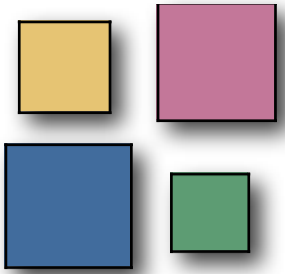


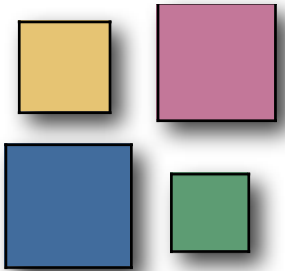
R Pakete

- Die eigentliche R Programmier-Sprache ist zwar schon recht umfangreich, bietet aber nur wenige statistische Funktionen.
- Nicht elementare Funktionen werden in R in sog. Paketen organisiert.
- Die R Standard-Installation kommt bereits einer Menge von Paketen, die für den “normalen” R Benutzer ausreichen sollte.
- Im Augenblick (31.1.2006) gibt es 674 zusätzliche Pakete zur R Installation.
- Es gibt **KEINE** Garantie für die Richtigkeit der Funktionen, die in nicht standard R Paketen angeboten werden.
- Nur die Standard-Pakete werden beim Start von R automatisch geladen, alle anderen muss der Benutzer selber laden.
- GUIs (z.B. JGR) können bei der Verwaltung der Pakete sehr hilfreich sein.



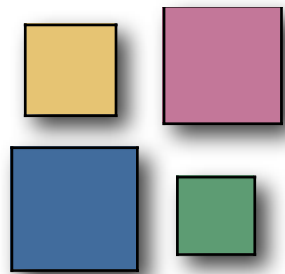
R Pakete verwalten:

- Die wichtigsten Funktionen um Pakete in R zu verwalten sind:
 - **library(mypackage)**
mypackage aktivieren
 - **detach("package:mypackage")**
mypackage aus dem Suchpfad entfernen
 - **installed.packages()**
listet die installierten Pakete
- Die Installation von Paketen ist je nach Art des Paketes (Source oder Binary) unterschiedlich.
 - **install.packages("mypackage")**
installiert das binary package mypackage von CRAN (<http://cran.r-project.org/>)
 - **\$ R CMD INSTALL mypackage.tgz**
installiert das source package lokal aus der Shell heraus.
- Die meisten GUIs können dem Benutzer diese Arbeit abnehmen.



Modelle in R

- Statistische Modelle können in R mit der sog. Formelsprache (siehe Folie 117) definiert werden.
- Voraussetzung ist im Normalfall, dass die Daten in einem Dataframe gespeichert sind.
- Die wichtigsten Modellklassen sind:
 - **lm()** einfaches lineare Modell
 - **glm()** generalisiertes lineares Modell
 - **anova()** Varianzanalyse
 - **gam()** generalisiertes additives Modell
- Alle Funktionen rund um diese Modelle sind generisch, d.h. es existieren für alle Typen
 - **print()** Ausgabe
 - **summary()** Zusammenfassung
 - **predict()** Vorhersage
 - **add()** Hinzufügen von Termen
 - **drop()** Entfernen von Termen
 - **step()** schrittweise Änderung eines Modells
 - ...



Modelle in R: Beispiel

- Lineares Modell für die Cars Daten:

```
> names(Cars)
[1] "mpg" "cylinders" "displacement" "horsepower" "weight"
[6] "acceleration" "model.year" "origin" "car.name"
```

```
> attach(Cars)
```

```
> plot(horsepower, mpg)
```

```
> l1 <- lm(mpg ~ horsepower)
```

```
> summary(l1)
```

Call:

```
lm(formula = mpg ~ horsepower)
```

...

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	39.935860	0.716580	55.73	<2e-16 ***
horsepower	-0.157845	0.006436	-24.53	<2e-16 ***

Multiple R-Squared: 0.606, Adjusted R-squared: 0.605 ...

```
> abline(l1)
```

```
#
```

```
# Switch from mpg to consumption
```

```
#
```

```
> Verbrauch <- 1/(mpg*1.6/3.84)*100
```

```
> plot(horsepower, Verbrauch)
```

```
> abline(lm(Verbrauch ~ horsepower))
```

```
> summary(lm(Verbrauch ~ horsepower))
```

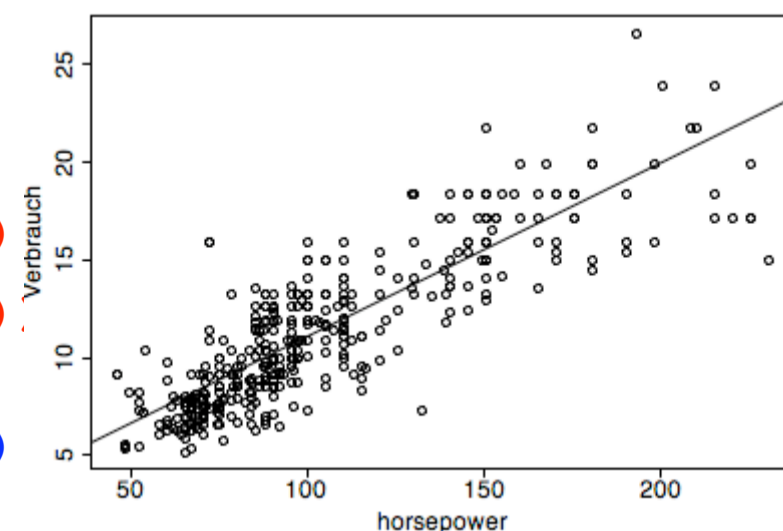
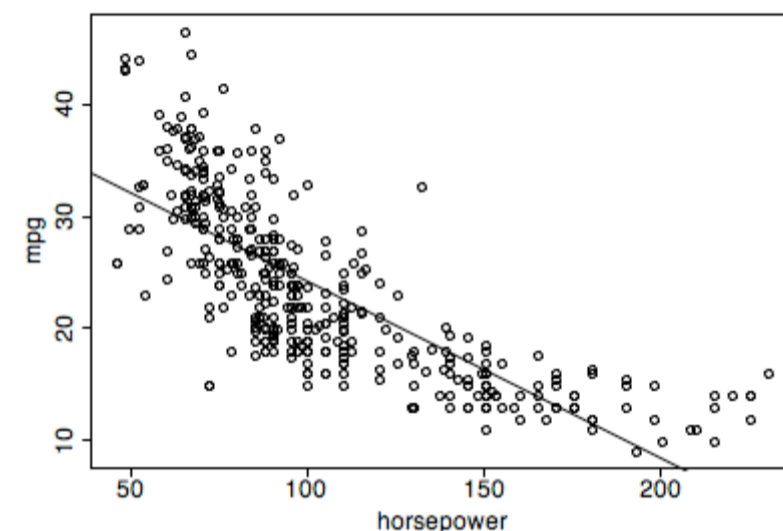
Call:

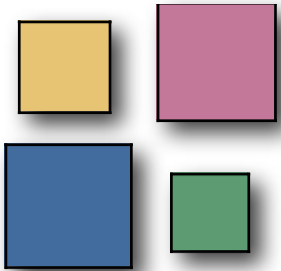
```
lm(formula = Verbrauch ~ horsepower)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.212708	0.303282	7.296	1.67e-12 ***
horsepower	0.088648	0.002724	32.545	< 2e-16 ***

Multiple R-Squared: 0.7304, Adjusted R-squared: 0.7297





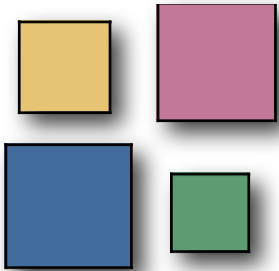
Stat. Tests

- Für statistische Tests gibt es leider kein einheitliches Framework, wie es für Modelle vorhanden ist.
- Dennoch sollten für Tests zumindest die generische `summary()` Funktion implementiert sein.
- Der Output der meisten Test Funktionen ist bedauerlicherweise oft relativ knapp und schwer zu lesen/interpretieren.
- Beispiel: t-test

```
> hist(mpg, freq=F)
> lines(density(mpg))
> qqnorm(mpg)
> t.test(mpg[origin=="U.S."], mpg[origin!="U.S."])
```

```
Welch Two Sample t-test
```

```
data: mpg[origin == "U.S."] and mpg[origin != "U.S."]
t = -13.6876, df = 307.782, p-value < 2.2e-16
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -10.48230  -7.84728
sample estimates:
mean of x mean of y
 20.08353  29.24832
```



Stat. Tests: Liste

- Wichtige Tests die in R verfügbar sind:
 - **binom.test**
exakter Binomialtest (Ein-Stichprobenproblem)
 - **chisq.test**
 χ^2 -Test für Kontingenztafeln
 - **cor.test**
Gepaarte Stichprobe, $H_0 : \rho = 0$, Bravais-Pearson, Spearman (u.a.)
 - **fisher.test**
Exakter Test von Fisher
 - **t.test**
t-Test, Ein- und Zweistichproben-Fall, sowie gepaarter t-Test
 - **var.test**
F-Test für den Vergleich der Varianzen zweier Stichproben
 - **bartlett.test**
Test auf Gleichheit der Varianzen in mehreren Stichproben
 - **wilcox.test**
Wilcoxon-Test, Ein- und Zweistichproben-Fall
 - **kruskal.test**
Kruskal-Wallis-Rangsummen-Test
 - **ks.test**
Kolmogorov-Smirnov-Test, Ein- und Zwei-Stichproben-Fall
 - **friedman.test**
Friedman-Rangsummen-Test
 - ...