Regression Analysis

Multiple Linear Regression

Nicoleta Serban, Ph.D.
Professor
School of Industrial and Systems Engineering

Predicting Demand for Rental Bikes: Regression Analysis

1

About This Lesson

2

Linear Regression Analysis in R

```
# Applying multiple linear regression model
model1 = lm(cnt ~ .,data=train)
summary(model1)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-79.4201	7.3917	-10.744	< 2e-16 ***
season2	41.7616	5.3578	7.794	6.93e-15 ***
season3	33.3129	6.3740	5.226	1.75e-07 ***
season4	67.2826	5.4338	12.382	< 2e-16 ***
yr1	86.2941	1.7468	49.401	< 2e-16 ***
mnth2	0.7558	4.3763	0.173	0.862893
mnth3	12.2441	4.9101	2.494	0.012655 *
mnth4	3.5236	7.2709	0.485	0.627950
mnth5	20.2297	7.7696	2.604	0.009233 **
mnth6	2.6876	7.9759	0.337	0.736150

⋮

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 101.6 on 13851 degrees of freedom
Multiple R-squared:  0.6869,    Adjusted R-squared:  0.6857
F-statistic: 595.7 on 51 and 13851 DF,  p-value: < 2.2e-16
```

In the full output there are 51 predictor rows in addition to the intercept.

$\hat{\sigma}$= 101.8

df = n-p-1 = 13,903 - 52 - 1 = 13,850

R^2 ≈ 0.6852 ≈ 68.5% variability explained

3

Coding Dummy Variables in R

```
## Create Dummy Variables
weathersit = data$weathersit
weathersit.1 = rep(0,length(weathersit))
weathersit.1[weathersit==1] = 1
weathersit.2 = rep(0,length(weathersit))
weathersit.2[weathersit==2] = 1
weathersit.3 = rep(0,length(weathersit))
weathersit.3[weathersit==3] = 1

## Include all dummy vars without intercept
fit.1 = lm(cnt ~ weathersit.1 + weathersit.2 +weathersit.3 - 1)

## Include 3 dummy variables with intercept
fit.2 = lm(cnt ~ weathersit.1 + weathersit.2)

## Use categorical variable
weathersit = as.factor(data$weathersit)
fit.3 = lm(cnt ~ weathersit)
```

summary(fit.1)

	Estimate	Std. Error	t value	Pr(>|t|)
weathersit.1	204.869	1.680	121.97	<2e-16 ***
weathersit.2	175.165	2.662	65.80	<2e-16 ***
weathersit.3	111.501	4.758	23.43	<2e-16 ***

summary(fit.2)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	111.501	4.758	23.43	<2e-16 ***
weathersit.1	93.369	5.046	18.50	<2e-16 ***
weathersit.2	63.665	5.452	11.68	<2e-16 ***

summary(fit.3)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	204.869	1.680	121.972	<2e-16 ***
weathersit2	-29.704	3.148	-9.437	<2e-16 ***
weathersit3	-93.369	5.046	-18.503	<2e-16 ***

4

Coding Dummy Variables in R

```
## Create Dummy Variables
weathersit  = data$weathersit
weathersit.1 = rep(0,length(weathersit))
weathersit.1[weathersit==1] = 1
weathersit.2 = rep(0,length(weathersit))
weathersit.2[weathersit==2] = 1
weathersit.3 = rep(0,length(weathersit))
weathersit.3[weathersit==3] = 1

## Include all dummy vars without intercept
fit.1 = lm(cnt ~ weathersit.1 + weathersit.2 +weathersit.3 - 1)

## Include 3 dummy variables with intercept
fit.2 = lm(cnt ~ weathersit.1 + weathersit.2)

## Use categorical variable
weathersit  = as.factor(data$weathersit)
fit.3 = lm(cnt ~ weathersit)
```

summary(fit.1)

	Estimate	Std. Error	t value	Pr(>|t|)
weathersit.1	204.869	1.680	121.97	<2e-16 ***
weathersit.2	175.165	2.662	65.80	<2e-16 ***
weathersit.3	111.501	4.758	23.43	<2e-16 ***

summary(fit.2)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	111.501	4.758	23.43	<2e-16 ***
weathersit.1	93.369	5.046	18.50	<2e-16 ***
weathersit.2	63.665	5.452	11.68	<2e-16 ***

summary(fit.3)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	204.869	1.680	121.972	<2e-16 ***
weathersit2	-29.704	3.148	-9.437	<2e-16 ***
weathersit3	-93.369	5.046	-18.503	<2e-16 ***

Codding Dummy Variables

R Sets the "first" class as being the baseline

- If a different class is the baseline, either use dummy variables or specify with 'contr.treatment'

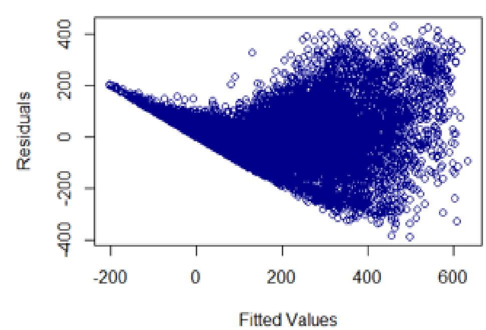
Be careful when using a model without intercept in R!

- No baseline comparison

Goodness of Fit: Constant Variance Assumption

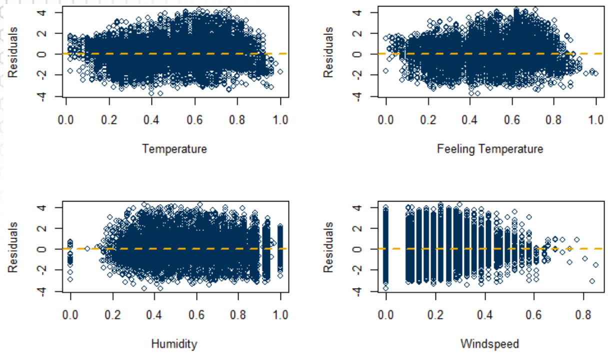
```
## Fitting the model
# Creating scatterplot of residuals vs fitted values

resids = rstandard(model1)
fits = model1$fitted
plot(fits,
     resids,
     xlab="Fitted Values",
     ylab="Residuals",
     main="Scatterplot",
     col="darkblue")
```



- The constant variance assumption does not hold -- the variance increases when moving from lower to higher fitted values.
- The residuals, at low y values, seem to follow a straight-line pattern. The linear pattern in the beginning suggests that the response variable stays constant for a range of predictor values.

Goodness of Fit: Linearity Assumption

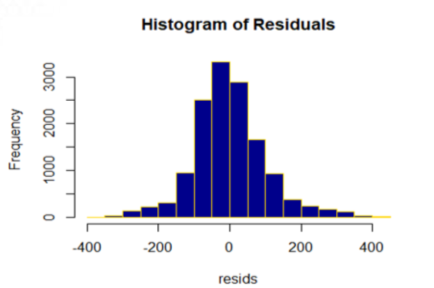


The residuals do not vary with the any of the numeric predicting variables. No transformation of the predicting variable is needed.

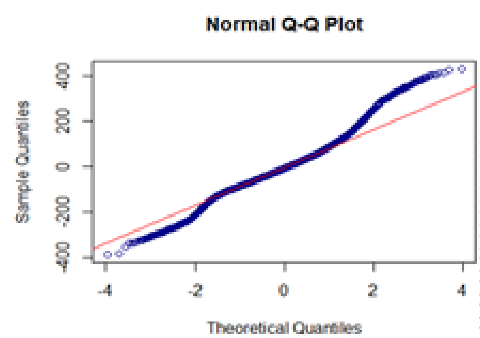
7

Goodness of Fit: Normality Assumption

```
## Checking normality
# histogram
hist(resids,
     nclass=20,
     col="darkblue",
     border="gold",
     main="Histogram of residuals")
```




```
# q-q plot
qqnorm(resids,
       col="darkblue")
qqline(resids,
       col="red")
```



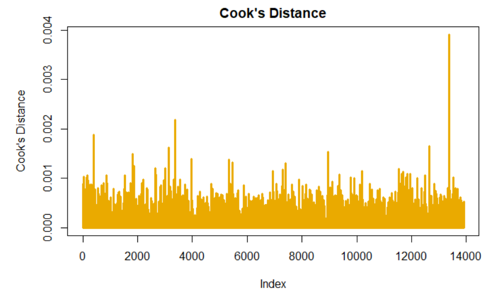
8

Goodness of Fit: Outliers

 # Cook's Distance

```
cook = cooks.distance(model1)

plot(cook,
     type="h",
     lwd=3,
     col="darkred",
     ylab = "Cook's Distance",
     main="Cook's Distance")
```



There is one observation with a Cook's Distance noticeably higher than the other observations. However, its Cook's distance is close to 0.004, suggesting that there are likely no outliers.

9

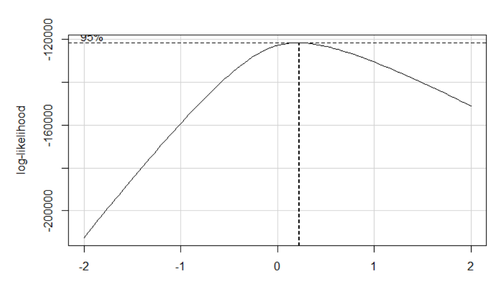
Transformation of the Response Variable

Box Cox transformation

```
bc <- boxcox(model1)
lambda <- bc$x[which(bc$y==max(bc$y))]
```

Fitting the model with square root transformation

```
model2<-lm(sqrt(cnt)~.,data=train)
summary(model2)
```



- The optimal value of lambda or the power provided by the Box Cox transformation is 0.22.
- Generally, when the response data consist of count data, a theoretically recommended transformation is the square root, corresponding to a 0.5 power transformation.

10

Regression Analysis after Transformation

Fitting the model with square root transformation

*model2<-**lm**(**sqrt**(cnt)~.,data=train)*
***summary**(model2)*

Find Insignificant Values

***which**(**summary**(model2)$coeff[,4]>0.05)*

mnth2	mnth4	mnth6	mnth7	mnth8	mnth10	mnth11	weekday1
6	8	10	11	12	14	15	41

Multicollinearity

***vif**(model2)*

	GVIF	Df	GVIF^(1/(2*Df))
season	165.308	3	2.343
yr	1.025	1	1.012
mnth	323.778	11	1.300
hr	1.771	23	1.012
holiday	1.121	1	1.059
weekday	1.137	6	1.011
weathersit	1.386	2	1.085
temp	51.283	1	7.161
atemp	43.748	1	6.614
hum	1.921	1	1.386
windspeed	1.251	1	1.118

Model Performance

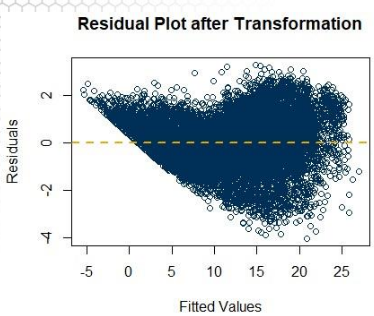
***summary**(model2)$r.squared*

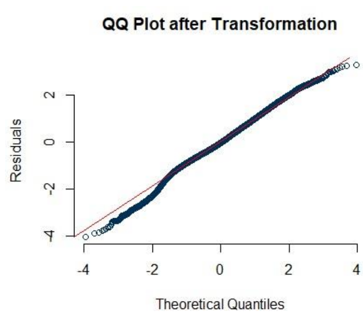
[1] 0.786535

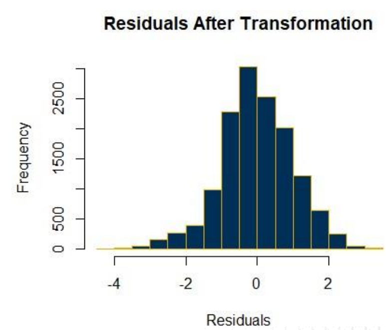
As VIFs of the season, mnth, temp, atemp factors are greater than max(10, 1/(1-R^2)), it indicates there is a problem of multicollinearity in the linear model. So, we should not use all the predictors in the model.

11

Goodness of Fit after Transformation







The constant variance assumption is still violated. The transformation has not improved the goodness of fit even though the model performance is better with respect to the coefficient of determination.

12

Removing Low Demand Data

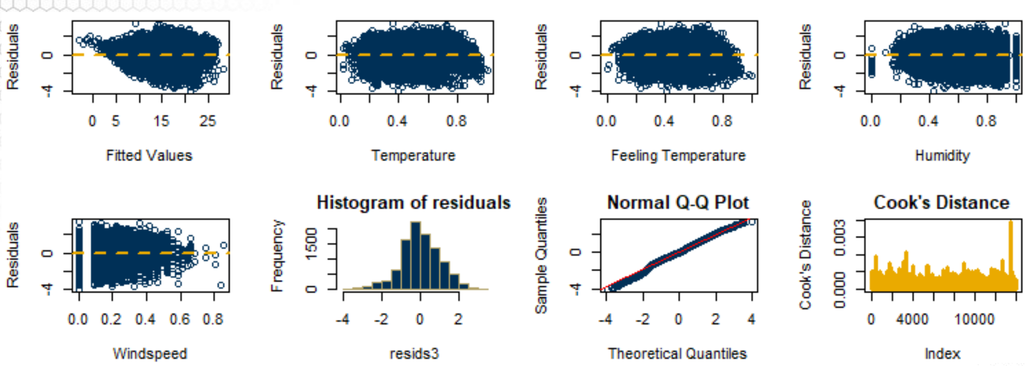
```
## Remove data for hours 0-6
hrs <- as.numeric(data$hr)
data_red <- data[which(hrs>=7),]
## Test/Train Data
set.seed(9) # for uniformity
sample_size <- floor(0.8*nrow(data_red))
picked <- sample(seq_len(nrow(data_red)),size = sample_size)
train_red <- data_red[picked, -c(1,2,9,15,16)]
test_red <- data_red[-picked, -c(1,2,9,15,16)]

## Fitting the model with square root transformation
model3<-lm(sqrt(cnt)~.,data=train_red)
summary(model3)$r.squared
[1] 0.6579021
df<-which(summary(model3)$coeff[,4]>0.05)

 mnth7 mnth11 mnth12  hr14  hr15  hr20
    11     15     16    23    24    29
```

13

Goodness of Fit without Low Demand Data



- The constant variance assumption is still violated even for the model without the low demand data and with the transformed response.
- The implication of the constant variation assumption violation is that the uncertainty in predicting bike demand when in high demand will be higher than estimated using the multiple regression models in this lesson.

14

Summary

Georgia Tech

15