Introduction to Statistical Tests in R

BICF@UTSouthwestern.edu

Outline

- Normal Distribution & Normality Test
- Welch's Two-sample T-test
- Wilcoxon Rank Sum Test
- Pearson Correlation
- Multivariate Regression
- Negative Binomial & DESeq
- Principal Component Analysis

Normal Distribution & Normality Test

Normal distribution: a *continuous* distribution

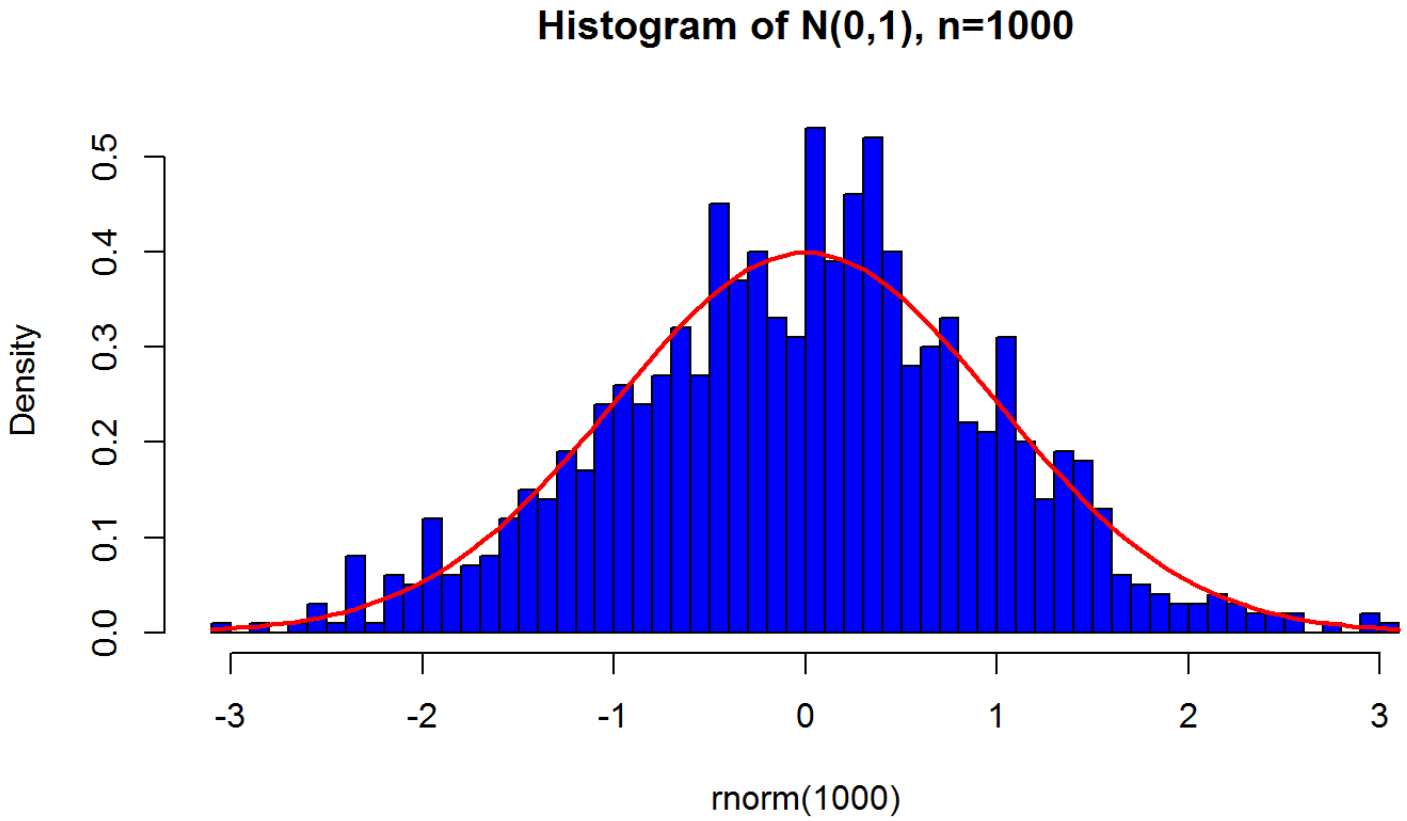
$$X \sim N(\mu, \sigma^2)$$

Density function:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

- Fully specified by mean μ and variance σ^2
- $e \approx 2.718$; $\pi \approx 3.142$.
- *Bell shaped*
- *Symmetric* around μ (mean/median/mode)
- *Histogram proportion limit, $n \to +\infty$*

Normal Distribution & Normality Test



Normal Distribution & Normality Test

Empirical Rule:

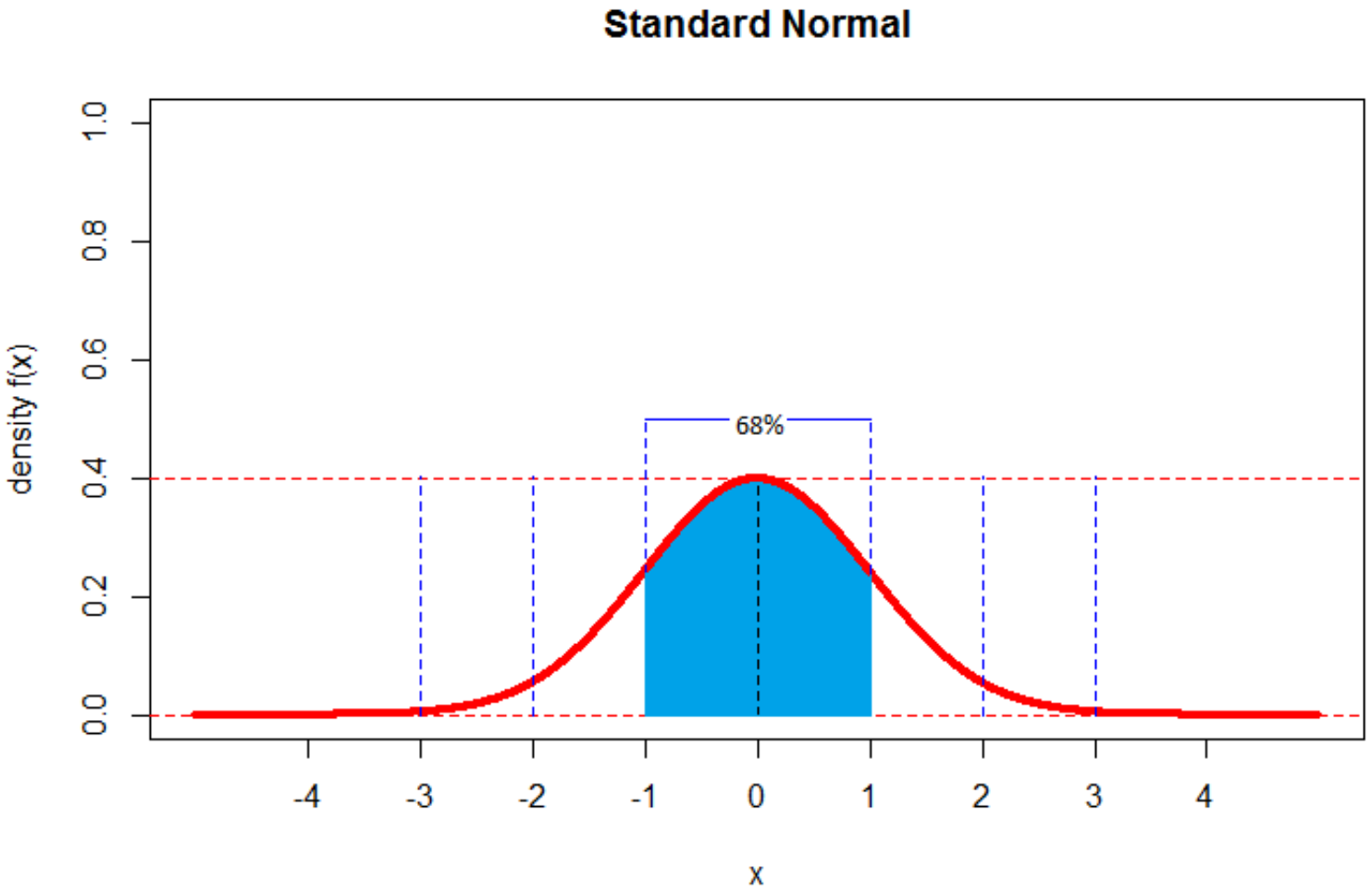
Given *any* $N(\mu,\ \sigma^2)$:

- 68% data $\in [\mu - \sigma,\ \mu + \sigma]$
- 95% data $\in [\mu - 2\sigma,\ \mu + 2\sigma]$
- 99.7% data $\in [\mu - 3\sigma,\ \mu + 3\sigma]$

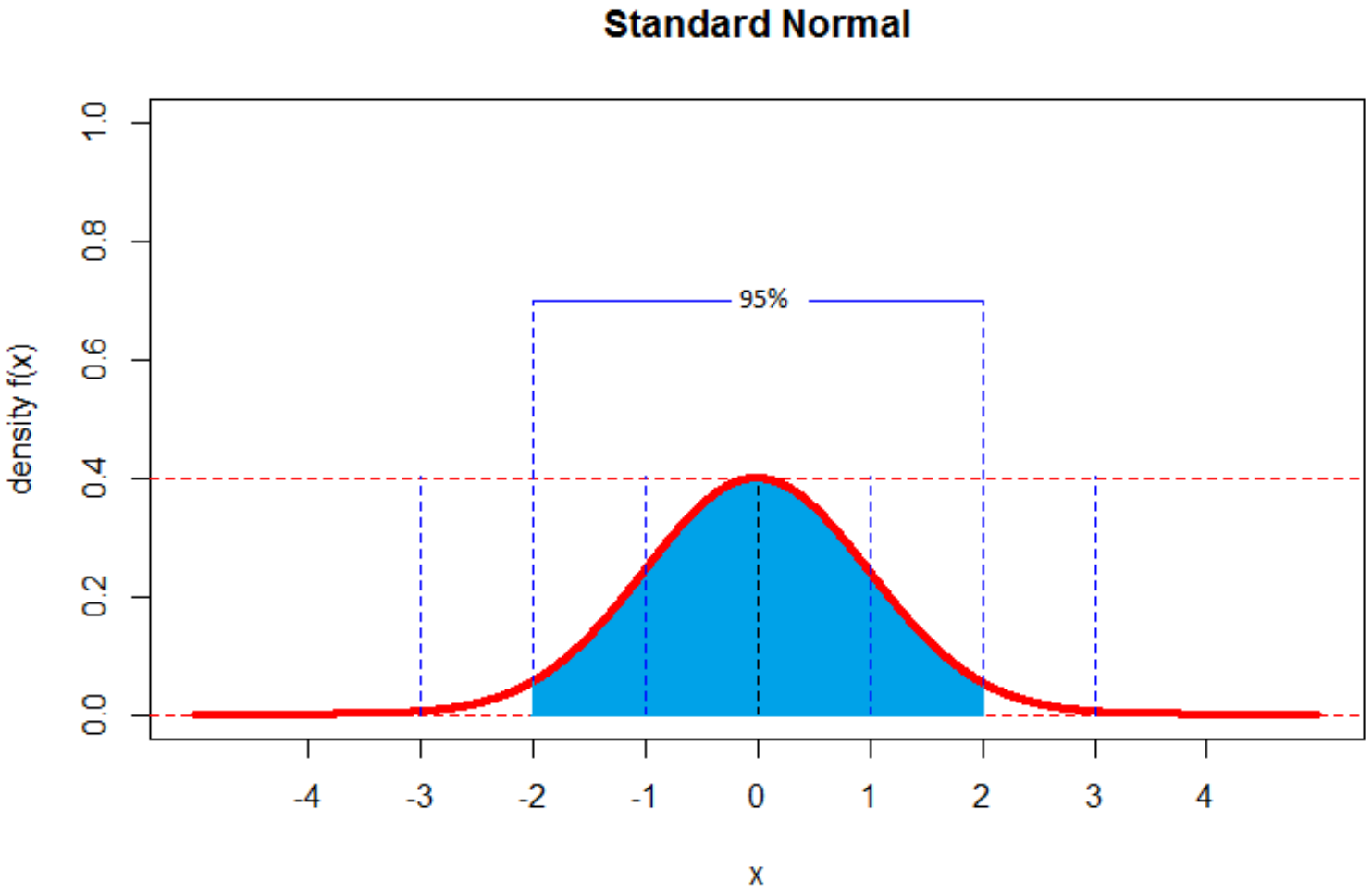
Standard Normal $N(0,1)$: $\mu = 0,\ \ \sigma^2 = 1$

- 68% data $\in [-1,\ 1]$
- 95% data $\in [-2,\ 2]$
- 99.7% data $\in [-3,\ 3]$

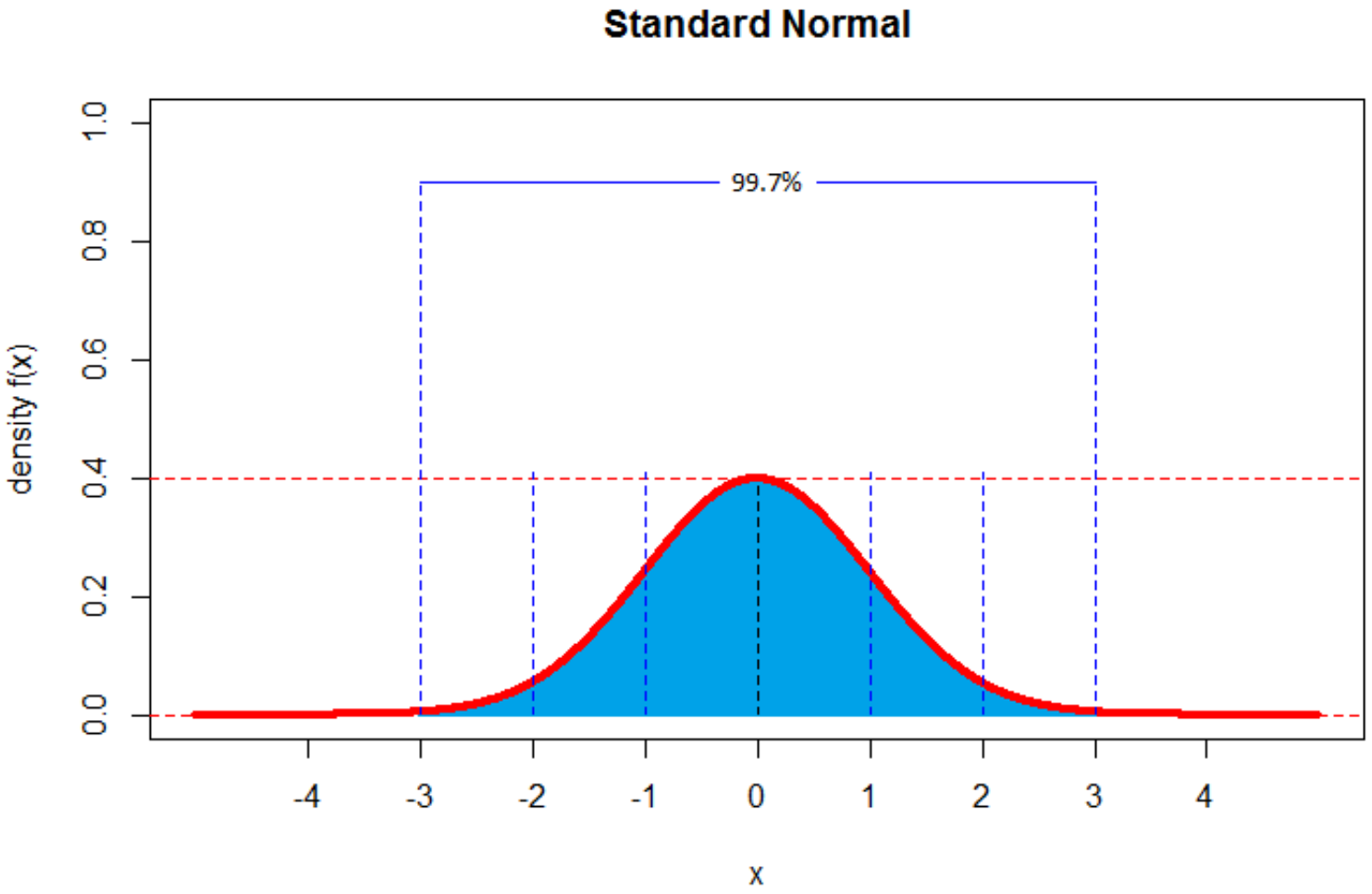
Normal Distribution & Normality Test



Normal Distribution & Normality Test

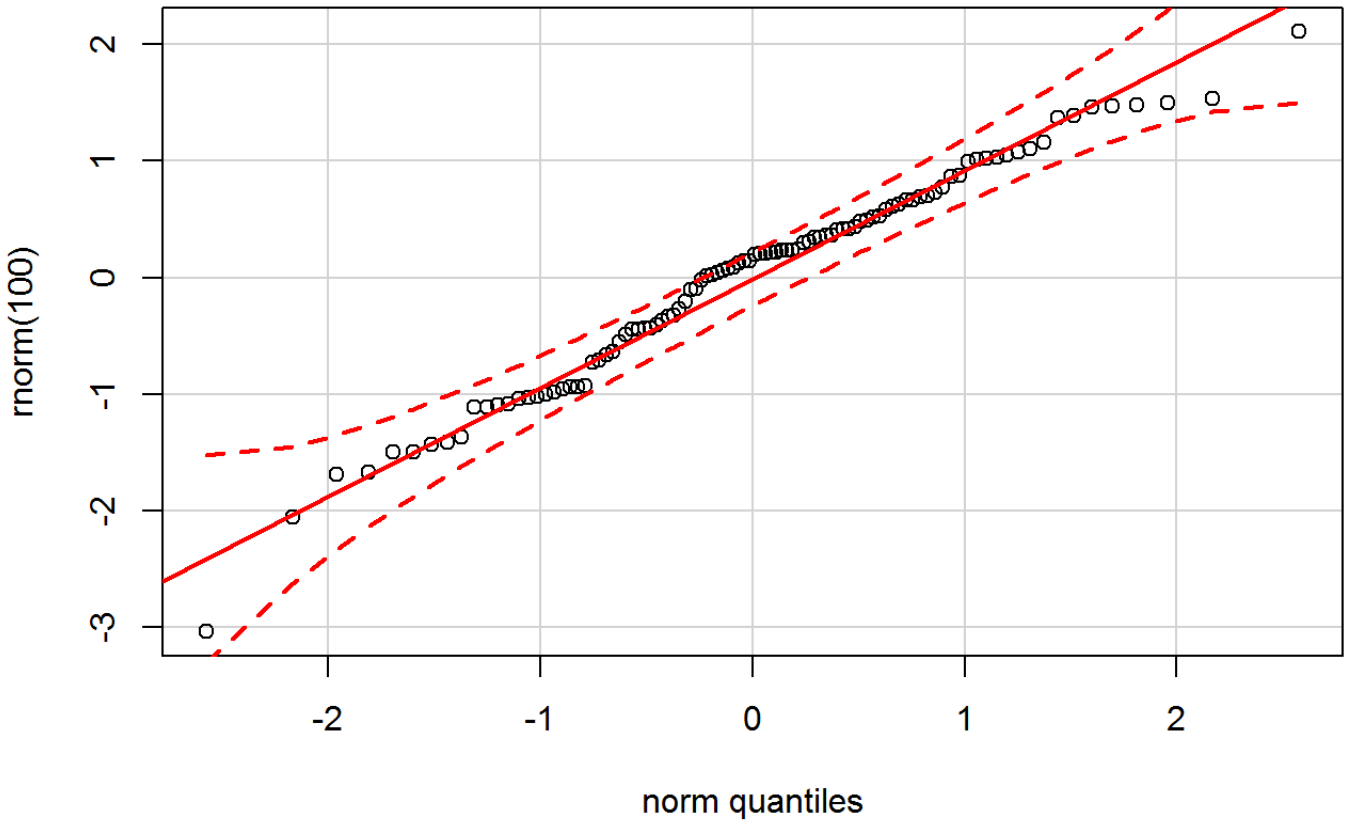


Normal Distribution & Normality Test



Normal Distribution & Normality Test

```
set.seed(234); car::qqPlot(rnorm(100)) # QQ-plot with 95% CI
```



Normal Distribution & Normality Test

```
shapiro.test(rnorm(100)) # Shapiro-Wilk test of normality

## The calculation of the p value is exact for n = 3, otherwise
## approximations are used, separately for 3 < n <12 and n >11

## Missing values are allowed, but the number of non-missing values
## must be between 3 and 5000.
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rnorm(100)
## W = 0.98084, p-value = 0.1544
```

Welch's Two-sample T-test

$$H_0: \mu_x = \mu_y \quad vs. \quad H_1: \mu_x \neq \mu_y$$

$$T = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}} \sim t_{d.f.}$$

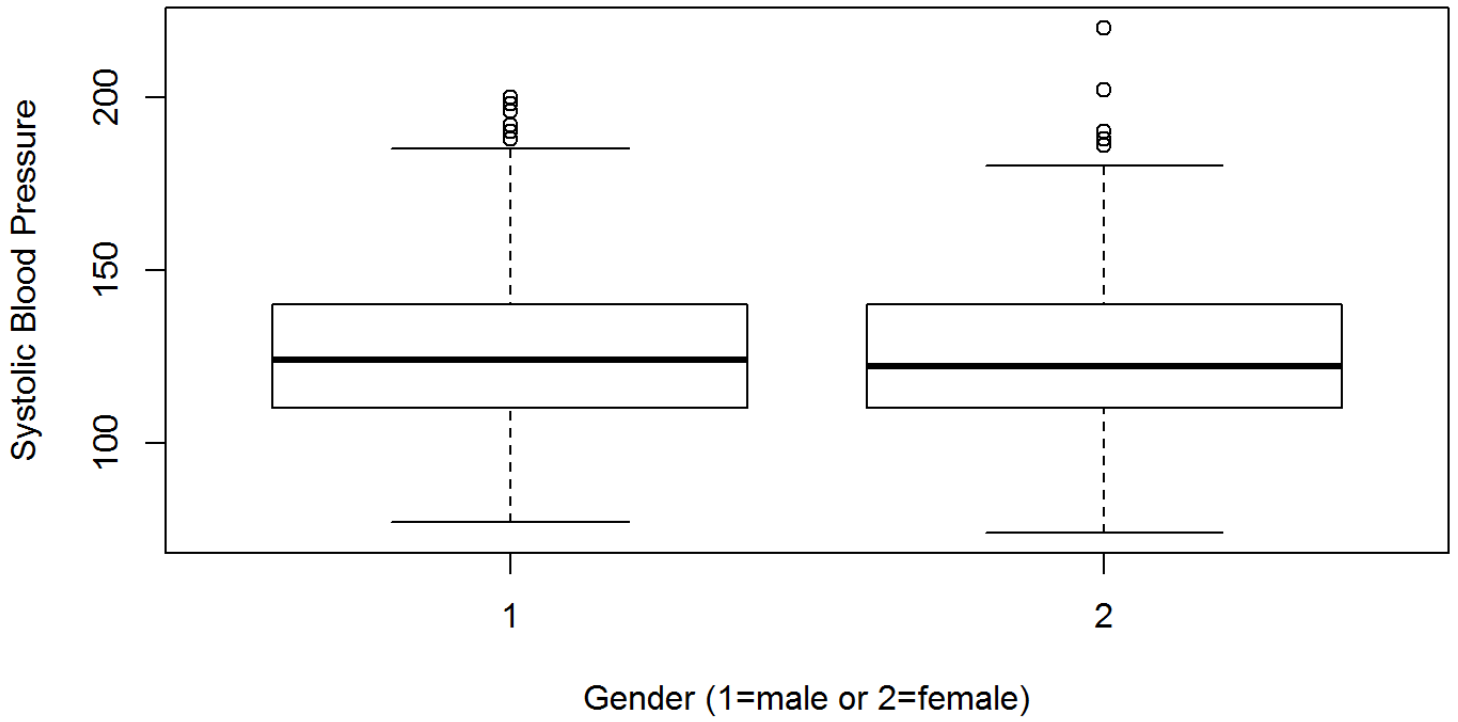
$$d.f. = \left(\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}\right)^2 \Big/ \left[\frac{\left(\frac{s_x^2}{n_x}\right)^2}{n_x - 1} + \frac{\left(\frac{s_y^2}{n_y}\right)^2}{n_y - 1}\right]$$

Assumption:

- Sample x and y are independent.
- For small samples, both x and y population need to be normal.

Welch's Two-sample T-test

- Example Data:



Welch's Two-sample T-test

- Two-sided test in R:

```
t.test(SYSBP~SEX)
```

```
##
##  Welch Two Sample t-test
##
## data:  SYSBP by SEX
## t = 1.6755, df = 2506.5, p-value = 0.09395
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1634916  2.0833293
## sample estimates:
## mean in group 1 mean in group 2
##        126.0106        125.0507
```

One-sided Two-sample T-test

$$H_0: \mu_M = \mu_F \quad vs. \quad H_1: \mu_M > \mu_F$$

```
t.test(SYSBP[SEX==1], SYSBP[SEX==2], "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  SYSBP[SEX == 1] and SYSBP[SEX == 2]
## t = 1.6755, df = 2506.5, p-value = 0.04698
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.01722994        Inf
## sample estimates:
## mean of x mean of y
##  126.0106  125.0507
```

Wilcoxon Rank Sum Test

H_0: two populations follow the same distribution

H_1: two populations follow different distributions

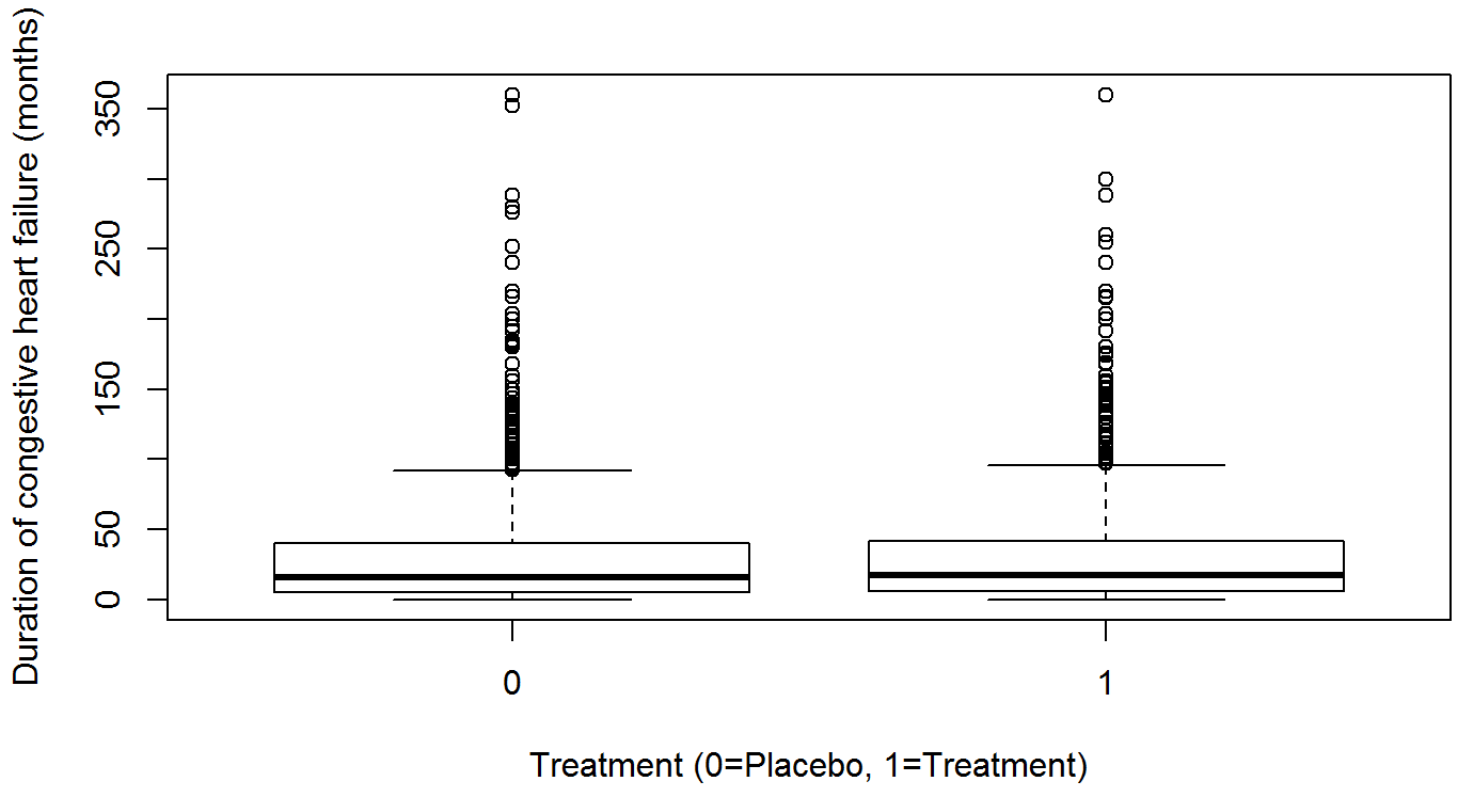
T = Sum of ranks of sample x in the combined data

Assumption:

- Sample x and y are independent.
- *The population follows a continuous distribution.*

Wilcoxon Rank Sum Test

- Example Data:



Wilcoxon Rank Sum Test

- Two-sided test in R:

```
wilcox.test(CHFDUR~TRTMT)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  CHFDUR by TRTMT
## W = 5665900, p-value = 0.2627
## alternative hypothesis: true location shift is not equal to 0
```

Wilcoxon Rank Sum Test

- One-sided test in R:

```
wilcox.test(CHFDUR[TRTMT==0], CHFDUR[TRTMT==1], "less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  CHFDUR[TRTMT == 0] and CHFDUR[TRTMT == 1]
## W = 5665900, p-value = 0.1314
## alternative hypothesis: true location shift is less than 0
```

Pearson Correlation

- Pearson Correlation Coefficient:

$$r = \hat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- $r \in [-1,\ \ 1]$
- $r = 0$: no linear association; does not imply independence
- $r = 1$: fit perfectly by an increasing line
- $r = -1$: fit perfectly by a decreasing line

Pearson Correlation

- Hypothesis Test of $\rho = 0$:

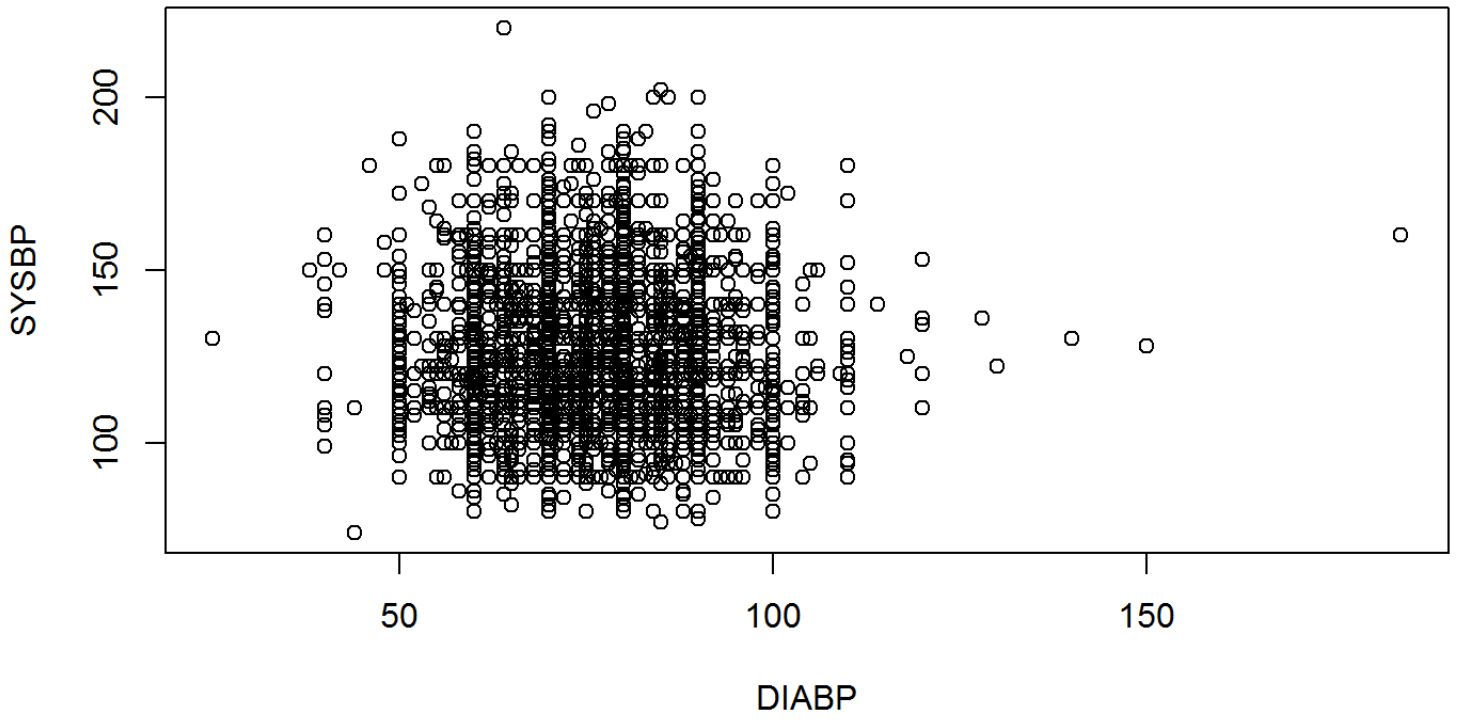
Assumption: both sample drawn from normal distributions

$$T = r\sqrt{\frac{n-2}{1-r^2}} \sim t_{n-2}$$

- Holds approximately for non-normal distributions if sample sizes are large.
- Asymptotic confidence interval by Fisher's Z-transformation.

Pearson Correlation

- Example Data:



Pearson Correlation

- $H_0: \rho = 0 \quad vs. \quad H_1: \rho \neq 0$

```
cor.test(DIABP, SYSBP)
```

```
##
##  Pearson's product-moment correlation
##
## data:  DIABP and SYSBP
## t = 0.83224, df = 6790, p-value = 0.4053
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.01368677  0.03387401
## sample estimates:
##       cor
## 0.01009933
```

Pearson Correlation

- $H_0: \rho = 0 \quad vs. \quad H_1: \rho > 0$

```
cor.test(DIABP, SYSBP, "greater")
```

```
##
##  Pearson's product-moment correlation
##
## data:  DIABP and SYSBP
## t = 0.83224, df = 6790, p-value = 0.2027
## alternative hypothesis: true correlation is greater than 0
## 95 percent confidence interval:
##  -0.00986294  1.00000000
## sample estimates:
##       cor
## 0.01009933
```

Multivariate Regression

Model Setup

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

$$\epsilon \sim N(0, \sigma^2), \quad i.i.d.$$

Equivalently

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

$$\epsilon \sim MVN(0, \sigma^2 I)$$

or

$$\boldsymbol{y|X} \sim MVN(\boldsymbol{X\beta}, \sigma^2 \boldsymbol{I})$$

Multivariate Regression

Coefficient Test:

$$T = \frac{\hat{\beta}}{\sqrt{Var(\hat{\beta})}} \sim t_{n-p} \quad or \quad N(0,1)$$

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

$$\begin{aligned} Var(\widehat{\boldsymbol{\beta}}) &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' \; Var(\boldsymbol{y}|\boldsymbol{X}) \; \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\ &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' \; (\sigma^2\boldsymbol{I}) \; \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\ &= \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1} \end{aligned}$$

Multivariate Regression

Overall Significance Test

H_0: The fit of the full model is the same as the intercept-only model

H_1: The full model fits the data better than the intercept-only model

Equivalently,

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_p = 0 \quad vs. \quad H_1: \beta_j \neq 0 \text{ for some } j$$

$$F = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 /(p)}{\sum_{i}^{n}(y_i - \hat{y}_i)^2 /(n-p-1)} \sim F_{(p,n-p-1)}$$

Multivariate Regression

Goodness of fit:

- *Coefficient of determination*

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i}^{n}(y_i - \bar{y})^2}$$

Proportion of the response variance that is explained by the regression, although not bounded between 0 and 1.

- *Adjusted R squared*

$$R^2_{adj} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 /(n-p-1)}{\sum_{i}^{n}(y_i - \bar{y})^2 /(n-1)} < R^2$$

R^2_{adj} adjusts for number of $X's$ relative to sample size.

Multivariate Regression

- Example Data: Response

```
summary(KLEVEL) # Baseline Serum Potassium level (mEq/l)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   4.100   4.300   4.397   4.600 434.000     801
```

```
summary(KLEVEL[- which(KLEVEL==434)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   4.100   4.300   4.326   4.600   6.300     801
```

```
length(KLEVEL)
```

```
## [1] 6800
```

Multivariate Regression

- Example Data: Response

```
DIGdata <- DIGdata[-which(KLEVEL==434),]; attach(DIGdata);
```

```
## The following objects are masked from DIGdata (pos = 6):
##
##     ACEINHIB, ACTLIMIT, AGE, ANGINA, BMI, CHESTX, CHFDUR,
##     CHFETIOL, CREAT, CREV, CREVDAYS, CVD, CVDDAYS, DEATH,
##     DEATHDAY, DIABETES, DIABP, DIG, DIGDAYS, DIGDOSE, DIGDOSER,
##     DIGUSE, DIURET, DIURETK, DWHF, DWHFDAYS, EJFMETH, EJFPER,
##     ELEVJVP, EXERTDYS, FUNCTCLS, HEARTRTE, HOSP, HOSPDAYS, HYDRAL,
##     HYPERTEN, ID, KLEVEL, KSUPP, MI, MIDAYS, NHOSP, NITRATES,
##     NSYM, OCVD, OCVDDAYS, OTH, OTHDAYS, PEDEMA, PREVMI, PULCONG,
##     RACE, RALES, REASON, RESTDYS, RINF, RINFDAYS, S3, SEX, STRK,
##     STRKDAYS, SVA, SVADAYS, SYSBP, TRTMT, UANG, UANGDAYS, VASOD,
##     VENA, VENADAYS, WHF, WHFDAYS
```

```
length(KLEVEL)
```

```
## [1] 6799
```

Multivariate Regression

- Example Data: Regressors

```
table(DIG) # Digoxin Toxicity (0=No, 1=Yes)
```

```
## DIG
##    0    1
## 6701   98
```

```
table(DIABETES) # History of Diabetes (0=No or Unkonw, 1=Yes)
```

```
## DIABETES
##    0    1
## 4866 1933
```

```
table(SEX) # 1=Male, 2=Female
```

```
## SEX
##    1    2
## 5280 1519
```

Multivariate Regression

- Example Data: fit regression with only main effects

```
summary(lm(KLEVEL ~ DIG + DIABETES + SEX))
```

```
##
## Call:
## lm(formula = KLEVEL ~ DIG + DIABETES + SEX)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3445 -0.2700 -0.0113  0.2887  1.9887
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.27755    0.02066 207.029  < 2e-16 ***
## DIG         -0.18386    0.05463  -3.365 0.000769 ***
## DIABETES     0.03315    0.01446   2.292 0.021948 *
## SEX          0.03380    0.01566   2.158 0.030978 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5057 on 5994 degrees of freedom
##   (801 observations deleted due to missingness)
## Multiple R-squared:  0.003505,   Adjusted R-squared:  0.003007
## F-statistic: 7.029 on 3 and 5994 DF,  p-value: 0.0001028
```

Multivariate Regression

- Example Data: fit regression with interaction effects

```
summary(lm(KLEVEL ~.^3, data=data.frame(DIG, DIABETES, SEX)))
```

```
##
## Call:
## lm(formula = KLEVEL ~ .^3, data = data.frame(DIG, DIABETES, SEX))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3502 -0.2595 -0.0100  0.2900  1.9900
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.27056    0.02417 176.685   <2e-16 ***
## DIG             -0.39556    0.19004  -2.082   0.0374 *
## DIABETES         0.07042    0.04516   1.559   0.1190
## SEX              0.03945    0.01872   2.107   0.0352 *
## DIG:DIABETES    -0.04542    0.37968  -0.120   0.9048
## DIG:SEX          0.16555    0.13721   1.207   0.2277
## DIABETES:SEX    -0.03021    0.03492  -0.865   0.3870
## DIG:DIABETES:SEX  0.02521    0.27985   0.090   0.9282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5058 on 5990 degrees of freedom
##   (801 observations deleted due to missingness)
## Multiple R-squared:  0.003983,   Adjusted R-squared:  0.002819
## F-statistic: 3.422 on 7 and 5990 DF,  p-value: 0.00118
```

Multivariate Regression

Diagnostic Plots:

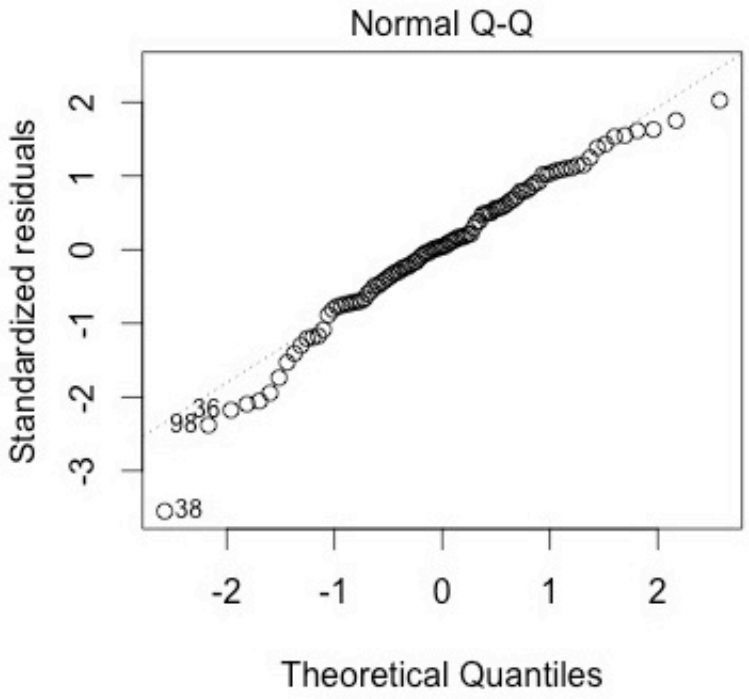
1. Check linearity and homoscedasticity:
   * *Residuals vs. Fitted Plot*
   * *Scale-Location Plot*

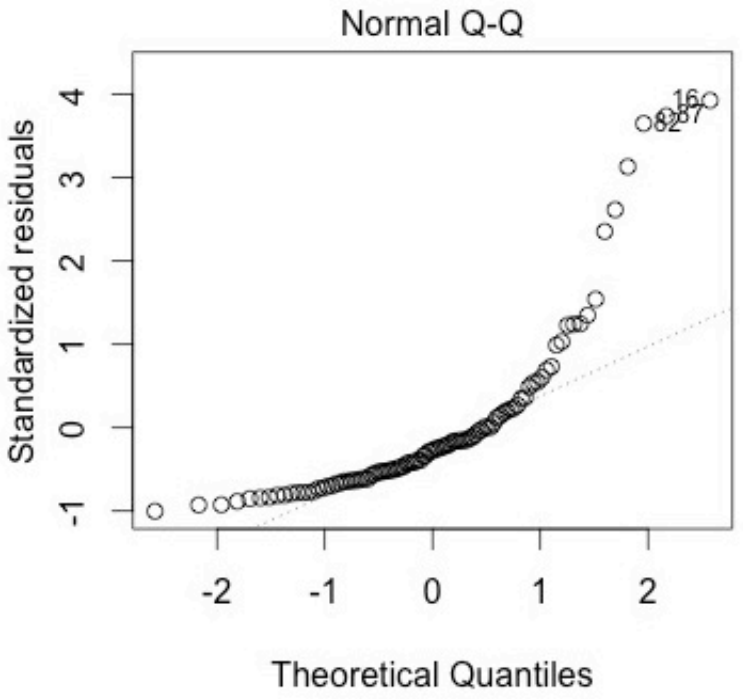
2. Check residual normality:
   * *Q-Q Plot*

3. Identify influential data points:
   * *Residuals vs. Leverage Plot*

Multivariate Regression

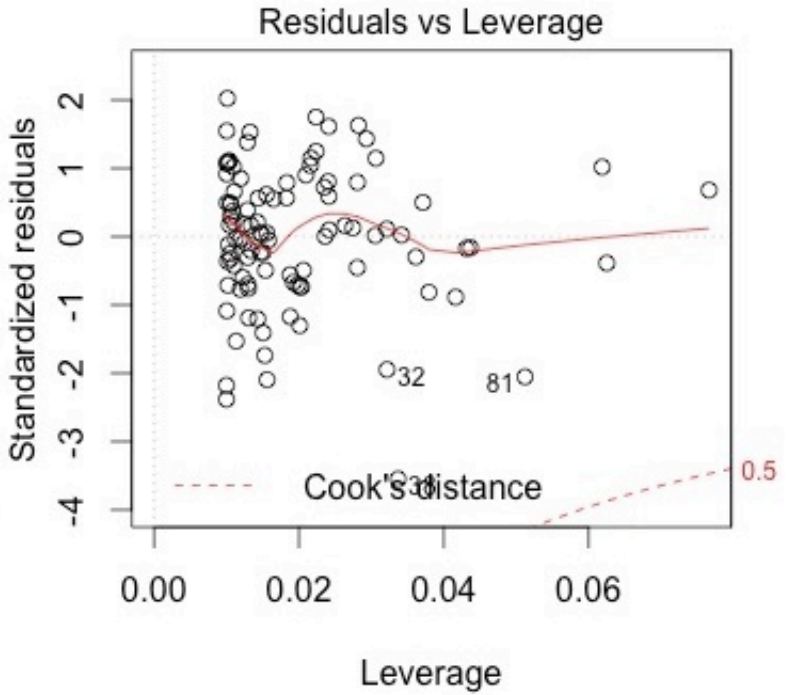
- Q-Q plot:

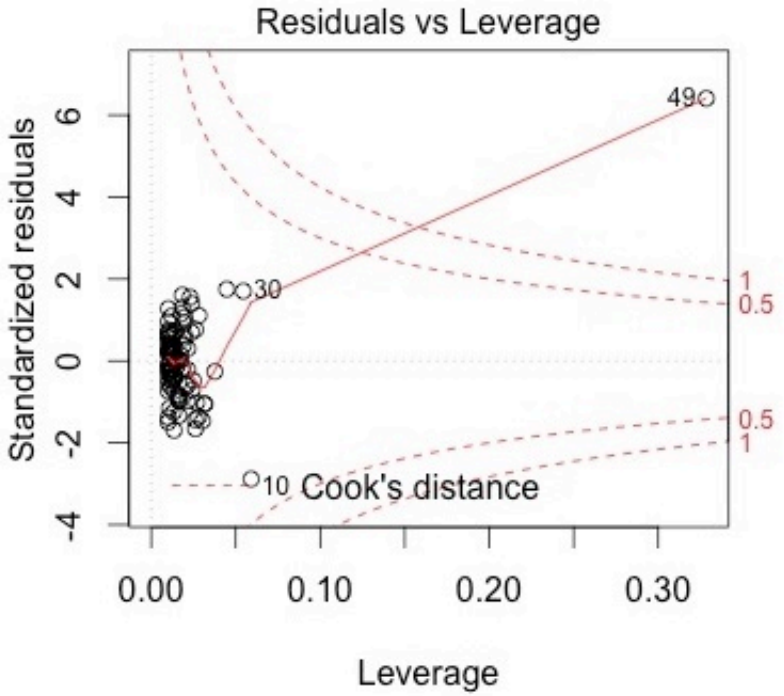




Multivariate Regression

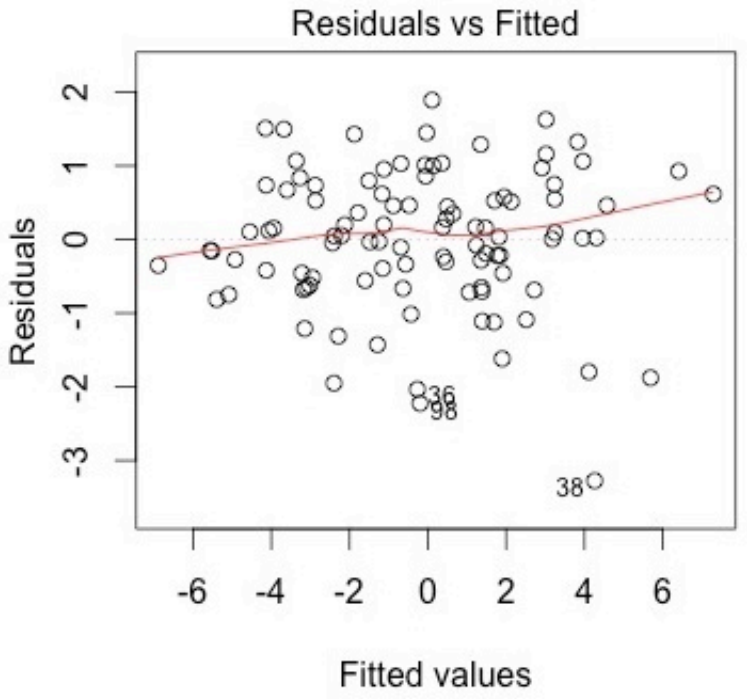
- Residuals vs. Leverage Plot:

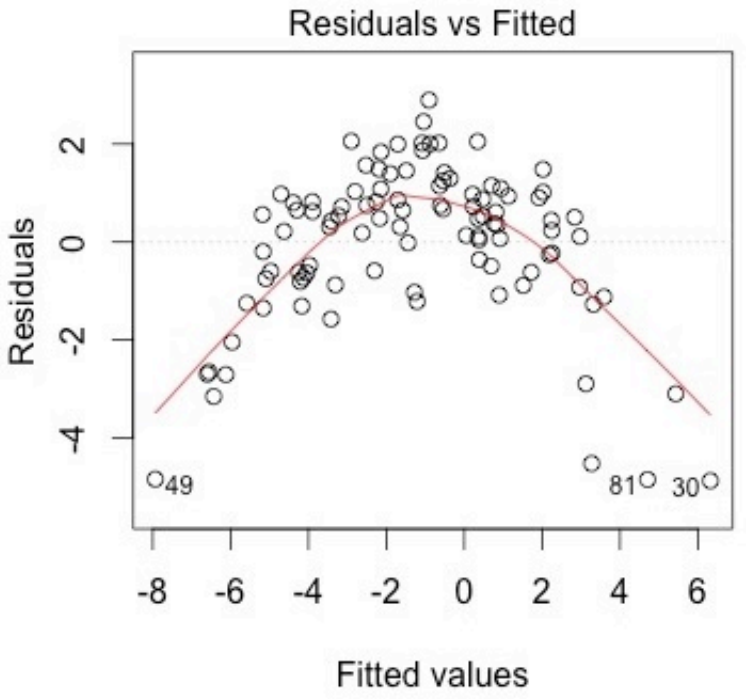




Multivariate Regression

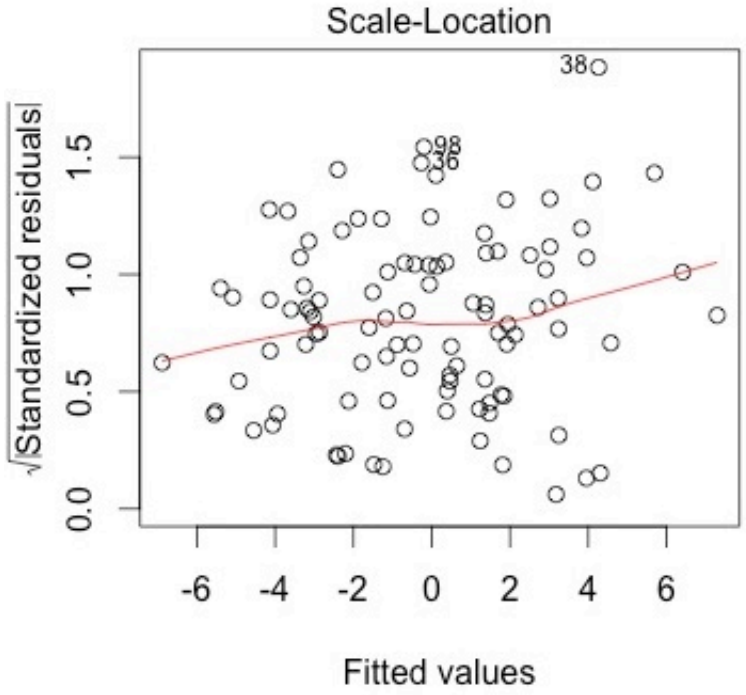
- Residuals vs. Fitted Plot:

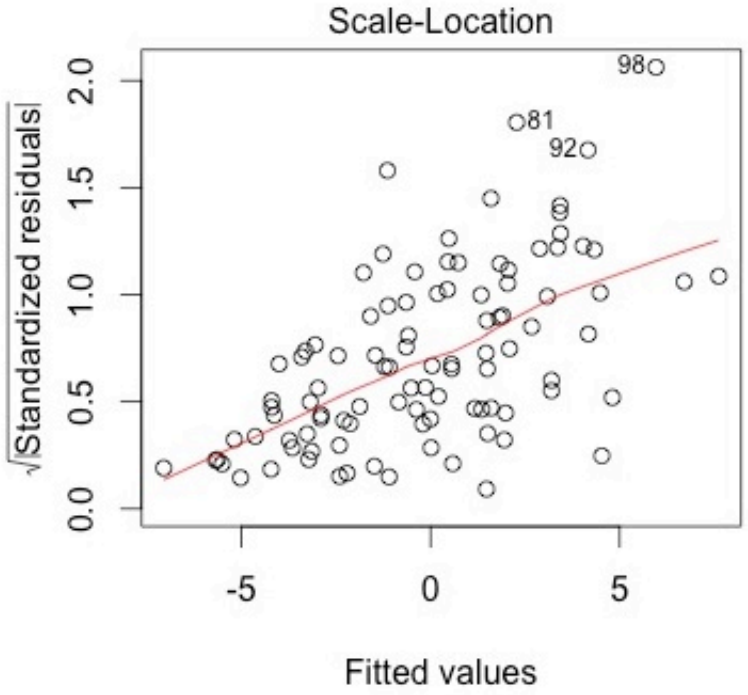




Multivariate Regression

- Scale-Location Plot:



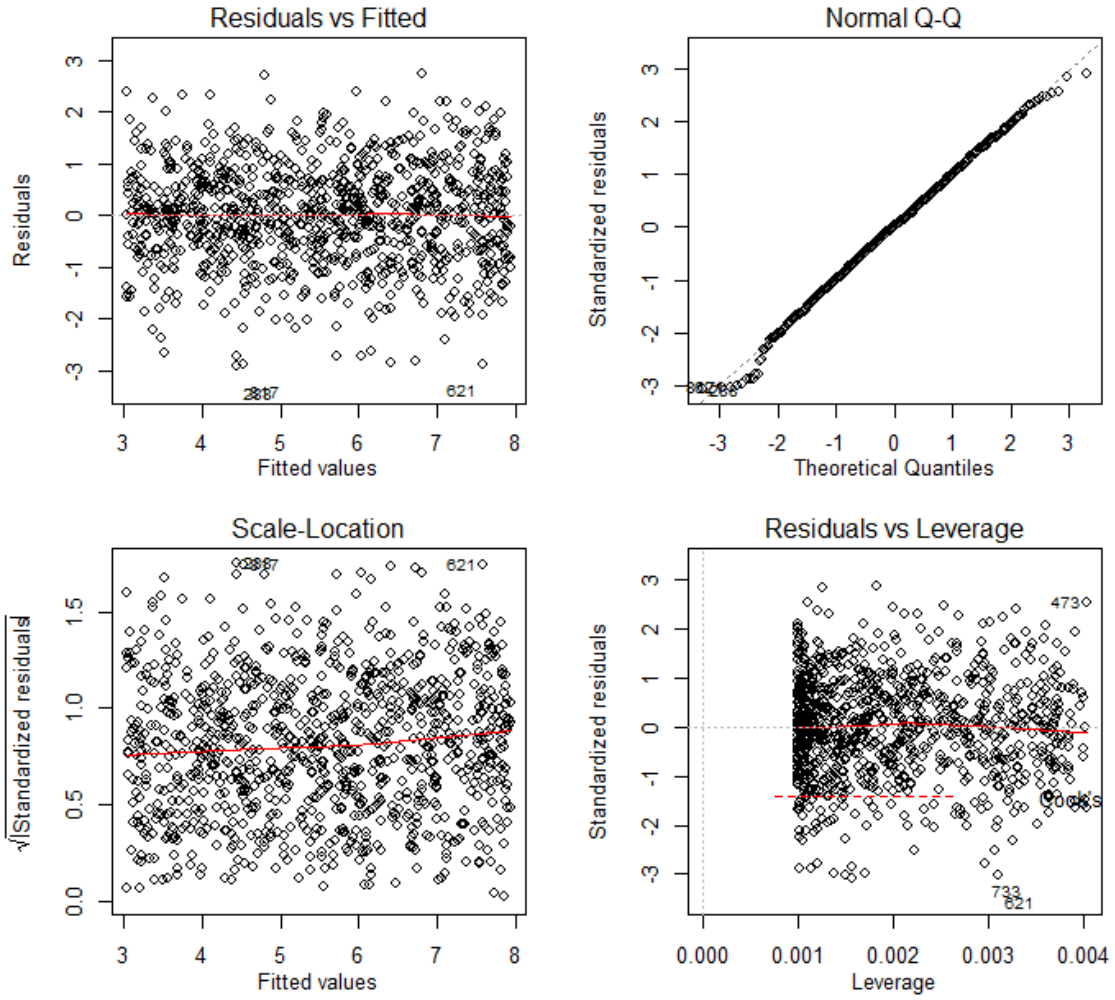


Multivariate Regression

- Simulation with uniform regressor in linear form:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x+rnorm(1000)
par(mfrow=c(2,2)); plot(lm(y~x));
```

Multivariate Regression



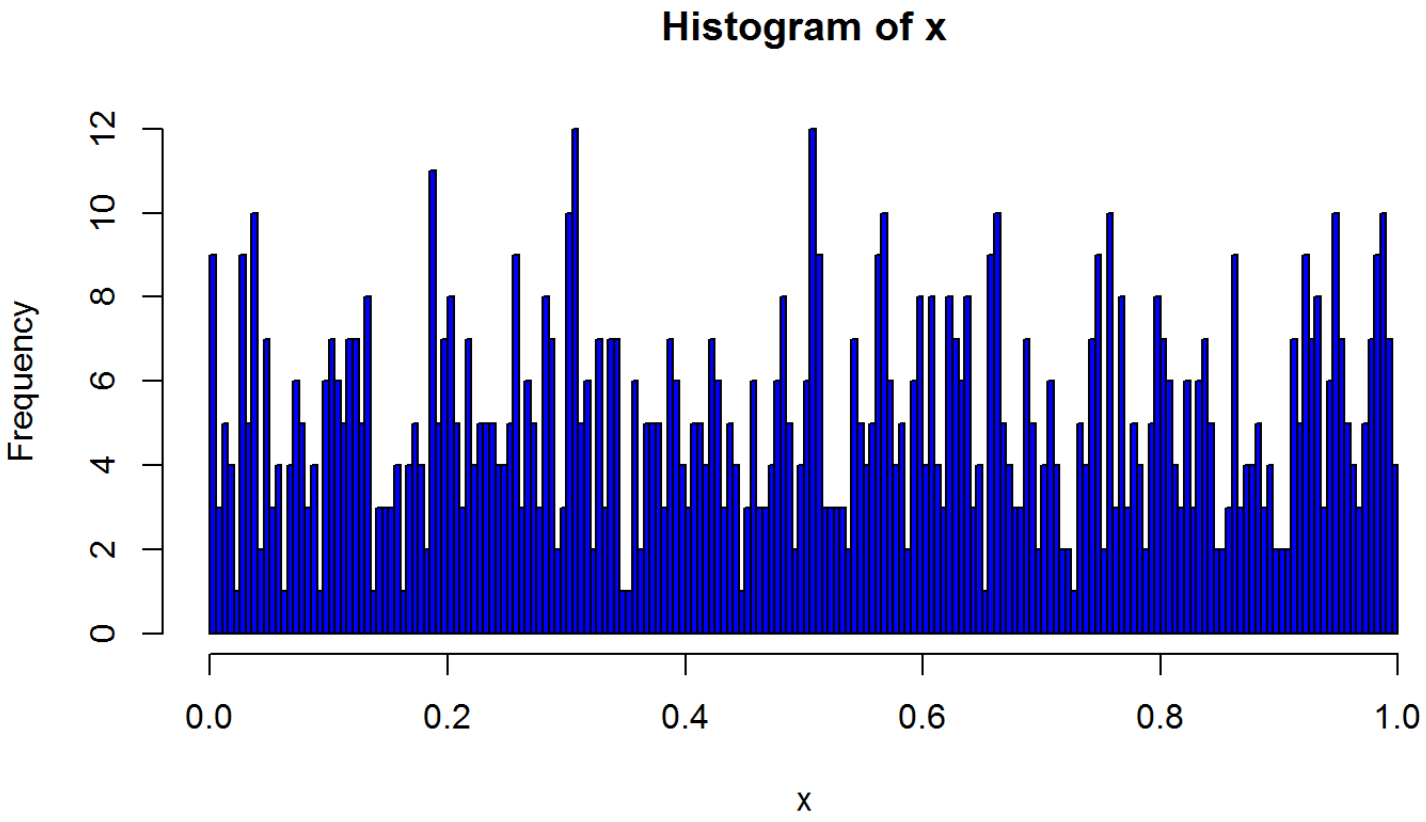
Multivariate Regression

- Fitting summary:

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.91933 -0.62956  0.01084  0.63819  2.73178
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.04449    0.06028   50.51   <2e-16 ***
## x            4.88928    0.10306   47.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9486 on 998 degrees of freedom
## Multiple R-squared:  0.6928, Adjusted R-squared:  0.6925
## F-statistic:  2251 on 1 and 998 DF,  p-value: < 2.2e-16
```

Multivariate Regression

- Uniform regressor:

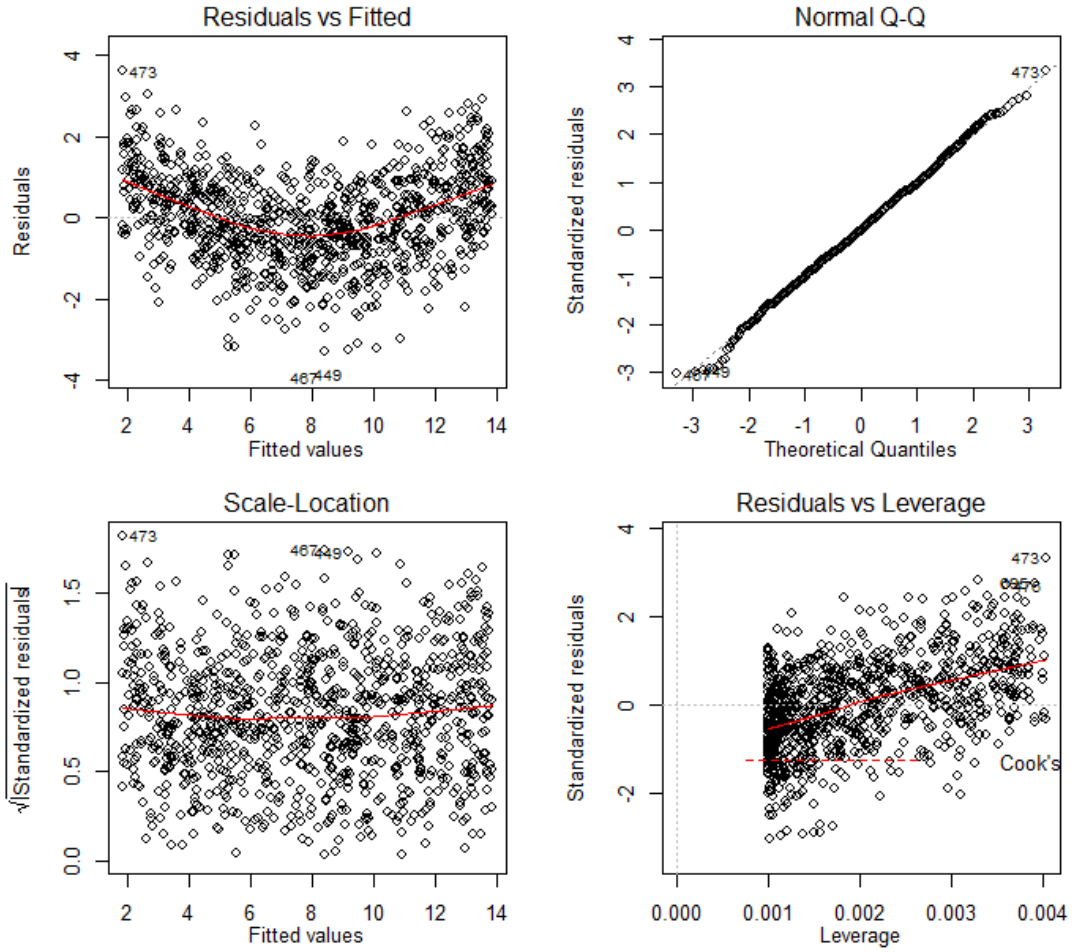


Multivariate Regression

- Simulation with uniform regressor in quadratic form: case 1

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x+7*x^2+rnorm(1000)
par(mfrow=c(2,2)); plot(lm(y~x));
```

Multivariate Regression



Multivariate Regression

- Fitting summary:

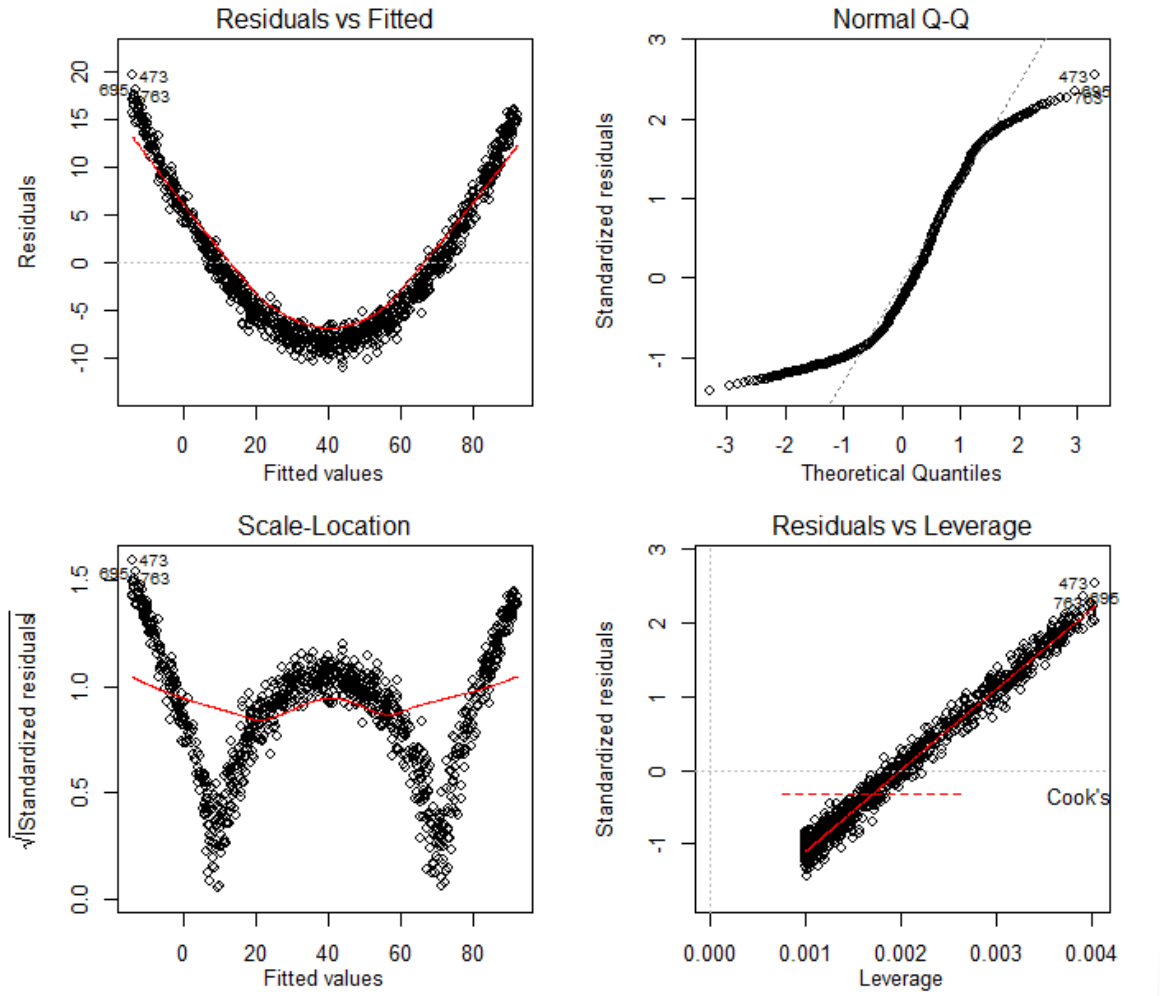
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3057 -0.6886 -0.0230  0.7566  3.6079
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.83989    0.06924   26.57   <2e-16 ***
## x           11.98389    0.11839  101.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.09 on 998 degrees of freedom
## Multiple R-squared:  0.9112, Adjusted R-squared:  0.9112
## F-statistic: 1.025e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Multivariate Regression

- Simulation with uniform regressor in quadratic form: case 2

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x+100*x^2+rnorm(1000)
par(mfrow=c(2,2)); plot(lm(y~x));
```

Multivariate Regression



Multivariate Regression

- Fitting summary:

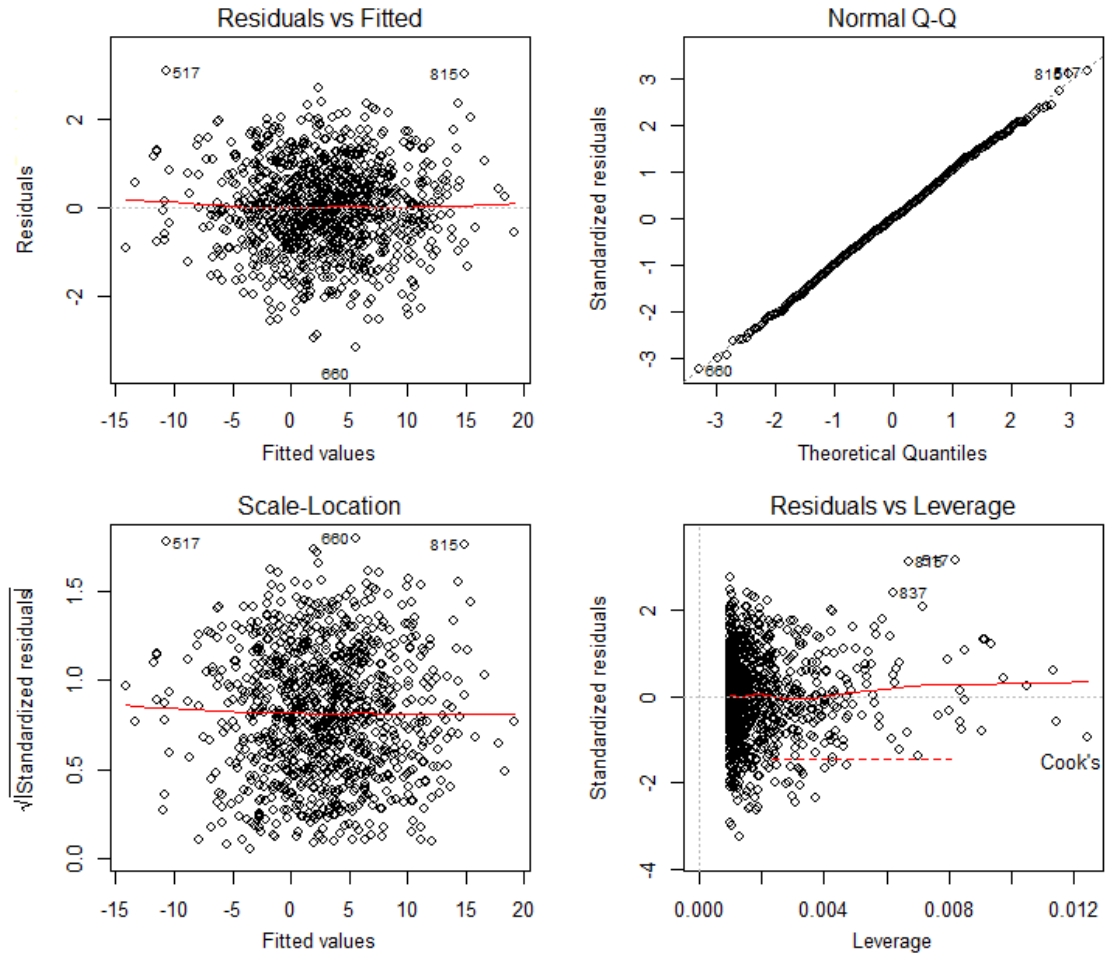
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.037  -6.767  -2.016   6.182  19.554
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -14.1642     0.4897  -28.92   <2e-16 ***
## x           106.2410     0.8374  126.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.707 on 998 degrees of freedom
## Multiple R-squared:  0.9416, Adjusted R-squared:  0.9416
## F-statistic: 1.61e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Multivariate Regression

- Simulation with normal regressor in linear form:

```
set.seed(1234)
x <- rnorm(1000)
y <- 3+5*x+rnorm(1000)
par(mfrow=c(2,2)); plot(lm(y~x));
```

Multivariate Regression



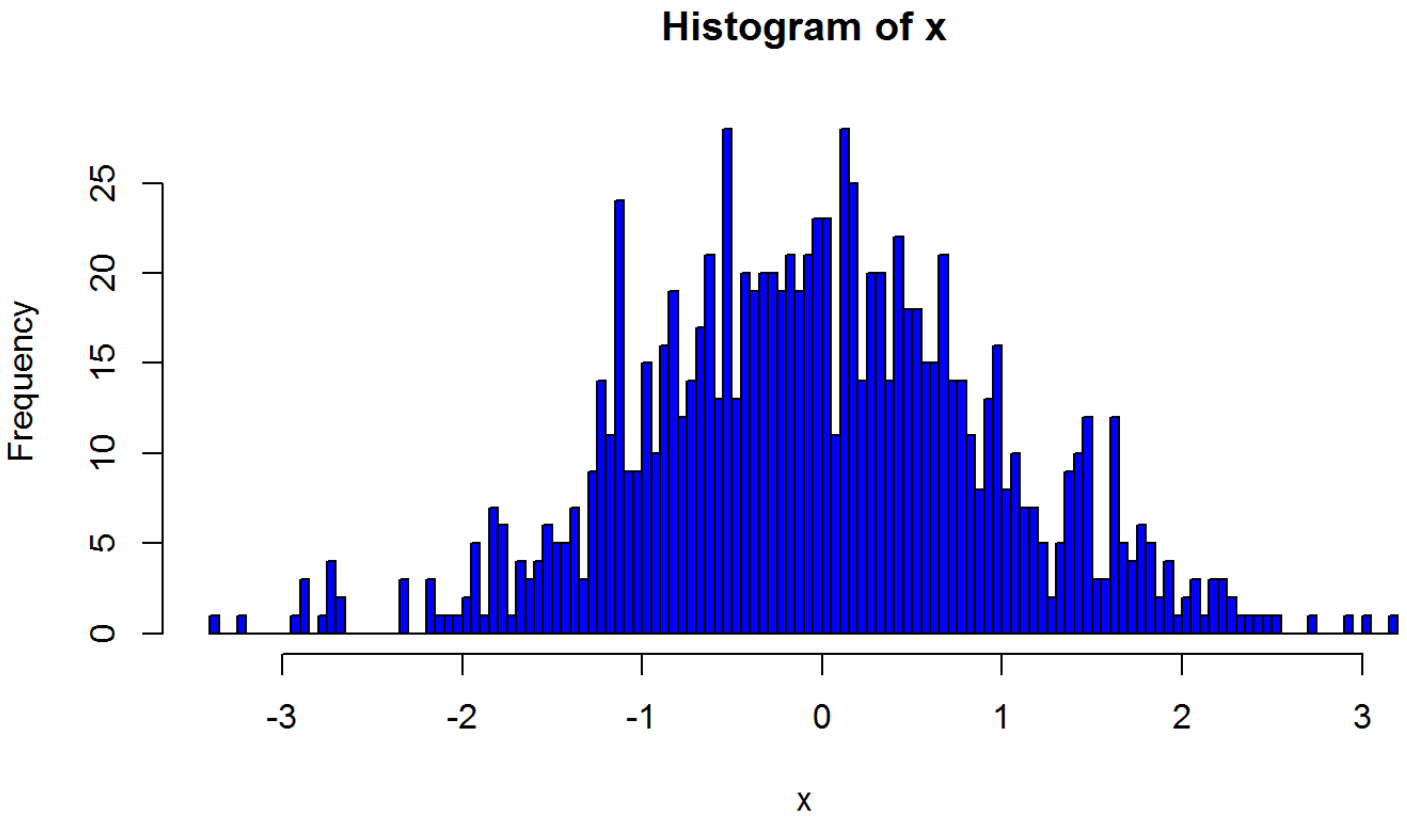
Multivariate Regression

- Fitting summary:

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1661 -0.6439  0.0145  0.6537  3.0684
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.01599    0.03100   97.28   <2e-16 ***
## x            5.05571    0.03109  162.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9801 on 998 degrees of freedom
## Multiple R-squared:  0.9636, Adjusted R-squared:  0.9636
## F-statistic: 2.644e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Multivariate Regression

- Normal regressor:

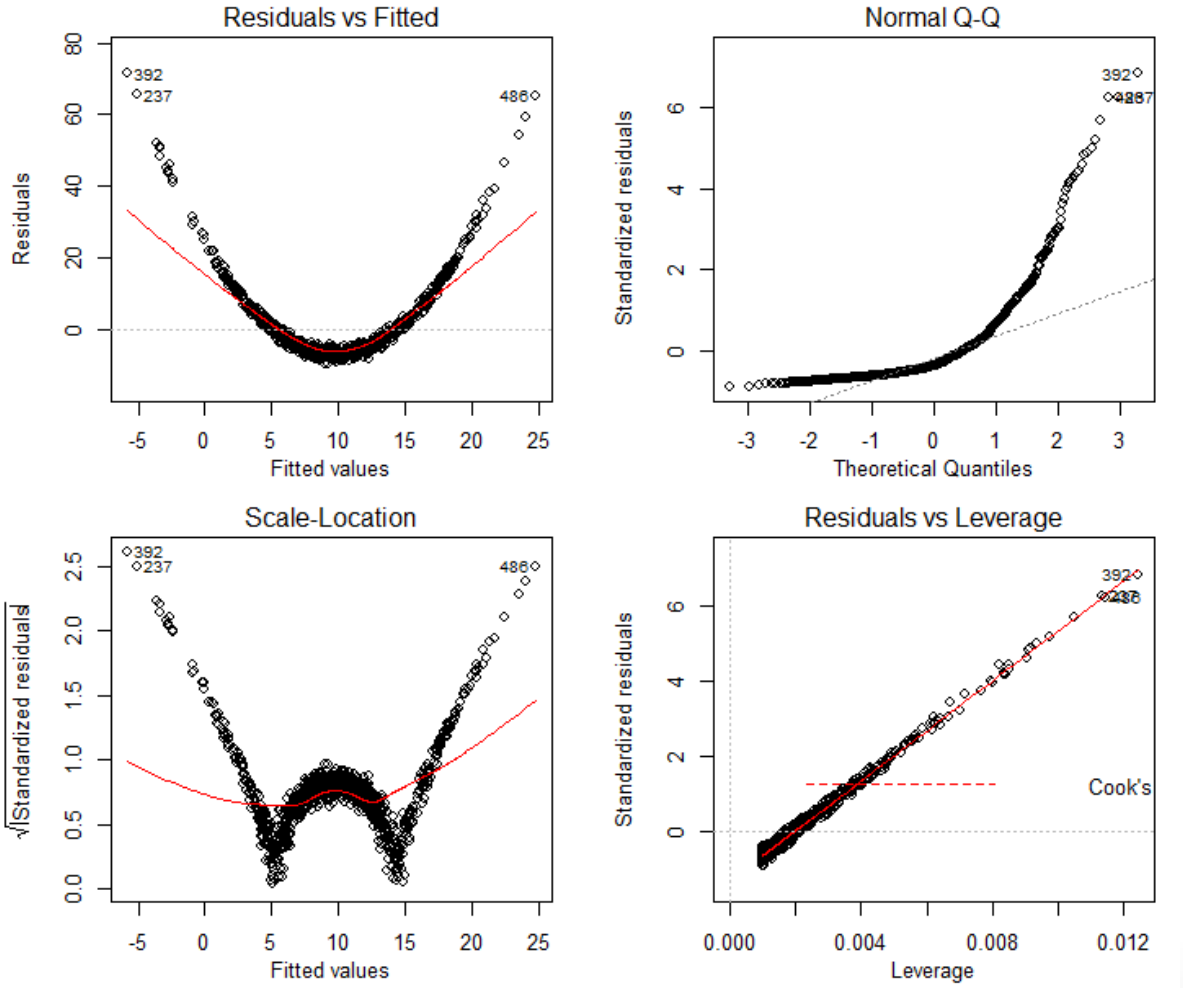


Multivariate Regression

- Simulation with normal regressor in quadratic form:

```
set.seed(1234)
x <- rnorm(1000)
y <- 3+5*x+7*x^2+rnorm(1000)
par(mfrow=c(2,2)); plot(lm(y~x));
```

Multivariate Regression



Multivariate Regression

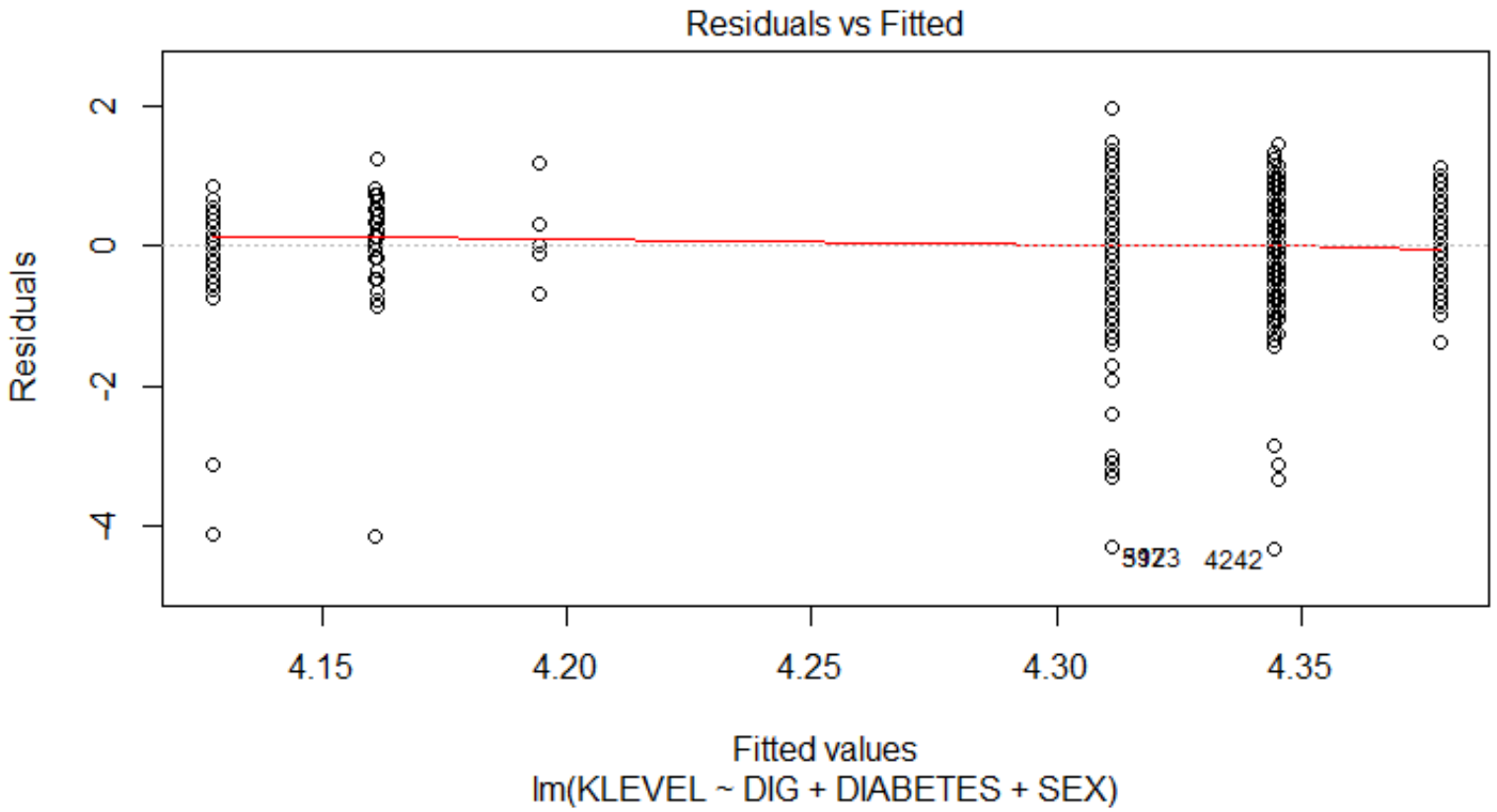
- Fitting summary:

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.727 -6.123 -3.983  1.715 71.486
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.9659     0.3328   29.94   <2e-16 ***
## x             4.6471     0.3338   13.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.52 on 998 degrees of freedom
## Multiple R-squared:  0.1627, Adjusted R-squared:  0.1618
## F-statistic: 193.9 on 1 and 998 DF,  p-value: < 2.2e-16
```

Multivariate Regression

- Example Data: diagnostic plots of regression with only main effects

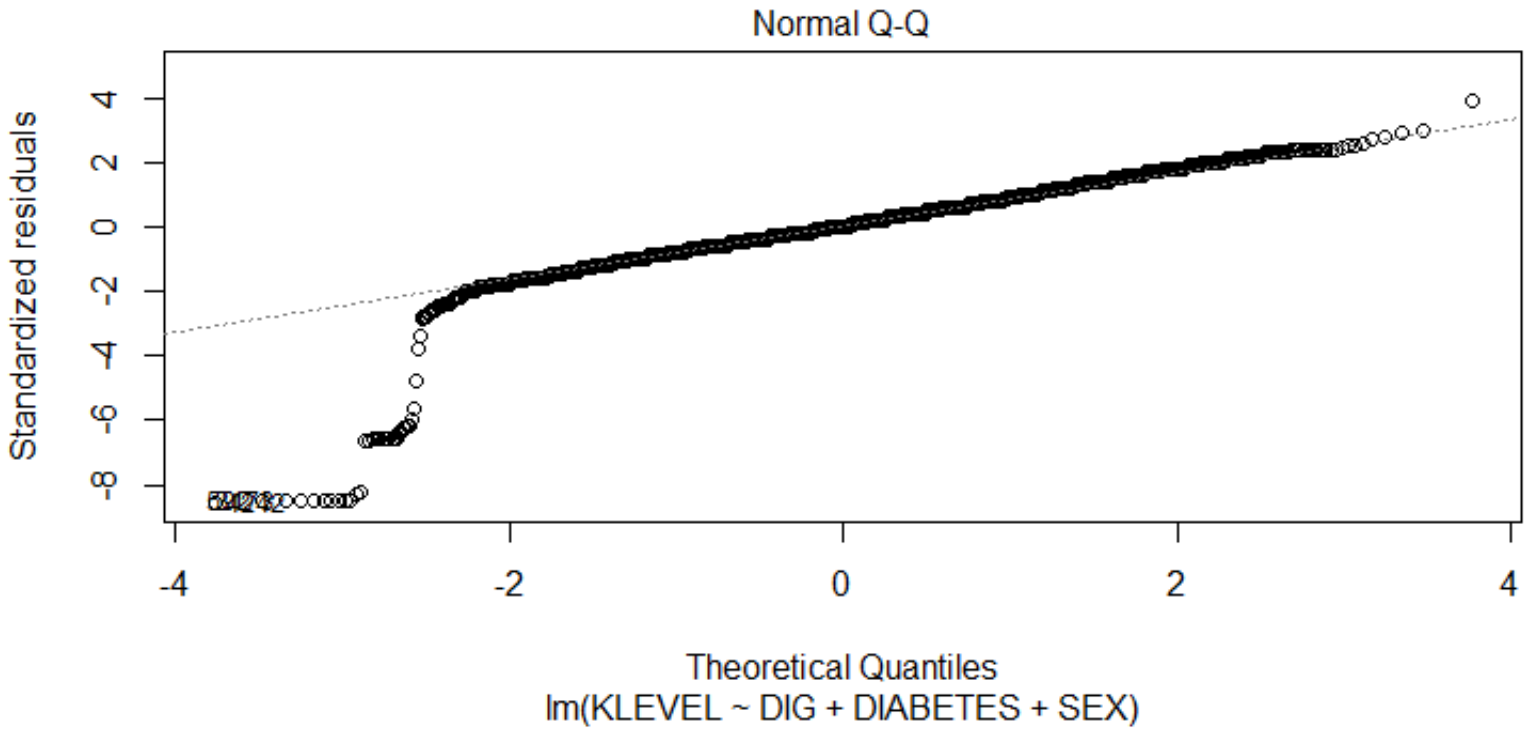
```
plot(lm(KLEVEL ~ DIG + DIABETES + SEX))
```



Multivariate Regression

- Example Data: diagnostic plots of regression with only main effects

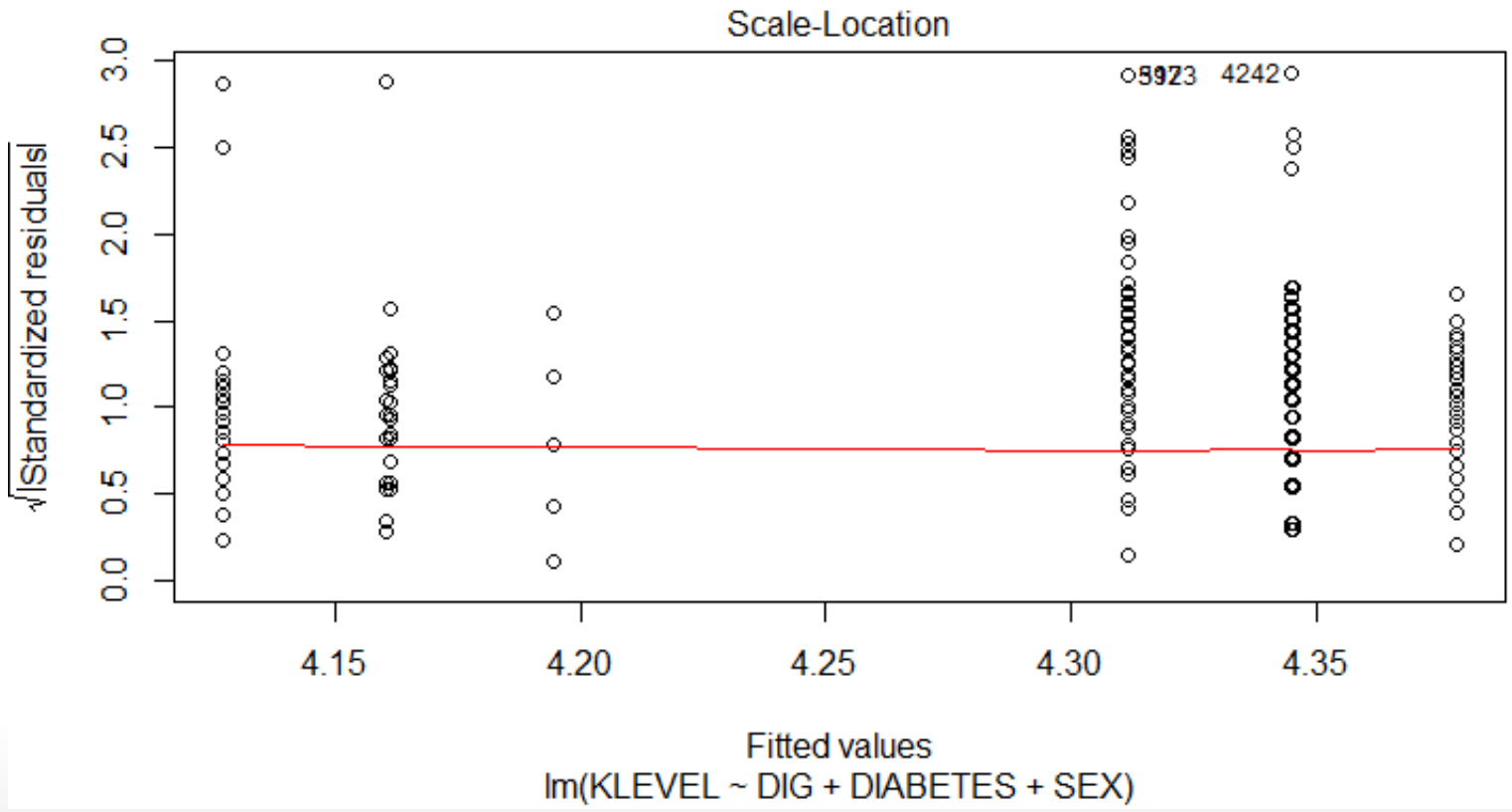
```
## Hit <Return> to see next plot:
```



Multivariate Regression

- Example Data: diagnostic plots of regression with only main effects

```
## Hit <Return> to see next plot:
```



Multivariate Regression

- Example Data: diagnostic plots of regression with only main effects

```
## Hit <Return> to see next plot:
```

Multivariate Regression

- Tests for checking constant variance assumption:

```
## Breush Pagan Test:
lmtest::bptest(lm(KLEVEL ~ DIG + DIABETES + SEX))
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm(KLEVEL ~ DIG + DIABETES + SEX)
## BP = 25.631, df = 3, p-value = 1.139e-05
```

```
## Non-constant Variance Score Test:
car::ncvTest(lm(KLEVEL ~ DIG + DIABETES + SEX))
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 204.2421    Df = 1     p = 2.478274e-46
```

Multivariate Regression

Multicollinearity

- Linear relationship exists among predictors, without the response variable.
- **Evaluation of $(X'X)^{-1}$ is unstable.**
- Variance inflation factor (VIF) > 10.

$$VIF(\hat{\beta}_i) = \frac{1}{1 - R_i^2}$$

- Condition number $\kappa > 15$.

$$k = \sqrt{\frac{|\lambda_{max}|}{|\lambda_{min}|}}, \quad \lambda_{min} \neq 0$$

Multivariate Regression

- Check multicollinearity:

```
car::vif(lm(KLEVEL ~., data=data.frame(DIG, DIABETES, SEX)))
```

```
##      DIG DIABETES      SEX
## 1.000550 1.000082 1.000481
```

```
car::vif(lm(KLEVEL ~.^3, data=data.frame(DIG, DIABETES, SEX)))
```

```
##              DIG         DIABETES              SEX     DIG:DIABETES
##        12.103871         9.747427         1.429230        12.352165
##           DIG:SEX     DIABETES:SEX DIG:DIABETES:SEX
##        11.986131        10.179364        12.206203
```

```
kappa(lm(KLEVEL ~., data=data.frame(DIG, DIABETES, SEX)))^0.5
```

```
## [1] 3.174667
```

```
kappa(lm(KLEVEL ~.^3, data=data.frame(DIG, DIABETES, SEX)))^0.5
```

```
## [1] 11.33542
```

Multivariate Regression

- Simulation with uniform regressor in quadratic form:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x + 100*x^2 +rnorm(1000)
car::vif(lm(y~x+I(x^2)))
```

```
##        x   I(x^2)
## 15.90843 15.90843
```

Multivariate Regression

- Fitting summary:

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.91890 -0.62897  0.01039  0.63878  2.73195
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.04651    0.09059   33.63   <2e-16 ***
## x             4.87741    0.41127   11.86   <2e-16 ***
## I(x^2)      100.01171    0.39283  254.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9491 on 997 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.9991
## F-statistic: 5.632e+05 on 2 and 997 DF,  p-value: < 2.2e-16
```

Negative Binomial & DESeq

Negative Binomial: a *discrete* distribution

$$X \sim NB(r, p)$$

$$\mu = E[X] = \frac{rp}{1-p}, \qquad \sigma^2 = Var[X] = \frac{rp}{(1-p)^2}$$

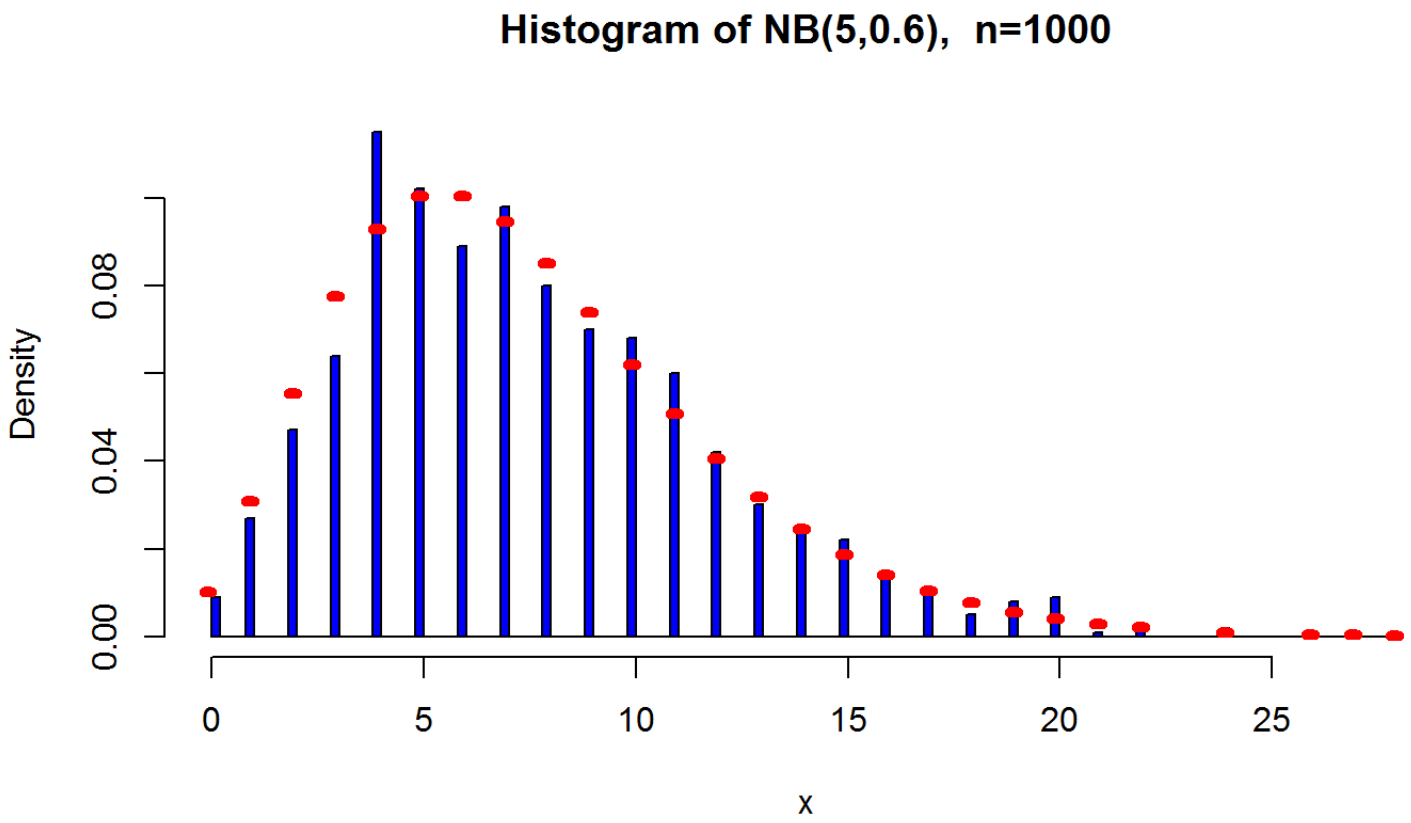
- X: number of successes until r of failures have occurred
- p: probability of success in each trial

Density function:

$$f(x) = \Pr(X = x | r, p) = \binom{x + r - 1}{x} p^x (1-p)^r$$

- Fully specified by r, p
- *Asymmetric*, mean≠median≠mode, in general
- *Histogram proportion limit,* $n \to +\infty$

Negative Binomial & DESeq



Negative Binomial & DESeq

```
## sample simulation
r <- 5
p <- 0.6
set.seed(1234)
x <- rnbinom(1000, size = r, prob=(1-p))

## Histogram with density function
h <- hist(x, breaks=200, plot=F)
h$density = h$counts/sum(h$counts)
plot(h,freq=FALSE, col="blue", main="Histogram of NB(5,0.6),  n=1000")
tr <- apply(matrix(unique(x), nc=1), 1,
            function(i){pp <- dnbinom(i, size = r, prob=(1-p));
            segments(i-0.2,pp, i, pp, lty=1, lwd=5, col="red")})
```

Negative Binomial & DESeq

DESeq

Function: typically used to decide whether, for a given gene/class, under multiple conditions, an observed difference in read counts is significant

- RNA-Seq: class → target transcript
- ChIP-Seq: class → binding region

Assumption: read counts follow negative binomial distributions

- Discrete, positive, skewed → (log-)normal is unsuitable
- Small number of replicates → insufficient for rank based or permutation methods

Negative Binomial & DESeq

DESeq Read Counts Model

$$K_{ij} \sim NB(\mu_{ij}, \sigma_{ij}^2)$$

$$\mu_{ij} = q_{i\rho} \times s_j$$

$$\sigma_{ij}^2 = \mu_{ij} + s_j^2 \times v_{i\rho} = \mu_{ij} + s_j^2 \times f(q_{i\rho})$$

$$i = 1, \ldots, n \qquad j = 1, \ldots, m$$

K_{ij}: number of reads in sample j assigned to gene i

$q_{i\rho}$: averaged read counts of gene i in condition ρ

s_j: size factor, adjusting for the coverage or sampling depth of library j

Negative Binomial & DESeq

Parameter Estimates:

$$\hat{s}_j = \text{median}_i \left\{ \frac{k_{ij}}{\left(\prod_{v=1}^{m} k_{iv}\right)^{\frac{1}{m}}} \right\}$$

$$\hat{q}_{i\rho} = \frac{1}{m_\rho} \sum_{j \in \rho} \frac{k_{ij}}{\hat{s}_j}$$

$$\hat{v}_{i\rho} = \frac{1}{m_\rho - 1} \sum_{j \in \rho} \left(\frac{k_{ij}}{\hat{s}_j} - \hat{q}_{i\rho} \right)^2 - \frac{\hat{q}_{i\rho}}{m_\rho} \sum_{j \in \rho} \frac{1}{\hat{s}_j}$$

Note: $\hat{v}_{i\rho}$ is unbiased (proof see *Anders & Huber, Genome Biology 2010, 11:R106*)

Negative Binomial & DESeq

Structure of $\sigma_{ij}^2 = \mu_{ij} + s_j^2 \times v_{i\rho}$:

$$K_{ij}|q_{i\rho} \sim \text{Poisson}(q_{i\rho}s_j)$$

$$\begin{aligned} Var[K_{ij}] &= E\left[Var[K_{ij}|q_{i\rho}]\right] + Var\left[E[K_{ij}|q_{i\rho}]\right] \\ &= E[q_{i\rho}s_j] + Var[q_{i\rho}s_j] \\ &= s_j E[q_{i\rho}] + s_j^2 Var[q_{i\rho}] \end{aligned}$$

$$\Rightarrow \quad \sigma_{ij}^2 = q_{i\rho}s_j + s_j^2 \times v_{i\rho}$$

Negative Binomial & DESeq

Testing for differential expression

$$K_{iA} = \sum_{j:\rho(j)=A} K_{ij} \quad , \quad K_{iB} = \sum_{j:\rho(j)=B} K_{ij}$$

$$K_{iA} \sim NB(\mu_{iA}, \sigma^2_{iA}), \qquad K_{iB} \sim NB(\mu_{iB}, \sigma^2_{iB})$$

$$\mu_{iA} = \sum_{j \in A} q_{iA} \times s_j \,, \qquad \sigma^2_{iA} = \sum_{j \in A} q_{iA} \times s_j + s_j^2 \times f(q_{iA})$$

$$\mu_{iB} = \sum_{j \in B} q_{iB} \times s_j \,, \qquad \sigma^2_{iB} = \sum_{j \in B} q_{iB} \times s_j + s_j^2 \times f(q_{iB})$$

Under the null of equal expression:

$$q_{iA} = q_{iB} = q_{i0} \quad \leftarrow \quad \frac{1}{m} \sum_j \frac{k_{ij}}{s_j}$$

Negative Binomial & DESeq

P-value Calculation:

$$p_i = \frac{1}{\sum_{a+b=k_{is}} p(a,b)} \sum_{\substack{a+b=k_{is} \\ p(a,b) \le p(k_{iA},k_{iB})}} p(a,b)$$

$$p(a,b) = \Pr[K_{iA} = a, K_{iB} = b]$$
$$= \Pr[K_{iA} = a] \times \Pr[K_{iB} = b] \leftarrow \text{independence}$$

$$K_{iS} = K_{iA} + K_{iB}$$

Principal Component Analysis

Goal of PCA:

- Summarize a $n \times p$ data matrix containing p variables by uncorrelated linear combinations of the original p variables.
- These uncorrelated linear combinations are called principal components or principal axes.
- The principal components are ordered; the 1st component has the largest variance, corresponding to the maximum possible portion of the variation in the original data matrix that can be explained by a linear combination of the original p variables.

Principal Component Analysis

Assumption of PCA:

- Relationships among the variables are approximately linear.

Data Preparation:

- Normalization (centering and rescaling) the variable measurements are often needed prior to application of PCA, if the goal of the analysis is to decompose variation in data matrix induced by relationships between variables.

Principal Component Analysis

Simulation 1:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x
data <- cbind(x,y)
data.pca <- prcomp(data, center=T, scale=T)
print(data.pca)
```

```
## Standard deviations:
## [1] 1.414214e+00 8.918918e-16
##
## Rotation:
##         PC1        PC2
## x 0.7071068 -0.7071068
## y 0.7071068  0.7071068
```

Principal Component Analysis

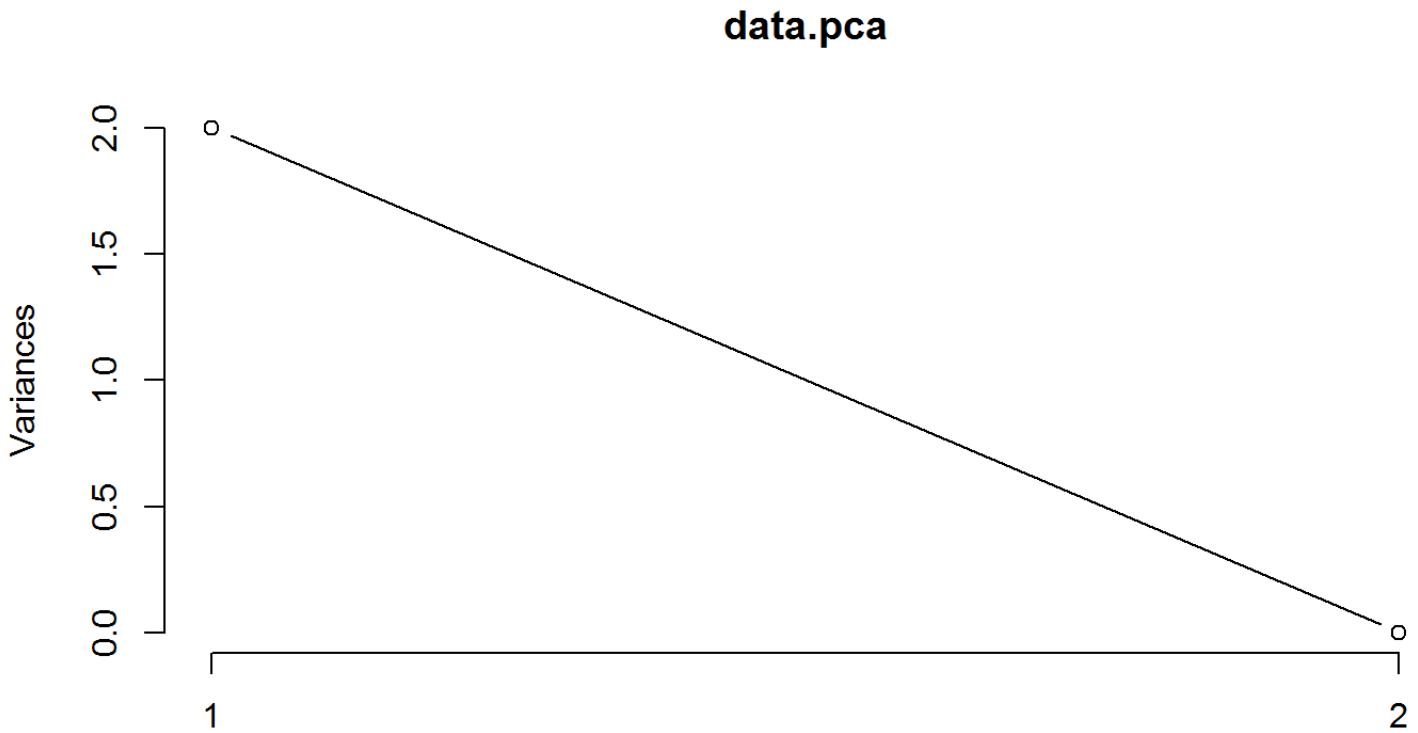
Simulation 1:

```
summary(data.pca)
```

```
## Importance of components:
##                           PC1       PC2
## Standard deviation     1.414 8.919e-16
## Proportion of Variance 1.000 0.000e+00
## Cumulative Proportion  1.000 1.000e+00
```

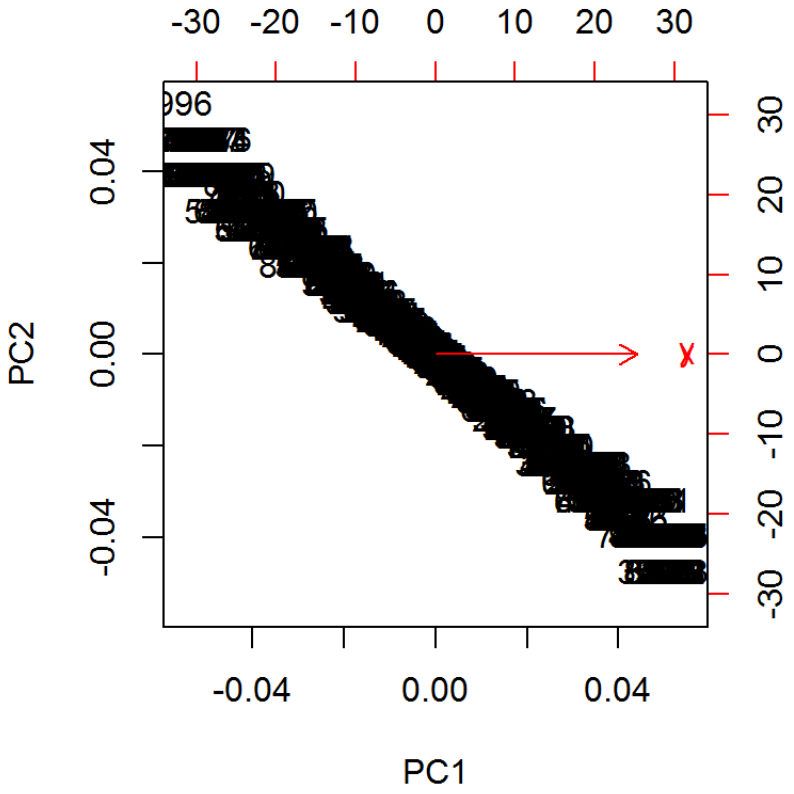
Principal Component Analysis

```
plot(data.pca, type = "l")
```



Principal Component Analysis

```
biplot(data.pca)
```



Principal Component Analysis

Simulation 2:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x
data <- cbind(x,y)
data.pca <- prcomp(data, center=F, scale=F)
print(data.pca)
```

```
## Standard deviations:
## [1] 5.7551213 0.1518755
##
## Rotation:
##           PC1         PC2
## x -0.09822225  0.99516450
## y -0.99516450 -0.09822225
```

Principal Component Analysis

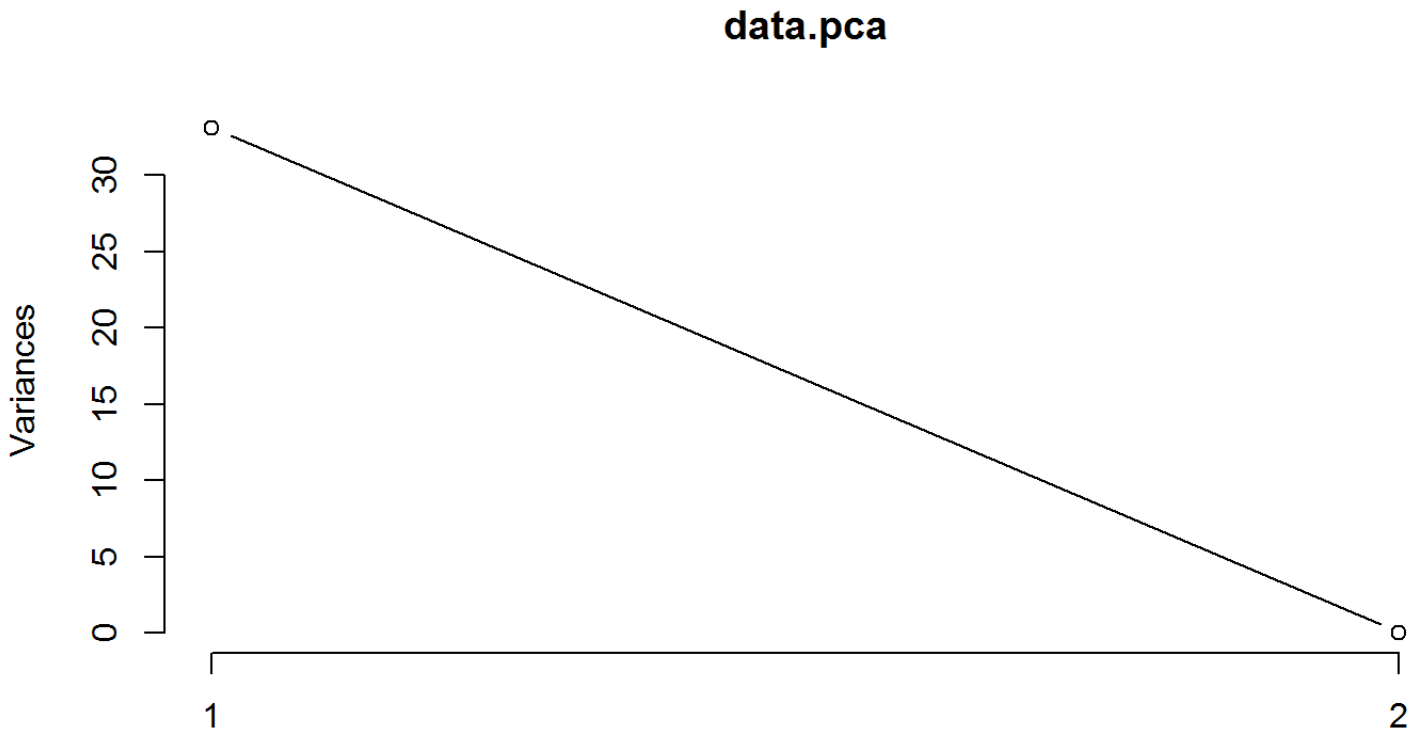
Simulation 2:

```
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2
## Standard deviation     5.7551 0.1519
## Proportion of Variance 0.9993 0.0007
## Cumulative Proportion  0.9993 1.0000
```

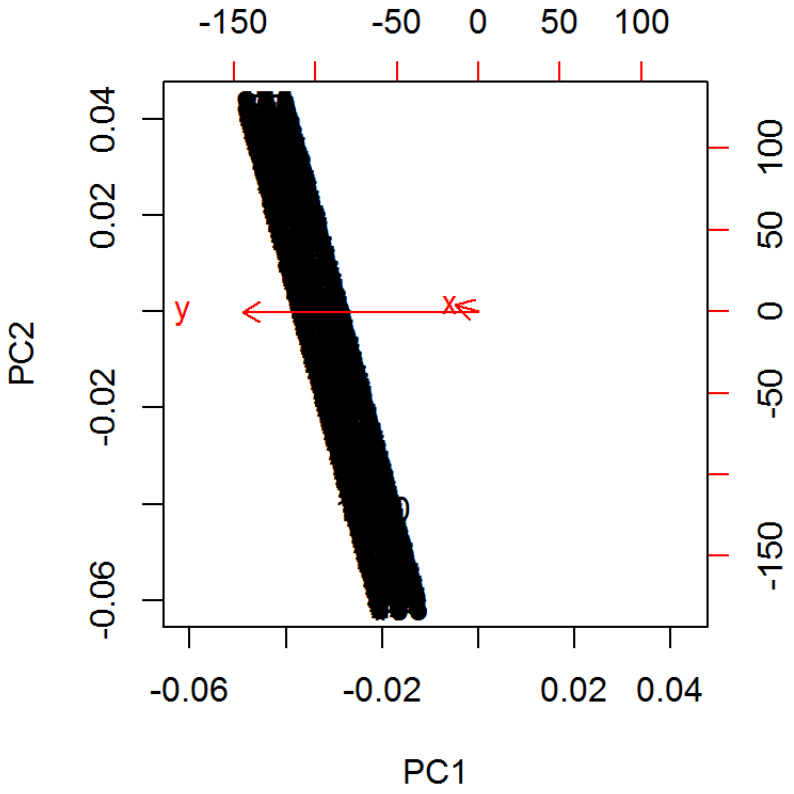
Principal Component Analysis

```
plot(data.pca, type = "l")
```



Principal Component Analysis

```
biplot(data.pca)
```



Principal Component Analysis

Simulation 3:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x
z <- 1+4*x
data <- cbind(x,y,z)
data.pca <- prcomp(data, center=T, scale=T)
print(data.pca)
```

```
## Standard deviations:
## [1] 1.732051e+00 1.059140e-15 5.623615e-17
##
## Rotation:
##         PC1        PC2         PC3
## x 0.5773503  0.8155967 -0.03832384
## y 0.5773503 -0.3746089  0.72548937
## z 0.5773503 -0.4409878 -0.68716553
```

Principal Component Analysis

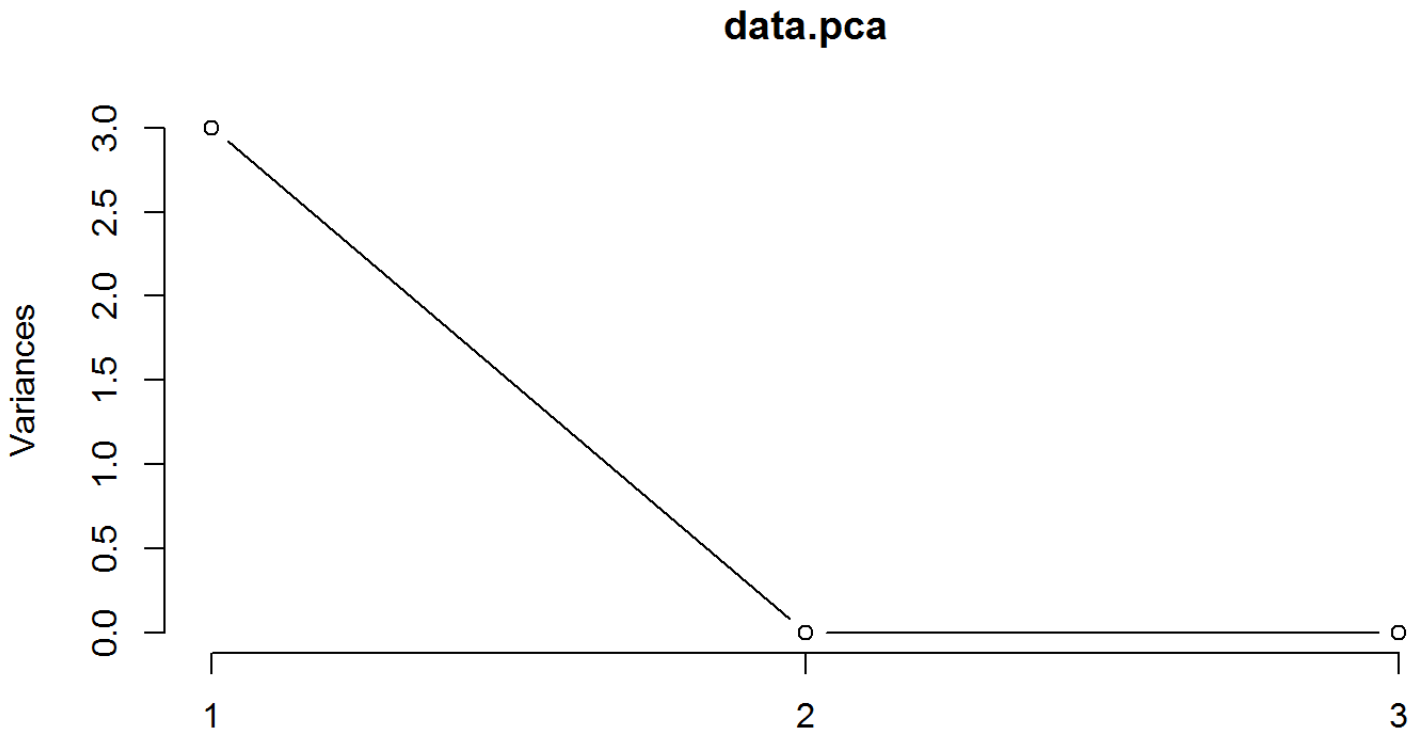
Simulation 3:

```
summary(data.pca)
```

```
## Importance of components:
##                           PC1       PC2       PC3
## Standard deviation     1.732 1.059e-15 5.624e-17
## Proportion of Variance 1.000 0.000e+00 0.000e+00
## Cumulative Proportion  1.000 1.000e+00 1.000e+00
```

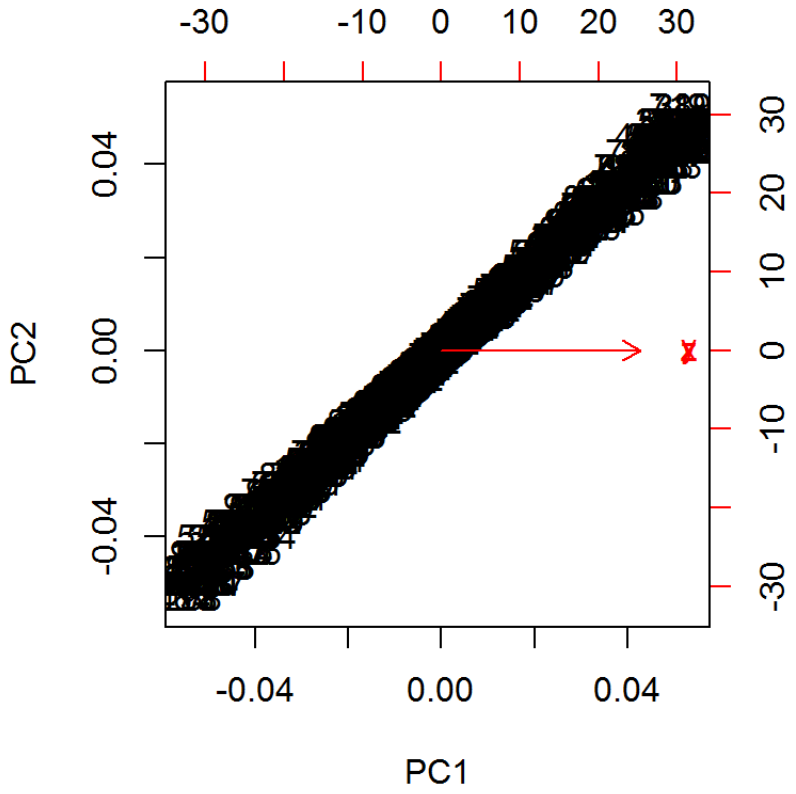
Principal Component Analysis

```
plot(data.pca, type = "l")
```



Principal Component Analysis

```
biplot(data.pca)
```



Principal Component Analysis

Simulation 4:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x
z <- rnorm(1000, 0, 4.5)
data <- cbind(x,y,z)
data.pca <- prcomp(data, center=F, scale=F)
print(data.pca)
```

```
## Standard deviations:
## [1] 5.7566177 4.2673551 0.1518084
##
## Rotation:
##           PC1          PC2          PC3
## x -0.09818521 -0.002283965  0.995165538
## y -0.99458822 -0.033909218 -0.098206074
## z  0.03396958 -0.999422307  0.001057779
```

Principal Component Analysis

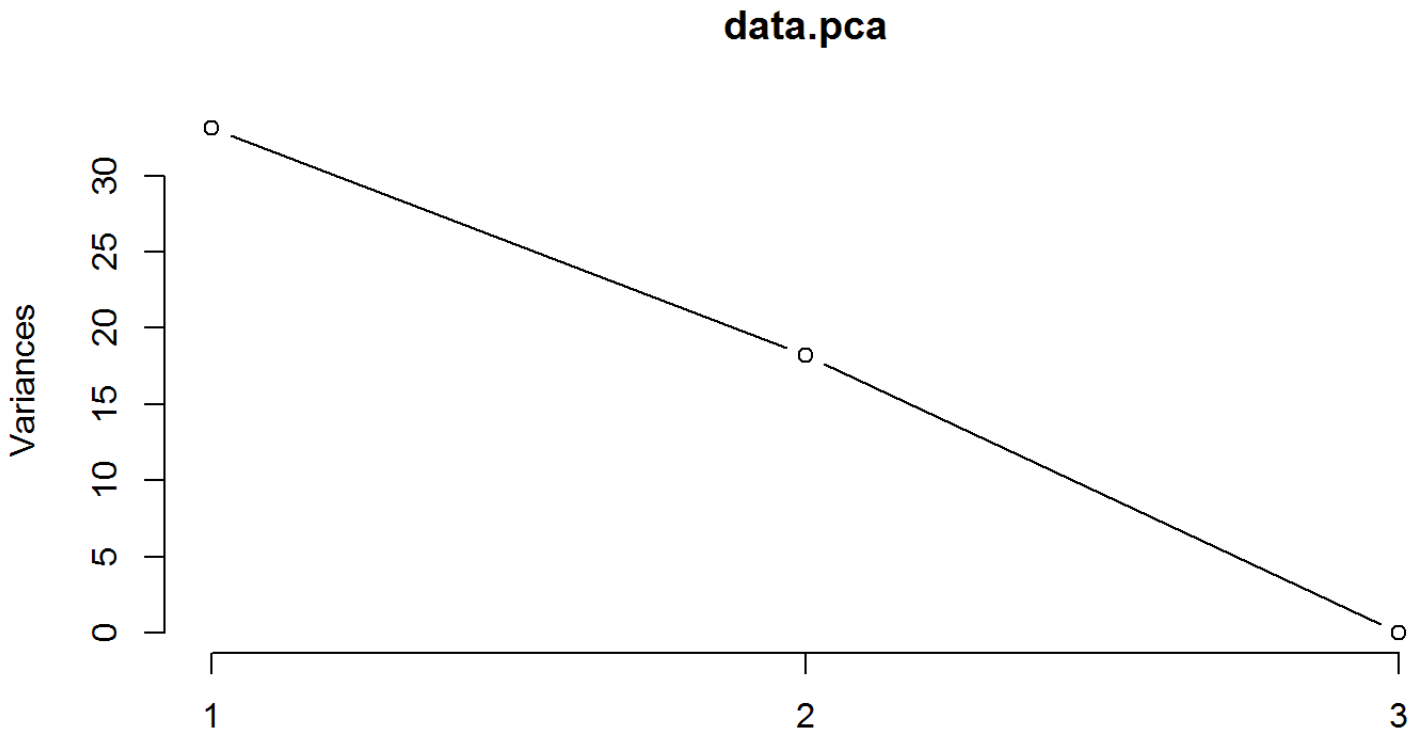
Simulation 4:

```
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3
## Standard deviation     5.7566 4.2674 0.15181
## Proportion of Variance 0.6451 0.3545 0.00045
## Cumulative Proportion  0.6451 0.9996 1.00000
```

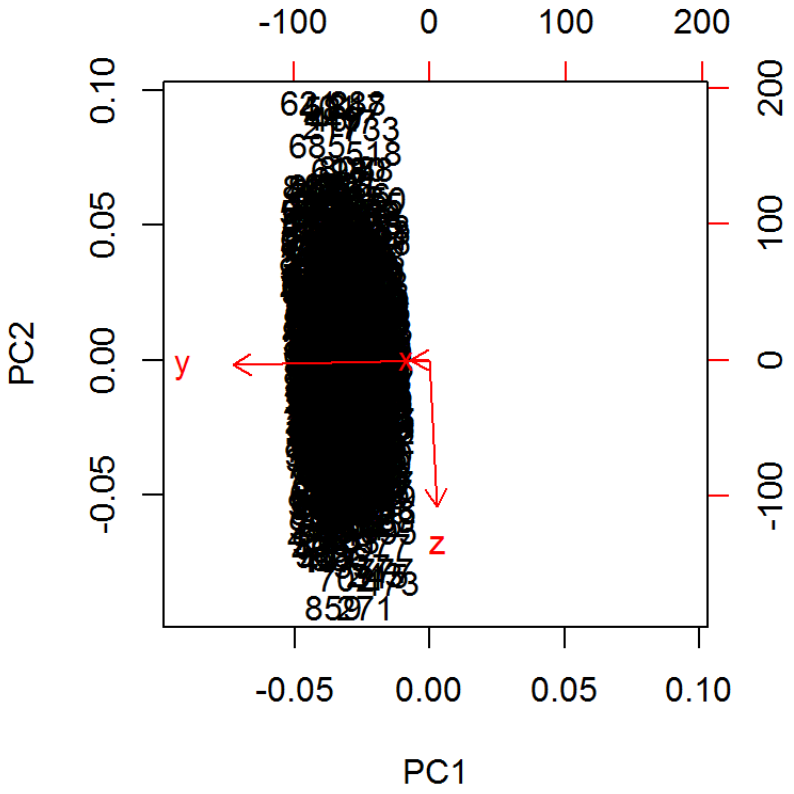
Principal Component Analysis

```
plot(data.pca, type = "l")
```



Principal Component Analysis

```
biplot(data.pca)
```



Principal Component Analysis

Simulation 5:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x
z <- rnorm(1000, 0, 10)
data <- cbind(x,y,z)
data.pca <- prcomp(data, center=F, scale=F)
print(data.pca)
```

```
## Standard deviations:
## [1] 9.4886659 5.7531846 0.1518085
##
## Rotation:
##            PC1        PC2           PC3
## x  0.002416151 0.09817756  0.9951659806
## y  0.019642803 0.99497222 -0.0982061345
## z -0.999804142 0.01978513  0.0004755204
```

Principal Component Analysis

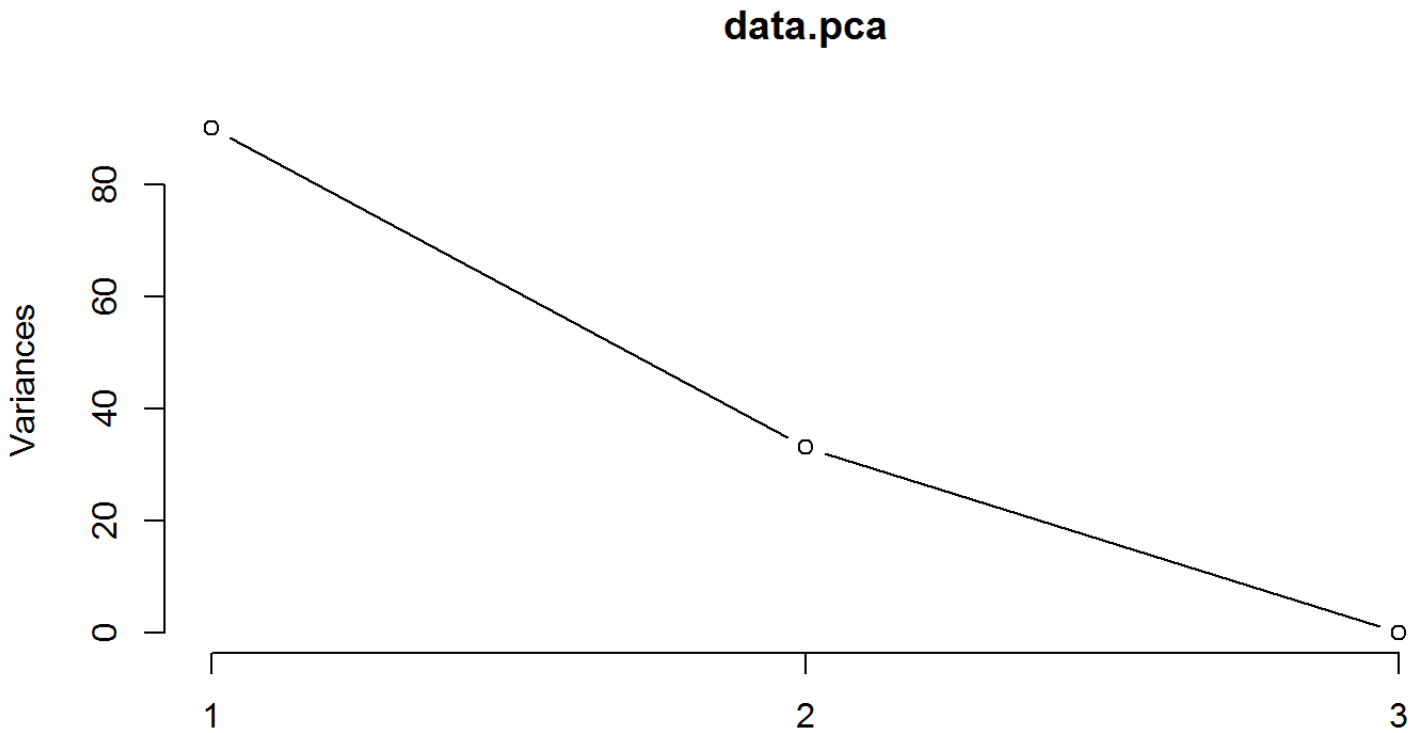
Simulation 5:

```
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3
## Standard deviation     9.4887 5.7532 0.15181
## Proportion of Variance 0.7311 0.2688 0.00019
## Cumulative Proportion  0.7311 0.9998 1.00000
```

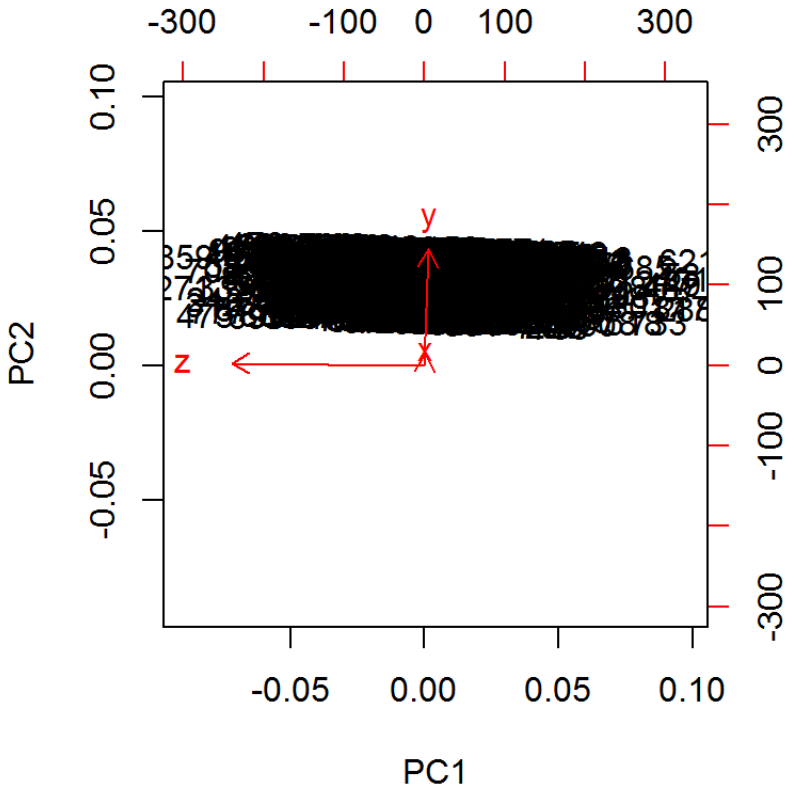
Principal Component Analysis

```
plot(data.pca, type = "l")
```



Principal Component Analysis

```
biplot(data.pca)
```



Principal Component Analysis

Simulation 6:

```
set.seed(1234)
x <- runif(1000)
y <- 3+5*x
z <- rnorm(1000, 0, 10)
data <- cbind(x,y,z)
data.pca <- prcomp(data, center=T, scale=T)
print(data.pca)
```

```
## Standard deviations:
## [1] 1.415028e+00 9.988468e-01 8.936546e-16
##
## Rotation:
##           PC1        PC2           PC3
## x  0.70629508 0.03387124  7.071068e-01
## y  0.70629508 0.03387124 -7.071068e-01
## z -0.04790117 0.99885208 -3.816392e-17
```

Principal Component Analysis

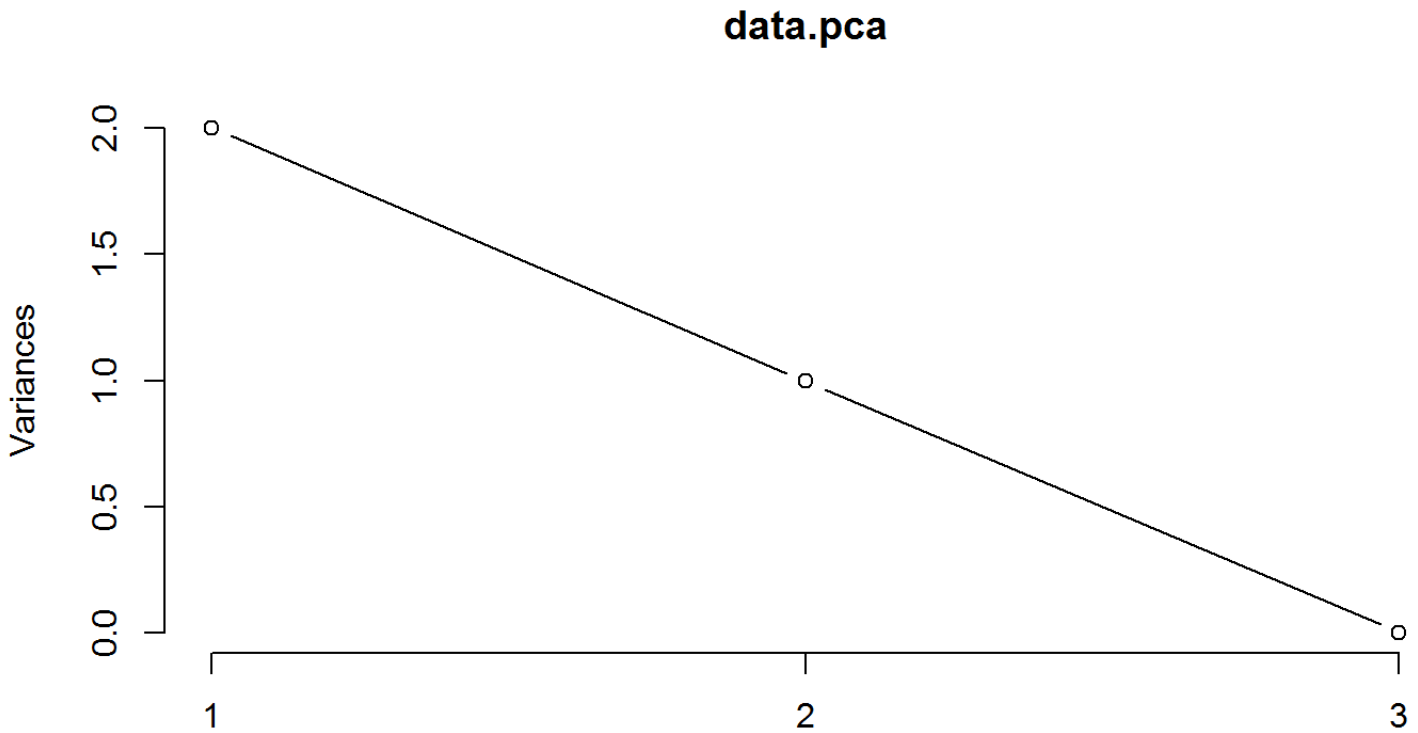
Simulation 6:

```
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2       PC3
## Standard deviation     1.4150 0.9988 8.937e-16
## Proportion of Variance 0.6674 0.3326 0.000e+00
## Cumulative Proportion  0.6674 1.0000 1.000e+00
```

Principal Component Analysis

```
plot(data.pca, type = "l")
```



Principal Component Analysis

```
biplot(data.pca)
```