Linear Regression Assumptions

Randy Johnson

Linear Regression Assumptions

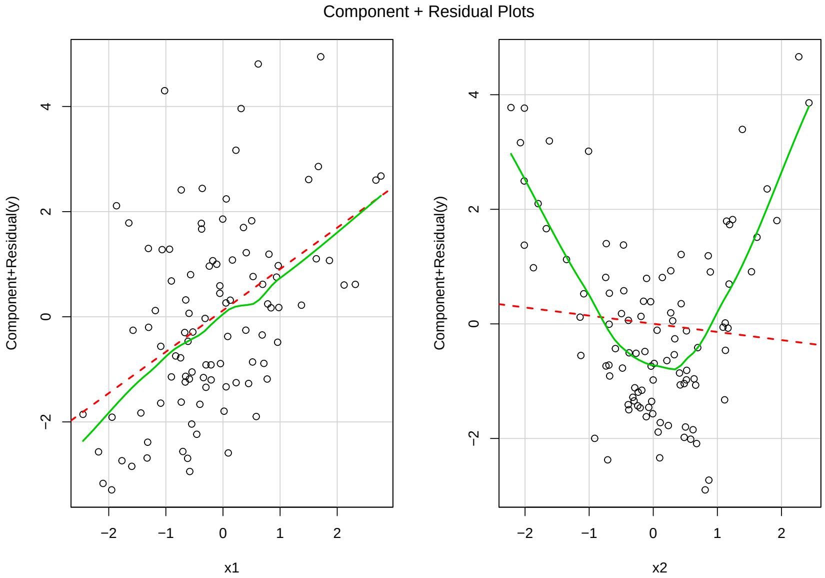
- Linear relationship
- Multivariate Normality
- No/little multicollinearity
- No autocorrelation
- Homoscedasticity

Linear Relationship

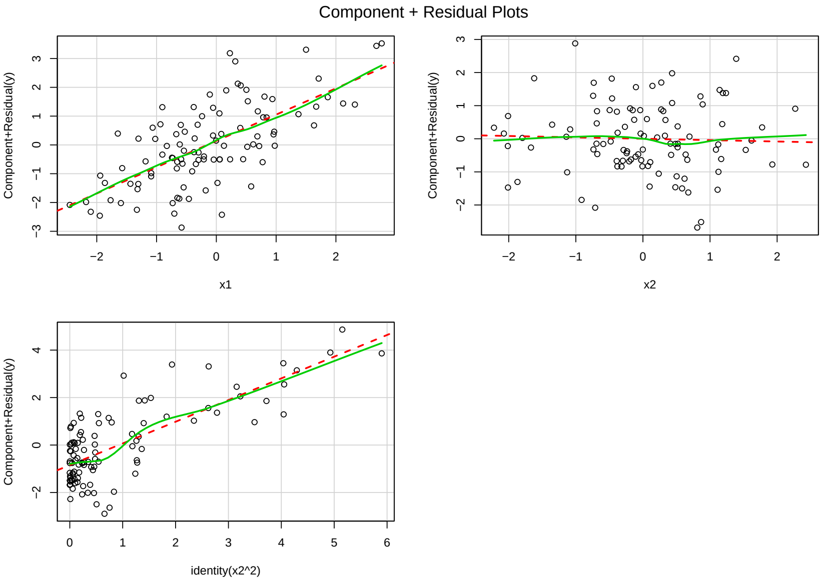
```r
require(dplyr)
set.seed(239478)
tmp <- data_frame(x1 = rnorm(100),
                  x2 = rnorm(100),
                  y = x1 + x2^2 + rnorm(100))
model1 <- lm(y ~ x1 + x2, data = tmp)
model2 <- lm(y ~ x1 + x2 + identity(x2^2), data = tmp)
```

```r
require(car)
crPlots(model1)
crPlots(model2)
```

Linear Relationship



Linear Relationship



Multivariate Normality

```r
set.seed(234987)
mvnExample <- data_frame(x1 = rnorm(100),
                         y1 = x1 + rnorm(100),
                         y2 = x1 + rt(100, 3))
mvnModel1 <- lm(y1 ~ x1, data = mvnExample)
mvnModel2 <- lm(y2 ~ x1, data = mvnExample)
```

Multivariate Normality

```r
require(broom)
with(augment(mvnModel1), shapiro.test(.std.resid))
```

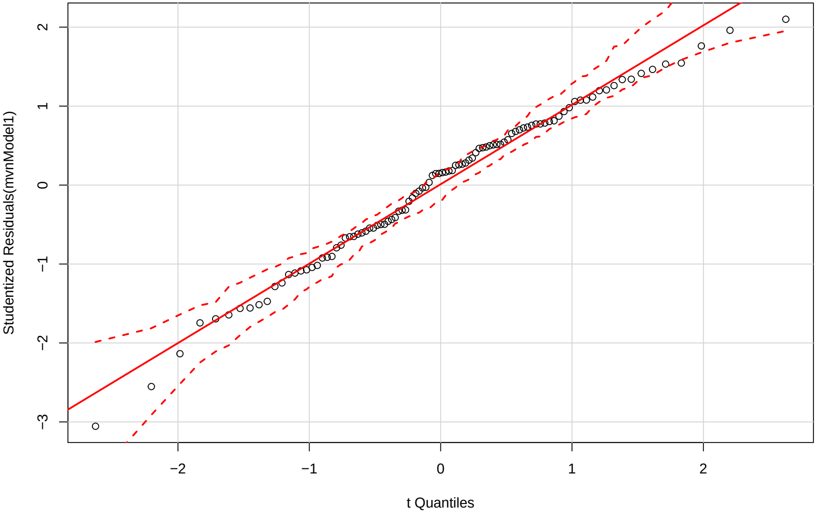
```
##
##  Shapiro-Wilk normality test
##
## data:  .std.resid
## W = 0.98544, p-value = 0.3417
```

```r
with(augment(mvnModel2), shapiro.test(.std.resid))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .std.resid
## W = 0.91781, p-value = 1.079e-05
```

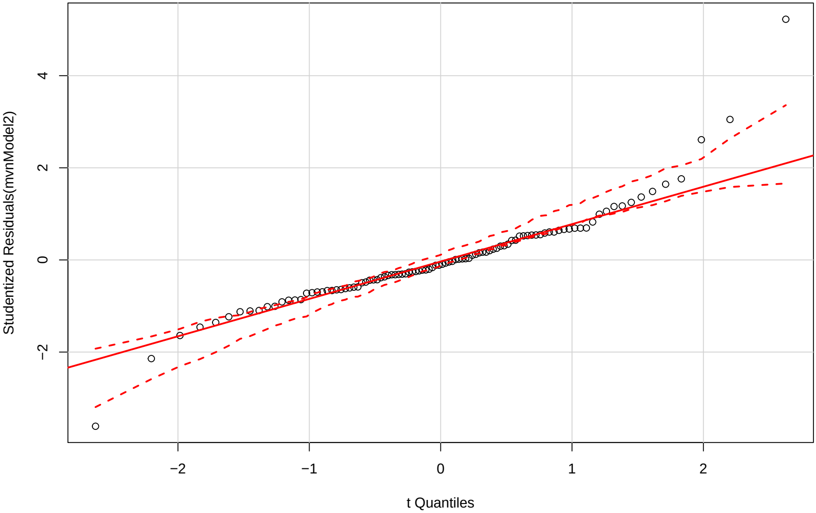
Multivariate Normality

```
qqPlot(mvnModel1)
```



Multivariate Normality

```
qqPlot(mvnModel2)
```



Little/no Multicollinearity

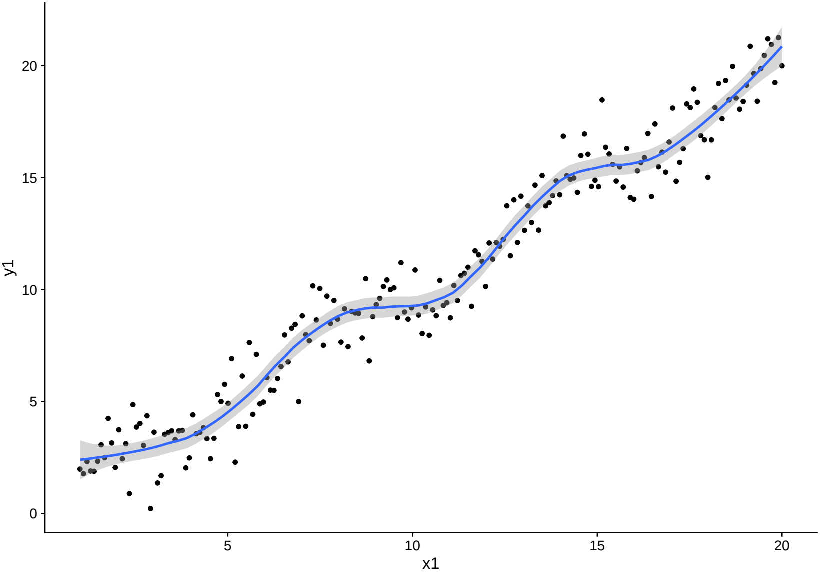
```
set.seed(239)
data_frame(x1 = rnorm(100),
           x2 = x1 + rnorm(100),
           x3 = rnorm(100),
           y = x1 + x2 + x3 + rnorm(100)) %>%
    lm(formula = y ~ x1 + x2 + x3) %>%
    vif()
```

```
##       x1       x2       x3
## 2.271759 2.271338 1.005520
```

No Autocorrelation

```
set.seed(2347890)
dat <- data_frame(x1 = seq(1, 20, length = 200),
                  y1 = sin(x1) + x1 + rnorm(200))

g <- ggplot(dat, aes(x1, y1)) +
     geom_point() +
     geom_smooth(span = .3, method = 'loess')
```

No Autocorrelation



No Autocorrelation

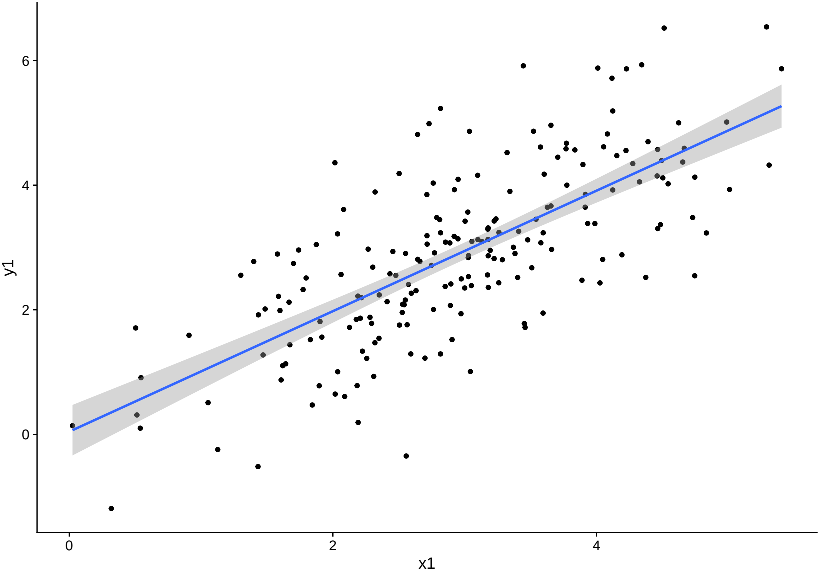
```
lm(y1 ~ x1, data = dat) %>%
    durbinWatsonTest()
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.1940815      1.609687   0.002
##  Alternative hypothesis: rho != 0
```

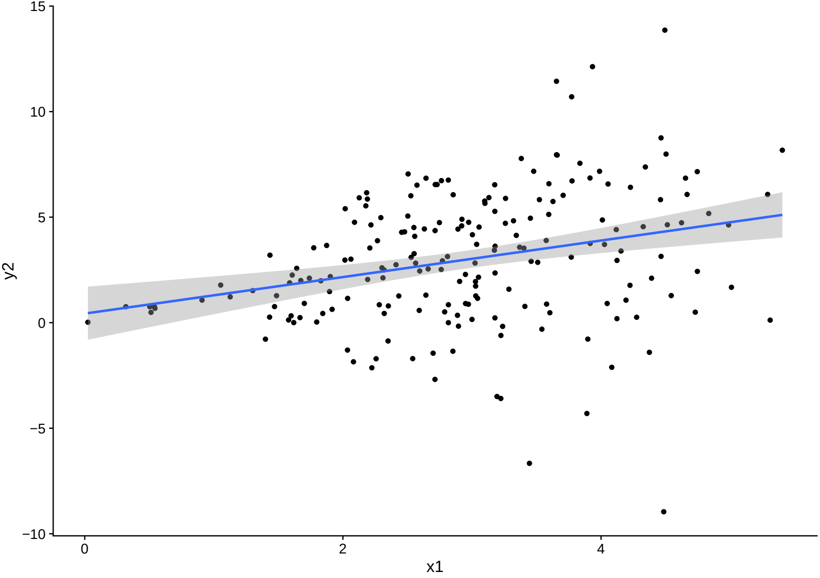
Homoscetasticity

```r
set.seed(234897)
dat <- data_frame(x1 = rnorm(200) + 3,
                  y1 = x1 + rnorm(200),
                  y2 = x1 + rnorm(200)*x1)

g1 <- ggplot(dat, aes(x1, y1)) +
      geom_point() +
      geom_smooth(method = 'lm')

g2 <- ggplot(dat, aes(x1, y2)) +
      geom_point() +
      geom_smooth(method = 'lm')
```

Homoscedasticity

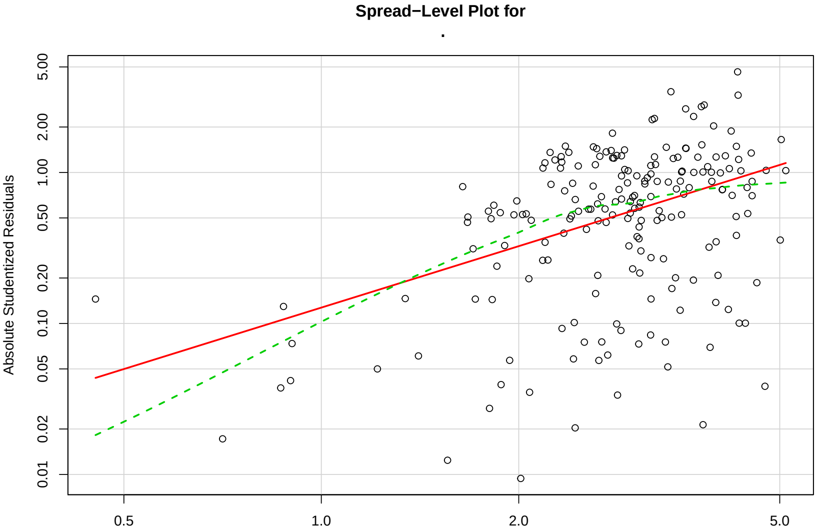


Homoscedasticity



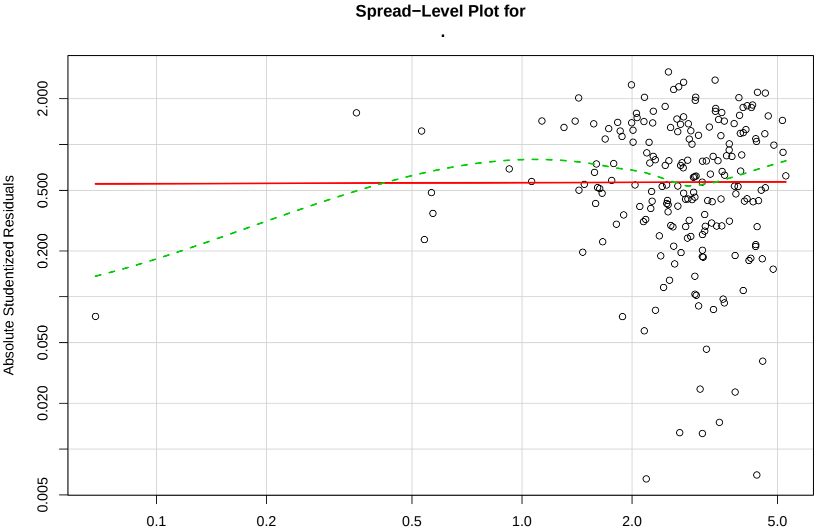
Homoscedasticity

```
lm(y2 ~ x1, data = dat) %>%
    spreadLevelPlot()
```



Homoscedasticity

```
lm(y1 ~ x1, data = dat) %>%
    spreadLevelPlot()
```



Homoscedasticity

```r
lm(y1 ~ x1, data = dat) %>%
    ncvTest()
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.05980537    Df = 1     p = 0.8068038
```

```r
lm(y2 ~ x1, data = dat) %>%
    ncvTest()
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 28.80365    Df = 1     p = 8.010014e-08
```

Homoscedasticity

```r
lm((y2^(1/4)) ~ x1, data = dat) %>%
    ncvTest()
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.652509    Df = 1     p = 0.1986178
```