

Linear Regression

Objectives

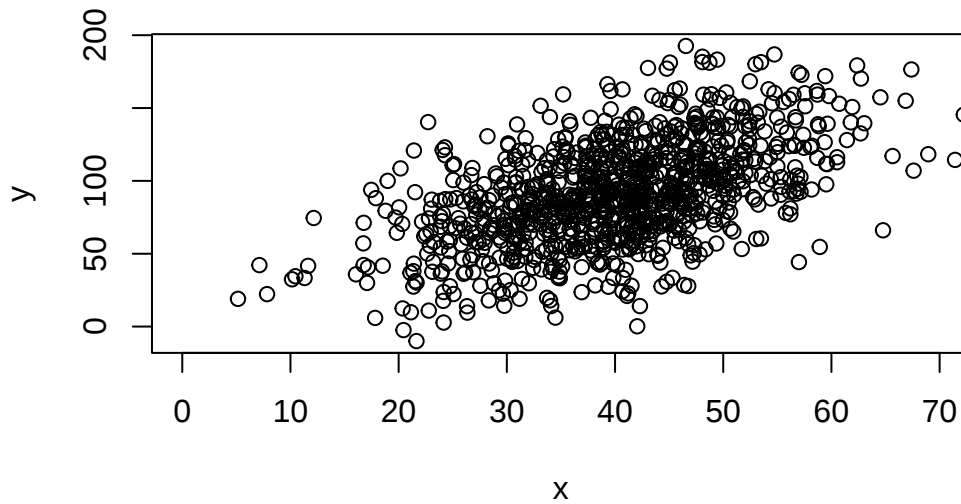
- fundamentals
- assumptions
- lm / glm functions
- confidence intervals
- case study

Why model data?

In ecology, we use models to,

- describe relationships among outcomes and processes
- to estimate hidden (latent) processes
- predict unobserved values
- forecast future outcomes, such as responses to management

Linear Regression (motivation)



Linear Regression (Equation)

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Annotations for the equation:

- observed value for the i th term of Y (points to y_i)
- constant term where $x_i = 0$ (points to β_0)
- slope of X term representing the change in Y for a unit change in X (points to β_1)
- observed value for the i th term of X (points to x_i)
- error term representing $y_i - \bar{y}$ (points to ε_i)

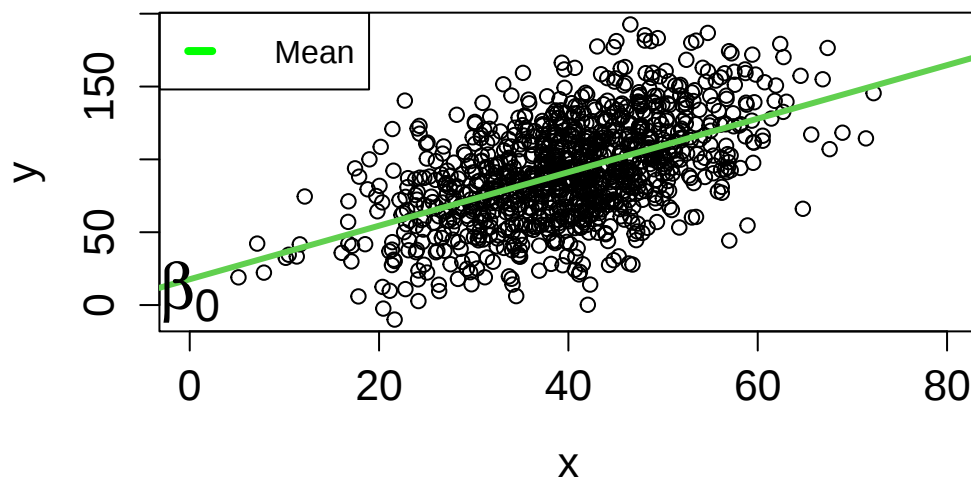
Linear Regression (Equation 2)

$$y_i \sim \text{Normal}(\mu_i, \sigma) \mu_i = \beta_0 + \beta_1 \times x_i$$

Linear Regression Line 1

$$\hat{\mu}_i = 9.06 + 2 \times x_i$$

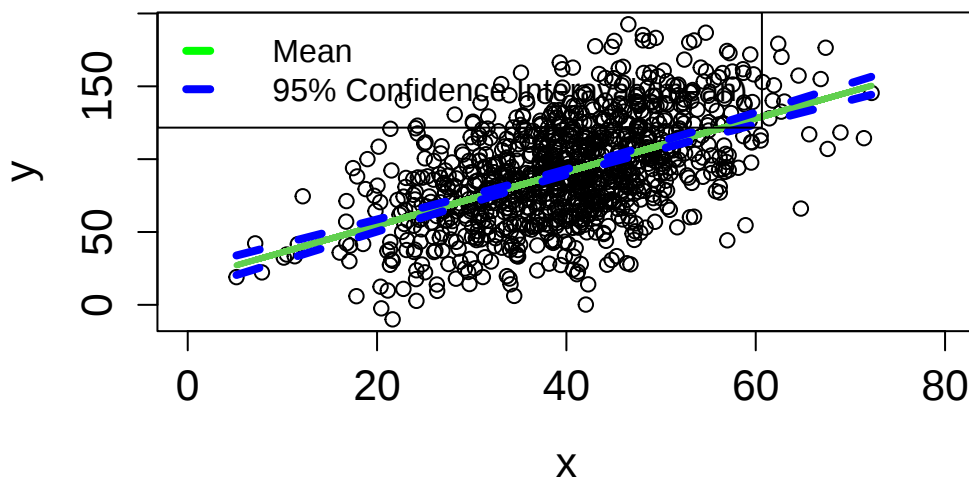
Intercept ($\hat{\beta}_0$) = 9.06, Slope of x ($\hat{\beta}_1$) = 2.0



Linear Regression Line 2

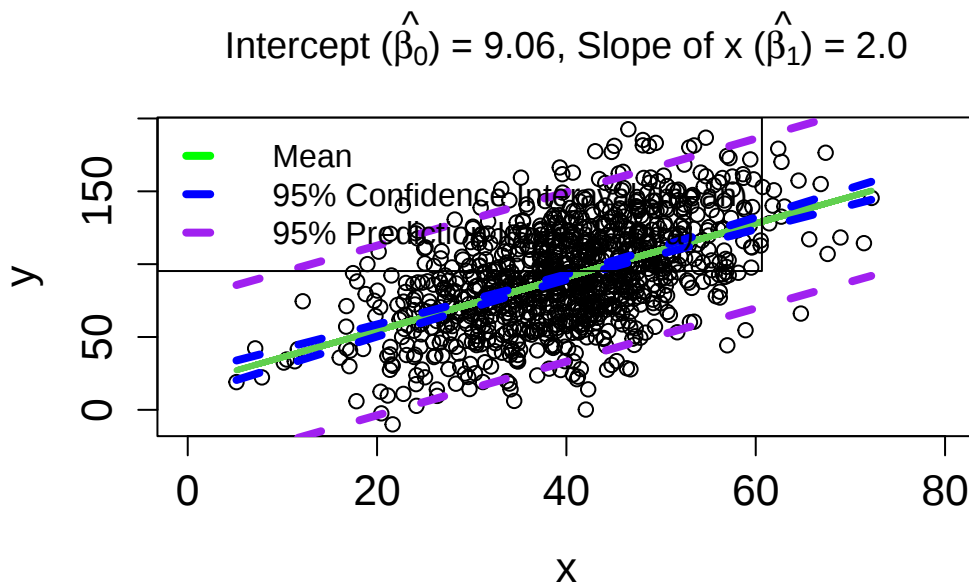
$$\hat{\mu}_i = 9.06 + 2 \times x_i$$

Intercept ($\hat{\beta}_0$) = 9.06, Slope of x ($\hat{\beta}_1$) = 2.0



Linear Regression Line 3

$$\hat{\mu}_i = 9.06 + 2 \times x_i$$



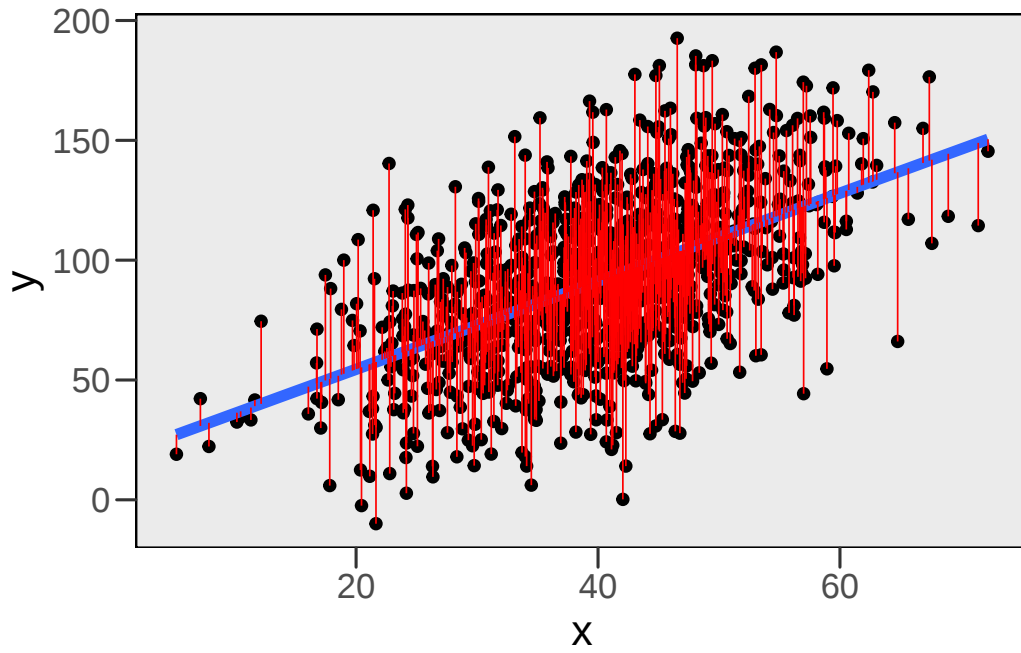
Residuals

```
library(ggplot2)
library(ggthemes)
m1=broom::augment(lm.out)

ggplot(m1, aes(x, y)) +
  geom_point() +
  stat_smooth(method = lm, se = FALSE, linewidth=2) +
  geom_segment(aes(xend = x, yend = .fitted), color = "red", size = 0.3)+theme_base()
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.

`geom\_smooth()` using formula = 'y ~ x'

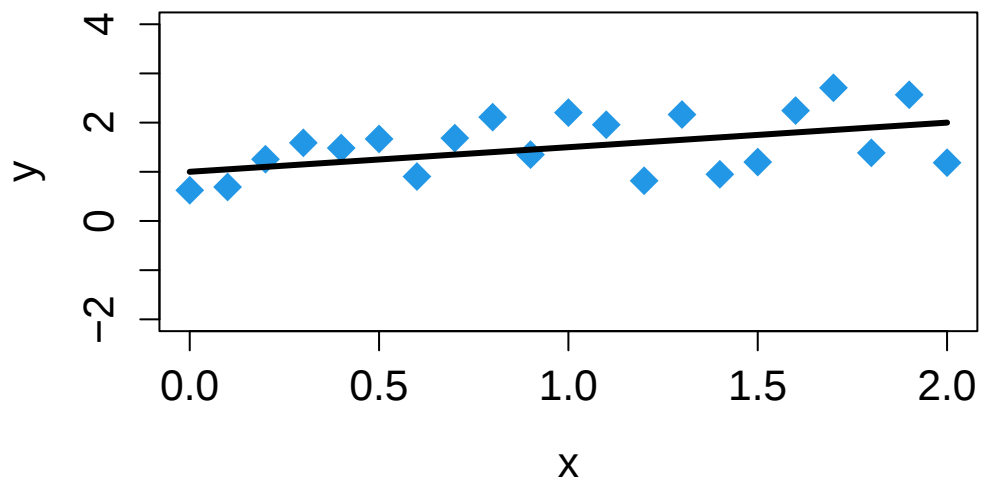


One Sample

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

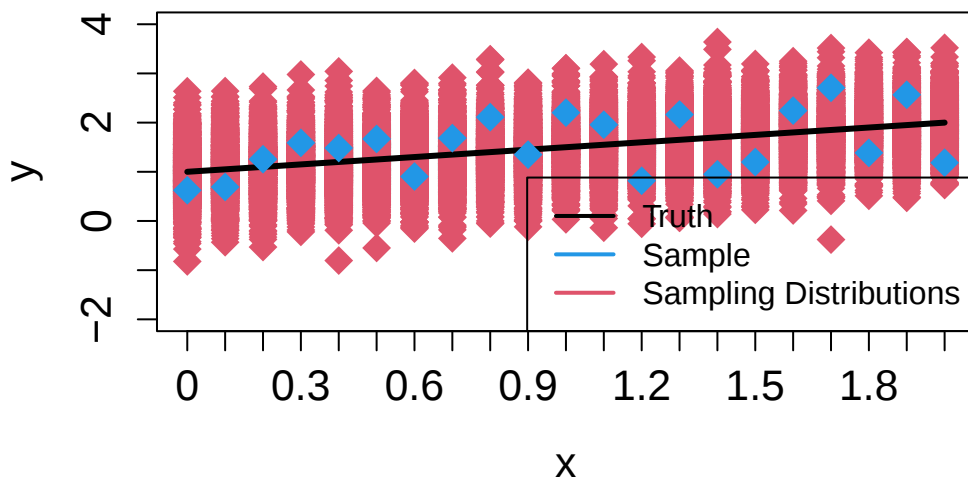
$$\mu_i = \beta_0 + \beta_1 x_i$$

$$\mu_i = 1 + 0.5 \times x_i$$



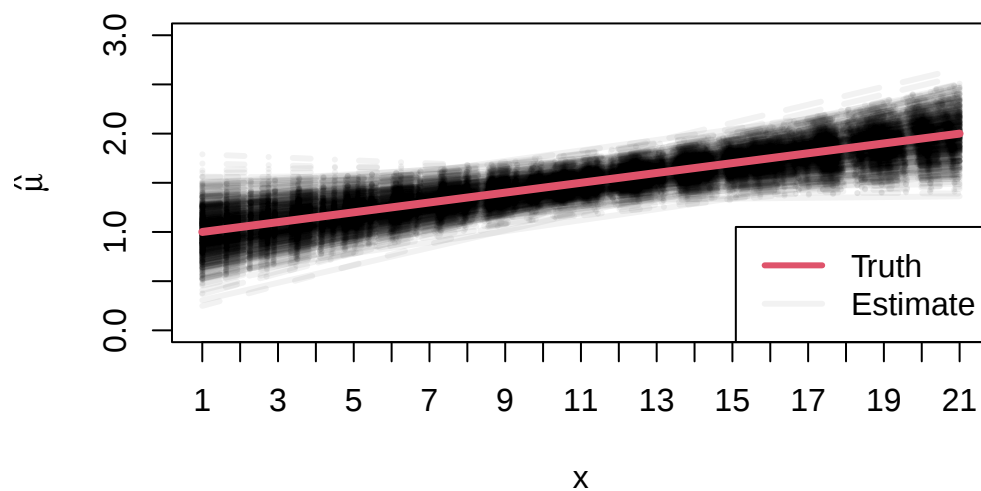
Sampling Distributions of y

$$\mu_i = 1 + 0.5 \times x_i$$



Sampling Distributions of $\hat{\mu}$

$$\mu_i = 1 + 0.5 \times x_i$$



Assumptions

Independence of the errors

$$\text{Correlation}(\epsilon_i, \epsilon_j) = 0, \forall \text{ pairs of } i \text{ and } j$$

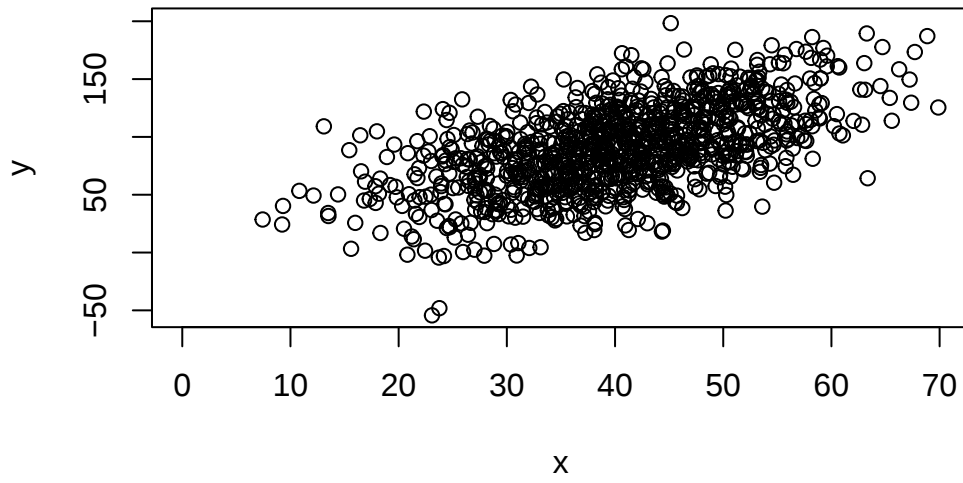
This means that knowing how far observation i will be from the true regression line tells us nothing about how far observation j will be from the regression line.

Assumptions

Homogeneity of the variance

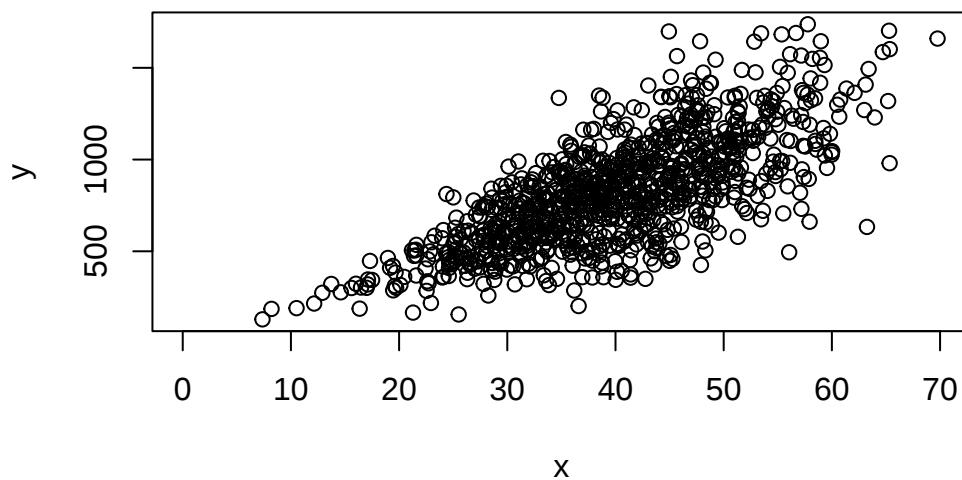
$$\text{var}(\epsilon_i) = \sigma^2$$

Constancy in the scatter of observations above and below the line, going left to right.



Assumptions

Heteroskedasticity



Assumptions

Linearity

$$E[y_i|x_i] = \mu_i = \beta_0 + \beta_1 \times x_i$$

The hypothesis about the variables included in the model (e.g., x_i) characterizes the mean well.

Assumptions

Normality

$$\epsilon_i \sim \text{Normal}(0, \sigma)$$

Each i^{th} residual

- comes from a Normal distribution with a mean of zero
- is symmetrically distributed around zero
- varies around zero by σ , which is the same for each residual.

Assumption Violations

Robustness

Linearity and constant variance are often more important than the assumption of normality (see e.g., [Knief & Forstmeier, 2021](#) and references therein)

This is especially true for large sample sizes

Intercept-Only Model

$$y_i \sim \text{Normal}(\mu_i, \sigma^2) \mu_i = \beta_0$$

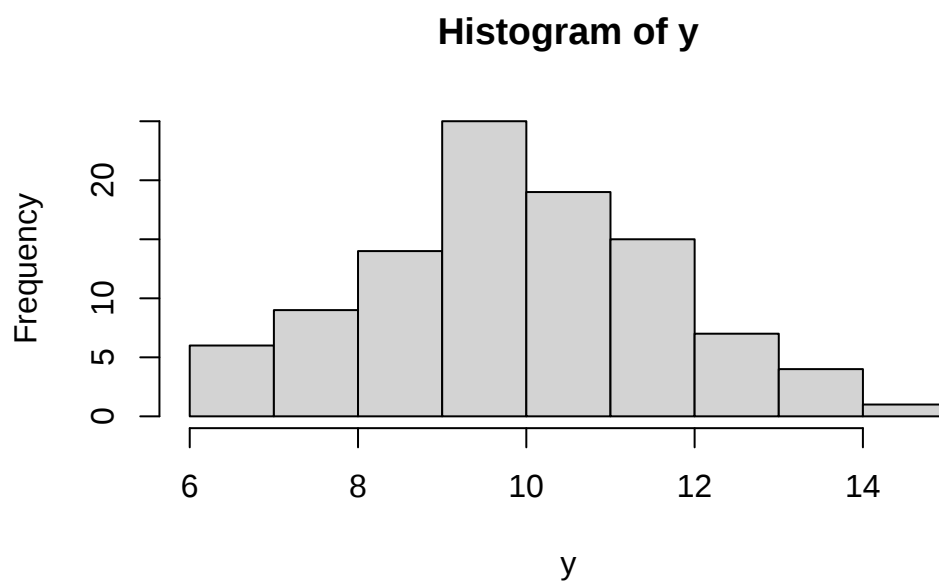
Generate Data

```
#Setup parameters
n = 100 # sample size
beta0 = 10 # true mean
sigma = 2 # true std.dev

# Simulate a data set of observations
set.seed(43243)
y = rnorm(n, mean = beta0, sd = sigma)
```

Visualize Intercept-Only Model

```
hist(y)
```



Fit Intercept-Only Model

```
# Fit model/hypothesis using ordinary least squares or maximum likelihood
model1.0 = lm(y~1)

model1.1 = glm(y~1)

model1.2 = glm(y~1, family=gaussian(link = identity))

# Compare Results
data.frame(intercept=c(model1.0$coefficients,model1.1$coefficients, model1.2$coefficients)
           SE = c(summary(model1.0)$coefficients[, 2], summary(model1.1)$coefficients[, 2]
           )
```

	intercept	SE
1	9.913821	0.1765343
2	9.913821	0.1765343
3	9.913821	0.1765343

Fit Intercept-Only Model

```
# Summary of model results
summary(model1.0)
```

Call:

```
lm(formula = y ~ 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.7815	-1.2733	-0.0581	1.1558	4.4979

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	9.9138	0.1765	56.16	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.765 on 99 degrees of freedom

Fitted-values

```
#Predict response for all data
preds = predict(model1.0, se.fit = TRUE)
preds
```

\$fit

1	2	3	4	5	6	7	8
9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821
9	10	11	12	13	14	15	16
9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821
17	18	19	20	21	22	23	24
9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821
25	26	27	28	29	30	31	32
9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821
33	34	35	36	37	38	39	40
9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821	9.913821
41	42	43	44	45	46	47	48

```

9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    49      50      51      52      53      54      55      56
9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    57      58      59      60      61      62      63      64
9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    65      66      67      68      69      70      71      72
9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    73      74      75      76      77      78      79      80
9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    81      82      83      84      85      86      87      88
9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    89      90      91      92      93      94      95      96
9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821 9.913821
    97      98      99     100
9.913821 9.913821 9.913821 9.913821

```

```
$se.fit
```

```

[1] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[8] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[15] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[22] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[29] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[36] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[43] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[50] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[57] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[64] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[71] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[78] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[85] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[92] 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343 0.1765343
[99] 0.1765343 0.1765343

```

```
$df
```

```
[1] 99
```

```
$residual.scale
```

```
[1] 1.765343
```

What is a CI?

. . .

“A confidence interval for a parameter is an interval computed using sample data ...

. . .

“... by a method that will contain the parameter for a specified proportion of all samples.

. . .

The success rate (proportion of all samples whose intervals contain the parameter) is known as the confidence level.” [R. H. Lock et al. \(2020\)](#)

What is a CI?

Key

- the parameter we are trying to estimate is a fixed unknown (i.e., it is not varying across samples)
- the endpoints of our confidence interval are random and will change every time we collect a new data set (the endpoints themselves actually have a sampling distribution!)

[More at Stats4Ecologists](#)

What is a CI?

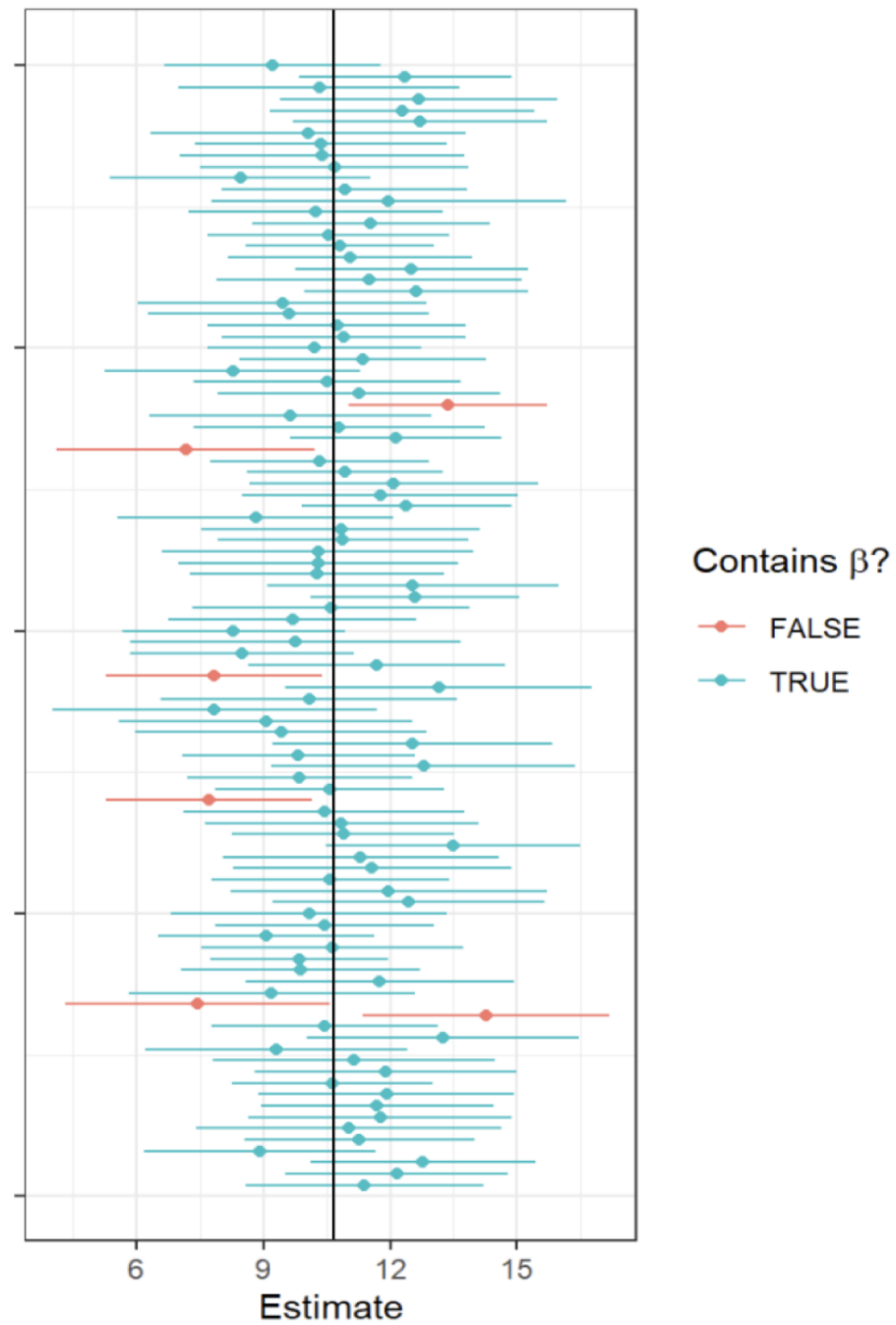


Figure 1.14: First 100 CI's associated with the simulated data. Points give estimates of the slope parameter and lines indicate 95% CIs. Green intervals capture the true parameter value and red intervals do not

Confidence Intervals

Normal Approximation

Assume the sampling distribution of the parameter is Normal

```
# Get 90% confidence intervals (Type I error = 0.1)
c(
  (preds$fit + preds$se.fit*qnorm(0.05))[1],
  (preds$fit + preds$se.fit*qnorm(0.95))[1]
)
```

```
      1      1
9.623448 10.204195
```

Use the t-distribution instead

```
c(
  (preds$fit + preds$se.fit*qt(0.05,df=(n-1)))[1],
  (preds$fit + preds$se.fit*qt(0.95,df=(n-1)))[1]
)
```

```
      1      1
9.620705 10.206937
```

```
CI.Normal=confint(model1.0, level=0.9)
CI.Normal
```

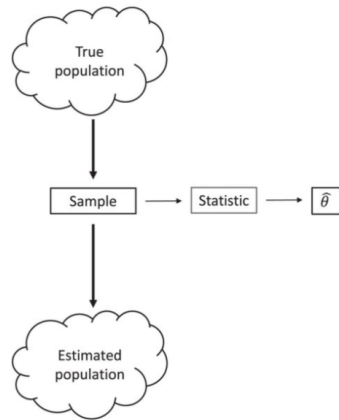
```
      5 %      95 %
(Intercept) 9.620705 10.20694
```

Bootstrapping

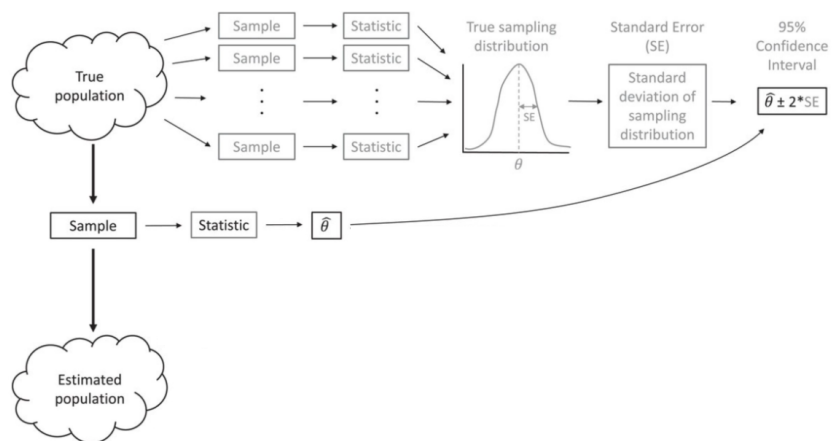
Instead of relying on the 95% intervals from an assumed normal/t distribution, we will create an empirical distribution by resampling our data.

See, [Stats4Ecologists](#)

Bootstrapping (idea)



Bootstrapping (idea)



Bootstrapping (idea)

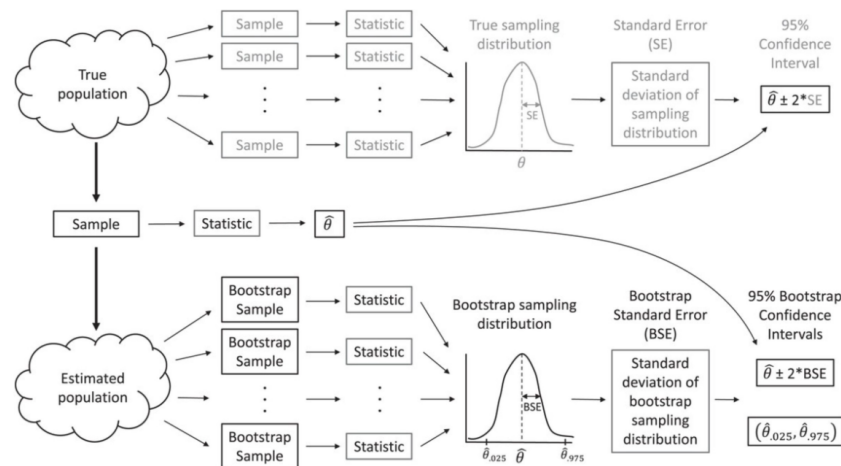


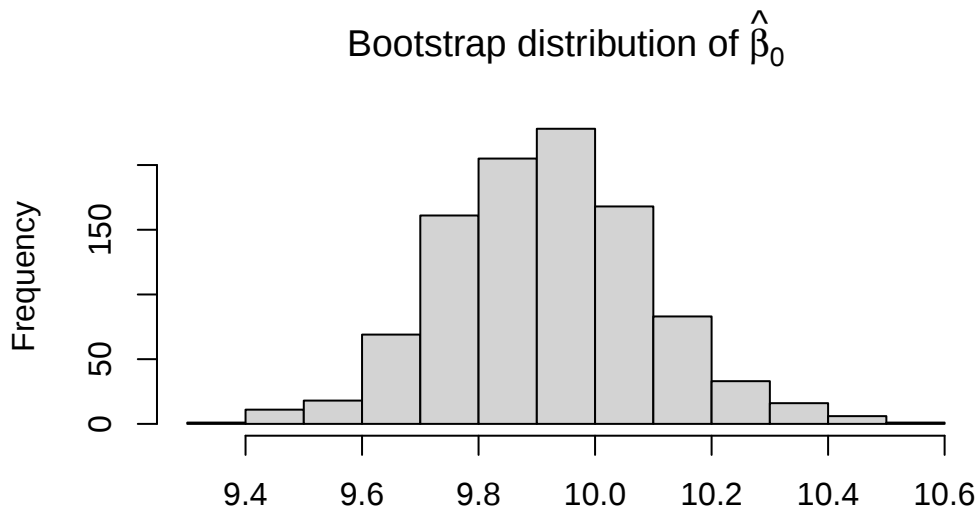
FIGURE 2.4: The bootstrap allows us to estimate characteristics of the sampling distribution (e.g., its standard deviation) by repeatedly sampling from an estimated population. Figure from Fieberg, J. R., Vitense, K., and Johnson, D. H. (2020). Resampling-based methods for biologists. PeerJ, 8, e9089. DOI: 10.7717/peerj.9089/fig-2

Bootstrapping (code)

```
# Setup
nboot <- 1000 # number of bootstrap samples
nobs <- length(y)
bootcoefs <- rep(NA, nboot)
# Start loop
for(i in 1:nboot){
  set.seed(43243+i)
  # Create bootstrap data set by sampling original observations w/ replacement
  bootdat <- y[sample(1:nobs, nobs, replace=TRUE)]
  # Calculate bootstrap statistic
  glmboot <- glm(bootdat ~ 1)
  bootcoefs[i] <- coef(glmboot)
}
```

Bootstrapping (code)

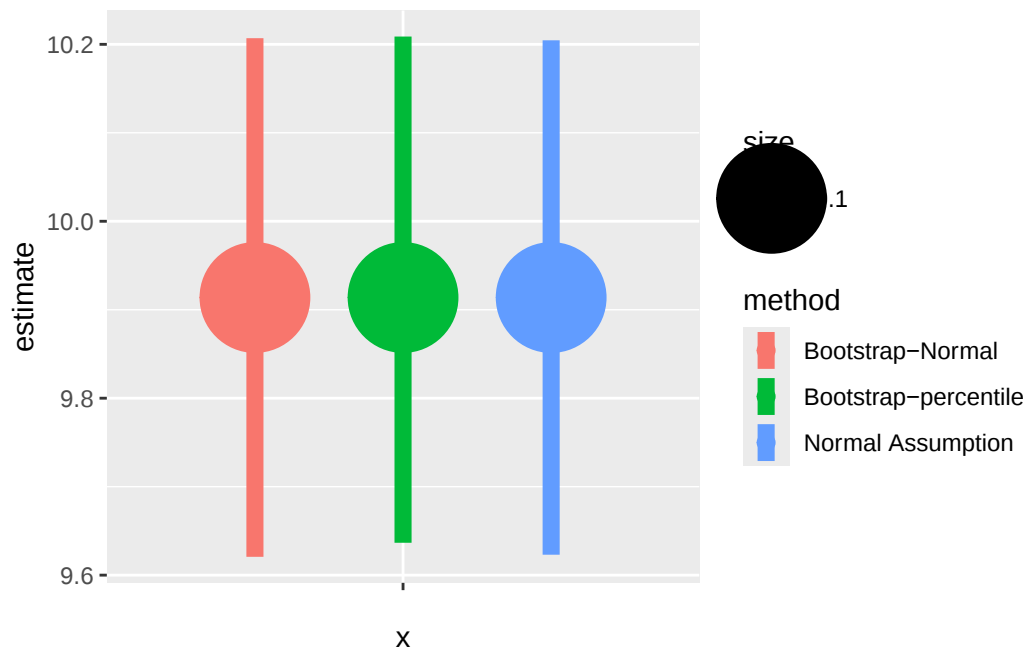
```
par(mfrow = c(1, 1))  
hist(bootcoefs, main = expression(paste("Bootstrap distribution of ", hat(beta)[0])), xlab =
```



Bootstrapping (code)

```
# Calculate bootstrap standard errors  
boot.se = sd(bootcoefs)  
  
# bootstrap-normal CI  
boot.normal = c(  
  (preds$fit+boot.se*qnorm(0.05))[1],  
  (preds$fit+boot.se*qnorm(0.95))[1]  
)  
  
# bootstrap percentile  
confdat.boot.pct <- quantile(bootcoefs, probs = c(0.05, 0.95))
```

Comparison



Prairie Dog Calling



y = # of calls per 5 minute at a prairie dog colony

temp = temperature, degrees F

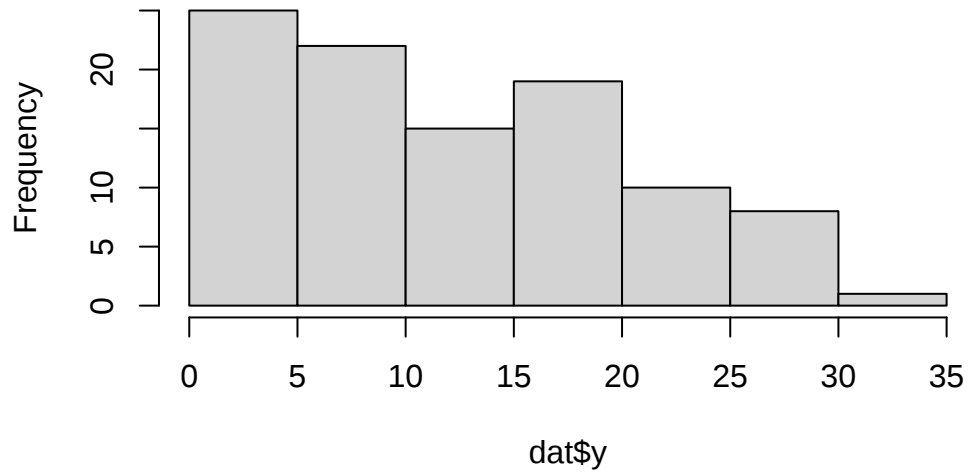
dist.human = distance of nearest human activity to colony

```
head(dat)
```

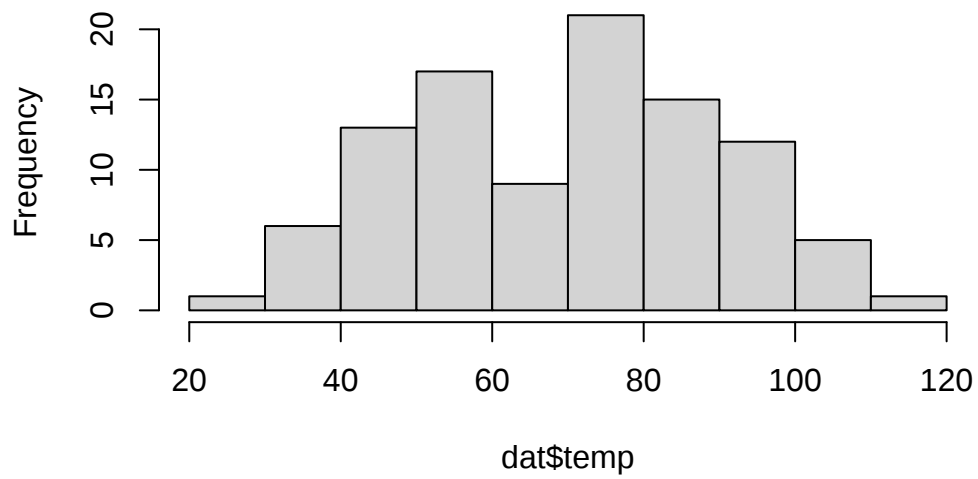
	y	temp	dist.human
1	0.38129119	39.96298	11.75578
2	0.57770546	47.95317	14.63583
3	0.00347806	56.13182	22.89228
4	0.00000000	29.76491	21.14168
5	2.67643282	40.25998	32.41442
6	1.71519689	38.31469	35.99573

Exploratory

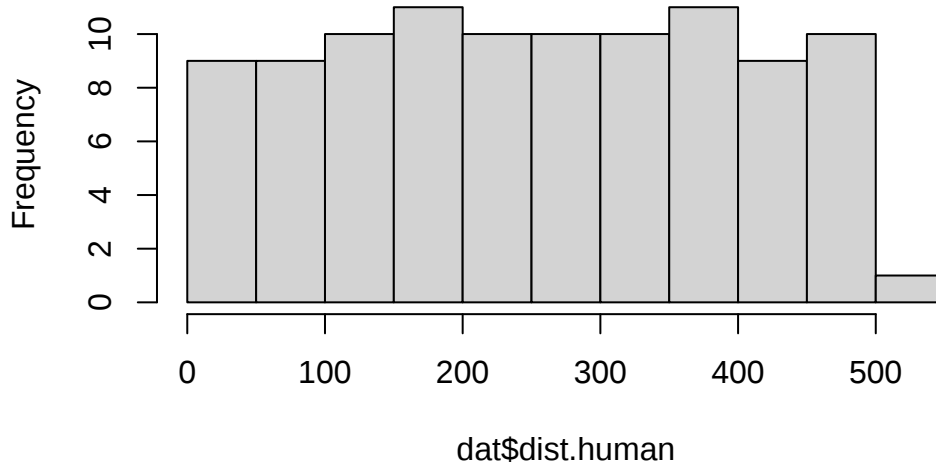
Histogram of dat\$y



Histogram of dat\$temp



Histogram of dat\$dist.human



Model Fitting 1

```
model = lm(y ~ temp + dist.human, data=dat)
summary(model)
```

Call:

```
lm(formula = y ~ temp + dist.human, data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.6714	-1.6690	-0.1506	1.4910	5.0041

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.312197	1.074281	-7.737	9.65e-12 ***
temp	0.156393	0.024605	6.356	6.75e-09 ***
dist.human	0.036010	0.003388	10.629	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.338 on 97 degrees of freedom
 Multiple R-squared: 0.9233, Adjusted R-squared: 0.9217
 F-statistic: 583.8 on 2 and 97 DF, p-value: < 2.2e-16

Model Notation 1

```
equatiomatic::extract_eq(model)
```

$$y = \alpha + \beta_1(\text{temp}) + \beta_2(\text{dist.human}) + \epsilon \quad (1)$$

```
equatiomatic::extract_eq(model, use_coefs = TRUE)
```

$$\hat{y} = -8.31 + 0.16(\text{temp}) + 0.04(\text{dist.human}) \quad (2)$$

Model Fitting 2

Mean-center the intercept. Slopes do not change.

```
model = lm(y ~ I(temp - mean(temp)) + dist.human, data=dat)
summary(model)
```

Call:

```
lm(formula = y ~ I(temp - mean(temp)) + dist.human, data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.6714	-1.6690	-0.1506	1.4910	5.0041

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.614311	0.895284	2.920	0.00435 **
I(temp - mean(temp))	0.156393	0.024605	6.356	6.75e-09 ***
dist.human	0.036010	0.003388	10.629	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.338 on 97 degrees of freedom
Multiple R-squared: 0.9233, Adjusted R-squared: 0.9217
F-statistic: 583.8 on 2 and 97 DF, p-value: < 2.2e-16

Normalizing x

Mean-center and standardize by std. deviation

```
dat$temp.sc = scale(dat$temp, center=TRUE, scale = TRUE)  
dat$dist.sc = scale(dat$dist.human, center=TRUE, scale = TRUE)
```

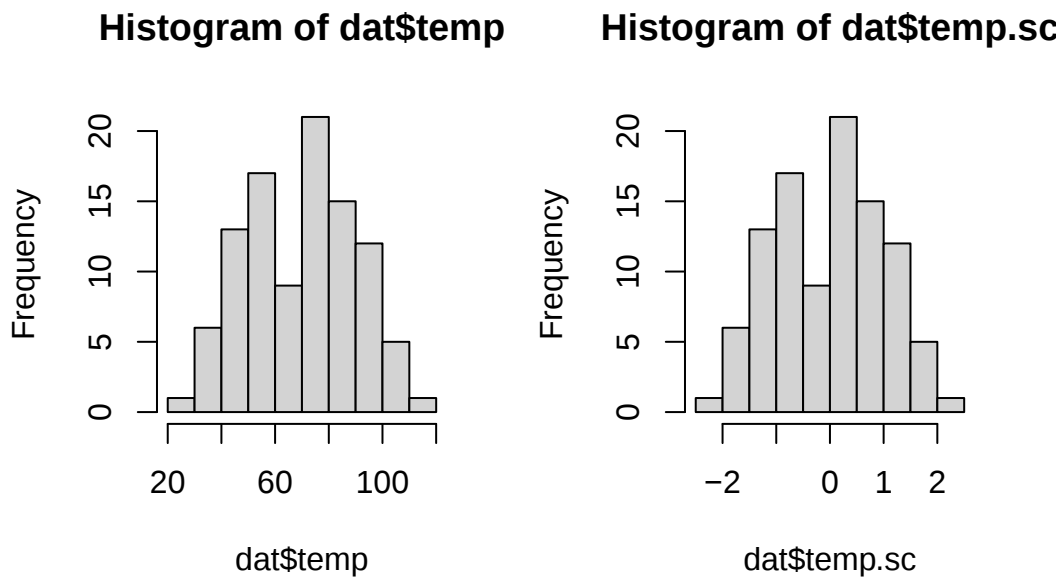
```
mean(dat$temp.sc)
```

```
[1] 1.40337e-16
```

```
sd(dat$temp.sc)
```

```
[1] 1
```

Comparison



Model Fitting 3

What does the slope mean now?

$$\mu_i = \beta_0 + \beta_1 \times 1 + \beta_2 \times 0$$

```
model = lm(y ~ temp.sc + dist.sc, data=dat)
summary(model)
```

Call:

```
lm(formula = y ~ temp.sc + dist.sc, data = dat)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-5.6714	-1.6690	-0.1506	1.4910	5.0041

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	11.8002	0.2338	50.471	< 2e-16 ***
temp.sc	3.0958	0.4871	6.356	6.75e-09 ***
dist.sc	5.1771	0.4871	10.629	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.338 on 97 degrees of freedom

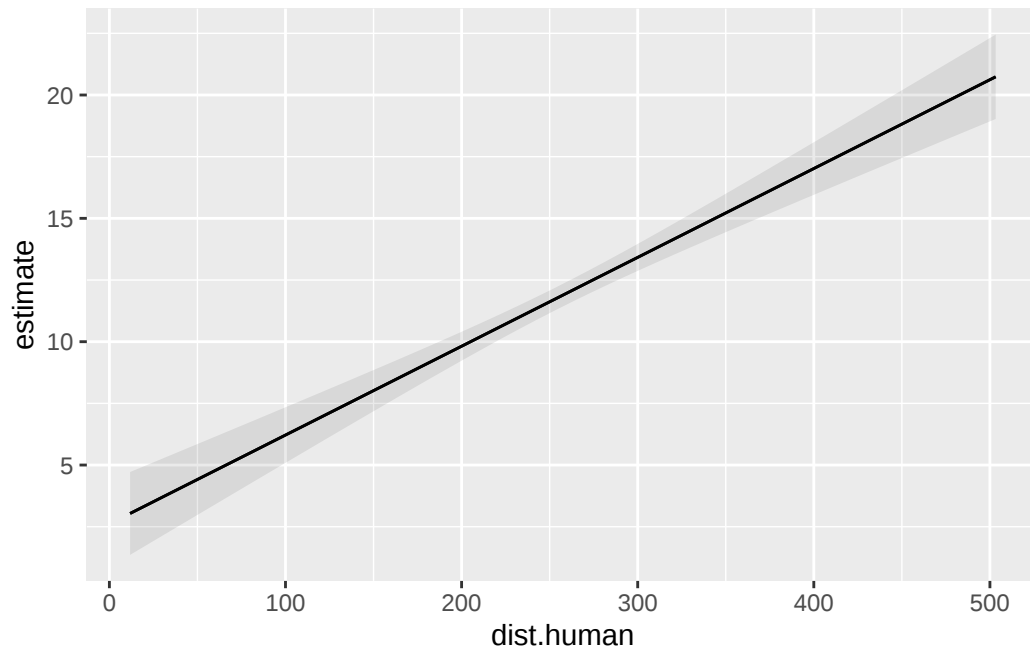
Multiple R-squared: 0.9233, Adjusted R-squared: 0.9217

F-statistic: 583.8 on 2 and 97 DF, p-value: < 2.2e-16

Marginal Predictions

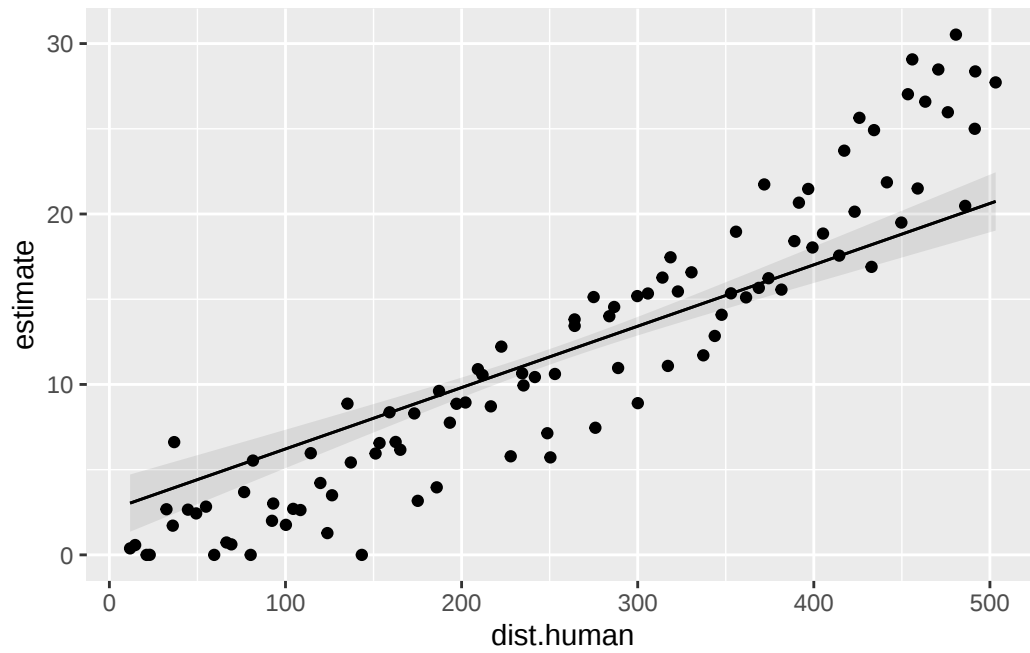
$$\hat{\mu}_i = \hat{\beta}_0 + \hat{\beta}_1 \times 0 + \hat{\beta}_2 \times \text{dist.human}_i$$

```
# Plot marginal predictions of dist.human
marginaleffects::plot_predictions(model, condition=c("dist.human"))
```



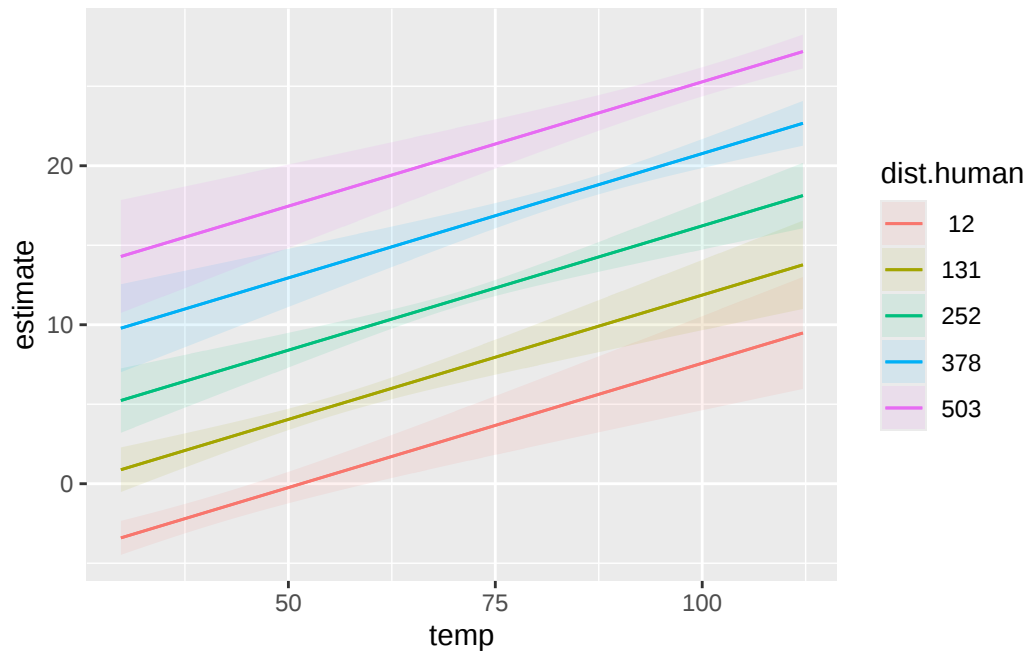
Add data points

```
# Plot marginal predictions of dist.human
plot1 = marginaleffects::plot_predictions(model, condition=c("dist.human"))
plot1+geom_point(data=dat, aes(x=dist.human,y=y))
```



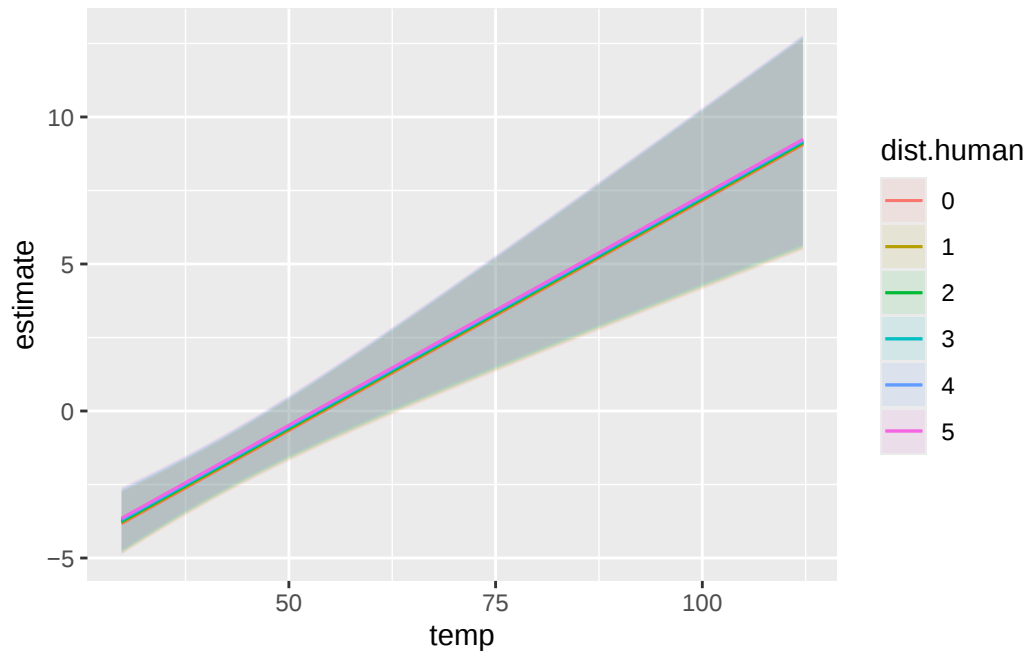
Combined Predictions

```
marginaleffects::plot_predictions(model, condition=list("temp","dist.human"))
```



Combined Predictions

```
marginaleffects::plot_predictions(model, condition=list("temp","dist.human" = 0:5))
```



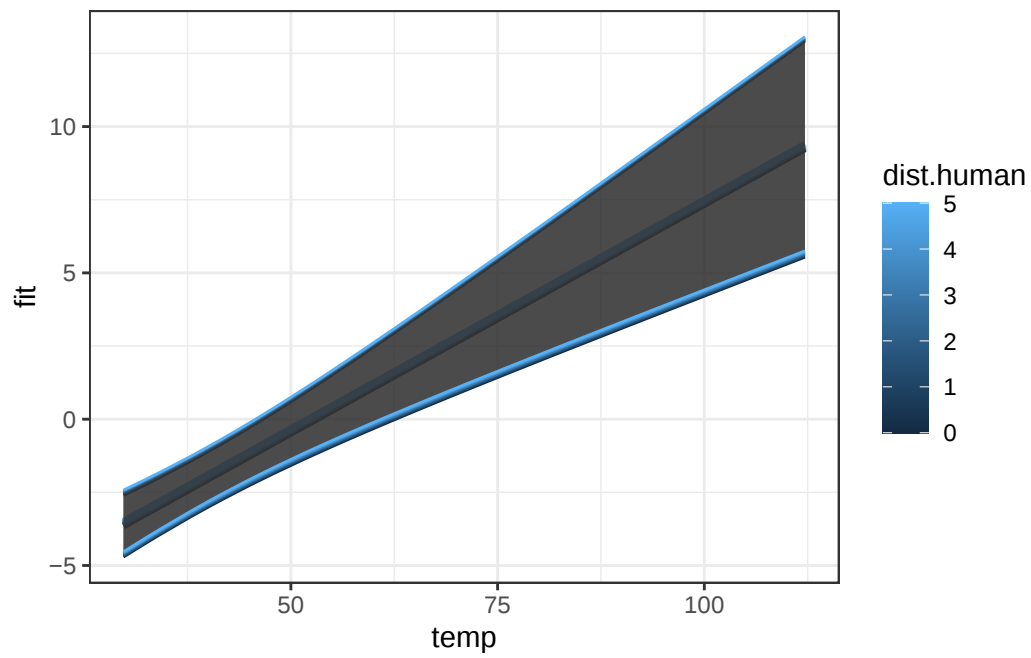
Taking Control

```
newdata = expand.grid(dat$temp,0:5)
colnames(newdata)=c("temp","dist.human")

preds = predict(model,newdata = newdata,interval="confidence", level=0.95)

pred.plot = data.frame(newdata,preds)
```

Combined Predictions 2



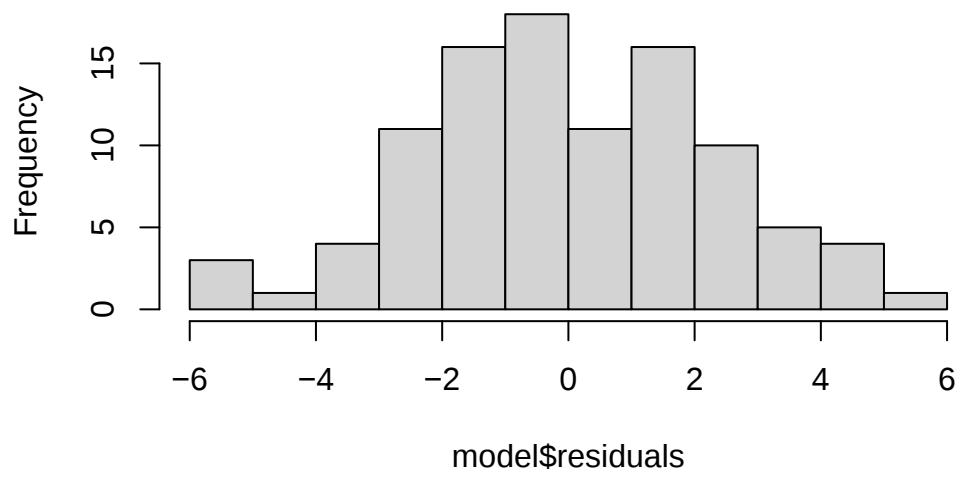
Evaluating Assumptions

Largely done based on the residuals

$$y_i - \hat{y}_i$$

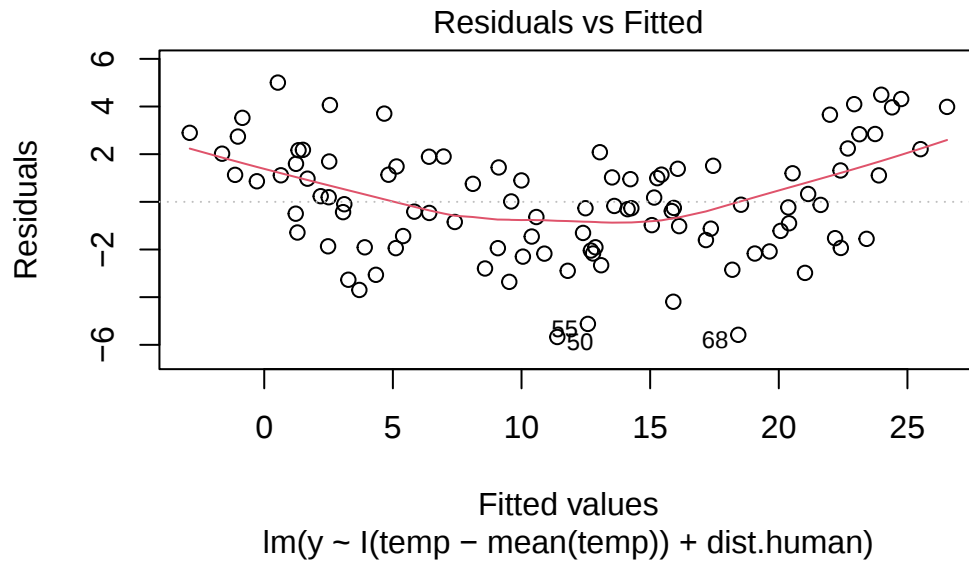
```
hist(model$residuals)
```

Histogram of model\$residuals



Linearity Assumption

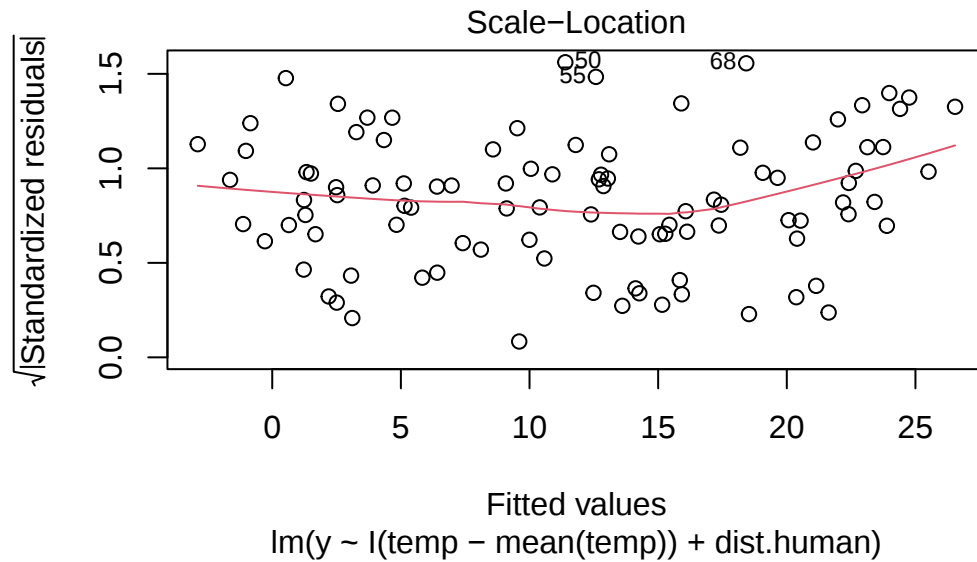
```
plot(model, 1)
```



Ideally, there will be no pattern and the red line should be roughly horizontal near zero.

Homogeneity of Variance

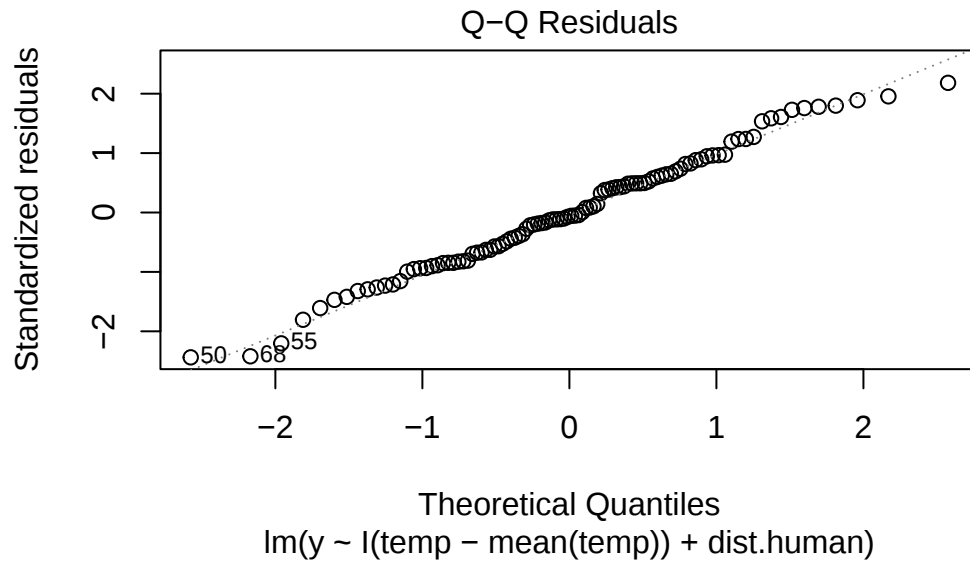
```
plot(model, 3)
```



Residuals should be spread equally along the ranges of predictors. We want a horizontal red line; otherwise, suggests a non-constant variances in the residuals (i.e., heteroscedasticity).

Normality of Residuals

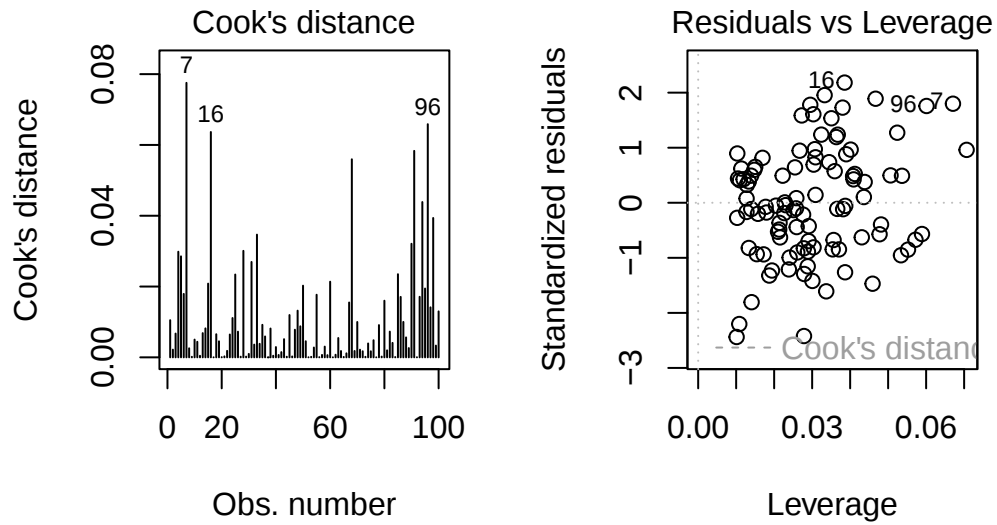
```
plot(model, 2)
```



Shows theoretical quantiles versus empirical quantiles of the residuals. We want to see circles on the dotted line.

Outliers

```
par(mfrow=c(1,2))
plot(model,4)
plot(model,5)
```



Outlier: extreme value that can affect the β estimate. Leverage plot: points in the upper right and lower right corner.

Exploring Assumptions

Nicer looking Plots

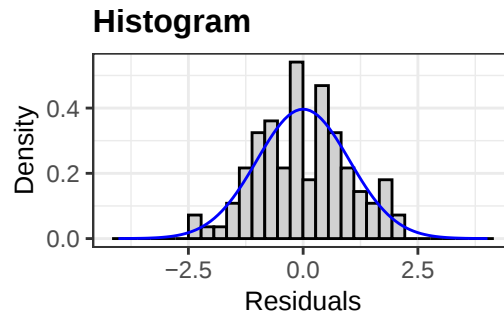
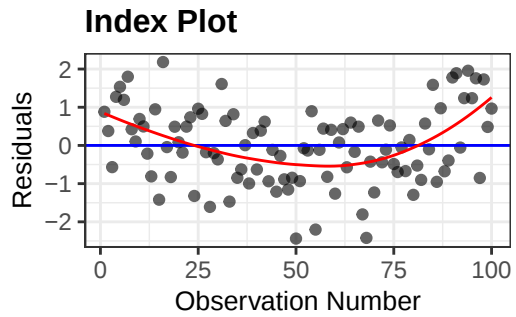
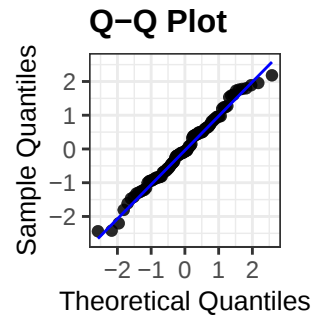
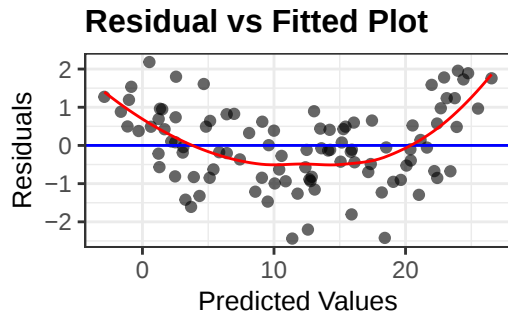
```
library(ggResidpanel)
```

```
Attaching package: 'ggResidpanel'
```

```
The following object is masked from 'package:datasets':
```

```
penguins
```

```
resid_panel(model)
```



Go to Lab