Introduction to EDA

Pramod Duvvuri

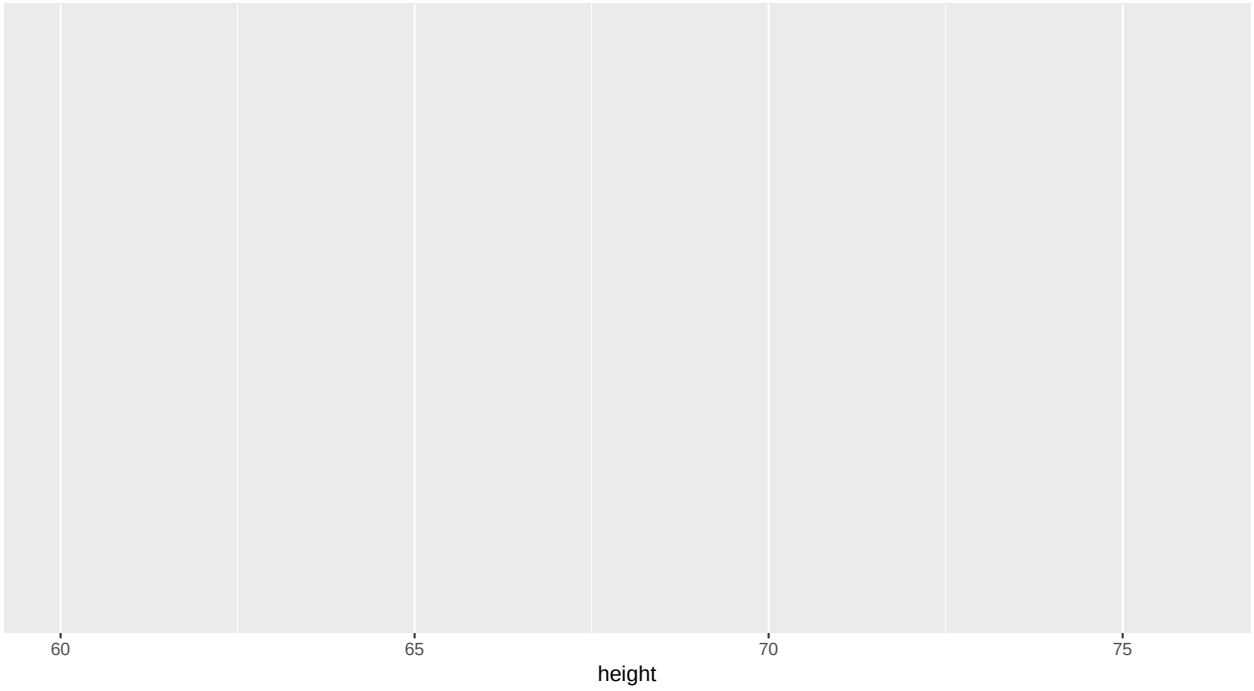
3/11/2019

Univariate Data

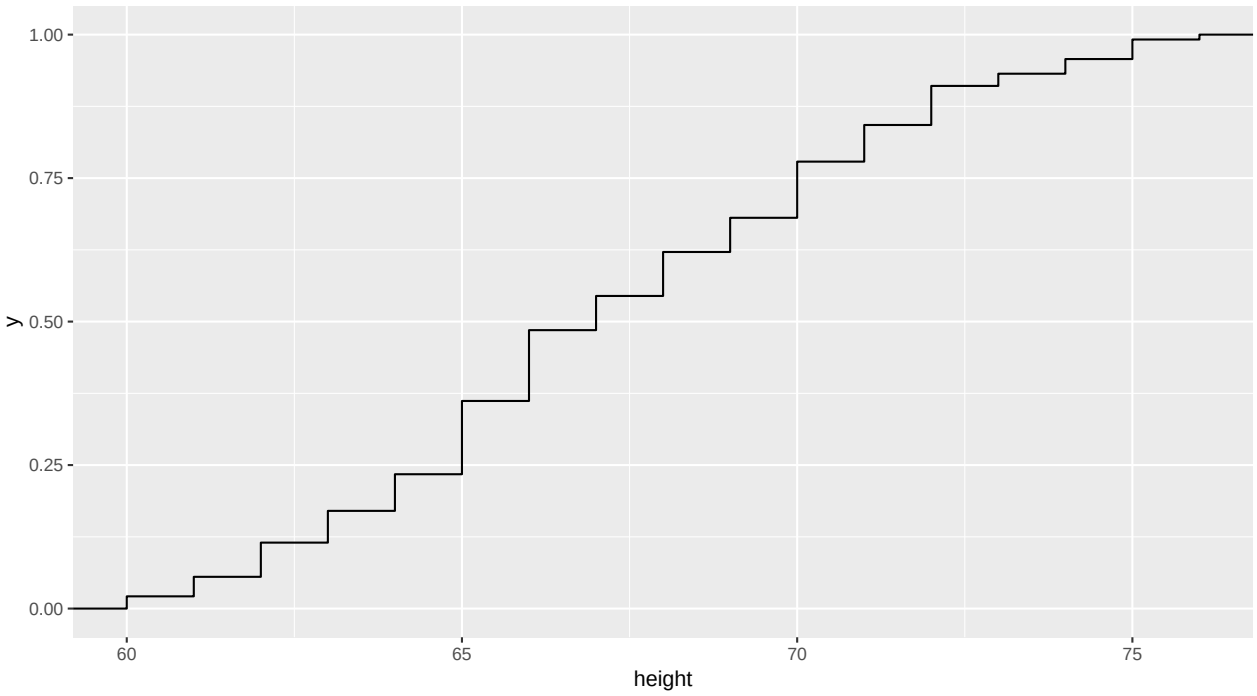
Learning ggplot2

```
# Load installed packages
library('lattice')
library('ggplot2')
```

```
ggplot(singer, aes(x = height))
```

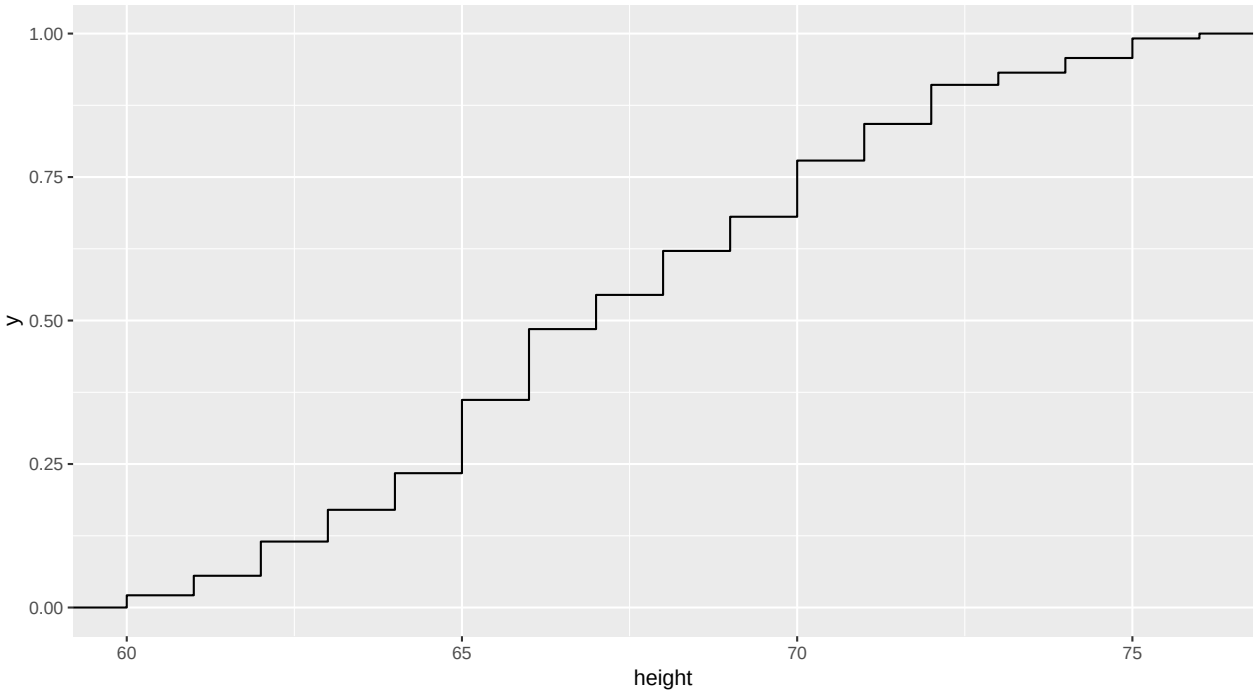



```
# ECDF in ggplot2
ggplot(singer, aes(x = height)) + stat_ecdf()
```

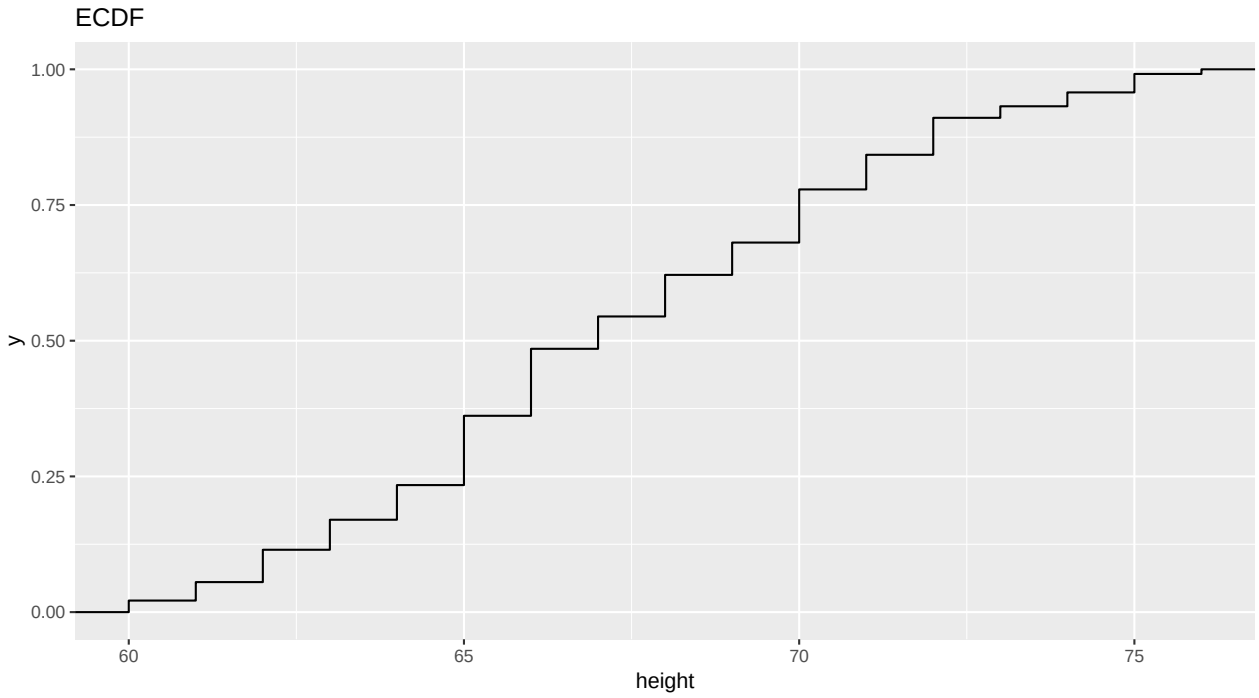


Basic Plots

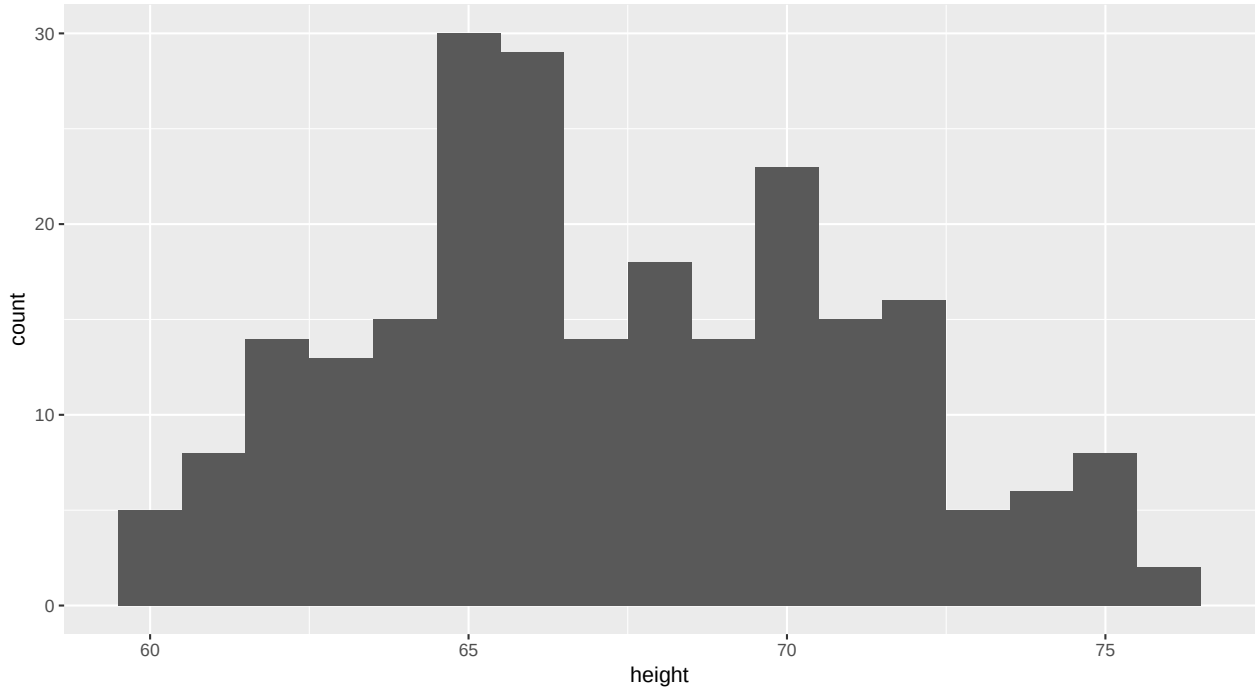
```
# Alternate way to get ECDF
singer_gg <- ggplot(singer, aes(x = height))
singer_gg + stat_ecdf()
```



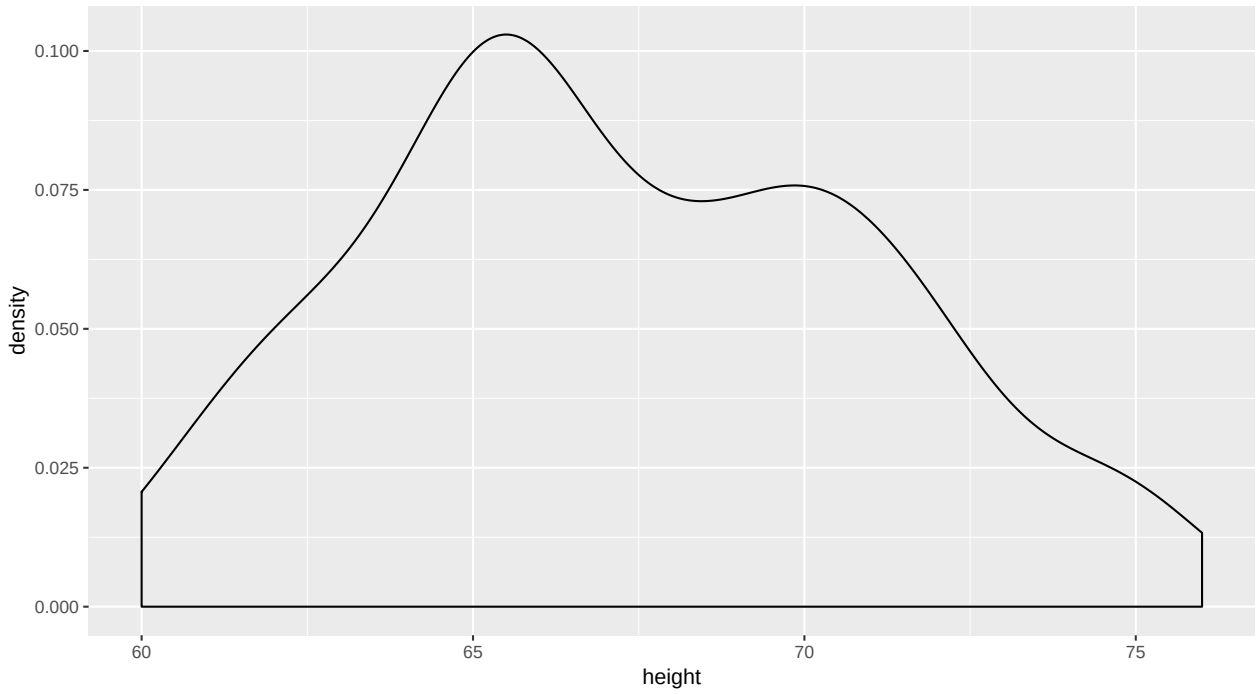

```
# Add Labels
singer_gg + stat_ecdf() + ggtitle('ECDF')
```



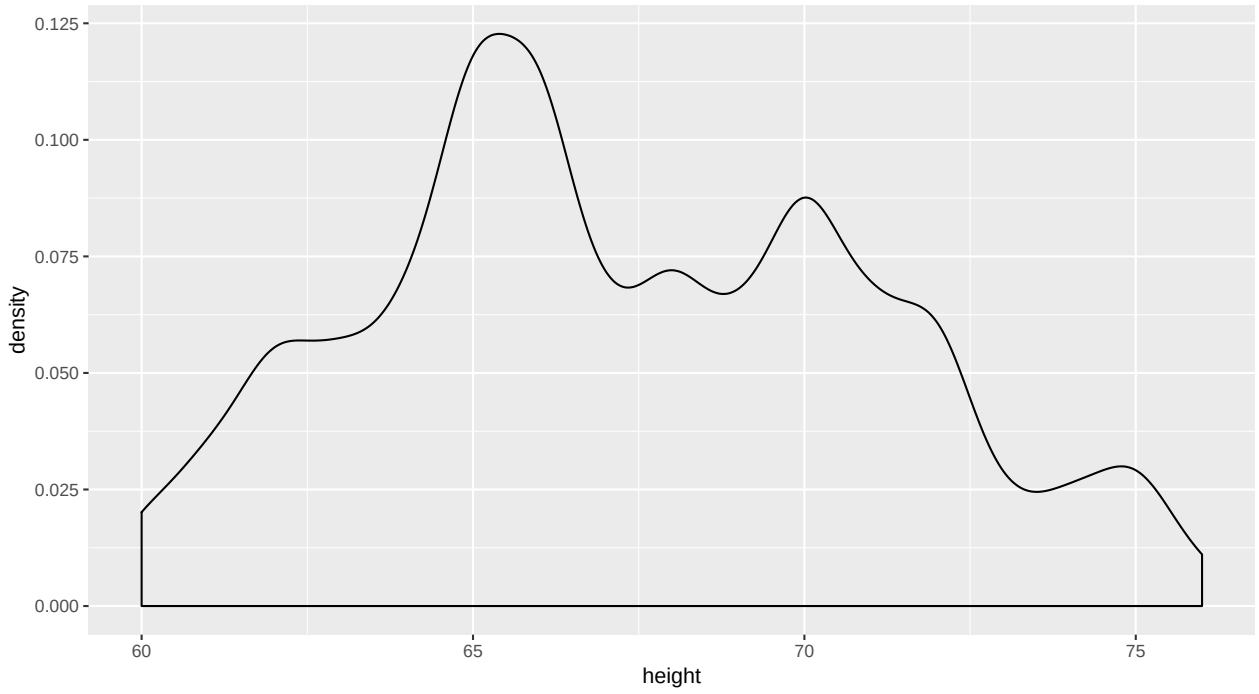

```
# Histogram
#ggplot(singer, aes(x= height)) + geom_histogram()
ggplot(singer, aes(x= height)) + geom_histogram(binwidth = 1)
```



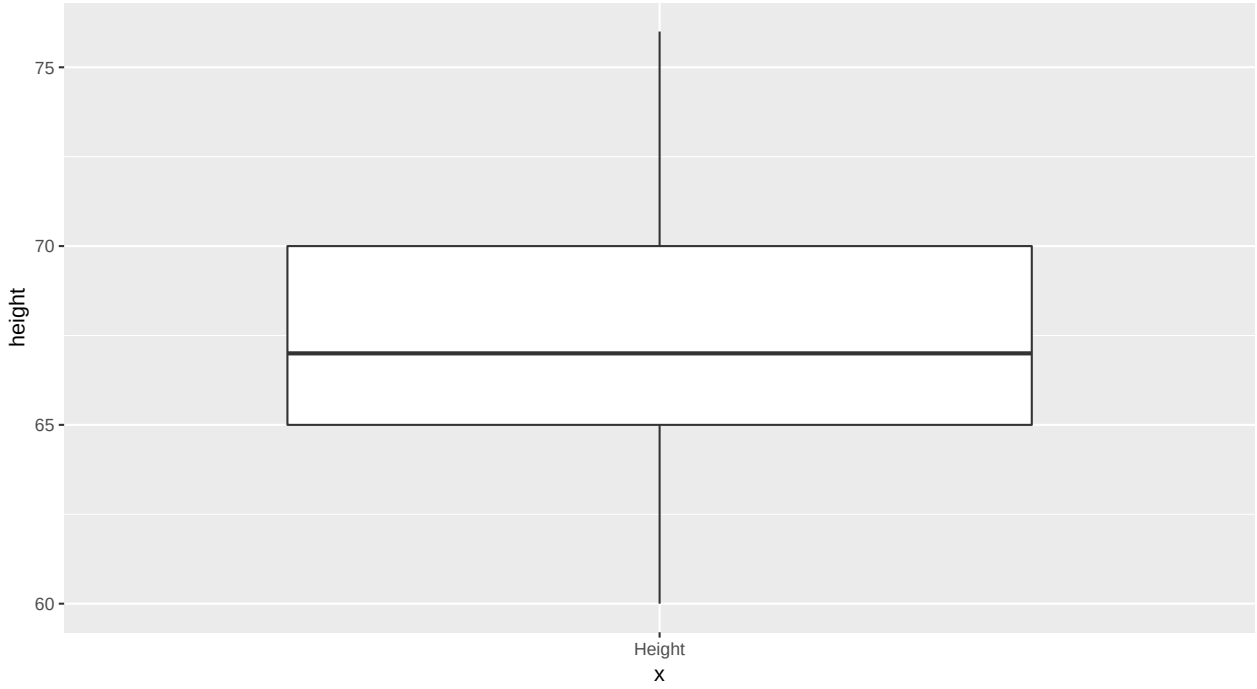

```
# Density Plot
ggplot(singer, aes(x= height)) +
  geom_density()
```



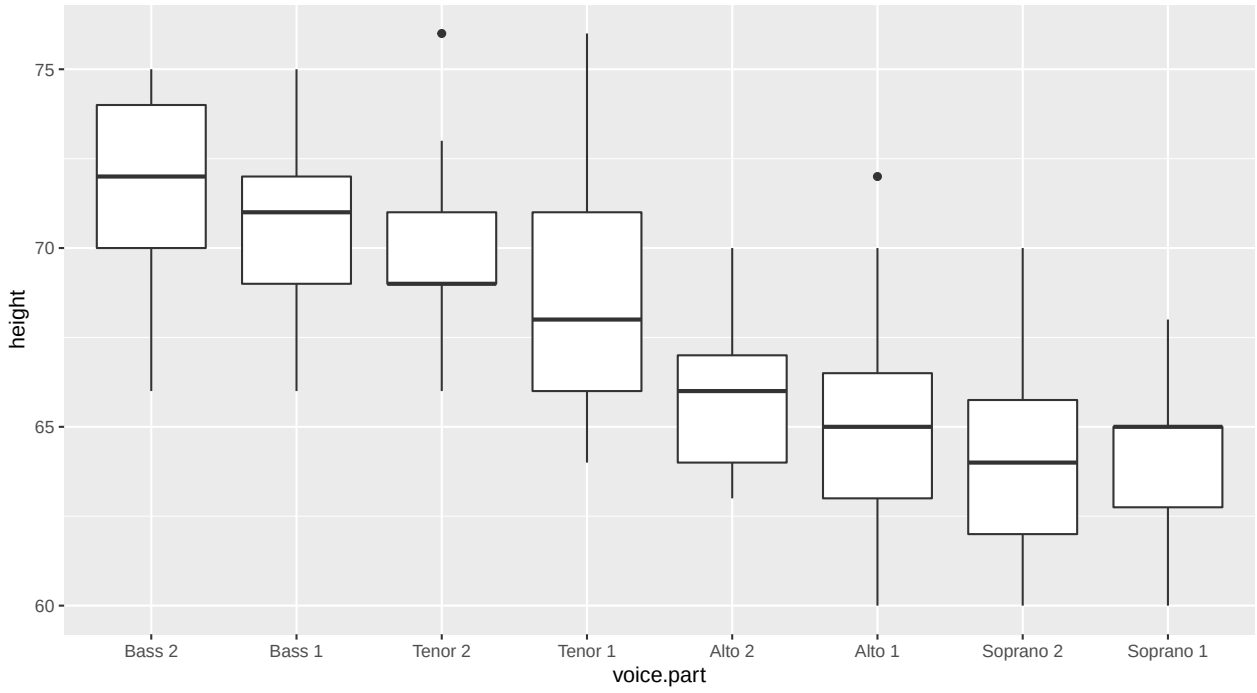

```
ggplot(singer, aes(x= height)) +
  geom_density(adjust = 0.5)
```



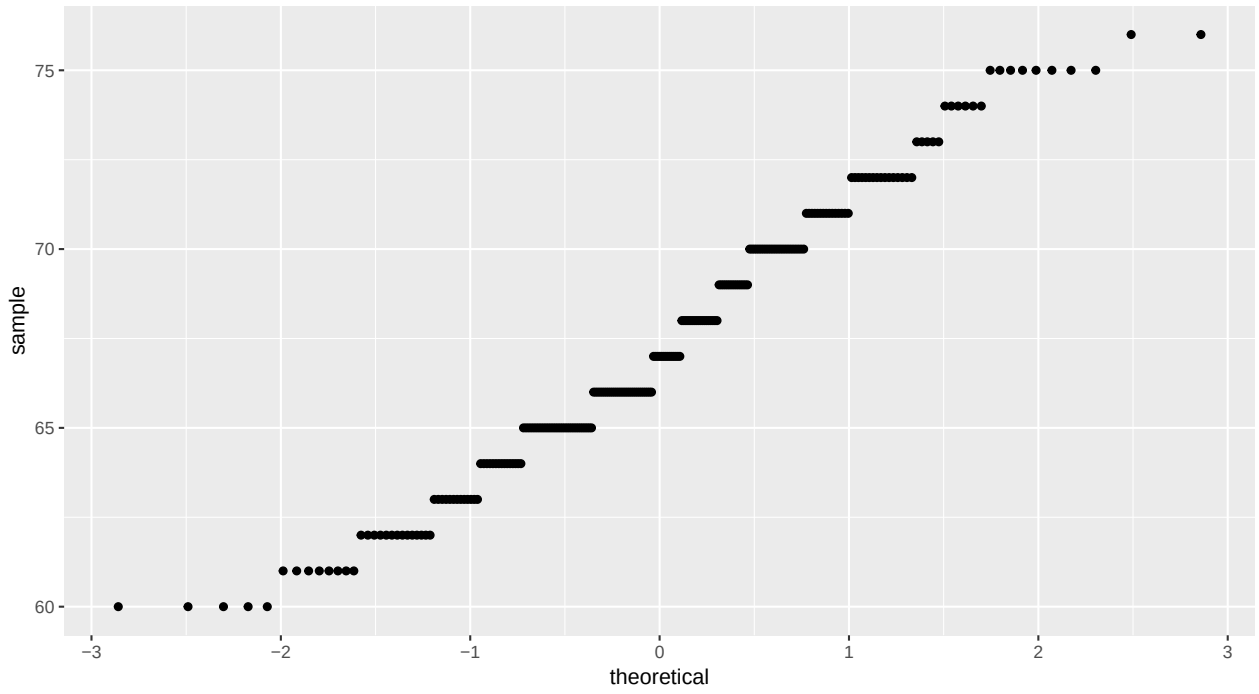

```
# Boxplot
ggplot(singer, aes(x= "Height", y = height)) + geom_boxplot()
```



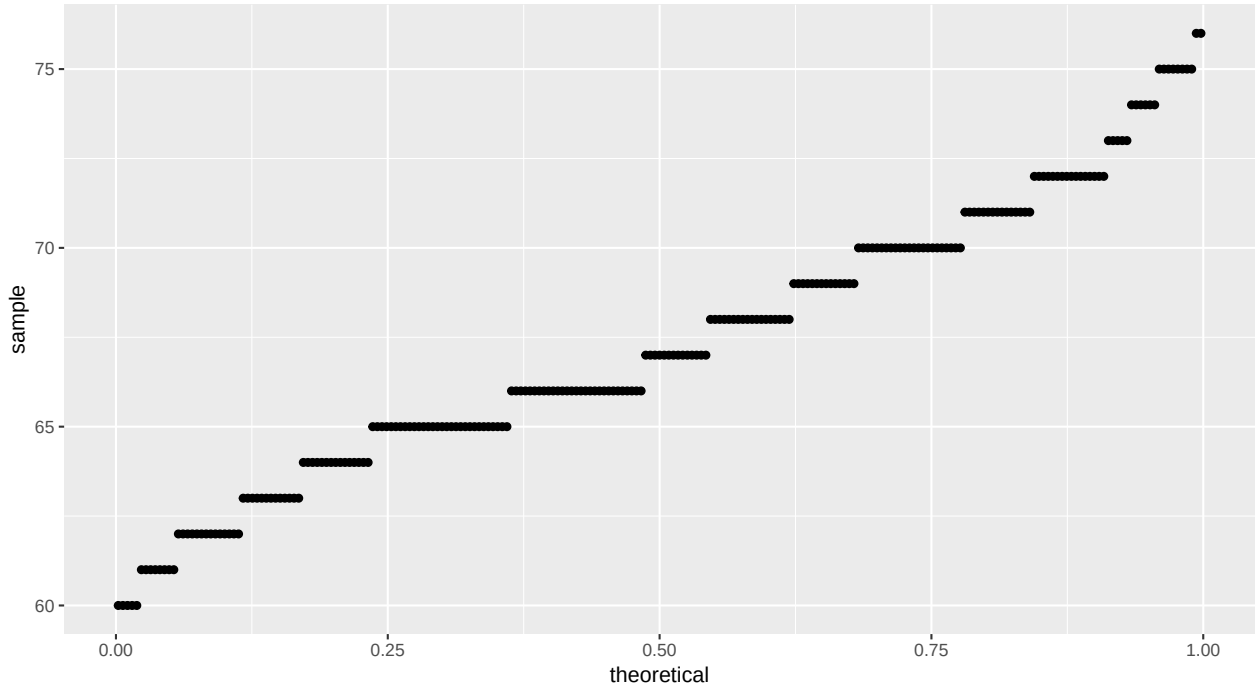

```
# Multiple Boxplots
ggplot(singer, aes(x = voice.part, y = height)) + geom_boxplot()
```



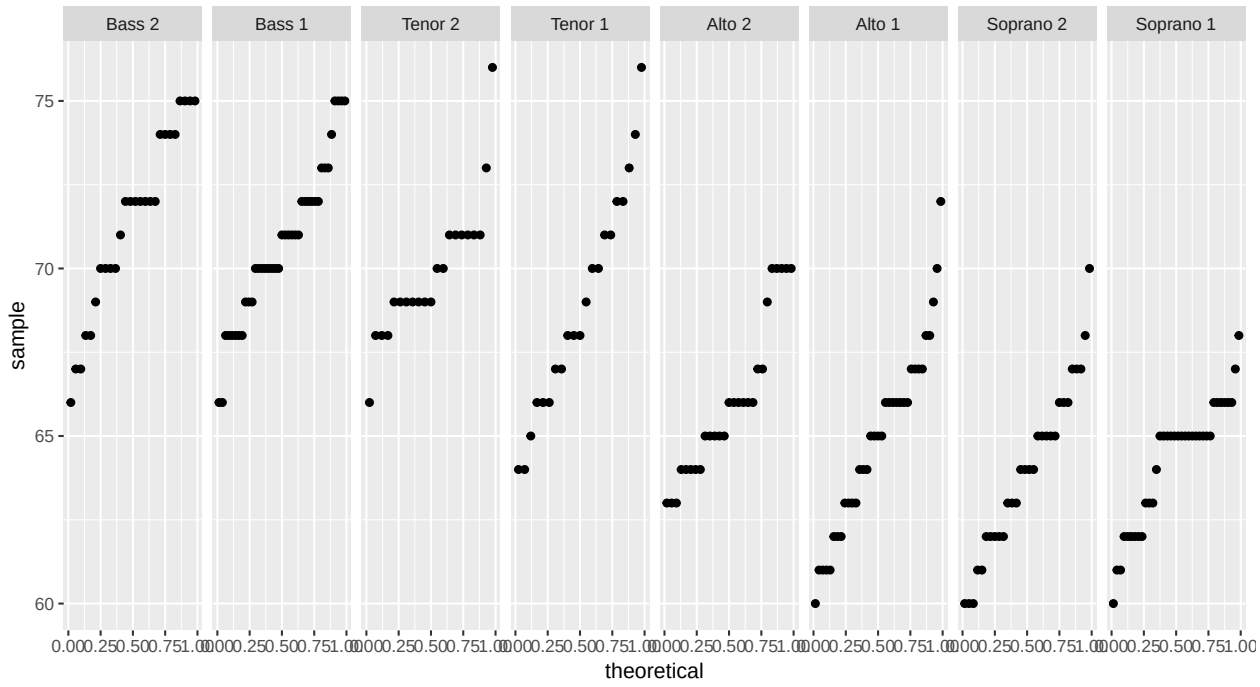

```
# QQ-Plot using a Normal Distribution
ggplot(singer, aes(sample = height)) + stat_qq()
```



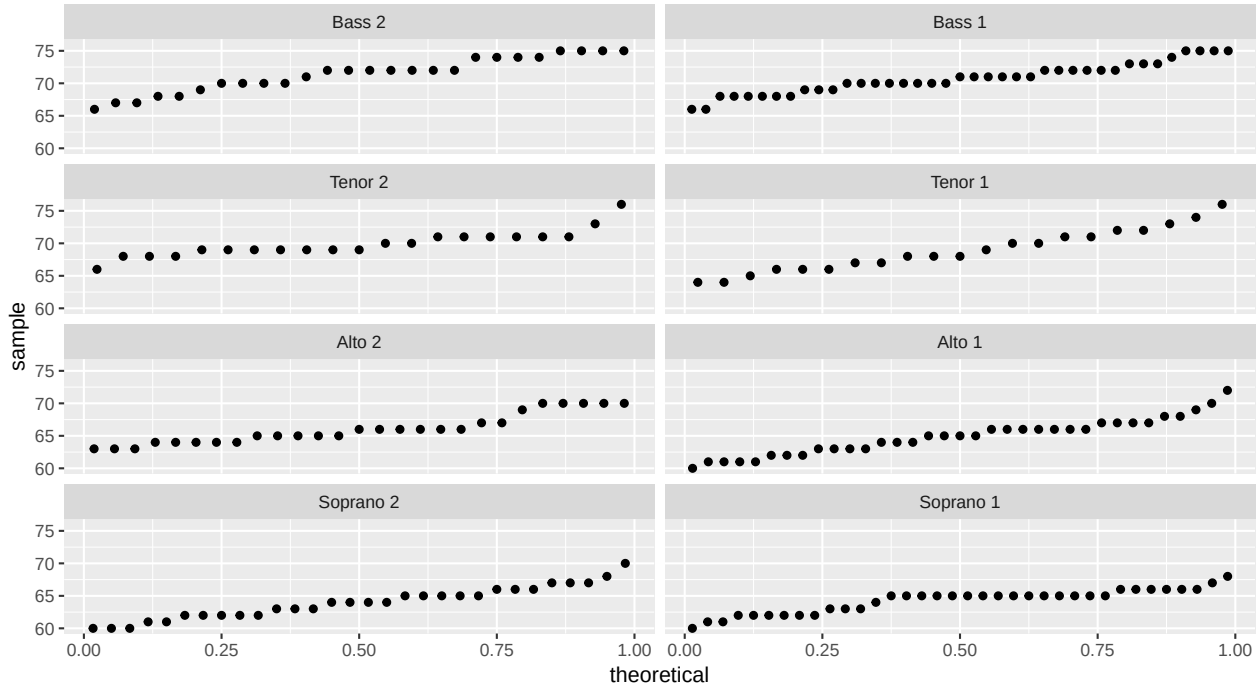

```
# QQ-Plot using a Uniform Distribution
ggplot(singer, aes(sample = height)) +
  stat_qq(distribution = qunif)
```




```
# Facet
ggplot(singer,aes(sample = height)) +
  stat_qq(distribution = qunif) +
  facet_grid(~voice.part)
```

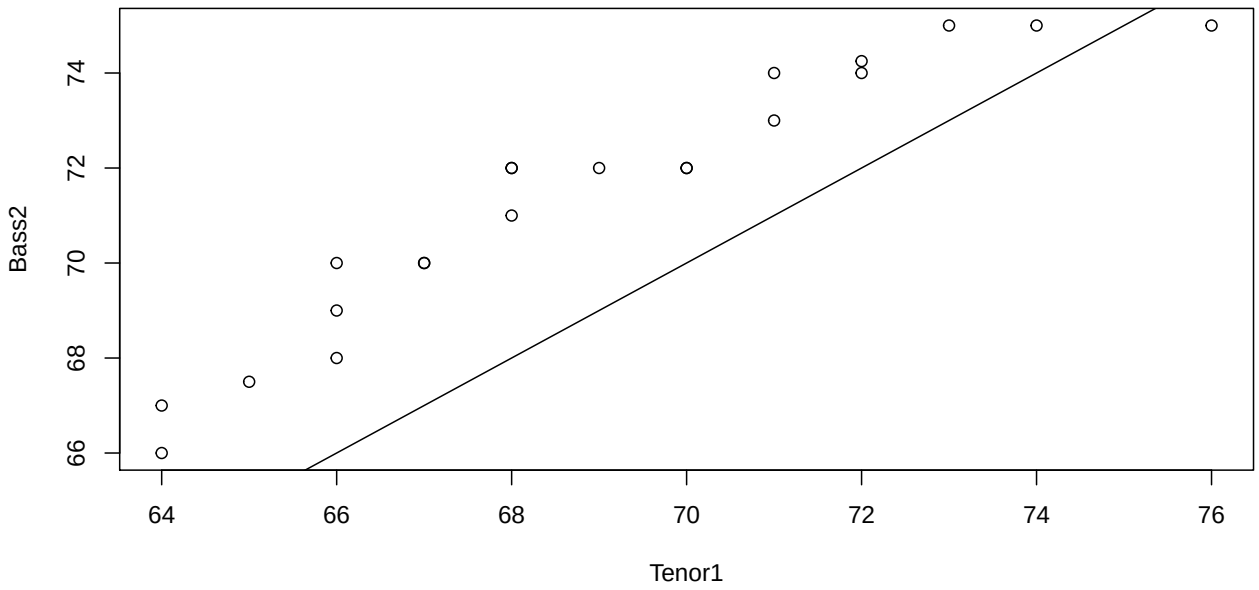



```
# The plot above looks cramped, so lets try a grid with columns
# Facet with grid display
ggplot(singer, aes(sample = height)) +
  stat_qq(distribution = qunif) +
  facet_wrap(~voice.part, ncol = 2)
```

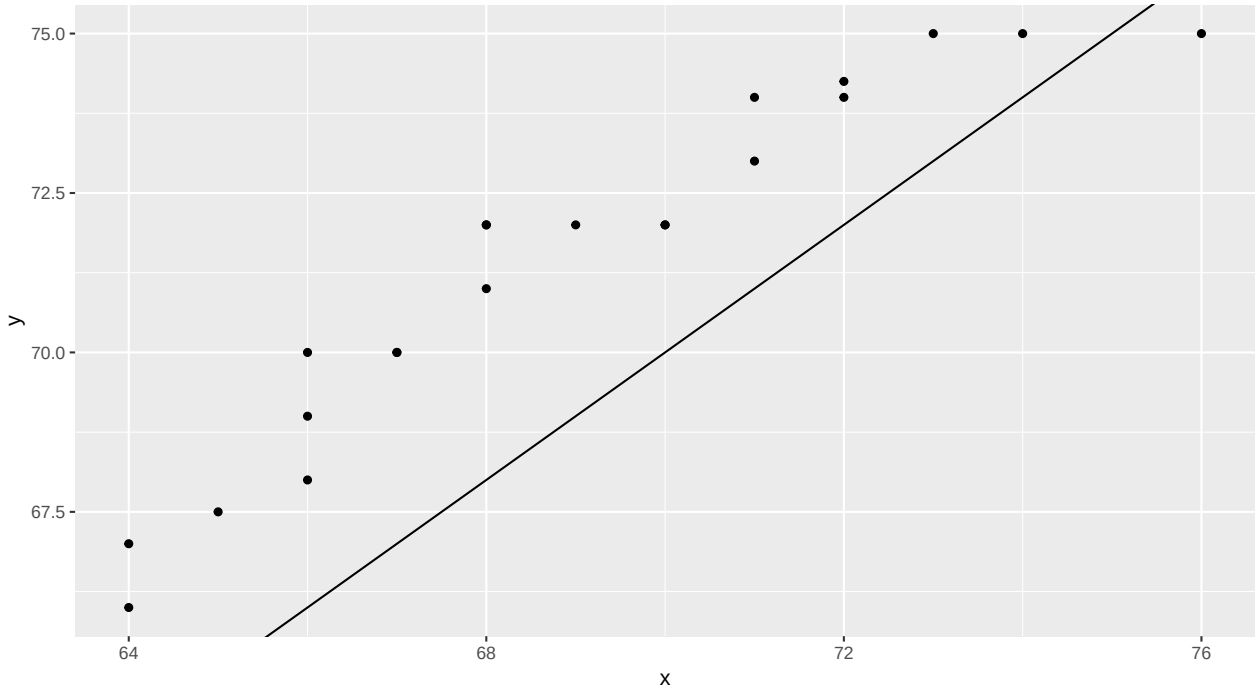



```
# QQ-Plot using Base R function qqplot()
Tenor1 = singer$height[singer$voice.part == "Tenor 1"]
Bass2 = singer$height[singer$voice.part == "Bass 2"]
qqplot(Tenor1, Bass2)
```

```
abline(0, 1)
```

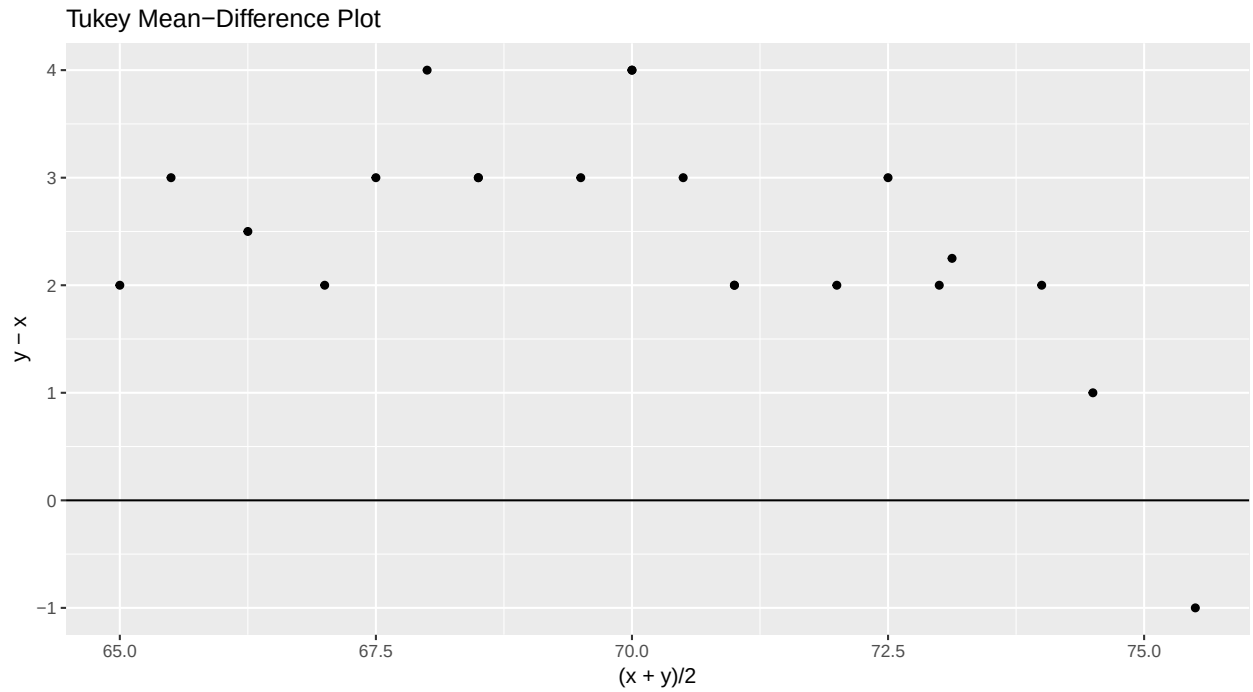



```
# Using ggplot
#library(tibble)
#qq_df <- as.tibble(qqplot(Tenor1, Bass2, plot.it = FALSE))
qq_df <- as.data.frame(qqplot(Tenor1, Bass2, plot.it = FALSE))
ggplot(data = qq_df, mapping =
         aes(x = x, y = y)) +
  geom_point() +
  geom_abline()
```



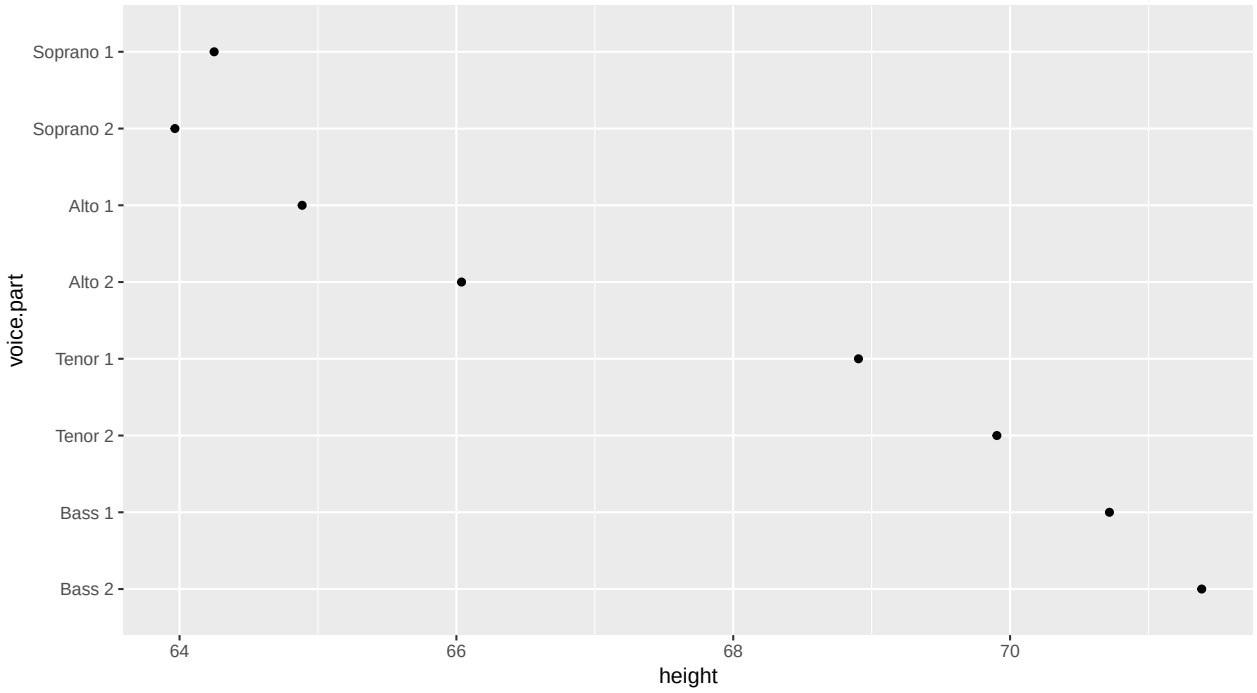

```
## Tukey-Mean difference Plot
ggplot(data = qq_df, mapping = aes(x = (x + y)/2, y = y - x)) +
```

```
  geom_point() +
  geom_abline(slope = 0) + ggtitle("Tukey Mean-Difference Plot")
```




```
singer_means = aggregate(height ~ voice.part, FUN = mean, data = singer)
```

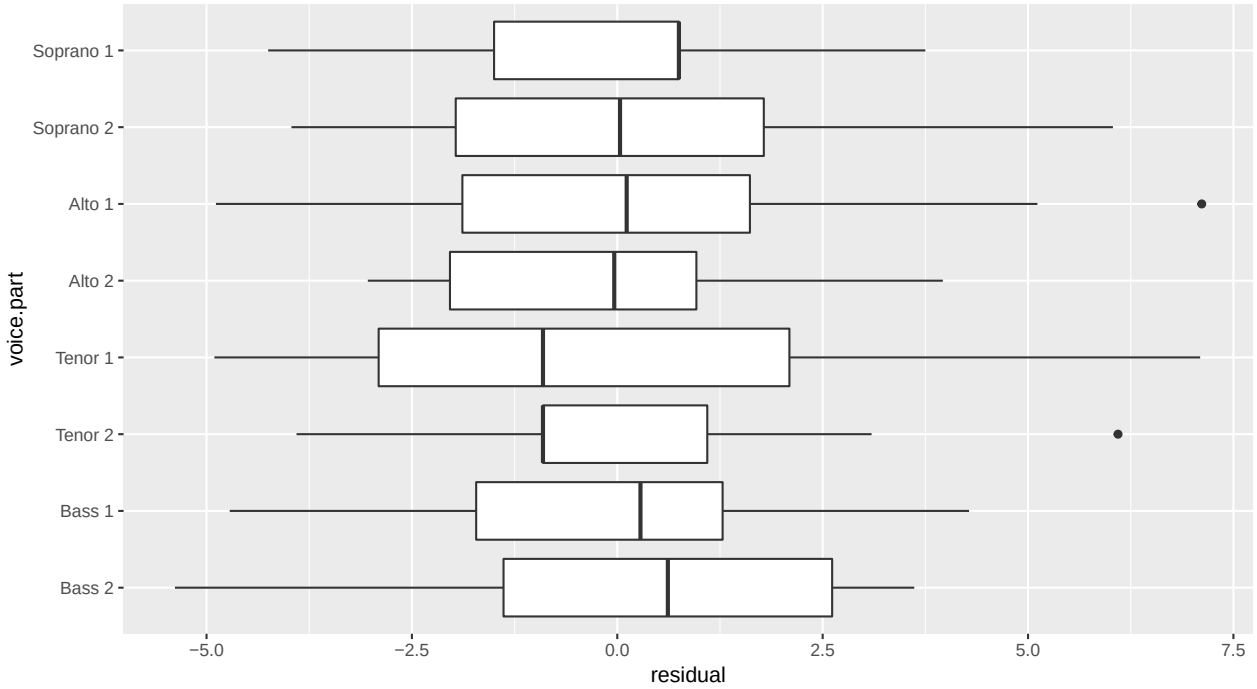
```
ggplot(singer_means, aes(x = voice.part, y = height)) + geom_point() + coord_flip()
```



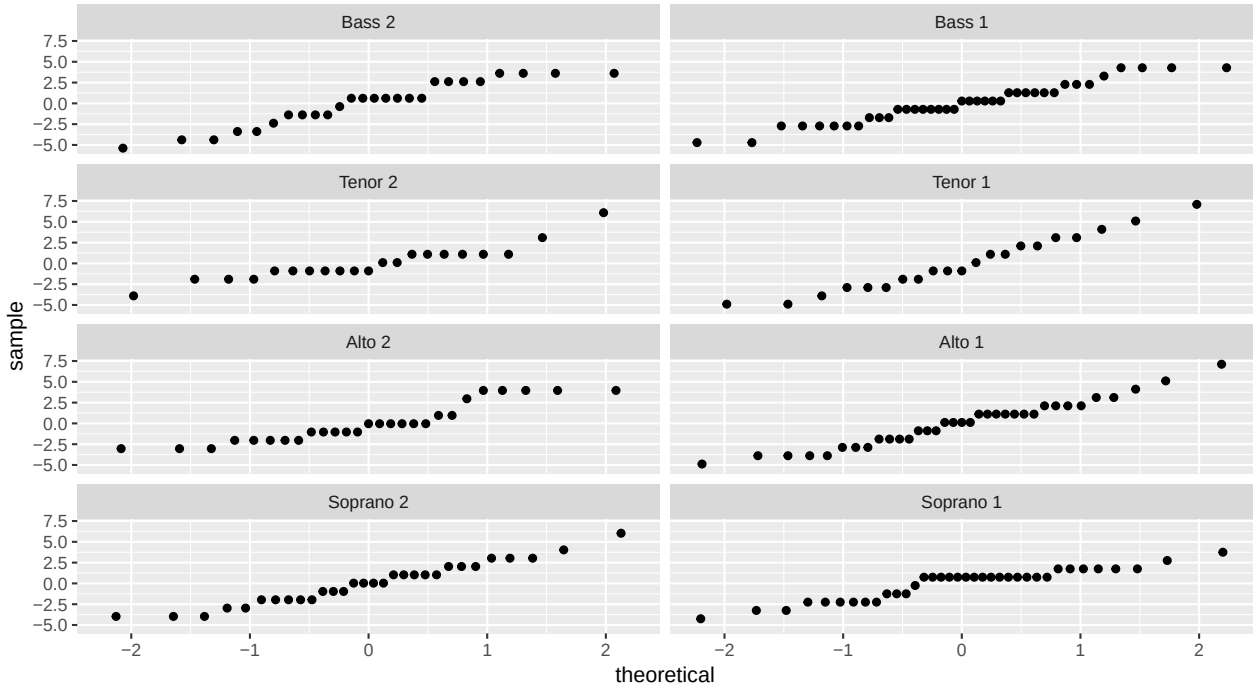

```
# Fitting a linear model
singer.lm = lm(height ~ voice.part, data = singer)
```

```r
# Extracting residual values
singer.res = data.frame(voice.part = singer$voice.part, residual = residuals(singer.lm))
```

```r
# Observing Residuals using boxplots
ggplot(singer.res, aes(x = voice.part, y = residual)) + geom_boxplot() + coord_flip()
```




```r
# Checking normality of residuals
ggplot(singer.res, aes(sample = residual)) +
  stat_qq() +
  facet_wrap(~voice.part, ncol = 2)
```



```r
# QQ Plot for Normality
ggplot(singer.res, aes(sample = residual)) +
  stat_qq()
```