Diagnostics and Assumptions

Adam J Sullivan, PhD

02/07/2018

Assumptions of Linear Regression

- ***Linearity:*** Function f is linear.
- Mean of error term is 0.

$$E(\varepsilon) = 0$$

- ***Independence:*** Error term is independent of covariate.

$$Corr(X, \varepsilon) = 0$$

- ***Homoscedacity:*** Variance of error term is same regardless of value of X.

$$Var(\varepsilon) = \sigma^2$$

- ***Normality:*** Errors are normally Distributed

Diagnostics for Linear Regression

- A remarkable paper came out in 1973 called by Francis J. Anscombe.
- We will explore this data as we discuss diagnostics and transformations for simple linear regression.
- These examples show how much your regression output may mislead you if you are not careful about the assumptions.

The Data

```
anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

The Model Set ups

- Notice in this data that for X_1 - X_3 we have the same data points but all of the Y values differ.
- Let us first consider their regression outputs from R:

```
mod1 <- lm(y1 ~ x1, data=anscombe)
mod2 <- lm(y2 ~ x2, data=anscombe)
mod3 <- lm(y3 ~ x3, data=anscombe)
mod4 <- lm(y4 ~ x4, data=anscombe)
```

Coefficients

```
library(broom)
library(dplyr)
tidy1 <- tidy(mod1, conf.int = T)
tidy2 <- tidy(mod2, conf.int = T)
tidy3 <- tidy(mod3, conf.int = T)
tidy4 <- tidy(mod4, conf.int = T)

knitr::kable(bind_rows(tidy1,tidy2, tidy3, tidy4)[,-c(3,4)])
```

Coefficients

term	estimate	p.value	conf.low	conf.high
(Intercept)	3.0000909	0.0257341	0.4557369	5.5444449
x1	0.5000909	0.0021696	0.2333701	0.7668117
(Intercept)	3.0009091	0.0257589	0.4552982	5.5465200
x2	0.5000000	0.0021788	0.2331475	0.7668525
(Intercept)	3.0024545	0.0256191	0.4587013	5.5462078
x3	0.4997273	0.0021763	0.2330695	0.7663851
(Intercept)	3.0017273	0.0255904	0.4592412	5.5442134
x4	0.4999091	0.0021646	0.2333841	0.7664341

Model Stats

```
glance1 <- glance(mod1)
glance2 <- glance(mod2)
glance3 <- glance(mod3)
glance4 <- glance(mod4)

knitr::kable(bind_rows(glance1,glance2, glance3, glance4))
```

Model Stats

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.res
0.6665425	0.6294916	1.236603	17.98994	0.0021696	2	-16.84069	39.68137	40.87506	13.76269	
0.6662420	0.6291578	1.237214	17.96565	0.0021788	2	-16.84612	39.69224	40.88593	13.77629	
0.6663240	0.6292489	1.236311	17.97228	0.0021763	2	-16.83809	39.67618	40.86986	13.75619	
0.6667073	0.6296747	1.235696	18.00329	0.0021646	2	-16.83261	39.66522	40.85890	13.74249	

What do we notice?

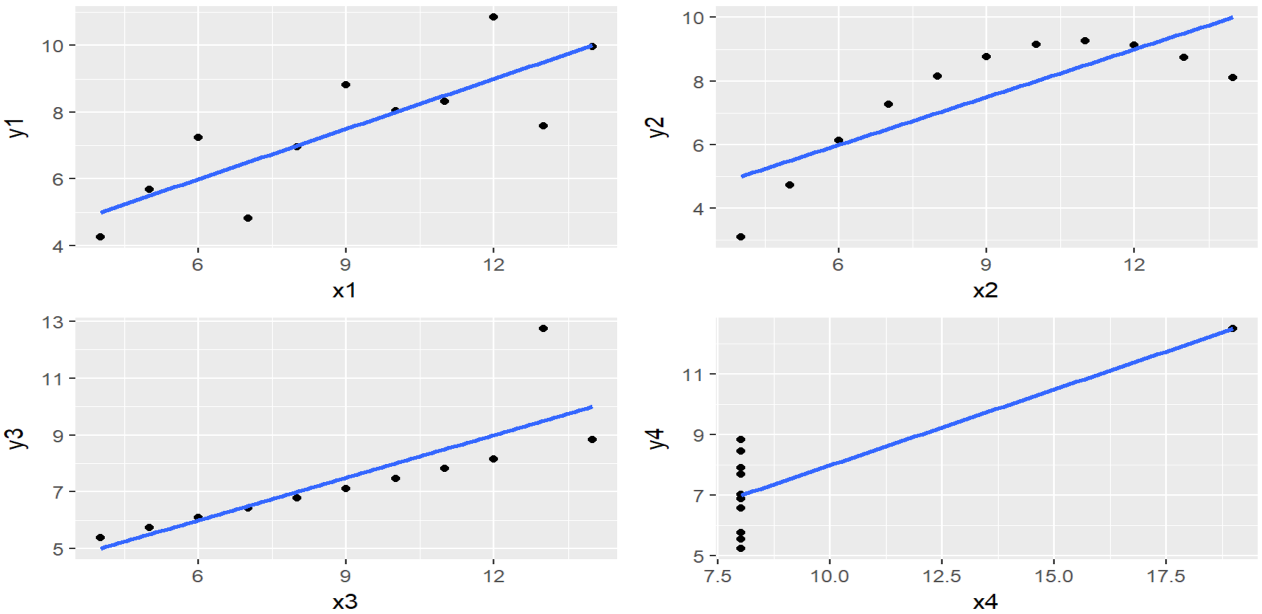
- All of the regression coefficients are the same.
- All of the regression diagnostics are the same.
- They appear to be the same models.

What About the Assumptions

- Lets check our model plots.

```
library(ggplot2)
library(gridExtra)
p1 <- ggplot(anscombe, aes(x1,y1)) + geom_point() + geom_smooth(method="lm", se=FALSE)
p2 <- ggplot(anscombe, aes(x2,y2)) + geom_point() + geom_smooth(method="lm", se=FALSE)
p3 <- ggplot(anscombe, aes(x3,y3)) + geom_point() + geom_smooth(method="lm", se=FALSE)
p4 <- ggplot(anscombe, aes(x4,y4)) + geom_point() + geom_smooth(method="lm", se=FALSE)
grid.arrange(p1,p2,p3, p4, ncol=2)
```

What About the Assumptions



What Do we Notice?

- We can see that the line looks appropriate for model 1 but not for the other 3.
- For Model 2 it appears the data is curved.
- For model 3 it appears that an outlier is really driving the model.
- Finally for Model 4 it appears that we have only one differing X value and that is driving the slope.
- What are the values of these regression lines?

What Does this Mean?

- If we look at these closely we can see that these are almost the exact same regressions.
- This is a major issue for us if we just blindly ran our regressions.
- We need to discuss tools to help us not make these mistakes.

Residuals to the Rescue

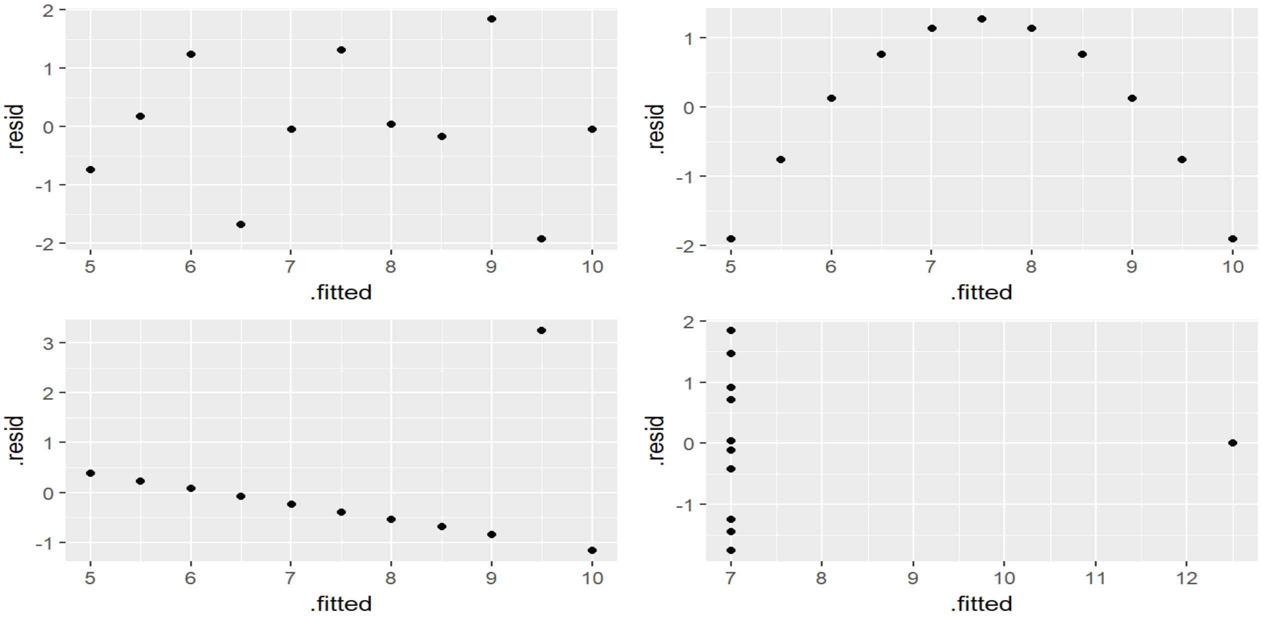
Residuals to the Rescue!!

- One method to help us evaluate a linear fit and to check assumptions is to consider the residuals.
- The benefit of examining the residuals is that unlike the plots previously is that we can evaluate them regardless of how many predictors are in the model.

Enter Residual Plots

```
library(ggplot2)
p1 <- ggplot(mod1, aes(.fitted, .resid)) + geom_point()
p2 <- ggplot(mod2, aes(.fitted, .resid)) + geom_point()
p3 <- ggplot(mod3, aes(.fitted, .resid)) + geom_point()
p4 <- ggplot(mod4, aes(.fitted, .resid)) + geom_point()
grid.arrange(p1,p2,p3, p4, ncol=2)
```

Enter Residual Plots



What do we see?

- Looking at the plots we can see that our residuals take on different patterns.
- This is due to how we defined residual error as

$$\varepsilon_i = Y_i - \beta_0 - \beta_1 X_i.$$

- We first look at Model 1 there is no pattern to these residuals and they seem to be randomly spread around 0.
- This is indicator of a good linear fit. Recall that we assume that $E(\varepsilon_i) = 0$, so we would expect to see the residuals spread around 0 and without pattern.

Patterns in Residuals

- Patterns in residuals show us that our model is not an adequate summary of the data.
- Consider what happens when our line is truly linear in nature then

$$Y_i = E(Y_i|X_i = x_i) + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

- We then fit our regression line $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$. This leads to the residuals

$$\hat{\varepsilon}_i = y_i - \hat{y}_i = \beta_0 + \beta_1 x_i + \varepsilon_i - \hat{\beta}_0 - \hat{\beta}_1 x_i \approx \varepsilon_i$$

- So the residuals are randomly distributed and centered about 0.

Quadratic Patterns in Residuals

- In the second figure, we can see that we have a quadratic pattern in this.
- This happens when the true model is quadratic

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$$

then we again fit our linear model $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

- This leads to the residuals

$$\hat{\varepsilon}_i = y_i - \hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i - \hat{\beta}_0 - \hat{\beta}_1 x_i \approx \beta_2 x_i^2 + \varepsilon_i$$

- So we have a quadratic relationship given our x.

Quadratic Patterns in Residuals

- This means we may have been better off by choosing a model that would include a quadratic term for x.
- In model 2 of Anscombe's data had we run a model with a quadratic term we would then have

```
anscombe$x2sq <- anscombe$x2^2
mod2a <- lm(y2 ~ x2 +  x2sq, data=anscombe)
tidy(mod2a, conf.int=T)[,-c(3,4)]
glance(mod2a)

ggplot(mod2a, aes(.fitted, .resid)) + geom_point()
```

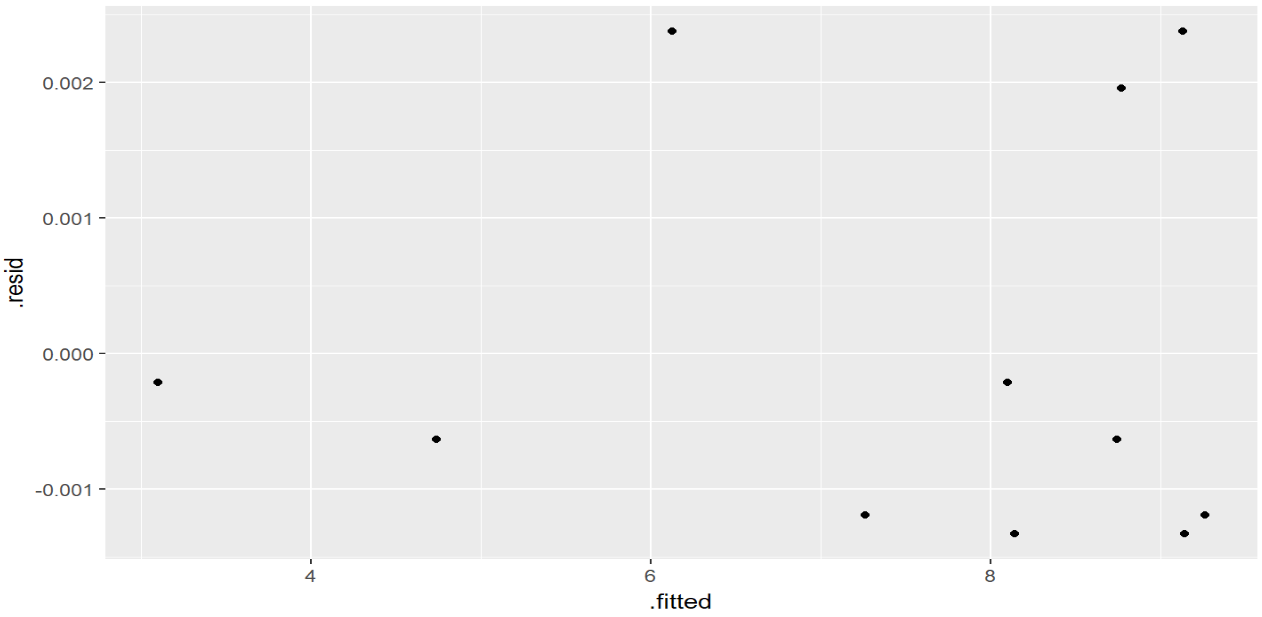
Quadratic Patterns in Residuals

term	estimate	p.value	conf.low	conf.high
(Intercept)	-5.9957343	0	-6.005719	-5.9857494
x2	2.7808392	0	2.778441	2.7832375
x2sq	-0.1267133	0	-0.126845	-0.1265816

Quadratic Patterns in Residuals

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.resic
0.9999995	0.9999993	0.0016725	7378133	0	3	56.47107	-104.9421	-103.3506	2.24e-05	

Quadratic Patterns in Residuals



Transformations

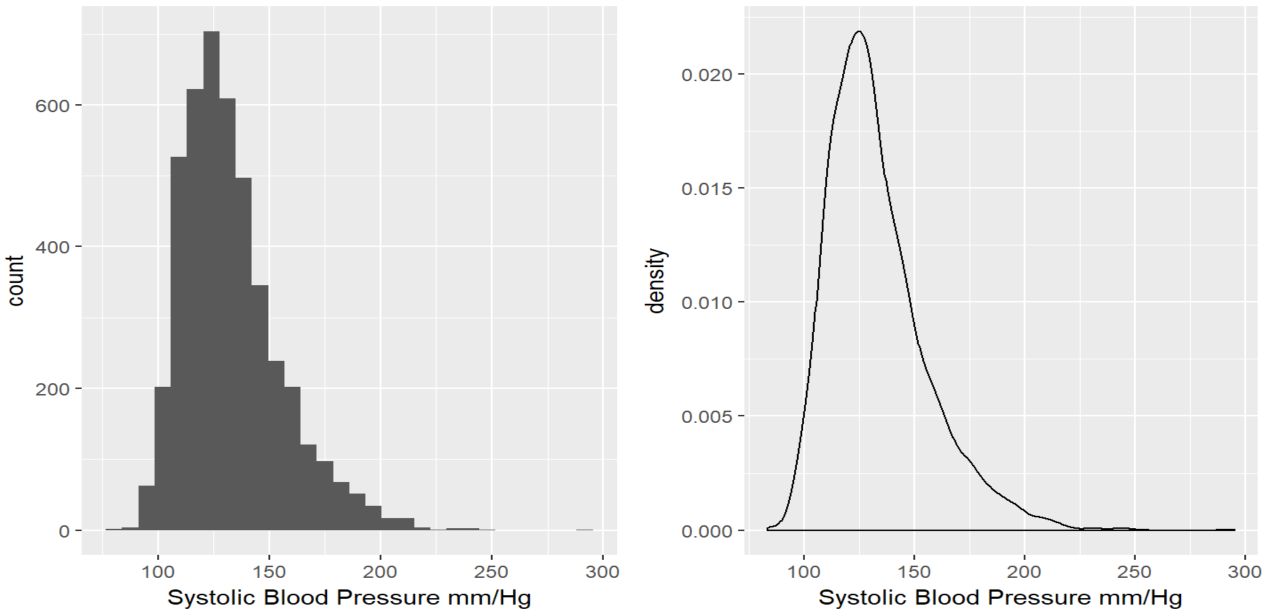
- When we run into issues with some sort of pattern in the residuals it can often help to transform the data in order to run a model over this transformation.
- One popular method is a log transform.
- We will consider this in R.

How do We know what to Do?

- We will consider the framingham heart study data for this.

```
library(haven)
fhs <- read_dta("fhs.dta")
```

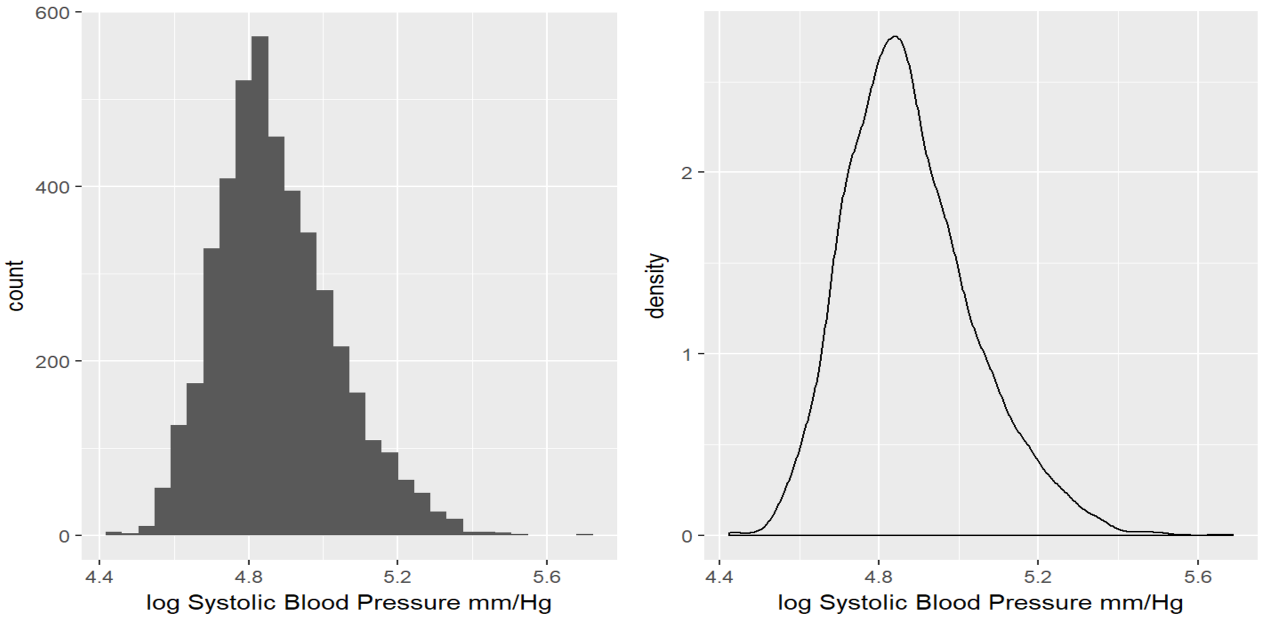
Systolic Blood Pressure in FHS



What do we see?

- This looks skewed so we can try and determine the best transformation so that this is normally distributed.
- Sometimes a log transform helps to pull in skewed values.

What do we see?



Model with Regular Data

- We can then run this transformed variable back into the model we had for this.
- Consider the model of Systolic Blood Pressure on BMI considering age as a confounder as well.

```
mod <- lm(sysbp1 ~ bmi1 + age1, data=fhs)
```

Model Diagnostics and Coefficients

term	estimate	p.value	conf.low	conf.high
(Intercept)	47.1359586	0	42.4701578	51.8017595
bmi1	1.5253813	0	1.3833394	1.6674232
age1	0.9281535	0	0.8609144	0.9953927

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residua
0.2346825	0.2343355	19.5657	676.4637	0	3	-19392.34	38792.68	38818.26	1688986	441

Regression on Transformed data

term	estimate	p.value	conf.low	conf.high
(Intercept)	4.2589597	0	4.2258423	4.2920770
bmi1	0.0110897	0	0.0100815	0.0120979
age1	0.0066283	0	0.0061510	0.0071056

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residu
0.2394446	0.2390998	0.138875	694.5117	0	3	2452.895	-4897.79	-4872.219	85.09104	441

What do we notice?

- When we see this latter model it appears that there is a large difference in coefficients.
- We need to remember that the coefficient for BMI in this case reflects an average change in the transformed Y.
- So in our case we see that for people of the same age a one unit increase in BMI leads to a 0.0110897 increase in log systolic blood pressure.
- This really does not have any meaning for us.
- **NEVER interpret regression coefficients in the context of a transformed variable**.

Interpreting Transformed Regressions

- Instead you must go back and find out what a one unit increase in BMI will do for systolic blood pressure.
- We need to consider the math of what is happening.
- We know that:

$$\begin{aligned}\log(\mu_Y) &= \beta_0 + \beta_1 X \\ \mu_Y &= \exp(\beta_0 + \beta_1 X) \\ \mu_Y &= e^{\beta_0} e^{\beta_1 X}\end{aligned}$$

Interpreting Transformed Regressions

- Then consider a change in μ_Y by a change in X:

$$\begin{aligned}\frac{\partial \mu_Y}{\partial X} &= \beta_1 e^{\beta_0} e^{\beta_1 X} \\ &= \beta_1 * \mu_Y\end{aligned}$$

- Thus a one unit change in X leads to β_1 times μ_y, so if $\beta_1 = 0.3$ we would see a μ_Y change by $0.3\mu_Y$ or a 30% increase.

Interpreting Transformed Regressions

- This shows that for those who are the same age a one unit increase in BMI leads to an increase in systolic blood pressure of 1.1%.

Tools for Checking Validity of a Model

When fitting a regression model we will take these steps to verify the validity of the model:

1. Regression Model is Linear in parameters.
2. Residuals are normally distributed.
3. Mean of Residuals is 0.
4. Homoscedasticity of variances.
5. Variables and residuals are not correlated.
6. No Influential Points or Outliers

Linear in Parameters

- We say it is linear in parameters if the β values are linear in nature.
- Consider the 2nd Anscombe model:

$$E[Y|x] = \beta_0 + \beta_1 x + \beta_2 x^2$$

- Even though `x2` has been transformed to a square term, the $beta$ values are still linear.

Standardized Residuals

- When we are concerned with residuals, everything is in context with the original values of the problem.
- A standardized residual takes into account the standard deviation of the residual

$$\text{Standardized } \hat{\varepsilon}_i = \frac{\hat{\varepsilon}_i}{\boldsymbol{sd}\left(\hat{\varepsilon}_i\right)}$$

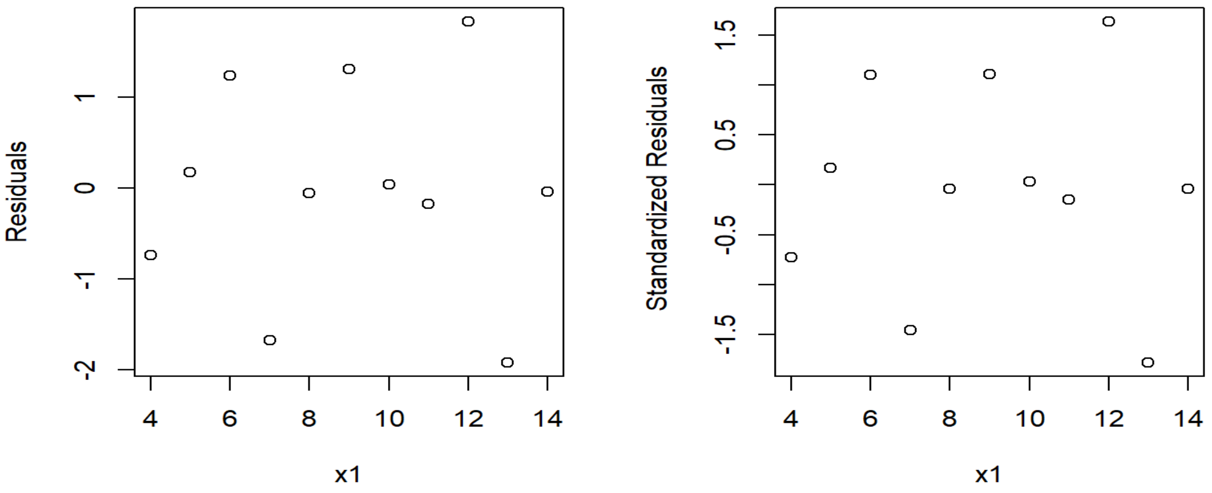
Why Standardize?

- This way we already assumed that $E(\varepsilon) = 0$ and the $var(\varepsilon) = \sigma^2$.
- By standardizing we now have a mean of 0 still but this time a variance of 1.
- This means that with the residuals being roughly normal we expect to see about 95% of our residuals falling between ± 2.

Standardized Residuals Plots

```
par(mfrow=c(1,2))
plot(anscombe$x1, resid(mod1), xlab="x1", ylab="Residuals")
plot(anscombe$x1, rstandard(mod1),xlab="x1", ylab=" Standardized Residuals")
```

Standardized Residuals Plots

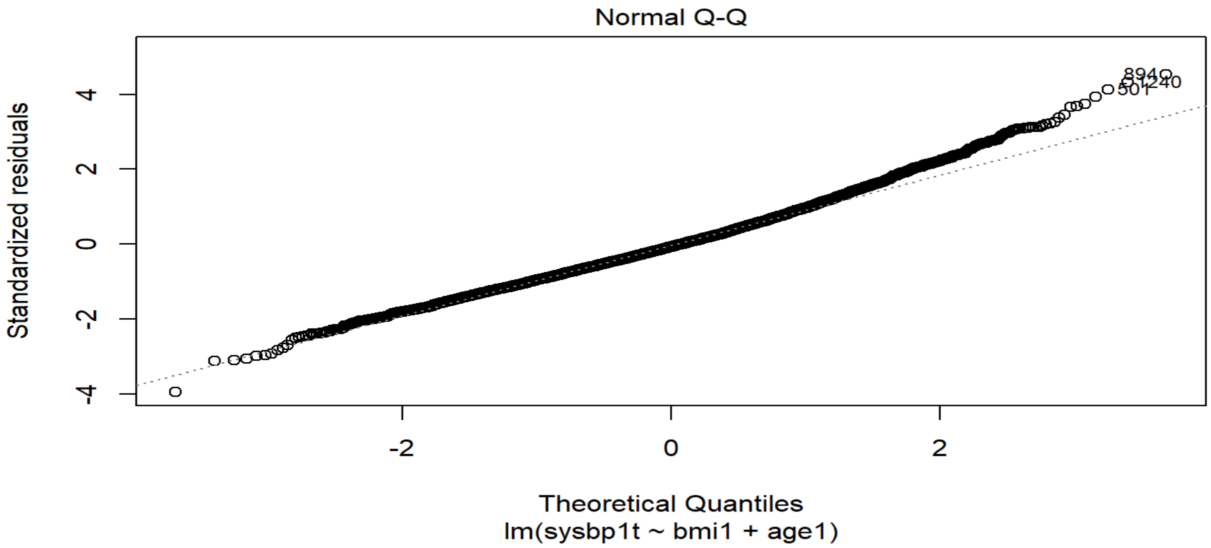


Standardized Residuals Plots

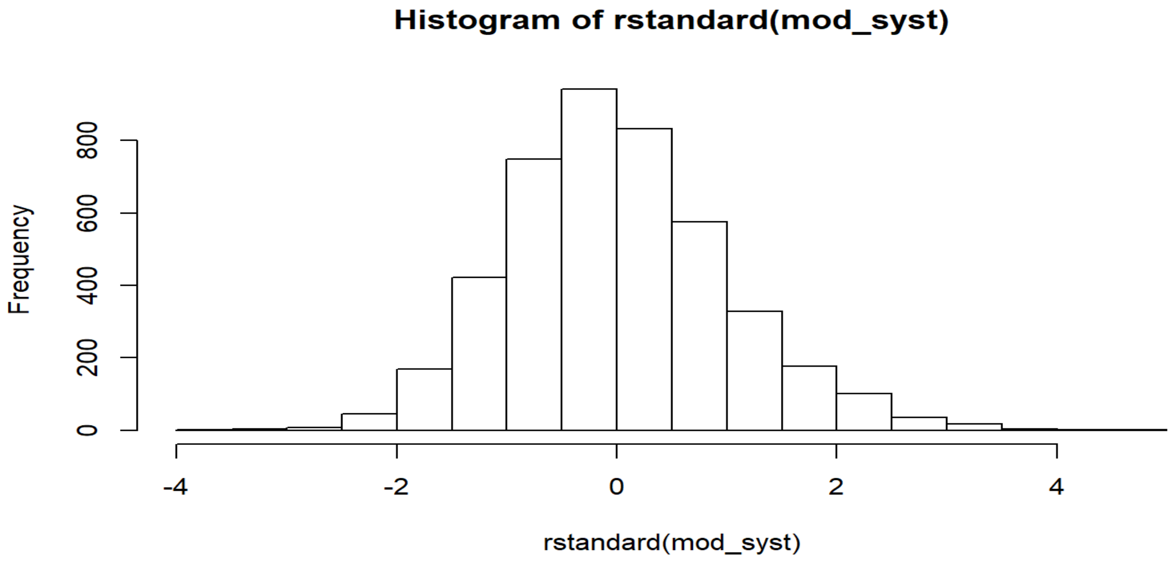
- The residuals shows us that the residuals did not change by much and we can still see the pattern is the exact same as before but the range of the residuals is what has changed.

Assessing Normality of Residuals: QQ-Plot

- Recall our Transformed Systolic Blood Pressure.



Assessing Normality of Residuals: Histogram



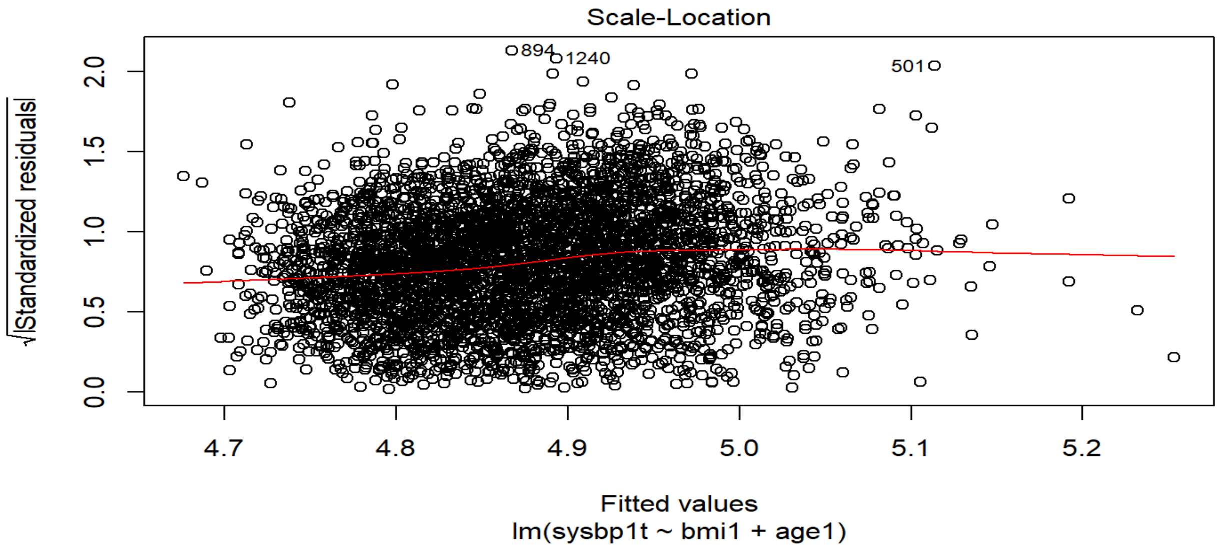
Mean of Residuals

- We can test if the mean of residuals is zero with a simple mean function.

```
mean(mod_syst$residuals)
```

```
## [1] -1.101276e-18
```

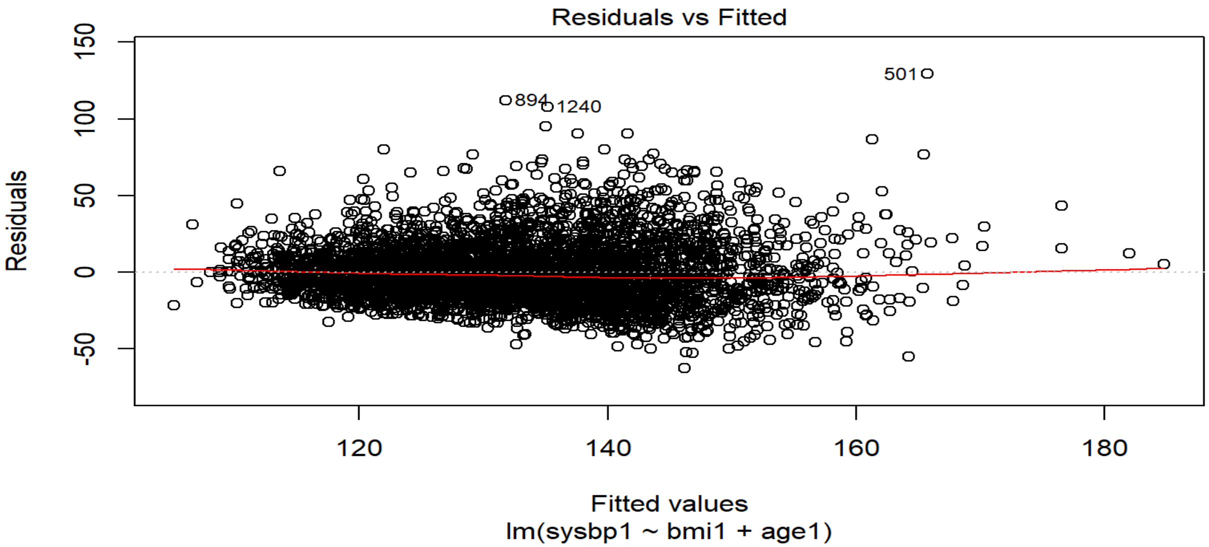
Homoscedasticity of residuals



Homoscedasticity of residuals

- We can see that there is no pattern to the residuals.
- They appear to be flat and not have a difference in width of the range of values.
- If we saw a pattern like a cone shape then we would not have homoscedasticity.

Untransformed Systolic Blood Pressure



Correlation of Residuals and Covariates:

```
cor.test(anscombe$x2, mod2$residuals)
cor.test(anscombe$x2sq, mod2a$residuals)
```

Correlation of Residuals and Covariates:

```
##
##  Pearson's product-moment correlation
##
## data:  anscombe$x2 and mod2$residuals
## t = -2.2251e-16, df = 9, p-value = 1
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.599875  0.599875
## sample estimates:
##           cor
## -7.416841e-17
##
##
##  Pearson's product-moment correlation
##
## data:  anscombe$x2sq and mod2a$residuals
## t = -1.2525e-16, df = 9, p-value = 1
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.599875  0.599875
## sample estimates:
##           cor
## -4.174919e-17
```