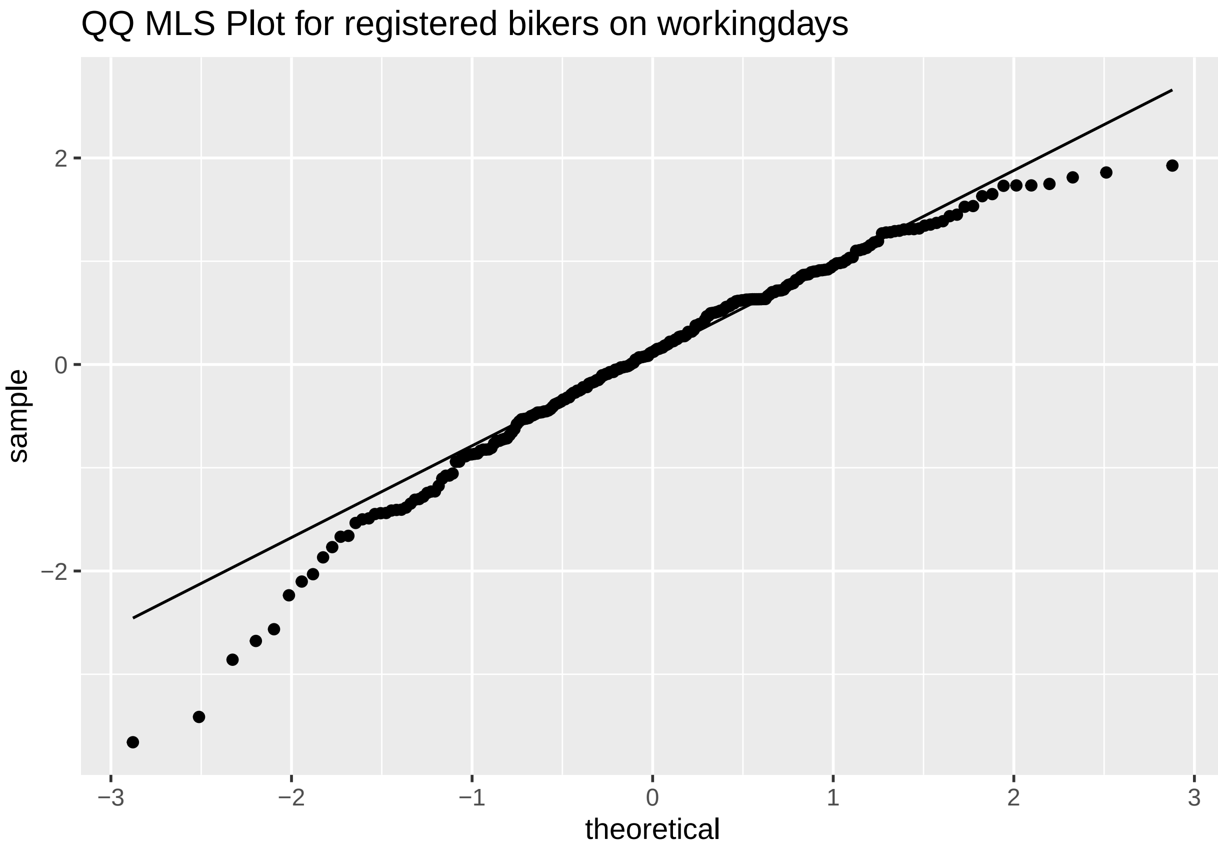

```
FittedGLS_registered.nworkingday = fitted(model.registered.nworkingday)

ggplot() +
geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') + geom_point(aes(x=Fitte
geom_point(aes(x=FittedGLS_registered.nworkingday, y=StanResGLS.registered.nworkingday, color = "GLS"),
labs(y = "Standarized Residual") + labs(x = "Fitted value") +
ggtitle("Standarized Residuals WLS Plot (Fitted) for registered bikers on workingdays")
```



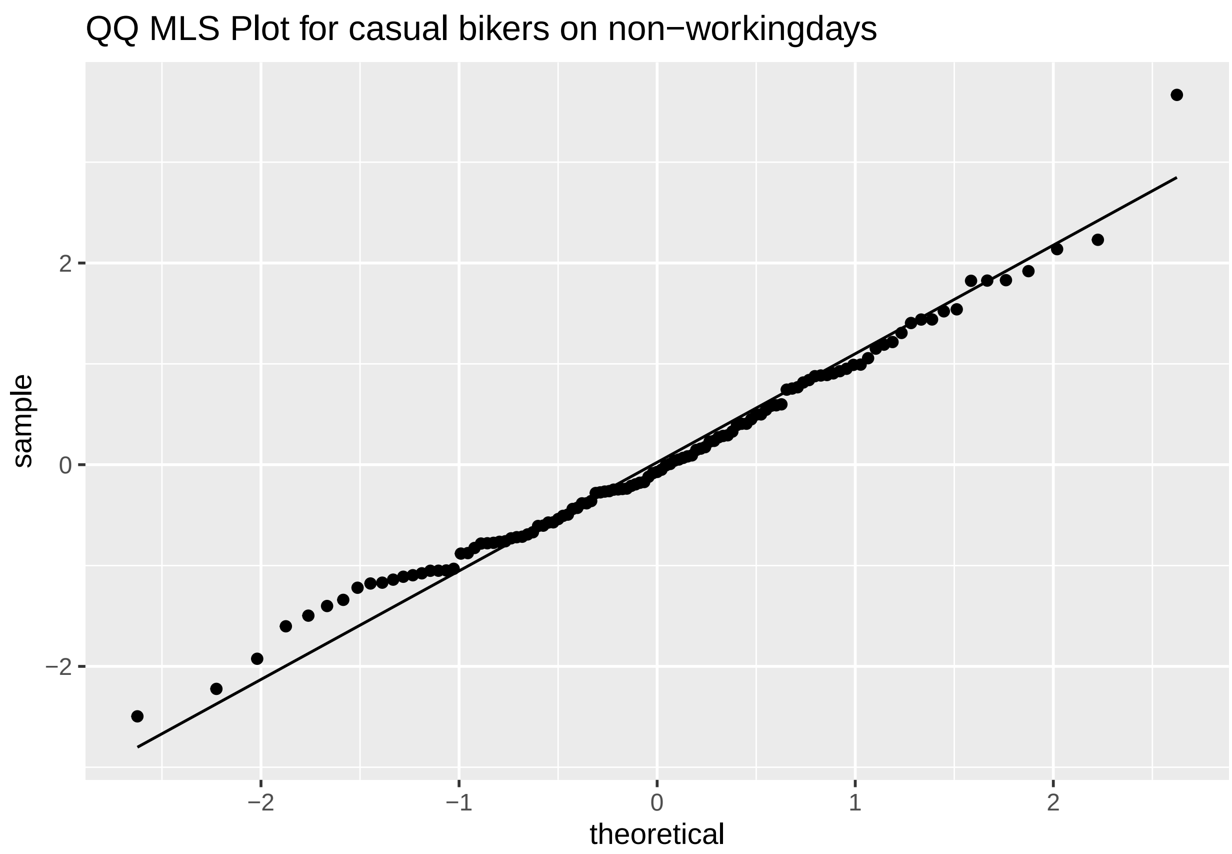

```r
p <- ggplot(data.frame(StanResGLS.casual.workingday), aes(sample = StanResGLS.casual.workingday)) +
ggtitle("QQ MLS Plot for casual bikers on workingdays")
p + stat_qq() + stat_qq_line()
```



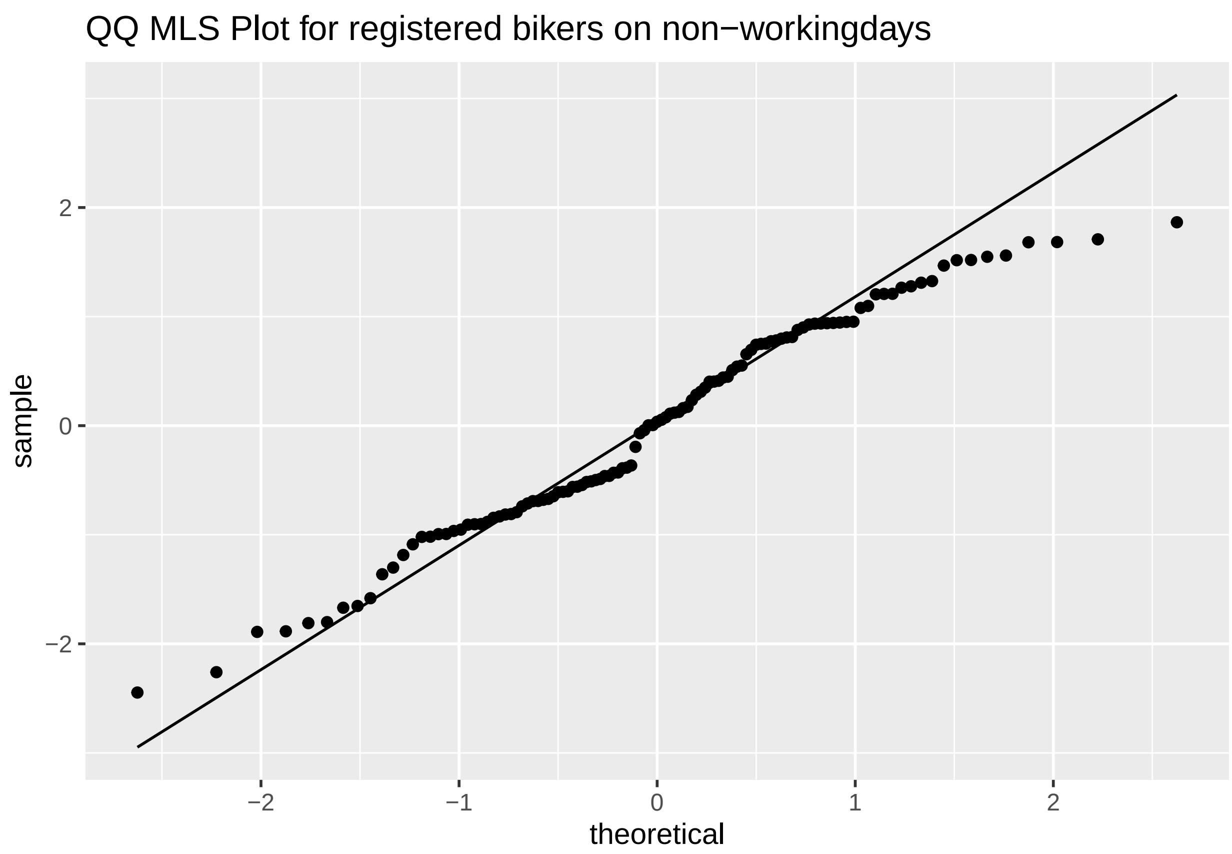

```
p <- ggplot(data.frame(StanResGLS.registered.workingday), aes(sample = StanResGLS.registered.workingday)
ggtitle("QQ MLS Plot for registered bikers on workingdays")
p + stat_qq() + stat_qq_line()
```



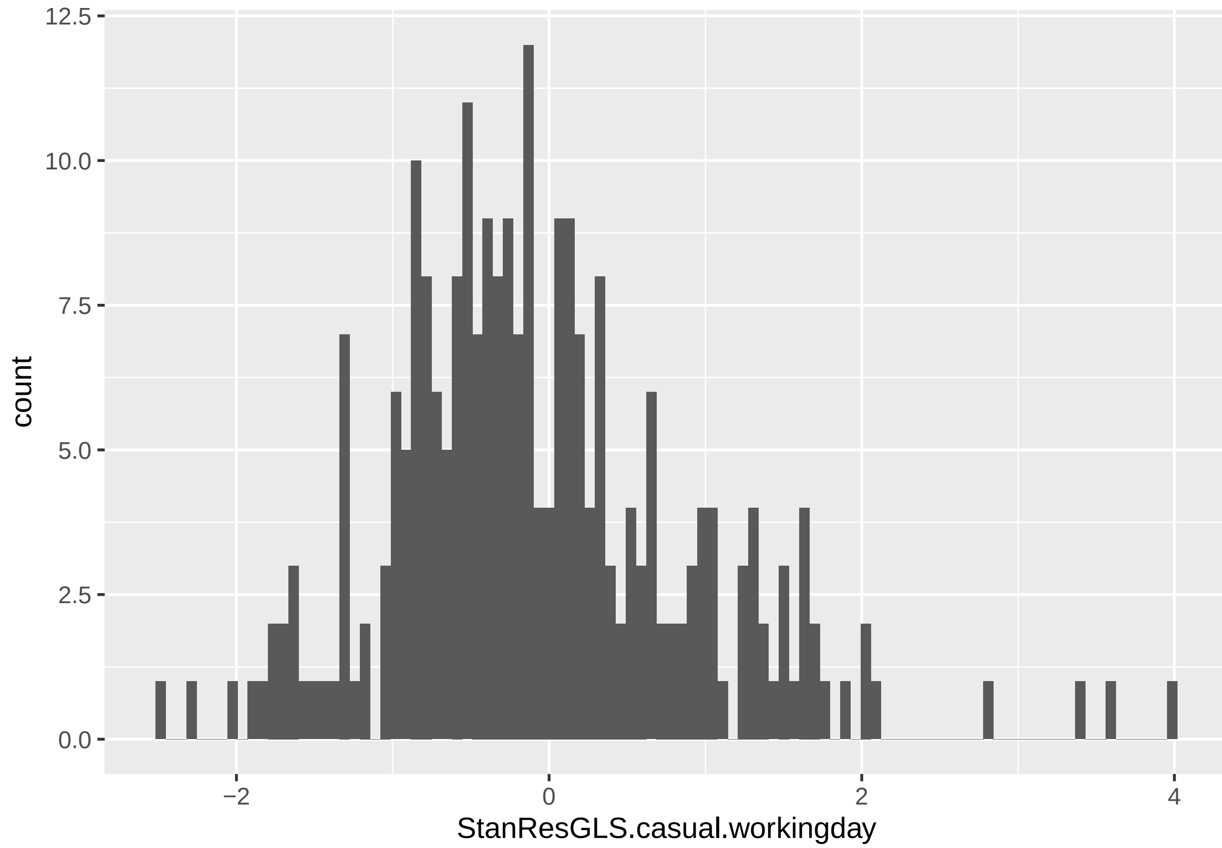

```r
p <- ggplot(data.frame(StanResGLS.casual.nworkingday), aes(sample = StanResGLS.casual.nworkingday)) +
ggtitle("QQ MLS Plot for casual bikers on non-workingdays")
p + stat_qq() + stat_qq_line()
```



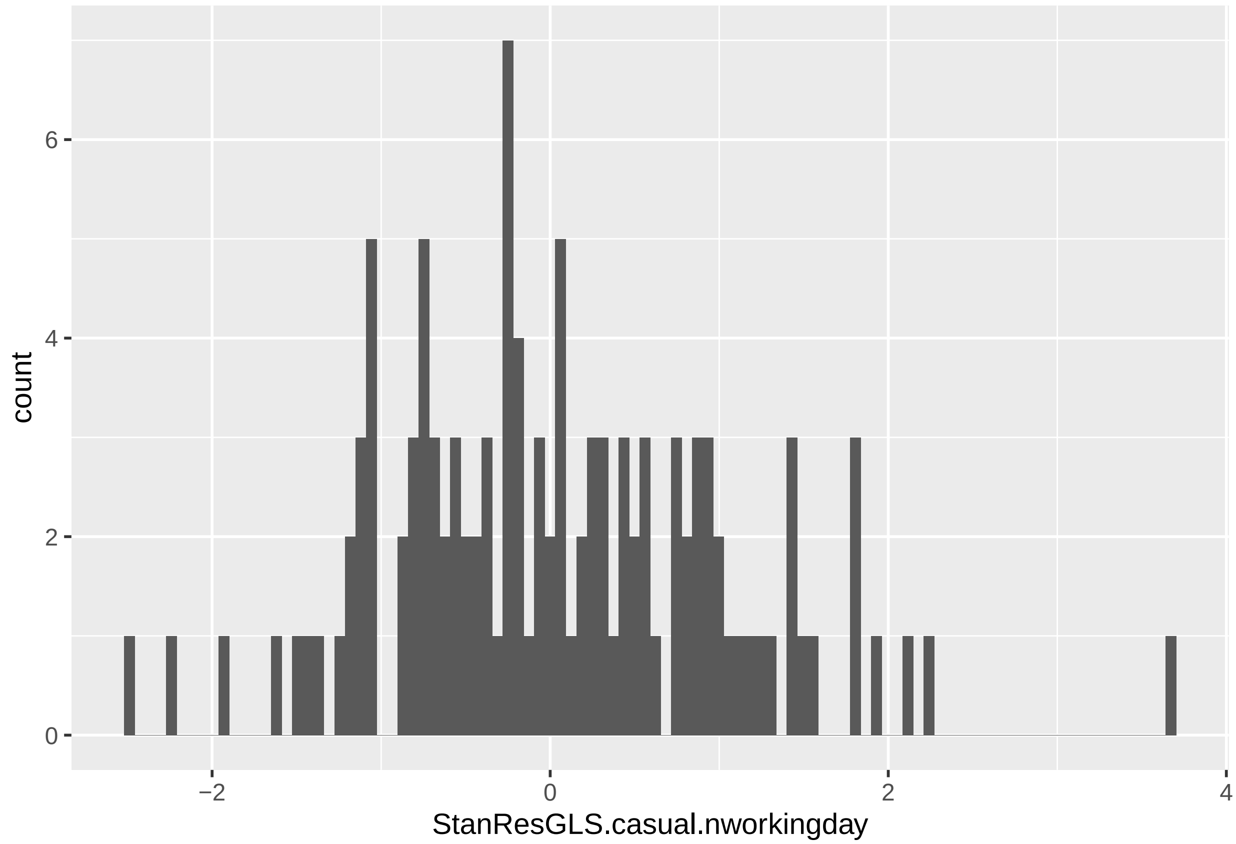

```
p <- ggplot(data.frame(StanResGLS.registered.nworkingday), aes(sample = StanResGLS.registered.nworkingda
ggtitle("QQ MLS Plot for registered bikers on non-workingdays")
p + stat_qq() + stat_qq_line()
```



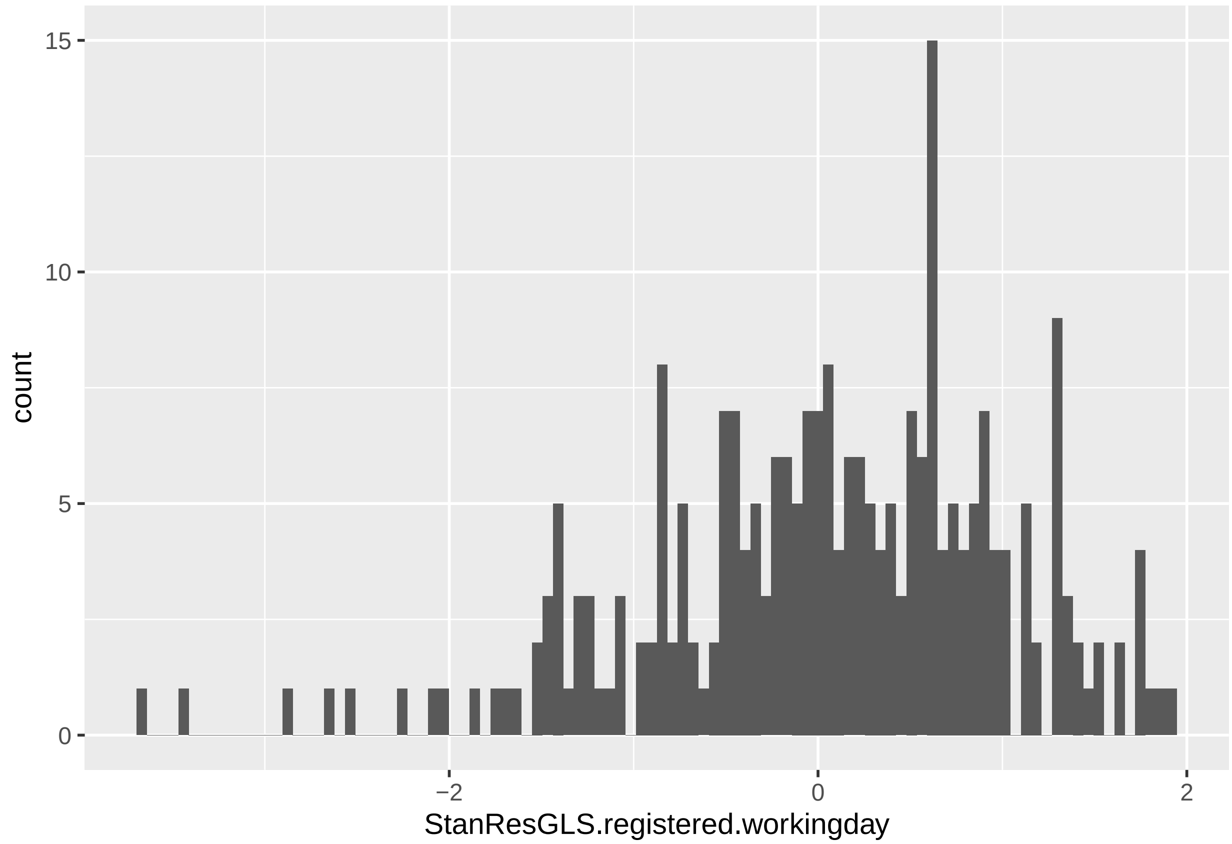

```
p1 <- ggplot(data = data.frame(StanResGLS.casual.workingday), aes(x = StanResGLS.casual.workingday)) + g

p1
```



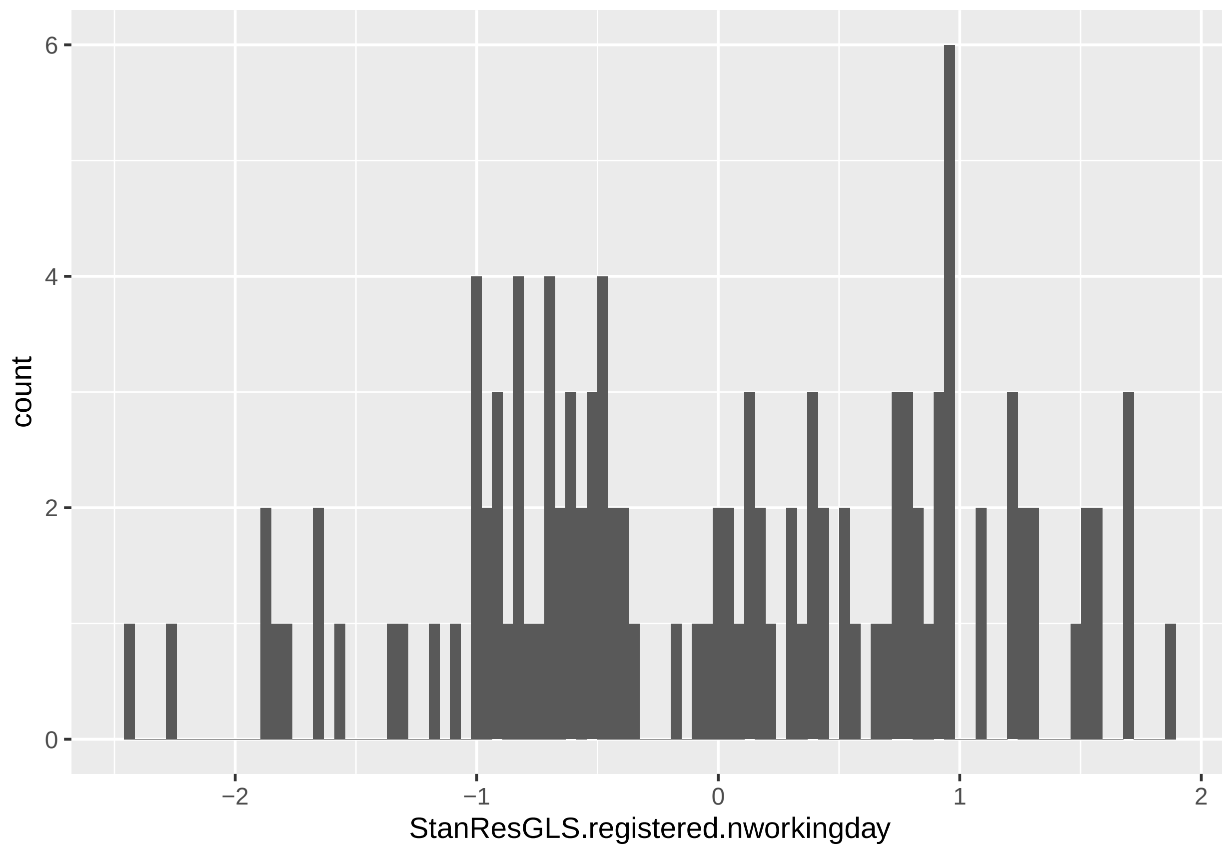

```
p2 <- ggplot(data = data.frame(StanResGLS.casual.nworkingday), aes(x = StanResGLS.casual.nworkingday))

p2
```




```
p3 <- ggplot(data = data.frame(StanResGLS.registered.workingday), aes(x = StanResGLS.registered.working

p3
```




```
p4 <- ggplot(data = data.frame(StanResGLS.registered.nworkingday), aes(x = StanResGLS.registered.nworkin
p4
```



Validation with model 2

```
p.casual.workingday <- predict(m.gls.casual.workingday, validate.workingday)
error.casual.workingday <- (p.casual.workingday- validate.workingday$casual)
RMSE_validation.caual.workingday <- sqrt(mean(error.casual.workingday^2))
RMSEGLS.casual.workingday <- sqrt(mean(resid(m.gls.casual.workingday)^2))

p.casual.nworkingday <- predict(m.gls.casual.nworkingday, validate.nworkingday)
error.casual.nworkingday <- (p.casual.nworkingday- validate.nworkingday$casual)
RMSE_validation.caual.nworkingday <- sqrt(mean(error.casual.nworkingday^2))
RMSEGLS.casual.nworkingday <- sqrt(mean(resid(m.gls.casual.nworkingday)^2))
```

Square root mean square error for validation data set

```
RMSE_validation.caual.workingday
```

```
## [1] 353.0044
```

```
RMSE_validation.caual.nworkingday
```

```
## [1] 761.3066
```

square root mean square error for training data set

```r
RMSEGLS.casual.workingday
```

```
## [1] 151.8887
```

```r
RMSEGLS.casual.nworkingday
```

```
## [1] 382.2737
```

```r
p.registered.workingday <- predict(m.gls.registered.workingday, validate.workingday)
error.registered.workingday <- (p.registered.workingday- validate.workingday$registered)
RMSE_validation.registered.workingday <- sqrt(mean(error.registered.workingday^2))
RMSEGLS.registered.workingday <- sqrt(mean(resid(m.gls.registered.workingday)^2))

p.registered.nworkingday <- predict(m.gls.registered.nworkingday,  validate.nworkingday)
error.registered.nworkingday <- (p.registered.nworkingday- validate.nworkingday$registered)
RMSE_validation.registered.nworkingday <- sqrt(mean(error.registered.nworkingday^2))
RMSEGLS.registered.nworkingday <- sqrt(mean(resid(m.gls.registered.nworkingday)^2))
```

Square root mean square error for validation data set

```r
RMSE_validation.registered.workingday
```

```
## [1] 1109.121
```

```r
RMSE_validation.registered.nworkingday
```

```
## [1] 1674.663
```

square root mean square error for training data set

```r
RMSEGLS.registered.workingday
```

```
## [1] 490.0298
```

```r
RMSEGLS.registered.nworkingday
```

```
## [1] 613.6237
```

Relative mean square error

```r
mean((error.casual.workingday)^2) / mean((validate.workingday$casual)^2)
```

```
## [1] 0.1715408
```

```r
mean((error.casual.nworkingday)^2) / mean((validate.nworkingday$casual)^2)
```

```
## [1] 0.1646588
```

```r
mean((error.registered.workingday)^2) / mean((validate.workingday$registered)^2)
```

```
## [1] 0.04596131
```

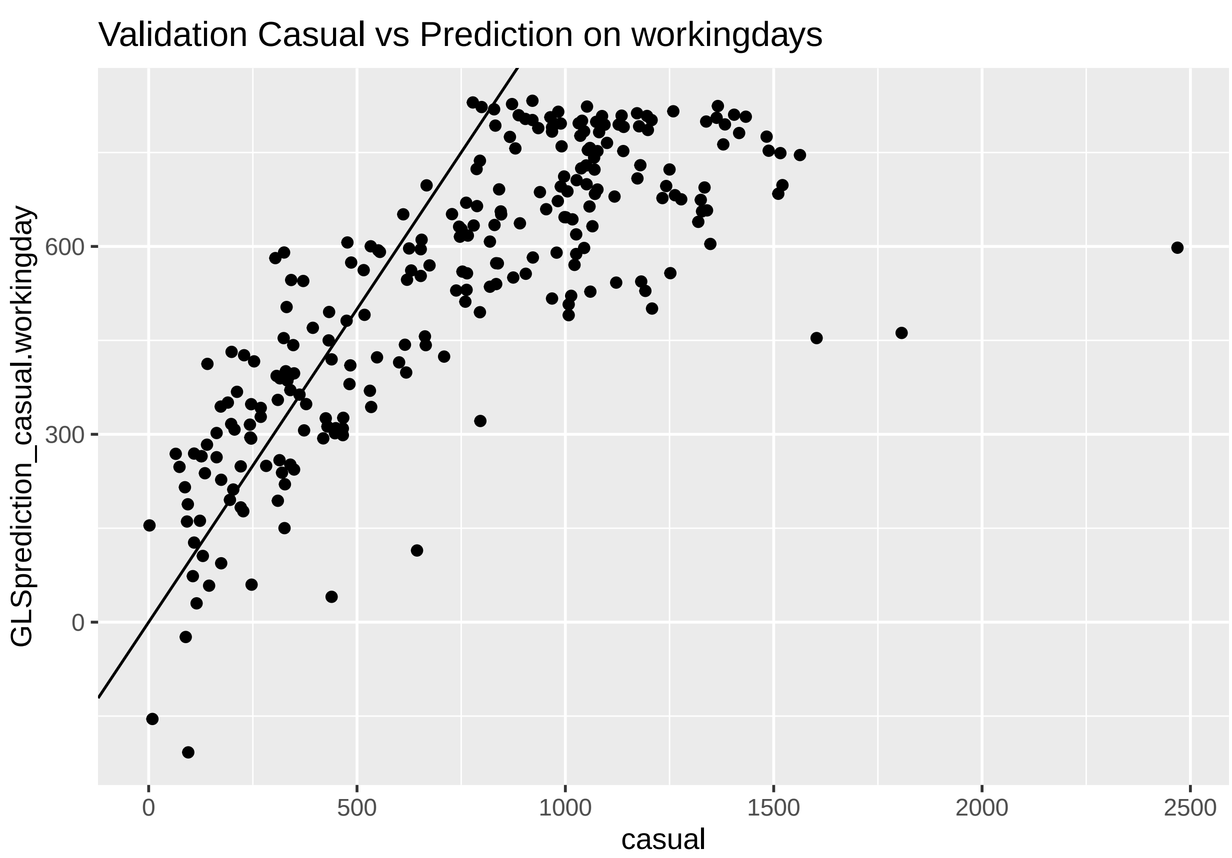
```r
mean((error.registered.nworkingday)^2) / mean((validate.nworkingday$registered)^2)
```

```
## [1] 0.1878861
```

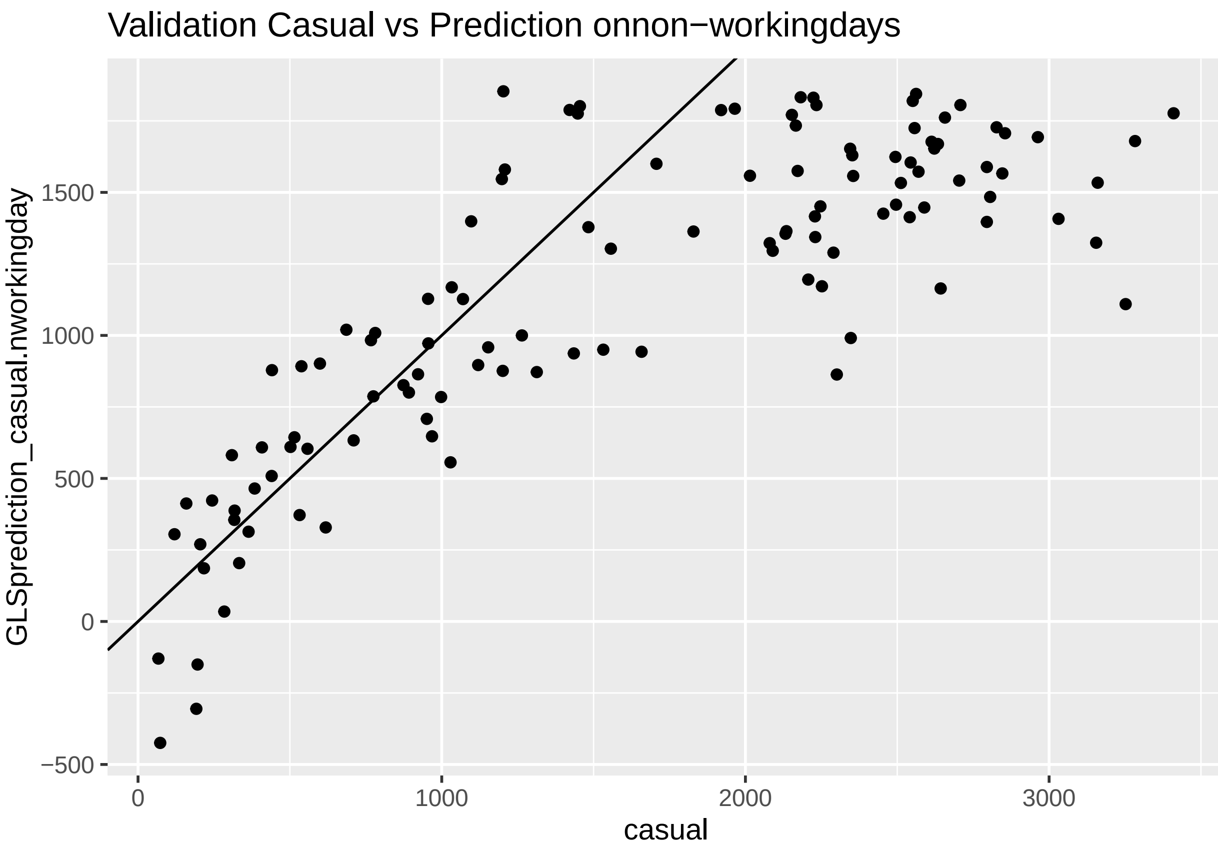
Our model predicts the bike data in 2012 with mean error of 23 percent and 16 percent within the true value of casual and registered counts respectively. However, our model have twice as large of square root of mean square error with the validation data set than with the training data set.

```r
validate.workingday <- validate.workingday %>% mutate(GLSprediction_registered.workingday = predict(m.gl

validate.nworkingday <- validate.nworkingday %>% mutate(GLSprediction_registered.nworkingday = predict(r
```

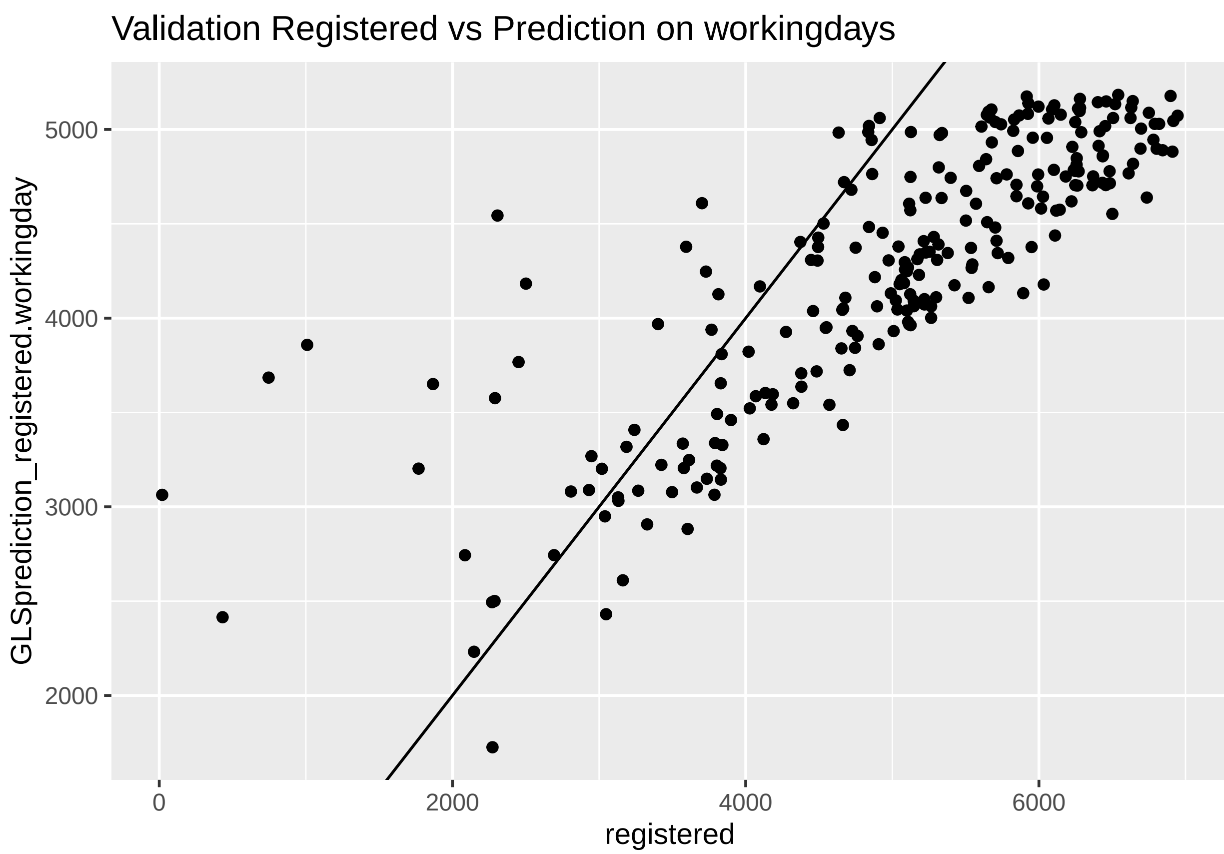
```r
ggplot(validate.workingday, aes(x = casual, y = GLSprediction_casual.workingday)) + geom_point() +
geom_abline(intercept = 0, slope = 1) +
ggtitle("Validation Casual vs Prediction on workingdays")
```



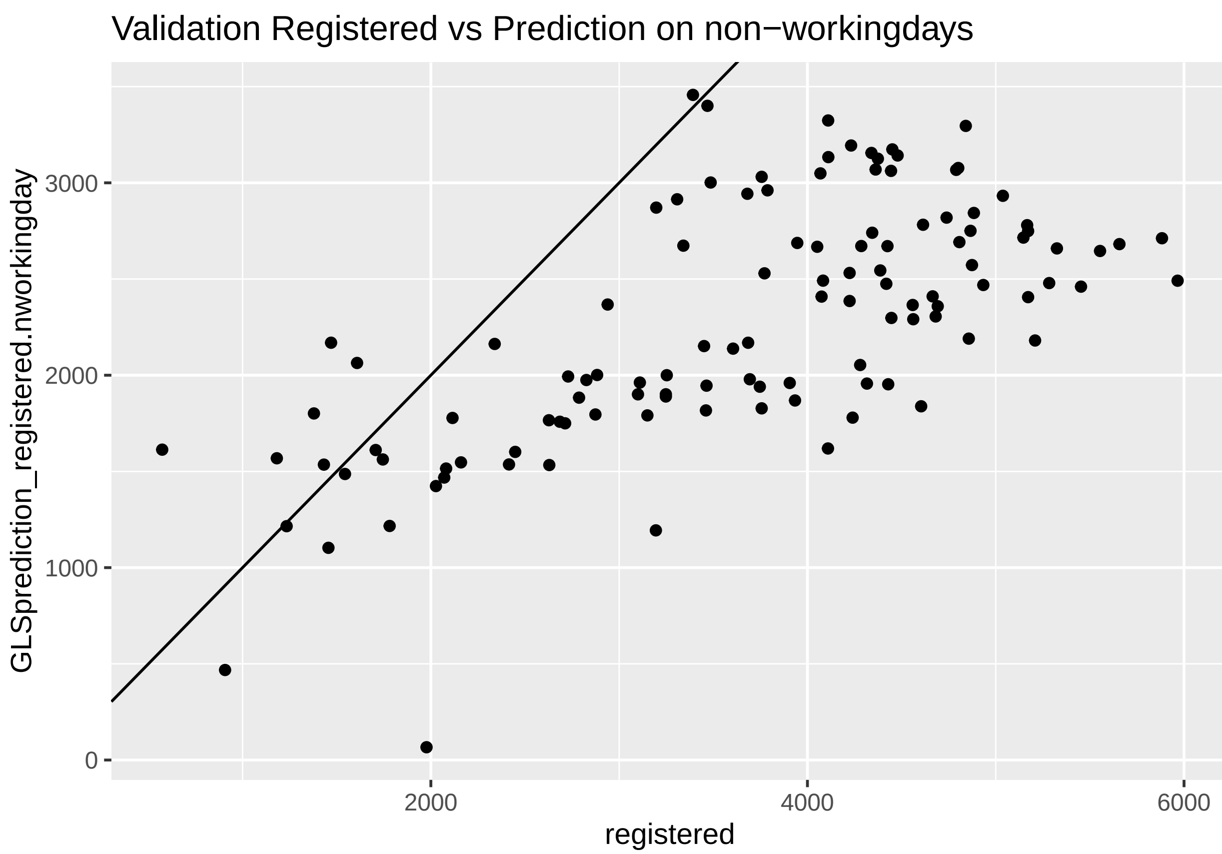
```r
ggplot(validate.nworkingday, aes(x = casual, y = GLSprediction_casual.nworkingday)) + geom_point() +
geom_abline(intercept = 0, slope = 1) +
ggtitle("Validation Casual vs Prediction onnon-workingdays")
```



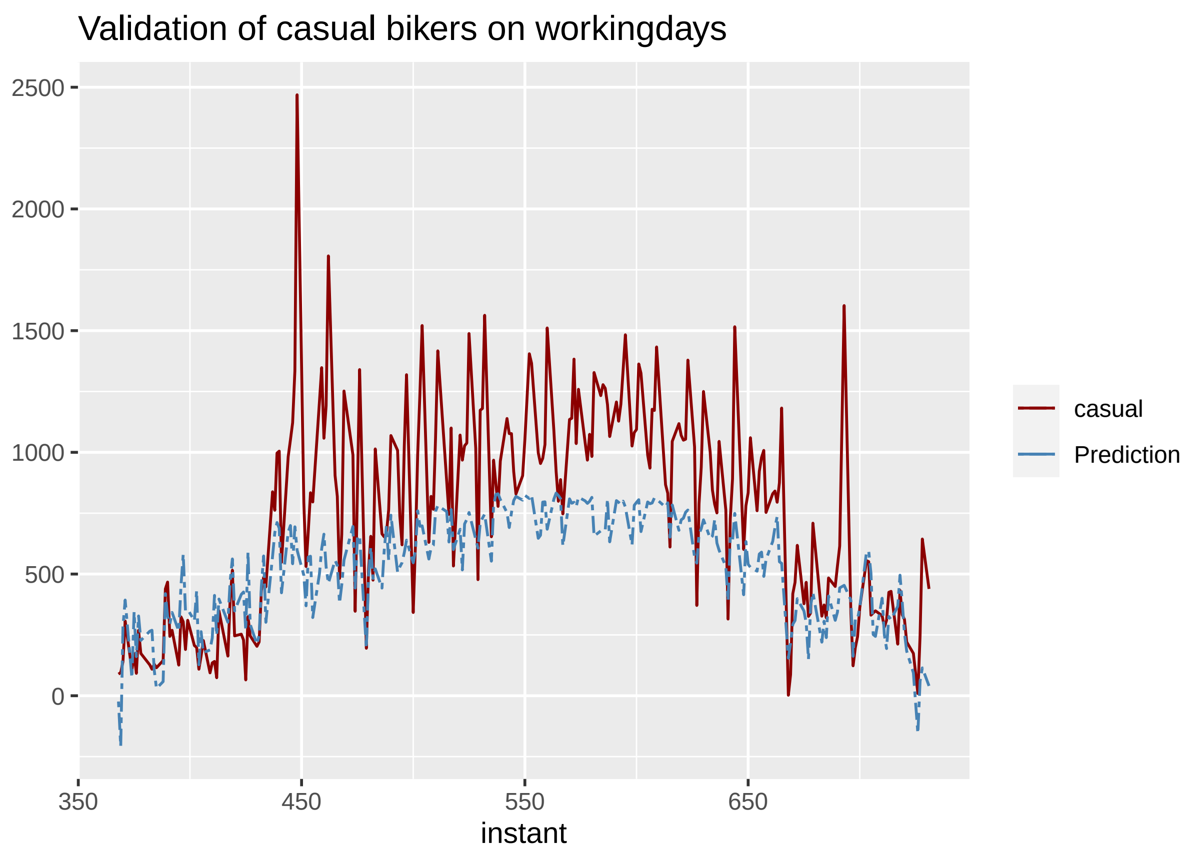

```r
ggplot(validate.workingday, aes(x = registered, y = GLSprediction_registered.workingday)) + geom_point(
geom_abline(intercept = 0, slope = 1) +
ggtitle("Validation Registered vs Prediction on workingdays")
```



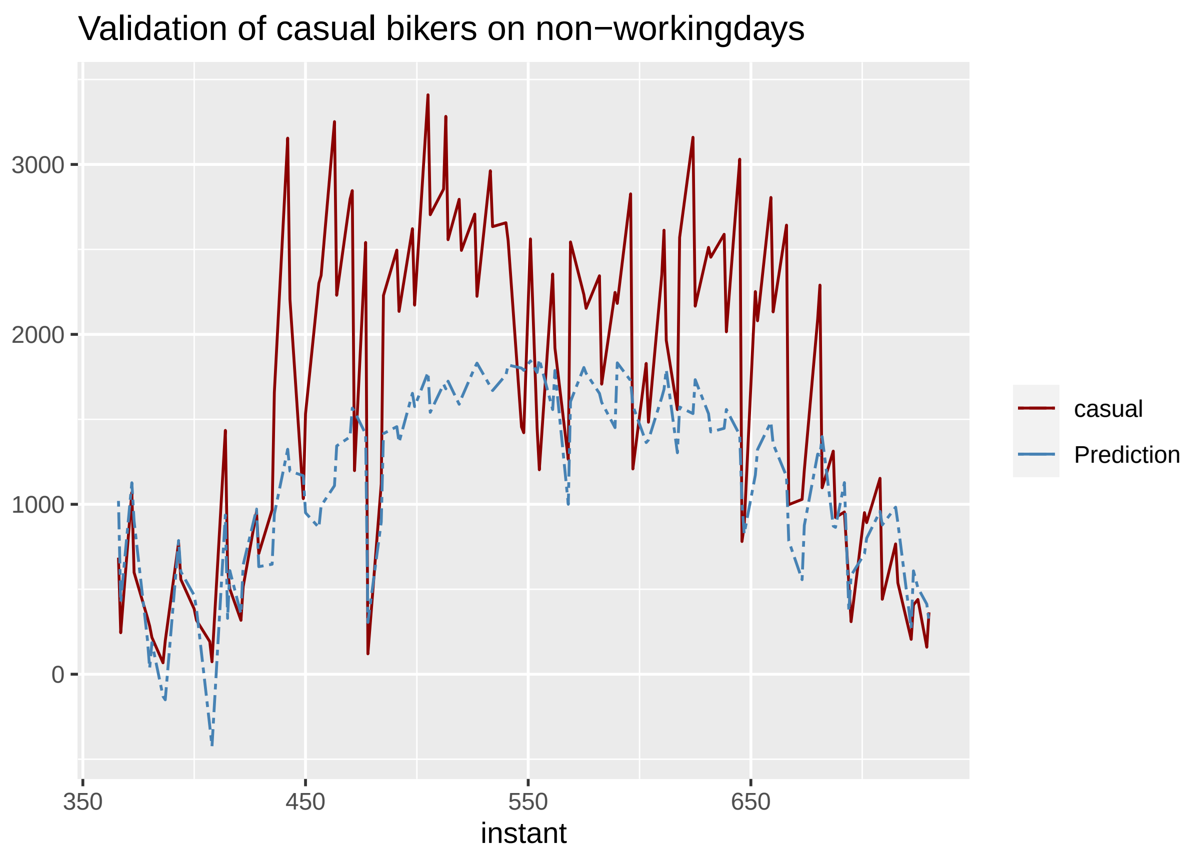

```
ggplot(validate.nworkingday, aes(x = registered, y = GLSprediction_registered.nworkingday)) + geom_poin
geom_abline(intercept = 0, slope = 1) +
ggtitle("Validation Registered vs Prediction on non-workingdays")
```



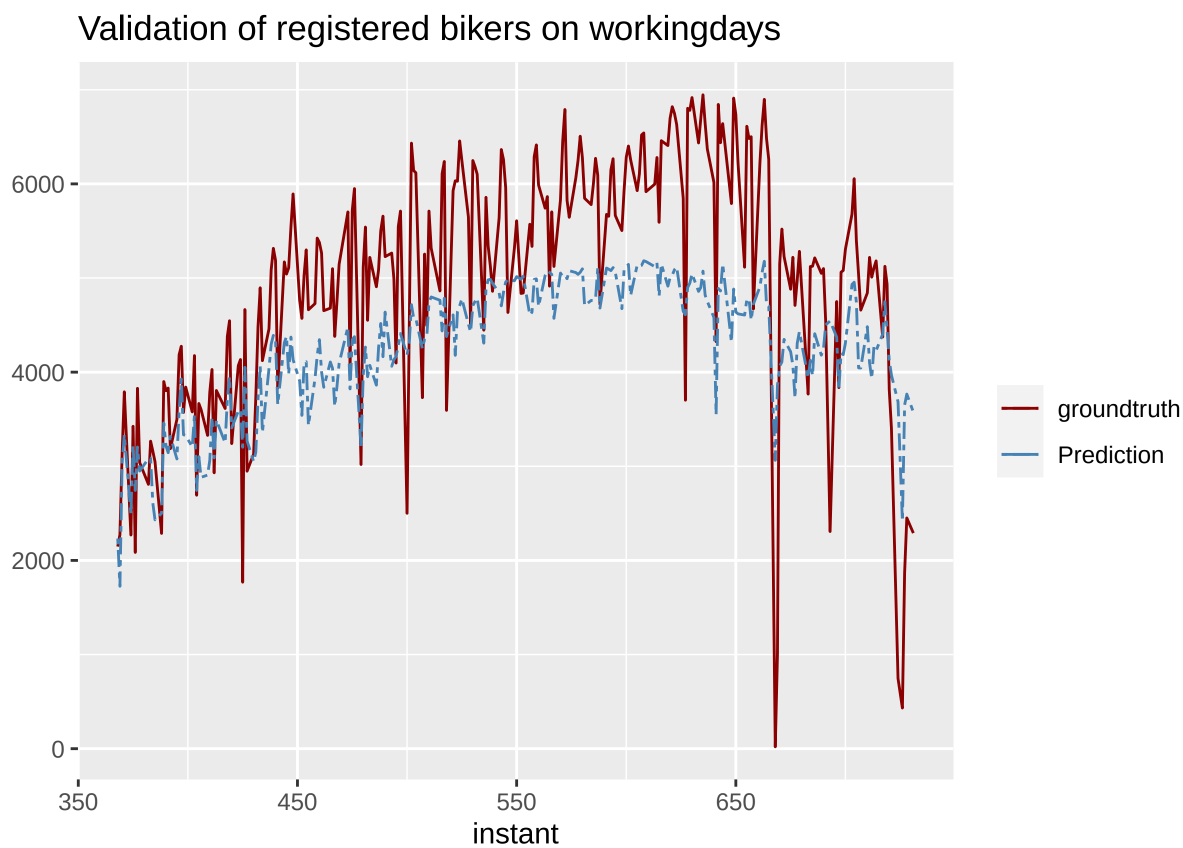

```
ggplot(data = validate.workingday, aes(x = instant)) +
geom_line(aes(y = casual, color = "casual")) +
geom_line(aes(y = GLSprediction_casual.workingday, color="Prediction"), linetype="twodash") +
scale_color_manual(name = element_blank(), labels = c("casual","Prediction"),
values = c("darkred", "steelblue")) + labs(y = "") +
ggtitle("Validation of casual bikers on workingdays")
```



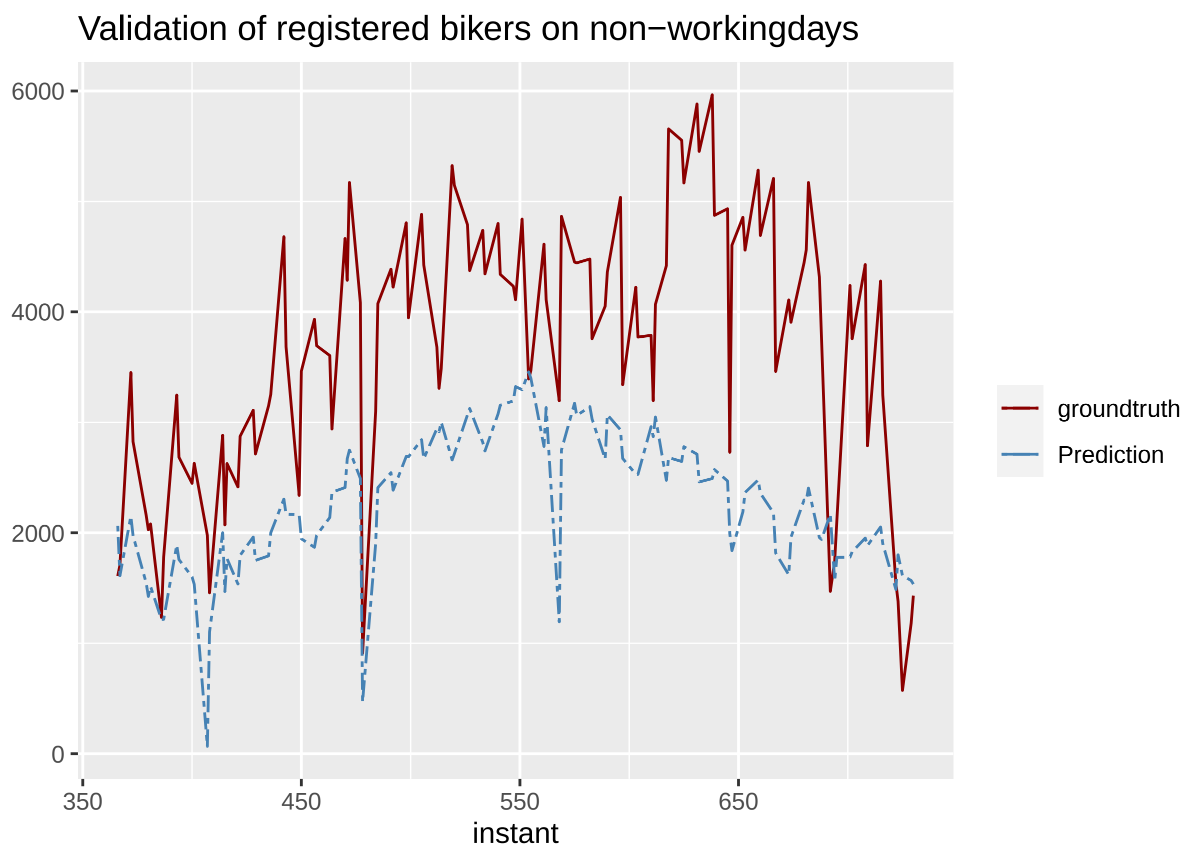

```
ggplot(data = validate.nworkingday, aes(x = instant)) +
geom_line(aes(y = casual, color = "casual")) +
geom_line(aes(y = GLSprediction_casual.nworkingday, color="Prediction"), linetype="twodash") +
scale_color_manual(name = element_blank(), labels = c("casual","Prediction"),
values = c("darkred", "steelblue")) + labs(y = "") +
ggtitle("Validation of casual bikers on non-workingdays")
```




```r
ggplot(data = validate.workingday, aes(x = instant)) +
geom_line(aes(y = registered, color = "groundtruth")) +
geom_line(aes(y = GLSprediction_registered.workingday, color="Prediction"), linetype="twodash") +
scale_color_manual(name = element_blank(), labels = c("groundtruth","Prediction"),
values = c("darkred", "steelblue")) + labs(y = "") +
ggtitle("Validation of registered bikers on workingdays")
```

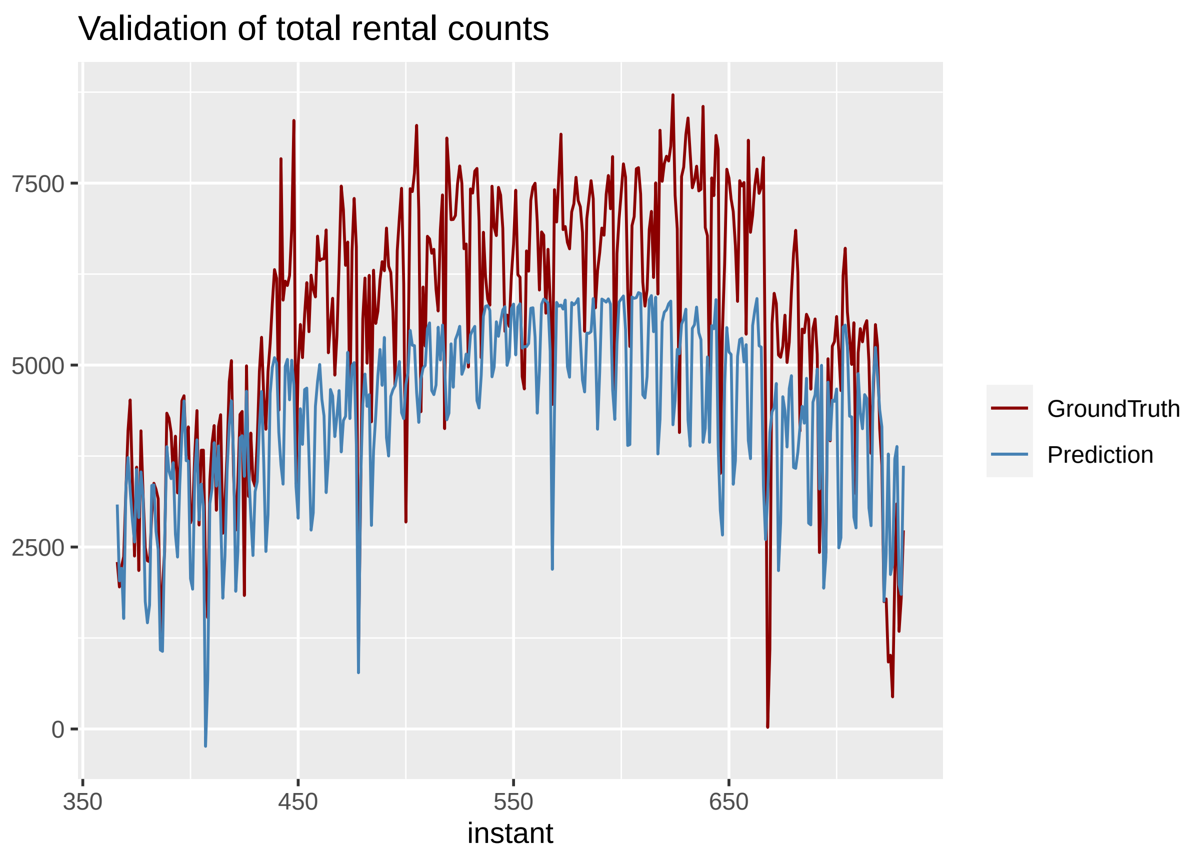



```r
ggplot(data = validate.nworkingday, aes(x = instant)) +
geom_line(aes(y = registered, color = "groundtruth")) +
geom_line(aes(y = GLSprediction_registered.nworkingday, color="Prediction"), linetype="twodash") +
scale_color_manual(name = element_blank(), labels = c("groundtruth","Prediction"),
values = c("darkred", "steelblue")) + labs(y = "") +
ggtitle("Validation of registered bikers on non-workingdays")
```




```
validate.nworkingday<- validate.nworkingday %>% mutate(GLSpred.total = GLSprediction_registered.nworkin

validate.workingday<- validate.workingday %>% mutate(GLSpred.total = GLSprediction_registered.workingda

temp1<- subset(validate.nworkingday, select = c(instant,GLSpred.total, cnt))
temp2<- subset(validate.workingday, select = c(instant,GLSpred.total, cnt))
GLStotal<- rbind(temp1, temp2)
```

```
ggplot(data = GLStotal, aes(x = instant)) +
geom_line(aes(y = cnt, color = "GroundTruth")) +
geom_line(aes(y = GLSpred.total, color="Prediction")) +
scale_color_manual(name = element_blank(), labels = c("GroundTruth","Prediction"),
values = c("darkred", "steelblue")) + labs(y = "") +
ggtitle("Validation of total rental counts")
```



Validation of total rental counts
7500
5000
2500
0
350
450
550
650
instant
GroundTruth
Prediction